**פרויקט – מבוא לבינה מלאכותית – 67842**

מגישים:

ניר בלוקה (206760811) | רועי מאיר (211442843) | רון בנימין (207837089)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*תוכן עניינים\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* להקטין סיכוי לסמסטר נוסף במצב התחלתי
* לשים לב למשקל הנמוך של ממוצע על הניקוד

1. **מבוא**  
   הבעיה שבחרנו לפתור היא בעיית אופטימיזציה.  
   הבעיה שנבחרה היא מציאת תוכנית לימודים חוקית ואופטימלית לתואר כך שהממוצע של התכנית המוצעת יהיה מקסימלי. זאת על מנת לתת מענה לסטודנטים לתכנון של מסלול לימודים המאפשר להם להשיג את הממוצע הטוב ביותר.  
   כלומר, עבור שנתון של מסלול לימודים מסוים, המכיל את כמות נקודות הזכות הדרושות למסלול, רשימת הקורסים המוצעים (שם, מספר, נ"ז, דרישות קדם, ממוצע, סמסטר), ועבור דרגת עומס לכל סמסטר (מבחינת כמות נ"ז), נרצה למצוא תוכנית לימודים העומדת בכל הדרישות וממקסמת את הממוצע הכללי בתואר.  
     
   מצאנו את בעיה זו מעניינת משתי סיבות עיקריות:

* מדובר בבעיה שאנחנו וכלל הסטודנטים מכירים מקרוב, כולנו משקיעים זמן רב בתכנון תוכנית הלימודים של התואר, וכולנו יודעים כמה זה מורכב לקחת את כלל השיקולים בחשבון כדי לסיים את התואר עם ממוצע טוב ככל הניתן.
* מבחינה חישובית הבעיה היא קשה מאוד – ישנם המון סידורים של תוכניות לימודים חוקיות (מספר אקספוננציאלי במספר הקורסים המוצעים), ומציאת התוכנית שממקסמת את הממוצע עלולה לגרום לבדיקת כלל האפשרויות.  
  הבעיה שאותה אנו מנסים לפתור דומה למספר בעיות קשות.  
  נשים לב שפתרון לבעיה שלנו הוא בפרט בחירה של קורסים שסכום הנ"ז הכולל שלהם שווה בדיוק למספר הנקודות הדרוש במסלול הלימודים. בעיה זו, ללא ההתייחסות למקסום הממוצע או חוקיות התכנית, היא קשה בפני עצמה, שכן היא דומה לבעיית הסכומים החלקיים (Subset Sum Problem) שהיא בעיה NP-Complete.  
  בנוסף, בבואנו למקסם את ממוצע הקורסים הנלקחים אף ללא התחשבות בחוקיות התכנית, פתרון הבעיה דומה לפתרון בעיית התרמיל השלם (0-1 Knapsack), שהיא בעיה NP-Hard. בדומה לבחירתם של קורסים שיניבו את הממוצע המקסימלי, בבעיית התרמיל נבחר פריטים כך שערכם הכולל יניב את הערך המקסימלי.  
  בנוסף, באחד המאמרים שקראנו[[1]](#footnote-1) בנושא, נאמר כי מציאת תת קבוצה אופטימלית של קורסים עם דרישות קדם זו בעיה NP-Hard וקל להיווכח שזוהי תת בעיה של הבעיה שאנו מנסים לפתור ולכן בפרט הבעיה שלנו היא NP-Hard בעצמה.

בחרנו לנסות לפתור את בעיה זו באמצעות שתי שיטות שנלמדו בקורס:

* **חיפוש בגרף –** תכננו למדל את הבעיה כבעיית חיפוש בגרף בה מציאת מסלול קצר ביותר בגרף החיפוש שקול לפתרון הבעיה. כלומר, בעזרת המסלול הקצר ביותר בגרף זה, נוכל לחלץ את תכנית חוקית ואופטימלית למסלול הלימודים.

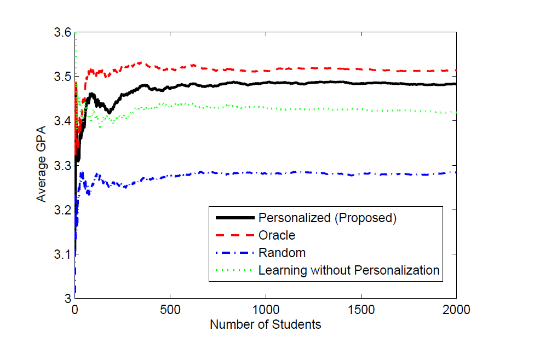
בפרט, תכננו להשתמש באלגוריתם A\* על מנת להשיג פתרון בזמן ריצה מתקבל על הדעת.

* **חיפוש מקומי** - (Local Search) – תכננו למדל את הבעיה כבעיית חיפוש מקומי בה מתחילים מתוכנית מסוימת, ואותה מנסים לשפר בהדרגתיות בעזרת אלגוריתמי חיפוש שונים הנלמדו בקורס.

1. **עבודה קודמת**

לא מצאנו עבודה שמספיק דומה לנושא שבחרנו בפרויקטים משנים קודמות בקורס. עם זאת, כן מצאנו שני מאמרים באינטרנט שמנסים לפתור בעיות דומות לבעיה שלנו:

* המאמר הראשון מתייחס לבניית מערכת מותאמת אישית ומשתנה עבור כל סטודנט, שמשיגה איזון בין סיום מהיר של התואר לבין מקסום הממוצע. זאת תוך התחשבות באילוצים דומים לבעיה שלנו אך בתוספת התייחסות למספר פרמטרים על הסטודנט עצמו לדוגמה ציון ה-SAT שלו.   
  כותבי המאמר השתמשו בשני אלגוריתמים עיקריים כדי לפתור בעיה זו:
  + Forward-Search Backward-Induction Algorithm – אלגוריתם שלא למדנו ומנסה לשבץ קורסים לפי דרישות הקדם שלהם בעזרת תכנון דינאמי.
  + Multi-Armed Bandit Algorithm – אלגוריתם שלא למדנו ומתמקד במציאת המלצות לקורסים למיקסום הממוצע ומזעור הזמן לסיום התואר, בהסתמך על נתונים של סטודנטים קודמים עם פרופיל דומה לסטודנט שמנסה לבנות את המערכת.

במאמר זה, החוקרים השוו את תוצאותיהם אל מול מאגר ציונים של אוניברסיטת UCLA כדי למדוד את איכות התוצאות שקיבלו. בפרט הם מדדו את איכות הפתרון שלהם למול מודלים שונים. הראשון הינו מודל "אורקל" אשר מכיר סטטיסטיקות של ציוני מסלולים על בסיס העבר ועל כן מהווה חסם עליון לפתרון, והשני הוא מודל רנדומלי ועל כן מהווה חסם תחתון לכל פתרון סביר. ניתן לראות שהתוצאות שלהם היו קרובות למודל ה- "אורקל".

ההנחות במאמר הן שקיימים מסלולי תואר בהם קורסים עם דרישות קדם חובה, כאשר כל קורס יכול להילמד בסמסטר אחד או יותר בשנה (אצלם יש רבעונים), בנוסף מניחים קיום מאגר כנ"ל.

* המאמר השני מתייחס לבניית מערכת קורסים חוקית תוך פיזור העומס ורמת הקושי בין הסמסטרים. כותבי המאמר השתמשו בשלושה אלגוריתמים על מנת לפתור את הבעיה:
  + Maximum Prerequisite Weightage (MPW) Algorithm, אשר מעדיף לשבץ בתחילה קורסים שהם קורסי קדם של קורסים רבים תוך התחשבות בנסיון לאזן את הקושי של הקורס.
  + Difficulty Approximation (DA) Algorithm, אשר מתמקד באיזון דרגת העומס והקושי בין הסמסטרים השונים ומכוון למזער פונקציית הפסד (Loss) מעל דרגת הקושי בין סמסטרים.
  + Adaptive Genetic Algorithm (AGA), אלגוריתם עם גישה היוריסטית שבאה לידי ביטוי בפונקציית ה- Fitness, בה נבחרת סדרת קורסים תוך התחשבות לקורסי קדם וקושי, אשר מתפתחת לאורך איטרציות בריצת האלגוריתם.

במאמר זה, השוו החוקרים את הפתרונות של האלגוריתמים השונים וכן של אלגוריתם רנדומלי, והתמקדו במדידת והשוואת פיזור הקושי בין הסמסטרים. בנוסף, השוו בין מספר הנ"ז שנלקחו בכל סמסטר.

ההנחות במאמר הן שישנם מסלולי תואר עם קורסים, לכל קורס ישנם קורסי קדם חובה, וכן שישנו אינדקס של קושי לקורסים וכל קורס מכיל ערך סטטי המייצג את הקושי שלו.

1. Personalized Course Sequence Recommendations - Jie Xu, Member, IEEE, Tianwei Xing, Student Member, IEEE, and Mihaela van der Schaar, Fellow, IEEE [↑](#footnote-ref-1)