



## هوش مصنوعی

پاییز ۱۴۰۱

استاد: محمدحسین رهبان

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

گردآورندگان: محمدطاها جهانی نژاد، امیررضا میرزایی، محمدرضا مفیضی

مهلت ارسال: ۱ آذر

## Adversarial Search and CSP

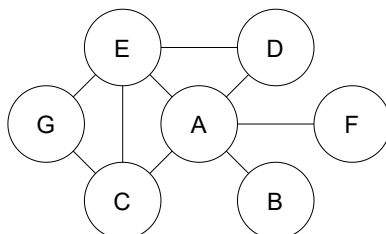
تمرین دوم

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص شده است.
- در طول ترم امکان ارسال با تاخیر پاسخ همه‌ی تمارین تا سقف ۱۰ روز و در مجموع ۲۰ روز، وجود دارد. پس از گذشت این مدت، پاسخ‌های ارسال شده پذیرفته نخواهند بود. همچنین، به ازای هر روز تأخیر غیر مجاز ۱۰ درصد از نمره تمرین به صورت ساعتی کسر خواهد شد.
- همکاری و هم‌فکری شما در انجام تمرین مانعی ندارد اما پاسخ‌های ارسال شده هر کس حتماً باید توسط خود او نوشته شده باشد.
- در صورت هم‌فکری و یا استفاده از هر منابع خارج درسی، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفاً تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.

### سوالات (۱۰۰ نمره)

۱. (۲۰ نمره) به سوال‌های زیر پاسخ کوتاه بدهید.

(آ) در CSP زیر اگر قبل از هر مقداردهی arc consistency را اجرا کنیم با ترتیب مقداردهی A, B, C, D, E, F, G در کدام متغیرها باید backtrack کنیم؟



(ب) یک درخت بازی متخاصم را در نظر بگیرید که در آن ریشه یک گره ماکسیمم است و مقدار minimax بازی  $x$  است. اکنون، یک درخت مشابه را نیز در نظر بگیرید که در آن هر گره مینیمم با یک گره شانس (با توزیع احتمال دلخواه اما شناخته شده) جایگزین می‌شود. مقدار expectimax درخت بازی تغییر یافته  $y$  است. نشان دهید  $x$  کمتر یا مساوی  $y$  است.

حل.

(آ) متغیرهای A و B و C. هر راس را می‌توان تا زمانی که یک cutset مقداردهی شود، backtrack کرد. توجه داشته باشید که مقادیر B در قسمت اول هیچ تأثیری بر بقیه CSP پس از مقداردهی A ندارد. با این حال، به دلیل روشی که جستجوی backtracking اجرا می‌شود، اگر راه‌حل ثابتی برای مقدار معین A وجود نداشته باشد، B همچنان قبل از A مقدار داده می‌شود.

(ب) اگر گره مینیمم بهینه عمل کند، تضمین می‌شود که گره ماکسیمم می‌تواند به  $x$  دست یابد و اگر گره مینیمم به‌طور غیربهینه عمل کند (مثلاً با عمل تصادفی) گره ماکسیمم می‌تواند بهتر عمل کند.

۲. (۲۰ نمره) در یک هتل مشغول به کار هستید و قرار است ۷ کارمند را برای رسیدگی به طبقات مختلف ساختمان بفرستید. ساختمان هتل ۶ طبقه دارد و هر کارمند باید به یک طبقه برود (اما ممکن است چند کارمند با هم به یک طبقه بروند یا هیچ کارمندی به یک طبقه نرود). متأسفانه کارمندان شرایط خاص خود را برای رفتن به طبقات هتل دارند:

- کارمند شماره ۲ نمی‌خواهد با کارمند شماره ۴ به یک طبقه برود.
- کارمند شماره ۵ فقط می‌تواند به سه طبقه اول برود.
- کارمند شماره ۶ دوست دارد به طبقات شماره فرد برود.
- کارمند شماره ۷ می‌خواهد به طبقه‌ای پایین‌تر از طبقه کارمند شماره ۶ برود.
- کارمند شماره ۵ هم می‌خواهد به طبقه‌ای پایین‌تر از طبقه کارمند شماره ۲ برود.
- کارمند شماره ۱ فقط می‌تواند به طبقه ۵ برود.
- کارمند شماره ۴ می‌خواهد به طبقه‌ای بالاتر از طبقه کارمند شماره ۷ برود.
- اگر کارمند شماره ۱ قرار باشد با یک نفر به یک طبقه برود آن فرد کارمند شماره ۷ نخواهد بود.
- کارمند شماره ۷ نمی‌تواند به طبقه ۶ برود.
- کارمند شماره ۳ نمی‌تواند به طبقات ۴ یا بالاتر برود.
- کارمند شماره ۲ نمی‌تواند به طبقه ۵ برود.
- کارمند شماره ۲ می‌خواهد به طبقه‌ای پایین‌تر از طبقه کارمند شماره ۳ برود.

(آ) مساله گفته شده را با یک CSP مدل می‌کنیم که در آن متغیرها کارمندان شماره ۱ تا ۷ و دامنه مقادیر طبقات ۱ تا ۶ است. دامنه هر کدام از متغیرها را بعد از اعمال محدودیت‌های unary تعیین کنید.

(ب) هیوریستیک MRV را اجرا کنید و بگویید کدام متغیر باید قبل از بقیه مقداردهی شود.

(ج) می‌خواهیم ابتدا کارمند شماره ۶ را مقداردهی کنیم. با اجرای هیوریستیک LCV بگویید کدام ترتیب مقداردهی برای کارمند شماره ۶ بهتر است.

(د) گراف محدودیت این CSP را رسم کنید و یک جواب قابل قبول برای آن (در صورت وجود) بیابید.

حل.

(آ) به صورت زیر است:

کارمند	طبقه					
۱	۵					
۲	۱	۲	۳	۴	۶	
۳	۱	۲	۳			
۴	۱	۲	۳	۴	۵	۶
۵	۱	۲	۳			
۶	۱		۳	۵		
۷	۱	۲	۳	۴	۵	

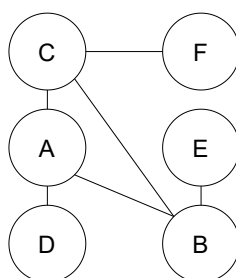
(ب) کارمند شماره ۱، زیرا کمترین مقادیر را در دامنه خود دارد.

(ج) کارمند شماره ۶ با ترتیب ۵، ۳، ۱ مقداردهی می‌کنیم. زیرا با این ترتیب محدودیت روی مقادیر متغیرهای تحت تاثیر کارمند شماره ۶ افزایش می‌یابد. تنها محدودیتی که توسط کارمند ۶ ایجاد می‌شود "کارمند شماره ۷ می‌خواهد به طبقه‌ای پایین‌تر از طبقه کارمند شماره ۶ برود." است. یعنی با اجرای forward checking و قرار دادن ۵، ۳ و ۱ برای کارمند ۶، دامنه کارمند ۷ به ترتیب {۱، ۲، ۳، ۴} و {۱، ۲} و {} خواهد بود.

(د) گراف محدودیت این CSP ساختاری درختی خواهد داشت. یک جواب قابل قبول برای مساله به صورت زیر است:

کارمند	طبقه
۱	۵
۲	۲
۳	۳
۴	۶
۵	۱
۶	۵
۷	۴

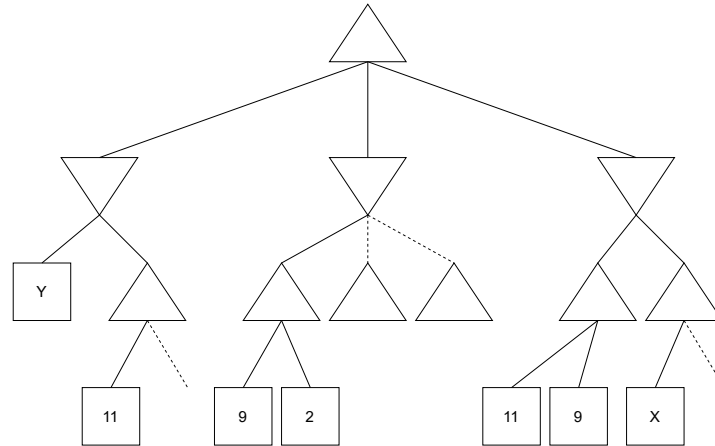
۳. (۲۰ نمره) نمودار زیر گراف محدودیت یک CSP را نشان می‌دهد که فقط محدودیت‌های باینری دارد و در ابتدا هیچ متغیری مقداردهی نشده است.



- (آ) اگر متغیر  $A$  را مقداردهی کنیم، دامنه کدام متغیرها بعد از اعمال forward checking روی  $A$  تغییر خواهد کرد؟
- (ب) اگر متغیر  $A$  را مقداردهی و forward checking را روی آن اجرا کنیم سپس متغیر  $B$  را مقداردهی کنیم، با اجرای forward checking روی  $B$  دامنه کدام متغیرها تغییر خواهد کرد؟
- (ج) اگر متغیر  $A$  را مقداردهی کنیم، دامنه کدام متغیرها بعد از اعمال arc consistency تغییر خواهد کرد؟
- (د) اگر متغیر  $A$  را مقداردهی و arc consistency را اجرا کنیم سپس متغیر  $B$  را مقداردهی کنیم، با اجرای arc consistency دامنه کدام متغیرها تغییر خواهد کرد؟
- حل.

- (آ) اجرای forward checking روی  $A$  یال‌هایی را در نظر می‌گیرد که  $A$  راس آن باشد. بنابراین دامنه رئوس  $B$ ،  $C$  و  $D$  تغییر خواهد کرد.
- (ب) مشابه قسمت قبل اجرای forward checking روی  $B$  یال‌هایی را در نظر می‌گیرد که  $B$  راس آن باشد. اما از آنجایی که  $A$  قبلاً مقداردهی شده است دامنه آن تغییری نخواهد کرد. بنابراین دامنه رئوس  $C$  و  $E$  تغییر خواهد کرد.
- (ج) اجرای arc consistency روی هر متغیر مقداردهی نشده که مسیری به متغیر مقداردهی شده دارد، تأثیر می‌گذارد. این به این دلیل است که تغییر در دامنه  $X$  منجر به تغییر همه یال‌هایی می‌شود که  $X$  یک طرف آن است، بنابراین تغییرات در گراف منتشر می‌شود. همچنین تنها زمانی دامنه  $A$  تغییر می‌کند که دامنه آن خالی شده باشد، در این صورت الگوریتم نیاز به backtracking دارد. بنابراین در نظر گرفتن دامنه‌های جدید در این مورد منطقی نیست. پس دامنه همه متغیرها به غیر از  $A$  تغییر خواهد کرد.
- (د) پس از دادن یک مقدار به  $A$ ، اعمال arc consistency، مقداردهی‌های بعدی و اجرای arc consistency منجر به تغییر در دامنه  $A$  نمی‌شوند. پس دامنه  $D$  نیز تغییر نخواهد کرد زیرا تنها یالی که ممکن است باعث تغییر شود، از  $A$  به  $D$  است که هرگز استفاده نمی‌شود. بنابراین دامنه متغیرهای  $C$ ،  $E$  و  $F$  تغییر خواهد کرد.

۴. (۲۰ نمره) درخت minimax زیر که لایه بالایی آن یک گره ماکسیمم است را در نظر بگیرید:



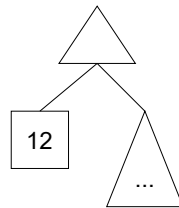
محدوده  $X$  و  $Y$  را طوری تعیین کنید که خطوط نقطه‌چین در هر سه آلفا بتا از چپ به راست هرس شوند. خطوط دیگر نباید برای مقادیر موجود در محدوده شما هرس شوند. توجه کنید که هرس از چپ به راست انجام می‌شود و گره‌های آخر در صورت مساوی بودن هرس می‌شوند.

حل. برای هرس شدن اولین شاخه باید  $Y \leq 11$  باشد. برای شاخه‌های دوم و سوم باید  $Y \leq 9$  باشد و برای شاخه آخر داریم  $11 \leq X$  و  $Y < 11$ . پس:

$$9 \leq Y < 11$$

$$11 \leq X < \infty$$

۵. (۲۰ نمره) در درخت بازی زیر می‌خواهیم utility کلی را بیشینه کنیم. زیر درخت چپ دارای utility ۱۲ است و زیر درخت راست مقدار utility نامشخصی دارد. فردی به شما گفته است که مقدار زیر درخت راست یکی از مقادیر  $-4$ ،  $-8$  و  $21$  است. می‌دانیم که احتمال به دست آمدن هر کدام از این مقادیر یکسان است، اما بدون بررسی این زیر درخت مقدار آن را نخواهیم دانست.



حالا شما ۳ انتخاب دارید. مقدار چپ را انتخاب کنید، مقدار راست را انتخاب کنید یا با دادن هزینه  $c = 1$  زیر درخت راست را ببینید و با فهمیدن utility آن یک تصمیم بگیرید.

(آ) مقدار مورد انتظار<sup>۱</sup> utility برای انتخاب آخر چند است؟

(ب) به ازای چه مقداری از  $c$  باید مورد آخر را انتخاب کنیم؟

حل.

<sup>۱</sup>Expected

(آ) داریم:

$$12 \times \frac{2}{3} + 21 \times \frac{1}{3} - 1 = 14$$

(ب) با توجه به قسمت قبل اگر  $12 > 15 - c$  تصمیم آخر را انتخاب می‌کنیم. پس  $c < 3$ .