

RACIONALIDADE LIMITADA

Luís Morgado

ISEL-ADEETC

RACIONALIDADE

Um sistema é racional se faz a “*acção certa*”
dado o **conhecimento** que possui [Russell & Norvig, 2003]

- **Capacidade de agir** no sentido de conseguir o **melhor resultado possível** perante os **objectivos** que se pretende atingir

Problema:
Complexidade computacional

RACIONALIDADE

- **RACIONALIDADE ILIMITADA**

- Informação completa
- Recursos ilimitados
- **Optimização**

- **RACIONALIDADE LIMITADA**

- Informação incompleta
- Recursos limitados
- **Satisfação**

- Processo de decisão orientado para a satisfação de níveis de referência sub-ótimos
- O processo de decisão pára quando o nível de referência é satisfeito
- Qualidade da solução vs. recursos utilizados

RACIONALIDADE LIMITADA

- **RESTRIÇÕES**

- Complexidade computacional
- Recursos limitados

- **MÉTODOS PARA LIDAR COM A COMPLEXIDADE**

- Aproximação
- Abstracção
- Modularização
- Focagem



APROXIMAÇÃO

- **MÉTODOS HEURÍSTICOS**

- Respostas **imprecisas** mas **expeditas**

- Por **eliminação de restrições** do domínio do problema
 - Por exploração de **regularidades** no ambiente
 - Por **memória** de situações relacionadas
 - Por **frequência de ocorrência**
 - Por **categorização**
 - Por **ancoragem** a um referencial prévio

- Plausibilidade biológica

- Heurísticas simples e rápidas
 - *Fast and frugal heuristics* [Gigerenzer, 1999]
 - Respostas de natureza afectiva

HEURÍSTICA

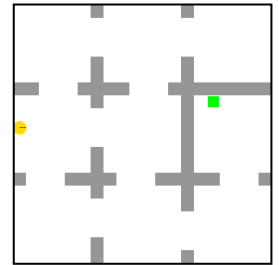
- **Aproximação** para resolução de um problema de forma expedita
- Tem por objectivo **tornar viável a resolução de um problema com restrições de recursos** (tempo, memória), quando isso não é possível com um método óptimo
- Tem inerente um **compromisso entre resposta expedita e:**
 - **Soluções sub-óptimas**
 - **Não completude**
 - **Exactidão**
 - **Precisão**

FUNÇÃO HEURÍSTICA

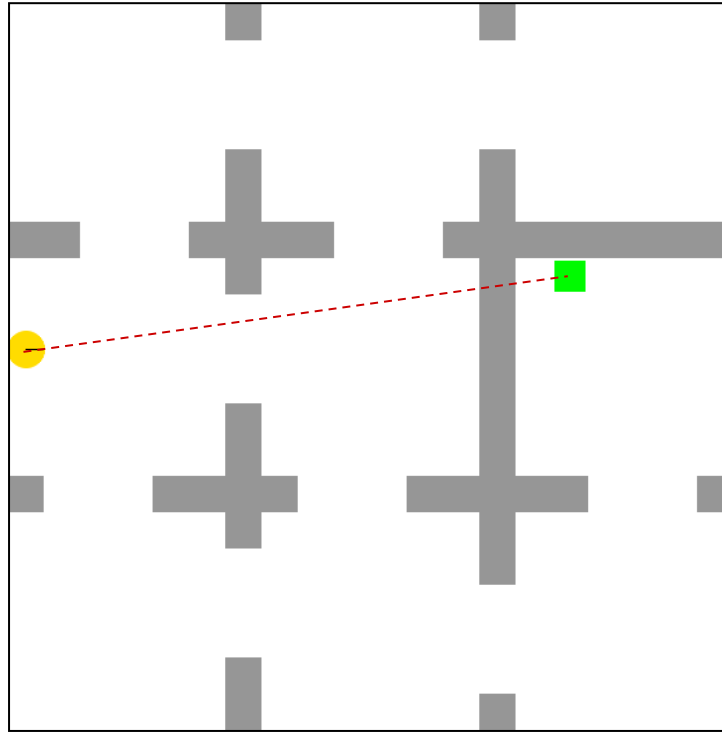
- Representa uma **estimativa do custo ou utilidade** de uma determinada situação ou acção para a resolução de um problema
- Reflecte **conhecimento acerca do domínio do problema**
- O seu valor é uma **aproximação ao valor real** da medida em causa
- **Exemplo**
 - Avaliação de um estado em relação ao objectivo a atingir
 - Independente do percurso até ao estado
- **Heurística admissível**
 - A estimativa de custo é sempre inferior ou igual ao custo efectivo mínimo
 - **Optimista**

FUNÇÃO HEURÍSTICA

- Como definir uma **heurística admissível**
 - No caso geral, uma heurística admissível é obtida através do **relaxamento de restrições** associadas ao problema
 - Exemplo: Navegação autónoma
 - h_1 - Distância de Manhattan
 - Corresponde a retirar a restrição:
 - » Não movimentação através de obstáculos
 - h_2 - Distância de Euclidiana
 - Corresponde a retirar as restrições:
 - » Não movimentação através de obstáculos
 - » Não movimentação em diagonal



HEURÍSTICA ADMISSÍVEL



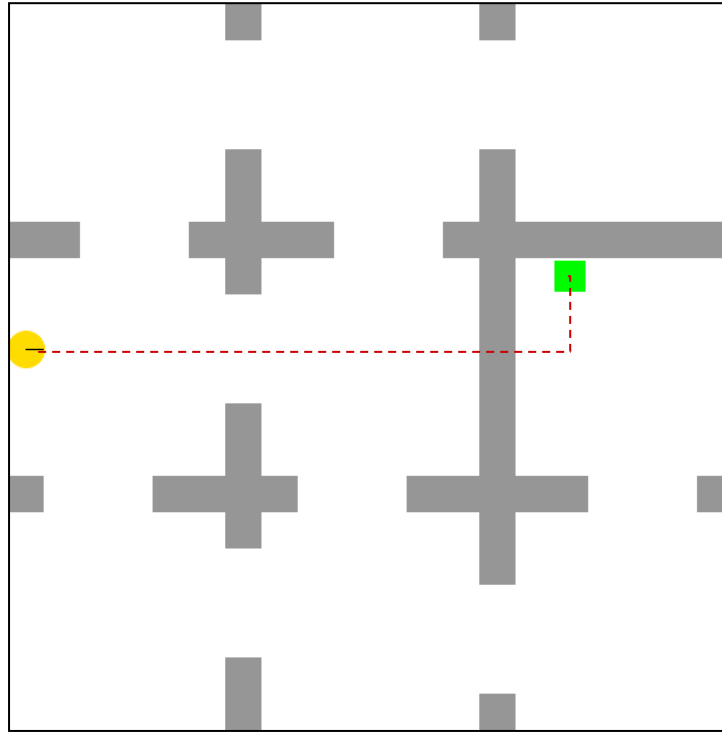
h_1 – Distância Euclidiana

$$h_1(n) = \sqrt{(x_n - x_{obj})^2 + (y_n - y_{obj})^2}$$

Admissível?

SIM

HEURÍSTICA ADMISSÍVEL



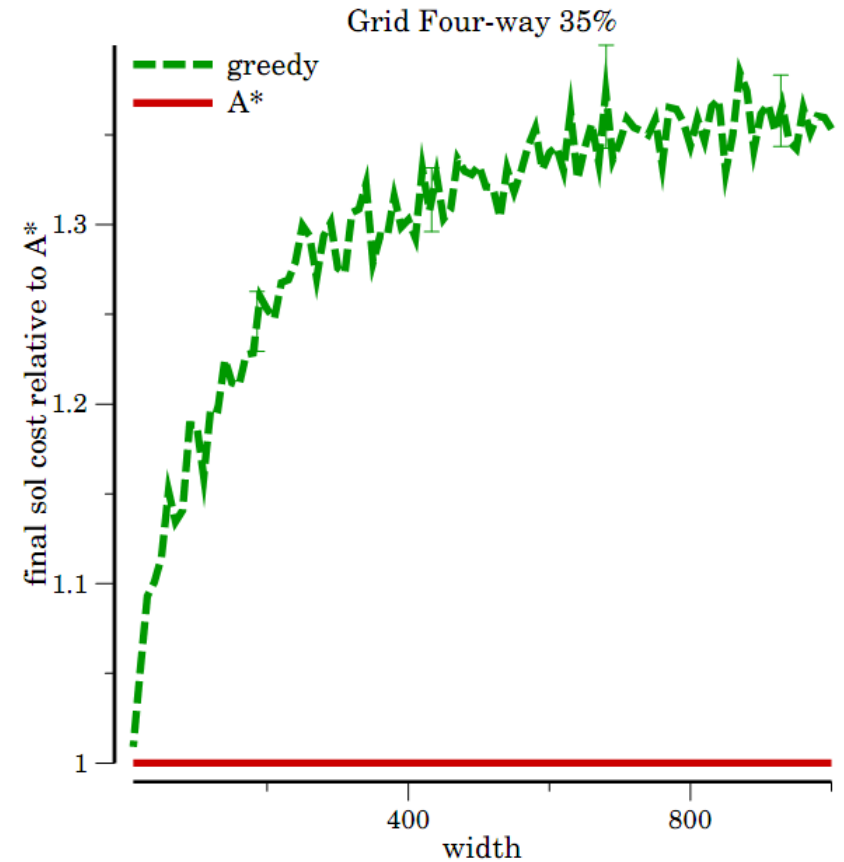
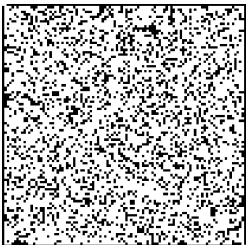
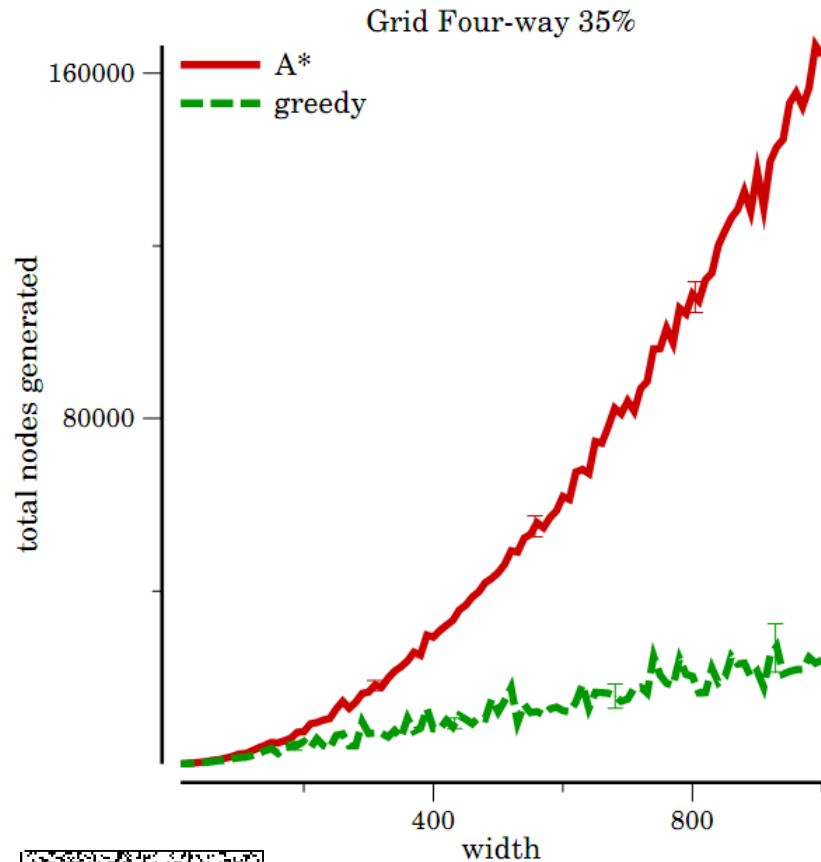
h_2 – Distância de Manhattan

$$h_2(n) = |x_n - x_{obj}| + |y_n - y_{obj}|$$

Admissível?

- SIM : Se não forem possíveis movimentos diagonais
- NÃO : Caso contrário

ESFORÇO vs. QUALIDADE DA SOLUÇÃO



PROCURA A* PONDERADA (*WEIGHTED A**)

ALGORITMO DE PROCURA ϵ -Óptimo

Definition 6.1. (ϵ -Optimality) A search algorithm is ϵ -optimal if it terminates with a solution of maximum cost $(1 + \epsilon) \cdot \delta(s, T)$, with ϵ denoting an arbitrary small positive constant.

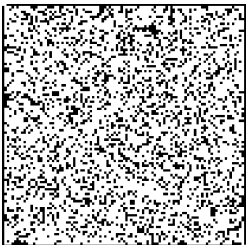
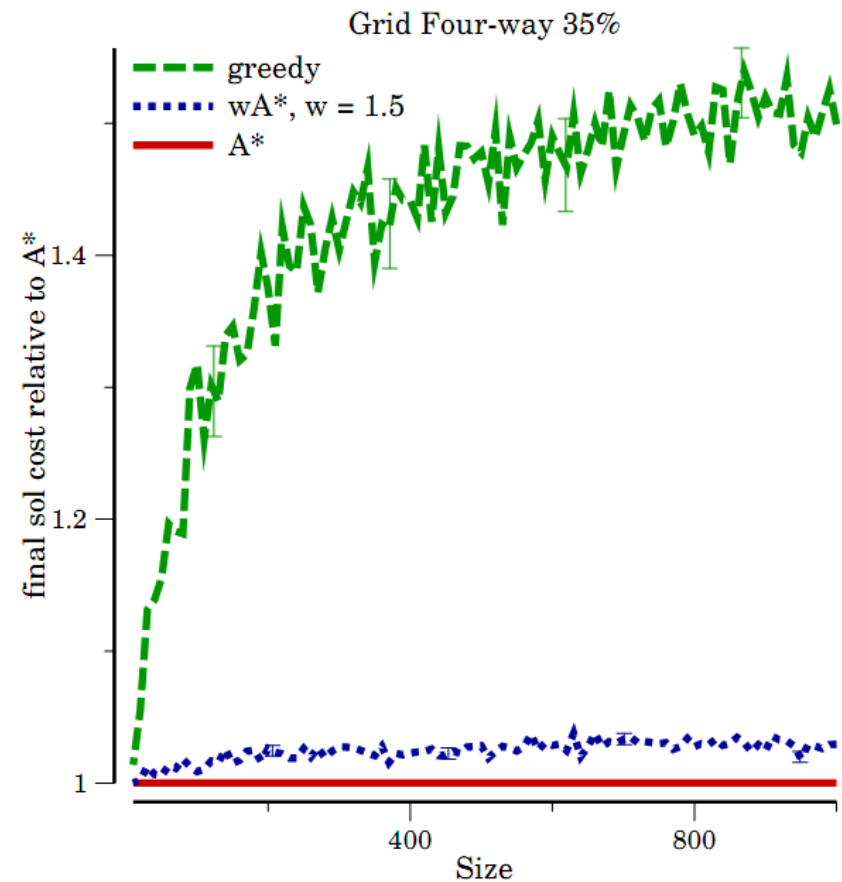
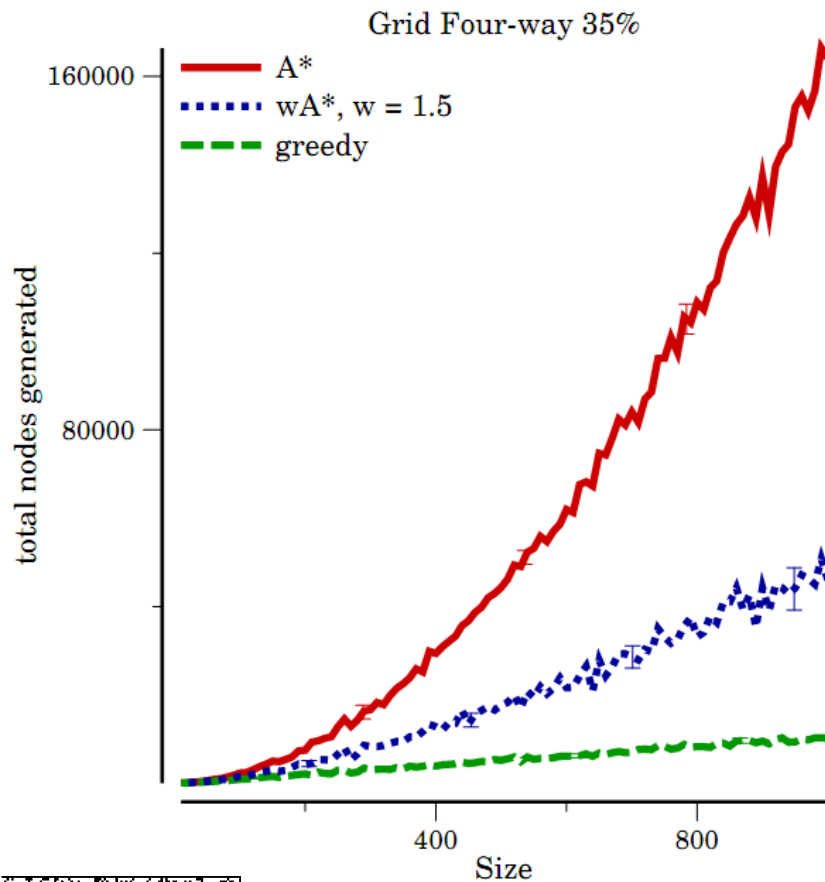
Lemma 6.2. A* where $f(u) = g(u) + (1 + \epsilon) \cdot h(u)$ for an admissible estimate h is ϵ -optimal.

Proof. For nodes u in *Open* that satisfy invariant (I) (Lemma 2.2) we have $f(u) = \delta(s, u) + h(u)$ and $g(u) = \delta(s, u)$ due to the reweighting process. Therefore,

$$\begin{aligned} f(u) &\leq \delta(s, u) + \delta(u, T) + \epsilon \cdot \delta(u, T) \\ &\leq \delta(s, T) + \epsilon \cdot \delta(u, T) \\ &\leq \delta(s, T) + \epsilon \cdot \delta(s, T) \\ &\leq (1 + \epsilon) \cdot \delta(s, T). \end{aligned}$$

Thus, if a node $t \in T$ is selected we have $f(t) \leq (1 + \epsilon) \cdot \delta(s, T)$.

PROCURA A* PONDERADA (*WEIGHTED A**)



ABSTRACÇÃO

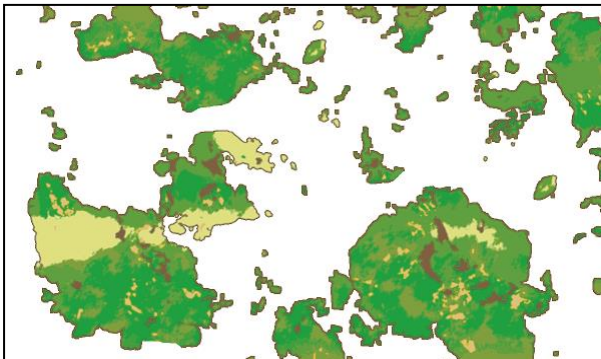
O processo de descrição de conhecimento a *diferentes níveis de detalhe* (quantidade de informação) e *tipos de representação* (estrutura da informação).

[Korf, 1980]

ABSTRACÇÃO

- **Redução de detalhe**

- Identificação de *características principais*
- Redução de complexidade da representação
- Redução de complexidade do processamento

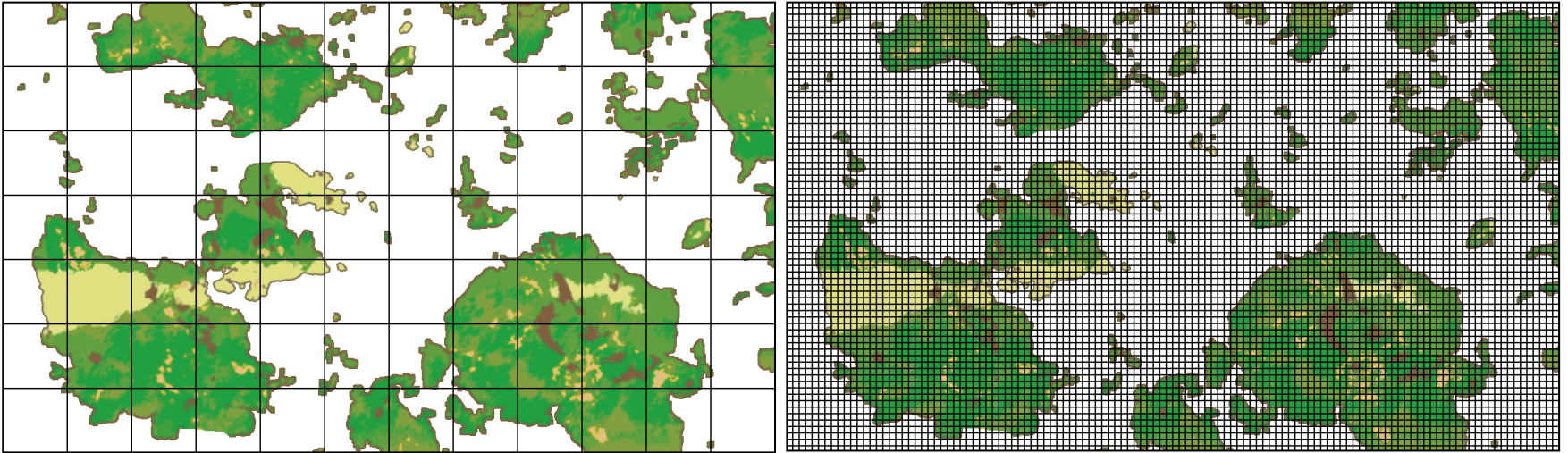


- **Problemas**

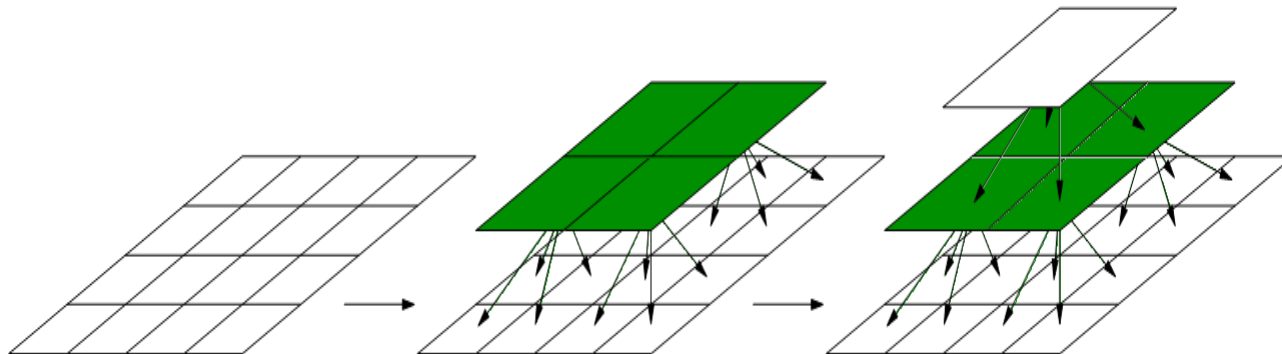
- Qualidade da solução
- Método não completo
 - Pode não ser possível encontrar solução, apesar de existir solução em níveis de detalhe superior

ABSTRACÇÃO

ABSTRACÇÃO LINEAR EM GRELHA



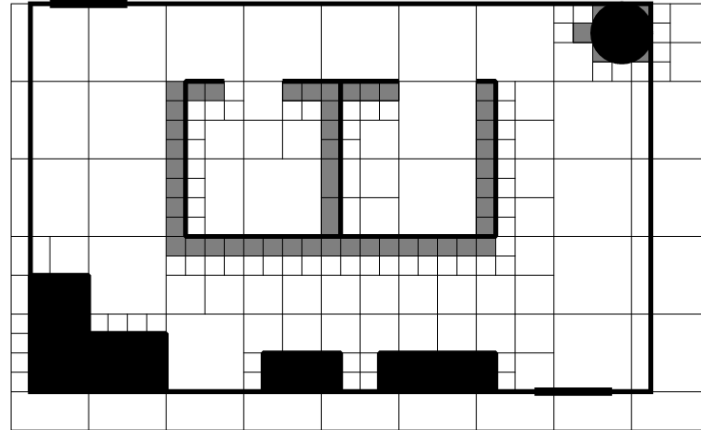
ABSTRACÇÃO LINEAR HIERÁRQUICA



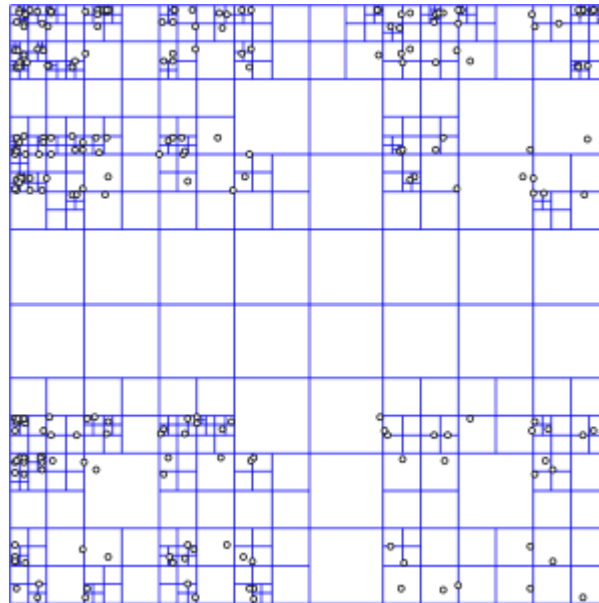
[Sturtevant & Buro, 2005]

ABSTRACÇÃO

ABSTRACÇÃO NÃO-LINEAR

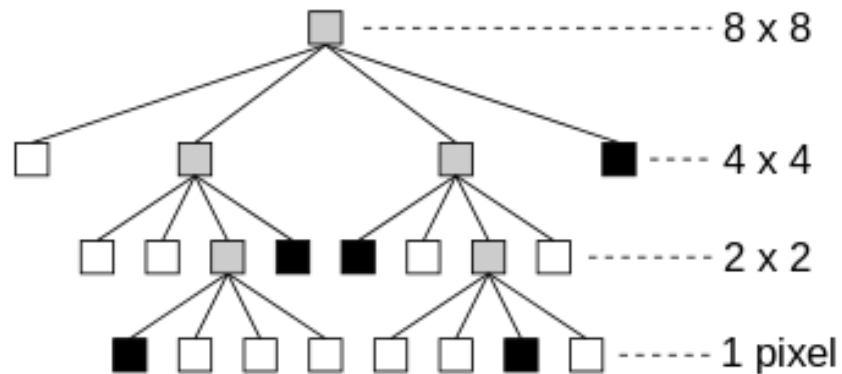
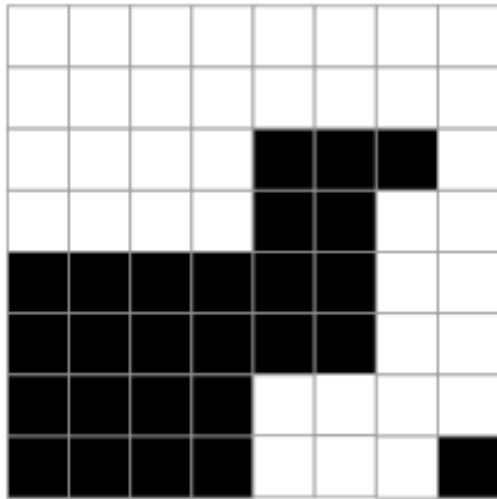


EXEMPLO: *QUADTREES*

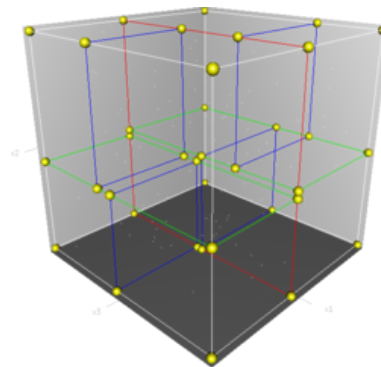


ABSTRACÇÃO NÃO-LINEAR

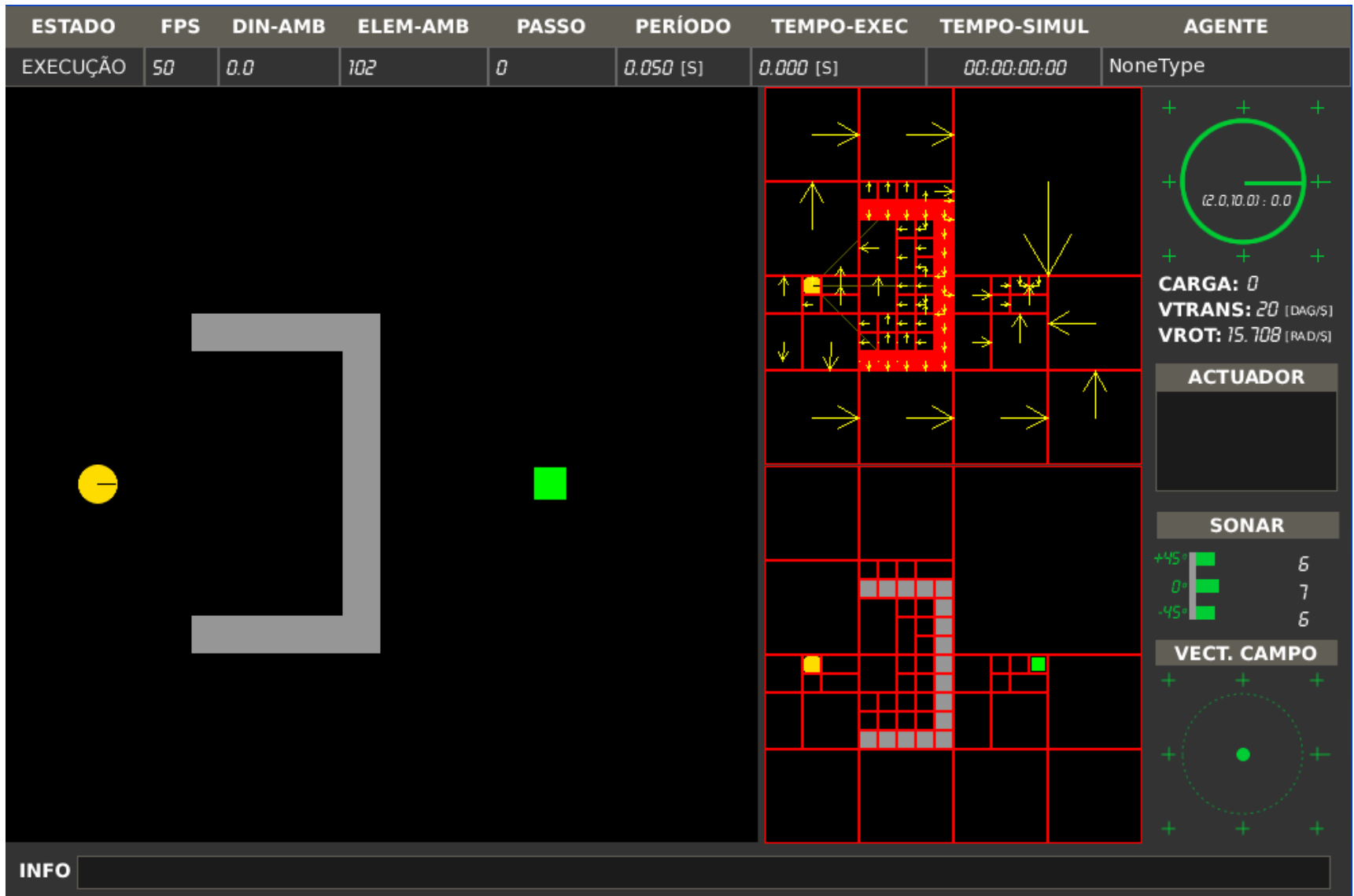
QUADTREE - Abstracção não-linear em espaços bidimensionais



KD-TREE - Abstracção não-linear em espaços n -dimensionais

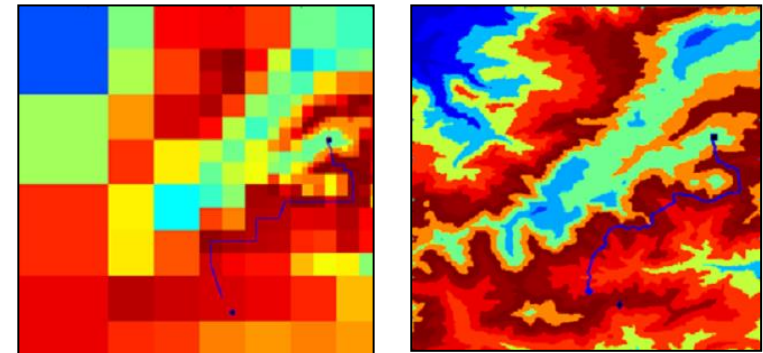
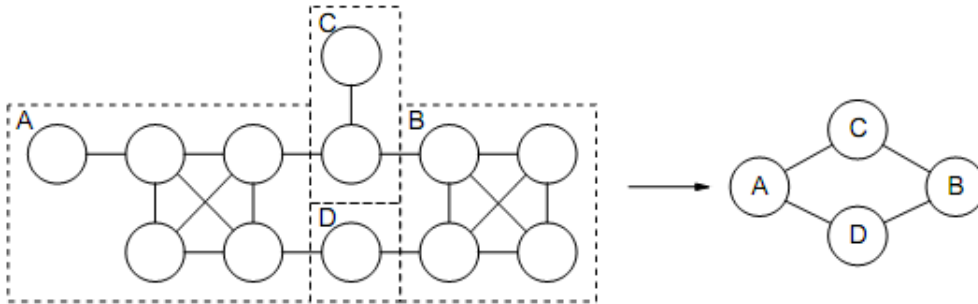
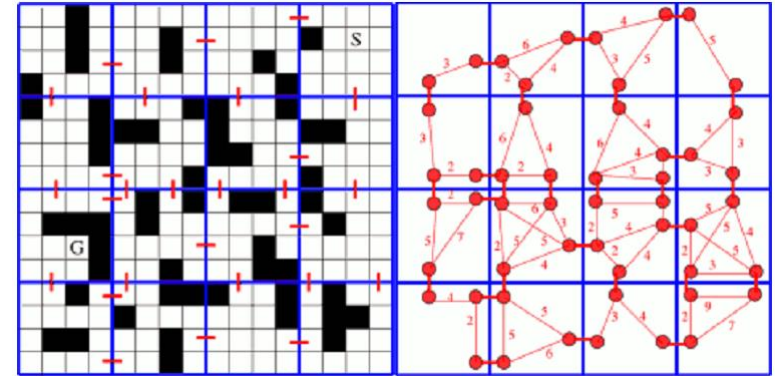
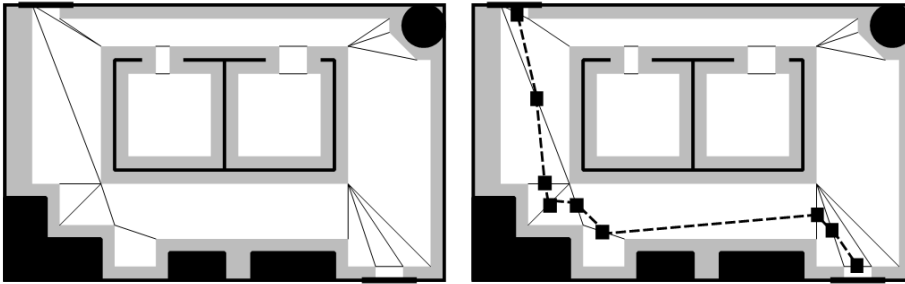


ABSTRACÇÃO COM BASE EM *QUADTREES*



ABSTRACÇÃO

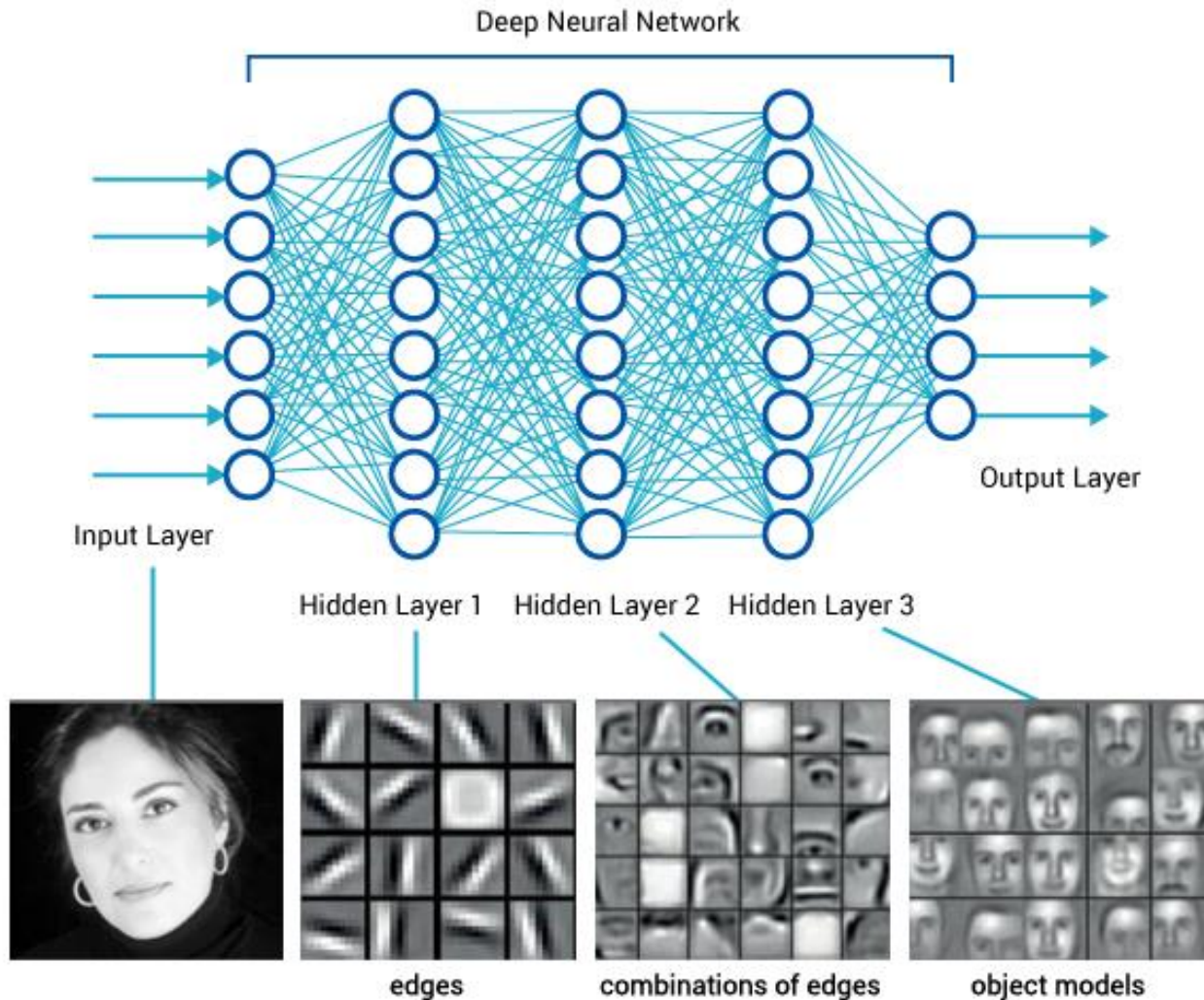
OUTRAS FORMAS DE ABSTRAÇÃO



A capacidade de abstracção depende de conhecimento do domínio do problema

ABSTRACÇÃO

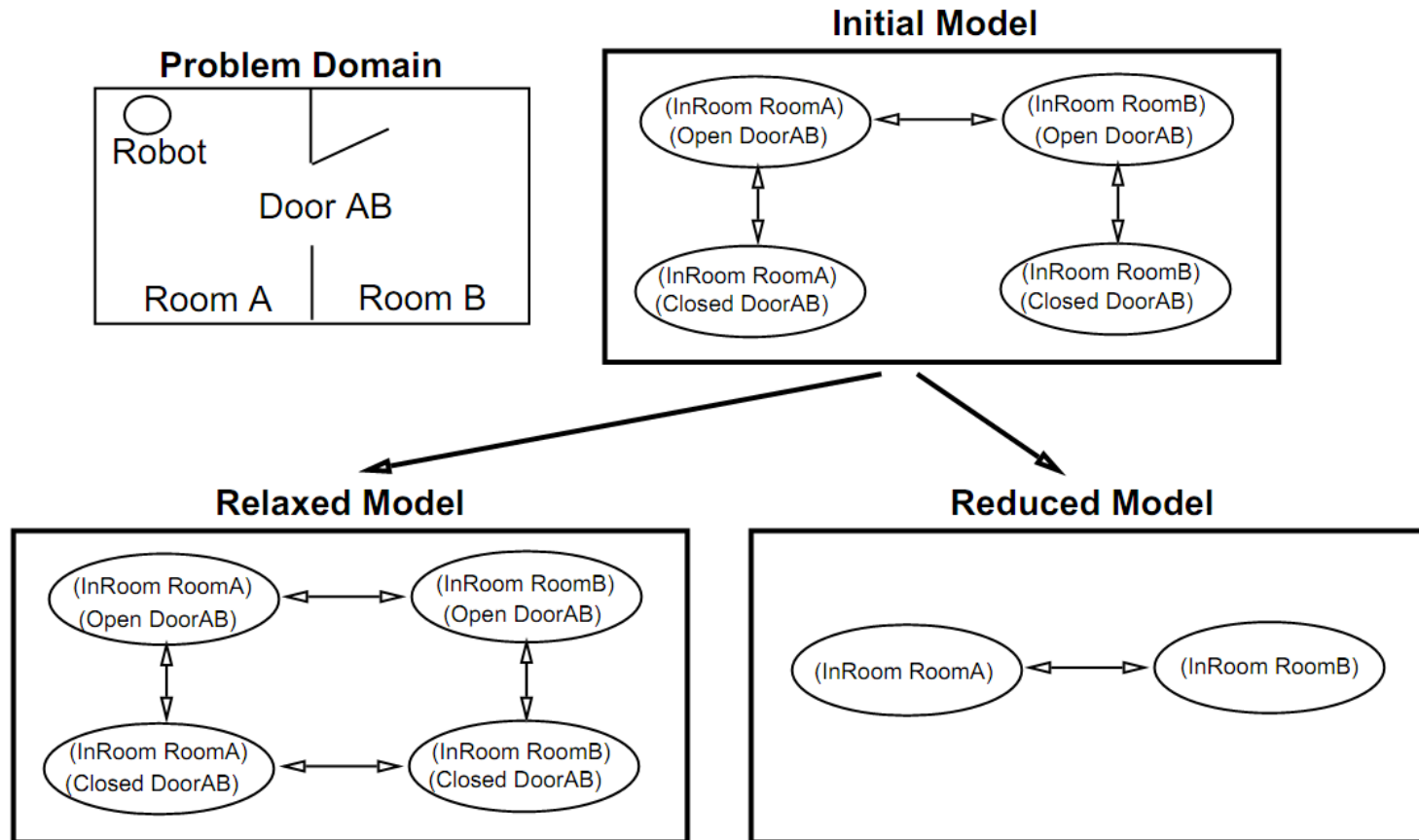
COM BASE EM REDES NEURONAIS



ABSTRACÇÃO

BASEADA EM CONHECIMENTO DO DOMÍNIO DO PROBLEMA

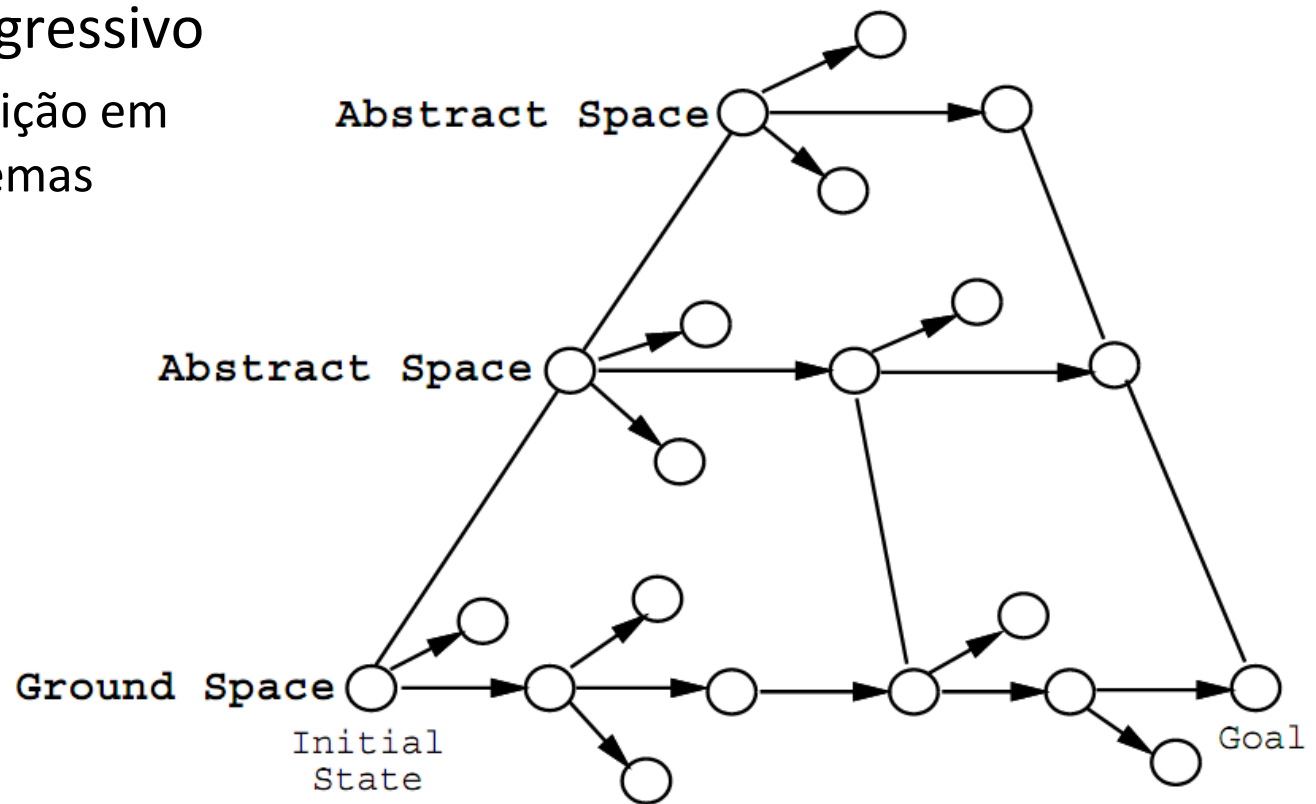
- *Redução* de detalhe
- *Relaxamento* de restrições



ABSTRACÇÃO

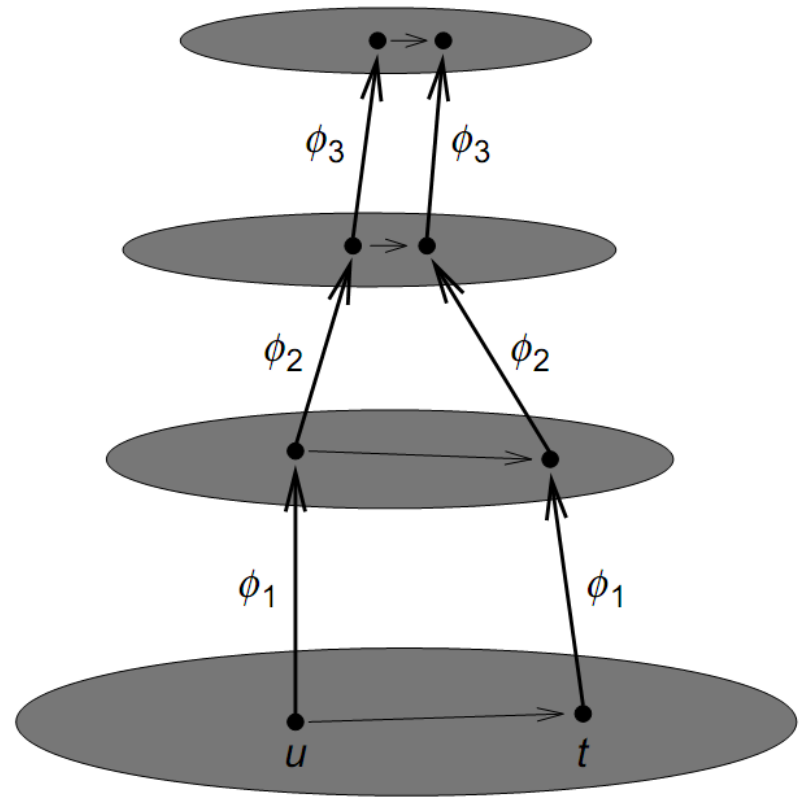
PLANEAMENTO HIERÁRQUICO

- Estratégias de alto nível
 - Planos abstractos
- Detalhe progressivo
 - Decomposição em sub-problemas



PROCURA A* HIERÁRQUICA

- Níveis de transformação
 - $\phi_1, \phi_2, \phi_3, \dots$
- Função heurística de um nível gerada a partir de níveis de abstração superiores
 - Exemplo
 - $\phi_2(\phi_1(u)), \phi_2(\phi_1(t))$
- Função heurística não consistente
- Pré-computação da função heurística a níveis abstractos para optimização de desempenho



MODULARIZAÇÃO

- Associada à abstracção
- Redução de complexidade por decomposição combinatória
 - divisão em sub-problemas com relação causal
 - Resolução sequencial
 - Soluções parciais
 - Composição da solução global em função das soluções parciais
- Exemplo
 - Navegação através de pontos de referência (*waypoints*)



EXEMPLO: 15-PUZZLE

$N \approx 10^{12}$

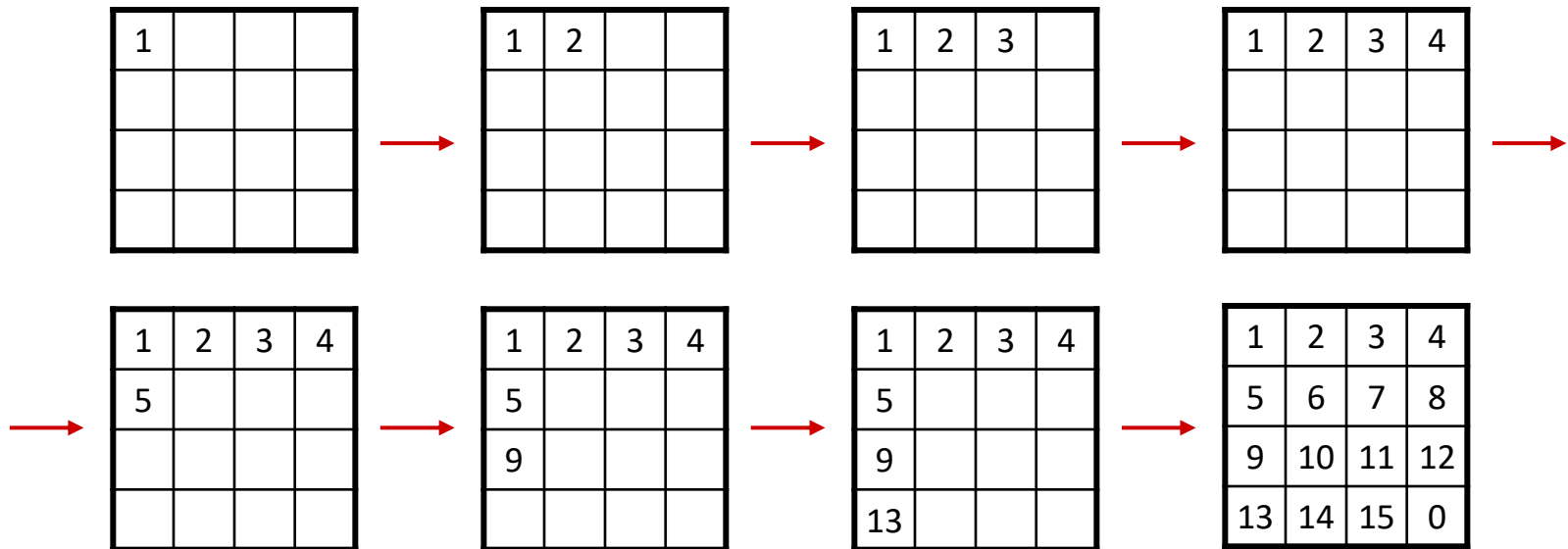
9	2	12	6
5	7	14	13
3	4	1	11
15	10	8	0



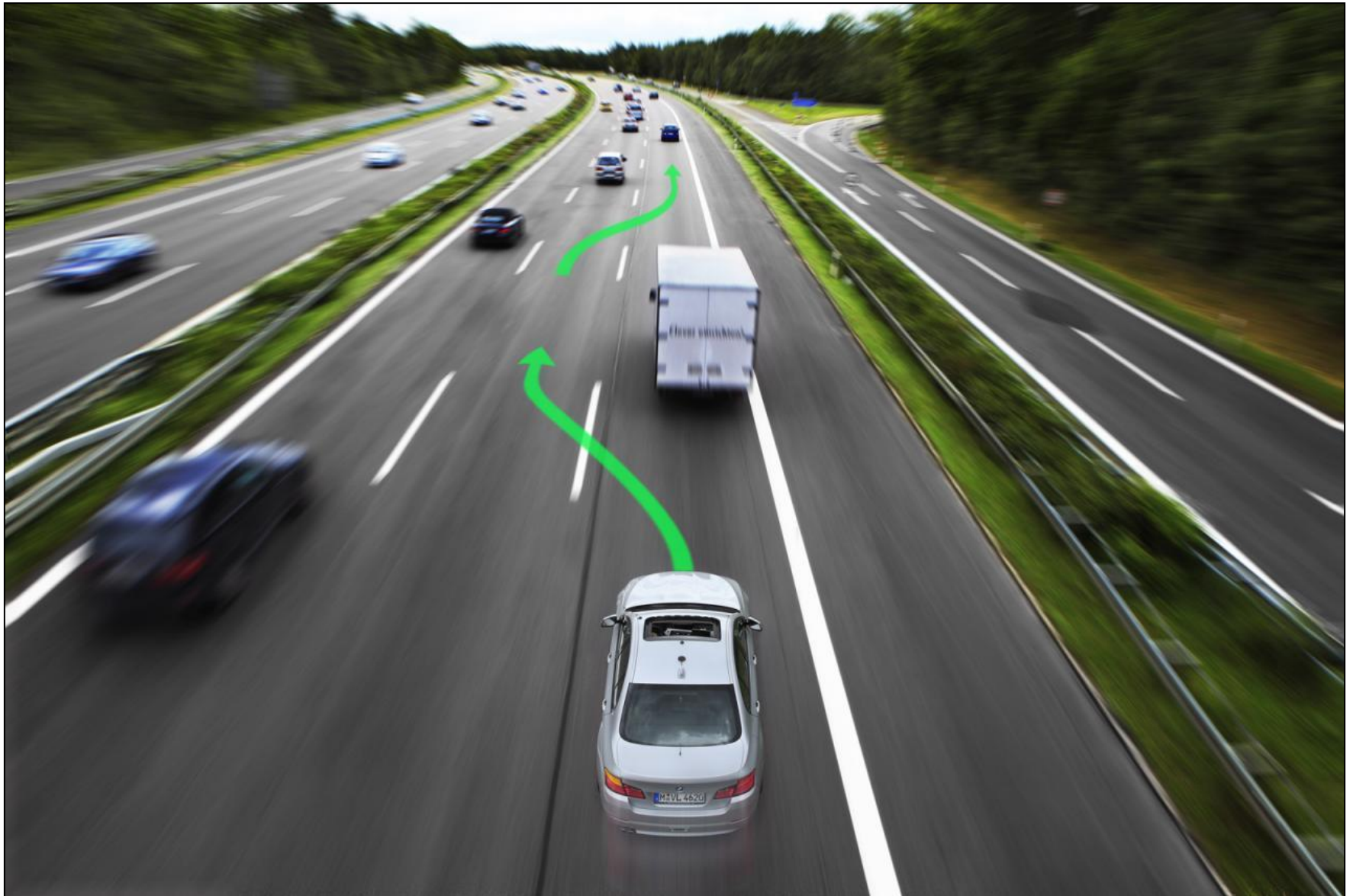
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	0

Plano abstracto guia raciocínio concreto

- Decomposição em sub-problemas
- Resolução independente de cada sub-problema



OPERAÇÃO EM TEMPO-REAL EM AMBIENTES COMPLEXOS E DINÂMICOS



FOCAGEM

- **SELECÇÃO**
 - Os recursos computacionais são orientados para aspectos específicos de um problema
- **ELIMINAÇÃO**
 - Os restantes aspectos não são considerados
- **PROBLEMA DA RELEVÂNCIA**
 - Quais os aspectos a seleccionar
 - Quais os aspectos a eliminar
- **VERTENTES DE FOCAGEM**
 - Espaço (Memória)
 - Tempo

FOCAGEM

- VERTENTES DE FOCAGEM

- ESPAÇO

- Limitação de memória

- *Iterative Deepening A** (IDA*)
 - *Memory Bounded A** (MA*, SMA*)
 - *Beam Search*

- Mecanismos de atenção

- TEMPO

- Limitação de tempo de processamento

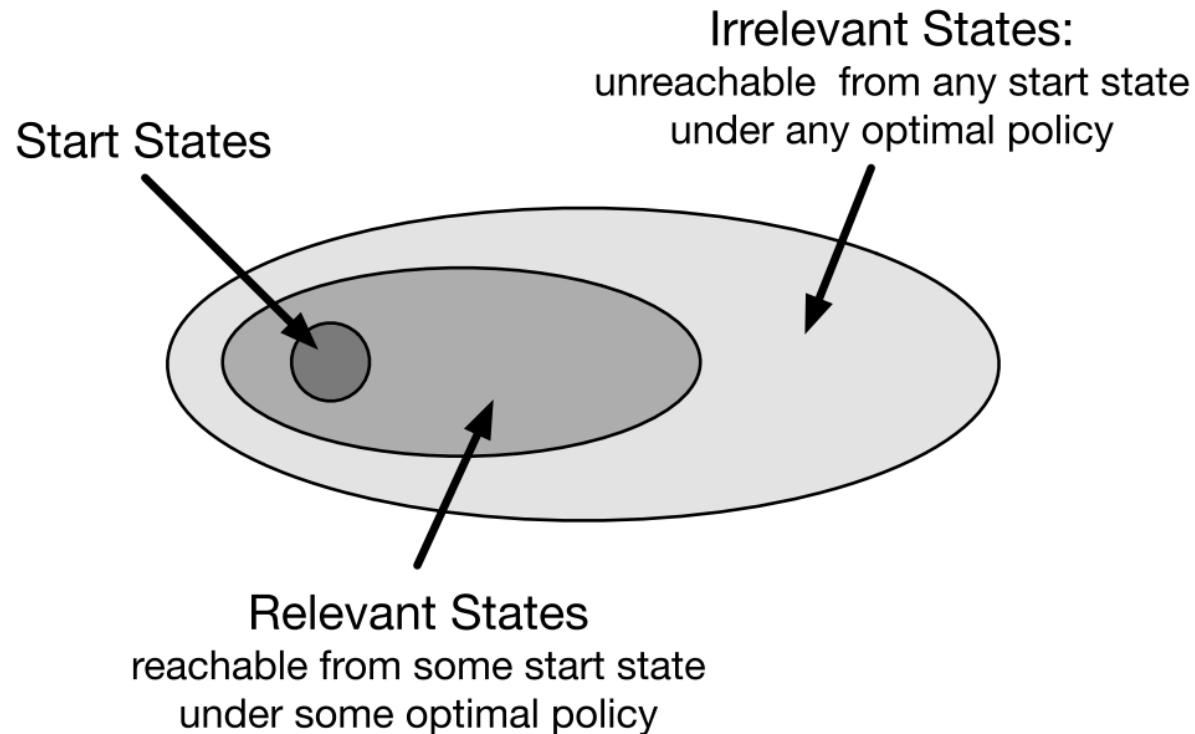
- Algoritmos *Anytime*

- Raciocínio em tempo-real

- *Real-Time Dynamic Programing* (RTDP)
 - *Real-Time Adaptive A** (RTAA*)

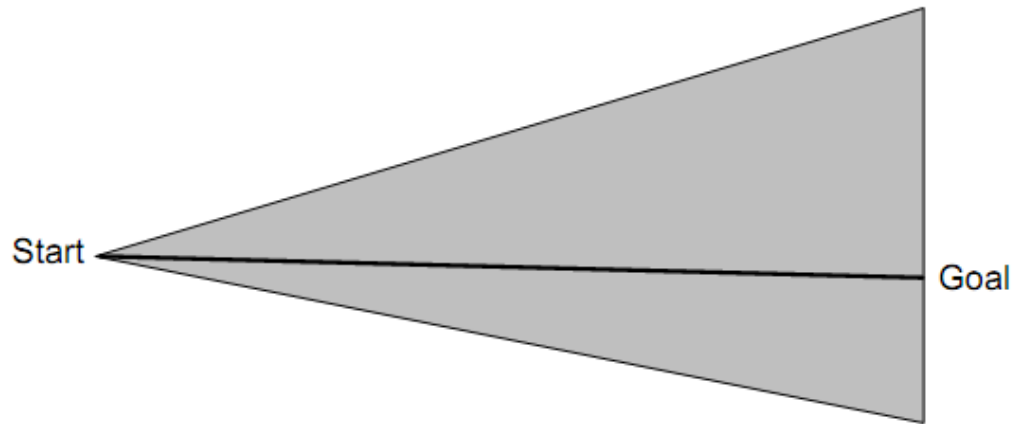
FOCAGEM

RACIOCÍNIO DE CONTEXTO LOCAL

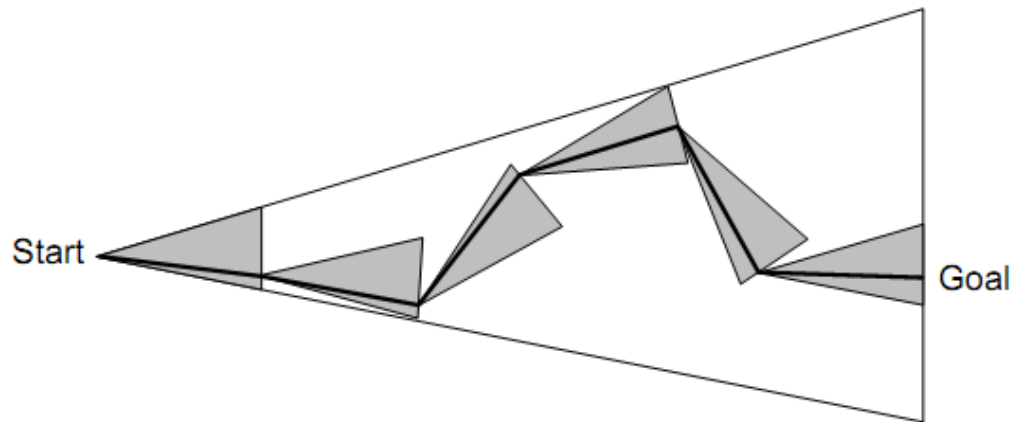


MÉTODOS DE PROCURA EM TEMPO REAL

PROCURA CENTRADA NO AGENTE



(a) Optimal offline search



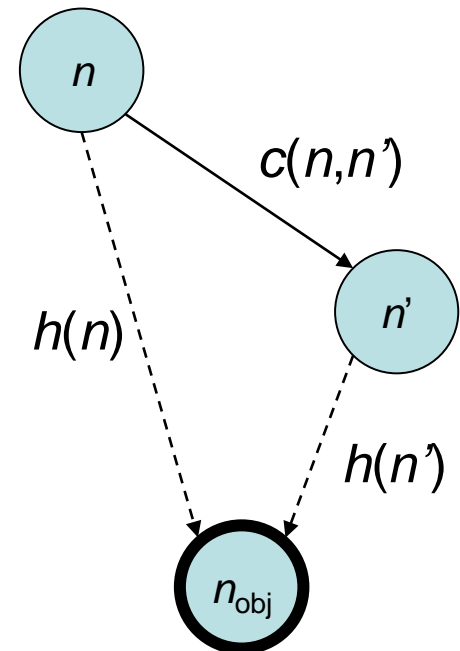
(b) Agent-centered search

MÉTODOS DE PROCURA EM TEMPO REAL

- Intercala planeamento com execução
- Composição de procuras
 - Procura âmbito local
 - Solução parcial
 - Estimativa de valor global
- Actualização de valor global
 - Para resolver problema de soluções localmente óptimas mas globalmente não completas
- Diferentes métodos
 - Exemplo
 - LRTA* (*Learning Real Time A**)
 - RTAA* (*Real Time Adaptive A**)

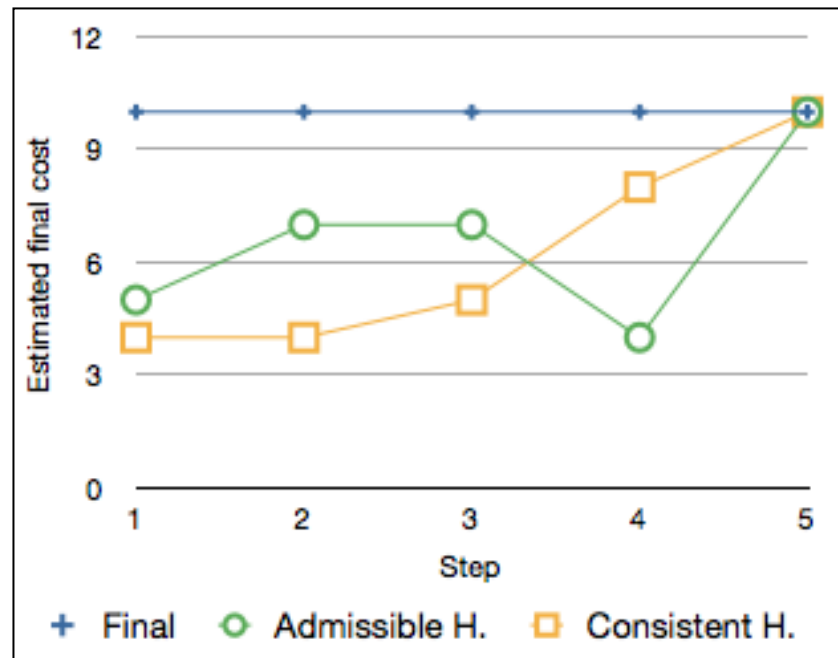
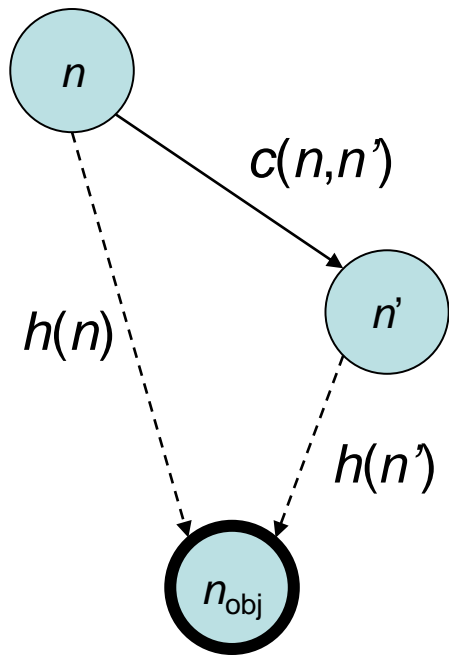
PROCURA A*

- **Heurística consistente** (ou monótona)
 - Para cada nó n , seu sucessor n' e custo de transição $c(n, n')$
 - $h(n) \leq c(n, n') + h(n')$
 - Para um nó objectivo
 - $h(n_{\text{obj}}) = 0$
- Uma heurística consistente é também admissível
- Uma heurística admissível pode não ser consistente



RTAA* - *REAL TIME ADAPTIVE A**

HEURÍSTICA CONSISTENTE



$$h(n) \leq c(n, n') + h(n')$$

MÉTODOS DE PROCURA EM TEMPO REAL

RTAA* - *REAL TIME ADAPTIVE A**

Heurística consistente

$$g(v) + \delta(v, T) \geq \delta(s, T)$$

$$\delta(v, T) \geq \delta(s, T) - g(v)$$

$$\delta(v, T) \geq f(\bar{u}) - g(v)$$

Procedure RTAA*

Input: Search task with initial h -values

Side Effect: Updated h -values

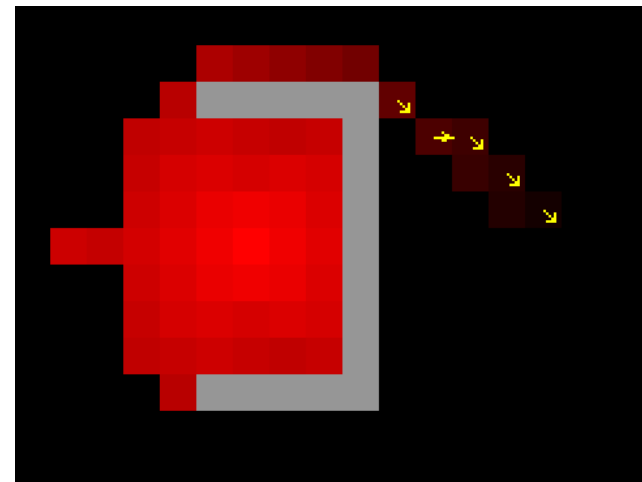
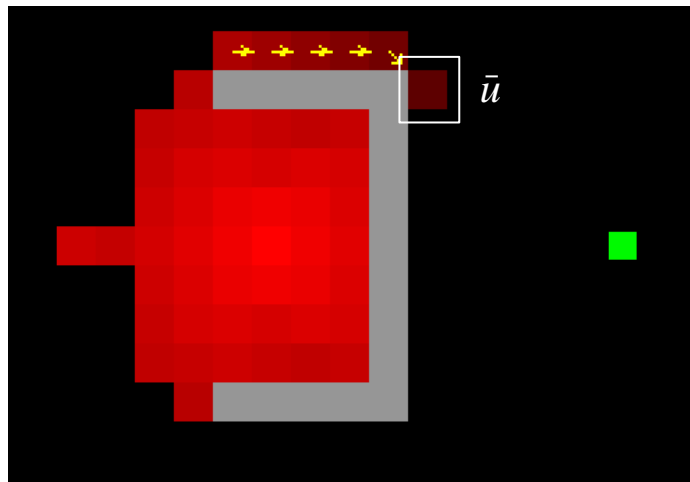
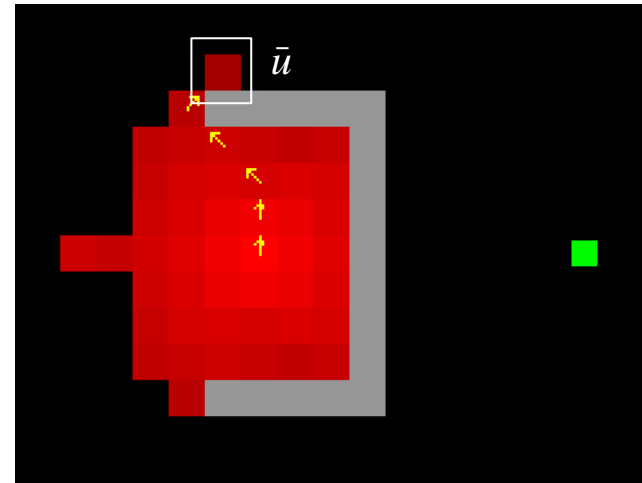
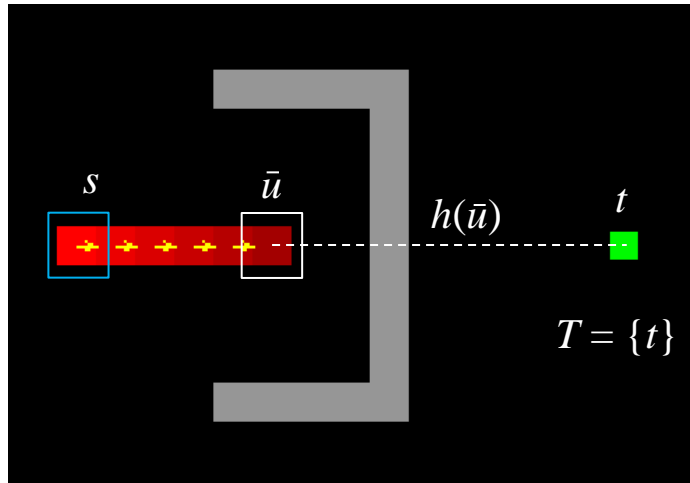
```
 $u \leftarrow s$  ;; Start in start state
while ( $u \notin T$ ) ;; While goal not achieved
     $(\bar{u}, \text{Closed}) \leftarrow A^*(u, \text{lookahead})$  ;;  $A^*$  search until lookahead states have been expanded
    if ( $\bar{u} = \text{false}$ ) return false ;; Return if goal unachievable
    for each  $v \in \text{Closed}$  ;; For each expanded state
         $h(v) \leftarrow g(\bar{u}) + h(\bar{u}) - g(v)$  ;; Update  $h$ -value
    repeat ;; Repeat
         $a \leftarrow \text{SelectAction}(A(u))$  ;; Select action on minimal-cost path from  $u$  to  $\bar{u}$ 
         $u \leftarrow a(u)$  ;; Execute action
    until ( $u \notin \text{Closed}$ ) ;; Until local search space exited (optional)
```

Algorithm 11.5

RTAA*.

\bar{u} - state that was about to be expanded when the A^* search terminates

RTAA* - *REAL TIME ADAPTIVE A**



MÉTODOS DE PROCURA EM TEMPO REAL

- Características principais
 - **Utilização de conhecimento heurístico** para guiar a procura
 - **Regulação do esforço de procura** entre cada acção
 - **Resposta rápida**
 - Espaço local de procura menor
 - Soluções parciais de menor qualidade
 - » Maior esforço de execução
 - **Resposta lenta**
 - Espaço de procura maior
 - Soluções parciais de melhor qualidade
 - » Menor esforço de execução
 - **Melhoria progressiva** do desempenho

REFERÊNCIAS

[Korf, 1980]

R. Korf, *Toward a model of representation changes*, *Artificial Intelligence*, Volume 14, Issue 1, 1980

[Edelkamp & Schrodl, 2012]

S. Edelkamp, S. Schrodl, *Heuristic Search Theory and Applications*, Morgan Kaufmann, 2012

[Pearl, 1984]

J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, 1984

[Murphy, 2000]

R. Murphy, *An Introduction to AI Robotics*, MIT Press, 2000

[Thayer & Ruml, 2007]

J. Thayer, W. Ruml, *A Survey of Suboptimal Search Algorithms*, University of New Hampshire, 2007

[Sturtevant & Buro, 2005]

N. Sturtevant, M. Buro, *Partial Pathfinding Using Map Abstraction and Refinement*, Proceedings of the 20th national conference on artificial intelligence (AAAI-05), 2005

[Botea *et al.*, 2004]

A. Botea M. Muller, J. Schaeffer, *Near Optimal Hierarchical Path-Finding*, Journal of Game Development, Volume 1, 2004

[Cowlagi , Tsiotras, 2007]

R. Cowlagi , P. Tsiotras, *Beyond Quadtrees: Cell Decompositions for Path Planning using Wavelet Transforms*, Proceedings of the 46th IEEE Conference on Decision and Control, 2007

[Knoblock, 1991]

A. Knoblock, *Automatically Generating Abstractions for Problem Solving*, PhD Thesis, Carnegie Mellon University, 1991