



המחלקה למתמטיקה

Department of Mathematics

**פרויקט מסכם לתואר בוגר במדעים (B.Sc)
במתמטיקה שימושית**

ניווט המון על תחומים ממושקלים באמצעות שדות זרימה

אריאל קנטור

**Crowd Pathfinding On Weighted Domains Using
Flow Maps**

Ariel Kantor

Advisor:

Dr. Mark Berman

מנחה:

ד"ר מרק ברמן

Karmiel

כרמיאל

2017

תוכן עניינים	
1	תיאור הפרויקט..... 3
2	סימולציית המון במחשב..... 3
2.1	ניווט בגרף וייצוג סביבה רציפה..... 3
2.2	מחירי תנועה ואלגוריתם DIJKSTRA..... 4
2.3	התחמקות ממכשולים..... 5
2.4	התנהגות הכוון (STEERING BEHAVIOR)..... 5
2.5	שדות זרימה..... 7
3	הגישה המאקרוסקופית לתנועת המון..... 9
3.1	היסטוריה על הגישה המאקרוסקופית..... 10
3.2	גישת הרצף לתנועת הולכי רגל..... 11
3.3	מודל הרצף הדימני של HUGHES לזרימת הולכי רגל..... 11
3.4	המשוואה האייקונית..... 13
3.5	ניסוח מחדש של מודל HUGHES והקשר למשוואה האייקונית..... 13
3.6	FAST MARCHING METHOD לפתרון נומרי של המשוואה האייקונית..... 15
3.6.1	אלגוריתם Fast Marching..... 17
3.6.2	חישוב הפוטנציאל..... 18
4	אלגוריתם CONTINUUM CROWDS..... 21
4.1	המודל המתמטי ל-CONTINUUM CROWDS..... 21
4.1.1	צפיפות ומהירות..... 22
4.1.2	חישוב המסלול האופטימלי..... 23
4.1.3	שימוש באי-נוחות לחיזוי תנועה..... 23
4.2	המודל במחשב..... 23
4.2.1	חישוב הצפיפות והאי-נוחות לחיזוי תנועה..... 24
4.2.2	חישוב שדה המחירים..... 25
4.2.3	חישוב שדה הפוטנציאל..... 25
4.2.4	חישוב שדה השיפועים..... 27
4.2.5	תנועת הסוכנים..... 28
4.2.6	מניעת התנגשויות מיקרוסקופית..... 29
4.3	הרצות האלגוריתם..... 30
4.3.1	תיאור התוכנה..... 30
4.3.2	תופעות מתהוות..... 33
4.3.3	השפעת המשקלים על התנועה..... 34
4.3.4	דוגמאות..... 39
5	סיכום..... 41
6	רשימת מקורות..... 42

1 תיאור הפרויקט

הפרויקט עוסק בשאלה של איך ניתן לנווט המון (מספר רב מאוד של הולכי רגל) בשטח לא אחיד. נתאר מודל ואלגוריתם המאפשר לנווט המון בהתחשב בתנאי השטח בו ההמון נע וגם ובמצב התנועה של ההמון. הסוכנים נעים למטרות שלהם תחת השפעת הסביבה וגם הדדית מסוכנים אחרים.

יש שתי גישות עיקריות למודלים כאלה:

1. גישה מיקרוסקופית: בגישה זו מתייחסים לכל סוכן כפרט יחיד. עבור כל אחד מהם ניתן לכתוב משוואות תנועה, בדרך כלל משוואה דיפרנציאלית רגילה (מ"ר).

2. גישה מאקרוסקופית: בגישה זו מתייחסים לסוכנים כקבוצה. במודלים אלה לא ניתן להבחין בין הסוכנים ומתייחסים לתכונות של כל הקבוצה כגון הזרימה, המהירות הממוצעת והצפיפות של הסוכנים. אחר כך מיישמים את המודל לסימולציה שבה יש התייחסות גם לפרמטרים מאקרוסקופיים וגם להשפעות על כל סוכן בנפרד.

בפרויקט זה נתמקד בגישה המאקרוסקופית לתנועת המון. קודם ניתן סקירה היסטורית על המודל. אחר כך נראה מודל מתמטי המשתמש בגישה זו ונסביר כיצד ניתן להפוך אותו למודל המתאים להרצה וסימולציה במחשב. בסוף נראה אלגוריתם המתבסס על מודל זה שנותן פתרון הנדסי ודיסקרטי למודל.

על מנת לבדוק את פעולת האלגוריתם והתנועה המתקבלת, נכתבה גרסה שלו במנוע Unity.

2 סימולציית המון במחשב

ניווט במחשב הוא מציאה של דרך להגיע מנקודה אחת לאחרת במרחב אשר ניתן להפוך אותו לגרף. בדרך כלל מחפשים את מסלול הקצר ביותר בין שתי נקודות. תחום המחקר הזה בנוי ברובו על גבי אלגוריתם Dijkstra.

בהתאם למטרה של תוכנת הניווט יש לשים לב גם לכך שזמן ההגעה סביר ושתנועת הסוכנים מציאותית. לדוגמה הולכי הרגל במשחק מחשב לא אמורים ללכת אחורה או להסתובב במקום ללא סיבה.

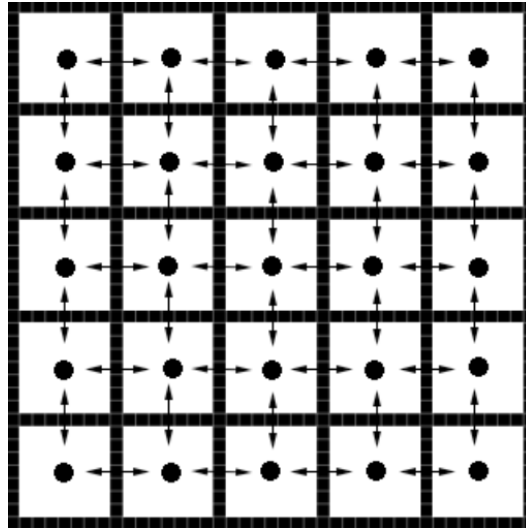
סימולציה של תנועת המון היא תהליך בו ממדלים תנועה של מספר רב מאוד של הולכי רגל במחשב. פותחו הרבה אלגוריתמים המאפשרים רמות שונות של דיוק בסימולציה בהתאם לצרכים. לדוגמה במשחקי מחשב ידרש שילוב של ביצועים טובים ביחד עם מידול תנועה מציאותי, בזמן שבסימולציה של מצבי חירום הדגש יהיה על הריאליזם.

2.1 ניווט בגרף וייצוג סביבה רציפה

אלגוריתמים לניווט בגרף כגון Dijkstra [1] משתמשים בגרפים מכוונים עם משקלים על הקשתות. גרף הוא ייצוג מופשט של קבוצה של אובייקטים מקושרים זה לזה. האובייקטים מכונים קודקודים (באנגלית Vertices) וקבוצתם מסומנת באות V , והקישורים ביניהם נקראים קשתות (באנגלית Edges) וקבוצתם מסומנת באות E . גרף עם קבוצת קודקודים V המקושרים בקבוצת קשתות E מסומן $G = (V, E)$.

ישנן מספר דרכים לייצג סביבה רציפה כגרף. לדוגמה ניתן להציב קודקודים בנקודות ציון וליצור גרף המכיל נקודות ציון וקשתות ביניהן. במקרה זה אפשר לייצג את האזורים מסביב לכל קודקוד באמצעות כל מיני צורות, כגון עיגולים, מלבנים, פוליגונים קמורים, או באמצעות מספר גרפים בעלי רזולוציות משתנות בהתאם לרמת המורכבות של אזורים. במקרה זה תהיה היררכיה של גרפים שמהם נבחר להשתמש בהתאם לצורך.

לוח משבצות הוא מקרה פרטי של גרף שבו כל משבצת היא קודקוד המחובר לשכונת המשבצות שלו והאזור מסביב למשבצת קבוע בגודלו. בזכות המבנה הפשוט והגמישות של גרף זה נעשה בו שימוש נרחב בסימולציות על מנת לבצע דיסקרטיזציה לסביבה רציפה בשביל שנוכל להשתמש באלגוריתמים לניווט בגרפים.



תמונה 1: לוח משבצות 4-שכנית הוא אלוויגרף שבו לכל צומת ישנם 4 שכנים (למעט הפינות).

כיוון שאנו עוסקים בייצוג של סביבה רציפה לא ניתן להתייחס למשבצות כקודקודים בגרף כפי שהם. מרכזי המשבצות הם אכן הקודקודים אבל יש צורך להוסיף מידע לגבי יחסי המרחק בין מרכזי המשבצות, השטח שהמשבצת מכסה בסביבה, כגון הגובה, סוג הקרקע (אספלט, דשא, בוץ וכו'...) והממדים של המשבצת.

הקשתות המחברות את המשבצות של הלוח בעצם מחברות בין המרכזים והן מכילות מידע אניסטרופי – מידע שמשתנה בהתאם לכיוון של משבצת בהתאם למשבצת אחרת. דוגמה אחת למידע כזה היא הפרש הגבהים במעבר בין משבצת למשבצת, או במילים אחרות השיפוע של השטח.

ניווט בגרף מאפשר לסוכן למצוא מסלול מהנקודה בה הוא נמצא כרגע לנקודת המטרה אליה הוא מנסה להגיע, ולשלב בחישוב מודל החלטה שמשקלל את היתרונות והחסרונות של תכונות שיש למסלול. באופן כללי הסוכן ינסה למזער את המרחק שהוא צריך לנוע, את הזמן שייקח לו להגיע למטרה ולהקטין את המאמץ שלו על מנת לשמר אנרגיה [2].

2.2 מחירי תנועה ואלגוריתם Dijkstra

בגרפים ללא משקל על הקשתות, המעבר מקודקוד לקודקוד נחשב לפעולה שלוקחת יחידה אחת של משקל. המחיר של לעבור מקודקוד לקודקוד צמוד הוא בדיוק 1 ומחירי כל הקשתות זהים. למרות זאת, ברוב מקרי הניווט ישנם מחירים שונים, לדוגמה במשחק המחשב Civilization לעבור דרך מישור או מדבר יעלה למשל 1, אבל לעבור דרך יער או גבעות יעלה 5.

אלגוריתם Dijkstra [1] הוא אלגוריתם לחיפוש המסלול הקצר ביותר שבנוי על מנת לפתור בעיה כזו. מחירי התנועה נלקחים בחשבון כאשר בודקים מקומות שאליהם מגיעים. האלגוריתם תמיד ימצא את המסלול הקצר ביותר.

האלגוריתם עובד על גרף ממושקל ולכן דורש את המידע של משקלי הקשתות שהם מחירי המעבר בין הקודקודים. על מנת לעבוד האלגוריתם גם דורש שהמשקלים יהיו חיוביים. מחיר להגיע לקודקוד מסוים יהיה סכום משקלי הקשתות שמובילות אליו מקודקוד המוצא. לכן המרחק מנקודת המוצא ניתן ע"י הנוסחה הבאה:

$$DistanceFromOrigin = \sum_{e \in \text{All edges in path}} EdgeCost_e$$

כאשר $EdgeCost$ הוא משקל הקשתות. הסכום נלקח על כל הקשתות במסלול. האלגוריתם מחפש למזער את הסכום הזה.

האלגוריתם בנוי בצורה הבאה [3]:

נקרא לקודקוד המקור ב-S. נסמן את כל שאר הקודקודים כך שנדע שעדיין לא ביקרנו בהם, ונגדיר את מרחקים מ-S כאינסוף. לולאת האלגוריתם:

- כל עוד נותרו קודקודים שלא ביקרנו בהם :
 - נקרא לקודקוד הנוכחי X (באיטרציה הראשונה זהו קודקוד המקור S) ונסמנו כקודקוד שביקרנו בו.
 - עבור כל קודקוד Y שהוא שכן של X:

- נעדכן את מרחקו של Y כך שיהיה שווה לערך המינימלי בין שני ערכים: בין מרחקו הנוכחי השמור לבין הסכום של משקל הקשת המחברת בין X לבין Y והמרחק בין S ל- X (במילים אחרות המרחק המעודכן מ- S ל- Y דרך X).

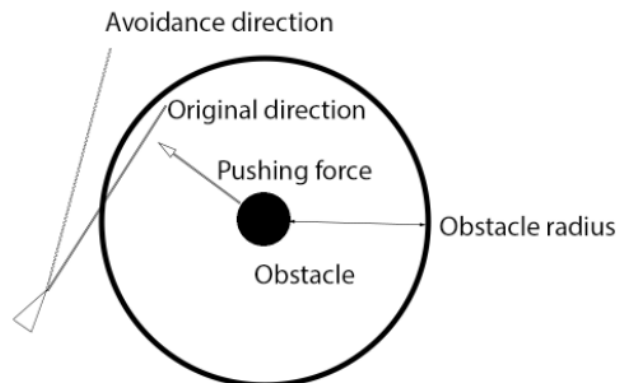
- בוחרים קודקוד X חדש בתור הקודקוד שמרחקו בשלב הזה מצומת המקור S הוא הקצר ביותר מבין כל הקודקודים בגרף שטרם ביקרנו בהם.

אנו מבקרים צמתים ומגדירים אותם צומת נוכחי רק כאשר המרחק אל צמתים אלה הוא מינימלי, כלומר אנו יודעים בוודאות שיש לנו מסלול קצר ביותר מהצומת הנוכחי לנקודת ההתחלה. בצורה כזו אנו ממשיכים לרוץ על הגרף עד שמגיע מצב שבו ניתן להגדיר את צומת היעד כצומת נוכחי. בנקודה זו סיימנו את החיפוש ועלינו פשוט לעקוב חזרה מהיעד להתחלה במסלול המחירים הכי נמוכים – מהיעד נלך לצומת השכן עם המחיר הנמוך ביותר, וכך נמשיך עד שנגיע להתחלה. ניתן לשמור את המסלול שקיבלנו והוא יהיה המסלול הקצר ביותר.

האלגוריתם לא מתייחס למרחק ליעד אלא רק למרחק מנקודת ההתחלה, ומתפשט מנקודת ההתחלה החוצה.

2.3 התחמקות ממכשולים

התחמקות ממכשולים מתחלקת לשני מקרים: גלובלית ולוקלית. הגירסה הגלובלית היא בדרך כלל חלק ממנגנון הניווט. גירסה זו דורשת ידע של מבנה הגרף ומאפשרת לסוכנים להתחמק ממכשולים סטטיים כמו קירות. בגירסה הלוקלית לסוכנים יש בדרך כלל רק ידע לגבי הסביבה שקרובה אליהם ולכן המידע בידם מאוד מצומצם. יש סוגים שונים של התחמקות לוקלית והנפוץ ביניהם משתמש בחשבון וקטורים בין הסוכן והמכשול על מנת לבצע התחמקות. במקרה שזהו מכשול ללא יכולת החלטית משלו, הסוכן יבצע התחמקות בעצמו. במקרה שאלה שני סוכנים במסלול התנגשות האלגוריתם ישתמש במיקומים היחסיים שלהם ובכיווני התנועה על מנת להסית אותם ממסלול התנגשות תוך כדי שמירה על המסלול שלהם.

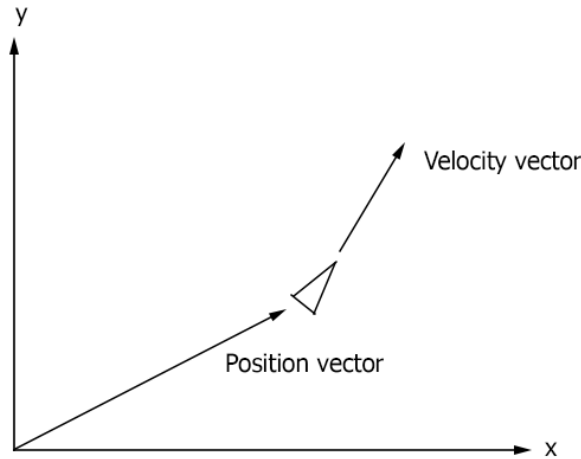


תמונה 2: דוגמה לצורה בה ניתן לגרום לסוכן להתחמק ממכשול. הסוכן מחשב אם המסלול שלו עובר ברדיוס ההשפעה של המכשול. אם כן, או מחשב כוח ההרחקה של המכשול שבאמצעותו מטים את וקטור התנועה של הסוכן.

2.4 התנהגות הכוון (Steering Behavior)

התנהגות הכוון מתארת איך סוכנים ינועו ויסתובבו כאשר הם מגיבים לשינוי בכיוון התנועה. התנהגות זו לוקחת בחשבון תכונות של הסוכן כגון מהירות מינימלית ומקסימלית, קצב סיבוב, תאוצה וכיוון. המטרה היא לאפשר לסוכנים לנוע בצורה מציאותית. דוגמה מפורסמת להתנהגות כזו היא Boids [4]. הסוכנים נשלטים על ידי כוחות וחישובם יחד מאפשר למצוא את וקטור המהירות שבו הסוכן ישתמש.

כיוון שמיקום, מהירות וכיוון הסוכן במחשב מיוצגים באמצעות וקטורים, יישום הכוחות במחשב גם בצורת וקטורים מהווה גישה טובה לבעיה.



תמונה 3: מיקום, מהירות וכיוון הסוכן כפי שהם מיוצגים במחשב.

בכל צעד חישוב, המיקום החדש מחושב ע"י שיטת אוילר בצורה הבאה:

$$newPosition = oldPosition + velocity$$

וקטור המהירות מחושב כך שתמיד יצביע בכיוון המטרה

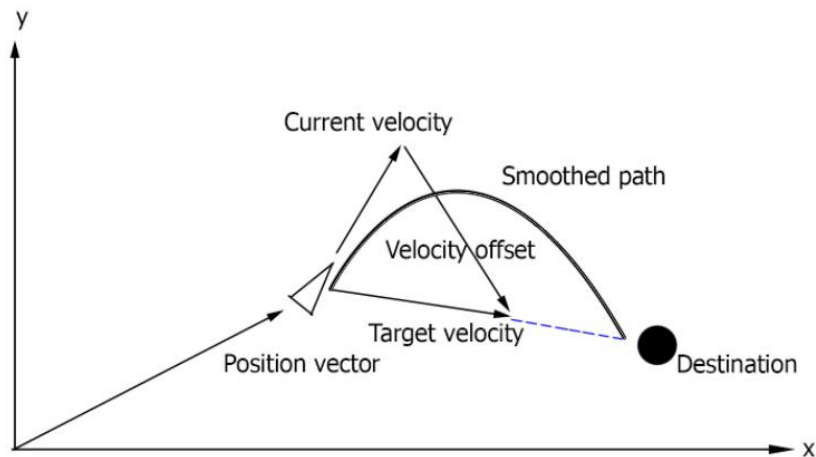
$$velocity = \frac{targetPosition - currentPosition}{\|targetPosition - currentPosition\|} * speedVal$$

כאשר $speedVal$ הוא ערך מהירות נתון כלשהו אשר מייצג את מהירות התנועה הנוכחית. במצב כזה כאשר הסוכן צריך לפנות, שינוי הכיוון מתבצע מייד – אין החלקה של התנועה מכיוון אחד לשני. מיותר לציין שצורת תנועה זו אינה מציאותית, ולכן יש צורך בכוחות נוספים שנקראים כוחות הכוון אשר יאפשרו לסוכן לשנות את וקטור המהירות באופן הדרגתי וחלק.

לשם כך מגדירים שני כוחות הממומשים כווקטורים שנקראים מהירות רצויה ($target\ velocity$) וכוח הכוון ($steering\ force$). הראשונה מהווה וקטור הכיוון בו הסוכן רוצה לנוע בסופו של דבר (החישוב של $newPosition$), והשני וקטור המצביע בכיוון שבו הסוכן צריך להסתובב. המהירות הרצויה היא אותה מהירות המחושבת בכיוון המטרה, וכוח ההכוון מוגדר בצורה הבאה:

$$steeringForce = \frac{targetVelocity - currentVelocity}{\|targetVelocity - currentVelocity\|} * turnRate$$

כאשר $targetVelocity$ הוא וקטור המהירות (והכיוון) הרצוי, $currentVelocity$ הוא וקטור המהירות (והכיוון) הנוכחי ו- $turnRate$ הוא מקדם בין 0 ל-1 המייצג את קצב הסיבוב (מהירות זוויתית).



תמונה 4: המסלול אשר הסוכן לוקח באמצעות התנהגות ההכוון.

לאחר החישוב אנו מעדכנים את המיקום של הסוכן עם ה- $steeringForce$ באמצעות שיטת אוילר. המסלול המתקבל בתמונה מדגים איך פעולת הסיבוב מחליקה את שינוי הכיוון.

2.5 שדות זרימה

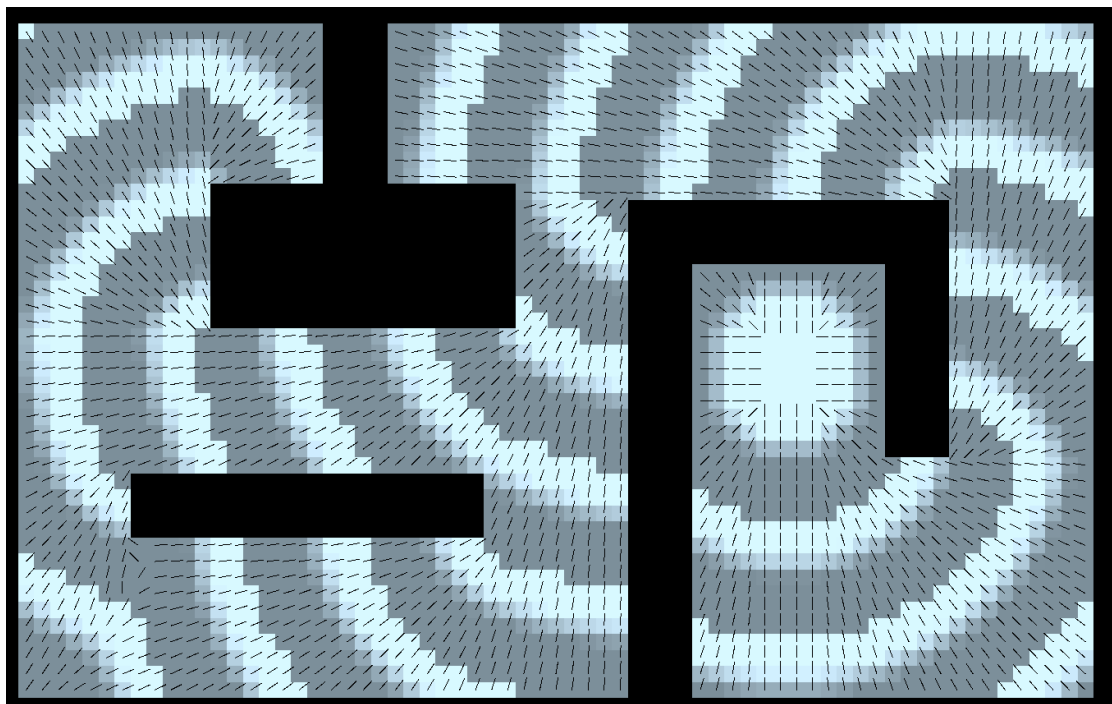
השימוש ב-Dijkstra שהוצג קודם לכן מאפשר לכל סוכן יחיד למצוא מסלול למטרה. צורה זו של ניווט מאפשרת שימוש בהתנהגות עצמאית עבור כל סוכן ולתת לכל סוכן מטרה שונה – משהו שמתאים מאוד לסימולציה מיקרוסקופית של קהל. אבל ישנם מצבים מסוימים בהם אלגוריתמים מבוססי Dijkstra (כמו A^* [5]) לא עומדים בדרישות:

- כאשר הגרף צפוף (מספר הקשתות גדול מאוד, עד כדי מספר הקודקודים בריבוע), למשל בלוח משבצות.
- כאשר ישנו מספר גדול מאוד של יחידות המחפשות מסלול לאותו יעד.
- כאשר הסביבה דינמית (משתנה מרגע לרגע).
- כאשר מיקום היחידות משתנה בכל רגע נתון – כאשר היחידות מושפעות מכוחות חוץ מהמסלול.

עבור כל אחד מהמקרים האלה יש פתרונות נפרדים: אם הגרף צפוף, מקביליות בין בקשות היחידות למציאת מסלול תקצר את זמן החישוב. אם ישנו מספר גדול של יחידות המחפשות להגיע לאותו היעד ניתן להשתמש באלגוריתם Dijkstra ללא נקודת עצירה. בצורה זו נמצא מסלולים קצרים ביותר ליעד מכל הקודקודים – ניווט באמצעות שדות זרימה משתמש בגירסה מורחבת של רעיון זה עבור תחום רציף. אם הסביבה דינמית אז ניתן להשתמש באלגוריתם (כגון Field-D* [6]) אשר מעדכן את המסלולים בצורה יעילה כאשר משהו בגרף משתנה. אם היחידות מושפעות מכוחות שלא קשורים למסלול אז צריך לעדכן את המסלול שלהן בכל פעם שהן זזות וזה גורם לצורך בהמון עדכונים למסלולי הסוכנים – ניתן לפתור את זה במידה מסוימת על ידי כך שנמנע שינוי של המסלול עד סף מסוים שיפעיל את העדכון במקרה שמצב היחידות השתנה מספיק.

הבעיה בפתרונות אלה היא שילוב ביניהם יכול להיות יקר מבחינת משאבים וקשה למימוש. במקרה של תחום רציף עם סביבה דינמית ועם מספר רב של סוכנים הפגיעה בביצועים תהיה משמעותית עד כדי כך שלא יתאפשר להשתמש בהם בזמן אמת.

שדה זרימה [7] (או שיפועים, גרדיאנטים, וקטורים) הוא מושג שבעבר הופיע בעיקר בסימולציות של תנועת נוזלים ובבעיות של התפשטות גלים. בשדות זרימה כל וקטור פונה לכיוון הצומת הקרוב ליעד, ומשתמשים בוקטור הזה בשביל להשפיע על וקטור המהירות של הסוכנים.

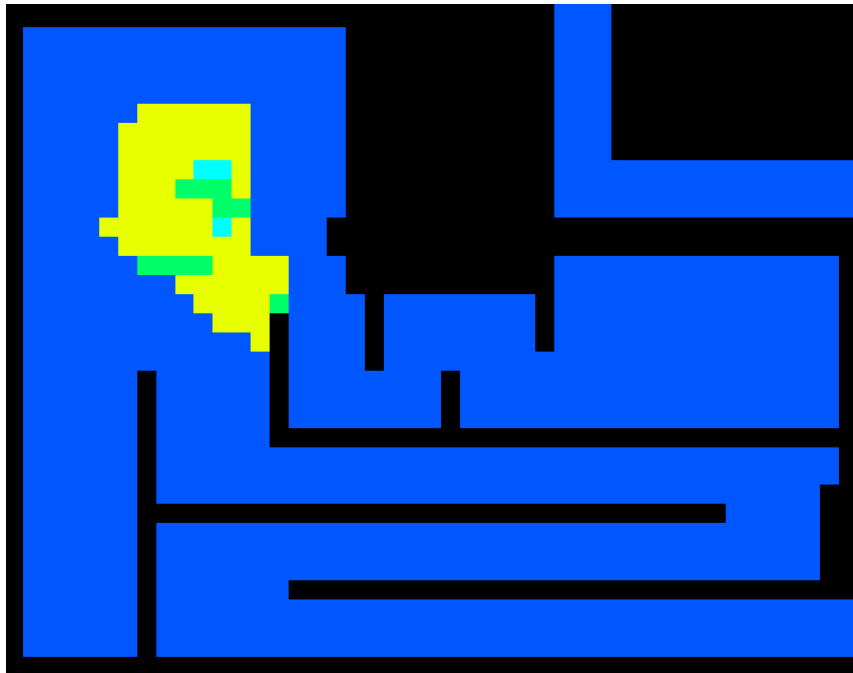


תמונה 5: שדה זרימה של אלגוריתם Continuum Crowds שיוסבר בפרק 4.

יש צורך לבצע דיסקרטיזציה לתחום הרציף כדי שהמחשב יוכל לפתור את הבעיות האלה ולשם כך נוצרים שלושה שדות. השדה הראשון הוא **שדה המחירים** (Cost Field) אשר מחזיק מידע שקשור למחיר המעבר בשטח, כגון צפיפות, מהירות ממוצעת, שיפוע הגובה ותנאי השטח (דשא, מדבר, יער...). השדה השני נקרא **שדה האינטגרציה או שדה הפוטנציאל** (Integration/Potential Field) ומחושב באמצעות שדה המחירים. השדה האחרון זה **שדה הזרימה או שדה השיפועים** עצמו ומשתמש בשדה הפוטנציאל כדי לחשב את התוצאה הסופית.

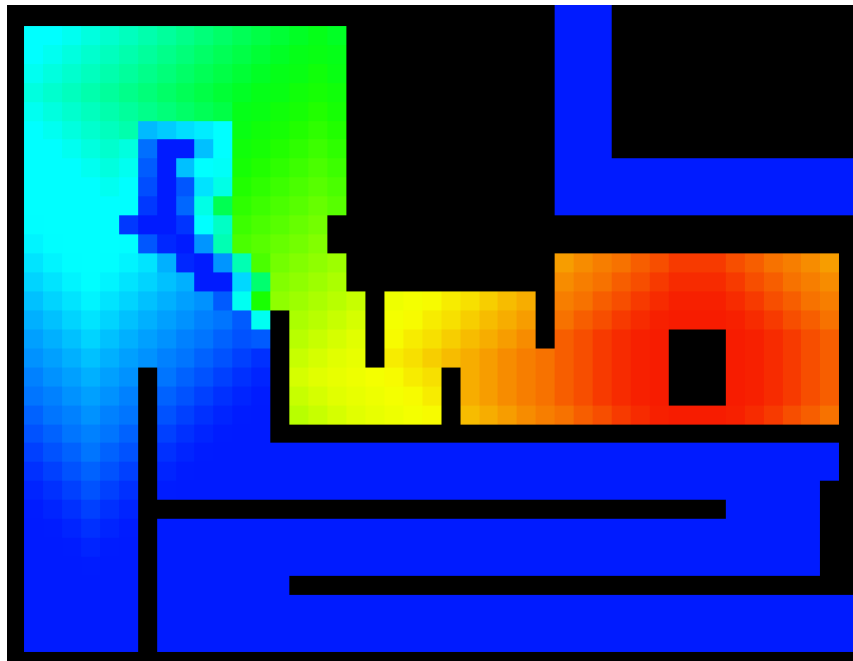
ישנם שלושה צעדים בחישוב שדה הזרימה:

1. ליצור שדה מחירים – שדה זה נוצר כחלק ממבנה ותכונות התחום ואפשרי שיושפע ממיקום ותנועת היחידות. כאשר שדה המחירים משתנה, צריך לחשב מחדש את שדות הפוטנציאל והזרימה.



תמונה 6: שדה המחירים עם מחיר גבוה מסומן בצבעים בהירים

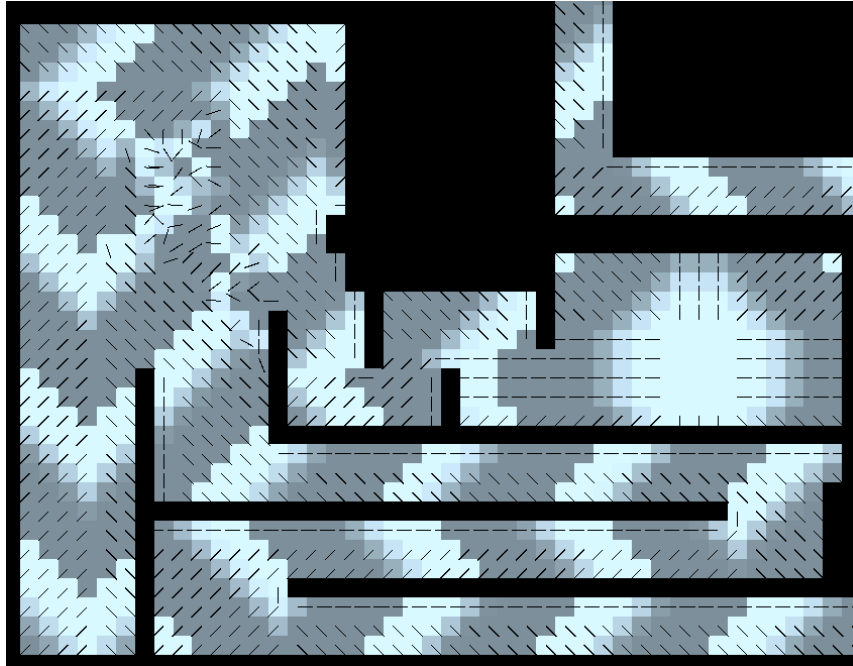
2. חישוב שדה פוטנציאל – זהו החלק הכי חשוב ויצירתו צורכת הכי הרבה עבודה חישובית. החישוב עובד כמו אלגוריתם Dijkstra. האלגוריתם יעצור כאשר חישב את המרחקים – או הפוטנציאל – לכל צומת בתחום מהמקור. שדה הפוטנציאל יכול לשמש גם לבדיקת נגישות, הרי לפי אלגוריתם Dijkstra אם לא עברנו בצומת מסוים, המרחק אליו יהיה אינסוף.



תמונה 7: שדה הפוטנציאל שבו הצבעים נעשים קרים יותר ככל שהפוטנציאל גדל

3. בניית שדה הזרימה – כאשר בונים את שדה הזרימה לוקחים את שדה הפוטנציאל ומשתמשים בו בכדי למצוא את הוקטורים בשדה. מחשבים את זה בכך שעוברים בכל צומת ומשווים את ערך הפוטנציאל בצומת הזה לערכים בצמתים

השכנים שלו. הוקטור בצומת יהיה מכיוון בכיוון הצומת השכן עם הפוטנציאל הכי נמוך. עבור לוח משבצות, שהוא המקרה הכי נפוץ של שימוש בשדה זרימה, אנו מחשבים את הפרשי הפוטנציאל על ציר ה-X והפרשי הפוטנציאל על ציר ה-Y, ובחרים את ההפרש הקטן יותר בכל ציר על מנת לבחור בכיוון הוקטור.



תמונה 8: וקטורים של שדה הזרימה ואזורי פוטנציאל קבוע של אלגוריתם Dijkstra (אזורים אלה מראים את צורת ההתפשטות של האלגוריתם בתחום)

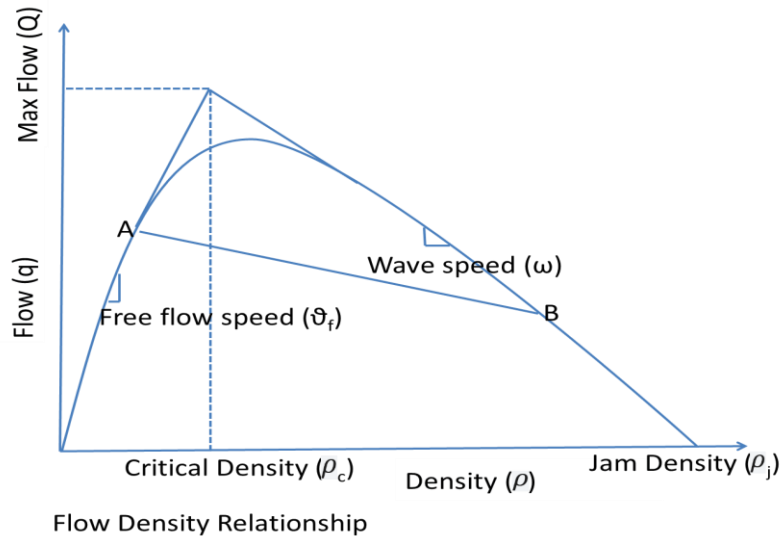
מתמטית שדה הזרימה הוא שדה גרדיאנטים (Gradient Field) שמקבלים משדה הפוטנציאל והסוכנים נעים בהתאם לשיפועים בשדה הזרימה.

3 הגישה המאקרוסקופית לתנועת המון

תנועת המון היא ענף במתמטיקה ובהנדסה אזרחית שבה חוקרים איך תעבורה (אנשים או רכבים) מתנהגת. המטרה היא להבין ולפתח מערכות תעבורה אופטימליות עבור מעבר יעיל באזורים כגון ערים. תעבורה מתנהגת בצורה מורכבת וקשה לחיזוי. כתוצאה מצורות חשיבה ותגובה שונות של בני אדם, תעבורה נוטה להראות תופעות כגון קיבוץ והתפשטות של גלי הדף בכיוונים שונים בהתאם לצפיפות.

ישנן שלוש תכונות עיקריות שבהן משתמשים במודלים לתנועה:

1. מהירות v : כיוון שלא ניתן למדוד מהירות של כל סוכן, משתמשים במהירות הממוצעת של סוכנים באזור נתון במשך זמן נתון.
2. צפיפות ρ : צפיפות מוגדרת כמספר הסוכנים ליחידת שטח. בהקשר לזרימה של המון שתי הצפיפויות הכי חשובות הן הצפיפות הקריטית ρ_c וצפיפות פקק ρ_j . הצפיפות המקסימלית המותרת תחת זרימה חופשית היא הצפיפות הקריטית, והצפיפות המקסימלית האפשרית בתנאים דחוסים היא צפיפות הפקק כפי שרואים בתמונה 9.
3. זרימה q : זרימה, או שטף (flux), מוגדרת כמספר הסוכנים העוברים נקודה כלשהי בכל יחידת זמן.



תמונה 9: דיווח בין זרימה לצפיפות [8,9,10]

על מנת להשתמש בדיאגרמה זו יש להניח מספר הנחות בסיסיות:

- הקשר בין צפיפות התעבורה ומהירותה הוא: ככל שיש יותר תעבורה, מהירותם תהיה קטנה יותר.
- בשביל למנוע צפיפות גבוהה ועל מנת לשמור על זרימה יציבה, מספר הסוכנים הנכנסים למערכת צריך להיות קטן או שווה למספר הסוכנים היוצאים מהמערכת.
- בצפיפות קריטית ובמהירות הקריטית המתאימה לה, מצב הזרימה ישתנה מיציב ללא יציב.

3.1 היסטוריה על הגישה המאקרוסקופית

מודלים מאקרוסקופיים מתייחסים להמון כגוף רציף בעל תכונות שאותן ממצעים כגון דחיסות ומהירות, ומשתמשים במשוואות דיפרנציאליות הנובעות מדינמיקת נוזלים כמו משוואת שימור המסה:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot q = 0$$

במאמר [9] משנת 1955 נמצא קשר פונקציונלי בין זרימה וריכוז עבור תנועת כלי רכב בכבישים סואנים. זהו המאמר הראשון בו נעשה שימוש במודל של גל קינמטי. החלק הראשון של המאמר [8] השתמש בתנועת מים כדוגמה לתיאוריה של גלים קינמטיים. בחלק השני הכותבים Lighthill ו-Whitham הרחיבו את התיאוריה לכבישים צרים וסואנים.

יש לציין כי מודל זה הוא חד-ממדי ומניח כי הכביש הומוגני ושטוח.

התיאוריה מתבססת על כך שבכל נקודה בכביש, הזרם q (המוגדר כרכבים ליחידת זמן) הוא פונקציה של צפיפות k (רכבים ליחידת מרחק), משמע מתקיים $q = f(\rho)$. מתוך כך מגיעים למסקנה כי שינויים בזרם מופצים אחורה במעיין 'גלים קינמטיים' שמהירותם היא שיפוע הגרף של הזרם כפונקציה של הריכוז. ממצא נוסף הוא שהנחות אלה מתאימות גם לתנועה של הולכי רגל במצב בעל צפיפות גבוהה. מתייחסים אל הזרם q , הצפיפות ρ , והמהירות הממוצעת הממושקלת לפי הצפיפות על-ידי המשוואה הבאה:

$$v = \frac{q}{\rho}$$

מכאן, מהירות נקודתית $c = \partial q / \partial \rho$ היא מהירות התפשטות הגל העובר דרך כלי הרכב, ואם נציב את המשוואה עבור מהירות ממוצעת ממושקלת כאן, נקבל את הקשר בין מהירות הגל לצפיפות המראה כי, בהנחה שהיחס בין הצפיפות למהירות התחבורה הוא שלילי, גם מהירות התפשטות הגל תרד ככל שהצפיפות תעלה:

$$c = \frac{\partial q}{\partial \rho} = \frac{\partial(\rho v)}{\partial \rho} = v + \rho \frac{\partial v}{\partial \rho}$$

ברור כי מתקיים $c \leq v$ (מהירות הגל קטנה-שווה למהירות תנועת כלי הרכב). עבור $\rho \in [0, \rho_c]$ הגל מתפשט בכיוון הנגדי לתנועה, משמע $c \geq 0$. עבור $\rho \in (\rho_c, \rho_j]$ הגל מתפשט בכיוון של תנועה, משמע $c < 0$. היחסים האלה מראים שמצב התעבורה במקום מסוים יכול להיות מושפע על-ידי מצב התעבורה בכיוון התנועה, או על-ידי מצב התעבורה נגד כיוון התנועה (מאחורי כלי הרכב במיקום הנוכחי).

ב-1956 יצא מאמר בשם [11] שבו מוצג מודל נוסף עבור כביש סואן המרחיב את המודל הקודם על-ידי המרת התעבורה לפונקציית התפלגות רציפה (זו גם הסיבה שמודל זה לא עובד עבור מספר מועט של כלי רכב). המודל מניח קשר ישיר בין מהירות הרכבים והצפיפות של הרכבים באזור, ולכן בהינתן צפיפות ρ ומהירות מקסימלית על הכביש a , המהירות V ניתנת על ידי:

$$V = a(1 - \rho)$$

אם חושבים על התעבורה כנוזל דחיס עם דחיסות התלויה במיקום ובזמן והמהירות הנ"ל, נוכל להשתמש במשוואת השימור של החומר

$$\frac{\partial \rho}{\partial t} + \frac{\partial(V\rho)}{\partial x} = 0$$

על-ידי הצבת משוואת המהירות במשוואת השימור בכדי לקבל נוסחה המקשרת בין דינמיקת נוזלים לבין תנועה של קבוצה דחוסה של אנשים:

$$\frac{\partial \rho}{\partial t} + a(1 - 2\rho) \frac{\partial \rho}{\partial x} = 0$$

שני מאמרים אלה בנו את מודל LWR (Lighthill-Whitham-Richards) לתנועת כלי רכב. מודל זה היווה את הבסיס למחקר מאקרוסקופי של תנועת קבוצות גדולות של כלי רכב והולכי רגל, והתייחס לתנועה זו כאל זרימה רציפה.

3.2 גישת הרצף לתנועת הולכי רגל

גישה זו מתייחסת לקבוצת אנשים כישות אחת, בכך שמחליפים מידע מדויק על מיקום האנשים בצפיפות, ומהירות האנשים בשדה זרימה המציג מהירות ממוצעת. תפיסה זו זהה לצורה בה התייחסו Lighthill ו-Whitham לתנועת כלי רכב בכביש סואן.

ישנם מספר הבדלים משמעותיים מהמודל הקודם: התנועה היא דו-ממדית ולכן הולכי רגל יכולים לשנות כיוונים בכל נקודת זמן באזור פתוח בהתאם לתנאי הדרך, והאזור הוא לא הומוגני ואינו בהכרח שטוח.

חקירת תנועה של הולכי רגל היא תחום חדש יחסית שכן התחיל בו מחקר מעמיק רק בכ-30 השנים האחרונות והיכולת למדוד את המאפיינים המאקרוסקופיים (מהירות, צפיפות וזרימה) של הולכי הרגל היה גורם משמעותי לכך.

נחקרו מספר נושאים, ביניהם התנהגות בבחירת מסלול. לשם כך מודלים רבים בחרו להשתמש באמת מידה שנקראת אי-נוחות על מנת למדל בחירה בין, למשל, מדרגות ודרגנוע, או הבדלים בגבהים. נמצא שהולכי רגל בוחרים מסלול לפי מספר פרמטרים כגון זמן ההגעה, המרחק, אי-נוחות במסלול (עלולה לגרום להם לבחור מסלול יותר ארוך או שלוקח יותר זמן), כיוון ומהירות הזרימה של הולכי הרגל מסביבם.

בשנות ה-90 המאוחרות התחילו להיבנות מודלים המבוססים על תנועה דו-ממדית – הולכי הרגל מסוגלים לנוע בכל כיוון באופן חופשי בתחום הדו-ממדי. גישה זו הועילה מאוד למידול וסימולציה של התנועה וההתנהגות שלהם בהקשר המאקרוסקופי של זרימת הולכי רגל. באמצעות שימוש בפונקציית מחיר, ניתן היה למדל בחירת מסלול כבעיית אופטימיזציה שבה הולכי הרגל יבחרו במסלול בעל המחיר הכי נמוך.

בהקשר של גישת הרצף (Continuum Approach), מודל Hughes מ-2002 [12] סיפק גישה דינמית לבעיה של זרימת הולכי רגל אשר מיוצגים כפונקציית צפיפות ונעים על גרף על-ידי מיזעור של מחיר התנועה המיידית למטרה. במילים אחרות המודל עונה על "עיקרון שיווי משקל משתמש" (העיקרון הראשון של וודרופ). הצפיפות מבוססת על חוק השימור ועיקרון המיזעור מבוסס על זמן ההגעה.

3.3 מודל הרצף הדינמי של Hughes לזרימת הולכי רגל

מודל זה מתייחס להמון כ"נוזל חושב" אשר נע במהירות המקסימלית האפשרית בכיוון של יעד משותף, כאשר תנאי הדרך נלקחים בחשבון. הולכי הרגל מעדיפים להימנע מאזורים צפופים והנחה זו משתתפת בשדה הפוטנציאל ששיפועו נותן את כיוון התנועה. נראה בהמשך כי ניתן לתאר את שדה הפוטנציאל כפתרון למשוואה האייקונומלית (Eikonal Equation) שאותו מחשבים באמצעות

המחיר שניתן על ידי צפיפות הקהל בכל נקודה, ונותן את המסלול האופטימלי ליעד. בכל נקודת זמן כל הולכי הרגל מסתכלים על שדה הפוטנציאל ונעים בכיוון הנגדי לשדה השיפועים שלו. הגישה הזו אנלוגית למודל LWR המאקרוסקופי לתנועת כלי רכב ומהווה קירוב טוב לתנועה זו.

התחום בו מתבצעת התנועה הוא אזור דו-ממדי עם קואורדינטות מיקום x, y , מכיל שלושה אזורים עיקריים מלבד האזור בו התנועה מותרת: אזור התחלתי בו נכנסים הסוכנים לתחום, אזור יעד בו הסוכנים עוזבים את התחום, ואזורים בהם התנועה אסורה (קירות). הצפיפות היא פונקציה של מקום וזמן $\rho = \rho(x, y, t)$, ונסמן $\vec{v} = \vec{v}(x, y, t)$ וקטור המהירות הממוצעת של הסוכנים בכל נקודה בקואורדינטות x, y , כאשר $v_1 = v_1(x, y, t)$ הוא רכיב המהירות הממוצעת בכיוון ציר ה- x ו- $v_2 = v_2(x, y, t)$ הוא רכיב המהירות הממוצעת בכיוון ציר ה- y . נשים לב שכיוון שעוסקים במספר סוכנים גדול מאוד, הסטייה מהערך הצפוי בצפיפות והמהירות היא זניחה ולכן ניתן להשתמש בערכים הממוצעים של המהירות והצפיפות בכל נקודה.

כפי שהיה במודל LWR עבור כלי רכב, ניתן להתייחס לתנועת הסוכנים כזרימה ולהשתמש במשוואת הרצף, שהיא זהה למשוואת שימור המסה:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0$$

$$\text{כאשר } \nabla \cdot (\rho v) = \frac{\partial(\rho v_1)}{\partial x} + \frac{\partial(\rho v_2)}{\partial y}$$

ישנן שלוש הנחות יסוד לגבי תנועה של הולכי רגל המאפשרות לבנות את המודל:

1. מהירות הולכי הרגל תלויה בצפיפות ובזרימה סביבם. ז"ל המהירות תלויה בתנועת הסוכנים מסביב לכל סוכן יחיד. מודל LWR גם השתמש בהנחה זו בתיאור תנועת כלי הרכב, ובמהלך השנים מספר רב של מחקרים אישר את ההנחה בהקשר של תנועת הולכי רגל.
2. קיים שדה פוטנציאל המהווה סוג של חוש המכוון למטרה בו משתמשים הסוכנים כך שנעים בכיוון מאונך לקווים של פוטנציאל קבוע (כיוון שאין יתרון בלנוע על קווי פוטנציאל קבוע), וגם עבור כל זוג סוכנים במקומות שונים בעלי אותו פוטנציאל לא ינסו להחליף מקום כיוון שאין יתרון בלעשות זאת. הנחה זו תלויה בכמה מאזור התנועה גלוי לסוכנים ולכן לא מתאימה לתנועת כלי רכב, שכן נהיגה היא מצב בו קשה מאוד לראות מעבר לרכבים מסביב.
3. הולכי רגל מחפשים את המסלול שממזער את זמן ההגעה המשוער שלהם אבל ממתנים התנהגות זו כדי להמנע ממצבים של צפיפות גבוהה מאוד. הנחה זו מתאימה רק כאשר יש לסוכנים יעד כלשהו ובנוסף מניחים כי המיתון הזה פריד כך שמימזער זמן התנועה הוא פונקציה של הצפיפות.

מהנחה 2 מגדירים פונקציית פוטנציאל $\phi = \phi(x, y, t)$ ומניחים שהולך רגל בנקודה (x, y) ובזמן t ינוע במאונך לקווי הפוטנציאל הקבוע:

$$\hat{\phi}_x = -\frac{\frac{\partial \phi}{\partial x}}{\sqrt{\left(\frac{\partial \phi}{\partial x}\right)^2 + \left(\frac{\partial \phi}{\partial y}\right)^2}} = -\frac{\frac{\partial \phi}{\partial x}}{\|\nabla \phi\|}, \quad \hat{\phi}_y = -\frac{\frac{\partial \phi}{\partial y}}{\sqrt{\left(\frac{\partial \phi}{\partial x}\right)^2 + \left(\frac{\partial \phi}{\partial y}\right)^2}} = -\frac{\frac{\partial \phi}{\partial y}}{\|\nabla \phi\|}$$

או במילים אחרות בכיוון הנגדי לשיפוע כאשר $\hat{\phi}_x, \hat{\phi}_y$ הם מקדמי התנועה (קוסינוסים) בכיוונים x ו- y . גודל המהירות הוא סקלר התלוי בצפיפות המקומית $u = u(\rho(x, y, t), x, y) = u(x, y, t)$ ומהנחה 1 מקבלים את גדלי המהירויות בכל אחד מהצירים:

$$v_1 = u \hat{\phi}_x, \quad v_2 = u \hat{\phi}_y$$

מהנחה 3 ניתן להסיק שאם ישנם שני סוכנים שנמצאים באותו פוטנציאל, אז לאחר זמן מסוים הפוטנציאל שלהם עדיין יהיה שווה והסיבה לכך היא שהזמן הוא מדד לפוטנציאל (גם מהנחה 3). לכן המרחק בין פוטנציאלים שונים הוא יחסי למהירות התנועה ולא תלוי במיקום ההתחלתי על קו פוטנציאל קבוע. מתוך כך מקבלים את היחס:

$$\frac{1}{\|\nabla \phi\|} = g \|v\| = g \sqrt{v_1^2 + v_2^2} = gu$$

כאשר $g = g(\rho(x, y, t), x, y) = g(x, y, t)$ הוא גורם המאפשר אי-נוחות בתנאי צפיפות גבוהים ושווה למרחק שסוכן יסכים לנוע כדי להימנע מאותם תנאים, והפונקציות u ו- g עונות על התנאים הבאים:

1. $u(0)$ סופי – המהירות ההתחלתית צריכה להיות סופית.

$$2. \quad u(\rho_{max}) = 0 \text{ כאשר } \rho_{max} \text{ זו צפיפות הפקק.}$$

$$3. \quad -\frac{\partial u(p)}{\partial p} \leq 0 \text{ הקשר בין המהירות לצפיפות הוא כך שכל שהצפיפות גדולה יותר, המהירות קטנה יותר.}$$

$$4. \quad \frac{\partial g(p)}{\partial p} \geq 0, \quad g(\rho) \geq 1, \text{ אי הנוחות גודלת עם הצפיפות.}$$

המשוואות הנ"ל מהוות מודל Hughes המתאר תנועת המון כאשר מתייחסים להמון הזה כישות אחת. על מנת לענות על דרישות הידע של הנחה 2, יש להניח שהסוכנים הגבוהים יכולים לראות את כל השטח באופן ברור והסוכנים הנמוכים מקבלים מידע מלא לגבי מצב האזור מהסוכנים הגבוהים (במילים אחרות לכל הסוכנים יש ידע מלא לגבי מצב התנועה באזור). המודל לא מצפה מהסוכנים לנבא את השינוי בתנאים אלא פשוט להגיב לשינוי בהם. את פונקציות המהירות והאי-נוחות יש לבחור לבד ובחירה חכמה תתן תנועה יותר מציאותית.

דוגמה פשוטה לפונקציית מהירות לינארית היא

$$u(\rho) = A - B\rho$$

כאשר A ו-B הם קבועים חיוביים המהווים מהירות מקסימלית ומקדם צפיפות בהתאמה. מתוך משוואה זו והקשר שניתן במודל LWR הזרימה q ניתנת ע"י:

$$q = \rho(A - B\rho)$$

$$\text{כאשר ערכו המקסימלי הוא } \frac{A^2}{4B} \text{ עבור } \rho = \frac{A}{2B}.$$

במצב בו המהירות היא אניסטרופית (תלויה בכיוון) נשנה את הגדרת פונקציית המהירות כך:

$$u(\rho(x, y, t), \hat{\phi}_x, x, y) = u(x, \hat{\phi}_x, y, t)$$

כאשר $\hat{\phi}_x$ קוסינוס הזווית על ציר ה-x. החישובים שמגיעים למשוואות הגדרת המודל לא משתנים. ישנם מצבים רבים בהם דרושה מהירות אניסטרופית אבל המודל לא מרחיב על כך.

3.4 המשוואה האייקונומלית (Eikonal Equation)

המשוואה האייקונומלית היא משוואה דיפרנציאלית חלקית ולא לינארית שמופיעה בעיקר בבעיות הקשורות להתפשטות גלים. למשוואה יש שימוש במספר תחומים והרלוונטי ביניהם הוא פתרון של בעיות מציאת המסלול הקצר ביותר בסביבה רציפה.

המשוואה היא מהצורה (בדו-מימד)

$$\|\nabla\phi(x, y)\| = c(x, y)$$

כאשר $c(x, y)$ הוא מחיר התנועה מנקודה (x, y) ונקרא גם 'רמת האיטיות' (slowness) של הגל. משמעות משוואה זו היא שאורך וקטור השיפוע של הפוטנציאל בכל נקודה בתחום הנבדק שווה למחיר באותה נקודה שזה אומר שבאזורים בהם המחיר גבוה הפוטנציאל יגדל מהר יותר מאשר באזורים בעלי מחיר נמוך. בהקשר של בעיות מציאת המסלול הקצר ביותר ניתן להתייחס לפוטנציאל כזמן ההגעה לכל נקודה של חזית הגל היוצא מהיעד ומתפשט בכל התחום. באזורים בהם מחיר התנועה גבוה לגל יותר קשה לעבור ולכן יקח לו יותר זמן לעבור דרכם, וזמן ההגעה יגדל. הצורה המוצגת כאן היא גרסה אניסטרופית של המשוואה ולכן חזית הגל יכולה להתקדם במהירויות שונות מכל נקודה לסביבתה, בגרסה האיזוטרופית מחיר התנועה דרך נקודה לא תלוי בכיוון התנועה דרכה ולכן ניתן להתייחס לגרסה זו כמקרה פרטי. ניתן לראות את התחום כקבוצת רמות (Level Set) – כל קבוצה מכילה נקודות בעלי אותו ערך פוטנציאל. דבר זה דומה לקווי מתאר של מפת גובה ומסביר טוב איך סוכנים נעים "במורד הגבעה" בניגוד לכיוון השיפוע.

בחלק הבא של פרק זה נראה בין השאר שהמשוואה האייקונומלית עונה על עיקרון שיווי משקל משתמש ובכך ניתן להשתמש בה למציאת המסלול האופטימלי. ישנה הוכחה נוספת לכך ב-[15] אשר מוכיחה כי פתרון של משוואה זו מאפשר למצוא את המסלול הקצר ביותר בין שתי נקודות בתחום, וספציפית על גרף בצורת לוח משבצות במחשב.

3.5 ניסוח מחדש של מודל Hughes והקשר למשוואה האייקונומלית

כעת נראה איך ניתן להעביר את המודל לצורה נגישה מבחינה נומרית. כפי שצוין קודם מודל Hughes לתנועת המון מתייחס להמון כישות אחת המבססת את תנועתה על משוואת הרצף ומתאר שדה פוטנציאל ואיך סוכן ישתמש בו על מנת לנווט. המודל נותן בסיס תיאורטי טוב לסימולציה של תנועה, אבל לא מתאר באופן ברור איך ניתן ליצור את השדה ועל מנת להשתמש במודל בסימולציה ממוחשבת צריך קודם כל להסביר איך ליצור אותו. בחלק הזה נראה ניסוח מחדש של המודל כך שהוא מקיים את

עיקרון שיווי משקל משתמש של וורדרופ. דבר זה יסביר למה המודל נותן פתרון אופטימלי וגם יראה את הקשר בינו לבין המשוואה האייקונומית שעליה נסביר בהמשך.

ג'ון וורדרופ [13] היה מתמטיקאי אנגלי שבעבודתו הציג שני עקרונות המגדירים שיווי משקל ברשתות תנועה ומציגים שתי הנחות התנהגות לגבי מיזעור עלות כוללת של תנועה ברשת, כאשר הראשון ביניהם והרלוונטי מתאר פיזור נתיבי נסיעה על פני נתיבים חלופיים כתוצר של צפיפות. הוא קובע כי:

עבור כל רכב יחיד, זמני הנסיעה דרך כל מסלול בשימוש (מסלול אופטימלי) הם קטנים או שווים לזמני הנסיעה דרך כל מסלול שאינו בשימוש (לא אופטימלי). במילים אחרות הסוכנים מחפשים למזער את מחיר התנועה המיידית ליעד.

זרימות תנועה אשר מקיימים את העיקרון הנ"ל נחשבים ב"שיווי משקל משתמש", כיוון שכל משתמש בוחר במסלול האופטימלי.

לתזכורת בפרק 3.3 הצפיפות היא פונקציה של מקום וזמן $\rho = \rho(x, y, t)$. הוקטור $\vec{v} = \vec{v}(x, y, t)$ הוא וקטור המהירות הממוצעת של הסוכנים בכל נקודה בקואורדינטות x, y , כאשר $v_1 = v_1(x, y, t)$ הוא רכיב המהירות הממוצעת בכיוון ציר ה-x ו- $v_2 = v_2(x, y, t)$ הוא רכיב המהירות הממוצעת בכיוון ציר ה-y, הפונקציה $\phi = \phi(x, y, t)$ פונקציית פוטנציאל כאשר $\hat{\phi}_x, \hat{\phi}_y$ הם מקדמי התנועה בכיוונים x ו-y והם תמיד בכיוון הנגדי לשיפוע. גודל המהירות הוא סקלר התלוי בצפיפות המקומית $u = u(\rho(x, y, t), x, y) = u(x, y, t)$.

הוכחה זו הופיעה ב-[14] משנת 2009 ומסבירה את הקשר בין המודל למה שנקרא המשוואה האייקונומית ושניתן להשתמש במשוואה זו על מנת לחשב את המסלול הקצר ביותר בין כל שתי נקודות בתחום נתון. נסמן $\vec{q}(x, y, t) = (q_1(x, y, t), q_2(x, y, t))$ וקטור הזרימה בכל נקודה באזור ההליכה ולפי משוואת הקשר בין המהירות, הצפיפות והזרימה:

$$q_i = \rho v_i, \quad i = 1, 2$$

$$\|q\| = \sqrt{q_1^2 + q_2^2} = \sqrt{v_1^2 + v_2^2} * \rho = u\rho$$

אם נציב את הזרימה במשוואת השימור נקבל

$$\frac{\partial \rho}{\partial t} + \nabla \cdot q = 0$$

$$\nabla \cdot q = \frac{\partial q_1}{\partial x} + \frac{\partial q_2}{\partial y}$$

נגדיר פונקציה $c = c(\rho(x, y, t), x, y) = c(x, y, t)$ פונקציית מחיר מקומי המתארת את המחיר של מעבר יחידת אורך אחת בזמן t ותלויה בצפיפות באזור ובזמן.

ניזכר שלפי הנחה 2 הסוכנים נעים בכיוון הנגדי לשיפוע הפוטנציאל. לכן על מנת שהסוכנים יבחרו במסלול שמזער את מחיר התנועה למטרה נדרוש שמתקיים:

$$c \frac{\vec{q}}{\|\vec{q}\|} = -\nabla \phi$$

או במילים אחרות נדרוש שהמחיר משתנה הכי מהר בכיוון הנגדי לשיפוע הפוטנציאל. מכאן ניתן לראות שווקטור הזרימה q מקביל ומנוגד לשיפוע הפוטנציאל $\nabla \phi$, ובנוסף מתקיים:

$$\|\nabla \phi\| = c$$

משוואה זו נקראת המשוואה האייקונומית, נראה שהיא מקיימת את עיקרון שיווי משקל משתמש.

עבור כל מסלול p מנקודה כלשהי (x, y) למטרה (x_0, y_0) בו הסוכנים בוחרים להשתמש נבצע אינטגרציה של המחיר המקומי על גבי המסלול כדי לקבל את המחיר הכולל להגעה למטרה:

$$\Pi_p(x, y, t) = \int_p c(x, y, t) ds = \int_p c(x, y, t) \frac{\vec{q}}{\|\vec{q}\|} \cdot \vec{ds} = - \int_p \nabla \phi \cdot \vec{ds} = \phi(x, y, t) - \phi(x_0, y_0, t)$$

כאשר $\frac{\vec{q}}{\|\vec{q}\|}$ וקטור יחידה המקביל לכיוון ההתקדמות האופטימלי. קיבלנו שהמחיר הכולל בכל נקודת זמן אינו תלוי במסלול שאותו בוחרים כל עוד זהו מסלול אופטימלי. כעת, עבור כל מסלול $\tilde{p}$ שלא נמצא בשימוש מתקיים:

$$\Pi_p(x, y, t) = \int_p c(x, y, t) ds \geq \int_p c(x, y, t) \frac{\vec{q}}{\|\vec{q}\|} \cdot \vec{ds} = - \int_p \nabla \phi \cdot \vec{ds} = \phi(x, y, t) - \phi(x_0, y_0, t)$$

אי-השוויון קורה כי במסלול שאינו בשימוש וקטור הזרימה אינו מקביל ל- $\vec{ds}$ ולכן $\frac{\vec{q}}{\|\vec{q}\|} \cdot \vec{ds} > ds$. לכן עבור מסלולים שאינם בשימוש המחיר המידי הכולל גדול או שווה לזה של המסלולים שכן בשימוש. אם אנו מגדירים את הפונאציאל ביעד כ-0 אז נקבל שהפונאציאל שווה למחיר בכל נקודה, ולכן ניתן לפרש אותו כמחיר המידי הכולל למטרה. במילים אחרות המשוואה האייקונולית מספקת את התנאים לשיווי משקל משתמש וגם ניתן להשתמש בה על מנת למצוא את המסלולים האופטימליים. אם נבחר פונקציית מחיר ספציפית:

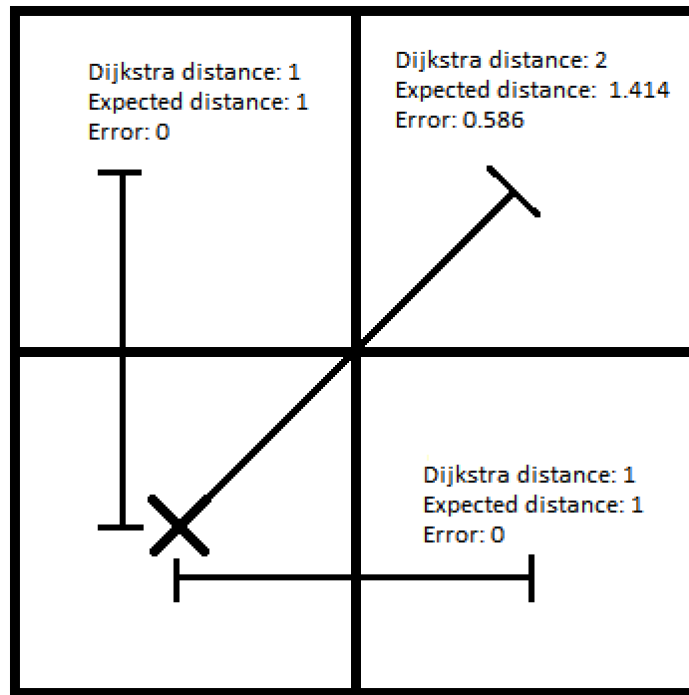
$$c = \frac{1}{ug}$$

ונציב במשוואה $c \frac{\vec{q}}{\|\vec{q}\|} = -\nabla \phi$ נקבל את המודל המקורי של Hughes ובכך נוכיח כי המודל גם מקיים את העיקרון ולכן ניתן להתייחס למסלול המתקבל במודל כזה הממזער את זמן ההגעה המידי.

3.6 Fast Marching Method לפתרון נומרי של המשוואה האייקונולית

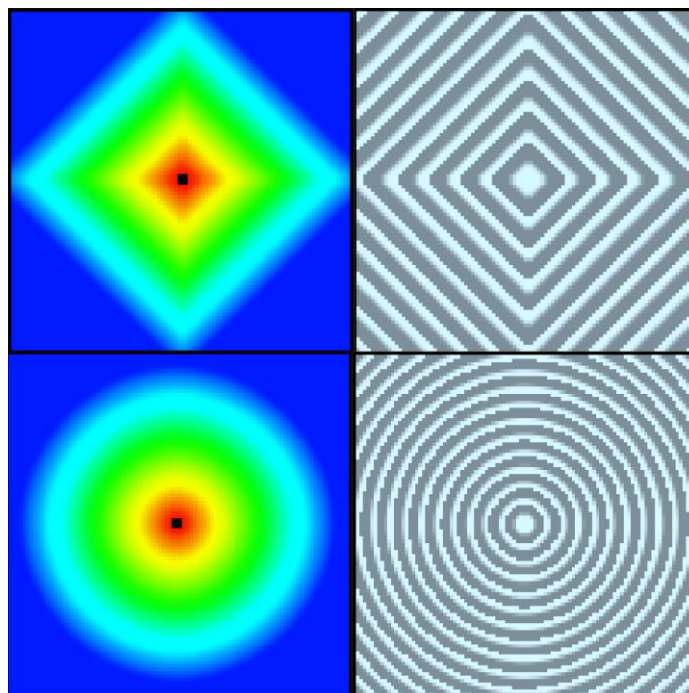
החלק הראשון של הפרק הסביר כיצד ניתן להשתמש בשדות זרימה על מנת לנווט סוכנים. בחלק השני הראנו שבאמצעות ניסוח מחדש של מודל זרימת הולכי הרגל של Hughes אפשר להגיע למשוואה האייקונולית (שעליה הסברנו בקצרה בחלק השלישי) ושניתן להשתמש במשוואה למציאת מסלול אופטימלי עבור הסוכנים. בחלק זה נסביר על אלגוריתם נומרי לפתרון המשוואה האייקונולית שהוא הפונאציאל בכל נקודה שממנו ניתן לחשב את שדה השיפועים, ובכך ליצור שדה זרימה במחשב.

השיטה שייכת למשפחת שיטות שנקראת Level Set Methods שבהן משתמשים במידע בחזית של גל מתקדם על מנת לחשב את כיוון ההתקדמות שלו ולאחר מכן את זמן ההגעה לכל נקודה בחזית. ברמה הכללית האלגוריתם מתאר התקדמות גל מעגלי מנקודת התחלה כלשהי כאשר בכל נקודה רדיוס הגל והעקמומיות שלו משתנה בהתאם לתנאי השפה המוגדרים על-ידי פונקציית המחיר. השיטה המתוארת כאן עובדת על לוח משבצות. כל משבצת מחוברת לשכנותיה מצפון, דרום, מזרח ומערב בגרף שכנויות. מבחינה זו ומבחינת סדר הפעולות היא זהה לאלגוריתם Dijkstra שתואר מוקדם יותר, וההבדל נובע מצורת החישוב של המחיר עבור כל קודקוד בגרף. על מנת לבצע סימולציה טובה צריך רמת דיוק מסוימת ו-Dijkstra עבור לוח משבצות בשיטת 4 שכנים סובל מדיוק נמוך בגלל שגיאות קירוב.



תמונה 10: גודל השגיאה באלגוריתם Dijkstra 4-כיווני על לוח משבצות

הסיבה לכך היא צורת ההתקדמות של הגל באלגוריתם Dijkstra ובזמן שאפשר לקחת שכונה 8-כיוונית כדי לשפר את הדיוק, תמיד תישאר שגיאה שמעותית יחסית בנוסף לפגיעה בביצועים שכן במצב זה בודקים 8 שכנים לכל צומת. לעומת זאת באמצעות אלגוריתם Fast Marching השגיאה קטנה משמעותית למרות שהתפשטות הגל היא 4-כיוונית. בנוסף לכך, בזמן שב-Dijkstra אותה שגיאה עלולה לגדול ככל שהרדיוס גדל כתוצאה ממידע לא מספק בחזית, ב-Fast Marching משתמשים ביותר מידע בחזית כדי למזער את השגיאה בכל מצב, או במילים אחרות האלגוריתם משמר את צורתו המעגלית של הגל הרבה יותר טוב מאשר Dijkstra. התמונה הבאה ממחישה את ההבדל בין שני האלגוריתמים. ב-Dijkstra צורת ההלום מראה איך הגל מתפשט ושהשגיאה גדלה ככל שהרדיוס גדל, וב-Fast Marching ניתן לראות איך צורת הגל מעגלית בכל רדיוס וכך שומרת על שגיאה נמוכה. זה מסביר למה השיטה משמעותית יותר מדויקת במציאת המסלול האופטימלי על גבי משטח רציף.



תמונה 11: ההבדל בין התפשטות הגלים של Dijkstra (למעלה) ושל FMM (למטה). משמאל רואים את הפוטנציאל המחושב ומימין את צורת ההתפשטות ואת קווי המתאר בהם הפוטנציאל שווה (עקומי איזופוטנציאל). ניתן לראות שבשיטת FMM פוטנציאל במרחקים שווים מנקודת ההתחלה גם שווה, לעומת בשיטת Dijkstra

Fast Marching method [15] הוא אלגוריתם נומרי מבוסס Dijkstra (להשוואה ראו [16]) המתאים לתחום רציף לפתרון המשוואה האייקונית על לוח משבצות בסיבוכיות $O(M \log M)$ כאשר M זהו מספר המשבצות בלוח. זוהי שיטה המאפשרת מעקב אחר חזית גל המתפשטת בתחום (עקום סגור במקרה הדו-ממדי). הרעיון מאחורי האלגוריתם הוא לשפר את שלב העדכון של הצמתים הפתוחים (ראו 2.2) לעומת Dijkstra. ניתן לתאר את פתרון המשוואה האייקונית כגלים מעגליים מסביב לנקודת התחלה כלשהי, כאשר הרדיוס של גל בכל נקודה יחסי לפונקציית המחיר. משתמשים במידע מחזית הגל על מנת למצוא קבוצה נוספת של נקודות ומהן מחשבים את התקדמות הגל שוב. כך ממשיכים עד שמצאנו את ערך הפוטנציאל בכל נקודה בתחום. האלגוריתם הנ"ל מחקה את הבנייה הזו על לוח המשבצות.

3.6.1 אלגוריתם Fast Marching

הרעיון הוא לעבוד החוצה מקבוצה של משבצות הנותנות את תנאי ההתחלה. המשבצות המהוות תנאי ההתחלה מוגדרות כ-Frozen, ואנו מחשבים את המרחקים (הפוטנציאל) ביניהם לבין המשבצות השכנות. משבצות אשר חישבנו את המרחק אליהן אבל עוד לא הקפאנו נמצאות בחזית הגל ולכן מוגדרות כ-Narrowband. עבור כל איטרציה בלולאה המרכזית, אנו לוקחים את המשבצת בעלת המרחק הכי קטן ומקפיאים אותה, ולאחר מכן מחשבים את המרחקים של השכנות שלה. ניתן לראות פעולה זו כחזית של גל מתקדם אשר מתפשט מתנאי ההתחלה ומקפיא משבצות שעבר מעליהן.

הגדרות: Frozen – קבוצת כל המשבצות עבורן מצאנו את הפוטנציאל האופטימלי.

Narrowband – קבוצת כל המשבצות הנמצאות בחזית הגל, בהן מחשבים את הפוטנציאל.

Far – כל המשבצות שעוד לא הגענו אליהן.

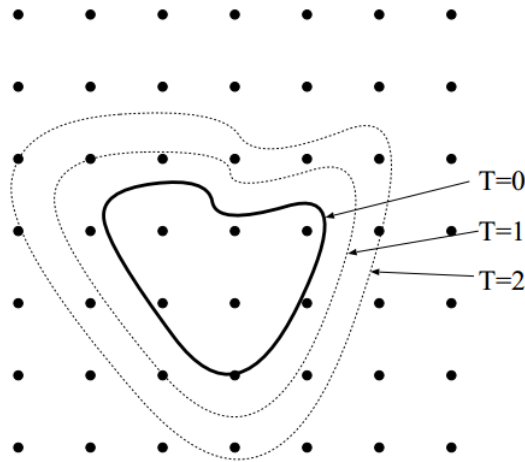
1. איתחול: הגדר מספר משבצות מטרה ותן להם ערך פוטנציאל 0. הגדר אותן Frozen והוסף את כל השכנות שלהן בשכנות 4-כיוונית לקבוצת Narrowband. הגדר את כל שאר המשבצות כ-Far עם ערך פוטנציאל אינסוף.
2. לולאה: הוצא מ-Narrowband והגדר כ-Frozen את המשבצת בעלת הפוטנציאל הכי נמוך. ז"א מצאנו עבור משבצת זו את הפוטנציאל האופטימלי.
3. עבור כל שכן לא קפוא של המשבצת שהוצאנו לחשב את הפוטנציאל שלו ולהוסיף אותו ל-Narrowband. אם השכן כבר ב-Narrowband אז נחשב את הפוטנציאל שלו מחדש ונעדכן את מקומו ברשימה (אם הפוטנציאל החדש קטן מהישן).
4. תחזור ל-2 עד אשר כל המשבצות נמצאות במצב Frozen.

הפסאודוקוד כפי שנתון ב-[17]:

```

Initialization() {
    for each cell c in I //list of initial nodes {
        Freeze c;
        for each neighbour cn of c {
            compute potential p at cn;
            if cn is not in Narrowband {
                tag cn as Narrowband;
                insert (p, cn) into H; //H stands for heap, it's the Narrowband list
            }
            else
                decrease key of cn in H to p; //update position of cn in H
        }
    }
}
Loop() {
    while H is not empty {
        Extract c from top of H;
        Freeze c;
        for each neighbor cn of c
            if cn is not frozen {
                compute potential p at cn
                if cn is not in Narrowband {
                    tag cn as Narrowband;
                    insert (p, cn) into H;
                }
                else
                    decrease key of cn in H to p;
            }
    }
}
}

```



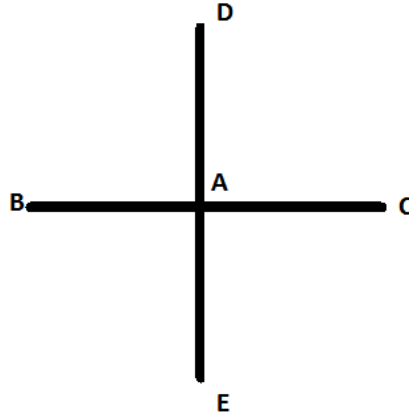
תמונה 12: התקדמות הגל בזמנים שונים, מתוך [17]

3.6.2 חישוב הפוטנציאל

כפי שנכתב קודם, הפוטנציאל מחושב על ידי פתרון של המשוואה אייקוואלית. אנו מחשבים את ערך הפוטנציאל עבור משבצות שנמצאות ב-Narrowband (חזית הגל) כך שאורך הגרדיאנט $\|\nabla\phi\|$ שווה למחיר c . ϕ הוא זמן ההגעה של חזית הגל למשבצת ו- c הוא מחיר המתאר את מהירות הגל במשבצת בצורה כלשהי. לפי [15,17] ניתן להשתמש בקירוב מסדר ראשון הבא לחישוב אורך הגרדיאנט בריבוע:

$$\|\nabla\phi\|^2 = \begin{cases} \max(V_A - V_B, V_A - V_C)^2 + \\ \max(V_A - V_D, V_A - V_E)^2 \end{cases}$$

כאשר V_A הוא ערך הפוטנציאל הלא ידוע ו- V_B, V_C, V_D, V_E הם ערכי הפוטנציאל במשבצות השכנות.



תמונה 13: מיקומי המשבצות עבורן ביחס למשבצת הלא ידועה

בשביל לפתור את המשוואה האייקונומית נציב בה את המשוואה הנ"ל וכיוון שאנו יודעים מהו מחיר התנועה c ניתן יהיה למצוא את V_A :

$$c^2 = \begin{cases} \max(V_A - V_B, V_A - V_C, 0)^2 + \\ \max(V_A - V_D, V_A - V_E, 0)^2 \end{cases}$$

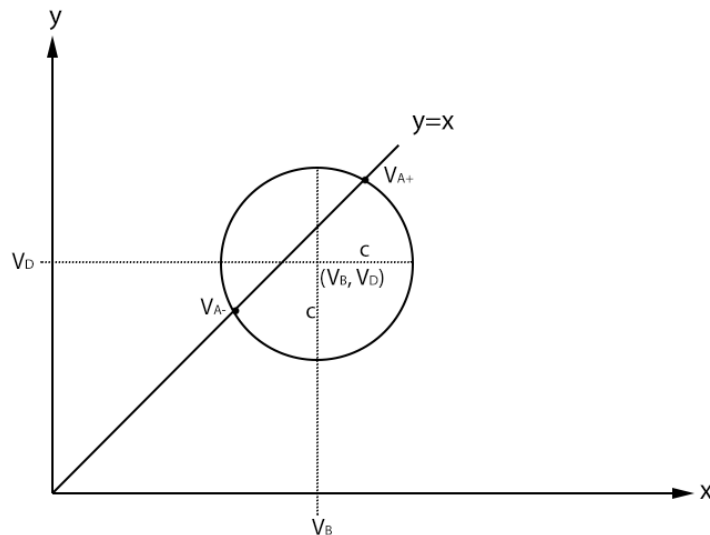
וכדי לפתור את המשוואה אנו צריכים למצוא את הערך הקטן בין V_B ו- V_C ובין V_D ל- V_E . בנוסף לכך מותר לנו להשתמש רק בערכים קפואים (למרות שבמאמר המקורי בו הוצגה השיטה [15] לא השתמשו רק בערכים קפואים). אם אחד הממדים (מאונך או מאוזן) לא קפוא אז אנו מורידים את הביטוי המתאים מהמשוואה (נקרא הורדת מימד). אם ישנם שני פתרונות אנו בוחרים את הגדול מבין השניים, כיוון ש- V_A צריך להיות גדול ממש מערכים הידועים, אחרת יכול להיווצר מצב שבו הגל "נע אחורה בזמן". לדוגמה, אם שני הערכים בציר ה- y אינם קפואים, המשוואה מצטמצמת ל:

$$c^2 = \max(V_A - V_B, V_A - V_C, 0)^2 \xrightarrow{V_B < V_C} V_A = V_B + c$$

כעת נראה את נוסחת הפתרון: נניח בלי הגבלת הכלליות ש- V_B, V_D הם הערכים המינימליים.

$$(V_A - V_B)^2 + (V_A - V_D)^2 = c^2$$

משוואה זו מבטאת נקודות חיתוך בין מעגל $(x - a)^2 + (y - b)^2 = c^2$ והישר $y = x$.



תמונה 14: החיתוך בין המעגל לישר. משתי נקודות החיתוך המתקבלות בוחרים בגדולה יותר בתנאי שהיא גם גדולה יותר מקורדינאטות מרכז המעגל.

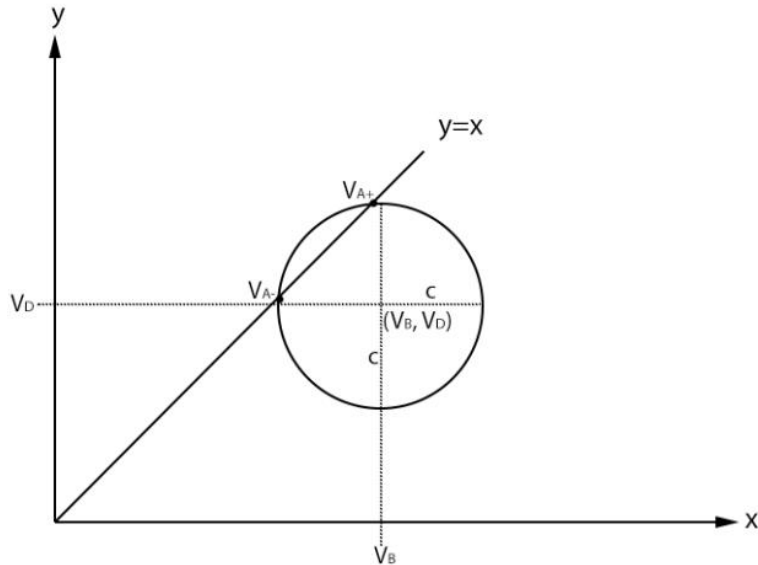
על ידי העברת אגפים מתקבלת משוואה ריבועית מהצורה: $ax^2 + bx + c = 0$

$$(V_A - V_B)^2 + (V_A - V_D)^2 - c^2 = 2V_A^2 + 2(V_B + V_D)V_A + V_B^2 + V_D^2 - c^2 = 0$$

על ידי שימוש בנוסחת השורשים הידועה לפתרון משוואה ריבועית אנו מקבלים:

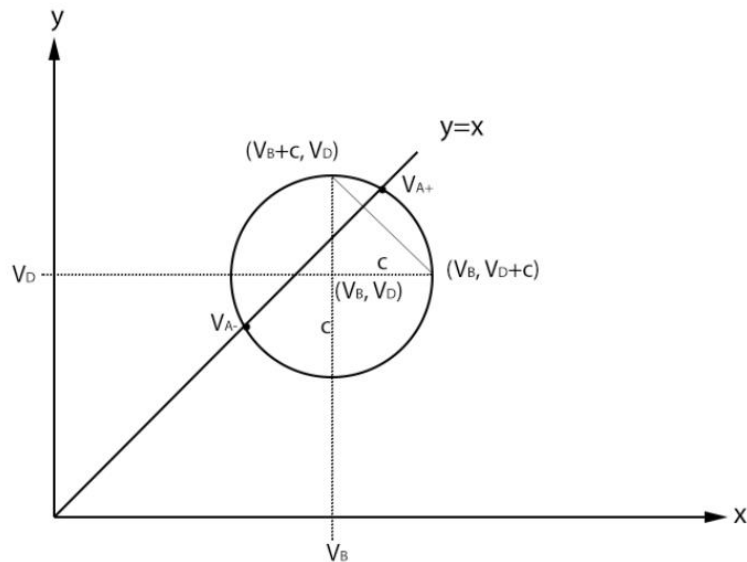
$$V_{A1,2} = \frac{V_B + V_D \pm \sqrt{2c^2 - (V_B - V_D)^2}}{2}$$

הפתרון הנבחר צריך להיות גדול ממש משני ערכי הכניסה V_B, V_D המהווים מרכז המעגל על מנת שלא ייווצר מצב שבו הגל נע אחורה בזמן, תנאי זה יוצר בעיה כיוון שגם אם קיימים שני פתרונות למשוואה, אין הבטחה שאחד מהם יהיה גדול מערכי הכניסה.



תמונה 15: מצב בו אף אחד מהפתרונות אינו גדול ממש משני ערכי הכניסה.

כדי למצוא תנאי המאפשר לזהות מתי מצב זה קורה, יש תחילה לשים לב שפתרון הוא תקף כאשר הוא נמצא ברביע הראשון של המעגל, כי במצב זה הוא יהיה גדול משני ערכי הכניסה. כאשר הישר $y = x$ חותך את המעגל ברביע הראשון, נקודת החיתוך נותנת פתרון תקף. המרחק האופקי ממרכז המעגל לישר הוא $|V_B - V_D|$ ולכן על מנת שנקודת החיתוך תהיה ברביע הראשון נדרוש שמתקיים $c > |V_B - V_D|$. דרך נוספת להגיע לאותו הפתרון היא לבדוק חיתוך: אם הישר חותך את הקשת המחברת בין $(V_B + c, V_D)$ לבין $(V_B, V_D + c)$. במקרה זה נקודת החיתוך של הישר עם המעגל תהיה ברביע הראשון.



תמונה 16: חיתוך בין הקשת המוצגת והישר $y=x$ בצורה שנותנת פתרון תקף.

חישוב משוואת הישר העובר בין $(V_B + c, V_D)$ לבין $(V_B, V_D + c)$ נותן:

$$y = -x + (V_B + V_D + c)$$

בנקודת החיתוך $y = x$ ולכן מיקום נקודת החיתוך הוא $\left(\frac{V_B+V_D+c}{2}, \frac{V_B+V_D+c}{2}\right)$. בכדי שהנקודה תהיה ברביע הראשון צריכים להתקיים התנאים $\frac{V_B+V_D+c}{2} > V_D$ וגם $\frac{V_B+V_D+c}{2} > V_B$ וכשמפשטים אותם מתקבל $c > V_D - V_B$ וגם $c > V_B - V_D$, זאת ניתן לצמצם לתנאי:

$$c > |V_B - V_D|$$

שהוא אותו התנאי שקיבלנו קודם. לסיכום הפתרון במשבצת מסוימת מחושב ע"י:

$$V_A = \begin{cases} \frac{V_B + V_D + \sqrt{2c^2 - (V_B - V_D)^2}}{2}, & c > |V_B - V_D| \\ \min(V_B, V_D) + c, & \text{otherwise} \end{cases}$$

נשים לב שהתנאי הנ"ל מטפל בהצלחה גם במצבים בהם הדיסקרימיננטה שווה ל-0 (המעגל משיק לישר) וקטנה מ-0 (המעגל לא נוגע בישר).

4 אלגוריתם Continuum Crowds

אלגוריתם Continuum Crowds [18] הוא מודל זמן-אמת המתאר תנועת המון באמצעות דינמיקת רצף. המודל משתמש בשדה פוטנציאל דינמי (משתנה מרגע לרגע) המשלב ניווט גלובלי עם התחמקות ממכשולים נעים כגון סוכנים אחרים. סימולציות עם המודל יכולות לרוץ בזמן אמת כך שמתאפשרת אינטראקציה, ומדגימות תופעות המתרחשות גם בהמון אמיתי.

מודל האלגוריתם מבוסס ישירות על עבודתו של Hughes המוצגת ב-3.3. משתמשים בפונקציית פוטנציאל דומה על מנת ליצור שדה זרימה המנווט סוכנים לכיוון היעד שלהם. המודל גם משתמש בטכניקה של Hughes לקיבוץ הסוכנים לקבוצות והשימוש בשדות אי-נוחות בשביל לתאר העדפות של סוכנים. האלגוריתם משתמש בגרסת המודל עם המשוואה אייקוולית. המאמר המקורי של Hughes עסק רק בתכונות האנליטיות של המודל אותו יצר, ובמאמר של Continuum Crowds הכותבים נותנים מודל מתמטי יישומי המבוסס על תכונות אלה כך שניתן יהיה לממש את המודל במחשב.

האלגוריתם גם משפר מעט את המודל עצמו בחישוב המהירות ובכך מאפשר למערכת להראות תופעות מעניינות שקורות גם בחיים האמיתיים כמו למשל יצירת שיירות של סוכנים, והשינוי בפונקציית המחיר משנה את ההגדרה של Hughes לאופטימליות של מסלול.

4.1 המודל המתמטי ל-Continuum Crowds

המודל מבוסס על 4 הנחות יסוד שדומות מאוד להנחות במודל Hughes.

1. לכל סוכן יש מטרה G אליה הוא מנסה להגיע. כמעט בכל מודל תנועה קיים יעד אליו הסוכנים מנווטים, וכך גם במודל זה. במודל Hughes וגם במודל זה הנחת היסוד הזו המתארת כוח כלשהו המניע את הסוכנים ליעד. היעד יכול להיות מקום יחיד, מספר אזורים נפרדים שביניהם יבחר הסוכן בקרוב מביניהם, או ישות דינמית כלשהי שאחריה הסוכן יעקוב.
2. אנשים נעים במהירות המקסימלית האפשרית. בשביל לייצג זאת קיים שדה מהירות מקסימלית f כך שכל סוכן במיקום x ובכיוון θ נע במהירות $\dot{x} = f(x, \theta)n_\theta$ כאשר n_θ זה וקטור הכיוון המנורמל $(\cos(\theta), \sin(\theta))$. מהירות הסוכנים תלויה בפרמטרים סביבתיים כגון שיפוע האזור בו נע, צפיפות האנשים סביבו (כמו במודל Hughes) וכיוון הזרם. לסוכנים יהיה קשה יותר לנוע נגד הזרם ולכן ינועו לאט יותר.
3. קיים שדה אי-נוחות g התלוי במיקום ובזמן שעבורו, בהנחה שכל השאר שווה, סוכנים יעדיפו לעבור בנקודה x במקום בנקודה x' אם מתקיים $g(x) < g(x')$.
4. נסמן Π קבוצת כל המסלולים מנקודה כלשהי x למטרה G . בהנחה שהמטרה G , שדה המהירות f והאי-נוחות g קבועים, סוכן בנקודה x יבחר במסלול $P \subseteq \Pi$ הממזער את הביטוי הבא:

$$\alpha \int_P 1 ds + \beta \int_P 1 dt + \gamma \int_P g dt \xrightarrow{ds=f dt} \alpha \int_P 1 ds + \beta \int_P \frac{1}{f} ds + \gamma \int_P \frac{g}{f} ds$$

כאשר האינטגרל הראשון מסמל את אורך המסלול, השני את הזמן, והשלישי אי-נוחות. המקדמים הם משקלים שונים עבור כל אחד מהרכיבים. מכאן אם נאחד את האינטגרלים נקבל את אינטגרל פונקציית המחיר המסמל את המחיר הכולל למטרה, שמופיע גם בניסוח מחדש של מודל Hughes ב-3.5:

$$\int_P \frac{\alpha f + \beta + \gamma g}{f} ds \equiv \int_P C ds$$

C הוא שדה המחירים, ומחושב לפי הביטוי בתוך האינטגרל לפי אורך המסלול ds. זהו השדה שמשתמשים בו עבור חישוב המסלול ומהווה גירסה יישומית של פונקציית המחיר במודל Hughes.

4.1.1 צפיפות ומהירות

מהירות היא משתנה תלוי בצפיפות, והצפיפה היא שעבור צפיפות נמוכה המהירות תהיה גבוהה, ועבור צפיפות גבוהה המהירות תהיה נמוכה.

על מנת לייצג את הצפיפות מגדירים שדה צפיפות כללי ρ ומתייחסים לכל סוכן כשדה צפיפות ρ_i במקום גוף מוצק. שדה הצפיפות הפרטי לסוכן מוגדר כפונקציה של המרחק ממנו כך שבתוך רדיוס r הצפיפות לא יורדת מערך כלשהו $\bar{\rho}$ ולא עולה מאותו ערך מחוץ לרדיוס הנבחר. במילים אחרות, בשביל לייצג את גוף האדם משתמשים ברדיוס מסביב למיקום הסוכן כך שעד הרדיוס הזה, התוספת לצפיפות היא מקסימלית. כדי לקבל את שדה הצפיפות הכללי אנו סוכמים את השדות:

$$\rho = \sum_i \rho_i$$

באמצעות שדה הצפיפות מחשבים גם את שדה המהירות הממוצעת $\bar{v}$ (שדה וקטורי) המעיד על וקטור המהירות הממוצע של זרימת ההמון:

$$\bar{v} = \frac{\sum_i \rho_i \dot{x}_i}{\rho}$$

תחום הצפיפות מחולק לשלושה חלקים: $\rho \leq \rho_{min}$, $\rho_{min} < \rho < \rho_{max}$ ו- $\rho \geq \rho_{max}$ כאשר ρ_{min} ו- ρ_{max} הם הצפיפות הקריטית וצפיפות הפקק בהתאמה (ראו פרק 3).

- צפיפות נמוכה $\rho \leq \rho_{min}$: בצפיפות נמוכה המהירות f שווה למהירות הטופולוגית f_T .
 - המהירות הטופולוגית נשלטת ע"י השיפוע המקסימלי s_{max} והשיפוע המינימלי s_{min} , בהנחה שהבדלי הגבהים בשטח תחומים בין שני השיפועים.
 - בהינתן שדה גובה h השיפוע ניתן ע"י $\nabla h(x) \cdot n_\theta$.
 - מהירות הזרימה בצפיפות נמוכה תהיה שווה למהירות הטופוגרפית כל עוד מתקיים האי-שוויון $\rho_{min} \geq \bar{\rho}$.
 - פונקציית המהירות הטופולוגית שנבחרה במודל מהווה אינטרפולציה לינארית בין מהירות שיפוע שלילי מקסימלי f_{max} לבין מהירות שיפוע חיובי מקסימלי f_{min} :

$$f_T(x, \theta) = f_{max} + \frac{\nabla h(x) \cdot n_\theta - s_{min}}{s_{max} - s_{min}} (f_{min} - f_{max})$$

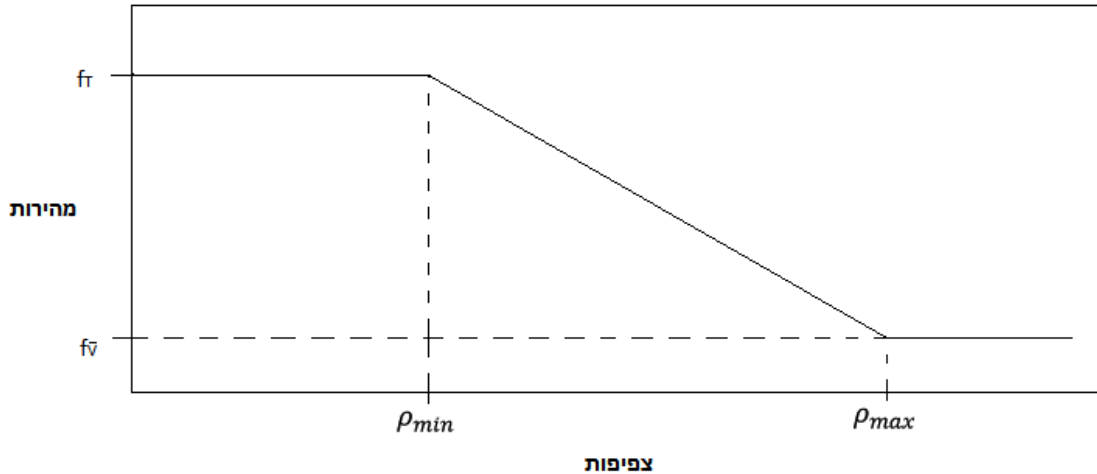
- צפיפות גבוהה $\rho_{max} \leq \rho$: בצפיפות גבוהה המהירות שווה למהירות הזרימה $f_{\bar{v}}$.
 - מהירות הזרימה היא המהירות הממוצעת $\bar{v}$ במרחק r ממיקום המדידה.
 - ההיסט גורם לסוכנים לדגום את המהירות הממוצעת באזור אליו הם נכנסים, אחרת מהירות של סוכן תושפע על ידי המהירות הקודמת שלו.
 - מהירות הזרימה מוגדרת להיות אי-שלילית כיוון שקהל יכול להאט אנשים שהולכים נגדו, אבל לעולם לא למשוך אותם אחורה (לפחות לא במודל זה).

$$f_{\bar{v}}(x, \theta) = \bar{v}(x + rn_\theta) \cdot n_\theta$$

- צפיפות בינונית $\rho_{min} \leq \rho < \rho_{max}$:
 - במהירויות בינוניות אנו מבצעים אינטרפולציה לינארית בין מהירות הזרימה למהירות הטופוגרפית.
 - דוגמים את הצפיפות במרחק r , אחרת הסוכן יושפע מהצפיפות שלו.

$$f(x, \theta) = f_T(x, \theta) + \frac{\rho(x + rn_\theta) - \rho_{min}}{\rho_{max} - \rho_{min}} (f_{\bar{v}}(x, \theta) - f_T(x, \theta))$$

שדה המהירות תלוי בכיוון ולכן אניזטרופי ולכן כל המהירויות המופיעות בחישוב מכוונות באותו הכיוון. כיוון שמהירות הזרימה מוגדרת להיות אי-שלילית, המהירות המינימלית תמיד תהיה חיובית בכיוון התנועה, גם אם זרימת הסוכנים באותה נקודה היא בכיוון הנגדי θ .



תמונה 17: חישוב המהירות בהתאם לצפיפות

4.1.2 חישוב המסלול האופטימלי

שלב זה מסביר את הרקע התיאורטי למציאת המסלול האופטימלי. מניחים כי קיימת פונקציית ϕ על המשטח כך שבכל נקודה היא שווה למחיר המסלול האופטימלי ליעד. ברמה האינטואיטיבית הגיוני שכל סוכן יעקוב בכיוון הנגדי לגרדיאנט של הפונקציה הזו כיוון שכך המחיר יקטן הכי מהר. בנוסף לכך, ניתן לבנות את הפונקציה הזאת ע"י כך שנעקוב אחר כל המסלולים האופטימליים מהיעד החוצה, בזמן שמבצעים סכימה (או אינטגרציה בגירסה הרציפה) של המחיר על גבי המסלול. זו הגדרה של המשוואה האייקונית $\|\nabla\phi(x)\| = C(x, \theta)$ עם פונקציית פוטנציאל ϕ ופונקציית מחיר C , וביעד $\phi = 0$. לאחר פתרון המשוואה באמצעות Fast Marching Method נקבל את שדה הפוטנציאל שממנו נמצא שיפועים באותה צורה כמו בהסבר על שדות זרימה. הסוכנים ינועו בכיוון הנגדי לגרדיאנט ובמהירות:

$$\dot{x} = f(x, \theta) \frac{\nabla\phi}{\|\nabla\phi\|}$$

יש לשים לב כי פונקציית המחיר תלויה בכיוון, ז"א אניסטרופית. שיטת FMM כפי שהיא אינה מתאימה לשדה מחירים אניסטרופי כיוון שהיא מדמה התפשטות של גל מעגלי ולכן דורשת שבכל נקודה, מחיר תנועה הגל דרך הנקודה לא תלוי בכיוון. בהמשך נראה שינוי לשיטה כך שבמקום גל מעגלי נשתמש בגל אליפטי המתאים יותר לתנועה אניסטרופית.

4.1.3 שימוש באי-נוחות לחיזוי תנועה

המודל משתמש בתמונה סטטית מתעדכנת של כל הסביבה על מנת לבצע ניווט ולכן לא לוקח בחשבון תנועה של מכשולים אלא רק מעדכן את מיקומם, מה שעלול לגרום לבעיות כאשר סוכנים נעים בכיוונים מנוגדים זה אל זה. הרי כשבני אדם הולכים ברחוב הם יתחמקו מאלה הבאים מולם.

על מנת למדל זאת ניתן להשתמש בשדה האי-נוחות בכך שעבור כל סוכן, נחשב מה יהיה המיקום שלו בעוד זמן מסוים ובהתאם למיקום החזוי הזה נוסיף לשדה האי-נוחות ערך באותה צורה כמו לצפיפות.

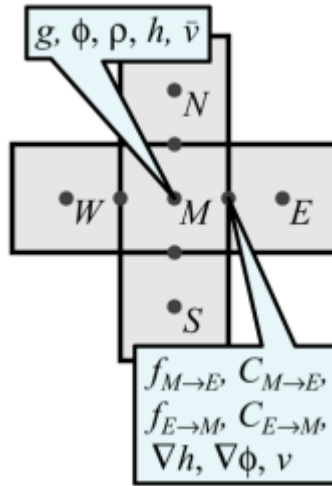
$$g(x + \alpha \cdot rn_\theta) = \beta \sum_i \rho_i(x + \alpha \cdot rn_\theta)$$

כאשר α מספר הקפיצות קדימה בזמן, β הוא פרמטר פרופורציה, והסכום נלקח עבור כל הסוכנים שישפיעו על הצפיפות בנקודה זו לאחר קידום המיקום שלהם בהתאם ל- α .

הסיבה שלא מוסיפים זאת לשדה הצפיפות היא כמובן שזה רק מקום חזוי ולכן אין השפעה אמיתית על צפיפות האזור.

4.2 המודל במחשב

על מנת לחשב את שדות הצפיפות (סקלרי), המהירות הממוצעת (וקטורי), המהירות (סקלרי אניסטרופי), המחירים (סקלרי אניסטרופי), הפוטנציאל (סקלרי) והשיפועים (וקטורי) צריך קודם כל לבחור במבנה נתונים ודיסקרטזציה לסביבת התנועה שתתאים לחישובים. השטח הופך ללוח משבצות, כאשר בכל משבצת מחזיקים את הערכי השדות בה.



תמונה 18: מבנה לוח המשבצות מתוך [continuumcrowds]

- וקטור בשביל שיפוע השטח.
- ערך יחיד בשביל שדה הצפיפות וערך נוסף לשדה האי-נוחות.
- וקטור יחיד בשביל כיוון וגודל הזרימה (מהירות ממוצעת) במשבצת.
- עבור שדות מהירות ומחירים שהם שדות אניסטרופיים מחזיקים 4 ערכים בכל כיוון ראשי, ז"א $\theta = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} = \{E, N, W, S\}$ המתאימים לכיוון של כל אחת מהמשבצות השכנות. הערכים מסמלים ערכי מהירות ומחיר מעבר בין המשבצת הנוכחית למשבצות מסביבה.

4.2.1 חישוב הצפיפות והאי-נוחות לחיזוי תנועה

על מנת לחשב את שדה המהירות צריך קודם כל לבנות את שדה הצפיפות. בשביל לעשות זאת "מועכים" את הסוכנים על שדה הצפיפות.

ישנן שתי דרישות פונקציית ההמרה לשדה הצפיפות:

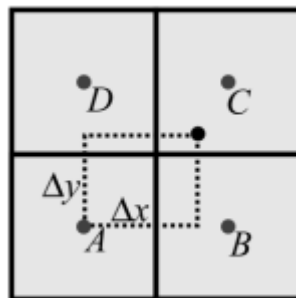
- השדה צריך להיות רציף מבחינת מיקומי הסוכנים – ניתן למדל זאת באמצעות טכניקת מעיכה בילינארית (פעולה דומה לאקסטרפולציה).
- כל סוכן יוסיף לא פחות מ- $\bar{\rho}$ למשבצת בה הסוכן נמצא ולא יותר מ- $\bar{\rho}$ לכל משבצת שכנה.

כדי לספק את הדרישה השנייה:

- הצפיפות מתווספת ללוח המשבצות לפי הנוסחאות הבאות:

$$\rho_A = \min(1 - \Delta x, 1 - \Delta y)^\lambda \quad \rho_B = \min(\Delta x, 1 - \Delta y)^\lambda$$

$$\rho_C = \min(\Delta x, \Delta y)^\lambda \quad \rho_D = \min(1 - \Delta x, 1 - \Delta y)^\lambda$$



תמונה 19: מיקום מרכזי המשבצות ביחס למיקום הסוכן [continuumcrowds]

- λ זה קצב הדעיכה של הצפיפות בהתאם למרחק הסוכן.
- עבור $\bar{\rho} = \frac{1}{2^\lambda}$ צפיפות כלשהו $\bar{\rho}$, λ מחושבת לפי $\bar{\rho} = \frac{1}{2^\lambda}$.

עם שדה הצפיפות ניתן לחשב את שדה המהירות הממוצעת ומשם את שדה המהירות האניסוטרופי לפי הנוסחאות שניתנו קודם לכן, בשביל שדה המחירים.

עבור האי-נוחות לחיזוי תנועה משתמשים באותה שיטה, אבל במקום להשתמש במיקומים הנוכחיים של הסוכנים, עבור כל סוכן נחשב איפה יימצא לאחר כמות צעדים נתונה ונוסיף לשדה האי-נוחות במקום לשדה הצפיפות.

4.2.2 חישוב שדה המחירים

כפי שהוסבר קודם לכן המהירות היא אניסוטרופית ותלויה בכיוון הזרימה של ההמון. לכן צריך לחשב את המחיר בכל אחד מארבעת הכיוונים בלוח המשבצות שלנו. עבור משבצת (i, j) כלשהי וכיוון θ אנו מחשבים מחיר המעבר ממשבצת זו למשבצת $(i, j)_\theta$ (המשבצת השכנה בכיוון θ) כך:

$$C_{i,j_\theta} = \frac{W_L \cdot f_{i,j_\theta} + W_T + W_D \cdot g_{i',j'}}{f_{i,j_\theta}}$$

כאשר W_L חשיבות אורך המסלול, W_T חשיבות זמן הנסיעה ו- W_D חשיבות אי-נוחות המסלול. את המהירות לוקחים במעבר בין המשבצת הנוכחית לשכנה בכיוון הדרוש, ואת אי-הנוחות דוגמים במשבצת i', j' שהיא המשבצת השכנה אליה נעים, מה שיאפשר לסוכנים לחזות מכשולים יותר טוב.

4.2.3 חישוב שדה הפוטנציאל

לאחר יצירת שדה המחירים יש לנו את תנאי ההתחלה ($\phi = 0$) וגם תנאי השפה (שדה המחירים עצמו) ואנו יכולים כעת לפתור את המשוואה האייקונית:

$$\|\nabla\phi(x)\| = C(x, \theta)$$

על מנת לבנות את שדה הפוטנציאל וממנו את שדה השיפועים.

המחיר אניסוטרופי ולכן שימוש בשיטה מחלק 3.6 לא יעבוד טוב במיוחד כיוון ששם מחיר התנועה מותאם לרדיוס של מעגל – מה שאומר שמחיר התנועה שווה בכל הכיוונים (אניסוטרופי). על מנת לקחת בחשבון את התלות של המחיר בכיוון צריך נוסחה שמאפשרת זאת. בשביל למדל את התלות הזו על לוח משבצות הכותבים מציעים את הנוסחה הבאה:

$$\left(\frac{\phi_{i,j} - \phi_x}{C_x}\right)^2 + \left(\frac{\phi_{i,j} - \phi_y}{C_y}\right)^2 = 1$$

- כאשר אם $C_x = C_{i,j_W}$ אז $\phi_x = \phi_{i-1,j}$ וגם $\phi_{i-1,j} + C_{i,j_W} \leq \phi_{i+1,j} + C_{i,j_E}$ אחרת $C_x = C_{i,j_E}$ וגם $\phi_x = \phi_{i+1,j}$ ○
- וגם אם $C_y = C_{i,j_S}$ אז $\phi_y = \phi_{i,j-1}$ וגם $\phi_{i,j-1} + C_{i,j_S} \leq \phi_{i,j+1} + C_{i,j_N}$ אחרת $C_y = C_{i,j_N}$ וגם $\phi_y = \phi_{i,j+1}$ ○

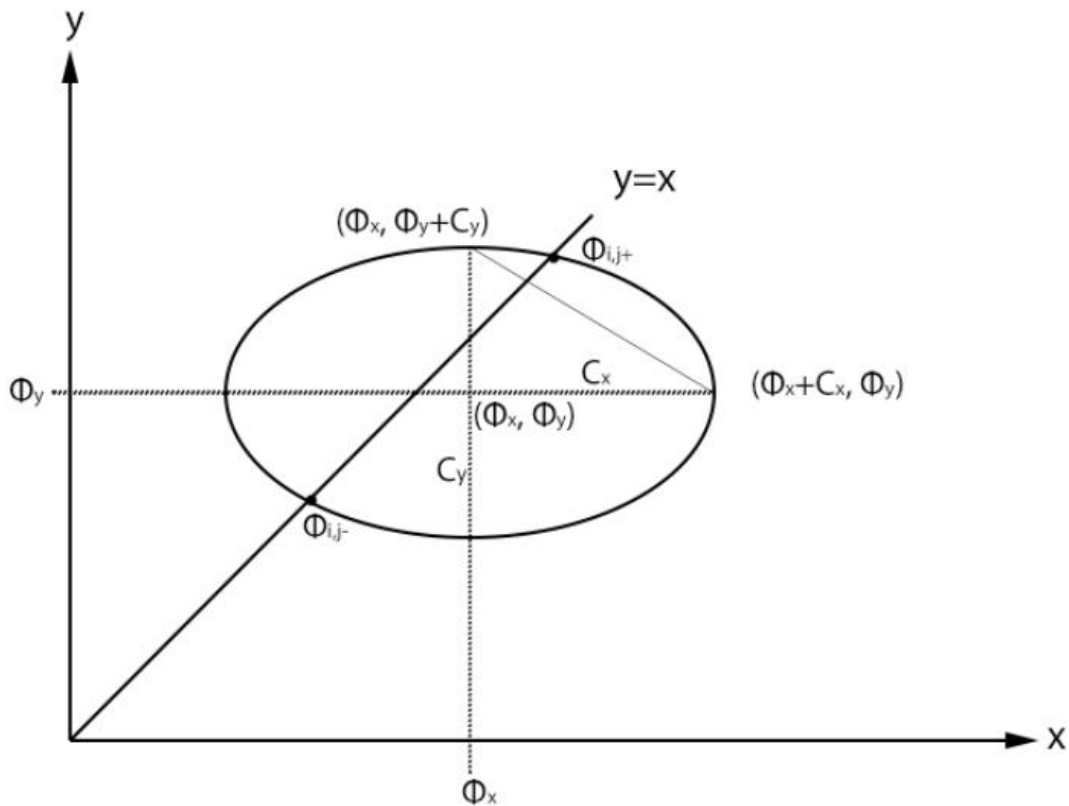
$\phi_{i,j}$ הוא הפוטנציאל הלא ידוע במשבצת (i, j) , $C_{i,j_{\{E,N,W,S\}}}$ הוא מחיר התנועה דרך משבצת (i, j) לכל אחד מארבעת הכיוונים. הבחירה בין הכיוונים נעשית באמצעות הסכום של הפוטנציאל במשבצת בכיוון ומחיר התנועה בכיוון, ואנו בוחרים בכיוונים עם הסכום הנמוך יותר בכל כיוון בנפרד.

המאמר על האלגוריתם [18] לא נותן הסבר מפורט איך להשתמש בנוסחת ההפרשים המוצגת. ניסיון להשתמש בה כפי שהיא עם התנאי שחושב עבור עיגול לא עבד. נוסחה אחרת לחישוב אניסוטרופי מתוך [19] השתמשה בסכום המחירים בכיווני x ו-y כתחליף בנוסחה המקורית, ואיפשרה לפתור את המשוואה האייקונית ולסוכנים לנווט ליעד, אבל הסוכנים נטו להצטופף לטורים גם כאשר לא היו צריכים כגון במצב בו כל הסוכנים נעים באותו הכיוון. לבסוף

לעומת הנוסחה בפרק 3.6.2 המתארת חיתוך בין הישר $y = x$ למעגל, הצורה הזו מתארת חיתוך בין אליפסה

$$\left(\frac{x-h}{a}\right)^2 + \left(\frac{y-k}{b}\right)^2 = 1$$

לישר $x = y$. כפי שידוע (h, k) מרכז האליפסה, ו- a, b הם המרחקים האופקי והאנכי ממרכז האליפסה להיקף. בשיטה שהוצגה קודם ישנו מחיר אחד לכל כיוון המבוטא באמצעות רדיוס המעגל. כאן על מנת לאפשר מחירים שונים בכיוון האופקי והאנכי, המחיר האניסוטרופי המינימלי בכל מימד מבוטא ע"י a, b .



תמונה 20: דיאגרמה של פתרון המשוואה.

אם נפתח את הנוסחה נקבל שוב משוואה ריבועית:

$$(C_x^2 + C_y^2)\phi_{i,j}^2 - 2(\phi_x C_y^2 + \phi_y C_x^2)\phi_{i,j} + (\phi_x^2 C_y^2 + \phi_y^2 C_x^2 - C_x^2 C_y^2) = 0$$

וכשנפתור נקבל:

$$\phi_{i,j,1,2} = \frac{\phi_x C_y^2 + \phi_y C_x^2 \pm \sqrt{-C_x^2 C_y^2 [(\phi_x - \phi_y)^2 - (C_x^2 + C_y^2)]}}{C_x^2 + C_y^2}$$

הפוטנציאל המחושב $\phi_{i,j}$ צריך להיות גדול יותר משכניו ϕ_x, ϕ_y ולכן לא די שיהיה קיים פתרון, אבל התנאי מפרק 3.6.2 מתאים רק למעגל. על מנת למצוא תנאי מתאים נבצע את אותו התהליך שבו אנו בודקים האם נקודת החיתוך של $y = x$ עם הקשת המחברת בין $(\phi_x, \phi_y + C_y)$ ל- $(\phi_x + C_x, \phi_y)$ נמצאת ברביע הראשון של האליפסה.

משוואת הקשת היא:

$$y = -\frac{C_y}{C_x}x + \left(\frac{C_y}{C_x}\phi_x + C_y + \phi_y\right)$$

ולכן נקודת החיתוך היא $\left(\frac{C_y\phi_x + C_x C_y + C_x \phi_y}{C_x + C_y}, \frac{C_y\phi_x + C_x C_y + C_x \phi_y}{C_x + C_y}\right)$. עלינו לבדוק שהקואורדינטה השמאלית גדולה מ- ϕ_x וגם הקואורדינטה הימנית גדולה מ- ϕ_y :

$$\begin{cases} \frac{C_y \phi_x + C_x C_y + C_x \phi_y}{C_x + C_y} > \phi_x \\ \frac{C_y \phi_x + C_x C_y + C_x \phi_y}{C_x + C_y} > \phi_y \end{cases}$$

כשנפשט את התנאי נקבל:

$$\begin{cases} C_y > \phi_x - \phi_y \\ C_x > \phi_y - \phi_x \end{cases}$$

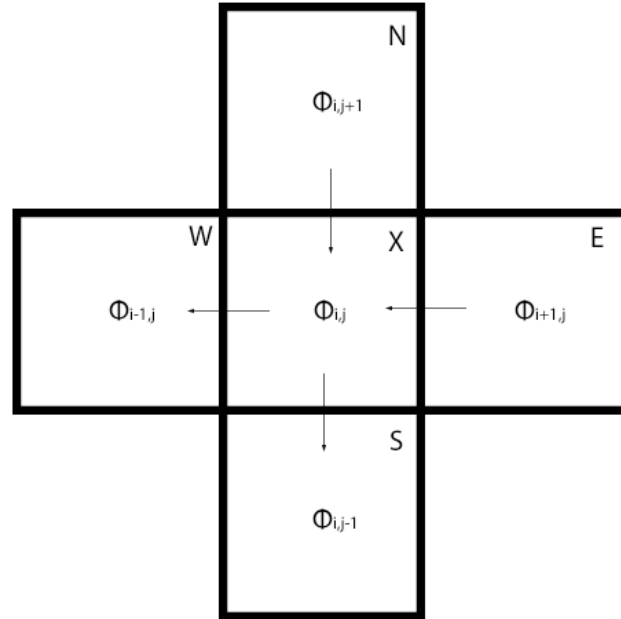
לסיכום צורת הפתרון שלנו למשבצת i, j היא:

$$\phi_{i,j,2} = \begin{cases} \frac{\phi_x C_y^2 + \phi_y C_x^2 \pm \sqrt{-C_x^2 C_y^2 [(\phi_x - \phi_y)^2 - (C_x^2 + C_y^2)]}}{C_x^2 + C_y^2}, & \begin{matrix} C_y > \phi_x - \phi_y \\ C_x > \phi_y - \phi_x \end{matrix} \\ \min(\phi_x + C_x, \phi_y + C_y), & \text{otherwise} \end{cases}$$

מערכת זו מטפלת בהצלחה גם במצבים בהם הדיסקרימיננטה שווה ל-0 (האליפסה משיקה לישר) וקטנה מ-0 (האליפסה לא נוגעת בישר).

4.2.4 חישוב שדה השיפועים

לאחר שנבנה שדה הפוטנציאל, ניתן להשתמש בו לחישוב שדה השיפועים. חישוב הגרדיאנט מתבצע בכיוון החיובי של הצירים בכדי שהסוכנים ינווטו לפי שדה השיפועים. לאחר החישוב מתקבל וקטור בכל משבצת המצביע בכיוון הגרדיאנט של פונקציית הפוטנציאל.



תמונה 21: דוגמה למשבצת עם המשבצות השכנות. החצים מראים מהו כיוון חישוב השיפוע.

עבור כל ציר נחשב את ההפרש בין המשבצת בכיוון החיובי למשבצת המרכזית, ובין המשבצת המרכזית למשבצת בכיוון השלילי (לפי החצים בתמונה).

$$\begin{aligned} diff_x^+ &= \phi_{i+1,j} - \phi_{i,j} & diff_y^+ &= \phi_{i,j+1} - \phi_{i,j} \\ diff_x^- &= \phi_{i,j} - \phi_{i-1,j} & diff_y^- &= \phi_{i,j} - \phi_{i,j-1} \end{aligned}$$

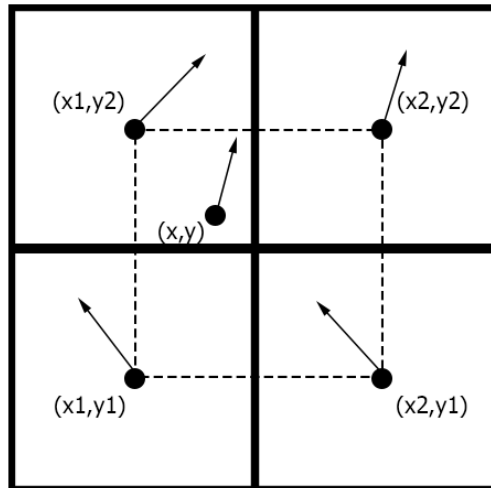
ואז קואורדינטות הגרדיאנט מחושבות לפי הנוסחה הבאה:

$$grad_{i,j,x,y} = \begin{cases} diff_{x,y}^- & diff_{x,y}^- \leq diff_{x,y}^+ \\ diff_{x,y}^+ & otherwise \end{cases}$$

כמו במודל Hughes הסוכנים ינועו בכיוון הנגדי לגרדיאנט על מנת להגיע ליעד.

4.2.5 תנועת הסוכנים

תנועת הסוכנים מבוססת על מה שמוסבר בפרק 2.4 עם מספר שינויים ותוספות בשביל להתאים יותר טוב למודל. כל סוכן דוגם את שדה הזרימה על מנת לדעת מהו כיוון התנועה הרצוי. מיקום הדגימה הוא לא במרכז הסוכן אלא במרחק מסוים מהמרכז שלו כאשר המרחק המינימלי שאפשר לבחור הוא רדיוס הסוכן, כיוון שכך הדגימה לא תהיה מושפעת מצפיפות ומהירות הסוכן עצמו. על מנת ששדה השיפועים יהיה רציף, בלקיחת הדגימה מתבצעת אינטרפולציה בלינארית בין ערכי השיפועים במרכזי המשבצות מסביב לנקודת הדגימה, כדי לחשב את הערך בנקודה.



תמונה 22: מבנה של שדה הזרימה עם דגימה בנקודה (x,y) . הוקטור בנקודה מתקבל באמצעות אינטרפולציה בלינארית בין הוקטורים במשבצות הסובבות.

אינטרפולציה בלינארית היא הרחבה של אינטרפולציה לינארית עבור פונקציות של שני משתנים. הרעיון הוא לבצע אינטרפולציה בכיוון אחד, ואז לבצע בכיוון השני עם הערכים שחושבו. למרות השם המטעה, האינטרפולציה היא לא לינארית אלא ריבועית. קודם כל מבצעים אינטרפולציה לינארית בציר x:

$$f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} grad(x_1, y_1) + \frac{x - x_1}{x_2 - x_1} grad(x_2, y_1)$$

$$f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} grad(x_1, y_2) + \frac{x - x_1}{x_2 - x_1} grad(x_2, y_2)$$

לאחר מכן מבצעים אינטרפולציה לינארית בציר y בשביל לקבל את ההערכה בנקודה הרצויה:

$$grad(x, y) \approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2)$$

לאחר חישוב הגרדיאנט במיקום הדגימה מחושבת באותה הצורה המהירות מתוך שדה המהירות. באמצעות התנהגות ההכוון מפרק 2.4 מחשבים את הכיוון המוחלק, וכיוון זה מכפילים במהירות המחושבת:

$$velocity = smoothedDirection * speed$$

ואז משתמשים בשיטת אוילר בכדי לקבל את המיקום החדש:

$$newPosition = oldPosition + deltaTime * velocity$$

כאשר $deltaTime$ הוא הזמן שנבחר לצעד סימולציה יחיד (לדוגמה 0.0332 עבור 30 צעדים של סימולציה בשנייה אחת).

צריך לקחת בחשבון מצב שבו הסוכן עלול להתנגש בקיר. במצב זה הוחלט להסתובב במקום (שמאלה) אם זווית הפגיעה בקיר חדה מדי, עד שהסוכן כבר לא נע לתוך הקיר.

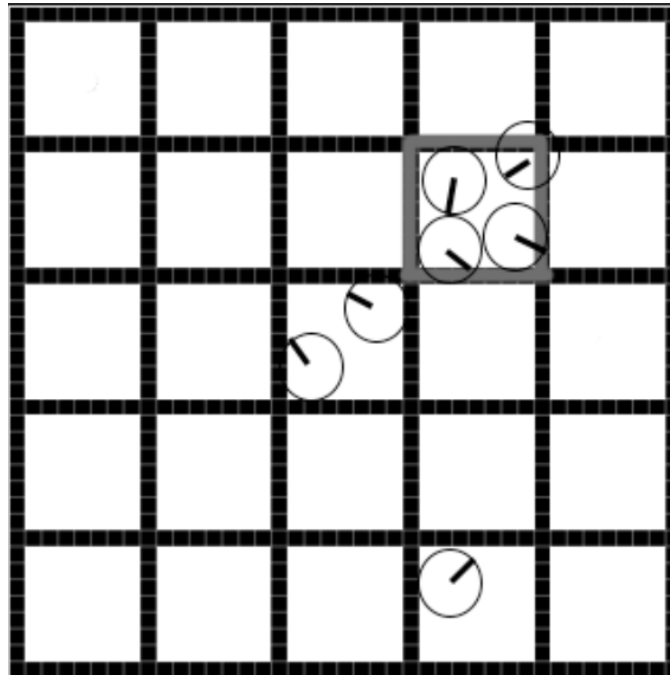
עם שיטת הכוון זו נוצרת בעיה שבה אם סוכן נע לעבר קיר משמאלו הוא יבצע סיבוב שלם לפני שימשיך. פתרון בעיה זו דורש מספר רב של בדיקות ולכן לא מומש. מומשה גירסה פשוטה יותר שמסוגלת לפתור בעיה זו בחלק מהמקרים.

4.2.6 מניעת התנגשויות מיקרוסקופית

בחישובי הצפיפות יכולים למנוע התנגשויות ברזולוציה של לוח המשבצות, אבל כתוצאה מהגודל שנבחר לכל משבצת אם יש שני סוכנים באותה משבצת הם עלולים להתנגש. דבר זה גורם לתוצאות לוואי בלתי רצויות מבחינת הסימולציה, ולכן צריך להוסיף שיטה שתאפשר אכיפה של מרחק מינימלי בין הסוכנים. על מנת לזהות התנגשויות צריך לבדוק האם יש חיתוך בין זוגות של סוכנים. הזיהוי עצמו מתבצע באמצעות בדיקה פשוטה יחסית של בדיקת המרחק בין מרכזים של כל שני סוכנים והשוואתו לרף מסוים (לדוגמה קוטר הסוכן). במקרה שהמרחק קטן או שווה לרף מוחלט כי הסוכנים מתנגשים ומתבצעת הרחקה שלהם באופן סימטרי.

לבדוק את כל הזוגות האפשריים על הלוח זה תהליך מאוד יקר עם מספר סוכנים גבוה. למשל עבור 1000 סוכנים יש לבדוק $499500 = \binom{1000}{2}$ זוגות. לכן מומלץ מאוד להשתמש בשיטה שתצמצם את מספר ההשוואות שעלינו לעשות.

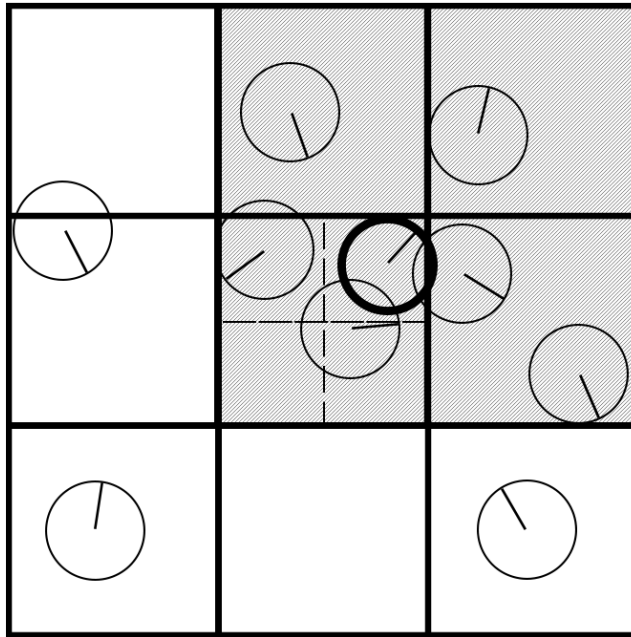
הרעיון מאחורי טכניקת ה-Binning (ראו [20]) היא לחלק את התחום לאוסף של פחים (bins) שהם אזורים שווי גודל המכסים את התחום באותה צורה כמו רשת משבצות (לדוגמה גודל של כל פח יהיה שווה לגודל של משבצת). כל פח יכול להכיל מספר מוגבל מאוד של סוכנים, למשל 4 לכל היותר, וכל סוכן מעבר למספר הזה לא נכלל בבדיקה. בזמן שזה לא פתרון מושלם, כיוון שהצפיפות מפרידה את הסוכנים בכל מקרה, מצב בו יהיו יותר מארבעה סוכנים במשבצת הוא נדיר מאוד.



תמונה 23: התחום מחולק לפחים ריבועיים בגודל המשבצות באלגוריתם. הפח האפור מלא, בפח מתחתיו יש שני סוכנים.

לכל סוכן i נשמור סכום וקטורי displacementSum_i שיסכום את כל הזזות שעושים לסוכן זה ביחד לפני שהוא מוזז באמת, בכך כשנזיז אותו הוא ינוע בכיוון שיאפשר לו להתרחק מהסוכנים סביבו (בהנחה שגם הסוכנים האלה יזוזו). שלבי השיטה הם כך:

1. עבור כל סוכן i בקבוצה, נכניס אותו לפח j בתנאי שמתקיים כי מיקום הסוכן נמצא בתחום הפח. אם הפח המתאים לסוכן כבר מלא (הפח הגיע למכסת הסוכנים) לא נכניס את הסוכן לפח הזה משיקולי ביצועים (ולכן כמובן לא ישתתף בחישוב).
2. עבור כל סוכן i , נבדוק את המיקום שלו ביחס למרכז הפח. נבצע בדיקת התנגשות עם כל הסוכנים שנמצאים בפח של הסוכן הנבחר, וגם בפחים הצמודים לפח הנוכחי בצדדים הקרובים לסוכן הנבחר. בתמונה 24 ישנה דוגמה לכך. הסוכן המסומן בשחור נמצא ברביע הימני העליון של הפח הנבדק. לפי כך הפחים הקרובים אליו הם מימין, מלמעלה, ובאלכסון ימין ולמעלה מהפח הנוכחי. לכן הבדיקה מתבצעת בין הסוכן המסומן לבין כל הסוכנים שנמצאים בפח של הסוכן ובכל הפחים המסומנים.



תמונה 24: הסוכן הנמצא ברביע הראשון של הפח המרכזי נבדק להתנגשויות מול כל הסוכנים הנמצאים בתחום האפור.

3. לכל זוג i, k סוכנים עוברים מתקיים $distance_{ik} < radius_i + radius_k$ (המרחק בין מרכזיהם קטן מסכום הרדיוסים שלהם), נוסף ל- $displacementSum_i$ את הערך

$$displacement_{ik} = \frac{radius_i + radius_k - distance_{ik}}{2} * \frac{position_i - position_k}{distance_{ik}}$$

ערך זה הוא מחצית הגודל הרדיאלי של החפיפה בין הסוכנים (המרחק על קו החיבור של מרכזי הסוכנים, שבו הם חופפים). לפני שמוסיפים ערך זה לסכום המדובר צריך לבדוק האם הוקטור החדש יזיז את הסוכן לתוך קיר. ז"א אם המיקום

$$oldPosition_i + displacement_{ik} + radius_i * direction_{ik}$$

או המיקום

$$oldPosition_i + displacement_{ik} + displacementSum_i$$

נמצא בתוך קיר (בדיקה פשוטה של האם הפוטנציאל במיקום החדש הוא אינסוף). אם כן, לא נוסף את $displacement_{ik}$ ל- $displacementSum_i$, אחרת כן נוסף.

4. לאחר הסכימה נעדכן את המיקום של כל סוכן i בסכום המתאים לו:

$$newPosition_i = oldPosition_i + displacementSum_i$$

ישנו מקרה קצה שבו שני סוכנים נמצאים בדיוק באותו מקום. מקרה זה מופיע רק בהתחלה בתנאי ששני הסוכנים מוקמו כך ולכן נדיר ביותר. הפתרון למקרה זה הוא להזיז אותם מעט באקראי בכיוונים שונים ובכך לאפשר להם לבצע את סדרת הפעולות הנ"ל ללא בעיה. כיוון שזו פעולה פשוטה ולאחריה ההסבר הקודם רלוונטי, לא נרחיב עליה.

4.3 הרצות האלגוריתם

בחלק זה נשתמש בגירסה של האלגוריתם שנבנתה במנוע Unity על מנת לבדוק איך האלגוריתם מגיב ואילו תופעות מתרחשות במספר מצבים ועם פרמטרים שונים.

4.3.1 תיאור התוכנה

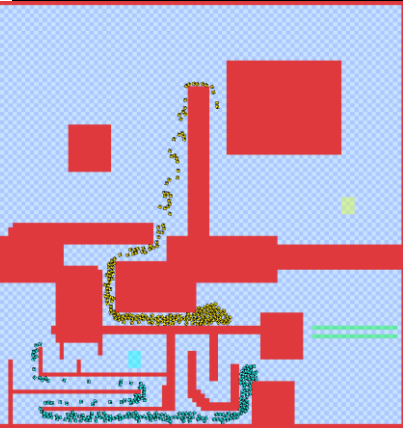
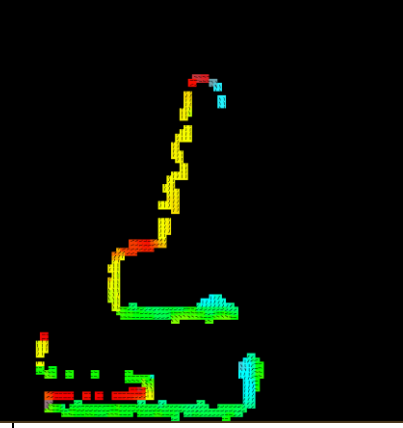
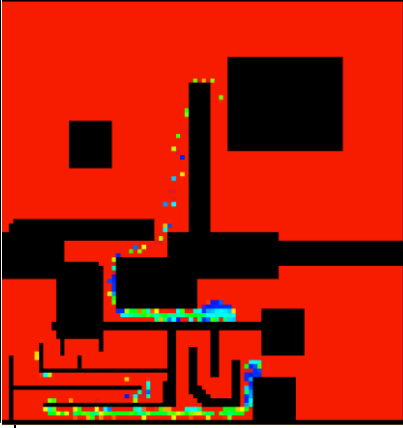
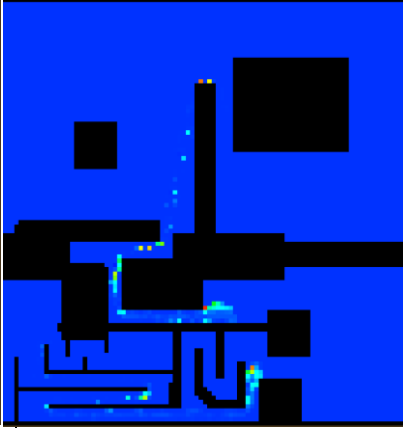
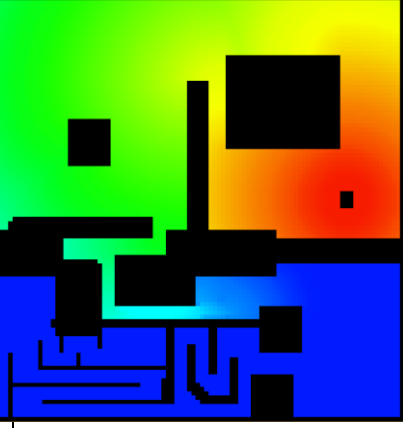
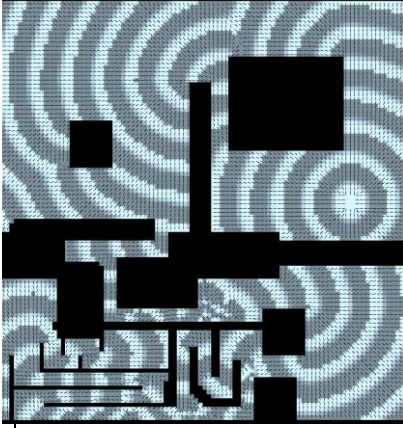
התוכנה בנויה כפי שמסבר בפרק זה ובמאמר המקורי. ניתן להתחיל, להשהות, לאתחל ולהתקדם צעד צעד בסימולציה. המידע במיקום העכבר מציג נתוני שדות לגבי המשבצת המוצבעת. ישנן שתי קבוצות של סוכנים שניתן לשנות את גודלן, אזור תחילת התנועה והיעד שלהן, לשנות את המהירות המקסימלית וניתן גם להציב סוכנים ידניים. לכל קבוצה יש משקלים נפרדים למרחק ליעד, זמן ההגעה והאי-נוחות (בחשוב המחיר). ניתן לשנות את הצפיפות הקריטית וצפיפות הפקק (לכל הקבוצות יחד, כי צפיפות היא מקדם אובייקטיבי). ניתן להפעיל ולכבות את תוסף האי-נוחות לחיזוי תנועה ולשנות את מספר הצעדים העתידיים לחיזוי. ניתן להפעיל ולכבות את מנגנון מניעת ההתנגשויות. ניתן להעלים את הסוכנים בכל אחת מהקבוצות.

בנוסף לתכונות הרגילות שהופיעו במאמר, ישנם כמה תוספים (אשר לא משפיעים על צורת הריצה של האלגוריתם אם לא מפעילים אותם):

1. לכל קבוצה ישנו מקדם צפיפות המאפשר לשנות עד כמה הסוכנים בקבוצה ישפיעו על שדה הצפיפות. ההנחה היא שצפיפות היא פרמטר אובייקטיבי ולכן ישנו שדה צפיפות יחיד המכיל את תוצאות חישובי הצפיפות. הסיבה למקדם זה היא שיהיה אפשר להבדיל בין "גדלים" של סוכנים מבלי לשנות את הרדיוס שלהם – דבר שיגרום לצריכת יותר זמן בשלב יצירת שדה הצפיפות. לדוגמה, אנשים שנושאים תיקים, שקיות וכדומה תופסים יותר מקום מאשר אנשים שלא נושאים כלום, ולכן ישפיעו יותר על הצפיפות. המקדם מוכפל ישירות בצפיפות בכל משבצת.
2. אנו מוסיפים הנחה כי אי-נוחות היא סובייקטיבית ולכן לכל קבוצה יש שדה אי-נוחות נפרד, ובנוסף מקדם אי-נוחות אשר משפיע על הקבוצה האחרת. הרעיון מאחורי מקדם זה הוא שיש קבוצות שדרכן רוב האנשים לא רוצים לעבור, כביכול כמה 'מאיימת' הקבוצה על האחרת. המקדם משתתף בחישוב שדה אי-הנוחות כמכפלה של היחס בין המקדם של הקבוצה האחרת לקבוצה הנוכחית, כך שההשפעה יחסית לכמה 'מאיימים' סוכני הקבוצות נראים אחת אחד לשני. ניתן לשנות את שדות האי-נוחות של כל קבוצה בנפרד.
3. לכל קבוצה ישנם משקלים נפרדים משל עצמם.

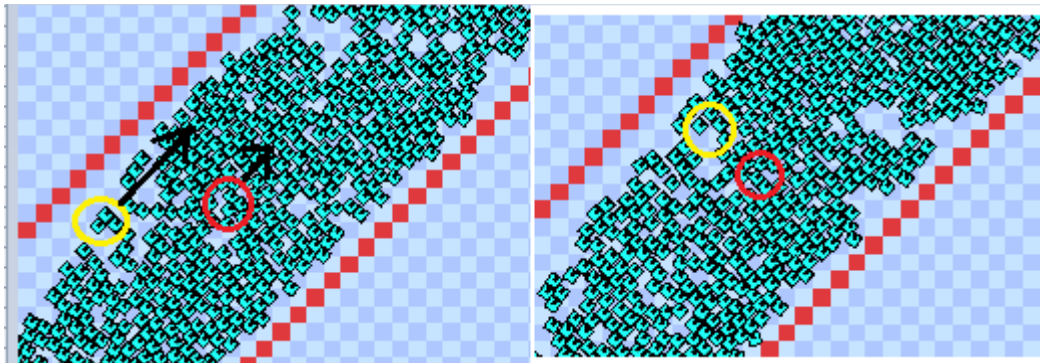
למפה יש 9 מצבי תצוגה: מפה נקייה (רק קירות ואזורי התחלה ויעדים), מפת אי-נוחות, מפת צפיפות, מפת וקטור מהירות ממוצעת, מפת מהירות אניסטרופית, מפת מחיר אניסטרופי, מפת פוטנציאל, תצוגה של אזורי פוטנציאל קבוע (גלים), ומפת וקטור הגרדיאנט. למפות הסקלריות יש תצוגת גרדיאנט צבע שאת גבולותיו ניתן לשנות. למפות הוקטוריות יש גרדיאנט מעגלי שצבעו מורה על הכיוון של הוקטור. המפות האניסטרופיות הן מפות סקלריות שעבורן ניתן לשנות את הכיוון המוצג (צפון, דרום, מזרח ומערב).

מעל כל תצוגה ניתן להראות שני סוגים של מפות שיפועים: מפת וקטורים של שדה המהירות הממוצעת, ומפת וקטורים של הכיוון הנגדי לפוטנציאל (כיוון התנועה).

		
תמונה 25: שדה צפיפות	תמונה 26: שדה אי-נוחות של הקבוצה הראשונה (סוכנים לא מוצגים)	תמונה 27: מפה נקיה עם סוכנים, קירות אזורי התחלה בירוק ואזורי יעד בכחול וצהוב (בהתאם לקבוצה)
		
תמונה 28: שדה מחירים, בכיוון מערב	תמונה 29: שדה מהירות מחושבת, בכיוון מערב	תמונה 30: שדה מהירות ממוצעת עם תצוגת וקטורים
		
תמונה 31: שדה הגרדיאנט עם תצוגת וקטורי גרדיאנט שלילי	תמונה 32: שדה קווי פוטנציאל קבוע עם תצוגת וקטורי גרדיאנט שלילי	תמונה 33: שדה הפוטנציאל של הקבוצה הראשונה

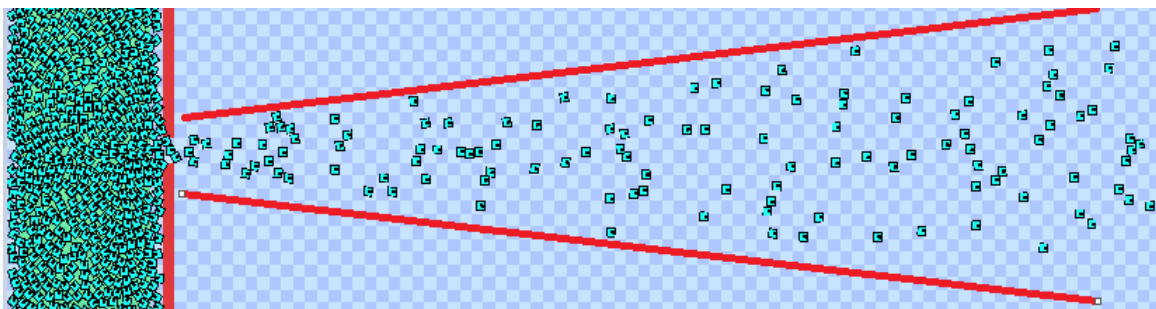
4.3.2 תופעות מתהוות

בחלק זה נציג תופעות הקורות בתנועת המון אמיתי אשר מופיעות גם במודל הנ"ל. הראשונה היא אפקט הקצה, שבו יחידות הקרובות לקצה התנועה במצב של צפיפות גבוהה, נעות מהר יותר מאשר יחידות במרכז.



תמונה 34: אפקט הקצה. היחידות ליד הקירות נעות מהר יותר מהיחידות במרכז הזרימה.

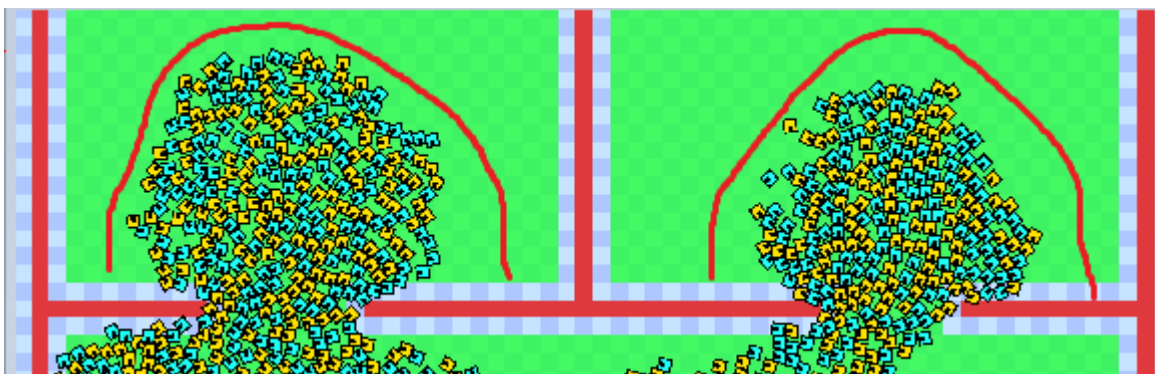
הסיבה לאפקט זה היא שהצפיפות בצדדים נמוכה יותר מהצפיפות במרכז, ובנוסף נדמה כי יחידות מהירות נוטות לעבור בצדדים בגלל אותה צפיפות נמוכה. התופעה השנייה שנראה היא אפקט הגל, שבו כאשר מספר רב של יחידות מנסות לצאת דרך מעבר מעבר צר, נוצר מעין קונוס של יחידות מהצד השני של המעבר כי כשהיחידות עוברות את המעבר הן מתפזרות באופן הדומה לגל.



תמונה 35: אפקט הגל ואפקט הפקק

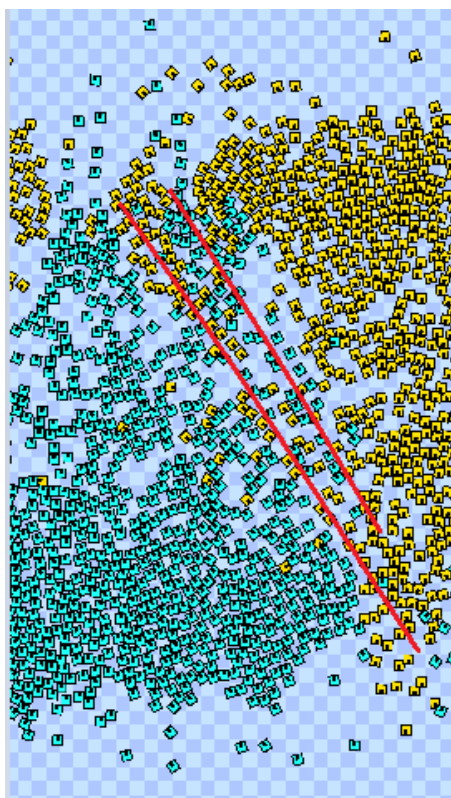
תמונה זו מראה גם את אפקט הפקק במעבר הצר.

התופעה השלישית היא התאגדות הסוכנים במעברים צרים בצורה מעגלית.



תמונה 36: התאגדות קמורה של סוכנים במעבר צר

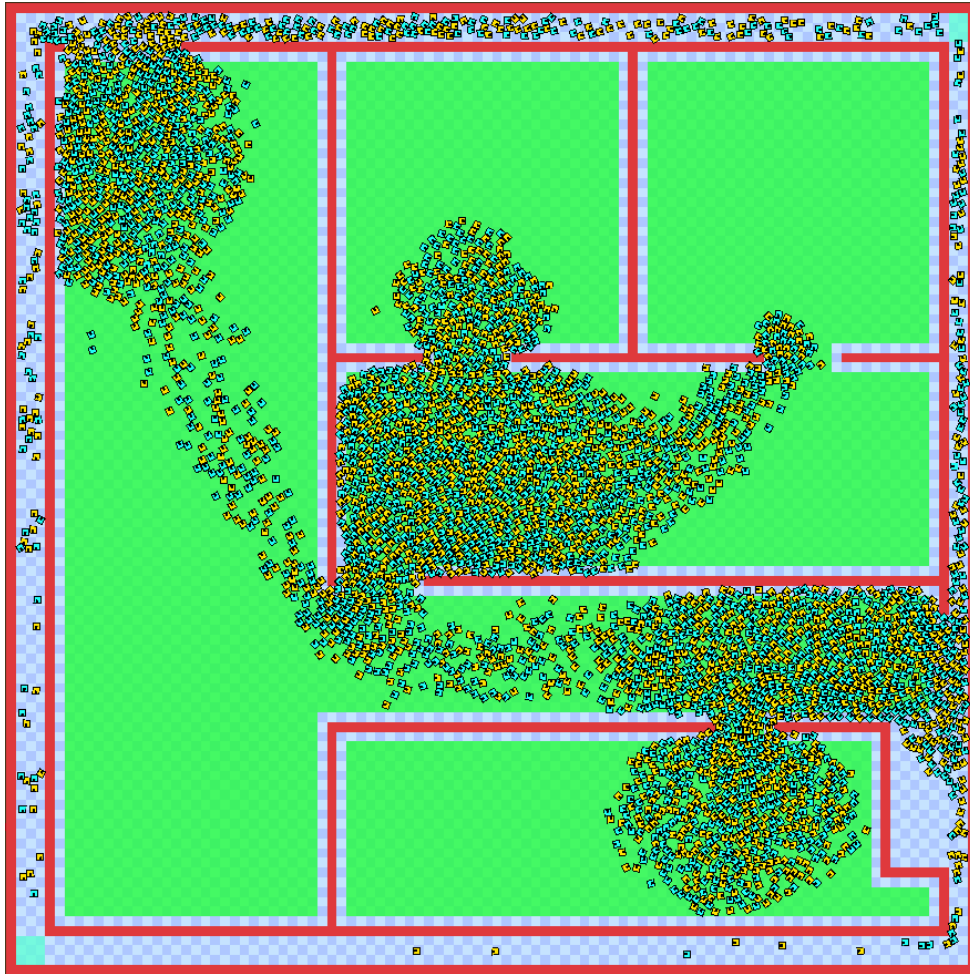
התמונה הבאה מתארת יצירת שבילים של סוכנים. תנועת סוכנים מהירה יותר כאשר הם נעים בזרימה המכוונת בכיוון התנועה שלהם, ולכן מעדיפים להצטופף ביחד ליצירת נתיבים של זרימה, אשר יוצרים אפקט של שבילים.



תמונה 37: אפקט השביל

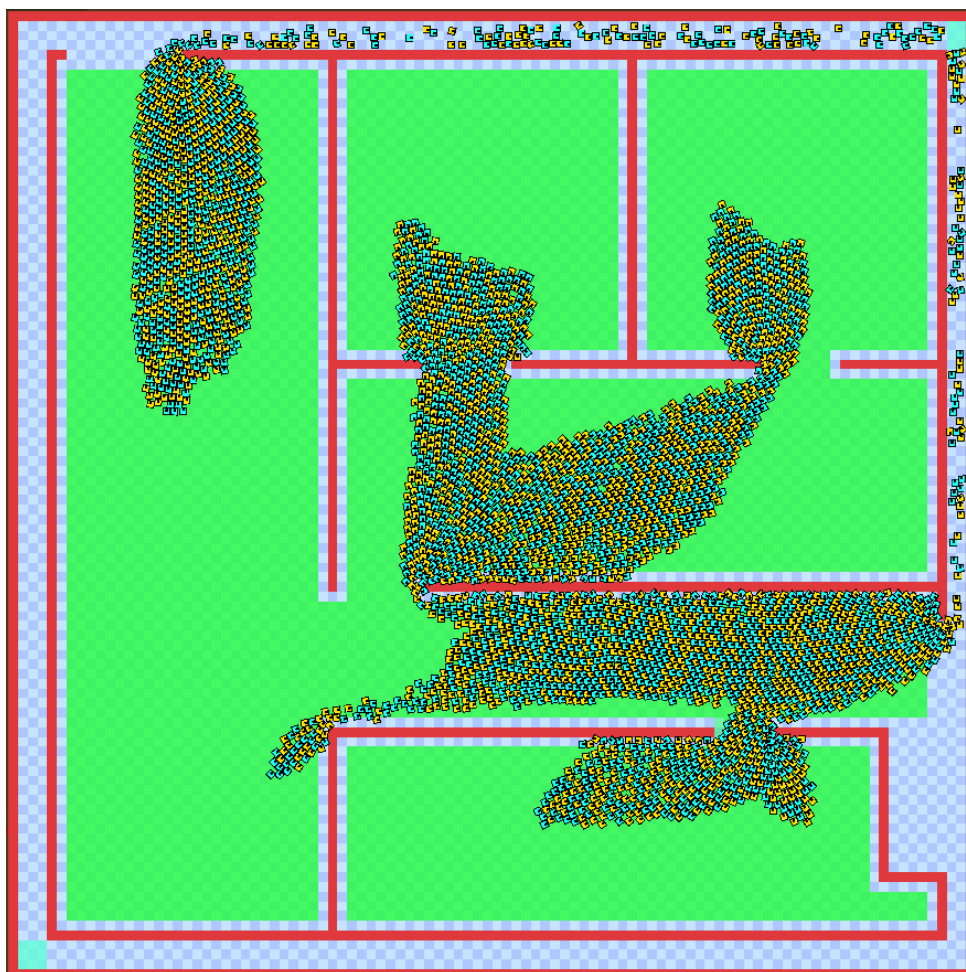
4.3.3 השפעת המשקלים על התנועה

בחלק זה נראה איך המשקלים של פונקציית המחיר משפיעים על תנועת הסוכנים. התמונות הן לאחר כ-500 צעדי אלגוריתם. משקלים שווים (1,1,1): כאשר כל המשקלים שווים מקבלים תנועה שבה לסוכן אין העדפה בין מרחק, זמן הגעה ואי-נוחות. שדה המחירים שיחושב יקח בחשבון את שלושת הקריטריונים באופן שווה. כתוצאה מכך סוכנים עלולים לקחת מסלול שזמן ההגעה בו יותר ארוך אבל המרחק יותר קצר ויותר "נוח" לעבור בו. כיוון שאנו משתמשים באי-נוחות רק על מנת לחזות תנועה עתידית האפקט המתקבל משדה האי-נוחות דומה לשדה הצפיפות אבל מספר צעדים בעתיד.



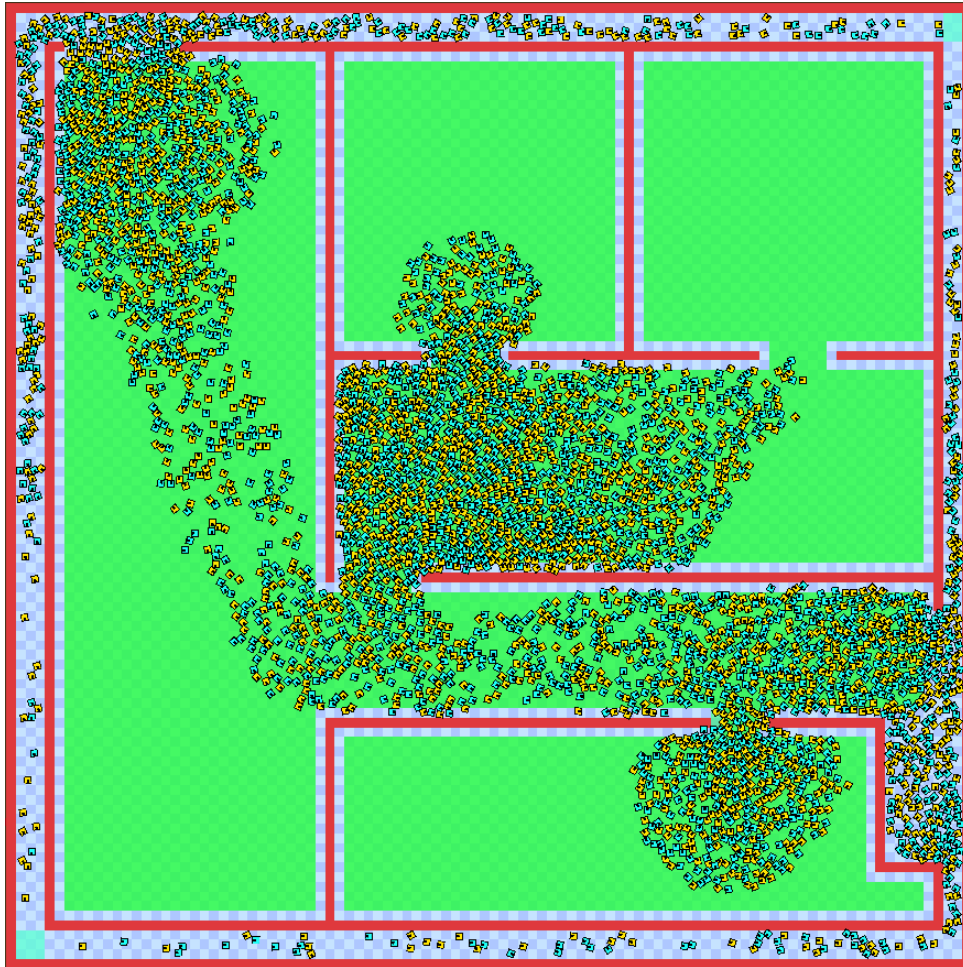
ניתן לראות שהסוכנים בוחרים בין היציאות בהתאם למרחק שלהם מהן, ובהתאם לתנאי הצפיפות. לאחר שהמעבר העליון והימני מתמלאים מספיק, הם מתחילים ללכת במעברים השמאלי והתחתון כיוון שכעת המרחק הנמוך לא שווה את המן שייקח להגיע ליעדים (שמאל תחתון וימין עליון).

פונקציית מחיר התלויה רק במרחק $(1,0,0)$: הסוכנים מחפשים למזער את המרחק מהמטרה שלהם ללא קשר בזמן ההגעה. לכן הסוכנים יחכו כמה זמן שיתאפשר בכדי לקצר את המסלול שלהם.



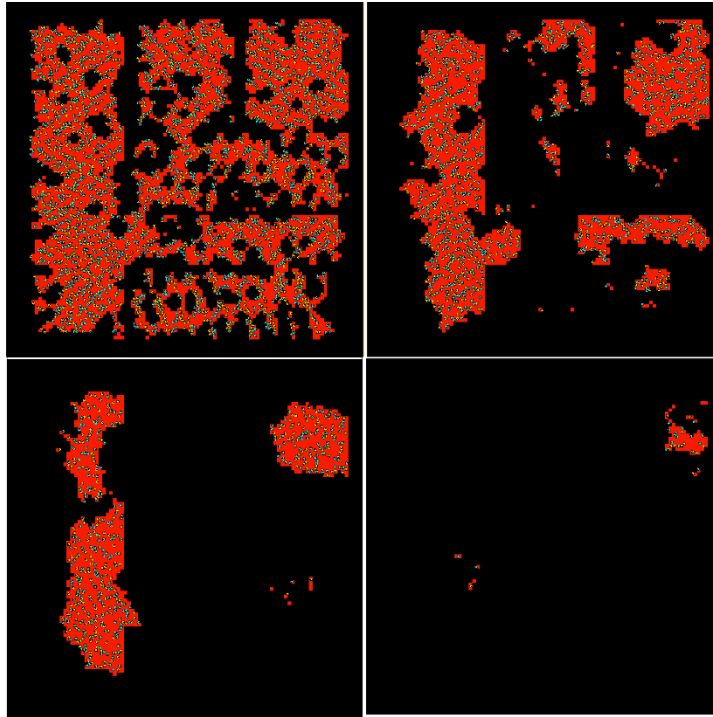
ניתן לראות את הסוכנים מתאגדים בפילות ומעברים צרים ללא תנועה עד אשר הדרך מתפנה, הרי כל תנועה מיותרת תאריך את המסלול. במקום קשתות קמורות במעברים מתקבלות צורות חדות כיוון שהסוכנים אינם מנסים לעקוף את האזורים היותר דחוסים.

פונקציית מחיר התלויה רק בזמן $(0,1,0)$: במצב זה הסוכנים מחפשים למזער את זמן ההגעה המיידית שלהם. זה חשוב כיוון שהם לא בהכרח יגיעו בסופו של דבר יותר ליעד מהסיבה שמצב השדות משתנה מרגע לרגע.

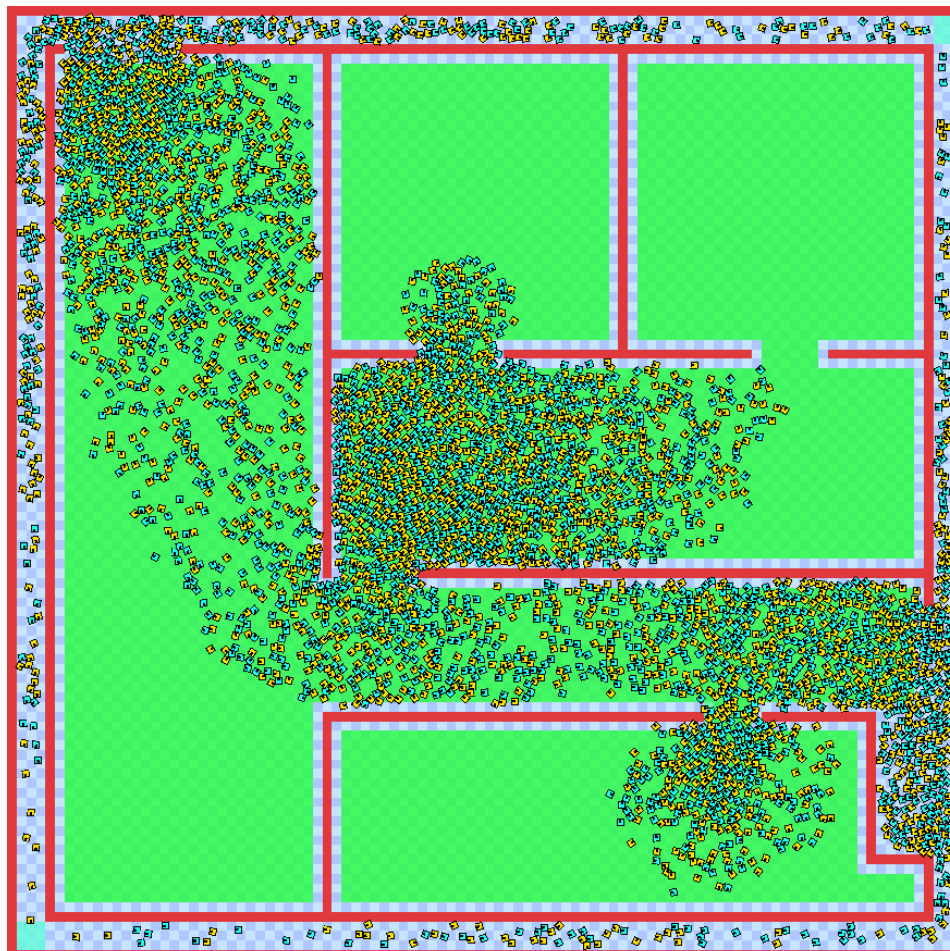


ניתן לראות המון תנועה של הסוכנים בקשתות מאחורי מעברים צרים. הסוכנים לא מתחשבים באורך המסלול ולכן אין מה שעוצר אותם מלנוע כך בחיפוש אחר המסלול שייקח הכי מעט זמן.

פונקציית מחיר התלויה רק באי-נוחות $(0,0,1)$: אי-נוחות משפיעה על הפוטנציאל רק דרך החיזוי שאנו מבצעים איתה, כך שניתן לפרש אי-נוחות כסוג של צפיפות עתידית. סוכנים ינסו באופן כללי להימנע מאזורים צפופים כיוון שהאי-נוחות שם גבוהה. יש לחלק את המצב לשני מקרים. במקרה הראשון, אי-נוחות באזור ללא סוכנים מוגדרת כאפס. במקרה זה, בכל מקום שבו אין סוכנים הפוטנציאל יהיה גם אפס, מה שיגרום לסוכנים להסתובב עד היעלמות (כי פוטנציאל אפס אמור להיות מוגדר רק ביעד). מקרה זה מראה למה מודל Hughes הגדיר את האי-נוחות להיות גדולה מ-1, אחרת תהיה בעיה בחישוב המסלול כי הפוטנציאל לא בהכרח יהיה גדול מאפס באזורים שהם לא היעד. איפה שהפוטנציאל מחושב כ-0 הסוכנים יחשבו שזה חלק מהמטרה ולכן ייעלמו.

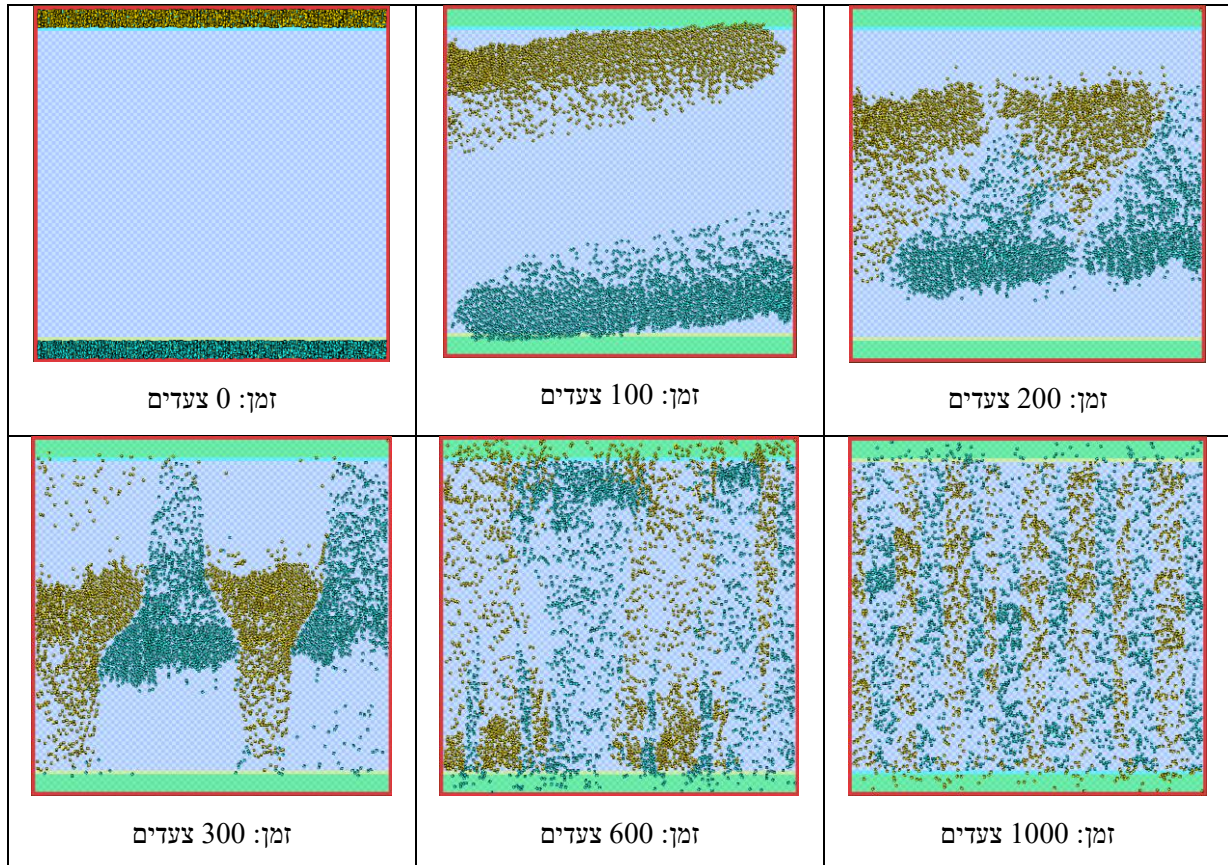


במקרה השני, אנו מגדירים את האי-נוחות הבסיסית ל-1. ניתן לראות שהסוכנים נעים לעבר היעד ללא התחשבות באורך המסלול (הסוכנים מסתובבים הרבה ללא סיבה) או בזמן (הסוכנים משנים כיוון לעיתים תכופות). הסוכנים יבחרו בכל נקודת זמן את המסלול הפחות עמוס למטרה. תוצר לוואי של מצב זה הוא שכל המסלולים ליעדים עמוסים פחות או יותר ברמה שווה.

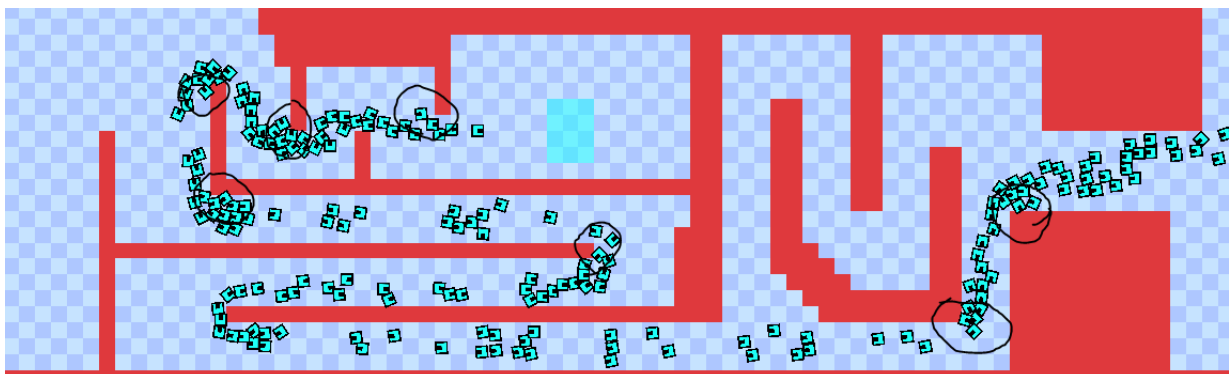


4.3.4 דוגמאות

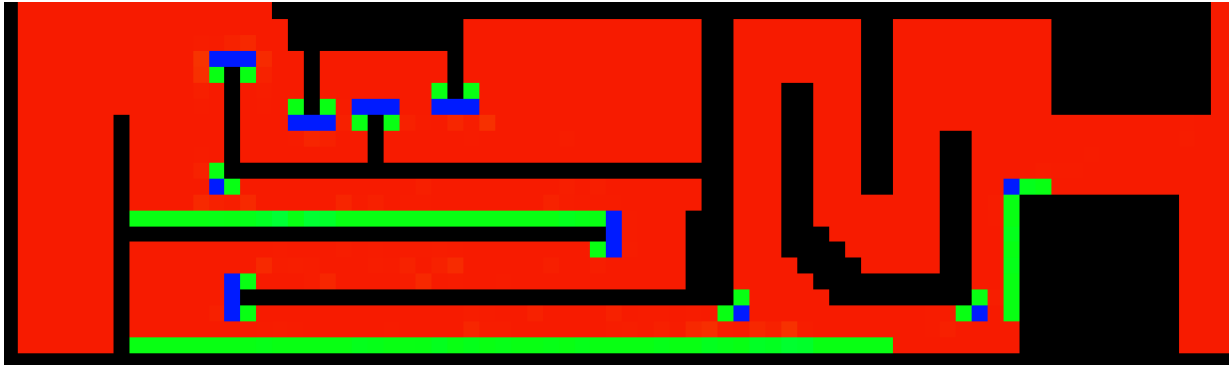
דוגמה 1 – מעבר של שתי קבוצות אחת דרך השנייה. בדוגמה זו נראה איך שתי קבוצות עוברות אחת דרך השנייה ונוצרים נתיבים.



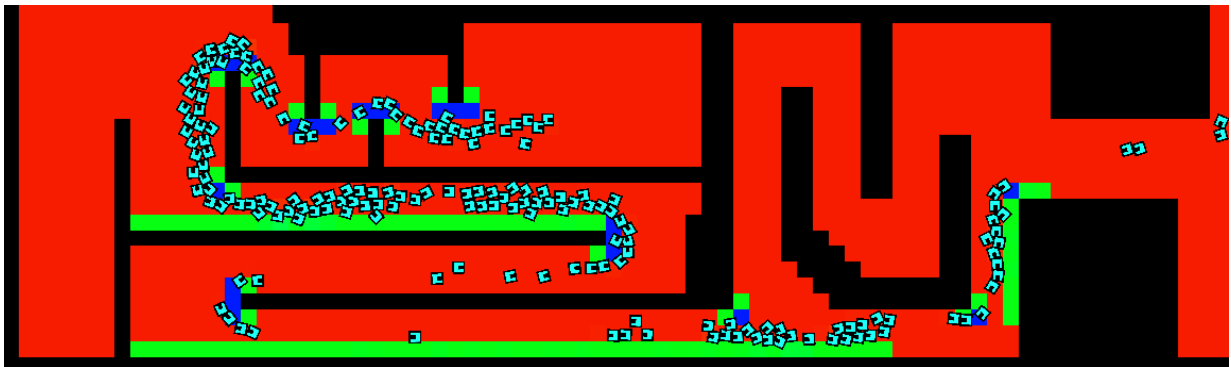
דוגמה 2 – שימוש בשדה האי-נוחות לפתרון בעיית סיבובים ליד קירות. הבעיה נוצרת כתוצאה משיטת התנועה שנבחרה לסוכנים. בדוגמה זו נראה כיצד הצבת אי-נוחות גבוהה סמוך לקירות מחליקה את התנועה לידם. בתמונה הבאה מוקפים בשחור סוכנים שמתנהגים בצורה מוזרה ליד קירות ופינות.



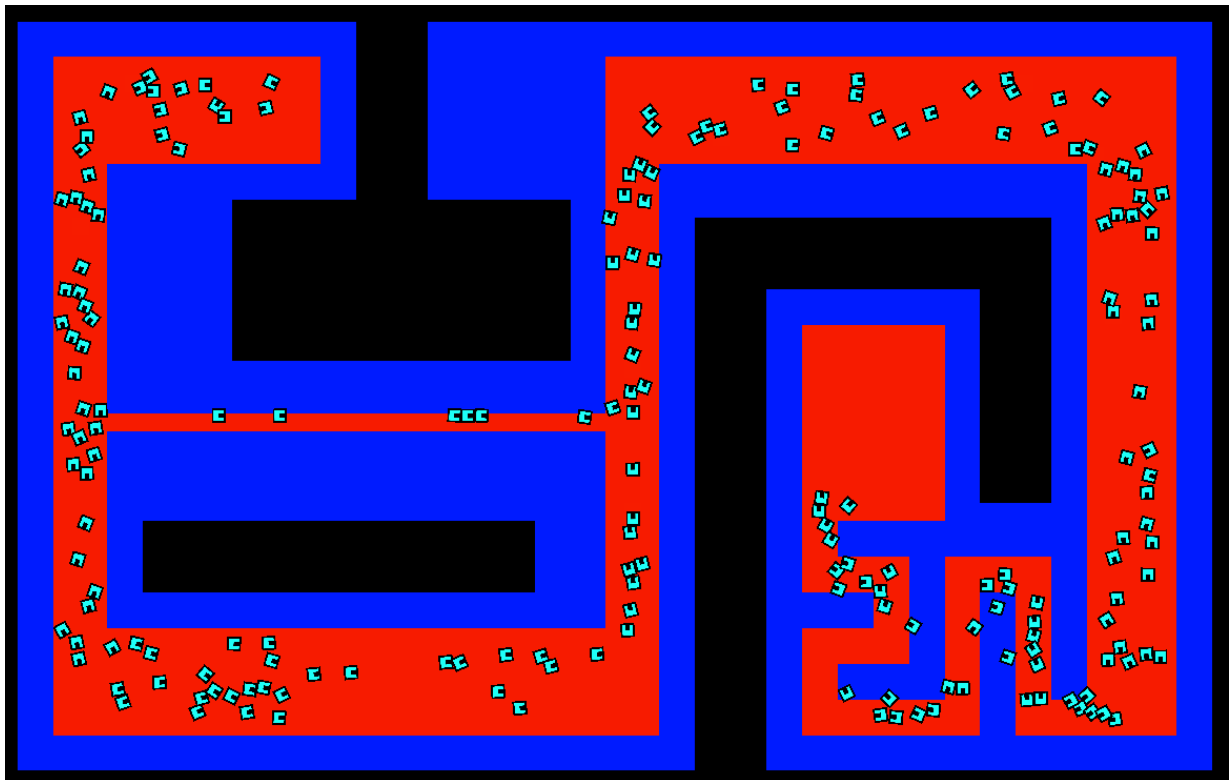
בתמונה הבאה הוספו אזורי אי-נוחות ליד קירות ופינות בעייתיות:



התוצאה היא תנועה חלקה יותר בפניות, מה שמלמד שאולי כדאי להגדיר אזורי אי-נוחות ליד קירות על מנת לקבל תנועה יותר מציאותית:



דוגמה 3 – שימוש בשדה האי-נוחות בכדי ליצור מדרכה. על-ידי שימוש בשדה האי-נוחות ניתן לגרום לסוכנים להעדיף לעבור במסלולים מסוימים. ניתן לראות שכאשר המסלול הצר מתמלא מספיק, הסוכנים יבחרו לעבור מסביב.



דוגמה 4 – כביש דו סטרי. בזמן שהתנהגות ההכוון שנבנתה עבור האלגוריתם לא מתאימה לתנועת כלי רכב, עדיין ניתן לבנות מצב בו ישנם שני כיוונים שונים שבהם הסוכנים "נוסעים", ומתקיימות תופעות כגון עקיפה של סוכנים אישיים (לעיתים נדירות תוך כדי מעבר לנתיב הנגדי) ויצירת שיירות של מכוניות איפה שלא ניתן לעקוף.



5 סיכום

עבודה זו הציגה גישה יישומית למידול של תנועת המון במחשב. באמצעות מפות זרימה ניתן לנווט מספר רב של סוכנים ליעד משותף ללא חישוב נפרד של המסלולים. המודלים המתמטיים השייכים לגישה המאקרוסקופית משתמשים במשוואות הקשורות לזרימת נוזלים, ולכן ניתן לתאר תנועה של המון כנוזל לא דחיס. מודל Hughes שעליו דיברנו נותן קשר בין מודלים של זרימת המון לבין המשוואה האייקונומית, בה ניתן להשתמש על מנת ליצור שדה זרימה במחשב. בסוף הוצג אלגוריתם המשתמש באותה משוואה אייקונומית על מנת להשיג סימולציה של תנועת המון במחשב במהירויות סבירות.

פרק 4.3 הראה כיצד האלגוריתם מגיב ואילו תופעות הוא מסוגל ליצור. הדוגמאות המובאות בפרק זה מראות את חשיבות המשקלים בפונקציית המחיר, ושכאמצעות שינוי של אותם משקלים ניתן לדמות מספר רב של התנהגויות המון שונות. למשל – סוכנים המחכים במעבר צר (בקשת) נוטים לזוז הרבה והצבת משקל נמוך יותר לזמן או גבוה יותר למרחק יכול להקנות תנועה יותר רגועה.

למרות זאת, האלגוריתם לא מאפשר שליטה גרנדולרית על הסוכנים, כיוון ששינוי במשקלים משפיע על כל הסוכנים (דבר שנפתר ברמת הקבוצות על-ידי הקצאת משקלים נפרדים לכל קבוצה). בנוסף לכך, דוגמה לתופעה שלא קיימת במודל היא גלי "עצור ולך". במצב בו הצפיפות גבוהה מצפיפות הפקק, תנועה של המון מפסיקה להתנהג בצורה דומה לנוזל ומוחלפת במעיין תנועה דיסקרטית שבה ההמון עוצר לרגע ואז נע קדימה, ושוב. לא ניתן למדל תופעה זו באמצעות שיטות ממעלה ראשונה [21] ולכן זה גם לא מופיע במודל Hughes או ב-Continuum Crowds.

הכלי מתאים במיוחד לאזורים קטנים יחסית כיוון שההנחה (הקיימת במודל עליו הוא מבוסס) שהסוכנים יכולים לראות את כל אזור הסימולציה בכל נקודת זמן (משמע שיש להם את כל המידע). לא הגיונית באזורים גדולים ומלאים במכשולים גבוהים (כגון בניינים). ניתן לבנות מודל התפשטות מידע בין סוכנים כך שאזור יעדכן את הפתרון שלו למשוואה האייקונומית רק אם קיבל הוראה לעשות זאת.

הכלי לא בנוי לסימולציה של תנועת כלי רכב, אבל אם מוסיפים מודל של תאוצה (דבר פשוט יחסית להוספה), והתנהגות הכוון שונה עבור כלי רכב, ניתן להשיג תנועה מציאותית יחסית בתנאים של כביש סואן.

לסיכום, אלגוריתם זה מאפשר למדל קהל בצורה סבירה ברמת המאקרו. הסוכנים מגיבים לשינויים במסלול באופן הגיוני והתנועה בגדול מציאותית. האינטרקטיביות של האלגוריתם היא יתרון חזק כיוון שניתן לבנות מצבים בסביבה דינמית ולבדוק אותם בזמן אמת. התוצאה המתקבלת ממנו טובה בעיקר למחקר מחשב ולסצנות בסרטים. אין סימולציה של הפסיכולוגיה האנושית ולא ניתן להסתמך על האלגוריתם לקבלת תוצאות מדויקות ברמת המיקרו. ניתן לצפות בעזרתו תוצאות תנועה ברמת המאקרו (למשל בשינוי הרוחב של יציאות חירום) וגם האלגוריתם מאפשר סימולציה מהירה, אינטרקטיבית ובזמן אמת. לכן מומלץ להשתמש בכלי זה כמדד ראשוני לבדיקות מהירות כדי לקבל אינדיקציה, ולאחר מכן להשתמש בשיטות איטיות ומדויקות יותר בשביל לבדוק מקרים לעומק. יש לאלגוריתם פוטנציאל לשימוש נרחב בתחום ההנדסה האזרחית ובתחום הצבאי למטרות מידול של קרבות גדולים (כיוון שההנחה שהסוכנים רואים את כל התחום הגיונית באזורים פתוחים).

- [1] E.W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* 1, 269-271, 1959.
- [2] Stephen J. Guy, Jatin Chhugani, Sean Curtis, Pradeep Dubey, Ming Lin, and Dinesh Manocha. PLEdrians: A Least-Effort Approach to Crowd Simulation. *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, 2010. <http://gamma.cs.unc.edu/PLE/pubs/PLE.pdf>
- [3] Dijkstra's algorithm - https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- [4] Craig W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. In *Computer Graphics*, 25-34, 1987. <http://www.red3d.com/cwr/boids/>.
- [5] P. Hart, N. Nilsson and B. Raphael, A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Science and Cybernetics*, SSC-4(2), 100-107, 1968.
- [6] Dave Ferguson, Anthony Stentz. Field D*: An Interpolation-Based Path Planner and Replanner. *Robotics Research/ Results of the 12th International Symposium ISRR*, pp 239-253, 2007.
- [7] Elijah Emerson. Crowd Pathfinding and Steering Using Flow Field Tiles. *GameAIPro*, C23, 307-317, 2013.
- [8] Lighthill, M.J., Whitham, G.B., On Kinematic Waves. I: Flood movement in long rivers. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229, 1955.
- [9] Lighthill, M.J., Whitham, G.B., On Kinematic Waves. II: A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229, 317. 1955.
- [10] Traffic Flow - https://en.wikipedia.org/wiki/Traffic_flow
- [11] Richards, P., Shock waves on the highway. *Operations research* 42-51, 1956.
- [12] Hughes, R.L., A continuum theory for the flow of pedestrians. *Transportation Research Part B* 36 (6), 507-535. 2002.
- [13] Wardrop, J. G.; Whitehead, J. I., "Correspondence. Some Theoretical Aspects of Road Traffic Research". *ICE Proceedings: Engineering Divisions*. 1 (5): 767. 1952.
- [14] Huang, L., Wong, S., Zhang, M., Shu, C.W., Lam, W., Revisiting Hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm. *Transportation Research Part B: Methodological* 43, 127-141.
- [15] James A. Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. In *Proc. Nat. Acad. Sci*, 1591-1595, 1995. <http://www.imm.dtu.dk/~rlar/04351/lectures/LevelSets/1591.pdf>
- [16] James A. Sethian and Alexander Vladimirsky. Ordered Upwind Methods for Static Hamilton–Jacobi Equations: Theory and Algorithms. *SIAM J. Numer. Anal.*, 41:325–363, 2003.
- [17] Bærentzen, J.A., On the implementation of fast marching methods for 3D lattices. *Informatics and Mathematical Modeling*, Technical University of Denmark. 2001. http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/841/pdf/imm841.pdf
- [18] Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum Crowds. *ACM Trans. Graph*, 25:1160–1168, 2006. <http://grail.cs.washington.edu/projects/crowd-flows/>
- [19] Clément Pêtrès, Yan Pailhas, Yvan Petillot, and Dave Lane. Underwater Path Planning Using Fast Marching Algorithms. *IEEE Oceans Europe 2005*, June 2005.
- [20] Helmut Duregger, Simulation of Large and Dense Crowds on the GPU using OpenCL. Institute of Computer Science, Faculty of Mathematics, Computer Science and Physics, University of Innsbruck. October 18, 2011.
- [21] Monima Twarogowska, Paola Goatin, Regis Duvigneau, “Comparative study of macroscopic pedestrian models”, The conference on Pedestrian and Evacuation Dynamics 2014, *Transportation Research Procedia* 2 (2014) 477-485, 2014