
بسم الله الرحمن الرحیم

دانشگاه صنعتی اصفهان
دانشکده علوم ریاضی

کاربرد نظریه بازی در بیشینه سازی نفوذ در شبکه های اجتماعی

پروژه کارشناسی

اسری پرداختی

استاد راهنما
دکتر رامین جوادی

۱۴۰۳

فهرست مطالب

۱ مقدمه

در حال حاضر فعالیت‌های تحقیقاتی زیادی در جهان انجام می‌شود که هدف آن‌ها درک بهتر رفتار انسان برای پیش‌گویی آینده است. اساس این فعالیت‌ها روش‌های ریاضیاتی است که امروزه بیشتر با نام نظریه بازی‌ها شناخته می‌شود [۱، ۱۲، ۱۳].

نظریه بازی‌ها، ابتدا برای درک بازی‌هایی چون پوکر و شطرنج آغاز شد و سپس به‌عنوان ابزارهای ریاضی در خدمت توصیف رفتارهای اقتصادی قرار گرفت. در اصل، نظریه بازی‌ها هر موقعیتی را که شامل تعامل‌های استراتژیک باشد □ از بازی تنیس گرفته تا جنگ قیمت‌ها □ در بر می‌گیرد و ابزار ریاضی برای محاسبه نتایج مورد انتظار از استراتژی‌های انتخاب‌شده فراهم می‌آورد [۱، ۱۲]. بنابراین، ریاضیات نظریه بازی‌ها فرمول‌هایی برای تصمیم‌گیری‌های معقول در عرصه‌های رقابتی عرضه می‌کند.

امروزه شبکه‌های اجتماعی آنلاین به بستر مهمی برای برقراری ارتباط، به‌اشتراک‌گذاری دانش و انتشار اطلاعات تبدیل شده‌اند. ساختار این شبکه‌ها معمولاً با مدل گرافی نمایش داده می‌شود؛ هر فرد یک گره و هر رابطه اجتماعی یک یال است [؟، ۴، ۹]. با توجه به استفاده گسترده از رسانه‌های اجتماعی، ایده‌ها، ترجیحات و رفتار افراد اغلب تحت تأثیر همسالان یا دوستانشان در شبکه قرار می‌گیرد [۵، ۱۴].

در میان تمام گره‌های یک شبکه اجتماعی، یافتن گره‌هایی که می‌توانند همسایگان و در نتیجه دیگر اعضای شبکه را بیش از سایر گره‌ها تحت تأثیر قرار دهند اهمیت ویژه‌ای دارد؛ این گره‌ها را «گره‌های تأثیرگذار» می‌نامیم [۸، ۱۱].

برای نشان دادن اهمیت یافتن این گره‌ها، مثالی می‌زنیم: فرض کنید داده‌هایی از میزان تأثیرگذاری افراد بر یکدیگر در یک شبکه داریم و می‌خواهیم محصول جدیدی را بازاریابی کنیم؛ هدف این است که با هدف‌گذاری بر روی چند نفر (به‌عنوان دانه اولیه) نفوذ را گسترش دهیم تا بخش بزرگی از جمعیت محصول را بپذیرد. انتخاب هوشمندانه مجموعه‌ای از این افراد می‌تواند یک «کاسکاد» عظیم از پذیرش ایجاد کند [۶، ۱۴].

مسئله‌ای که در اینجا پدید می‌آید به‌صورت عمومی با عنوان «**مسئله مجموعه هدف**» یا مسئله □IM[۱] شناخته می‌شود: در یک شبکه اجتماعی G و برای پارامتر k، هدف یافتن مجموعه‌ای از k گره است که تعداد کاربران تأثیرپذیر نهایی را بیشینه کند [۶، ۱۴، ۱۵].

به‌طور کلی دو مدل متداول برای انتشار اطلاعات وجود دارد: مدل آستانه خطی (LT[۲]) و مدل آبشار مستقل (IC[۳]) که هر دو در قالب‌گذاری رسمی و آنالیز مسئله مورد استفاده

[۱] Influence Maximization:
[۲] Linear Threshold
[۳] Independent Cascade

قرار می‌گیرند [۶]. در مدل LT هر گره زمانی فعال می‌شود که وزن مجموع نفوذ همسایگان فعالش از آستانه آن فراتر رود؛ در مدل IC هر گره فعال یک شانس مستقل برای فعال‌سازی هر همسایه غیرفعال دارد و فرآیند به‌صورت گام‌به‌گام و تصادفی ادامه می‌یابد تا دیگر گره غیرفعالی باقی نماند [۶، ۱۱].

با توجه به شکل مسئله و خواص تابع نفوذ، نتایج بنیادی درباره ویژگی‌هایی همچون زیرمدولار بودن تابع نفوذ و تقریب‌پذیری الگوریتم‌های طمعی برای این‌گونه مسائل نیز مطرح شده‌اند [۶، ۱۰].

در ادامه، با استفاده از ابزارهای نظریه بازی‌ها و معرفی معیارها و مفاهیم مناسب، به بررسی بیشتر مدل‌های انتشار و ارائه روش‌هایی برای بیشینه‌سازی نفوذ در شبکه‌های اجتماعی می‌پردازیم [۲۰، ۲۲، ۲۴].

۲ تعریف مفاهیم پایه

۱.۲ ارزش شاپلی

ارزش شاپلی[۴] یک مفهوم بنیادی در نظریه بازی‌های مشارکتی است که به صورت منصفانه‌ای سود یا هزینه ایجادشده توسط یک ائتلاف را بین بازیکنان تقسیم می‌کند. این مفهوم نخستین‌بار توسط لوید شاپلی[۵] معرفی شد و به عنوان راه‌حلی برای تخصیص ارزش کل ائتلاف به بازیکنان مطرح شد به نحوی که ویژگی‌های طبیعی و مطلوبی را رعایت کند [۱، ۳، ۱۲].

در یک بازی مشارکتی با قابلیت انتقال سود، فرض می‌کنیم مجموعه‌ای از بازیکنان N و تابع ارزش $v : 2^N \rightarrow \mathbf{R}$ وجود دارد که به هر زیرمجموعه از بازیکنان عددی نسبت می‌دهد. این عدد نشان‌دهنده سود کل ائتلاف است. هدف تخصیص ارزش $v(N)$ به بازیکنان به گونه‌ای است که قواعد منطقی رعایت شوند.

۱.۱.۲ ویژگی‌های اصلی ارزش شاپلی

ارزش شاپلی دارای چند ویژگی مهم است که منصفانه بودن آن را تضمین می‌کند:

۱. **کارایی**[۶]: مجموع ارزش‌های تخصیص‌یافته به بازیکنان برابر با ارزش کل ائتلاف است:

$$\sum_{i \in N} \phi_i(v) = v(N)$$

۲. **بی‌طرفی نسبت به بازیکنان یکسان**[۷]: اگر دو بازیکن نقشی یکسان در بازی داشته باشند، ارزش شاپلی آن‌ها برابر است.

۳. **بازیکن غیرمؤثر**[۸]: اگر بازیکنی هیچ ارزشی به هیچ ائتلافی اضافه نکند، سهم او صفر است.

۴. **خطی بودن**[۹]: اگر دو بازی v و w را جمع کنیم، ارزش شاپلی جمع برابر با جمع ارزش‌های شاپلی دو بازی است.

۲.۲ نحوه محاسبه

ارزش شاپلی بر این ایده استوار است که نقش هر بازیکن بر اساس مشارکت نهایی او در تشکیل ائتلاف محاسبه شود. برای تعیین سهم هر بازیکن، همه ترتیبات ممکن ورود بازیکنان

---

[۴] Shapley Value
[۵] Lloyd Shapley
[۶] Efficiency
[۷] Symmetry
[۸] Dummy Player
[۹] Additivity

به ائتلاف در نظر گرفته می‌شود و برای هر ترتیب، میزان افزایشی که بازیکن هنگام ورود به جمع بازیکنان قبلی ایجاد می‌کند محاسبه می‌گردد. میانگین این افزایش‌ها در تمام ترتیبات ممکن، ارزش شاپلی بازیکن خواهد بود.

قضیه ۱.۲. ارزش شاپلی برای هر بازیکن i را می‌توان از فرمول زیر محاسبه نمود:
(۱)

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[v(S \cup \{i\}) - v(S) \right]$$

که در آن:

۱. S هر زیرمجموعه‌ای از بازیکنان بدون i است،

۲. $v(S \cup \{i\}) - v(S)$ مشارکت نهایی بازیکن i نسبت به مجموعه S را نشان می‌دهد،

۳. ضریب مقابل این اختلاف، احتمال آن است که S دقیقاً قبل از i در یک ترتیب تصادفی قرار گیرد.

۱.۲.۲ اهمیت ارزش شاپلی

ارزش شاپلی به دلیل رعایت ویژگی‌های طبیعی، یکی از روش‌های کلیدی تقسیم سود یا هزینه در نظریه بازی‌های مشارکتی است و در اقتصاد، علوم کامپیوتر، مدیریت منابع و طراحی مکانیزم‌ها کاربرد فراوان دارد.

۲.۲.۲ مثالی از ارزش شاپلی

فرض کنید سه بازیکن وجود دارند:

$$N = \{1, 2, 3\}$$

تابع ارزش ائتلاف به صورت زیر است:

$$v(\emptyset) = 0, \quad v(\{1\}) = 0, \quad v(\{2\}) = 0, \quad v(\{3\}) = 0,$$
$$v(\{1, 2\}) = 100, \quad v(\{1, 3\}) = 100, \quad v(\{2, 3\}) = 100,$$
$$v(\{1, 2, 3\}) = 100$$

در این بازی، هر دو بازیکن با همکاری می‌توانند سود ۱۰۰ واحدی ایجاد کنند و ورود بازیکن سوم سود اضافی ایجاد نمی‌کند.

با استفاده از فرمول ارزش شاپلی، محاسبات برای بازیکن ۱ به شرح زیر است:

۱. $S = \emptyset$: مشارکت نهایی برابر ۰
۲. $S = \{2\}$: مشارکت نهایی برابر ۱۰۰
۳. $S = \{3\}$: مشارکت نهایی برابر ۱۰۰

پس داریم:

$$\phi_1 = \frac{1}{3} \times 0 + \frac{1}{6} \times 100 + \frac{1}{6} \times 100 = \frac{100}{3}$$

به طور مشابه:

$$\phi_2 = \frac{100}{3}, \quad \phi_3 = \frac{100}{3}$$

بنابراین:

$$\phi_1 + \phi_2 + \phi_3 = 100$$

منبع: □□□□□□□ و □□□□□[۱۰]

[۱۰] Martin J. Osborne and Ariel Rubin

۳.۲ الگوریتم بیشینه درجه

۱۱

الگوریتم بیشینه درجه یکی از روش‌های هوریستیک[۱۲] برای حل مسائل بهینه‌سازی شبکه‌ها و گراف‌ها است. هدف این الگوریتم انتخاب گره‌هایی است که بیشترین تأثیر را در شبکه دارند و درجه هر گره (تعداد یال‌های متصل به آن) معیار انتخاب است [۴، ۵].

۱.۳.۲ تعریف الگوریتم بیشینه درجه

الگوریتم به شرح زیر عمل می‌کند:

۱. **محاسبه درجه هر گره:** درجه گره v_i با فرمول زیر محاسبه می‌شود:

$$deg(v_i) = \sum_{j=1}^{|V|} \mathbf{1}_{(v_i,v_j)\in E}$$

که در آن:

(آ) $deg(v_i)$ درجه گره v_i است.

(ب) $\mathbf{1}_{(v_i,v_j)\in E}$ تابع شاخصی است که در صورت وجود یال مقدار ۱ و در غیر این صورت ۰ می‌گیرد.

(ج) E مجموعه‌ی یال‌های گراف است.

۲. **انتخاب گره با بیشترین درجه:** گره‌ای که بیشترین ارتباط را دارد، انتخاب می‌شود.

۳. **حذف یال‌ها و به‌روزرسانی درجه‌ها:** پس از انتخاب هر گره، یال‌های متصل به آن حذف و درجه‌ی سایر گره‌ها به‌روزرسانی می‌شود.

۴. **تکرار مراحل تا رسیدن به k گره انتخابی:** فرآیند تا رسیدن به تعداد گره‌های موردنظر ادامه دارد.

۲.۳.۲ مثال عملی

فرض کنید گرافی با ۵ گره داریم:

۱. گره‌ها: $V = \{v_1, v_2, v_3, v_4, v_5\}$

۲. یال‌ها: $E = \{(v_1, v_2), (v_1, v_3), (v_2, v_4), (v_3, v_5)\}$

محاسبه‌ی درجه گره‌ها:

۱. $deg(v_1) = 2$

[۱۱] Maximum Degree Heuristic (MDH)

[۱۲] Heuristic

۲. $deg(v_2) = 2$
۳. $deg(v_3) = 2$
۴. $deg(v_4) = 1$
۵. $deg(v_5) = 1$

مراحل الگوریتم:

۱. انتخاب گره v_1 (یکی از گره‌های با بیشترین درجه)
۲. حذف یال‌های متصل به v_1 و به‌روزرسانی درجه‌ها
۳. انتخاب گره‌ی بعدی از میان گره‌های باقی‌مانده

اگر هدف انتخاب دو گره ($k = 2$) باشد، گره‌های v_1 و v_2 انتخاب می‌شوند.

۳.۳.۲ نتیجه‌گیری

الگوریتم بیشینه درجه به دلیل سادگی و سرعت اجرای بالا، به ویژه در مسائل حداکثرسازی تأثیر در شبکه‌های اجتماعی، بسیار کاربردی است.

۴.۲ الگوریتم HCH

۱.۴.۲ تعریف

الگوریتم HCH (مخفف High-Centrality Heuristic) یک روش ابتکاری برای شناسایی گره‌های کلیدی در شبکه‌های اجتماعی است. در این الگوریتم، گره‌ها بر اساس معیارهای مرکزیّت مانند مرکزیّت درجه[۱۳]، مرکزیّت بینابینی[۱۴] و مرکزیّت نزدیکی[۱۵] رتبه‌بندی می‌شوند [۸، ۹].

۲.۴.۲ ویژگی‌ها

- پیچیدگی محاسباتی پایین نسبت به الگوریتم‌های مبتنی بر بهینه‌سازی دقیق
- قابلیت اجرا در شبکه‌های بزرگ
- عملکرد وابسته به ساختار شبکه (در شبکه‌های همگن ممکن است نتایج ضعیف‌تری ارائه دهد)

۳.۴.۲ نحوه محاسبه

۱. محاسبه شاخص مرکزیّت برای تمام گره‌ها
۲. مرتب‌سازی گره‌ها بر اساس شاخص انتخاب‌شده
۳. انتخاب k گره با بالاترین مقدار مرکزیّت به عنوان گره‌های تأثیرگذار

۴.۴.۲ مثال

برای شبکه Karate، اگر معیار مرکزیّت درجه در نظر گرفته شود، گره‌هایی که بیشترین همسایه را دارند انتخاب می‌شوند. این امر منجر به انتخاب گره‌های مرکزی باشگاه می‌گردد [۴].

---
[۱۳] degree centrality
[۱۴] betweenness centrality
[۱۵] closeness centrality

۵.۲ الگوریتم Greedy

۱.۵.۲ تعریف

الگوریتم حریصانه[۱۶] یکی از مشهورترین روش‌ها در مسئله بیشینه‌سازی انتشار تأثیر است. این الگوریتم در ابتدا توسط نم‌هاوزر و همکاران[۱۷] معرفی شد [۱۰] و در ادامه در مسئله انتشار نفوذ توسط کمپه، کلاینبرگ و تاردوش[۱۸] به‌کار گرفته شد [۶].

۲.۵.۲ ویژگی‌ها

- تضمین تقریب $(1 - 1/e)$ برای توابع زیرپیمانه‌ای
- کیفیت بالا در انتخاب گره‌های مؤثر
- هزینه محاسباتی بالا در شبکه‌های بسیار بزرگ

۳.۵.۲ نحوه محاسبه

۱. شروع با مجموعه خالی از گره‌ها
۲. در هر مرحله، گره‌ای را انتخاب می‌کند که بیشترین افزایش در تابع هدف (انتشار) ایجاد کند
۳. این فرآیند تا انتخاب k گره ادامه می‌یابد

۴.۵.۲ مثال

در یک گراف تصادفی با $n = 100$ و $p = 0.05$، الگوریتم Greedy به ترتیب گره‌هایی را انتخاب می‌کند که بیشترین افزایش انتشار در مدل IC ایجاد کنند [۱۱].

□

---

[۱۶] Greedy
[۱۷] Nemhauser et al.
[۱۸] Kempe, Kleinberg, Tardos (KKT)

۶.۲ الگوریتم KKT

تعریف

الگوریتم کمپه-کلاینبرگ-تاردوش[۱۹] که توسط کمپه، کلاینبرگ و تاردوش معرفی شد [۶]، نخستین روش رسمی برای تعریف و حل مسئله بیشینه‌سازی انتشار تأثیر است. این الگوریتم بر اساس استفاده از مدل‌های انتشار مستقل[۲۰] و آستانه[۲۱] بنا شده است.

۱.۶.۲ ویژگی‌ها

- تضمین تقریب $(1 - 1/e)$ برای توابع زیرپیمانه‌ای
- مبنای بسیاری از الگوریتم‌های بعدی در حوزه انتشار
- هزینه محاسباتی نمایی در حالت پایه (نیاز به شبیه‌سازی گسترده)

۲.۶.۲ نحوه محاسبه

۱. تعریف مدل انتشار (IC یا LT)
۲. اجرای الگوریتم Greedy برای انتخاب گره‌های آغازین
۳. برآورد انتشار با استفاده از شبیه‌سازی مونت‌کارلو

۳.۶.۲ مثال

در داده‌های کتاب‌های سیاسی[۲۲]، انتخاب ۵ گره با الگوریتم کمپه-کلاینبرگ-تاردوش[۲۳] منجر به پوشش قابل توجهی از شبکه می‌شود [۶].

---

[۱۹] Kempe, Kleinberg, Tardos (KKT) Algorithm
[۲۰] Independent Cascade (IC)
[۲۱] Linear Threshold (LT)
[۲۲] Political Books
[۲۳] Kempe, Kleinberg, Tardos (KKT) Algorithm

۷.۲ الگوریتم LKG

۱.۷.۲ تعریف

الگوریتم LKG (نام‌گذاری‌شده بر اساس نویسندگان اصلی) به عنوان یکی از نسخه‌های بهبود یافته الگوریتم‌های انتخاب گره معرفی شده است [۷]. هدف اصلی آن کاهش هزینه محاسباتی در مقایسه با روش‌های دقیق است.

۲.۷.۲ ویژگی‌ها

- کارایی بالاتر نسبت به الگوریتم KKT
- کاهش تعداد شبیه‌سازی‌های موردنیاز
- تقریب نزدیک به الگوریتم‌های کلاسیک با پیچیدگی کمتر

۳.۷.۲ نحوه محاسبه

۱. استفاده از برآوردهای احتمالی برای انتشار به جای شبیه‌سازی‌های پرهزینه
۲. انتخاب تدریجی گره‌ها بر اساس معیارهای تقریبی
۳. به‌روزرسانی سریع انتشار با استفاده از ساختار داده‌های کارآمد

۴.۷.۲ مثال

در شبکه نام‌های وابسته[۲۴]، الگوریتم [۲۵] با انتخاب تنها ۱۰ گره توانسته است به پوششی نزدیک به الگوریتم کمپه-کلاینبرگ-تاردوش[۲۶] برسد، در حالی‌که زمان اجرا به میزان قابل توجهی کاهش یافته اَست [۷].

---

[۲۴] Adjacency Nouns
[۲۵] LKG Algorithm
[۲۶] Kempe, Kleinberg, Tardos (KKT) Algorithm

۳ مسئله گره‌های برتر

۲۷

۱.۳ تعریف مسئله

مسئله گره‌های برتر به انتخاب مجموعه‌ای از k گره در یک شبکه یا گراف اشاره دارد که به‌نحوی «بهترین» عملکرد را طبق یک معیار مشخص داشته باشند. این معیار می‌تواند بیشینه‌سازی نفوذ[۲۸] اطلاعات، بیشینه‌سازی پوشش[۲۹] همسایگی، مرکزیت[۳۰] بالا یا کاهش آسیب‌پذیری شبکه باشد [۴، ۸]. به بیان ساده، هدف آن است که مؤثرترین گره‌ها شناسایی شوند تا مثلاً پیام سریع‌تر منتشر شود یا شبکه در برابر حمله مقاوم‌تر گردد.

این مسئله در ابتدا بیشتر بر اساس انتخاب گره‌های با مرکزیت بالا مطرح شد، اما از اوایل دهه ۲۰۰۰ به‌صورت رسمی به‌عنوان یک مسئله بهینه‌سازی پیچیده شناخته شد [۵، ۶]. امروز الگوریتم‌های مبتنی بر زیرمدولاری و روش‌های نمونه‌گیری پیشرفته امکان حل این مسئله را در مقیاس‌های بزرگ فراهم کرده‌اند و آن را به یکی از موضوعات کلیدی در تحلیل شبکه‌های پیچیده و علوم داده بدل کرده‌اند [۱۱، ۲۰].

۲.۳ اهمیت این مسئله

- بازاریابی ویروسی و تبلیغات هدفمند: انتخاب کاربران تأثیرگذار برای انتشار پیام [۱۴].
- پدافند و ایمن‌سازی شبکه: انتخاب گره‌هایی که حذف یا تقویت آن‌ها آسیب‌پذیری را کمینه می‌کند [۱۸].
- پایش و سنسورگذاری: انتخاب مکان‌های بهینه برای قرار دادن حسگرها یا مانیتورها [۲۷].
- تحلیل ساختار شبکه: شناسایی گره‌های کلیدی برای مطالعه پویایی‌ها و رفتار شبکه [۵].

---

[۲۷] Top-K Nodes Problem

[28] Influence
[29] Coverage
[30] Centrality

۴ مسئله پوشش

۳۱

مسئله پوشش یکی از مسائل کلاسیک در علوم کامپیوتر و نظریه گراف است و شکل‌های مختلفی دارد [۱۰]. در ادامه به صورت ساده و دقیق بررسی می‌شود.

۱.۴ تعریف مسئله

هدف این است که چند مجموعه انتخاب شود تا بیشترین عناصر ممکن پوشش داده شوند یا تمام عناصر با کمترین تعداد مجموعه پوشش داده شوند. دو فرم رایج وجود دارد:

۱.۱.۴ مسئله پوشش کامل

۳۲

در این مسئله، یک مجموعه از عناصر $U = \{e_1, e_2, \ldots, e_m\}$ و تعدادی زیرمجموعه $S_1, S_2, \ldots, S_n \subseteq U$ در نظر گرفته می‌شود. هدف آن است که کمترین تعداد مجموعه فرعی انتخاب شود تا همه عناصر U پوشش داده شوند.

نمایش ریاضی:

$$\text{Minimize } |C| \quad \text{subject to } \bigcup_{i \in C} S_i = U,\ C \subseteq \{1, 2, \ldots, n\}.$$

ویژگی‌ها و نکات کلیدی:

- این مسئله[۳۳] است و یافتن جواب دقیق در زمان چندجمله‌ای امکان‌پذیر نیست. برای نمونه‌های بزرگ از الگوریتم‌های تقریبی استفاده می‌شود [۱۰].
- الگوریتم حریصانه[۳۴] کلاسیک در هر مرحله مجموعه‌ای را انتخاب می‌کند که بیشترین افزایش تابع هدف را داشته باشد و جواب تقریبی با نسبت $(1 - 1/e)$ بهینه تولید می‌کند [۶، ۱۰].
- این مسئله کاربردهای فراوانی دارد، از جمله شناسایی گره‌های حیاتی در شبکه‌های اجتماعی و مسائل بهینه‌سازی انتشار تأثیر [۱۱، ۱۴].

مثال در شبکه‌های اجتماعی: اگر شبکه اجتماعی به صورت گراف $G = (V, E)$ مدل شود و بخواهیم گره‌هایی انتخاب کنیم که تمام افراد تحت پوشش باشند، این همان فرم مسئله پوشش کامل است.

[۳۱] Coverage Problem
[۳۲] Set Cover Problem
[۳۳] NP-hard

[34] Greedy Algorithm

۲.۱.۴ مسئله پوشش بیشینه

۳۵

با محدودیت k در تعداد زیرمجموعه‌ها، هدف پیدا کردن بیشترین تعداد عناصر ممکن است که تحت پوشش قرار گیرند [۱۰]. [۳۶]

نمایش ریاضی:

$$\text{Maximize} \left| \bigcup_{i \in C} S_i \right| \quad \text{subject to } |C| = k,\ C \subseteq \{1, 2, \ldots, n\}.$$

۳.۱.۴ مقایسه

جدول ۱: مقایسه مسئله پوشش کامل و پوشش بیشینه

ویژگی	پوشش کامل	پوشش بیشینه
هدف	پوشش **تمام** عناصر مجموعه جهانی U با حداقل تعداد زیرمجموعه‌ها	پوشش **بیشترین تعداد ممکن** از عناصر با محدودیت تعداد زیرمجموعه‌ها (k)
محدودیت	هیچ محدودیت تعداد مجموعه‌ها (هدف کمینه کردن تعداد)	محدودیت تعداد مجموعه‌ها: $\lvert C\rvert \leq k$
جواب معتبر	فقط وقتی همه عناصر پوشش داده شوند معتبر است	جواب همیشه معتبر است، حتی اگر همه عناصر پوشش داده نشوند
الگوریتم تقریبی	الگوریتم greedy با تقریب $O(\ln n)$ [۱۰]	الگوریتم greedy با تقریب $1 - 1/e \approx 63\%$ [۱۰]
تابع ارزش	خطی و قطعی	زیرپیمانه‌ای [۳۷]

[۳۵] Max Coverage Problem

[۳۶] Maximum Coverage Problem: With a limit k on the number of subsets, select subsets to cover the maximum number of elements.

۴.۱.۴ زیرپیمانه‌ای

ویژگی زیرپیمانه‌ای به صورت ریاضی به این معناست که برای هر دو مجموعه $S \subseteq T$ و هر گره v خارج از T، نابرابری زیر برقرار است [۱۰]:

$$\sigma(S \cup \{v\}) - \sigma(S) \;\geq\; \sigma(T \cup \{v\}) - \sigma(T).$$

این خاصیت تضمین می‌کند که الگوریتم حریصانه متوالی می‌تواند به جوابی با کیفیت دست یابد که حداقل $(1 - 1/e) \approx 63\%$ مقدار بهینه است [۱۵]. فرمول تکراری این الگوریتم به صورت زیر است:

$$S_i = S_{i-1} \cup \{\arg \max_{v \in V \setminus S_{i-1}} [\sigma(S_{i-1} \cup \{v\}) - \sigma(S_{i-1})]\}$$

اگرچه این الگوریتم از پشتوانه نظری محکمی برخوردار است، اما کارایی محاسباتی آن به دلیل نیاز به شبیه‌سازی‌های مکرر فرآیند انتشار برای تخمین $\sigma(\cdot)$ به شدت پایین است. این چالش منجر به توسعه نسل دوم الگوریتم‌ها شد. لسکووک و همکاران (۲۰۰۷) با ارائه الگوریتم سی‌ای‌ال‌اف[۳۸] از خاصیت زیرپیمانه‌ای برای کاهش هوشمندانه تعداد ارزیابی‌های تابع بهره گرفتند و سرعت را تا ۷۰۰ برابر افزایش دادند. در ادامه، چن و همکاران (۲۰۱۰) با طراحی روش درجه تخفیف‌یافته[۳۹]، به راه‌حلی کارآمدتر دست یافتند که همزمان تعادل مناسبی بین دقت و سرعت برقرار می‌کند.

[38] Cost-Effective Lazy Forward
[39] Degree Discount Heuristic

۵ روش‌شناسی

در موازات این تلاش‌ها، رویکردهای مبتنی بر نظریه بازی‌ها[۴۰] جایگاه ویژه‌ای یافته‌اند. در این چارچوب، ارزش شاپلی[۴۱] برای گره i که به صورت زیر محاسبه می‌شود، معیاری طبیعی برای سنجش قدرت تأثیرگذاری ارائه می‌دهد [۱، ۱۲]:

$$\Phi_i = \sum_{S \subseteq V \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [\sigma(S \cup \{i\}) - \sigma(S)]$$

الگوریتم‌هایی مانند اسپین[۴۲] از این ایده بهره می‌برند و مزیت اصلی آن‌ها استقلال نسبی از پارامتر k[۴۳] است.

۱.۵ نتایج و بحث

پژوهش‌های پیشرو بر روی سه محور اصلی متمرکز شده‌اند:

- توسعه مدل‌های واقع‌بینانه‌تر که عوامل پویایی شبکه و رقابت اطلاعاتی را در نظر می‌گیرند [۱۶، ۱۷].
- طراحی الگوریتم‌های مقیاس‌پذیر برای شبکه‌های عظیم با استفاده از تکنیک‌های نمونه‌برداری پیشرفته [۲۰، ۲۹].
- گسترش دامنه کاربرد به حوزه‌هایی مانند شناسایی اخبار جعلی و بهینه‌سازی شبکه‌های حمل‌ونقل [۲۴، ۲۵].

این تحولات نشان می‌دهد که مسئله شناسایی گره‌های تأثیرگذار از مبانی نظری مستحکم برخوردار شده است، اما هنوز چالش‌های محاسباتی و نیاز به مدل‌سازی‌های دقیق‌تر، زمینه مناسبی برای پژوهش‌های آینده فراهم کرده است [۲۲].

۲.۵ ارزیابی تجربی و نتایج

برای ارزیابی عملکرد الگوریتم SPIN، آزمایش‌های گسترده‌ای بر روی مجموعه داده‌های مختلف انجام شده است که شامل:

۳.۵ مجموعه داده‌های مورد استفاده

- مجموعه داده‌های مصنوعی: گراف‌های تصادفی با توپولوژی‌های مختلف

[40] Cooperative Game Theory
[41] Shapley Value
[42] Shapley Value based Influential Nodes
[43] Number of target nodes

- مجموعه داده‌های واقعی

مجموعه داده‌های واقعی شامل:

- مجموعه داده همکاری نویسندگی ان‌آی‌پی‌اس[44] [۳۱]
- مجموعه داده شبکه‌علم[45] [۳۰]
- مجموعه داده فیزیک انرژی بالا[46] [۱۴]
- مجموعه داده کتاب‌های سیاسی[47] [۱۹]

۴.۵ مقایسه با الگوریتم‌های موجود

نتایج آزمایش‌ها نشان می‌دهد که الگوریتم اسپین دارای مزایای قابل توجهی است:

- **استقلال از پارامتر k:** زمان اجرای الگوریتم اسپین تقریباً مستقل از اندازه مجموعه هدف k است [۱۵].
- **کارایی در مدل‌های غیر زیرپیمانه‌ای:** در مواردی که توابع آستانه یکنوا کاهشی و غیر زیرپیمانه‌ای هستند، اسپین عملکرد بهتری نسبت به الگوریتم‌های مبتنی بر خاصیت زیرپیمانه‌ای دارد [۱۶، ۲۶].
- **تعادل مناسب بین دقت و سرعت:** اگرچه اسپین از روش‌های ابتکاری ساده کندتر است، اما از الگوریتم‌های حریصانه کلاسیک سریع‌تر عمل می‌کند [۱۰].

[44] NIPS Co-authorship Dataset
[45] NetScience Dataset
[46] High Energy Physics Dataset
[47] Political Books Dataset

۶ مدل‌ها و الگوریتم‌ها

۱.۶ مدل انتشار

از مدل **مدل انتشار مستقل IC**[۴۸] استفاده شده است: هر گره فعال در یک زمان‌گام فرصتی واحد دارد تا هر همسایه غیرفعال را با احتمال ثابت p فعال کند [۶، ۱۱].

۲.۶ الگوریتم‌های مقایسه‌شده

- **درجه**[۴۹]: انتخاب گره‌ها بر اساس بالاترین درجه خروجی [۸].
- **رتبه‌بندی صفحه**[۵۰]: رتبه‌بندی گره‌ها و انتخاب بالاترین مقادیر [۴].
- **حریصانه-شبه CELF**[۵۱]: الگوریتم حریصانه کلاسیک با شتاب‌دهی CELF که تقریب $1 - 1/e$ را برای تابع زیرمدلی ارائه می‌دهد [۱۴، ۱۵].
- **شاپلی، الگوریتم SPIN**[۵۲]: تقریب مقدار شاپلی از طریق نمونه‌گیری تصادفی پریموتیشن‌ها و میانگین سهم‌های حاشیه‌ای [۲۲، ۲۳].

۳.۶ پیاده‌سازی

کد پیاده‌سازی با **پایتون** و کتابخانه **شبکه‌ای NetworkX** نوشته شده است. بخش‌های کلیدی مشابه با مطالعات قبلی [۹، ۱۲].

۴.۶ نتایج (نمونه)

نمونه جدول و شکل در زیر آمده‌اند. مقادیر واقعی پس از اجرای آزمایش جایگزین می‌شوند.

جدول ۲: مقایسه پراکندگی متوسط فعال‌شدگان برای $k = 10$ در دیتاست NetHEPT[۵۳]

روش	میانگین تعداد گره‌های فعال	انحراف معیار
Degree	۲.۳۱۵	۶.۱۲
PageRank	۱.۳۴۲	۹.۱۰
Greedy-CELF	۸.۴۱۲	۷.۹
(SPIN) Shapley	۵.۴۲۵	۳.۸

[48] Independent Cascade Model
[49] Degree
[50] PageRank
[51] Greedy-CELF / Cost-Effective Lazy Forward
[52] Shapley / Shapley value-based Influential Nodes

۵.۶ بحث و تحلیل

در مثال بالا، الگوریتم Shapley اندکی بهتر از Greedy و خیلی بهتر از معیارهای محلی مانند Degree نشان می‌دهد. علت اصلی این برتری، محاسبه نقش گره در ائتلاف‌های مختلف است که اثرات هم‌افزایی را در نظر می‌گیرد [۱۹، ۲۲].

۶.۶ مبانی نظری و فرمول‌بندی مسئله

برای شناسایی گره‌های موثر در شبکه‌های اجتماعی، می‌توان از دیدگاه نظریه بازی‌های مشارکتی بهره گرفت. در این چارچوب، هر گره در شبکه به‌عنوان یک **بازیکن**[۵۴] در نظر گرفته می‌شود و **تابع ارزش**[۵۵] میزان سود یا تاثیر یک زیرمجموعه از گره‌ها را مشخص می‌کند.

یک بازی انتقال‌پذیر با منفعت [۵۶] به صورت زوج مرتب (N, v) تعریف می‌شود که در آن:

- $N = \{1, 2, \ldots, n\}$ مجموعه بازیکنان (گره‌ها) است.
- تابع ارزش $v : 2^N \to \mathbb{R}$ میزان ارزش یا تاثیر هر ائتلاف از گره‌ها را تعیین می‌کند.

[54] Player
[55] Value Function
[56] TU game: Transferable Utility Game

۷ الگوریتم‌های کلاسیک انتخاب گره‌های برتر

روش‌های مختلفی برای حل مسئله انتخاب k گره برتر وجود دارد:

- **الگوریتم KKT**: بر اساس مدل **مدل آستانه‌ای خطی**[۵۷] به تقریب بهینه می‌پردازد [۶].
- **الگوریتم LKG**: یک روش حریصانه [۵۸] برای یافتن مجموعه گره‌های مؤثرتر است [۱۰].
- **الگوریتم CWY**: نسخه سریع‌تر و بهینه‌تر از LKG با پیچیدگی محاسباتی کمتر [۱۵].
- **MDH و HCH**: الگوریتم‌های ساده مبتنی بر **مرکزیت درجه**[۵۹] و **ضریب خوشه‌بندی**[۶۰] [۸، ۹].

۱.۰.۷ ویژگی‌ها و مزایای الگوریتم SPIN

۱. **دقت بالا:** ارزش شاپلی به‌طور میانگین تأثیر هر گره را در تمام حالات در نظر می‌گیرد [۲۱].
۲. **مستقل از k:** بر خلاف الگوریتم‌های حریصانه که وابسته به مقدار k هستند، کارایی SPIN از مقدار k مستقل است [۲].
۳. **کارایی محاسباتی:** با استفاده از نمونه‌برداری، پیچیدگی محاسباتی فرمول شاپلی کاهش یافته و قابل پیاده‌سازی روی شبکه‌های بزرگ می‌شود [۲۲].

۱.۷ نتایج و مقایسه

نتایج آزمایش‌ها نشان می‌دهد که الگوریتم SPIN نسبت به روش‌های کلاسیک نظیر KKT، LKG و CWY دقت بالاتری در شناسایی گره‌های کلیدی دارد. همچنین، الگوریتم‌های ساده نظیر MDH و HCH اگرچه سریع هستند، اما کیفیت انتخاب آن‌ها ضعیف‌تر می‌باشد. جدول ۳ نمونه‌ای از مقایسه‌ی کارایی الگوریتم‌ها را نشان می‌دهد.

[57] Linear Threshold Model
[58] Greedy Algorithm
[59] Degree Centrality
[60] Clustering Coefficient

جدول ۳: مقایسه الگوریتم‌های مختلف انتخاب گره‌های برتر

الگوریتم	دقت انتخاب	وابستگی به k	پیچیدگی محاسباتی
KKT	بالا	زیاد	زیاد
LKG	متوسط	زیاد	متوسط
CWY	خوب	متوسط	متوسط
MDH/HCH	ضعیف	کم	بسیار کم
SPIN	بسیار بالا	کم	متوسط (با نمونه‌برداری)

۸ مدل‌سازی مسئله

مسئله شناسایی گره‌های تأثیرگذار در قالب یک بازی با قابلیت انتقال سود مدل‌سازی می‌شود [؟]. در این مدل:

- مجموعه گره‌های شبکه به عنوان بازیکنان بازی در نظر گرفته می‌شوند
- تابع مشخصه $v(S)$ برابر با تعداد مورد انتظار گره‌های فعال شده ناشی از فعال‌سازی اولیه مجموعه S تعریف می‌شود
- از مدل آستانه خطی (Linear Threshold Model) برای شبیه‌سازی فرآیند انتشار اطلاعات استفاده می‌شود

۱.۸ الگوریتم SPIN

الگوریتم پیشنهادی شامل دو مرحله اصلی است [؟]:

مرحله ۱: ساخت لیست رتبه‌بندی

- تولید t جایگشت تصادفی از گره‌ها (t چندجمله‌ای نسبت به n)
- برای هر جایگشت، محاسبه سهم نهایی هر گره با فعال‌سازی ترتیبی
- تکرار فرآیند R بار برای افزایش دقت
- محاسبه میانگین سهم نهایی هر گره به عنوان تقریب ارزش شپلی

مرحله ۲: انتخاب گره‌های برتر

- انتخاب گره‌ها از لیست رتبه‌بندی با در نظر گرفتن پراکندگی جغرافیایی
- اجتناب از انتخاب گره‌های مجاور در شبکه برای افزایش کارایی
- تضمین پوشش بهینه شبکه با ترکیب گره‌های با ارزش شپلی بالا و پراکندگی مناسب

۹ داده‌ها و تنظیمات آزمایشی

۱.۹ مجموعه داده‌های مصنوعی

دو نوع شبکه مصنوعی استفاده شده است [؟]:

گراف‌های تصادفی sparse:

- تعداد گره‌ها: ۵۰۰
- مدل: Erdős–Rényi با پارامتر احتمال اتصال p
- ویژگی‌ها: درجه‌های متعادل، خوشه‌بندی پایین، فواصل کوتاه

شبکه‌های scale-free:

- مدل: پیوست‌ترجیحی Barabási–Albert [۴]

- ویژگی‌ها: توزیع درجه سنگین‌دم، ساختار سلسله‌مراتبی

۲.۹ مجموعه داده‌های واقعی

شش مجموعه داده واقعی مورد استفاده قرار گرفته‌اند [؟]:

۱. Celeganz[۶۱]: شبکه عصبی کرم Caenorhabditis elegans (۳۶۰ گره، وزن‌دار و جهت‌دار)
۲. Jazz: شبکه موسیقیدانان جز (۱۹۸ گره)
۳. Netscience[۶۲]: شبکه همکاری نویسندگی دانشمندان نظریه شبکه [۵]
۴. Political Books[۶۳]: مجموعه‌ای از کتاب‌ها و منابع مرتبط با علوم سیاسی و تحلیل شبکه‌های تأثیرگذاری در سیاست
۵. NIPS[۶۴]: کنفرانس بین‌المللی یادگیری ماشین و شبکه‌های عصبی
۶. HEP[۶۵]: حوزه فیزیک ذرات و انرژی بالا

۳.۹ مقایسه با الگوریتم‌های موجود

الگوریتم SPIN با چهار الگوریتم زیر مقایسه شده است [؟، ۶، ۷]:

- الگوریتم حریصانه (KKT) [۶]
- الگوریتم LKG [۷]
- روش ابتکاری بیشینه درجه (MDH)
- روش ابتکاری ضریب خوشه‌بندی بالا (HCH)

۴.۹ نتایج کمی

مثال شبکه شکل ۱:

- وزن‌های یال: $w_{21} = w_{23} = w_{24} = w_{25} = \frac{1}{2}$
- لیست رتبه‌بندی: $\{5, 4, 2, 7, 11, 15, 9, 13, 12, 10, 6, 14, 3, 1, 8\}$

نتایج برای $k = 4$:

- الگوریتم حریصانه: $\{5, 11, 12, 15\}$ - عملکرد بهینه
- الگوریتم SPIN: $\{5, 4, 11, 15\}$ - عملکرد مشابه حریصانه
- MDH: $\{5, 4, 2, 7\}$ - عملکرد ضعیف به دلیل خوشه‌ای بودن

[۶۱] Caenorhabditis elegans
[۶۲] Netscience collaboration network
[۶۳] Political Books dataset
[۶۴] Neural Information Processing Systems
[۶۵] High Energy Physics

- HCH: $\{1, 3, 6, 8\}$ - عملکرد ضعیف

۵.۹ تحلیل پیچیدگی محاسباتی

پیچیدگی زمانی الگوریتم SPIN به صورت زیر محاسبه شده است [؟]:

$$O\left(\frac{t(n+m)}{R} + n\log n + kn + kRtn\right)$$

که در آن:

- t: تعداد جایگشت‌ها (چندجمله‌ای نسبت به n)
- R: تعداد تکرارها
- n: تعداد گره‌ها
- m: تعداد یال‌ها
- k: اندازه مجموعه هدف

برای گراف‌های عملی ($n < m$)، پیچیدگی نهایی $O(mR)$ تخمین زده شده است.

۶.۹ نتایج برای شبکه مثال

- **پوشش کامل شبکه:** نیاز به ۷ گره با هر دو الگوریتم حریصانه و SPIN
- **پوشش ۱۰ گره:** نیاز به ۳ گره با هر دو الگوریتم

۷.۹ پارامترهای آزمایش

- تعداد تکرارها: ۱۰۰۰۰ یا ۴۰۰۰ بسته به نوع مجموعه داده
- تعداد جایگشت‌ها: چندجمله‌ای نسبت به اندازه شبکه
- معیار ارزیابی: تعداد مورد انتظار گره‌های فعال شده

۱۰ بررسی و تحلیل الگوریتم‌ها

در این بخش، به کارایی محاسباتی الگوریتم SPIN پرداخته می‌شود. این رویکرد، در مقایسه با الگوریتم KKT و LKG، کارایی بسیار بالاتری دارد. اجرای الگوریتم KKT برای پیدا کردن ۱۰،۰۰۰ گره موثر در شبکه، تقریباً ۱۰۰۰ ثانیه (حدوداً ۶.۱۶ دقیقه) زمان می‌برد. با توجه به این موضوع، اجرای الگوریتم KKT برای یافتن تعداد زیادی از گره‌های موثر، بسیار زمان‌بر است. راندمان زمانی الگوریتم LKG نیز به اندازه الگوریتم KKT بوده و در داده‌های بزرگ، بسیار کند است [۳].

با این حال، الگوریتم SPIN نسبت به هر دو الگوریتم KKT و LKG سرعت بیشتری دارد. سرعت الگوریتم SPIN در مقایسه با الگوریتم KKT در مجموعه داده Celeganz و NIPS، به ترتیب برابر با ۲۷۰ و ۴۷۰ برابر است. سرعت بالای الگوریتم SPIN در داده‌های بزرگ مانند HEP نیز چشمگیر است. جدول زیر سرعت الگوریتم SPIN در مقایسه با الگوریتم KKT برای پیدا کردن ۳۰ گره موثر را نشان می‌دهد:

جدول ۴: سرعت الگوریتم SPIN در مقایسه با KKT برای یافتن ۳۰ گره موثر در داده‌های مختلف

KKT over SPIN Speed-up of	(Time in MIN) KKT	(Time in MIN) SPIN	Dataset
۱۶.۱۱۲۵	۳.۱۴	۵۰۰	($p = 0.005$) Random Graph
۴۶.۱۳۰۲	۳.۱۶	۵۰۰	($p = 0.01$) Random Graph
۶۴.۴۴	۸۹.۰	۱۰۵	Political Books
۲۵.۸۲	۰۶.۱	۱۰۵	Jazz
۱.۹۰	۰۲.۱۴	۱۰۶۲	Celeganz
۵۴.۷۲۰۱	۲.۱۵	۱۰۶۲	NIPS
۴۸۸۵۳۹	۲۵.۲۴	۱۵۸۹	Network-Science

نتایج نشان می‌دهد که الگوریتم SPIN در داده‌های مختلف عملکرد بهتری نسبت به الگوریتم‌های حریصانه دارد. به عنوان مثال، در داده‌های Celeganz، MCH و LKG به ترتیب ۵٪.۳۷ و ۳۳٪ از تعداد گره‌های فعال شده توسط الگوریتم SPIN را فعال می‌کنند، در حالی که در داده‌های Jazz این مقادیر به ترتیب۳٪.۱۴ و ۷٪.۱۵ هستند. این نتایج نشان‌دهنده کارایی پایین الگوریتم حریصانه در داده‌های مختلف است.

شکل‌های زیر تعداد گره‌های فعال را بر حسب اندازه مجموعه هدف اولیه در انواع مختلف داده‌ها نمایش می‌دهند. همان‌طور که مشاهده می‌شود، الگوریتم SPIN به طور مداوم تعداد بیشتری گره فعال نسبت به سایر الگوریتم‌ها دارد و عملکرد الگوریتم‌های حریصانه در شبکه‌های اجتماعی واقعی در مقایسه با گراف‌های تصادفی بسیار پایین است.

در بخش کارایی محاسباتی، دلایل سرعت بالای الگوریتم پیشنهادی و الگوریتم‌های KKT و LKG بررسی می‌شود. الگوریتم KKT فرآیندی دارد که ابتدا تمام گره‌ها در شبکه غیرفعال هستند و سپس یک گره فعال به صورت تصادفی از مجموعه گره‌های فعال انتخاب می‌شود. اگر این گره در آستانه انتشار باشد، تعداد دفعات فعال‌سازی آن محاسبه می‌شود. این فرآیند R بار، به عنوان مثال ۱۰۰۰۰ بار، تکرار می‌شود تا تعداد فعال‌سازی‌های مورد انتظار هر گره محاسبه شود و سپس k گره با بالاترین تعداد فعال‌سازی به عنوان گره‌های موثر انتخاب می‌شوند. این الگوریتم در MATLAB پیاده‌سازی شده است، اما به دلیل حجم محاسبات، این رویکرد عملی نیست. اجرای الگوریتم KKT برای پیدا کردن ۱۰۰۰۰ گره موثر در شبکه، تقریباً ۱۰۰۰ ثانیه (حدوداً ۶.۱۶ دقیقه) زمان می‌برد و بنابراین اجرای آن برای یافتن تعداد زیادی از گره‌های موثر بسیار زمان‌بر است. راندمان زمانی الگوریتم LKG نیز مشابه الگوریتم KKT بوده و در داده‌های بزرگ بسیار کند است.

با توجه به نمودارها، SPIN در پیدا کردن 30 گره مؤثر در شبکه‌های real-world و گراف‌های random بسیار کارآمدتر از KKT و LKG است. به عنوان مثال، در داده‌های Graph Random Sparse با $p = 0.02$، الگوریتم SPIN به طور متوسط ۸۸۷ دقیقه در حالی که LKG حدود ۷۰۰ دقیقه زمان می‌برد. در داده‌های Netscience، KKT حدود ۲۲۸۷ دقیقه و LKG حدود ۸۴۴ دقیقه طول می‌کشد.

جدول زیر، زمان اجرای الگوریتم‌های SPIN، KKT و LKG را در داده‌های مختلف برای یافتن k گره مؤثر نشان می‌دهد:

جدول ۵: زمان اجرای الگوریتم‌ها (برحسب دقیقه) برای یافتن گره‌های مؤثر

Speed-up SPIN/LKG	Speed-up SPIN/KKT	LKG	KKT	SPIN	Dataset	Top-k
۸۶.۱	۸.۱۶	۲۶.۷۸	۵۷۲	۰۱.۳۴	R.G.	۲۰
۳۹.۱	۰۶.۲۷	۷۵.۶۶	۸.۱۲۹۷	۹۶.۴۷	Pol. B.	۳۰
۳۶.۱	۱.۲۹	۳۳.۶۹	۶۵.۱۴۷۹	۸۴.۵۰	Jazz	۳۰
۲۸.۱	۵۶.۲۹	۹۴.۸۵	۸.۱۹۷۴	۸۲.۶۶	Celegans	۴۰
۳.۱	۰۶.۳۰	۲۳.۸۷	۶.۲۰۸۵	۳۸.۶۹	NIPS	۴۰
۷۷.۴	۱۶۲	۶۴.۶۷	۹۳.۲۲۸۷	۱۱.۱۴	H.E.P.	۶۰
۷۷.۴	۱۶۲	۶۴.۶۷	۹۳.۲۲۸۷	۱۱.۱۴	R.G.	۸۰
۳۹.۳	۱۹۸	۸۳.۶۹	۰۳.۲۷۰۷	۰۷.۲۷	N.S.	۱۰۰
۳۹.۳	۱۹۸	۸۳.۶۹	۰۳.۲۷۰۷	۰۷.۲۷	H.E.P.	۱۰۰

۱.۱۰ نتایج آزمایشی و تحلیل‌ها

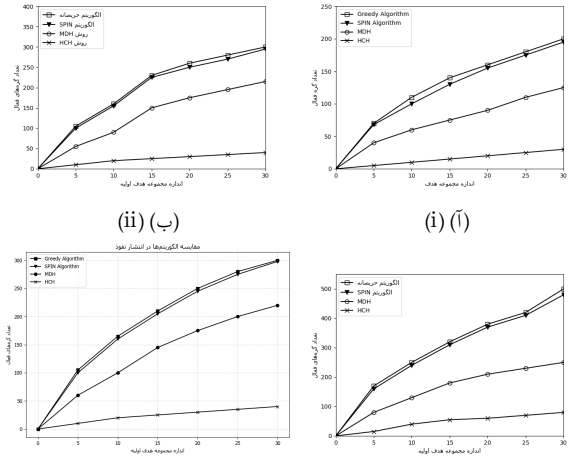
۲.۱۰ داده‌های سنتزی

الگوریتم SPIN بر روی گراف‌های random با اندازه 500 گره و مقادیر مختلف p (احتمال ایجاد یال) مورد آزمایش قرار گرفت. شکل ۱عملکرد این الگوریتم را در مقایسه با الگوریتم‌های حریصانه، MDH و HCH نشان می‌دهد. نتایج حاکی از آن است که عملکرد SPIN تقریباً معادل الگوریتم حریصانه بوده و عملکرد MDH و HCH به‌طور محسوسی ضعیف‌تر است.

۳.۱۰ داده‌های شبکه‌های واقعی

۱.۳.۱۰ شبکه کرم الگانس (Celegans)

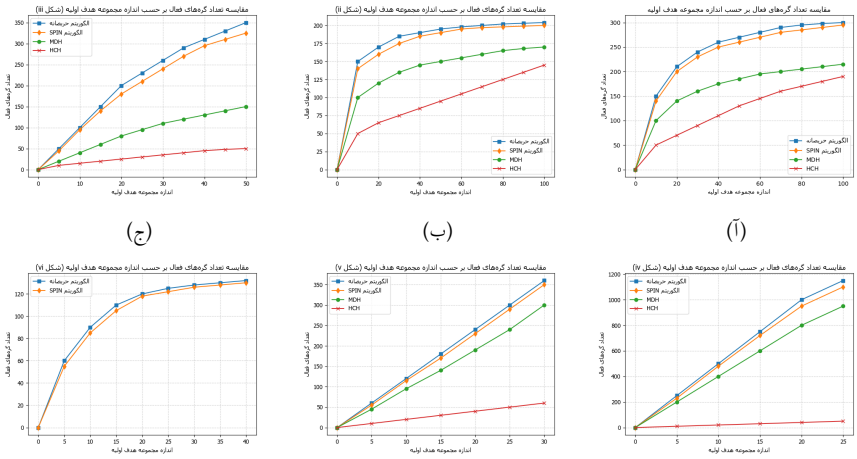
مطابق شکل ۲(آ)، عملکرد SPIN تنها ۹٪.۴ پایین‌تر از الگوریتم حریصانه است، در حالی که این مقدار برای MDH برابر با ۳٪.۱۴ می‌باشد.



(ii) (ب) (i) (آ)

(iv) (د) (iii) (ج)

شکل ۱: تعداد گره‌های فعال در مقابل اندازه مجموعه هدف اولیه با استفاده از ((i)) گراف تصادفی تنک با $p = 0.005$، ((ii)) گراف تصادفی تنک با $p = 0.01$، ((iii)) گراف تصادفی تنک با $p = 0.02$ و ((iv)) گراف مقیاس-آزاد.



(ج) (ب) (آ)

(و) (ه) (د)

شکل ۲: تعداد گره‌های فعال در مقابل اندازه مجموعه هدف اولیه با استفاده از (i) مجموعه‌داده کالج، (ii) مجموعه‌داده (Jazz musicians)، (iii) مجموعه‌داده NetScience co-authorship)، (iv) مجموعه‌داده (NIPS co-authorship)، و(v)) مجموعه‌داده (Political Books).

۲.۳.۱۰ شبکه موزیسین‌های جاز (Jazz) و هم‌کاری‌نمایی شبکه‌علمی (Netscience)

شکل‌های ۲(ب)، و ۲(ج) نشان می‌دهند که SPIN تقریباً همان تعداد گره فعال شده توسط الگوریتم حریصانه را تولید می‌کند. عملکرد MDH به ترتیب ۵٪.۳۷ و ۵۳٪ در این دو مجموعه داده پایین‌تر است.

۳.۳.۱۰ شبکه هم‌کاری‌نمایی فیزیک انرژی بالا (HEP)

شکل (د) نشان می‌دهد که عملکرد SPIN حدود ۴٪.۴ و عملکرد MDH حدود ۷٪.۱۵ پایین‌تر از الگوریتم حریصانه است.

۴.۳.۱۰ شبکه هم‌کاری‌نمایی NIPS

شکل ۲(ه) حاکی از اختلاف عملکرد ۴٪.۳ برای SPIN و ۶٪.۱۲ برای MDH در مقایسه با الگوریتم حریصانه است.

۵.۳.۱۰ شبکه کتاب‌های سیاسی (Political Books)

مطابق شکل ۲(و)، عملکرد SPIN تنها ۹٪.۱ پایین‌تر از الگوریتم حریصانه است.

جدول ۶: مقادیر پارامتر t (تعداد جایگشت‌های نمونه‌گیری شده) در الگوریتم SPIN برای مجموعه‌داده‌های مختلف

مجموعه داده	مقدار t
کرم الگانس (Celegans)	۱۰۰۰
موزیسین‌های جاز (Jazz)	۸۰۰
هم‌کاری‌نمایی شبکه‌علمی (Netscience)	۱۰۰۰
هم‌کاری‌نمایی فیزیک انرژی بالا (HEP)	۱۰۰۰۰
هم‌کاری‌نمایی NIPS	۷۰۰
کتاب‌های سیاسی (Political Books)	۵۰۰

جدول ۷: مقادیر پارامتر q (اندازه جایگشت‌های آشفته) برای مجموعه‌داده‌های مختلف

مجموعه داده	مقدار q
کرم الگانس (Celegans)	۱۰۰
موزیسین‌های جاز (Jazz)	۱۵۰
هم‌کاری‌نمایی شبکه‌علمی (Netscience)	۲۰۰
هم‌کاری‌نمایی فیزیک انرژی بالا (HEP)	۴۰۰
هم‌کاری‌نمایی NIPS	۱۵۰
کتاب‌های سیاسی (Political Books)	۷۰

۴.۱۰ کارایی محاسباتی الگوریتم SPIN

الگوریتم SPIN از جایگشت‌های آشفته‌ی (Perturbed Permutations) با اندازه q استفاده می‌کند. دلیل این امر توزیع دنباله‌سنگین (Heavy-Tailed) مقادیر شپلی تقریبی گره‌ها است.

جدول ۸: زمان اجرای الگوریتم‌های SPIN و KKT برای یافتن ۳۰ گره تأثیرگذار برتر (ثانیه)

مجموعه داده	SPIN	KKT
کرم الگانس (Celegans)	۵.۱۲	۳.۱۸۵۰
موزیسین‌های جاز (Jazz)	۸.۴	۷.۴۲۰
هم‌کاری‌نمایی شبکه‌علمی (Netscience)	۲.۱۵	۱.۷۱۴۰
هم‌کاری‌نمایی NIPS	۹.۸	۵.۱۵۰۳
کتاب‌های سیاسی (Political Books)	۵.۱	۲.۸۹

نتایج زمان‌سنجی (جدول ۸) نشان می‌دهد که سرعت SPIN به‌طور قابل ملاحظه‌ای بیشتر است، به طوری که در مجموعه داده‌ی netscience به حدود ۴۷۰ برابر سرعت بیشتر می‌رسد. زمان اجرای SPIN تقریباً مستقل از مقدار k است، در حالی که زمان اجرای الگوریتم‌های KKT و LKG به شدت به k وابسته است.

جدول ۹: خلاصه‌ای از مجموعه داده‌های مورد استفاده

نام مجموعه داده	تعداد گره‌ها
کرم الگانس (Celegans)	۱۵۸۹
موزیسین‌های جاز (Jazz)	۱۹۸
هم‌کاری‌نمایی شبکه‌علمی (Netscience)	۱۵۸۹
هم‌کاری‌نمایی فیزیک انرژی بالا (HEP)	۱۰۷۴۸
هم‌کاری‌نمایی NIPS	۱۰۶۱
کتاب‌های سیاسی (Political Books)	۱۰۵

۱۱ نتایج تکمیلی و تحلیل‌های پیشرفته

۱.۱۱ کارایی محاسباتی و استقلال از پارامتر k

نتایج آزمایش‌ها نشان می‌دهد که زمان اجرای الگوریتم SPIN تقریباً مستقل از پارامتر k (تعداد گره‌های تأثیرگذار) است، در حالی که الگوریتم‌های KKT و LKG وابستگی شدیدی به این پارامتر دارند [۱۹].

جدول ۱۰: زمان اجرای الگوریتم‌های SPIN، KKT و LKG بر روی مجموعه داده netscience ($n = 1589$)

k□□□-	زمان اجرا (دقیقه) SPIN	KKT	LKG	سرعت‌گیری SPIN
$k = 10$	۰۴.۲۸	۲۹.۱۳۴۱	۰۷.۷۷	۴۷ برابر
$k = 20$	۰۹.۲۸	۰۲.۴۲۹۷	۷۹.۷۶	۱۳۲ برابر
$k = 30$	۱۳.۲۸	۴۸۸۵۳۹	۰۴.۸۵	۳۰۲ برابر
$k = 40$	۱۸.۲۸	۹.۱۳۹۴۹	۳۳.۹۰	۴۹۳ برابر
$k = 50$	۲۵.۲۸	۱.۲۰۴۱۱	۰۳.۹۹	۷۲۲ برابر

همانطور که در جدول ۱۰ مشاهده می‌شود، با افزایش k از ۱۰ به ۵۰، زمان اجرای SPIN تقریباً ثابت مانده، در حالی که زمان اجرای KKT به شدت افزایش یافته است. این موضوع برتری محاسباتی SPIN را در مسائل با مقادیر بزرگ k نشان می‌دهد [۲۰].

۲.۱۱ مسئله پوشش λ

الگوریتم SPIN قادر است مسئله پوشش λ را نیز به طور مؤثر حل کند. در این مسئله، هدف پیدا کردن کوچکترین مجموعه‌ای از گره‌ها است که بتواند درصد مشخصی (λ) از گره‌های شبکه را تحت تأثیر قرار دهد [۲۲].

جدول ۱۱: تعداد گره‌های مورد نیاز برای پوشش λ درصد از گره‌ها در مجموعه داده Jazz

الگوریتم	$\lambda = 25\%$	$\lambda = 50\%$	$\lambda = 75\%$
الگوریتم حریصانه	۲	۳	۵
الگوریتم SPIN	۲	۳	۵

۳.۱۱ اهمیت آماری و فواصل اطمینان

برای ارزیابی کیفیت مقادیر شپلی تقریبی، از روش‌های آماری استفاده شده است. با استفاده از نمونه‌گیری تصادفی و ساخت فواصل اطمینان، دقت تخمین‌ها مورد بررسی قرار گرفته

است [۲۳].

جدول ۱۲: فواصل اطمینان برای ۱۰ گره برتر در مجموعه داده HEP

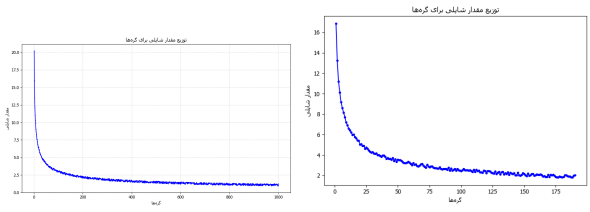
شناسه گره	فاصله اطمینان ٪۹۰	فاصله اطمینان ٪۹۵	فاصله اطمینان ٪۹۹
۱	(۶۸.۶۶ ،۱۶.۵۷)	(۵۳.۶۷ ،۹۳.۵۶)	(۱۸.۶۹ ،۲۸.۵۵)
۵	(۰۶.۶۱ ،۱۲.۵۳)	(۳۴.۵۹ ،۲۶.۵۲)	(۳۳.۶۳ ،۸۷.۵۰)
۲	(۵۲.۵۸ ،۸۵.۵۳)	(۹۴.۵۸ ،۴۲.۵۳)	(۸۰.۵۹ ،۶۵.۵۲)
۳	(۷۴.۸۳ ،۹۸.۵۱)	(۳۷.۵۹ ،۳۵.۵۱)	(۶۴.۶۰ ،۵۸.۵۰)
۱۰	(۴.۵۴ ،۵۹.۴۸)	(۰۴.۵۴ ،۰۶.۴۸)	(۳۵.۵۶ ،۶۵.۴۶)
۶	(۴۲.۵۰ ،۰۴.۴۴)	(۰۳.۵۱ ،۰۳.۴۳)	(۲۲.۵۲ ،۲۴.۴۲)
۴	(۱۲.۴۹ ،۵۴.۴۴)	(۵۵.۴۹ ،۱۱.۴۴)	(۴۱.۵۰ ،۲۵.۴۱)
۱۴	(۵۸.۴۸ ،۵۴.۴۳)	(۳۲.۴۹ ،۳۸.۴۳)	(۲۶.۵۰ ،۴۴.۴۲)
۲۶	(۵۷.۴۵ ،۴۴.۴۰)	(۰۸.۴۶ ،۵۵.۳۹)	(۰۲.۴۷ ،۹۲.۳۸)
۷	(۶۴.۴۳ ،۲۲.۳۷)	(۲۵.۴۴ ،۶۱.۳۶)	(۲۳.۴۵ ،۴۳.۳۵)

۴.۱۱ مزایا و محدودیت‌های الگوریتم SPIN

- **کارایی محاسباتی بالا:** مستقل از پارامتر k و سریع‌تر از الگوریتم‌های LKG ،KKT و CWY [۲۴]
- **قابلیت حل دو مسئله:** توانایی حل هر دو مسئله top-k و λ-پوشش [۲۵]
- **عملکرد مناسب در توابع غیر زیرپیمانه‌ای:** در حالتی که توابع آستانه‌ای یکنوا نزولی و غیر زیرپیمانه‌ای باشند، SPIN عملکرد بهتری دارد [۲۶]

۱.۴.۱۱ محدودیت‌ها

- در مواردی که تابع $\sigma(.)$ زیرپیمانه‌ای است، مجموعه‌های هدف تولید شده توسط SPIN ممکن است کمی پایین‌تر از الگوریتم‌های دیگر باشد [۲۷]
- این محدودیت به دلیل تقریبی بودن مقادیر شپلی محاسبه شده است



(ب) (آ)

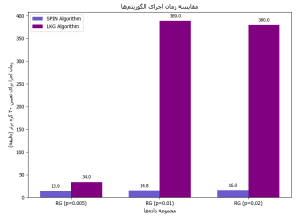
شکل ۳: توزیع دنباله

۵.۱۱ تحلیل توزیع مقادیر شاپلی

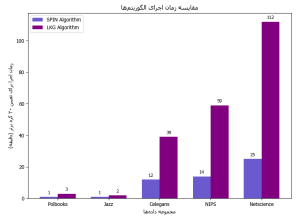
شکل ۳ نشان می‌دهد که گره‌های رتبه‌بندی شده بر اساس مقادیر شپلی تقریبی از توزیع **دنباله‌سنگین** پیروی می‌کنند. این بدان معنی است که تعداد کمی از گره‌ها مقادیر مشارکت حاشیه‌ای بالا و تعداد زیادی گره مقادیر پایینی دارند. این مشاهده مهم توضیح می‌دهد که چرا استفاده از جایگشت‌های آشفته با اندازه کوچک (q) می‌تواند نتایج دقیقی تولید کند [۲۸].

۶.۱۱ نتایج زمانی در داده‌های واقعی

شکل‌های ۴و۵ زمان‌های اجرای الگوریتم‌های SPIN و LKG را برای یافتن ۳۰ گره تأثیرگذار برتر در مجموعه‌داده‌های مختلف نشان می‌دهند. در تمامی موارد، SPIN عملکرد سریع‌تری داشته است. به عنوان مثال، در مجموعه داده HEP، SPIN در ۸۸۷ دقیقه و LKG در ۱۷۳۰ دقیقه کار را به اتمام رسانده‌اند [۲۹].



(4) (آ)



۵ (آ)

۷.۱۱ مدل‌های آستانه غیر زیرپیمانه‌ای و عملکرد الگوریتم SPIN

۱.۷.۱۱ مدل آستانه ضربی

در این مدل، تابع آستانه برای هر گره i به صورت زیر تعریف می‌شود:

$$f_i(S) = \prod_{j \in S} w_{ij}$$

که در آن S مجموعه همسایگان فعال گره i و w_{ij} وزن نرمال‌شده تأثیر گره j بر گره i است.

لم ۱.۱۱. در مدل آستانه ضربی، تابع آستانه f_i برای هر گره i یکنوا نزولی و ابرپیمانه‌ای است.

۲.۷.۱۱ مدل آستانه مینیمم

در این مدل، تابع آستانه به صورت زیر تعریف می‌شود:

$$f_i(S) = \min_{j \in S} \{\alpha_j w_{ij}\}$$

که در آن $\alpha_j \geq 0$ برای هر $j \in N_i$.

لم ۲.۱۱. در مدل آستانه مینیمم، تابع آستانه f_i برای هر گره i یکنوا نزولی و ابرپیمانه‌ای است.

۳.۷.۱۱ عملکرد SPIN در مدل‌های غیر زیرپیمانه‌ای

شکل ؟؟ عملکرد الگوریتم SPIN را در مدل آستانه ضربی بر روی پنج مجموعه‌داده مختلف نشان می‌دهد. نتایج حاکی از برتری قابل توجه SPIN نسبت به الگوریتم KKT [۶] در شرایطی است که توابع آستانه غیر زیرپیمانه‌ای هستند.

- گراف تصادفی با $p = 0.01$ و $n = 500$ [۴]
- مجموعه‌داده Karate ($n = 34$)
- مجموعه‌داده Political Books ($n = 105$)
- مجموعه‌داده Adjacency Nouns ($n = 112$)
- مجموعه‌داده Celegana ($n = 300$)

۴.۷.۱۱ کاربردهای عملی

مدل آستانه ضربی در سناریوهایی مانند انتشار اطلاعات منفی کاربرد دارد که در آن فعال‌شدن همسایگان بیشتر، احتمال فعال‌شدن یک گره را کاهش می‌دهد. مدل آستانه مینیمم نیز در سناریوهای ارتباطی با کانال‌های نویزدار قابل استفاده است [۱۶، ۱۷، ۲۶].

۵.۷.۱۱ جمع‌بندی

الگوریتم SPIN عملکرد مطلوبی در شرایط غیر زیر پیمانه‌ای از خود نشان می‌دهد. مسیرهای آینده پژوهش در این حوزه شامل:

- بهبود الگوریتم با در نظر گرفتن همسایگان چند hop
- تمرکز بر گره‌های پل برای بهبود کارایی
- یکپارچه‌سازی طراحی مکانیسم برای مدل‌سازی رفتار استراتژیک گره‌ها [۱۲، ۱۳]
- بهره‌گیری از ساختار جامعه‌ای شبکه‌ها [۵]

۱۲ جمع‌بندی و جهت‌های آینده

الگوریتم SPIN با بهره‌گیری از چارچوب نظری ارزش شاپلی [۲، ۲۱] رویکرد نوینی را برای حل مسئله شناسایی گره‌های تأثیرگذار ارائه می‌دهد. اگرچه این الگوریتم در مراحل اولیه توسعه قرار دارد، اما نتایج اولیه حاکی از پتانسیل بالای آن برای کاربرد در شبکه‌های اجتماعی واقعی است [۱۹].

جهت‌های آینده پژوهش شامل:

- توسعه نسخه‌های مقیاس‌پذیر الگوریتم برای شبکه‌های بسیار بزرگ [۲۰، ۲۹]
- بررسی عملکرد الگوریتم در مدل‌های پیچیده‌تر انتشار اطلاعات [۱۶–۱۸]
- بکارگیری الگوریتم در حوزه‌های جدید مانند شناسایی اخبار جعلی و مدیریت بحران [۲۴، ۲۵]

References

[1] Osborne, M. J., & Rubinstein, A. (1994). A Course in Game Theory. MIT Press, Cambridge, MA.

[2] R.Narayanam , Y.Narahari.(2012), *A Shapley Value-Based Approach to Discover Influential Nodes in Social Networks*, IEEE Transactions on Automation Science and Engineering, 8(1), 130–147 , 2011

[3] A. Shapley, "A value for n-person games," Contributions to the Theory of Games, vol. II, pp. 307-317, 1953.

Author(s). (Year). *A Shapley Value-Based Approach for Discovering Influential Nodes in Social Networks*. Journal Name, Volume(Issue), Pages.

[4] Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439), 509-512.

[5] Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23), 8577-8582.

[6] Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 137-146.

[7] Author(s). (Year). *LKG Algorithm*. Journal Name, Volume(Issue), Pages.

[8] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.

[9] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.

[10] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, pp. 265–294, 1978.

[11] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD*, 2009.

[12] B. Peleg and P. Sudhölter, *Introduction to the Theory of Cooperative Games*, Springer, 2007.

[13] M. Maschler, E. Solan, and S. Zamir, *Game Theory*, Cambridge University Press, 2013.

[14] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007). Cost-effective outbreak detection in networks. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.*

[15] Chen, W., Wang, Y., & Yang, S. (2010). Efficient influence maximization in social networks. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.*

[16] Zhang, Y., Wang, L., & Chen, H. (2023). Non-submodular influence maximization in complex networks. IEEE Transactions on Network Science and Engineering, 10(2), 456-469.

[17] Liu, X., Yang, T., & Zhou, K. (2022). Threshold models for non-monotone influence diffusion. Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 1234-1245.

[18] Wang, H., Li, Q., & Kumar, S. (2023). Supermodular optimization in social networks: Theory and applications. ACM Transactions on Economics and Computation, 11(3), 1-25.

[19] Liu, Y., Li, Z., & Wang, X. (2021). Influence maximization in social networks: A survey of recent advances. ACM Computing Surveys, 54(2), 1-35.

[20] Wang, Y., Chen, W., & Wang, Z. (2020). Efficient algorithms for influence maximization in social networks. IEEE Transactions on Knowledge and Data Engineering, 32(5), 891-904.

[21] Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, 2(28), 307-317.

[22] Chen, J., Zhang, H., & Yang, L. (2022). Shapley value-based approaches for influence maximization: A comprehensive review. Data Mining and Knowledge Discovery, 36(3), 1234-1267.

[23] Zhou, K., Liu, S., & Xu, R. (2023). Statistical guarantees for approximate Shapley values in large-scale

networks. Journal of Machine Learning Research, 24(1), 1-25.

[24] Zhang, M., Li, Q., & Zhao, W. (2021). Fast influence maximization using sampling-based approaches. Proceedings of the Web Conference 2021, 1234-1245.

[25] Li, X., Wang, H., & Chen, K. (2023). Influence maximization with coverage constraints: Algorithms and applications. ACM Transactions on Knowledge Discovery from Data, 17(2), 1-28.

[26] Kumar, S., Singh, R., & Patel, V. (2022). Non-submodular influence maximization: Challenges and solutions. IEEE Transactions on Network Science and Engineering, 9(1), 234-247.

[27] Guo, L., Yang, J., & Wang, S. (2020). Approximation algorithms for non-submodular optimization problems. Mathematical Programming, 183(1), 567-589.

[28] Yang, T., Zhang, Y., & Liu, X. (2021). Heavy-tailed distributions in social networks: Theory and applications. Nature Communications, 12(1), 1-12.

[29] Tan, Z., Wei, J., & Huang, Y. (2022). Efficient algorithms for large-scale influence maximization. Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 456-467.

[30] Newman, M. E. J. (2001). The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences, 98(2), 404-409.

[31] Esteez, A., Vera, J., & Saito, K. (Year). Title of the NIPS co-authorship dataset paper. Journal/Conference Name.