



**UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA CIVIL**

JEFFERSON WELLANO OLIVEIRA PINTO

**OTIMIZAÇÃO UTILIZANDO METAMODELOS E ALGORÍTMOS
EVOLUCIONÁRIOS**

Recife

2011

JEFFERSON WELLANO OLIVEIRA PINTO

**OTIMIZAÇÃO UTILIZANDO METAMODELOS E ALGORÍTMOS
EVOLUCIONÁRIOS**

Trabalho de Graduação
apresentado à Universidade
Federal de Pernambuco para
obtenção do título de
Engenheiro Civil.

Orientadora: Prof. Dra. Silvana M. B. Afonso

Recife
2011

Catálogo na fonte
Bibliotecário Marcos Aurélio Soares da Silva, CRB-4 / 1175

P659o Pinto, Jefferson Wellano Oliveira.

Otimização utilizando metamodelos e algoritmos evolucionários /
Jefferson Wellano Oliveira Pinto. - Recife: O Autor, 2011.

61 folhas : Il, Graf., Tabs.

Orientadora: Prof^a. Silvana M. B. Afonso.

TCC (Graduação) – Universidade Federal de Pernambuco. CTG.
Curso de Engenharia Civil, 2011.

Inclui Bibliografia.

1. Engenharia Civil. 2.Otimização estrutural. 3.Modelo Substituto.
4. Algoritmo de Enxame de Partículas. I.Afonso, Silvana
M.B.(Orientadora). II. Título.

A comissão examinadora abaixo assinada,
aprova o Trabalho de Conclusão de Curso.

**OTIMIZAÇÃO UTILIZANDO METAMODELOS E ALGORÍTMOS
EVOLUCIONÁRIOS**

elaborado por
Jefferson Wellano Oliveira Pinto.

como requisito para obtenção do grau de
Engenheiro Civil.

COMISSÃO EXAMINADORA:

Prof.^a. Dr.^a. Silvana Maria Bastos Afonso (Orientadora)

Prof. Dr. Afonso Henrique Sobreira de Oliveira

Eng. Msc. Leonardo Correia de Oliveira

RECIFE

2011

RESUMO

A otimização tem sido utilizada nos diversos ramos da engenharia para uma variedade de objetivos distintos e refere-se à obtenção da melhor solução para um determinado problema. Na prática esses problemas requerem um extensivo processamento computacional para realizar uma simples análise. Além disso, quando se deseja aperfeiçoar um problema, a obtenção do projeto ótimo pode tornar-se inviável uma vez que o procedimento de otimização requer sucessivas avaliações das funções e suas derivadas.

Entretanto, são apontados na literatura inúmeras alternativas para superar tais dificuldades. Uma delas refere-se à criação de modelos substitutos, que são construídos a partir da simplificação da função real complexa, através de aproximações locais utilizadas com a estratégia de Otimização por Aproximação Sequencial (SAO). Outra alternativa, é o uso de otimizadores que utilizam algoritmos metaheurísticos (que não requerem cálculos de gradientes no processo de otimização), que, geralmente, imitam algum fenômeno da natureza e são comumente chamados de algoritmos evolucionários. Neste contexto, será apresentada a Otimização utilizando o algoritmo de Enxame de Partículas (PSO).

Palavras-chave: Otimização estrutural. Modelo substituto. Otimização por aproximação sequencial. Algoritmo de Enxame de partículas.

ABSTRACT

The optimization has been used in several branches of engineering for a variety of different purposes and with the aim of obtaining the best solution for a given problem. In practice these problems require extensive computer processing to perform a single analysis. Also, when it is required to improve a problem, obtaining the optimal design could be impossible once the optimization procedure requires repeated function evaluations and their derivatives.

However, in the literature several alternatives are mentioned to overcome such difficulties. One of them refers to the creation of surrogate models, which are built from the simplification of the complex real function, through local approximations to be used by Sequential Approximation Optimization (SAO) strategy. Another alternative is the use of optimizers based on metaheuristics algorithms (algorithms that do not require gradient calculations in the optimization process), which often mimics some phenomenon of nature and are commonly called evolutionary algorithms. In this context it will be presented the Particle Swarm Optimization (PSO).

Keywords: Optimization. Surrogate models. Particle swarm optimization.

LISTA DE ILUSTRAÇÕES

Figura 3.1 - Problema de distribuição ótima de espessura numa viga (SILVA, E. C. N., 2010).....	17
Figura 4.1 – Amostragem do hipercubo latino com cinco subintervalos em cada variável X_1 e X_2	25
Figura 5.1 – Evolução do esquema SAO no problema de otimização	28
Figura 6.1 - Função Peaks	39
Figura 6.2 - Ponto ótimo da função Peaks	39
Figura 6.3 - Função Branin-Hoo.....	40
Figura 6.4 – Pontos ótimos da função Branin-Hoo	41
Figura 6.5 - Função Rastrigin	43
Figura 6.6 - Ponto ótimo da função Rastrigin.....	43
Figura 6.7 - Função Rastrigin aproximada com 40 pontos.....	43
Figura 6.8 - Função Rastrigin aproximada com 60 pontos.....	44
Figura 6.9 - Função Rastrigin aproximada com 80 pontos.....	44
Figura 6.10 - Função Ackley	45
Figura 6.11 - Ponto ótimo da função Ackley.....	46
Figura 6.12 - Função Ackley aproximada com 40 pontos.....	46
Figura 6.13 - Função Ackley aproximada com 60 pontos.....	47
Figura 6.14 - Função Ackley aproximada com 80 pontos.....	47
Figura 6.15 - Função Rosenbrock.....	49
Figura 6.16 - Ponto ótimo da função Rosenbrock	49
Figura 6.17 - Contornos do problema textbook. A região viável está na intersecção das duas restrições g_1 (sólido) e g_2 (pontilhado).....	51
Figura 6.18 - Exemplos clássicos de treliças bidimensionais.....	52
Figura 6.19 – Histórico de iterações da estratégia SAO.....	55
Figura 6.20 - Pórtico plano de madeira.....	56
Figura 6.21 - Função objetivo e restrições do pórtico de madeira.....	57

LISTA DE TABELAS

Tabela 3.1 Divisão dos problemas de Programação Matemática.....	20
Tabela 3.2 Etapas do algoritmo SQP.....	23
Tabela 5.1- Procedimentos da estratégia SAO	28
Tabela 5.2- Procedimentos da estratégia PSO	32
Tabela 6.1- Resultado da otimização da função Peaks.....	39
Tabela 6.2- Resultado da otimização da função Branin-Hoo	41
Tabela 6.3- Resultado da otimização da função Rastrigin	44
Tabela 6.4- Resultado da otimização da função Rastrigin	45
Tabela 6.5- Resultado da otimização da função Ackley.....	47
Tabela 6.6- Resultado da otimização da função Ackley.....	48
Tabela 6.7- Resultado da otimização da função Rosenbrock	49
Tabela 6.8- Resultado da otimização da função Rosenbrock	50
Tabela 6.9- Resultado da otimização da função Textbook.....	51
Tabela 6.10- Resultado da otimização da treliça de 10 barras.....	54
Tabela 6.11- Resultado da otimização da treliça de 64 barras.....	54
Tabela 6.12- Resultado da otimização do pórtico plano.....	57

LISTA DE ABREVIATURAS E SIGLAS

GA	Genetic Algorithm
LHS	Latin Hypercube Sampling
PS	Particle Swarming
PSO	Particle Swarm Optimization
RC	Região de Confiança
SA	Simulated Annealing
SAO	Sequential Approximation Optimization
SQP	Sequential Quadratic Programming

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Considerações Gerais.....	11
1.2	Justificativa	13
1.3	Objetivos.....	13
2	METODOLOGIA	14
3	OTIMIZAÇÃO	16
3.1	Introdução	16
3.2	Elementos para Formulação de um Problema de Otimização	16
3.2.1	Variáveis de Projeto	16
3.2.2	Função Objetivo	17
3.2.3	Restrições	17
3.2.4	Formulação Padrão do Problema de Otimização	18
3.3	Programação Matemática	19
3.3.1	Programação Quadrática Sequencial - SQP	21
4	MODELO SUBSTITUTO - KRIGAGEM	24
4.1	Geração de amostras para o modelo substituto.....	25
5	ESTRATÉGIAS DE OTIMIZAÇÃO	27
5.1	Otimização por Aproximação Sequencial (SAO).....	27
5.1.1	Formulação Matemática	27
5.1.2	Região de Confiança	29
5.1.3	Critérios de Convergência.....	29
5.2	Otimização via Enxame de Partículas (PSO)	31
5.2.1	Metodologia	31
5.2.2	Consideração de restrições no problema.....	33
5.2.3	Critérios de Convergência.....	34

6	EXEMPLOS	38
6.1	Introdução	38
6.2	Problemas considerando funções analíticas.....	38
6.2.1	Função Peaks.....	38
6.2.2	Branin-Hoo.....	40
6.2.3	Rastrigin	42
6.2.4	Ackley	45
6.2.5	Rosenbrock.....	48
6.2.6	Textbook.....	50
6.3	Problemas de Otimização Estrutural	51
6.3.1	Treliças	52
6.3.2	Pórtico	55
7	CONCLUSÕES	58
8	BIBLIOGRAFIA	59

1 INTRODUÇÃO

1.1 Considerações Gerais

O projeto ótimo requer a formulação adequada de um modelo de otimização (variáveis de projeto, função objetivo e restrições), bem como a utilização de um conjunto de técnicas numéricas bastante sofisticadas envolvendo o uso sequencial de simuladores, combinados com algoritmos de otimização. A experiência tem mostrado que o projeto ótimo de problemas reais da Engenharia leva a problemas de otimização complexos, com um grande número de variáveis de projeto e restrições em combinação com procedimentos de análises sofisticados (KEANE; NAIR, 2005).

Para tais problemas, porém, uma simples simulação pode demorar muitos minutos, horas ou mesmo dias para ser concluído. Como resultado, tarefas rotineiras, tais como otimização de projeto, exploração do espaço de projeto, análise da sensibilidade, quantificação de incerteza tornam-se impossíveis, uma vez que eles requerem muitas avaliações de função (simulação).

Métodos aproximados objetivando a construção de metamodelos (ou modelos substitutos) (KEANE; NAIR, 2005; GIUNTA;WATSON, 1988; FORRESTER et al., 2008; AFONSO et al., 2008) estão sendo cada vez mais utilizados nos projetos de engenharia, para superar/minimizar tais inconvenientes. Estas estratégias têm sido utilizadas objetivando uma resposta computacional rápida, e também para se obter um comportamento mais suave para uma dada resposta, livre de ruído numérico, que é um problema típico de funções avaliadas a partir de um procedimento numérico. Especificamente para aplicações em procedimentos de otimização que se baseiam no uso de funções provenientes de simuladores numéricos, este ultimo inconveniente citado deve ser evitado uma vez que podem tapear o processo de se determinar o ponto de ótimo.

O grande desafio científico nesta área é a geração de um modelo substituto (metamodelos) o mais preciso possível, utilizando-se o menor número possível de avaliações do modelo de alta fidelidade. As técnicas mais populares na construção de modelos substituto utilizando dados amostrais são superfícies de resposta (KEANE; NAIR, 2005; GIUNTA;WATSON, 1988; FORRESTER; SOBESTER; KEANE, 2008; AFONSO; HOROWITZ; WILLMERSDORF, 2008), método da base reduzida (AFONSO; HOROWITZ; WILLMERSDORF, 2008), krigagem (KEANE;

NAIR, 2005; FORRESTER; SOBESTER; KEANE, 2008; AFONSO et al., 2008) e redes neurais artificiais (NAIDU, 2004; KARTAM et al., 1997). Para a maioria dos problemas, a natureza da verdadeira função não é conhecida a priori, por isso não é claro qual modelo substituto será mais preciso.

A maioria dos procedimentos de otimização utiliza algoritmos que fazem uso de gradientes devido principalmente à eficiência computacional destes no processo de obtenção do ponto ótimo. No entanto, na última década algoritmos evolucionários, que não requerem cálculo de gradientes tem atraído grande atenção da comunidade científica (MICHALEWICZ; DASGUPTA, 1997; GOLDBERG, 1989; GELLAT et al., 1983; VENTER, 2002).

Dentre as alternativas existentes dessa classe de algoritmos (Genetic Algorithm – GA (GOLDBERG, 1989), Simulated Annealing - SA (VECCHI, 1983) e Particle Swarming - PS (VENTER, 2002)), o nosso grupo tem investido na implementação computacional/uso do algoritmo de aglomeração de partículas (PSO – Particle Swarm Optimization).

Embora as técnicas acima requeiram muito mais avaliações de funções para encontrar uma solução ótima, quando comparadas com algoritmos que utilizam o cálculo de gradientes, os algoritmos baseados em procedimentos evolucionários apresentam várias vantagens, a saber: facilidade de programação; não necessitam da garantia de continuidade nas funções envolvidas na definição do problema; mais adequado na determinação de um ótimo global ou próximo do global; e são adequados na solução de problemas discretos. A utilização de modelos substitutos, de baixo custo computacional, porém de precisão aferida, constitui uma alternativa bastante promissora no uso de tais algoritmos, evitando desta forma uma grande demanda de tempo computacional, característica inerente das metodologias evolucionárias acopladas a simuladores numéricos.

Nesta combinação de estratégias, as avaliações de funções (grande número) requeridas pelo algoritmo evolucionário não mais se dá através de simulações numéricas do problema real, e sim através de cálculos rápidos que consideram o uso dos modelos substitutos.

As aplicações deste trabalho dirigiram-se ao uso de funções empíricas multimodais, objetivando mostrar a potencialidade da metodologia em se determinar a solução global. Em seguida funções provenientes da simulação numérica de problemas estruturais (COOK, 1981) foram estudadas. Foram conduzidas

otimizações uniobjetivo considerando a abordagem PSO, no contexto do uso de modelos substitutos. As soluções obtidas pelo PSO foram comparadas com aquelas obtidas utilizando Otimização Sequencial Aproximada (SAO) (SILVA, 2009).

1.2 Justificativa

O presente projeto de conclusão de curso, na área de Engenharia Civil faz parte de um tema que está relacionado com os novos empreendimentos em implantação no estado por se tratar do desenvolvimento de ferramentas automáticas capazes de obter um projeto final otimizado.

1.3 Objetivos

O principal objetivo do presente trabalho é a aplicação de ferramentas baseadas no uso de estratégias de otimização que não necessitam de avaliação de gradientes e são elas: PSO, PSO com krigagem.

Os resultados foram comparados com técnicas que usam o gradiente, SAO com krigagem.

Ferramentas computacionais objetivando a otimização de projetos da Engenharia estrutural utilizando modelos substitutos às simulações numéricas e otimizadores da classe evolucionária; trabalho de conclusão de curso concluído e defendido.

2 METODOLOGIA

Especificamente, as ferramentas de otimização utilizando o algoritmo evolucionário PSO e o algoritmo sequencial SAO, no contexto do uso de modelos substitutos, foram construídas no ambiente MATLAB (MATLAB, 2010).

No que se segue, são apresentadas as ferramentas que foram utilizadas neste trabalho.

Ambiente de desenvolvimento: MATLAB.

Código de desenvolvimento: os estudos/implementações analisados foram conduzidos a partir do programa OPT-PSO (OLIVEIRA, 2008) que é formado por um conjunto de ferramentas computacionais produzidos no ambiente MATLAB e tem interfaces gráficas simplificadas e personalizadas. Da mesma forma, foram conduzidos estudos utilizando a implementação do algoritmo SAO no ambiente MATLAB, através do programa OPTRUSS desenvolvido por Afonso e Horowitz, 1998.

Procedimento de otimização: (1) Uso da estratégia evolucionária via aglomeração de partículas (VENTER, 2002; PARSOPOULOS; VRAHATIS, 2002; OLIVEIRA, 2008). Este método faz uso de um vetor velocidade para atualizar a posição de cada partícula. A posição de cada partícula é atualizada baseada no comportamento social da população de indivíduos. Parâmetros relacionados com a aptidão das partículas (que combina a função objetivo do problema e penaliza as restrições violadas) entram na avaliação da velocidade. (2) Uso da estratégia de otimização sequencial por aproximação (SILVA, 2009).

Metodologias de Aproximação: No que se refere à metodologia de aproximação existem vários aspectos importantes e que foram alvo de investigação, entre eles pode-se citar: (1) A escolha da estrutura das expressões aproximadas, modelos numéricos simplificados, entre outros; (2) O plano para os experimentos numéricos, isto é, a escolha de pontos para os quais a simulação numérica é conduzida. Foram consideradas aproximações baseadas na construção de modelos de krigagem (kriging) (KEANE; NAIR, 2005; GIUNTA; WATSON, 1988; FORRESTER et al., 2008; AFONSO et al., 2008; SIMPSON et al., 2001).

Aproximação via Krigagem: São técnicas baseadas no ajuste de superfície a partir de dados amostrais. A idéia central da técnica é a consideração de uma correlação entre os pontos. Neste sentido, um procedimento gaussiano é associado à metodologia. Devido a este aspecto, uma aproximação global sobre o espaço de

projeto investigado pode ser obtido com esta metodologia. As principais vantagens desta técnica são: a habilidade de tratar espaços de dados irregulares, a habilidade de lidar com funções com vários mínimos e máximos locais juntamente com a interpolação exata da resposta nos pontos das amostras. No processo de construção do modelo substituto, via esta técnica, uma atenção especial deve ser dada na estimativa dos parâmetros usados no modelo, bem como na escolha da estratégia para o plano dos experimentos (KEANE; NAIR, 2005; FORRESTER; SOBESTER; KEANE, 2008; AFONSO et al., 2008).

 Análise de casos: Funções analíticas de aspectos variados, problemas estruturais submetidos a carregamentos diversos.

3 OTIMIZAÇÃO

3.1 Introdução

Em um projeto de engenharia está envolvido, normalmente, um grande número de variáveis. Cabe ao projetista encontrar uma combinação para estas variáveis que resulte em um projeto o mais eficiente e, idealmente, o mais econômico possível. Normalmente, a determinação desta melhor configuração para os parâmetros de projeto depende fundamentalmente da experiência e da intuição do projetista. Técnicas de otimização podem ser utilizadas para auxiliar o projetista na determinação do projeto ótimo. O objetivo da otimização é encontrar, de acordo com o critério escolhido, a melhor solução entre as diversas combinações possíveis para as variáveis de projeto, mantendo a viabilidade do projeto, que está associada a restrições que devem ser satisfeitas (AFONSO, 1995).

3.2 Elementos para Formulação de um Problema de Otimização

Na formulação de um problema de otimização estão presentes os seguintes conceitos: variáveis de projeto, função objetivo, restrições e domínio viável e inviável que serão descritos a seguir.

3.2.1 Variáveis de Projeto

As variáveis de projeto são os parâmetros do problema que podem ser alterados para otimizar o sistema. Por exemplo, no caso de uma estrutura podem representar uma certa dimensão que será alterada, área da seção de uma viga, ou o valor de uma propriedade do material de que é feita (por exemplo, o módulo de elasticidade).

A escolha das variáveis de projeto é crítico para o sucesso da otimização. A Figura 3.1 ilustra a situação em que a escolha das variáveis de projeto influencia o resultado final da otimização. O resultado da Figura 3.1a é uma solução ótima local que não é interessante, e que é obtida devido à escolha das variáveis h_1 , h_2 , h_3 , h_4 e h_5 . Mudando-se a definição das variáveis para h_1 , Δh_2 , Δh_3 , Δh_4 , Δh_5 , como mostrado na Figura 3.1b outra solução ótima local é obtida, apresentando-se uma configuração mais interessante para ser produzida.

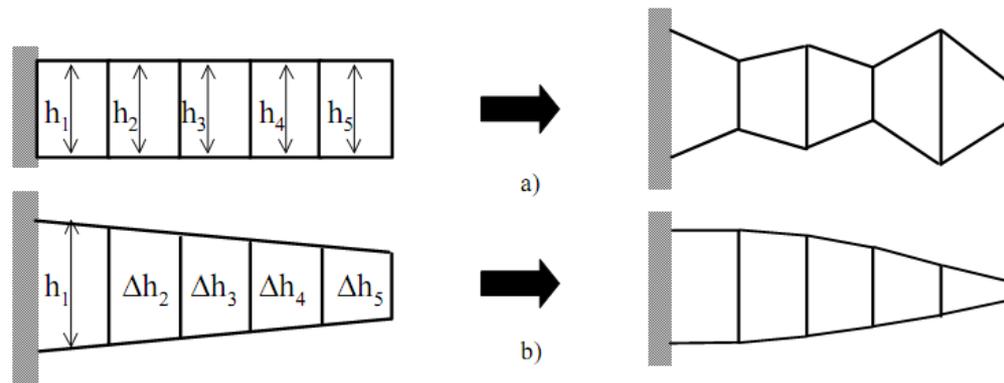


Figura 3.1 - Problema de distribuição ótima de espessura numa viga (SILVA, E. C. N., 2010)

3.2.2 Função Objetivo

A função objetivo deve quantificar o que queremos otimizar e será função das variáveis de projeto escolhidas. A função objetivo deve ser usada como uma medida da eficiência do projeto. Ela pode ser classificada em uniobjetivo, quando existe apenas um objetivo ou multiobjetivo (ou multicritério), quando se quer otimizar vários objetivos de uma só vez.

O sucesso da otimização vai depender também da formulação da função objetivo. Assim, é importante se perder um tempo para encontrar uma expressão matemática (deslocamento, frequência de ressonância, rigidez, volume, etc.) adequada que quantifique corretamente a eficiência do projeto.

É importante chamar atenção para algumas equivalências clássicas de função objetivo que podem tornar o problema matematicamente mais simples. Assim, maximizar f é a mesma coisa que minimizar $-f$ ou $1/f$ (a menos da singularidade em $f=0$), ou maximizar $k \times f$ (onde k é uma constante) e maximizar $|x|$ pode ser substituído por maximizar x^2 , o que evita singularidades na derivada da função objetivo.

3.2.3 Restrições

Essencialmente, as restrições são as limitações impostas para se obter a solução otimizada. São classificadas em três tipos: laterais, igualdade e desigualdade.

Considerando um conjunto de variáveis de projeto $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$, uma restrição lateral é do tipo:

$$x_{\min_i} \leq x_i \leq x_{\max_i} \quad i = 1, \dots, n \quad (3.1)$$

uma restrição de desigualdade é uma equação do tipo:

$$g_j(\mathbf{x}) \geq 0, \quad j = 1, \dots, n_g \quad (3.2)$$

e uma restrição de igualdade pode ser representada por uma equação do tipo:

$$h_k(\mathbf{x}) = 0, \quad k = 1, \dots, n_e \quad (3.3)$$

Outro ponto importante é a normalização das restrições. É muito comum existir restrições cujas ordens de grandeza dos valores são diferentes. Assim, enquanto o valor de uma restrição de tensão mecânica é da ordem de MPa, o valor de uma restrição de deslocamento é da ordem de centésimos de milímetro. A presença de valores tão distantes num algoritmo de otimização pode gerar problemas de estabilidade numérica, prejudicando o resultado final da otimização. Assim, deve-se normalizar a restrição como mostrado abaixo:

$$g_j(\mathbf{x}) \leq g_{\max_j} \Rightarrow \frac{g_j(\mathbf{x})}{g_{\max_j}} \leq 1 \Rightarrow \bar{g}_j(\mathbf{x}) - 1 \leq 0 \quad (3.4)$$

Com relação ao estado, a restrição é classificada em ativa e inativa. Uma restrição está ativa quando:

$$g_j(\mathbf{x}) = 0 \quad (3.5)$$

e uma restrição está inativa quando a igualdade acima não é satisfeita.

No final da otimização, espera-se que algumas das restrições estejam ativas, caso contrário, as que estão inativas não seriam, a princípio, necessárias no problema de otimização, pois não influenciam o problema. Por outro lado, existem restrições que se tornam ativas durante o processo de otimização e depois ficam inativas ao final, dessa forma é muito difícil saber de antemão, quais as restrições que influenciam ou não o resultado da otimização e assim, todas devem ser consideradas.

3.2.4 Formulação Padrão do Problema de Otimização

A formulação do problema de otimização pode ser descrita, matematicamente, pela seguinte expressão:

$$\begin{aligned}
\text{Minimize} \quad & f = f(\mathbf{x}) \\
\text{Sujeito à} \quad & g_i(\mathbf{x}) \leq g_i, \quad i = 1, 2, \dots, m_1 \\
& h_j(\mathbf{x}) = h_j, \quad j = 1, 2, \dots, m_2 \\
& x_l \leq \mathbf{x} \leq x_u, \quad k = 1, 2, \dots, n
\end{aligned} \tag{3.6}$$

Onde f é a função objetivo, g a restrição de desigualdade, h a restrição de igualdade, \mathbf{x} a variável de projeto, x_l e x_u são os limites inferior e superior da variável de projeto, m_1 e m_2 o número de restrições de desigualdade e de igualdade, respectivamente, e n é o número de variáveis de projeto.

3.3 Programação Matemática

A Programação Matemática pode ser considerada como a primeira linha de métodos para resolução de problemas de otimização. Ela trata o problema de forma iterativa, isto é, através de gradientes, funcionais, operações matriciais, etc. Por isto, normalmente, necessita de várias informações iniciais, tais como o espaço da região viável da função e a convexidade do problema (CASTRO, 2001).

Um ponto fraco da programação matemática é ainda não ter desenvolvido um método que possibilite a busca de soluções ótimas globais, ou seja, seus algoritmos podem convergir e se prender a qualquer solução ótima local, o que dependerá basicamente da solução de partida adotada (CASTRO, 2001).

Para resolver alguns tipos de problemas de otimização, e lidar com problemas restritos de várias variáveis é comum definir a função Lagrangeana $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ do problema original como segue (OLIVEIRA, 2008):

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{j=1}^1 \lambda_j g_j(\mathbf{x}) + \sum_{k=1}^m \lambda_{t+k} h_k(\mathbf{x}) \tag{3.7}$$

Onde $\boldsymbol{\lambda}$ é o vetor com os multiplicadores de Lagrange associados às restrições g_j e h_k no ponto \mathbf{x} . Pode ser mostrado (VANDERPLAATS, 1984; HAFTKA, e GÜRDAL, 1993) que a condição de mínimo local desta função na solução \mathbf{x}^* satisfaz as condições necessárias de Karush, Kuhn e Tucker (KKT):

1. Viabilidade:

$$\begin{cases} g_j(\mathbf{x}^*) \leq 0 & j = 1, \dots, l \\ h(\mathbf{x}^*) = 0, & k = 1, \dots, m \end{cases} \quad (3.8)$$

2. Estacionaridade:

$$\exists \boldsymbol{\lambda} \in \mathbb{R}^{l+m} \mid \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \nabla f(\mathbf{x}^*) + \sum_{j=1}^l \lambda_j \nabla g_j(\mathbf{x}^*) + \sum_{k=1}^m \lambda_{l+k} \nabla h_k(\mathbf{x}^*) = 0 \quad (3.9)$$

$$\text{com } \lambda_j \geq 0 \quad j = 1, \dots, l$$

3. Complementaridade:

$$\lambda_j g_j(\mathbf{x}^*) = 0, \quad j = 1, \dots, l \quad (3.10)$$

Os algoritmos de Programação Matemática são iterativos, onde, através da especificação de um vetor inicial de variáveis \mathbf{x}^0 , uma sequência de pontos é gerada e, se bem sucedida, converge para o ponto solução \mathbf{x}^* . A forma mais comum de atualização das variáveis, para este procedimento iterativo, é dada pela equação:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^* S^{k+1} \quad (3.11)$$

onde S é o vetor da direção de busca e α^* representa o tamanho do passo na direção de S (VANDERPLAATS, 1984; NOCEDAL; WRIGHT, 2000).

A seleção do algoritmo de otimização depende fundamentalmente do problema envolvido. Isto é importante para se obter uma otimização confiável e um alto nível de eficiência (tempo de computação, taxa de convergência). Em geral, o tipo de problema é determinado pelas características das funções objetivo e das restrições envolvidas (AFONSO, 1995). A classificação mais simples é apresentada na Tabela 3.1:

Tabela 3.1 Divisão dos problemas de Programação Matemática.

Tipo de Otimização	$f(\mathbf{x})$	$h(\mathbf{x}), g(\mathbf{x})$
Programação Linear	linear	linear
Programação Quadrática	quadrática	linear
Programação Não-linear	não-linear	não-linear

As formas mais complexas de programação matemática envolvem funções não-lineares. O método de programação não-linear é agrupado em duas categorias:

1. *Métodos indiretos*: envolvem técnicas para solucionar o problema original de otimização, definido pela equação (3.6), em um problema de otimização irrestrito equivalente através de algum artifício. Alguns exemplos de métodos indiretos são:

- Métodos de penalidade ou barreira;
- Método dual;
- Lagrangeano aumentado

2. *Métodos diretos*: Esses métodos resolvem diretamente o problema original em vez de convertê-lo num problema equivalente. Alguns exemplos de métodos diretos são:

- Método de Newton
- Método do Gradiente Conjugado
- Métodos Quasi-Newton
- Programação Quadrática Sequencial (SQP)

Neste trabalho, nas estratégias que realizam a busca local, utiliza-se o algoritmo de Programação Quadrática Sequencial (SQP – Sequential Quadratic Programming), descrito a seguir.

3.3.1 Programação Quadrática Sequencial - SQP

O SQP surgiu para tratar de problemas de otimização restrita. A idéia básica desse método consiste em resolver uma sequência de subproblemas quadráticos a cada iteração e definir a direção de busca para ser a solução deste subproblema (NOCEDAL; WRIGHT, 2000). Essa sequência deve convergir para um ponto de mínimo \mathbf{x}^* .

O algoritmo SQP utiliza a informação de segunda ordem da função a otimizar, assim sendo, considera-se uma expansão de segunda ordem da série de Taylor da função $f(\mathbf{x})$ em $\bar{\mathbf{x}}$, de acordo com:

$$f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \mathbf{H}(\bar{\mathbf{x}}) \mathbf{d} \quad (3.12)$$

onde $\mathbf{d} = \mathbf{x} - \bar{\mathbf{x}}$ é a direção de busca, $\nabla f(\bar{\mathbf{x}})$ é o vetor gradiente de f em $\bar{\mathbf{x}}$, de quem os elementos são definidos por:

$$\nabla f_i = \frac{\partial f}{\partial x_i} \quad (3.13)$$

e $\mathbf{H}(\bar{\mathbf{x}})$ é a matriz Hessiana, de quem os elementos H_{ij} são definidos por:

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} \quad (3.14)$$

É assumido que o vetor gradiente e a matriz Hessiana, na equação (3.12) irão proporcionar uma boa aproximação da função verdadeira, especialmente se $\bar{\mathbf{x}}$ estiver próximo ao ponto ótimo.

A função objetivo do subproblema a ser resolvido é tal que o coeficiente do termo linear é formado pelo gradiente da função objetivo do problema principal enquanto para o termo quadrático, uma aproximação da Hessiana da função Lagrangeana (esquema BFGS – método Broyden-Fletcher-Goldfarb-Shanno) (VANDERPLAATS, 1984) do problema principal é usada. Uma descrição detalhada deste algoritmo pode ser encontrada em Powell (1978) e Vanderplaats (1984). Em suma, são descritos na Tabela 3.2 as principais etapas envolvidas no algoritmo SQP convencional.

Tabela 3.2 Etapas do algoritmo SQP.

Etapa	Descrição
SQP 1.	Estabelecer uma solução inicial \mathbf{x}_0 ;
SQP 2.	Configurar uma aproximação inicial para a matriz Hessiana dos termos quadráticos da função objetivo;
SQP 3.	Resolver o subproblema para encontrar a direção de busca \mathbf{d} ;
SQP 4.	Realizar uma busca linear para determinar o tamanho do passo α na direção \mathbf{d} ;
SQP 5.	Atualizar a solução, remetendo-a para a posição indicada;
SQP 6.	Checar a convergência do algoritmo SQP:
6.1.	Se o mínimo local for encontrado: o processo termina.
6.2.	Caso contrário: atualiza-se a matriz Hessiana via o esquema BFGS e volta para o passo SQP 3.

4 MODELO SUBSTITUTO - KRIGAGEM

A teoria de krigagem foi desenvolvida pelo matemático francês Georges Matheron a partir dos trabalhos de Daniel G. Krige (VALENTE, 1982). Muito usado em geoestatística, parte do princípio que os pontos em um dado espaço de projeto são correlacionados.

O modelo de krigagem (JONES et al, 1998) pode ser escrito como:

$$f(x) = \sum_{j=1}^k \beta_j N_j(x) + Z(x) \quad (4.1)$$

Onde o primeiro termo é uma regressão linear dos dados com k regressores, e $Z(x)$ (o erro) é uma função aleatória (em geral segue uma distribuição gaussiana normal), com média zero, variância σ^2 , e covariância não nula.

O primeiro termo é responsável pelo aspecto global ou tendência ao longo do espaço de projeto, ao passo que $Z(x)$ é responsável por criar um desvio 'localizado' em relação ao modelo global. A abordagem tradicional é chamada de krigagem ordinária, em que uma função de ordem zero (constante) é empregada para o primeiro termo da equação (4.1).

A matriz de covariância $Z(x)$ é dada por

$$\text{cov}[Z(x^i), Z(x^j)] = \sigma^2 \mathbf{R}[R(x^i, x^j)] \quad (4.2)$$

onde \mathbf{R} é a matriz de correlação $m \times m$ com valores unitários ao longo da diagonal e $R(x^i, x^j)$ é função de correlação entre qualquer dos dois m pontos de dados amostrados x^i e x^j . Diferentes formas de função de correlação podem ser empregadas, como: exponencial, exponencial generalizada, Gaussiana, entre outros.

Neste trabalho considerou-se uma função de correlação Gaussiana da forma:

$$R(x^i, x^j) = \exp\left(-\sum_{k=1}^{n_v} \theta_k |x_k^i - x_k^j|^2\right) \quad (4.3)$$

Onde n_v é o número total de variáveis, θ_k os parâmetros de correlação utilizados para ajustar o modelo. Outros tipos de função de correlação também podem ser empregados, no entanto, a função de correlação Gaussiana é mais

utilizada na literatura (AFONSO, 2008). A expressão do preditor para este modelo é obtida através da minimização do erro médio quadrático (AFONSO, 2008).

4.1 Geração de amostras para o modelo substituto

O primeiro passo para a construção de um modelo substituto baseado no ajuste de dados é a geração de uma amostra de pontos. Estes são locais no espaço de projeto em que os valores da resposta dos modelos de alta fidelidade serão calculados para construir o modelo aproximado. A seleção da amostra é uma etapa muito importante, uma vez que para casos onde a avaliação da função requer um alto custo computacional, deve-se procurar um plano de amostragem eficaz, o que significa o número mínimo de pontos que garantirão um modelo substituto com boa precisão.

Neste trabalho foi utilizada a amostragem do hipercubo latino (LHS – *Latin Hypercube Sampling*) (FORRESTER et al, 2008; GIUNTA, 2002). Para obter uma amostra LHS, o intervalo de cada dimensão é dividido em p subintervalos de igual probabilidade. Para um domínio de projeto com n dimensões, este particionamento resulta num total de p^n subintervalos no domínio. Em seguida, p pontos são selecionadas aleatoriamente no domínio de projeto obedecendo a certas restrições, tais como: cada ponto é aleatoriamente colocado dentro de um subintervalo do domínio e para cada projeção unidimensional deste ponto só haverá um e apenas um ponto em cada subintervalo.

Essa explicação é facilmente compreendida através da Figura 4.1, que apresenta uma possível amostra de cinco pontos no domínio 2D (x_1, x_2) .

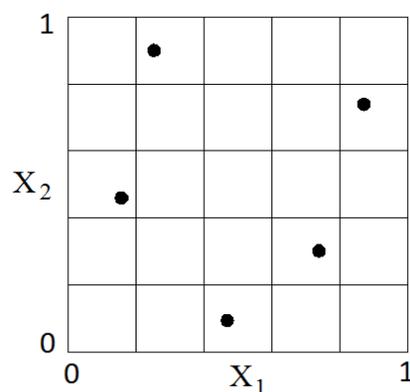


Figura 4.1 – Amostragem do hipercubo latino com cinco subintervalos em cada variável X_1 e X_2

A aleatoriedade inerente ao processo significa que existe mais de uma possibilidade do arranjo de amostragem atender aos critérios do LHS. Como a amostragem LHS é de natureza estocástica, é aconselhável executar tal programa várias vezes e selecionar a melhor amostra para uso. Para selecionar a melhor amostra LHS, é determinado Δ como (KEANE e NAIR, 2005):

$$\Delta = \sum_{i=1}^{m-1} \left(\sum_{j=i+1}^m \frac{1}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \right) \quad (4.4)$$

onde m é o número total pontos da amostra. A amostra LHS que dá o valor mínimo para Δ é a amostra selecionada.

5 ESTRATÉGIAS DE OTIMIZAÇÃO

5.1 Otimização por Aproximação Sequencial (SAO)

Na Otimização por Aproximação Sequencial (SAO), o algoritmo de otimização vai operar sobre um modelo substituto, de baixo custo computacional, ao invés de se fazer uso das respostas obtidas da simulação real. A metodologia SAO decompõe o problema original de otimização em sequências de subproblemas de otimização, confinados dentro de uma sub-região do espaço de projeto inicial.

5.1.1 Formulação Matemática

Um esquema de Região de Confiança (RC) é usado para atualizar as variáveis de projeto para cada subproblema (iteração do SAO). Matematicamente, cada subproblema k pode ser definido como:

$$\begin{aligned}
 &\text{Minimize} && \hat{f}^k(\mathbf{x}) \\
 &\text{Sujeito à:} && \hat{g}_i^k(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\
 &&& x_l \leq x_l^k \leq \mathbf{x} \leq x_u^k \leq x_l, \quad k = 0, 1, 2, \dots, k_{\max} \\
 &\text{Onde} && x_l^k = x_c^k - \frac{\Delta^k}{2} \qquad x_u^k = x_c^k + \frac{\Delta^k}{2}
 \end{aligned} \tag{5.1}$$

Nas equações acima, $\hat{f}^k(\mathbf{x})$ e $\hat{g}^k(\mathbf{x})$ são respectivamente a funções objetivo e restrição baseadas no modelo substituto. Sendo x_c^k o ponto central da RC, Δ^k é o tamanho da RC e x_l^k , x_u^k são respectivamente os limites inferiores e superiores da variável de projeto na iteração k do SAO (GIUNTA e ELDRED, 2000).

Na Tabela 5.1 são apresentados os principais procedimentos implementados na estratégia SAO (SILVA, 2009).

Tabela 5.1- Procedimentos da estratégia SAO

Etapas SAO	Descrição
1.	Assumir o tamanho e ponto inicial da região de confiança;
2.	Gerar amostras no interior da região de confiança;
3.	Computar as funções objetivo e restrições reais nas amostras;
4.	Construir as funções substitutas (aproximadas) para as funções objetivo e restrições;
5.	Otimizar o problema usando as funções aproximadas, onde as variáveis de projeto são limitadas pelas fronteiras da sub-região;
6.	Recalcular as funções objetivo e restrições reais no ponto ótimo encontrado;
7.	Checar os critérios de convergência internos do otimizador:
7.1.	Se alcançada: o problema da iteração z está resolvido, continue com o processo.
7.2.	Caso contrário: a função aproximada não é adequada, o processo é finalizado.
8.	Manter/reduzir/aumentar a nova sub-região de acordo com o esquema da região de confiança;
9.	Checar os critérios de convergência global da estratégia SAO;
9.1.	Se alcançada: o projeto ótimo é obtido, o processo é finalizado com sucesso.
9.2.	Caso contrário: voltar para o passo SAO 3.

A Figura 5.1 ilustra um problema de otimização utilizando o processo iterativo e sequencial do esquema SAO. Os quadrados denotam a sub-região de cada subproblema do SAO. Os pontos dentro de cada sub-região são caracterizados pelas amostras de dados, que são usadas para gerar um modelo substituto, no qual o algoritmo de otimização vai atuar, ao invés do problema original (ELDRED et al., 2009).

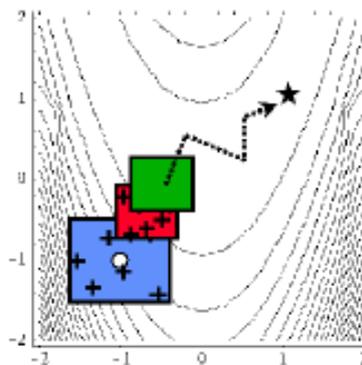


Figura 5.1 – Evolução do esquema SAO no problema de otimização

5.1.2 Região de Confiança

Para a atualização do tamanho da região de confiança (RC) Δ^k para cada subproblema de otimização, foi considerado o termo de aceitação da função objetivo ρ^k que controla o tamanho da RC (GUINTA e ELDERED, 2000). Este parâmetro demonstra a precisão da função substituta no ponto ótimo x_*^k e pode ser calculado como:

$$\rho^k = \frac{f(x_c^k) - f(x_*^k)}{\hat{f}(x_c^k) - \hat{f}(x_*^k)}, \quad (5.2)$$

Para o próximo tamanho da RC a atualização segue dessa forma:

$$\begin{aligned} \Delta^{k+1} &= 0.5\Delta^k, & \text{se } \rho^k \leq 0, \\ &= 0.5\Delta^k, & \text{se } 0 < \rho^k \leq 0.25, \\ &= \Delta^k, & \text{se } 0.25 < \rho^k < 0.75 \text{ ou } \rho^k > 1.25, \\ &= 2\Delta^k, & \text{se } 0.75 \leq \rho^k \leq 1.25. \end{aligned} \quad (5.3)$$

A próxima iteração x_c^{k+1} é obtida de acordo com:

$$\begin{aligned} x_c^{k+1} &= x_*^k, & \text{se } \rho^k > 0, \\ &= x_c^k, & \text{se } \rho^k \leq 0. \end{aligned} \quad (5.4)$$

5.1.3 Critérios de Convergência

Dois critérios de convergência são adotados quando se considera o esquema de otimização sequencial aproximada. O primeiro deles é definido a partir da convergência local durante o processo de otimização do subproblema, o qual é confinado a uma sub-região do espaço de projeto. O ponto ótimo será aceito segundo os critérios internos de convergência do otimizador, ou seja, da função FMINCON do MATLAB (Mathworks, 2009). Tal função foi escolhida, pois contém o algoritmo SQP já implementado. Um dos critérios descritos abaixo deve ser atingido de modo que a solução seja considerada como solução do subproblema. Os critérios são:

- A medida da otimalidade de primeira ordem é menor que a tolerância da função (TolFun) e a máxima violação das restrições são menores que a tolerância (TolCon) da mesma;

- O melhoramento do ponto ótimo \mathbf{x}_*^k obtido foi menor que a tolerância (TolX) do mesmo;
- O melhoramento da função objetivo aproximada $\hat{f}(\mathbf{x}_*^k)$ foi menor que a tolerância (TolFun) da mesma;
- O número máximo de iterações é maior que o número máximo de iterações permitido (MaxIter).

onde, os termos TolFun, TolCon, TolX e MaxIter são as tolerâncias internas da função FMINCON do MATLAB.

Os critérios acima descritos são necessários durante cada iteração k da estratégia SAO. Contudo, um segundo critério de convergência é necessário para avaliar a convergência global durante as iterações do processo de otimização sequencial. O término do processo sequencial é concluído se um dos seguintes critérios for atingido:

- Após um número prescrito, três para os problemas aqui considerados, de subsequentes iterações do processo SAO, o resultado da função aproximada obtido não demonstrou nenhuma melhoria,
- A equação (5.5) busca representar matematicamente a condição de convergência adotada, onde a função real e a aproximada são comparadas.

$$tol = \left| \frac{f(\mathbf{x}_*^k) - \hat{f}(\mathbf{x}_*^k)}{f(\mathbf{x}_*^k)} \right| \leq 10^{-4} \quad (5.5)$$

O valor da tolerância acima assumido representa um índice de confiabilidade da função aproximada $\hat{f}(\mathbf{x}_*^k)$ com relação à função real $f(\mathbf{x}_*^k)$ calculada no ponto ótimo \mathbf{x}_*^k igual a 99.99%.

Ao atingir um dos critérios acima, o processo de otimização sequencial é finalizado e o último ponto ótimo obtido é considerado como solução ótima do problema (SILVA, 2009).

5.2 Otimização via Enxame de Partículas (PSO)

O algoritmo do PSO (Particle Swarm Optimization) foi originalmente desenvolvido por um psicólogo e um engenheiro elétrico em 1995 (Kennedy e Eberhart, 1995) e foi inspirado no comportamento de revoada visto em muitas espécies de pássaros.

No modelo proposto (POMEROY, 2003), os pássaros são atraídos para uma área de pouso, um poleiro. Na simulação, os pássaros começariam a explorar o ambiente sem um destino em particular e espontaneamente uma revoada estaria formada até que um deles encontre uma área de pouso. As regras que modelam o comportamento dos pássaros são simples: os pássaros escolhem a direção e a velocidade do voo (essencialmente, cada pássaro tenta permanecer próximo aos outros enquanto também tenta não colidir com os mesmos); um dos pássaros se desvia do restante do bando com o intuito de pousar num poleiro, o que acarreta num movimento dos pássaros mais próximos para a mesma direção; quando esses pássaros descobrem um poleiro, eles pousam conduzindo o restante do bando para o mesmo local.

Encontrar um poleiro é análogo a encontrar uma solução em um campo de possíveis soluções, e a maneira que uma das aves influencia seus vizinhos em direção a ele, aumentando a probabilidade dos mesmos encontrá-lo, está ligada a cognição social da mente (a mente e, portanto, a inteligência é social) (Oliveira, 2008).

Logo, o algoritmo PSO busca um equilíbrio entre a individualidade e a socialização (características que influenciam o comportamento). De maneira ideal, se quer que os indivíduos, aqui chamados de partículas, prefiram ser individualistas (semelhante ao pássaros que não querem colidir uns aos outros) e também queiram saber onde boas soluções tem sido encontradas pelos outros, então se pode "aprender com o que o outros fizeram".

5.2.1 Metodologia

O roteiro de um algoritmo básico do PSO é descrito na Tabela 5.2

Tabela 5.2- Procedimentos da estratégia PSO

Etapas PSO	Descrição
1.	A população é inicializada com uma distribuição aleatória das partículas por todo o espaço de projeto;
2.	O vetor velocidade para cada partícula é calculado;
3.	A posição de cada partícula é atualizada usando a posição prévia e o vetor velocidade atualizado;
4.	Checa-se a convergência;
5.	Se a convergência foi atingida, o processo é interrompido; caso contrário retorna ao passo 2.

No PSO, cada partícula candidata a solução do problema é representada por um ponto no espaço de projeto e uma velocidade. Durante o processo iterativo, esta velocidade é atualizada, fazendo com que a partícula se mova dentro do espaço de projeto, sendo este processo de atualização influenciado por decisões individuais (da partícula) e sociais (do enxame). A atualização da posição, mostrada a seguir, segue um esquema baseado na equação cinemática da função horária do espaço em um movimento uniforme.

$$p_{t+1}^i = p_t^i + v_{t+1}^i \Delta t \quad (5.6)$$

Como visto na equação (5.6), a posição p da partícula i na iteração $t+1$ é atualizada a partir da sua posição na iteração anterior, t , somada a sua velocidade v na iteração $t+1$. O fator Δt representa o tempo, que no PSO é representado pelas iterações, o que torna o valor de Δt unitário. A formulação para obtenção da velocidade v_{t+1}^i (SHI e EBERHART, 1998) é mostrada a seguir.

$$v_{t+1}^i = wv_t^i + c_1r_1(pb_{best}^i - p_t^i) + c_2r_2(g_{best} - p_t^i) \quad (5.7)$$

Onde v_t^i é a velocidade da partícula i na iteração t , pb_{best}^i representa a melhor posição já encontrada pela partícula i , g_{best} é a melhor posição encontrada pelo enxame, r_1 e r_2 são números aleatórios entre zero e um. O parâmetro w é denominado inércia e influencia fortemente no comportamento do algoritmo: valores altos facilitam um comportamento mais global enquanto valores baixos um comportamento mais local. Os termos c_1 e c_2 são denominados parâmetros de confiança e representam quanta confiança a partícula tem nela mesma e na

população, respectivamente. Valores típicos para esses parâmetros são $c_1 = c_2 = 2$ (OLIVEIRA, 2008).

5.2.2 Consideração de restrições no problema

A estratégia básica do PSO foi definida para problemas irrestritos e como a maioria dos problemas práticos apresenta restrições, se faz necessário acrescentar ao algoritmo a capacidade de lidar com tais funções. As metodologias utilizadas para resolver tais problemas consistem na transformação dos problemas restritos em problemas irrestritos. Foi decidido lidar com problemas restritos através do método das funções de penalidade. Uma função de penalidade quadrática foi considerada. Esta técnica é frequentemente usada no algoritmo genético (VENTER e SOBIESZCZANSKI-SOBIESKI, 2002). Em tais casos, a função objetivo é penalizada como mostrado na equação (5.8).

$$P(\mathbf{x}) = \bar{f}(\mathbf{x}) + \mu\alpha(\mathbf{x}) \quad (5.8)$$

A função objetivo é penalizada quando uma ou mais de uma das restrições é violada. No esquema anterior $P(\mathbf{x})$ é a nova função objetivo (penalizada), $\bar{f}(\mathbf{x})$ é a função objetivo normalizada do problema, como é apresentada na equação (5.9), μ é o parâmetro de penalidade (é adotado $\mu = 10^8$) e $\alpha(\mathbf{x})$ é dado através da equação (5.10).

$$\bar{f}(\mathbf{x}) = \frac{f(\mathbf{x})}{f(\mathbf{x}_0)} \quad (5.9)$$

$$\alpha(\mathbf{x}) = \sum_{i=1}^{nr} \max [0; g_i(\mathbf{x})]^2 \quad (5.10)$$

Na equação anterior, $g(\mathbf{x})$ representa o conjunto de todas as restrições do problema em questão.

Quando uma partícula viola as restrições, além da construção da função de penalidade, o vetor velocidade da partícula que violou alguma das restrições deve ser atualizado de forma a garantir a volta da mesma para o espaço viável do projeto. Uma alternativa apresentada é calcular uma direção viável, que seria uma direção que induziria a partícula a melhorar a função objetivo dentro do espaço viável do projeto. Infelizmente, isso requeria informações do gradiente de cada partícula. Um

dos atrativos do algoritmo PSO é o fato de não serem necessárias informações sobre gradientes.

No lugar das informações sobre o gradiente é proposta uma modificação simples na atualização do vetor velocidade para partículas com restrições violadas. Considerando uma partícula com pelo menos uma restrição violada na iteração t , a velocidade dessa partícula na iteração t é reiniciada para o vetor nulo, e o vetor velocidade na iteração $t+1$ (VENTER e SOBIESZCZANSKI-SOBIESKI, 2002) é obtido através da equação

$$v_{t+1}^i = c_1 r_1 (pbest^i - p_t^i) + c_2 r_2 (gbest - p_t^i) \quad (5.11)$$

A velocidade da partícula i na iteração $t+1$ é então influenciada apenas pela melhor posição já encontrada por si mesma e pela melhor posição encontrada pela população até a presente iteração. Na maioria dos casos o novo vetor velocidade apontará em direção à região viável do espaço de projeto. O resultado é o retorno, da partícula com restrições violadas, para região viável do projeto.

Em outra situação na consideração de restrições no problema, as partículas são capazes de reconhecer posições violadas e dão preferência a soluções viáveis. Neste segundo esquema, as partículas são comparadas seguindo os seguintes critérios:

- Se duas partículas são viáveis, escolhemos a que tenha o melhor valor da função objetivo;
- Entre uma partícula viável e outra inviável, escolhemos sempre a viável;
- Se duas partículas estão inviáveis, escolhemos aquela com o menor índice de inviabilidade.

Esta técnica é utilizada na seleção de $pbest$ e $gbest$. Cada partícula calcula um índice de inviabilidade que identifica o quão inviável a partícula está. Caso a mesma se encontre em uma posição viável, o valor do índice é zero.

5.2.3 Critérios de Convergência

A essência de base populacional do PSO permite a concepção de medidas que dão uma idéia de agregação que as partículas tem atingido até um determinado momento, que, por sua vez, dá uma idéia de convergência ao método.

Apesar do fato de que a agregação de partículas não necessariamente significa a convergência do método, mais melhorias na melhor solução encontrada, após a reunião das partículas, são improváveis devido à perda de diversidade.

Seguindo a metáfora de inspiração na psicologia social do método, isso indicaria que os indivíduos na população teriam chegado a um acordo, ou ainda, que a diversidade de crenças está perdida.

Para condições de finalização do processo de otimização são considerados dois procedimentos que são realizados com frequências diferentes. O primeiro procedimento é relacionado à agregação (realizado com uma frequência maior) e o segundo procedimento é relacionado ao monitoramento da melhor aptidão da população (realizado com uma frequência menor).

5.2.3.1 Primeira condição de finalização

Como citado anteriormente, a primeira condição de finalização trata da agregação de partículas. Essa condição é dividida em duas relações:

Relação 1: essa relação avalia a razão da diferença entre a aptidão de cada partícula e a melhor aptidão encontrada pelo enxame nas últimas 100 iterações em relação à diferença entre a pior e a melhor aptidão encontrada, na iteração onde é realizada a verificação (INNOCENTE, 2006). Se essa razão for menor do que um valor determinado, a probabilidade de convergência aumenta. O esquema pode ser representado da seguinte forma:

$$\frac{\sum_{k=t-99}^t \frac{\sum_{i=1}^m (c_k^i - cgbest_k)}{m}}{100 \times (cgworst_t - cgbest_t)} = \frac{\sum_{k=t-99}^t (\bar{c}_k - cgbest_k)}{100 \times (cgworst_t - cgbest_t)} \leq 1 \times 10^{-12} \quad (5.12)$$

onde \bar{c}_k representa o valor médio das aptidões de todas as partículas, $cgbest_k$ é a melhor aptidão encontrada na população e $cgworst_k$ é a pior aptidão encontrada na população na iteração k ;

Relação 2: essa relação avalia a razão da diferença entre a atual melhor aptidão encontrada e a melhor aptidão encontrada na 100ª iteração precedente em relação à diferença entre a pior e a melhor aptidão encontrada na iteração onde é

realizada a verificação (INNOCENTE, 2006). O esquema adotado é representado da seguinte forma:

$$\frac{abs(cgworst_t - cgbest_{t-99})}{100 \times (cgworst_t - cgbest_t)} = \frac{cgbest_{t-99} - cgbest_t}{100 \times (cgworst_t - cgbest_t)} \leq 1 \times 10^{-15} \quad (5.13)$$

onde $cgbest_{t-99}$ é o valor da melhor aptidão encontrada na iteração $t - 99$.

Apesar de a primeira condição ser avaliada com mais frequência, ela não é avaliada em todas as iterações. Para realização das verificações da primeira condição de finalização, é necessário que:

- tenha acontecido, no mínimo, 10% do máximo de iterações a serem realizadas;
- seja dado um intervalo entre duas verificações consecutivas.

Nos estudos realizados, foi considerado um intervalo de 50 iterações entre duas avaliações consecutivas.

5.2.3.2 Segunda condição de finalização

A segunda condição de finalização trata da verificação das mudanças observadas na melhor aptidão da população. Nessa condição, é conferido se ocorreu alguma melhoria significativa na melhor função aptidão encontrada até o momento, num determinado número de iterações.

A verificação segue o seguinte esquema:

$$cgbest_{t-0.25t_{\max}} - cgbest_t = 0 \quad (5.14)$$

Onde t_{\max} é o número máximo de iterações a serem realizadas e $cgbest_{t-0.25t_{\max}}$ é a melhor aptidão encontrada na iteração $t - 0.25t_{\max}$.

Essa condição é avaliada com uma frequência menor, como já mencionado antes, e do mesmo modo que a primeira condição de finalização, alguns critérios precisam ser atendidos antes da realização da segunda avaliação.

Um dos critérios, para realização da segunda avaliação, está ligado diretamente à primeira condição de finalização. Para realização da verificação da segunda condição de finalização, é necessário que:

- tenha acontecido, no mínimo, 25% do número máximo de iterações a serem realizadas;
- dez avaliações, da primeira condição de finalização, sejam realizadas sem serem atendidas.

Deve ser observado que ambas as condições justificam o encerramento do processo iterativo. Se a primeira for atingida, implica que as partículas atingiram um elevado grau de agregação, e que a taxa de melhoria da solução atingiu um limiar inferior admissível. Cumprindo a segunda, implica que, embora as partículas ainda não tenham atingido um elevado grau de agregação, uma melhoria significativa na melhor solução encontrada é improvável (INNOCENTE, 2006).

6 EXEMPLOS

6.1 Introdução

Com o intuito de comparar as estratégias de otimização locais e globais, especificamente através do uso das técnicas SAO e PSO foram analisados exemplos de funções analíticas e de problemas estruturais. Os resultados obtidos com as duas técnicas acima foram comparados através de códigos computacionais que fazem uso do SQP como otimizador.

6.2 Problemas considerando funções analíticas

Em cada problema foram analisadas as funções utilizando algoritmo SQP do MATLAB (função FMINCON) e a estratégia SAO, nesses dois casos utilizando valores iniciais indicados nas tabelas correspondentes de cada problema, e no caso do PSO, foram analisados sem uso de modelo substituto e utilizando a krigagem com 40, 60 e 80 pontos para criação do modelo substituto.

O número máximo de iterações permitidas para o SAO foi 100 e para o PSO 1000.

A tolerância admitida para a convergência do SAO foi de 10^{-5} .

Os parâmetros de vizinhança do PSO foram $c_1 = c_2 = 2$. A inércia foi iniciada com $w = 1.4$. A quantidade de partículas consideradas foi $np = 20$.

As funções que podem ter o número de variáveis de projeto aumentado foram consideradas de ordem dois.

6.2.1 Função Peaks

A Peaks (Figura 6.1) é uma função de duas variáveis, existente no MATLAB, obtida transladando e redimensionando distribuições Gaussianas.

Ela é definida por $(x_1, x_2) \in [-3, 3] \times [-3, 3]$ como segue:

$$f(x, y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10 \left(\frac{x}{5} - x^3 - y^5 \right) e^{-x^2-y^2} - \frac{1}{3} e^{-(x+1)^2-y^2} \quad (6.1)$$

O mínimo global é dado por: $x = [0.2283, -1.6255]$, $f(x) = -6.5511$, representado pela estrela vermelha da Figura 6.2.

Os resultados obtidos pelas diversas estratégias consideradas estão indicados na Tabela 6.1

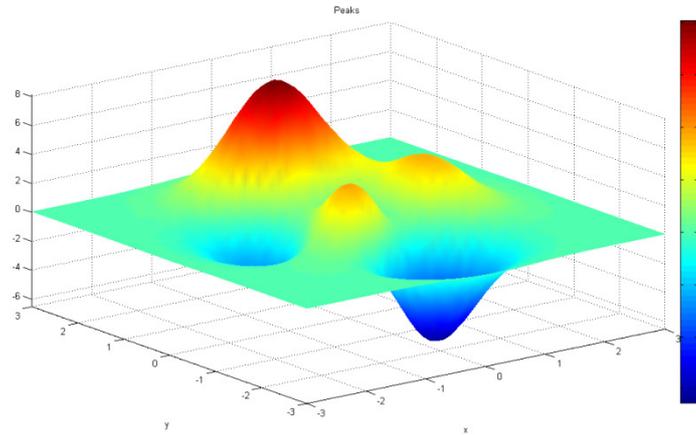


Figura 6.1 - Função Peaks

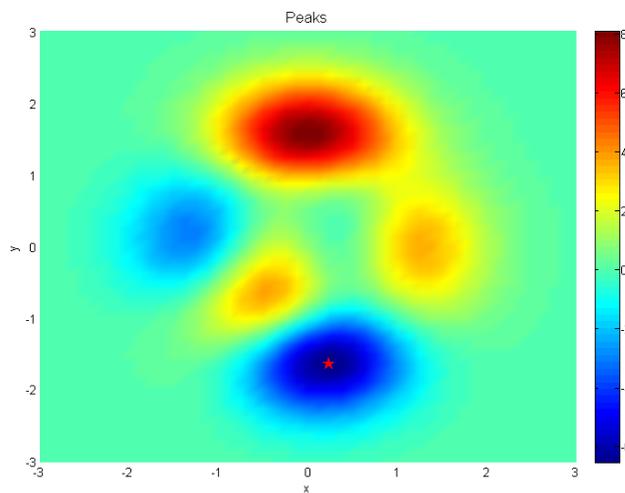


Figura 6.2 - Ponto ótimo da função Peaks

Tabela 6.1- Resultado da otimização da função Peaks

Otimizador	x_0	x_1^*	x_2^*	$f(x^*)$	Avaliação da função analítica
FMINCON	0; 1.5	0.2283	-1.6255	-6.5511	8
SAO	0; 1.5	0.2307	-1.6231	-6.5510	30
PSO		0.2283	-1.6255	-6.5511	600
PSO_kriging 40 pts		0.3277	-1.7229	-6.3014	40
PSO_kriging 60 pts		0.2312	-1.6797	-6.5089	60
PSO_kriging 80 pts		0.2323	-1.6848	-6.5005	80

Como pode ser observado, o SAO se aproxima bastante do ponto ótimo em poucas iterações, apenas cinco, totalizando trinta avaliações da função real. Já o PSO, com modelo substituto, apresenta resultados abaixo da solução analítica, mesmo tendo o número de avaliações da função real, dado pelo número de pontos

utilizados para construir o modelo, maior que o total de avaliações do SAO. Uma observação importante a fazer é que as estratégias locais dependem do ponto inicial, enquanto que a estratégia PSO, não.

6.2.2 Branin-Hoo

A função Branin-Hoo tem sido estudada intensivamente na literatura da otimização global.

Ela é definida por $(x_1, x_2) \in [-5, 10] \times [0, 15]$ como segue:

$$f(x, y) = \left(y - \frac{5.1}{4\pi^2} x^2 + \frac{5}{\pi} x - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x) + 10 \quad (6.2)$$

Destaca-se que a Branin-Hoo possui três mínimos $(-3.14, 12.27)$, $(3.14, 2.27)$ e $(9.42, 2.47)$ mostrados na Figura 6.4 e o valor do mínimo associado é de aproximadamente 0.3979, como mostrado na Tabela 6.2 cada execução do PSO fornece um dos três mínimos, assim foram feitas várias análises a fim de obter cada um desses mínimos. Com relação ao SAO, foram escolhidos diferentes pontos iniciais, representados pelo '+' vermelho, mostrado na Figura 6.4, para obter cada um dos três mínimos, indicados pela estrela vermelha da Figura 6.4.

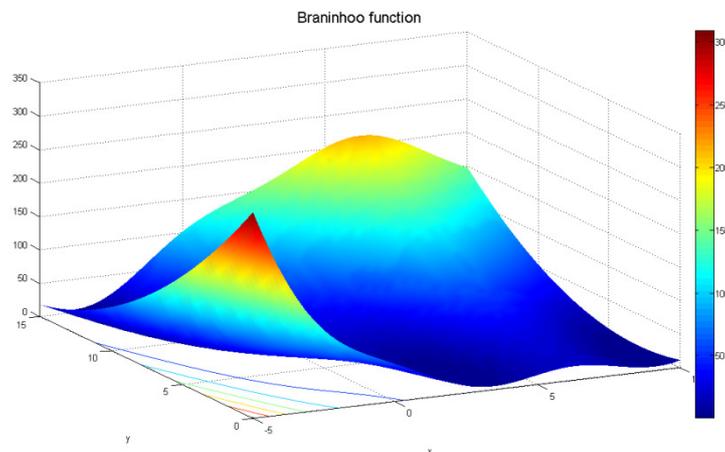


Figura 6.3 - Função Branin-Hoo

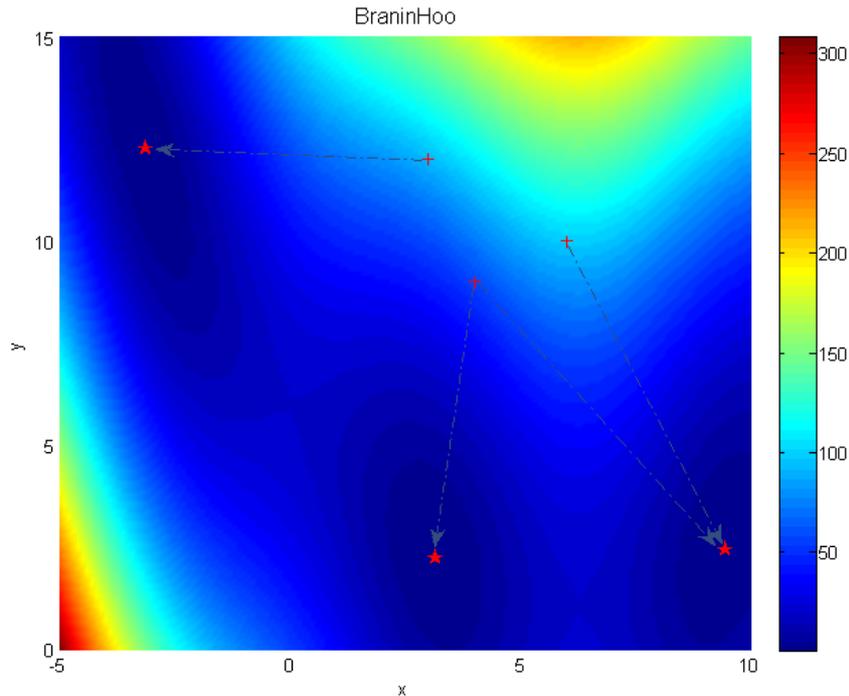


Figura 6.4 – Pontos ótimos da função Branin-Hoo

Tabela 6.2- Resultado da otimização da função Branin-Hoo

Otimizador	x_0	x_1^*	x_2^*	$f(x^*)$	Avaliação da função analítica
FMINCON	6; 10	9.4248	2.4751	0.3979	25
	4; 9	-3.1416	12.2750	0.3979	25
	3; 12	-3.1416	12.2752	0.3979	27
SAO	6; 10	9.4251	2.4769	0.3979	96
	4; 9	3.1422	2.2665	0.3979	72
	3; 12	-3.1407	12.2734	0.3979	121
PSO		9.4248	2.4750	0.3979	600
		3.1416	12.2750	0.3979	600
		-3.1416	2.2750	0.3979	600
PSO_kriging 40 pts		9.4599	2.4189	0.4112	40
		3.1436	2.2798	0.3979	40
		-3.1407	12.2655	0.3979	40
PSO_kriging 60 pts		9.4276	2.4873	0.3980	60
		3.1415	2.2749	0.3979	60
		-3.1416	12.2752	0.3979	60
PSO_kriging 80 pts		9.4219	2.4629	0.3980	80
		3.1416	2.2771	0.3979	80
		-3.1473	12.3013	0.3982	80

Nesta função, todas as técnicas utilizadas apresentaram bons resultados, com exceção apenas do PSO utilizando modelo substituto com quarenta pontos que em um dos mínimos, conseguiu apenas se aproximar da solução, que mostra a

influência da quantidade de pontos utilizados para a construção do modelo substituto.

6.2.3 Rastrigin

A função Rastrigin (Figura 6.5) é uma função não convexa usada largamente como um problema de teste de desempenho para algoritmos de otimização. É um exemplo típico de função não linear multimodal. Foi proposta inicialmente por Rastrigin como uma função bidimensional e foi generalizada por Mühlenbein. É um problema bastante difícil devido ao seu grande espaço de busca, está limitada em $x_i \in (-5.12, 5.12)$, e seu grande número de mínimos locais.

A função é definida por:

$$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \quad (6.3)$$

A solução ótima do problema (Figura 6.6) é o vetor $x = (0, \dots, 0)$ com $f(x) = 0$. Por ser um método de busca local, o SAO encontra apenas o mínimo local mais próximo ao ponto inicial. Esse tipo de função é adequado para algoritmos de busca global como o PSO, como mostrado na Tabela 6.3, o PSO sem uso de modelo substituto encontrou a solução ótima e quando utilizado o modelo substituto não foi muito eficaz, mas se aproximou do ótimo global. As Figuras 6.7, 6.8 e 6.9 mostram as superfícies de resposta resultantes do modelo aproximado gerado na estratégia PSO para 40, 60 e 80 pontos, respectivamente. A partir destas figuras se nota que quanto mais pontos utilizados para gerar o modelo substituto, mais a função aproximada se aproxima da função real. Essas figuras também mostram que o ponto ótimo da função aproximada avaliado na função real, representado por um quadrado magenta, está muito próximo do ponto ótimo procurado, representado por uma estrela vermelha.

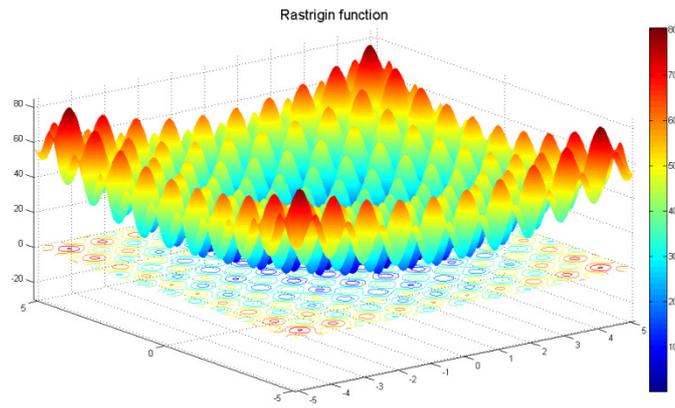


Figura 6.5 - Função Rastrigin

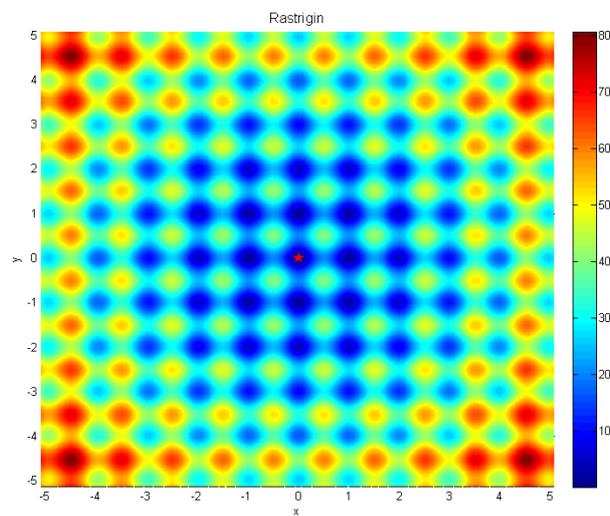


Figura 6.6 - Ponto ótimo da função Rastrigin

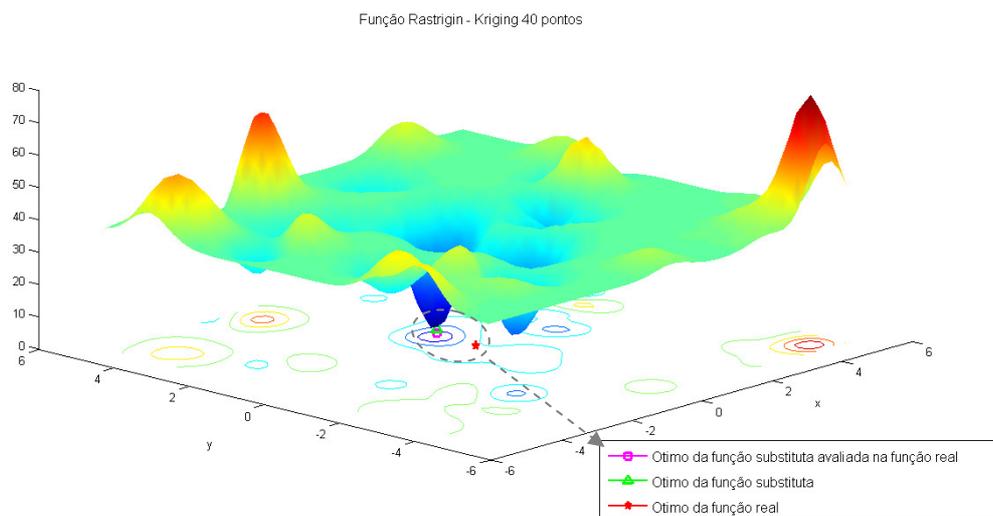


Figura 6.7 - Função Rastrigin aproximada com 40 pontos

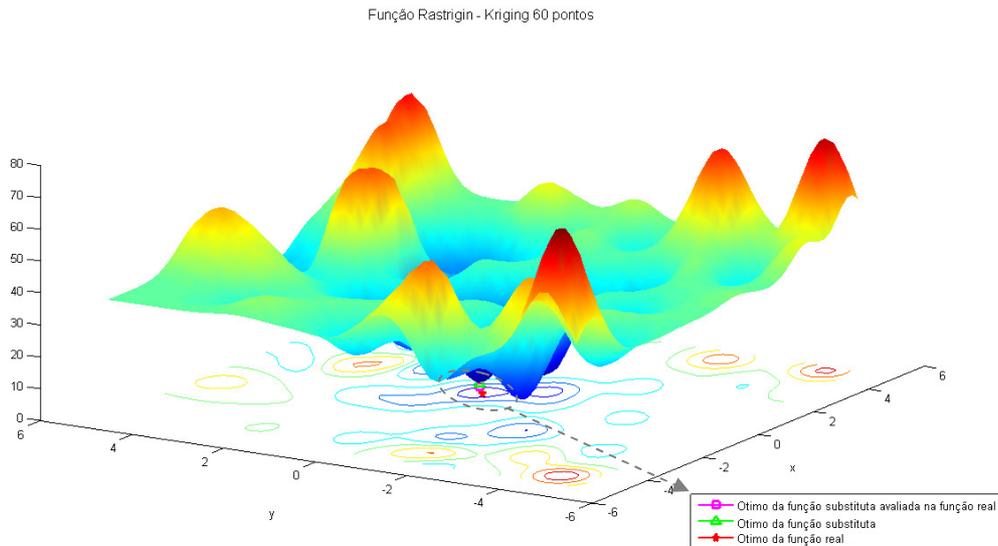


Figura 6.8 - Função Rastrigin aproximada com 60 pontos

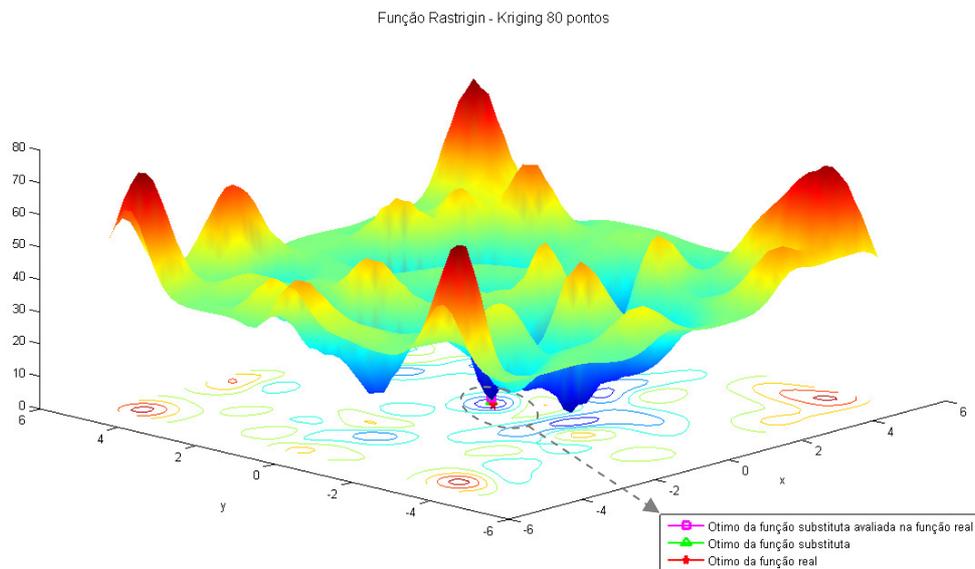


Figura 6.9 - Função Rastrigin aproximada com 80 pontos

Tabela 6.3- Resultado da otimização da função Rastrigin

Otimizador	x_0	x_1^*	x_2^*	$f(x^*)$	Avaliação da função analítica
FMINCON	-4; 3	0.59e-5	-0.445e-5	1.084e-8	7
SAO	-4; 3	3.9769	2.9838	24.8757	11
PSO		0	0	0	600
PSO_kriging 40 pts		-0.0291	1.0089	1.2015	40
PSO_kriging 60 pts		0.0437	0.0903	1.9512	60
PSO_kriging 80 pts		0.0479	0.0831	1.7917	80

Neste trabalho também foi considerada a influência da quantidade de variáveis da função Rastrigin no PSO. Os resultados encontram-se na Tabela 6.4. Verifica-se que o PSO também foi eficaz na solução deste problema considerando muitas variáveis.

Tabela 6.4- Resultado da otimização da função Rastrigin

n	$f(x^*)$
5	0
10	0
15	0
20	0

6.2.4 Ackley

A função Ackley tem como característica possuir muitos ótimos locais. Foi originalmente definida para duas dimensões (Figura 6.10), mas o problema foi generalizado para n dimensões.

A função, descrita pela equação (6.4), está limitada em $x_i \in (-32.768, 32.768)$.

$$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e \quad (6.4)$$

A solução ótima do problema bidimensional é o vetor $x = (0, 0)$ com $f(x) = 0$ (Figura 6.11).

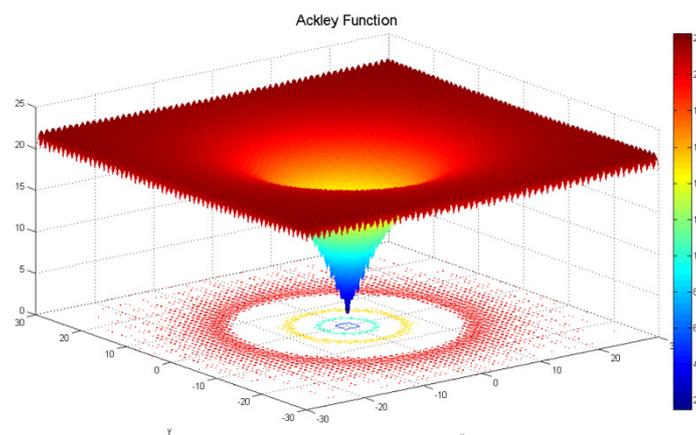


Figura 6.10 - Função Ackley

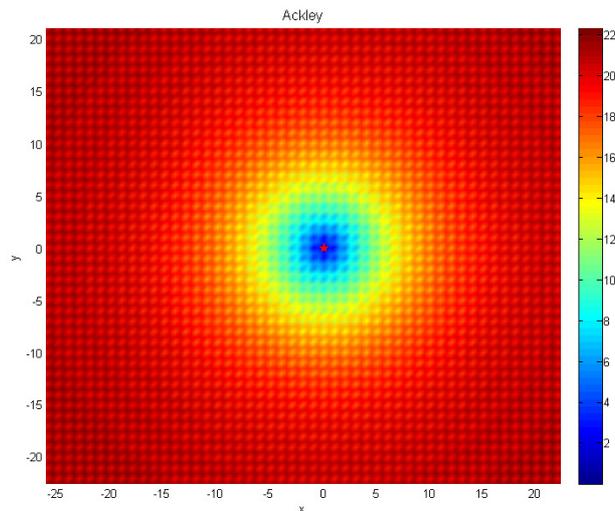


Figura 6.11 - Ponto ótimo da função Ackley

Os resultados obtidos pelas estratégias utilizadas estão dispostos na Tabela 6.5. Embora a função Ackley seja multimodal verifica-se que a estratégia SAO encontrou a região do ponto ótimo, isso é devido ao modelo substituto que suavizou os ruídos desta função. No caso da estratégia PSO nota-se que quando se utiliza o modelo substituto, a eficácia da estratégia diminui, também devido à suavização dos ruídos que pode esconder o ótimo global da função. As Figuras 6.12, 6.13 e 6.14 mostram a função Ackley livre de ruídos numéricos, utilizando modelo substituto com 40, 60 e 80 pontos. A diferença da resposta da função aproximada para a real é tão pequena comparado com a diferença de eficiência, que o uso do modelo substituto é justificado.

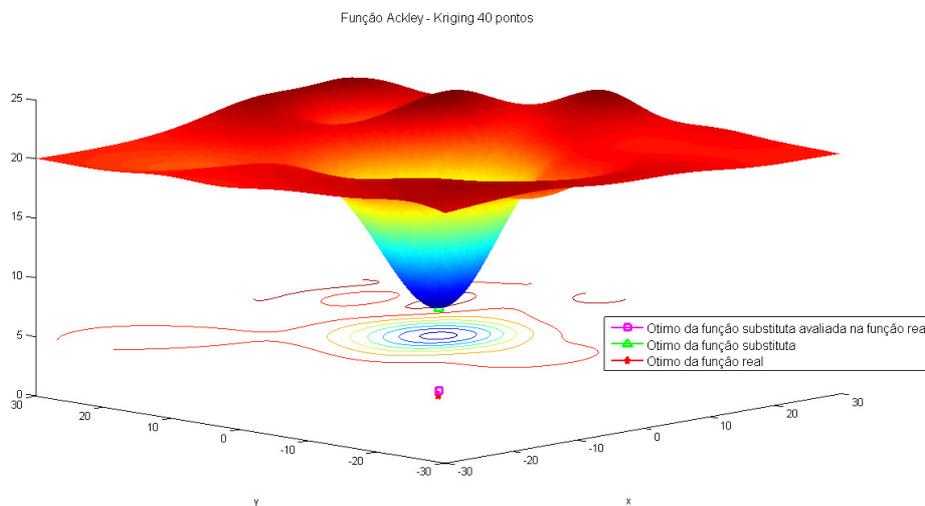


Figura 6.12 - Função Ackley aproximada com 40 pontos

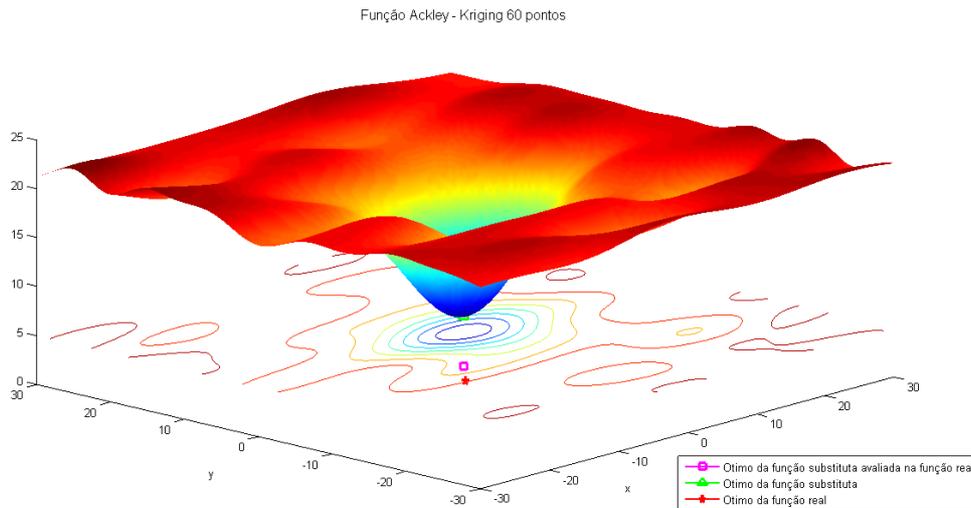


Figura 6.13 - Função Ackley aproximada com 60 pontos

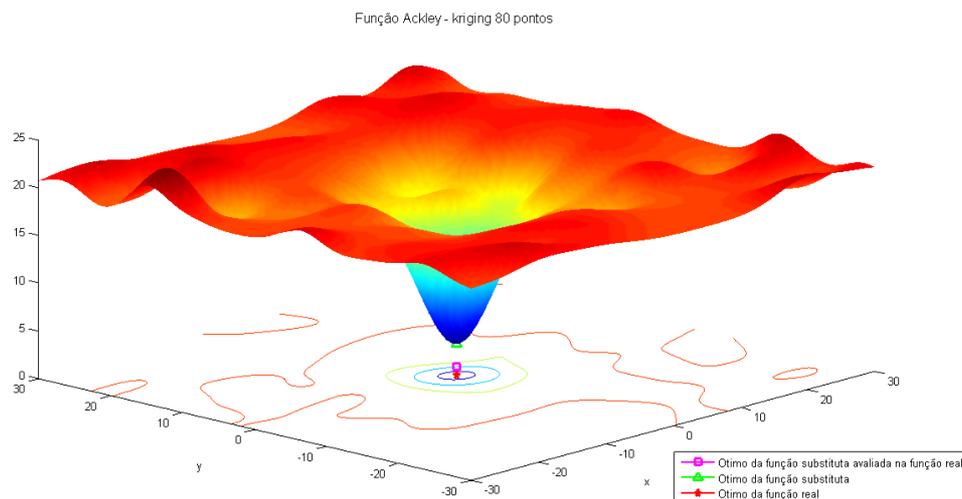


Figura 6.14 - Função Ackley aproximada com 80 pontos

Tabela 6.5- Resultado da otimização da função Ackley

Otimizador	x_0	x_1^*	x_2^*	$f(x^*)$	Número de avaliação de função
FMINCON	2.5; 2.5	0	0	0	69
SAO	2.5; 2.5	-0.0075	-0.0047	0.0270	96
PSO		0	0	0	600
PSO_kriging 40 pts		0.0769	0	0.3676	40
PSO_kriging 60 pts		-0.2111	0.0556	1.5207	60
PSO_kriging 80 pts		-0.1037	-0.1037	0.9137	80

Foi avaliada a influência da quantidade de variáveis da função Ackley na convergência da otimização via o PSO e o SAO. Os resultados encontram-se na

Tabela 6.6. Mais uma vez, foi demonstrado que o PSO conseguiu encontrar a solução ótima de um problema com muitas variáveis. Como explicado acima, devido à suavização dos ruídos da função em questão, a estratégia SAO foi menos eficaz, mas conseguiu encontrar a região do ótimo global para até dez variáveis.

Tabela 6.6- Resultado da otimização da função Ackley

n	Otimizador	$f(x^*)$
5	SAO	0.0061
	PSO	8.88e-016
10	SAO	0.0159
	PSO	8.88e-016
15	SAO	0.9583
	PSO	8.88e-016
20	SAO	9.0027
	PSO	8.88e-016

6.2.5 Rosenbrock

A função Rosenbrock (Figura 6.15) também conhecida como função banana, é uma função não convexa originalmente de duas variáveis, que pode ser formulada para n-variáveis, utilizada como teste para problemas de otimização. Foi introduzida por Rosenbrock em 1960.

A função apresenta um mínimo global no interior de um longo vale estreito da forma parabólica.

A função é definida por:

$$f(x) = \sum_{i=1}^{n-1} \left[(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2 \right] \quad (6.5)$$

A função está limitada em $-2 \leq x_i \leq 2$. O mínimo global é obtido no ponto $x = (1, \dots, 1)$ no qual a função vale zero.

Encontrar o vale é trivial, no entanto, convergir para o mínimo global é difícil. Utilizando o PSO, o mínimo global foi encontrado em todos os testes. Já utilizando o SAO apenas o vale foi encontrado, mas devido à suavidade da função nesse vale, a função não avançou para o ótimo. A Figura 6.16 mostra o ótimo global, estrela vermelha, e alguns resultados do SAO, onde os 'x' em amarelo são soluções encontradas a partir de uma região de confiança inicial de 0.15 e os '+' verdes a

partir de uma região de confiança inicial de 0.20, as melhores soluções em cada um desses casos são mostradas na Tabela 6.7.

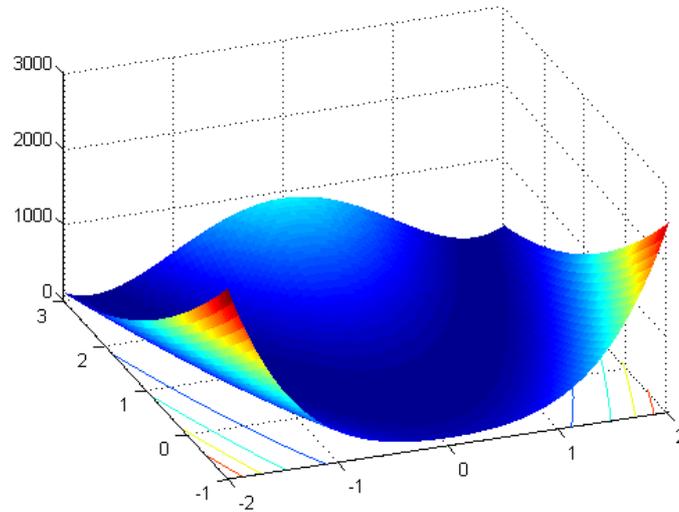


Figura 6.15 - Função Rosenbrock

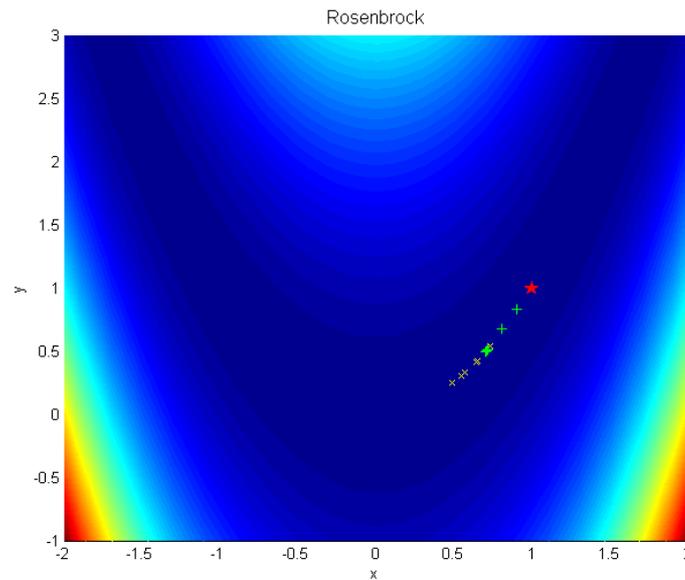


Figura 6.16 - Ponto ótimo da função Rosenbrock

Tabela 6.7- Resultado da otimização da função Rosenbrock

Otimizador	x_0	x_1^*	x_2^*	$f(x^*)$	Número de avaliação de função
FMINCON	0; -1	1	1	0	116
SAO_RC15	0; -1	0.7339	0.5395	0.0709	110
SAO_RC20	0; -1	0.9041	0.8296	0.0242	73
PSO		1	1	0	600
PSO_kriging 40 pts		1.0016	1.0034	6.46e-6	40
PSO_kriging 60 pts		0.9994	0.9986	4.37e-6	60
PSO_kriging 80 pts		1.0012	1.0025	2.41e-6	80

Verifica-se, neste caso, que a quantidade de pontos utilizados para o modelo substituto do PSO não influenciou o resultado, com apenas quarenta pontos foi possível encontrar o ótimo global.

Assim como nos problemas de n -variáveis anteriores, foi verificada a influência da quantidade de variáveis no PSO. Os resultados encontram-se na Tabela 6.8, que mostra que o resultado foi encontrado para apenas cinco variáveis.

Tabela 6.8- Resultado da otimização da função Rosenbrock

n	$f(\mathbf{x}^*)$
5	7.81997e-007
10	8.85960
15	12.46416
20	18.86210

6.2.6 Textbook

O exemplo em análise é um problema de otimização sujeito a restrições não lineares. O problema é formulado como:

$$\begin{aligned}
 \text{Minimize} \quad & f = (x_1 - 1)^4 + (x_2 - 1)^4 \\
 \text{Sujeito à} \quad & g_1 = x_1^2 - \frac{x_2}{2} \leq 0 \\
 & g_2 = x_2^2 - \frac{x_1}{2} \leq 0 \\
 & 0.5 \leq x_1 \leq 5.8 \\
 & -2.9 \leq x_2 \leq 2.9
 \end{aligned} \tag{6.6}$$

Contornos desse problema estão ilustrados na Figura 6.17.

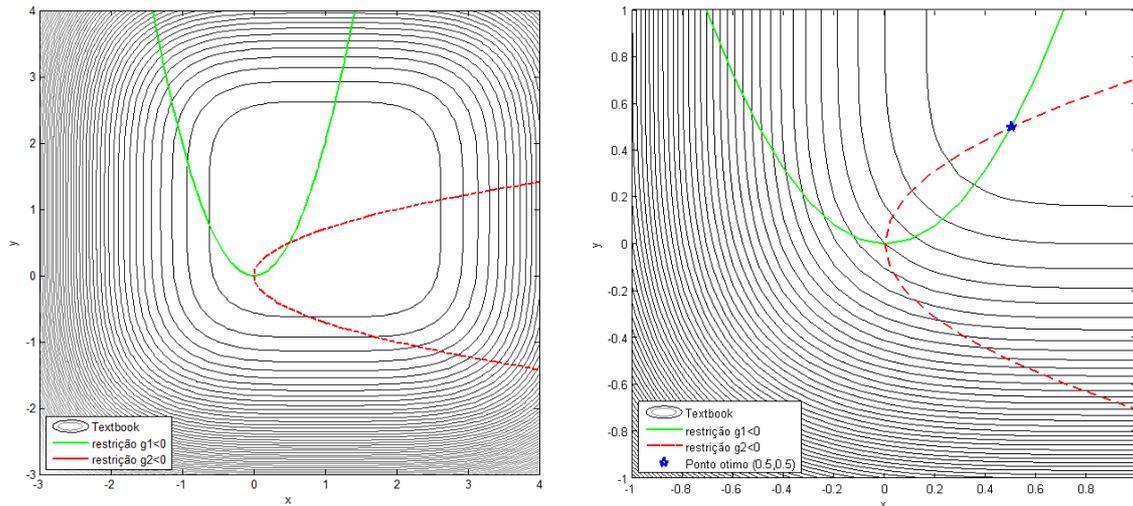


Figura 6.17 - Contornos do problema textbook. A região viável está na intersecção das duas restrições g_1 (sólido) e g_2 (pontilhado)

Para o problema textbook, o mínimo sem restrição ocorre em $(x_1, x_2) = (1, 1)$. Entretanto, a inclusão das restrições move o mínimo para $(x_1, x_2) = (0.5, 0.5)$.

Como pode ser visto na Tabela 6.9, todas as técnicas encontraram o ótimo global sem grandes dificuldades. Vale observar que a estratégia SAO foi bastante eficiente na solução deste problema, encontrando a solução com menos avaliações da função real que o algoritmo SQP (representado pelo FMINCON).

Tabela 6.9- Resultado da otimização da função Textbook

Otimizador	x_0	x_1^*	x_2^*	$f(x^*)$	Número de avaliações da função
FMINCON	0; -1	0.5	0.5	0.125	18
SAO	0; -1	0.5	0.5	0.125	11
PSO		0.5	0.5	0.125	600
PSO_kriging 40 pts		0.5	0.5	0.125	40
PSO_kriging 60 pts		0.5	0.5	0.125	60
PSO_kriging 80 pts		0.5	0.5	0.125	80

6.3 Problemas de Otimização Estrutural

Foram estudados problemas práticos de engenharia estrutural envolvendo treliças bidimensionais e um pórtico plano.

6.3.1 Treliças

Os modelos de treliças bidimensionais aqui considerados são exemplos clássicos da literatura (KIRSCH, 1981). As treliças são estruturas de análise simples, no entanto o seu estudo é importante tanto para se avaliar as estratégias utilizadas quanto para verificar as implementações conduzidas. Os exemplos investigados tem as seguintes denominações: a) treliça com 10 barras e b) treliça com 64 barras. Na Figura 6.18 são fornecidas as informações com relação às dimensões geométricas, ao carregamento estático externo e às condições de contorno.

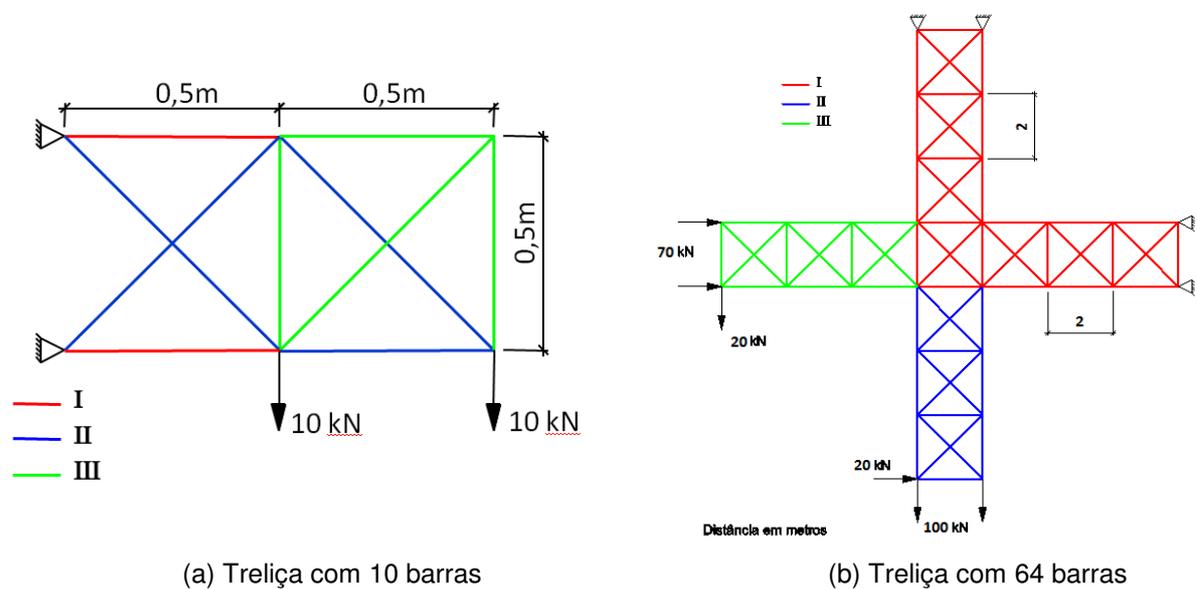


Figura 6.18 - Exemplos clássicos de treliças bidimensionais.

Percebe-se que três cores diferentes (vermelho, azul e verde) são usadas na representação gráfica dos casos de treliças. Tais cores definem três regiões (I, II e III) que representam o agrupamento das áreas das barras em três distintas variáveis de projeto. As variáveis estão limitadas entre $0.1\text{cm}^2 \leq x_i \leq 10\text{cm}^2$.

Esta estratégia é bastante utilizada no processo de otimização, pois reduz o número de variáveis de projeto tornando-o mais viável em termos computacionais. Por outro lado damos também um caráter mais prático aos resultados obtidos, uma vez que projetar e/ou montar uma estrutura contendo apenas três possíveis áreas é mais simples do que montar uma estrutura com, por exemplo, 64 possíveis áreas.

A função objetivo é o volume da treliça, calculado como pela expressão da equação (6.7) e as restrições são limitações nas tensões das barras da treliça, dadas pela equação (6.8).

$$vol = \sum_{i=1}^{n_{elem}} A_i l_i \quad (6.7)$$

$$\sigma^e = \frac{N^e}{A^e} \quad (6.8)$$

onde l é o comprimento de cada barra A é sua seção transversal, N é o esforço normal do elemento e e n_{elem} é o número de elementos que compõem a treliça.

Os esforços em cada elemento são obtidos com o pós-processamento dos resultados obtidos na análise realizada através do Método dos Elementos Finitos, considerando a equação (6.9)

$$f^e = \mathbf{K}^e \times \mathbf{u}^e \quad (6.9)$$

onde \mathbf{K}^e é a matriz de rigidez do elemento, e f^e e \mathbf{u}^e são, respectivamente, o vetor de esforços e deslocamentos nodais do elemento e .

O módulo de elasticidade do material é igual a 210 GPa. O valor das variáveis de projeto para a configuração inicial é adotada igual a 5 cm².

6.3.1.1 Exemplo a) Treliça com 10 barras

Nesse problema o valor máximo permitido para tensão em qualquer barra é $\sigma = 25 \text{ MPa}$ (tanto para tração quanto para compressão).

Na otimização com o PSO foram considerados os seguintes parâmetros:

- número máximo de iterações permitidas de 3000;
- parâmetros de vizinhança $c_1 = c_2 = 2$;
- inércia $w = 1.4$;
- quantidade de partículas $np = 20$.

Os resultados da otimização utilizando o SAO, o PSO e o SQP, calculado com a função FMINCON do MATLAB, são apresentados na Tabela 6.10.

Tabela 6.10- Resultado da otimização da treliça de 10 barras

Otimizador	$x_0 (cm^2)$	$x_1^* (cm^2)$	$x_2^* (cm^2)$	$x_3^* (cm^2)$	$f(x^*) (cm^3)$	Número de avaliações da função real
FMINCON	5; 5; 5	8.0476	5.7241	0.1	2327.3034	9
SAO	5; 5; 5	8.0476	5.7241	0.1	2327.3034	49
PSO		8.0476	5.7241	0.1	2327.3034	500

Os resultados mostram que os métodos utilizados foram eficazes na solução desse problema estrutural, embora o custo computacional do PSO tenha sido mais elevado, comparando-se aos casos de funções analíticas.

É interessante notar que o valor das variáveis x_1 e x_2 no ótimo aumenta em relação ao ponto inicial, isso se deve à restrição imposta que inicialmente não é atendida.

6.3.1.2 Exemplo a) Treliça com 64 barras

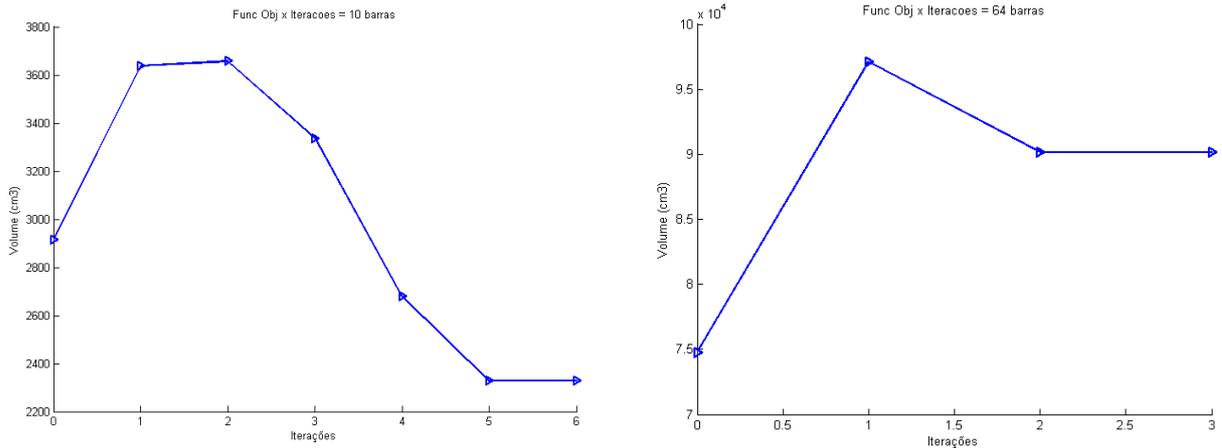
Como o carregamento atuante nesse problema é até dez vezes maior que no problema anterior, o valor máximo permitido para tensão em qualquer barra é adotado $\sigma = 250 MPa$.

Tabela 6.11- Resultado da otimização da treliça de 64 barras

Otimizador	$x_0 (cm^2)$	$x_1^* (cm^2)$	$x_2^* (cm^2)$	$x_3^* (cm^2)$	$f(x^*) (cm^3)$	Número de avaliações da função
FMINCON	5; 5; 5	6.7061	5.7524	4.7642	90156.3610	5
SAO	5; 5; 5	6.7061	5.7524	4.7642	90156.3610	28
PSO		6.7061	5.7524	4.7642	90156.3610	500

Também neste problema as estratégias SAO e PSO tiveram sua eficácia comprovada.

A Figura 6.19 mostra o histórico das iterações da estratégia SAO para as duas treliças aqui consideradas. Percebe-se que a função objetivo inicialmente cresce para então decrescer, isso se deve à restrição imposta que não é atendida no ponto inicial.



(a) Treliza com 10 barras

(b) Treliza com 64 barras

Figura 6.19 – Histórico de iterações da estratégia SAO

6.3.2 Pórtico

Nesta seção é apresentado o pórtico de madeira da Figura 6.20. O objetivo deste problema é encontrar as dimensões dos pilares e da viga que minimizem o volume da estrutura, levando em consideração que não se deve ultrapassar a tensão admissível (tração e/ou compressão) nas seções críticas do pórtico no topo do pilar, na extremidade da viga e no centro do vão.

O pórtico bidimensional aqui considerado é construído a partir de dois pilares conectados rigidamente a uma viga. A viga tem comprimento L e os pilares αL , a seção transversal tanto da viga quanto do pilar é quadrada, onde a largura e altura são iguais a x_1 para os pilares e x_2 para a viga. As bases dos pilares são engastadas e uma carga distribuída q é aplicada no vão da viga.

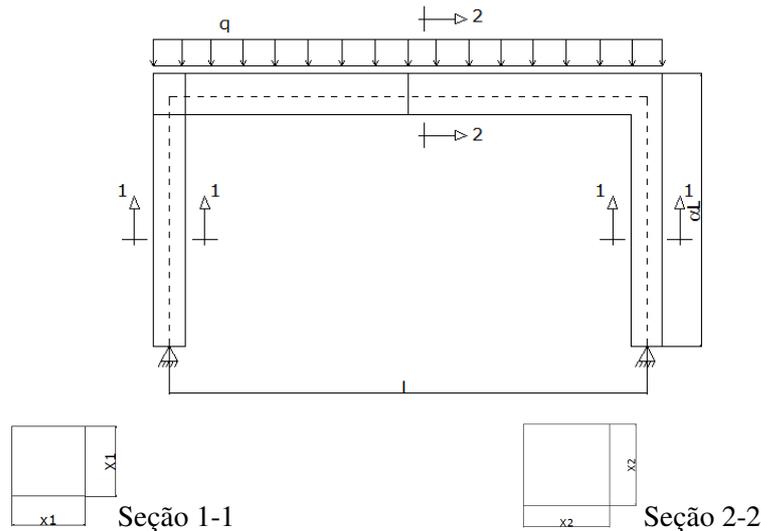


Figura 6.20 - Pórtico plano de madeira

A formulação do problema de otimização é definida segundo a equação (6.10), onde a função objetivo $V(x)$ é o volume da estrutura, as restrições g_1, g_2 e g_3 são, respectivamente, as tensões no topo dos pilares, nos extremos da viga e no centro do vão e as variáveis de projeto x_1 e x_2 são, respectivamente, as dimensões dos pilares e da viga.

$$\begin{aligned}
 &\text{Minimize} && V(x) = 2\alpha l x_1^2 + l x_2^2 \\
 &\text{Sujeito à:} && \\
 &\quad \langle \text{Topo do pilar} \rangle && g_1(x) = \sigma_1(x) - \sigma_{adm}; \quad g_1 \leq 0 \\
 &\quad \langle \text{Extremidade da} && g_2(x) = \sigma_2(x) - \sigma_{adm}; \quad g_2 \leq 0 \\
 &\quad \quad \text{viga} \rangle && \\
 &\quad \langle \text{Centro do vão} \rangle && g_3(x) = \sigma_3(x) - \sigma_{adm}; \quad g_3 \leq 0 \\
 &\text{Onde} && \\
 &&& \sigma_1(x) = \frac{ql}{2x_1^2} + \frac{3ql^2 x_1}{6x_1^4 + 4\alpha x_2^4} && (6.10) \\
 &&& \sigma_2(x) = \frac{x_1^4 (x_2 + 6\alpha l)}{x_2^3 (6x_1^4 + 4\alpha x_2^4)} \frac{ql}{2\alpha} \\
 &&& \sigma_3(x) = \frac{x_1^4 x_2 + \alpha l (3x_1^4 + 6\alpha x_2^4)}{x_2^3 (6x_1^4 + 4\alpha x_2^4)} \frac{ql}{2\alpha}
 \end{aligned}$$

A Figura 6.21 mostra o espaço de projeto com duas curvas de nível da função objetivo e as três restrições. A constante α foi adotada igual a um, o comprimento da viga $l = 5.5m$, a carga distribuída $q = 15 \text{ kgf/cm}$ e a tensão admissível $\sigma_{adm} = 130 \text{ kgf/cm}^2$. Os limites inferior e superior foram considerados iguais a 5 cm e 35 cm, respectivamente.

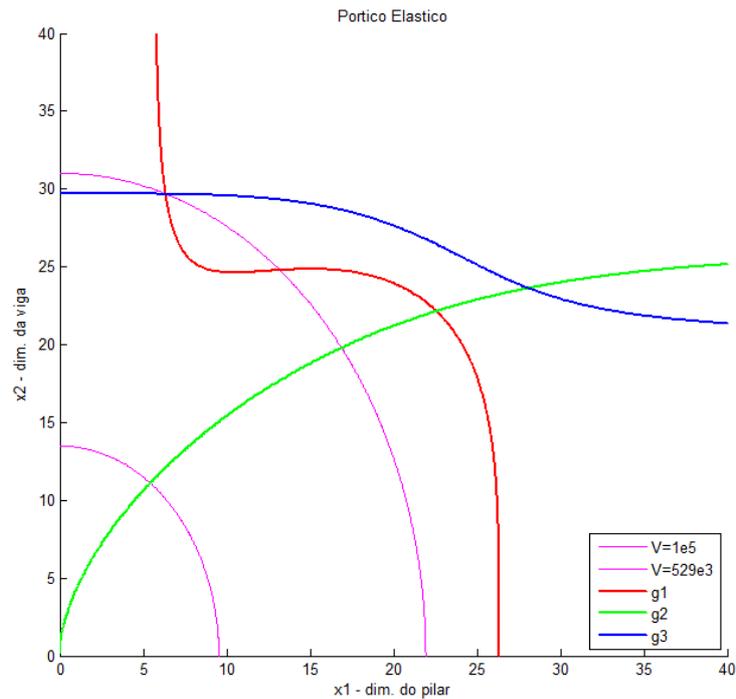


Figura 6.21 - Função objetivo e restrições do pórtico de madeira

A Tabela 6.12 apresenta os resultados obtidos pelo SQP (FMINCON), SAO e pelo PSO. Os valores obtidos do volume da estrutura são praticamente iguais, assim sendo, as técnicas que utilizam modelo substituto são mais adequadas, por terem sido mais eficientes.

Tabela 6.12- Resultado da otimização do pórtico plano

Otimizador	x_0	x_1^*	x_2^*	$f(x^*)$	Número de avaliações da função
FMINCON	27; 11	6.3533	29.6718	0.52863	31
SAO	27; 11	6.3533	29.6718	0.52863	28
PSO		6.3545	29.6628	0.52849	600
PSO_kriging 40 pts		6.3545	29.6629	0.52850	40
PSO_kriging 60 pts		6.3545	29.6628	0.52849	60
PSO_kriging 80 pts		6.3545	29.6628	0.52849	80

7 CONCLUSÕES

Neste trabalho, foram analisados vários exemplos e duas diferentes estratégias de otimização e com os resultados obtidos pode-se chegar às seguintes conclusões a respeito da utilização de técnicas de otimização:

- As estratégias de otimização podem auxiliar os profissionais da engenharia a escolher uma melhor configuração de problemas práticos, como as treliças e o pórtico aqui estudados, a fim de obter um projeto viável em termos das restrições impostas, e econômico, que para os casos estudados é a economia do material utilizado.
- Ambos os métodos foram eficazes na solução dos problemas estruturais.
- Nos problemas analíticos, os métodos se comportaram de acordo com suas limitações. O SAO encontrou dificuldades em funções com elevado número de ótimos locais, onde a estratégia pode se perder e em funções muito suaves, onde o gradiente calculado se aproxima de zero.
- O PSO tem a inconveniência de precisar de um elevado número de avaliação de funções, mas o uso de modelos substitutos melhorou sua eficiência, que embora continue avaliando muitas funções, estas são funções aproximadas, que em problemas complexos diminui bastante o custo computacional do método. Além disto, é o procedimento adequado quando a função objetivo for multimodal.

Para trabalhos futuros é sugerida a combinação destes métodos, ou seja, iniciar o problema com o PSO para encontrar uma provável região de ótimo global e a partir de um ponto desta região iniciar o SAO.

8 BIBLIOGRAFIA

ADAMS, B. M.; BOHNHOFF, W. J.; EDDY, J. P.; ELDRED M. S.; GAY, D. M.; HOUGH, P. D.; SWILER, L. P.; HART, W. E.; WATSON, J.P.; GRIFFIN, J. D.; KOLDA, T. G.; WILLIAMS P. J.; MARTINEZ, M. L. DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.0 User's Manual, Sandia National Laboratories, USA, 2009.

AFONSO, S. M. B. Shape Optimization of Shells Under Static and Free Vibration Conditions, PhD thesis, Department of Civil Engineering, University of Wales, Swansea, 1995.

AFONSO, S. M. B. e HOROWITZ, B. (1998), Utilização do MATLAB no Desenvolvimento de Ferramentas Educacionais no Ensino da Otimização Estrutural, in: V Congresso de Engenharia Mecânica do Norte-Nordeste. Fortaleza: UFC. – vol. II, pp. 243-430.

AFONSO, S.M.B.; HOROWITZ, B.; WILLMERSDORF, R.B. “Comparative Study of Surrogate Models For Engineering Problems”, in 7th. ASMO_UK/ISSMO International Conference on Engineering Design Optimization, Bath-UK, 7-8, July, 2008.

ALBUQUERQUE, T. M. M. Análise e otimização de problemas térmicos e estruturais bidimensionais através do método da base reduzida. Dissertação de Mestrado, Recife , 2005.

ATREK, E.; GALLAGHER, R. H.; RAGSDALL, K. M.; ZIENKIEWICZ, O. C. *New Directions in Optimum Structural Design*, John Wiley e Sons, Bath, Avon, 1984.

BEALE, E. M. L. Introduction to optimization, John Wiley e Sons, London, Great Britain, 1988.

BEHLING, R. Minimização de Quadráticas Convexas em Caixas sobre Variedades Afins, um sub-problema de PQS, Dissertação de Mestrado, Florianópolis, 2006.

CASTRO, R. E. Otimização de Estruturas com Multiobjetivos via Algoritmo Genético, Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, 2001.

COLLETTE, Y.; SIARRY, P. ,Multiobjective Optimization – Principles and Case Studies. 1st edition, 2003.

COOK, R. D. Concepts and Applications of Finite Elements Analysis, 2nd edition, John Wiley & Sons, Singapore, 1981.

FORRESTER, A.; SOBESTER, A.; KEANE, A. "Engineering Design Via Surrogate Modelling: A Practical Guide", Wiley, 2008.

GELLAT Jr., C. D.; KIRPATRICK, S.; VECCHI, M. P. Optimization by Simulated Annealing, Science 220, p. 671-680, 1983.

GIUNTA, A. A.; WATSON, L.T. A Comparison of Approximation Modeling Techniques: Polynomial versus Interpolating Models, AIAA-98-4758, 1998

GOLDBERG, D. E. Genetic Algorithms In Search, Optim., And Machine Learning, 1989.

HAFTKA, R. T.; GURDAL, Z. Elements of Structural Optimization, 3th Revision, Kluwer Academic Publishers, Dodrecht, Netherlands, 1993.

INNOCENTE, M. S. (2006), Population-Based Methods: Particle Swarm Optimization – *Development of a General-Purpose Optimizer and Applications*, master's thesis, Universitat Politècnica de Catalunya, Barcelona.

KARTAM, N.; FLOOD, I.; GARRETT, J. H.; AGRAWAL, G. Artificial Neural Networks for Civil Engineers: Fundamentals and Applications, ASCE Publications, 1997

KEANE, A. J.; NAIR, P. B. Computational Approaches for Aerospace Design: The pursuit of Excellence, Willey, 2005.

KIRSCH, U. "Optimum Structural Design - Concepts, Methods and Applications", McGraw-Hill, New-York, 1981

KIRSCH, U. Structural Optimization: Fundamentals and Applications, Springer-Verlag, Berlin, 1993.

MATLAB, The MathWorks, Inc., Version 7.10.0.499 (R2010a), Copyright 1984-2010.

MICHALEWICZ, Z.; DASGUPTA, D., editors, Evolutionary Algorithms In Engineering Applications, Springer Verlag, 1997

NAIDU, S. L. S. Neural Network Surrogate Model For Multidisciplinary Design Optimization, M. Tech. Dissertation, Indian Institute of Technology, 2004.

NOCEDAL, J.; WRIGHT, S. J. Numerical Optimization, Springer-Verlag, Rensselaer, NY, 2000.

OLIVEIRA, L. C. de. Otimização Utilizando o Algoritmo Evolucionário do Enxame de Partículas. Dissertação de mestrado (Engenharia Civil) - Universidade Federal de Pernambuco, 2008.

PARSOPOULOS, K. E.; VRAHATIS, M. N. Particles Swarm Optimization Method in Multiobjective Problems, ACM 1-58113-445, p.603-607, 2002.

POMEROY, P. (2003), *An Introduction to Particle Swarm Optimization*, in: <http://www.adaptativeview.com/articles/ipsoprnt.html> (Dezembro, 2005).

POWELL, M. J. D. Algorithms for Nonlinear Constraints that Use Lagrangian Functions, Math. Prog., 14, p. 224-248, 1978.

SHI, Y. H. e EBERHART, R. C., *A Modified Particle Swarm Optimizer*, Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998.

SHI, Y. H. e EBERHART, R. C., *Parameter Selection in Particle Swarm Optimization*, Evolutionary Programming VII, Lecture Notes in Computer Science, pp. 591-600, 1998.

SILVA, E. C. N. Otimização Aplicada ao Projeto de Sistemas Mecânicos, Departamento de Engenharia Mecatrônica e Sistemas Mecânicos, Escola Politécnica da USP, Apostila, 2010.

SILVA, M. F. da, Estratégias de Aproximação para a Otimização Estrutural. Dissertação – Departamento de Engenharia Civil, Universidade Federal de Pernambuco, Recife-PE, 2009.

SIMPSON, T.W.; MAUERY, T.M.; KORTE, J.K.; MISTREE, F. Kriging Models for Global Approximations in simulation-Based Multidisciplinary Design Optimization, AIAA Journal, 39(12):2233-2241,2001

VANDERPLAATS, G. N. *Numerical Optimization Techniques for Engineering Design: with Applications*, McGraw-Hill, New York, 1984.

VENTER, G. and SOBIESZCZANSKI-SOBIESKI, J., *Particle Swarm Optimization*, in: 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 22-25, Denver, Colorado, 2002.