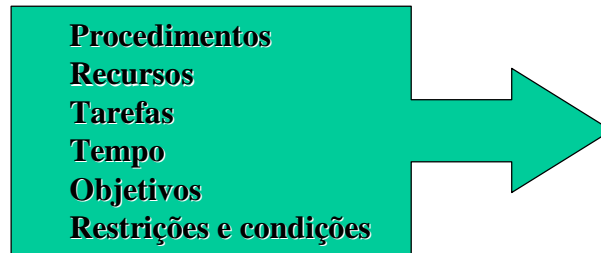


GA em Otimização de Planejamento

- Planejamento envolve:



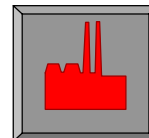
- Otimizar:

– “Alocar , no tempo, os recursos para a execução das tarefas, respeitando as restrições e condições, de modo a alcançar os objetivos do problema.”



Exemplos

- Problema do Caixeiro Viajante
- Otimização da Distribuição (Rota de veículos)
- Alocação de Salas; Timetabling
- Planejamento de Vôos de Cias Aéreas
- Otimização de Estoque/Produção
- Otimização da Produção Industrial



Variáveis Típicas do Planejamento

- **Restrições de Recursos**

- número de instâncias de cada recurso/máquina;
- diferenças entre as instâncias (velocidade, tempo máximo de operação, capacidade etc).
- paradas de manutenção
- máquinas com programação especial: laminadores de aço (largura, espessura e dureza)

- **Restrições Temporais**

- horário de funcionamento preferenciais;
- tempo de transporte de material entre máquinas

- **Reajuste das Máquinas**

- tarefas exigem setup de máquinas (automático ou manual).



Variáveis Típicas do Planejamento (cont.)

- **Prioridade**

- tarefas possuem prioridades diferentes (prazo de entrega, emergência, manutenção, tipo de cliente etc).

- **Estoque**

- matéria prima: disponibilidade, ordem de desempilhamento

- **Reprogramação**

- reprogramação das máquinas em casos de contingências

- **Precedência**

- certas tarefas não podem ser programadas antes que outras tenham terminado.



Características do Planejamento

- Há muitas condições e restrições que não podem ser expressas matematicamente;
- Métodos de busca podem falhar devido aos requisitos de tempo;
- Podar o espaço de busca reduz o tempo de execução mas limita desempenho;
- Heurísticas são úteis para acelerar a busca;
- GA é um técnica adequada a problemas mal estruturados como os de planejamento.



Problema Simples de Planejamento de Produção

- **90 tarefas:** (a, b, c, d, e, f.....)
- cada tarefa possui um **peso** associado a sua importância (lucro, prioridade, benefício etc)
- **30 recursos:** apenas uma instância de cada recurso
- tarefas requerem de **1 a 3 horas** para execução
- programação para um total de **40 horas de produção**
- algumas tarefas têm **restrições** nas primeiras 12 horas do planejamento
- **Objetivo maximizar a soma dos pesos** das tarefas planejadas nas primeiras 40 horas



Modelagem do Algoritmo Genético

1. Problema
2. **Representação**
3. Decodificação
4. Avaliação
5. Operadores
6. Técnicas
7. Parâmetros



Representação

- Cromossoma \equiv permutação (lista) de tarefas

$P_1 = (a, b, c, d, e, \dots, t)$

$P_2 = (d, s, e, h, g, \dots, i)$

- cromossoma codifica a **ordem e a vez** (posição) nas quais as tarefas serão planejadas
- requer decodificador \equiv construtor de planejamentos legais



Decodificador do Cromossoma

- Constrói soluções LEGAIS
- Concentra todo o conhecimento no domínio do problema: restrições, recursos, horários, etc.
- Regra Principal:

“Se uma tarefa está planejada na hora t , uma outra tarefa não pode ser planejada em t , exceto se a interseção dos recursos requisitados é vazia”



Decodificador do Cromossoma

- 1 Pega a primeira tarefa da lista;
- 2 Coloca a tarefa no planejamento a partir de $t=0$;
- 3 Pega próxima tarefa e procura colocá-la no planejamento, considerando as restrições presentes, a partir de $t=0$ até $t=40$ horas;
- 4 Vai para 3 se não terminou a lista.



Avaliação

- Uma possível função considera
 - pesos das tarefas ➔ maximizar a soma das planejadas
 - restrições das tarefas ➔ penalizar se violação
 - período ➔ não planejar se além de t=40 horas

$$A_i = \sum p_t - \sum p_n + \sum p_p + \sum p_v/2$$

- p_t = pesos de todas as tarefas
- p_n = pesos das tarefas não planejadas
- p_p = pesos das tarefas planejadas
- p_v = pesos das tarefas que violam restrição

- $A_i \geq 0$; $A_i = 0$ ➔ nenhuma planejada; $A_i = 2 \sum p_t$ ➔ todas



Avaliando Operadores

- Mutaç o Baseada em Ordem
- Mutaç o Baseada em Posiç o
- Mutaç o por Embaralhamento
- Crossover Baseado em Ordem
- Crossover Baseado em Posiç o
- Recombinaç o de Adjac ncias

Aspectos importantes

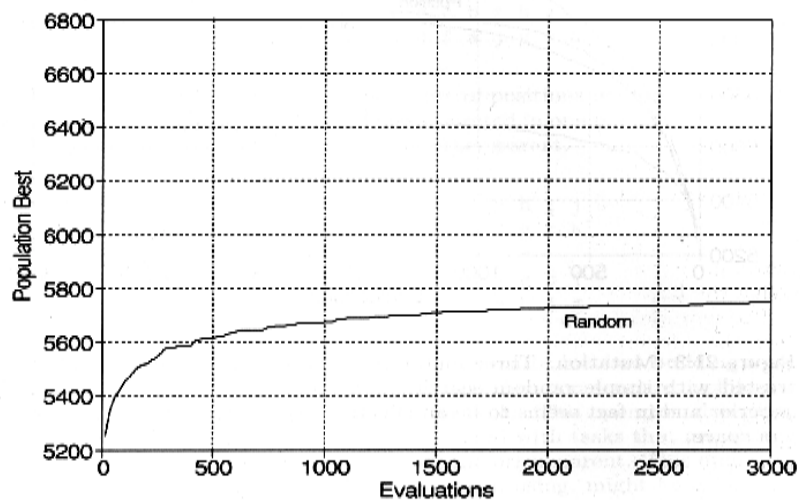
- Ordem Relativa
 - tarefas anteriores podem impedir o planejamento das tarefas posteriores
- Posiç o da Tarefa
 - tarefas no in cio da lista t m maior chance de serem planejadas

Avaliação de Desempenho

- Avaliação dos Operadores isoladamente:
 - só mutação
 - só crossover
- Combinando os Melhores Operadores
- Busca Aleatória para comparar resultados
 - gera uma lista de tarefas e avalia
- Espaço de Representação = $90! = 10^{138}$
- Média de 50 Experimentos de 3000 Indivíduos:
média dos melhores a cada instante



Busca Aleatória



Mutação Baseada em Ordem

- duas tarefas são selecionadas aleatoriamente e a segunda é colocada na frente da primeira

- Exemplo:

(<u>a</u> b c d <u>e</u> f)	(a <u>b</u> c <u>d</u> e f)
↓	↓
(<u>e</u> <u>a</u> b c d f)	(a <u>d</u> <u>b</u> c e f)

- operador preserva ordem relativa de parte do cromossoma



Mutação Baseada em Posição

- troca as posições de duas tarefas escolhidas aleatoriamente

(<u>a</u> b c d <u>e</u> f)	(a <u>b</u> c <u>d</u> e f)
↓	↓
(<u>e</u> b c d <u>a</u> f)	(a <u>d</u> c <u>b</u> e f)

- operador não preserva a ordem relativa das posições selecionadas em relação as tarefas do meio



Mutação por Embaralhamento

- embaralha sub-lista escolhida aleatoriamente

(a b c d e f)	(a b c d e f)
↓	↓
(a c d b e f)	(a b c f d e)

- operador tem maior poder de dispersão da população

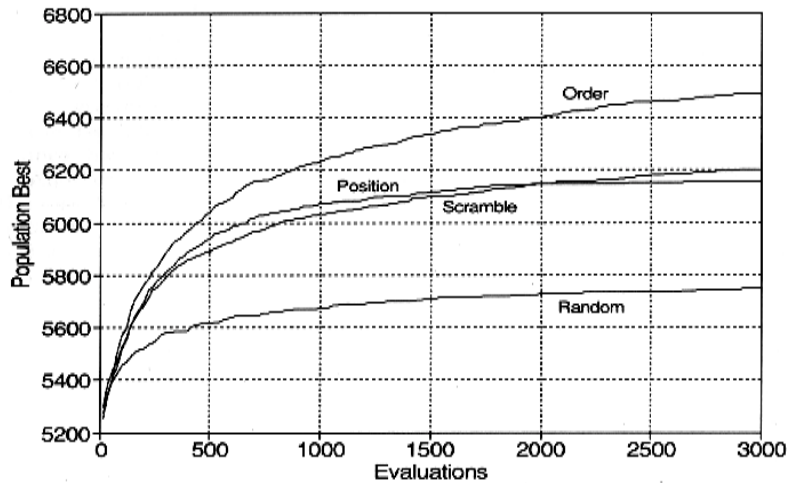


Resultados da Mutação

- Testes sem crossover e com elitismo
- Mutação é mais efetiva que busca aleatória
- Mutação baseada em ordem é mais efetiva
- Embaralhamento é melhor que busca aleatória
- Operadores de mutação são heurísticos, por isso são melhores que busca aleatória



Resultados da Mutaç o



Crossover de Ordem

- Posiç es s o selecionadas aleatoriamente
- Ordem das tarefas nas posiç es selecionadas em um genitor   imposta nas tarefas correspondentes no outro genitor

$P_1 = a \ b \ c \ d \ e \ f \ g \ h \ i \ j$
 $P_2 = e \ i \ b \ d \ f \ a \ j \ g \ c \ h$
 posiç es * * * *

$F_1 = a \ _ \ c \ d \ e \ _ \ _ \ h \ _ \ j$
 $F_2 = _ \ i \ _ \ d \ f \ a \ j \ g \ _ \ _$

$F_1 = a \ i \ c \ d \ e \ b \ f \ h \ g \ j$
 $F_2 = b \ i \ c \ d \ f \ a \ j \ g \ e \ h$



Crossover de Posição

- Posições são selecionadas aleatoriamente
- Posições das tarefas em um genitor são impostas nas tarefas correspondentes no outro genitor

$P_1 = a \ b \ c \ d \ e \ f \ g \ h \ i \ j$
 $P_2 = e \ i \ b \ d \ f \ a \ j \ g \ c \ h$
 posições * * * *

$F_1 = _ \ i \ b \ _ \ f \ _ \ _ \ g \ _ \ _$
 $F_2 = _ \ b \ c \ _ \ e \ _ \ _ \ h \ _ \ _$

$F_1 = a \ i \ b \ c \ f \ d \ e \ g \ h \ j$
 $F_2 = i \ b \ c \ d \ e \ f \ a \ h \ j \ g$



Recombinação de Adjacências

- Crossover combina a informação de adjacências entre as tarefas presentes nos genitores

$P_1 = a \ b \ c \ d \ e \ f$
 $P_2 = b \ d \ c \ a \ e \ f$

$P_1 \Rightarrow ab \ bc \ cd \ de \ ef \ fa$ informação de adjacência
 $P_2 \Rightarrow bd \ dc \ ca \ ae \ ef \ fb$ informação de adjacência

$F = b \ c \ d \ e \ a \ f$
 $F \Rightarrow bc \ cd \ de \ ea \ af \ fb$ informação de adjacência



Recombinação de Adjacências

- Operador foi originalmente criado para o problema do Caixeiro Viajante (TSP)



- No TSP temos:
 - informação de adjacência é importante
 - direção (ordem) entre 2 cidades não importa (A B = B A)

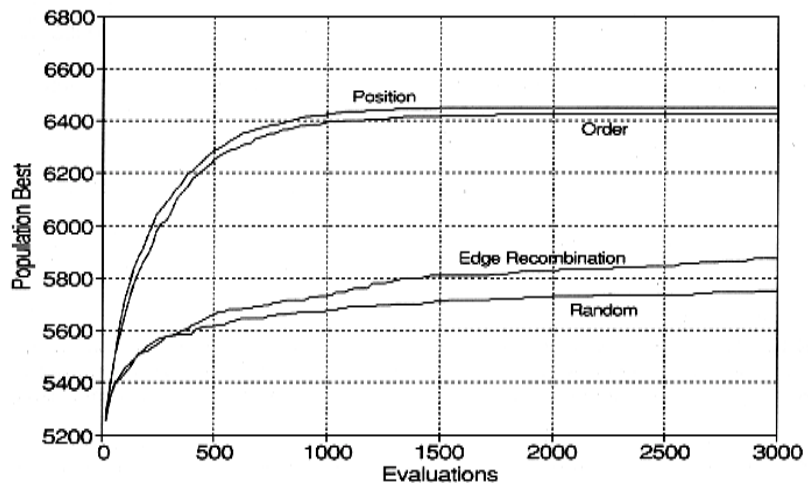


Resultados do Crossover

- Testes sem mutação e com elitismo (rápida evolução no início, nada acontece após 1000 indivíduos)
- Crossover de ordem apresenta resultado equivalente ao de posição: **são de fato o mesmo operador !!**
- Recombinação de Adjacências é equivalente a busca aleatória no problema de planejamento



Resultados do Crossover



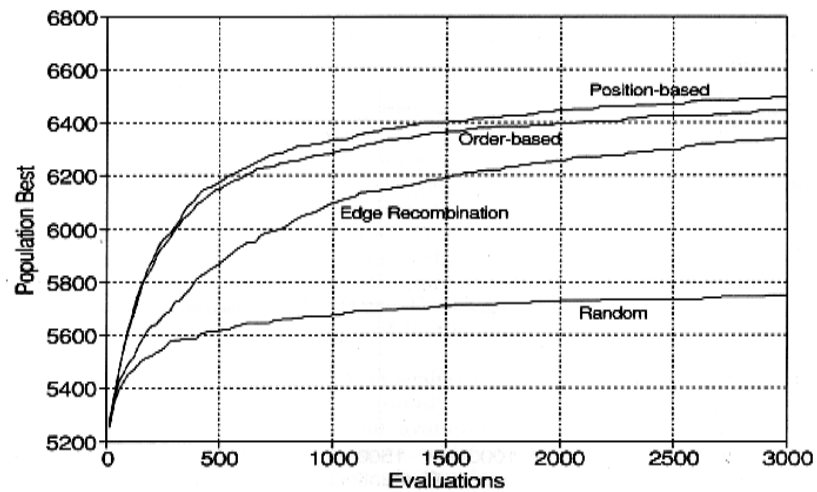
Combinando Crossover e Mutação

- 1 Mutação de Ordem (50%) + Crossover de Ordem (50%)
- 2 Mutação de Ordem (50%) + Crossover de Posição (50%)
- 3 Mutação de Ordem (50%) + Recomb. de Adjacências (50%)

- Curvas menos inclinadas no início e menos planas no final
- Variando os pesos do crossover e mutação, pode-se melhorar o desempenho: aumentando-se a mutação e diminuindo-se o crossover, lentamente, durante a evolução.



Crossover (50%) + Mutaç o (50%)



Variando-se os Pesos de Crossover + Mutaç o

