אוניברסיטת בר-אילן

פיתוח מודל לניתוב רכבים המאופיין במשכי נסיעה תלויי זמן וסטוכסטיים

אורן נחום

עבודה זו מוגשת כחלק מהדרישות לשם קבלת תואר מוסמך בפקולטה למדעי החברה, החוג המשולב למדעי החברה, ניהול לוגיסטיקה.

רמת-גן תשס"ט

עבודה זו נעשתה בהדרכתם של ד״ר יובל הדס ופרופ׳ קונסטנטין קוגן,
מן החוג המשולב למדעי החברה של אוניברסיטת בר-אילן.

הבעת תודה

ברצוני להודות לד״ר יובל הדס על תרומתו להצלחת עבודת מחקר זו. ד״ר יובל הדס השקיע מזמנו, היה נכון לעזור ולענות לשאלותיי בכל עת, הציע רעיונות ובחן את רעיונותיי ועל כך התודה. לפרופ׳ קונסטנטין קוגן ברצוני להודות על שלקח על עצמו את הנחיית עבודת תזה זו תוך ידיעה שאין היא קשורה לתחום התמחותו, וכן על העזרה במודל מתמטי של הבעיה הנחקרת.

כמו כן, ברצוני להביע תודה לפרופ׳ צבי רז ז״ל, אשר עזר בכתיבת הצעת המחקר לעבודת מחקר זו, ואשר עקב מצבו הבריאותי נבצר ממנו להשתתף בהנחיית עבודה זו.

תקציר

בעיית ניתוב רכבים (Vehicle Routing Problem – VRP) הוא שם כולל לקבוצה של בעיות שבהם יש להקצות קבוצה של מסלולי נסיעה עבור צי של כלי רכב. כלי הרכב יוצאים ממחסן, עוברים דרך לקוחות, מספקים להם סחורה, וחוזרים למחסן. המטרה בבעיית ניתוב הרכבים היא הורדת עלויות ע״י מציאת מסלולים אופטימאליים, בד״כ המסלולים הקצרים ביותר.

בעיית ניתוב הרכבים נוסחה לראשונה ע״י Dantzig ו-Ramser, אשר הציגו אלגוריתם היוריסטי לפתרון הבעיה, שפעל בשני שלבים. בשלב הראשון נבנה פתרון ראשוני, ובשלב השני שופר הפתרון הראשוני ע״י איחוד מסלולים למספר קבוצות. Clarke ו-Wright הציגו את אלגוריתם Savings. גם אלגוריתם זה פועל בשני שלבים, בראשון בונים קבוצת פתרונות ראשוניים, ובשני מאחדים את המסלולים למסלולים חדשים על-סמך החיסכון המתקבל מחיבור שני מסלולים למסלול אחד. Gillett ו-Miller הציעו את אלגוריתם Sweep לפתרון בעיית ניתוב הרכבים. אלגוריתם זה מחלק את קבוצת הלקוחות למספר תת-קבוצות, כאשר את המסלול הקצר ביותר עבור כל תת-קבוצה, אנו מוצאים ע״י פתרון בעיית סוכן נוסע (Traveling Salesman Problem). שיטת האשכולות (Cluster Method) הוצגה ע״י Fisher ו-Jaikumar, גם באלגוריתם זה אנו תחילה מחלקים את קבוצת לקוחות למספר תת-קבוצות, ולאחר מכן אנו מוצאים את המסלול הקצר ביותר עבור כל אחת מתת-הקבוצות. חלוקת קבוצת הלקוחות למספר תת-קבוצות נעשת בעזרת פתרון בעיית השמה כללית (Generalized Assignment Problem – GAP).

בנוסף להיוריסטיקות השונות שפותחו לפתירת בעיית ניתוב הרכבים, הוצגו גם אלגוריתמים המבוססים על מטה-היוריסטיקות, אשר משתמשות בהיוריסטיקות פשוטות ליצירת מרחב פתרונות בסיסי ואשר מבצעות חיפוש עמוק יותר באותו מרחב הפתרונות, תוך שימוש בפעולות כדוגמת איחוד פתרונות, על מנת להשיג פתרון טוב יותר. Bullnheimer הציג אלגוריתם המבוסס על Ant System לפתרון בעיית ניתוב הרכבים. Berger ו-Barkaoui הציגו אף הם אלגוריתם לפתרון בעיית ניתוב הרכבים המבוסס על אלגוריתמים גנטיים. בנוסף הוצגו מספר אלגוריתמים נוספים לפתרון בעיית ניתוב הרכבים המתבססים על מטה-היוריסטיקות נוספות, כגון, חיפוש Tabu, Deterministic Annealing, Simulated Annealing ועוד.

במקביל להתפתחות ההיוריסטיקות והמטה-היוריסטיקות, שמספקות פתרונות קרובים לפתרון האופטימאלי, עסקו מספר חוקרים בפיתוח הרחבות לבעיה הבסיסית של ניתוב הרכבים. המטרה שעמדה מול אותם חוקרים הייתה ליצור מודלים קרוב יותר ליישומים של העולם האמיתי שבו קיימים מספר רב של אילוצים. בין ההרחבות השונות שפותחו ניתן למצוא את בעיית ניתוב הרכבים עם חלוקת הובלות. בבעיה זו, האילוץ שלכל לקוח יכול להגיע רק כלי רכב אחד אינו קיים. Burrows, לדוגמא, הציע אלגוריתם מבוסס על אלגוריתם Savings לפתרון בעיה זו. הרחבה נוספת שפותחה היא בעיית ניתוב רכבים עם אילוצי חלונות זמן. בבעיה זו מבוססת על בעיית ניתוב הרכבים כאשר אליה מתווסף אילוץ נוסף, שלפיו יש לספק ללקוח את הסחורה בחלון זמן מסוים. Solomon היה הראשון שהציג הרחבות של מספר היוריסטיקות לפתרון בעיית ניתוב רכבים, כגון אלגוריתם Savings ואלגוריתם Sweep, כך ישמשו לפתרון בעיית ניתוב רכבים עם חלונות זמן.

יתכנו מצבים שבהם לארגון יש יותר ממחסן אחד, והם מעוניינים לספק סחורה ממספר מחסנים בו זמנית. עבור ארגונים כאלו פותחה בעיית ניתוב הרכבים ממספר מחסנים. Wren ו-Holliday תארו שיטה לפתרון בעיית ניתוב הרכבים ממספר מחסנים המתחלקת לשני שלבים: פיתוח פתרון ראשוני (ע"י אלגוריתם Sweep), שלאחריו מופעלות מספר היוריסטיקות לשיפור פתרון זה. גם Golden הציג אלגוריתם מבוסס על אלגוריתם Savings לפתרון בעיה.

בעולם האמיתי, באזורים כגון אזורים אורבניים צפופים, זמן הנסיעה מלקוח אחד לשני תלוי הן במרחק בין שני הלקוחות, והן בשעת היום. אם לא נתייחס לעובדה שעבור צירי נסיעה מסוימים, זמני הנסיעה משתנים בהם בהתאם לשעות היום, אנו עלולים לקבל תוצאות רחוקות מהתוצאות האופטימאליות. לכן פותחו אלגוריתמים לפתרון בעיית ניתוב הרכבים תלויי הזמן. Malandraki הציג שני אלגוריתמים לפתרון בעיית ניתוב הרכבים תלויי זמן, הראשון הוא אלגוריתם חמדני, והשני הוא אלגוריתם של סיעוף וחסימה שמתאים לבעיות קטנות. Ichoua, Gendreau ו-Potvin הציגו אף הם אלגוריתם לפתרון הבעיה.

בעיית ניתוב רכבים סטוכסטיים מתעוררת כאשר אחד או יותר ממשתני הבעיה הם אקראיים. Tillman הציע פתרון מבוסס על אלגוריתם Savings לבעיית ניתוב רכבים עם דרישות סטוכסטיות כאשר ישנם מספר מחסנים. Stewart ו-Golden הציגו מודל CCP (תכנות תחת אילוצים) ושני מודלים של SPR (תכנות סטוכסטי). במודלים מבוססי SPR, Stewart ו-Golden משייכים "עונש" לקטעים במסלול שיש להם סבירות גבוהה שיגרמו להארכת המסלול האמיתי. בנוסף משתמשים Stewart ו-Golden בשיטת ה-Savings על מנת להגיע לפתרון. Bertsimas מציג מספר אלגוריתמים לפתרון בעיית ניתוב רכבים עם לקוחות סטוכסטיים, שבה עבור כל לקוחו בעל דרישה קבועה וידועה קיימת הסתברות כל שהיא שנצטרך להגיע אליו.

מטרת עבודה זו היא פיתוח מודל לניתוב רכבים המאופיין במשכי נסיעה תלויי זמן וסטוכסטיים וכן פיתוח אלגוריתם היוריסטי למציאת פתרון קרוב לפתרון האופטימאלי עבור המודל, כשהמטרה היא למצוא אוסף מסלולים שסך זמן הנסיעה בהם יהיה מינימאלי. אנו נתייחס לכך שעבור צירים מסוימים (חלקם או כולם) זמן הנסיעה בציר משתנה לאורך היום, וכן הוא מתנהג סטוכסטית, כלומר, לכל שעה בשעות היום ישנה הסתברות שזמן הנסיעה הוא P_1 והסתברות אחרת שזמן הנסיעה הוא P_2 וכדומה. מודל זה הוא ניסיון נוסף לקרב את בעיית ניתוב הרכבים לעולם האמיתי, כשלמיטב ידיעתי, נכון להיום לא קיימת בספרות התייחסות לשילוב זה של בעיות.

מכוון שהמטרה הכללית של כל ארגון היא הורדת עלויות למינימום ההכרחי, הרי שגם במודל זה ההנחה היא שהורדת זמן הנסיעה תצמצם את הוצאות הארגון, כאשר זמן הנסיעה משפיע על העלויות במספר אופנים כגון עלויות הדלק, האחזקה (טיפולים מונעים וכו'), תחזוקה (תיקון תקלות ותאונות), קנסות וכו'.

בעיית ניתוב הרכבים היא בעיית אופטימיזציה קשה, אשר לא ניתן לפתור אותה באופן אופטימאלי עבור בעיות בסדר גדול. הוספת אילוצים שונים, הופכת את הבעיה לקשה עוד יותר, ולכן האלגוריתם המוצג לפתרון הבעיה הוא אלגוריתם היוריסטי.

בעבודה זו האלגוריתם שיוצג הוא אלגוריתם המבוסס על אלגוריתם Savings, אשר עבר התאמה והרחבה על מנת שיוכל להתמודד עם תלות בזמן ועם סטוכסטיות. ישנם מספר תנאים בסיסיים אשר צריכים להתקיים על מנת שנוכל להשתמש באלגוריתם זה. התנאים הנם:

1. מספר הלקוחות הוא קבוע.
2. הדרישה של כל לקוח ידועה וקבועה.
3. דרישת כל אחת מהלקוחות קטנה מיכולת ההובלה של כלי רכב בודד בצי.
4. סך כל הדרישות של כל הלקוחות גדול מיכולת ההובלה של כלי רכב בודד בצי.
5. כל כלי הרכב בצי הם בעלי אותה קיבולת.
6. כל כלי הרכב נעים באותו פרק זמן ביום (לדוגמא, כל כלי הרכב נעים בין 8 בבוקר ל-8 בערב).
7. פרק הזמן ביום הוא מחזורי, כלומר כלי רכב שלא סיים את המסלול שלו בסוף פרק הזמן הנוכחי, ממשיך את המסלול שלו באותה הנקודה בתחילת פרק הזמן הבא.
8. מספר תת-פרקי הזמן הוא קבוע עבור כל הקשתות בגרף.
9. אורכי כל תת-פרקי הזמן שווה.
10. עבור כל קשת בגרף, נתונים לנו אורך הקשת ועבור כל פרק זמן נתונות פונקצית מהירות ופונקצית ההתפלגות שלה.

אלגוריתם Savings עליו מבוסס האלגוריתם המוצע בעבודה זו מיועד עבור בעיית ניתוב רכבים דטרמיניסטית. אנו נשתמש במספר כלים ותוספות על מנת שנוכל לפתור את בעיית ניתוב הרכבים תלויי הזמן והסטוכסטיים בעזרת אלגוריתם Savings.

הכלי הראשון שבו נשתמש הוא להפוך את הבעיה מבעיה תלויית זמן וסטוכסטית, לבעיה דטרמיניסטית. מעבר מבעיה תלויית זמן וסטוכסטית לבעיה דטרמיניסטית משמש ככלי לקבלת החלטה עבור אלגוריתם Savings.

בעבודה זו אנו נשתמש בשלוש שיטות על מנת להפוך את הבעיה תלויית הזמן והסטוכסטית לבעיה דטרמיניסטית.

1. **חישוב ממוצע** – חישוב הממוצע מתבצע בשני שלבים. בשלב הראשון, עבור כל יחידת זמן, אנו מחשבים את ממוצע מהירות הנסיעה בקטע ביחידת הזמן. השלב השני הוא חישוב הממוצע הכללי, כלומר, עבור כל קטע אנו סוכמים את המהירויות הממוצעות שחושבו עבור כל אחת מיחידות הזמן, ומחלקים במספר יחידות הזמן.
2. **חישוב הערך הטוב ביותר** – חישוב הערך הטוב ביותר הוא למעשה חיפוש המהירות המהירה ביותר האפשרית בין כל יחידות הזמן, ללא התחשבות בהסתברות לקבלת אותה המהירות.
3. **חישוב הערך הגרוע ביותר** – בדומה לחישוב הערך הטוב ביותר, חישוב הערך הגרוע ביותר הוא חיפוש המהירות הנמוכה ביותר האפשרית בין כל יחידות הזמן, גם כאן אין אנו מתחשבים בהסתברות לקבלת אותה המהירות.

הכלי השני שאותו אנו מוסיפים לאלגוריתם ה-Savings הוא סימולציה. סימולציה הנה טכניקה שמשתמשת במחשב על מנת לחקות פעולות במציאות מורכבת. מטרת הסימולציה היא לייצג מאפיינים מסוימים בהתנהגות של מערכת על מנת לקבל תחזיות לגבי התנהגות המערכת בתנאים שונים.

בעבודה זו נעשה שימוש בסימולציה על מנת לדעת מהו זמן המעבר במסלול נתון. בעבודה זו, אנו מעוניינים לדעת מהיא המהירות המקסימאלית, שבאחוז מסוים מהמקרים (נניח 80%) נקבל את המהירות הזאת, או מהירות נמוכה ממנה. אם נבצע את הסימולציה מספר רב של פעמים, ונשמור את התוצאה המתקבלת מכל אחת מהסימולציות, נוכל להשתמש בתוצאות הללו לשם ביצוע אותם ניתוחים סטטיסטיים.

ישנם מספר פרמטרים הבאים לידי ביטוי באלגוריתם המוצע בעבודה זו :

n – מספר הלקוחות.

m – מספר זוגות המסלולים המועמדים לאיחוד, ושהחיסכון שלהם מחושב ע״י סימולציה.

r – מספר החזרות על האלגוריתם.

אנו נציג את האלגוריתם ונראה כיצד הפרמטרים הללו באים ליידי ביטוי.

האלגוריתם:

1. בניית פתרון ראשוני. הפתרון הראשוני הוא בעל n מסלולים. כל מסלול יוצא מהמחסן, עובר דרך לקוח בודד, וחוזר למחסן. בין n המסלולים, אין שני מסלולים אשר עוברים דרך אותו הלקוח.
2. עבור כל זוג מסלולים, אנו מחשבים את הרווח המתקבל מחיבורם. ע״י שימוש בנתונים דטרמיניסטיים.
3. עבור m זוגות המסלולים, שהרווח המתקבל מחיבורם הוא הגדול היותר (על-פי הנתונים הדטרמיניסטיים), ואשר עומדים באילוצים השונים של הבעיה, אנו מחשבים את הרווח האמיתי אשר יתקבל מחיבורם ע״י שימוש בסימולציה.
4. את זוג המסלולים אשר קיבל בעזרת סימולציה את הרווח הגדול ביותר אנו מחברים למסלול חדש.
5. אנו חוזרים על שלבים 2 עד 4 כל עוד קיימים זוגות של מסלולים אשר חיבורם מהווה רווח ״חיובי״, והם עונים על תנאי הבעיה.

את האלגוריתם אנו מריצים 3 פעמים, כאשר בפעם הראשונה אנו משתמשים בשיטת הממוצע בכדי להפוך את הבעיה מבעיה תלויית זמן וסטוכסטית לדטרמיניסטית. בפעם השניה או משתמשים בשיטת הערך הטוב ביותר, ובפעם השלישית בשיטת הערך הגרוע ביותר. הפתרון של האלגוריתם הוא הפתרון הטוב ביותר שהתקבל ע״י שימוש ב-3 השיטות.

נקודה נוספת, שאין אליה התייחסות באלגוריתם ה-Savings המקורי, היא התמודדות עם מצב שבו עבור שני זוגות מסלולים מתקבל אותו החיסכון. במקרה כזה, על-פי אלגוריתם Savings, אנו בוחרים את אחד הזוגות באופן אקראי. מכוון שבחירה של זוג כלשהו משפיעה על התוצאה הסופית, אנו יכולים לקבל תוצאות שונות כאשר נחבר מסלולים שונים. מסיבה זו, אם

אנו נתקלים במצב שבו אנו צריכים לבחור אקראית בין שני זוגות מסלולים, אנו חוזרים על האלגוריתם מספר קבוע של פעמים, r, בכדי שנוכל כל פעם לבחור בין זוג שונה של מסלולים.

סיבוכיות האלגוריתם נבחנה ונמצא שהיא $O(n^3m)$, כלומר, הגורם המשפיע ביותר על סיבוכיות האלגוריתם הוא מספר הקדקודים בגרף והגורם השני בחשיבותו הוא מספר זוגות המסלולים אשר אנו שומרים בכדי לחשב את החיסכון המתקבל מחיבורם ע"י סימולציה. על סמך ניתוח הסיבוכיות אנו מצפים שזמן ריצת האלגוריתם יגדל באופן ישר ככל שמספר זוגות המסלולים המועמדים לאיחוד, ושהחיסכון שלהם מחושב ע"י סימולציה גדל (m), אבל בפועל, על סמך בדיקות שבוצעו מסתבר שזמן הריצה אכן גדל באופן ליניארי, אך הוא נמוך מהמצופה, וכן הוא הולך וקטן ככל שמספר הלקוחות גדל. באופן דומה, זמן הריצה אינו גדל באופן שציפינו כאשר כמות הלקוחות (n) גדלה, אלא הרבה פחות. ההסבר לשני ההבדלים הללו הוא תנאים באלגוריתם שגורמים להפסקת ריצת האלגוריתם ואינם מאפשרים לעבור (כי אין צורך בכך) על כל האפשרויות לחיבור מסלולים.

היבט חשוב נוסף, שיש לבחון באלגוריתם הוא איכות התוצאות המתקבלות ממנו. איכות התוצאות משמעותה היא עד כמה המסלולים המתקבלים ע"י הפעלת האלגוריתם קרובים לתוצאה האופטימאלית.

תוצאות האלגוריתם נבדקו במספר תנאי ריצה שונים.

1. נתוני הבעיה הם דטרמיניסטיים בלבד.
2. נתוני הבעיה סטוכסטיים בלבד. עבור בעיות סטוכסטיות אנו נבדוק את הגורמים הבאים:
   א. השפעת אחוז הקשתות שמתנהגות סטוכסטית.
   ב. השפעת טווח מהירות הנסיעה.
   ג. השפעת מספר ההסתברויות למהירות לכל קשת ביחידת זמן.
3. נתוני הבעיה הם סטוכסטיים ותלויי זמן. זו למעשה הבעיה המקורית איתה האלגוריתם אמור להתמודד. מכוון ולא קיים אלגוריתם שתוכנן במיוחד עבור סוג זה של בעיות, אנו צופים התוצאות שיתקבלו יהיו טובות יותר מכל תוצאה אשר תתקבל מאלגוריתם אחר אשר ינסה להתמודד עם סוג זה של בעיות.

מניתוח התוצאות התקבלו המסקנות הבאות:

1. הגדלת מספר פרקי הזמן בבעיה, מגדילה את הפער בין תוצאת אלגוריתם Savings והאלגוריתם המוצע ובין הפתרון האופטימאלי.
2. הגדלת אחוז הקשתות שמתנהגות סטוכסטית, מגדילה את הפער בין תוצאות אלגוריתם המוצע ובין הפתרון האופטימאלי. הגדלת אחוז הקשתות שמתנהגות סטוכסטית אינה משפיעה על תוצאת אלגוריתם Savings.
3. הגדלת מספר ההסתברויות למהירות לכל קשת ביחידת זמן, מגדילה את הפער בין תוצאות אלגוריתם Savings והאלגוריתם המוצע ובין הפתרון האופטימאלי.

4. למספר ההסתברויות למהירות לכל קשת ביחידת זמן, השפעה גדולה יותר מאשר לאחוז הקשתות המתנהגות סטוכסטית באלגוריתם המוצע בלבד וזאת מכוון שלמספר ההסתברויות למהירות לכל קשת ביחידת זמן אין השפעה על אלגוריתם Savings.
5. טווח מהירויות קטן יותר מאפשר לאלגוריתם Savings והאלגוריתם המוצע להגיע לתוצאות שהנן קרובות יותר לפתרון האופטימאלי.

בהתאם למסקנות אלו, נבנו 20 בעיות, המאופיינות בטווח מהירויות גדול, מספר הסתברויות למהירות לכל קשת ביחידת זמן שהינו יחסית גדול, אחוז הקשתות המתנהגות סטוכסטית הוא 100% ומספר יחידות הזמן הוא 24. מטרת הבעיות הללו היא לבחון את איכות התוצאות המתקבלות מהאלגוריתם, כאשר הבעיה היא סטוכסטית ותלויית זמן.

מהתוצאות שהתקבלו עולה שבממוצע תוצאת האלגוריתם סוטה בכ-6% מהתוצאה האופטימאלית, בעוד שאלגוריתם Saving סוטה בכ-17% מהתוצאה האופטימאלית. Cordeau, Gendreau, Potvin ו-Semet ביצעו השוואה בין מספר אלגוריתמים נפוצים, ובין היתר הראו שעבור בעיית VRP הבסיסית, סטיית אלגוריתם Savings מהפתרון האופטימאלי (או הפתרון הטוב ביותר הידוע), עומדת בממוצע על 6%. על סמך התוצאות שלנו, עבור בעיות סטוכסטיות ותלויות זמן, האלגוריתם המוצע מספק פתרונות כשאחוז הסטייה מהתוצאה האופטימאלית קרוב לאחוז הסטייה של אלגוריתם Savings עבור בעיות ניתוב רכבים רגילות.

בנוסף לבדיקת איכות הפתרונות של האלגוריתם, נבדק גם השפעת מספר זוגות המסלולים המיועדים לאיחוד ואשר החיסכון שלהם מחושב בעזרת סימולציה. על סך תוצאות הבדיקות שנערכו, נראה ששימוש ב-3 עד 5 מועמדים לאיחוד נותן את השיפור המשמעותי ביותר בתוצאת האלגוריתם. שימוש במספר גדול יותר של מועמדים משפר לרוב את התוצאה, אך משך הזמן שמוקדש לחיפוש התוצאה האופטימאלית גדל ביחס ישר לכמות המועמדים לאיחוד, בעוד שהשיפור הוא קטן.

1. כללי

ארגונים רבים, מנהלים צי של כלי רכב המשמש לצרכים לוגיסטיים שונים כגון שינוע תוצרת ממחסן מרכזי לנקודות מכירה, צי של רכבים ממונעים קטנים המשמשים לשינוע חומרים במפעל, וכו׳. בעקבות הפעלת תהליכי שיפור והתייעלות, ארגונים רבים מעוניינים לייעל את תפעול צי הרכבים שלהם. אחת הדרכים לייעול צי רכבים היא למצוא מסלולי נסיעה שעלות הנסיעה בהם נמוכה יותר. ברוב המקרים, מסלולים קצרים יותר נחשבים למסלולים יעילים יותר. אולם עבור ארגונים המספקים שירותים כגון שרותי חירום (מד״א, משטרה, כיבוי אש וכו׳), שרותי שליחויות וכו׳, שינויים במהירות הנסיעה במסלול במהלך היום, עקב פקקי תנועה למשל, וכן מצבים בהם חוסר וודאות ושונות בזמן הנסיעה במסלול הנסיעה, יכולים להפוך מסלול קצר למסלול שאינו מאפשר לנו לספק את השרות לשביעות רצון הלקוחות.

בעבודת תזה זו נסקור את נושא בעיית ניתוב הרכבים, ואף נציג פתרונות שונים לבעיה. בנוסף, נסקור מספר הרחבות שונות לבעיה. אנו נשלב את מודל תלות זמן הנסיעה בקטע בזמן ביום ואת מודל זמני הנסיעה האקראיים בקטע למודל חדש, וכן נציג הרחבה של אלגוריתם קיים אשר נותן פתרונות קרובים לאופטימאליים לבעיה החדשה שלנו.

2. מבוא

ארגונים רבים, בסדר גודל בינוני ומעלה, מנהלים צי של כלי רכב המשמש לצרכים לוגיסטיים שונים. צי זה יכול להיות צי משאיות המשמשות לשינוע תוצרת ממחסן מרכזי לנקודות מכירה, צי של רכבים ממונעים קטנים המשמשים לשינוע חומרים במפעל, וכו׳.

החל משנות ה-80, אנו עדים לתהליכי שיפור והתייעלות של ארגונים רבים בתהליכים הלוגיסטיים השונים הקיימים בהם ([50]). מאחר וצי כלי הרכב של הארגון מהווה נקודת מפתח בשרשרת האספקה של הארגון, לתפקודם האופטימאלי יש חשיבות אסטרטגית רבה. בנוסף, תכנון נכון יכול לחסוך כסף לארגון. למשל, אם סך מרחק הנסיעה יהיה הקצר ביותר האפשרי נוכל לחסוך בעלויות הנובעות מקיצור המרחק, כגון עלויות הדלק, האחזקה (טיפולים מונעים וכו׳), תחזוקה (תיקון תקלות ותאונות), קנסות וכו׳.

אולם לא תמיד אנו רוצים או יכולים להתבסס על המסלולים הקצרים ביותר. למשל, בכדי להוריד את עלויות צריכת הדלק עדיף לחשוב על זמן קצר יותר ולא על מרחק, היות ולעיתים המסלול הקצר ביותר הוא לא המסלול המהיר ביותר. ישנם שיקולים נוספים שיכולים להשפיע גם הם. חברה המובילה חומרים מסוכנים, ואשר צריכה לעבור באזורים עירוניים, תעדיף לנסוע במסלול המהיר ביותר, כך שבמידה ותהיה תקלה כלשהי, הסיכון שאנשים יפגעו יהיה קטן יותר. חברות שמספקות שירותי הסעות לבתי ספר, אינן יכולות להרשות לעצמן לאחר או להקדים, היות והן צריכות לאסוף תלמידים, אשר מחכים להם בנקודות מסוימות בשעות מסוימות, במקרים כאלו, שיבושים לא צפויים בדרך יכולים לגרום לאיחור במסלול, ובסופו של דבר אף לאיחור ההגעה לבית הספר, ולכן אנו נרצה למצוא את המסלול הקצר ביותר האפשרי, אך בעל אמינות גבוהה נגד שיבושי תנועה.

Larsen ([49]) מציג סקירה של בעיות בהן לזמן חשיבות מרבית. בין הבעיות Larsen מציג את בעיית שרותי החירום. בבעיה זו יש להקצות מסלולים לרכבי חירום (משטרה, מד״א, כיבוי אש וכו׳), כך שכל רכב יכול להגיע מנקודת המוצא שלו אל היעד במהירות המקסימאלית. ברוב המקרים, אי אפשר לתכנן מסלולים מראש, היות והקריאות מתקבלות בזמן אמת ואינן ידועות. שיטות לניתוב רכבים של שרותי חירום כוללות לרוב ניתוח של מיקום כלי הרכב השונים ,המרחק לנקודת היעד וניתוח עומסים על צירי הנסיעה השונים, כך שכלי הרכב שיכול להגיע במהירות הרבה ביותר לנקודת היעד, הוא זה שנשלח.

בעיה נוספת שמוצגת ע״י Larsen היא פעילותן של חברות שליחויות. חברות שליחויות הנן חברות אשר מספקות שרותי משלוח ואיסוף של פריטים מלקוח ללקוח. כל לקוח יכול להשתמש בשרותי הדואר בכדי לשלוח פריטים ללקוח אחר, היתרון בשימוש בשירותיהם של חברות שליחויות הוא במהירות ההפצה. בעוד שמשלוח בדואר יכול לקחת יום / יומיים, משלוח בעזרת חברת שליחויות לוקח בין מספר דקות למספר שעות בלבד. חברות השליחויות, אשר נדרשות לזמני שליחה מהירים, צריכים להתמודד עם חוסר מידע מראש של דרישות הלקוחות ויעדיהם. בנוסף, בעיקר בסביבה עירונית, זמני הנסיעה משתנים במהלך היום, ועל כך גם יש לתת את הדעת.

Cole ([17]) מזכיר את חשיבותה של הפצה מהירה בניהול שרשרת ההפצה, במיוחד עבור חברות אשר עובדות בשיטת Just In Time, ואשר תלויות בקבלת חומרי הגלם שלהם מספק חיצוני. עבור חברות כאלו, עיכוב בקבלת חומר גלם מהספק משמעו השבתת קו הייצור.

תחום שבו מהירות הפצת הסחורה מהווה נדבך חשוב הוא תחום מוצרי הפרמצבטיקה. הפצה מהירה של מוצרי הפרמצבטיקה הנה הכרחית על מנת להבטיח את יעילות ואפקטיביות מערכות הבריאות. הפצה של מוצרי הפרמצבטיקה משמע העברה של כמויות גדולות של פריטים שונים, שלרוב צריכים להגיע במהירות לנקודות היעד. לדוגמה, בארה״ב, מערך ההפצה צריך לספק סחורה ליותר מ-130000 בתי מרקחת, תוך 12 שעות מזמן ההזמנה, כאשר מספר הפריטים השונים שהם מספקים הוא כ-10000 (מתוך מחקר שנערך ב-2004 ע״י Booz Allen Hamilton לבקשת Healthcare Distribution Management Association (HDMA)).

בנוסף לצורך הרגיל, היום יומי, של תרופות וציוד רפואי, ישנם מצבי חירום בסדרי גודל רחבי הקיף, כדוגמת התקפות טרור, פגעי טבע וכו׳, שבעקבותם יש דרישה גבוהה לתרופות וציוד רפואי. לרוב, למרות ההסתברות הנמוכה למצבים כאלו, מחזיקות רוב המדינות מלאי של תרופות וציוד רפואי, אשר בשעת הצורך נשלח לאזור מצב החירום (Emergency Staging Area – ESA). לדוגמה, בארה״ב, הממשל הפדרלי מחזיק מלאי חירום לאומי אסטרטגי (Strategic National Stockpile – SNS), המכיל בין היתר 300 מיליון מנות של חיסון אבעבועות שחורות, ואנטיביוטיקה המספיקה לטיפול ב-20 מיליון אנשים נגד אנטראקס. בנוסף, מערכות מלאי המנוהלות ע״י הספק (Vendor Managed Inventory - VMI) גם כן פותחו בכדי להגדיל את מלאי החירום הלאומי האסטרטגי, כאשר הספקים מחויבים לספק את הסחורה תוך 21 עד 36 שעות ([6], [1]). בזמן של מצב חירום רפואי, יש להוביל ולהפיץ הציוד הרפואי שנמצא במלאי החירום הלאומי האסטרטגי וכן את זה שנמצא אצל הספק לאזור האירוע ולמרכזי ההפצה אשר אליהם האוכלוסייה פונה על מנת לקבל עזרה. הובלה והפצה מהירים של מוצרים בכמויות גדולות דורשים תכנון זהיר וביצוע מקצועי בכדי להציל עד כמה שיותר חיי אדם, במיוחד באזורים עירוניים המאוכלסים בצפיפות.

ספקי מוצרי הפרמצבטיקה בארה״ב מתמודדים עם הצורך בהפצה מהירה של מוצרים ע״י בחינה ומיקום קפדני של מתקנים החברה השונים, בניהם מתקני היצור, המחסנים וכו׳. וכן באופן שבו מתוכננים מסלולי ההפצה, אשר צריכים להתמודד עם עומסים אפשריים וחסימות צירי נסיעה.

בכדי לעמוד בדרישות היום יומיות, וכן בדרישות לא סטנדרטיות אחרות, עומדים לרשותנו מודלים מתחום חקר ביצועים, אשר מהווים חלק חשוב בפתרון בעיות לוגיסטיות אלו. באופן כללי, ניתן לתאר את שתי הבעיות העיקריות תוך שימוש ברשת של דרכים המשמשת להפצה, כאשר בעיית מיקום מתקני הייצור והמחסנים (Facility Location Problem) מתייחסת למיקום המתקנים על גבי רשת ההפצה, ועבור בעיית ההפצה אנו מדברים על בעיית ניתוב רכבים (Vehicle Routing Problem – VRP) על גבי הרשת, שתאפשר להפיץ את הסחורה באופן יעיל.

3. סקירה ספרותית

בעיית ניתוב רכבים (Vehicle Routing Problem – VRP) הוא שם כולל לקבוצה של בעיות שבהם יש להקצות סדרה של מסלולי נסיעה עבור צי של כלי רכב, היוצאים מאתר מרכזי אחד (או מספר אתרים) ואשר צריכים לעבור דרך מספר לקוחות. המטרה העומדת בעת הקצאת המסלולים היא לספק (או לאסוף) סחורה ללקוחות, תוך הורדת עלויות ההובלה וההפצה השונות של צי כלי הרכב של הארגון, למשל ע״י מציאת המסלולים הקצרים ביותר לכל אחד מכלי הרכב, תוך עמידה באילוצים שונים, כגון אילוצי הקיבולת של כלי הרכב. כל אחד מכלי הרכב שעוזבים את המחסן, חוזר בסופו של המסלול אל המחסן עצמו ([10]).

ישנו מגוון רב של בעיות ניתוב רכבים, הנבדלים זה מזה באילוצים השונים המתווספים לבעיה הבסיסית. Laporte ,Gendreau ו-Seguin ([35]) הציגו במאמרם את שלושת האילוצים השכיחים ביותר הקיימים עבור VRP :

1. אילוץ קיבולת - דרישות הלקוחות הנמצאים על מסלול ההפצה של המשאיות אינו יכול לעלות על הקיבולת של משאית בודדת.
2. אילוץ אורך – האורך של כל אחד ממסלולי ההפצה אינו יכול לעלות אורך קבוע וידוע מראש.
3. אילוץ חלונות זמן – אל כל לקוח יש להגיע בחלון זמן אשר נקבע ע״י הלקוח.

שלושת האילוצים הללו הולידו בין היתר את בעיות ניתוב הרכבים הבאות :

- ניתוב רכבים עם חלוקת הובלות - ווריאציה שבה ניתן לספק סחורה ללקוח ע״י יותר ממשאית אחת (למשל אם קיים לקוח שהדרישה שלו גדולה מקיבולת של משאית בודדת).
- ניתוב רכבים עם חלונות זמן - אספקת הסחורה ללקוח צריכה להתבצע בשעות מסוימות במשך היום.

כמו כן קיימות אפשרויות נוספות שאינן נובעות ישירות משלושת האילוצים הנ״ל, כגון :

- ניתוב מסלולים ממספר אתרים – המשאיות של הארגון יוצאות ממספר אתרים. בווריאציות מסוימות כל משאית חוזרת לאתר ממנו היא יצאה, ובווריאציות אחרות ישנה אפשרות שהמשאיות יחזרו לאתר אחר.
- גרסאות סטוכסטיות של הבעיה, שבהן חלק מהנתונים אינם קבועים, והם בעלי הסתברות כלשהי שהם יתרחשו (כגון דרישות לקוחות משתנים, זמני נסיעה משתנים וכו׳).

בהמשך אנו נסקור לעומק מספר הרחבות לבעיית ניתוב הרכבים הבסיסית.

3.1. בעיות בסיסיות

ההרחבות השונות של בעיית ניתוב רכבים מבוססות על שתי בעיות בסיסיות, בעיית הסוכן הנוסע ובעיית מחלק הדואר הסיני ([32]). עבור שתי הבעיות, אנו מדברים על גרף קשיר משוקלל בלתי מכוון (גרף שבו קיים מסלול בין כל זוג קדקודים, ואשר לקשתות שלו אין כיוון מוגדר), אשר בו n קדקודים ו-m קשתות המחברות בין הקדקודים השונים של הגרף.

בבעיית מחלק הדואר הסיני (Chinese Postman Problem - CPP) ([59]), אנו מעוניינים למצוא את המסלול בעל המשקל המינימאלי, העובר דרך כל הקשתות של הגרף (בכל קשת רק פעם אחת, במידה וזה אפשרי, ואם לא, אז במינימום הנדרש). במושגי תורת הגרפים, בעיית

מחלק הדואר הסיני מוגדרת כמציאת מעגל אויילר בגרף שלם ממושקל (ראה [25], פרק 1). מציאת מעגל אויילר היא בעיה השייכת לקבוצת בעיות הNP-Complete (קבוצת בעיות הכרעה, שהתשובה אליהן היא כן או לא, שניתן לבדוק את הפתרון שלהם בזמן ריצה פולינומיאלי, אך לא ניתן למצוא את הפתרון עצמו בזמן ריצה פולינומיאלי) (ראה [19]), כך שלא ניתן לפתור בזמן ריצה סביר. מחקרים שנעשו בתחום תורת הגרפים תרמו לפתוח אלגוריתמים לפתרון הבעיה, לדוגמא ניתן לבדוק אם בכלל קיים מעגל אויילר עבור גרף מסוים, ע"י בדיקת דרגות כל הקדקודים שלו, אשר צריכות להיות זוגיות.

את בעיית מחלק הדואר הסיני ,עבור גרף בעל N+1 קדקודים, הממוספרים מ-0 עד N, אפשר לנסח כבעיית תכנות ליניארי בשלמים באופן הבא ([59]) :

$$\min \ Z = \sum_{i=0}^{N}\sum_{j=0}^{N} d_{ij}x_{ij} \quad (1.1)$$

$$\sum_{k=0}^{N} x_{ki} - \sum_{k=0}^{N} x_{ik} = 0 \quad i = 0,1,\ldots,N \quad (1.2)$$

$$x_{ij} + x_{ji} \geq 1 \quad \forall (i,j) \in E \quad (1.3)$$

$$x_{ij} \geq 0 \quad (1.4)$$

$$x_{ij} \in \mathbb{Z} \quad (1.5)$$

d_{ij} מציין את המרחק בין קודקוד i ולקודקוד j. x_{ij} היינו משתנה המציין כמה פעמים אנו עוברים בקשת היוצאת מקודקוד i ומגיעה לקודקוד j. כאמור, אנו מעוניינים למצוא את המסלול בעל המשקל המינימאלי, העובר דרך כל הקשתות של הגרף, כפי שמצוין ב-(1.1). (1.2) מציין שמספר הקשתות המגיעות לכל קודקוד חייב להיות שווה למספר הקשתות היוצאות מאותו הקודקוד. תנאי זה מבטיח שאנו לא נעצור בקודקוד מסוים, וש"מעגליות" המסלול תתקיים. (1.3) מציין שחייבים לעבור לפחות בפעם אחת בכל אחת מהקשתות (כש-E היא קבוצת הקשתות), כאשר (1.4) מבטיח ש- x_{ij} לא יקבל ערכים שליליים, ו-(1.5) מציין ש- x_{ij} יכול לקבל רק מספרים שלמים.

בבעיית הסוכן נוסע (Traveling Salesman Problem - TSP) ([32]), המטרה היא למצוא את המסלול בעל המשקל הקטן ביותר, העובר דרך כל קדקודי הגרף, בכל קודקוד רק פעם אחת בלבד, פרט לנקודת ההתחלה, אליה אנו חוזרים בסוף המסלול. במושגי תורת הגרפים, בעיית הסוכן נוסע מוגדרת כמציאת מעגל המילטוני בגרף שלם ממושקל (ראה [25], פרק 10). מציאת מעגל המילטוני אף היא שייכת לקבוצת בעיות ה-NP-Complete, ומכאן שגם לבעיה זו לא קיים אלגוריתם שפותר אותה בזמן סביר.

Miller, Tucker ו-Zemlin ([53]),שהושפעו ממחקרים שבהם הוצגו בעיות מתמטיות כבעיות תכנות ליניארי בשלמים, הציגו את בעיית הסוכן נוסע כבעיית תכנות ליניארי בשלמים. לדעתם, הגידול במחקרים ובבעיות אשר מוגדרות כבעיות תכנות ליניארי בשלמים, יביא בעקבותיו שיטות יעילות ומהירות לפתרון בעיות אלו. הניסוח הפורמאלי של בעיית הסוכן נוסע כבעיית תכנות ליניארי בשלמים עבור גרף בעל N+1 קדקודים (הממוספרים מ-0 עד N), מוגדר באופן הבא ([20]) :

$$\min \quad Z=\sum_{i=0}^{N}\sum_{j=0}^{N}d_{ij}x_{ij} \tag{1.6}$$

תחת האילוצים:

$$\sum_{i=0}^{N}x_{ij}=1 \quad \forall j\in\{0,1,2,..,N\} \tag{1.7}$$

$$\sum_{j=0}^{N}x_{ij}=1 \quad \forall i\in\{0,1,2,..,N\} \tag{1.8}$$

$$x_{ii}=0 \quad \forall i\in\{0,1,2,...,N\} \tag{1.9}$$

$$x_{ij}\geq 0 \quad \forall i,j\in\{0,1,2,...,N\} \tag{1.10}$$

$$x_{ij}\in\{0,1\} \tag{1.11}$$

d_{ij} מציין את המרחק בין קודקוד i ולקודקוד j. x_{ij} היינו משתנה החלטה שמקבל את הערך 1 אם אנו בוחרים לנוע מקודקוד i לקודקוד j, ואת הערך 0 אם אנו בוחרים לא לנוע בין שני הקדקודים הללו, תנאי זה מצוין על-ידי (1.10), אשר מציין שמשתנה ההחלטה x_{ij} אינו יכול לקבל ערכים שליליים, ו-(1.11) אשר מציין בפרוש שהוא צריך לקבל את אחד מהערכים 0 או 1. אנו מעוניינים להביא למינימום את אורך המסלול שבו יש לעבור, כפי שמצוין ב-(1.6).

בכדי שנעבור בכל אחד מהקדקודים רק פעם אחת, אנו צריכים להיכנס לכל אחד מהקדקודים פעם אחת בלבד, ולצאת מכל אחד מהקדקודים גם כן רק פעם אחת בלבד, תנאי זה מובא ב-(1.7) וב-(1.8). כמו כן, אנו מוסיפים את תנאי (1.9) בכדי להימנע ממעגלים (כניסה ויציאה מאותו הקודקוד).

Miller, Tucker ו-Zemlin ([53]),אשר הציגו את בעיית הסוכן נוסע כבעיית תכנות ליניארי בשלמים, לא התייחסו לבעיית הסוכן נוסע הקלאסית, כי אם לבעיית סוכן נוסע מורחבת יותר. בבעיית הסוכן נוסע הקלאסית ([32]), סוכן אשר יוצא מנקודה מסוימת צריך לעבור דרך n ערים שונות ולחזור לנקודת המוצא שלו, כאשר המטרה היא לעשות את זה בדרך הקצרה ביותר. Miller, Tucker ו-Zemlin הרחיבו את בעיית הסוכן נוסע ע״י הוספת מספר תנאים:

1. הסוכן חייב לחזור לנקודת המוצא שלו t פעמים.
2. על הסוכן לבקר בלא יותר מ-p ערים במסלול.

המשמעות של תנאים אלו היא שלא ניתן למצוא מסלול בודד שבעזרתו ניתן לעבור דרך n הערים (אלא אם $t=1$ ו-$p>n$), ויש צורך לבנות מספר מסלולים שונים, כאשר אין שני מסלולים העוברים דרך אותה העיר בכדי לפתור את הבעיה.

הרחבות כגון אלו היוו את הבסיס לבעיית ניתוב הרכבים.

3.2. בעיית שיגור המשאיות

בעיית ניתוב הרכבים (Vehicle Routing Problem – VRP) נוסחה לראשונה ע״י Dantzig ו-Ramser ([21]), אשר קראו לה בעיית שיגור המשאיות (Truck Dispatching Problem). בבעיית שיגור המשאיות על ארגון לספק סחורה ל-n לקוחות שונים, הממוקמים במקומות שונים ממחסן מרכזי אחד. לצורך כך יש לארגון צי כלי רכב ייעודי לצורכי הפצה. הקיבולת של כל אחד מכלי הרכב בצי הנה גדולה מדרישה של לקוח בודד (כל לקוח).

Dantzig ו-Ramser ([21]) היו מעוניים למצוא את המסלולים שבהם כלי הרכב בצי צריכים לנוע, כך שיוכלו לספק את הסחורה לכל אחד מהלקוחות, וכן שהאורך של סך המסלולים הוא מינימאלי. לצורך כך הם הציגו אלגוריתם היוריסטי שפעל בשני שלבים. בשלב הראשון נבנה פתרון ראשוני פשוט, בו כל כלי רכב בצי משרת רק לקוח אחד. בשלב השני שופר הפתרון הראשוני ע״י איחוד מסלולי הרכבים למספר קבוצות. יעד המודל היה מילוי מקסימאלי של כלי הרכב של הארגון, ולכן האלגוריתם יכול לכלול בקבוצת הלקוחות, גם לקוחות אשר גיאוגרפית ממוקמים רחוק אחד מהשני. בעיית ניתוב הרכבים שאיתה התמודדו Dantzig ו-Ramser הנה בעיית ניתוב הרכבים הבסיסית, אשר ידועה גם בשם בעיית ניתוב הרכבים תלויית קיבולת (Capacitated Vehicle Routing Problem – CVRP), מכוון שהאילוץ היחיד שבו יש להתחשב הוא קיבולת כלי הרכב, אותה אסור לעבור.

VRP היא בעיית אופטימיזציה קומבינטורית קשה, ונכון להיום ניתן לפתור באופן אופטימאלי בעיות בסדר גודל קטן (עד 50 לקוחות). מאז שהבעיה הוצגה באופן פורמאלי לראשונה ע״י Dantzig ו-Ramser, נערך מחקר רב בתום ה-VRP, שבמהלכו פותחו מספר היוריסטיקות לפתרון הבעיה. בתחילת המחקר, הושם דגש על פיתוח אלגוריתמים היוריסטיים שמספקים פתרון העומד באילוצי הבעיה, שהדגש הוא על מהירות השגת הפתרון, וכאשר במידת האפשר בוצעו פעולות אופטימיזציה אחדות ([18]). בעשור האחרון הדגש הוא על פיתוח אלגוריתמים מטה-היוריסטיים (אלגוריתמים אשר משתמשים באלגוריתמים פשוטים יותר ליצירת אוסף פתרונות ראשוניים, ואשר מחפשים פתרון טוב יותר בהתבסס על הפתרונות הראשוניים) המשתמשות בשתי שיטות עיקריות: חיפוש מקומי וחיפוש באוכלוסייה, ומטרתן השגת פתרון מדויק יותר. בהמשך אנו נבחן נסקור מספר מטה-היוריסטיקות פופולאריות לפתרון VRP.

סקירה ספרותית זו מחולקת לשני חלקים, השקולים להתפתחות חקר הבעיה במהלך השנים. תחילה אנו נבחן מספר היוריסטיקות ומטה-היוריסטיקות, אשר פותחו עבור ה-VRP הבסיסי. במקביל לפיתוח אותן היוריסטיקות פותחו, כאמור, הרחבות מתקדמות יותר ל-VRP, אשר מכילות אילוצים רבים יותר מהבעיה הבסיסית. אנו נבחן מספר דוגמאות להרחבות מתקדמות אלו, ונראה כיצד השתמשו בהיוריסטיקות הבסיסיות לפתרון הבעיות המורחבות.

3.3. היוריסטיקות קלאסיות

ההיוריסטיקות והמטה-היוריסטיקות המתוארות בחלק זה הן עבור VRP בסיסי (CVRP). אנו מעוניינים למצוא את המסלול הקצר ביותר (או המסלול בעלות הנמוכה ביותר) עבור צי כלי

הרכב שלנו, אשר יאפשר לנו לספק את הדרישות של כל הלקוחות שלנו. בנוסף מתקיימים התנאים הבאים:

1. מספר הלקוחות הוא קבוע, והדרישה של כל לקוח ידועה.
2. כל כלי הרכב בצי הם בעלי אותה קיבולת.
3. דרישת כל אחד מהלקוחות קטנה מיכולת ההובלה של כלי רכב בודד בצי.
4. סך כל הדרישות של כל הלקוחות גדול מיכולת ההובלה של כלי רכב בודד בצי.

3.3.1. אלגוריתם Savings

Clarke ו-Wright בחנו את בעיית שיגור המשאיות ואת האלגוריתם שהציעו Dantzig ו-Ramser לפתרון הבעיה. Clarke ו-Wright שמו לב לכך, שבעקבות אילוץ שהכניסו Dantzig ו-Ramser לבעיית שיגור המשאיות, אילוץ שגרם לכך שב-N השלבים הראשונים ניתן היה לחבר יחדיו למסלול אחד רק לקוחות שהדרישה שלהם אינה עולה על $C/2^{N-1}$ (כאשר C הנה יכולת הקיבולת של המשאית), לקוחות אשר נמצאים רחוק אחד מהשני חוברו יחדיו למסלול אחד, וזאת למרות שניתן לראות בברור שמסלול ישיר מהמחסן לאותם הלקוחות יהיה קצר יותר. המשמעות הייתה שהאלגוריתם נותן יותר משקל למילוי המשאיות עד למקסימום האפשרי, מאשר להשיג את המרחק הכולל המינימאלי ([61]). ואכן, התעלמות מאילוץ זה גרמה לאלגוריתם לספק תוצאה טובה יותר. למרות ש-Clarke מ-Wright הראו כי ניתן לשפר את האלגוריתם של Dantzig ו-Ramser, הם הבינו כי אלגוריתם חדש יכול לספק תוצאות טובות יותר, ועבודתם הייתה אלגוריתם חדש אשר כיום מוכר כאלגוריתם Savings ([16],[42],[4],[18]).

מרכז עבודתם של Clarke ו-Wright, שהושפעה מעבודתם של Dantzig ו-Ramser, הוא מציאת אוסף מסלולים עבור צי כלי רכב של ארגון שצריך לספק סחורה ל-n לקוחות שונים, הממוקמים במקומות שונים ממחסן מרכזי אחד, כשסך האורך הכולל של כל המסלולים הוא מינימאלי. כמו כן, בדומה לבעיה שעמדה מול Dantzig ו-Ramser, יכולת הקיבולת של כלי רכב בצי גדולה מהדרישה של לקוח בודד, וכן לא קיימים שני מסלולים אשר מגיעים לאותו לקוח ולא קיים לקוח שאין מסלול אשר מגיע אליו.

Clarke ו-Wright ציינו, שאם קיים בצי כלי רכב בעל קיבולת שגדולה מסך הדרישות של כל הלקוחות, הרי שבמקרה זה אנו עוסקים בבעיית סוכן נוסע.

בדומה לאלגוריתם של Dantzig ו-Ramser ([21]), אלגוריתם Savings פועל גם הוא בשני שלבים, בניית פתרון ראשוני ושיפורו, כאשר ההבדל הוא באופן ביצוע השיפור. כאמור הפתרון ראשוני הוא קבוצה של מסלולים, כשכל אחד מהמסלולים מוקצה לכלי רכב בודד בצי. כל אחד מהמסלולים עובר דרך לקוח אחד בלבד, ואין שני מסלולים אשר עוברים דרך אותו הלקוח. הפתרון הראשוני הוא אוסף המסלולים $P_0P_iP_0$, כאשר $i=\{1,2,\ldots,N\}$ (N הוא מספר הלקוחות).

בשלב השיפור, האלגוריתם מחשב את "החיסכון" המתקבל מחיבור של שני מסלולים. את החיסכון אנו מגדירים כמרחק שבין הקודקוד האחרון במסלול הראשון למחסן פלוס המרחק בין המחסן לקודקוד הראשון במסלול השני פחות המרחק בין הקודקוד האחרון במסלול הראשון

לקודקוד הראשון במסלול השני (לדוגמה, עבור זוג המסלולים הראשוניים $P_0P_iP_0$ ו- $P_0P_jP_0$ החיסכון שווה ל- $S_{ij} = P_iP_0 + P_0P_j - P_iP_j$). את "החיסכון" אנו מחשבים לכל זוגות המסלולים. את זוג המסלולים שקיבל את ערך ה"חיסכון" הגבוה ביותר אנו מחברים יחדיו למסלול חדש יחיד (בתנאי שהמסלול הנ"ל עומד באילוצים השונים של הבעיה, אחרת מחברים את זוג המסלולים שקיבל את ערך ה"חיסכון" השני בגובהו). את חישוב ה"חיסכון" וחיבור המסלולים, יש לבצע כל עוד ישנם זוגות מסלולים שעבורם קיים "חיסכון" חיובי, וחיבור המסלולים הללו עומד באילוצי הבעיה. מכוון שבכל איטרציה אנו מחפשים את המסלולים שחיבורם ייתן את החיסכון הגדול ביותר לאיטרציה הנוכחית, ללא שיקולים נוספים, כגון האם איחוד המסלולים בשלב מאוחר יותר ייתן חיסכון גדול יותר, הרי שאלגוריתם זה הוא אלגוריתם חמדני.

Clarke ו-Wright ציינו במאמרם שבמידה ונמצא יותר מזוג מסלולים אחד אשר ה"חיסכון" המתקבל מחיבורם הוא זהה, אנו בוחרים את אחד מהזוגות באופן אקראי.

המחברים הוסיפו וציינו, שלמרות שהאלגוריתם מציין את הסדר שבו יש לעבור מלקוח ללקוח בכל אחד מהמסלולים, עדיין כדאי עבור כל מסלול ומסלול לפתור את בעיית סוכן נוסע לקבוצת הלקוחות השייכים לאותו המסלול, על-מנת לקבל את סדר המעברים האופטימאלי האמיתי.

בכדי לבדוק את יעילות האלגוריתם שלהם, השתמשו Clarke ו-Wright באותה דוגמה שבה השתמשו Dantzig ו-Ramser בכדי להדגים את האלגוריתם שלהם. התוצאה שהתקבלה אכן הייתה טובה יותר, אבל השיפור היה מזערי. אולם, בשימוש בתסריטי בדיקה אחרים, הצליח אלגוריתם Savings להשיג שיפור של עד 17%.

השיטה שהוצעה ע"י Clarke ו-Wright פשוטה וקלה ליישום, והיא היוותה בסיס למספר חבילות תוכנה בתחום ההובלה וההפצה ([42],[18]). התוצאות המתקבלות משימוש באלגוריתם Savings טובות יותר מאשר אלו המתקבלות מהאלגוריתם של Dantzig ו-Ramser, אך עבור רוב הבעיות הן עדיין רחוקות מהפתרון האופטימאלי.

3.3.2. אלגוריתם Sweep

האלגוריתמים ההיוריסטיים שקדמו לאלגוריתם Sweep ([4],[18],[36]), אשר הוצע ע"י Gillett ו-Miller בשנת 1974, ניסו לפתור את בעיית ניתוב הרכבים כבעיה אחת גדולה, בעוד שאלגוריתם Sweep מחלק את קבוצת הלקוחות למספר תת-קבוצות, ועבור כל תת-קבוצה מוצא את הנתיב האופטימאלי (או הקרוב לאופטימאלי). היתרון בחלוקת קבוצת הלקוחות לתת-קבוצות היינו בזמן ריצה אשר גדל באופן ליניארי כאשר הבעיה (כמות הלקוחות) גדלה, בניגוד לאלגוריתמים אחרים, שבהם זמן הריצה גדל באופן אקספוננציאלי ([36]).

אלגוריתם Sweep, משתמש בנתונים בסיסיים לצורך פעולתו. אנו צריכים לדעת את המיקום של כל לקוח במערכת צירים אוקלידית ביחס למחסן (המחסן נמצא בנקודה $(0,0)$ בעוד שכל לקוח ולקוח נמצא בנקודה (x_i, y_i)). מעבר לכך נתונות הדרישות של כל לקוח ולקוח, יכולת ההובלה של כלי הרכב וכן אילוצים שונים נוספים במידה ויש.

בתחילת עבודת האלגוריתם אנו מחשבים את הקורדינטות הפולאריות של כל לקוח ולקוח ביחס למחסן, שכאמור נמצא בנקודה (0,0). לאחר החישוב הנ"ל, אנו ממיינים את הלקוחות על פי הזווית של הקורדינטה הפולארית, מהקטן לגדול. מיון זה מהווה את הבסיס לחלוקת הלקוחות לקבוצות. את המסלול הראשון אנו מתחילים עם הלקוח שיצא ראשון במיון, אליו אנו מוסיפים את הלקוח שיצא שני וכן הלאה, כאשר אנו סוכמים את הדרישות של כל הלקוחות. אנו נפסיק להוסיף לקוחות לקבוצה במידה והוספת לקוח נוסף תגרום למצב שבו סכום הדרישות גדול מיכולת ההובלה של כלי הרכב של הספק. לאחר שיש בידנו קבוצת לקוחות, אנו נדרשים להפעיל אלגוריתם לפתרון TSP, על מנת למצוא את המסלול הקצר ביותר אשר יוצא מהמחסן, עובר בין כל אותם הלקוחות וחוזר בסוף למחסן. כמו כן, יש לבדוק אם אותו מסלול עומד באילוצים השונים של בעיה, כאשר במידה ולא, אנו מורידים את הלקוח האחרון שהוספנו לקבוצה, מפעילים מחדש את אלגוריתם TSP ובודקים מחדש את התוצאה, עד אשר נקבל קבוצה אשר עומדת באילוצים. אנו ממשיכים ובונים קבוצות חדשות באופן זהה, כאשר כל קבוצה מתחילה מהלקוח בעל הזווית של הקורדינטה הפולארית הקטנה ביותר שלא שייכת עדיין לקבוצה. בסוף התהליך, יש בידינו מספר קבוצות ומספר מסלולים, אשר הושגו בעזרת שימוש ב-TSP.

ניתן להפעיל את אלגוריתם Sweep באופן הפוך, שבו במקום למיין את הלקוחות על פי הזווית של הקורדינטה הפולארית, מהקטן לגדול, אנו ממיינים מהגדול לקטן. המסלולים והקבוצות המתקבלים לאחר שימוש באלגוריתם "הפוך" זה, לרוב יהיו שונים, ולכן עומדת בפני המשתמש האפשרות לבחור בתוצאה הטובה ביותר המתקבלת משני האלגוריתמים הללו.

Gillett ו-Miller ביצעו השוואה בין מספר אלגוריתמים קיימים. הם הריצו את אלגוריתם Sweep כשהם משתמשים בתסריטים אשר שימשו במחקרים קודמים. התוצאות שהוצגו על-ידם הראו שאמנם האלגוריתם לא הצליח להשיג פתרונות טובים יותר מהפתרונות שהיו ידועים עד כה, אך ב-10 מקרים מתוך 12, התוצאות היו זהות לתוצאות הטובות ביותר, ובשני המקרים האחרים, ההבדל היה זניח ביותר. היתרון של האלגוריתם אם כן, הוא אינו בתוצאותיו, כי אם ביכולתו לפתור VRP בעל מספר רב יחסית של לקוחות, וכן בזמן הריצה שלו. כאשר כמות הלקוחות גדלה, זמן הריצה של האלגוריתם גדל באופן ליניארי כל עוד מספר הלקוחות בכל תת-קבוצת לקוחות נשאר קבוע באופן יחסי, כמו כן, זמן הריצה של האלגוריתם גדל ברביעית ביחס לממוצע הלקוחות שנמצאים בכל מסלול, אם מספר הלקוחות במסלול נשאר יחסית קבוע ([36]).

3.3.3. שיטת האשכולות

שיטת האשכולות (Cluster Method) ([18],[4]) הוצגה ע"י Fisher ו-Jaikumar (ולכן היא מוכרת גם בשם האלגוריתם של Fisher ו-Jaikumar). בדומה לאלגוריתם Sweep, גם כאן אנו תחילה מחלקים את הלקוחות לקבוצות, כשלכל קבוצה מוקצה כלי רכב, ולאחר מכן אנו מוצאים את המסלול הקצר ביותר עבור כל אחד מכלי הרכב. לרוב החלוקה לקבוצות נעשת ע"י קיבוץ הלקוחות יחדיו בהתאם למרחק שלהם אחד מהשני, שכל קבוצת לקוחות חייבת לעמוד באילוצי הבעיה. לאחר שחילקנו את הלקוחות לקבוצות, יש לפתור בעיית סוכן נוסע עבור כל אחת מהקבוצות, בעזרת אלגוריתמים קיימים שונים.

כאמור, בשלב הראשון נעשת חלוקה של הלקוחות לקבוצות, כאשר חלוקת הלקוחות לקבוצות נעשת בעזרת פתרון בעיית השמה כללית (Generalized Assignment Problem – GAP). בבעיית השמה כללית אנו משייכים n פעולות ל-m סוכנים. בצורה פורמאלית, בעיית השמה כללית מנוסחת באופן הבא ([30]):

אנו מסמנים ב-J את קבוצת הפעולות,

$$J = \{1, 2, ..., n\} \tag{1.12}$$

וכן, אנו מסמנים ב-I את קבוצת הסוכנים.

$$I = \{1, 2, ..., m\} \tag{1.13}$$

c_{ij} מכיל את הערך שסוכן i יבצע את פעולה j. b_i הוא כמות המשאבים העומדים לרשות סוכן i. a_{ij} הוא כמות המשאבים הנדרשת על-ידי סוכן i לבצע את פעולה j. x_{ij} היינו משתנה החלטה, המציין אם סוכן i יבצע את פעולה j ($x_{ij} = 1$), או לא ($x_{ij} = 0$).

פונקציה המטרה היא:

$$\min \quad Z = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \tag{1.14}$$

תחת האילוצים:

$$\sum_{i=1}^{m} x_{ij} = 1 \quad \forall j \in J \tag{1.15}$$

$$\sum_{j=1}^{n} a_{ij} x_{ij} < b_i \quad \forall i \in I \tag{1.16}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \tag{1.17}$$

מכוון ש-GPA היא בעיה NP-Hard ([18]), פותרים אותה לרוב בעזרת רלקצציה לגראנג'יאנית, כאשר עבור בעיית ניתוב הרכבים זה נעשה באופן הבא:

עבור קבוצה קבועה של k רכבים אנו בוחרים j_k נקודות אשר יהוו נקודות ראשוניות לקבוצות שלנו. אנו מחשבים את העלות של הקצאת לקוח i לקבוצה k על פי הנוסחה

$$d_{ij_k} = \min\left(c_{1i} + c_{ij_k} + c_{j_k 1}, c_{1 j_k} + c_{j_k i} + c_{i0}\right) - \left(c_{0 j_k} + c_{j_k 0}\right)$$

לאחר שחישבנו את d_{ij_k} יש באפשרותנו לפתור את בעיית ההשמה הכללית. שלאחריה כאמור יש ברשותנו k קבוצות, עבור k כלי רכב. עבור כל אחת מהקבוצות יש לפתור בעיית סוכן נוסע.

3.4. מטה-היוריסטיקות

מטה-היוריסטיקות, בניגוד להיוריסטיקות, מבצעות חיפוש עמוק יותר במרחב הפתרונות, ומאפשרות ביצוע פעולות כדוגמת איחוד פתרונות, על מנת להשיג פתרון טוב יותר. אומנם הפתרונות המתקבלים משימוש במטה-היוריסטיקות קרובים יותר לפתרון האופטימאלי, אבל

לרוב דיוק זה מושג על חשבון מהירות ופשטות האלגוריתם ([18]). קיימים מספר רב של מטה-היוריסטיקות המשמשות לפתרון בעיות ניתוב רכבים, אנו נסקור מספר קטן של מטה-היוריסטיקות פופולאריות לפתרון VRP.

3.4.1. מערכות נמלים - Ant Systems

Ant system הוא אלגוריתם היוריסטי המיועד לפתרון בעיות אופטימיזציה קומבינאטוריות ([26]). Ant systems הושפע מהאופן שבו נמלים מתקשרות בניהן בזמן חיפש אוכל. תצפיות על אופן בו נמלים מחפשות אחר אוכל, הראו שנמלים מסמנות את המסלול למזון בו הם עברו ע"י שימוש בפרומון (חומר ריח), כאשר כמות הפרומון תלויה באורך המסלול ואיכות האוכל. נמלים אחרות נמשכות לפרומון, ועוקבות אחריו לאורך המסלול, כאשר אף הן מפרישות פרומון לאורך המסלול, אשר מושך בתורו נמלים נוספות וחוזר חלילה. בסופו של התהליך אנו מקבלים מספר מסלולים, כאשר אלו שהם עשירים יותר באוכל וקרובים יותר לקן הנמלים מבקרים בהם בתדירות גדולה יותר. אלגוריתמים מתחום ה-Ant systems פועלים באופן דומה. "נמלים" מלאכותיות מחפשות את הפתרון האופטימאלי במרחב הפתרונות, כאשר פונקצית המטרה שקולה לאיכות המזון, ומסלולים שעברנו דרכם נשמרים בזיכרון, כשם שפרומון משמש לסימון וחזרה על המסלול. לתהליך חיפוש הפתרונות מאפיינים אקראיים, המעבר מצומת אחד לצומת שני היינו אקראי ביסודו, אך הוא מושפע מעצמת הפרומונים הקיימים בקשתות השונות היוצאות מהצומת הראשון לשאר הצמתים.

Dorigo, Manziezzo ו-Colorni ([26]) הציגו אלגוריתם שמבוסס על Ant systems לפתרון בעיית הסוכן נוסע. אלגוריתם מתחיל כאשר ישנה "נמלה" בכל אחד מקדקודי הגרף. בכל איטרציה, ה"נמלים" בוחרות את הקודקוד אליו הן נעות בהסתברות שהיא פונקציה של המרחק אל הקודקוד וכמות ה"פרומון" שקיימת על הקשת שמובילה לאותו הקודקוד. בכדי שיתקבלו מסלולים שעומדים בתנאי בעיית הסוכן נוסע, ל"נמלים" אסור לחזור לקדקודים שהן ביקרו בהם לפני כן. בכל קשת שהן עוברות בהן, ה"נמלים" משאירים כמות מסוימת של "פרומון", כמות ה"פרומון" תלויה בכמות ה"פרומון" הקיימת כבר על הקשת וכן באורך הקשת. בנוסף, קיימת פונקציה המתארת את התאדות ה"הפרומון" לאורך זמן. פונקציה זו משמשת בכל איטרציה על מנת להפחית את כמות ה"הפרומונים" בכל קשת.

Dorigo, Manziezzo ו-Colorni ([26]) מימשו את האלגוריתם שהציעו ואף השוו את התוצאות שהתקבלו ממנו אל מול תוצאות של מספר אלגוריתמים ידועים נוספים. האלגוריתם אומנם סיפק את התוצאה הטובה אך הוא לא הצליח לשפרה בהשוואה לתוצאות שהתקבלו מהאלגוריתמים האחרים.

השימוש הראשון ב-Ant System לפתרון VRP הוצג ע"י Bullnheimer, Hartl ו-Strauss בשנת 1997, שהשתמשו בה לפתרון CVRP ([12]). TSP ו-VRP הנן, כאמור, בעיות הקרובות אחת לשניה, בעקבות זה הושפעו Bullnheimer, Hartl ו-Strauss בעבודתם מהאלגוריתם שהציגו Dorigo, Manziezzo ו-Colorni, כאשר ההבדל העיקרי בין האלגוריתמים הוא בכך, שכאשר "נמלה" עוברת לקודקוד אשר יוצר מסלול אשר לא עומד באילוצי הבעיה (למשל יגרום לכך שהדרישות יהיו גדולות מקיבולת כלי הרכב), אנו נשנה את הבחירה שלנו לקודקוד המחסן.

Bullnheimer, Hartl ו-Strauss בחנו את האלגוריתם שפיתח בעזרת 14 תסריטי בדיקה, שהכילו בין 50 ל-199 לקוחות, בנוסף למחסן. אומנם האלגוריתם שהוצג על-ידם השיג פתרונות טובים, אך הוא לא הצליח לשפר את הפתרונות הטובים ביותר הידועים לאותם 14 תסריטי הבדיקה.

Bell ו-Mullen ([5]) הציגו אף הם אלגוריתם מבוסס Ant systems לפתרון VRP. האלגוריתם שהציעו Bell ו-Mullen פועל באופן דומה אלגוריתם של Dorigo, Manziezzo ו-Colorni וכן לאלגוריתם של Bullnheimer, Hartl ו-Strauss. Bullnheimer, Hartl ו-Strauss טענו שלפתרון VRP שיש צורך בכמות "נמלים" הזהה לכמות הקדקודים שקיימת בגרף, וכן שיש למקם את כל אחת מה"נמלים" בקודקוד אחר של הגרף. לעומתם, Bell ו-Mullen הציבו את כל ה"נמלים" שלהם במחסן כנקודת המוצא. בנוסף למסלולים אשר ניבנו ע"י ה"נמלים" בכל אחת מהאיטרציות, Bell ו-Muller מציעים להוסיף פרוצדורות שונות לשיפור המסלולים המתקבלים. לדוגמא, ע"י שימוש בהיוריסטיקת 2-opt ניתן לבדוק פרמוטציות שונות של כל אחד מהמסלולים שנבנו ע"י ה"נמלים" ולבחור את המסלול הטוב יותר מבין כל הפרמוטציות בתור המסלול שנבנה ע"י ה"נמלה". שינוי נוסף שהכניסו Bell ו-Muller לאלגוריתם הוא שימוש במספר Ant systems בו זמנית, כאשר כל מערכת אחראית בנפרד על מציאת מסלול בודד לכלי רכב יחיד. שימוש במספר מערכות מתאפשר כאשר כל מערכת משתמשת בסוג שונה של "פרומון" שאינו משפיע על שאר המערכות. ההשראה לשימוש במספר מערכות היא מהעובדה שבכל מושבת נמלים ניתן למצוא קבוצות שונות של נמלים, כשכל קבוצה עוסקת בתפקיד שונה במושבה. השימוש במספר מערכות נעשה יעיל יותר ככל שמספר כלי הרכב הנדרשים לפתרון גדל.

לאחר ש-Bell ו-Mullen בחנו את האלגוריתם שלהם, הן ע"י שימוש במערכת בודדת והן ע"י שימוש במספר מערכות, עבור בעיות קטנות, הם הגיעו לתוצאות שלא היו רחוקות יותר מ-1% מהתוצאה האופטימאלית הידועה. אולם, ככל שהבעיה גדלה, כך התוצאה שהתקבלה מאלגוריתם הייתה רחוקה יותר מהפתרון האופטימאלי הידוע, כאשר שימוש במספר מערכות סיפק תוצאה טובה יותר מאשר שימוש במערכת אחת. תוצאות אלו דומות לתוצאות שהתקבלו במחקרים קודמים, אשר מצביעים על כך ששימוש ב-Ant system יכול לשמש לפתרון VRP, אך אין באפשרותו לספק פתרונות טובים יותר מאלגוריתמים קיימים אחרים .

3.4.2. אלגוריתמים גנטיים

בדומה לאלגוריתמים המבוססים על Ant system, אשר פועלים באופן דומה לפעולתן של נמלים בטבע, אלגוריתמים גנטיים הנם אלגוריתמים שמחקים את מנגנון הברירה הטבעית הקיים בטבע על מנת לפתור בעיות אופטימיזציה קומבינאטוריות שונות. הרעיון והאלגוריתמים הראשונים פותחו עוד בשנות ה-70. אלגוריתמים גנטיים יוצרים קהילה של פרטים (אוסף ראשוני של פתרונות אפשריים). כל פרט מדורג, על מנת למצוא את הפרטים הטובים ביותר. לאחר מכן, עורכים זיווג (מיזוג) בין פרטים אלו על מנת ליצור קהילה חדשה, טובה במקצת מקודמתה, ומעבירים אותה את אותו התהליך. לאחר מספר איטרציות של הפעולה, הפרטים ישתפרו דרמטית ויציגו פתרון אפשרי או כמעט אפשרי לבעיה הנתונה (לסקירה נוספת ראה [37]).

Baker ו-Ayechew ([3]) הציעו אלגוריתם גנטי לפתרון VRP. את הקבוצה הראשונית של הפתרונות מציעים Baker ו-Ayechew לבנות בעזרת שימוש באלגוריתם Sweep או בעזרת שימוש באלגוריתם של Fisher ו-Jaikumar (שני האלגוריתמים פועלים כאמור בשני שלבים – חלוקה לקבוצות ופתרון TSP, כאשר אנו משתמשים רק בשלב הראשון של החלוקה לקבוצות).
שני פתרונות נבחרים באקראי, כאשר הפתרון טוב יותר הוא פתרון האב. אנו חוזרים ובוחרים אקראית שני פתרונות נוספים, כאשר הפתרון טוב יותר ישמש כאם. יצירת פתרון חדש מקבלת ע"י בחירה אקראית של נקודה בפתרונות ההורים ובניית פתרון חדש שחלקו הראשון נלקח מהפתרון האב וחלקו השני נלקח מפתרון האם. פתרון נוסף מורכב כך שחלקו הראשון נלקח מפתרון האם וחלקו השני נלקח מפתרון האב. במידה ואחד מהפתרונות החדשים זהה לפתרון שכבר קיים באוכלוסייה, אין אנו מתייחסים אליו. Baker ו-Ayechew מצאו שניתן לשפר את התוצאות ע"י הכנסת מוטציות לפתרונות החדשים. הכנסת מוטציות נעשת ע"י בחירה אקראית של שני קדקודים בכל פתרון והחלפתם בערך אקראי אחר. בזמן שפתרונות חדשים נכנסים לאוכלוסייה, אנו מסירים מהאוכלוסייה את הפתרונות הגרועים ביותר עד כה, כך שגודל האוכלוסייה נשאר קבוע.

Baker ו-Ayechew בדקו את האלגוריתם הגנטי שלהם מול 14 תסריטי בדיקה נפוצים שתוצאותיהם ידועים. התוצאות שהתקבלו הראו שהאלגוריתם מספק תוצאות טובות שקרובות מאד לפתרונות האופטימליים הידועים (בממוצע סטייה של 2.5% מהפתרון האופטימאלי). כמו כן, באף אחד מהמקרים לא הצליח האלגוריתם להשיג פתרון יותר טוב מהפתרון הידוע עד כה.

Berger ו-Barkaoui הציגו פתרון ל-CVRP מבוסס על אלגוריתם גנטי ([7]). באלגוריתם זה נעשה שימוש בשתי אוכלוסיות של פתרונות, כשכל אחת מהאוכלוסיות משתפרת בנפרד, כשהמטרה היא הקטנת המרחק הכולל של הפתרונות, וכן, בכל דור (שלב של שיפור) מספר פרטים מכל אוכלוסייה מוחלפים עם האוכלוסייה השניה. בכל פעם שמתקבל פתרון שהוא הפתרון הכי טוב עד כה, מתבצע על הפתרון הזה שלב נוסף של עיבוד שמטרתו לשפר את אותו הפתרון עוד יותר. לאחר ששתי האוכלוסיות הראשונות נבנו באופן אקראי, מתבצע שלב השיפור והחלפת הפרטים בין האוכלוסיות עד לנקודה שבה מספר הפרטים הזהים בין שתי האוכלוסיות נמוך ממספר קבוע. התכנסות האלגוריתם לפתרון מתקבלת כאשר הפתרונות משתפרים בפחות מאחוז מסוים לאורך מספר קבוע של שלבי שיפור.

Berger ו-Barkaoui הציגו כלים חדשים לבניית פתרון חדש מזוג פתרונות הורים קיימים. בכל אחד מהפתרונות ההורים מחפשים את האזורים השונים שבהם מופיע פתרון טוב, כאשר שילוב של שני פתרונות טובים (אחד מכל הורה) עשוי ליצור פתרון טוב יותר. גם באופן שבו מבוצעות מוטציות יש שינוי. Berger ו-Barkaoui מציעים לבצע חיפוש בין הפתרונות על מנת למצוא מספר פתרונות דומים, ולבצע מוטציות על בסיס אותם פתרונות. מוטציה נוספת שהוצעה היא להחליף לקוח אחד או יותר בין מספר מסלולים. מוטציה זו משלימה את המוטציה הקודמת, ומטרתה לבצע שינויים שהמוטציה הראשונה לא מסוגלת לבצע. מוטציה שלישית מבוססת על שינוי סדר הלקוחות בכל פתרון, כאשר המטרה היא למצוא מסלול בעל מרחק קצר יותר בתוך הפתרון עצמו. מוטציה זו מבוצעת בכל פעם שנמצא פתרון שהוא פתרון הכי טוב עד כה, כפי שתואר קודם לכן.

Berger ו-Barkaoui בחנו את האלגוריתם שלהם בעזרת מספר תסריטי בדיקה, וכן ערכו השוואה בין תוצאות האלגוריתם שלהם לתוצאות שהתקבלו ממספר אלגוריתמים נפוצים אחרים. האלגוריתם לא הראה יתרון בולט על פני אלגוריתמים קיימים, אך הוא בהחלט שקול להם בטיב התוצאות שהוא מספק. בנוסף, האלגוריתם הוכח כאלגוריתם שדרישותיו החישוביות נמוכות מאשר אלגוריתמים אחרים.

3.4.3. חיפוש טאבו - Tabu Search

אחת מהמטה-היוריסטיקות היעילות ביותר לפתרון VRP היא Tabu Search. הרעיון העומד מאחורי Tabu Search הוא ביצוע חיפוש מקומי ע״י מעבר, באיטרציה t, מפתרון x_t לפתרון טוב יותר x_{t+1} שקיים בקרבתו. בניגוד להיוריסטיקות רגילות, ב-Tabu Search הפתרון יכול ״להתקלקל״ במהלך האיטרציות השונות, וכתוצאה מכך אנו יכולים להיקלע לאיטרציות מעגליות (מעבר מפתרון א׳ לפתרון ב׳, ובחזרה מפתרון ב׳ לפתרון א׳ וחוזר חלילה). בכדי להימנע ממעגלים, פתרונות הכוללים חלק מהמאפיינים של פתרונות קודמים מוגדרים כאסורים למשך מספר איטרציות. הגדרה של פתרון כ״אסור״ יכולה להתבטל אם קריטריון מסוים מתקיים, לדוגמה כאשר פתרון ״אסור״ הוא טוב יותר מכל פתרון שהוצע מוקדם יותר ([18]).

Gendreau, Laporte, Musaraganyi ו-Taillard הציעו אלגוריתם המבוסס על Tabu search לפתרון בעיית ניתוב רכבים בעלת צי רכבים הטרוגני (הרחבה של בעיית ניתוב הרכבים שבה לארגון יש מספר כלי רכב שונים בקיבולת שלהם) ([34]). האלגוריתם עושה שימוש באלגוריתם GENIUS (אלגוריתם הכנסה כללי שפותח עבור פתרון TSP), על מנת לקבל אוסף של פתרונות התחלתיים. Tabu search עצמו מתבצע על אוסף הפתרונות הראשוניים שקיבלנו. אנו מחפשים בכל קבוצה את הפתרונות הטובים ביותר, ומנסים לשפרם, בין היתר ע״י החלפת קדקודים בין הפתרונות של שני הקבוצות. אנו ממשיכים את הפעולה, כאשר כל פעם אנו מתקדמים ומתקרבים לפתרון טוב יותר.

Gendreau, Laporte, Musaraganyi ו-Taillard בחנו את האלגוריתם שלהם מול פתרונות קיימים. בכל המקרים הם הצליחו להשיג פתרונות שקרובים מאד לפתרון האופטימאלי הידוע, ואף במקרים מסוימים הצליחו למצוא פתרונות אופטימאליים חדשים.

Gendreau, Hertz ו-Laporte הציעו גם הם אלגוריתם מבוסס על Tabu search לפתרון בעיית ניתוב רכבים ([33]). אנו בוחרים באופן אקראי קודקוד כלשהו, v_i. עם קבוצת הקדקודים $(v_0, v_i, v_{i+1}, ..., v_n, v_1, ..., v_{i-1})$ ומתחילים לבנות מסלולים בעזרת אלגוריתם GENIUS. אנו בונים לכל היותר $\bar{m}$ ($\bar{m}$ הוא המספר המקסימאלי של רכבים) מסלולים היוצאים מקודקוד v_0 כך שהמסלול הראשון יכלול את כל הקדקודים, לא כולל, v_i, כאשר הכללתו מפר את אילוצי הקיבולת. המסלול השני מתחיל מקודקוד v_i, וכולל את כל הקדקודים, עד שאנו מגיעים שוב לקודקוד שמפר את אילוצי הקיבולת. באופן דומה אנו בונים את שאר המסלולים. המסלול האחרון יכלול תמיד את כל הקדקודים שנשארו לא משויכים למסלול בנקודה זו. אוסף המסלולים הזה הוא פתרון, s, בין אם הוא עומד באילוצי הבעיה או לא. אנו מבצעים חיפוש בין

התוצאות, בכוונה למצוא אפשרות לשיפור התוצאות ע"י העברה של קדקודים מפתרון אחד לפתרון שני. אם לא היה שיפור במשך מספר קבוע של איטרציות, אנו מפסיקים את התהליך.

Gendreau, Hertz ו-Laporte בחרנו את האלגוריתם שלהם על 14 תסריטי בדיקה שקיימים בספרות. תסריטי הבדיקה כללו בעיות שהכילו בין 50 ל-199 קדקודים, בנוסף לקודקוד המחסן. בכל המקרים, התוצאות שהתקבלו היו קרובות מאד לתוצאות האופטימאליות שהתקבלו, ואף במספר מקרים, האלגוריתם הצליח למצוא פתרון אופטימאלי חדש. אלגוריתמים המבוססים על Tabu search הנם אלגוריתמים בעלי פוטנציאל גבוה למציאת פתרונות טובים ואף פתרונות אופטימאליים.

כפי שניתן לראות משני האלגוריתמים שהוצגו כאן. ההבדל העיקרי בין האלגוריתמים הוא אופן בניית הפתרונות ההתחלתיים, וכן ישנו שוני באופן החיפוש בין השכנים לשינויים פוטנציאלים בפתרונות.

על בסיס Tabu Search ניבנו הרחבות שונות, לדוגמה Granular Tabu Search. Granular Tabu Search מבוסס על העובדה שהסיכוי שקשתות בעלות משקל (מרחק) גבוה יהיו שייכות לפתרון האופטימאלי הוא נמוך. ולכן, אם נסיר קשתות אלו, אנו נמנע ממספר פתרונות שהסיכוי שהם פתרונות אופטימאליים קטן ביותר ([18]). שימוש בגישה זו משאיר אותנו עם בעיה המכילה 10% עד 20% ממספר הקשתות המקורי, שעליה אנו מפעילים Tabu search מקורי.

3.5. הרחבות לבעיית ניתוב רכבים

במקביל להתפתחות ההיוריסטיקות והמטה-היוריסטיקות, שמאפשרים לקבל פתרונות קרובים לפתרון האופטימאלי, עסקו מספר חוקרים בפיתוח הרחבות לבעיה הבסיסית של ניתוב רכבים. המטרה שעמדה מול אותם חוקרים הייתה ליצור מודלים קרוב יותר ליישומים של העולם האמיתי שבו קיימים מספר רב של אילוצים.

בסעיף זה אנו נבחן מספר הרחבות נפוצות:

- ניתוב רכבים עם חלוקת הובלות
- ניתוב רכבים עם חלונות זמן
- ניתוב רכבים ממספר מחסנים
- ניתוב רכבים תלוי זמן
- ניתוב רכבים סטוכסטיים

3.5.1. ניתוב רכבים עם חלוקת הובלות

בבעיית ניתוב רכבים קלאסית השרות של כל לקוח ולקוח נעשה ע"י כלי רכב אחד בלבד. מספר חוקרים ניסו לפתור גרסה מוחלשת של בעיית הקצאת המסלולים, שבה אילוץ זה לא קיים. גרסה זו של הבעיה נקראת בעיית ניתוב רכבים עם חלוקת הובלות (Split Delivery Vehicle Routing Problem – SDVRP). היעדרותו של אילוץ זה מהווה יתרון בפתרון בעיה, במידה וישנם לקוחות שהדרישות שלהם קרובות, או אף גדולות, מיכולת ההובלה של כלי יחיד בצי.

שימוש באלגוריתם Savings לפתרון בעיית ניתוב רכבים עם חלוקת הובלות הוצע ע"י Burrows ([13], [42]). השינוי שהוצע ע"י Burrows פשוט, חלוקת דרישות כל הלקוחות למספר דרישות קטנות יותר, ואז הפעלת אלגוריתם Savings הרגיל.

Dror ו-Trudeau הציגו אלגוריתם המבוסס על שני שלבים לפתרון SDVRP ([15],[28]). בשלב הראשון אנו פותרים VRP, ועי"כ למעשה בונים מסלולים התחלתיים. פתרון ה-VRP נעשה ע"י שימוש בווריאציה של אלגוריתם Savings, אשר מספקת מסלולים אשר אינם בהכרח עומדים בכל האילוצים (למשל, סך הדרישה במסלול מסוים יכולה להיות גדולה מקיבולת כלי הרכב המשויך לאותו המסלול). השלב השני כולל החלפת מסלולים ע"י שימוש ב-k-split. K-split מאפשר חלוקת דרישה של לקוח i בין k מסלולים, בתנאי שסך הקיבולת הפנויה בכלי הרכב שמוקצים עבור אותם k מסלולים גדולה מדרישת לקוח i, וכן, החיסכון המתקבל (מבחינת מרחק) מחלוקת הדרישה בין אותם k מסלולים הוא חיובי. את החיסכון המתקבל מחלוקה בין k מסלולים ניתן לחשב בעזרת הנוסחה $SAV_k(p)=\sum_{t=1}^{k}\left(C_{i_t j_t}-C_{i_t p}-C_{pj_t}\right)+C_{b_p p}+C_{pa_p}-C_{b_p a_p}$, כאשר הנקודות i_k ו- j_k הן שתי נקודות עוקבות במסלול k, הנקודה a_p היא הנקודה הנמצאת לפני הנקודה p (מיקום לקוח i), והנקודה b_p נמצאת אחרי הנקודה p.

בשלב השני נעשה שימוש בטכניקה של הוספת מסלולים. ישנם מיקרים שבהם הוספת מסלולים, שמונעת את הצורך בחלוקת קודקוד לשניים או יותר מסלולים אחרים, יכולה להקטין את המרחק הכולל. הוספת מסלולים היא פעולה הופכית ל-k-split. בנוסף נעשה שימוש בהחלפת קדקודים בין מסלולים וכן רוטינות לשיפור מסלולים קיימים. אנו ממשיכים להשתמש בכל אחד מטכניקות השיפור עד שלא מושג שיפור נוסף בתוצאה.

Dror ו-Trudeau הראו כי קיים יתרון בחלוקת הובלות ([28],[42]). כאשר דרישת הלקוח מהווה בין 70% ל-90% מיכולת ההובלה של כלי הרכב, הושג שיפור של יותר מ-10% במרחק הנסיעה הכולל שהושג לעומת שימוש ב-VRP סטנדרטי.

Mullaseril, Dror ו-Leung הציעו פתרון מבוסס על SDVRP לפתרון בעיית הפצת מזון במשק חי ([15],[54]). בבעיית הפצת מזון במשק חי אנו צריכים להעביר סוגים שונים של מזון, מנקודת הכנת המזון, לנקודות שונות במשק על מנת להאכיל את החיות השונות. לרשות המשק עומדים שישה רכבים, כאשר בכל אחד מכלי הרכב שמשמשים להפצת המזון לא ניתן להעביר יותר מסוג אחד של מזון בו זמנית. המטרה היא לצמצם את כמות כלי הרכב שידרשו לצורך הפצת המזון וכל לצמצם את המרחק הכולל הנדרש עבורם. האלגוריתם לפתרון הבעיה עובד ב-4 שלבים. השלב הראשון הוא יצירת פתרון ללא חלוקות (כאשר הפתרון עומד בתנאי הבעיה). בשלב השני משפרים את הפתרון ע"י שימוש בהחלפת קשתות. בשלב השלישי מייצרים חלוקת מסלולים ע"י שימוש ב-k-split, ובשלב האחרון נעשה שימוש בהוספת מסלולים על מנת לשפר את הפתרון. בכל אחד מהשלבים אנו מנסים להקטין את המרחק הכולל של המסלולים.

שתי היוריסטיקות משמשות לבניית הפתרון הראשוני. ההיוריסטיקה הראשונה היא סריקת מסלולים. בהיוריסטיקה זו נבנים מסלולים שעומדים באילוצים, אחד אחרי השני. לבניית

המסלולים אנו מתחילים במחסן ומוסיפים למסלול קשת המוביל לקודקוד הקרוב ביותר, תוך עמידה באילוצים, עד שאין אפשרות להוסיף קשתות יותר ואז אנו חוזרים למחסן. ההיוריסטיקה השניה היא augment-merge. בשלב הראשון של אלגוריתם זה, כל קשת שייכת למסלול אחר. בשלב השני, אנו מאחדים מסלולים על מנת ליצור חיסכון, תוך שמירה על אילוצי הבעיה. אנו חוזרים על שלב האיחוד עד שלא ניתן לאחד יותר מסלולים.

העברת קשתות בין מסלולים, משמשת אותנו בשלב השני של ההיוריסטיקה, והיא דומה במשמעותה להחלפת הקדקודים שתוארו אצל Dror ו-Trudeau, כך גם לגבי k-split, המשמש בשלב השלישי והוספת המסלולים המשמש בשלב הרביעי.

מימוש האלגוריתם ובדיקתו הראו שהיוריסטיקה זו מסוגלת להוריד את המרחק הכולל באופן ממשי, בהשוואה להיוריסטיקות קיימות אחרות. ב-80% מהמקרים הנבדקים הצליחה ההיוריסטיקה זו להוריד את המרחק הכולל בעד 10% ([54],[15]).

Chen, Golden ו-Wasil ([15]) הציגו סקירה של מספר בעיות SDVRP ומספר היוריסטיקות ששימשו לפתרונן. בנוסף הציעו Chen, Golden ו-Wasil אלגוריתם נוסף לפתרון SDVRP. השלב הראשון שלהם בפתרון הוא בניית מסלולים ראשוניים ע״י שימוש באלגוריתם לפתרון CVRP, למשל אלגוריתם Savings. עבור בכל מסלול בפתרון הראשוני שהתקבל אנו מתייחסים לנקודה או שתיים מנקודות הקצה שבמסלול ואל c (פרמטר כלשהו) הנקודות השכנות הקרובות אליהן. נקודות אלו יכולות לחלוק את הדרישות שלהן עם השכנים שלהן, כך שיתכנו שלושה מצבים, הנקודות נשארות במסלולן המקורי, הנקודות עברו למסלולים אחרים או שהנקודות חולקות מסלולים שונים כולל את המסלול המקורי. ההחלטה האם להעביר נקודה מסוימת מהמסלול המקורי או לא נעשת ע״י מציאת המסלולים ששימוש בהם ייתן את החיסכון המקסימאלי. מציאת אותם מסלולים נעשת ע״י שימוש בתכנות מעורב בשלמים (mixed integer programming). בדיקה חישובית הראתה שתוצאות האלגוריתם השיג היו נמוכות מתוצאות שהושגו ע״י אלגוריתמים אחרים, אם כי עדיין השיגו תוצאות טובות.

3.5.2. ניתוב רכבים עם חלונות זמן

בעיית ניתוב רכבים עם אילוצי חלונות זמן (Vehicle Routing Problem with Time Windows – VRPTW) הנה הרחבה של ה-VRP הבסיסי שבה נוסף אילוץ חדש, שלפיו יש לספק ללקוח את הסחורה בחלון זמן מסוים. כל חלון זמן מורכב לרוב משני זמנים, a_i ו-b_i. אנו מעוניינים שהזמן שבו נשרת את לקוח i, t_i, יענה על האילוץ $a_i \le t_i \le b_i$. קיימות וריאציות של הבעיה המאפשרות מספר חלונות זמן עבור כל לקוח, ואילו וריאציות אחרות מאפשרות רק חלון זמן אחד. שתיים מבין הווריאציות העיקריות הן VRPTW שבהן חלונות הזמן ״קשיחים״ וחייבים לעמוד בהם, ו-VRPTW שבהן חלונות הזמן הם ״גמישים״. במקרה של חלון זמן ״קשיח״, יכול כלי הרכב להגיע לפני תחילת חלון הזמן, ולחכות במקום עד לזמן המתאים. ואילו חלון זמן ״גמיש״ משמעו שניתן לחרוג מחלון הזמן, כאשר חריגה כזו גוררת אחריה קנס כלשהו.

Solomon היה הראשון שהציג הכללות של מספר היוריסטיקות לפתרון VRP עם חלונות זמן "קשיחים" ([57]). אלגוריתם Savings, כאמור, מתחיל עם פתרון התחלתי שבו יוצרים מספר מסלולים, כאשר כל מסלול יוצא מהמחסן, עובר דרך לקוח אחד וחוזר למחסן. מספר המסלולים ההתחלתיים זהה למספר הלקוחות. כדי להתאים את אלגוריתם Saving ל-VRPTW, Solomon מוסיף נתון נוסף לכל מסלול וזה זמן היציאה מהמחסן, כך שכל מסלול מותאם בצורה כזו שכלי הרכב יגיע ללקוח בחלון הזמן המתאים. בכדי לבחור איזה מסלולים אנו צריכים לחבר, אנו כאמור מחשבים את החיסכון המתקבל מחיבור שני מסלולים ובוחרים את שני המסלולים שהחיסכון עבורם הוא הגדול ביותר, בתנאי שזמן היציאה של המסלול השני גדול מזמן היציאה של המסלול הראשון בתוספת זמן השהייה במסלול זה. המסלולים שמתקבלים על סמך אלגוריתם זה הנם מסלולים שישנם בהם זמני המתנה גבוהים בין לקוח ללקוח ([57]).

אלגוריתם נוסף שזה לטיפול ע"י Solomon הוא אלגוריתם השכן הקרוב. אלגוריתם חמדני זה בונה מסלולים ע"י חיפוש הקודקוד הקרוב ביותר אשר עומד באילוצים השונים, כולל אילוצי חלונות הזמן. בכל פעם שלא נמצא קודקוד כזה, אנו מתחילים מסלול חדש.

אלגוריתם Sweep אף הוא הותאם לפתרון VRPTW. אלגוריתם Sweep עובד בשני שלבים, שלב בניית קבוצות לקוחות ושלב בניית המסלולים. שלב בניית המסלולים שונה כך שהמסלולים נבנו על סמך חלונות הזמן, כאשר לקוחות שלא נכנסו למסלול הוצאו מהקבוצה. לאחר בניית המסלול, מבוצע שלב החלוקה לקבוצות פעם נוספת, ולאחריו בניה של מסלול נוסף, כך עד שכל הלקוחות משויכים לקבוצות.

האלגוריתם שנתן את התוצאות הטובות ביותר הוא אלגוריתם ההכנסה (Insertion Heuristic). באלגוריתם ההכנסה אנו בונים מסלול, כאשר בכל פעם אנו מחשבים היכן במסלול יש להכניס את כל אחד מהלקוחות, בניית המסלול מפסיקה כאשר הוספת לקוח מפרה את תנאי הבעיה. בכדי לבדוק אם הוספת לקוח למסלול לא מפרה את תנאי חלונות הזמן, ניתן לבדוק את כל הלקוחות לאורך המסלול. Solomon הראה שניתן להשתמש בשיטה דומה לשיטה שבה השתמש באלגוריתם ה-Saving בכדי להחליט אם ניתן להכניס לקוח מסוים לנקודה מסוימת במסלול. שיטה זו הנה יעילה יותר מבחינת זמני ביצוע.

בנוסף לאלגוריתמים שהציע, פיתח Solomon קבוצה של בעיות לצורכי בדיקה, שכל אחת מהן הכילה 100 לקוחות, אשר בהמשך שימשה חוקרים אחרים צרכי השוואה. את קבוצת הבעיות שפיתח סלומון ניתן לחלק ל-6 תת-קבוצות. תת-הקבוצות R1 ו-R2 הם בעלי התפלגות אחידה של הלקוחות, תת-קבוצות C1 ו-C2 הם בעלי מקבצים של לקוחות ותת-הקבוצות RC1 ו-RC2 הם שילוב של התפלגות אחידה ומקבצים של לקוחות. בנוסף, לתת-הקבוצות R1, C1 ו-RC1 חלונות הזמן הם קצרים, בעוד של לתת-קבוצות R2, C2 ו-RC2 חלונות זמן ארוכים, דבר שמאפשר לשרת אותם עם כלי רכב אחד. בעזרת קבוצת בעיות אלו, מצא סלומון שהיוריסטיקת Insertion מספקת את התוצאות הטובות ביותר ([42],[57]).

Desrochers, Desrosiers ו-Solomon הציעו אלגוריתם אופטימיזציה לפתרון VRPTW ([23]). הפתרון שלהם מתבסס על חלוקה לקבוצות (set partitioning). מתוך קבוצת כל המסלולים האפשריים בגרף, נבחרים מסלולים כך שסך האורך של המסלולים הוא מינימאלי, אין שני

מסלולים אשר מכילים את אותו הקודקוד (פרט לקודקוד המחסן) וכן כל המסלולים יחד כוללים את כל קדקודי הגרף. את בעיית החלוקה לקבוצות לא ניתן לפתור באופן ישיר בגלל גודלה, ולכן Desrochers, Desrosiers ו-Solomon משתמשים בשיטת בניית עמודות (column generation) על מנת ליצור גרסה פשוטה יותר של הבעיה. בשיטה זו הבעיה מחולקת שני חלקים, הבעיה העיקרית והבעיה המשנית, כאשר בבעיה המשנית מרוכזים רוב הגורמים שיכולים להשפיע מבחינת אופטימיזציה, ואשר בה מושקע רוב המאמץ לפתרון. את פתרון הבעיה המשנית פותרים Desrochers, Desrosiers ו-Solomon בעזרת שיטת סיעוף וחסימה (branch and bound).

בבדיקה של האלגוריתם, האלגוריתם הוכיח את עצמו כיעיל למדי עבור מספר בעיות בגודל פרקטי. האלגוריתם אף הצליח לפתור בצורה אופטימאלית בעיות גדולות עד פי שישה מבעיות שנפתרו בצורה אופטימאלית קודם לכן.

קיימים פחות מחקרים בנושא VRPTW עם חלונות זמן "גמישים". פתרון VRPTW עם חלונות זמן "גמישים" הוצג ע"י Koskosidis, Powell ו-Solomon, אשר הרחיבו את האלגוריתם של Fisher ו-Jaikumar ([42],[4],[18],[47]). באלגוריתם שלהם, Koskosidis, Powell ו-Solomon הראו כיצד ניתן להפוך בעיה עם חלונות זמן "קשיחים" לבעיה עם חלונות זמן "גמישים". Koskosidis, Powell ו-Solomon מציינים מספר גורמים שהניעו אותם לבחור במודל בעל חלונות זמן "גמישים". בין הסיבות ניתן למצוא שהמודל בעל חלונות זמן "גמישים" הוא מקיף יותר ממודל בעל חלונות זמן "קשיחים", בעולם האמיתי אנו נפגוש יותר מקרים עם חלונות זמן "גמישים" מאשר "קשיחים" (לא סביר שספק ישקיע סכומים גבוהים מאד רק בשביל שיוכל לעמוד בחלון זמן מסוים ולכן לרוב תהיה התגמשות מסוימת) ובנוסף, אלגוריתם לפתרון VRP עם חלונות זמן "גמישים" ימצא לרוב פתרונות שאלגוריתם לפתרון VRP עם חלונות זמן "קשיחים" לא ימצא. מימוש האלגוריתם הראה שהבעיה העיקרית של המודל היא בזמן הריצה שלו, שגדול עד פי שנים מאלגוריתמים אחרים. מצד שני, האלגוריתם סיפק תוצאות מרשימות, שהכותבים מאמינים שהן התוצאות האופטימאליות עבור הבעיות אותן הם בדקו ([47]).

3.5.3. ניתוב רכבים ממספר מחסנים

לארגון יתכן ויש יותר ממחסן אחד, שממנו ניתן לספק סחורה ללקוחות. אם הלקוחות מקובצים סביב המחסנים, אז ניתן לראות את הבעיה כמספר בעיות VRP קטנות יותר. אבל, אם הלקוחות והמחסנים נמצאים בפיזור, אז עומדת בפנינו בעיה חדשה – ניתוב רכבים ממספר מחסנים (Multi-Depot Vehicle Routing Problem – MDVRP).

בדומה ל-VRP גם ב-MDVRP אנו נדרשים להקצות מחסן ללקוח. צי של רכבים הנמצא בכל אחד מהמחסנים. צי רכבים המשויך למחסן מסוים משרת את לקוחות המחסן, וחוזר בסופו של התהליך אל אותו המחסן ממנו הוא יצא.

Wren ו-Holliday ([42],[61]) תארו שיטה המתחלקת לשני חלקים: פיתוח פתרון ראשוני, שלאחריו מופעלות מספק היוריסטיקות לשיפור פתרון זה. בכדי ליצור את הפתרון הראשוני Wren ו-Holliday השתמשו בשיטה הדומה לשיטת אלגוריתם Sweep של Gillet ו-Miller ([4],[18],[36]). בעת בניית הפתרון הראשוני, נבחן כל קודקוד ונבדק המרחק שלו אל הקודקוד

האחרון במסלול שנבנה ואל המחסן הקרוב ביותר. אם המרחק למחסן הקרוב קצר יותר מהמרחק לקודקוד האחרון במסלול, הקודקוד החדש ישויך למסלול חדש, אחרת הוא יוסף למסלול הקיים. לאחר שלב הקמת המסלולים הראשוניים, משתמשים ב-7 היוריסטיקות בכדי לשפר את המסלולים. היוריסטיקות הללו כוללות בין היתר מהלכים של הזזת קדקודים בין מסלולים, הסרת מסלולים והעברת הקדקודים השייכים לאותם למסלולים למסלול אחר ואיחוד מסלולים למסלול אחד. השימוש באותן היוריסטיקות נעשה כל עוד ניתן להשיג שיפור במסלולים.

בדיקה של האלגוריתם הראתה שהוא משיג תוצאות טובות יותר יחסית לאלגוריתמים האחרים שהיו קיימים לפתרון MDVRP.

Golden, Magnanti ו-Nguyen ([38]) הציגו שתי היוריסטיקות לפתרון MDVRP. ההיוריסטיקה הראשונה היא התאמה של אלגוריתם Savings. ההיוריסטיקה השניה היא עבור בעיות גדולות, שבהן יעילות הביצוע היא בעלת חשיבות. ההיוריסטיקה השניה מבוססת על שני חלקים, שיוך כל לקוח למחסן, ובניית מסלולים עבור כל מחסן. את הלקוחות אנו מחלקים לשתי קבוצות, קבוצת הלקוחות הקרובים למחסן וקבוצת הלקוחות הרחוקים יותר, בהתאם למרחק של כל לקוח מהמחסן הקרוב ביותר אליו והמחסן השני הקרוב אליו. לקוחות שלא ניתן לשייך אותם בוודאות למחסן מסוים, משויכים למחסן בהתאם לאלגוריתם Savings. לאחר שכל הלקוחות שויכו למחסנים, אלגוריתמים רגילים לפתרון VRP משמשים למציאת המסלולים עבור כל מחסן בנפרד ([42]).

3.5.4. ניתוב רכבים תלוי זמן

בעיית ניתוב רכבים תלוי זמן (Time Dependent Vehicle Routing Problem – TDVRP) היא הרחבה נוספת ל-VRP הבסיסי. צי כלי רכב בעל קיבולת קבועה צריך לשרת מספר קבוע של לקוחות בעלי דרישה קבועה, ממחסן מרכזי. כל לקוח צריך לשייך לאחד מכלי הרכב של הצי, ולכל כלי רכב יש להקצות נתיב נסיעה קבוע, כך שסך הזמן שיידרש לכל כלי הרכב בצי להשלים את המסלולים שהוקצו להם יהיה מינימאלי. זמן הנסיעה בין שני לקוחות, או בין המחסן ללקוח, תלוי במרחק שבין שתי הנקודות ובשעת היום ([51]).

בהרחבות לבעיית ניתוב הרחבים שהוצגו קודם לכן, המטרה הייתה מציאת מסלולים כאלו, כך שהמרחק הכולל של כל המסלולים יהיה מינימאלי, מתוך ההנחה שמרחק קצר ביותר משמעו עלויות קטנות יותר. אולם ההנחה שעלות הנסיעה ליחידת מרחק (למשל עלות הנסיעה לקילומטר) היא קבועה וידועה באופן דטרמיניסטי אינה נכונה, והיא מהווה קירוב לעלות האמיתית. במציאות, באזורים כגון אזורים עירוניים צפופים, זמן הנסיעה מנקודה אחת לשנייה הוא בד"כ אינו פונקציה של המרחק בין שתי הנקודות, היות ולרוב מהירות הנסיעה אינה קבועה, והיא מוכתבת מגורמים שונים, כדוגמת תאונות דרכים, תנאי מזג האוויר וגורמים אקראיים אחרים. גורם חשוב נוסף, שמשפיע באופן סדיר על זמן הנסיעה, הוא שינוי בנפח כלי הרכב הנוסעים בציר הנסיעה (עומסי ופקקי תנועה). שינויים בנפח כלי הרכב מתרחשים באופן סדיר, הן ברמה היומית והן הרמה השבועית, חודשית, וכדומה ([51]).

אם לא נתייחס לתלות שבין שעת היום לזמן הנסיעה, אנו עלולים לקבל תוצאות רחוקות מהתוצאות האופטימאליות, עם מבנה מסלולים שונה ועם מספר שונה של רכבים שנדרש לביצוע

המשימה. בנוסף, אם ישנם אילוצים נוספים, כדוגמת חלונות זמן או זמני שהייה מקסימאליים אצל הלקוח, אנו עלולים לקבל מסלולים, שבפועל אינם עומדים באילוצים הללו ([51]).

את TDVRP ניתן לחלק ל-4 קטגוריות בהתאם למאפייני זמן הנסיעה ופונקצית העלות.

א. מודלים מבוססים זמן נסיעה פשוט ופונקצית עלות. במודלים אלו, מתעלמים משינויים קיצוניים בזמני הנסיעה.

ב. מודלים מבוססי זמן נסיעה דיסקרטי ופונקצית עלות. במודלים כאלו מחלקים את מרחב הזמן למספר חלקים קבועים, כמו כן פונקצית העלות ופונקצית הזמן הן פונקציות מדרגה יחסית לנקודת ההתחלה. הנחות אלו נמצאות בשימוש מרבית בעיות הקצאת הרכבים תלוי הזמן. הבעייתיות במודל הוא השימוש בפונקציות מדרגה, שכן בעולם האמיתי השינויים הן בזמן והן בעלות הם רציפים.

ג. מודלים מבוססי זמן נסיעה דיסקרטי ופונקצית רציפות. מודלים אלו הם קרובים ביותר לעולם האמיתי, העולם הרציף. אולם למודלים כאלו עדיין יש מגבלות. בין המגבלות ניתן למצוא פונקציות שמהוות הערכה למה שנצפה בעולם האמיתי. שימוש בפונקציות רציפות מעלה את רמת הסיבוכיות של הבעיה, ולעיתים ישנן פונקציות שמתמטית בלתי ניתן לפתור אותן.

ד. מודלים מארקוביים עבור זמן נסיעה ופונקצית עלות.

מספר המאמרים בנושא TDVRP הוא קטן ביותר לעומת שאר בעיות ה-VRP ([43]). בעיות קרובות ל-TDVRP נחקרו באופן יותר אינטנסיבי. בניהן ניתן למצוא את הזרימה המקסימאלית תלוית זמן, בעיית בחירת מסלול תלוי זמן ובעיית הסוכן נוסע תלוי זמן.

בעיית הזרימה המקסימאלית תלויית זמן הוצגה לראשונה ע״י Ford ו-Fulkerson ב-1958 ([31]), ופיתוחים ופתרונות רבים לבעיה הוצגו מאז. בבעיית הזרימה המקסימאלית תלויית זמן אשר בחנו Ford ו-Fulkerson הייתה נתונה רשת, אשר לכל צלע של הרשת משויכים שני מדדים חיוביים, האחד, ערך זרימה, והשני, זמן המעבר מקודקוד אחד של הצלע אל הקודקוד השני. בנוסף ישנם שני קדקודים ברשת, אשר נקראים קודקוד המקור (ממנו אנו רוצים להעביר סחורה) וקודקוד היעד (אליו אנו רוצים להעביר את הסחורה). Ford ו-Fulkerson רצו לדעת מהי כמות הסחורה שניתן להעביר מקודקוד המקור לקודקוד היעד בפרק זמן נתון, כאשר בכל אחד מהקדקודים ניתן להעביר את הסחורה לקודקוד הבא ישירות עם הגעתה, או לחכות פרק זמן מסוים ואז להעביר אותה הלאה.

בעיית הזרימה המקסימאלית תלויית הזמן של Ford ו-Fulkerson היא מקרה פרטי של בעיות המסלול הקצר תלוי זמן. בבעיה זו מספר רב של נוסעים ״מתחרים״ בניהם על תנועה ברשת בכדי להגיע ליעדם בזמן. ניתן למצוא את הנוסעים בין הקשתות השונות של הרשת, ולא רק במסלולים הקצרים ביותר, בהתאם למודלים שונים שמדמים את תנועת הנוסעים. מודלים ראשונים לבעיה זו הוצגו ב-1984 ע״י Marguier ו-Ceder ([52], [43]).

בעיית הסוכן נוסע תלוי הזמן החלה להופיע בספרות בשנות ה-60. Tucker ,Miller ו-Zemlin ([53]) הציגו מודל לפתרון בעיית סוכן נוסע תלוי זמן, שבה נבנה מסלול המילטוני בעל משקל

מינימאלי אשר עובר דרך *n* קדקודים, כאשר המשקל הקשת (זמן הנסיעה בקשת) העוברת מקודקוד *i* לקודקוד *j* תלוי בשעת היום ([43]).

Bowman ([11]) הציג לראשונה את בעיית תזמון הייצור כבעיה התלויה בזמן. Picard ו-Queyranne ([55]) הציגו את בעיית תזמון הייצור כבעיית הסוכן הנוסע תלוי הזמן (TDTSP). בבעיה בה עסקו Picard ו-Queyranne יהיה צורך לבצע *n* פעולות שונות ($J_1, J_2, \ldots, J_n$), כאשר עלות ה-Set-up, הדרושה להכנת המכונה לאחר סיום פעולה J_i, ולפני התחלת פעולה J_j, נתונה. המכונה מתחילה ממצב התחלתי, ולאחר הפעולה האחרונה יש להביאה למצב סופי. Picard ו-Queyranne היו מעוניינים לדעת מהוא סדר הפעולות שיש לבצע, כך שסך עלות ה-Set-up תהיה מינימאלית. היות ולצורך מציאת רצף הפעולות שעבורו יתקבל סך זמן Set-up מינימאלי, יש צורך לבדוק את כל הרצפים האפשריים, הרי שלמעשה אנו עוסקים בבעיית TDTSP.

Picard ו-Queyranne הציגו שני ניסוחים של הבעיה כבעיית תכנות ליניארי בשלמים, שהיו הרחבות של ניסוחי TSP שהוצגו קודם לכן. באלגוריתם שהציעו Picard ו-Queyranne הוצג ה-TDTSP כגרף רב-צדדי, שבו כל קבוצה יצגה את אחת הפעולות, וכל אחד מהקדקודים השייכים לקבוצה ייצג אחד מהמיקומים האפשריים בסדר הפעולות. כל אחת מהקשתות בגרף קיבלה את המשקל המתאים ביחס לקודקוד ממנו היא יצאה והקודקוד אליו היא הגיע. ע״י שימוש באלגוריתם למציאת המסלול הקצר ביותר, התקבל מסלול, אשר אם אין בו שתי נקודות אשר הן בעלות אותו מיקום בסדר הפעולות, הרי שקיבלנו את סדר הפעולות האופטימאלי, אחרת, אך יש לנו את החסם התחתון של העלות המינימאלית.

Malandraki ו-Daskin ([51]) בחנו הן את בעיית הקצאת המסלולים תלוי זמן ואת הן בעיית הסוכן נוסע תלוי הזמן (TDTSP), שהינו מקרה פרטי של TDVRP, שבו דרישת כל הלקוחות קטנה מיכולת ההובלה של כלי רכב יחיד.

Malandraki ו-Daskin התייחסו לשני מאפיינים של TDVRP ([44],[51]). המאפיין הראשון הוא ש-TDVRP היא בעיה א-סימטרית. המאפיין השני הוא שהיוריסטיקות ההכנסה וה-k-opt אינן ניתנות להתאמה בקלות לפתרון TDVRP מכוון שזמן המעבר מקודקוד לקודקוד תלוי הן בכיוון והן בזמן ההתחלה. מאפיין נוסף ([44]) שלא הוזכר ע״ Malandraki ו-Daskin הוא שאי-השוויון המשולש (המסלול $A \to B \to C$ ארוך מהמסלול $A \to C$) אינו מתקיים ב-TDTSP.

Malandraki ו-Daskin ([51]) הציגו שלושה אלגוריתמים חמדניים לפתרון TDVRP. באלגוריתם הראשון אנו בונים מסלולים אשר מתחילים במחסן. הוספת קודקוד למסלול נעשת ע״י בחירה אקראית של קודקוד מתוך *n* הקדקודים אשר ניתן להגיע אליהם מהקודקוד האחרון במסלול, הם אינם מפירים את אילוצי הבעיה וזמן ההגעה אליהם הוא הקצר ביותר. אנו חוזרים על התהליך עד שכל הקדקודים מוצמדים למסלולים. האלגוריתם השני דומה לראשון, כאשר השוני הוא באופן הוספת הקדקודים למסלול. קודקוד מתווסף למסלול בתנאי שניתן להגיע אליו מהקודקוד האחרון במסלול, הוא אינו מפר את אילוצי הבעיה וזמן ההגעה אליו הוא הקצר ביותר. באלגוריתם השלישי, אנו בונים את המסלולים במקביל בדומה לאלגוריתם השני. עבור כל קודקוד אחרון בכל אחד מהמסלולים שנבנו, אנו מחפשים קודקוד אשר אינו מפר את אילוצי

הבעיה וזמן ההגעה אליו הוא הקצר ביותר. אנו מוסיפים את הקודקוד שזמן ההגעה אליו הוא הקצר ביותר למסלול המתאים. בנוסף לאלגוריתמים אלו, מביאים Malandraki ו-Daskin אלגוריתם המבוסס על היוריסטיקת cutting plane, שיטה בה אנו פותרים בעיית תכנות ליניארי לא בשלמים, כאשר אנו בודקים אם הפתרון האופטימאלי שהתקבל הוא פתרון במספרים שלמים או לא, במידה ולא, אנו מוסיפים אילוצים לבעיה, עד שמתקבל פתרון במספרים שלמים.

תוצאות חישוביות הראו שהאלגוריתם המבוסס על היוריסטיקת cutting plane מספק תוצאות טובות יותר מהאלגוריתמים החמדניים, אולם ניתן להשתמש בו עבור בעיות קטנות יחסית. כמו כן, אחוז הסטייה מהפתרון האופטימאלי גדל ככל שהבעיה גדלה.

Ichoua, Gendreau ו-Potvin ([43]) הציגו מודל שבו הזמן (T) מחולק לחלקים שווים, $t_1, t_2, ..., t_n = T$, ומהירות הנסיעה בחלקים השונים עבור כל קשת בגרף ידועה. יתר לכן, הם מניחים שזמן הנסיעה על הקשת עצמה אינו קבוע והוא יכול להשתנות, כאשר עוברים בקשת בין חלקים שונים של הזמן, למשל מ- t_1 ל- t_2. פונקצית הזמן שבה הם השתמשו הייתה פונקצית מדרגה, שהבטיחה שמירה על עקרון ה-FIFO, כלומר, אם יש באפשרותנו לעבור בקשת בשני זמנים ביום, אז בזמן המוקדם יותר אנו גם נסיים את המעבר בקשת מוקדם יותר. את פונקצית החיפוש שלהם הם שילבו במטה-היוריסטיקה Tabu Search באופן מקבילי. האלגוריתם הראה שיפור כמעט בכל תסריטי הבדיקה שבוצעו. המודל נבדק גם בסביבה דינאמית, שבה דרישת הלקוחות לא ידועה בהתחלה, אלא מתגלה עם הזמן. גם עבור מודל זה האלגוריתם הראה שיפור ברור תסריטי הבדיקה שבוצעו.

3.5.5. ניתוב רכבים סטוכסטיים

VRP קלאסי אינו מתייחס לנתון חשוב שקיים במציאות – סטוכסטיות. בעיית ניתוב רכבים סטוכסטיים (Stochastic Vehicle Routing Problem – SVRP) מתעוררת כאשר אחד או יותר ממשתני הבעיה הם אקראיים ([35]). בעיות סטוכסטיות שונות מאחיותיהן הדטרמניסטיות בכמה אספקטים מרכזיים. שיטת הפתרון שונה, מספר מהמאפיינים הבסיסיים של VRP דטרמיניסטי לא מתקיימים בבעיה הסטוכסטית, וכן מתודולוגיות הפתרון הן הרבה יותר מורכבות.

את קבוצת בעיות ניתוב הרכבים הסטוכסטיים נהוג לחלק לתת-קבוצות בהתאם למשתנה המתנהג באופן אקראי. תתי-הקבוצות הנפוצות ל-SVRP הן ניתוב רכבים עם דרישות סטוכסטיות (VRPSD) וניתוב רכבים עם זמני נסיעה סטוכסטיים (VRPSTT). הקיף המחקר הנעשה בתחום ה-VRP הסטוכסטי קטנה בהרבה בהשוואה לכמות המחקר הנעשת בתחום ה-VRP הדטרמיניסטי. Dror, Laporte ו-Trudeau מניחים שהסיבה לכמות הקטנה במחקר SVRP היא בגלל הסיבוכיות הגדולה גם כך של VRP, שהופכת בעיות סטוכסטיות לקשות יותר אפילו לניתוח אנאליטי ([27]). בשנים האחרונות, למרות מורכבות הבעיה, ההתעניינות בבעיות הסטוכסטית גוברת, והתקדמות רבה נעשתה בתחום.

כשם ש-VRP עצמו מבוסס על בעיות "פשוטות" יותר (שהן עדיין בעיות NP-Hard), כגון בעיית הסוכן נוסע ובעיית מחלק הדואר הסיני, כך מבוסס ה-SVRP על הגרסאות הסטוכסטיות של אותן הבעיות.

בעיית סוכן נוסע עם לקוחות סטוכסטיים (Probabilistic Traveling Salesman Problem) הוצגה ע"י Jalliet בשנת 1985 ([27]). בבעיה זו, לכל לקוח, קיימת הסתברות מסוימת P_i שנצטרך לעבור בו, כאשר אנו מעוניינים לבנות מסלול אשר עובר דרך כל הלקוחות שיש צורך להגיע אליהם ושאינו עובר דרך לקוחות שאין צורך להגיע אליהם. Jalliet הציג פתרון לבעיה המבוסס על שני שלבים. בשלב הראשון אנו בונים מעגל המילטוני העובר דרך כל הקדקודים (ראה [25], פרק 10). בשלב השני, אנו נעים על-פי המסלול, כאשר במידת הצורך אנו מדלגים על קדקודים שאנו מגלים שאין צורך לעבור דרכם ([35]). בעיית סוכן נוסע סטוכסטית נוספת היא בעיית סוכן נוסע עם זמני נסיעה סטוכסטיים, שבה המשקלים על הקשתות הם אקראיים. מספר פתרונות הוצגו לבעיה זו. פתרונות אלו מבוססים גם הם על שני שלבים. בשלב הראשון ישנו ניסיון לבנות את המסלול הקצר ביותר אשר ההסתברות לסיים אותו בפרק זמן נתון היא מקסימאלית ([35]).

בעיית ניתוב רכבים עם דרישות סטוכסטיות (Vehicle Routing Problem with Stochastic Demands – VRPSD) היא ככל הנראה בעיית הקצאת המסלולים הסטוכסטית הנחקרת ביותר. VRPSD הנה בעיית ניתוב רכבים שבה דרישות הלקוחות סטוכסטיות, קרי, אינן ידועות בשעת תכנון המסלולים, אך ההסתברויות לדרישות השונות אצל הלקוחות השונים ידועות.

האלגוריתם ההיוריסטי הראשון לפתרון VRPSD הוצע ע"י Tillman ([60]). Tillman הציע פתרון שמבוסס על אלגוריתם Savings כאשר ישנם מספר מחסנים ([35],[60]) וכאשר דרישות הלקוחות מתפלגות פאוסונית. Tillman הוסיף הגבלות נוספות ל-VRPSD שבה עסק, כגון, הגבלה על אורך מסלול (כל אחד מהמסלולים אינו יכול להיות גדול ממס' קילומטרים נתון) וכן "עונשים" על כלי רכב שהנם כמעט ריקים או מלאים יתר על מהמידה. אלגוריתם Savings פותר את בעיית ניתוב הרכבים ע"י איחוד מסלולים פשוטים למסלולים מורכבים יותר על בסיס החיסכון שמתקבל מאיחוד המסלולים, בהתאם לנוסחה $S_{jk} = d_{0j} + d_{0k} - d_{jk}$. Tillman אשר התבסס על אלגוריתם Savings, הראה שישנם מצבים שבהם בוחרים לאחד שני מסלולים ולשייכם למחסן אחד (רחוק יותר) במקום למחסן שני (קרוב יותר) בגלל שהחיסכון המתקבל הוא גדול יותר, אך במציאות עדיף לשייך את המסלול המאוחד למחסן השני (הקרוב יותר) שכן באופן בולט המסלול יהיה קצר יותר. בכדי להתגבר על בעיה זאת הציע Tillman לשנות את האופן שבו יש לחשב את החיסכון. Tillman הציע לחשב את החיסכון על פי הנוסחא $S_{jk}^{\,i} = \tilde{d}_j^{i} + \tilde{d}_k^{i} - d_{jk}$, כאשר $\tilde{d}_j^{i} = \min_s d_j^{\,s} - \left(d_j^{\,i} - \min_s d_j^{\,s}\right)$ שמשמעו שהמרחק בין מחסן i ללקוח j שווה לפעמיים המרחק הקצר ביותר בין הלקוח ובין כל המחסנים פחות המרחק בין מחסן i ללקוח j. מכוון ש-Tillman עוסק בבעיה סטוכסטית הוא מגדיר שתי פונקציות עלות, הראשונה היא עלות עודף, כלומר העלות של העברת סחורה מעבר לנדרש, והשניה היא עלות חוסר, העלות או הקנס שיש לשלם בעקבות זה שלא ניתן לספק את דרישות כל הלקוחות במסלול. שתי הפונקציות מבוססות

על ההסתברויות השונות של דרישות הלקוחות. כשאנו מדברים על חלוקת סחורה, המטרה בהקצאת המסלולים היא להביא למינימום את העלות הכוללת (עלות העודף פלוס עלות החוסר). Tillman אף הראה כיצד ניתן לחשב את ערכי המינימום, וכיצד לשלבם באלגוריתם ה-Savings בשלב קבלת ההחלטות לגבי המסלולים המיועדים לאיחוד.

Stewart ו-Golden ([58]) וכן Golden ו-Yee הציגו מודל CCP (Chance-Constrained Programming) עם שינוי קל של אלגוריתם Savings ([35]). Golden ו-Stewart התייחסו במודל שלהם להתפלגות פואסונית, בעוד ש-Golden ו-Yee התייחסו במודלים שלהם לסוגים שונים של התפלגויות. Golden ו-Stewart הציגו מודל CCP, שיטה שבה מחפשים מסלולים שההסתברות שנצליח לבצע אותם גבוהה מסף מסוים שקבענו לעצמנו, כשתכנון כזה אינו לוקח בחשבון את עלות התיקונים במסלולים שנצטרך לעשות במידת הצורך, ושני מודלים של SPR (Stochastic Programming), כשבעזרת SPR אנו מוצאים מסלולים ראשונים כך שעלות התיקונים במסלולים אלו בשלב השני יהיו מינימאליים. במודל ה-SPR הראשון, Stewart ו-Golden משייכים "עונש" לקטעים במסלול ביחס להסתברות שקיבולת כלי הרכב תגדל, ואילו במודל ה-SPR השני שיוך "עונש" נעשה ביחס להסתברות שדרישת הלקוחות במסלול תגדל מקיבולת כלי הרכב. הם סבירות גבוהה שיגרמו להארכת המסלול האמיתי. בנוסף, הציגו Golden ו-Stewart שתי היוריסטיקות. הראשונה, מבוססת על אלגוריתם Savings. השניה, הנקראת GLM, מבוססת על רלקסצייה לג'ראנג'יאנית. Golden ו-Yee הראו כיצד ניתן להשתמש באלגוריתם Savings כאשר נעשה שימוש בממוצע וסטיית התקן על מנת לחשב את החיסכון המתקבל מחיבור המסלולים ([39]).

Bertsimas ([9]), אשר היה מעוניין למצוא את קבוצת המסלולים שאורכם הכולל הוא המינימאלי, הציג שתי אסטרטגיות על-מנת לפתור VRPSD.

באסטרטגיה הראשונה, לכל כלי הרכב בצי נבנים מסלולים קבועים מראש, שמגיעים לכל לקוח פעם אחת, בהתבסס על ההסתברויות לדרישות הלקוחות. כלי הרכב נעים באותם מסלולים קבועים מראש, ועוברים דרך כל הלקוחות. במידה וכלי רכב מגיע ללקוח, ואין ביכולתו לשרת את הלקוח, כלי הרכב חוזר למחסן, מעמיס סחורה נוספת, וחוזר למסלול שלו, לאותה הנקודה שבה הוא הפסיק אותו.

בעוד שהאסטרטגיה הראשונה מתאימה למצב שבו דרישות הלקוחות ידועות לנו רק בזמן ההגעה ללקוח, האסטרטגיה השנייה מתאימה למצב שנו דרישות הלקוחות ידועות לנו לפני ההגעה ללקוח. גם באסטרטגיה השניה, לכל כלי רכב בצי נבנים מסלולים קבועים מראש, שמגיעים לכל לקוח פעם אחת, בהתבסס על ההסתברויות לדרישות הלקוחות. אולם, בעוד שבאסטרטגיה הראשונה כלי הרכב נעים באותם מסלולים קבועים מראש, ועובר דרך כל הלקוחות, באסטרטגיה השניה אנו משתמשים במידע הידוע לנו מראש על מנת לשפר את אותם מסלולים קבועים מראש. שיפור במסלול יכול להתקבל במידה ואנו יודעים שללקוח מסוים אין כל דרישה. במצב כזה, אנו יכולים פשוט לדלג על אותו הלקוח ולקצר את הדרך במידת האפשר. לדוגמה, אם יש לנו שלושה לקוחות, 1, 2 ו-3, ואנו יודעים ש-2 אינו דורש סחורה באותו היום, אנו נשנה את המסלול כך שלא נעבור דרך לקוח 2, אלא מלקוח 1 ניסע ישירות ללקוח 3. גם

באסטרטגיה זו, במידה וכלי רכב מגיע ללקוח, ואין ביכולתו לשרת את הלקוח, כלי הרכב חוזר למחסן, מעמיס סחורה נוספת, וחוזר למסלול שלו, לאותה הנקודה שבה הוא הפסיק אותו.

באסטרטגיה השניה, דרישות הלקוחות ידועות לנו מלכתחילה, ולכן טבעי יותר לבנות מסלולים חדשים בהתאם לדרישות הידועות. אך ישנם מיקרים שבהם מסיבות פרקטיות לא נרצה לעשות זאת. למשל, כאשר לא עומד לרשותנו זמן מחשב, או שהזמן הדרוש לחישוב המסלולים הוא רב ואין לנו אפשרות להקצות אותו. אפשרות נוספת היא שהדרישות מתגלות תוך כדי תנועה, אך לפני ההגעה ללקוח, כך שאנו צריכים מסלול כלשהו לנוע לפיו.

Bertsimas הראה ששימוש באסטרטגיות אלו הוא די יעיל במיוחד כאשר התפלגות הדרישות בין הלקוחות היא זהה. כמו כן, הוצג אלגוריתם יעיל לחישוב אורכו של המסלול הראשוני.

בניתוב רכבים עם זמני נסיעה סטוכסטיים (VRPSTT) אנו לא יודעים בוודאות מהי מהירות הנסיעה בכל קטע, ולכן, לדוגמא, בכל מקרה שאנו צריכים להגיע לקודקוד מסוים בנקודת זמן מסוימת אין לנו אפשרות לבנות מסלול אשר יבטיח לנו שנגיע בזמן.

Laporte, Louveaux ו-Mercure הציגו פתרון מבוסס CCP לפתרון VRPSTT ([48]). מטרתם היא להביא למינימום את העלויות המסלולים, כאשר הם מבטיחים שעבור אחוז מסוים מין המקרים, זמן השהייה הכולל במסלולים לא יעבור סף קבוע. כמו כן, הם מציגים אלגוריתם סיעוף וחלוקה (branch and cut) לפתרון הבעיה. בכדי לבדוק את האלגוריתם שלהם, נבחרו מספר בעיות בנות 10, 15 ו-20 לקוחות. כישלון הוגדר בשני מצבים, כאשר האלגוריתם לא הגיע לפתרון תוך 300 שניות, או כאשר צריכת הזיכרון המחשב הגיע לגודל מסוים. התוצאות שלהם הראו שהתמודדות האלגוריתם עם הבעיה תלויה בערכו של רכיב ה"עונש" שהוכנס לאלגוריתם ואין השפעה ישירה של מספר הרכבים או הלקוחות.

Kenyon ו-Morton הציגו פתרון ל-VRPSTT שבה לא הייתה הגבלת קיבולת לרכבים ([46]). בבעיית ניתוב הרכבים שם זמני נסיעה סטוכסטיים שבה עסקו Kenyon ו-Morton לא היה שימוש בגרף שלם, אולם ניתן היה להגיע מהמחסן לכל לקוח ולחזור בחזרה למחסן. הם הציגו שני מודלים השונים ביניהם בפונקצית המטרה. במודל הראשון חיפשו Kenyon ו-Morton מסלולים כאלו שסך הזמן שלהם הוא המינימאלי. במודל השני, חיפשו Kenyon ו-Morton מסלולים כך שההסתברות לסיים אותם תוך פרק זמן נתון היא המקסימאלית. שתי שיטות פתרון מבוססות על סיעוף וחלוקה פותחו. הראשונה פותרת גרסה דטרמיניסטית של הבעיה הסטוכסטית, והיא ניתן ליישום כאשר מספר המשתנים הסטוכסטיים הוא קטן יחסית. השיטה השניה מאחדת את שיטת הסיעוף וחלוקה עם שיטת מונטה-קארלו (שיטה המשתמשת בידע סטטיסטי על הבעיה וכן על חילול מספר רב של מקרים אקראיים בעזרת מחשב, על מנת לקבל קרוב לפתרון).

משילוב של VRPSD ו-VRPSC מתקבלת בעיית ניתוב רכבים בעלת לקוחות ודרישות סטוכסטיים (Vehicle Routing Problem with Stochastic Customers and Demands – VRPSCD). ככל שמוסיפים אילוצים ואי-ודאויות לבעיה, הבעיה הופכת לקשה יותר. VRPSCD היא בעיה קשה במיוחד, שאפילו את פונקציה המטרה לא קל לחשב ([35]).

3.5.6. סיכום היוריסטיקות והרחבות VRP

הטבלה הבאה היא סיכום של האלגוריתמים השונים (ללא המטה-יוריסטיקות). הרשימה המופיעה בטבלה כוללת את שם האלגוריתם, אשר לרוב מיוצג ע״י מחבריו, הרחבת ה-VRP אותה הוא אמור לפתור וכן אם הוא מבוסס או משתמש באלגוריתם אחר, ואם כן איזה. למשל אלגוריתם Saving משמש לפתרון CVRP, כמו כן, אלגוריתמים אחרים מתבססים עליו לפתרון SDVRP, VRPTW, MDVRP ו-SVRP (מצוינים ע״י שימוש בכתב נטוי).

האלגוריתם	משמש לפתרון	מבוסס / משתמש ב :
Dantzig ו-Ramser ([21])	CVRP	
Clarke ו-Wright (Savings) ([16])	CVRP, *SDVRP*, *VRPTW*, *MDVRP*, *SVRP*	
Gillett ו-Miller (Sweep) ([36])	CVRP, *VRPTW*, MDVRP	
Fisher ו-Jaikumar (Cluster Method) ([30])	CVRP, *VRPTW*	
Burrows ([13])	SDVRP	Savings
Dror ו-Trudeau ([28])	SDVRP	Savings
Mullaseril, Dror ו-Leung ([54])	SDVRP	
Chen, Golden ו-Wasil ([15])	SDVRP	
Solomon ([57])	VRPTW	Savings, Sweep
Desrochers, Desrosiers ו-Solomon ([23])	VRPTW	
Koskosidis, Powell ו-Solomon ([47])	VRPTW	Cluster Method
Wren ו-Holliday ([61])	MDVRP	Sweep
Golden, Magnanti ו-Nguyen ([38])	MDVRP	Savings
Malandraki ו-Daskin ([51])	TDVRP	
Ichoua, Gendreau ו-Potvin ([43])	TDVRP	
Tillman ([60])	SVRP	Savings
Stewart ו-Golden ([58])	SVRP	Savings
Golden ו-Yee ([39])	SVRP	Savings
Bertsimas ([9])	SVRP	
Laporte, Louveaux ו-Mercure ([48])	SVRP	

טבלה 1 - סיכום אלגוריתמים המוזכרים בפרק

מהטבלה ניתן לראות שרבים מהמודלים שהוצגו משתמשים באלגוריתם Savings כבסיס לפתרון המודל. בעבודה זו גם כן, משמש אלגוריתם Savings כבסיס לפתרון הבעיה המוצגת.

ניתן למצוא עוד מחקרים רבים בתחום ה-VRP. אופי הבעיה והרצון להשליך את העולם האמיתי על המודלים השונים מוליד הרחבות רבות של הבעיה.

4. ניתוב רכבים המאופיין במשכי נסיעה תלויי זמן וסטוכסטיים

כפי שתואר בפרק הסקירה הספרותית, VRP נחקר רבות ב-45 השנים האחרונות, שבמהלכן פותחו שיטות רבות לפתרון הבעיה ולפתרון הרחבות רבות ומורכבות יותר של VRP, שכוללות בתוכן אילוצים שונים. ישנן שתי מטרות עיקריות בחקר VRP. הראשונה, פיתוח מודלים מדויקים ומהירים יותר לפתרון הבעיה. השנייה, פיתוח מודלים קרובים יותר למציאות, הכוללים בתוכם יותר ויותר אילוצים. את שתי המטרות הללו קשה לאחד יחדיו למטרה בודדת, שכן לרוב מודלים מורכבים יותר אשר כוללים בתוכם אילוצים רבים יותר, גורמים לכך שהאלגוריתמים קיימים יספקו תוצאות פחות מדויקות, ואילו לאלגוריתמים חדשים ייקח זמן רב יותר להגיע לפתרון מדויק.

בפרק זה יוצג מודל לניתוב רכבים המאופיין במשכי נסיעה תלויי זמן וסטוכסטיים וכן אלגוריתם היוריסטי למציאת פתרון קרוב לפתרון האופטימאלי עבור המודל, כשהמטרה היא למצוא אוסף מסלולים שסך זמן הנסיעה בהם יהיה מינימאלי. אנו נתייחס לכך שעבור צירים מסוימים (חלקם או כולם) זמן הנסיעה בציר משתנה לאורך היום, וכן הוא מתנהג סטוכסטית, כלומר, לכל שעה בשעות היום ישנה הסתברות שזמן הנסיעה הוא P_1 והסתברות אחרת שזמן הנסיעה הוא P_2 וכדומה. מודל זה הוא ניסיון נוסף לקרב את בעיית ניתוב הרכבים לעולם האמיתי, כשלמיטב ידיעתי, נכון להיום לא קיים בספרות התייחסות לשילוב זה של בעיות.

מכוון שהמטרה הכללית של כל ארגון היא הורדת עלויות למינימום ההכרחי, הרי שגם במודל זה ההנחה היא שהורדת זמן הנסיעה תצמצם את הוצאות הארגון, כאשר זמן הנסיעה משפיע על העלויות במספר אופנים כגון עלויות הדלק, האחזקה (טיפולים מונעים וכו'), תחזוקה (תיקון תקלות ותאונות), קנסות וכו'.

4.1. ניסוח מודל מתמטי

את בעיית ניתוב הרכבים נהוג לתאר ([29]) תוך שימוש בגרף, $G=(V,E)$, כאשר $V=\{v_0,v_1,...,v_n\}$ היא קבוצת קדקודים, כש-v_0 מייצג את המחסן, ושאר הקדקודים מייצגים לקוחות, וכן $E=\{(v_i,v_j):v_i,v_j\in V, i\neq j\}$ היא קבוצת הקשתות, המייצגות את המסלולים הקיימים בין הקדקודים. לכל לקוח יש דרישה חיובית מסוימת, d_i, כאשר ניתן לראות את דרישת המחסן d_0 כ-0. בנוסף קיימים V כלי רכב בארגון עם קיבולת מקסימלית D. כמו כן קיימת פונקצית עלות $C(i,j)$, אשר מחזירה את עלות המעבר, c_{ij}, בקשת (i,j).

כאמור, אנו מעונינים למצוא מספר מסלולים המקיימים את התנאים הבאים:

1. כל מסלול מתחיל ומסתיים במחסן (קודקוד v_0).
2. בכל מסלול אנו מגיעים ללקוח פעם אחת בלבד.
3. אין שני מסלולים אשר עוברים דרך אותו הלקוח.
4. סכום דרישות הלקוחות בכל מסלול אינו גדול מקיבולת משאית בודדת.
5. סך פונקציות העלות של כל המסלולים הוא מינימאלי.

לצורך עבודה זו, אנו נניח שמספר כלי הרכב, V, שעומדים לרשותנו אינו מוגבל (או לפחות שווה למספר המסלולים הנדרשים על מנת למצוא מסלולים העומדים בתנאים שצוינו קודם לכן). ניתן להגדיר את בעיית ניתוב הרכבים כבעיית תכנות ליניארי באופן הבא:

אנו נגדיר את $x_{ij}^{\ v}$ כמשתנה החלטה שמשמעו האם הקשת (i,j) נכללת במסלול של כלי רכב v (ואז $x_{ij}^{\ v}=1$) או לא ($x_{ij}^{\ v}=0$), הגדרה זו תבוא בהמשך לידי ביטוי באילוץ (2.9). אנו מעוניינים להביא את סך העלות למינימום, ולכן פונקציית המטרה היא:

$$\min \quad Z=\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{v=1}^{V}C_{ij}x_{ij}^{\ v} \tag{2.1}$$

המשוואות הבאות הן אילוצי הבעיה.

$$x_{ii}^{\ v}=0 \quad \forall i\in\{0,1,\ldots,n\}\ ,\ v\in\{1,2,\ldots,V\} \tag{2.2}$$

$$\sum_{j=1}^{n}x_{0j}^{\ v}\le 1 \quad \forall v\in\{1,2,\ldots,V\} \tag{2.3}$$

$$\sum_{i=1}^{n}x_{i0}^{\ v}\le 1 \quad \forall v\in\{1,2,\ldots,V\} \tag{2.4}$$

$$\sum_{j=1}^{n}\sum_{v=1}^{V}x_{ij}^{\ v}=1 \quad \forall i\in\{0,1,\ldots,n\}, i\ne j \tag{2.5}$$

$$\sum_{i=1}^{n}\sum_{v=1}^{V}x_{ij}^{\ v}=1 \quad \forall j\in\{0,1,\ldots,n\},\ j\ne i \tag{2.6}$$

$$\sum_{i=0}^{n}x_{ip}^{\ v}-\sum_{j=0}^{n}x_{pk}^{\ v}=0 \quad \forall v\in\{1,2,\ldots,V\},\forall p\in\{0,1,\ldots,n\} \tag{2.7}$$

$$\sum_{i=0}^{n}d_i\left(\sum_{j=0}^{n}x_{ij}^{\ v}\right)\le D \quad \forall v\in\{1,2,\ldots,V\} \tag{2.8}$$

$$x_{ij}^{\ v}\in\{0,1\} \quad \forall i,j\in\{0,1,\ldots,n\}\ ,\ v\in\{1,2,\ldots,V\} \tag{2.9}$$

כפי שהוגדר קודם לכן, קבוצת הקשתות הקיימת בגרף $G=(V,E)$ אינה כוללת קשתות מהסוג (i,i), אילוץ (2.2) מטרתו להבטיח שקשתות כאלו לא יהוו חלק מהפתרון. אילוצים (2.3) ו-(2.4) מטפלים במספר הרכבים היוצאים מהמחסן. אילוץ (2.3) מטפל במספר הפעמים שכלי רכב יוצא מהמחסן. אילוץ זה מבטיח לנו שכל כלי רכב יצא לא יותר מפעם אחת מהמחסן. באופן דומה, אילוץ (2.4) מטפל במספר הפעמים שכלי רכב חוזר למחסן. אילוץ זה מבטיח לנו שכל כלי רכב לא יחזור יותר מפעם אחת למחסן. שני אילוצים אלו יחדיו מבטיחים לנו שכל כלי רכב שיוצא מהמחסן חוזר אליו בסופו של דבר. אילוצים (2.5), (2.6) ו-(2.7) מטפלים בדרגות הכניסה והיציאה של שאר הקדקודים בגרף. אילוץ (2.5) מבטיח לנו שדרגת הכניסה לכל קודקוד היא 1, ואילוץ (2.6) מבטיח שדרגת היציאה מכל קודקוד היא 1. אילוץ (2.7) משמעותו שמכל קודקוד אליו נכנסו אנו

גם יוצאים. שלושת אילוצים אלו יחדיו מבטיחים לנו שכל קודקוד נמצא על מסלול אחד בלבד. אילוץ (2.8) מוודא שהקיבולת של כל אחד מכלי הרכב לא תהייה גדולה מהקיבולת המקסימאלית.

אנו נרחיב את המודל הנ"ל באופן הבא. נניח שפונקצית העלות $C(i, j)$ אינה תלויה רק בקשת דרכה אנו עוברים, אלה גם תלויה בזמן שאנו עוברים בה, כלומר אנו מקבלים פונקציה חדשה $C(i, j,t)$. אנו נמשיך ונרחיב את פונקצית העלות שלנו ע"י הוספת גורם של אקראיות. פונקצית העלות שלנו מורכבת משני גורמים המושפעים מהקשת דרכה אנו עוברים ומהזמן שאנו עוברים בה. אנו יכולים לראות את פונקצית העלות תלויית הזמן והאקראית כפונקציה המורכבת משני חלקים $C(i, j,t) = c(i, j,t) + \xi(i, j,t)$. החלק $c(i, j,t)$ היינו חלק קבוע של הפונקציה, כלומר עבור כל קשת בנקודת זמן נתונה אנו נקבל ערך קבוע. החלק $\xi(i, j,t)$ של פונקצית העלות הוא בעל מאפיינים סטוכסטיים. עבור כל קשת נתונה בנקודת זמן נתונה, אנו נקבל כל פעם ערכים שונים, בהתאם להתפלגות שמוגדרת ע"י הפונקציה.

אנו נראה כעת כיצד ניתן לשלב פונקציה התלויה בקשת ובזמן במודל לפתרון בעיית ניתוב הרכבים.

נניח שיש לנו משתנה החלטה x_{ij}^{tv}, אשר מקבל את הערכים 0 או 1, שמשמעו האם כלי רכב v מתחילי לעבור בקשת (i,j) בזמן t. בנוסף קיים אילוץ אשר אומר שבכל קשת ניתן לעבור רק פעם אחת, כלומר, לא קיימים שני משתנה החלטה, $x_{ij}^{t_1v_1}$ ו- $x_{ij}^{t_2v_2}$, עבור אותה הקשת אשר מקבלים את הערך 1. המשמעות של אילוץ זה הוא שאם נסכום את כל משתני ההחלטה של אותה הקשת, עבור כל הרכבים האפשריים וכל נקודות הזמן האפשריות, אנו נקבל את אחד מהסכומים האפשריים, 0 או 1. מכוון שזמן הוא דבר רציף, אנו למעשה צריכים להשתמש באינטגרל בכדי לסכום את כל משתנה ההחלטה של קשת מסוימת, ולכן אם הזמן שלנו מתחיל ב-0 ומסתיים בזמן T, הרי שסכום כל משתנה ההחלטה עבור קשת כלשהי שווה ל- $\sum_{v=1}^{V}\int_0^T x_{ij}^{tv}dt$. נבחן את הגדרת האינטגרל המוחלט. נניח שקיימת פונקציה $f(x)$ שמוגדרת בתחום $a \le x \le b$. נחלק את הקטע $[a,b]$ ל-n חלקים שווים, $\Delta x = \frac{a-b}{n}$. האינטגרל המוחלט של הפונקציה $f(x)$ בין $x=a$ ל- $x=b$ מוגדרת כ:

$$\int_a^b f(x)dx = \lim_{n\to\infty}\left\{ f(a)\Delta x + f(a+\Delta x)\Delta x + f(a+2\Delta x)\Delta x + ... + f(a+(n-1)\Delta x)\Delta x\right\}$$

על סמך הגדרה זו, אם נחלק את קטע הזמן בן 0 ל-T ל-m חלקים שווים, כש-m גדול עד כמה שניתן (שואף לאינסוף), אנו נקבל ש- $\sum_{v=1}^{V}\sum_{t=0}^{T} x_{ij}^{tv}$ הוא קרוב ל- $\sum_{v=1}^{V}\int_0^T x_{ij}^{tv}dt$. אם כך, אנו יכולים להפוך את משתנה הזמן הרציף למשתנה דיסקרטי ע"י חלוקה של קטע הזמן למספר רב של חלקים קטנים. כשיש בידינו אפשרות להפוך את משתנה הזמן מרציף לדיסקרטי, אנו יכולים להביא תאור פורמאלי של בעיית ניתוב הרכבים המאופיינים במשכי נסיעה תלויי זמן וסטוכסטיים כבעיית תכנות ליניארי.

על מנת להביא ניסוח פורמאלי של בעיית ניתוב הרכבים תלויי הזמן והסטוכסטים אנו צריכים לטפל בפונקצית העלות. היות ופונקצית העלות היא סטוכסטית, אין היא מביאה ערך יחיד, ולכן

יש צורך להשתמש במדד אחר, לרוב בתוחלת. תוחלת היא הערך הצפוי להתקבל מפונקציה כלשהי, והוא שווה לממוצע הפונקציה. בהתאם לזאת, במקום להשתמש בפונקצית העלות המקורית, $C_{ij}^{\ t}$, אנו נשתמש ב- $\bar{C}_{ij}^{\ t}$, כש- $\bar{C}_{ij}^{\ t} = E\left(C_{ij}^{\ t}\right)$.

על סמך הנאמר לעיל פונקצית המטרה היא:

$$\min \ Z = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{t=0}^{T}\sum_{v=1}^{V}\bar{C}_{ij}^{\ t} x_{ij}^{\ tv} \tag{2.10}$$

האילוצים שונו והותאמו כך שיהיו תלויי זמן באופן הבא:

$$x_{ii}^{\ tv} = 0 \quad \forall i \in \{0,1,...,n\}, v\{1,2,...,V\}, t \in \{0,1,...,T\} \tag{2.11}$$

$$\sum_{j=1}^{n}\sum_{t=0}^{T} x_{0j}^{\ tv} \le 1 \quad \forall v \in \{1,2,...,V\} \tag{2.12}$$

$$\sum_{i=1}^{n}\sum_{t=0}^{T} x_{i0}^{\ tv} \le 1 \quad \forall v \in \{1,2,...,V\} \tag{2.13}$$

$$\sum_{j=1}^{n}\sum_{t=0}^{T}\sum_{v=1}^{V} x_{ij}^{\ tv} = 1 \quad \forall i \in \{0,1,...,n\}, i \ne j \tag{2.14}$$

$$\sum_{i=1}^{n}\sum_{t=0}^{T}\sum_{v=1}^{V} x_{ij}^{\ tv} = 1 \quad \forall j \in \{0,1,...,n\}, j \ne i \tag{2.15}$$

$$\sum_{i=0}^{n}\sum_{t=0}^{T} x_{ip}^{\ tv} - \sum_{j=0}^{n}\sum_{t=0}^{T} x_{pk}^{\ tv} = 0 \quad \forall v \in \{1,2,...,V\}, \forall p \in \{0,1,...,n\} \tag{2.16}$$

$$\sum_{i=0}^{n} d_i \left(\sum_{j=0}^{n}\sum_{t=0}^{T} x_{ij}^{\ tv}\right) \le D \quad \forall v \in \{1,2,...,V\} \tag{2.17}$$

$$P\left(\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{t=0}^{T}\sum_{v=1}^{V} C_{ij}^{\ t} x_{ij}^{\ tv} < C^*\right) \ge \alpha \tag{2.18}$$

$$x_{ij}^{\ tv} \in \{0,1\} \quad \forall i,j \in \{0,1,...,n\}\ ,\ v \in \{1,2,...,V\}\ ,\ t \in \{0,1,..,T\} \tag{2.19}$$

בדומה למודל הקודם, קבוצת הקשתות הקיימת בגרף $G=(V,E)$ אינה כוללת קשתות מהסוג (i,i), אילוץ (2.11) הותאם כך, שעבור כל נקודת זמן שהיא, קשתות כאלו לא יהוו חלק מהפתרון. אילוצים (2.12) ו-(2.13) מטפלים במספר הרכבים היוצאים מהמחסן, וגם הם הותאמו להיות תלויי זמן. שני אילוצים אלו יחדיו מבטיחים לנו שכל כלי רכב שיוצא מהמחסן חוזר אליו בסופו של דבר. אילוצים (2.14), (2.15) ו-(2.16) מטפלים בדרגות הכניסה והיציאה של שאר הקדקודים בגרף, וגם הם תלויי זמן, כך שלא קיים קודקוד אשר בשתי נקודות זמן שונות יוצאים כלי רכב שונים ממנו. אילוץ (2.17) מוודא שהקיבולת של כל אחד מכלי הרכב לא תהייה גדולה מהקיבולת המקסימאלית, וגם הוא הותאם לתלות בזמן. אילוץ (2.18) אחראי על החלק הסטוכסטי של הבעיה. אילוץ זה אומר שאוסף המסלולים שאנו מחפשים יהיה כזה כך שבהסתברות מסוימת, α,

זמן הנסיעה באותם מסלולים לא יעלה על ערך קבוע C^*. אילוץ (2.19) אחראי לכך שמשתנה ההחלטה שלנו הוא בינארי.

4.2. אלגוריתם לפתרון בעיית ניתוב רכבים בעלי משכי נסיעה תלויי זמן וסטוכסטיים

בסעיף 4.1 הוצג מודל מתמטי לבעיית ניתוב רכבים המאופיין במשכי נסיעה תלויי זמן וסטוכסטיים. בסעיף זה של העבודה יוצג אלגוריתם לפתרון בעיה זו.

בעיית ניתוב הרכבים היא בעיית אופטימיזציה קשה, אשר לא ניתן לפתור אותה באופן אופטימאלי עבור בעיות בסדר גדול. הוספת אילוצים שונים, הופכת את הבעיה לקשה עוד יותר, ולכן רוב האלגוריתמים משתמשים בהיוריסטיקות שונות בכדי לפתור בעיות ניתוב רכבים, כפי שהוצג בפרק הסקירה הספרותית. מכיוון שהמודל שמוצג בעבודה זו הוא הרחבה של בעיה ניתוב הרכבים הבסיסית, האלגוריתם המוצג לפתרון הבעיה גם הוא אלגוריתם היוריסטי.

ישנן מספר שיטות נפוצות שיש לקחת בחשבון בעת תיכון אלגוריתם חדש אשר בא לפתרון בעיה קיימת או הרחבה של בעיה קיימת.

1. התאמת אלגוריתם קיים – בפרק סקירת הספרות הוצגו מספר אלגוריתמים אשר התבססו על אלגוריתם קיים, ואשר הרחיבו אותו והתאימו אותו לפתרון בעיה חדשה. שתי אפשרויות עומדות בפנינו כאשר אנו מעוניינים להשתמש באלגוריתם קיים. האפשרות הראשונה היא לבצע מהלך מקדים על הבעיה, כך שלאחר השלב הזה הבעיה תראה ותתנהג כמו הבעיה שבאלגוריתם שלה אנו מעוניינים להשתמש. האפשרות השניה היא לעשות שינוי באלגוריתם עצמו, כך שאחרי השינוי הוא יהיה מסוגל לפתור בעיות מהסוג שאנו מעוניינים בהם.
2. שילוב של מספר אלגוריתמים קיימים – ניתן לשלב בין מספר אלגוריתמים קיימים, כאשר כל אחד מהם מטפל בחלק אחר של הבעיה, על מנת לקבל פתרון מלא. עבור בעיית ניתוב הרכבים המאופיין במשכי נסיעה תלויי זמן וסטוכסטיים, ניתן לנסות לשלב בין אלגוריתמים לפתרון VRP סטוכסטי לאלגוריתמים לפתרון TDVRP דטרמיניסטי.
3. שימוש במטה-היוריסטיקות - קימות היום מספר מטה-היוריסטיקות המשמשות לפתרון VRP (את חלקן סקרנו בפרק הסקירה הספרותית). ישנה אפשרות לפתח אלגוריתם מבוסס על מטה-היוריסטיקות, כאשר ניתן להשתמש בין היתר במטה-היוריסטיקות הבאות:
   - Ant Algorithms
   - Constraint Programming
   - Deterministic Annealing
   - Genetic Algorithms
   - Simulated Annealing
   - Tabu Search

בעבודה זו האלגוריתם שיוצג הוא אלגוריתם המבוסס על אלגוריתם Savings ([16]), אשר עבר התאמה והרחבה על מנת שיוכל להתמודד עם תלות בזמן ועם סטוכסטיות. ישנם מספר תנאים בסיסיים אשר צריכים להתקיים על מנת שנוכל להשתמש באלגוריתם זה. התנאים הנם:

1. מספר הלקוחות הוא קבוע.
2. הדרישה של כל לקוח ידועה וקבועה.
3. דרישת כל אחת מהלקוחות קטנה מיכולת ההובלה של כלי רכב בודד בצי.
4. סך כל הדרישות של כל הלקוחות גדול מיכולת ההובלה של כלי רכב בודד בצי.
5. כל כלי הרכב בצי הם בעלי אותה קיבולת.
6. כל כלי הרכב נעים באותו פרק זמן ביום (לדוגמא, כל כלי הרכב נעים בין 8 בבוקר ל-8 בערב).
7. פרק הזמן ביום הוא מחזורי, כלומר כלי רכב שלא סיים את המסלול שלו בסוף פרק הזמן הנוכחי, ממשיך את המסלול שלו באותה הנקודה בתחילת פרק הזמן הבא.
8. מספר תת-פרקי הזמן הוא קבוע עבור כל הקשתות בגרף.
9. אורכי כל תת-פרקי הזמן שווה.
10. עבור כל קשת בגרף, נתונים לנו אורך הקשת ועבור כל פרק זמן נתונות פונקצית מהירות ופונקצית ההתפלגות שלה. אם אנו מקבלים נתונים אימפריים, אז סך ההסתברויות עבור כל יחידת זמן צריך להסתכם ל-1, כלומר, צריך להתקיים $\forall t \in \{t_1, t_2, ..., t_n\} \quad \sum_{i=1}^{m} p_i^t = 1$.

4.2.1. הסיבות לבחירת אלגוריתם Savings

ישנן שתי סיבות עיקריות לבחירת אלגוריתם Savings כאלגוריתם הבסיסי עליו מבוסס האלגוריתם שפותח בעבודה זו.

הסיבה הראשונה היא פשטותו של האלגוריתם. הרעיון העומד מאחורי האלגוריתם הוא איחוד זוג מסלולים למסלול אחד חדש בהתאם להיקף החיסכון שיתקבל מאיחוד אותם זוג המסלולים. אלגוריתם Savings הוא אלגוריתם חמדני מבחינה זו שבכל איטרציה הוא בוחר לאחד את זוג המסלולים שבאותו השלב הוא בעל החיסכון הגדול ביותר. בשל פשטותו, מרבים להשתמש בו הן מעשית והן אקדמאית, והוא נלמד ומוצג ראשון כאלגוריתם לפתרון בעיית ניתוב רכבים.

הסיבה השנייה היא חשיבותו האקדמאית של האלגוריתם. אלגוריתם Savings הוצג לראשונה בשנת 1964, ומאז הוא שימש כבסיס לאלגוריתמים שונים לפתרון הרחבות שונות של בעיית ניתב הרכבים. בעבודת ההכנה שנעשתה עבור עבודת מחקר זה יותר מ-25% מהמאמרים שנסקרו כללו התייחסות לאלגוריתם Savings. בין המאמרים שנסקרו נתן למצוא גרסאות של אלגוריתם Savings לפתרון SVRP ,MDVRP ,VRPTW ,SDVRP ,CVRP ועוד'.

Gendreau, Potvin ו-Semet ([18]) ביצעו השוואה בין מספר אלגוריתמים נפוצים, ובין היתר הראו שסטיית אלגוריתם Savings מהפתרון האופטימאלי (או הפתרון הטוב ביותר הידוע), עומדת בממוצע על 6%. Ballou ו-Agrawal ([4]) ביצוע השוואה של מספר אלגוריתמים נפוצים

לפתרון בעיות VRP. מסקנתם הייתה שמבין האלגוריתמים שהם בחנו אלגוריתם Savings סיפק את התוצאות הטובות ביותר. איכות תוצאת האלגוריתם ביחס לפשטותו מהווים סיבה נוספת לבחור באלגוריתם כבסיס לאלגוריתם חדש.

4.2.2. אלגוריתם STDVRP

כאמור, האלגוריתם המוצע בעבודה זו (אנו נתייחס לאלגוריתם זה בשם אלגוריתם STDVRP) מבוסס על אלגוריתם Savings. להלן תזכורת קצרה לגבי אופן פעולתו של אלגוריתם Savings.

1. תחילה יש לבנות אוסף של מסלולים ראשוניים. המסלולים הראשוניים הם מסלולים היוצאים מהמחסן, עוברים דרך לקוח בודד וחוזרים למחסן. אין שני מסלולים אשר עוברים דרך אותו הלקוח, ואין לקוח אשר לא קיים עבורו מסלול.
2. אנו יוצרים רשימה המכילה את החיסכון המתקבל מאיחוד כל זוגות המסלולים האפשריים. איחוד מתקבל ע"י חיבור הקודקוד האחרון של המסלול הראשון לקודקוד הראשון של המסלול השני. הרווח מחושב באופן הבא: $s = C_{i0} + C_{0j} - C_{ij}$. הרשימה ממוינת כך שבתחילת הרשימה נמצא החיסכון הגבוה ביותר, במקום השני נמצא החיסכון השני בגובהו וכך הלאה.
3. את זוג המסלולים שהחיסכון המתקבל מחיבורם הוא המקסימאלי (וגדול מ-0) ואשר חיבורם אינו מפר אילוצים שונים של הבעיה, אנו מחברים למסלול אחד.
4. אנו חוזרים על שלב 3 כל עוד אפשר.

החלק העיקרי של האלגוריתם Savings הוא חישוב רשימת החיסכון המתקבל מחיבור מסלולים. חישוב החיסכון מתבצע פעם אחת בלבד, דבר שמאפשר פעולה מהירה יותר של האלגוריתם. הוספה של תלות כלשהי, כגון תלות בזמן, הופכת את חישוב החיסכון לפעולה מסובכת יותר, אשר, לרוב, לא ניתנת לחישוב באופן חד פעמי.

החיסכון, כאמור, מוגדר כזמן הנחסך מהצורך לחזור למחסן מהקודקוד האחרון במסלול הראשון וכן מהמעבר מהמחסן לקודקוד הראשון של המסלול השני, עקב חיבור הקדקודים האחרון במסלול הראשון והראשון במסלול השני, וקבלת מסלול אחד חדש.

נניח וזמני הנסיעה במסלול הראשון והשני היו ידועים לנו, והחלטנו לחבר את שני המסלולים למסלול אחד. ברגע שביצענו את החיבור, משתנה זמן ההתחלה של המסלול השני, ואין אנו יכולים להגדיר את זמן הנסיעה במסלול החדש כסכום זמני הנסיעה של שני המסלולים המקוריים. בעקבות זה, גם רשימת החסכונות שנבנתה משתנה ויש לחשבה מחדש (לפחות עבור כל חסכונות שקשורים למסלול החדש שנבנה), ולצורך זה יש לחשב את זמן הנסיעה במסלול החדש. הוספה של אקראיות למערכת אף מקשה על חישוב זמני הנסיעה.

בעבודה זו אנו נשתמש בשני כלים בכדי להתמודד עם האקראיות ועם התלות בזמן. הכלי הראשון הוא המרת הבעיה מבעיית תלויית זמן וסטוכסטית לבעיה דטרמיניסטית, והכלי השני הוא שימוש בסימולציה על מנת לחשב את זמני הנסיעה במסלול שהינו תלוי בזמן ומתנהג סטוכסטית.

4.2.2.1 המרת הבעיה מבעיה תלויית זמן וסטוכסטית לבעיה דטרמיניסטית

השלב הראשון שלנו הוא להפוך את הבעיה מבעיה תלויית זמן וסטוכסטית, לבעיה דטרמיניסטית. הפיכת הבעיה מתלויית זמן וסטוכסטית לבעיה דטרמיניסטית, תאפשר לנו להשתמש באלגוריתם Savings כמות שהוא. יש לציין, שהמעבר מבעיה תלויית זמן וסטוכסטית לבעיה דטרמיניסטית אינה מאפשרת לפתור את הבעיה, אלא היא עוזרת לנו ומשמשת ככלי לקבלת החלטה עבור אלגוריתם Savings.

בעבודה זו אנו נשתמש בשלוש שיטות על מנת להפוך את הבעיה תלויית הזמן והסטוכסטית לבעיה דטרמיניסטית.

השיטות הללו הן:

1. **חישוב ממוצע** – חישוב הממוצע מתבצע בשני שלבים. בשלב הראשון, עבור כל יחידת זמן, אנו מחשבים את ממוצע מהירות הנסיעה בקטע ביחידת הזמן. השלב השני הוא חישוב הממוצע הכללי, כלומר, עבור כל קטע אנו סוכמים את המהירויות הממוצעות שחושבו עבור כל אחת מיחידות הזמן, ומחלקים במספר יחידות הזמן.
2. **חישוב הערך הטוב ביותר** – חישוב הערך הטוב ביותר הוא למעשה חיפוש המהירות המהירה ביותר האפשרית בין כל יחידות הזמן, ללא התחשבות בהסתברות לקבלת אותה המהירות. אם נסתכל על התהליך כתהליך דו-שלבי, הרי שבשלב הראשון לכל קטע זמן אנו נחפש את המהירות המקסימאלית האפשרית, ובשלב השני, מבין כל המהירויות המקסימאליות אנו נבחר את המהירות הגדולה ביותר.
3. **חישוב הערך הגרוע ביותר** – בדומה לחישוב הערך הטוב ביותר, חישוב הערך הגרוע ביותר הוא חיפוש המהירות הנמוכה ביותר האפשרית בין כל יחידות הזמן, גם כאן אין אנו מתחשבים בהסתברות לקבלת אותה המהירות. כתהליך דו-שלבי, בשלב הראשון לכל קטע זמן אנו נחפש את המהירות המינימאלית האפשרית, ובשלב השני, מבין כל המהירויות המינימאליות אנו נבחר את המהירות הקטנה ביותר.

4.2.2.2 פונקצית זמן הנסיעה

פונקצית העלות, $C(i, j,t) = c(i, j,t) + \xi(i, j,t)$, מוגדרת כאמור עבור כל אחת מהקשתות של הגרף, והיא תלויה בזמן. בעבודה זו, העלות מוגדרת כזמן הנסיעה בקטע, וכאשר אנו מדברים על מציאת מסלולים בעלי עלות כולל מינימאלית, אנו למעשה מדברים על מציאת מסלולים שסך זמן הנסיעה בהם הוא מינימאלי.

בעת מימוש האלגוריתם, ניתן להגדיר את פונקצית העלות, כלומר את פונקצית זמן הנסיעה בקשת, במספר אופנים. בעבודה זו פונקצית העלות מוגדרת באופן הבא. עבור כל קשת בגרף נתון לנו המרחק של אותה הקשת. מכוון שאנו מתייחסים לגרף מכוון, הרי שמרחק הקשת (i, j) שונה ממרחק הקשת (j,i). את נתוני המרחקים של הקשתות השונות ניתן לשמור במטריצת שכנות (Adjacency matrix), כפי שהדבר נעשה במימוש האלגוריתם בעבודה זו. נתון נוסף שיש לשמור

עבור כל קשת הוא מהירות הנסיעה בקשת. מהירות הנסיעה בקשת תלויה בזמן שבו עוברים בקשת. כמו כן, יש לקחת בחשבון גם את ההתנהגות הסטוכסטית של מהירות הנסיעה.

קודם לכן ראנו כיצד ניתן להפוך את משתנה הזמן הרציף למשתנה דיסקרטי. בכדי שנוכל להשתמש במשתנה דיסקרטי במקום משתנה רציף, אנו צריכים לחלק את טווח הזמן למספר רב של חלקים שווים, כך שכל אחת מהחלקים יהיה קטן כל כך, עד שישאף ל-0. כך אנו יכולים להחזיק רשימה של מהירויות (ובנוסף פונקצית התפלגות מסוימת עבור החלק הסטוכסטי של המהירות) עבור כל קשת, שתלויה בשעת התחלת הנסיעה בקטן. ברוב המקרים, לא יהיה שינוי במהירות הנסיעה בקטע בין הזמן t_i לבין הזמן t_{i+1}, כאשר $t_{i+1}-t_i \approx 0$, כלומר, כשהפרשי הזמנים קטנים מאד, ולכן לצרכים פרקטיים אנו מחלקים את טווח הזמן למספר חלקים קטן, שההפרש ביניהם יחסית גדול.

בהתאם למודל שהוצג בסעיף 4.1, משתנה ההחלטה x_{ij}^{tv} מקבל את הערך 1 כלי רכב v מתחיל לנוע בקשת בקטע הזמן $[t,t+1)$, ו-0 אם לא. הבעייתיות בהצגה כזו, היא באופן חישוב פונקצית המטרה, $\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{t=0}^{T}\sum_{v=0}^{V}\bar{C}_{ij}^{\,t}x_{ij}^{tv}$, אשר אינה לוקחת בחשבון את זמן התחלת התנועה בקשת המדויקת. המשמעות היא, שאם אנו נכנס לקשת בנקודת זמן מסוימת הנמצאת בטווח $[t,t+1)$, אך אנו לא נוכל לסיים את התנועה בקשת בטווח הזמן $[t,t+1)$, ונצטרך לנוע בקשת גם בטווח הזמן $[t+1,t+2)$, ומהירות הנסיעה בשני טווחי הזמן היא שונה, זמן הנסיעה $c_{ij}^{\,t}x_{ij}^{tv}$ יהיה שונה מזמן הנסיעה בפועל.

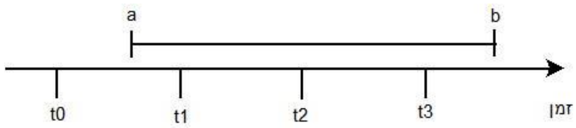
בכדי להתגבר על הבעיה הזאת, אנו נציג פונקצית זמן משופרת, אשר בה נעשה שימוש באלגוריתם STDVRP, שמומש בעבודה זו. פונקצית זמן הנסיעה המשופרת, מקבלת כפרמטר את זמן תחילת התנועה בקשת, ומחזירה את הזמן שייקח לנוע באותה הקשת.

```
1.  TempLenght = Edge.Length
2.  Index = ⌊StartTime*TotalTime/NumberOfTimePeriods⌋
3.  TimeLeft = (TotalTime/NumberOfPeriods)*(Index+1)-StartTime
4.  TravelTime = 0
5.  Start loop
6.    Speed=Get random speed of edge in time period = Index
7.    TravelTime=TravelTime+TempLength/Speed
8.    if TempLength/Speed <= TimeLeft then
9.      exit loop and return TravelTime as edge travel time
10.   TempLength=TempLength-(TimeLeft*Speed)
11.   Index=Index+1
12.   TimeLeft=TotalTime/NumberOfPeridos
13.   repeat loop
```

אנו משתמשים במספר משתני עזר על מנת לחשב את זמן הנסיעה בקשת. בהתחלה המשתנה TempLenght מכיל את אורך הקשת (שורה 1). המשתנה Index מכיל את קטע הזמן בו אנו נמצאים (שורה 2). את קטע הזמן בו אנו נמצאים ניתן לחשב באופן פשוט. אם את סך הזמן שלנו

אנו מציינים כ-TotalTime ואת מספר קטעי הזמן שלנו אנו מציינים כ-NumberOfTimePeriods, הרי שאורכו של קטע זמן שווה ל-TotalTime חלקי NumberOfTimePeriods. אנו מציינים כ-StartTime את הזמן שבו אנו מתחילים לנוע בקשת. אם נחלק את StartTime באורכו של קטע זמן, הרי שהחלק השלם של התוצאה הוא מספר קטע הזמן בו אנו מתחילים לנוע בקשת (שורה 2). אם, כאמור, Index הוא מספר קטע הזמן שבו אנו מתחילים לנוע בקשת, אז Index+1 הוא מספר קטע הזמן הבא בתור. אם נכפיל את Index+1 באורכו של קטע זמן, ונפחית ממנו את StartTime, נקבל את הזמן שנשאר לנו לנוע באותו קטע זמן. אנו מסמנים את אותו פרק זמן כ-TimeLeft (שורה 3). משתנה העזר האחרון שאנו צריכים הוא TravelTime. משתנה זה יכיל את זמן התנועה בקשת המחושב ע״י הפונקציה, והוא מאותחל בערך 0 (שורה 4). לאחר שיש בידינו את הערכים המתאימים במשתני העזר, אנו יכולים להתחיל לחשב את הזמן שייקח לנו לנוע בקשת, כאשר אנו מתחילים בזמן StartTime. אנו מחשבים את מהירות הנסיעה (Speed) בקשת בפרק הזמן Index, החישוב נעשה בעזרת פונקצית עזר אותה נתאר בהמשך (שורה 6). זמן התנועה בקשת בשלב זה שווה ל-TravelTime אליו אנו מוסיפים את TempLength חלקי Speed, כלומר זמן התנועה בקשת שווה לזמן התנועה עד כה בתוספת זמן חדשה אשר שווה לאורך הקטע הנותר חלקי המהירות (שורה 7). התוספת שהתווספה, TempLenght חלקי Speed, היא למעשה זמן הנסיעה בקטע TempLength (אורך הקטע בקשת שעוד לא נעו בו). אם הזמן הזה קטן או שווה לזמן שנותר לנו לנוע בפרק הזמן שבו אנו נמצאים, הרי שבשלב זה יש בידנו את זמן הנסיעה בקשת (שורות 8 ו-9). אך אם אנו גולשים לפרק הזמן הבא, אנו צריכים לבצע תיקון, שכן המהירות בפרק הזמן הבא כבר שונה. על כן, אנו מתקנים את המשתנה TempLength, כך שיכיל את אורך המסלול שנשאר לאחר הורדת הקטע שאותו ניתן לעבור בפרק הזמן הנוכחי (שורה 10), אנו מגדילים ב-1 את מונה קטע הזמן, Index (שורה 11), ומשנים את המשתנה TimeLeft, כך שעכשיו הוא שווה לערכו של פרק זמן שלם (שורה 12). לאחר שינויים אלו, אנו מבצעים את החישוב מחדש (שורות 5 עד 13).

כדי להבין את הפונקציה בצורה טובה יותר ופשוטה יותר, נסתכל על התרשים הבא.

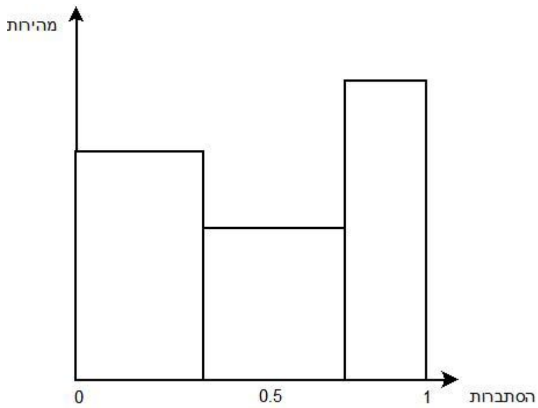


תרשים 1 - פונקציית זמן נסיעה בקשת

התרשים מתאר קשת אשר מתחילים לנוע בה בנקודה a, כאשר $t_0 < a < t_1$, ומסיימים לנוע בה בנקודה b, כאשר $b > t_3$. כדי לחשב את זמן הנסיעה בפונקציה אנו צריכים לדעת מהוא אחוז הזמן שבו אנו נמצאים בפרק הזמן $[t_0, t_1)$, אותו אנו נכפיל במהירות הנסיעה בפרק זמן זה. אל התוצאה המתקבלת אנו מוסיפים את אחוז הזמן שבו אנו נמצאים בפרק הזמן $[t_1, t_2)$ מוכפל

במהירות הנסיעה בפרק זמן זה, ואת אחוז הזמן שבו אנו נמצאים בפרק הזמן $[t_2,t_3)$ אף הוא מוכפל במהירות הנסיעה בפרק זמן זה. ולבסוף אנו מוסיפים את אחוז הזמן שבו אנו נמצאים בפרק הזמן $[t_3,...)$, שגם אותו אנו מכפילים במהירות הנסיעה באותו פרק זמן. אם עבור הקשת (i,j) נסמן את אחוז הזמן שבו אנו בפרק הזמן $[t_k,t_{k+1})$ כ-${Tp^{i,j}}_{k,k+1}$ ואת מהירות הנסיעה בפרק זמן זה כ-${s^{i,j}}_{k,k+1}$ הרי שזמן הנסיעה בקשת הוא $\sum_{k=0}^{T-1} {Tp^{i,j}}_{k,k+1}{s^{i,j}}_{k,k+1}$

נשאר לנו לדון באופן קבלת מהירות הנסיעה בקטע באופן שהוא תלוי זמן ומתנהג סטוכסטית. באלגוריתם המוצע, לכל קשת מצורף, כאמור, מבנה נתונים המכיל מידע על הקשת. בין המידע ניתן למצוא את אורך הקשת, את פונקצית מהירות וכן את פונקצית ההתפלגות שלה. מכוון שאנו רוצים לקבל כל פעם מהירות נסיעה בקטע זמן מסוים וידוע, אנו ראה כיצד זה נעשה. בעזרת פונקצית המהירות ופונקצית ההתפלגות אנו בונים טבלה המכילה נתונים אימפריים של מהירויות וההסתברות לקבל אותן. התרשים הבא מתאר טבלה בת שלוש מהירויות וההסתברות לקבל את אותן המהירויות.



תרשים 2 - מהירויות וההסתברות לקבלת ליחידת זמן

ציר ה-X ומציין את ההסתברות, והוא נע בין 0 ל-1, וציר ה-Y מציין את המהירות. כפי שניתן לראות, סך ההסברות הוא 1. בכדי לקבל מהירות כל שהיא באופן אקראי ובהתאם להסתברויות המתוארות, אנו צריכים להגריל מספר כלשהו בין 0 ל-1 באופן אקראי. הגרלת מספרים באופן אקראי נעשת באופן פשוט ע״י מחשב, בעזרת פונקציה המספקת מספרים פסאודו-אקראיים (מספרים פסאודו-אקראיים הם מספרים אשר נראים אקראיים למרות שהם מסופקים ע״י פונקציה מתמטית, ומתאימים ליישומים סטטיסטיים). אם הטבלה מכילה m מהירויות, וההסתברות לקבלת המהירות הראשונה, s_1, שווה ל-p_1-0, באופן דומה, ההסתברות לקבלת המהירות השניה, s_2, שווה ל-p_2-p_1. כך מחושבים שאר המהירויות, כאשר ההסתברות לקבלת המהירות האחרונה, s_m, שווה ל-$1-p_{m-1}$. נניח וקיבלנו את ההסברות P, באופן אקראי. אנו מתחילים ובודקים, אם מתקיים $0 \le P \le p_1$, אז המהירות שלנו היא s_1, באופן דומה, אם

מתקיים $p_1 < P \le p_2$, אז המהירות שלנו היא s_2, כך לגבי שאר המהירויות כאשר עבור המהירות האחרונה, s_m , התנאי הוא $p_{m-1} < P \le 1$.

4.2.2.3 סימולציה

סימולציה הנה טכניקה שמשתמשת במחשב על מנת לחקות פעולות במציאות מורכבת. מטרת הסימולציה היא לייצג מאפיינים מסוימים בהתנהגות של מערכת על מנת לקבל תחזיות לגבי התנהגות המערכת בתנאים שונים. שימוש בסימולציה נעשה לרוב כאשר השימוש בתהליך האמיתי הנו יקר מאד או מסכן חיי אדם. בין המקרים שבהם משתמשים בסימולציה אפשר למצוא סימולציה של פיצוץ גרעיני (פיצוץ גרעיני אמיתי יקר ומסכן חיי אדם), בדיקת השערות שאי אפשר לאמתם בניסוי, כגון השערת המפץ הגדול, אימון טייסים בעזרת סימולטור (חוסך שעות טיסה יקרות) וכדומה.

סוג נוסף של מערכות שבהן נהוג להשתמש בסימולציה הן מערכות המתנהגות באופן סטוכסטי. מערכות סטוכסטיות הן מערכות שיש בהן מרכיב של אקראיות, כך שמערכת בתנאים מסוימים יכולה לפעול בדרכים שונות. בעזרת מחשב ניתן לבנות מודל אשר מדמה את המערכת, ואשר כולל את מרכיב האקראיות. ע״י הפעלת המודל על-גבי מחשב מספר רב של פעמים, ניתן לראות כיצד המערכת פועלת, וכן לנתח את פעולתה. ניתוח פעולת המערכת יכול להצביע על נטייה של המערכת לפעול בצורה מסוימת באחוז מסוים של המקרים, ובצורה אחרת בשאר המקרים. במידה ואכן מתקיים מצב כזה, יותר קל לצפות את התנהגות המערכת. ביצוע סימולציה חוסך זמן רב וכסף רב, בהשוואה להפעלת ובחינת המערכת האמיתית.

סימולציה בעזרת מחשב, מאפשרת לנו לבצע פעולת שבמציאות עלולות לקחת זמן רב ביותר, לעיתים אפילו שנים. היא יכולה לשמש גם ככלי לקבלת החלטות, שכן ע״י סימולציה של מספר מודלים אפשריים, אנו יכולים לדעת מי מהמודלים הוא היעיל יותר בתנאים הנתונים.

בעבודה זו נעשה שימוש בסימולציה על מנת לדעת מהו זמן המעבר במסלול נתון. עבור כל מסלול נתון, נתונים לנו הנתונים הבאים: סדר הקשתות במסלול, אורך כל קשת וכן פונקצית זמן הנסיעה בקשת, כאשר פונקצית זמן הנסיעה בקשת היא תלויית זמן וסטוכסטית.

נניח שהמסלול שלנו הוא $e_1, e_2, ..., e_n$, סימולציית מעבר במסלול זה נעשת באופן הבא: אנו מתחילים את הסימולציה בזמן $t = 0$, בעזרת המחשב אנו בוחרים אקראית מספר, p_1 , בין 0 ל-1, אנו מחשבים את זמן המעבר בקשת e_1 כאשר הזמן הוא t והסתברות היא p_1 , ומקבלים ערך מסוים, נניח t_1. אנו ממשיכים ומחשבים את זמן הנסיעה בקשת e_2, כאשר זמן תחילת התנועה בקשת הוא t_1, ויש לנו מספר אקראי חדש בין 0 ל-1, p_2 . זמן התנועה בקשת e_2 הוא t_2. אנו ממשיכים ומחשבים את זמן התנועה בשאר הקשתות של המסלול, כאשר עבור הקשת האחרונה, e_n, זמן תחילת התנועה בקשת הוא $t_1 + t_2 + ... + t_{n-1}$, ומספר האקראי שלנו בין 0 ל-1 הוא p_n . אנו מקבלים שזמן התנועה בקשת האחרונה, e_n, הוא t_n. זמן התנועה במסלול כולו הוא סכום זמני התנועה בכל אחת מהקשתות, והוא שווה ל- $t_1 + t_2 + ... + t_n$. בסעיף 4.2.2.2 הגדרנו את זמן

הנסיעה בקשת (i,j) כ- $\sum_{k=0}^{T-1} Tp^{i,j}{}_{k,k+1} s^{i,j}{}_{k,k+1}$, על סמך הגדרה זו ניתן להגדיר את זמן המעבר במסלול כ- $\sum_{l=0}^{n}\sum_{k=0}^{T-1} Tp^{e_l}{}_{k,k+1} s^{e_l}{}_{k,k+1}$.

אם נבצע את הסימולציה מספר רב של פעמים, ונשמור את התוצאה המתקבלת מכל אחת מהסימולציות, נוכל להשתמש בתוצאות הללו לשם ביצוע ניתוחים סטטיסטיים. בעבודה זו, אנו מעוניינים לדעת מהי המהירות המקסימאלית, שבאחוז מסוים מהמקרים (נניח 80%) נקבל את המהירות הזאת, או מהירות נמוכה ממנה. נניח וביצענו 100 סימולציות ואנו רוצים לדעת מהי המהירות המקסימאלית שב-80% מהמקרים לא נעבור אותה. בכדי לעשות זאת, אנו ממיינים את התוצאות של 100 הסימולציות שלנו, ובודקים מהוא הערך הנמצא במיקום מס׳ 80 ($100\cdot 80\% = 80$). הערך הזה הוא המספר אותו אנו מחפשים.

4.2.2.4 האלגוריתם

בשלב זה יש בידינו את כל הכלים הדרושים לנו על מנת להציג ולהבין את אלגוריתם STDVRP, המוצע בעבודה זו לפתרון בעיית ניתוב רכבים המאופיינים במשכי נסיעה תלויי זמן וסטוכסטיים.

ישנם מספר פרמטרים הבאים לידי ביטוי באלגוריתם STDVRP :

n – מספר הלקוחות.

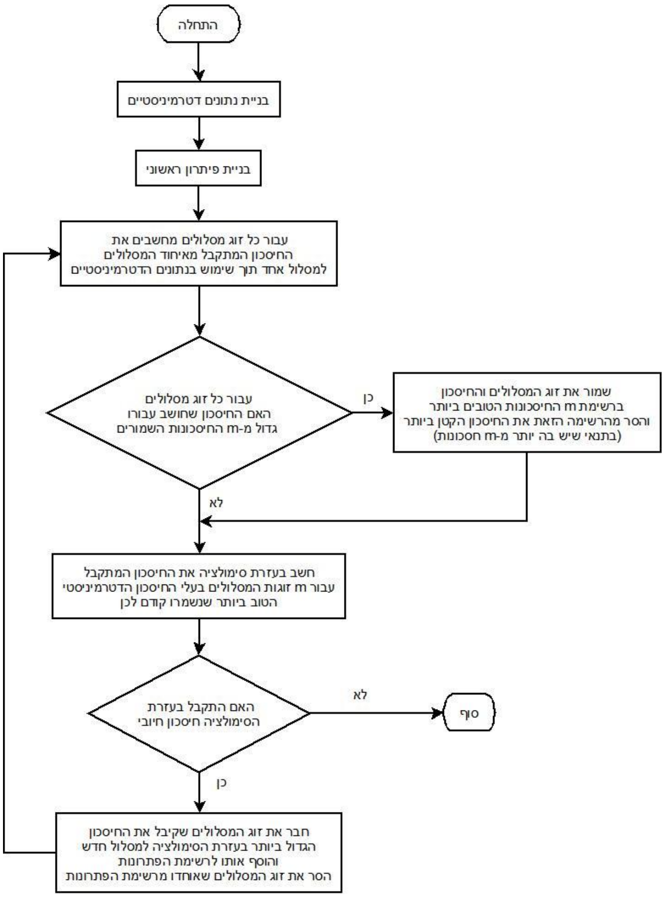
m – מספר זוגות המסלולים המועמדים לאיחוד, ושהחיסכון שלהם מחושב ע״י סימולציה.

r – מספר החזרות על האלגוריתם.

אלגוריתם STDVRP, כאמור, מבוסס על אלגוריתם Saving, ולכן, במבט ראשון הוא נראה דומה לו. להלן האלגוריתם.

1. בניית פתרון ראשוני. הפתרון הראשוני הוא בעל n מסלולים. כל מסלול יוצא מהמחסן, עובר דרך לקוח בודד, וחוזר למחסן. בין n המסלולים, אין שני מסלולים אשר עוברים דרך אותו הלקוח.
2. עבור כל זוג מסלולים, אנו מחשבים את הרווח המתקבל מחיבורם ע״י שימוש בנתונים דטרמיניסטיים.
3. עבור m זוגות המסלולים, שהרווח המתקבל מחיבורם הוא הגדול היותר, ואשר עומדים באילוצים השונים של הבעיה, אנו מחשבים את הרווח האמיתי אשר יתקבל מחיבורם ע״י שימוש בסימולציה.
4. את זוג המסלולים אשר קיבל בעזרת סימולציה את הרווח הגדול ביותר אנו מחברים למסלול חדש.
5. אנו חוזרים על שלבים 2 עד 4 כל עוד קיימים זוגות של מסלולים אשר חיבורם מהווה רווח ״חיובי״, והם עונים על תנאי הבעיה.

להלן תרשים זרימה המתאר את אופן פעולת האלגוריתם.



תרשים 3 - תרשים זרימה המתאר את אופן פעולת האלגוריתם

לפני הפעלת האלגוריתם יש להפוך את הבעיה מבעיה סטוכסטית תלויית זמן לבעיה דטרמיניסטית. לנו, כאמור, יש שלוש שיטות לעשות זאת. תחילה, אנו משתמשים בשיטת הממוצע, ומריצים את האלגוריתם. את התוצאה המתקבלת מריצת האלגוריתם ומשימוש בשיטת הממוצע, אנו שומרים. כעת, אנו הופכים את הבעיה לדטרמיניסטית ע״י שימוש בשיטת הערך הטוב ביותר. אנו מריצים את האלגוריתם, ובמידה ומתקבלת כעת תוצאה טובה יותר, אנו שומרים אותה בתור הפתרון שלנו. כשלב אחרון, אנו הופכים את הבעיה לדטרמיניסטית ע״י שימוש בשיטת הערך הגרוע ביותר. לאחר הרצת האלגוריתם, אנו משווים את התוצאה המתקבלת לעומת התוצאה הטובה ביותר שיש לנו, ובמידה ומתקבלת תוצאה טובה יותר, היא תהווה את הפתרון הסופי של האלגוריתם.

כאמור, הפיכת הבעיה מתלויית זמן וסטוכסטית לדטרמיניסטית, משמשת אותנו לקבלת החלטות בלבד, ולא לצורך חישוב הפתרון עצמו. מכוון שהמעבר מתלות בזמן וסטוכסטיות משנה את זמני המסלולים, ואין אנו יכולים לחשב לפיו את החיסכון האמיתי, אנו משתמשים בסימולציה על מנת לדעת לאילו מבין המסלולים יש את החיסכון האמיתי הגדול ביותר. בעזרת הנתונים הדטרמיניסטיים, אנו מחשבים את החסכונות המתקבלים מחיבור שני מסלולים. עבור

m (מספר שנקבע קודם לכן) זוגות המסלולים שעבורם התקבל החיסכון הדטרמיניסטי הגדול ביותר, אנו מחשבים את החיסכון האמיתי ע״י סימולציה. חישוב החיסכון האמיתי נעשה ע״י חישוב זמנו של המסלול הראשון בעזרת סימולציה, אליו אנו מחברים את זמנו של המסלול השני, אותו גם חישבנו בעזרת סימולציה. מסכום זה אנו מחסרים את זמנו של המסלול המאוחד, שכאמור, גם הוא חושב בעזרת סימולציה.

באלגוריתם ה-Savings יכול להווצר מצב שבו עבור שני זוגות מסלולים מתקבל אותו החיסכון. במקרה כזה, אנו בוחרים את אחד הזוגות באופן אקראי. מכוון שבחירה של זוג כלשהו משפיעה על התוצאה הסופית, אנו יכולים לקבל תוצאות שונות כאשר נחבר מסלולים שונים. מסיבה זו, אם אנו נתקלים במצב שבו אנו צריכים לבחור אקראית בין שני זוגות מסלולים, אנו חוזרים על האלגוריתם מספר קבוע של פעמים, r, בכדי שנוכל כל פעם לבחור בין זוג שונה של מסלולים.

4.3. ניתוח האלגוריתם

בשלב זה של העבודה אנו נציג ניתוח של האלגוריתם. הניתוח יתייחס לשני מאפיינים של האלגוריתם. המאפיין הראשון הוא סיבוכיות האלגוריתם, ואילו המאפיין השני הוא איכות הפתרונות המתקבלים מריצת האלגוריתם.

4.3.1. סיבוכיות האלגוריתם

סיבוכיות אלגוריתמית מתייחסת לתכונות אחדות של אלגוריתם, ובפרט קצב הגידול של זמן הביצוע שלו יחסית לגודל הקלט. את זמן הביצוע אין אנו בוחנים ביחידות זמן (שניות, דקות וכו׳), שכן משך הזמן לביצוע פעולות משתנה מסביבת ריצה אחת לשנייה, כמות ה״עבודה״ הניתנת לביצוע בצעד בודד עשויה להיות שונה בין מודלים חישוביים שונים - ובוודאי בין מחשבים שונים. הסימון הנפוץ ביותר לציון סיבוכיות אלגוריתם הוא ״$\boldsymbol{O}$״. בפשטות, אם זמן הפעולה של אלגוריתם מיוצג ע״י פונקציה כלשהי, $f(n)$, וקיימת פונקציה, $g(n)$, אשר חוסמת מלמעלה את פונקצית זמן הפעולה של האלגוריתם, אז סיבוכיות האלגוריתם הוא $\boldsymbol{O}(g(n))$. לרוב, את הסיבוכיות בודקים מול המצב הגרוע ביותר של האלגוריתם, כלומר, המצב שבו האלגוריתם יצטרך לעשות את מרב הפעולות על מנת להגיע לתוצאה.

מכוון שבאלגוריתם לפתרון בעיית ניתוב הרכבים תלויי זמן וסטוכסטיים שהצגנו ישנם מספר פרמטרים הניתנים לשינוי, אנו נבחן כיצד שינוי באותם הפרמטרים משפיע על סיבוכיות האלגוריתם.

נתבונן בחלק הראשי של האלגוריתם, כפי שהוא מובא להלן :

```
Generate initial solutions
Saved = True
while Saved=True
  Saved=False
  Saving = -1
  for i=1 to n
    if Solution(i).NumberOfNodes>0 then
      for j=1 to n
```

```
9.          if Solution(j).NumberOfNodes>0 and i<>j then
10.           Saving = Calculate saving using solutions i and j
11.           if Saving >= 0 and new path does not validate other constraints then
12.             Add new path to candidate list
13.    SearchCandidateList(i,j)
14.    if i<>-1 and j<>-1 then
15.      Marge Solution(i) and Solution(j)
16.      Clear Solution(j)
17.      Saved = True
```

להלן ניתוח המציג את השפעת מספר הלקוחות, n, על סיבוכיות האלגוריתם. בתחילת ריצת האלגוריתם אנו יוצרים סט של מסלולים שמהווים את הפתרון הראשוני (שורה 1). בנייה של מסלול בודד אינה תלויה במספר הלקוחות, והיא קבועה מבחינת זמן הביצוע, ולכן הסיבוכיות של בניית מסלול בודד היא $\boldsymbol{O}(1)$. מכוון שמספר המסלולים הראשוניים שאנו בונים, שווה למספר הלקוחות, הסיבוכיות של בניית המסלולים הראשוניים היא $n\boldsymbol{O}(1)$, השקול ל-$\boldsymbol{O}(n)$. לאחר שיש בידינו את סט הפתרונות הראשוניים, יש לחשב את החיסכון המתקבל מחיבור כל זוג פתרונות (שורות 6 עד 12). הפעולה של חישוב החיסכון, גם היא אינה תלויה במספר הלקוחות, ולכן הסיבוכיות שלה היא גם $\boldsymbol{O}(1)$. בהתחלה אנו צריכים לחשב את החיסכון המתקבל מחיבור n-1 מסלולים עם n-1 מסלולים שונים, שזו סיבוכיות של $\boldsymbol{O}(n^2)$. כל חיסכון שמתקבל ושהינו חיובי, אנו שולחים לפונקציה שתפקידה הוא לשמור את m הפתרונות הטובים ביותר (שורות 11 ו-12). אנו נבחן את הסיבוכיות של הפונקציה הנ"ל מאוחר יותר, אך לצורך המשך ניתוח האלגוריתם נניח שסיבוכיות הפונקציה היא $\boldsymbol{O}(a)$, ולכן הסיבוכיות של האלגוריתם בשלב זה היא $\boldsymbol{O}(n)+\boldsymbol{O}(n^2\cdot\boldsymbol{O}(a))$. לאחר שחישבנו ושמרנו את החסכונות, יש בידינו רשימה של m החסכונות הטובים ביותר. בתוך הרשימה הזאת אנו מחפשים את זוג המסלולים שהחיסכון המתקבל מאיחודם הוא מקסימאלי, וזאת בעזרת שימוש בסימולציה. גם בפונקציה זו אנו נדון מאוחר יותר, אולם לצורך המשך ניתוח האלגוריתם נניח שסיבוכיות הפונקציה היא $\boldsymbol{O}(b)$, ולכן הסיבוכיות של האלגוריתם בשלב זה הוא $\boldsymbol{O}(n)+\boldsymbol{O}(n^2\cdot\boldsymbol{O}(a))+\boldsymbol{O}(b)$. שאר הפעולות שנשאר לנו לעשות, חיבור המסלולים למסלול חדש (שורה 14), והסרת המסלולים הישנים מרשימת המסלולים (שורה 15), הם בעלי סיבוכיות של $\boldsymbol{O}(1)$. מכוון שלרוב סיבוכיות של $\boldsymbol{O}(1)$ אינה מתווספת לסיבוכיות הכללי, סיבוכיות האלגוריתם עד כה נשארת $\boldsymbol{O}(n)+\boldsymbol{O}(n^2\cdot\boldsymbol{O}(a))+\boldsymbol{O}(b)$. אנו צריכים לחזור על כל התהליך הזה, כל עוד ישנם זוגות מסלולים אשר ניתן לחבר אותם ולקבל חיסכון חיובי (שורות 3 ו-17). בהנחה שתנאי הבעיה מאפשרים זאת, אנו יכולים לחזור על התהליך הזה n פעמים, כלומר לחבר את כל המסלולים הראשוניים למסלול אחד ארוך. ומכאן שהסיבוכיות הכוללת של האלגוריתם היא $\boldsymbol{O}(n)+n(\boldsymbol{O}(n^2\cdot\boldsymbol{O}(a))+\boldsymbol{O}(b))$.

בכדי לדעת את הסיבוכיות הכוללת של האלגוריתם, אנו צריכים לחשב את הסיבוכיות של הפונקציה השומרת את m החסכונות הטובים ביותר. להלן הפונקציה.

```
if Saving>WorstSaving or ValuesInList<m then
  AddToList(Value)
  if ValuesInList>m then
```

```
4.      RemoveFromList first value in list
5.    WorstSaving = Value at the beginning of the list
```

ישנם שני מצבים שבהם נשמור ערך חדש ברשימה שלנו. המצב הראשון הוא, כשהרשימה לא מלאה, כלומר, לא שמרנו יותר מ-m חסכונות. המצב השני, כאשר אנו רוצים לשמור חיסכון שהוא יותר טוב מהחיסכון הכי נמוך ברשימה (שורה 1). בצורה הפשוטה ביותר, הרשימה שלנו בנויה מרשימה ממוינת של איברים, כאשר האיבר הטוב ביותר נמצא בסוף הרשימה, והאיבר הגרוע ביותר נמצא בתחילת הרשימה. הרשימה יכולה להכיל $m+1$ איברים. בכדי להוסיף איבר לרשימה, יש לעבור על כל איברי הרשימה ולבדוק היכן ברשימה יש להוסיף את האיבר החדש (שורה 2). בגלל אופן פעולת הרשימה, שמירת חיסכון היא פעולה בעלת סיבוכיות של $\boldsymbol{O}(m)$. במידה והרשימה מכילה בשלב זה $m+1$ איברים, יש להסיר את האיבר הגרוע ביותר (שורות 3 ו-4), פעולה זו היא גם בעלת סיבוכיות של $\boldsymbol{O}(m)$. אם כך, הסיבוכיות הכוללת של שמירת חיסון היא $\boldsymbol{O}(2m)$, לרוב אין זה נהוג להתייחס למקדמים, ולכן הסיבוכיות היא $\boldsymbol{O}(m)$. הסיבוכיות הכוללת של האלגוריתם בשלב זה היא $\boldsymbol{O}(n)+n(\boldsymbol{O}(n^2\cdot m)+\boldsymbol{O}(b))$.

נחשב את הסיבוכיות של הפונקציה המחשבת את החיסכון הטוב ביותר בעזרת סימולציה, מתוך הרשימה של m החסכונות הטובים ביותר. להן הפונקציה.

```
1.  Saving=-1
2.  SavedDiff=0
3.  SavingIndex=0
4.  for i=1 to BestSavingList.Count
5.    TempSaving = SimulateSaving using list(i).Path1 and list(i).Path2
6.    if Saving>=TempSaving
7.      if Saving=TempSaving
8.        if |SavedDiff|>|TempSaving-List(i).Saving|
9.          SavedDiff=TempSaving-List(i).Saving
10.         Saving=TempSaving
11.         SavingIndex=i;
12.       if |SavedDiff|=|TempSaving-List(i).Saving|
13.         Randomly choose
14.           SavedDiff=TempSaving-List(i).Saving
15.           SavingIndex=i;
16.     if Saving<>TempSaving
17.       SavedDiff = TempSaving-List(i).Saving
18.       Saving=TempSaving
19.       SavingIndex=i
```

אנו מתחילים ומחשבים את החיסכון המתקבל מחיבור זוג מסלולים למסלול אחד ע"י שימוש בסימולציה. הסיבוכיות של סימולציה תלויה באורכו של כל מסלול, ולכן עבור מסלול בגודל מקסימאלי היא $\boldsymbol{O}(n)$, ומכוון שאנו חוזרים על הסימולציה r פעמים, וכן מבצעים את זה 3 פעמים, פעם אחת עבור המסלול הראשון, פעם שניה עבור המסלול השני ובפעם השלישית עבור המסלול המאוחד, הרי שסיבוכיות הסימולציה היא $\boldsymbol{O}(3rn)$ (שורה 5). אם החיסכון החדש גדול מהחיסכון שיש לנו עד כה, אנו שומרים את החיסכון הזה (שורות 6 עד 11). אם החיסכון החדש

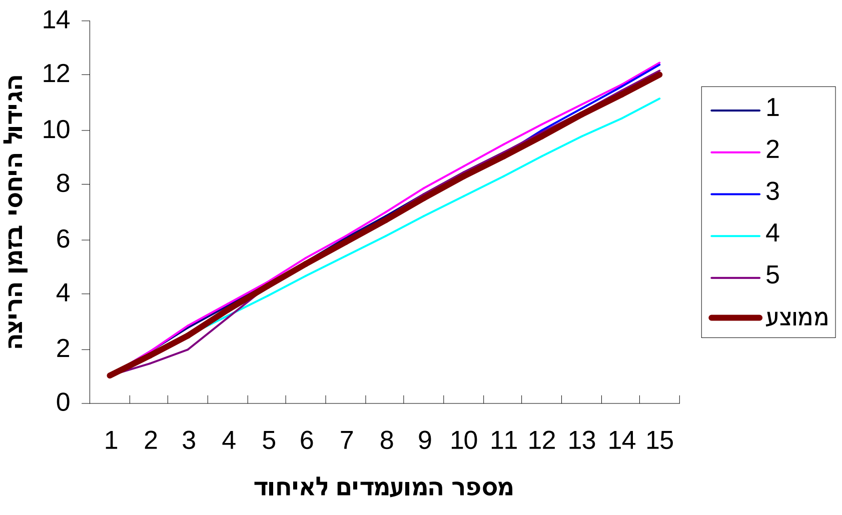
שווה לחיסכון שחישבנו עד לשלב זה אנו בודקים את ההפרש בין החיסכון שהתקבל מהסימולציה ובין החיסכון שהתקבל מהחישוב הדטרמיניסטי. אם ההפרש זהה (עבור זוג המסלולים החדש וזוג המסלולים שהיווה את החיסכון הטוב ביותר עד כה), אנו אקראית מחליטים אם החיסכון הזה יהיה החיסכון החדש (שורות 12 עד 14). אך הם שונים, סביר להניח שלזוג בעל ההפרש הגדול יותר יש יתרון בנקודת זמן אחרת, ולכן אנו שומרים הזוג שההפרש אצלו קטן יותר (שורות 8 עד 11). את החישוב הזה יש לבצע עבור m זוגות המסלולים, ולכן הסיבוכיות של הפונקציה היא $\boldsymbol{O}(3rnm)$.

הסיבוכיות הכוללת של האלגוריתם היא $\boldsymbol{O}(n)+n(\boldsymbol{O}(n^2\cdot m)+3rnm)$, כלומר $\boldsymbol{O}(n+n^3\cdot m+3rn^2m)$. סיבוכיות כזאת נהוג לרשום כ-$\boldsymbol{O}(n^3m)$, כלומר, הגורם המשפיע ביותר על סיבוכיות האלגוריתם הוא מספר הקדקודים בגרף והגורם השני בחשיבותו הוא מספר זוגות המסלולים אשר אנו שומרים בכדי לחשב את החיסכון המתקבל מחיבורם ע״י סימולציה.

בסעיף 4.3.2.21 מוצגות תוצאות של 20 בעיות שפתרונן הושג ע״י הרצה של האלגוריתם. עבור כל אחת מהבעיות הורץ האלגוריתם מספר פעמים, כאשר כל פעם מספר המסלולים שאת חסכונם שומרים ובודקים בעזרת סימולציה השתנה. מבין 20 הבעיות שנבדקו, 5 בעיות הכילו 50 לקוחות, 5 בעיות נוספות הכילו 75 לקוחות, עוד 5 בעיות הכילו 100 לקוחות וה-5 הבעיות הנותרות הכילו 150 לקוחות.

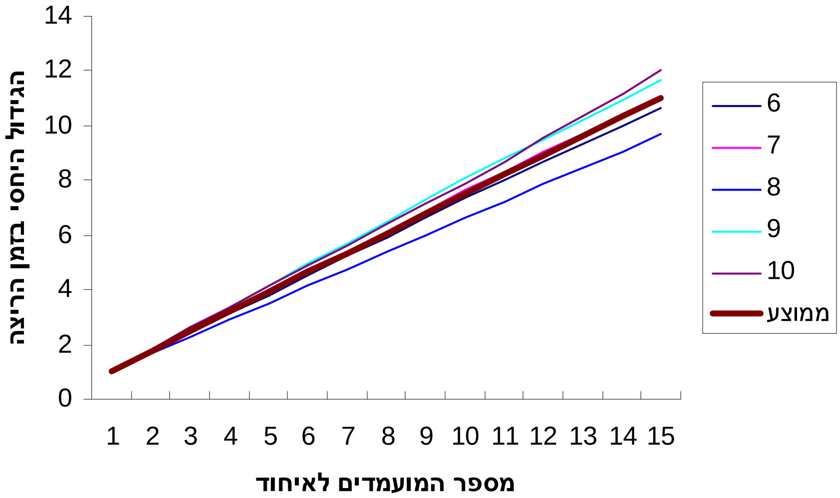
אנו, כאמור, פותרים כל בעיה מספר פעמים, כאשר בין כל פתרון ופתרון מבדיל מספר המסלולים שהחיסכון שלהם חושב ע״י סימולציה (m). על פי ניתוח האלגוריתם שהוצג, אנו מצפים שבעיה שבה m=10 למשל, תרוץ פי 10 יותר זמן מאשר בעיה שבה m=1.

כאמור יש לנו 4 קבוצות של בעיות, כאשר כל קבוצה נבדלת מהשניה במספר הלקוחות שלה. בכדי לראות אם זמני הריצה של אלגוריתם STDVRP, עבור ערכים שונים של m, מתאימים לתוצאות הצפויות מניתוח האלגוריתם, אנו נבחן את התוצאות שהתקבלו המריצות השונות. בכדי להקל על הצגת הנתונים, אנו נציג אותם על-גבי גרפים.



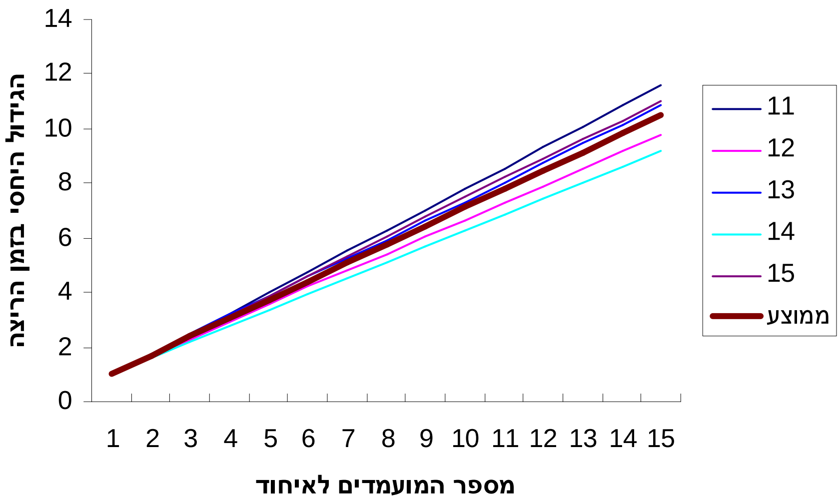
תרשים 4 - הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד בבעיות בנות 50 לקוחות

בכל קבוצת בעיות ישנן חמש בעיות, אותן פתרנו בעזרת האלגוריתם. בקבוצת הבעיות הראשונה ישנם 50 לקוחות. בכל אחת מהבעיות, זמן הריצה שהתקבל כאשר m=1, נחשב לנקודת הייחוס, ובשאר הריצות שבוצעו זמן הריצה מושווה ביחס לזמן הריצה שהתקבל עבור m=1. כפי שניתן לראות מהגרפים, זמן הריצה אינו משתנה באופן זהה עבור כל הבעיות. הסיבה לכך היא שישנם תנאי יציאה שונים, אשר באים לידי ביטוי בצורה שונה עבור כל בעיה ובעיה. אבל, עדיין, ניתן לראות בברור, שהשינוי בזמן הריצה של הבעיה, כאשר יש שינוי במספר זוגות המסלולים שהחיסכון שלהם הוא הגדול ביותר, והם מועמדים לאיחוד, הוא ליניארי. אנו ציפינו שעבור m=2, זמן הריצה יהיה כפול מזמן הריצה כאשר m=1, אבל בפועל נראה שבממוצע זמן הריצה הוא מעט פחות. עבור m=5 זמן הריצה הממוצע, גדול פי ארבעה מאשר זמן הריצה כאשר m=1, ואילו כאשר m=15 הוא גדול בממוצע, פי 12 ולא פי 15 כפי שאנו מצפים. כפי שכבר אמרנו, תנאי יציאה באלגוריתם, אשר מתרחשים כאשר מסלולים אינם עומדים באילוצים, הם הגורם לכך שזמני הריצה אינה מתנהגים כפי שצפוי.



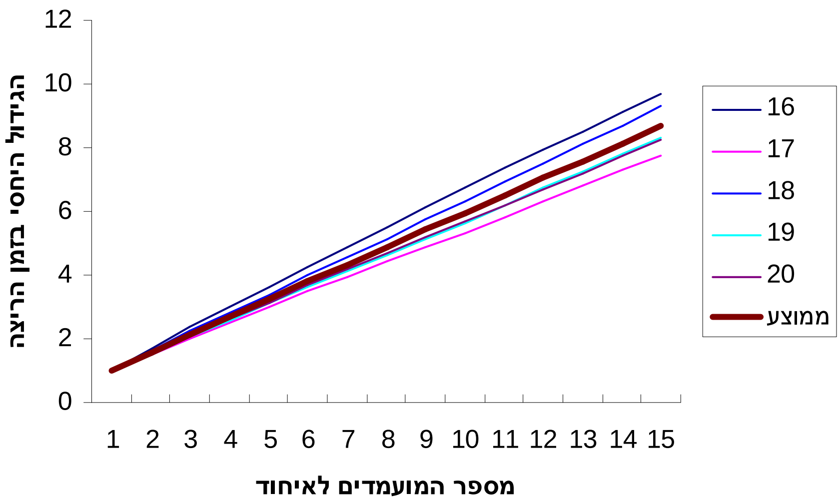
תרשים 5 - הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד בבעיות בנות 75 לקוחות

בקבוצת הבעיות השנייה ישנם 75 לקוחות. כפי שניתן לראות מהגרף, גם בקבוצה זו, זמן הריצה אינו משתנה באופן זהה עבור כל הבעיות, אך הוא משתנה ליניארית. גם עבור קבוצת הבעיות הזאת, התוצאות הנן נמוכות ממה שצפוי על סמך ניתוח האלגוריתם. כאשר m=15 זמן הריצה גדול פי 11 מאשר זמן הריצה כאשר m=1, זמן ריצה זה אף נמוך מזמן הריצה, פי 12, אשר התקבל כאשר היו לנו בעיות בנות 50 לקוחות.



תרשים 6 - הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד בבעיות בנות 100 לקוחות

בקבוצת הבעיות השלישית ישנם 100 לקוחות. גם בקבוצה זו זמן הריצה משתנה באופן ליניארי, וגם בקבוצה זאת, זמן הריצה לא משתנה כפי שצפוי סמך ניתוח סיבוכיות האלגוריתם. כאשר m=15 זמן הריצה גדול פי 10.5 בממוצע, מאשר זמן הריצה כאשר m=1, זמן ריצה זה אף נמוך מזמן הריצה, פי 12, אשר התקבל כאשר היו לנו בעיות בנות 50 לקוחות, ומפי 11, כאשר היו לנו 75 לקוחות.



תרשים 7 - הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד בבעיות בנות 150 לקוחות

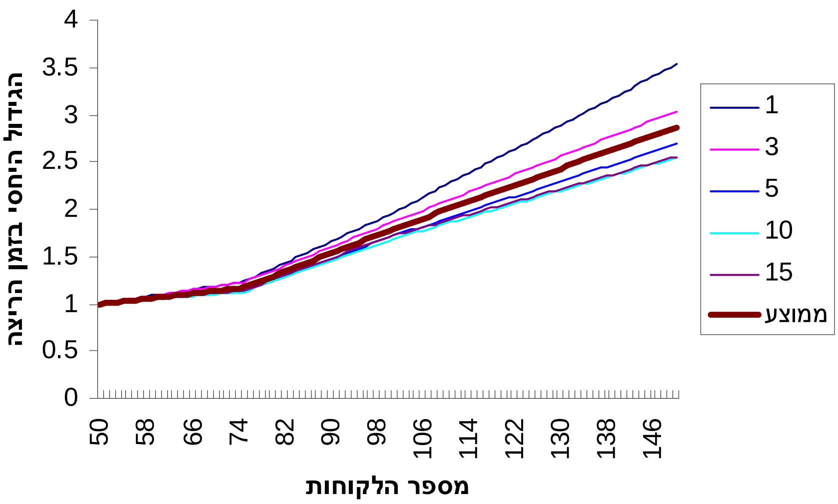
בקבוצת הבעיות הרביעית ישנם 150 לקוחות. גם בקבוצה זו, כפי שניתן לראות מהגרפים, זמן הריצה אינו משתנה באופן זהה עבור כל הבעיות, אך הוא משתנה ליניארית. גם עבור קבוצת הבעיות הזאת, התוצאות הנן נמוכות ממה שצפוי על סמך ניתוח האלגוריתם. אנו מניחים שהמגמה שנצפתה בקבוצות הקודמות ממשיכה גם עם קבוצה זו, וזמן הריצה היחסי יהיה אף קטן יותר ממה שהתקבל עד כה. ואכן, כאשר m=15 זמן הריצה גדול פי 8.5 מאשר זמן הריצה

כאשר m=1, זמן ריצה זה אף נמוך מזמן הריצה, של פי 10.5, אשר התקבל כאשר היו לנו בעיות בנות 100 לקוחות.

אמנם אנו ציפינו שזמן הריצה יגדל ביחס ישר ככל שמספר זוגות המסלולים המועמדים לאיחוד יגדל. אבל כאמור, בעת חישוב סיבוכיות אלגוריתם, אנו מחשבים את סיבוכיות האלגוריתם במצב הגרוע ביותר, המצב שבו האלגוריתם יעשה את מרב הפעולות לו הוא נדרש. בפועל, ככל שמספר הלקוחות גדול יותר, אנו רואים שמספר הפעולות שהאלגוריתם נדרש לעשות ביחס למספר הלקוחות הוא קטן יותר.

משתנה שהשפעתו על זמן ריצת האלגוריתם גדולה יותר הוא מספר הלקוחות (n). על סמך ניתוח סיבוכיות האלגוריתם, אנו יודעים שאם זמן הריצה עבור n=1 הוא x אז עבור n=2 זמן הריצה יהיה $8x$, ועבור n=3, $27x$, וכך הלאה.

אם זמן הריצה עבור n=1 הוא x, אנו מצפים שזמן הריצה עבור n=50 יהיה 50^3x. עבור n=150, אנו מצפים שזמן הריצה יהיה 150^3x. היחס בין שני זמני הריצה אמור להיות $150^3x/50^3x=27$. כלומר, בעיה בת 150 לקוחות אמורה לרוץ פי 27 יותר זמן. הגרף הבא מתאר את השינוי בזמן הריצה כאשר מספר הלקוחות משתנה עבור מספר בדיקות אשר בהם מספר זוגות המסלולים המועמדים לאיחוד שונה.



תרשים 8 - הגידול בזמן הריצה ביחס למספר הלקוחות

n=50 הוא נקודת הייחוס שלנו, ושאר הזמנים הם ביחס אליו. אנו כאמור מצפים שעבור n=150, זמן הריצה יהיה גבוה פי 27, אבל בפועל, בממוצע, הוא גדול רק פי 2.5. הסבר אפשרי לכך הוא שככל שמספר הלקוחות גדל, אך הנתונים האחרים לא משתנים, מספר המסלולים שנוצרים לנו גדול יותר, מכוון שאילוצי הבעיה מונעים מאתנו לבנות מסלולים ארוכים יותר. האילוצים הללו גורמים לכך שהאלגוריתם יסיים את פעילותו מהר יותר, ומספר רב של מסלולים אפשריים, נפסלים ולא נבדקים.

4.3.2. דיוק תוצאות האלגוריתם

היבט חשוב נוסף, שיש לבחון באלגוריתם STDVRP הוא איכות התוצאות המתקבלות ע"י האלגוריתם. איכות התוצאות משמעותה היא עד כמה המסלולים המתקבלים ע"י הפעלת אלגוריתם STDVRP קרובים לתוצאה האופטימאלית.

אנו נבדוק את תוצאות האלגוריתם במספר תנאי ריצה שונים.

1. נתוני הבעיה הם דטרמיניסטיים בלבד. כאמור, פונקצית העלות, שהינה פונקצית זמן הנסיעה, מורכבת משני חלקים , סטוכסטי ודטרמיניסטי. כאשר החלק הסטוכסטי מחזיר ערך קבוע, למשל 0, עבור כל קלט שהוא מקבל, הרי שאז המודל כולו מתנהג באופן דטרמיניסטי.
2. נתוני הבעיה סטוכסטיים בלבד. הבעיה מתנהגת בצורה סטוכסטית בלבד אם החלק הקבוע של פונקצית העלות, מחזיר את אותה המהירות עבור כל פרק זמן (תלות הזמן לא מתקיימת). עבור החלק הסטוכסטי אנו נבדוק את המאפיינים הבאים :
   א. השפעת אחוז הקשתות שמתנהגות סטוכסטית.
   ב. השפעת טווח מהירות הנסיעה.
   ג. השפעת מספר ההסתברויות למהירות לכל קשת ביחידת זמן.
3. נתוני הבעיה הם סטוכסטיים ותלויי זמן. זו למעשה הבעיה המקורית איתה אלגוריתם STDVRP אמור להתמודד. מכוון ולא קיים אלגוריתם שתוכנן במיוחד עבור סוג זה של בעיות, אנו צופים התוצאות שיתקבלו יהיו טובות יותר מכל תוצאה אשר תתקבל מאלגוריתם אחר אשר ינסה להתמודד עם סוג זה של בעיות.

לרוב, כאשר מפתחים אלגוריתם חדש, משתמשים בתסריטי בדיקה קיימים, כלומר שנבנו ופתרו אותן בעבר, ומשווים את התוצאות המתקבלות מהאלגוריתם לתוצאות ידועות של שהתקבלו מאלגוריתמים קודמים שפותחו לפתרון אותה הבעיה. במקרה שלנו, מדובר בבעיה שלא נחקרה קודם לכן, ולכן לא קיימים תסריטי בדיקה שאפשר להתבסס עליהם. מכוון שבמקרה שלנו, לא קיימים תסריטי בדיקה, לצורך בדיקת האלגוריתם יבנו מספר תסריטי בדיקה באופן הבא :

1. קבוצת הבעיות הראשונה מיועדת לבדוק את תוצאות האלגוריתם, כאשר קיימת תלות בזמן בלבד. לצורך הבדיקה יבנו 4 קבוצות של בעיות, כאשר בכל קבוצה 10 בעיות. בקבוצה הראשונה יהיו 2 יחידות זמן, בקבוצה השניה 6, בשלישית 12 וברביעית 24. הבעיות לא יכילו מרכיבים סטוכסטיים.
2. קבוצת הבעיות השניה מיועדת לבדוק את תוצאות האלגוריתם, כאשר הבעיה מתנהגת באופן סטוכסטי, אך ללא תלות בזמן. את בחינת הסטוכסטיות אנו נחלק למספר תתי-בדיקות.

א. עבור נתונים אימפריים, האם למספר הנתונים בטבלת ההתפלגות ישנה השפעה על איכות תוצאת האלגוריתם. אנו נבנה את קבוצת הבעיות הסטוכסטיות, כך שב-60 בעיות יהיו 3 מהירויות בטבלת ההתפלגויות, וב-60 בעיות נוספות, יהיו 5 מהירויות. עבור התפלגות שאינה אמפירית, כגון התפלגות נורמאלית, במקרה כזה, אנו מגרילים

מספר רב של מספרים בהתאם להתפלגות הנורמאלית (או כל פונקצית התפלגות אחרת), ועל סמך הנתונים "האמפיריים" שקיבלנו, אנו בונים לנו טבלת התפלגות "נורמאלית".

ב. גורם נוסף אותו נבדוק הוא אחוז הקשתות שמתנהגות סטוכסטית. את קבוצת הבעיות הסטוכסטיות אנו נבנה כך שב-40 בעיות 25% מהקשתות הן סטוכסטיות, קבוצה נוספת של 40 בעיות יהיו עם 50% קשתות סטוכסטיות ו-40 בעיות נוספות עם 100% קשתות סטוכסטיות.

ג. הגורם השלישי שאותו אנו נבדוק, הוא פיזור הערכים המתפלגים סטוכסטית, כלומר ההשפעה של טווח מהירות הנסיעה (כאשר עבור קשת באותה האורך, מהרות נסיעה שונה משמעותה זמן נסיעה שונה) על איכות התוצאה המתקבלת. לצורך בדיקה זו אנו נבדוק קבוצת בעיה שבה מהירות הנסיעה היא בן 50 ל-120 (70 בעיות) וקבוצה של בעיות שבה מהירות הנסיעה היא בין 80 ל-120 (70 בעיות).

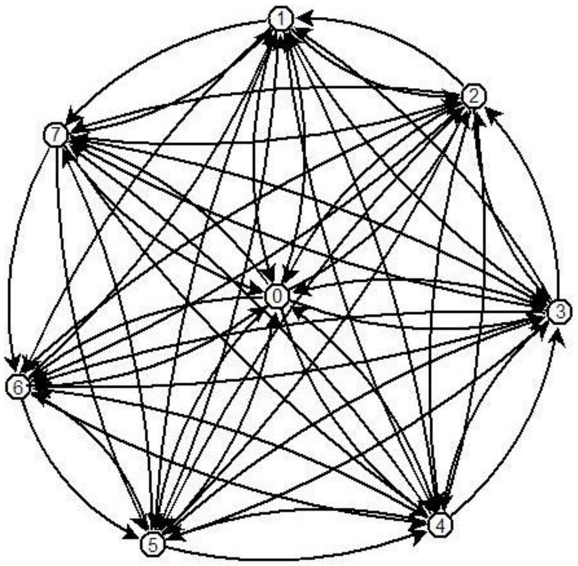
3. שילוב של סטוכסטיות ותלות בזמן. יבוצעו 20 בדיקות אשר יכילו שילוב של הבדיקות תלויות הזמן והבדיקות הסטוכסטית, וייבנו בהתאם לתוצאות הבדיקות הקודמות.

הטבלה הבאה מסכמת את תסריטי הבדיקות השונות אשר יבנו לצורך בדיקת איכות התוצאות המתקבלות מהאלגוריתם.

מס"ד	מספר הבעיות	מספר יחידות הזמן	מספר ההסתברויות למהירות לקשת ביחידת זמן	אחוז קשתות סטוכסטיות	טווח מהירויות
1	10	2	1	0	80-120
2	10	6	1	0	80-120
3	10	12	1	0	80-120
4	10	24	1	0	80-120
5	10	1	3	25%	50-120
6	10	1	3	25%	80-120
7	10	1	5	25%	50-120
8	10	1	5	25%	80-120
9	10	1	3	50%	50-120
10	10	1	3	50%	80-120
11	10	1	5	50%	50-120
12	10	1	5	50%	80-120
13	10	1	3	100%	50-120
14	10	1	3	100%	80-120
15	10	1	5	100%	50-120
16	10	1	5	100%	80-120
17	10	1	100	100%	50-120
18	10	1	100 (ה. נורמאלית)	100%	50-120
19	10	1	10 (ה. נורמאלית)	100%	50-120
20	20	תלוי בבדיקות הקודמות (24)	תלוי בבדיקות הקודמות (5)	תלוי בבדיקות הקודמות (100%)	תלוי בבדיקות הקודמות (50-120)

מכיוון שאין בידינו תסריטי בדיקה ותוצאות ממחקרים קודמים, ועל אחת כמה וכמה אין בידנו תוצאות אופטימליות אמיתיות של תסריטי הבדיקה אחרים, נאלצנו לחשב את התוצאה האופטימאלית של תסריטי הבדיקה שלנו באופן עצמאי, ע"י מעבר על כל אחד מהפתרונות האפשריים וחישוב הזמן הדרוש בכדי לעבור בו, בכל אחד מתסריטי הבדיקות, ועבור כל תסריט בדיקה, שמירת הפתרון האופטימאלי שהתקבל. תהליך זה הוא מורכב מאד ודורש זמן מחשב רב. עקב הסיבוכיות הרבה בחישוב זה, והכמות הרבה של תסריטי הבדיקה המתוכננים בעבודה זו, הוחלט שגודל הבעיה, כלומר, מספר הלקוחות בכל בעיה יהיה 7 (8 קדקודים כולל המחסן).

בשביל לעמוד על הסיבוכיות הרבה של הבעיה, התרשים הבא מתאר גרף בעל 7 לקוחות ומחסן, אשר בו קיימות קשתות בין כל אחד מהקדקודים (גרף שלם). בהנחה שכל קשת יכולה להיות שייכת לפתרון האופטימאלי, קל לראות מין התרשים כמה רב הוא מספר המסלולים הפוטנציאלים הקיימים בתרשים זה.



תרשים 9 - גרף שלם קשיר, עם מחסן ו-7 לקוחות

להלן התוצאות שהתקבלו מתסריטי הבדיקות השונים ע"י שימוש באלגוריתם.

4.3.2.1 תוצאות קבוצה מס' 1

מס' יחידות זמן : 2

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 1

אחוז הקשתות המתנהגות סטוכסטית : 0

טווח מהירויות הנסיעה : 80-120

בקבוצה זו 10 בעיות אשר מטרתן לבדוק כיצד משפיעה החלוקה לקבוצות זמן, על תוצאות אלגוריתם STDVRP מול התוצאות האופטימאליות. מכוון שאלגוריתם STDVRP מבוסס על אלגוריתם Saving, ובכדי שתהייה לנו נקודת יחוס נוספת, מוצג כאן גם הפתרון שמתקבל ע"י

אלגוריתם Savings, כאשר כל קשת תלויית זמן מתורגמת לקשת שאינה תלויית זמן, והמהירות בה היא המהירות הממוצעת של זמני הנסיעה בקשת ביחידות הזמן השונות.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	69.1	69.5	68.5	0.9	1.5
2	69.4	69.7	60.4	14.9	15.4
3	68.0	64.6	60.3	12.8	7.1
4	64.8	60.6	53.9	20.2	12.4
5	59.5	59.5	58.3	2.1	2.1
6	67.0	65.2	62.3	7.5	4.7
7	66.1	66.1	59.5	11.1	11.1
8	64.2	63.8	62.7	2.4	1.8
9	68.8	66.8	66.8	3.0	0
10	67.1	67.2	65.5	2.4	2.6

תוצאת אלגוריתם Saving, תוצאת אלגוריתם STDVRP והתוצאה האופטימאלית, כפי שהן מובאות הן הזמן הכולל הנדרש לנוע בכל המסלולים של הפתרון. אחוז הסטייה מחושב כתוצאת האלגוריתם חלקי התוצאה האופטימאלית, פחות אחת, כפול 100.

כפי שניתן לראות אחוז הסטייה של אלגוריתם STDVRP מהתוצאה האופטימאלית הוא בין 0% ל-15.4%, כשהממוצע הוא 5.9 וסטיית התקן היא 5.1. בעוד, שאם היינו משתמשים באלגוריתם Savings היינו מקבלים תוצאות שהסטייה שלהם מהתוצאה האופטימאלית הוא בין 0% ל-20.2%, כשהממוצע הוא 7.7 וסטיית התקן היא 6.3.

ההבדל בין תוצאת האלגוריתם לתוצאת אלגוריתם Saving אינו גדול, וזאת כנראה מכוון שבבעיה זו ישנן שתי יחידות זמן בלבד. בקבוצות הבאות, מספר יחידות הזמן גדול יותר, כך שנוכל לראות אם אכן למספר יחידות הזמן ישנה השפעה על תוצאות האלגוריתם, בין ביחס לפתרון האופטימאלי והן ביחס לאלגוריתם Saving.

4.3.2.2 תוצאות קבוצה מס' 2

מס' יחידות זמן : 6

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 1

אחוז הקשתות המתנהגות סטוכסטית : 0

טווח מהירויות הנסיעה : 80-120

גם בקבוצה זו 10 בעיות אשר מטרתן לבדוק כיצד משפיעה החלוקה לקבוצות זמן, על תוצאות אלגוריתם STDVRP מול התוצאות האופטימאליות. קבוצת בעיות זו היא המשך לקבוצת הבעיות הקודמת, ודומה לה, כאשר ההבדל היחיד הוא במספר יחידות הזמן, אשר היה 2 בקבוצה הקודמת, והיינו 6 בקבוצת הבעיות הנוכחית.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	66.4	66.4	66.4	0	0
2	73.3	73.3	63.6	15.3	15.3
3	61.7	61.7	58.5	5.5	5.5
4	67.8	67.6	59.5	13.9	13.6
5	59.0	57.1	57.1	3.3	0
6	63.2	63.2	63.2	0	0
7	67.7	67.7	67.5	0.3	0.3
8	75.5	72.1	72.1	4.7	0
9	67.9	67.7	62.8	8.1	7.8
10	60.1	60.1	60.1	0	0

מהתוצאות המופיעות בטבלה ניתן לראות, שקיימים שלושה מקרים שבהם הן תוצאת אלגוריתם Savings והן תוצאת אלגוריתם STDVRP שוות לתוצאה האופטימאלית. כמו כן, ישנו מקרה אחד שבו תוצאת אלגוריתם Savings זהה לתוצאת אלגוריתם STDVRP, והן גדולות מהתוצאה האופטימאלית. אחוזי הסטייה, הן של אלגוריתם STDVRP והן של אלגוריתם Saving, הם בין 0% ל-15.3%, כשהממוצע עבור אלגוריתם STDVRP הוא 4.3 וסטיית התקן היא 5.7. עבור אלגוריתם Savings הממוצע הוא 5.1 וסטיית התקן היא 5.4.

ההבדל בין תוצאת האלגוריתם לתוצאת אלגוריתם Saving עדיין אינו גדול, איך תוצאות המתקבלות מאלגוריתם בכל המקרים הנן טובות יותר מהתוצאות שהתקבלו מה-Savings.

4.3.2.3 תוצאות קבוצה מס' 3

מס' יחידות זמן : 12

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 1

אחוז הקשתות המתנהגות סטוכסטית : 0

טווח מהירויות הנסיעה : 80-120

אנו ממשיכים עם קבוצה נוספת בת 10 בעיות, כאשר הפעם מספר יחידות הזמן עבור כל קשת הוא 12.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	70.3	70.2	70.2	0.1	0
2	66.7	65.9	58.3	14.4	13.0
3	62.3	62.3	59.8	4.2	4.2
4	70.0	70.0	57.9	20.9	20.9
5	59.7	60.1	58.7	1.7	2.4
6	77.2	66.2	64.9	19.0	2.0
7	58.8	61.7	57.5	2.3	7.3
8	62.3	63.0	62.3	0	1.1

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
9	87.2	85.4	69.0	26.4	23.8
10	62.0	60.8	57.6	7.6	5.6

ניתן לראות בתוצאות המופיעות בטבלה, שהתוצאות המתקבלות מאלגוריתם STDVRP הן ברוב המקרים טובות יותר, או שוות לתוצאות שהתקבלו מאלגוריתם Savings, פרט למקרה אחד שבו אלגוריתם Savings הגיע לתוצאה האופטימאלית, ואילו האלגוריתם הגיעה לתוצאה שרחוקה ב-1.1% מהתוצאה האופטימאלית.

אחוזי הסטייה של אלגוריתם STDVRP הם בן 0% ל-23.8%, כשהממוצע עבור האלגוריתם הוא 7.8 וסטיית התקן היא 8.0. אחוזי הסטייה של אלגוריתם Savings הם בין 0% ל-26.4%, כשהממוצע הוא 9.7 וסטיית התקן היא 9.2.

ההבדל הקטן בין תוצאת האלגוריתם לתוצאת אלגוריתם Saving עדיין נשמר, וכן, לרוב תוצאות המתקבלות מאלגוריתם טובות יותר מהתוצאות שהתקבלו מה-Savings.

4.3.2.4 תוצאות קבוצה מס' 4

מס' יחידות זמן : 24

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 1

אחוז הקשתות המתנהגות סטוכסטית : 0

טווח מהירויות הנסיעה : 80-120

קבוצת הבאה בת 10 בעיות, היא קבוצת הבעיות האחרונה אשר מיועדת לבדוק את התמודדות אלגוריתם STDVRP עם בעיות תלויות זמן (ואשר אינן סטוכסטיות), כשהפעם מספר יחידות הזמן עבור כל קשת הוא 24.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	64.7	65.9	64.7	0	1.9
2	70.7	67.8	59.8	18.2	13.4
3	61.3	61.3	61.0	0.5	0.5
4	70.5	70.5	59.4	18.7	18.7
5	60.9	61.8	58.8	3.6	5.1
6	80.8	79.8	67.4	19.9	18.4
7	66.5	73.9	66.0	0.8	12.0
8	75.1	75.1	66.1	13.6	13.6
9	79.9	75.6	64.4	24.1	17.4
10	64.7	64.7	57.2	13.1	13.1

כפי שניתן לראות, המגמה הכללית נמשכת, התוצאות שמתקבלות מאלגוריתם STDVRP הן לרוב טובות יותר מהתוצאות המתקבלות מאלגוריתם Savings. אחוזי הסטייה של אלגוריתם STDVRP הם בין 0.5% ל-18.7%, כשהממוצע עבור האלגוריתם הוא 11.4 וסטיית התקן היא 6.3. אחוזי הסטייה של אלגוריתם Savings הם בין 0% ל-21.1%, כשהממוצע הוא 11.3, וסטיית התקן היא 8.7.

מהתוצאות המופיעות בסעיפים 4.3.2.1, 4.3.2.2, 4.3.2.3 ו-4.3.2.4 ניתן להסיק, שעבור בעיות תלויות זמן, אלגוריתם STDVRP נותן תוצאות טובות יותר, מאשר אם היינו משתמשים באלגוריתם Savings הרגיל. גם אחוזי הסטייה של אלגוריתם STDVRP מין התוצאה האופטימאלית הם אינם גדולים מאחוזי הסטייה של אלגוריתם Savings המקורי עבור בעיית CVRP. כמו כן אפשר לראות שמספר יחידות הזמן אשר קיימות לכל קשת אינו מהווה גורם שמשפיע משמעותית על איכות התוצאות, כלומר, לא נראתה ירידה משמעותית באיכות התוצאות ככל שמספר יחידות הזמן גדל.

4.3.2.5 תוצאות קבוצה מס' 5

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 3

אחוז הקשתות המתנהגות סטוכסטית : 25%

טווח מהירויות הנסיעה : 50-120

קבוצות הבעיות הבאות מתוכננות לבדוק את השפעת המרכיב הסטוכסטי של הבעיה על תוצאות אלגוריתם STDVRP. שלושה מרכיבים נבדקים בחלק הסטוכסטי. המרכיב הראשון הוא מספר ההסתברויות למהירות בכל קשת ליחידת זמן. המרכיב השני הוא אחוז הקשתות המתנהגות סטוכסטית, והמרכיב השלישי הוא טווח המהירויות. בקבוצה הבעיות הנוכחית, יש 10 בעיות אשר אחוז הקשתות בהן המתנהגות סטוכסטית הוא 25%, טווח המהירויות הוא 50-120, ומספר ההסתברויות למהירות לכל קשת ביחידת זמן הוא 3.

גם בקבוצת בעיות אלו, אנו נשווה את תוצאת אלגוריתם STDVRP מול התוצאות האופטימאליות, וכן מול הפתרון שמתקבל ע"י אלגוריתם Savings, כאשר כל קשת סטוכסטית מתורגמת לקשת שאינה סטוכסטית, והמהירות בה היא המהירות הממוצעת המשוקלל של זמני הנסיעה בקשת.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	86.5	79.3	79.3	9.1	0
2	78.6	78.1	71.5	9.9	9.2
3	80.7	74.7	74.7	8.0	0
4	73.0	72.4	66.6	9.6	8.7
5	73.1	69.0	69.0	5.9	0

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
6	88.2	75.9	75.3	17.1	0.8
7	95.6	79.3	79.3	20.6	0
8	75.6	73.6	72.3	4.6	1.8
9	79.7	71.2	68.3	16.7	4.2
10	66.2	64.1	63.0	5.1	1.7

מהנתונים שהתקבלו מהרצת הבדיקות ניתן לראות שאחוז הסטייה של אלגוריתם STDVRP מהתוצאה האופטימאלית הוא בן 0% ל-9.2%, כשהממוצע הוא 2.6 וסטיית התקן היא 3.4. בעוד, שעבור אלגוריתם Savings סטיית התוצאות מהתוצאות האופטימאליות היא בין 4.6% ל-20.6, כשהממוצע הוא 10.7 וסטיית התקן היא 5.3.

בעוד שעבור בעיות תלויות זמן, לא נמצא הבדל גדול בין הפתרון שהתקבל מאלגוריתם Savings לבין הפתרון שהתקבל מאלגוריתם STDVRP, עבור בעיות סטוכסטיות ההבדל הוא גדול יחסית, והן טובות פי 4 לערך מהתוצאות שהתקבלו מאלגוריתם Savings.

4.3.2.6 תוצאות קבוצה מס' 6

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 3

אחוז הקשתות המתנהגות סטוכסטית : 25%

טווח מהירויות הנסיעה : 80-120

קבוצת בעיות הבאה מיועדת גם היא לבדוק את התמודדות אלגוריתם STDVRP עם הבעיה הסטוכסטית. 10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הקודמת, פרט לטווח המהירויות שבקבוצת בעיות אלו הוא קטן יותר ועומד על 80-120.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	66.9	65.1	65.1	2.8	0
2	74.8	65.8	63.9	17.1	3.0
3	67.1	66.3	62.2	7.9	6.6
4	71.4	63.9	61.9	15.3	3.2
5	61.5	60.0	60.0	2.5	0
6	60.3	56.8	56.8	6.2	0
7	80.9	78.6	74.2	9.0	5.9
8	52.1	52.9	52.9	-1.5	0
9	59.1	54.5	54.5	8.4	0
10	66.7	63.8	63.8	4.5	0

מהנתונים ניתן לראות שסטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעים בין (-1.5)% לבין 17.1%, כאשר הממוצע הוא 7.2 וסטיית התקן היא 5.4. עבור אחת הבדיקות אלגוריתם ה-Savings נתן תוצאה יותר טובה מהתוצאה האופטימאלית, יש ליחס זאת לסטוכסטיות של הבעיה, שגורמת למצב שעבור מסלולים זהים, אנו יכולים לקבל כל פעם תוצאות שונות. סטיות התוצאות שהתקבלו מאלגוריתם STDVRP נעות בין 0% ל-6.6%, כשממוצע הוא 1.9 וסטיית התקן היא 2.5.

גם עבור קבוצת בדיקות זאת, התוצאות שהתקבלו מאלגוריתם STDVRP טובות יותר משמעותי מהתוצאות שהתקבלו מאלגוריתם Savings. בנוסף, התוצאות אף טובות יותר מהתוצאות שהתקבלו בקבוצת הבדיקות הקודמת, יתכן שבגלל שטווח המהירויות קטן יותר. ישנן עוד קבוצות של בדיקות אשר נעשת בהן השוואה של השפעת טווח המהירות, ובמידה ואכן הוא משפיע על איכות התוצאות של האלגוריתם אנו נראה את זה גם בהמשך.

4.3.2.7 תוצאות קבוצה מס' 7

מס' יחידות זמן : 1
התפלגות : אמפירית
מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 5
אחוז הקשתות המתנהגות סטוכסטית : 25%
טווח מהירויות הנסיעה : 50-120

גם קבוצת בעיות הבאה מיועדת לבדוק את התמודדות אלגוריתם STDVRP עם הבעיה הסטוכסטית. 10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הבעיות שהוצגה בסעיף 4.3.2.5, כאשר ההבדל הוא במספר ההסתברויות למהירות קשת ביחידת זמן, שבקבוצה זו הוא 5 במקום 3.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	95.4	91.9	86.9	9.8	5.8
2	99.7	78.9	74.1	34.5	6.5
3	75.1	67.0	67.0	12.1	0
4	90.9	80.2	69.4	31.0	15.6
5	74.9	73.3	69.9	7.2	4.9
6	77.2	67.4	59.6	29.5	13.1
7	70.7	66.3	62.0	14.0	6.9
8	88.9	63.7	63.4	40.2	0.5
9	95.2	83.8	83.8	13.6	0
10	80.2	68.4	64.3	24.7	6.4

כפי שניתן לראות הטבלה, גם במקרה זה, התוצאות המתקבלות מאלגוריתם STDVRP הן טובות יותר, מתוצאות שהתקבלו מאלגוריתם Savings. אחוזי הסטייה של אלגוריתם STDVRP הם בן 0% ל-15.6%, כשהממוצע עבור אלגוריתם STDVRP הוא 6.0 וסטיית התקן היא 5.0.

אחוזי הסטייה של אלגוריתם Savings הם בין 7.2% ל-40.2%, כשהממוצע הוא 21.7 וסטיית התקן היא 11.1.

ההבדל בין תוצאת אלגוריתם STDVRP לתוצאת אלגוריתם ה-Saving הוא משמעותי, ואלגוריתם STDVRP מראה יתרון ברור. בנוסף, נראה שהעלאת מספר ההסתברויות למהירות בכל קשת מגדילה את אחוז הסטייה של אלגוריתם STDVRP מהתוצאה האופטימאלית, אך עדיין הוא נמוך ועומד על 6% בממוצע, כאשר בשימוש ב-Savings רגיל הסטייה הממוצע גדולה פי 3 לערך.

4.3.2.8 תוצאות קבוצה מס' 8

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 5

אחוז הקשתות המתנהגות סטוכסטית : 25%

טווח מהירויות הנסיעה : 80-120

מטרת קבוצת הבעיות הבאה היא לבדוק את התמודדות אלגוריתם STDVRP עם הבעיה הסטוכסטית. 10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הקודמת, פרט לטווח המהירויות שבקבוצת בעיות אלו הוא קטן יותר ועומד על 80-120.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	66.8	65.5	65.5	2.0	0
2	68.1	62.1	60.7	12.2	2.3
3	61.6	61.6	59.4	3.7	3.7
4	65.8	58.7	53.9	22.1	8.9
5	61.6	58.9	58.6	5.1	0.5
6	72.3	67.1	65.8	9.9	2.0
7	63.7	59.3	56.0	13.8	5.9
8	63.8	62.3	62.2	2.6	0.2
9	63.3	63.7	61.6	2.8	3.4
10	61.5	61.5	58.6	4.9	4.9

מהנתונים ניתן לראות שסטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעות בין 2.0% לבין 22.1%, כשהממוצע הוא 7.9 וסטיית התקן היא 6.2. סטיות התוצאות שהתקבלו מאלגוריתם STDVRP נעות בין 0% ל-8.9%, כשממוצע הוא 3.2 וסטיית התקן היא 2.7.

התוצאות שהתקבלו מאלגוריתם STDVRP טובות יותר משמעותי מהתוצאות שהתקבלו מאלגוריתם Savings, וכן הן טובות יותר מהתוצאות שהתקבלו בקבוצת הבדיקות הקודמת. נראה שלהקטנת טווח המהירות יש אכן השפעה על דיוק אלגוריתם STDVRP, וככל שהטווח קטן יותר, התוצאות קרובות יותר לתוצאה האופטימאלית. מגמה זו מתבטאת גם באלגוריתם

STDVRP וגם באלגוריתם ה-Savings. כאמור,בהמשך ישנן עוד קבוצות של בדיקות אשר נעשת בהן השוואה של השפעת טווח המהירות, ובמידה ואכן הוא משפיע על איכות התוצאות של אלגוריתם STDVRP אנו נראה את זה גם בהמשך.

4.3.2.9 תוצאות קבוצה מס' 9

מס' יחידות זמן : 3

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 3

אחוז הקשתות המתנהגות סטוכסטית : 50%

טווח מהירויות הנסיעה : 50-120

קבוצת בעיות הבאה מיועדת לבדוק את התמודדות אלגוריתם STDVRP עם הבעיה הסטוכסטית. 10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הבעיות שהוצגה בסעיף 4.3.2.5, כאשר ההבדל הוא במספר הקשתות המתנהגות סטוכסטית ואשר עומד על 50% (במקום 25%).

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	89.4	84.0	69.5	28.6	20.9
2	85.9	83.0	63.2	35.9	31.3
3	76.8	65.1	60.1	27.8	8.3
4	86.0	69.8	69.8	23.2	0
5	82.3	70.3	67.5	21.9	4.1
6	90.0	76.4	75.8	18.7	0.8
7	66.8	58.9	56.8	17.6	3.7
8	80.4	73.2	72.1	11.5	1.5
9	74.2	72.3	70.0	6.0	3.3
10	79.3	68.6	68.6	15.6	0

מהנתונים שהתקבלו מהרצת הבדיקות ניתן לראות שאחוז הסטייה של אלגוריתם STDVRP מהתוצאה האופטימאלית הוא בין 0% ל-31.3%, כשהממוצע הוא 7.4 וסטיית התקן היא 9.9. בעוד, שעבור אלגוריתם Savings סטיית התוצאות מהתוצאות האופטימאליות היא בין 4.6% ל-35.9, כשהממוצע הוא 10.7 וסטיית התקן היא 5.3.

בהשוואה לתוצאות שהתקבלו בסעיף 4.3.2.5, נראה הגדלה של אחוז הקשתות המתנהגות סטוכסטית מפחית את אחוז הדיוק של אלגוריתם STDVRP. התוצאות שהתקבלו הן עדיין טובות יותר מאשר התוצאות שהתקבלו מאלגוריתם ה-Saving. כמו כן, נראה שתוצאות אלגוריתם ה-Saving לא הושפעו משינוי אחוז הקשתות שמתנהגות סטוכסטית.

4.3.2.10 תוצאות קבוצה מס' 10

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 3

אחוז הקשתות המתנהגות סטוכסטית : 50%

טווח מהירויות הנסיעה : 80-120

קבוצת בעיות הבאה מכילה 10 הבעיות, זהות לבעיות שהיו בקבוצה הקודמת, פרט לטווח המהירויות שבקבוצת בעיות אלו שהוא קטן יותר ועומד על 80-120.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	65.9	65.5	65.4	0.8	0.2
2	71.7	66.3	62.0	15.6	6.9
3	67.3	63.3	60.9	10.5	3.9
4	64.8	62.9	54.3	19.3	15.8
5	65.0	64.2	61.0	6.6	5.2
6	58.6	56.6	55.1	6.4	2.7
7	67.3	67.3	66.6	1.1	1.1
8	68.7	65.4	65.1	5.5	0.5
9	81.9	72.2	72.2	13.4	0
10	72.3	69.7	65.6	10.2	6.3

מהנתונים ניתן לראות שסטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעות בין 0.8% לבין 19.3%, כאשר הממוצע הוא 8.9 וסטיית התקן היא 5.7. סטיות התוצאות שהתקבלו מאלגוריתם STDVRP נעות בין 0% ל-15.8%, כשממוצע הוא 4.3 וסטיית התקן היא 4.5.

התוצאות שהתקבלו מאלגוריתם STDVRP טובות יותר משמעותי מהתוצאות שהתקבלו מאלגוריתם Savings. בנוסף, התוצאות אף טובות יותר מהתוצאות שהתקבלו בקבוצת הבדיקות הקודמת, מה שמחזק את הטענה שטווח המהירויות קטן יותר מאפשר לקבל תוצאות טובות יותר. כמו כן, גם קבוצת בדיקות אלו תומכת בטענה שככל שאחוז הקשתות המתנהגות סטוכסטית גבוה יותר תוצאות אלגוריתם STDVRP יהיו פחות טובות.

4.3.2.11 תוצאות קבוצה מס' 11

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 5

אחוז הקשתות המתנהגות סטוכסטית : 50%

טווח מהירויות הנסיעה : 50-120

גם קבוצת בעיות הבאה מיועדת לבדוק את התמודדות אלגוריתם STDVRP עם הבעיה הסטוכסטית. 10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הבעיות שהוצגה בסעיף 4.3.2.9, כאשר ההבדל הוא במספר ההסתברויות למהירות קשת ביחידת זמן, שבקבוצה זו הוא 5 במקום 3.

מס״ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	84.0	80.9	79.5	5.7	1.8
2	87.7	76.9	70.1	25.1	9.7
3	90.0	71.6	70.6	27.5	1.4
4	81.1	75.9	70.9	14.4	7.1
5	73.0	69.2	67.2	8.6	3.0
6	89.4	80.9	78.8	13.5	2.7
7	66.6	55.8	54.2	22.9	3.0
8	112.2	92.2	92.2	21.7	0
9	104.4	89.1	84.8	23.1	5.1
10	90.7	83.0	71.9	26.1	15.4

כפי שניתן לראות הטבלה, גם במקרה זה, התוצאות המתקבלות מאלגוריתם STDVRP הן טובות יותר, מתוצאות שהתקבלו מאלגוריתם Savings. אחוזי הסטייה של האלגוריתם הם בן 0% ל-15.4%, כשהממוצע עבור האלגוריתם הוא 4.9 וסטיית התקן היא 4.4. אחוזי הסטייה של אלגוריתם Savings הם בין 5.7% ל-27.5%, כשהממוצע הוא 18.9 וסטיית התקן היא 7.3.

יתרון אלגוריתם STDVRP על אלגוריתם Saving הוא ברור. בנוסף, למרות שצפוי שהעלאת מספר ההסתברויות למהירות בכל קשת תגדיל את אחוז הסטייה של אלגוריתם STDVRP מהתוצאה האופטימאלית, המצב הוא הפוך לגמרי.

4.3.2.12 תוצאות קבוצה מס׳ 12

מס׳ יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 5

אחוז הקשתות המתנהגות סטוכסטית : 50%

טווח מהירויות הנסיעה : 80-120

10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הקודמת, פרט לטווח המהירויות שבקבוצת בעיות אלו הוא קטן יותר ועומד על 80-120.

מס״ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	73.7	66.9	66.9	10.2	0
2	68.9	67.0	58.1	18.6	15.3
3	66.3	62.9	60.8	9.0	3.5

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
4	64.1	60.5	57.8	10.9	4.7
5	71.8	63.3	62.8	14.3	0.8
6	60.8	56.5	55.2	10.1	2.4
7	58.6	58.6	58.5	0.2	0.2
8	72.5	65.5	65.5	10.7	0
9	60.0	58.0	58.0	3.4	0
10	71.1	71.1	67.2	5.8	5.8

מהנתונים ניתן לראות שסטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעות בין 0.2% לבין 18.6%, כשהממוצע הוא 9.3 וסטיית התקן היא 5.0. סטיות התוצאות שהתקבלו מאלגוריתם STDVRP נעות בין 0% ל-15.3%, כשממוצע הוא 3.3 וסטיית התקן היא 4.5.

התוצאות שהתקבלו מאלגוריתם STDVRP טובות יותר משמעותי מהתוצאות שהתקבלו מאלגוריתם Savings, וכן הן טובות מעט יותר מהתוצאות שהתקבלו בקבוצת הבדיקות הקודמת. נראה שלהקטנת טווח המהירות יש אכן השפעה על דיוק אלגוריתם STDVRP, וככל שהטווח קטן יותר, התוצאות קרובות יותר לתוצאה האופטימאלית. מגמה זו מתבטאת גם באלגוריתם STDVRP וגם באלגוריתם ה-Savings. בנוסף נראה שהגדלת אחוז הקשתות המתנהגות סטוכסטית גם משפיע על תוצאת אלגוריתם STDVRP, אולם ההשפעה היא קטנה יותר.

4.3.2.13 תוצאות קבוצה מס' 13

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 3

אחוז הקשתות המתנהגות סטוכסטית : 100%

טווח מהירויות הנסיעה : 50-120

קבוצת בעיות הבאה מיועדת לבדוק את התמודדות אלגוריתם STDVRP עם הבעיה הסטוכסטית. 10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הבעיות שהוצגה בסעיף 4.3.2.5 ובסעיף 4.3.2.9, כאשר ההבדל הוא במספר הקשתות המתנהגות סטוכסטית ואשר עומד על 100% (במקום 25% ו-50%).

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	80.2	73.6	73.1	9.7	0.7
2	73.0	71.9	65.8	10.9	9.3
3	72.2	66.6	66.6	8.4	0
4	85.1	83.0	78.0	9.1	6.4
5	76.8	70.3	69.1	11.1	1.7
6	86.9	78.7	76.1	14.2	3.4
7	89.4	82.3	74.3	20.3	10.8

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
8	112.5	119.6	101.7	10.6	17.6
9	89.6	70.5	66.5	34.7	6.0
10	77.2	71.9	66.5	16.1	8.1

מהנתונים שהתקבלו מהרצת הבדיקות ניתן לראות שאחוז הסטייה של אלגוריתם STDVRP מהתוצאה האופטימאלית הוא בין 0% ל-17.6%, כשהממוצע הוא 6.4 וסטיית התקן היא 5.1. בעוד, שעבור אלגוריתם Savings סטיית התוצאות מהתוצאות האופטימאליות היא בין 8.4% ל-34.7%, כשהממוצע הוא 14.5 וסטיית התקן היא 7.6.

בהשוואה לתוצאות שהתקבלו בסעיף 4.3.2.5, נראה הגדלה של אחוז הקשתות המתנהגות סטוכסטית מפחית את אחוז הדיוק של אלגוריתם STDVRP. אולם נתון זה לא מתיישב עם התוצאות שהתקבלו בסעיף 4.3.2.9. בהשוואה לאלגוריתם ה-Saving, נראה ככל שמספר הקשתות המתנהגות סטוכסטית גדל, תוצאת אלגוריתם ה-Saving תהיה רחוקה יותר מהתוצאה האופטימאלי.

4.3.2.14 תוצאות קבוצה מס' 14

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 3

אחוז הקשתות המתנהגות סטוכסטית : 100%

טווח מהירויות הנסיעה : 80-120

10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הקודמת, פרט לטווח המהירויות שבקבוצת בעיות אלו הוא קטן יותר ועומד על 80-120.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	64.8	64.8	64.8	0	0
2	65.1	64.2	60.6	7.4	5.9
3	60.0	59.2	58.3	2.9	1.5
4	61.9	61.9	61.9	0	0
5	63.8	59.8	59.8	6.7	0
6	69.4	62.1	60.9	14.0	2.0
7	65.5	65.2	58.3	12.3	11.8
8	57.6	59.8	57.3	0.5	4.4
9	76.7	76.1	66.6	15.2	14.3
10	67.4	62.5	57.0	18.2	9.6

מהנתונים ניתן לראות שסטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעות בין 0% לבין 18.2%, כאשר הממוצע הוא 7.7 וסטיית התקן היא 6.5. סטיות התוצאות שהתקבלו מאלגוריתם STDVRP נעות בין 0% ל-14.3%, כשממוצע הוא 5 וסטיית התקן היא 5.

התוצאות שהתקבלו מאלגוריתם STDVRP טובות יותר מהתוצאות שהתקבלו מאלגוריתם Savings. כמו כן, התוצאות אף טובות יותר מהתוצאות שהתקבלו בקבוצת הבדיקות הקודמת, כך שהמגמה שטווח מהירות קטן יותר מאפשר לאלגוריתם STDVRP לספק תוצאות קרובות יותר לתוצאה האופטימאלית נמשכת.

4.3.2.15 תוצאות קבוצה מס' 15

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 5

אחוז הקשתות המתנהגות סטוכסטית : 100%

טווח מהירויות הנסיעה : 50-120

10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הבעיות שהוצגה בסעיף 4.3.2.13, כאשר ההבדל הוא במספר ההסתברויות למהירות קשת ביחידת זמן, שבקבוצה זו הוא 5 במקום 3.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	84.9	78.8	78.8	7.7	0
2	89.7	83.2	74.2	20.9	12.1
3	100.9	70.9	65.0	55.2	9.1
4	93.0	73.1	65.4	42.2	11.8
5	91.2	70.3	70.3	29.7	0
6	76.9	67.3	66.5	15.6	1.2
7	65.8	62.8	62.1	6.0	1.1
8	67.8	64.6	62.2	9.0	3.9
9	65.6	53.8	53.8	21.9	0
10	81.5	77.9	74.9	8.8	4.0

כפי שניתן לראות הטבלה, גם במקרה זה, התוצאות המתקבלות מאלגוריתם STDVRP הן טובות יותר, מתוצאות שהתקבלו מאלגוריתם Savings. אחוזי הסטייה של אלגוריתם STDVRP הם בן 0% ל-12.1%, כשהממוצע עבור האלגוריתם הוא 4.3 וסטיית התקן היא 4.6. אחוזי הסטייה של אלגוריתם Savings הם בין 6.0% ל-55.2%, כשהממוצע הוא 21.7 וסטיית התקן היא 15.5.

ההבדל בין תוצאת אלגוריתם STDVRP לתוצאת אלגוריתם ה-Saving הוא משמעותי, והאלגוריתם מראה יתרון ברור. בנוסף, נראה שהעלאת מספר ההסתברויות למהירות בכל קשת

אכן מגדילה את אחוז הסטייה הממוצעת מהפתרון האופטימאלי, אך במקביל, העלאת אחוז הקשתות המתנהגות סטוכסטית עוזרת להקטין את אחוז הסטייה.

4.3.2.16 תוצאות קבוצה מס' 16

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 5

אחוז הקשתות המתנהגות סטוכסטית : 100%

טווח מהירויות הנסיעה : 80-120

10 הבעיות הנמצאות בקבוצה זו זהות לבעיות שהיו בקבוצה הקודמת, פרט לטווח המהירויות שבקבוצת בעיות אלו הוא קטן יותר ועומד על 80-120.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	72.8	65.2	65.2	11.7	0
2	72.5	69.5	63.2	14.7	10.0
3	66.3	63.4	58.9	12.6	7.6
4	61.5	60.7	57.3	7.3	5.9
5	69.3	62.0	61.2	13.2	1.3
6	58.7	58.0	56.9	3.2	1.9
7	65.1	63.4	63.4	2.7	0
8	56.0	52.8	52.0	7.7	1.5
9	53.7	53.7	51.9	3.5	3.5
10	52.2	50.7	50.7	3.0	0

מהנתונים ניתן לראות שסטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעות בין 2.7% לבין 14.7%, כשהממוצע הוא 8.0 וסטיית התקן היא 4.5. סטיות התוצאות שהתקבלו מאלגוריתם STDVRP נעות בין 0% ל-10.0%, כשממוצע הוא 3.2 וסטיית התקן היא 3.3.

המגמה בה תוצאות אלגוריתם STDVRP טובות יותר מהתוצאות אלגוריתם Savings נשמרת. כמו כן הן טובות מעט יותר מהתוצאות שהתקבלו בקבוצת הבדיקות הקודמת. לאורך כל הבדיקות, כאשר טווח המהירויות היה קטן יותר, תוצאות אלגוריתם STDVRP היו טובות יותר. בקבוצת בדיקות זו, לא נראה שהגדלת אחוז הקשתות המתנהגות סטוכסטית הייתה השפעה כל שהיא על תוצאות אלגוריתם STDVRP.

4.3.2.17 תוצאות קבוצה מס' 17

מס' יחידות זמן : 1

התפלגות : אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 100

אחוז הקשתות המתנהגות סטוכסטית : 100%

טווח מהירויות הנסיעה : 50-120

בקבוצות הבעיות הקודמות אשר בדקו אספקטים שונים בחלק הסטוכסטי של הבעיה, נמצא שגורמים שונים יכולים להשפיע על תוצאות אלגוריתם STDVRP, אך לא נמצאו הפרשים גדולים בתוצאות האלגוריתם STDVRP. 10 הבעיות הנמצאות בקבוצה זו מטרתן לבדוק אם להגדלה קיצונית במספר ההסתברויות למהירות לכל קשת ביחידת זמן יש השפעה על תוצאות אלגוריתם STDVRP. מכוון שלא נמצאו הפרשים גדולים בין תוצאות אלגוריתם STDVRP בקבוצות הבעיות הקודמות, הוחלט לקחת את המקרים ה"קשים" יותר ולשלבם בקבוצת הבעיות הנוכחית, כלומר, אחוז הקשתות המתנהגות סטוכסטית הוא 100%, טווח המהירויות הוא טווח רחב, 50-120, ומספר ההסתברויות למהירות לכל קשת ביחידת זמן הוגדל באופן קיצוני ל-100.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	91.6	91.0	76.2	20.2	19.4
2	85.6	85.6	78.6	8.9	8.9
3	81.9	78.3	73.5	11.4	6.5
4	72.4	72.2	63.4	14.2	13.9
5	69.6	66.5	57.7	20.6	15.3
6	99.8	99.8	84.2	18.5	18.5
7	68.3	66.6	62.2	9.8	7.1
8	68.3	68.3	60.5	12.9	12.9
9	80.9	75.3	69.0	17.2	9.1
10	67.3	63.4	62.3	8.0	1.8

מהנתונים ניתן לראות שסטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעים בין 8.0% לבין 20.6%, כאשר הממוצע הוא 14.2 וסטיית התקן היא 4.5. סטיות התוצאות שהתקבלו מאלגוריתם STDVRP נעות בין 1.8% ל-19.4%, כשממוצע הוא 11.3 וסטיית התקן היא 5.3.

התוצאות שהתקבלו מאלגוריתם STDVRP טובות יותר מהתוצאות שהתקבלו מאלגוריתם Savings. אולם, בהתייחס לתוצאות קודמות, התוצאות פחות טובות. בממוצע, אחוז הסטייה מהתוצאה האופטימאלית שהמתקבל מאלגוריתם STDVRP גדול פי 2 מהתוצאות שהתקבלות בבדיקות הקודמות. הגידול בסטייה נובע כנראה ממספר ההסתברויות למהירות לכל קשת ביחידת זמן, אשר גדול בבעיה זו באופן קיצוני (100) לעומת שאר הבעיות (3 ו-5).

4.3.2.18 תוצאות קבוצה מס' 18

מס' יחידות זמן : 1

התפלגות : נורמאלית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 100

אחוז הקשתות המתנהגות סטוכסטית : 100%

טווח מהירויות הנסיעה : 50-120

בקבוצות הבעיות הקודמות ההתפלגות שבה השתמשנו הייתה התפלגות אמפירית. בכדי לקבל טבלת התפלגות אמפירית אנו אוספים תוצאות מהשטח. את ההפרש בין התוצאה הגבוהה ביותר והתוצאה הנמוכה ביותר אנו מחלקים למספר חלקים קבוע (במקרה שלנו 3 ו-5). עבור כל חלק, אנו סופרים כמה מהתוצאות שקיבלנו נמצאות בו. ע״י חלוקה של מספר התוצאות שנמצאות בכל חלק וחלק, מספר התוצאות הכולל, ניתן לדעת מהי ההסתברות לקבלת ערך (מהירות) שהוא בטווח של כל חלק וחלק.

אנו מעוניינים לבדוק כיצד מתנהג אלגוריתם STDVRP עבור התפלגויות שאינן אמפיריות, במקרה שלנו, התפלגות נורמאלית. בכדי לא לשנות את אופן מימוש האלגוריתם, אנו נסתכל על ההתפלגות הנורמאלית כאמפירית ע״י הגרלת תוצאות רבות ע״י פונקציה אשר מחזירה מספרים אקראיים בהתאם להתפלגות הנורמאלית, אשר ישמשו תוצאות אמפיריות.

ב-10 הבעיות שבקבוצה זו, אחוז הקשתות המתנהגות סטוכסטית הוא 100%, טווח המהירויות הוא טווח רחב, 50-120, ובכדי שהנתונים ידמו עד כמה שניתן התפלגות נורמאלית, מספר ההסתברויות למהירות לכל קשת ביחידת זמן הוא 100.

מס״ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	92.1	61.1	61.1	50.7	0
2	67.9	49.4	49.4	37.4	0
3	112.2	78.1	68.3	64.3	14.3
4	115.9	63.6	59.7	94.1	6.5
5	98.5	70.7	70.4	39.9	0.4
6	82.8	71.7	60.7	36.4	18.2
7	97.0	78.0	66.3	46.3	17.6
8	176.2	135.3	56.0	214.3	141.6
9	105.1	59.0	49.6	111.9	20.4
10	137.9	87.5	53.1	159.7	64.8

מהנתונים המופיעים בטבלה, ניתן לראות שסטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעות בין 36.4% לבין 214.3%, כשהממוצע הוא 85.5 וסטיית התקן היא 57.3. סטיות התוצאות שהתקבלו מהאלגוריתם STDVRP נעות בין 0% ל-141.6%, כשממוצע הוא 28.4 וסטיית התקן היא 41.8.

נראה שעבור התפלגות נורמאלית, אלגוריתם STDVRP מתקשה לספק תוצאות שקרובות לתוצאה האופטימאלית. למרות זאת, אלגוריתם Saving, מספק תוצאות שהן פי 3 רחוקות ביותר בממוצע מהתוצאות של אלגוריתם STDVRP.

4.3.2.19 תוצאות קבוצה מס׳ 19

מס׳ יחידות זמן : 1

התפלגות : נורמאלית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן : 10

אחוז הקשתות המתנהגות סטוכסטית : 100%

טווח מהירויות הנסיעה: 50-120

בעקבות התוצאות שהתקבלו עבור קבוצת הבעיות הקודמת, ואשר היו גרועות יחסית לתוצאות שהתקבלו עד כה בקבוצות הבעיות האחרות, אנו מעונינים לבדוק אם ההשפעה העיקרית היא העובדה שמדובר בהתפלגות נורמאלית, או שזה בגלל שמספר ההסתברויות למהירות לכל קשת ביחידת זמן הייתה 10.

ב-10 הבעיות שבקבוצה זו דומות לבעיות שהיו בקבוצה הקודמת, אחוז הקשתות המתנהגות סטוכסטית הוא 100%, טווח המהירויות הוא טווח רחב, 50-120, אולם הפעם, מספר ההסתברויות למהירות לכל קשת ביחידת זמן הוא 10 במקום 100.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	67.5	64.0	55.7	21.2	14.9
2	54.7	49.4	49.4	10.7	0
3	52.9	48.2	47.4	11.6	1.7
4	55.9	53.3	47.6	17.4	12.0
5	50.4	47.8	40.9	23.2	16.9
6	60.3	57.3	51.8	16.4	10.6
7	51.4	51.5	45.5	13.0	13.2
8	54.3	54.3	48.4	12.2	12.2
9	45.5	42.4	39.6	14.9	7.1
10	52.7	51.0	49.5	6.5	3.0

כפי שניתן לראות מהנתונים המופיעים בטבלה, סטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעות בין 6.5% ל-23.2%, כשהממוצע הוא 14.7 וסטיית התקן היא 4.8. סטיות התוצאות שהתקבלו מהאלגוריתם STDVRP נעות בין 0% ל-16.9%, כשממוצע הוא 9.2 וסטיית התקן היא 5.6.

על פי התוצאות שהתקבלו, נראה שעבור התפלגות נורמאלית, אלגוריתם STDVRP יכול לספק תוצאות שקרובות לתוצאה האופטימאלית, בתנאי מספר ההסתברויות למהירות לכל קשת ביחידת זמן אינו גדול.

4.3.2.20 תוצאות קבוצה מס' 20

מס' יחידות זמן: 24

התפלגות: אמפירית

מספר ההסתברויות למהירות לכל קשת ביחידת זמן: 10

אחוז הקשתות המתנהגות סטוכסטית: 100%

טווח מהירויות הנסיעה: 50-120

קבוצת הבעיות הבאה, המכילה 20 בעיות, היא שילוב של הבעיה תלויית הזמן והבעיה הסטוכסטית. לאחר שבחננו את המקרה תלוי הזמן בנפרד, וראינו שאין השפעה רבה של מספר

חלקי הזמן על איכות התוצאות שמתקבלות מאלגוריתם STDVRP, וכמו כן, בחנו היבטים סטוכסטיים שונים של הבעיה, וגם לגביהם ראינו שאמנם ישנם הבדלים שונים בתוצאות, אבל לא הבדלים משמעותיים, נותר לנו לבדוק כיצד אלגוריתם STDVRP מתמודד עם הבעיה המשולבת. כאמור, מכוון שלא ראינו שיש השפעה מרובה של מספר חלקי הזמן על האלגוריתם, אנו נבחר את המקרה הקשה יותר, שבו הזמן מחולק ל-24 יחידות. גם מבחינת המדדים הסטוכסטיים נבחרו המקרים הקשים יותר, טווח זמן רחב של 50-120, אחוז קשתות המתנהגות סטוכסטית של 100% ומספר ההסתברויות למהירות לכל קשת ביחידת זמן היא 5.

מס"ד	תוצאת Savings	תוצאת האלגוריתם	תוצאה אופטימאלית	אחוז הסטייה בין ה-Savings לתוצאה האופטימאלית	אחוז הסטייה בין האלגוריתם לתוצאה האופטימאלית
1	72.3	71.8	69.8	3.6	2.9
2	71.2	58.4	58.4	21.9	0
3	92.9	74.7	68.9	34.8	8.4
4	84.2	69.7	69.7	20.8	0
5	79.9	73.9	73.9	8.1	0
6	102.8	87.3	75.4	36.3	15.8
7	83.9	76.3	71.7	17.0	6.4
8	91.7	90.1	79.4	15.5	13.5
9	83.7	80.6	79.5	5.3	1.4
10	77.3	75.6	68.3	13.2	10.7
11	83.4	76.3	72.0	15.8	6.0
12	84.7	78.4	74.8	13.2	4.8
13	81.0	77.4	76.0	6.6	1.8
14	101.6	88.0	73.6	38.0	19.6
15	92.0	75.6	69.9	31.6	8.2
16	93.7	84.9	80.0	17.1	6.1
17	77.8	74.0	74.0	5.1	0
18	103.0	95.9	82.6	24.7	16.1
19	77.3	62.6	62.6	23.5	0
20	73.6	73.3	73.3	0.4	0

כפי שניתן לראות מהנתונים המופיעים בטבלה, סטיות התוצאות שהתקבלו עבור אלגוריתם ה-Savings נעות בין 0.4% ל-38.0%, כשהממוצע הוא 17.6 וסטיית התקן היא 11.0. סטיות התוצאות שהתקבלו מהאלגוריתם STDVRP נעות בין 0% ל-19.6%, כשממוצע הוא 6.1 וסטיית התקן היא 6.1.

התוצאות מראות שאלגוריתם STDVRP מסוגל להתמודד בצורה יפה בבעיות שהן תלויות זמן וסטוכסטיות. אנו רואים שממוצע הסטיות של אלגוריתם STDVRP מהתוצאה האופטימאלית הוא 6.1%, סטייה שנחשבת נמוכה במושגים של אלגוריתם Savings, עליו מבוסס אלגוריתם STDVRP. בנוסף, כפי שניתן לראות, תוצאות אלגוריתם STDVRP טובות יותר מאשר תוצאות אלגוריתם Savings הפשוט, ולכן עבור בעיות תלויות זמן וסטוכסטיות, כדאי להשתמש באלגוריתם STDVRP.

4.3.2.21 השפעת מס׳ המועמדים על תוצאת האלגוריתם

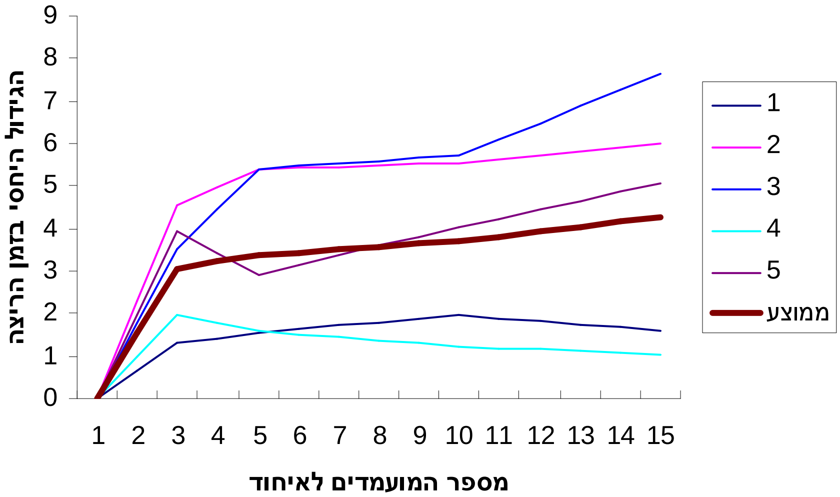
פרמטר נוסף באלגוריתם אשר ניתן לשינוי ויכול להשפיע על דיוק האלגוריתם, הוא מספר המסלולים המועמדים לאיחוד, ושאיחודם נבדק בעזרת סימולציה. בכדי לבדוק את השפעת מס׳ המסלולים המועמדים לאיחוד על דיוק האלגוריתם ניבנו 20 תסריטי בדיקה באופן אקראי, כך ש-5 הכילו 50 לקוחות, 5 נוספים הכילו 75 לקוחות, עוד 5 הכילו 100 לקוחות וה-5 הנותרים הכילו 150 לקוחות. כל אחד מתסריטי הבדיקה הורץ 5 פעמים, כאשר בפעם הראשונה מס׳ המסלולים המועמדים לאיחוד הוא 1, בפעם השניה הוא 3, בשלישית 5, ברביעית 10 ובחמישית 15. עבור כל הרצה נרשמו הן התוצאה והן זמן הריצה, כאשר המטרה היא לראות את אחוז השיפור בתוצאה ביחס לזמן ריצת האלגוריתם.

הטבלה הבאה מסכמת את סך הריצות של האלגוריתם.

מס״ד	מספר הלקוחות		תוצאות האלגוריתם כאשר מס׳ המועמדים לבדיקה הוא :				
			1	3	5	10	15
1	50	תוצאה	293.0	289.2	288.4	287.2	288.3
		זמן הריצה	146	401	643	1232	1753
2		תוצאה	286.3	273.2	270.9	270.4	269.2
		זמן הריצה	154	433	688	1340	1915
3		תוצאה	302.4	291.7	286.0	285.0	279.3
		זמן הריצה	156	396	665	1298	1936
4		תוצאה	329.6	323.1	324.4	325.6	326.2
		זמן הריצה	109	273	431	829	1218
5		תוצאה	292.9	281.3	284.3	281.1	278.0
		זמן הריצה	145	281	626	1227	1762
6	75	תוצאה	516.3	507.3	509.3	502.3	505.5
		זמן הריצה	157	384	599	1151	1675
7		תוצאה	493.7	486.9	485.1	479.3	480.7
		זמן הריצה	174	437	682	1328	1931
8		תוצאה	616.1	604.5	604.6	601.0	599.6
		זמן הריצה	123	282	434	814	1192
9		תוצאה	445.0	442.5	443.2	443.4	442.4
		זמן הריצה	220	564	914	1775	2567
10		תוצאה	450.7	447.5	440.3	431.8	440.1
		זמן הריצה	203	528	841	1601	2439
11	100	תוצאה	565.6	560.7	557.1	548.3	541.0
		זמן הריצה	339	841	1353	2640	3934
12		תוצאה	676.5	681.4	676.3	676.3	660.9
		זמן הריצה	237	536	850	1575	2324

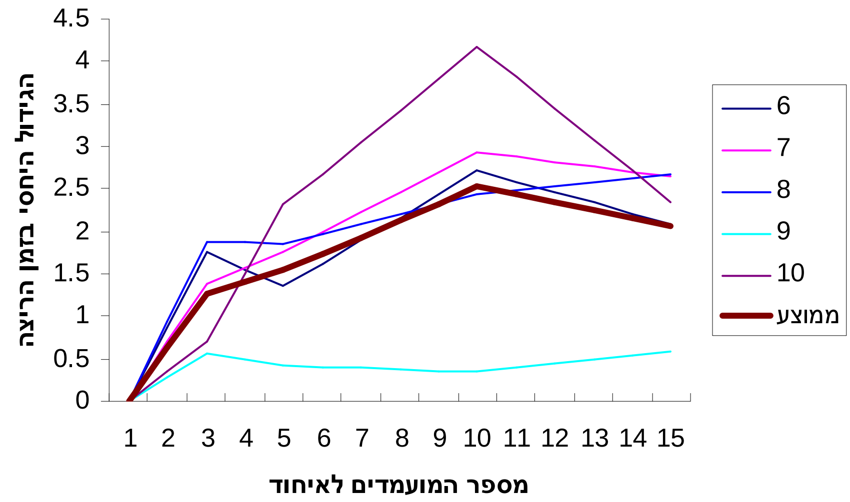
מס"ד	מספר הלקוחות		תוצאות האלגוריתם כאשר מס' המועמדים לבדיקה הוא:				
			1	3	5	10	15
13		תוצאה	585.0	582.2	582.4	581.3	574.4
		זמן הריצה	313	777	1216	2289	3407
14		תוצאה	803.5	782.7	772.8	774.0	774.8
		זמן הריצה	193	424	651	1212	1771
15		תוצאה	597.4	580.7	578.5	569.3	569.5
		זמן הריצה	292	715	1121	2198	3210
16	150	תוצאה	851.8	846.4	829.0	832.5	833.4
		זמן הריצה	590	1394	2130	3999	5714
17		תוצאה	1058.6	1051.4	1058.2	1059.6	1056.9
		זמן הריצה	414	840	1245	2211	3220
18		תוצאה	876.1	850.4	848.8	842.2	841.1
		זמן הריצה	562	1248	1900	3555	5221
19		תוצאה	968.9	964.2	960.3	951.9	956.4
		זמן הריצה	458	942	1421	2585	3817
20		תוצאה	957.5	952.3	935.1	942.2	934.5
		זמן הריצה	481	992	1516	2734	3959

בכדי לראות בצורה נוחה כיצד משפיע מספר המסלולים המועמדים לאיחוד, ושאיחודם נבדק בעזרת סימולציה על תוצאת האלגוריתם, אנו נציג את התוצאות שהתקבלו על גבי גרפים.



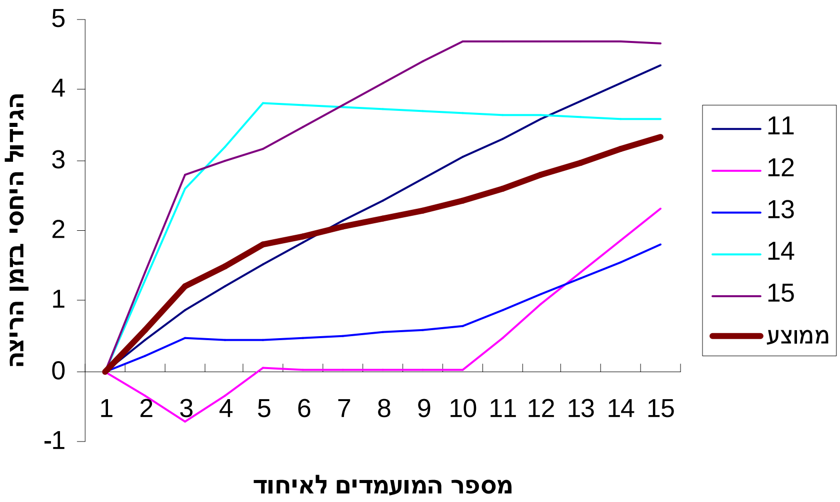
תרשים 10 - השיפור בתוצאת האלגוריתם באחוזים כפונקציה של מספר המועמדים לאיחוד בבעיות בנות 50 לקוחות

מהגרף קל לראות, שעבור 50 לקוחות, השיפור המשמעותי בתוצאת האלגוריתם מתקבל כאשר מספר המסלולים המועמדים לאיחוד הוא 3 (שיפור ממוצע של כ-3% בתוצאה המתקבלת משימוש במועמד אחד לאיחוד). לרוב שימוש במספר גדול יותר של מסלולים מועמדים לאיחוד נותן תוצאה טובה יותר, אולם השיפור הוא משמעותית קטן יותר (לדוגמא, שימוש ב-15 מועמדים לאיחוד נותן בממוצע שיפור של 4% בתוצאה המתקבלת משימוש במועמד אחד לאיחוד), ובהתחשב בפרק הזמן הנדרש להשגת התוצאה, אשר נמצא ביחס ישר לכמות המועמדים לאיחוד, נראה שברוב המקרים, אין סיבה להשתמש בערך הגדול מ-3.



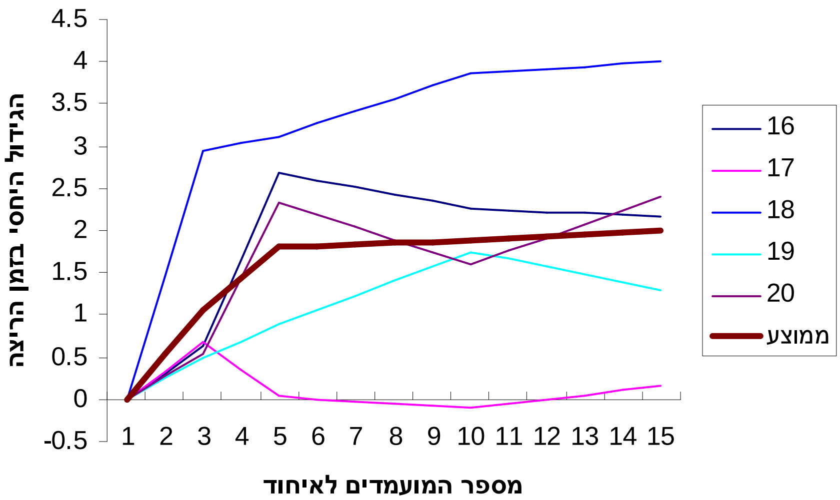
תרשים 11- השיפור בתוצאת האלגוריתם באחוזים כפונקציה של מספר המועמדים לאיחוד בבעיות בנות 75 לקוחות

גם עבור 75 לקוחות, השיפור המשמעותי בתוצאת האלגוריתם מתקבל כאשר מספר המסלולים המועמדים לאיחוד הוא 3 (שיפור ממוצע של קרוב ל-1.5% בתוצאה המתקבלת משימוש במועמד אחד לאיחוד). הגדלה של מספר המועמדים לאיחוד מ-3 ל-10 מגדילה את השיפור בתוצאה באחוז נוסף. אך גם במקרה זה השיפור מתון יותר לעומת השימוש ב-3 מועמדים לאיחוד. שימוש במספר גדול יותר של מועמדים מ-10, מקטין את בשיפור בתוצאה. מעבר מ-10 מועמדים לאיחוד מ-15 מועמדים, הקטין את השיפור בכאחוז.



תרשים 12 - השיפור בתוצאת האלגוריתם באחוזים כפונקציה של מספר המועמדים לאיחוד בבעיות בנות 100 לקוחות

בעוד שעבור 50 ו-75 לקוחות ניתן לראות מגמה ברורה של שיפור בתוצאה ככל שמספר המועמדים לאיחוד גדל, עבור 100 לקוחות המגמה לא כל כך חד משמעית. באופן כללי, גידול מספר המועמדים לאיחוד משפר את תוצאת האלגוריתם, אולם ניתן לראות שישנם מיקרים שבהם הגדלת מספר המועמדים לאיחוד פוגמת בתוצאת האלגוריתם. גם מבין הבעיות שבהם הוצג שיפור, לא ניתן להגיד שיש מגמה כללית אחידה לכל הבעיות. בממוצע ניתן לאמר שהשיפור המשמעותי מתקבל משימוש ב-3 מועמדים לאיחוד (שיפור של כאחוז אחד), מעבר לכך, שימוש ב-5 מועמדים לאיחוד, משפר את התוצאה בכחצי אחוז נוסף, ואילו מעבר ל-15 מועמדים לאיחוד, משפר את התוצאה בכאחוז אחד נוסף.



תרשים 13 - השיפור בתוצאת האלגוריתם באחוזים כפונקציה של מספר המועמדים לאיחוד בבעיות בנות 150 לקוחות

עבור 150 לקוחות, השיפור המשמעותי בתוצאת האלגוריתם מתקבל כאשר מספר המסלולים המועמדים לאיחוד הוא 5 (שיפור ממוצע של קרוב ל-1.5% בתוצאה המתקבלת משימוש במועמד אחד לאיחוד). הגדלה של מספר המועמדים לאיחוד מ-5 ל-10 מגדילה את השיפור הממוצע בפחות מחצי אחוז, ואף ניתן לראות מקרים שבהם השיפור הוא שלילי, כלומר מתקבלת תוצאה פחות טובה מאשר בשימוש ב-5 מועמדים לאיחוד.

בסיכום של ארבע קבוצות הבדיקה שלנו, ניתן לראות ששימוש ב-3 עד 5 מועמדים לאיחוד נותן את השיפור המשמעותי ביותר בתוצאת האלגוריתם. שימוש במספר גדול יותר של מועמדים משפר לרוב את התוצאה, אך משך הזמן שמוקדש לחיפוש התוצאה האופטימאלית גדל ביחס ישר לכמות המועמדים לאיחוד, בעוד שהשיפור הוא קטן.

5. דיון, מסקנות והצעות להמשך

5.1. מסקנות

בסעיף 4.3 בחנו את אלגוריתם STDVRP ואת תוצאותיו עבור בעיות קטנות (שמונה לקוחות) בהשוואה לתוצאות אלגוריתם Savings והפתרונות האופטימאליות. את האלגוריתם בחנו בעזרת 220 בעיות, כאשר כל קבוצת בעיות התמקדה בבחינה של היבט אחד של בעיית ניתוב הרכבים המאופיינים במשכי נסיעה תלויי זמן וסטוכסטיים. כל הבעיות הכילו 7 לקוחות + מחסן. מספר זה נבחר בכדי שנוכל בין היתר למצוא את הפתרון האופטימאלי של כל בעיה (ע״י מעבר על כל הפתרונות האפשריים), ואז נוכל להשוות את תוצאות אלגוריתמים Savings ו- STDVRP לפתרון האופטימאלי, ולהשוות בניהם.

40 בעיות התמקדו בבחינת השפעת התלות בזמן על אלגוריתם STDVRP. ב-40 בעיות אלו, ההיבט הסטוכסטי לא היה קיים. הבעיות התחלקו ל-4 קבוצות, כל קבוצה בת 10 בעיות. בבעיות בקבוצה הראשונה הייתה חלוקה ל-2 פרקי זמן, בקבוצה השניה ל-6 פרקי זמן, בשלישית ל-12 וברביעית הייתה חלוקה ל-24 פרקי זמן.

מספר קבוצה	טווח תוצאות אלגוריתם Savings	ממוצע אלגוריתם Savings	סטיית תקן אלגוריתם Savings	טווח תוצאות אלגוריתם STDVRP	ממוצע אלגוריתם STDVRP	סטיית תקן אלגוריתם STDVRP
1	0-20.2	7.7	6.3	0-15.4	5.9	5.1
2	0-15.3	5.1	5.4	0-15.3	4.3	5.7
3	0-26.4	9.7	9.2	0-23.8	7.8	8.0
4	0-21.1	11.3	8.7	0.5-18.7	11.4	6.4

טבלה 2 - השפעת תלות הזמן על תוצאות אלגוריתם STDVRP

כאשר בבעיה היו שני פרקי זמן, הסטייה הממוצעת של אלגוריתם STDVRP הייתה 5.9, ככל שמספר פרקי הזמן גדל, כך גדלה גם הסטייה הממוצעת, כך כשמספר פרקי הזמן היה 24, הסטייה הייתה 11.4, כמעט פי 2 מאשר בבעיות בהן יש שני פרקי זמן בלבד. גם אלגוריתם Savings התנהג באופן דומה, עבור בעיות שבהן היו שני פרקי זמן, הסטייה הממוצעת הייתה 7.7, כאשר עבור בעיות עם 24 פרקי זמן, הסטייה הממוצעת הייתה 11.3. כאשר לרוב, תוצאות אלגוריתם Savings רחוקות יותר מהפתרון האופטימאלי לעומת תוצאות אלגוריתם STDVRP.

180 בעיות בחנו השפעת גורמים סטוכסטיים על תוצאות אלגוריתם STDVRP. הגורמים שנבדקו הם אחוז הקשתות המתנהגות סטוכסטית, מספר ההסתברויות למהירות לכל קשת ביחידת זמן וטווח המהירויות.

מספר קבוצה	טווח תוצאות אלגוריתם Savings	ממוצע אלגוריתם Savings	סטיית תקן אלגוריתם Savings	טווח תוצאות אלגוריתם STDVRP	ממוצע אלגוריתם STDVRP	סטיית תקן אלגוריתם STDVRP
5	4.6-20.6	10.7	5.3	0-9.2	2.6	3.4
7	7.2-40.2	21.7	11.1	0-15.6	6.0	5.0
6	(-1.5)-17.1	7.2	5.4	0-6.6	1.9	2.5
8	2-22.1	7.9	6.2	0-8.9	3.2	2.7
9	4.6-35.9	10.7	5.3	0-31.3	7.4	9.9
11	5.7-27.5	18.9	7.3	0-15.4	4.9	4.4
10	0.8-19.3	8.9	5.7	0-15.8	4.3	4.5
12	0.2-18.6	9.3	5.0	0-15.3	3.3	4.5

מספר קבוצה	טווח תוצאות אלגוריתם Savings	ממוצע אלגוריתם Savings	סטיית תקן אלגוריתם Savings	טווח תוצאות אלגוריתם STDVRP	ממוצע אלגוריתם STDVRP	סטיית תקן אלגוריתם STDVRP
13	8.4-34.7	14.5	7.6	0-17.6	6.4	5.1
15	6.0-55.2	21.7	15.5	0-12.1	4.3	4.6
17	8-20.6	14.2	4.5	1.8-19.4	11.3	5.3
14	0-18.2	7.7	6.5	0-14.3	5	5
16	2.7-14.7	8	4.5	0-10	3.2	3.3

טבלה 3 - השפעת מספר ההסתברויות למהירות לכל קשת ביחידת זמן על תוצאות אלגוריתם STDVRP

הגדלת מספר ההסתברויות למהירות לכל קשת ביחידת זמן מגדילה את סיבוכיות הבעיה, ולכן הגיוני לצפות שבמקרים שבהם מספר ההסתברויות למהירות לכל קשת ביחידת זמן תהייה גדולה יותר, סטיית תוצאות אלגוריתם STDVRP מהתוצאה האופטימאלי תהייה גדולה יותר. אולם כפי שניתן לראות מהתוצאות, לא בכל המקרים הסטייה אכן הייתה גדולה יותר. עבור זוגות הבעיות 5 ו-7, 6 ו-8, 13,15 ו-17, אכן התקבלה סטיית גדולה יותר מהתוצאה האופטימאלית כאשר מספר ההסתברויות למהירות לכל קשת ביחידת זמן, לעומת זאת עבור קבוצת הבעיות 9 ו-11, 10 ו-12, ו-14 ו-16 התקבלו תוצאות הפוכות. מה שניתן לראות הוא שהמשותף בין הקבוצות 5 ו-7 וכן 6 ו-8 הוא אחוז הקשתות המתנהגות סטוכסטית שהינו 25%, המשותף בין קבוצות 9 ו-11 וכן 10 ו-12 הוא אחוז הקשתות המתנהגות סטוכסטית שהינו 50%. בשאר הקבוצות 13, 15 ו-17 וכן 14 ו-16, אחוז הקשתות המתנהגות סטוכסטית הוא 100%. לכן קשה להגיד שעליה באחוז הקשתות המתנהגות סטוכסטית יש לה השפעה המבטלת את השפעת מספר ההסתברויות למהירות לכל קשת ביחידת זמן. יש לציין, שעבור אלגוריתם Savings התוצאות מראות שהאלגוריתם מתנהג כפי שציפינו ממנו, כלומר בבעיות שבהן מספר ההסתברויות למהירות לכל קשת ביחידת זמן גדול יותר תוצאות האלגוריתם רחוקות יותר בממוצע מהתוצאה האופטימאלית.

מספר קבוצה	טווח תוצאות אלגוריתם Savings	ממוצע אלגוריתם Savings	סטיית תקן אלגוריתם Savings	טווח תוצאות אלגוריתם STDVRP	ממוצע אלגוריתם STDVRP	סטיית תקן אלגוריתם STDVRP
5	4.6-20.6	10.7	5.3	0-9.2	2.6	3.4
6	(-1.5)-17.1	7.2	5.4	0-6.6	1.9	2.5
7	7.2-40.2	21.7	11.1	0-15.6	6.0	5.0
8	2-22.1	7.9	6.2	0-8.9	3.2	2.7
9	4.6-20.6	10.7	5.3	0-31.3	7.4	9.9
10	0.8-19.3	8.9	5.7	0-15.8	4.3	4.5
11	5.7-27.5	18.9	7.3	0-15.4	4.9	4.4
12	0.2-18.6	9.3	5.0	0-15.3	3.3	4.5
13	8.4-34.7	14.5	7.6	0-17.6	6.4	5.1
14	0-18.2	7.7	6.5	0-14.3	5	5
15	6.0-55.2	21.7	15.5	0-12.1	4.3	4.6
16	2.7-14.7	8	4.5	0-10	3.2	3.3

טבלה 4 - השפעת טווח המהירויות על תוצאות אלגוריתם STDVRP

היבט נוסף שנבחן הוא השפעת טווח המהירויות על תוצאות אלגוריתם STDVRP. מכוון שלמהירות השפעה ישירה על הזמן הרי שטווח המהירויות משמעו טווח זמני הנסיעה. טווח מהירויות קטן משמעו פחות שינויים. ככל שמספר השינויים או טווח השינויים קטן יותר, הבעיה מתקרבת יותר לבעיה הדטרמיניסטית, ולכן הגיוני שעבור בעיות שבהן טווח המהירויות קטן יותר תוצאות אלגוריתם STDVRP תהיינה קרובות יותר לתוצאות האופטימאליות מאשר בעיות שבהן

טווח המהירויות גדול יותר. הנתונים שהתקבלו מהבדיקות שערכנו, אכן תומכים בטענה זו. נתונים דומים התקבלו גם עבור אלגוריתם Savings, כך שגם אצלו, ככל שטווח המהירויות היינו קטן יותר, התוצאות קרובות יותר לתוצאות האופטימאליות.

ההיבט האחרון שנבדק הוא השפעת אחוז הקשתות שמתנהגות סטוכסטית על תוצאת אלגוריתם STDVRP.

מספר קבוצה	טווח תוצאות אלגוריתם Savings	ממוצע אלגוריתם Savings	סטיית תקן אלגוריתם Savings	טווח תוצאות אלגוריתם STDVRP	ממוצע אלגוריתם STDVRP	סטיית תקן אלגוריתם STDVRP
5	4.6-20.6	10.7	5.3	0-9.2	2.6	3.4
9	4.6-20.6	10.7	5.3	0-31.3	7.4	9.9
13	8.4-34.7	14.5	7.6	0-17.6	6.4	5.1
6	(-1.5)-17.1	7.2	5.4	0-6.6	1.9	2.5
10	0.8-19.3	8.9	5.7	0-15.8	4.3	4.5
14	0-18.2	7.7	6.5	0-14.3	5	5
7	7.2-40.2	21.7	11.1	0-15.6	6.0	5.0
11	5.7-27.5	18.9	7.3	0-15.4	4.9	4.4
15	6.0-55.2	21.7	15.5	0-12.1	4.3	4.6
8	2-22.1	7.9	6.2	0-8.9	3.2	2.7
12	0.2-18.6	9.3	5.0	0-15.3	3.3	4.5
16	2.7-14.7	8	4.5	0-10	3.2	3.3

טבלה 5 - השפעת אחוז הקשתות המתנהגות סטוכסטית על תוצאות אלגוריתם STDVRP

ככל שאחוז הקשתות שמתנהג סטוכסטית הוא קטן יותר, הבעיה יותר קרובה לבעיה דטרמיניסטית, ולכן אנו מצפים שסטיית אלגוריתם STDVRP מהפתרון האופטימאלי תהיה קטנה יותר. עבור קבוצות הבעיות 5, 9 ו-13 וכן 6, 10 ו-14, תוצאות אלגוריתם STDVRP מתאימות לציפיות שלנו. אולם עבור קבוצות הבעיות 7, 11 ו-15 וכן 8, 12 ו-16, שינוי באחוז הקשתות שמתנהג סטוכסטית אינו גורר עמו שינוי בסטיית תוצאת אלגוריתם STDVRP מהפתרון האופטימאלי. המשותף לקבוצות 5, 9 ו-13 וכן 6, 10 ו-14 הוא מספר ההסתברויות למהירות לכל קשת ביחידת זמן שהינו 3. המשותף לקבוצות 7, 11 ו-15 וכן 8, 12 ו-16 הוא מספר ההסתברויות למהירות לכל קשת ביחידת זמן שהינו 5. אנו רואים, שהגדלה של מספר ההסתברויות למהירות לכל קשת ביחידת זמן מבטלת את השפעת אחוז הקשתות המתנהגות סטוכסטית על תוצאת אלגוריתם STDVRP. השפעה של אחוז הקשתות המתנהגות סטוכסטית על תוצאות אלגוריתם Savings לא נראה שקיימת.

נסכם את המסקנות שיש לנו עד כה (המסקנות רלוונטיות לבעיות קטנות בסדר גודל של שמונה לקוחות).

1. הגדלת מספר פרקי הזמן בבעיה, מגדילה את הפער בין תוצאת אלגוריתמים Savings ו-STDVRP לפתרון האופטימאלי.
2. הגדלת אחוז הקשתות שמתנהגות סטוכסטית, מגדילה את הפער בין תוצאות אלגוריתם STDVRP לפתרון האופטימאלי. הגדלת אחוז הקשתות שמתנהגות סטוכסטית אינה משפיעה על תוצאת אלגוריתם Savings.

3. הגדלת מספר ההסתברויות למהירות לכל קשת ביחידת זמן, מגדילה את הפער בין תוצאות אלגוריתמים Savings ו-STDVRP לפתרון האופטימאלי.
4. למספר ההסתברויות למהירות לכל קשת ביחידת זמן, השפעה גדולה יותר מאשר לאחוז הקשתות המתנהגות סטוכסטית באלגוריתם STDVRP בלבד וזאת מכוון שלמספר ההסתברויות למהירות לכל קשת ביחידת זמן אין השפעה על אלגוריתם Savings
5. טווח מהירויות קטן יותר מאפשר לאלגוריתמים Savings ו-STDVRP להגיע לתוצאות שהנן קרובות יותר לפתרון האופטימאלי.

בהתאם למסקנות אלו, נבנו 20 בעיות, המאופיינות בטווח מהירויות גדול (50-120), מספר הסתברויות למהירות לכל קשת ביחידת זמן שהינו יחסית גדול (5), אחוז הקשתות המתנהגות סטוכסטית הוא 100% ומספר יחידות הזמן הוא 24. מטרת הבעיות הללו היא לבחון את איכות התוצאות המתקבלות מאלגוריתם STDVRP, כאשר הבעיה היא סטוכסטית ותלויית זמן.

מהתוצאות שהתקבלו עולה שבממוצע תוצאת אלגוריתם STDVRP סוטה בכ-6% מהתוצאה האופטימאלית, בעוד שאלגוריתם Saving סוטה בכ-17% מהתוצאה האופטימאלית. Cordeau, Gendreau, Potvin ו-Semet ([18]) ביצעו השוואה בין מספר אלגוריתמים נפוצים, ובין היתר הראו שכאשר משתמשים באלגוריתם Savings לפתרון CVRP (בעיה המקורית שעבורה הוא תוכנן), סטיית האלגוריתם מהפתרון האופטימאלי (או הפתרון הטוב ביותר הידוע), עומדת בממוצע על 6%. על סמך התוצאות שלנו, עבור בעיות סטוכסטיות ותלויות זמן, אלגוריתם STDVRP מספק פתרונות כשאחוז הסטייה מהתוצאה האופטימאלית קרוב לאחוז הסטייה של אלגוריתם Savings עבור CVRP.

בנוסף לבדיקת איכות הפתרונות של אלגוריתם STDVRP, נבדק גם השפעת מספר זוגות המסלולים המיועדים לאיחוד ואשר החיסכון שלהם מחושב בעזרת סימולציה. על סך תוצאות הבדיקות שנערכו, נראה ששימוש ב-3 עד 5 מועמדים לאיחוד נותן את השיפור המשמעותי ביותר בתוצאת האלגוריתם. שימוש במספר גדול יותר של מועמדים משפר לרוב את התוצאה, אך משך הזמן שמוקדש לחיפוש התוצאה האופטימאלית גדל ביחס ישר לכמות המועמדים לאיחוד, בעוד שהשיפור הוא קטן.

5.2. הצעות למחקרי המשך

הבעיה העיקרית בה נתקלנו בעת בדיקת אלגוריתם STDVRP, היא חוסר בבעיות דומות מולן ניתן להשוות את תוצאות האלגוריתם. הרצון להשוות את תוצאות אלגוריתם STDVRP מול התוצאות האופטימאליות אילץ אותנו להשתמש במספר נמוך של לקוחות בבעיות שלנו, מכוון שעבור מספר נמוך של לקוחות יש באפשרותנו לחשב את הפתרון האופטימאלי. לאחר שביססנו והראנו שאלגוריתם STDVRP אכן מספק תוצאות טובות, כדאי לדעתי להשקיע זמן מחשב, ולמצוא פתרונות אופטימאליים עבור בעיות גדולות יותר, ולבחון את אלגוריתם STDVRP בעזרת אותן הבעיות.

ניתן להרחיב את אלגוריתם STDVRP כך שיתמודד עם תנאים נוספים. בזמן חיפוש המסלול הקצר ביותר, אנו לא אפשרנו המתנות לפני שממשיכים במסלול לאחר ההגעה ללקוח. יתכנו מצבים שבהם המתנה במספר מקומות תאפשר לנו ליצור מסלולים שזמן הנסיעה בהם קצר יותר, למרות שמבחינת זמן היציאה והחזרה למחסן הם ארוכים יותר. אם המתנות אינן מפריעות לארגון, אפשר לשלבן במסלולים (למשל, אפשר למצוא נקודה מתאימה במסלול שבה הנהג מקבל הפסקה לארוחת צהרים).

אילוץ נוסף אשר לרוב קיים ואין אנו מתייחסים אליו, הוא חלונות זמן. חלונות זמן מכתיבים מתי יש להגיע ללקוח. האלגוריתם המוצע אינו מתמודד עם אילוץ זה. מכוון והשימוש בחלונות זמן נפוץ, כדאי לנסות ולהוסיף אותו לאלגוריתם.

אפשרויות הרחבה נוספות הנן שימוש במטה-היוריסטיקות, כגון Genetic Algorithms אשר יכול לאפשר השגת תוצאות אשר קרובות יותר לפתרון האופטימאלי.

6. סיכום

בעיית ניתוב הרכבים היא בעיה ששעות מחקר רבות הושקעו בה במהלך 45 השנים האחרונות. במהלך השנים נעשה ניסיון לפתח אלגוריתמים מדויקים יותר ומהירים יותר לפתרון בעיית ניתוב הרכבים. הנוסף נעשה ניסיון להביא את המודל קרוב יותר למציאות, ולספק אלגוריתמים מתאימים לפתרון הבעיות המורכבות יותר.

בעבודה זו הוצג מודל ניתוב רכבים המאופיין במשכי נסיעה תלויי זמן וסטוכסטיים. פונקצית המטרה של מודל זה היא הקטנת משך זמן הנסיעה בצירים השונים המוקצים לצורך חלוקת סחורה ללקוחות ממחסן מרכזי. מודל זה הוא הרחבה נוספת לבעיית ניתוב הרכבים שמטרתה לקרב את המודלים השונים צעד נוסף לקראת המציאות המורכבת.

החשיבות של מודל זה באה לידי ביטוי במצבים כגון מצבי חירום, בהם יש חשיבות רבה לזמן ההגעה ליעד. חברות המתעסקות בהובלת חומרים מסוכנים, אשר מעונינות להקטין את זמן השהות שלהם באזורים מאוכלסים יכולות למצוא גם הן עניין במודל זה. חברות שליחים הפועלות בתחומי הערים הגדולות, יוכלו לתכן את מסלולי הנסיעה שלהם, כך שמשך זמן הנסיעה יהיה מינימאלי, והן יוכלו להגיע ליותר לקוחות ביום עבודה.

בנוסף למודל הוצג אלגוריתם לפתרון בעיית ניתוב הרכבים המאופיינים במשכי נסיעה תלויי זמן וסטוכסטיים. האלגוריתם שהוצג מבוסס על אלגוריתם Savings שהוצג ע"י Clarke ו-Wright. מתוצאות בדיקות שנערכו עולה שבממוצע תוצאת האלגוריתם סוטה בכ-6% מהתוצאה האופטימאלית, בעוד שבניסיון לפתור את אותן הבעיות ע"י אלגוריתם Saving, התקבלו תוצאות שסוטות בכ-17% מהתוצאה האופטימאלית. על סמך התוצאות שלנו, עבור בעיות סטוכסטיות ותלויות זמן, אלגוריתם STDVRP מספק פתרונות כשאחוז הסטייה מהתוצאה האופטימאלית קרוב לאחוז הסטייה של אלגוריתם Savings עבור CVRP.

7. מקורות

[1] "Department of Health and Human Services, Center for Disease Control and Preventive, http://www.bt.cdc.gov/stockpile/."

[2] http://neo.lcc.uma.es/radi-aeb/WebVRP/Problem_Instances/CVRPTWInstances.html#Solomon".

[3] B. M. Baker and M. A. Ayechew, "A Genetic Algorithm for the Vehicle Routing Problem", Computers & Operations Research, 30, 2003, pp. 787-800.

[4] R. H. Ballou and Y. K. Agarwal, "A Performance Comparison of Several Popular Algorithms for Vehicle Routing and Scheduling", Journal of Business Logistics, 9, 1988, pp. 51-65.

[5] J. E. Bell and P. R. Mullen, "Ant Colony Optimization Techniques for the Vehicle Routing Problem", Advanced Engineering Informatics, 18, 2004, pp. 41-48.

[6] D. Belson, "Storage, Distribution, and Dispensing of Medical Supplies," Create Report, Center for Risk and Economic Analysis of Terrorism Events, University of Southern California, Los Angeles, California, 2005.

[7] J. Berger and M. Barkaoui, "A new hybrid genetic algorithm for the capacitated vehicle routing problem", Journal of the Operational Research Society, 54, 2003, pp. 1254-1262.

[8] D. J. Bertsimas, Probabilistic Combinatorial Optimization Problems, Massachusetts Institute of Technology, 1988.

[9] D. J. Bertsimas, "A Vehicle Routing Problem With Stochastic Demand", Operations Research, 40 (3), 1992, pp. 574-585.

[10] L. Boding, "Routing and Scheduling of Vehicles and Crew: The State of the Art", Computer and Operations Research, 10, 1983, pp. 63-211.

[11] E. H. Bowman, "Production Scheduling by the Transportation Method of Linear Programming", Operations Research, 4 (1), 1956, pp. 100-103.

[12] B. Bullnheimer, R. F. Hartl, and C. Strauss, "Applying the Ant System to the Vehicle Routing Problem", Proc. 2nd International Conference on Metaheuristics, 1997.

[13] W. Burrows, "The Vehicle Routing Problem With Loadsplitting; A Heuristic Approach", Proc. 24th Annual Conference of the Operational Research Society of New Zealand, 1988, pp. 33-38.

[14] A. Chance and W. W. Cooper, "Chance-Constrained Programming", Management Science, 6, 1959.

[15] S. Chen, B. Golden, and E. Wasil, "The Split Delivery Vehicle Routing Problem: Applications, Algorithms, Test Problems, and Computational Results", Networks, 49 (4), 2007, pp. 318-329.

[16] G. Clarke and J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", Operations Research, 12, 1964, pp. 568-581.

[17] M. H. Cole, Service Considerations and the design of strategic distribution systems, Georgia Institute of Technology, 1995.

[18] J. F. Cordeau, M. Gendreau, G. Laporte, J. Y. Potvin, and F. Semet, "A Guide to Vehicle Routing Heuristics", Journal of the Operational Research Society, 53, 2002, pp. 512-522.

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms.

[20] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a Large-Scale Traveling-Salesman Problem", Journal of the Operations Research Society of America, 2 (4), 1954, pp. 393-410.

[21] G. B. Dantzig and J. H. Tamser, "The Truck Dispatching Problem", Management Science, 6 (1), 1959, pp. 80-91.

[22] S. K. Das, Developing State-Dependent Routes for the Vehicle Routing Problem Under Uncertainty, The Pennsylvania State University, 2004.

[23] M. Desrochers, J. Desrosiers, and M. M. Solomon, "A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows", Operations Research, 40 (2), 1992, pp. 324-354.

[24] M. Dessouky, F. Ordóñez, H. Jia, and Z. Shen, "Rapid Distribution of Medical Supplies," Delay Management in Health Care System, ed., Springer, 2006.

[25] R. Diestel, Graph Theory - Electronic Edition, 3 ed., Springer-Verlag, New-York, 2005.

[26] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a Colony of Cooperating Agents", IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics, 26 (1), 1996, pp. 29-41.

[27] M. Dror, G. Laporte, and P. Trudeau, "Vehicle Routing with Stochastic Demands:Properties and Solution Frameworks", Transportation Science, 23 (3), 1989, pp. 166-176.

[28] M. Dror and P. Trudeau, "Savings by Split Delivery Routing", Tansportation Science, 23, 1989, pp. 141-145.

[29] M. Fischetti, P. Toth, and D. Vigo, "A Branch-And-Bound Algorithm for the Capacitated Vehicle Routing Problem on Directed Graphs", Operations Research, 42 (5), 1994, pp. 846-859.

[30] M. L. Fisher, R. Jaikumar, and L. N. V. Wassenhove, "A Multiplier Adjustment Method for the Generalized Assignment Problem", Management Science, 32 (9), 1986, pp. 1095-1103.

[31] L. R. Ford and D. R. Fulkerson, "Constructing Maximal Dynamic Flows from Static Flows", Operations Research, 6 (3), 1958, pp. 419-433.

[32] H. Gabbay, "An Overview of Vehicular Scheduling Problems", Sloan School of Management, MIT, 1973.

[33] M. Gendreau, A. Hertz, and G. Laporte, "A Tabu Search Heuristic for the Vehicle Routing Problem", Management Science, 40 (10), 1994, pp. 1276-1290.

[34] M. Gendreau, G. Laporte, C. Musaraganyi, and É. D. Tallard, "A TABU Search Heuristic for the Heterogeneous Fleet Vehicle Routing Problem", Computers & Operations Research, 26, 1999, pp. 1153-1173.

[35] M. Gendreau, G. Laporte, and R. Seguin, "Stochastic Vehicle Routing", European Journal of Operational Research, 88, 1996, pp. 3-12.

[36] B. E. Gillett and L. R. Miller, "A Heuristic Algorithm for the Vehicle-Dispatch Problem", Operations Research, 22 (2), 1974, pp. 340-349.

[37] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Massachusetts, 1989.

[38] B. Golden, T. L. Magnanti, and H. Q. Nguyen, "Implementing vehicle routing algorithms", Networks, 7, 1977, pp. 113-148.

[39] B. L. Golden and J. R. Yee, "A Freamework for Probalistic Vehicle Routing", IIE Transactions, 11 (2), 1979, pp. 109-112.

[40] C. M. Grinstead and J. L. Snell, "Introduction to Probability," AMS, 1997.

[41] F. S. Hillier and G. J. Lieberman, "Simulation," Introduction to Operations Research, 7 ed., McGraw-Hill, 2001, pp. 1084-1155.

[42] C. Hjorring, The Vehicle Routing Problem and Local Search Metaheuristics, The University of Auckland, 1995.

[43] S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Vehicle Routing with Time-Dependent Travel Times", European Journal of Operational Research, 144, 2003, pp. 379-396.

[44] S. Jung, A Genetic Algorithm for the Vehicle Routing Problem with Time-Dependent Travel Times, University of Maryland, 2000.

[45] P. Kall and S. W. Wallace, Stochastic Programming, 1 ed., John Wiley & Sons, New-York.

[46] A. S. Kenyon and D. P. Morton, "Stochastic Vehicle Routing with Random Travel Times", Transportation Science, 37 (1), 2003, pp. 69-82.

[47] Y. A. Koskosidis, W. B. Powell, and M. M. Solomon, "An Optimization-Based Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints", Transportation Science, 26 (2), 1992, pp. 69-85.

[48] G. Laporte, F. Louveaux, and H. Mercure, "The Vehicle Routing Problem With Stochastic Travel Times", Transportation Science, 26 (3), 1992, pp. 161-170.

[49] A. Larsen, The Dynamic Routing Problem, Technical University of Denmark 2000.

[50] P. D. Larson and J. D. Kulchitsky, "Logistics Improvement Programs: The Dynamics between People and Performance", International Journal of Physical Distribution and Logistics Management, 29, 1999, pp. 88-102.

[51] C. Malandraki and M. S. Daskin, "Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms", Transportation Science, 26 (3), 1992, pp. 185-200.

[52] P. H. J. Marguier and A. Ceder, "Passenger waiting strategies for overlapping bus routes." Transportation Science, 18 (3), 1984, pp. 207-230.

[53] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer Programming Formulation of the Traveling Salesman Problem", Journal of the Association for Computing Machinery, 7, 1960, pp. 326-329.

[54] P. A. Mullaseril, M. Dror, and J. Leung, "Split-Delivery Routeing Heuristic in Livestock Feed Distribution", Journal of the Operation Research Society, 48, 1997, pp. 107-116.

[55] J. C. Picard and M. Queyranne, "The Time-Dependent Traveling Salesman Problem and Its Application to the Tardiness Problem in One-Machine Scheduling", Operations Research, 26 (1), 1978, pp. 86-110.

[56] T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter, "On the Capacitated Vehicle Routing Problem", Mathematical Programming, 94 (2-3), 2003, pp. 343-359.

[57] M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", Operations Research, 35 (2), 1987, pp. 254-265.

[58] W. R. Stewart and B. L. Golden, "A chance-constrained approach to the stochastic vehicle routing problem", Proc. 1980 Northeast AIDS Conference, 1980, pp. 33-35.

[59] R. Stricker, Public Sector Vehicle Routing: The Chinese Postman Probelm, Massachusetts Institute of Technology, 1970.

[60] F. A. Tillman, "The Multiple Terminal Delivery Problem with Probabilistic Demands", Transportation Science, 3 (3), 1969, pp. 192-204.

[61] A. Wren and A. Holliday, "Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points", Operational Research Quarterly (1970-1977), 23 (3), 1972, pp. 333-344.

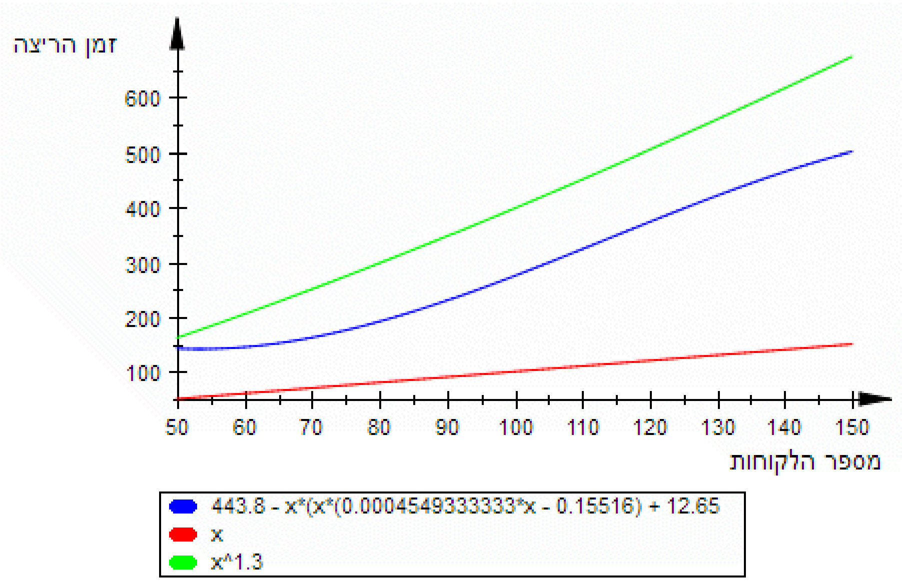
8. נספח א' - רשימת קיצורים

בעבודה זו אנו משתמשים במושגים רבים, אשר בספרות נהוג להתייחס אליהם בשמם המקוצר. בנספח זה מופיע רשימה של הקיצורים אשר נמצאים בשימוש בעבודה זו. לכל קיצור מוצמד שמו באנגלית ותרגומו לעברית.

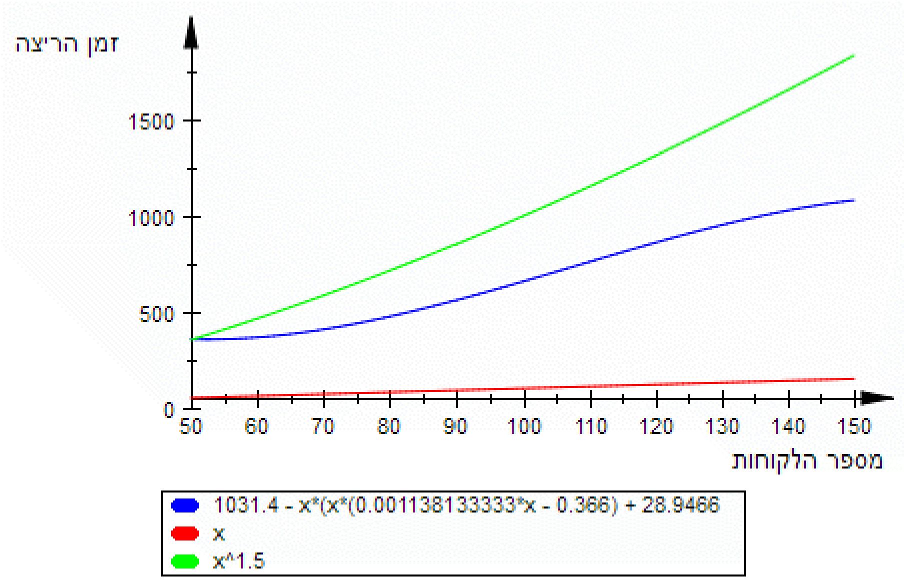
CCP	Chance Constrained Programming	תכנות תחת אילוצי הצלחה
CPP	Chinese Postman Problem	בעיית מחלק הדואר הסיני
CVRP	Capacitated Vehicle Routing Problem	בעיית ניתוב הרכבים תלויית קיבולת
GAP	Generalized Assignment Problem	בעיית השמה כללית
MDVRP	Multi-Depot Vehicle Routing Problem	בעיית ניתוב רכבים ממספר מחסנים
SDVRP	Split Delivery Vehicle Routing Problem	בעיית ניתוב רכבים עם חלוקת הובלות
SPR	Stochastic Programming	תכנות סטוכסטי
STDVRP	Stochastic Time Dependent Vehicle Routing Problem	בעיית ניתוב רכבים המאופיינת במשכי נסיעה תלויי זמן וסטוכסטיים
SVRP	Stochastic Vehicle Routing Problem	בעיית ניתוב רכבים סטוכסטית
TDTSP	Time Dependent Traveling Salesman Problem	בעיית הסוכן הנוסע תלוי זמן
TDVRP	Time Dependent Vehicle Routing Problem	בעיית ניתוב רכבים תלוי זמן
TSP	Traveling Salesman Problem	בעיית הסוכן הנוסע
VRP	Vehicle Routing Problem	בעיית ניתוב רכבים
VRPSCD	Vehicle Routing Problem with Stochastic Customers and Demands	בעיית ניתוב רכבים בעלת לקוחות ודרישות סטוכסטיים
VRPSD	Vehicle Routing Problem with Stochastic Demands	בעיית ניתוב רכבים עם דרישות סטוכסטיות
VRPSTT	Vehicle Routing Problem with Stochastic Traveling Time	בעיית ניתוב רכבים עם זמני נסיעה סטוכסטיים
VRPTW	Vehicle Routing Problem with Time Windows	בעיית ניתוב רכבים עם חלונות זמן

9. נספח ב' – הגידול בזמן הריצה כפונקציה של מספר הלקוחות – אינטרפולציה

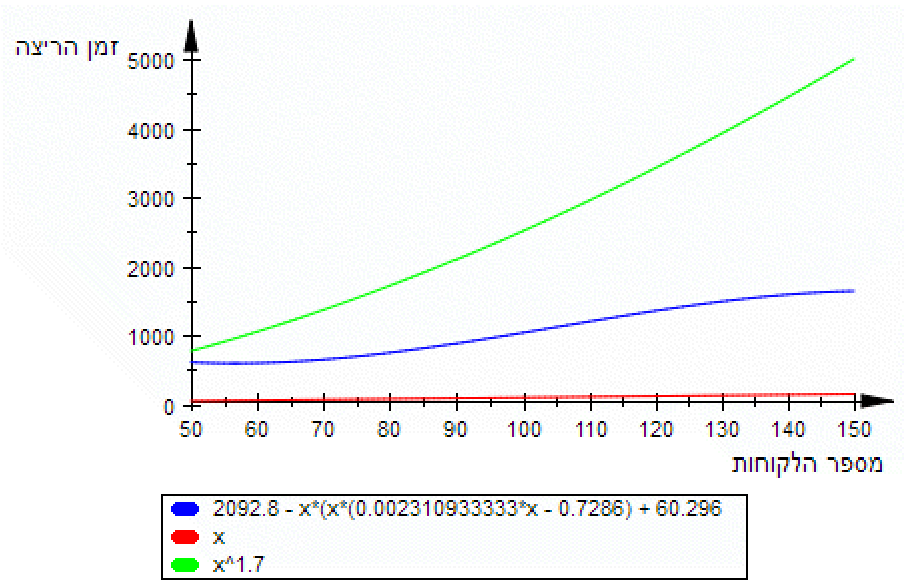
הגרפים הבאים מתארים את הגידול בזמן הריצה כפונקציה של מספר הלקוחות כפי שחושבו בעזרת אינטרפולציה מתמטית. בכל גרף פונקציית הגידול מצוינת ע"י העקום האמצעי, אשר מתחתיו מסורטט העקום $y = x$, ומעליו משורטט עקום בעל המבנה $y = x^a$, אשר חוסם אותו מלמעלה.



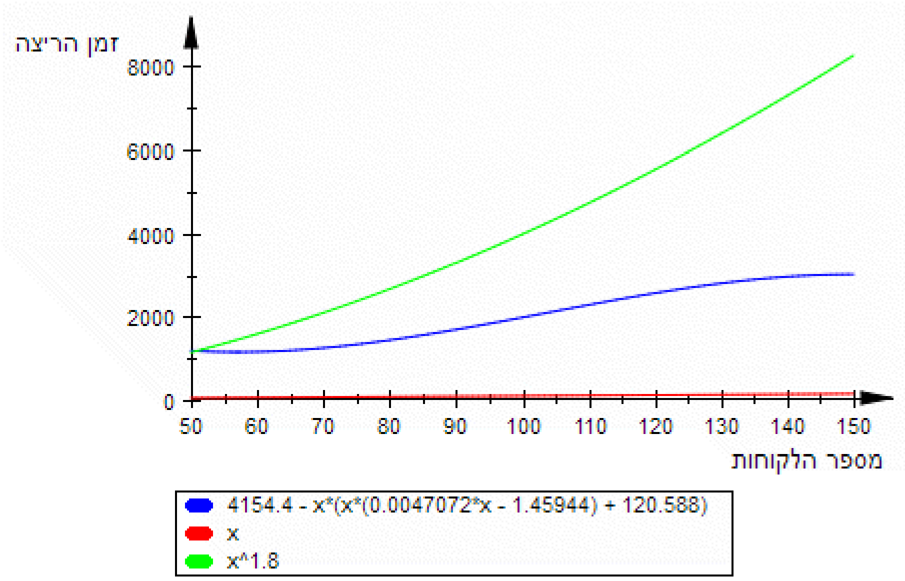
תרשים 14 - פונקציית הגידול בזמן הריצה כפונקציה של מספר הלקוחות כפי שחושבה בעזרת אינטרפולציה כאשר מספר המועמדיים לאיחוד הוא 1.



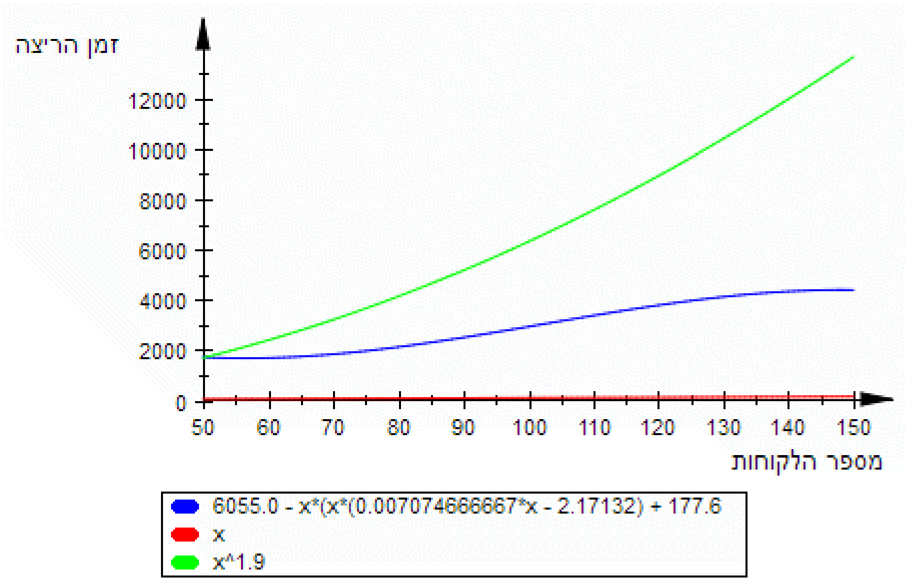
תרשים 15 - פונקציית הגידול בזמן הריצה כפונקציה של מספר הלקוחות כפי שחושבה בעזרת אינטרפולציה כאשר מספר המועמדיים לאיחוד הוא 3.



תרשים 16 - פונקציית הגידול בזמן הריצה כפונקציה של מספר הלקוחות כפי שחושבה בעזרת אינטרפולציה כאשר מספר המועמדיים לאיחוד הוא 5.



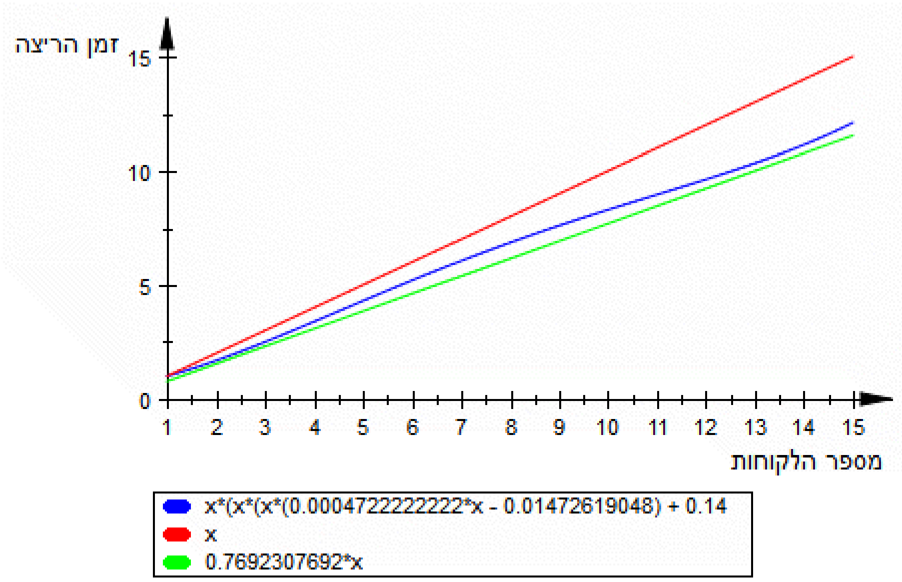
תרשים 17 - פונקציית הגידול בזמן הריצה כפונקציה של מספר הלקוחות כפי שחושבה בעזרת אינטרפולציה כאשר מספר המועמדיים לאיחוד הוא 10.



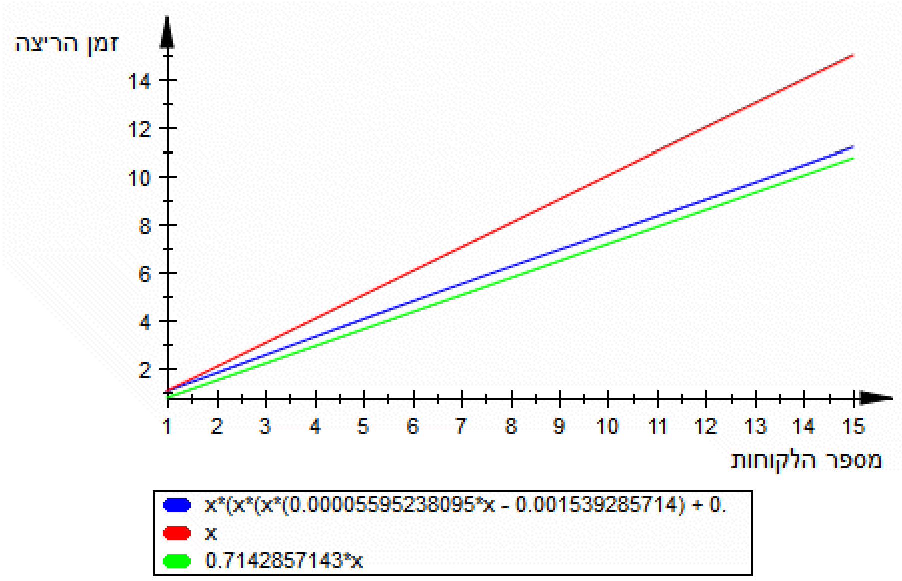
תרשים 18 - פונקציית הגידול בזמן הריצה כפונקציה של מספר הלקוחות כפי שחושבה בעזרת אינטרפולציה כאשר מספר המועמדיים לאיחוד הוא 15.

10. נספח ג' - הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד - אינטרפולציה

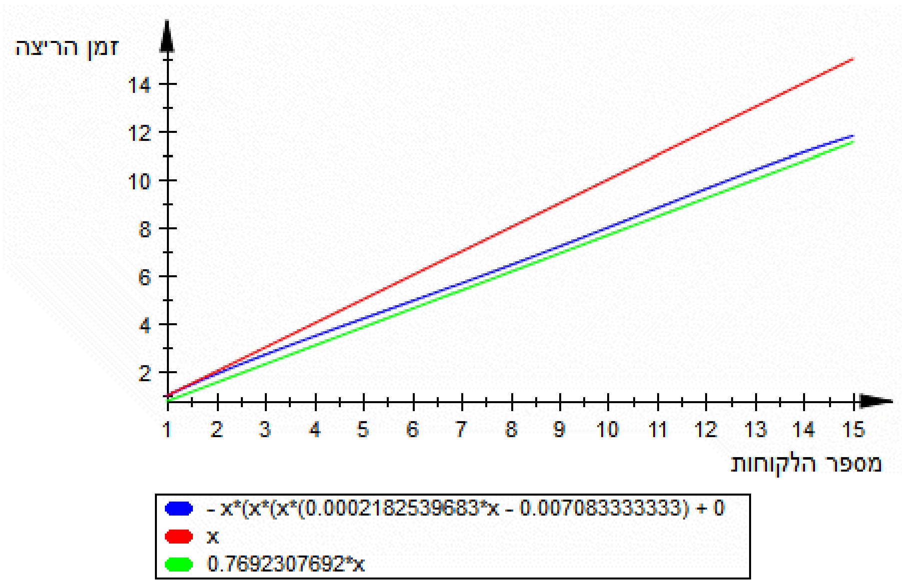
הגרפים הבאים מתארים את הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד כפי שחושבו בעזרת אינטרפולציה מתמטית. בכל גרף פונקצית הגידול מצוינת ע"י העקום האמצעי, אשר מעליו מסורטט העקום $y = x$, ומתחתיו משורטט עקום בעל המבנה $y = \frac{x}{a}$, אשר חוסם אותו מלמטה.



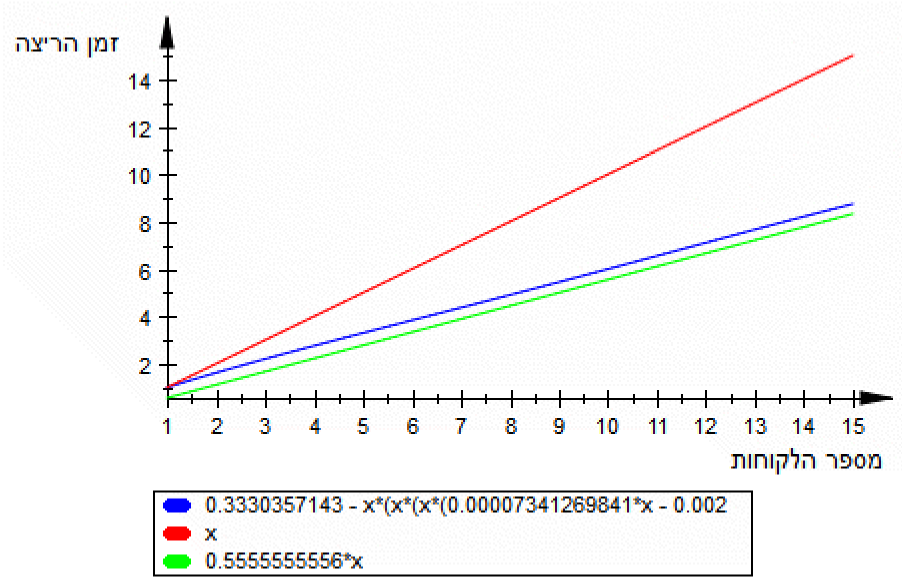
תרשים 19 - פונקציית הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד אשר החיסכון שלהם מחושב ע"י סימולציה, עבור 50 לקוחות.



תרשים 20 - פונקציית הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד אשר החיסכון שלהם מחושב ע"י סימולציה, עבור 75 לקוחות.



תרשים 21 - פונקציית הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד אשר החיסכון שלהם מחושב ע"י סימולציה, עבור 100 לקוחות.



תרשים 22 - פונקציית הגידול בזמן הריצה כפונקציה של מספר המועמדים לאיחוד אשר החיסכון שלהם מחושב ע״י סימולציה, עבור 150 לקוחות.

11. נספח ג' – קלט עבור אלגוריתם STDVRP

נתוני בעיה מס' 5 בקבוצת הבעיות מס' 17 עבור אלגוריתם STDVRP.

Truck capacity: 150

Customers demands:
Customer no. 1: 25
Customer no. 2: 9
Customer no. 3: 47
Customer no. 4: 17
Customer no. 5: 42
Customer no. 6: 57
Customer no. 7: 56

Adj. matrix:

	000	001	002	003	004	005	006	007
000	---	858	505	869	888	929	556	637
001	793	---	608	654	552	783	799	525
002	546	514	---	890	660	668	800	924
003	574	623	945	---	531	969	901	869
004	672	760	898	908	---	881	862	986
005	943	854	586	658	714	---	834	521
006	575	668	520	811	589	568	---	850
007	736	791	565	842	720	999	851	---

Edges information:

From 0 to 1		From 0 to 2		From 0 to 3	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
68	0.085700000	55	0.013800000	68	0.041900000
82	0.014811660	53	0.052268600	59	0.024144120
64	0.073218351	111	0.036423325	114	0.088632413
84	0.014046590	65	0.052055468	103	0.036771571
76	0.051251296	86	0.053094424	63	0.042772395
88	0.000152194	85	0.030664262	110	0.000306312
85	0.069158530	100	0.025135899	113	0.027633582
79	0.057200396	68	0.063417646	88	0.017117879
55	0.060717916	70	0.029012350	94	0.038558612
50	0.024211957	61	0.001867971	77	0.031243071
102	0.013738278	81	0.022350650	118	0.027924470
116	0.052722015	106	0.055729856	60	0.014328898
101	0.003478110	83	0.002369554	51	0.027146534
66	0.028343929	89	0.002528145	112	0.027447751
97	0.042236886	80	0.037863381	114	0.008311086
80	0.024458911	78	0.034517903	115	0.052283933
67	0.013267078	60	0.009007660	97	0.024180391
77	0.010655905	119	0.038613747	56	0.029988077
51	0.007140474	76	0.019020788	118	0.027281083
90	0.032803828	76	0.000420258	79	0.024639264
52	0.027771381	99	0.019228586	97	0.032269467
94	0.025981500	67	0.023395596	52	0.011789954
109	0.025225151	53	0.014409572	61	0.028187322
111	0.005172544	74	0.029931360	60	0.029308189
109	0.011046190	117	0.015278871	83	0.000343000
63	0.013529336	116	0.029568013	80	0.002483768
55	0.006274004	93	0.028312967	94	0.007952493
116	0.000349666	92	0.018855175	60	0.022857018
113	0.009815057	73	0.003709213	69	0.022924739
79	0.002698188	53	0.019730777	66	0.011028012
90	0.017276912	107	0.003957007	54	0.006416391
84	0.007741568	60	0.018165529	86	0.000953227
63	0.016025301	65	0.016385288	91	0.019063098
102	0.002504352	66	0.009679040	72	0.008056099
86	0.002000279	73	0.007716939	69	0.016721776

106	0.012974863	84	0.013954825	67	0.000183737
76	0.001128115	54	0.002198634	57	0.013231226
76	0.011798976	72	0.014318031	104	0.002212114
93	0.000728234	57	0.002909148	77	0.012339665
59	0.009603269	109	0.001242895	54	0.002725718
63	0.000188769	98	0.009351841	70	0.013061520
102	0.002383319	56	0.011413018	119	0.011193826
102	0.009057138	100	0.004977299	99	0.009527327
85	0.009463983	53	0.001729639	72	0.004112611
77	0.003373035	90	0.002674367	74	0.002510380
76	0.004700999	83	0.002138053	75	0.000690738
104	0.003626999	76	0.005108747	77	0.008943519
74	0.001126751	62	0.004752327	87	0.003374393
76	0.001465277	71	0.003322018	72	0.008226381
98	0.006801343	74	0.001701748	101	0.001957162
63	0.004881692	60	0.006449477	84	0.005324838
98	0.004554771	84	0.003678260	95	0.003815907
96	0.001853802	118	0.000626345	103	0.002722206
65	0.000005756	111	0.003241473	118	0.005144422
113	0.000368366	96	0.004042151	85	0.000011147
77	0.004861049	93	0.000678523	80	0.002546640
74	0.003260020	71	0.005826012	109	0.000207396
115	0.002787048	59	0.000496614	63	0.000540306
60	0.001258835	56	0.000978821	70	0.003182551
98	0.001629791	59	0.004076127	73	0.003649297
80	0.003475705	105	0.000307692	90	0.004195106
103	0.000574796	117	0.000330445	55	0.000082808
102	0.001034683	118	0.001303125	88	0.000739647
57	0.003574032	83	0.002474931	93	0.000710175
101	0.000826642	117	0.003544589	67	0.001527069
85	0.001457967	97	0.002970858	107	0.001533767
75	0.000743065	110	0.002488162	55	0.002948513
92	0.001553554	116	0.003189924	98	0.001899652
72	0.000099500	67	0.001931557	71	0.001374376
88	0.001298252	119	0.001964667	60	0.000862578
92	0.001475074	91	0.001674506	111	0.001408979
52	0.000905656	85	0.002656902	112	0.000974081
118	0.001389865	95	0.000909809	107	0.002488143
63	0.000392237	114	0.001017315	65	0.000867408
85	0.001908265	77	0.000214915	102	0.001043420
119	0.000244974	80	0.000294429	56	0.001628971
84	0.000740447	68	0.001334532	96	0.002127166
82	0.000848367	78	0.000182869	97	0.001110934
115	0.001150864	109	0.000881387	95	0.001149645
116	0.000785955	86	0.001410831	51	0.000985125
64	0.000550344	92	0.000654615	106	0.001270531
68	0.001241259	85	0.001577874	91	0.000183386
88	0.001653847	115	0.001481117	62	0.000056501
83	0.000185601	67	0.000977346	76	0.000064939
51	0.000687650	50	0.000360188	73	0.000899315
67	0.000173662	108	0.000686804	105	0.001168913
100	0.000383567	62	0.000636478	82	0.000065178
102	0.001303967	61	0.000267738	110	0.000135782
103	0.000557674	89	0.001175936	51	0.000885491
61	0.000280016	79	0.000656446	64	0.000709680
54	0.001124038	74	0.000058405	116	0.000288797
80	0.001037203	58	0.000872994	79	0.000838479
67	0.000388360	74	0.000499871	70	0.000108743
115	0.000035511	81	0.000231513	101	0.000071346
104	0.000198292	66	0.000723004	58	0.000107574
88	0.000699738	52	0.000633163	113	0.000605178
55	0.000131218	54	0.000205180	78	0.000480057
65	0.000265795	51	0.000167034	83	0.000064812
67	0.000290122	54	0.000414744	111	0.000611127
104	0.007724304	52	0.006263896	61	0.007325584

From 0 to 4		From 0 to 5		From 0 to 6	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
62	0.000700000	98	0.014500000	114	0.015100000

58	0.063555480	76	0.057454650	102	0.061260780
65	0.023487187	66	0.045195809	105	0.062899831
92	0.012315474	72	0.041052504	69	0.066879451
89	0.049766785	74	0.062629700	106	0.005715792
114	0.036727563	115	0.000467500	79	0.042717413
77	0.051165848	85	0.038078422	91	0.015952132
87	0.028585562	106	0.016515858	60	0.019404024
103	0.061190255	95	0.065314321	112	0.043172291
54	0.019973424	94	0.039395716	106	0.042748180
71	0.060032983	93	0.008485719	104	0.026526380
65	0.042719210	80	0.013073470	114	0.019422771
59	0.025399847	76	0.026364582	57	0.037120501
112	0.030151872	107	0.041088819	103	0.018288519
104	0.011169564	75	0.002651915	88	0.031576633
67	0.027244525	54	0.023958988	66	0.014539973
79	0.004786051	111	0.038337051	59	0.033891616
104	0.012132663	68	0.031556491	103	0.037858007
70	0.008163460	56	0.043214297	66	0.031179279
54	0.037258339	58	0.037933493	88	0.005568822
59	0.037065242	108	0.022645311	64	0.031663274
89	0.020065808	53	0.024921447	75	0.023589655
118	0.007870423	81	0.004119713	118	0.021372755
51	0.030777867	86	0.015473673	62	0.027085173
116	0.021017236	58	0.006568123	54	0.010234863
58	0.025869330	96	0.013057314	72	0.014287832
57	0.012013703	71	0.006941168	66	0.011325359
65	0.006208652	52	0.014245217	101	0.011316625
104	0.009419719	99	0.004160898	65	0.009104957
85	0.017005244	63	0.019055348	85	0.003122957
74	0.000659714	96	0.013469783	113	0.019379508
74	0.006987033	56	0.006304603	66	0.003751032
70	0.007166353	98	0.009805930	108	0.002237906
50	0.005472541	59	0.014320378	78	0.015921926
109	0.004851339	65	0.002398164	81	0.004880757
55	0.010535580	58	0.011425884	51	0.002049849
103	0.003324518	104	0.010795589	54	0.010226827
61	0.006068039	103	0.014873753	79	0.013533612
75	0.010664527	55	0.009380276	53	0.002994587
101	0.013523751	103	0.003219203	74	0.005568201
81	0.013088337	114	0.003904571	82	0.010472969
93	0.000953406	91	0.001727350	69	0.001802100
57	0.003943984	76	0.007075103	69	0.007565979
81	0.011261872	60	0.006883356	53	0.006972281
111	0.005512176	101	0.007618419	118	0.002618805
84	0.009111558	117	0.009086627	94	0.007788511
97	0.002828293	105	0.002972141	68	0.003483641
85	0.000938160	71	0.001915446	66	0.002246528
91	0.001955002	88	0.006344316	93	0.001370529
68	0.005413499	54	0.007505677	110	0.006833773
66	0.005986963	50	0.005436777	69	0.002120533
55	0.005215294	70	0.006455850	83	0.005422452
94	0.003782383	57	0.000082073	75	0.001257388
61	0.006116155	98	0.003758346	85	0.003809893
72	0.005234898	84	0.000312262	53	0.004921930
112	0.000666784	82	0.000168859	80	0.005578679
96	0.003315871	78	0.005147806	54	0.000427244
72	0.002068464	110	0.001730219	97	0.000353841
78	0.001366592	51	0.004519481	96	0.000376070
66	0.004704020	114	0.000411881	114	0.000790620
113	0.000251973	73	0.002222350	54	0.003560903
78	0.001485791	51	0.002017528	113	0.000823497
104	0.000771558	61	0.000780351	96	0.000918174
113	0.001903443	109	0.001663763	110	0.002843200
98	0.000226379	74	0.001955137	78	0.000642727
104	0.000558784	79	0.001117157	70	0.003506106
77	0.000462772	92	0.002631981	75	0.001790272
96	0.002697698	67	0.000454572	60	0.000609317
111	0.003294498	51	0.001778971	75	0.001741619
81	0.001821606	52	0.001031210	105	0.000299675
60	0.001624324	112	0.001371490	62	0.001459028
94	0.000056690	58	0.001719080	93	0.000503032
92	0.001041001	67	0.002225260	107	0.001868936

66	0.000215252	86	0.001827346	102	0.000485620
119	0.000291944	82	0.001884923	100	0.000458037
98	0.000842306	64	0.001057988	92	0.001479694
119	0.000748440	108	0.001630092	67	0.001785508
100	0.000432565	94	0.000721375	104	0.001278268
72	0.001819139	103	0.001140622	76	0.000656746
113	0.001018935	107	0.001189527	56	0.000324087
94	0.001533682	61	0.000298170	72	0.001789794
65	0.000530999	70	0.001114948	62	0.001723088
56	0.001495942	94	0.000147088	86	0.000980841
100	0.000994660	79	0.000630686	94	0.000960275
115	0.001044589	98	0.000710426	115	0.000799301
86	0.000758583	73	0.000380568	99	0.000222421
107	0.001022770	113	0.000482558	69	0.001218535
107	0.000526250	63	0.000682852	119	0.000826119
79	0.000539281	78	0.000258465	57	0.000948169
101	0.000545328	67	0.000168679	74	0.000041346
116	0.001059262	97	0.000004318	82	0.000819321
83	0.000684791	118	0.000282680	116	0.000270527
75	0.000510319	73	0.000573547	89	0.000794741
61	0.000114826	84	0.000403645	94	0.000644649
78	0.000524808	62	0.000458028	100	0.000560102
77	0.000139893	77	0.000123187	105	0.000059041
51	0.000003941	104	0.000076531	79	0.000558801
115	0.000445155	64	0.000343947	104	0.000266940
71	0.000347926	65	0.000083832	70	0.000575472
115	0.009055477	53	0.004847485	95	0.007222255

From 0 to 7		From 1 to 0		From 1 to 2	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
62	0.023900000	66	0.058600000	110	0.088100000
64	0.044998210	95	0.093951720	100	0.018876330
81	0.025605299	105	0.013135448	103	0.049920023
79	0.064833549	96	0.059987093	114	0.022257936
62	0.035980374	90	0.004645954	80	0.023065764
94	0.075559693	85	0.046411691	112	0.059594162
54	0.050382391	87	0.049326884	90	0.063188703
93	0.060747273	85	0.034236213	92	0.046507299
115	0.020146579	82	0.016376448	79	0.040537591
103	0.054702967	94	0.033597409	110	0.038687254
119	0.046547412	101	0.022822595	98	0.038009134
73	0.035059695	117	0.041384324	59	0.019120967
94	0.033415247	107	0.000210210	103	0.013632135
93	0.002140607	69	0.021853063	70	0.041342633
96	0.013716579	112	0.008760220	82	0.008349757
76	0.003792830	104	0.027356950	78	0.029716555
55	0.001838121	113	0.035611596	105	0.017839491
69	0.018583136	79	0.040496479	82	0.010789496
101	0.030500733	63	0.024726096	119	0.011150990
57	0.016769062	101	0.035881290	71	0.008839119
64	0.031862953	61	0.009026153	66	0.022991138
87	0.005498728	54	0.001640171	67	0.010413976
61	0.003914099	58	0.013662377	100	0.001268278
75	0.028662577	51	0.028271454	111	0.010831984
60	0.011781622	64	0.025467379	55	0.013510139
74	0.010699189	54	0.000858707	63	0.007548792
111	0.008940999	97	0.001208170	70	0.015870589
50	0.007685384	72	0.021317031	69	0.004503068
113	0.011679428	93	0.020969684	102	0.009302845
107	0.016042029	62	0.016427547	81	0.012508306
95	0.007732102	111	0.004775313	68	0.001377836
54	0.016723153	107	0.007573675	105	0.005936588
53	0.009785910	69	0.017027969	110	0.004664781
110	0.000254658	98	0.012569968	116	0.003078601
78	0.002847893	68	0.008555448	77	0.007321368
61	0.002400041	72	0.011824907	98	0.002610222
65	0.004960938	86	0.008530911	73	0.013394294
99	0.000844335	76	0.005610755	79	0.000671028
72	0.002424502	72	0.010458680	81	0.016477138

88	0.007973629	92	0.004120684	96	0.013443184
75	0.009935234	118	0.003596109	77	0.016394023
113	0.003080318	64	0.009480398	105	0.000437798
117	0.013005365	57	0.006574112	119	0.002416740
64	0.008787251	110	0.006948617	77	0.003668696
53	0.010872759	80	0.004240356	62	0.004465031
88	0.002354858	71	0.001949980	87	0.009332640
117	0.003761139	81	0.006393320	109	0.011018882
107	0.007990339	57	0.005650147	118	0.002225297
111	0.006064754	81	0.003706057	110	0.012242302
58	0.005072607	104	0.000923803	81	0.005704333
68	0.002699951	101	0.001256077	56	0.007852873
88	0.007302711	96	0.003185787	116	0.007303489
69	0.001430056	92	0.001048915	102	0.000735445
63	0.003804536	85	0.000114444	114	0.006066592
95	0.005868518	97	0.003940735	59	0.002014139
50	0.004236297	87	0.002999846	100	0.007569640
117	0.001610961	71	0.003039726	98	0.005285116
75	0.004888352	106	0.001841238	56	0.002156154
83	0.003782556	52	0.002876542	68	0.005038391
112	0.000369506	70	0.001628766	67	0.004336675
73	0.000571957	55	0.002464254	106	0.004010524
117	0.001829980	73	0.001843426	117	0.000493272
93	0.003607259	68	0.000092606	105	0.000270309
62	0.002965716	76	0.001456267	68	0.002751086
83	0.002815146	88	0.001065983	94	0.001209338
62	0.001852487	51	0.001470347	61	0.002153603
51	0.002549756	59	0.000892192	75	0.000656613
66	0.000024956	96	0.001481790	94	0.002898727
58	0.002288643	98	0.001058461	119	0.000147150
116	0.001702686	76	0.001556731	69	0.003808313
104	0.001321277	66	0.001348138	66	0.002704199
54	0.000188334	62	0.000834012	57	0.003290630
117	0.001538841	52	0.000815311	90	0.001915921
68	0.001170069	117	0.000865517	116	0.000410401
50	0.001417932	59	0.000714597	99	0.001970520
74	0.000064272	111	0.000676915	113	0.000275177
105	0.000056383	75	0.000265355	101	0.002025506
69	0.000198888	56	0.000927623	99	0.000966259
76	0.000143841	113	0.000929768	105	0.002104563
75	0.000508998	76	0.000484604	117	0.001565205
52	0.000497271	108	0.000746934	107	0.000800983
57	0.001076232	83	0.000163071	53	0.001781146
104	0.000473288	109	0.000629183	55	0.000491646
100	0.000040537	91	0.000579965	90	0.000978184
54	0.000441982	58	0.000039424	56	0.000759675
109	0.000660871	107	0.000279486	72	0.000375507
112	0.000150395	83	0.000227307	84	0.000543811
72	0.000773697	69	0.000127537	83	0.000145229
55	0.000893927	96	0.000407536	92	0.000046447
84	0.000770029	95	0.000326787	86	0.000288271
96	0.000448143	54	0.000326871	57	0.000186030
108	0.000511376	81	0.000407307	107	0.001411492
86	0.000102503	90	0.000162437	73	0.000092659
108	0.000742292	69	0.000324673	99	0.000811162
95	0.000074903	73	0.000259156	80	0.000308526
94	0.000020019	108	0.000219945	110	0.000165734
88	0.000403179	63	0.000231766	54	0.000677001
100	0.000265622	53	0.000083732	56	0.000939010
87	0.000172947	76	0.000244482	96	0.000218224
72	0.005811370	100	0.002304864	109	0.009838198

From 1 to 3		From 1 to 4		From 1 to 5	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
77	0.004700000	86	0.062900000	104	0.037800000
68	0.034138790	50	0.001967910	67	0.051573920
97	0.058726950	99	0.015149140	80	0.069116519
77	0.023553534	91	0.067066757	56	0.082383786
92	0.067937480	50	0.022175821	82	0.075305277

102	0.027815353	69	0.048515238	51	0.042123343
103	0.046596110	70	0.060622448	69	0.058586950
98	0.041614046	114	0.057078772	80	0.014694377
54	0.038289967	99	0.019138289	114	0.013698821
50	0.036705492	64	0.061053480	101	0.050812078
116	0.007191098	104	0.027989510	51	0.001914839
108	0.036396232	72	0.028262206	115	0.018322638
62	0.009394260	102	0.008185247	85	0.009770083
113	0.014570376	76	0.011437694	102	0.022273176
93	0.023696686	105	0.038846152	66	0.035497662
77	0.009251788	65	0.044566116	91	0.003869977
64	0.008830171	115	0.013941483	116	0.008080228
99	0.004084733	111	0.026516191	52	0.026109791
89	0.032061889	66	0.030497792	115	0.034971155
96	0.021824472	81	0.009702059	90	0.030329632
72	0.016565913	105	0.003030612	118	0.001282340
107	0.020974229	99	0.016214461	102	0.002086939
57	0.033081910	98	0.020223871	58	0.009931627
70	0.037588654	101	0.003201647	106	0.014374313
77	0.009953445	88	0.003680949	65	0.026570438
91	0.022274798	104	0.006497188	54	0.007057599
109	0.023382404	102	0.004635470	95	0.016345062
55	0.008201898	93	0.001233685	108	0.017375178
64	0.018687782	71	0.014454892	80	0.019400835
61	0.013435959	69	0.014998185	52	0.004680858
99	0.003404088	104	0.009531263	50	0.012045687
94	0.004558293	115	0.005920451	102	0.012277166
76	0.013997752	52	0.007945246	78	0.015646804
59	0.013636110	95	0.022606800	96	0.001321742
112	0.013070669	113	0.005864942	111	0.012113349
50	0.008691590	89	0.018718280	72	0.013983505
112	0.010377549	108	0.017245004	96	0.010808910
66	0.010500851	61	0.011753255	115	0.011269228
70	0.006026378	103	0.013313677	70	0.009044054
84	0.006289255	106	0.001805804	67	0.001750762
110	0.014560321	78	0.003863278	78	0.002503103
110	0.004773912	97	0.009305054	67	0.002735977
108	0.010033685	119	0.000025669	100	0.002803497
72	0.005887733	63	0.009931814	80	0.002560703
56	0.005151947	99	0.009837893	75	0.002425925
110	0.010482200	73	0.007858897	64	0.000321481
82	0.001584062	86	0.009021771	112	0.002777689
102	0.001908594	84	0.003822540	111	0.002750847
100	0.006574701	60	0.004163858	70	0.000558901
56	0.005022730	80	0.003071101	96	0.001434848
96	0.006408299	117	0.006215042	95	0.005192889
56	0.006688670	64	0.001971470	91	0.002834740
69	0.001293045	89	0.006242916	92	0.000399892
104	0.002411841	76	0.005466527	92	0.005121499
85	0.003719396	50	0.005616062	99	0.005060258
119	0.006120686	69	0.002495945	81	0.003089450
62	0.005093340	98	0.000778512	95	0.004310185
95	0.004708100	59	0.002648189	108	0.003456248
69	0.001703782	98	0.001258892	75	0.001916224
74	0.003539645	67	0.000186875	116	0.000349732
68	0.003402220	59	0.004181122	84	0.001493107
85	0.002206736	68	0.000309195	100	0.002314897
102	0.000508958	115	0.003619137	91	0.001043041
117	0.003270511	77	0.003728266	57	0.001117865
105	0.000253369	80	0.002659596	118	0.002026894
109	0.002684491	87	0.000967063	97	0.000695349
67	0.003697987	79	0.001432581	64	0.000166941
100	0.000568219	60	0.001973296	87	0.002184503
108	0.000265929	54	0.001186123	96	0.002472447
53	0.002227509	58	0.000345917	98	0.001814135
116	0.000034285	93	0.001005649	99	0.001624736
116	0.001864935	104	0.002050465	91	0.001196836
80	0.000029269	88	0.002155291	108	0.001428869
100	0.002014632	51	0.000822715	77	0.000097592
84	0.002649023	102	0.002082484	85	0.000272423
113	0.001032204	100	0.001727052	119	0.001769794
69	0.001035940	65	0.000692357	58	0.001106399

67	0.002217052	66	0.000159518	99	0.000519317
63	0.001262106	85	0.000221204	119	0.000017450
52	0.000323493	73	0.001184851	108	0.000573535
63	0.000065469	82	0.000666891	117	0.000512518
86	0.000829481	76	0.000379988	107	0.000212506
94	0.000582405	76	0.001398044	62	0.000388832
92	0.000341117	52	0.000322079	75	0.000935270
54	0.000817332	119	0.000622941	72	0.000225112
116	0.000316573	80	0.000336139	55	0.000783209
68	0.001083814	59	0.000148347	102	0.000391968
99	0.000369666	79	0.000871787	76	0.001138485
100	0.001299837	57	0.000511405	104	0.000272421
52	0.000956785	86	0.000047847	102	0.001151886
77	0.000331183	106	0.000024801	112	0.000062912
67	0.000952172	63	0.000457175	105	0.000920955
61	0.000469121	51	0.000606888	91	0.000820607
76	0.000839708	54	0.000017663	55	0.000565153
100	0.000693691	64	0.000052002	74	0.000203455
118	0.000180397	59	0.000820107	98	0.000545246
52	0.000753896	55	0.000312110	50	0.000279775
82	0.000714226	96	0.000140385	99	0.000297953
84	0.000359696	118	0.000384197	107	0.000327510
117	0.006748924	77	0.007105030	72	0.006823362

From 1 to 6 / Time interval no. 0 — From 1 to 7 / Time interval no. 0 — From 2 to 0 / Time interval no. 0

Speed	Probability	Speed	Probability	Speed	Probability
105	0.072600000	71	0.064500000	71	0.075000000
75	0.005100700	95	0.044623350	51	0.045510000
81	0.008761843	86	0.018530234	110	0.022866740
108	0.018179395	105	0.071532406	102	0.053795941
109	0.045663261	97	0.052052911	72	0.078115098
103	0.022177034	117	0.051140383	91	0.069137546
110	0.036410782	109	0.065785634	61	0.031533142
113	0.039950903	66	0.058697479	107	0.009298219
117	0.040186850	52	0.008138554	67	0.002213076
74	0.011091120	54	0.033278444	119	0.046307286
102	0.017426965	79	0.025203557	112	0.011494326
55	0.031392753	53	0.030593630	81	0.024130695
79	0.034831624	62	0.037169619	73	0.039635665
77	0.035494662	113	0.019173541	98	0.004762334
118	0.001567977	80	0.028531458	52	0.045119354
72	0.051893106	95	0.004849005	64	0.021480624
60	0.036856245	51	0.022978888	79	0.016322438
56	0.016232729	53	0.029711470	68	0.024236979
80	0.037934564	105	0.009538370	53	0.004093638
60	0.032195065	115	0.026533230	117	0.027221145
67	0.017253038	71	0.007227739	88	0.020654910
84	0.028197675	100	0.005775181	118	0.007947822
114	0.014272348	85	0.013197780	110	0.011073569
59	0.031712734	72	0.000461103	101	0.027970890
106	0.027979188	86	0.015271768	53	0.017000769
110	0.026243572	98	0.011139986	93	0.012759273
118	0.011472688	107	0.011289630	109	0.008410702
86	0.013630049	83	0.019275274	82	0.002419078
95	0.021019531	92	0.000919337	71	0.014129836
94	0.019401624	70	0.015646683	61	0.002546556
99	0.005149628	70	0.007060954	90	0.014638771
65	0.015562017	77	0.018903137	91	0.008181222
63	0.015029422	119	0.014369491	103	0.002219915
57	0.002686904	96	0.004063704	87	0.015347141
74	0.008803194	112	0.006403831	65	0.004068084
54	0.002344785	59	0.011055634	56	0.002586180
77	0.008683618	64	0.003343802	55	0.004622778
113	0.004267150	69	0.002376590	64	0.005031759
103	0.004562014	102	0.010968916	65	0.003405388
87	0.000213828	56	0.008509879	89	0.016010773
78	0.005914225	81	0.009860879	51	0.011442626
103	0.001040983	113	0.008717507	53	0.005829607
61	0.004720766	118	0.001996859	71	0.009344709

114	0.008040736	71	0.005008761	94	0.000996692
62	0.005684183	58	0.001920270	114	0.001988748
97	0.005378942	65	0.006233555	70	0.005456765
61	0.002180114	119	0.001544080	116	0.009467173
52	0.009038489	86	0.007130052	60	0.009073053
64	0.005515552	109	0.004940102	50	0.005502267
87	0.000382462	64	0.004177886	83	0.006596182
70	0.007471947	106	0.000404668	106	0.007735740
89	0.001663717	58	0.000629022	98	0.006015202
107	0.004804324	80	0.005415695	86	0.003234801
87	0.002389912	66	0.002996170	78	0.003776990
87	0.001232955	62	0.001476057	58	0.001024076
105	0.001785211	55	0.003603302	111	0.000586235
75	0.003627684	115	0.002550284	51	0.001037742
85	0.001263459	69	0.001662891	56	0.004555506
84	0.000699956	81	0.002063315	92	0.000848580
96	0.004176358	50	0.002948208	56	0.000234844
117	0.001971351	87	0.000359448	83	0.002353462
119	0.003796603	60	0.002282989	109	0.000243037
111	0.002267737	64	0.001225706	114	0.003089704
50	0.000652368	103	0.000009309	63	0.001050252
89	0.003727600	100	0.001014367	57	0.001455966
57	0.000361398	72	0.000063013	70	0.002739400
106	0.001753141	95	0.001299527	100	0.003355516
101	0.001497111	114	0.001105639	62	0.000229759
113	0.001733749	103	0.002208536	56	0.000864765
68	0.001428859	80	0.000113982	114	0.002998817
110	0.001588675	118	0.001838198	71	0.000151075
62	0.002719353	76	0.001907575	116	0.002523289
102	0.000972231	60	0.000991894	69	0.000621045
62	0.001295787	106	0.000845728	81	0.002331307
100	0.001836832	114	0.000243436	102	0.001014418
83	0.000167621	85	0.000655332	69	0.002275668
119	0.001552641	100	0.000790540	85	0.001065110
56	0.001694373	58	0.001254190	69	0.000846889
118	0.001322366	114	0.000777682	57	0.001464670
74	0.000484848	113	0.000081145	112	0.001891672
99	0.000585186	62	0.000364081	69	0.000356720
106	0.000902678	109	0.001323850	83	0.001530033
56	0.000987035	118	0.000082022	111	0.000446256
84	0.001157240	57	0.000714231	75	0.000342785
112	0.000087110	96	0.001194416	50	0.000061831
73	0.001156712	94	0.000783741	99	0.000624351
111	0.000218182	55	0.001096942	67	0.000432675
114	0.000442485	105	0.000828816	101	0.000672029
66	0.000010194	100	0.000309682	84	0.000144727
55	0.000506143	85	0.000675756	103	0.001068274
74	0.000448084	89	0.000297654	62	0.000588043
80	0.000704230	55	0.000488690	83	0.000977384
81	0.000023871	107	0.000756236	52	0.000539099
57	0.000054411	71	0.000154171	58	0.000344602
77	0.000716326	71	0.000146002	87	0.000915289
82	0.000113644	82	0.000199723	116	0.000458190
62	0.000730496	112	0.000250131	73	0.000025700
103	0.000339518	110	0.000465959	87	0.000127181
101	0.000378431	111	0.000296271	89	0.000148897
112	0.006168820	108	0.005401246	111	0.009582913

From 2 to 1		From 2 to 3		From 2 to 4	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
67	0.076600000	82	0.051900000	118	0.052300000
99	0.060852060	81	0.059919920	84	0.016489980
113	0.051839131	74	0.045652456	66	0.049912857
66	0.056749617	117	0.058639923	94	0.008019804
109	0.065066678	73	0.021635301	64	0.063487264
50	0.060691430	69	0.042533684	92	0.042433001
62	0.044602277	95	0.060672288	92	0.006138857
109	0.041318796	99	0.062938934	71	0.036614597
101	0.041538649	82	0.058835810	73	0.066880916

80	0.006509638	98	0.019395508	68	0.036372067
55	0.032569871	110	0.041171156	117	0.054057507
63	0.003693295	68	0.035037819	97	0.019628343
68	0.001969265	107	0.005476673	58	0.046277676
65	0.033379148	100	0.042746672	55	0.034846406
92	0.036683429	70	0.014832833	51	0.040169156
74	0.008606389	68	0.014652247	57	0.002814052
101	0.033657865	73	0.030754517	78	0.030538497
115	0.022785484	77	0.005997677	100	0.007821078
103	0.002791717	73	0.015345989	88	0.025654183
59	0.026815431	63	0.020115008	95	0.001797719
112	0.017360278	75	0.025411041	84	0.014238292
98	0.026022358	109	0.006445296	67	0.023770736
116	0.013931822	72	0.001169502	74	0.017201851
61	0.007393306	53	0.011021461	52	0.007654140
97	0.009516027	96	0.009982241	71	0.021025017
107	0.008400069	75	0.004088716	58	0.020977370
80	0.018779037	78	0.009812348	116	0.019446367
103	0.011126788	77	0.015040367	67	0.002544412
90	0.011332759	68	0.014906508	109	0.003509495
77	0.014498346	106	0.004497740	53	0.009026921
76	0.007630660	105	0.018482548	82	0.020394024
56	0.014194675	78	0.016490675	55	0.006888918
86	0.010277747	51	0.013818545	93	0.010298592
60	0.003394928	65	0.011358751	58	0.015925829
104	0.002254484	81	0.006758198	75	0.001846254
116	0.007485826	93	0.012148196	115	0.010806756
94	0.006385467	106	0.000529505	69	0.003348203
66	0.004831784	119	0.010956438	54	0.008498927
117	0.002324770	116	0.000158124	93	0.013753706
59	0.006580295	78	0.007627144	119	0.007354892
111	0.007906524	55	0.002685746	106	0.009824992
53	0.001465595	72	0.007837221	77	0.003293251
52	0.007474609	50	0.000716622	102	0.003650428
104	0.005890283	118	0.007341844	58	0.006547620
59	0.002255785	82	0.005238917	83	0.005812694
56	0.000800840	114	0.004154313	74	0.003550190
98	0.001636766	95	0.006016647	92	0.002137937
82	0.002381069	109	0.002202166	91	0.004702095
85	0.004152005	82	0.004865086	72	0.007214310
80	0.000884316	65	0.002159295	55	0.005908898
79	0.004627992	107	0.001123873	79	0.001032190
87	0.002365665	118	0.000663147	112	0.001245654
83	0.000809190	104	0.001703383	104	0.002913473
102	0.003790216	75	0.001609324	84	0.004451616
96	0.001155403	79	0.001661989	99	0.003827039
93	0.001338732	101	0.001014248	84	0.004621697
85	0.002695915	71	0.000164199	61	0.002298959
118	0.001720927	60	0.001986234	91	0.002929225
81	0.003119640	96	0.001966906	87	0.003353663
115	0.001753303	116	0.003330812	92	0.001093792
119	0.000921076	114	0.000974748	79	0.001851982
60	0.001218899	74	0.002289688	78	0.002806299
57	0.002229880	92	0.000017602	103	0.001848453
117	0.000926156	73	0.002480323	53	0.000109884
59	0.002014353	63	0.001642480	94	0.001626554
118	0.000537368	87	0.000141094	67	0.002180528
106	0.002073389	105	0.002442798	113	0.000497033
69	0.000011648	116	0.001910675	61	0.001824979
75	0.001078688	72	0.000364335	115	0.000534638
99	0.001583036	57	0.001309673	86	0.000748476
64	0.001573479	100	0.001166346	110	0.000981953
61	0.001295147	84	0.000125024	115	0.001919476
100	0.001043063	119	0.001190051	58	0.001048788
93	0.001269918	71	0.001017544	66	0.000287683
69	0.001133794	106	0.000551240	96	0.001967486
116	0.000612916	100	0.001418289	51	0.001016950
104	0.000848636	115	0.000406751	71	0.001407714
61	0.000090518	85	0.000693090	77	0.001526140
101	0.000098476	111	0.000316446	102	0.000188864
77	0.000572012	87	0.000939797	79	0.000342506
65	0.000312465	52	0.000586999	78	0.000639148

99	0.000653186	61	0.000858776	56	0.000272096
71	0.000274950	109	0.000656613	79	0.001025483
86	0.000374479	82	0.000302116	117	0.001198991
78	0.000643462	56	0.000269212	71	0.001023829
58	0.000531866	62	0.000412782	98	0.000334309
64	0.000038416	78	0.000249154	67	0.000298077
75	0.000032234	64	0.000240719	79	0.000021430
60	0.000155932	105	0.000213525	88	0.000555893
99	0.000182484	101	0.000691575	95	0.000249089
63	0.000485666	98	0.000485242	71	0.000775218
56	0.000068192	53	0.000523164	76	0.000518493
112	0.000173021	111	0.000334747	84	0.000505242
82	0.000404399	58	0.000262967	78	0.000069596
106	0.000071603	82	0.000214817	104	0.000143042
69	0.000096110	117	0.000030869	113	0.000522879
61	0.000021846	94	0.000162138	104	0.000170353
95	0.000303304	92	0.000091784	112	0.000234914
66	0.000114499	81	0.000317979	70	0.000405234
88	0.003263067	65	0.004297100	76	0.005145913

From 2 to 5 Time interval no. 0		From 2 to 6 Time interval no. 0		From 2 to 7 Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
67	0.082400000	91	0.086600000	115	0.085000000
100	0.023857600	115	0.069783760	74	0.085552500
97	0.026365401	101	0.062174517	94	0.054494701
119	0.033567490	58	0.027116028	84	0.048434550
76	0.033352380	66	0.057705916	63	0.000508563
60	0.038902216	76	0.039289356	102	0.066502487
59	0.070748451	109	0.030960263	118	0.036536699
84	0.005457371	61	0.008894456	72	0.056752613
79	0.048933925	67	0.005619029	57	0.028084407
100	0.023356437	87	0.004405368	74	0.032341822
110	0.012751622	104	0.054488382	75	0.039451749
56	0.001860952	67	0.049711367	80	0.033996179
112	0.033393295	57	0.025162578	63	0.032339311
117	0.023675715	87	0.035043922	90	0.031360346
50	0.039087430	70	0.004563364	114	0.014598305
107	0.010799229	59	0.032053012	59	0.001416183
71	0.037893916	82	0.023979292	89	0.023273553
54	0.004037009	74	0.032928892	68	0.028225812
95	0.018521854	106	0.003844725	112	0.009395263
106	0.017586338	56	0.014622085	108	0.025789370
79	0.029644463	119	0.008607396	57	0.016089708
78	0.031318643	106	0.011769290	81	0.016840286
102	0.011350122	69	0.028706555	56	0.014213951
94	0.012178632	63	0.025405537	95	0.021573842
78	0.010756976	114	0.017010253	65	0.008993588
111	0.000540944	108	0.007402239	96	0.000677643
96	0.030813174	75	0.012304078	76	0.004632647
95	0.001176078	101	0.015257475	88	0.002945075
100	0.019882795	93	0.000777445	78	0.011140691
75	0.003295790	96	0.004830378	102	0.014925293
73	0.015093391	64	0.007740440	55	0.010850857
76	0.016130503	53	0.002562651	98	0.011416348
90	0.021716239	94	0.012075517	111	0.009346842
62	0.007774439	56	0.004574055	74	0.003974712
110	0.008071167	58	0.005246927	70	0.005419244
80	0.000251820	73	0.002251577	51	0.008919484
113	0.006055179	50	0.001974382	96	0.004887313
116	0.002717315	55	0.008550524	84	0.007481904
106	0.005226549	55	0.013552613	73	0.007036121
69	0.010193166	93	0.002556269	94	0.004905642
114	0.010308176	71	0.008742739	56	0.004748594
51	0.010840786	106	0.008136787	89	0.003521513
109	0.011819579	68	0.009947726	101	0.000699762
88	0.004565897	75	0.010418444	95	0.002213052
58	0.000250286	109	0.006632993	57	0.000726010
75	0.012503677	66	0.007230092	81	0.001836444
52	0.008126032	81	0.007368412	89	0.006256664

55	0.009333532	65	0.004757312	106	0.002422698
73	0.007613701	64	0.005062191	113	0.003692603
66	0.007606088	101	0.004405768	106	0.002024457
59	0.002286850	108	0.006115349	67	0.001700576
84	0.000033604	104	0.003651161	98	0.003777267
112	0.007247109	72	0.003913295	80	0.004020580
57	0.002693176	71	0.001282744	76	0.003837720
51	0.003753599	81	0.001085034	102	0.003021340
66	0.003120516	92	0.003641870	107	0.002251207
79	0.005379626	78	0.002170644	118	0.000543245
64	0.004442108	115	0.002461458	105	0.001547796
89	0.003979371	90	0.001737148	59	0.000709156
106	0.004194117	111	0.003174004	54	0.002334588
77	0.004154540	69	0.000568214	53	0.000122272
67	0.003834989	78	0.001624617	64	0.000503537
52	0.001976478	54	0.001911387	50	0.002037249
117	0.000882003	97	0.000493810	65	0.001791489
66	0.000410004	112	0.001521197	96	0.000198344
106	0.000208495	72	0.001214638	109	0.001598714
53	0.001775653	60	0.002842865	117	0.000568591
79	0.002360206	74	0.000943774	72	0.001602026
58	0.000043633	109	0.000476974	112	0.001568719
105	0.003295031	71	0.001352282	52	0.001528864
98	0.000607525	86	0.000264651	100	0.001207184
71	0.001812597	84	0.001439980	88	0.000486507
104	0.002555279	53	0.001061165	69	0.001179169
87	0.000611043	75	0.001972093	70	0.000539877
83	0.000886992	114	0.001191547	61	0.000426839
69	0.001531984	51	0.001332530	68	0.000840251
116	0.000988777	106	0.001447534	90	0.000030133
79	0.000630558	74	0.000959775	72	0.000245058
50	0.001306063	86	0.000433676	58	0.000584950
117	0.000800714	88	0.000982467	102	0.000733888
112	0.001634833	51	0.000178916	95	0.000659701
58	0.001540890	74	0.001129062	67	0.000627363
99	0.000278785	112	0.000214519	115	0.000115822
77	0.001419689	91	0.000434939	97	0.000612651
76	0.000742257	58	0.000566313	108	0.000021546
91	0.000703091	100	0.000364875	90	0.000260239
113	0.001000706	58	0.000286540	63	0.000158582
56	0.000208941	70	0.000141792	53	0.000235234
87	0.000702818	96	0.000669058	84	0.000648624
54	0.000121087	78	0.000450296	113	0.000133114
116	0.000738228	64	0.000530393	81	0.000534200
102	0.000820818	66	0.000171938	88	0.000197018
107	0.000138161	82	0.000367290	107	0.000100770
57	0.000512507	92	0.000148344	57	0.000535486
53	0.000103100	94	0.000427223	115	0.000059284
95	0.000382773	108	0.000308097	95	0.000194151
53	0.000276950	94	0.000299060	54	0.000172049
56	0.000576303	87	0.000192798	64	0.000030743
55	0.000263666	103	0.000267958	84	0.000415382
113	0.006327983	101	0.004778328	76	0.004283508

From 3 to 0		From 3 to 1		From 3 to 2	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
90	0.056400000	75	0.031700000	80	0.005900000
59	0.070958720	55	0.016267440	106	0.068195260
115	0.042061310	95	0.076924231	94	0.037313961
76	0.080898489	114	0.061520116	93	0.058913569
50	0.054276939	77	0.016027688	87	0.023645800
66	0.042836920	77	0.025043401	67	0.044573537
88	0.014878542	77	0.009579212	86	0.046677368
108	0.006759504	83	0.058059575	78	0.063686943
114	0.053944479	93	0.042363188	89	0.027931914
103	0.001211669	55	0.037763364	77	0.030347972
101	0.044507286	56	0.006060092	85	0.039659235
100	0.020241240	100	0.041328605	101	0.012999129
63	0.013593262	85	0.044514694	66	0.005185491

109	0.049544191	99	0.039111072	72	0.036163960
119	0.036861137	102	0.030907956	72	0.036213306
92	0.020839034	92	0.029435948	119	0.017948591
102	0.002536217	115	0.033804687	98	0.026145065
111	0.009885102	103	0.004155723	104	0.000836998
79	0.021797096	79	0.028550263	82	0.019170681
111	0.024597448	96	0.035257432	75	0.014345684
101	0.032043616	75	0.021224020	69	0.016671916
115	0.011673784	51	0.020952087	84	0.024988206
79	0.022005532	58	0.004283848	91	0.021782072
85	0.003267476	104	0.010636668	104	0.021134350
76	0.025162338	97	0.013781340	63	0.029477589
89	0.000687934	65	0.006596908	73	0.024929436
92	0.003051246	55	0.017180570	115	0.009904543
78	0.018841795	76	0.006516671	107	0.017973667
108	0.016140754	92	0.009586853	72	0.012319989
88	0.013398543	96	0.019546672	81	0.011170525
94	0.006996719	56	0.003885470	55	0.018410358
119	0.003348311	116	0.015064230	74	0.014206014
58	0.012634668	91	0.013914829	70	0.013877328
110	0.007862757	76	0.010275764	109	0.013787237
74	0.008885142	75	0.005821001	108	0.005140224
97	0.005058904	73	0.013727490	79	0.001270884
118	0.001080402	115	0.001719023	54	0.000444854
69	0.010637286	64	0.006243181	64	0.000975254
109	0.004307906	55	0.000535742	62	0.000728950
117	0.005381597	87	0.004684786	94	0.007497128
102	0.008573029	119	0.004277782	72	0.007822504
93	0.000297896	53	0.003089845	83	0.003628836
68	0.006326942	78	0.004203667	77	0.000763226
112	0.006460936	88	0.002881085	81	0.003073021
96	0.000680721	78	0.005416594	103	0.000112384
79	0.006450554	85	0.000042232	98	0.009205365
51	0.001793281	54	0.008242436	50	0.007585818
53	0.003179568	110	0.001663736	69	0.002813707
118	0.006309006	98	0.002515090	56	0.005598365
68	0.005447344	56	0.003380100	77	0.005656291
113	0.000198141	103	0.001552426	104	0.005951943
103	0.005869620	50	0.002751315	107	0.005297776
69	0.001460793	61	0.002742362	89	0.001408726
53	0.003529928	72	0.007929922	110	0.005280042
109	0.003982692	82	0.006870404	63	0.003978298
117	0.002946385	64	0.003808582	90	0.001296030
105	0.000664774	111	0.001922417	115	0.004073729
114	0.005019025	112	0.004686726	65	0.001804658
103	0.004777522	59	0.003977024	82	0.000071577
114	0.002453535	75	0.001241070	79	0.000916313
108	0.001023841	64	0.000684519	78	0.002323077
107	0.001169149	64	0.004496802	62	0.003208208
61	0.000068493	110	0.001591034	98	0.001853985
72	0.003145324	108	0.000704955	56	0.000836532
51	0.003418426	67	0.000165671	102	0.002888146
105	0.000346675	103	0.003608978	119	0.001500323
54	0.000919386	116	0.000733925	96	0.002568402
87	0.002354878	91	0.003281798	107	0.001260574
116	0.002448002	73	0.002751733	109	0.001645966
102	0.000118632	56	0.000284142	103	0.000366196
86	0.000983433	52	0.002013222	65	0.001246575
104	0.000055622	104	0.001552117	56	0.000036411
57	0.002402591	69	0.002862215	79	0.000932255
111	0.002169779	59	0.002094442	118	0.000329241
69	0.000295094	111	0.000284791	91	0.001754508
92	0.000036658	52	0.000002514	110	0.000130399
106	0.001554251	113	0.002149511	68	0.000315476
110	0.001318203	60	0.001453030	72	0.000383491
64	0.000912209	117	0.000301531	96	0.001741388
94	0.000457753	81	0.001745633	58	0.000066340
112	0.000458042	83	0.000317700	60	0.000957891
85	0.001553092	99	0.000828277	60	0.000469718
109	0.001313519	60	0.000722785	103	0.000121561
80	0.000713354	92	0.001224730	91	0.000041121
91	0.000256984	99	0.000134458	51	0.001040602

117	0.001226220	79	0.000639129	89	0.001053080
64	0.001041061	91	0.000432777	67	0.000310759
51	0.000506094	63	0.000937280	50	0.001007966
82	0.000820364	74	0.001023412	82	0.000167329
62	0.000235837	91	0.001219827	78	0.000263972
59	0.000384722	64	0.000093681	88	0.000955368
85	0.000780126	68	0.000173984	95	0.000820946
72	0.000614831	50	0.000894797	51	0.000782859
70	0.000253885	113	0.000106310	66	0.000382919
78	0.000621429	51	0.001036568	90	0.000447051
86	0.000299883	109	0.000183426	107	0.000588371
62	0.000307776	70	0.000258989	87	0.000459275
60	0.000155101	92	0.000811408	50	0.000354416
109	0.000490392	112	0.000297484	77	0.000427638
61	0.005251899	73	0.008153759	68	0.005076066

From 3 to 4 — Time interval no. 0 | From 3 to 5 — Time interval no. 0 | From 3 to 6 — Time interval no. 0

Speed	Probability	Speed	Probability	Speed	Probability
54	0.040800000	84	0.021300000	109	0.052500000
63	0.067048080	83	0.038952260	64	0.064524750
114	0.069141774	84	0.069729282	104	0.034171142
96	0.017447815	118	0.073690563	100	0.042525086
66	0.039714223	62	0.066652645	79	0.063615415
71	0.058357626	91	0.026706114	112	0.074043562
93	0.026389395	55	0.018066307	117	0.063719490
107	0.055032968	106	0.007328460	62	0.040346867
89	0.047205536	102	0.033675446	106	0.054987529
54	0.005267650	86	0.060075769	56	0.028026139
95	0.007112577	99	0.049157909	93	0.034285649
83	0.044582161	64	0.043414818	86	0.003712211
99	0.010072674	96	0.037433282	87	0.013040139
97	0.029481265	80	0.018379594	70	0.007017183
97	0.012637472	112	0.016415996	98	0.020877802
72	0.006528952	115	0.034443572	86	0.007689794
102	0.041639867	117	0.026997374	114	0.019469420
102	0.020191764	82	0.028677965	72	0.006758061
119	0.018943635	87	0.008913262	52	0.019393081
53	0.034875296	62	0.013663547	118	0.028048523
60	0.019183616	54	0.019329160	61	0.017443775
56	0.015005396	103	0.009442191	88	0.012304077
99	0.019019754	83	0.006605797	81	0.016090817
79	0.020985052	110	0.012870063	90	0.015808504
76	0.007516725	98	0.010477992	60	0.014304014
85	0.005210047	71	0.001733204	117	0.002771856
113	0.025331164	54	0.000122934	111	0.006135885
110	0.021998448	102	0.016047115	78	0.003309449
115	0.008595146	119	0.015366755	50	0.002773649
119	0.015003331	101	0.015581836	60	0.011285000
57	0.001574349	71	0.011447930	56	0.012352792
86	0.018340359	116	0.014909148	106	0.019096154
61	0.002767184	112	0.013360357	68	0.013280110
51	0.012090706	78	0.005963676	55	0.008244015
93	0.005313344	105	0.000091841	94	0.000099629
107	0.001286514	90	0.007358138	94	0.004580177
98	0.002165299	71	0.001325121	119	0.004792637
115	0.001461428	111	0.001024477	76	0.001205632
52	0.000289363	55	0.012664900	60	0.001755681
55	0.001472799	73	0.005602877	94	0.015069563
104	0.013705956	112	0.011962531	100	0.009933658
64	0.005297745	99	0.005482333	50	0.001800555
102	0.001610902	54	0.009378836	87	0.005186550
66	0.009441918	102	0.009434773	72	0.008100157
52	0.009141881	116	0.007250209	56	0.002054781
88	0.006078041	106	0.007040878	62	0.010165976
83	0.003144098	87	0.003394955	76	0.005014502
106	0.000670941	58	0.003439101	99	0.005825458
51	0.001050871	97	0.006497965	101	0.005708222
70	0.005000675	80	0.004253864	59	0.006237959
91	0.006100447	108	0.002792065	83	0.000816576

56	0.000236860	98	0.003709395	115	0.001157735
81	0.003477442	108	0.004190236	114	0.005243169
84	0.005878295	102	0.004524972	58	0.004113978
52	0.005492738	59	0.002848737	110	0.002203683
112	0.006066389	89	0.000450074	118	0.005815869
72	0.005386640	72	0.001161987	83	0.000976238
117	0.000788465	108	0.001312716	104	0.002205389
111	0.000641318	110	0.001686111	88	0.003756545
65	0.000204089	101	0.000082053	77	0.004475917
114	0.001310837	51	0.001497277	77	0.001967375
98	0.004258924	79	0.003159837	76	0.002069457
105	0.001121650	64	0.002085019	108	0.003160594
92	0.001764817	115	0.002345129	112	0.002007443
73	0.003896592	118	0.000973250	91	0.002995098
55	0.003806441	77	0.000298171	74	0.001866472
101	0.002035263	81	0.001529362	118	0.003038422
85	0.000278941	91	0.000902071	85	0.001875594
111	0.000616529	68	0.000057554	104	0.002471460
116	0.001108116	95	0.000491080	65	0.002227602
94	0.001611983	58	0.001599119	70	0.001783758
101	0.000874739	108	0.000504136	105	0.001696158
113	0.001963151	67	0.001282253	63	0.000570409
68	0.001705363	87	0.000411960	80	0.000428470
75	0.001810918	93	0.001298229	100	0.000644626
63	0.002295738	93	0.000321278	53	0.000115588
111	0.000580697	65	0.000626313	66	0.000529216
93	0.000777214	97	0.000586035	74	0.000113485
50	0.000585512	54	0.001016555	100	0.000889520
66	0.000468422	83	0.000164024	72	0.000332181
107	0.000249643	112	0.001163618	105	0.001067343
79	0.000348533	50	0.000875134	119	0.000151065
98	0.001949261	70	0.000485901	118	0.000721372
62	0.001396104	96	0.000713633	114	0.000867180
76	0.001470226	99	0.000413083	95	0.000228007
68	0.001069820	55	0.000778955	92	0.000971195
61	0.001278988	107	0.000428083	81	0.000705173
78	0.000567528	113	0.000206537	89	0.000795516
85	0.000505595	92	0.000130233	93	0.000714085
113	0.000186377	100	0.000401332	67	0.000598661
117	0.000017414	62	0.000518977	70	0.000693197
50	0.000774359	66	0.000142322	104	0.000246804
97	0.000183903	103	0.000127301	100	0.000261544
52	0.000509366	112	0.000028920	69	0.000620974
115	0.000168068	118	0.000126951	89	0.000372207
54	0.000600377	73	0.000352930	112	0.000228296
115	0.000684868	88	0.000319571	88	0.000026269
84	0.000444836	73	0.000125228	54	0.000505890
54	0.000465615	103	0.000049449	102	0.000504307
91	0.007760807	118	0.005569724	104	0.006689807

From 3 to 7 Time interval no. 0		From 4 to 0 Time interval no. 0		From 4 to 1 Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
109	0.055700000	63	0.076600000	57	0.034800000
89	0.068556180	96	0.023085000	110	0.002026920
74	0.053683096	90	0.070674728	57	0.088419289
98	0.058284105	65	0.067283826	106	0.049598540
98	0.033988060	70	0.058244033	104	0.072613662
56	0.017150031	80	0.060201611	81	0.058547736
103	0.064137468	65	0.051384082	118	0.064263831
57	0.055511691	98	0.004858719	69	0.032431096
100	0.021703411	84	0.015749502	107	0.016306261
98	0.009940376	70	0.034543877	55	0.019637552
65	0.001122691	63	0.001988286	53	0.020770139
64	0.015013974	109	0.015258511	112	0.016055374
72	0.008396217	84	0.017632333	64	0.049200877
78	0.026035416	63	0.032712457	96	0.026048014
114	0.045714567	51	0.025838067	67	0.040839617
114	0.039297800	92	0.042663111	98	0.025200815
86	0.008557875	91	0.012841019	118	0.015904472

80	0.005298529	80	0.012391263	79	0.023178889
64	0.022037105	105	0.028429348	89	0.010427955
76	0.026121384	115	0.026314851	109	0.001768764
91	0.035793002	102	0.012787954	114	0.008962925
110	0.000098387	90	0.018634452	57	0.020445727
105	0.001704865	117	0.027828765	92	0.001210206
104	0.004337845	89	0.019103751	98	0.021244564
116	0.025745274	89	0.014892863	69	0.017702116
51	0.009918367	101	0.021140939	80	0.012148873
55	0.016167604	53	0.015332524	112	0.004604522
59	0.022543721	53	0.003046188	105	0.017956376
74	0.007274764	114	0.007145588	81	0.012955270
78	0.004755291	103	0.004353416	107	0.003564512
63	0.004967170	91	0.010055812	67	0.005426943
105	0.018458543	105	0.012356751	72	0.017816925
86	0.005320828	61	0.007282902	63	0.015691423
100	0.011531871	67	0.013231444	105	0.009593201
82	0.019103471	75	0.004962145	85	0.010603907
95	0.003362154	70	0.009453771	116	0.007875294
119	0.000500734	73	0.006427681	79	0.000475719
90	0.004166420	109	0.009027494	79	0.011135331
61	0.003494394	82	0.002522631	68	0.003525733
82	0.004359391	72	0.005360555	98	0.002141742
75	0.004516115	104	0.004490271	67	0.012256500
87	0.001774180	55	0.003739006	69	0.007037814
110	0.008939012	107	0.005966298	83	0.002485204
103	0.013767079	113	0.007057731	100	0.008386930
54	0.007855857	69	0.002080393	93	0.006673158
104	0.011318360	77	0.003023196	62	0.008922893
58	0.002015556	119	0.004823059	84	0.005572696
82	0.004156479	61	0.001225343	106	0.005008547
63	0.000634820	92	0.005320844	116	0.001657576
83	0.005742198	81	0.002925717	81	0.005338012
73	0.002256976	99	0.003715583	66	0.000826014
96	0.009464289	75	0.001029408	64	0.003900782
60	0.007349679	72	0.005090994	97	0.005369605
74	0.000417848	91	0.002357713	52	0.004585423
118	0.007098448	112	0.001070400	103	0.000087944
97	0.001012463	98	0.002881798	82	0.000892487
107	0.001048668	54	0.000929660	111	0.003112025
108	0.006497411	82	0.000479016	113	0.001257904
97	0.004377501	83	0.001468826	118	0.003067270
115	0.005672813	114	0.001771764	77	0.003862030
83	0.004170300	89	0.001382465	57	0.003654150
84	0.004335185	93	0.000288343	115	0.002680007
63	0.003319612	118	0.002905169	82	0.002915514
54	0.003426330	101	0.001968725	52	0.002000439
89	0.000806860	75	0.001874994	65	0.001914794
53	0.000381719	63	0.001676882	107	0.000079605
101	0.003488032	105	0.002024835	107	0.002125688
115	0.000487090	107	0.000243846	114	0.000260857
93	0.001886879	56	0.001995274	103	0.001365746
117	0.000092592	55	0.001681006	115	0.001104384
66	0.002273060	56	0.000750994	63	0.001603794
117	0.000538038	117	0.000054355	52	0.001019135
114	0.001863371	90	0.001298790	65	0.001583811
56	0.002661788	74	0.001274144	57	0.001194412
74	0.001352362	65	0.000069709	104	0.000685169
98	0.001178180	99	0.000340808	97	0.000303569
58	0.001155557	113	0.001449225	109	0.000593472
54	0.001173865	115	0.000421203	96	0.000280442
101	0.001569184	53	0.000371200	63	0.000433946
104	0.000777017	65	0.000579028	60	0.000767475
104	0.000932101	68	0.001152415	111	0.000311363
84	0.001266345	60	0.000256579	94	0.000008730
108	0.000872474	53	0.000067275	63	0.000416738
78	0.001278578	111	0.000267430	75	0.000899659
64	0.000889229	78	0.000377526	100	0.000662476
71	0.000147064	51	0.000606283	86	0.000254519
100	0.001078807	71	0.000100088	103	0.000614554
100	0.000834881	77	0.000153744	103	0.000164247
105	0.000305806	80	0.000549281	117	0.000221087

116	0.000539614	85	0.000220847	91	0.000303540
94	0.000077760	86	0.000832108	109	0.000118787
82	0.000005442	79	0.000248708	117	0.000423725
85	0.000492232	117	0.000720463	55	0.000326071
118	0.000179172	100	0.000560704	63	0.000173386
110	0.000308886	111	0.000315304	67	0.000347158
78	0.000658908	54	0.000476974	114	0.000180141
104	0.000392085	96	0.000337838	84	0.000103173
67	0.000488143	88	0.000074239	97	0.000393099
65	0.000127640	67	0.000363232	66	0.000010223
74	0.006417982	118	0.004578697	78	0.004078990

From 4 to 2		From 4 to 3		From 4 to 5	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
59	0.011100000	113	0.019000000	55	0.047700000
118	0.000890010	66	0.058958100	91	0.011141910
74	0.047128077	92	0.004518005	60	0.082916028
51	0.068025762	88	0.006055658	74	0.030381769
79	0.024178115	70	0.069727320	83	0.034935704
110	0.067894243	106	0.030386847	69	0.020853917
97	0.066913171	106	0.053305962	82	0.076280582
98	0.025128246	102	0.043587766	99	0.002644002
115	0.064259664	52	0.032222161	112	0.003119157
92	0.015174930	60	0.008869096	66	0.032155255
52	0.044235745	109	0.016968901	67	0.028748992
53	0.048765717	57	0.051067934	57	0.053915814
59	0.042853425	67	0.058596162	67	0.021627778
88	0.006817722	97	0.007982347	97	0.025796786
112	0.000139991	80	0.006842172	103	0.049928206
115	0.042824258	116	0.000053191	71	0.006307674
105	0.035334155	65	0.011966813	109	0.000707320
100	0.003184362	55	0.037172247	119	0.027920759
118	0.026691062	55	0.015640106	63	0.000354335
100	0.003763844	93	0.014432748	93	0.036024711
61	0.021459199	71	0.042050856	91	0.034840418
107	0.013396180	56	0.018107266	70	0.000631888
54	0.018486875	111	0.007614274	73	0.028015558
107	0.011089873	62	0.035523876	109	0.018970744
119	0.015238932	64	0.017886730	50	0.011245600
71	0.002530243	108	0.020617027	53	0.019771178
105	0.015123539	77	0.021386235	117	0.008703998
92	0.010603754	114	0.018814913	117	0.018142163
54	0.002911873	55	0.012287296	75	0.019646870
111	0.012680566	74	0.016586583	110	0.019133901
115	0.020297294	76	0.009864273	64	0.019968967
108	0.008962365	94	0.003896040	91	0.010933564
73	0.019060947	107	0.018286490	58	0.002967670
50	0.015506177	60	0.012122082	78	0.008729862
59	0.010191596	90	0.000790410	119	0.002014722
76	0.003708931	85	0.013914616	108	0.016106636
64	0.007795217	108	0.015454838	79	0.006952039
55	0.006277685	107	0.015304259	111	0.006342612
113	0.008989769	61	0.010223700	59	0.001488183
105	0.007288605	112	0.000894063	86	0.007505477
50	0.002154213	105	0.010562446	86	0.001805341
60	0.004414444	115	0.011976062	81	0.005890285
98	0.009904984	69	0.003874366	112	0.009174692
58	0.002495007	92	0.006028368	61	0.007130433
66	0.003175942	71	0.004603766	53	0.004251055
51	0.008671888	94	0.006446489	65	0.005309218
107	0.001891769	118	0.009557856	59	0.009346023
61	0.002585144	115	0.007864633	92	0.007218087
103	0.003441936	70	0.007401854	68	0.007261263
54	0.001012351	93	0.000603450	112	0.001592848
89	0.008076635	53	0.004340501	74	0.003452099
65	0.002867923	73	0.001151934	88	0.007387831
60	0.000528561	50	0.000899220	85	0.001231033
111	0.002098811	105	0.003620604	64	0.004065088
83	0.001193209	110	0.000031045	79	0.003895332

85	0.002859997	84	0.002507145	84	0.001753165
72	0.002057195	89	0.001685290	65	0.002387380
85	0.004826025	109	0.003622387	67	0.000484081
100	0.002759683	82	0.003894663	79	0.001100335
101	0.002714377	111	0.002462044	93	0.002674186
107	0.004152014	63	0.002968964	119	0.004835082
93	0.004109517	69	0.000723173	77	0.003355328
91	0.002327483	78	0.003889105	111	0.001879842
78	0.000012834	100	0.003417887	101	0.002727634
73	0.003391339	113	0.003437904	106	0.000734045
96	0.002035667	93	0.000789408	59	0.001608965
90	0.002539048	53	0.002648732	63	0.001423804
119	0.000149640	51	0.000558211	60	0.000477343
116	0.002678464	78	0.002259055	89	0.002214639
65	0.001988645	113	0.000410631	113	0.002775418
117	0.002479606	87	0.000141952	82	0.002120114
102	0.000173272	96	0.000804573	59	0.001620257
106	0.001645282	115	0.000607168	109	0.002999520
87	0.001307365	54	0.001755991	66	0.002160780
98	0.001004358	58	0.000720644	91	0.001387706
89	0.000708210	79	0.001308304	70	0.000207455
96	0.001321372	52	0.000291649	60	0.001099581
70	0.001679747	111	0.000905355	63	0.001169498
74	0.000772792	73	0.000364461	50	0.002028728
105	0.001195809	103	0.000258484	51	0.000242301
96	0.001557189	78	0.001852587	79	0.001246840
116	0.001102235	94	0.001649267	102	0.000475046
85	0.001481036	76	0.000411137	79	0.000235135
108	0.000864800	74	0.001176819	54	0.001574336
72	0.000591357	68	0.000969488	66	0.000226569
106	0.001149377	65	0.000953659	85	0.001403804
67	0.000119348	106	0.000921930	69	0.001319063
86	0.001080832	116	0.000170674	118	0.000517196
51	0.000538153	116	0.001117372	99	0.001168220
52	0.000909126	65	0.000172672	105	0.001082878
118	0.000750489	107	0.000033757	60	0.001044354
71	0.000697748	100	0.000617847	119	0.000266505
66	0.000244675	85	0.000044057	111	0.000047886
109	0.000657592	63	0.000869464	65	0.000699686
77	0.000122598	114	0.000508585	60	0.000558268
98	0.000323497	109	0.000187578	57	0.000529479
118	0.000319856	119	0.000619111	87	0.000595268
117	0.000045113	102	0.000432848	71	0.000583156
111	0.000037524	78	0.000509710	110	0.000563616
88	0.005102781	114	0.006406301	54	0.005812129

From 4 to 6 Time interval no. 0		From 4 to 7 Time interval no. 0		From 5 to 0 Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
75	0.053600000	93	0.026200000	94	0.053200000
81	0.002082080	89	0.064368180	111	0.068169600
79	0.076584183	67	0.003001125	93	0.031894284
67	0.040349619	100	0.029096425	51	0.055291868
106	0.065363345	57	0.015090149	112	0.002453477
57	0.009525260	52	0.032679052	90	0.074244032
57	0.073443562	56	0.074329030	105	0.066042599
81	0.042916083	61	0.050298720	77	0.053907314
82	0.023982322	74	0.070423238	96	0.055197146
76	0.055216250	118	0.030900836	62	0.011871193
72	0.040489341	105	0.025774286	94	0.037574268
90	0.001497699	104	0.046862740	54	0.041908186
109	0.017250834	56	0.031486890	92	0.008695973
83	0.028866566	111	0.013086620	68	0.012527177
60	0.024566842	65	0.011430464	82	0.040780685
107	0.037185065	78	0.012016798	93	0.010158170
54	0.001139827	64	0.017499716	109	0.032944961
104	0.017902003	93	0.027529164	107	0.022269726
87	0.009894998	106	0.012872138	106	0.009754428
81	0.009075459	52	0.018227449	93	0.021933601
118	0.010518457	99	0.010753790	117	0.028281932

73	0.030656043	90	0.001316256	57	0.012992789
50	0.017411180	105	0.004796889	91	0.023451964
80	0.025211218	79	0.029041864	97	0.015891388
110	0.005876598	103	0.001125030	81	0.018520416
77	0.014193274	114	0.002378552	86	0.017255888
98	0.021826116	82	0.016398350	68	0.011524889
90	0.015551712	56	0.006195614	85	0.014803856
116	0.009614175	83	0.023485619	82	0.000380791
61	0.001898426	89	0.018936776	57	0.006193682
72	0.014514499	61	0.012530319	56	0.010253477
72	0.005751213	57	0.022478575	62	0.012081539
117	0.005273631	103	0.018468891	78	0.003655765
54	0.001449868	62	0.015521460	56	0.010478150
63	0.000113593	68	0.011064905	56	0.007694060
56	0.004105828	109	0.000884737	52	0.003407658
114	0.002572929	117	0.010242540	71	0.004412565
114	0.000638855	104	0.009205306	67	0.005748693
77	0.013241668	109	0.012814112	92	0.004682653
51	0.005481105	51	0.015313827	68	0.005151699
113	0.010964908	61	0.001280475	85	0.004057010
99	0.014307116	85	0.002680750	107	0.003965932
59	0.004371311	71	0.004729037	80	0.004564911
60	0.003217951	105	0.009665606	119	0.001122917
81	0.001798236	70	0.005460020	76	0.005309766
84	0.001542105	59	0.000744358	62	0.004141171
56	0.004672373	101	0.004557769	63	0.001002777
113	0.003729975	70	0.002834462	92	0.001126780
117	0.000652104	62	0.007957589	105	0.000822958
112	0.008654755	68	0.000093567	55	0.001302931
82	0.003867713	116	0.007405926	110	0.003412423
92	0.009063516	86	0.002083623	92	0.002953937
58	0.001656811	62	0.008324350	110	0.000612690
91	0.005187883	57	0.007994606	92	0.003007156
112	0.004805157	99	0.004543175	79	0.002960716
67	0.001905218	74	0.005425647	71	0.002859828
77	0.006952776	100	0.004146840	103	0.001291754
62	0.004966103	83	0.000454015	113	0.001481836
117	0.002685301	117	0.005796797	113	0.001584462
60	0.004424036	89	0.002463575	84	0.001738906
58	0.005226920	119	0.002562736	72	0.002067394
93	0.000362799	79	0.004387298	88	0.000403934
86	0.005646676	78	0.003838367	89	0.000515184
58	0.002908862	71	0.002604359	61	0.000196099
65	0.003084975	67	0.002229998	110	0.000039628
54	0.001558175	73	0.000550560	110	0.000850297
108	0.001317167	83	0.000058587	83	0.001854558
109	0.002408782	50	0.001782276	106	0.001285334
54	0.003516797	90	0.001064412	116	0.000183096
107	0.003635415	50	0.001532878	100	0.000441510
101	0.002051397	66	0.002973850	57	0.000301675
74	0.002936689	50	0.002231047	104	0.000783011
112	0.000765022	56	0.001170717	91	0.000406764
100	0.000702417	73	0.000361527	51	0.001026475
63	0.000712617	113	0.000613421	70	0.000098605
74	0.001576821	70	0.000719911	72	0.000243335
113	0.000607953	60	0.002238117	97	0.000469848
93	0.001604551	51	0.001179099	117	0.001135666
101	0.001276163	70	0.001579931	64	0.000444878
73	0.000314535	106	0.000818987	60	0.000624183
52	0.000071043	90	0.001067765	119	0.000489919
72	0.000590072	97	0.000340214	114	0.000361754
82	0.001909338	67	0.000396169	119	0.000774696
77	0.000197160	68	0.001340184	78	0.000624703
118	0.001180040	91	0.000038696	94	0.000116510
91	0.000182838	84	0.001179378	90	0.000573346
88	0.000320118	113	0.000221007	111	0.000140368
114	0.000881203	63	0.000211564	91	0.000079180
83	0.001136986	81	0.001142778	58	0.000012929
62	0.000384015	101	0.000016275	91	0.000175487
67	0.001291433	81	0.000267505	66	0.000257326
75	0.001295566	97	0.000036279	106	0.000475494
69	0.000583869	67	0.000098433	98	0.000368637

80	0.000689120	118	0.000907192	102	0.000053300
97	0.000223879	119	0.000711209	51	0.000431740
107	0.000535931	61	0.001132968	87	0.000251370
85	0.000684785	83	0.000734743	117	0.000049710
114	0.000654082	109	0.000784148	69	0.000260686
69	0.000061017	62	0.000844928	69	0.000155212
81	0.008655698	81	0.009261866	87	0.003972758

From 5 to 1		From 5 to 2		From 5 to 3	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
66	0.011200000	71	0.013000000	116	0.061500000
59	0.049044480	96	0.043822800	65	0.044391050
87	0.029414348	112	0.045744094	86	0.016272783
83	0.071188680	51	0.078704883	53	0.071982566
108	0.069985318	104	0.014573362	84	0.039245070
63	0.062764041	85	0.027662927	91	0.019548518
111	0.026348837	67	0.077493895	79	0.065517163
91	0.047875822	51	0.026701725	96	0.035712845
59	0.044505365	65	0.056943498	89	0.056122627
72	0.007992354	90	0.059443082	94	0.015096509
71	0.044229642	101	0.008783374	64	0.029707382
105	0.033090879	66	0.032772869	102	0.042774924
53	0.041545191	100	0.040685361	119	0.012452788
103	0.042579310	57	0.036093511	58	0.039271997
50	0.035424567	56	0.020609765	119	0.002026817
102	0.018451498	118	0.013259482	108	0.038829445
112	0.014355771	50	0.002785567	92	0.018634412
50	0.010465117	97	0.005933613	106	0.021969316
96	0.010016394	117	0.016668417	58	0.006493411
60	0.011632140	108	0.002799552	55	0.006161656
69	0.010204277	84	0.002253109	67	0.024868953
62	0.000307686	97	0.006420160	103	0.002154228
74	0.022100499	99	0.028797329	74	0.014257198
55	0.021082028	82	0.009735772	81	0.027122218
98	0.023513422	56	0.018516788	70	0.006995633
59	0.016149785	101	0.005545332	50	0.020589273
64	0.012888168	86	0.006297969	51	0.001613868
89	0.009947286	53	0.004260710	86	0.011304637
72	0.010447910	112	0.001850254	110	0.010736410
81	0.013578692	100	0.003181065	83	0.006531438
88	0.009754110	75	0.007505153	54	0.005706849
104	0.016690889	104	0.008518984	108	0.010188124
55	0.010767255	63	0.005289131	95	0.015445254
114	0.006840316	73	0.004919175	95	0.019122120
75	0.003861558	72	0.022988631	54	0.001886351
52	0.010899535	78	0.018987486	77	0.006879551
88	0.011540998	51	0.011176875	104	0.007929139
110	0.010592073	83	0.014105088	50	0.008213057
91	0.007602488	83	0.001385701	102	0.010615467
69	0.002905353	112	0.018157515	80	0.008143286
102	0.000258648	73	0.014506707	105	0.003494832
65	0.001005700	97	0.008619882	85	0.009910314
99	0.000679613	101	0.012215203	52	0.001323870
68	0.002671421	105	0.011320935	108	0.006547859
107	0.004349309	51	0.006860846	115	0.004324521
118	0.001174219	68	0.011062480	98	0.007660668
109	0.003370212	99	0.008028188	99	0.008300067
106	0.005278514	99	0.002348668	57	0.000009442
88	0.001227195	69	0.003754733	110	0.002983485
114	0.004190529	118	0.004961708	79	0.005778414
105	0.003689628	81	0.002546922	104	0.004214088
75	0.002385326	117	0.001886334	105	0.001522893
109	0.001661288	60	0.008200005	109	0.004978718
58	0.001313439	61	0.000118970	116	0.004196444
99	0.003082323	100	0.002977710	53	0.000099036
111	0.002643564	96	0.006226905	98	0.001271538
67	0.000585714	61	0.005487200	50	0.006312622
117	0.002393108	104	0.003515392	83	0.005082382
72	0.002920907	90	0.001475890	101	0.000469593

69	0.000020668	89	0.002356648	79	0.003093762
81	0.002516059	77	0.002994905	111	0.000614846
96	0.003247438	61	0.004278621	51	0.003695804
71	0.003135607	93	0.002523986	78	0.003512043
119	0.000350087	56	0.001382829	98	0.004551689
61	0.000432883	106	0.001153956	75	0.000029426
81	0.001284280	77	0.000484060	94	0.001810515
110	0.001939253	56	0.001698081	68	0.003211724
87	0.000426135	83	0.000225983	104	0.001116949
96	0.002286199	110	0.001972170	82	0.001158540
68	0.001058702	76	0.003243294	113	0.001430035
117	0.000066522	94	0.002905069	61	0.000941811
92	0.000594629	55	0.002201528	83	0.002988007
95	0.001088547	110	0.001418386	98	0.001520316
113	0.001647954	101	0.002809537	98	0.002195741
83	0.000628713	116	0.000437438	72	0.000769010
78	0.001822064	95	0.000109699	68	0.001961822
112	0.001732394	106	0.001578814	112	0.002015450
63	0.000313851	57	0.002396051	102	0.001395278
115	0.000889068	57	0.000406121	82	0.000214413
57	0.001542615	91	0.001479252	104	0.000281454
71	0.000018605	109	0.001868721	115	0.001842636
55	0.000826135	64	0.000934260	64	0.000084053
59	0.000642369	118	0.001031451	76	0.000442102
96	0.000777164	100	0.001383206	113	0.000339200
91	0.000983062	71	0.001178825	58	0.000293189
109	0.000558745	97	0.000180701	74	0.000663795
77	0.000917132	110	0.001130707	62	0.000012265
53	0.000399836	103	0.000846816	76	0.000913013
50	0.000386838	81	0.000675971	93	0.000145501
63	0.000749026	114	0.000215146	50	0.001269184
118	0.000223059	103	0.000212561	50	0.001164021
94	0.000677263	74	0.000764415	95	0.000068598
101	0.000399102	92	0.000388330	102	0.000994785
85	0.000632105	70	0.000551348	87	0.000416563
81	0.000213628	68	0.000556093	53	0.000121064
65	0.000076798	115	0.000655147	108	0.000358942
99	0.000166759	111	0.000656320	88	0.000745971
57	0.000535812	117	0.000630848	83	0.000172394
116	0.000013333	89	0.000457920	55	0.000391070
57	0.005114826	96	0.007136105	88	0.008557900

From 5 to 4		From 5 to 6		From 5 to 7	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
84	0.066300000	66	0.090900000	73	0.092200000
60	0.088981610	89	0.014727420	75	0.060641040
109	0.051527822	56	0.000268312	118	0.008979885
112	0.039818167	62	0.068398977	115	0.011399235
61	0.003164164	63	0.042441252	54	0.043902009
75	0.008252291	111	0.056238358	66	0.023564623
112	0.010090601	86	0.047474777	113	0.050190603
52	0.023346505	82	0.002650249	104	0.028364904
60	0.021609825	98	0.018614768	97	0.036624764
87	0.049732213	96	0.007372802	97	0.031304861
86	0.010131111	119	0.001106552	89	0.042530268
50	0.021633076	116	0.054063904	80	0.013915266
119	0.052731439	53	0.048553024	117	0.007344250
79	0.029402639	74	0.032886095	107	0.030636337
63	0.027001173	92	0.013834764	61	0.031311478
63	0.027543394	92	0.043991203	116	0.035362769
90	0.013358918	98	0.013694326	69	0.037628918
86	0.010655776	70	0.005667625	108	0.020953399
85	0.004669552	100	0.016304412	96	0.029761106
112	0.010429178	66	0.015948744	53	0.014717064
95	0.038021418	115	0.010850313	85	0.033472053
111	0.002936993	112	0.006619404	55	0.023198364
82	0.009366757	114	0.022120124	65	0.005839936
95	0.023516313	79	0.010739014	103	0.014908773
80	0.030952779	91	0.016343998	82	0.019773986

115	0.004320190	90	0.018532789	85	0.016320670
50	0.022563629	115	0.017836849	56	0.013098047
55	0.009832101	87	0.006700403	104	0.014144928
95	0.023077640	117	0.021661774	84	0.002494926
117	0.025867194	57	0.007766201	99	0.017480862
63	0.008562126	95	0.015410111	70	0.006991170
71	0.014827799	60	0.023276175	118	0.008305307
109	0.017499402	113	0.000113503	91	0.008856340
87	0.011995710	51	0.005105065	51	0.001310255
93	0.015032836	75	0.007452034	108	0.004354239
106	0.000702115	72	0.006772976	101	0.014451927
78	0.005048148	52	0.005936265	84	0.003922066
55	0.010591833	62	0.014698494	118	0.008370628
69	0.014468179	58	0.014636380	93	0.009196092
111	0.010603022	113	0.008235489	52	0.009346514
65	0.010412716	77	0.000951519	75	0.003362338
56	0.003809550	50	0.002446553	56	0.004696169
90	0.000658989	98	0.001012139	56	0.003090763
87	0.009966434	115	0.002187135	94	0.007402367
64	0.002005245	103	0.001039221	53	0.001819575
90	0.007249892	61	0.015469790	51	0.005011274
60	0.008797728	68	0.013432413	53	0.001696484
53	0.002495000	100	0.004552332	58	0.001209091
90	0.002051863	57	0.011792245	118	0.005224700
65	0.005429299	68	0.010205565	60	0.005576012
99	0.000384788	88	0.002009227	62	0.005029161
50	0.004808775	114	0.002503610	98	0.001353630
54	0.006415709	60	0.002478850	119	0.004028043
100	0.004692013	85	0.006164726	60	0.005212103
93	0.002674945	76	0.007059904	90	0.001051944
53	0.004163065	56	0.001413122	51	0.004479719
114	0.001092512	107	0.005029948	56	0.004927380
103	0.002029945	112	0.004413822	83	0.004708747
68	0.005034102	89	0.000041936	69	0.002353695
54	0.003630112	66	0.005867485	75	0.001181371
68	0.002816122	54	0.005662543	111	0.001135168
78	0.001666661	82	0.001749630	51	0.002446118
65	0.000409283	61	0.001838569	67	0.000705935
112	0.001816127	53	0.000470702	110	0.001053851
67	0.002804685	89	0.004433213	60	0.002834998
54	0.000104060	93	0.000832142	93	0.003117702
99	0.000700158	86	0.004850677	88	0.000916771
53	0.000529633	115	0.000856434	90	0.001871706
105	0.001462574	59	0.003290011	92	0.002272399
68	0.002041927	50	0.001755983	80	0.001139210
73	0.002275333	88	0.001074646	59	0.001769082
67	0.001938410	102	0.003568222	60	0.001757309
59	0.002020483	117	0.003339916	62	0.000432290
104	0.002022378	61	0.001897355	81	0.000210167
61	0.001289743	54	0.000380071	99	0.000230787
51	0.001049964	105	0.000299423	110	0.000762210
62	0.000731201	100	0.001171033	67	0.001439673
77	0.000055600	109	0.001757809	107	0.001372689
95	0.000945245	101	0.001557099	95	0.000069738
78	0.001455378	55	0.001904560	116	0.001100507
119	0.000326573	64	0.000902844	101	0.001093733
83	0.001124964	56	0.001154111	96	0.000878601
76	0.000031113	63	0.001783227	89	0.000023771
50	0.000882640	95	0.001455606	66	0.000824758
60	0.000683813	99	0.001116640	74	0.000788547
91	0.000859130	82	0.001199414	76	0.000348378
101	0.000374568	50	0.001087691	57	0.000769058
82	0.000430703	70	0.000991145	60	0.000060378
79	0.000585888	59	0.000580141	60	0.000123047
70	0.000314662	72	0.000190692	82	0.000011014
73	0.000786852	61	0.000463609	102	0.000582913
73	0.000437005	51	0.000441693	91	0.000017455
103	0.000344469	51	0.000506261	117	0.000132052
109	0.000190687	80	0.000091378	102	0.000491525
107	0.000012395	59	0.000249085	85	0.000454294
86	0.000141294	81	0.000453998	52	0.000345335
54	0.000200017	97	0.000157855	116	0.000376190

```
98        0.000059848   66        0.000110085   102       0.000154411
74        0.000253569   62        0.000157131   59        0.000207831
77        0.005856519   104       0.008200888   71        0.005093969

From 6   to 0           From 6   to 1           From 6   to 2
Time interval no. 0     Time interval no. 0     Time interval no. 0
--------------------    --------------------    --------------------
Speed     Probability   Speed     Probability   Speed     Probability
--------- -----------   --------- -----------   --------- -----------
98        0.076100000   119       0.030100000   111       0.055000000
114       0.069292500   83        0.026769240   86        0.016159500
83        0.027689283   113       0.059417238   104       0.002043449
76        0.058380426   88        0.004683682   89        0.057461417
102       0.057794042   115       0.045797455   62        0.055724414
85        0.041365286   53        0.051077145   109       0.051582951
115       0.003213017   52        0.062572419   77        0.048388795
119       0.017919851   72        0.026696523   113       0.011632323
95        0.050757630   74        0.039979539   62        0.010600308
76        0.016371170   66        0.026442724   99        0.056418798
88        0.015109037   107       0.023053877   77        0.012191770
96        0.042959989   99        0.043807578   108       0.060286679
107       0.022491054   101       0.017515561   100       0.018394064
82        0.005856514   78        0.043529588   75        0.051038037
110       0.046353409   94        0.027121524   56        0.037276659
102       0.037930139   54        0.023524652   100       0.017092531
72        0.000615625   71        0.026605929   119       0.015705757
54        0.040201481   86        0.019885611   104       0.015143491
96        0.010127028   91        0.009834783   75        0.031853792
83        0.030806795   98        0.024121632   99        0.015453816
75        0.032307841   79        0.024840519   115       0.032269354
103       0.027650191   69        0.024908676   51        0.015462087
87        0.026252742   88        0.012359079   87        0.023242526
50        0.014474561   116       0.009924038   72        0.012625578
103       0.000319173   114       0.000590862   71        0.008557814
84        0.013864568   71        0.021022101   111       0.017043025
114       0.018749966   117       0.025519840   87        0.005630264
81        0.008308989   61        0.015617956   116       0.017003879
56        0.003006477   91        0.017520821   100       0.012808148
115       0.000992149   88        0.012672889   81        0.009759077
110       0.004787764   65        0.015044748   96        0.012987431
108       0.014396261   79        0.007385206   86        0.008074183
50        0.010729211   81        0.009831088   114       0.008828702
68        0.014059977   71        0.007575034   95        0.000687412
119       0.000194272   69        0.013760227   109       0.009551115
92        0.005154863   104       0.001518680   105       0.006591028
70        0.007031061   111       0.009048612   80        0.004782895
113       0.007090236   118       0.001978018   111       0.007778741
85        0.008792073   56        0.013334532   62        0.005316629
66        0.009315433   62        0.004206955   80        0.013758812
79        0.008783112   75        0.001033590   50        0.011820829
74        0.007909852   112       0.007231068   94        0.009103796
81        0.001216727   66        0.003957288   111       0.000919066
114       0.003655914   87        0.003986147   108       0.008073318
89        0.005048055   79        0.009151503   78        0.000920032
112       0.003780915   111       0.007849281   94        0.007785543
74        0.003058273   98        0.000907302   72        0.002300590
119       0.000724765   77        0.005724857   102       0.002632148
60        0.006359278   68        0.006932897   75        0.005071093
115       0.005306965   69        0.007188543   88        0.000459165
99        0.000520234   55        0.003248531   56        0.002211671
92        0.004095340   53        0.000437306   91        0.005828957
78        0.000608742   66        0.001302604   85        0.003427327
71        0.001984742   108       0.002076882   63        0.001627189
118       0.004293643   116       0.000167549   73        0.002932858
114       0.001547600   87        0.003542896   86        0.002231378
115       0.000372185   89        0.001102930   58        0.002919628
94        0.004150236   65        0.000704398   81        0.004228320
113       0.003701592   109       0.004201531   119       0.003954848
75        0.002916371   61        0.000442118   88        0.004115372
101       0.002576385   50        0.004442405   95        0.004002821
75        0.001008768   51        0.001666431   94        0.003116746
58        0.000259141   117       0.003893203   113       0.003791950
```

52	0.002367698	78	0.002918704	54	0.000628021
87	0.001990323	65	0.001068558	88	0.001960763
78	0.001266899	111	0.000729660	59	0.001805398
106	0.000767620	99	0.000608562	106	0.001122147
59	0.000675604	68	0.002796026	111	0.002093889
68	0.001232673	52	0.002406437	73	0.002323586
81	0.000619670	76	0.000048752	69	0.000948549
108	0.000970918	86	0.000516377	112	0.001387396
82	0.001165243	79	0.001047504	79	0.002118847
99	0.000874381	75	0.000669903	90	0.000565998
90	0.000393678	68	0.001227683	97	0.001632344
105	0.000013486	111	0.000624711	106	0.000026597
64	0.000250014	107	0.001039603	71	0.001108306
90	0.000111879	100	0.000407520	54	0.001188292
59	0.001289975	119	0.000320382	78	0.000325108
101	0.000342299	83	0.001400116	95	0.001250369
93	0.000413958	57	0.000599384	110	0.000712373
56	0.000556531	106	0.000107421	78	0.000015744
110	0.000210109	59	0.001218883	98	0.000993312
83	0.000438816	60	0.000567820	71	0.001150551
88	0.000656453	113	0.000667323	88	0.000824231
116	0.000337077	116	0.000591702	92	0.000725758
101	0.000308841	56	0.000065719	77	0.000119514
87	0.000039214	80	0.000234662	82	0.000144940
51	0.000205322	89	0.001283466	116	0.000047560
75	0.000676918	85	0.000905719	82	0.000608690
95	0.000534309	51	0.000955893	62	0.000526369
86	0.000663857	96	0.001142597	72	0.000181567
55	0.000413430	112	0.001089190	75	0.000417109
91	0.000185791	84	0.000448237	69	0.000374867
117	0.000534802	85	0.000511028	85	0.000066404
64	0.000169352	63	0.000105197	117	0.000461105
91	0.000001326	77	0.000351065	55	0.000329582
118	0.000082883	100	0.000063448	50	0.000466094
66	0.000255362	64	0.000418801	118	0.000498105
96	0.000183108	54	0.000664493	113	0.000343713
63	0.006109268	67	0.007789614	95	0.004824905

From 6 to 3		From 6 to 4		From 6 to 5	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
114	0.023800000	77	0.000100000	53	0.040500000
56	0.060329160	86	0.065193480	78	0.044520800
62	0.014562346	51	0.076552464	64	0.090491443
113	0.069581016	97	0.008838987	82	0.077996542
75	0.062213215	73	0.034142466	86	0.054941753
118	0.049633670	72	0.000244552	50	0.041492968
66	0.014109660	119	0.053133309	66	0.030292633
63	0.045310494	110	0.030928867	85	0.005082064
60	0.031834193	110	0.000438520	56	0.043888280
106	0.027722417	110	0.067126274	76	0.052912559
81	0.047471402	105	0.029715888	72	0.036199879
115	0.035364332	71	0.032946430	76	0.008959268
111	0.005957783	104	0.048952059	88	0.002978147
99	0.009269197	68	0.025101745	67	0.031097031
54	0.007643185	60	0.003949387	91	0.016054467
105	0.000990396	52	0.028535902	63	0.012170654
89	0.043539684	52	0.044419560	107	0.037635653
77	0.044571050	68	0.026126414	64	0.018564736
69	0.037239077	86	0.013426652	51	0.033190519
56	0.012910020	75	0.039249158	112	0.015152645
92	0.011497111	87	0.012090619	90	0.022482030
91	0.031620564	100	0.027339590	107	0.019582659
56	0.027059797	116	0.005004860	81	0.012926850
104	0.016146018	100	0.000065289	82	0.023106639
105	0.006390094	90	0.031397518	91	0.019019612
102	0.004843508	54	0.018465749	53	0.007786754
109	0.004702709	113	0.007189371	74	0.004220442
70	0.003399418	77	0.025370405	79	0.006886354
70	0.011463213	98	0.024322262	52	0.004309972

101	0.001528482	110	0.004612277	113	0.002634904
81	0.017132828	88	0.004859451	101	0.014615447
99	0.004645460	116	0.005464173	74	0.002743393
116	0.020711328	75	0.015843495	88	0.011887416
55	0.006954616	117	0.001831872	104	0.004840778
106	0.003700695	76	0.005423608	67	0.000089301
75	0.004401230	111	0.010060493	54	0.002231181
76	0.008124729	94	0.015661315	107	0.013274189
108	0.002196812	76	0.007310563	74	0.007807866
63	0.000372744	100	0.008111648	111	0.008567017
87	0.000355018	89	0.008848560	101	0.000782997
73	0.002395559	56	0.003461205	91	0.003819103
77	0.013770118	111	0.002485985	55	0.008554445
78	0.009350436	50	0.005239921	71	0.008452252
62	0.005441038	94	0.009416661	115	0.005820163
91	0.003595123	85	0.003607532	84	0.007700456
98	0.010919733	112	0.005659636	82	0.006457123
62	0.000406657	81	0.001627741	52	0.005676008
112	0.010857524	78	0.003143331	101	0.005415083
93	0.004344248	76	0.004528161	77	0.004210702
117	0.001463644	78	0.001663823	60	0.004342322
104	0.004416142	63	0.005636865	99	0.000851208
103	0.006023074	93	0.002460370	96	0.003199344
110	0.004843333	55	0.004232885	103	0.004673506
105	0.003589551	119	0.007836221	88	0.002106279
74	0.001405291	111	0.006552091	110	0.003732187
115	0.002387460	65	0.000935164	82	0.001429305
114	0.005944659	80	0.001521124	83	0.003329984
100	0.002411735	57	0.001804764	88	0.001488820
73	0.006604455	74	0.004496305	100	0.002040679
106	0.002323220	67	0.005208787	109	0.002399729
105	0.004667687	59	0.000891534	105	0.000513480
103	0.003323195	91	0.000295168	107	0.001176877
85	0.003236890	78	0.001735932	87	0.000783322
53	0.001935316	57	0.002254418	58	0.000175832
88	0.005028689	85	0.000296400	62	0.001586382
85	0.003855729	56	0.001137625	63	0.000585705
53	0.001797346	59	0.001678140	87	0.000046011
66	0.003922858	80	0.001993716	74	0.001127801
76	0.003075250	95	0.002340432	96	0.001195014
117	0.003182884	57	0.001861903	119	0.000934679
107	0.002500579	79	0.002249979	68	0.001077304
52	0.001285227	113	0.001911009	102	0.001408540
72	0.001175623	57	0.002335988	77	0.000235293
112	0.001284834	56	0.000599541	71	0.000373161
71	0.001519863	53	0.000313549	64	0.000620915
88	0.001797044	117	0.001767567	53	0.001485306
90	0.001418231	65	0.001218114	106	0.000699371
74	0.000161128	52	0.000032767	115	0.000884993
95	0.000471295	51	0.001399586	73	0.000420878
56	0.000637629	112	0.001725907	53	0.000749619
81	0.000890917	78	0.001126598	105	0.000705137
108	0.000761605	84	0.001702914	69	0.000390258
95	0.000254071	108	0.001856402	94	0.000326227
65	0.000915642	50	0.000081597	106	0.000601559
83	0.001358439	106	0.001097243	83	0.000128938
115	0.000833194	95	0.001231463	102	0.000287975
114	0.001347020	94	0.001127283	91	0.000535016
86	0.000016284	86	0.000703612	83	0.000343234
97	0.000448633	69	0.000027552	79	0.000165868
65	0.000119258	116	0.000860161	50	0.000453306
96	0.000754485	68	0.001103335	117	0.000961914
52	0.001185231	80	0.000446263	109	0.000798361
77	0.000857190	57	0.000492463	86	0.000341626
65	0.000436092	52	0.000694960	79	0.000107426
54	0.000159949	74	0.000405440	55	0.000664725
69	0.000146773	81	0.000081807	57	0.000292165
99	0.000616840	98	0.000349504	81	0.000355642
81	0.000324922	56	0.000793103	93	0.000524220
115	0.000536633	60	0.000069214	64	0.000293851
104	0.007967857	101	0.007796008	52	0.006025527

From 6 to 7
Time interval no. 0

Speed	Probability
107	0.060900000
68	0.068460390
95	0.086280385
59	0.024079828
116	0.073747101
87	0.018124453
72	0.026402110
84	0.032806493
115	0.016265620
78	0.005751456
103	0.040750442
105	0.052074943
94	0.008156887
119	0.018426976
115	0.017167266
67	0.000630848
110	0.037437904
105	0.017367803
83	0.004623478
80	0.008670113
66	0.020735840
60	0.025352004
111	0.016420017
119	0.029701191
96	0.024621648
109	0.009647631
70	0.015425989
102	0.003023637
107	0.007321679
56	0.011067967
119	0.016501122
75	0.011860733
100	0.010042351
78	0.008719439
92	0.001320044
51	0.013983388
107	0.002997712
110	0.003307675
87	0.001318464
55	0.002153351
62	0.000922028
99	0.001265255
85	0.012744304
100	0.003403831
55	0.008538814
92	0.002102837
51	0.011244674
65	0.005094330
62	0.005718725
118	0.003336159
56	0.004663521
72	0.005134366
107	0.000279428
87	0.000802672
102	0.004338993
90	0.005611425
66	0.006766580
71	0.006013622
52	0.001657766
55	0.004741315
106	0.000743214
78	0.004784860
83	0.003952461
106	0.003679845
117	0.001478760
53	0.003640049
57	0.002442800

From 7 to 0
Time interval no. 0

Speed	Probability
80	0.017300000
50	0.008746030
86	0.084441809
112	0.058174095
75	0.026602818
67	0.043536177
85	0.004871674
84	0.034412897
81	0.030392600
73	0.008644024
96	0.022466682
105	0.005415372
110	0.059080623
101	0.020976215
107	0.039210839
83	0.042858252
76	0.017743316
83	0.026607088
64	0.026462650
79	0.012155237
91	0.012624969
106	0.031384854
71	0.029234753
85	0.007642114
102	0.007238328
53	0.023554046
100	0.010407967
90	0.003079616
77	0.026366457
66	0.016070521
59	0.013084091
108	0.015701151
109	0.009586722
61	0.003018105
91	0.018825071
67	0.015677332
50	0.000815387
87	0.005861890
86	0.015892959
103	0.000877395
77	0.007190781
92	0.008091719
94	0.009460745
55	0.000366465
104	0.004325028
62	0.003212705
75	0.008758648
60	0.000548380
82	0.001262543
116	0.008597662
94	0.003253812
94	0.007136618
118	0.001469700
100	0.005304037
107	0.001694120
91	0.002833569
60	0.001166783
65	0.004322413
115	0.001554280
53	0.000655283
75	0.000815134
75	0.001364999
103	0.005730870
85	0.001130673
112	0.002514310
111	0.001972726
53	0.004373905

From 7 to 1
Time interval no. 0

Speed	Probability
111	0.044900000
77	0.071728010
117	0.053444005
84	0.062161606
88	0.044914333
61	0.069972078
54	0.011490687
76	0.056249840
77	0.057343665
97	0.046498808
59	0.040332686
107	0.005644343
71	0.017673990
50	0.033912851
91	0.011857353
53	0.022944733
110	0.018842275
117	0.028321614
67	0.018950975
77	0.001555489
107	0.012994242
110	0.008933272
112	0.012551724
97	0.010734992
78	0.014587669
67	0.019687684
86	0.014144152
71	0.000731745
93	0.004840585
80	0.010231468
91	0.011632426
60	0.003780501
87	0.008539997
57	0.007792760
65	0.007227996
93	0.010149698
91	0.007730084
53	0.007668477
118	0.005300679
66	0.008455842
101	0.005397527
78	0.008611976
54	0.001280516
56	0.001259900
65	0.007091216
66	0.006962392
63	0.005230409
82	0.000848348
71	0.003582330
105	0.001843490
76	0.000761261
65	0.004489832
119	0.002244614
71	0.003910736
95	0.002581943
54	0.000479337
51	0.002244009
108	0.000059032
68	0.000610118
62	0.001273683
81	0.003101177
71	0.000875584
69	0.001806088
90	0.002038436
61	0.002271425
75	0.000775974
98	0.001948094

53	0.002081316	79	0.000184170	81	0.001163972
101	0.000367538	110	0.003449598	75	0.000983305
68	0.001376742	93	0.001841902	118	0.001017405
74	0.001716830	50	0.001588696	110	0.000347771
86	0.001406258	64	0.002015851	86	0.000955906
55	0.002111870	104	0.002718994	109	0.000168916
115	0.000153114	58	0.000621785	118	0.000893619
95	0.000636879	89	0.001628021	119	0.000640884
80	0.001900359	58	0.000673501	111	0.000184832
84	0.000282751	55	0.002446706	81	0.000579727
88	0.001817959	84	0.001347378	112	0.000560242
57	0.000685535	118	0.001048801	109	0.001152103
57	0.001624585	53	0.000029141	92	0.000416449
66	0.001013998	97	0.001256431	113	0.000597325
59	0.000509524	103	0.000786566	100	0.000756223
84	0.000826153	74	0.000435364	70	0.000692124
93	0.000647170	73	0.000544427	80	0.000118649
112	0.000776332	102	0.000390681	60	0.000217521
64	0.001034289	103	0.000308461	96	0.000633758
51	0.000614851	93	0.001209423	63	0.000478589
84	0.000172071	56	0.001837719	86	0.000179963
93	0.000811811	114	0.001437378	80	0.000519545
109	0.000525325	81	0.001574404	80	0.000114009
113	0.000597961	108	0.000444369	95	0.000653305
65	0.000512552	94	0.000531745	102	0.000430583
115	0.000109232	75	0.000889548	52	0.000123941
83	0.000667434	71	0.000266048	112	0.000144845
62	0.000068976	78	0.001133075	119	0.000012051
89	0.000211474	53	0.000507803	101	0.000226358
50	0.000174312	101	0.000652821	114	0.000307580
67	0.000391984	104	0.000497433	96	0.000145505
84	0.000083527	95	0.000831000	83	0.000355216
53	0.005340276	64	0.008720727	70	0.004193001

From 7 to 2		From 7 to 3		From 7 to 4	
Time interval no. 0		Time interval no. 0		Time interval no. 0	
Speed	Probability	Speed	Probability	Speed	Probability
108	0.042800000	57	0.032800000	92	0.042800000
102	0.075140200	84	0.022052160	60	0.013017920
59	0.083001827	92	0.055385663	55	0.027003607
109	0.072714276	51	0.038170797	96	0.030908915
95	0.072561735	93	0.081667613	99	0.074535270
59	0.031708425	112	0.049583091	59	0.054548544
80	0.044416051	69	0.009076293	119	0.014083655
114	0.045750473	62	0.032789288	94	0.013821699
119	0.037446254	117	0.046339849	96	0.038068436
54	0.041287473	52	0.062138895	97	0.042855141
61	0.006389743	65	0.040526741	95	0.025415587
59	0.022964674	111	0.047705212	59	0.006354001
65	0.035092202	87	0.014934696	89	0.038906654
90	0.036851288	106	0.033238275	77	0.038935671
64	0.026636966	64	0.026795950	88	0.049079660
95	0.013822633	50	0.038564211	101	0.035941429
71	0.015415081	111	0.024929257	103	0.004310376
106	0.016664839	82	0.008513890	104	0.031144351
50	0.026229637	56	0.013291088	97	0.033587007
112	0.007188217	89	0.029738475	79	0.032082485
53	0.008828456	87	0.003150992	96	0.019851357
73	0.000569015	119	0.026984807	103	0.010947417
92	0.022398495	73	0.021505391	94	0.021142314
83	0.019270984	68	0.003025479	51	0.006915146
101	0.010073800	112	0.018848805	81	0.029198090
84	0.001422785	62	0.021038633	80	0.010343720
66	0.005408957	98	0.013824032	55	0.020844527
85	0.012438391	58	0.007390231	57	0.005180526
70	0.003922519	81	0.010401020	69	0.001140882
89	0.004960647	98	0.004951116	89	0.010216603
75	0.014033507	104	0.004931588	112	0.021508446
99	0.012790363	65	0.000093424	105	0.015390473
74	0.009137926	116	0.014783239	57	0.007106844

62	0.011378442	103	0.008477954	113	0.015432223
113	0.003912357	95	0.012348427	79	0.014746602
105	0.000600617	75	0.006852195	85	0.012223770
106	0.008161641	57	0.001312554	57	0.010967536
105	0.007786694	81	0.001207858	117	0.000979434
106	0.000977047	100	0.010421422	93	0.009631097
113	0.002450886	89	0.007195034	82	0.009936415
74	0.005789746	53	0.006287770	92	0.001631787
64	0.005484766	106	0.003625171	66	0.005378720
68	0.004247074	80	0.000332406	83	0.003133301
63	0.000104809	68	0.001514673	95	0.008387098
118	0.000802333	89	0.000544404	119	0.004420090
68	0.005289673	105	0.004075851	78	0.004731384
76	0.003744153	99	0.002076784	66	0.006879252
114	0.000892986	59	0.001670083	115	0.004284680
95	0.002603617	112	0.007164813	104	0.005878881
97	0.005090466	118	0.002569746	65	0.001316355
104	0.002746950	54	0.004174390	102	0.000718823
119	0.001841861	82	0.005691402	107	0.002638071
101	0.002019861	80	0.003261156	83	0.002831270
68	0.002022075	109	0.002861470	69	0.001124662
85	0.003007073	67	0.000047164	110	0.000100192
112	0.000738551	97	0.002388835	91	0.002144844
73	0.001091118	52	0.004329693	90	0.002822949
116	0.002594600	67	0.000432264	109	0.002938399
97	0.000052924	84	0.003185312	103	0.000889589
94	0.002730313	89	0.003369136	71	0.002821728
85	0.000523242	98	0.001927863	101	0.003250496
117	0.002513336	52	0.001860702	113	0.002641559
56	0.000061872	59	0.002544638	87	0.002483158
97	0.000646824	110	0.002280020	93	0.000356284
110	0.001846023	118	0.000565408	112	0.000742741
60	0.000974078	65	0.001785987	71	0.001568131
80	0.000015560	117	0.000381603	91	0.000505754
85	0.002114957	95	0.000849526	61	0.001042515
97	0.001980463	64	0.001797002	74	0.001539426
117	0.001824417	119	0.001598190	54	0.001792157
71	0.001483908	63	0.001226077	53	0.000735767
85	0.000520267	115	0.000914374	56	0.001663396
65	0.000147556	76	0.000457869	116	0.001447953
80	0.000180255	83	0.001126470	52	0.000823609
94	0.000415169	51	0.000049336	58	0.000285793
82	0.001104103	101	0.000134849	113	0.000695166
106	0.001451660	78	0.000089036	97	0.000547584
77	0.000897916	99	0.000670410	84	0.000924506
90	0.000293896	93	0.001154787	66	0.001023920
70	0.000956085	108	0.000685068	118	0.000577448
102	0.000061485	102	0.000010181	116	0.000133980
91	0.001076333	91	0.000023735	73	0.000509107
88	0.000609301	105	0.001064709	99	0.000814110
102	0.000349947	88	0.000878403	111	0.000280991
104	0.000147357	71	0.000809395	84	0.000281921
50	0.000878890	112	0.000551644	76	0.000025764
53	0.000746431	106	0.000595699	61	0.000282382
82	0.000666486	100	0.000000738	100	0.000227369
95	0.000542393	95	0.000362265	111	0.000617087
89	0.000006722	64	0.000591436	113	0.000357239
90	0.000429099	76	0.000077735	109	0.000018634
91	0.000438890	107	0.000595318	51	0.000383840
62	0.000336984	84	0.000069016	77	0.000404336
88	0.000335409	72	0.000510842	117	0.000071632
92	0.000530038	82	0.000088433	100	0.000177382
66	0.000251275	103	0.000167233	66	0.000183179
88	0.000129551	51	0.000204499	102	0.000211133
55	0.000128792	55	0.000084804	117	0.000381475
60	0.000315902	56	0.000421939	54	0.000199533
101	0.004566658	102	0.004204589	67	0.003615638

From 7 to 5
Time interval no. 0

Speed Probability

From 7 to 6
Time interval no. 0

Speed Probability

--------	-----------	--------	-----------
97	0.009700000	107	0.087500000
101	0.083086170	50	0.001003750
110	0.033748354	71	0.069729463
69	0.063937673	106	0.005303131
61	0.074071794	76	0.080300511
71	0.002868278	81	0.041740206
76	0.054211492	94	0.065226814
114	0.008547541	95	0.036160224
96	0.011521054	56	0.015509808
115	0.064711641	116	0.039257464
101	0.038108863	57	0.034947616
58	0.008054564	84	0.040348050
118	0.015656572	117	0.038058269
98	0.023238611	85	0.012101680
106	0.013374533	117	0.001125314
74	0.033473009	61	0.019253271
113	0.024792745	85	0.012373033
62	0.040893569	53	0.009001381
69	0.033343498	80	0.032888147
86	0.014506402	100	0.016941529
98	0.014483191	52	0.012864384
88	0.015248739	83	0.013528677
115	0.011463181	53	0.014388064
75	0.022653539	96	0.011867744
52	0.022261080	110	0.021528178
78	0.009223945	88	0.017064705
71	0.000379230	52	0.001174946
73	0.000807810	118	0.004727459
118	0.014997322	76	0.005613982
110	0.013795855	53	0.012400554
97	0.018740822	67	0.012230476
114	0.018470952	56	0.007420289
89	0.011601748	106	0.020621446
58	0.014670410	53	0.006372921
89	0.015664676	110	0.007697397
109	0.011753935	58	0.004293228
95	0.007164190	87	0.008103897
70	0.003905395	93	0.011710901
56	0.007989349	59	0.001166207
64	0.006964589	99	0.007322744
114	0.010030025	58	0.001627846
77	0.000210944	78	0.002970093
109	0.009557703	82	0.006968872
75	0.007741739	78	0.005868005
64	0.001152087	72	0.010429461
116	0.005281929	108	0.002280991
118	0.006755095	105	0.002223332
99	0.005547171	65	0.005231413
84	0.003864476	69	0.001401143
112	0.001723370	92	0.000570747
66	0.002837580	78	0.003255620
62	0.000286785	87	0.004141098
115	0.003004561	93	0.002820204
99	0.003694648	57	0.004985357
100	0.000075183	79	0.002033027
115	0.001408397	53	0.002527375
76	0.001042904	86	0.004420785
96	0.003073222	78	0.001025124
80	0.003491777	58	0.006565250
53	0.001981562	111	0.001931912
68	0.000749247	91	0.003859073
94	0.001344736	51	0.000055796
51	0.000503148	91	0.005686057
108	0.002166377	82	0.001839491
107	0.001249116	68	0.000745473
107	0.001196869	82	0.000026834
103	0.002340061	82	0.003524264
76	0.000925236	56	0.003969306
103	0.000775578	91	0.004536367
55	0.000582597	91	0.000686595
70	0.001257853	78	0.003458182

62	0.000385185	52	0.000310976
64	0.001257329	106	0.001976703
70	0.001357494	98	0.000826715
69	0.001665221	106	0.000288562
78	0.000900693	75	0.001342125
91	0.000476911	68	0.001060192
53	0.000014019	82	0.002615262
112	0.001067169	71	0.001510419
52	0.001003964	108	0.001941294
102	0.001183824	109	0.002001513
116	0.000305297	70	0.000670049
63	0.000523274	104	0.000098570
88	0.000479200	108	0.001958356
110	0.000257770	82	0.001268672
117	0.000092762	88	0.000284166
107	0.000712780	70	0.001375119
68	0.000563894	111	0.001345376
53	0.000346982	88	0.001554762
64	0.000289009	63	0.000087222
110	0.000071789	66	0.000340883
80	0.000604817	86	0.000262982
52	0.000154755	104	0.000093256
109	0.000278658	102	0.000208107
109	0.000225763	63	0.000367999
85	0.000005843	83	0.000093009
71	0.000179789	60	0.001204968
85	0.000171988	72	0.000774975
95	0.000489308	88	0.000876163
119	0.004996210	71	0.010728651

12. נספח ד' – פלט ריצה של אלגוריתם STDVRP

פלט אלגוריתם STDVRP עבור בעיה מס' 5 בקבוצת הבעיות מס' 17.

```
Heuristic method: Avrg., Running Num. 0
----- 14/09/2007 12:33:54  -----
New best solution:
0,1,5,3,4,0
0,7,2,6,0
Solution time: 69.6214106510297
-------------------------------
Heuristic method: Avrg., Running Num. 1
Heuristic method: Avrg., Running Num. 2
Heuristic method: Avrg., Running Num. 3
Heuristic method: Avrg., Running Num. 4
Heuristic method: Avrg., Running Num. 5
Heuristic method: Avrg., Running Num. 6
----- 14/09/2007 12:33:57  -----
New best solution:
0,1,2,5,3,4,0
0,7,6,0
Solution time: 68.7069607349939
-------------------------------
Heuristic method: Avrg., Running Num. 7
Heuristic method: Avrg., Running Num. 8
Heuristic method: Avrg., Running Num. 9
Heuristic method: Best, Running Num. 0
----- 14/09/2007 12:33:59  -----
New best solution:
0,4,1,7,2,0
0,6,5,3,0
Solution time: 66.5022356048659
-------------------------------
Heuristic method: Best, Running Num. 1
Heuristic method: Best, Running Num. 2
Heuristic method: Best, Running Num. 3
Heuristic method: Best, Running Num. 4
Heuristic method: Best, Running Num. 5
Heuristic method: Best, Running Num. 6
Heuristic method: Best, Running Num. 7
Heuristic method: Best, Running Num. 8
Heuristic method: Best, Running Num. 9
Heuristic method: Worst, Running Num. 0
Heuristic method: Worst, Running Num. 1
Heuristic method: Worst, Running Num. 2
Heuristic method: Worst, Running Num. 3
Heuristic method: Worst, Running Num. 4
Heuristic method: Worst, Running Num. 5
Heuristic method: Worst, Running Num. 6
Heuristic method: Worst, Running Num. 7
----- 14/09/2007 12:34:07  -----
New best solution:
0,4,1,7,2,0
0,6,5,3,0
Solution time: 66.3708542234846
-------------------------------
Heuristic method: Worst, Running Num. 8
Heuristic method: Worst, Running Num. 9
--- Best Solutions -----
3: 0,4,1,7,2,0 (L:3284, T:[65.10,39.74,27.65,32.84(80.50%)], Q:107)
5: 0,6,5,3,0 (L:2356, T:[46.90,28.20,19.80,33.66(81.00%)], Q:146)
Max capacty: 150
Solution time: 66.3708542234846
Algorithm running time: 14.421875(00:00:14.4218750)
```

13. English Summery

Vehicle Routing Problem (VRP) is a common name for a group of problems in which a set of routes has to be constructed for a fleet of vehicles. The vehicles start their routes at a depot, visit some customers, to whom they deliver goods, and return to the depot. The objective function for the vehicle routing problem is minimizing costs by finding optimal routes, which are usually the shortest routes.

The vehicle routing problem was first introduced by Dantzig and Ramser, who also provided a two stages heuristic algorithms for solving it. The first stage was constructing an initial solution, and at the second stage the initial solution was improved by merging routes to a number of groups. Clarke and Wright presented the Savings algorithm. This algorithm was also a two stages heuristic algorithm. The first stage was the construction of an initial solution, and at the second stage routes were merged together to new routes, based on saving gained by merging them. Gillett and Miller presented the Sweep algorithm for solving the vehicle routing problem. The Sweep algorithm divided the set of customers into a number of sub-sets. For each sub-set of customers, an optimal route was constructed by solving a traveling salesman problem. The Cluster method was introduced by Fisher and Jaikumar. In this algorithm the set of customers was divided to sub-set by solving a Generalized Assignment Problem (GAP). Similar to the Sweep algorithm, for each sub-set of customers, an optimal route is constructed by solving a traveling salesman problem.

In addition to the various heuristics developed for solving the vehicle routing problem, a number of algorithms based on meta-heuristics, which use simple heuristics for generating a space of basic solutions, and then search this space of solution for an optimal solutions using operations such as solutions merging, were also introduced. Bullnheimer presented an algorithm based on Ant System for solving the vehicle routing problem. Berger and Barkaoui also presented an algorithm for solving the vehicle routing problem, based on genetic algorithms. Other algorithms based on other meta-heuristics, such as Tabu Search, Deterministic Annealing, Simulated Annealing and so on, were also introduced.

Simultaneously to the development of heuristics and meta-heuristics, which provided an almost optimal solutions, a number of researchers developed extensions to the basic vehicle routing problem. The goal was to develop more realistic models, to adapt to the larger number of constraints of the real world. Among the various extensions of the vehicle routing problem we can find the Split Delivery Vehicle Routing Problem. In this extension, the constraint stating that each customer can be visited by only one vehicle does not exist. Burrows, for example, suggested a Savings based algorithm for solving the split delivery vehicle routing problem. Another extension developed is the vehicle routing problem with time windows. In this problem variant, each one or the customers has to be visited at a pre-specified time window. Solomon was the first to introduce the modification of a number of general algorithms, such as the Saving algorithm and the Sweep algorithms, for solving the vehicle routing problem with time windows.

In some cases, an organization has more than one depot from which he wishes to deliver goods. For such organizations, the Multi-Depot Vehicle Routing Problem was developed. Wren and Holliday presented a two stages algorithms for solving the multi-depot vehicle routing problem. The first stage was constructing an initial solution, using the Savings algorithm. The second stage was using various heuristics in order to improve the initial solution. Golden presented a Savings based algorithm for solving the multi-depot vehicle routing problem as well.

In the real world, in areas such as urban areas, the traveling time is dependant on both the distance between two customers and the time of day. Ignoring the fact that for some routes the traveling time changes throughout the day, we may get solutions that are far from the optimal solution. Malandraki developed two algorithms for solving the time dependent vehicle routing problem. The first algorithm was a greedy algorithm, and the second algorithm was a branch and bound based algorithm, suitable for small problems. Ichoua, Gendreau and Potvin, developed an algorithms for the time dependent vehicle routing problem as well.

A stochastic vehicle routing problem arouses when at least one of the problem's variables is random. Tillman suggested a solution based on the Savings algorithm for vehicle routing problem with stochastic demands when there are a number of depots. Stewart and Golden presented a CCP (constraints chance programming) model and two SPR (stochastic programming) models. For the SPR models, Stewart and Golden assign "punishment" for part of the route which have high a probability that lead to longer routes. Bertsimas presented a number of algorithms for the solution of the vehicle routing problem with stochastic customers, in which each customer that has a constant and known demand exists a certain probability that the customer needs to be reached.

The aim of this study is to develop a model for the stochastic time dependent vehicle routing problem as well as developing a heuristic algorithm for finding the optimal solution for the model, by finding a set of routes that have the minimal traveling time, taking into consideration the following properties: a) that for certain routes the traveling time varies during the day, and b) travel time is stochastic. As far as we know, such model does not exist in the literature.

The vehicle routing problem is a hard optimization problem, which cannot be optimally solved for large problems. Adding more constraints to the problem, complicates the problem, and makes it even harder to optimally be solved. Therefore, in most cases heuristics algorithms are used for solving the vehicle routing problem.

The algorithm presented in this study is based on the Saving algorithm, that was extended so that it will be able to solve the stochastic time dependent vehicle routing problem. There are a number of basic conditions that need to exist so that we will be able to use this algorithm. The conditions are:

1. The number of customer is constant.
2. The demand of each customer in known and constant
3. The demand of each customer is smaller than the capacity of each vehicle
4. The total demand of all customers is bigger than the capacity of each vehicle
5. All vehicles have the same capacity
6. All vehicles travel at the same time periods
7. The parts of day time are periodic
8. The number of time periods for each edge of the graph is constant
9. The length of each time period is equal
10. For each edge of the graph the length of the edge, the traveling time and there probabilities are given.

The saving algorithm on which the algorithm presented in this study is based upon is designed for a deterministic vehicle routing problem. We will be using a number of tools and techniques so that we will be able to solve the stochastic time dependent vehicle routing problem using the Savings algorithm.

The first tool that we use is transforming the problem from stochastic time dependent problem to a deterministic problem. The transformation from stochastic time dependent problem to a deterministic problem will be used for decision making for the Saving algorithm.

In this study we will use three methods for transforming the stochastic time dependent problem to a deterministic problem.

1. **Average calculation** – The average calculation will be made in two steps. The first step, for each time period we calculate the average velocity. The next step is the calculation of the total average, which is the avarge velocity of all time periods.
2. **Best value calculation –** The best value calculation is the actually finding the highest velocity in all time periods, regardless its probability.
3. **Worst value calculation** - The worst value calculation is the actually finding the lowest velocity in all time periods, regardless its probability.

The second tool we are going to use is simulation. Simulation is a technique which uses a computer to simulate process is a complicated reality. The aim of simulation is to represent certain characteristics of the behavior the system in order to receive forecast for the behavior of the system in different conditions.

In this study we will use simulation in order to calculate the traveling time of a given route.

The parameters used by the algorithms are:

n – number of customers

m – number of route pairs candidates for joining, and their actual saving is calculated by simulation

r – number repetitions of algorithm

We will represent the algorithm and show how the parameters are used.

The algorithm:

1. Construction an initial solution. The initial solution has n routes. Each route begins at the depot, passes through a single customer, and end at the depot. There are no two route that passes through the same customer.
2. For each pair of routes, we calculate the saving gained by merging the pair of routes to one route, using the deterministic values.
3. For m pair of routes, that have the highest savings values, and does not violate the problem's constraints, we calculate the actual savings using simulation.
4. The pair of routes that have the highest value of savings (calculated by simulation) is merged to one new route.

5. We repeat stages 2 to 4 as long as there are pair of routes with positive values of savings.

We repeat the algorithm three times, once for each calculation method.

An issue that has not been addressed in the original Savings algorithm is how to cope in situations in which for two pairs of routes receive the same saving value. In the original algorithm, in such situations, we randomly choose one of the pairs. Such strategy may lead to a number of solutions depending on the pair of routes that was chosen. For this reason, if we come across a situation were we need to randomly choose between two route pairs, we repeat the algorithm at a constant number of time, r, so that we will be able to choose between two different route pairs.

The complexity of the algorithm was calculated and found to be $\boldsymbol{O}(n^3m)$, which means that the most influential factors are the number of nodes in the graph and the number of route pairs candidates for joining, as well as their actual saving is calculated by simulation. Based on the complexity analysis, we expect that the running time of the algorithms will increase in linear proportion to the number of candidate merging route pairs and their saving will be calculated using simulation (m). But, in reality, based on observations, we found that the running time increase less than expected, and it also decreases as the number of customers increases. Similarly, the running time does not increase as the number of customer (n) increases, and was much less than expected. The reason for these two differenced is the conditions included in the algorithm, that does not allow going over all the options of merging routes.

Another important aspect is the quality of the result (how close the solutions of the algorithm to the optimal solutions) received by the algorithm.

The results are tested with a number of running conditions.

1. The entire problem's data are deterministic.
2. The entire problem's data are stochastic. For the stochastic problem we'll consider for following factors:
    a. The influence of the percentage of edges that act stochastically.
    b. The influence of the range of traveling time.
    c. The influence of number of probabilities for speed for each edge in a time unit.
3. The problem's data are time dependent and stochastic.

Analyzing the results the following conclusions were made:

1. Increasing the number of time periods in the problem increases that gap between the solution of both Savings algorithm and the suggested algorithm and the optimal solution
2. Increasing the number of stochastic edges increases that gap between the suggested algorithm's solution and the optimal solution
3. Increasing the number of probabilities for velocities in a time period for an edge increases that gap between both Savings algorithm and the suggested algorithm's solution and the optimal solution
4. The number of probabilities for velocities in a time period for an edge has a smaller effect than the number of stochastic edges (only for the suggested algorithm)
5. A smaller range of velocities enables the both algorithms to reach solutions that are closer to the optimal solution

Based upon the conclusions above, 20 problems were constructed which are characterized at a high range velocity, a high number of probabilities for velocities in a time period and a 100% edges that are stochastic.
From the above results, it seems that the algorithm in average deviates from the optimal solution of about 6%, while the original Saving algorithm deviates from the optimal solution of 17%.

In addition to the quality test for the algorithm problem solution, the impact of the number of route pairs that are dedicated to merge that their saving is calculated by simulation have also been tested. Based upon the study results ,the use of the third to the fifth candidate to merge give the most significant improvement for the algorithm results. The use of a higher number of candidates mostly improves the results, but the time dedicated to search the optimal result increases linearly to the number of candidates while the improvement is small.

This work was carried out under the supervision of Dr. Yuval Hadas and Prof. Konstantin Kogan, from the Interdisciplinary department of social sciences, logistics management, Bar-Ilan University.

Bar-Ilan University

Developing a Model for the Stochastic Time Dependent Vehicle Routing Problem

Oren Nahum

Submitted in partial fulfillment of the requirements for the Master's Degree of the faculty of social sciences, the interdisciplinary department of social sciences , logistics management, Bar-Ilan University

Ramat-Gan, Israel **2009**