

APPROXIMATION ALGORITHM AND APPLICATION FOR A NP-HARD PROBLEM SOLVING

Nguyen Dinh Dung

TNU - University of Information Technology and Communication

ARTICLE INFO	ABSTRACT
Received: 23/02/2021	In this paper, we introduce some NP-hard problems and present an approximation algorithm to find a cluster (subset) of the largest cardinality and subset of points of a given cardinality. There is a radically different way of dealing with difficult optimization problems: solve them approximately by a fast algorithm. This approach is particularly appealing for applications where a good but not necessarily optimal solution will suffice. Besides, in real-life applications, we often have to operate with inaccurate data to begin. Under such circumstances, going for an approximate solution can be a particularly sensible choice. Algorithm that is given is a polynomial-time approximation algorithm. This algorithm finds a feasible solution to problems when we consider the problem of searching a subset in a finite set of points of Euclidean space.
Revised: 24/5/2021	
Published: 27/5/2021	
KEYWORDS	
Polynomial-time algorithm	
Approximation algorithm	
NP – Hard	
Euclidean space	
Fast Approximate	

THUẬT TOÁN XẤP XỈ VÀ ỨNG DỤNG TÌM NGHIỆM BÀI TOÁN THUỘC LỚP NP - KHÓ

Nguyễn Đình Dũng

Trường Đại học Công nghệ thông tin và Truyền thông - ĐH Thái Nguyên

THÔNG TIN BÀI BÁO	TÓM TẮT
Ngày nhận bài: 23/02/2021	Trong bài báo này, chúng tôi giới thiệu một số bài toán thuộc lớp NP – khó (NP – Hard) và đề xuất một thuật toán xấp xỉ tìm lời giải cho bài toán tìm tập con lớn nhất, tập con có số phần tử xác định trước. Đối với mỗi bài toán tối ưu tổ hợp, hiện nay có khá nhiều phương pháp hữu hiệu với chi phí khá thấp về thời gian tính toán để tìm lời giải, có thể kể đến như thuật toán xấp xỉ nhanh. Phương pháp này đã được ứng dụng rộng rãi để giải các bài toán mà không yêu cầu đòi hỏi phải tìm được nghiệm chính xác, bởi ngay từ dữ liệu đầu vào của bài toán có thể đã là dữ liệu xấp xỉ. Vì vậy tìm được thuật toán xấp xỉ giải bài toán có chi phí thời gian tính toán thấp mà vẫn đảm bảo độ chính xác theo yêu cầu là mục tiêu của bài báo này cần đạt được. Theo cách tiếp cận này, chúng tôi xây dựng được thuật toán có độ phức tạp về thời gian tính toán là đa thức khi bài toán được xét trong không gian Euclidean và nghiệm tìm được có độ chính xác chấp nhận được.
Ngày hoàn thiện: 24/5/2021	
Ngày đăng: 27/5/2021	
TỪ KHÓA	
Thuật toán đa thức	
Thuật toán xấp xỉ	
NP – khó	
Không gian Euclidean	
Xấp xỉ nhanh	

1. Giới thiệu

Tìm lời giải cho bài toán thuộc lớp bài toán NP - khó là chủ đề được nhiều tác giả trong và ngoài nước quan tâm. Một bài toán được gọi là thuộc lớp bài toán NP - khó nếu bài toán đó không có thuật toán thời gian tính đa thức để giải nó (ngoại trừ trường hợp $P=NP$), mà chỉ có các thuật toán giải trong thời gian hàm mũ [1]. Bài toán tìm tập con chung lớn nhất là bài toán thuộc vào lớp bài toán này và có nhiều ứng dụng đa dạng khác nhau, có thể kể đến như lĩnh vực học máy [2], [3]. Trong lĩnh vực học máy, tiền xử lý dữ liệu là một bước rất quan trọng trong việc giải quyết bất kỳ một vấn đề nào. Hầu hết các bộ dữ liệu được sử dụng trong các vấn đề liên quan đến học máy cần được xử lý, làm sạch và biến đổi trước khi một thuật toán học máy có thể được huấn luyện trên những bộ dữ liệu này. Dữ liệu có thể không được thu thập trực tiếp bởi con người vì các lý do xoay quanh vấn đề về chi phí, cơ sở hạ tầng, con người. Do đó, dữ liệu có thể bị thiếu bởi một sai sót của máy móc, hoặc thực tế nó không tồn tại tại một thời điểm nhất định trong khi thu thập dữ liệu. Vì vậy việc tìm ra tập dữ liệu đủ lớn từ tập dữ liệu thu thập được và thỏa mãn các tính chất theo yêu cầu thực tiễn, đồng thời đảm bảo mô hình học máy không bị ảnh hưởng là rất cần thiết.

Bài toán tìm tập con có thể phát biểu như sau [4]: Trong không gian Euclide $\mathbb{R}^d$, cho tập X gồm N phần tử và hàm mục tiêu $f(U)$ xác định với mọi tập con $U \subseteq X$ và hàm ràng buộc $g(U)$. Cần tìm tập nghiệm tối ưu S sao cho:

$$f(S) = \text{opt}\{f(U) \mid U \subseteq X, g(U)\}$$

Hiện nay, có nhiều thuật toán tìm lời giải chính xác cho bài toán trên, như thuật toán vét cạn, thuật toán nhánh cận [5],... Tuy nhiên, các thuật toán tìm lời giải chính xác thường gặp phải hạn chế về chi phí thời gian tính toán khá cao khi kích thước dữ liệu đầu vào lớn, độ phức tạp thời gian tính toán thường là hàm mũ. Vì vậy, bài báo này tập trung vào nghiên cứu thuật toán xấp xỉ để tìm lời giải với độ chính xác chấp nhận được và độ phức tạp về thời gian tính toán là hàm đa thức.

Để tìm lời giải xấp xỉ cho bài toán, hiện nay, đã có nhiều kết quả được công bố. Năm 1991, Aggarwal [6] đã đề xuất thuật toán xấp xỉ giải bài toán trên với độ phức tạp về thời gian đa thức là $O(dN^{d+1})$. Tuy nhiên, trong trường hợp số chiều của không gian lớn thì chi phí về thời gian tính toán khá cao. Khắc phục nhược điểm này, năm 2012, Kelmanov [7] công bố thuật toán xấp xỉ - 2 với độ phức tạp tính toán là $O(dN^2)$, Shenmaier [8] công bố thuật toán đạt độ chính xác ε cho trước với độ phức tạp thời gian tính toán phụ thuộc liên tục vào sai số là

$$O(dN^{2/\varepsilon+1}(9/\varepsilon)^{3/\varepsilon})$$

Năm 2016, Shenmaier [4] đề xuất thuật toán dựa vào lược đồ Voronoi bậc cao với độ phức tạp tính toán đa thức là $O(dN^{d+1})$.

Về cơ bản những thuật toán trên đều cho độ phức tạp tính toán là đa thức nhưng chi phí thời gian tính toán là khá cao nếu số chiều không gian lớn. Một số thuật toán có thời gian tính toán chấp nhận được nhưng độ chính xác còn hạn chế [9]. Nhằm khắc phục 2 vấn đề này, chúng tôi đề xuất thuật toán xấp xỉ $1/2$ với độ phức tạp tính toán là đa thức. Trước hết chúng tôi trình bày một số bài toán cụ thể cho lớp bài toán trên.

2. Một số bài toán thuộc lớp NP – khó

Khái niệm về các lớp bài toán P, NP, NPC, NP – khó [5].

Định nghĩa 1: Bài toán quyết định là bài toán mà đầu ra chỉ có thể là “Yes” hoặc “No” (Đúng/sai, 0/1, chấp nhận/từ chối).

Định nghĩa 2: Bài toán **A** được gọi là “dẫn về được” bài toán **B** sau thời gian đa thức nếu có một thuật toán đa thức để giải bài toán **B** thì cũng có một thuật toán đa thức để giải bài toán **A**.

Định nghĩa 3: **P** là lớp các bài toán có thể giải được bằng thuật toán đơn định trong thời gian đa thức.

Định nghĩa 4: **NP** là lớp các bài toán có thể giải được bằng thuật toán không đơn định trong thời gian đa thức.

Định nghĩa 5: Một bài toán quyết định A được gọi là **NP** - đầy đủ (NPC) nếu như:
- A là một bài toán trong NP.

- Mọi bài toán trong NP đều có thể quy dẫn được về A .

Định nghĩa 6: Một bài toán A được gọi là **NP - khó** (NP-hard) nếu như sự tồn tại thuật toán đa thức để giải nó kéo theo sự tồn tại thuật toán đa thức để giải mọi bài toán trong NP.

Ký hiệu:

$\|x\|$ là chuẩn Euclide của x trong không gian $\mathbb{R}^d$.

$|S|$ là số phần tử thuộc S .

2.1. Bài toán tìm tập con có số phần tử cho trước

Cho trước $K \in \mathbb{R}$

Bài toán 1A. Tìm tập con $S \subseteq X$ sao cho:

$$f_1(S) = \sum_{x \in S} \|x - c(S)\|^2 \rightarrow \min, \quad (1)$$

trong đó

$$c(S) = \frac{1}{K} \sum_{x \in S} x \quad (2)$$

$$|S|=K \quad (3)$$

Bài toán 2A. Tìm tập con $S \subseteq X$ sao cho:

$$f_2(S) = \sum_{x \in S} \|x - c(S)\|^2 + \sum_{x \in X \setminus S} \|x\|^2 \rightarrow \min, \quad (4)$$

$$|S|=K \quad (5)$$

2.2. Bài toán tìm tập con có số phần tử lớn nhất

Bài toán 1B. Tìm tập con $S \subseteq X$ sao cho:

$$|S| \rightarrow \max, \quad (6)$$

$$g_1(S) = \sum_{x \in S} \|x - c(S)\|^2 \leq \alpha \sum_{x \in X} \|x - c(X)\|^2, \quad (7)$$

trong đó

$$c(X) = \frac{1}{|X|} \sum_{x \in X} x, \quad (8)$$

$$c(S) = \frac{1}{|S|} \sum_{x \in S} x, \quad (9)$$

$$\alpha \in (0, 1) \quad (10)$$

Bài toán 2B. Tìm tập con $S \subseteq X$ sao cho:

$$|S| \rightarrow \max, \quad (11)$$

$$\begin{aligned} g_2(S) &= \sum_{x \in S} \|x - c(S)\|^2 + \sum_{x \in X \setminus S} \|x\|^2 \\ &\leq \alpha \sum_{x \in X} \|x - c(X)\|^2, \end{aligned} \quad (12)$$

Trên đây là các bài toán được đề cập trong phân tích dữ liệu tại các tài liệu [9] và [10]. Kết quả trong [4] đã chỉ ra sự tồn tại nghiệm của các bài toán trên. Để thấy các bài toán 1B, 2B là những bài toán mà về mặt của ràng buộc không phụ thuộc vào tập nghiệm cần tìm mà là những hằng số không âm.

Các bài toán 1B, 2B có thể phát biểu như sau:

Bài toán 2.2B: Trong không gian Euclide $\mathbb{R}^d$ Cho tập $X = \{x_1, x_2, \dots, x_N\}$, số thực dương B , số nguyên dương M . Bài toán đặt ra là liệu có tồn tại tập con S sao cho $|S| \geq M$ và $g_i(S) \leq B$ ($i=1$ hoặc $i=2$).

Các kết quả trong [11] đã chỉ ra rằng, bài toán sau đây (Bài toán 2.2C) là bài toán NPC và bài toán Clique thuộc lớp NPC cũng quy dẫn được về Bài toán 2.2C.

Bài toán 2.2C (Bài toán thuộc lớp NPC): Trong không gian Euclide $\mathbb{R}^d$ Cho tập $X = \{x_1, x_2, \dots, x_N\}$, số thực dương C , số nguyên dương M . Bài toán đặt ra là liệu có tồn tại tập con S sao cho $|S| = M$ và $g_i(S) \leq C$ ($i=1$ hoặc $i=2$).

Dễ thấy Bài toán 2.2B và 2.2C là tương đương, vì vậy Bài toán 2.2B cũng thuộc lớp NPC, từ đó cho thấy các Bài toán 1B, 2B là những bài toán thuộc lớp NP - khó.

3. Thuật toán xấp xỉ $\frac{1}{2}$

Thuật toán 1A (thuật toán tìm lời giải cho Bài toán 1A)

Input: Tập $X = \{x_1, x_2, \dots, x_N\}$ và số nguyên dương K .

Output: Tập con S .

1. for ($i=1; i \leq N; i++$) {

2. for ($j=i; j \leq N; j++$) {

$dist(i, j) = \|x_i - x_j\|$;

$dist(j, i) = dist(i, j)$;

}

3. Với mỗi chỉ số i , sắp xếp $dist(i, j)$ theo thứ tự không giảm. Sau khi sắp xếp ta thu được dãy

$Y = \{y_1, y_2, \dots, y_N\}$;

4. $T(i) = 0$;

5. $j = 1$;

6. $S(i) = \emptyset$;

7. While ($j \leq K$) {

8. $T(i) = T(i) + \|x_i - y_j\|^2$;

9. $S(i) = S(i) \cup y_j$;

10. $j = j + 1$;

}

}

11. $l = 1$;

12. for ($i=1; i \leq N; i++$)

13. if ($T(i) < T(l)$) $l = i$;

14. Return $S(l)$;

Thuật toán 1B (thuật toán tìm lời giải cho Bài toán 1B)

Input: Tập $X = \{x_1, x_2, \dots, x_N\}$ và số thực α

Output: Tập con S

1. $B = \alpha \sum_{x \in X} \|x - c(X)\|^2$;

2. for ($i=1; i \leq N; i++$) {

3. for ($j=i; j \leq N; j++$) {

$dist(i, j) = \|x_i - x_j\|$;

$dist(j, i) = dist(i, j)$;

}

4. Với mỗi chỉ số i , sắp xếp $\text{dist}(i, j)$ theo thứ tự không giảm. Sau khi sắp xếp ta thu được dãy $Y = \{y_1, y_2, \dots, y_N\}$;

5. $T=0$;

6. $j=1$;

7. $S(i)=\emptyset$;

8. While($T + \|x_i - y_j\|^2 \leq B \ \&\& \ j \leq N$) {

9. $T = T + \|x_i - y_j\|^2$;

10. $S(i) = S(i) \cup y_j$;

11. $j = j + 1$;

}

12. $l=1$;

13. for ($i=1; i \leq N; i++$)

14. if ($|S(i)| > |S(l)|$) $l=i$;

15. Return $S(l)$;

Vì tính chất tương đương của Bài toán 1A và Bài toán 1B, nên trong mục này chúng tôi chỉ đánh giá 1 thuật toán, ở đây, chúng tôi chọn thuật toán 1B để đánh giá. Đánh giá độ phức tạp thuật toán 1B, ta lần lượt đánh giá thời gian thực hiện các lệnh theo thứ tự sau:

1. $O(dN)$.
2. $O(N)$.
3. $O(dN)$.
4. Nếu sắp xếp bằng thuật toán Heapsort [5] thì độ phức tạp là $O(N \log N)$.
- 5,6,7. $O(1)$.
8. Trong trường hợp xấu nhất là $O(dN)$.
9. $O(d)$.
- 10,11. $O(1)$.
12. $O(1)$.
13. $O(N)$.
- 14, 15. $O(1)$.

Áp dụng các quy tắc cộng và nhân, ta hoàn toàn thu được độ phức tạp của thuật toán là $O(dN + N(dN + N \log N + 1 + dN(d+1)) + 1 + N)$. Vậy độ phức tạp về thời gian tính toán là $O(N^2(\log N + d^2))$.

Nếu ta gọi $OPT(S)$ là nghiệm tối ưu của bài toán, $H(S)$ là nghiệm xấp xỉ tìm được theo thuật toán thì ta có:

$$|(OPT(S) - H(S)) / OPT(S)| \leq 1/2$$

Vì vậy thuật toán này còn gọi là thuật toán xấp xỉ $1/2$.

Bài toán 2B có thể được giải bằng thuật toán tương tự như thuật toán 1B.

4. Một số kết quả thực nghiệm

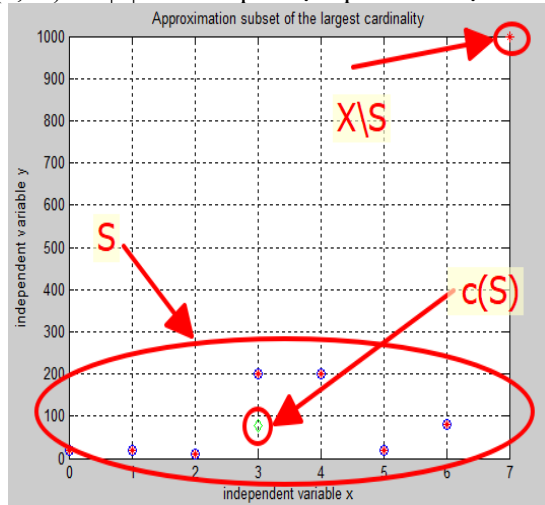
Để minh chứng cho sự đúng đắn, khả thi của thuật toán xấp xỉ trong mục 3. Chúng tôi cài đặt thuật toán 1B trên môi trường Matlab 2014 và chạy chương trình cho một số kết quả sau:

Kết quả thử nghiệm 1: Trong không gian $\mathbb{R}^2$, cho tập $X = \{(0,20), (1,20), (2,10), (3,200), (4,200), (5,20), (6,80), (7,1000)\}$. $\alpha = 1/8$. Nếu sử dụng thuật toán vét cạn thì ta có nghiệm đúng là $S^* = \{(0,20), (1,20), (2,10), (3,200), (4,200), (5,20), (6,80)\}$.

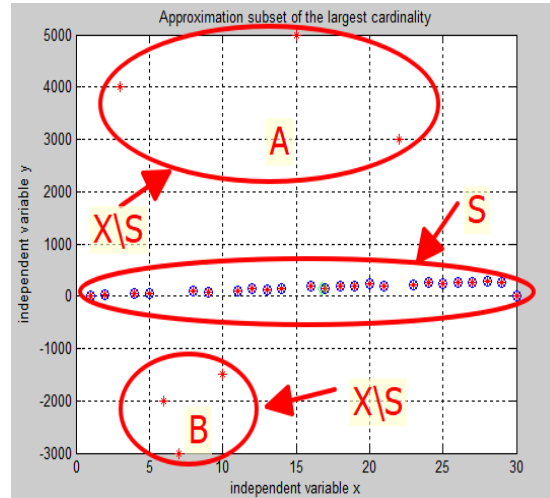
Sau khi chạy thuật toán 1B cho ta kết quả như sau:

$B = 9.8429e+04$; Tập nghiệm tìm được là $S = \{(0,20), (1,20), (2,10), (3,200), (4,200), (5,20), (6,80)\}$ là tập con lớn nhất và có tâm $c(S) = (3, 78.5714)$. Theo thuật toán tâm xấp xỉ được chọn là

(0,20) thì $|S|=7$. Kết quả trực quan thể hiện trên Hình 1.



Hình 1. Kết quả thử nghiệm 1



Hình 2. Kết quả thử nghiệm 2

Về ý nghĩa thực tiễn khi quan sát Hình 1, cho thấy điểm (7,1000) là dữ liệu nhiễu, không đáng tin cậy, nên trong quá trình làm sạch dữ liệu thì dữ liệu này có thể được loại bỏ mà không ảnh hưởng đến mô hình.

Kết quả thử nghiệm 2: Trong không gian $\mathbb{R}^2$, cho tập $X=\{(x,y)\}$ gồm 30 phần tử và được phân bố như Hình 2.

Sau khi chạy thuật toán với $\alpha = 1/30$ thì ta có $B = 2.1060e+06$, kết quả ta thu được nghiệm S là tập con lớn nhất có 24 phần tử. Hình 2 cho thấy dữ liệu thuộc S mô tả mô hình hàm hồi quy tuyến tính, dữ liệu thuộc miền A , miền B là dữ liệu nhiễu không thuộc vào tập dữ liệu đặc trưng, nên dữ liệu này hoàn toàn có thể loại bỏ mà không ảnh hưởng đến mô hình.

5. Kết luận

Trong bài báo này, chúng tôi giới thiệu một số bài toán thuộc lớp NP – khó và xây dựng được thuật toán xấp xỉ $\frac{1}{2}$ tìm lời giải cho bài toán tìm tập con. Thuật toán xây dựng có độ phức tạp là thời gian đa thức và chi phí tính toán tốt hơn các kết quả trước đó. Thuật toán được cài đặt trên môi trường Matlab 2014. Các kết quả thực nghiệm của chương trình là hoàn toàn phù hợp với lý thuyết làm minh chứng khẳng định tính đúng đắn, khả thi của thuật toán.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] M. Garey and D. Johnson, *Computers and Intractability. A Guide to the theory of NP-completeness*. W. H. Freeman and Company, New York, 1979.
- [2] J. W. Osborne, *Best Practices in Data Cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data*, 1st Edition, Los Angeles: SAGE Publication, 2013.
- [3] C. Aggarwal, *Data Mining*. Springer International Publishing, 2015.
- [4] V. V. Shenmaier, "Solving Some Vector Subset Problems by Voronoi Diagrams," *J. Appl. Indust. Math.*, vol. 10, no. 4, pp. 560-566, 2016.
- [5] J. Kleinberg and E. Tardos, *Algorithm Design*. Addison Wesley, first edition, 2006.
- [6] A. Aggarwal, H. Imai, N. Katoh, and S. Suri, "Finding k points with minimum diameter and related problems," *J. Algorithms*, vol. 12, no. 1, pp. 38-56, 1991.
- [7] A. V. Kelmanov and S. M. Romanchenko, "An Approximation Algorithm for Solving a Problem of Search for a Vector Subset," *J. Appl. Indust. Math.*, vol. 6, no. 1, pp. 90-96, 2012.
- [8] V. V. Shenmaier, "An Approximation Scheme for a Problem of Search for a Vector Subset," *J. Appl. Indust. Math.*, vol. 6, no. 3, pp. 381-386, 2012.

- [9] V. V. Shenmaier, "An Approximation Scheme for a Problem of Search for a Vector Subset," *J. Appl. Indust. Math.*, vol. 6, no. 3, pp. 381-386, 2012.
- [10] B. Aronov and S. Har-Peled, "On Approximating the Depth and Related Problems," *SIAM J. Comput.*, vol. 38, no. 3, pp. 899-921, 2008.
- [11] A. V. Kelmanov and A. V. Pyatkin, "NP-Completeness of Some Problems of Choosing a Vector Subset," *J. Appl. Indust. Math.*, vol. 5, no. 3, pp. 352-357, 2011.