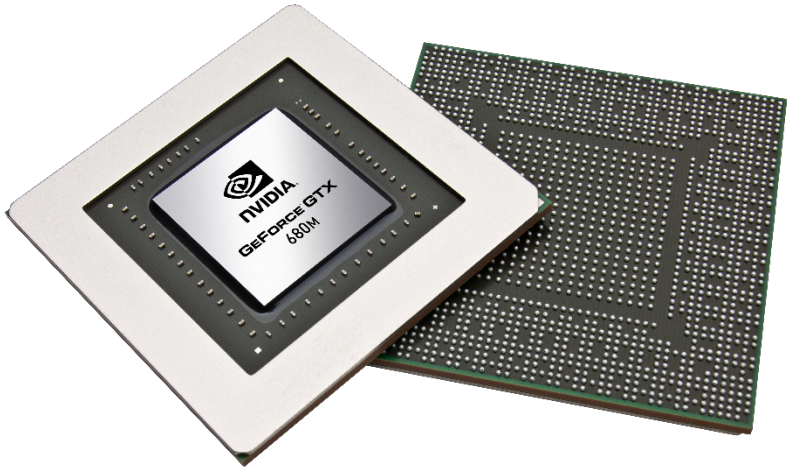


Filipo Novo Mó
Omar Andres Carmona Cortes
César Augusto Missio Marcon

Abordagem Paralela da Evolução Diferencial em GPU

ERAD 2015
XV Escola Regional de Alto Desempenho
22 a 24 de Abril de 2015
Gramado – RS - Brasil

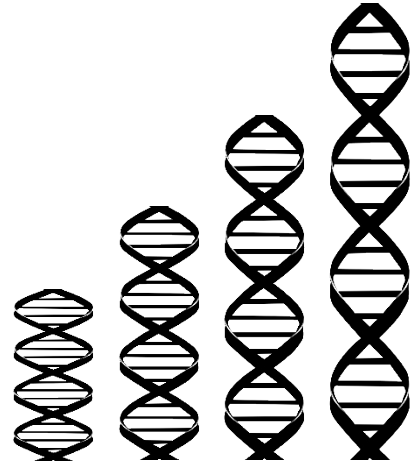
www.filipomor.com

O Algoritmo de Evolução Diferencial

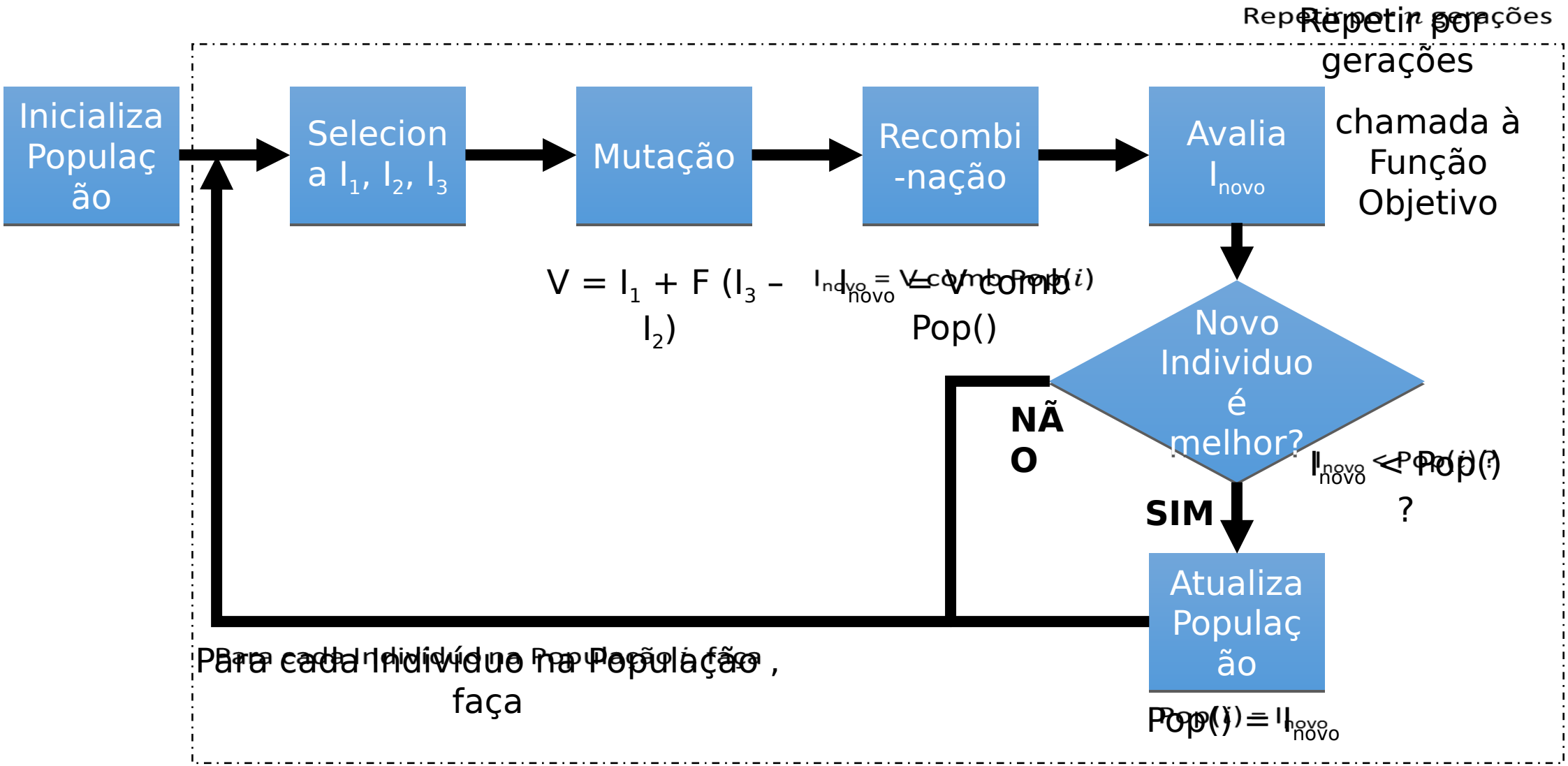
- Faz parte do grupo de algoritmos evolutivos.
- Adequado a problemas de natureza contínua e NP Completo.
- O objetivo é encontrar um vetor X tal que se minimize ou maximize uma função *(função objetivo)*.
- A principal vantagem da ED é que o algoritmo pode tratar funções objetivo que são descontinuas, não-diferenciáveis, não-lineares ou dependente de muitas variáveis.

O Algoritmo de Evolução Diferencial

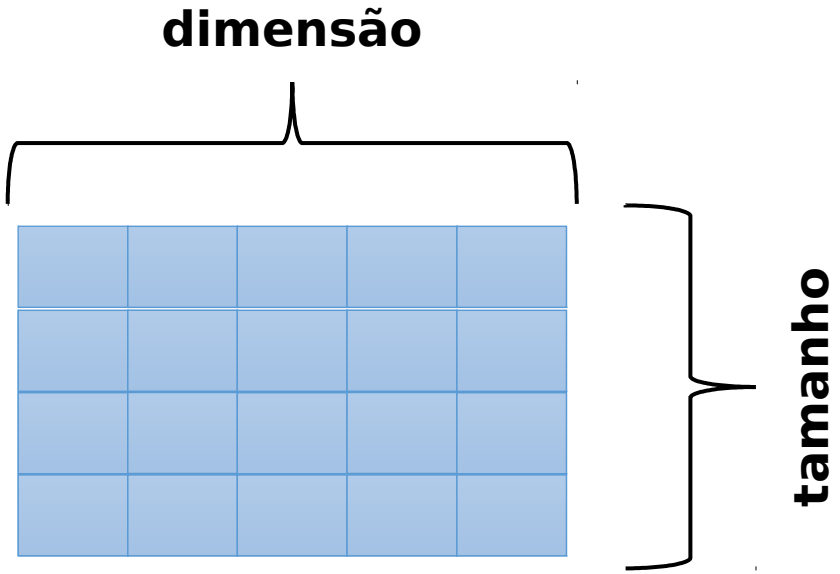
```
1    Inicializa População
2    Avalia(População)
3    Enquanto critério de parada não alcançado
4     Para cada individuo i na população (Pop) faça
5           Selecionar três individuos
6            V ← individuo3 + F * (individuo1 – individuo2)
7            Novo_individuo ← cruzamento entre V e Popi
8            Se o fitness(Novo_individuo) < fitness(Popi)
9               Substitui Popi por Novo_individuo
10           Fim-se
11       Fim-para
12   Fim-Enquanto
```



O Algoritmo de Evolução Diferencial



O Algoritmo de Evolução Diferencial

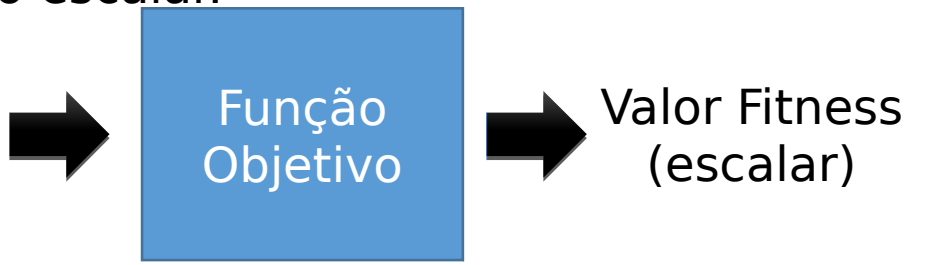


População

- Cada linha representa um indivíduo (ou solução proposta).
- A dimensão representa a discretização do problema.

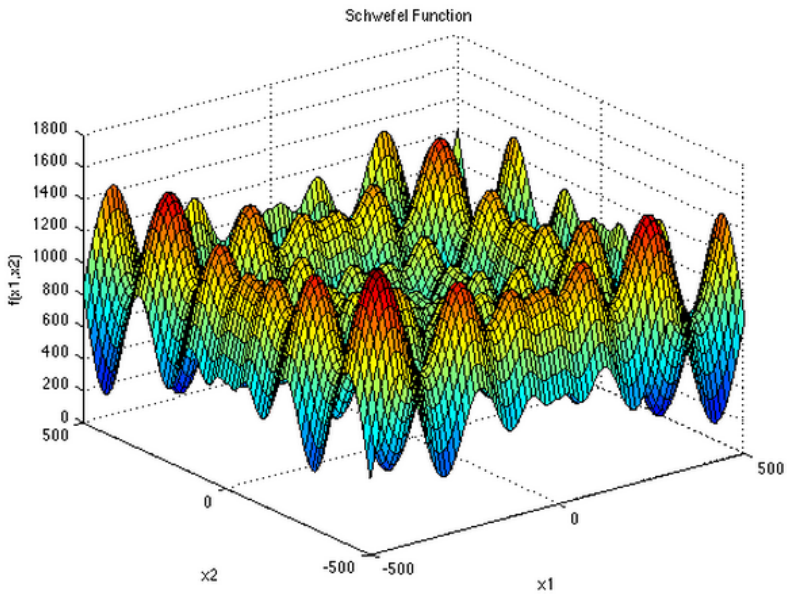
- A função objetivo avalia o Individuo a partir do valor das suas células (dimensão), gerando um resultado escalar.

Indivíduo



O Algoritmo de Evolução Diferencial

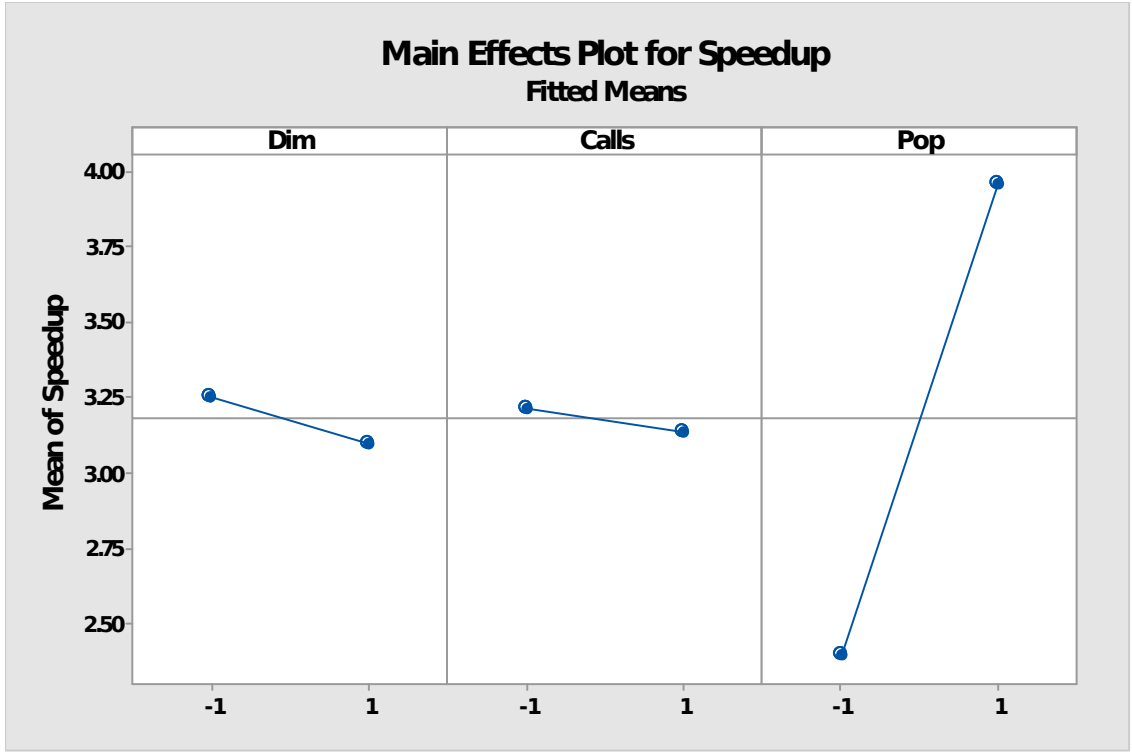
- Para avaliar a implementação foi utilizada a função de *benchmark* de *Schwefel*.
  - Possui vários mínimos locais.
  - Dimensão: d
  - O mínimo global é conhecido:
  - $f(x^*) = 0$, com $x^* = (420.9686, \ldots, 420.9686)$

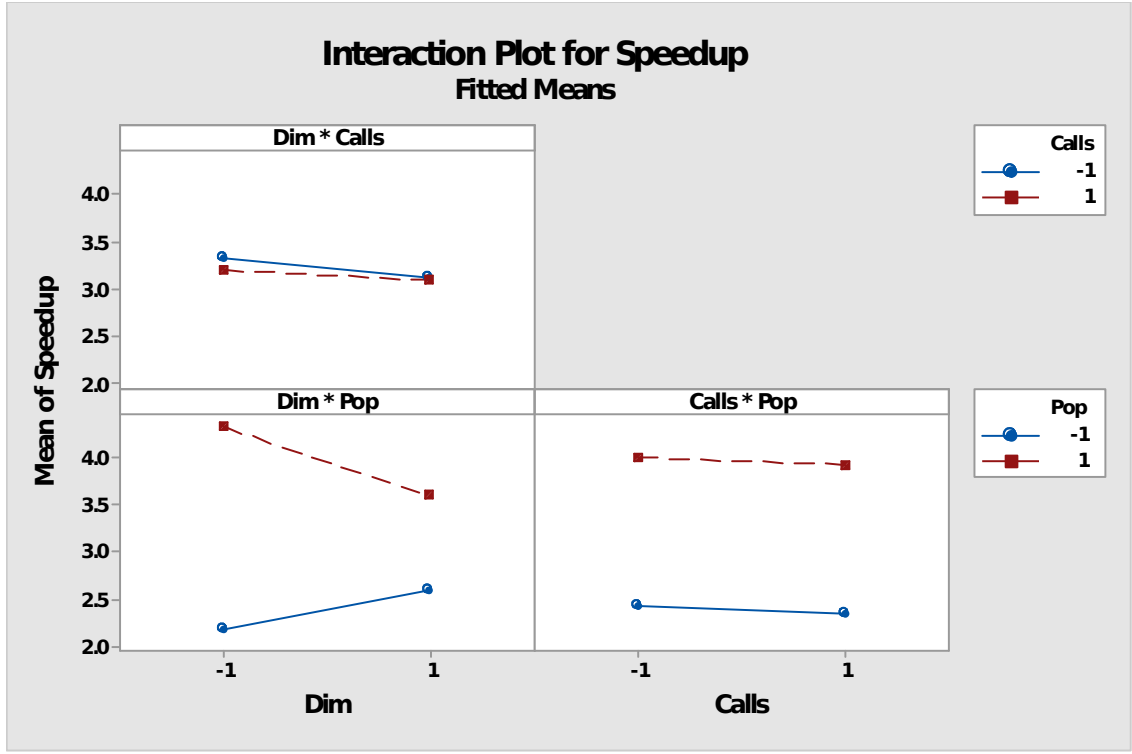


$$f(\mathbf{x}) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{|x_i|})$$

Implementação e Testes

- Parallel Computing Toolbox para GPU no MatLab.
- Equipamento:
  - Placa de Video (GPU) NVIDIA GTS 450.
  - CPU Intel i7 com 3.4Ghz.
  - 64Gb RAM.
  - Windows 7 64 bits.
- Testadas variações em:
  - Tamanho da população
  - Dimensão (quantidade de colunas)
  - Quantidade de chamadas a função objetivo.
- Cada experimento foi executado 30 vezes.





Dim	Calls	Pop	Speedup	Dim	Calls	Pop	Speedup
-1	-1	-1	2.242328005	-1	-1	+1	**4.402031153**
+1	-1	-1	2.631392488	+1	-1	+1	3.59878617
-1	+1	-1	2.139425192	-1	+1	+1	4.246680717
+1	+1	-1	2.571705205	+1	+1	+1	3.604751521

Valores testados:

- Dimensão: {500, 2000}
- Chamadas: {10^5, 10^6}
- População: {100, 2000}

Nas legendas:

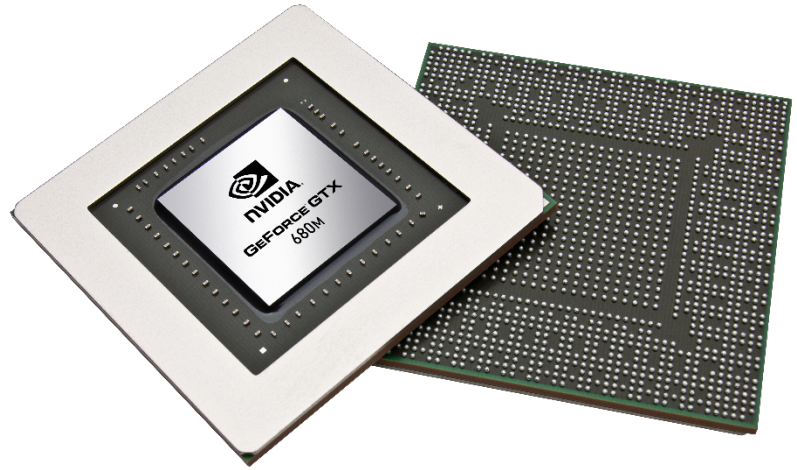
- baixo = -1

Conclusões

- Para a função objetivo utilizada, os maiores *speedups* foram obtidos com as maiores populações (maior do que 4).
- A dimensão (quantidade de colunas) parace ter pouco impacto no *speedup* da implementação paralela.
- Considerando que os parâmetros de entrada são definidos pelo problema a ser processado, a identificação deste tipo de relação serve de insumo ao dimensionamento de novas aplicações baseadas em

Próximos Passos

- Implementação do algoritmo em CUDA C.
- Generalização do Algoritmo para uso de Funções Objetivo diversas.
- Geração modulo de visualização.



OBRIGADO!

Filipo Novo Mó
Omar Andres Carmona Cortes
César Augusto Missio Marcon

Abordagem Paralela da Evolução Diferencial em GPU

ERAD 2015
XV Escola Regional de Alto Desempenho
22 a 24 de Abril de 2015
Gramado – RS - Brasil

www.filipomor.com