

# PCC175 - TÉCNICAS DE OTIMIZAÇÃO MULTI-OBJETIVO

AULAS 05 E 06 - ALGORITMOS EVOLUTIVOS MULTI-OBJETIVOS

---

Gladston Juliano Prates Moreira  
email: [gladston@ufop.edu.br](mailto:gladston@ufop.edu.br)

CSILab, Departamento de Computação  
Universidade Federal de Ouro Preto

3 de dezembro de 2024

# ALGORITMOS EVOLUTIVOS MULTI-OBJETIVOS

---

## Algoritmos Evolutivos Multi-objetivos

### Mono-objetivos vs. Multi-objetivos

Métodos Escalares vs. Algoritmos Evolutivos

*Técnicas baseadas em Dominância - Pareto*

*Nondominated Sorting Genetic Algorithm II: NSGA-II*

*Strength Pareto Evolutionary Algorithm II: SPEA2*

## **Algoritmos Evolutivos Mono-Objetivo:**

- ▶ um valor de função objetivo  $\rightarrow$  *fitness*;

## **Algoritmos Evolutivos Multiobjetivo (MOEA):**

- ▶ vetor de funções objetivo  $\rightarrow$  algum processamento adicional para tratar os vetores de *fitness*;

## **Algoritmos Evolutivos Mono-Objetivo:**

- ▶ um valor de função objetivo  $\rightarrow$  *fitness*;

## **Algoritmos Evolutivos Multiobjetivo (MOEA):**

- ▶ vetor de funções objetivo  $\rightarrow$  algum processamento adicional para tratar os vetores de *fitness*;

- ▶ Algoritmos Evolutivos Mono-Objetivo:
  - ▶ Manutenção de diversidade no espaço de variáveis → multimodalidade;
- ▶ Algoritmos Evolutivos Multiobjetivo (MOEA):
  - ▶ Manutenção de diversidade no espaço de variáveis:
    - ▶ Multimodalidade;
    - ▶ Melhor cobertura do conjunto das soluções eficientes;
    - ▶ Uniformidade;
  - ▶ Manutenção de diversidade no espaço de objetivos:
    - ▶ Melhor cobertura da fronteira de Pareto;
    - ▶ Convergência;
    - ▶ Uniformidade;

- ▶ Algoritmos Evolutivos Mono-Objetivo:
  - ▶ Manutenção de diversidade no espaço de variáveis → multimodalidade;
- ▶ Algoritmos Evolutivos Multiobjetivo (MOEA):
  - ▶ Manutenção de diversidade no espaço de variáveis:
    - ▶ Multimodalidade;
    - ▶ Melhor cobertura do conjunto das soluções eficientes;
    - ▶ Uniformidade;
  - ▶ Manutenção de diversidade no espaço de objetivos:
    - ▶ Melhor cobertura da fronteira de Pareto;
    - ▶ Convergência;
    - ▶ Uniformidade;

- ▶ Algoritmos Evolutivos Mono-Objetivo:
  - ▶ Manutenção de diversidade no espaço de variáveis → multimodalidade;
- ▶ Algoritmos Evolutivos Multiobjetivo (MOEA):
  - ▶ Manutenção de diversidade no espaço de variáveis:
    - ▶ Multimodalidade;
    - ▶ Melhor cobertura do conjunto das soluções eficientes;
    - ▶ Uniformidade;
  - ▶ Manutenção de diversidade no espaço de objetivos:
    - ▶ Melhor cobertura da fronteira de Pareto;
    - ▶ Convergência;
    - ▶ Uniformidade;

## Algoritmos Evolutivos Multi-objetivos

Mono-objetivos vs. Multi-objetivos

Métodos Escalares vs. Algoritmos Evolutivos

*Técnicas baseadas em Dominância - Pareto*

*Nondominated Sorting Genetic Algorithm II: NSGA-II*

*Strength Pareto Evolutionary Algorithm II: SPEA2*

## Métodos Escalares:

- ▶ Geralmente modificam o problema de otimização multiobjetivo e aplicam ferramentas tradicionais de otimização;
- ▶ Geralmente encontram uma única solução a cada execução: precisam ser executados  $N$  vezes, com conjuntos distintos de parâmetros, para obterem  $N$  soluções diferentes;
- ▶ Nem sempre conseguem uma cobertura uniforme da fronteira;
- ▶ Nem todas as possíveis soluções podem ser encontradas por alguns métodos (ex.: regiões não-convexas);
- ▶ Geralmente requerem algum conhecimento prévio sobre o problema (ex.: definir adequadamente os valores de  $\epsilon$  para o método de  $\epsilon$ -restrito)

## Métodos Escalares:

- ▶ Geralmente modificam o problema de otimização multiobjetivo e aplicam ferramentas tradicionais de otimização;
- ▶ Geralmente encontram uma única solução a cada execução: precisam ser executados  $N$  vezes, com conjuntos distintos de parâmetros, para obterem  $N$  soluções diferentes;
- ▶ Nem sempre conseguem uma cobertura uniforme da fronteira;
- ▶ Nem todas as possíveis soluções podem ser encontradas por alguns métodos (ex.: regiões não-convexas);
- ▶ Geralmente requerem algum conhecimento prévio sobre o problema (ex.: definir adequadamente os valores de  $\epsilon$  para o método de  $\epsilon$ -restrito)

## Métodos Escalares:

- ▶ Geralmente modificam o problema de otimização multiobjetivo e aplicam ferramentas tradicionais de otimização;
- ▶ Geralmente encontram uma única solução a cada execução: precisam ser executados  $N$  vezes, com conjuntos distintos de parâmetros, para obterem  $N$  soluções diferentes;
- ▶ Nem sempre conseguem uma cobertura uniforme da fronteira;
- ▶ Nem todas as possíveis soluções podem ser encontradas por alguns métodos (ex.: regiões não-convexas);
- ▶ Geralmente requerem algum conhecimento prévio sobre o problema (ex.: definir adequadamente os valores de  $\epsilon$  para o método de  $\epsilon$ -restrito)

## Métodos Escalares:

- ▶ Geralmente modificam o problema de otimização multiobjetivo e aplicam ferramentas tradicionais de otimização;
- ▶ Geralmente encontram uma única solução a cada execução: precisam ser executados  $N$  vezes, com conjuntos distintos de parâmetros, para obterem  $N$  soluções diferentes;
- ▶ Nem sempre conseguem uma cobertura uniforme da fronteira;
- ▶ Nem todas as possíveis soluções podem ser encontradas por alguns métodos (ex.: regiões não-convexas);
- ▶ Geralmente requerem algum conhecimento prévio sobre o problema (ex.: definir adequadamente os valores de  $\epsilon$  para o método de  $\epsilon$ -restrito)

## Métodos Escalares:

- ▶ Geralmente modificam o problema de otimização multiobjetivo e aplicam ferramentas tradicionais de otimização;
- ▶ Geralmente encontram uma única solução a cada execução: precisam ser executados  $N$  vezes, com conjuntos distintos de parâmetros, para obterem  $N$  soluções diferentes;
- ▶ Nem sempre conseguem uma cobertura uniforme da fronteira;
- ▶ Nem todas as possíveis soluções podem ser encontradas por alguns métodos (ex.: regiões não-convexas);
- ▶ Geralmente requerem algum conhecimento prévio sobre o problema (ex.: definir adequadamente os valores de  $\epsilon$  para o método de  $\epsilon$ -restrito)

## **Algoritmos Evolutivos:**

- ▶ São mais flexíveis;
- ▶ Largamente utilizados para tratar problemas de otimização multiobjetivo.
- ▶ Algoritmos populacionais → trabalham simultaneamente com múltiplas soluções;
- ▶ As múltiplas soluções podem ser obtidas em uma única execução;
- ▶ Tendem a ser menos susceptíveis à forma da Fronteira de Pareto e a eventuais discontinuidades.

## Algoritmos Evolutivos:

- ▶ São mais flexíveis;
- ▶ Largamente utilizados para tratar problemas de otimização multiobjetivo.
- ▶ Algoritmos populacionais → trabalham simultaneamente com múltiplas soluções;
- ▶ As múltiplas soluções podem ser obtidas em uma única execução;
- ▶ Tendem a ser menos susceptíveis à forma da Fronteira de Pareto e a eventuais discontinuidades.

## Algoritmos Evolutivos:

- ▶ São mais flexíveis;
- ▶ Largamente utilizados para tratar problemas de otimização multiobjetivo.
- ▶ Algoritmos populacionais → trabalham simultaneamente com múltiplas soluções;
- ▶ As múltiplas soluções podem ser obtidas em uma única execução;
- ▶ Tendem a ser menos susceptíveis à forma da Fronteira de Pareto e a eventuais discontinuidades.

## Algoritmos Evolutivos:

- ▶ São mais flexíveis;
- ▶ Largamente utilizados para tratar problemas de otimização multiobjetivo.
- ▶ Algoritmos populacionais → trabalham simultaneamente com múltiplas soluções;
- ▶ As múltiplas soluções podem ser obtidas em uma única execução;
- ▶ Tendem a ser menos susceptíveis à forma da Fronteira de Pareto e a eventuais discontinuidades.

## Algoritmos Evolutivos:

- ▶ São mais flexíveis;
- ▶ Largamente utilizados para tratar problemas de otimização multiobjetivo.
- ▶ Algoritmos populacionais → trabalham simultaneamente com múltiplas soluções;
- ▶ As múltiplas soluções podem ser obtidas em uma única execução;
- ▶ Tendem a ser menos susceptíveis à forma da Fronteira de Pareto e a eventuais descontinuidades.

## Algoritmos Evolutivos Multi-objetivos

Mono-objetivos vs. Multi-objetivos

Métodos Escalares vs. Algoritmos Evolutivos

*Técnicas baseadas em Dominância - Pareto*

*Nondominated Sorting Genetic Algorithm II: NSGA-II*

*Strength Pareto Evolutionary Algorithm II: SPEA2*

Relação de dominância:

- ▶ A relação de dominação impõe uma ordem parcial estrita;
- ▶ Em relação à geração e seleção do conjunto Pareto-ótimo, uma técnica de ordenação é necessária no algoritmo evolutivo - métodos de classificação e ordenação;
- ▶ Requer um processo de classificação e uma técnica para manter a diversidade da população.

Tipos de informação:

- ▶ hierarquia de dominância (*dominance rank*): Por quantos indivíduos o indivíduo é dominado?
- ▶ contagem de dominância (*dominance count*): Quantos indivíduos um indivíduo domina?
- ▶ profundidade dominância (*dominance depth*): Qual fronteira o indivíduo está localizado?

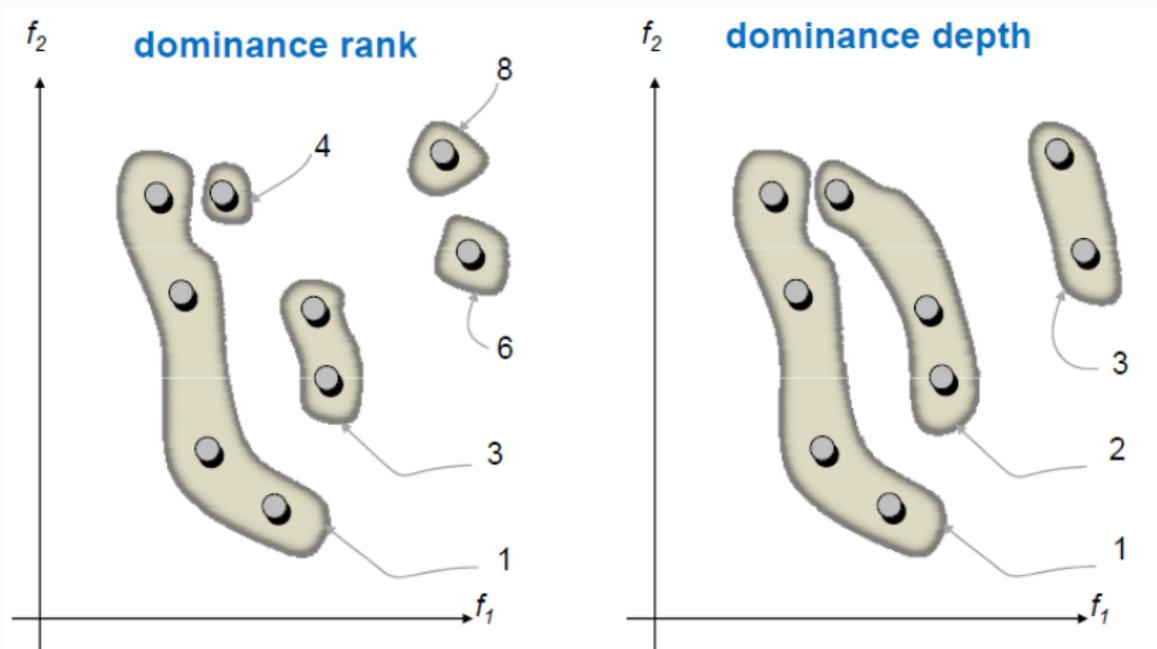
Tipos de informação:

- ▶ hierarquia de dominância (*dominance rank*): Por quantos indivíduos o indivíduo é dominado?
- ▶ contagem de dominância (*dominance count*): Quantos indivíduos um indivíduo domina?
- ▶ profundidade dominância (*dominance depth*): Qual fronteira o indivíduo está localizado?

Tipos de informação:

- ▶ hierarquia de dominância (*dominance rank*): Por quantos indivíduos o indivíduo é dominado?
- ▶ contagem de dominância (*dominance count*): Quantos indivíduos um indivíduo domina?
- ▶ profundidade dominância (*dominance depth*): Qual fronteira o indivíduo está localizado?

# Técnicas baseadas em Dominância - Pareto



Tipos de informação:

- ▶ **dominance rank**: MOGA, NPGA, MOMGA, MOMGAI
- ▶ **dominance count and rank**: SPEA, SPEA2
- ▶ **dominance depth**: NSGA, NSGA-II

Tipos de informação:

- ▶ **dominance rank**: MOGA, NPGA, MOMGA, MOMGAI
- ▶ **dominance count and rank**: SPEA, SPEA2
- ▶ **dominance depth**: NSGA, NSGA-II

Tipos de informação:

- ▶ **dominance rank**: MOGA, NPGA, MOMGA, MOMGAI
- ▶ **dominance count and rank**: SPEA, SPEA2
- ▶ **dominance depth**: NSGA, NSGA-II

Refinamento: aproximação do conjunto Pareto-ótimo envolve dois objetivos.

- ▶ Minimização à distância da fronteira ótima;
- ▶ Maximização da diversidade das soluções geradas.

Preservação da diversidade pode ser incorporado em MOEAs, para ter uma distribuição uniforme em toda fronteira Pareto estimada.

Refinamento: aproximação do conjunto Pareto-ótimo envolve dois objetivos.

- ▶ Minimização à distância da fronteira ótima;
- ▶ Maximização da diversidade das soluções geradas.

Preservação da diversidade pode ser incorporado em MOEAs, para ter uma distribuição uniforme em toda fronteira Pareto estimada.

Refinamento: aproximação do conjunto Pareto-ótimo envolve dois objetivos.

- ▶ Minimização à distância da fronteira ótima;
- ▶ Maximização da diversidade das soluções geradas.

Preservação da diversidade pode ser incorporado em MOEAs, para ter uma distribuição uniforme em toda fronteira Pareto estimada.

## Algoritmos Evolutivos Multi-objetivos

Mono-objetivos vs. Multi-objetivos

Métodos Escalares vs. Algoritmos Evolutivos

*Técnicas baseadas em Dominância - Pareto*

***Nondominated Sorting Genetic Algorithm II: NSGA-II***

*Strength Pareto Evolutionary Algorithm II: SPEA2*

- ▶ Proposto em: K. Deb, A. Pratap, S. Agarwal & T. Meyarivan  
“A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”.  
IEEE Trans. Evolutionary Computation, v. 6(2), 2002.
- ▶ Algoritmo Genético;
- ▶ Elitista;
- ▶ Utiliza ordenação dos indivíduos por critério de não-dominância;
- ▶ Manutenção de diversidade: *crowding distance* (espaço de objetivos).

- ▶ Proposto em: K. Deb, A. Pratap, S. Agarwal & T. Meyarivan "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". IEEE Trans. Evolutionary Computation, v. 6(2), 2002.
- ▶ Algoritmo Genético;
- ▶ Elitista;
- ▶ Utiliza ordenação dos indivíduos por critério de não-dominância;
- ▶ Manutenção de diversidade: *crowding distance* (espaço de objetivos).

- ▶ Proposto em: K. Deb, A. Pratap, S. Agarwal & T. Meyarivan "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". IEEE Trans. Evolutionary Computation, v. 6(2), 2002.
- ▶ Algoritmo Genético;
- ▶ Elitista;
- ▶ Utiliza ordenação dos indivíduos por critério de não-dominância;
- ▶ Manutenção de diversidade: *crowding distance* (espaço de objetivos).

- ▶ Proposto em: K. Deb, A. Pratap, S. Agarwal & T. Meyarivan "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". IEEE Trans. Evolutionary Computation, v. 6(2), 2002.
- ▶ Algoritmo Genético;
- ▶ Elitista;
- ▶ Utiliza ordenação dos indivíduos por critério de não-dominância;
- ▶ Manutenção de diversidade: *crowding distance* (espaço de objetivos).

- ▶ Proposto em: K. Deb, A. Pratap, S. Agarwal & T. Meyarivan "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". IEEE Trans. Evolutionary Computation, v. 6(2), 2002.
- ▶ Algoritmo Genético;
- ▶ Elitista;
- ▶ Utiliza ordenação dos indivíduos por critério de não-dominância;
- ▶ Manutenção de diversidade: *crowding distance* (espaço de objetivos).

- ▶ Atribuir uma nota a um indivíduo  $\rightarrow$  “*fitness*”;
- ▶ Divide os indivíduos em classes, de acordo com a não dominância:
  - ▶ Indivíduos não dominados  $\rightarrow$  classe 1;
  - ▶ Indivíduos dominados apenas pelos indivíduos da classe 1  $\rightarrow$  classe 2;
  - ▶ Indivíduos dominados apenas pelos indivíduos das classes 1 e 2  $\rightarrow$  classe 3;
  - ▶ ...;

- ▶ Atribuir uma nota a um indivíduo  $\rightarrow$  “*fitness*”;
- ▶ Divide os indivíduos em classes, de acordo com a não dominância:
  - ▶ Indivíduos não dominados  $\rightarrow$  classe 1;
  - ▶ Indivíduos dominados apenas pelos indivíduos da classe 1  $\rightarrow$  classe 2;
  - ▶ Indivíduos dominados apenas pelos indivíduos das classes 1 e 2  $\rightarrow$  classe 3;
  - ▶ ...;

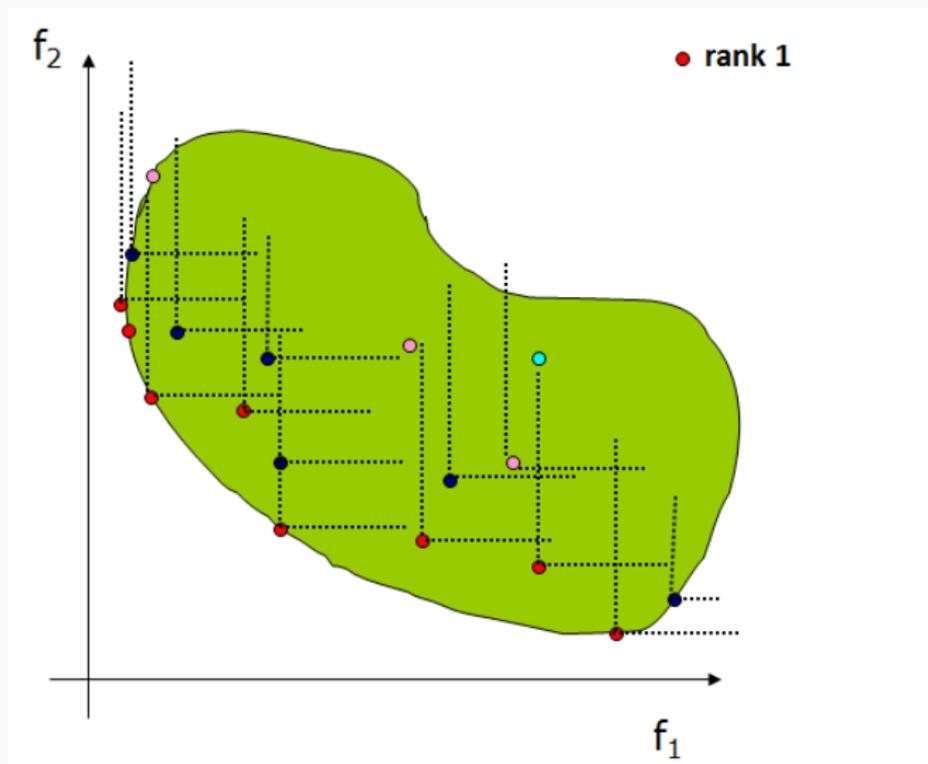
- ▶ Atribuir uma nota a um indivíduo  $\rightarrow$  “*fitness*”;
- ▶ Divide os indivíduos em classes, de acordo com a não dominância:
  - ▶ Indivíduos não dominados  $\rightarrow$  classe 1;
  - ▶ Indivíduos dominados apenas pelos indivíduos da classe 1  $\rightarrow$  classe 2;
  - ▶ Indivíduos dominados apenas pelos indivíduos das classes 1 e 2  $\rightarrow$  classe 3;
  - ▶ ...;

- ▶ Atribuir uma nota a um indivíduo  $\rightarrow$  “*fitness*”;
- ▶ Divide os indivíduos em classes, de acordo com a não dominância:
  - ▶ Indivíduos não dominados  $\rightarrow$  classe 1;
  - ▶ Indivíduos dominados apenas pelos indivíduos da classe 1  $\rightarrow$  classe 2;
  - ▶ Indivíduos dominados apenas pelos indivíduos das classes 1 e 2  $\rightarrow$  classe 3;
  - ▶ ...;

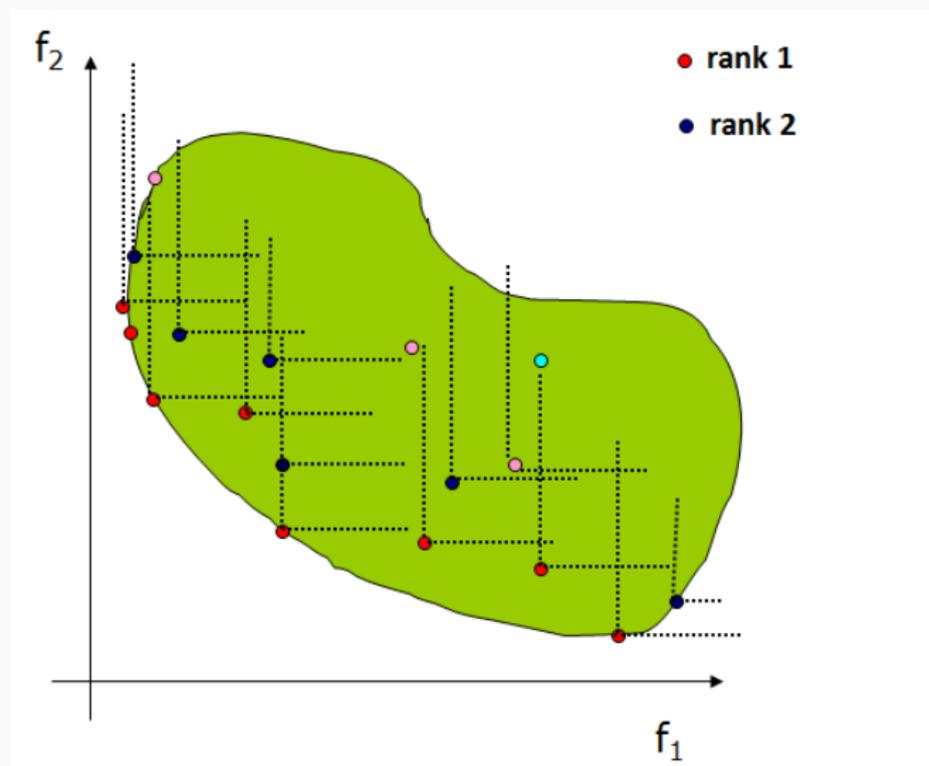
- ▶ Atribuir uma nota a um indivíduo  $\rightarrow$  “*fitness*”;
- ▶ Divide os indivíduos em classes, de acordo com a não dominância:
  - ▶ Indivíduos não dominados  $\rightarrow$  classe 1;
  - ▶ Indivíduos dominados apenas pelos indivíduos da classe 1  $\rightarrow$  classe 2;
  - ▶ Indivíduos dominados apenas pelos indivíduos das classes 1 e 2  $\rightarrow$  classe 3;
  - ▶ ...;

- ▶ Atribuir uma nota a um indivíduo  $\rightarrow$  “*fitness*”;
- ▶ Divide os indivíduos em classes, de acordo com a não dominância:
  - ▶ Indivíduos não dominados  $\rightarrow$  classe 1;
  - ▶ Indivíduos dominados apenas pelos indivíduos da classe 1  $\rightarrow$  classe 2;
  - ▶ Indivíduos dominados apenas pelos indivíduos das classes 1 e 2  $\rightarrow$  classe 3;
  - ▶ ...;

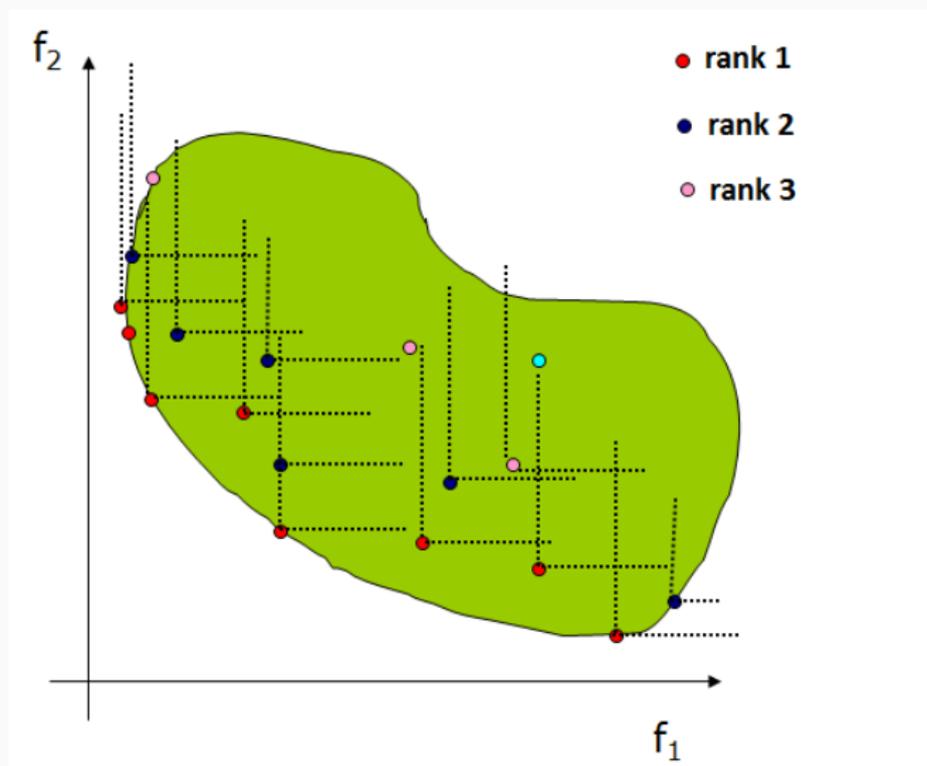
# NSGA-II: Ordenação por Não-Dominância



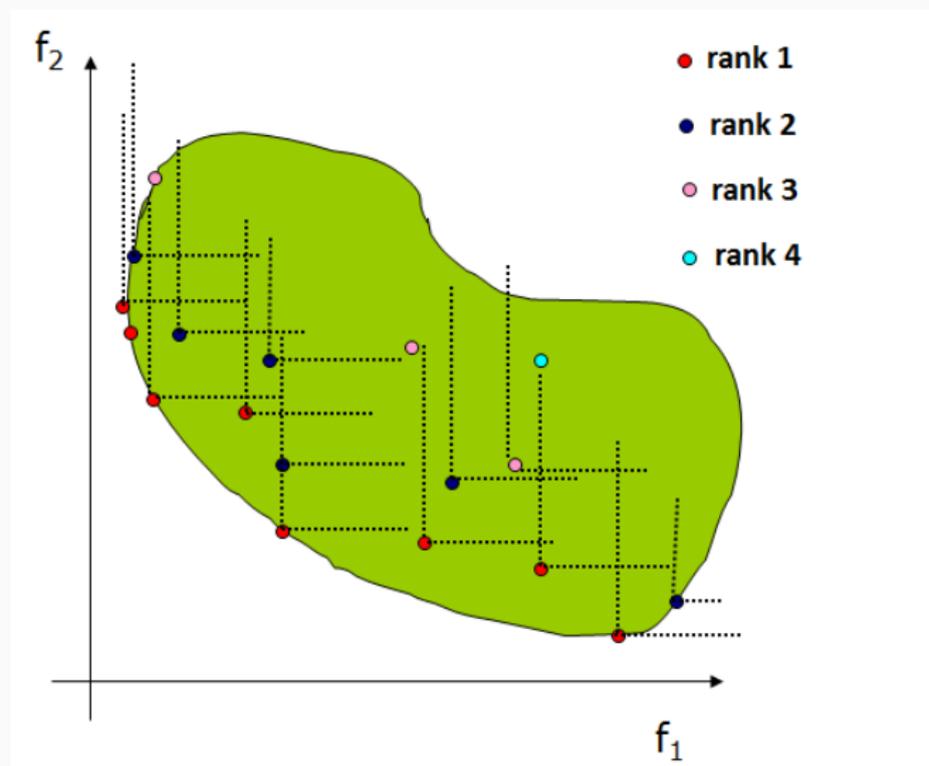
# NSGA-II: Ordenação por Não-Dominância



# NSGA-II: Ordenação por Não-Dominância



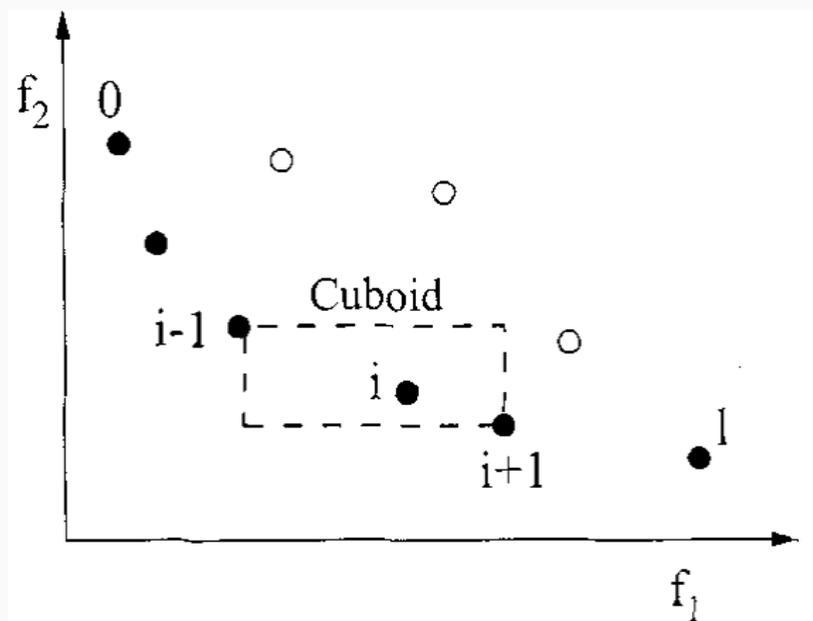
# NSGA-II: Ordenação por Não-Dominância



- ▶ A *crowding-distance* é usado como uma medida de ocupação na vizinhança de uma solução no espaço dos objetivos;
- ▶ Uma solução com elevado valor de *crowding-distance*, significa que os seus vizinhos estão longe e, portanto, sua vizinhança é pouco povoada.

- ▶ A *crowding-distance* é usado como uma medida de ocupação na vizinhança de uma solução no espaço dos objetivos;
- ▶ Uma solução com elevado valor de *crowding-distance*, significa que os seus vizinhos estão longe e, portanto, sua vizinhança é pouco povoada.

# NSGA-II: Diversidade por Crowding-Distance



- For each front  $F_i$ ,  $n$  is the number of individuals.
  - initialize the distance to be zero for all the individuals i.e.  $F_i(d_j) = 0$ , where  $j$  corresponds to the  $j^{th}$  individual in front  $F_i$ .
  - for each objective function  $m$ 
    - \* Sort the individuals in front  $F_i$  based on objective  $m$  i.e.  $I = \text{sort}(F_i, m)$ .
    - \* Assign infinite distance to boundary values for each individual in  $F_i$  i.e.  $I(d_1) = \infty$  and  $I(d_n) = \infty$
    - \* for  $k = 2$  to  $(n - 1)$ 
      - $I(d_k) = I(d_k) + \frac{I(k + 1).m - I(k - 1).m}{f_m^{max} - f_m^{min}}$
      - $I(k).m$  is the value of the  $m^{th}$  objective function of the  $k^{th}$  individual in  $I$

- ▶ Versão modificada do torneio binário:
  - ▶ Escolhidas duas soluções, escolhe-se a que possui melhor *rank*. Caso as duas tenham o mesmo *rank*, escolhe-se a solução com melhor (maior) valor de *crowding-distance*.
- ▶ Cruzamento e mutação: dependem do problema, não do algoritmo.

- ▶ Versão modificada do torneio binário:
  - ▶ Escolhidas duas soluções, escolhe-se a que possui melhor *rank*. Caso as duas tenham o mesmo *rank*, escolhe-se a solução com melhor (maior) valor de *crowding-distance*.
- ▶ Cruzamento e mutação: dependem do problema, não do algoritmo.

- ▶ Versão modificada do torneio binário:
  - ▶ Escolhidas duas soluções, escolhe-se a que possui melhor *rank*. Caso as duas tenham o mesmo *rank*, escolhe-se a solução com melhor (maior) valor de *crowding-distance*.
- ▶ Cruzamento e mutação: dependem do problema, não do algoritmo.

- ▶ Problemas com restrições são tratados com uma simples modificação na definição de dominância:
- ▶ Dadas duas soluções  $u$  e  $v$ , diz-se que  $u$  domina  $v$  com restrições se:
  - ▶  $u$  é factível e  $v$  não é factível;
  - ▶ Ambas as soluções são infactíveis, mas  $u$  viola menos as restrições que  $v$ ;
  - ▶ Ambas as soluções são factíveis e  $u$  domina  $v$ ;

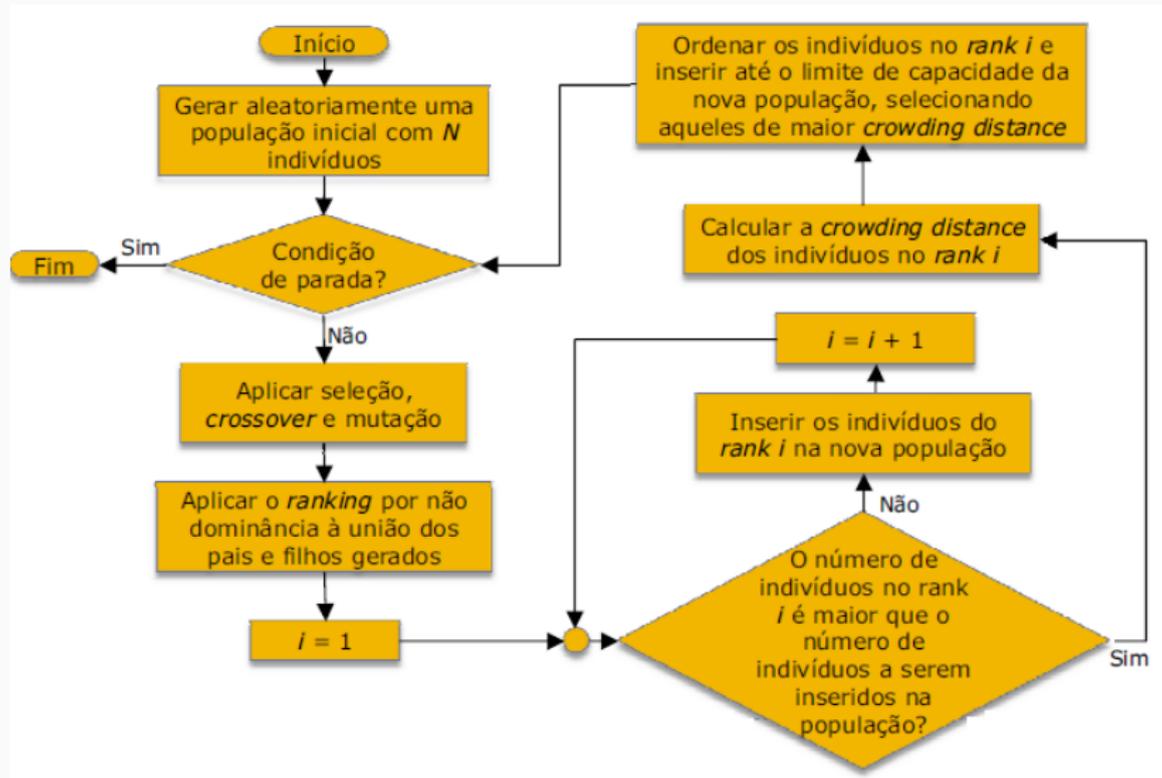
- ▶ Problemas com restrições são tratados com uma simples modificação na definição de dominância:
- ▶ Dadas duas soluções  $u$  e  $v$ , diz-se que  $u$  domina  $v$  com restrições se:
  - ▶  $u$  é factível e  $v$  não é factível;
  - ▶ Ambas as soluções são infactíveis, mas  $u$  viola menos as restrições que  $v$ ;
  - ▶ Ambas as soluções são factíveis e  $u$  domina  $v$ ;

- ▶ Problemas com restrições são tratados com uma simples modificação na definição de dominância:
- ▶ Dadas duas soluções  $u$  e  $v$ , diz-se que  $u$  domina  $v$  com restrições se:
  - ▶  $u$  é factível e  $v$  não é factível;
  - ▶ Ambas as soluções são infactíveis, mas  $u$  viola menos as restrições que  $v$ ;
  - ▶ Ambas as soluções são factíveis e  $u$  domina  $v$ ;

- ▶ Problemas com restrições são tratados com uma simples modificação na definição de dominância:
- ▶ Dadas duas soluções  $u$  e  $v$ , diz-se que  $u$  domina  $v$  com restrições se:
  - ▶  $u$  é factível e  $v$  não é factível;
  - ▶ Ambas as soluções são infactíveis, mas  $u$  viola menos as restrições que  $v$ ;
  - ▶ Ambas as soluções são factíveis e  $u$  domina  $v$ ;

- ▶ Problemas com restrições são tratados com uma simples modificação na definição de dominância:
- ▶ Dadas duas soluções  $u$  e  $v$ , diz-se que  $u$  domina  $v$  com restrições se:
  - ▶  $u$  é factível e  $v$  não é factível;
  - ▶ Ambas as soluções são infactíveis, mas  $u$  viola menos as restrições que  $v$ ;
  - ▶ Ambas as soluções são factíveis e  $u$  domina  $v$ ;

# NSGA-II: Fluxograma



## Algoritmos Evolutivos Multi-objetivos

Mono-objetivos vs. Multi-objetivos

Métodos Escalares vs. Algoritmos Evolutivos

*Técnicas baseadas em Dominância - Pareto*

*Nondominated Sorting Genetic Algorithm II: NSGA-II*

*Strength Pareto Evolutionary Algorithm II: SPEA2*

Principais diferenças em relação ao SPEA:

- ▶ Mecanismo de atribuição da *fitness*.
- ▶ Estimação da densidade das soluções.
- ▶ Regra de truncagem do arquivo.

- ▶ Proposto em: E. Zitzler, M. Laumanns & L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm”. Technical Report, ETH Zurich, 2001.
- ▶ Genético;
- ▶ Utiliza mecanismo de atribuição de *fitness* aos indivíduos;
- ▶ Mantém duas populações → indivíduos na busca e arquivo de soluções não-dominadas, ambas de tamanho fixo;
- ▶ Direciona a busca baseando-se na densidade de soluções na vizinhança de cada indivíduo (espaço de objetivos).

- ▶ Proposto em: E. Zitzler, M. Laumanns & L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm”. Technical Report, ETH Zurich, 2001.
- ▶ Genético;
- ▶ Utiliza mecanismo de atribuição de *fitness* aos indivíduos;
- ▶ Mantém duas populações → indivíduos na busca e arquivo de soluções não-dominadas, ambas de tamanho fixo;
- ▶ Direciona a busca baseando-se na densidade de soluções na vizinhança de cada indivíduo (espaço de objetivos).

- ▶ Proposto em: E. Zitzler, M. Laumanns & L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm”. Technical Report, ETH Zurich, 2001.
- ▶ Genético;
- ▶ Utiliza mecanismo de atribuição de *fitness* aos indivíduos;
- ▶ Mantém duas populações → indivíduos na busca e arquivo de soluções não-dominadas, ambas de tamanho fixo;
- ▶ Direciona a busca baseando-se na densidade de soluções na vizinhança de cada indivíduo (espaço de objetivos).

- ▶ Proposto em: E. Zitzler, M. Laumanns & L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm”. Technical Report, ETH Zurich, 2001.
- ▶ Genético;
- ▶ Utiliza mecanismo de atribuição de *fitness* aos indivíduos;
- ▶ Mantém duas populações → indivíduos na busca e arquivo de soluções não-dominadas, ambas de tamanho fixo;
- ▶ Direciona a busca baseando-se na densidade de soluções na vizinhança de cada indivíduo (espaço de objetivos).

- ▶ Proposto em: E. Zitzler, M. Laumanns & L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm”. Technical Report, ETH Zurich, 2001.
- ▶ Genético;
- ▶ Utiliza mecanismo de atribuição de *fitness* aos indivíduos;
- ▶ Mantém duas populações → indivíduos na busca e arquivo de soluções não-dominadas, ambas de tamanho fixo;
- ▶ Direciona a busca baseando-se na densidade de soluções na vizinhança de cada indivíduo (espaço de objetivos).

- ▶ Força  $S(i)$  do indivíduo  $i$ : número de indivíduos que  $i$  domina;
- ▶ *Fitness* Bruto  $R(i)$  do indivíduo  $i$ : soma das forças de todos os indivíduos que dominam  $i$  (quanto maior, pior o indivíduo);
- ▶ Densidade  $D(i)$  do indivíduo  $i$ : inverso da distância ao  $k$ -ésimo vizinho mais próximo ( $\sigma_i^k$ ) no espaço de objetivos (leva à exploração de regiões pouco-povoadas):

$$D(i) = \frac{1}{\sigma_i^k + 1}$$

$$F_{SPEA2}(i) = R(i) + D(i)$$

- ▶ Força  $S(i)$  do indivíduo  $i$ : número de indivíduos que  $i$  domina;
- ▶ *Fitness* Bruto  $R(i)$  do indivíduo  $i$ : soma das forças de todos os indivíduos que dominam  $i$  (quanto maior, pior o indivíduo);
- ▶ Densidade  $D(i)$  do indivíduo  $i$ : inverso da distância ao  $k$ -ésimo vizinho mais próximo ( $\sigma_i^k$ ) no espaço de objetivos (leva à exploração de regiões pouco-povoadas):

$$D(i) = \frac{1}{\sigma_i^k + 1}$$

$$F_{SPEA2}(i) = R(i) + D(i)$$

- ▶ Força  $S(i)$  do indivíduo  $i$ : número de indivíduos que  $i$  domina;
- ▶ *Fitness* Bruto  $R(i)$  do indivíduo  $i$ : soma das forças de todos os indivíduos que dominam  $i$  (quanto maior, pior o indivíduo);
- ▶ Densidade  $D(i)$  do indivíduo  $i$ : inverso da distância ao  $k$ -ésimo vizinho mais próximo ( $\sigma_i^k$ ) no espaço de objetivos (leva à exploração de regiões pouco-povoadas):

$$D(i) = \frac{1}{\sigma_i^k + 1}$$

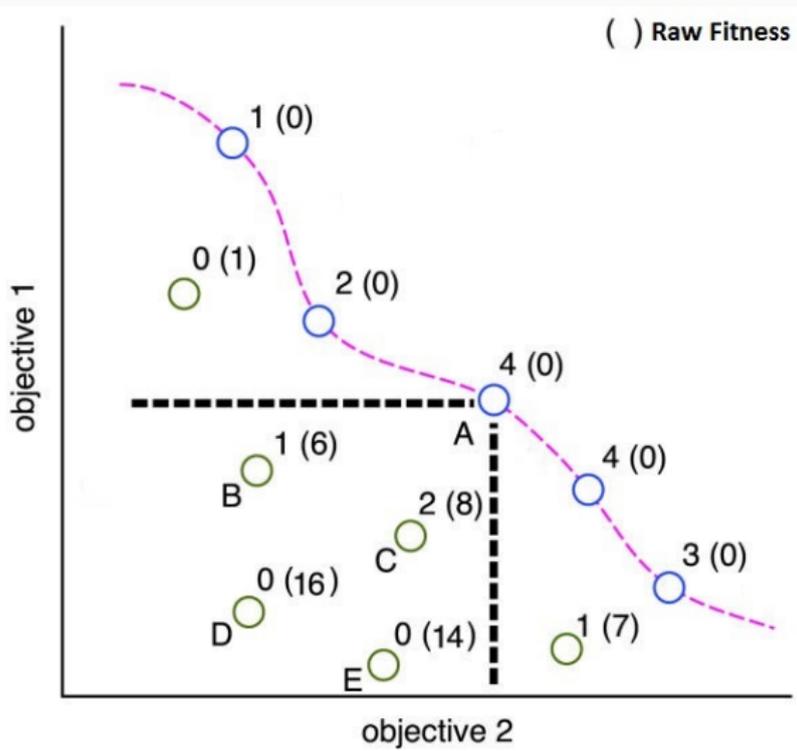
$$F_{SPEA2}(i) = R(i) + D(i)$$

- ▶ Força  $S(i)$  do indivíduo  $i$ : número de indivíduos que  $i$  domina;
- ▶ *Fitness* Bruto  $R(i)$  do indivíduo  $i$ : soma das forças de todos os indivíduos que dominam  $i$  (quanto maior, pior o indivíduo);
- ▶ Densidade  $D(i)$  do indivíduo  $i$ : inverso da distância ao  $k$ -ésimo vizinho mais próximo ( $\sigma_i^k$ ) no espaço de objetivos (leva à exploração de regiões pouco-povoadas):

$$D(i) = \frac{1}{\sigma_i^k + 1}$$

$$F_{SPEA2}(i) = R(i) + D(i)$$

# SPEA2: Cálculo da *Fitness*



$$F_{SPEA2}(i) = R(i) + D(i)$$

Diversidade no Espaço de Objetivos

Proximidade da Fronteira de Pareto

Trata as duas exigências em uma única métrica, que deve ser minimizada.

$$F_{SPEA2}(i) = R(i) + D(i)$$

Diversidade no Espaço de Objetivos

Proximidade da Fronteira de Pareto

Trata as duas exigências em uma única métrica, que deve ser minimizada.

# SPEA2: Atualização do Arquivo

- ▶ Deve manter o arquivo sempre com o mesmo número de soluções  $N_A$ , definido pelo usuário;
- ▶ Busca manter uma melhor cobertura da fronteira de Pareto, evitando a eliminação de soluções dos extremos da fronteira;
- ▶ É acionado na etapa de inserção das soluções não-dominadas da população no arquivo.
  
- ▶ Se o  $|\text{arquivo}|$  é inferior a  $N_A$ :
  - ▶ Completa o arquivo com os  $N_A - |\text{arquivo}|$  melhores indivíduos dominados (*fitness*);
- ▶ Se o  $|\text{arquivo}|$  é superior a  $N_A$ :
  - ▶ Elimina os indivíduos excedentes através de um processo iterativo, que exclui os indivíduos de menor distância aos seus  $k$  vizinhos mais próximos.

# SPEA2: Atualização do Arquivo

- ▶ Deve manter o arquivo sempre com o mesmo número de soluções  $N_A$ , definido pelo usuário;
- ▶ Busca manter uma melhor cobertura da fronteira de Pareto, evitando a eliminação de soluções dos extremos da fronteira;
- ▶ É acionado na etapa de inserção das soluções não-dominadas da população no arquivo.
  
- ▶ Se o  $|\text{arquivo}|$  é inferior a  $N_A$ :
  - ▶ Completa o arquivo com os  $N_A - |\text{arquivo}|$  melhores indivíduos dominados (*fitness*);
- ▶ Se o  $|\text{arquivo}|$  é superior a  $N_A$ :
  - ▶ Elimina os indivíduos excedentes através de um processo iterativo, que exclui os indivíduos de menor distância aos seus  $k$  vizinhos mais próximos.

# SPEA2: Atualização do Arquivo

- ▶ Deve manter o arquivo sempre com o mesmo número de soluções  $N_A$ , definido pelo usuário;
  - ▶ Busca manter uma melhor cobertura da fronteira de Pareto, evitando a eliminação de soluções dos extremos da fronteira;
  - ▶ É acionado na etapa de inserção das soluções não-dominadas da população no arquivo.
- 
- ▶ Se o  $|\text{arquivo}|$  é inferior a  $N_A$ :
    - ▶ Completa o arquivo com os  $N_A - |\text{arquivo}|$  melhores indivíduos dominados (*fitness*);
  - ▶ Se o  $|\text{arquivo}|$  é superior a  $N_A$ :
    - ▶ Elimina os indivíduos excedentes através de um processo iterativo, que exclui os indivíduos de menor distância aos seus  $k$  vizinhos mais próximos.

# SPEA2: Atualização do Arquivo

- ▶ Deve manter o arquivo sempre com o mesmo número de soluções  $N_A$ , definido pelo usuário;
- ▶ Busca manter uma melhor cobertura da fronteira de Pareto, evitando a eliminação de soluções dos extremos da fronteira;
- ▶ É acionado na etapa de inserção das soluções não-dominadas da população no arquivo.
  
- ▶ Se o  $|\text{arquivo}|$  é inferior a  $N_A$ :
  - ▶ Completa o arquivo com os  $N_A - |\text{arquivo}|$  melhores indivíduos dominados (*fitness*);
- ▶ Se o  $|\text{arquivo}|$  é superior a  $N_A$ :
  - ▶ Elimina os indivíduos excedentes através de um processo iterativo, que exclui os indivíduos de menor distância aos seus  $k$  vizinhos mais próximos.

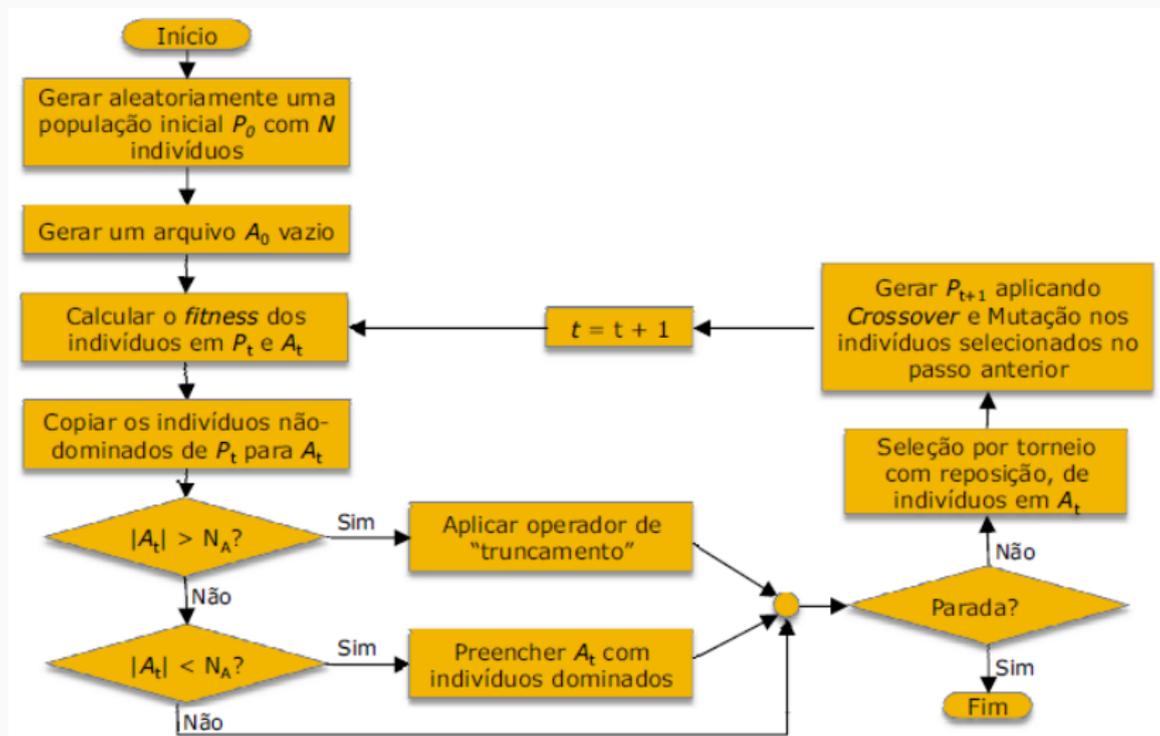
# SPEA2: Atualização do Arquivo

- ▶ Deve manter o arquivo sempre com o mesmo número de soluções  $N_A$ , definido pelo usuário;
- ▶ Busca manter uma melhor cobertura da fronteira de Pareto, evitando a eliminação de soluções dos extremos da fronteira;
- ▶ É acionado na etapa de inserção das soluções não-dominadas da população no arquivo.
  
- ▶ Se o  $|\text{arquivo}|$  é inferior a  $N_A$ :
  - ▶ Completa o arquivo com os  $N_A - |\text{arquivo}|$  melhores indivíduos dominados (*fitness*);
- ▶ Se o  $|\text{arquivo}|$  é superior a  $N_A$ :
  - ▶ Elimina os indivíduos excedentes através de um processo iterativo, que exclui os indivíduos de menor distância aos seus  $k$  vizinhos mais próximos.

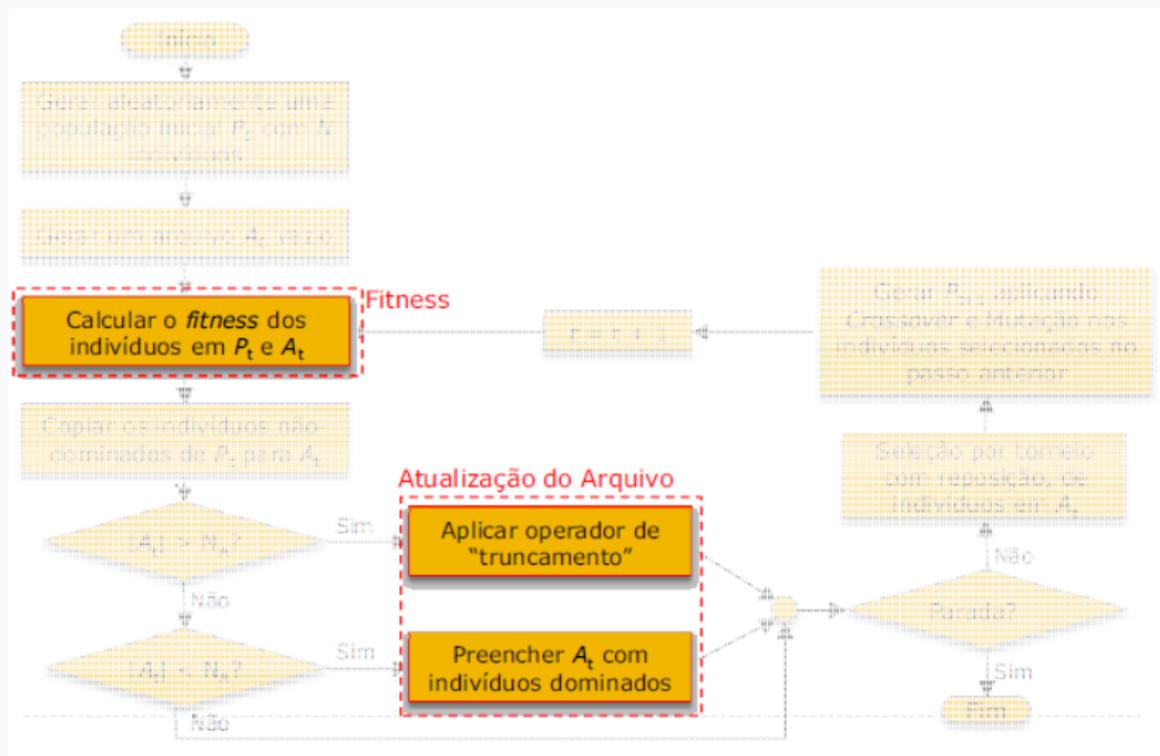
# SPEA2: Atualização do Arquivo

- ▶ Deve manter o arquivo sempre com o mesmo número de soluções  $N_A$ , definido pelo usuário;
- ▶ Busca manter uma melhor cobertura da fronteira de Pareto, evitando a eliminação de soluções dos extremos da fronteira;
- ▶ É acionado na etapa de inserção das soluções não-dominadas da população no arquivo.
  
- ▶ Se o  $|\text{arquivo}|$  é inferior a  $N_A$ :
  - ▶ Completa o arquivo com os  $N_A - |\text{arquivo}|$  melhores indivíduos dominados (*fitness*);
- ▶ Se o  $|\text{arquivo}|$  é superior a  $N_A$ :
  - ▶ Elimina os indivíduos excedentes através de um processo iterativo, que exclui os indivíduos de menor distância aos seus  $k$  vizinhos mais próximos.

# SPEA2: Fluxograma



# SPEA2: Fluxograma



- ▶ São os dois algoritmos mais utilizados (*benchmarks*);
- ▶ Ainda são considerados estado-da-arte;
- ▶ São algoritmos custosos:
  - ▶ NSGA-II:
    - ▶ Ranking por não dominância;
    - ▶ *Crowding Distance*: ordenação de todos os indivíduos por cada objetivo;
  - ▶ SPEA2:
    - ▶ Para cada indivíduo, deve ser calculada a distância a todos os demais indivíduos;
    - ▶ Tais vetores de distâncias devem ser ordenados para obtenção do  $k$ -ésimo vizinho mais próximo;

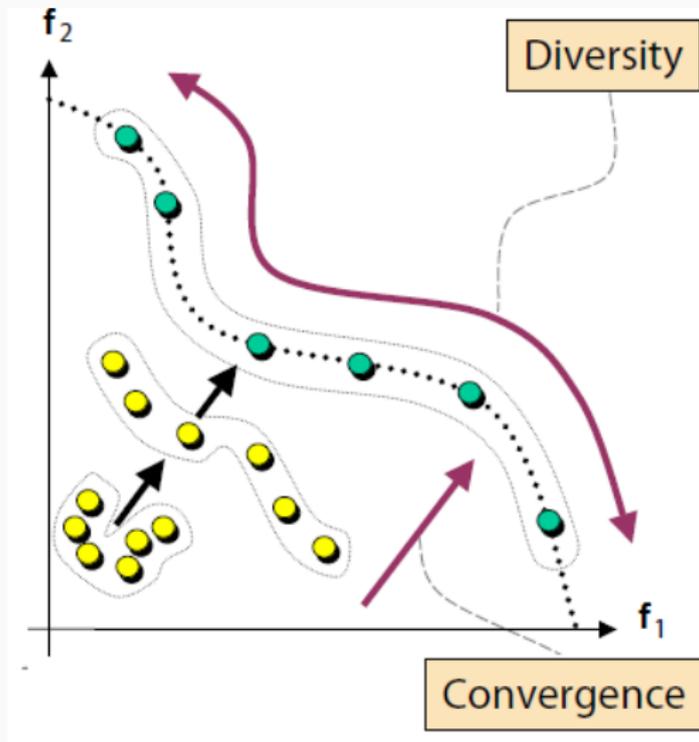
- ▶ São os dois algoritmos mais utilizados (*benchmarks*);
- ▶ Ainda são considerados estado-da-arte;
- ▶ São algoritmos custosos:
  - ▶ NSGA-II:
    - ▶ Ranking por não dominância;
    - ▶ *Crowding Distance*: ordenação de todos os indivíduos por cada objetivo;
  - ▶ SPEA2:
    - ▶ Para cada indivíduo, deve ser calculada a distância a todos os demais indivíduos;
    - ▶ Tais vetores de distâncias devem ser ordenados para obtenção do  $k$ -ésimo vizinho mais próximo;

- ▶ São os dois algoritmos mais utilizados (*benchmarks*);
- ▶ Ainda são considerados estado-da-arte;
- ▶ São algoritmos custosos:
  - ▶ NSGA-II:
    - ▶ Ranking por não dominância;
    - ▶ *Crowding Distance*: ordenação de todos os indivíduos por cada objetivo;
  - ▶ SPEA2:
    - ▶ Para cada indivíduo, deve ser calculada a distância a todos os demais indivíduos;
    - ▶ Tais vetores de distâncias devem ser ordenados para obtenção do  $k$ -ésimo vizinho mais próximo;

- ▶ São os dois algoritmos mais utilizados (*benchmarks*);
- ▶ Ainda são considerados estado-da-arte;
- ▶ São algoritmos custosos:
  - ▶ NSGA-II:
    - ▶ Ranking por não dominância;
    - ▶ *Crowding Distance*: ordenação de todos os indivíduos por cada objetivo;
  - ▶ SPEA2:
    - ▶ Para cada indivíduo, deve ser calculada a distância a todos os demais indivíduos;
    - ▶ Tais vetores de distâncias devem ser ordenados para obtenção do *k*-ésimo vizinho mais próximo;

- ▶ São os dois algoritmos mais utilizados (*benchmarks*);
- ▶ Ainda são considerados estado-da-arte;
- ▶ São algoritmos custosos:
  - ▶ NSGA-II:
    - ▶ Ranking por não dominância;
    - ▶ *Crowding Distance*: ordenação de todos os indivíduos por cada objetivo;
  - ▶ SPEA2:
    - ▶ Para cada indivíduo, deve ser calculada a distância a todos os demais indivíduos;
    - ▶ Tais vetores de distâncias devem ser ordenados para obtenção do  $k$ -ésimo vizinho mais próximo;

# Técnicas baseadas em Dominância - Pareto



## Density estimation techniques: [Silverman 86]

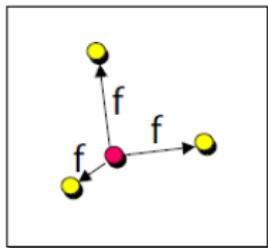
### Kernel

*MOGA, NPGA*

density estimate

=

sum of  $f$  values  
where  $f$  is a  
function of the  
distance



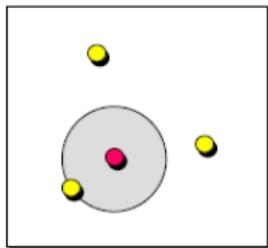
### Nearest neighbor

*NSGA-II, SPEA2*

density estimate

=

volume of the  
sphere defined by  
the nearest  
neighbor



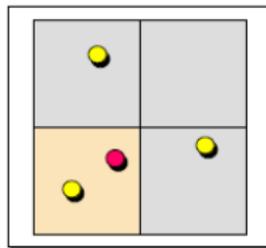
### Histogram

*PAES, PESA*

density estimate

=

number of  
solutions in the  
same box



## *Vector Evaluated Genetic Algorithm (VEGA)*

- ▶ Foi proposto por David Schaffer (meados dos anos 1980), é considerado a primeira implementação de um MOEA.
- ▶ Apenas o mecanismo de seleção do GA é modificado de modo que em cada geração um número de subpopulações é gerado efetuando seleção proporcional de acordo com cada função objetivo, por sua vez.

## *Vector Evaluated Genetic Algorithm (VEGA)*

- ▶ Foi proposto por David Schaffer (meados dos anos 1980), é considerado a primeira implementação de um MOEA.
- ▶ Apenas o mecanismo de seleção do GA é modificado de modo que em cada geração um número de subpopulações é gerado efetuando seleção proporcional de acordo com cada função objetivo, por sua vez.

## *Multi-Objective Genetic Algorithm (MOGA)*

- ▶ Proposto por Fonseca and Fleming (1993).
- ▶ *Rank* de um indivíduo corresponde ao número de indivíduos na população corrente que o dominam (+1). As soluções não dominadas tem *rank 1*.

## *Multi-Objective Genetic Algorithm (MOGA)*

- ▶ Proposto por Fonseca and Fleming (1993).
- ▶ *Rank* de um indivíduo corresponde ao número de indivíduos na população corrente que o dominam (+1). As soluções não dominadas tem *rank 1*.

## *Niched Pareto Genetic Algorithm (NPGA)*

- ▶ Proposto por Jeffrey Horn (1994).
- ▶ Usa seleção por torneio baseado na dominância Pareto. Dois indivíduos escolhidos aleatoriamente são comparados a um subconjunto da população. Vence o indivíduo não dominado. Caso ocorra empate o torneio é decidido através da (*fitness sharing*).

## *Niched Pareto Genetic Algorithm (NPGA)*

- ▶ Proposto por Jeffrey Horn (1994).
- ▶ Usa seleção por torneio baseado na dominância Pareto. Dois indivíduos escolhidos aleatoriamente são comparados a um subconjunto da população. Vence o indivíduo não dominado. Caso ocorra empate o torneio é decidido através da (*fitness sharing*).

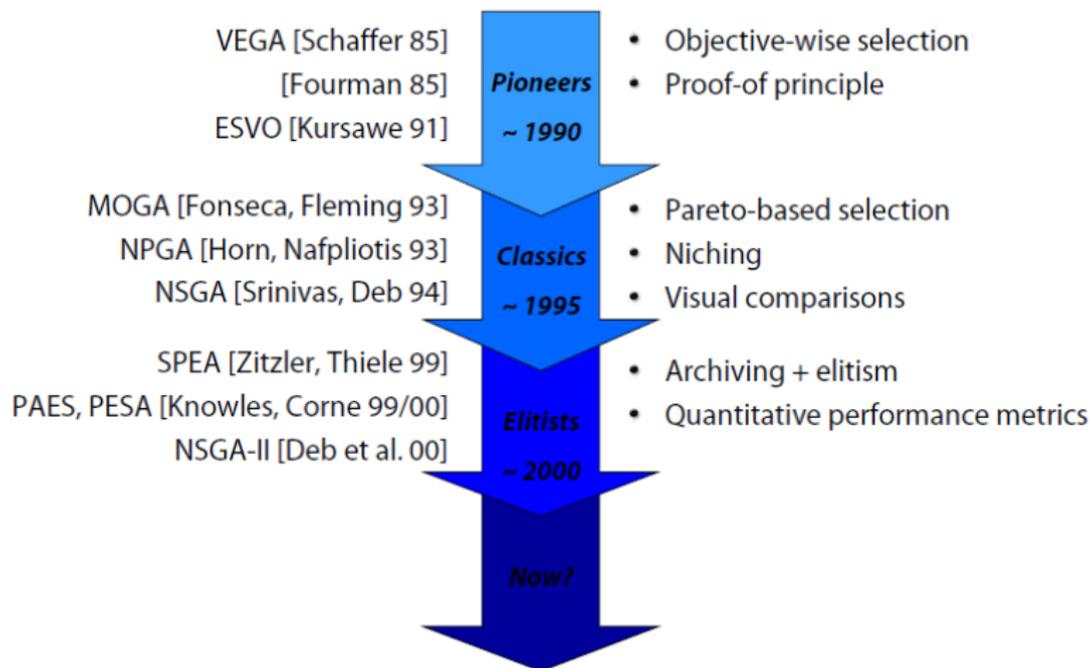
## *Pareto Archived Evolution Strategy (PAES)*

- ▶ Proposto por Joshua D. Knowles and David W. Corne. (1999)
- ▶ Usa uma estratégia evolutiva (1 + 1) combinado com um arquivo histórico que registra as soluções não dominadas encontradas anteriormente.

## *Pareto Archived Evolution Strategy (PAES)*

- ▶ Proposto por Joshua D. Knowles and David W. Corne. (1999)
- ▶ Usa uma estratégia evolutiva (1 + 1) combinado com um arquivo histórico que registra as soluções não dominadas encontradas anteriormente.

# Histórico dos MOEA's



- ▶ *Multiobjective Messy Genetic Algorithm (MOMGA, MOMGA-II);*
- ▶ *Pareto Envelope-based Selection Algorithm (PESA);*
- ▶ *Differential Evolution;*
- ▶ *Particle Swarm Optimization;*
- ▶ *Immune Algorithms; ...*

- ▶ Formas de relaxamento da dominância Pareto
- ▶ Métodos híbridos e outros bio-inspirados;
- ▶ Parâmetros de controle e auto-adaptação;
- ▶ Otimização com Muitos-objetivos (*Many-objective problems*) ...

- ▶ Formas de relaxamento da dominância Pareto
- ▶ Métodos híbridos e outros bio-inspirados;
- ▶ Parâmetros de controle e auto-adaptação;
- ▶ Otimização com Muitos-objetivos (*Many-objective problems*) ...

- ▶ Formas de relaxamento da dominância Pareto
- ▶ Métodos híbridos e outros bio-inspirados;
- ▶ Parâmetros de controle e auto-adaptação;
- ▶ Otimização com Muitos-objetivos (*Many-objective problems*) ...

- ▶ Formas de relaxamento da dominância Pareto
- ▶ Métodos híbridos e outros bio-inspirados;
- ▶ Parâmetros de controle e auto-adaptação;
- ▶ Otimização com Muitos-objetivos (*Many-objective problems*) ...

- ▶ Takahashi, R.H.C.; Notas de Aula: Otimização Escalar e Vetorial.  
<http://www.decom.ufop.br/moreira/disciplinas/OVE.zip>
- ▶ J. A. Ramírez, F. Campelo, F. G. Guimarães, Lucas S. Batista, Ricardo H. C. Takahashi, Notas de Aula de Otimização, 2010.  
<http://www.decom.ufop.br/moreira/disciplinas/Notas1.pdf>
- ▶ Coello Coello, C. A., Lamont, G. B., Van Veldhuizen, D. A.: Evolutionary Algorithms for Solving Multi- Objective Problems, 2nd. Ed., Springer, 2007.
- ▶ DEB K.: Multi-objective Optimization using Evolutionary Algorithms. John Wiley & Sons, 2001.
- ▶ E. Zitzler, M. Laumanns & L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm". Technical Report, ETH Zurich, 2001.
- ▶ E. Zitzler, M. Laumanns & L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization". In: K. Glannakoglou, D. Tsahalis, J. Periaux and T. Fogarty (Eds.), "Evolutionary Methods for Design, Optimization and Control", 2002.

- ▶ Knowles, J.D., Corne, D.W. (1999) The Pareto Archived Evolution Strategy : A New Baseline Algorithm for Pareto Multiobjective Optimisation. In Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99), pages 98-105.
- ▶ Horn, J.; Nafpliotis, N.; Goldberg, D.E., "A niched Pareto genetic algorithm for multiobjective optimization," Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on , vol., no., pp.82,87 vol.1, 27-29 Jun 1994
- ▶ Fonseca, Carlos M. and Fleming, Peter J. Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. Proceedings of the 5th International Conference on Genetic Algorithms, 416-423, 1993.