Plan-Based Reward Shaping for Multi-Agent Reinforcement Learning

INFO-F-409 – Learning dynamics

Jérome BASTOGNE, Maxime DESCLEFS, Simon PICARD

Université Libre de Bruxelles, Boulevard du Triomphe - CP 212, 1050 Brussels, Belgium

jbastogn@ulb.ac.be, mdesclef@ulb.ac.be, spicard@ulb.ac.be

January 15 2016

Introduction

Content

Field

- Reinforcement learning
- Multi-agent
- Reward shaping
- Plan

Aim of the work

- Is reward shaping efficient ?
- Which heuristic is good ?
- What happens when combining them ?
- Is there a gap between individual-plan and joint-plan based reward shaping ?
- Why is there a gap and how to reduce it ?

Reinforcement Learning

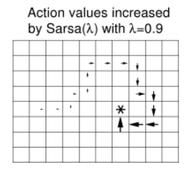
- Machine learning
- Goal directed
- Environment
- Agent
- Given actions
- Reward
- Repeated experiences
- Exploration $\rightarrow$ ϵ-Greedy

MDP and Algorithm

- MDP = $< S, A, T, R >$ Markov Property
- Temporal Difference Algorithm : $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s', a') - Q(s,a)]$
- r : reward α : learning rate γ : discount factor

Eligibility traces

- For current (s, a) :
  $\sigma = r + \gamma Q(s', a') - Q(s,a)$
- For all (s, a) in path :
  $Q(s,a) \leftarrow Q(s,a) + \alpha * \sigma * (\gamma * \lambda)^t$
- λ : decay rate



Reward Shaping

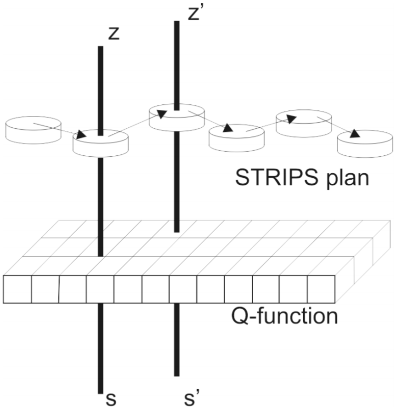
Basic

- Prior knowledge
- Better results
- $Q(s, a) \leftarrow Q(s, a) + \alpha[r + F(s, s') + \gamma Q(s', a') - Q(s, a)]$
- $F(s, s') = \gamma\phi(s') - \phi(s)$
- Potential function over a state

SARSA(λ) with reward shaping

- For current (s, a) : $\sigma = r + F(s, s') + \gamma Q(s', a') - Q(s, a)$
- For all (s, a) in path : $Q(s, a) \leftarrow Q(s, a) + \alpha * \sigma * (\gamma * \lambda)^t$

Plan Based Reward Shaping



Main idea

- Plan : set of subgoals
- Subgoals : state of the agent $\rightarrow$ domain specific
- To be followed
- Reward proportional to the distance of the step in the plan

Potential Function

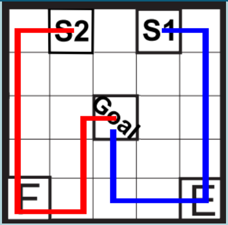
- $\phi(s) = \omega * CurrentStepInPlan$
- ω : scaling factor
- $\omega = MaxReward / NumStepsInPlan$
- Max shaping reward = max domain reward

Multi-Agent Planning

Centralized Planning

- Generate global plan
- Decompose it
- Assign task to multiple agents
- Divulge plans and goals

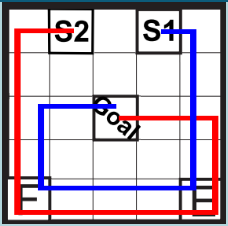
→ Joint-plan



Decentralized Planning

- Each agent set its own plan
- Do not divulge plans and goals

→ Individual-plan



Problem

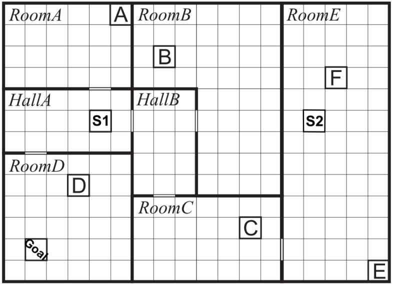


Figure – Multi-Agent, Flag-Collecting Problem Domain.

Description

- Two agents
- Six flags
- Seven rooms
- One goal
- Reward $\begin{cases} \textit{on goal} = \textit{Flags} * 100 \\ \textit{not on goal} = 0 \end{cases}$
- Agent knows its position
- Agent knows the flags it collected
- Episode : start to goal

Plan Handling

0	robot−in_hallA		
1	robot−in_roomA		
2	robot−in_roomA	taken_flagA	
3	robot−in_hallA	taken_flagA	
4	robot−in_hallB	taken_flagA	
5	robot−in_roomB	taken_flagA	
6	robot−in_roomB	taken_flagA	taken_flagB
7	robot−in_hallB	taken_flagA	taken_flagB
8	robot−in_hallA	taken_flagA	taken_flagB
9	robot−in_roomD	taken_flagA	taken_flagB

Figure – State based plan

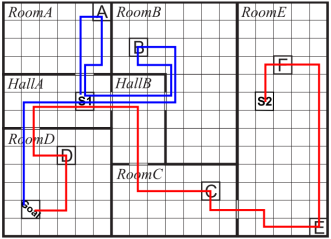


Figure – Joint-plan

- Action based to state based
- Each agent has an individual-plan and a joint-plan

Heuristics

Flag-Based

- $\phi(s) = \mathit{NumFlagsCollected} * \omega$
- $\omega = \mathit{MaxReward} / \mathit{MaxFlagsInWorld}$

Plan-Based

- $\phi(s) = \mathit{CurrentStepInPlan} * \omega$
- $\omega = \mathit{MaxReward} / \mathit{NumStepsInPlan}$
- Not in plan $\rightarrow$ last step in plan

Flag-Based and Plan-Based

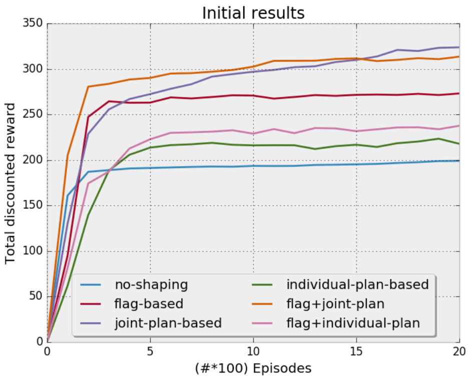
- $\phi(s) = \omega * (\mathit{CurrentStepInPlan} + \mathit{NumFlagsCollected})$
- $\omega = \mathit{MaxReward} / (\mathit{NumStepsInPlan} + \mathit{NumFlagsInWorld})$

Experiments

Modus operandi

- SARSA(λ)
- ϵ-Greedy
- $\alpha = 0.1$
- $\gamma = 0.99$
- $\epsilon = 0.1$
- $\lambda = 0.4$
- Q-values initialized to 0
- 2000 episodes
- Average over 30 simulations
- Discounted total reward over episodes
- Discounted total reward = $\mathit{reward} * \gamma^{\mathit{steps}}$
- Value averaged over 100 previous episodes

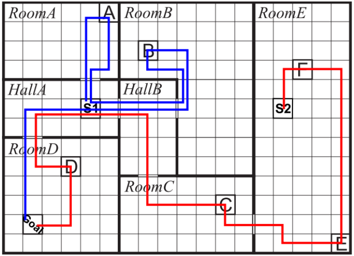
Initial Results

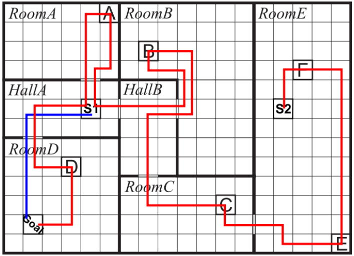


Analysis

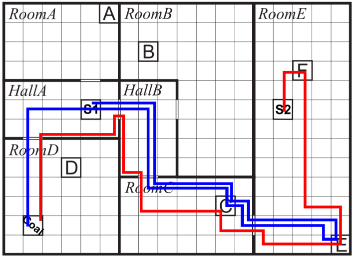
- Lower bound : no shaping
- Upper bound : joint-plan
- Individual-plan : poor results
- Flag-based : inefficient path
- Plan-based and flag-based : add knowledge

Conflicted Knowledge

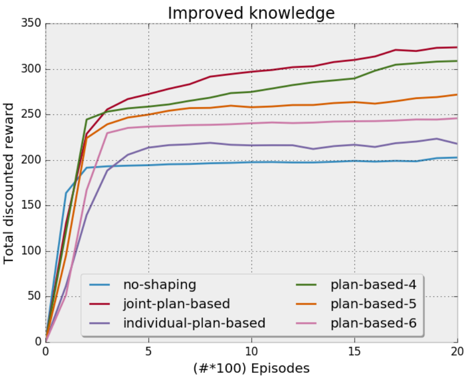




- Poor behaviour with individual plans
- Conflict knowledge
- How to avoid it ?
- **Make individual-plan based reward shaping as efficient as the joint-plan one**



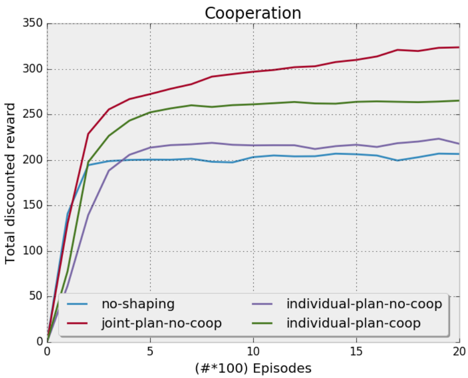
Partial Knowledge



Explanation and Analysis

- Delayed conflict
- Removed conflict
- Significant improvement
- Need global knowledge

Improved Cooperation



Explanation and Analysis

- Agents share collected flags
- Minimal communication
- Clear improvement
- Does not reach joint-plan
- Individual-plan remains non optimal

Conclusion

- Knowledge improves results
- Joint-plan is optimal
- Individual-plans leads to conflicted knowledge
- Posterior cooperation is not sufficient to overcome it
- Transforming the plan to avoid conflict works $\rightarrow$ Is it possible to automate it ?

Other and Future Work

Other work

- Exploration can overcome conflict knowledge but needs more episodes
- Abstract-MDP reward shaping

Future Work

- Use specific MARL algorithms
- Modify potential function according to domain

Knowledge revision

Interpret conflict knowledge as bad or incomplete knowledge and use knowledge revision

$\rightarrow$ Two steps :

- Implement knowledge revision for multi-agent
- Experiment it with individual-plans