|  |  |  |
| --- | --- | --- |
| **Seção** | **Resumo** | **Observações** |
| 1. Introduction | -Há, no momento, muitas propostas de CC, utilizando RL  -Vantagem sobre os algoritmos sob medida (bespoke)  - Adaptabilidade  -Apesar disso, ainda em estágio inicial e requer  -Decisão sobre qual model  -Treinamento (Quais estados? Qual a recompensa? Como limitar os parâmetros?)  -Aderência ao mundo real  -Vantagens na utilização de simuladores de rede (domínio da situação, custo dos experimentos na vida real, repodutibilidade)  -Daí RayNet seria um training playground  -RayNet - Primeiro a integrar o RayLib com o OMNeT.  -A comunicação direta, proposta pelo RayNet, o coloca entre os mais eficientes | Mostrando o espaço a ser ocupado pelo RayNet. Bem delineado! |
| 2 BACKGROUND | 2.1 Reinforcement Learning  -Maximizar a recompensa, mapeando Estados em ações  -Mapear Estados em ações a fim de que uma funçãio objetivo seja maximizada numa trjetória S0A0R1S1A1R2....StAtRt+1....  -Obtenção de trajetórias pode ser um problema    2.2 Controle de Congestionamento    -OK    2.3 OMNet++ Simulator    - ns3 like    2.4 Ray an RayLib    -Destaca a atribuição de políticas específicas para cada agente  -Possibilidade de vetorização para múltiplos agentes | Seção Teórica, bem interessante!  Sobre a seção 2.4  Uma pequena sugestão seria introduzir o conceito de episódio em 2.1 e relacionar com a fila de eventos do OMNet, em 2.3  Explicar melhor a colocação  “*Instead of having the agent actively step the environment and wait for the returned tuple, Rllib provides the ExternalEnv interface, which permits querying a policy for actions and logging end-of-step tuples*” linhas 48 e 49, página 6 |
| 3 DESIGN PRINCIPLES | Destaca as principais características da plataforma  -Separação entre o ambiente de simulação e o aprendizado  -Permite que um agente seja trienado em múltiplos setups  -Permite estabelecer granularidades distintas, o que permite a diversificação da natureza do treinamento.  -Múltiplos Agentes  -Reprodutibilidade  -Eficiência e Escalabilidade | Melhorar a figura 3 com contrastes e cores;  Como há foco no desacoplamento, seria mais interessante descrever a arquitetura com base em algum “Design pattern”. |
| 4.RayNet Architeture | 4.1 Overview  -Trainer e worker - Ray Process  -Dentro do environment fica a simulação (OMNet)  -A vetorização é se houver mais de um worker.  -O "trainer" delega políticas de avaliação (o q deve ser maximizado??) aos workers...  -O worker interage com o environment (OMNeT++) via API, por meio dos métodos initialise(), step(), reset()  -Initialise() - Inicia o simulador OMNeT  -Reset - Leva o OMNeT a um estado inicial aleatório, início de um episódio  -step(action) - The worker passa a ação como parâmetro (action) e o environment retorna o novo estado, o valor da recompensa, e um booleano indicando se é ou não o passo final    4.2 Event looping and Envioronment Step  -Um step pode ser composto por um ou mais eventos de simulação    4.3 RayNet Environment  -environment = RL agents + sptepper and Broker  -Stepper - se encarrega de ajustar o estado inicial dos agentes.  -Broker repassa as ações para os agentes e colhe as observações e recompesnas para o worker.  -A comunicação entre os módulos menscionados é via signal/subscribe  -Resumindo, o worker chama o initialise, que faz toda a integração sgnal/slot entre o Stepper, o borker e os agentes. Após o initialise, o worker chama o reset para estabelecer o estado inicial, que é alcançado  pela execução de um ou mais eventos OMNeT. | Trazer a nota de rodapé 7 e 8 para as respectivas páginas (Linha 11 e Linha 47 pg 9)  Indicar o Pybind na figura 2, ou em um possível diagrama de classe.  Levar a ideia “the step does not need to be defined by the trainer or worker(s)” (linha 40 pg 10) para dentro de (2)  A seção 4.3 requer as correções mais críticas:  - os componentes *RL agents, the Stepper, Broker, OMNeT++ API, OMNeT++ modules* não são indicados na figura, o que dificultou muito  - Há uma “sobrecarga” do termo step (função, conjunto de eventos, marca de reset e marca de fim de fim de evento). Procurar designar melhor. |
| 5 LEARNING CONGESTION CONTROL WITH RAYNET | -Descrição dos pressupostos para treinamento do protocolo  -Recompensa baseada em throughput, delay e perdas, conforme equação | Bem teórico, uma vez que o foco é demonstrar as potencialidades da plataforma |
| 6 EXPERIMENTING WITH RAYNET | -O objetivo é demonstrar a capacidade de segregação entre simulador e componentes de aprendizado e, com isso, a riqueza de experimentos possíveis  **-Varying environment parameters**  -Demostrou que funciona pela apresentação dos gráficos  -A possibilidade de variação de parâmetros é o foco. A análise dos resultados é secundária.  -O que foi trinado se sai bem no intervalo de trinamento!    **-Verifing learning parameters and hyper-parameters**  -tratando da questão do tuning dos parâmetros na arquitetura, que é facilitada pela compartimentalização  -Demonstra, pelos gráficos, que é possível variar a política de aprendizado.    **-Efficience and Scalability**  -Demonstrar que a plataforma e eficiente, comparando-a com o Open AI Gym  -Ficaram praticamente empatados em recompensa média por tempo de treinamento, CPU e Memória. | Requer maiores detalhes de reprodutibilidade  As possibilidades são comprovadas pela exibição de gráficos, tendo e, vista destacar as potencialidades da plataforma.  Como é dito no trabalho a análise profunda dos resultados apresentados pelo trabalho foge do seu escopo.  Há uma referência perdida na linha 17 da página 16  Nomear a figura 17 para 16, para ser coerente com a ordem do texto. |