

חיפוש משולב Best First + DFS

↑ מחפשים Best First עד לניצול (כמעט) כל הזיכרון. כאשר אוזלים משאבי הזיכרון מבצעים חיפוש DFS (או Backtracking) מהצומת הטוב ביותר. (ניתן לבצע העמקה הדרגתית מהצומת הטוב ביותר).

↑ משלב חסכון בזיכרון (מול DFS) עם שמירת ענפים העלולים להיות קריטיים (מול Best First).

Greedy Search

ואריאציה על Best-first search.



פונקציה הערכה יוריסטית $h(n)$ = הערכה של המרחק מ- n למטרה.



Greedy search מפתח את הצומת שנראה כקרוב ביותר למטרה.



Greedy search אינו אופטימאלי ואינו שלם.



מספר הצמתים שיפתח במקרה הגרוע הוא :



$$O(B^L)$$

A* Search

אסטרטגיה זו הינה ואריאציה משופרת של
Best – First search.

$g(n)$ – מחיר המסלול הקצר ביותר מצומת
ההתחלה לצומת n .

$h(n)$ – מחיר מוערך של המסלול ה"זול" ביותר
מצומת n לצומת המטרה.

$f(n) = g(n) + h(n)$ – מחיר מוערך של מסלול
הפתרון ה"זול" ביותר דרך n .

אסטרטגיה זו שלמה ואופטימאלית בהינתן h שאף
פעם לא מעריכה יתר על המידה (overestimate)
את המחיר לצומת המטרה.

A* Search – cont'

האלגוריתם: 


שים את צומת ההתחלה, s ב- OPEN. 

אם OPEN ריקה: 

סיים עם שגיאה – לא קיים פתרון!

אחרת:

המשך.

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


אם n הוא צומת המטרה: 


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


אחרת:

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

לכל בן n' של n בצע:

אם n' לא נמצא ב- OPEN ולא ב- CLOSED, חשב $g(n')=g$ את $h(n')$ וחשב את $f(n')=g(n')+h(n')$. כאשר $g(n')=g(n)+c(n, n')$ ו- $g(s)=0$. 

אם n' נמצא ב- OPEN או ב- CLOSED, כוון את מצביעיו למסלול בו $g(n')$ הוא הנמוך ביותר. 

אם עדכנו את מצביעי n' ו- n' נמצא ב- CLOSED, העבר אותו ל- OPEN. 

עבור ל-2. 

תכונות A^*

הגדרה: אלגוריתם חיפוש הוא קביל (admissible) אם מובטח שהוא ימצא פתרון אופטימאלי (בעל מחיר מינימאלי) כאשר קיים פתרון.



נגדיר: $h^*(n)$ - המחיר הנמוך ביותר של מסלולים ממצב n למטרה (המחיר האמיתי למטרה).



הגדרה: פונקציה יוריסטית אופטימית אם מתקיים:



$$\forall n \quad 0 \leq h(n) \leq h^*(n)$$

A^* עוצר על גרפים סופיים.



A^* שלם על גרפים סופיים ואינסופיים.



A^* קביל בתנאי ש- h אופטימית.



תכונות A^* - המשך

↑ נגדיר C^* - מחיר אופטימאלי (מינימאלי) לפתרון.

↑ למה: בכל שלב לפני ש- A^* עוצרת, קיים ב-
OPEN צומת חהשייך למסלול האופטימאלי
המקיים $f(n) \leq C^*$.

↑ הערה: A^* מבטיח מציאת מסלול אופטימאלי,
מהירות ההגעה לפתרון תלויה בפונקציה
היוריסטית h .

Iterative Deepening A* Search

↑ תזכורת: העמקה הדרגתית - טכניקה להקטנת דרישות הזיכרון.

↑ כל איטרציה היא בעצם DFS כאשר ההגבלה לא נעשית על העומק אלא על $f(n)$.

↑ אסטרטגיה זו שלמה ואופטימאלית (מ- A^*).

↑ דרישות הזיכרון של אסטרטגיה זו פרפורציונאליות לאורך המסלול המקסימאלי המפותח. (חסכוני)

חיפוש דו-כיווני

כאשר ניתן להשתמש בפונקציה הופכית לפונקצית 

$$[successors(s)]^{-1} = s \quad \text{המעבר:}$$

ניתן לחפש בשני כיוונים במקביל: ממצב ההתחלה לכיוון המצב הסופי ומהמצב הסופי לכיוון מצב ההתחלה.