Controle de temperatura em uma estufa agrícola utilizando algoritmo de aprendizado por reforço

Allan de Vasconcelos Ferreira Souza
Universidade Federal de Viçosa
Rodolpho Vilela Alves Neves
Universidade Federal de Viçosa

***Resumo*—The increase in agricultural production, the search for more sustainable methods with a conscious consumption of natural resources and the growing demand for organic food, have led to an increase in the use of greenhouses in agricultural production. Therefore, automated control systems have been used to manipulate the micro climatic conditions of greenhouses and are being improved with new monitoring, modeling and control technologies. This paper intend to apply a Reinforcement Learning model to control the internal temperature of a greenhouse, training an agent that will be capable to learn the best way to use the actuators, in order to reduce the energy consumption on operation and keep the temperature under control.**

I. Introdução

Uma casa de vegetação é uma estrutura agrícola que protege o cultivo de plantas dos fatores ambientais que podem prejudicar o desenvolvimento do cultivo, como o excesso de chuvas, calor e pragas. Tais estruturas podem ser de caráter parcial ou pleno, onde o parcial funciona como um guarda-chuva sobre o plantio enquanto o pleno permite obter uma maior proteção através das cortinas laterais da estrutura. Esta proteção promove o efeito estufa, como são mais conhecidas estas estruturas, que permite um maior controle do ambiente interno [1].

Essas estruturas criam um microclima na área do plantio e permitem um maior controle nas variáveis que afetam a produção. Portanto, diversos métodos de automação são empregados para o controle das condições climáticas, com o objetivo de atingir a melhor condição para determinada plantação. Essas condições são alteradas através do controle dos atuadores que afetam os parâmetros, por exemplo, de luminosidade, umidade do ar e taxa de ventilação internas à estufa.

Diversos modelos de inteligência artificial (IA) vem sendo empregados na automatização dos atuadores para controlar o ambiente produtivo. Por exemplo, conforme apresentado em [2], modelos de Redes Neurais Artificiais e *Support Vector Machines* são bastante empregados no gerenciamento da plantação, do solo e da água utilizada na irrigação.

Para reforçar a importância da inteligência artificial na automação, ocorreu um campeonato [5] entre times internacionais para o controle de uma estufa autônoma para produção de horticultura. Os participantes implementaram modelos de AI no controle de estufas idênticas para a produção de pepino, e foram avaliados em quesitos como produtividade, facilidade na implementação dos modelos e utilização racional dos recursos.

Outro modelo que vem sendo utilizado é o Aprendizado por Reforço, tendo bons resultados no gerenciamento dos atuadores na estufa. Como no trabalho de Byunghyun e Soobin [3], onde os autores aplicaram um modelo conhecido como ator-critico. Sendo o ator um modelo responsável pela distribuição de probabilidade sobre as ações que o agente pode escolher em um determinado estado. A parte do crítico é representada pela estimativa do retorno esperado ao agente, baseada em suas escolhas seguindo a política definida pelo ator.

O artigo de Tchamitchian et al. [4], aplicou um modelo menos complexo baseado apenas em um algoritmo para a estimativa do valor médio recebido pelo agente em cada intervalo de tempo da simulação. Tal modelo capta menos as mudanças do ambiente como no ator-crítico, porém seu treinamento é mais rápido. Ambos trabalhos obtiveram êxito no controle da temperatura do ambiente, porém utilizaram um modelo da estufa muito simplificado em sua simulação, com poucos parâmetros de dimensão da estrutura, material de cobertura, tipo de estrutura do teto e desenvolvimento da plantação.

Este trabalho se propõem a aplicar um modelo de Aprendizado por Reforço conhecido como *Deep Q-Learning Network* (DQN) no controle da temperatura interna de uma estufa para horticultura. Tal modelo permite o treinamento de um agente para manipular os atuadores utilizando-se dos parâmetros das condições climáticas do local através da simulação do ambiente produtivo.

Contudo, será utilizado um modelo mais completo do ambiente da estufa do que os trabalhos previamente citados, com a possibilidade de adaptação de seus parâmetros baseado nas condições locais de operação do empreendimento. Além disso, será proposto uma forma de operação para a estufa, levando em conta a leitura dos sensores, o processamento e armazenamento dos dados para retreinar o modelo conforme necessário e os equipamentos utilizados para prototipar a operação.

II. Aprendizado por Reforço

O Aprendizado por Reforço pode ser visto como uma terceira classe de aprendizado em Machine Learning, ao lado de aprendizado supervisionado e não-supervisionado. O foco deste aprendizado é treinar um agente capaz de interagir com o ambiente de treinamento, através da realização de ações que permitam transitar entre diferentes estados do sistema [6].

A cada instante de tempo, é gerada uma determinada recompensa R_t que sinaliza ao agente a eficácia da ação escolhida no estado ao qual se encontrava. O intuito deste aprendizado é maximizar as recompensas obtidas em cada transição do sistema, escolhendo as melhores ações para cada estado. Naturalmente, conforme aumenta a complexidade do problema, ações realizadas em um instante de tempo podem impactar recompensas futuras, levando o agente a escolher entre tentar novas ações e explorar outras já executadas anteriormente.

Para contabilizar recompensas futuras de uma determinada ação em um estado é utilizado uma *value function*. Esta função servirá para estimar recompensas futuras para as ações disponíveis em um estado, permitindo a maximização da recompensa total ao longo do processo, e não apenas no estado atual que se encontra o agente.

A *policy* é um outro elemento importante em Reinforcement Learning, ela é responsável por mapear os estados a probabilidade de escolher cada ação disponível. Denotada pela função $\pi(a|s)$, onde $s \in S$ é o conjunto de estados e $a \in A(s)$ é o conjunto de ações disponíveis em um determinado estado. Os algoritmos *on-policy* atualizam a estimativa de recompensas futuras escolhendo as ações seguindo uma única *policy*, enquanto um algoritmo *off-policy* utiliza uma *target policy* que será treinada enquanto uma outra *behaviour policy* gera as ações durante o treinamento, normalmente de forma aleatória para explorar o espaço de estados existente.

Entre diversos métodos de Aprendizado por Reforço, como Monte Carlo e Programação Dinâmica, um método que ganhou bastante evidência é o *Temporal-Difference Learning*, justamente por utilizar características de ambos métodos em suas previsões. Este método foi apresentado em 1988 por Sutton [12], onde uma de suas características principais é o fato de sua previsão ser incremental, não sendo necessário chegar ao passo final para atualizar os pesos ocorridos durante a fase de treinamento da previsão. Este fato permite uma convergência mais rápida. A seguir é apresentado mais informações sobre este método.

A. Temporal-Difference Learning

Também conhecido como *TD Learning*, este método permite o aprendizado diretamente das ações do agente sob o ambiente, sem a necessidade de um modelo que explicita todas as probabilidades de transição de um estado para o outro. Além disso, não é necessário terminar um episódio de treinamento para atualizar as estimativas das recompensas, sendo esta renovada a cada mudança de estado.

A *value function* básica de um TD Learning, dado uma *policy*, pode ser descrita como:

$$V(S_t) \approx V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \quad (1)$$

Onde a *value function* do estado S no instante t é atualizada com a recompensa R observada no próximo instante, corrigida pela diferença entre a *value function* de S_{t+1} com a do estado atual previamente visitada. Esta atualização ainda contém dois hiperparâmetros, sendo α a taxa de aprendizagem e γ a taxa de desconto que controla o impacto de recompensas futuras no estado atual.

B. Deep Q-Learning Network

O *Q-Learning* é um algoritmo de *TD-Learning* que não necessita de um modelo de transição e é *off-policy*. Este utiliza uma *action-value function*, $Q(s, a)$, onde o valor futuro das recompensas é estimado não apenas para o estado em si, mas também para cada ação em um determinado estado. Sendo a sua função apresentada como:

$$Q(s,a) \approx Q(s,a) + \alpha[R_{t+1} + \gamma \max_a Q(s',a) - Q(s,a)] \quad (2)$$

Quando um problema possui muitos estados a serem visitados, torna-se inviável um agente transitar por todos eles. Com isto, são utilizados aproximadores para as *action-value functions*, $Q(s, a, \theta) \approx Q^*(s, a)$, onde θ representa um vetor de pesos a serem estimados no modelo. O *Q-Learning* é instável a aproximadores lineares [7], sendo utilizado redes neurais profundas como aproximadores não-lineares [8].

Este tipo de rede é empregado para estimar o *Q-value* do próximo estado para todas as possíveis ações, obtendo uma aproximação futura das recompensas. Assim, pode ser escolhida a ação com maior retorno para um *Q-value* alvo [9]. A rede neural pode ser treinada minimizando a seguinte função custo, que é alterada a cada iteração:

$$L_t(\theta_t) = \mathbb{E}_{\rho(a|s)}[(y_t - Q(s,a,\theta))^2] \quad (3)$$

Onde $y_t = \mathbb{E}_{s' \sim \mathcal{S}}[r + \gamma \max_{a'} Q(s', a', \theta_{t-1})]$ é o alvo de cada iteração t, r é a recompensa obtida na iteração e $\rho(a|s)$ é a distribuição de probabilidade conhecida como distribuição do comportamento, referente as escolhas das ações dado um estado. Normalmente, utiliza-se uma estratégia ϵ-*greedy*, onde é selecionada a ação com melhor *Q-value* com probabilidade 1-ϵ ou uma ação aleatória com probabilidade ϵ.

Ao otimizar a função custo, os parâmetros θ_{t-1} da iteração anterior são mantidos fixos. Esta otimização é obtida através do cálculo do gradiente estocástico descendente da função custo,$\nabla_\theta L_t(\theta_t)$, conforme a seguinte equação:

$$\nabla_\theta L_t(\theta_t) = [r + \gamma \max_{a'} Q(s', a', \theta_{t-1}) - Q(s, a, \theta_i)] \nabla_{\theta_i} Q(s, a, \theta_i) \quad (4)$$

III. Modelo Estufa

Um agente precisa de um ambiente para interagir e conseguir aprender quais ações levam a melhores recompensas em determinados estados. Logo, é necessário construir um modelo de simulação de uma estufa que consiga refletir a dinâmica climática interna em resposta as alterações que o agente irá provocar ao acionar os atuadores necessários para o controle da temperatura interna da estufa.

O modelo escolhido [10] utiliza uma equação diferencial ordinária para calcular a variação da temperatura interna em relação ao tempo ($\frac{dT_{in}}{dt}$). Também é levado em consideração as dimensões da estufa e o material da cobertura do teto, permitindo adequá-lo a diferentes tipos de estruturas. O modelo é apresentado a seguir e os parâmetros e variáveis relacionadas estão listadas na tabela 1.

$$\frac{1}{C_p.\rho.H}(Q_{GRin} - L.E - (T_{in} - T_{out})(q_v.C_p.\rho + w.k)) \quad (5)$$

Tabela I
PARÂMETROS DO MODELO

Símbolo	Valor	Unidade	Descrição
C_p	1010	$J\,kg^{-1}K^{-1}$	Calor específico do ar úmido
ρ	1.2	$kg_{arseco}\,m^{-1}$	Massa específica do ar
L	2.5E6	$J\,kg^{-1}$	Calor latente do vapor de água
H	6.3	m	Altura da estufa
w	2.3	$-$	Razão entre a área da cobertura e a área do solo
k	6.2	$J\,m^{-2}\,°C^{-1}h^{-1}$	Coeficiente de transmissão de calor da cobertura
τ	0.87	-	Transmitância do material da cobertura
ρ_g	0.5	-	Reflexão da radiação solar no solo
T_{in}		$°C$	Temperatura interna do ar
T_{out}		$°C$	Temperatura externa do ar
Q_{GRin}		$W\,m^{-2}$	Radiação solar absorvida dentro da estufa
E		$kg\,m^{-2}s^{-1}$	Taxa de evapotranspiração dentro da estufa
E_C		$kg\,m^{-2}s^{-1}$	Taxa de evapotranspiração do sistema de refrigeração
E_S		$kg\,m^{-2}s^{-1}$	Taxa de evapotranspiração dos aspersores
E_T		$kg\,m^{-2}s^{-1}$	Taxa de evapotranspiração da plantação interna
N		h^{-1}	Troca de ar por hora
q_v		$m\,s^{-1}$	Taxa de ventilação

Os parâmetros da tabela 1 foram extraídos de [4], onde as primeiras linhas representam constantes previamente definidas pelo ambiente e pelo tipo de estufa utilizada, como a sua altura e material da cobertura. As demais linhas são calculadas da seguinte forma:

$$Q_{GRin} = \tau(1 - \rho_g) \quad (6)$$

$$E = E_C + E_S + E_T \quad (7)$$

$$q_v = (volume * N)/(3600 * largura * comp.) \quad (8)$$

As variáveis E_C e E_S dependem diretamente da condição de operação dos atuadores do sistema de refrigeração evaporativo e dos aspersores, respectivamente. A forma de operação dos atuadores é resultado do treinamento do agente no controle do ambiente, que será apresentado mais adiante.

Para calcular a evapotranspiração da plantação interna à estufa, diversos modelos propõem utilizar uma gama de parâmetros como tamanho da folha, condições do vento e incidência de radiação solar. Conforme mostrado em [4], a irradiação solar é o fator que possui maior correlação com a transpiração da plantação e, para simplificar o cálculo deste parâmetro do modelo, foi proposta uma fórmula linear em relação a radiação solar absorvida dentro da estufa. Essa taxa de evapotranspiração da plantação é apresentada como:

$$E_T = 10^{-3}(0.3 \cdot \tau \cdot Q_{GRout} + 2.1) \quad (9)$$

A figura 1 ilustra o balanço de energia do modelo da estufa. Para regiões frias, poderia ser incluido um aquecedor na modelagem e na operação, porém, este não é necessário para o local escolhido para a operação.

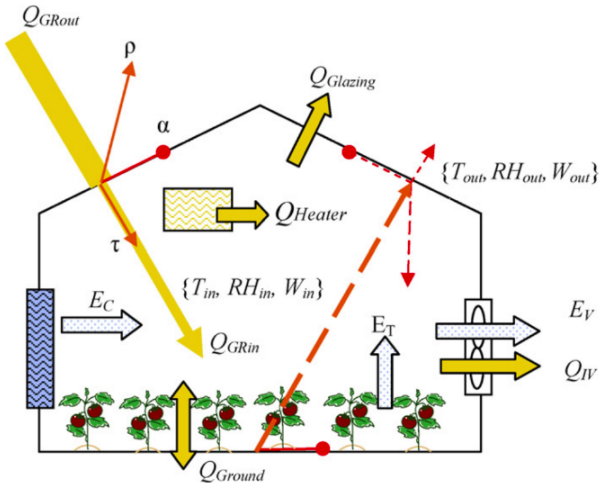

Figura 1. Balanço energético interno [10]. Parâmetros listados na tabela 1.

A temperatura externa é obtida diretamente dos dados utilizados para a simulação. Neste caso, foi utilizado uma estação meteorológica na região de Campos dos Goytacazes, ao norte do estado do Rio de Janeiro [11], no ano de 2020. A temperatura interna é calulada através dos incrementos obtidos do modelo mencionado.

Para entender como é a condição climática do local, foi realizado uma análise exploratória dos dados da estação meteorológica. Esta análise separou os dias em grupos semelhantes utilizando o algoritmo *Kmeans*, onde o valor ideal foi de 4 grupos. Para a separação, foram utilizados a temperatura média, radiação média, umidade relativa média e velocidade do vento média, sendo todas as variáveis separadas entre os períodos da manhã e tarde. A figura 2 ilustra a variação da temperatura ao longo do ano de 2020.

Conforme apresentado na figura 3, pode-se notar que o grupo 0 apresenta a maior temperatura média tanto no período da manhã quanto da tarde, abrangendo as temperaturas mais elevadas. Este agrupamento obteve os dias com menor recompensa devido ao maior consumo energético no controle da temperatura.

IV. CONDIÇÕES DE OPERAÇÃO

Como a região de operação da estufa possui uma temperatura elevada, os atuadores utilizados visam o controle da temperatura interna para que, em dias de alta concentração de calor, não ultrapassem o limite estabelecido de 35 ºC. Cada instrumento de atuação afeta diretamente uma variável do modelo, por exemplo, um exaustor de ar aumenta a taxa de ventilação q_v, diminuindo a concentração de calor na parte interna da estufa.

Figura 2. Temperaturas máxima, média e mínima ao longo do ano.

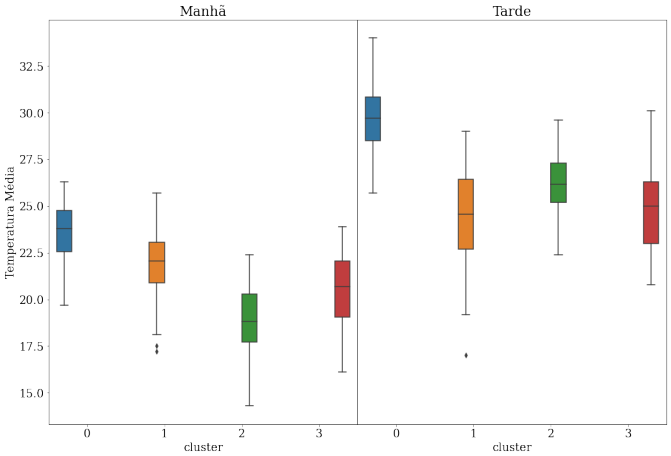

Figura 3. Temperatura média por cluster

Além disso, cada atuador consome uma determinada quantidade de energia medida em kWh dividida pelo tempo de operaçao do atuador. Este tempo no modelo foi definido em 10 minutos, ou seja, a cada intervalo de tempo o agente irá tomar uma ação de configuração de acionamento dos atuadores. Com isto, o objetivo principal do agente será escolher a melhor configuração dos atuadores sem haver gasto desnecessário com energia e mantendo o controle interno da temperatura.

Tabela II
ATUADORES DA ESTUFA

Atuador	**Variável Afetada**	**Consumo(kWh)**
Exaustor 01	q_v, $N = 30$	0,123
Exaustor 02	q_v, $N = 30$	0,123
Cooler	E_C	0,3
Tela sombrite	Q_{GRin}	0,0001
Aspersor	E_S	0,05

Um outro ponto importante é a janela de operação da automação. Como a temperatura crítica ocorre durante o dia, o treinamento e monitoramento ocorrerá das 6 horas às 20 horas. Isto diminui a quantidade de dados para treinamento do agente sem prejudicar o controle do processo.

V. TREINAMENTO DO AGENTE

Uma parte crucial em modelos de Reinforcement Learning é a definição da estratégia de recompensas ao agente. Tal pontuação deve deixar claro o objetivo principal a ser alcançado e quais perdas devem ser evitadas.

Após a definição das recompensas, o agente pode ser treinado utilizando o algoritmo *Deep Q-Learning* [8]. Este agente, basicamente, é uma Rede Neural treinada para aprender, através das condições climáticas do modelo da estufa, como melhor operar os atuadores para o controle da temperatura interna. O agente não possui nenhuma informação a priori sobre como operar os atuadores, apenas recebe as condições climáticas e as recompensas baseadas em suas ações.

A. Recompensa

Foi definido que, a cada rodada que a temperatura fica abaixo do limiar e nenhum atuador é ativado, o agente recebe 3 pontos de recompensa. Caso a temperatura continue abaixo do estipulado mas a energia gasta pelos atuadores seja menor do que 70% da capacidade total, o agente recebe 2 pontos. Caso ultrapasse os 70%, este recebe apenas 1 ponto.

A maior penalização ocorre quando a temperatura interna ultrapassa o *set point*, sendo descontado 10 pontos da recompensa ao agente. Porém, em dias muito quentes, isto provocava uma desistência do agente em não deixar ultrapassar o limite da temperatura. Isto ocorria pois o mesmo evitava acionar os atuadores para não receber uma penalização devido ao gasto de energia. Assim, foi adicionado uma penalização menor, de apenas 5 pontos ao invés dos 10 pontos, caso a variação da temperatura seja negativa em cada ciclo, ou seja, a temperatura interna está em declínio.

B. Treinamento

No início do treinamento é definido quantos episódios serão rodados, onde cada episódio corresponde a um dia escolhido aleatoriamente da base de dados da estação meteorológica. Assim, o ambiente da estufa é inicializado com os dados do dia escolhido a partir das 6 horas da manhã até às 20 horas, período de atividade da automação. Cada rodada ocorre num intervalo de 10 minutos dentro do dia escolhido, ou seja, o agente irá treinar neste episódio com as condições climáticas deste dia.

O processo de treinamento ocorre usando uma estratégia ϵ-*greedy*, onde a probabilidade de escolher uma ação aleatória é equivalente ao valor de ϵ. Enquanto há probabilidade de $1 - \epsilon$ de seguir a distribuição de comportamento onde é escolhida a ação com maior *Q-Value*, sendo assim usado uma Rede Neural para estimar estes valores.

VI. IMPLANTAÇÃO

Para a implantação do sistema de automação da estufa, foi utilizado um Raspberry Pi 4 que armazena os dados coletados do ambiente e avalia quais atuadores devem ser ativados baseados nos pesos treinados pela Rede Neural. Um Arduino Uno é empregado para a coleta dos dados através de sensores externos e internos à estufa e os envia, através de uma rede

Algoritmo 1 Treinamento do agente

$n\_episodio = 5000$
while $episodio \neq n\_episodio$ **do**
  Inicializar: Ambiente de simulação
  $env = Ambiente()$
  while $rodada \neq ultima\_rodada$ **do**
    $\epsilon = max(1 - episodio/n\_episodio, 0.01)$
    $estado = env.get\_estado\_atual()$
    $atuadores = agente.jogar\_uma\_rodada(estado, \epsilon)$
    {**Ativar Atuadores**}
    $env.set\_atuadores(atuadores)$
    {**Calcular Recompensa**}
    $recompensa = env.calcular\_recompensa()$
    {**Próxima Rodada**}
    $env.proximo\_estado()$
  end while
end while

local Wi-fi, utilizando o protocolo HTTP com uma API que recebe as requisições no Raspberry Pi. Um módulo ESP8266 é utilizado para realizar a conexão do Arduino à rede Wi-fi e enviar suas requisições.

O Arduino verifica a cada 10 minutos de operação qual a próxima configuração gerada pelo agente através de uma requsição GET enviada ao Raspberry. Como resposta, recebe um arquivo no formato JSON contendo a ativação de cada relé respectivo a cada atuador.

Como o Raspberry PI possui uma capacidade de processamento limitada para a resolução de modelos como Redes Neurais, apesar de estar avançando a cada nova versão, é interessante utilizar uma máquina virtual remota para o treinamento inicial do modelo. A fase inicial de otimização dos pesos da rede é mais intensiva em processamento, após isto, as decisões podem ser processadas diretamento no Raspberry PI. Outro ponto importante de utilizar este serviço externo é a capacidade de armazenamento dos dados gerados, como histórico de utilização dos atuadores e dados meteorológicos do local, que servem para auditoria e monitoramento remoto do sistema.

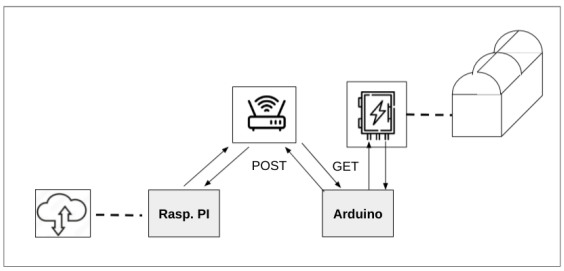

Figura 4. Esquema de operação da estufa

VII. Análise e discussão dos resultados

O primeiro gráfico da figura 5 confronta a temperatura interna e externa à estufa, onde esta é controlada perto dos 35 ºC pré-estabelecido. O gráfico do meio apresenta a recompensa recebida pelo agente em cada momento da simulação. Nota-se que, quando a temperatura interna chega perto do limiar, a recompensa cai pois o agente precisa ativar os atuadores para não extrapolar o limite. Além disso, os atuadores são ativados momentos antes de chegar ao limite para evitar que haja um descontrole em sua aproximação. O último esquema mostra a configuração dos atuadores escolhidos pelo agente em determinado período de tempo, havendo alguns momentos intermitentes de ativação para reduzir ao máximo o consumo da temperatura.

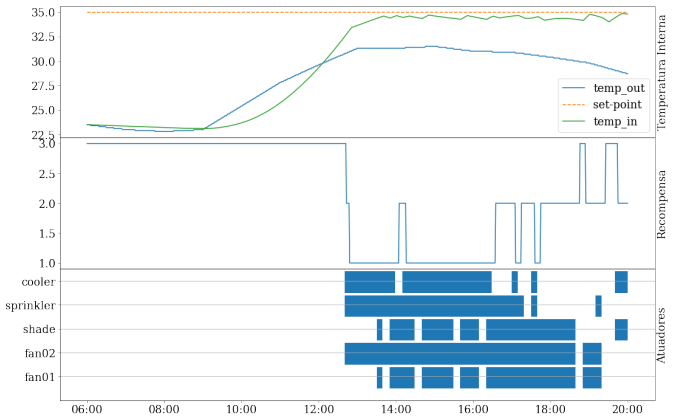

Figura 5. Temperatura, recompensa e configuração dos atuadores com *set-point* em 35 ºC.

A figura 6 apresenta o comportamento do agente quando o *set-point* é reduzido para 33 ºC, mantendo-se o mesmo dia para critério de comparação. O agente conseguiu manter o controle da temperatura com a redução do limite estabelecido, porém foi necessário aumentar o tempo de ativação dos atuadores e utilizar a capacidade máxima em momentos que a temperatura ficou na faixa dos 33 ºC.

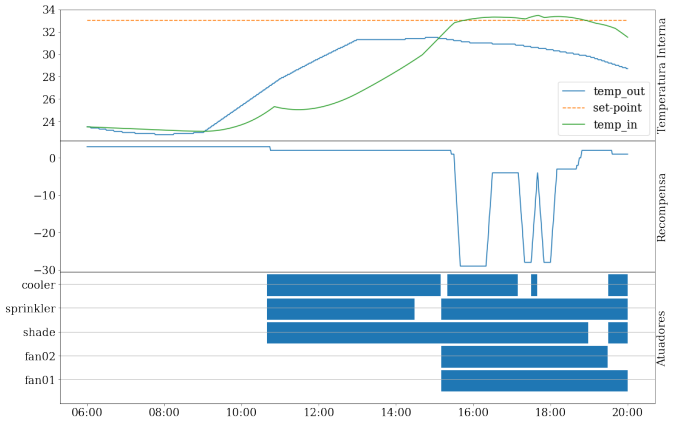

Figura 6. Temperatura, recompensa e configuração dos atuadores com *set-point* em 33 ºC.

Outro ponto importante observado foi a antecipação na utilização de alguns atuadores para segurar ao máximo a elevação da temperatura interna com o aumento externo da temperatura. A combinação escolhida dos atuadores permitiu ficar abaixo de 70% do consumo energético total, o que não levou a penalização extra por ultrapassar o limite de consumo.

Quando a temperatura externa ultrapassa o limite estabelecido e os atuadores não são suficientes para controlar a temperatura, o agente escolhe em não os ativar e logo a temperatura extrapola o *set-point*. Então, só após um determinado tempo, os atuadores são ativados para voltar a faixa aceitável de operação. Os gráficos da figura 7 ilustram tal comportamento do agente e a figura 8 comprova que, mesmo com todos os atuadores ativados, a temperatura iria extrapolar o *set-point* desejado.

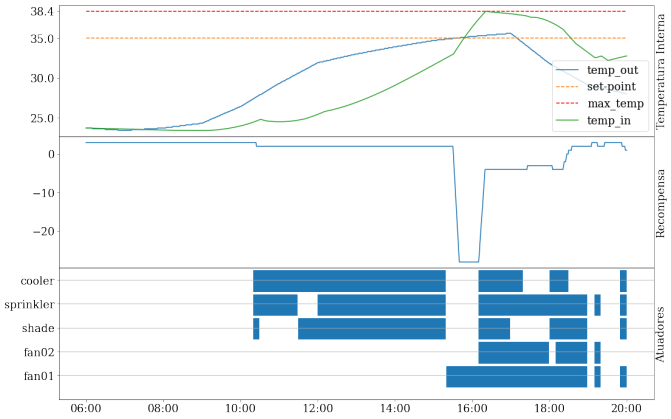

Figura 7. Temperatura, recompensa e configuração dos atuadores com *set-point* em 35 ºC extrapolado.

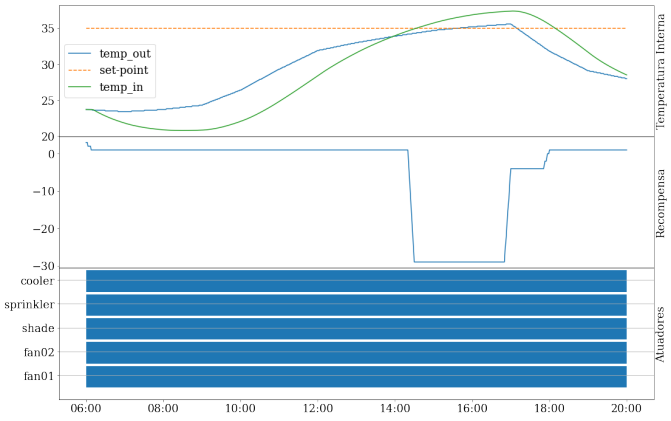

Figura 8. Temperatura, recompensa e configuração dos atuadores com *set-point* em 35 ºC com ativação total dos atuadores.

Este comportamento pode ser corrigido ao adionar mais atuadores para amenizar a elevação da temperatura, por exemplo inserindo mais um *cooler*. Entretanto, os dias que aconteceram essa extrapolação representam apenas 6% dos dias de operação em um ano, ou seja, o novo atuador ficaria ocioso a maior parte do tempo.

Ao final, foi realizado um teste com um controle do tipo *on-off*. Este controlador foi configurado para ativar todos os atuadores quando a temperatura chega à 95% da temperatura do *set-point* (35 ºC) sendo desligado caso a temperatura volte a ficar abaixo deste ponto. Este controlador alcançou 183 pontos de recompensa contra 205 do agente.

A figura 9 mostra a configuração adotada por este método que pode ser comparado com a do agente na figura 5, ambos para o mesmo *set-point* e dia de operação. Ficando evidenciado que a melhor utilização dos atuadores, e não a completa ativação de todos os atuadores, resulta em um acréscimo na recompensa através de um melhor consumo de energia por parte do agente, sem perder o controle da temperatura do ambiente.

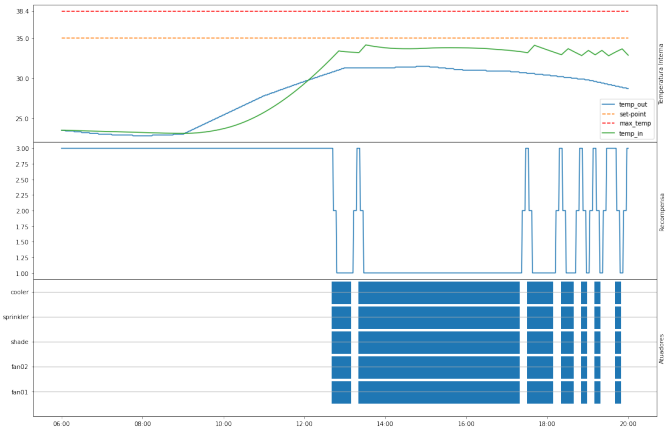

Figura 9. Temperatura, recompensa e configuração dos atuadores com *set-point* em 35 ºC com controlador *on-off*.

VIII. Conclusão

O principal objetivo do trabalho foi treinar um agente para controlar os atuadores em uma estufa de plantas, a fim de controlar a temperatura interna da mesma, utilizando Aprendizado por Reforço. Um ambiente simulado mais elaborado do que os trabalhos anteriores permite uma adequar melhor a localidade e ao tipo de estrutura utilizado no local.

O agente conseguiu controlar a temperatura da estufa em aproximadamente todos os dias do ano com exceção de 15 dias. Estes foram os de calor muito intenso, acima de 36 ºC, com baixa ventilação e alta irradiação solar. Mesmo com a plena utilização dos atuadores da estufa, a temperatura não se manteve abaixo do *set-point* de 35 ºC.

Situação mais comum foram de dias quentes, perto dos 31 ºC, onde o agente conseguiu controlar a temperatura perto do *set-point* sem utilizar energia excessiva para este controle. A figura 5 ilustra esta situação comumente ocorrida.

O comportamento do agente de relaxar o limite do *set-point* é um ponto interessante para uma análise futura. Ver o quão tolerável o produto é a esses dias mais quentes e se seria financeiramente viável produzir nos dias mais quente, dado o acréscimo de um atuador que ficaria ocioso a maior parte do ano. O período com esses dias caíram entre os meses de dezembro à fevereiro, onde talvez o aumento do preço final do produto possa viabilizar um aumento na capacidade de resfriamento da estufa.

Outra possível melhoria em estudos futuros seria a utilização de novas variáveis de entrada para a rede neural no treinamento do agente, como a previsão do tempo para o dia. O objetivo principal seria tentar diminuir o tempo de treinamento do agente ao promover mais informaçõe que podem ser úteis na identificação das condições que o agente irá enfrentar durante o dia de operação.

REFERÊNCIAS

[1] REIS, N. Construção de estufas para produção de hortaliças nas regiões Norte, Nordeste e Centro-Oeste. Circulação técnica, Brasília-DF. 2005. Disponível em: www.embrapa.br/documents/1355126/9124396/Constru%C3%A7%C3%A3o+de+estufas.pdf. Acesso em: nov/2021.

[2] LIAKOS, K. G. et al. Machine Learning in Agriculture: A Review, Sensors 18, no. 8: 2674, 2018.

[3] BYUNGHYUN. B,; SOOBIN, K. Control of nonlinear, complex and black-boxed greenhouse system with reinforcement learning, International Conference on Information and Communication Technology Convergence, IEEE, 2017

[4] TCHAMITCHIAN. M. et al. Daily temprature optimisation in greenhouse by reinforcement learning. IFAC, 16th Triennial World Congress, Elsevier, 2005

[5] HEMMING, S. et al. Remote Control of Greenhouse Vegetable Production with Artificial Intelligence—Greenhouse Climate, Irrigation, and Crop Production, MDPI, 2019

[6] SUTTON, R.S.; BARTO, A.G. Reinforcement Learning: An Introduction, 2.ed, The MIT Press, 2018.

[7] MAEI, H.R. et al. Toward Off-Policy Learning Control with Function Approximation, Proceedings of the 27th International Conference on Machine Learning (ICML-10), Omnipress, 2010.

[8] MNIH,V. et al. Playing Atari with Deep Reinforcement Learning

[9] GÉRON, A. Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow, 2.ed, O'Reilly Media, 2019

[10] FITZ-RODRÍGUEZ, E. et al. Dynamic modeling and simulation of greenhouse environments under several scenarios: A web-based application, Computers and Eletronics in Agriculture, Elsevier, p.105-116, 2010

[11] DADOS HISTÓRICOS ANUAIS, Instituto Nacional de Meteorologia, 2021. Disponível em: https://portal.inmet.gov.br/dadoshistoricos. Acesso em: 05 de fev. de 2021.

[12] SUTTON, R.S. Learning to predict by the methods of temporal differences, Machine Learning 3,9-44, Springer, 1988