A Simulator-based Planning Framework for Optimizing Autonomous Greenhouse Control Strategy

Zhicheng An*[1], Xiaoyan Cao*[2], Yao Yao*[1], Wanpeng Zhang[3], Lanqing Li[4], Yue Wang[3], Shihui Guo[2] and Dijun Luo[4]

[1]Tsinghua-Berkeley Shenzhen Institute, Tsinghua University

[2]School of Informatics, Xiamen University

[3]Tsinghua University [4]Tencent AI Lab

Abstract

The rapidly growing global population presents challenges and demands for efficient production of healthy fresh food. Autonomous greenhouses equipped with standard sensors and actuators contribute to producing higher yields. However, they require skilled and expensive labor, as well as a large amount of energy. An autonomous greenhouse control strategy, powered by AI algorithms by optimizing the yields and resource use simultaneously, offers an ideal solution to the dilemma. In this paper, we propose a two-stage planning framework to automatically optimize greenhouse control problem. First, we take advantage of cumulative planting data and horticulture knowledge to build a multi-modular simulator. Second, two AI algorithms are applied as planning methods to obtain optimal control strategies based on the simulator. We evaluate our framework on a cherry tomato planting dataset and demonstrate how the simulator is able to simulate greenhouse planting processes with high accuracy and fast speed. Moreover, the control strategies produced by the AI algorithms all obtain superhuman performance in terms of net profits.

Problem Statement

In this work, we formulate the greenhouse control problem as a deterministic Markov decision process (MDP) problem.

The MDP contains four components:

- State space: Variables across outside weather, indoor climate, crop state and production. Each variable is a real value with different units which encode different information.
- Action space: We focus on the actions, i.e., temperature, CO_2 concentration, lighting, and irrigation as the four most important variables.
- Reward function: We use net profit as reward, calculated by subtracting costs from gains.
- Transition function: No explicit expression.

Optimization objective:

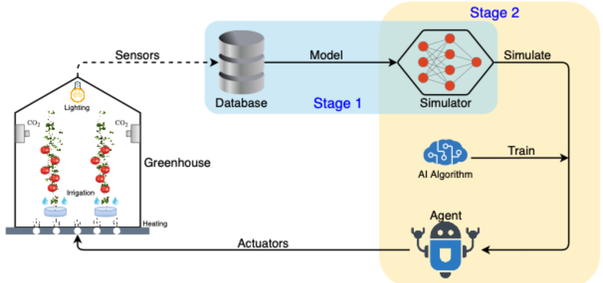
$$\underset{\pi}{\operatorname{argmax}} \sum_{t=0}^{T-1} \gamma^t r_t$$

$$\text{s.t.} \begin{cases} \gamma = 1, & \\ a_t = \pi(s_t) & (i = 0, \cdots, T-1), \\ s_{t+1} = \mathcal{P}(s_t, a_t) & (i = 0, \cdots, T-1), \\ r_t = \mathcal{R}(s_t, a_t, s_{t+1}) & (i = 0, \cdots, T-1), \end{cases}$$

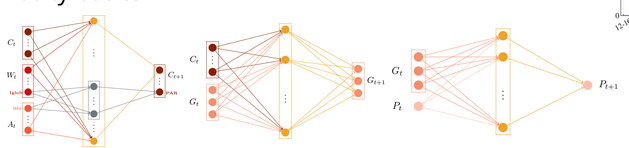
s and a represent state variables and action variables. $\mathcal{P}$ is the transition function. $\mathcal{R}$ is reward function.

Methodology

A general two-stage framework to optimize the autonomous greenhouse control strategy. In the first stage, we use the collected planting data in the real greenhouse as input to train a simulator that can approximate the real state transition function $\mathcal{P}$. In the second stage, two AI algorithms are applied to obtain the suboptimal policy in the simulator.



Simulator as an approximation: we build a simulator based on neural networks to approximate and fit the transition function. The state transition function is divided into three modules trained in a data-driven manner, including greenhouse climate module, crop growth module and production module. The first two modules are calculated on an hourly basis, while the production module is calculated on a daily basis.



Optimization algorithms: we use two typical AI algorithms to solve the MDP.

- Soft Actor-Critic: A state-of-the-art off-policy reinforcement-learning algorithm.
- Elitist Genetic Algorithm: A classic heuristic algorithm that can accelerate convergence in our scenario to obtain the suboptimal strategy.

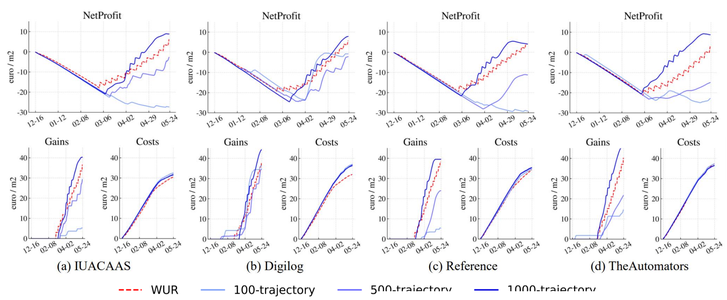
Experiments

Dataset: We evaluate our framework on a tomato dataset collected from a reliable simulator built by Wageningen University & Research (WUR).

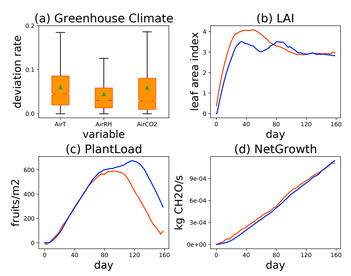
Simulator analysis: We trained three simulators based on 100, 500 and 1000 planting trajectories and one simulator without prior horticultural knowledge. The simulator based on 1000 trajectories has the best simulation accuracy as the below table shows.

Simulator	AirT	AirRH	$AirCO_2$	PAR	LAI	PlantLoad	NetGrowth	FW
no_prior	-72.320	-96.200	-437.526	-157.749	-4.903	-1920.908	-0.540	-16.455
100-trajectory	0.841	0.612	0.872	0.934	-0.135	0.383	0.227	0.278
500-trajectory	0.872	0.625	0.906	0.935	0.214	0.460	0.685	0.554
1000-trajectory	**0.961**	**0.851**	**0.966**	**0.935**	**0.630**	**0.829**	**0.959**	**0.952**

Economic variable analysis: We use five real planting records for the simulation on WUR and our simulators respectively. The result below shows that only the simulator trained on 1000 trajectories could simulate the harvest situation and costs accurately as well as WUR's, but yield is still higher in the second half.

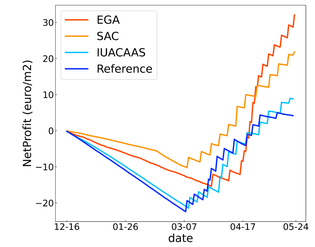


The simulator based on 1000 trajectories and WUR's simulator has similar predictions for important intermediate variables throughout the planting process as the right figure shows.



Performance Comparison: we compare four strategies in our simulator.

- IUACAAS and Reference policies: A research group's policy and human-expert policy.
- EGA: policy generated by elitist genetic algorithm.
- SAC: policy generated by soft actor-critic method.



Method	HeatCost	CO_2Cost	ElecCost	LaborCost
IUACAAS	6.550	**1.902**	17.714	5.271
Reference	8.018	2.531	19.524	5.271
SAC	**5.526**	2.296	**4.100**	5.271
EGA	6.230	2.123	12.302	5.271

The resulting figure and the table show that the two AI algorithms we apply greatly exceed the IUACAAS and Reference on net profit. EGA has the best net profit gains, while SAC has the lowest heat cost and electricity cost.

Conclusion

In this paper, we propose a two-stage planning framework to optimize the control strategy for autonomous greenhouse planting. First, we model the autonomous greenhouse control as a deterministic MDP problem and point out that the state transition function $\mathcal{P}$ is the key challenge. Then we build a multi-modular simulator based on neural networks in a data-driven style to approximates $\mathcal{P}$ and verify its accuracy. Then we apply two AI algorithms to obtain suboptimal strategies which outperform comparing policy.