Game Theory and Reinforcement Learning to develop Organic Distributed Systems

Tobias Ziermann, Stefan Wildermann, Prof. Dr. Jürgen Teich

Hardware-Software-Co-Design

Universität Erlangen-Nürnberg

`tobias.ziermann@informatik.uni-erlangen.de`

22.02.10

Agenda

- Game Theory
- Learning in games
- Medium access example
- Conclusion

Game-Theoretic Analysis

- Game Theory can be used as formal tool to define and analyze multi-agent decision problems
- A game consists of
  - a set of players N = {1,..., n}
  - a set of actions A available to each player i
  - a specification of payoffs u(a) for each combination of actions
- The normal form representation for two player:

		Player 2	
		A	B
Player 1	A	2,5	-5,0
	B	0,-1	5,0

Different Assumptions

- Cooperative vs. Non-Cooperative
  - Coordination Game:
    Battle of the Sexes

male

female

	Home	Party
Home	10,5	0,0
Party	0,0	5,10

- Incomplete Information: Bayesian Games
  - Player 1 only knows his own reward:

male

female

	Home	Party
Home	10	0
Party	0	5

Extending the Model

- Strategies represent the actions a player chooses:
  - Pure Strategies: $s_i = a \in A$
  - Mixed Strategies: $s_i = \mathbf{p}, where \left|\mathbf{p}\right| = \left|A\right|$
- Static vs. Repeated
- Example:

$$\mathbf{p}_i = (0{,}5 \quad 0{,}5)$$

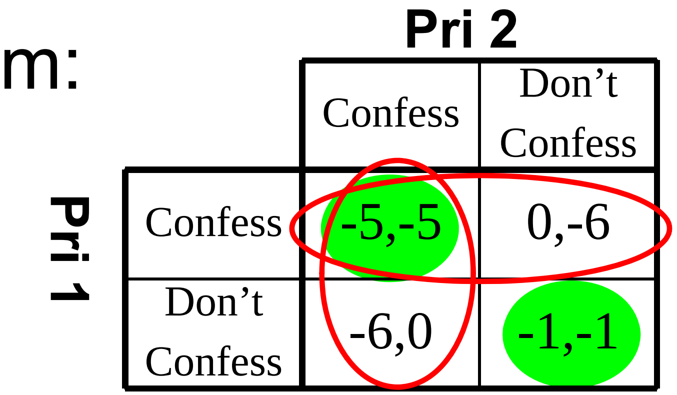
$$u(\mathbf{p}_1, \mathbf{p}_2) = \frac{15}{4} = 3{,}75$$

		male	
		Home	Party
female	Home	10,5	0,0
	Party	0,0	5,10

Equilibria

- Equilibria are important characteristics to evaluate stable states in games

- Dominant Strategy Equilibrium:
  - Prisoner's Dilemma



		Pri 2	
		Confess	Don’t Confess
Pri 1	Confess	-5,-5	0,-6
	Don’t Confess	-6,0	-1,-1

- Nash Equilibrium:
  - No player achieves a higher payoff when choosing another strategy than his current strategy, given the other players keep their current strategies

  - At least one Nash Equilibrium with mixed Strategies

Learning the Strategy

- Learning in Games for Static or repeated Tasks [1]:
  - Fictitious Play:
    - Adapt strategy assuming others players have fixed strategy
  - Replicator Dynamics:
    - Adapt strategy assuming other players strategy grows with the performance of the strategy

- Learning with multi-agent reinforcement learning for dynamic tasks [2]:
  - Agents receive reward for their action and adapt strategy depending on the reward

Goals of Learning

- Adapt to dynamic environment
  - Number of players change
  - Environment is not known in advance

- Convergence to stationary strategies
  - Maximize common return
  - Maximize individual return

Example

- Modeling access to Controller Area Network (CAN)
- Static: Players have the strategies {send, wait}
- The payoff is
  - 1, if granted access to the medium
  - 0, else
- Goal:
  - Achieve fair Bandwidth distribution

		Player 2	
		wait	send
Player 1	wait	0,0	0,1
	send	1,0	0,1

Priority-based Medium Access with Player 2 having higher priority

Example (2)

- Formal Utility:

$$u_1(\mathbf{p}) = u_2(\mathbf{p}) = \ldots = u_n(\mathbf{p})$$

- Trick:
  - Force a small amount of bandwidth ε to be free

$$u_1(\mathbf{p}) = u_2(\mathbf{p}) = \ldots = u_n(\mathbf{p}) = \frac{1-\varepsilon}{n}$$

$$u_i(p) = \begin{cases} p_i \cdot \prod_{j>i} (1 - p_j), & if \prod_{j>i} (1 - p_j) \geq \varepsilon \\ 0, & else \end{cases}$$

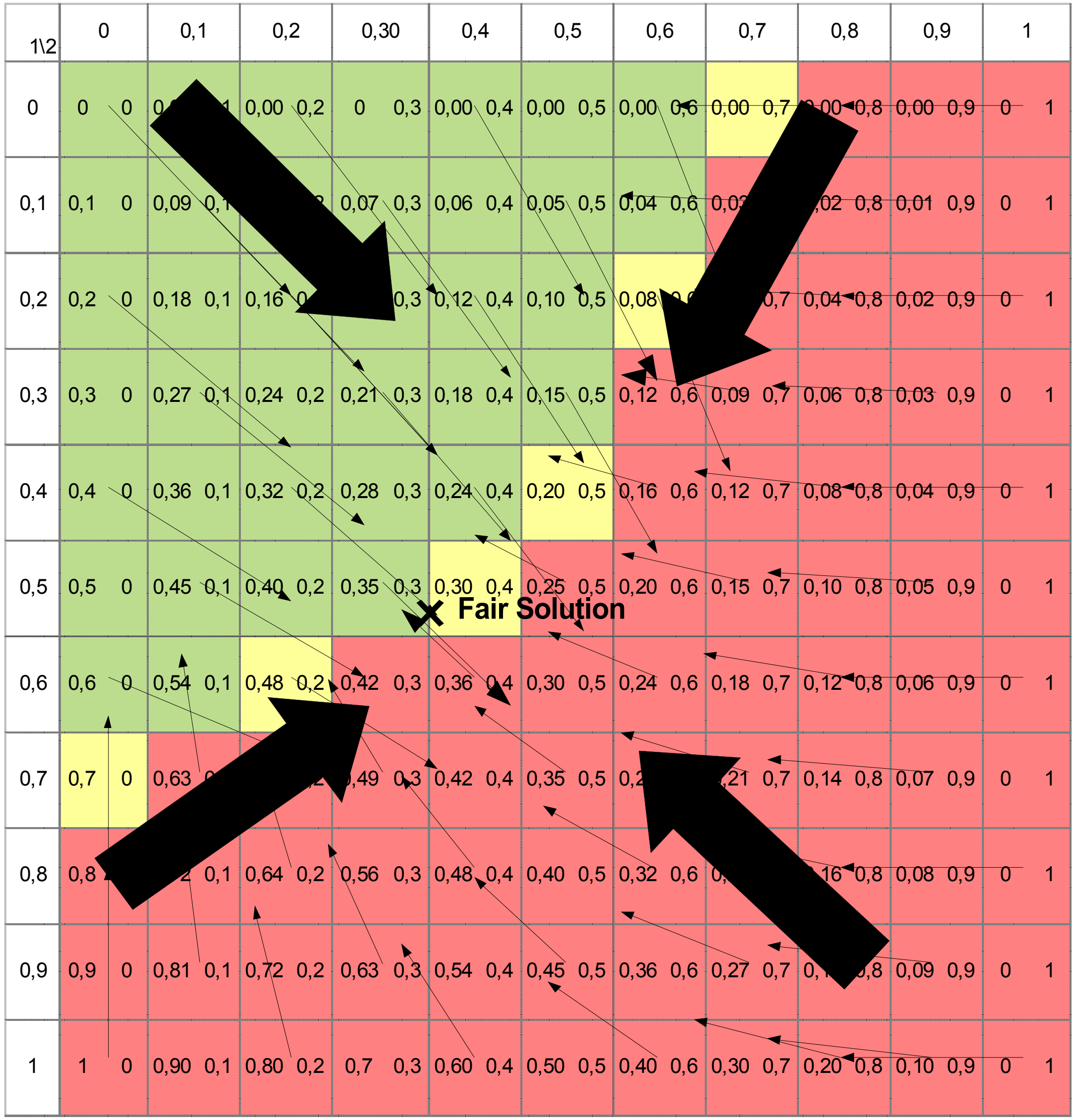
Example (3)

- Repeated Game:

$$\varepsilon = 0,3$$

- Reinforcement Learning:

```
foreach learning interval {
        calculate(load); //overall utilization
        calculate(success); //individual utilization
        if (load > 1-ε) {
                p -= learningrate * success;
        } else {
                p += learningrate * (1-success);
        }
}
```



1\2	0	0,1	0,2	0,30	0,4	0,5	0,6	0,7	0,8	0,9	1
0	0 0	[illegible]	0,00 0,2	0 0,3	0,00 0,4	0,00 0,5	0,00 0,6	0,00 0,7	0,00 0,8	0,00 0,9	0 1
0,1	0,1 0	0,09 0,1	[illegible]	0,07 0,3	0,06 0,4	0,05 0,5	0,04 0,6	[illegible]	[illegible] 0,8	0,01 0,9	0 1
0,2	0,2 0	0,18 0,1	0,16 [illegible]	[illegible] 0,3	0,12 0,4	0,10 0,5	0,08 [illegible]	[illegible] 0,7	0,04 0,8	0,02 0,9	0 1
0,3	0,3 0	0,27 0,1	0,24 0,2	0,21 0,3	0,18 0,4	0,15 0,5	0,12 0,6	0,09 0,7	0,06 0,8	0,03 0,9	0 1
0,4	0,4 0	0,36 0,1	0,32 0,2	0,28 0,3	0,24 0,4	0,20 0,5	0,16 0,6	0,12 0,7	0,08 0,8	0,04 0,9	0 1
0,5	0,5 0	0,45 0,1	0,40 0,2	0,35 0,3	0,30 0,4	0,25 0,5	0,20 0,6	0,15 0,7	0,10 0,8	0,05 0,9	0 1
0,6	0,6 0	0,54 0,1	0,48 0,2	0,42 0,3	0,36 0,4	0,30 0,5	0,24 0,6	0,18 0,7	0,12 0,8	0,06 0,9	0 1
0,7	0,7 0	0,63 [illegible]	[illegible]	[illegible] 0,3	0,42 0,4	0,35 0,5	[illegible]	[illegible] 0,7	0,14 0,8	0,07 0,9	0 1
0,8	0,8 [illegible]	[illegible] 0,1	0,64 0,2	0,56 0,3	0,48 0,4	0,40 0,5	0,32 0,6	[illegible]	[illegible] 0,8	0,08 0,9	0 1
0,9	0,9 0	0,81 0,1	0,72 0,2	0,63 0,3	0,54 0,4	0,45 0,5	0,36 0,6	0,27 0,7	[illegible] 0,8	0,09 0,9	0 1
1	1 0	0,90 0,1	0,80 0,2	0,7 0,3	0,60 0,4	0,50 0,5	0,40 0,6	0,30 0,7	0,20 0,8	0,10 0,9	0 1

Conclusion

- Game theory can be used to:
  - Model
  - Understand
  - Proof the behavior of multi-agent systems

- With reinforcement learning:
  - Achieve the proven behavior
  - Achieve behavior not proven

- Limits:
  - Reinforcement learning is parameter dependent
  - Time response can not be modeled (no real-time)
  - Only for certain systems behavior can be proven

Thanks for your Attention

- Project page:
  - www12.informatik.uni-erlangen.de/research/organicbus/

- Contact:
  - Tobias Ziermann
  - tobias.ziermann@informatik.uni-erlangen.de
  - www12.informatik.uni-erlangen.de/people/ziermann

- References:
  - [1] Fudenberg et al "The Theory of Learning in Games"
  - [2] Busonium et al "A Comprehensive Survey of Multiagent Reinforcement Learning"