

UNIVERSIDADE FEDERAL DO RIO GRANDE  
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E FÍSICA  
CURSO DE MATEMÁTICA APLICADA

Thiago Ávila Pouzada

*Algoritmo Genético Modelado por Cadeias de Markov*

Rio Grande  
2013

Thiago Ávila Pouzada

*Algoritmo Genético Modelado por Cadeias de Markov*

Monografia apresentada ao Curso de Matemática Aplicada da FURG, como requisito para a obtenção parcial do grau de BACHAREL em Matemática Aplicada.

**Orientador: Catia Maria dos Santos Machado**

**Doutora em Engenharia de Produção - UFSC**

Rio Grande

2013

Pouzada, Thiago

Algoritmo Genético Modelado por Cadeias de Markov / Thiago

Pouzada - 2013

35.p

1. Estatística; 2. Probabilidade.. I.Título.

CDU 536.21

Thiago Ávila Pouzada

*Algoritmo Genético Modelado por Cadeias de Markov*

Monografia apresentada ao Curso de Matemática Aplicada da FURG, como requisito para a obtenção parcial do grau de BACHAREL em Matemática Aplicada.

Aprovado em 8 de abril de 2013

**BANCA EXAMINADORA**

---

Catia Maria dos Santos Machado

Doutora em Engenharia de Produção - UFSC

---

Paul Gehrard Kinas

PhD em Estatística - University of British Columbia (Canadá)

---

Elaine Corrêa Pereira

Doutora em Engenharia de Produção - UFSC

*À minha mãe, pelo apoio incondicional.*

*À minha irmã, pela força e fé em mim.*

*À minha namorada, pela paciência e pelo alento em todos os momentos.*

*À apaixonante e desafiadora Matemática, por proporcionar intrigantes problemas e motivar o nosso estudo.*

*“A Matemática é o alfabeto com o qual  
Deus escreveu o Universo.”*

*Galileu Galilei*

## Resumo

O Algoritmo Genético (AG) é um algoritmo construído para encontrar máximo ou mínimo de uma função que representa alguma característica do processo que está sendo modelado. Esse algoritmo possui mecanismos que o faz escapar de ótimos locais. Trabalhando com um conjunto de pontos, chamado de população, ele gera uma outra população que sempre será aceita. A forma como o próximo ponto ou a próxima população é gerada obedece propriedades estocásticas. Nesse trabalho, mostramos que a fundamentação matemática que descreve a evolução do AG é a teoria das cadeias de Markov, sendo descrito por uma cadeia de Markov homogênea. Posteriormente, mostramos como a aplicação do AG pode solucionar um importante problema de otimização combinatória, chamado de Problema de Alocação de Berços (PAB).

Palavras-chave: Problema de Alocação de Berços (PAB), Heurística, Otimização Combinatória.

## Abstract

The Genetic Algorithm (GA) is an algorithm used to find maximum or minimum values of a function which represents some special characteristic of the process that is being modelled. This algorithm has mechanisms that prevent it from getting stuck in local maximum or minimum values. Working with a set of points, called population, it creates another population that will always be accepted. The next point or next population is created obeying stochastic properties. In this work, we show that the Markov chain theory describes the evolution of this algorithm, being modelled by a stationary Markov chain. We further show how the GA's application can solve an important combinatorial optimization problem, known as Berth Allocation Problem (BAP).

Key-words: Berth Allocation Problem (BAP), Heuristic, Combinatorial Optimization.

## Agradecimentos

Primeiramente, agradeço a Deus, por tornar possível tudo em minha vida. Por iluminar sempre o meu caminho e por criar forças em mim, principalmente nos momentos mais difíceis.

À minha família, por toda educação, apoio e compreensão prestados, não somente durante o estudo deste trabalho, mas também ao longo de toda minha vida.

À minha namorada Jéssica Vaz, cujo apoio desde o começo foi indispensável. A conclusão com êxito desse estudo se deve muito a ti.

À Professora Catia Machado, pela orientação e apoio incondicionais, pela paciência e incentivo, pela disponibilidade quase integral e pelo apoio prestado ao longo do estudo deste trabalho e ao longo de toda a minha graduação. Sem a sua ajuda, nada disso seria possível.

Aos professores e funcionários do IMEF, que foram responsáveis pela minha formação dentro da universidade e prestaram suas funções com maestria. A vocês, deixo a certeza que o objetivo de vocês foi plenamente realizado.

# Sumário

<b>1</b>	<b>. Introdução</b>	<b>8</b>
1.1	Considerações Iniciais . . . . .	8
1.2	Objetivos . . . . .	9
1.3	Importância do Trabalho . . . . .	9
1.4	Motivação do Estudo . . . . .	10
1.5	Estrutura do Trabalho . . . . .	10
<b>2</b>	<b>. Cadeias de Markov</b>	<b>12</b>
2.1	Processos Estocásticos . . . . .	12
2.2	Processos Markovianos . . . . .	12
2.3	Cadeias de Markov . . . . .	13
2.3.1	Cadeias de Markov Homogêneas . . . . .	13
<b>3</b>	<b>. O Algoritmo Genético</b>	<b>15</b>
3.1	Convergência do Algoritmo Genético . . . . .	17
<b>4</b>	<b>. Modelagem do Algoritmo Genético por Cadeias de Markov</b>	<b>19</b>
4.1	Exemplo de modelagem do AG por Cadeias de Markov . . . . .	19
4.1.1	Operação de Seleção . . . . .	20
4.1.2	Operação de Cruzamento . . . . .	22
4.1.3	Operação de Mutação . . . . .	23
<b>5</b>	<b>. Algoritmo Genético Aplicado ao Problema de Alocação de Berços</b>	<b>26</b>
5.1	O Problema de Alocação de Berços (PAB) . . . . .	26
5.2	Aplicação do PAB pelo Algoritmo Genético . . . . .	27

---

	7
5.3 Modelagem do Problema . . . . .	28
<b>6 . Conclusões</b>	<b>32</b>
6.1 Considerações Finais . . . . .	32
6.2 Sugestões para Pesquisas Futuras . . . . .	33
<b>Referências Bibliográficas</b>	<b>34</b>

# 1 . Introdução

## 1.1 Considerações Iniciais

Tentativas de emulação de fenômenos naturais são bastante antigos, dentro da história da humanidade. Quando um algoritmo que tenta imitar algum processo natural é desenvolvido, utilizando suas propriedades com a finalidade de otimizar uma função ou um processo, ele está desenvolvendo o que chamamos de **heurística**. Segundo Kirkpatrick [6], existem duas estratégias básicas para heurísticas: “dividir e conquistar” e melhoramentos iterativos. No primeiro, divide-se o problema em subproblemas de tamanho administrável, então resolve-se os subproblemas. Depois, as soluções dos subproblemas devem ser juntadas de alguma forma para obter a solução do problema original. No entanto, este método produz soluções muito boas quando os subproblemas são naturalmente disjuntos, e a divisão é de forma apropriada a fim de que os erros cometidos na união das soluções não prejudiquem os ganhos obtidos na aplicação de métodos melhores na solução dos subproblemas. No melhoramento iterativo, começamos com o sistema numa configuração conhecida. Uma operação de reorganização padrão é aplicada em todas as partes do sistema até que uma configuração reorganizada que melhore a função seja descoberta. A configuração reorganizada então se torna a nova configuração do sistema, e o processo continua até que nenhuma outra melhor seja encontrada. Melhoramento iterativo consiste de uma busca no espaço coordenado através de passos de reorganização das configurações, que nos leve para valores menores da função. Visto que esta busca usualmente fica presa em um mínimo local que não é o mínimo global, é de costume realizar várias vezes o processo, começando em diferentes configurações aleatoriamente geradas, e guardando o melhor resultado. Como uma outra alternativa, surgiram algoritmos que possuem mecanismos para impedir que essa busca fique presa em ótimos locais, com o objetivo de melhorar os métodos heurísticos tradicionais sem prejudicar a sua principal característica, a flexibilidade. Dentre eles está o Algoritmo Genético (AG), sendo desenvolvido como uma heurística de processos naturais.

O AG introduzido por Holland [7] é uma heurística baseada no processo de seleção natural de uma população. Neste processo a população passa por três fases:

seleção, cruzamento e mutação. Após isso acontecer, teremos uma nova população que também passará por estas três etapas até que uma condição de parada seja atingida. Rudolph [10] fala que o AG canônico não converge quase certamente para uma população que tenha o ponto de ótimo no seu conjunto de pontos. No entanto, considerando uma modificação neste algoritmo, a saber: o melhor ponto da população inicial é guardado em uma posição, e este ponto não participa como um ponto da população nas etapas de seleção, cruzamento e mutação. Além disso, esse ponto é substituído se um ponto melhor que ele é encontrado nas próximas populações. Com essas modificações, o AG é chamado **AG com elitismo**. O AG com elitismo converge mais rapidamente para a solução, sendo que deve existir um cuidado ao trabalhar com elitismo, para não haver uma convergência prematura (Taufer [13]). É nesse contexto que o AG é muito utilizado na resolução de problemas do mundo real, no entanto poucos trabalhos apontam na direção dos fundamentos matemáticos do algoritmo, que é foco principal do nosso trabalho.

## 1.2 Objetivos

Este trabalho tem como objetivo geral mostrar que o Algoritmo Genético pode ser modelado através de **Cadeias de Markov Homogêneas**. Para alcançar o objetivo geral, os seguintes objetivos específicos são considerados no decorrer do trabalho:

- Mostrar a modelagem do Algoritmo Genético pela Matriz de Transição  $P$ ;
- Mostrar a Convergência do Algoritmo Genético;
- Aplicar todas as etapas do Algoritmo Genético sobre um exemplo didático;
- Mostrar a importância do Algoritmo Genético na vertente aplicada.

## 1.3 Importância do Trabalho

As técnicas clássicas de otimização são confiáveis e possuem aplicações na engenharia e em diversas outras ciências. Porém, estas técnicas apresentam dificuldades numéricas e problemas de robustez, muitas vezes relacionados com a falta de continuidade das funções a serem otimizadas ou de suas restrições, funções não convexas, existência de ruídos nas funções, necessidade de trabalhar com valores discretos para as

variáveis, existência de mínimos ou máximos locais, entre outros. Dessa forma, os estudos de métodos heurísticos, como o Algoritmo Genético, vem sendo conduzidos no sentido de vencer tais dificuldades, principalmente devido ao avanço dos recursos computacionais.

## 1.4 Motivação do Estudo

O Algoritmo Genético é um algoritmo iterativo amplamente utilizado na resolução de problemas de otimização, em especial, otimização combinatória. Devido à sua facilidade de implementação e principalmente por sua robustez, muitas pesquisas são desenvolvidas em testes competitivos com outros algoritmos. Entretanto, poucos estudos apontam na direção de sua fundamentação matemática. Assim, motivados pela Dissertação intitulada *Modelagem dos Algoritmos Genético Simples e Simulated Annealing por Cadeias de Markov*, de Neto [2], apresentamos o AG Simples, Clássico ou Canônico, modelado por meio de Cadeias de Markov Homogêneas.

## 1.5 Estrutura do Trabalho

Este trabalho é estruturado em 6 capítulos, abordados da seguinte forma:

- Capítulo 1: Mostramos uma breve introdução ao estudo.
- Capítulo 2: Apresentamos as Cadeias de Markov, definindo alguns conceitos. Tratamos do estudo de Processos Estocásticos e Markovianos. Logo após, entendemos o que são Cadeias de Markov Homogêneas, requisito fundamental para os capítulos seguintes.
- Capítulo 3: Nesse capítulo, entendemos o que é o Algoritmo Genético e o seu funcionamento, apresentando definições e teoremas fundamentais e suas demonstrações, culminando na sua convergência.
- Capítulo 4: No quarto capítulo, apresentamos como funciona a modelagem do Algoritmo Genético Simples através das Cadeias de Markov, apresentando um exemplo didático de fácil entendimento. É nesta parte do trabalho que as etapas da referida modelagem são especificadas.

- 
- Capítulo 5: Uma vez compreendida a modelagem em questão, aplicamos na vertente desejada, que é o Problema de Alocação de Berços. Apresentamos esse problema e propomos a sua solução através do Algoritmo Genético, modelado por Cadeias de Markov.
  - Capítulo 6: No capítulo das conclusões, traçamos algumas observações sobre os objetivos desejados e alcançados, bem como possíveis segmentos de estudos, dentro dessa área de concentração.

## 2 . Cadeias de Markov

Primeiramente, faremos algumas definições e conceitos das Cadeias de Markov, as quais serão necessárias para o entendimento da modelagem do Algoritmo Genético.

### 2.1 Processos Estocásticos

Um Processo Estocástico é definido como uma coleção de variáveis aleatórias  $X(t)$  indexadas por um parâmetro  $t$  que pertence a um conjunto  $T$ .

Normalmente,  $T$  é utilizado como o conjunto dos inteiros não-negativos, representando o tempo (não havendo problemas em utilizar outros conjuntos), e  $X(t)$  é uma característica mensurável de interesse, no tempo  $t$ .

Os valores assumidos por  $X(t)$  são chamados de **estados** e o conjunto de todos os possíveis estados de um Processo Estocástico é chamado de **Espaço de Estados**.

Se o Espaço de Estados é discreto, ele também pode ser chamado de **cadeia**.

### 2.2 Processos Markovianos

Um Processo Estocástico é dito ser um **Processo Markoviano** se:

$$P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k, X(t_{k-1}) = x_{k-1}, \dots, X(t_0) = x_0\} = \\ P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k\}$$

Para  $t_0 \leq t_1 \leq \dots \leq t_k \leq t_{k+1} = 0, 1, 2, \dots$  e toda sequência  $k_0, k_1, \dots, k_{t-1}, k_t, k_{t+1}$ .

Ou seja, a probabilidade condicional de qualquer evento futuro, dado qualquer evento passado e o estado presente  $X(t_k) = x_k$ , é independente do evento passado, dependendo somente do estado presente.

Logo, um Processo Estocástico é um Processo Markoviano se o estado futuro depende somente do estado presente e não dos estados passados. Esse tipo de Processo

Estocástico também é denominado de *memoryless process* (processo sem memória), uma vez que os estados passados são “esquecidos”.

As probabilidades condicionais  $P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k\}$  são denominadas **Probabilidades de Transição** e representam a probabilidade do estado  $X(t_{k+1})$  ser menor ou igual a  $x_{k+1}$  no instante  $t_{k+1}$ , uma vez que o estado  $X(t_k)$  é  $x_k$  no instante  $t_k$ .

## 2.3 Cadeias de Markov

Um Processo Markoviano é dito ser uma **Cadeia de Markov** quando as variáveis aleatórias  $X(t)$  estão definidas em um espaço de estados discreto  $E$ .

Quando o tempo é discreto, a Cadeia de Markov é dita ser uma **Cadeia de Markov em Tempo Discreto**. Nesse caso, tem-se:

$$P\{X(k+1) = x_{k+1} | X(k) = x_k, X(k-1) = x_{k-1}, \dots, X(1) = x_1, X(0) = x_0\} = P\{X(k+1) = x_{k+1} | X(k) = x_k\},$$

para toda sequência  $0, 1, 2, \dots, k-1, k, k+1$ .

Portanto, as probabilidades de transição  $P\{X(k+1) = x_{k+1} | X(k) = x_k\}$  representam a probabilidade do estado  $X(k+1)$  ser  $x_{k+1}$  no tempo  $k+1$  dado que o estado  $X(k)$  é  $x_k$  no tempo  $k$ .

### 2.3.1 Cadeias de Markov Homogêneas

**Definição 2.3.1.** Uma Cadeia de Markov é homogênea ou estacionária no tempo se a probabilidade de ir de um estado a outro é independente do tempo em que o passo é dado, isto é, para um espaço de estados  $E$  e para todos os estados  $x_{k+1}, x_k \in E$ , temos:

$$P\{X(k+1) = x_{k+1} | X(k) = x_k\} = P\{X(k+m) = x_{k+1} | X(k+m-1) = x_k\}$$

Para  $m = -(n-1), -(n-2), \dots, -1, 0, 1, 2, \dots$ .

Onde  $P\{X(k+1) = x_{k+1} | X(k) = x_k\}$  representa a probabilidade de o estado  $X(k+1)$  ser  $x_{k+1}$  no tempo  $k+1$ , dado que o estado  $X(k)$  é  $x_k$  no tempo  $k$ .

Se a condição de estacionariedade falha, então a Cadeia de Markov é dita não-homogênea ou não-estacionária.

Dada uma Cadeia de Markov Homogênea com espaço de estados  $E = \{1, 2, \dots, n\}$ , podemos organizar as probabilidades  $p_{ij} = P\{X(k+1) = j | X(k) = i\}$  em uma matriz  $n \times n$  onde  $p_{ij}$  ocupa a  $i$ -ésima linha e a  $j$ -ésima coluna desta matriz. Esta matriz  $P = (p_{ij})$  é chamada de **matriz de transição** de probabilidade da cadeia.

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{bmatrix}$$

Essa matriz de transição tem as seguintes propriedades:

- Todas as entradas são não negativas;
- A soma das entradas em cada linha  $i$  é 1.

A matriz de transição contém toda informação necessária para descrever o movimento da cadeia ao longo dos estados em  $E$ .

Em geral, se queremos as probabilidades de transição de  $n$  passos, essas probabilidades são dadas pela matriz  $P^n$ , onde os seus elementos são denotados por:

$$p_{ij}^{(h)} = P\{X(h+m) = j | X(m) = i\}$$

Se estamos interessados em saber onde o processo está em um tempo particular, devemos primeiro saber onde o processo começa.

**Definição 2.3.2.** Um vetor  $f^{(0)} = (f_1^{(0)}, f_2^{(0)}, \dots, f_n^{(0)})$  dado por uma distribuição de probabilidade sobre os estados é dito um **vetor inicial** se:

$$f_i^{(0)} = P\{X_0 = i\} \geq 0$$

$$\sum_{i=1}^n f_i^{(0)} = 1$$

Maiores detalhes são encontrados no trabalho de Neto [2].

### 3 . O Algoritmo Genético

A ideia dos Algoritmos Genéticos vem de uma analogia feita por Holland [7] com processos naturais de seleção, nos quais indivíduos com características genéticas melhores têm maiores chances de sobrevivência, enquanto indivíduos menos aptos tendem a desaparecer. Na natureza, há o processo de cruzamento, onde os indivíduos sobreviventes passarão para os seus descendentes suas características genéticas, que por sua vez terão chances de saírem também vencedores. Por outro lado, novos indivíduos poderão surgir através do processo de mutação, no qual a natureza insere material genético diferente. Se este ser que sofreu mutação estiver tão ou mais capacitado à sobrevivência quanto os atuais, terá grandes chances de sobrevivência no futuro processo de seleção.

Os Algoritmos Genéticos tentam simular este processo natural da seguinte maneira: temos uma função positiva  $f : \Omega \rightarrow \mathbb{R}$  (chamada função objetivo), da qual queremos encontrar seu(s) ponto(s) de ótimo (máximo).  $\Omega$  é discretizado, ou seja, é criada uma malha de pontos de  $\Omega$ , chamada de **espaço de busca**. Assim, a solução será procurada dentro deste espaço de busca. Quanto mais precisão desejarmos, mais pontos a malha terá. Cada ponto desta malha (chamado de **cromossomo** ou **indivíduo**) é um candidato ao ponto de ótimo. A quantidade de pontos desta malha geralmente é uma potência de 2, possibilitando que, na enumeração dos pontos da malha, cada ponto seja escrito na **forma binária**. Cada **bit (0 ou 1)** é chamado de **gene**. Um cromossomo é mais apto do que o outro se sua imagem pela  $f$  for maior.

A ideia então é começar com um conjunto de pontos gerados aleatoriamente (chamando de população inicial) e utilizar a função objetivo  $f$  na etapa de seleção para “melhorar” a população inicial. Logo depois, na etapa de cruzamento, haverá uma troca de bits entre pares de cromossomos escolhidos, segundo uma probabilidade (probabilidade de cruzamento). Finalmente, temos a etapa de mutação, onde bits poderão ser trocados segundo uma probabilidade (probabilidade de mutação). Depois destas três etapas teremos uma nova população de mesmo tamanho que a inicial e que passará pelas mesmas etapas, gerando novas populações. O Algoritmo Genético que iremos tratar é o descrito por Goldberg [11], conhecido como *Simple Genetic Algorithm*.

Em linhas gerais, o Algoritmo Genético Simples possui uma população de tamanho fixo e estrutura de dados do tipo cadeias binárias. A seleção natural entre os seus indivíduos é proporcional ao *fitness* (aptidão), através do Método da Roleta. O cruzamento entre os seus indivíduos é simples, e a mutação é pontual.

Resumindo, o AG inicia sua busca com uma população  $(s_1^0, s_2^0, \dots, s_n^0)$ , geralmente aleatoriamente escolhida, a qual é chamada de população inicial. A seguir, cria uma população  $(s_1^{t+1}, s_2^{t+1}, \dots, s_n^{t+1})$  no tempo  $t + 1$  a partir de uma população no tempo  $t$ . Para atingir esse objetivo, os indivíduos da população do tempo  $t$  passam pelas fases de seleção, cruzamento e mutação. Ao fim dessas operações é gerada a população no tempo  $t + 1$ , esperando-se que seja “melhor”, no sentido de ter pontos com as imagens maiores que as imagens da população anterior. Esse procedimento se repete até que um critério de parada seja satisfeito.

A seguir mostraremos os teoremas fundamentais na modelagem do AG por Cadeias de Markov.

Consideremos uma Cadeia de Markov  $\{X_n\}_{n \in \mathbb{N}}$ , com espaço de estados  $E_1$ , que é o conjunto de todas as populações formadas por  $k$  cromossomos. Gostaríamos de saber quem é a matriz de transição desta cadeia, e a resposta está demonstrada no teorema abaixo.

**Teorema 3.0.1 (Teorema da Matriz de Transição).** *A evolução do Algoritmo Genético é modelada pela matriz de transição  $P$ , que é decomposta por um produto de matrizes estocásticas  $P = SCM$ , onde  $S$ ,  $C$  e  $M$  modelam as transições realizadas nos passos de seleção, cruzamento e mutação, respectivamente.*

*Demonstração.* Denotemos por  $X_s, X_c$  e  $X_m$  as etapas intermediárias entre o passo  $n$  e  $n + 1$ , ou seja,  $X_s$  representa a etapa em que o ponto inicial é a população atual e o resultado é a população resultante da seleção. Vemos que esta etapa é uma Cadeia de Markov, pois o resultado da seleção só depende da etapa anterior. Assim podemos construir sua matriz de transição por  $S_{(A,B)} = P(X_s = B | X_n = A)$ . Da mesma forma as matrizes de transição do cruzamento e da mutação podem ser construídas da seguinte forma:  $C_{(A,B)} = P(X_c = B | X_s = A)$  e  $M_{(A,B)} = P(X_m = B | X_c = A)$ . Note que  $X_m$  será o resultado da mutação, ou seja, a próxima população, assim:

$$M_{(A,B)} = P(X_m = B|X_c = A) = P(X_{n+1} = B|X_c = A)$$

Com isso temos:

$$\begin{aligned} p_{AB} &= P(X_{n+1} = B|X_n = A) \\ &= \frac{P(X_{n+1} = B, X_n = A)}{P(X_n = A)} \\ &= \sum_{C \in E_1} \frac{P(X_{n+1} = B, X_s = C, X_n = A)}{P(X_n = A)} \\ &= \sum_{D \in E_1} \sum_{C \in E_1} \frac{P(X_{n+1} = B, X_c = D, X_s = C, X_n = A)}{P(X_n = A)} \\ &= \sum_{D \in E_1} \sum_{C \in E_1} P(X_{n+1} = B|X_c = D, X_s = C, X_n = A)P(X_c = D|X_s = C, X_n = A) \\ &\quad \cdot P(X_s = C|X_n = A) \\ &= \sum_{D \in E_1} \sum_{C \in E_1} P(X_{n+1} = B|X_c = D)P(X_c = D|X_s = C)P(X_s = C|X_n = A) \\ &= \sum_{D \in E_1} \sum_{C \in E_1} S_{(A,C)}C_{(C,D)}M_{(D,B)} = SCM_{(A,B)} \end{aligned}$$

Como esta igualdade vale quaisquer que sejam  $A, B \in E_1$ , temos a matriz de transição  $P = SCM$ .

□

### 3.1 Convergência do Algoritmo Genético

Daremos condições sobre as matrizes de transição, de modo que o AG convirja. Para tanto, relembremos algumas definições e finalizaremos a seção provando o teorema que garante a convergência.

**Definição 3.1.1.** A probabilidade  $p_m$  de partir de uma população inicial  $Y = (y_1, y_2, \dots, y_k)$  para uma população final  $X = (x_1, x_2, \dots, x_k)$  é dada por:

$$p_m^{H(X,Y)}(1 - p_m)^{nl - H(X,Y)}$$

Onde  $nl$  é o comprimento do cromossomo, ou seja, o número de bits ou genes que o cromossomo possui;  $H(X, Y)$  é chamada de **Distância de Hamming** entre  $X$  e

$Y$ , e representa o número de bits ou genes que devem ser alterados dentro do cromossomo pela última etapa, a mutação, para transformar  $Y$  em  $X$ .

**Definição 3.1.2.** Uma matriz é dita positiva quando todos os seus elementos são positivos.

**Definição 3.1.3.** Uma matriz quadrada é dita estocástica quando todos os elementos são não negativos e a soma dos elementos de cada linha é igual a 1.

**Teorema 3.1.1 (Teorema da Positividade da Matriz P).** *A matriz  $P$ , obtida na construção feita pelo teorema anterior, é positiva.*

*Demonstração.* Na seção anterior mostramos que as entradas da matriz de transição da mutação  $M$  são dadas por  $M(X, Y) = p_m^{H(X, Y)}(1 - p_m)^{nl - H(X, Y)}$ . Como  $p_m > 0$ , temos que todas as entradas desta matriz são positivas, portanto  $M$  é positiva. Como o produto de matrizes estocásticas resulta em uma matriz estocástica, temos que  $SC$  é estocástica. Logo, podemos concluir que  $P = SCM$  é positiva.

□

**Proposição 3.1.1.** *Toda Cadeia de Markov Homogênea irredutível, aperiódica com espaço de estados finito é convergente.*

A Proposição 3.1.1 pode ser estudada com maiores detalhes em Neto [2].

**Teorema 3.1.2 (Teorema da Convergência da Cadeia de Markov Representada pela Matriz P).** *A Cadeia de Markov representada pela matriz  $P = SCM$  é convergente.*

*Demonstração.* Como  $P$  é positiva, temos que todas as suas entradas são positivas. Isso implica que  $P(A, B) > 0$  e  $P(B, A) > 0$  para todas as populações  $A, B \in E_1$ . Então implica que a cadeia é irredutível. Temos também que todos os estados possuem o mesmo período, e como  $P(A, A) > 0$  pelo fato de  $P$  ser positiva, temos que o período da cadeia é 1 e a cadeia é aperiódica. Como a quantidade de pontos do espaço de estados é finito, temos que a cadeia em questão é finita, irredutível e aperiódica. Logo, pela Proposição 3.1.1, temos que a cadeia é convergente.

□

## 4 . Modelagem do Algoritmo Genético por Cadeias de Markov

Nesta seção, modelaremos o Algoritmo Genético através das Cadeias de Markov. Vimos no capítulo anterior que a população seguinte é gerada levando em conta apenas a anterior. Vimos no capítulo 2 que a propriedade de Markov diz exatamente isso, ou seja, um processo no qual o presente só depende do passado mais próximo. Desta forma, é natural considerar uma Cadeia de Markov em que o espaço de estados sejam todas as possíveis populações.

### 4.1 Exemplo de modelagem do AG por Cadeias de Markov

Com um exemplo de fácil entendimento, consideremos o problema no qual queremos minimizar uma função simples.

$$\text{Min} f(x) = |x - 20|, \text{ sujeito à } x \in \{0, 1, 2, \dots, 63\}.$$

Discretizamos da seguinte forma, sobre o alfabeto binário  $A = \{0, 1\}$ :

$x_i$	<b>Cromossomo</b>
0	000000
1	000001
2	000010
3	000011
⋮	⋮
62	111110
63	111111

Aleatoriamente, foram escolhidos para formar a população inicial os indivíduos:

Indivíduo	$x_i$	Cromossomo	$f(x_i)$
$y_1$	1	000001	19
$y_2$	6	000110	14
$y_3$	15	001111	5
$y_4$	24	011000	4
$y_5$	26	011010	6
$y_6$	34	100010	14
$y_7$	53	110101	33

Onde  $f(x_i)$  é o valor da função objetivo para cada indivíduo selecionado, para  $i = 1, 2, \dots, 7$ .

Dessa forma, o espaço de estados da Cadeia de Markov é definido como:

$$E = (y_1, y_2, y_3, y_4, y_5, y_6, y_7), \quad y_i \in E \quad \forall \quad i = 1, 2, 3, \dots, 7.$$

População Inicial			
$c_i$	Cromossomo	$f(x_i)$	$F_a(x_i)$
$c_1$	000001	19	19
$c_2$	000110	14	33
$c_3$	001111	5	38
$c_4$	011000	4	42
$c_5$	011010	6	48
$c_6$	100010	14	62
$c_7$	110101	33	95

Onde  $F_a(x_i)$  é a função objetivo acumulada de todos os indivíduos. Seja  $F_a(x_i)$  definida na tabela anterior, como:

$$F_a(x_i) = \sum_{i=1}^7 f(x_i) = 95$$

#### 4.1.1 Operação de Seleção

No Método da Roleta, os indivíduos tem uma probabilidade de ser selecionados proporcionalmente ao valor de suas imagens pela função objetivo, ou seja, é o valor de  $f$

no ponto sobre o somatório do valor da função em todos os pontos da população. Seja a frequência relativa  $fa$  definida como:

$$fa_i = \frac{f(x_i)}{F_A(x_i)}$$

$c_i$	$f(x_i)$	$fa_i$	$F_A$
$c_1$	19	0,2	0,2
$c_2$	14	0,15	0,35
$c_3$	5	0,05	0,4
$c_4$	4	0,042	0,442
$c_5$	6	0,063	0,505
$c_6$	14	0,15	0,655
$c_7$	33	0,35	1

Onde  $F_A$  são os indivíduos distribuídos proporcionalmente à função objetivo.

Suponha que tenham sido gerados os seguintes números aleatórios:

$$r_1 = 0,61 \rightarrow 0,6 < 0,61 < 1, \text{ selecionamos } c_7.$$

$$r_2 = 0,008 \rightarrow 0 < 0,008 < 0,2, \text{ selecionamos } c_1.$$

$$r_3 = 0,47 \rightarrow, \text{ selecionamos } c_5.$$

$$r_4 = 0,80 \rightarrow, \text{ selecionamos } c_7.$$

$$r_5 = 0,34 \rightarrow, \text{ selecionamos } c_2.$$

$$r_6 = 0,38 \rightarrow, \text{ selecionamos } c_3.$$

$$r_7 = 0,007 \rightarrow, \text{ selecionamos } c_1.$$

Assim, temos:

População Intermediária	Cromossomo	População Provisória
$c'_1$	110101	$c_7$
$c'_2$	000001	$c_1$
$c'_3$	011010	$c_5$
$c'_4$	110101	$c_7$
$c'_5$	000110	$c_2$
$c'_6$	001111	$c_3$
$c'_7$	000001	$c_1$

Observamos que os cromossomos  $c_1$  e  $c_7$  aparecem duas vezes cada. Os demais cromossomos não se repetem.

Dada uma população  $y = y_1, y_2, \dots, y_7$ , então a probabilidade de o algoritmo gerar a população  $x_1, x_2, \dots, x_7$  após a seleção é:

$$P = \frac{f(b_i)}{\sum_{j=1}^7 f(b_j)} > 0$$

Podemos observar que a etapa de seleção é uma Cadeia de Markov.

### 4.1.2 Operação de Cruzamento

A probabilidade de cruzamento  $p_c \in [0, 1]$  é especificada de acordo com a necessidade do problema, pelo usuário.  $r_i$  gera números aleatórios.

Se  $r_i < p_c$ , o indivíduo  $i$  é escolhido para o cruzamento.

O primeiro selecionado é cruzado com o segundo selecionado, o terceiro é cruzado com o quarto, e assim segue, até que tenhamos  $\frac{i}{2}$  pares, sendo  $i$  a quantidade de indivíduos selecionados.

Seja  $p_c = 0,3$  a probabilidade de cruzamento, e os seguintes números aleatórios selecionados:

$$\begin{aligned} r_1 &= 0,5; 0,5 > 0,3; \text{ não selecionamos } c_1. \\ r_2 &= 0,17; 0,17 < 0,3; \text{ selecionamos } c_2 \rightarrow 000001. \\ r_3 &= 0,4; 0,4 > 0,3; \text{ não selecionamos } c_3. \\ r_4 &= 0,2; 0,2 < 0,3; \text{ selecionamos } c_4 \rightarrow 110101. \\ r_5 &= 0,23; 0,23 < 0,3; \text{ selecionamos } c_5 \rightarrow 000110. \\ r_6 &\text{ não é selecionado.} \\ r_7 &\text{ não é selecionado.} \end{aligned}$$

Logo, selecionamos  $c_2, c_4$  e  $c_5$  para o cruzamento.

De acordo com a lei de seleção (o primeiro selecionado cruza com o segundo, e assim por diante), temos que  $c_2$  irá cruzar com  $c_4$ .

É interessante salientar que há um ponto de corte para a população ser selecionada também. Ele será um número aleatório  $t$  entre 1 e  $nl - 1$ , onde  $nl$  é o número de bits ou genes dentro do cromossomo (no nosso exemplo,  $nl = 6$ ). Por exemplo:

Seja  $t = 3$ .

$c_2^i$	000:001	$c_1$	19
$c_4^i$	110:101	$c_7$	33

Trocando os gens dos cromossomos e fazendo os cruzamentos, temos:

$c_2^j$	000:101	15
$c_4^j$	110:001	29

Assim, após a etapa de cruzamento, a população será dada por:

$c_1^i$	110101
$c_2^j$	000101
$c_3^i$	011010
$c_4^j$	110001
$c_5^i$	000110
$c_6^i$	001111
$c_7^i$	000001

### 4.1.3 Operação de Mutação

A mutação opera independentemente em cada bit do indivíduo da população através de uma probabilidade  $p_m \in [0, 1]$ , chamada **probabilidade de mutação**.

Para aplicar o operador de mutação, é necessário gerar  $7 \times 6 = 42$  números aleatórios  $r_i$ , onde 7 é o número de indivíduos da população, 6 é o tamanho do cromossomo e  $r_i \in [0, 1]$ . Seja  $p_m = 0,01$  a probabilidade de mutação, se  $r_i < p_m$ , será feita a mutação no bit correspondente. Considere que foram gerados 42 números aleatórios,  $r_i \in [0, 1]$  e que dois tiveram probabilidades menores que  $p_m$ . Foram os seguintes:

$$r_{13} = 0,009 < p_m$$

$$r_{39} = 0,0025 < p_m$$

Considerando a população atual e pensando na organização da população em um vetor da forma  $(c_1, c_2', c_3, c_4', c_5, c_6, c_7)$ , onde o primeiro bit será o primeiro bit de  $c_1$ , no caso 1; o sexto elemento será o sexto elemento de  $c_1$ , no caso 1, o sétimo bit será o primeiro elemento de  $c_2'$ , no caso 0, e assim sucessivamente, temos:

$c_1$	110101
$c_2'$	000101
$c_3$	$\underbrace{0}_{bit13} 11010$
$c_4'$	110001
$c_5$	000110
$c_6$	001111
$c_7$	$00 \underbrace{0}_{bit39} 001$

Submetendo os bits 13 e 39 ao processo de mutação, temos:

$c_1$	110101
$c_2'$	000101
$c_3''$	111010
$c_4'$	110001
$c_5$	000110
$c_6$	001111
$c_7''$	001001

Concluindo a geração do algoritmo, temos:

<b>Indivíduo</b>	<b>Cromossomo</b>	$x_i$	$f(x_i)$
$c_1$	110101	53	33
$c_2'$	000101	5	15
$c_3''$	111010	58	38
$c_4'$	110001	49	29
$c_5$	000110	6	14
$c_6$	001111	15	5
$c_7''$	001001	9	11

Se pegarmos o indivíduo  $c_3$ , no qual trocamos o primeiro bit 0 por 1, a Distância de Hamming, definida no capítulo 3, será de 1, pois alteramos apenas um bit dentro do cromossomo. Então, apenas no primeiro bit do cromossomo, que foi trocado, representaremos por  $p_m$  como a probabilidade de troca. Nos outros cinco bits, como não houve alterações, representaremos por  $1 - p_m$ . Logo, a probabilidade de que o cromossomo  $c_3$  se transforme no cromossomo  $c_3''$  será:

$$p_m \cdot (1 - p_m) = p_m \cdot (1 - p_m)^5$$

Conforme definimos no capítulo 3, a probabilidade de que o cromossomo  $c_3$  se transforme no cromossomo  $c_3''$  é:

$$H(X, Y) = 1 \text{ e } nl = 6;$$

$$p_m^1 (1 - p_m)^{6-1} = p_m (1 - p_m)^5$$

Logo, foi verificado que a relação estabelecida no capítulo 3 é verdadeira.

Terminada esta etapa, concluímos uma geração do algoritmo. Observamos que a função objetivo aumentou em relação à função objetivo inicial. O valor dessa era 4, e após o término da geração passou a ser 5. No Algoritmo Genético podem ocorrer algumas soluções de piora. Mais detalhes podem ser encontrados em Barrico [15].

## 5 . Algoritmo Genético Aplicado ao Problema de Alocação de Berços

Segundo Pizzolato [5], na vertente aplicada estão aqueles que fazem pesquisa visando resolver desafios do mundo real, notadamente complexos, beneficiando-se das contribuições proporcionadas pela área acadêmica. O trabalho desenvolvido por Pereira [1] que trata do Problema de Alocação de Berços (PAB) seguiu a metodologia básica da Pesquisa Operacional, a qual passou pela identificação do problema, a formulação de um modelo matemático com o uso de hipóteses simplificadoras, a resolução do modelo, a validação dos resultados e o posterior oferecimento de propostas para implementação. Em relação à resolução do problema de otimização combinatória PAB, os métodos exatos garantem encontrar a solução ótima, no entanto, podem gastar um tempo proibitivo para gerar tal solução, nesse caso os métodos heurísticos são largamente utilizados e merecem estudos teóricos. Os métodos heurísticos são baseados na simples ideia de que uma solução ótima local pode ser melhorada na medida em que são aplicados procedimentos de busca local em uma solução que pode ser obtida fazendo a exploração no espaço de soluções. O Algoritmo Genético, método baseado em busca populacional, consiste em manter um conjunto de boas soluções e procura combiná-las, de forma a tentar produzir soluções ainda melhores.

### 5.1 O Problema de Alocação de Berços (PAB)

O PAB é um problema que atribui aos navios que chegam a um determinado porto posições de atracação disponíveis ao longo de um cais, as quais chamamos de **berços**. Porém, o problema enfrenta duas decisões correlacionadas: onde e quando os navios devem atracar. Os navios que chegam ao porto irão atracar no berço mais conveniente, ou em um berço livre que possa recebê-los. Caso não haja um berço livre adequado à operação do navio, esse irá para uma fila de navios para aguardar a sua atracação. Assim, o tempo em que o navio aguarda um berço de atracação na fila é o parâmetro utilizado como principal nível de serviço na área portuária (Cordeau [8]). Na prática,

o PAB possui um número grande de soluções possíveis, por se tratar de um problema de Otimização Combinatória. Para resolver um problema dessa natureza, a ideia mais simples é combinar, ou seja, enumerar todas as possíveis soluções, criar todos os subconjuntos existentes a partir do conjunto e das regras de restrição e escolher o de menor custo. A própria palavra combinação, no contexto matemático, sugere uma grande quantidade de possibilidades, quando tratamos de muitos elementos considerados - como no caso do PAB. Portanto, este procedimento torna-se impraticável à medida que o conjunto inicial cresce. Um algoritmo exato poderá não encontrar um ótimo em tempo hábil, ou seja, o algoritmo poderá levar décadas para encontrar a solução desejada. Assim, a busca pelo Algoritmo Genético ganha importância e hoje representa uma opção muito utilizada na resolução de Problemas de Otimização Combinatória (Silva [12]).

## 5.2 Aplicação do PAB pelo Algoritmo Genético

A escolha do Algoritmo Genético na resolução do PAB se deve ao fato de o algoritmo apresentar vantagens como versatilidade, eficiência e simplicidade. Além disso, a modelagem do problema possui características próprias, cuja técnica heurística AG (Simples, Clássico ou Canônico) pode sofrer modificações tornando-se adaptável ao problema específico. A facilidade de implementação e principalmente a robustez da técnica na resolução de problemas como o problema de alocação e programação de navios a berços tem despertado o interesse de muitos pesquisadores. O PAB possui então, como principal objetivo, minimizar os custos referentes ao porto e ao armador, que é relacionado ao tempo de serviço; nesse caso, o tempo total de atendimento, considerado desde o momento da chegada do navio ao porto, sua atracação e sua saída do berço. Em relação ao local de atracação, na dimensão espacial, há restrições em relação à profundidade da água (calado do berço) e com a distância máxima em relação à localização mais favorável ao longo do cais, calculadas com relação à localização da saída do contêiner e para o espaço reservado para a entrada do contêiner. Na dimensão temporal, as restrições são expressas como janelas de tempo para o tempo total de conclusão de serviço do navio. Algumas janelas de tempo são suaves e podem ser relaxadas com um custo adequado.

## 5.3 Modelagem do Problema

Formulação matemática:

**Minimizar:**

$$\begin{aligned}
 Z^* = & w_0 \sum_{i \in N} \sum_{k \in M} v_i \left( T_i^k - a_i + t_i^k \sum_{j \in N \cup d(k)} x_{ij}^k \right) + \\
 & w_1 \sum_{i \in N} \sum_{k \in M} \sum_{j \in N \cup d(k)} x_{ij}^k (max(0, a_i - T_i^k) + max(0, T_i^k + t_i^k - b_i)) + \\
 & w_2 \sum_{k \in M} (max(0, s^k - T_{o(k)}^k) + max(0, T_{d(k)}^k + e^k))
 \end{aligned}$$

**Sujeito à:**

$$\begin{aligned}
 \sum_{k \in M} \sum_{j \in N \cup d(k)} x_{ij}^k &= 1 \quad \forall i \in N \\
 \sum_{j \in N \cup d(k)} x_{o(k)j}^k &= 1 \quad \forall k \in M \\
 \sum_{i \in N \cup o(k)} x_{id(k)}^k &= 1 \quad \forall k \in M \\
 \sum_{j \in N \cup d(k)} x_{i,j}^k - \sum_{j \in N \cup o(k)} x_{j,i}^k &= 0 \quad \forall k \in M, \quad \forall i \in N \\
 T_i^k + t_i^k - T_j^k &\leq (1 - x_{ij}^k) M_{ij}^k \quad \forall k \in M, \quad \forall (i, j) \in A^k \\
 x_{ij}^k &\in \{0, 1\} \quad \forall k \in M, \quad \forall (i, j) \in A^k
 \end{aligned}$$

**Onde:**

$N$  = conjunto de navios;

$M$  = conjunto de berços;

$x_{ij}^k \in \{0, 1\} \quad \forall k \in M, \quad \forall (i, j) \in A^k, \quad x_{ij}^k = 1$  se o navio  $j$  é atendido pelo berço  $k$  após o navio  $i$ ;

$T_i^k \quad \forall k \in M, \quad i \in N$  = horário que o navio  $i$  atracou no berço  $k$ ;

$T_{o(k)}^k \quad \forall k \in M$  = horário que o primeiro navio atracou no berço  $k$ ;

$T_{d(k)}^k \quad \forall k \in M$  = horário que o último navio atracou no berço  $k$ ;

$t_i^k$  = duração do atendimento do navio  $i$  no berço  $k$ ;

$a_i$  = horário de chegada para o navio  $i$ ;

$b_i$  = horário de término de janela de tempo para o navio  $i$ ;

$v_i$  = valor (custo) de tempo de serviço para o navio  $i$ ;

$s^k$  = horário de abertura do berço  $k$ ;

$e_k$  = horário de fechamento do berço  $k$ ;

$M_{ij} = \max\{b_i + t_i^k - a_j, 0\}, \quad \forall k \in M \quad \text{e} \quad \forall (i, j) \in N.$

Os resultados obtidos por Mauri et al. [9] utilizando a técnica *Simulated Annealing* (SA) para a resolução do modelo apresentou instigou o estudo da fundamentação teórica do Algoritmo Genético para o mesmo modelo. O Algoritmo Genético aplicado ao Problema de Alocação de Berços tenta simular o processo natural como apresentado no Capítulo 3. A geração da população inicial é dada através das heurísticas de distribuição e programação abaixo:

• **Heurística de Distribuição:**

1. **CRIAR** ( $m$  berços vazios);
2. **CRIAR** (uma lista  $L$  com todos os navios);
3. **ORDENAR** (a lista  $L$  pelo horário de chegada dos navios ao porto);
4. **PARA** (cada navio  $j$  em  $L, j = 1, 2, \dots, n$ ) FAÇA
5.       **SELECIONAR** (um berço  $i, i = 1, 2, \dots, m$ );
6.       **SE** (o berço  $i$  não puder atender ao navio  $j$ )
7.               **VOLTAR** (para o passo 5);
8.       **SENÃO**
9.               **ATRIBUIR** (o navio  $j$  ao berço  $i$ );
10. **FIM PARA.**

• **Heurística de Programação:**

1. **SEJA** ( $k$  um berço qualquer);
2. **PARA** (cada navio  $i$  atribuído a  $k$ ) FAÇA

3.

$$T_i^k = \max(a_i, s^k), \text{ se } i = 1;$$

$$T_i^k = \max(a_i, T_{i-1}^k + t_{i-1}^k), \text{ se } i > 1.$$

4. **FIM PARA**;5. **CALCULAR** (a função objetivo para o berço  $k$ ).

Os indivíduos são representados através de um vetor numérico tal que a sequência de navios representa os navios  $N_i$  a serem alocados em um determinado berço  $B_k$ , conforme mostrado na tabela abaixo:

$B_k$					
$N_1$	$N_2$	$N_3$	$\dots$	$\dots$	$N_i$

Dessa forma, cada indivíduo (cromossomo) determina uma solução para o PAB que são as alocações dos navios nos berços. O custo dessa programação é o valor da função objetivo, ou aptidão do indivíduo da população. Para a seleção dos indivíduos foi utilizado o **Método da Roleta** e o operador de cruzamento utilizou informações heurísticas. Esse método vem descrito nos seguintes passos:

1. Escolher um navio aleatório de um dos cromossomos pais;
2. Comparar os custos de alocação para os próximos navios em ambos os pais e selecionar o menor;
3. Se o menor custo escolhido formar um ciclo na rota parcial, então escolher um custo aleatório que não introduza um ciclo;
4. Repetir os passos “1” e “3” até que todos os navios estejam incluídos no berço.

$P_1$								
$N_1$	$N_4$	$N_2$	$N_8$	$N_5$	$N_7$	$N_3$	$N_6$	$N_9$
$P_2$								
$N_7$	$N_5$	$N_3$	$N_4$	$N_1$	$N_8$	$N_6$	$N_9$	$N_2$
$O_1$								
$N_8$	$N_5$	$N_3$	$N_6$	$N_9$	$N_2$	$N_7$	$N_4$	$N_1$

Nas tabelas apresentadas acima, temos um exemplo do operador de cruzamento, onde foi escolhido aleatoriamente o navio  $N_8$ , e dessa forma é analisado o custo de alocação para  $N_5$  e  $N_6$ , que são, respectivamente, os próximos navios em  $P_1$  e  $P_2$ . Supondo que o menor custo seja para a sequência  $N_8 \rightarrow N_5$ , segue-se com a análise de custo para a próxima alocação. Quando chega-se ao navio  $N_7$  temos um ciclo, dessa forma escolhemos aleatoriamente qualquer navio que não introduza um ciclo. No exemplo foi escolhido o navio  $N_4$ , restando apenas o navio  $N_1$  para fechar o cromossomo.

A partir dos indivíduos selecionados, o **operador de mutação** faz trocas aleatórias de alguns genes dos indivíduos, fazendo com que novas áreas no espaço de busca possam ser pesquisadas. Dessa forma evita a convergência prematura da população, mantendo a diversidade populacional e fugindo de mínimos locais.

O operador de mutação implementado foi o proposto por Pereira [1], chamado de **operador de inversão simples**, onde dois pontos no cromossomo são selecionados aleatoriamente e a subsequência entre estes pontos é invertida, conforme mostrado nas tabelas abaixo:

$B_1$								
$N_1$	$N_4$	$N_2$	$N_8$	$N_5$	$N_7$	$N_3$	$N_6$	$N_9$
$B_2$								
$N_1$	$N_4$	$N_2$	$N_3$	$N_7$	$N_5$	$N_8$	$N_6$	$N_9$

Observe que os indivíduos em vermelho foram trocados de lugar no cromossomo, simplesmente invertendo-se a sua ordem de colocação.

Finalmente, como **Critério de Parada**, diferentes critérios podem ser escolhidos para finalizar a execução do AG, tais como:

- Após um dado numero de gerações, ou seja, um total de ciclos de evolução;
- Quando a aptidão média ou do melhor indivíduo não melhora mais;
- Quando as aptidões dos indivíduos de uma população se tornarem parecidas;
- Perda de diversidade da população (o AG começa a convergir quando não há mais soluções que melhorem a função objetivo).

## 6 . Conclusões

### 6.1 Considerações Finais

Quando tratamos do Algoritmo Genético modelado por Cadeias de Markov, dentro da literatura existem muitos estudos de aplicações e implementações. No nosso trabalho, o objetivo principal é incentivar mais estudos sobre este assunto, dentro de centros de pesquisa e universidades, pois como já foi comentado, o AG é uma ferramenta muito poderosa na solução de problemas de otimização combinatória, como por exemplo o PAB. A efetividade de um método heurístico como o AG é extraordinariamente maior do que a efetividade de um método exato. Ainda, é perfeitamente aplicável e confiável em uma série de problemas importantes de caráter complexo.

Além disso, buscamos instigar o estudo aprofundado desta modelagem, especialmente abordando a fundamentação matemática que baseia todo o processo. É muito importante entender a implementação e aplicação da ferramenta, como existem muitas referências, porém não menos importante é compreender a base matemática, sendo ela contribuinte e determinante para o interesse acerca do assunto. Os estudos já existentes ainda carecem de uma abordagem matemática mais ampla e profunda.

Dessa forma, podemos dizer que os objetivos específicos foram alcançados, pois o AG modelado por Cadeias de Markov se mostrou de fato uma excelente alternativa para problemas, como na vertente que trabalhamos. A principal contribuição deste estudo é apresentar as etapas do AG sobre o exemplo didático do capítulo 4.

Dentro da aplicação matemática, o trabalho proporcionou o entendimento de diversos aspectos da teoria, unindo-os com a sua utilização direta na realidade. O próprio estudo de Cadeias de Markov se mostrou fundamental, com uma grande compatibilidade às necessidades do Algoritmo Genético.

O Algoritmo Genético se mostrou uma técnica muito efetiva, dentro do Problema de Alocação de Berços. A sua convergência e flexibilidade justificam a sua utilização, ainda que existam outros métodos da mesma grandeza de funcionalidade. Tendo em vista o modelo apresentado, é esperado que essa ferramenta se torne mais usual,

inclusive para problemas e modelos semelhantes.

A proposta do trabalho foi alcançada com êxito, porém inúmeras possibilidades de sequência se apresentam, como segue na seção seguinte.

## 6.2 Sugestões para Pesquisas Futuras

Como alternativas de contribuições para este estudo, ou mesmo possíveis segmentos deste trabalho, citamos alguns pontos relevantes abaixo:

- Estudar o Algoritmo *Simulated Annealing* (SA), outra ferramenta heurística poderosa, através do entendimento de Cadeias de Markov Não-homogêneas;
- Comparar os algoritmos AG e SA, uma vez que ambos são excelentes alternativas na tomada de decisões em problemas de otimização combinatória;
- Implementar a modelagem estudada em um porto real, no qual algumas restrições e características diferenciadas podem surgir, de modo que o problema seja mais aproximado da realidade;
- Modelar o Algoritmo Genético através de Cadeias de Markov Não-homogêneas.

## Referências Bibliográficas

- [1] Pereira, E. D. *O Problema de Alocação de Berços: Um estudo das heurísticas Simulated Annealing e Algoritmo Genético*. Dissertação de Mestrado, Pós-Graduação em Modelagem Computacional. FURG, Rio Grande, 2013.
- [2] Neto, J. C. R. *Modelagem dos Algoritmos Genético Simples e Simulated Annealing por Cadeias de Markov*. Dissertação de Mestrado, Pós-Graduação em Matemática Aplicada e Estatística. UFRN, Natal, 2010.
- [3] Nogueira, F. *Modelagem e Simulação - Cadeias de Markov*. UFJF, Juiz de Fora. Disponível em: <http://www.ufjf.br/epd042/files/2009/02/cadeiaMarkov1.pdf>. Último acesso em 10 de março de 2013.
- [4] Rodrigues, M. A. P. *Problema do Caixeiro Viajante: Um Algoritmo para resolução de problemas de grande porte baseado em busca local dirigida*. Dissertação de Mestrado, Pós-Graduação em Engenharia de Produção. UFSC, Florianópolis, 2000.
- [5] Pizzolato, N. D. *Revisão de Desafios Aplicados em Localização com base em Modelos da p-mediana e suas variantes*. Revista Podes, 4, pp. 13-42, 2012.
- [6] Kirkpatrick, S.; Gellat, D. C.; Vecchi. M. P. *Optimization by Simulated Annealing*. Science, v.220, p. 671-680, 1983.
- [7] Holland, J. H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [8] Cordeau, J. F.; Laporte, G.; Mercier, A. *A unified tabu search heuristic for vehicle routing problems with time windows*. Journal of the Operational Research Society, v.52, p. 928-936, 2001.
- [9] Mauri, G. R.; Oliveira, A. C. M.; Lorena, L. A. N. *Heurística baseada no Simulated Annealing aplicada ao problema de alocação de berços*. GEPROS - Gestão da Produção, Operações e Sistemas, Ano 3, v.1, n.1, p. 113-127, 2008.

- [10] Rudolph, G. *Convergence analysis of canonical genetic algorithms*. IEEE Transactions on Neural Networks, special issue on Evolutionary Computation, 5, 96-101, 1994.
- [11] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA: Addison Wesley, 1989.
- [12] Silva, A. S. N. *Estudo e Implementação, Mediante Recozimento Simulado, do Problema de Alocação de Salas*. Monografia, Departamento de Ciência da Computação. Universidade Federal de Lavras, 2005.
- [13] Taufer, F. S. G. *Análise dos Operadores de Cruzamento do Algoritmo Genético Aplicado ao Problema do Caixeiro Viajante*. Dissertação de Mestrado, Pós-Graduação em Modelagem Computacional. FURG, Rio Grande, 2012.
- [14] Clarke, A. B.; Disney, R. L. *Probabilidade e Processos Estocásticos*. Editora Livros Técnicos e Científicos. Rio de Janeiro, 1979.
- [15] Barrico, C. M. C. S. *Optimização Evolucionária Multi-Objectivo em Ambientes Incertos - Pesquisa de Soluções Robustas*. Tese de Doutorado, Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Ciências e Tecnologia. Universidade de Coimbra, 2007. Disponível em: <http://www.di.ubi.pt/cbarrico/Doutoramento/Download/TeseDoutoramento.pdf>. Último acesso em 15 de março de 2013.