BÀI TOÁN LẬP KẾ HOẠCH SẢN XUẤT VỚI CÁC CÔNG VIỆC XUNG ĐỘT LẪN NHAU

Lê Đăng Nguyên
Phòng Khảo thí và Đảm bảo chất lượng
Email: nguyenld@dhhp.edu.vn

Ngày nhận bài: 07/10/2019
Ngày PB đánh giá: 28/10/2019
Ngày duyệt đăng: 25/11/2019

TÓM TẮT: Xuất phát từ các ứng dụng trong thực tế, chúng tôi nghiên cứu bài toán lập kế hoạch sản xuất trong đó các công việc có thể bị xung đột khi thực hiện cùng thời điểm trên các máy khác nhau, và có sự ràng buộc về mặt thời gian. Cụ thể, chúng ta cần phân chia các công việc để xử lý trên các máy giống nhau. Các công việc này có thời gian xử lý giống nhau, nhưng một số có thể bị xung đột khi được thực hiện cùng thời điểm. Tất cả các máy bắt đầu hoạt động cùng lúc và phải kết thúc tại một thời điểm cho trước. Do giới hạn về mặt thời gian nên sẽ có thể có một số công việc không được xử lý. Yêu cầu đặt ra là cần tìm một số lượng lớn nhất công việc có thể xử lý được. Trong bài báo này, chúng tôi đưa ra hai kết quả NP- khó cho bài toán được nghiên cứu trong trường hợp nhóm các công việc xung đột có kích thước tối đa bằng 3. Chúng tôi cũng thảo luận một số câu hỏi mở, có thể là các hướng nghiên cứu tiếp theo trong tương lai.

Từ khoá: *Lập kế hoạch sản xuất, xung đột, đồ thị, độ phức tạp, NP-khó*

A STUDY ON THE COMPLEXITY OF A MACHINE SCHEDULING PROBLEM WITH CONFLICTING JOBS AND TIME CONSTRAINT

ABSTRACT: Motivated by real-world applications, we consider in this paper the following machine scheduling problem in which heterogeneous jobs needed to assign to a set of different machines at the same time under time constraint may conflict. Specifically, we need to distribute the jobs to the identical machines. These jobs have equal processing times, but some of them may be conflicting and therefore they should not be processed concurrently. So all machines have to start the processing jobs at the same time and must stop at a given time. Due to the time constraint, there could be some jobs not being processed by any certain machines and our goal is thus to choose a maximum number of jobs that can be done by them. We present two NP-hardness results for the machine scheduling problem in the case when sets of conflicting jobs have the size of at most 3. Also, we raise some open questions that would be interesting for future work.

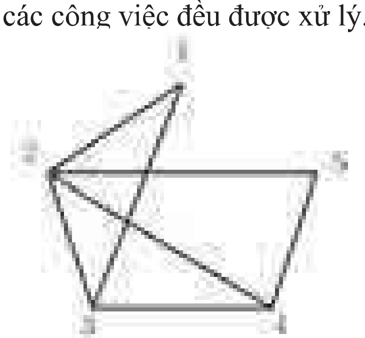
Keywords: *Machine scheduling, conflict jobs, graph, complexity, NP-hardness*

1. GIỚI THIỆU

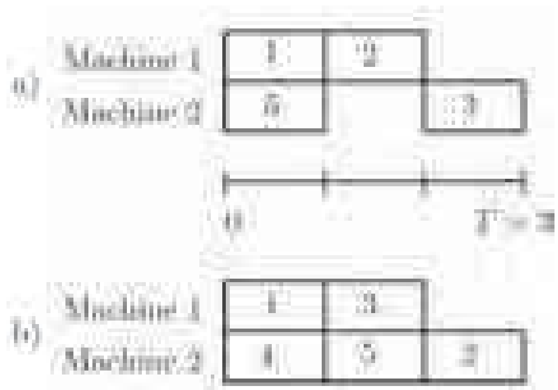
Trong bài báo này chúng tôi nghiên cứu bài toán tối ưu hoá lập kế hoạch sản xuất, trong đó yêu cầu việc phân chia các công việc thực hiện trên các máy sao cho tổng số công việc được thực hiện là nhiều nhất. Bài toán được mô tả cụ thể như sau. Có n công việc cần được phân chia để xử lý trên m máy giống nhau (tức thời gian xử lý cùng một công việc trên các máy là như nhau). Giả thiết rằng các công việc có thời gian xử lý là giống nhau trên tất cả các máy (giả thiết mỗi công việc có độ dài hay thời gian xử lý đều bằng 1), và tại mỗi thời điểm, một máy chỉ xử lý được tối đa một công việc. Sự xung đột giữa các công việc được mô tả bằng một siêu đồ thị (*hypergraph*) $H = (X, E)$, trong đó X là tập các đỉnh tương ứng với các công việc, E là một tập con khác rỗng của 2^X. Đặt $k = max_{e \in E}|e|$, trong đó $|e|$ là số đỉnh kề với cạnh e. Mỗi cạnh của H biểu diễn một tập con các công việc xung đột lẫn nhau, những công việc này sẽ không thể xử lý song song trên các máy khác nhau. Ta định nghĩa một phép phân chia công việc cho các máy như sau:

Định nghĩa 1. Một phép phân chia công việc (gọi tắt là lời giải) là một phép lựa chọn các công việc cụ thể để xử lý trên mỗi máy và khoảng thời gian tương ứng sao cho tại mỗi thời điểm bất kì, các công việc đang được xử lý trên các máy là không xung đột lẫn nhau.

Ví dụ 1. Xét ví dụ sau đây với $m = 2$, $n = 5, k = 2$ và $T = 3$. Đồ thị xung đột G được cho bởi Hình 1 dưới đây. Hình 1a cho ta một phép phân chia công việc cho các máy trong đó tổng số công việc được xử lý là 4. Dễ thấy rằng nếu công việc 1 và 5 được xử lý cùng thời điểm trên hai máy thì sẽ có ít nhất một công việc trong số 3 công việc còn lại không được xử lý, lý do bởi $T = 3$. Hình 2b đưa ra một giải pháp phân chia khác trong đó tất cả các công việc đều được xử lý.



Hình 1: Đồ thị xung đột trong ví dụ 1.



Hình 2: Mô phỏng hai phép phân chia công việc có thể. Các công việc được thực hiện trên mỗi máy theo thứ tự từ trái sang phải.

Ha và các cộng sự [10] trình bày một ứng dụng thực tế trong mạng máy tính. Ở đây, chúng tôi đưa ra thêm một ứng dụng của bài toán trên, cụ thể là bài toán cắm trại, được phát biểu như sau. Một nhóm gồm m sinh viên cần thực hiện các khâu chuẩn bị cho việc cắm trại. Để thực hiện được việc này, họ cần phải xử lý một tập hợp gồm n công việc khác nhau. Mỗi một công việc được xử lý bởi một sinh viên và cần sử dụng một số lượng các dụng cụ nhất định. Một nhóm công việc là xung đột, tức là không thể thực hiện được đồng thời, nếu chúng cần sử dụng chung một số dụng cụ nào đó. Do thời gian là có hạn nên cần phải bố trí các công việc cho sinh viên thực hiện sai cho tại mỗi thời điểm, không có công việc nào xung đột nhau, và tổng số công việc được xử lý là nhiều nhất có thể.

2. TỔNG QUAN NGHIÊN CỨU

Bài toán lập kế hoạch sản xuất với các công việc xung đột lẫn nhau đã được nghiên cứu bởi một số tác giả trước đây, tuy nhiên hầu hết đều xem xét hàm mục tiêu khác và không có ràng buộc về thời gian. Ứng dụng của bài toán này có thể tìm hiểu thêm trong các tài liệu [1,2,4-7,11,12]. Có thể thấy rằng độ phức tạp của bài toán phụ thuộc chủ yếu vào cấu trúc của đồ thị G biểu diễn sự xung đột giữa các công việc. Bằng cách rút gọn từ bài toán PARTITION[(1)], ta có thể chỉ ra rằng bài toán là NP-khó dạng yếu, ngay cả khi đồ thị H là rỗng, tức là các không có bất kì sự xung đột nào giữa các công việc (Garey và Johnson [9]). Với đồ thị G tổng quát, Baker và Coffman [1] chứng minh rằng bài toán là NP-khó dạng mạnh trong trường hợp có các máy không nhỏ hơn 3. Kết quả này đúng cho nhiều lớp đồ thị khác nhau, bao gồm: phần bù của đồ thị so sánh (*comparability graph*) với $m \geq 3$, đồ thị khoảng (*interval graph*) với $m \geq 4$, đồ thị hoán vị (*permutation graph*) với $m \geq 6$. Baker và Coffman cũng xem xét trường hợp khi các công việc có thời gian xử lý như nhau và đưa ra một thuật toán thời gian đa thức để giải, dựa trên thuật toán ghép cặp trên đồ thị. Trong trường hợp đồ thị tổng quát, Even và các cộng sự [6] đưa ra một thuật toán thời gian đa thức cho trường hợp độ dài của các công việc chỉ nhận giá trị 1 hoặc 2. Tuy nhiên nếu cho phép độ dài công việc có thể nhận thêm giá trị là 3, thì bài toán trở thành bài toán NP-khó (Bendraouche và Boudhar [2]), thậm chí ngay cả khi đồ thị xung đột được có dạng đồ thị hai phần (*bipartite graph*). Even và các cộng sự [6] thiết kế thuật toán xấp xỉ với hệ số xấp xỉ $4/3$ cho trường hợp này, trong khi thuật toán của Garey và Graham [8] thậm chí có hệ số xấp xỉ bằng $3/2$. Họ cũng chứng minh rằng nếu thời gian xử lý các công việc là không giống nhau, và nhận giá trị trong miền $\{1,2,3,4\}$ thì bài toán là APX-khó. Bendraouche, Boudhar và Oulamara [3] chứng minh rằng kết quả của Bendraouche và Boudhar [2] vẫn đúng khi thời gian xử lý công việc chỉ nhận hai giá trị khác nhau cho trước bất kì. Thuật toán xấp xỉ trong trường hợp tổng quát với độ dài công việc tuỳ ý được cho bởi [8], với hệ số xấp xỉ bằng $(m+1)/2$, sử dụng thuật toán tham lam (*greedy list scheduling algorithm*).

Gần đây, Ha và các cộng sự [10] nghiên cứu bài toán lập kế hoạch sản xuất với công việc xung đột, với hàm mục tiêu tối đa hoá số công việc được xử lý, giả thiết rằng thời gian thực cho các máy hoàn thành công việc bị giới hạn. Cụ thể, các tác giả tập trung xem xét trường hợp $k = 2$, và chứng minh một số kết quả về độ phức tạp của bài toán trong trường hợp này. Đặc biệt, các tác giả đã đề xuất hai mô hình MILP (*Mixed Integer Linear Programming*) cho bài toán ở dạng tổng quát và chỉ ra rằng các mô hình này có thể giúp giải quyết hiệu quả bài toán trên máy tính với kích thước lớn.

3. KẾT QUẢ NGHIÊN CỨU

Trong phần này chúng tôi sẽ trình bày hai kết quả NP-khó cho bài toán lập kế hoạch sản xuất, trong trường hợp $k = 3$.

Định lý 1. Bài toán CSPCJ (Capacitated Scheduling Problem with Conflict Jobs - bài toán lập kế hoạch sản xuất với các công việc xung đột) là NP-khó khi $m = 3$ và $k = 3$.

Chứng minh. Ta chứng minh rằng bài toán quyết định của CSPCJ, kí hiệu bởi Q-CSPCJ, là NP-đầy đủ. Bài toán này được phát biểu như sau:

Đầu vào:	Hai số nguyên dương n, m, T, siêu đồ thị H, và một số dương $r > 0$.
Câu hỏi:	Liệu có tồn tại một phép phân chia công việc trên các máy sao cho tổng số các công việc được xử lý ít nhất bằng r?

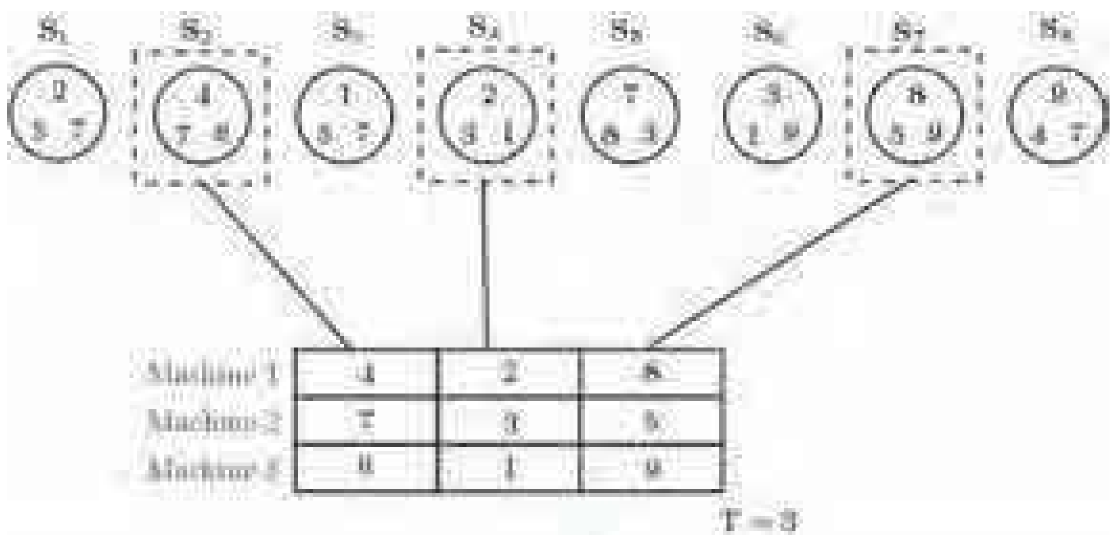
(1) Bài toán Partition được phát biểu như sau: cho tập hữu hạn các số nguyên dương $S = \{a_1, \ldots, a_n\}$, hỏi rằng có thể chia tập S thành hai tập con trong đó tổng các phần tử trong mỗi tập con là bằng nhau?

Trước hết, dễ thấy rằng bài toán Q-CSPCJ thuộc lớp NP. Bởi vì, cho trước một phép phân chia, ta có thể kiểm tra trong thời gian đa thức xem số công việc được xử lý trong phép phân chia này là lớn hơn hay nhỏ hơn hoặc bằng r. Để chứng minh Q-CSPCJ là NP-khó, ta sử dụng phép quy dẫn từ bài toán NP-đầy đủ EXACT COVER BY THREE-SETS (kí hiệu bởi X3C). Bài toán X3C được phát biểu như sau: Cho A là một tập hợp gồm $3q$ phần tử, và $S = \{S_1, S_2, \dots, S_p\}$ là tập con khác rỗng của 2^A, bao gồm các tập con gồm đúng 3 phần tử của A. Hỏi rằng liệu có tồn tại tập con S^* của S sao cho mỗi phần tử của A được tìm thấy trong đúng một phần tử của S^*?

Gọi $I = (A, S)$ là đầu vào đối với bài toán X3C, ta xây dựng đầu vào tương ứng I' của bài toán Q-CSPCJ như sau. Đặt $n = 3q, m = 3, T = q, r = 3q$. Ta định nghĩa đồ thị xung đột $H = (A, E)$, trong đó $e \in E$ nếu $e \neq S_i$ với mọi $i = 1,2,\dots,p$. Nói cách khác mỗi tập S_i biểu diễn nhóm các công việc không xung đột lẫn nhau. Dễ thấy rằng cách xây dựng I' từ I như trên có thể thực hiện trong thời gian đa thức. Ta chứng minh rằng tồn tại một lời giải S^* tương ứng với đầu vào $I = (A, S)$ nếu và chỉ nếu tồn tại một phép phân chia công việc cho bài toán Q-CSPCJ tương ứng với đầu vào I'. Thật vậy, giả sử tồn tại một lời giải S^*, ta xây dựng một phép phân chia công việc như sau. Với mỗi phần tử $P \in S^*$, ta chia các phần tử của P cho 3 máy (tức là mỗi máy một công việc), và tiến hành xử lý các công việc này song song. Dễ thấy rằng số công việc được xử lý đúng bằng $r = 3q$. Ngược lại, nếu tồn tại một phép phân chia công việc sao cho $r = 3q$, ta cũng dễ dàng xây dựng được một lời giải cho bài toán X3C.

Từ kết quả trên suy ra rằng Q-CSPCJ là NP-khó, do X3C là NP-khó. Do Q-CSPCJ là bài toán quyết định của CSPCJ nên CSPCJ là NP-khó. □

Ví dụ 2: Xét ví dụ minh hoạ sau với $q = 3, n, 9, m = 3, p = 8$. Các tập $S_1, S_2, \dots, S_8$ cho như trong hình 3 dưới đây. Các tập con $\{S_2, S_4, S_7\}$ lập thành một lời giải cho bài toán X3C. Từ lời giải này ta xây dựng được một phép phân chia công việc cho các máy như sau: các công việc $4,2,8$ được chia cho máy 1, công việc $3,5,7$ được chia cho máy 2, công việc $1,6,9$ được chia cho máy 3.

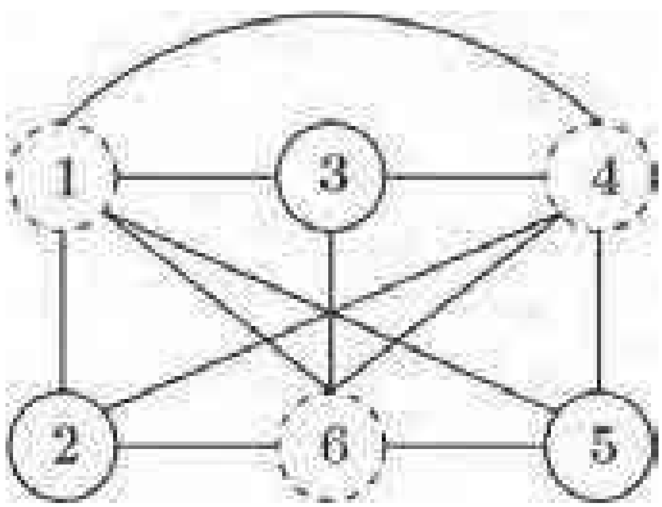


Hình 3: Minh hoạ cách xây dựng lời giải cho bài toán Q-CSPCJ từ X3C

Định lý 2. Bài toán CSPCJ là NP-khó khi $T = 1, k = 3$.

Chứng minh. Sau đây ta sẽ chứng minh rằng bài toán quyết định Q-CSPCJ với $T = 1$, $k = 3$ là NP-đầy đủ, bằng cách chỉ ra một phép quy dẫn từ bài toán NP-đầy đủ INDEPENDENT SET (IS). Bài toán được định nghĩa như sau. Cho đồ thị $G = (V, E)$ và một số dương s, hỏi rằng liệu có tồn tại một tập con V' của V có kích thước s, sao cho không có bất kì hai đỉnh nào trong V' là kề nhau. Gọi $I = (G, s)$ là đầu vào đối với bài toán IS, ta xây dựng đầu vào tương ứng I' của bài toán Q-CSPCJ như sau. Đặt $n = m = |V|, r = s$. Chú ý rằng T cố định bằng 1. Siêu đồ thị $H = (A, F)$ được định nghĩa như sau: tập A đồng nhất với V, với mỗi cạnh $e = (u, v) \in E$, ta định nghĩa $|V| - 2$ cạnh $f = (u, v, t)$, với $t \in A \setminus \{u, v\}$. Dễ thấy rằng I' được xây dựng I trong thời gian đa thức (xem ví dụ minh hoạ trong Hình 4 và Bảng 1 dưới đây).

Ta chứng minh rằng tồn tại một lời giải V' tương ứng với đầu vào $I = (G, s)$ nếu và chỉ nếu tồn tại một phép phân chia công việc cho bài toán Q-CSPCJ tương ứng với đầu vào I'. Thật vậy, giả sử V' có kích thước $|V'| = s$, khi đó dễ thấy rằng tập các công việc tương ứng với các đỉnh trong V' là một lời giải cho bài toán Q-CSPCJ với đầu vào I', vì đây là tập đỉnh độc lập. Chiều ngược lại có thể chứng minh hoàn toàn tương tự. □



***Hình 4:** Đồ thị $G = (V, E)$. Các đỉnh được khoanh bởi hình tròn dạng nét liền lập thành một tập đỉnh độc lập.*

Bảng 1: Biểu diễn siêu đồ thị xung đột $H = (A, F)$ dưới dạng liệt kê.

Siêu đồ thị $H = (A, F)$	
Tập đỉnh A	1,2,3,4,5,6
Tập cạnh F	(1,2,3), (1,2,4), (1,2,5), (1,2,6), (1,3,4), (1,3,5), (1,3,6), (1,4,5), (1,4,6), (2,3,4), (2,3,6), (3,4,5), (3,4,6), (4,5,6)

4. KẾT LUẬN

Chúng ta vừa đi nghiên cứu độ phức tạp tính toán của bài toán lập kế hoạch sản xuất trong đó có sự xung đột giữa một số công việc khi được thực hiện đồng thời trên các máy khác nhau. Yêu cầu của bài toán là cần tìm một số lớn nhất các công việc có thể xử lý. Chúng ta cũng đã đưa ra hai kết quả NP-khó cho bài toán cho trường hợp các nhóm công việc xung đột có kích thước tối đa bằng $k = 3$. Các kết quả trước đây [11] chỉ áp dụng cho trường hợp $k = 2$. Các hướng nghiên cứu tiếp theo có thể tập trung vào các trường hợp sau: i) thời gian xử lý các công việc khác nhau là không giống nhau; ii) các công việc có trọng số khác nhau và yêu cầu tìm tập con các công việc được xử lý có tổng trọng số lớn nhất.

TÀI LIỆU THAM KHẢO

1. B. S. Baker and E. G. Coffman (1996), "*Mutual exclusion scheduling*," Theoretical Computer Science, vol. 162, no. 2, pp. 225–243.
2. M. Bendraouche and M. Boudhar (2012), *"Scheduling jobs on identical machines with agreement graph,"* Computers & OR, vol. 39, no. 2, pp. 382–390.
3. M. Bendraouche, M. Boudhar, and A. Oulamara (2015*), "Scheduling: Agreement graph vs resource constraints,"* European Journal of Operational Research, vol. 240, no. 2, pp. 355–360.
4. H. L. Bodlaender and K. Jansen (1995), *"Restrictions of graph partition problems. part I,"* Theoretical Computer Science, vol. 148, no. 1, pp. 93–109.
5. H. L. Bodlaender and K. Jansen (1993), *"On the complexity of scheduling incompatible jobs with unit-times,"* in Mathematical Foundations of Computer Science 1993, 18th International Symposium, MFCS'93, Gdansk, Poland, August 30 - September 3, 1993, Proceedings, pp. 291–300.
6. G. Even, M. M. Halldórsson, L. Kaplan, and D. Ron (2009), *"Scheduling with conflicts: online and offline algorithms,"* Journal of Scheduling, vol. 12, no. 2, pp. 199–224.
7. F. Gardi (2009), *"Mutual exclusion scheduling with interval graphs or related classes, part I,"* Discrete Applied Mathematics, vol. 157, no. 1, pp. 19–35.
8. M. R. Garey and R. L. Graham (1975), *"Bounds for multiprocessor scheduling with resource constraints,"* SIAM Journal of Computing, vol. 4, no. 2, pp. 187–200.
9. M. Garey and D. Johnson (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company.
10. M. H. Ha, D. M. Vu, Y. Zinder, and T. T. Nguyen (2019), *"On The Capacitated Scheduling Problem with Conflict Jobs,"* in Proceedings of the 11th IEEE International Conference on Knowledge and Systems Engineering (KSE), October 24-26, 2019, Da Nang, Vietnam.
11. M. M. Halldórsson, G. Kortsarz, A. Proskurowski, R. Salman, H. Shachnai, and J. A. Telle (2003), *"Multicoloring trees,"* Information and Computation, vol. 180, no. 2, pp. 113-129.
12. K. Jansen (2003*), "The mutual exclusion scheduling problem for permutation and comparability graphs,"* Information and Computation, vol. 180, no. 2, pp. 71–81.