

## Agents

Environment	CS-GO	Sudoku
Fully/Partially observable	Partially observable	Fully observable
Deterministic/Stochastic	Stochastic (strategic)	Deterministic
Episodic/Sequential	Sequential	Sequential
Static/Dynamic/Semi-Dynamic	Dynamic	Static
Discrete/Continuous	Continuous	Discrete
Single-agent/Multi-agent	Multi-agent	Single-agent

## Search

سوال اول

الف) خیر. استفاده از BFS در صورتی توصیه می شود و جواب بهینه را به ما می دهد که وزن تمام الیاد گراف با یکدیگر برابر باشد.

استفاده از DFS در صورت جواب optimal را به ما می دهد اما با گاهی ما متوجه می شویم که آن جواب را به ما می دهد.

Iterative deepening search (Depth-limited Search) و الگوریتم می توان از آن استفاده کرد و این که می توان جواب

Subject:

Date:

optimal در پیدا کردن loop infinite نیوفتاد. همچنین اگر حاصل های زیادی برای یک مسئله باشد

(راه های زیاد) DFS می تواند بهتر از BFS نتیجه بدهد!

این قسمت را با فرض graph search بودن حل کردیم یعنی state تکراری را وارد frontier نکردیم

current state	Frontier	Explored	Path	Total cost
S	A5, B10, C12	S	S	0
A	G105, B10, C12	SA	SA	5
B	G105, D15, F17, I12, C12	SAB	SB	10
C	G105, A5, F17, I12, F14	SABC	SC	12
I	G105, A5, F17, F14	SABCI	SBI	12
E	G105, D15, F17, H35	SABCE	SCE	14
D	G105, F17, H35	SABCIED	SBD	15
F	G42, H25	SABCIEDF	SBF	17
H	G37	SABCIEDFH	SBFH	25
G	—	SABCIEDFHG	SBFHG	37

پایان در B و D  
بهترین حالت  
در frontier

↓  
مستقیم↓  
بهترین حالت



Date \_\_\_\_\_

(الف)

**PAPCO.**

بیا

Admissibility:  $h$  heuristic هر نود از مقدار واقعی آن کوچکتر مساوی است.

$S: 36 \leq 42$  ;  $A: 40 \leq 100$  ;  $C: 30 \leq 31$  ;  $B: 25 \leq 32$  ;  $I: 23 \leq 34$  ;  $E: 28 \leq 33$

$D: 15 \leq 25$  ;  $H: 10 \leq 12$  ;  $F: 19 \leq 25$

مجموعه node ها پس از بررسی Admissible است.

Consistency:  $\text{cost}(A \rightarrow C) \geq h(A) - h(C)$  ; for each node

مثال:  $S \rightarrow B \rightarrow 10 < 11 \rightarrow$  پس consistent نیست!

## سوال سوم

(الف) در سه تا از چهار رابطه  $\text{local search}$ ، به علاوه روش های جستجو بررسی دارند:

① محدودیت حافظه داشتن (زیر الگوریتم های  $\text{local search}$  مقدار حافظه بسیار کمی استفاده می کنند).

② در فضای دارای بینهایت  $\text{state}$  (یعنی دارای  $\text{state}$  های بسیار زیاد و یا فضای های بی پایان) می توانست جواب های منطقی ای

پیدا کنند که الگوریتم های سرچ سیستماتیک قادر نیستند.

③ در فضای  $\text{partially observable}$  سرچ تکریم هدف نزدیک می شوند.



ب)

① تنظیم کردن الگوریتم بدگونهای که به agent مقداری اجازه اشتباه کردن و در مسیر اشتباه پرسش رفتن هم بدهیم؛

به امید اینکه در state ای قرار بگیرد که به max value بازنگر دود به سمت global max حرکت کند. برای این

کار از الگوریتم Simulated Annealing استفاده می‌کنیم؛ به این صورت که با تفریق کردن یک تابع دما و یک احتمال،

احتمال حرکت اشتباه را به agent می‌دهیم. در ابتدا دما زیاد و در نتیجه این احتمال زیاد است و هر چه پرسش می‌رویم،

دما کاهش داده و احتمال خطا را کم می‌کنیم.

② افزایش دادن مقدار جابجایی در موقعیت‌هایی که هیچ یک از عملیات‌ها وضعیت بهتری از state فعلی

ماند و شرط این افزایش دادن متناسب با الگوریتم.