בינה מלאכותית - תרגיל תאורטי מס' 1

אורן סמואל, ת"ז 200170694

שאלה 1

סעיף 1

נשלים את הפורמליזציה של הבעיה.

$$
\begin{aligned}
\mathcal{S} &= \{(c,m,b)\ \mid 0 \leq c,m \leq 3,\ b \in \{0,1\}\} \\
\mathcal{I} &= \{(3,3,0)\} \\
\mathcal{G} &= \{\{0,0,1\}\} \\
\mathcal{A} &= \{(c,m,b)\ :\ \text{move } c \text{ cannibals and } m \text{ missionaries from bank } b\}
\end{aligned}
$$

נתאר את פונקציית המעברים באופן מלא:

העברה מגדת המקור לגדת היעד:

שני מיסיונרים: $(c,m,0 \mid\ (m \geq 2)\ \wedge\ (m-2 = c \vee m = 2)) \to (c,m-2,1)$

שני קניבלים: $(c,m,0 \mid c \geq 2\ \wedge\ (m = 3\ \vee\ m = 0)) \to (c-2,m,1)$

קניבל ומיסיונר: $(c,m,0 \mid c,m \geq 1,\ c = m) \to (c-1,m-1,1)$

קניבל אחד: $(c,m,0 \mid\ (c \geq 1)\ \wedge\ (m = 0 \vee m = 3)) \to (c-1,m,1)$

מיסיונר אחד: $(c,m,0 \mid\ (m \geq 1)\ \wedge\ (m-1 = c\ \vee\ m = 1)) \to (c,m-1,1)$

העברה מגדת היעד לגדת המקור:

שני מיסיונרים: $(c,m,1 \mid\ (m \leq 1)\ \wedge\ (m+2 = c\ \vee\ m = 1)) \to (c,m+2,0)$

שני קניבלים: $(c,m,1 \mid\ (c \leq 1)\ \wedge\ (m = 0\ \vee\ m = 3)) \to (c+2,m,0)$

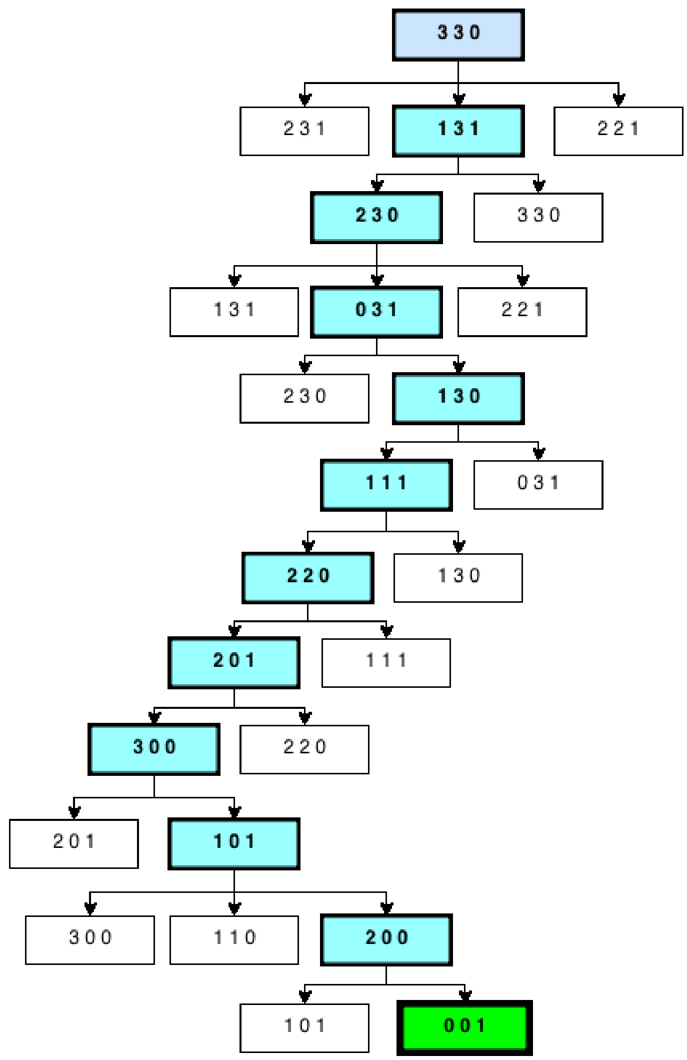
קניבל ומיסיונר: $(c,m,1 \mid c \leq m\ \wedge\ c \leq 2\ \wedge\ m \leq 2) \to (c+1,m+1,0)$

קניבל אחד: $(c,m,1 \mid c \leq 2\ \wedge\ (m = 0\ \vee\ m = 3)) \to (c+1,m,0)$

מיסיונר אחד: $(c,m,1 \mid m \leq 2\ \wedge\ (m+1 = c\ \vee\ m = 2)) \to (c,m+1,0)$

סעיף 2

נמצא פתרון אופטימלי לבעיה באמצעות UCS.
מפאת גודל עץ החיפוש, נציג כאן רק את המסלול אל התוצאה (ואת החלופות בכל צעד):



סעיף 3

לא ייתכן מצב שבו נשקפת סכנה למיסיונרים - פונקציית המעבר בנויה כך שתמיד מתקדמים אל מצב שבו המסיונרים בשתי הגדות לא נמצאים במיעוט.
מכאן, שלא ייתכן מעבר למצב שבו המיסיונרים נאכלים ע"י הקניבלים.

שאלה 2

$ID-DFS-2$ הוא שלם תחת ההנחה שעומק העץ סופי – נניח ש־G קודקוד מטרה בעץ. אז הוא נמצא בעומק כלשהו d_G.

מכיוון שהאלגוריתם סורק את כל העץ עד עומק d_{max} בכל איטרציה שהגיעה ל־$cutoff$, קיימת איטרציה בה $d_{max} \geq d_G$, ובה יימצא הקודקוד G.

$ID-DFS-2$ הוא נאות תחת ההנחה שעומק העץ סופי – האלגוריתם תמיד יגיע לעומק המקסימלי, יסרוק את כל הקודקודים בעומק זה, וכשלא ימצא פתרון יחזיר $failure$.

$ID-DFS-2$ אינו אופטימלי: נניח שהתחלנו את האיטרציה שבה $d_{max}=8$, ויש שני פתרונות – פתרון אופטימלי על הענף הימני ביותר, בעומק 5, ופתרון אחר בענף השמאלי ביותר, בעומק 8.

במקרה כזה, האלגוריתם יגיע לפתרון בעומק 8 לפני זה שבעומק 5, ויחזיר פתרון לא אופטימלי.

סיבוכיות זמן: במקרה הגרוע ביותר, d הוא מהצורה 2^k+1, ואז יהיה עלינו לעבור על כל הקודקודים בעץ עד עומק $d-1$ לפני שנתקדם ל־d_{max} חדש שיאפשר להגיע עד עומק d. לכן הסיבוכיות היא $O\left(b^d\right)$.

סיבוכיות מקום: בדומה ל־DFS, בכל רגע נתון אנו צריכים לשמור בזיכרון רק את המסלול מהשורש עד לקודקוד הנוכחי (ואת הילדים המיידיים של כל קודקוד). בשונה מ־DFS, האלגוריתם יעצור באיטרציה הראשונה בה $d_{max}>d$.

מכאן שסיבוכיות המקום הכוללת תהיה $O\left(b\cdot d\right)$.

שאלה 3

נגדיר את קבוצת המצבים שלנו להיות $S=\{0\dots n\}^n$.

נניח שיש לנו מצב מטרה יחיד $G=(n,0,0,\dots,0)$.

נגדיר את מצב ההתחלה להיות $(0,\dots,0)$, ונגדיר שניתן לעבור ממצב למצב ע"י הגדלה של אחת מכניסות הוקטור ב־1.

במצב כזה, אלגוריתם DFS יתקדם לאורך הענף השמאלי ביותר של העץ ויגדיל את הכניסה הראשונה ב־1 בכל מעבר, עד שיגיע ל־G תוך $O\left(n\right)$ צעדים.

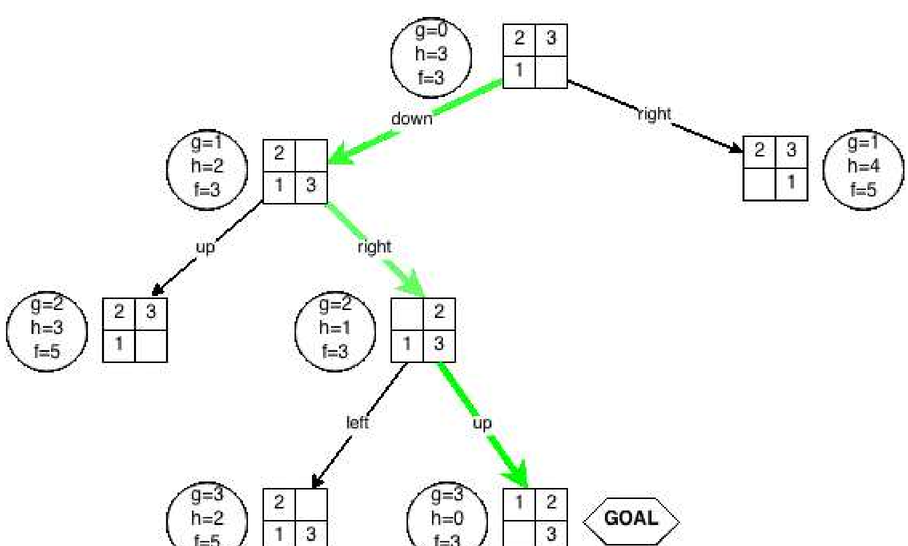
לעומתו, אלגוריתם $ID-DFS$ תחילה יעשה מעבר מלא על העץ עד לעומק d_{max}, לאחר מכן יגדיל את d_{max} ויעשה מעבר מלא על הקודקודים עד עומק זה, וכן הלאה.

מכיוון שהעץ המתאים כאן הוא בגובה n, ולכל קודקוד יש n ילדים, נקבל ש־$ID-DFS$ יעבוד בסיבוכיות זמן $O\left(n^n\right)$.

שאלה 4

מצורף גרף החיפוש עבור הבעיה.

החיצים המודגשים מציינים את המסלול האופטימלי אל הפתרון.



שאלה 5

h אינה אדמיסבילית. נניח שיש על הלוח שתי נקודות x_1, x_2, כך שהקבוצה $\{position, x_1, x_2\}$ יוצרת משולש שווה צלעות שאורך צלעו 1.
המרחק מכל נקודה לזו הקרובה ביותר אליה הוא 1, ולכן $h(s) = 3$. לעומת זאת, הפתרון האופטימלי עבור פקמן הוא התקדמות לנקודה הראשונה (מרחק 1) ואז לשניה (מרחק 1), ובסה"כ מרחק 2.
כלומר, במקרה זה הפונקציה h נותנת הערכה שגבוהה מהפתרון האופטימלי, ולכן היא אינה אדמיסבילית.

שאלה 6

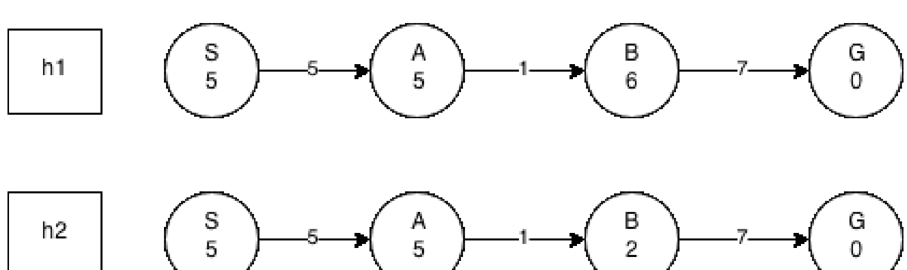
טענה: אם h היא ϵ־אדמיסבילית, ו־A^* (בחיפוש עץ) המשתמש ב־h מחזיר פתרון C, אז $C \leq C^* + \epsilon$.
הוכחה:
h היא ϵ־אדמיסבילית, אז יש h^* כך שלכל מצב $s \in S$ מתקיים $h(s) \leq h^*(s) + \epsilon$.
כעת, בריצת A^* עם h, עלות ההגעה לקודקוד s היא $g(s)$. נוסיף ערך זה לשני הצדדים ונקבל:
$$g(s) + h(s) \leq g(s) + h^*(s) + \epsilon$$
מכיוון ש־h^* אדמיסבילית, תמיד מתקיים $g(s) + h^*(s) \leq C^*$, ומכאן $g(s) + h(s) \leq C^* + \epsilon$.
עבור קודקוד מטרה $v \in G$ מתקיים $g(v) = C$, ונתון ש־$h(v) = 0$, ובסה"כ נקבל $C \leq C^* + \epsilon$ כנדרש.

שאלה 7

טענה: אם h יוריסטיקה קונסיסטנטית, אז היא אדמיסבלית.
הוכחה:
צ"ל שלכל קודקוד n מתקיים $h(n) \leq c(n)$, כאשר $c(n)$ הוא העלות המינימלית האמיתית מ־n לקודקוד מטרה.
נוכיח באינדוקציה על אורך המסלול מ־n לקודקוד מטרה.
בסיס: אם n קודקוד מטרה בעצמו, אז מהגדרת קונסיסטנטיות, $h(n) = 0$, כלומר $h(n) = c(n)$ ולכן h אדמיסבילית.
אם n איננו קודקוד מטרה, נניח נכונות למסלול באורך k ונוכיח עבור מסלול באורך $k+1$.
מקונסיסטנטיות h, מתקיים $h(n) \leq c(n, a, n') + h(n')$ לכל יורש n' של n שנוצר מהפעולה a.
אבל לכל n' כנ"ל, המסלול הקצר ביותר ממנו לקודקוד מטרה הוא באורך $k-1$, ולכן מקיים את הנחת האינדוקציה - $h(n') \leq c(n')$.
לכן $h(n) \leq c(n, a, n') + c(n')$ לכל יורש n' של n, ובפרט לזה עבורו $c(n, a, n') + c(n')$ מינימלי.
נשים לב שערך מינימלי זה הוא בדיוק $c(n)$, ולכן $h(n) \leq c(n)$, כלומר h אדמיסבילית כנדרש.

שאלה 8

טענה: אם h_1 שולטת על h_2 ו־h_1 קונסיסטנטית, אז h_2 קונסיסטנטית.
דוגמה נגדית:



בשרטוט לעיל מתואר אותו הגרף עם שתי יוריסטיקות שונות h_1, h_2, שתיהן אדמיסביליות, ו־h_1 שולטת על h_2.
עם זאת, נשים לב ש־h_1 קונסיסטנטית, אבל h_2 אינה קונסיסטנטית - עבור הקודקוד A מתקיים: $h(A) = 5$, $c(A, B) + h(B) = 1 + 2 = 3$, כלומר $h(A) > c(A, B) + h(B)$.
בסה"כ נקבל ש־h_1 קונסיסטנטית ושולטת על h_2, אבל h_2 אינה קונסיסטנטית, בסתירה לטענה.

שאלה 9

סעיף 1

בהינתן בעיית knapsack $P=\langle I,(v_1,\ldots,v_n),(w_1,\ldots,w_n)\rangle$, נרצה לקבל את התת־קבוצה של I הממקסמת את סכום הערכים v_j תוך שמירה על סכום משקלים קטן מ־W.
נגדיר פונקציית ערך ופונקציית משקל עבור מצבים:

$$\forall J\subseteq I\ :\ val(J)=\sum_{j\in J}v_j\quad weight(J)=\sum_{j\in J}w_j$$

וכעת נגדיר את הפונקציה המתארת את דרישות הבעיה:

$$f(P)=\underset{J\subseteq I}{argmax}\{V=val(J)\ \mid weight(J)\le W\}$$

מצב בבעיה P הוא תת־קבוצה $J\subseteq I$.
שני מצבים $J_1,J_2\subseteq I$ הם שכנים אם $|J_1\triangle J_2|=1$, כלומר הם שונים זה מזה בדיוק באיבר אחד ־ בין אם בתוספת, בגריעה או בהחלפה של איבר ־ תחת האילוץ לפיו לא ניתן לעבור למצב J עבורו $\sum_{j\in J}w_j>W$
המצב ההתחלתי הוא $J_s=\emptyset$.
נגדיר פונקציית ערך באופן הבא:
כעת כדי לבצע gradient ascent, נרצה לעבור ממצב J למצב שכן K, כך שהערך של K גדול מהערך של J, וגם גדול מבין כל השכנים של J, כלומר:

$$val(K)\ge val(J)\ \wedge\ \forall L\in neighbors(J)\ :\ val(K)\ge val(L)$$

סעיף 2

כנראה ש־ simulated annealing ייתן ביצועים טובים יותר, מפני שבבעיה שמולנו יש בעיה חמורה של נקודות מקסימום מקומי.
נניח, לדוגמה, ש־$W=6$ ויש לנו בבעיה את הפריטים הבאים:

$$\begin{aligned}(v_1,w_1)&=(4,4)\\(v_2,w_2)&=(2.5,3)\\(v_3,w_3)&=(2.5,3)\end{aligned}$$

במצב כזה, לפריט 1 יש את הערך הסגולי הגבוה ביותר (1), אך אם נכניס אותו, לא יוותר לנו מקום לפריטים נוספים, ונחזיר פתרון שערכו $V=4$.
לעומת זאת, הפתרון האופטימלי הוא פריטים $2,3$, שכן משקלם הכולל הוא 6, וערכם הוא $V'=5>V$.
מכאן, שבמצב הנ"ל היינו מרוויחים מביצוע מהלך "שלילי", בו היינו מוציאים את פריט 1, ומכניסים במקומו את אחד מהפריטים $2,3$.

סעיף 3

פוקנציית כשירות

בהינתן מצב $J\subseteq I$, פונקציית הכשירות תהיה הערך של הפתרון, תוך שמירה על אילוץ המשקל:

$$fit(J)=\begin{cases}val(J) & weight(J)\le W\\ 0 & o.w\end{cases}$$

סלקציה

ניתן לבצע סלקציה בשתי צורות, שאין לי דרך טובה לחזות מי מהן עדיפה:
הראשונה היא 'חיתוך' של המצבים החלשים ביותר ־ כלומר, להתקדם לאיטרציה הבאה עם, לדוגמה, 80% המצבים עבורם fit הוא הגדול ביותר.
השניה היא "ראש בראש" ־ לבחור זוגות רנדומליים של מצבים, ומכל זוג לבחור את המצב עבורו fit גבוה יותר.

הכלאה

אם נתייחס למצב, שוב, בתור וקטור באורך n מעל $\{0,1\}$, נוכל לבצע רקומבינציה פשוטה של שני מצבים J_1, J_2 ע"י בחירת נקודה רנדומלית $k \in \{1,\dots,n\}$, ויצירת וקטור חדש J_3 מ־k הכניסות הראשונות ב־J_1 ו־$n-k$ הכניסות האחרונות ב־J_2.
שיפור אפשרי עבור בחירת k ע"י מציאת הנקודה הממקסמת את "הערך הסגולי" של הפתרון החדש שנוצר, כלומר הנקודה עבורה $\frac{val(J_3)}{weight(J_3)}$ מקסימלי.
ניתן להכניס כאן אלמנט הסתברותי נוסף ־ מכיוון שייתכן שחלק מההורים נותנים תוצאה טובה בעצמם, ניתן להחליט שמבצעים רקומבינציה רק בהסתברות 75% (לדוגמה), ובהסתברות 25% משאירים את ההורים הנבחרים כפי שהם.

מוטציה

אם נתייחס למצב J בתור וקטור באורך n מעל $\{0,1\}$, מוטציה תהיה היפוך של כל כניסה בוקטור בסיכוי $\frac{1}{n}$.
כלומר, בתוחלת, בכל מצב נוריד או נוסיף פריט יחיד.
אם הפעולה גורמת למעבר על אילוץ המשקל, נחזיר את הוקטור למצבו המקורי.

סעיף 4

כמו בסעיף 2, כנראה שאלגוריתמים גנטיים יתנו כאן תוצאה טובה יותר.
אלגוריתמי iterative improvement ייטו יותר להיתקע בנקודות מקסימום לוקאלי, בעוד שהאלמנטים הסטוכסטיים באלגוריתם הגנטי ימנעו מצב זה, ויכוונו את סט המצבים, בסיכוי גבוה יותר, לעבר נקודת המקסימום הגלובלית.
עם זאת, ללא אלגוריתם מדויק ובדיקת ביצועים אמיתית, לא ניתן להגריע כאן באופן מובהק.

שאלה 10

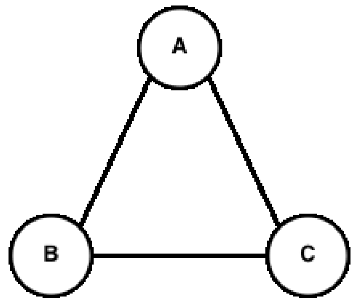
סעיף 1

נתאר את הבעיה באופן פורמלי:
נגדיר את קבוצת המשתנים: $\mathcal{X} = \{A, B, C\}$. כל משתנה מייצג את זמן היציאה של הרכבת המתאימה לו.
הדומיין הוא אחד לכל המשתנים: $\mathcal{D} = \{8, 9, 10\}$. כלומר, לכל רכבת אנו רוצים להתאים זמן יציאה ־ 8:00, 9:00 או 10:00.
קבוצת האילוצים היא $\mathcal{C} = \{\langle\langle A,B\rangle, R_{AB}\rangle, \langle\langle A,C\rangle, R_{AC}\rangle, \langle\langle B,C\rangle, R_{BC}\rangle\}$ כאשר:

$$\begin{aligned}
R_{AB} &= \{(8,10),(9,8),(10,8),(10,9)\} \\
R_{AC} &= \{(9,8),(10,9),(10,8)\} \\
R_{BC} &= \{(8,9),(8,10),(9,10),(10,8)\}
\end{aligned}$$

סעיף 2

גרף האילוצים הוא גרף מלא על שלושת המשתנים:



סעיף 3

בתחילת הריצה, כל הערכים ייתכנו לכל המשתנים:

$$A : \{8, 9, 10\}\,,\;\; B : \{8, 9, 10\}\,,\;\; C : \{8, 9, 10\}$$

מכיוון שלכל המשתנים אותו מספר ערכים אפשרי, נבחר שרירותית את A.
הערך 8 עבור A ישאיר את B עם ערך אחד אפשרי ואת C ללא ערכים אפשריים.
הערך 9 עבור A ישאיר את B עם ערך אחד אפשרי ואת C עם ערך אחד אפשרי.
הערך 10 עבור A ישאיר את B עם שני ערכים אפשריים ואת C עם שני ערכים אפשריים.
לכן נגדיר $A = 10$, ונתקן את הדומיינים של B, C ע"י שימוש ב־arc consistency:

$$A : \{10\}\,,\;\; B : \{8, 9\}\,,\;\; C : \{8, 9\}$$

כעת ל־B, C אותו מספר ערכים אפשרי, אז נבחר שרירותית את B.
הערך 8 עבור B משאיר את C עם שני ערכים אפשריים.
הערך 9 עבור B משאיר את C עם ערך אפשרי אחד.
לכן נגדיר $B = 8$, ונתקן את הדומיינים של A, C ע"י שימוש ב־arc consistency:

$$A : \{10\}\,,\;\; B : \{8\}\,,\;\; C : \{9\}$$

הגענו לנקודה בה לכל משתנה יש ערך אחד אפשרי, כך שכל הערכים עומדים באילוצים, ולכן סיימנו.