Introdução à otimização evolucionária com objetivos múltiplos

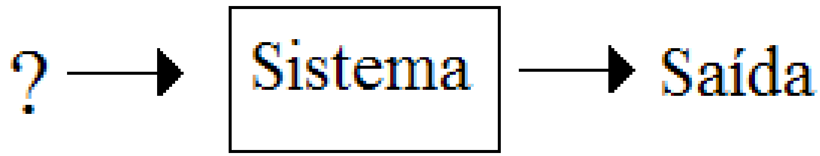
Aluízio Fausto Ribeiro Araújo
Arthur Gonçalves de Carvalho

Roteiro

- Otimização
  - Objetivo Único
  - Objetivos Múltiplos
- Geração de Soluções
  - Métodos Clássicos
  - Otimização Evolucionária
- Métricas
  - Corretude
  - Diversidade
- Softwares & Frameworks
- Conclusões & Trabalhos Futuros

Otimização

- Procedimento de busca por soluções ótimas
- Qualidade mensurada de acordo com objetivo(s) [Funções]
  - Qual a entrada fornecida ao sistema que o leva a uma saída ótima?



Otimização – Objetivo Único

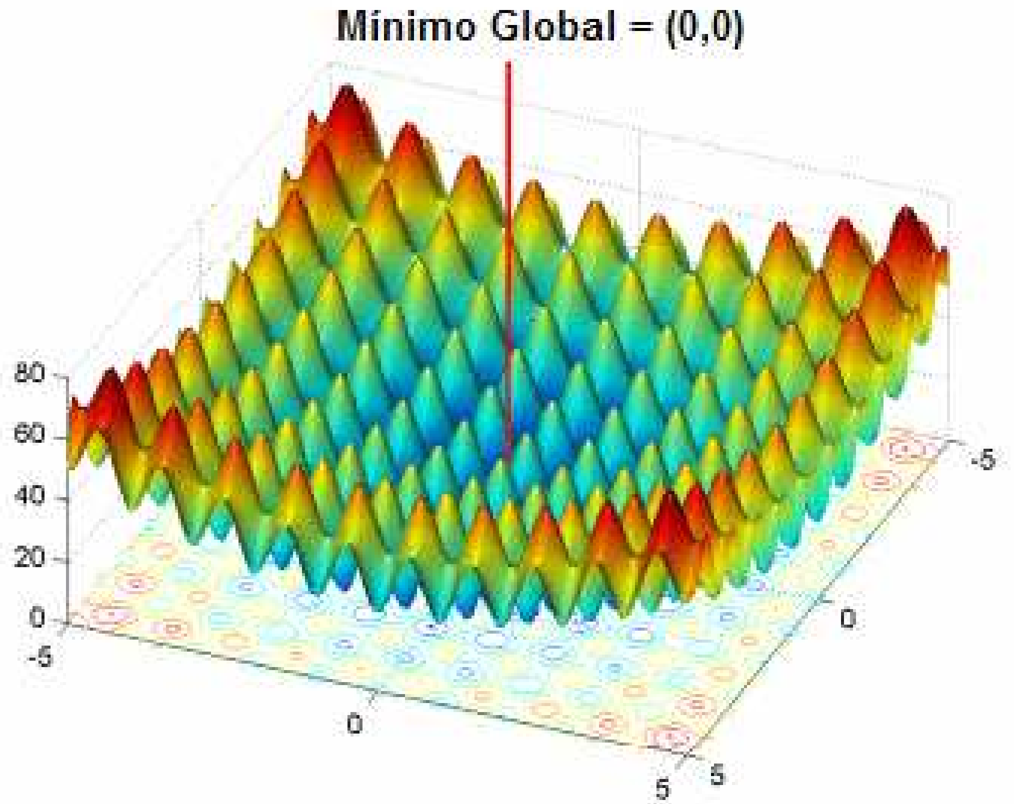
- Resulta na busca pela <u>melhor</u> solução (Mínimos ou Máximos Globais)

- Técnicas baseadas em gradientes, algoritmos evolucionários, busca com heurísticas...

- Formalização:

Dada uma função $f : A -> R$ *desejamos encontrar um elemento em A tal que* $f(x_0) \leq f(x)$ *para todo* $x \in A$

Otimização – Objetivo Único



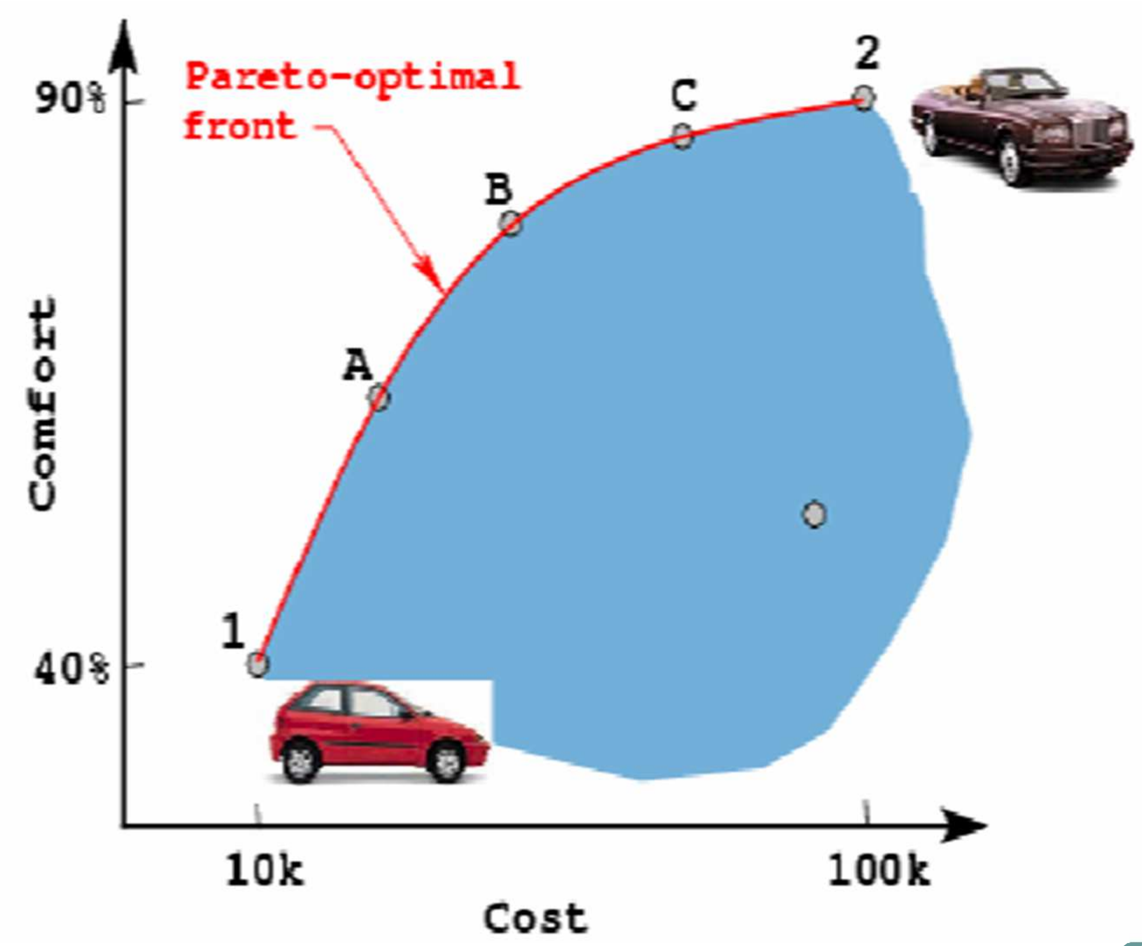
Minimizar $Ras(x) = 20 + x_1^{\ 2} + x_2^{\ 2} - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$

Otimização – Múltiplos Objetivos

- Exemplo:

Multi-Objective
Optimization
Using
Evolutionary
Algorithms

Kalyanmoy Deb



Otimização – Múltiplos Objetivos

- Objetivos múltiplos e possivelmente conflitantes:
  - Decisões;
  - Estruturas.

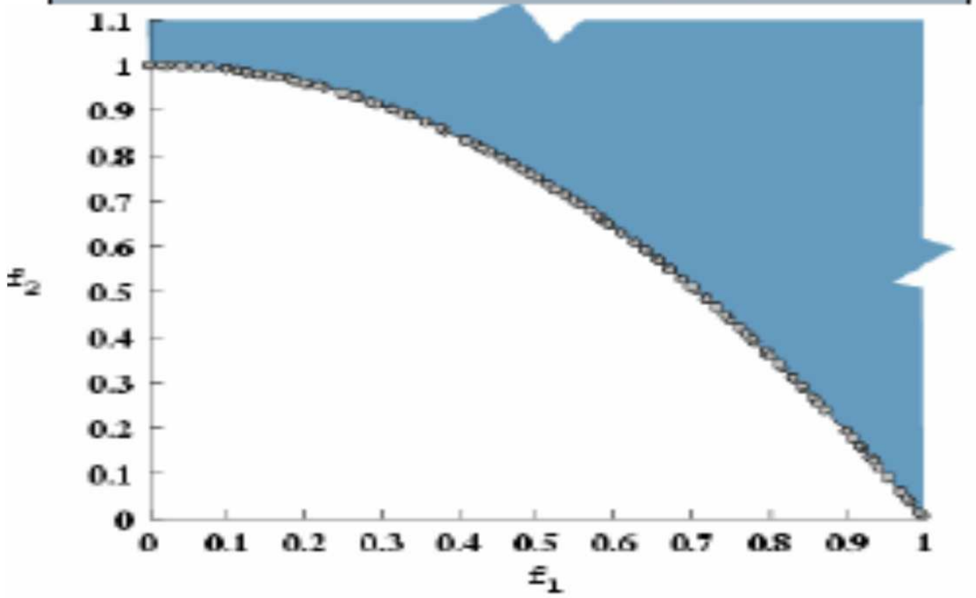
- Diversas soluções plausíveis (Trade-offs).

Otimização – Múltiplos Objetivos

(Min) $f_1(x) = x_1$

(Min) $f_2(x) = g\left[1 - \left(\frac{f_1}{g}\right)^2\right]$

Where $g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$

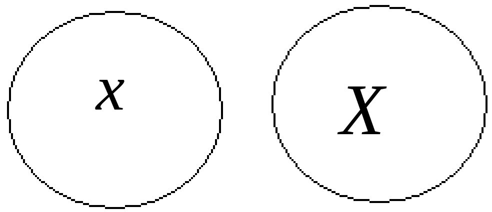


Otimização – Múltiplos Objetivos

- Formalização:

$$
\begin{array}{ll}
\text{Min/Max} & (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\
\text{Restrições} & g_j(\mathbf{x}) \geq 0 \\
 & h_k(\mathbf{x}) = 0 \\
 & \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}
\end{array}
$$

$$X = \{x_1, x_2, \ldots, x_n\}$$



Espaço de Decisão

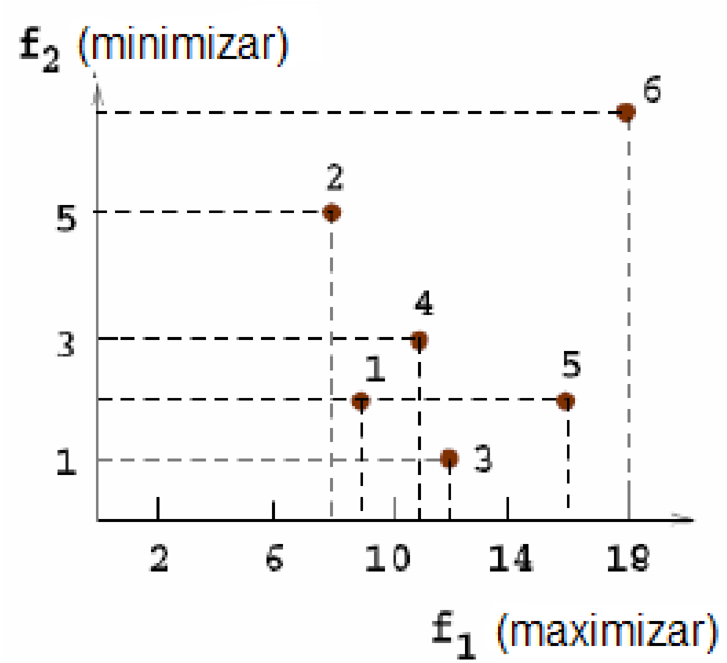
Espaço de Busca

Otimização – Múltiplos Objetivos

- Procedimento "ideal" na otimização com múltiplos objetivos:
  - Encontrar soluções ótimas;
  - Encontrar soluções esparsas.
- Tomada de Decisão Multi-critério (*Multiple criterion decision-making* - MCDM)

Otimização – Múltiplos Objetivos

- Soluções ótimas são obrigatoriamente soluções não- dominadas:
- X domina Y se:
  - X não é pior que Y em todos os objetivos
  - X é melhor que Y em ao menos um objetivo



Otimização – Múltiplos Objetivos

- Pontos Ótimos de Pareto = Conjunto de todas as soluções não-dominadas do espaço.

- Formalização:

Um ponto x será ótimo de Pareto se e somente se não existe outro ponto x tal que F(x) < F(x*) e exista ao menos uma função objetivo tal que* $f_i(x^*) < f_i(x)$

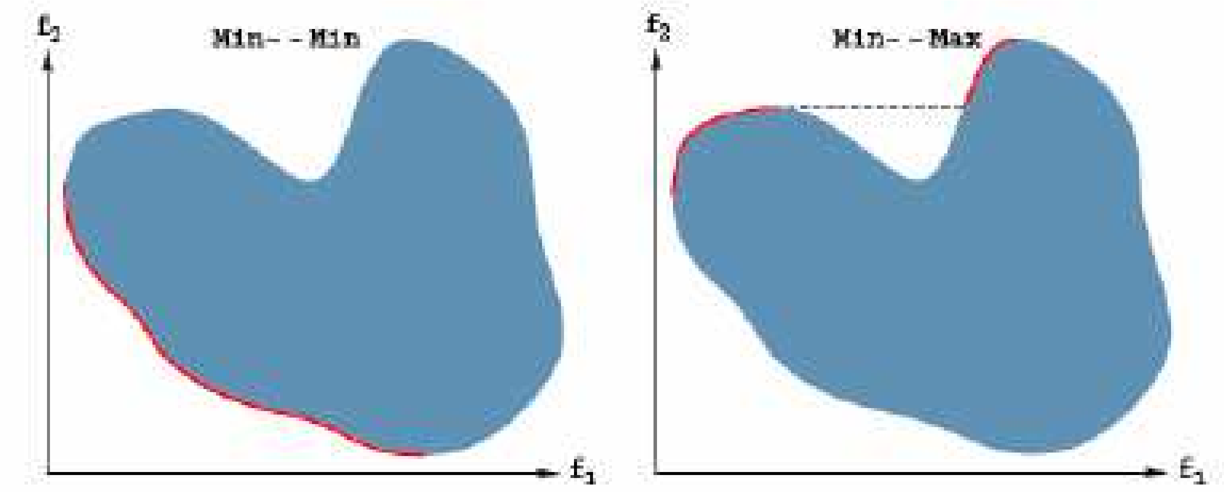
A1

Slide 12

A1 Arthur; 23/11/2007

Otimização – Múltiplos Objetivos

- Fronteira de Pareto = Curvas formadas pelo conjunto de pontos ótimos de Pareto.



Otimização – Múltiplos Objetivos

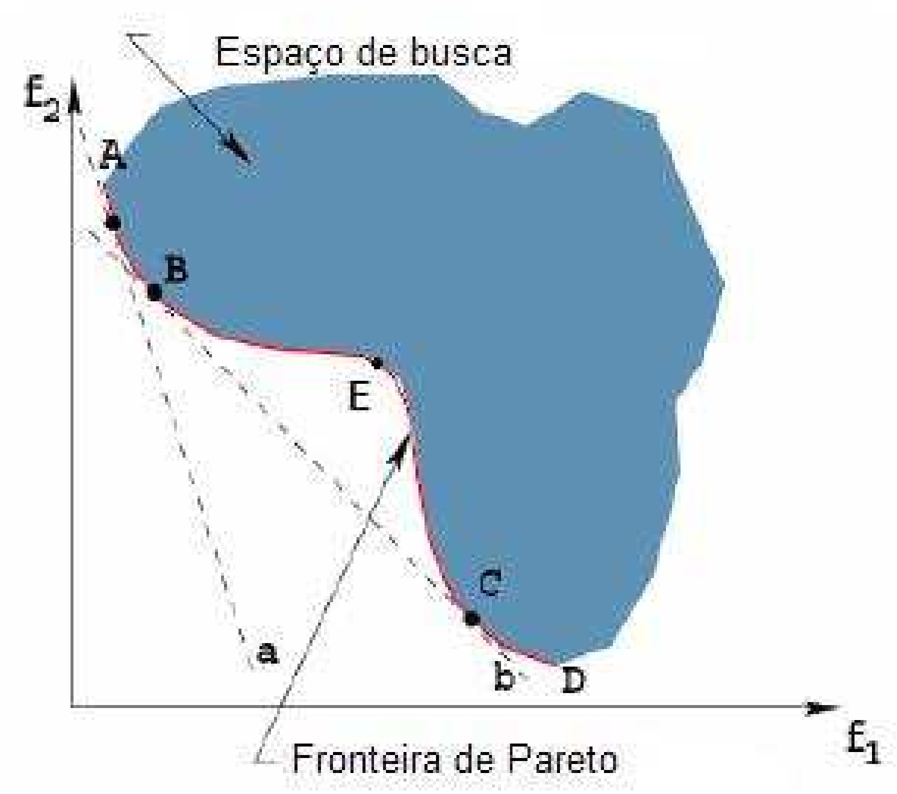
- Estratégia da otimização com múltiplos objetivos:
  - Encontrar um conjunto de soluções mais próximas da fronteira de Pareto;
  - Encontrar o conjunto de soluções mais diversas possível.

Geração de Soluções

- Métodos clássicos: Transformação do problema
  - Soma ponderada
  - *E*-constraint
- Métodos modernos: Populações
  - Algoritmos Genéticos
  - Teoria dos Jogos

Métodos Clássicos

- Soma Ponderada



Minimizar/Maximizar

$$F(x) = \sum_{m=1}^{M} W_m f_m(x),$$

Relativo às restrições

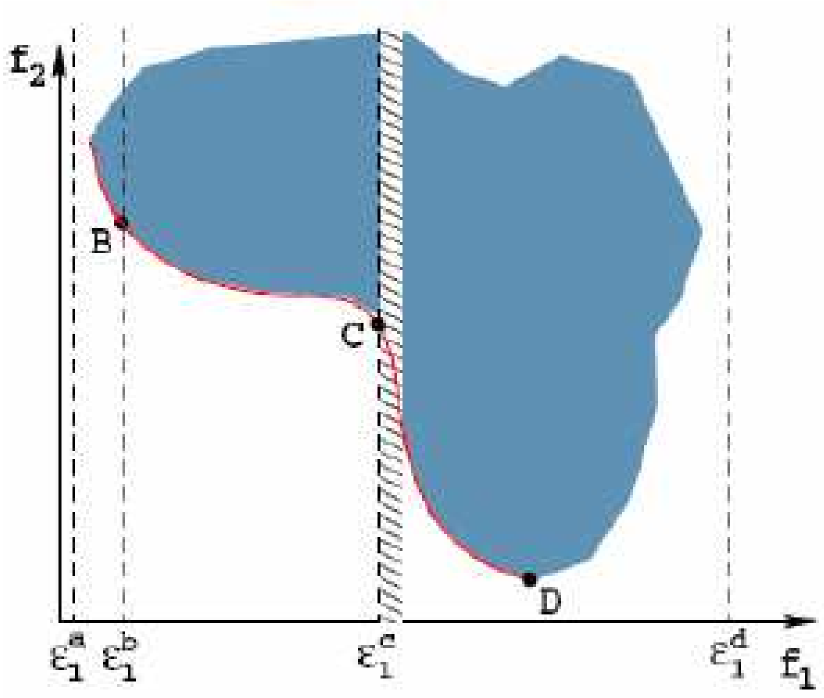
$$g_j(x) \geq 0, \quad j = 1,2,\ldots, J;$$

$$h_k(x) = 0, \quad k = 1,2,\ldots, K;$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1,2\ldots n$$

Métodos Clássicos

- *E-Constraint*



Minimizar/Maximizar

$$f_\mu(x)$$

Relativo às restrições

$$f_m(x) \le \varepsilon_m, \quad m = 1,2,\ldots M \quad e \quad m \ne \mu$$

$$g_j(x) \ge 0, \quad j = 1,2,\ldots, J;$$

$$h_k(x) = 0, \quad k = 1,2,\ldots, K;$$

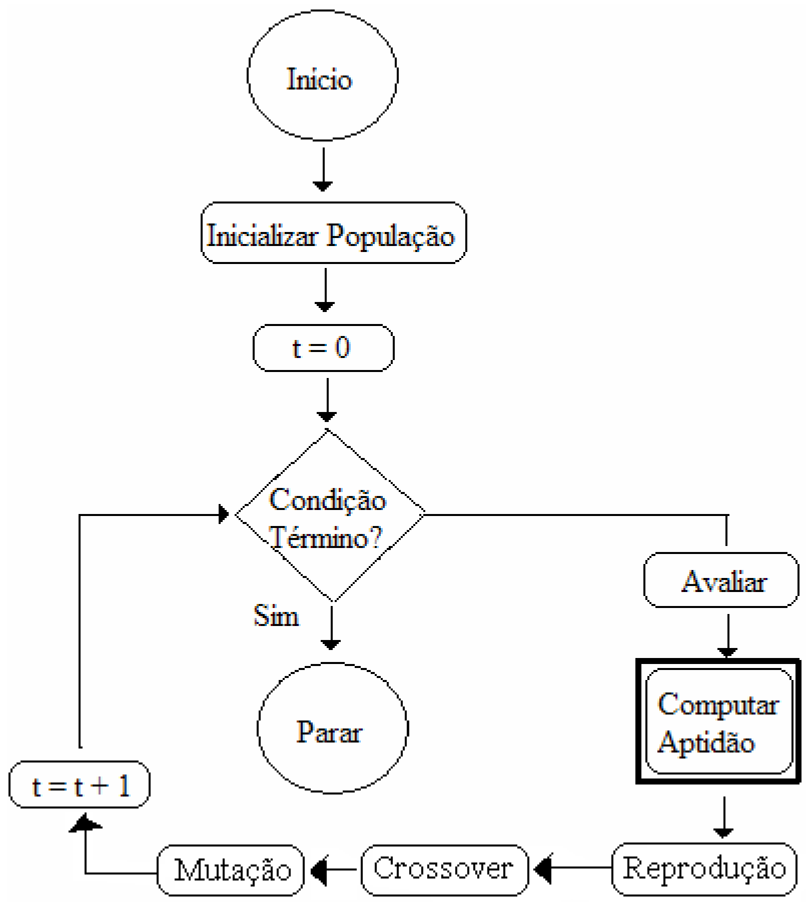
$$x_i^{(L)} \le x_i \le x_i^{(U)}, \quad i = 1,2\ldots n$$

Otimização Evolucionária

- Computação Evolucionária: Abordagem inspirada no processo evolucionário aqui empregada para solucionar problemas de otimização. Ela considera:
  - Aptidão (indivíduos mais aptos);
  - Tentativa-e-erro (caráter estocástico);
  - Populações;
  - Operadores evolucionários (Seleção e Variação).

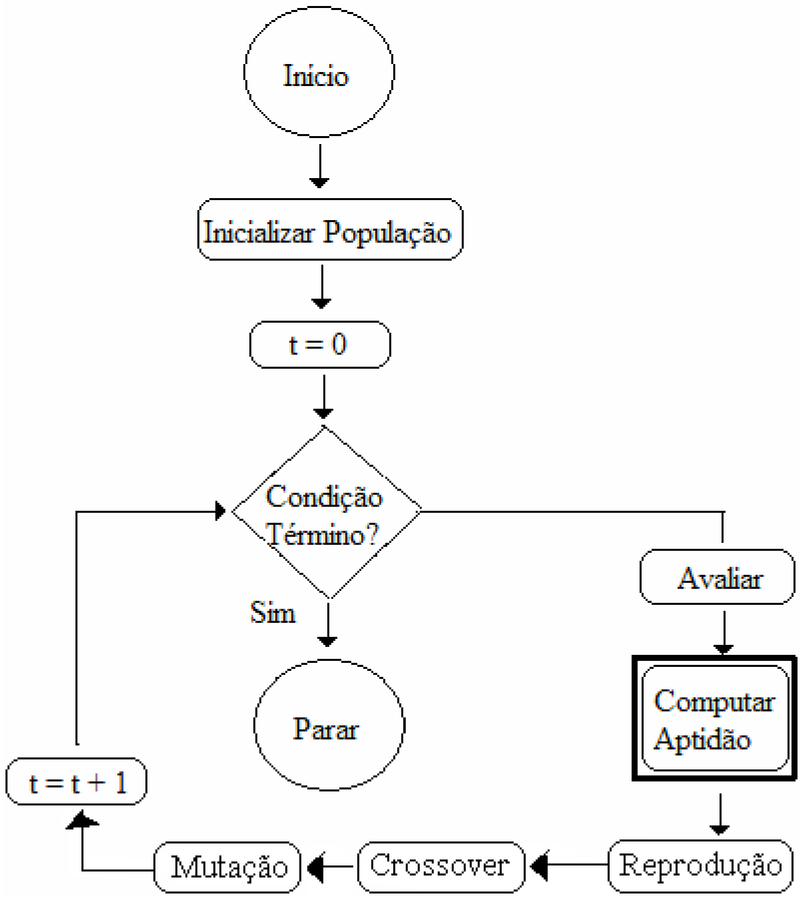
Otimização Evolucionária

1. Representação
2. Inicialização
3. Função de aptidão
4. Seleção
5. Operadores genéticos



Otimização Evolucionária

6. Sobreviventes (Próxima Geração)
7. Condição de término



Otimização Evolucionária

- Metas Genéricas de um MOEA
  - Meta 1: Preservação de pontos não-dominados (elitismo X não elitismo);
  - Meta 2: Progressão ou condução da fronteira do Pareto conhecida (FP_C) para a fronteira do Pareto verdadeira (FP_V);
  - Meta 3: Geração e manutenção de diversidade de pontos sobre a Fronteira do Pareto;
  - Meta 4: Entrega de número limitados do FP_C para o tomador de decisões.

Otimização Evolucionária

- Passos de Projeto do Procedimento Geral (meta-nível) de um MOEA
  - Passo 0: Defina o MOP
    - Função objetivo, representação, restrições, integração ao MOEA.
  - Passo 1: O MOEA gera a fronteira do Pareto (FP_C)
    - Determinação de conjuntos não-dominados (FP_{conh})s a cada geração, convergindo para próximo de FP_V, executada um número de vezes que permita atingir alguma métrica.
  - Passo 2: O MOEA busca gerar distribuição uniforme ao longo da fronteira do Pareto
    - Isto ocorre no final de cada geração.

Otimização Evolucionária

- Passos de Projeto do Procedimento Geral (meta-nível) de um MOEA (continuação)
  - Passo 3: Selecione alguns pontos ótimos na FP
    - Apresente estes pontos ao tomador de decisão.
  - Passo 4: Determine o conjunto de Pareto ótimo (FP_{conh})
    - Implemente valores de variáveis de decisão como selecionado pelo tomador de decisões.
  - Passo 5: Visualize o processamento do algoritmo e seus resultados como apropriado para melhorar desempenho do MOEA
    - Considere eficiência e efetividade.

Otimização Evolucionária

- Operadores dos MOEAs organizados para atingir seus 4 objetivos primários:
  - Meta 1: Preservação de pontos não-dominados
    - Ordenamento baseado em dominância;
    - Abordagens Não-Pareto X Pareto;
    - Arquivamento e elitismo da população de cromossomos.
  - Meta 2: Progressão para FP_v
    - Convergência para FP_v;
    - Geração de pontos fenótipos não-dominados;
    - Manipulação de blocos construtivos explícita X Não-explícita;
    - Métricas de desempenho qualitativo X quantitativo e comparações visuais;
    - Modelos MOEA probabilísticos, incorporação de busca local, etc.

Otimização Evolucionária

- Operadores dos MOEAs organizados para atingir seus 4 objetivos primários (cont):
  - Manutenção de diversidade nos pontos do FP_{conh}
    - Preservação de diversidade;
    - Emprego de nichos e compartilhamento de aptidões a engatinhar na fronteira do Pareto;
    - Pontos diversos, uniformemente distribuídos no FP_{conh}.
  - Meta 4: Disponibilização de um número limitado de pontos do Pareto para o tomador de decisões.

Otimização Evolucionária

- Compartilhamento de Aptidão (*Fitness sharing*)
  - Motivação = ENCONTRAR e MANTER múltiplas soluções ótimas.

- Cenário
  - População = N
  - Soluções ótima = q ($q << N$)
  - m_i = Ocupação de i

Otimização Evolucionária

- Função hipotética

$$f_i' = \frac{f_i}{m_i}, 1 \leq i \leq q$$

- Problema
  - Identificar soluções pertencentes aos ótimos (nichos).

Otimização Evolucionária

- Goldberg
  - Estimar número de soluções de cada nicho Nc

$$Sh(d) = \begin{cases} 1 - \left(\dfrac{d}{\sigma_{share}} \right)^{\alpha}, se \quad d \leq \sigma_{share} \\ 0, caso \quad contrário \end{cases}$$

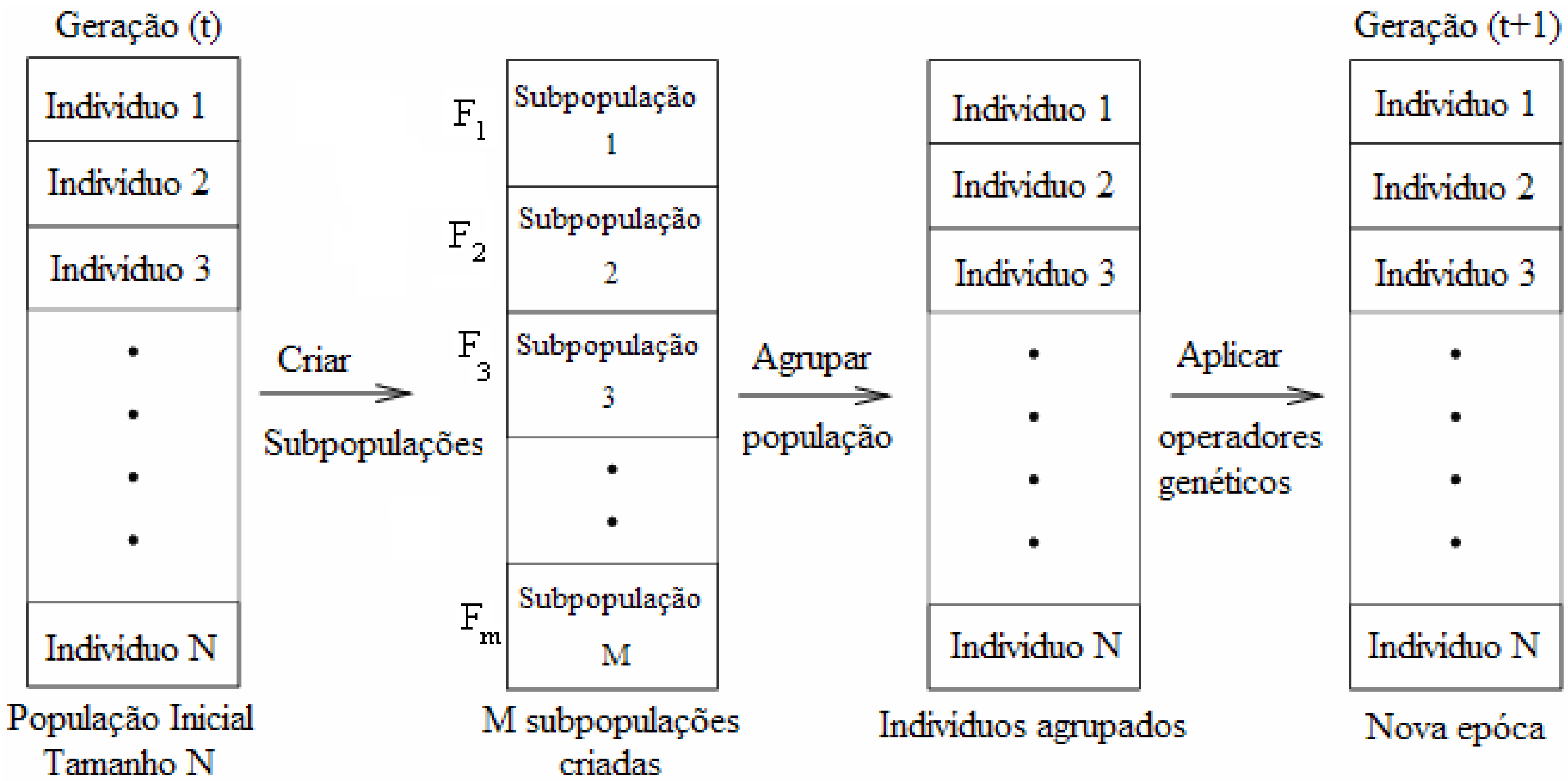
  - Logo,

$$N_i = \sum_{j=1}^{N} Sh(d_{ij}) \text{ e } f_i' = \frac{f_i}{N_i}$$

Otimização Evolucionária

- VEGA – Vector Evaluated Genetic Algorithm
  - David Schaffer (1984) – PhD thesis.
  - Pioneiro.
  - Não utiliza conceitos relacionados a ótimos de Pareto.
  - Divide população em blocos.
  - Cada bloco é reproduzido com uma função objetivo.

Otimização Evolucionária



Otimização Evolucionária

- VEGA
  - VANTAGENS
    - Intuitivo
  - DESVANTAGENS
    - Soluções tendenciosas

Otimização Evolucionária

- MOGA - Multi-Objective Genetic Algorithm (1993)
  - Carlos Fonseca e Peter J. Fleming (1993).
  - Pioneiro
    - Soluções não-dominadas.
  - Gera ranking de soluções.
  - Não garante diversidade.

Otimização Evolucionária

- Para cada solução em uma dada geração:
  - Calcule o número de soluções que dominam a solução considerada;
  - Determine o rank de acordo com este cálculo.
- Organize a população de acordo com um rank

$$r_i = 1 + n_i$$

- Ordene em ordem crescente.
- Assinale aptidão a cada indivíduo através de interpolação do melhor (rank 1) para o pior (último rank) como aptidão compartilhada.

Otimização Evolucionária

- MOGA
  - VANTAGENS
    - Intuitivo.
    - Encontra várias soluções não-dominadas.
  - DESVANTAGENS
    - Não garante diversidade.
    - Computacionalmente custoso.

Otimização Evolucionária

- NSGA-II – Elitist Nondominated Sorting Genetic Algorithm
  - Seguiu o NSGA de Srinivas e Deb (1994).
  - Deb et al. propuseram o NSGA-II (2000, 2002).
  - Algoritmo para encontrar múltiplas soluções ótimas do Pareto para resolver um problema de otimização multi-objetivo.

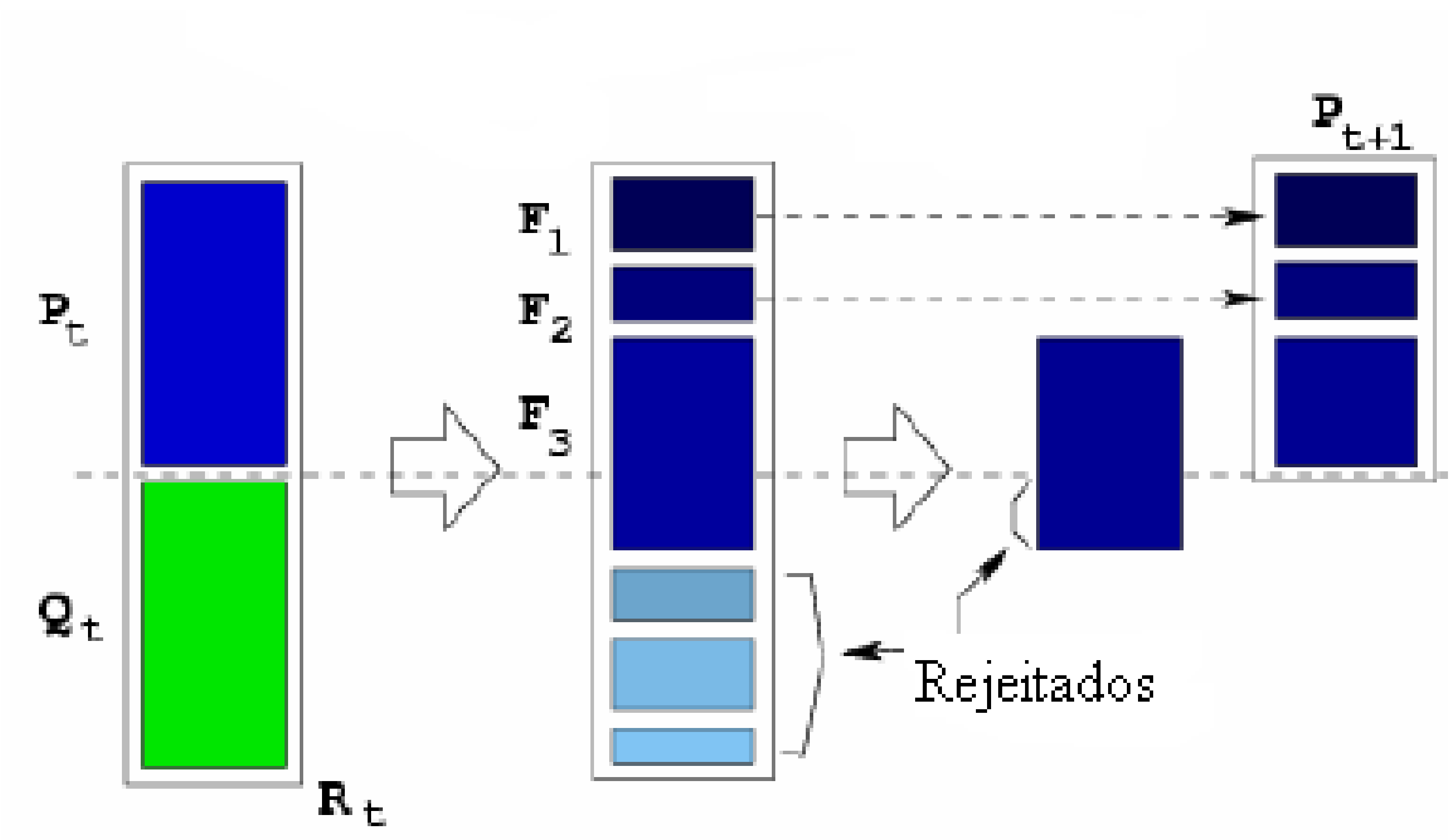
Otimização Evolucionária

- NSGA-II – Elitist Nondominated Sorting Genetic Algorithm
  - Possui três características:
    - Emprega um princípio elitista, com aptidão baseada em ordenamento (*ranking*);
    - Emprega um mecanismo explicito de preservação de diversidade, compartilhamento de aptidão (*fitness sharing*);
    - Enfatiza soluções não dominadas.

Otimização Evolucionária

- Combina conjuntos de pais (P_t) e filhos (Q_t)
  - Considera grupos de soluções com *ranks* melhores $F_1 > F_2 > F_3 > ...$
- Cria ranking
  - Baseado em dominância.
  - Ordena as soluções de acordo com este *rank*.
  - Rejeita soluções com *rank* mais baixo.
- Seleciona N melhores soluções
  - Soluções rejeitas com base em menor diversidade.

Otimização Evolucionária



Otimização Evolucionária

- Algoritmo do NSGA-II
  - Crie a população intermediária através da união dos indivíduos em uma geração com seus filhos: $R_t = P_t \cup Q_t$.
  - Ordene R_t e gere as frentes F_i, i=1, 2, ...
  - Faça $P_{t+1} = P_{t+1} + F_i$, enquanto $P_{t+1} < |P_t|$.
  - Complete a população com indivíduos da frente próxima frente, empregando *crowding distance* para ordenar indivíduos.
  - Crie próxima prole Q_{t+1} empregando seleção por *crowding tournament*, operadores de mutação e cruzamento.

Otimização Evolucionária

- Peculiaridade na seleção
  - Muitas soluções em um mesmo nível.
  - Soluções de nichos menores possuem prioridade.
  - Proporciona diversidade.

- Seleção – *Crowding Tournament*
  - Uma solução *i* ganha um torneio com outra solução *j* uma das condições for verdadeira:
    - $r_i < r_j$, melhor rank de *i* com respeito a j;
    - $r_i = r_j$, e $d_i > d_j$, melhor *crowding distance* de i.

Otimização Evolucionária

- VANTAGENS
  - Elitismo;
  - Diversidade;
  - Prova de convergência (seleção com torneio simples).

- DESVANTAGENS
  - Estimar parâmetro do compartilhamento de aptidão.
  - Ordenação na população de tamanho 2N.
  - Perda de convergência (*Crowding tournament*).

Otimização Evolucionária

- SPEA-2 – Elitist Nondominated Sorting Genetic Algorithm
  - Possui três características:
    - Emprega um princípio elitista, com aptidão baseada em ordenamento (*ranking*);
    - Emprega um mecanismo explicito de preservação de diversidade, compartilhamento de aptidão (*fitness sharing*);
    - Enfatiza soluções não dominadas.

- http://www.ccs.neu.edu/home/kunkle/papers/techreports/moeaComparison.pdf
- http://e-collection.library.ethz.ch/eserv/eth:24689/eth-24689-01.pdf
- http://www.cleveralgorithms.com/nature-inspired/evolution/spea.html
- https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0CFcQFjAGahUKEwjy_brpt4nJAhUJk5AKHcLhA6g&url=http%3A%2F%2Fwww.polymtl.ca%2Fnamp%2Fdocweb%2FModules_Web%2FM15_Part1_Tier1_MultiObjective.ppt&usg=AFQjCNHLHE8SejW1yjS30Bb2nR3TI7KDNQ
- https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&ved=0CEgQFjAEahUKEwiWz9XOtonJAhXIi5AKHd7xCWA&url=http%3A%2F%2Fwebspace.ulbsibiu.ro%2Fadrian.florea

Métricas

- Comparar algoritmos
  - Correção
    - Set coverage Metric
    - Generational distance
  - Diversidade
    - Spacing
    - Spread

Métricas - Corretude

- Set Coverage Metric:

$$C(P^*, Q) = \frac{|\{q \in Q \mid \exists p \in P^* : p \prec q\}|}{|Q|}$$

- Generational Distance:

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^{\,p}\right)^{\frac{1}{p}}}{|Q|} \quad d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{m=1}^{M} \left(f_m^{(i)} - f_m^{*(k)}\right)^2}$$

Métricas - Diversidade

- Spacing
  - Distribuição
  - Não considera extensão

$$S = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} \left(d_i - \bar{d}\right)^2}$$

$$d_i = \min_{k \in Q \wedge k \neq i} \sum_{m=1}^{M} | f_m^i - f_m^k | \quad \bar{d} = \sum_{i=1}^{|Q|} \frac{d_i}{|Q|}$$

Métricas - Diversidade

- Spread

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{|Q|} | d_i - \bar{d} |}{\sum_{m=1}^{M} d_m^e + | Q | \bar{d}}$$

- d_m^e = Distância entre extremos de Q e P*
- d_i = Menor distância entre soluções

Softwares & Frameworks

- JMetal
- PISA
- KEA
- GUIMOO

Softwares & Frameworks

- JMetal – Metaheuristic Algorithms in Java
  - Facilidade e reuso
  - NSGA-II, SPEA-II, PAES, AbYSS ...
  - ZT1-ZT6
  - Representação binária, real, permutação, real codificado em binário
  - Open e Free

Softwares & Frameworks

- PISA
  - Dois módulos como programas independentes:
    - Específico ao problema (representação)
    - Específico ao algoritmo (seleção, aptidão)
  - Poucos algoritmos
    - NSGA-II, SPEA2
  - Muitos problemas
    - ZT1-ZT6, caxeiro-viajante
  - C/C++

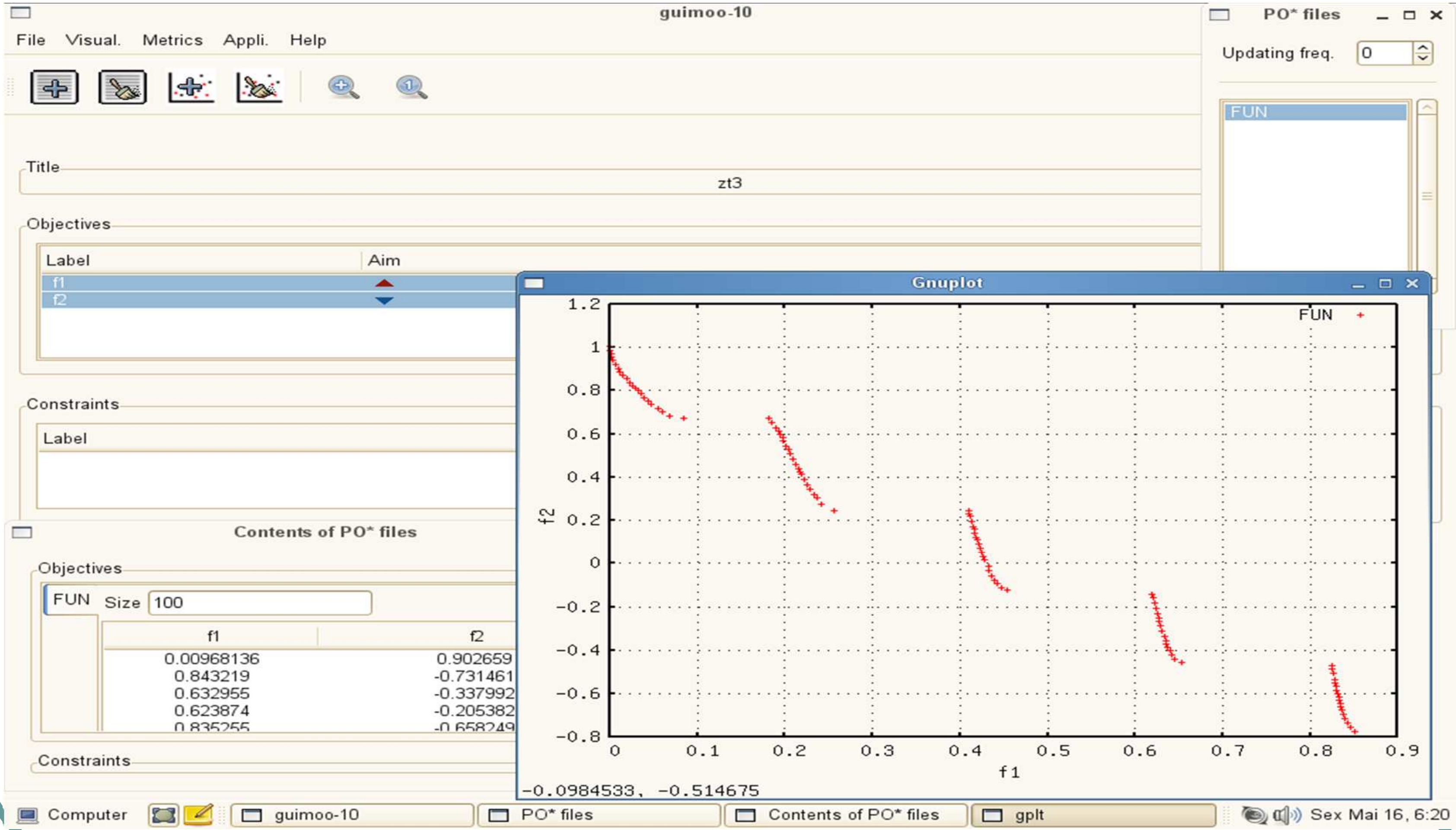
Softwares & Frameworks

- KEA – Kit for Evolutionary Algorithms
  - Java (interface em c++)
  - Interface gráfica (vários bugs)
  - http://ls11-www.cs.uni-dortmund.de/people/schmitt/Daten/Kea/kea.jsp
- Visualização em tempo real
  - GNUPlot
- Vários problemas e algoritmos implementados

Softwares & Frameworks

- Guimoo – Graphical User Interface for Multi-Objective Optimization
  - Análise de resultados
  - Métricas
    - R-metics
    - Entropia
    - Hypervolume

Softwares & Frameworks



Conclusões & Trabalhos Futuros

- AG como melhor técnica para solução de MOOP
  - Não-dominância
  - Diversidade (*fitness sharing*)
  - Elitismo
- Novas formas de elitismo em MOEA
- Novas funções *fitness sharing*
- Relação “Teoria dos jogos X Otimização com objetivos múltiplos”

Referências

- Livros
  - Evolutionary Algorithms in Solving Multi-Objective Problems – Coello Coello
  - Multi-Objective Optimization Using Evolutionary Algorithms – Kalyanmoy Deb
  - Introduction to Evolutionary Computing – A. E. Eiben, J. E. Smith
- Artigos
  - Genetic algorithms with sharing for multimodal function optimization – Goldberg
  - On finding the maxima of a set of vector – H. T. Kung, F. Luccio, F. P. Preparata