



INTELIGÊNCIA ARTIFICIAL PARA SISTEMAS AUTÓNOMOS

FINAL

André Fonseca
a39758@alunos.isel.pt

Daniel Santos
a32078@alunos.isel.pt

Professor
Paulo Vieira
pjvieira@deetc.isel.pt

Junho, 2018

INTRODUÇÃO

A definição de inteligência é a habilidade de aprender e resolver problemas, independentemente de se tratar de inteligência humana ou máquina. Quando se trata de uma entidade máquina denomina-se de inteligência artificial que pode estar presente não só em máquinas, como também em software.

“The science and engineering of making intelligent machines”

John McCarthy

A inteligência artificial consiste no estudo e na arquitectura de agentes inteligentes - ou seja, um sistema que percepçiona o meio ambiente envolvente e toma acções que maximizem a sua probabilidade de sucesso (esta é a definição adoptada por Peter norvig em “Artificial Intelligence: A Modern Approach”).

A potencialidade da inteligência artificial pode ser comparada ao aparecimento de novos processos de manufactura, tal como no período da revolução industrial. Uma possível segunda revolução da humanidade.

Ao longo do tempo, a definição de inteligência artificial tem vindo a sofrer alterações. Inicialmente, em 1985, era esperado que fosse a modelação do pensamento humano através da mente de um computador, no sentido literal. Em 1991, surgiu uma nova definição que sugeria que os computadores deviam ter um comportamento humano. Mais tarde, foi definido o termo de racionalidade em que foram estudadas as capacidades mentais de forma a desenvolver modelos computacionais. Por fim, em 1998, definiu-se que a inteligência artificial baseia-se em agir racionalmente - através do estudo e desenvolvimento de agentes inteligentes. Actualmente, não se mencionam os primeiros termos adoptados na definição de inteligência artificial. Isto é: fazer com que as máquinas tenham uma mente ou com que simulem acções humanas. Apenas se pretende que as máquinas realizem comportamentos racionais.

Desenhar sistemas inteligentes que tomam acções de forma a maximizarem o seu objectivo, ou seja, agentes que tomam a decisão certa tendo em conta o conhecimento que tem disponível. Este conhecimento pode ser um espectro muito grande de variáveis, tais como o ambiente, experiência previamente adquirida, etc.

Para qualquer tipo de problema da actualidade pode-se encontrar uma solução do estado de arte da inteligência artificial. Desde o reconhecimento de fala (através de técnicas de processamento de linguagem natural em assistentes pessoais); de caligrafia (em tradução de linguagens); na área da robótica (como cirurgia robótica e navegação); em sistemas de recomendação (que têm como base os acontecimentos históricos de forma a sugerir possíveis acções); etc.

Nesse sentido, o terceiro projecto tem como objectivos a implementação e concretização de modelos e arquitecturas de agentes inteligentes, tendo como base uma plataforma de simulação de agentes. Pretende-se a concepção de um agente para operar num ambiente composto por três tipos de elementos: agente, alvo e obstáculo. O ambiente é estático,

alterando-se apenas por acção do agente. Este deverá ter a capacidade de recolher alvos. Quando o agente recolhe um alvo, transporta-o como carga.

O projecto será dividido em várias etapas de implementação e concepção:

1. Agente reactivo para recolha de alvos.
2. Agente deliberativo para recolha de alvos - com base no raciocínio automático, através da procura em espaços de estados; capaz de minimizar a distância percorrida na recolha de alvos.
3. Agente deliberativo para recolha de alvos - com base em processos de decisão de Markov; capaz de minimizar a distância percorrida na recolha de alvos.
4. Agente capaz de aprender a partir da experiência - com base em mecanismos de aprendizagem por reforço.

Links

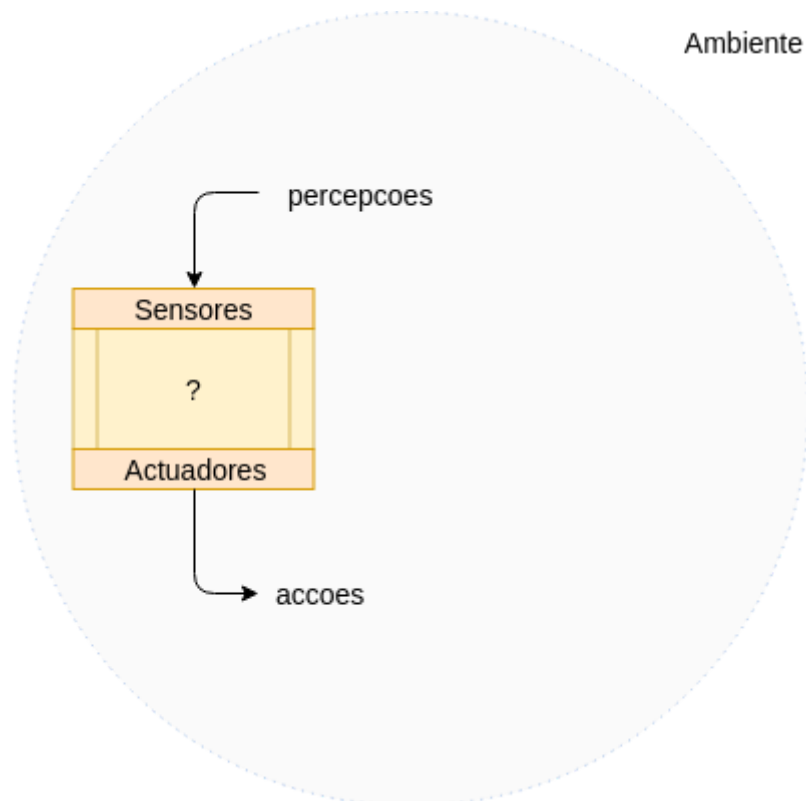
❏ <https://gitlab.com/andrefonseca/ISEL-LEIM-IASA>

AGENTES INTELIGENTES

AGENTE

O agente é um sistema, como por exemplo uma aplicação de software ou um robot que observa - através de sensores - e realiza acções - através de actuadores, num meio ambiente, de maneira a atingir os seus objectivos. Estes podem também aprender ou usar conhecimento para atingir os seus objectivos.

$$f : P \rightarrow A$$



RACIONALIDADE

Um agente racional é aquele que realiza a acção correcta. No sentido de alcançar a acção correcta, este segue uma sequência de acções de acordo com a percepção recebida pelos seus sensores. Esta sequência faz com que o ambiente passe por uma sequência de estados. Caso a sequência seja a desejada, então o agente teve um bom desempenho. Esta noção é definida como *performance measure* e avalia qualquer sequência de estados do ambiente.

A racionalidade é definida como tomada / a escolha (?) da melhor acção, tendo em conta o conhecimento adquirido até ao momento. Esta definição pode ser traduzida para o contexto de agente como: para cada possível sequência de percepções, o agente deve seleccionar a acção que maximiza o seu desempenho consoante a sequência de percepções e,

também, quaisquer conhecimento e experiência que o agente tenha. Assim, conclui-se que a racionalidade depende dos seguintes factores:

- O desempenho (*performance measure*) que determina o critério de sucesso.
- O conhecimento *à priori* do ambiente.
- As acções possíveis de tomar.
- A sequência de acções até ao momento.

A recolha de informações é um processo importante na definição de racionalidade, pois pode fornecer dados complementares para o processo de raciocínio. Por vezes, o processo de recolha de informações pode ser realizado pela exploração do ambiente.

O processo de aprendizagem faz também parte da definição de racionalidade. Um agente pode ou não ter algum conhecimento sobre o ambiente em que está inserido, mas à medida que este aprende o seu conhecimento sobre o ambiente vai-se alterando. É assim esperado que o agente maximize o que aprende de cada percepção.

Quando um agente tem pouca ou nenhuma experiência deve de explorar, através de acções aleatórias e predefinidas ou ter algum conhecimento prévio e a capacidade de aprender. Tal como a evolução dos seres vivos fornece reflexos para que os animais possam sobreviver o suficiente até aprenderem por si mesmos, tornando-se independentes e autónomos. Assim, a adição do processo de aprendizagem permite ao agente ter sucesso numa grande variedade de ambientes.

AMBIENTES

Para resolver um problema de forma autónoma é necessário caracterizar o tipo de ambiente no qual o agente está inserido, isto porque um determinado tipo de ambiente afecta directamente a arquitectura da solução.

A quantidade de possíveis ambientes é, obviamente, vasta, mas no entanto, é possível identificar um conjunto de características.

- **Totalmente ou parcialmente observável:** Um ambiente é considerado totalmente observável se os sensores do agente conseguem captar todos os aspectos relevantes para a tomada de decisão - a mais conveniente, porque não é necessário manter um estado interno do ambiente. No entanto, pode ser parcialmente observável, devido a ruído captado ou, também, às imprecisões nos sensores ou, até mesmo, à falta de informação.
- **Agente único ou múltiplos agentes:** Caracteriza-se pelo número de objectos/agentes presentes num determinado ambiente. Quando existe a presença de mais do que um único agente é importante definir como irá ser realizada a interacção entre agentes. Se é uma interacção cooperativa, onde maximizar o desempenho de um agente é também benéfico para o desempenho dos outros; ou competitiva, onde maximizar o desempenho de um resulta na diminuição do desempenho de outros agentes.
- **Determinístico ou estocástico:** Se o próximo estado de um ambiente for determinado através do seu estado actual e da acção executada pelo agente - ambiente determinístico. Caso o ambiente seja apenas parcialmente observável poderá ser estocástico, isto porque na maioria das situações reais complexas é praticamente impossível observar todos os aspectos. Então, por fins práticos, considera-se de estocástico. Um ambiente diz-se incerto quando não é totalmente observável ou não determinístico.
- **Episódico ou sequencial:** Um ambiente sequencial necessita de memória para determinar qual a próxima melhor acção. Esta pode afectar todas as futuras decisões. Por outro lado, os ambientes episódicos estão divididos em pequenas tarefas, nas quais não necessitam de memória - pois a acção a ser seleccionada apenas depende da percepção actual. Um exemplo seria a análise de uma imagem de raios-X para determinar se existe algum problema de saúde, em que uma imagem não está relacionada com qualquer uma das outras e a decisão actual em nada afecta a próxima.
- **Estático ou dinâmico:** Ambientes estáticos são aqueles que não sofrem alterações enquanto um agente faz a deliberação de qual a acção a tomar. Caso o ambiente sofra alterações então é dinâmico; caso não se altere com o tempo, mas antes a classificação do desempenho do agente seja alterada com o tempo estamos perante um estado semi-dinâmico.

- Discreto ou contínuo: Definido pela maneira como são analisados o tempo, as percepções e acções do agente. Um ambiente discreto define-se por um número finito de estados ou intervalos de tempo, por exemplo um jogo de xadrez em que existe um conjunto de percepções e acções discretas. Um ambiente diz-se contínuo quando pode ser medido quantitativamente qualquer que seja o nível de precisão. Por exemplo, conduzir um carro em que as acções são contínuas e vão variando consoante o tempo (velocidade).
- Conhecido ou desconhecido: Esta distinção refere-se mais precisamente ao agente ou ao constructor do agente. É o conhecimento do estado seguinte ou a sua probabilidade, quando num ambiente estocástico, do ambiente dada uma ou uma sequência de acções - independentemente se o ambiente é totalmente observável ou não.

O ambiente mais difícil de se encontrar é um parcialmente observável, com múltiplos agentes, estocástico, sequencial, dinâmico, contínuo e desconhecido.

ARQUITECTURAS DE AGENTE

A inteligência artificial de um agente tem como principal função o mapeamento de percepções para acções. Assume-se que este programa de software irá ser executado em algum tipo de dispositivo de computação com componentes de sensores e actuadores, físicos ou virtuais.

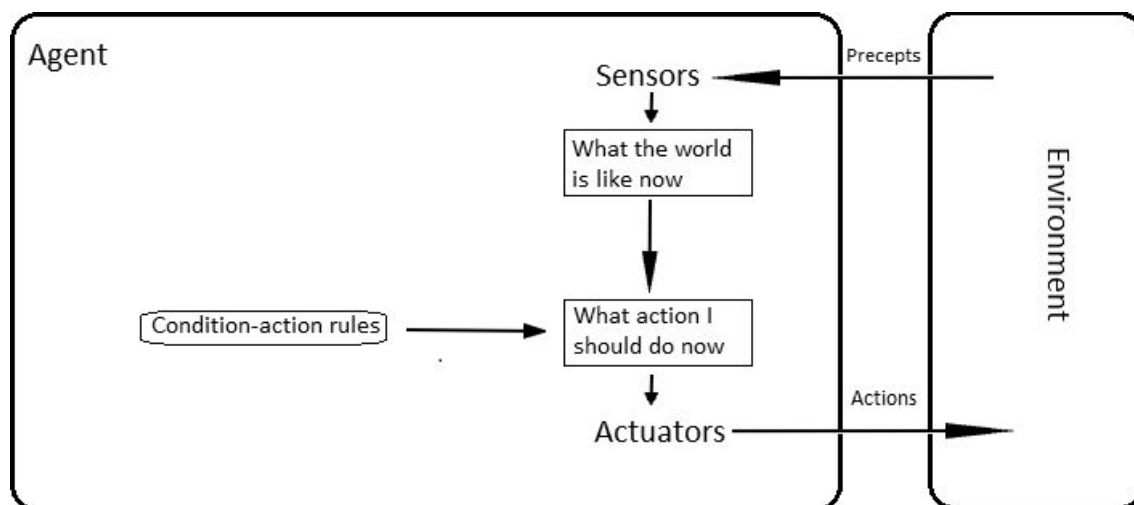
O programa de software precisa de ser adequado aos componentes do dispositivo, as acções determinadas pelo programa de software e os actuadores têm de ser compatíveis; o mesmo com os sensores e percepções esperadas.

AGENTES REACTIVOS

Agentes reactivos simples actuam com base na percepção actual, ignoram as percepções anteriores e relacionam a percepção e uma acção como base em regras previamente implementadas (*if condition then action*).

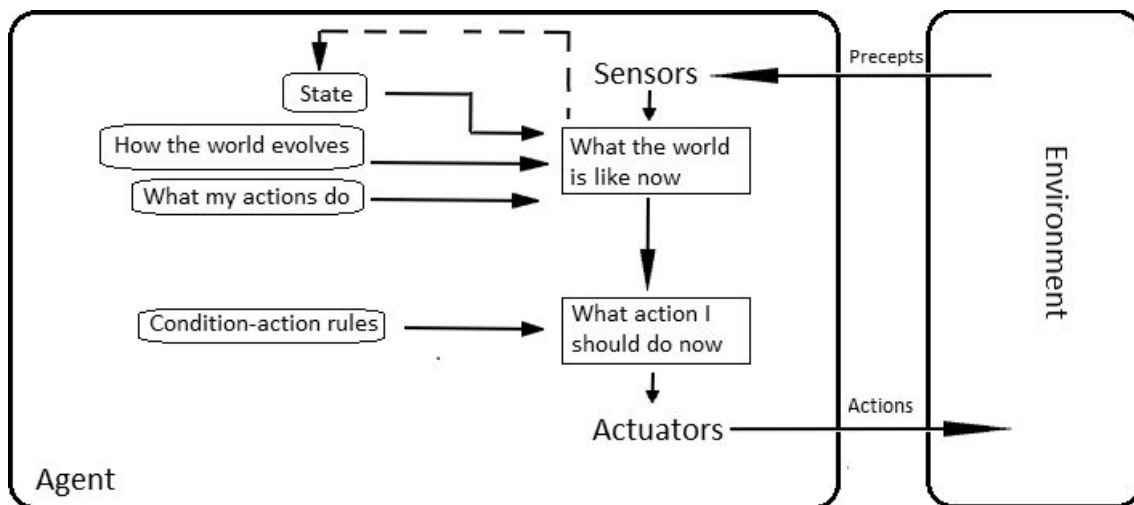
Este tipo de agentes são bastante simplistas, mas possuem inteligência limitada, sendo impossível a implementação da totalidade das regras para todas as situações possíveis de encontrar num ambiente. Apenas conseguem ter sucesso em ambientes totalmente observáveis. Alguns possuem informação relativamente ao seu estado actual, o que permite desprezar algumas condições previamente accionadas.

Em ambientes parcialmente observáveis entram em ciclos infinitos que só podem ser interrompidos através de acções geradas de forma aleatória.



Alguns agentes reactivos conseguem operar em ambientes parcialmente observáveis. O modelo do mundo (ambiente) é mantido internamente, o que descreve as partes não observáveis.

Este agente deve de manter algum tipo de modelo interno que depende do histórico de percepções e, assim, reflecte alguns aspectos não observáveis do estado actual. O histórico de percepções e o impacto de determinadas acções no ambiente podem ser determinados através da utilização do modelo interno.



AGENTES DELIBERATIVOS

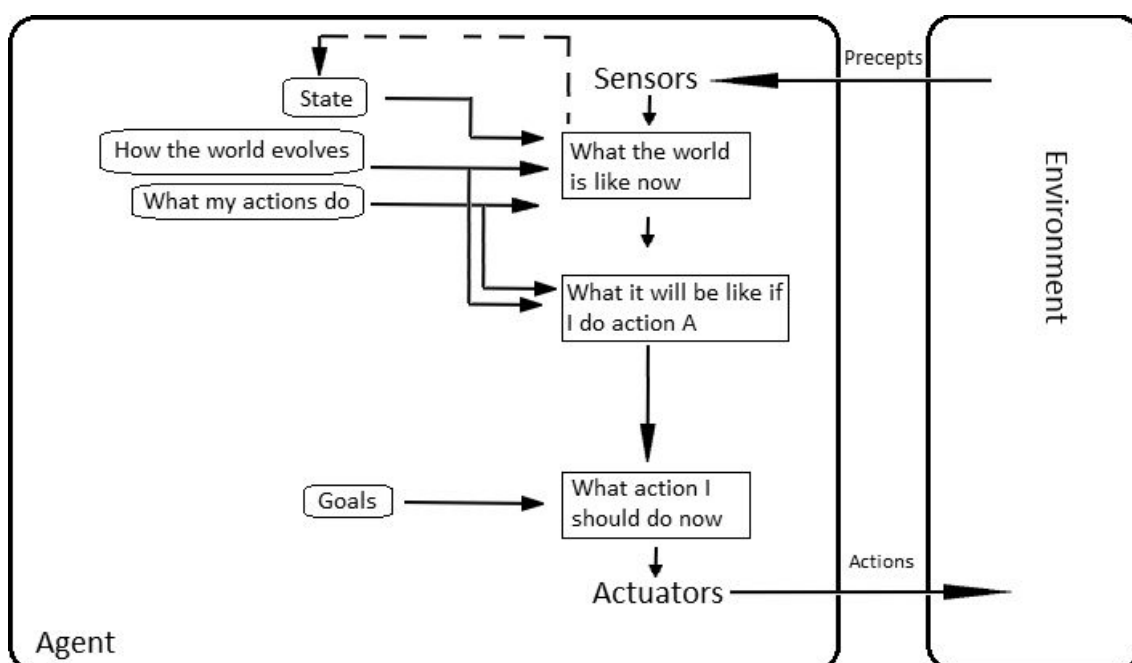
Agentes deliberativos expandem as capacidades dos agentes reactivos ao utilizarem informação prestada pelo objectivo a atingir e, também, ao descrever situações que são desejáveis. Isto permite ao agente decidir entre múltiplas possibilidades, seleccionando a acção que lhe permite atingir o objectivo.

O raciocínio prático é o processo de decidir o que fazer através de acções. Permite ao agente explorar as várias opções possíveis num determinado estado. Este processo é dividido em raciocínio sobre fins e meios:

- Fins: Onde se decide o que fazer - **deliberar**. Resulta em objectivos a atingir.
- Meios: Onde se decide como fazer - **planear**. Resulta num plano de acções.

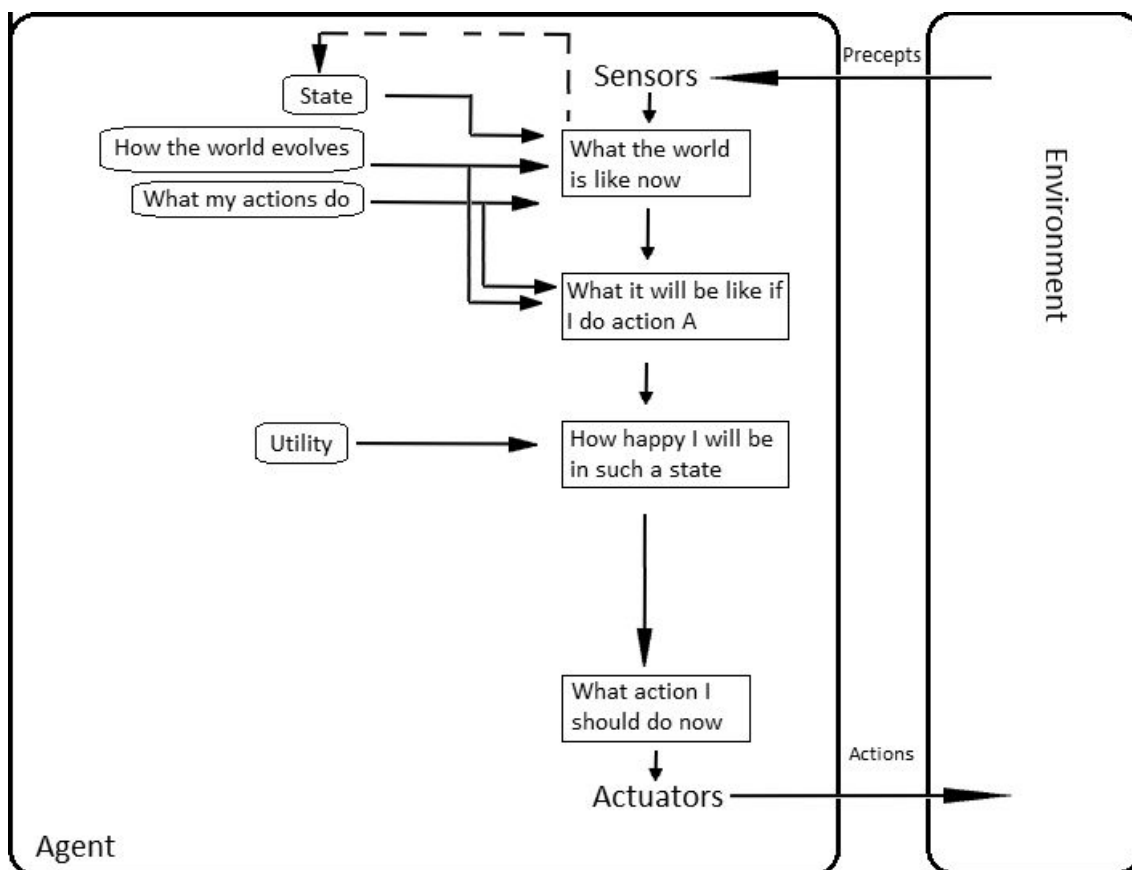
O processo geral de tomada de decisão é composto por:

1. **Observar o mundo:** Receber uma percepção e actualizar o modelo interno do mundo.
2. **Reconsiderar:** Condição em que verifica se o modelo interno do mundo foi actualizado através da observação do mundo ou se existe um plano de acções pendente.
 - 2.1. **Deliberar:** Processo de identificação de objectivos através do modelo interno do mundo.
 - 2.2. **Planear:** Processo que permite ao agente determinar a sequências de acções de forma a atingir o objectivo.
3. **Executar:** Seleccionar a acção a ser executada.

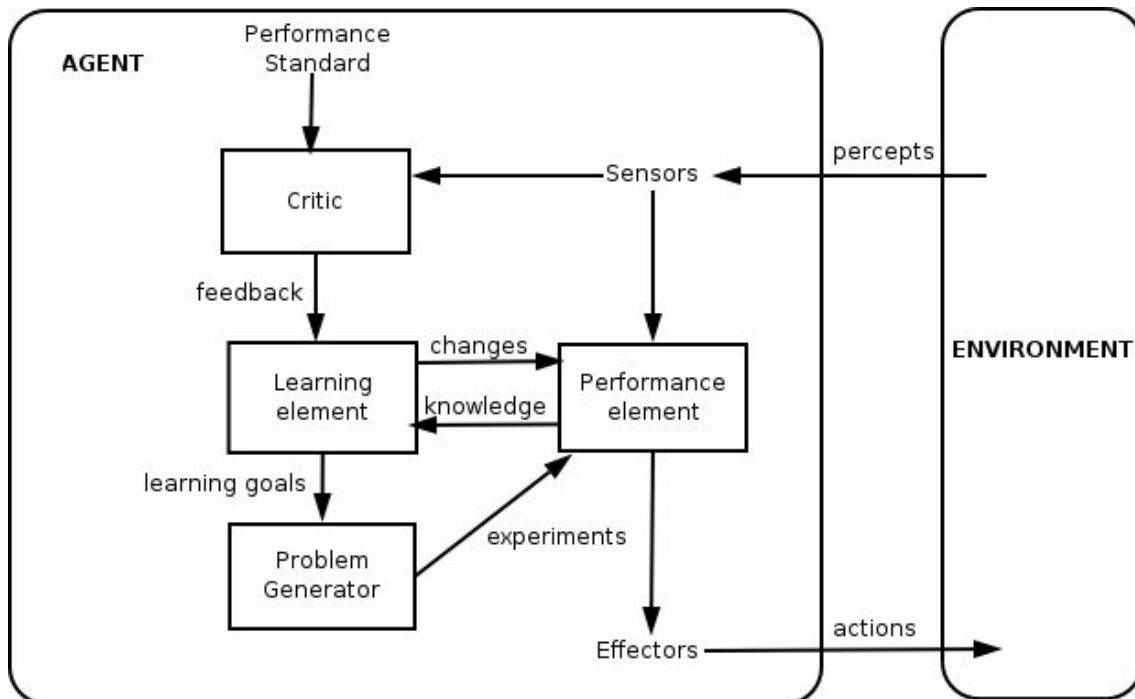


Decidir se um estado é ou não o objectivo a atingir não é suficiente. Torna-se necessário medir o quanto desejável é um estado em particular. Esta métrica pode ser obtida através do uso de uma função de utilidade que calcula a utilidade de um estado – e que deve também permitir o cálculo entre diferentes estados do mundo. A função utilidade pode ser descrita como a quantificação da “felicidade” que o agente possui.

Um agente deliberativo selecciona a acção que maximiza a utilidade expectável da mesma, isto é, aquilo que o agente espera obter, em média, dadas as probabilidades e utilidades de cada acção resultante. Este tipo de agentes precisa de modelar e manter um estado interno do mundo e das tarefas que envolvem uma grande quantidade de pesquisa e processamento de percepções, representações, raciocínio e aprendizagem.



[TODO]



ESTRATÉGIAS DE PESQUISA

Numa arquitectura deliberativa o processo de planeamento permite ao agente determinar as sequências de acções de forma a atingir o objectivo.

Para o atingir é necessário executar uma sequência de acções, tal como, a solução gerada por uma estratégia de pesquisa. A solução é uma sequência de acções e os algoritmos de pesquisa analisam várias sequências de acções possíveis. As sequências possíveis iniciam-se no nó-raiz que contém o estado inicial. Assim, formam uma árvore de pesquisa com ramos que correspondem às acções e os nós expandidos por essas acções com os seus respectivos estados.

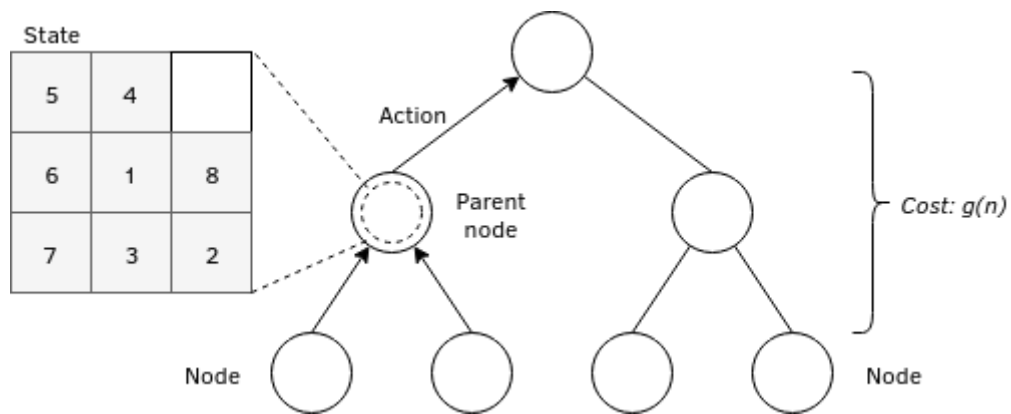
Após ser encontrada uma solução através da estratégia de pesquisa torna-se necessário elaborar um plano de acções, onde a sequência de acções corresponde à sequência de nós da solução.

De maneira a analisar várias acções expande-se o nó actual aplicando as possíveis acções, e assim, obtém-se um novo conjunto de nós que são adicionados a uma estrutura de dados. A esta estrutura de dados chama-se de fronteira, que guarda em memória a ordem pela qual os nós devem de ser explorados. Este processo repete-se até que o nó-solução seja encontrado ou até que não exista mais nenhum nó para ser expandido.

A diferença entre os algoritmos das estratégias de pesquisa é a maneira como se escolhe o próximo nó a ser expandido.

De maneira a evitar caminhos redundantes é utilizada uma outra estrutura de dados que guarda em memória todos os estados e respectivos nós expandidos. Assim, todos os novos nós expandidos que coincidam com outros previamente explorados, ou seja, contidos na memória de explorados, podem ser descartados.

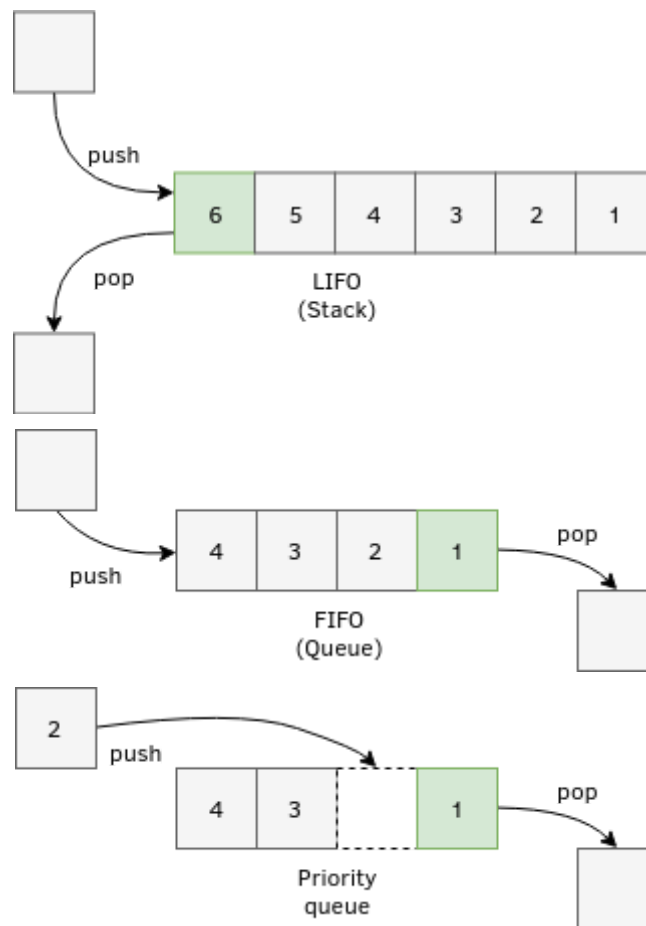
- **Estado:** estado que corresponde ao nó;
- **Pai:** o nó na árvore de pesquisa que gerou o nó actual;
- **Acção:** a acção que foi aplicada ao nó-pai que gerou o nó actual;
- **Custo de caminho:** o custo, denominado de $g(n)$, do nó inicial ao nó actual.



ESTRUTURAS DE DADOS

As estruturas de dados dividem-se em três variantes comuns:

- **FIFO:** *First In First Out*, remove o elemento mais antigo do contentor - *queue*;
- **LIFO:** *Last In First Out*, remove o elemento mais recente do contentor - *stack*;
- **Priority queue:** remove o elemento com maior prioridade segundo uma determinada ordem.



ANÁLISE DA ESTRATÉGIA DE PESQUISA

Ao existirem múltiplas estratégias de pesquisa é necessário atribuir critérios matemáticos para que estas possam ser analisadas e comparadas:

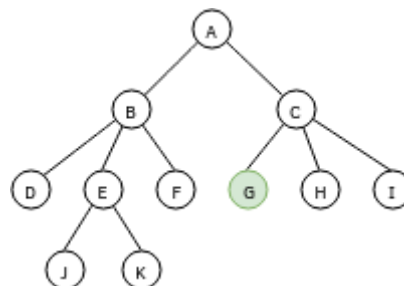
- **Completa:** garantia de ser encontrado uma solução;
- **Ótima:** garantia de encontrar a solução ótima;
- **Complexidade espacial:** número máximo de nós armazenados na memória de fronteira, ou seja, quanta memória é necessária para executar o algoritmo;
- **Complexidade temporal:** número de nós explorados durante a pesquisa, ou seja, quanto tempo é necessário para realizar a pesquisa.

ESTRATÉGIAS DE PESQUISAS NÃO-INFORMADAS

As estratégias de pesquisas não-informadas, também chamadas de pesquisas cegas, não usam qualquer informação sobre os estados para além do que é fornecido pelo problema. As suas únicas funções são expandir e explorar nós sucessores e distinguir o estado-solução de outros estados.

BREADTH-FIRST SEARCH

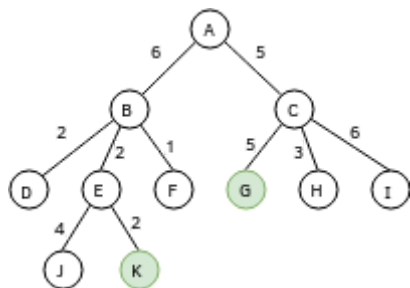
Breadth-first, ou “pesquisa em largura” em português, expande inicialmente o nó-raiz e, posteriormente, todos os seus sucessores consecutivamente após cada nível da árvore de procura. Esta pesquisa usa uma memória FIFO (queue) de maneira a explorar primeiro os nós mais superficiais.



No explorado	Fronteira	Explorados
	A	
A	B, C	
B	C, D, E, F	A
C	D, E, F, G, H, I	A, B
D	E, F, G, H, I	A, B, C
E	F, G, H, I, J, K	A, D, C, D
F	G, H, I, J, K	A, D, C, D, E
G	H, I, J, K	A, D, C, D, E, F

UNIFORM-COST SEARCH

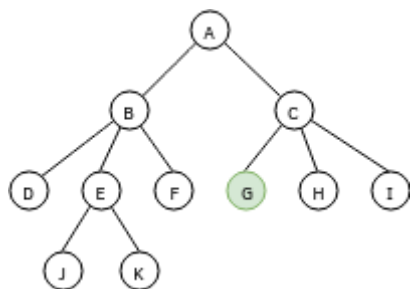
A pesquisa uniforme é uma adaptação da pesquisa em largura, sendo que a única diferença consiste na exploração do nó que tem menor custo. Isto acontece através da utilização de uma memória *priority queue* como fronteira.



No explorado	Fronteira	Explorados
	A	
A	C5, B6	
C	B6, H8, G10, I11	A
B	F7, H8, D8, E8, G10, I11	A, C
	H8, D8, E8, G10, I11	
F	D8, E8, G10, I11	A, C, B
H	E8, G10, I11	A, C, B, F
D	G10, K10, I11, J14	A, C, B, F, H
E	K10, I11, J14	A, C, B, F, H, D
G		A, C, B, F, H, D, E

DEPTH-FIRST SEARCH

Depth-first search, ou “pesquisa em profundidade” em português, expande o nó de maior profundidade da árvore de procura. Nesse sentido, esta pesquisa utiliza uma memória LIFO (stack) como fronteira.

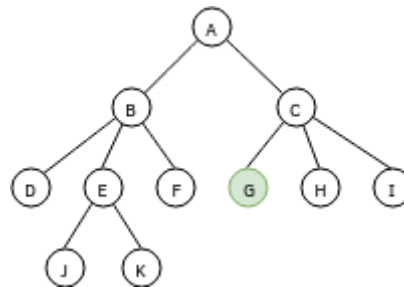


No explorado	Fronteira	Explorados
	A	
A	B, C	
B	D, E, F, C	A
D	E, F, C	A, B
E	J, K, F, C	A, B, D
J	K, F, C	A, B, D, E
K	F, C	A, B, D, E, J
F	C	A, B, D, E, J, K

C	G, H, I	A, B, D, E, J, K, F
G	H, I	A, B, D, E, J, K, F, C

ITERATIVE DEPTH-FIRST SEARCH

A pesquisa iterativa executa a pesquisa em profundidade iterando sobre o nível de profundidade. Começa pelo nível 0, depois o 1, 2 e assim sucessivamente até ser encontrado o nó-solução.



No explorado	Fronteira	Explorados
	A	
A		A

	A	
A	B, C	
B	C	A
C		A, B
		A, B, C

	A	
A	B, C	
B	D, E, F, C	A
D	E, F, C	A, B
E	F, C	A, B, D
F	C	A, B, D, E
C	G, H, I	A, B, D, E, F
G	H, I	A, B, D, E, F, C

COMPARAÇÃO DAS ESTRATÉGIAS DE PESQUISAS NÃO-INFORMADAS

Estratégia de pesquisa	Completa	Optima	Complexidade espacial	Complexidade temporal
BFS	Sim ¹	Sim ³	$O(b^d)$	$O(b^d)$
UCS	Sim ^{1,2}	Sim	$O(b^{1+C*/\epsilon})$	$O(b^{1+C*/\epsilon})$
DFS	Não	Não	$O(b*m)$	$O(b^m)$
IDS	Sim ¹	Sim ³	$O(b*d)$	$O(b^d)$

¹ Completa se b for finito

² Completa se custo de passo $\geq \epsilon$

³ Ótimo se os custos de passos forem idênticos

b Número máximo de nós sucessores de qualquer nó - *branching factor*

- d Profundidade do nó que contém a solução - *depth*
- C* Custo da solução ótima
- ϵ Custo mínimo de uma transição de estado ($\epsilon > 0$)
- m Profundidade da árvore de procura

Ao comparar as quatro estratégias de pesquisa não-informadas conseguimos concluir:

- BFS: tem complexidade espacial exponencial - o que em termos computacionais é uma grande desvantagem, mas, por outro lado, garante que se encontra uma solução ótima (dada as condições);
- DFS: apesar de ter uma complexidade espacial menor do que a BFS, tem a desvantagem de não ser completa nem ótima, correndo assim o risco de não encontrar nenhuma solução;
- UCS: tem a capacidade de encontrar a solução ótima (dadas as condições), no entanto a complexidade espacial e temporal pode ser bastante mais elevada do que as restantes pesquisas;
- IDS: combina os benefícios da DFS e BFS em que a sua complexidade espacial é modesta tendo a capacidade de encontrar a solução ótima (dadas as condições).

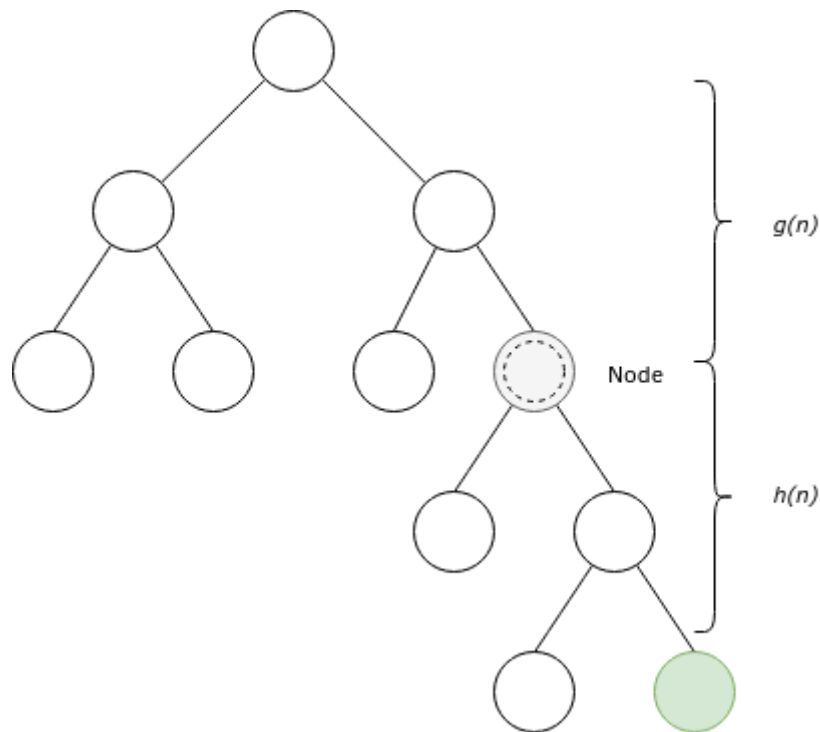
ESTRATÉGIAS DE PESQUISAS INFORMADAS

As estratégias de pesquisas informadas utilizam informação específica do problema que pode conter factores internos ou externos do problema. Assim, melhoram a sua eficiência em comparação às estratégias de pesquisa não-informadas.

A selecção do nó para expansão é realizada através de uma função de avaliação $f(n)$. Esta avaliação é construída através de uma estimativa, assim, o nó com menor valor de $f(n)$ será expandido primeiro. A sua implementação é idêntica à estratégia UCS com excepção de se usar $f(n)$ em vez de $g(n)$ para ordenar a memória *priority queue* que serve de fronteira.

A maioria dos algoritmos de estratégias de pesquisas informadas incluem como uma componente de $f(n)$ uma função heurística, denominada de $h(n)$ - sendo $h(n)$ o menor valor estimado do caminho entre o estado do nó actual até ao estado do nó-solução.

Uma heurística deve de ser admissível, nunca deve de superestimar o custo de atingir o nó-solução, isto é, ser optimista. Pode-se considerar admissível se valor proveniente de $h(n)$ for menor ou igual do que o valor real necessário para atingir o objectivo. No caso de um problema de distâncias entre vários pontos o valor de $h(n)$ admissível seria a distância euclidiana entre dois pontos.



GREEDY BEST-FIRST SEARCH

Este algoritmo expande o nó que esteja mais próximo do objectivo através da avaliação da função heurística $f(n) = h(n)$. A cada passo tenta aproximar-se o mais possível do nó-solução.

A* SEARCH

Este é o algoritmo mais conhecido deste tipo de pesquisas. Os nós são avaliados através do mínimo valor da combinação de $g(n)$ e $h(n)$:

$$f(n)=g(n)+h(n)$$

COMPARAÇÃO DAS ESTRATÉGIAS DE PESQUISAS INFORMADAS

Estratégia de pesquisa	Completa	Optima	Complexidade espacial	Complexidade temporal
Greedy BFS	Não	Não	$O(b^m)$	$O(b^m)$
A*	Sim	Sim	$O(b^d)$	$O(b^d)$

O desempenho de estratégias de pesquisa informadas depende da qualidade da heurística utilizada. Uma boa heurística pode ser definida removendo certas definições ou obstáculos do problema, guardando custos pré-calculados de sub-problemas, ou aprendendo com a experiência do problema em questão.

PROCESSOS DE DECISÃO DE MARKOV

O processo de decisão de Markov consiste uma estrutura que ajuda a tomar decisões em ambientes estocásticos. O objectivo passa por construir um mapa interno com a acção óptima de cada estado do ambiente - política.

Este processo é mais optimizado em relação à elaboração de um plano, isto porque a política permite ao agente realizar as acções óptimas, mesmo que algo mude enquanto executa.

A propriedade de Markov refere-se ao facto de a probabilidade de atingir o próximo estado é influenciado pela acção a escolhida.

Especificamente, é dado pelo modelo de transição $T(s, a, s')$. Então, o próximo estado s' depende do estado actual s e da acção escolhida, não havendo qualquer dependência dos estados e acções anteriores.

$$U^{\pi^*} = \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \cdot U^{\pi}(s')]$$

APRENDIZAGEM

- Conceito de aprendizagem e aprendizagem por reforço

Q-LEARNING

- Algoritmo Q-Learning

CONCLUSÃO