

Meta-heurísticas

Inteligência Artificial

Prof. Giancarlo Guizzardi

Prof. Flávio Varejão

Programa de Pós-Graduação em Informática

Universidade Federal do Espírito Santo

Agenda

- Motivação
- Heurísticas
- MetaHeurísticas
 - Baseadas em Soluções Parciais
 - Baseadas em Soluções Completas
 - Busca Local
 - Busca Populacional
 - Algoritmos Genéticos
- Aplicações

Motivação

Motivação

- Problemas NP
 - Bem definidos
 - Descritor de espaço de estados e operadores conhecidos
 - Função objetivo
 - Não é suave
 - Possui ótimos locais
 - Algoritmo conhecido resolve por enumeração
 - Intratável computacionalmente
- Soluções aproximadas são suficientes

Heurísticas

Heurísticas

- Procedimentos de resolução
 - Específicos para o problema a ser resolvido
 - Acham solução, mas não garantem que seja ótima
 - Normalmente baseados na experiência
- Exemplo
 - Caixeiro Viajante
 - A partir da primeira cidade, vá para a cidade mais próxima ainda não visitada até que todas as cidades tenham sido visitadas. Então, retorne a primeira cidade.

Heurísticas

□ Problema

- Desenvolver heurísticas específicas para problema a ser resolvido pode ser difícil e/ou trabalhoso

Meta-heurísticas

Meta-heurísticas

- Procedimentos heurísticos gerais aplicáveis a um variedade de problemas
 - Tipicamente inclui características para fazer com que o procedimento de busca fuja de ótimos locais
- Definem infraestrutura para criação de procedimentos heurísticos
- Duas categorias principais
 - Baseadas em Soluções Parciais
 - Baseadas em Soluções Completas

Baseadas em Soluções Parciais

Baseadas em Soluções Parciais

- Construção de Solução
 - Busca realizada na medida que a solução é construída
- Procedimentos
 - Hill Climbing
 - A*
 - Beam Search
 - Branch and Bound

Hill Climbing

- Estratégia gulosa
 - Decisões baseadas em informação local
 - Uso de função estimativa heurística para escolher o próximo elemento da solução
 - Não há retrocesso na busca

Hill Climbing

□ Algoritmo

procedimento *HillClimbing* ()

$sm \leftarrow \emptyset$

enquanto *sm não está completa* **faça**

Avaliar elementos ainda não selecionados

$r \leftarrow$ *elemento melhor avaliado*

$sm \leftarrow sm \cup r$

fim-enq

retorna *sm*

fim-proc

A*

□ Algoritmo

procedimento A^* ()

incluir na fila f todos os possíveis estados de partida ordenados pela função de avaliação

enquanto f não está vazia **faça**

Remover o primeiro s da fila f

se s for solução **então retorna** s

Avaliar os próximos estados de s

Incluí-los na fila ordenada

fim-enq

fim-proc

A*

- Função de avaliação

- Leva em conta a soma da

- Função objetivo do estado atual

- Função heurística

- Estimativa do custo para se chegar a solução

Beam Search

- Busca em amplitude
 - Uso de função objetivo para avaliação dos estados
 - Limita total de estados mantidos as m melhores expansões dos estados atuais

Beam Search

□ Algoritmo

procedimento *BeamSearch* (m)

*avaliar todos os possíveis estados iniciais e
manter os m melhores na fila f*

enquanto *solução não está completa* **faça**

Expandir todos os estados na fila f e avaliá-los

Atualizar f com m melhores estados

fim-enq

retorna *o estado de melhor avaliação*

fim-proc

Branch and Bound

- Requer (estimativa de) valor de função objetivo de uma solução (trivial)
- Dividido em duas etapas
 - Expansão de estados correntes
 - Poda de ramos que apresentam custo superior ao da solução atual

Branch and Bound

- Função de avaliação leva em conta
 - Função objetivo do estado atual
 - Função heurística
 - Estimativa do custo para se chegar a solução
 - Otimista
 - Avalia sempre igual ou inferior a custo real
 - Garante achar o ótimo
 - Não otimista
 - Pode perder ótimo

Branch and Bound

□ Algoritmo

procedimento BranchBound ()

obter custo de uma solução trivial

incluir na fila ordenada f todos os possíveis estados de partida com avaliação inferior ao custo da solução trivial

enquanto f não está vazia **faça**

Remover o primeiro s da fila f

se s for solução **então retorna** s

Avaliar os próximos estados de s

Incluí-los na fila ordenada se custo inferior ao da solução trivial

fim-enq

fim-proc

Baseadas em Soluções Completas

Baseadas em Soluções Completas

- Necessário ter soluções completas para o problema para aplicar a heurística
 - Vetor descritor de estado totalmente preenchido
- A partir das soluções completas são feitas modificações com o intuito de encontrar uma melhor (boa) solução
- Duas subcategorias
 - Busca Local
 - Busca Populacional

Busca Local

Busca Local

- Busca na vizinhança de uma ou mais soluções
- Fácil de compreender/aprender
- Obtém soluções rapidamente
- Pouco tempo de implementação
- Necessário pouco ou nenhum conhecimento do problema

Busca Local

□ Algoritmo Geral

1. Gerar solução inicial
2. Modificar levemente a solução
3. Avaliar nova solução
4. Repetir passos 2 e 3 até que nenhuma melhoria significativa é encontrada

Busca Local

- Procedimentos
 - Deepest descent
 - Multistart descent
 - Tabu search
 - Simulated annealing

Busca Local

- Necessárias funções para
 - Gerar solução inicial (s)
 - Determinar vizinhança ($V(s)$)
 - Função objetivo (f)
- Minimização de função objetivo

Deepest Descent

- Algoritmo

procedimento *Deepest_Descent* (s)

repetir

Escolher $s' \in V(s) / f(s') \leq f(s'') \forall s'' \in V(s)$

se $f(s') < f(s)$ **então**

$s \leftarrow s'$

fim-se

até $f(s') \geq f(s), \forall s' \in V(s)$

fim-proc

Multistart Descent

□ Algoritmo

procedimento *Multistart_Descent*

iter = 1

$f(sm) = +\infty$

repetir

Escolher s' aleatoriamente

$s \leftarrow \text{Deepest_Descent}(s')$

se $f(s) < f(sm)$ **então**

$sm \leftarrow s$

$f(sm) \leftarrow f(s)$

fim-se

iter = *iter* + 1

até *iter* = *iterMax*

fim-proc

Tabu Search

- Movimento para pior solução é permitido
- Previne volta a soluções já visitadas através da lista tabu
 - Evita ciclo na busca
- Lembra a melhor solução encontrada até o momento
- Critérios de parada
 - Tempo de cpu, número de iterações, número de iterações sem melhorias

Simulated Annealing

- Analogia com processo físico de resfriamento de sólido superaquecido
 - Na medida que esfriam, entram em estados de menor energia
 - Ao longo do processo pode entrar em estados com maior energia
 - Ocorre de modo aleatório
 - Menos frequente na medida que a temperatura se reduz

Simulated Annealing

- Busca não determinística
- Soluções melhores são sempre aceitas
- Movimento para pior solução é permitido
 - Quanto pior a solução, menor a chance de ser aceita

Busca Populacional

Busca Populacional

- Baseada numa população de soluções
- Soluções são combinadas para gerar nova população

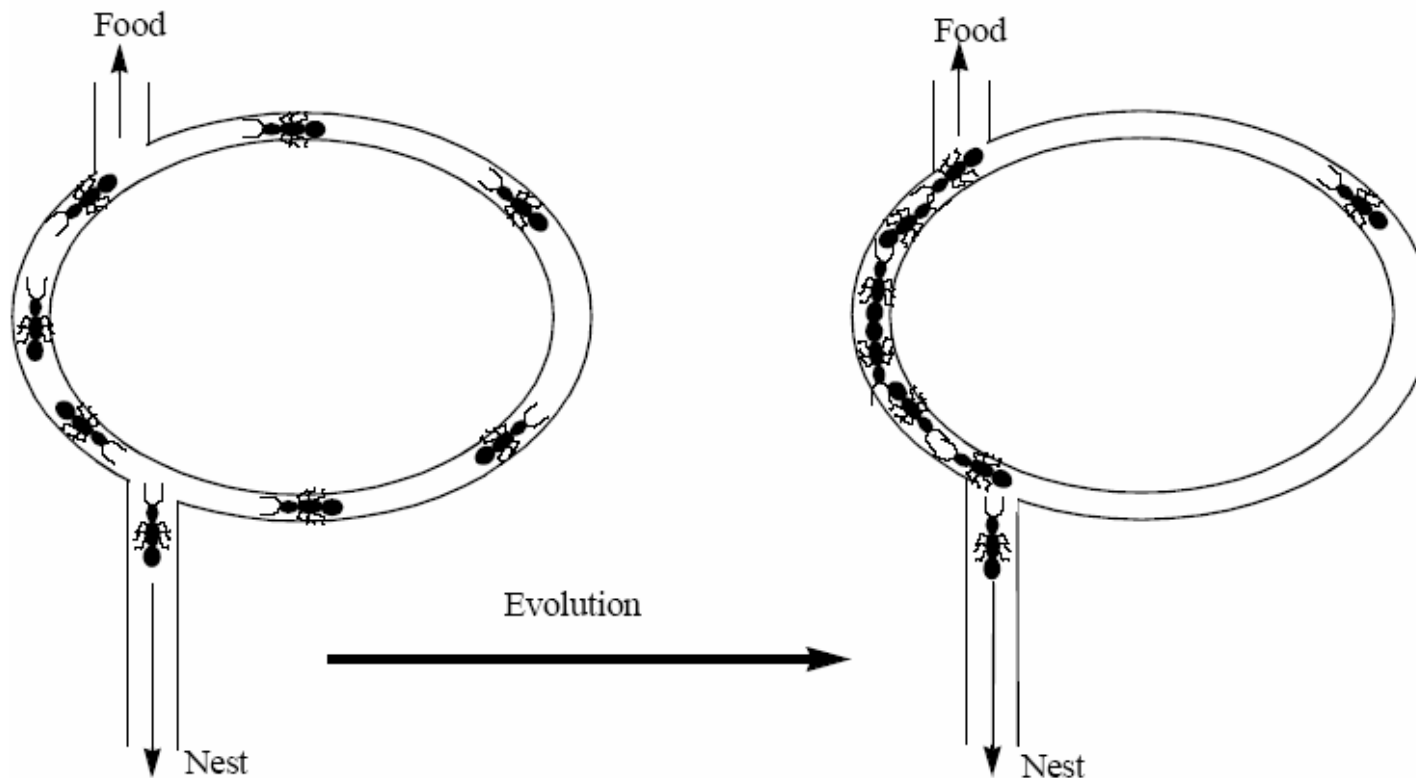
Busca Populacional

- Procedimentos
 - Colônia de Formigas
 - Algoritmos Genéticos

Colônia de Formigas

Colônia de Formigas

□ Experimento com formigas reais



Colônia de Formigas

- Experimento com formigas reais
 - Inicialmente metade vai pelo menor e metade vai pelo maior
 - Formigas que usam o menor caminho vão e voltam mais rapidamente
 - Ocorre maior depósito de feromônios no menor caminho
 - Ao final formigas usam sempre o menor caminho

Algoritmos Genéticos

Algoritmos Genéticos

- ❑ São programas evolutivos baseados na teoria de seleção natural e hereditariedade. Favorecem os candidatos mais promissores para a solução de um dado problema
- ❑ Desenvolvido por John Holland (1975) e seus alunos
- ❑ Popularizado por David Goldberg (1989)

Algoritmos Genéticos

□ Características

- Podem trabalhar com uma codificação do conjunto de parâmetros ou com os próprios parâmetros
- Trabalham com uma população e não com um único ponto
- Utilizam informações de custo ou recompensa
- Utilizam regras de transição estocásticas e não determinísticas

Algoritmos Genéticos

□ Características

- São fáceis de implementar em computadores
- Adaptam-se bem a computadores paralelos
- São facilmente combinados com outras técnicas
- Funcionam com parâmetros contínuos ou discretos

Algoritmos Genéticos

- Classe de procedimentos com um conjunto de passos distintos e bem especificados, no qual cada um destes passos possui muitas possíveis variações

Representação

- ❑ Os parâmetros do problema são representados como genes em um cromossomo
- ❑ Cada gene pode assumir valores específicos, sendo cada um destes valores chamados de alelo do gene
- ❑ Um cromossomo representa um indivíduo, sendo composto por uma configuração de alelos
- ❑ A posição de um gene num cromossomo corresponde a um locus gênico

Representação

Evolução Natural Problema Computacional

Indivíduo

Solução de um problema

População

Conjunto de soluções

Cromossomo

Representação de uma solução

Gene

Parte da representação de uma solução

Crossover

Operador de Busca

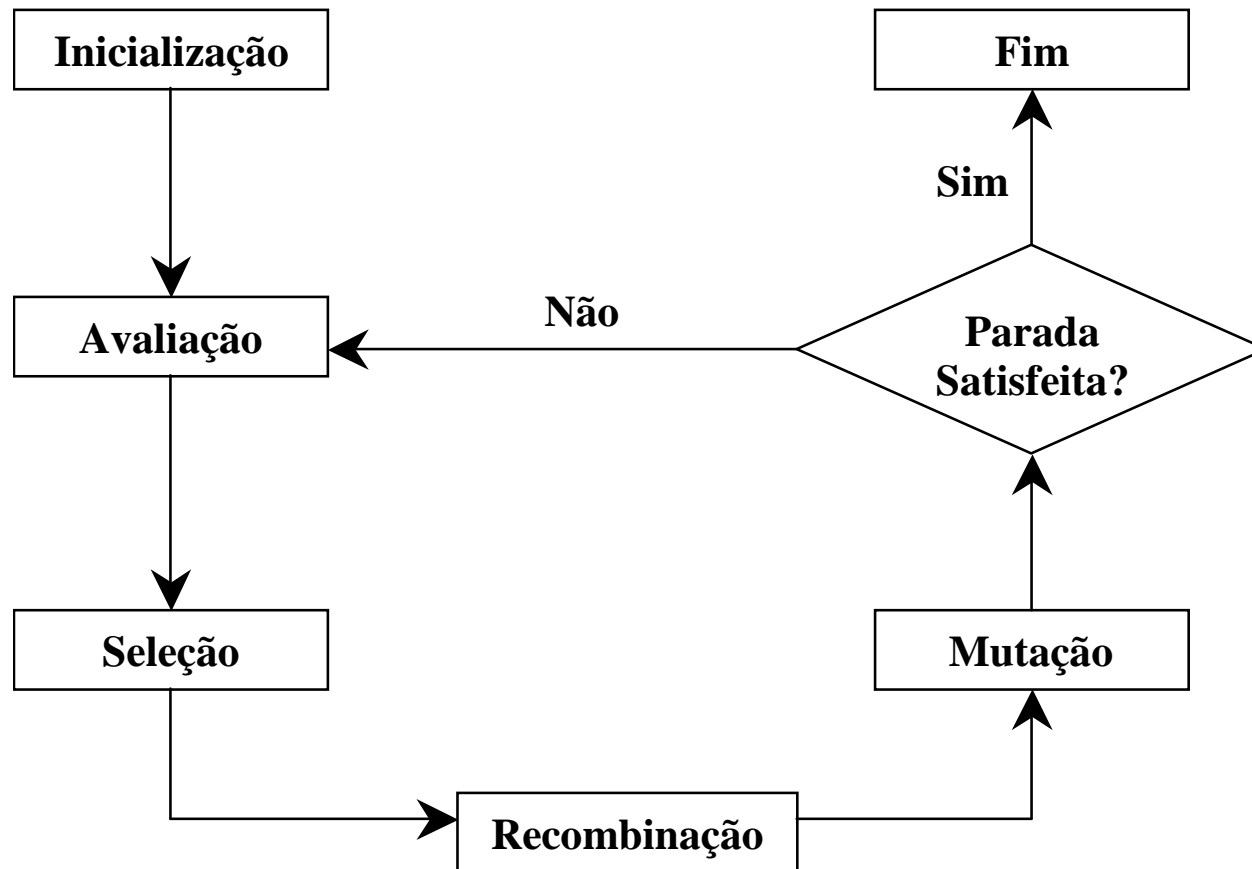
Mutação

Operador de Busca

Seleção natural

Reutilização de boas aproximações

Estrutura Básica



Genético

□ Algoritmo

procedimento *Genético()*

gerar população inicial e avaliar

enquanto *não satisfaz critério de parada* ***faça***

selecionar os mais aptos

gerar novos através de crossover e mutação

repor inviáveis

avaliar nova população

fim-enq

retorna *melhor*

fim-proc

Seleção

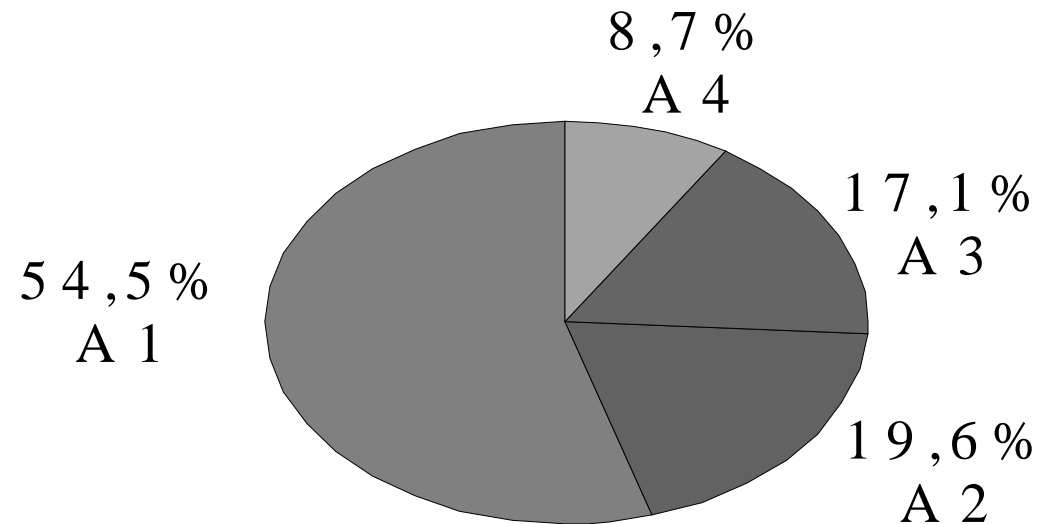
- O princípio básico do funcionamento dos AGs é que um critério de seleção vai fazer com que, depois de muitas gerações, o conjunto inicial de indivíduos gere indivíduos mais aptos

Seleção

- Uso de função objetivo como avaliação de aptidão
 - A aptidão pode ser vista como uma nota que mede o quão boa é a solução codificada por um indivíduo
 - Normalmente baseada no valor da função-objetivo, específica para cada problema
- Métodos de Seleção
 - Roleta
 - Torneio
 - Amostragem Universal Estocástica

Método da Roleta

- Aptidão usada para definir fatia
- Valor aleatório para selecionar cromossomo
- Processo repetido até gerar os n indivíduos necessários



Método do Torneio

- ❑ Escolha aleatória de m indivíduos
- ❑ Uso de função de aptidão para escolher o melhor
- ❑ Processo repetido até gerar os n indivíduos necessários

Método da Amostragem

- ❑ Método da roleta com n agulhas igualmente espaçadas
- ❑ Roleta é girada uma única vez

Operadores Genéticos

- Cruzamento
 - Cruzamento de pais para gerar dois filhos
 - Taxa de crossover
 - Tipos
 - Ponto Único
 - Dois Pontos
 - Multiponto
- Mutação

Cruzamento

□ Ponto Único

Pais	{	1 1 0 0 1		0 1
		0 1 1 1 1		
Filhos	{	1 1 0 1 1		
		0 1 1 0 1		

Cruzamento

□ Dois Pontos

<i>p a i</i> ₁	0 1 0	0 1 1 0 0 0	1 0 1 0 1 1
<i>p a i</i> ₂	0 0 1	0 0 1 1 1 0	0 0 1 1 0 1
<i>filh o</i> ₁	0 1 0	0 0 1 1 1 0	1 0 1 0 1 1
<i>filh o</i> ₂	0 0 1	0 1 1 0 0 0	0 0 1 1 0 1

Cruzamento

□ Multipontos

<i>p a i</i> ₁	1	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1
<i>p a i</i> ₂	0	0	1	0	0	1	1	1	0	0	1	1	0	1	1	1	0	0	0
<i>filh o</i> ₁	1	0	1	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	1
<i>fillh o</i> ₂	0	0	1	0	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0

Mutação

- Mudança aleatória de alelo
- Taxa de mutação
 - Significativamente inferior a de cruzamento

Antes da mutação	0 1 1 0 1
Depois	0 <u>0</u> 1 0 1

Parâmetros Genéticos

- Tamanho da população
- Taxa de cruzamento
- Taxa de mutação
- Intervalo de geração
 - Percentual de renovação da população

Parâmetros Genéticos

- Critério de parada
 - Número de gerações
 - Convergência da função de aptidão na população
 - Não melhoria da aptidão do melhor indivíduo após um número de gerações

Exemplo

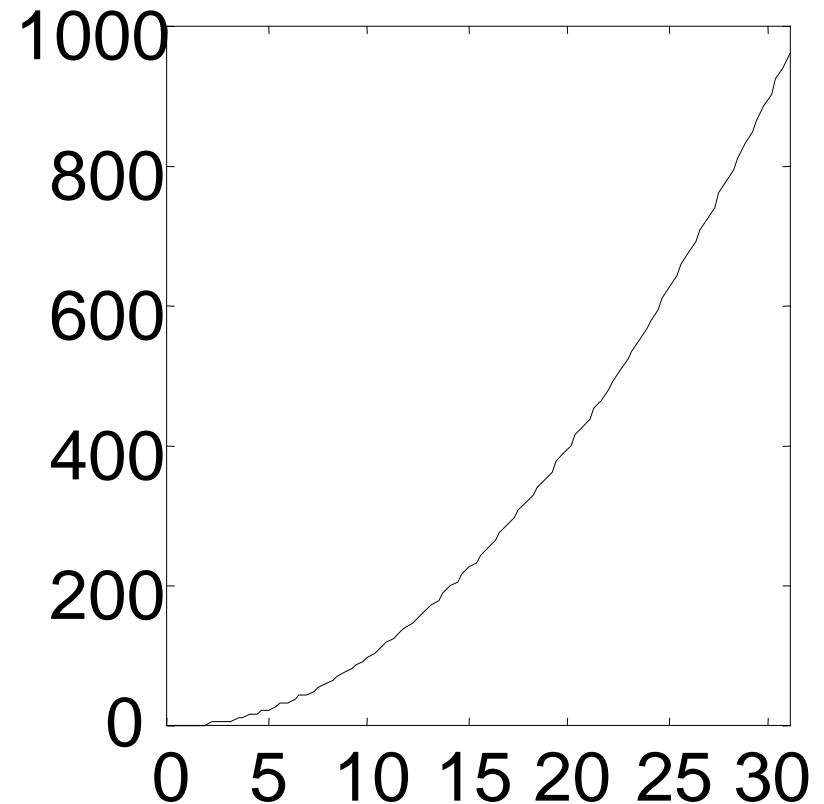
Problema: Use um AG para encontrar o ponto máximo da função para encontrar o ponto máximo da função

$$f(x) = x^2$$

com x sujeito as seguintes restrições:

$$0 \leq x \leq 31$$

x é inteiro



Exemplo

□ Cromossomos binários com 5 bits

□ 0 = 00000

□ 31 = 11111

□ Aptidão

■ Neste problema a aptidão pode ser a própria função objetivo

■ Exemplo

□ $\text{Aptidão}(00011) = f(3) = 9$

Exemplo

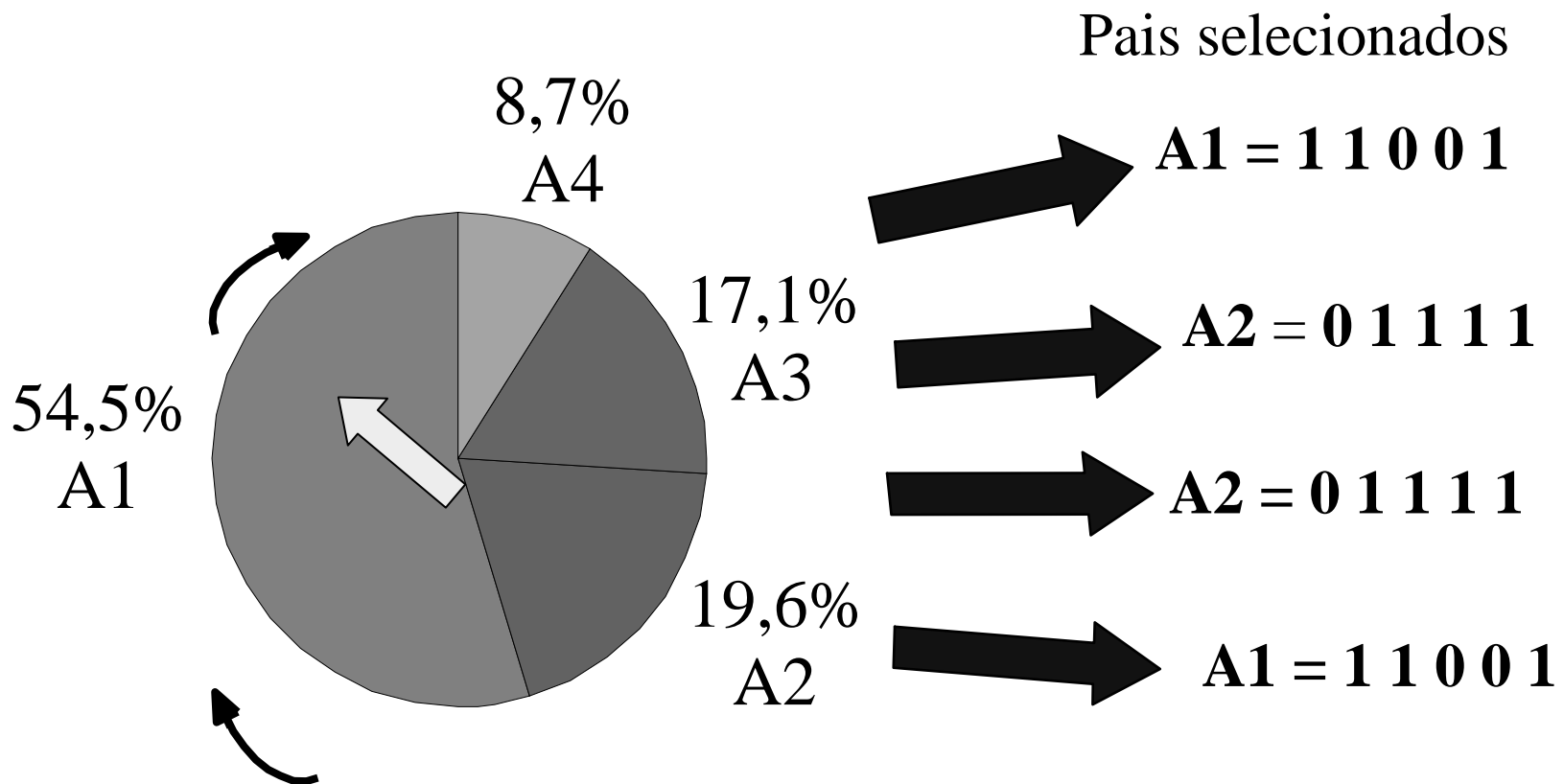
É aleatória (mas quando possível, o conhecimento da aplicação pode ser utilizado para definir população inicial)

	cromossomos	x	$f(x)$	Prob. de seleção
Pop. inicial	$A_1 = 1\ 1\ 0\ 0\ 1$	25	625	54,5%
	$A_2 = 0\ 1\ 1\ 1\ 1$	15	225	19,6%
	$A_3 = 0\ 1\ 1\ 1\ 0$	14	196	17,1%
	$A_4 = 0\ 1\ 0\ 1\ 0$	10	100	8,7%

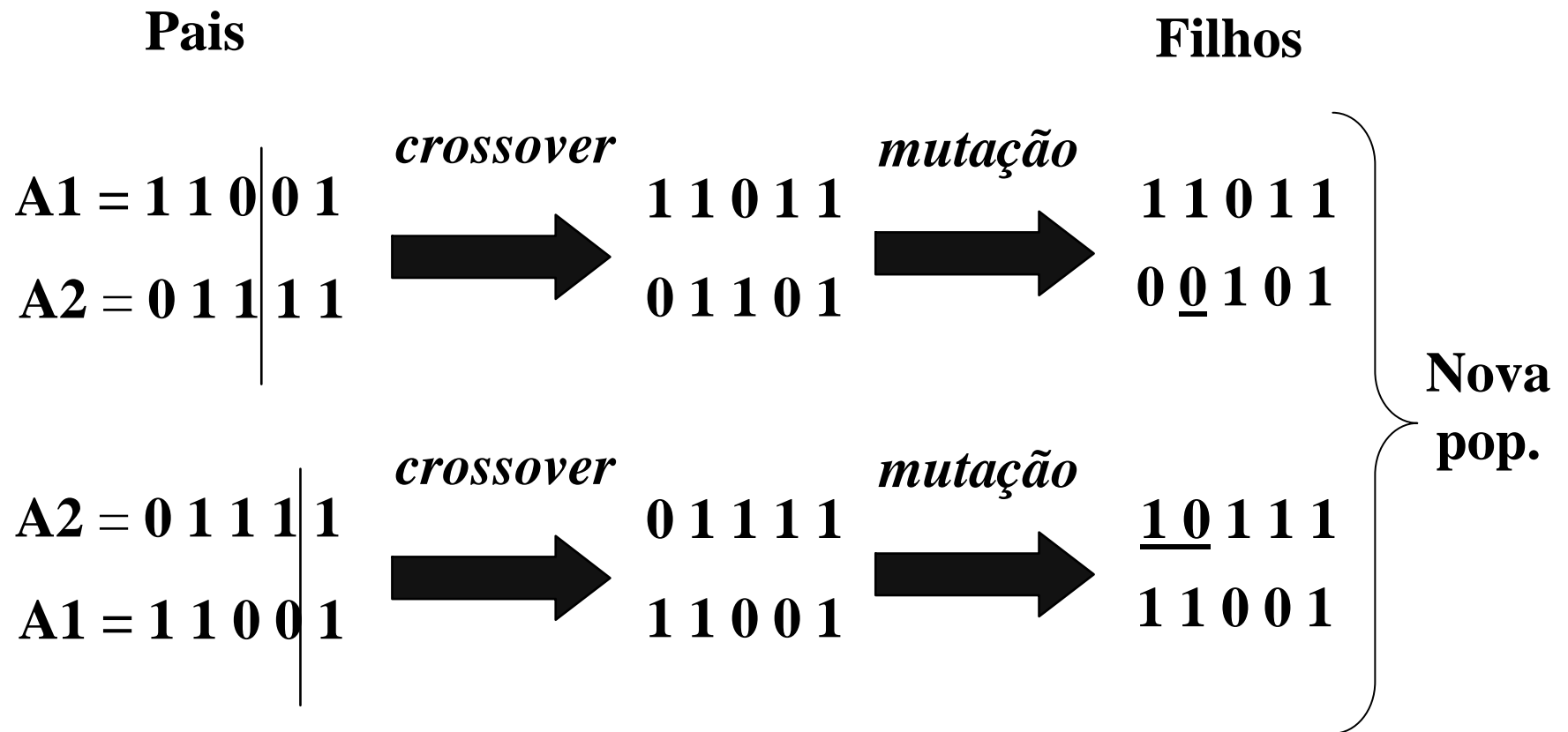
**Probabilidade de seleção
proporcional a aptidão**

$$p_i = \frac{f(x_i)}{\sum_{k=1}^N f(x_k)}$$

Exemplo



Exemplo



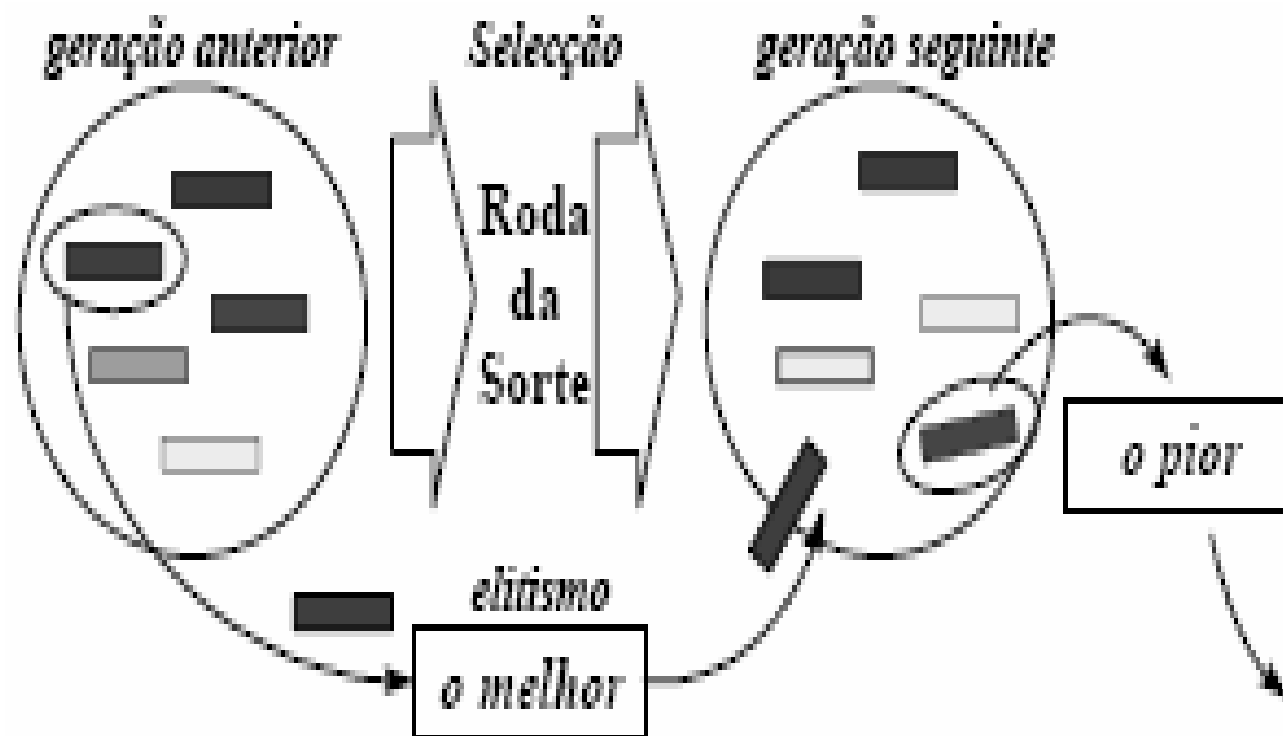
Exemplo

cromossomos	x	$f(x)$	prob. de seleção	
1	1 1 0 1 1	27	729	29,1%
2	1 1 0 0 1	25	625	24,9%
3	1 1 0 0 1	25	625	24,9%
4	1 0 1 1 1	23	529	21,1%

Elitismo

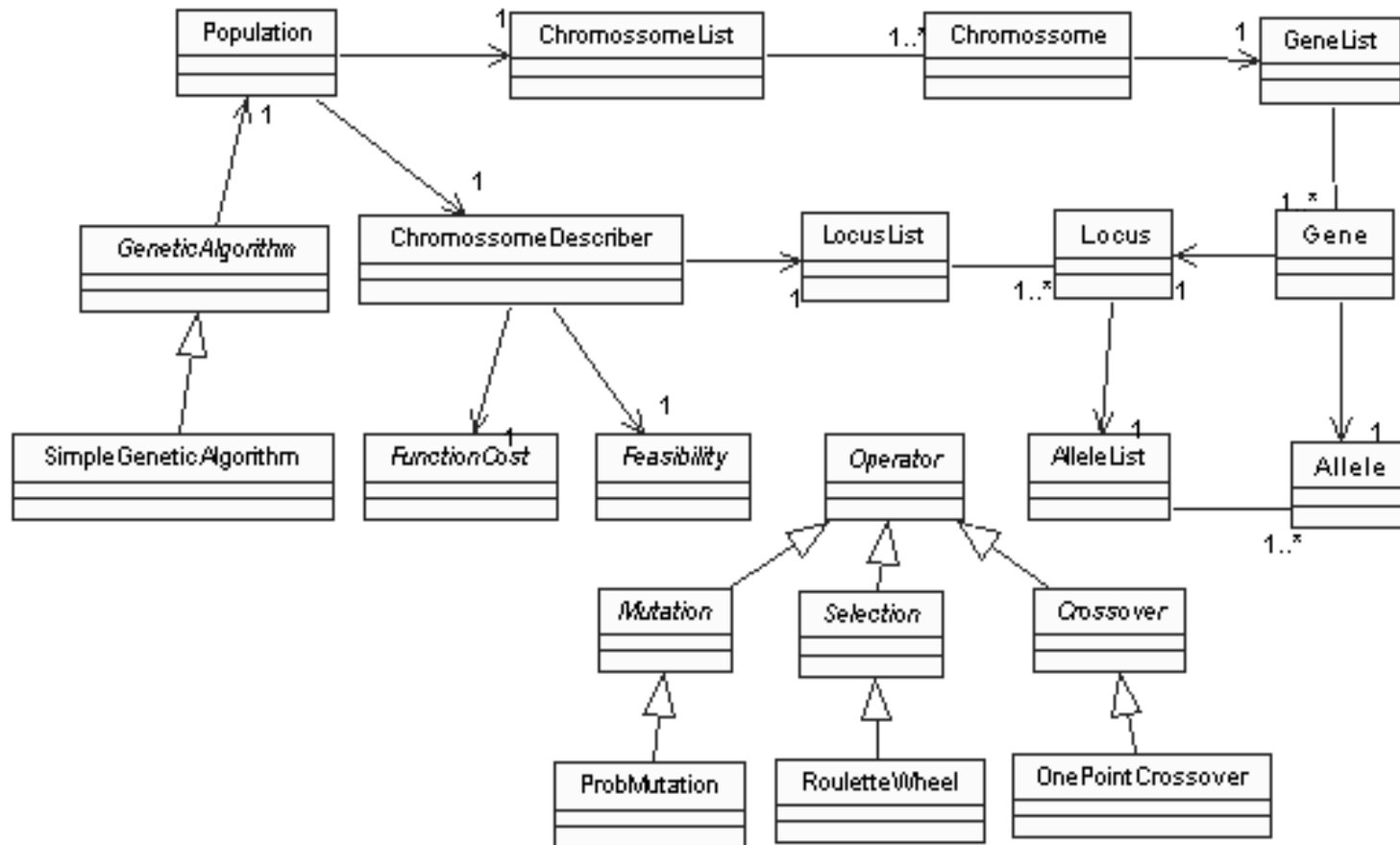
- ❑ Um elemento que tenha maior aptidão que outro tem também maior probabilidade de ser selecionado
- ❑ Nada impede que seja selecionado o pior, perdendo-se assim talvez o melhor elemento da população, que poderia levar a uma convergência mais rápida
- ❑ Para tentar minimizar este possível problema, *elitismo* pode ser adicionado à seleção
- ❑ Percentual de indivíduos com melhor aptidão são mantidos na nova geração

Elitismo



Ferramentas e Aplicações

Framework Orientado a Objetos de um Algoritmo Genético Simples

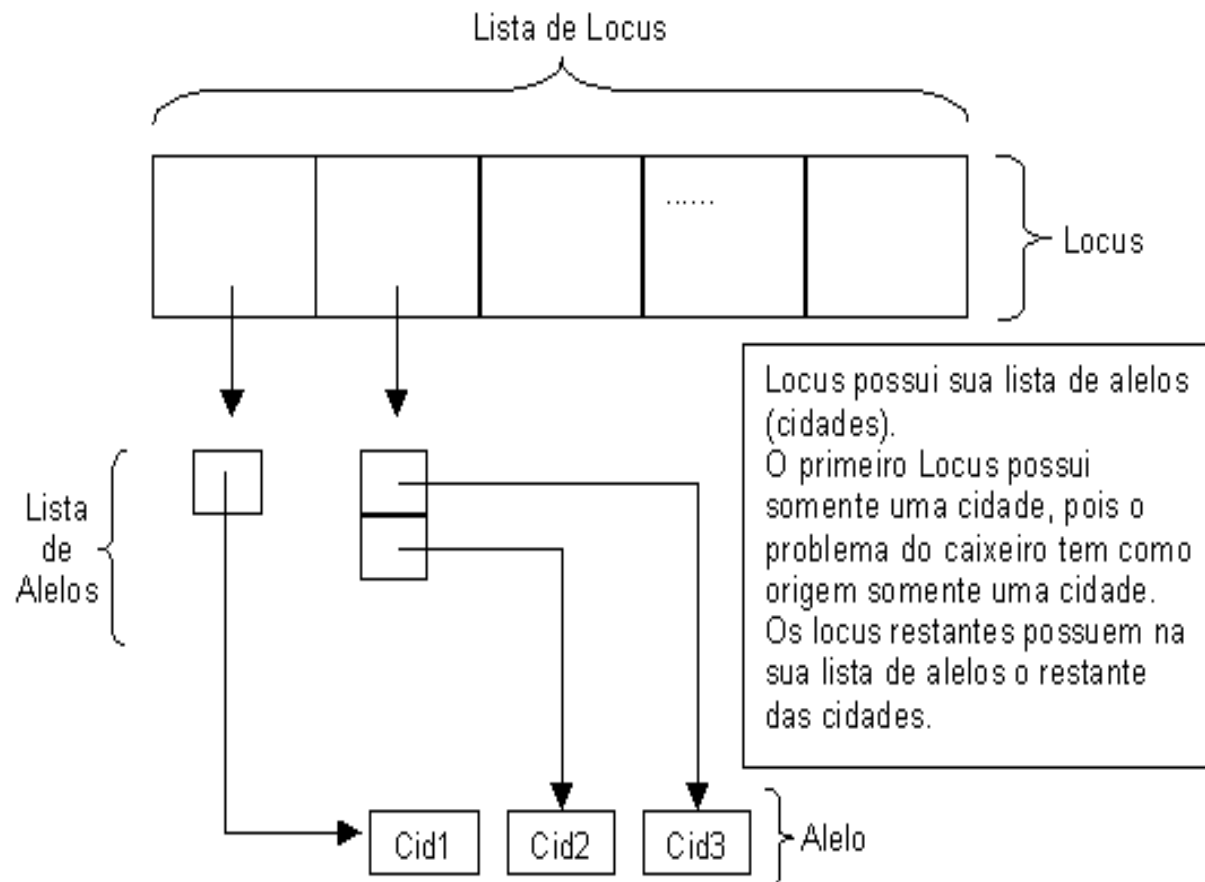


Aplicações do Framework

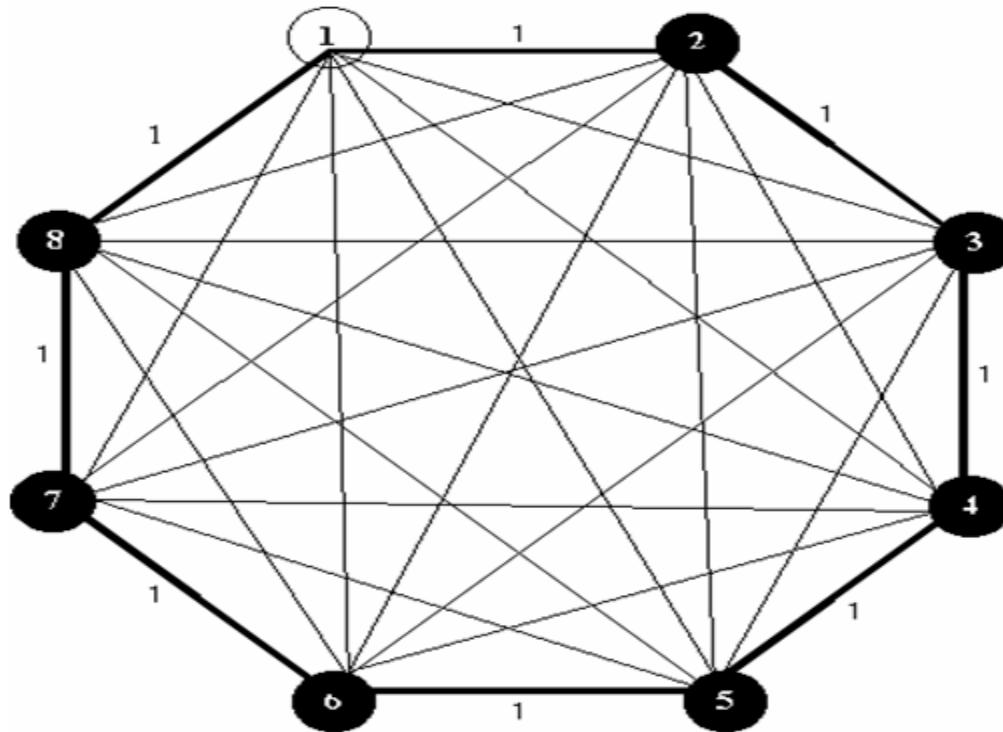
□ Caixeiro Viajante

- Cada gene do cromossomo representa uma cidade, sendo que a ordem dos locus representa a rota. O tamanho do cromossomo é igual a quantidade de cidades envolvidas, que por sua vez é igual ao tamanho da lista de locus

Caixeiro Viajante



Caixeiro Viajante



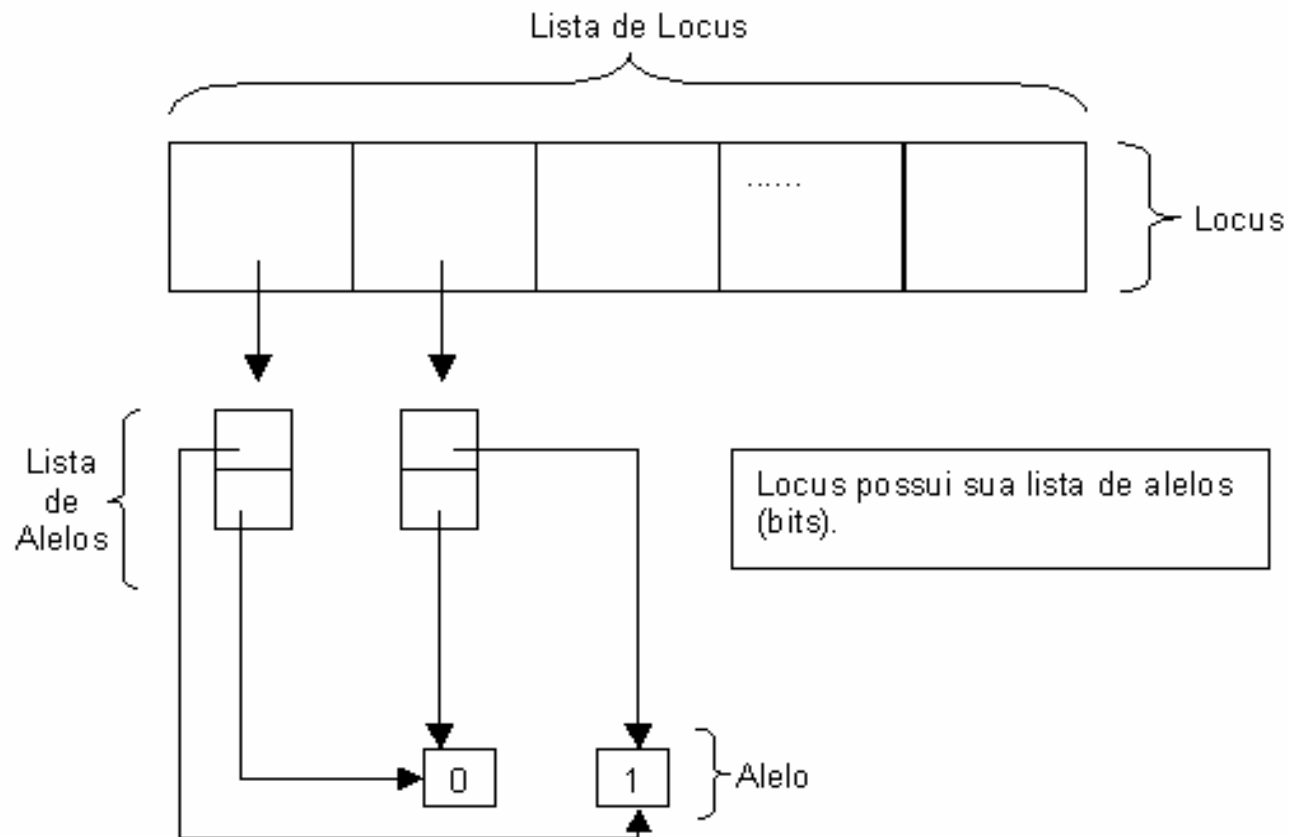
Melhores caminhos:

- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 1$ ou
 - $1 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$
- onde a distância é mínima = 8

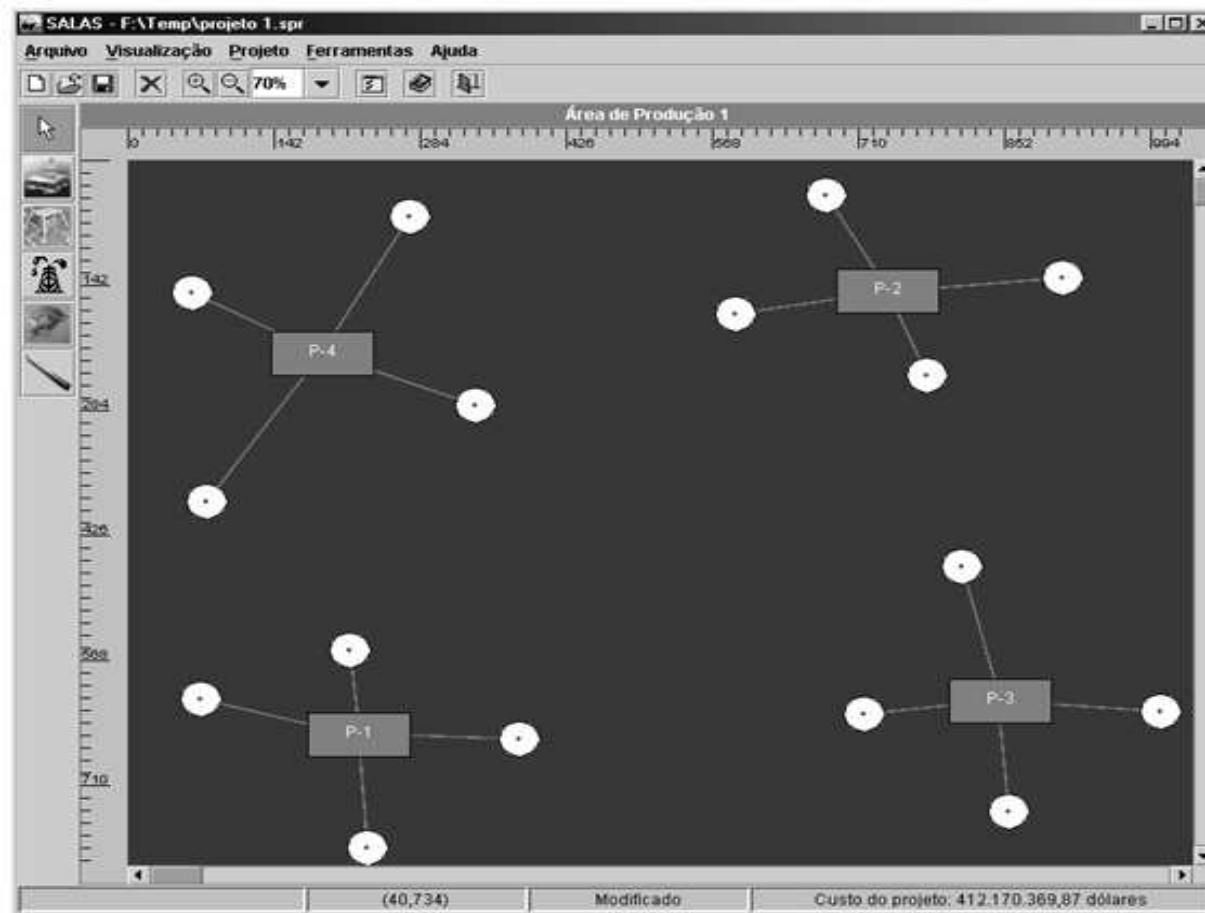
Aplicações do Framework

- Agrupamento de Poços no Layout Submarino
 - O objetivo geral do projeto de layout submarino é a localização de poços e plataformas para exploração de petróleo em águas profundas, minimizando as distâncias entre os elementos e, por consequência, os custos dos dutos condutores da massa líquida entre poços e plataformas
 - O tamanho do cromossomo é igual a quantidade de poços existentes, e cada gene neste cromossomo representa uma plataforma;

Agrupamento de Poços no Layout Submarino



Agrupamento de Poços no Layout Submarino



Aplicações do Framework

- Mecanismo automático para busca de parâmetros de técnicas de classificação
 - Objetivo é determinar a configuração de valores dos parâmetros da técnica que otimizam o desempenho do classificador
 - Um gene do cromossomo, na aplicação, representa um valor de um parâmetro da técnica

Aplicações de Meta-heurísticas

ESAN

- Escalonamento de equipes de estiva e de navios para atendimento em sistemas portuários
 - Escalonamento de navios
 - Escalonamento de equipes dada uma fila de navios

ESAN

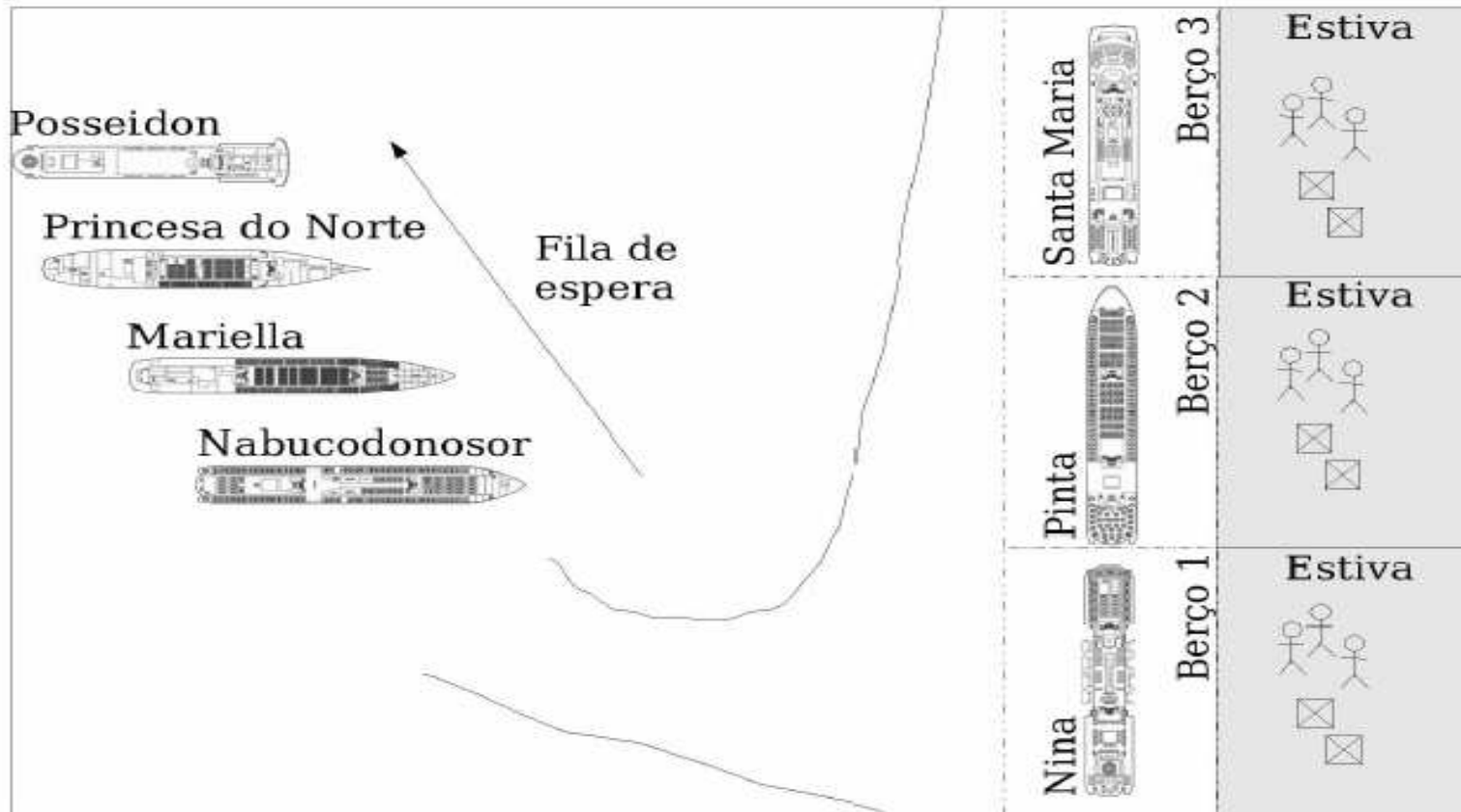
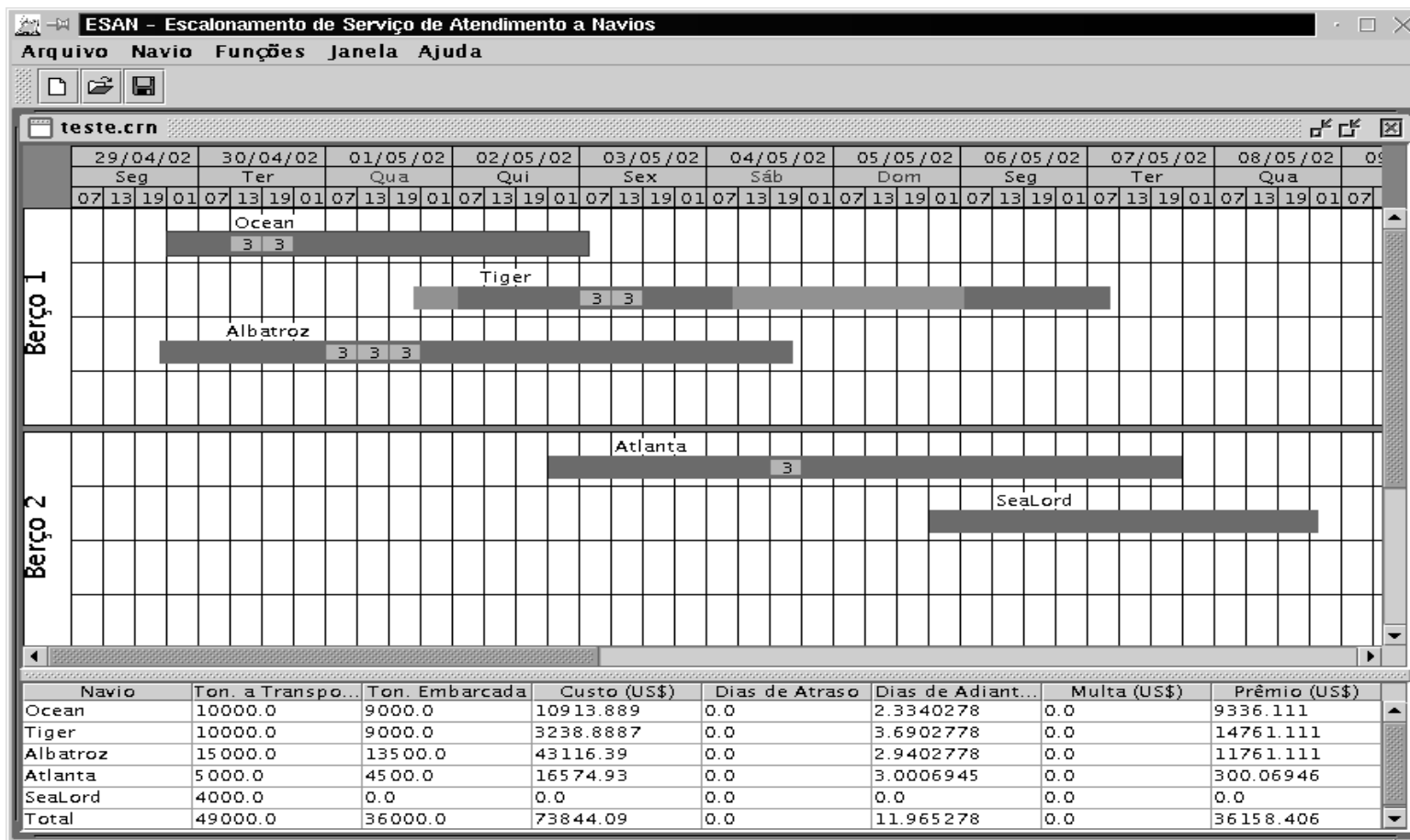


Ilustração 1. Disposição dos navios na fila e nos berços

Distribuição de Navios nos Berços



ESAN

- Meta-heurísticas utilizadas
 - Escalonamento de Navios
 - Beam Search
 - Branch and Bound Heurístico
 - Algoritmo Genético
 - Escalonamento de Equipes
 - Beam Search
 - Branch and Bound Exato

Perspectivas e Desafios

Conclusões

- Meta-heurísticas podem ser aplicadas a vários problemas bem definidos complexos
- Contudo
 - Existe uma grande variedade de heurísticas
 - Escolha de qual aplicar é difícil
 - Devem ser ajustadas para cada aplicação
 - Ajuste pode ser a etapa mais difícil

Conclusões

□ Contudo

- Cada uma requer a especificação de parâmetros que podem afetar o seu desempenho
 - Obter bons valores para os parâmetros requer experimentação
- Solução ótima não é garantida