

## סיכום מאמר: Hybrid recursive number partitioning (Korf, 2011)

מגיש: כפיר גולדפרב

### אבסטרקציה:

הבעיה: בהינתן  $n$  מספרים טבעיים (אי-שליליים) ומספר  $k$ , נדרש לחלק את המספרים ל- $k$  קבוצות שונות כך שסכום הקבוצות כמה שיותר קרוב (סכומים קרובים בין כל הקבוצות).

קיימים מספר אלגוריתמים ידועים אשר יודעים לחלק בצורה טובה עבור חלוקה דו-כיוונית (חלוקה עבור 2 קבוצות), אבל חלוקה שהיא מולטי-כיוונית (חלוקה עבור מספר  $k$  קבוצות) היא יותר מאתגרת ומסובכת, במאמר זה ננסה לשלב מספר אלגוריתמים הפותרים בעיה זו בצורה מולטי-כיוונית.

כותבי המאמר עשו ניסויים על האלגוריתמים המוצגים במאמר על מספרים הבנויים מ-31 ביטים.

### הקדמה:

דוגמה לחלוקה בין 2 קבוצות, נניח שנקלט קיבלנו את הקבוצה:  $\{8,7,6,5,4\}$  אם נחלק אותם לשני קבוצות:  $\{8,7\}$  ו-  $\{6,5,4\}$  הסכום של כל קבוצה הוא 15, במצב כזה ניתן לומר שהתוצאה היא אופטימלית (כי שווה לחלוטין) או במילים אחרות: חלוקה מושלמת, במידה והסכום של המספרים לא ניתן לחלק במספר התתי-קבוצות – אז סכומי התתי-קבוצות בחלוקה מושלמת יהיה בהפרש של 1.

חלוקת מספרים היא כנראה הבעיה ב-NP-complete הכי פשוטה לתיאור,

יישום אחד הוא ה-multi-processor scheduling (מתזמן של נספר מעבדים/ליבות) מאת Garey and Johnson, 1979, בהינתן קבוצה של משימות (לכל משימה מקושר גם זמן השלמה – הזמן שלוקח למשימה להשלים), וגם 2 או יותר מעבדים/ליבות, נקצה כל משימה למעבד (שיבצע אותה) כך שכל המשימות יבוצעו כמה שיותר מהר.

יישום נוסף של חלוקת מספרים הוא voting-manipulation (מניפולציה של הצבעה) מאת Walsh 2009, קיימים 3 פונקציות טבעיות עבור חלוקת מספרים:

1. למזער את הקבוצה עם הסכום הגדול ביותר.
2. למקסם את הקבוצה עם הסכום הכי קטן.
3. למזער את ההפרש בין הקבוצה עם הסכום הכי קטן לקבוצה עם הסכום הכי גדול.

עבור חלוקת מספרים של שני קבוצות, שלושת השיטות הנ"ל זהות מבחינת יישום (שוות ערך), אבל עבור חלוקה של מספר קבוצות (יותר מ-2) אז שיטות אלה לא שוות ערך, בחרנו את המזעור של הקבוצה עם הסכום הכי גבוה אשר מתאים למזעור הזמן הכולל במתזמן.

שימוש בסט של אלגוריתמים יכול למקסם את הקבוצה עם הסכום המינימלי, שזו המטרה של היישום של המניפולציה של הצבעה, למזער את הקבוצה עם הסכום המקסימלי יכול לאפשר לאלגוריתם חלוקת המספרים שלנו להיות כמו אלגוריתם אריזה, באלגוריתם אריזה כל קבוצת מספרים מקבלת קופסה כאשר כל קבוצה אינה מתאימה לגודל הקבוצה שלה, המטרה היא להתאים כל קבוצה בקופסה שלה בצורה מקסימלית כך שנשתמש כמה שפחות בקופסאות.

בפועל, קירובים היוריסטיים לאריזות קופסה כמו הורדה בהתאמה הטובה ביותר (best-fit decreasing) משתמשים רק בכמה פחים יותר מאשר הגבול התחתון (lower bound) כמו הסכום של כל המספרים המחולקים על ידי קיבולת קופסה,