

UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA DE PRODUÇÃO

**OTIMIZAÇÃO IRRESTRITA MONO-OBJETIVO  
POR ENXAME DE PARTÍCULAS ASSISTIDA  
POR POLINÔMIOS CANÔNICOS DE MISTURAS**

**Wesley Gabriel de Mendonça Pinto**

Itajubá, Junho de 2018

UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA DE PRODUÇÃO

**Wesley Gabriel de Mendonça Pinto**

**OTIMIZAÇÃO IRRESTRITA MONO-OBJETIVO  
POR ENXAME DE PARTÍCULAS ASSISTIDA  
POR POLINÔMIOS CANÔNICOS DE MISTURAS**

Dissertação submetida ao Programa de Pós-Graduação  
em Engenharia de Produção como parte dos requisitos  
para obtenção do título de **Mestre em Ciências em  
Engenharia de Produção.**

**Área de Concentração:** Qualidade e Produtos

**Orientador:** Prof. Dr. Anderson Paulo de Paiva

Itajubá, Junho de 2018

UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA DE PRODUÇÃO

**Wesley Gabriel de Mendonça Pinto**

**OTIMIZAÇÃO IRRESTRITA MONO-OBJETIVO  
POR ENXAME DE PARTÍCULAS ASSISTIDA  
POR POLINÔMIOS CANÔNICOS DE MISTURAS**

Dissertação aprovada por banca examinadora em 28  
de abril de 2018, conferindo ao autor o título de  
**Mestre em Ciências em Engenharia de Produção.**

**Banca examinadora:**

Prof. Dr. Anderson Paulo de Paiva

Prof. Dr. Rafael Coradi Leme

Prof. Dr. Rogério Santana Peruchi

Itajubá, Junho de 2018

## DEDICATÓRIA

*À Ronilde, minha querida mãe e minha esposa Ana Paula.*

## **AGRADECIMENTOS**

Primeiramente a Deus, o grande engenheiro do universo.

Em especial ao meu orientador, Prof. Dr. Anderson Paulo Paiva, pelo incentivo, paciência, amizade e disponibilidade a toda prova. Pela sua capacidade de ensinar mesmo no silêncio, que supera o aprendizado acadêmico. Muito obrigado!

À minha esposa Ana Paula pelo incentivo, aos meus pais e irmãos pelo apoio incondicional.

Aos professores do Programa de Pós-graduação em Engenharia de Produção e a todos os funcionários da secretaria da PRPPG e do DRE, o meu muito obrigado.

Aos colegas do GEPE de Qualidade com os quais compartilhei dúvidas, cafés, conversas e risadas que tornaram o ambiente mais agradável e produtivo.

À CAPES e à UNIFEI pelo apoio financeiro e estrutural à pesquisa brasileira, que através do programa "Demanda Social" viabilizaram a realização deste trabalho.

Enfim, a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

## EPÍGRAFE

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas graças a Deus, não sou o que era antes”.

Marthin Luther King.

## RESUMO

A dificuldade de encontrar soluções eficientes para problemas complexos de otimização tem levado diversos pesquisadores a desenvolverem e utilizarem ferramentas computacionais como algoritmos, a fim de auxiliar na resolução de problemas de otimização. Uma das dificuldades encontradas para se utilizar um algoritmo de otimização é a maneira como se deve configurá-lo, uma vez que, configurado de modo incorreto pode influenciar no desempenho do algoritmo, levando-o a soluções inviáveis. Desta forma, o presente trabalho tem por objetivo melhorar a performance de um algoritmo de otimização conhecido como enxame de partículas (PSO), visando calibrar os parâmetros de configuração do algoritmo, com intuito de encontrar um ajuste próximo do ideal melhorando a eficiência e eficácia deste otimizador. Para tal, foi aplicado um método denominado planejamento de experimentos (DoE), que possibilita encontrar parâmetros significativos que influenciam na performance do ambiente modelado, além de proporcionar soluções viáveis para resolução final de determinados sistemas. O procedimento proposto foi aplicado na otimização das funções esférica, rosenbrock e rastrigin, respectivamente, por se tratarem de funções contínuas, de sentido de minimização para otimização e duas dimensões. A utilização deste procedimento proporcionou uma nova configuração aos parâmetros do algoritmo enxame de partícula (*PSO*), ou seja, cada função de teste utilizada recebeu parâmetros únicos após a otimização do algoritmo. Dessa forma, as respostas tempo e número de iteração coletadas de cada função apresentou resultados significativos quanto aos parâmetros encontrados por essa otimização do *PSO* em relação aos parâmetros sugeridos pela literatura.

**Palavras-chave:** Otimização, planejamento de experimentos, computação evolucionária, enxame de partículas.

## ABSTRACT

The difficulty of finding efficient solutions to complex optimization problems has led several researchers to develop and use computational tools as algorithms to help solve optimization problems. One of the difficulties encountered in using an optimization algorithm is the way to configure it, since, incorrectly configured, can influence the performance of the algorithm, leading to non-viable solutions. Thus, the present work aims to improve the performance of an optimization algorithm known as particle swarm (PSO), aiming to adjust the configuration parameters, in order to find an adjustment close to the ideal, improving the efficiency and effectiveness of this algorithm as well as in improving the quality of systems and / or processes also known as optimization problems. For this, a method called Experimental Planning (DoE) was applied, which allows finding significant parameters that influence the performance of the modeled environment, in addition to providing viable solutions for the final resolution of certain systems. The proposed procedure was applied in the optimization of test functions recognized by the literature in order to evaluate the parameters of the optimization algorithm. From this application it was possible to obtain an efficient and effective method in relation to the time and number of steps that the algorithm took to solve problems of optimization of tests highlighted in the literature. It is worth noting that the results obtained by the test functions in relation to the parameters of the algorithm were very influential, since the times and numbers of steps that the optimizer took to find the optimal solution presented a lower result than the parameters configured in the literature parameters configured in the literature.

**Keywords:** Evolutionary computation, particle swarm, design of experiments, optimization



## LISTA DE FIGURA

<b>Figura 1</b> - Taxonomia de ajuste de parâmetro do algoritmo .....	3
<b>Figura 2</b> - Fluxograma do algoritmo de PSO .....	8
<b>Figura 3</b> - Estrutura geral de um processo.....	12
<b>Figura 4</b> - Restrição da região do espaço de um arranjo de misturas. ....	16
<b>Figura 5</b> - Arranjo de Mistura Pura .....	17
<b>Figura 6</b> - Representação de um arranjo de misturas simplex <i>centroid</i> .....	18
<b>Figura 7</b> - Demonstração gráfica de arranjo de misturas combinado com variável de processo .....	21
<b>Figura 8</b> - Classificação da Pesquisa .....	29
<b>Figura 9</b> - Método proposto para aperfeiçoamento do PSO .....	32
<b>Figura 10</b> – Representação gráfica da função Esférica .....	37
<b>Figura 11</b> - Análise dos resíduos para a resposta distância .....	40
<b>Figura 12</b> - Encontro do parâmetro e solução ótimo da função esférica .....	41
<b>Figura 13</b> - Passos do processo de otimização da função esférica .....	42
<b>Figura 14</b> – Representação gráfica da função Rosenbrock .....	45
<b>Figura 15</b> - Análise dos resíduos para a resposta distância .....	48
<b>Figura 16</b> - Encontro do parâmetro e solução ótimo da função Rosenbrock .....	48
<b>Figura 17</b> - Passos do processo de otimização da função esférica .....	50
<b>Figura 18</b> – Representação gráfica da função Rastrigin.....	52
<b>Figura 19</b> - Análise dos resíduos para a resposta distância .....	56
<b>Figura 20</b> - Encontro do parâmetro e solução ótimo da função Rastrigin.....	56
<b>Figura 21</b> - Passos do processo de otimização da função esférica .....	60

## LISTA DE TABELAS

<b>Tabela 1</b> - Estrutura de desenvolvimento do trabalho .....	4
<b>Tabela 2 (a)</b> - Arranjo experimental gerado pelo método DoE.....	37
<b>Tabela 2 (b)</b> - Arranjo experimental gerado pelo método DoE.....	38
<b>Tabela 3</b> - Determinação dos coeficientes do modelo da Distância com sinal ruído .....	38
<b>Tabela 4</b> - Análise de variância para a resposta distância .....	39
<b>Tabela 5</b> - Configuração ótima encontrado após a otimização.....	41
<b>Tabela 6</b> - Análise <i>Two Sample-t</i> para os tempos.....	43
<b>Tabela 7</b> - Análise <i>Two Sample-t</i> para as iterações .....	44
<b>Tabela 8</b> - Comparação entre tempos e iterações na otimização da função esférica.....	44
<b>Tabela 9 (a)</b> - Resposta distância extraída.....	45
<b>Tabela 9 (b)</b> - Resposta distância extraída.....	45
<b>Tabela 10</b> - Determinação dos coeficientes do modelo da distância .....	46
<b>Tabela 11</b> - Análise de variância para a resposta distância .....	47
<b>Tabela 12</b> - Configuração ótima da otimização realizada na função <i>Rosembrock</i> .....	49
<b>Tabela 13</b> - Análise <i>Two Sample-t</i> dos tempos.....	50
<b>Tabela 14</b> - Análise <i>Two Sample-t</i> para as iterações .....	51
<b>Tabela 15</b> - Comparação dos tempos entre a configuração padrão x método atual .....	51
<b>Tabela 16</b> - Resposta distância extraída .....	53
<b>Tabela 17</b> - Determinação dos coeficientes do modelo da distância .....	53
<b>Tabela 18</b> - Remoção dos <i>outliers</i> das variáveis respostas.....	54
<b>Tabela 19 (a)</b> - Determinação dos coeficientes do modelo após remoção dos <i>outliers</i> .....	54
<b>Tabela 19 (b)</b> - Determinação dos coeficientes do modelo após remoção dos <i>outliers</i> .....	54
<b>Tabela 20</b> - Análise de variância para a resposta distância .....	55
<b>Tabela 21</b> - Configuração ótima da otimização realizada na função <i>Rastrigin</i> .....	57
<b>Tabela 22</b> - Análise <i>Two Sample-t</i> para os tempos.....	58
<b>Tabela 23</b> - Análise <i>Two Sample-t</i> para as iterações .....	59
<b>Tabela 24</b> - Comparação dos tempos entre a configuração padrão x método atual .....	59

## LISTA DE ABREVIATURAS E SIGLAS

ABC	<i>Artificial Bee Colony</i>
ACO	<i>Ant Colony Optimization</i>
ANOVA	Método Análise de Variância
BA	<i>Bat Algorithm</i>
BFO	<i>Bacterial Foraging Optimization</i>
DOE	Planejamento de Experimentos
Fitness	Melhor Valor
FSS	<i>Fish School Search</i>
MSR	Metodologia de Superfície de Resposta
OLS	<i>Ordinary Least Square</i>
PSO	<i>Particle Swarm Optimization</i>
S/N	Relação Sinal-Ruído ( <i>Signal-to-Noise Ratio</i> )

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
1.1	Considerações iniciais.....	1
1.2	Justificativa .....	2
1.3	Objetivos.....	4
1.3.1	Objetivo geral .....	4
1.3.2	Objetivos específicos.....	4
1.4	Estrutura do trabalho.....	4
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>5</b>
2.1	Computação Natural .....	5
2.2	Otimização por enxame de partículas.....	6
2.2.1	Definição .....	6
2.2.2	Estrutura do Enxame de Partícula Padrão .....	7
2.2.3	Modificações do algoritmo PSO.....	9
2.3	Planejamento de Experimento .....	11
2.3.1	Arranjo de Misturas .....	15
2.3.2	Arranjo de Misturas com variáveis de processo.....	21
2.3.3	Método Taguchi.....	22
2.4	Desirability.....	23
<b>3</b>	<b>MÉTODO E DESENVOLVIMENTO DA PESQUISA.....</b>	<b>28</b>
3.1	Considerações Iniciais .....	28
3.1.1	Característica do método de pesquisa.....	28
3.1.2	Classificação do método de pesquisa .....	29
3.2	Método proposto para aperfeiçoamento do algoritmo de otimização.....	31
<b>4</b>	<b>ANÁLISE DOS RESULTADOS .....</b>	<b>36</b>
4.1	Considerações Iniciais .....	36
4.2	Aplicação do método na função teste esférica.....	37
4.3	Aplicação do método na função teste Rosembrock .....	45
4.4	Aplicação do método na função teste Rastrigin.....	52
4.5	Considerações Finais .....	60
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>61</b>
5.1	SUGESTÕES PARA TRABALHOS FUTUROS .....	62

<b>REFERÊNCIAS .....</b>	<b>63</b>
<b>ANEXOS A – Código do Algoritmo <i>Particle swarm optimization</i> .....</b>	<b>68</b>
<b>Código do algoritmo .....</b>	<b>70</b>
<b>ANEXOS B – Tempo e Iterações encontrados para as 26 configurações com 50 rodadas da função Esférica .....</b>	<b>75</b>
<b>ANEXOS C – Tempo e Iterações encontrados para as 26 configurações com 50 rodadas da função Rosembrock.....</b>	<b>87</b>
<b>ANEXOS D – Tempo e Iterações encontrados para as 26 configurações com 50 rodadas da função Rastrigin .....</b>	<b>101</b>

# 1 INTRODUÇÃO

## 1.1 Considerações iniciais

De modo geral, a otimização é conceituada como processo encarregado de encontrar melhores condições para as funções, sejam elas de máximo ou mínimos (ALBRECHT, 2005). Rao (2009) enfatiza que o propósito da utilização desta técnica é que o engenheiro seja capaz de maximizar a eficiência de alguns sistemas, minimizar os esforços necessários, ou também encontrar o melhor caminho.

Com o auxílio do método de otimização, pode-se chegar à conclusão que a melhoria na qualidade de determinados sistemas se deve ao direcionamento em que o processo é encaminhado, ou seja, para o ponto onde o resultado traz algum tipo de satisfação ao aplicador, como por exemplo, no desenvolvimento de produtos acabados e de produtos com excelência qualidade. Mas para que isso ocorra, os sistemas devem ser avaliados a fim de encontrar o seu grau de complexidade, onde se possa desenvolver técnicas matemáticas e algoritmos capazes de solucionar de forma rápida e eficiente o problema através da otimização.

Dessa forma, a técnica de otimização se mostra uma ferramenta muito útil a pesquisadores que pretendem ajustar processos e/ou sistemas. Nesta pesquisa, o problema a ser analisado é o algoritmo de otimização, visto que no processo de otimizar um determinado sistema o mesmo passa por metodologias matemáticas e algoritmos, ou seja, a forma como se procede e soluciona os problemas de otimização não se dão apenas nos ajustes matemáticos, mas sim em adequações de calibração onde os algoritmos são capazes de apresentar informações necessárias para a busca do ponto ótimo de uma determinada função.

O algoritmo otimização por enxame de partículas (*Particle Swarm Optimization – PSO*) foi baseado no comportamento dos pássaros devido a características das aves, que se comunicam entre si, com o intuito de encontrar melhores pontos de alimentação e até mesmo de ninhos (KENNEDY e EBERHART, 1995; POLI, R., KENNEDY, J. e BLACKWELL, T. , 2010). Segundo Albrecht (2005), este método tem recebido diversas melhorias e apresentado excelentes resultados. Ainda assim, os autores (BOLTON e PARADA, 2015) acreditam que apesar do algoritmo de otimização enxame de partículas possuir um desempenho formidável, a concepção de um algoritmo eficiente para resolução de problemas complexos de otimização permanece em um ciclo constante de desafios. Neste sentido, o presente trabalho apresentou o objeto de estudo desta dissertação para se obter uma melhor eficiência ao algoritmo PSO,

identificando os fatores que influenciam o desempenho deste método de otimização. Em (EIBEN e SMIT, 2011), os autores destacam que os parâmetros de um algoritmo podem influenciar significativamente a solução final, tornando-se um problema para quem utiliza tal algoritmo de otimização.

Com o intuito de contornar tal deficiência quanto a calibração dos parâmetros que influenciam na performance do algoritmo, diversas pesquisas foram realizadas para identificar e otimizar tais parâmetro. Dentre as técnicas encontradas, destaca-se o trabalho de EIBEN *et al.*, (1999) que apresenta uma taxonomia, isto é, uma maneira de controlar, classificar e organizar os parâmetros de qualquer tipo de algoritmo, vale ressaltar que nesta técnica encontra-se a abordagem de planejamento de experimentos (*DOE*). Segundo Montgomery (2012), a técnica de planejamento de experimentos constitui de estratégias matemáticas e estatísticas com a finalidade de desenvolver arranjos experimentais eficientes, econômicos e balanceados.

Dada a problemática deste tema, esta pesquisa investigará a suposição de que os parâmetros do algoritmo de otimização enxame de partículas influenciam diretamente a resolução de problemas complexos de otimização, visto que a concepção de desenvolver um algoritmo eficiente e eficaz na solução de problemas de otimização tem sido um desafio bastante relevante. Para se alcançar tal objetivo e obter um melhor desempenho ao algoritmo de otimização será aplicado uma combinação de técnicas matemáticas e estatísticas começando por planejamento de experimentos, distância euclidiana, Taguchi, análise de regressão, mínimos quadrados ordinários (OLS) e por fim a otimização dos parâmetros utilizando *desirability*. Através das respostas obtidas, pode-se observar que os parâmetros sugeridos pela literatura não se mostrou tão eficiente quanto a busca pela solução ótima das funções *benkchmark*.

A seção a seguir apresentará uma justificativa de utilizar a otimização como ferramenta de melhoria da qualidade, bem como uma breve pesquisa sobre as formas de tratamento que auxiliam no aperfeiçoamento de algoritmos.

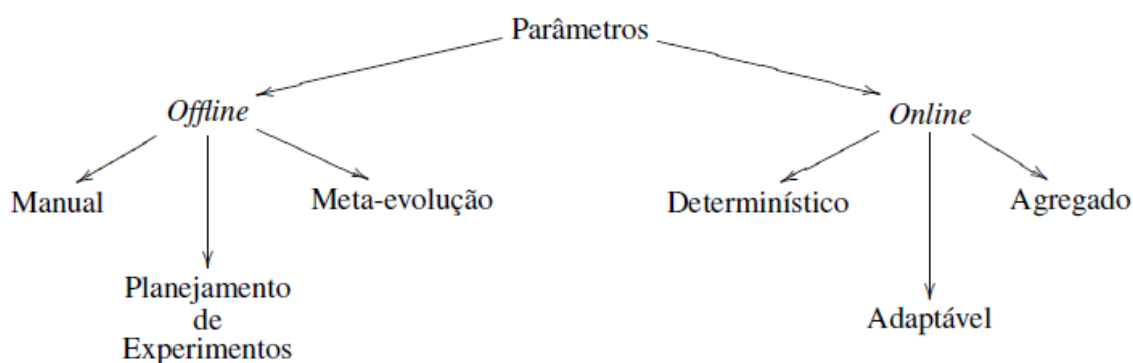
## 1.2 Justificativa

Partindo da premissa que o algoritmo é uma técnica que auxilia na resolução de problemas de otimização, existe um pequeno impasse nestes algoritmos, isto é, um déficit em relação a calibração dos parâmetros, ou seja, como se deve configurar os parâmetros de um algoritmo de otimização (EIBEN e SMIT, 2011; EIBEN *et al.*, 1999). Portanto, a pesquisa se justifica na limitação do ajuste do otimizador, uma vez que, a configuração dos parâmetros influencia diretamente a solução final de um problema de otimização causando uma má

qualidade na resposta. Com isto, o autor Esmin (2005) destaca que sempre haverá a necessidade de se ter melhores algoritmos de otimização para solução de processo.

Para solucionar o problema com a configuração dos parâmetros, Eiben *et al* (1999) sugerem duas formas de ajuste a *off-line* e *online*. Este método de ajuste define um padrão para os parâmetros do algoritmo, ou seja, através desta técnica os algoritmos se sujeitaram a uma abordagem de tentativa e erro até encontrarem o melhor ajuste viável através de ferramentas matemáticas e estatísticas. Vale ressaltar para que tal ajuste seja realizado de forma concisa em um processo não se deve generalizar para todos os problemas de otimização. Nesta etapa os valores para cada parâmetro do algoritmo se modificam de acordo com números de iterações que sofre, ou seja, a cada etapa do algoritmo o mesmo sofre alterações em seus resultados.

A figura 2, apresenta a taxonomia dos ajustes *off-line* e *online* (EIBEN e SMIT, 2011; EIBEN *et al.*, 1999).



**Figura 1** - Taxonomia de ajuste de parâmetro do algoritmo  
**Fonte** - (Eiben; Smit, 2011; Eiben *et al.*, 2007)

Neste trabalho, será atribuída a forma *off-line*, devido à maneira como os parâmetros do algoritmo *particle swarm optimization* (PSO) foram modelados. Os parâmetros escolhidos a serem controlados foram: peso de inércia ( $w$ ), aprendizagem cognitivos ( $c1$ ) e aprendizagem sociais ( $c2$ ). A escolha destes fatores se justificam através do trabalho de (JORDEHI e JASNI, 2013), onde o autores listam alguns parâmetros e as diversas maneiras de ajustes que o algoritmo passou nos últimos anos.

As modificações mais comuns de calibração dos parâmetros do algoritmo de otimização enxame por partícula se encontram nos seguintes trabalhos: *TVAC-PSO Based Optimal Reactive Power Dispatch for Reactive Power Cost Allocation under Deregulated Environment* (ACHAYUTHAKAN e ONGSAKUL, 2009), *Automatic design of algorithms for optimization problems* (BOLTON e PARADA, 2015), *Chaotic Inertia Weight in Particle Swarm Optimization* (FENG *et al.*, 2007), *A Novel Particle Swarm Optimization Based on the Self-Adaptation Strategy of Acceleration Coefficients* (LI e XU, 2009), *An adaptive fuzzy weight*



*PSO algorithm* (LIU *et al.*, 2010), *Adaptive Particle Swarm Optimization using velocity Information of Swarm* (YASUDA e IWASAKI, 2004), *A New Particle Swarm Optimization Algorithm with Random Inertia Weight and Evolution Strategy* (YUE-LIN e YU-HONG, 2007).

## 1.3 Objetivos

### 1.3.1 Objetivo geral

O objetivo geral deste trabalho é realizar a otimização no algoritmo de otimização enxame de partículas (PSO) utilizando a técnica de planejamento de experimentos com a finalidade de melhorar o desempenho deste otimizador.

### 1.3.2 Objetivos específicos

Como objetivo específicos desta dissertação, temos:

- Aplicar um procedimento experimental para a otimização do algoritmo utilizando planejamento de experimentos (DoE);
- Aplicar métodos matemáticos nos resultados obtidos das funções testes;
- Verificar a influência dos parâmetros na performance do algoritmo.

## 1.4 Estrutura do trabalho

Com o intuito de cumprir os objetivos propostos, esta dissertação se divide em 5 capítulos, apresentados na Tabela 1.

**Tabela 1** - Estrutura de desenvolvimento do trabalho

Capítulo	Conteúdo	Atividade
1	Introdução	Apresentar a relevância do projeto, os objetivos e a estruturação dos capítulos do trabalho.
2	Fundamentação Teórica	Revisão documental e bibliográfica.
3	Definição do método de pesquisa	Trata-se do método de pesquisa, que será utilizado como forma de condução do trabalho e a condução da pesquisa de acordo com os critérios estabelecidos pelo método adotado.
4	Aplicação do método proposto	Método de pesquisa junto ao método proposto que será utilizado como forma de condução do trabalho para o aperfeiçoamento do otimizador.
5	Conclusão	Conclusão do autor relativa à pesquisa realizada.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Computação Natural

A computação natural é um termo que vem sendo utilizado por pesquisadores a fim de representar uma grande gama de sistemas computacionais criados através de mecanismo inspirados biologicamente e/ou natural (BALLARD, 2000; DE CASTRO, 2007; DE CASTRO, 2006; GELENBE, 2012). Nas últimas décadas, diversos estudos vêm sendo realizados nesta área, proporcionando um amplo crescimento de novas abordagens da computação natural.

Estas abordagens se caracterizam dentro de três subáreas da computação natural (DE CASTRO, 2007):

- **Computação inspirada na natureza:** Nesta categoria, foram desenvolvidas estratégias computacionais que utilizam características inspiradas pela natureza. Estas ferramentas computacionais solucionam problemas em diversos campos, tais como: redes neurais artificiais (HAYKIN S., 1999), computação evolutiva (BACK *et al* 1997), inteligência de enxame (CHEW *et al*, 2013; DORIGO *et al.*, 2010; NEDJAH e MOURELLE, 2006) e os sistemas imunológicos artificiais ( SONODA *et al.*, 2018; DE CASTRO *et al*, 2007).
- **Estudos sobre a natureza através da computação:** Esta área envolve métodos computacionais capazes de reproduzir (imitar) um comportamento natural (mimetismo) e até mesmo padrões e processos biológicos da maneira que são identificados na natureza. As categorias de pesquisa para esta área são os estudos sobre vida e organismos artificiais (ADAMI, 1999; LANGTON, 1997).
- **Computação com mecanismo natural:** As estratégias desta área se concentram em um novo paradigma da computação, onde utilizam de elementos retirados da natureza como cadeias de DNA ou bits quânticos, a fim de desenvolver computadores com estruturas naturais trocando o silício por outro material natural, tornando-o um produto candidato a substituir os computadores digitais. Esta área pode ser representada pelos seguintes campos de estudos: computação molecular (DE SILVA, 2013; GHEORGHE, 2005) e a computação quântica (PAVICIC, 2005).

Estas subáreas têm-se mostrado eficientes, pois contribuem com poderosas ferramentas computacionais capazes de solucionar problemas complexos de diversas áreas do

conhecimento, além de proporcionar oportunidades de novas concepções para a computação natural (PATON, 1994; PATON *et al.*, 2004).

A seção a seguir descreve a característica do algoritmo de otimização por enxame de partículas, que se encontra no campo da computação inspirada pela natureza, que por sua vez, está dentro da área do conhecimento inteligência por enxame.

## 2.2 Otimização por enxame de partículas

### 2.2.1 Definição

A otimização por enxame de partículas (PSO) foi apresentada em 1995 por Kennedy e Eberhart (1995), como uma metaheurística estocástica que simula a interação dos pássaros e cardumes de peixes em busca de “alimentos” e/ou “ninhos”. Esta técnica foi inspirada pela natureza, que por sua vez, surgiu de uma das grandes áreas da computação natural, conhecida como “Inteligência de Enxame” (DE CASTRO, 2007).

Esta abordagem vem sendo utilizada como ferramenta computacional a fim de solucionar problemas de otimização de diversas áreas do conhecimento, sendo elas: computação, engenharias, entre outras. O autor Poli (2007), destaca em seu trabalho o uso do algoritmo de otimização *PSO* em determinadas categorias, sendo elas: Antenas (PANTOJA *et al.*, 2007), biomedicina (QIU *et al.*, 2014), ajuste de controle (CHEN *et al.*, 2018a), redes de distribuição (MOSBAH *et al.*, 2016) entre outros.

Segundo Parpinelli e Lopes (2011), diversos algoritmos vêm surgindo através desta abordagem computacional inspirada pela natureza, tais como:

- os algoritmos de otimização por enxame de formiga (*Ant colony optimization - ACO*) (DORIGO e DI CARO, 1999);
- algoritmos de cardume artificial (*Fish School Search - FSS*) (BASTOS FILHO *et al.*, 2008; NESHAT *et al.*, 2014);
- algoritmo por colônia de abelhas artificiais (*Artificial bee colony - ABC*) (AKAY; KARABOGA, 2012);
- algoritmos de otimização por colônia de bactéria (*Bacterial Foraging Optimization – BFO*)(PASSINO, 2002; TEODORO, PARPINELLI e LOPES, 2010);
- algoritmo de otimização por enxame de morcego (*Bat algorithm – BA*) (YANG, 2010); e o que abordaremos neste trabalho:
- o algoritmo de otimização por enxame de partículas (KENNEDY e EBERHART, 1995).

Para Millonas (1994), os algoritmos de otimização por inteligência de enxame apresentam as seguintes características: princípios da proximidade, princípio da qualidade, princípio da reposta diversa, princípio da estabilidade, princípio da adaptabilidade. A existência destas características proporciona um importante equilíbrio e estabilidade ao método. Além destas características, os algoritmos de inteligência de enxame possuem uma abordagem populacional, onde utilizam uma população de indivíduos que pode ser denominada enxame (POLI, KENNEDY e BLACKWELL, 2007).

De acordo com Millonas (1994), o comportamento desta população se conduz ao ponto de solução para o problema em questão, devido a sua interação com os demais agentes (partículas) no espaço de busca. Em uma abordagem simples, o enxame se movimenta conforme a interação das partículas da população e a troca de informação em relação ao espaço de busca (ENGELBRECHT, 2007).

Na próxima seção, será descrita a estrutura do algoritmo de otimização enxame por partícula.

### 2.2.2 Estrutura do Enxame de Partícula Padrão

Como descrito na seção anterior, o algoritmo de PSO, utiliza de uma população de indivíduos que se movimentam no espaço de busca para encontrar a melhor solução para o problema em questão.

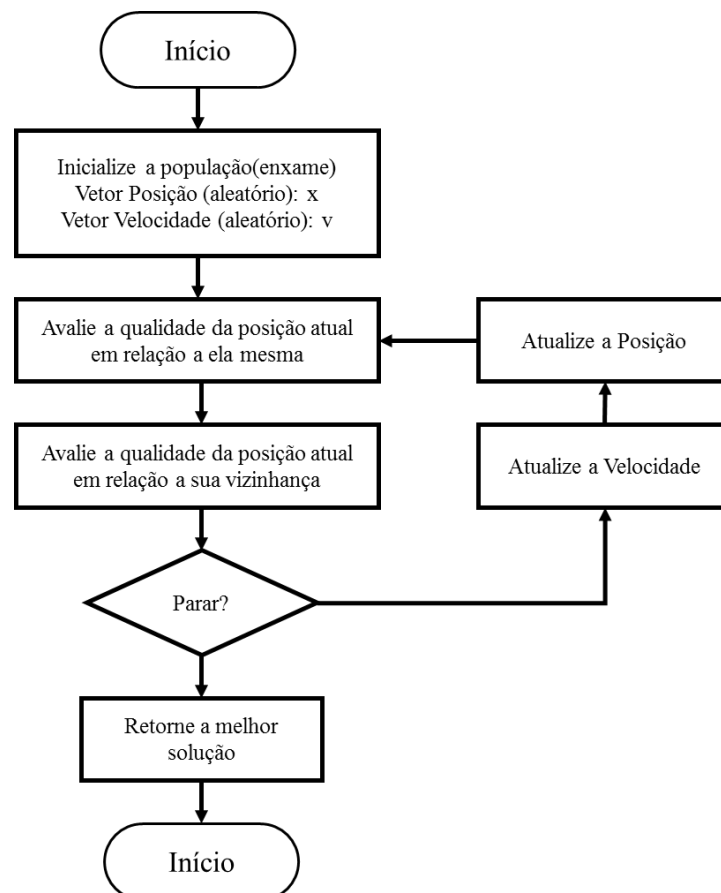
Para o movimento do PSO, diversos aspectos devem ser analisados para um bom funcionamento do método. Uma das direções a ser seguidas, se deve aos valores a serem encontrados pela interação das partículas, como:

- (i) Possuir um melhor valor para partícula individual conhecida como aprendizagem cognitiva;
- (ii) Possuir um melhor valor para partícula global do enxame, denominada como aprendizagem social;
- (iii) Possuir uma velocidade aleatória para todos os indivíduos do enxame.

Segundo De Castro (2007), as posições e velocidades das partículas se modificam a cada passo iterativo entre os indivíduos do enxame, tornando possível o movimento das partículas. Para Jiao, Lian e Gu (2008), a partícula  $i$  em uma iteração  $k$  se movimenta em um espaço de busca, devido a dois fatores: a velocidade e a posição que são representadas como  $\vec{V}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$  e  $\vec{X}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , respectivamente. Em seguida, a posição de armazenamento da melhor partícula individual visitada é  $\vec{P}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$  e denominada

como  $p_{best}$ . Já a melhor partícula global se destaca como  $\vec{P}_g = (p_{g1}, p_{g2}, \dots, p_{gD})$  denominada como  $g_{best}$ , conhecida também como a melhor partícula do enxame.

Ainda Jiao, Lian e Gu (2008), durante a execução do algoritmo a avaliação da qualidade das respostas acontece a cada iteratividade entre as partículas, atualizando suas melhores partículas individuais ( $p_{best}$ ) e a melhor do enxame ( $g_{best}$ ). Logo em seguida, as partículas se destinam a suas novas posições. A figura 3 ilustra a maneira com que as partículas se movimentam para encontrar suas melhores soluções durante sua execução.



**Figura 2** - Fluxograma do algoritmo de PSO

**Fonte** - Nascimento *et al.*, (2012)

Como demonstrado na figura 3, as partículas  $i$  fazem suas atualizações nas  $k$  iterações, onde cada movimento destes agentes são calculados através de suas velocidades e posições, como apresenta as equações a seguir (KENNEDY e EBERHART, 1995):

$$V_{id+k} = V_{id} + c1 \times \text{rand}() \times (P_{id} - X_{id}) + c2 \times \text{rand}() \times (P_{gd} - X_{id}) \quad (2.1)$$

$$X_{id+k} = V_{id+k} + X_{id} \quad (2.2)$$

Onde  $V_{id+k}$  representa a velocidade de cada partícula  $i$  na sua iteração  $k$ ,  $V_{id}$  é a velocidade atual da partícula naquele instante. As variáveis  $c1$  e  $c2$  são duas constantes positivas

conhecidas como parâmetros de confiança e possui valores iguais a 2, que influenciam os componentes cognitivos e sociais respectivamente (KENNEDY e EBERHART, 1995). A variável conhecida como  $rand()$  utiliza uma função aleatória distribuída uniformemente entre 0 e 1.  $P_{id}$  é a melhor posição individual visitada pela partícula  $i$  e, por fim,  $P_{gd}$  é a melhor posição global visitada pela partícula  $i$ .

A seção a seguir, descreve as modificações que o algoritmo de otimização enxame de partículas (PSO) passou durante anos de estudos.

### 2.2.3 Modificações do algoritmo PSO

Como descrito na literatura, o algoritmo de otimização por enxame de partículas surgiu com intuito de solucionar problemas de otimização de diferentes áreas do conhecimento. Por outro lado, modificações devem ser realizadas nos algoritmo de otimização para que apresente melhores resultados na resolução de problemas complexos de otimização (CORNE *et al*, 1999).

Inicialmente, Shi e Eberhart (1998) apresentaram um novo parâmetro para uma obtenção de uma nova concepção do algoritmo PSO. Este parâmetro se denomina peso da inércia ( $w$ ), com esta modificação o algoritmo melhoraria sua velocidade de convergência, equilibrando também a velocidade de cada partícula do enxame.

A partir desta nova abordagem, pode-se ter a seguinte equação para o algoritmo principal:

$$V_{id+k} = w \times V_{id} + c1 \times rand() \times (P_{id} - X_{id}) + c2 \times rand() \times (P_{gd} - X_{id}) \quad (2.3)$$

$$X_{id+k} = V_{id+k} + X_{id} \quad (2.4)$$

Ainda Shi e Eberhart (1998) afirmam que se utilizarem os valores de  $w$  próximos a 1.0, estariam encorajando os agentes (partículas) a explorarem mais o espaço de busca, ou seja, explorarem a amplitude da região. Por outro lado, se utilizarem valores abaixo de 1.0, incentivaria as partículas a explorarem um espaço de profundidade, ou seja, fazendo com que os agentes se esforcem a encontrar uma área melhor. O uso deste novo parâmetro, proporcionou uma melhora no desempenho do algoritmo, além de desaparecer com o mecanismos de controle da velocidade das partículas.

Diversas melhorias para o algoritmo de otimização por enxame de partículas vem sendo aplicadas, além da modificação feita pelos autores (SHI e EBERHART, 1998). Outra contribuição que modificou significativamente o algoritmo foi implementada por Clerc (1999), que teve o intuito de avaliar as características de convergência e estabilidade. Após diversas análises realizadas pelo autor, conseguiu-se confirmar que ao utilizar este novo parâmetro o

enxame do algoritmo pôde chegar a uma convergência mais instantânea em uma região de ótimo local logo após alguns números de iterações. Além desta confirmação, outra comprovação pode ser consolidada, o ajuste da velocidade dos agentes (partículas) do enxame acontece de forma equilibrada, ou seja, a velocidade de locomoção dos indivíduos do enxame ocorre de maneira estável.

O fator de constrição proposto por Clerc (1999), apresenta uma estrutura de execução semelhante ao algoritmo original e ao algoritmo proposto por (SHI e EBERHART, 1998), com uma pequena alteração como demonstrado a equação a seguir:

$$V_{id+k} = \chi * \{V_{id} + c1 * rand() * (P_{id} - X_{id}) + c2 * rand() * (P_{gd} - X_{id})\} \quad (2.5)$$

$$X_{id+k} = V_{id+k} + X_{id} \quad (2.6)$$

$$\varphi = c1 + c2 \quad (2.7)$$

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (2.8)$$

Onde o fator de constrição ( $\chi$ ) é calculado pelas equações 2.7 e 2.8, respectivamente. A nova velocidade de atualização das partículas é dada pela equação (2.5) com o fator adicionado. E a equação (2.6) é posição que a partícula vai permanecer no instante  $k$ .

Para Clerc e Kennedy (2002) e Clerc (1999), só se tem convergência em um ponto quando  $\varphi > 4$ . Por outro lado, se  $\varphi$  for menor ou igual a 4, as partículas se movimentaram de maneira vagarosa, fazendo com que não haja possibilidade de convergência no ponto, seja ele ponto de mínimo ou ponto de máximo.

Ainda segundo Clerc (1999), os valores das variáveis  $c1$  e  $c2$  devem ser semelhantes e iguais a 2,05, enquanto  $\varphi$  sendo igual ou maior que 4,1 e  $\chi$ , o fator de constrição, deverá ser 0,7298.

Além destas duas novas concepções, existem outras maneiras capazes de melhorar a performance deste algoritmo de otimização. A tabela 2, mostra uma seleção de trabalhos realizados que propuseram outras modificações para este otimizador de problemas complexos.

Na tabela 2, os autores apresentam diversas técnicas de alteração para o PSO, destacando o trabalho de (SHI e EBERHART, 1999), que por sua vez, trouxe uma forma de ajustar os

parâmetros do algoritmo de otimização conforme seu método. Os autores aplicaram uma formulação para o parâmetro peso da inércia ( $w$ ) de tal maneira que o algoritmo decrementasse linearmente até seu limite final.

Com base nesta técnica, esse trabalho utilizou os valores destacados pelos autores acima citados para realizar o processo de modificação do algoritmo PSO, conforme o direcionamento deste trabalho.

**Quadro 1** - Modificações realizadas ao algoritmo do *PSO*

<b>Ajustes dos Parâmetros do PSO</b>		
<b>Tipo de Ajuste</b>	<b>Parâmetros</b>	<b>Referência</b>
Adaptação do parâmetro utilizando função linear.	$w$	Nikam (2010).
Ajuste do parâmetro utilizando um método de decremento linear.	$w$	Y. Shi e Eberhart (1999).
Adaptação do peso de inércia através de uma função não linear utilizando um método determinístico.	$w$	Jiao, Lian e Gu (2008).
Ajuste do parâmetro peso de inércia de forma constante.	$w=\text{constante}$ .	Y. Shi e Eberhart (1998).
Ajuste dos parâmetros $w$ utilizando o decremento linear em função ao número de iteração, $c1$ com ajuste através das informações de $P_{\text{best}}$ e $G_{\text{best}}$ , $c2$ incrementando através de uma função linear com base no número de iterações.	$w, c1, c2$ .	J. Wang (2011).
Ajuste através de autômatos de aprendizagem para os parâmetros.	$w, c1, c2$ .	Hashemi (2011).
Ajuste através de função linear.	$w, c1, c2$ .	K. Yasuda, K. Yazawa, e M. Motoki (2010).

**Fonte** - Próprio Autor

Na próxima seção, será descrito em detalhes a abordagem utilizada para o ajuste dos parâmetros do algoritmo.

## 2.3 Planejamento de Experimento

Nas últimas décadas, diversos pesquisadores vêm utilizando prática de experimentos em busca de informações detalhadas sobre determinados processos. O autor Ronald A. Fisher foi o pioneiro no desenvolvimento da metodologia de Projeto e Análise de Experimento, por volta de 1920 e 1930, em seguida, a técnica foi aprimorada por outros autores importantes como Hunter, Box e Taguchi, entre outros, a fim de encontrar informações importantes sobre processos.

A definição de um experimento projetado é um teste ou séries de testes que são usados para analisar um determinado processo. Esta abordagem tem intuito de investigar os fatores de entrada de um processo realizando alterações propositalmente a fim de observar mudanças nas variáveis de saída (MONTGOMERY, 2004, 2012).

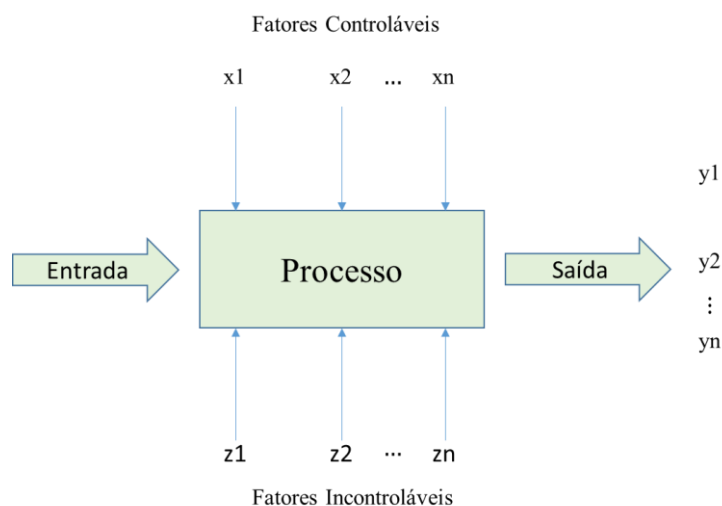


Segundo Silva e Silva (2008), o planejamento de experimento (DOE – Design of Experiments) é uma técnica utilizada para delinear fatores de entrada de um processo. Esta técnica auxilia na redução de perdas de materiais, aumento no nível da qualidade de um produto e a melhora da produção. Para Montgomery (2012), o planejamento de experimento se faz eficiente, pois avalia a influência das variáveis de entrada de um processo, além destas, observa também as mudanças que ocorre nas variáveis-respostas.

Já MAYER *et al.* (2001), o planejamento de experimento utiliza uma abordagem combinatória entre a matemática e a estatística promovendo arranjos experimentais eficazes, balanceados e econômicos. Além dos benefícios citados, os autores Konda *et al.* (1999) ressaltam que a técnica vêm sendo empregada eficazmente nos setores indústrias, e em diversas áreas do conhecimento, a fim de solucionar problemas complexos de sistemas, contribuindo na melhoria contínua de processo.

De forma matemática, a metodologia empregada gera equações não lineares através da combinação de técnicas estatística, tais como: ANOVA, Testes 2-sample t e análise de regressão. Com estes métodos trabalhado de maneira conjunta a otimização, pode-se chegar a funções objetivas e restrições para diversos processos (MONTGOMERY, 2004, 2012).

A figura 4 a seguir, demonstra uma estrutura de um processo generalizado:



**Figura 3** - Estrutura geral de um processo  
**Fonte** - MONTGOMERY (2009)

Conforme ilustrado na figura 4, o processo é a combinação de fatores controláveis e incontroláveis. Este processo está ligado à união de máquinas, ferramentas, operações, métodos, pessoas e outros recursos que transformam as variáveis de entradas em saídas, podendo ser uma ou mais repostas a serem analisadas (MONTGOMERY, 2004, 2012). Os fatores deste processo, normalmente podem ser controláveis fazendo com que o experimentador utilize de ferramentas capazes de adequar o processo. Em outros casos, o processo poderá ter

ruídos, ou seja, onde haverá fatores incontrolláveis em meio ao processo, ao qual o experimentador terá que tratá-los através de métodos que não comprometam o processo.

Segundo o mesmo autor, muitas vezes os fatores influenciam nas respostas de saídas dos experimentos, por isso, se faz necessário incluir os seguintes objetivos para a experiência:

- Determinar quais as variáveis possuem maior domínio nas respostas  $y$ ;
- Determinar valores aos fatores controláveis ( $x$ 's) que influenciam em  $y$  de forma a deixar valores próximos ao seu valor nominal.
- Determinar valores aos fatores controláveis ( $x$ 's) que influenciam na variabilidade de  $y$ .
- Determinar valores aos fatores controláveis ( $x$ 's) que influenciam os efeitos das variáveis não controláveis  $Z_1, Z_2, \dots, Z_n$  sejam minimizados.

Para Gomes (2010), a técnica de Projeto e Análise de experimento vem sendo empregada eficazmente nos setores industriais, e em diversas áreas do conhecimento, contribuindo para o desenvolvimento de produto e processo a partir da combinação destas ferramentas.

Os principais benefícios alcançados pela técnica de projeto e análise de experimento, segundo MONTGOMERY (2009), são:

- Melhoria do rendimento dos processos;
- Redução de variabilidade e maior conformidade com especificações nominais;
- Redução do tempo de desenvolvimento de produtos ou processos;
- Redução de custos.

Ainda segundo MONTGOMERY (2009), o Projeto e Análise de Experimento divide-se em três princípios básicos que são: aleatorização, replicação e a blocagem. O primeiro princípio se baseia na execução aleatória dos experimentos, a fim de, detectar algum efeito desconhecido sobre os fatores, validando a investigação. Já o segundo princípio, consiste na repetição do experimento, ou seja, diversas vezes o experimento será rodado, fazendo com que gere uma variação para a variável de resposta, utilizando a mesma para uma avaliação de erro experimental. E por fim, o último princípio se baseia na homogeneidade de um experimento que é a condição do mesmo. Esta técnica permite avaliar se há falta de homogeneidade no experimento.

Desta maneira, o experimento deve ser fundamentado de duas formas, a abordagem do projeto de experimento e a análise estatística dos dados. Vale ressaltar, que o emprego destas duas abordagens se fazem necessárias quanto ao entendimento sobre os objetos de estudo, a

coleta de dados e do entendimento básico referente às ferramentas de análises. Desse modo, Montgomery (2004) divide a execução do planejamento e análise experimento em sete etapas, são elas:

1. Definição do problema;
2. Escolha dos fatores e definição dos níveis de trabalho;
3. Seleção das variáveis de resposta;
4. Escolha do projeto experimental;
5. Execução dos experimentos;
6. Análise estatística dos dados;
7. Conclusões e recomendações.

Após a definição do problema a ser estudado, a escolha dos fatores a serem tratados e a seleção das variáveis respostas que são influenciadas pelos fatores, a etapa importante a ser seguida é a escolha do projeto experimental, também conhecida como arranjo experimental.

Com relação aos arranjos experimentais, compreende na literatura que as técnicas mais utilizadas são as Planejamento Fatorial Completo, o Planejamento Fatorial Fracionado, os arranjos de Taguchi, Metodologia de Superfície de Resposta e os Arranjos de Misturas.

O Quadro 2 mostra as principais características de alguns tipos de arranjos experimentais mais comuns, bem como suas vantagens, desvantagens e aplicações em cada arranjo.

**Quadro 2 (a) - Principais técnicas de Arranjos Experimentais e suas características**

<b>Arranjo Experimental</b>	<b>Vantagens</b>	<b>Desvantagens</b>	<b>Aplicações</b>
<b>Fatorial Completo (<math>2^k</math>)</b>	Permite varredura completa da região de estudo, pois utiliza todos os fatores e respectivos níveis.	Não identifica variação intermediária, pois só trabalha em dois níveis. Vale ressaltar, que demanda um nº elevado de corridas para problemas com muita variável.	A aplicação deste arranjo se dá em processos onde já se tem um prévio domínio e onde os testes não demandam custo ou tempo altíssimos.
<b>Fatorial Fracionado (<math>2^{k-p}</math>)</b>	Permite uma pré-análise do processo com um nº moderado de rodadas.	Não realiza uma varredura completa na região experimental.	A aplicação deste arranjo se dá em processos onde se deseja um pré-conhecimento e onde a literatura é limitada, além de testes que demandam custo ou tempo altíssimos.

**Quadro 2 (b)** - Principais técnicas de Arranjos Experimentais e suas características

<b>Arranjo Experimental</b>	<b>Vantagens</b>	<b>Desvantagens</b>	<b>Aplicações</b>
<b>Taguchi</b>	Permite a análise de um processo com muitas variáveis de entrada com um número extremamente reduzido de experimentos.	Apresenta uma ideia do processo, mas proporciona modelos matemáticos não confiáveis.	A aplicação deste arranjo se dá em processos onde há pouco ou nenhum conhecimento prévio de comportamento para processos com dispersões altas ou testes que demandam custo ou tempo altíssimos.
<b>Metodologia de Superfície de Resposta (MSR)</b>	Permite a verificação de variações intermediárias do processo.	Por realizar poucas corridas nos níveis +2 e - 2, podem proporcionar erros na extrapolação dos níveis.	A aplicação deste arranjo se dá em processos que são bem conhecidos e com baixa dispersão.

**Fonte** - Gomes (2010)

Para a criação destes arranjos experimentais, diversos softwares foram desenvolvidos por pesquisadores a fim de auxiliar em uma análise adequada das informações por parte do experimentador. O experimentador poderá construir seu próprio arranjo experimental definindo as informações corretas que são: os níveis do arranjo experimental, números de fatores e o tipo de arranjo experimental. A partir disto, o software irá gerar planilhas eletrônicas com a ordem dos experimentos guiando o experimentador de forma correta.

Como visto no quadro 2, foram apresentadas as características presentes em cada um dos arranjos experimentais do planejamento e análise de experimento, bem como suas vantagens e desvantagens de utilização. Vale ressaltar que o arranjo experimental de misturas, não se esboçou no quadro, pois tais conceitos serão apresentados na próxima seção.

### 2.3.1 Arranjo de Misturas

Segundo Montgomery (2009), os arranjos de misturas são definidos como classe especial de experimento do tipo superfície de respostas; um experimento de misturas possui fatores ou ingredientes de uma mistura, e consequentemente, os níveis não são independentes. Em adição, o autor explica que no mais simples experimento de misturas, a resposta do processo, se dá em forma de proporções dos fatores que o compõem. Cornell (2011) exemplifica um arranjo de misturas, como uma medida para realização de um ponche de frutas, onde os

valores para a execução do mesmo não são independentes, ou seja, quando os valores de  $x_1, x_2, \dots, x_{q-1}$  são conhecidos, o valor de  $x_q$  é automaticamente determinado. Por exemplo, um ponche de frutas que possui três variáveis, sendo uma delas a melancia representa 60% do sabor do ponche, e portanto, seu valor para a mistura será igual a  $x_1 = 0,60$ . Outra fruta (variável) é o abacaxi, este sabor representará 25%, ou seja,  $x_2 = 0,25$ . E por fim, a última fruta (variável) que sempre será a diferença entre a soma das demais frutas representará 15% do sabor da mistura, portanto,  $x_3 = 0,15$ . Com isso temos que um ponche de frutas com 100% dos sabores misturados de acordo com as equações (2.9) e (2.10).

Com relação a esta característica presente no arranjo de misturas, (MONTGOMERY, 2012) indaga que os experimentos de misturas devem ser planejados e conduzidos através de arranjos específicos. Ainda o mesmo autor, descreve que os experimentos abordados pelos arranjos de misturas podem ser representados graficamente pela figura 5, onde as figuras (a) e (b) ilustram a região de um arranjo de misturas para 2 componentes, e respectivamente, representa um arranjo de mistura com 3 componentes, vale destacar que a região conhecida nesta restrição é denominada *simplex*.

O autor Cornell (2011), ainda destaca que a estrutura de restrição deste arranjo se encontra na seguinte equação:

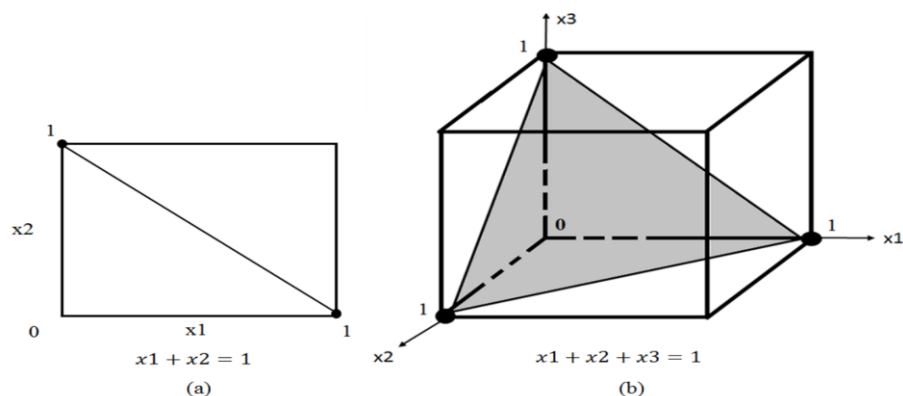
$$0 \leq x_i \leq 1, \quad i = 1, 2, \dots, q \quad (2.9)$$

e

$$\sum_{i=1}^q x_i = x_1 + x_2 + \dots + x_q = 1.0 \quad (2.10)$$

Onde:

- $q$  representa os fatores ou ingredientes da mistura.
- $x$  representa a proporção de cada fator da mistura.

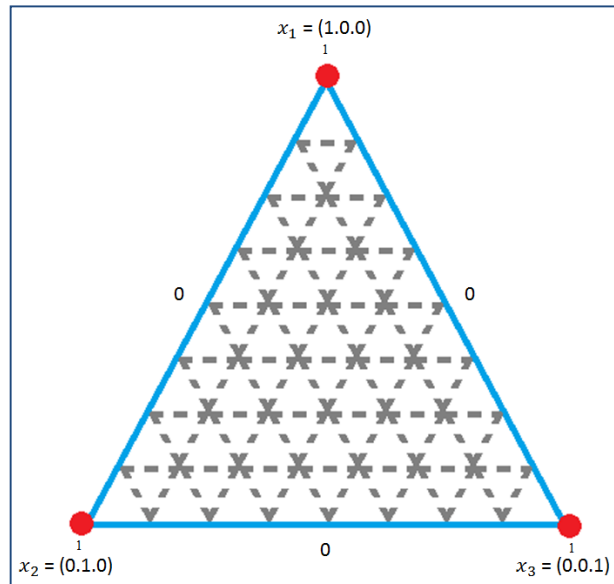


**Figura 4** - Restrição da região do espaço de um arranjo de misturas.

**Fonte** - CORNELL (2011)

A figura 4 (a) mostra que os valores dos componentes não podem ser negativos, e se expressam como frações da misturas, tornando uma única unidade, ou seja, 100%. É possível perceber que se a região denominada como *simplex*, possui  $q$  igual a 2, tem-se então uma reta, por outro lado, se  $q$  for igual a 3 considera-se um triângulo equilátero, mas se  $q$  for igual a 4 haverá uma figura representativa de um tetraedro, e por fim, se  $q$  for maior que 4, torna-se impossível representar a figura no espaço tridimensional (CORNELL, 2011; REIS e ANDRADE, 1996).

Com isso, é possível perceber que a figura 4 (b), ilustra um triângulo de coordenadas que permite a relação entre componentes de uma mistura composta por três elementos. Em uma mistura, os elementos da mesma se limitam e/ou se restringem conforme a soma de seus componentes, tornando-os então uma só unidade. Através deste princípio, torna-se possível a visualização dos valores máximos e mínimos dos três elementos que compõem as misturas. Partindo deste princípio, a figura 5 apresenta o arranjo geral deste sistema triangular de coordenadas, que por sua vez, ilustra a forma como serão distribuídas as proporções em relação ao total, sendo ela igual a 100%. Vale destacar que cada vértice do triângulo representa uma mistura completa, ou seja, uma “Mistura Pura”.



**Figura 5** - Arranjo de Mistura Pura

Fonte - CORNELL (2011)

Estes arranjos de misturas podem seguir três abordagens para otimização de processo, tais como: *simplex-lattice*, *simplex-centroid* e *extreme vértice*. Cada um destes arranjos, possuem propriedades peculiares para determinados tipos de problemas de otimização. Para a pesquisa em questão, adota-se o arranjo *simplex-lattice*.

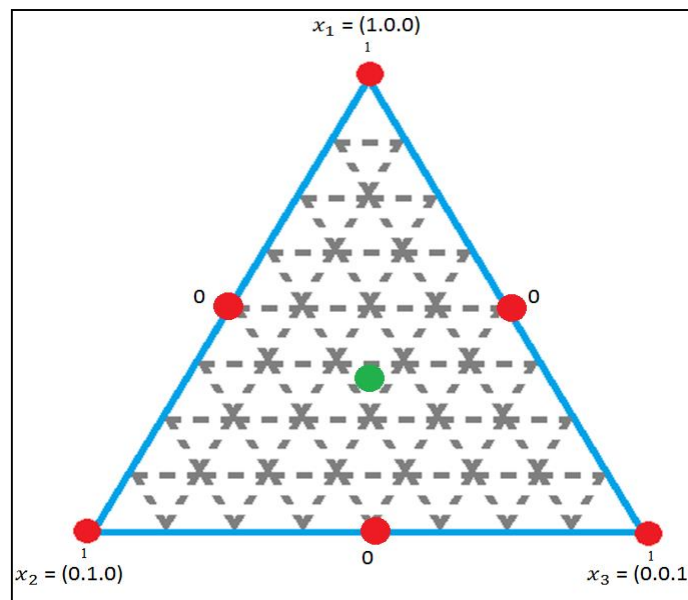
Este método, mostra que as de  $q$  variáveis de entrada são ajustadas em  $2^q - 1$  pontos, construídos pelos seguintes critérios:

- $q$  permutações de  $(1,0,0, \dots, 0)$ ;
- $\binom{q}{2}$  permutações de  $(\frac{1}{2}, \frac{1}{2}, 0, \dots, 0)$ ;
- $\binom{q}{3}$  permutações de  $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, \dots, 0)$ ;
- e assim por diante até alcançar o centroide  $(\frac{1}{k}, \frac{1}{k}, \dots, \frac{1}{k})$ .

Com isso, vale ressaltar que estes arranjos *simplex*, possuem possíveis combinações de misturas e suas proporções são definidas através das equações de restrição (2.9) e (2.10), respectivamente. Para se gerar um número total de experimentos ( $N$ ), deve-se observar a quantidade de grau *lattice* ( $m$ ) que será utilizado, portanto a Equação (2.11) a seguir ilustra o número de experimentos que serão determinados.

$$N = \frac{(q + m - 1)!}{m! \cdot (q - 1)!} \quad (2.11)$$

Com base no conceito já apresentado sobre o arranjo de misturas *simplex centroid*, a figura 6 ilustra um modelo construído com três variáveis de entrada e considerando um grau *lattice* 2, respeitando os limites de restrição do arranjo. A figura 6 ilustra a configuração dos pontos do arranjo, de acordo com o número do grau do polinômios definido.



**Figura 6** - Representação de um arranjo de misturas *simplex centroid*  
**Fonte** - Gomes, (2013), Adaptado de Cornell, (2011)

Gomes (2013) ainda destaca que os modelos matemáticos usados para a representação das respostas neste modelo de misturas apresentam algumas diferenças em relação aos modelos do RMS (Metodologia de Superfície de Respostas), devido à relação dos polinômios usados

para ambas metodologias. Além disto, a base de modelamento dos experimentos de misturas são no entanto, a grande diferença se dá pela existência da seguinte restrição  $\sum_{i=1}^k x_i = 1$ . Sendo assim, os modelos de misturas dependem do comportamentos das respostas analisadas, para se tornar um modelo linear, quadrático, cúbico (CORNELL, 2011).

As equações a seguir, representam as formas canônicas para cada modelo de misturas:

1. Modo Linear:

$$y(x) = \sum_{i=1}^n \beta_i x_i \quad (2.12)$$

2. Modelo Quadrático:

$$y(x) = \sum \beta_i x_i + \sum \sum_{i < j} x_i x_j \quad (2.13)$$

3. Modelo Cúbico Especial:

$$y(x) = \sum \beta_i x_i + \sum \sum_{i < j} x_i x_j + \sum \sum \sum_{i < j < k} x_i x_j x_k \quad (2.14)$$

4. Modelo Cúbico Completo:

$$\begin{aligned} y(x) = & \sum \beta_i x_i + \sum \sum_{i < j} x_i x_j \\ & + \sum \sum_{i < j} \delta_{ij} x_i x_j (x_i - x_j) + \sum \sum \sum_{i < j < k} x_i x_j x_k \end{aligned} \quad (2.15)$$

5. Modelo Quártico Especial:

$$\begin{aligned} y(x) = & \sum \beta_i x_i + \sum \sum_{i < j} x_i x_j + \sum \sum \sum_{i < j < k} x_i^2 x_j x_k + \sum \sum \sum_{i < j < k} x_i x_j^2 x_k \\ & + \sum \sum \sum_{i < j < k} x_i x_j x_k^2 + \sum \sum \sum_{i < j < k} x_i x_j x_k x_l \end{aligned} \quad (2.16)$$

6. Modelo Quártico Completo:



$$\begin{aligned}
y(x) = & \sum \beta_i x_i + \sum \sum_{i < j} x_i x_j + \sum \sum_{i < j} \gamma_{ij} x_i x_j (x_i - x_j) \\
& + \sum \sum_{i < j} \delta_{ij} x_i x_j (x_i - x_j)^2 + \sum \sum \sum_{i < j < k} x_i^2 x_j x_k \\
& + \sum \sum \sum_{i < j < k} x_i x_j^2 x_k + \sum \sum \sum_{i < j < k} x_i x_j x_k^2 \\
& + \sum \sum \sum_{i < j < k} x_i x_j x_k x_l
\end{aligned} \tag{2.17}$$

Segundo Gomes (2013), para se estimar os coeficientes dos modelos, pode-se utilizar o métodos dos mínimos quadrados ordinários (*Ordinary Least Square – OLS*). Através deste método, torna-se possível a construção de uma função aproximada que associa as respostas de interesses com a variável do processo. Para avaliar um modelo construído pelo método *OLS*, deve-se analisar a significância apresentada pelo mesmo. As formas de analisar o modelo, são:

- **Teste de Hipóteses para significância de regressão:** Esta análise pode ser feita através de teste estatístico como, ANOVA (Análise de Variância) que permite investigar o ajuste do modelo, além de verificar quais os termos do modelo são os mais significativos. Além disto, determina a relação linear existente entre a respostas e qualquer variável da regressão.
- **Teste da falta de ajuste:** Neste teste, se houver a existência de observações repetidas, o mesmo pode ser utilizado para estimar o erro aleatório. A partir desta estimativa, pode-se inferir se o modelo representa satisfatoriamente as observações, ou se há possibilidade de acrescentar mais termos ao modelo.
- **Estimativas  $R^2$  e  $R^2(adj)$ :** Esta análise verifica o coeficiente de determinação ( $R^2$ ) do modelo, onde o mesmo representa o percentual de variação na resposta explicada pelo modelo matemático. Em conjunto ao  $R^2$ , tem-se o  $R^2(adj)$  que auxilia na avaliação do coeficiente de determinação do modelo, devido à importância significativa no “superajuste” do mesmo. Este “superajuste” ocorre quando não se adiciona um termo ao  $R^2$ , podendo então tornar o modelo não significativo. O autor Montgomery (2012) afirma que um valor alto para  $R^2(adj)$  torna-se preferível.
- **Teste de normalidade dos Resíduos:** O resíduo pode ser determinado a partir da seguinte modelo de regressão  $r_i = y_i - \hat{y}_i$ , onde o mesmo representa supostamente o erro experimental, segundo o modelo ajustado. Com isso, o resíduo torna-se capaz de avaliar a normalidade do modelo, podendo detectar ponto fora da tendência, ou seja, *outliers*.

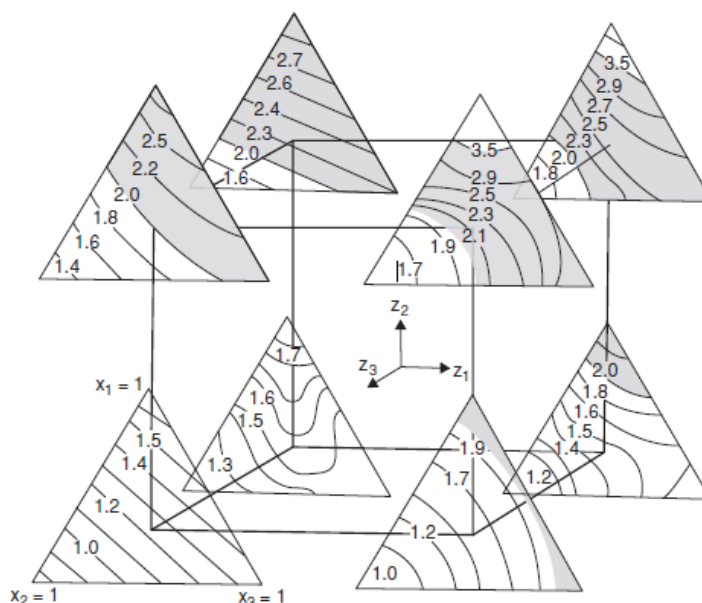
### 2.3.2 Arranjo de Misturas com variáveis de processo

Como visto, um arranjo de misturas é um projeto experimental modificado que permite encontrar a formulação ideal para as misturas. Ao longo do processo, o mesmo poderá ser incrementado com uma variável de processo, onde é denominado um arranjo cruzado. Este arranjo cruzado é uma técnica muito conhecida em *DOE*, pois são eficientes e envolvem uma formulação inovadora para otimização dos processos de misturas (LEE e GILMORE, 2005).

Para Myers *et al.* (2001), essa abordagem de arranjo de misturas se faz diferente das outras categorias de experimentos de misturas, devido à sua interdependência em relação aos componentes do experimento; destacando que os valores das proporções dos experimentos devem seguir a mesma restrição da estrutura do arranjo, ou seja, o valor da soma das proporções devem totalizar uma unidade, porém, o único diferencial desta abordagem se dá pela variável de processo que compõem o arranjo, sendo esta uma variável ao qual independe dos valores dos componentes.

CORNELL (2011) afirma que ao adicionar a variável de processo em um experimento de misturas, o mesmo poderá intensificar ainda mais o experimento, podendo melhorar a exploração do processo onde se consegue observar as variáveis que mais afetam a respostas, e ainda, verificar se houve alterações nas propriedades da misturas ou até mesmo nas variáveis do processo que podem modificar a mistura ótima.

Para que se consiga compreender melhor um arranjo de misturas combinado com variável de processo, Cornell (2011) ilustra na figura 7 um exemplo, onde se tem três tipos de peixes de água salgada ( $x_1, x_2, x_3$ ) e três variáveis de processo ( $z_1, z_2, z_3$ ).



**Figura 7** - Demonstração gráfica de arranjo de misturas combinado com variável de processo

**Fonte** - Cornell, (2011)

Em seu exemplo apresentado pela figura 7, (CORNELL, 2011) explica que os triângulos nos vértices representam o espaço de composição dos três tipos de peixes. Estes triângulos são posicionados nos oitos vértices de um cubo atuando como diferentes combinações de configuração baixa e alta das variáveis de processo. Outro fator importante que a figura ilustra, são os triângulos aos quais apresentam texturas sombreadas, pois nestas áreas é que se encontram todas as combinações ideais para o arranjo de misturas.

As análises a serem realizadas neste tipo de arranjo de misturas envolvem a modelagem da função que será vinculada aos componentes da misturas e fatores de processo. E ainda para a verificação dos modelos, utiliza-se de forma semelhante os arranjo de misturas original.

### 2.3.3 Método Taguchi

O método Taguchi surgiu pelo engenheiro e estatístico Dr. Genichi Taguchi, com intuito de conduzir um experimento com mais eficiência e auxiliando na redução de custos experimentais (HUANG, HUNG e YANG, 2016; PIMENTA *et al.*, 2012). Este método traz uma proposta que define a qualidade de um produto com a menor variabilidade da função produto, além de avaliar as consequências de seu uso (ALMEIDA e TOLEDO, 1989; GHANI, CHOUDHURY e HASSAN, 2004). Segundo Almeida e Toledo (1989), é possível destacar que Taguchi associa o conceito de perda à qualidade, facilitando a quantificação em valores monetários da qualidade do mesmo.

Segundo Pimenta *et al.* (2012), a metodologia se destaca em duas abordagens importantes:

- A busca pela melhoria contínua com menor desperdício, reduzindo a variabilidade do produto, e;
- Repercutir diretamente em redução de custos em experimentações, utilizando a otimização do processo.

Bilga *et al.* (2016) destacam duas abordagens principais deste método que são as matrizes ortogonais e a relação Sinal Ruído (S/N). As matrizes ortogonais são usadas para economia do tempo e custos de experimentos (LI *et al.*, 2016). Já S/N que será tratada neste trabalho, é utilizada para medir as características de desempenho dos valores desejados das respostas (PIMENTA *et al.*, 2012).

Com isto, a relação S/N é dividida em três equações, tais como (MANIVEL e GANDHINATHAN, 2016):

Quanto menor-melhor:

$$S/N = -10 * \log 1/n (\sum y^2) \quad (2.18)$$

Quanto maior-melhor:

$$S/N = -10 * \log 1/n (\sum 1/y^2) \quad (2.19)$$

Quanto nominal-melhor:

$$S/N = 10 * \log \frac{\bar{y}}{S_y^2} \quad (2.20)$$

Onde  $\bar{y}$  é a média do dado da resposta a serem observadas,  $S_y^2$  é a variância de  $y$ ,  $n$  é o número de observações e  $y$  são os dados observados de cada características da resposta.

De modo geral, o método Taguchi utiliza deste conceito de perda, para medir a características de desempenho de um produto em relação à qualidade, partindo de uma métrica de aproximação  $L(y)$  por uma função quadrática extraída a partir da expansão em série de Taylor (ALMEIDA e TOLEDO, 1989). Essa metodologia, mostra um espaço no qual sempre existirá uma tendência a variabilidade de produto em ambiente de processo.

## 2.4 Desirability

Segundo Vera Candioti *et al.* (2014), o método de *desirability* vem sendo um método amplamente utilizada nos processos industriais, por ser uma técnica de otimização de múltiplas respostas, onde a otimização ocorre simultaneamente (MENDES *et al.*, 2016). Além do mais, o método é visto como característica de “qualidade” para processos de múltiplas respostas, pois se houver uma só fora dos limites desejados, isto torna-se indesejável (DEL CASTILLO, 2007).

Em 1965, o método ou função *desirability* foi apresentado por Harrington, responsável por trazer uma técnica capaz de tratar respostas simultaneamente, posteriormente a técnica foi aperfeiçoada pelos autores Derringer e Suich (1980). Os autores Van Gyseghem *et al.* (2004) dizem que o método *desirability* é caracterizado como método multicritérios para avalia um determinado conjunto de respostas simultâneas. Para Lima *et al.* (2011), o método é utilizado para determinar melhores condições de ajuste para processos, com isto, torna possível a otimização de diversas respostas simultaneamente (MENDES *et al.*, 2016).

Este método possui um princípio bem simples em como deve ser aplicado a otimização de múltiplas respostas: em primeira etapa, deve-se realizar uma conexão entre as variáveis

independentes e as respostas, isto pode ocorrer quando se utiliza outros métodos, tais como *OLS* e *MSR*. Através desta aglutinação, pode-se então utilizar a formulação de Harrington (1965), onde cada resposta individual do conjunto original passa por uma transformação, tal que  $d_i$  pertencente ao intervalo  $0 \leq d_i \leq 1$ . Partindo disto, o valor da  $d_i$  aumenta gradativamente quando a  $i$ -ésima resposta se aproxima dos limites determinados.

Segundo Montgomery (2012) e Bashiri *et al.* (2008) a Equação (2.21) a seguir, descreve a combinação das respostas transformadas por meio da média geométrica, encontrando o índice global D:

$$D = (d_1(y_1) * d_2(y_2) * ... * d_n(y_n))^{\frac{1}{n}} \quad (2.21)$$

Onde o valor D avalia os níveis do conjunto combinado de respostas e  $n$  representa o número de respostas. Vale ressaltar que o índice global (D), é um valor pertencente ao intervalo entre 0 e 1, onde será maximizado assim que todas as respostas se aproximarem o máximo possível de suas especificações (PAIVA, 2006). Para Rossi (2001), o valor de D tem que se aproximar de 1 para que as respostas originais alcancem os limites de especificações propostos pelo pesquisador. Porém, se o valor de D aproximar de 0, isto significa, que os valores para suas respostas não são desejáveis.

Segundo Paiva (2008), ao utilizar o método *desirability* o ponto de ótimo geral do sistema só será alcançado se houver a maximização da média geométrica extraída a partir das funções individuais, destaca ainda que uma das vantagens de se utilizar a métrica do método é que a solução global poderá ser alcançada de maneira equilibrada, de tal forma que, todas as respostas atinjam os valores esperados, fazendo com que o algoritmo seja forçado a aproximar os valores das especificações impostas pelo pesquisador.

Para Ortiz Francisco *et al.*, (2004), a desvantagem que a função *desirability* apresenta é o aumento da não linearidade de D, ou seja, conforme o número de variáveis respostas vão aumentando, a técnica poderá conduzir a ótimos locais apenas. Mas como observado na literatura, o método é um dos mais conhecidos e utilizados, e por isso, torna-se uma das ferramentas amplamente empregadas na indústria.

O quadro 3, apresenta os sentidos de otimização, propriedades dos sentidos de otimização e forma gráfica que o método *desirability* emprega:

**Quadro 3** - Tipos de método de otimização de Derringer

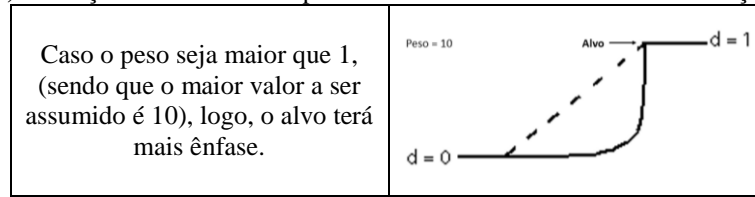
Qual finalidade?	Atributos	Gráfico
Minimizar	Nesta etapa ocorre que o valor da função <i>desirability</i> aumenta enquanto que o valor da resposta original se aproxima do valor alvo mínimo. Isto se dá no limite abaixo do alvo, $d=1$ ; e acima do limite superior, $d=0$ .	
Normalizar	Nesta etapa ocorre que o valor da função <i>desirability</i> só aumenta apenas quando a resposta se move em direção ao alvo. $d=0$ ; no alvo $d=1$ .	
Maximizar	Nesta etapa ocorre que o valor da função <i>desirability</i> só aumenta quando o valor da resposta também aumenta. Abaixo do limite inferior, $d=0$ ; acima do alvo, $d=1$ .	

Fonte - PAIVA, (2006).

Como apresentado no quadro 3, o algoritmo proposto por (Derringer, 1994) possui maneira peculiares quanto ao fator de otimização, uma vez que, o pesquisador pretende otimizar uma função, o mesmo deve definir o sentido de sua otimização para alcançar o valor que se pretende encontra, seja ele de máximo, mínimo ou normal de uma resposta. Além desta característica, outra condição que é bem vista na otimização, são os limites que também são fatores que se esperam deste tipo de otimização, e por fim, as respectivas importâncias das respostas individuais que se deseja obter, o quadro 4 a seguir estabelecem os critérios desta melhoria.

**Quadro 4 (a)** - Relações de níveis de importâncias entre alvos e os limites da função *Desirability*

Caso o peso seja menor que 1, (sendo que o menor valor a ser assumido é 0,1), logo, o alvo terá menos ênfase.	
Para um peso igual a 1, o nível de importância dada ao alvo torna-se igual ao nível de importância estabelecidas aos limites.	

**Quadro 5 (b)** - Relações de níveis de importâncias entre alvos e os limites da função *Desirability*

Fonte - PAIVA, (2006).

Quando se trata da otimização aplicando o método *desirability*, os autores (MURPHY *et al.*, 2005; MURPHY, 2004) destacam que quando se aplica a maximização de uma resposta, a formulação transformada da função *desirability* é empregada da seguinte maneira:

$$d_i = \begin{cases} 0 & Y < LSL \\ \left[ \frac{Y - L_i}{T_i - L_i} \right]^n & L_i < Y < T_i \\ 1 & Y > T_i \end{cases} \quad (2.21)$$

Onde  $L_i$  e  $T_i$  são, respectivamente, o maior e o menor valor considerável para a  $i$ -ésima resposta. O expoente  $n$  atribuída na Equação (2.21), representa a preponderância dos limites, isto é, os maiores valores que a unidade deve utilizar quando a resposta cresce rapidamente acima de  $L_i$ . Logo,  $d_i$  aumenta lentamente, na mesma proporção que a resposta é maximizada. Desta forma, ao se maximizar  $D$  a  $i$ -ésima resposta deverá possuir um valor acima de  $L_i$ .

Para minimizar uma função *desirability* deve-se utilizar a Equação (2.22):

$$d_i = \begin{cases} 0 & Y < LSL \\ \left[ \frac{H_i - Y_i}{H_i - T_i} \right]^n & H_i < Y < T_i \\ 1 & Y > T_i \end{cases} \quad (2.22)$$

Onde  $H_i$  representa o valor aceitável mais alto a  $i$ -ésima resposta.

E por fim, o último método aplicado para otimização da função *desirability* que se trata de uma otimização normalizada, onde se possui um alvo de uma resposta. Esta otimização possui uma transformação bilateral, isto é, ocorre quando a resposta de interesse possui duas restrições a de máximo e a de mínimo. Para tal, a formulação desta função se dá pela seguinte Equação (2.23):

$$d_i = \begin{cases} 0 & Y < LSL \\ \left[ \frac{Y - L_i}{T_i - L_i} \right]^n & L_i < Y < T_i \\ \left[ \frac{H_i - Y_i}{H_i - T_i} \right]^n & H_i < Y < T_i \\ 0 & Y > T_i \end{cases} \quad (2.23)$$

Além destas abordagens de otimização pelo método *desirability*, o autor Derringer (1994) propõe outro tratamento para o método, trazendo uma definições ponderada para dentro da função, ou seja, definindo uma média geométrica ponderada para diferentes respostas, tornando-se possível aplicar importâncias para as respostas, fazendo com que houvesse mudança no método; desta forma, a função passou a ser descrita da seguinte maneira:

$$D = (d_1(y_1)^{w_1} * d_2(y_2)^{w_2} * ... * d_n(y_n)^{w_n})^{\frac{1}{\sum w_i}} \quad (2.24)$$

Onde  $w_i$  representa a importância relativa para cada resposta.

Embora exista diversas maneiras e mais robustas de aplicar o método, esta pesquisa se define pela aplicação do método tradicional, por se tratar de um abordagem mais comum utilizada pela literatura.



## 3 MÉTODO E DESENVOLVIMENTO DA PESQUISA

### 3.1 Considerações Iniciais

Neste capítulo, será apresentado uma abordagem sistemática para a otimização do algoritmo de otimização enxame por partícula (*PSO*). Esta seção tem por objetivo descrever os principais procedimentos aplicados ao tema proposto para fins de conhecimento do leitor, bem como sua natureza, abordagens, objetivos e métodos de pesquisa.

Estes procedimentos serão aplicados conforme descrito na literatura, através de um método de pesquisa “experimental”, onde este será capaz de conduzir o pesquisador a um ponto que concretizará o aprimoramento do otimizador. A seção a seguir mostra o surgimento do método aplicado à dissertação e suas formas detalhadas para o aprimoramento do objeto de estudo.

#### 3.1.1 Característica do método de pesquisa

Segundo Gil (2008), o método de pesquisa se caracteriza pelo objetivo fundamental de chegar à veracidade dos fatos de uma ciência. O autor supracitado enfatiza que para um conhecimento se tornar científico o mesmo deve ser identificado como operações mentais e técnicas que possibilitam sua verificação, destaca também que cientistas e filósofos da ciência falam que há uma diversidade de métodos que são determinados pelos tipos de objeto que se pretende investigar e/ou pela classe de assuntos que se deseja desvendar. Para Andery *et al* (2004), os métodos de pesquisa consistem em um conjunto de normas e princípios, com o qual a comunidade acadêmica concorda com a construção do conhecimento científico.

Dado que, o método de pesquisa possui diversas classes, o que será abordado neste trabalho engloba as técnicas aplicadas à área da engenharia de produção. Em (Berto e Nakano, 2000; Filippini, 1997), os autores apresentam levantamentos dos métodos de pesquisa mais comuns e utilizados na área da engenharia de produção, estes são classificados em sete tipos de metodologia a serem aplicadas, tais como:

1. Levantamento tipo *survey*;
2. Estudo de caso;
3. Modelagem (ou modelamento);
4. Simulação;
5. Estudo de campo;
6. Experimento;

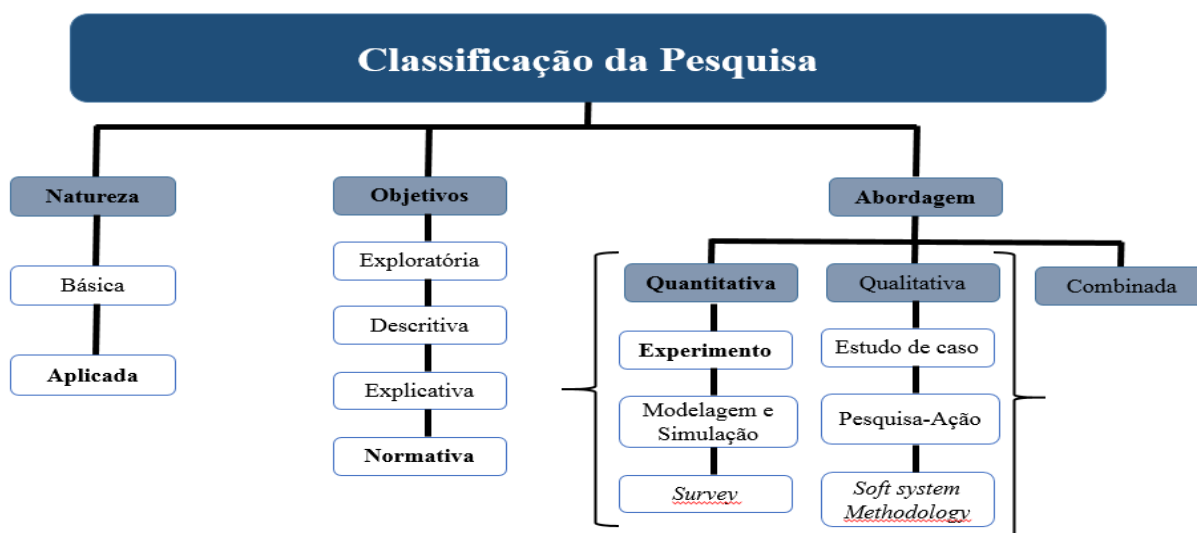
### 7. Teórico/Conceitual.

Como visto anteriormente, sete metodologias foram mostradas para ilustrar o quanto o método de pesquisa se tornou importante para o desenvolvimento de um trabalho científico. Desta forma, esta dissertação se conduzirá no método classificado como experimento.

Para melhor entendimento ao método de pesquisa utilizado neste trabalho, a seção a seguir apresenta a classificação da presente pesquisa, bem como sua, natureza, abordagem e objetivos.

### 3.1.2 Classificação do método de pesquisa

Segundo Miguel *et al.* (2012), os métodos de pesquisa podem ser classificados conforme a figura 8:



**Figura 8** - Classificação da Pesquisa  
Fonte - Miguel *et al.*, (2012).

Esta dissertação aplicou a seguinte classificação de pesquisa para o objeto de estudo:

- Natureza: Aplicada;
- Objetivo: Normativo;
- Abordagem: Quantitativa, sendo um experimento.

Quanto à natureza deste objeto de estudo se classifica como aplicada, pois tem o intuito de realizar aplicações práticas a fim de alcançar os resultados esperados (APPOLINÁRIO, 2012). Em relação aos objetivos deste método de pesquisa, engloba duas categorias que predominam este método, as pesquisas axiomáticas e as pesquisas empíricas (BERTRAND e FRANSO, 2002). De modo geral, as pesquisas classificadas como axiomáticas têm o intuito de esclarecer a estrutura do problema descrito no modelo, ou seja, têm a preocupação de obter soluções para o modelo em questão. E as pesquisas empíricas têm a responsabilidade de

assegurar que exista adequação entre o modelo desenvolvido e o processo real. Vale ressaltar que este trabalho será classificado como pesquisa axiomática tendo como objetivo esclarecer a estrutura do algoritmo PSO, visando aprimorar a estrutura do otimizador através de modelos matemáticos em busca de encontrar uma solução viável ao modelo em questão.

A abordagem utilizada para esta dissertação é classificada como quantitativa, pois trata-se de modelos abstratos descritos em linguagens matemáticas e computacionais, que utilizam de técnicas analíticas e experimentais para cálculo de valores numéricos da propriedade do sistema. Destaca-se ainda, que os modelos quantitativos compreendem um conjunto de variáveis de controle que se alteram em um domínio específico e variáveis de desempenho que inferem na qualidade das decisões obtidas a partir das relações causais e quantitativas definidas entre essas variáveis (BERTRAND e FRANSSO, 2002).

Ao realizar estudos sobre as áreas do método de pesquisa em questão, percebeu-se que o objeto de estudo teve um direcionamento conforme a figura apresentada por MIGUEL *et al.* 2012, onde notou-se que o método a ser utilizado é o experimento, uma vez que, se tem o intuito de realizar testes para verificação de algum tipo de causa e efeito que o resultado sofre sob variáveis controladas. Esta técnica tem-se tornado referência em investigar a causalidade de sistemas. Outra relação que esta técnica emprega é facilidade que o pesquisador encontra para estabelecer relações de causa e efeito, tornando o experimento um modelo de delineamento da pesquisa (BRYMAN, 1989).

Para Gil (2008), o experimento demonstra ser o melhor exemplo de pesquisa científica. O autor ainda destaca, que este método é essencial para delineamento da pesquisa por depender de um objeto de estudo, variáveis que influenciam determinados sistemas a serem selecionadas, formas de controle destas variáveis e da observação dos efeitos que as variáveis produzem no objeto.

Em Martins *et al.* (2012), os autores destacam algumas etapas para o planejamento de um experimento, tais como:

1. Planejamento do Experimento: esta fase o pesquisador define as variáveis a serem exploradas.
2. Operacionalização das variáveis: será a etapa onde ocorrerá a mensuração das variáveis definidas pelo pesquisador.
3. Estabelecimento das relações causais (hipóteses): ocorre a definição das variáveis dependentes e independente por parte do experimentador.
4. Definição das técnicas de análise dos dados do experimento: o pesquisador nesta etapa define uma técnica estatística para condução do experimento.

5. Montagem do banco de ensaio: preparação da coleta de dados por parte do pesquisador.
6. Especialização do tempo para a condução do experimento: o tempo nesta etapa torna-se fundamental para pesquisas a serem realizadas no futuro.
7. Projeto do Experimento: nesta etapa o pesquisador deve seguir o estabelecimento dos níveis para cada variável de controle, sequência de cada evento da experimentação e a definição do número de eventos do experimento.
8. Realização do experimento e coleta de dados: nesta fase o pesquisador deverá seguir o que foi considerado na etapa anterior.
9. Análise estatística: o pesquisador deverá aplicar aos dados colhidos uma técnica de análise estatística.
10. Análise dos Resultados: neste passo o pesquisador verificará com a literatura se houve modificações nos resultados obtidos pela análise dos experimentos realizados.
11. Conclusão: o pesquisador apresentará os resultados em relação à hipótese estabelecida na etapa 3.
12. Por fim, redação e publicação dos resultados.

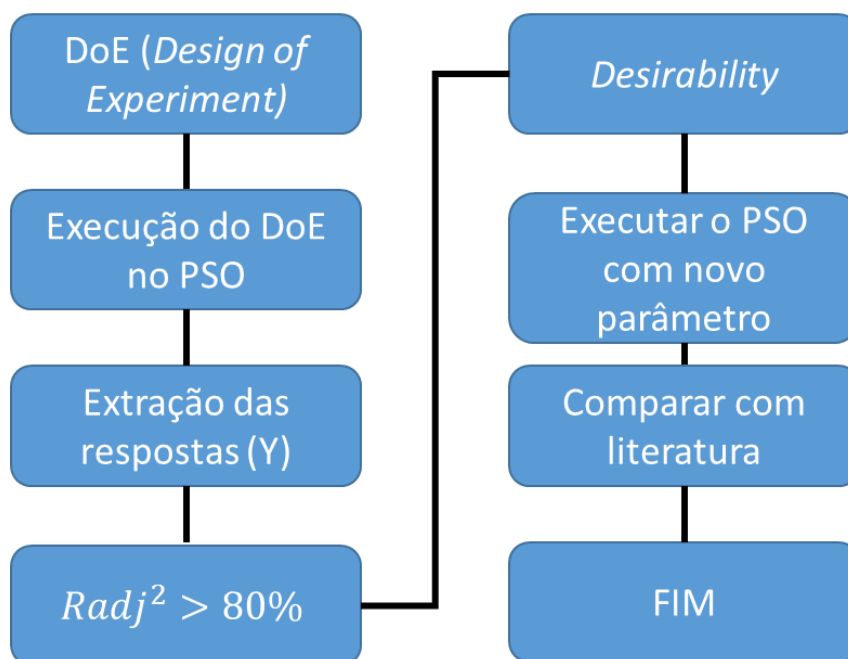
Visto que o método experimental deverá seguir algumas etapas para conduzir o pesquisador a um bom experimento. A seção a seguir apresenta método proposto pelo autor desta pesquisa utilizando a técnica de experimento.

### **3.2 Método proposto para aperfeiçoamento do algoritmo de otimização**

Como apresentado na seção anterior, o método aplicado a este trabalho foi a técnica de experimento. O método de pesquisa experimental está sendo utilizado para aperfeiçoar o objeto de estudo de tal forma a aumentar a eficiência e eficácia da performance do algoritmo *PSO*.

Para realização do método proposto, a otimização irrestrita por enxame de partícula assistido por polinômios canônicos de misturas vai empregar 3 técnicas amplamente discutidas na literatura.

Através da combinação destas técnicas, o método proposto visa avaliar a influência dos parâmetros do algoritmo de otimização sob o controle realizado por arranjos experimentais e determinar a qualidade na solução final do algoritmo de otimização através das respostas tempo e iteração.



**Figura 9** - Método proposto para aperfeiçoamento do PSO

Fonte - Próprio Autor

Dessa forma, a figura 9 mostra sete etapas que essa dissertação seguiu para o aperfeiçoamento do algoritmo de otimização enxame por partícula, a fim de aumentar a qualidade deste algoritmo. Estas etapas serão apresentadas a seguir:

- 1º. Passo:** Nesta primeira etapa, foi realizado diversos estudos sobre como deveria ser tratado os parâmetros de um algoritmo de otimização. Partindo disto, utilizou-se uma taxonomia desenvolvida por Eiben e Smit (2011) e Eiben *et al* (2007), onde em seu trabalho mostrava as várias maneiras de conduzir um ajuste dos parâmetros de um algoritmo de otimização. Dessa forma, aplicou-se a esta dissertação uma abordagem *off-line*, ou seja, uma maneira de tratar os parâmetros de forma concisa, conduzindo o algoritmo a uma definição padrão transformando-o em uma técnica de tentativa e erro até encontrar uma solução próxima do adequado. Vale ressaltar que nesta abordagem existe 3 meios de realizar o tratamento dos parâmetros de algoritmo de otimização: os manuais, o planejamento de experimentos e os da meta-evolução. Visto que na literatura havia diversas maneiras de ajustar um algoritmo, assim ficou decidido que a maneira que iria proceder essa pesquisa seria pelo planejamento de experimentos. Ainda nesta etapa, quatro trabalhos apresentados na literatura contribuíram significativamente para o desenvolvimento desta dissertação, onde serviram como modelos padrões a serem seguidos para a construção deste método. A pesquisa dos autores Kennedy e Eberhart (1995) auxilia este trabalho em relação ao algoritmo padrão, pois através da pesquisa realizada por estes autores tornou-se possível o

desenvolvimento do otimizador. Em consequente, os trabalhos de Shi e Eberhart (1998) e Kennedy (2010) proporcionando um entendimento em relação aos parâmetros do algoritmo de otimização, tornando possível a implementação destes trabalhos em conjunto a esta pesquisa. E por fim, os autores Clerc e Kennedy (2002), que trouxeram formas diversificadas de ajustar os mesmos parâmetros do algoritmo enxame por partículas (*PSO*). Visto que diversas aplicações foram realizadas no algoritmo padrão, esta dissertação em seu primeiro passo construirá um arranjo experimental utilizando o planejamento de experimento (*DOE*) para os parâmetros deste otimizador. Os parâmetros selecionados para aplicação do *DOE* foram o aprendizado cognitivo (*c1*), aprendizado social (*c2*) e peso da inércia (*w*) conforme apresentado nos trabalhos citados acima. Para realizar a calibração do algoritmo de otimização optou-se por selecionar o arranjo de misturas da técnica de planejamento de experimentos (*DOE*), uma vez que, ambos parâmetro de atração apresentam uma características de proporção demonstrada na equação (2.1) apresentados pelos trabalhos de (SINČÁK, 2002). Sendo assim, o tipo escolhido de arranjo de misturas foi o *simplex lattice* com dois componentes para grau um *lattice* de 10 e uma variável de processo, gerando então 26 configurações de experimentos para execução do algoritmo.

- **2º. Passo:** Para a segunda etapa do fluxograma, foi desenvolvido o algoritmo de otimização que se encontra indexado em ANEXO (A). Este algoritmo enxame por partícula (*PSO*) foi desenvolvido em MATLAB e foram efetuados testes com o auxílio de um computador Intel core i5 N450 2.4 GHz com 4GB de RAM, no qual, foram testados os valores do arranjo experimental gerados no passo anterior para explorar a eficiência e eficácia do algoritmo. Estes experimentos foram realizados em funções *benchmark* como esférica, rosenbrock e rastrigin.
- **3º. Passo:** Com o objetivo de reduzir a dispersão das partículas do algoritmo *PSO* no espaço de busca por soluções ótimas das funções *benchmark*, foram extraídas as respostas distância, tempo e iteração. Para se obter a resposta distância dessa dissertação foi aplicada uma métrica de dissimilaridade conhecida como distância euclidiana, com o intuito de reduzir a variabilidade da dispersão das melhores partículas do enxame denominadas *Gbest* em relação ao ótimo já conhecido de cada função *benchmark* empregada nesta dissertação, ou seja, a Equação (3.1) apresenta dois vetores de valores que define o espaço multidimensional da função que se pretende otimizar.

$$D_{xy} = \sqrt{\sum_{i=1}^p (X_i - Y_i)^2} \quad (3.1)$$

Onde,  $D_{xy}$  representa a distância entre dois vetores.  $X_i$  representa o vetor de  $G_{best}$  obtido a partir da execução do algoritmo de otimização enxame por partícula e  $Y_i$  um vetor do ponto ótimo já conhecido pela função *benchmark*. Outra resposta coletada foi o tempo, onde o mesmo tem por finalidade contabilizar a quantidade em segundos que os valores obtidos pelo arranjo experimental para os parâmetros do algoritmo de otimização levaria para encontrar a melhor solução, tornando-o uma regra de quanto menor for o tempo melhor a eficiência do otimizador. Além destas respostas, outra foi utilizada para auxiliar na eficiência da performance do algoritmo. A resposta iteração teve a finalidade de verificar a quantidade de passos que a configuração gerada pelo arranjo experimental encontrava a solução da função que se pretende otimizar. Logo após a coleta das respostas distância, tempo e iteração no terceiro passo, foi utilizado outro método para melhoria da variabilidade dos dados e aumento da qualidade da resposta. O método aplicado é conhecido como Taguchi e a técnica utilizada para redução da dispersão dos dados foi a sinal/ruído, uma vez que, a proposta tem por finalidade diminuir a dispersão das distâncias calculadas através das melhores partículas do enxame. Dentre as técnicas apresentadas seção (2.3.3), a estratégia usada nesta dissertação foi a quanto menor/melhor, como descrito na equação (2.18). Este método teve o objetivo de transformar a respostas distância coletada do algoritmo de otimização.

- **4º. Passo:** No quarto passo, será aplicado uma análise de variância (ANOVA) com o intuito de verificar a relação entre a variável dependente conhecida como distância com as variáveis independentes conhecidas como atração cognitiva (c1), atração social (c2) e peso da inércia (w). A variável dependente foi estimada através da equação (2.15) representando um modelo cúbico completo, feito pelo software *Minitab*®.
- **5º. Passo:** No quinto passo, foi utilizada a técnica de otimização *desirability* com o intuito de obter novos parâmetro para o algoritmo de otimização enxame por partícula, sendo aplicada a todas funções *benchmark* desta dissertação. Nesta etapa, o sentido de otimização foi a maximização da resposta distância, uma vez que, aplicada a técnica de sinal/ruído nesta resposta, tornou-se possível a busca pela maior valor da distância, significando que quanto maior for a distância menor a dispersão das partículas na

busca pela solução ótima das funções *benchmark* que se pretende otimizar. Partindo desta otimização, novos valores para os parâmetros do algoritmo de otimização foram encontrados.

- **6º. Passo:** Após encontrar novos parâmetros para o algoritmo de otimização enxame por partícula, foi realizado novos testes nas funções *benchmark* executando esses novos parâmetros e coletando novamente as mesmas respostas.
- **7º. Passo:** Nesta etapa, foi aplicado uma comparação dos parâmetros ótimos encontrados pela otimização em oposição aos parâmetros sugeridos pela literatura. As respostas tempo e iteração foram submetidas a um teste de hipóteses do tipo *two sampe-t*, com o intuito de verificar se os novos parâmetros possui uma eficiência e eficácia na performance do algoritmo de otimização. Este teste de hipótese pode auxiliar na validação da performance, pois através deste teste de hipóteses pode-se afirmar a nível de confiança que os valores obtidos das respostas iteração e tempo através dos novos parâmetros encontrado se mostraram melhores que os valores dos parâmetros sugeridos pela literatura.

Visando apresentar um padrão a ser seguido pelo leitor, o método proposto acompanhará os passos da figura 10 com o intuito de auxiliar na sequência lógica de raciocínio. A seção a seguir, apresentará as respostas distância, iteração e tempo, bem como os resultados para cada resposta aplicada as suas funções *benchmark* apresentada nesta dissertação.



## 4 ANÁLISE DOS RESULTADOS

### 4.1 Considerações Iniciais

Esta seção tem por objetivo de esclarecer a aplicação do método proposto nas funções *benchmark*, bem como suas análises feitas dos resultados obtidos em busca da melhoria do desempenho do algoritmo de otimização.

De acordo com Ahuja, Magnanti e Orlin (1993), a análise empírica tem como objetivo estimar o comportamento prático de novos métodos, testando o desempenho através de aplicações em problemas. Destacam também que existem outras abordagens básicas para medição do desempenho de um método, sendo que para esta pesquisa foi adotado a análise empírica.

Com a intenção de verificar a performance do algoritmo de otimização através do método proposto, foram usadas funções de testes conhecidas pela literatura. São elas:

- Função Esférica;
- Função Rosembrock;
- Função Rastrigin.

Para investigar o desempenho de um algoritmo de otimização, os trabalhos de CHEN *et al.*, (2018b) e SHI e EBERHART (1999) utilizam estas e outras funções para testes com o intuito de representar o aumento da qualidade das soluções pelo método computacional (COUTO *et al.*, 2015). Estas funções foram escolhidas por se tratarem de funções unimodais e multimodais, ou seja, ambas mostram sentidos de otimização de máximo ou mínimo, sendo que o diferencial entre elas se dá na quantidade de sentido de otimização que a função pode apresentar, isto é, nas funções unimodais pode-se obter um único máximo ou mínimo, enquanto que nas multimodais pode-se encontrar várias soluções de máximos ou mínimos sejam elas locais ou globais.

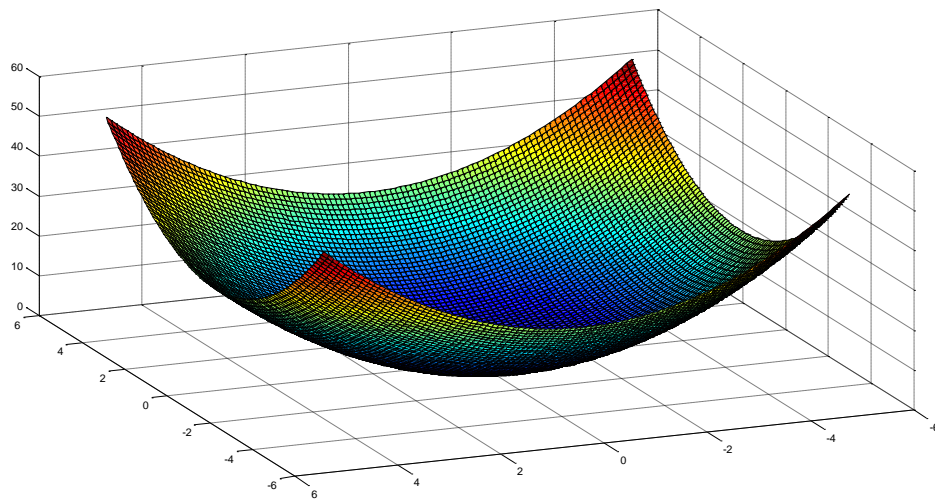
Desta forma, esta seção possui o propósito de apresentar uma leitura breve sobre os tipos de funções que foram utilizadas, bem como os trabalhos de onde foram utilizadas para justificar o uso destas funções de testes aplicando o método proposto. Visando atingir com êxito a proposta de realizar o aperfeiçoamento do algoritmo de otimização enxame de partículas testando em funções que possuem seus problemas de otimização já definidos.

## 4.2 Aplicação do método na função teste esférica

Para uso de teste do método proposto, foi escolhido dentro de todas funções teste da literatura, a função esférica; pois a mesma apresenta uma forma convexa, contínua e unimodal, que se faz adequada para testes de otimização.

A função esférica é definida pela equação (4.1) e demonstrada graficamente pela figura 10:

$$f(x_1, \dots, x_n) = \sum_{i=1}^n (x_i^2) \quad (4.1)$$



**Figura 10** – Representação gráfica da função Esférica

Através do algoritmo, o método proposto irá percorrer nesta função o seguinte espaço de busca  $-5.12 \leq x_n \leq 5.12$  considerando o valor zero como o mínimo global, ou seja, adotando-o como ponto ótimo da função (PANT *et al.*, 2009).

Para realizar o experimento através deste método, o algoritmo foi configurado através do arranjo de misturas com variável de processo definindo os seguintes parâmetros de calibração: aprendizagem cognitiva (c1), aprendizagem social (c2) e peso da inércia (w); para um grau do *simplex-lattice* igual 10 gerando então 26 experimentos conforme tabela 2:

**Tabela 2 (a)** - Arranjo experimental gerado pelo método DoE

Parâmetros do algoritmo			Resposta
c1	c2	w	Distância
1,00	0,00	0,4	13,70
0,90	0,10	0,4	155,31
0,80	0,20	0,4	717,97
0,70	0,30	0,4	771,05
0,60	0,40	0,4	779,15
0,50	0,50	0,4	780,94
0,40	0,60	0,4	764,22
0,30	0,70	0,4	744,17
0,20	0,80	0,4	715,07

**Tabela 2 (b)** - Arranjo experimental gerado pelo método DoE

Parâmetros do algoritmo			Resposta
0,10	0,90	0,4	692,88
0,00	1,00	0,4	659,39
0,75	0,25	0,4	759,05
0,25	0,75	0,4	730,64
1,00	0,00	0,9	13,55
0,90	0,10	0,9	50,46
0,80	0,20	0,9	57,92
0,70	0,30	0,9	61,88
0,60	0,40	0,9	64,07
0,50	0,50	0,9	68,07
0,40	0,60	0,9	64,72
0,30	0,70	0,9	59,37
0,20	0,80	0,9	51,20
0,10	0,90	0,9	44,23
0,00	1,00	0,9	38,82
0,75	0,25	0,9	56,74
0,25	0,75	0,9	54,14

Após a geração do arranjo experimental, foi realizado a execução de cada experimento por 50 vezes, com a finalidade de assegurar a validade da resposta alcançada pela repetição do processo. Além de aplicar a repetição, o algoritmo teve uma calibração na população de 80 partículas com um número máximo de 200 iterações. Logo após a realização dos experimentos já configurados em suas funções de testes, foram coletados as respostas distância transformada pela técnica de sinal ruído de Taguchi conforme demonstrado na tabela 7, além também de ter reunidos as repostas tempos e iteração encontradas no anexo (B).

Em seguida, foram realizadas análises estatísticas em relação à resposta distância através do planejamento de experimento (DOE) pela técnica de análise de regressão (ANOVA), onde foi possível estimar os coeficientes do modelo para verificação da explicação da coleta de dados desta resposta, de acordo com a tabela 3:

**Tabela 3** - Determinação dos coeficientes do modelo da Distância com sinal ruído

Estimação dos coeficientes por meio de regressão da resposta distância					
Termos	Coef	SE Coef	T	P	VIF
c1	17,3	8,038	*	*	3,863
c2	344,5	7,875	*	*	4,072
c1*c2	962,0	28,411	33,86	0,000	4,738
c1*c2*(-)	962,4	47,106	20,43	0,000	2,037
c1*c2*(-) <sup>2</sup>	826,3	101,520	8,14	0,000	2,793
w	-307,1	7,875	-38,99	0,000	11,387
c1*w	306,7	11,073	27,70	0,000	7,330
c1*c2*w	-810,5	28,411	-28,53	0,000	4,738
c1*c2*(-)*w	-874,5	47,106	-18,57	0,000	2,037
c1*c2*(-) <sup>2</sup> *w	-765,4	101,520	-7,54	0,000	2,793
S = 11,6686		PRESS = 208146			
R-sq=99,93%		R-sq(press)=92,90%		R-sq(adj)=99,89%	

Fonte - Próprio autor

Visto que os coeficientes de determinação indicam o modelo que foi explicado através dos dados coletados, (MONTGOMERY, 2012) sugere verificar o  $R-Sq(adj)$ , pois tal termo indica a medida descritiva de qualidade do ajuste do modelo de regressão. O autor destaca também que o termo  $R-Sq(adj)$  seja acima de 80%, uma vez que, o propósito deste termo é a definição da explicação da coleta de dados da resposta que se pretende otimizar. Além de analisar o tal termo, a tabela 4 demonstra que o “p\_valor” dos termos apresentam valores abaixo de 5%, ou seja, o “p\_valor” determina que os termos do modelo de regressão estão ajustados significativamente, implicando na explicação dos efeitos das interações entre os componentes e a variável de processo

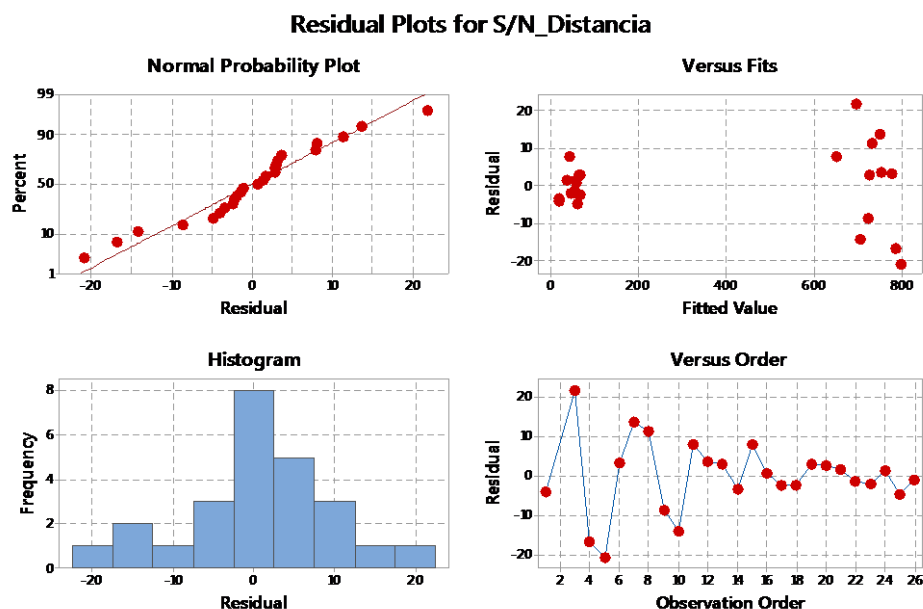
**Tabela 4** - Análise de variância para a resposta distância

<b>Análise de Variância para a resposta distância</b>						
<b>Regressão</b>	<b>DF</b>	<b>Seq SS</b>	<b>Adj SS</b>	<b>Adj MS</b>	<b>F</b>	<b>P</b>
	9	2930656	2930656	325628	2391,58	0,000
Componentes						
Linear	1	89140	118849	118849	872,89	0,000
Quadrático	1	198074	156101	156101	1146,48	0,000
c1*c2	1	198074	156101	156101	1146,48	0,000
Cúbico completo	1	54311	56833	56833	417,41	0,000
c1*c2*(-)	1	54311	56833	56833	417,41	0,000
Quártico Completo	1	1565	9020	9020	66,25	0,000
C1*c2*(-)2	1	1565	9020	9020	66,25	0,000
Componente*w						
Nenhum componente	1	2361818	206992	206992	1520,5	0,000
w	1	2361818	206992	206992	1520,5	0,000
Linear	1	52910	104442	104442	767,07	0,000
c1*w	1	52910	104442	104442	767,07	0,000
Quadrático	1	118423	110794	110794	813,73	0,000
c1*c2*w	1	118423	110794	110794	813,73	0,000
Cúbico completo	1	46675	46931	46931	344,68	0,000
c1*c2*(-)*w	1	46675	46931	46931	344,68	0,000
Quártico Completo	1	7739	7739	7739	56,84	0,000
c1*c2*(-)2*w	1	7739	7739	7739	56,84	0,000
Erro residual	15	2042	2042	136		
Total	24	2932698				

Considerando a análise realizada, um modelo quártico completo foi adequado para a resposta distância de acordo com a tabela 3, sendo assim este ajuste de modelo pode ser descrito pela seguinte equação (4.2):

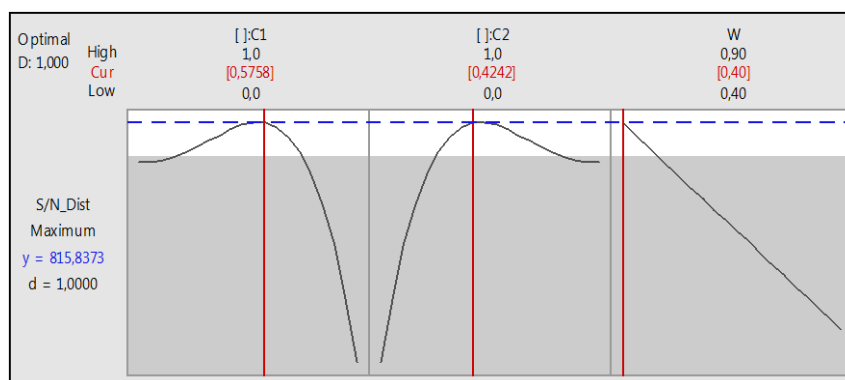
$$\begin{aligned}
 \hat{y} = & 17,3c1 + 344,5c2 + 962,0c1c2 - 962,4c1c2(c1 - c2) \\
 & - 826,3c1c2(c1 - c2)^2 - 307,1w + 306,7c1w \\
 & - 810,5c1c2w - 874,5c1c2(c1 - c2)w - 765,4c1c2(c1 \\
 & - c2)^2w
 \end{aligned} \tag{4.2}$$

Depois de realizar a adequação do modelo, a figura (XXXX) apresenta graficamente uma análise de resíduos para a resposta distância, onde a mesma foi submetida a uma investigação a fim de identificar se os dados foram distribuídos normalmente, se apresentam assimetrias ou contém algum outliers, além de verificar se os resíduos possuem variâncias constantes e se os resíduos não são correlacionados.



**Figura 11** - Análise dos resíduos para a resposta distância para a função esférica

Após a estimação dos coeficientes, análise de variância, adequação do modelo e análise dos resíduos, o próximo passo é a otimização da variável resposta distância sinal ruído. Para a otimização deste modelo foi utilizado o método de otimização *desirability*, por se tratar de uma técnica bem conhecido e utilizado na literatura para problemas complexos de otimização. Nesta fase, o método de otimização auxiliou na maximização da variável resposta distância, uma vez que, o propósito desta etapa era encontrar a maior solução para a resposta distância, considerando que quanto maior for essa resposta, menor será a dispersão das soluções no espaço de busca aumentando a qualidade da solução final. Além disto, novos parâmetros para o algoritmo de otimização enxame de partícula (*PSO*) serão obtidos, portanto comparações entre os valores sugeridos pela literatura e os valores obtidos por essa otimização serão necessárias para identificar a melhor estratégia.



**Figura 12** - Encontro do parâmetro e solução ótimo da função esférica

Pode-se observar pela figura 12, que os resultados obtidos aos parâmetros de aprendizagem cognitiva (c1), aprendizagem social (c2) e peso da inércia (w) do algoritmo de otimização recebem 0,5758 e 0,4242 respectivamente, equivalendo às proporções em formas de porcentagens. Para conversão do valor exato para esta otimização, deve-se multiplicar a porcentagem pelo valor total das misturas. Sendo assim, para esta dissertação foi utilizado a este algoritmo um valor de 4,1 como valor total da somatória dos parâmetros c1 e c2, uma vez que, ambos parâmetros empenham-se uma proporção da Equação (2.1). Os trabalhos de (MENDES; KENNEDY; NEVES, 2004) exploram de maneira mais detalhada a forma como se deve tratar ambos parâmetros.

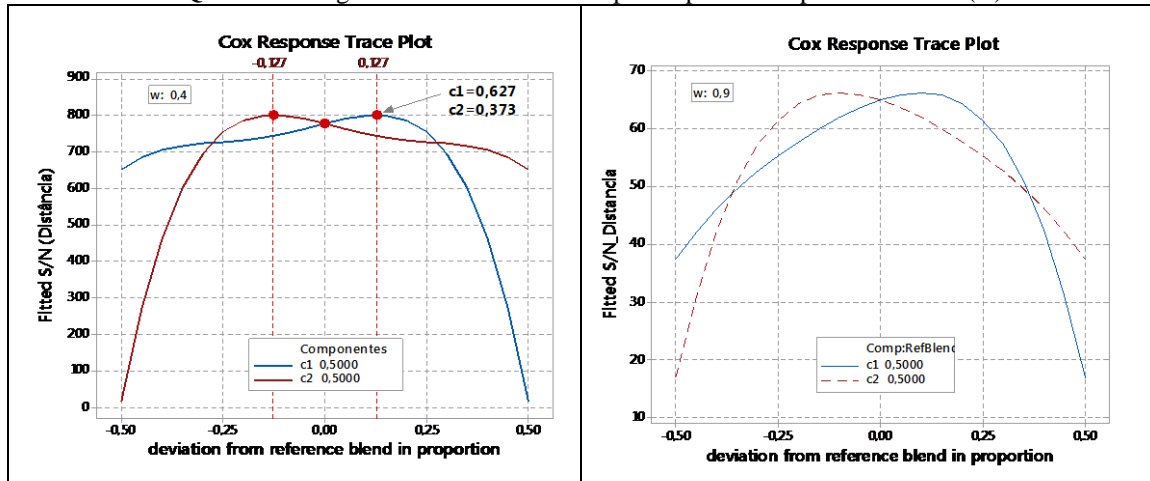
**Tabela 5** - Configuração ótima encontrado após a otimização

Otimização da Função Esférica		
c1	c2	w
2,3601	1,7392	0,4

Após realizar tal procedimento, a tabela 5 apresenta a configuração ótima para os parâmetros do algoritmo PSO, então deve-se comparar os valores encontrados pela otimização com a configuração que a literatura propõem. Segundo trabalho de Kennedy e Eberhart (1995), sugere que os valores para estes parâmetros do algoritmo padrão sejam 2 para c1 e c2, respectivamente, enquanto o peso da inércia sugere um valor entre 0,9 até 1,2 onde foi escolhido o valor 0,9 para a configuração deste parâmetro, de acordo com o trabalho de (SHI e EBERHART, 1998).

O quadro 6, reforça em forma de duas figuras o resultado obtido para esta função de teste que o parâmetro peso de inércia (w) demonstra ao passo que os valores do parâmetro alternando entre si, o resultado obtido para a resposta na maioria das vezes será melhor para o valor alcançado por esta otimização.

Quadro 6 - Figuras do resultado obtido para o parâmetro peso de inércia (w)



A figura 13 apresenta os passos do processo em que a otimização ocorreu, dividindo-a em seis imagens que representam a busca pela melhor solução na otimização da função esférica.

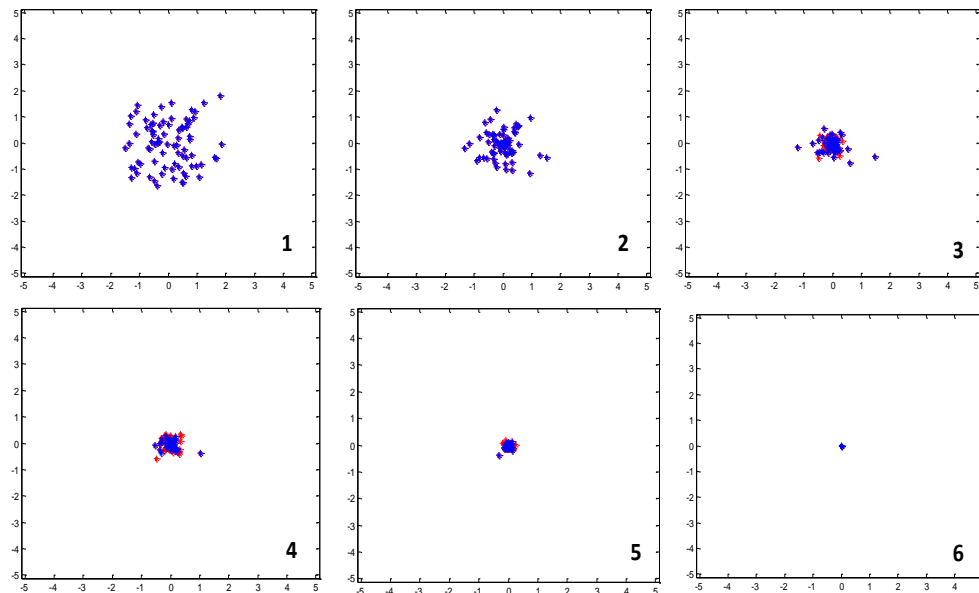


Figura 13 - Passos do processo de otimização para a função esférica

Ao observar a figura 13, pode-se compreender que o processo de otimização realizado na função de teste ocorreu de forma significativa, devido a forma como foi descrito no trabalho de (PANT *et al.*, 2009), dado que o propósito desta otimização é encontrar um mínimo global  $f(x) = 0$  em  $x = (0, \dots, 0)$  da equação (4.1).

Deste modo, a comparação entre os parâmetros citados pela literatura e método proposto ocorre através do tempo de execução em relação à busca pela solução ótima da função teste. Para Cormen *et al.*, (2002), o tempo de execução é um período constante de tempo exigido para executar cada linha do pseudocódigo. Ainda o mesmo autor destaca que cada linha pode demorar um período diferente de outra linha.

Dessa forma, um teste de hipóteses foi proposto a fim de comparar o tempo de execução de cada configuração gerado pelo arranjo de misturas. Este tempo foi extraído de cada parâmetro gerado pelo arranjo experimental e seus resultados foram indexados no ANEXO(B), tabela B.1. Partindo destes resultados, o teste de hipóteses utilizado foi do tipo *two sample-t*, onde o mesmo tem por finalidade comparar as médias. Com propósito de evidenciar o nível de confiança, a diferença entre os tempos obtidos pelo arranjo experimental, em refutação com os tempos obtidos pelo método de otimização aplicado, a Equação (4.3) representa em forma de comparação tal teste de hipóteses:

$$\begin{cases} H_0 : \mu_{otimização} = \mu_{Literatura} \\ H_1 : \mu_{otimização} < \mu_{Literatura} \end{cases} \quad (4.3)$$

Ao observar a tabela 6, pode-se perceber a existência de uma diferença em relação aos parâmetros encontrados pela otimização utilizando o método proposto em comparação aos parâmetros usados pela literatura. Usando o método *two sample-t* pode-se afirmar ao nível de 95% de confiança que as médias dos tempos de execução para ambas amostras são diferentes, dado que seu *p\_value* apresenta um valor abaixo de 5%, significando que erro presente nesta análise é mínimo. Portanto, fica comprovado que o método proposto se fez eficiente e eficaz quanto ao tempo de execução da otimização e na busca pelo ponto de ótimo da função esférica que se pretende otimizar.

<b>Tabela 6 - Análise Two Sample-t para os tempos</b>		
<b>Two Sample-T para Tempo_da_Oti vs Tempo_Literatura</b>		
<b>Amostras Individuais</b>		
<b>Estatísticas</b>	<b>Tempo_da_Oti</b>	<b>Tempo_Literatura</b>
<b>N</b>	50	50
<b>Média</b>	25,111	28,101
<b>Desvio Padrão</b>	0,28243	2,3526
<b>Diferença entres as amostras</b>		
<b>Estatísticas</b>		<b>Diferença</b>
<b>Diferença</b>		-2,9900
<b>95% CI</b>		-2,4284
<b>Diferença = Tempo_da_Oti - Tempo_Literatura</b>		
<b>P_Valor</b>	0,001	
<b>T_Valor</b>	-8,92	

Outra forma de avaliar a performance do algoritmo em relação aos parâmetros ótimos gerados pelo DOE com o parâmetro ótimo obtido pela otimização da resposta distância é o método iterativo, ou seja, através deste método podemos averiguar o quanto a resposta aproxima da solução ótima. Este método tem como principal objetivo alterar suas soluções ao



passo que todos os componentes sejam substituídos por novos e melhores a fim de alcançar a convergência, isto é, encontrar a solução ótima (SAAD, 2003). Desta forma, os resultados obtidos das iterações para cada configuração do arranjo experimental rodados nas mesmas condições se encontram indexados em (ANEXO B), tabela B.2.

Logo que coletadas, todas as iterações inclusive a do parâmetro de otimização, deve-se então aplicar um teste de hipóteses (*two sample t*) com intuito de verificar se as amostras se assemelham entre si. As amostras utilizadas para este teste são as fornecidas pela literatura em sentido oposto ao da configuração obtida pela otimização deste trabalho. Sendo assim, a tabela 7 a seguir, apresenta o resultado obtido pela análise estatística usando o teste de hipótese *two sample t* para investigar se as médias da iteração obtidas pela configuração da literatura são menores que a média obtida pelo método proposto.

Tabela 7 - Análise Two Sample-t para as iterações		
Two Sample-T para Iteração_da_Oti vs Iteração_Literatura		
Amostras Individuais		
Estatísticas	Iteração_da_Oti	Iteração_Literatura
N	50	50
Média	6,42	26,8
Desvio Padrão	5,9078	46,879
Diferença entres as amostras		
Estatísticas	Diferença	
Diferença	-20,3800	
95% CI	-9,1814	
Diferença = Tempo_da_Oti - Tempo_Literatura		
P_Valor	0,002	
T_Valor	-3,05	

Ao observar a tabela 7, percebe-se que as médias encontradas pelas amostras se diferem entre si. Dessa forma, pode-se assegurar a nível de 95% de confiança que o número de iteração obtido pela configuração da literatura é maior que o número de iteração obtido pelo método proposto, isto é, pode-se deduzir que número de passo para se encontrar a melhor solução se encontra no método proposto de modo mais eficiente e eficaz.

A tabela 8, apresenta os resultados encontrados pelo processo de otimização dos tempos e iterações calibrados conforme os parâmetros sugeridos pela literatura e pelo arranjo de experimental gerado através do planejamento de experimento (*DOE*).

<b>Tabela 8 - Comparação entre tempos e iterações na otimização da função esférica</b>									
<b>Comparações dos Tempos e Iterações</b>									
Tempo encontrado para a configuração da literatura					Tempo encontrado para a configuração do método				
<b>c1</b>	<b>c2</b>	<b>w</b>	<b>Tempo</b>	<b>Iteração</b>	<b>c1</b>	<b>c2</b>	<b>w</b>	<b>Tempo</b>	<b>Iteração</b>
2	2	0,9	28,101	26,8	2,3601	1,7392	0,4	25,111	6,42

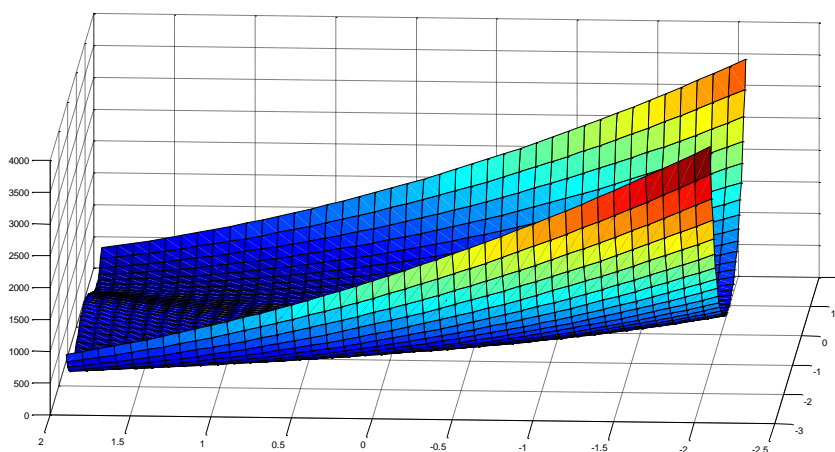
Na seção a seguir, serão apresentados os resultados obtidos pela otimização do algoritmo *PSO* usando as funções de testes Rosembrock e Rastrigin respectivamente.

### 4.3 Aplicação do método na função teste Rosembrock

Como visto na seção anterior, o método se fez eficiente para otimização da função de teste conhecida como esférica. Para esta seção, a função de teste escolhida dentro todas funções teste da literatura foi a Rosembrock. Esta função possui uma representação gráfica que tem como característica um grande vale no qual seu ponto de mínimo localizado é  $f(x) = 0$  em que  $x = (1, \dots, 1)$ .

A função Rosembrock é dada pela seguinte Equação (4.4) e demonstrada graficamente pela figura 14:

$$f(x_1, \dots, x_n) = \sum_{i=1}^{n-1} [(1 - x_i^2) + (100 * (x_{i+1} - x_i^2)^2] \quad (4.4)$$



**Figura 14** – Representação gráfica da função Rosenbrock

O algoritmo fará uma varredura na função através do espaço de busca, sendo as restrições de espaço de busca os seguintes limitantes  $x_i \in [-2,048; 2,048]$  de acordo com o trabalho de (VANDENBERGH e ENGELBRECHT, 2004). Após ajustar a função que será otimizada, deve-se então utilizar o arranjo experimental já gerado com os 26 experimentos para a otimização da mesma. Como visto na seção anterior, o algoritmo foi ajustado conforme o pesquisador declarou para a função anterior, sendo assim o algoritmo seguirá os mesmos procedimentos para esta função.

Por se tratar de outra função, os resultados extraídos para a variável resposta distância são demonstrados de acordo com a tabela 9 a seguir:

Tabela 9 (a) - Resposta distância extraída			
Parâmetros do algoritmo			Resposta
c1	c2	w	Distância
1,00	0,00	0,4	2,55
0,90	0,10	0,4	13,56
0,80	0,20	0,4	14,25
0,70	0,30	0,4	16,79



Sendo assim, a tabela 11 apresenta os resultados obtido por esta análise em relação a variável resposta.

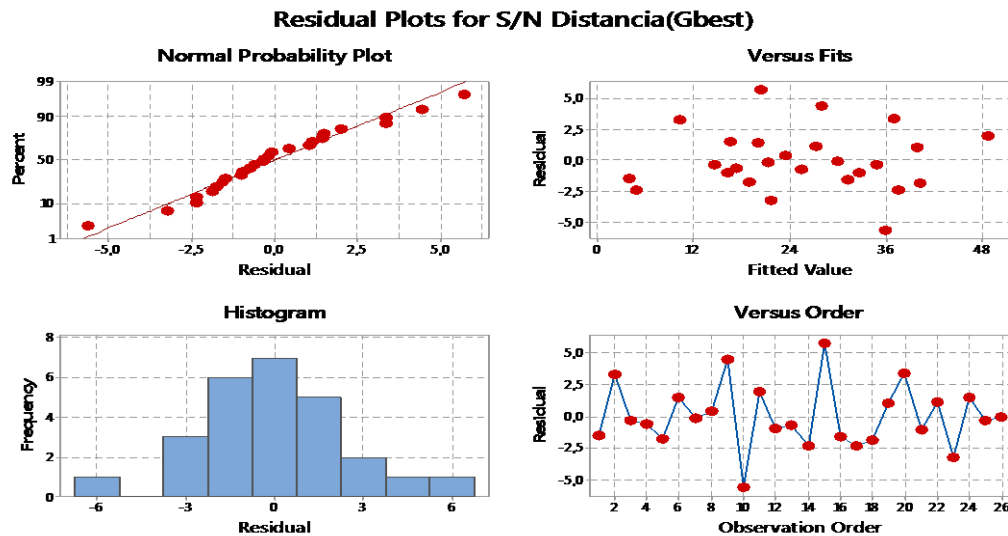
**Tabela 11** - Análise de variância para a resposta distância

Análise de Variância para a resposta distância						
Regressão	DF	Seq SS	Adj SS	Adj MS	F	P
	9	3154,25	3154,254	350,473	36,45	0,000
Componentes						
Linear	1	753,95	917,758	917,758	95,44	0,000
Quadrático	1	364,57	365,999	365,999	39,06	0,000
c1*c2	1	364,57	365,999	365,999	39,06	0,000
Cúbico completo	1	247,60	247,600	247,600	25,75	0,000
c1*c2*(-)	1	247,60	247,600	247,600	25,75	0,000
Quártico Completo	1	1,57	1,573	1,573	0,16	0,691
c1*c2*(-)2	1	1,57	1,573	1,573	0,16	0,691
Componente*w						
Nenhum componente	1	311,11	572,464	572,464	59,53	0,000
w	1	311,11	572,464	572,464	59,53	0,000
Linear	1	582,72	316,831	316,831	32,95	0,000
c1*w	1	582,72	316,831	316,831	32,95	0,000
Quadrático	1	888,99	884,534	884,538	91,98	0,000
c1*c2*w	1	888,99	884,534	884,538	91,98	0,000
Cúbico completo	1	2,83	2,828	2,828	0,29	0,595
c1*c2*(-)*w	1	2,83	2,828	2,828	0,29	0,595
Quártico Completo	1	0,91	0,909	0,909	0,09	0,762
c1*c2*(-)2*w	1	0,91	0,909	0,909	0,09	0,762
Erro residual	16	153,86	153,859	9,616		
Total	25	3308,11				

Ao observar a tabela 11, pode-se identificar que alguns termos apresentam “p\_valor” abaixo de 5%, isto significa que existe ajuste significativo para alguns modelo da variável resposta. A partir desta análise, um modelo quártico completo foi adequado para a resposta distância conforme apresentado na tabela 10, sendo assim este ajuste de modelo pode ser descrito pela seguinte equação (4.5):

$$\begin{aligned}
 \hat{y} = & 4,46c1 + 32,64c2 + 45,60c1c2 + 63,50c1c2(c1 - c2) - 10,50c1c2(c1 - c2)^2 \\
 & - 16,12w + 16,56c1w + 70,89c1c2w - 6,79c1c2(c1 - c2)w \\
 & + 7,98c1c2(c1 - c2)^2w
 \end{aligned}
 \quad (4.5)$$

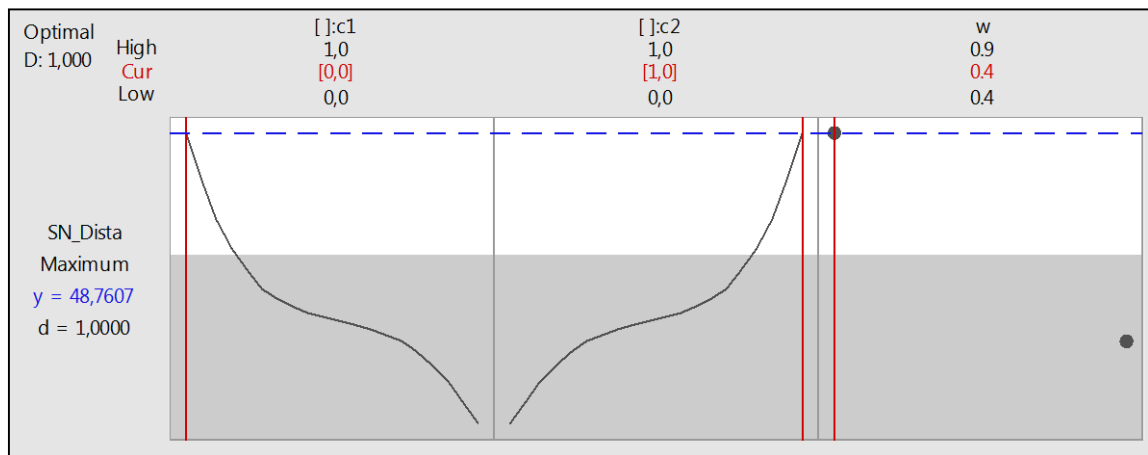
Depois de realizar a adequação do modelo, realizou-se uma análise de resíduos para a variável resposta com o intuito de investigar a distribuição dos dados. A figura 15 a seguir, apresenta a análise detalhada sobre os resíduos da resposta distância:



**Figura 15** - Análise dos resíduos para a resposta distância para a função *Rosenbrock*

Com pode ser visto, os dados foram distribuídos normalmente, apresentam assimetrias ou contém algum *outliers* e os resíduos não são correlacionados.

Após a estimação dos coeficientes, análise de regressão, análise de variância e análise dos resíduos, o próximo passo é a otimização da variável resposta conhecida como distância. Nesta etapa empregou-se para a otimização desta resposta o método *desirability*, que por meio desta técnica tornou-se possível encontrar novos valores para os parâmetros do algoritmo de otimização enxame por partícula (*PSO*), como demonstrado na figura 16.



**Figura 16** - Encontro do parâmetro e solução ótimo para a função *Rosenbrock*

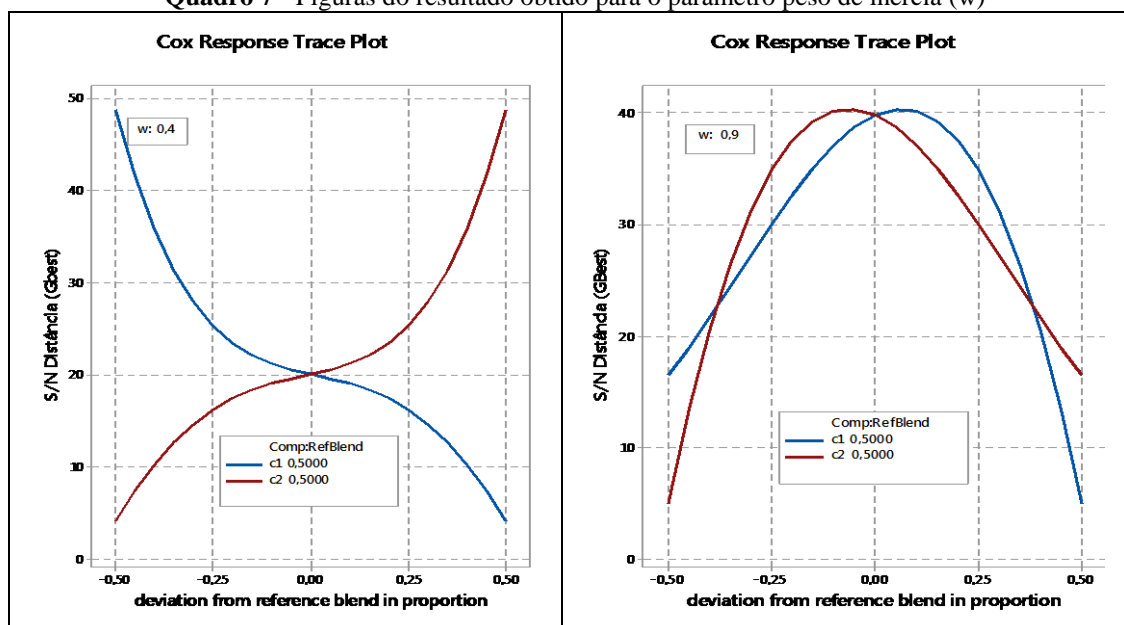
Onde *c1* e *c2* são os parâmetros do algoritmo que recebe 0 e 1 respectivamente, equivalendo às proporções em formas de porcentagens. Como visto na seção anterior, para esta dissertação foi utilizado um somatória dos parâmetros *c1* e *c2*, equivalendo um total de 4,1 conforme o adotado pelo trabalho de (MENDES; KENNEDY; NEVES, 2004), encontrando assim uma nova configuração ótima para os parâmetros do algoritmo, conforme ilustrado a tabela 12.

**Tabela 12** - Configuração ótima da otimização realizada na função Rosembrock

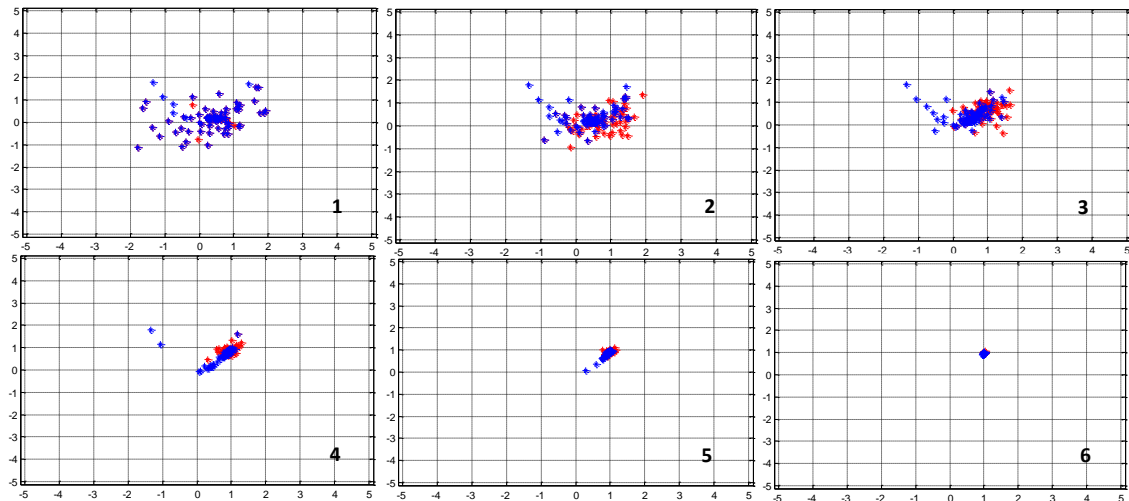
Otimização da Função Rosembrock		
c1	c2	w
0	4,1	0,4

Após encontrados os parâmetros ótimos desta função, deve-se então compará-los com a configuração que literatura propõe. De acordo com a seção anterior, os trabalhos de (Kennedy e Eberhart, 1995), trazem um ajuste padrão para este algoritmo sugerindo que os parâmetro de aprendizagem cognitivo (c1) e aprendizagem social (c2) sejam equivalentes a 2. No trabalho de (Shi e Eberhart, 1998) sugerem valores entre 0,9 até 1,2 para a peso da inércia (w). Visto que os valores diferem dos obtidos pela criação do arranjo experimental, torna-se um impasse dizer qual seria o melhor método apresentado. Por outro lado, pode-se afirmar que a comparação entre tempos das configurações obtidas tanto pelo arranjo experimental quanto aos sugeridos pela literatura, pode desencadear uma análise de eficiência e eficácia para efeito de conferência da qualidade de um algoritmo conforme (CORMEN *et al.*, 2009), além de ser útil na forma de avaliar a performance do algoritmo de otimização.

O quadro 7, reforça em forma de duas figuras o resultado obtido para esta função de teste que o parâmetro peso de inércia (w) demonstra ao passo que os valores do parâmetro alternando entre si, o resultado obtido para a resposta na maioria das vezes será melhor para o valor alcançado por esta otimização.

**Quadro 7** - Figuras do resultado obtido para o parâmetro peso de inércia (w)

A figura 17 a seguir, demonstra os passos que o algoritmo leva para localizar a solução para a função *Rosenbrock*.



**Figura 17** - Passos do processo de otimização para a função *Rosenbrock*

Com o intuito de confirmar o desempenho do algoritmo de otimização enxame de partículas, a resposta “tempo de execução” foi submetida a um teste de hipóteses com a finalidade de investigar se as médias destas respostas se assemelham entre si. Com base na tabela 13, a mesma apresenta um valor de *p\_value* abaixo de 5%, ou seja, pode-se afirmar o nível de 95% de confiança que o tempo de otimização dos parâmetros ajustados pela otimização do método proposto neste trabalho se faz eficiente e eficaz melhorando a performance do algoritmo de otimização. Visto que, o tempo de otimização dos parâmetros configurados pela literatura estão acima do tempo encontrado pelo método proposto por esta dissertação.

**Tabela 13** - Análise *Two Sample-t* dos tempos  
***Two Sample-T* para Tempo\_da\_Oti vs Tempo\_Literatura**

Amostras Individuais		
Estatísticas	Tempo_da _Oti	Tempo_Literatura
N	50	50
Média	25,252	26,877
Desvio Padrão	0,45903	1,2459
Diferença entres as amostras		
Estatísticas	Diferença	
Diferença	-1,6254	
95% CI	-1,3119	
Diferença = Tempo_da _Oti - Tempo_Literatura		
P_Valor	0,001	
T Valor	-8,66	

Além de usar o tempo, existe outra maneira para se avaliar a performance de um algoritmo. Este método é denominado método iterativo, ou seja, esta técnica tem o intuito de resolver problemas complexos de otimização por meio de algoritmos que trocam suas piores soluções por soluções melhores a fim de aproximar do ponto de ótimo de um função. Para (SAAD, 2003), este método tem como principal propósito alterar soluções ao passo que todos os componentes sejam substituídos por novos e melhores encontrando suas soluções ótimas.

Desta forma, os resultados obtidos das iterações para cada configuração do arranjo experimental executados nas mesmas condições, se encontram indexados em (ANEXO C), tabela C.1 e C.2, respectivamente.

Logo que coletadas todas as iterações inclusive ao do parâmetro de otimização, deve-se então aplicar um teste de hipóteses com intuito de verificar se as amostras se assemelham entre si. Sendo assim, a tabela 14 a seguir apresenta resultados obtidos pela análise estatística usando o teste de hipótese *two sample-t* para investigar se as médias da iteração obtida pela configuração do método proposto é menor que a média obtida pela configuração sugerida pela literatura.

Tabela 14 - Análise <i>Two Sample-t</i> para as iterações		
Two Sample-T para Iteração_da_Oti vs Iteração_Literatura		
Amostras Individuais		
Estatísticas	Iteração_da_Oti	Iteração_Literatura
N	50	50
Média	95,6	123,2
Desvio Padrão	9,5640	53,224
Diferença entres as amostras		
Estatísticas	Diferença	
Diferença	-27,6	
95% CI	-14,793	
Diferença = Tempo_da_Oti - Tempo_Literatura		
P_Vvalor	0,001	
T_Vvalor	-3,61	

Ao observar a tabela 14, nota-se que o valor de *p\_value* é inferior ao 5%, isto significa a um nível de 95% de confiança que as média do número de iteração gerados pelos parâmetros encontrados pela otimização da respostas distância em refutação aos parâmetros sugeridos pela literatura são baixas, portanto pode-se confirmar que existe uma melhora em relação à eficiência e eficácia no algoritmo *PSO*.

Com isto, a tabela 15 mostra os tempos de execução e as iterações obtidas dos parâmetros ótimos obtidos pela otimização da resposta e comparados com os valores sugeridos pela literatura:

<b>Tabela 15 - Comparação dos tempos entre a configuração padrão x método atual</b>									
<b>Comparações dos Tempos e Iterações</b>									
Tempo encontrado para a configuração da literatura					Tempo encontrado para a configuração do método				
<b>c1</b>	<b>c2</b>	<b>w</b>	<b>Tempo</b>	<b>Iteração</b>	<b>c1</b>	<b>c2</b>	<b>w</b>	<b>Tempo</b>	<b>Iteração</b>
2	2	0,9	26,877	123,2	0	4,1	0,4	25,252	95,7

Ao observar a tabela, pode-se identificar que os parâmetros usados pela literatura possuem um tempo e iteração maior em relação aos parâmetros obtidos pela otimização, assegurando que o método proposto tem suas vantagens em relação à qualidade do algoritmo



padrão. Portanto, fica justificado que o método proposto se fez eficiente e eficaz quanto ao tempo e iteração de otimização e na busca pelo ponto de ótimo da função Rosembrock.

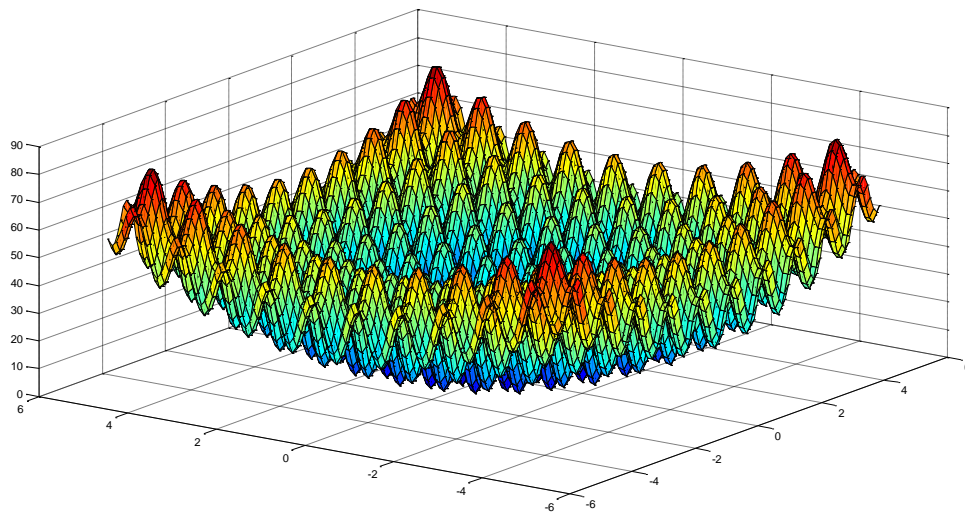
Na seção a seguir, serão apresentados os resultados obtidos pela otimização do algoritmo enxame de partícula usando as funções teste Rastrigin.

#### 4.4 Aplicação do método na função teste Rastrigin

Como visto na seção anterior, o método se fez também eficiente para otimização da função teste Rosembrock. Para esta seção, será apresentada a última função teste escolhida para representar a otimização da performance do algoritmo PSO. Esta função apresenta uma superfície com vários ótimos locais, dificultando a varredura e busca pelo ótimo global. Sua principal característica é de grandes vales e picos no qual seu ponto de mínimo localizado é  $f(x) = 0$  em que  $x = (0, \dots, 0)$ .

A função Rastrigin pode ser formulada conforme a Equação (4.6) e demonstrada graficamente pela figura 14:

$$f(x_1, \dots, x_n) = \sum_{i=1}^n [(x_i^2 - 10\cos(2\pi x_i)) + 10] \quad (4.6)$$



**Figura 18** – Representação gráfica da função Rastrigin

O algoritmo fará uma varredura na função através do espaço de busca, sendo as restrições de espaço de busca os seguintes limitantes  $x_i \in [-5,12 \ 5,12]$  conforme apresentado em (VANDENBERGH e ENGELBRECHT, 2004). Após ajustar a função que será otimizada, deve-se então utilizar o arranjo experimental já gerado com os 26 experimentos para a otimização da mesma.

Por se tratar da função Rastrigin, os resultados extraídos para a variável resposta distância foram outros. Sendo assim, os resultados obtidos das distâncias para cada configuração do arranjo experimental serão apresentados conforme a tabela 16:

Tabela 16 - Resposta distância extraída			
Parâmetros do algoritmo			Resposta
c1	c2	w	Distância
1,00	0,00	0,4	2,88
0,90	0,10	0,4	3,59
0,80	0,20	0,4	38,20
0,70	0,30	0,4	12,26
0,60	0,40	0,4	8,58
0,50	0,50	0,4	4,25
0,40	0,60	0,4	4,02
0,30	0,70	0,4	1,35
0,20	0,80	0,4	1,47
0,10	0,90	0,4	2,41
0,00	1,00	0,4	2,88
0,75	0,25	0,4	14,02
0,25	0,75	0,4	2,41
1,00	0,00	0,9	2,86
0,90	0,10	0,9	13,85
0,80	0,20	0,9	10,03
0,70	0,30	0,9	17,00
0,60	0,40	0,9	17,03
0,50	0,50	0,9	16,98
0,40	0,60	0,9	11,01
0,30	0,70	0,9	13,87
0,20	0,80	0,9	7,44
0,10	0,90	0,9	4,06
0,00	1,00	0,9	2,86
0,75	0,25	0,9	17,00
0,25	0,75	0,9	8,04

Como realizado na seção anterior, foram feitas as análises estatísticas em relação à resposta distância através do *DOE* pela análise de regressão, onde foram estimados os coeficientes do modelo para verificação da qualidade e explicação da coleta de dados, de acordo com a tabela 17:

Tabela 17- Determinação dos coeficientes do modelo da distância					
Estimação dos coeficientes por meio de regressão da resposta distância					
Termos	Coef	SE Coef	T	P	VIF
c1	1,94	4,169	*	*	4,061
c2	2,46	4,169	*	*	4,061
c1*c2	33,44	14,754	2,27	0,038	4,587
c1*c2*(-)	64,48	24,979	2,58	0,020	2,120
c1*c2*(-) <sup>2</sup>	58,09	51,841	1,12	0,279	2,821
w	0,07	4,169	0,02	0,987	11,796
c1*w	1,85	5,758	0,32	0,753	7,746
c1*c2*w	16,89	14,754	1,15	0,269	4,587
c1*c2*(-)*w	-32,62	24,979	-1,31	0,210	2,120
c1*c2*(-) <sup>2</sup> *w	-61,61	51,841	-1,19	0,252	2,821
S = 6,18952			PRESS = 2416,79		
R-sq=62,90%		R-sq(press)=0,00%		R-sq(adj)=42,04%	

Aplicando o conhecimento já utilizado nas funções anteriores, nota-se que o  $R-Sq(adj)$  está abaixo de 80%, ou seja, os resultados obtidos para validar o modelo não foram significativos. Desse forma foi aplicado outra abordagem nesta etapa, onde resultou na remoção de alguns experimentos que não condiziam com a explicação do modelo, tornando possível a melhorar a explicação do modelo. A tabela 18 a seguir, apresenta os resultados obtidos da resposta distância, onde foi aplicado a remoção de algumas distâncias.

**Tabela 18** - Remoção dos *outliers* das variáveis respostas

Parâmetros do algoritmo			Resposta
c1	c2	w	Distância
1,00	0,00	0,4	2,88
0,90	0,10	0,4	3,59
0,80	0,20	0,4	*
0,70	0,30	0,4	*
0,60	0,40	0,4	8,58
0,50	0,50	0,4	4,25
0,40	0,60	0,4	4,02
0,30	0,70	0,4	1,35
0,20	0,80	0,4	1,47
0,10	0,90	0,4	2,41
0,00	1,00	0,4	2,88
0,75	0,25	0,4	*
0,25	0,75	0,4	2,41
1,00	0,00	0,9	2,86
0,90	0,10	0,9	13,85
0,80	0,20	0,9	*
0,70	0,30	0,9	17,00
0,60	0,40	0,9	17,03
0,50	0,50	0,9	16,98
0,40	0,60	0,9	11,01
0,30	0,70	0,9	*
0,20	0,80	0,9	7,44
0,10	0,90	0,9	4,06
0,00	1,00	0,9	2,86
0,75	0,25	0,9	17,00
0,25	0,75	0,9	8,04

Logo após remover os experimentos menos significativos, foi verificado novamente o ajuste do modelo ( $R-Sq(adj)$ ) para então prosseguir com a otimização desta função. A tabela 19 a seguir, demonstra o resultado obtido após a remoção dos termos menos significativos da resposta distância a fim de ajustar o modelo.

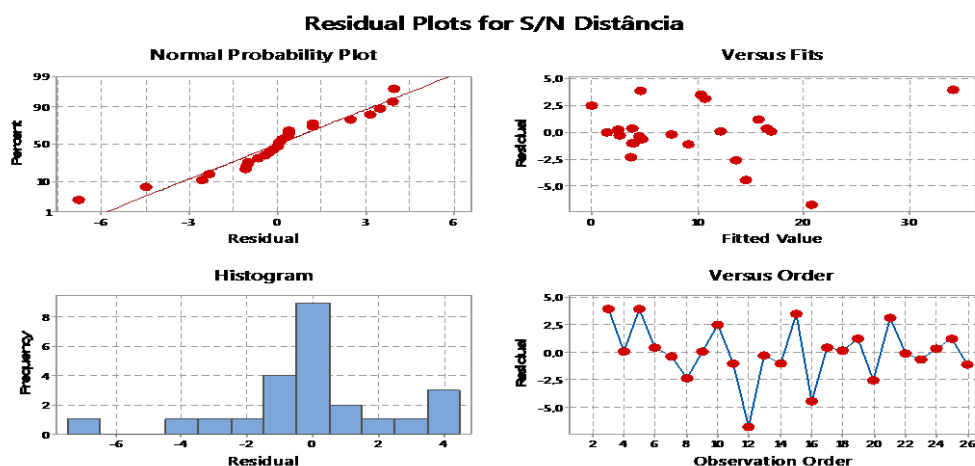
**Tabela 19 (a)** - Determinação dos coeficientes do modelo após remoção dos *outliers*

Estimação dos coeficientes por meio de regressão da resposta distância					
Termos	Coef	SE Coef	T	P	VIF
c1	84,6	16,847	*	*	195,139
c2	3,2	2,194	*	*	4,146
c1*c2	-136,5	34,598	-3,94	0,001	92,116
c1*c2*(-)	-151,0	46,855	-3,22	0,006	26,429
c1*c2*(-) <sup>2</sup>	-132,9	54,691	-2,43	0,029	10,628
w	-0,7	2,194	-0,32	0,755	11,129
c1*w	-80,1	16,690	-4,80	0,000	191,525
c1*c2*w	186,8	34,598	5,40	0,000	92,116



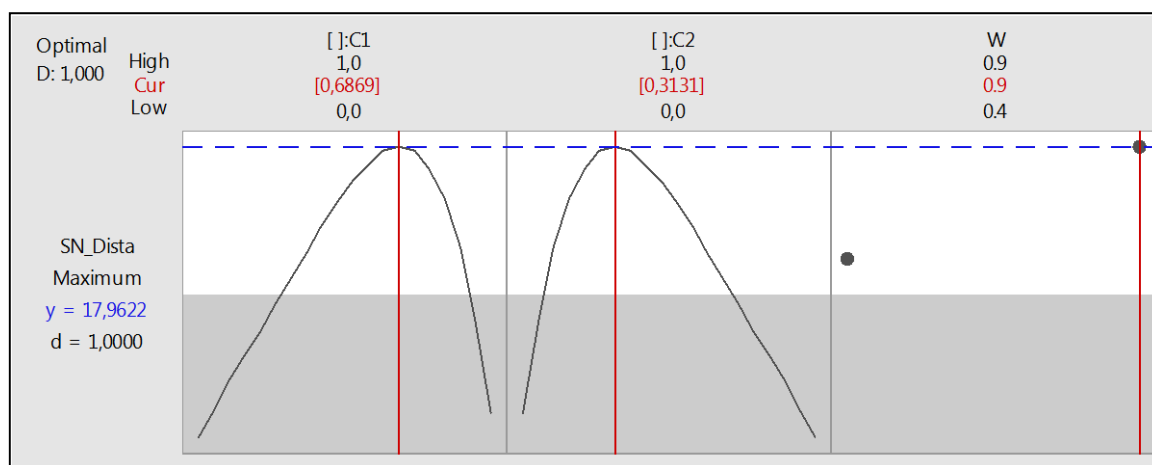
$$\begin{aligned}\hat{y} = & 84,6c_1 + 3,2c_2 - 136,5c_1c_2 - 151,0c_1c_2(c_1 - c_2) - 132,9c_1c_2(c_1 - c_2)^2 \\ & - 0,7w - 80c_1w + 186,8c_1c_2w + 182,9c_1c_2(c_1 - c_2)w \\ & + 129,4c_1c_2(c_1 - c_2)^2w\end{aligned}\quad (4.7)$$

Após realizar a adequação do modelo, realizou-se uma análise de resíduos para a variável resposta com o intuito de investigar a distribuição dos dados. A figura 19 a seguir, apresenta a análise detalhada sobre os resíduos da resposta distância:



**Figura 19** - Análise dos resíduos para a resposta distância para a função *Rastrigin*

Em seguida, aplicou-se a otimização para a resposta que possui um novo ajuste significativo de modelo. Para esta otimização, empregou-se também o método *desirability*, onde encontrou-se para otimização desta função os seguintes resultados:



**Figura 20** - Encontro do parâmetro e solução ótimo para função Rastrigin

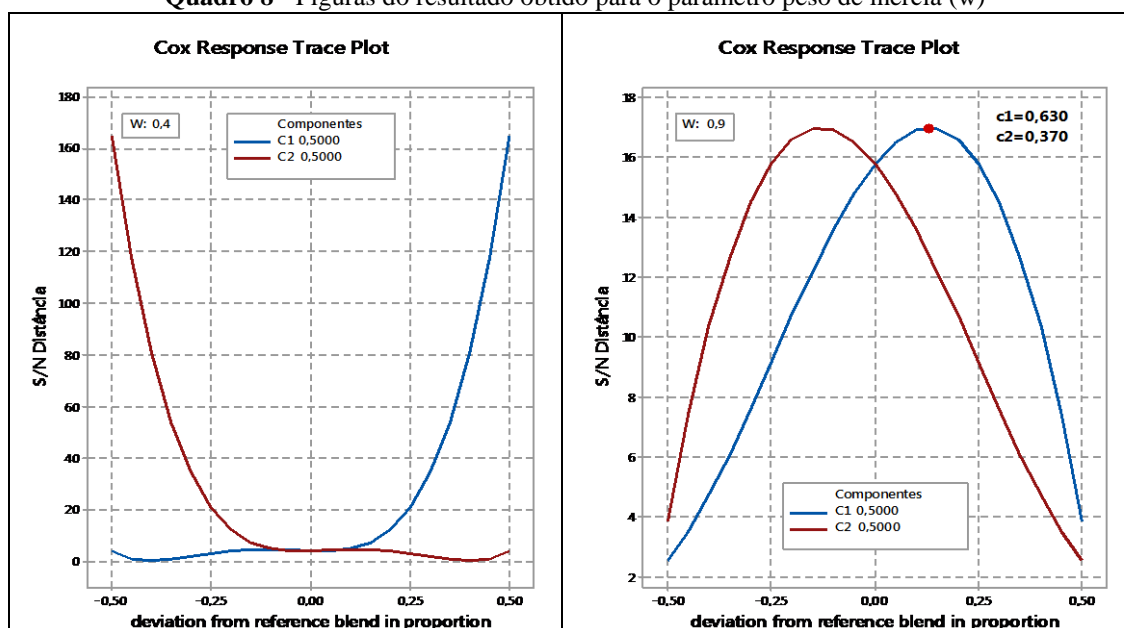
Onde  $c_1$  e  $c_2$  são os parâmetros do algoritmo que recebe 0,6869 e 0,3131 respectivamente, equivalendo às proporções em formas de porcentagens. Como visto na seção anterior, para esta dissertação foi utilizado um somatório dos parâmetros  $c_1$  e  $c_2$  conforme o adotado pelo trabalho de (Mendes, Kennedy e Neves, 2004), encontrando assim a configuração ótima para o algoritmo conforme ilustrado a tabela 20.

**Tabela 20** - Configuração ótima da otimização realizada na função Rastrigin

Otimização da Função Rastrigin		
c1	c2	w
2,81629	1,28371	0,4

Após encontrados os parâmetros ótimos desta função, deve-se então compará-los com a configuração que literatura propõem. De acordo com a seção anterior, os trabalhos de Kennedy e Eberhart (1995) e Shi e Eberhart (1998), trazem um ajuste padrão para este algoritmo. Para Cormen *et al.*, (2009) a forma mais adequada para avaliar se um algoritmo se faz eficiente e eficaz se deve a um fator imprescindível que é o tempo de execução. Este tempo tem por objetivo contabilizar um período constante em que o tempo exige para executar cada linha de código (CORMEN *et al.*, 2002).

O quadro 8, reforça em forma de duas figuras o resultado obtido para esta função de teste que o parâmetro peso de inércia (w) demonstra ao passo que os valores do parâmetro alternando entre si, o resultado obtido para a resposta na maioria das vezes será melhor para o valor alcançado por esta otimização.

**Quadro 8** - Figuras do resultado obtido para o parâmetro peso de inércia (w)

Com o intuito de averiguar a diferença existente entre os tempos de execução do algoritmo em relação aos parâmetros obtidos pelo arranjo de misturas e o padrão apresentados pela literatura, o mesmo foi submetido a um teste de hipóteses conhecido com *two sample-t*, onde o técnica tem por finalidade comparar as médias existente em ambas amostra, vale ressaltar que os valores para os tempos se encontram indexados no ANEXO (D), na tabela D.1. Com propósito de evidenciar a nível de confiança a diferença entre as médias dos tempos obtidos pelo arranjo experimental, em refutação com as médias dos tempos obtidos pelo método

de otimização aplicado, a Equação (4.8) representa em forma de comparação tal teste de hipóteses:

$$\begin{cases} H_0 : \mu_{otimização} = \mu_{Literatura} \\ H_1 : \mu_{otimização} < \mu_{Literatura} \end{cases} \quad (4.8)$$

Com base na Equação (4.8), pode-se realizar a comparação entre os tempos obtidos pela configuração encontrada da otimização da resposta distância em relação aos valores padrões sugeridos pela literatura. Dessa forma, a tabela 21 a seguir apresenta os resultados apontados pelo teste de hipóteses aplicado aos tempos.

Tabela 21 - Análise <i>Two Sample-t</i> para os tempos		
Two Sample-T para Tempo_da_Oti vs Tempo_Literatura		
	Amostras Individuais	
Estatísticas	Tempo_da_Oti	Tempo_Literatura
N	50	50
Média	25,040	25,714
Desvio Padrão	0,18015	0,22158
Estatísticas	Diferença	
Diferença	-0,67434	
90% CI	(-0,74143 ; -0,60725)	
Diferença = Tempo_da_Oti - Tempo_Literatura		
P_Valor	< 0,001	
T_Valor	-28,45	

Pode-se observar que a tabela 21 mostra que o tempo de execução entre os parâmetros de configuração do algoritmo *PSO* possui médias diferentes, isto é, o *p-value* encontrado está abaixo de 5%, tornando a hipótese nula rejeitada, ou seja, aceitando a hipótese alternativa fazendo com que haja melhora significativa da qualidade do algoritmo

Diferente do tempo, existe outra abordagem para analisar a performance do algoritmo de otimização. Esta abordagem é conhecida como método iterativo, isto é, este método tem por finalidade executar a cada passo uma comparação entre as soluções, armazenando as melhores e descartando as piores a fim de se aproximar da solução final da função em que se pretende otimizar (SAAD, 2003).

A tabela 26 a seguir, contém os resultados obtidos pela análise estatística do teste de hipóteses *two sample-t*, onde se pretende observar as diferenças entre as configurações encontradas pela otimização em confronto a configuração sugerida pela literatura. Para conferência dos dados extraídos da iteração vide ANEXO (D), tabela D.2.

Tabela 22 - Análise Two Sample-t para as iterações		
Two Sample-T para Iteração_da_Oti vs Iteração_Literatura		
Amostras Individuais		
Estatísticas	Iteração_da_Oti	Iteração_Literatura
N	50	50
Média	98,68	107,52
Desvio Padrão	9,5640	53,224
Diferença entres as amostras		
Estatísticas	Diferença	
Diferença	-8,84	
90% CI	(-28,667 ; 10,987)	
Diferença = Tempo_da_Oti - Tempo_Literatura		
P_Valor	<0,230	
T_Valor	-0,74	

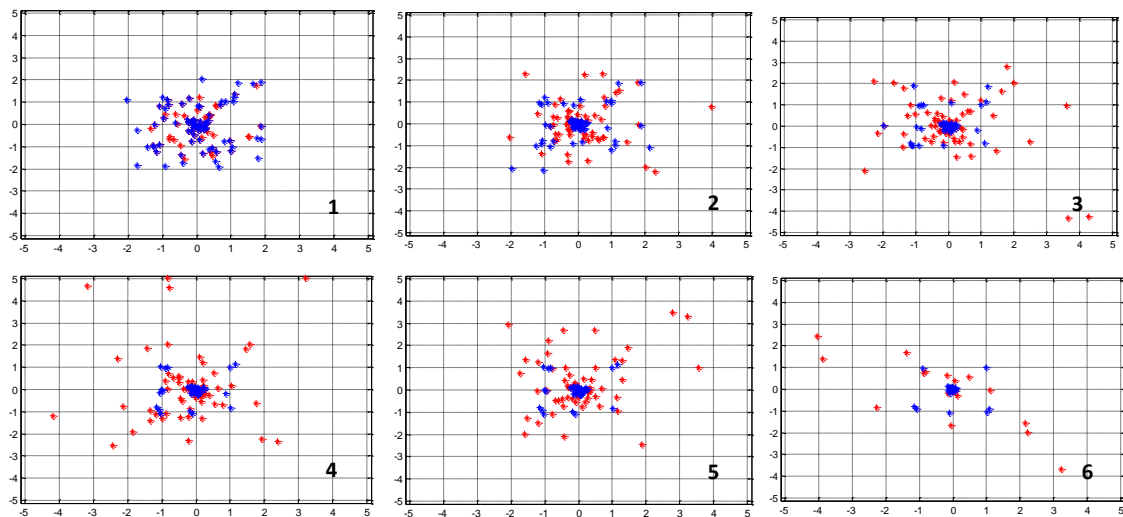
Como apresentado na tabela 22, pode-se observar que o valor de *p\_value* é superior ao 5%, ou seja, pode-se então afirmar a um nível de 95% de confiança que não há evidências o suficiente que a média da iteração obtida pela configuração da otimização da resposta distância seja menor que o número de iteração encontrada pela configuração da literatura.

Para isto, a tabela 23 a seguir mostra os dois tempos e iterações obtidas das configurações que se pretende confrontá-las para enfim definir qual será a configuração que se destaca na performance do algoritmo de otimização:

<b>Tabela 23 - Comparação dos tempos entre a configuração padrão x método atual</b>									
<b>Comparações dos Tempos e Iterações</b>									
Tempo encontrado para a configuração da literatura					Tempo encontrado para a configuração do método				
<b>c1</b>	<b>c2</b>	<b>w</b>	<b>Tempo</b>	<b>Iteração</b>	<b>c1</b>	<b>c2</b>	<b>w</b>	<b>Tempo</b>	<b>Iteração</b>
2	2	0,9	25,714	107,5	2,81629	1,28371	0,9	25,040	98,7

. Para complementar a tabela 23, a figura 21 a seguir representa graficamente os passo e tempo que algoritmo encaminhou para localizar a solução da função *Rastrigin*.





**Figura 21** - Passos do processo de otimização para a função *Rastrigin*

Ao contrapor os valores obtidos de ambas configurações, denota-se que os tempos e iterações encontrados para esta função foram diferentes entre si. Sendo assim, a fim de evidenciar uma análise mais adequada dos dados o mesmo foi submetido a um teste de hipóteses com o intuito de validar uma paridade de ambas respostas encontradas. Dessa forma, conclui-se que o método se fez eficiente em relação ao tempo pelo busca da solução da função em que se pretende otimizar, porém não eficaz em relação ao número de passos que o mesmo desloca a fim de encontrar a solução final desta função.

## 4.5 Considerações Finais

Este capítulo apresentou a aplicação do método de otimização proposto, otimização irrestrita mono-objetivo por enxame de partículas assistidas por polinômios canônicos de misturas em funções teste reconhecidas pela literatura. Esta proposta foi aplicada a três funções presentes na literatura: Esférica, Rosembrock e Rastigin.

Ao aplicar o método proposto nestas funções, o mesmo promoveu a redução do tempo de processamento e aumentou a qualidade final da performance do otimizador quanto ao quesito de passos, além de fornecer um grande questionamento quanto aos parâmetros destacados na literatura, uma vez que o algoritmo padrão segue uma abordagem simples em relação à forma de ajuste.

Dessa forma, este capítulo encerra o passo de análise dos resultados e aplicação do método como ferramenta de melhoria em busca de parâmetros ótimos a algoritmos de otimização.

## 5 CONCLUSÃO

Na busca incessante por melhoria na qualidade de processo e/ou sistemas, os algoritmos de otimização vêm sendo utilizados como ferramentas capazes de auxiliar gestores no controle das variáveis que influenciam diretamente ou indiretamente determinados sistemas. Tendo em vista este desfecho, tem-se então um pequeno impasse para que todo o processo obtenha realmente sua qualidade total em relação ao produto que se entrega ao mercado. Este problema se encontra especificamente nos algoritmos de otimização, pois estes otimizadores trazem soluções que, na maioria das vezes, não são triviais. Portanto há uma necessidade de aplicar outras abordagens como técnicas matemáticas e estatísticas capazes de auxiliar com êxito na qualidade final de determinados algoritmo de otimização.

Este trabalho sugeriu um procedimento experimental capaz de otimizar os parâmetros do algoritmo PSO, através da utilização de um arranjo de misturas combinadas com variáveis de processo gerado pela técnica de planejamento de experimento (*DOE*). O método proposto foi aplicado na otimização de função de testes reconhecidas na literatura. E seu principal objetivo é avaliar a influência que os parâmetros do otimizador realizam nas respostas que foram obtidas.

Em suma, a aplicação do planejamento de experimento combinado com outras técnicas matemáticas, no algoritmo de otimização enxame de partículas (*PSO*), permitiu as seguintes conclusões:

- Através de diversas técnicas combinadas, pode-se encontrar novo valores para o algoritmo *PSO*, a cada função de teste como pode ser observado nas tabelas 8,15 e 23;
- Como pode ser visto, as figuras 13,17 e 20 apresentam uma população de soluções em busca do ótimo global para suas funções de teste, denota-se que para encontrar essas soluções as mesmas não exibem dispersões muito grande no espaço de busca.
- Pode-se perceber ao analisar as figuras dos quadros 6,7 e 8 que a resposta distância se modifica ao passo que alternam os valores dos parâmetros aprendizagem cognitiva ( $c_1$ ), aprendizagem social ( $c_2$ ) e peso da inércia ( $w$ ), melhorando ou piorando o resultado desta resposta;
- Uma das principais diferenças obtidas nestes resultados foram os tempos de execução e o números de passos encontrados para cada função de teste, uma vez

que, aplicado a otimização na resposta distância S/N de cada função teste, foi descoberto que os tempos e iteração obtidas pelos parâmetros da otimização se mostraram mais eficiente e eficazes quanto aos tempos e iteração obtidos pelos parâmetros configurados conforme a literatura;

## 5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Diante dos desafios e oportunidades encontradas em todas as etapas desta pesquisa, é possível identificar como possibilidades de trabalhos futuros:

- Aplicar o método proposto a diferentes problemas de otimização mono-objetiva sem restrição para verificação da funcionalidade do método e da existência de outros parâmetros para este algoritmo.
- Aplicar o método proposto em problemas de casos reais que possuem otimização mono-objetiva sem restrição para verificação da funcionalidade do método e da existência de outros parâmetros para este algoritmo.
- Construir um algoritmo que solucione problemas de otimização multiobjetivo e adaptar o método proposto a tal problema de otimização com suas restrições de espaço de busca e restrições às funções de caso real.
- Realizar o desenvolvimento do algoritmo em outra linguagem a fim de observar melhor o retrospecto do otimizador.

## REFERÊNCIAS

- ACHAYUTHAKAN, C. AND ONGSAKUL, W. (2009). TVAC-PSO Based Optimal Reactive Power Dispatch for Reactive Power Cost Allocation under Deregulated Environment. In: 2009 **IEEE Power & Energy Society General Meeting**. Calgary: IEEE, pp.1-9.
- ADAMI, C. Introduction to artificial life. 1. ed. New York, NY: **Springer**, 1999. p. 1-376.
- AHUJA, R.; MAGNANTI, T.; ORLIN, J. NETWORK FLOWS: Theory, Algorithms, and Applications. 1. ed. Englewood Cliffs N.J.: **Prentice Hall**, 1993. p. 1-864.
- AKAY, B.; KARABOGA, D. A modified Artificial Bee Colony algorithm for real-parameter optimization. **Information Sciences**, v. 192, n. 0020-0255, p. 120-142, 2012.
- ALBRECHT, C. **Algoritmos evolutivos aplicados a síntese e otimização de sistemas de ancoragem**. Rio de Janeiro, Rio de Janeiro, Brazil: Tese de Doutorado – Programa de Pós Graduação de Engenharia em Ciências Oceânica - Universidade Federal do Rio de Janeiro, 2005.
- ALMEIDA, H. S.; TOLEDO, J. C. Método Taguchi: qualidade voltada para o projeto do produto e do processo. **Revista de Administração**, v. 24, n. 4, p. 62-68, 1989.
- ANDERY, M. et al. Para compreender a ciência: Uma perspectiva histórica. 1. ed. Rio de Janeiro (RJ): Garamond: **São Paulo: EDUC**, 2004, p. 1-436.
- APPOLINÁRIO, F. Metodologia da Ciência: Filosofia e Prática da Pesquisa. 2. ed. São Paulo: **Cengage Learning**, 2012. p. 1-240
- BÄCK, THOMAS; HAMMEL, ULRICH; SCHWEFEL, HANS-PAUL. Evolutionary computation: Comments on the history and current state. **IEEE Transactions on Evolutionary Computation**, 1997. p. 3-17.
- BALLARD, D. An introduction to natural computation. 1. ed. Cambridge, Mass: **MIT Press**, 2000.
- BASHIRI, M.; SHIRI, M.; BAKHTIARIFAR, M. A Robust Desirability-based Approach to Optimizing Multiple Correlated Responses. **International Journal of Industrial Engineering & Production Research**, v. 26, n. 2018, Volume 29, Number 2, p. 119-128, 2015.
- BASTOS FILHO, C. et al. A novel search algorithm based on fish school behavior. 2008 **IEEE International Conference on Systems, Man and Cybernetics**, n. 1062-922X, 2008.
- BERTO, R.; NAKANO, D. A produção científica nos anais do encontro nacional de engenharia de produção: um levantamento de métodos e tipos de pesquisa. **Production**, v. 9, n. 2, p. 65-75, 1999.
- BERTRAND, J.; FRANSOO, J. Operations management research methodologies using quantitative modeling. **International Journal of Operations & Production Management**, v. 22, n. 2, p. 241-264, 2002.
- BILGA, P.; SINGH, S.; KUMAR, R. Optimization of energy consumption response parameters for turning operation using Taguchi method. **Journal of Cleaner Production**, v. 137, p. 1406-1417, 2016.
- BOLTON, C. AND PARADA, V. Automatic design of algorithms for optimization problems. **IEEE**, pp.1-5, 2015.
- BRYMAN, A. Research Methods and Organization Studies. 1. ed. London: **Routledge**, 1989. p. 1-300.
- CASTRO, LEANDRO NUNES DE. Fundamentals of natural computing: an overview. **Physics of Life Reviews**, 2007, p. 1-36.
- CHEN, Y. et al. Particle swarm optimizer with crossover operation. **Engineering Applications of Artificial Intelligence**, v. 70, n. 0952-1976, p. 159-169, 2018b.
- CLERC, M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. **Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)**, v. 3, n. 0-7803-5536-9, p. 1951-1957.
- CLERC, M.; KENNEDY, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 1, p. 58-73, 2002.
- CORMEN, T. et al. Algoritmos: Tradução da 2ª edição Americana. 2. ed. Rio de Janeiro: **Campus**, 2002. p. 1-915.
- CORMEN, T. et al. Introduction to algorithms. 3ed. Cambridge (Inglaterra): **Mit Press**, 2009. p. 1-1292.

- CORNE, D.; DORIGO, M.; GLOVER, F. New ideas in optimization. 1ed. Maidenhead: **McGraw-Hill**, 1999. p. 1-450.
- CORNELL, J. A primer on experiments with mixtures. 33. ed. Hoboken, N.J.: **Wiley**, 2011. p. 1-368.
- COUTO, D.; BARSANTE, L.; SILVA, C. Otimização de funções multimodais via técnica de inteligência computacional baseada em Colônia de Vaga-lumes. **Proceedings of the XXXVI Iberian Latin American Congress on Computational Methods in Engineering**, 2015.
- DA SILVA, H.; DA SILVA, M. Aplicação de um projeto de experimentos (DOE) na soldagem de tubos de zircaloy-4. Disponível em: <<http://www.fmepro.org/ojs/index.php/rpe/article/view/31>>. Acesso em: 3 jan. 2018.
- DE CASTRO, L. Fundamentals of Natural Computing. 1. ed. Hoboken: **CRC Press**, 2006. p. 1-696.
- DE SILVA, A. Molecular logic-based computation. 1. ed. Cambridge: **RSC Pub.**, 2013. p. 1-416.
- DEL CASTILLO, E. Process optimization. 1. ed. New York, NY: **Springer**, 2007. p. 1-459.
- DERRINGER, G. A balancing act, Optimizing a products properties. *Quality Progress* 27, 51-57, 1994.
- DERRINGER, G.; SUICH, R. Simultaneous Optimization of Several Response Variables. **Journal of Quality Technology**, v. 12, n. 4, p. 214-219, 1980.
- DORIGO, M. et al. Swarm Intelligence. 1. ed. Berlin, Heidelberg: **Springer-Verlag Berlin Heidelberg**, 2010. p. 1-582.
- DORIGO, M.; DI CARO, G. Ant colony optimization: a new meta-heuristic. **Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)**, v. 2, n. 0-7803-5536-9, p. 1470-1477, 1999.
- EIBEN, A. AND SMIT, S. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. **Swarm and Evolutionary Computation**, pp.19-31.
- EIBEN, A., HINTERDING, R. AND MICHALEWICZ, Z. (1999). Parameter Control in Evolutionary Algorithms. **Studies in Computational Intelligence**, pp.124-141.
- ENGELBRECHT, A. Computational intelligence. 2. ed. Chichester, England: **John Wiley & Sons**, 2007. p. 1-628.
- ESMIN, A. **Estudo de Aplicação do Algoritmo de Otimização por Enxame de Partícula na Resolução de Problemas de Otimização Ligados ao SEP**. Itajubá, Minas Gerais, Brazil: Tese de Doutorado - Programa de Pós-Graduação em Engenharia Elétrica -. Universidade Federal de Itajubá, 2005.
- FENG, Y., TENG, G., WANG, A. AND YAO, Y. (2007). Chaotic Inertia Weight in Particle Swarm Optimization. **In: Innovative Computing, Information and Control, 2007. ICICIC '07. Second International Conference on. Kumamoto: IEEE**, pp.471-475.
- FILIPPINI, R. Operations management research: some reflections on evolution, models and empirical studies in OM. **International Journal of Operations & Production Management**, v. 17, n. 7, p. 655-670, 1997.
- GELENBE, EROL. Natural Computation. **The Computer Journal**, 2012, p. 848-851.
- GHANI, J.; CHOUDHURY, I.; HASSAN, H. Application of Taguchi method in the optimization of end milling parameters. **Journal of Materials Processing Technology**, v. 145, n. 1, p. 84-92, 2004.
- GHEORGHE, M. Molecular computation models. 1. ed. Hershey, PA: **Idea Group Pub.**, 2005. p. 1-303.
- GIL, A. Métodos e técnicas de pesquisa social. 6. ed. São Paulo: **Atlas**, 2008. p. 1-216.
- GOMES, J. **Análise e otimização da soldagem de revestimento de chapas de aço abnt 1020 com utilização de arame tubular inoxidável austenítico**. Itajubá, Minas Gerais, Brazil: Dissertação de Mestrado - Programa de Pós Graduação em Engenharia de Produção, 2010.
- GOMES, J. **Método dos polinômios canônicos de misturas para otimização multi-objetivo**. Itajubá, Minas Gerais, Brazil: Tese de Doutorado - Programa de Pós Graduação em Engenharia de Produção, 2013.
- GUO, L. AND CHEN, X. (2009). A Novel Particle Swarm Optimization Based on the Self-Adaptation Strategy of Acceleration Coefficients \*. **In: Computational Intelligence and Security, 2009. CIS '09. International Conference on. Beijing, China: IEEE**, pp.277-281.
- HARRINGTON, E.C.JR. The desirability function. *Industrial Quality Control* 21, 494-498, 1965.
- HASHEMI, A.; MEYBODI, M. A note on the learning automata based algorithms for adaptive parameter selection in PSO. **Applied Soft Computing**, v. 11, n. 1, p. 689-705, 2011.

- HAYKIN, S. Neural networks. 2. ed. Delhi: **Pearson Education**, 1999. p. 1-823.
- HUANG, M.; HUNG, Y.; YANG, Z. Validation of a method using Taguchi, response surface, neural network, and genetic algorithm. *Measurement*, v. 94, p. 284-294, 2016.
- J. WANG. Particle swarm optimization with adaptive parameter control and opposition. **Journal of Computational Information Systems**, 7:4463– 4470, 2011.
- JIAO, B.; LIAN, Z.; GU, X. A dynamic inertia weight particle swarm optimization algorithm. **Chaos, Solitons & Fractals**, v. 37, n. 3, p. 698-705, 2008.
- JORDEHI, A. AND JASNI, J. (2013). Parameter selection in particle swarm optimisation: a survey. **Journal of Experimental & Theoretical Artificial Intelligence**, [online] pp.527-542. Available at: <http://dx.doi.org/10.1080/0952813X.2013.782348> [Accessed 4 Feb. 2018].
- KENNEDY, J. AND EBERHART, R. (1995). Particle Swarm Optimization. **Proceedings of ICNN'95 - International Conference on Neural Networks**, pp.1942-1948.
- KONDA, R. et al. Design of experiments to study and optimize process performance. **International Journal of Quality & Reliability Management**, v. 16, n. 1, p. 56-71, 1999.
- LANGTON, C. Artificial life. 1. ed. Cambridge, Mass.: **MIT Press**, 1997. p. 1-336.
- LEE, K.; GILMORE, D. Formulation and process modeling of biopolymer (polyhydroxyalkanoates: PHAs) production from industrial wastes by novel crossed experimental design. **Process Biochemistry**, v. 40, n. 1, p. 229-246, 2005.
- LI, C. et al. A method integrating Taguchi, RSM and MOPSO to CNC machining parameters optimization for energy saving. **Journal of Cleaner Production**, v. 135, p. 263-275, 2016.
- LIU, C., OUYANG, C., ZHU, P. AND TANG, W. (2018). An Adaptive Fuzzy Weight PSO Algorithm. **In: Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on**. Shenzhen, China: IEEE, pp.8-10.
- MANIVEL, D.; GANDHINATHAN, R. Optimization of surface roughness and tool wear in hard turning of austempered ductile iron (grade 3) using Taguchi method. *Measurement*, v. 93, p. 108-116, 2016.
- MAYER, D.; BELWARD, J.; BURRAGE, K. Robust parameter settings of evolutionary algorithms for the optimisation of agricultural systems models. **Agricultural Systems**, v. 69, n. 3, p. 199-213, 2001.
- MENDES, R. et al. Multiobjective portfolio optimization of ARMA–GARCH time series based on experimental designs. **Computers & Operations Research**, v. 66, p. 434-444, 2016.
- MENDES, R.; KENNEDY, J.; NEVES, J. The Fully Informed Particle Swarm: Simpler, Maybe Better. **IEEE Transactions on Evolutionary Computation**, v. 8, n. 3, p. 204-210, 2004.
- MIGUEL, P. et al. Metodologia De Pesquisa Em Engenharia De Produção.2. ed. Rio de Janeiro: **Elsevier Editora Ltda**, 2012. p. 1-280.
- MILLONAS, M. M. Swarms, phase transitions, and collective intelligence. **In C.G. Langton (Ed.), Artificial Life III**, pp. 417-445. Reading, MA: Addison-Wesley, (1994).
- MONTGOMERY, D. (2012). Design and analysis of experiments. 8th ed. Hoboken, N.J: **Wiley**.
- MONTGOMERY, D. Design and analysis of experiments. 5. ed. Hoboken: **John Wiley & Sons**, 2004. p. 1-684.
- MONTGOMERY, D. Design and analysis of experiments. 7. ed. Hoboken: **Wiley**, 2009. p. 1-333
- MOSBAH, M. et al. Optimal of shunt capacitor placement and size in Algerian distribution network using particle swarm optimization. **8th International Conference on Modelling, Identification and Control (ICMIC)**, p. 192-197, 2016.
- MURPHY, T. Multivariate Quality Control Using Loss-Scaled Principal Components.. doctor – Georgia **Institute of Technology**, 2004.
- MURPHY, T.; TSUI, K.; ALLEN, J. A review of robust design methods for multiple responses. **Research in Engineering Design**, v. 16, n. 3, p. 118-132, 2005.
- NEDJAH, N.; MOURELLE, L. Swarm Intelligent Systems. 1. ed. Berlin, Heidelberg: **Springer-Verlag Berlin Heidelberg**, 2006. p. 1-184.

- NESHAT, M. et al. Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. **Artificial Intelligence Review**, v. 42, n. 4, p. 965-997, 2012.
- NIKNAM, T.; AZAD FARSANI, E. A hybrid self-adaptive particle swarm optimization and modified shuffled frog leaping algorithm for distribution feeder reconfiguration. **Engineering Applications of Artificial Intelligence**, v. 23, n. 8, p. 1340-1349, 2010.
- ORTIZ, F. et al. A Genetic Algorithm Approach to Multiple-Response Optimization. **Journal of Quality Technology**, v. 36, n. 4, p. 432-450, 2004.
- PAIVA, A. **Metodologia de Superfície de Resposta e Análise de Componentes Principais em Otimização de Processos de Manufatura com Múltiplas Respostas Correlacionadas**. Itajubá, Minas Gerais, Brazil: Tese de Doutorado - Programa de Pós Graduação em Engenharia de Produção, 2006.
- PAIVA, E. J. **Otimização de processos de manufatura com múltiplas respostas baseada em índices de capacidade**. Itajubá, Minas Gerais, Brazil: Dissertação de Mestrado - Programa de Pós Graduação em Engenharia de Produção, 2008.
- PANT M., THANGARAJ R., ABRAHAM A. (2009) Particle Swarm Optimization: Performance Tuning and Empirical Analysis. In: **Abraham A., Hassanien AE., Siarry P., Engelbrecht A. (eds) Foundations of Computational Intelligence. Studies in Computational Intelligence, vol 203**. Springer, Berlin, Heidelberg.
- PANTOJA, M. et al. Particle-Swarm Optimization in Antenna Design: Optimization of Log-Periodic Dipole Arrays. **IEEE Antennas and Propagation Magazine**, v. 49, n. 4, p. 34-47, 2007.
- PARPINELLI, R.; LOPES, H. New inspirations in swarm intelligence: a survey. **International Journal of Bio-Inspired Computation**, v. 3, n. 1, p. 1, 2011.
- PASSINO, K. Biomimicry of bacterial foraging for distributed optimization and control. **IEEE Control Systems Magazine**, v. 22, n. 3, p. 52-67, 2002.
- PATON, R. et al. *Computation in cells and tissues*. 1. ed. Berlin: **Springer**, 2004. p. 1-345.
- PATON, R. *Computing with biological metaphors*. 1. ed. London [u.a.]: **Chapman & Hall**, 1994. p. 1-504.
- PAVIČIĆ, M. *Quantum computation and quantum communication*. 1. ed. New York: **Springer**, 2006. p. 1-223.
- PIMENTA, C. D.; SILVA, M. B.; RIBEIRO, R. B.; CLARO, F. A. E. Método Taguchi aplicado na identificação dos fatores causadores da descarbonetação do arame de aço SAE 51B35, durante tratamento térmico de esferoidização. **GEPROS. Gestão da Produção, Operações e Sistemas**, Ano 7, nº 2, abr-jun/2012, p. 97-108.
- POLI, R., KENNEDY, J. AND BLACKWELL, T. (2007). Particle swarm optimization: An overview. **Swarm Intelligence**, pp.33-57.
- QIU, J. et al. Using animal instincts to design efficient biomedical studies via particle swarm optimization. **Swarm and Evolutionary Computation**, v. 18, n. 2210-6502, p. 1-10, 2014.
- RAO, S. (2009). *Engineering optimization: theory and praxis*. 4th ed. New York, N.Y: **Wiley**, p.813.
- REIS, CÉSAR; ANDRADE, JOÃO CARLOS DE. Planejamento experimental para misturas usando cromatografia em papel. **Química Nova**, 1996, p. 313-319. Disponível em: <[http://quimicanova.sbq.org.br/imagebank/pdf/Vol19No3\\_313\\_v19\\_n3\\_16.pdf](http://quimicanova.sbq.org.br/imagebank/pdf/Vol19No3_313_v19_n3_16.pdf)>. Acesso em: 4 fev. 2018.
- ROSSI, F. Blending response surface methodology and principal components analysis to match a target product. **Food Quality and Preference**, v. 12, n. 5-7, p. 457-465, 2001.
- SAAD, Y. *Iterative Methods for Sparse Linear Systems*. 2. ed. Beijing: **Society for Industrial and Applied Mathematics**, 2003. p. 1-528.
- SHI, Y.; EBERHART, R. A modified particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation Proceedings. **IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)**, n. 0-7803-4869-9, p. 69-73, 1998.
- SHI, Y.; EBERHART, R. A modified particle swarm optimizer. IEEE International Conference on Evolutionary Computation Proceedings. **IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)**, 1998.
- SHI, Y.; EBERHART, R. Empirical study of particle swarm optimization. **Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)**, n. 0-7803-5536-9, p. 1945-1950, [S.d.].
- SINČÁK, P. *Intelligent technologies--theory and applications*. 76. ed. Amsterdam: **IOS Press**, 2002. p. 214-220.

SONODA, DABIT; DE SOUZA, A.C. ZAMBRONI; DA SILVEIRA, PAULO MÁRCIO. Fault identification based on artificial immunological systems. **Electric Power Systems Research**, 2018, p. 24-34.

TAN, Y.; SHI, Y.; TAN, K. Advances in Swarm Intelligence. 1. ed. Berlin, Heidelberg: **Springer-Verlag Berlin Heidelberg**, 2010. p. 1-746.

TEODORO, F. R.; PARPINELLI, R. S.; LOPES, H. S. Otimização por colônia de bactérias aplicada a problemas de engenharia. Tradução. **Asociación Argentina de Mecánica Computacional**, 2010, p. 9615-9627.

VAN GYSEGHEM, E. et al. Selection of reversed-phase liquid chromatographic columns with diverse selectivity towards the potential separation of impurities in drugs. **Journal of Chromatography**, v. 1042, n. 1-2, p. 69-80, 2004.

VANDENBERGH, F.; ENGELBRECHT, A. A Cooperative Approach to Particle Swarm Optimization. **IEEE Transactions on Evolutionary Computation**, v. 8, n. 3, p. 225-239, 2004.

VERA CANDIOTI, L. et al. Experimental design and multiple response optimization. Using the desirability function in analytical methods development. **Talanta**, v. 124, n. 0039-9140, p. 123-138, 2014.

YANG, X. A New Metaheuristic Bat-Inspired Algorithm. Nature Inspired Cooperative Strategies for Optimization (NCSO 2010) (Eds. J. R. Gonzalez et al.), **Studies in Computational Intelligence**, v. 284, p. 65-74, 2010.

YASUDA, K. AND IWASAKI, N. Adaptive particle swarm optimization using velocity information of swarm. In: Systems, Man and Cybernetics, **IEEE International Conference on. The Hague, Netherlands: IEEE**, pp.3475-3481,2004.

YASUDA, K.; YAZAWA, K.; MOTOKI, M. Particle Swarm Optimization with Parameter Self-Adjusting Mechanism. **IEEJ Transactions on Electrical and Electronic Engineering**, v. 5, n. 2, p. 256-257, 2010.

YUE-LIN, G.; YU-HONG, D. A New Particle Swarm Optimization Algorithm with Random Inertia Weight and Evolution Strategy. **International Conference on Computational Intelligence and Security Workshops (CISW 2007)**, 2007.



## ANEXOS A – Código do Algoritmo *Particle swarm optimization*

Tabela A.1 (a) – Código do algoritmo

### Código do algoritmo

---

```

clc;
clear all;
close all
format long
cont=0;

k=1;
for k = 1:2
    t1(k,:) = clock;

%função a ser minimizada! Função Rastrigin
f = @(x1,x2) (x1^2 - 10*cos(2*pi*x1)+10)+(x2^2 - 10*cos(2*pi*x2)+10);
%função a ser minimizada! Função Rosenbrock
%f = @(x1,x2) (1-x1)^2 + 100*(x2-x1^2)^2;
%f = @(x1,x2) (1-x1).^2 + 100.*(x2-(x1.^2)).^2;
%função a ser minimizada! Função Esférica
%f = @(x1,x2) (x1*x1)+(x2*x2)
%função a ser minimizada! Função

%%
Np = 80; %Números Partículas
Min = repmat([-2.048 -2.048 ],Np,1); %Limites inferiores
para o espaço
Max = repmat([2.048 2.048 ],Np,1); %limites Superiores para o espaço

n = size(Max,2); %Números de cordenadas(variaveis)
Maxiter = 200; %Números de iterações

C1 = 2.23655;
C2 = 1.86345;
W = 0.9;%1;
K = 1;%0.729844;

[X] = particulas(Np,Min,Max); %As partículas são criados
aleatoriamente no espaco de solução

[FO] = funcionAdaptacion(f,X,Np); %A função de cada indivíduo é
calculado (função deve ser melhorado para tornar geral)

d=0; %se d = 1, então a velocidade
inicial aleatória é definida, %se d = 0, em seguida, a
velocidade inicial é ajustado para zero

V = velocidade(Np,n,d); %Cálculo de velocidades
inciais;

```

---

Tabela A.1 (b) – Código do algoritmo

## Código do algoritmo

---

```

iter = 1;
Pbest = X;

[v,p]= min(FO);                                %Menor valor da função objetivo
Gbest = X(p,:);                                %Melhor valor de todas as partículas

figure(iter)
%plotagem de duas variaveis
%   plot(X(:,1),X(:,2),'*r')                    %gráfico das partículas
%   xlim([-5.12,5.12])
%   ylim([-5.12,5.12])

%   plot3(X(:,1),X(:,2),X(:,3),'*r')            %gráfico das partículas
%   xlim([-5.12,5.12])
%   ylim([-5.12,5.12])
%   zlim auto;

contiter = 0;
arm_Gbest = 0;
while iter<=Maxiter
    %[C1,C2,W,K] = calcular(Wmin,Wmax,Cmin,Cmax,Maxiter,alpha,iter);

    [V] = CalVelocidades(V,C1,C2,W,K,Gbest,Np,X,iter,Pbest);
    [X] = posiciones(V,X,Np);
    [FO] = funcionAdaptacion(f,X,Np);            %Cálcula a função para
cada individuo
    [Pbest,FObest] = Pmelhor(f,Pbest,X,FO,Np);    % Guarda a melhor
posição visitadas pela partícula
    Gbest = Gmelhor(Gbest,Pbest,FObest);
    pegaminimoFObest = min(FObest);

%Contador de iteração
    if (Gbest ~= arm_Gbest)
        if (Gbest <= arm_Gbest)

            end
            contiter = iter;

        end
        arm_Gbest = Gbest;
    end

    pause(0.1);
    iter = iter+1;

%   %Plotando o gráfico por(iteração) de duas variaveis
%
    plot(X(:,1),X(:,2),'*r',Pbest(:,1),Pbest(:,2),'*b')    %Gráfico das

```

---

**Tabela A.1 (c) – Código do algoritmo**

---

```

xlim([-5.12,5.12])
ylim([-5.12,5.12])
%
end

arm_contiter(k,:) = contiter;

if k==1
resultad1=Pbest
end
if k==2
resultad2=Pbest
end
if k==3
resultad3=Pbest
end
if k==4
resultad4=Pbest
end
if k==5
resultad5=Pbest
end
if k==6
resultad6=Pbest
end

if k==7
resultad7=Pbest
end
if k==8
resultad8=Pbest
end
if k==9
resultad9=Pbest
end
if k==10
resultad10=Pbest
end
if k==11
resultad11=Pbest
end
if k==12
resultad12=Pbest
end

if k==13
resultad13=Pbest
end
if k==14
resultad14=Pbest
end
if k==15
resultad15=Pbest
end

```

---

Tabela A.1 (d) – Código do algoritmo

## Código do algoritmo

---

```

if k==16
resultad16=Pbest
end
if k==17
resultad17=Pbest
end
if k==18
resultad18=Pbest
end

if k==19
resultad19=Pbest
end
if k==20
resultad20=Pbest
end
if k==21
resultad21=Pbest
end
if k==22
resultad22=Pbest
end
if k==23
resultad23=Pbest
end
if k==24
resultad24=Pbest
end

if k==25
resultad25=Pbest
end
if k==26
resultad26=Pbest
end
if k==27
resultad27=Pbest
end
if k==28
resultad28=Pbest
end
if k==29
resultad29=Pbest
end
if k==30
resultad30=Pbest
end

if k==31
resultad31=Pbest
end
if k==32
resultad32=Pbest
end
if k==33
resultad33=Pbest
end

```

---

Tabela A.1 (e) – Código do algoritmo

## Código do algoritmo

---

```

if k==34
resultad34=Pbest
end
if k==35
resultad35=Pbest
end
if k==36
resultad36=Pbest
end
  if k==37
resultad37=Pbest
end
if k==38
resultad38=Pbest
end
if k==39
resultad39=Pbest
end
if k==40
resultad40=Pbest
end
if k==41
resultad41=Pbest
end
if k==42
resultad42=Pbest
end
if k==43
resultad43=Pbest
end
if k==44
resultad44=Pbest
end
if k==45
resultad45=Pbest
end
if k==46
resultad46=Pbest
end
if k==47
resultad47=Pbest
end
if k==48
resultad48=Pbest
end

if k==49
resultad49=Pbest
end
if k==50
resultad50=Pbest
end
resultado(k,:)=Gbest

t2(k,:) = clock;

```

---

Tabela A.1 (f) – Código do algoritmo

## Código do algoritmo

---

```

temporun = etime(t2,t1);

k=k+1;
clear -k
end

Cálculo das velocidades:
function [V] = CalVelocidades(V,C1,C2,W,K,Gbest,Np,X,iter,Pbest)

for i=1:1:Np    %cálculo de novas velocidades baseados na experiencia do
enxame
V(i,:) = K * W * (V(i,:) + C1*rand*(Pbest(i,:) - X(i,:)) +
C2*rand*(Gbest'-X(i,:))');
End

Função de Adaptação:

function [FO]=funcionAdaptacion(f,X,Np)

for i=1:1:Np
FO(i,1) = f(X(i,1),X(i,2));
end
end

Cálculo das Melhores partículas do Enxame:

function [Gbest] = Gmelhor(Gbest,Pbest,FObest);

[v,p] = min(FObest)

Gbest = Pbest(p,:)

end
Aleatoriedade de todas as partículas:

for i=1:1:Np    %todas as partículas são criados aleatoriamente
for j=1:1:size(Min,2)
X(i,j) = ((Max(i,j) - Min(i,j))*rand) + Min(i,j);
end
end

Cálculo das Melhores partículas individuais:

function [Pbest,FObest] = Pmelhor(f,Pbest,X,FO,Np)
%Para duas variáveis
%
%
for i=1:1:Np
c = f(Pbest(i,1),Pbest(i,2))           %Função objetivo das
melhores posicões das partículas      %Para o caso de
Minimização
if FO(i,1) < c
Pbest(i,:) =X(i,:);
end
FObest(i,1) = c;
end

```

---

Tabela A.1 (f) – Código do algoritmo

---

**Código do algoritmo**

---

Atualização da posição de cada partícula:

```
function [X] = posiciones(V,X,Np)

for i=1:1:Np           %atualizando a posição
    X(i,:) = X(i,:) + V(i,:);
end
```

Definição da velocidade aleatória de cada partícula:

```
function [V] = velocidade(Np,n,d)

if d==1

    for i=1:1:Np        %criando todas as velocidades aleatórias
        for j=1:1:n
            V(i,j) = 2*rand;
        end
    end
else
    V=zeros(Np,n);
end
```

---

## ANEXOS B – Tempo e Iterações encontrados para as 26 configurações com 50 rodadas da função Esférica

**Tabela B.1 (a)** – Dados para estudos dos tempos na performance do algoritmo

N	Tempos								
	Tempo_Conf_1	Tempo_Conf_2	Tempo_Conf_3	Tempo_Conf_4	Tempo_Conf_5	Tempo_Conf_6	Tempo_Conf_7	Tempo_Conf_8	Tempo_Conf_9
1	32,4690	26,7030	28,3730	29,0110	32,6040	28,3670	28,8400	28,2130	27,5390
2	28,1870	26,3230	26,5090	25,3730	41,7970	28,0440	27,5150	27,4680	26,9710
3	27,4320	26,1940	26,3640	25,1480	26,8760	27,9790	26,6280	27,5840	27,0750
4	28,1530	26,2660	26,0620	25,1460	26,8010	27,7090	27,2730	27,3970	27,1550
5	27,7370	26,3420	25,7530	25,3140	26,4040	27,8560	27,4430	27,4660	27,2630
6	28,2350	26,4970	26,2050	25,2380	26,9520	28,0160	27,2860	27,5500	27,1780
7	27,5880	26,4880	25,8340	25,2090	27,6680	27,9670	26,8440	27,2300	27,3560
8	27,6470	26,4770	26,0300	25,1300	26,7540	28,0810	27,6570	27,5540	27,1770
9	27,2640	26,3540	25,9220	25,3670	27,2350	28,1850	26,7840	27,4660	27,1600
10	27,5730	26,7290	25,9310	25,3710	27,0540	27,7090	27,0120	27,4910	26,9490
11	27,6070	26,2360	26,3330	25,1670	27,1360	27,8320	26,9010	27,2900	27,0090
12	27,5520	26,4090	26,0190	25,1180	26,7750	28,0160	27,7780	27,3790	26,6500
13	27,4380	26,0700	26,7980	25,0850	26,6940	27,9990	27,1280	27,3220	26,8700
14	27,7440	26,4600	26,1820	24,9230	26,7100	27,9550	27,0030	27,4140	27,1040
15	27,8130	26,4090	26,1230	24,8680	26,6750	28,1470	27,2530	27,5390	27,0230
16	27,7360	26,6040	26,6620	24,6330	26,8650	27,9270	27,2630	27,2760	26,8800
17	27,5790	26,9240	26,2760	24,9800	26,5250	27,8880	27,2590	27,5440	27,2090
18	28,0720	26,6120	26,1490	25,0170	26,9170	28,0560	26,9610	27,3620	27,0410
19	27,6700	26,7150	26,5410	25,1500	26,9170	27,8990	27,3110	27,4700	27,0900
20	27,5010	26,3940	26,3180	24,8550	26,8890	27,9890	27,4440	27,5960	27,0180
21	27,5930	26,3350	26,7900	25,0990	26,9820	27,9060	27,2020	27,7050	27,1640
22	27,6290	26,0460	25,7660	25,1370	27,0130	28,0130	27,2400	27,7230	27,3640
23	27,6320	26,0810	26,2190	25,0820	26,9670	27,8560	27,3070	27,5490	27,8230



**Tabela B.1 (b)** – Dados para estudos dos tempos na performance do algoritmo

N	Tempos								
	Tempo_Conf_1	Tempo_Conf_2	Tempo_Conf_3	Tempo_Conf_4	Tempo_Conf_5	Tempo_Conf_6	Tempo_Conf_7	Tempo_Conf_8	Tempo_Conf_9
24	27,3880	27,6500	26,2550	25,1670	26,9860	27,6330	27,2550	27,7040	27,4090
25	27,6560	26,6090	26,2420	25,1570	26,8230	28,9680	27,3580	27,6790	27,3200
26	27,5530	25,9770	26,4110	24,9880	26,6520	28,6710	26,8070	27,6300	26,9070
27	27,6460	26,3560	26,2100	25,0650	27,1240	30,4260	27,2690	27,5440	27,0470
28	27,1800	26,3950	26,3590	25,2020	27,0720	29,0290	27,5380	27,5670	27,1900
29	27,5570	26,3380	26,2270	25,2070	27,2060	28,1530	27,4670	27,5870	27,0460
30	27,5810	26,2390	26,3380	25,1300	26,9100	28,2180	27,5160	27,6180	27,2890
31	27,2020	26,4690	26,3220	24,7870	26,6190	28,2470	28,0120	27,7020	27,3340
32	27,1810	26,3050	26,3000	25,1630	26,9220	27,8660	27,7850	27,6630	27,7760
33	27,1830	26,2190	26,2570	24,9150	26,9120	27,6140	27,4590	27,3900	27,1920
34	27,2840	26,4890	26,1800	25,0860	27,1790	27,6380	27,6130	27,4070	27,4770
35	27,1520	26,1110	26,2920	25,1910	26,3290	28,0000	26,7040	27,2030	27,2930
36	28,3340	26,1090	26,2490	25,2730	26,9950	28,0870	27,1530	27,4520	27,4160
37	27,5700	25,9660	26,2510	25,3320	26,7990	27,8960	26,6820	27,4250	27,1560
38	27,3630	25,7300	26,4510	25,1600	27,0510	27,9670	26,4560	27,5660	27,3460
39	27,2570	26,6570	26,3210	25,1530	26,7330	28,1550	27,9710	26,8260	27,2360
40	27,3730	26,6470	26,2120	25,1850	26,7890	28,0750	26,9570	26,9080	27,3230
41	27,4860	26,4380	26,2850	25,1340	26,9530	27,9580	27,1450	28,1310	27,4170
42	27,3670	26,1910	25,9660	25,1350	26,9820	28,8140	27,4700	26,8150	27,1410
43	27,8120	26,5090	25,7500	25,2220	27,2440	29,5530	27,5460	27,4840	27,4530
44	27,5230	26,3360	25,8860	25,2560	27,2370	28,9750	27,2730	27,3920	27,3170
45	27,7350	26,5390	26,0200	25,1580	26,9390	28,3290	27,3350	27,3590	27,1890
46	27,7060	26,5640	26,1690	25,5740	27,0630	28,5910	27,2460	27,3540	27,1070
47	27,7720	26,4760	25,9520	25,1140	27,0260	28,4710	27,9540	27,5060	27,6200
48	27,7210	26,6540	26,2530	25,2930	27,0520	28,6100	27,3330	27,3430	27,2990
49	32,3790	26,4040	26,3440	25,7940	27,2700	28,5280	27,2480	27,5370	27,0670
50	28,0710	26,3710	26,1890	25,4600	27,4610	29,2190	27,7500	27,5420	27,3800

**Tabela B.1 (c)** – Dados para estudos dos tempos na performance do algoritmo

Tempos								
Tempo_Conf_10	Tempo_Conf_11	Tempo_Conf_12	Tempo_Conf_13	Tempo_Conf_14	Tempo_Conf_15	Tempo_Conf_16	Tempo_Conf_17	Tempo_Conf_18
32,8110	28,8270	27,7290	28,1090	28,2930	30,9090	28,1530	28,2790	28,9890
25,4510	27,5490	27,7770	26,6810	26,8610	25,7260	27,6780	28,0680	28,1920
25,5950	27,5040	27,2820	26,9540	26,7930	25,4780	27,6820	26,5110	27,9990
25,2730	27,4960	27,2670	27,1580	27,0520	25,7770	27,8070	26,0100	27,4760
25,2790	27,3570	27,3150	27,5890	27,7960	25,2410	27,6600	25,9520	27,4240
25,4230	27,6280	27,3970	26,7470	26,6820	25,4320	27,5490	26,2500	27,4770
25,7240	27,5310	27,2620	26,8090	27,1340	25,4450	27,3990	26,5820	27,4520
26,8130	27,4770	27,0040	27,4940	30,8500	25,3980	27,6090	26,8300	27,6190
25,5490	27,5680	27,1950	26,9750	27,1870	25,2720	27,5520	26,6640	27,5520
25,0020	27,4950	26,8880	26,5910	27,5550	25,3520	27,5190	26,8030	27,7680
24,9240	27,5360	26,9720	26,3220	27,8060	25,2640	27,4230	26,9050	27,8600
25,4140	27,5230	26,9700	26,7090	27,4640	25,3050	27,3990	26,5580	27,6750
25,6320	27,2970	27,1950	26,2530	26,9700	25,2290	27,6520	26,7440	27,7140
26,0050	27,3810	27,0920	26,4660	27,0530	25,4090	27,6460	26,7360	27,7050
26,2140	27,1490	27,2800	26,5390	27,2850	25,5970	27,6110	26,6990	27,4310
25,6510	27,2780	26,7480	26,6700	27,7800	25,5560	27,6260	26,6980	27,5370
25,5390	27,7230	26,9490	26,6340	27,4870	25,7190	28,1000	26,4230	27,4890
25,4160	27,3760	27,3300	26,7700	27,3970	25,3220	27,6710	27,9690	27,7660
25,1590	27,3930	27,1990	27,0130	27,4810	25,2590	27,8010	27,6060	27,8090
25,5570	27,4080	26,9680	26,9700	27,5710	28,1400	27,8730	27,6590	27,7300
25,8390	27,0400	27,1450	26,9060	27,4770	26,1480	27,7200	27,7210	27,4920
25,7270	26,8490	27,1890	26,7140	27,4070	26,2150	27,9070	27,8450	27,5860
25,8540	27,1900	27,0850	26,5510	27,4570	26,0720	27,7100	27,7510	27,3070
25,9640	27,4300	26,9500	26,8560	27,4950	26,2840	27,6530	27,6770	27,5910
25,7800	27,7460	26,8470	26,5770	27,4720	26,3560	27,5720	27,5060	27,7770
25,8980	27,3470	26,9160	26,5500	27,3760	26,1870	27,7570	26,8360	27,4890
25,5890	27,5730	27,1690	26,4050	27,6040	26,4380	27,7850	26,5490	27,0670
26,0050	27,4440	27,1590	26,9160	27,4100	26,1620	27,8200	26,6630	27,7670
25,7610	27,2680	27,0880	26,6880	27,2560	26,1630	27,6880	27,6140	27,7610

**Tabela B.1 (d)** – Dados para estudos dos tempos na performance do algoritmo

Tempos								
Tempo_Conf_10	Tempo_Conf_11	Tempo_Conf_12	Tempo_Conf_13	Tempo_Conf_14	Tempo_Conf_15	Tempo_Conf_16	Tempo_Conf_17	Tempo_Conf_18
25,7830	27,5810	27,3880	26,5790	27,1300	26,2040	27,6440	26,9070	27,7650
25,8010	27,2470	27,1530	27,0620	27,2850	26,1820	27,7030	27,5370	28,1950
25,8070	27,5140	27,2740	26,8540	27,2890	26,3720	27,7350	27,7720	27,8890
25,7850	27,3910	26,9060	26,7940	27,4250	26,0830	27,5840	27,7170	27,9760
25,8300	27,1940	27,2670	26,8540	27,1330	26,2140	27,5590	27,7840	26,9560
25,9160	27,4580	27,0170	26,8870	27,3750	26,2780	27,7850	27,8700	26,7140
25,9340	27,3940	27,2740	26,7800	27,5340	26,0220	27,5180	27,7100	27,2080
25,8200	27,1950	27,2560	27,4050	26,8630	25,8500	27,8600	27,6120	28,1780
25,6200	27,3880	26,6710	26,9680	26,8810	26,0550	27,6260	26,4950	27,4430
25,8890	27,5500	26,4710	26,7200	27,8330	26,2530	27,8430	27,0270	27,6480
25,8570	27,4470	27,1350	26,9000	26,8120	26,1460	27,6340	27,2540	27,5980
25,7440	27,5710	26,7620	26,5370	26,7740	26,0060	27,6110	26,9380	27,2130
25,8510	27,5440	26,5150	27,4370	25,3050	25,9560	27,6370	26,8510	27,4540
25,7170	27,3690	26,6670	30,3600	26,0590	25,9120	27,8050	26,8140	26,9840
25,6160	27,3010	26,7820	28,1070	26,5940	26,2890	27,7790	27,5490	27,8240
25,5640	27,6690	26,6140	26,9970	27,2450	26,1990	27,7220	27,4190	26,4960
25,8510	27,7790	26,6840	27,4270	26,8950	25,8920	27,6570	26,3190	26,6960
25,8150	27,7140	26,7830	27,7360	26,7010	26,2140	27,6740	26,8140	26,6180
25,9000	27,6970	27,2710	26,5500	27,0020	26,2990	27,7870	27,3490	26,5170
25,6690	27,8190	26,8600	26,8780	27,2570	26,7890	27,7210	26,9520	26,9860
25,7490	27,7950	26,8800	26,8430	27,3710	26,0920	27,6350	27,2590	27,5260

**Tabela B.1 (e)** – Dados para estudos dos tempos na performance do algoritmo

Tempos								
Tempo_Conf_19	Tempo_Conf_20	Tempo_Conf_21	Tempo_Conf_22	Tempo_Conf_23	Tempo_Conf_24	Tempo_Conf_25	Tempo_Conf_26	Tempo_da_oti
29,1450	27,3270	28,0930	32,2280	27,310	28,4280	27,1430	28,1920	25,318
28,4880	26,8730	27,7290	27,1630	25,098	27,4590	25,7340	27,7120	25,054
27,9950	26,8370	27,5260	27,4450	25,054	27,7530	25,5440	27,2570	24,976
44,2460	26,8420	27,6660	27,2370	25,057	27,5650	25,5390	27,0550	25,085
28,0610	26,9720	32,3570	27,3110	25,092	27,6290	25,5660	27,0710	24,975
28,0260	26,7330	27,7160	27,6550	24,912	27,0960	25,3830	26,8930	24,945
27,9630	26,8330	27,4940	27,4890	24,839	27,3340	25,6600	27,2030	25,006
27,6350	26,6390	27,3880	27,4610	25,105	27,4860	25,6310	27,0580	25,007
27,7620	27,0390	27,4800	27,6070	25,129	27,5450	25,1710	27,3790	25,023
27,9850	26,9840	27,5100	27,4750	25,115	27,4770	25,6230	27,2710	25,038
27,9430	27,0420	27,6130	27,4300	25,208	27,1420	25,6120	27,2070	25,022
27,6770	26,9300	27,7050	26,8300	25,152	27,2690	25,2930	27,0920	25,038
27,8350	27,0160	27,4850	26,9330	25,222	27,3820	25,5900	27,2790	25,007
28,0940	27,2400	27,7850	26,5530	25,164	27,0570	25,5050	27,0530	24,96
27,7960	26,8760	27,5460	27,0270	25,160	27,5030	25,5270	27,2900	25,007
28,0220	26,9800	27,5840	26,8260	25,166	27,6240	25,4310	27,1250	25,038
27,9220	26,8300	27,3590	25,9090	25,049	27,3800	25,4910	27,2100	25,022
27,9060	27,0000	27,5330	25,9500	25,194	27,1630	25,2920	26,6730	24,976
27,8010	27,1650	27,6110	27,4480	25,398	27,3160	25,5410	27,1320	24,976
27,8580	26,9370	27,5960	27,4670	25,107	27,2840	25,5000	27,1490	24,975
27,7080	26,8230	27,5510	27,4160	24,898	27,3170	25,5610	27,2520	24,992
27,7950	27,4690	27,5000	27,6050	24,722	27,2040	25,5170	27,1080	25,022
27,8690	26,9130	27,5790	29,3510	25,466	27,0300	25,4320	27,0360	24,991
27,7640	27,2180	27,6680	27,5660	25,222	27,0470	25,6810	27,3310	25,007
27,4040	26,7920	27,3560	27,6670	25,941	27,5740	25,5300	27,3030	25,023
27,1660	27,0670	27,6860	27,2390	26,892	27,6240	25,5650	27,2180	25,022

**Tabela B.1 (f)** – Dados para estudos dos tempos na performance do algoritmo

Tempos								
Tempo_Conf_19	Tempo_Conf_20	Tempo_Conf_21	Tempo_Conf_22	Tempo_Conf_23	Tempo_Conf_24	Tempo_Conf_25	Tempo_Conf_26	Tempo_da_oti
27,3870	27,0330	27,6140	27,6470	26,216	27,0570	25,5040	27,1960	25,007
27,3980	26,8670	27,6010	27,6410	26,639	27,0900	25,5820	27,2910	25,022
27,5290	27,0200	27,5220	27,5490	27,255	27,0850	25,4660	26,9940	25,007
27,9130	26,7650	27,5670	27,6710	26,308	27,0860	25,6880	27,2240	24,991
27,4630	27,0770	27,7200	27,8460	26,214	27,3090	25,5620	27,1940	24,976
27,2730	27,4100	27,7570	27,6880	26,097	27,1680	25,5970	27,4450	25,023
27,6030	27,8470	27,6260	27,5860	26,349	27,7070	25,8180	27,2650	25,022
27,8080	27,3930	27,7640	27,4860	25,973	26,7010	25,5920	27,4320	25,007
27,2410	26,9120	27,8320	27,7320	26,068	26,8220	25,6260	27,1550	25,007
27,4550	27,8340	27,5040	27,8970	26,069	27,0830	25,5870	27,2110	24,975
27,3740	26,8890	27,6120	27,6730	26,186	26,8260	25,4140	27,3560	25,376
27,5360	27,3520	27,7610	27,5840	25,912	27,0490	25,4440	27,4150	25,303
27,7330	26,8480	27,7930	27,3670	25,320	27,1330	25,5140	27,6090	26,333
27,5290	26,8920	27,5140	27,8620	25,346	27,2610	25,6390	27,0010	25,911
27,7720	26,6880	27,7170	27,7240	25,411	26,9500	25,5020	27,0190	25,179
27,4440	26,9260	27,7850	28,1030	25,603	26,9020	25,4440	27,3890	25,1
27,7710	27,0250	27,8040	27,2210	25,381	26,4930	25,4860	27,1780	24,835
27,9590	26,8860	27,6520	27,2790	25,442	26,7130	25,5430	27,4490	25,475
27,6290	27,1170	27,7700	27,3760	25,411	27,0790	25,5100	27,1890	24,929
28,1470	27,1020	27,6840	27,1580	25,460	27,2490	25,4010	27,4090	25,569
27,8620	26,9600	27,6150	27,0830	25,256	27,3650	25,5650	27,3010	25,365
27,8630	26,9760	27,4980	27,1720	25,366	27,1460	25,6340	27,4560	24,586
27,6210	27,0400	27,5350	27,3320	25,382	27,1630	25,5390	27,1340	25,724
27,8690	27,0860	27,5970	27,5010	25,563	27,2680	25,5950	26,9640	25,319

**Tabela B.2 (a)** – Dados para estudos das iterações na performance do algoritmo

N	Iterações								
	Iter_conf_1	Iter_conf_2	Iter_conf_3	Iter_conf_4	Iter_conf_5	Iter_conf_6	Iter_conf_7	Iter_conf_8	Iter_conf_9
1	1	4	7	5	6	4	1	1	12
2	1	3	17	4	12	14	1	5	1
3	1	6	1	2	31	1	1	23	2
4	1	15	14	10	1	1	3	5	17
5	1	6	3	17	40	9	7	2	1
6	1	3	3	1	5	1	2	4	1
7	1	21	5	2	2	9	1	1	2
8	1	4	2	6	2	2	5	7	1
9	1	20	3	17	1	3	21	1	1
10	1	9	3	1	8	5	5	4	7
11	200	4	2	8	2	7	1	22	1
12	200	15	10	2	33	19	6	1	9
13	1	5	7	19	22	2	5	7	6
14	200	6	7	5	7	5	4	5	16
15	1	14	2	3	2	1	7	4	1
16	1	4	2	14	16	25	1	11	4
17	1	8	6	5	11	2	4	1	2
18	1	2	6	26	7	9	11	1	8
19	1	6	3	4	1	2	1	8	1
20	1	6	8	15	1	1	26	10	1
21	1	3	18	6	1	1	3	3	4
22	200	64	2	1	15	9	4	1	1
23	200	4	4	1	3	12	2	1	1
24	1	16	5	15	2	1	13	6	7

**Tabela B.2 (b)** – Dados para estudos das iterações na performance do algoritmo

N	Iterações								
	Iter_conf_1	Iter_conf_2	Iter_conf_3	Iter_conf_4	Iter_conf_5	Iter_conf_6	Iter_conf_7	Iter_conf_8	Iter_conf_9
25	200	3	41	1	4	2	4	10	4
26	200	12	3	2	1	5	5	17	8
27	1	2	4	1	8	2	19	9	7
28	200	5	12	21	2	2	2	3	8
29	1	17	19	2	5	2	1	6	3
30	1	6	19	2	6	1	5	4	12
31	200	4	17	2	6	1	5	1	18
32	200	3	8	5	1	2	6	1	4
33	1	6	3	12	2	9	9	1	4
34	1	4	15	6	11	3	3	30	5
35	1	4	11	3	18	4	3	1	1
36	1	1	9	7	3	23	10	20	3
37	1	9	3	6	17	16	3	11	12
38	1	12	8	3	4	2	4	9	10
39	200	5	54	4	10	1	3	1	9
40	1	3	9	1	10	9	12	9	1
41	1	17	22	10	6	6	14	2	2
42	200	9	4	3	2	7	13	2	2
43	200	2	4	6	8	1	29	6	11
44	200	5	3	1	1	11	24	8	4
45	1	7	7	2	7	13	4	13	4
46	1	1	21	3	7	1	2	1	4
47	200	9	6	12	15	3	1	7	26
48	1	11	8	7	5	7	3	1	2
49	1	12	3	2	25	14	3	10	7
50	1	2	18	8	9	12	1	1	3

**Tabela B.2 (c)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_10	Iter_conf_11	Iter_conf_12	Iter_conf_13	Iter_conf_14	Iter_conf_15	Iter_conf_16	Iter_conf_17	Iter_conf_18
18	13	1	6	1	1	9	6	10
7	13	7	11	200	3	1	1	4
8	1	4	3	1	1	8	4	3
7	5	3	4	1	1	2	30	1
1	4	9	30	200	1	2	2	4
1	12	4	2	1	18	30	1	22
10	20	8	7	1	1	1	11	4
32	16	17	16	1	5	1	2	1
1	7	2	18	1	11	2	149	9
2	3	6	1	1	200	1	1	15
1	7	5	4	1	1	1	32	9
20	15	10	4	200	1	3	1	1
1	2	3	13	1	3	11	158	3
5	1	3	1	1	3	18	7	74
1	4	3	10	1	200	2	4	1
2	1	1	1	1	70	5	1	76
4	6	7	1	200	9	13	4	39
1	4	1	1	200	7	11	3	2
22	16	22	10	200	7	1	1	4
16	6	2	20	1	30	9	84	1
2	6	1	2	1	8	5	2	1
1	12	11	7	1	5	3	4	1
16	1	1	3	1	28	2	1	4
2	3	3	5	200	136	10	22	9
3	28	3	2	1	1	20	1	10
1	3	15	5	1	1	1	4	135



**Tabela B.2 (d)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_10	Iter_conf_11	Iter_conf_12	Iter_conf_13	Iter_conf_14	Iter_conf_15	Iter_conf_16	Iter_conf_17	Iter_conf_18
12	31	12	3	1	3	1	1	50
9	8	1	1	200	4	51	1	33
21	18	6	12	1	2	7	82	12
15	21	4	1	1	9	1	77	1
1	6	15	4	1	2	15	1	125
29	11	8	1	1	1	3	37	1
10	2	4	19	1	2	2	47	3
10	2	6	7	1	3	1	6	20
3	1	10	3	1	1	9	5	1
7	4	10	17	200	1	1	1	1
5	5	1	10	200	2	4	1	4
5	14	7	15	1	22	2	1	3
4	1	2	5	1	200	38	2	50
12	3	3	1	1	4	7	9	5
1	9	6	1	1	2	12	3	1
1	1	3	1	1	1	1	1	16
1	1	1	4	200	1	1	1	36
6	10	2	1	1	2	148	12	15
4	6	3	12	1	3	2	3	1
2	2	5	14	200	2	35	101	3
5	2	5	1	1	21	27	5	2
2	14	1	9	1	4	6	1	8
1	23	3	3	1	3	157	156	10
14	27	3	11	1	13	19	4	1

**Tabela B.2 (e)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_19	Iter_conf_20	Iter_conf_21	Iter_conf_22	Iter_conf_23	Iter_conf_24	Iter_conf_25	Iter_conf_26	Iter_oti
6	1	3	2	200	7	4	15	3
1	7	5	2	15	1	1	13	1
200	12	62	200	2	1	2	1	1
33	1	1	25	1	48	2	1	3
3	168	2	38	1	3	1	8	4
1	1	1	10	86	13	169	10	2
4	187	1	9	3	5	4	1	3
1	5	1	1	1	17	1	1	1
1	1	3	1	1	200	101	1	6
1	1	200	13	15	3	108	1	17
9	4	56	1	6	8	4	1	2
2	15	1	5	55	28	3	3	12
15	72	1	35	1	5	1	1	1
5	1	20	1	7	1	4	14	1
3	95	3	1	1	15	5	20	3
1	122	1	1	1	1	12	3	1
85	1	70	7	3	21	19	200	9
1	1	38	52	3	10	52	11	1
1	1	1	2	21	200	19	1	1
10	4	1	1	1	1	200	1	16
32	3	3	6	1	6	72	23	1
3	3	200	62	1	45	1	16	13
1	10	30	20	1	13	9	1	8
17	109	69	1	3	7	200	11	10
79	1	9	82	200	5	2	2	1

**Tabela B.2 (f)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_19	Iter_conf_20	Iter_conf_21	Iter_conf_22	Iter_conf_23	Iter_conf_24	Iter_conf_25	Iter_conf_26	Iter_oti
33	2	104	187	1	8	4	2	17
105	8	1	34	62	1	1	3	9
1	3	36	1	15	18	1	1	6
144	1	1	1	1	4	31	2	13
72	15	8	4	2	1	5	41	1
14	54	9	6	1	1	59	16	6
22	2	31	44	1	32	1	1	21
5	12	4	7	2	3	2	2	14
18	9	39	1	1	58	11	1	6
3	83	137	1	104	7	11	3	2
3	5	4	4	90	3	9	4	13
5	26	1	1	116	11	4	1	11
60	7	10	1	7	2	17	1	3
16	1	78	1	1	1	1	3	5
1	2	1	4	4	3	1	43	4
1	200	101	6	42	14	52	1	1
1	7	5	4	1	11	9	1	9
3	1	67	6	4	1	5	2	2
31	33	1	62	18	87	200	84	1
3	11	8	1	3	1	1	1	4
200	13	9	4	1	1	4	4	19
3	3	3	12	6	1	1	110	4
35	30	1	9	2	200	13	28	7
1	2	1	76	38	1	5	1	2
45	200	2	5	42	2	27	1	20

## ANEXOS C – Tempo e Iterações encontrados para as 26 configurações com 50 rodadas da função Rosembrock

**Tabela C.1 (a) – Dados para estudos dos tempos na performance do algoritmo**

N	Tempos								
	Tempo_Conf_1	Tempo_Conf_2	Tempo_Conf_3	Tempo_Conf_4	Tempo_Conf_5	Tempo_Conf_6	Tempo_Conf_7	Tempo_Conf_8	Tempo_Conf_9
1	28,2110	28,4480	26,5830	29,0980	29,2240	28,5750	27,5720	32,644	34,305
2	28,0960	28,0580	25,8690	27,6120	27,6490	27,7530	26,6260	28,181	27,347
3	28,1300	28,0690	25,6660	27,5850	27,5980	27,8630	26,6730	28,175	27,752
4	28,1340	28,0780	25,2150	27,4520	27,5030	28,0730	26,4420	28,004	27,320
5	28,1280	28,0870	24,8310	27,7250	27,5070	27,8210	26,6280	28,178	27,394
6	28,0920	28,0380	25,0750	27,6870	27,8090	27,9350	26,4050	28,031	27,519
7	28,0500	28,0630	25,3220	27,6080	27,7790	28,0060	26,6460	27,961	27,625
8	28,0800	28,0340	25,5370	27,7190	27,6900	27,6810	26,7530	27,919	27,577
9	28,1360	28,0490	25,8510	27,4920	27,5510	27,9640	26,6430	28,026	27,757
10	28,0990	28,0160	25,8410	27,7630	27,1930	27,7840	26,4280	28,062	27,813
11	28,1250	28,1510	25,7120	27,2890	27,7730	27,9240	26,5170	28,266	27,735
12	28,0840	28,0080	25,8330	27,7470	27,8980	27,9900	26,6190	28,28	27,546
13	27,8540	28,1080	25,7350	27,6720	27,9230	27,8320	26,8880	28,38	26,304
14	28,1470	28,0640	25,9430	27,7710	27,7060	27,9280	26,8990	28,313	27,487
15	28,1030	28,0960	25,9110	27,6470	27,4380	27,9680	26,3790	28,214	27,461
16	28,1110	28,0550	26,0530	27,5880	27,1650	27,8820	26,5760	28,365	27,508
17	28,1000	28,0340	25,5990	27,6900	27,8680	27,8520	26,9740	28,237	27,496
18	28,1160	28,0620	25,6050	27,7190	27,6290	27,8210	26,8460	28,469	27,437
19	28,0880	28,0520	25,6030	27,7630	27,6250	27,9060	26,3760	28,532	27,463
20	28,1120	28,0930	26,3480	27,7060	27,8350	27,9600	26,9390	28,384	27,147
21	27,8880	28,1480	26,2250	27,7870	27,9820	27,9740	26,6330	28,335	27,603
22	28,1410	28,0090	25,6710	27,7960	28,0130	27,9700	27,0490	28,223	27,148
23	28,1740	28,1460	25,6230	27,6110	28,0040	27,9320	26,6460	28,311	27,720

**Tabela C.1 (b)** – Dados para estudos dos tempos na performance do algoritmo

N	Tempos								
	Tempo_Conf_1	Tempo_Conf_2	Tempo_Conf_3	Tempo_Conf_4	Tempo_Conf_5	Tempo_Conf_6	Tempo_Conf_7	Tempo_Conf_8	Tempo_Conf_9
24	28,1350	28,0310	25,5780	27,6460	27,9080	27,7960	26,7040	28,339	27,718
25	28,1950	28,1100	25,5030	27,7770	27,8530	27,9290	26,6770	28,413	27,791
26	28,0270	27,9860	25,5060	28,1110	27,7100	27,8670	26,6100	28,279	27,726
27	28,1150	28,0600	25,5550	27,5820	27,7540	27,9310	26,4560	28,327	27,721
28	28,0820	28,1090	25,6600	27,7050	27,7680	27,8590	26,7520	28,327	27,752
29	28,0780	28,0970	25,6260	27,6140	27,8700	28,0020	26,4730	28,301	27,895
30	28,0790	28,0220	25,5350	27,6690	27,8490	27,8910	26,5550	28,231	27,783
31	28,1860	28,1540	25,4950	27,7210	27,8530	27,9900	26,8090	28,332	27,721
32	28,1060	27,9980	25,5940	27,6130	27,7970	27,8800	26,6490	28,331	27,855
33	28,1940	28,0480	25,7090	27,5080	27,8290	27,9240	26,4120	28,292	27,836
34	28,1100	28,0940	25,7390	27,6580	27,8360	27,8960	27,0250	28,41	27,823
35	28,1860	28,1390	25,8840	27,7770	27,7230	27,8980	26,9580	28,358	27,813
36	28,0500	28,1200	26,4340	27,7890	27,9730	27,8520	26,5540	28,374	27,768
37	28,1870	28,1040	26,3210	27,6560	27,9140	27,8790	26,7230	28,426	27,782
38	28,1890	27,9850	26,1230	27,7490	27,9390	27,9080	26,8130	28,383	27,520
39	28,1880	28,0950	26,0550	27,6420	28,0160	27,9210	26,8640	28,199	27,708
40	28,1040	28,0920	26,1740	27,6910	27,9720	27,8960	26,9930	28,291	27,666
41	28,0240	28,0990	25,6780	27,5980	28,0140	27,9180	26,2150	28,379	27,706
42	28,0560	28,0820	25,5590	27,7470	28,0840	27,8790	26,8830	28,421	27,522
43	28,1760	28,0420	25,3050	27,6900	27,9790	27,9900	26,8720	28,393	27,780
44	28,1280	28,1550	25,5530	27,7370	27,9870	27,9920	26,9150	28,409	27,707
45	28,1230	28,1270	25,8850	27,6130	28,0080	27,9710	26,4170	28,474	27,666
46	28,0970	27,9910	25,7750	27,5770	27,9630	27,6740	26,4890	28,117	27,709
47	28,0800	28,0270	25,9270	27,7530	27,9510	27,8980	26,2180	28,608	27,565
48	28,1110	28,0460	26,1600	27,7470	27,8980	27,8970	26,7300	29,517	27,582
49	28,1210	28,0250	26,0940	27,4910	27,9240	27,9660	26,7050	28,242	27,807
50	28,1400	28,0280	26,0650	27,4990	27,9520	27,8670	26,7210	28,431	27,692

**Tabela C.1 (c)** – Dados para estudos dos tempos na performance do algoritmo

Tempos							
Tempo_Conf_10	Tempo_Conf_11	Tempo_Conf_12	Tempo_Conf_13	Tempo_Conf_14	Tempo_Conf_15	Tempo_Conf_16	Tempo_Conf_17
28,6770	28,2780	28,0900	28,3190	27,0160	27,2280	27,3050	28,315
27,6260	27,7330	27,9010	28,0670	26,1410	28,9090	27,2570	25,907
27,2660	27,8490	27,8960	28,0590	26,1340	28,8160	26,9800	25,598
27,4090	27,8740	27,8310	27,9860	25,9580	26,1760	26,7180	25,771
26,9860	27,6000	27,7800	27,9360	25,8380	31,7360	26,7410	25,662
27,4220	27,8670	27,7870	28,0890	26,3170	32,3870	27,2820	25,79
27,5780	27,9200	27,8160	27,9830	26,4740	32,8880	27,3040	25,682
27,4180	27,9840	27,9390	28,0860	26,8620	32,9960	27,3660	25,822
27,5230	27,9730	27,8190	28,1690	26,7250	33,3820	27,1220	25,68
27,3910	27,9840	27,8700	28,0790	26,9700	32,3470	27,2670	25,792
27,7230	28,0380	27,6780	28,0670	26,7200	32,7650	27,2510	25,774
27,4590	27,9500	27,7890	28,1860	26,7530	32,0120	27,2350	25,742
27,4810	27,9860	27,8740	28,1140	26,7060	32,0090	27,1510	25,893
27,3030	27,9420	27,7900	28,0750	26,7430	27,2430	26,8330	25,638
27,2340	27,9390	27,8690	28,1610	26,8290	25,9570	26,9970	25,81
27,7260	27,9990	27,8800	28,1370	26,3320	25,8440	27,0460	25,868
27,3030	27,9980	27,7670	28,0940	26,7310	26,0350	27,2340	25,841
27,4760	27,9120	27,8990	28,0950	27,0340	26,1320	26,9380	25,752
39,3580	27,9980	27,9950	28,0300	26,9660	30,4790	27,1060	25,658
26,9530	27,9690	27,9520	28,1280	27,1280	32,2220	27,2610	25,709
26,8800	28,0480	27,9580	28,0830	26,9650	32,6140	27,0020	25,794
27,0700	28,0180	27,8360	28,0490	27,2190	32,8120	26,8680	25,818
27,4020	27,9380	27,7910	28,1070	27,1250	32,1910	26,4560	25,951
27,2730	27,9400	27,8980	28,0290	27,2600	33,0960	26,1940	25,838
27,0460	28,0770	27,9590	28,1910	27,0480	31,5460	26,5240	25,625

**Tabela C.1 (d)** – Dados para estudos dos tempos na performance do algoritmo

Tempos							
Tempo_Conf_10	Tempo_Conf_11	Tempo_Conf_12	Tempo_Conf_13	Tempo_Conf_14	Tempo_Conf_15	Tempo_Conf_16	Tempo_Conf_17
26,9750	27,9800	27,8310	28,0350	27,0820	32,0020	26,5440	25,863
27,0010	27,8840	27,8510	28,1360	26,8090	32,9790	26,5100	25,836
26,8750	27,9950	27,8440	27,9900	26,9310	32,8490	26,4440	25,728
26,9280	28,0080	28,0300	28,1410	27,2340	32,7080	26,5910	25,746
26,9230	28,0260	27,9160	28,1380	26,7860	32,2830	26,2200	25,853
27,0950	27,9170	27,8970	28,1170	27,0910	32,1320	26,1400	25,923
27,1420	28,0420	27,9130	28,1180	27,0880	32,7390	26,3410	25,886
27,1170	27,9890	27,9340	28,0960	26,5020	29,3970	26,5410	25,835
27,0130	27,9490	27,9140	28,0710	26,7470	26,0160	27,0190	25,819
27,0530	28,0360	27,9440	27,9780	26,2310	26,1330	27,0920	25,784
27,1320	28,0180	27,8530	28,0810	26,8050	25,8170	27,1590	25,891
26,9210	28,0460	27,9050	28,0760	26,5690	26,0030	27,3010	25,777
27,1100	27,9680	27,9490	28,1600	26,9210	31,6670	27,0070	25,782
27,2110	27,9890	28,0040	28,0330	27,0830	32,3360	26,9390	25,819
27,2820	27,3170	27,9900	28,0780	26,7650	30,9140	27,2380	25,613
27,4190	27,7220	28,0390	28,0790	26,8800	31,3850	27,1230	29,767
27,1580	27,9080	28,0310	28,1270	27,3020	29,4220	26,8280	26,142
27,1260	27,9330	27,9500	27,9660	27,1240	29,9520	27,0130	25,833
27,3550	27,9230	27,7970	28,2540	26,6400	28,1850	27,0530	25,846
27,1300	27,8690	27,7890	28,0800	26,3540	27,4790	26,8720	26,35
26,4880	27,8020	28,0430	28,2230	26,8270	27,3570	26,9380	25,915
26,6200	27,9410	28,0380	28,1340	27,0350	27,1670	26,8850	25,997
27,0200	27,9890	27,8410	28,0740	26,9660	27,2150	26,8560	26,153
27,2960	27,9320	28,0230	28,0270	26,8530	27,6220	26,8390	26,358
27,2850	27,9100	27,9750	28,1550	26,9180	27,6090	26,8550	26,387

**Tabela C.1 (e) – Dados para estudos dos tempos na performance do algoritmo**

<b>Tempos</b>							
<b>Tempo_Conf_18</b>	<b>Tempo_Conf_19</b>	<b>Tempo_Conf_20</b>	<b>Tempo_Conf_21</b>	<b>Tempo_Conf_22</b>	<b>Tempo_Conf_23</b>	<b>Tempo_Conf_24</b>	<b>Tempo_Conf_25</b>
29,5350	31,5050	28,5730	27,980	36,8330	27,7930	29,0170	28,2780
26,9550	27,1800	27,6220	27,344	27,3060	26,7120	27,8060	27,6960
27,3630	29,1870	27,5190	27,507	27,3580	26,7660	27,8380	27,6010
27,6700	27,4130	27,6230	27,419	27,5150	26,7870	27,6490	27,6760
27,6580	27,6260	27,4050	27,427	27,5240	27,0120	27,7970	26,9170
27,5780	27,3790	27,6860	27,640	27,6080	26,6860	28,0860	26,6930
27,7000	27,5570	27,8040	27,342	27,1930	25,5300	27,7620	27,6550
27,5930	27,6960	27,7040	27,509	27,6630	25,7090	27,6840	27,7220
27,6340	28,5010	27,6250	27,664	27,5120	25,5260	27,5790	27,5680
27,4540	27,3750	27,6780	27,787	27,4740	25,5360	27,7540	27,6270
27,5590	27,3800	27,5770	27,486	27,5450	25,5170	28,2240	27,7720
27,7750	27,0100	27,6190	27,657	27,1800	25,5550	27,7270	27,8040
28,0510	27,1150	27,6500	27,766	27,9310	25,4440	27,7880	27,8190
27,0030	26,7020	27,4440	27,273	27,2070	25,5230	27,7800	27,5750
27,7170	27,4720	27,6880	27,544	26,7660	25,4810	27,8450	27,7550
27,5410	27,3630	27,3580	27,516	27,1150	26,8490	28,2830	27,7350
27,5830	27,4080	27,5170	27,470	27,2370	25,6300	27,7300	27,7930
28,0050	26,3680	27,6780	27,562	27,7990	25,7350	27,8660	27,8820
27,5240	26,8710	27,5480	27,615	27,4750	25,6160	27,8900	27,7480
27,7440	27,4220	27,5940	27,643	27,3530	25,6760	27,8730	27,6770
29,5080	26,2620	27,4060	27,646	27,6620	25,3730	27,7700	27,8560
27,5530	25,5590	27,9210	27,513	27,9060	25,5350	27,5470	28,1140
27,2670	27,3890	27,3470	27,688	27,3720	25,3450	27,7470	26,6980
27,4690	27,4760	27,5470	27,534	27,2680	25,6100	27,7850	26,4500



**Tabela C.1 (f)** – Dados para estudos dos tempos na performance do algoritmo

Tempos							
Tempo_Conf_18	Tempo_Conf_19	Tempo_Conf_20	Tempo_Conf_21	Tempo_Conf_22	Tempo_Conf_23	Tempo_Conf_24	Tempo_Conf_25
27,7860	26,2310	27,5690	27,581	27,6280	25,5580	28,0840	27,7610
27,6250	26,2720	27,6590	27,676	27,6890	26,9230	27,7880	27,6910
27,9270	25,9240	27,5060	27,609	27,5820	26,3030	27,7150	27,9260
27,6560	26,7730	27,5260	27,609	27,5000	26,3120	27,8230	27,4870
27,4020	26,1570	27,5170	27,625	27,3950	26,2400	27,7160	27,5850
27,6300	25,7670	27,5370	27,594	27,3270	26,6040	27,9090	27,8400
27,5380	26,1160	27,4860	27,517	27,3810	26,5600	27,6200	28,0900
27,4690	26,0370	27,5740	27,536	27,3810	26,5050	27,6900	27,4860
27,6890	26,2930	27,4240	27,580	27,5420	26,3380	27,6570	28,6800
27,5040	25,4480	27,5190	27,562	27,4440	26,4660	27,5710	28,1030
27,2460	25,5490	27,6740	27,515	27,4850	26,3900	27,6500	27,6770
27,8010	26,2110	27,5180	27,585	27,6240	26,4180	27,7950	27,6840
27,4610	25,3920	27,5190	27,597	27,5510	27,0840	27,6740	27,7750
27,2030	25,7530	27,3260	27,633	27,2550	26,9680	27,8140	28,1360
27,6270	26,0060	27,6780	27,666	27,4190	27,1750	27,7030	27,4700
27,5950	25,4750	27,5490	27,410	27,7530	26,9420	27,6860	28,0380
27,5180	25,7190	27,5790	27,347	27,9220	27,0560	27,7830	28,2690
27,3270	26,2570	27,6120	27,499	27,3710	27,0240	27,7170	27,4670
27,3810	25,5450	27,4040	27,627	27,5250	26,8530	27,7910	27,7200
27,6090	25,6310	27,3650	27,623	27,5470	27,2350	27,6810	27,5800
27,5420	26,3020	27,5040	27,454	27,6160	25,8960	27,7680	27,4370
27,3950	29,4330	27,6760	27,633	27,5610	25,8270	27,8860	29,2300
26,5890	27,1030	27,5810	27,497	27,3470	25,7740	27,6820	27,8440
25,7900	30,2940	27,5730	27,476	27,5470	25,7350	27,6190	27,1910
26,2820	26,7010	27,6550	27,529	27,9560	25,7580	27,6470	27,7860
26,9320	26,2650	27,6250	27,579	27,1710	25,6960	27,6870	27,9840

**Tabela C.1(g)** – Dados para estudos dos tempos na performance do algoritmo

<b>Tempos</b>	
<b>Tempo_Conf_26</b>	<b>Tempo_da_oti</b>
28,3180	25,413
27,8700	25,178
27,9930	24,945
27,8320	24,835
27,7530	25,35
27,8920	25,225
27,9560	24,741
27,8280	25,335
27,7340	25,069
27,9270	25,834
27,9960	24,913
27,9060	24,913
27,8440	26,427
27,9600	25,584
27,8610	24,866
28,0140	25,194
27,9260	24,664
27,9970	24,835
27,8630	25,023
27,9640	25,022
30,0160	24,913
27,7360	25,023
27,9430	25,1
27,8560	24,991
27,8610	24,851

**Tabela C.1(h)** – Dados para estudos dos tempos na performance do algoritmo

<b>Tempos</b>	
<b>Tempo_Conf_26</b>	<b>Tempo_da_oti</b>
27,8020	25,038
27,7060	25,802
27,7970	26,068
27,9830	24,991
27,8920	25,194
27,8010	24,914
28,2630	24,913
27,6560	24,991
27,7240	25,085
27,9060	25,053
27,9580	24,914
28,2640	24,96
27,8420	24,913
27,8120	25,537
27,7390	25,422
27,7520	24,991
27,8360	25,194
28,2590	25,319
27,9280	26,567
27,8570	26,208
27,7940	26,52
27,7720	25,727
27,8090	25,298
27,7560	25,351
27,9070	25,385

**Tabela C.2 (a)** – Dados para estudos das iterações na performance do algoritmo

N	Iterações								
	Iter_conf_1	Iter_conf_2	Iter_conf_3	Iter_conf_4	Iter_conf_5	Iter_conf_6	Iter_conf_7	Iter_conf_8	Iter_conf_9
1	1	1	8	200	86	81	91	86	91
2	1	45	91	83	78	76	82	75	87
3	1	51	89	136	87	76	17	11	92
4	1	199	153	200	84	90	31	32	88
5	1	75	3	86	81	100	78	85	76
6	1	33	90	198	85	84	36	84	87
7	1	196	35	85	80	83	82	85	14
8	1	195	88	81	80	200	25	93	78
9	1	52	85	83	48	83	83	78	82
10	1	41	96	89	82	82	82	82	79
11	1	102	89	82	81	67	83	86	87
12	1	52	159	188	81	85	82	90	90
13	1	192	87	199	85	80	1	95	82
14	1	199	128	197	46	40	87	87	86
15	1	127	139	200	80	79	80	88	87
16	1	39	93	83	77	78	81	91	14
17	1	198	158	198	24	78	87	81	45
18	1	30	199	81	198	76	46	54	86
19	1	57	145	198	86	77	77	81	90
20	1	199	133	83	79	83	80	82	200
21	1	91	196	74	58	83	90	71	82
22	1	60	185	86	80	78	87	85	88
23	1	172	9	81	186	81	73	85	88
24	1	29	132	87	36	200	89	92	87
25	1	198	76	148	84	200	83	4	86

**Tabela C.2 (b)** – Dados para estudos das iterações na performance do algoritmo

N	Iterações								
	Iter_conf_1	Iter_conf_2	Iter_conf_3	Iter_conf_4	Iter_conf_5	Iter_conf_6	Iter_conf_7	Iter_conf_8	Iter_conf_9
26	1	197	128	162	85	79	84	78	87
27	1	200	143	56	198	83	84	83	87
28	1	152	3	199	69	80	90	78	85
29	1	200	28	196	74	200	78	86	87
30	1	46	5	85	82	77	81	84	83
31	1	56	147	83	79	74	123	84	86
32	1	199	84	199	75	81	83	85	86
33	1	49	139	200	82	80	81	85	84
34	1	47	86	85	79	82	85	80	88
35	1	190	1	78	200	81	30	86	86
36	1	44	196	168	154	80	90	76	74
37	1	199	11	79	78	75	62	83	93
38	1	198	177	90	83	82	11	83	1
39	1	59	89	89	67	83	79	80	85
40	1	61	199	8	83	22	80	87	83
41	1	200	200	84	193	82	78	14	88
42	1	65	58	84	86	82	83	78	87
43	1	45	27	77	200	83	83	87	90
44	1	57	150	81	191	188	77	38	80
45	1	53	200	155	75	75	79	83	30
46	1	199	142	198	70	83	83	87	83
47	1	194	88	88	106	92	1	1	17
48	1	54	7	197	75	84	88	76	84
49	1	56	136	84	83	83	78	85	40
50	1	200	89	200	79	84	69	1	200

**Tabela C.2 (c)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_10	Iter_conf_11	Iter_conf_12	Iter_conf_13	Iter_conf_14	Iter_conf_15	Iter_conf_16	Iter_conf_17	Iter_conf_18
100	94	89	84	1	28	28	156	194
92	92	15	88	1	8	8	140	194
93	93	85	84	1	2	2	38	23
91	94	86	88	1	42	42	23	137
93	76	183	84	1	195	195	6	9
87	96	200	86	1	45	45	178	179
91	96	81	1	1	8	8	112	130
91	98	166	86	1	119	119	21	184
81	94	85	1	1	140	140	7	144
91	92	86	101	1	83	83	151	194
87	95	190	80	1	121	121	157	127
91	95	200	83	1	108	108	120	19
92	90	195	77	1	13	13	77	60
93	96	160	81	1	2	2	55	175
87	97	200	86	1	8	8	151	52
87	87	200	80	1	30	30	33	90
90	96	79	86	1	64	64	193	76
83	95	86	75	1	22	22	11	120
92	91	47	83	1	52	52	122	87
86	89	195	68	1	19	19	101	163
90	93	157	95	1	78	78	92	158
90	92	88	89	1	9	9	174	193
90	94	84	86	1	8	8	125	16
91	87	194	1	1	52	52	50	187
92	103	192	85	1	23	23	117	115
89	88	10	82	1	49	49	75	111

**Tabela C.2 (d)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_10	Iter_conf_11	Iter_conf_12	Iter_conf_13	Iter_conf_14	Iter_conf_15	Iter_conf_16	Iter_conf_17	Iter_conf_18
79	84	146	85	1	67	67	43	90
85	90	12	86	1	16	16	92	137
92	94	200	88	1	56	56	98	178
89	98	159	84	1	73	73	50	175
87	200	200	85	1	94	94	43	190
87	89	85	84	1	153	153	173	154
86	78	198	84	1	7	7	96	27
94	90	179	89	1	110	110	138	146
92	104	85	85	1	35	35	185	133
89	94	90	57	1	10	10	186	38
87	90	78	97	1	1	1	28	58
92	92	27	83	1	87	87	64	195
89	92	198	83	1	46	46	51	153
86	200	185	83	1	80	80	96	179
92	96	83	87	1	4	4	166	184
89	89	85	88	1	30	30	59	187
91	93	83	14	1	50	50	45	173
90	103	85	75	1	53	53	45	134
95	94	58	78	1	12	12	101	76
91	97	199	80	1	108	108	81	48
99	96	145	88	1	75	75	124	196
89	92	148	86	1	95	95	120	127
95	97	196	85	1	75	75	124	45
88	94	85	78	1	90	90	121	157

**Tabela C.2 (e)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_19	Iter_conf_20	Iter_conf_21	Iter_conf_22	Iter_conf_23	Iter_conf_24	Iter_conf_25	Iter_conf_26	Iter_oti
178	178	32	4	1	1	136	83	93
59	59	73	137	84	24	10	45	95
134	134	52	13	22	18	149	41	88
186	186	70	11	68	52	157	40	96
20	20	109	38	82	14	148	63	95
146	146	71	5	144	45	23	103	96
183	183	151	127	79	37	46	60	95
132	132	156	135	1	72	145	6	94
53	53	96	113	102	31	7	72	93
101	101	162	45	8	1	61	154	94
55	55	108	57	9	2	178	70	93
199	199	120	50	42	1	197	83	91
132	132	193	10	40	1	83	103	95
181	181	187	123	132	1	73	64	100
118	118	34	1	4	86	144	56	96
68	68	112	20	13	13	46	11	97
93	93	164	102	67	12	19	75	105
51	51	19	135	14	24	39	26	97
106	106	170	81	1	47	63	36	89
175	175	133	50	29	36	131	174	95
178	178	100	186	5	8	85	1	93
174	174	6	84	28	14	31	140	93
191	191	37	189	18	58	87	54	93
115	115	80	80	77	34	143	28	92



Tabela C.2 (f) – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_19	Iter_conf_20	Iter_conf_21	Iter_conf_22	Iter_conf_23	Iter_conf_24	Iter_conf_25	Iter_conf_26	Iter_oti
147	147	1	17	36	24	1	51	95
126	126	128	100	43	63	51	48	92
26	26	37	68	70	68	122	16	98
38	38	134	36	48	104	138	159	89
123	123	69	106	9	1	141	91	93
11	11	24	24	1	2	76	9	87
146	146	145	29	1	13	86	50	96
175	175	77	74	41	22	2	36	97
126	126	117	10	21	9	48	104	92
198	198	195	145	69	6	114	94	158
85	85	86	76	1	8	37	57	95
59	59	179	48	5	49	38	132	96
173	173	30	4	33	27	82	63	95
163	163	70	162	20	3	38	86	94
75	75	159	108	23	14	106	29	95
157	157	27	89	74	26	17	68	92
175	175	129	85	29	1	59	95	92
127	127	120	25	76	25	5	94	97
185	185	9	83	19	38	33	6	101
97	97	167	1	118	28	171	24	96
106	106	136	15	34	40	19	144	95
156	156	158	13	71	1	156	124	89
96	96	95	131	22	52	28	14	97
164	164	39	61	121	43	4	14	92
159	159	70	45	15	65	139	46	97
39	39	139	99	127	19	50	82	92

## ANEXOS D – Tempo e Iterações encontrados para as 26 configurações com 50 rodadas da função Rastrigin

**Tabela D.1 (a)** – Dados para estudos dos tempos na performance do algoritmo

N	Tempos								
	Tempo_Conf_1	Tempo_Conf_2	Tempo_Conf_3	Tempo_Conf_4	Tempo_Conf_5	Tempo_Conf_6	Tempo_Conf_7	Tempo_Conf_8	Tempo_Conf_9
1	32,1990	28,009	30,5550	26,1160	28,0090	28,251	28,205	27,7030	28,2650
2	28,4560	27,894	25,4640	25,5640	27,8940	27,86	27,925	27,5920	27,2640
3	27,9900	27,897	25,4460	25,6280	27,8970	27,69	27,613	27,4720	26,3190
4	27,7580	28,074	25,6670	25,4200	28,0740	27,632	28,141	27,5930	26,0280
5	27,9530	27,848	25,8970	25,5540	27,8480	27,715	27,742	26,6720	27,2850
6	27,9070	27,962	25,6500	25,4990	27,9620	27,97	27,804	26,1250	27,5800
7	27,8140	27,995	25,6580	25,4980	27,9950	27,658	27,854	25,9170	27,5570
8	28,0040	28,000	25,4130	25,5220	28,0000	26,598	27,822	26,0070	27,5600
9	27,5840	27,895	25,2690	25,4300	27,8950	27,134	27,839	26,1660	27,3800
10	26,9190	27,630	25,2090	25,5650	27,6300	27,405	27,719	25,8220	27,6090
11	27,4690	27,651	25,6970	25,4560	27,6510	27,766	27,785	25,8170	26,7520
12	27,3740	27,755	25,4870	25,5200	27,7550	27,848	27,753	25,9870	27,4940
13	28,2070	27,579	25,5230	25,6000	27,5790	27,875	27,689	26,1080	27,8600
14	27,8270	27,633	25,5390	25,3640	27,6330	27,883	27,864	26,1710	27,4180
15	27,6950	27,605	25,6280	25,5250	27,6050	27,946	27,749	26,0130	27,0870
16	27,7980	27,642	25,7260	25,3870	27,6420	27,949	27,817	25,8930	28,1380
17	27,9090	27,666	25,6460	25,5340	27,6660	28,027	27,846	26,3580	27,8000
18	27,8470	27,670	26,2640	25,4060	27,6700	27,997	27,706	27,2600	26,8930
19	27,6870	27,575	26,3860	25,2980	27,5750	27,623	27,782	27,3740	26,9390
20	27,8120	27,439	26,4210	25,5660	27,4390	27,678	27,812	27,2940	27,2280
21	27,9760	27,539	25,6650	25,4230	27,5390	27,709	27,723	26,8020	28,1180
22	28,1180	27,657	25,9990	25,4900	27,6570	27,304	27,737	27,2310	27,4080
23	26,6810	27,711	25,9220	25,4180	27,7110	27,293	27,8	28,8950	27,3490
24	27,7380	27,561	25,8040	25,5380	27,5610	27,064	27,749	27,0130	27,1180
25	27,9050	27,717	25,8700	25,4180	27,7170	27,563	27,738	26,5930	26,8370
26	27,9390	27,828	25,8610	24,9440	27,8280	27,604	27,719	26,9310	27,3600

**Tabela D.1 (b)** – Dados para estudos dos tempos na performance do algoritmo

N	Tempos								
	Tempo_Conf_1	Tempo_Conf_2	Tempo_Conf_3	Tempo_Conf_4	Tempo_Conf_5	Tempo_Conf_6	Tempo_Conf_7	Tempo_Conf_8	Tempo_Conf_9
27	27,9810	27,708	25,6660	24,8980	27,7080	27,288	27,63	27,7520	27,5570
28	27,9070	27,608	25,7320	24,9130	27,6080	26,158	27,825	27,3570	27,4800
29	27,9110	27,680	25,6150	24,9750	27,6800	26,613	27,723	27,7510	27,6150
30	26,1990	27,576	25,9900	24,9450	27,5760	26,04	27,786	27,6720	27,3330
31	26,8720	27,474	25,2880	24,8820	27,4740	25,912	27,812	27,5500	27,4100
32	27,4400	27,531	25,3500	24,8980	27,5310	26,493	27,642	27,5160	27,6510
33	27,5030	27,773	25,4750	24,9750	27,7730	26,847	27,688	27,4570	27,4980
34	27,5490	27,670	25,1780	24,9760	27,6700	26,824	27,781	27,1480	27,1530
35	27,4570	27,474	24,7100	25,0220	27,4740	27,679	27,647	27,1730	27,5550
36	26,2390	27,502	24,9290	24,9140	27,5020	27,148	27,704	27,5080	27,6080
37	27,3900	27,683	25,2570	24,9130	27,6830	27,216	27,674	27,6210	26,7920
38	26,5220	27,738	25,0380	24,9750	27,7380	27,175	27,814	27,5300	25,5580
39	26,9150	27,638	25,0530	24,9450	27,6380	26,139	27,798	27,4910	27,6150
40	27,2380	27,630	25,2570	25,0070	27,6300	26,502	27,799	27,6170	27,5760
41	27,0430	27,781	25,1630	24,8820	27,7810	27,055	27,875	27,3430	27,6100
42	26,7080	27,537	25,1000	24,9750	27,5370	26,914	27,861	27,4270	27,5990
43	27,0920	27,473	25,1010	24,8980	27,4730	27,111	27,801	27,3030	27,1530
44	26,9920	27,659	24,9130	24,9760	27,6590	27,507	27,606	27,3940	27,6320
45	27,2300	27,555	24,7570	24,9910	27,5550	27,416	27,85	27,5800	27,1580
46	27,1630	27,700	24,9760	24,9290	27,7000	27,418	27,754	27,2690	27,6240
47	27,6600	27,437	25,1470	24,9130	27,4370	27,521	27,905	27,5160	27,4120
48	26,9290	27,755	25,0540	24,9760	27,7550	27,594	27,844	27,5790	26,5220
49	27,3370	27,604	25,1000	25,0220	27,6040	26,591	27,692	27,1760	27,5150
50	27,1330	27,757	25,0540	25,8310	27,7570	26,07	27,799	27,3330	27,4230

**Tabela D.1 (c)** – Dados para estudos dos tempos na performance do algoritmo

Tempos							
Tempo_Conf_10	Tempo_Conf_11	Tempo_Conf_12	Tempo_Conf_13	Tempo_Conf_14	Tempo_Conf_15	Tempo_Conf_16	Tempo_Conf_17
28,7040	26,156	26,146	26,266	26,4190	28,142	27,4510	25,9010
27,4710	25,225	24,851	25,253	26,2070	27,768	27,3530	26,0600
27,5970	25,256	25,319	24,960	26,0340	27,862	27,1510	25,8780
27,6900	25,039	25,131	25,117	26,3950	27,846	26,9640	26,0550
27,7520	25,084	24,929	24,960	25,9440	27,877	27,2200	26,0140
27,6440	25,023	25,226	24,960	26,1900	27,768	27,0220	26,0070
27,6610	24,913	24,928	24,804	26,0070	27,893	27,0830	25,8330
27,7000	24,960	25,070	24,929	26,0270	27,862	27,0920	26,1540
27,1690	25,194	25,178	24,928	26,1560	27,846	27,1860	25,8600
26,0040	24,945	25,038	25,366	26,0300	27,893	27,2760	26,0060
25,5400	24,991	25,132	24,710	26,0350	27,908	27,0090	26,0970
25,5650	25,007	25,038	24,742	26,1920	27,909	27,3370	25,8340
25,5250	25,038	25,116	25,007	26,2260	27,892	27,3920	25,8940
25,5620	24,975	25,007	24,929	26,2030	27,862	27,2670	25,2700
25,4950	25,054	24,975	25,334	26,1020	27,706	27,1700	25,6940
25,5440	25,069	25,257	24,929	26,1340	27,83	27,2280	25,6800
25,6010	24,960	24,975	25,381	26,1340	27,628	27,0160	26,1410
25,5200	25,054	25,116	24,945	26,3260	27,955	27,2070	26,0210
25,7270	25,022	25,070	24,929	26,2740	27,831	27,0080	25,9960
27,3160	25,023	25,022	25,038	26,2470	27,83	27,0010	25,9300
27,3200	24,960	25,116	25,209	26,1860	27,94	27,2340	25,8160
27,4180	24,913	25,085	25,070	26,0780	27,799	26,9080	25,9170
27,3400	24,960	25,225	25,162	26,2280	27,955	27,1120	26,1840
27,2600	25,069	25,038	25,085	25,3330	27,94	27,0890	25,9120

**Tabela D.1 (d)** – Dados para estudos dos tempos na performance do algoritmo

Tempos							
Tempo_Conf_10	Tempo_Conf_11	Tempo_Conf_12	Tempo_Conf_13	Tempo_Conf_14	Tempo_Conf_15	Tempo_Conf_16	Tempo_Conf_17
27,1740	24,992	25,194	25,023	25,2250	27,939	26,9080	26,0450
27,3310	24,960	25,210	24,929	25,1630	27,893	27,2860	25,9470
27,2840	24,975	25,069	24,913	25,1000	27,878	27,0280	25,9530
27,3610	25,038	25,069	25,241	25,1630	27,814	27,2380	26,0700
25,8140	25,054	24,929	25,022	25,2410	27,987	27,1890	26,0250
26,5800	24,960	25,522	25,194	25,2100	27,861	27,0800	25,9980
26,7140	25,038	24,929	24,929	25,0850	27,94	26,7510	26,0550
26,6060	25,085	25,007	25,163	25,2090	27,955	26,7390	26,0860
26,6140	25,054	25,194	25,100	25,2570	27,846	27,1410	25,7900
26,9600	25,069	24,897	25,241	25,2250	27,909	27,0800	26,0940
26,7420	25,007	25,038	24,789	25,2560	27,799	26,9740	26,0050
26,6780	24,960	25,038	25,287	25,1320	27,924	27,0330	26,1720
32,2640	24,991	25,101	25,163	25,2720	27,846	27,2760	26,1050
26,0830	25,038	25,178	25,054	25,1780	27,893	27,5150	26,0690
25,4760	24,944	25,148	25,116	25,2410	27,815	27,6910	25,9330
25,4910	25,038	25,069	24,975	25,1000	27,924	27,6700	25,1620
25,5140	24,961	24,975	25,413	25,1010	27,877	27,4370	24,9910
25,3550	25,022	24,836	25,038	25,1310	27,784	27,7260	24,9910
25,3440	25,085	24,710	24,975	25,3190	27,908	27,7830	24,9760
25,4160	24,975	24,913	25,070	25,1320	27,8	27,7000	24,9750
25,5540	25,569	25,007	24,960	25,2250	27,861	27,8320	24,9450
25,4280	24,929	25,038	25,085	25,1010	27,831	27,5100	25,0070
25,2850	24,975	25,460	25,038	25,1000	27,939	27,5130	24,9910
25,5220	24,820	25,693	25,100	25,4590	27,846	27,7060	25,0070
25,6170	25,173	25,241	25,194	25,2260	27,893	27,7260	25,0220
25,3090	25,051	25,022	25,085	25,1160	27,862	27,8110	24,9760

**Tabela D.1 (e)** – Dados para estudos dos tempos na performance do algoritmo

Tempos							
Tempo_Conf_18	Tempo_Conf_19	Tempo_Conf_20	Tempo_Conf_21	Tempo_Conf_22	Tempo_Conf_23	Tempo_Conf_24	Tempo_Conf_25
25,3970	26,030	25,760	26,5740	25,8290	27,8190	25,6650	25,7640
25,0380	25,873	25,606	26,3140	25,7160	27,7660	25,9250	25,8250
25,0380	25,696	25,746	26,5710	26,0210	27,8120	25,7500	25,7140
24,9760	25,664	25,758	26,3590	25,9150	27,8330	25,7130	25,8440
25,0380	25,927	25,955	26,4890	25,8000	27,5460	25,6880	25,2800
25,0220	25,954	25,954	26,3960	26,0560	27,6870	25,9590	25,3530
25,0380	25,905	25,995	26,5510	26,1590	27,8260	25,6590	25,3980
24,9600	25,813	25,921	26,2590	25,9430	27,9710	25,7530	25,3880
24,9290	25,800	25,889	26,5150	25,7440	27,6010	25,7880	25,7770
24,9920	25,822	25,859	26,7230	25,7870	27,8140	25,7200	25,7630
25,0220	25,891	26,066	26,5540	25,8460	27,8660	25,7360	25,8410
24,9910	25,899	26,011	26,4040	25,7870	27,7540	25,6750	25,7070
24,9450	25,683	26,069	26,2220	26,1020	27,6280	25,5610	25,5980
25,0070	25,377	25,888	26,5920	25,6680	27,8400	25,8100	25,9920
24,9440	25,504	25,903	26,3600	26,0740	27,6820	25,8540	25,7830
24,8980	25,522	25,868	26,4880	25,9410	27,3710	25,7930	25,7750
24,9130	25,791	25,916	26,5170	26,0540	27,0820	25,7780	25,6590
25,0690	25,998	26,128	26,3800	25,5760	27,2780	25,5780	25,8100
25,3656	26,035	25,726	26,4990	25,8080	27,6600	25,6620	25,7510
22,8230	25,775	25,818	26,6760	25,8020	27,9100	25,6810	25,7220
21,9810	26,032	25,887	26,5040	25,7430	27,7830	25,6400	25,8310
21,9330	25,944	25,927	26,3670	25,6730	27,6230	25,6410	25,7950
22,4950	26,180	25,906	26,3290	25,9880	27,8310	25,6340	25,9910
24,9450	25,466	25,787	25,7740	25,6430	26,7850	26,0610	25,8320
24,8980	25,631	25,904	25,7400	25,0070	27,4410	25,4100	25,7880

**Tabela D.1 (f)** – Dados para estudos dos tempos na performance do algoritmo

Tempos							
Tempo_Conf_18	Tempo_Conf_19	Tempo_Conf_20	Tempo_Conf_21	Tempo_Conf_22	Tempo_Conf_23	Tempo_Conf_24	Tempo_Conf_25
24,9910	25,740	25,905	25,7710	25,0220	27,9080	25,0220	25,9460
25,0380	25,670	25,929	25,9740	25,0070	27,8300	24,9130	25,7850
24,9910	25,553	25,884	25,7400	24,9600	27,9250	25,0380	25,7040
24,9910	25,358	25,910	25,8490	25,3080	27,9700	25,0070	25,8320
24,9440	25,429	26,036	25,8180	25,9380	27,9560	24,9910	25,6170
24,9450	25,433	25,786	25,7090	25,5240	27,8300	25,8130	25,7620
25,0070	25,556	25,984	25,8340	25,4450	27,9710	25,8770	25,7140
24,8970	25,313	26,113	25,6460	25,9590	27,9400	25,4940	25,5780
24,9760	25,268	25,810	25,8650	25,8300	27,9390	25,7330	25,8080
24,9910	25,376	26,003	25,7560	25,5910	27,7990	25,7210	25,7130
25,0230	25,322	26,054	25,6620	25,6410	27,8620	25,8220	25,8850
24,9910	25,455	26,118	25,7400	25,9020	27,8150	25,7840	26,0230
25,0690	25,779	25,979	25,7710	25,8900	27,9860	25,8410	25,8810
25,0540	25,771	26,089	25,8650	25,9830	28,1810	25,7540	25,8920
25,0530	25,615	25,780	25,7870	25,9740	27,6900	25,7280	26,0310
24,9920	25,750	25,959	26,8860	25,8800	27,8080	25,7230	25,6860
24,9750	25,677	25,972	26,3930	25,9930	27,4730	25,6900	25,9710
24,9760	26,025	25,972	26,2770	25,8960	27,7000	25,8080	25,8160
24,8660	25,739	26,237	26,2170	25,8610	27,7510	25,6570	25,7110
24,9920	25,940	25,664	26,4040	26,0180	27,6670	25,7510	25,8660
24,9910	25,726	25,895	26,3000	25,8210	27,7060	25,7800	25,8210
25,0850	25,720	25,862	26,5630	25,9660	27,7750	25,7020	25,8540
24,9440	25,821	26,112	26,4860	26,0630	27,9130	25,8730	25,8500
24,9130	25,718	26,016	26,5870	26,1700	27,8620	25,7030	25,8900
24,9290	25,734	26,009	26,6740	25,6070	27,8290	25,8180	25,5540

**Tabela D.1 (g)** – Dados para estudos dos tempos na performance do algoritmo

<b>Tempos</b>	
<b>Tempo_Conf_26</b>	<b>Tempo_da_oti</b>
25,9480	25,318
25,5300	25,054
25,3530	24,976
25,5240	25,085
25,6700	24,975
25,5640	24,945
25,8450	25,006
25,6110	25,007
25,6820	25,023
25,5610	25,038
25,7270	25,022
25,5350	25,038
25,3970	25,007
25,5340	24,96
25,5130	25,007
25,6940	25,038
25,7080	25,022
25,6780	24,976
25,4260	24,976
25,7610	24,975
25,6020	24,992
25,5680	25,022
25,4790	24,991
25,4950	25,007
25,5070	25,023



**Tabela D.1 (h)** – Dados para estudos dos tempos na performance do algoritmo

<b>Tempos</b>	
<b>Tempo_Conf_26</b>	<b>Tempo_da_oti</b>
25,6880	25,022
25,5260	25,007
25,5670	25,022
25,5660	25,007
25,6740	24,991
25,4710	24,976
25,5330	25,023
25,5580	25,022
25,5010	25,007
25,5550	25,007
25,2630	24,975
24,9600	25,376
24,9760	25,303
24,9290	26,333
24,9750	25,911
24,9450	25,179
25,1000	25,1
25,0540	24,835
24,9130	25,475
24,9600	24,929
24,9760	25,569
24,9290	25,365
24,9600	24,586
24,8970	25,724
24,9290	25,319

**Tabela D.2 (a)** – Dados para estudos das iterações na performance do algoritmo

N	Iterações								
	Iter_conf_1	Iter_conf_2	Iter_conf_3	Iter_conf_4	Iter_conf_5	Iter_conf_6	Iter_conf_7	Iter_conf_8	Iter_conf_9
1	1	3	183	56	54	77	51	35	61
2	1	16	170	50	48	47	43	38	58
3	1	30	13	32	40	47	40	47	47
4	1	52	65	118	54	24	52	36	21
5	1	110	96	58	67	32	43	41	33
6	1	131	110	47	52	24	73	45	53
7	1	1	94	48	134	47	24	57	60
8	1	2	100	98	46	43	53	42	90
9	1	75	69	85	68	58	54	53	54
10	1	1	88	65	55	43	61	39	47
11	1	24	30	49	42	52	45	47	57
12	1	1	69	43	41	52	47	7	21
13	1	82	97	65	57	48	50	45	65
14	1	3	136	45	30	60	43	52	66
15	1	22	63	107	41	43	44	46	49
16	1	130	129	54	44	54	37	43	50
17	1	156	45	50	50	83	47	52	68
18	1	73	81	102	41	44	83	53	46
19	1	130	50	49	35	51	38	105	47
20	1	71	109	39	51	41	48	45	61
21	1	157	54	43	38	46	56	52	49
22	1	152	144	90	47	151	42	100	53
23	1	55	40	38	56	66	46	38	45
24	1	3	134	128	52	51	50	51	134
25	1	129	162	47	42	54	50	48	41

**Tabela D.2 (b)** – Dados para estudos das iterações na performance do algoritmo

N	Iterações								
	Iter_conf_1	Iter_conf_2	Iter_conf_3	Iter_conf_4	Iter_conf_5	Iter_conf_6	Iter_conf_7	Iter_conf_8	Iter_conf_9
26	1	118	133	53	51	46	61	37	51
27	1	126	35	137	53	108	46	49	52
28	1	114	169	53	52	66	71	53	26
29	1	44	178	90	45	28	48	43	43
30	1	1	196	59	36	54	61	53	69
31	1	126	132	52	89	48	49	71	46
32	1	120	200	51	30	129	45	36	36
33	1	27	132	69	62	41	57	37	35
34	1	115	49	153	40	50	47	49	57
35	1	40	70	34	176	44	37	48	49
36	1	60	49	70	173	42	43	37	44
37	1	22	42	55	53	42	46	42	55
38	1	79	118	196	42	45	45	43	57
39	1	11	181	39	48	48	47	52	39
40	1	1	69	79	44	47	38	45	42
41	1	166	123	49	48	44	41	51	50
42	1	188	4	32	104	46	52	42	55
43	1	28	3	53	46	45	43	41	38
44	1	180	161	56	49	72	46	41	41
45	1	170	112	42	192	57	46	44	56
46	1	110	110	53	101	102	34	47	50
47	1	166	122	84	101	40	41	49	42
48	1	71	110	49	163	39	40	40	96
49	1	1	100	43	68	50	42	47	46
50	1	94	51	48	47	97	39	48	43

**Tabela D.2 (c)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_10	Iter_conf_11	Iter_conf_12	Iter_conf_13	Iter_conf_14	Iter_conf_15	Iter_conf_16	Iter_conf_17	Iter_conf_18
47	39	43	92	1	27	145	9	192
66	55	177	39	1	99	52	168	158
60	59	113	46	1	122	1	18	70
44	77	47	44	1	18	53	71	149
37	58	170	46	1	13	22	7	198
39	56	50	61	1	13	90	113	162
55	64	57	75	1	1	1	19	125
58	68	55	57	1	115	32	20	48
60	48	145	52	1	42	77	18	91
59	73	48	65	1	78	154	25	1
53	53	67	51	1	1	105	94	59
46	65	25	61	1	6	50	31	97
65	70	40	40	1	88	116	81	68
55	52	49	48	1	10	21	64	97
49	55	48	46	1	1	82	3	96
56	49	68	57	1	99	12	59	67
40	52	112	51	1	28	17	158	14
44	47	42	48	1	49	92	5	4
38	53	85	45	1	65	169	181	35
23	42	50	42	1	44	3	1	60
47	60	53	41	1	1	29	63	173
49	46	72	65	1	5	20	137	143
45	56	53	52	1	9	58	141	134

**Tabela D.2 (d)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_10	Iter_conf_11	Iter_conf_12	Iter_conf_13	Iter_conf_14	Iter_conf_15	Iter_conf_16	Iter_conf_17	Iter_conf_18
54	45	151	50	1	1	103	95	157
37	60	52	51	1	42	53	114	156
43	51	55	37	1	117	149	13	64
84	49	130	65	1	72	17	44	70
49	48	21	36	1	22	56	103	124
42	47	59	48	1	14	34	58	1
44	96	200	51	1	89	45	6	91
176	51	45	47	1	25	6	10	44
45	91	112	54	1	4	173	1	9
28	60	96	54	1	16	57	24	167
25	51	39	37	1	179	1	190	57
41	56	52	35	1	1	67	61	170
59	54	62	59	1	16	9	10	25
29	89	48	44	1	8	54	1	100
69	50	113	45	1	1	21	5	200
83	44	51	41	1	41	24	78	4
66	57	130	58	1	1	8	15	198
33	71	181	39	1	33	14	62	198
44	62	53	57	1	4	64	169	157
44	50	48	49	1	7	2	120	59
66	61	28	50	1	3	2	59	63
62	45	46	64	1	49	69	48	13
61	46	99	39	1	150	34	82	76
58	58	54	48	1	98	24	8	55
46	42	48	49	1	131	1	1	2
35	51	165	40	1	60	104	12	62
32	73	170	33	1	25	49	1	44

**Tabela D.2 (e)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_19	Iter_conf_20	Iter_conf_21	Iter_conf_22	Iter_conf_23	Iter_conf_24	Iter_conf_25	Iter_conf_26	Iter_oti
16	142	62	73	125	10	20	38	52
15	16	127	161	20	6	1	128	122
115	76	3	25	1	59	26	1	51
166	53	130	24	23	1	52	98	193
195	111	13	1	38	28	136	101	11
118	29	4	66	1	5	42	38	169
45	87	62	138	56	10	3	11	147
124	91	84	31	10	3	19	45	19
10	131	110	5	18	4	46	82	92
144	142	1	1	7	1	41	1	31
144	115	22	76	50	73	72	16	90
39	45	137	1	1	1	54	101	34
77	172	57	36	1	1	11	198	132
129	1	93	17	10	48	1	16	80
65	12	5	125	96	52	46	127	87
111	27	11	90	130	1	133	115	188
119	137	2	40	110	27	183	26	103
169	122	99	43	2	1	1	17	26
58	179	29	112	1	17	154	30	50
35	169	14	98	26	21	156	150	185
113	57	103	4	13	8	58	39	1
174	147	83	128	60	1	45	55	6
182	100	125	74	5	8	70	68	43
133	11	26	47	92	48	163	124	89
127	200	63	21	49	1	91	191	14

**Tabela D.2 (f)** – Dados para estudos das iterações na performance do algoritmo

Iterações								
Iter_conf_19	Iter_conf_20	Iter_conf_21	Iter_conf_22	Iter_conf_23	Iter_conf_24	Iter_conf_25	Iter_conf_26	Iter_oti
188	44	42	192	7	14	1	172	138
6	15	98	158	93	41	37	52	49
153	5	138	83	157	13	9	14	113
161	194	115	176	75	6	23	1	142
23	1	13	6	106	1	66	102	47
84	196	139	143	42	26	143	23	199
83	37	12	37	19	2	57	65	47
59	178	50	5	16	2	50	18	69
66	104	58	28	11	1	29	7	12
130	159	8	25	9	1	50	23	151
95	83	99	2	1	37	52	36	179
123	1	185	49	1	55	42	64	32
30	133	125	44	44	45	79	41	147
190	66	95	58	28	72	108	12	149
30	170	58	123	61	36	1	5	139
167	105	2	23	32	2	13	46	60
149	9	23	192	1	36	61	59	1
133	51	61	29	1	2	5	107	174
168	134	32	1	65	1	1	84	96
100	48	21	3	2	34	23	72	150
174	65	1	17	45	1	9	10	150
13	73	14	5	1	29	49	103	168
113	80	73	3	54	8	158	152	140
187	80	89	18	14	62	1	61	197
128	152	168	96	62	31	91	163	170