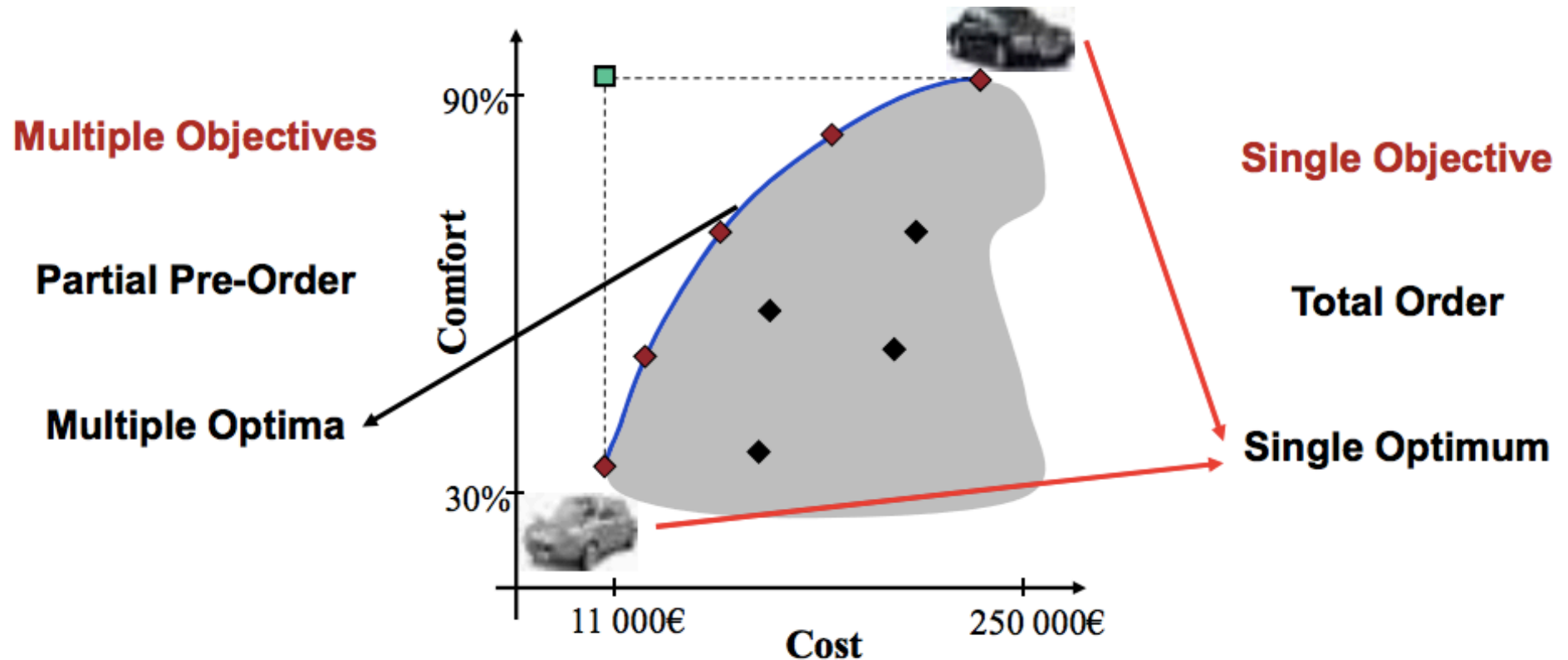


Tópicos em Computação Evolucionária Otimização Multi-objetiva

Computação Natural

Gisele L. Pappa

Otimização de Um ou Muitos Objetivos



- Otimização de um objetivo é um caso particular da otimização de múltiplos objetivos

Otimização Multi-objetiva

- Grande parte dos problemas requer a otimização de dois ou mais critérios (objetivos)
 - Abordagem convencional para tratar desse problema:
 - **Agregação de objetivos:** Combinar os objetivos em uma fórmula, atribuindo pesos a diferentes objetivos
 - $\text{Fitness} = \frac{2}{3} \text{Objective_1} + \frac{1}{3} \text{Objective_2}$
- (assumindo que os objetivos estão normalizados para retornar valores dentro do mesmo intervalo, como 0..1)

Otimização Multi-objetiva

- Desvantagens da abordagem convencional
 - **Objetivos** diferentes são normalmente **não-comensuráveis**, isto é, eles medem aspectos diferentes da qualidade de uma solução, que não deveriam ser adicionados ou subtraídos em uma mesma fórmula
 - Retorna **uma solução**, enquanto em problemas MO pode ser conveniente retornar um **conjunto de soluções**, representando diferentes configurações (*trade-offs*) entre os objetivos
- Solução: Algoritmos Multi-Objetivos baseados na **dominância de Pareto**

Otimização Multi-objetiva

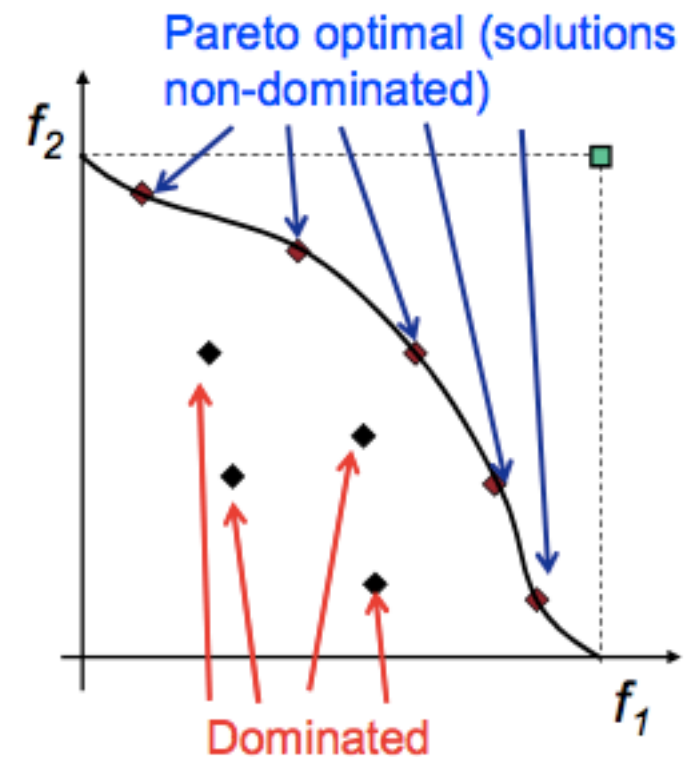
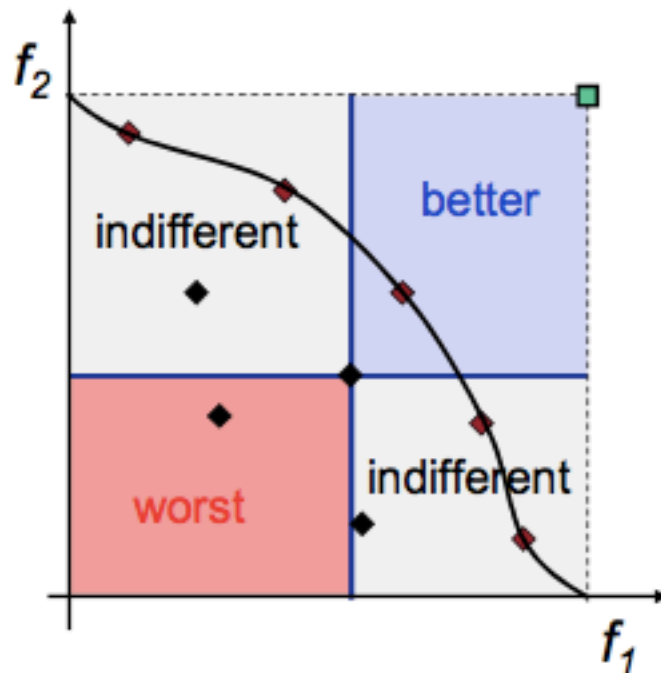
- Exige que o algoritmo funcione em dois passos:
 1. Busca
 - Algoritmos evolucionários ou qualquer outro algoritmo de busca
 - Baseada no conceito de dominância de Pareto
 2. Tomada de Decisão
 - Escolha de uma solução entre o conjunto de soluções retornadas

Dominância de Pareto

- Um solução S_1 domina uma solução S_2 se e apenas se:
 - S_1 não é pior que S_2 em nenhum objetivo
 - S_1 é obrigatoriamente melhor que S_2 com respeito a pelo menos um objetivo
- O conjunto de soluções não-dominadas é chamado frente de Pareto

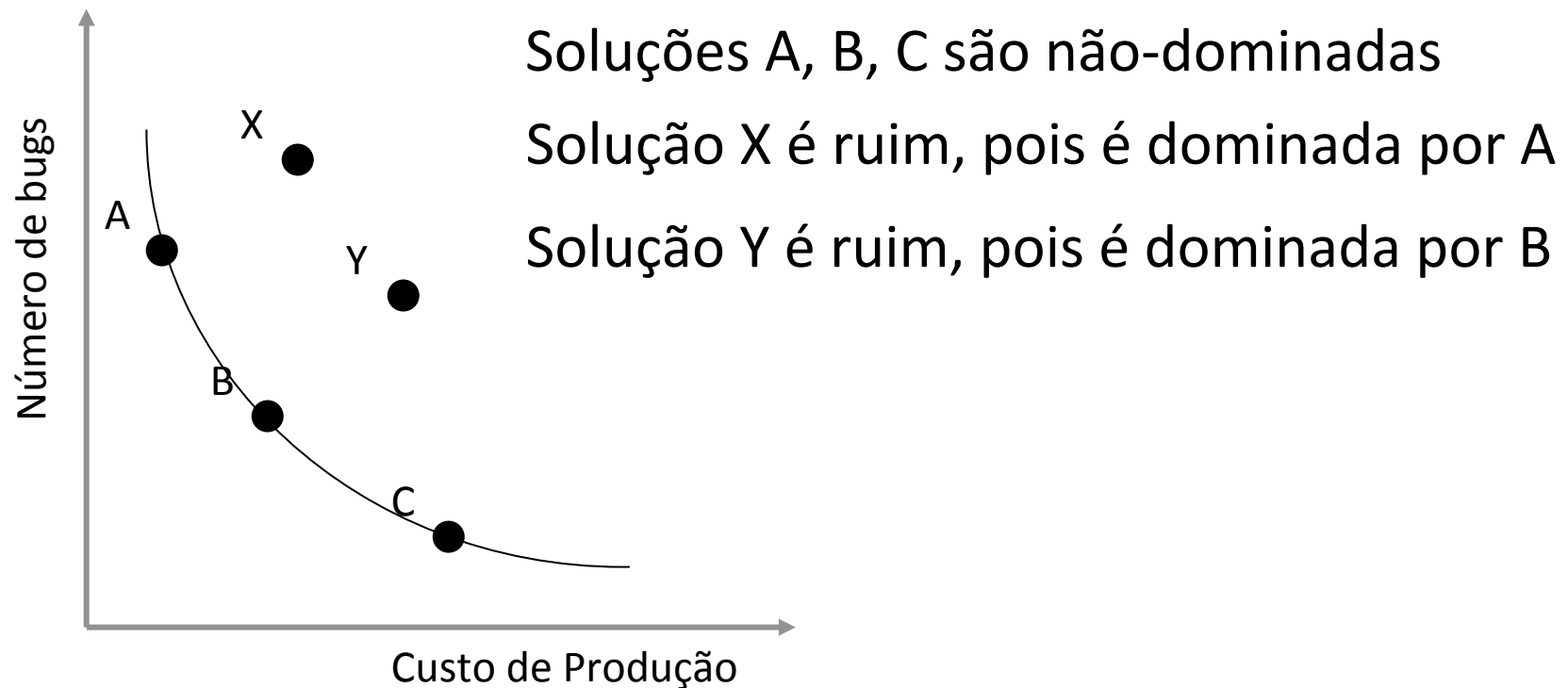
Conceito de Dominância de Pareto

maximize $f_1(x_i), f_2(x_i)$



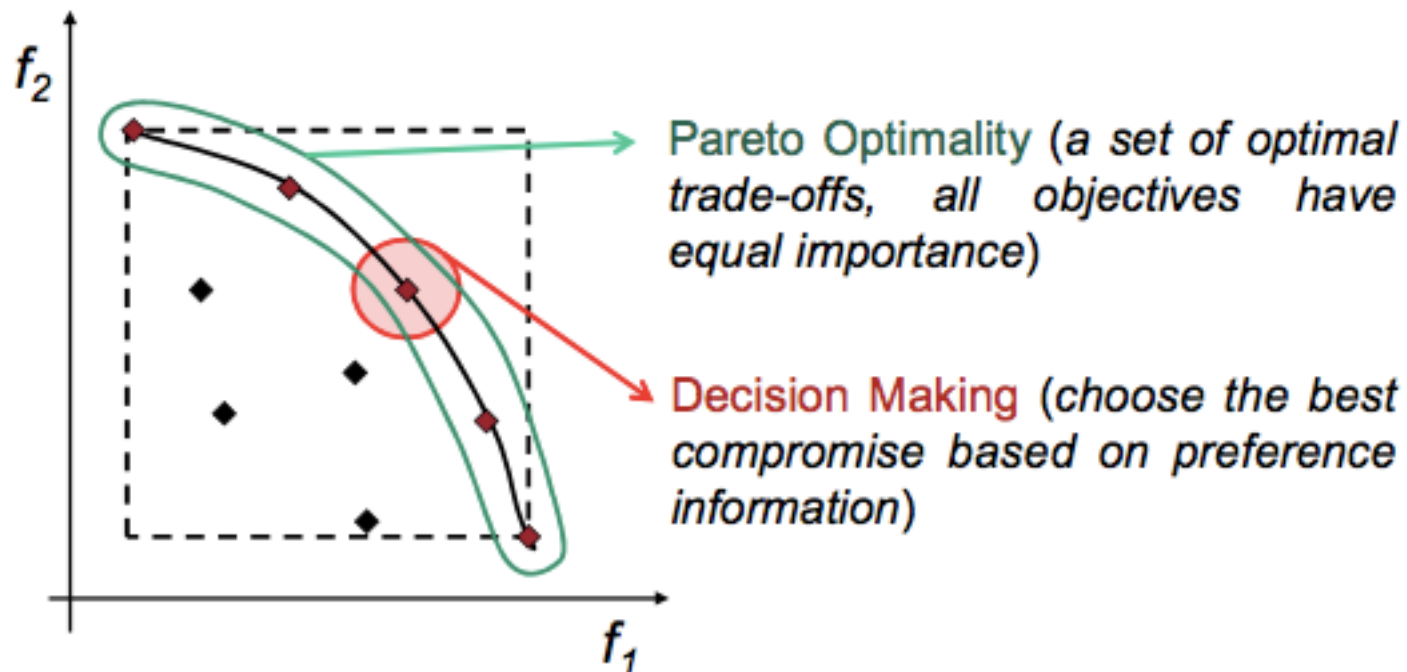
Conceito de Dominância de Pareto

- **Minimizar** o custo de produção e o número de bugs encontrados em um programa



Processo de tomada de Decisão

- Após o processo de busca, uma solução deve ser escolhida pelo usuário

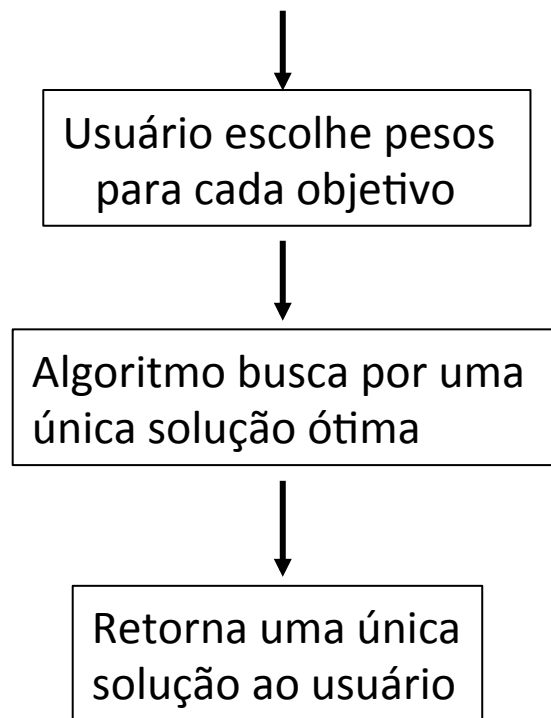


Processo de tomada de Decisão

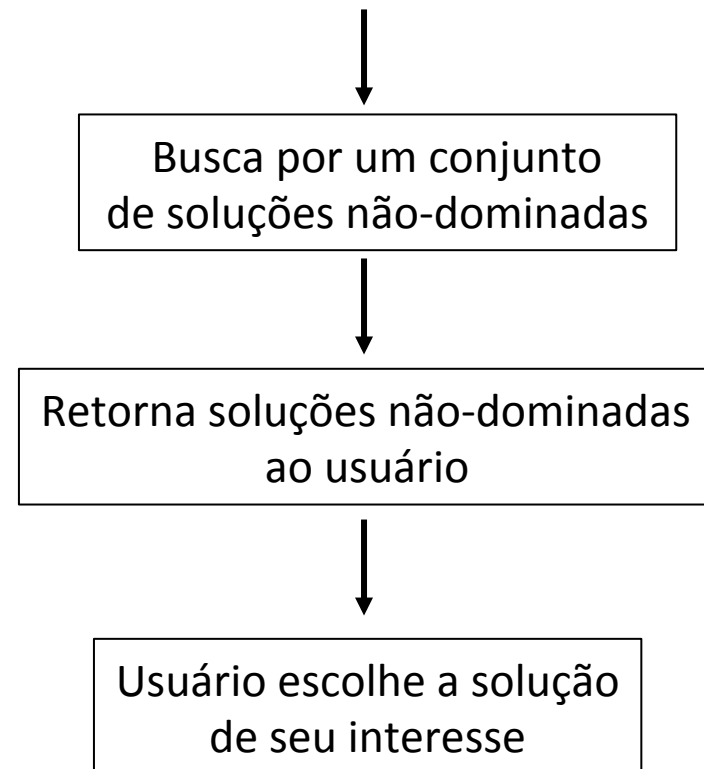
- Deve incluir informações sobre as **preferências do usuário**, já que isso não é considerado durante a busca do algoritmo
- Essa informação de preferências pode ser fornecida pelo usuário em 3 momentos:
 - A priori – gera uma otimização de um objetivo ou baseada em pesos.
 - **A posteriori** – busca encontra múltiplas soluções
 - Progressivamente – durante a busca
 - **Combinação das 3 anteriores.**

Otimização Multi-objetiva

Fórmula combinando objetivos



MO baseada em Pareto



Perspectiva Histórica

**Primeiros algoritmos
(agregação de funções)**

Schaffer (1985) – VEGA
Kursawe (1990) – VOES

Clássicos

Fonseca and Fleming (1993) – MOGA
Srinivas and Deb (1994) – NSGA
Horn, Nafpliotis and Goldberg (1994) – NPGA

Elitistas

Zitzler and Thiele (1999) – SPEA, (2001) – SPEA2
Deb and co-authors (2000) – NSGA-II
Knowles and Corne (2000) – PAES, PESA

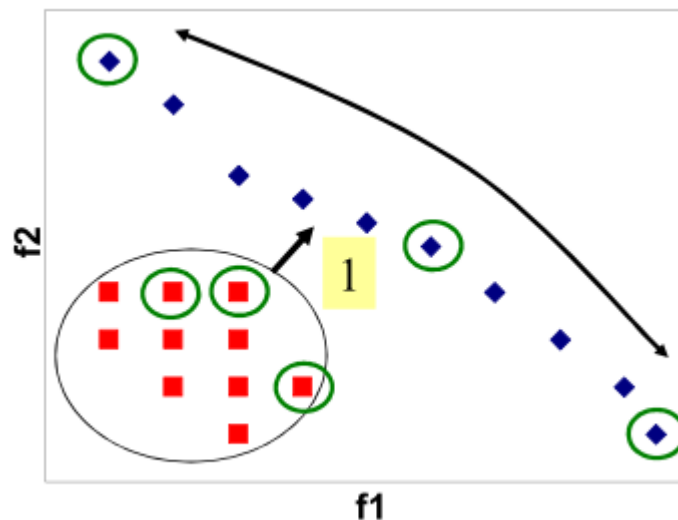
**Incorporação de
preferências**

Fleisher (2003) – Simulating Annealing
Zitzler and Künzli, (2004) – IBEA
Emmerich et al. (2005) – SMS-EMOA
Zitzler et al. (2008) – SPAM

O que muda em relação a um AG tradicional?

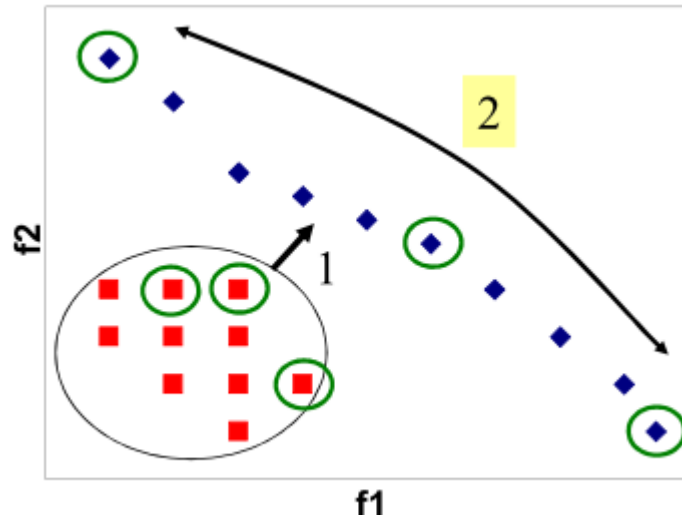
Algoritmos Evolucionários Multiobjetivos

- EAs são bons métodos para encontrar múltiplas soluções presentes no fronte de Pareto em uma única rodada (paralelismo implícito)
- Quando se fala em MOEAs, queremos:
 - Guiar a busca na direção do fronte de Pareto



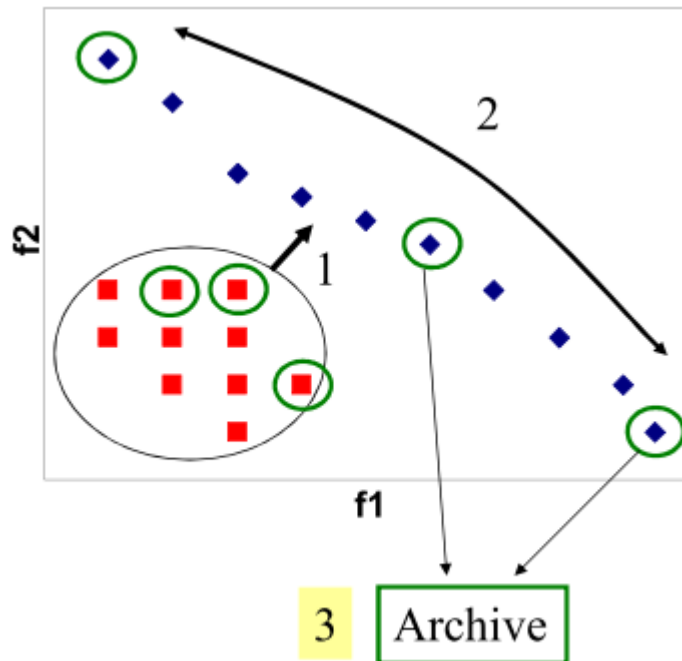
Fitness

Algoritmos Evolucionários Multiobjetivos



- Manter um conjunto diverso de soluções não-dominadas

Estimar a densidade



- Não deixar que soluções não dominadas sejam perdidas

Elitismo

Algoritmos Evolucionários Multiobjetivos

- Cálculo da fitness e seleção
- 3 esquemas principais de seleção:
 - Seleção de Objetivos “por Troca” (*Switching objectives*) – a cada seleção, um objetivo diferente é considerado como fitness
 - Seleção por Agregação com parâmetros variados – cada objetivo recebe um peso, mas esse peso varia de indivíduo para indivíduo dentro de uma mesma geração
 - Seleção por Pareto – cálculo da fitness baseado no conceito de dominância de Pareto- várias abordagens

Perspectiva Histórica

**Primeiros algoritmos
(agregação de funções)**

Schaffer (1985) – VEGA
Kursawe (1990) – VOES

Clássicos

Fonseca and Fleming (1993) – MOGA
Srinivas and Deb (1994) – NSGA
Horn, Nafpliotis and Goldberg (1994) – NPGA

Elitistas

Zitzler and Thiele (1999) – SPEA, (2001) – SPEA2
Deb and co-authors (2000) – NSGA-II
Knowles and Corne (2000) – PAES, PESA

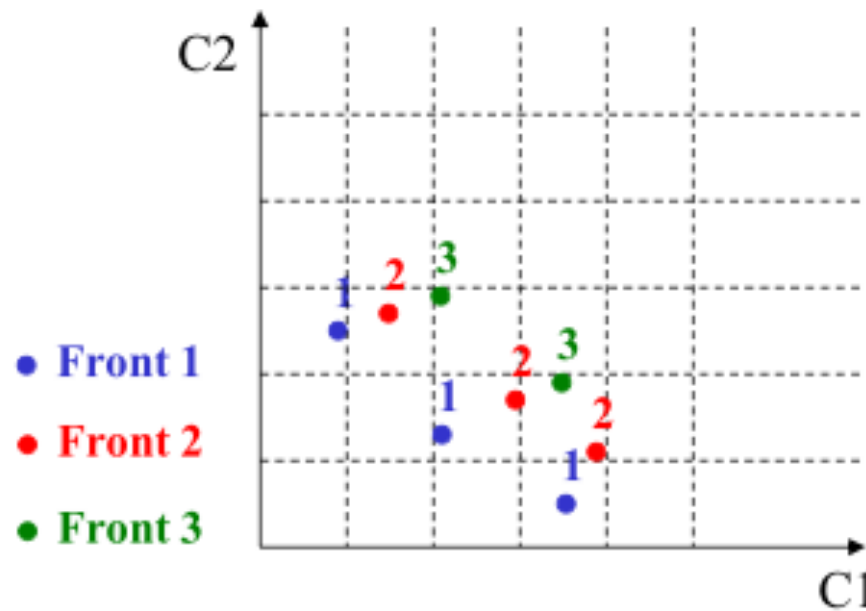
**Incorporação de
preferências**

Fleisher (2003) – Simulating Annealing
Zitzler and Künzli, (2004) – IBEA
Emmerich et al. (2005) – SMS-EMOA
Zitzler et al. (2008) – SPAM

Algoritmos Clássicos

- MOGA (Fonseca e Fleming 1993) - FFGA
 - Propõe um rank baseado em Pareto
 - Utiliza niching para manter diversidade
- O ranking de um indivíduo é igual ao número de indivíduos na população pelos quais ele é dominado + 1
- Atribui um valor de fitness através de interpolação
- A fitness final é calculada utilizando fitness sharing de todos os indivíduos com o mesmo ranking

Algoritmos Clássicos



Step 1- Ranking Function

$$FO(\text{Front 1}) > FO(\text{Front 2}) > FO(\text{Front 3})$$

Step 2- Sharing

$$FO'_i = \frac{FO_i}{m'_i}$$

M_i é o número de indivíduos no frente i

Perspectiva Histórica

**Primeiros algoritmos
(agregação de funções)**

Schaffer (1985) – VEGA
Kursawe (1990) – VOES

Clássicos

Fonseca and Fleming (1993) – MOGA
Srinivas and Deb (1994) – NSGA
Horn, Nafpliotis and Goldberg (1994) – NPGA

Elitistas

Zitzler and Thiele (1999) – SPEA, (2001) – SPEA2
Deb and co-authors (2000) – NSGA-II
Knowles and Corne (2000) – PAES, PESA

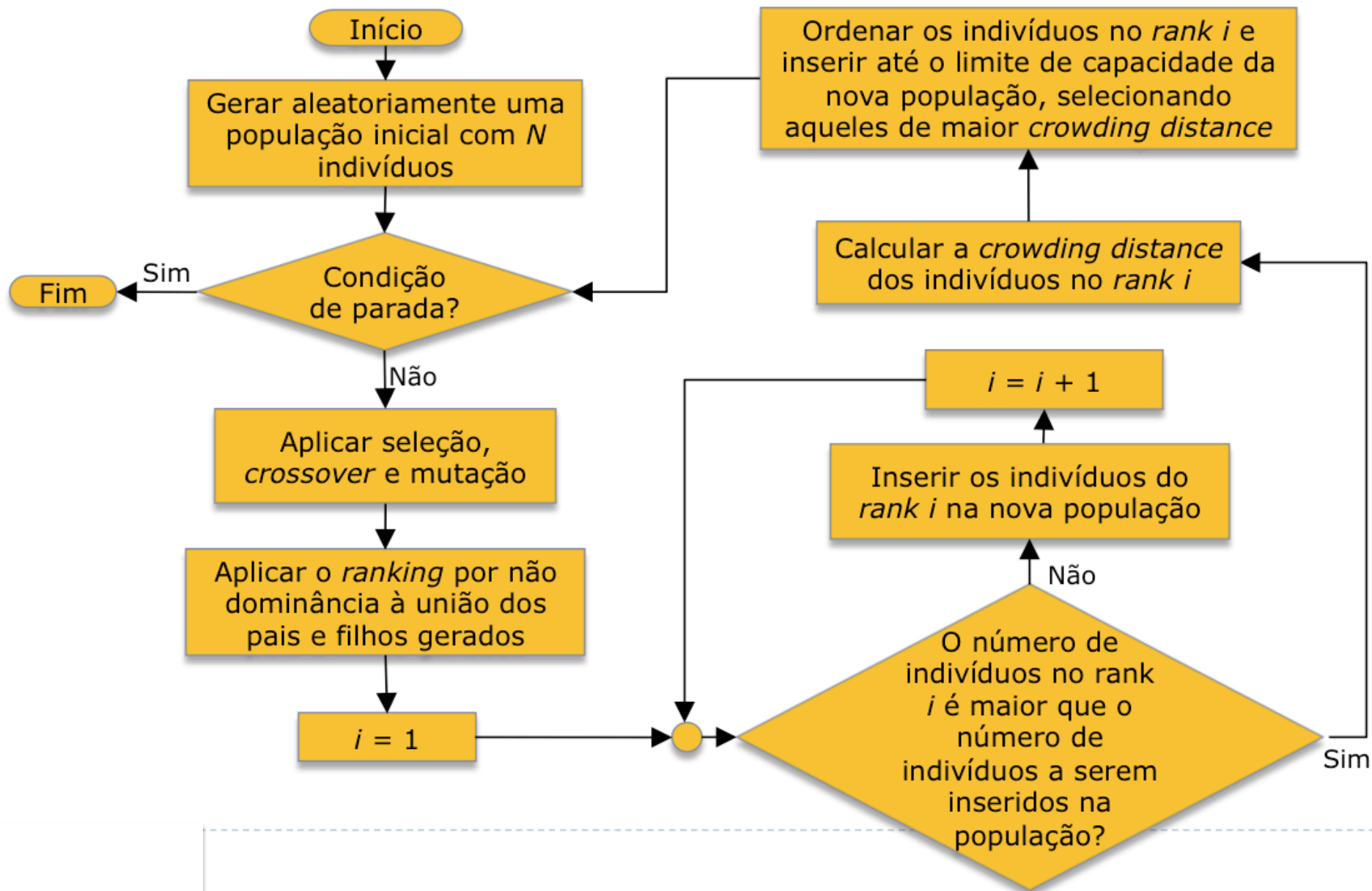
**Incorporação de
preferências**

Fleisher (2003) – Simulating Annealing
Zitzler and Künzli, (2004) – IBEA
Emmerich et al. (2005) – SMS-EMOA
Zitzler et al. (2008) – SPAM

Seleção por Pareto

- As mais populares são as utilizadas no
 - NSGA II (*Nondominated Sorting Genetic Algorithms*)
 - SPEA 2 (*Strength Pareto Evolutionary Algorithm*)

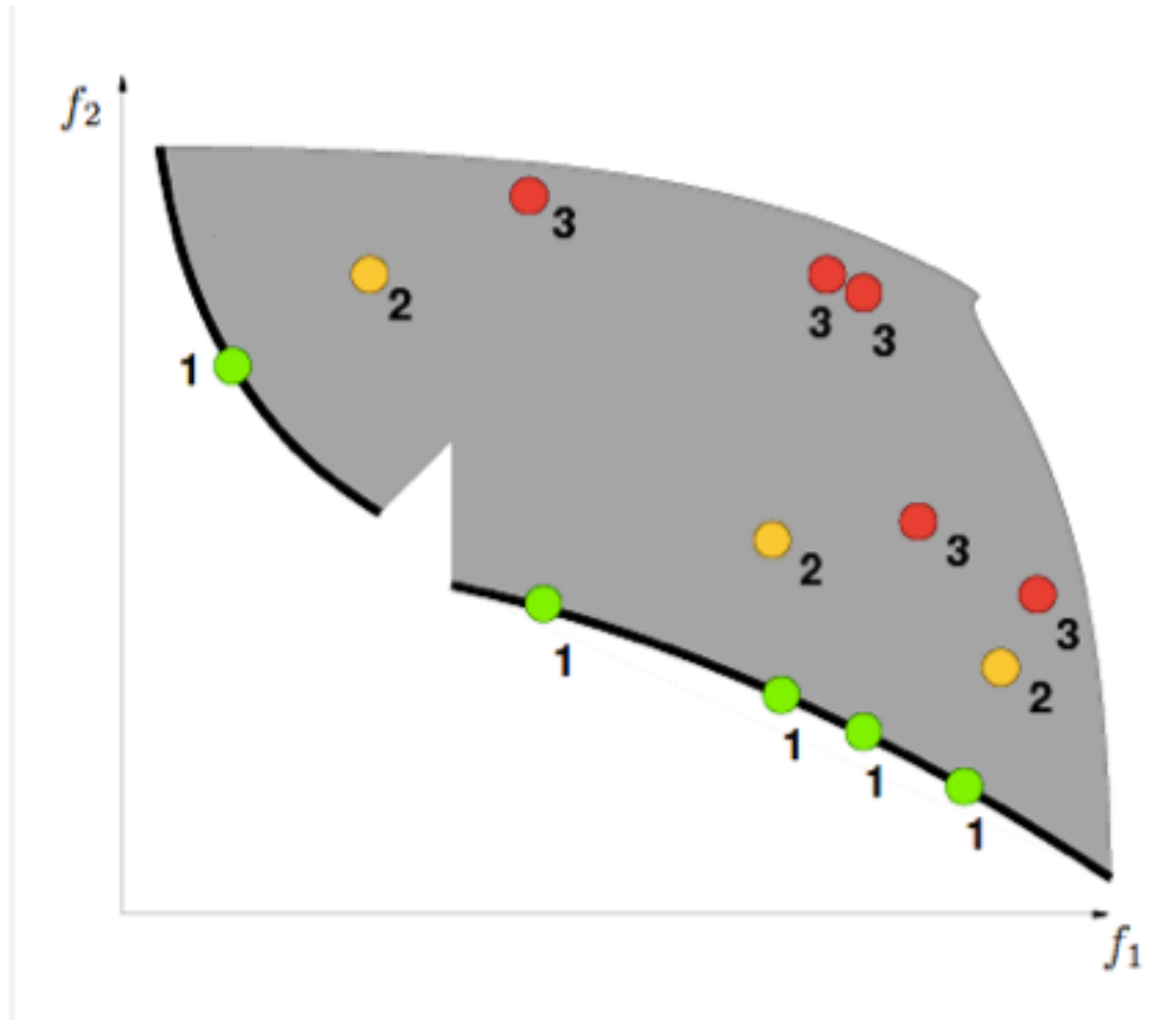
NSGA-II



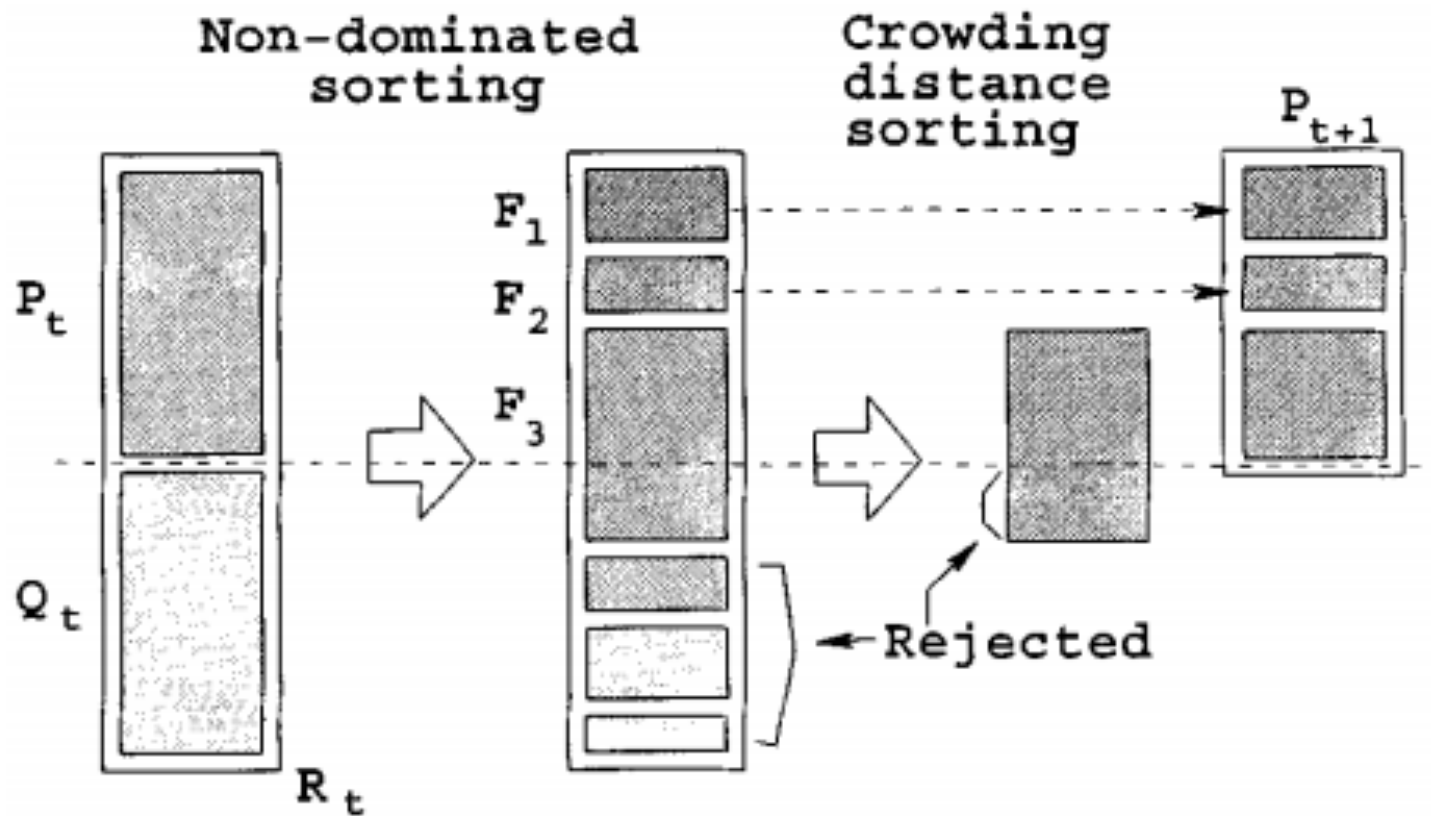
NSGA-II

- Inicialização – anterior ao loop
 - Cria população inicial P_0
 - Ordena P_0 baseada em não-dominância
 - Atribui fitness de acordo com o ranking de dominância: quanto menor o valor, maior a fitness
 - Seleção por torneio
 - Mutação e cruzamento criam Q_0

NSGA-II



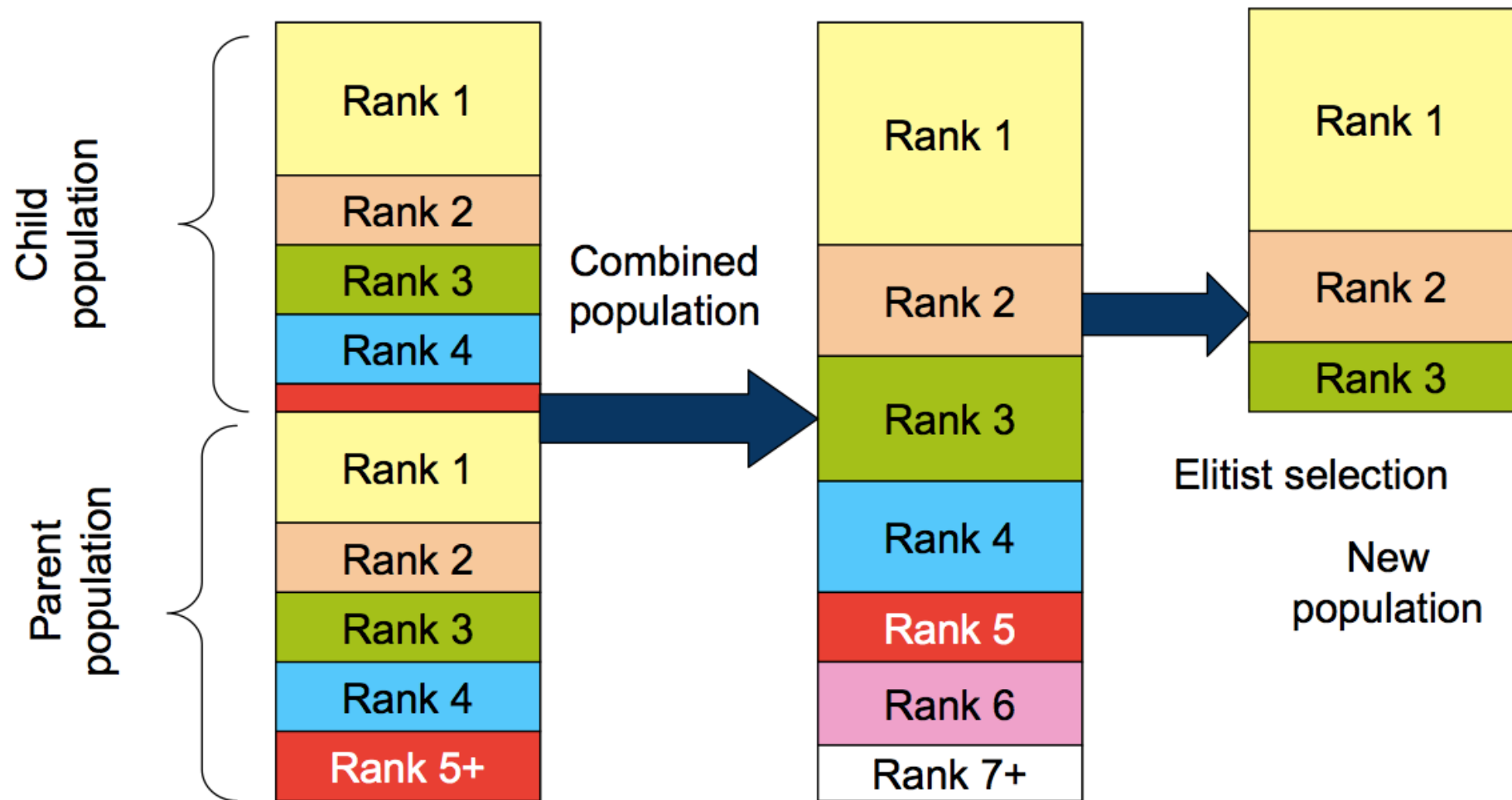
NSGA- II



NSGA-II

- Loop principal
 - $R_t = P_t + Q_t$
 - Ordeno R_t de acordo com dominância
 - Crio P_{t+1} adicionando a ela os melhores indivíduos de R_t
 - Crio Q_{t+1} com seleção por torneio, cruzamento e mutação em P_{t+1}
 - Nesse loop principal, uso tanto o id do fronte quanto crowding distance como critério de seleção.

NSGA-II (Elitism)

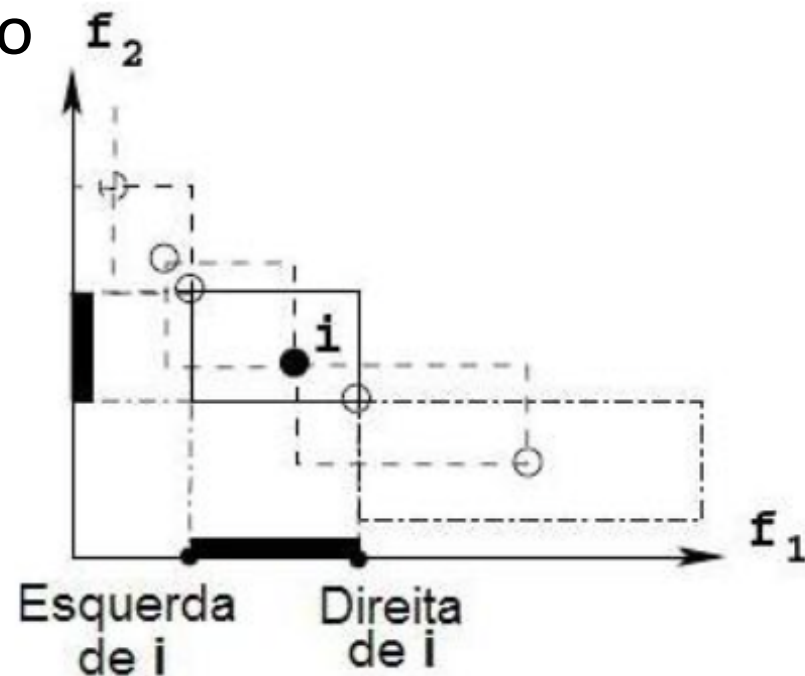


NSGA-II - Niching

- Distância por *Crowding*:
 - Usa como métrica a distância média entre dois vizinhos
 - Calcula o perímetro de um cubóide definido pelos 2 vizinhos mais próximos no mesmo fronte
- Seleção por torneio: usa a distância por *crowding* como métrica de desempate

NSGA-II - Niching

- O *crowding distance* busca estimar a vizinhança de cada indivíduo no espaço de objetivos do problema.
- Ela corresponde ao volume do hiper-cubo formado pelas soluções imediatamente anteriores e posteriores a cada objetivo



NSGA-II - Niching

crowding-distance-assignment(\mathcal{I})

$l = |\mathcal{I}|$

for each i , set $\mathcal{I}[i]_{\text{distance}} = 0$

for each objective m

$\mathcal{I} = \text{sort}(\mathcal{I}, m)$

$\mathcal{I}[1]_{\text{distance}} = \mathcal{I}[l]_{\text{distance}} = \infty$

for $i = 2$ to $(l - 1)$

$\mathcal{I}[i]_{\text{distance}} = \mathcal{I}[i]_{\text{distance}} + (\mathcal{I}[i+1].m - \mathcal{I}[i-1].m) / (f_m^{\max} - f_m^{\min})$

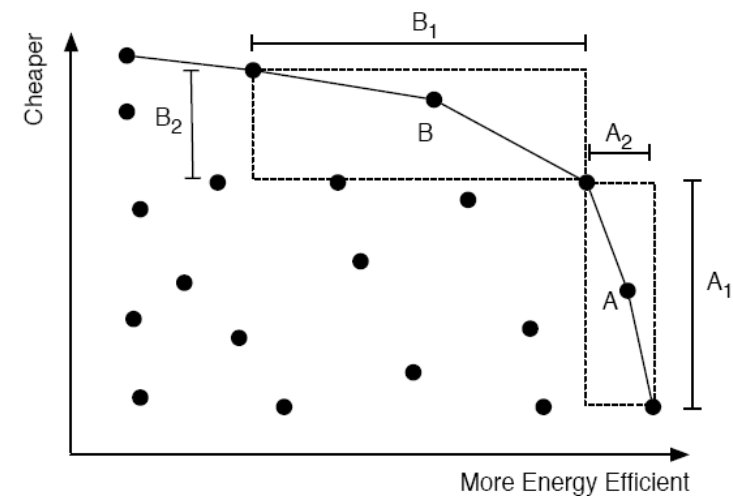
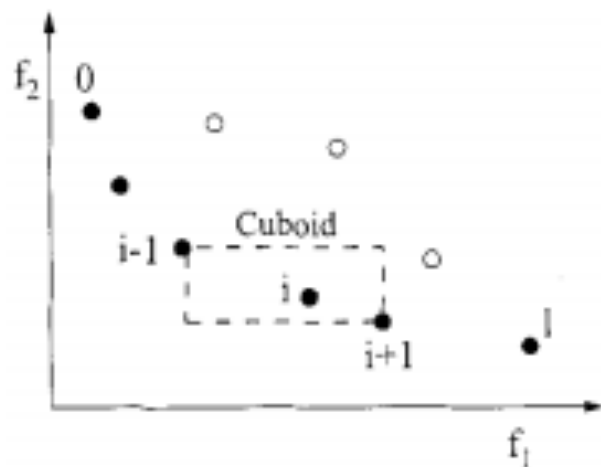
number of solutions in \mathcal{I}

initialize distance

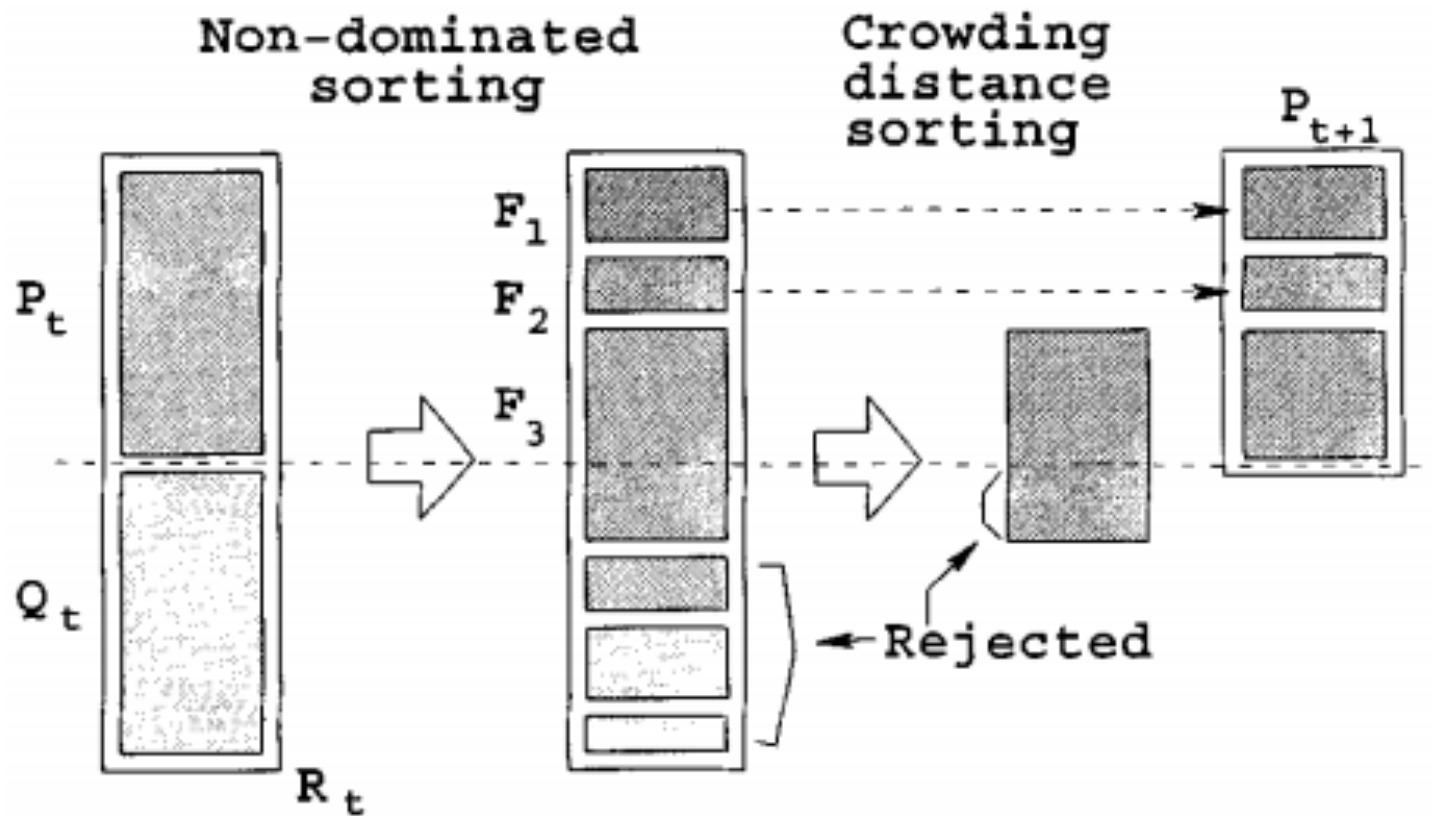
sort using each objective value

so that boundary points are always selected

for all other points



NSGA- II



NSGA-II

$R_t = P_t \cup Q_t$

$\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$

$P_{t+1} = \emptyset$ and $i = 1$

until $|P_{t+1}| + |\mathcal{F}_i| \leq N$

$\text{crowding-distance-assignment}(\mathcal{F}_i)$

$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$

$i = i + 1$

Sort(\mathcal{F}_i, \prec_n)

$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$

$Q_{t+1} = \text{make-new-pop}(P_{t+1})$

$t = t + 1$

combine parent and offspring population

$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$, all nondominated fronts of R_t

until the parent population is filled

calculate crowding-distance in \mathcal{F}_i

include i th nondominated front in the parent pop

check the next front for inclusion

sort in descending order using \prec_n

choose the first $(N - |P_{t+1}|)$ elements of \mathcal{F}_i

 use selection, crossover and mutation to create

 a new population Q_{t+1}

increment the generation counter

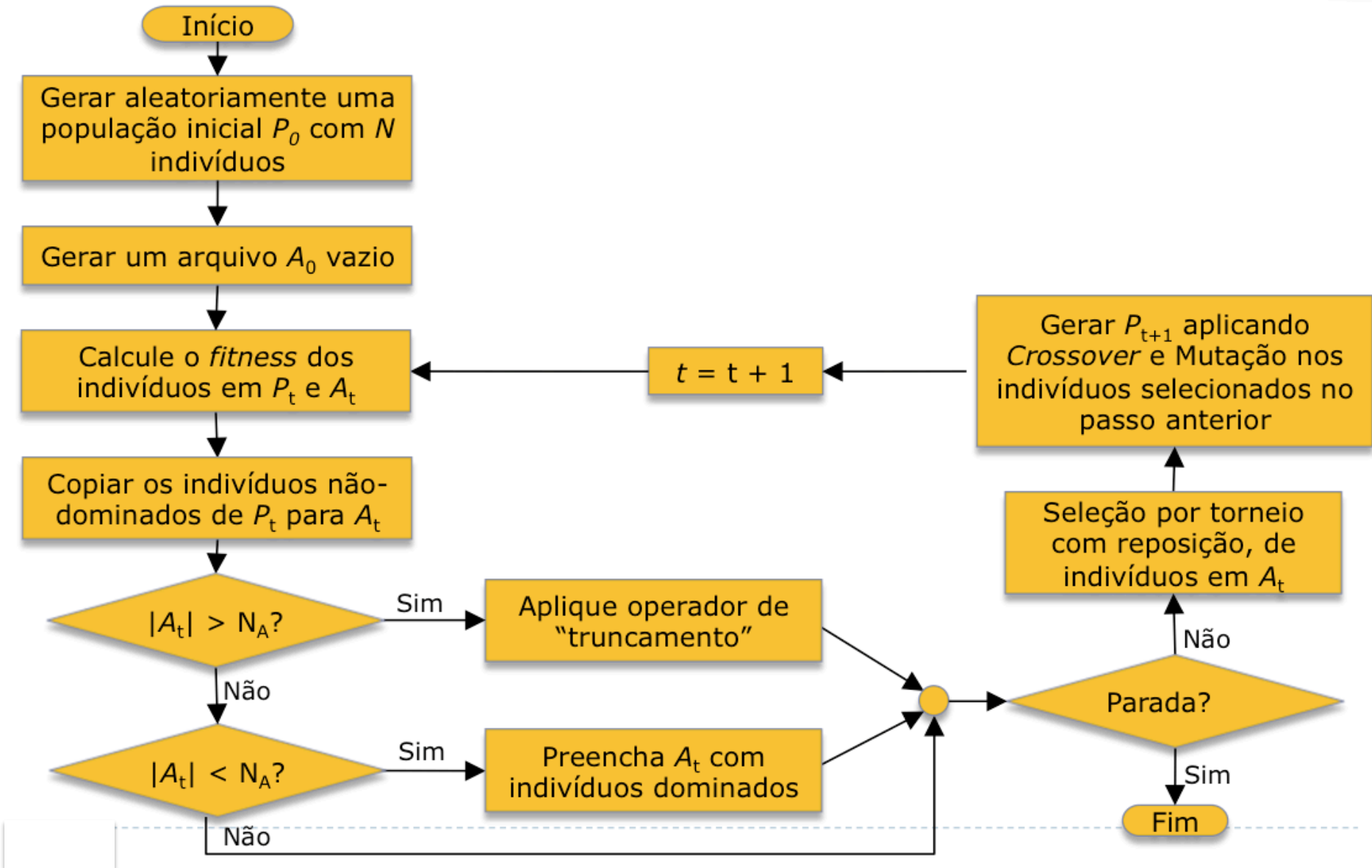
Seleção por Pareto

- As mais populares são as utilizadas no
 - NSGA II (*Nondominated Sorting Genetic Algorithms*)
 - SPEA 2 (*Strength Pareto Evolutionary Algorithm*)

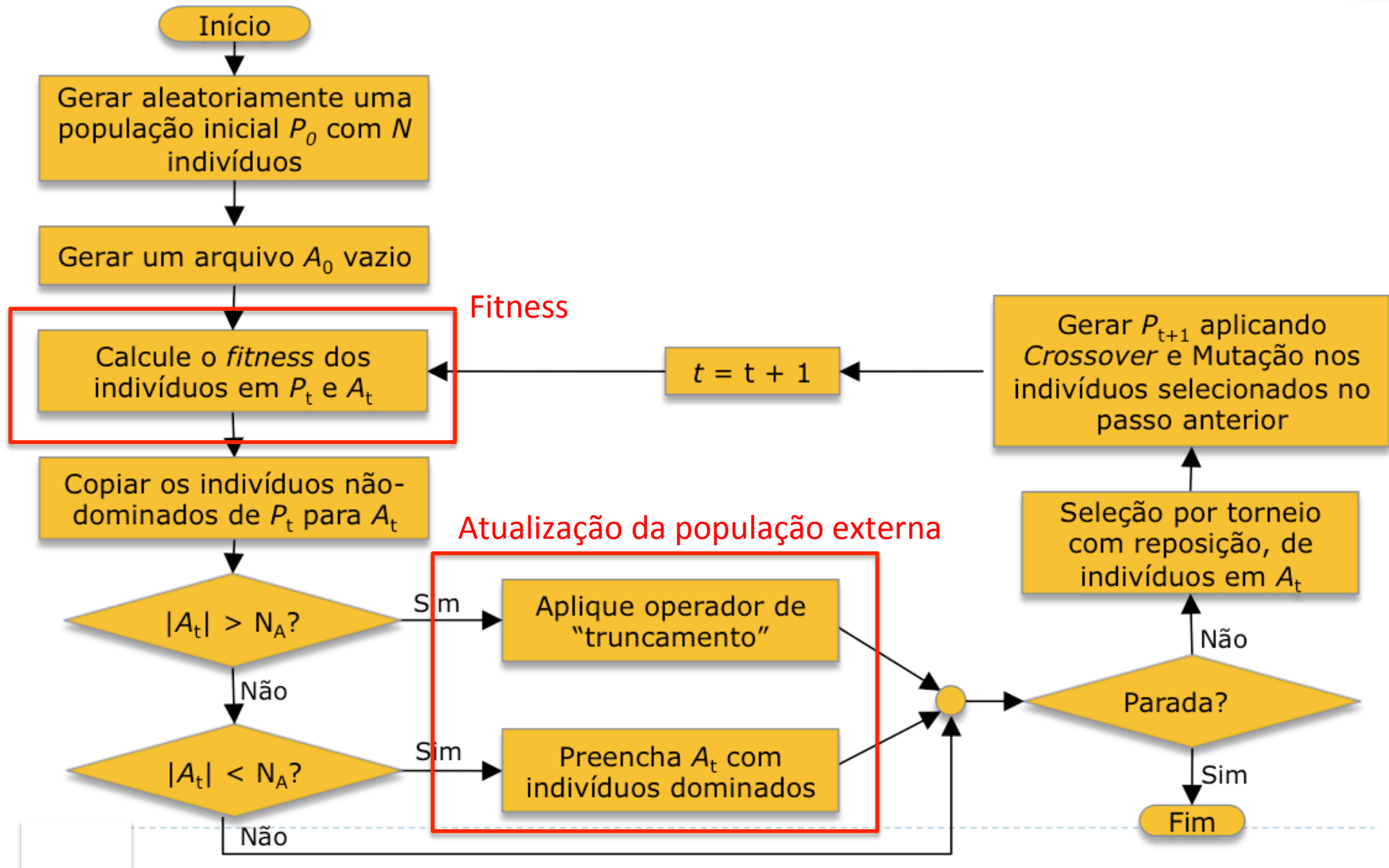
SPEA-2

- Trabalha com a população e um conjunto externo de indivíduos
- Conjunto externo: contém todos os indivíduos não-dominados únicos da população
 - Limitado por um tamanho máximo
- Fitness é calculada tanto para indivíduos da população quanto do conjunto externo

SPEA-2



SPEA-2



SPEA-2

Gera a população inicial P_0 e um arquivo externo vazio A_0
 $t=1$

Enquanto ($t < \text{NumeroGeracoes}$)

 Calcula a fitness dos indivíduos em P_t e A_t

A_{t+1} = indivíduos não-dominados em P_t and A_t .

 se (tamanho de $A_{t+1} > N$) reduza A_{t+1}

 senão

 Insira em A_{t+1} com os indivíduos não dominados de P_t e A_t .

 se ($t > T$) retorne A_{t+1}

 Use seleção por torneio ($k=2$) usando indivíduos de A_{t+1}

 Aplique mutação de cruzamento e crie P_{t+1}

SPEA2- Cálculo da Fitness

- Cada indivíduo i é associado a uma força S
 $S(i)$ = número de indivíduos que i domina
- Fitness bruto $R(i)$
 - Soma as forças de todos os indivíduos que dominam i (quanto maior, pior)
- Densidade $D(i)$

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

distância ao k -ésimo vizinho mais próximo no espaço de objetivos

 - Leva à exploração de regiões pouco povoadas

$$\mathbf{F}_{\text{SPEA2}}(i) = \mathbf{R}(i) + \mathbf{D}(i)$$

SPEA2: Cálculo do *Fitness*

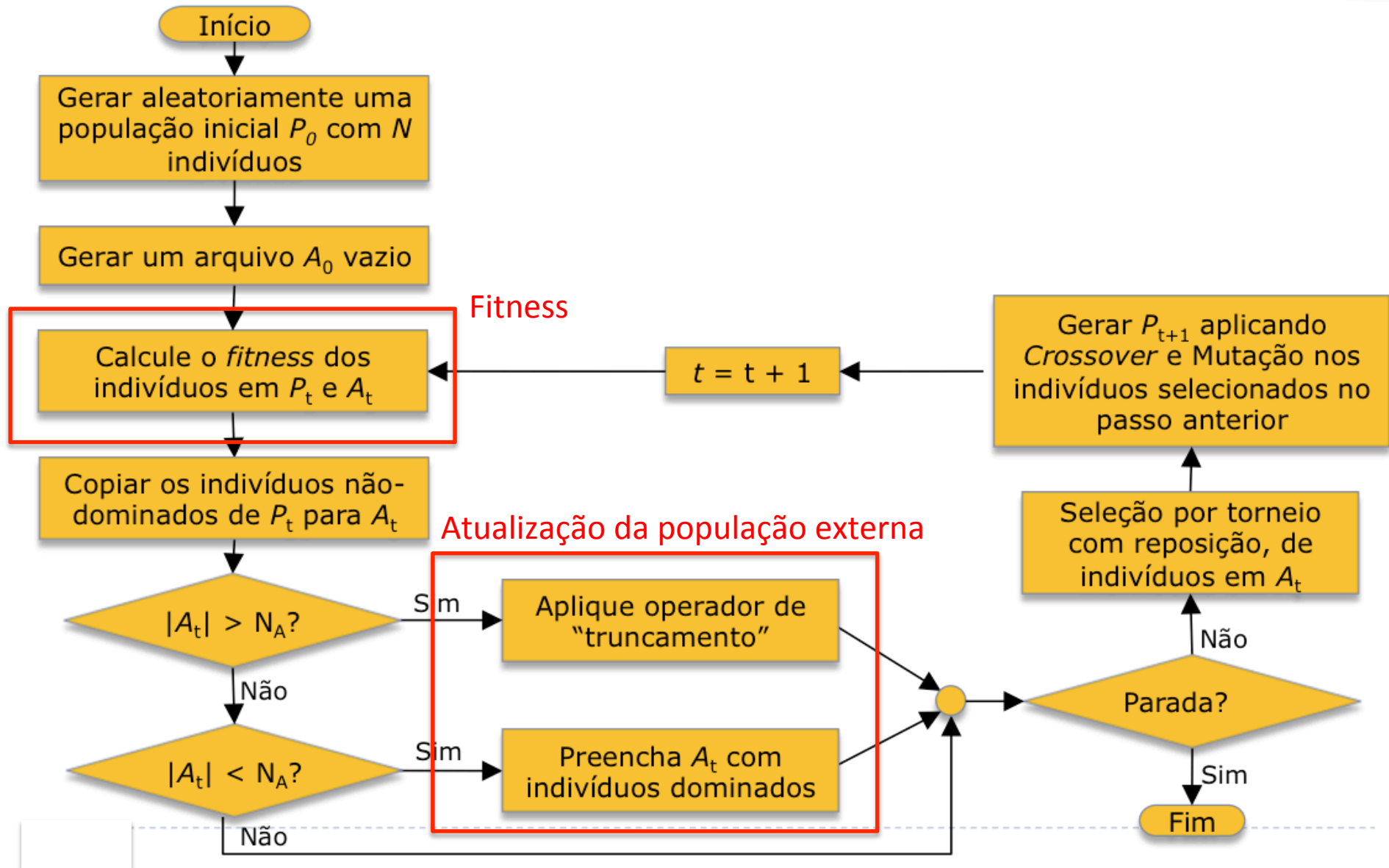
$$F_{\text{SPEA2}}(i) = R(i) + D(i)$$

Diversidade no Espaço de Objetivos

Proximidade da Fronteira de Pareto

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

SPEA-2



SPEA2 – Atualização da população externa

- População externa tem sempre tamanho N_A (definido pelo usuário)
- Mantêm uma boa cobertura da fronteira de pareto
- Quando ela ainda não está completa
 - Completada com melhores indivíduos dominados (de acordo com fitness)
- Quando está completa
 - Elimina os indivíduos de menor distância a seus k vizinhos mais próximos

NSGA-II e SPEA2

- São os algoritmos mais utilizados
- Ainda considerados estado da arte
- São custosos
 - NSGA-II
 - Ranking por não dominância
 - Crowding distance: ordenação de todos os indivíduos por cada objetivo
 - SPEA2
 - Calcula distâncias entre indivíduos par a par
 - Ordena essas distâncias para encontrar o vizinho mais próximo

Leitura Recomendada

- *SPEA2: Improving the Strength Pareto Evolutionary Algorithm For Multiobjective Optimization*, E. Zitzler and K. Giannakoglou and D. Tsahalis and J. Periaux and K. Papailiou and T. Fogarty , 2002.
- *A fast and elitist multiobjective genetic algorithm: NSGA-II*, K Deb, A Pratap, S Agarwal, T Meyarivan, *Evolutionary Computation, IEEE Transactions on* 6 (2), 182-197