

Busca Informada e além da busca convencional/clássica

Observações:

- Pontos: 3 pontos
- Trabalho em Grupo
- Somente um dos integrantes envia o trabalho via SIGAA (O primeiro da lista do grupo)
- Observar a Data e horário de entrega, Tarefas via e-mail não serão consideradas.
- Qualquer copia de Tarefa não terá direito de avaliação, sendo colocada a nota de zero(0) a todos os envolvidos na copia.
- O sistema SIGAA recebe somente um arquivo, de essa forma precisa criar um único arquivo em formato **“.zip”** contendo todo o material solicitado para avaliação.
- O que deve ser entregue: Os códigos fonte das novas classes (estendidas ou implementadas) e o relatório explicando todas suas observações e descobertas.

Considerações Iniciais

Um outro tópico do Capítulo 3 do livro de Russell e Norvig, da 3ra Edição, trabalha/estuda os algoritmos de busca informada(com heurísticas), e no capítulo 4 deste mesmo livro trata sobre os assuntos de Estratégias de busca que estão além das convencionais ou além das estratégias clássicas. Aqui vamos nos concentrar em algumas dessas estratégias, algumas que trabalham com heurísticas e outras estratégias além das estratégias convencionais:

Estratégia de Busca	Código Fonte no aima-java aima.core.search.uninformed.
Best-first Search	aima.core.search.informed.BestFirstSearch.java
3.5.1 Greedy best-first search	aima.core.search.informed.GreedyBestFirstSearch.java
3.5.2 A* Search	aima.core.search.informed.AstarSearch.java
3.5.3 RBFS – Recursive best-first search	aima.core.search.informed.RecursiveBestFirstSearch.java
4.1.1 Hill-climbing search	aima.core.search.local.HillClimbingSearch.java
4.1.2 Simulated annealing	aima.core.search.local.SimulatedAnnealingSearch.java
4.1.4 Genetic Algorithms	aima.core.search.local.GeneticAlgorithm.java

Vamos manter os mesmo ambientes:

Ainda temos na mesma literatura alguns contextos ou ambientes onde podem ser aplicadas estas estratégias. Como por exemplo:

Problema/contexto	Código fonte no aima0java
Problema do Caixero viajante	aima.gui.demo.agent.MapAgentDemo.java
Problema do quebra cabeça de oito peças	aima.gui.demo.search.EigthPuzzleDemo.java
Problema das oito rainhas	aima.gui.demo.search.NqueensDemo.java

Considerando sempre uma abordagem de agentes que percebem o ambiente e agem no ambiente após uma sequencia de percepções se pede o seguinte:

O que deve ser feito:

- Crie uma pasta com o nome do Grupo (por exemplo 'Grupo01') e dentro dela coloque as implementações para as atividades solicitadas abaixo.
- Utilizar cada uma das estratégias informadas/heurísticas e estratégias além da busca clássica (primeira tabela acima) com cada um dos problemas ou contextos aqui citados (segunda tabela acima)
- O objetivo é :
 - Pegar o melhor algoritmo de busca cega (Guia 02) e trazer para este contexto com o intuito de comparar com as estratégias de esta Guia 03. Explicar os motivos da escolha do algoritmos de busca cega.
 - Na sequencia comparar todos os algoritmos (considere o melhor Algoritmo de busca cega + algoritmos com heurística + algoritmos além da busca classica) em cada problema. Para isso você deve realizar os experimentos e tomar nota dos resultados em uma tabela comparativa para cada problema.

Exemplo:

Problema xxxxx

Configuração do ambiente: <descrever a configuração do ambiente>

Estratégias de busca		Variável 1	Variável 2	Variável 3	... adicionar outras variáveis ...
Estratégia (busca cega)	1				
Estratégia (outros algoritmos)	2				
...					

- Descrever vantagens e desvantagens observadas em cada algoritmo e o motivo.
- Nenhum dos exercícios (para a avaliação) exige interface/tela gráfica para mostrar o resultado, tudo pode ser mostrado na consola ou reutilizado no relatório. Mas se preferir acrescentar interface gráfica não terá nenhum problema.
- Entender como a abordagem de POO e Inteligência Artificial está inserida na Programação/Codificação e sua solução quando trabalham com os diversos algoritmos de busca considerando a arquitetura de Agentes.
 - Descrever suas descobertas no relatório.