

Universidade de Pernambuco (UPE)
Escola Politécnica de Pernambuco (POLI)
Programa de Pós-graduação em Engenharia da Computação (PPGEC)
Programa de Pós-graduação em Engenharia de Sistemas (PPGES)

Relatório da Prática de Inteligência de Enxames
ABC vs. PSO vs. FSS

Aluno: Carlos Henrique Maciel Sobral Timoteo

Professor: Dr. Carmelo Bastos Filho

27 de Junho de 2013

Lista de Figuras

2.1	Função Rastrigin para $n = 2$	5
4.1	Boxplot ABCxPSOxFSS	8

Sumário

1	Introdução	3
1.1	Algoritmo ABC	3
1.2	Algoritmo PSO	3
1.3	Algoritmo FSS	4
1.4	Objetivo da medição	4
2	Planejamento	5
2.1	Função para otimização	5
2.1.1	Rastrigin	5
2.2	Tabela de Experimentos	5
3	Operação	7
4	Resultados	8
4.1	Resultados obtidos	8
4.2	Comparação das Execuções	9
5	Conclusão	10

Capítulo 1

Introdução

A motivação deste estudo é comparar três técnicas de otimização e aprender como desenvolver um algoritmo de otimização baseado em cardume de peixes, mais especificamente o *FSS* (do inglês *Fish School Search*, Busca de Cardume de Peixes) e um algoritmo de otimização baseado em enxame de abelhas, mais especificamente o *ABC* (do inglês *Artificial Bee Colony*, Colônia Artificial de Abelhas), adquirindo a sensibilidade quanto aos parâmetros das técnicas e desempenho de cada. O objeto de estudo deste trabalho são os algoritmos clássicos do ABC, PSO e FSS.

Um experimento será montado e, se necessário, um estudo estatístico será efetuado para melhorar a compreensão do desempenho de cada algoritmo.

1.1 Algoritmo ABC

O ABC é um dos algoritmos mais recentemente definidos por Dervis Karaboga em 2005[1], motivado pelo comportamento inteligente de abelhas. É tão simples como Particle Swarm Optimization (PSO) e Evolução Diferencial (DE), e utiliza apenas os parâmetros de controle comuns, tais como o tamanho da colônia e número do ciclo máximo. O ABC como uma ferramenta de otimização, fornece um procedimento de pesquisa de base populacional, em que os indivíduos chamados alimentos posições são modificados pelas abelhas artificiais com o tempo e objetivo da abelha é descobrir os locais de fontes de alimentos com quantidade elevada de néctar e, finalmente, o que tem o maior néctar.

1.2 Algoritmo PSO

O PSO é um algoritmo inspirado no voo de aves em busca de alimentos. Foi idealizado por Kennedy e Eberhart em 1995 [2]. Cada partícula é uma solução candidata para um problema e, por meio de troca de informações, o conjunto de partículas (o enxame) realiza, iterativamente, uma busca para encontrar uma "boa" solução para o problema. A solução é representada pela posição da partícula no espaço de busca. Não é possível afirmar que a solução final é a melhor pois seria necessário uma exploração total do espaço de busca e um estudo de convergência para o algoritmo implementado.

Como há movimentação de partículas pelo espaço de busca, é clara a presença de uma velocidade e de uma posição. O algoritmo original do PSO apresenta duas equações que definem a atualização da velocidade e da posição iterativamente. Estas equações são, respectivamente

$$\vec{v}[t+1] = \omega \vec{v}[t] + C_1 r_1 (\vec{p}_{Best} - \vec{x}[t]) + C_2 r_2 (\vec{g}_{Best} - \vec{x}[t]) \quad (1.1)$$

e

$$\vec{x}[t+1] = \vec{x}[t] + \vec{v}[t+1] \quad (1.2)$$

onde \vec{v} e \vec{x} são, respectivamente, a velocidade e a posição da partícula. C_1 e C_2 são números tal que $C_1, C_2 \in \mathbb{R}$ e representam a importância que a partícula dá a, respectivamente, informação cognitiva e a informação do enxame. r_1 e r_2 são números uniformemente distribuídos no intervalo $[0, 1]$ e ω representa a inércia da partícula.

1.3 Algoritmo FSS

Como qualquer outra técnica inteligente com base na população, o FSS [3] se beneficia muito do comportamento coletivo emergente que aumenta a capacidade de sobrevivência mútua. De um modo geral, FSS é composto de operadores que podem ser agrupados nas seguintes categorias: alimentação, natação e reprodução. Juntas, esses operadores fornecem: (i) a capacidade de pesquisa de alta-dimensional, (ii) a seleção automática entre pesquisa e exploração, e (iii) orientação auto-adaptável para as soluções procuradas.

1.4 Objetivo da medição

Será realizada a análises do desempenho dos algoritmos ABC, PSO e FSS. A análise consiste de executar cada algoritmo o mesmo número de vezes, nas melhores configurações possíveis e utilizando a mesma função de *fitness*. A métrica de comparação será o melhor *fitness* por iteração (ou conjunto de iterações).

No segundo capítulo tem-se a descrição do planejamento dos experimentos assim como o levantamento de hipóteses. No Capítulo três apresenta-se como os testes serão realizados e os resultados comparados. No quarto e último capítulo apresentam-se os resultados obtidos e finaliza-se com as conclusões.

Capítulo 2

Planejamento

Neste capítulo descreve-se o planejamento que foi realizado para a realização dos experimentos.

2.1 Função para otimização

2.1.1 Rastringin

A função Rastringin é definida tal que

$$f(\vec{x}) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)] \quad (2.1)$$

onde $A = 10$, $x_i \in [-5, 0; 5, 0]$ e n indica o número de dimensões que está se utilizando. Esta função possui mínimo global $f(\vec{x}) = 0$ quando $\vec{x} = \vec{0}$. A Figura 2.1 apresenta o formato da função quando $n = 2$

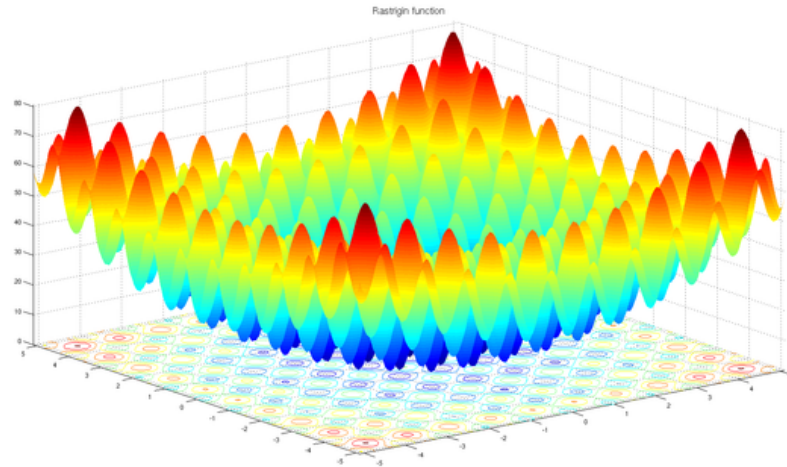


Figura 2.1: Função Rastringin para $n = 2$.

Possuindo a equação (2.1) apenas mínimo, o objetivo de se utilizar as técnicas ABC, PSO e FSS será para minimizar, isto é, encontrar o valor mínimo da função Rastringin.

2.2 Tabela de Experimentos

Os parâmetros para o ABC são:

1. **iterações** = 2000
2. **dimensões** = 20
3. Limite **superior** da Rastrin = 5.0
4. Limite **inferior** da Rastrin = -5.0
5. Quantidade de **abelhas** = 30
6. Quantidade de **escoteiras** = 2
7. Limite de **estagnação** = 100

Os parâmetros para o PSO são:

1. **iterações** = 2000
2. **dimensões** = 20
3. Quantidade de **partículas** = 30
4. **cp** = 2.05
5. **cg** = 2.05
6. Usando **inércia**
7. Usando Coeficiente de Constricção de **Clerk**

Os parâmetros para o FSS são:

1. **iterações** = 2000
2. **dimensões** = 20
3. Quantidade de **peixes** = 200
4. **peso mínimo** = 1
5. **peso máximo** = 5000
6. **passo inicio** = 10
7. **passo fim** = 0.01
8. **passo volitivo inicio** = 1
9. **passo volitivo fim** = 0.001

Temos no total 3 configurações diferentes para realizar experimentos. Cada experimento será realizado trinta vezes e, então, serão comparados.

Capítulo 3

Operação

O experimento consiste de gerar um sumário estatístico e um boxplot. Também será realizado o teste não paramétrico de Wilcoxon para comparar os casos em que uma simples inspeção visual não for suficiente para saber qual algoritmo foi melhor que outro.

O tipo de erro utilizado no teste de hipótese não paramétrico de Wilcoxon é do Tipo II, o que significa que é o caso em que a hipótese nula é aceita, mesmo que realmente ela não seja verificada. Na prática, aceitam-se os falsos-positivos.

O mesmo computador será utilizado na realização de todos os testes, variando apenas o *seed* para o gerador pseudo-aleatório. Como se trata do mesmo algoritmo sendo executado antes e depois de algumas mudanças, pode-se realizar o pareamento dos dados sem problemas.

Como saída do programa, tem-se o melhor resultado já visitado pelo enxame para cada 2000 iterações. A configuração do computador utilizado para realizar os experimentos é: Intel Core 2 Duo 2,00GHz, 4GB de memória RAM, sistema operacional Windows 7.

Cada configuração foi executada trinta vezes, no mesmo computador. Apesar do computador possuir mais de um núcleo, o programa desenvolvido não utilizou disso para acelerar seu processamento, executando assim o seu código em apenas um núcleo.

Capítulo 4

Resultados

Este capítulo apresenta os resultados obtidos para as 3 configurações investigadas.

4.1 Resultados obtidos

A figura a seguir é o *boxplot* de cada algoritmo.

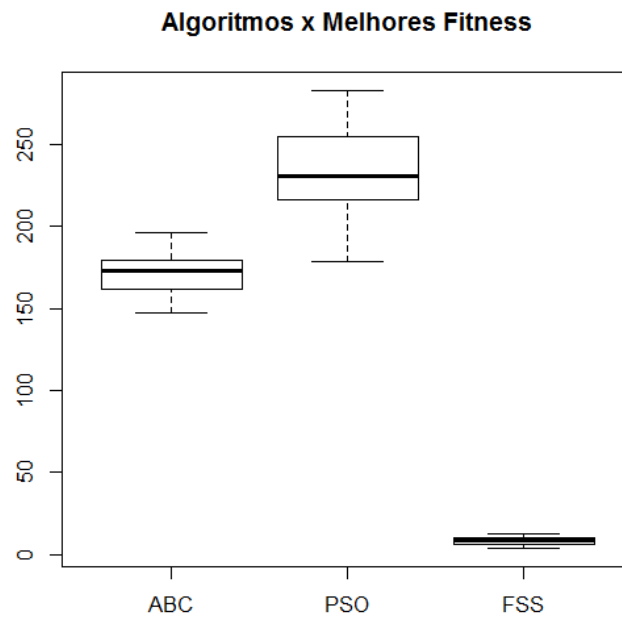


Figura 4.1: Boxplot ABCxPSOxFSS

O Sumário estatístico para cada algoritmo é apresentado a seguir: ABC

1. Mínimo :146.9
2. 1o Quartil:162.8
3. Mediana :172.5
4. Média :171.6
5. 3o Quartil:179.3

6. Máximo :196.0

PSO

1. Mínimo :178.6

2. 1o Quartil:216.6

3. Mediana :230.6

4. Média :234.2

5. 3o Quartil:254.3

6. Máximo :282.7

FSS

1. Mínimo : 3.980

2. 1o Quartil: 6.266

3. Mediana : 8.480

4. Média : 8.265

5. 3o Quartil: 9.721

6. Máximo :12.934

4.2 Comparação das Execuções

Como não houve problema para verificar o melhor e o pior resultado, não exibimos o Teste de Wilcoxon Pareado. Utilizando a linguagem R teríamos um código assim para mostrar as análises realizadas:

```
#Entrada dos dados
```

```
ABC_box <- read.table ( file = 'C:\\fitnessABC.txt ', header = TRUE)
```

```
FSS_box <- read.table ( file = 'C:\\fitnessFSS.txt ', header = TRUE)
```

```
PSO_box <- read.table ( file = 'C:\\fitnessPSO.txt ', header = TRUE)
```

```
#Sumario Estatistico
```

```
summary(ABC_box)
```

```
summary(FSS_box)
```

```
summary(PSO_box)
```

```
#Boxplot
```

```
mat <- cbind(ABC = ABC_box$fitnessABC ,
```

```
PSO = PSO_box$fitnessPSO , FSS = FSS_box$fitnessFSS)
```

```
boxplot(as.data.frame(mat), main = "Algoritmos x Melhores Fitness")
```

Capítulo 5

Conclusão

A partir dos resultados do boxplot, podemos observar que o algoritmo que apresenta o pior desempenho é o PSO, já que trata-se de um problema de otimização multi-modal. Em segundo lugar ficou o ABC que é capaz de resolver problemas multimodais. A melhor implementação é o FSS e apresentou resultados bem melhores que as outras.

De modo geral, é possível utilizar esses algoritmo para a minimização ou maximização de problemas monomodais e multimodais, em espaços de busca numéricos contínuos.

Referências Bibliográficas

- [1] C. Karaboga, D.; Ozturk. A novel clustering approach: Artificial bee colony (abc) algorithm. *Appl. Soft Comput.*, 11(1):652–657, 2011.
- [2] J.; Eberhart. R. Kennedy. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1995.
- [3] F. B. Filho, C.; L. Neto. A novel search algorithm based on fish school behaviour. *Proceedings of IEEE System Man and Cybernetics Society*, 2008.