מיני פרויקט ביישומים של מדעי המחשב

הוגש ע״י: עודד סנדק- 316081256 שחר קמחי - 209490960

מבוא:

בפרויקט הבא אנו עומדים לחקור בעיה מוכרת במדעי המחשב, בעיית הbin-packing. בחלק הראשון של הדו״ח נסקור כיצד השתמשנו באלגוריתמים אבולוציוניים על מנת לפתור את בעיית הbin-packing. בחלק השני נתאר את האתר שיצרנו אשר מספק סביבה נעימה וידידותית למשתמש על מנת לחקור תוצאות של אלגוריתמים מוכרים לפתרון בעיית הbin-packing אל מול תוצאות של האלגוריתם האבולוציוני שכתבנו.

תיאור הבעיה:

נתונים n חפצים, לכל אחד מהם נפח a_1,...,a_n. בנוסף נתונים מספר לא מוגבל של דליים שכל אחד מהם בנפח c , כך שכל אחד מ-n החפצים בעל נפח קטן או שווה ל-c . יש לארוז את החפצים במספר מינימלי של דליים, כך שבכל דלי הנפח הכולל של החפצים לא יעלה על נפח הדלי. באופן פורמלי, יש לחלק את המספרים a_1,...,a_n למספר מינימלי של קבוצות זרות, שסכום כל קבוצה לא יעלה על c.

תיאור הפתרון בעזרת אלגוריתם אבולוציוני:

מבנה הפרט:

הפרט הינו מערך בגודל n (כמספר החפצים) כאשר כל תא i מתאר את מספר הדלי בו החפץ בנפח a_i נמצא, והערך בכל תא אינו גדול מ-n, כלומר במקרה הגרוע ביותר, הפרט יהיה מערך בגודל n שמכיל n מספרים שונים (כלומר כל חפץ נמצא בדלי משל עצמו) . לדוגמא עבור המערך של החפצים הבא: [4,5,8,6,12,10] ודלי בגודל 12, פרט שפותר את הבעיה יכול להיראות כך: [1,2,1,2,3,4]. נציין כי המספרים במערך אינם חייבים להיות עוקבים ופתרון שמיצג את אותה חלוקה לדליים יכול להיראות כך [3,6,3,6,1,5].

מבנה האוכלוסייה:

האוכלוסייה בנויה מ200 וקטורים בגודל n כאשר בכל וקטור ערכים מ1 עד n . הוקטורים נוצרים ע״י הפונקציה GAIntVectorCreator.

האבולוציה:

fitness: פונקציית הfitness מקבלת פרט ומחשבת את הערך שלו ע״י כך שהיא מחשבת את מספר האיברים השונים בוקטור. לאחר מכן הפונקציה עוברת על ערכי הפרט ויוצרת מערך חדש בגודל n בו בתא הi נמצא סכום משקלי החפצים שנמצאים בדלי ה-i בוקטור (כלומר נקבל מערך שמכיל את הנפח שהוכנס לכל דלי). לאחר מכן הפונקציה עוברת על המערך החדש ובודקת שאף ערך במערך אינו גדול מהנפח של הדלי. אם נמצא תא בעל ערך גדול מנפח הדלי, פונקציית הfitness תחזיר אינסוף. רק לאחר שעברנו על כל ערכי המערך החדש, וראינו שבאף דלי אין יותר פריטים ממה שהוא יכול להכיל, נחזיר את מספר האיברים השונים בוקטור שחישבנו בהתחלה, וזה יהיה ערך הפרט.

selection: הselection יבוצע בשיטת ה tournament , כאשר כל פעם 4 פרטים יתחרו ביניהם למי הfitness הטוב ביותר, כאשר המנצח מבניהם ממשיך לדור הבא.

mutation: על כל אחד מהפרטים באוכלוסייה אנו מבצעים מוטציה של כניסה אחת מוקטור הפרט בהסתברות של 0.05.

crossover: בהסתברות של 0.5 נבצע crossover של שתי כניסות בוקטור.

מבנה התוכנה:

התוכנה שבנינו הינה אתר אינטרנט. בראש הדף ניתן הסבר קצר על מה האתר כולל ומה התכונות של הדברים שהוא מציג. לאחר מכן ישנן 3 תיבות קלט שאליהן מכניסים את הפרטים עבור פתרון הבעיה, נפח הדלי, מספר האיברים במערך החפצים ומספר הדורות שאנו מעוניינים שהאלגוריתם האבולוציוני יבצע. ליד כל תיבת קלט ישנו גם כפתור שמאפשר בחירה של ערך רנדומלי. התוכנה מבצעת את האלגוריתמים הבאים לפתרון בעיית הbin-packing:
Next fit, first fit, best fit, first fit decreasing, וכמובן האלגוריתם האבולוציוני. לאחר לחיצה על הכפתור מוצג מערך בו נמצאים הנפחים של החפצים שעליהם בוצע כל אחד מהאלגוריתמים, ונפח הדלי שאליו ניסו להכניס את החפצים. האתר גם מציג השוואה בין כמות הדליים הנדרשים לפי כל אלגוריתם, והשוואה של זמן הריצה שלהם. התוכנה פועלת באופן הבא: צד הלקוח הכולל את תיבות הקלט, הכפתורים, הטבלאות וכל שאר הרכיבים אותם אנו רואים על המסך נכתבו באמצעות angular. כאשר לוחצים על הכפתור של פתרון הבעיה, נשלחת בקשת API לשרת express (node js) . השרת מקבל את המידע מצד הלקוח, ומעביר אותו לקובץ python אשר מריץ כל אחד מהאלגוריתמים הידועים לפתרון הבעיה המצוינים לעיל, ובנוסף קורא לקובץ python נוסף שמבצע את האלגוריתם האבולוציוני באמצעות הספרייה ec-kity. המידע מריצת האלגוריתמים מוחזר ללקוח בעזרת השרת.

הדגמת ריצה ותוצאות:

הדרך הטובה ביותר לראות הדגמת ריצה, תוצאות וגרפים, היא פשוט להריץ את האתר בעצמכם. בקובץ ה-README של הפרויקט בגיטהאב ישנן הוראות כיצד להריץ את הפרויקט על המחשב שלכם. האתר מובן וידידותי, ויספק לך חוויה נעימה, תוצאות והשוואות. בנוסף יש בקובץ ה- README יש קישור לסרטון ביוטיוב שמראה כיצד להריץ את הפרויקט ודוגמת ריצה שלו.

מסקנות:

לאחר הרצות רבות וניסויים רבים של האתר (אתם יותר ממוזמנים גם לנסות בעצמכם) הגענו למסקנה שהאלגוריתם האבולוציוני מגיע בערך אל אותן תוצאות כמו האלגוריתמים המוכרים לפתרון הבעיה, גם אם מספר הדורות שהוא רץ גדול מאוד. ההבדל העיקרי בין האלגוריתם האבולוציוני לאלגוריתמים המוכרים הוא שזמן הריצה של האלגוריתם האבולוציוני ארוך בהרבה מזמן הריצה של האלגוריתמים המוכרים, וכפי שציינו קודם אינו נותן תוצאות טובות יותר, ולכן המסקנה העיקרית מהפרויקט היא שאלגוריתמים אבולוציוניים אינם אופטימליים בפתרון בעיית הbin-packing .