מסמך איפון

רקע:

בעיית ליקוט מוצרים הנה בעיה של איסוף מוצרים לשילוח, כאשר המטרה הינה מזעור סה״כ זמן ליקוט של מספר הזמנות n, אנחנו מניחים שיש פונקציות שמתארת את סה״כ זמן הליקוט אשר תהיה תלויה בכמות המוצרים ללקט ובכמות העצירות שהעובד יצטרך לעשות. ברור להניח שפונקציה זו הינה חיובית עולה.

מתברר לפי (כתבה 1, עמוד 3, וכתבה 3 עמוד 1) שזו בעיה שכרגע נחשבת בעיית np,

עכשיו לבעיה זו יש כמה תתי בעיות קיימים, הבעיות שאנחנו ננסה לפתור יתחשבו בכמה מצבים.

מלקטים בהזמנה: המוצרים של כל הזמנה חייבים להיות תחת אותו מלקט (זה אומר שלא יכול שאדם 1 ילקט חלק מהזמנה x ואדם 2 ילקט חלק אחר מאותה הזמנה x) , דרישה זו הינה סטנדרטית בכל המאמרים, ובבסיסי הנתונים אשר נמצאים להשוואה.

המיקומים של כל המוצרים ידועים לפני.

כיום לבעיה זו היו כמה דרכים אשר מנסים למצוא פתרון ״טוב״ (הכוונה לפתרון אשר קרוב לפתרון למקרה הטוב ביותר), הדרכים הנפוצות שנמצאו הינם:

אלגוריתמים כגון

k-means

metropolis

seed algorithm

וגם אלגוריתמים מותאמים לחברות ספציפיות (מאמר 2, עמוד 4 CCA

תכנות לינארי

תכנות דינאמי .

עכשיו לכל אחד מהמקרים האלה היו חסרונות, (כמובן כי אף אחד לא הגיע לפתרון האידאלי).

המקורות שלנו לבדיקות הולכים להיות כמה מסדי נתונים אשר נותנים מידע על בעיות

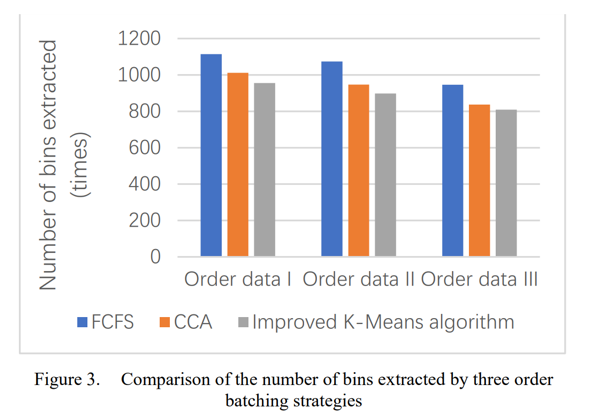
OBP = Order batching problem

בסיס נתונים זה הוזכר בכתבה 3 עמוד 6,

<https://github.com/johanoxenstierna/OBP_instances>

בסיס נתונים זה יאפשר לנו לבדוק את הפתרונות שלנו לעומת הפתרונות האידאלים.

במחקר 2, טוענים אשר גירסא משופרת של k-means מביא לתוצאה הטובה ביותר, (כתבה 2, עמוד 5)



Article 1

<https://www.sciencedirect.com/science/article/pii/S0305054812000020>

article 2