פרויקט סיכום בקורס מבוא לאופטימיזציה – פתרון בעיית שיבוץ מערכת שעות

אלן ברונשטיין 206228751, נעה עבו 208523514

המחלקה למדעי המחשב, אוניברסיטת בר אילן

# הקדמה

סטודנטים רבים מדי שנה נתקלים בצורך לשבץ את הקורסים שהם מחויבים לקחת וקורסים לבחירתם במערכת שעות סופית ומוגבלת בכמות השעות הקיימות ביום לימודים. בפרויקט זה בחרנו לבחון ולפתור את בעיית שיבוץ הקורסים – תהליך שיבוץ כלל הקורסים שנלקחים באותה השנה ע"י סטודנט כלשהו למערכת שעות שנתית אופטימלית הן מבחינת שעות והן מבחינת איכות הקורסים.

כדי למצוא פתרון אופטימלי לבעיה יש ראשית להגדירה היטב.

## הגדרת הבעיה

כדי להגדיר את הבעיה היטב נחלק את תיאורה לשלושה חלקים:

#### הנחות והגדרות

* ראשית נגדיר הנחות כלליות ובסיסיות:
  + לכל קורס קיימים לפחות סט אחד של קבוצות (כלומר לקורס קיימת רק הרצאה) ולכל היותר שני סטים (הקורס מכיל הרצאה ותרגול), ובכל סט ישמר מספר הקבוצה.
  + משך כל קורס יהיה מינימום שעה ומספר שלם.
  + יום למידה מוגדר כך:
    - בימים א'-ה' בין השעות 08:00-21:00
    - ביום ו' בין השעות 08:00-16:00
    - ביום שבת אין קורסים

כלומר סה"כ ישנן 73 שעות למידה שבועיות.

* + שעות הבוקר יוגדרו בין השעות 08:00-14:00.
  + שעות הערב יוגדרו בין השעות 14:00-21:00.
  + במקרה של התנגשויות (חפיפת זמנים בין קורסים) נניח את הדברים הבאים:
    - לעולם לא תהיה התנגשות בין שני קורסי חובה –
    - במצב של התנגשות בין קורס חובה לקורס בחירה, העדיפות תמיד תהיה לקורס החובה.
    - במצב של התנגשות בין שני קורסי בחירה, הקורס המומלץ יותר (המספר המייצג את ההמלצה עליו יהיה גבוה יותר) יבחר.  
      אם שני הקורסים מומלצים במידה שווה אז הקורס בעל מספר השעות הגדול יותר יבחר, ואם שני הקורסים מומלצים במידה שווה ובעלי משקל שווה, אחד מהם יבחר רנדומלית.
  + קורסי בחירה לעולם לא יהיו קורסי קדם לקורסים אחרים.
* שנית נגדיר את מבנה כל קורס:
  + מספר הקורס – מספר שלם, חיובי וייחודי לכל קורס.
  + שם הקורס – ישמש לתצוגה בלבד במערכת השעות הסופית.
  + האם הקורס סמסטריאלי או שנתי.
  + האם מדובר בקורס חובה או בחירה – משמש כדי ליצור עדיפות לקורסי החובה על הבחירה. קורסי החובה יהיו מיוצגים על ידי השנה בה לוקחים אותם (א', ב', ג' או ד') וקורסי בחירה יהיו מיוצגים על ידי המחרוזת "בחירה".
  + מספר השעות של אותו הקורס – מספר שלם וחיובי בטווח [1,73] (הטווח המגביל את כמות השעות של כל קורס מוסבר בהנחות הבסיסיות לעיל).
  + מספר הקבוצות – רשימה הבנויה ממספרים עוקבים כך שכל מספר מסמל קבוצת קורס, המכילה לפחות מספר אחד.
  + זמני הקורס – רשימה של זמני הקבוצות השונות של הקורס. כל תיאור זמן כולל את היום שבו מתרחש הקורס, את שעת ההתחלה ואת שעת הסיום.
  + המלצה – מספר שלם וחיובי בטווח [1,10] כך שציון 1 הוא הנמוך ביותר בעוד שציון 10 הוא הגבוה ביותר.
  + רשימת קורסי קדם – רשימה של מספרי קורסים אותם המשתמש היה צריך לסיים בהצלחה לפני שיבוץ קורס זה.
* לבסוף נגדיר את המערכות וקידוד הקורסים:
  + נגדיר קורס כמחרוזת בינארית באורך X:
    - מכיוון שלכל קורס יש לכל היותר 73 קבוצות, נזדקק ל-7 ביטים על מנת לייצג את כל הקבוצות .כלומר, אנו מקבלים כי X=7. לדוגמה:
      * לא נבחרה אף קבוצה – 0000000.
      * נבחרה קבוצה 01 – 0000001.
      * נבחרה קבוצה 02 – 0000010.
      * נבחרה קבוצה 73 – 1001001.

נשים לב! כי המחרוזת החל מ 1001010 עד 1111111 אינן מחרוזות חוקיות .

* + נגדיר מערכת באופן הבא:
    - * מכיוון שיתכנו לכל היותר 73 קורסים, נשרשר את המחרוזות 73 פעמים. סה"כ נקבל כי מערכת בנויה מ511 ביטים – 511 ביטים אלו מייצגים מערכת מסוימת. למשל:
      * 00000000000000.........0000000 - המערכת בה לא נבחרה אף קבוצה.
      * 00000010000001.........0000001 – המערכת בה עבור כל קורס נבחרה קבוצה 01.
  + נגדיר סוגים של מערכות (סה"כ נגדיר 4 סוגים) :
    - מערכת חוקית ⇔ עבור כל קורס חובה בחרנו קבוצה קיימת מהטווח 01 עד 73, ועבור כל קורס בחירה בחרנו קבוצה קיימת מהטווח 00 עד 73.
    - מערכת רעה ⇔ המערכת חוקית, וגם קיימים לפחות 2 קורסים שמתנגשים.
    - מערכת טובה ⇔ המערכת חוקית, וגם כל קורס לא מתנגש עם כל קורס אחר.
    - מערכת אופטימלית ⇔ מערכת טובה שבה אנו ממקסמים את מה שאנחנו מנסים להשיג.

#### מגבלות

את מגבלות הבעיה חילקנו לשתי קטגוריות:

* מגבלות קבועות, כלומר מגבלות שלא תלויות בקלט שהמשתמש מכניס למערכת:
  + לא ניתן לבצע חפיפה בזמני שני קורסים.
  + לא ניתן לבחור קורס מסוים אם המשתמש לא סיים בהצלחה את כל קורסי הקדם של אותו הקורס.
* מגבלות המשתמש, כלומר מגבלות שהמשתמש מכניס למערכת ותלויות בו לחלוטין:
  + ימים ואו שעות בהן הוא לא יכול ללמוד, כלומר לא ניתן לשבץ קורסים בימים ובשעות הללו.

#### פונקציות מטרה

את מטרות הבעיה חילקנו לשתי קטגוריות:

* מטרות קבועות, כלומר מטרות שלא תלויות בקלט שהמשתמש מכניס למערכת:
  + שיבוץ הקורסים בצורה שכמות החלונות תהיה מינימלית.
* מטרות המשתמש, כלומר מטרות שהמשתמש מכניס למערכת ותלויות בו לחלוטין:
  + מקסום שיבוץ הקורסים בשעות הבוקר/ערב/בימים שלמים.

## שלבי הפתרון

#### הקלט

המשתמש יתן למערכת כקלט את השנה בה הוא לומד, קורסי חובה עליהם הוא צריך לחזור (אם יש כאלה), קורסי הבחירה שבחר לאחר שעברו סינון בשלב הראשון, בחירת פונקציית מטרה וימים ואו שעות בהן הוא לא יכול ללמוד.

#### הפלט

המערכת תחזיר מערכת שעות אופטימלית לפי בחירות המשתמש.

נחלק את האלגוריתם לשני שלבים:

#### שלב ראשון – Pre-processing (שלב סינון הקורסים)

לאחר שהמשתמש יזין את הנתונים וההעדפות שלו, המערכת תסנן את הקורסים לפי הקריטריונים הבאים:

* סינון קורסי קדם: עוברים על כל קורסי הבחירה ועל הקורסים שצריך לחזור עליהם – אם נמצא קורס בחירה שאחד מקורסי הקדם שלו הוא קורס שצריך לחזור עליו, נמחק את הקורס מהרשימה.
* סינון העדפות משתמש: אם נמצא קורס בחירה שנמצא ביום ואו שעה שהוגדרו על ידי המשתמש שלא ניתן ללמוד בהם, נמחק את הקורס מהרשימה.
* סינון קורסי בחירה ראשוני: אם קיימים מעל 74 קורסי חובה, נזרוק את המערכת, מכיוון שלא ניתן לסדר מערכת שעות לפי ההנחות וההגדרות שהגדרנו.
* סינון התנגשויות מלאות בקורסי בחירה: במצב בו שני קורסי בחירה מתנגשים בצורה מלאה (כלומר היום, שעת ההתחלה ושעת הסיום שבו מתקיימים הקורסים זהים) וציון ההמלצה שלהם זהה גם הוא, אז נמחק את אחד הקורסים באופן שרירותי.

#### שלב שני – שלב הבחירה

עבור כל פונקציית מטרה (קבועה ולפי בחירת המשתמש) המערכת תחזיר את קבוצת קורסים שתואמת לבחירות המשתמש ומניבה את הציון המקסימלי עבור השילוב של הפעלת פונקציות המטרה.

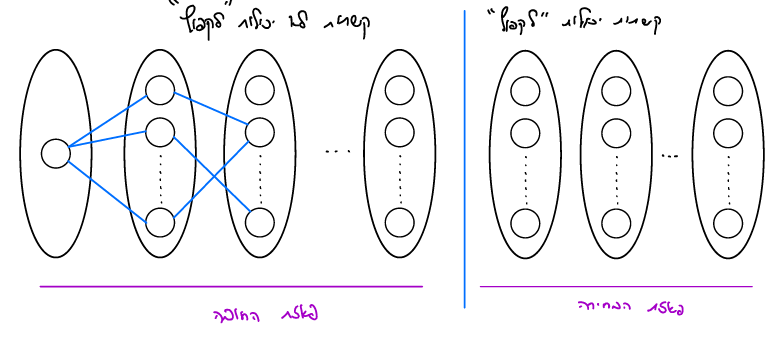
## הדרך למציאת הפתרון

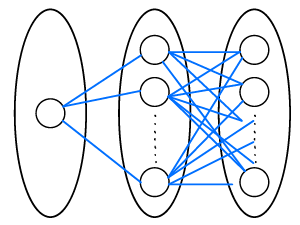
בשלב זה נתאר את התהליך והאלגוריתמים שהגענו אליהם בדרך למציאת האלגוריתם היעיל ביותר לפתרון הבעיה:

#### פתרון נאיבי

נעבור על כל הסידורים האפשריים של כל קבוצות הקורסים ונבחר את האופטימלי ביותר.  
אופן חישוב כמות הסידורים האפשריים שקול לבעיית סידור אובייקטים בשורה, לכן המערכת תעבור על !73 אפשרויות, כלומר זמן הריצה יהיה – סדר גודל גדול ולא אופטימלי.

#### פתרון עם גרף שכבות

נגדיר גרף שכבות, כך שכל שכבה מייצגת סט של קורס, כל קודקוד מייצג קבוצת קורס וקיימת קשת המחברת שני קודקודים אמ"מ הקשת היא בין שכבה ל-, וששתי הקבוצות לא חופפות בזמנים. נשים לב כי מספר השכבות הוא לכל היותר 146 (לפי ההנחה שלכל קורס יש לכל היותר שני סטים, הרצאה ותרגול, כלומר פי שניים ממספר הקורסים האפשריים לשיבוץ) ומספר הקודקודים הוא לכל היותר 73 (כמספר הקורסים האפשריים לשיבוץ). נגדיר שצריך קודקוד התחלה יחיד בגרף השכבות, כלומר יהיו לכל היותר 73 גרפים שונים לאחר פיצול השכבה הראשונה כך שבכל גרף יהיה קודקוד התחלה יחיד. הגרף יהיה מחולק לשני חלקים: החלק הראשון יהיה מורכב מקורסי החובה, כך שבחלק זה קשתות מתחברות רק לשכבות עוקבות (כלומר לא ניתן לחבר קשת בין שכבה ל- כאשר k > 1), והחלק השני יהיה מורכב מקורסי הבחירה, כך שבחלק זה קשתות יכולות "לדלג" לשכבות הבאות (נשים לב שעדיין נשמר סדר מעבר הקשתות – מהנמוך לגבוה):  
האלגוריתם יסכום את משקלי הקשתות (שיכילו בתוכן את שקלול הציונים של פונקציות המטרה) ויחזיר את המסלול בעל המשקל המקסימלי, כלומר המערכת האופטימלית בעלת הציון הגבוה ביותר עבור פונקציות המטרה.

הבעיה עם הפתרון זה הוא שהקשתות הן "דינמיות", כלומר הגרף מניח באופן שגוי שהמסלול שקול למסלול .

בנוסף, אם נרצה לוודא שהקשתות לא יהיו "דינמיות", אז נצטרך ליצור "קליק" בגרף, כלומר ליצור קשת בין כל קודקוד לכל קודקוד בשכבה שאחריו:  
הבעיה עם נסיון שיפור זה הוא שזמן הריצה אפילו גבוה מזה של הנאיבי – .

#### פתרון תכנות דינמי

בניסיון לייעל את הפתרון שאמצעות גרף השכבות, העצנו פתרון המחזיר את המערכת האופטימלית באופן הבא: במקום להריץ את האלגוריתם כל פעם עד השכבה האחרונה, האלגוריתם יזכור את מה שכבר נסרק, הוא יעצור בקודקוד ויבדוק האם חישב את תת העץ הזה בעבר, כלומר באופן תיאורטי הגרף יסרק פחות פעמים בכך שעבור כל קודקוד בשכבה גבוהה נוכל בשלב מסוים למצוא מערכת אופטימלית ב-O(1) (נשים לב שהאלגוריתם לא יזכור אלו קורסים נבחרו, אלא אלו שעות נבחרו).

להלן הפסאודו קוד:

*כל עוד ניתן לבחור:*

*הזכר אם ראית מערכת כזו בעבר:*

*אם כן, המשך לבחור מאותה הנקודה שאתה זוכר.*

*אחרת, בחר באקראי קורס בתנאי שהמערכת חוקית.*

*זכור את המערכת החדשה.*

*אם המערכת טובה יותר מהמערכת הקודמת, שמור אותה.*

הבעיה עם פתרון זה היא שנצטרך לזכור מצבים שונים. אומנם, האלגוריצם יכול לרוץ בצורה טובה על קלטים בגודל "הגיוני" – האלגוריתם מדמה את ההתנהגות האנושית בבחירת הקורסים בכך שהוא זוכר מערכות שנתקלנו בהן בעבר ומנסה לשפר אותן.

#### ~~פתרון חמדני~~

~~[When using headings, don’t skip levels. If you need a heading 3, 4, or 5 with no text following it before the next heading, just add a period at the end of the heading and then start a new paragraph for the subheading and its text.]~~

## הפתרון – אלגוריתם אבולוציוני (גנטי)

#### הסבר השיטה

השיטה מחולקת לחמישה שלבים:

1. **שלב הבריאה:** ניצור כמות מסוימת של מערכות – כל מערכת בשלב זה תהיה חוקית, אך לא בהכרח תהיה אופטימלית או אפילו טובה.
2. **שלב הבחירה (המבול):** בשלב הזה באמצעות פונקציה מסוימת שנגדיר בהמשך, נבחר אלו מהפתרונות יוכל להמשיך לדור הבא, ואלו מהפתרונות ימותו. הבחירה תתבצע כך: נפעיל פונקציה או "אסון טבע" כלשהו, מי שישרוד את אסון הטבע יוכל להמשיך לדור הבא, ומי שלא ימות ולא יזכה להמשיך לדור הבא. בכך יובטח לנו שרק המערכות הכי טובות יזכו להמשיך לדור הבא.
3. **שלב הזיווג:** בשלב זה נבחר את המערכות שנזווג, כתוצאה מהזיווג יוולדו מערכות חדשות (שלב זה הוא רק בחירת הזוגות).
4. **שלב היצירה:** בשלב זה נוצרות המערכות החדשות מלקיחת גנים מההורים שלהם.
5. **שלב המוטציה:** בשלב זה נעבור על כל אחד מהילדים ובהסתברות מסוימת נבצע בו מוטציה. שלב זה פוטנציאלי ליצור לנו פתרונות חדשים ואף טובים יותר שלא יכלנו למצוא משלב ההתאמה.
6. **שלב הלידה:** בשלב זה נוסיף את המערכות החדשות שנוצרו לאוכלוסיה הקיימת.

#### פסאודו קוד

נגדיר את אחרית הימים להיות זמן או מספר דורות שניתן לאלגוריתם לרוץ.

1. בצע את שלב הבריאה.
2. בצע את שלב המבול.
3. בצע את שלב הזיווג.
4. בצע את שלב היצירה.
5. בצע את שלב המוטציה.
6. בצע את שלב הלידה.
7. בצע את שלבים 2 עד 6 עד מציאת הפתרון האופטימלי, או אחרית הימים, מה שמגיעקודם.

#### הסבר האלגוריתם

1. **ביצוע שלב הבריאה:** שלב זה הוא יחסית פשוט – ניתן לאלגוריתם לבחור כמה מערכות הוא יוצר. אם קורה והמערכות נכחדות (הכחדות מוגדרת כאשר נותרה מערכת אחת באוכלוסיה לכל היותר), ובהנחה ואסונות הטבע היו סבירים, אנו ניצור בריאה חדשה עם אוכלוסיה התחלתית גבוהה מהאוכלוסיה הקודמת. חשוב לציין – באלגוריתם שלנו נעדיף לבצע פעם אחת את הבריאה ולהמנע כמה שיותר מהכחדות מלאה או הרוב, אך נשאיר את האופציה לבריאות נוספות במידה וכן הגענו להכחדות הנ"ל.
2. **ביצוע שלב המבול:** בשלב זה המבול מקדם אותנו לפונקציית המטרה. חשוב לציין כי עוצמת המבול תשתנה מדור לדור בצורה דינמית – נשתמש ב"מבול חלש" שיכחיד מעטים, אך נחזק אותו ככל שהאוכלוסיה שלנו מתחזקת, וגם נחליש אותו ככל שהאוכלוסיה מתחלשת בהתאם. כמו כן המבול נועד לשמור על יחס ילודה-תמותה, בנוסף לשיטת ממשל-סין בשלב 4 – אם נראה כי הילודה גבוהה מידי נפעיל מבול חזק יותר, וההפך אם התמותה גבוהה מידי.
3. **ביצוע שלב הזיווג:** בשלב זה נמיין את האוכלוסייה שלנו לפי מהחזק לחלש. הזיווג יתבצע לפי שיטת "הטובים לטיס" או "החזקים לחזקים" – כל אחד יזווג למי שהכי קרוב אליו מבחינת חוזק, בהנחה שכבר לא זווג למישהו אחר.
4. **ביצוע שלב היצירה:** בשלב אנו בוחרים 2 דברים – כמה ילדים יהיה לכל זוג ואלו גנים נקח לאב ולאם. בחירת כמות הילדים תהיה אקראית ותעבוד לפי שיטת ממשל-סין. נרצה כי מדור לדור ישמר יחס ילודה-תמותה מסוים – אם הילודה תהיה גבוהה מידי לא נאפשר לזוג ליצור הרבה ילדים, ואם אחוז התמותה יהיה גבוה נדחוף זוגות ליצור הרבה ילדים. אופן בחירת הגנים יתבצע בצורה אקראית.
5. **ביצוע שלב המוטציה:** את שלב זה נבצע על האוכלוסיה החלשה, שהסבירות להכחדותה גבוהה. בהסתברות מסוימת האלגוריתם ישנה מספר מסוים של גנים, וינסה לתת לאוכלוסיה הזו סיכוי לשרוד.
6. **ביצוע שלב הלידה:** שלב פשוט שבו נוסיף את כל הילדים החדשים לאוכלוסיה הקיימת.

#### הסבר נכונות

נראה כי תמיד נגיע לפתרון האופטימלי אם נריץ מספיק זמן או מספיק דורות – בהסתברות גבוהה נגיע לפתרון אופטימלי או אחד שמאוד קרוב אליו. נשים לב שמדור לדור אנו מקבלים כי ה"אלפה" מתחזק או שווה לעוצמה של האלפה בדור הקודם (מכיוון שבשלב המבול אנחנו מוחקים את החלשים ומשאירים את החזקים, בפרט את האלפה, או הכי חזק נשאר). הדרך היחידה שהאלפה ימחק היא אם המבול מחק את כולם, וזה בסתירה לכך שאנו רוצים כי המבול יהיה "מבול חלש".

#### בדיקת קבועים להנבת מערכת אופטימלית ביותר

* מקדם הגדלת האוכלוסיה במקרה של הכחדות – 2 (נגדיל פי 2 מהבריאה הקודמת).
* ציון הניתן לכל מערכת – 0-100:
  + 100 מתקבל עבור מערכת אופטימלית.
  + 60-99 מתקבל עבור מערכת טובה.
  + 10-59 מתקבל עבור מערכת רעה.
  + 0-9 מתקבל עבור מערכת לא חוקית.
* פיצוץ אוכלוסין רמה 3: פי 3.5 מהאוכלוסיה בבריאה.
* פיצוץ אוכלוסין רמה 2: פי 12.25 מהאוכלוסיה בבריאה.
* פיצוץ אוכלוסין רמה 1: פי 42.875 מהאוכלוסיה בבריאה.
* סכנת הכחדה רמה 3: 0.75 מהאכלוסיה בבריאה.
* סכנת הכחדה רמה 2: 0.5 מהאוכלוסיה בבריאה.
* סכנה הכחדה רמה 1: 0.25 מהאוכלוסיה בבריאה.
* מקדם ילודה מצב רגיל: 1-5.
* מקדם ילודה פיצוץ אוכלוסין רמה 3: 1-3.
* מקדם ילודה פיצוץ אוכלוסין רמה 2 : 1-2.
* מקדם ילודה פיצוץ אוכלוסין רמה 1 : 1.
* מקדם ילודה בסכנת הכחדה רמה 3 : 3-5.
* מקדם ילודה בסכנת הכחדה רמה 2 : 4-5.
* מקדם ילודה בסכנת הכחדה רמה 1 : 5.
* מקדם מבול מצב רגיל : נבחר 50% טובים ביותר.
* מקדם מבול פיצוץ אוכלוסין רמה 3 : נבחר 28% טובים ביותר.
* מקדם מבול פיצוץ אוכלוסין רמה 2 : נבחר 8% טובים ביותר.
* מקדם מבול פיצוץ אוכלוסין רמה 1 : נבחר 3% טובים ביותר.
* מקדם מבול בסכנת הכחדה רמה 3 : נבחר 60% טובים ביותר.
* מקדם מבול בסכנת הכחדה רמה 2 : נבחר 70% טובים ביותר.
* מקדם מבול בסכנת הכחדה רמה 1 : נבחר 80% טובים ביותר.
* קבלת מוטציה עבור מערכת לא חוקית : 100%.
* קבלת מוטציה עבור מערכת רעה : 75%.
* קבלת מוטציה עבור מערכת טובה : 36%.
* לקיחת גנים מההורים ילד 1 : 259 אב , 252 אם.
* לקיחת גנים מההורים ילד 2 : 154 אב , 357 אם.
* לקיחת גנים מההורים ילד 3 : 357 אב , 154 אם.
* לקיחת גנים מההורים ילד 4 : 434 אב , 77 אם.
* לקיחת גנים מההורים ילד 5 : 77 אב , 434 אם.

**ביבליוגרפיה**

https://towardsdatascience.com/genetic-algorithm-implementation-in-python-5ab67bb124a6

https://towardsdatascience.com/using-genetic-algorithms-to-schedule-timetables-27f132c9e280

<https://en.wikipedia.org/wiki/Genetic_algorithm>

<https://github.com/ahmedfgad/GeneticAlgorithmPython>

Tables

Table 1

[Table Title]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Head | Column Head | Column Head | Column Head | Column Head |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |

Note: [Place all tables for your paper in a tables section, following references (and, if applicable, footnotes). Start a new page for each table, include a table number and table title for each, as shown on this page. All explanatory text appears in a table note that follows the table, such as this one. Use the Table/Figure style, available on the Home tab, in the Styles gallery, to get the spacing between table and note. Tables in APA format can use single or 1.5 line spacing. Include a heading for every row and column, even if the content seems obvious. A default table style has been setup for this template that fits APA guidelines. To insert a table, on the Insert tab, click Table.]

Figures title:

Figure 1. [Include all figures in their own section, following references (and footnotes and tables, if applicable). Include a numbered caption for each figure. Use the Table/Figure style for easy spacing between figure and caption.]

For more information about all elements of APA formatting, please consult the APA Style Manual, 6th Edition.