

אסטרטגיות חיפוש

• אסטרטגית חיפוש מגדירה כיצד יש לחפש במרחב.

• האסטרטגיות נבדלות בצורה בה הן בוחרות את הצומת הבא לפיתוח.

• האסטרטגיות נבדלות גם בהחלטה אילו מצבים ישמרו בזיכרון.

Depth First Search (DFS)

- חיפוש לעומק - ניתנת עדיפות לצמתים ה"עמוקים" ביותר בגרף החיפוש.
- לכל צומת הנבחר לפיתוח, מיוצרים כל בניו לפני שצומת נוסף נחקר.
- לאחר פיתוח צומת, אחד הבנים החדשים שנוצרו נבחר לפיתוח וכך הלאה עד שמסיבה מסוימת ההתקדמות נחסמת.
- כאשר נוצרת חסימה, התהליך מתחדש מהמקום בו פותחו הצמתים העמוקים ביותר, כלומר, מנקודת ההחלטה הקרובה ביותר בה קיימות אלטרנטיבות שטרם נחקרו.
- האסטרטגיה טובה כאשר קיימים מספר פתרונות השווים באיכותם או כאשר קיימות אזהרות מהימנות לכיוונים השגויים.

Depth First Search – cont'

➤ אסטרטגיה זו עלולה להיות מסוכנת במקרה של גרף חיפוש גדול מאוד, אינסופי או בגרף המכיל מעגלים.

➤ לכן, בדרך כלל, מוסיפים לאלגוריתמי DFS חסם עומק (depth bound) המהווה כלל עצירה המחזיר את הפיתוח אחורה לאלטרנטיבה העמוקה ביותר שלא עברה את החסם.

כלומר, התוכנית תיסוג כאשר יתקיים אחד משני התנאים:

➤ חריגה מהעומק המקסימאלי (depth bound).

➤ צומת זוהה כמבוי סתום.

Depth First Search – cont'

האסטרטגיה: ➤

מבני הנתונים בשימוש:

- מחסנית OPEN: תכיל את כל הצמתים שיוצרו. אסטרטגית ה- depth first תשים את הבנים החדשים ביותר שיוצרו בראש המחסנית ותבחר לפיתוח את הצומת הנמצא בראש המחסנית. (LIFO)
- רשימה CLOSED: תכיל את כל הצמתים שפותחו לפי סדר פיתוחם. אסטרטגית ה- depth first תוסיף לסוף הרשימה את הצומת המפותח בכל שלב. בכל רגע נתון הרשימה CLOSED מכילה מסלול יחיד מצומת ההתחלה לצומת הנוכחי המפותח (כשנגיע לצומת המטרה נוכל לשחזר את מסלול הפתרון מרשימה זו).

Depth First Search – cont'

האלגוריתם:



1. שים את צומת ההתחלה על OPEN.

2. אם OPEN ריקה
סיים שגיאה.

אחרת:

המשך.

➤ הסר את הצומת העליון מ- OPEN (נקרא לצומת זה n),
שים אותו ב- CLOSED.

➤ אם העומק של n שווה לחסם העומק (depth bound):
נקה את CLOSED ועבור ל-2.

אחרת:

המשך.

➤ פתח את n , ייצר את כל בניו ושים אותם בראש OPEN
(ללא חשיבות לסדר). כל בן ישמור מצביע אל האב (n).

➤ אם אחד הבנים הוא צומת המטרה:
סיים. הפתרון הוא המסלול המתקבל הליכה
לאחור מצומת המטרה דרך המצביעים לאבות.

אחרת:

המשך.

➤ אם אחד הבנים הוא מבוי סתום:
הסר אותו מ- OPEN ונקה את CLOSED.

➤ עבור ל-2.

Depth First Search – cont'

הגדרה: אסטרטגיה תקרא

(צמתי) הפתרון לא משפיע על סדר פיתוח הצמתיים.

➤ אלגוריתם ה-DFS, כפי שהוצג לעיל, הינו אלגוריתם לא מיועד לחלוטין.

ואריאציה על DFS:

הבנים (successors) נפעיל עליהם פונקצית הערכה יוריסטית. בעת הכנסת הבנים ל- OPEN נסדרם לפי ערכי הפונקציה היוריסטית כך שהצומת הבא לפיתוח יהיה בעל הערך היוריסטי הגבוה ביותר.

➤ הוואריאציה הנ"ל הינה גרסה מיועדת חלקית (partly informed) של חיפוש DFS.

Depth First Search – cont'

נגדיר: ➤

D – Depth bound

B – Branching degree

L – Solution path length

$$D * B$$

דרישות הזיכרון של DFS: ➤

מספר צמתים שיפותחו: ➤

best case: L

worst case: $\frac{B^{D-1}-1}{B-1}$

בעיה: כאשר הגרף אינו עץ או עלולים לפתח את אותו צומת יותר מפעם אחת! ➤

פתרון: בכל פעם נבדוק האם אחד מהצמתים החדשים שיוצרו (successors) כבר נמצא על מסלול השוטטות בעץ, במידה ונמצא נמחק אותו מ-OPEN. ➤

Depth First Search – cont'

➤ הגדרה: אלגוריתם חיפוש יקרא שלם אם מובטח כי יעצור וימצא פתרון במידה וקיים כזה.

➤ אמנם מובטח כי DFS עם הגבלת עומק יעצור, אך לא מובטח כי ימצא את הפתרון גם אם קיים כזה.

● DFS שלם אם $D \geq L$.

● DFS אינו שלם אם $L > D$.

➤ כאשר נמצא פתרון, לא מובטח כי זהו הפתרון בעל המסלול הקצר ביותר מצומת ההתחלה.