Optimal Algorithm of Isolated Toughness for Interval Graphs

Fengwei Li[1], Qingfang Ye[1], Hajo Broersma[2], and Xiaoyan Zhang[3(✉)]

[1] College of Basic Science, Ningbo University of Finance and Economics, Ningbo 315327, Zhejiang, People's Republic of China
fengwei.li@hotmail.com, fqy-y@163.com

[2] Faculty of EEMCS, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
h.j.broersma@utwente.nl

[3] School of Mathematical Science and Institute of Mathematics, Nanjing Normal University, Nanjing 210023, Jiangsu, People's Republic of China
zhangxiaoyan@njnu.edu.cn

Abstract. Factor and fractional factor are widely used in many fields related to computer science. The isolated toughness of an incomplete graph G is defined as $i\tau(G) = \min\{\frac{|S|}{i(G-S)} : S \in C(G), i(G-S) > 1\}$. Otherwise, we set $i\tau(G) = \infty$ if G is complete. This parameter has a close relationship with the existence of factors and fractional factors of graphs. In this paper, we pay our attention to computational complexity of isolated toughness, and present an optimal polynomial time algorithm to compute the isolated toughness for interval graphs, a subclass of cocomparability graphs.

Keywords: Isolated toughness · Factor · Fractional factor · Interval graph · Polynomial time algorithm

1 Introduction

Throughout this paper, we use Bondy and Murty [1] for terminology and notations not defined here and consider finite simple undirected graphs only. The vertex set of a graph G is denoted by V and the edge set of G is denoted by E. For $X \subseteq V(G)$, let $\omega(G-X)$ and $i(G-X)$, respectively, denote the number of components, the number of components which are isolated vertices in $G-X$. We use $\delta(G)$ and $\kappa(G)$ to denote the minimum degree and connectivity of G, respectively. For any $X \subseteq V$, denote $G[X]$ to be the subgraph of G induced by X. Let $\kappa(G)$ denotes the connectivity of graph G. A subset $X \subseteq V$ is a *cutset* of a graph $G = (V, E)$ if $G - X$ has more than one component. Note that $X = \emptyset$ is a cutset of G if and only if G is disconnected. We let $C(G)$ denote the set of all cutsets of G. A *clique* of a graph is an induced subgraph that is a complete graph.

Y. Zhang et al. (Eds.): PDCAT 2020, LNCS 12606, pp. 379–388, 2021.
https://doi.org/10.1007/978-3-030-69244-5_34

The study of the factor and fractional factor of graphs is a new problem raised in recent years [21]. Let g and f be two non-negative integer-valued functions defined on $V(G)$ such that $g(x) \leq f(x)$ for each $x \in V(G)$. A *(g, f)-factor* of G is a spanning subgraph H of G such that $g(x) \leq d_H(x) \leq f(x)$ holds for each $x \in V(G)$. Similarly, H is an *f-factor* of G if $g(x) = f(x)$ for each $x \in V(G)$.

Fractional factors can be considered as the rationalization of the traditional factors by replacing integer-valued function by a more generous "fuzzy" function (i.e., a $[0, 1]$-valued indicator function). Fractional factors have wide-ranging applications in areas such file transfer problems in computer networks, timetable problems and scheduling, etc.

In 1973, Chvátal [5] introduced the notion of toughness for studying Hamiltonian cycles and regular factors in graphs. The *toughness* [5] of an incomplete connected graph G is defined as

$$\tau(G) = \min\{\frac{|S|}{\omega(G-S)} : S \in C(G), \omega(G-S) > 1\}.$$

This parameter has become an important graph invariant for studying various fundamental properties of graphs. In particular, Chvátal conjectured that k-toughness implies a k-factor in graphs and this conjecture was confirmed positively by Enomoto et al. [6].

Motivated from Chvátal's toughness by replacing $\omega(G-X)$ with $i(G-X)$ in the above definition, Ma and Liu [15] introduced the *isolated toughness*, $i\tau(G)$, as a new parameter to investigate and discuss the necessary and sufficient condition for a graph to have a (fractional) factor of the graph.

Definition 1 [15]. The *isolated toughness* of an incomplete connected graph G is defined as

$$i\tau(G) = \min\{\frac{|S|}{i(G-S)} : S \in C(G), i(G-S) > 1\},$$

where the maximum is taken over all the cutsets of G. Especially, for complete graph K_n, define $i\tau(K_n) = \infty$.

The following result is basic in fractional factor theory.

Theorem 1 [21]. *A graph G has a fractional 1-factor iff $i(G-X) \leq |X|$ for any $X \subseteq V(G)$.*

Thus, we can easily get the following theorem which provides a characterization for the existence of fractional 1-factors in terms of $i\tau(G)$.

Theorem 2 [22]. *Let G be a graph of order $n \geq 2$. Then G has a fractional 1-factor iff $i\tau(G) \geq 1$.*

Ma and Liu [15] proved that graph G has a fractional 2-factor if $i\tau(G) \geq 2$ and $\delta(G) \geq 2$. Furthermore, they showed that graph G has a fractional k-factor if $i\tau(G) \geq k$ and $\delta(G) \geq k$, and if $\delta(G) \geq i\tau(G) \geq a - 1 + \frac{a}{b}$, then G has a fractional $[a, b]$-factor, where $a < b$ are two positive integers [16]. Ma and Yu [19] proved that if G is a graph with $\delta(G) \geq a$, $i\tau(G) \geq a - 1 + \frac{a-1}{b}$, and $G - S$ has no $(a-1)$-regular component for any subset $S \subseteq V(G)$, then G has an $[a, b]$-factor. For more results about (isolated) toughness condition for existence of (fractional) factor in graphs we refer to [17,18].

In this paper, we discuss the computational complexity of isolated toughness in graphs and we give a polynomial time algorithm to compute isolated toughness for interval graphs.

2 Preliminaries

In this section, we recall some definitions, notations and lemmas which will be used throughout the paper.

First, we define the minimal cutset.

Definition 2 [11]. A subset $X \subseteq V$ of G is called an *a,b-cutset* for nonadjacent vertices a and b of graph G if the removal of X separates a and b in distinct connected components. If no proper subset of X is an a, b-cutset of graph G, then X is called a *minimal a,b-cutset* of G. A *minimal cutset* X of G is a set of vertices such that X is a minimal a, b-cutset for some nonadjacent vertices a and b.

The following Lemma provides an easy test of whether or not a given vertex set X is a minimal cutset [12].

Lemma 1 [12]. *Let X be a cutset of the graph $G = (V, E)$. Then X is a minimal cutset if and only if there are at least two different connected components of $G-X$ such that every vertex of X has a neighbor in both of these components.*

For $k \in \{0, 1, 2 \ldots, n\}$ we define $i_k(G)$ as the maximum number of isolated vertices the graph G can obtain after accurately removing k vertices from G, i.e., $i_n(G) = 0$ and for $k < n$

$$i_k(G) = max\{i(G - S) : S \subseteq V, |S| = k\}.$$

It is easy to see that for any incomplete graph G, we have

$$i\tau(G) = \min\{\frac{k}{i_k(G)} : i_k(G) > 1\}.$$

The following theorem give a formula to compute the $i_k(G)$ for $k \in \{0, 1, 2 \ldots, n\}$.

Theorem 3. *Let G be an incomplete graph and let $k \in \{0, 1, 2 \ldots, n\}$. If $i_k(G) > 1$ then*

$$i_k(G) = \max_{|X^*| \leq k} \max_{0 \leq r_{h+1}, r_{h+2}, \ldots, r_p \leq n} \left\{ \sum_{j=h+1}^{p} i_{r_j}(G[C_j]) + h : \sum_{j=h+1}^{p} r_j = k - |X^*| \right\},$$

where the maximum is taken over all minimal cutsets X^* of the graph G, and over all nonnegative integer vectors $(r_{h+1}, r_{h+2}, \ldots, r_p)$. Furthermore, $C_1, C_2, \ldots, C_h$ are the components of $G - X^*$ which are isolated vertices, and $C_{h+1}, C_{h+2}, \ldots, C_p$ are the connected components of $G - X^*$ which are not isolated vertices.

Proof. First let X be a cutset of G with $|X| = k$ and $i(G-X) = i_k(G) > 1$. Let X^* be a minimal cutset of G that is a subset of X, we suppose $C_1, C_2, \ldots, C_h$ be the components of $G-X^*$ which are isolated vertices, and let $C_{h+1}, C_{h+2}, \ldots, C_p$ be the connected components of $G - X^*$ which are not isolated vertices. Then $C_1, C_2, \ldots, C_h$ are also the components of $G - X$, so, we consider the sets $X_j = X \cap C_j$, $j \in \{h+1, h+2, \ldots, p\}$. Then, we know that

$$i_k(G) = i(G - X) = \sum_{j=h+1}^{p} i(G[C_j - X_j]) + h \leq \sum_{j=l+1}^{p} i_{|X_j|}(G[C_j]) + h$$

$$\leq \max_{|X^*| \leq k} \max_{0 \leq r_{h+1}, r_{h+2}, \ldots, r_p \leq n} \left\{ \sum_{j=h+1}^{p} i_{r_j}(G[C_j]) + h : \sum_{j=h+1}^{p} r_j = k - |X^*| \right\}.$$

On the other hand, let X^* be a minimal cutset of G. Furthermore let $C_1, C_2, \ldots, C_h$ be the components of $G - X^*$ which are isolated vertices, and let $C_{h+1}, C_{h+2}, \ldots, C_p$ be the connected components of $G - X^*$ which are not isolated vertices. Let $(r_{h+1}, r_{h+2}, \ldots, r_p)$ be a vector making the right hand side of the above formula to be maximal. For every $j \in \{h+1, h+2, \ldots, p\}$, we choose a set X_j of $G[C_j]$ such that $|X_j| = r_j$ and $i_{r_j}(G[C_j]) + h = i(G[C_j - X_j]) + h$. Thus, $X = X^* \cup (\cup_{j=h+1}^{p} X_j)$ is a subset of G and

$$|X| = |X^*| + \sum_{j=h+1}^{p} |X_j| = |X^*| + \sum_{j=h+1}^{p} r_j = k.$$

Furthermore, we have

$$\max_{|X^*| \leq k} \max_{0 \leq r_{h+1}, r_{h+2}, \ldots, r_p \leq n} \left\{ \sum_{j=h+1}^{p} i_{r_j}(G[C_j]) + h : \sum_{j=h+1}^{p} r_j = k - |X^*| \right\}$$

$$= \sum_{j=h+1}^{p} i_{r_j}(G[C_j]) + h = i(G[C_j - X_j]) + h \leq i_k(G).$$

This completes the proof. ∎

Theorem 4. *Let G be an incomplete graph, let X^* be a minimal cutset of G and let $C_1, C_2, \ldots, C_h$ are the components of $G - X^*$ which are isolated vertices, and $C_{h+1}, C_{h+2}, \ldots, C_p$ are the connected components of $G - X^*$ which* *are not isolated vertices. For every* $j \in \{h+1, h+2\dots, p\}$, *let the list* H_j *be* $(i_0(G[C_j]), i_1(G[C_j]), \dots, i_{|C_j|}(G[C_i]))$. *There is an algorithm computing*

$$\max_{0\leq r_{h+1}, r_{h+2}, \dots, r_p \leq n} \left\{ \sum_{j=h+1}^{p} i_{r_j}(G[C_j]) + h : \sum_{j=h+1}^{p} r_j = k - |X^*| \right\},$$

for every $k \geq |X^*|$ *from the list* $(H_{h+1}, H_{h+2}, \dots, H_p)$ *in time* $O(n^3)$.

Proof. Let X^* be a minimal cutset of G, we suppose $C_1, C_2, \dots, C_h$ be the components of $G - X^*$ which are isolated vertices, and let $C_{h+1}, C_{h+2}, \dots, C_p$ be the connected components of $G - X^*$ which are not isolated vertices. let $i_j^{(r)}(G - X^*)$ $(h+1 \leq r \leq p)$ to be the largest number of isolated vertices of the graph $G[\cup_{j=h+1}^{p} C_j]$ after the deleting of $k - |X^*|$ vertices in $\cup_{j=h+1}^{r} C_j$. Thus, $i_j^{(p)}(G - X^*)$ is exactly the largest number of isolated vertices of $G[\cup_{j=h+1}^{p} C_j]$ can have after the removal of $k - |X^*|$ vertices in $\cup_{j=h+1}^{p} C_j$ which is exactly

$$\max_{0\leq r_{h+1}, r_{h+2}, \dots, r_p \leq n} \left\{ \sum_{j=h+1}^{p} i_{r_j}(G[C_j]) + h : \sum_{j=h+1}^{p} r_j = k - |X^*| \right\},$$

Let the list $L^{(r)}$ be

$$(i_0^{(r)}(G - X^*), i_1^{(r)}(G - X^*), \dots, i_{|\cup_{j=h+1}^{t} C_j|}^{(r)}(G - X^*)),$$

$h+1 \leq r \leq p$. Then $L^{(h+1)} = H_{h+1}$.

Furthermore, the algorithm iteratively computes for $r = h+2, h+3, \dots, p$ the list $L^{(r)}$ from $L^{(r-1)}$ and H_r by using

$$i_k^{(r)}(G - X^*) = \max \left\{ i_a^{(r-1)}(G - X^*) + i_b^{(r)}(G[C_r]) : a + b = k \right\}.$$

The calculation of an entry $i_k^{(r)}(G - X^*)$ can be completed in time $O(n)$. Hence, we can compute all $O(n^2)$ entries in time $O(n^3)$ by the algorithm.
This completes the proof. ∎

3 Isolated Toughness for Interval Graphs

An undirected graph G is called an *interval graph* if its vertices can be put into one to one correspondence with a set of intervals ℓ of a linearly ordered set (like the real line) such that two vertices are connected by an edge if and only if their corresponding intervals have nonempty intersection [9].

Interval graphs are a well-known family of perfect graphs with plenty of nice structural properties [4,7,9,10,20]. Kratsch et al. [13] computed the toughness and the scattering number for interval and other graphs. Li and Li [14] proved the problem of computing the neighbor scattering number of an interval graph can be solved in polynomial time. Broersma et al. [3] gave linear-time algorithms for computing the scattering number and Hamilton-connectivity of interval graphs. In this section, we prove that there exists polynomial time algorithm for computing isolated toughness of an interval graph.

The following lemmas give some useful properties of interval graphs.

Lemma 2 [9]. *Any induced subgraph of an interval graph is an interval graph.*

Lemma 3 [2]. *Any interval graph with order n and size m can be recognized in $O(m+n)$ time.*

Lemma 4 [8]. *A graph G is an interval graph if and only if the maximal cliques of G can be linearly ordered, such that, for every vertex v of G, the maximal cliques containing v occur consecutively.*

We call such a linear ordering of the maximal cliques of an interval graph a *consecutive clique arrangement*. Booth and Lueker [2] give a linear time PQ-tree algorithm for interval graphs, meanwhile, this algorithm can compute a consecutive clique arrangement of the interval graph too.

The following lemma determine the minimal cutsets of an interval graph.

Lemma 5 [13]. *Let G be an interval graph and let $L_1, L_2, \cdots, L_t$, $t \leq n$, be a consecutive clique arrangement of G. Then the set of all minimal cutsets of G consists of vertex set $\mathcal{C}_s = L_s \cap L_{s+1}$, $s \in \{1, 2, \cdots, t-1\}$.*

From Lemma 5, we know that an interval graph G of order n possess at most n minimal cutsets.

Definition 3 [13]. Let G be an interval graph with consecutive clique arrangement $L_1, L_2, \cdots, L_t$. We define $L_0 = L_{t+1} = \emptyset$. For all l, r with $1 \leq l \leq r \leq t$ we define $\mathfrak{P}(l,r) = (\cup_{i=l}^{r} L_i) - (L_{l-1} \cup L_{r+1})$. A set $\mathfrak{P}(l,r)$, $1 \leq l \leq r \leq t$, is said to be a *piece* of G if $\mathfrak{P}(l,r) \neq \emptyset$ and $G[\mathfrak{P}(l,r)]$ is connected. Furthermore, $V = \mathfrak{P}(1,t)$ is a piece of G (even if G is disconnected).

It is obvious that cliques in $G[\mathfrak{P}(l,r)]$ are listed in the same order as that they are listed in graph G.

Lemma 6 [13]. *Let X be a minimal cutset of connected subgraph $G[\mathfrak{P}(l,r)]$, $1 \leq l \leq r \leq t$. Then there exists a minimal cutset $\mathcal{C}_s$ of G, $1 \leq s \leq r$, such that $X = \mathcal{C}_s \cap \mathfrak{P}(l,r) = \mathcal{C}_s - (L_{l-1} \cup L_{r+1})$. Moreover, every connected component of $G[\mathfrak{P}(l,r) - X]$ is a piece of G.*

From the definition of piece of G, any interval graph contains two kind of pieces. A piece is named *complete* if it induces a complete graph. Otherwise, we call it *incomplete*. For every complete piece $G[\mathfrak{P}(l,r)]$, $l \leq r$, holds

$$i_k(G[\mathfrak{P}(l,r)]) = \begin{cases} 0, if\ k \in \{0,1,2,\ldots,|G[\mathfrak{P}(l,r)]|-2\} \\ 1, if\ k = |G[\mathfrak{P}(l,r)]|-1 \end{cases} \quad (1)$$

The incomplete piece $G[\mathfrak{P}(l,r)]$, $1 \le l \le r \le t$, has minimal cutsets, and for every $k \in \{\kappa(G[\mathfrak{P}(l,r)]), \ldots, |G[\mathfrak{P}(l,r)]| - 2\}$, the following equality holds

$$i_k(G[\mathfrak{P}(l,r)]) = \max \sum_{j=h+1}^{p} i_{r_j}(G[C_i]) + h, \tag{2}$$

where the maximum is taken over all $\mathcal{C}_s \cap \mathfrak{P}(l,r)$, $s \in \{l+1, l+2, \cdots, r-1\}$, that are minimal cutsets of $G[\mathfrak{P}(l,r)]$, satisfying the condition that $|\mathcal{C}_s \cap \mathfrak{P}(l,r)| \le k$ and over all nonnegative integer vectors $(r_{l+1}, r_{l+2}, \ldots, r_p)$ fulfilling the condition that $\sum_{j=h+1}^{p} r_j = k - |\mathcal{C}_s \cap \mathfrak{P}(l,r)|$. $C_1, C_2, \cdots, C_h$ are the components of $G[\mathfrak{P}(l,r) - \mathcal{C}_s]$ which are isolated vertices, and $C_{h+1}, C_{h+2}, \cdots, C_p$ are the connected components of $G[\mathfrak{P}(l,r) - \mathcal{C}_s]$ which are not isolated vertices.

Let G be a complete interval graph. Then $i\tau(G) = \infty$. Otherwise, based on Theorem 3, the isolated toughness of incomplete interval graphs can be computed by Algorithm 1.

In the following theorem, we prove the correctness of Algorithm 1 and make clear that the algorithm can be executed in polynomial time.

Theorem 5. *Algorithm 1 outputs the isolated toughness for an input interval graph G of order n within time complexity $O(n^6)$.*

Proof. The correctness of the algorithm can be deduced from the Theorem 3 and Lemma 6. It is easy to see that the steps at lines 2–3 can be performed in $O(1)$ time. The steps at lines 5–11 and 19 can be executed in time $O(n^4)$ in a straightforward manner.

In the steps at line 12 and lines 16–18, an $O(n+m)$ algorithm can be used to test connectedness and calculation components for up to n^2 graphs $G[\mathfrak{P}(l,r)]$. If $G[\mathfrak{P}(l,r)]$ is disconnected and Q_j is a component, then $Q_j = \mathfrak{P}(l_j, r_j)$ with $l(j) = \min\{l(v) : v \in Q_j\}$ and $r(j) = \max\{l(v) : v \in Q_j\}$ which can be computed in time $O(n)$. Hence, the steps at line 12 and lines 16–18 can be executed in time $O(n^4)$.

The steps at lines 13–15 can be executed for at most n^3 triples (s,l,r) with $l \le s \le r$. If $\mathfrak{P}(l,r) - \mathcal{C}(s) \neq \emptyset$, then the components of $G[\mathfrak{P}(l,r) - \mathcal{C}(s)]$ are computed as indicated in Lemma 6, by using the marks of (l,s) and $(s+1,r)$, namely, if the mark is 'complete' or 'incomplete', then (l,s) and $(s+1,r)$, respectively, are stored, and if the mark is 'disconnected', then the corresponding linked list is added. Thus the linked list of (s,l,r) can be computed in time $O(n)$. From Lemma 1 we know that $\mathfrak{P}(l,r) \cap \mathcal{C}(s)$ is a minimal cutset of $G[\mathfrak{P}(l,r) - \mathcal{C}(s)]$ if and only if there are at least two components in the list of (s,l,r) such as every vertex of $\mathfrak{P}(l,r) \cap \mathcal{C}(s)$ has a neighbour in them. From the consecutive clique arrangement, it suffices to check the two components Q_j of $G[\mathfrak{P}(l,s)]$ with the two largest values of r_j and the two components Q_j of $G[\mathfrak{P}(s+1,r)]$ with the two smallest values of l_j, this can be done in time $O(n)$. Hence, the steps at lines 13–15 can be executed in time $O(n^4)$.

The steps on lines 20–22 require that the right side of the Eq. (2) be calculated for each $k \in \{\kappa(G[\mathfrak{P}(l,l+d)]), \ldots, |\mathfrak{P}(l,l+d)| - 2\}$. The list $H_j =$

Algorithm 1: Algorithm Isolated Toughness

Input: An interval graph G with consecutive clique arrangement $L_1, L_2, \cdots, L_t$.
Output: Isolated toughness $i\tau(G)$.

begin
  $L_0 \leftarrow \emptyset$;
  $L_{t+1} \leftarrow \emptyset$;
  for $w \leftarrow 0$ *to* $t+1$ **do**
    compute $l(v) = \min\{w : v \in L_w\}$ and $r(v) = \max\{k : v \in L_w\}$ for every $v \in V$, and then compute all minimal cutsets $\mathcal{C}_s = L_s \cap L_{s-1}$, $s \in \{1, 2, \cdots, t-1\}$. For all l, r $(1 \le l \le r \le t)$ compute the vertex set $\mathfrak{P}(l, r)$;
    if $\mathfrak{P}(l, r) = \emptyset$ **then**
      mark (l, r) ‘empty’;
    end
    if $\mathfrak{P}(l, r) \neq \emptyset$ *and* $G[\mathfrak{P}(l, r)]$ *is a complete induced subgraph* **then**
      mark (l, r) ‘complete’.
    end
    For all nonmarked tuples (l, r), check whether $G[\mathfrak{P}(l, r)]$ is connected;
    if $G[\mathfrak{P}(l, r)]$ *is connected* **then**
      mark (l, r) ‘incomplete’, and for every $s \in \{l, l+1, \ldots, r-1\}$, compute the components $Q_j = \mathfrak{P}(l_j, r_j)$ of $G[\mathfrak{P}(l, r) - \mathcal{C}_s]$, $1 \le j_t \le k$. Check whether $\mathcal{C}_s \cap \mathfrak{P}(l, r)$ is a minimal cutset of $G[\mathfrak{P}(l, r)]$, and if so mark (s, i, j) ‘minimal’, store $(l_1, r_1), (l_2, r_2), \ldots, (l_k, r_k)$ in a linked list with a pointer from (s, l, r) to the head of this list, and compute $\kappa(G[\mathfrak{P}(l, r)]) = \min\{|\mathcal{C}_s \cap \mathfrak{P}(l, r)|\}$ for (s, l, r) marked ‘minimal’.
    end
    if $G[\mathfrak{P}(l, r)]$ *is disconnected* **then**
      compute the components $Q_j = \mathfrak{P}(l_j, r_j)$, $1 \le j \le q$, of $G[\mathfrak{P}(l, r)]$ and store $(l_1, r_1), (l_2, r_2), \ldots, (l_q, r_q)$ in a linked list with a pointer from (l, r) to the head of this list.
    end
    For every pair (l, r) marked ‘complete’ compute $i_k(G[\mathfrak{P}(l, r)])$, $k \in \{0, 1, \ldots, |\mathfrak{P}(l, r)|\}$, according to equation (1);
    for $d \leftarrow 1$ *to* t *and for* $l \leftarrow 1$ *to* $t-d$ **do**
      if $(l, l+d)$ is marked ‘incomplete’, compute $i_k(G[\mathfrak{P}(l, r)])$ for every $k \in \{\kappa(G[\mathfrak{P}(l, l+d)]), \ldots, |\mathfrak{P}(l, l+d)| - 2\}$ according to equation (2). Set $i_k(G[\mathfrak{P}(l, r)]) = 0$ for $k = |\mathfrak{P}(l, l+d)]|$ or $k < \kappa(G[\mathfrak{P}(l, l+d)])$, and let $i_k(G[\mathfrak{P}(l, r)]) = 1$ for $k = |\mathfrak{P}(l, l+d)]| - 1$.
    end
  end
end

$(i_0(Q_j), i_1(Q_j), \ldots, i_{|Q_j|}(Q_j)$, $j \in \{1, 2, \ldots, t\}$, for each component $Q_j = (l_j, r_j)$ of $G[\mathfrak{P}(l, l+d) - \mathcal{C}(s)]$ can be determined in constant time $O(n^2)$ by table look-up, since these lists of smaller pieces are already known. Thus

$$\max_{0 \leq r_{h+1}, r_{h+2}, \ldots, r_p \leq n} \left\{ \sum_{j=h+1}^{p} i_{r_j}(G[Q_j]) + h : \sum_{j=h+1}^{p} r_j = k - |\mathfrak{P}(l, l+d) \cap \mathcal{C}(s)| \right\}$$

can be evaluated in time $O(n^3)$ for a given minimal cutset $\mathfrak{P}(l, l+d) \cap \mathcal{C}(s)$ and for every k with $|\mathfrak{P}(l, l+d) \cap \mathcal{C}(s)| \leq k$ in time $O(n^3)$.

Consequently, from the above analysis we know that the running time of isolated toughness algorithm is $O(n^6)$.
This completes the proof. ∎

Acknowledgements. This work was supported by NSFC (No.11871280), Natural Science Foundation of Zhejiang Province(China) (No. LY17A010017) and Qing Lan Project. Especially, the authors are very thankful to anonymous referees for their constructive suggestions and critical comments, which led to this improved version.

References

1. Bondy, J., Murty, U.: Graph Theory with Applications. Macmillan, London and Elsevier, New york (1976)
2. Booth, K., Lueker, G.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. J. Comput. System Sci. **13**(3), 335–379 (1976)
3. Broersma, H., Fiala, J., Golovach, P., Kaiser, T., Paulusma, D., Proskurowski, A.: Linear-time algorithms for scattering number and hamilton-connectivity of interval graphs. J. Graph Theor. **79**(4), 282–299 (2015)
4. Carlisle, M.C., Lloyd, E.L.: On the k-coloring of intervals. In: Dehne, F., Fiala, F., Koczkodaj, W.W. (eds.) ICCI 1991. LNCS, vol. 497, pp. 90–101. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-54029-6_157
5. Chvátal, V.: Tough graphs and hamiltonian circuits. Discrete Math. **5**, 215–228 (1973)
6. Enomoto, H., Jackson, B., Katerinis, P., Saito, A.: Toughness and the existence of k-factors. J. Graph Theor. **9**, 87–95 (1985)
7. Fabri, J.: Automatic Storage Optimization. UMI Press Ann Arbor, MI (1982)
8. Gilmore, P., Hoffman, A.: A characterization of comparability graphs and of interval graphs. Can. J. Math. **16**(99), 539–548 (1964)
9. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press (1980)
10. Jungck, J., Dick, O., Dick, A.: Computer assisted sequencing, interval graphs and molecular evolution. Biosystem **15**, 259–273 (1982)
11. Kloks, T., Kratschz, D.: Listing all minimal separators of a graph. SIAM J. Comput. **27**(3), 605–613 (1998)
12. Kloks, A.J.J., Kratsch, D., Spinrad, J.P.: Treewidth and pathwidth of co comparability graphs of bounded dimension. Computing Science Note. Eindhoven University of Technology, Eindhoven, The Netherlands **9346** (1993)

13. Kratsch, D., Klocks, T., Müller, H.: Computing the toughness and the scattering number for interval and other graphs. IRISA resarch report. France (1994)
14. Li, F., LI, X.: Neighbor-scattering number can be computed in polynomial time for interval graphs. Comput. Math. Appl. **54**(5), 679–686 (2007)
15. Ma, Y., Liu, G.: Isolated toughness and the existence of fractional factors in graphs. Acta Appl. Math. Sinica (in Chinese) **62**, 133–140 (2003)
16. Ma, Y., Liu, G.: Fractional factors and isolated toughness of graphs. Mathematica Applicata **19**(1), 188–194 (2006)
17. Ma, Y., Wang, A., Li, J.: Isolated toughness and fractional (g, f)-factors of graphs. Ars Comb. **93**, 153–160 (2009)
18. Ma, Y., Yu, Q.: Isolated toughness and existence of f-factors. In: Akiyama, J., Chen, W.Y.C., Kano, M., Li, X., Yu, Q. (eds.) CJCDGCGT 2005. LNCS, vol. 4381, pp. 120–129. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70666-3_13
19. Ma, Y., Yu, Q.: Isolated toughness and existence of $[a, b]$-factors in graphs. J. Combin. Math. Combin. Comput. **62**, 1–12 (2007)
20. Ohtsuki, T., Mori, H., Khu, E., Kashiwabara, T., Fujisawa, T.: One dimensional logic gate assignment and interval graph. IEEE Trans. Circ. Syst. **26**, 675–684 (1979)
21. Scheinerman, E.R., Ullman, D.H.: Fractional Graph Theory. John Wiley and Son Inc., New York (1997)
22. Yang, J., Ma, Y., Liu, G.: fractional (g, f)-factor in graphs. Acta Mathematica Scientia **16**(4), 385–390 (2001)