

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARCELO BRANCO DO NASCIMENTO

**APLICAÇÃO DO ALGORITMO DE OTIMIZAÇÃO BAYESIANO
AO PROBLEMA DE ALOCAÇÃO
DE BERÇOS**

São Luís
2013

MARCELO BRANCO DO NASCIMENTO

**APLICAÇÃO DO ALGORITMO DE OTIMIZAÇÃO BAYESIANO
AO PROBLEMA DE ALOCAÇÃO
DE BERÇOS**

Monografia apresentada ao curso de
Ciência da Computação da Universidade
Federal do Maranhão, como parte dos
requisitos necessários para obtenção do
grau de Bacharel em Ciência da
Computação.

Orientador: Prof. Dr. Alexandre César
Muniz de Oliveira

São Luís
2013

Nascimento, Marcelo Branco do.

Aplicação do algoritmo de otimização bayesiano ao problema de alocação de berços/ Marcelo Branco do Nascimento. – São Luís, 2013.

56 f.

Impresso por computador (fotocópia).

Orientador: Alexandre César Muniz de Oliveira.

. Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2013.

1. Redes Bayesianas. 2. Algoritmo de otimização bayesiano. 3. Problema de alocação de berços. I. Título.

CDU 004: 519.226.5

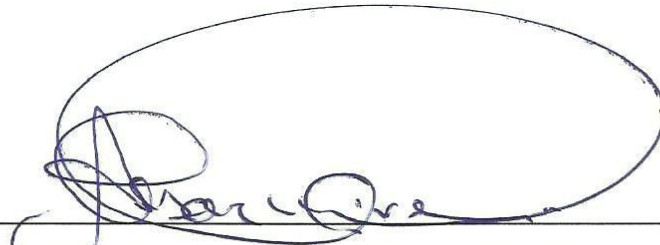
MARCELO BRANCO DO NASCIMENTO

**APLICAÇÃO DO ALGORITMO DE OTIMIZAÇÃO BAYESIANO
AO PROBLEMA DE ALOCAÇÃO
DE BERÇOS**

Monografia apresentada ao curso de
Ciência da Computação da Universidade
Federal do Maranhão, como parte dos
requisitos necessários para obtenção do
grau de Bacharel em Ciência da
Computação.

Aprovada em 12 / 12 / 2013

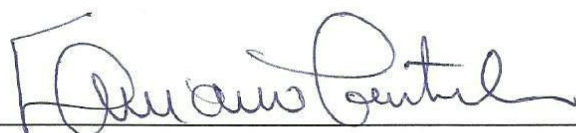
BANCA EXAMINADORA



Prof. Dr. Alexandre César Muniz de Oliveira
Departamento de Informática/UFMA - Orientador



Prof. MSc. Tarcísio Souza Costa
Departamento de Educação Profissional/IFMA - Membro



Prof. Dr. Luciano Reis Coutinho
Departamento de Informática/UFMA - Membro

À minha família

AGRADECIMENTOS

Agradeço à minha família, que ofereceu todo o suporte necessário para o meu crescimento moral e profissional.

Aos professores do curso de Ciência da Computação, tendo destaque ao meu orientador pela Alexandre César Muniz de Oliveira pelos conhecimentos passados e pelo acompanhamento na realização dos trabalhos.

Aos amigos do curso onde pude trocar informações e adquirir experiências de vida.

Aos meus amigos que não são do curso, mas também me deram apoio na realização da monografia e nos quesitos pessoais.

E por fim a todos aqueles que contribuíram direta ou indiretamente na construção desse documento.

*“A mente que se abre a uma nova idéia jamais
voltará ao seu tamanho original.”*

(Albert Einstein)

RESUMO

Este trabalho apresenta uma metodologia com aplicação de redes bayesianas para a resolução de problemas de otimização linear. Essa metodologia é conhecida na literatura como algoritmo de otimização bayesiano e baseia-se no processo de modelagem matemática do problema de alocação de berços, dada a função objetivo e as restrições, seguida da modelagem para ajuste nos conceitos de redes bayesianas, obtendo assim um modelo consistente. Para isso foi apresentado alguns conceitos introdutórios sobre teoria dos grafos e teoria das probabilidades.

É também realizada um estudo em cima dos algoritmos de estimação de distribuição mostrando implicações em relação ao modelo dos algoritmos de otimização bayesianos.

O modelo foi testado e comparado com resultados reais a fim de que haja a garantia da utilização dessa técnica para resolução de problemas afins. Os resultados adquiridos apresentaram boa conformidade com os resultados dos ótimos reais.

Palavras-chave: Redes Bayesianas. Problema de Alocação de Berço. Algoritmos de Otimização Bayesiano. Algoritmos de Estimação de Distribuição.

ABSTRACT

This paper presents a methodology for the application of Bayesian Networks for solving linear optimization problems. This methodology is known in the literature as Bayesian optimization algorithm and is based on the mathematical modeling process of the berth allocation problem, given the objective function and constraints, followed by modeling to fit the concepts of Bayesian Networks, thus achieving a consistent model. For this some introductory concepts about graph theory and probability theory was presented.

It also performed a study on the estimation of distribution algorithms showing implications for the model of Bayesian optimization algorithms.

The model was tested and compared with other methodologies to ensure that there is a guarantee of this technique for solving related problems. The results obtained showed good agreement with the results of real optimal.

Keywords: Bayesian Networks. Berth Allocation Problem. Bayesian Optimization Algorithms. Estimation of Distribution Algorithms.

LISTA DE FIGURAS

Figura 2.1 - Espaço Amostral	19
Figura 2.2 - Exemplo de grafo não direcionado com 4 nós e 4 arestas	20
Figura 2.3 - Aresta formando um laço num vértice.....	21
Figura 2.4 - Exemplo Grafo com uma cadeia.....	22
Figura 2.5 - Exemplo de grafo com um caminho.....	22
Figura 2.6 – Tipos grafos quanto à conexidade. a) Grafo Conexo. b) Grafo Desconexo. c) Grafo Totalmente Desconexo.....	23
Figura 2.7 - Exemplo de DAG	24
Figura 3.1 - Exemplo de esquema	26
Figura 3.2 - Exemplo de crossover entre dois indivíduos.....	26
Figura 3.3 - Exemplo de mutação	27
Figura 3.4 - Fluxogramas de Algoritmos Evolutivos. a) Fluxograma do AE. b) Fluxograma do EDA.	28
Figura 3.5 - Modelo sem Dependências.....	29
Figura 3.6 - Modelo com Dependência em Pares	30
Figura 3.7 - Modelo com Múltiplas Dependências. a) BOA. b) ECGA.....	31
Figura 3.8 - Um exemplo de Rede Bayesiana binária	33
Figura 3.9 - Etapas de uma iteração no BOA.....	35
Figura 4.1 - Modelagem do PAB	39
Figura 5.1 - Conversão da matriz binária NxL para inteira	45
Figura 5.2 - Transformação do vetor de inteiros para a matriz de NxM	45
Figura 5.3 - Vetor de ocorrências.....	47

LISTA DE TABELAS

Tabela 4.1 - Matriz de Navios (N) X Marés (M)	40
Tabela 4.2 - Matriz Navios (N) X Berços (L).....	43
Tabela 5.1 - Matriz de ganhos para 5 navios	47
Tabela 5.2 - Vetor de probabilidade para um vértice sem pais, no caso de haver 4 berços no problema.....	48
Tabela 5.3 - Tabela de probabilidade para um vértice v_2 que possui o pai v_1 , no caso de haver 4 berços no problema.....	48
Tabela 6.1 - Tabela com resultados comparativos entre o Ótimo Real e o Ótimo obtido pelo BOA	51

SUMÁRIO

1. INTRODUÇÃO	14
1.1. Motivação.....	14
1.2. Objetivos	15
1.3. Organização do Trabalho.....	15
2. FUNDAMENTAÇÃO TEÓRICA	16
2.1. Probabilidade	16
2.1.1. Axiomas e Proposições.....	17
2.1.2. Frequência relativa.....	17
2.1.3. Probabilidade Condicional	18
2.1.4. Eventos Independentes	18
2.1.5. Teorema de Bayes.....	19
2.2. Grafos	20
2.2.1. Adjacência	21
2.2.2. Grau	21
2.2.3. Laço	21
2.2.4. Cadeia.....	22
2.2.5. Caminho.....	22
2.2.6. Conexidade.....	22
2.2.7. Ciclo.....	23
2.2.8. Ordenação Topológica.....	23
3. COMPUTAÇÃO EVOLUTIVA	25
3.1. Algoritmos Evolucionários	25
3.1.1. Esquemas e Blocos de Construção	26
3.1.2. Crossover e Mutação	26
3.2. Algoritmos de Estimação de Distribuição (AED)	27
3.2.1. Modelos de Estimação.....	29
3.3. Algoritmo de Otimização Bayesiano (BOA).....	32
3.3.1. Redes Bayesianas	32

3.3.2. Metodologia do BOA.....	34
3.4. Métrica para avaliação	36
3.4.1. Algoritmo K2	36
4. APRESENTAÇÃO DO PROBLEMA	38
4.1. Berços Homogêneos.....	39
4.2. Berços Heterogêneos.....	41
5. APLICAÇÃO DO BOA PARA O PROBLEMA DE ALOCAÇÃO DE BERÇOS .	44
5.1. Modelagem do Problema	44
5.2. Parâmetros de Entrada da Rede Bayesiana	46
5.3. Construção da Rede	47
5.4. Desenvolvimento da Nova Geração de Indivíduos.....	48
6. RESULTADOS	50
7. CONCLUSÃO.....	52
REFERÊNCIAS.....	53
ANEXO 1	56

1. INTRODUÇÃO

Os portos que operam com navios graneleiros possuem grande importância na competitividade econômica mundial em relação ao fluxo de mercadorias entre nações, uma vez que eles são pontos centrais na manutenção do comércio exterior. Com isso, torna-se cada vez mais necessário o aumento da eficiência das operações portuárias de cada nação, uma vez que valoriza os aspectos político-econômicos, humanos e sociais da mesma.

O aproveitamento de recursos de forma inteligente e rápida vem sendo um dos principais requisitos para o crescimento desse mecanismo de transporte, pois oferece agilidade na administração portuária, uma vez que é mais custoso projetarem os portos de formas diferentes para disporem de recursos acima da demanda exigida (BARROS, 2010).

Surge então a problemática de no contexto de planejamento de operações portuárias, que é conhecido como Problema de Alocação de Berços, representado pela sigla PAB.

1.1. Motivação

Redes Bayesianas se enquadram na área de algoritmos probabilísticos. Elas realizam análises com base em dependências de estados, diminuindo o espaço de busca por inferir dados a partir de soluções promissoras. Isso faz o custo computacional diminuir em relação a um método guloso.

O PAB é considerado um problema que possui um custo computacional muito grande para certos tipos de instâncias. Ele tem como objetivo minimizar o custo total para realizar uma operação de carregamento ou descarregamento, levando em consideração os números de marés, quando e onde o navio deve atracar, e em qual berço o navio deve escolher atracar.

Em relação à aquisição, representação e utilização do conhecimento, as Redes Bayesianas tornam-se uma ótima ferramenta para a resolução de problemas complexos, como é o caso da Alocação de Berços que por sua vez é considerado um problema computacionalmente complexo.

Os algoritmos que utilizam Redes Bayesianas para problemas de otimização são conhecidos como Algoritmos de Otimização Bayesianos.

1.2. Objetivos

Este trabalho tem como objetivo principal propor uma metodologia baseada em Otimização por Redes Bayesianas para a resolução de problemas de otimização linear, tendo como demonstração a resolução do Problema de Alocação de Berços em Portos Graneleiros.

Os objetivos específicos são:

- 1) Modelar um Algoritmo de Otimização Bayesiano de acordo com os requisitos do PAB;
- 2) Construir a Rede Bayesiana com base na métrica K2;
- 3) Validar o Algoritmo de Redes Bayesianas com base em resultados de Ótimos Reais.

1.3. Organização do Trabalho

Este capítulo apresentou os objetivos que foram alcançados e o motivação na utilização de Redes Bayesianas para problemas de otimização linear.

O Capítulo 2 apresenta de forma não aprofundada uma breve base de conhecimento para o entendimento do funcionamento da metodologia proposta. A Seção 2.1 introduz alguns conceitos de probabilidade. A seção 2.2 mostra alguns conceitos de grafos.

O Capítulo 3 traz conceitos sobre Algoritmos de Estimação de Distribuição e suas relações com Algoritmos de Otimização Bayesianos, fazendo comparações com os Algoritmos Genéticos.

O Capítulo 4 trata a respeito do Problema de Alocação de Berços, levando em consideração a função objetivo e as restrições do problema.

O desenvolvimento da metodologia aplicado no PAB é mostrado no Capítulo 5, com base nos conceitos dos Capítulos 3 e 4, além de mostrar como é realizada a adaptação do algoritmo ao problema.

Os resultados são expostos no Capítulo 6, fazendo comparações com os resultados reais e gerando implicações no modelo.

O Capítulo 7 traz a conclusão, resumindo o que foi proposto, propondo melhorias e sugerindo a continuação em trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Em relação aos fenômenos naturais ou artificiais, pode-se analisá-los através de modelos determinísticos ou não determinísticos.

Os modelos determinísticos são modelos matemáticos onde um dado experimento repetido várias vezes sob as mesmas condições terá um resultado fixo e invariável, diferentemente dos não determinísticos, também conhecidos como probabilísticos, onde aparecem resultados distintos.

2.1. Probabilidade

O termo probabilidade se refere ao estudo da aleatoriedade e da incerteza (DEVORE, 2006).

Um experimento é qualquer ação ou processo cujo resultado está sujeito à incerteza (DEVORE, 2006).

Para entender a probabilidade da ocorrência de um fenômeno, devem-se compreender noções básicas sobre teoria das probabilidades.

O espaço amostral, representado por S , equivale ao conjunto de todos os resultados possíveis de um dado experimento. Nesse caso um evento pode ser definido como qualquer subconjunto do espaço amostral S de um experimento aleatório.

Esses espaços amostrais podem ser finitos ou infinitos, como por exemplo:

$$S = \{0, 1, 2, 3, 4, 5, 6\} \text{ (finito)}$$

$$S = \{n: n \text{ é inteiro e } n < 0\} \text{ (infinito)}$$

Pode-se definir probabilidade de um evento N , como uma função definida em um espaço amostral S onde cada evento desse espaço é relacionado a um número real p . Assim:

$$P(N) = p \quad (2.1)$$

onde p equivale à:

$$p = \frac{n_A}{n_S} \quad (2.2)$$

sendo:

- n_A é número de ocorrências do evento A ;

- n_S é total de modos possíveis equiprováveis.

2.1.1. Axiomas e Proposições

Os eventos probabilísticos devem seguir os seguintes axiomas segundo as definições clássicas:

- $P(S) = 1$, onde S é o espaço amostral;
- $0 \leq P(N) \leq 1$, onde N é um evento qualquer no espaço amostral S ;
- Se A e B forem eventos mutuamente exclusivos, ou seja, não podem ocorrer ao mesmo tempo, $(A \cap B) = \emptyset$, então:

$$P(A \cup B) = P(A) + P(B) \quad (2.3)$$

Além dos axiomas acima existem algumas propriedades essenciais:

- $\sum_{i=1}^n P(A_i) = 1$, onde n é número de eventos do espaço amostral e A_i um evento desse espaço;
- $P(\emptyset) = 0$, onde \emptyset (conjunto vazio) representa um evento impossível;
- $P(\bar{A}) = 1 - P(A)$, onde \bar{A} representa o complemento do evento A ;
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$, em que $A \subset S$ e $B \subset S$ (Teorema da Soma).

2.1.2. Frequência relativa

Os axiomas da definição clássica de probabilidade não determinam completamente uma atribuição de probabilidade para eventos (DEVORE, 2006), uma vez que não levam em consideração a quantidade de vezes que um evento é repetido sob as mesmas condições.

Existe outro modelo na literatura que admite que a probabilidade de um evento corresponda à quantidade de ocorrências (frequência) em um número N de repetições. Assim:

$$P(A) = \frac{n_A}{N} \quad (2.4)$$

2.1.3. Probabilidade Condicional

A probabilidade de certo evento ocorrer pode ser influenciada pela probabilidade da ocorrência prévia de outro evento.

Define-se então que a probabilidade de um evento B ocorrer, sabendo que o evento A já aconteceu é chamada de probabilidade condicional B dado A, a qual pode ser representada por $P(B|A)$, denotada pela expressão:

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad (2.5)$$

onde $P(A) > 0$.

A partir da equação (2.5), pode-se deduzir que:

$$P(A \cap B) = P(B|A) \cdot P(A) \quad (2.6)$$

A equação (2.6) é definida como *regra da multiplicação*. Para eventos que possuem mais de duas etapas, pode ser utilizada a seguinte expressão:

$$P(A_1 \cap A_2 \dots \cap A_n) = \prod_{t=1}^n P(x_t | x_1 \cap \dots \cap x_{t-1}) \quad (2.7)$$

2.1.4. Eventos Independentes

A e B são eventos independentes se a ocorrência de A não afeta a ocorrência de B e vice-versa, ou seja:

$$P(A \cap B) = P(A) \cdot P(B) \quad (2.8)$$

Quando dois eventos são independentes, supondo A e B, então:

$$P(A|B) = P(A) \quad (2.9)$$

e

$$P(B|A) = P(B)$$

Considerando que existam três eventos (A, B, C), pode-se dizer que o evento A é condicionalmente independente de B dado o evento C apenas se:

$$P(B|A, C) = P(B|C) \quad (2.10)$$

e

$$P(B,C|A) = P(B|C).P(A|C)$$

Nas equações (2.10), pode-se notar que a ocorrência do evento B não altera a probabilidade da ocorrência do evento A tendo o conhecimento do evento C.

2.1.5. Teorema de Bayes

Dado um espaço amostral S, como mostra a Figura 2.1, pode-se dizer que A_i é uma partição de S se $A_i \subset S$.

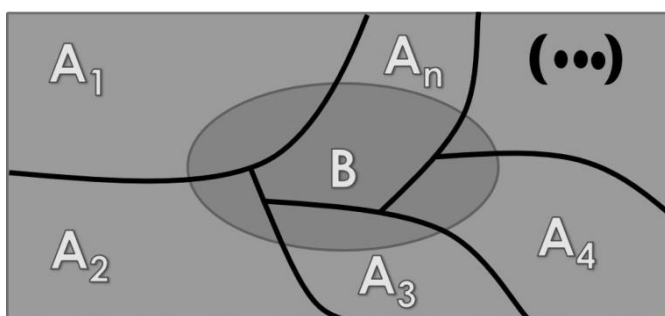


Figura 2.1 - Espaço Amostral

Considerando que A_1, A_2, \dots, A_n são eventos mutuamente exclusivos, de acordo com a equação (2.3), então:

$$S = \bigcup_{i=1}^n A_i$$

Pela Figura 2.1 pode-se também afirmar que $P(B)$ é igual ao somatório das probabilidades de A_i e B ocorrerem, ou seja:

$$P(B) = \sum_{i=1}^n P(A_i \cap B) \quad (2.11)$$

Desenvolvendo a equação (2.11) a partir da equação (2.6) pode-se chegar à conclusão de que:

$$P(B) = \sum_{i=1}^n P(B|A_i).P(A_i) \quad (2.12)$$

A equação (2.12) é conhecida como Teorema da Probabilidade Total.

Utilizando-se a equação (2.5) juntamente com as equações (2.6) e (2.12), é possível formular o Teorema de Bayes definido na equação (2.13).

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{\sum_{i=1}^n P(B|A_i).P(A_i)} \quad (2.13)$$

Essa equação mune-se de um mecanismo que faz um ajuste da probabilidade da ocorrência de um evento através da combinação de novas informações obtidas.

2.2. Grafos

Um grafo pode ser definido como um par formado por um conjunto de vértices (ou nós) e um conjunto de arestas (ou arcos) ,ou seja, $G = (V,A)$ como mostra a Figura 2.2.

Os grafos podem ser **direcionados** (digrafos) ou **não direcionados**.

As arestas são pares ordenados de vértices do conjunto V . As arestas são representadas por (u,v) para grafos não direcionados e $\langle u,v \rangle$ para grafos direcionados.

Os conjuntos de aresta e vértice podem ser representados por $A(G)$ e $V(G)$, respectivamente.

Os grafos têm a capacidade de representar abstratamente através de uma modelagem matemática e gráfica, certos comportamentos de alguns eventos.

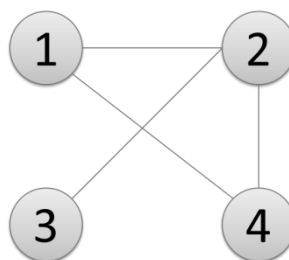


Figura 2.2 - Exemplo de grafo não direcionado com 4 nós e 4 arestas

O estudo na área de grafos é bastante grande. Para utilizar Redes Bayesianas na otimização de problemas torna-se necessário citar alguns conceitos relevantes que serão mostrados a seguir.

2.2.1. Adjacência

Pode-se dizer que dois vértices u e v são adjacentes (ou vizinhos) se existir uma aresta (u,v) em $G = (V,A)$. A aresta que liga esses vértices é chamada de incidente.

2.2.2. Grau

O grau de um vértice é dado pela quantidade de arestas incidentes do mesmo. Considerando que g_i para representar o grau do nó i em um grafo $G = (V,A)$, então:

$$grau = \frac{1}{2} \times \left(\sum_{i=1}^n g_i \right) \quad (2.13)$$

Em relação aos grafos direcionados, faz-se necessário definir algumas noções especializadas de grau para posterior aplicação no BOA, que são o grau de emissão e grau de recepção.

- *grau de emissão*: corresponde ao número de arcos que partem de um vértice v ;
- *grau de recepção*: diz respeito ao número de arcos que chegam em v .

2.2.3. Laço

Um laço pode ser considerado como uma aresta a qual liga um vértice a ele mesmo, ou seja, do tipo (u,u) .

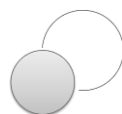


Figura 2.3 - Aresta formando um laço num vértice

2.2.4. Cadeia

Uma cadeia pode ser considerada como qualquer sequência de arestas adjacentes que ligam dois vértices, ou seja, dado um vértice v_1 qualquer no grafo é possível chegar a um vértice v_2 do mesmo grafo através de um conjunto de arestas. Na Figura 2.4, os vértice 1 e 2 são conectados pela cadeia (1, 2, 3).

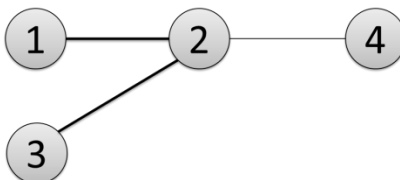


Figura 2.4 - Exemplo Grafo com uma cadeia

2.2.5. Caminho

Um caminho é uma cadeia na qual todas as arestas deste caminho têm a mesma orientação. Esse conceito é aplicado apenas para grafos dirigidos. Nota-se que na Figura 2.5, os vértice 2 e 5 são conectados pelo caminho (2, 4, 3, 5)

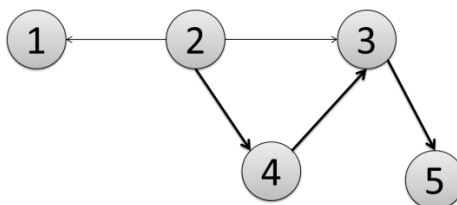


Figura 2.5 - Exemplo de grafo com um caminho

2.2.6. Conexidade

Um grafo $G(V,A)$ é dito conexo quando há ao menos uma cadeia entre cada par de vértices. Caso o contrário ele é considerado desconexo.

Pode-se afirmar que um grafo que não possui arestas é definido como totalmente desconexo.

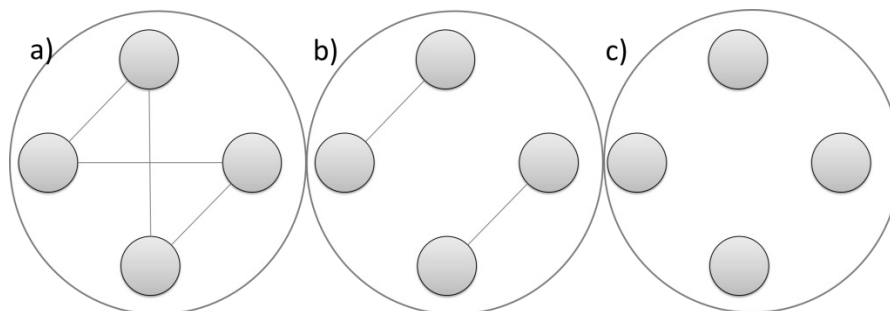


Figura 2.6 – Tipos grafos quanto à conectividade. a) Grafo Conexo. b) Grafo Desconexo. c) Grafo Totalmente Desconexo.

2.2.7. Ciclo

Um ciclo é uma cadeia fechada, ou seja, é uma cadeia onde o vértice inicial é igual a vértice de destino.

A estrutura cíclica de um grafo é um fator muito importante na implementação das Redes Bayesianas, pois elas são vistas como um DAG (Grafo Dirigido Acíclico), as quais serão mostradas posteriormente.

2.2.8. Ordenação Topológica

Uma ordenação topológica de grafo $G = (V,A)$ consiste em ordenar linearmente todos os vértices de um grafo dirigido acíclico de forma que para toda aresta do tipo $\langle u, v \rangle$, têm-se que v aparece antes de u na ordenação (CORMEN, 2002).

Grafos dirigidos com ciclos não admitem ordenação topológica, pois não é possível ordenar linearmente.

Pode-se citar um exemplo simples da execução de um conjunto de tarefas a serem cumpridas onde existem dependências entre essas tarefas, em outras palavras uma tarefa X depende de uma tarefa Y para ser realizada. Supondo que cada tarefa corresponda a um vértice de um grafo, é possível montar um DAG do problema, cujas arestas representam as dependências. O problema seria definir qual é a melhor ordem para executar essas tarefas levando em consideração as relações de dependências entre elas.

O problema citado anteriormente mostra de forma simples uma das aplicações da ordenação topológica de um grafo.

Um algoritmo simples para ordenação topológica:

Algoritmo 1 – Pseudocódigo de Ordenação Topológica

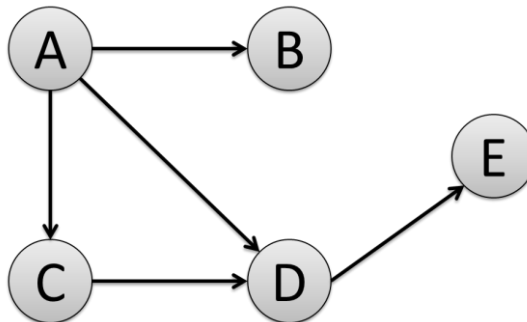
Ordenação_Topológica (GRAFO)

Inicializa a lista SAÍDA como vazia

Enquanto existir vértice em GRAFO **faça** Verificar o vértice u que tem o menor grau de entrada Inserir o u na lista SAÍDA Excluir vértice u de GRAFO

Na prática é possível encontrar diferentes ordenações topológicas corretas num DAG. Isso depende de qual vértice inicia a ordenação. Deve-se iniciar sempre do vértice que não possui predecessores.

Observa-se que na Figura 2.7 que ordenação pode começar tanto pelo vértice B como pelo vértice E.

**Figura 2.7 - Exemplo de DAG**

Algumas possíveis ordenações para o grafo da Figura 2.7 são:

- A, C, D, B, E
- A, C, D, E, B

Dada um grafo $G = (V, A)$, a ordenação topológica realiza uma busca em profundidade que demora $\Theta(V+A)$, juntamente com a inserção de um vértice na lista que é $O(1)$, logo a complexidade do tempo total gasto para as inserções é de $O(V)$. Com isso a complexidade da ordenação topológica é $\Theta(V+A)$.

3. COMPUTAÇÃO EVOLUTIVA

A Computação Evolutiva é um ramo da computação que utiliza metodologias evolutivas (baseadas na teoria de Darwin) (DARWIN, 1859) para o desenvolvimento de algoritmos que solucionem problemas complexos.

Pode-se estudar os algoritmos evolutivos a partir de duas abordagens:

- Algoritmos Evolucionários;
- Algoritmos de Estimação de Distribuição.

3.1. Algoritmos Evolucionários

Algoritmos Evolucionários (AE) estão preocupados com a investigação computacional de sistemas que se assemelham a versões simplificadas dos processos e mecanismos de evolução para atingir os efeitos desses processos e mecanismos. Os algoritmos que se enquadram dentro da área da Computação Evolucionária são os que buscam explorar as propriedades dos campos relacionados de Genética de Populações, Ecologia Populacional, Biologia Coevolutiva e Biologia do Desenvolvimento. (BROWNLEE, 2012)

Fazendo um paralelo com a Teoria da Evolução de Darwin (DARWIN, 1859), os indivíduos da população e as variáveis ou características do problema equivalem respectivamente aos cromossomos e aos genes. Cada cromossomo é composto por um conjunto de genes. O conjunto de cromossomos (indivíduos) formam uma população. Os genes podem ser binários ou inteiros ou compostos por números reais.

Pode-se citar alguns exemplos de algoritmos desse tipo (BROWNLEE, 2012):

- Algoritmos Genéticos;
- Programação Genética;
- Evolução Diferencial;
- Evolução Gramatical;
- entre outros.

Os AE's utilizam como operadores para a geração de novos indivíduos os mecanismos de Mutação e Crossover. Para compreender o funcionamento desses

operadores deve-se definir primeiramente os conceitos de esquema e bloco de construção (BB – *Building Blocks* em inglês).

3.1.1. Esquemas e Blocos de Construção

Pode-se definir um esquema como um subconjunto de genes que se repetem em algumas posições de alguns cromossomos, como é mostrado na Figura 3.1.

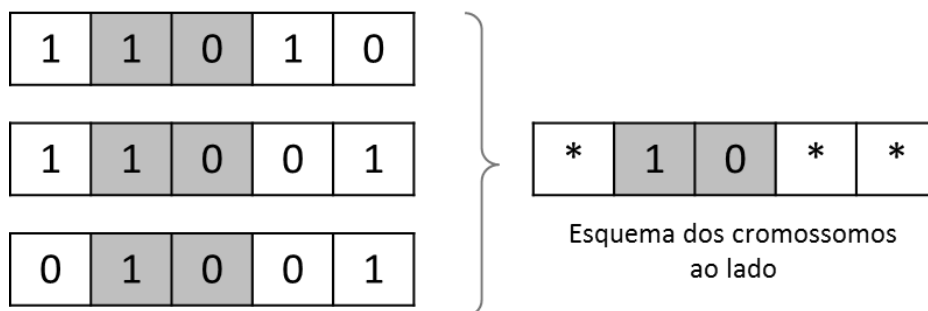


Figura 3.1 - Exemplo de esquema

Um bloco de construção pode ser definido como um esquema que representa uma instância da partição de um problema (CROCOMO, 2012). Por exemplo, se considerasse como ponto de partição (corte) as três primeiras variáveis do cromossomo 11010, então o BB dele seria 110**.

3.1.2. Crossover e Mutação

A operação de *crossover* consiste na recombinação de genes de dois ou mais indivíduos com base em um ou mais pontos de cortes, como ilustra a Figura 3.2 para gerar um novo indivíduo.

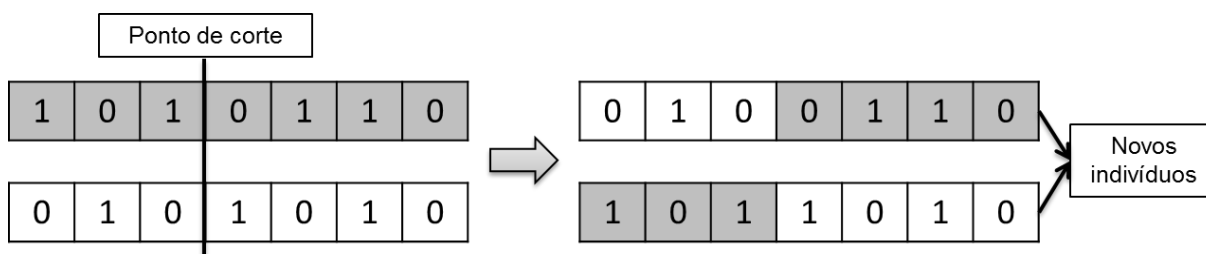


Figura 3.2 - Exemplo de crossover entre dois indivíduos

A realização do crossover pode ser feita de várias formas, sendo que as mais comuns são:

- Crossover de um ponto de corte, o qual é definido um ponto de corte;

- Crossover de múltiplos pontos de corte, onde é definido mais de um ponto de corte;
- Crossover uniforme, onde os genes do novo indivíduo são gerados a partir da seleção probabilística de cada gene dos indivíduos selecionados.

O operador de mutação é responsável por alterar um ou mais genes do indivíduo selecionado para a geração de um ou mais novos indivíduos, como mostra a Figura 3.3.

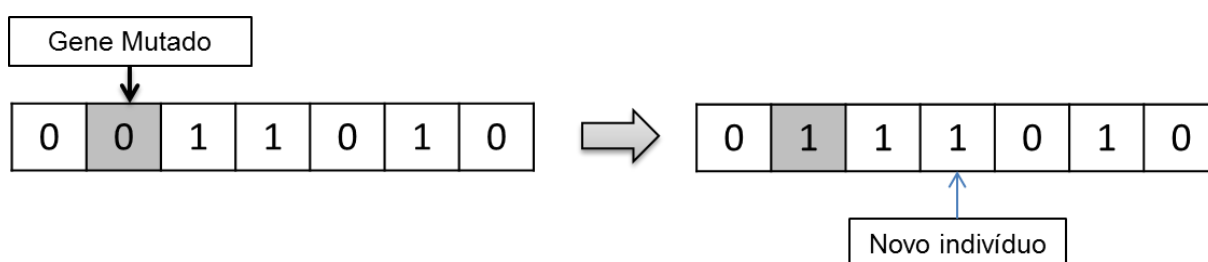


Figura 3.3 - Exemplo de mutação

A alteração de cada gene pela mutação é baseada em um valor de probabilidade que é conhecida como taxa de mutação. Essa taxa inverte o valor de um bit (gene), no caso do cromossomo ser binário, ou realiza uma distribuição de probabilidade no caso do cromossomo ser inteiro ou real.

3.2. Algoritmos de Estimação de Distribuição (AED)

Em inglês *Estimation of Distribution Algorithm* (EDA), também conhecidos como *Population Model-Building Genetic Algorithms* (PMBGA), é uma extensão da Computação Evolucionária que modela uma população de soluções candidatas através de um modelo probabilístico (BROWNLEE, 2011).

Um dos problemas que os Algoritmos Evolucionários enfrentam é a questão de analisar as dependências entre variáveis. Caso fosse possível obter essas informações, seria possível de imediato saber qual é o melhor ponto de corte para a realização dos operadores genéticos como mutação e crossover. A cada nova iteração para encontrar as melhores soluções, a falta desses dados poderia implicar na quebra de blocos promissores – blocos que teriam grande probabilidade de estarem no ótimo global ou que conduziriam para encontrá-lo – do cromossomo, dificultando em alguns casos a convergência dos indivíduos e a busca da melhor

solução no caso do tamanho do espaço de busca seja grande, ou seja, o cromossomo tenha um grande número de genes. Na literatura isso é conhecido como *linkage problem*.

Para resolver esse problema o EDA utiliza maneiras de representar a dependências entre as variáveis a partir de uma distribuição de probabilidade, em vez de operadores genéticos.

A estrutura do EDA é bastante parecida com a do AG como mostra Figura 3.4.

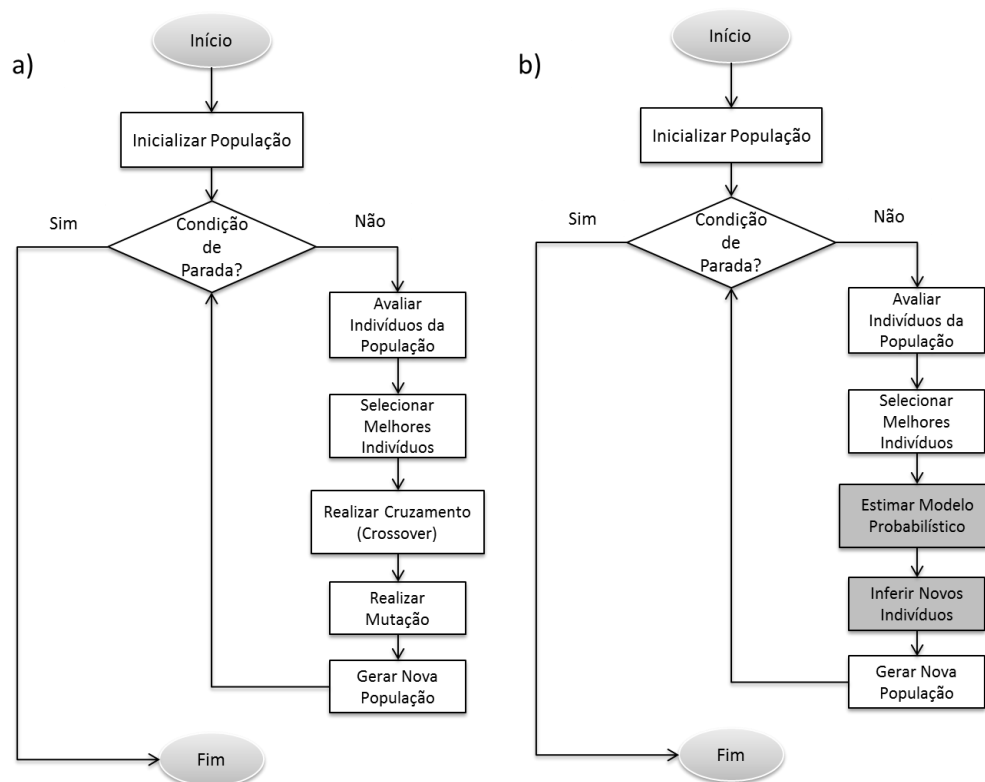


Figura 3.4 - Fluxogramas de Algoritmos Evolutivos. a) Fluxograma do AE. b) Fluxograma do EDA.

Os algoritmos de Estimação de Distribuição estimam a distribuição de um determinado conjunto de soluções candidatas para à partir da população estimar a geração de novos indivíduos.

Enquanto nos Algoritmos Genéticos as relações de dependências são colocadas de forma implícita, os EDA's explicitam essas relações, aumentando o grau de exploração das regiões próximas a um indivíduo.

Estimar soluções promissoras não é uma tarefa simples. Essa etapa leva em consideração o custo computacional e a precisão dos parâmetros. Para alguns casos o custo computacional pode ser muito elevado para encontrar uma precisão relevante.

3.2.1. Modelos de Estimação

De acordo com a complexidade do modelo probabilístico existem diferentes formas de estimar parâmetros (GONÇALVES, 2011), os principais são:

- Modelos sem dependência;
- Modelos com dependências em pares;
- Modelos com múltiplas dependências;
- Modelos de mistura.

➤ Modelo sem dependências

Esse modelo considera que não existam dependências entre as variáveis, ou seja as variáveis do problema são independentes.

Baseado nessa informação a distribuição de probabilidade conjunta mostrada na equação (3.1) pode ser desenvolvida a partir da equação (2.8).

$$P(A) = \prod_{i=1}^n p(A_i) \quad (3.1)$$

onde n representa o número de variáveis.

Esse modelo pode ser representado na Figura 3.5, através de um grafo sem arestas.

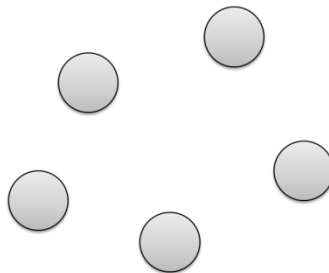


Figura 3.5 - Modelo sem Dependências

Algoritmos que usam esse modelo:

- *Population Based Incremental Learning* (PBIL);
- *Univariate Marginal Distribution Algorithm* (UMDA);
- *Stochastic Hill Climbing with Learning by Vectors of Normal Distributions* (SHCLVND);
- *Reinforcement Learning Estimation of Distribution Algorithm* (RELEDA);
- *Compact Genetic Algorithm* (CGA).

➤ **Modelo com dependências em pares**

As variáveis possuem dependências apenas com os pais, formando pares.

A distribuição de probabilidade conjunta mostrada na equação (3.2) pode ser deduzida a partir da equação (2.6).

$$P(A) = \prod_{i=1}^n p(A_i | A_{j(i)}) \quad (3.2)$$

Esse modelo pode ser representado na Figura 3.6, através de um grafo direcionado acíclico.

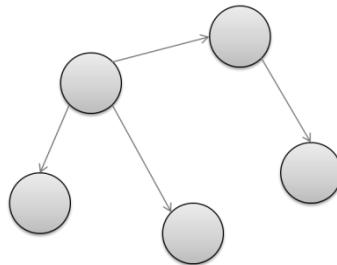


Figura 3.6 - Modelo com Dependência em Pares

Algoritmos que usam esse modelo:

- *Combining Optimizers with Mutual Information Trees* (COMIT);
- *Mutual Information Maximizing Input Clustering* (MIMIC);
- *Bivariate Marginal Distribution Algorithm* (BMDA).

➤ **Modelo com múltiplas dependências**

Todas as dependências são consideradas no modelo.

A distribuição de probabilidade conjunta mostrada na equação (3.3) pode ser deduzida a partir da equação (2.7).

$$P(A) = \prod_{i=1}^n p(A_i | A_1, A_2, \dots, A_{i-1}) \quad (3.3)$$

Esse modelo pode ser representado de duas formas diferentes na Figura 3.7, representando dois algoritmos distintos respectivamente.

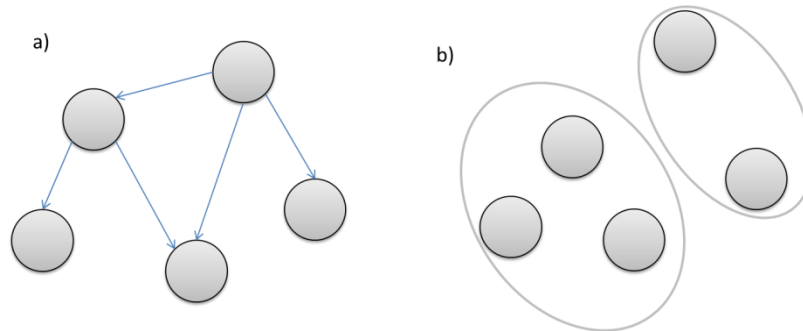


Figura 3.7 - Modelo com Múltiplas Dependências. a) BOA. b) ECGA.

Algoritmos que usam esse modelo:

- *Bayesian Optimization Algorithm* (BOA);
- *Hierarchical Bayesian Optimization Algorithm* (hBOA);
- *Extended Compact Genetic Algorithm* (ECGA);
- *Estimation of Bayesian Network Algorithm* (EBNA).

➤ **Modelo de mistura**

São modelos flexíveis para problemas multimodais. Agrupam soluções candidatas para deduzir a distribuições de probabilidades para cada grupo.

A distribuição de probabilidade conjunta é mostrada na equação (3.4).

$$P(A) = \prod_{i=1}^n \pi_i p(A_i) \quad (3.4)$$

onde π_i representa o peso do componente i da estrutura.

O peso de cada componente são obtidos através de um procedimento dedicado, como por exemplo o algoritmo EM (*Expectation-Maximization*) (DEMPSTER & LAIRD & RUBIN, 1977).

Algoritmos que usam esse modelo:

- *Adaptative Gaussian Mixture* (AMix);

- *Estimation of Mixtures of Distributions Algorithm* (EMDA).

3.3. Algoritmo de Otimização Bayesiano (BOA)

O Algoritmo de Otimização Bayesiano (BOA) tem como idéia principal a utilização de Redes Bayesianas para representação de informações levando em consideração as dependências entre variáveis. Cada vértice do grafo representa uma variável do problema, enquanto as arestas representam o grau de dependência entre os valores dessas variáveis.

3.3.1. Redes Bayesianas

“A principal vantagem de raciocínio probabilístico sobre raciocínio lógico é fato de que agentes podem tomar decisões racionais mesmo quando não existe informação suficiente para se provar que uma ação funcionará” (RUSSEL & NORVIG, 2004).

Redes Bayesianas (RB) têm a capacidade de analisar grandes quantidades de dados para extrair conhecimentos úteis em tomada de decisões, controlar ou prever o comportamento de um sistema, diagnosticar as causas de um fenômeno, entre outras. Com isso ela é utilizada em várias áreas, dentre elas a área médica, industrial, em sistemas inteligentes; marketing, econômica, de finanças, de gestão. (LUNA, 2004)

De acordo com o tipo de aplicação, a utilização prática de uma rede bayesiana pode ser parecida com a forma de aplicar uma rede neural. Naturalmente, na escolha do método sucede diferentes critérios, como a facilidade, o custo e o tempo necessário na implantação de uma solução. (LUNA, 2004)

Uma Rede Bayesiana consiste das seguintes propriedades:

- Um conjunto de variáveis e um conjunto de dependências (arcos) ligando as variáveis;
- A rede é um grafo dirigido sem ciclos (DAG);
- Para cada variável existe uma tabela de probabilidade composta por fatores de dependências de suas adjacências num grafo;
- Cada variável possui um conjunto limitado de estados mutuamente exclusivos. Elas podem ser discretas ou contínuas. Em relação aos valores discretos podem subdividir-se em binários e inteiros.

Formalmente uma Rede Bayesiana pode ser definida como um par (G,P) , onde G é um DAG, $P = \{p(x_1|p_1), \dots, p(x_n|p_n)\}$ é um conjunto de n Distribuições de Probabilidade Condicional, uma para cada variável, e π_i é o conjunto de pais do nó x_i em G . Essa distribuição pode ser deduzida da equação (3.3) como:

$$p(x) = \prod_{i=1}^n p(x_i|\pi_i) \quad (3.4)$$

A Figura 3.8 mostra um exemplo de rede Bayesiana binária onde cada variável aceita os valores *true* (T) ou *false* (F).

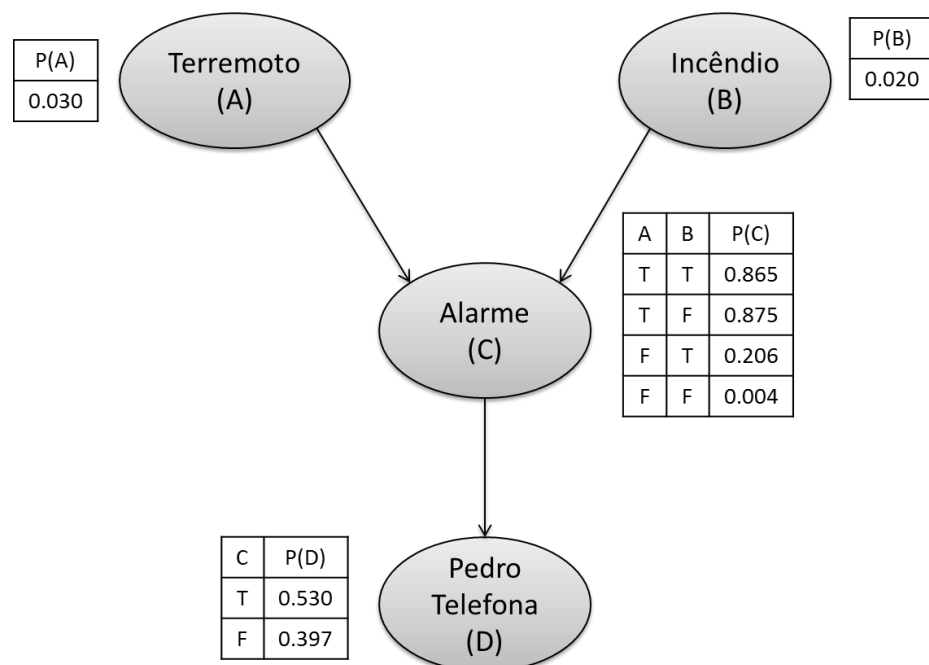


Figura 3.8 - Um exemplo de Rede Bayesiana binária

A Rede Bayesiana tem a capacidade de analisar a influência probabilística não direta de uma variável para as demais da rede. Através dela é possível obter também uma representação gráfica das variáveis de uma tabela de probabilidade que compõe um determinado problema.

Essa representação, por ser baseada nos princípios de Bayes mostrado também na equação (2.12), mune de adaptabilidade, ou seja, com o advento novas informações implicarão alterações nas dependências, ou seja, os parâmetros das relações de dependências entre as redes podem se alterar devido a uma nova

aquisição de dados formando novas implicações na inferência de indivíduos, no caso do BOA.

3.3.2. Metodologia do BOA

O BOA deve ter a capacidade de aprender uma rede que esteja de acordo com as dependências e independências que compõe o problema respectivamente (PELIKAN, 2005). Para isso ele realiza duas outras tarefas:

1. *Aprendizado da estrutura*: é composto por um grafo acíclico dirigido (DAG) conexo, no qual os vértices correspondem às variáveis do modelo e as arestas às dependências condicionais entre os vértices. Está relacionado às dependências qualitativas entre os nós;
2. *Aprendizado das probabilidades condicionais*: correspondem ao conjunto de tabelas de probabilidades condicionais específico para cada variável e suas dependências. Está relacionado aos aspectos quantitativos entre os nós.

Na maioria das vezes não existe o conhecimento prévio da estrutura de dos parâmetros da rede. Para resolver esse problema normalmente propõe-se automatizar o aprendizado de rede através de soluções candidatas.

Uma população de indivíduos é inicialmente criada, onde cada indivíduo representa uma solução e respectivamente uma RB. Isso pode ser visto como uma amostra de dados, os quais podem ser modelados por uma função de distribuição de probabilidades. As relações entre variáveis presentes nas soluções são modeladas por essa função que mostra em cada passo as soluções mais promissoras. Com isso pode-se desenvolver novos indivíduos inicialmente com valores de *fitness* altos, no caso de maximização, ou vice-versa no caso de minimização.

O *fitness* é o valor que avalia um indivíduo (uma solução) com base na função objetivo e as restrições. Em português é conhecido como “aptidão”.

Observa-se então que esse modelo probabilístico permite gerar indivíduos com grande probabilidade de terem baixos valores de *fitness*, no caso de minimização da função objetivo. A geração desses novos indivíduos é um dos princípios fundamentais para os algoritmos baseados em estimação de distribuição.

Basicamente o pseudocódigo do BOA é mostrado a seguir:

Algoritmo 2 – Pseudocódigo Algoritmo de Otimização Bayesiano

Inicializa População

Avalia cada indivíduo da população

MelhorIndivíduo = SeleccionamelhorIndivíduo(População)

Enquanto não atingir condição de parada **faça**

Seleciona um amostra da população

Constrói uma Rede Bayesiana da amostra

Estima as Dependências da Rede Bayesiana

Gera novos indivíduos

Avalia cada novo indivíduo da população

Adiciona os novos indivíduos na população

MelhorIndivíduo = SeleccionamelhorIndivíduo(NovaPopulação)

Pode-se resumir o comportamento do algoritmo através da Figura 3.9.

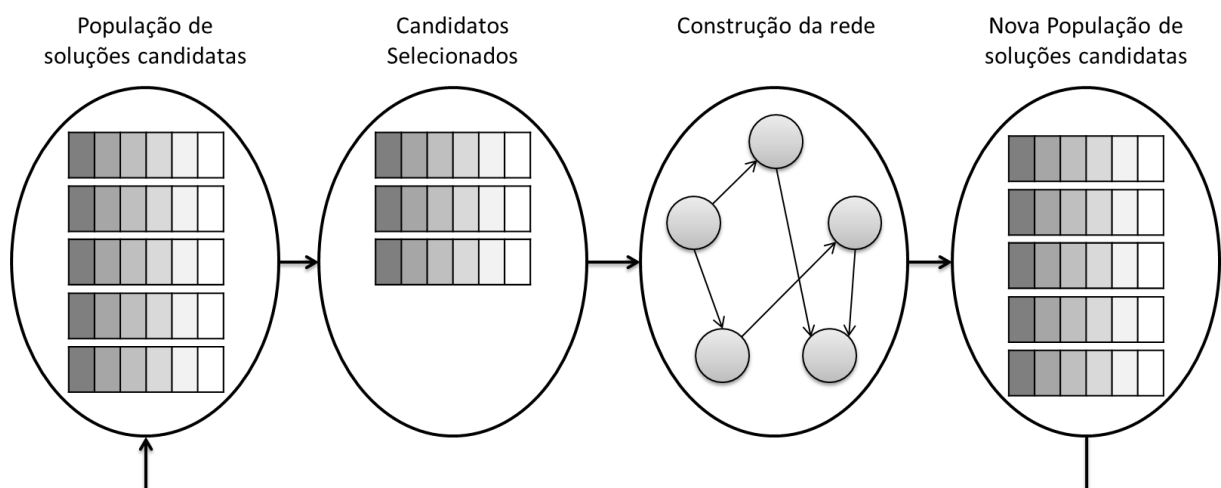


Figura 3.9 - Etapas de uma iteração no BOA.

Para analisar a qualidade com que a Rede Bayesiana é criada, usa-se normalmente um algoritmo para pontuar matematicamente este valor. Esses

algoritmos são geralmente baseados em cima de uma expressão matemática o qual se denomina na literatura como métrica bayesiana.

3.4. Métrica para avaliação

A métrica de pontuação da rede avalia a qualidade da disposição de dependências entre as variáveis na qual modele melhor os dados de um determinado problema.

Encontrar a melhor estrutura da Rede Bayesiana para um conjunto de variáveis é considerado um problema NP (SANTOS 2007).

O tempo computacional nesse caso é levado em consideração, pois a busca por uma rede ideal não é tão simples assim, pois demanda um custo computacional muito grande caso haja a necessidade de encontrar uma rede bem modelada (com todas as arestas possíveis sem formar ciclos).

Para contornar esse problema normalmente deve-se escolher bem a métrica que será aplicada e tentar balancear o custo com a qualidade da rede.

Existem várias métrica para análise de uma RB. Entre elas pode-se citar:

- Algoritmo K2 (COOPER & HERSKOVITS, 1992);
- Algoritmo B (BUNTIME, 1991);
- Algoritmo PC (SPIRITES, 1991).

A métrica utilizada nesse trabalho será a métrica K2, uma vez que ela não depende de informação a priori para construir a Rede Bayesiana no BOA, ou seja não existe a informação de quais arestas (relações de dependências) existem na rede inicialmente.

3.4.1. Algoritmo K2

Essa métrica (COOPER & HERSKOVITS, 1992) realiza uma busca gulosa na rede para maximizar o valor de $P(B_s, D)$. O valor de $P(B_s, D)$ na equação (3.5) representa uma pontuação de uma determinada aresta. Essa busca gulosa é feita a partir de um grafo inicialmente sem aresta. Na medida em que são calculadas as pontuações, as arestas que possuem pontuações relevantes são adicionadas na rede. Isso permite analisar a verossimilhança em relação aos dados, fazendo com que adquira incluir conhecimento a priori sobre a estrutura e os parâmetros condicionais.

$$P(B_s, D) = P(B_s) \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (3.5)$$

Considerando X_i a variável do vértice i pode-se dizer que $P(B_s)$ é um valor de informação prévia da sobre a qualidade da rede, que na maioria das vezes assume o valor 1. O n representa a quantidade de variáveis. O valor de q_i indica o número de combinações de uma variável X_i em relação às suas adjacências. O valor r_i diz respeito ao número de possíveis valores associados a uma variável X_i . O valor de N_{ijk} é o número de casos em D em que X_i assume o k -ésimo valor possível em relação ao j -ésima possível instância para o conjunto de pais dessa variável. E por último o N_{ij} que representa o número de ocorrências que possuem o conjunto de pais de X_i .

Para saber a contribuição de um vértice i da rede no valor da pontuação $P(B_s, D)$ usa-se a equação (3.6) (COOPER & HERSKOVITS, 1992).

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (3.6)$$

Onde π_i representa o conjunto das variáveis pais do vértice i .

4. APRESENTAÇÃO DO PROBLEMA

O Problema de Alocação de Berços (PAB) trata do problema de transferir matéria-prima (bauxita, manganês, ferro e alumina) através de navios graneleiros, levando em consideração a quantidade de marés para realizar a operação, a quantidade de estoque por janela de maré e o número de berços para a realização da operação de carregamento/descarregamento.

Em relação ao PAB, pode-se considerar dois modelos matemáticos de acordo com as características dos berços. Eles são:

- **Modelo de Berços Homogêneos:** nessa modelagem os berços possuem as mesmas capacidades de carregamento ou descarregamento.
- **Modelo de Berços Heterogêneos:** nessa modelagem os berços possuem diferentes capacidades de carregamento ou descarregamento, uma vez que é levado em consideração as diferentes infraestruturas dos berços.

O cenário com berços heterogêneos é claramente o mais próximo da realidade, sendo o respectivo modelo evoluído do primeiro (BARROS, 2010).

O horizonte de planejamento nas operações com navios pode ser dividido em $|M|$ janelas de tempo (BARROS, 2010), devido aos pequenos intervalos para a realização do atracamento em relação aos navios de marés.

Para representar as janelas de tempo, a literatura recomenda usar o acrônimo em inglês TTW (*Tidal Time Window*).

Considerando que N seja o número de navios, M o número de TTW's (janelas de marés favoráveis) e L o número de berços, e admitindo que o PAB pode ser classificado como um problema de transporte, pode-se ilustrá-lo com a Figura 4.1.

Em alguns casos também é tratado a questão de prioridade e defeitos imprevistos, como por exemplo, defeito no motor do navio.

No modelo dos berços homogêneos é levado em consideração o *demurrage* representado pela variável d_i , que nada mais é do que o valor de multa por sobrestadia, ou seja, o tempo que o navio permanece em serviço além daquele definido em contrato (BARROS, 2010).

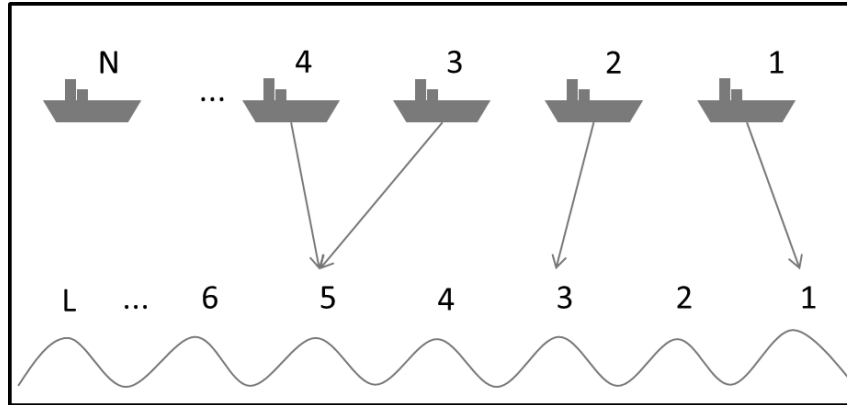


Figura 4.1 - Modelagem do PAB

4.1. Berços Homogêneos

No modelo de berços homogêneos a função objetivo é dada por:

$$\min \sum_{i=1}^N \sum_{j=1}^M j \frac{d_i}{h_i} x_{ij} + \sum_{i=1}^N d_i \left(\frac{1}{2} h_i - a_i - t_i + \frac{1}{2} \right) \quad (4.1)$$

e as restrições são:

$$\sum_{j=1}^{a_i-1} x_{ij} = 0, \quad \forall i \in N \quad (4.2)$$

$$\sum_{j=a_i}^M x_{ij} = h_i, \quad \forall i \in N \quad (4.3)$$

$$\sum_{i=1}^N x_{ij} < |L|, \quad \forall j \in M \quad (4.4)$$

$$\sum_{l=1}^{j-1} x_{il} - j x_{ij-1} + j x_{ij} \leq j, \quad \forall i \in N, \forall j > 1 \in M \quad (4.5)$$

$$\sum_{i=1}^N \sum_{l=1}^j \frac{q_{ik}}{h_i} x_{il} \geq j w_k - e_k, \quad \forall j \in M, \forall k \in K \quad (4.6)$$

onde:

- N – número total de navios;

- M – número total de marés disponíveis;
- L – número de berços
- K – quantidade total de obras-primas operadas no porto;
- a_i – TTW de chegada do navio i ;
- h_i – tempo de tratamento do navio i ;
- d_i – *demurrage* para o navio i ;
- e_k – quantidade de estoque inicial da carga de granel k ;
- c_k – quantidade de consumo ou produção da carga de granel k ;
- q_{ik} – quantidade de carga do navio i em relação a carga k .

A variável de decisão é o x_{ij} , que pode assumir o valor 1 se o navio i foi alocado na TTW j ou 0 caso contrário.

A equação (4.2) diz respeito que os navios não devem atracar antes da maré de chegada.

A equação (4.3) coloca que o navio deve permanecer carregando ou descarregando em h_i TTW's até terminar a operação.

A equação (4.4) impõe que o número de navios alocados não exceda o total de posições de atracação.

A equação (4.5) previne que haja interrupção da operação de carregamento ou descarregamento durante o atendimento nos berços. Essa expressão baseia-se em modelar o problema como se fosse uma matriz binária de navios X TTW's, na qual cada posição da matriz é representada por uma variável x_{ij} . A duração da operação do navio i é representado como 1's consecutivos como mostra a Figura 4.2.

N\M	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	1	1	1	1	0	0	0	0	0	0
2	0	0	0	0	1	1	1	1	0	0	0	0
3	0	0	0	0	0	0	1	1	1	0	0	0
4	0	0	0	0	0	0	0	0	0	1	1	0

Tabela 4.1 - Matriz de Navios (N) X Marés (M)

Em uma instância com N navios e M marés, tem-se a quantidade de variáveis igual à $N \times M$, ou seja, se existem 10 navios e 20 marés, então há $10 \times 20 = 200$ variáveis no problema.

A equação (4.6) diz respeito aos níveis de estoque, na qual não permite que haja sequências de atracções nas quais o nível de qualquer carga manipulada atinja um valor abaixo de um determinado limiar pré-definido (BARROS, 2010).

4.2. Berços Heterogêneos

O modelo de Berços heterogêneos pode ser considerado uma evolução do modelo de berços homogêneos.

Nessa abordagem a função objetivo torna-se:

$$\min \sum_{i=1}^N \sum_{j=a_i}^M \sum_{l=1}^L \left| \frac{j - a_i + 1}{h_{ij}} \right| \times y_{ijl} \quad (4.7)$$

cujas restrições são:

$$\sum_{j=1}^{a_i-1} x_{ij} = 0, \quad \forall i \in N \quad (4.8)$$

$$\sum_{j=a_i}^M x_{ij} = \sum_{l=1}^L h_{il} \times u_{il}, \quad \forall i \in N \quad (4.9)$$

$$\sum_{i=1}^N y_{ijl} \leq 1, \quad \forall j \in M, l \in L \quad (4.10)$$

$$\sum_{l=1}^L u_{il} = 1, \quad \forall i \in N \quad (4.11)$$

$$\sum_{z=1}^{j-1} x_{iz} - j \times x_{ij-1} + j \times x_{ij} \leq j, \quad \forall i \in N, \forall j > a_i \in M \quad (4.12)$$

$$\sum_{i=1}^N \sum_{z=1}^j \sum_{l=1}^L \frac{q_{ik}}{h_i} \times y_{izl} \geq j \times w_k - e_k, \quad \forall j \in M, k \in K \quad (4.13)$$

onde:

- N – número total de navios;
- M – número total de marés disponíveis;
- L – número de berços
- K – quantidade total de obras-primas operadas no porto;
- a_i – TTW de chegada do navio i ;
- v_l – tempo de tratamento do navio i ;
- e_k – quantidade de estoque inicial da carga de granel k ;
- c_k – quantidade de consumo ou produção da carga de granel k ;
- q_{ik} – quantidade de carga do navio i em relação a carga k .
- h_{il} – tempo de carregamento/descarregamento do navio i num berço l .

A equação (4.8) é equivalente a equação (4.2), ou seja, os navios não devem atracar antes da maré de chegada.

A equação (4.9) coloca que o navio deve permanecer carregando ou descarregando em h_i TTW's até terminar a operação.

A equação (4.10) impõe que o número de navios alocados não exceda o total de posições de atracação.

A equação (4.11) impõe que o número de navios alocados não exceda o total de posições de atracação.

A equação (4.13) é uma evolução da equação (4.6), onde garante os níveis de estoque maiores que zero.

O valor h_{il} pode ser calculado a partir da expressão:

$$h_{il} = \left\lceil \frac{\sum_{k=1}^{|k|} q_{ik}}{v_l} \right\rceil \quad (4.14)$$

Neste modelo existem três variáveis de decisão que são: x_{ij} , u_{il} e y_{ijl} . A variável y_{ijl} é formada a partir de associação das variáveis x_{ij} e u_{il} baseado na técnica de contingência de variáveis, com a finalidade de tratar o modelo do problema de forma linear. Para isso serão descritas mais algumas equações:

$$y_{ijl} \geq x_{il} + u_{il} - 1, \quad \forall i \in N, j \in M, l \in L \quad (4.15)$$

$$y_{ijl} \leq x_{ij}, \quad \forall i \in N, j \in M, l \in L \quad (4.16)$$

$$y_{ijl} \leq u_{il}, \quad \forall i \in N, j \in M, l \in L \quad (4.17)$$

A variável de decisão é o x_{ij} assumir o valor 1 se o navio i foi alocado na TTW j ou 0 caso contrário.

A variável de decisão é o u_{il} assumir o valor 1 se o navio i foi alocado no berço l ou 0 caso contrário.

Note que a variável y_{ijl} pode assumir o valor 1 ou 0, baseado nas equações (4.15) a (4.17). Ela indica que se o navio i utilizou no berço l na TTW j .

No modelo heterogêneo além da matriz de navios X TTW's, existe a matriz de navios X berços uma vez que é levado em consideração berço em que o navio atraca.

N\L	1	2	3
1	0	0	1
2	0	1	0
3	1	0	0
4	0	1	0
5	0	1	0
6	0	0	1

Tabela 4.2 - Matriz Navios (N) X Berços (L)

A Tabela 4.2 mostra um exemplo de configuração da matriz de navios e berços. Nota-se que para cada linha da matriz deve haver no máximo um berço alocado, considerando a equação (4.11).

5. APLICAÇÃO DO BOA PARA O PROBLEMA DE ALOCAÇÃO DE BERÇOS

Para que seja possível adaptar o BOA às especificações do problema, é necessário primeiramente escolher o tipo de modelo do PAB.

O modelo a ser analisado é o dos berços heterogêneos por ser mais próximo da realidade.

5.1. Modelagem do Problema

Primeiramente é necessário definir que tipo de variáveis a rede bayesiana trabalhará. De acordo com o problema, as variáveis seriam binárias, todavia o custo computacional seria muito grande para encontrar uma simples solução, uma vez que o número de variáveis (n_v) seria para o modelo de berços heterogêneos:

$$n_v = N \times M \times L \quad (5.1)$$

De acordo com a equação (4.22) se a instância tivesse 10 navios, 15 TTW's e 4 berços, o número de variáveis seria 600, e conseqüentemente a Rede Bayesiana teria 600 nós, considerando que essa é uma instância pequena.

Para solucionar isso se optou por considerar apenas a matriz de navios X berços mostrada na Tabela 4.2. A partir dessa tabela, é criado um vetor de variáveis discretas inteiras. Para realizar essa adaptação, é preciso garantir que, na inicialização dos parâmetros, os indivíduos gerados não infrinjam as equações (4.9), (4.11) e (4.12).

Nesse modelo o número de estados para cada vértice na Rede Bayesiana seria igual ao número de berços, como é mostrado a Figura 5.1:

$N \setminus L$	1	2	3	
1	0	0	1	3
2	0	1	0	2
3	1	0	0	1
4	0	1	0	2
5	0	1	0	2
6	0	0	1	3

Figura 5.1 - Conversão da matriz binária $N \times L$ para inteira

Essa adaptação faz com que o valor n_v seja igual à N . Com isso, o número de vértice da RB passaria a ser igual ao número de navios.

Ao adotar essa metodologia, deve-se fazer algumas considerações sobre o cálculo das probabilidades do algoritmo como será mostrado na Seção 5.4.

As restrições consideradas no problema seriam apenas as equações (4.8), (4.10) e (4.13) do problema, contudo elas não seriam alteradas totalmente de imediato, pois é possível converter esse novo vetor em uma matriz navio \times TTW para encontrar o x_{ij} , como mostra a Figura 5.2.

$N \setminus M$	1	2	3	4	5	6	7	8	9	10	11	
1	1	1	0	0	0	0	0	0	0	0	0	3
2	0	1	1	1	1	0	0	0	0	0	0	2
3	0	1	1	1	0	0	0	0	0	0	0	1
4	0	0	0	0	0	1	1	0	0	0	0	2
5	0	0	0	0	0	0	0	1	1	1	0	2
6	0	0	0	1	1	1	1	0	0	0	0	3

Figura 5.2 - Transformação do vetor de inteiros para a matriz de $N \times M$

A equação (4.8) não seria alterada uma vez que a maré de chegada é dada como parâmetro de entrada (veja ANEXO 1).

A equação (4.10) se tornaria uma variante da equação (4.4), uma vez que somente seria usado o x_{ij} na criação do vetor. Isso garante a integridade da matriz, sem a necessidade de verificar cada posição individualmente. Em outras palavras a equação (4.4) poderia ser usada nesse caso.

A equação (4.13) também teria uma pequena variante pelo mesmo motivo anterior. Nesse caso, apenas é preciso saber quanto de cada matéria prima tem cada navio em uma TTW j . Com isso poderiam ser utilizados parâmetros de entrada juntamente com o x_{ij} adquirido, como mostrado na Figura 5.2.

Para cada violação de restrição será acrescentado um valor, como penalidade, no resultado da função objetivo, como mostrado abaixo:

- Restrição de número máximo marés para a realização do carregamento/descarregamento: 100 para cada violação;
- Restrição de maré de chegada: 100 para cada violação;
- Restrição de violação de Estoque por maré: 50 para cada violação.

Esses valores foram obtidos com base nos valores obtidos pelo algoritmo e nos valores reais encontrados, a partir da contagem média do número de casas decimais desses valores.

Resumindo têm-se algumas características da adaptação da modelagem:

- Possui três restrições para avaliar uma solução candidata;
- Cada indivíduo da população será um vetor de números inteiros de tamanho N (número de navios);
- A criação de indivíduos a partir dessa nova modelagem faz com que algumas restrições não precisem ser verificadas.

5.2. Parâmetros de Entrada da Rede Bayesiana

Como mostrado no Capítulo 3, a rede bayesiana é criada a partir de uma população de indivíduos (soluções candidatas). Logo se deve determinar o tamanho da população inicial.

A rede terá como critério de parada um número fixo de iterações. Todavia também haverá como critério verificar se todos os indivíduos convergiram, ou seja, não é possível encontrar um indivíduo melhor com o passar das iterações.

Por último é determinado a quantidade de indivíduos novos gerados a cada iteração. Existe também um parâmetro opcional que indica a quantidade máxima de pais que cada nó da rede deve possuir, tendo como foco a diminuição da cardinalidade no espaço de busca.

5.3. Construção da Rede

Com a criação da população o próximo passo é ordenar a população de forma crescente – pois o problema é de minimização – através do valor de fitness (valor de avaliação) de cada indivíduo. Deve-se então selecionar os Q primeiros indivíduos para criar a rede.

Assumindo como exemplo que exista 4 berços no problema então cada vértice pode assumir quatro valores distintos. A partir disso deve-se criar um vetor contendo o número de ocorrências de seus valores em relação às suas adjacências como mostra a Figura 5.3.

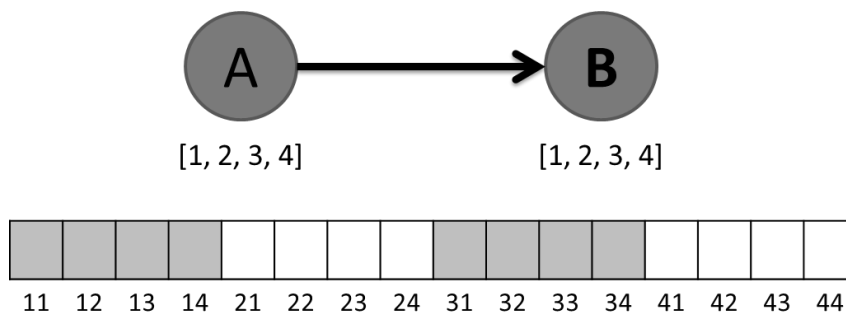


Figura 5.3 - Vetor de ocorrências

Observe que a rede inicialmente não possui dependências (arestas). As dependências serão adicionadas aplicando o vetor de ocorrências na equação (3.5). Com isso será gerado um número que será incluído numa matriz de ganhos como mostra a Tabela 5.1.

	A	B	C	D	E
A	-1,0	0.00476	0.00142	0.00476	0.00023
B	0.00132	-1,0	0.00066	0.03703	0.00066
C	0.00089	0.00148	-1,0	0.00148	0.00148
D	0.00132	0.03400	0.00066	-1,0	0.00066
E	0.00027	0.00066	0.00277	0.0027	-1,0

Tabela 5.1 - Matriz de ganhos para 5 navios

A conexão que tiver o maior ganho será adicionada na rede como aresta. No caso da Tabela 5.1 o maior ganho é de $B \rightarrow D$. Note que o ganho de $B \rightarrow D$ é diferente do ganho de $D \rightarrow B$.

Ao acrescentar uma nova dependência deve-se tomar o cuidado para que não forme um ciclo com as demais ou um laço com o próprio vértice.

O número máximo de arestas que a rede possui é definido pelo usuário.

5.4. Desenvolvimento da Nova Geração de Indivíduos

Com a estrutura da rede criada, falta agora atribuir a tabela de probabilidade de cada vértice.

Deve-se fazer uma ordenação topológica para verificar a ordem de dependências entre os nós.

Após a ordenação, a atribuição de valores às arestas ocorrer a partir de uma distribuição de probabilidade conjunta pra cada vértice. Assim cada vértice receberá um vetor contendo a probabilidade de assumir um determinado valor.

$p(1)$	$p(2)$	$p(3)$	$p(4)$
--------	--------	--------	--------

Tabela 5.2 - Vetor de probabilidade para um vértice sem pais, no caso de haver 4 berços no problema

Se houve quatro berços como mostra a Tabela 5.2, então o vértice poderia assumir quatro valores diferentes, sendo que cada valor tem sua respectiva probabilidade de ocorrer para aquele vértice.

Para vetores com um os mais pais deve-se utilizar a distribuição de probabilidade mostrada na equação (3.4). Assim seria criada uma tabela de probabilidades como mostra a Tabela 5.3.

v_2/v_1	1	2	3	4
1	$p(1 1)$	$p(1 2)$	$p(1 3)$	$p(1 4)$
2	$p(2 1)$	$p(2 2)$	$p(2 3)$	$p(2 4)$
3	$p(3 1)$	$p(3 2)$	$p(3 3)$	$p(3 4)$
4	$p(4 1)$	$p(4 2)$	$p(4 3)$	$p(4 4)$

Tabela 5.3 - Tabela de probabilidade para um vértice v_2 que possui o pai v_1 , no caso de haver 4 berços no problema

Na Tabela 5.3 o vértice v_2 é representado pela coluna e o vértice v_1 é representado pela linha da matriz.

Com a geração desses vetores é possível gerar novos indivíduos que serão acrescentados na população.

Com o passar das iterações as probabilidades vão se estreitando para certos valores, então os indivíduos da população irão convergir para um único de tipo de indivíduo.

6. RESULTADOS

Para validar Algoritmo de Otimização Bayesiano (BOA) foi necessário realizar vários teste com instâncias que correspondem a possíveis cenários reais.

O BOA encontrou o ótimo global em alguns casos, e em outros encontrou valores muito próximos, como mostrado na Tabela 6.1.

Os parâmetros de entrada da rede foram:

- Número de iterações: 800;
- Número de indivíduos da população inicial: 200;
- Número de indivíduos da amostra: 80;
- Número de indivíduos inferidos: 10;
- Número de máximo de pais para cada nó: 2;
- Número máximo de arestas na RB: 160.

Os resultados reais das instâncias foram encontrados a partir da utilização do software CPLEX (BARROS, 2010).

Nas instâncias onde não foi possível encontrar o ótimo real como, por exemplo, a 30-70-4, o BOA conseguiu achar um valor ótimo aceitável. Como não há o ótimo real para a comparação então não é possível avaliar de forma precisa o desempenho do algoritmo nesse caso, contudo é possível afirmar que a solução é válida, pois não infligiu restrições.

Instâncias	Navios	Berços	TTW	Ótimo	Obtido
10-15-4	10	4	15	29	29
10-20-3	10	3	20	44	44
10-25-2	10	2	25	43	44
15-30-4	15	4	30	77	79
15-35-3	15	3	35	86	86
15-40-2	15	2	40	121	125
20-40-3	20	3	40	107	114

20-40-4	20	4	40	108	113
20-55-2	20	2	55	297	317
30-70-4	30	4	70	-	210
30-75-3	30	3	75	208	219
30-80-2	30	2	80	-	949

Tabela 6.1 - Tabela com resultados comparativos entre o Ótimo Real e o Ótimo obtido pelo BOA

O número de indivíduos selecionados da população para a criação da rede influência diretamente na precisão das probabilidades na rede. Quando se computa os ganhos, a variação de valores de um único vértice, em relação à todas as soluções candidatas da população, aumenta a gama de possibilidades para encontrar novos indivíduos com bons *fitness*.

O número de filhos gerados ajuda a diversificar as probabilidades da rede. Quando esse número é pequeno, existe uma grande chance do indivíduo que contém o ótimo global seja parecido com indivíduos da população atual. Na área de Pesquisa Operacional isso pode indicar em alguns casos a convergência em um ótimo local.

7. CONCLUSÃO

Este trabalho apresentou o desenvolvimento de uma metodologia para a resolução de problemas de otimização linear, tendo como demonstração a resolução do Problema de Alocação de Berços em Portos Graneleiros utilizando os recursos de Redes Bayesianas.

Na etapa de modelagem do PAB, foi levado em consideração o fato de se utilizar vetores com números não binários, a fim de que não afetasse a metodologia dos Algoritmos Otimização Bayesianas e também não alterasse a forma como é analisado o modelo do Problema de Alocação de Berços em Portos Graneleiros.

O algoritmo em alguns poucos casos encontrava um ótimo um pouco longe do esperado, contudo a mudança nos parâmetros de entrada fazia com que ele encontrasse de forma mais precisa tais informações.

Esse trabalho ainda pode ser talvez melhorado, levando em consideração os seguintes aspectos:

- Escolha da métrica de pontuação;
- Alteração da modelagem de adaptação do problema do PAB;
- Usar generalizações do Algoritmo de Otimização Bayesiano, como por exemplo, o hBOA (*Hierarchal Bayesian Optimization Algorithm*) (PELIKAN, 2005);

Além desses fatores, pode-se estudar o caso de fazer hibridização com outras metodologias a fim de que melhore o desempenho em encontrar o ótimo global.

De maneira geral, a metodologia apresentada neste trabalho conseguiu mostrar a eficiência de se utilizar Redes Bayesianas, no sentido de conseguir achar uma solução relevante com pouco custo computacional, para resolver problemas de otimização linear.

REFERÊNCIAS

- BARROS, V. H. "Problema de alocação de berços em portos graneleiros com restrição de estoque e condições favoráveis de maré." Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal do Maranhão, São Luis, 2010, 85.
- BROWNLEE, J. *Clever Algorithms: Nature-Inspired Programming Recipes*. Editora LuLu, 2012.
- BUNTINE, W. "Theory Refinement on Bayesian Networks." *Proc. 7th Conf. Uncertainty in Artificial Intelligence*. 1991.
- CASTRO, P. A. D, e F.J. VON ZUBEN. "Algoritmos de Estimação de Distribuição", "Manual de Computação Evolutiva e Metaheurística." *Manual de Computação Evolutiva e Metaheurística*, 2012: pp. 22.
- COOPER, G. F., e E. HERSKOVITS. "A Bayesian Method for the Induction of Probabilistic." *Machine Learning*. Amsterdam, 1992.
- CORMEN, T. H.; LEISERSON, C. E.; L., RIVEST R.; STEIN, C.. *Algoritmos - Teoria e Prática*. Tradução: Vandenberg D. de Souza. Campus, 2002.
- CROCOMO, M. K. "Algoritmo de Otimização Bayesiano com Detecção de Comunidades." Tese (Doutorado em Ciências - Ciências da Computação e Matemática Computacional), USP, São Carlos, 2012.
- CROCOMO, M. K., e A. C. B. DELBEM. "Otimização Bayesiana com Detecção de Comunidades." Instituto de Ciências Matemáticas e de Computação, São Carlos, 2011.
- DARWIN. "On the origin of species or the preservation of favoured races in the struggle for life." Kent, UK: John Murray, 1859.
- DEMPSTER, A. P., N. M. LAIRD, e D. B. RUBIN. "Maximum Likelihood from Incomplete Data via the EM." *Journal of the Royal Statistical Society. Series B (Methodological)*. 1977.
- DEVORE, Jay L. *Probabilidade e estatística : Para engenharia e ciências*. Tradução: Joaquim Pinheiro Nunes da Silva. São Paulo, SP, 2006.
- GONÇALVES, A. R. "Otimização em ambientes dinâmicos com variáveis contínuas empregando algoritmos de estimação de distribuição." Dissertação (Mestrado em Engenharia Elétrica), Faculdade de Engenharia Elétrica e de Computação, Campinas, 2011.

- HOLLAND, J. H. "Adaptation in natural and artificial systems." Michigan: The University of Michigan Press, 1975.
- JUNIOR, V. V., R. G. SANTOS, T. S. COSTA, e A. C. M OLIVEIRA. "Greedy Heuristic for Berth Allocation in Tidal Bulk Ports." *Jornada de Informática do Maranhão (JIM 2012)/Escola Regional de Computação dos Estados do Ceará, Maranhão e Piauí (ERCEMAPI 2012)*, setembro 2012.
- JURKIEWICZ, Samuel. "Grafos - Uma Introdução." *Volume 5 do material utilizado no Programa de Iniciação Científica OBMEP*. 30 de junho de 2009.
- KRUSE, R., e C. BORGELT. "An Empirical Investigation of the K2 Metric." *6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*. Toulouse, 2001.
- LANGE, T. "Ambiente computacional para modelagem e simulação de redes bayesianas." Universidade do Vale do Itajaí, São José, 2008.
- LUNA, J. E. O. "Algoritmos EM para Aprendizagem de Redes Bayesianas a partir de Dados Incompletos." Dissertação (Mestrado em Ciência da Computação), Universidade Federal de Mato Grosso do Sul, 2004.
- MARIANA, A. C. *Disponível em:*
<<http://www.inf.ufsc.br/grafos/definicoes/definicao.html>>. s.d. (acesso em novembro de 2013).
- OCENASEK, J., e M. LAUMANN. "Bayesian Optimization Algorithms for Multi-Objective Optimization." *7th International Conference on Parallel Problem Solving from Nature*. Granada, 2002.
- PELIKAN, M. *Hierarchical Bayesian Optimization Algorithm*. Springer, 2005.
- PIFER, A. C. "Estudo Comparativo de Métricas de Pontuação para Aprendizagem de Redes Bayesianas." Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal do Rio Grande do Norte, Natal, 2006.
- REGOLIN, E. N. "Programação Genética e Algoritmos de Estimação de Distribuição." Dissertação (Mestrado em Informática), Universidade Federal do Paraná, Curitiba, 2004.
- RUSSEL, S. J., e P. NORVIG. *Inteligência Artificial*. 2ª ed. Rio de Janeiro: Elsevier, 2004.
- SANTOS, E. B. "A ordenação das variáveis no processo de otimização de classificadores." Dissertação (Mestrado em Ciência da Computação), São Carlos, 2007.

SPIRITES, P., C. GLYMOUR, e R. SCHEINES. "An algorithm for fast recovery of sparse causal graphs." *Social Science Computer Review*. 1991.

VEIGA, S. C. A., e W. T. SILVA. "Redes Bayesianas: Uma Visão Geral." Brasília, 2002.

ANEXO 1
EXEMPLO DE INSTÂNCIA
(10 NAVIOS, 15 JANELAS DE MARÉS, 4 BERÇOS)

```

data;
set N := 1 2 3 4 5 6 7 8 9 10;
set M := 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15;
set K := Alumina Bauxita Manganês Ferro;
set L := 1 2 3 4;
param Mares := 15;
param v :=
1 3
2 4
3 2
4 3;
param a :=
1 1
2 1
3 2
4 4
5 5
6 6
7 6
8 8
9 9
10 10;

param e :=
Bauxita 13
Alumina 14
Manganês 17
Ferro 15;
param ck :=
Bauxita 3
Alumina 2
Manganês 2
Ferro 2;
param q : Bauxita Alumina Manganês Ferro :=
1 0 3 0 0
2 0 10 0 0
3 8 0 0 0
4 0 4 0 0
5 0 0 3 0
6 0 0 0 10
7 0 0 10 0
8 13 0 0 0
9 14 0 0 0
10 0 0 0 5;
end;

```