الگوریتم‌های کارگاه احاطه‌گری

داود بخشش
دانشگاه بجنورد

نسخه ۱۱.۲۰۲۶

نمادگذاری

در سراسر این صفحه $G = (V, E)$ یک گراف ساده و بدون جهت با $n = |V|$ رأس و $m = |E|$ یال است. $N(v)$ مجموعه‌ی همسایه‌های v و $N[v] = N(v) \cup \{v\}$ است و برای مجموعه‌ی X می‌نویسیم $N[X] = \bigcup_{v \in X} N[v]$. Δ و δ بیشینه و کمینه‌ی درجه‌اند و $H(k) = \sum_{i=1}^{k} 1/i \le \ln k + 1$ عدد هارمونیک است. پارامترهایی که صفحه محاسبه می‌کند عبارت‌اند از γ (احاطه‌گری)، γ_t (کل)، γ_c (همبند)، i (احاطه‌گری مستقل)، γ_{pr} (جفتی)، $\gamma_{\times 2}$ (۲-احاطه‌گری)، γ_p (کامل)، α (استقلال)، β (پوشش رأسی)، γ_R (رومی) و d (عدد دامنه‌ای)؛ تعریف‌ها در [۱۱] آمده است. این مسئله‌ها در گراف‌های کلی سخت‌اند-NP (برای مسئله‌های کلاسیک نگاه کنید به [۸])؛ برای همین صفحه روی گراف‌های بزرگ از روش‌های ابتکاری و روی گراف‌های کوچک از جست‌وجوی کامل استفاده می‌کند.

۱ احاطه‌گر حریصانه

هر رأس v مجموعه‌ی $N[v]$ را پوشش می‌دهد؛ پس پیدا کردن مجموعه‌ی احاطه‌گر همان مسئله‌ی پوشش مجموعه‌ای روی مجموعه‌ی V است. قاعده‌ی حریصانه هر بار رأسی را برمی‌دارد که بیشترین تعداد رأسِ هنوز پوشش‌نیافته را پوشش بدهد.

تحلیل

- درستی. تا وقتی رأسی پوشش‌نیافته باشد، خودِ همان رأس دست‌کم ۱ سود دارد؛ پس هر دور رأس تازه‌ای پوشش می‌دهد و حلقه با $C = V$ تمام می‌شود، یعنی S احاطه‌گر است.
- زمان. حداکثر n دور و در هر دور بررسی همه‌ی رأس‌ها و همسایه‌هایشان: $O(n(n+m))$.
- تضمین. این همان الگوریتم حریصانه‌ی پوشش مجموعه‌ای با مجموعه‌هایی $N[v]$ به اندازه‌ی حداکثر $\Delta + 1$ است؛ پس $|S| \le H(\Delta+1)\,\gamma(G) \le (\ln(\Delta+1)+1)\gamma(G)$ [۱۴، ۱۵، ۱].

۲ احاطه‌گر کل حریصانه

در احاطه‌گر کل هر رأس، حتی رأس‌های خود S، باید همسایه‌ای در S داشته باشد؛ پس هر رأس فقط همسایگی باز $N(v)$ را پوشش می‌دهد و ایده‌ی حریصانه باز هم پوشش مجموعه‌ای است. اگر رأسی همسایه نداشته باشد، احاطه‌گر کل وجود ندارد [۱۳].

تحلیل

- درستی. رأس u دقیقاً وقتی با v پوشش می‌یابد که $u \in N(v)$ باشد؛ پس $C = V$ یعنی هر رأس همسایه‌ای در S دارد.
- زمان. $O(n(n+m))$.
- تضمین. پوشش مجموعه‌ای با مجموعه‌هایی به اندازه‌ی حداکثر Δ؛ پس $|S| \le H(\Delta)\,\gamma_t(G) \le (\ln\Delta + 1)\gamma_t(G)$ [۱].

الگوریتم ۱ Greedy-Dominating-Set

ورودی: گراف $G = (V, E)$
خروجی: یک مجموعه‌ی احاطه‌گر S

$S \leftarrow \emptyset$; $C \leftarrow \emptyset$ ▷ C = covered vertices
while $C \neq V$ **do**
  pick v maximizing $|N[v] \setminus C|$
  $S \leftarrow S \cup \{v\}$; $C \leftarrow C \cup N[v]$
end while
return S

الگوریتم ۲ Greedy-Total-Dominating-Set

ورودی: گراف G
خروجی: یک احاطه‌گر کل، یا پیامی که چنین مجموعه‌ای نیست

$S \leftarrow \emptyset$; $C \leftarrow \emptyset$
while $C \neq V$ **do**
  pick v maximizing $|N(v) \setminus C|$
  if $|N(v) \setminus C| = 0$ **then** ▷ some vertex has no neighbor
    return “no total dominating set”
  end if
  $S \leftarrow S \cup \{v\}$; $C \leftarrow C \cup N(v)$
end while
return S

۳ احاطه‌گر همبند (رشد حریصانه)

از رأسی با بیشترین درجه شروع می‌کنیم. در هر گام رأسی را اضافه می‌کنیم که از قبل احاطه شده باشد (یعنی به S وصل است) و بیشترین رأس تازه را احاطه کند. چون هر رأس تازه به S وصل است، زیرگراف القایی $G[S]$ همبند می‌ماند. اگر پیش از احاطه‌شدن همه‌ی رأس‌ها چنین رأسی پیدا نشود، G ناهمبند است. گوها و خولر [۹] این نوع رشد حریصانه را بررسی کرده‌اند: آنها نشان می‌دهند قاعده‌ی ساده می‌تواند بد عمل کند، و گونه‌ای که جفت رأس‌ها را هم در نظر می‌گیرد (یک رأس احاطه‌شده همراه با یک همسایه‌ی سفید) ضریب $2(1 + H(\Delta))$ می‌دهد. صفحه همان قاعده‌ی ساده را اجرا می‌کند، پس برایش ضریب تقریبی ادعا نمی‌شود.

تحلیل

- درستی. رأس $u \in C \setminus S$ به رأسی از S وصل است؛ پس اگر $G[S]$ همبند باشد، $G[S \cup \{u\}]$ هم همبند است. مجموعه‌ی نهایی احاطه‌گر است چون $C = V$.
- زمان. $O(n(n + m))$.

۴ احاطه‌گر جفتی (جفت‌های حریصانه)

احاطه‌گر جفتی یک مجموعه‌ی احاطه‌گر است که زیرگراف القایی‌اش تطابق کامل دارد [۱۲]. رأس u را که هنوز احاطه نشده برمی‌داریم، آن را با همسایه‌ی w جفت می‌کنیم که بیشترین رأس تازه را پوشش می‌دهد و هر دو را اضافه می‌کنیم. هر جفت یک یال است؛ پس جفت‌های انتخاب‌شده به‌طور ساختاری یک تطابق کامل برای $G[S]$ می‌سازند. رأس بدون همسایه مسئله را بی‌جواب می‌کند.

تحلیل

- درستی. در پایان $C = V$ است؛ پس S احاطه‌گر است و $G[S]$ شامل یال‌های مجزای uw است که همه‌ی S را می‌پوشانند.

الگوریتم ۳ Greedy-CDS

ورودی: گراف همبند G
خروجی: یک مجموعه‌ی احاطه‌گر همبند S

$v \leftarrow$ vertex of maximum degree; $S \leftarrow \{v\}$; $C \leftarrow N[v]$
while $C \neq V$ **do**
  pick $u \in C \setminus S$ maximizing $|N(u) \setminus C|$
  if $|N(u) \setminus C| = 0$ **then** ▷ G is disconnected
    return “disconnected”
  end if
  $S \leftarrow S \cup \{u\}$; $C \leftarrow C \cup N[u]$
end while
return S

الگوریتم ۴ Greedy-Paired-Dominating-Set

ورودی: گراف G
خروجی: یک احاطه‌گر جفتی، یا پیامی که چنین مجموعه‌ای نیست

$S \leftarrow \emptyset$; $C \leftarrow \emptyset$
while $C \neq V$ **do**
  $u \leftarrow$ any vertex of $V \setminus C$
  if $N(u) = \emptyset$ **then**
    return “no paired dominating set”
  end if
  $w \leftarrow$ neighbor of u maximizing $|N[w] \setminus C|$
  $S \leftarrow S \cup \{u, w\}$; $C \leftarrow C \cup N[u] \cup N[w]$
end while
return S ▷ the edges uw form a perfect matching of $G[S]$

- زمان. $O(n(n+m))$. جواب همیشه معتبر است؛ برای این روش ابتکاری ضریب تقریبی ادعا نمی‌شود. برای گراف‌های بدون رأس منزوی $\gamma \le \gamma_{pr} \le 2\gamma$ [۱۲].

۵ ۲-احاطه‌گر (حریصانه)

در یک ۲-احاطه‌گر هر رأس بیرون S دست‌کم دو همسایه در S دارد [۶]. برای هر رأس «کمبود» را می‌شماریم و در هر گام رأسی را اضافه می‌کنیم که بیشترین کمبود را برطرف کند: کمبود خودش صفر می‌شود و هر همسایه‌ی نیازمندِ بیرون S یک واحد کمتر کمبود دارد.

تحلیل

- درستی. حلقه فقط وقتی تمام می‌شود که کمبود هر رأس صفر شده باشد، و این دقیقاً همان تعریف است. حلقه تمام می‌شود چون افزودن هر رأسِ با سود مثبت $\sum \mathrm{def}$ را دست‌کم ۱ کم می‌کند.

- زمان. چندجمله‌ای: حداکثر n دور و در هر دور بررسی همه‌ی رأس‌ها و همسایه‌هایشان. ضریب تقریبی ادعا نمی‌شود.

۶ احاطه رومی از روی احاطه‌گر حریصانه

تابع رومی یک برچسب‌گذاری $f : V \to \{0,1,2\}$ است که هر رأس با $f(v)=0$ همسایه‌ای مانند u با $f(u)=2$ دارد؛ وزن آن $\sum_v f(v)$ و کمینه‌ی وزن γ_R است [۳، ۱۶]. اگر مجموعه‌ی احاطه‌گر S را ۲ و بقیه را ۰ برچسب بزنیم، وزن $2|S|$ می‌شود. مرحله‌ی بهبود، هر ۲ را در صورت امکان به ۱ کاهش می‌دهد.

الگوریتم ۵ Greedy-۲-Dominating-Set

ورودی: گراف G
خروجی: یک ۲-احاطه‌گر S

$S \leftarrow \emptyset$ ▷ $\mathrm{def}(v) = 0$ if $v \in S$, else $\max\{0,\ 2 - |N(v) \cap S|\}$
while $\sum_v \mathrm{def}(v) > 0$ **do**
  pick $x \notin S$ maximizing $\mathrm{def}(x) + |\{u \in N(x) \setminus S : \mathrm{def}(u) > 0\}|$
  $S \leftarrow S \cup \{x\}$
end while
return S

الگوریتم ۶ Roman-From-Greedy-DS

ورودی: گراف G
خروجی: یک تابع رومی f

$S \leftarrow$ Greedy-Dominating-Set(G)
$f(v) \leftarrow 2$ for $v \in S$; $f(v) \leftarrow 0$ otherwise
for $v \in S$ **do** ▷ improvement step
  if every $u \in N(v)$ with $f(u) = 0$ has some $w \in N(u) \setminus \{v\}$ with $f(w) = 2$ **then**
    $f(v) \leftarrow 1$
  end if
end for
return f

تحلیل

- درستی. در آغاز f معتبر است چون S احاطه‌گر است. کاهش $f(v)$ از ۲ به ۱ بی‌خطر است: خودِ v به همسایه‌ی ۲ نیاز ندارد و هر همسایه‌ی ۰ هنوز همسایه‌ی دیگری با برچسب ۲ دارد.
- زمان. $O(n(n+m))$. وزن حداکثر $2|S|$ است و برای همین به‌طور کلی $\gamma \le \gamma_R \le 2\gamma$ [۳]. برای این روش، وزن $\gamma_R \le 2H(\Delta+1)$ است، چون $|S| \le H(\Delta+1)\gamma \le H(\Delta+1)\gamma_R$.

۷ پوشش رأسی: تقریب ۲ با تطابق ماکسیمال

یال‌ها را یکی‌یکی می‌بینیم و هر وقت یالی هنوز هیچ سری در پوشش نداشت، هر دو سرش را اضافه می‌کنیم. یال‌هایی که این کار را کرده‌اند دوبه‌دو از هم جدا هستند و یک تطابق ماکسیمال M می‌سازند [۵].

تحلیل

- درستی. هر یال یا هنگام بررسی از قبل پوشیده است یا هر دو سرش اضافه می‌شود.
- تضمین. یال‌های انتخاب‌شده مجزا هستند و هر پوشش رأسی باید از هر کدام یک سر داشته باشد؛ پس $|C| = 2|M|$ ولی $|M| \le \beta(G)$ و در نتیجه $|C| \le 2\beta(G)$. زمان $O(m)$. همچنین $\alpha + \beta = n$ [۷].

۸ مجموعه مستقل ماکسیمال حریصانه (کمترین درجه اول)

هر بار رأسی با کمترین درجه در گراف باقی‌مانده را برمی‌داریم، به S اضافه می‌کنیم و آن را با همسایه‌هایش حذف می‌کنیم. برداشتن رأس‌های کم‌درجه در هر گام رأس‌های کمی را حذف می‌کند و معمولاً جا برای مجموعه‌ی بزرگ‌تری باقی می‌گذارد.

الگوریتم ۷ Matching-Vertex-Cover

ورودی: گراف $G = (V, E)$
خروجی: یک پوشش رأسی C

```
C ← ∅
for each edge uv ∈ E (in input order) do
    if u ∉ C and v ∉ C then
        C ← C ∪ {u, v}
    end if
end for
return C
```

الگوریتم ۸ Greedy-Independent-Set

ورودی: گراف G
خروجی: یک مجموعه‌ی مستقل ماکسیمال S

```
S ← ∅; H ← G
while V(H) ≠ ∅ do
    v ← vertex of minimum degree in H
    S ← S ∪ {v}; H ← H − N_H[v]
end while
return S
```

تحلیل

- درستی. رأس‌ها فقط همراه با همسایه‌ای در S یا هنگام انتخاب شدن حذف می‌شوند؛ پس S مستقل است و رأسی به آن اضافه نمی‌شود: S ماکسیمال است.
- تضمین. هر مجموعه‌ی مستقل ماکسیمال دست‌کم $n/(\Delta+1)$ رأس دارد. برای همین قاعده‌ی کمترین درجه، هالدورسون و رادهاکریشنان ضریب بهتر $(\Delta+2)/3$ را روی گراف‌هایی با بیشینه درجه‌ی Δ ثابت کرده‌اند: $\alpha(G) \le \frac{\Delta+2}{3}|S|$ [۱۰]. زمان $O(n(n+m))$.

۹ احاطه‌گر مستقل حریصانه (بیشترین درجه اول)

یک احاطه‌گر مستقل دقیقاً همان مجموعه‌ی مستقل ماکسیمال است [۴]: مجموعه‌ی مستقل ماکسیمال احاطه‌گر است (وگرنه می‌شد رأس دیگری به آن افزود) و هر مجموعه‌ی مستقل احاطه‌گر ماکسیمال است. برای اینکه مجموعه کوچک بماند، این گونه رأسی با بیشترین درجه را اول برمی‌دارد تا هر گام تا می‌شود رأس بیشتری حذف شود.

تحلیل

- درستی. مانند الگوریتم قبلی، S مجموعه‌ی مستقل ماکسیمال است و پس هم مستقل است و هم احاطه‌گر.
- زمان $O(n(n+m))$. اندازه‌ی S کران بالایی برای $i(G)$ است و $\gamma \le i \le \alpha$ [۴]. ضریب تقریبی ادعا نمی‌شود.

۱۰ جواب دقیق (ماسک بیتی، تا ۲۰ رأس)

چون این مسئله‌ها سخت‌اند-NP [۸]، جواب دقیق زمان نمایی می‌خواهد. برای $n \le 20$ صفحه همه‌ی زیرمجموعه‌های $X \subseteq V$ را، به شکل ماسک بیتی، بررسی می‌کند. دو جدول یک‌بار با برنامه‌ریزی پویا روی ماسک‌ها پر می‌شود: همسایگی بسته‌ی هر زیرمجموعه، و اینکه $N[X]$ مستقل است یا نه. بعد هر مسئله فقط یک آزمون سریع روی هر زیرمجموعه است و زیرمجموعه‌هایی که از بهترین جواب فعلی بهتر نمی‌شوند کنار گذاشته می‌شوند.

الگوریتم ۹ Greedy-Independent-Dominating-Set

ورودی: گراف G
خروجی: یک احاطه‌گر مستقل S

$S \leftarrow \emptyset$; $H \leftarrow G$
while $V(H) \neq \emptyset$ **do**
    $v \leftarrow$ vertex of maximum degree in H
    $S \leftarrow S \cup \{v\}$; $H \leftarrow H - N_H[v]$
end while
return S

الگوریتم ۱۰ Exact-Search

ورودی: گراف G با $n \leq 20$ و یک مسئله‌ی P
خروجی: یک مجموعه‌ی بهینه برای P، اگر وجود داشته باشد

for $X = 1, \ldots, 2^n - 1$, with x the lowest vertex of X **do**
    $\mathrm{cl}[X] \leftarrow \mathrm{cl}[X \setminus \{x\}] \cup N[x]$
    $\mathrm{ind}[X] \leftarrow \mathrm{ind}[X \setminus \{x\}] \wedge (N(x) \cap X = \emptyset)$
end for
$best \leftarrow$ none
for $X \subseteq V$ **do**
    if $|X|$ cannot improve $|best|$ **then**
        continue
    end if
    if $\mathrm{Test}_P(X)$ **then**
        $best \leftarrow X$
    end if
end for
return $best$

تحلیل

- آزمون‌ها. احاطه‌گر: $\mathrm{cl}[X] = V$. کل: $\bigcup_{v \in X} N(v) = V$. همبند: احاطه‌گر و $G[X]$ همبند. احاطه‌گر مستقل: $\mathrm{ind}[X]$ و $\mathrm{cl}[X] = V$. جفتی: احاطه‌گر و $G[X]$ دارای تطابق کامل. ۲-احاطه‌گر / کامل: هر $v \notin X$ دست‌کم ۲ / دقیقاً ۱ همسایه در X دارد. مجموعه‌ی مستقل: $\mathrm{ind}[X]$. پوشش رأسی: $\mathrm{ind}[V \setminus X]$.
- احاطه رومی. فرض کنید X مجموعه‌ی رأس‌های با برچسب ۲ باشد. هر رأس بیرون $N[X]$ باید برچسب ۱ بگیرد و بقیه ۰؛ پس بهترین وزن برای این X برابر $2|X| + |V \setminus N[X]|$ است و آن را روی همه‌ی X ها کمینه می‌کنیم.
- زمان در بدترین حالت $O(2^n \cdot n)$ و حافظه $O(2^n)$.

۱۱ افراز دامنه‌ای

عدد دامنه‌ای $d(G)$ بیشترین تعداد مجموعه‌ی احاطه‌گر دوبه‌دو مجزاست و $d(G) \leq \delta + 1$ برقرار است [۴]. هر مجموعه‌ی احاطه‌گر یک زیرمجموعه‌ی احاطه‌گر مینیمال دارد؛ پس کافی است همه‌ی احاطه‌گرهای مینیمال را پیدا کنیم و بزرگ‌ترین خانواده‌ی دوبه‌دو مجزا از آن‌ها را با جست‌وجوی بازگشتی بیابیم.

تحلیل

- درستی. اگر هر مجموعه‌ی یک خانواده‌ی بهینه را با یک احاطه‌گر مینیمالِ درون آن جایگزین کنیم، مجموعه‌ها مجزا می‌مانند؛ پس جست‌وجو فقط در $\mathcal{M}$ چیزی از دست نمی‌دهد. افزودن رأس‌های باقی‌مانده به یک مجموعه، احاطه‌گر بودنش را حفظ می‌کند.

الگوریتم ۱۱ Domatic-Partition

ورودی: گراف G با $n \le 20$
خروجی: مجموعه‌های احاطه‌گر مجزای $D_1, \dots, D_k$

$\mathcal{M} \leftarrow$ all minimal dominating sets of G; $best \leftarrow \emptyset$
Search$(s,\ U,\ \mathcal{F})$ ▷ backtracking over $\mathcal{M}$
if $|\mathcal{F}| > |best|$ **then**
    $best \leftarrow \mathcal{F}$
end if
if $|best| = \delta + 1$ or the step limit is reached **then**
    return stop
end if
for $D \in \mathcal{M}$ from index s with $D \cap U = \emptyset$ **do**
    Search(next, $U \cup D,\ \mathcal{F} + [D]$)
end for
add vertices that are in no set to D_1
return $best$

الگوریتم ۱۲ Check-Selection

ورودی: گراف G، یک انتخاب S (یا برچسب‌های f) و یک مسئله‌ی P
خروجی: رأس‌ها و یال‌هایی که تعریف را نقض می‌کنند

for $v \in V$ **do**
    $k(v) \leftarrow |N(v) \cap S|$
end for
for $v \in V$ **do**
    compare $k(v)$ with the definition of P ▷ e.g. $v \in S$ or $k(v) \ge 1$
end for
for $uv \in E$ **do**
    check independence / cover conditions on u, v
end for
test connectivity or a perfect matching of $G[S]$ if P requires it

- زمان. نمایی. جست‌وجو پس از $2 \cdot 10^6$ گام قطع می‌شود و آن‌وقت صفحه فقط کران پایین $d(G) \ge k$ را گزارش می‌کند.

۱۲ بررسی انتخاب شما

هر انتخابی که بکنید، صفحه برای هر رأس می‌شمارد چند همسایه‌اش در S هستند (و در احاطه رومی چند همسایه برچسب ۲ دارند) و با تعریف مقایسه می‌کند. همبندی $G[S]$ با جست‌وجو در گراف و تطابق کامل در احاطه‌گر جفتی با یک جست‌وجوی بازگشتی کوتاه (تا $|S| = 24$) بررسی می‌شود.

تحلیل

- زمان $O(n + m)$ برای بیشتر مسئله‌ها.

۱۳ گراف دیسک واحد

در حالت دیسک واحد، هر بار رأسی جابه‌جا شود یال‌ها از نو ساخته می‌شوند: دو رأس مجاورند اگر فاصله‌ی اقلیدسی‌شان از شعاع r بیشتر نباشد [۲].

الگوریتم ۱۳ Unit-Disk-Graph

ورودی: نقطه‌های $p_1, \ldots, p_n$ در صفحه و شعاع r
خروجی: گراف $G = (V, E)$

$E \leftarrow \emptyset$
for all pairs $i < j$ **do**
    if $\|p_i - p_j\| \leq r$ **then**
        add the edge ij to E
    end if
end for
return (V, E)

الگوریتم ۱۴ Read-Ipe

ورودی: یک فایل XML از Ipe
خروجی: یک گراف G

$V \leftarrow$ all marks and small circles (after group transformations)
for each polyline or polygon and each consecutive pair of points p, q **do**
    $a \leftarrow$ vertex nearest to p within tolerance, else a new vertex at p
    $b \leftarrow$ the same for q
    if $a \neq b$ **then**
        add the edge ab (once)
    end if
end for
scale and center everything to fit the canvas

تحلیل

- زمان $O(n^2)$ در هر به‌روزرسانی.

۱۴ خواندن فایل‌های Ipe

Ipe [۱۷] نقاشی را به‌صورت XML ذخیره می‌کند. صفحه نشانه‌ها (mark) و دایره‌های کوچک را رأس می‌کند، و هر پاره‌خطِ یک خط‌شکسته یا چندضلعی را به یالی بین نزدیک‌ترین رأس‌ها به دو سرش تبدیل می‌کند (با تلورانس کوچک). سری که رأسی نزدیکش نباشد رأس تازه می‌شود. تبدیل‌های گروه‌ها اعمال می‌شود و نتیجه به اندازه‌ی بوم درمی‌آید.

تحلیل

- زمان $O(n \cdot k)$ در بدترین حالت، که k تعداد پاره‌خط‌هاست.

مراجع

[۱] V. Chvátal، A greedy heuristic for the set-covering problem، Mathematics of Operations Research ۴(۳)، ۲۳۳–۲۳۵ (۱۹۷۹).

[۲] D. S. Johnson، B. N. Clark، C. J. Colbourn، Unit disk graphs، Discrete Mathematics ۸۶(۱–۳)، ۱۶۵–۱۷۷ (۱۹۹۰).

[۳] S. T. Hedetniemi، S. M. Hedetniemi، P. A. Jr.، Dreyer E. J. Cockayne، Roman domination in graphs، Discrete Mathematics ۲۷۸(۱–۳)، ۱۱–۲۲ (۲۰۰۴).

[۴] S. T. Hedetniemi، E. J. Cockayne، Towards a theory of domination in graphs، Networks ۷(۳)، ۲۴۷–۲۶۱ (۱۹۷۷).

[۵] Cormen، T.H.، Leiserson، C.E.، Rivest، R.L.، Stein، C.: Introduction to Algorithms، ۴th edn. Press MIT (۲۰۲۲).

[۶] Fink، J.F.، Jacobson، M.S.: On n-domination in graphs: Algorithm and Graph Theory، Graph Algorithms and Applications، Computer Science، pp. ۲۸۳–۳۰۰. Wiley New York (۱۹۸۵).

[۷] Gallai، T.: Über extreme Punkt- und Kantenmengen. Annales Universitatis Scientiarum Budapesti- de Rolando Eötvös Nominatae، Sectio Mathematica، ۲، ۱۳۳–۱۳۸ (۱۹۵۹).

[۸] Garey، M.R.، Johnson، D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman، H. W. San Francisco (۱۹۷۹).

[۹] Guha، S.، Khuller، S.: Approximation algorithms for connected dominating sets. Algorithmica ۲۰(۴)، ۳۷۴–۳۸۷ (۱۹۹۸) doi: https://doi.org/10.1007/PL00009201.

[۱۰] Halldórsson، M.M.، Radhakrishnan، J.: Greed is good: Approximating independent sets in sparse and bounded-degree graphs. Algorithmica ۱۸(۱)، ۱۴۵–۱۶۳ (۱۹۹۷) doi: https://doi.org/10.1007/BF02523693.

[۱۱] Haynes، T.W.، Hedetniemi، S.T.، Slater، P.J.: Fundamentals of Domination in Graphs. Dekker، York New (۱۹۹۸).

[۱۲] Haynes، T.W.، Slater، P.J.: Paired-domination in graphs. Networks ۳۲(۳)، ۱۹۹–۲۰۶ (۱۹۹۸).

[۱۳] Henning، M.A.، Yeo، A.: Total Domination in Graphs. Springer Monographs in Mathematics، Springer (۲۰۱۳).

[۱۴] Johnson، D.S.: Approximation algorithms for combinatorial problems. Journal of Computer and System Sciences ۹(۳)، ۲۵۶–۲۷۸ (۱۹۷۴).

[۱۵] Lovász، L.: On the ratio of optimal integral and fractional covers. Discrete Mathematics ۱۳(۴)، ۳۸۳–۳۹۰ (۱۹۷۵).

[۱۶] Stewart، I.: Defend the Roman Empire! Scientific American ۲۸۱(۶)، ۱۳۶–۱۳۹ (۱۹۹۹).

[۱۷] Cheong، O.: Ipe editor. drawing extensible the https://ipe.otfried.org.