

Coordinated- Motion-Planning

מנחה: ד"ר גבריאל ניבש

מגישים: אלישיב אלימלך, אוהד כהן, עמית אלבז

<https://github.com/amit1021/Coordinated-Motion-Planning/tree/CMP> :Git

רקע

- תחרות עולמית בפתרון בעיות אופטימיזציה גאומטריות קשות.
- תכנון תנועה מרובות אובייקטים.
- המנצחים נקבעו על ידי שתי קטגוריות שונות:
 - פתרון הבעיה בכמה שפחות צעדים.
 - פתרון הבעיה בכמה שפחות יחידות זמן.

תיאור

- קיים לוח M בגודל $m \times m$ ועליו n רובוטים, ו- n מיקומים סופיים אליהם צריכים להגיע הרובוטים, ו- x מכשולים.
- קבוצת הרובוטים - $A = \{a_1, \dots, a_n\}$
- קבוצת המיקומים הסופיים של הרובוטים - $B = \{b_1, \dots, b_n\}$
- בתחילת המשחק כל רובוט ממוקם במקום M_{ij} כלשהו.
- המשחק יסתיים כאשר כל רובוט a_i יגיע אל מיקומו הסופי b_i
- בכל יחידת זמן הרובוטים מקבוצה A יוכלו לזוז לכיוון צפון, דרום, מזרח או מערב במהירות אחידה כך שהם לא נפגשים בשום רובוט על הלוח, לא במהלך יחידת הזמן ולא בסופה.

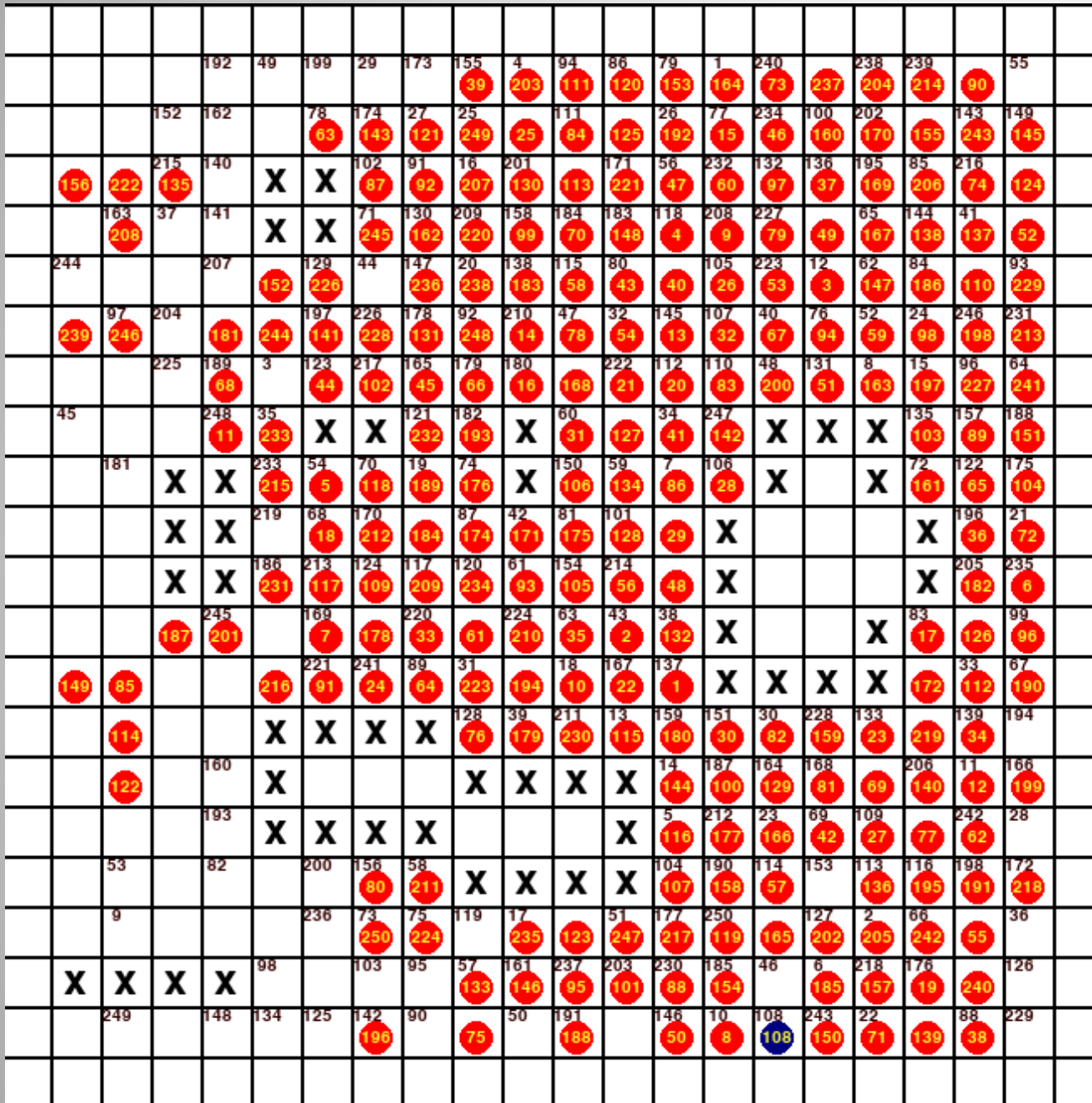
מטרה

הרובוטים מקבוצה A יגיעו למיקומם הסופי
מקבוצה B בכמה שפחות צעדים.

עיגול אדום – רובוט שלא הגיע ליעד.

עיגול כחול – רובוט שהגיע ליעד.

מספר – מיקומו הסופי של הרובוט עם המספר
X – מכשול



אלגוריתם

- הגדלת הלוח בהתאם למספר השורות שאליהן נעביר את הרובוטים בעזרת פונקציית הערכה שבנינו.
- מעבר על הלוח והוצאת הרובוטים למסגרת באופן בו יעשו כמה שפחות צעדים.

```
def row_space(number_of_robots, n):  
    # Finds how many lines to add to the board  
    row_space = 0  
    squares = 0  
    while number_of_robots > 0:  
        number_of_robots -= (n - squares) * 4  
        squares += 2  
        row_space += 1  
    return row_space + 2
```

אלגוריתם

- מעבר על הלוח בצורת ספירלה (מבפנים כלפי חוץ) להזזת הרובוטים למקומם הסופי.

```
if eb.board_final_state[low_row][i] >= 1:
    robot = find_robot_by_number(eb.board_final_state[low_row][i])
    if robot != -1:
        robot_queue_node = bfs(eb.board, robot.current_place, robot.end_place)
        if robot_queue_node != -1:
            number_of_steps += move_robot(eb, robot, robot_queue_node)
        else:
            get_robot_out(eb, robot)
```

אלגוריתם

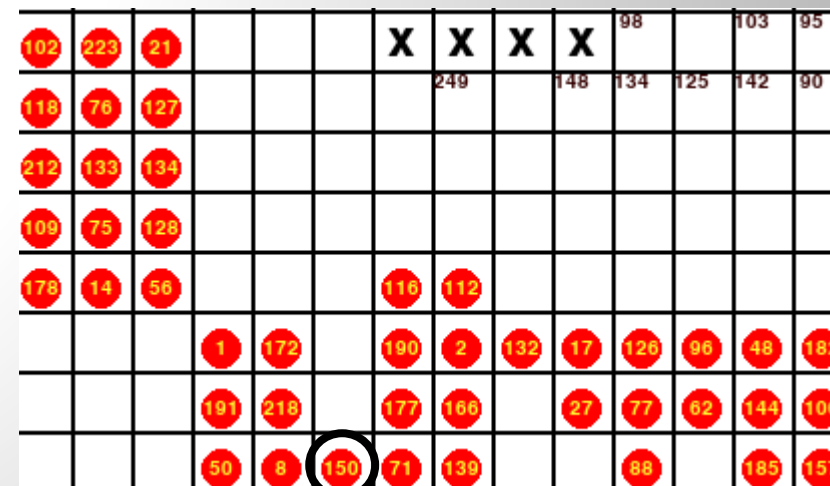
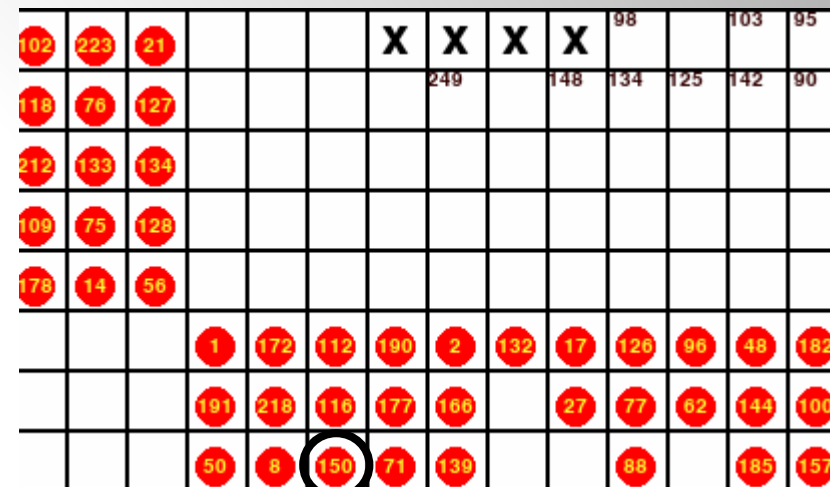
- במקרה בו לרובוט אין מסלול פנוי (חסום ע"י רובוטים אחרים וע"י מכשולים), הרובוטים שחוסמים אותו זזים כמו "צינור" החוצה, הכנסת הרובוט הבעייתי והחזרת הרובוטים למקומם.
קיימים שני מקרים:

1. רובוט לא יכול לצאת מ"מסגרת" הלוח.
2. הדרך אל מיקומו הסופי של הרובוט חסומה.

דוגמא 1

```
while counter > 0:
    # returns the robot
    r = find_robot_by_number(eb.board[p.x + counter][p.y])
    # validity test
    if r.robot_number == 0:
        counter -= 1
        continue
    # create a point to which the robot will move
    point_temp = Point(p.x + number_steps_out + 1, p.y + counter)
    # return the route to this point
    robot_queue_node = bfs(eb.board, r.current_place, point_temp)

    # If there is no path to the point, find a new point, as long as there is no correct path find a new point
    plus_steps = 1
    while robot_queue_node == -1:
        point_temp = Point(p.x + number_steps_out + plus_steps, p.y + counter)
        if is_valid(p.x + number_steps_out + plus_steps, p.y + counter, eb.n, eb.n):
            robot_queue_node = bfs(eb.board, r.current_place, point_temp)
            plus_steps += 1
        else:
            break
    # if there is a way, move the robot to the new point
    if robot_queue_node != -1:
        dest = Point(r.current_place.x, r.current_place.y)
        # add to back list that will later return to the point from which it came
        robot_point = RobotPoint(r, dest)
        robot_prev_place.append(robot_point)
        # move robot to new place
        number_of_steps += move_robot(eb, r, robot_queue_node)
        counter -= 1
```



מצב התחלתי

166	286	144	170	278	194	263	247	119	196	267	173	149	18	34	38	113	120	118	104	32
98	258		285	48	116	310	248	206	39	306	260	241	151		156	129	192	56	315	
186	160	275	235	28	178	259	111	191	201	80	193	157	41		156	202	53	142	2	169
1	79	104		270	143	65	220	232	251	184		X	X	X		199	161	5	264	
133	105	225	192	50	256	19	58	54	229	97	150					6	102	172	46	195
169	155	195	137	272	207	121	213	309	69	88	134				X	244	188	26		
131	189	89	60	237	26	96	211	205	218	25	135	X			X	167	75	179	55	24
181	30	174	51	152	219	271			154	250		X			X	123	72	29		
31	224	42	5	76	44	124	94	232	281	128	164					73	27	143	90	9
175	320		41	91	136	266	165		108	301	293	X			X	287	13	92	83	
212	66	62	145	15	1	253		136	99	84	123	X			X	209	101	187	92	65
29	214	137	269	37	140	182	147	115	125	17	95	X	X	X		145	101	268	62	52
78	92	156	53	126	297	246		X	117	209	254					142	300	135	283	255
221	176	4	72	64	11	117	X	20	83	3	122	79	51	47		207	68	61	158	220
234	176	168	35	47	138	23		X	231	99	21	170	298	107		X	X	X	205	238
103	69	49	86	21	121	45	X	12	197	261	223	181	7	X	X	X		234	246	245
114	71	78	146	175	153	30	X	249	273	250	22	231	57	X		X		193	76	221
242	37	275	49	256	131	105		X	282	18	19	235	222	22		X	X	X	166	128
240	126	93	161	40	208	8	X	162	227	63	127	180	272	X	X	X		254	283	258
291	146	80	307	63	59	313		X	236	319	262	286	54	120	114	215	86	237	243	239
265	185	82	116	151	204	152	X	141	168	87	139	243	138	100	23	198	288	313	305	
320	107	271	309	262	155	177	134	310	33	296	239	159	244	248	206	316		159	150	
268	91	233	59	203	39	108	236	303	311	210	292	293	217		306		89	71	67	33
289	184	252	10	280	88	16	300	56	13	298	132	321	302	255		318		55	253	
213	165	199	301	70	194	230	X	X	190	163	290	208	10	97	284	93	178	149	249	
98	251	85	48	291	200	242	X	X	158	203	180	82	269	28	17	74	24	132	113	
299		74	81		77	219	X	X	141	9	106	75	32	50	81	273	144	42	38	
52	315	282		264	110	287	X		X	X	68	302	182	124	186	245	191	73	167	
	227	308	57	228	4	200	X	X	X	X	46	247	288	16	183	119	25	305	12	
	257	171	148		188	215	X	X	X	X	112	274	277			241				

אתחול המסגרת

		28	12	66	77	98	113	72	149	129	61	73	147	141	187	68	135	197	159	189	144	59	143	23	142	37	56		
		186	84	109	155	46	5	3	22	33	29	126	58	102	151	146	191	40	9	14	152	91	46	153	107	181	63		
120	115																											86	188
106	1																											92	60
121	110																											11	51
136	114										65		49						107	154				30				168	125
134	94								9			106	103	54	110	72	87	33	19	115	183	44						140	67
179	196									160				40	130	5	156	75	125	192	161	21		89				16	185
145	172				162	152		7			14	80	26	151	58	35	64	11	93	51	61	39	69	148				24	74
156	123										70	163	79	48	31	38	23	12	159	96				167				31	10
175	116										143	90	37	86		22	56		195	66	52	73		116				20	76
163	160					172		16	139		47	131	126		74		84	98	28	77	101	146						161	111
117	122								196	94	190	175			166	68	194	81	15	171	42	140	29		3			34	177
96	112									133	109	149			182	17	197	177	10	2	178		88					53	19
154	87								108		24				99		173	20	59		82	41	67					30	7
180	45									113	168	165		45	25	141	188	63	91		76	97	102					36	78
182	182									122	127		135		128		62		111		117	179	85	92				26	43
139	173				169		136					137		187		46	71	60	185		193							21	42
193	195					123		118	176		147			150					32	78	1	43						52	80
130	150							199	174		112				114		18	142	144	119	34		6						13
100	105									198	155			100			83	13		27	132	53	55						70
108	17					181				170					95	50			8		36	184							165
132	176					157				158	189		180			191	4		129		57								198
183	82								121					105			104			124	134		153						54
170	184									186	120		164						145					138					62
174	93																												65
128	190																												32
131	103																												15
		194	2	162	57	71	4	6	167	41	146	137	166	157	50	47	89	81	44	35	99	75	97	199	25	79	38		
		118	171	127	169	95	119	18	64	90	39	133	164	176	88	101	104	27	124	55	136	85	48	8	69	83	158		

מצב סופי

			192	49	199	29	173	155	4	94	86	79	1	240		238	239		55
		152	162		78	174	27	25		111		26	77	234	100	202		143	149
		215	140	X	X	102	91	16	201		171	56	232	132	136	195	85	216	
	163	37	141	X	X	71	130	209	158	184	183	118	208	227		65	144	41	
244			207		129	44	147	20	138	115	80		105	223	12	62	84		93
	97	204			197	226	178	92	210	47	32	145	107	40	76	52	24	246	231
		225	189	3	123	217	165	179	180		222	112	110	48	131	8	15	96	64
45			248	35	X	X	121	182	X	60		34	247	X	X	X	135	157	188
	181	X	X	233	54	70	19	74	X	150	59	7	106	X		X	72	122	175
		X	X	219	68	170		87	42	81	101		X				X	196	21
		X	X	186	213	124	117	120	61	154	214		X				X	205	235
			245		169		220		224	63	43	38	X			X	83		99
					221	241	89	31		18	167	137	X	X	X	X		33	67
				X	X	X	X	128	39	211	13	159	151	30	228	133		139	194
			160	X				X	X	X	X	14	187	164	168		206	11	166
			193	X	X	X	X				X	5	212	23	69	109		242	28
	53		82		200	156	58	X	X	X	X	104	190	114	153	113	116	198	172
	9				236	73	75	119	17		51	177	250		127	2	66		36
X	X	X	X	98		103	95	57	161	237	203	230	185	46	6	218	176		126
	249		148	134	125	142	90		50	191		146	10	108	243	22		88	229

שלב שיפור הצעדים

- לאחר שהצלחנו שכל הרובוטים יגיעו אל יעדם בכל הלוחות, הגענו לשלב שיפור הצעדים.
- בשלב הראשון, רובוטים שכבר היו ביעדם לא הוזזו ממקומם.
- בשלב השני, שיפרנו את המרחק אליו זזו הרובוטים שפינו את הדרך לרובוט ללא דרך.
- בשלב השלישי, שיפרנו את הזזת הרובוטים למסגרת, על ידי חלוקת הלוח לרבעים.

אינדיקציה לרמת שיפור הצעדים

מספר צעדים לאחר השיפור הסופי	מספר צעדים לאחר שכל הלוחות צלחו	מספר לוח
45,071	59,949	2
88,925	104,965	4
47,069	69,909	12
163,033	205,586	153
162,551	220,290	163
28,780	36,997	203