Algoritmos Meméticos

Universidade Federal do Paraná
Tópicos em Inteligência Artificial
Dra. Aurora Pozo

Alunos:
Ademir Roberto Freddo – *freddo@utfpr.edu.br*
Robison Cris Brito – *robison@utfpr.edu.br*

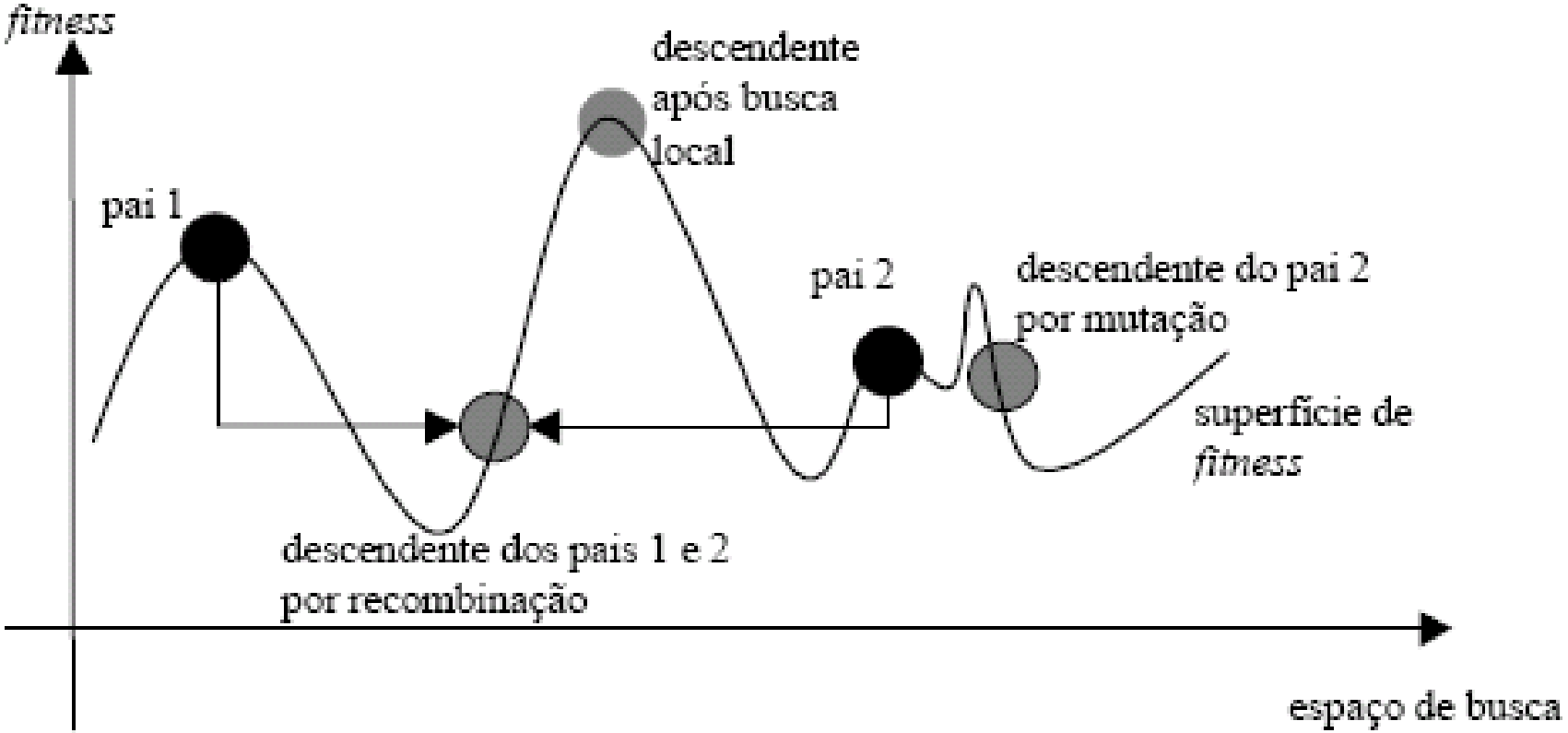
Introdução

- Meta-heurística;
- Baseados na Teoria dos Algoritmos Evolutivos;
- Trabalho de Moscato [1];
- Termo memes;
  - Evolução cultural
- Algoritmos Genéticos Híbridos
  - Busca Local
  - Evoluir autonomamente (sem sofrer mutações ou recombinações)

Meméticos e Busca Local

- Partem de informações contidas no genótipo;
- Concentrar a busca em determinadas regiões promissoras do espaço;
- Direcionam a evolução.

Operadores Evolutivos x Busca Local



Características

- Incorporar a maior quantidade de conhecimento possível;
- Conceito ontogênica;
- Etapas
  - População Inicial;
  - Geração;
  - Função de Avaliação;
  - Recombinação
  - Mutação

Algoritmo - Inicialização

procedure Algoritmo Memético:
begin
    initializePopulation *Pop_principal* usando Gera_Pop_Inicial();
    forEach indivíduo i ∈ *Pop_principal* **do** *i* := Busca Local(*i*);
    forEach indivíduo i ∈ *Pop_principal* **do** avaliaFitness(*i*);
    repeat /*loco das gerações */
        for *i* := 1 **to** *#recombinações* **do**
            selectToMerge um conjunto *Spar* ⊆ *Pop_principal;*
            novo_indifíduo := Recombinar*(Spar);*
            Avalia_Fitness(*novo_indivíduo*);
            addInPopulation indivíduo *novo_indivíduo* **to** *Pop_Intermediária*;
        endFor;
        for *i* := 1 **to** *#mutações* **do**
            selectToMutate indivíduo *i* ∈ *Pop_Intermediária;*
            im := Efatuar_Mutação(i)*;*
            Avalia_Fitness(*im*);
            addInPopulation indivíduo *im* **to** *Pop_Intermediária*;
        endFor;
        forEach indivíduo *i* ∈ *Pop_intermediária* **do** *i* := Busca_Local(*i*);
        forEach indivíduo *i* ∈ *Pop_intermediária* **do** Avalia_Fitness(*i*);
        Pop_principal := Seleciona_Pop (*Pop_intermediária, Pop_principal);*
        Apagar(*Pop_intermediária*);
    until (condição_de_parada = True);
end;

Algoritmo - Recombinação

procedure Algoritmo Memético:
begin
initializePopulation *Pop_principal* usando Gera_Pop_Inicial();
forEach indivíduo i ∈ *Pop_principal* **do** *i* := Busca Local(*i*);
forEach indivíduo i ∈ *Pop_principal* **do** avaliaFitness(*i*);
repeat /*loco das gerações */
for *i* := 1 **to** *#recombinações* **do**
selectToMerge um conjunto *Spar* ⊆ *Pop_principal;*
novo_indifíduo := Recombinar*(Spar);*
Avalia_Fitness(*novo_indivíduo*);
addInPopulation indivíduo *novo_indivíduo* **to** *Pop_Intermediária*;
endFor;
for *i* := 1 **to** *#mutações* **do**
selectToMutate indivíduo *i* ∈ *Pop_Intermediária;*
im := Efatuar_Mutação(i)*;*
Avalia_Fitness(*im*);
addInPopulation indivíduo *im* **to** *Pop_Intermediária*;
endFor;
forEach indivíduo *i* ∈ *Pop_intermediária* **do** *i* := Busca_Local(*i*);
forEach indivíduo *i* ∈ *Pop_intermediária* **do** Avalia_Fitness(*i*);
Pop_principal := Seleciona_Pop (*Pop_intermediária, Pop_principal);*
Apagar(*Pop_intermediária*);
until (condição_de_parada = True);
end;

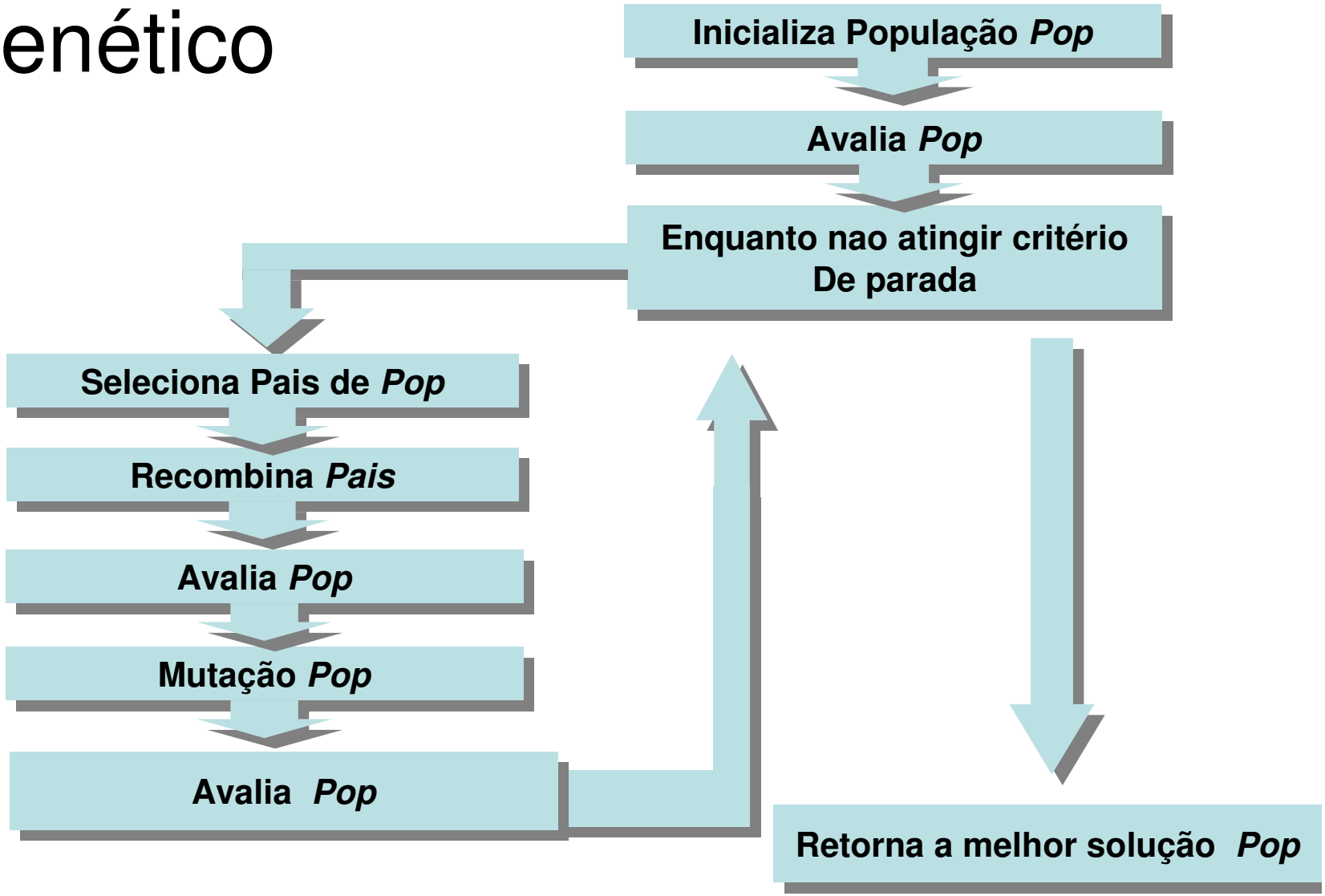
Algoritmo - Mutação

procedure Algoritmo Memético:
begin
initializePopulation *Pop_principal* usando Gera_Pop_Inicial();
forEach indivíduo i ∈ *Pop_principal* **do** *i* := Busca Local(*i*);
forEach indivíduo i ∈ *Pop_principal* **do** avaliaFitness(*i*);
repeat /*loco das gerações */
for *i* := 1 **to** *#recombinações* **do**
selectToMerge um conjunto *Spar* ⊆ *Pop_principal;*
novo_indifíduo := Recombinar*(Spar);*
Avalia_Fitness(*novo_indivíduo*);
addInPopulation indivíduo *novo_indivíduo* **to** *Pop_Intermediária*;
endFor;
for *i* := 1 **to** *#mutações* **do**
selectToMutate indivíduo *i* ∈ *Pop_Intermediária;*
im := Efatuar_Mutação(i)*;*
Avalia_Fitness(*im*);
addInPopulation indivíduo *im* **to** *Pop_Intermediária*;
endFor;
forEach indivíduo *i* ∈ *Pop_intermediária* **do** *i* := Busca_Local(*i*);
forEach indivíduo *i* ∈ *Pop_intermediária* **do** Avalia_Fitness(*i*);
Pop_principal := Seleciona_Pop (*Pop_intermediária, Pop_principal);*
Apagar(*Pop_intermediária*);
until (condição_de_parada = True);
end;

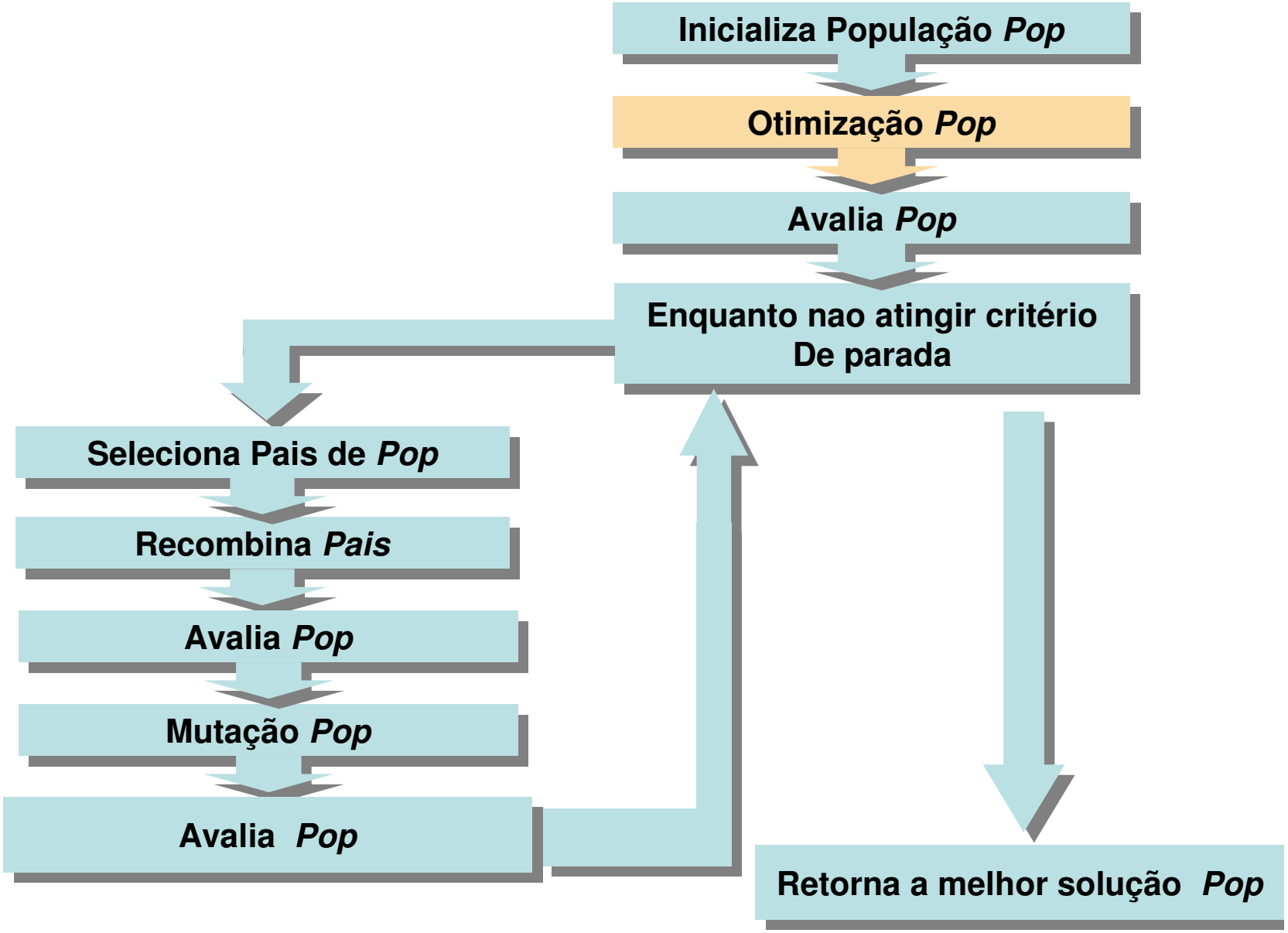
Algoritmo – Atualização da População

procedure Algoritmo Memético:
begin
    initializePopulation *Pop_principal* usando Gera_Pop_Inicial();
    forEach indivíduo i ∈ *Pop_principal* **do** *i* := Busca Local(*i*);
    forEach indivíduo i ∈ *Pop_principal* **do** avaliaFitness(*i*);
    repeat /*loco das gerações */
        for *i* := 1 **to** *#recombinações* **do**
            selectToMerge um conjunto *Spar* ⊆ *Pop_principal;*
            novo_indifíduo := Recombinar*(Spar);*
            Avalia_Fitness(*novo_indivíduo*);
            addInPopulation indivíduo *novo_indivíduo* **to** *Pop_Intermediária*;
        endFor;
        for *i* := 1 **to** *#mutações* **do**
            selectToMutate indivíduo *i* ∈ *Pop_Intermediária;*
            im := Efatuar_Mutação(i)*;*
            Avalia_Fitness(*im*);
            addInPopulation indivíduo *im* **to** *Pop_Intermediária*;
        endFor;
        forEach indivíduo *i* ∈ *Pop_intermediária* **do** *i* := Busca_Local(*i*);
        forEach indivíduo *i* ∈ *Pop_intermediária* **do** Avalia_Fitness(*i*);
        Pop_principal := Seleciona_Pop (*Pop_intermediária, Pop_principal);*
        Apagar(*Pop_intermediária*);
    until (condição_de_parada = True);
end;

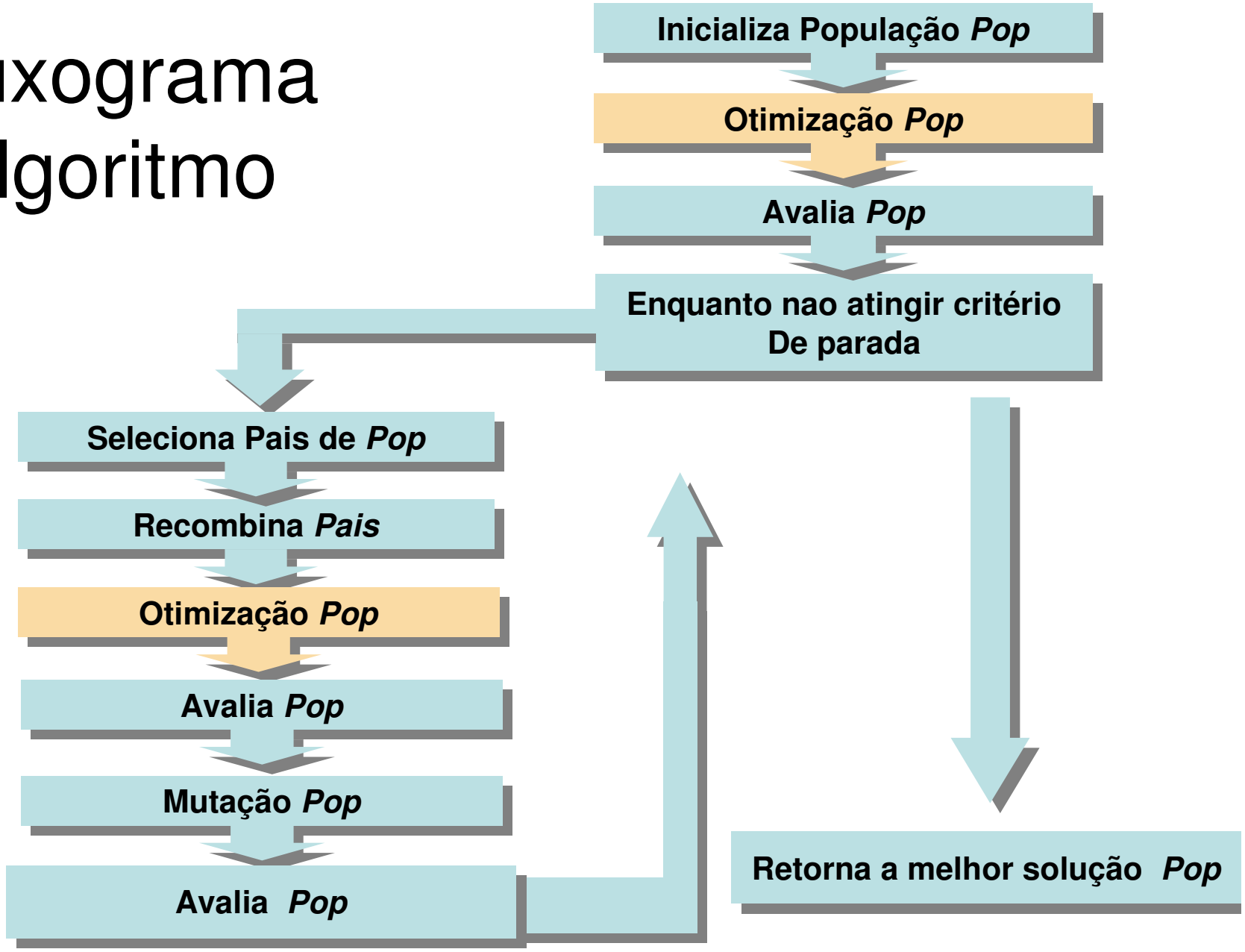
Algoritmo Genético



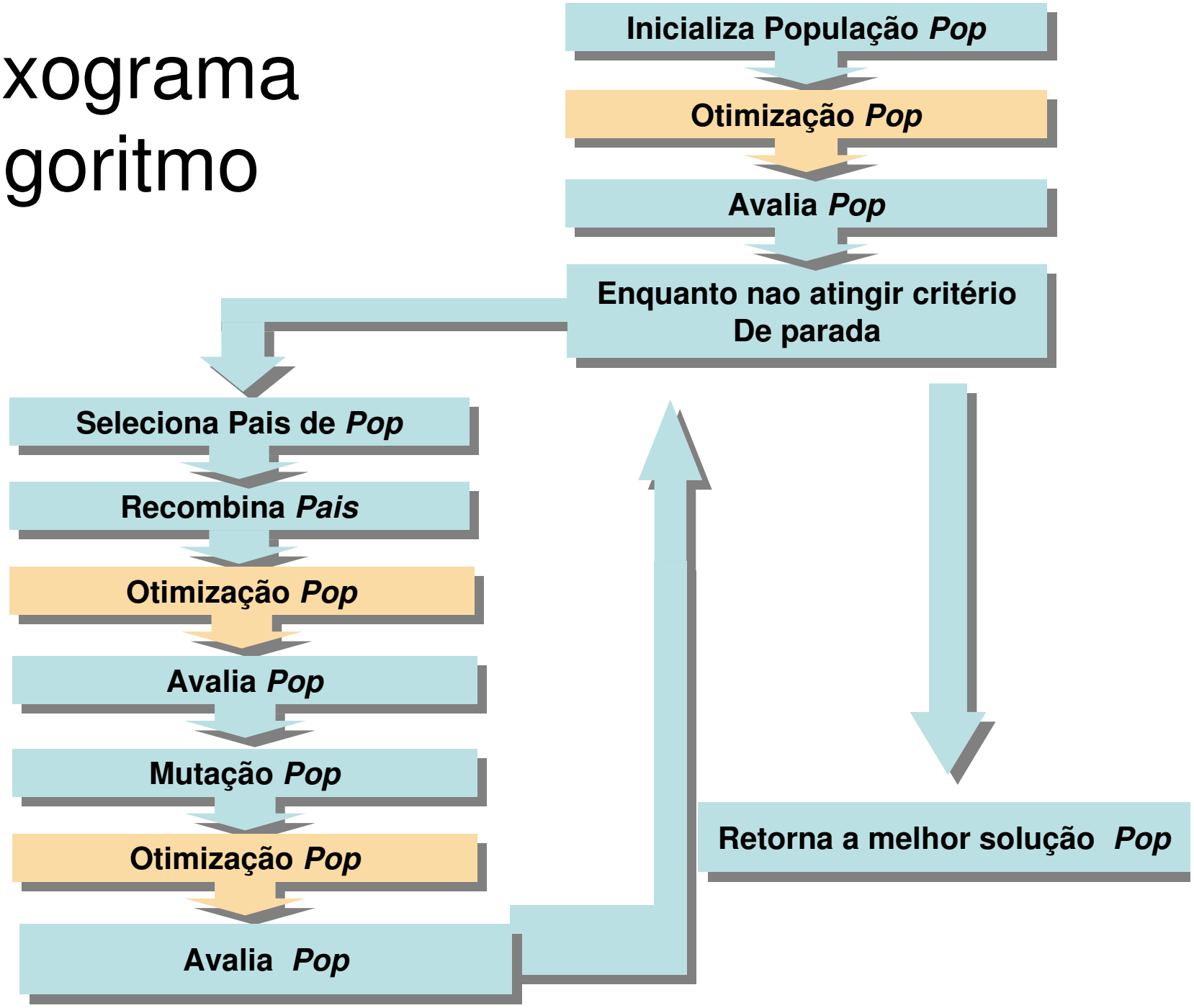
Fluxograma - Algoritmo



Fluxograma Algoritmo



Fluxograma Algoritmo

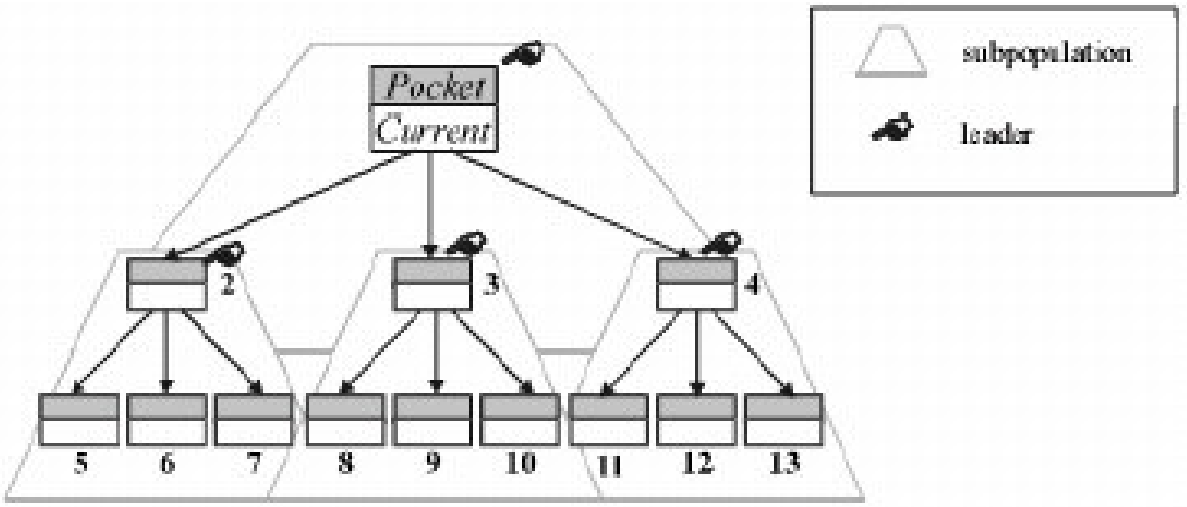


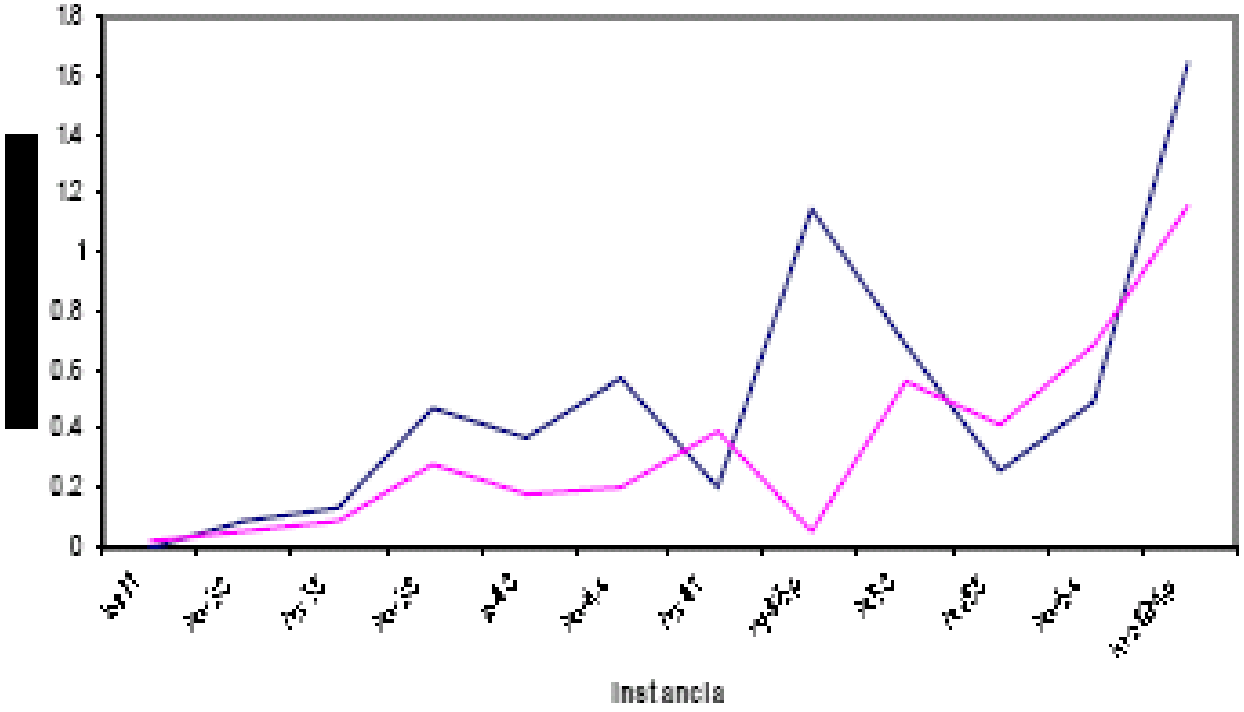
Genético x Memético

```
Inicializar população P;
Repita
    Selecione uma subpopulação P';
    Para i := 1 até nr_cruzamento faça
        Escolha S1,S2 ∈ P' – aleatoriamente;
        Filho := cruzamento(S1,S2);
        Se f(S1) >= f(S2)
            então Saux := S1;
            senão Saux := S2;
        Se Saux > f(Filho)
            então Filho := buscalLocal(Filho)
                Filho substitui Saux em P;
        FimSe
    FimPara
     Para i := 1 até nr_mutacoes faça
        Selecione um cromossomo Sj em P;
         Sj := mutação(Sj );
         Sj := buscaLocal(Sj );
     FimPara
Até que critério_parada seja satisfeito;
```

Aplicações

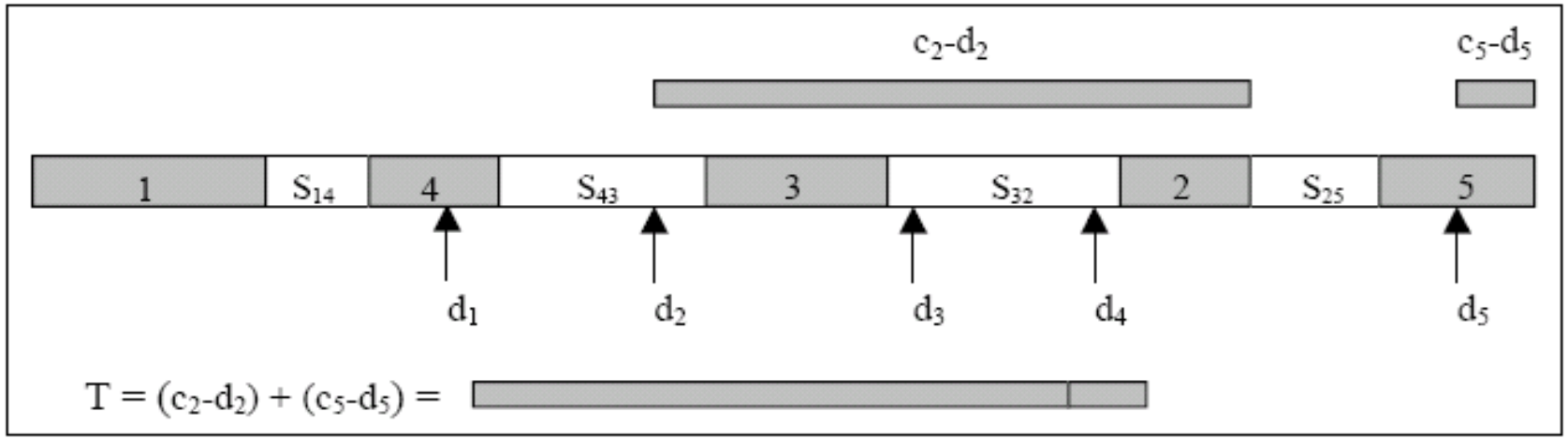
Caixeiro Viajante [4]



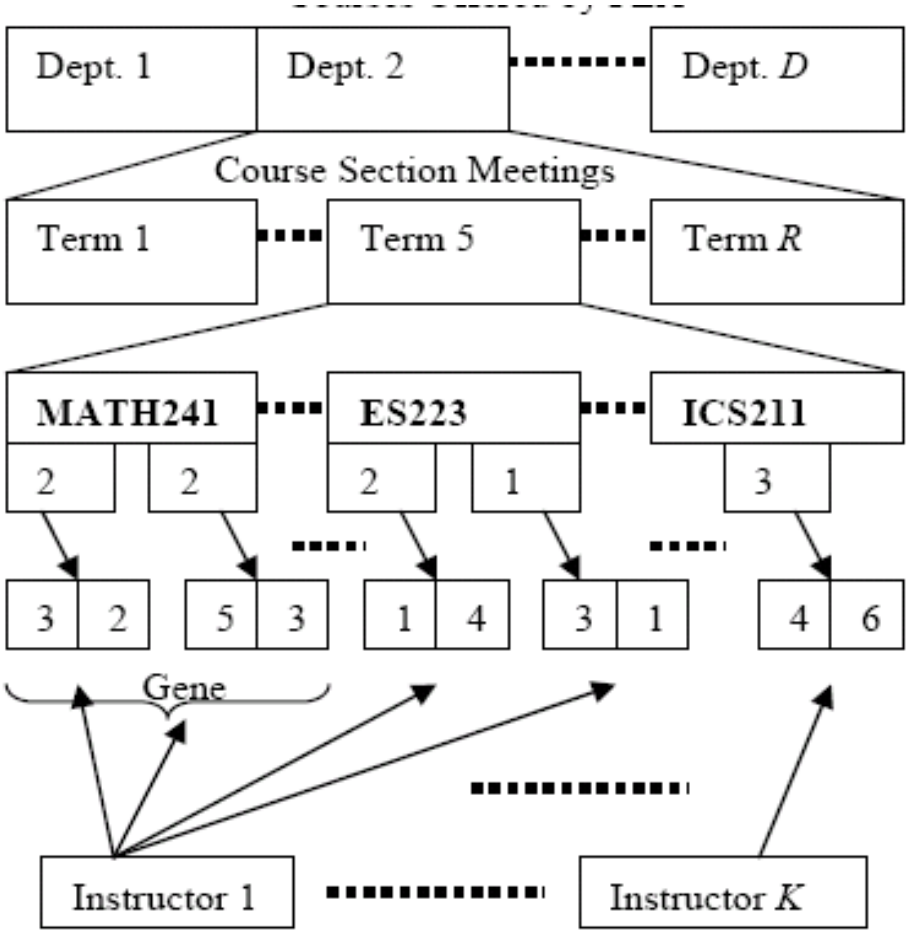


Seqüênciamento em Máquina Simples[??]

- Dado *n* tarefas a serem processadas em uma máquina, uma lista de tempos de processamento ($t_1...t_n$) e outra com datas de entrega ($d_1...d_n$) e uma matriz com tempos de preparação onde s_{ij} é o tempo de preparação da tarefa j depois da máquina ter processado a tarefa i.
- Algoritmo Memético Paralelo



Timetabling [8]



Tupla: (V,D,C)
V: Cursos
D: variáveis de domínio
C: restrições

Conclusões

- Evolução do Algoritmo Genético
- Variações de fluxograma
- Maior esforço computacional que Algoritmos Genéticos
- Fornecem melhores resultados
- Qual método de busca local utilizar?
  - Tamanho da vizinhança (busca custosa)
    - Redução da vizinhança
- Técnicas paralelas (Algoritmos Evolutivos Paralelos Híbridos)

Referências Bibliografias

- [1] MOSCATO, P. **On Evolution, Search, Optimizaton, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms**, Tech. Rep. Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.
- [2] MOSCATO, P. & NORMAN, M. G. **A memetic approach for the travelling salesman problem– implementation of a computational ecology for optimisation on message-passing systems,** Proceedings of the International Conference on Parallel Computing and Transputer Applications, Amsterdam, IOS Press, 1992.
- [3] MOSCATO, P. **Memetic Algorithms: A Short Introduction,** in Corne, D., Dorigo, M. & Glover, F. (eds.) New Ideas in Optimization, McGraw-Hill, pp. 219-234, 1999.
- [4] L. Buriol, P. M. França, P. Moscato, "**A New Memetic Algorithm for the Asymmetric Traveling Salesman Problem**". *Journals of Hueristics*. Volume 10, Issue 5 (Septiembre 2004) pp. 483-506.
- [5] **Memetic Algorithms Home Page** Disponível em: http://www.densis.fee.unicamp.br/~moscato/memetic_home.html. Acessado em 11/04/2007.
- [6] GOLDBARG, Marco César; GOLDBARG, Elizabeth Ferreira Gouvêa; MEDEIROS NETO, Francisco Dantas de. **Algoritmos evolucionários na determinação da configuração de custo mínimo de sistemas de co-geração de energia com base no gás natural. Pesqui. Oper.**, Rio de Janeiro, v. 25, n. 2, 2005. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382005000200005&lng=en&nrm=iso>. Acesso em: 11 Apr 2007. Pré-publicação. doi: 10.1590/S0101-74382005000200005
- [7] NORONHA, Thiago Ferreira de ; SILVA, Marcelo Mariano da ; ALOISE, D. J. **Um Algoritmo Memético de Grupamento para o Problema de Bin Packing 1-D**. REIC. Revista eletrônica de iniciação científica, http://www.sbc.org.br/reic/edi, v. I, n. II, p. 1-15, 2001.
- [8]. BURKE, E., NEWALL, J.P. and WEARE, R. F, **A Memetic Algorithm for University Exam Timetabling.** in Burke. E.K. and Ross, P.(eds), The Practice and Theory of Automated Timetabling., p.241-250, 1996.
- [9]. PAECHTER, B. CUMMING, A., NORMAL, M. G. and LUCHIAN, H. **Extensions to a Memetic Timetabling system.** In Burke, E. K. and Ross, P., The Practice and Theory of Automated Timetabling, p.251-265, 1996.
- [10] Tese de Mestrado Concilio, Ricardo **Contribuições à solução de problemas de escalonamento pela aplicação conjunta de computação evolutiva e otimização com restrições** / Ricardo Concilio.--Campinas, SP: [s.n.], 2000. Disponível em: http://www.dca.fee.unicamp.br/~vonzuben/research/rico_mest.html
- [11] Houck, C.R.; Joines, J.A.; Kay, M.G. & Wilson, J.R. (1997). **Empirical Investigation of Benefits of Partial Lamarckianism.** Technical Report V-10, Department of Industrial Engineering, North Carolina State University.
- [12] AREIBI, S.; MOUSSA, M.; ABDULLAH, H. **A Comparison of Genetic/Memetic Algorithms and Heuristic Searching**. International Conference on Artificial Intelligence IC-AI 2001, Las Vegas, Nevada, June 25, 2001.
- [13] GARCIA J. V.; MENDES, A. S'.; FRANÇA P. M.; MOSCATO, P. A;. Algoritmo Memético Paralelo Aplicado a Problemas de Sequenciamento em Máquina Simples. Departamento de Engenharia de Sistemas. Universidade Estadual de Campinas.

Algoritmos Meméticos

Universidade Federal do Paraná
Tópicos em Inteligência Artificial
Dra. Aurora Pozo

Alunos:
Ademir Roberto Freddo – *freddo@utfpr.edu.br*
Robison Cris Brito – *robison@utfpr.edu.br*