ETH zürich

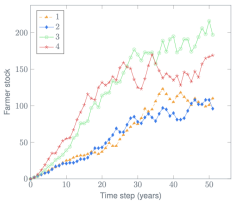
Computational Social Science

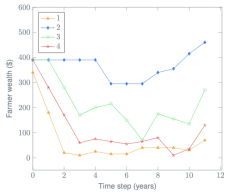
CropWar

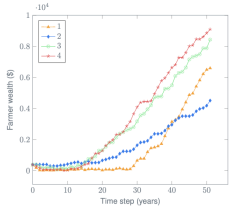
851-0101-86 S Complex Social Systems: Modeling Agents, Learning, and Games

CropWar: Agent-based simulation of agricultural interactions

C.Golling, G.Mourouga, A.Moser, O.Schmidt, A.H.Keshavarzzadeh







Content

Introduction and motivation

`CropWar` aims at modelling **interactions between farmers** growing crops in a **geographic area** with a finite amount of **water ressources** and selling them on a **market** to feed a population.

Introduction and motivation

`CropWar` aims at modelling **interactions between farmers** growing crops in a **geographic area** with a finite amount of **water ressources** and selling them on a **market** to feed a population.
The model is organised as follows:

Introduction and motivation

`CropWar` aims at modelling **interactions between farmers** growing crops in a **geographic area** with a finite amount of **water ressources** and selling them on a **market** to feed a population.
The model is organised as follows:

- v1.1 basic version: growing, selling, stocking

Introduction and motivation

`CropWar` aims at modelling **interactions between farmers** growing crops in a **geographic area** with a finite amount of **water ressources** and selling them on a **market** to feed a population.
The model is organised as follows:

- v1.1 basic version: growing, selling, stocking
- v1.2 implementation of the `Map` class: spatial expansion

Introduction and motivation

`CropWar` aims at modelling **interactions between farmers** growing crops in a **geographic area** with a finite amount of **water ressources** and selling them on a **market** to feed a population.
The model is organised as follows:

- v1.1 basic version: growing, selling, stocking
- v1.2 implementation of the `Map` class: spatial expansion
- v1.3 implementation of the `Market` model: dynamic pricing and agent personalities

Introduction and motivation

`CropWar` aims at modelling **interactions between farmers** growing crops in a **geographic area** with a finite amount of **water ressources** and selling them on a **market** to feed a population.
The model is organised as follows:

- v1.1 basic version: growing, selling, stocking
- v1.2 implementation of the `Map` class: spatial expansion
- v1.3 implementation of the `Market` model: dynamic pricing and agent personalities
- v2.1 reinforcement learning: identifying the optimal strategy against Introvert agents

Introduction and motivation

`CropWar` aims at modelling **interactions between farmers** growing crops in a **geographic area** with a finite amount of **water ressources** and selling them on a **market** to feed a population.
The model is organised as follows:

- v1.1 basic version: growing, selling, stocking
- v1.2 implementation of the `Map` class: spatial expansion
- v1.3 implementation of the `Market` model: dynamic pricing and agent personalities
- v2.1 reinforcement learning: identifying the optimal strategy against Introvert agents
- v2.2 reinforcement learning: identifying the optimal strategy against Trader agents

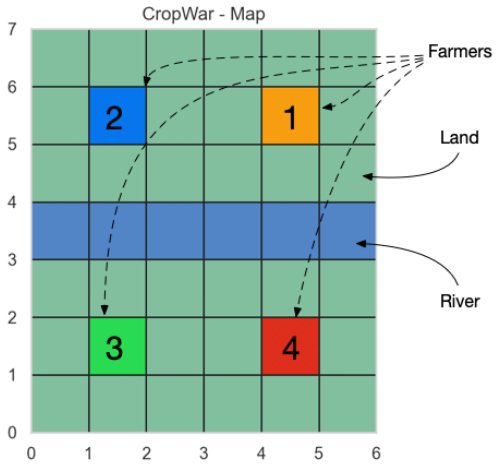
Deterministic versions

Basic version: selling and stocking

v1.1: Basic version

Deterministic versions

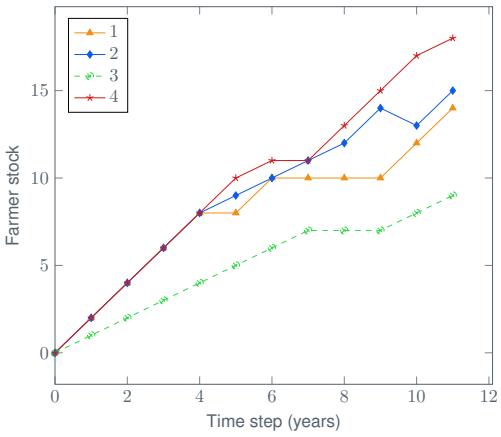
Basic version: selling and stocking



In v1.1 farmers only have access to one cell on which they can choose to grow one of two crops (A or B)

Deterministic versions

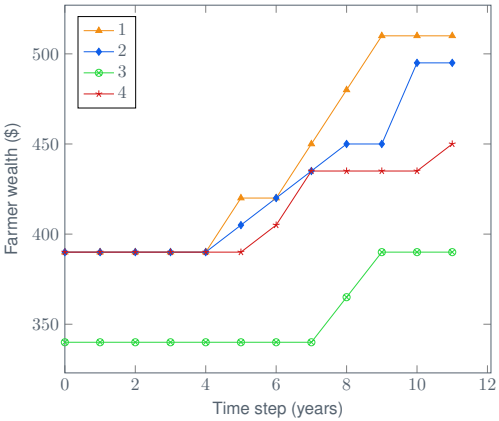
Basic version: selling and stocking



This plot shows the evolution of stock as a function of time:
Farmers 1,2 and 4 grow crop A
Farmer 3 grows crop B
At each time step, they chose to stock or sell their yield

Deterministic versions

Basic version: selling and stocking

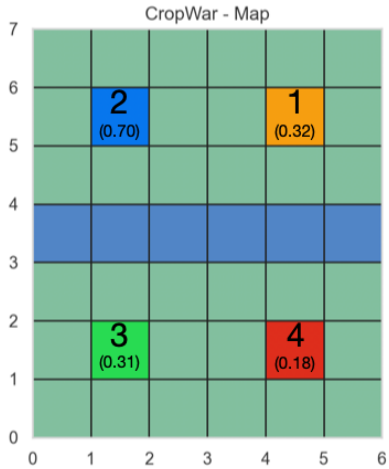


This plot shows the corresponding evolution of the farmer's wealth as a function of time. Selling corresponds to an increase in wealth, stocking to a constant value.

v1.2: Spatial expansion

Deterministic versions

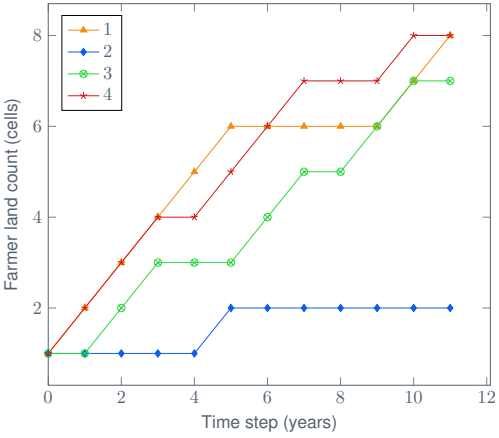
The Map class: expansion

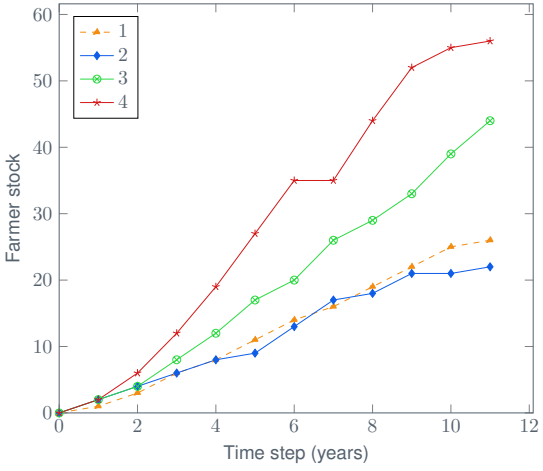


The `buy_threshold` (in brackets) indicates the tendancy of farmers to expand their farms by buying neighbouring land in order to grow more crops

Deterministic versions

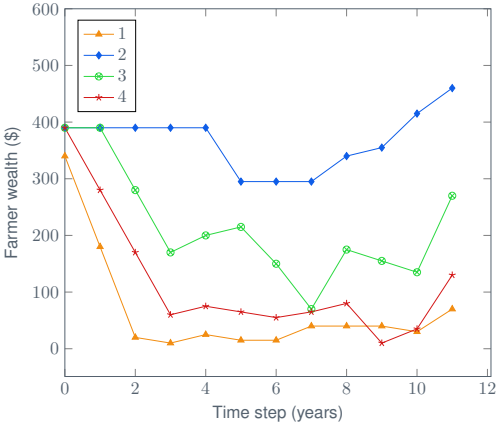
The Map class: expansion





Deterministic versions

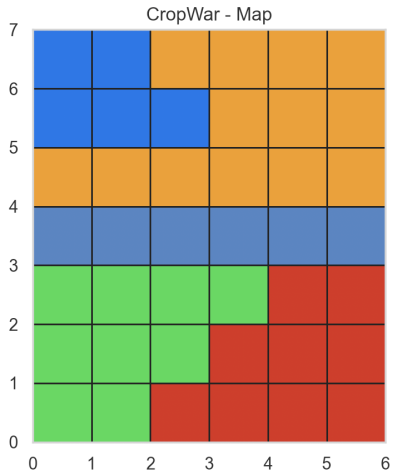
The Map class: expansion



Farmer 2 seems to remain wealthier than other farmers over 10 time steps, due to not spending money to expand

Deterministic versions

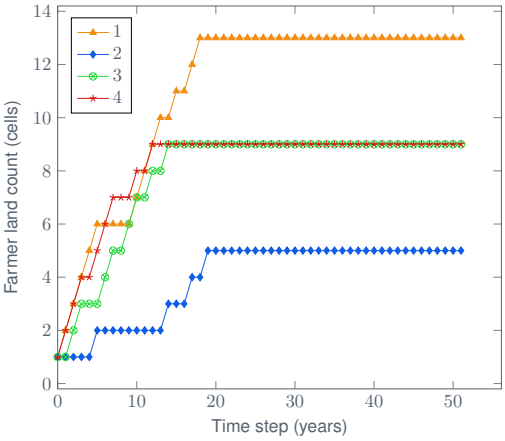
The Map class: expansion

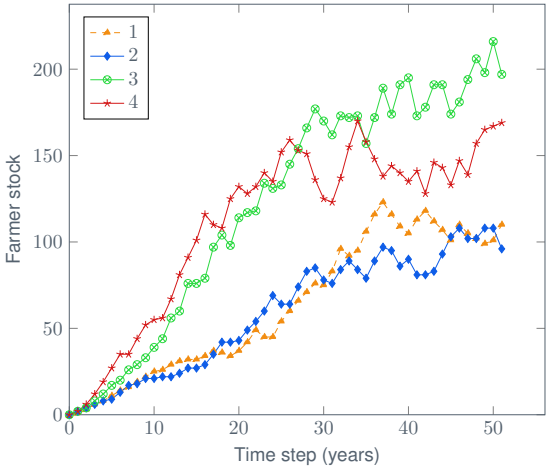


If we extend the simulation to 50 time steps, all available land is bought by farmers

Deterministic versions

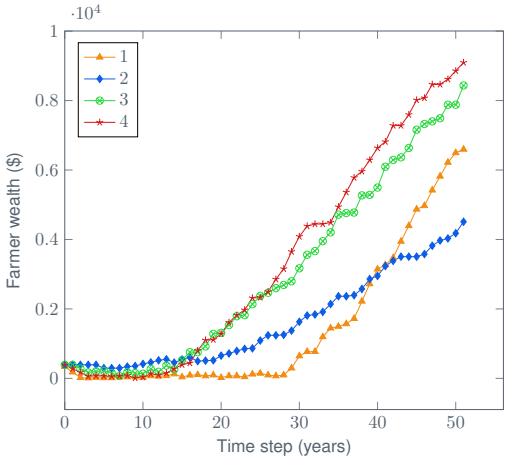
The Map class: expansion





Deterministic versions

The Map class: expansion



On the long run, farmers with more land become wealthier, although the type of crop also seems to have an importance

v1.3: Dynamic pricing of crops

In v1.3 we implement a `Market` class, which updates the price of crops based on supply and demand.

In v1.3 we implement a `Market` class, which updates the price of crops based on supply and demand.

1. Harvesting period: agents harvest according to their crop choice and add the harvest yield to the stock.

In v1.3 we implement a `Market` class, which updates the price of crops based on supply and demand.

1. Harvesting period: agents harvest according to their crop choice and add the harvest yield to the stock.
2. Interaction period: market interaction takes place.

In v1.3 we implement a `Market` class, which updates the price of crops based on supply and demand.

1. Harvesting period: agents harvest according to their crop choice and add the harvest yield to the stock.
2. Interaction period: market interaction takes place.
3. Strategy period: the Agent's personality influences its decision for the next time step.

In v1.3 we also implement agent personalities `Traders` and `Introverts`

In v1.3 we also implement agent personalities `Traders` and `Introverts`

- `Traders` agents react to market prices and may change the crops they were groing.

Deterministic versions

The Market model: dynamic pricing

In v1.3 we also implement agent personalities `Traders` and `Introverts`

- `Traders` agents react to market prices and may change the crops they were groing.
- `Introverts` agents pick a crop and stick with it.

Deterministic versions

The Market model: dynamic pricing

In v1.3 we also implement agent personalities `Traders` and `Introverts`

- `Traders` agents react to market prices and may change the crops they were groing.
- `Introverts` agents pick a crop and stick with it.

Some of these agents may also have different expanding strategies

Deterministic versions

The Market model: dynamic pricing

In v1.3 we also implement agent personalities `Traders` and `Introverts`

- `Traders` agents react to market prices and may change the crops they were groing.
- `Introverts` agents pick a crop and stick with it.

Some of these agents may also have different expanding strategies

- Stationary agents remain on their initial cell

Deterministic versions

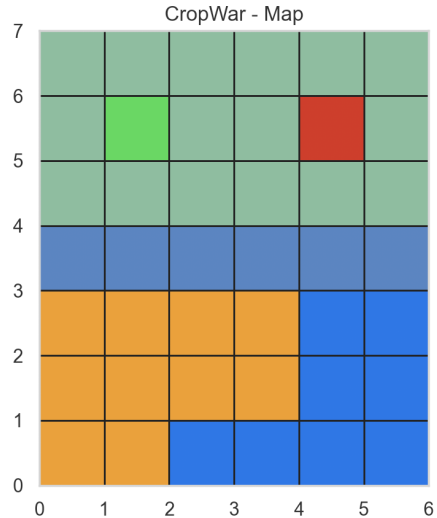
The Market model: dynamic pricing

In v1.3 we also implement agent personalities `Traders` and `Introverts`

- `Traders` agents react to market prices and may change the crops they were groing.
- `Introverts` agents pick a crop and stick with it.

Some of these agents may also have different expanding strategies

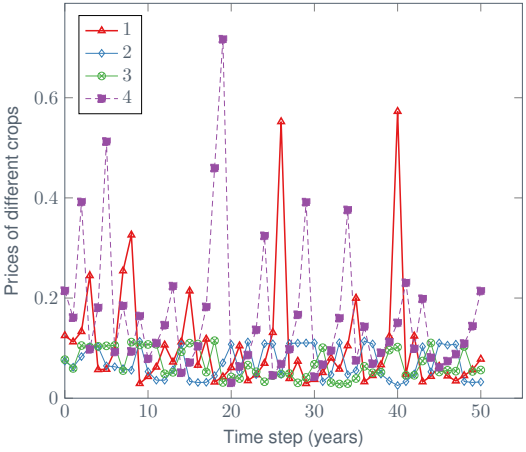
- Stationary agents remain on their initial cell
- Expanding agents may buy neighbouring land

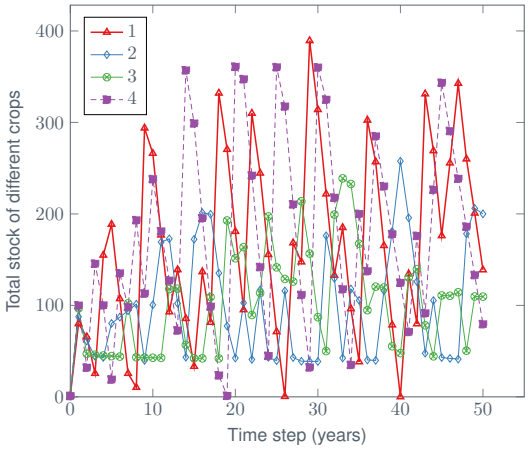


Simulation with 2 stationary `Introverts` and 2 expanding `Traders`

Deterministic versions

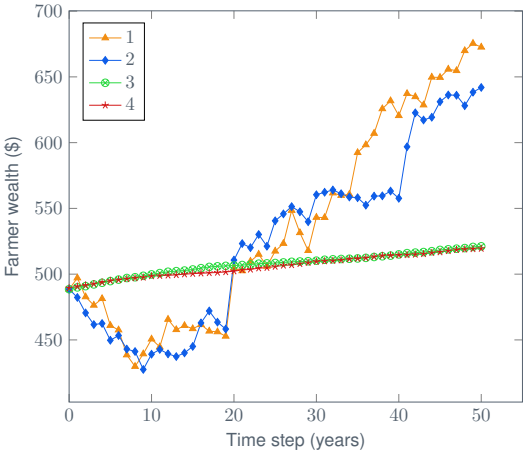
The Market model: dynamic pricing





Deterministic versions

The Market model: dynamic pricing



Farmer wealth as a function of time: expanding `Taders` agents take more risks, which seem to pay off on the long term

Reinforcement Learning

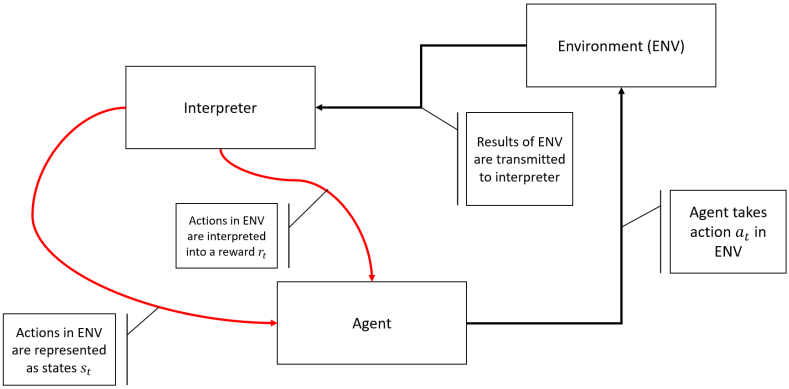
Reinforcement Learning versions

Introduction

The preceding versions v1.1 - v1.3 defined the deterministic agents and the model of the environment in which actions take place.
In order to see unpredictable, emerging behaviour, the agents would need to adjust to evolutions of the market and learn.

Reinforcement Learning versions

Introduction



Feedback loop in RL. An agent decides to do a certain action a_t based on his observation of the environment. This affects the environment, which yields a new state. The effect of the agents' action is then interpreted to update the strategy of the agent.

Reinforcement Learning versions

Proximal Policy Optimization

$$\underbrace{L(\theta)}_{\substack{\text{Policy}\\ \text{loss}\\ \text{function}}} = \underbrace{\hat{\mathbf{E}}_t}_{\substack{\text{expectation}\\ \text{at iteration}\\ t}} \left[\log(\underbrace{\pi_\theta(a_t|s_t)}_{\substack{\text{stochastic}\\ \text{policy}}}) \underbrace{\hat{A}_t}_{\substack{\text{estimator of}\\ \text{advantage of}\\ \text{current action}}} \right] \tag{1}$$

$$\underbrace{L(\theta)}_{\substack{\text{Policy}\\ \text{loss}\\ \text{function}}} = \underbrace{\hat{\mathbf{E}}_t}_{\substack{\text{expectation}\\ \text{at iteration}\\ t}} \left[\log(\underbrace{\pi_\theta(a_t|s_t)}_{\substack{\text{stochastic}\\ \text{policy}}}) \underbrace{\hat{A}_t}_{\substack{\text{estimator of}\\ \text{advantage of}\\ \text{current action}}} \right] \quad (1)$$

where:

$$\underbrace{L(\theta)}_{\substack{\text{Policy}\\ \text{loss}\\ \text{function}}} = \underbrace{\hat{\mathbf{E}}_t}_{\substack{\text{expectation}\\ \text{at iteration}\\ t}} \left[\log(\underbrace{\pi_\theta(a_t|s_t)}_{\substack{\text{stochastic}\\ \text{policy}}}) \underbrace{\hat{A}_t}_{\substack{\text{estimator of}\\ \text{advantage of}\\ \text{current action}}} \right] \tag{1}$$

where:

- $\pi_\theta(a_t|s_t)$ is given as transition probabilities in the MC of taking action a_t in state s_t. It is a neural network, that suggests an action for a given state based on previous training experience.

$$\underbrace{L(\theta)}_{\substack{\text{Policy}\\ \text{loss}\\ \text{function}}} = \underbrace{\hat{\mathbf{E}}_t}_{\substack{\text{expectation}\\ \text{at iteration}\\ t}} \left[\log(\underbrace{\pi_\theta(a_t|s_t)}_{\substack{\text{stochastic}\\ \text{policy}}}) \underbrace{\hat{A}_t}_{\substack{\text{estimator of}\\ \text{advantage of}\\ \text{current action}}} \right] \tag{1}$$

where:

- $\pi_\theta(a_t|s_t)$ is given as transition probabilities in the MC of taking action a_t in state s_t. It is a neural network, that suggests an action for a given state based on previous training experience.
- $\hat{A}_t$ is computed as a discounted reward. This neural net is updated with the experience (i.e. reward) that the agent collects in an environment.

Reinforcement Learning versions

ML agent trained against introverts

v2.1: RL agent trained againt Introvert farmers

The reward function that was used for training is given by

$$\frac{1}{3}\Big(\underbrace{b^2}_{\substack{\text{Fraction of}\\ \text{Global Budget}}} + \underbrace{r^2}_{\substack{\text{Current}\\ \text{Ranking}}} + \underbrace{s^2}_{\substack{\text{Fraction of}\\ \text{Global supply}}}\Big) \tag{2}$$

where r is 1 for first, 0.66 for second, 0.33 for third, and 0.0 for last agent.

Reinforcement Learning versions

ML agent trained against introverts

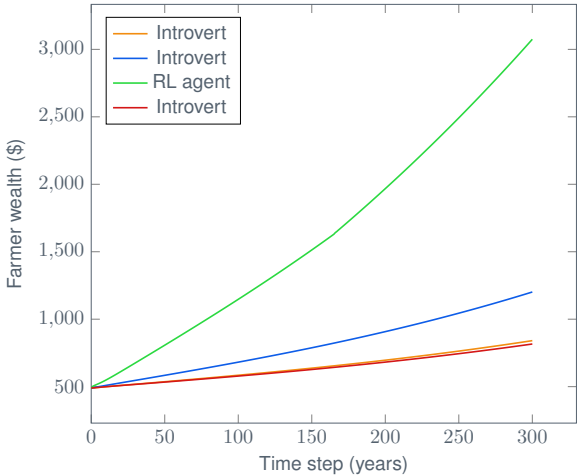


Figure: Budget evolution over 300 iterations

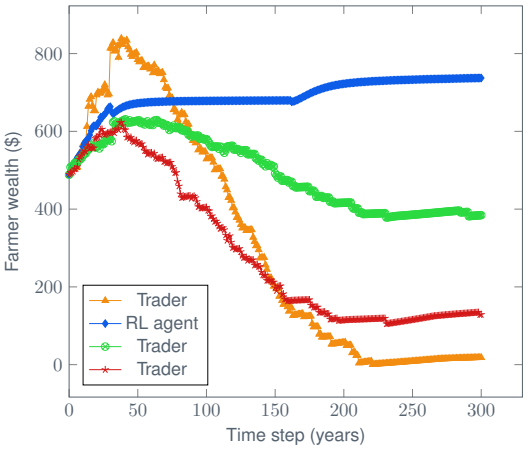
Reinforcement Learning versions

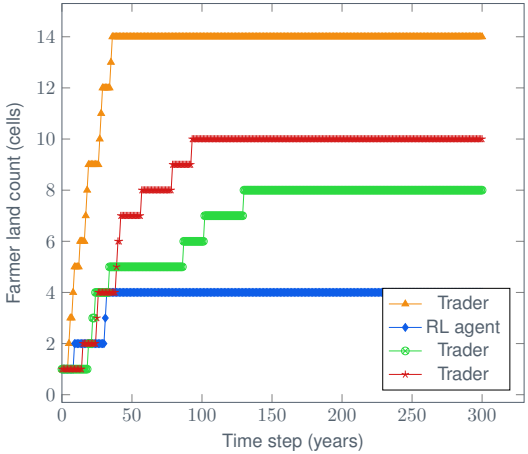
ML agent trained against traders

v2.1: RL agent trained againt Trader farmers

Reinforcement Learning versions

ML agent trained against traders





Reinforcement Learning versions

ML agent trained against traders

Figure: Stock evolution for the three `Trader` agents and the RL agent.

Crop War in perspective

Outlook and Perspective