Subject: AI - HW01
Date 1402/12/18

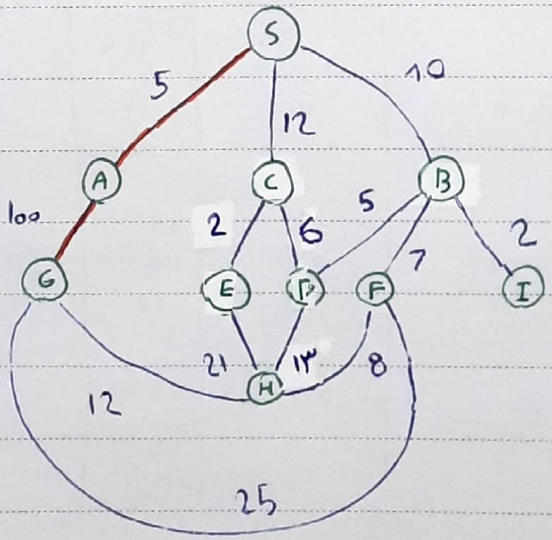
810100175
Mohammad Sadeghi

① Agents

Environment	CS - GO	Sudoku
Fully / Partially ob	Partially_observable	fully observable
Deterministic / Stochastic	Stochastic	Deterministic
Episodic / sequential	Episodic	Sequential
Static / Dynamic / semi-dy	Semi-Dynamic	Static
Discrete / continuous	Continiuous	Discrete
singl / multi-agent	multi-agent	single_agent

② Search

1- الف) با اجرای الگوریتم های dfs و bfs به جواب زیر می‌رسیم :



$S \to A \to G$ | $|V| = 100$ ⇒ این جواب بهینه نیست زیرا مسیر زیر :

$S \xrightarrow{12} C \xrightarrow{2} E \xrightarrow{21} H \xrightarrow{12} G$ با طول ۴۷ وجود دارد !

Subject :
Date

* استفاده ازین 2 الگوریتم زمانی مناسب است که وزن یالها برابر باشند.

1-ب) جدول UCS :

state	path	cost	Ex	Front
S	S	۰	۰	S
S	S	۰	S	A , B , C
A	S A	۵	S , A	B , C , G
B	S B	۱۰	S , A , B	C , G , D , F , I ۱۲ ۱۰۵ ۱۵ ۱۷ ۱۲
C	S C	۱۲	S , A , B , C	G۱۰۵ D۱۵ F۱۷ I۱۲ E۱۴
I	S B I	۱۲	S , A , B , C , I	G۱۰۵ D۱۵ F۱۷ E۱۴
E	S C E	۱۴	S , A , B , C , I , E	G۱۰۵ D۱۵ F۱۷ H۲۰۵
D	S B D	۱۵	SABC IED	G۱۰۵ F۱۷ H۲۰۸
F	S B F	۱۷	SABC IEDF	G۴۲ H۲۰۵
H	S B F H	۲۵	SABC IEDF H	G ۳۷
G	S B F H G	۳۷	SABC IEDF HG	-

$$\min(G) = 37 \text{ , } S \xrightarrow{10} B \xrightarrow{5} F \xrightarrow{8} H \xrightarrow{12} G$$

Subject:
Date

2 - (الف)

Frontier	Explored	Cost+h	Cost	Path	State
$A_{۴۵}$, $B_{۳۵}$, $C_{۴۲}$	S	۳۶	۰	S	S
$A_{۴۵}$, $C_{۴۲}$, $D_{۳۰}$, $F_{۳۶}$, $I_{۳۵}$	S, B	۳۵	۱۰	S B	B
$A_{۴۵}$, $C_{۴۲}$, $F_{۳۶}$, $I_{۳۵}$, $H_{۳۸}$	S, B, D	۳۰	۱۵	S B D	D
$A_{۴۵}$, $C_{۴۲}$, $F_{۳۶}$, $H_{۳۸}$	S, B, D, I	۳۵	۱۲	S B I	I
$A_{۴۵}$, $C_{۴۲}$, $H_{۳۵}$, $G_{۴۲}$	S, B, D, I, F	۳۶	۱۷	S B F	F
$A_{۴۵}$, $C_{۴۲}$, $G_{۴۷}$	S, B, D, I, F, H	۳۵	۲۵	S B F H	H
$A_{۴۵}$, $C_{۴۲}$	S, B, D, I, F, H, G	۳۷	۳۷	S B F H G	G
$A_{۴۵}$, $E_{۴۲}$	S B D I F H G C	۴۲	۱۲	S C	C
$A_{۴۵}$	S B D I F H G C E	۴۲	۱۴	S C E	E
–	S B D I F H G C E A	۴۵	۵	S A	A

2-ب) در تمام سطر ها Cost+h ≥ Cost-Real است که نشان می دهد هیوریستیک ما admissable است. اما Consistent نیست چون:

$h(S) - h(B) > cost(S \to B)$ ⇒ ۳۶ − ۲۵ = ۱۱ > ۱۰

PAPCO

Subject:
Date

3 الف) الگوریتم local search را زمانی ترجیح می دهیم که به دنبال بیشینه کردن تابع هدف یا معیاری هستیم که Performance ربات ما را افزایش می دهد. همچنین در مسائلی که فضای جستجوی agent بسیار وسیع و پرازداده باشد، استفاده از global search، جستجوی در تمام راه حل های ممکن بسیار پرهزینه است و اگر با محدودیت منابع و زمان مواجه باشیم (که در اکثر مواقع نیز در واقعیت اینگونه است) عملا برای ما به صرفه نیست. در این مواقع می توان از الگوریتم های جستجوی محلی بهره برد که در هر لحظه راه حلی را ادامه می دهد که برای او معیار بالاتری داشته باشد.

3-ب) برای حل مشکل ماکزیمم محلی دو روش وجود دارد:

1) اینکه ما اجازه دهیم ربات از چند initial state مختلف شروع به حرکت کند و از راه حل هایی که یافت، بهترین را در نظر بگیریم (البته این روش باز ممکن است بهترین جواب را ندهد)

2) روش Simulated Annealing که به ربات اجازه می دهد در هر state با یک احتمال بتواند حرکت غیر بهینه که این احتمال با افزایش Performance ربات به صفر میل می کند! انجام دهد