

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

ĐỀ CƯƠNG NGHIÊN CỨU

Lê Anh Chiến – 20020634

Đỗ Đức Mạnh – 20020688

Nguyễn Thiên Hảo – 21020906

Bài toán phân nhiệm và tìm đường đi trong kho hàng cho đa robot

Ngành: Kỹ thuật Robot

Môn: Robot phân tán (RBE3045 1)

Giảng viên hướng dẫn: Phạm Duy Hưng

HÀ NỘI, 4/2024

MỤC LỤC

1. Giới thiệu	3
2. Tổng quan tình hình nghiên cứu.....	3
2.1. Bài toán phân nhiệm.....	3
2.2. Bài toán tìm đường đi.....	4
3. Phương pháp nghiên cứu	5
3.1. Bài toán phân nhiệm.....	5
3.1.1. Cấu trúc mạng sử dụng Attention Based Embeddings.....	7
3.2. Bài toán tìm đường	8
3.2.1. MR (Merge and Restart).....	8
3.2.2. PC (Prioritize Conflicts)	9
4. Nội dung nghiên cứu	9
4.1. Mục tiêu tổng quát.....	9
4.2. Mục tiêu thành phần	9
4.2.1. Bài toán phân nhiệm.....	9
4.2.2. Bài toán tìm đường	10
5. Kế hoạch thực hiện	10
5.1. Bài toán phân nhiệm.....	10
5.2. Bài toán tìm đường	10
6. Tài liệu tham khảo	10

BẢNG HÌNH ẢNH

Hình 1: Cấu trúc mạng chính sách	6
Hình 2: CBS thuần.....	8

1. Giới thiệu

Những tiến bộ gần đây về trí tuệ nhân tạo (AI) và robot đã góp phần vào việc sử dụng rộng rãi tự động hóa trong môi trường kho bãi. Một nghiên cứu điển hình gần đây chỉ ra rằng doanh số bán hàng thương mại điện tử tăng hơn 50% sau đại dịch COVID-19 trong bối cảnh thiếu hụt lao động con người, do đó chứng tỏ nhu cầu tự động hóa các trung tâm/kho hàng phân phối và xử lý đơn hàng. Tự động hóa kho hàng, được sử dụng rộng rãi bởi các nhà bán lẻ trực tuyến như Amazon, Alibaba, v.v., có thể được thực hiện bằng cách sử dụng nhiều robot và hệ thống ra quyết định. Chúng ta đang xem xét nhiệm vụ lấy và giao một bưu kiện từ điểm A đến điểm B. Bằng cách thiết kế một hệ thống ra quyết định nhiều robot phù hợp, chúng em có thể cải thiện hiệu quả và thông lượng nhiệm vụ. Vấn đề thiết kế một hệ thống như vậy có thể được chia thành hai phần: phân bổ nhiệm vụ và điều hướng robot. Nói chung, các phương pháp tham lam được sử dụng để giải quyết việc phân bổ nhiệm vụ và lập kế hoạch điều hướng đa robot.

Trong đề cương nghiên cứu này, bọn em sẽ thực hiện hai nhiệm vụ chính là phân nhiệm (task allocation) và tìm đường đi (path planning) cho đa robot trong kho hàng.

2. Tổng quan tình hình nghiên cứu

2.1. Bài toán phân nhiệm

Phân bổ nhiệm vụ cho nhiều robot (MRTA) đã được nghiên cứu rộng rãi trong lĩnh vực robot và các lĩnh vực liên quan như kết nối taxi, giao đồ ăn, tìm kiếm và cứu hộ, v.v. Các vấn đề về MRTA thường được giải quyết bằng cách sử dụng các phương pháp tiếp cận dựa trên thị trường và các phương pháp tiếp cận dựa trên tối ưu hóa cho môi trường khác nhau. Các phương pháp tiếp cận dựa trên thị trường dựa trên hệ thống đấu giá trong đó mỗi robot đặt giá thầu cho một nhiệm vụ với giá bán cơ sở do nhà môi giới xác định. Quá trình đàm phán dựa trên lý thuyết thị trường, trong đó nhóm tìm cách tối ưu hóa chức năng khách quan dựa trên khả năng của robot để thực hiện các nhiệm vụ cụ thể. Các giải pháp này, có thể hiệu quả, mạnh mẽ và có thể mở rộng nhưng khó khăn trong việc thiết kế các hàm chi phí và doanh thu phù hợp. Ngược lại, các phương pháp tiếp cận dựa trên tối ưu hóa vượt trội hơn các phương pháp tiếp cận dựa trên thị trường trong việc xử lý các kịch bản MRTA quy mô lớn (chẳng hạn như 50 nhiệm vụ và 15 robot). Ý tưởng là sử dụng phương pháp tối ưu hóa kết hợp kết hợp tìm kiếm dạng bảng với phương pháp tìm kiếm ngẫu nhiên. Có thể kết hợp tìm kiếm dạng bảng với phương pháp nhiều để giải các tác vụ tĩnh, trong đó các tác vụ không được nhập liên tục mà được xác định trước. Hơn nữa, có nghiên cứu khác đề xuất một cách tiếp cận đồng thời để giải quyết các vấn đề điều hướng và phân bổ nhiệm vụ cho một hệ thống nhiều robot, trong đó các phương pháp mô phỏng và tối ưu hóa đàn kiến được áp dụng. Ngược lại với các cách tiếp cận hiện có được đề cập ở trên, vấn đề của chúng em là phân bổ nhiệm vụ một mặt hàng duy nhất theo tuần tự trong cài đặt kho và chúng em mong muốn phát triển một thuật toán có thể dễ dàng triển khai bằng bất kỳ kỹ thuật điều hướng nào. Trong công việc này, nhiều cách tiếp cận dựa trên đấu giá và tối ưu

hóa tập trung vào các phương pháp tham lam hoặc các sửa đổi của chúng, tạo nên đường cơ sở của chúng em.

Bài toán so khớp taxi: Bài toán phân bổ nhiệm vụ trong công việc này có liên quan chặt chẽ đến bài toán so khớp taxi, trong đó mục tiêu là kết nối tài xế với khách hàng đồng thời giảm thời gian chờ đợi của tài xế, tăng thu nhập của tài xế và giảm quãng đường lái xe. Vấn đề kết hợp nhiệm vụ nhận và giao hàng cho tài xế ban đầu được giải quyết bằng cách sử dụng tối ưu hóa tổ hợp. Gần đây, vấn đề tương tự đã được giải quyết thông qua kỹ thuật RL đa robot, chia bản đồ thành phố thành nhiều vùng điều phối (lưới) và sau đó tìm hiểu chính sách chỉ định trình điều khiển cho một yêu cầu từ điểm lưới (trạng thái) nhất định.

Cả bài toán so khớp taxi và *bài toán MRTA* đều là bài toán ra quyết định tuần tự về mặt phân bổ nhiệm vụ. Tuy nhiên, các nhà kho có bố cục cố định và tương đối nhỏ với số lượng robot được triển khai nhỏ hơn và cố định. Ngược lại, các nền tảng chia sẻ chuyển đi xử lý hàng nghìn phương tiện và hàng triệu đơn đặt hàng trên một khu vực rộng lớn như thành phố. Do đó, kỹ thuật phân bổ đi chung xe tập trung vào việc phân phối nhiệm vụ và robot hơn là tính toán vị trí tọa độ chính xác của chúng.

Do đó, công thức đa robot để phân bổ nhiệm vụ sử dụng học tăng cường cho các ứng dụng so khớp taxi không thể được áp dụng trực tiếp vào cài đặt kho.

Phân bổ nhiệm vụ trong kho: Phân bổ nhiệm vụ trong kho cũng đã được nghiên cứu, trong đó nó được xây dựng dưới dạng chương trình tuyến tính số nguyên và được giải bằng phương pháp xử lý bảng. Các phương pháp này phù hợp nhất với cài đặt tĩnh, trong đó tất cả các rô-bốt đều có mặt cùng lúc để giải quyết vấn đề khớp. Trong thực tế, các robot khác nhau cần thời gian khác nhau để hoàn thành một nhiệm vụ. Do đó, mục tiêu của chúng em ở đây là thiết kế các giải pháp phân bổ nhiệm vụ động để giao nhiệm vụ cho robot bất cứ khi nào nó có sẵn.

Bài toán phân công nhiệm vụ trong kho cũng được nghiên cứu sơ bộ trong bài toán nhận và giao hàng đa robot. Công nghệ tiên tiến nhất hiện nay sử dụng giải pháp tham lam để phân bổ nhiệm vụ. Nó liên tục phân bổ và hủy phân bổ các nhiệm vụ dựa trên chi phí cận biên hoặc sự tiếc nuối về chi phí cận biên. Những kỹ thuật như vậy không thể tách rời khỏi phần điều hướng. Nhưng các tác giả đã đề xuất một giải pháp tách rời, giao nhiệm vụ cho các robot trước tiên theo cách tham lam và sau đó sử dụng chiến lược Tìm kiếm dựa trên xung đột (CBS) để lên kế hoạch cho các đường dẫn không xung đột. Chúng em sử dụng cách tiếp cận tham lam thường được sử dụng này (còn gọi là MPDM) làm một trong những đường cơ sở.

2.2. Bài toán tìm đường đi

Bài toán tìm đường đa robot (Multi-Agent Path Finding- MAPF) được định nghĩa bởi đồ thị $G=(V,E)$ và tập hợp các robot được kí hiệu $a_1 \dots a_k$ trong đó mỗi a_i có điểm khởi đầu $s_i \in V$ và điểm đích $g_i \in V$. Tại mỗi bước lặp, robot có thể di chuyển tới vị trí liền kề hoặc đợi tại vị trí hiện tại. Nhiệm vụ là lập kế hoạch một chuỗi các hành động di chuyển/chờ cho mỗi robot a_i , chuyển nó từ s_i sang g_i sao cho các robot không xung đột, tức là chiếm cùng một vị trí vào cùng một thời điểm, đồng thời nhằm mục đích giảm thiểu hàm chi phí tích lũy.

Trong đề cương nghiên cứu này, chúng em tập trung vào việc tối ưu trong bài toán MAPF, tức là trong đó chi phí của kế hoạch thu được phải được giảm thiểu. Có một loạt các thuật toán giải quyết tối ưu các biến thể khác nhau của MAPF bằng cách sử dụng các kỹ thuật tìm kiếm khác nhau hoặc bằng cách tổng hợp nó thành các bài toán NP-hard đã biết khác. Mỗi ý tưởng đều có ưu và nhược điểm. Thuật toán nào hoạt động tốt nhất trong hoàn cảnh nào là một câu hỏi nghiên cứu mở.

Tìm kiếm dựa trên xung đột (CBS) là một công cụ giải quyết MAPF tối ưu rất hiệu quả. CBS có hai cấp độ. Cấp thấp tìm thấy đường đi tối ưu cho từng robot. Nếu các đường dẫn bao gồm xung đột, thì cấp độ cao, thông qua hành động phân chia sẽ áp đặt các ràng buộc lên các robot xung đột để tránh những xung đột này.

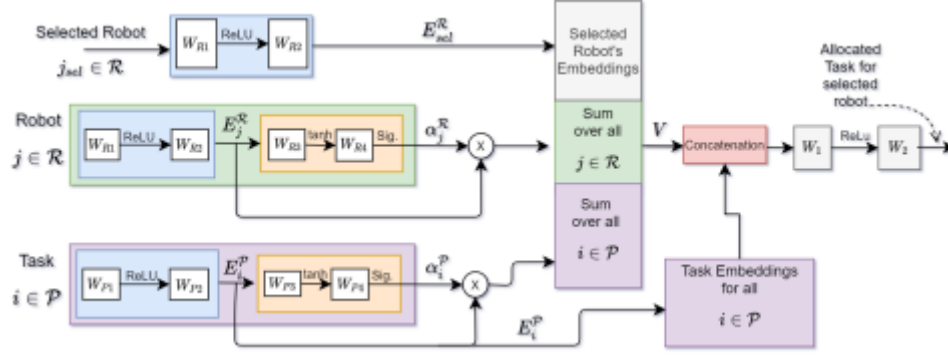
Hai cải tiến cho CBS đã được đề xuất. Đầu tiên, Meta-agent CBS (MA-CBS) tổng quát hóa CBS bằng cách hợp nhất các nhóm nhỏ robot thành siêu robot khi có lợi. Chính sách hợp nhất chính là hợp nhất các robot mà số lượng xung đột được thấy cho đến nay vượt quá ngưỡng B nhất định; để có những lựa chọn tốt về B , MA-CBS sẽ giảm thời gian chạy. Thứ hai, cải tiến bỏ qua (BP) thành (MA)CBS (tức là CBS có hoặc không áp dụng hành động hợp nhất tùy chọn) gần đây đã được đề xuất. (MA)CBS tùy ý chọn đường dẫn ở cấp độ thấp. Khi BP được thêm vào trên (MA)CBS, trước tiên chúng em cố gắng tìm một đường dẫn thay thế (bỏ qua) cho một trong các robot xung đột, do đó tránh được nhu cầu thực hiện phân tách và thêm các ràng buộc mới. (MA)CBS+BP tiếp tục giảm thời gian chạy. Một biến thể của CBS giải quyết MAPF dưới mức tối ưu cũng xuất hiện.

3. Phương pháp nghiên cứu

3.1. Bài toán phân nhiệm

Bài toán phân bổ nhiệm vụ nhiều robot trong môi trường kho được giải quyết bằng các phương pháp tối ưu tiêu chuẩn. Nhưng trên thực tế, các nhiệm vụ xuất hiện ở những thời điểm khác nhau và tính khả dụng của robot cũng thay đổi theo thời gian, điều này khiến cho việc giải quyết vấn đề thông qua các phương pháp tối ưu hóa tiêu chuẩn trở nên khó khăn. Để giải quyết vấn đề này, trong nghiên cứu này, chúng em xây dựng bài toán phân bổ nhiệm vụ nhiều robot trong môi trường kho như một bài toán học tăng cường. Mục tiêu là phân bổ nhiệm vụ cho nhiều robot sao cho nhiệm vụ được thực hiện một cách hiệu quả (tức là các tài nguyên như pin được sử dụng tốt hơn và thông lượng được tối đa hóa).

Việc xây dựng là một thách thức vì nó yêu cầu xác định Quy trình Quyết định Markov (MDP) cho vấn đề.



Hình 1: Cấu trúc mạng chính sách

Quy trình của chúng em lấy tính năng của robot và nhiệm vụ làm đầu vào để tạo ra các phân nhúng có chiều cố định. Tiếp theo, các phân nhúng này được trộn lẫn để tạo ra trạng thái toàn cục cho hệ thống, được sử dụng để thực hiện hành động chọn nhiệm vụ cho robot đã chọn.

Mô hình MDP: Đặt thời gian là $t \in N$ và M robot. Chúng em không giới hạn vào một môi trường cố định mà có thể tổng quát hóa được môi trường kho hàng. Sau khi phân nhiệm tại thời điểm t , nhiệm vụ đó sẽ được loại bỏ khỏi ngăn đợi và nhiệm vụ mới được thêm vào. Chúng em sẽ giả sử rằng tại mỗi thời điểm t , chỉ một con robot cho một nhiệm vụ nhất định. Chúng em sẽ thực hiện bằng việc thêm một nhiễu với phân bố Gauss. Điều này đảm bảo không có hai con robot nào có sẵn tại một điểm. Tất cả các con robot khác sẽ thực hiện nhiệm vụ trước đây và sẽ sang thời điểm $t+1$ ngay khi một con robot hoàn thành nhiệm vụ.

Trạng thái: Mỗi nhiệm vụ sẽ được biểu diễn bởi 4 tham số chính gồm điểm khởi đầu, điểm đích của nhiệm vụ, khoảng cách giữa địa điểm hiện tại với điểm khởi đầu và độ dài của quãng đường phải đi. Mỗi robot được biểu diễn bởi 2 tham số chính là vị trí hiện tại và thời gian còn lại để hoàn thành nhiệm vụ được giao. Vậy trạng thái sẽ bao gồm vị trí của robot, thời gian hoàn thành nhiệm vụ và dãy các nhiệm vụ.

Hành động: Một hành động trong môi trường là việc chọn nhiệm vụ để thực thi từ dãy các nhiệm vụ có sẵn trong ngăn đợi. Chúng em sẽ định nghĩa chính sách với đầu vào là trạng thái hiện tại và đầu ra phân phối của các hành động có thể xảy ra. Trong quá trình huấn luyện mô hình, chúng em sẽ chọn hành động từ phân phối đó.

Phần thưởng: Thiết kế một phần thưởng phù hợp với công thức RL là tối quan trọng bởi đích quyết định cho hành vi thực thi của chính sách. Chúng em xác định phần thưởng là số âm của chi phí, là thời gian robot cần để đạt được điểm gốc của nhiệm vụ từ vị trí hiện tại của nó. Chúng em sử dụng sơ đồ điều hướng để tính toán chi phí gần đúng của

đường đi bằng cách cho robot chạy qua đường dẫn đó. Điều này có ý nghĩa vì trong thời gian di chuyển từ vị trí hiện tại của robot đến nơi xuất phát nhiệm vụ, robot không thực hiện bất kỳ nhiệm vụ nào và do đó chúng em mong muốn giảm thiểu thời gian này. Cụ thể, điều này được định nghĩa là tổng độ trễ di chuyển (TTD), là thời gian tăng thêm mà robot di chuyển khi nó không thực hiện bất kỳ nhiệm vụ nào. Giảm thiểu TTD cuối cùng sẽ giảm thiểu khoảng thời gian tạm thời trong việc phân bổ nhiệm vụ lâu dài.

Chuyển trạng thái: Tại một trạng thái nào đó, chúng em sẽ có một danh sách các nhiệm vụ trong ngăn đợi và một con robot. Chúng em sẽ thực thi hành động được chọn dựa vào chính sách để quyết định nhiệm vụ sẽ được thực thi cho robot đó. Dãy danh sách sẽ loại bỏ nhiệm vụ đó, nhiệm vụ mới được thêm vào và robot sẽ đang không có sẵn. Thuật toán của bọn em sẽ đợi đến khi có sẵn robot để thực thi.

3.1.1. Cấu trúc mạng sử dụng Attention Based Embeddings

Mục tiêu của một bài toán RL là tối đa hóa giá trị mong đợi của phần thưởng tích lũy bởi chính sách π_θ được tham số hóa bởi θ . Hàm mục tiêu có thể biểu diễn là:

$$\max_{\theta} J(\theta) \quad (1)$$

trong đó $J(\theta) = V_{\pi_\theta}(s_0)$. Ở đây, $V_{\pi_\theta}(s_0)$ là hàm giá trị với trạng thái khởi đầu s_0 có biểu thức là:

$$V_{\pi_\theta}(s_0) = E_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s, a_t \sim \pi_\theta(\cdot | s_t) \right]$$

Công thức tại (1) cần dùng đến thuật toán chính sách gradient và được tính là:

$$\nabla_{\theta} J(\theta) = \frac{1}{1 - \gamma} E_t [\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot A_t] \quad (2)$$

Trong đó E_t là trung bình thực nghiệm của một gói mẫu hữu hạn, A_t là giá trị của hàm lợi ích $A_t = Q_{\pi_{\theta_t}}(s_t, a_t) - V_{\theta_t}(s_t)$. Chúng em sẽ sử dụng thuật toán PPO để giải quyết bài toán này bởi sự hiệu quả trong xử lý dữ liệu và thực thi đáng tin cậy.

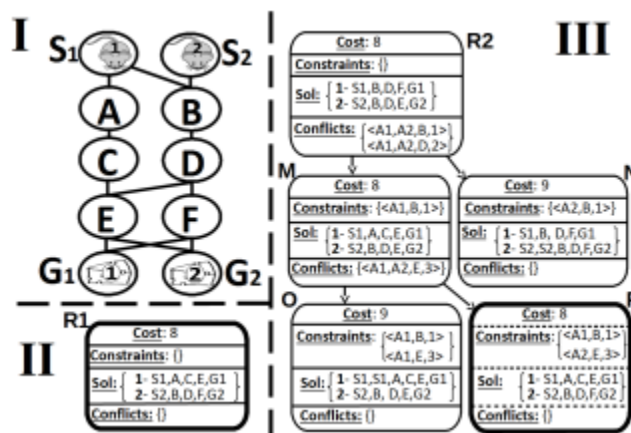
Để đạt được mục tiêu này, phần quan trọng nhất là thiết kế chính sách π_θ , chính sách này chủ yếu được định nghĩa là mạng nơ-ron sâu (DNN) và θ tương ứng với các trọng số của DNN. Việc thiết kế kiến trúc mạng nơ-ron cụ thể cho chính sách là rất quan trọng và đầy thách thức vì nó sẽ kiểm soát lớp chính sách mà chúng ta sẽ tối ưu hóa mục tiêu không lời của (1). Một kiến trúc đơn giản sẽ chỉ đơn giản là ghép nối tất cả thông tin về robot và nhiệm vụ (vị trí, tính khả dụng,...) và chuyển thông tin đó làm đầu vào cho mạng. Kích thước lớp đầu vào sau đó sẽ phụ thuộc vào số lượng robot và nhiệm vụ. Hơn nữa, số lượng tham số có thể huấn luyện sẽ tiếp tục tăng cùng với sự gia tăng số lượng nhiệm vụ

và robot. Loại kiến trúc này không thực tế do các vấn đề về khả năng mở rộng liên quan đến số lượng robot và nhiệm vụ trong môi trường kho nhất định.

Để giải quyết vấn đề về khả năng tăng số lượng robot, chúng em sẽ dùng cơ chế attention được biểu diễn trong Hình 1. Kiến trúc được đề xuất có thể xử lý số lượng robot và nhiệm vụ tùy ý trong bất kỳ bố cục kho nào thông qua việc sử dụng các tính năng nhúng toàn cầu. Trạng thái của robot sẽ làm đầu vào của mô hình để xuất ra những robot rồi cùng với những nhiệm vụ đi qua tiếp mô hình để tìm ra được giá trị cho robot và tác vụ. Tiếp theo, chúng em sẽ tính toán trung bình trọng số của tất cả những robot cũng như những nhiệm vụ. Hai kiểu nhúng này cùng với những của robot được chọn chính là trạng thái của hệ thống.

3.2. Bài toán tìm đường

Trong bài nghiên cứu này, chúng em sẽ đề xuất dùng thuật toán ICBS (Improved-CBS) khi có thêm hai cải tiến so với (MA)CBS+BP.



Hình 2: CBS thuần

3.2.1. MR (Merge and Restart)

Trong một nghiên cứu trước, họ đề xuất MA-CBS. Thuật toán này sẽ hợp nhất hai phần tử xung đột tại cùng một thời điểm thay vì phân tách hành động của CBS thuần.

Nhược điểm của việc hợp nhất là chi phí tính toán lớn của lớp thấp để giảm kích thước cây. Việc này có thể được giải quyết bằng việc yêu cầu một số lượng B lần phân chia trước khi thực hiện hợp nhất. Nhưng, MA-CBS vẫn lặp lại nhiều lần việc tìm kiếm.

Giải pháp cho vấn đề trên được giải quyết bởi MR. Nếu biết trước sẽ có cặp hợp nhất, chi phí tính toán sẽ được giảm thiểu đáng kể bằng cách thực hiện hợp nhất từ đầu, tại nút gốc của cây CT (Conflicted tree). Chúng em đề xuất sử dụng chính sách hợp nhất với sự cân bằng giữa độ đơn giản và tính hiệu quả. Ngay khi một quyết định hợp được thực hiện thì loại bỏ cây CT hiện tại và thực hiện lại (restart) việc tìm kiếm từ node mới tại chỗ những phần tử được hợp nhất từ đầu lúc tìm kiếm.

3.2.2. PC (Prioritize Conflicts)

Ban đầu, BP (bypass) được đề xuất để cải thiện cho lớp thấp trong việc tìm đường tối ưu cho robot. Cụ thể là đối với CBS thuần nhất thì việc chọn lựa nhánh để đi tiếp có thể dẫn tới việc số node được mở rộng thì lúc này để tránh việc chia nhánh thì có thể đi vòng qua node xung đột nếu đó là một đường vòng hữu ích. Một ví dụ từ hình 2, tình huống trong hình 2.II chỉ có 1 node trong khi hình 2.III sinh ra tận 5 node. Trong trường hợp tệ nhất, việc tìm trước có thể thất bại dẫn đến BP hoạt động như một CBS thuần.

Dễ thấy kể cả khi dùng BP thì thuật toán vẫn rất nhạy cảm với quyết định chia nhánh khi gặp xung đột. Để giải quyết nhược điểm này, chúng em sẽ cần phân biệt 3 loại xung đột và biểu diễn thứ tự ưu tiên.

- Xung đột cardinal: khi thêm điều kiện vào node con đều khiến cả hai đều tăng chi phí.
- Xung đột semi-cardinal: khi thêm điều kiện vào node con khiến một node tăng chi phí còn node còn lại thì không.
- Xung đột non-cardinal: khi thêm điều kiện vào node con khiến cả hai không tăng chi phí.

Khi chạy trên một cây gặp xung đột, nó sẽ kiểm tra loại xung đột. Nếu đó là xung đột cardinal thì chắc chắn sẽ phân chia. Nếu bên cạnh node đang phân chia, node có chi phí tương tự sẽ được chọn và tiếp tục mở rộng. Nếu không có xung đột cardinal thì sự ưu tiên kế tiếp là xung đột semi cardinal với việc chọn node có chi phí ít nhất và cuối cùng là non-cardinal với việc chọn ngẫu nhiên. Một điểm chú ý là BP nên được dùng cho xung đột semi cardinal và non cardinal.

4. Nội dung nghiên cứu

4.1. Mục tiêu tổng quát

- Hoàn thành và mô phỏng bài toán với phương pháp đề ra.
- Đánh giá ưu và nhược điểm của phương pháp đề ra.
- So sánh với một số thuật toán liên quan

4.2. Mục tiêu thành phần

4.2.1. Bài toán phân nhiệm

- Thí nghiệm trên môi trường ngẫu nhiên, theo nhóm và được chỉ định. Chúng em xem xét phương pháp được chỉ định để tạo nhiệm vụ trong đó các khu vực cụ thể để nhận và giao hàng được xác định và các nhiệm vụ được lấy mẫu ngẫu nhiên từ chúng. Trong trường hợp của chúng em, chúng em xác định các khu vực để nhận và giao hàng bằng cách sử dụng phân phối Gaussian và lấy mẫu ngẫu nhiên các nhiệm vụ của chúng em từ các phân phối đó.

- Thí nghiệm trên môi trường kho hàng được thiết lập sẵn.
- Đánh giá khả năng của thuật toán khi tăng số lượng robot.

4.2.2. Bài toán tìm đường

- So sánh PC và MR với CBS+BP và MA-CBS+BP
- Đánh giá khả năng của thuật toán khi tăng số lượng robot.

5. Kế hoạch thực hiện

5.1. Bài toán phân nhiệm

- Phân tích rõ mô hình RL được dùng
- Lập trình cho bài toán và debug
- Chạy các mô hình khác từ các nghiên cứu trước để phân tích

5.2. Bài toán tìm đường

- Phân tích rõ mô hình CBS từ đó tìm hiểu kỹ hơn ICBS
- Lập trình cho bài toán và debug
- Chạy các thuật toán khác từ các nghiên cứu trước để phân tích

6. Tài liệu tham khảo

F. Xue, H. Tang, Q. Su, and T. Li, “Task allocation of intelligent warehouse picking system based on multi-robot coalition,” *KSII Transactions on Internet and Information Systems*, vol. 13, no. 7, 2019.

G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant. Conflict-based search for optimal multiagent path finding. *Artificial Intelligence Journal (AIJ)* , 2015.

G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant. Meta-agent conflict-based search for optimal multi-agent path finding. In *SOCS*, 2012.

E. Boyarski, A. Felner, G. Sharon, and R. Stern. Don’t split, try to work it out : Bypassing conflicts in multi-agent pathfinding. In *ICAPS-2015*, Jerusalem, Israel, June 7-11, 2015.

G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant. Meta-agent conflict-based search for optimal multi-agent path finding. In *SOCS*, 2012.

Eli Boyarski, Ariel Felner, Roni Stern, Guni Sharon, David Tolpin, Oded Betzalel, and Solomon Eyal Shimony. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding. In *IJCAI*, pages 740–746, 2015.

Aakriti Agrawal, Amrit Singh Bedi, and Dinesh Manocha. RTAW: An Attention Inspired Reinforcement Learning Method for Multi-Robot Task Allocation in Warehouse Environments. In *ICRA*, 2023.

J. Holler, R. Vuorio, Z. Qin, X. Tang, Y. Jiao, T. Jin, S. Singh, C. Wang, and J. Ye, “Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem,” in 2019 IEEE International Conference on Data Mining (ICDM). IEEE, 2019, pp. 1090–1095.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in NeuRIPS, 2017.

H. Ma, J. Li, T. Kumar, and S. Koenig, “Lifelong multi-agent path finding for online pickup and delivery tasks,” arXiv preprint arXiv:1705.10868, 2017.