Universidade Federal do Paraná

Tópicos em Inteligência Artificial

Drª Aurora Pozo

GRASP

Greedy Randomized Adaptive Search Procedures

Ademir Roberto Freddo

Robison Cris Brito

Roteiro

- Introdução
- GRASP
- Implementação
- Tecnologias Utilizadas
- Experimentos
- Resultados Obtidos
- Conclusões

Introdução

Objetivo:

"Implementação da Metaheurística GRASP para o problema do Caixeiro Viajante"

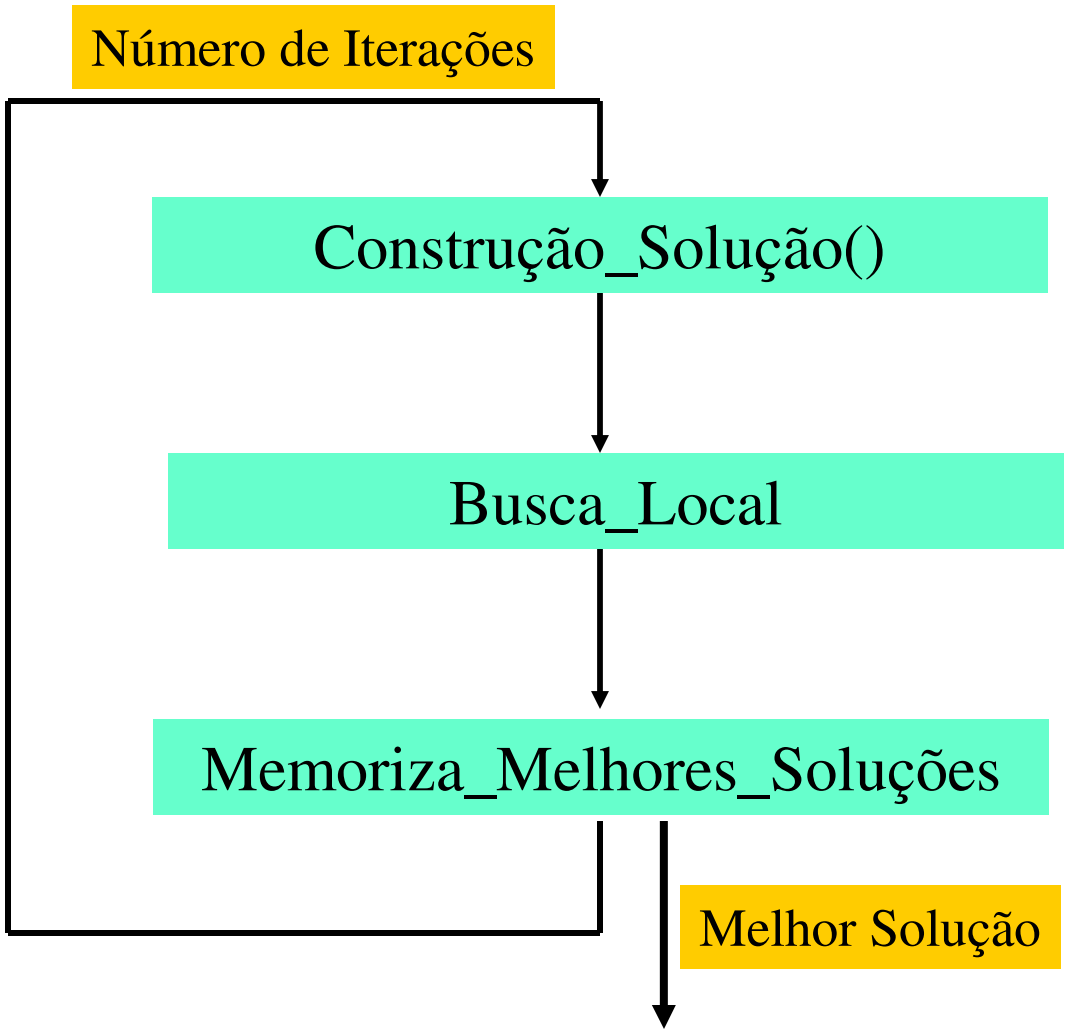
Motivação:

- Heurísticas;
- Metaheurísticas;
- Hibridização;
- Aplicações para o problema do Caixeiro Viajante;

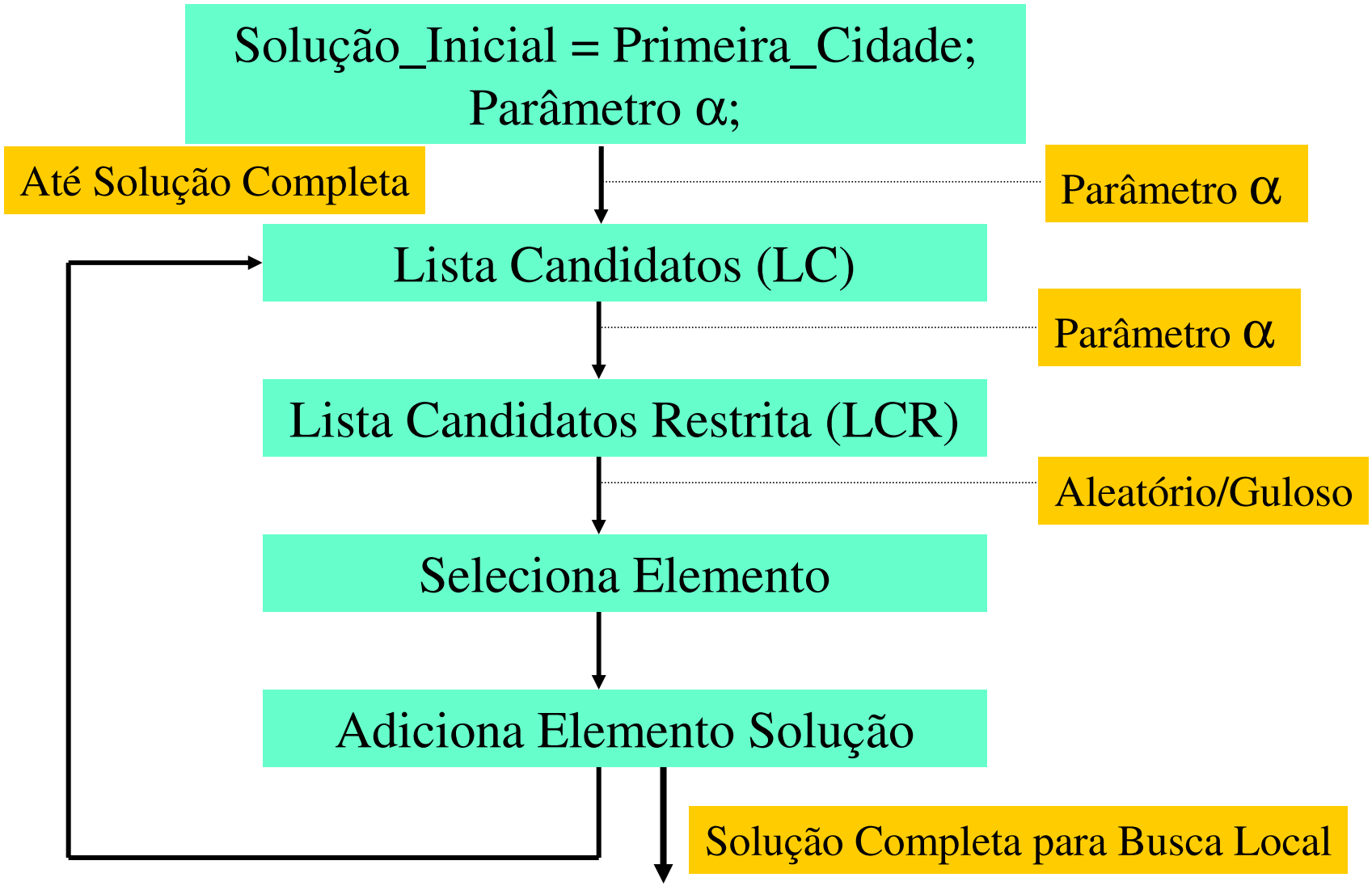
GRASP

- Metaheurística para geração rápida de soluções;
- Thomas A. Feo e Mauricio G. C. Resende (1989).
- Duas Fases:
  - Heurística Construtiva : Geração Gulosa, Randômica e Adaptativa;
  - Busca Local: Busca na solução encontrada.

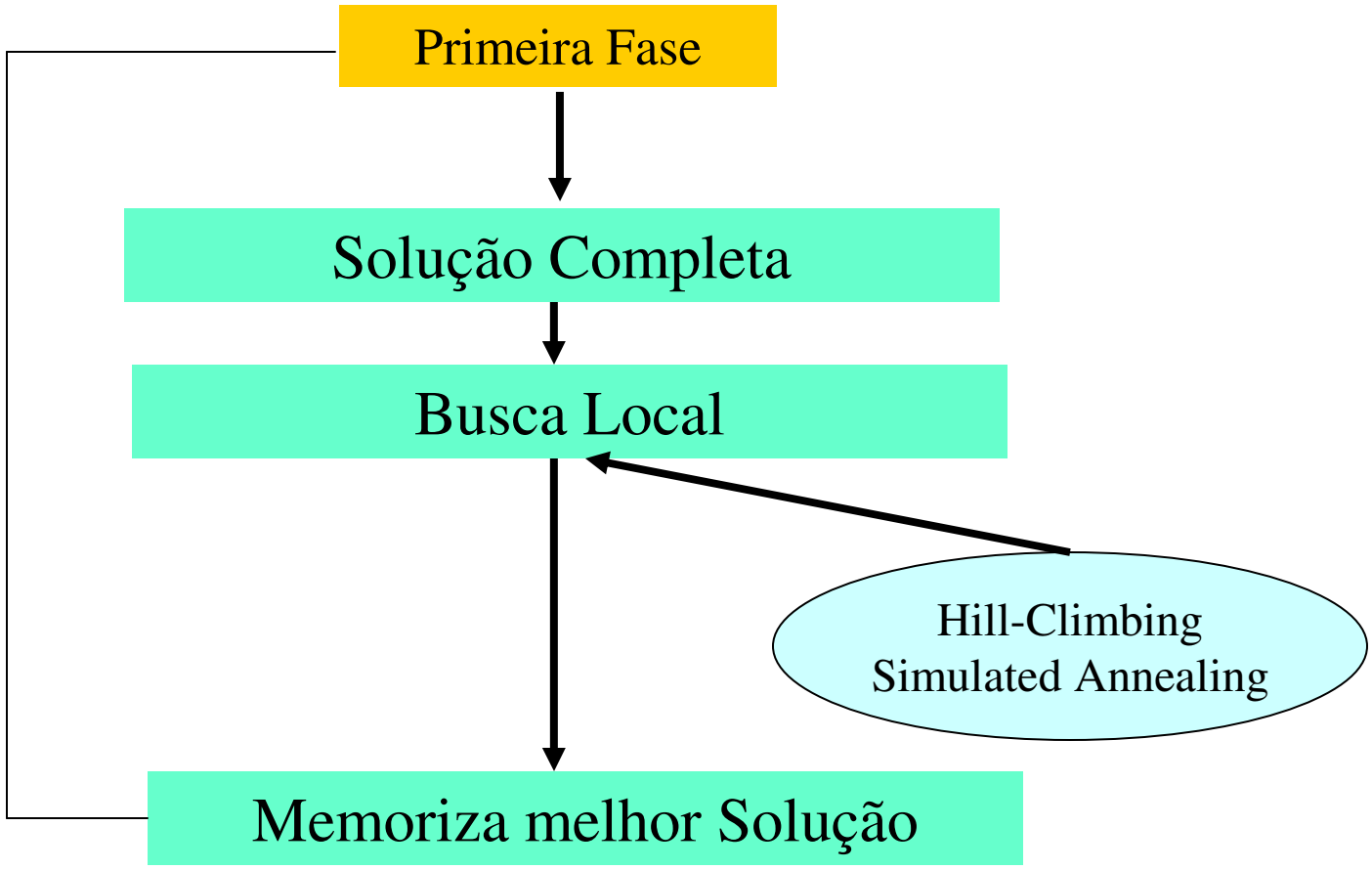
Algoritmo GRASP



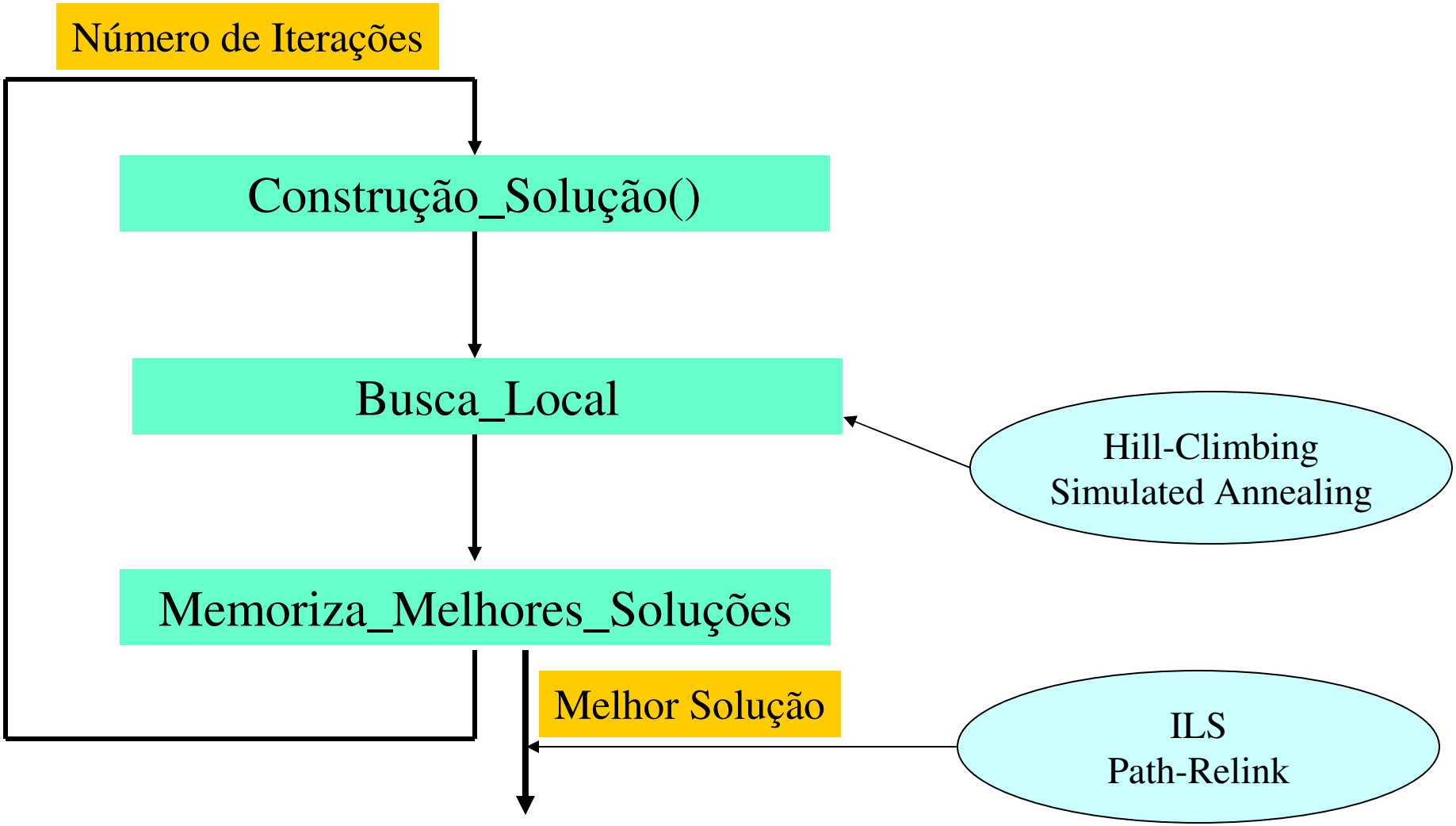
Fase: Construção



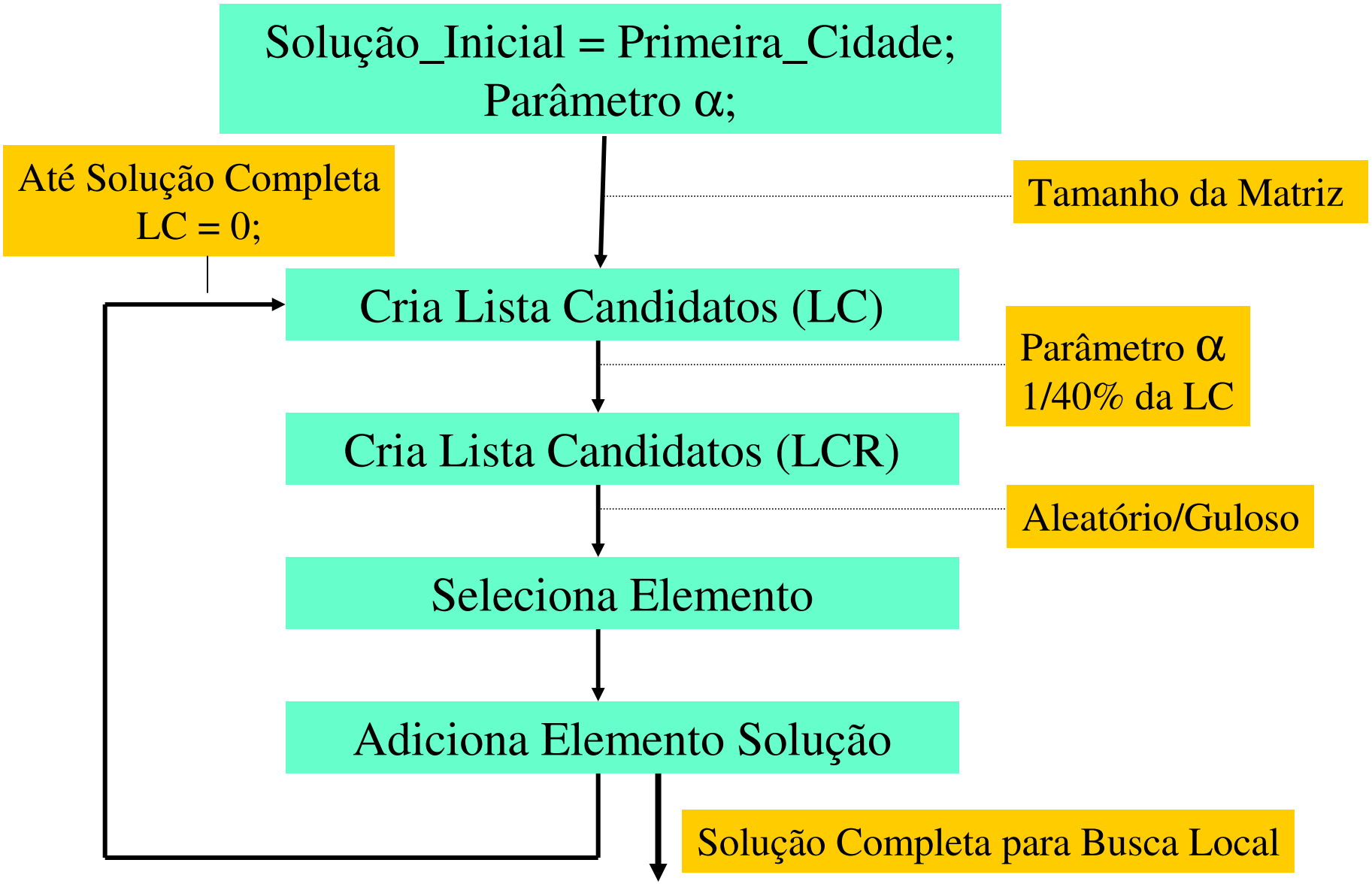
Fase: Busca Local



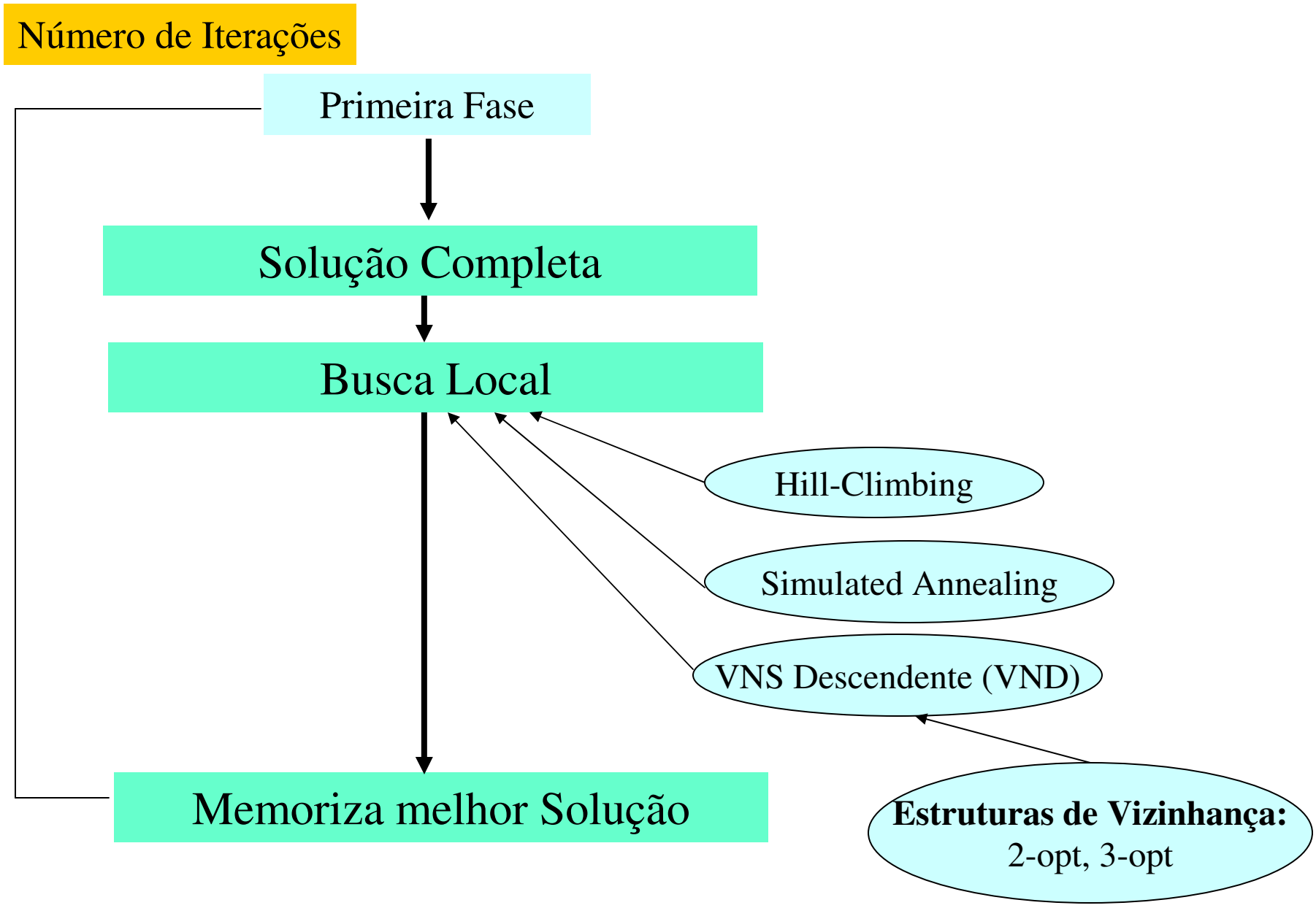
Algoritmo GRASP



Implementação - Fase: Construção



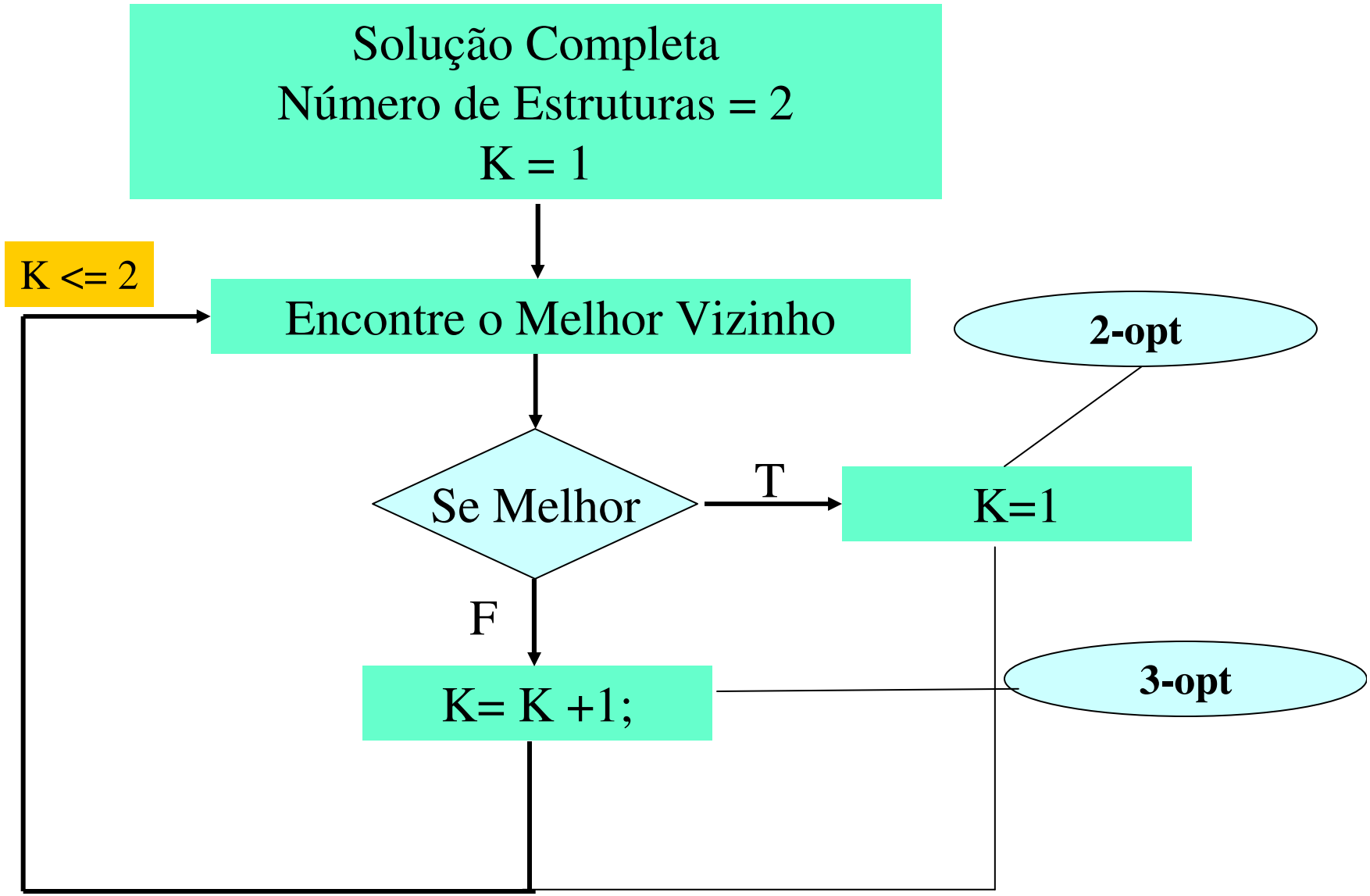
Fase: Busca Local



Tecnologias Utilizadas

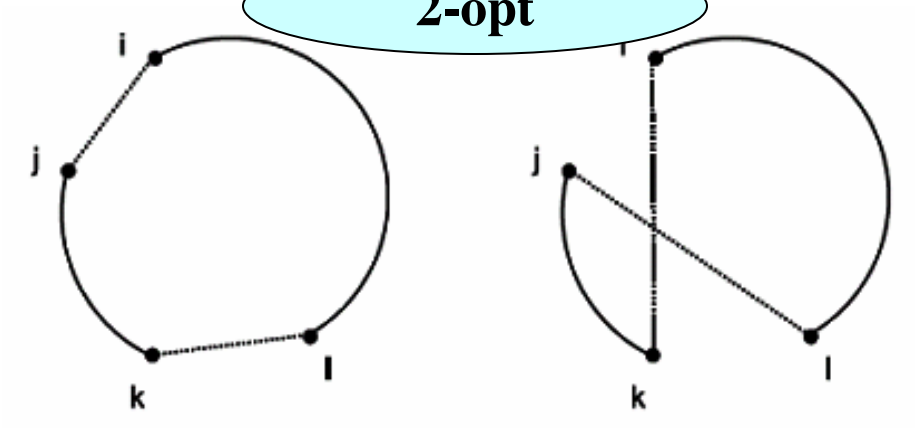
- Hill-Climbing
  - Busca Local com troca entre vizinhos;
- SA
  - Temperatura Inicial: 30;
  - Temperatura Final: 0.1;
  - Número de Iterações: 60% da Matriz;
  - Taxa de Redução de Temperatura: 0.2%

Tecnologias Utilizadas: VND

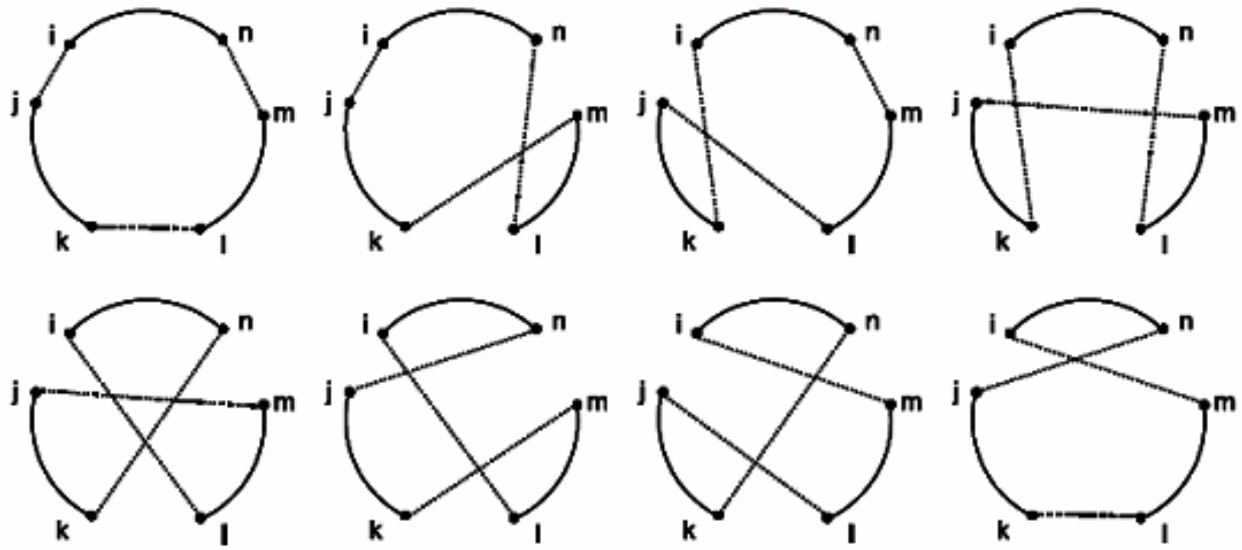


Estruturas de Vizinhança (VND)

2-opt



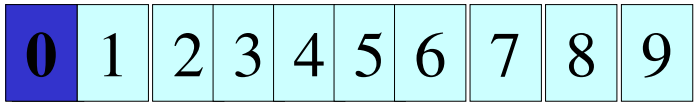
3-opt



Experimentos

- Instâncias: dantzig42, fri26, gr48, hk48
- Hibridizações:
  - GRASP/HC
  - GRASP/SA
  - GRASP/VND
- Para cada hibridização:
  - Aleatório, Puramente Guloso (valor de alfa);
- Número de execuções:
  - 10 execuções para cada instância;
  - Total: 10 x 4 instâncias x 3 hibridizações x 2 alfa = 240

Construção da Lista

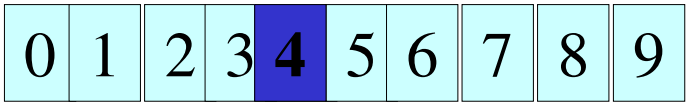


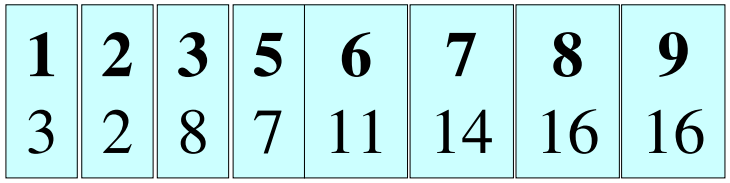
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
10	9	8	1	2	14	16	20	30

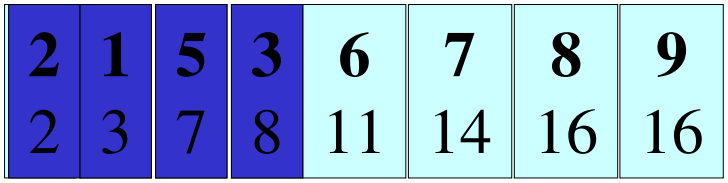
4	**5**	**3**	**2**	**1**	**6**	**7**	**8**	**9**
1	2	8	9	10	14	16	20	30

Solução: 0, 4, ...

Construção da Lista







Solução: 0, 4, 2, ...

Construção da Lista

1	**3**	**5**	**6**	**7**	**8**	**9**
3	8	7	11	14	16	16

.....

Construção da Lista

Solução: 1,.....

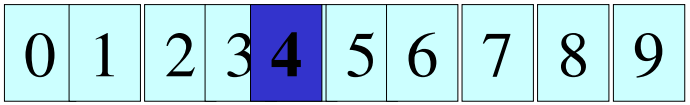
Construção da Lista

Solução: 2,.....

Construção da Lista

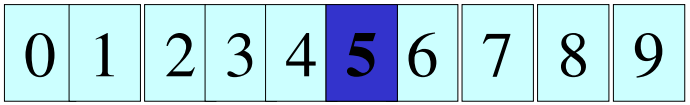
Solução: 3,.....

Construção da Lista



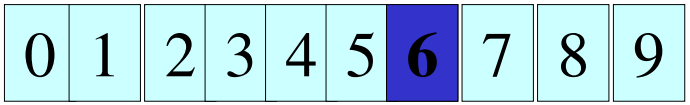
Solução: 4,.....

Construção da Lista



Solução: 5,.....

Construção da Lista



Solução: 6,.....

Algoritmo Busca Local

```
for( int i=0; i < tamanhoMatriz; i++) {
    int estadoInicial[] = CoreGrasp.gerarEstadoInicialGuloso( i, distanciaEntre, alfa
    int estadoAposHC[] = CoreGrasp.executarHillClimbing( estadoInicial,distanciaEntre

    int custo = CoreGrasp.calculaDistancia( distanciaEntre, estadoAposHC );

    if ( custo < melhorRota.getCusto() ) {
        melhorRota.setEstado( estadoAposHC.clone() );
        melhorRota.setCusto( custo );
    }
```

Algoritmo Simuated Annealing

```
for( int i=0; i < tamanhoMatriz; i++) {
    int estadoInicial[] = CoreGrasp.gerarEstadoInicialGuloso( i, distanciaEntre,alfa
    int estadoAposHC[] = CoreGrasp.executarSA( estadoInicial, distanciaEntre,
                CoreGrasp.temperaturaInicial, CoreGrasp.temperaturaFinal,
                CoreGrasp.taxaResfriamentoTemperatura,
                CoreGrasp.numeroIteracoesTemperatura );

    int custo = CoreGrasp.calculaDistancia( distanciaEntre, estadoAposHC );

    if ( custo < melhorRota.getCusto() ) {
        melhorRota.setEstado( estadoAposHC.clone() );
        melhorRota.setCusto( custo );
    }
```

Algoritmo VND/VNS

```
for( int i=0; i < tamanhoMatriz; i++) {
   int melhorEstado[] = CoreGrasp.gerarEstadoInicialGuloso( i, distanciaEntre, alfa
   int melhorCusto = CoreGrasp.calculaDistancia( distanciaEntre, melhorEstado );

   int k = 1;

   while( k <= 2 ) {

       if ( k == 1 ) {
           estadoAposOpt = CoreGrasp.melhor2Opt( melhorEstado, distanciaEntre );
       } else if ( k ==2 ) {
           estadoAposOpt = CoreGrasp.melhor3Opt( melhorEstado, distanciaEntre );
       }

       custoAposOpt = CoreGrasp.calculaDistancia( distanciaEntre, estadoAposOpt )

       if ( custoAposOpt < melhorCusto ) {
           melhorCusto = custoAposOpt;
           melhorEstado = estadoAposOpt.clone();
           k=1;
       } else {
           k++;
       }
   }
```

Resultados – GRASP Guloso

Melhores Resultados

Instâncias	Melhor Custo	GRASP/HC	GRASP/SA	GRASP/VND
Dantzig 42	**699**	863	863	**699**
fri26	**937**	955	955	**937**
gr48	**5046**	5781	5779	5093
hk48	**11461**	12110	12006	11470

Média

Instâncias	Melhor Custo	GRASP/HC	GRASP/SA	GRASP/VND
dantzig42	**699**	918	916	710
fri26	**937**	1057	1059	954
gr48	**5046**	6228	6211	5147
hk48	**11461**	13263	12116	11571

Tempo

Instâncias	GRASP/HC	GRASP/SA	GRASP/VND
dantzig42	0,01	1,94	1,42
fri26	0,001	2,65	0,10
gr48	0,01	12,52	2,85
hk48	0,02	36,40	1,60

Resultados – GRASP Aleatório

Melhores Resultados

Instâncias	Melhor Custo	GRASP/HC	GRASP/SA	GRASP/VND
Dantzig 42	**699**	1027	938	699
fri26	**937**	1345	1341	937
gr48	**5046**	7828	6712	5046
hk48	**11461**	17012	16504	11470

Média

Instâncias	Melhor Custo	GRASP/HC	GRASP/SA	GRASP/VND
dantzig42	**699**	1919	1892	714
fri26	**937**	1765	1862	958
gr48	**5046**	13771	11920	5151
hk48	**11461**	30164	28658	11785

Tempo

Instâncias	GRASP/HC	GRASP/SA	GRASP/VND
Dantzig 42	0,001	3,30	2,94
fri26	0,001	1,81	0,17
gr48	0,01	32,60	6,24
hk48	0,01	52,34	5,34

Resultados - Comparação

	GRASP/HC			GRASP/SA			GRASP/VND		
	Custo	**%**	**GRASP**	**Custo**	**%**	**GRASP**	**Custo**	**%**	**GRASP**
Dantzig 42	863	23,46%	G	863	23,46%	G	**699**	0,00%	G
fri26	955	1,92%	G	955	1,92%	G	**937**	0,00%	G
gr48	5781	14,57%	G	5779	14,53%	G	**5046**	0,00%	A
hk48	12110	5,66%	G	12006	4,76%	G	11470	0,08%	G/A

Comparação utilizando os melhores valores obtidos

	GRASP/HC			GRASP/SA			GRASP/VND		
	Custo	**%**	**GRASP**	**Custo**	**%**	**GRASP**	**Custo**	**%**	**GRASP**
Dantzig 42	918	31,33%	G	916	31,04%	G	710	1,57%	G
fri26	1057	12,81%	G	1059	13,02%	G	954	1,81%	G
gr48	6228	23,42%	G	6211	23,09%	G	5147	2,00%	G
hk48	13263	15,72%	G	12116	5,72%	G	11571	0,96%	G

Comparação utilizando médias de execução

Conclusões

- Facilidade de Hibridização do GRASP
  - Diversificação/Intensificação
- Características do GRASP
  - Amostragem Rápida
- Importância da Primeira Fase do GRASP
- Fase Gulosa – Melhores Resultados
- GRASP/HC melhores resultados do que GRASP/SA com tempo inferior
- GRASP/VND melhores resultados com complexidade computacional baixa
- GRASP/VND escolha das estruturas de vizinhança adequadas

Universidade Federal do Paraná

Tópicos em Inteligência Artificial

Drª Aurora Pozo

GRASP

Greedy Randomized Adaptive Search Procedures

Ademir Roberto Freddo

Robison Cris Brito