Contents

	name	symbol	value	unit	source	explanation
0	`size`					Number of agents
1	`shape`					Shape for arrays
2	`three_circles_flag`					Boolean indicating if agent is modeled with three circle model
3	`orientable_flag`					Boolean indicating if agent is orientable
4	`active`					Boolean indicating if agent is active
5	`goal_reached`					Boolean indicating if goal is reahed
6	`mass`	m		kg	fds+evac	Mass
7	`radius`	r		m	fds+evac	Radius
8	`r_t`	r_t		m	fds+evac	Radius of torso
9	`r_s`	r_s		m	fds+evac	Radius of shoulder
10	`r_ts`	r_{ts}		m	fds+evac	Distance from torso to shoulder
11	`position`	x		m		Position
12	`velocity`	v		$\frac{\mathrm{m}}{\mathrm{s}}$		Velocity
13	`target_velocity`	v_0	5	$\frac{\mathrm{m}}{\mathrm{s}}$		Target velocity
14	`target_direction`	e				Target direction
15	`force`	f		N		Force
16	`force_adjust`	f_{adj}		N		Adjusting force
17	`force_agent`	f_{agent}		N		Agent to agent force
18	`force_wall`	f_{wall}		N		Agent to wall force
19	`inertia_rot`	I_{rot}	4	$\mathrm{m}^2\,\mathrm{kg}$	fds+evac	Rotational moment
20	`angle`	φ	$[-\pi, \pi]$	rad		Angle
21	`angular_velocity`	ω		$\frac{\mathrm{rad}}{\mathrm{s}}$		Angular velocity
22	`target_angle`	φ_0	$[-\pi, \pi]$	rad		Target angle
23	`target_angular_velocity`	ω_0	4π	$\frac{\mathrm{rad}}{\mathrm{s}}$	fds+evac	Target angular velocity
24	`torque`	M		N m		Torque
25	`position_ls`	x_{ls}		m		Position of the left shoulder
26	`position_rs`	x_{rs}		m		Position of the right shoulder
27	`front`	x_{front}		m		Position of the front
28	`tau_adj`	τ_{adj}	0.5	s	fds+evac	Characteristic time for agent adjusting its movement
29	`tau_adj_rot`	τ_{adjrot}	0.2	s	fds+evac	Characteristic time for agent adjusting its rotational movement
30	`k`	k	1.5	N	power law	Social force scaling constant
31	`tau_0`	τ_0	3	s	power law	Interaction time horizon
32	`mu`	μ	12000	$\frac{\mathrm{kg}}{\mathrm{s}^2}$	fds+evac	Compression counteraction constant
33	`kappa`	κ	40000	$\frac{\mathrm{kg}}{\mathrm{m\,s}}$	fds+evac	Sliding friction constant
34	`damping`	c_d	500	N	fds+evac	Damping coefficient for contact force
35	`a`	A	2000	N	helbing	Scaling coefficient for social force
36	`b`	B	0.08	m	helbing	Coefficient for social force
37	`std_rand_force`	ξ/m	0.1		fds+evac	Standard deviation for random force from truncated normal distribution
38	`std_rand_torque`	η/I_{rot}	0.1		fds+evac	Standard deviation for random torque from truncated normal distribution
39	`f_soc_ij_max`		2000	N		Truncation for social force with agent to agent interaction
40	`f_soc_iw_max`		2000	N		Truncation for social force with agent to wall interaction
41	`sight_soc`		7	m		Maximum distance for social force to effect
42	`sight_wall`		7	m		Maximum distance for social force to effect

	name	adult	male	female	child	eldery	symbol	explanation
0	`radius`	0.255	0.27	0.24	0.21	0.25	r	Total radius of the agent
1	`dr`	0.035	0.02	0.02	0.015	0.02	dr	Difference bound for total radius
2	`k_t`	0.5882	0.5926	0.5833	0.5714	0.6	k_t	Ratio of total radius and radius torso
3	`k_s`	0.3725	0.3704	0.375	0.3333	0.36	k_s	Ratio of total radius and radius shoulder
4	`k_ts`	0.6275	0.6296	0.625	0.6667	0.64	k_{ts}	Ratio of total radius and distance from torso to shoulder
5	`v`	1.25	1.35	1.15	0.9	0.8	v	Walking speed of agent
6	`dv`	0.3	0.2	0.2	0.3	0.3	dv	Difference bound for walking speed
7	`mass`	73.5	80.0	67.0	57.0	70.0	m	Mass of an agent
8	`mass_scale`	8.0	8.0	6.7	5.7	7.0	dm	Standard deviation of mass of the agent

1 2D vectors

Table 1: Some operations for vectors in 2D continuous space.

Notation	Python	Return
$\mathbf{x}$	`array([x, y])`	$(x, y) \in \mathbb{R}^2$
$\lVert\mathbf{x}\rVert$	`hypot`(x, y)	$d \in [0, \infty)$
angle($\mathbf{x}$)	`arctan2`(y, x)	$\varphi \in [-\pi, \pi]$
$R(90°) \cdot \mathbf{x}$		$(-y, x)$
$R(-90°) \cdot \mathbf{x}$		$(y, -x)$

2 Agent

Table 2: Shoulder, torso and total radii.

	Total		Torso	Shoulder	
	r	$\pm$	$k_t = \frac{r_t}{r}$	$k_s = \frac{r_s}{r}$	$k_{ts} = \frac{r_{ts}}{r}$
adult	0.255	0.035	0.5882	0.3725	0.6275
child	0.210	0.015	0.5714	0.3333	0.6667
eldery	0.250	0.020	0.6000	0.3600	0.6400
female	0.240	0.020	0.5833	0.3750	0.6250
male	0.270	0.020	0.5926	0.3704	0.6296

Table 3: Properties

r	m		Total radius
r_t	m		Torso radius
r_s	m		Shoulder radius
r_{ts}	m		Distance from torso to shoulder
m	kg	80	Mass
I	$\mathrm{kg} \cdot \mathrm{m}^2$	4.0	Rotational moment
$\mathbf{x}$	m		Position
$\mathbf{v}$	m/s		Velocity
v_0	m/s		Goal velocity
$\hat{\mathbf{e}}_0$			Goal direction
$\hat{\mathbf{e}}$			Target direction
φ	rad	$[-\pi, \pi]$	Body angle
ω	rad/s		Angular velocity
φ_0	rad	$[-\pi, \pi]$	Target angle
ω_0	rad/s	0.4π	Max angular velocity
p		$0-1$	Herding tendency

2.1 Circular

Table 4: Relative

$\tilde{\mathbf{x}} = \mathbf{x}_i - \mathbf{x}_j$	Relative position
$\tilde{\mathbf{v}} = \mathbf{v}_i - \mathbf{v}_j$	Relative velocity

$d = \|\tilde{\mathbf{x}}\|$	Distance
$\hat{\mathbf{n}} = \tilde{\mathbf{x}}/d$	Normal vector
$\hat{\mathbf{t}} = R(-90°) \cdot \hat{\mathbf{n}}$	Tangent vector

Total radius and relative distance

$$\begin{aligned} \tilde{r} &= r_i + r_j \\ h &= d - \tilde{r} \end{aligned}$$

2.2 Three circles

$$\begin{aligned} \mathbf{x}_r &= \mathbf{x}_c + \hat{\mathbf{t}} r_{ts} \\ \mathbf{x}_l &= \mathbf{x}_c - \hat{\mathbf{t}} r_{ts} \\ \hat{\mathbf{t}} &= \begin{bmatrix} -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \end{aligned}$$

$$\mathbf{r}_{tot} = \begin{bmatrix} r_t & r_s & r_s \end{bmatrix}_i + \begin{bmatrix} r_t \\ r_s \\ r_s \end{bmatrix}_j$$

$$\begin{aligned} \mathbf{d} &= \left\| \begin{bmatrix} \mathbf{x}_c & \mathbf{x}_r & \mathbf{x}_l \end{bmatrix}_i - \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_r \\ \mathbf{x}_l \end{bmatrix}_j \right\| \\ &= \left\| \begin{bmatrix} 0 & \hat{\mathbf{t}} r_{ts} & -\hat{\mathbf{t}} r_{ts} \end{bmatrix}_i - \begin{bmatrix} 0 \\ \hat{\mathbf{t}} r_{ts} \\ -\hat{\mathbf{t}} r_{ts} \end{bmatrix}_j + (\mathbf{x}_i - \mathbf{x}_j) \right\| \\ &= \left\| \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \left(\hat{\mathbf{t}} r_{ts}\right)_i - \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} \left(\hat{\mathbf{t}} r_{ts}\right)_j + \tilde{\mathbf{x}} \right\| \\ &= \left\| \mathbf{k} \left(\hat{\mathbf{t}} r_{ts}\right)_i - \mathbf{k}^T \left(\hat{\mathbf{t}} r_{ts}\right)_j + \tilde{\mathbf{x}} \right\| \\ &= \left\| \mathbf{c}_i - \mathbf{c}_j^T + \tilde{\mathbf{x}} \right\| \end{aligned}$$

$$\mathbf{h} = \mathbf{d} - \mathbf{r}_{tot}$$

1. Find

$$h = \min(\mathbf{h})$$

and track minimizing values

$$\hat{\mathbf{e}}_{ij}, k_i, k_j, r_i, r_j$$

2.

$$\begin{aligned} \mathbf{r}_i^{moment} &= \mathbf{x}_i^c + k_i \cdot \hat{\mathbf{t}}_i r_i^{ts} + r_i \hat{\mathbf{e}}_{ij} \\ \mathbf{r}_j^{moment} &= \mathbf{x}_j^c + k_j \cdot \hat{\mathbf{t}}_j r_j^{ts} - r_j \hat{\mathbf{e}}_{ij} \end{aligned}$$

3. Return $(\tilde{\mathbf{x}}, r_{tot}, h, \mathbf{r}_i^{moment}, \mathbf{r}_j^{moment})$

3 Linear wall

3.1 Properties

$\mathbf{p}_0$	Start point
$\mathbf{p}_1$	End point
$h_{iw} = d_{iw} - r_i$	
$l_w = \lVert \mathbf{p}_1 - \mathbf{p}_0 \rVert$	Length
$\hat{\mathbf{t}}_w = (\mathbf{p}_1 - \mathbf{p}_0)/l_w$	
$\hat{\mathbf{n}}_w = R(90°) \cdot \hat{\mathbf{t}}_w$	

3.2 Absolute distance

Solving linear system of equations determining the position of the agent $\mathbf{x}_i$ in relation to wall

$$\begin{cases} \mathbf{p}_0 + l_{n_0}\hat{\mathbf{n}}_w = \mathbf{x}_i + l_{t_0}\hat{\mathbf{t}}_w \\ \mathbf{p}_1 + l_{n_1}\hat{\mathbf{n}}_w = \mathbf{x}_i + l_{t_1}\hat{\mathbf{t}}_w \end{cases}$$

$$\begin{cases} l_{n_0}\hat{\mathbf{n}}_w - l_{t_0}\hat{\mathbf{t}}_w = \mathbf{x}_i - \mathbf{p}_0 = \mathbf{q}_0 \\ l_{n_1}\hat{\mathbf{n}}_w - l_{t_1}\hat{\mathbf{t}}_w = \mathbf{x}_i - \mathbf{p}_1 = \mathbf{q}_1 \end{cases}$$

In matrix form

$$\begin{bmatrix} l_{n_0} & l_{n_1} \\ l_{t_0} & l_{t_1} \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 \end{bmatrix}$$

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \hat{\mathbf{n}}_w & -\hat{\mathbf{t}}_w \end{bmatrix} = \begin{bmatrix} -t_1 & -t_0 \\ t_0 & -t_1 \end{bmatrix} \\ \mathbf{A}^{-1} &= \frac{1}{t_0^2 + t_1^2} \begin{bmatrix} -t_1 & t_0 \\ -t_0 & -t_1 \end{bmatrix} = \begin{bmatrix} -t_1 & t_0 \\ -t_0 & -t_1 \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{n}}_w \\ -\hat{\mathbf{t}}_w \end{bmatrix} = \mathbf{A}^T \end{aligned}$$

Conditions

$$\begin{aligned} l_n &= l_{n_0} \vee l_{n_1} = \hat{\mathbf{n}}_w \cdot \mathbf{q}_0 \vee \hat{\mathbf{n}}_w \cdot \mathbf{q}_1 \\ l_t &= l_{t_1} + l_{t_0} = -\hat{\mathbf{t}}_w \cdot \mathbf{q}_1 - \hat{\mathbf{t}}_w \cdot \mathbf{q}_0 \end{aligned}$$

Distance between agent and linear wall

$$d_{iw} = \begin{cases} \lVert \mathbf{q}_0 \rVert & l_t > l_w \\ |l_n| & \text{otherwise} \\ \lVert \mathbf{q}_1 \rVert & l_t < -l_w \end{cases}$$

Normal vector away from the wall

$$\hat{\mathbf{n}}_{iw} = \begin{cases} \hat{\mathbf{q}}_0 & l_t > l_w \\ \operatorname{sign}(l_n)\hat{\mathbf{n}}_w & \text{otherwise} \\ \hat{\mathbf{q}}_1 & l_t < -l_w \end{cases}$$

3.3 Velocity relative distance

$\tilde{\mathbf{x}} = \mathbf{x}_{iw}$	Relative position
$\tilde{\mathbf{v}} = \mathbf{v}_{iw} = \mathbf{v}_i$	Relative velocity
$\tilde{r} = r_{iw}$	Total radius
$d = \lVert \tilde{\mathbf{x}} \rVert$	Distance
$h = d - \tilde{r}$	Relative distance

$$\begin{aligned} \mathbf{q}_0 &= \mathbf{p}_0 - \mathbf{x} \\ \mathbf{q}_1 &= \mathbf{p}_1 - \mathbf{x} \\ \hat{\mathbf{n}}_{iw} &= -\operatorname{sign}(\hat{\mathbf{n}}_w \cdot \mathbf{q}_0)\hat{\mathbf{n}}_w \end{aligned}$$

$$\begin{aligned} \boldsymbol{\alpha} &= [\operatorname{angle}(\mathbf{q}_0), \operatorname{angle}(\mathbf{q}_1), \operatorname{angle}(\hat{\mathbf{n}}_{iw})] \\ \varphi &= \operatorname{angle}(\mathbf{v}) \\ \boldsymbol{\alpha}_2 &= \boldsymbol{\alpha} - \varphi \mod 2\pi \end{aligned}$$

$$i = (\arg\min(\boldsymbol{\alpha}_2), \arg\max(\boldsymbol{\alpha}_2))$$

Intersection

$$\begin{aligned} \mathbf{x} + a\mathbf{v} &= \mathbf{p}_0 + b(\mathbf{p}_1 - \mathbf{p}_0), \quad a \in \mathbb{R}^+, \quad b \in [0,1] \\ [\mathbf{v}, -\mathbf{p}] \cdot [a, b] &= \mathbf{q}_0, \quad \mathbf{p} = \mathbf{p}_1 - \mathbf{p}_0 \end{aligned}$$

4 Motion

4.1 Differential systems

Position and velocity

$$m\frac{d^2}{dt^2}\mathbf{x}(t) = \mathbf{f}(t)$$

Rotational motion

$$I\frac{d^2}{dt^2}\varphi(t) = M(t)$$

4.2 Social force model

Total force exerted on the agent is the sum of movement adjusting, social and contact forces between other agents and wall.

$$\mathbf{f}_i(t) = \mathbf{f}_i^{adj} + \sum_{j\neq i}\left(\mathbf{f}_{ij}^{soc} + \mathbf{f}_{ij}^{c}\right) + \sum_{w}\left(\mathbf{f}_{iw}^{soc} + \mathbf{f}_{iw}^{c}\right) + \boldsymbol{\xi}_i$$

4.2.1 Adjusting force

Force adjusting agent's movement towards desired in some characteristic time

$$\mathbf{f}^{adj} = \frac{m}{\tau^{adj}}(v_0 \cdot \hat{\mathbf{e}} - \mathbf{v})$$

4.2.2 Social force

Psychological force for collision avoidance. Naive velocity independent equation

$$\mathbf{f}^{soc} = A \exp\left(-\frac{h}{B}\right)\hat{\mathbf{n}}$$

Improved velocity dependent algorithm

$$\begin{aligned}
\mathbf{f}^{soc} &= -\nabla_{\tilde{\mathbf{x}}} E(\tau) \\
&= -\nabla_{\tilde{\mathbf{x}}}\left(\frac{k}{\tau^2}\exp\left(-\frac{\tau}{\tau_0}\right)\right) \\
&= -\left(\frac{k}{a\tau^2}\right)\left(\frac{2}{\tau} + \frac{1}{\tau_0}\right)\exp\left(-\frac{\tau}{\tau_0}\right)\left(\tilde{\mathbf{v}} - \frac{a\tilde{\mathbf{x}} + b\tilde{\mathbf{v}}}{d}\right),
\end{aligned}$$

where

$$\begin{aligned}
a &= \tilde{\mathbf{v}} \cdot \tilde{\mathbf{v}} \\
b &= -\tilde{\mathbf{x}} \cdot \tilde{\mathbf{v}} \\
c &= \tilde{\mathbf{x}} \cdot \tilde{\mathbf{x}} - \tilde{r}^2 \\
d &= \sqrt{b^2 - ac}, \quad b^2 - ac > 0 \\
\tau &= \frac{b - d}{a}.
\end{aligned}$$

4.2.3 Contact force

Physical contact force

$$\mathbf{f}^c = -h \cdot \left(\mu \cdot \hat{\mathbf{n}} - \kappa \cdot (\mathbf{v} \cdot \hat{\mathbf{t}})\hat{\mathbf{t}}\right), \quad h < 0$$

with damping

$$\mathbf{f}^c = -h \cdot \left(\mu \cdot \hat{\mathbf{n}} - \kappa \cdot (\mathbf{v} \cdot \hat{\mathbf{t}})\hat{\mathbf{t}}\right) + c_n \cdot (\mathbf{v} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}, \quad h < 0$$

4.2.4 Random Fluctuation

Uniformly distributed random fluctuation force

$$\boldsymbol{\xi} = f \cdot \begin{bmatrix}\cos(\varphi) & \sin(\varphi)\end{bmatrix},$$

where

$$f \in [0, f_{max}], \quad \varphi \in [0, 2\pi]$$

4.3 Rotational motion

Total torque exerted on agent, is the sum of adjusting contact and social torques

$$M_i(t) = M_i^{adj} + \sum_{j\neq i}\left(M_{ij}^{soc} + M_{ij}^{c}\right) + \sum_{w}\left(M_{iw}^{soc} + M_{iw}^{c}\right) + \eta_i(t)$$

4.3.1 Adjusting torque

Torque adjusting agent's rotational motion towards desired

$$M^{adj} = \frac{I}{\tau}\left((\varphi(t) - \varphi^0)\omega^0 - \omega(t)\right)$$

4.3.2 Social torque

Torque from social forces acting with other agent or wall

$$\mathbf{M}^{soc} = \mathbf{r}^{soc} \times \mathbf{f}^{soc}$$

4.3.3 Contact torque

Torque from contact forces acting with other agent or wall

$$\mathbf{M}^{c} = \mathbf{r}^{c} \times \mathbf{f}^{c}$$

4.3.4 Related equations

Torque calculated using cross product

$$\mathbf{M} = \mathbf{r} \times \mathbf{f} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ R_1 & R_2 & R_3 \\ f_1 & f_2 & f_3 \end{vmatrix}$$

which in two dimensions is

$$M = \begin{vmatrix} R_1 & R_2 \\ f_1 & f_2 \end{vmatrix} = R_1 \cdot f_2 - R_2 \cdot f_1$$

5 Integrators

5.1 Excplicit Euler Method

Updating using discrete time step Δt

$$\begin{aligned} t_0 &= 0 \\ t_1 &= t_0 + \Delta t \\ &\vdots \\ t_k &= t_{k-1} + \Delta t \end{aligned}$$

Acceleration on an agent

$$\begin{aligned} a_k &= \mathbf{f}_k / m \\ \mathbf{v}_{k+1} &= \mathbf{v}_k + a_k \Delta t \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{v}_{k+1} \Delta t \end{aligned}$$

Angular acceleration

$$\begin{aligned} \alpha_k &= M_k / I \\ \omega_{k+1} &= \omega_k + \alpha_k \Delta t \\ \varphi_{k+1} &= \varphi_k + \omega_{k+1} \Delta t \end{aligned}$$

5.2 Velocity verlet

Velocity verlet algorithm

$$\begin{aligned} \mathbf{v}_{k+\frac{1}{2}} &= \mathbf{v}_k + \frac{1}{2} a_k \Delta t \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{v}_{k+\frac{1}{2}} \Delta t \\ \mathbf{v}_{k+1} &= \mathbf{v}_{k+\frac{1}{2}} + \frac{1}{2} a_{k+1} \Delta t \end{aligned}$$

or more simply

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{v}_k \Delta t + \frac{1}{2} a_k \Delta t^2 \\ \mathbf{v}_{k+1} &= \mathbf{v}_k + \frac{1}{2} \left(a_k + a_{k+1} \right) \Delta t \end{aligned}$$

6 Navigation

6.1 Theory

Navigation algorithm is a function that takes at least coordinate $\mathbf{x}$ as an argument and returns an unit vector $\hat{\mathbf{e}}$ that is used as target direction for the agent

$$f(\mathbf{x}, \ldots) \to \hat{\mathbf{e}}$$

6.2 Manual construction

6.3 Fluid flow

One way to find suitable function is to solve how *incompressible*, *irrotational* and *inviscid* fluid (ideal fluid) would flow out of the constructed space.

7 Spatial game

Spatial game for egress congestion.

7.1 Game matrix

$T_i = \lambda_i / \beta$	Estimated evacuation time
λ_i	Number of other agents closer to the exit
β	Capacity of the exit
$T_{ij} = \left(T_i + T_j \right) / 2$	Average evacuation time
T_{ASET}	Available safe egress time
$T_0 (= T_{ASET})$	Time difference between T_{ASET} and T_i before agents start playing the game
$C > 0$	Cost of conflict

Number of other agents closer to the exit can be solved

$$\lambda = \operatorname{argsort} \|\mathbf{p}_0 - \mathbf{x}\|$$

where $\mathbf{p}_0$ is the center of the exit

Cost function

$$u(T_i, T_{ASET}), \quad u'(T_i) \geq 0, \quad u''(T_i) \geq 0$$

Increase/decrease in cost

$$\Delta u(T_{ij}) = u(T_{ij}) - u(T_{ij} - \Delta T) \approxeq u'(T_{ij}) \Delta T$$

	Impatient	Patient
Impatient	C, C	$-\Delta u(T_{ij}), \Delta u(T_{ij})$
Patient	$\Delta u(T_{ij}), -\Delta u(T_{ij})$	$0, 0$

	Impatient	Patient
Impatient	$\frac{C}{\Delta u(T_{ij})}, \frac{C}{\Delta u(T_{ij})}$	$-1, 1$
Patient	$1, -1$	$0, 0$

i) Prisoner's dilemma (PD)

$$0 < \frac{C}{\Delta u(T_{ij})} \leq 1$$

ii) Hawk-dove (HD)

$$\frac{C}{\Delta u(T_{ij})} > 1$$

Assumptions

1. Game is not played

$$T_{ij} \leq T_{ASET} - T_0$$

2. Cost function starts to increase quadratically

$$T_{ij} > T_{ASET} - T_0$$

3. Game turns into prisoner's dilemma

$$u'(T_{ASET}) = C, \quad T_{ij} \geq T_{ASET}$$

Cost function that meets the assumptions

$$u(T_{ij}) = \begin{cases} 0 & T_{ij} \leq T_{ASET} - T_0 \\ \frac{C}{2T_0} \left(T_{ij} - T_{ASET} + T_0\right)^2 & T_{ij} > T_{ASET} - T_0 \end{cases}$$

Derivative

$$u'(T_{ij}) = \begin{cases} 0 & T_{ij} \leq T_{ASET} - T_0 \\ \frac{C}{T_0} \left(T_{ij} - T_{ASET} + T_0\right) & T_{ij} > T_{ASET} - T_0 \end{cases}$$

Loss/gain of overtaking

$$\Delta u(T_{ij}) \approxeq u'(T_{ij}) \Delta T = \frac{C}{T_0} \left(T_i - T_{ASET} + T_0\right) \Delta T$$

Value parameter of the game matrix

$$\frac{C}{\Delta u(T_{ij})} \approxeq \frac{T_0}{T_{ij} - T_{ASET} + T_0}$$

7.2 Settings and best-response dynamics

	Unit	Value	
		4	von Neumann neighborhood
		8	Moore neighborhood
r_n	m	0.40	Distance to agent that is considered as neighbor
v_i			Loss defined by game matrix
S			Set of strategies $\{\text{Patient}, \text{Impatient}\}$
s			Strategy $\in \{\text{Patient}, \text{Impatient}\}$

The best-response strategy

$$s_i^{(t)} = \arg\min_{s_i' \in S} \sum_{j \in N_i} v_i \left(s_i', s_j^{(t-1)}; T_{ij}\right)$$

$s_j^{(t-1)}$ strategy neighbor played on period $t-1$

Updating strategy using poisson process.