

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

ĐỖ VŨ GIA CĂN

KHÓA LUẬN TỐT NGHIỆP

**TÌM HIỂU CÁC PHƯƠNG PHÁP HỌC TĂNG CƯỜNG
CHO BÀI TOÁN ĐIỀU KHIỂN TÍN HIỆU GIAO THÔNG**

TỰ ĐỘNG

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, 2022

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

ĐỖ VŨ GIA CĂN

KHÓA LUẬN TỐT NGHIỆP

**TÌM HIỂU CÁC PHƯƠNG PHÁP HỌC TĂNG CƯỜNG
CHO BÀI TOÁN ĐIỀU KHIỂN TÍN HIỆU GIAO THÔNG
TỰ ĐỘNG**

CỦ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN

TS. LƯƠNG NGỌC HOÀNG

TP. HỒ CHÍ MINH, 2022

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số
ngày của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

1. – Chủ tịch.
2. – Thư ký.
3. – Ủy viên.
4. – Ủy viên.

LỜI CẢM ƠN

Lời đầu tiên, tôi xin được gởi một lời cảm ơn sâu sắc đến thầy Lương Ngọc Hoàng vì đã tận tình giúp đỡ, động viên, định hướng cho tôi trong những ngày đầu và xuyên suốt quá trình nghiên cứu và hoàn thiện đề tài khóa luận. Nếu không có sự hướng dẫn từ thầy, mục tiêu của khóa luận sẽ không thể hoàn thành.

Tiếp theo, tôi xin được gởi lời cảm ơn đến quý thầy cô giảng viên trong trường Đại học Công Nghệ Thông Tin nói chung và khoa Khoa Học Máy Tính nói riêng vì đã tận tình giảng dạy và giúp tôi có được nhiều kiến thức chuyên môn để làm hành trang cho việc hoàn thành đề tài nghiên cứu này.

Bên cạnh đó, tôi cũng không quên gởi lời cảm ơn đến bạn Nguyễn Trọng Thoại, người bạn đồng hành của tôi đã giúp đỡ và hỗ trợ tôi trong những ngày đầu mới bước vào nghiên cứu đề tài.

Và cuối cùng, tôi muốn bày tỏ lòng biết ơn to lớn của mình đối với những thành viên trong gia đình tôi vì họ là luôn là điểm tựa vững chắc trong suốt những ngày tháng sinh viên của tôi, và là nguồn động lực to lớn để thôi thúc tôi hoàn thành tốt đề tài này.

Mục lục

TÓM TẮT KHOÁ LUẬN

xii

1 TỔNG QUAN	1
1.1 Đặt vấn đề	1
1.2 Bài toán điều khiển tín hiệu đèn giao thông	3
1.2.1 Phát biểu bài toán	3
1.2.2 Thách thức	3
1.2.3 Hướng tiếp cận	4
1.3 Mục tiêu của khóa luận	6
1.4 Đối tượng và phạm vi nghiên cứu	7
1.4.1 Đối tượng	7
1.4.2 Phạm vi nghiên cứu	7
1.5 Nội dung thực hiện	7
1.6 Cấu trúc khóa luận	8
2 CÁC CÔNG TRÌNH LIÊN QUAN VÀ CƠ SỞ LÝ THUYẾT	9
2.1 Các công trình liên quan	9
2.1.1 Thuật toán ITSC	9
2.1.2 Mô hình FRAP	11
2.1.3 Các công trình thực hiện những bộ đánh giá khác	12
2.2 Cơ sở lý thuyết	13
2.2.1 Giới thiệu về Học tăng cường	13
2.2.2 Mô hình hóa bài toán Điều khiển tín hiệu giao thông	17
2.2.3 Điều khiển đa tác nhân	19
3 CÁC BỘ ĐIỀU KHIỂN ĐỀ XUẤT CHO BÀI TOÁN ĐIỀU KHIỂN TÍN HIỆU ĐÈN GIAO THÔNG	24

3.1	Bộ điều khiển cơ bản	24
3.2	Bộ điều khiển dựa trên Học tăng cường	26
3.2.1	Deep Q-Network (DQN)	26
3.2.2	Double Deep Q-Network (Double DQN)	28
3.2.3	Proximal Policy Optimization (PPO)	29
3.2.4	MPLight	31
3.2.4.1	Khái niệm Pressure	31
3.2.4.2	Mô hình FRAP	32
3.2.5	Extended MPLight	36
4	THỰC NGHIỆM	37
4.1	Bộ mô phỏng được sử dụng	37
4.2	Các chỉ số đánh giá	39
4.3	Các thiết lập thực nghiệm	40
4.4	Kết quả thực nghiệm	42
5	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	49
5.1	Kết luận	49
5.2	Hướng phát triển	50

Danh sách hình vẽ

1.1	Tình trạng ùn tắc giao thông tại các giao lộ	2
1.2	Các hướng tiếp cận cho bài toán Điều khiển tín hiệu đèn giao thông. Dấu mũi tên màu đỏ mô tả hướng tiếp cận chính mà chúng tôi nghiên cứu trong luận văn này.	4
1.3	Minh họa bài toán Điều khiển tín hiệu đèn giao thông dựa theo hướng tiếp cận dựa trên học tập	6
2.1	Sơ đồ minh họa thuật toán ITSC	10
2.2	Minh họa các hướng di chuyển tại một ngã tư. Các dấu mũi tên chỉ ra các hướng di chuyển bị ràng buộc bởi tín hiệu giao thông	11
2.3	Bộ mô phỏng Manhattan và bản đồ có 4 giao lộ trong bộ mô phỏng AIM	12
2.4	Minh họa một bước thời gian của bài toán Học tăng cường	14
2.5	Các thuật toán Học tăng cường được phân loại dựa trên tính chất của không gian trạng thái và tập hành động. Dấu mũi tên màu đỏ thể hiện thuật toán này được dựa trên thuật toán trước đó.	16
2.6	Minh họa 8 hướng di chuyển phụ thuộc vào tín hiệu đèn tại một ngã tư và 8 hướng chuyển động giao thông tương ứng.	18
2.7	Vấn đề điều khiển đa tác nhân đối với môi trường có nhiều giao lộ.	20
2.8	Minh họa mô hình đa tác nhân độc lập. Mũi tên màu đỏ thể hiện tác nhân thực hiện hành động lên môi trường và mũi tên màu xanh lá thể hiện trải nghiệm mà môi trường trả về cho tác nhân khi sau khi thực hiện hành động.	21
2.9	Minh họa mô hình đa tác nhân kết hợp. Mũi tên màu đỏ thể hiện tác nhân thực hiện hành động lên môi trường và mũi tên màu xanh lá thể hiện điểm thưởng môi trường trả về cho tác nhân khi sau khi thực hiện hành động.	22

3.1	Minh họa quy trình thiết kế chiến lược thủ công.	25
3.2	DQN sử dụng kiến trúc mạng AlexNet	27
3.3	Minh họa những làn đường đi vào và làn đường ra tại một giao lộ. Làn màu tím thể hiện cho làn đường đi vào và làn màu xanh thể hiện cho làn đường ra	32
3.4	Thiết kế của mô hình FRAP	33
4.1	Mô phỏng bản đồ của hai thành phố Ingolstadt và Cologne dựa trên 3 cấp độ. Những vùng màu xanh đánh dấu các giao lộ.	38
4.2	Mô phỏng bản đồ được thiết kế theo dạng lưới. Những vùng màu xanh đánh dấu các giao lộ.	38
4.3	Minh họa các giao lộ với cấu trúc khác nhau.	39
4.4	Chỉ số queue length trung bình sau 10 random seeds trên 8 bản đồ khác nhau. Trục x: thể hiện số lần huấn luyện trong 1 seeds. Các đường nét đứt là đường trung bình được lấy trên 100 lần huấn luyện.	44
4.5	Chỉ số delays trung bình sau 10 random seeds trên 8 bản đồ khác nhau. Trục x: thể hiện số lần huấn luyện trong 1 seeds.Các đường nét đứt là đường trung bình được lấy trên 100 lần huấn luyện.	45
4.6	Chỉ số duration trung bình sau 10 random seeds trên 8 bản đồ khác nhau. Trục x: thể hiện số lần huấn luyện trong 1 seeds.Các đường nét đứt là đường trung bình được lấy trên 100 lần huấn luyện.	46
4.7	Chỉ số waiting time trung bình sau 10 random seeds trên 8 bản đồ khác nhau. Trục x: thể hiện số lần huấn luyện trong 1 seeds.Các đường nét đứt là đường trung bình được lấy trên 100 lần huấn luyện.	47

Danh sách bảng

4.1	Bảng các siêu tham số của hai thuật toán DQN và DoubleDQN	41
4.2	Bảng các siêu tham số của hai thuật toán PPO	41
4.3	Bảng hiệu suất tốt nhất của các thuật toán Học tăng cường. MP-Light* là ký hiệu của Extended MPLight.	48

Danh sách thuật toán

1	Thuật toán DQN	28
2	Thuật toán DoubleDQN	29

Danh mục từ viết tắt

RL	Reinforcement Learning
SUMO	Simulation of Urban Mobility
SCATS	Sydney Coordinated Adaptive Traffic System
SCOOT	Split Cycle Offset Optimisation Technique
RHODES	Real-time Hierarchical Optimized Distributed Effective System
DQN	Deep Q - Learning
A2C	Advantage Actor Critic
PPO	Proximal Policy Optimization
ITSC	Intelligent Traffic Signal Control
FRAP	Flipping Rotation All Phases
AIM	Autonomous Intersection Management
DNN	Deep Neural Network
CNN	Convolution Neural Network
MDP	Markov Decision Process
MARL	Multi Agent Reinforcement Learning
PG	Policy Gradient
CPI	Conservative Policy Iteration
MPLight	Max Pressure Light
TraCI	Traffic Control Interface

TÓM TẮT KHOÁ LUẬN

Ngày nay, sự gia tăng dân số tại đã dẫn đến mật độ giao thông ngày càng cao tại các thành phố lớn và tình trạng ùn tắc là điều khó tránh khỏi, kéo theo nhiều hệ lụy như làm ô nhiễm môi trường và không khí khi khói bụi trên đường phố ngày càng nhiều. Điều này đã tạo nên những thách thức trong việc tìm ra những giải pháp tối ưu giao thông để làm ổn định và cân bằng cuộc sống trong các đô thị. Một trong những giải pháp có thể được xem xét tới là việc điều khiển và phối hợp các tín hiệu giao thông tại các giao lộ một cách hợp lý để phù hợp với lưu lượng xe trên từng làn đường, tránh gây ùn tắc tại giao lộ. Theo những nghiên cứu của chúng tôi về các phương pháp trước đây, có hai hệ thống đó là hệ thống điều khiển hẹn giờ trước (Pre-timed) và hệ thống điều khiển kích hoạt (Actuated), tuy nhiên cả hai hệ thống này chủ yếu dựa trên một mô hình giao thông nhất định hoặc trên các quy tắc giao thông đã được xác định trước, vì vậy khó để cung cấp những giải pháp tối ưu để điều chỉnh sao cho phù hợp với lưu lượng xe đang liên tục gia tăng.

Trong những năm gần đây, lĩnh vực học tăng cường (RL - Reinforcement Learning) đang cho thấy nhiều ứng dụng của nó trong việc đưa ra các giải pháp giải quyết các tác vụ phức tạp trong thực tế. Vì vậy, trong khóa luận tốt nghiệp này, chúng tôi xin trình bày những nghiên cứu về cách áp dụng các thuật toán RL vào bài toán điều khiển tín hiệu giao thông tự động để tìm ra các chiến lược tối ưu phù hợp với các tình huống giao thông thực tế đồng thời vẫn tuân thủ các nguyên tắc trong giao thông, trong đó đầu vào sẽ là trạng thái của lưu lượng giao thông hiện tại, lựa chọn việc thực hiện giữ tín hiệu đèn hoặc chuyển sang tín hiệu khác phụ thuộc kết quả hàm điểm thưởng thiết kế cho mỗi trạng thái riêng biệt và xuất ra trạng thái ở giai đoạn tiếp theo. Sau khi tìm hiểu được cách áp dụng, chúng tôi sẽ tiến hành so sánh độ hiệu quả giữa các thuật toán trong mỗi tình huống giao thông riêng biệt dựa trên các tiêu chí đánh giá như tổng thời gian chờ trung bình của các xe tại các giao lộ (waiting time), trung bình tổng số xe chờ trên làn đường (queue length) và trung bình tổng độ trễ (delays). Bên cạnh đó, chúng tôi còn tiến hành thiết kế các chiến lược thời gian cố định (fixed-time) cho từng tình huống giao thông để so sánh xem liệu các thuật toán RL có hoạt động tốt hơn

hay các chiến lược này hay không. Tất cả sẽ được trình bày ở phần thực nghiệm và kết quả.

Cuối cùng, để so sánh và đánh giá các kết quả, chúng tôi tiến hành thực nghiệm dựa trên bộ dữ liệu mô phỏng giao thông SUMO (Simulation of Urban Mobility).

Chương 1

TỔNG QUAN

Trong chương này, chúng tôi sẽ giới thiệu tổng quan về bài toán điều khiển tín hiệu giao thông tự động, những thách thức gặp phải và các hướng tiếp cận đã có trước đó đối với bài toán này. Tiếp theo, chúng tôi sẽ tóm tắt về đối tượng và phạm vi cũng như mục tiêu nghiên cứu trong khóa luận này. Ở cuối chương, chúng tôi sẽ trình bày về những nội dung đã thực hiện và bối cảnh chính của khóa luận.

1.1 Đặt vấn đề

Sự gia tăng dân số không ngừng qua mỗi năm tại các trung tâm kinh tế trọng điểm của đất nước hay cụ thể hơn là các thành phố và các khu đô thị lớn đã gây ra rất nhiều hệ lụy xấu đến nhiều mặt của đời sống, xã hội, trong đó lĩnh vực giao thông cũng không tránh khỏi những ảnh hưởng. Thực tế, hiện trạng ùn tắc giao thông và kẹt xe đang diễn ra tràn lan trên các con đường lớn tại các thành phố, đặc biệt là trong những giờ cao điểm. Điều này không chỉ gây nên những cảm giác khó chịu cho người tham gia giao thông, tạo ra cho họ sự căng thẳng và về lâu dài là những ảnh hưởng nghiêm trọng về mặt sức khỏe của những người dân sống trên thành phố mà còn đặt ra rất nhiều những tiêu cực lên môi trường bởi khói bụi, ô nhiễm tiếng ồn hay sự gia tăng khí thải từ phương tiện giao thông vào môi trường, v.v. cùng với đó là rất nhiều vấn đề to lớn ảnh hưởng đến kinh tế. Vì thế, việc tìm ra những giải pháp để góp phần giảm thiểu tắc nghẽn giao thông là rất cần thiết.

Một trong những giải pháp được những người làm việc trong lĩnh vực giao thông hướng đến đó là điều hòa lưu lượng xe tại các giao lộ lớn, những nơi được

xem là nút thắt giao thông và có nhiều xe cộ qua lại hằng ngày, bởi thực tế, việc làn đường hẹp, cộng thêm việc nhiều phương tiện không được lưu thông do phải chờ đèn đỏ quá lâu sẽ dẫn đến việc ùn tắc hàng dài tại những khu vực đó.



HÌNH 1.1: Tình trạng ùn tắc giao thông tại các giao lộ¹

Theo những quan sát của chúng tôi tại các giao lộ vào những giờ cao điểm, thường sẽ có cảnh sát giao thông tham gia điều phối lưu lượng xe. Giải pháp này tuy có thể giảm thiểu được phần nào ùn tắc, nhưng lại gây nên sự tốn kém về mặt nhân lực và chi phí để thực hiện. Do đó, các nhà nghiên cứu hướng đến việc tìm ra các chiến lược để điều khiển tín hiệu đèn tự động, sao cho thích ứng với các tình huống giao thông thực tế.

Trong những năm gần đây, các thuật toán Học tăng cường (RL) đã được các nhà nghiên cứu quan tâm nhiều hơn với mong muốn tạo ra những ứng dụng hay những hệ thống vận hành tự động mà không cần sự tương tác với con người. Mặc dù những thuật toán này trước đó được sử dụng chủ yếu trong các trò chơi điện

¹Ảnh được lấy từ: bit.ly/3Ny0dYw

tử hay nổi bật nhất là cờ vây. Tuy nhiên, với những sự tiến bộ và cải tiến qua thời gian, DRL đang dần cho thấy sự đa dạng và mạnh mẽ của chúng trong việc giải quyết các tác vụ phức tạp trong thế giới thực như xe tự lái, dự đoán xu hướng kinh tế, sự biến động thị trường, v.v.

Nhờ tính hiệu quả của các thuật toán RL trong nhiều lĩnh vực khác nhau nên chúng tôi đã tiến hành tìm hiểu và nghiên cứu để áp dụng chúng vào việc tìm ra những chiến lược để tự động hóa việc điều khiển tín hiệu đèn giao thông tại các giao lộ.

1.2 Bài toán điều khiển tín hiệu đèn giao thông

1.2.1 Phát biểu bài toán

Về bản chất, điều khiển tín hiệu giao thông là việc lựa chọn giữ nguyên màu đèn tín hiệu hiện tại hoặc đổi màu đèn tín hiệu của mỗi hộp đèn (xanh lá, đỏ hoặc vàng) nhằm thay đổi trạng thái giao thông tại các giao lộ để các phương tiện di chuyển an toàn và hiệu quả qua những khu vực này và đồng thời vẫn bảo đảm đúng những nguyên tắc giao thông được quy định.

- Đầu vào: trạng thái của lưu lượng giao thông hiện tại
- Đầu ra: trạng thái của lưu lượng giao thông được cho phép tại giai đoạn tiếp theo

Trong khóa luận này, chúng tôi sẽ tìm cách áp dụng một số thuật toán Học tăng cường trong việc lựa chọn hành động, thiết kế nên những chiến lược điều khiển đèn tín hiệu để đáp ứng những nhu cầu giao thông thay đổi liên tục tại các giao lộ.

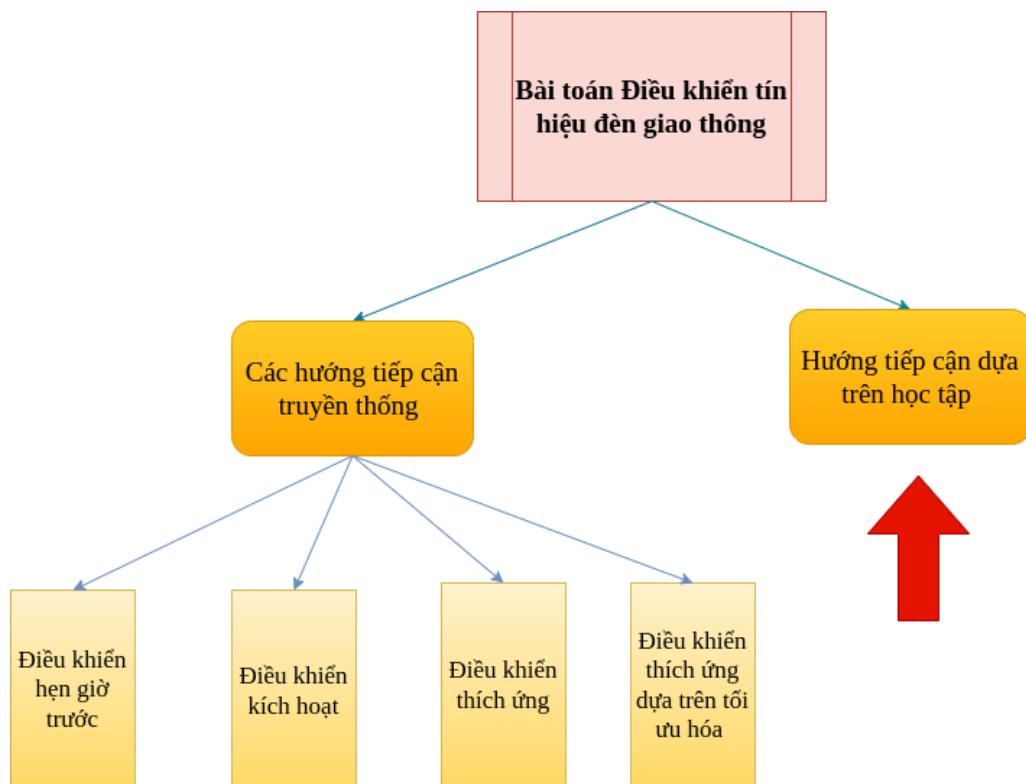
1.2.2 Thách thức

Theo như yêu cầu bài toán, có thể thấy rằng việc có được thông tin về trạng thái giao thông tại những giao lộ là rất quan trọng. Tuy nhiên chi phí lắp đặt các camera video hay các máy dò để theo dõi chuyển động của các phương tiện tại mỗi giao lộ là rất lớn.

Mặc dù khoa học công nghệ hiện nay đã phát triển với sự ra đời của các thiết bị GPS, các bộ cảm biến hay đặc biệt là công nghệ Big Data đã giúp cho việc nắm bắt và sử dụng dữ liệu trong giao thông trở nên hiệu quả và tiện lợi hơn, tuy nhiên việc áp dụng chúng trong một khu vực quy mô lớn cũng gây ra sự tốn kém rất lớn về mặt chi phí.

Do đó, với những người thực hiện nghiên cứu còn là sinh viên như chúng tôi, để thực nghiệm và kiểm tra hiệu quả trực tiếp vào một trường thực tế là rất khó khăn.

1.2.3 Hướng tiếp cận



HÌNH 1.2: Các hướng tiếp cận cho bài toán Điều khiển tín hiệu đèn giao thông. Dấu mũi tên màu đỏ mô tả hướng tiếp cận chính mà chúng tôi nghiên cứu trong luận văn này.

Để giải quyết tốt bài toán đặt ra thì việc tìm ra những chiến lược để điều khiển tín hiệu một cách hiệu quả là rất quan trọng. Hiện nay, có rất nhiều các phương

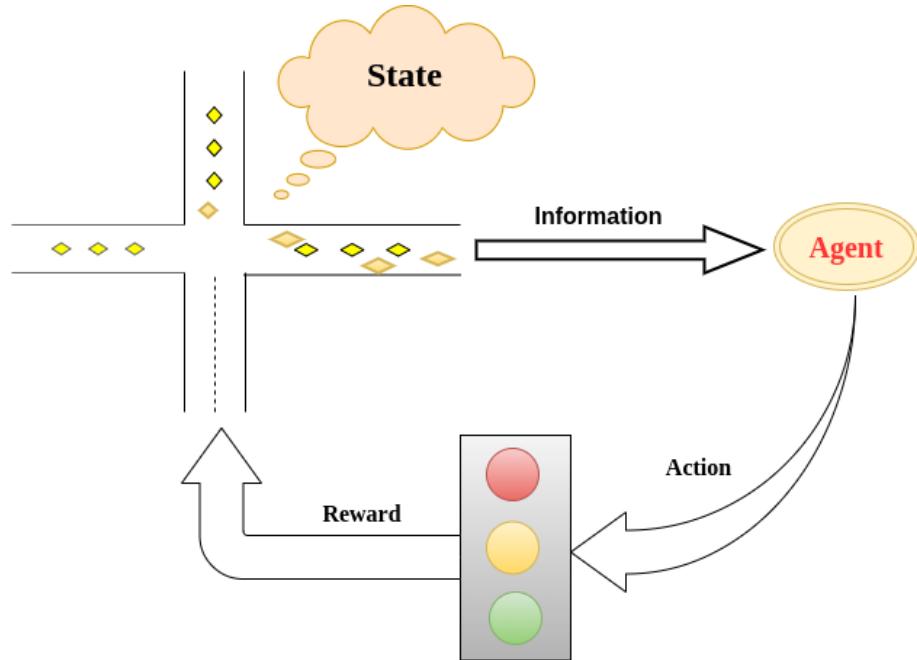
pháp để thiết kế các chiến lược, tuy nhiên đa phần chủ yếu dựa trên hai hướng tiếp cận chính: i) các phương pháp điều khiển tín hiệu truyền thống; ii) phương pháp điều khiển tín hiệu dựa trên học tập.

Các phương pháp truyền thống có thể được phân loại theo 4 hướng chính bao gồm:

- Điều khiển hẹn giờ trước (Pre-timed Control): sử dụng những quan sát trước đó của con người để đặt trước một khoảng gian cố định cho đèn xanh, đèn đỏ và đem vào thực hiện bất chấp lưu lượng giao thông thực tế.
- Điều khiển kích hoạt (Actuated Control): sử dụng những định nghĩa, những quy tắc giao thông được xác định trước đó để đưa ra các quyết định thay đổi thời gian của màu đèn.
- Điều khiển thích ứng (Adaptive Control): được sử dụng rộng rãi trong các hệ thống điều khiển đèn tín hiệu tại nhiều thành phố lớn hiện nay. Hướng tiếp cận này dựa trên các chiến lược được thiết kế thủ công và lựa chọn hành động sao cho trạng thái hiện tại được tối ưu nhất dựa trên lưu lượng giao thông nhận được từ vòng lặp cảm biến. Một số hệ thống sử dụng phổ biến chiến lược này có thể kể đến SCATS [16], SCOOT [11] hay RHODES. [14]
- Điều khiển thích ứng dựa trên tối ưu hóa (Optimized-based Adaptive Control): Hướng tiếp cận này thường dựa trên việc mô hình hóa bài toán điều khiển tín hiệu đèn giao thông thành bài toán tối ưu hóa. Vì thế, để cách tiếp cận này cho thấy sự hiệu quả thì cần phải có những giả định chắc chắn để xây dựng mô hình sao cho phù hợp các định nghĩa và quy tắc giao thông, điều này đôi khi dẫn đến việc khó áp dụng trong môi trường thực tế.

Điều khiển tín hiệu dựa trên học tập (Learning-based Signal Control): Cách tiếp cận này không dựa trên những định nghĩa được xác định trước trong giao thông, những kế hoạch được thiết kế thủ công hay các mô hình lưu lượng giao thông như những hướng tiếp cận truyền thống mà chủ yếu dựa trên các thuật toán để học trực tiếp từ các giao lộ, chủ yếu là các thuật toán học tăng cường. Cụ thể, hướng tiếp cận này được thực hiện như sau: mỗi giao lộ được xem như một tác nhân (agent), trạng thái (state) là những mô tả về tình trạng giao thông tại các giao lộ, hành động (action) là việc lựa chọn đèn tín hiệu cho mỗi giao lộ

và điểm thưởng (reward) dựa trên các chỉ số được dùng để đánh giá sự hiệu quả giao thông (ví dụ như độ trễ, độ dài hàng đợi tại các làn đường, thời gian chờ trung bình của tất cả các xe tại giao lộ v.v.)



HÌNH 1.3: Minh họa bài toán Điều khiển tín hiệu đèn giao thông dựa theo hướng tiếp cận dựa trên học tập

Trong khóa luận này, chúng tôi dự định sẽ trình bày những nghiên cứu sâu hơn về hướng tiếp cận này.

1.3 Mục tiêu của khóa luận

Trong khóa luận này, chúng tôi hướng đến việc hoàn thành hai mục tiêu chính bao gồm:

- Áp dụng các thuật toán Học tăng cường để giải quyết bài toán Điều khiển tín hiệu đèn giao thông dựa trên bộ mô phỏng SUMO - gồm những tình huống giao thông được lấy cảm hứng từ một số thành phố lớn trong thế giới thực.
- So sánh, đánh giá độ hiệu quả giữa các thuật toán và so với các chiến lược được thiết kế thủ công và các chiến lược ngẫu nhiên.

1.4 Đối tượng và phạm vi nghiên cứu

1.4.1 Đối tượng

Trong đề tài khóa luận này, chúng tôi tập trung nghiên cứu về một số thuật toán nổi bật trong lĩnh vực Học tăng cường như DQN, A2C, IPPO, v.v. và cách áp dụng chúng trong việc tìm ra những chiến lược để giải quyết vấn đề ùn tắc giao thông tại các giao lộ.

1.4.2 Phạm vi nghiên cứu

Về phạm vi nghiên cứu, chúng tôi sẽ xem xét và đánh giá độ hiệu quả của các thuật toán sau khi áp dụng chúng dựa trên một bộ mô phỏng các tình huống giao thông có một giao lộ, và những khu vực lớn có nhiều giao lộ, cùng với đó là những khu vực được quy hoạch theo kiểu mạng lưới và những con đường mà các giao lộ nằm trên một trực dọc.

Tất cả sẽ được trình bày kỹ hơn ở phần 4.1

1.5 Nội dung thực hiện

Nội dung mà chúng tôi thực hiện trong khóa luận này được trình bày như sau:

- Tìm hiểu về bài toán Điều khiển tín hiệu đèn giao thông và những hướng tiếp cận đã có trước đó để giải quyết bài toán.
- Tìm hiểu về các thuật toán học tăng cường và học hỏi cách thức nghiên cứu của tác giả để áp dụng vào việc giải quyết bài toán đặt ra.
- Tìm hiểu về các môi trường mô phỏng giao thông để có thể tiến hành thực nghiệm.
- Xây dựng bài toán Điều khiển tín hiệu đèn dựa trên bộ mô phỏng giao thông được đề xuất.
- Chạy thực nghiệm và đánh giá, so sánh độ hiệu quả giữa các thuật toán dựa trên bài toán đặt ra.

- Thiết kế các chiến lược cố định theo cách thủ công và các chiến lược ngẫu nhiên để cho thấy sự nổi bật của các thuật toán Học tăng cường.

1.6 Cấu trúc khóa luận

Khóa luận được chia thành 5 chương chính, cấu trúc được trình bày như sau.

- Chương 1: Trình bày tổng quan về bài toán Điều khiển tín hiệu đèn giao thông.
- Chương 2: Trình bày những nghiên cứu về các công trình liên quan và đưa ra các cơ sở lý thuyết.
- Chương 3: Trình bày chi tiết và cách thức hoạt động của các bộ điều khiển được sử dụng trong quá trình thực nghiệm.
- Chương 4: Trình bày chi tiết về bộ mô phỏng được sử dụng, các thiết lập thực nghiệm, kết quả thực nghiệm và đánh giá kết quả thu được.
- Chương 5: Rút ra kết luận và hướng phát triển trong tương lai.

Chương 2

CÁC CÔNG TRÌNH LIÊN QUAN VÀ CƠ SỞ LÝ THUYẾT

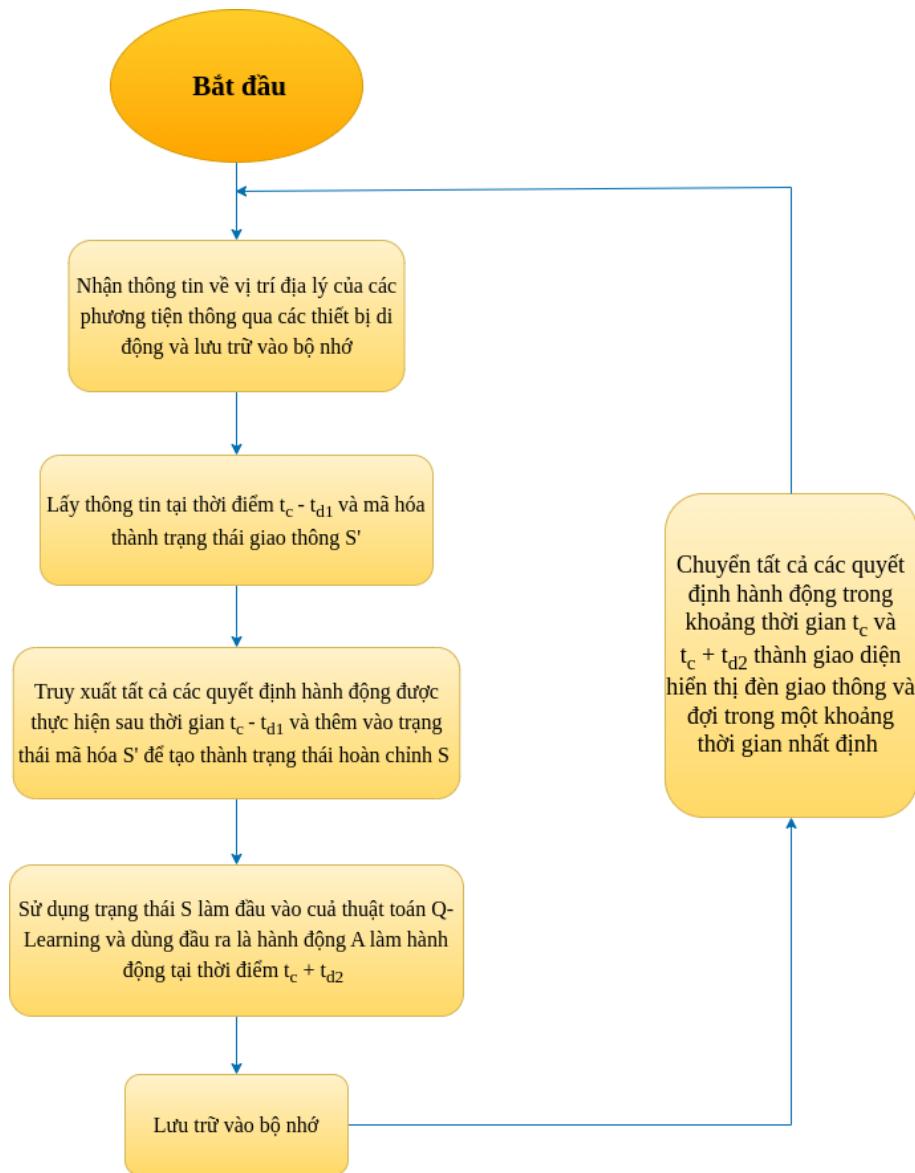
Trong chương này, chúng tôi sẽ trình bày một số công trình nghiên cứu liên quan việc giải quyết bài toán Điều khiển tín hiệu giao thông theo hướng tiếp cận dựa trên học tập, cùng với đó là tổng quan về các cơ sở lý thuyết làm nền tảng trong khóa luận này. Phần 2.1 trình bày những mô hình và thuật toán đã được áp dụng và thực nghiệm cho bài toán Điều khiển tín hiệu đèn giao thông cùng với các công trình nghiên cứu trên các bộ mô phỏng khác nhau. Phần 2.2 sẽ trình bày kiến thức về ý tưởng và những thành phần trong các thuật toán Học tăng cường cũng như phân loại các thuật toán. Bên cạnh đó, phần này cũng đề cập đến việc mô hình hóa bài toán Điều khiển tín hiệu đèn giao thông dưới dạng một quy trình quyết định Markov.

2.1 Các công trình liên quan

Trong nội dung này, chúng tôi trình bày một số công trình liên quan đến bài toán Điều khiển tín hiệu đèn theo hướng tiếp cận dựa trên học tập. Bên cạnh đó, chúng tôi cũng trình bày những công trình nghiên cứu liên quan được thực hiện trên những bộ mô phỏng khác nhau.

2.1.1 Thuật toán ITSC

¹Ảnh được lấy từ [20]

HÌNH 2.1: Sơ đồ minh họa thuật toán ITSC¹

ITSC là thuật toán được sử dụng trong các hệ thống điều khiển giao thông thông minh dựa trên điện toán đám mây hoặc các công nghệ yêu cầu tài nguyên tính toán từ xa. Trong đó, xe cộ hoặc các phương tiện giao thông sẽ gửi thông tin về vị trí địa lý của chúng lên các máy chủ đám mây thông qua các thiết bị di động hoặc các thiết bị theo dõi định kỳ. Máy chủ đám mây sẽ tiếp nhận và tổng hợp thông tin, sau đó đưa ra các quyết định về pha đèn giao thông cho giao lộ tương ứng bằng cách gởi quyết định tới tất cả các phương tiện có liên quan (hoặc trong

một trường hợp khác là gửi quyết định xuống các bộ điều khiển tín hiệu, nơi tiếp nhận những nhu cầu để thực hiện việc giữ hoặc chuyển pha tín hiệu).

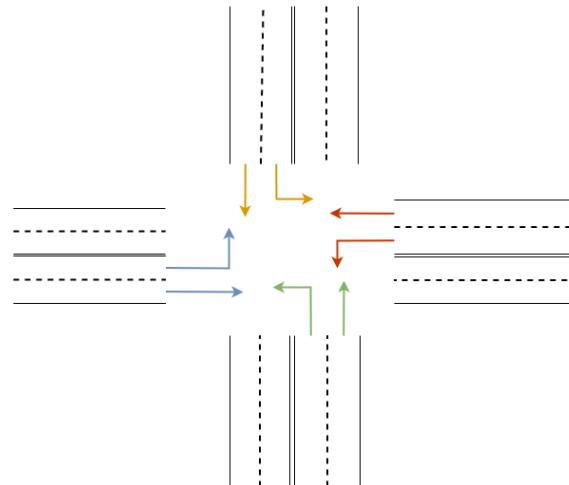
Tuy nhiên, các hệ thống này thường gặp một vấn đề lớn đó là độ trễ (latency). Nếu một trong hai bên là các phương tiện chậm trễ trong việc gửi thông tin lên các máy chủ hoặc các máy chủ bị trễ trong việc nhận thức được thực tế và đưa ra quyết định thì có thể dẫn đến các hậu quả hết sức nghiêm trọng.

Theo đó, các nhà nghiên cứu về các hệ thống điều khiển giao thông thông minh đã xem độ trễ là việc tất nhiên phải xảy ra và con người không thể kiểm soát chúng. Thuật toán ITSC được thiết kế để tìm ra giải pháp điều khiển giao thông tối ưu dựa trên hai tham số t_{d1} và t_{d2} .

Hình 2.3 minh họa thuật toán ITSC. Trong đó:

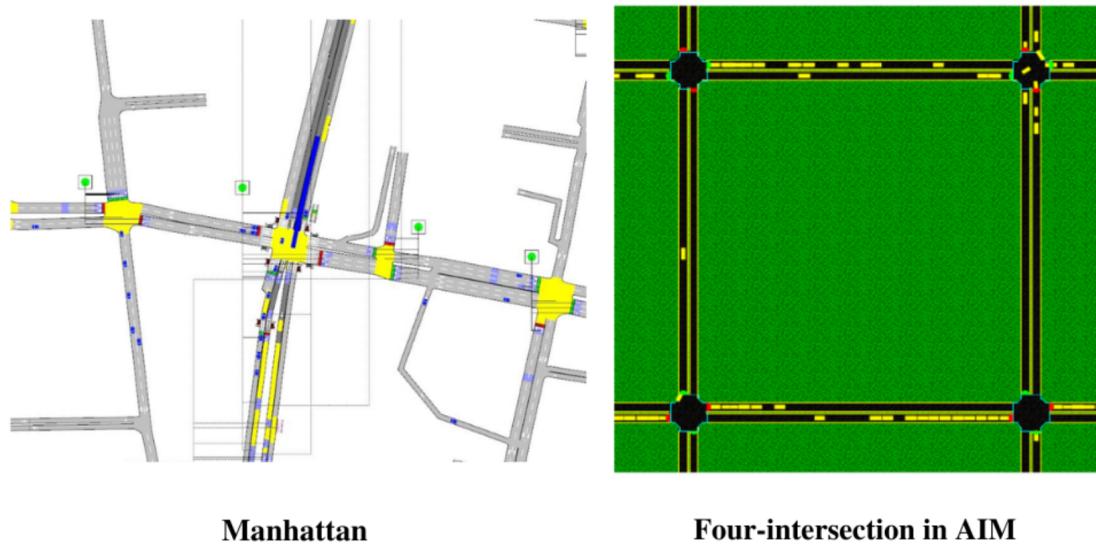
- t_c : thời điểm hiện tại
- t_{d1} : độ trễ của các phương tiện trong việc gửi thông tin
- t_{d2} : độ trễ của máy chủ đám mây trong việc đưa ra quyết định.

2.1.2 Mô hình FRAP



HÌNH 2.2: Minh họa các hướng di chuyển tại một ngã tư. Các dấu mũi tên chỉ ra các hướng di chuyển bị ràng buộc bởi tín hiệu giao thông

Một thách thức khác đặt ra đối với bài toán Điều khiển tín hiệu đèn giao thông đó là không gian tìm kiếm là rất lớn.



HÌNH 2.3: Bộ mô phỏng Manhattan và bản đồ có 4 giao lộ trong bộ mô phỏng AIM²

Ví dụ như tình huống giao thông tại giao lộ có 4 luồng giao thông (ngã tư) và 8 hướng di chuyển phụ thuộc vào tín hiệu được minh họa ở Hình 2.2. Nếu mỗi làn đường đều có n phương tiện thì kích thước không gian trạng thái qua 8 giai đoạn là $8xn^8$.

Vì thế để cắt giảm kích thước không gian trạng thái, Yaunhao Xiong và các cộng sự đã nghiên cứu và đề xuất mô hình FRAP [21], dựa trên những nguyên tắc về cạnh tranh giai đoạn trong điều khiển tín hiệu giao thông để đạt được sự bất biến trong các trường hợp lật và xoay trong lưu lượng giao thông.

2.1.3 Các công trình thực hiện những bộ đánh giá khác

Bên cạnh bộ mô phỏng giao thông SUMO mà chúng tôi sử dụng trong khóa luận này, đã có rất nhiều công trình nghiên cứu về bài toán Điều khiển tín hiệu đèn giao thông theo hướng tiếp cận dựa trên học tập được thực hiện trên những bộ mô phỏng khác.

Năm 2019, Chang Liu và các cộng sự đã trình bày nghiên cứu của họ trên bộ mô phỏng thử nghiệm CityFlow [19]. Mặc dù cung cấp một tình huống giao

²Ảnh được lấy từ <https://www.aimsun.com/aimsun-next-case-studies/manhattan-traffic-model-mtm/> và [15]

thông sát với nhu cầu trong thế giới thực và được sử dụng phổ biến là Manhattan, New York, tuy nhiên với những sự hỗ trợ hạn chế và độ hiểu chính giao thông không được chặt chẽ, bộ mô phỏng này đã không được các chuyên gia trong lĩnh vực giao thông đánh giá cao.

Ngoài ra, một nghiên cứu về phương pháp điều khiển đèn giao thông phối hợp được trình bày bởi Tong Thanh Pham, Tim Brys và Matthew E.Taylor [15] đã thực hiện dựa trên bộ mô phỏng AIM [9]. Nhưng hạn chế lớn nhất của bộ mô phỏng này là nó chủ yếu bao gồm những tình huống giao thông theo kiểu dạng mạng lưới đối xứng đơn giản và ít liên quan đến thực tế.

2.2 Cơ sở lý thuyết

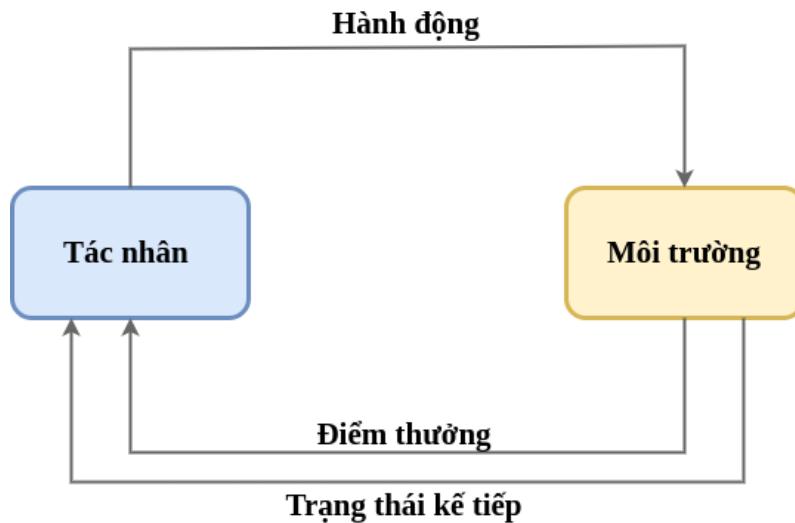
2.2.1 Giới thiệu về Học tăng cường

Học tăng cường là một phần con trong lĩnh vực Trí tuệ nhân tạo, bắt nguồn từ lý thuyết tối ưu. Về căn bản thì Học tăng cường là một vòng lặp phản hồi có điều kiện thông qua nhiều bước thời gian. Tại mỗi bước thời gian, một tác nhân tương tác với môi trường bằng cách quan sát các mô tả trạng thái trong môi trường đó và phản hồi lại thông qua việc thực hiện một hành động khả thi. Môi trường sẽ chấp nhận hành động và thay đổi sang trạng thái kế tiếp. Sau đó, nó sẽ gửi thông tin và điểm thưởng tại trạng thái kế tiếp đó lại cho tác nhân, và chuyển sang bước thời gian tiếp theo. Hình 2.4 minh họa một bước thời gian (t) của một bài toán Học tăng cường.

Nói tóm lại bài toán học tăng cường có hai thành phần chính đó là **tác nhân** tương tác với **môi trường** được mô hình hóa dưới dạng một quy trình quyết định Markov (MDP) để tìm ra chiến lược làm tối ưu hàm điểm thưởng - được xem như là một hàm đánh giá độ tốt xấu cho mục tiêu đặt ra.

Một quy trình quyết định Markov được xác định bởi:

- Một tập hữu hạn các trạng thái S .
- Một tập các hành động khả thi A .



HÌNH 2.4: Minh họa một bước thời gian của bài toán Học tăng cường

- Một hàm xác suất chuyển đổi trạng thái $P(s_t, a_t, s_{t+1})$ hay còn gọi là hàm dịch chuyển giúp xác định xác suất tại trạng thái s_t thực hiện hành động a_t để chuyển sang thái kế tiếp s_{t+1} .
- Một hàm điểm thưởng $R(s_t, a_t)$ để xác định điểm thưởng nhận được khi thực hiện hành động a_t tại trạng thái s_t .
- Một hệ số chiết khấu $\gamma \in (0, 1)$

Trong đó: s_t là trạng thái tại bước thời gian thứ t ; s_T là trạng thái tại bước thời gian cuối cùng.

Theo đó, bài toán Học tăng cường sẽ bắt đầu tại bước thời gian $t = 0$, kết thúc tại thời điểm $t = T$ và một quá trình như vậy sẽ được xem là một "episode". Một chuỗi những trải nghiệm qua một episode được gọi là một "trajectory", ký hiệu τ . Ta có:

$$\tau = (s_0, a_0, r_0), (s_1, a_1, r_1), \dots, (s_T, a_T, r_T) \quad (2.1)$$

Dựa vào những khái niệm trên, chúng ta có điểm thưởng tích lũy tại một episode là:

$$R(\tau) = r_0 + \gamma \cdot r_1 + \gamma^2 \cdot r_2 + \dots + \gamma^T \cdot r_T = \sum_{t=0}^T \gamma^t \cdot r_t \quad (2.2)$$

Điểm thưởng kỳ vọng qua nhiều trajectory là:

$$J(\tau) = E_{\tau \sim \pi}[R(\tau)] = E_{\tau}[\sum_{t=0}^T \gamma^t \cdot r_t] \quad (2.3)$$

Với π được gọi là một chiến lược, cái ánh xạ các trạng thái với các hành động. ($\pi: S \rightarrow A$)

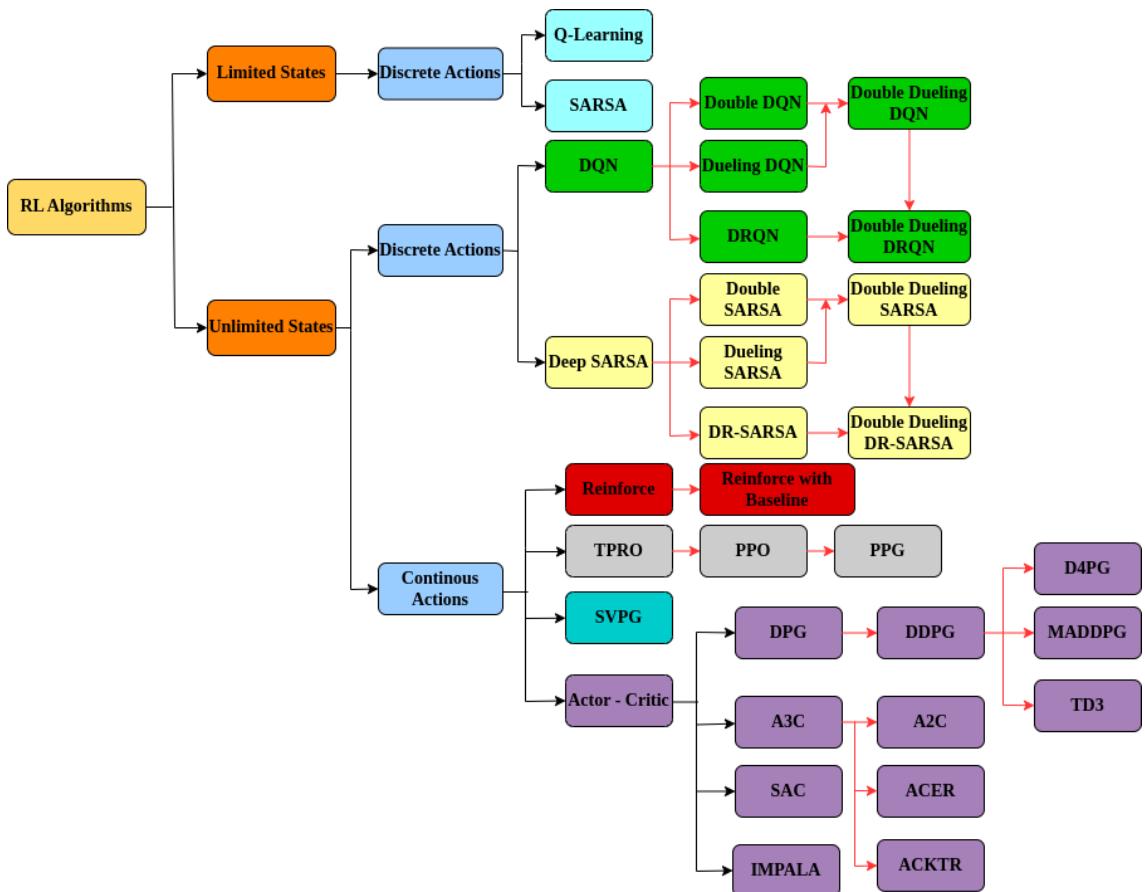
Dựa vào không gian trạng thái và tập hành động có sẵn trong môi trường, các thuật toán Học tăng cường có thể chia làm ba loại chính, tuy nhiên hầu hết thành phần bên trong chúng đều có sử dụng các mạng Neural sâu (Deep Neural Network - DNN) để tận dụng khả năng ước tính xấp xỉ vượt trội của chúng. Hình 2.5 phác thảo sơ bộ về ba phân lớp chính của các thuật toán học tăng cường cùng với các thuật toán tiêu biểu của mỗi lớp.

Theo đó, ba loại chính của các thuật toán Học tăng cường là:

- **Các thuật toán với môi trường có không gian trạng thái bị giới hạn và không gian hành động rời rạc.** Đây là những thuật toán thích hợp để áp dụng cho những tác vụ có môi trường đơn giản. Những thuật toán này sẽ điều khiển những tác nhân lựa chọn một trong những hành động đã được cho trước và đưa môi trường đến những trạng thái đã được biết trước.
- **Các thuật toán với môi trường có không gian trạng thái không bị giới hạn và không gian hành động rời rạc.** Trong một số trò chơi như Snake hay Sokoban, chúng được xem là những trò chơi phức tạp vì có không gian trạng thái lớn nhưng những hành động khả thi mà tác nhân có thể thực hiện chỉ giới hạn trong một số lượng hữu hạn.

Những thuật toán trong loại này rất hữu dụng để giải quyết những bài toán trong môi trường như vậy vì trong thuật toán sẽ có một hoặc nhiều mạng DNN, phổ biến nhất là mạng Neural tích chập (Convolution Neural Networks - CNN) để thuận lợi cho việc xử lý và trích xuất những đặc trưng từ những trạng thái nhận được từ môi trường và trả về những hành động có sẵn.

³Ảnh được khai thác từ [2]



HÌNH 2.5: Các thuật toán Học tăng cường được phân loại dựa trên tính chất của không gian trạng thái và tập hành động. Đầu mũi tên màu đỏ thể hiện thuật toán này được dựa trên thuật toán trước đó.³

- Các thuật toán với môi trường có không gian trạng thái không bị giới hạn và không gian hành động liên tục.** Những thuật toán thuộc loại này thường được sử dụng trong các bài toán có không gian tìm kiếm tương đối lớn và không gian hành động là liên tục, không bị giới hạn trong một số hành động nhất định.

Ưu điểm của không gian hành động liên tục so với không gian hành động rời rạc đó là nó có thể cung cấp những mô tả thực tế hơn về sự chuyển động của sự vật trong các tình huống thực. Vì thế, các thuật toán này thích hợp cho việc giải quyết các tác vụ phức tạp trong đời sống thực tế.

2.2.2 Mô hình hóa bài toán Điều khiển tín hiệu giao thông

Để có thể xử lý bài toán Điều khiển tín hiệu bằng các thuật toán học tăng cường, thì môi trường chứa các giao lộ có nút tín hiệu giao thông phải được mô hình hóa dưới dạng một quy trình quyết định Markov (MDP).

Để có làm rõ hơn về cách mô hình hóa bài toán về dạng MDP, chúng tôi cung cấp những định nghĩa sau:

Các phương tiện di chuyển qua giao lộ luôn sẽ xuất phát từ một đường (đường đi vào) và đi ra khỏi giao lộ từ một đường khác (đường ra) và mỗi đường như vậy sẽ bao gồm một hoặc nhiều làn đường.

Định nghĩa 1 (Chuyển động giao thông.) Một chuyển động giao thông được định nghĩa là lưu lượng giao thông chuyển động qua giao lộ từ một làn đường đi vào đến một làn đường ra theo hướng di chuyển xác định.

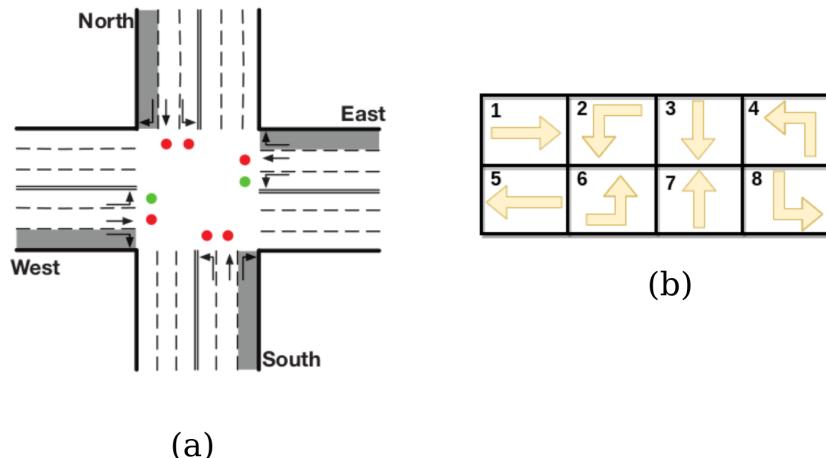
Định nghĩa 2 (Tín hiệu di chuyển) Tín hiệu di chuyển là tín hiệu quyết định cho sự di chuyển hoặc dừng lại của các hướng chuyển động giao thông bị phụ thuộc bởi tín hiệu đèn.

Các phương tiện qua lại tại giao lộ, xuất phát từ một đường đến luôn di chuyển theo một hướng nhất định trong ba lựa chọn là: rẽ trái, rẽ phải hoặc đi thẳng qua. Hình 2.6(a) đưa ra một ví dụ về các hướng di chuyển có thể của các phương tiện khi đi qua một ngã tư. Trong đó, hướng rẽ phải được tô đen để minh họa rằng đây là hướng di chuyển không phụ thuộc vào tín hiệu. Thực tế, tại một số giao lộ trong các thành phố lớn, các phương tiện được cho phép rẽ phải bất chấp tín hiệu đèn như thế nào và trong khóa luận này, chúng tôi sẽ áp dụng quy tắc đó vào trong mô hình.

Tại mỗi chuyển động giao thông, các tín hiệu di chuyển có thể được mã hóa về hai bit 0 và 1. Với 1 đại diện cho tín hiệu "xanh", và 0 đại diện cho tín hiệu "đỏ".

Định nghĩa 3 (Giai đoạn tín hiệu - signal phase.) Việc điều khiển tín hiệu đèn được phân ra theo một tập các giai đoạn (phases). Mỗi giai đoạn tín hiệu được xác định là một tập các hướng chuyển động giao thông được cho phép.

Bên cạnh đó, chúng ta cũng có thể sử dụng một vector 8-bit để đại diện cho một giai đoạn tín hiệu.



HÌNH 2.6: Minh họa 8 hướng di chuyển phụ thuộc vào tín hiệu đèn tại một ngã tư và 8 hướng chuyển động giao thông tương ứng.

Như trong hình 2.6, hướng chuyển động số 2 và số 6 được kích hoạt vì có tín hiệu xanh ở mỗi hướng di chuyển, tức là các phương tiện ở hướng Đông và hướng Tây được cho phép rẽ trái tại giai đoạn tín hiệu này. Vector đại diện cho giai đoạn tín hiệu này là: [0, 1, 0, 0, 0, 1, 0, 0].

Trong khóa luận này, chúng tôi quy ước S_i là tập tất cả các giai đoạn và $s \in S_i$ là mỗi giai đoạn tín hiệu tại giao lộ thứ i . Tại mỗi bước thời gian t , tác nhân RL hay được xem là bộ điều khiển tín hiệu có nhiệm vụ cho phép sự kích hoạt của một số sự kết hợp những giai đoạn không xung đột để làm tối ưu một mục tiêu dài hạn. Cụ thể, một tác nhân RL trong bài toán Điều khiển tín hiệu đèn giao thông được định nghĩa như sau:

- **Trạng thái (S):** Tác nhân quan sát không gian trạng thái được xác định bởi số lượng các phương tiện trên những đường đi vào và giai đoạn tín hiệu hiện tại được kích hoạt.
- **Hành động (A):** Tại mỗi bước thời gian t , tác nhân chọn một giai đoạn tín hiệu kích hoạt như một hành động a_t của nó để chuyển trạng thái của môi

trường sang bước thời gian kế tiếp.

Nếu giai đoạn tín hiệu được chọn khác với giai đoạn tín hiệu tại thì một giai đoạn màu vàng được phát sinh trong môi trường sẽ tự động kích hoạt trong một khoảng thời gian được quy định trước.

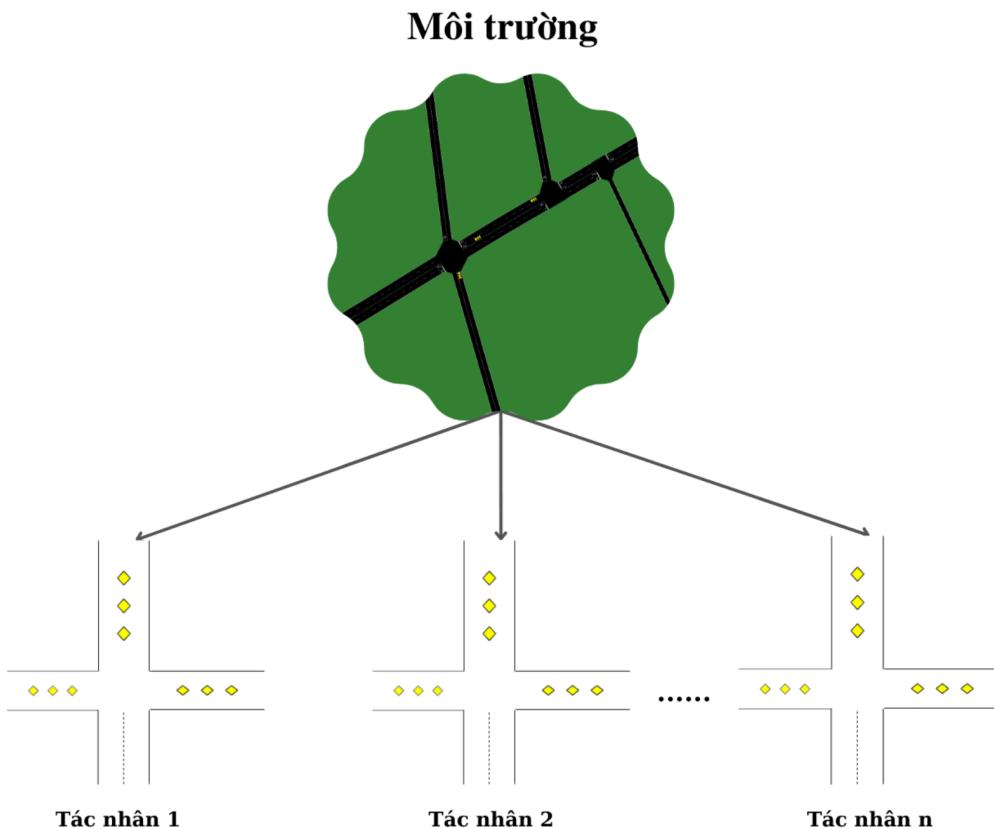
- **Hàm dịch chuyển P:** Hàm dịch chuyển được xác định thông qua sự thay đổi trạng thái giao thông tại giao lộ tuân theo những chỉ định tín hiệu. Những sự thay đổi này có thể được quan sát thông qua một môi trường mô phỏng giao thông hoặc thông qua những bộ cảm biến được lắp đặt trong thế giới thực.
- **Điểm thường R:** Có rất nhiều các chỉ số có được dùng như là điểm thường để đánh giá độ hiệu quả các tác nhân RL, và hầu hết các chỉ số đều liên quan đến những yếu tố ảnh hưởng đến sự ùn tắc giao thông. Trong khóa luận này, chúng tôi đề xuất các chỉ số gồm: độ dài hàng đợi (số lượng xe phải đợi) trong tất cả làn đường (queue length), tổng thời gian trễ (delays), khoảng thời gian di chuyển (duration) và tổng thời gian chờ của các phương tiện (total waiting time) làm hàm điểm thường.

Chi tiết hơn về các chỉ số này sẽ được trình bày ở phần.....

2.2.3 Điều khiển đa tác nhân

Khi giải quyết bài toán Điều khiển tín hiệu đèn giao thông, chúng ta không chỉ xem xét việc xử lý trên bản đồ chỉ có một giao lộ. Thay vào đó, đối với nhiều tình huống giao thông trong thực tế, chúng ta cũng phải tìm cách để giảm thiểu ùn tắc giao thông trong một khu vực trọng điểm có nhiều giao lộ.

Trong trường hợp này, khu vực giao thông cần được xử lý sẽ được chia thành nhiều khu vực nhỏ, trong đó mỗi khu vực nhỏ đều chứa một giao lộ và được quản lý bởi một tác nhân. Mỗi tác nhân sẽ vẫn hoạt động bằng cách quan sát môi trường và lựa chọn thực hiện hành động dựa theo những quan sát đó. Một vấn đề mà có nhiều tác nhân cùng tương tác với một môi trường và mỗi tác nhân đều dựa trên những mô hình của thuật toán Học tăng cường để cập nhật lược của chúng như vậy được gọi là vấn đề **Học tăng cường đa tác nhân** (MARL) và cụ thể hơn trong bài toán Điều khiển tín hiệu giao thông thì sẽ được gọi là **điều khiển đa tác nhân**.

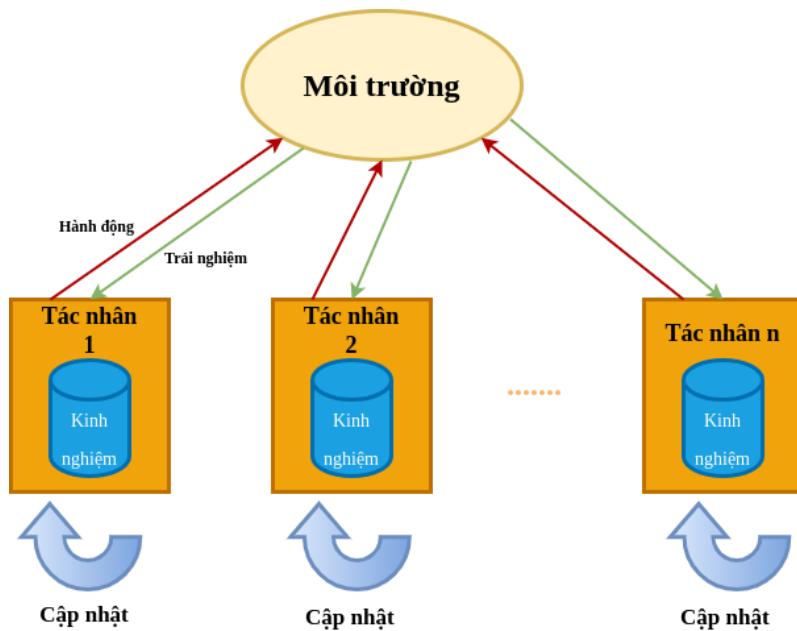


HÌNH 2.7: Vấn đề điều khiển đa tác nhân đối với môi trường có nhiều giao lô.

Về tổng quan, có hai cách tiếp cận chính theo hai mô hình để tìm ra chiến lược tối ưu đối với vấn đề có đa tác nhân đó là: *mô hình đa tác nhân độc lập* (independent agents model) và *mô hình đa tác nhân kết hợp* (cooperative agents model).

- **Mô hình đa tác nhân độc lập:** Trong mô hình này, mỗi tác nhân được huấn luyện độc lập để tìm ra chiến lược tối ưu của riêng nó và xem các tác nhân khác như một phần của môi trường. Hình 2.8 minh họa mô hình điều khiển đa tác nhân độc lập.

Có thể thấy, mô hình này được thiết kế rất đơn giản vì thế nó sẽ góp phần giảm thiểu độ phức tạp trong các hệ thống áp dụng chúng và đồng thời đảm bảo được sự ổn định cũng như khả năng mở rộng trên các môi trường có nhiều tác nhân.



HÌNH 2.8: Minh họa mô hình đa tác nhân độc lập. Mũi tên màu đỏ thể hiện tác nhân thực hiện hành động lên môi trường và mũi tên màu xanh lá thể hiện trải nghiệm mà môi trường trả về cho tác nhân khi sau khi thực hiện hành động.

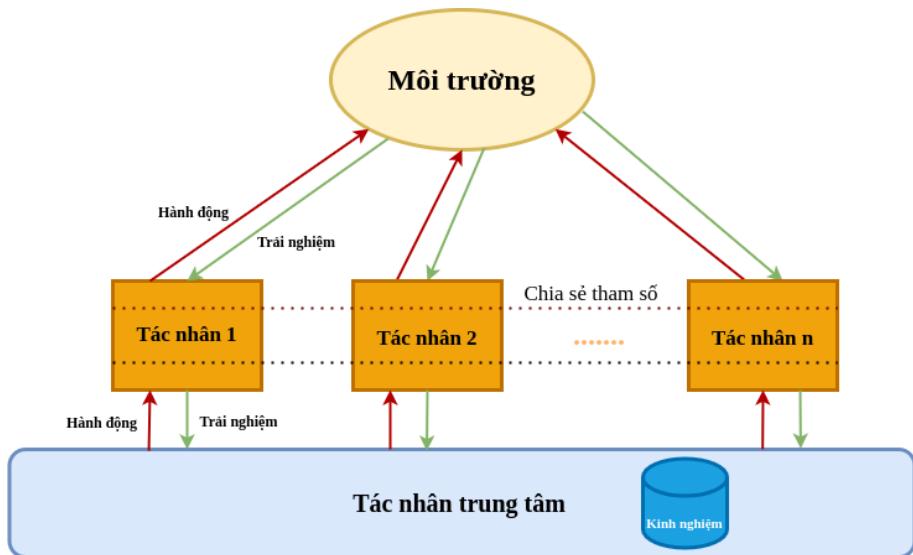
Tuy nhiên, một điểm hạn chế lớn của mô hình này đó là các tác nhân sẽ không biết trạng thái cũng như hành vi của các tác nhân khác. Điều này sẽ làm cho môi trường không cố định bởi vì các tác nhân liên tục cập nhật và thay đổi chiến lược của chúng và dẫn đến môi trường luôn có những thay đổi bên ngoài tầm kiểm soát của riêng một tác nhân, làm cho các tác nhân phải đang học để tối ưu một mục tiêu chuyển động.

Bên cạnh đó, việc các tác nhân chỉ cố gắng tối ưu điểm thưởng của riêng nó dựa trên những quan sát cục bộ làm cho mô hình khó hội tụ về một giải pháp để tối ưu toàn bộ hệ thống.

- **Mô hình đa tác nhân kết hợp:** Trong mô hình đa tác nhân kết hợp, các tác nhân học những trải nghiệm từ môi trường và gởi thông tin về một tác nhân trung tâm.

Tác nhân trung tâm này có nhiệm vụ tổng hợp những trải nghiệm nhận được, sau đó phân tích và tìm ra chiến lược tốt nhất để tối ưu điểm thưởng

toàn cục và cuối cùng là phân phối lại hành động cho từng tác nhân cục bộ để tiếp tục học hỏi và nhận những trải nghiệm từ môi trường. Hình 2.9 minh họa thiết kế của mô hình đa tác nhân kết hợp.



HÌNH 2.9: Minh họa mô hình đa tác nhân kết hợp. Mũi tên màu đỏ thể hiện tác nhân thực hiện hành động lên môi trường và mũi tên màu xanh lá thể hiện điểm thưởng môi trường trả về cho tác nhân khi sau khi thực hiện hành động.

So với mô hình đa tác nhân độc lập thì mô hình đa tác nhân kết hợp có độ phức tạp lớn hơn vì ngoài việc thiết kế thêm một tác nhân trung tâm để tổng hợp và phân tích trải nghiệm thì phải thiết kế thêm một cơ chế để các tác nhân có thể học hỏi kinh nghiệm lẫn nhau thông qua việc chia sẻ tham số, tuy nhiên điều này giúp được khôi lượng học tập cho mỗi tác nhân cũng như tạo ra được môi trường cố định vì các tác nhân cục bộ lúc này đã biết về những thay đổi trong chiến lược và hành vi của nhau. Và việc thiết lập được môi trường cố định góp phần đảm bảo cho việc tìm ra một chiến lược hội tụ cho việc tối ưu mục tiêu toàn cục của mô hình.

Bên cạnh những ưu điểm nổi bật, thì có mô hình đa tác nhân kết hợp cũng có những hạn chế của riêng nó. *Thứ nhất*, trong một môi trường có không gian trạng thái lớn, khi đó không gian hành động cũng sẽ tăng lên theo. Lúc này, tác nhân trung tâm sẽ phải xem xét quá nhiều hành động và dẫn đến

sự chậm trễ trong việc ra quyết định, điều này sẽ gây ra độ trễ cao khi áp dụng vào các vấn đề thực tế.

Thứ hai, khi gặp phải một môi trường có quá nhiều tác nhân (khoảng vài ngàn tác nhân trở lên) thì tại mỗi bước thời gian, tác nhân trung tâm phải tương tác với rất nhiều tác nhân cục bộ khác và sự sự quá tải trong giao tiếp có thể xảy ra khiến tác nhân trung tâm không hoạt động trong một số thời điểm.

Chương 3

CÁC BỘ ĐIỀU KHIỂN ĐỀ XUẤT CHO BÀI TOÁN ĐIỀU KHIỂN TÍN HIỆU ĐÈN GIAO THÔNG

Trong chương này, chúng tôi sẽ trình bày về những bộ điều khiển đã được chúng tôi triển khai để xử lý bài toán Điều khiển tín hiệu đèn giao thông. Về căn bản thì chúng được chia thành 2 nhóm chính bao gồm: *bộ điều khiển cơ bản* và *bộ điều khiển dựa trên Học tăng cường*.

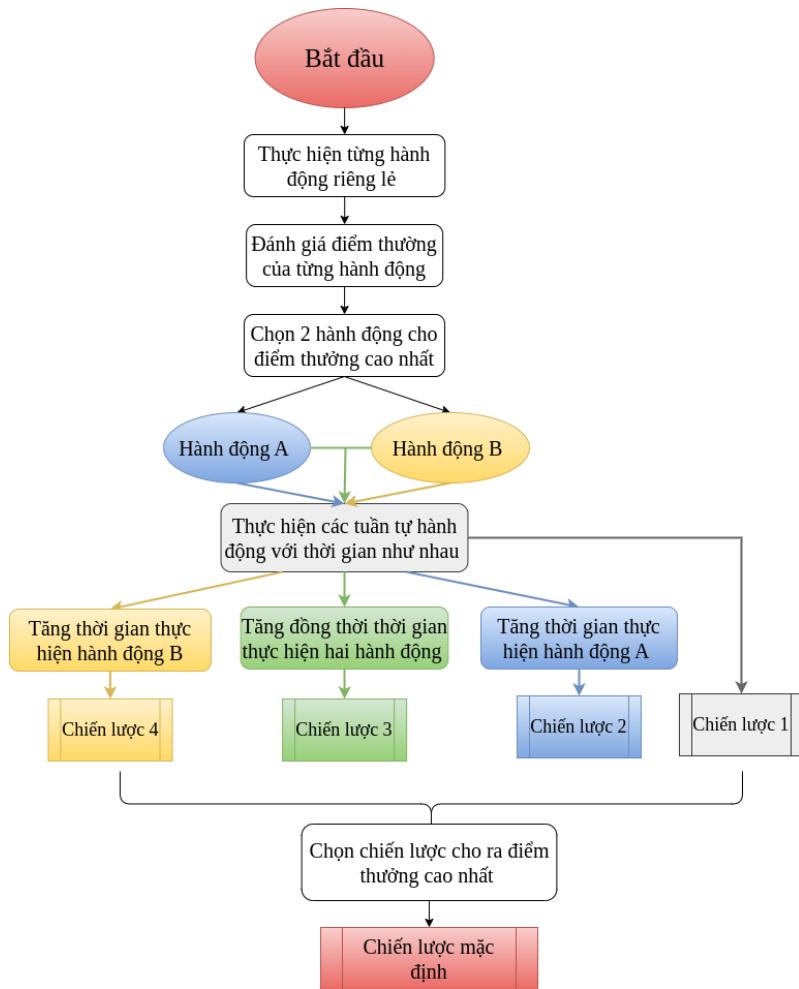
3.1 Bộ điều khiển cơ bản

Như đã đề cập ở trên, đối với bài toán Điều khiển tín hiệu giao thông, thi điều quan trọng là các tác nhân điều khiển phải thiết kế được những chiến lược tốt để thay đổi các giai đoạn tín hiệu phù hợp tại các giao lộ.

Bộ điều khiển cơ bản sẽ sử dụng các tác nhân thiết kế chiến lược theo kiểu truyền thống và các chiến lược này sẽ được đem đi so sánh với các chiến lược được thiết kế bởi bộ điều khiển dựa trên Học tăng cường. Cụ thể, bộ điều khiển cơ bản sẽ bao gồm hai tác nhân là *tác nhân ngẫu nhiên* và *tác nhân mặc định*. Chi tiết về hai loại tác nhân này sẽ được trình bày dưới đây:

- Tác nhân ngẫu nhiên: là loại tác nhân sẽ đưa ra các hành động một cách ngẫu nhiên theo từng giai đoạn thời gian mà không quan tâm đến tình trạng lưu lượng giao thông tại các giao lộ hiện tại là ra sao. Tuy nhiên, các nguyên tắc giao thông vẫn phải được đảm bảo.

- Tác nhân mặc định: Tại mỗi bản đồ khác nhau, chúng tôi sẽ thiết kế các chiến lược mặc định khác nhau và các tác nhân này chỉ việc thực hiện tuân tự các hành động của chiến lược đó. Hình 3.1 minh họa quy trình chúng tôi thiết kế chiến lược mặc định cho tác nhân.



HÌNH 3.1: Minh họa quy trình thiết kế chiến lược thủ công.

Tại mỗi trong môi trường (được hiểu là bản đồ mô phỏng lưu lượng giao thông tại giao lộ) sẽ có một tập hành động khả thi. Mỗi hành động được đánh số 0, 1, 2,... và quy định màu đèn cụ thể cho mỗi đèn tín hiệu trong môi trường. Các môi trường khác nhau sẽ có số lượng hành động khả thi khác nhau và các hành động cũng sẽ không giống nhau.

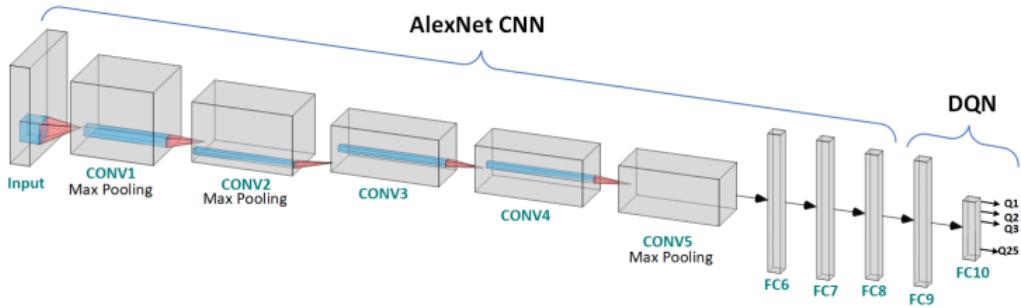
Về quy trình thiết kế chiến lược cho mỗi bản đồ, chúng tôi thực hiện như sau. Đầu tiên, chúng tôi cho tác nhân trong môi trường thực hiện duy nhất một hành động và tính toán điểm thưởng nhận được. Cứ như vậy, áp dụng đối với tất cả các hành động. Tiếp theo, chúng tôi chọn ra hai động cho ra điểm thưởng tốt nhất, ví dụ hai hành động được chọn ra là **hành động A** và **hành động B**. Trong bước kế tiếp, chúng tôi thực hiện tuần tự tất cả các hành động khả thi với thời gian như nhau trong cùng một episode và đánh dấu đây là *chiến lược 1*. Bước tiếp theo, chúng tôi tăng thời gian thực hiện hành động A lên, sau mỗi lần tăng thì đều tính toán điểm thưởng nhận được và chúng tôi sẽ tăng cho đến khi tìm được mức thời gian hợp lý để thiết lập cho hành động A và đánh dấu đó là *chiến lược 2*. Thực hiện tương tự đối với hành động B, chúng tôi sẽ có được *chiến lược 3*. Sau đó, chúng tôi tiến hành tăng đồng thời thời gian thực hiện hành động A và B lên trong cùng một episode, cũng đánh giá điểm thưởng sau mỗi lần tăng và dừng lại cho đến khi tìm được mức thời gian hợp lý có thể thiết lập cho hai động A và B, đánh dấu đây là *chiến lược 4*. Cuối cùng, chúng tôi sẽ so sánh điểm thưởng nhận được từ 4 chiến lược và chọn chiến lược cho kết quả tốt nhất làm chiến lược mặc định cần tìm.

3.2 Bộ điều khiển dựa trên Học tăng cường

Bộ điều khiển dựa trên Học tăng cường là các bộ điều khiển sử dụng tác nhân học tăng cường để tìm ra chiến lược điều khiển tối ưu dựa trên những trải nghiệm có được thông qua việc tương tác với môi trường. Hay nói cách khác, chúng tôi sẽ áp dụng các thuật toán học tăng cường vào việc tối ưu các bộ điều khiển tín hiệu này. Dưới đây, chúng tôi sẽ trình bày tổng quan về một số thuật toán đã được áp dụng và triển khai trong khóa luận này.

3.2.1 Deep Q-Network (DQN)

DQN là thuật toán hoạt động tốt trong môi trường có trạng thái không giới hạn và không gian hành động rời rạc. Nó được phát triển dựa trên ý tưởng của thuật toán Q-Learning, một thuật toán phổ biến được dùng cho bài toán điều khiển tín hiệu đèn giao thông nhờ tính đơn giản và hiệu quả của nó [4] [8].



HÌNH 3.2: DQN sử dụng kiến trúc mạng AlexNet¹

Thực tế, DQN là một phiên bản kết hợp của Q-Learning và học sâu. Nó sử dụng Neural Network để tạo ra những giá trị ước tính Q-value tương ứng với số lượng hành động khả thi tại một trạng thái nhất định, sau đó tác nhân DQN sẽ lựa chọn hành động cho ra kết quả Q-value cao nhất để thực hiện và quan sát những thay đổi của môi trường hiện tại và điểm thưởng được tạo ra từ hành động. Trong đó, Q-value là điểm thưởng kỳ vọng nhận được từ việc thực hiện hành động a tại trạng thái s và được ký hiệu là $Q(s,a)$.

Hình 3.2 minh họa một tác nhân DQN sử dụng kiến trúc mạng AlexNet để ước tính các giá trị Q-value.

Một trong những đặc điểm nổi bật của thuật toán DQN để giúp nó trở nên hiệu quả hơn so với Q-Learning đó là *bộ lưu trữ trải nghiệm* (*Experience Buffer*). Những trải nghiệm mà tác nhân học được trong quá trình tương tác với môi trường sẽ được lưu trữ trong bộ nhớ này để phục vụ cho việc tái sử dụng ở những bước thời gian tiếp theo. Mỗi khi huấn luyện một tác nhân DQN thì một hoặc nhiều lô dữ liệu (data-batch) được lấy mẫu ngẫu nhiên từ bộ nhớ để cập nhật bộ tham số θ của mạng. Mã giả của thuật toán DQN được minh họa như trong **Thuật toán 1**.

Một số ký hiệu được dùng trong giải thuật trên:

- $\delta_{s_{t+1}^i} = 0$ trong trường hợp s_{t+1}^i là trạng thái kết thúc, ngược lại giá trị này mặc định là 1

¹Ảnh được lấy từ [3]

Thuật toán 1 Thuật toán DQN

```

1: function DQN(MAX_STEP,  $\alpha$ ,  $\tau$ , B, U, N, K,  $\theta$ )
2:   Khởi tạo: hệ số học  $\alpha$ , số lượng trajectories  $\tau$ , số lô trên một bước huấn
      luyện B, số lần cập nhật trên mỗi lô U, kích thước lô N, kích thước bộ lưu trữ
      trải nghiệm K
3:   Khởi tạo ngẫu nhiên: Bộ tham số  $\theta$ 
4:   for  $m \leftarrow 1$  to MAX_STEP do
5:     Thu thập và lưu trữ những trải nghiệm  $(s_t, a_t, r_t, s_{t+1})$ 
6:     for  $b \leftarrow 1$  to B do
7:       Lấy mẫu một lô b của những trải nghiệm từ bộ lưu trữ trải nghiệm
8:       for  $u \leftarrow 1$  to U do
9:         for  $i \leftarrow 1$  to N do
10:           $y_i = r_i + \delta_{s_{t+1}^i} \cdot \gamma \cdot \max_{a_{t+1}^i} Q^{\pi_\theta}(s_{t+1}^i, a_{t+1}^i)$      $\triangleright$  Tính Q-values
11:        end for
12:         $L(\theta) = \frac{1}{N} \sum_i (y_i - Q^{\pi_\theta}(s_i, a_i))^2$                        $\triangleright$  Tính hàm Loss
13:         $\theta = \theta - \alpha \cdot \nabla_\theta L(\theta)$                                  $\triangleright$  Cập nhật tham số mạng
14:        end for
15:      end for
16:      Cập nhật  $\tau$ 
17:    end for

```

- $Q^{\pi_\theta}(s, a)$ là giá trị điểm thưởng khi thực hiện hành động a dựa theo chiến lược π được quyết định bởi bộ trọng số θ tại trạng thái s . Lưu ý rằng mỗi bộ trọng số θ khác nhau của mạng sẽ cho ra một chiến lược π khác nhau.

3.2.2 Double Deep Q-Network (Double DQN)

Double DQN là biến thể của thuật toán DQN nhằm khắc phục một điểm hạn chế lớn của thuật toán này đó là đánh giá cao giá trị hành động. Như đã đề cập ở phần trên, tác nhân DQN có xu hướng chọn hành động cho ra Q-value cao nhất ở trạng thái tiếp theo, điều này dễ dẫn đến việc thuật toán rơi vào một điểm tối ưu cục bộ và không tìm thấy hành động tối ưu trong quá trình huấn luyện. Đối với môi trường có không gian trạng thái và hành động lớn thì vấn đề này càng dễ xảy ra.

Để quá trình huấn luyện trở nên ổn định hơn thì Double DQN đã ra đời với ý tưởng là thiết kế thêm một mạng Neural Network mục tiêu với cấu trúc tương tự với mạng ban đầu nhưng bộ trọng số của mạng này sẽ không thay đổi liên tục

mà chỉ thay đổi sau một khoảng thời gian cố định. Cụ thể, trong thuật toán của Double DQN, sẽ có hai mạng Neural Network khác nhau. Một mạng mục tiêu với bộ trọng số φ cố định, và mạng còn lại với bộ trọng số θ được cập nhật liên tục. Sau một khoảng thời gian định kỳ thì sẽ gán $\varphi = \theta$. Mã giả của thuật toán DQN được trình bày trong **Thuật toán 2**

Thuật toán 2 Thuật toán DoubleDQN

```

1: function DQN(MAX_STEP,  $\alpha$ ,  $\tau$ ,  $B$ ,  $U$ ,  $N$ ,  $K$ ,  $F$ ,  $\theta$ ,  $\varphi$ )
2:   Khởi tạo: hệ số học  $\alpha$ , số lượng trajectories  $\tau$ , số lô trên một bước huấn luyện  $B$ , số lần cập nhật trên mỗi lô  $U$ , kích thước lô  $N$ , kích thước bộ lưu trữ trải nghiệm tần số cập nhật mạng mục tiêu  $F$ .
3:   Khởi tạo ngẫu nhiên: Bộ tham số  $\theta$ 
4:   Khởi tạo: Bộ tham số ban đầu của mạng mục tiêu  $\varphi = \theta$ 
5:   for  $m \leftarrow 1$  to MAX_STEP do
6:     Thu thập và lưu trữ những trải nghiệm  $(s_t, a_t, r_t, s_{t+1})$ 
7:     for  $b \leftarrow 1$  to  $B$  do
8:       Lấy mẫu một lô  $b$  của những trải nghiệm từ bộ lưu trữ trải nghiệm
9:       for  $u \leftarrow 1$  to  $U$  do
10:        for  $i \leftarrow 1$  to  $N$  do
11:           $y_i = r_i + \delta_{s_{t+1}^i} \cdot \gamma \cdot Q^{\pi_\varphi}(s_{t+1}^i, \max_{a_{t+1}^i} Q^{\pi_\theta}(s_{t+1}^i, a_{t+1}^i))$   $\triangleright$  Tính  $Q\_values$ 
12:        end for
13:         $L(\theta) = \frac{1}{N} \sum_i (y_i - Q^{\pi_\theta}(s_i, a_i))^2$   $\triangleright$  Tính hàm Loss
14:         $\theta = \theta - \alpha \cdot \nabla_\theta L(\theta)$   $\triangleright$  Cập nhật tham số mạng
15:        end for
16:      end for
17:    end for
18:    Cập nhật  $\tau$ 
19:    if ( $m \% F == 0$ ) then:  $\varphi = \theta$   $\triangleright$  Cập nhật tham số cho mạng mục tiêu
20:    end if
21:  end for

```

3.2.3 Proximal Policy Optimization (PPO)

PPO thuộc nhóm thuật toán hoạt động trên môi trường có không gian trạng thái không giới hạn và không gian hành động liên tục. Các thuật toán thuộc loại này huấn luyện các tác nhân tìm ra chiến lược được tham số hóa π_θ để tối ưu hóa tổng điểm thưởng nhận được bởi vì việc tính toán hết các Q-value cho mỗi trạng thái là rất khó khăn do số lượng hành động lớn.

Cụ thể, công việc của các tác nhân là tối ưu hóa hàm mục tiêu phụ thuộc vào bộ tham số θ (3.1) và sử dụng gradient ascent để cập nhật bộ tham số.

$$J(\theta) = \sum_{s \in S} \rho_{\pi_\theta}(s) \sum_{a \in A} Q^{\pi_\theta}(s, a) \pi_\theta(a|s) \quad (3.1)$$

trong đó, $\rho_{\pi_\theta}(s)$ là xác suất để trạng thái thay đổi dựa theo chiến lược π_θ .

Phương pháp này còn được gọi là **Đạo hàm chiến lược** (Policy Gradient - PG). Có rất nhiều nhóm thuật toán khác nhau được phát triển dựa trên ý tưởng PG, một trong số đó là nhóm thuật toán Actor-Critic. Tất cả các thuật toán Actor-Critic đều có hai thành phần:

- Mạng Actor: điều chỉnh tham số θ của chiến lược π_θ
- Mạng Critic: học một hàm giá trị để đánh giá cặp (s, a) .

PPO là thuật toán được xây dựng dựa trên nhóm thuật toán Actor-Critic, với ý tưởng là thay thế hàm mục tiêu $J(\theta)$ ban đầu bởi hàm mục tiêu của riêng nó để giúp quá trình huấn luyện trở nên ổn định hơn.

Trong PPO, hàm mục tiêu thay thế được ký hiệu là $J^{CPI}(\theta)$ và có công thức như ở 3.2

$$J^{CPI}(\theta) = E_t[r_t(\theta) \cdot A_t^{\pi_{\theta_{old}}}] \approx J(\pi_\theta) - J(\pi_{\theta_{old}}) \geq 0 \quad (3.2)$$

với $E_t[KL(\pi_\theta || \pi_{\theta_{old}})] < \delta$

Trong đó:

- $\pi_{\theta_{old}}$: chiến lược trước khi cập nhật tham số
- π_θ : chiến lược sau khi cập nhật tham số
- $r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{old}}(a_t | s_t)}$: tỷ lệ xác suất hành động giữa hai chiến lược liên tiếp.
- $A^{\pi_{\theta_{old}}} = Q^{\pi_{\theta_{old}}}(s_t, a_t) - V^{\pi_{\theta_{old}}}(s_t)$: đo lường mức độ tốt hay xấu của một hành động tại một trạng thái nhất định (advantage function)
- $J(\pi_\theta) - J(\pi_{\theta_{old}})$: đo lường sự cải thiện giữa hai chiến lược, cụ thể ở đây là chiến lược trước và sau khi cập nhật tham số.
- $KL(\pi_\theta || \pi_{\theta_{old}}) = \int_{-\infty}^{\infty} \pi_\theta \log \frac{\pi_\theta}{\pi_{\theta_{old}}}$: một chỉ số đo lường sự khác biệt giữa hai trạng thái [1]

- δ : hằng số giới hạn độ lớn của KL

Trong công thức được giới thiệu ở 3.2, cần có thêm một ràng buộc giới hạn độ lớn KL để đảm bảo rằng thuật toán có thể tìm ra chiến lược tối ưu π^* vì nó làm hạn chế sự khác biệt giữa chiến lược mới và chiến lược cũ (sau quá trình cập nhật tham số), tức là chiến lược hiện tại chỉ xem xét các chiến lược xung quanh khu vực của nó trong không gian chiến lược, và khu vực này được gọi là *vùng tin cậy*.

Tuy nhiên, PPO có một biến thể khác, loại bỏ ràng buộc vùng tin cậy này ở hàm mục tiêu để tạo thành một phiên bản đơn giản hơn và giúp giảm độ phức tạp trong việc tính toán KL sau mỗi bước.

$$J^{CPI}(\theta) = E_t[min[r_t(\theta)A_t^{\pi_{\theta_{old}}}, clip(r_t(\theta), 1 - \epsilon, 1 + \epsilon).A_t^{\pi_{\theta_{old}}}]] \quad (3.3)$$

với ϵ là hệ số xác định vùng lân cận

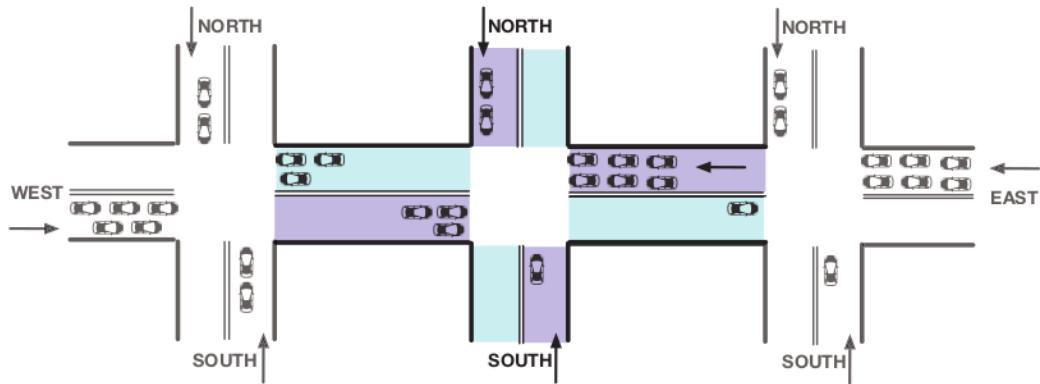
Mặc dù thuật toán PPO dễ hội tụ về một chiến lược tối ưu vì các bước đạo hàm cập nhật tham số của nó có sự giới hạn, tuy nhiên việc biểu diễn không gian đặc trưng chưa hoàn chỉnh vì mặc dù nó có thể thu thập thông tin vị trí phương tiện nhưng lại bỏ qua các đặc trưng phi tuyến của luồng giao thông, điều này làm nó dễ rơi vào các chiến lược tối ưu cục bộ và dẫn đến hiệu suất trong một số trường hợp nằm dưới mức tối ưu. Một giải pháp để tối ưu hơn sự hiệu quả của thuật toán PPO cho bài toán Điều khiển tín hiệu đèn giao thông đã được Liben và Xiaohui giới thiệu trong [10], nhưng nó nằm ngoài phạm vi khóa luận này.

3.2.4 MPLight

MPLight là một phương pháp được lấy ý tưởng dựa trên thuật toán DQN nhưng có một số khác biệt là nó sử dụng khái niệm *pressure* để làm điểm thưởng, đánh giá độ tốt xấu của chiến lược và sử dụng *mô hình FRAP* làm kiến trúc mạng bên trong nó thay vì sử dụng các kiến trúc mạng như LeNet, AlexNet, .v.v thông thường trong DQN.

3.2.4.1 Khái niệm Pressure

Pressure của một giao lộ được xem là sự chênh lệch giữa tổng số phương tiện trong những làn đường đi vào và tổng số phương tiện trong làn đường ra tại giao lộ đó tại một thời điểm nhất định.



HÌNH 3.3: Minh họa những làn đường đi vào và làn đường ra tại một giao lộ. Làn màu tím thể hiện cho làn đường đi vào và làn màu xanh thể hiện cho làn đường ra.²

Xét giao lộ ở giữa trong hình 3.3, chúng ta có tổng số xe trên những làn đường màu tím (làn đường đi vào) là 12 và tổng số xe trên những làn đường màu xanh (làn đường ra) là 4. Lúc này, ta có pressure tại giao lộ đó (ký hiệu là P) là:

$$P = |12 - 4| = 8 \quad (3.4)$$

Tại mỗi giai đoạn tín hiệu $s \in S_i$, thì pressure $p(s)$ sẽ được tính toán và tác nhân sẽ lựa chọn hành động dẫn đến một trạng thái có pressure thấp nhất, điều này giúp giảm tải lưu lượng giao thông tại một giao lộ. Như vậy, mục tiêu của một tác nhân MPLight sẽ là tối ưu tổng điểm thưởng pressure tại một giao lộ.

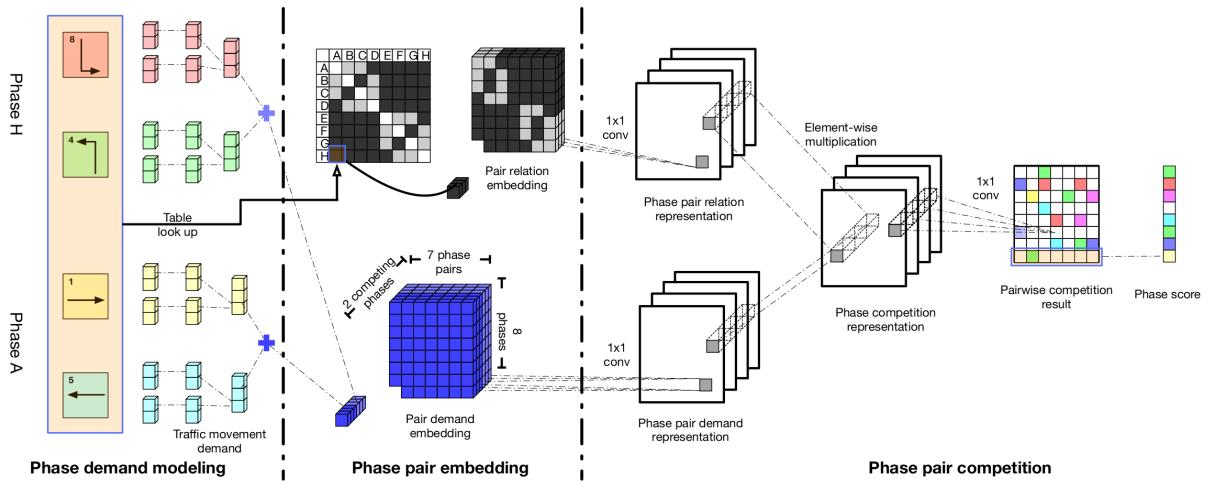
$$r_i = -P_i \quad (3.5)$$

với r_i và P_i lần lượt là điểm thưởng và pressure tại giao lộ thứ i .

3.2.4.2 Mô hình FRAP

Mô hình FRAP đóng vai trò như một mạng Deep Neural Network thông thường trong DQN với nhiệm vụ là tạo ra những ước tính Q-value của các giai đoạn tín hiệu (được hiểu là những hành động khả thi) tại một trạng thái nhất định.

²Nguồn: [7]



HÌNH 3.4: Thiết kế của mô hình FRAP.³

Về cơ bản, mô hình FRAP được thiết kế dựa trên hai nguyên tắc:

- Nguyên tắc cạnh tranh:** Hướng dịch chuyển giao thông có đồng phương tiện hơn thì được xem là có nhu cầu tín hiệu "xanh" cao hơn. Khi có sự xung đột tín hiệu xảy ra, thì hướng chuyển động có nhu cầu hơn sẽ được ưu tiên di chuyển.
- Nguyên tắc bất biến:** Tín hiệu điều khiển giao thông không bị ảnh hưởng bởi tác động của lật và xoay tại các giao lộ đối xứng.

Hình 3.4 minh họa thiết kế của mô hình FRAP và dựa trên thiết kế này, quá trình dự đoán Q-value đã được chia làm 3 giai đoạn được trình bày dưới đây.

Giai đoạn 1: Phase Demand Modeling. Mục tiêu trong giai đoạn này là tạo ra được những biểu diễn cho nhu cầu của từng giai đoạn tín hiệu.

Ban đầu, mô hình sẽ nhận đầu vào là hai vector đặc trưng được lấy từ bộ mô phỏng là f_i^v số lượng xe trên mỗi hướng dịch chuyển và f_s^v thể hiện cho giai đoạn tín hiệu hiện tại. Sau đó, hai vector đặc trưng này được truyền qua hai lớp fully-connected khác nhau để tạo thành hai vector lớp ẩn h_i^v và h_i^s :

$$h_i^v = \text{ReLU}(w^v \cdot f_i^v + b^v) \quad (3.6)$$

³Nguồn: [21]

$$h_i^s = \text{ReLU}(w^s \cdot f_i^s + b^s) \quad (3.7)$$

Tiếp theo, hai vector lớp ẩn này được kết hợp lại và truyền qua một lớp output để tạo ra một vector biểu diễn cho nhu cầu trên hướng chuyển động giao thông thứ i dựa theo 3.8

$$d_i = \text{ReLU}(w^h [h_i^v, h_i^s] + b^h) \quad (3.8)$$

Lưu ý rằng, có tổng cộng 8 hướng chuyển động giao thông đã được trình bày ở phần 2.2.2, và chúng tôi đang xem xét ở hướng chuyển động thứ i $\in \{1, \dots, 8\}$ và trong quá trình hoạt động tại giai đoạn này, các tham số học được tại tất cả các hướng chuyển động luôn được chia sẻ cho nhau.

Để có được vector biểu diễn nhu cầu của giai đoạn tín hiệu **p** tại trạng thái s bất kỳ thì chúng ta chỉ cần cộng hai vector biểu diễn nhu cầu của hai hướng di chuyển không xung đột:

$$d(p) = d_i + d_j \quad (3.9)$$

với $i, j \in \{1, \dots, 8\}$, $i \neq j$ và $p_i = p_j = 1$

Trong hình 2.6, hai hướng chuyển động giao thông 3 và 7 được xem là hai hướng không xung đột.

Giai đoạn 2: Phase Pair Embedding Mục tiêu ở giai đoạn này là tạo ra những biểu diễn về sự cạnh tranh giữa hai giai đoạn tín hiệu tại một trạng thái s $\in S$.

Tại một trạng thái s bất kỳ, sẽ có rất nhiều giai đoạn tín hiệu khác nhau và theo nguyên tắc cạnh tranh thì một giai đoạn tín hiệu sẽ được thực hiện nếu nhu cầu của nó cao hơn tất cả các giai đoạn còn lại. Ở đây, chúng tôi quan tâm đến hai khía cạnh căn bản của sự cạnh tranh bao gồm: *quan hệ* và *nhu cầu* của chúng.

Để biểu diễn được sự cạnh tranh giữa hai giai đoạn tín hiệu, thì thiết kế của mô hình đã tạo ra hai bộ nhúng (embeddings) để biểu diễn hai khía cạnh này là: **Pair relation embedding** (khối màu đen) và **Pair demand embedding** (khối màu xanh) như trong hình 3.4. Giả sử, chúng ta có giai đoạn tín hiệu **p** và giai đoạn tín hiệu cạnh tranh với nó là **q** ($p \neq q$), chúng ta sẽ làm rõ hơn về quá trình biểu diễn sự cạnh tranh giữa một cặp (**p, q**) dưới đây:

- *Pair relation embedding (E)*: Khi một cặp giai đoạn (**p, q**) được xác định, thì quan hệ của cặp này sẽ được thiết lập theo một vector nhúng **e(p, q)**. Tập

hợp các vector nhúng của tất cả cặp giao đoạn tín hiệu sẽ tạo thành một hình khối Pair relation embedding - biểu diễn quan hệ giữa cặp giao đoạn cạnh tranh.

- *Pair demand embedding (D)*: Hình khối Pair demand embedding - biểu diễn nhu cầu của các cặp giao đoạn cạnh tranh được hình thành bằng cách ghép nối các vector biểu diễn nhu cầu của một cặp giao đoạn p, q tạo thành một vector $[d(p), d(q)]$ và sau đó tập hợp tập hợp tất cả vector này lại.

Giai đoạn 3: Phase pair competition. Trong giai này, mô hình sẽ lấy hai tập nhúng E và D thu được từ giai đoạn 2 làm đầu vào và dự đoán Q-value cho mỗi giao đoạn tín hiệu và đồng thời xem xét sự cạnh tranh giữa các giao đoạn với nhau.

Đầu tiên, mô hình sẽ xử lý hai tập đầu vào E và D bằng cách đưa từng tập qua K lớp tích chập với bộ lọc 1x1 [13] (chọn bộ lọc 1x1 để thuận tiện cho việc chia sẻ tham số giữa các cặp giao đoạn). Tại đây, lớp thứ k $\in K$ được biểu diễn như sau:

$$H_k^r = \text{ReLU}(w_k^r \cdot H_{k-1}^r + b_k^r) \quad (3.10)$$

$$H_k^d = \text{ReLU}(w_k^d \cdot H_{k-1}^d + b_k^d) \quad (3.11)$$

trong đó $H^r = E$ và $H^d = D$

Sau đó, thực hiện một phép nhân tương ứng các phần tử trên tensor (element-wise) giữa H^r và H^d sẽ tạo ra được những vector biểu diễn sự cạnh tranh giao đoạn (ký hiệu là H^c):

$$H^c = H^r \otimes H^d \quad (3.12)$$

Tiếp theo, những vector thu được ở bước trên sẽ được truyền qua một lớp tích chập 1x1 khác để thu được một ma trận mà mỗi phần tử biểu diễn sự cạnh tranh giữa các cặp tương ứng. Ký hiệu ma trận này là C, ta có:

$$C = \text{ReLU}(w^c \cdot H^c + b^c) \quad (3.13)$$

Tại mỗi hàng của ma trận (trong hình 3.4) mô hình sẽ chọn ra giao đoạn tín hiệu được ưu tiên nhất tạo thành một phase score hay dễ hiểu hơn là các Q-value của mỗi giao đoạn tín hiệu.

Như đã đề cập ở phần 2.1.2, mô hình FRAP hỗ trợ cho việc giảm kích thước của không gian trạng thái. Điều này là bởi vì nguyên tắc bất biến của mô hình FRAP làm nó không bị ảnh hưởng bởi các phép lật và xoay. Tức là đối với một giai đoạn tín hiệu bất kỳ ta luôn có một giai đoạn tín hiệu khác có quan hệ đối xứng với nó thì mô hình FRAP sẽ chỉ xem xét một trường hợp, điều này giúp nó giảm thiểu đáng kể số mẫu kinh nghiệm mà tác nhân phải khai thác.

3.2.5 Extended MPLight

Extended MPLight thực chất là một phiên bản mở rộng của MPLight và được triển khai tương tự như nhau. Tuy nhiên trong bản mở rộng này, tác nhân sẽ quan sát thêm một số thông tin từ môi trường như: sự chênh lệch giữa *tốc độ trung bình*, *tổng thời gian chờ trung bình* của tất cả phương tiện trên làn đường đi vào và làn đường ra.

Chương 4

THỰC NGHIỆM

4.1 Bộ mô phỏng được sử dụng

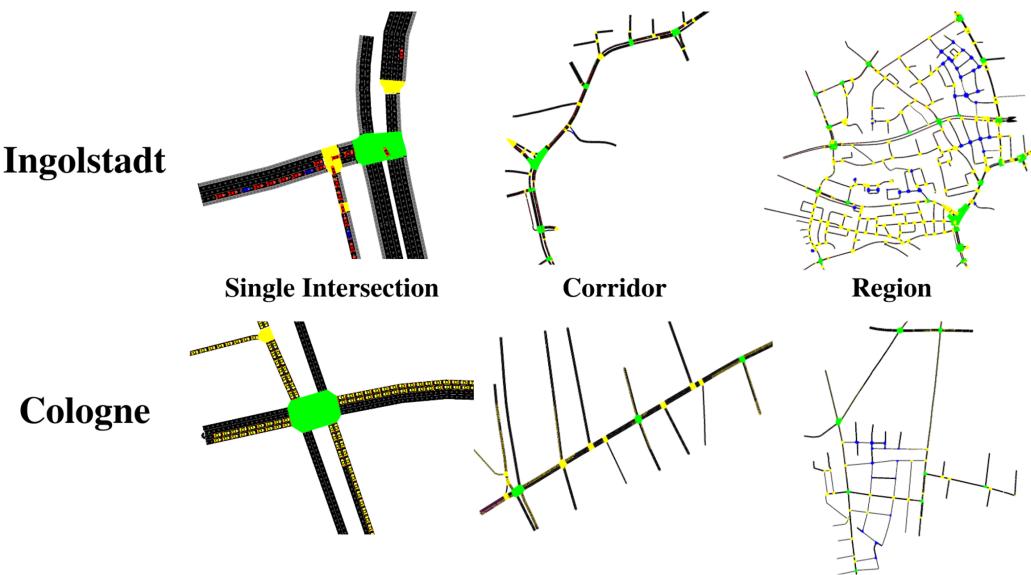
Trong khóa luận này, chúng tôi sử dụng giao diện OpenAI GYM [6] để hỗ trợ cho việc triển khai các thuật toán Học tăng cường và chọn bộ mô phỏng SUMO làm môi trường hoạt động và đánh giá độ hiệu quả của các bộ điều khiển Học tăng cường đối với bài toán Điều khiển tín hiệu đèn giao thông.

SUMO là một phần mềm mô phỏng lưu lượng giao thông mã nguồn mở cung cấp những mô phỏng về các tình huống giao thông trong thế giới thực và cho phép thực hiện các hành động thay đổi tín hiệu dựa trên các phương pháp điều khiển khác nhau thông qua giao diện điều khiển giao thông (TraCI). Trong phạm vi khóa luận này, chúng tôi sử dụng hai mạng lưới giao thông được trích xuất từ hai thành phố trong thế giới thực Cologne và Ingolstadt theo các ba cấp độ được minh họa trong hình 4.1 gồm:

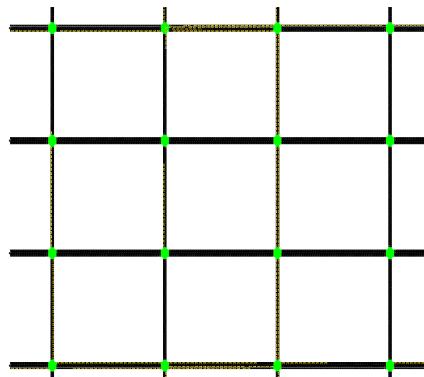
- Bản đồ đơn giao lộ (Single Intersection).
- Bản đồ có nhiều giao lộ nằm dọc theo một tuyến đường huyết mạch (Corridor).
- Bản đồ có nhiều giao lộ trong một khu vực lớn (Region).

Bên cạnh đó, chúng tôi còn tiến hành những thực nghiệm của mình trên bản đồ được thiết kế theo dạng mạng lưới có kích thước 4×4 với tất cả 16 ngã tư được minh họa như hình 4.2.

Tất cả các mạng lưới giao thông được chúng tôi sử dụng trong phần thực nghiệm này bao gồm đầy đủ các cấu trúc giao lộ khác nhau gồm ngã ba, ngã tư, ngã năm .v.v như trong hình 4.3.



HÌNH 4.1: Mô phỏng bản đồ của hai thành phố Ingolstadt và Cologne dựa trên 3 cấp độ. Những vùng màu xanh đánh dấu các giao lộ.



HÌNH 4.2: Mô phỏng bản đồ được thiết kế theo dạng lưới. Những vùng màu xanh đánh dấu các giao lộ.

Bộ mô phỏng SUMO được cộng đồng những chuyên gia trong lĩnh vực giao thông đánh giá cao và đã được sử dụng nhiều trong những nghiên cứu khác nhau đối với lĩnh vực này [18] [17] nhờ sự đa dạng trong các tình huống giao thông, giao diện hợp lý có sẵn và thêm nhiều các chỉ số để đánh giá tình trạng giao thông tại các giao lộ, thứ sẽ được trình bày ở phần tiếp theo.



Ngã ba **Ngã tư** **Ngã năm**

HÌNH 4.3: Minh họa các giao lộ với cấu trúc khác nhau.

4.2 Các chỉ số đánh giá

Trong phần này, chúng tôi sẽ trình bày các chỉ số đánh giá được dùng làm hàm điểm thưởng để đánh giá độ hiệu quả của các bộ điều khiển Học tăng cường và ý nghĩa của các chỉ số này trong vấn đề giao thông trong thực tế.

Đầu tiên, giống với đa số các công trình nghiên cứu liên quan trước đó, chúng tôi cũng áp dụng chỉ số **queue length** làm điểm thưởng để đánh giá độ hiệu quả. Chỉ số queue length của một giao lộ **thể hiện tổng tất cả các phương tiện trên các làn đường đi vào của giao lộ đó**. Mặc dù tiêu chí này đơn giản và phản ánh được mức độ ùn tắc tại các giao lộ, tuy nhiên nó không làm rõ được lợi ích của tác vụ điều khiển tín hiệu đối với thời gian di chuyển của các phương tiện.

Để giải quyết vấn đề này, chúng tôi đã đề xuất thêm 3 chỉ số đánh giá khác được trích xuất từ thông tin của các phương tiện bao gồm:

- **duration:** là thời gian một phương tiện hoàn thành quãng đường di chuyển của nó từ khi xuất hiện trong bộ mô phỏng cho đến khi thoát ra khỏi bộ mô phỏng.
- **waitingTime:** tổng thời gian chờ của một phương tiện khi xuất hiện trên bộ mô phỏng (được tính là khoảng thời gian mà tốc độ của phương tiện đó không lớn hơn 0.1 m/s).

- **delays:** tổng thời gian trễ của tất cả các xe khi tiếp cận giao lộ. (bao gồm thời gian mà xe phải chờ trước khi bắt đầu tiếp cận giao lộ và thời gian mà xe di chuyển dưới tốc độ lý tưởng khi băng qua giao lộ do phải đi chậm lại).

Nếu tất cả các chỉ số trên càng thấp thì chứng tỏ độ hiệu quả của tác nhân được sử dụng càng tốt.

Lưu ý rằng trong cấu hình của bộ mô phỏng SUMO có lưu trữ những thông tin của tất cả các phương tiện xuất hiện trên bản đồ. Thông tin này bao gồm thời gian khởi hành của mỗi phương tiện, quá trình di chuyển của chúng trên khu vực mô phỏng và thời gian mà các phương tiện thoát ra khỏi bộ mô phỏng. Thông tin được ghi lại ngay khi phương tiện đến được đích đến của chúng và không xòn xuất hiện trên bản đồ.

Ngoài ra, SUMO còn cung cấp một bộ cảm biến tại các giao lộ với bán kính được người dùng tự thiết lập để xác định tình trạng giao thông tại những khu vực này.

4.3 Các thiết lập thực nghiệm

Để quá trình huấn luyện và đánh giá được ổn định, chúng tôi đã thiết lập một số đặc điểm chung cho môi trường như *khoảng thời gian dành cho tín hiệu vàng là 3 giây* và *mỗi bước thời gian trong quá trình huấn luyện tương ứng với 10 giây*. Đây là hai đặc tính mà bộ mô phỏng SUMO cho phép người dùng được tùy ý điều chỉnh.

Bên cạnh đó, đối với những giao lộ không có tín hiệu trong bản đồ thì SUMO sẽ kích hoạt cơ chế điều khiển mặc định qua những giao lộ này. Ngoài ra, tốc độ di chuyển tối đa cho phép của một phương tiện là 90 km/h, tốc độ tăng tốc và giảm tốc tối đa của phương tiện được thiết lập mặc định lần lượt là: 1 m/s^2 và 5 m/s^2 . Các giả định cảm biến của SUMO được thiết lập là khả dụng trong bán kính là 200m.

Tổng kết lại, chúng tôi sẽ tiến hành thực nghiệm đối với 5 bộ điều khiển Học tăng cường được trình bày ở phần 3.2 trên 8 bản đồ như hình 4.1 và 4.2, trong đó bản đồ dạng mạng lưới có 2 biến thể:

- grid4x4: các phương tiện được phép di chuyển trên tất cả các làn đường

- arterial4x4: các phương tiện chỉ được phép trên những làn đường chính (theo như trong hình 4.2 thì các phương tiện không được phép di chuyển trên hai đường ngang ở giữa).

Mỗi thuật toán được huấn luyện qua **5 random seeds**, mỗi random seeds sẽ bao gồm **100 episodes** và độ hiệu quả được so sánh với các bộ điều khiển được trình bày ở phần 3.1 dựa trên 4 chỉ số đánh giá điểm thưởng được trình bày ở phần 4.2. Trong đó, các chỉ số này sẽ được lấy trung bình qua 5 seeds.

Thiết lập các siêu tham số		
Siêu tham số	Giá trị	Ý nghĩa
γ	0.99	Hệ số chiết khấu
α	0.0005	Hệ số học của mạng
B	32	Batch_size
U	1	Số lần cập nhật trên mỗi batch
target_update	500	Thời gian cập nhật mạng mục tiêu
K	50,000	Kích thước bộ lưu trữ trải nghiệm

BẢNG 4.1: Bảng các siêu tham số của hai thuật toán DQN và DoubleDQN

Thiết lập các siêu tham số		
Siêu tham số	Giá trị	Ý nghĩa
γ	0.99	Hệ số chiết khấu
α_A	0.0005	Hệ số học của mạng Actor
α_C	0.0005	Hệ số học của mạng Critic
β	0.01	Trọng số điều hướng Entropy
ϵ	0.2	Hệ số xác định vùng lân cận
Epochs	10	Số epoch
B	64	Batch_size
U	1	Số lần cập nhật trên mỗi batch

BẢNG 4.2: Bảng các siêu tham số của hai thuật toán PPO

Cuối cùng, các bộ điều khiển được huấn luyện trên cùng không gian trạng thái như nhau, không gian hành động như nhau và số lần huấn luyện cũng như nhau. Riêng các thuật toán Học tăng cường chịu ảnh hưởng bởi các siêu tham số

được thiết lập mặc định xuyên suốt quá trình thực nghiệm. Bảng 4.1 trình bày các siêu tham số mặc định cho hai thuật toán DQN và DoubleDQN, bảng 4.2 trình bày các siêu tham số của thuật toán PPO.

4.4 Kết quả thực nghiệm

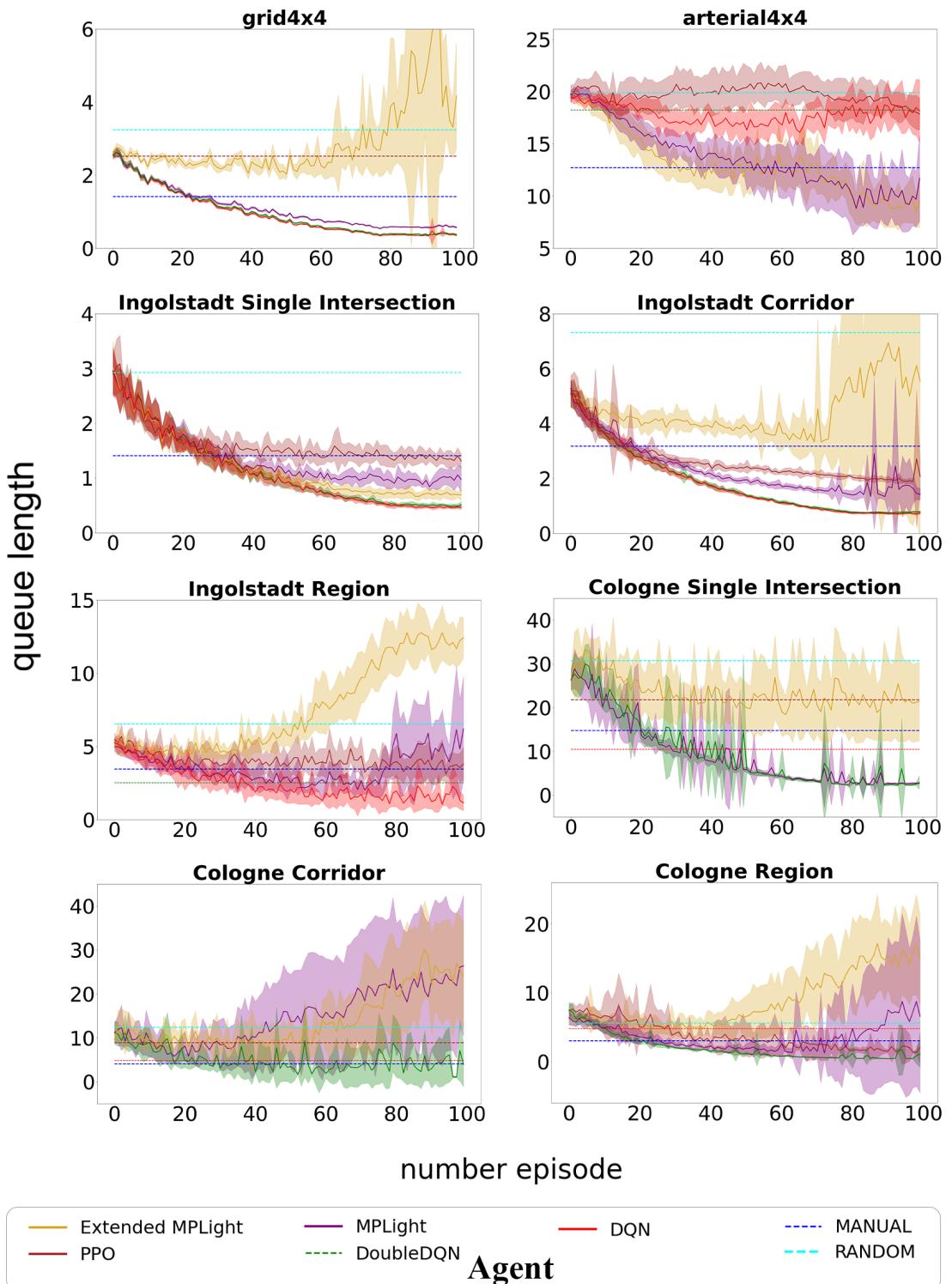
Hình 4.4, hình 4.5, hình 4.6 và hình 4.7 minh họa độ hiệu quả của bộ điều khiển học tăng cường so với bộ điều khiển cơ bản dựa trên lần lượt 4 chỉ số: **queue length**, **delays**, **duration** và **waiting time** đối với bài toán Điều khiển tín hiệu đèn giao thông. Kết quả được thể hiện trên 8 tình huống giao thông khác nhau được trình bày ở phần 4.1.

Nhìn chung, các chiến lược được tìm ra bởi các thuật toán Học tăng cường hầu hết đều cho thấy độ hiệu quả tốt hơn các chiến lược ngẫu nhiên. Tuy nhiên, vẫn có một số ngoại lệ, đặc biệt là đối với Extended MPLight. Cụ thể, độ hiệu quả của phiên bản MPLight mở rộng cho ra kết quả tệ hơn chiến lược ngẫu nhiên trên các bản đồ Cologne có nhiều giao lộ (bao gồm Cologne Corridor và Cologne Region) và bản đồ Ingolstadt Region (bao gồm 21 giao lộ), trong khi phiên bản MPLight cũng cho ra kết quả tệ hơn chiến lược ngẫu nhiên trên tình huống giao thông Cologne Corridor. Điều này là do trong bản đồ Cologne Corridor (gồm tất cả 3 giao lộ) có chứa một ngã năm, điều này làm cho tính bất biến với các tình huống lật xoay vốn là ưu điểm của MPLight trở nên không hiệu quả, cùng với việc các phương tiện được giả lập di chuyển qua giao lộ này khá nhiều (theo quan sát của chúng tôi) đã dần đến kết quả tệ của hai phiên bản MPLight trên tình huống giao thông Cologne Corridor.

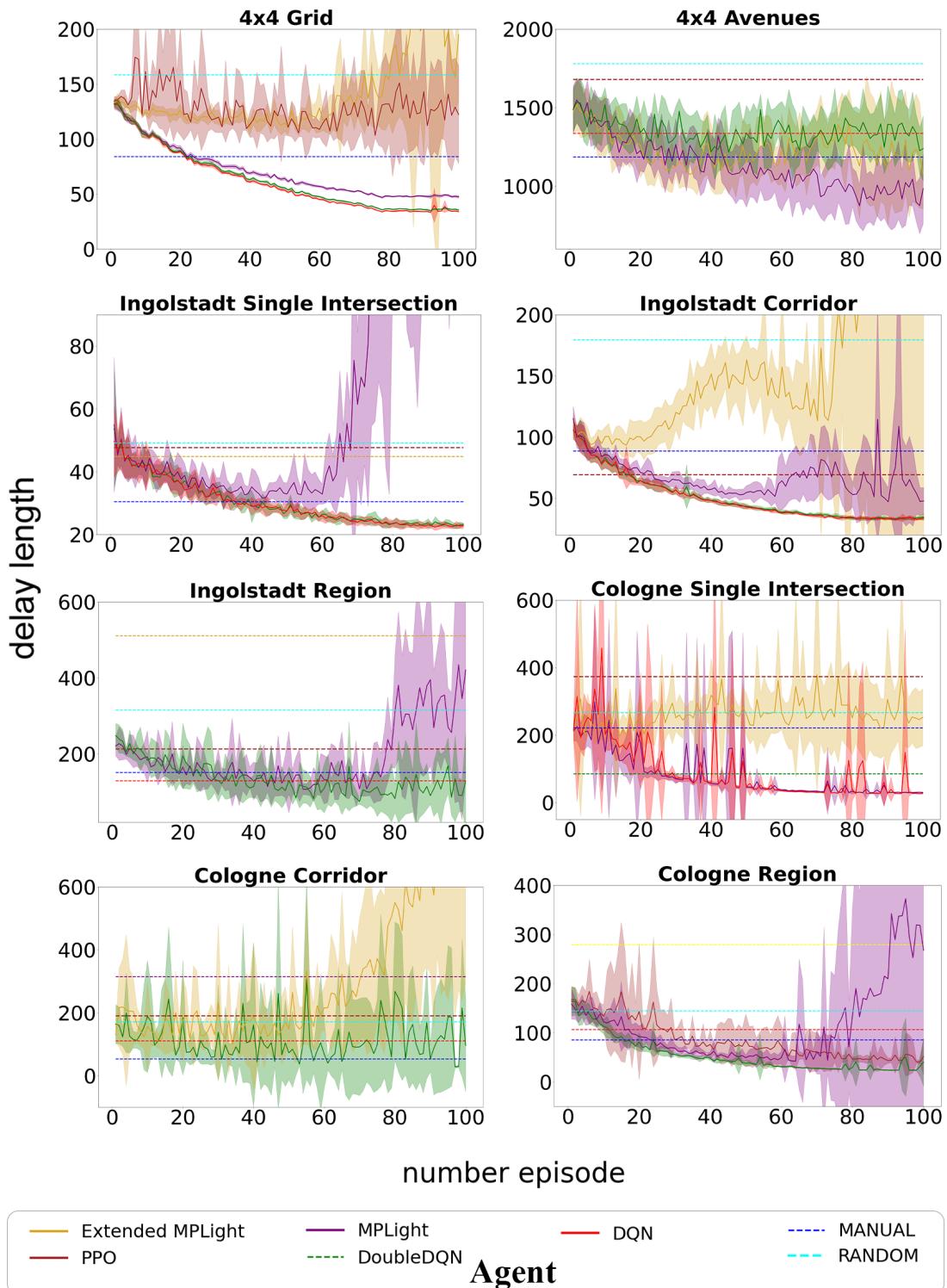
Ngoài ra, các chiến lược học tăng cường còn cho thấy độ hiệu quả tốt hơn khi so sánh với các chiến lược được thiết kế thủ công. Đặc biệt, độ hiệu quả của hai thuật toán DQN và DoubleDQN là vượt trội so với chiến lược tự thiết kế trên hầu hết các tình huống giao thông, ngoại trừ Cologne Corridor. Điều này nói lên rằng đối với các thuật toán Học tăng cường tỏ ra không hiệu quả trong việc điều khiển giao thông tại ngã năm. Đối với PPO, thuật toán này chỉ cho độ hiệu quả tốt hơn chiến lược thủ công trên hai bản đồ mô phỏng là: Ingolstadt Corridor và Cologne Region (chứa lần lượt 7 và 8 giao lộ). Bên cạnh đó, đối với phiên bản MPLight thông thường độ hiệu quả thấp hơn chiến lược ngẫu nhiên khi áp dụng trên các

bản đồ: Ingolstadt Single Intersection, Cologne Corridor và Cologne Region và nguyên nhân thì như đã đề cập ở trên là do trong các tình huống giao thông mô phỏng này có chứa các ngã ba và ngã năm, trong khi MPLight chỉ hoạt động tốt trên giao lộ là ngã tư. Điều này có thể được minh chứng rõ khi nhìn vào kết quả tại hai bản đồ Grid4x4 và Arterial4x4, các bản đồ có hình dạng mạng lưới và chỉ bao gồm các ngã tư. Tuy nhiên, các chiến lược học được từ phiên bản MPLight mở rộng (Extended MPLight) thì đều cho ra kết quả tệ hơn các thiết kế thủ công trên tất cả các bản đồ mô phỏng tương ứng.

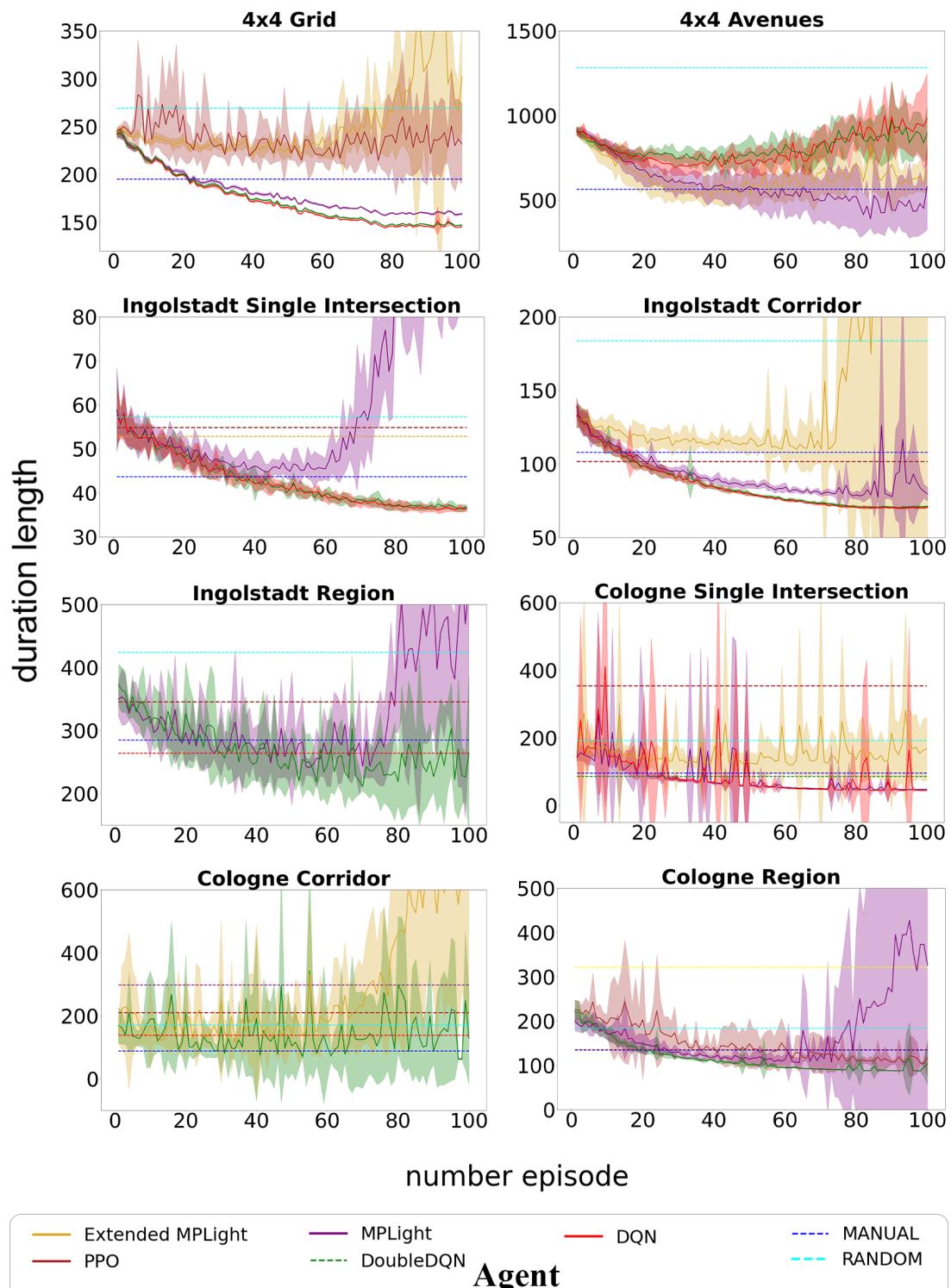
Cuối cùng, dựa vào bảng 4.3, chúng tôi có thể đưa ra những so sánh về hiệu suất giữa các thuật toán Học tăng cường đối với bài toán Điều khiển tín hiệu đèn giao thông trên tất cả các bản đồ mô phỏng, dựa trên 4 số liệu được báo cáo là: **queue length, delays, duration, waiting time**. Theo đó, trên tình huống Arterial4x4 thì Extended MPLight bản mở rộng cho hiệu suất queue length tốt nhất, trong khi ở ba chỉ số còn lại thì MPLight lại cho kết quả tốt hơn. Ở tất cả các tình huống còn lại thì thuật toán DQN đều cho thấy sự vượt trội về mặt hiệu suất so với các thuật toán khác. Mặt khác, hiệu suất của thuật toán Double DQN hầu như chênh lệch không quá nhiều so với thuật toán DQN và hiệu suất của thuật toán này cũng là tương đối tốt. Bên cạnh đó, hiệu suất của thuật toán PPO là không ổn định và không được tốt khi đem so sánh với MPLight và DQN. Đối với hai phiên bản của MPLight, trong hầu hết các tình huống giao thông thì việc thêm thông tin cảm biến vào hàm trạng thái (tức Extended MPLight) không mang lại nhiều lợi ích và thậm chí dẫn đến hiệu suất đi xuống trong nhiều trường hợp và ngược lại trong một số tình huống mà bản đồ giao thông đơn giản như Ingolstadt Single Intersection hoặc Arterial4x4 thì việc thêm thông tin cảm biến giúp hội tụ về chiến lược tốt nhất nhanh hơn.



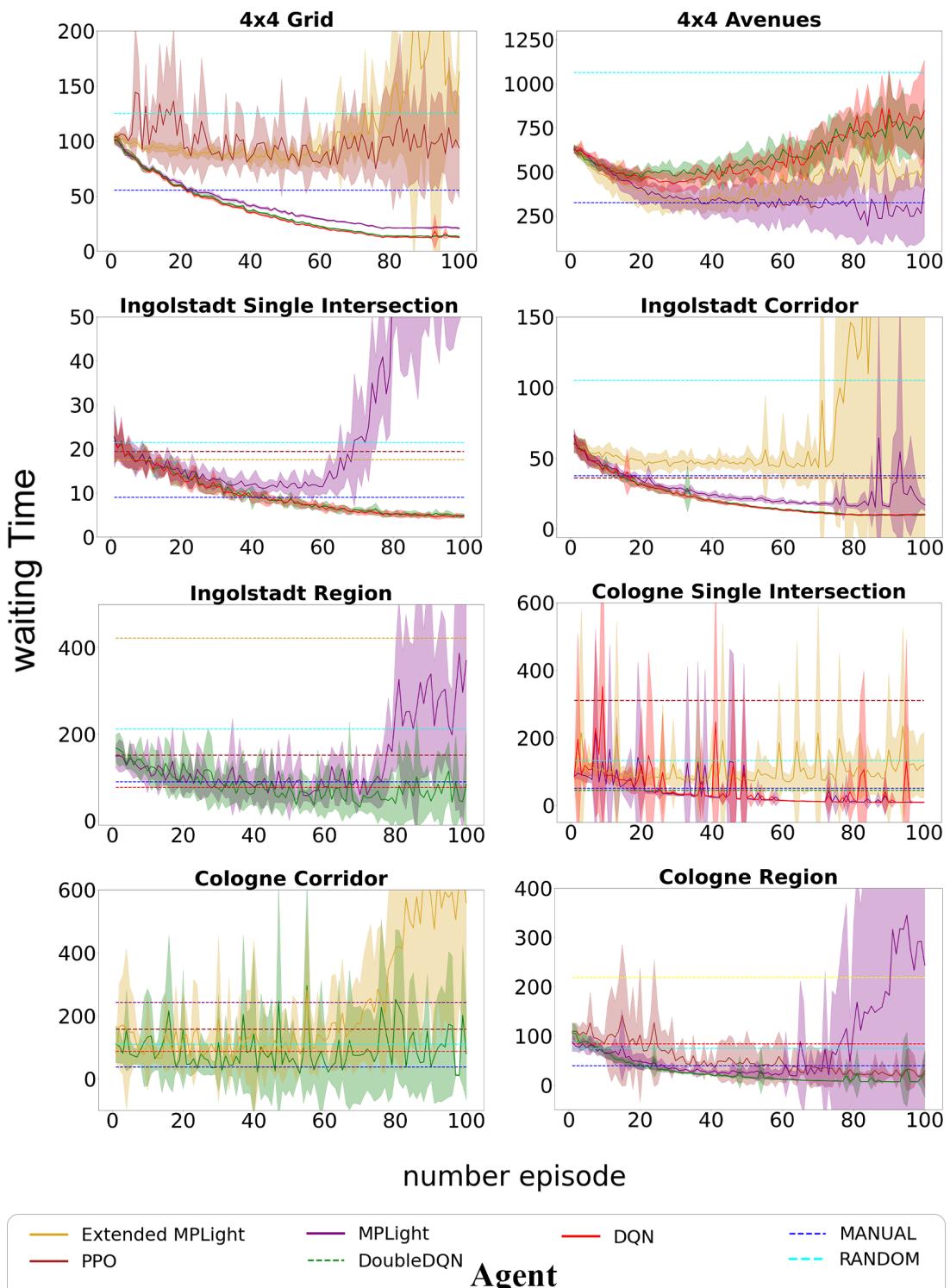
HÌNH 4.4: Chỉ số **queue length** trung bình sau 10 random seeds trên 8 bản đồ khác nhau. Trục x: thể hiện số lần huấn luyện trong 1 seeds. Các đường nét đứt là đường trung bình được lấy trên 100 lần huấn luyện.



HÌNH 4.5: Chỉ số **delays** trung bình sau 10 random seeds trên 8 bản đồ khác nhau. Trục x: thể hiện số lần huấn luyện trong 1 seeds.Các đường nét đứt là đường trung bình được lấy trên 100 lần huấn luyện.



HÌNH 4.6: Chỉ số **duration** trung bình sau 10 random seeds trên 8 bản đồ khác nhau. Trục x: thể hiện số lần huấn luyện trong 1 seeds. Các đường nét đứt là đường trung bình được lấy trên 100 lần huấn luyện.



HÌNH 4.7: Chỉ số **waiting time** trung bình sau 10 random seeds trên 8 bản đồ khác nhau. Trục x: thể hiện số lần huấn luyện trong 1 seeds.Các đường nét đứt là đường trung bình được lấy trên 100 lần huấn luyện.

	Queue Length				
	MPLight	MPLight*	DQN	DoubleDQN	PPO
Arterial4x4	8.8	8.5	16.1	16.89	17.86
Grid4x4	0.54	2.03	0.34	0.26	1.82
Ing.Single	0.84	6.67	0.46	0.49	1.31
Ing.Corr.	1.34	3.32	0.71	0.74	1.89
Ing.Reg.	2.02	4.48	1.04	1.2	3.11
Col.Single	2.54	18.38	2.21	2.27	13.46
Col.Corr.	5.8	7.04	0.95	1.06	4.59
Col.Reg.	1.28	4.6	1.57	0.46	1.37
	Delays				
	MPLight	MPLight*	DQN	DoubleDQN	PPO
Arterial4x4	875.3	1074.3	1191.7	1217.6	1515.8
Grid4x4	46.9	110.2	33,5	35	103.3
Ing.Single	31.6	36.5	22.3	22.7	36.6
Ing.Corr.	47.2	93.5	32.7	32.6	55.2
Ing.Reg.	106.5	189.8	72.3	74.8	162.7
Col.Single	29.6	197.5	26.7	27	136.8
Col.Corr.	79.8	100	26.8	28.3	71.3
Col.Reg.	42	107.5	35.9	23.7	40.6
	Duration				
	MPLight	MPLight*	DQN	DoubleDQN	PPO
Arterial4x4	389.7	517.7	691.9	709.8	976.7
Grid4x4	157.2	220.4	144.3	146.1	214.6
Ing.Single	44.8	47.3	36	36.4	48.4
Ing.Corr.	77.2	109.8	69.8	70	89.8
Ing.Reg.	241.4	320.3	208.3	211.3	297.7
Col.Single	46.7	119.9	44.2	44.6	121.7
Col.Corr.	106.1	111	61.4	62.7	98.1
Col.Reg.	106.3	166.8	70.3	87.8	105.8
	Waiting Time				
	MPLight	MPLight*	DQN	DoubleDQN	PPO
Arterial4x4	191.7	307.6	430.6	468.4	704.7
Grid4x4	20.6	80.1	11.9	12.9	74.6
Ing.Single	11.1	13.2	4.5	4.7	13.9
Ing.Corr.	15.2	43.2	9.6	9.7	26.4
Ing.Reg.	57.8	123.9	29.4	31.5	106.4
Col.Single	9.2	70	8.6	8.8	71
Col.Corr.	50.3	53.7	10.4	11.5	43.9
Col.Reg.	18.7	62.7	6.7	6.8	18.1

BẢNG 4.3: Bảng hiệu suất tốt nhất của các thuật toán Học tăng cường. MPLight* là ký hiệu của Extended MPLight.

Chương 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong chương này chúng tôi rút ra một số kết luận từ việc nghiên cứu các phương pháp Học tăng cường cho bài toán Điều khiển tín hiệu giao thông và những kết quả đạt được qua quá trình thực nghiệm. Bên cạnh đó, chúng tôi sẽ nêu ra các hướng phát triển mà chúng tôi đề xuất nhằm giúp cải thiện kết quả tốt hơn trong tương lai.

5.1 Kết luận

Trong khóa luận này, chúng tôi đã tìm hiểu và áp dụng các thuật toán Học tăng cường cho bài toán Điều khiển tín hiệu giao thông dựa trên bộ mô phỏng SUMO. Đồng thời, chúng tôi cũng so sánh độ hiệu quả của các chiến lược tìm được bởi các thuật toán Học tăng cường với các chiến lược ngẫu nhiên và các chiến lược được thiết kế thủ công.

Kết quả đạt được cho thấy hai thuật toán DQN và Double DQN cho ra hiệu suất tốt trên hầu như tất cả các tình huống mô phỏng ngoại trừ một tình huống giao thông phức tạp, chẳng hạn như có nhiều phương qua lại tại một ngã năm. Trong khi đó, thuật toán PPO cho ra kết quả không ổn định và tùy từng tình huống kết quả có thể tốt hoặc tệ hơn chiến lược được thiết kế thủ công và đối với thuật toán này, chúng tôi nhận thấy rằng thuật toán này cần được trải qua quá trình huấn luyện với thêm nhiều episode hơn [5] [12]. Bên cạnh đó, phiên bản MPLight cũng cho thấy sự hiệu quả trên các tình huống giao thông đơn giản và đạt được hiệu quả cực kỳ cao trên bản đồ chỉ bao gồm các ngã tư và bản mở rộng

của MPLight với cải tiến là thêm các thông tin cảm biến vào trạng thái chỉ hoạt động tốt hơn phiến bản thông thường trên bản đồ đơn giao lộ. Tuy nhiên, chiến lược được thiết kế bởi các thuật toán Học tăng cường hầu hết đều tốt hơn chiến lược được thiết kế ngẫu nhiên.

5.2 Hướng phát triển

Từ các kết quả thu được qua quá trình thực nghiệm và đánh giá, chúng tôi nhận thấy rằng bài toán này còn có thể phát triển nhiều hơn trong tương lai để cải thiện hiệu suất của các thuật toán Học tăng cường:

- Tìm hiểu và áp dụng thêm nhiều thuật toán Học tăng cường khác để tìm ra các thuật toán hiệu quả hơn cho tác vụ này.
- Đầu tư thêm nhiều tài nguyên để tăng thời gian huấn luyện các thuật toán tốt và đa dạng hướng tiếp cận, chiến lược khác nhau để tối ưu hóa độ hiệu quả của các thuật toán.
- Tìm kiếm thêm nhiều giải pháp để cải thiện hiệu suất của các thuật toán hiện tại.
- Tiến hành thực nghiệm trên nhiều tình huống giao thông phức tạp hơn, đặc biệt là các bản đồ có ngã năm.

Bibliography

- [1] Joshua Achiam et al. *Constrained Policy Optimization*. 2017. DOI: 10.48550/ARXIV.1705.10528. URL: <https://arxiv.org/abs/1705.10528>.
- [2] Fadi AlMahamid and Katarina Grolinger. “Reinforcement Learning Algorithms: An Overview and Classification”. In: *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2021. DOI: 10.1109/ccece53047.2021.9569056. URL: <https://doi.org/10.1109/2Fccece53047.2021.9569056>.
- [3] Aqeel Anwar and Arijit Raychowdhury. *Autonomous Navigation via Deep Reinforcement Learning for Resource Constraint Edge Nodes using Transfer Learning*. 2019. DOI: 10.48550/ARXIV.1910.05547. URL: <https://arxiv.org/abs/1910.05547>.
- [4] Sahar Araghi et al. “Q-learning method for controlling traffic signal phase time in a single intersection”. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. 2013, pp. 1261–1265. DOI: 10.1109/ITSC.2013.6728404.
- [5] James Ault and Guni Sharon. “Reinforcement Learning Benchmarks for Traffic Signal Control”. In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Ed. by J. Vanschoren and S. Yeung. Vol. 1. 2021. URL: <https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/f0935e4cd5920aa6c7c996a5ee53a70f-Paper-round1.pdf>.
- [6] Greg Brockman et al. *OpenAI Gym*. 2016. DOI: 10.48550/ARXIV.1606.01540. URL: <https://arxiv.org/abs/1606.01540>.
- [7] Chacha Chen et al. “Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control”. In: *AAAI Conference on Artificial Intelligence*. 2020.

- [8] Yit Kwong Chin et al. "Exploring Q-Learning Optimization in Traffic Signal Timing Plan Management". In: *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*. 2011, pp. 269–274. DOI: 10.1109/CICSSyN.2011.64.
- [9] Kurt M. Dresner and Peter Stone. "A Multiagent Approach to Autonomous Intersection Management". In: *J. Artif. Intell. Res.* 31 (2008), pp. 591–656.
- [10] Liben Huang and Xiaohui Qu. "Improving traffic signal control operations using proximal policy optimization". In: *IET Intelligent Transport Systems* (Oct. 2022), n/a–n/a. DOI: 10.1049/itr2.12286.
- [11] P B Hunt et al. "THE SCOOT ON-LINE TRAFFIC SIGNAL OPTIMISATION TECHNIQUE". In: *Traffic engineering and control* 23 (1982).
- [12] Bálint Kővári et al. "Traffic Signal Control via Reinforcement Learning for Reducing Global Vehicle Emission". In: *Sustainability* 13.20 (2021). ISSN: 2071-1050. DOI: 10.3390/su132011254. URL: <https://www.mdpi.com/2071-1050/13/20/11254>.
- [13] Min Lin, Qiang Chen, and Shuicheng Yan. *Network In Network*. 2013. DOI: 10.48550/ARXIV.1312.4400. URL: <https://arxiv.org/abs/1312.4400>.
- [14] P. Mirchandani and Fei-Yue Wang. "RHODES to intelligent transportation systems". In: *IEEE Intelligent Systems* 20.1 (2005), pp. 10–15. DOI: 10.1109/MIS.2005.15.
- [15] Tong Pham, Tim Brys, and Matthew Taylor. "Learning Coordinated Traffic Light Control". In: Jan. 2013.
- [16] A.G. Sims and K.W. Dobinson. "The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits". In: *IEEE Transactions on Vehicular Technology* 29.2 (1980), pp. 130–137. DOI: 10.1109/T-VT.1980.23833.
- [17] Yaofeng Song et al. *A SUMO Framework for Deep Reinforcement Learning Experiments Solving Electric Vehicle Charging Dispatching Problem*. 2022. DOI: 10.48550/ARXIV.2209.02921. URL: <https://arxiv.org/abs/2209.02921>.

- [18] Qinwen Wang et al. *An Opponent-Aware Reinforcement Learning Method for Team-to-Team Multi-Vehicle Pursuit via Maximizing Mutual Information Indicator*. 2022. DOI: 10.48550/ARXIV.2210.13015. URL: <https://arxiv.org/abs/2210.13015>.
- [19] Huichu Zhang et al. “CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario”. In: *The World Wide Web Conference*. ACM, 2019. DOI: 10.1145/3308558.3314139. URL: <https://doi.org/10.1145%2F3308558.3314139>.
- [20] Rusheng Zhang, Xinze Zhou, and Ozan K. Tonguz. *Using AI for Mitigating the Impact of Network Delay in Cloud-based Intelligent Traffic Signal Control*. 2020. DOI: 10.48550/ARXIV.2002.08303. URL: <https://arxiv.org/abs/2002.08303>.
- [21] Guanjie Zheng et al. *Learning Phase Competition for Traffic Signal Control*. 2019. DOI: 10.48550/ARXIV.1905.04722. URL: <https://arxiv.org/abs/1905.04722>.