TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH

HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE

Tập 17, Số 3 (2020): 547-554

Vol. 17, No. 3 (2020): 547-554

ISSN:
1859-3100

Website: *http://journal.hcmue.edu.vn*

Research Article

GREEDY ALGORITHMS FOR OPTIMIZATION PROBLEMS ON UNIT INTERVAL GRAPHS

Nguyen Ngoc Trung

Ho Chi Minh University of Education

Corresponding author: Nguyen Ngoc Trung – Email: trungnn@hcmue.edu.vn

Received: March 01, 2020; Revised: March 22, 2020; Accepted: March 29, 2020

ABSTRACT

In this paper, we show a very special property of unit interval graphs and use that property to introduce greedy algorithms for classical optimization problems on them: finding minimum dominating set, maximum independent set and maximum matching. These algorithms are all linear-time concerning the number of vertices of the graph and simple for implementation.

Keywords: unit interval graph; greedy algorithm; optimization problem

1. Introduction

In the history of graph theory, optimal problems such as finding minimum dominating set, maximum independent set and maximum matching are difficult problems. They have been in consideration for decades and are all proved to be NP-hard. Therefore solving these problems in the general case is impossible. Luckily, we still can have solutions in some specific cases when graphs are simple enough.

Unit interval graphs (UIGs) are in a special class of intersection graphs in which vertices are unit intervals in a real line. In UIGs, the two vertices are connected if and only if they intersect. In considering optimization problems on UIGs, some authors have shown that these problems could be solved by linear algorithms as follows:

- Chang (1998) introduced linear algorithms for minimum dominating set in UIGs and circular-arc graphs;
- Hsu and Tsai (1991) introduced a linear algorithm for maximum independent set in UIGs;
- We still see no linear algorithm for maximum matching in UIGs.

Cite this article as: Nguyen Ngoc Trung (2020). Greedy algorithms for optimization problems on unit interval graphs. *Ho Chi Minh City University of Education Journal of Science, 17*(3), 547-554.

Among the algorithms mentioned above, none of them is with a greedy approach. Recall that the benefits of a greedy algorithm is that it is not recursive, fast and very easy for implementation. Therefore, we can use them to solve optimization problems, which are NP-hard in the general case, in linear time. Moreover, by having greedy algorithms for optimization problems on UIGs, we can think about continuing to open them to other graph classes such as the class of unit disk graphs and so on.

For that reason, in this paper, we will introduce greedy algorithms for these optimization problems in UIGs. Our contributions are linear-time greedy algorithms for following optimization problems on unit interval graphs:

- Finding a minimum dominating set;
- Finding a maximum independent set;
- Finding a maximum matching.

The paper is organized as follows. In Section 2, we introduce some terminology and notations, give exact definitions, and recall some previous results. In Section 3, we derive structural properties of the unit interval graphs and based on that to propose greedy algorithms for optimization problems on unit interval graphs and their analysis. Finally, in Section 4, we conclude all the results of earlier sections and give some of our further researches.

2. Preliminaries

2.1. Unit interval graphs

Unit interval graphs are a special class of (geometric) intersection graphs. Now, we first consider the definitions of these graphs (Van Leeuwen, 2005).

Definition 2.1.

Let S be a set of geometric objects. Then the graph $G = (V, E)$, where each vertex corresponds to an object in S and two vertices are connected by an edge if and only if the two corresponding objects intersect, is called an intersection graph. The graph G is said to be realized by S.

In this definition, tangent objects are assumed to intersect. We can now formally define (unit) interval graphs. Denote by $x_i \in \boldsymbol{R}$ the center and by r_i the radius of an interval I_i.

Definition 2.2.

A graph G is an interval graph if and only if there exists a set of intervals $I = \{I_i \mid i = 1, \dots, n\}$, such that G is the intersection graph of I. The set of intervals is called an interval representation of G.

An interval graph can be given without its interval representation. In this case, it is assumed that each vertex knows the vertices adjacent to it. However, knowing an interval representation can help to more efficiently solve problems on interval graphs.

Definition 2.3.

A graph G is a unit interval graph if and only if G is an interval graph and the radii of a set of intervals realizing G are equal.

Usually, the common radius is 1, but often it is assumed to be $\frac{1}{2}$.

Observe that (unit) interval graphs are a good model for mobile ad hoc networks limited in 1-dimension space. Each node of the network corresponds to an interval center and the transmission range of a node corresponds to the radius of the interval. In a unit interval graph, all nodes are assumed to have the same transmission range.

2.2. Problem definitions

We consider various classical optimization problems on graphs, relevant to (unit) disk graph models of mobile ad hoc networks.

Definition 2.4.

Let $G = (V, E)$ be a graph. A set $S \subset V$ is an independent set if and only if there are no $u, v \in S$, such that $(u, v) \in E$.

We are usually looking for a ***maximum independent set***. An independent set is maximum if and only if there is no independent set of a greater size.

In the context of mobile ad hoc networks, an independent set of a (unit) interval graph can be seen as a set of nodes that can transmit simultaneously without signal interference.

Definition 2.5.

Let $G = (V, E)$ be a graph. A set $S \subset V$ is a dominating set if and only if for each vertex v either $v \in S$ or there exists a vertex $u \in S$ for which $(u, v) \in E$.

A dominating set in a mobile ad hoc network can be seen as a set of emergency transmitters capable of reaching every node in the network, or as central nodes in node clusters. A connected dominating set can be used as a backbone for easier and faster communications. The problem is to find a ***minimum dominating set***.

Definition 2.6.

Let $G = (V, E)$ be a graph, a matching M in G is a set of pairwise non-adjacent edges, none of which are loops; that is, no two edges share a common vertex.

A vertex is matched if it is an endpoint of one of the edges in the matching. Otherwise, the vertex is unmatched.

A maximum matching (also known as maximum-cardinality matching) is a matching that contains the largest possible number of edges. There may be many maximum matchings. The matching number $\nu(G)$ of a graph G is the size of a maximum matching. Every maximum matching is maximal, but not every maximal matching is a maximum matching. The problem is to find a ***maximum matching***.

3. Greedy algorithms

In this section, we will introduce greedy algorithms for optimization problems on a unit interval graph.

As defined above, we consider a unit interval graph G = (V,E), where:

- $V = \{I_1, I_2, \dots, I_n,\}$, every interval I_i has center $x_i \in R$ and radius $\frac{1}{2}$.
- $E = \{(I_i, I_j) \middle| I_i, I_j \in V, i \neq j \text{ and } |x_i - x_j| \leq 1\}$

Without loss generality, we can suppose that all intervals are sorted by ascending order of its center, that is $\forall i, j \in \{1, \dots, n\}, i < j \Rightarrow x_i \leq x_j$. By this, we have the following property of the unit interval graph:

Lemma 3.1.

For every $i, j, k \in \{1, \dots, n\}$, if $i < k < j \text{ and } (I_i, I_j) \in E$ then $(I_i, I_k) \in E \text{ and } (I_k, I_j) \in E$.

Proof.

Since $(I_i, I_j) \in E$, we have $|x_i - x_j| \leq 1$. Now, from $i < k < j$, we have $x_i \leq x_k \leq x_j$. Therefore, $|x_i - x_k| \leq |x_i - x_j| \leq 1$, which means $(I_i, I_k) \in E$. Similarity, we also have $(I_k, I_j) \in E$. □

The meaning of this lemma is that when we sort vertices by ascending order if two vertices are connected then they are connected to all vertices between them. As a result, all of them form a complete subgraph of G.

Definition 3.1.

Given a vertex $I_i \in V$, the vertex $I_j \in V, j \geq i$, is called the rightmost neighbor of I_i if and only if there is no $I_k \in V$ such that I_k is a neighbor of I_i and $k > j$.

Note that the rightmost vertex of a given vertex always exists and can be itself.

3.1. Algorithm for finding a minimum dominating set

For finding the minimum dominating set of a given unit interval graph, we first sort all vertices by ascending order of its center. Then, by using the property as shown in Lemma 3.1, we scan from the beginning to the end of the sorted vertex list by the following idea:

- For a vertex I_i, find its rightmost neighbor I_j;
- Add I_j to the minimum dominating set because I_j is the furthest vertex that can dominates I_i;
- Find I_k the rightmost neighbor of I_j;
- Now it is easy to see that all vertices from I_i to I_k are connected to I_j, which means, they have been dominated by I_j. Therefore, we continue scanning from I_{k+1}.

In summary, we have the following algorithm for finding a minimum dominating set of the unit interval graph G:

```
Input: A unit interval graph G with vertices are sorted by ascending order of its center.
Output: $M \subset V$ is a minimum dominating set of $G$
Init $M = \varnothing$, the set of dominating set
Init $i = 1$
While $i \leq n$ do
    Find $I_j$, the rightmost neighbor of $I_i$
    $M = M \cup \{I_j\}$
    Find $I_k$, the rightmost neighbor of $I_j$
    $i = k + 1$
EndWhile
```

3.2. Algorithm for finding a maximum independent set

For finding the maximum independent set of a given unit interval graph, we also first sort all vertices by ascending order of its center. Then, by using the property as shown in Lemma 3.1, we scan from the beginning to the end of the sorted vertex list by the following idea:

- For a vertex I_i, add I_i to the maximum independent set;
- Find I_j the rightmost neighbor of I_j;
- Now by Lemma 3.1, all vertices from I_{i+1} to I_j are connected to I_i, which means, they cannot be in the maximum independent set. Therefore, we continue scanning from I_{j+1}.

In summary, we have the following algorithm for finding a maximum independent set of a unit interval graph G:

Input: A unit interval graph G with vertices are sorted by ascending order of its center.

Output: $M \subset V$ is a maximum independent set of G

```
Init $M = \varnothing$, the set of independent set
Init $i = 1$
While $i \leq n$ do
    $M = M \cup \{I_i\}$
    Find $I_j$, the rightmost neighbor of $I_i$
    $i = j + 1$
EndWhile
```

3.3. *Algorithm for finding a maximum matching*

In this situation, we still use the very nice property of the unit interval graph that has been shown in Lemma 3.1, but in a little bit different aspect as in the following corollary.

Corollary 3.2.

For every $i \in \{1, \dots, n-1\}$, if $(I_i, I_{i+1}) \notin E$ then $(I_i, I_k) \notin E$ for every $i < k \leq n$.

Proof.

This corollary could be proved easily by contradiction from Lemma 3.1. □

The meaning of this corollary is that, if a vertex I_i is not connected with its next vertex I_{i+1} then it cannot be neighbor of all vertices to the right of I_{i+1}. This is very important for finding a maximum matching set because now we can just look at matching candidates of two consecutive vertices.

For finding the maximum matching set of a given unit interval graph, using Corollary 3.2, we scan from the beginning to the end of the sorted vertex list and process with the following idea: For a vertex I_i,

- If I_i is connected with I_{i+1} then add $\{I_i, I_{i+1}\}$ to the maximum matching set. After that, continue from I_{i+2};
- Otherwise, just skip it and do nothing because it won't have any neighbor to the right.

In summary, we have the following algorithm for finding a maximum matching set of a unit interval graph G:

```
Input: An unit interval graph G with vertices are sorted by ascending order of its center.
Output: $M \subset V \times V$ is a maximum matching set of G
Init $M = \varnothing$, the set of independent set
Init $i = 1, first = TRUE$
While $i \leq n$ do
        If $first! = TRUE$ then
            If $\{I_{i-1}, I_i\} \in E$ then
                $M = M \cup \{I_{i-1}, I_i\}$
            EndIf
        EndIf
        $first = NOT\ first$
        $i = i + 1$
EndWhile
```

It is easy to see that this algorithm together with the two previous algorithms all have linear complexity because they scan from the beginning to the end of sorted vertices and process each vertex once.

4. Conclusion and further research

We discovered a very nice special property of unit interval graphs and applied it to introduce three linear greedy algorithms for finding the minimum dominating set, maximum independent set and maximum matching set in a unit interval graph. As shown above, these algorithms are very easy to implement.

This result is not trivial because these three problems all are proved to be NP-hard in general graphs. This can open new approaches for optimization problems in other graph classes. We believe that by identifying similar properties of graphs, we can also have greedy algorithms for these classical optimization problems.

For the next research, we continue considering this technique for other more complicated graph classes such as unit disk graphs. We hope that we can have effective algorithms with polynomial complexity for optimization problems on unit disk graphs as well as other classes of graphs.

❖ ***Conflict of Interest:*** *Author has no conflict of interest to declare.*

REFERENCES

Chang, M. (1998). Efficient algorithms for the domination problems on interval and circular-arc graphs. *SIAM Journal of Computing*, *27*(6), 1671-1694.

Heggernes, P., Lokshtanov, D., Mihai, R., & Papadopoulos, C. (2008). Cutwidth of split graphs, threshold graphs, and proper interval graphs. Proceedings of WG 2008, *Lecture Notes in Computer Science*, *5344*, 218-229.

Hsu, W., & Tsai, K. H. (1991). Linear time algorithms on circular-arc graphs. *Information Processing Letters*, *40*(3), 123-129.

Van Leeuwen, E. J., (2005). Approximation Algorithms for Unit Disk Graphs. In: Kratsch D. (eds) Graph-Theoretic Concepts in Computer Science, *Lecture Notes in Computer Science*, *3787*, 351-361.

Yuan, J., & Zhou, S., (1995). Optimal labelling of unit interval graphs. *Applied Mathematics-a Journal of Chinese Universities Series B*, *10*, 337-344.

MỘT SỐ THUẬT TOÁN THAM LAM CHO CÁC BÀI TOÁN TỐI ƯU TRÊN ĐỒ THỊ KHOẢNG ĐỀU

Nguyễn Ngọc Trung

Trường Đại học Sư phạm Thành phố Hồ Chí Minh

Tác giả liên hệ: Nguyễn Ngọc Trung – Email: trungnn@hcmue.edu.vn

Ngày nhận bài: 01-3-2020; ngày nhận bài sửa: 22-3-2020; ngày duyệt đăng: 29-3-2020

TÓM TẮT

Trong bài báo này, chúng tôi chỉ ra một tính chất rất đặc biệt của đồ thị khoảng đều và sử dụng nó để đề xuất một số thuật toán tham lam cho các bài toán tối ưu cổ điển trên lớp đồ thị này bao gồm: tìm tập đỉnh bao quát nhỏ nhất, tìm tập đỉnh độc lập lớn nhất và tìm một cách ghép đôi lớn nhất. Các thuật toán này đều có độ phức tạp tuyến tính theo số đỉnh của đồ thị và có thể được lập trình dễ dàng.

Từ khóa: đồ thị khoảng đều; thuật toán tham lam; các bài toán tối ưu