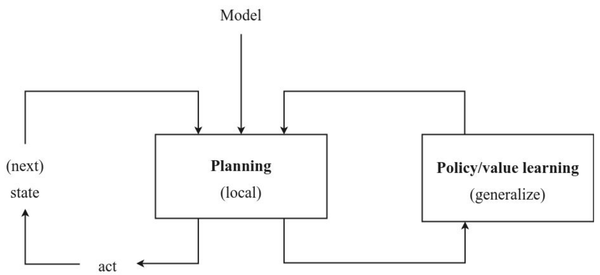
Think too fast nor too slow:
The computational trade-off between planning and reinforcement learning

Thomas Moerland, Anna Deichler, Simone Baldi, Joost Broekens and Catholijn Jonker
Delft University of Technology, The Netherlands

Multi-step Approximate Real-time Dynamic Programming (MSA-RTDP)

1. Multi-step: multi-step lookahead
2. Approximate: function approximation for policy/value
3. Real-time: On trace from some start state

Recently very succesful class of algorithms, e.g., AlphaGo Zero



Question: How should we trade-off planning and learning/acting?
In other words: how long should we plan before every real step?

Idea

	+	-
High planning budget per timestep (think slow)	More accurate training targets	Less training targets & real steps
Low planning budget per timestep (think fast)	More training targets & real steps	Less accurate training targets

There might be a trade-off between planning too short and too long!

Approach

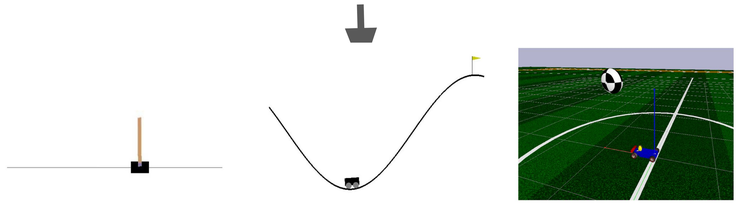
A. Use AlphaGoZero algorithm:
1. **Plan:** MCTS
2. **Train:** Neural network, approximation of policy $\pi_\theta(a|s)$ and value $V_\theta(s)$
3. **Act/real-step**

B. Fix total training time budget on each test task, but *vary the planning budget per timestep*.

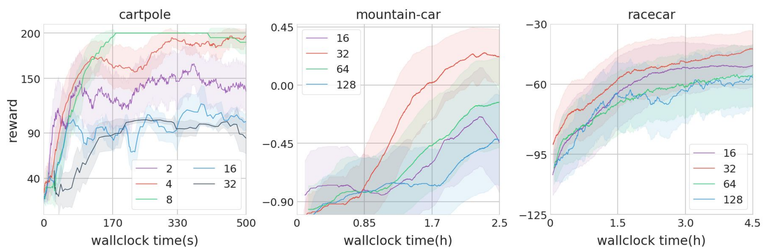
C. Look at effect of different step-wise planning budgets on performance.

Experiments

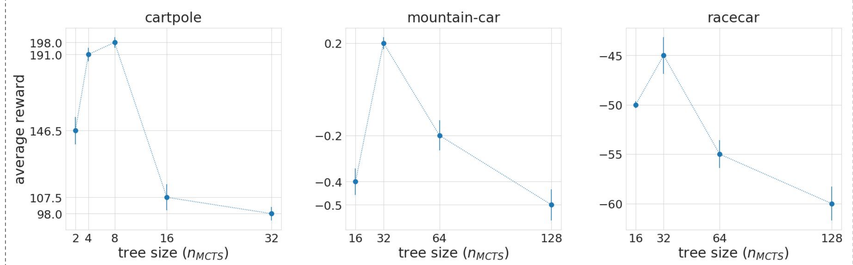
A. *Three tasks:* Cartpole, MountainCar and RaceCar



B. *Learning curves:*

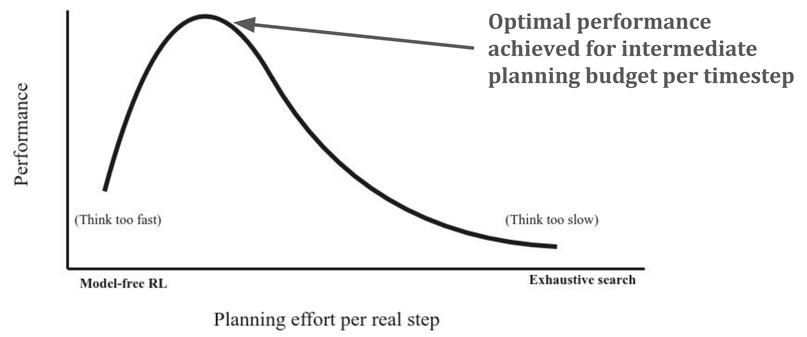


C. *Key plot*: Planning budget versus final 15% training performance (prev. graph).



Discussion

We face a new spectrum between full planning and full learning:
- No planning at every timestep = model-free RL
- Full planning at every timestep = exhaustive search



Future work

How should the planning budget per timestep depend on the *context*, in the form of:
- the type of task
- the data seen so far in the task

Email: T.M.Moerland@tudelft.nl