INVESTIGAÇÃO DE ESQUEMAS ADAPTATIVOS PARA OTIMIZAÇÃO DE HEURÍSTICAS

Renato Antonio Rabuske

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS (D.Sc.)

Aprovada por:

João Lizardo R. H. de Araujo
(Presidente)

Paulo Oswaldo Boaventura Netto

Nelson Maculan Filho

Rajamani Doraiswami

Demétrio Alonso Ribeiro

Rio de Janeiro, RJ - Brasil

Dezembro de 1980

RABUSKE, RENATO ANTONIO

INVESTIGAÇÃO DE ESQUEMAS ADAPTATIVOS PARA OTIMIZAÇÃO DE HEURÍSTICAS (Rio de Janeiro, 1980)

, P. 29.7 cm (COPPE/UFRJ , D. Sc. , Engenharia de Sistemas, 1980)

Tese - Universidade Federal do Rio de Janeiro. Faculdade de Engenharia

ESQUEMAS ADAPTATIVOS - I. COPPE/UFRJ

II. Título (série)

Ã minha esposa Márcia e aos
meus filhos Allan e Patrícia

## AGRADECIMENTOS

- A DEUS, que com sua onisciência, assiste e lidera todo saber.

- Ao Professor João Lizardo Rodrigues Hermes de Araújo, pela eficiente orientação, pela solicitude no atendimento, pela disponibilidade e pasciência, pela ajuda que deu.

- Aos professores da banca examinadora, por terem aceito participar da mesma.

- Aos professores Jaime Luiz Szwarcfiter e Paulo Oswaldo Boaventura Netto, pela participação na banca das provas de qualificação.

- A todos os colegas com que convivi durante o curso de Doutorado, pelo estímulo, humor e companhia que proporcionaram.

- Aos colegas, professores do Departamento de Ciências Estatísticas e da computação da Universidade Federal de Santa Catarina, por terem aceito suportar o Ônus de meu afastamento.

- Ã CAPES, pelo auxilio financeiro que proporcionou.

- Ã Universidade Federal de Santa Catarina por permitir meu afastamento.

- Ã Professora Vânia conceição Tavares, por ter defendido meus interesses temporais na função de Procuradora.

- A todos os que, direta ou indiretamente, contribuiram para a realização deste trabalho.

# RESUMO

O presente trabalho tem como objetivo central montar um esquema adaptativo estocástico e testá-lo verificando suas potencialidades e limitações. O enfoque heurístico vale-se de resultados apresentados por ARAUJO[3].

A evolução do trabalho processa-se levando avante concomitantemente aspectos teóricos e experimentais.

O Desenvolvimento da pesquisa abrange os seguintes aspectos:

a) construção de um Esquema Adaptativo e otimização de seus parâmetros.

b) Eliminação da tendenciosidade do esquema, seja através de fatores de correção, seja pelo uso de fórmulas onde se evita ao máximo o recurso a valores estimados.

c) Otimização da Aplicação do Esquema e outros aspectos como escalarização de parâmetros, aceleração da busca, instabilidades ocasionais, sensibilidade do esquema e sua aplicação na determinação de caminhos mínimos ou quase-mínimos.

d) comparação dos resultados obtidos com os esperados, permitindo concluir que o esquema apresenta um bom índice de desempenho.

# A B S T R A C T

The present work has as its central objective to establish an adaptive stochastic scheme and test it checking its potentials and limitations. The heuristic approach uses results presented by ARAUJO[3].

The evolution of the work procedes taking into account simultaneously theoretical and experimental aspects.

The development of the research encompasses the following aspects:

a) Construction of an Adaptive Scheme and optimization of its parameters.

b) Elimination of the bias of the scheme, either by means of correction factors, or by use of formulas where the use of expected values is mostly avoided.

c) Optimization of the Scheme Application and other aspects as parameter scaling, search acceleration, occasional instabilities, sensitivity of the scheme and its application in the determination of minimal or quasi-minimal paths.

d) Comparison of the obtained results with the expected ones, allowing to conclude that the scheme presents a good performance index.

# I N D I C E

CAPÍTULO V

CAPÍTULO VII



APÊNDICES



ANEXOS

# C A P Í T U L O I

## 1.1 - INTRODUÇÃO

O presente trabalho propõe-se como objetivo, estudar um esquema adaptativo estocástico, observando suas deficiências, indicando alternativas, descobrindo suas limitações e estudando uma forma conveniente de aplicação.

Para conseguir o acima, foram seguidos os seguintes passos, traduzidos em diversos capítulos, cujos resumos seguem.

<u>Capítulo II</u>

É um apanhado das principais definições e conceitos necessários para um bom entendimento do trabalho. Os principais teoremas e corolários da programação heurística são ai enunciados, não havendo nenhuma preocupação em demonstrá-los, pois as provas se encontram nas referências bibliográficas. Acentuamos de modo especial o Algoritmo A*, base deste trabalho e acrescentamos o Algoritmo B*, que é uma das Últimas novidades que surgiram na bibliografia especializada.

<u>Capítulo III</u>

Pretende ser uma revisão bibliográfica sobre heurística e busca heuristica. Revela a escassez de autores que se aventuram em encarar aspectos probabilísticos dentro da heurís tica. Na parte final do capitulo foi dado um enfoque mais detalhado sobre os antecedentes, ou seja, a origem deste trabalho.

## Capítulo IV

Neste capitulo formulamos um problema e definimos seus parâmetros básicos. Em seguida sugerimos uma primeira proposição de solução e apresentamos as ferramentas de que o esquema se serve para atingir seu objetivo. A cada nova hipótese segue um experimento prático e uma análise dos resultados. Cada argumento, cada hipótese é questionada e tentativas de otimização são analisadas.

## Capítulo V

Em vista de uma tendenciosidade latente no esquema, mostrada no capítulo IV, a análise foi dirigida para possíveis formas de eliminá-las. São efetuadas quatro tentativas, três delas apresentando visível sucesso.

## Capítulo VI

Neste capítulo fazemos testes com hipóteses que no decorrer do trabalho foram ventiladas. Encaramos aspectos como $\underline{\underline{S}}_t$ matriz escalar, $\underline{\underline{\Sigma}}_t$ conhecido ou estimado a priori, aspectos da instabilidade de $\underline{\underline{S}}_t^{-1}$, influência de $\theta$ na busca, variação na dimensão de $\underline{\underline{S}}$ e $\underline{\underline{A}}$, aspectos sobre o caminho mínimo e sua obtenção através da aplicação do esquema proposto, otimização da aplicação do esquema e sua sensibilidade, bem como uma nova forma de expansão da árvore de busca.

Capítulo VII

Relata as conclusões e sugestões encarando em separado três aspectos do trabalho, quais sejam: Estimativa dos parâmetros (referente ao capítulo IV), Correção da tendenciosidade (cap. V) e Sobre a Aplicação do Esquema (cap. VI).

# C A P Í T U L O I I

## CONCEITOS BÁSICOS DA PROGRAMAÇÃO HEURÍSTICA

Os conceitos que seguem foram coletados em diversos livros e artigos especializados, destacando-se especialmente HART ET AL[15], NILSSON[23], ARAUJO[4] e GOMEZ[12], onde o leitor poderá aprofundar os tópicos enunciados neste capítulo.

O objetivo deste capitulo é reunir conceitos que posteriormente serão de alguma forma úteis, evitando-se definições extra-contexto.

### 2.1 - Grafo de um Problema

Suponhamos que um problema admita a seguinte formulação:

Sejam S o conjunto de estados iniciais (conhecidos), T o conjunto de estados desejados (terminais) e $\Gamma$ um conjunto de operações sobre os estados, isto é, se $e_i$ é um estado do problema e $\gamma \in \Gamma$ é uma operação aplicável a $e_i$, então $\gamma(e_i) = (e_i, C_i)$, onde $e_i$ é um estado do problema e $C_i$ é o custo associado a aplicação de $\gamma$ a $e_i$.

Assim posto, o problema é isomorfo a um grafo ponderado e orientado a que chamaremos de grafo do problema e terá como notação $G_s(S, \Gamma)$, ou simplesmente $G_s$:

Definição 2.1

$G_s(S, \Gamma)$ é chamado Grafo de um problema, se e somente se:

(i) $S \subset V(G_s)$, o conjunto de vértices (ou nós) de $G_s$.

(ii) $n_i \varepsilon V(G_s)$, $\gamma \in \Gamma$ onde $\gamma(n_i) = (n_j, C_{ij})$

$n_j \varepsilon V(G_s)$ e $C_{ij}$ = custo do arco $(n_i, n_j)$ e $n_j$ é determinado sucessor de $n_i$.

## 2.2 - Conceituação

### 6 - Grafos

É a classe de grafos com custos de transição $C_{ij}$ superiores a algum $\delta > 0$.

### Custos Aditivos

Seja $L(n_o, n_r)$ um caminho de $n_o$ a $n_r$. Os custos são aditivos se pudermos calcular o custo de L como

$$K_L(n_o, n_r) = \sum_{j=0}^{r-1} C_{j,j+1} \tag{II-1}$$

### Definição 2.2

seja $n \varepsilon V(G_s)$, então

$$g(n) = \begin{cases} \min\limits_{s \varepsilon S} K_L(s,n) & \text{se } L(s,n) \text{ existe} \\ +\infty & \text{caso contrário} \end{cases} \tag{II-2}$$

$$h(n) = \min_{t \varepsilon T} K_L(n,t) \tag{II-3}$$

### Definição 2.3

O custo mínimo de um caminho de S a T, devendo passar por um nó $n \in V(G_S)$, é definido

$$f(n) = g(n) + h(n) \quad \text{(II-4)}$$

Definição 2.4

$L(n_o, n_r)$ é dito solução de um problema se e somente se, $n_o \in S$ e $n_r \in T$.

Definição 2.5

$L(n_o, n_r)$ é uma solução Ótima se e somente se

(i) $n_o \in S$

(ii) $n_r \in T$

(iii) $K_L(n_o, n_r) = f(S)$.

Definição 2.6

Um algoritmo de busca é admissivel, se para todo $\delta$ - grafo tendo uma solução finita, o algoritmo garante encontrar uma solução Ótima em um número finito de passos. Note-se que resolver o problema equivale a explorar parcialmente $G_s$ até achar uma solução.

Definição 2.7

Se a busca da solução é feita seletivamente, utilizando informações específicas ao domínio do problema, dizemosque a busca é heuristica e o algoritmo usado de Algoritmo de busca heurística.

## 2.3 - Busca Heurística

Suponhamos a possibilidade de calcular uma função de mérito $\hat{f}(n)$ para cada $n \varepsilon V(G_s)$. Podemos usar o algoritmo de busca ordenada de HART ET AL[15], sendo necessário armazenarmos:

(i) Descrição do estado que o nó n representa.

(ii) Distinção entre nós abertos e fechados.

Seja $F(n) = 0 \rightarrow n$ aberto

$F(n) = 1 \rightarrow n$ fechado.

(iii) Valor de $\hat{f}$ dos nós abertos.

(iv) Apontador P para seu melhor antecessor.

Sejam S: conjunto de vértices iniciais

T: conjunto de vértices terminais

Algoritmo de Busca Ordenada

P1 - Para todo $s \varepsilon S$, calcule $f(s)$, faça $F(s) = 0$, $P(s) = 1$ e guarde.

P2 - Escolha entre os nós abertos o nó n com melhor $\hat{f}$; Se houver mais de um, desempate arbitráriamente, mas sempre a favor de nós terminais. Se não há nó aberto, para com fracasso.

P3 - Se $n \varepsilon T$, faça $F(n) = 1$ e pare. Seguindo-se os apontadores P, obtém-se a solução.

P4 - Se não, faça $F(n) = 1$ e gere todos os sucessores de n; Para cada sucessor m de n, calcule $\hat{f}(m)$. Se m é um novo nó, ou se o novo valor de

$\hat{f}(m)$ é melhor do que o anterior, faça $P(m)=n$, $F(m) = 0$ e guarde-o com o novo $\hat{f}(m)$.

P5 - Volte a P2.

Observações

Este algoritmo só para com fracasso se o problema não tem solução. O caso mais importante quanto à forma de $\hat{f}$ é: $\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$ (Algoritmo A*), onde $\hat{h}(n)$ (função heurística), é uma estimativa de $h(n)$ e $\hat{g}(n) = k(s,n)$. Se $\hat{h}(n)$ depende do estado da busca, seu valor deve ser atualizado para todos os nós abertos ao fim do passo P4.

Teorema 2.1 (HART ET AL[15])

Se $\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$ e $\hat{h}(n) \leq h(n)$ para todos os nós de $G_s$, então A* é admissivel e além disso, somente nós tendo $\hat{f}(n) \leq f(s)$ são fechados. Este resultado é extensível a grafos finitos com custos de transformação não negativos.

Definição 2.8

Uma função heurística é dita consistente se e somente se, para todo m e $n \in V(G_s)$

(i) $\hat{h}(n) \leq h(n)$

(ii) $\hat{h}(m) - \hat{h}(n) \leq K_L(m,n) \longrightarrow m < n$

Teorema 2.2 (HART ET AL[15])

Se $\hat{h}$ é consistente, quando A* fecha um nó, já achou o caminho Ótimo até ele.

Proposição 2.1 (NILSSON[23])

Se $\hat{h}$ é consistente, e a sequência de nós fechados por A* é $(n_1^*, n_2^*, \ldots, n_r^*)$, então, para todo par i,j (i < j) temos $\hat{f}(n_i) \leq \hat{f}(n_j)$.

Definição 2.9

Seja $\theta_n^A$ o conjunto de índices usado pelo algoritmo A no nó n. Então, se $\theta_n^{A^*} \subset \theta_n^A$ para todos os nós n em $G_s$, podemos dizer que o algoritmo A é não mais informado que o algoritmo $A^*$.

Seja $\mathcal{A}^*$ o conjunto de todos os algoritmos que agem como $A^*$ quando não há empates, mas que resolvem os empatesdeforma difereiite.

Teorema 2.4 (HART ET AL[15])

Seja A qualquer algoritmo admissível, não mais informado que os algoritmos em $\mathcal{A}^*$, e suponha que a condição de consistência seja satisfeita por h(.) usado nos algoritmos em $\mathcal{A}^*$, então, para todo $\delta$-grafo $G_s$, existe um $A^* \in \mathcal{A}^*$ tal que todo nó não terminal expandido por $A^*$, também o será por A.

Corolário 2.1

Seja $N(A^*)$ e N(A) o número de nós visitados pelos Algoritmos $A^*$ e A respectivamente e as premissas do teorema 2.4 estejam satisfeitas, então, para qualquer $\delta$-grafo $G_s$, $N(A^*) \leq N(A)$, ocorrendo a igualdade se e somente se A expande exatamente os mesmos nós não-terminais que $A^*$.

Corolário 2.2

Se $R(A^*)$ é o número de empates que ocorreu na aplicação de $A^*$ e se as premissas do teorema 2.4 estão satisfeitas, então, para qualquer 6-grafo $G_s$, $N(A^*) \leqslant N(A) + R(A^*)$.

Definição 2.11 (VANDERBRUG[30])

Um algoritmo A é dito Completo se, para todo 6-grafo $G_s$ possuindo solução finita, ele encontra uma num número finito de passos.

Teorema 2.5 (ARAUJO[4])

Se existe $a > 0$ e $z \in \mathbb{R}$ tais que $z \leqslant \hat{h}(n)$ 6 $\alpha\hat{h}(n)$ para todo $n \in V(G_s)$, então $A^*$ é completo.

Definição 2.12 (ARAUJO[4])

Se $\hat{h}$ é uma variável aleatória, então T, o número de nós fechados por $A^*$, também o é, e se o algoritmo satisfaz $E(T)$, $E(T^2) < \infty$, então é chamado Completo em Média Quadrática.

## 2.4 - Medidas de Desempenho

Um dos primeiros métodos, proposto por DORAN[8], foi a penetrância $\rho = k/T$, onde k é o custo da solução achada e T o custo de fechamento dos nós.

SLAGLE[28], introduziu a razão de profundidade $DR = \log T / \log N_o$, sendo $N_o$ o custo total da busca (medido nas mesmas unidades que T), empregando $\hat{h} = 0$, ou seja, o algoritmo de Dijkstra.

NILSSON[23] propôs o fator de ramificação efetiva R de uma árvore de profundidade k, possuindo T nós, e R satisfaz

$$R \frac{R^k - 1}{R - 1} = T$$

Nilsson sugere que T seja o número total de nós gerados, o que é um lapso, levando-se em conta a estrutura de $A^*$, pois, mesmo numa busca perfeita, o número de nós gerados é maior que k, pois todos os sucessores do nó fechado são gerados.

P e R dependem das dimensões dos problemas. Isto dificulta a avaliação de uma mesma função heuristica empregadaem problemas diferentes.

Embora DR não tenha este inconveniente, requer o conhecimento de $N_o$.

ARAUJO[4], contorna estes problemas propondo medir o conteúdo de informação de uma função heuristica $\hat{h}(n)$, por um número, tal que a busca usando $\alpha h(n)$ fecha o mesmo número de nós (ou dá o mesmo custo de fechamento) que empregando $\hat{h}(n)$. $\alpha$ é aproximadamente independente das dimensões, sendo essencialmente determinado por $\hat{h}(n)$.

Assim, pressupondo que $G_s$ seja uma árvore com fator de ramificação constante B (número de sucessores por nó), com arcos sendo percorridos em ambos os sentidos, tendo custo unitário e que exista uma solução Ótima de comprimento k, fornece como medida de desempenho a "informação heuristica equivalente":

$$a = \frac{\log B - \log R}{\log B + \log R} \quad ; \quad R = B^{(1-\alpha)/(1+\alpha)}$$

onde R satisfaz a equação

$$\frac{R^k - 1}{R - 1} = T$$

$$\text{e } B = \frac{\text{N? de nós gerados}}{\text{N? de nós fechados}} = \frac{G}{T}$$

isto é

$$G = B.T = B\,\frac{R^k - 1}{R - 1}$$

## 2.5 - O Algoritmo $B^*$

BERLINER[7], apresenta um algoritmo, por ele denominado $B^*$, que prova que um arco na raiz de uma árvore de busca é melhor que qualquer outro dos concorrentes. Para cada nó provê duas avaliações: Uma otimista, outra pessimista. Estabelece com isto limites para avaliações na subárvore deste ramos. Estes limites são gradativamente reajustados (reduzindo o intervalo) durante a busca, até coincidirem.

O domínio de $B^*$ inclui tanto problemas de busca 1-pessoa quanto de 2-pessoas (adversários). Nestes Últimos, uma pessoa procura maximizar uma função enquanto outra procura minimizá-la.

O que distingue $B^*$ da busca "O melhor-primeiro":

(1) Uma busca o melhor-primeiro é entendida como "achar um objetivo a partir do nó que mais promete", sendo este fechado. $B^*$ está interessado somente em achar "o melhor primeiro passo" em direção ao objetivo, enquanto a prova (*) não estiver completa. A sutileza está no fato de que é sem sentido estender uma ramificação, melhorando o seu valor, se isto não muda o "status" da prova.

(2) $B^*$ pode escolher uma estratégia, mesmo que esteja ainda na raiz da árvore.

O algoritmo usa as seguintes variáveis:

CURNODE : guarda a pista do nó corrente

DEPTH : recorda a distância do CURNODE à raiz

OPTIM : limite a cada passo

PESSIM : limite a cada passo

PARENT : apontador para o pai do nó

MAXOPTIM: guarda a pista do valor mais otimista de todos os sucessores de CURNODE;

MAXPESS : guarda a pista do valor "pessimista" de todos os sucessores de CURNODE.

observação:

No passo 4 há dois testes que dependem do ponto de vista como é encarado o problema. O operador "'" indica que dependendo deste ponto de vista, > torna-se < e < torna-se >.

P1 - DEPTH ← 0; CURNODE ← 0; OPTIM (0) ← -∞; PESSIM (0) ← ∞.

P2 - Se CURNODE ainda não foi expandido, então gerar e avaliar os sucessores, dando a todos um apontador para CURNODE;

P3 - BESTNODE ← nome do sucessor de CURNODE com melhor valor OPTIM;

ALTERN ← Nome do Sucessor com segundo melhor valor OPTIM;

MAXOPTIM ← OPTIM (BESTNODE);

MAXPESS ← Valor do melhor PESSIM de todos os sucessores;

P4 - Se [MAXOPTIM >' PESSIM (CURNODE)] E [MAXPESS <' OPTIM (CURNODE)], vai a P5

**(*) Prova: refere-se à forma como trabalha B*, provando otimalidade para determinado sucessor em cada raiz da árvore de busca.**

PESSIM (CURNODE) ← MAXOPTIM;

OPTIM (CURNODE) ← MAXPESS;

Se DEPTH > 0

então CURNODE ← PARENT (CURNODE);

DEPTH ← DEPTH - 1;

Vai a P3;

Se DEPTH = 0

então se PESSIM (BESTNODE) $\geqslant$ OPTIM (ALTERN)

então sai com ANSWER = BESTNODE;

P5 - Se DEPTH = 0, então decide estratégia;

Se estratégia = DISPROVEREST, então CURNODE ← ALTERN;

Se estratégia = PROVEREST, então CURNODE ← BESTNODE;

P6 - Se (DEPTH $\neq$ 0), então CURNODE ← BESTNODE;

P7 - DEPTH ← DEPTH + 1; Vai a P2.

O leitor poderá encontrar mais detalhes em [7], onde aparece também uma comparação com outros algoritmos de busca.

# C A P Í T U L O  III

## HEURÍSTICA E BUSCA HEURÍSTICA

### 3.1 - Heurística em Geral

A busca heuristica foi uma das idéias importantes no desenvolvimento da pesquisa em Inteligência artificial. É um conceito não totalmente definido e foi usado como um guarda-chuva para muitas técnicas computacionais que são dificieis de classificar ou analisar. Isto é bom porque leva a imaginação por caminhos isentos, a tentar técnicas que funcionam em problemas complexos.

Contudo, a não conceituação leva à repetição. Seguidamente análises analíticas levam a procedimentos mais eficientes.

É difícil atingir as origens do uso da heuristica como processo facilitador de tarefas. Aparentemente sempre andou junto com a ciência como fonte inspiradora, mantendo-se transparente.

Contudo, em fins da década 50 e início da 60, apareceramdiversostrabalhos que procuravam tornar concreta a intervenção da heuristica na solução de diversos problemas.

Para rever o que ocorreu, é preciso ir a bibliografia da época, não tão especifica quanto a atual, e proceder cuidadosaç pesquisas.

Os programas de busca heuristica de NEWELL, SIMON e SHAW[21,22], geraram grande entusiasmo. Aplicando as mesmas téc nicas a diversos problemas de domínios diferentes, demonstraram empíricamente o caráter geral da busca heuristica.

Já um pouco mais recente é a tentativa de DORAN e MICHIE[8,18], com seus Graph Traverser Programs (GTP), em que os mecanismos de busca são programas de busca em grafos, dirigidos por funções heuristicas que são estimadores de distâncias em grafos.

Os GTPs foram usados para comparar a eficiência de diferentes funções heuristicas na solução de quebra-cabeças.

HART, NILSSON e RAPHAEL[15], baseados em experiências anteriores de Nilsson sobre determinação do menor caminho em grafos, descobriram como usar melhor a informação para melhorar a eficiência do cálculo do menor caminho. Eles generalizaram isto para o conceito de estimação heurística e fizeram um algoritmo que usa esta informação no cálculo do menor caminho. Também produziram os primeiros teoremas sobre a eficiência como função de exatidão da função heurística.

MARTELLI[16], analisa a complexidade de algoritmos, em particular o A*, para o qual atribui $O(2^N)$ passos no <u>pior caso</u>.

Apresenta, além disso, um algoritmo B, que, conforme ARAUJO[6], é dominado por uma subclasse de A*, com dadas regras de desempate.

GELPERIN[11], aponta algumas falhas em demonstrações sobre otimalidade de A*, feitas anteriormente por Nilsson, e apresenta provas novas. Além disso, mostra através de exemplos , que:

1) O valor de $\hat{h}(n)$ pode ser uma função do estado da busca assim como da informação heurística disponível. Esta é uma

idéia já antiga, ventilada por Nilsson e repetida por Blaydon em [8].

2) Que existem algoritmos de busca admissiveis que não podem ser simulados por qualquer algoritmo A*.

POHL[26], apresenta um modelo particular de busca heuristica para achar um caminho num grafo dirigido, o "Heuristic Path Algorithm" (HPA). O HPA tem seu protótipo em MOORE[20]. O que o diferencia de A* é $f(n) = (1-w)\, g(n) + w h(n)$, sendo $0 \leqslant w \leqslant 1$. Se W = 0.5, HPA e A* serão idênticos.

A função avaliação pode ser extendida a uma forma linear geral,

$$f(n) = \left(1 - \sum_{i=1}^{K} w_i\right) g(n) + \sum_{i=1}^{K} w_i h_i(n)$$

$$\text{com} \quad \sum_{i=1}^{K} w_i \geqslant 0 \quad ; \quad \sum_{i=1}^{K} w_i \leqslant 1$$

podendo os $w_i$ serem funções dos nós. Isto permitiria um sistema com aprendizado, medir o desempenho de cada termo como contribuição para a estimação da distância.

HARRIS[14], introduz mais uma restrição sobre a heuristica, a "Bandwidth Condition" (BC), que permite à busca heuristica lidar melhor com problemas práticos de tempo e espaço. O tempo para a busca pode ser reduzido se aproximarmos melhor os valores de h(.). Pode ser impraticável determinar uma heurística que nunca viole a condição de admissibilidade. Em muitos casos é fácil achar um novo valor para h(.), embora sendo este superestimado. BC requer que

$$h(n) - d \leqslant \hat{h}(n) \leqslant h(n) + e$$

o que é um afrouxamento da admissibilidade. Claro é que, se $e = 0$ e $d > h(n)$ para todos os nós, BC se reduz a admissibilidade.

A busca BC minimiza a dependência crítica em relação à heurística e elimina a necessidade da fixação a priori de restrições ao processo de busca.

## 3.2 - Heurística, Probabilidade e Processos Estocásticos

No início da década 70 apareceram os primeiros trabalhos relacionando heurística à probabilidade e processos estocásticos. Está atrás desta tentativa de relacionamento, a intenção de mostrar que grande parte dos sucessos da primeira podem ter uma explicação segura nesta última, embora, até o momento, não se tenha a palavra final sobre o assunto. Creio mesmo que tão cedo não se terá, pois, a heurística é uma tentativa de usar a forma como o intelecto humano encara os problemas.

A maioria dos problemas admitem soluções "ad hoc", e então para haver maior síntese nos tipos de soluções, deve-se encarar a possibilidade de reuni-los em famílias de problemas, o que deve ser tentado com mais intensidade, considerando a importância do assunto.

Entre os autores que mais se destacaram no enfoque probabilístico em problemas heurísticos estão ARAUJO[1,2,3,4 e 5] e VANDERBRUG[31].

Abordaremos, dentro da ordem cronológica de publi cação, os principais aspectos enfocados, bem como, os resultados obtidos.

VANDERBRUG[31], determina o número total esperado de nós expandidos para diversas suposições probabilísticas em relação à profundidade do objetivo, do número de sucessores do nó, e do número de tais sucessores que são expandidos.

Apresenta um modelo probabilístico para estratégias de busca heurística. O modelo considera como medida de desempenho o valor esperado do número de nós a serem fechados, até que se ache a solução. O modelo pode ser aplicado a uma classe de problemas caracterizada por restrições em suas estruturas e na localização dos nós objetivos.

A fim de exemplificar, Vanderbrug desenvolve um modelo nas seguintes condições: O espaço de busca dever ser uma árvore sendo Poisson tanto o número de sucessores por nó como a profundidade da melhor solução.

As principais limitações do modelo são:

i) O espaço de busca é árvore.

ii) O pior caso é considerado como se produzisse resultados semelhantes a busca breadth-first.

iii) Considera como medida de eficiência o número de nós expandidos (não considera custos computacionais, etc ...).

ARAUJO[4], aborda em primeiro lugar a eficiência da busca heuristica, já anteriormente enfocada. Em seguida aborda a busca heuristica sob o ponto de vista randômico. Como resultado preliminar estima o número de nós fechados E(T) e o número de nós gerados E(G), respeitadas as condições do problema.

Considera dois casos:

i) A distribuição de $\hat{\alpha}$(vide 2.4), tem átomos.

ii) A distribuição de $\hat{\alpha}$ é absolutamente contínua.

Na parte final é apresentado um algorítmo para avaliar E(T) e E(G). Embora um pouco trabalhosa, a implementação deste algorítmo é sem dúvida viável em relação ao tempo e espaço. O importante é que ele proporciona condições para investigar que valor ou valores dos coeficientes de $\hat{h}(.)$ minimizarão os custos da busca.

Em [1], Araujo aprofunda alguns dos resultados obtidos em [4] e discute rápidamente os resultados de VANDERBRUG[31], fazendo as seguintes considerações:

a) O modelo não tem relação com a estrutura do algorítmo de busca.

b) Para que os resultados seja corretos, requer-se:

b1) Todos os nós à profundidade $K+1$ sejam terminais, (K é o comprimento de caminho ótimo).

b2) A expressão que calcula o número esperado de nós fechados, não inclua o Último.

b3) Ao iniciar-se a busca, o algorítmo já saiba que a melhor solução tem comprimento K.

ARAUJO[2], analisa o problema probabilístico do caminho mínimo, visto através da programação heurística.

Sendo o problema do caminho mínimo definido como: Achar uma tripla $(s^*, t^*, \gamma^*)$, com $s^* \in S$, $t^* \in T$, $\gamma^* \in \Gamma(s^*, t^*)$ tal que $K_{\gamma^*}(s^*, t^*) = \min\{K_\gamma(s,t)/s \in S,\ t \in T,\ \gamma \in \Gamma(s,t)\}$, e sendo as letras com chapéu consideradas variáveis aleatórias, en foca dois casos:

i) Achar (s*, t*, y*) tal que

$$E\left[\hat{K}_{\gamma^*}(s^*, t^*)\right] = \min_{s,t,\gamma} \{E\left[\hat{K}_{\gamma}(s,t)\right]\}$$

(Problema do caminho mínimo esperado).

ii) Dado a $\epsilon (0,1)$, achar (s*, t*, y*, d*), tal que

$$Pr\left[\hat{K}_{\gamma^*}(s^*, t^*) \leqslant d^*\right] \geqslant a \quad e \quad d^* = \min\{d / Pr\left[\hat{K}_{\gamma}(s,t) \leqslant d\right] \geqslant \alpha\}$$

A esta formulação denomina "Problema Probabilístico do Caminho Mínimo". Os custos dos arcos são aproximados por normais independentes (custos não aditivos).

Para resolver (ii), apresenta um algorítmo, uma extensão de A* que denomina Ã.

ARAUJO[3], considera buscas heurísticas como processos estocásticos, e procura formular o problema da otimização das heurísticas empregadas.

Como este trabalho inspirou em parte a presente pesquisa e dele adotamos parte da terminologia, vamos estender-nos um pouco mais sobre o mesmo.

### 3.2.1 - Otimização de $\hat{h}(n)$ - Formulação do Problema

Suponhamos que $\hat{h}(n)$ é da forma $\phi(\hat{H}(n), w)$, onde $\hat{H}(n)$ é um vetor de medições sobre n e $w \in \Omega$ é um vetor de parâmetros a determinar de modo a otimizar o desempenho de A*.

Seja: $\hat{Z}(W)$ , o custo total da geração de nós durante a busca;

$G_s$ , um grafo probabilístico, portanto $\hat{Z}(W)$ será uma variável aleatória;

$P(W,e)$, a probabilidade de pararmos numa solução de custo igual ou superior a $f(S) + E$; ($f(S)$ também é variável aleatória).

Para chegar à formulação final do Problema, Araujo descarta diversas alternativas, que, sob aspectos ponderáveis , tornam-se inviáveis e chega a seguinte:

Minimizar em $\Omega$

$$E\left[(\Phi(\hat{H}(h),\ W) - \theta\, h)^2\right] \qquad \text{(III-1)}$$

ou

$$E\left[\left(\frac{\Phi(\hat{H}(h),W)}{h} - \theta\right)^2\right] \qquad \text{(III-1')}$$

onde 8 é um escalar determinado em separado.

Seja $W(\theta)$ a solução de (III-1) ou (III-1') que dá a menor dispersão das estimativas em torno de $\theta h(n)$. Se a parametrização de W por 8 for tratável, então pode-se escolher 8 de modo a minimizar.

$$\{E\left[\hat{Z}(W(\theta))\right] \ /\ \theta \in R^+,\ P(W(\theta)),\ \varepsilon) \leqslant \alpha\} \qquad \text{(III-1")}$$

reduzindo a otimização a um problema unidimensional. Para obter $E\left[\hat{Z}(W(8))\right]$, recorremos aos modelos de ARAUJO[1,4]. Como, devido a dificuldades no cálculo e desconhecimento das distribuições en volvidas, a formulação (III-1") pode tornar-se inconveniente, e por outro lado, se temos uma idéia da ordem de grandeza de $f(S)$ e não conhecemos a distribuição de $h(n)$ (o que é frequente), a seguinte formulação é viável:

i) $\forall \theta \in R^+$, $W(\theta)$ minimiza

$$E\left[(\Phi(A(h), W) - \theta h)^2 / h=h_0\right]$$

onde $h_0$ é especificado.

ii) $\theta = \theta_0$, onde $\theta_0$ satisfaz $P(\theta_0, \varepsilon) \leq a$ .

Para valores pequenos de $\varepsilon$ e a, Araujo sugere para $\theta_0$, valor próximo a

$$\theta(\alpha, \varepsilon) = \sup\{\theta \in R^+ / P(\theta, \varepsilon) \leq \alpha\}$$

e apresenta para $P(\theta, \varepsilon)$ longa e complexa dedução.

3.2.2 - O caso Linear - Forma Geral de solução

$$\text{Se} \quad \mu(h,W) = E\left[\frac{\Phi(\hat{H}(h),W)}{h} / h\right] \quad \text{e}$$

$$\sigma^2(h,W) = \text{Var}\left[\frac{\Phi(\hat{H}(h),W)}{h} / h\right]$$

então (III-1) e (III-1') podem ser assim reescritos:

```
C-----------------------------------------------------------------------
C**** PROGRAMACAO E TESTE DO ESQUEMA ADAPTATIVO
C-----------------------------------------------------------------------
      DIMENSION B(15,15),SIGMA(15,15),R(16),ABARRA(15)
      DIMENSION SA(10,15,15),IFILA(2000),SEG(50),SEH(50)
      DIMENSION W0(15),G(2000),H(2000),IPAI(2000),A(15)
      DIMENSION HC(2000),S(15,15),W(15),STIL(15,15),EHCP(15)
      DIMENSION VML(15),AI(15,30),SO(15,15),EHCF(15),ISUC(2000)
      DIMENSION AX(15),M1(50,120),VETARG(9),ISIN1(9),ISIN2(9)
      DIMENSION  FDIS(130),DENS(130),NAVE(5),ESCALA(9)
      DIMENSION NOSP(80),NACE(80)
      REAL L(2000),Y(16),D(2000)
      COMMON N,NP,NSUC,IP,IU,NV,P(2000),C(2000),WTIL(15)
      COMMON HCM(4,2000)
      DATA ITRACO,IBRANC,ISIN1,ISIN2/'-',' ','.','*','+','=',
     1'0','A','X','Y','L','1','2','3','4','5','6','7','8','9'/
C
C**** NAVE(I)=POSITIVO, SERA RODADA TENTATIVA I
C     NAVE(I)=NEGATIVO, TENTATIVA I NAO SERA RODADA.
C
      HCPQ=1.
      NAVE(1)=1
      NAVE(1)=-1
      NAVE(2)=1
      NAVE(2)=-1
      NAVE(3)=-1
      NAVE(3)=1
      NAVE(4)=1
      NAVE(4)=-1
      NAVE(5)=1
      NAVE(5)=-1
      NALTER=5
C
C**** FCORRE=POSITIVO, ENTAO RODA  A-TIL COM CORRECAO
C
      FCORRE=-1.
      FCORRE=1.
C
C**** FAZ IESTAB=1 SE RODAR ESTABILIDADE
C     IESTAB=0 SE SE RODAR CONVERGENCIA
C
      IESTAB=1
      IESTAB=0
C
C**** IMPRA=1-IMPRIME GRAFICO  IMPRA=0-NAO IMPRIME
C
      IMPRA=1
      IMPRA=0
C
C**** NNOS=NUMERO DE NOS A SEREM EXPANDIDOS
C
      NNOS=660
      NNOS=120
      NNOS=200
      NNOS=300
      FTG=1.
      FTH=1.
```