17. לנוכח בעיות NP-שלמות

בפרקים הקודמים, הודגמה טכניקה לזיהוי בעיות חישוביות קשות. טכניקה זו מתמצה ביכולת להוכיח כי בעיה חישובית נתונה הינה **NP**-שלמה. החשיבות המעשית של זיהוי בעיות חישוביות קשות הינה רבה. זיהוי בעיה חישובית כקשה, מפסיק את החיפוש העקר אחר אלגוריתמים יעילים הפותרים אותה. אולם, הפסקת החיפוש העקר אינה מספקת את איש המעשה, משום שהבעיה החישובית מקורה בבעיה ישומית אמיתית שעליו לפתור (שתקרא מעתה **הישום**). בנקודה זו, צריך להדגיש כי (ברוב המקרים) הבעיה המופשטת, שאיש המעשה מנסה למצוא לה פתרון, אינה זהה עם הישום, אלא מהווה ניסוח מופשט שלו. כעת, כאשר נתגלה כי הבעיה המופשטת הינה קשה, מוצע לאיש המעשה לבחון מחדש את הישום, מתוך כוונה למצוא ניסוח מופשט מספק, אשר כן ניתן להציג לו פתרון יעיל. הניסוח המחודש של הבעיה החישובית המופשטת, יכול להתבסס על האבחנה כי למעשה די לפתור את הבעיה החישובית על תת-קבוצה של הקלטים האפשריים שנבחנו קודם לכן. תיקוותנו היא שלמרות שהבעיה הינה קשה, לגבי הקלטים האפשריים שנבחנו קודם לכן, הרי שהיא אינה קשה לקבוצה המצומצמת של קלטים אפשריים המתאימים לישום שמעניין אותנו. כמובן שצמצום אוסף הקלטים האפשריים, אף פעם אינו הופך את הבעיה לקשה יותר. דוגמאות יובאו בסעיף 17.1.

הניסוח המחודש של הבעיה החישובית, יכול להתבסס על האבחנה כי למעשה ניתן להרחיב את קבוצת הפלטים הקבילים לקלט נתון או לצמצמה. כפי שנראה בסעיף 17.2, ניסוח מחודש שכזה עשוי לעשות את הבעיה לקלה יותר לחישוב, אך גם עלול להפוך אותה לקשה יותר. יש להזהר במיוחד שלא לבלבל בין צמצום והרחבה של קבוצת הקלטים האפשריים לבעיה (שהשפעתו ברורה תמיד - הצמצום יכול רק להקל וההרחבה יכולה רק להקשות) לבין צמצום והרחבה של קבוצת הפלטים הקבילים לכל קלט (דבר שעשוי במקרים מסוימים להפוך את הבעיה לקלה יותר, ועלול במקרים אחרים להפכה לקשה יותר). אם גם החיפוש אחר ניסוחים מספקים, אשר מובילים לבעיות שניתן לפתרן ביעילות, נכשל, לא ניתן אלא לנסות ולהציג אלגוריתמים שסיבוכיותם נמוכה ככל האפשר (אפילו לא יהיו אלו אלגוריתמים יעילים). טכניקות לפיתוח אלגוריתמים כאלו, הכוללות את שיטת **פצל וחסום** (branch and bound) ו**תכנות דינמי** (dynamic programming) לא יסקרו בקורס זה.

בהמשך פרק זה, נתרכז בדוגמאות להשפעה של צמצום קבוצת הקלטים האפשריים או הרחבתה, ושל צמצום קבוצת הפלטים הקבילים או הרחבתה, על סיבוכיות בעיות חישוביות. נראה בניתוח סיבוכיות ממוצעת מעין הכללה של גישת צמצום הקלט, ובפתרונות מקורבים נראה מקרה פרטי של גישת הרחבת הפלט הקביל.

17.1 צמצום קבוצת הקלטים האפשריים

הבעיה, VC_d (ראה הגדרה 16.4) לכל שלם d, הינה צמצום של הבעיה VC. הקלט בבעיה VC_d, מוגבל לגרפים בעלי דרגה מקסימלית d, בעוד שבבעיה VC, הקלט הוא גרף כלשהו. השאלה בשתי הבעיות היא זהה, "האם יש כיסוי

של קשתות הגרף הנתון, ע"י מספר נתון של צמתים?". לכל $d \geq 1$, הבעיה VC_d היא לפיכך צמצום הקלטים הקבילים לבעיה VC. בפרק 16, ראינו כי צמצום זה אינו הופך את הבעיה לקלה יותר, כאשר $d \geq 3$. במילים אחרות, ראינו כי לכל $d \geq 3$, הבעיה VC_d היא **NP**-שלמה (ולפיכך פתירה בזמן פולינומיאלי אם VC פתירה בזמן פולינומיאלי). לעומת זאת, קל לראות כי הבעיה VC_2 קלה לפתרון. נזכיר כי בבעיה VC_2, הקלט הוא גרף שדרגתו המקסימילית היא 2 ומספר שלם K, ויש לענות על השאלה, "האם קיים כיסוי של קשתות הגרף ע"י K צמתים לכל היותר?".

<u>תרגיל 17.1:</u> הראה כי הבעיה VC_2 פתירה בזמן פולינומיאלי (רמז: הגרף שבקלט הינו אוסף של מעגלים, מסלולים וצמתים מבודדים).

תוצאה כללית יותר (ראה תרגיל 17.2) היא שהבעיה $2XC$ ניתנת לפתרון בזמן פולינומיאלי. קראנו לתוצאה זו כללית יותר, משום ש- $VC_2 \propto_p SC_2 \propto_p 2XC$ (הוכח ע"י התאמת ההוכחות המקבילות מפרק 16) ולפיכך קיום אלגוריתם פולינומיאלי לפתרון הבעיה $2XC$, גורר קיום אלגוריתם כזה עבור הבעיה VC_2 (זוהי הוכחה אלטרנטיבית ומסורבלת יותר של תרגיל 17.1).

<u>תרגיל 17.2:</u> הבעיה $2XC$ פתירה בזמן פולינומיאלי.

<u>סקיצת פתרון:</u> הקלט לבעיה $2XC$, הינו קבוצה של תת-קבוצות בגודל 2 (כלומר, זוגות לא סדורים) של קבוצה נתונה. השאלה היא "האם קיים כיסוי מדויק של הקבוצה, ע"י תת-אוסף של הזוגות הנתונים. ניתן לזהות את הקלט לבעיה עם גרף, באופן שצמתי הגרף הינם הקבוצה הנתונה וקשתות הגרף הינם אוסף הזוגות הנתונים. כיסוי מדויק של הקבוצה הנתונה, ע"י תת-אוסף של הזוגות הנתונים, מתאים **לשידוך מושלם** (perfect matching) בגרף (**שידוך** בגרף, הינו אוסף של קשתות זרות בצמתים, **ושידוך מושלם** הינו **שידוך** אשר מכיל את כל צמתי הגרף). ידועים אלגוריתמים פולינומיאלים למציאת שידוך מקסימום בגרף.

צמצום אחר של הקלט לבעיה VC, הגורם לה להיות קלה, הינו צמצום הקלט לגרפים דו-צדדים (bipartite).

<u>תרגיל 17.3</u> (למתקדמים): הראה אלגוריתם פולינומיאלי, אשר בהנתן גרף דו-צדדי, מוצא כיסוי בצמתים, בגודל מינימום, של הקשתות בגרף.

<u>הערה:</u> התרגיל הינו קשה והקושי בו אינו רלוונטי לקורס.

<u>הדרכה:</u> האלגוריתם נובע באופן מיידי מהמשפט הבא בתורת הגרפים: "גודל כיסוי מינימום של גרף דו-צדדי זהה לגודל שידוך מקסימום בגרף". קל לראות, כי הכיסוי לא יכול להיות קטן יותר (כי לא יצליח אפילו לכסות את קשתות השידוך בגרף הנתון). הוכחת הכיוון השני קשה קצת יותר והיכרות עם המושג של **מסלול חילופין** (בגרף

עם שידוך) מועילה בהחלט. בהנתן שידוך מקסימום בגרף הדו-צדדי, יש לבחור לכיסוי צומת אחת מכל קשת של השידוך. במקרה בו השידוך הינו מושלם, ניתן לבחור את כל הצמתים מאותו צד בגרף. המקרה המסובך הינו כאשר השידוך אינו מושלם. במקרה זה, בחירת הצומת מותנת באורך **מסלול חילופין** קצר ביותר, בינו לבין צומת שאינו נמצא בשידוך. השלם רעיון זה להוכחה מלאה.

דוגמא נוספת לצמצום קלט, ההופך בעיה **NP**-שלמה לבעיה הניתנת לפתרון בזמן פולינומיאלי, הינה הצמצום של הבעיה SAT לפסוקים שכל פסוקית בהם מכילה שני ליטראלים לכל היותר, כלומר זוהי הבעיה $2SAT$. זכור כי הבעיה $3SAT$ (כלומר, צמצום SAT לפסוקים בעלי פסוקיות בגודל שלוש לכל היותר) הינה **NP**-שלמה.

<u>תרגיל 17.4:</u> הראה אלגוריתם פולינומיאלי, אשר בהנתן פסוק בצורה CNF המכיל בכל פסוקית שני ליטראלים לכל היותר, מכריע האם קיימת הצבת ערכי אמת אשר מספקת את הפסוק.

<u>הערה:</u> גם תרגיל זה קשה וקושיו אינו רלוונטי לקורס.

<u>סקיצת פתרון:</u> גורם מרכזי בפתרון, הינו הפרוצדורה הבאה, הנקראת force, אשר מקבלת פסוק ϕ ומחזירה פסוק ϕ' (שניהם בצורה CNF ומכילים שני ליטראלים לכל היותר בכל פסוקית) כך שהפסוק ϕ ניתן לסיפוק אסם הפסוק ϕ' ניתן לסיפוק, והפסוק ϕ' אינו מכיל פסוקיות בעלות ליטראל אחד בלבד או פסוקיות אשר שני הליטראלים שלהן מהווים את אותו משתנה או משתנה ושלילתו. לעיתים הפרוצדורה מחזירה ערך אמת (כלומר T או F) במקום פסוק, ובמקרה כזה אנו מתייחסים אל T כאל "ניתן לסיפוק", ואל F כאל "לא ניתן לסיפוק".

<u>הפרוצדורה force</u>

<u>קלט:</u> פסוק ϕ בצורה CNF המכיל בכל פסוקית שני ליטראלים לכל היותר.

<u>האלגוריתם:</u>

(1) השמט את כל הפסוקיות המכילות משתנה ושלילתו (כלומר, פסוקית מהצורה $x_i \vee \overline{x_i}$) והחלף את הפסוקיות המכילות שתי הופעות של אותו ליטראל, בפסוקיות המכילות הופעה יחידה שלו (כלומר, החלף את הפסוקית $x_i \vee x_i$ בפסוקית x_i). סמן את הפסוק המתקבל ב-ϕ' (יתכן $\phi' = \phi$).

(2) אם ϕ' ריק, עצור והחזר T.

(3) אם אין ב-ϕ' פסוקית בעלת ליטראל יחיד (כלומר, כל הפסוקיות ב-ϕ' מכילות שני ליטראלים) עצור והחזר את ϕ'.

(4) אם ϕ' מכילה פסוקיות בעלת ליטראל בודד x_i וכן פסוקית של הליטראל הבודד $\overline{x_i}$, עצור והחזר F (לא ניתן לספק את ϕ').

(5) אם ϕ' מכילה פסוקית בעלת ליטראל בודד, x_i, השמט אותה ואת כל הפסוקיות בהן מופיע הליטראל x_i, והשמט מכל הפסוקיות את הופעת הליטראל $\overline{x_i}$ (כלומר, החלף פסוקית $x_j \vee \overline{x_i}$ בפסוקית x_j) וחזור ל-(2).

(6) באופן סימטרי, טפל בפסוקית המכילה ליטראל בודד $\overline{x_i}$.

הוכח כי הפרוצדורה לחישוב force, ניתנת למימוש בזמן פולינומיאלי, וכן כי הפלט ϕ' המוחזר על ידה, ניתן לסיפוק אמם הקלט ϕ ניתן לסיפוק. שים לב כי יתכן מאד שהפרוצדורה תחזיר פלט ϕ' הזהה לפלט ϕ, אך זה לא יקרה אם יש ב-ϕ פסוקית בעלת ליטראל בודד. תכונות אלו, אינן משתמשות כלל בעובדה כי הקלט מכיל רק פסוקיות בגודל 2 לכל היותר. התכונה הבאה של הפרוצדורה force משתמשת בעובדה זו באופן קריטי: "הפסוקיות המופיעות בפלט (ϕ') מופיעות גם בקלט (ϕ)". אותן פסוקיות אשר שונו במהלך הפרוצדורה force, הושמטו בסופו של דבר. שים לב, כי כל שינוי של הפרוצדורה force בפסוקית כלשהי, מוריד את מספר הליטראלים בפסוקית, וכן כי הפלט, אינו מכיל פסוקיות בעלות ליטראל בודד (פסוקיות כאלו מושמטות במהלך force). מכיוון שהקלט ϕ, מכיל רק פסוקיות בגודל 2 לכל היותר, שינוי של פסוקית במהלך force, גורם להשמטת הפסוקית מהפלט. שים לב כי אילו הכיל הקלט פסוקיות בגודל 3, הפרוצדורה force עלולה היתה להחזיר פסוקיות שלא היו בקלט (אלו פסוקיות המתקבלות מהשמטת ליטראל בודד מפסוקית). הרחבנו את הדיון בתכונה זו, משום שהיא מרכזית להוכחת הנכונות של האלגוריתם שיוצג כעת.

<u>קלט:</u> פסוק ϕ בצורה CNF, המכיל שני ליטראלים לכל היותר בכל פסוקית.

<u>האלגוריתם:</u>

(1) $\phi_1 \leftarrow \text{force}(\phi)$.

(2) אם ϕ_1 הינו ערך אמת, עצור והחזר ערך זה כפלט (זו התשובה לשאלה, "האם ϕ ניתן לסיפוק?").

(3) הצב $\phi_2 \leftarrow \phi_1$. בחר משתנה x_i כלשהו, המופיע בנוסחא ϕ_1.

(3.1) (נסיון להציב ל-x_i ערך T) השמט מהפסוק ϕ_1 פסוקיות המכילות את הליטראל x_i, השמט את הליטראל $\overline{x_i}$ משאר הפסוקיות, הפעל את הפרוצדורה force על הפסוק שהתקבל והצב את התוצאה ל-ϕ_3. אם ϕ_3 זהה לערך F, בצע את (3.2), אחרת בצע $\phi_1 \leftarrow \phi_3$ ולך ל-(2).

(3.2) (נסיון להציב ל-x_i ערך F) השמט מהפסוק ϕ_2 פסוקיות המכילות את הליטראל $\overline{x_i}$, השמט את x_i משאר הפסוקיות, הפעל את force על הפסוק שהתקבל ואת התוצאה שהתקבלה הצב ל-ϕ_3. בצע $\phi_1 \leftarrow \phi_3$ ולך ל-(2).

הוכח כי אלגוריתם זה הינו פולינומיאלי וכן כי הוא מחזיר תמיד תשובה נכונה. שים לב, שקל לראות כי אם הקלט ϕ אינו ניתן לסיפוק, אז האלגוריתם עונה "לא". על מנת לודא כי האלגוריתם נכון גם במקרה שהקלט ϕ ניתן לסיפוק, יש לשים לב כי הבחירה השרירותית לנסות להציב למשתנה ערך T תחילה אינה מזיקה. כלומר, יש להראות כי אם

בשלב מסוים ϕ_1 ניתן לסיפוק ובצעד (3.1) לא נתקבל ערך F, אז גם ϕ_3 המתאים ניתן לסיפוק. טענה זו נובעת ע"י שימוש בעובדה (שצוינה לעי"ל) כי ϕ_3 המתקבלת (מהפרוצדורה force) מכילה רק פסקאות המופיעות בפסוק ϕ_1 ולפיכך אם ϕ_3 (שאינה זהותית F) אינה ניתנת לסיפוק, בהכרח גם ϕ_1 אינה ניתנת לסיפוק. ודא כי הינך מבין את הגרירה האחרונה. באופן מופשט, ניתן לתאר את האלגוריתם כדלקמן: האלגוריתם פועל באיטרציות. בכל איטרציה, האלגוריתם בוחר משתנה ומנסה להציב לו ערך T. אם לא מתקבלת סתירה מיידית (ע"י ביצוע force) האלגוריתם ממשיך עם הפסוק המפושט (ע"י force) ומובטח לו שגם פסוק זה ניתן לסיפוק (בהנחה שהפסוק המקורי ניתן לסיפוק). אם התקבלה סתירה מיידית (ע"י ביצוע force) האלגוריתם מנסה להציב לאותו משתנה ערך F. הצבה שניה זו, חייבת להצליח אפס הפסוק המקורי ניתן לסיפוק.

ראינו דוגמאות של בעיות **NP**-שלמות ודוגמאות של צמצום קלט עבורן הבעיות המצומצמות ניתנות לפתרון בזמן פולינומיאלי. נושא הקשור לצמצום הקלט (אם כי קשר זה לא יהיה ברור בתחילה) הינו הנושא של סיבוכיות ממוצעת של אלגוריתמים. הממוצע נלקח מעל כל הקלטים החוקיים באורך מסוים. יתכן כי למרות שבעיה מסוימת הינה **NP**-שלמה, הרי קיים אלגוריתם הפותר אותה באופן שזמן הריצה <u>הממוצע</u> שלו, על קלטים באורך נתון, הינו פולינומיאלי. פרוש הדבר הוא שהאלגוריתם רץ כנראה זמן רב על מספר מועט של קלטים "קשים", אך על רוב הקלטים הוא רץ זמן סביר (כלומר, זמן פולינומיאלי).

<u>הגדרה 17.1:</u> לבעיה חישובית יש **סיבוכיות ממוצעת אחידה** פולינומיאלית, אם קיים פולינום, P, וקיימת מכונת טיורינג, M, הפותרת את הבעיה כך שזמן הריצה של M (המסומן t_M) מקיים עבור כל n גדול דיו,

$$\left[\frac{1}{2^n}\sum_{x\in\{0,1\}^n} t_M(x)\right] \le P(n)$$

שים לב, כי לעובדה שלבעיה יש סיבוכיות ממוצעת (אחידה) פולינומיאלית, יש משמעות מועילה, רק אם הקלטים האפשריים ליישום המעשי, מופיעים בפיזור אחיד (לפיכך נקראת הגדרה 17.1 **סיבוכיות ממוצעת אחידה**). ההגדרה אינה רלוונטית לישומים בהם פיזור ההסתברות של הקלט שונה מהפיזור האחיד (ואלו כמעט כל היישומים העולים על הדעת. היוצא מן הכלל הסביר היחיד הינו "ישומים קריפטוגרפיים"). הבעיה הניצבת בפני תאוריה של סיבוכיות ממוצעת, היא "מהם אותם פיזורים, שביחס אליהם יש לחשב את זמן הריצה הממוצע (משוקלל)?". נושא זה הינו ודאי מעבר לקורס הנוכחי.

<u>הערה למתקדמים:</u> נציין כי צמצום קלט הינו מקרה פרטי של ניתוח סיבוכיות המתייחס למחלקה של פיזורים אשר מרוכזת רק על חלק מהקלטים האפשריים.

לסיום, נדגיש שוב (את המובן מאליו) כי צמצום של הקלטים האפשריים, אף פעם אינו הופך בעיה חישובית לקשה יותר, בעוד שהרחבת הקלטים האפשריים, אף פעם אינו הופך את הבעיה לקלה יותר. כל זה, בניגוד למצב ביחס

לצמצום והרחבה של קבוצת הפלטים הקבילים (שידון בסעיף הבא).

17.2 צמצום והרחבה של פתרונות קבילים

כבר ציינו בפרק זה, כי פעולת צמצום והרחבה של קבוצת הפלטים הקבילים לקלט נתון, יכולה לשנות את קושיה של בעיה חישובית באופן "מפתיע". בפרט, הרחבה של פלטים קבילים לקלט נתון, עלולה להפוך את הבעיה לקשה יותר, כאשר לאותו קלט לא היו כלל פלטים קבילים לפני ההרחבה. לדוגמא, הדרישה למצוא, בגרף נתון, צמתים מבודדים או מסלול המילטוני, הינה הרחבה של הפלט הקביל בבעית מציאת צמתים מבודדים (שהיא כמובן "קלה" לפתרון). ההרחבה מקשה (תחת ההנחה **P**≠**NP**) את הבעיה החישובית, לגבי אותם גרפים שאין צמתים מבודדים. קודם לכן, היה על האלגוריתם פשוט להודיע כי אין צמתים מבודדים בגרף הנתון, ואילו כעת עליו למצוא מסלול המילטוני. נראה כעת מספר דוגמאות טבעיות יותר, של צמצום קבוצת הפתרונות הקבילים או הרחבתה, לבעיות **NP**-שלמות.

<u>דוגמא 17.1:</u> עיין בבעיה החישובית הבאה: "בהנתן בפסוק בצורה *CNF*, מצא הצבה של ערכי אמת אשר מספקת לפחות מחצית מן הפסוקיות". שים לב, כי בעיה זו מהוה הרחבה של פתרונות קבילים לבעיה *SAT* (בה נדרש כי ההצבה תספק את כל הפסוקיות). אלגוריתם פשוט ויעיל לבעיה המורחבת, פועל באיטרציות. בכל איטרציה מציבים למשתנה אחד את ערך האמת הגורם ליותר פסוקיות לקבל ערך T. הנוסחא מפושטת בהתאם (פסוקיות המסופקות ע"י הצבה זו מוצאות מהפסוק, ועבור פסוקיות המכילות את שלילת הליטראל שסופק, ליטראל זה מוצא מתוכן) וניגשים לאיטרציה הבאה.

<u>דוגמא 17.2:</u> עיין בבעיה החישובית הבאה: "בהנתן פסוק בצורה *CNF*, מצא הצבה אשר מספקת לפחות את <u>המחצית הראשונה</u> של הפסוקיות". גם בבעיה זו נעשתה הרחבה של קבוצת הפתרונות הקבילים לבעיה *SAT*,אלא שבניגוד לדוגמא 17.1, אין הרחבה זו עושה את הבעיה לקלה יותר (הוכח ע"י רדוקציה מהבעיה *SAT*).

<u>דוגמא 17.3:</u> עיין בבעיה החישובית הבאה: "בהנתן פסוק בצורה *CNF*, מצא הצבה בה כל המשתנים שווי ערך, אשר מספקת את הפסוק". בבעיה זו ישנו צמצום של הפתרונות הקבילים לבעיה *SAT*, אך צמצום זה הופך את הבעיה לקלה ממש (הראה זאת).

<u>דוגמא 17.4:</u> עיין בבעיה החישובית הבאה: "בהנתן פסוק בצורה *CNF*, מצא הצבה המספקת את הפסוק, בה בדיוק למחצית המשתנים יש ערך אמת T". גם בבעיה זו ישנו צמצום של הפתרונות הקבילים לבעיה *SAT*, אך צמצום זה אינו הופך את הבעיה לקלה יותר. הראה כי בעיה זו הינה **NP**-שלמה, ע"י רדוקציה מהבעיה *SAT* אליה (רמז: ברדוקציה, הנוסחא ϕ מועתקת לנוסחה $\phi_1 \wedge \phi_2$, כאשר ϕ_1 זהה ל-ϕ ואילו ϕ_2 הינה נוסחא במשתנים חדשים, אשר בה y_i מופיע במקום $\overline{x_i}$ ואילו $\overline{y_i}$ מופיע במקום x_i).

אם נסכם את שראינו בארבע הדוגמאות האחרונות, הרי שהרחבת קבוצת הפתרונות הקבילים לבעיה חישובית או צמצומה, עשויות להפוך את הבעיה לקלה יותר וכן יכולות שלא לשנות כלל את סיבוכיותה. חשוב לחזור ולציין כי הרחבת קבוצת הפתרונות הקבילים של בעיה חישובית או צמצומה, עלולות גם להפוך אותה לקשה יותר.

דוגמא 17.5: הבעיה הבאה הינה **NP**-קשה: "בהנתן גרף G ושלם K, מצא את השידוך המושלם ה K-י (בסדר לקסיקוגרפי של שידוכים מושלמים)". הוכחת הטענה הינה מעבר לקורס זה (ראה פרק 18). שים לב, כי לא ברור כלל כי הבעיה היא ב-**NP**. כמו כן שים לב כי בהנתן גרף G, ניתן למצוא בזמן פולינומיאלי שידוך מושלם ב-G (במידה ויש שידוך כזה). לפיכך, צמצום הבעיה, הופך אותה כנראה לקשה יותר.

דוגמא נוספת בה צמצום בעיה חישובית הופך אותה כפי הנראה (כלומר תחת ההנחה **P** $\neq$ **NP**) לקשה יותר, מופיעה בתרגיל הבא.

תרגיל 17.5: נתונה הבעיה הבאה:

קלט: גרף $G=(V,E)$, בעל משקלים שלמים על קשתותיו (כלומר, לכל קשת $e \in E$, מתאים מספר שלם הנקרא משקל) ומספר טבעי K.

הכרעה: האם קיים בגרף עץ פורש אשר סכום המשקלים שעל קשתותיו קטן מ-K, אשר דרגת כל צומת בו היא 2 לכל היותר?

הוכח כי בעיה זו הינה **NP**-שלמה, וכן כי ללא ההגבלה על דרגת הצמתים הבעיה נמצאת במחלקה **P**.

הדרכה: שים לב כי ללא ההגבלה על דרגת הצמתים ניתן להכריע את הבעיה ע"י מציאת עץ פורש מינימום בגרף (שכידוע לאלגוריתם זה ישנה סיבוכיות זמן פולינומיאלית), השוואת משקלו של עץ זה למספר K והכרעה בהתאם. בנוכחות ההגבלה על דרגת הצמתים בגרף, מתקבל כי עץ כזה הינו בעצם מסלול המילטוני בגרף (אשר מציאת מסלול כזה היא בעיה **NP**-שלמה). לכן, באופן טבעי, נעשה רדוקציה מבעית מציאת מסלול המילטוני בגרף, לבעיה שלפנינו. הרדוקציה תעתיק את הגרף (ללא המשקולות) $G=(V,E)$, הניתן כקלט לבעית מציאת מסלול המילטוני, לקלט לבעיה שלפנינו ע"י בחירת כל משקולות הגרף ל-1, וכן $K=|V|$ (הגרף עצמו יועתק בשלמותו).

מקרה פרטי של הרחבת קבוצת הפתרונות הקבילים, הוא המושג של **פתרונות מקורבים**. נבהיר את המושג ע"י דוגמא.

דוגמא 17.6: פתרון אופטימלי של כיסוי בצמתים - Optimal Vertex Cover (OVC).

קלט: גרף G.

פתרון אופטימלי: כיסוי של קשתות G, במספר מינימום של צמתים.

פתרון מקורב: כיסוי של קשתות G, במספר צמתים שאינו עולה על כפליים המספר המינימום הנ"ל.

שים לב, כי מציאת כיסוי של גרף, ע"י מספר מינימום של צמתים הינה בעיה **NP**-קשה. זכור כי הבעיה VC הינה להכריע, על הקלט גרף G שלם K, "האם קיים כיסוי של G בגודל K?". נראה כי ניתן למצוא פתרון מקורב כנ"ל בזמן פולינומיאלי.

<u>אלגוריתם קירוב לבעית הכיסוי:</u> בהנתן קלט, גרף G:

(1) מצא שידוך מקסימלי (לאו דוקא מקסימום) בגרף G.

(2) עבור כל קשת, (u,v), המופיעה בשידוך, קח את הצמתים u ו-v לכיסוי המקורב.

<u>הערה</u> (אבחנה בין מקסימום למקסימלי): שידוך מקסימלי, הוא שידוך שלא ניתן להוסיף לו קשת נוספת, כך שאוסף הקשתות שיתקבל הינו שידוך בגרף. שידוך מקסימום, הינו שידוך כך שאין בגרף שידוך גדול יותר. באופן פורמלי: M הינו שידוך מקסימלי, בגרף $G=(V,E)$, אם עבור כל קשת $e \in E \setminus M$ הקבוצה $M \bigcup \{e\}$ אינה שידוך בגרף G. לעומת זאת, M הינו שידוך מקסימום, בגרף $G=(V,E)$ אם לכל שידוך M', בגרף G, מתקיים $|M'| \leq |M|$ (זכור: שידוך M, הינו קבוצת קשתות כך שלכל $e_1,e_2 \in M$ אין לקשתות e_1 ו-e_2 צומת משותף).

נחזור לאלגוריתם הקירוב לבעית הכיסוי. על מנת להשלים את האלגוריתם, נציג אלגוריתם פשוט למציאת שידוך מקסימלי. האלגוריתם הינו אלגוריתם "חמדני" (greedy algorithm). האלגוריתם סורק את קשתות הגרף זו אחר זו (לפי סדר כלשהו) ומוסיף קשת לשידוך אם זה אפשרי.

<u>אלגוריתם למציאת שידוך מקסימלי</u>

בהנתן קלט, גרף G:

(1) אתחל $M \leftarrow \varnothing$.

(2) עבור על כל הקשתות ב-E בסדר כלשהו. עבור כל קשת $e \in E$, אם $M \bigcup \{e\}$ הינו שידוך, בצע $M \leftarrow M \bigcup \{e\}$.

שים לב כי קל לבדוק האם קבוצת קשתות מהווה שידוך.

אלגוריתם ה"קירוב" שלנו ניתן למימוש בזמן פולינומיאלי, נותר להראות שהוא אמנם "מקרב" כהלכה (כלומר, מחזיר כיסוי בגודל כפליים מן האופטימום לכל היותר). נראה תחילה כי הפלט של האלגוריתם מהווה אמנם כיסוי, ולאחר מכן נראה כי גודלו אינו עולה על כפליים האופטימום.

<u>טענה 17.1:</u> יהי M שידוך מקסימלי בגרף $G=(V,E)$, אז הקבוצה $U_M \triangleq \{v : \exists (u,v) \in M\}$, מהווה כיסוי של E. שים לב כי $|U_M| = 2 \cdot |M|$.

הוכחה: נניח בדרך השלילה, כי U_M אינו כיסוי. כלומר, קיימת קשת (u,v), שאינה מכוסה ע"י U_M. במקרה זה $M \cup \{(u,v)\}$ מהווה בהכרח שידוך בגרף G, בסתירה לכך ש-M הינו שידוך מקסימלי. □

טענה 17.2: יהי M שידוך כלשהו בגרף $G=(V,E)$, אז כל כיסוי של G מכיל $|M|$ קשתות לפחות.

הוכחה: כל כיסוי של G, חייב להכיל לפחות צומת אחד מכל קשת $e \in M \subseteq E$. הטענה נובעת מהיות קשתות השידוך זרות בצמתיהן. □

בשילוב טענות 17.1 ו-17.2, מתקבל כי גודל הכיסוי שנמצא, אינו עולה על כפליים האופטימום.

ניסוח כללי לבעית אופטימיזציה (ביחס לבעית חיפוש דו-מקומית): תהי f פונקציה המעתיקה זוגות של מחרוזות אל הממשיים האי-שליליים (כלומר, $f : \{0,1\}^* \times \{0,1\}^* \to \mathrm{R}^+$).

קלט: מחרוזת x.

פלט אופטימלי: מחרוזת y_0 כך שמתקיים, $f(x,y_0) = \min_y\{f(x,y)\}$.

פלט מקורב (ביחס r): מחרוזת y כך שמתקיים $f(x,y) \le r \cdot f(x,y_0)$, כאשר y_0 הוא הפתרון האופטימלי.

הערה: שים לב, כי ניתן לנסח את בעית החיפוש, של יחס דו-מקומי R, כבעית מציאת האופטימום של הפונקציה,

$$f_R(x,y) = \begin{cases} 0 & (x,y) \in R \\ 1 & (x,y) \notin R \end{cases}$$

בבעית הכיסוי האופטימלי, השתמשנו בפונקצית מחיר אחרת.

$$f_{OVC}(G,U) = \begin{cases} |U| & U \text{ כיסוי של } G \\ \infty & \text{אחרת} \end{cases}$$

המשמעות של ∞ הינה מספר גדול וניתן (למען הפורמליות) להחליפו ב-$2^{|V|}$ למשל. הבחירה של פונקצית המחיר תלויה ביישום. במקרים מסוימים (כגון כיסוי אופטימלי בצמתים) ישנם מועמדים טבעיים כגון גודל הפתרון.

נסיים פרק זה בדיון באלגוריתם קירוב לבעיה חישובית קשה יותר מבעית מציאת מסלול המילטוני. הבעיה ידועה בשם **בעית הסוכן הנוסע**.

הגדרה 17.2: בעית הסוכן הנוסע - Traveling Salesman.

קלט: מטריצה סימטרית, $\{d_{i,j}\}_{n \times n}$, של מספרים חיוביים (המייצגים מרחקים בין n ערים).

פלט אופטימלי: מסלול (מעגלי) קצר ביותר, העובר ב-n הערים. כלומר סדרה, $i_1, \ldots, i_n$, המהווה פרמוטציה של השלמים, $1 \cdots n$, כך ש-

$$\sum_{j=1}^{n-1} d_{i_j,i_{j+1}} + d_{i_n,i_1}$$

הינו קטן ביותר (ביחס לכל הפרמוטציות האפשריות).

<u>תרגיל 17.6:</u> הראה אלגוריתם פולינומיאלי, אשר מוצא מסלול מעגלי (לא בהכרח פשוט) שארכו קטן מפעמיים אורך המסלול המעגלי הקצר ביותר בגרף.

<u>הדרכה:</u> מצא עץ פורש מינימום, T, של גרף מלא (Clique) עם המשקולות הנתונים (זכור כי ישנו אלגוריתם פולינומיאלי כזה). השתמש במסלול מעגלי המסייר על כל העץ, כך שאינו חוצה אף קשת פעמיים באותו כיוון. ברור כי אורך המסלול הינו, לכל היותר, כפליים ממשקל העץ. טענת הקירוב נובעת מכך שמסלול מעגלי קצר ביותר בגרף, מכיל ממש עץ פורש.

<u>תרגיל 17.7:</u> הראה כי אם מטריצת המרחקים מקיימת את אי-שיוויון המשולש (ראה להלן) אז ניתן למצוא ביעילות, מסלול מעגלי <u>פשוט</u> שארכו קטן מפעמיים אורך המסלול המעגלי הקצר ביותר בגרף.

<u>אי-שיוויון המשולש:</u> קבוצה $\{d_{i,j}\}$ מקיימת את אי-שיוויון המשולש, אם לכל i,j,k מתקיים, $d_{i,j}+d_{j,k} \geq d_{i,k}$ (כלומר, המרחק בין i ל-k, כפי שהוא מוגדר במטריצה, אינו עולה על המרחק על גבי מסלול כלשהו בין i ל-k).

<u>הדרכה:</u> השתמש במסלול מתרגיל 17.6, והחלף כל קשת, החוזרת לצומת שכבר הופיע במסלול (פרט לצומת האחרון במסלול) בקשת לצומת הבא במסלול שעדיין לא הופיע (אם אין כזה, החלף בקשת לצומת הראשון במסלול).

שים לב, כי בתרגיל האחרון בצענו בו-זמנית, צמצום של הקלטים האפשריים לבעית הסוכן הנוסע והרחבה של הפתרונות הקבילים עבורם (כלומר, קירוב ביחס 2). מציאת מסלול פשוט ביחס קירוב קבוע (ואפילו חסום "באופן סביר" בממדי הבעיה) נשארת קשה אם לא מגבילים את הקלטים האפשריים.

<u>תרגיל 17.8:</u> הראה כי, תחת ההנחה **P** ≠ **NP**, אין אלגוריתם פולינומיאלי אשר מוצא מסלול מעגלי פשוט, אשר ארכו נמצא ביחס קבוע לאורך המסלול (המעגלי) הפשוט הקצר ביותר.

<u>הדרכה:</u> השתמש ברדוקציה מבעית מציאת מעגל המילטוני. הרדוקציה ממפה את גרף הקלט, $G=(V,E)$, למטריצת מרחקים $\{d_{u,v}\}_{n\times n}$, כך שלכל $(u,v)\in E$ $d_{u,v}=1$, אחרת $d_{u,v}=\infty$ (∞ יכול להיות למשל 2^{n^3}). הראה כי אלגוריתם, המוצא מסלול מעגלי פשוט שאורכו אינו עולה על 2^n מן האופטימום, מאפשר למצוא את המעגל ההמילטוני בגרף G (במידה ומעגל כזה קיים).