Constructivist Approach to State Space Adaptation in Reinforcement Learning

Maxime Guériau[1], Nicolás Cardozo[2] and Ivana Dusparic[1]

[1]School of Computer Science and Statistics,
Trinity College Dublin
Ireland
maxime.gueriau@scss.tcd.ie, @maximegueriau
ivana.dusparic@scss.tcd.ie, @ivanadusparic

[2]Systems and Computing Engineering
Department,
Universidad de los Andes
Colombia
n.cardozo@uniandes.edu.co, @ncardoz

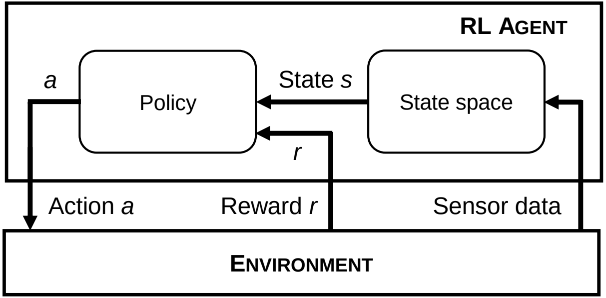
13th IEEE International Conference on Self-Adaptive and Self-Organizing Systems
Umeå, Sweden

enable connecting · communities

IRISH RESEARCH COUNCIL
An Chomhairle um Thaighde in Éirinn

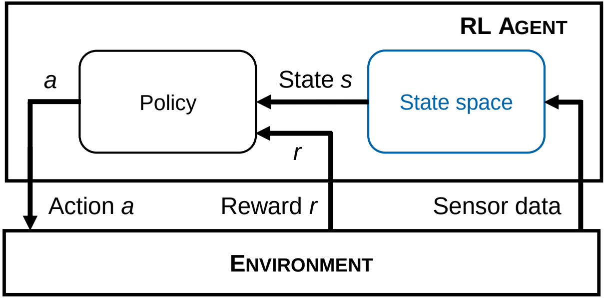
Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

State space adaptation in reinforcement learning



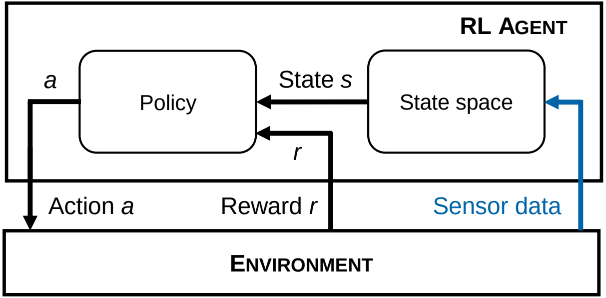
- When does an agent need to adapt its state space?

State space adaptation in reinforcement learning



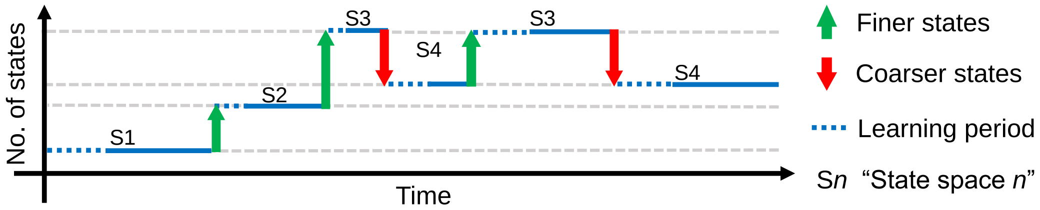
- When does an agent need to adapt its state space?
  - when its original state space is too big/small

State space adaptation in reinforcement learning



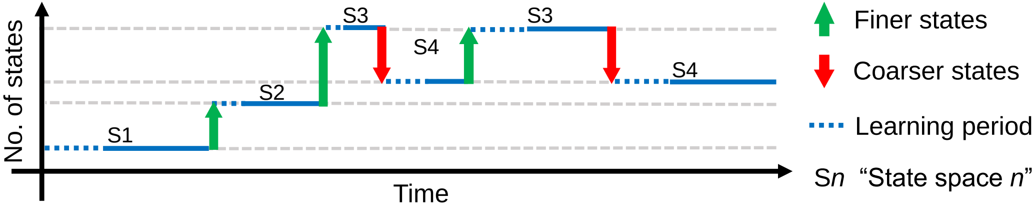
- When does an agent need to adapt its state space?
  - when its original state space is too big/small
  - when sensors are added or removed dynamically
  - when sensors input granularity changes over time

State space adaptation in reinforcement learning



- How to enable this dynamic adaptation?

State space adaptation in reinforcement learning



- How to enable this dynamic adaptation?

1. By generating, learning or adapting one state space
2. By switching between several state spaces

Existing approaches

1. Generating, learning or adapting one state space

- State space refinement methods, usually from a grid-based state space
  - ✓ State aggregation techniques [1] allow to reduce a state space size
  - ✓ And states can be divided into finer ones [6]
  - ✗ But it highly depends on the initial grid granularity
- Function approximators can generate a state space from the agent inputs
  - ✓ Allows for an adaptive input space partitioning [9]
  - ✗ But can be specific to the RL algorithm (*e.g.* TD in [9])

Existing approaches

1. Generating, learning or adapting one state space
   - Clustering techniques enable a dynamic state space generation from continuous inputs
     - ✓ Using supervised algorithms like Vector Quantization [2]
     - ✓ Using a self-organizing network like Growing Neural Gas [3, 12]
     - ✓ Can adapt the state space where the policy is updated (GNG-Q [3]) or for tracking rewards (TD-GNG [12])
     - ✗ However this process can be hard to apply online [3]
     - ✗ Or requires additional mechanisms to control the state space size [12]

Existing approaches

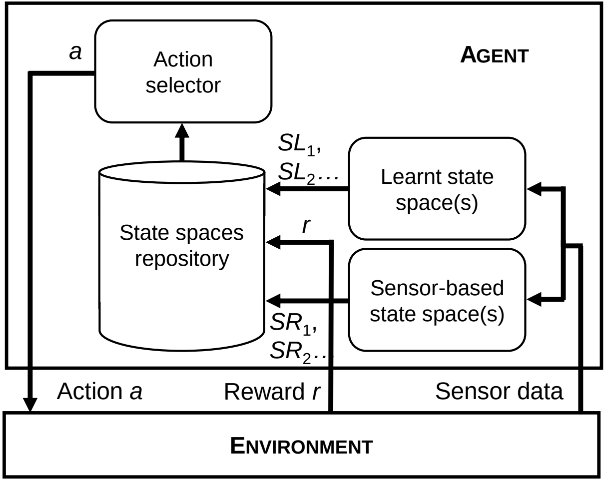
2. Switching between several state spaces
   - ✓ Can applied for multi-objective RL [11]
   - ★ Enables to cope with environment/observation changes
   - ★ Allows to keep different state space granularities

Constructivist approaches

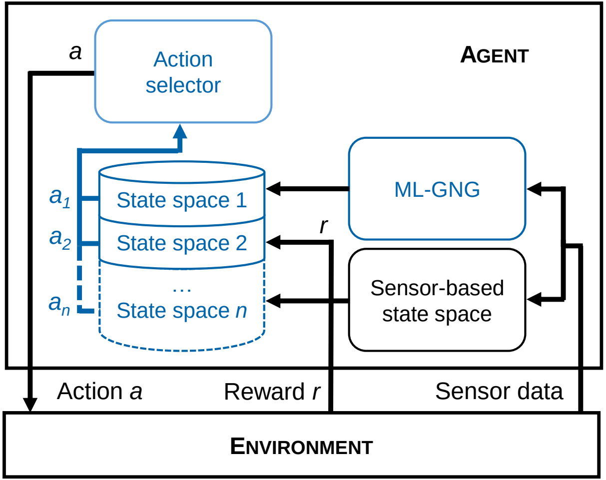
- Inspired from a theory [8] that models human mind construction process
- Models the continuous construction and adaptation of knowledge through accommodation and assimilation
- A framework with RL has been proposed, but at a conceptual level [10]

Outline

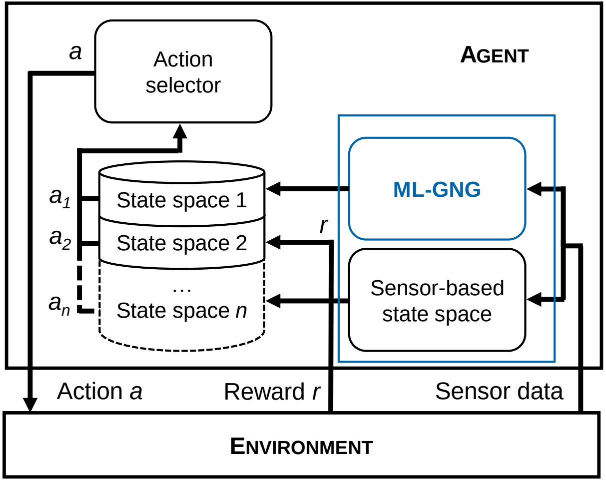
Con-RL: Constructivist RL for dynamic state space adaptation



Con-RL: Constructivist RL for dynamic state space adaptation



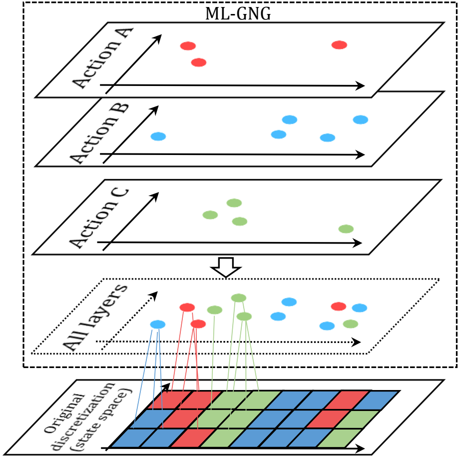
Dynamic state space learning



Dynamic state space learning

A Multi-Layer Growing Neural Gas

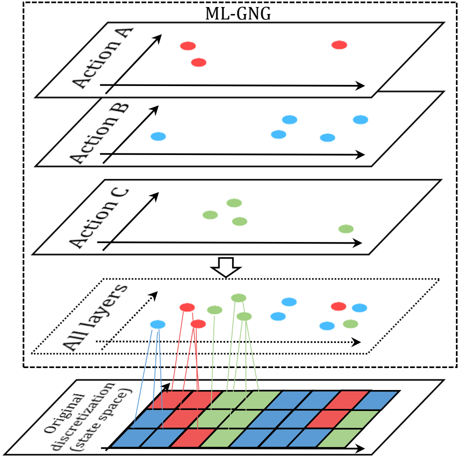
- Each layer is a Growing Neural Gas (GNG) [4], specialized in one action
- Each layer is a self-organizing network that learns where actions are taken in the input space from the sensor-based state space
- A layer is triggered when an action has been executed θ times for the same state



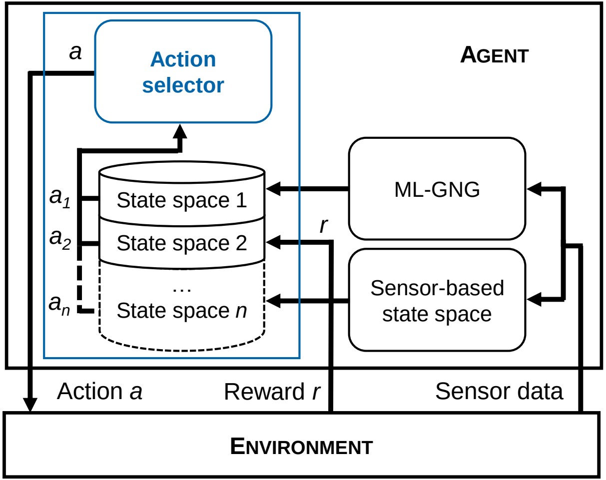
Dynamic state space learning

ML-GNG:

- combines all layers as a new learnt state space
- provides the agent with a generalization of the underlying Q-learning policy
- allows to speed-up learning by simplifying the sensor-based state space during firsts episodes



Dynamic state space selection



Dynamic state space selection

Action selection relies on:

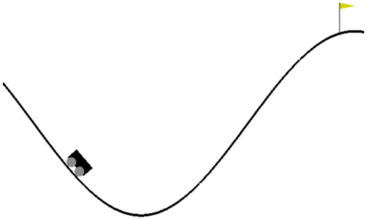
- a *confidence* value:
  - distance from current input to the nearest GNG node in ML-GNG
  - number of times the same action was executed in the given state
- two configurable thresholds (one for ML-GNG and one for the sensor-based state space)

The action selector picks:

- the policy from the representation with the highest confidence if one or both are above the defined threshold
- a random action if none reaches this condition (to allow more exploration)

Evaluation

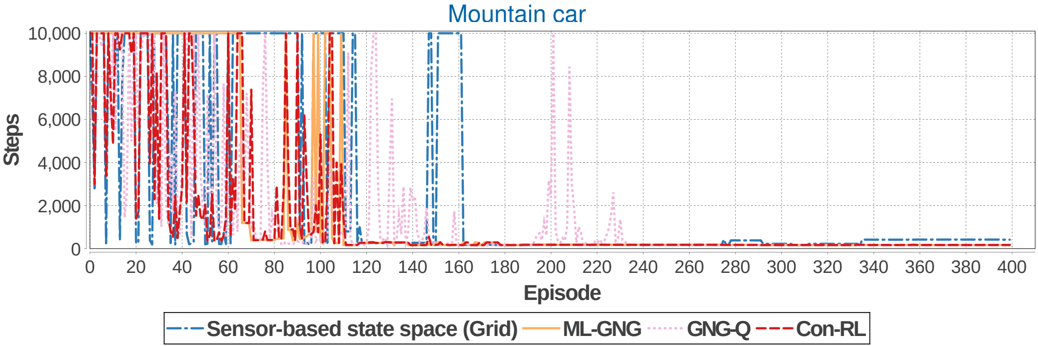
Mountain car



Parameter	Value/Range
State: Position	$[-1.2, 0.6]$ **(goal at 0.6)**
State: Velocity	$[-0.07, 0.07]$
Actions	Left -1, Neutral 0 or Right 1
Reward	100 if at the goal, -10 otherwise.

- Sensor-based discretization: 10x10 grid-based state space
- Q-learning parameters: $\alpha = 0.1$, $\gamma = 0.9$, epsilon decay policy $\epsilon = \exp^{-Et}$, $E = 0.015$
- ML-GNG parameters: $\lambda = 10$, $a_{max} = 200$, $\alpha = 0.5$, $\beta = 0.05$, $k = 1000$, $\epsilon_b = 0.5$, $\epsilon_n = 0.1$ and $\theta = 20$
- Q-learning GNG (GNG-Q [3]) parameters: $\alpha = 0.1$, $\gamma = 0.95$, $\lambda = 1000$, $a_{max} = 100$, $\epsilon_b = 0.5$, and $\epsilon_n = 0.1$

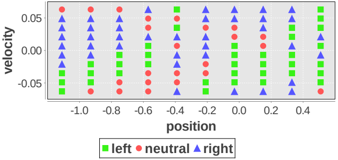
Evaluation



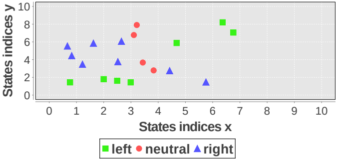
- ✓ ML-GNG builds up on an existing state space and learns from previously taken actions
- ✘ GNG-Q requires more time to converge
- ✓ Con-RL speeds-up learning at early episodes and ensures long-term performance

Evaluation

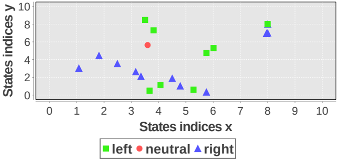
Mountain car



Policy of sensor-based state space (grid)



ML-GNG nodes position and action



GNG-Q nodes position and learnt actions

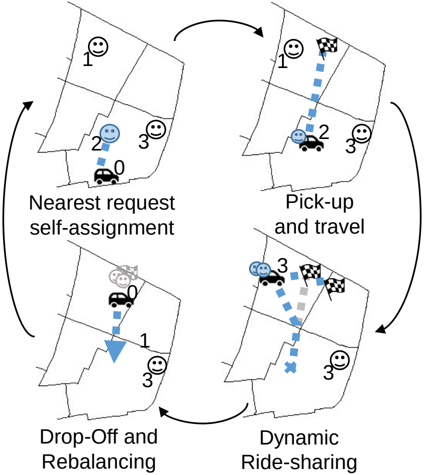
- ✓ Grid, GNG-Q and ML-GNG converge to similar policies
- ✓ ML-GNG provides a generalisation of the sensor-based state space
- ✓ Con-RL dynamically adapts the representation

Evaluation

Shared Autonomous Mobility on Demand [5]

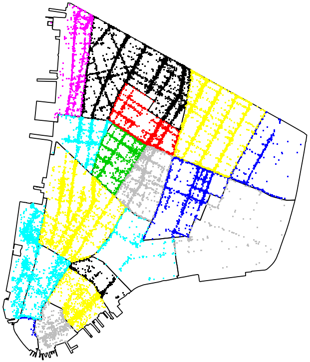
Parameter	Value/Range
State: Occupancy	0,1,2,3,4 **(goal $>$ 1)**
State: Req. in own zone	$0, 1, 2, \ldots, 10+$
State: Req. in neighb. zone	$0, 5, 10, \ldots 20+$
Actions	pick up, rebalance, idle
Reward	100 at goal, 0 otherwise

- Each car is an agent, learning how to serve requests
- Goal is to travel with one passenger or more



Evaluation

Shared Autonomous Mobility on Demand [5]



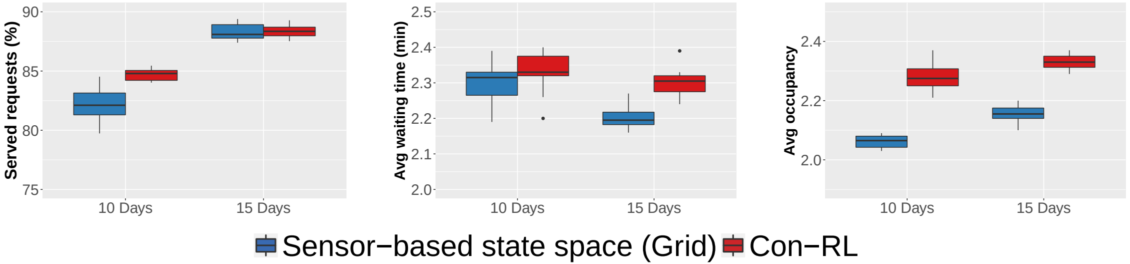
NYC taxi requests data [7]
15 consecutive Tuesdays
(7am–10am)

- 200 SAMoD vehicles agents
- Sensor-based state space = 275 states:
  - 5 occupancies
  - 11 own zone requests number
  - 5 neighbouring zones requests number
- Q-learning parameters: $\alpha = 0.1$, $\gamma = 0.9$, epsilon decay policy $\epsilon = \exp^{-Et}$, $E = 0.001$
- ML-GNG parameters: $\lambda = 10$, $a_{max} = 200$, $\alpha = 0.5$, $\beta = 0.05$, $k = 1000$, $\epsilon_b = 0.5$, $\epsilon_n = 0.1$ and $\theta = 20$

Evaluation

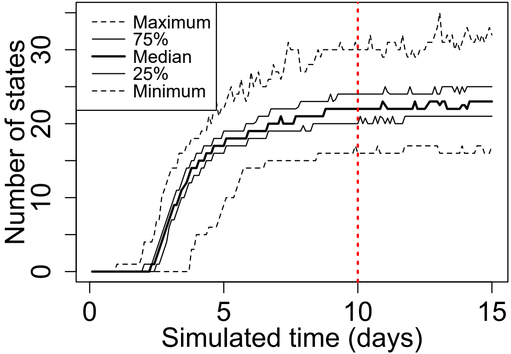
Shared Autonomous Mobility on Demand [5]

	5 days		8 days		10 days		15 days	
	Grid	Con-RL	Grid	Con-RL	Grid	Con-RL	Grid	Con-RL
Served requests (%)	52.898	73.692	71.625	80.324	82.201	84.703	88.26	88.367
Avg waiting time (min)	3.071	2.807	2.57	2.594	2.304	2.329	2.203	2.304
Avg occupancy	2.274	2.492	2.103	2.327	2.063	2.282	2.154	2.33

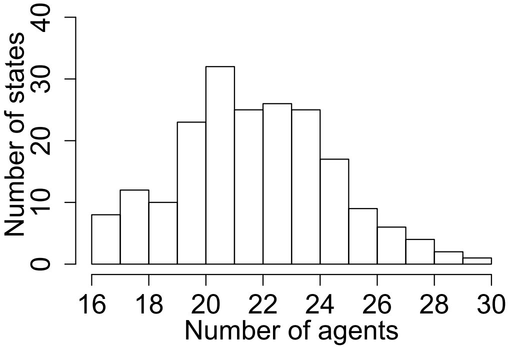


Evaluation

Shared Autonomous Mobility on Demand [5]



Evolution of ML-GNG number of states during learning



Distribution of ML-GNG number of states after 10 days

Conclusions

Summary

- We proposed Con-RL: an approach for autonomous state space learning and adaptation
- Con-RL combines:
  - ML-GNG, a multi-layered clustering technique to learn optimized state space at runtime;
  - A state space selector, that picks the most suitable representation to base the action decision on
- Con-RL was evaluated in two case studies:
  - A single agent mountain car scenario
  - A multi-agent ride-sharing simulation

Conclusions

Achievements and remaining challenges

- ✓ Con-RL can remove the need for manual state space specification:
  - ✓ it reduces the size of the sensor-based state space to lower the learning time;
  - ✓ but it also allows for an accurate long-term policy learning.
- ★ The behaviour of Con-RL needs further investigation:
  - ★ when new sensors are added/removed at runtime
  - ★ when more representations/sensors are available at the same time

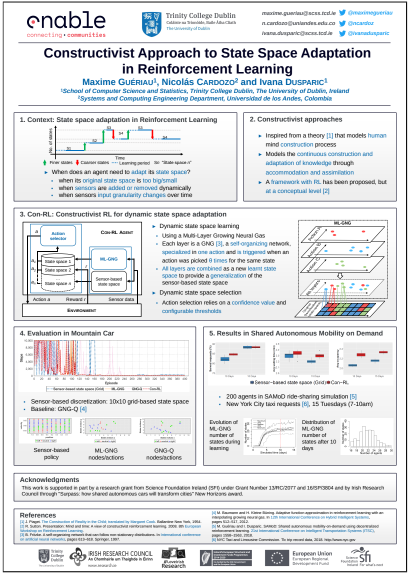
Thank you for your attention

Maxime GUÉRIAU
PhD., Research Fellow
School of Computer Science and Statistics
Trinity College Dublin
Ireland

maxime.gueriau@scss.tcd.ie

`maxime.gueriau.fr`

@maximegueriau



References I

[1] D. Abel, D. E. Hershkowitz, and M. L. Littman. Near optimal behavior via approximate state abstraction. In Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48, pages 2915–2923. JMLR. org, 2016.

[2] M. Abramson, P. Pachowicz, and H. Wechsler. Competitive reinforcement learning in continuous control tasks. In Proceedings of the International Neural Network Conference, 2003.

[3] M. Baumann and H. Kleine Büning. Adaptive function approximation in reinforcement learning with an interpolating growing neural gas. In 12th International Conference on Hybrid Intelligent Systems, pages 512–517, 2012.

[4] B. Fritzke. A self-organizing network that can follow non-stationary distributions. In International conference on artificial neural networks, pages 613–618. Springer, 1997.

[5] M. Guériau and I. Dusparic. SAMoD: Shared autonomous mobility-on-demand using decentralized reinforcement learning. 21st International Conference on Intelligent Transportation Systems (ITSC), pages 1558–1563, 2018.

[6] J. A. Martín H, J. de Lope, and D. Maravall. Robust high performance reinforcement learning through weighted k-nearest neighbors. Neurocomputing, 74(8):1251 – 1259, 2011.

References II

[7] NYC Taxi and Limousine Commission. Tlc trip record data, 2018. URL http://www.nyc.gov.

[8] J. Piaget. The Construction of Reality in the Child; translated by Margaret Cook. Ballantine New York, 1954.

[9] K. Samejima and T. Omori. Adaptive internal state space construction method for reinforcement learning of a real-world agent. Neural Networks, 12(7-8):1143–1155, 1999.

[10] R. Sutton. Presentation: Mind and time: A view of constructivist reinforcement learning. 2008. 8th European Workshop on Reinforcement Learning.

[11] K. Van Moffaert and A. Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. The Journal of Machine Learning Research, 15(1):3483–3512, 2014.

[12] D. C. D. L. Vieira, P. J. L. Adeodato, and P. M. Goncalves. A temporal difference gng-based approach for the state space quantization in reinforcement learning environments. In IEEE 25th International Conference on Tools with Artificial Intelligence, pages 561–568, Nov 2013. doi: 10.1109/ICTAI.2013.89.