

Uniform Cost Search

(ואריאציה על BFS)

Cheapest – first strategy ↑
את המסלול הזול ביותר מצומת ההתחלה לצומת
המטרה. (מציאת פתרון בעל מחיר מינימאלי)

החיפוש מתקדם בשכבות של מחירים (costs) ↑
שווים.

מבטיח כי צומת המטרה הראשון שימצא הינו הזול ↑
ביותר.

Uniform Cost Search – cont'

במימוש: כאשר נבחר צומת לפיתוח (מתוך OPEN) נבחר את הצומת הזול ביותר (בעל ה-cost הנמוך ביותר).

המחיר (cost) בכל צומת הוא מחיר המסלול מצומת ההתחלה ועד לצומת זה.

אלגוריתם זה אינו בהכרח שלם.

כדי להבטיח מציאת פתרון: כאשר מתקדמים במסלול מחירו לא יקטן.

$$\forall n \text{ cost}(\text{successor}(n)) \geq \text{cost}(n)$$


Hill climbing search

האסטרטגיה: 

- פתח צומת
- בדוק מהם בניו החדשים (newly developed successors)
- בחר את ה"בן המוצלח" ביותר ועבור ל-1.

אסטרטגיה הבוחרת את הכיוון התלול ביותר ביחס למצב הנוכחי (כמו מטפסי הרים הבוחרים את הכיוון תלול ביותר על מנת להגיע לפסגת ההר מהר ככל האפשר).

אסטרטגיה המבוססת על אופטימיזציה לוקאלית. 

שימוש באסטרטגיה זו כמעט ואינו מצריך מאמץ חישובי, אין צורך לזכור את ניסיונות העבר או את המסלול שהוביל אותנו למצב הנוכחי. בכל רגע בחיפוש נשמר צומת אחד בלבד. 

Hill climbing search – cont'

חסרונות האסטרטגיה:



פונקצית הערכה מטעה עלולה להביא למסלול חיפוש עמוק מאוד ואף אינסופי שלמעשה אינו מוביל לפתרון.



כאשר מגיעים למקסימום לוקאלי (צומת בעל ערך גבוה יותר מבניו) – אין לאן "לטפס" יותר ולכן החיפוש מסתיים ללא מציאת הפתרון.



(הדרך היחידה לצאת ממבוי סתום זה היא להתחיל את החיפוש מחדש מצומת שעדיין לא פותח)

4. אסטרטגיה זו הינה בלתי הפיכה (**irrevocable**) כיוון שלא ניתן לחזור אחורה לאלטרנטיבות שלא נבחרו לפיתוח.

5. רמה – אזור בו פונק' ההערכה היא שטוחה (flat) ואז אסטרטגית החיפוש תהיה רנדומאלית (random walk)

6. האסטרטגיה אינה שלמה.

Hill climbing search – cont'

יתרונות האסטרטגיה: 

יעילה כאשר פונקצית ההערכה שבידינו מאוד אינפורמטיבית, מגינה עלינו מפני מקסימום לוקאלי ולכן מובילה אותנו ישירות ובמהירות אל ה"פסגה" (הפתרון).

דרישות זיכרון נמוכות. 

Hill climbing על ואריאציות 

Simple hill climbing

במהלך פיתוח הבנים: ברגע שנמצא את הבן הראשון ה"מוצלח" יותר מהצומת הנוכחי - נעבור אליו.

Best – First Search

אסטרטגיה זו בכל שלב, בוחרת לפתח את הצומת
ה"טוב ביותר" (מבחינה יוריסטית) מבין כל
הצמתים שהתגלו עד כה.

האלגוריתם:

מבני הנתונים בשימוש:

- OPEN – רשימת כל הצמתים שהתגלו אך טרם פותחו.
- CLOSED – רשימת כל הצמתים שפותחו.

$f(\cdot)$ – פונקציית הערכה יוריסטית
(heuristic evaluation function)

Best – First Search – cont'

האלגוריתם:

שים את צומת ההתחלה, s ב- OPEN.

אם OPEN ריקה:

סיים עם שגיאה – לא קיים פתרון!

אחרת:

המשך.

6. הוצא מ- OPEN צומת n בעל f מינימאלי, שים אותו ב- CLOSED.

7. פתח את צומת n , ייצר את כל בניו ועדכן לכל אחד מהם מצביע חזרה לאב (n).

8. אם אחד מבניו של n הוא צומת המטרה:

סיים בהצלחה – הפתרון הוא המסלול המתקבל בהליכה לאחור מצומת המטרה ל- s דרך המצביע לאב בכל צומת.

לכל n של n בצע:

חשב את $f(n')$

אם n' לא נמצא ב- OPEN ולא ב- CLOSED הוסף אותו ל- OPEN ועדכן בו את ערך $f(n')$.

אם n' נמצא ב- OPEN או ב- CLOSED, השווה את $f(n')$ עם הערך הקיים ב- n' :

אם הערך הקיים ב- n' (old value) נמוך יותר: התעלם מהצומת החדש שנוצר.

אחרת:

החלף את n' ה"ישן" עם הצומת החדש שיוצר.

אם n' ה"ישן" נמצא ב- CLOSED:

העבר אותו ל- OPEN.

עבור ל-2.

Best – First Search – cont'

יתרונות:



חיפוש מהיר כאשר הפונקציה היוריסטית טובה.



לא סובל מבעיית מינימום לוקאלי.



חסרונות:



דורש זיכרון רב.




מוצא פתרונות לא אופטימליים.



חיפוש אלומה (Beam Search)

(ואריאציה על Best First)

מגביל את מספר הצמתים שנשמרים לפיתוח לקבוע מסוים k . 

אם פיתוח צומת מסוים (הוספת בניו) מגדיל את מספר הצמתים לערך הגדול מ- k , נבחר את ה- k הטובים ביותר (לפי הערך היוריסטי). 

Hill Climbing search שקול ל-Beam search כאשר $k=1$. 

Best First search שקול ל-Beam search כאשר k שווה אינסוף. 

משפר את דרישות הזיכרון מול Best First. 