**Slide 1**

Em xin kính thưa hội đồng, các thầy cô và các bạn sinh viên có mặt tại đây hôm nay. Em xin trình bày khóa luận về đề tài “Tích hợp các phép biến hình cấu trúc cây cải tiến suy luận tiến hóa theo tiêu chuẩn parsimony trong MPBoot”.

**Slide 2**

Em xin trình bày 6 phần như sau.

**Slide 3**

Đầu tiên là Giới thiệu, em xin đi qua một số khái niệm chính.

**Slide 4**

Phylogenetic (tiến hóa sinh học) là lĩnh vực nghiên cứu về quá trình tiến hóa và mối quan hệ giữa các loài dựa trên đặc điểm di truyền và hình thái của chúng. Nó giúp xây dựng các cây tiến hóa biểu thị sự đa dạng sinh học và hỗ trợ cho các nghiên cứu khác.

**Slide 5**

Đây là một cây tiến hóa đơn giản (sau đây gọi là cây) mô tả quan hệ tiến hóa giữa mèo, hổ, báo và sư tử. Từ tổ tiên chung nhất của 4 loài, dần dần các đột biến xảy ra tách thành các loài khác nhau.

**Slide 6**

Tiến hóa sinh học được ứng dụng rộng rãi trong nhiều lĩnh vực như y học, địa chất học, khoa học môi trường hay gần đây nhất là dịch tễ học vào các chủng Covid19.

**Slide 7**

Bài toán trung tâm chúng ta quan tâm tới là bài toán xây dựng cây tiến hóa. Bài toán này có đầu vào là các sắp hàng đa chuỗi (gọi là các MSA). Những MSA này có thể là các đoạn DNA của các loài đã được định dạng diễn tả bằng các xâu kí tự với độ dài cố định. Đầu ra sẽ là cây tiến hóa mô tả MSA một cách chính xác nhất. Cây đầu ra có dạng một cây nhị phân có n đỉnh lá chính là n loài được cho ở đầu vào.

**Slide 8**

Phát triển từ bài toán trung tâm trên, em đề cập giải quyết bài toán xây dựng cây tiến hóa bootstrap. Ngoài việc xây dựng cây tối ưu nhất, ta sẽ xây dựng tập cây bootstrap và sử dụng chúng để đánh giá độ tin cậy cho các cạnh bằng phương pháp bootstrap. Tập cây bootstrap được tạo ra từ tối ưu các MSA bootstrap bằng cách sinh ngẫu nhiên các cột từ MSA gốc.

**Slide 9**

Đây là ví dụ việc đánh giá độ tin cậy của một nhánh. Ta thấy nhánh Sư tử với Báo này xuất hiện 2 / 3 lần trong tập bootstrap. Khi đó độ tin cậy của nhánh này sẽ là 2 / 3 hay 67%

**Slide 10**

Ở mô hình chuẩn, sau khi tạo ra các MSA bootstrap sẽ tìm cây tối ưu (hay tree search) với mỗi MSA đó.

**Slide 11**

Mô hình xấp xỉ MPBoot được phát triển từ mô hình Bootstrap chuẩn. Mô hình này tìm các cây bootstrap xấp xỉ tối ưu chỉ qua một lần tree search duy nhất. Nhằm nâng cao hiệu quả của cây chính và tập cây bootstrap, em thực hiện cải tiến công đoạn tree search của mô hình xấp xỉ này.

**Slide 12**

Để đánh giá giữa 2 cây xem cây nào là tốt hơn, chúng ta sử dụng điểm số MP. Điểm số MP của 1 cây là số lượng tối thiểu các đột biến cần thiết để giải thích dữ liệu. Mục tiêu là tìm cây có điểm MP thấp nhất. Bài toán này là bài toán NP-đầy đủ (do số lượng cây nhị phân n lá tăng nhanh theo hàm lũy thừa). Vì thế, phương pháp chung là ta sẽ khám phá lấy mẫu không gian các cây nhị phân n lá, tính điểm MP và chọn cây kết quả là cây có điểm MP thấp nhất.

**Slide 13**

Nhằm lấy mẫu không gian các cây một cách hiệu quả, người ta thường dùng các phép biến đổi cây với tập lân cận từ nhỏ tới lớn bao gồm NNI (tráo đổi hàng xóm gần nhất), SPR (cắt và nối lại cây con) và TBR (phân đôi cây và nối lại).

**Slide 14**

Trên slide.

**Slide 15**

Tiếp theo, em sẽ đi vào đề xuất đầu tiên là tích hợp phép TBR vào MPBoot

**Slide 16**

Trong thuật toán MPBoot gốc, có 3 pha gồm pha khởi tạo, pha khám phá và pha tinh chỉnh. Dễ thấy trong cả 3 pha này SPR được sử dụng trong toàn bộ framework. Do đó, việc thay thế SPR bằng phép TBR toàn diện hơn có thể dẫn đến cây MP và tập bootstrap tốt hơn.

**Slide 17**

Đây là một ví dụ về phép TBR. Ta có ở hình A là cây ban đần T^lst với cạnh cắt R và cạnh nối I\_1, I\_2. Hình B là sau khi cắt cạnh R và hình C là khi nối I\_1 và I\_2 lại thông qua R.

**Slide 18**

Sau khi thực hiện biến đổi bằng TBR, ta cần tính điểm MP của cây nhận được. Tiếp cận trực tiếp là duyệt post-order lại toàn bộ cây và tính theo thuật toán Fitch hoặc Sankoff. Tuy nhiên nhiều cây con không thay đổi cấu trúc (điểm MP) sau một phép TBR. Vì thế ta chỉ cần tính lại điểm của những đỉnh có thay đổi điểm số.

**Slide 19**

Em đề xuất tính toán nhanh một phép biến đổi TBR như sau. Các đỉnh in xanh cần phải tính lại điểm (Các đỉnh trên đường đi từ I\_1 đến I\_2 của T^lst). Sử dụng kĩ thuật đổi gốc như trên, số lượng đỉnh cần phải tính lại điểm giảm xuống O(maxtrav) (giới hạn độ dài 2 cạnh nối) ở hầu hết các trường hợp thay vì phải duyệt toàn bộ cây.

**Slide 20**

Tiếp theo là việc tìm kiếm cây lân cận sử dụng TBR. Cho trước cạnh cắt R, khoảng bán kính [mintrav, maxtrav] (điều kiện về khoảng cách giữa 2 cạnh nối). Em thiết kế 2 chiến lược tìm kiếm lân cận là tìm kiếm tốt nhất và tìm kiếm tốt hơn.

Sau đó cập nhật cây T bằng T^bestNei tìm được.

Thủ tục tìm kiếm lặp lại nếu điểm số MP của cây T vẫn được cải thiện.

**Slide 21**

Về chiến lược tìm kiếm tốt nhất (best), Chi tiết đã được trình bày như ở trên. Để ngắn gọn, với mỗi cạnh cắt R, ta sẽ tìm được cây lân cận tốt nhất T^bestNei của T sau khi xét tất cả cặp cạnh nối I\_1 và I\_2 thỏa mãn.

**Slide 22**

Về chiến lược tìm kiếm tốt hơn (better). Thay vì ở trên với mỗi cạnh cắt R, ta chỉ tìm được duy nhất 1 cây lân cận tốt nhất T^bestNei, thì ở chiến lược BETTER, ta tìm được một cây T^bestNei với mỗi cặp (R, I\_1).

**Slide 23**

Tiếp theo là đề xuất về phiên bản MPBoot2.

**Slide 24**

Đầu tiên, tích hợp TBR vào MPBoot, thay thế toàn bộ leo đồi sử dụng SPR bằng TBR trong MPBoot. Tinh chỉnh bán kính khảo sát và điều kiện dừng.

Một số tính năng mới như:

* Checkpoint cho phép chương trình tiếp tục từ checkpoint cuối cùng đã lưu trong trường hợp bị gián đoạn hoặc gặp sự cố. Tính năng này giúp tránh phải chạy lại toàn bộ phân tích, từ đó tiết kiệm thời gian và tài nguyên tính toán
* Cải thiện xử lý với bộ dữ liệu lớn
* Hỗ trợ dữ liệu morphology, nhị phân

**Slide 25**

Tiếp theo là đề xuất về một phiên bản khác là MPBoot-RL.

**Slide 26**

Ý tưởng là 3 phép NNI, SPR và TBR có các điểm mạnh khác nhau nên kết hợp chúng có khả năng cho kết quả tốt

**Slide 27**

Thuật toán ACO (Ant Colony Optimization) - tối ưu đàn kiến - là phương pháp nghiên cứu lấy cảm hứng từ việc mô phỏng hành vi của đàn kiến trong tự nhiên nhằm mục đích giải quyết các bài toán tối ưu phức tạp trong thực tế. Các cá thể kiến trao đổi thông tin trên đường đi thông qua vết mùi (Pheromone) để lại trên đường đi. Các đường đi có nồng độ mùi ít hơn sẽ được loại bỏ, cuối cùng tất cả đàn kiến sẽ đi trên con đường có khả năng trở thành con đường ngắn nhất từ tổ đến nguồn thức ăn.

**Slide 28**

Cấu trúc đồ thị của MPBoot-RL khá đơn giản. Gồm đỉnh bắt đầu tới 3 đỉnh NNI, SPR và TBR và về đỉnh kết thúc. Khi đó những cạnh sử dụng để lưu vết mùi là những cạnh sau đây.

**Slide 29**

Thông tin heuristic sẽ gồm độ ưu tiên cho 3 phép NNI, SPR và TBR được cho ở tham số đầu vào.

**Slide 30**

Quy tắc cập nhật mùi sử dụng Max-Min trơn như sau:

* Sau 1 thế hệ kiến, gồm một số vòng lặp leo đồi (hay còn gọi là lời giải). Những cạnh e mà thuộc ít nhất 1 lời giải tốt sẽ được cập nhật đến giá trị pheromone max. Ngược lại, nếu không thuộc lời giải tốt nào thì sẽ được cập nhật về giá trị pheromone min.
* Vậy thế nào là 1 lời giải tốt? 1 lời tốt ở đây sẽ là lời giải có điểm MP của cây mới tốt hơn hoặc bằng với điểm MP tốt nhất từ trước đó mà thuật toán đã tìm được. Giả sử nếu trong 1 thế hệ kiến không có lời giải nào thỏa thì sẽ có 1 cơ chế fallback để chọn NNI là lời giải tốt.
* Do sau 1 thế hệ, mỗi cạnh được cập nhật đúng 1 lần nên em đặt tên là SMMAS-once.

**Slide 31**

Ngoài ra, em còn thử nghiệm thêm 1 quy tắc cập nhật khác tương tự. Đó là thay vì mỗi cạnh được cập nhật đúng 1 lần thì gọi k là số lượng lời giải tốt mà cạnh e tham gia. Thì khi cập nhật lên pheromone max sẽ cập nhật k lần. Từ đó phản ánh đúng giá trị mà cạnh e mang lại trong 1 thế hệ kiến. Em đặt tên là SMMAS-multiple

**Slide 32**

Quy trình bước đi ngẫu nhiên thì em sử dụng công thức mặc định. Là tỉ lệ của tích của pheromone và độ ưu tiên heuristics của từng phép biến đổi.

**Slide 33**

Tiếp đến là phần thực nghiệm và kết quả

**Slide 34**

Em so sánh các phương pháp sau đây: (Đọc tiếp trên slide…)

**Slide 35**

Các bộ dữ liệu sử dụng gồm Yule Harding để kiểm tra độ chính xác bootstrap và TreeBASE để kiểm tra điểm MP và thời gian thực thi trên dữ liệu sinh học DNA và protein

**Slide 36**

Gồm 4 hình nhỏ thể hiện số bộ dữ liệu mà từng phương pháp đạt được điểm MP tốt nhất. Hàng trên là DNA, dưới là protein, cột bên trái là chi phí biến đổi đều, cột bên phải là chi phí biến đổi không đồng đều.

Ở các bộ dữ liệu DNA thì TNT là phương pháp tốt nhất nhưng với protein, nhất là chi phí không đồng đều thì TNT thua so với các phương pháp MPBoot. Trong các phương pháp còn lại thì các phiên bản TBR và phiên bản ACO-MUL sẽ tốt hơn

**Slide 37**

Về thời gian thực thi, thấy 2 phương pháp ACO- là 2 trong những phương pháp nhanh nhất ngay cả đối với chi phí biến đổi đồng nhất hay không đồng nhất. Còn với điểm MP (quay lại slide trước), phương pháp ACO-MUL cũng là một trong các phương pháp cho điểm MP tốt nhất. Còn những phương pháp tốt khác thì thời gian thực thi rất là lâu.

**Slide 38**

Về độ chính xác bootstrap, các phiên bản mới cho độ chính xác bootstrap tương đương với MPBoot gốc.

**Slide 39**

Em có khảo sát tỉ lệ sử dụng 3 phép biến đổi cây NNI, SPR và TBR trên 115 bộ dữ liệu TreeBASE với độ khó tương ứng của từng bộ dữ liệu được Pythia dự đoán. Dễ thấy rằng bộ dữ liệu mà Pythia đánh giá càng “khó” (chỉ số càng lớn), thì phép biến đổi SPR và TBR sẽ được sử dụng càng nhiều. Không những thế, trong những bộ dữ liệu khó đó, tỷ lệ tìm kiếm sử dụng phép TBR thường sẽ lớn hơn so với SPR.

**Slide 40**

Đọc trên slide

**Slide 41**

Đọc trên slide

**Slide 42**

Một phần của khóa luận này đã được công bố tại hội nghị KSE.

**Slide 43**

Em xin cảm ơn thầy cô và hội đồng đã theo dõi và lắng nghe.