Morcego em GPU

Contextualização

Aplicar o algorítmo do morcego em GPU.

C CUDA na Amazon.

Variáveis:

- Frequência: proporcional a intensificação
- Amplitude: proporcional a diversificação
- Posição infuenciada pela amplitude

Como?

Primeiramente separar a população paralelamente (DAO, 2015).

Em um segundo passo tornar o processo de varredura paralelo?

Modelagem matemática

Pseudo Código Original

Bat Algorithm

Objective function $f(\mathbf{x})$, $\mathbf{x} = (x_1, ..., x_d)^T$
Initialize the bat population $\mathbf{x}_i$ $(i = 1, 2, ..., n)$ *and* $\mathbf{v}_i$
Define pulse frequency f_i *at* $\mathbf{x}_i$
Initialize pulse rates r_i *and the loudness* A_i
while *(t <Max number of iterations)*
Generate new solutions by adjusting frequency,
and updating velocities and locations/solutions [equations (2) to (4)]
    if $(rand > r_i)$
    Select a solution among the best solutions
    Generate a local solution around the selected best solution
    end if
    Generate a new solution by flying randomly
    if $(rand < A_i \ \& \ f(\mathbf{x}_i) < f(\mathbf{x}_*))$
    Accept the new solutions
    Increase r_i *and reduce* A_i
    end if
Rank the bats and find the current best $\mathbf{x}_*$
end while
Postprocess results and visualization

Figure 1:pseudo-code.png

Pseudo Código Jelson's

Parâmetros: n, α, λ
Inicializa morcegos $\vec{x}_i$
Avalia $f(\vec{x}_i)$ para todos os morcegos
Atualiza melhor morcego $\vec{x}_*$
while critério de parada não atingido **do**
  for $i = 1$ **to** n **do**
    $f_i = f_{min} + (f_{max} - f_{min})\beta, \beta \in [0, 1]$
    $\vec{v_i}^{t+1} = \vec{v_i}^{t} + (\vec{x_i}^{t} - \vec{x_*}^{t})f_i$
    $\vec{x}_{temp} = \vec{x_i}^{t} + \vec{v_i}^{t+1}$
    if $rand < r_i$, $rand \in [0, 1]$ **then** {Faz busca local}
      $\vec{x}_{temp} = \vec{x}_* + \epsilon A_m, \epsilon \in [-1, 1]$
    end if
    Realiza pertubação em uma dimensão de $\vec{x}_{temp}$
    if $rand < A_i$ **or** $f(\vec{x}_{temp}) \leq f(\vec{x}_i)$, $rand \in [0, 1]$ **then** {Aceita solução temporária}
      $\vec{x}_i = \vec{x}_{temp}$
      ${r_i}^{t+1} = 1 - exp(-\lambda t)$
      ${A_i}^{t+1} = \alpha {A_i}^{t}$
    end if
    Atualiza melhor morcego $\vec{x}_*$
  end for
end while
Pós-processamento

Figure 2:pseudo-code-v2.png

Funções de Benchmarks

- Ackley
- Griewank
- Rastrigin
- Sphere

Ackley

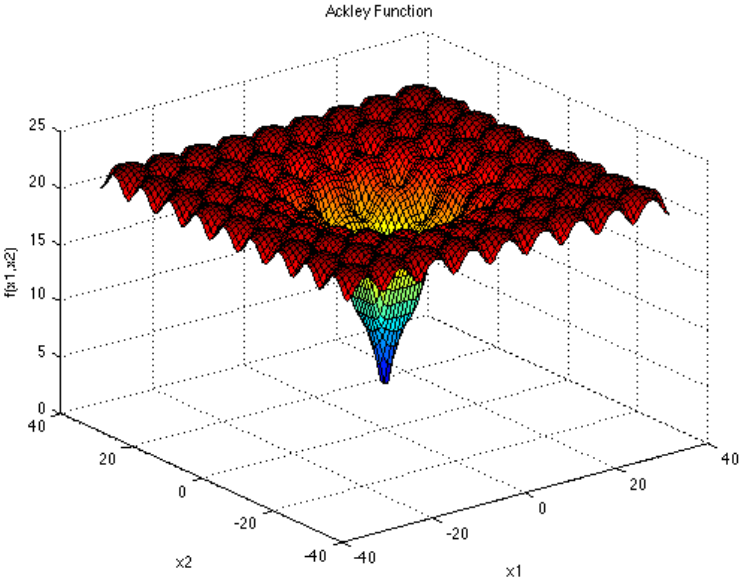

Figure 3:ackley.png

Griewank

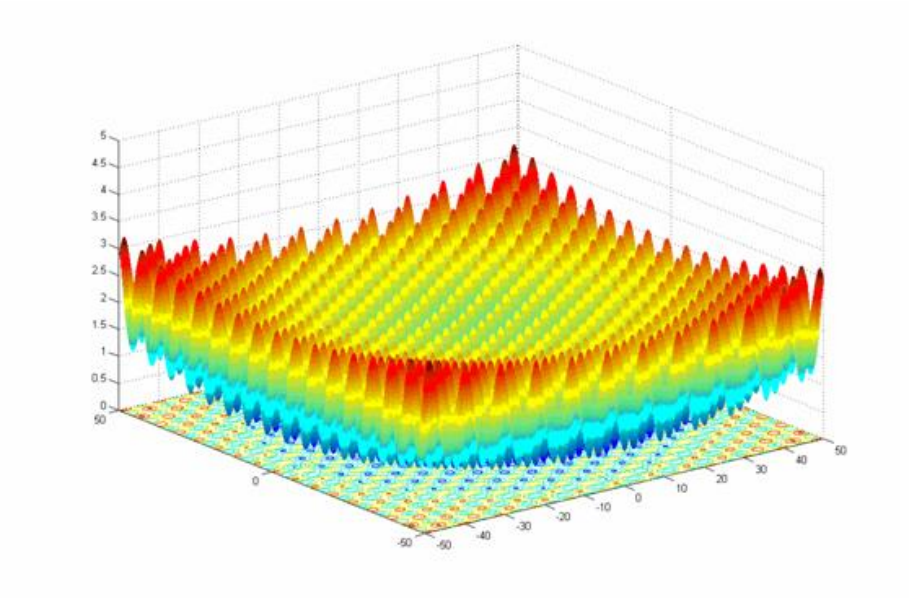

Figure 4:griewank.jpg

Rastrigin

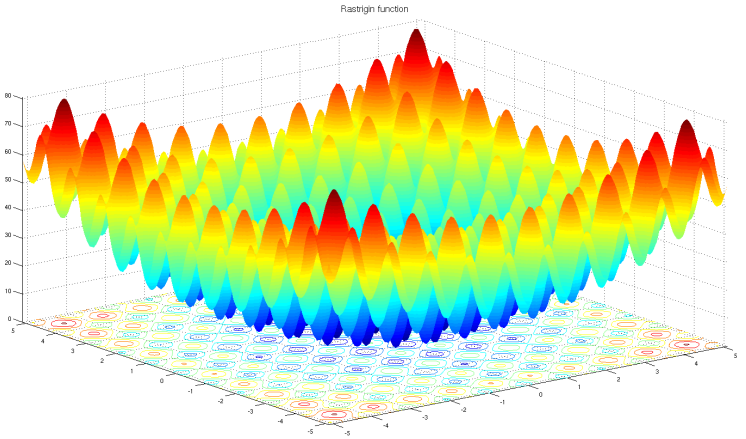

Figure 5:rastringin.png

Sphere

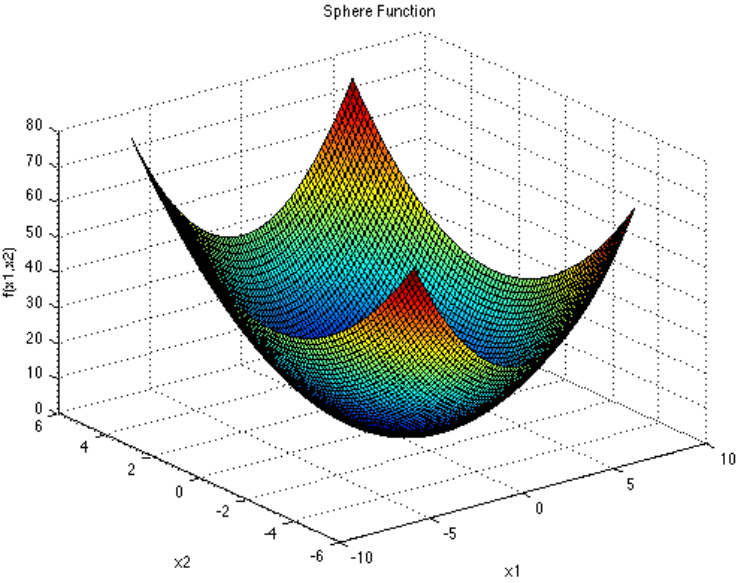

Figure 6:sphere.png

Benchmarks Yang's 2010

- Ackley
- De Jong
- Easoms
- Eggcrate
- Griewank
- Michalewicz
- Rastrigin
- Rosenbrocks
- Schewefels
- Schuberts

Benchmarks Jelson's

- Ackley
- Griewank
- Rastrigin
- Sphere

Benchmarks Adis

- Ackley
- Griewank
- Rastrigin
- Rosenbrok
- Sphere

Benchmarks Li

- Ackley
- Eliptic
- Rastrigin
- Rosenbrocks
- Schwefel
- Sphere

Referências

- Bat Algorithm: An Overview and it's Applicaions, S Induja, International Journal of Advanced Research in Computer and Communication Engineering, 2016
- A New Metaheuristics Bat-Inspired Algorithm, Xin-She Yang, Department of Engineering, Cambridge, 2010
- Análise de Sensibilidade dos Parâmetros do Bat Algorithm e Comparação de Desempenho, Jelson A. Cordeiro, Rafael Stubs Parpinelli, Heitor Silvério Lopes
- Parallel bat algorithm for optimizing makespan in job scheduling problems, Thi-Kien Dao, Tien-Szu Pan, Trong-The Nguyen, Jeng-Shyang Pan,2015, Springer Sience Review
- Improved Hibridized Bat algorithm for Global Numerical Optimization, Adis Alihodzic, Milan Tuba, 2014, International Conference on Computer Modeling and simulation
- Benchmark Functions for the CEC'2013 Special Session and Competition on Large-Scale Global Optimization, Xiaodong Li, Ke Tang, Mohammad N. Omidvar, Zhenyu Yang, Kai Qin, 2013