# TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Nhiều năm gần đây, các bài toán tối ưu ngày càng chứng tỏ được sức mạnh to lớn của nó. Đặc biệt là bài toán vận tải, các bài toán tối ưu đường đi, chi phí được quan tâm rất nhiều vì tính thực tiễn của bài toán. Trong đó có vấn đề đặt kho hàng, nhà máy tại các trung tâm sao cho chi phí vận chuyển là nhỏ nhất. Trong đồ án này, tác giả tập trung nghiên cứu bài toán P – Median. Bài toán này đã được chỉ ra là bài toán thuộc lớp NP- Khó được chỉ ra trên đồ thị tổng quát và nghiên cứu một số cái giải bài. Kết quả thực nghiệm cho thấy một số giải thuật khá hiệu quả đưa ra lời giải cho bài toán song thời gian thực hiện còn tương đối lớn.

Nội dung của đồ án gồm 3 chương:

Chương 1: Bài toán p-median và ứng dụng: Phát biểu bài toán và ứng dụng trong thực tiễn.

Chương 2: Một số thuật toán giải bài toán p-median: Đưa ra một số giải thuật để giải bài toán p – median.

Chương 3: Kết quả thực nghiệm: Mô tả cài đặt thuật toán và hệ thống chương trình. Đưa ra các bộ dữ liệu thực nghiệm và kết quả.

# ABSTRACT OF THESIS

In recent years, optimization problems have increasingly shown its great strength. Especially the transport problem, the optimization of the road, the cost is paid much attention because of the practicality of the problem. There is the problem of placing warehouses, factories at centers so that transportation costs are minimal. In this project, the author focuses on the P - Median problem. This problem has been shown to be problematic in the NP-Difficult class indicated on the generalized graph and studied some of the solutions. Experimental results show that some algorithms are quite effective in solving the problem, but the execution time is relatively large.

The content of the project consists of 3 chapters:

Chapter 1: The p-median problem and application: Speaking the problem and applying it in practice.

Chapter 2: Some algorithms for solving the p-median problem: Give some algorithms to solve the p-median problem.

Chapter 3: Experimental Results: Describes algorithms and system settings. Produce experimental data sets and results.

# Lời cảm ơn

Đầu tiên, em xin được gửi lời cảm ơn chân thành đến thầy Nguyễn Đức Nghĩa, người đã tận tình giúp đỡ em trong quá trình hoàn thiện đồ án này. Thầy là một nhà giáo tâm huyết với những giảng giải kĩ càng, đi sâu vào bản chất vấn đề, cũng như có những gợi ý mở rộng và những lưu ý của thầy cho các vấn đề cần giải quyết. Trong quá trình hoàn thiện cuốn đồ án này đã cho em biết thêm rất nhiều kiến thức bổ ích phục vụ cho quá trình học tập cũng như công tác, làm việc sau này.

Em xin bày tỏ lòng kính trọng và biết ơn đến các thầy cô trong viện Công nghệ thông tin và truyền thông nói chung và bộ môn Khoa học máy tính nói riêng, những người luôn tận tâm giảng dạy, truyền đạt kiến thức và kinh nghiệm sống quý báu cho chúng em. Những kiến thức, kinh nghiệm đó là hành trang vô giá sẽ theo em trong suốt quá trình học tập và làm việc sau này.

Xin gửi lời cảm ơn tới các bạn học chuyên ngành Khoa học máy tính K57, một tập thể hòa đồng, gắn bó luôn giúp đỡ nhau trong quá trình học tập. Nhờ đó mình đã học hỏi được rất nhiều điều để hoàn thiện bản thân.

Cuối cùng, Xin kính chúc mọi người luôn vui vẻ, mạnh khỏe và thành công trong cuộc sống!

Hà Nội, tháng 06 năm 2017

Nguyễn Đức Thịnh

# Lời nói đầu

Ngày nay, các dịch vụ công cộng đang rất phát triển và ngày càng mở rộng theo quy mô dân số. Từ đó có rất nhiều vấn đề phát sinh về việc cung ứng hàng hóa sao cho hợp lý với chi phí nhỏ nhất. Việc xã hội ngày càng phát triển thì nhu cầu của khách hàng với các dịch vụ công cộng ngày càng được gia tăng, phần lớn các trạm cung ứng cần phải đáp ứng một số lượng lớn yêu cầu từ khách hàng vốn luôn luôn tăng lên vì vậy dẫn đến quá tải. Một số loại hình dịch vụ mua bán online rất phát triển, vì vậy vấn đề đặt các nhà kho trong thành phố sao cho chi phí vận chuyển là nhỏ nhất cũng rất được quan tâm bởi xã hội. Đặc biệt hơn nữa một số loại hình dịch vụ mang tính khẩn cấp như cứu thương, cứu hỏa… cũng đặt ra vấn đề sao cho việc đáp ứng kịp thời là nhanh nhất để giảm thiểu thiệt hại.

Các bài toán thực tế trên đều được dẫn về bài toán p-median. Tuy nhiên, bài toán p-median cũng có lịch sử hàng trăm năm nghiên cứu và phát triển. Khởi nguồn của bài toán này bắt đầu từ một câu hỏi của một nhà toán học Pháp Pierre de Fermat được đưa ra từ thế kỉ XVII: Cho một tam giác trên mặt phẳng, hãy tìm một điểm (gọi là trung vị - median) trong mặt phẳng đó sao cho tổng khoảng cách từ điểm đó đến ba đỉnh của tam giác là nhỏ nhất. Đến đầu thế kỉ XX, Alfred Weber đã tổng quát hóa bài toán trên và đưa thêm trọng số vào các đỉnh với số đỉnh n ≥ 3, số trung vị k ≥ 1. Năm 1960, Kakimi phát triển bài toán p-median trên một đồ thị gồm n đỉnh. Bài toán p-median trên đồ thị tổng quát đã được xác định thuộc lớp bài toán toán NP-khó, nghĩa là không có thuật toán hiệu quả để giải bài toán này với thời gian tính đa thức, trừ khi P = NP. Các hướng tiếp cận có tính ứng dụng của bài toán này hầu hết đều thuộc nhóm các giải thuật giải gần đúng.

Trong đồ án tác giả tập trung giải bài toán p-Median theo một số giải thuật được nghiên cứu trước đây để tìm ra được cách giải tối ưu phù hợp với các mô hình dữ liệu cũng như mô hình thực tế khác nhau. Các giải thuật được cài đặt và chạy thử nghiệm trên các bộ dữ liệu mà các nhà nghiên cứu đã xây dựng để so sánh và đánh giá hiệu năng.

# Chương 1: Bài toán toán p-Median và ứng dụng

## 1.1 Phát biểu bài toán:

### 1.1.1 Phát biểu bằng lời:

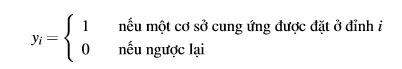
Bài toán thực tế: Trên một địa điểm có các điểm yêu cầu (các khu dân sinh, các nhà máy …) cung ứng các dịch vụ (y tế, cứu hỏa, các nhà máy có thể yêu cầu kho hàng, hay nhà máy nguyên liệu…). Chúng ta cần phải đặt các trạm cung ứng các dịch vụ để phục vụ các điểm yêu cầu sao cho chi phí vận chuyển hay thời gian cung ứng là nhỏ nhất.

### 1.1.2 Mô hình hóa bài toán:

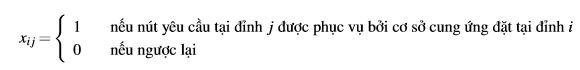
Mô hình hóa bài toán p-median trên đồ thị tổng quát

Cho đồ thị đầy đủ G = (V,E) có n đỉnh. Gọi p là số cơ sở cung ứng cần phải chọn ra từ các đỉnh của đồ thị G. Gọi I là tập các đỉnh của đồ thị mà tại đó đặt các cơ sở cung ứng, J là tập các đỉnh còn lại của đồ thị, chính là vị trí đặt các nút yêu cầu. Đưa vào các biến gán x và biến quyết định y:

* Vị trí đặt cơ sở cung ứng:



* Nút yêu cầu được gán cho cơ sở cung ứng:



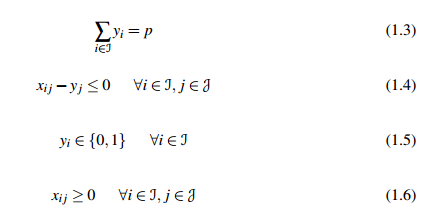
* Số lượng yêu cầu tại nút j là wj . Nếu đặt cơ sở cung ứng tại j thì coi như không có yêu cầu nào tại j, và wj = 0.
* Chi phí đơn vị cij là chi phí để cơ sở cung ứng đặt tại i phục vụ nút yêu cầu tại j.

Như vậy, bài toán p-median có thể được mô tả như là một bài toán tối ưu:



Thỏa mãn các ràng buộc sau:





Trong đó hàm mục tiêu 1.1 tối thiểu tổng chi phí yêu cầu.

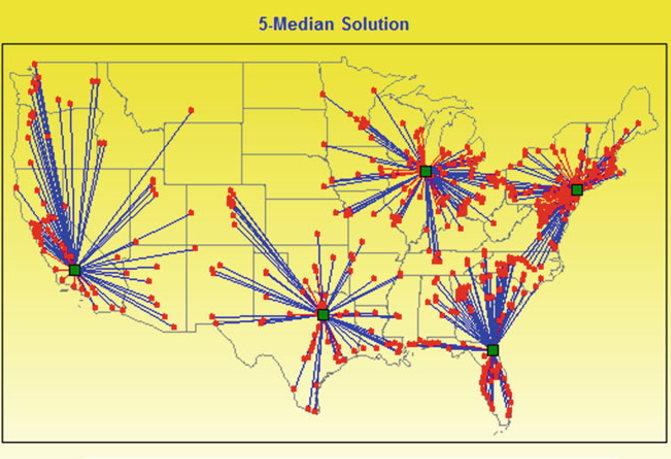
Ràng buộc 1.2 qui định mỗi nút đều phải có một cơ sở cung ứng cho nó.

Ràng buộc 1.3 là có đúng p nút cung ứng.

Ràng buộc 1.4 là quy định các nút yêu cầu chỉ được gán cho các cơ sở cung ứng đã mở.

Ràng buộc 1.5 cho thấy bài toán p- median là bài toán quy hoạch tuyến tính biến Bun. Ràng buộc 1.6 quy định các biến phải có giá trị không âm.

Biến gán có thể không chỉ nhận các giá trị {0,1}. Nếu chi phí đơn vị từ một nút yêu cầu đến cơ sở cung ứng mở gần nó nhất hoàn toàn nhỏ hơn chi phí đơn vị giữa nút đó với bất kì một cơ sở cung ứng mở khác thì biến gán tương ứng cho nút yêu cầu đó sẽ nhận giá trị nhị phân. Nghĩa là khi đó, tất cả các yêu cầu ở nút đó sẽ được gán cho cơ sở cung ứng mở gần nhất. Nếu chi phí đơn vị giữa một nút yêu cầu với hai hay nhiều cơ sở cung ứng mở bằng nhau, và các chi phí đơn vị đó nhỏ hơn chi phí đơn vị giữa nút yêu cầu với bất kì cơ sở cung ứng mở nào khác, biến gán cho thể chỉ ra rằng yêu cầu được chia ra cho các cơ sở cung ứng gần nhất.



## 1.2 Ứng dụng thực tiễn:

Bài toán p – median có rất nhiều ứng dụng trong thực tiễn hiện nay ví dụ như:

* ***Ứng dụng trong quản lý đô thị:***

Khi hình thành một khu đô thị các nhà quản lý phải phân bổ tài nguyên (các dịch vụ công cộng) hợp lý nhằm giảm thiểu chi phí hoạt động cũng như giảm thiểu các thiệt hại nếu xảy ra sự cố. Ví dụ như các bệnh viện, trung tâm cứu hỏa… cần phải bố trí hợp lý sao cho khi xảy ra sự cố có thể có mặt kịp thời giảm thiểu thiệt hại.

* ***Ứng dụng trong việc phân bố kho hàng trong khu công nghiệp:***

Trong một khu công nghiệp việc phân bố kho hàng sao cho chi phí vận chuyển đến các nhà máy là nhỏ nhất cũng là một trong số những ứng dụng của bài toán p-median.

# Chương 2: Một số giải thuật giải bài toán p-median

## 2.1 Độ phức tạp của bài toán p-median:

### 2.1.1 Độ phức tạp tính toán của bài toán

Gọi T­A(X) là thời gian tính của giải thuật A với đầu vào X. Khi đó thời gian tính trong tình huống tồi nhất của giải thuật A với dữ liệu đầu vào X được định nghĩa như sau:



Trong đó n là kích thước bộ dữ liệu.

Việc đánh giá đúng độ phức tạp của bài toán là một vấn đề hết sức phức tạp. Vì vậy chúng ta đưa ra các cận trên và cận dưới cho nó. Và chúng ta quan tâm đến cận trên trong các tình huống tồi nhất vì đây là tiêu chí đánh giá thuật toán. Thuật toán nhanh hơn sẽ cho cận trên tốt hơn. Và cận trên được định nghĩa như sau:

*Tp(n) ≤ O(f(n)).*

Ngoài ra chúng ta còn quan tâm đến cận dưới của bài toán để đánh giá độ khó của nó.

*Tp(n) ≥ Ω(f(n)).*

Để đưa ra cận dưới chúng ta cần phải chỉ ra rằng:

+ Đưa ra được giải thuật giải bài toán p với thời gian tính *Ω(f(n))*.

+ Mọi giải thuật giải bài toán p đều đòi hỏi thời gian tính *Ω(f(n)).*

### 2.1.2 Bài toán NP- Khó:

***Bài toán quyết định:***

Bài toán quyết định là bài toán mà đầu ra chỉ có “yes” hoặc “no” ( Đúng/Sai, Chấp nhận/ Từ chối)

Ví dụ:

* Bài toán cái túi: cho túi kích thước S và các mặt hàng kích cỡ s và khối lượng m. Hỏi xếp các mặt hàng vào túi sao cho đạt khối lượng là lớn nhất.
* Bài toán người du lịch dạng quyết định: Hỏi có thể tìm được hành trình của người du lịch với tổng chi phí không vượt quá một giá trị C cho trước.

Điểm chung của bài toán quyết định là bằng chứng ngắn gọn để kiểm tra. Việc đơn giản chỉ là xác nhận câu trả lời của bộ dữ liệu đầu vào.

Ví dụ:

* Bài toán cái túi chúng ta dễ dàng kiểm tra được câu trả lời cho bộ dữ liệu có thỏa mãn hay không trong thời gian đa thức.
* Bài toán người du lịch thì việc kiểm tra các thành phố đã cho có phải là một hành trình với chi phí C cho trước hay không cũng rất đơn giản.

Lớp các bài toán P, NP, NP-Khó

Đầu tiên ta nêu khái niệm với các lớp bài toán:

Định nghĩa :

+ Ta gọi P là lớp các bài toán có thể giải được trong thời gian đa thức.

+ Ta gọi NP là lớp các bài toán quyết định mà có thể kiểm chứng lời giải trong thời gian đa thức.

+ Ta gọi bài toán NP – Khó là lớp bài toán mà được quy dẫn về từ một bài toán NP – Đầy đủ. Có thể nói H là một bài toán NP đầy đủ , L là một bài toán NP khó khi và chỉ khi L có thể quy về H trong thời gian đa thức.

+ Ta gọi bài toán NP- Đầy đủ là lớp bài toán NP- Khó nằm trong NP. Hay con gọi là những bài toán khó nhất trong lớp NP.

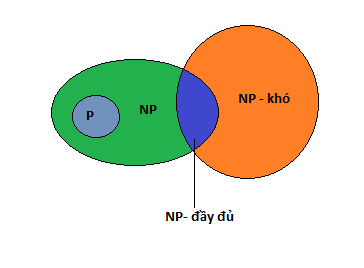
Có rất nhiều bài toán NP – Khó cũng như NP- Đầy đủ như bài toán cái túi, Bài toán TSP,bài toán tô màu đồ thị…..

Nhưng cũng có những bài toán là NP khó nhưng không phải là NP đầy đủ ví dụ như bài toán dừng.

***Tóm lại:***

* Các bài toán thuộc lớp P là các bài toán có thể giải được trong thời gian đa thức.
* Các bài toán thuộc lớp NP là lớp các bài toán quyết định mà có thể kiểm chứng lời giải trong thời gian đa thức.
* Các bài toán thuộc lớp NP- Khó là các bài toán có độ khó ít nhất bằng các bài toán khó nhất trong lớp NP.
* Các bài toán thuộc lớp NP- Đầy đủ là bài toán NP – Khó và nằm trong NP.

Như vậy ta có thể biểu diễn các lớp bài toán như sau:



### 2.1.3 Bài toán P-Median:

***Bài toán P-Median trên đồ thị:***

Có 3 điểm quan trọng của bài toán P – Media đó là:

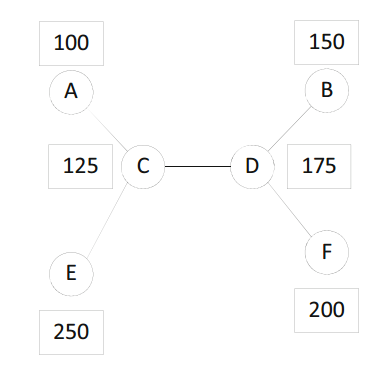
* Thứ nhất: Bài toán P- Median là bài toán NP-Khó trên đồ thị. Kariv và Hakimi đã chỉ ra điều đó vào năm 1979. Nhưng tin tốt là có rất nhiều cách tiếp cận cũng như các thuật toán hiệu quả để giải quyết bài toán p-Median
* Thứ hai: Hakimi đã chỉ ra có ít nhất một giải pháp tối ưu cho bài toán p- Median bao gồm xác định vị trí nhà máy trên nút. Để chỉ ra điều này là đúng chúng ta xem xét một giải pháp xác định nhà máy tại một điểm nào đó trên cạnh giữa hai điểm yêu cầu A và B. Gọi DA là yêu cầu phục vụ của điểm A. DB là yêu cầu phục vụ của điểm B. Nếu DA > DB ta có thể di chuyển nhà máy về điểm A. Và ngược lại, điều đó có thể giúp chúng ta giảm giá trị hàm mục tiêu.
* Thứ ba: Chi phí hoặc khoảng cách sẽ giảm khi bố trí thêm các cơ sở. Điều này là hiển nhiên, Nếu tồn tại một giải pháp tối ưu cho P cơ sở thì việc thêm một cơ sở vào một trong các note còn lại không phải là cơ sở thuộc P sẽ giảm được chi phí. Và nếu bố trí p+1 cơ sở sẽ luôn luôn có chi phí thấp hơn so với p cơ sở.

***Bài toán P-Median trên cây:***

Trong khi bài toán P-Median trên đồ thị là bài toán NP-Khó thì bài toán này lại có thể giải được trong thời gian đa thức.Chúng ta minh họa điều này với một thuật toán có thời gian tuyến tính cho việc tìm 1-Median trên cây được đưa ra bởi Goldman năm 1971. Thuật toán này cũng giải thích tại sao vấn đề lại được gọi là bài toán “median” (trung tâm). Một nút trên cây mà có tổng yêu cầu lớn hơn hoặc bằng tổng yêu cầu của toàn nút trên cây thì hiển nhiên điểm đó là trung tâm. Dù có di chuyển nhà máy đến các điểm khác thì cũng sẽ làm tăng giá trị hàm mục tiêu. Điều này được minh họa như sau:

Giả sử không có một nút nào có yêu cầu lớn hơn hoặc bằng tổng yêu cầu.

Gọi một nút chỉ có kết nối tới một nút khác là nút đầu.



Thuật toán của chúng ta sẽ được mô tả như sau:

* 1. Chọn một nút đầu bất kì loại bỏ nút đó ra khỏi cây và ta biến đổi lại cây. Cập nhật lại yêu cầu của nút nối với nút vừa loại bỏ.
* 2. Kiểm tra yêu cầu của các nút trên cây nếu xuất hiện nút nào có yêu cầu lớn hơn tổng yêu cầu thì dừng lại và nút đó là trung tâm. Nếu ngược lại ta tiếp tục thực hiện bước 1.

Thuật toán trên chúng ta có thể chạy trong thời gian đa thức để tìm ra được trung tâm.

## 2.2 Một số cách tiếp cận và giải bài toán P-Median

Một số thuật toán cơ bản và cải tiến:

+ Thuật toán Myopic: Thuật toán Myopic xuất phát từ bài toán 1-Median. Thuật toán tìm ra các median bằng các tìm 1-media của các phần của đồ thị. Thuật toán là tối ưu với bài toán 1-median nhưng không tối ưu với các bài toán khác.

+ Thuật toán tìm kiếm láng giềng: Được đề xuất bởi Maranzana năm 1964 bằng việc tìm kiếm lân cận các ứng cử viên với tập các median ban đầu.

+ Thuật toán Trao đổi: Được đề xuất bởi Teitz và Bart 1968 là một cách cải tiến của thuật toán tìm kiếm láng giềng Thuật toán có nhiều cách thực hiện trên các nền tảng nhằm cải tiến thời gian.

+ Thuật toán Genetic: Được phát triển bởi Mayerle 1996 dựa trên sự phát triển chọn lọc tự nhiên nhằm loại bỏ các khả năng xấu để giảm bớt các trường hợp xảy ra.

+ Thuật toán tối ưu hóa bầy đàn: Được phát triển bởi Eberhat và Kennedy đã đưa ra một mô hình tối ưu bầy đàn dựa trên việc tìm thức ăn của đàn cá và chim.

+ Một số thuật toán khác: Thuật toán lọc (Filtering), Tìm kiếm kiếm dựa trên Lagrangian và đa số các thuật toán khác đều phát triển dựa trên các thuật toán cơ bản nêu trên.

## 2.3 Giải thuật tham lam (Myopic Algorithm):

Cách tiếp cận đơn giản nhất là sử dụng giải thuật tham lam để giải các bài toán NP-Khó. Đây là một trong số những cách tiếp cận đơn giản nhất để giải bài.

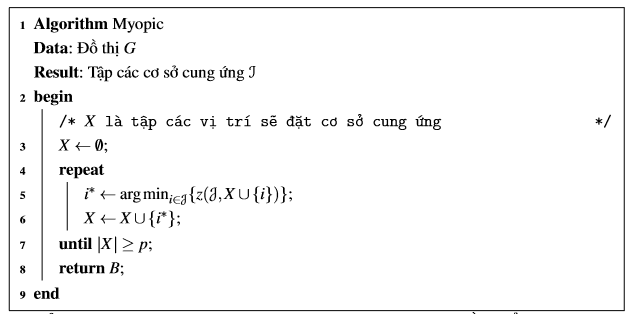
Ý tưởng: Tất cả các nút ứng cử viên được kiểm tra và ứng cử viên nào làm giảm tổng chi phí lớn nhất thì chúng ta thêm vào lời giải. Thuật toán tiếp tục kiểm tra cho đến khi thêm đủ p cơ sở cung ứng trong lời giải.

Chúng ta định nghĩa tổng chi phí cung ứng từ một nút ứng cử viên đến các nút còn lại như sau:



Trong đó X là ứng cử viên đang được xét.

Chú ý rằng việc xét các ứng cử viên phải phụ thuộc cả vào tập các điểm yêu cầu trước và sau khi thêm một ứng cử viên nào đó vào lời giải.  
 Giải thuật Myopic:



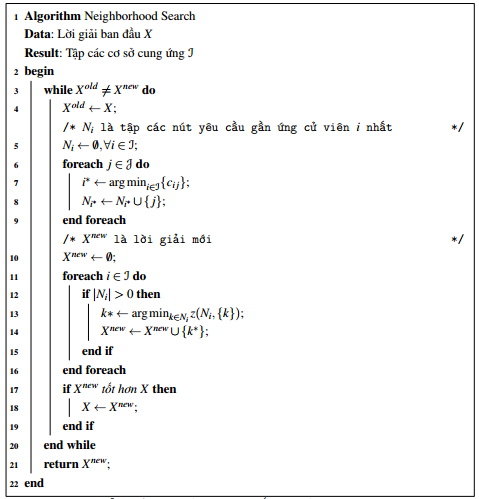
Đầu tiên ta khởi tạo tập lời giải. Bước tiếp theo ta tìm ứng cử viên làm giảm tổng chi phí lớn nhất. Sau đó thêm vào lời giải. Lặp lại bước trên cho đến khi tìm đủ p cơ sở thì giải thuật kết thúc.

Giải thuật tham lam dễ dàng có thực hiện trong một thời gian ngắn kể cả với bộ dữ liệu lớn. Nhưng không có đảm bảo việc giải thuật có thể đưa ra lời giải tối ưu và kết quả khá kém. Chúng ta có thể biết được rằng tối ưu cục bộ có thể chưa chắc là tối ưu toàn cục vì thế kết quả không được chính xác. Nếu p = 1 thì thuật toán tham lam sẽ đưa ra một lời giải tối ưu.

## 2.4 Thuật toán tìm kiếm láng giềng (Neighborhood Search Algorithm):

***Ý tưởng:***

Dựa trên kết quả tối ưu của bài toán 1-Median. Bắt đầu với một lời giải chấp nhận được có thể là lời giải từ giải thuật tham lam, hoặc khởi tạo ngẫu nhiên. Chúng ta tìm kiếm các lời giải lân cận với lời giải ban đầu, trong tập hợp các nút lân cận với lời giải chúng ta sinh ra các lời giải lân cận. Trong các lời giải lân cận vừa tìm được chúng ta lấy ra một lời giải tốt nhất và tiếp tục tìm kiếm như bước trên. Giải thuật dừng khi không có thay đổi trong lời giải hoặc đạt giới hạn điều kiện lặp.



Đầu tiên khởi tạo một lời giải bất kì.

Từ bước 5-> 9 Khởi tạo các tập láng giềng dựa trên lời giải ban đầu.

10 Khởi tạo lời giải mới.

11-> 16 tìm kiếm 1-Median của các tập láng giềng và thêm đỉnh này vào lời giải mới.

17-> 19 kiểm tra lời giải vừa tìm được nếu tốt hơn lời giải cũ thì cập nhật lời giải cũ. Nếu không thì dừng.

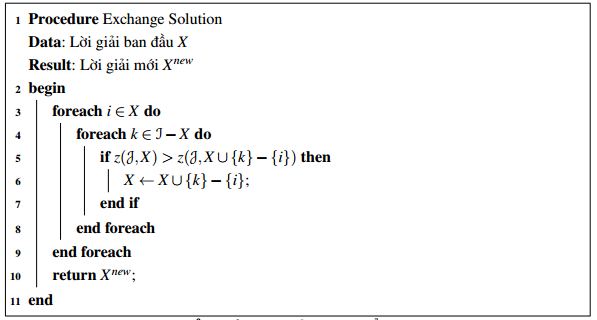
Giải thuật này giả sử tập các cơ sở cung ứng và tập điểm yêu cầu là cùng một tập. Tuy rằng giải thuật luôn tìm ra được lời giải tối ưu (1-median) cho mỗi láng giềng nhưng sẽ không tối ưu cho lời giải tối ưu toàn cục.

## 2.5 Giải thuật trao đổi:

Giải thuật trao đổi được đề xuất bởi Teitz và Bart năm 1968. Là một cách tiếp cận khác để giải bài toán p-Median và có hiệu năng tốt hơn giải thuật tìm kiếm láng giềng.

***Ý tưởng:***

Giải thuật cải thiện lời giải bằng cách thay đổi một nút trong lời giải bằng một nút khác không có trong lời giải. Nếu việc trao đổi có thể cải thiện lời giải chúng ta chấp nhận nút đó và ngược lại. Giải thuật dừng lại khi số vòng lặp đạt tối đa hoặc điều kiện kết thúc.



Bước đầu tiên chúng ta sẽ khởi tạo một lời giải. Bước tiếp theo với mỗi nút thuộc lời giải chúng ta tính hàm mục tiêu của lời giải chứa nút đó và một nút khác trong đồ thị. Nếu giá trị hàm mục tiêu tốt hơn chúng ta trao đổi hai đỉnh đó và ngược lại. Kết thúc khi điều kiện lặp tối đa hoặc đạt giá trị tối ưu.

Có rất nhiều cách để thực hiện giải thuật trao đổi, thay vì khởi tạo ngẫu nhiên một lời giải ta có thể khởi tạo lời giải bằng cách chạy giải thuật tham lam và đầu ra của giải thuật tham lam chúng ta lấy làm giá trị khởi tạo của giải thuật trao đổi.

## 2.6 Giải thuật di truyền:

### 2.6.1 Giới thiệu giải thuật di truyền:

Giải thuật di truyền được phát minh để bắt chước quá trình tiến hóa trong tự nhiên. Nhiều người, nhà sinh học, ngạc nhiên rằng cuộc sống phức tạp mà chúng ta đang quan sát thấy được tiến hóa trong thời gian tương đối ngắn. Ý tưởng của giải thuật di truyền là sử dụng sức mạnh của tiến hóa để giải quyết các vấn đề tối ưu hóa.

Ý tưởng tính toán tiến hóa được giới thiệu lần đầu bởi I.Renchenberg vào những năm 60. Ý tưởng của ông đã được nhiều nhà nghiên cứu khác phát triển.

Người đầu tiên đề xuất là John Holland, ông đề xuất giải thuật di truyền vào năm 1975 cùng các học trò.

Năm 1992, John Koza sử dụng giải thuật di truyền để thực hiện một số tác vụ.

Giải thuật di truyền là giải thuật tìm kiếm dựa trên sự thích nghi của các cá thể trong quần thể nhờ vào việc tiến hóa, chọn lọc tự nhiên và di truyền học.

Giải thuật đại diện cho việc tìm kiếm ngẫu nhiên để giải vấn đề tối ưu hóa. Mặc dù được nói là ngẫu nhiên nhưng giải thuật di truyền không ngẫu nhiên mà thay vào đó là việc khai thác thông tin lịch sử để tìm kiếm vào khu vực có khả năng tốt hơn trong toàn bộ không gian tìm kiếm.

Việc cơ bản nhất trong giải thuật di truyền là việc mô phỏng quá trình phát triển của loài trong tự nhiên nhờ tiến hóa. Vì trong tự nhiên, sự cạnh tranh giữa các cá thể để có thể tồn tại trong một môi trường giới hạn thì các cá thể phù hợp hay mạnh hơn sẽ chiếm ưu thế và loại bỏ các cá thể yếu.

***Tính đúng đắn của giải thuật:***

Giải thuật di truyền dựa trên ý tưởng của sự phát triển trong tự nhiên mà cho đến nay sự chọn lọc trong tự nhiên đã phản ánh chính xác cách thức phát triển của các loài vì vậy tính đúng đắn là tất yếu.

Xuất phát từ quần thể gồm các nhiễm sắc thể ngẫu nhiên được sinh ra. Giải thuật di truyền tiến hành ước lượng đánh giá chúng nhằm xác định xem chúng có khả năng sinh sản ra các cá thể mới tốt hơn hay không. Độ tốt của lời giải phụ thuộc vào quần thể ban đầu. Nếu chúng hoàn toàn ngẫu nhiên và phân bố trên toàn bộ không gian tìm kiếm chúng ta có thể tìm được lời giải tối ưu nhất và ngược lại.

Với tầm ứng dụng rộng rãi chúng ta có thể sử dụng giải thuật di truyền trong rất nhiều trường hợp khác nhau của các bài toán tối ưu. Điển hình như các không gian nhiều chiều, các không gian khó có thể mô tả nhưng có thể biểu diễn lời giải một cách dễ dàng… Do đó giải thuật di truyền là một phương pháp thô nhưng rất tổng quát.

### 2.6.2 Sơ đồ thực hiện và các toán tử trong giải thuật di truyền.

#### Nhiễm sắc thể:

Mọi cơ thể sống đều được phát triển từ các tế bào. Trong các tế bào chứa các cặp nhiễm sắc thể và nhiễm sắc thể bao gồm các gen. Các gen trong nhiễm sắc thể là thành phần chính trong việc di truyền. Các gen biểu diễn các đặc tính của loài cũng như của các cá thể.

Trong quá trình tạo ra một cá thể mới, quá trình lai ghép sẽ diễn ra đầu tiên. Các gen của cha, mẹ sẽ kết hợp với nhau tạo ra một các thể hoàn toàn mới. Con mới được tạo ra có thể bị đột biến nhưng tỉ lệ là rất thấp và thường kém hơn so với cha mẹ. Nhưng chính đột biến lại là bước ngoặt tiến hóa cho giống nòi. Độ tương thích của cá thể có thể coi như là thời gian sống của cá thể đó trong loài.

Tương tự như trong tự nhiên, giải thuật di truyền cũng biểu diễn lời giải theo nhiễm sắc thể. Các gen khác nhau sẽ đưa ra các đặc tính khác nhau của lời giải. Việc tìm kiếm trong giải thuật di truyền được mô tả tương tự như việc lai ghép, đột biến trong tự nhiên. Việc đánh giá cá thể mới so với cá thể cũ chính là hàm mục tiêu cũng như là lời giải cho bài toán.

#### Không gian tìm kiếm:

Khi giải một bài toán, chúng ta sẽ tìm lời giải tốt nhất cho nó. Tập tất cả các lời giải chấp nhận được sẽ được gọi là không gian tìm kiếm. Mỗi điểm trong không gian được gọi là một lời giải chấp nhận được và đều được đánh giá bởi một hàm mục tiêu (Tính tối ưu của lời giải). Trong giải thuật di truyền chúng ta tìm kiếm lời giải bằng cách tìm kiếm từ các lời giải chấp nhận được.

Việc tìm kiếm lời giải tương đương với việc tìm kiếm giá trị tối ưu cho bài toán. Nếu thời gian cho phép chúng ta có thể xác định được toàn bộ không gian tìm kiếm và sẽ đưa ra được lời giải tối ưu nhất cho bài toán. Tuy nhiên, trong thực tế chúng ta không thể lưu trữ cũng như tính toán được toàn bộ không gian tìm kiếm vì thế chúng ta chỉ có thể biết một số điểm trong đó và tiến hành tiến hóa để tìm được lời giải mong muốn.

Việc tìm kiếm là rất phức tạp, vì chúng ta không thể biết được việc tìm kiếm lời giải tối ưu nhất bắt đầu từ đâu và ở khu vực nào trong không gian tìm kiếm. Có rất nhiều phương pháp tìm kiếm lời giải thích hợp như HillClimbing, Tabu search, Filtering, Genetic… nhưng chưa hẳn là tối ưu (thường là tối ưu cục bộ). Những lời giải tìm được bằng các phương pháp này được coi là tốt và chấp nhận được trong một số điều kiện khác nhau.

Rất nhiều bài toán NP-Khó được giải theo giải thuật di truyền(Genetic Algorithm) vì khả năng tìm kiếm lời giải tương đối nhanh cũng như các giá trị chênh lệch có thể chấp nhận được. Ví dụ như: Bài toán cái túi, Bài toán người du lịch, ……

#### Sơ đồ giải thuật di truyền:

Giải thuật di truyền dựa trên thuyết tiến hóa của Darwin. Để tìm được lời giải của bài toán, giải thuật đã thực hiện một quá trình tiến hóa.

Giải thuật bắt đầu từ một tập các lời giải gọi là quần thể trong đó mỗi lời giải là một cá thể. Lời giải mới sẽ được sinh ra từ các lời giải ban đầu với mong muốn xây dựng một quần thể mới tốt hơn quần thể ban đầu.

***Sơ đồ giải thuật di truyền:***

1. Bắt đầu: Khởi tạo một quần thể với n cá thể (thỏa mãn là lời giải của bài toán).
2. Đánh giá: Đánh giá khả năng tương thích cho mỗi cá thể (giá trị hàm mục tiêu).
3. Tạo ra quần thể mới: chúng ta lặp đi lặp lại các bước dưới đây để tạo ra một quần thể mới.

+ Chọn hai nhiễm sắc thể bố mẹ từ các nhiễm sắc thể ban đầu đã được sắp xếp theo thứ tự tương thích.

+ Lai ghép hai nhiễm sắc thể được để tạo ra con lai mới, nếu lai ghép không thành công thì con mới sẽ y nguyên bố mẹ.

+ Đột biến xảy ra với tỉ lệ thấp.

+ Chấp nhận con lai và đưa vào quần thể mới.

1. Thay thế: Các cá thể mới tạo ra nếu có độ tương thích cao hơn sẽ thay thế các cá thể cũ. Đây chính là chọn lọc tự nhiên.
2. Điều kiện dừng: Giải thuật dừng nếu điều kiện cuối cùng được thỏa mãn. Giải thuật kết thúc và đưa ra giá trị tốt nhất trong quần thể hiện tại.

Sơ đồ giải thuật di truyền bên trên là sơ đồ chung. Khi áp dụng cho các bài toán khác nhau chúng ta có thể biến đổi để phù hợp cho từng bài toán.

Có rất nhiều câu hỏi sẽ được đặt ra ở đây ví dụ như:

+ Cách mã hóa lời giải như thế nào?

+ Chọn bố mẹ thế nào để lai ghép

Điều này với mỗi bài toán khác nhau chúng ta có thể mã hóa các lời giải theo nhiều cách khác nhau và việc chọn bố mẹ lai ghép có thể được thực hiện bằng nhiều cách nhưng chủ yếu sẽ chọn những bố mẹ tốt để lai ghép với nhau nhằm hi vọng sinh ra con tốt hơn.

+ Cũng có thể sau khi lai ghép chúng ta loại bỏ hoàn toàn các cá thể cũ. Điều đó sẽ là mất đi các gen tốt nhất trong quần thể ban đầu. Như vậy việc để lại một số cá thể tốt nhất từ quần thể ban đầu là hợp lý nhằm đảm bảo các lời giải tốt nhất có thể tồn tại đến khi kết thúc giải thuật.

#### Các toán tử trong giải thuật di truyền:

***Toán tử lai ghép:***

Sau khi đã quyết định sử dụng loại mã hóa nào chúng ta sẽ chuyển sang giai đoạn lai ghép. Đầu tiên chúng ta lựa chọn những cá thể cha mẹ và tạo ra con lai. Một số cách lai ghép cơ bản:

Lai ghép 1 điểm cắt: Chúng ta chọn ngẫu nhiên một điểm lai ghép và sao chép phần trước hoặc sau của điểm lai ghép từ bố sang mẹ và ngược lại. Lai ghép một điểm cắt có thể mô tả như sau:

Cha : 10100111|0011010

Mẹ: 11001101|0010011

Con 1: 11001101|0011010

Con 2: 10100111|0010011

Lai ghép nhiều điểm cắt: Chúng ta chọn 2 hoặc nhiều hơn điểm lai ghép trên bố và mẹ và tiến hành sao chép như một điểm cắt.

Cha : 101|00111|0011010

Mẹ: 110|01101|0010011

Con 1: 110|00111|0010011

Con 2: 101|01101|0011010

Toán tử lai ghép là khá phức tạp và phụ thuộc vào nhiều yếu tố trong các bài toán khác nhau vì thế chọn được toán tử lai ghép thích hợp sẽ làm tăng tính hiệu quả trong việc tìm lời giải.

***Toán tử đột biến:***

Sau khi lai ghép được con lai chúng ta có thể tiến hành đột biến. Đột biến với tỉ lệ thấp nhưng có thể ngăn việc tìm ra các cực trị địa phương. Toán tử đột biến sẽ thay đổi ngẫu nhiên con mới thu được sau khi lai ghép. Chúng ta có thể mô tả như sau:

Con 1 ban đầu: 110001110010011

Con 2 ban đầu: 101011010011010

Con 1 đột biến: 110**1**01110010011

Con 2 đột biến: 1**1**1011010011010

Việc đột biến cũng tương tự như lai ghép phụ thuộc chủ yếu vào việc mã hóa các cá thể.

#### Các tham số trong giải thuật di truyền:

***Xác suất lai ghép, đột biến:***

Xác suất lai ghép: cho biết tần suất lai ghép. Nếu không có toán tử lai ghép con sẽ giống hệt cha mẹ ban đầu. Ngược lại con lai sẽ được tạo ra từ một phần của cha và mẹ. Nếu xác suất lai ghép là 100% thì toàn bộ cha mẹ đều được lai ghép để tạo ra con lai. Nếu xác suất lai ghép là 0% thì toàn bộ con mới sẽ là bản sao của thế hệ cũ nhưng không hoàn toàn giống nhau vì chúng ta còn có xác suất đột biến.

Việc thực hiện lai ghép mong muốn con mới sinh ra sẽ giữ các phần tốt của cha mẹ và do đó các cá thể mới sẽ tốt hơn. Nhằm mong muốn giữ lại các cá thể tốt vì vậy chúng ta nên chuyển một phần của quần thể cũ sang quần thể mới.

Xác xuất đột biến: Cho biết tần suất đột biến trong quần thể. Nếu không có đột biến con lai được tạo ra sẽ giữ nguyên. Nếu xuất hiện đột biến con lai sẽ được biến đổi một phần. Nếu xác suất đột biến là 100% thì toàn bộ con lai sẽ bị đột biến. Nếu xác suất đột biến là 0% thì toàn bộ con lai sẽ giữ nguyên.

Việc thực hiện đột biến nhằm tránh khỏi các giá trị cục bộ địa phương vì thế đột biến là nhân tố quan trọng trong quá trình tìm ra lời giải tối ưu. Nếu đột biến thường xuyên thì giải thuật sẽ thành tìm kiếm ngẫu nhiên mà không còn là giải thuật di truyền.

***Một số tham số khác:***

Ngoài hai tham số quan trọng trên thì chúng ta cũng có một tham số cũng rất quan trọng đó là kích thước quần thể. Nếu trong quần thể có ít nhiễm sắc thể thì đồng nghĩa với việc lai ghép cũng ít xảy ra và khả năng tìm kiếm cũng bị giới hạn trong một không gian hẹp. Mặt khác nếu có quá nhiều nhiễm sắc thể đồng nghĩa với việc công việc tính toán tăng lên vì vậy giải thuật sẽ trở nên chậm chạp. Các nghiên cứu cho thấy nếu chọn được kích thước quần thể thích hợp thì giải thuật sẽ chạy tốt và việc tăng thêm kích thước quần thể cũng không ảnh hưởng đến hàm mục tiêu.

Vì vậy việc chọn kích thước quần thể là rất quan trọng trong giải thuật để đảm bảo giải thuật chạy nhanh cũng như có thể tìm kiếm trong toàn bộ không gian.

### 2.6.3 Giải thuật di truyền cho bài toán p-median:

Với số đỉnh nhỏ hơn 100, các giải thuật bên trên chạy tương đối tốt đặc biệt là giải thuật trao đổi cho kết quả rất tốt. Nhưng khi số đỉnh lớn hơn số median tăng lên cùng với đó số cạnh của đồ thị cũng tăng lên nên các giải thuật trên chạy rất chậm. Vì vậy phần này trình bày một giải thuật hiệu quả cho bài toán p-median, đó là giải thuật di truyền.

#### Mã hóa:

Xét bài toán với n đỉnh được đánh số từ 1, 2, 3, 4, … , n cần chọn p-median. Ta sẽ mã hóa lời giải bằng cách lấy số thứ tự các đỉnh và sắp xếp theo thứ tự tăng dần. Ta sử dụng mảng p phần tử trong đó mỗi phần từ là số thứ tự của đỉnh được chọn và đã được sắp xếp.

Một cách khác ta có thể mô tả lời giải bằng một xâu nhị phân độ dài n trong đó với các đỉnh được chọn sẽ nhận giá trị 1 và ngược lại. Như vậy tổng số bit 1 trên xâu được biểu diễn phải chính bằng p.

Ví dụ: n = 10, p = 3 một nhiễm sắc thể có thể được biểu diễn như sau:

Y = [3,5,6] hoặc Y = 0010110000 chính là lời giải của bài toán.

#### Hàm đánh giá:

Hàm đánh giá độ tốt của nhiễm sắc thể ( cá thể) cũng chính bằng giá trị hàm mục tiêu cần tối ưu. Chính bằng

Trong đó :

+ n là số đỉnh trong đồ thị.

+ p là số median.

+ cij là chi phí đi từ đỉnh i đến đỉnh j.

+ xij nhận giá trị nhị phân có giá trị bằng 1 nếu đỉnh i được phục vụ bởi đỉnh j và ngược lại.

#### Khởi tạo quần thể ban đầu:

Sinh ngẫu nhiên các nhiễm sắc thể là các xâu nhị phân hoặc mảng có đúng p giá trị 1 hoặc p phần tử.

Chúng ta tiến hành sinh như sau:

1. Sinh ngẫu nhiên 1 số trong khoảng từ 1 đến n.
2. Đưa giá trị đó vào lời giải.
3. Lặp lại bước 1 cho đến khi tìm được đủ p giá trị.

Việc sinh ngẫu nhiên các nhiễm sắc thể sẽ đảm bảo các nhiễm sắc thể sẽ phân bố rộng khắp trên không gian tìm kiếm.

Ngoài ra chúng ta nên đặt một số gen tốt vào trong quần thể điều đó có thể rút ngắn quá trình tìm kiếm. Trong phần này chúng ta có thể cho thêm 1 gen chính là lời giải của giải thuật Myopic vào trong quần thể.

#### Lựa chọn bố mẹ lai ghép:

***Lựa chọn bố mẹ:***

Việc lựa chọn bố mẹ lai ghép sao cho hợp lý cũng giảm thời gian tính toán tương đối lớn. Trong phần này việc chọn bố mẹ lai ghép là ngẫu nhiên nhưng các các thể tốt hơn sẽ có tỉ lệ được chọn cao hơn. Để chọn nhiễm sắc thể lai ghép ta chọn theo phương pháp sau:

1. Tính tổng chi phí của mỗi cá thể là C. Tính độ tốt của cá thể bằng cách lấy thương số giữa tổng chi phí và chi phí của các cá thể. Gọi thương số đó là d. Tính tổng các thương số gọi giá trị đó là D.
2. Sinh ngẫu nhiên 1 số d trong khoảng từ 1 -> D.

Sinh ngẫu nhiên một số n trong khoảng từ 1 đến K. Trong đó K là số lượng cá thể trong quần thể.

1. Bắt đầu từ vị trí n tính hiệu các chi phí khi nào chi phí vượt quá d thì lấy cá thể tại vị trí đó vào lai ghép.
2. Lặp lại để tìm cá thể lai ghép tiếp theo.

Rõ ràng việc lựa chọn như vậy sẽ đảm bảo rằng việc lấy các cá thể lai ghép là ngẫu nhiên. Nhưng cũng đảm bảo rằng các cá thể tốt hơn sẽ có tỉ lệ được chọn cao hơn.

***Lai ghép:***

Để tạo ra các thể mới chúng ta tiến hành lai ghép. Trong tự nhiên con lai sẽ thừa hưởng đặc tính của bố mẹ và trong giải thuật di truyền cũng vậy. Ta sẽ lai ghép giữa bố và mẹ, chọn lựa các gen tốt để tạo ra con và mong muốn tạo ra con tốt hơn bố mẹ, điều đó cũng gần với lời giải tối ưu hơn.

Toán tử lai ghép:

1. Với mỗi cặp bố mẹ chúng ta sắp xếp thành 2 phần: Phần gen chung và phần gen riêng. Phần gen chung là phần chung của cả bố và mẹ. Phần riêng là phần gen khác nhau giữa bố và mẹ. Trong bài toán này phần chung sẽ là các đỉnh mà đều chứa ở cả bố và mẹ. Ngược lại là phần riêng.
2. Từ phần gen riêng chúng ta lấy ra ½ gen của bố và mẹ rồi hoán đổi cho nhau. Điều này đảm bảo rằng con sinh ra cũng là lời giải chấp nhận được.

Nhận xét: Trong khi lai ghép nếu bố mẹ giống nhau thì sẽ sinh ra con lai hoàn toàn giống bố mẹ.

#### Đột biến:

Để đảm bảo việc đa dạng trong quần thể chúng ta cần phải tiến hành đột biến nhằm tránh trường hợp con giống hệt bố mẹ ta thực hiện toán tử đột biến như sau.

Thay thế một đỉnh bất kì trong con lai bằng một đỉnh khác ngoài đồ thị. Như vậy phép đột biến này sẽ có thể tốt hoặc xấu sau khi xảy ra. Thường thì sẽ xấu hơn con lai ban đầu. Để tránh điều đó ta tìm các tập đỉnh theo thứ tự giảm dần theo giải thuật Myopic và thay thế đỉnh tốt nhất vào một gen trong con lai. Điều này sẽ tăng khả năng tạo ra con lai mới có độ tốt lớn hơn.

Sơ đồ đột biến có thể miêu tả như sau:

1. Gọi V là tập các đỉnh trên đồ thị:

Tính chi phí phục vụ của mỗi đỉnh cho toàn đồ thị.

Sắp xếp theo thứ tự tăng dần chi phí.

1. Gọi S là tập các đỉnh trong nhiễm sắc thể của con lai.

vi  là một đỉnh trong tập V\S. Thay thế 1 đỉnh bất kì trong S bằng vi và cập nhật chi phí.

#### Thay thế:

Sau khi tiến hành lai ghép ta sẽ giữ lại 1 phần của quần thể cũ và chuyển sang quần thể mới. Phần này là những cá thể tốt nhất và thay thế những cá thể tồi hơn trong quần thể cũ bằng những cá thể tốt hơn trong quần thể mới.

Tức là sau khi lai ghép ra một cá thể mới chúng ta sẽ loại bỏ 1 cá thể tồi nhất trong quần thể và thay vào đó là cá thể mới.

#### Sơ đồ giải thuật:

1. Khởi tạo mảng chi phí đối với mỗi đỉnh.
2. Sắp xếp mảng chi phí.
3. Khởi tạo xác suất lai ghép, đột biến.
4. Khởi tạo quần thể.
5. Tính giá trị chi phí các phần tử trong quần thể.
6. Chuyển một số cá thể tốt nhất sang quần thể mới.
7. Chọn bố mẹ.
8. Lai ghép.
9. Kiểm tra con lai nếu tốt hơn cá thể tồi nhất trong quần thể thì thêm vào quần thể mới.
10. Cập nhật giá trị tốt nhất.
11. Lặp lại bước 7 cho đến khi đạt điều kiện dừng.

Như vậy giải thuật sẽ chạy cho đến khi đạt số vòng lặp tối đa hoặc sau một số lượng vòng lặp nào đó mà không cải thiện giá trị tốt nhất thì giải thuật sẽ dừng.

## 2.7 Giải thuật tối ưu hóa bầy đàn (Particle Swarm Optimization)

### 2.7.1 Giới thiệu giải thuật tối ưu hóa bầy đàn và bài toán:

#### Giải thuật tối ưu hóa bầy đàn:

Giải thuật tối ưu hóa bầy đàn là một trong số các chiến lược tìm kiếm trong tính toán tiến hóa, được đề xuất bời Kennedy và Eberhat dựa trên sự liên kết trong cộng đồng của các loài động vật kiếm ăn theo đàn ví dụ như: Cá, Chim, Ong.... Mặt khác nó có thể coi là một phương pháp tính toán tiến hóa với cách thức dựa trên việc thăm dò, tìm kiếm trong không gian tìm kiếm dựa trên các cá thể. Việc khai thác thông tin của các cá thể chính là cách tìm ra lời giải tối ưu cho bài toán.

Mặt khác nó không giống với các phương pháp khác trong tính toán tiến hóa như sử dụng các toán tử lai ghép, đột biến. Một ưu điểm của giải thuật này là dễ sử dụng với các thông số ít hơn để dễ theo dõi và điều chỉnh.

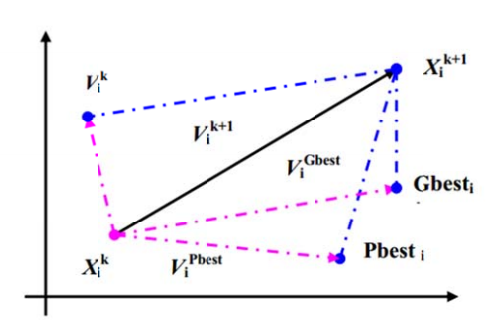
Trong tối ưu hóa bầy đàn, các lời giải tiềm năng được gọi là các cá thể, di chuyển xung quanh trong không gian tìm kiếm với vận tốc được cập nhật liên tục theo kinh nghiệm của bản thân, của các cá thể lân cận hay kinh nghiệm của toàn đàn.

Tối ưu hóa bầy đàn được sử dụng rộng rãi trên nhiều lĩnh vực đặc biệt là các bài toán NP-Khó như lập lịch, mạng noron và các bài toán tối ưu.

Tối ưu hóa bầy đàn được khởi tạo bằng một nhóm cá thể ngẫu nhiên và sau đó tìm nghiệm tối ưu bằng cách cập nhật theo hai vị trí tốt nhất.

Giá trị thứ nhất là vị trí tốt nhất mà nó đã đạt được ở thời điểm hiện tại. Giá trị này được gọi là Pbest

Giá trị thứ hai là vị trí tốt nhất mà đã đạt được từ thời điểm bắt đầu đến hiện tại. giá trị này được gọi là Gbest



Trong đó:

Xik là vị trí của cá thể thứ i ở thời điểm thứ k.

Vik là vận tốc của cá thể thứ i ở thời điểm thứ k.

Pbesti là vị trí tốt nhất của cá thể thứ i nào đó trong quần thể.

ViPbestlà vận tốc của cá thể tốt nhất ở thời điểm hiện tại.

Gbesti là vị trí tốt nhất của toàn bộ quần thể từ khi bắt đầu tìm kiếm đến hiện tại.

ViGbest là vận tốc của cá thể tốt nhất từ khi bắt đầu tìm kiếm đến hiện tại.

Xik+1 Là vị trí của các thể thứ i ở thời điểm k+1

Vik+1 Là vận tốc của cá thể thứ i ở thời điểm k+1.

Vận tốc và vị trí của các cá thể trong quần thể sẽ được cập nhật theo từng thời điểm và được tính theo công thức sau.

Vik + 1= \* Vik + c1 \* r­­­1 \* (Pbest – Xik) + c2 \* r2 \* (Gbest – Xik)

Xik + 1= Xik + Vik + 1

Trong đó:

là hệ số quán tính.

c1 , c2 là hệ số giá tốc nhận giá trị trong khoảng từ 1,5 đến 2.5

r1 , r2 là các số ngẫu nhiên nhận giá trị trong khoảng từ 0 đến 1

Giá trị của trọng số rất quan trọng trong việc tìm kiếm. Nếu giá trị lớn cho phép mở rộng phạm vi tìm kiếm. Ngược lại nếu cho phép làm tăng sự thay đổi để nhanh đạt giá trị tối ưu địa phương. Bởi vậy việc giảm dần theo thời gian sẽ mang đến hiệu quả tốt nhất trong việc tìm kiếm.

Sơ đồ của giải thuật tối ưu hóa bầy đàn:

* + - 1. Khởi tạo quần thể và các giá trị liên quan.
      2. Tính toán các giá trị tối ưu cục bộ và toàn cục.
      3. Cập nhật vận tốc các cá thể theo giá trị tối ưu.
      4. Lặp lại bước 2 cho đến khi đạt điều kiện dừng.
      5. Đưa ra kết quả.

#### Giới thiêu bài toán:

Việc lựa chọn các cơ sở trong bài toán đặt vị trí cơ sở là cực kì phức tạp. Hơn nữa việc lựa chọn ra một vị trí tối ưu nhất là rất tốn kém. Các vấn đề về việc đặt các cơ sở là một trong số các vấn đề được nghiên cứu rộng rãi nhất trong các bài toán tối ưu hóa NP-Khó. Dựa trên một số yêu cầu khác nhau mà bài toán được chia ra làm bốn loại cơ bản:

+ Bài toán p- Median.

+ Bài toán p- Center.

+ Bài toán Uncapacitated facility location.

+ Bài toán Capacitated facility location.

Trong phần này chúng ta sẽ dùng giải thuật tối ưu hóa bầy đàn giải bài toán UFL. Trong bài toán UFL có hai vấn đề cần tối ưu đó là chi phí để xây dựng một nhà máy và chi phí vận chuyển từ nhà máy đó để đáp ứng các yêu cầu từ phía khách hàng. Năng lực cung cấp của các cơ sở được giả định là vô hạn.   
Các vấn đề UFL cũng được gọi là các vấn đề đặt vị trí các cơ sở (Facility Location Problem), vấn đề đặt vị trí nhà kho (Warehouse Location Problem)....

Trong các bài toán UFL có số các cơ sở n và số khách hàng m. Mỗi cơ sở có chi phí cố định fcj là chi phí xây dựng cơ sở thứ j. Chi phí để vận chuyển từ cơ sở i đến khách hàng j là cij và bài toán này giả định các cơ sở có thể cung cấp dịch vụ không giới hạn cho khách hàng. Chúng ta cần phải tối ưu hóa hàm mục tiêu như sau:

 (1)

 i = 1,…,m (2)

|  |
| --- |
| *0 ≤ xij≤ yj or yj ∈ {0; 1}; i= 1,…,m ; j=1,… ,n* (3) |

Trong đó:

xij là số lượng được cung cấp từ cơ sở j đến khách hàng thứ i.

y­j là ứng cử viên trong các nhà máy. Nếu được chọn yj nhận giá trị 1 và ngược lại.

* + 1. Chính là hàm mục tiêu.
    2. Là điều kiện đảm bảo rằng mỗi khách hàng sẽ được phục vụ bởi 1 cơ sở.
    3. Các biến phải là nguyên.

Tương tự như các bài toán p-Median được nêu ra trước đây bài toán này cũng có rất nhiều cách giải chính xác nhưng đều bị hạn chế bởi độ lớn của các bộ dữ liệu. Với các bộ dữ liệu nhỏ cách giải chính xác hoàn toàn có thể tìm được ra các giá trị tối ưu. Nhưng với các bộ dữ liệu lớn, số cạnh và số đỉnh rất lớn để tìm ra kết quả chính xác sẽ tốn chi phí rất lớn về cả thời gian và tài chính.

Kratica đã chỉ ra rằng sử dụng các giải thuật di truyền để tìm ra giá trị tối ưu mang lại hiệu quả rất tốt.

Một số phương pháp tìm kiếm cục bộ khác cũng được đưa ra nhưng đều mang lại hiệu quả không tốt bằng các giải thuật di truyền ví dụ như:

* Al-Sultan và Al- Fawzan đưa ra phương pháp tìm kiếm tabu đưa ra được kết quả rất tốt nhưng chi phí tính toán và thời gian tính rất lớn cũng như bị giới hạn trong việc áp dụng thực tiễn.

- Gen và Vaithyanathan đề xuất phương pháp tiếp cận theo mạng noron để giải quyết bài toán này nhưng kết quả không được tốt lắm.

### 2.7.2 Áp dụng giải thuật tối ưu hóa bầy đàn giải bài toán UFL:

#### Biểu diễn:

Trong phần này sẽ giới thiệu việc áp dụng giải thuật tối ưu hóa giải bài toán UFL cụ thể hơn là bài toán Warehouse Location Problem.

Ta biểu diễn mỗi một cá thể trong quần thể bằng 3 vector:

+ Vị trí X : Xi = [xi1, …., xin]. Biểu diễn vị trí của cá thể thứ i trong không gian tìm kiếm.

+ Vận tốc V: Vi = [vi1, … , vin]. Biểu diễn vận tốc của cá thể thứ i.

+ Lời giải Y : Yi = [yi1, …. , yin] Biểu diễn lời giải của bài toán dưới dạng nhị phân. Với cơ sở i được chọn thì yi sẽ nhận giá trị 1 và ngược lại sẽ nhận giá trị 0.

#### Khởi tạo:

Chúng ta sẽ tiến hành khởi tạo các giá trị của X và V như sau:

xij = xmin + (xmax – xmin) \* r1.

vij = vmin + (vmax – vmin) \* r2.

Trong đó chúng ta chọn giá trị xmin = -10, xmax = 10, vmin = -4, vmax = 4. Các giá trị r1 , r2 là các giá trị được sinh ngẫu nhiên trong khoảng [0,1].

Vector vị trí X đại diện cho các giá trị vị trí của các cơ sở tại những thời điểm khác nhau nhưng nó không biểu diễn lời giải cho bài toán. Như vậy chúng ta biểu diễn lời giải cho bài toán bằng vector Y.

Chúng ta tính toán vector Y như sau:

yi =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vecto của cá thể thứ i | Tại thời điểm thứ k | | | | |
|  | 1 | 2 | 3 | 4 | 5 |
| Vector vị trí *(Xi)* | 1.8 | 3.01 | -0.99 | 0.72 | -5.45 |
| Vector vận tốc *(Vi)* | -0.52 | 2.06 | 3.56 | 2.45 | 1.45 |
| Vector lời giải *(Yi)* | 1 | 1 | 0 | 0 | 1 |

Ta có ma trận chi phí và xây dựng như sau:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Vị trí các cơ sở | | **1** | **2** | **3** | **4** | **5** |
| Chi phí xây dựng | | 12 | 5 | 3 | 7 | 9 |
| Khách hàng | **1** | 2 | 3 | 6 | 7 | 1 |
| **2** | 0 | 5 | 8 | 4 | 12 |
| **3** | 11 | 6 | 14 | 5 | 8 |
| **4** | 19 | 18 | 21 | 16 | 13 |
| **5** | 3 | 9 | 8 | 7 | 10 |
| **6** | 4 | 7 | 9 | 6 | 0 |

Hàm mục tiêu của bài toán sẽ được tính bằng cách sau:

Chi phí xây dựng + min(chi phí vận chuyển đến mỗi khách hàng) =

{ (12 + 5 + 9) + min(2,3,1) + min(0,5,12) + min(11,6,8) + min(19,18,13) +min(3,9,10) + min(4,7,0)} = 26 + 1 + 0 + 6 + 13 + 3 + 0 = 49.

Tại mỗi thời điểm chúng ta cập nhật giá trị Pbest chính là cá thể có chi phí nhỏ nhất.

G­best là giá trị nhỏ nhất của Pbest tính đến thời điểm hiện tại.

#### Tìm kiếm:

Sau khi cập nhật các giá trị Pbest và G­­best chúng ta bắt đầu cập nhật vị trí của mỗi cá thể như sau:

vi(t+1) = \* vi(t) + c1 \* r­­­1 \* (Pbest – xi(t)) + c2 \* r2 \* (Gbest – xi(t))).

và

xi(t) = xi(t) + vi(t).

Chúng ta tiếp tục cập nhật giá trị của Y theo X ta sẽ được một lời giải mới.

#### Kết thúc:

Sau khi đạt điều kiện dừng chúng ta đưa ra giá trị Gbest chính là giá trị tối ưu.

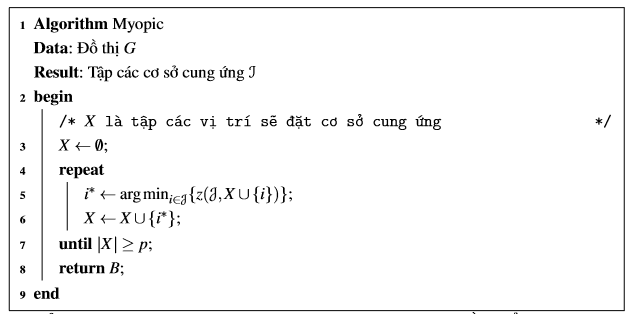
Mô tả giải thuật theo Eberhart và Kennedy như sau:

Begin  
 Khởi tạo ngẫu nhiên các giá trị  
 Với mỗi cá thể  
 Tính toán chọn các ứng cử viên  
 Từ ứng cử viên tìm được tính chi phí.  
 Cập nhật giá trị Pbest   
 Cập nhật giá trị Gbest  
 End  
 Do {  
   
 Với mỗi cá thể  
 Tính vector vận tốc mới.  
 Cập nhật vector vị trí  
 Cập nhật vector lời giải  
 Tính toán chi phí  
 Cập nhật lại các giá trị Pbest, Gbest   
 } While (Điều kiện dừng)   
End

# Chương 3: Kết quả thực nghiệm

## 3.1 Mô tả cài đặt:

### 3.1.1 Giải thuật tham lam (Myopic Algorithm):

Trong giải thuật tham lam, với mỗi đỉnh chúng ta đều phải tính chi phí từ một đỉnh để đáp ứng yêu cầu cho toàn bộ các đỉnh còn lại trên đồ thị. Ta đặt X là tập các vị trí sẽ đặt cơ sở (lời giải cho bài toán). 

Với mỗi đỉnh thực hiện thủ tục tìm Ɀ. Thủ tục tìm Ɀ được tính toán như sau:



Chúng ta duyệt tất cả các điểm không nằm trong lời giải và tính tổng chi phí từ đỉnh đó cho toàn bộ đồ thị. Và lấy ra điểm tối ưu nhất nằm trong bước (5). Đỉnh i\* chính là đỉnh có chi phí cung cấp cho toàn bộ các đỉnh khác là nhỏ nhất. Ta sẽ thêm đỉnh i\* vào tập lời giải và lặp lại thủ tục trên cho tới khi tìm đủ p nhà máy. Điều kiện dừng nằm trong bước số (7).

Chương trình gồm các thủ tục sau:

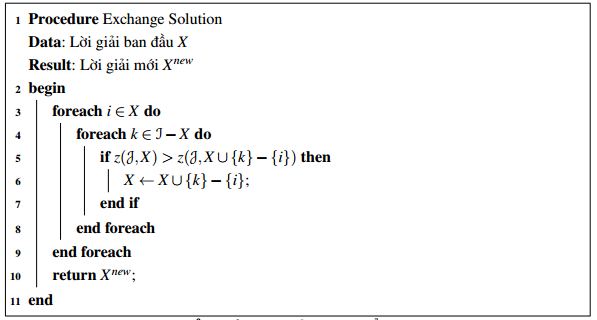
- readData: Thực hiện đọc các giá trị đầu vào: số lượng đỉnh, số lượng median, chi phí. Sau khi đọc toàn bộ chi phí chúng ta thực hiện biến đổi đồ thị thành một đồ thị đầy đủ bằng giải thuật Floyd.

- costValue(): Thực hiện tính chi phí của mỗi lời giải.

- getMin(), myOpic(): Thực hiện tính chi phí từ một đỉnh nào đó trong lời giải đến toàn bộ các đỉnh khác trên đồ thị và đưa ra đỉnh tối ưu nhất.

- print(): In ra giá trị tối ưu và lời giải tối ưu.

### 3.1.2 Giải thuật trao đổi (Exchange Algorithm):

Khác với giải thuật tham lam giải thuật trao đổi không tìm đỉnh tối ưu cục bộ trong toàn đồ thị (1-Median) mà tìm các lời giải mới bằng cách thay một đỉnh bất kì trong lời giải cũ bằng một đỉnh mới trên đồ thị. Giải thuật được thực hiện theo sơ đồ dưới đây:

Tại bước (1) Chúng ta khởi tạo một lời giải cho bài toán. Từ lời giải này sẽ tìm kiếm đến lời giải tối ưu. Lời giải ban đầu không ảnh hưởng đến kết quả tìm kiếm.

Chúng ta thực hiện thủ tục trao đổi từ bước (2) đến bước (8).

Với mỗi đỉnh trong lời giải chúng ta thay thế bằng một đỉnh bất kì trong đồ thị nếu đỉnh đó làm cho lời giải tốt hơn. Việc thực hiện thay thế được mô tả trên bước (6).

Sau khi duyệt toàn bộ lời giải và toàn bộ các đỉnh trên đồ thị nếu không làm tăng giá trị hàm mục tiêu (giảm chi phí) thì chúng ta kết thúc giải thuật và đưa ra lời giải mới.

Chương trình gồm các thủ tục chính sau đây:

- readData(): Đọc dữ liệu đầu vào.

- costValue(): tính chi phí của lời giải.

- exchange(): trao đổi một đỉnh trong lời giải với một định bất kì khác. nếu đỉnh đó làm giảm chi phí thì chấp nhận và ngược lại.

### 3.1.3 Giải thuật di truyền (Generic Algorithm):

Chương trình gồm các thủ tục chính sau:

- readData(): Thực hiện đọc dữ liệu đầu vào.

- init(): Sinh ngẫu nhiên quần thể với kích thước 100 hoặc 500.

- costValue(): tính chi phí của lời giải.

- getTotalCost(): Tính tổng chi phí của toàn bộ quần thể.

- select(): chọn ngẫu nhiên 2 bố(mẹ) trong quần thể để tiến hành lai ghép. Việc chọn ngẫu nhiên nhưng vẫn dựa theo phương pháp ưu tiên chọn bố mẹ tốt.

- doCross(): Tiến hành lai hai cá thể bố mẹ với nhau tạo ra con mới.

- mutation(): Tiến hành đột biến cá thể mới.

- update(): Cập nhật cá thể mới vào quần thể ban đầu.

Trong chương trình chính chúng ta thực hiện lặp việc chọn và lai ghép các cá thể với nhau theo số lượng nhất định để tìm ra lời giải tối ưu.

### 3.1.4 Giải thuật tối ưu hóa bầy đàn (Partical Swarm Optimization):

Các lớp trong chương trình:

* Person(): Chứa thông tin một cá thể bao gồm vector vị trí X, vector vận tốc V, và vector lời giải Y cùng chi phí là value.
* PSONew(): Chứa toàn bộ các hàm, thủ tục thực hiện giải thuật tối ưu hóa bầy đàn.

Chương trình bao gồm các thủ tục sau:

- readData(): đọc dữ liệu đầu vào.

- init(): khởi tạo quần thể một cách ngẫu nhiên.

- getFitness(), costValue(): Tính giá trị hàm mục tiêu của mỗi cá thể.

- finding(): Mỗi cá thể trong quần thể tiến hành tìm kiếm lời giải tối ưu trong không gian tìm kiếm.

- print(): In ra giá trị tối ưu trong toàn bộ quá trình tìm kiếm.

## 3.2 Dữ liệu thực nghiệm:

Các bộ dữ liệu thực nghiệm được lấy trên thư viện OR-Library đã được các nhà khoa học xây dựng trước đây.

***Mô tả chi tiết:***

* Đối với bài toán p-Median cơ bản:

+ Bộ dữ liệu gồm 40 file từ **OR-Library.**

**+** Các bộ dữ liệu được đánh số theo thứ tự từ 1, 2 ,…, 40 với tên gọi pmed1, pmed2, ….., pmed40.

**+** Mỗi bộ dữ liệu đều theo định dạng như sau :

**Số đỉnh số cạnh số median**

**Với mỗi cạnh:**

**Đỉnh thứ nhất đỉnh thứ hai chi phí giữa hai đỉnh.**

+ Với mỗi bộ dữ liệu chúng ta cần chạy giải thuật Floyd để tạo ra một đồ thị đầy đủ. Với mỗi cạnh i và j đều có chi phí > 0.

+ Bộ dữ liệu nhỏ nhất gồm 100 đỉnh và 200 cạnh, Bộ dữ liệu lớn nhất là 900 đỉnh và 16200 cạnh.

* Đôi với bài toán Uncapacitated warehouse location

**+** Bộ dữ liệu gồm 43 file được có tên theo dạng capX trong đó X là số thứ thự của bộ dữ liệu.

**+** Mỗi bộ dữ liệu đều có dạng như sau:

**Số ứng cử viên Số khách hàng**

**Với mỗi ứng cử viên:**

**Khả năng phục vụ Chi phí xây dựng**

**Với mỗi khách hàng:**

**Yêu cầu Chi phí đối với mỗi cơ sở.**

**+** Bộ dữ liệu nhỏ nhất bao gồm 16 cơ sở và 50 khách hàng, bộ dữ liệu lớn nhất gồm 100 cơ sở và 1000 khách hàng.

## 3.3 Kết quả thực nghiệm:

### 3.3.1: Giới thiệu môi trường thực nghiệm:

Thử nghiệm các giải thuật di truyền và tối ưu hóa bầy đàn được tiến hành chạy 20 lần đối với mỗi bộ dữ liệu khác nhau trên cùng một máy tính Intel Core I5 3210M 2.5Ghz, Ram 8gb, Window 10 professional. Mục đich của thực nghiệm là đánh giá so sánh các giải thuật về cả giá trị tối ưu và thời gian tính toán.

### 3.3.2: Các tiêu chí đánh giá:

Một số tiêu chí đánh giá:

Tỉ lệ phần trăm sai khác giữa giá trị tối ưu và giá trị tìm được:

RPE(Relative Percent Error) =

Trong đó H là giá trị tìm được. O là giá trị tối ưu của bài toán.

Tỉ lệ phần trăm sai khác trung bình:



Trong đó Hi là giá trị tìm được trong lần chạy thứ i. Oi là giá trị tối ưu.

### 3.3.3: Các giá trị thực nghiệm:

Bảng kết quả so sánh giữa các giải thuật:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Giải thuật  Bộ dữ liệu | myOpic  (1) | exChange  (2) | Generic  (3) | (1)  ARPE | (2)  ARPE | (3)  ARPE | Optimal |
| Pmed1 | 5891 | 5819 |  | 0.012 |  |  | 5819 |
| Pmed5 | 1378 | 1355 |  | 0.017 |  |  | 1355 |
| Pmed10 | 1295 | 1267 |  | 0.032 |  |  | 1255 |
| Pmed15 | 1761 | 1736 |  | 0.019 |  |  | 1729 |
| Pmed20 | 1866 | 1905 |  | 0.043 |  |  | 1789 |
| Pmed25 | 1896 | 2153 |  | 0.037 |  |  | 1828 |
| Pmed30 | 2037 |  |  | 0.024 |  |  | 1989 |
| Pmed34 | 3097 |  |  | 0.028 |  |  | 3013 |
| Pmed35 | 10406 |  |  | < 10-3 |  |  | 10400 |
| Pmed37 | 5118 |  |  | 0.012 |  |  | 5057 |
| .Pmed40 | 5190 |  |  | 0.012 |  |  | 5128 |

|  |  |  |
| --- | --- | --- |
| Time  (s) |  |  |
| 0.076 | 0.105 |  |
| 0.152 | 0.92 |  |
| 0.699 | 21.46 |  |
| 2.275 | 160 |  |
| 7.037 | 260.1 |  |
| 16.71 | 249.1 |  |
| 35.69 |  |  |
| 38.63 |  |  |
| 3.31 |  |  |
| 37.03 |  |  |
| 64.5 |  |  |