揚州大學
YANGZHOU UNIVERSITY

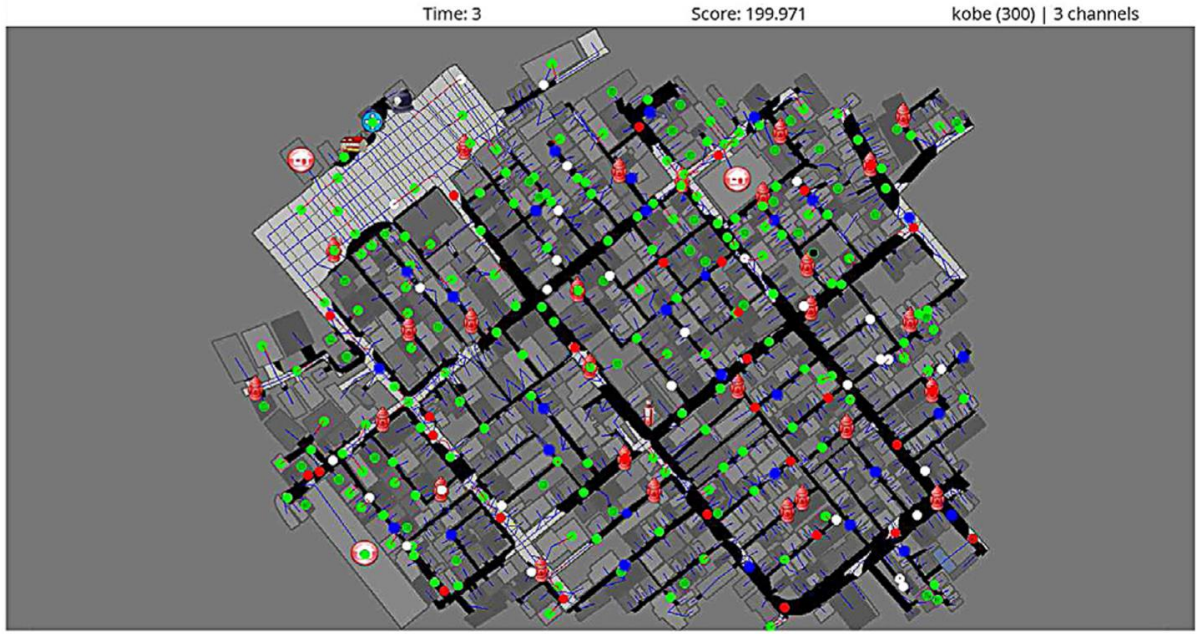
Game theory and Personality Traits in Task Scheduling Problem of Swarm Robot Systems

Jieke SHI
School of Information Engineering, Yangzhou University
10 Jan 2020

Outline

- **Preliminaries**
- Modeling Using Game Theory
- Personality Traits
- Convergence Proof
- Simulation Results
- Future Work

Preliminaries



- Continuous damage of tasks
- Centralized scheduling strategy
- Real-time decentralized scheduling strategy

Centralized scheduling strategy

A dynamic auction approach for differentiated tasks under cost rigidities (DAACR)

$$\min \sum_{i\in B}\sum_{j\in T_2} q_{ij} cost_i^j$$

$$\text{s.t.} \sum_{j|(i,j)\in D} q_{ij} = 1, \forall i = 1, 2, ..., n$$

$$\sum_{i|(i,j)\in D} q_{ij} = 1, \forall j = 1, 2, ..., n$$

$$q_{ij} = 0, 1, \forall (i, j) \in D$$

$$|p_j - cost_i^j - \max_{k\in D(i)} \{p_k - cost_i^k\}| \le \epsilon$$

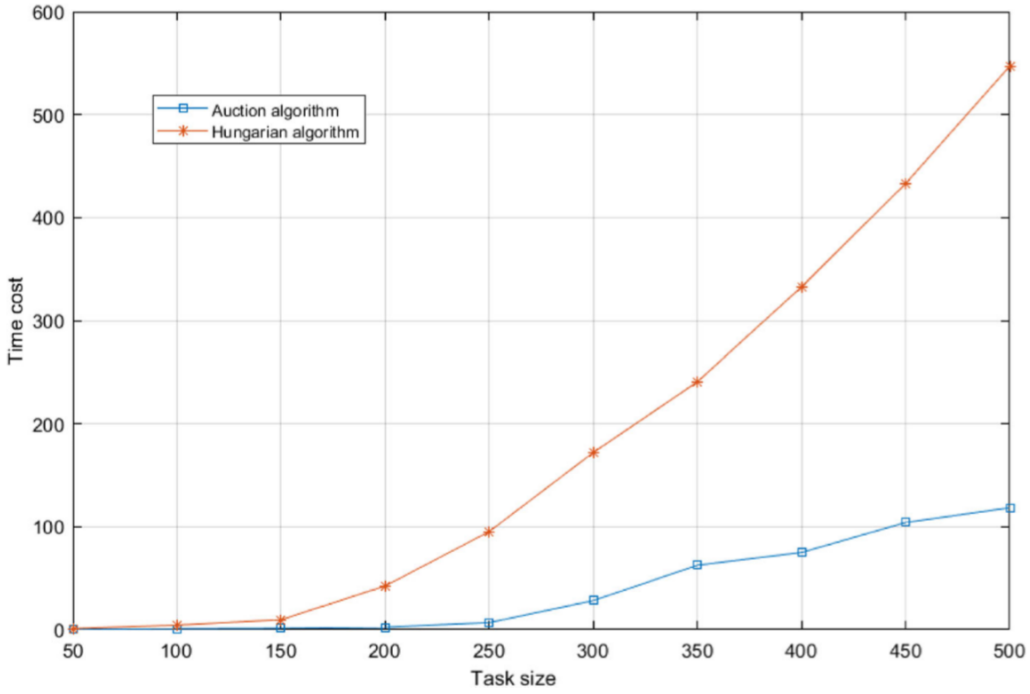


Fig. 1 Time cost comparison between Auction algorithm and Hungarian algorithm

Centralized scheduling strategy

Step 3: bidding phase All robots bid for j_i which is the most gainful task, the bidding price of the robot is determined as:

$$a_{ij_i} = p_{j_i} - u_i + v_i - \epsilon = cost_i^{j_i} + v_i - \epsilon \tag{15}$$

Step 4: allocation phase For each task j, if the $B(j)$ is not empty, we update the price of the task j to the highest bid price:

$$p_j = \max_{i \in B(j)} a_{ij} \tag{16}$$

The task is assigned to the highest bidding robot i_j, and at the same time, the two-tuples related to the robot i_j and the task j are removed from the A which is an infeasible allocation, and the new two-tuples (i_j, j) are added to the set A.

Centralized scheduling strategy

We define that the optimal solution will minimize the total damage suffered by the system during the whole progress. This can be represented as an integer programming, the formula is that:

$$\min \quad \sum_{t=0}^{t_d} \sum_{j=1}^{M} H(w_j^t)$$

$$\text{s.t.} \quad \sum_{j=1}^{M} C_j^t \le N$$

$$\sum_{j=1}^{M} \sum_{i=1}^{C_j^t} c_i \le \sum_{j=1}^{M} w_j^t \tag{4}$$

$$w_j^{t+1} = w_j^t - \sum_{i=1}^{C_j^t} c_i$$

$c_i$is a no-negative constant, $\forall i$,

C_j^t is a no-negative integer, $\forall j, \forall t$

According to Definition 1, the local optimal solution can be determined by computing the following integer programming.

$$\min \quad \sum_{j=1}^{M} H(w_j^t)$$

$$\text{s.t.} \quad \sum_{j=1}^{M} C_j^t \le N \tag{6}$$

$$\sum_{j=1}^{M} \sum_{i=1}^{C_j^t} c_i \le \sum_{j=1}^{M} w_j^t$$

C_j^t is a no-negative integer, $\forall j, \forall t$

Centralized scheduling strategy

The optimal scheduling for every time unit t, that is, the minimization of the damage $H(w_j^t)$, can eventually lead to the global optimal solution, when $\frac{\partial}{\partial t}H(w_j^t) \geq 0$.

The scheduling scheme obtained by the greedy strategy is $C[0, t_d] = \{C^0, C^1, \ldots, C^{t_d}\}$. Assume there exists a better solution $C_*[0, t_d] = \{C_*^0, C_*^1, \ldots, C_*^{t_d}\}$, which differs from the greedy solution. This means that at some time t_0 the better solution must assign agents differently than the greedy solution. The scheduling scheme of our greedy algorithm at time t_0 is $C^{t_0} = (C_1^{t_0}, C_2^{t_0}, \ldots, C_M^{t_0})$, the better scheduling is $C_*^t = (C_{1*}^{t_0}, C_{2*}^{t_0}, \ldots, C_{M*}^{t_0})$. By definition of the greedy algorithm, we know $\sum_{j=1}^{M} H(w_j^{t_0}) \leq \sum_{j=1}^{M} H_*(w_j^{t_0})$. Thus we prove by induction that $\sum_{t=0}^{t_d}\sum_{j=1}^{M} H(w_j^{t_0}) \leq \sum_{t=0}^{t_d}\sum_{j=1}^{M} H(w_j^t)$.

If $\sum_{j=1}^{M} H(w_j^{t_0}) < \sum_{j=1}^{M} H_*(w_j^{t_0})$, then there must exist $\sum_{t=0}^{t_d}\sum_{j=1}^{M} H(w_j^t) < \sum_{t=0}^{t_d}\sum_{j=1}^{M} H_*(w_j^t)$. If $\sum_{j=1}^{M} H(w_j^{t_0}) = \sum_{j=1}^{M} H_*(w_j^{t_0})$, there also exists $\sum_{t=0}^{t_d}\sum_{j=1}^{M} H(w_j^t) < \sum_{t=0}^{t_d}\sum_{j=1}^{M} H_*(w_j^t)$.

So, according to the greedy strategy, we can conclude $\sum_{j=1}^{M} H(w_j^{t_0+1}) \leq \sum_{j=1}^{M} H_*(w_j^{t_0+1})$ at the next unit time $t_0 + 1$.

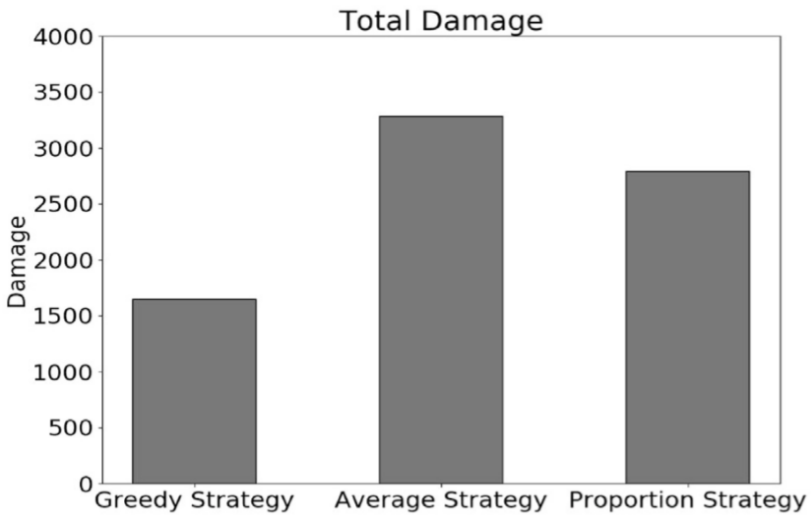


Fig. 1. Shows the total damage comparison among three methods.

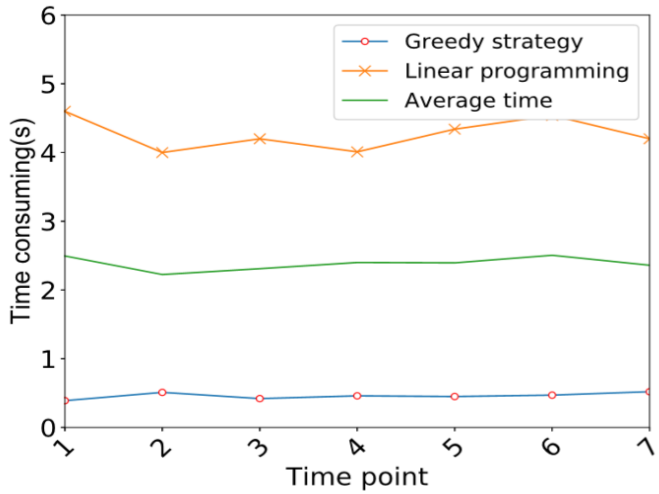


Fig. 2. Shows the time comparison among two methods.

Real-time decentralized scheduling strategy

- $Ag = \{a_1, a_2, \ldots, a_n\}$ is the set of agents who participate in a negotiation. The negotiation host is not included in Ag.
- $Ac = (Ac_1, Ac_2, \ldots, Ac_n)$ is a tuple in which Ac_i represents the action set of negotiation agent $i \in Ag$.
- W is a non-empty set of states. $w_0 \in W$ is the initial state of a negotiation. $T \subset W$ is the set of terminal states $w \in W$.
- $L = (L_1, L_2, \ldots, L_n)$ is a tuple where each $L_i : W \backslash T \rightarrow 2^{A_i}$ is the legal function for a_i. It determines the actions a_i can take at some state.
- A so-called state update function is defined as $u : W \times Ac_i \rightarrow W$ to represent at what condition, a state will change to another.
- Agr is the set of all possible negotiation agreements. The function $Q : T \rightarrow Agr$ maps each terminal state to an agreement.

Algorithm 1: A real-time decentralized scheduling algorithm

```
1  repeat
2      U(1) = U(−1) = ∅;
3      for each i ∈ Ag do
4          propose(u_i(−1), u_i(1));
5          U(1) = U(1) ∪ u_i(1);
6          U(−1) = U(−1) ∪ u_i(−1);
7          repeat
8              i = arg min U(−1);
9              delete i from U(1) and U(−1);
10             j = arg max U(1);
11             if u_i(−1) < u_j(1) then
12                 agent i gives a task to agent j;
13                 delete j from U(1) and U(−1);
14             else
15                 Algorithm Stops
16         until true
17 until true
```

Real-time decentralized scheduling strategy

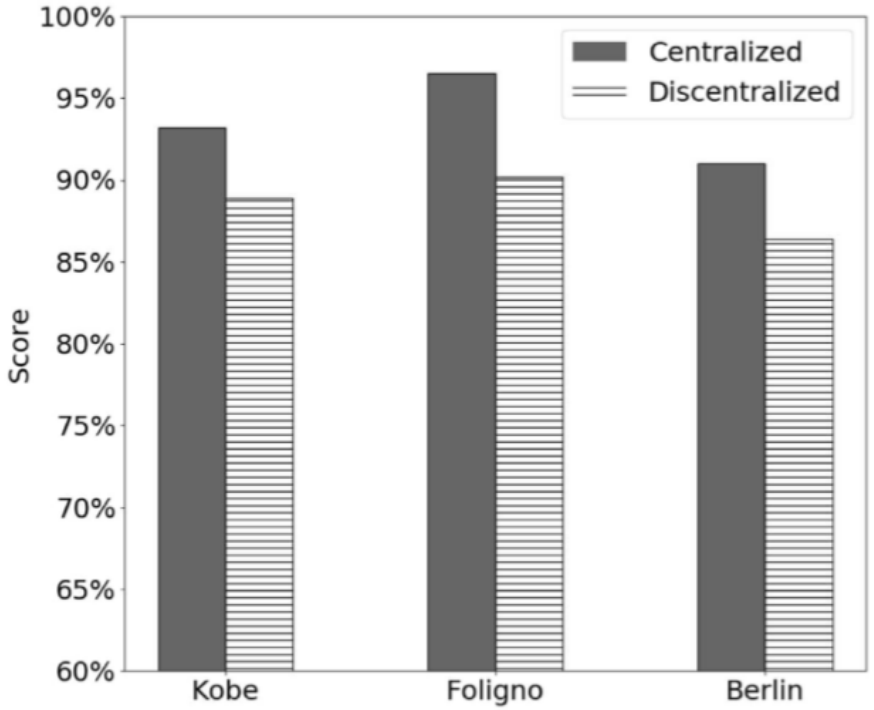


Fig. 4. Comparing the algorithms across three maps under unrestricted communication.

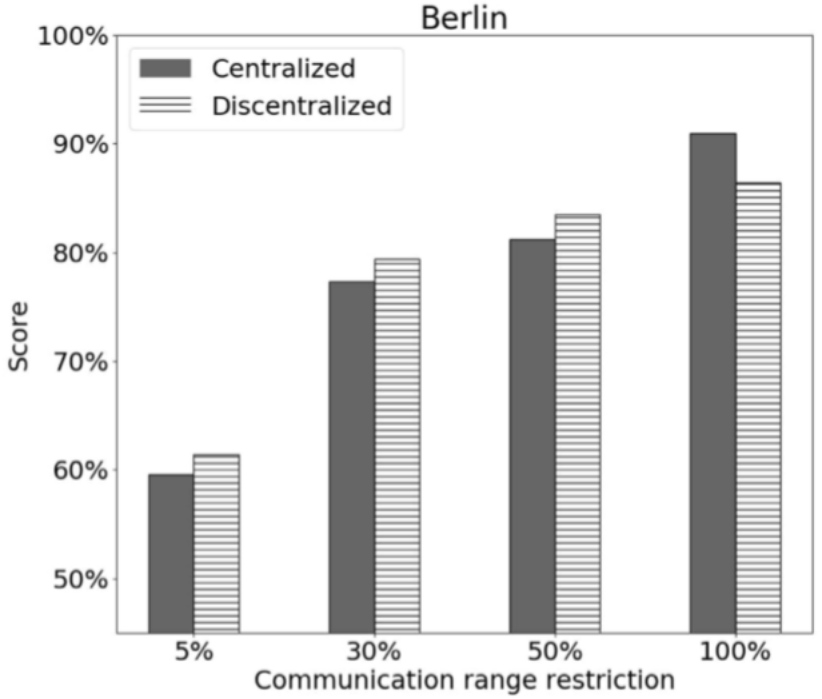


Fig. 7. Comparing the algorithms across Berlin map under restricted communication.

Outline

- Preliminaries
- **Modeling Using Game Theory**
- Personality Traits
- Convergence Proof
- Simulation Results
- Future Work

Modeling Using Game Theory

	Heads	Tails
Heads	$1, -1$	$-1, 1$
Tails	$-1, 1$	$1, -1$

One of the major successes in the field of economics and social sciences in the past decades has been the application of Game Theory to the modeling of social interactions of rational entities for the prediction of outcomes of conflicts among them.

Modeling Using Game Theory

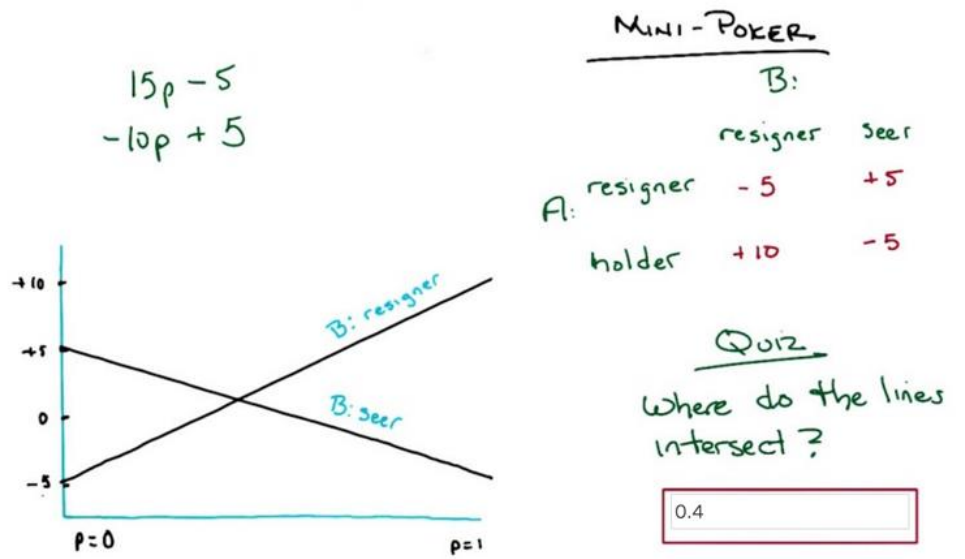
		Column Player	
		C	D
Row Player	C	3,3	-1,5
	D	5,-1	1,1

Prisoner's dilemma game is a typical model of such interesting scenario. In a prisoner's dilemma game, each of the players has a set A = {C,D} of two available actions, where C denotes Cooperation and D denotes Defection.

For both players, defection D is the only dominant strategy for any rational agent. Thus, mutual defection (D,D) is theunique Nash equilibrium. However, mutual cooperation (C,C) maximizes social welfare, the sum of utilities of both players. Therefore, out of the two possible actions C and D in a PD game, choosing cooperation C is socially beneficial, yet there is always a lure for any agent to exploit its opponent's cooperation C via playing defection D.

Modeling Using Game Theory

In the case of mixed strategies, the payoff functions are the expectations of the players. The formulation of the problem is a linear system of inequalities and characterizes a linear programming problem. More specifically, the problem becomes linear forms in the probabilities with which the various players play their pure strategies.



Outline

- Preliminaries
- Modeling Using Game Theory
- **Personality Traits**
- Convergence Proof
- Simulation Results
- Future Work

Personality Traits

Fig. 1. Examples for natural swarms (a) flock of birds; (b) school of fishes; and (c) colony of honey bees.

One of the suggestions of the Theory of Evolution is that animals have emotions. Moreover, these emotions are shared by the same species. Also, traits of personality (term that is used interchangeably with emotions) are important to the maintenance of objectives and collaboration

Personality Traits

Definition 1. *For an robot i, we define its personal traits as*

$$\alpha_i = [\alpha_1, \alpha_2, ..., \alpha_n]^T \tag{1}$$

and for each one of the personality traits, the reward functions are represented by the vector

$$\beta_i = [\beta_1, \beta_2, ..., \beta_n]^T \tag{2}$$

The reward functions can be arbitrary functions that represent the extent to which personality traits affect the action of the robot. When the action chosen by the agent is γ, its utility function can be formulated as:

$$U(s, \alpha, \gamma) = f(\sum_{i=1}^{n} \alpha_i \beta_i(s_t, \gamma, t)) \tag{3}$$

In addition, we need to consider the evolution process of some personality traits, the dynamics of the personality trait vector can be described by the following general difference equation,

$$\alpha_{t+1} = \alpha_t + \eta \mathcal{H}(\alpha_t, \beta_t) \tag{4}$$

where the function $\mathcal{H}(\alpha_t, \beta_t)$ depends on the application under consideration.

Outline

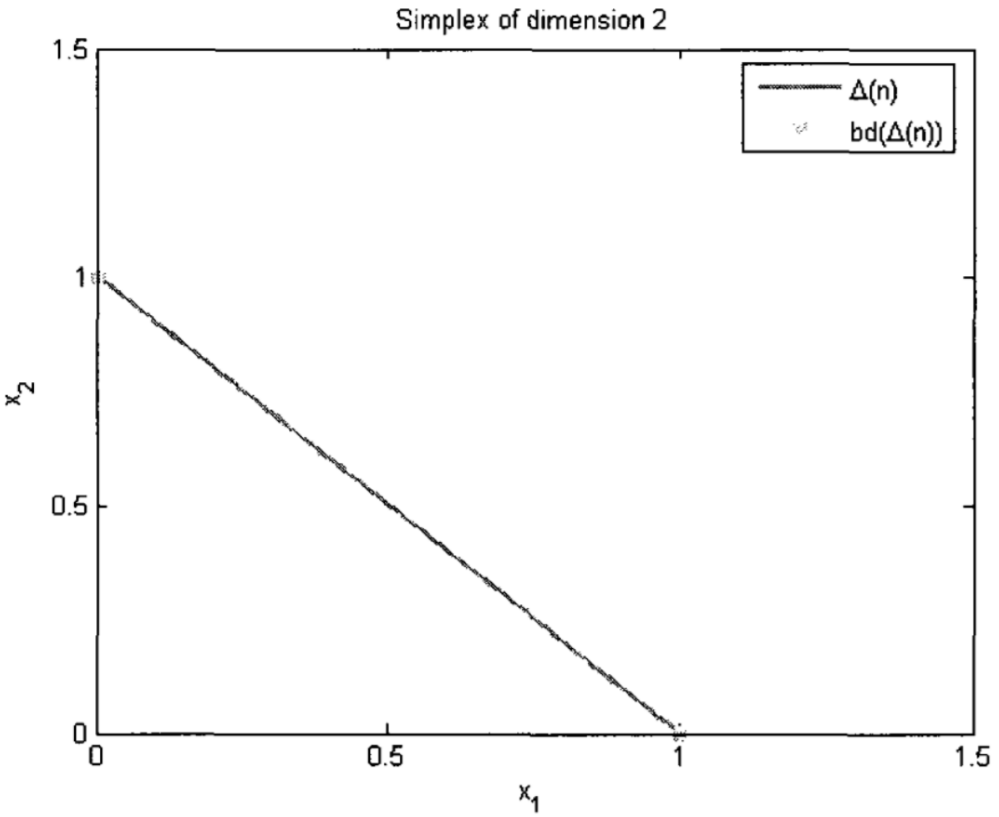
- Preliminaries
- Modeling Using Game Theory
- Personality Traits
- **Convergence Proof**
- Simulation Results
- Future Work

Definition 2. *Define* $x = [x_1, x_2, ..., x_n]$, n *denotes the numbers of strategies.*

$\Delta(n)$ *denotes the simplex in* $\mathcal{R}^n$*,i.e.,*$\{x \in \mathcal{R}_n | x_i \geq 0,\ \forall i = 1, ..., n;\ and\ \sum_{i=1}^{n} x_i = 1\}$.

$Int(\Delta(n))$ *denotes the set of interior points of a simplex, i.e.,*$\{x \in \Delta(n) | x_i > 0,\ \forall i = 1, ..., n;\}$.

$v_i \in \Delta(n)$ *denotes the* i_{th} *vertex of the simplex* $\Delta(n)$*, i.e., the vector whose* i_{th} *term equals* 1 *and remaining terms equal* 0, $\{x \in \Delta(n) | x_i = 1\ and\ x_j = 0, \forall j \neq i\}$.



For one zero-sum game, the utility function of player P_1 is

$$U_1(p_1, p_2) = p_1^T M p_2 \tag{5}$$

For player P_2,

$$U_2(p_2, p_1) = -p_2^T M^T p_1 \tag{6}$$

Define the best response mapping for player P_1 as

$$\varepsilon_1(p_2) = \arg \max_{x \in \Delta(n)} U_1(p_1, p_2) \tag{7}$$

For player P_2,

$$\varepsilon_2(p_1) = \arg \max_{x \in \Delta(m)} U_2(p_2, p_1) \tag{8}$$

Define the empirical frequency, $q_i(k)$. The empirical frequency of player P_i can be calculated as the running average of the actions of player P_i, that is

$$q_1(k) = q_1(k-1) + \frac{1}{k}(v_{a_1(k-1)} - q_1(k-1)) \tag{9}$$

And,

$$q_2(k) = q_2(k-1) + \frac{1}{k}(v_{a_2(k-1)} - q_2(k-1)) \tag{10}$$

Mapping personality traits to the strategy space by transcation matrix A,

$$q_1 = A_1 \alpha_1 \tag{11}$$

$$q_2 = A_2 \alpha_2 \tag{12}$$

The functions used to update the personality traits are

$$\mathcal{H}_1(\alpha_1, \beta_1) = \beta_1 - \alpha_1 \tag{15}$$
$$\mathcal{H}_2(\alpha_2, \beta_2) = \beta_2 - \alpha_2 \tag{16}$$

Then the personality dynamics are

$$\widetilde{\alpha}_1 = A_1^T (A_1 A_1^T)^{-1} \varepsilon_1(p_2) - \alpha_1 \tag{17}$$
$$\widetilde{\alpha}_2 = A_2^T (A_2 A_2^T)^{-1} \varepsilon_2(p_1) - \alpha_2 \tag{18}$$

Using these definitions, we can find the strategy dynamics for player P_1 is

$$\begin{aligned} \widetilde{q}_1(t) &= A_1 \widetilde{\alpha}_1(t) \\ &= A_1 (A_1^T (A_1 A_1^T)^{-1} \varepsilon_1(p_2) - \alpha_1) \\ &= \varepsilon_1(q_2(t)) - q_1(t) \end{aligned} \tag{19}$$

In the same way, the strategy dynamics for player P_2 is

$$\widetilde{q}_2(t) = \varepsilon_2(q_1(t)) - q_2(t) \tag{20}$$

We can know the best strategy is that

$$\max_{x\in\Delta(n)} U_1(x, q_2) = (\varepsilon_1(q_2))^T M q_2 \tag{22}$$

So, we can get

$$\mathcal{U}_1(q_1, q_2) = (\varepsilon_1(q_2) - q_1)^T M q_2 \tag{23}$$

In the same way,

$$\mathcal{U}_2(q_2, q_1) = -(\varepsilon_2(q_1) - q_2)^T M^T q_1 \tag{24}$$

And note that by definition $\mathcal{U}_1(q_1, q_2) \geq 0$ and $\mathcal{U}_1(q_2, q_1) \geq 0$ with equality if and only if $\varepsilon_2(q_1) = q_2$ and $\varepsilon_1(q_2) = q_1$.

Lemma 1. *Define* $\widetilde{\mathcal{U}}_i(t) = \mathcal{U}_i(q_i(t), q_{-i}(t))$, *then* $\dot{\widetilde{\mathcal{U}}}_1(t) = -\widetilde{\mathcal{U}}_1(t) + \dot{q}_1^T M \dot{q}_2$, $\dot{\widetilde{\mathcal{U}}}_2(t) = -\widetilde{\mathcal{U}}_2(t) + \dot{q}_2^T M^T \dot{q}_1$.

Theorem 1. *The solutions of equ.19 and equ.20 satisfy* $\lim_{t\to\infty}(q_1(t) - \varepsilon_1(q_2(t))) = 0$ *and* $\lim_{t\to\infty}(q_2(t) - \varepsilon_2(q_1(t))) = 0$.

Proof 2. *By Lemma 1, we can get* $\dot{\widetilde{\mathcal{U}}}_1(t) \leq -\widetilde{\mathcal{U}}_1(t) + \dot{q}_1^T M \dot{q}_2$, $\dot{\widetilde{\mathcal{U}}}_2(t) \leq -\widetilde{\mathcal{U}}_2(t) + \dot{q}_2^T M^T \dot{q}_1$. *Define function* $\mathcal{U}_{12}(t) = \widetilde{\mathcal{U}}_1(t) + \widetilde{\mathcal{U}}_2(t)$. *Taking its derivative, we can get* $\dot{\mathcal{U}}_{12}(t) = \dot{\widetilde{\mathcal{U}}}_1(t) + \dot{\widetilde{\mathcal{U}}}_2(t) = -\widetilde{\mathcal{U}}_1(t) - \widetilde{\mathcal{U}}_2(t)$. *Since* $\widetilde{\mathcal{U}}_2 \geq 0$ *and* $\widetilde{\mathcal{U}}_2 \geq 0$ *with equality only at the equilibrium point of equ.19 and equ.20, which means* $\dot{\mathcal{U}}_{12}(t) \leq 0$ *is with equality only if and only if at the equilibrium point of equ.19 and equ.20,* $\mathcal{U}_{12}$ *is a Lyapunov function and this theorem follows the from standard Lyapunov arguments.*

Outline

- Preliminaries
- Modeling Using Game Theory
- Personality Traits
- Convergence Proof
- **Simulation Results**
- Future Work

Algorithm 1 Zero-sum game.

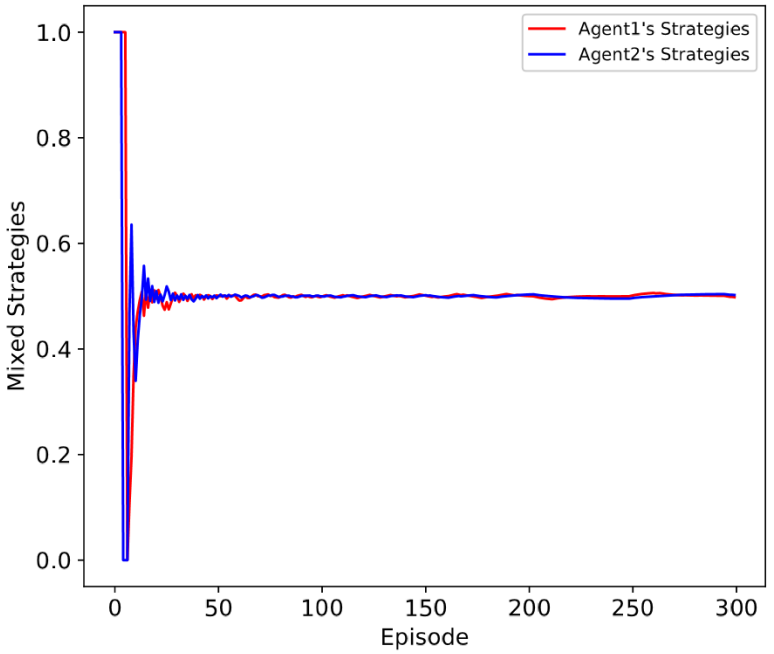
Input: $n = 0.1$, $t = 0.01$

Output: Mixed strategy

1: Set the number of personality traits for player A and player B;
2: Initialize the personality traits α
3: Define the payoffs and mappings from the personality traits spaces to the strategies spaces of the game.
4: Players initialize the empirical frequency as 0.
5: **for** $i = 1$ to 300 do **do**
6: Players calculates the strategy according to its probability distribution.
7: Both players play their actions and record the payoffs.
8: Update personality traits and empirical frequency of players.
9: **end for**

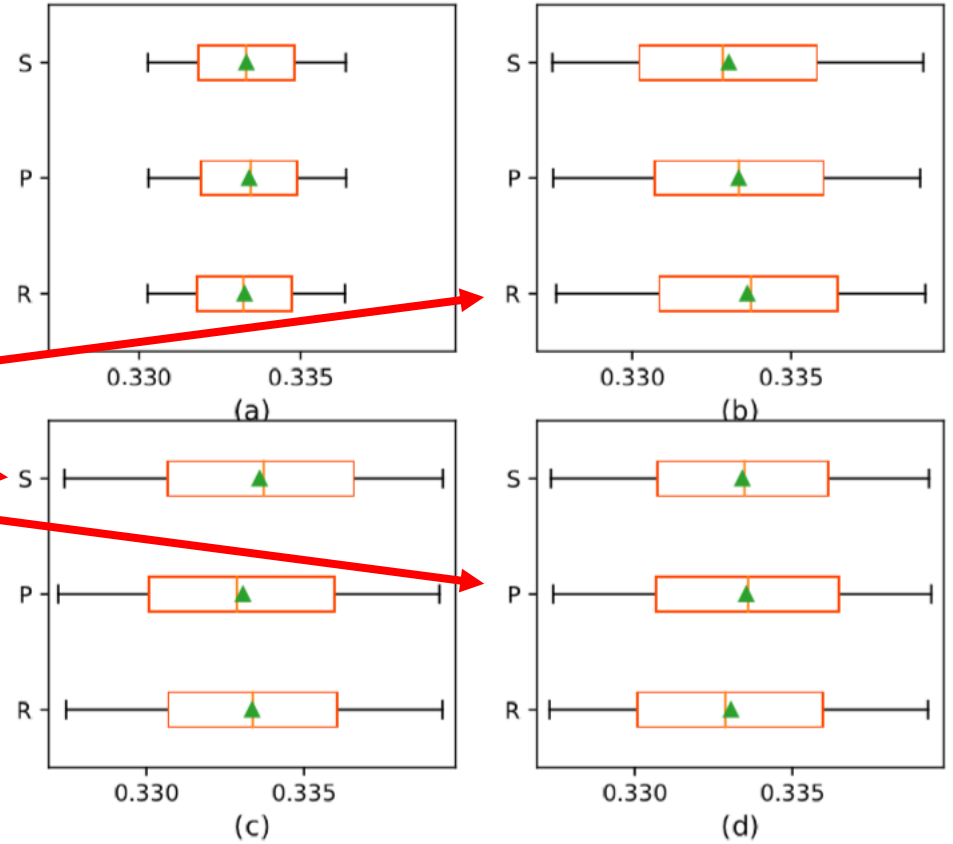
EXAMPLE OF A ZERO-SUM GAME

	head	**tail**
head	(-1, 1)	(1, -1)
tail	(1, -1)	(-1, 1)



EXAMPLE OF A ROCK-PAPER-SCISSORS GAME

	Rock	Paper	Scissors
Rock	(0, 0)	(-1, 1)	(1, -1)
Paper	(1,-1)	(0, 0)	(-1, 1)
Scissors	(-1, 1)	(1, -1)	(0, 0)

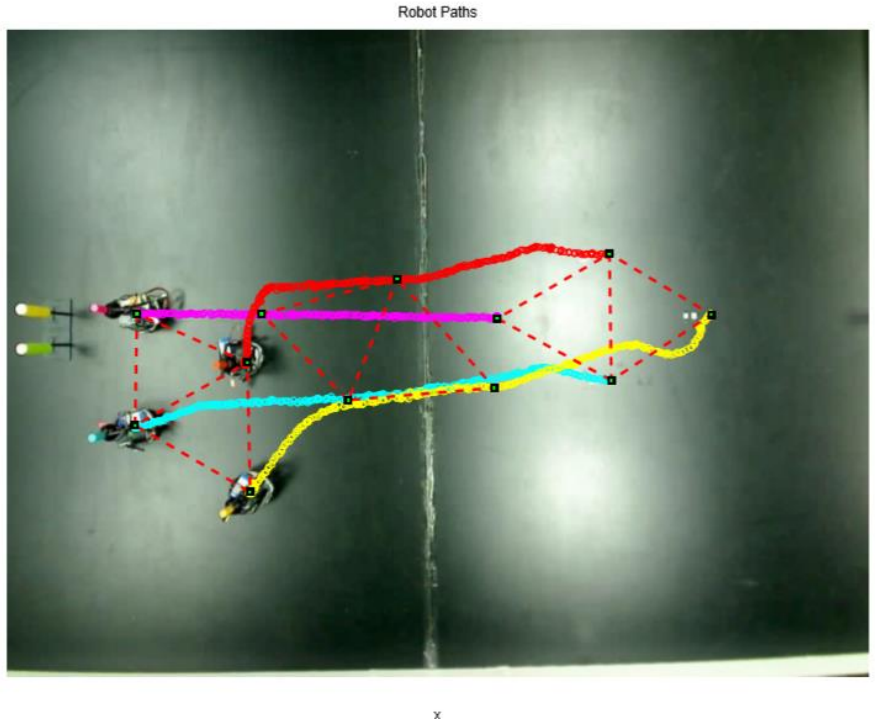


Outline

- Preliminaries
- Modeling Using Game Theory
- Personality Traits
- Convergence Proof
- Simulation Results
- **Future Work**

Humans have norms,
but do agents / robots have them ?

A straightforward example of this is when two drivers arrive at an intersection simultaneously from neighboring streets. While each player has the incentive of not yielding, myopic decisions by both can lead to undesirable accidents. Both drivers yielding, however, also creates inefficiency. Ideally, we would like norms like “yield to the driver on right”, which serves all drivers in the long run.



Humans have norms, but do agents / robots have them ?

Behavioral norms are key ingredients that allow agent coordination where societal laws do not sufficiently constrain agent behaviors. Whereas social laws need to be enforced in a top-down manner, norms evolve in a bottom-up manner and are typically more self-enforcing. While effective norms can significantly enhance performance of individual agents and agent societies, there has been little work in multiagent systems on the formation of social norms.

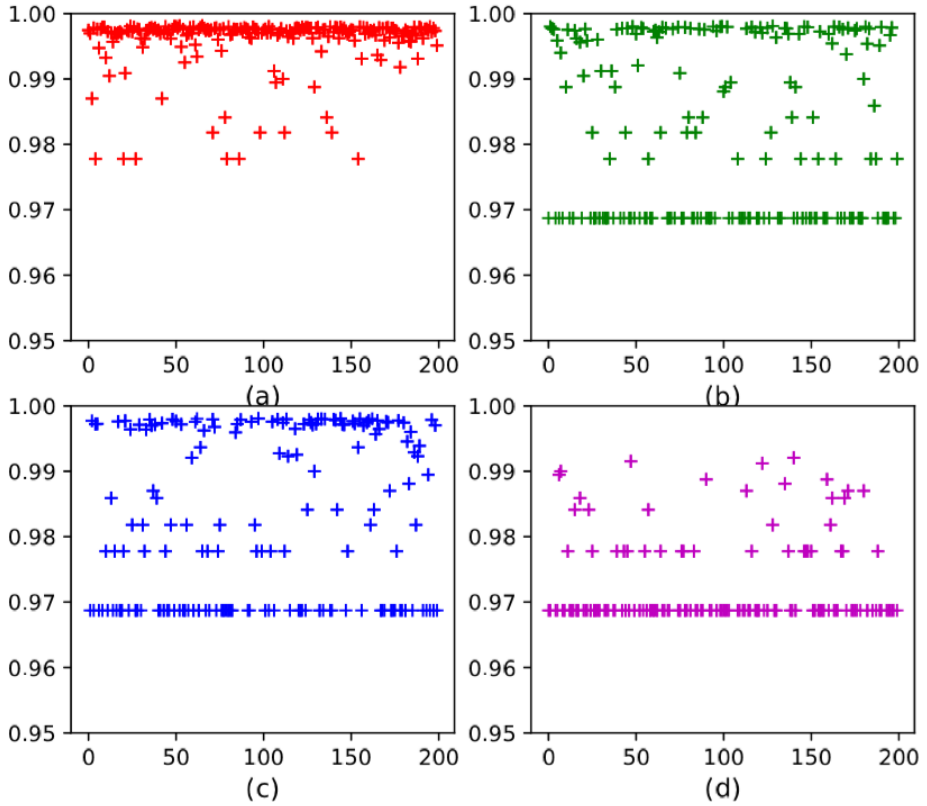
for *a fixed number of epoch* **do**
  repeat
    remove randomly agents p_{row} and p_{col} from the population ask each agent to select an action;
    send the joint action to p_{row} and p_{col} for policy update;
  until *all agents have been selected during the epoch* ;

	G	Y_L
G	-1, -1	3, 2
Y_R	2, 3	1, 1

(a) social dilemma game

	0	1
0	4, 4	-1, -1
1	-1, -1	4, 4

(b) coordination game



揚州大學
YANGZHOU UNIVERSITY

Game theory and Personality Traits in Task Scheduling Problem of Swarm Robot Systems

Jieke SHI
School of Information Engineering, Yangzhou University
10 Jan 2020