מימוש תכנון דינאמי עבור בעיית n-res

בשלב זה עסקנו במימוש של תכנון דינאמי עבור הבעיה של מכונה בודדת עם מטלות ממושקלות שבמהלך העבודה צריך שתהיה הפסקה ועבודה שהתחילה לפני ההפסקה לא יכולה להמשיך לאחריה אלא צריכה להתחיל מחדש, כפי שתכנו קלרר וסטרוסביץ'[[1]](#endnote-1).

# מבוא

במאמר, עסקו קלרר וסטרוביץ' בבעיות תרמיל NP קשות בתזמון תהליכים ובמציאת פתרונות מקורבים מצורת FPTAS.

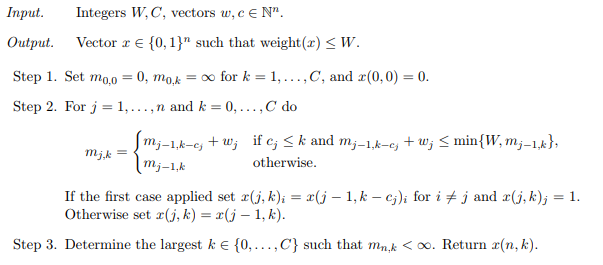
בעיות NP קשות הן קבוצה של אלפי בעיות שקל מבחינה חישובית לבדוק אם פתרון מוצע אכן מהווה פתרון, אך כיום לא ידועה דרך חישובית יעילה למצוא פתרון כזה ומציאת פתרון כזה תוביל למציאת פתרון לכל בעיה שניתן לבדוק פתרון שלה בדרך יעילה[[2]](#footnote-1).

מאחר שלא ידועה דרך יעילה למצוא פתרון לבעיות כאלו, משתמשים באלגוריתמי קירוב, סוג אלגוריתמי הקירוב שנחשב הכי יעיל שאפשר הוא FPTAS, אלגוריתם קירוב כזה נותן עבור כל ε שנבחר, קירוב של S\*(1+ ε) בסיבוכיות של פולינומית עם המשתנים n, 1/ ε.[[3]](#footnote-2) ישנם בעיות NP קשות שהוכח שלא ניתן למצוא להם קירוב בצורה זו.[[4]](#footnote-3)

בעיית התרמיל היא אחת הבעיות שנתן להם פתרון בצורת FPTAS. [[5]](#endnote-2) בעיית התרמיל היא בעיה של השמת חפצים בעלי סכום מקסימלי של ערכים בתרמיל עם נפח מוגבל, ומנוסחת כך בצורה פורמלית:

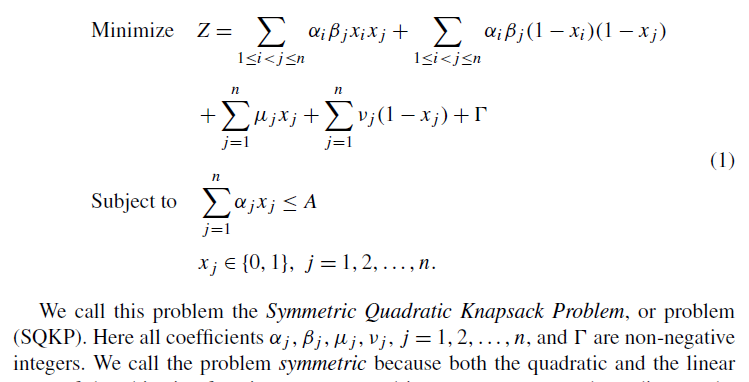
בהנתן ווקטורים (c מייצג ערך, w מייצג גודל) וW (גודל התרמיל) מקסם את תחת התנאי

הבעיה היא NP קשה אך קיים לה אלגוריתם תכנון דינאמי פסאודו פולינומי ידוע כדלהלן[[6]](#footnote-4):



# התכנון הדינאמי

במאמר הוצגה הבעיה שלהלן:



והובאו שני דרכים איך לפתור אותו בעזרת תכנון דינאמי, PDP,DDP

## PDP

הפתרון המוצע ממש דומה לתכנות הדינמי שהוצג לעיל עם שינויים מזעריים, קודם כל, המטרה היא למצוא מינימום , כמו כן פונקציית הערך שונה: צריך ליצור כדלעיל טבלה בגודל A\*n ולהחליף את הפונקציה לחישוב השווי של הוספת החפץ ה k+1 לתרמיל ,אם השווי שהוכנס עד כה הוא , המשקל של החפצים שעברנו עליהם הוא והמשקל של מה שהוכנס מתוכם הוא להיות:

ואת הפונקציה לחישוב המקרה שלא הוספנו (פונקציה שלא הייתה קיימת קודם, אבל כעת גם אפשרות זו מוסיפה לערך) להיות

.

ואז לכל תא בטבלה צריך לראות האם עדיף להגיע אליו עם להוסיף ערך או בלי.

המאמר הראה שאלו אכן הפונקציות הנדרשות כדי לחשב את הערך החדש

## DDP

הפתרון דומה לקודם אך שונה, כמו כן נראה שחסר חלק מהאלגוריתם במאמר (הוא לא טרח להציג אותו לגמרי אלא דילג על חלקים בטענה שזה כמו הקודם, אך נראה שיש כאן חלק קצת מורכב).

במקום שכל עמודה מהA עמודות תייצג איזה משקל של חפצים החלטנו להכניס פנימה, היא תייצג

את המשקל של החפצים שבחרנו לא להכניס פנימה עד כה שיקרא  *שהוא שווה ל*

*כעת במקום שכל שורה מייצגת כמה משקל בחרנו להכניס עד כה, היא מייצגת כמה משקל בחרנו שלא להכניס עד כה.*

*הפונקציות של בחירה האם להכניס או לא להכניס קלות לחישוב כי כל לעיל מוחלף ב ולכן הן:*

*האלגוריתמים שקולים מבחינת התוצאה.*

*במאמר לא צוינו עוד הבדלים ונטען שם שהסיבוכיות של שני האלגוריתמים הוא . באלגוריתם הראשון קל לראות אכן שמדובר בטבלה בת A\*n עמודות שעוברים עליה פעם אחת, אך בשני נראה שאם השורות מייצגות את המשקל שבחרנו שלא להכניס עד כה אזי יכולות להיות הרבה יותר מA שורות והסיבוכיות יכולה להיות גרועה. לכן ננסה לפתור בעיה זו.*

*עבור כל טור שמייצג שהחלטנו לגבי k חפצים עד כה ייתכן שהחלטנו לא להכניס חפצים כלל (כי יכול להיות שנחליט אפילו לא להכניס חפצים כלל) ולכן מקסימום משקל החפצים שהחלטנו לא להכניס יכול להגיע עד , מצד שני, מאחר שאסור להכניס יותר ממשקל כולל של A אזי לא יכול להיות שהחלטנו לא להכניס פחות מ ולכן תחום המשקלים שחוקי לא להכניס הוא בגודל A+1. לכן הטור המינימלי ייצג שבחרנו לא להכניס* משקל של ווהמקסימלי ייצג שבחרנו שלא להכניס משקל של , שזה בעצם כל מה שהחלטנו לגביו עד כה.

## FPTAS

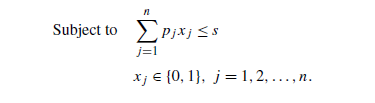
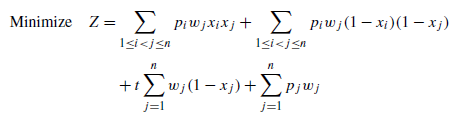
השלב הבא של המאמר הוא להראות אלגוריתם שמקיים את תנאי FPTS בהנחה שקיים אלגוריתם שנותן חסם עליון במכפלה של קבוע על התוצאה המיטבית.

בגדול האלגוריתם דומה לאלגוריתם המקביל בבעיית התרמיל ששם מקצרים את טווח המשקלים ביחס מסויים ואת הגודל המקסימלי באותו יחס ומעגל ואז שולחים לאלגוריתם הדינאמי. אך הוא הרבה יותר מורכב ולא התבקשנו לדון בו

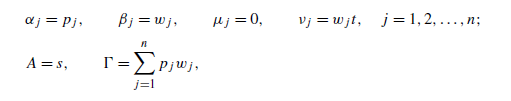
# הבעיה שלנו

הבעיה בה המאמר דן בפרק 3 היא בעיה בה יש מכונה אחת וכמות של משימות, לכל אחת מהן ידוע הזמן שהיא לוקחת - . ולכל אחד יש חשיבות - , והזמן בו מסתיימת כל משימה הוא , כמו כן יש אינטרוול זמן [s,t] בו המכונה מושבתת מעבודה. קיימת בעיה בה עבודה שהופסקה באמצע (crossover) ממשיכה לאחר מכן מהמקום שבו הייתה, בעיה זו נקראת res. הבעיה שלנו היא במקרה שבו עבודה שהופסקה באמצע צריכה להתחיל מההתחלה לאחר מכן. המטרה היא להביא למינימום את .

המאמר מוכיח שהבעיה שקולה לבעיה:



בעיה זו איזומורפית לבעיה הקודמת שהוצגה בצורה זו:



ולכן ניתן **וזה מה שנדרשנו** לממש את התכנון הדינאמי עבור בעיה זו.

לאחר מכן המאמר מביא אלגוריתם שמביא חסם עליון במכפלה של 4 על התוצאה המיטבית ובכך סולל את הדרך לאלגוריתם קירוב מצורת FTAS כמו שהבאנו קודם.

1. Kellerer, H., & Strusevich, V. A. (2010). Fully polynomial approximation schemes for a symmetric quadratic knapsack problem and its scheduling applications. *Algorithmica*, *57*(4), 769-795.‏ [↑](#endnote-ref-1)
2. מדובר באחת השאלות הקשות והנחקרות במדעי המחשב, פורמלית: אם יש פתרון אזי P=NP ואם אין אזי P≠NP [↑](#footnote-ref-1)
3. למשל [↑](#footnote-ref-2)
4. בתנאי ש P≠NP [↑](#footnote-ref-3)
5. Alexander Souza, Combinatorial Algorithms Lecture Notes, Winter Term 10/11 Humboldt University Berlin [↑](#endnote-ref-2)
6. הגירסא היותר מקובלת היא תכנון דינאמי שהטבלה היא W\*n אבל בשביל השלב הבא עובדים עם C\*n כמו שיתואר [↑](#footnote-ref-4)