

یک روش جدید حریصانه مبتنی بر مدل آبخاری برای محاسبه‌ی حداکثر سازی نفوذ در شبکه‌های اجتماعی

* عسگرعلی بویر ** حمید احمدی بنی

* دانشیار، گروه مهندسی کامپیوتر، دانشکده فناوری اطلاعات و مهندسی کامپیوتر، دانشگاه شهید مدنی آذربایجان، تبریز
** کارشناسی ارشد، گروه مهندسی کامپیوتر، دانشکده فناوری اطلاعات و مهندسی کامپیوتر، دانشگاه شهید مدنی آذربایجان، تبریز
تاریخ دریافت: ۱۳۹۸/۰۴/۲۶ تاریخ پذیرش: ۱۳۹۹/۰۲/۱۸

چکیده

در مسئله حداکثر سازی نفوذ، هدف یافتن حداقل تعدادی گره هست که بیشترین انتشار و نفوذ را در شبکه داشته باشند. مطالعات راجع به حداکثر سازی نفوذ و انتشار به صورت گسترده‌ای در حال گسترش است. در سال‌های اخیر الگوریتم‌های زیادی در زمینه مسئله حداکثر سازی نفوذ در شبکه‌های اجتماعی ارائه شده است. این مطالعات شامل بازاریابی ویروسی، گسترش شایعات، اتخاذ نوآوری و شیوع بیماری‌های همه‌گیر و ... است. هر یک از مطالعات پیشین دارای کاستی‌هایی در یافتن گره‌های مناسب و یا پیچیدگی زمانی بالا هستند. در این مقاله، روشی جدید با عنوان ICIM-GREEDY برای حل مسئله حداکثر سازی نفوذ ارائه کرده‌ایم. در الگوریتم ICIM-GREEDY دو معیار مهم که در کارهای انجام‌شده قبلی در نظر گرفته نشده‌اند را در نظر می‌گیریم، یکی قدرت نفوذ و دیگری حساسیت به نفوذ. این دو معیار همیشه در زندگی اجتماعی انسان‌ها وجود دارد. روش پیشنهادی روی دیتاست‌های استاندارد مورد ارزیابی قرار گرفته شده است. نتایج به دست آمده نشان می‌دهد که روش مذکور نسبت به دیگر الگوریتم‌های مقایسه شده از کیفیت بهتری در پیدا کردن نودهای بانفوذ در ۳۰ گره Seed برخوردار است. همچنین این روش از لحاظ زمانی نیز نسبت به الگوریتم‌های مقایسه شده به لحاظ همگرایی نسبتاً سریع، بهتر عمل می‌کند.

واژه‌های کلیدی: مدل آبخاری مستقل، حداکثر سازی نفوذ، انتشار، شبکه اجتماعی

۱- مقدمه

مدل پایه‌ای آبخاری مستقل و آستانه‌ی خطی معرفی شده است. انتشار اطلاعات و نفوذ در زمان و مراحل گسسته انجام می‌شود. هر نود V در گراف G در دو حالت فعال یا غیرفعال است. در حالت فعال یک نود تحت نفوذ قرار گرفته است یعنی یک شایعه یا یک ایده یا یک محصول جدید را پذیرفته است. در حالت غیرفعال ایده یا محصول جدید را نپذیرفته است که ناشی از نپذیرفتن یا نرسیدن ایده یا محصول جدید یا یک شایعه باشد. برای درک بهتر مدل‌های انتشار اغلب از مدل‌های هم‌ارزی استفاده می‌شود. مدل‌های تصادفی را هم-

مطالعه گسترش نفوذ در شبکه‌های اجتماعی و شبکه‌های پیچیده به صورت قابل توجهی افزایش پیدا کرده است. حداکثر سازی نفوذ برای اولین بار توسط دومنیکس و ریچاردسون در سال ۲۰۰۱ مطرح شد [۱]. کمپ و همکارانش برای اولین بار به مسئله حداکثر سازی نفوذ پرداختند [۲]. در شبکه‌های اجتماعی و البته مسئله حداکثر سازی نفوذ به دنبال نودهایی با ضریب نفوذ بالا هستیم که بیشترین تأثیر را بر روی شبکه داشته باشند. برای حداکثر سازی نفوذ دو

استاد دانشگاه نسبت به دانشجو دارای درجه پایین‌تری به دلیل برقراری رابطه با افرادی خاص است. ولی نودهایی که با استاد دانشگاه در ارتباط هستند دارای قدرت نفوذ بیشتری هستند. فرض کنید هر دو نود استاد دانشگاه و دانشجو ۵۰ نود را فعال می‌کنند ولی آیا قدرت نفوذ هر دو یکسان است، مسلماً این‌طور نیست. حساسیت به نفوذ، در نودها می‌تواند متفاوت باشد که یک نود می‌تواند در مقابل نفوذ (اتخاذ محصول جدید یا ایده) مقابله کند یا به راحتی آن را بپذیرد. در این مقاله برای محاسبه قدرت نفوذ می‌توانیم از دو فاکتور مهم نودهای حاشیه و نودهای هسته استفاده کنیم. به این صورت نودهایی که در هسته قرار دارند مسلماً قدرت نفوذ بیشتری دارند و نودهایی که در حاشیه قرار دارند قدرت نفوذ کمتری دارند.

ادامه مقاله به این صورت سازمان‌دهی شده است. در بخش ۲ یک دسته‌بندی کلی از الگوریتم‌های ارائه‌شده در زمینه حداکثر سازی نفوذ مورد بررسی و ارزیابی قرار گرفته است. در بخش ۳ الگوریتم ICIM-GREEDY با استفاده از فاکتورهای جدید حساسیت به نفوذ و قدرت نفوذ ارائه‌شده است. تحلیل نتایج در بخش ۴ صورت گرفته است و در بخش ۵ به نتیجه‌گیری پرداخته‌ایم.

۲- کارهای انجام شده

حداکثر سازی نفوذ، به‌عنوان یک روش الگوریتمی برای بازاریابی و ویروسی برای اولین بار توسط دومینگوس و ریچاردسون در سال ۲۰۰۱ پیشنهاد شده است [۱]. کمپ و همکارانش در سال ۲۰۰۳ برای اولین بار به تدوین و فرموله سازی مسئله‌ی حداکثر سازی نفوذ به‌عنوان یک مسئله بهینه‌سازی تصادفی گسسته پرداختند [۲]. مسئله حداکثر سازی نفوذ را می‌توان طبق رابطه‌ی (۱) مطرح کرد که در یک گراف $G=(V,E)$ ، یک مدل تصادفی بر روی G ، مجموعه‌ی نودهای اولیه فعال را با $|S| < k$ نشان می‌دهد که هدف آن بیشینه‌سازی $\sigma_G(S^*)$ است که به‌صورت زیر محاسبه می‌شود:

$$(1) \quad \sigma_G(S) = \sum_{v \in V} p_v \cdot \mathbb{1}_{\{v \text{ is active}\}}$$

وانگ و همکارانش در سال ۲۰۱۰ اثبات کردند که محاسبه گسترش نفوذ $\sigma_G(S)$ در یک گراف اجتماعی

ارز می‌گویند که دارای توزیع احتمال برای نودهای فعال شده توسط نودهای S_0 یکسان باشد. مدل‌های LT و IC دل‌های پایه‌ای در حداکثر سازی نفوذ است [۲]. هرچند در سال‌های اخیر مدل‌های دیگر نیز در این زمینه ارائه‌شده است، ولی پایه و اساس مدل‌های ارائه‌شده، مدل LT و IC است [۳-۹].

مدل آبخاری مستقل برای اولین بار توسط کمپ و همکارانش مطرح شد [۲]. در مدل آبخاری مستقل هرگاه نودی فعال می‌شود با احتمالی می‌تواند نودهای همسایه خود را فعال کند این احتمال برابر با وزن یال ارتباطی u و v یعنی p_{uv} است. فعال سازی تنها یک بار صورت می‌گیرد به این معنی که اگر نود v ن انست نود u را فعال کند، دیگر شانس برای فعال سازی نود u ن رد. در مدل آستانه خطی احتمال اثرگذاری نود v ب نود u با وزن بین یال‌ها، که با $b_{v,w}$ نشان می‌دهیم، بیان می‌کنیم. هر نود v دارای یک حد آستانه θ_v است که به‌صورت تصادفی در بازه $[0,1]$ به‌صورت تصادفی انتخاب می‌شود. حداکثر سازی نفوذ برای هر دو مدل آبخاری مستقل و حد آستانه خطی یک مسئله NP -hard است [۱۰]. در مدل بخاری مستقل و آستانه خطی (LT) تابع خروجی $\sigma(0)$ ، خواص زیر بخشی و یکنوایی دارد که $\sigma(0)$ تابع گسترش نفوذ است. الگوریتم حریصانه یک رویکرد طبیعی برای انتخاب مجموعه‌ای از گره‌ها است که حداکثر نفوذ را داشته باشند. الگوریتم حریصانه می‌تواند ضریب تقریبی $1 - \frac{1}{n} - \epsilon$ ارائه دهد که n تعداد نودهای گراف است و ϵ عددی بسیار ناچیز است [۲].

در این مقاله یک الگوریتم جدید به نام ICIM-GREEDY را ارائه می‌دهیم. در مقالات ارائه‌شده در سال‌های اخیر دو نکته بسیار مهم را در نظر گرفته نشده است [۱۱-۱۷]، در صورتی که در دنیای واقعی این دو معیار وجود دارد. در دنیای واقعی نفوذ افراد با یکدیگر متفاوت است، افراد در جامعه با توجه به خصوصیات اجتماعی قدرت نفوذ متفاوتی دارند. در این مقاله نفوذ گره‌ها با توجه به دو معیار دنیای واقعی سنجیده می‌شود: قدرت نفوذ و حساسیت به نفوذ. قدرت نفوذ برای نودها متفاوت است به‌عنوان مثال فرض کنید یک دانشجو مسلماً دارای درجه بالایی است و

گراف مشخص می‌کند و نودهای هسته را به‌عنوان نود seed در نظر می‌گیرد [۲۱]. در ادامه، چانگ و همکارانش الگوریتم StaticGreedyDU را برای بهبود زمان اجرای الگوریتم NewGreedyIC مطرح کردند [۲۲]. اما الگوریتم StaticGreedyDU تقریب بهینه را تضمین نمی‌کند. در ادامه برای بالا بردن سرعت محاسبات گسترش نفوذ، گویال و همکارانش الگوریتم SIMPATH را مطرح کردند که با استفاده از محدود کردن شمارش مسیرهای ساده، نودهای seed را انتخاب می‌کند [۲۳]. الگوریتم Simpath زمان اجرای مناسبی برای انتخاب k نود seed دارد و همچنین سربار حافظه مصرفی کمی دارد ولی ضریب تقریب بهینه را تضمین نمی‌کند به همین دلیل الگوریتم MIA توسط وانگ و همکارانش مطرح شد که الگوریتمی مقیاس‌پذیر است و با استفاده از درخت arborescence گسترش نفوذ به‌صورت محلی محاسبه می‌شود [۱۰]. الگوریتم MIA به دلیل اینکه محاسبات گسترش نفوذ را بدون استفاده از شبیه‌سازی مونت‌کارلو انجام می‌دهد، زمان اجرای مناسبی دارد. ولی این الگوریتم سربار حافظه مصرفی بالایی دارد. سپس، چن و همکارانش الگوریتم CIM که مبتنی بر تشخیص جامعه می‌باشد را مطرح کردند که الگوریتم CIM ابتدا تشخیص جامعه انجام می‌شود و پس از آن نودهای کاندید تولید می‌شوند و از نودهای کاندید نودهای seed انتخاب می‌شوند [۲۴]. هرچند الگوریتم CIM زمان اجرای مناسبی دارد ولی تقریب بهینه را تضمین نمی‌کند. چن و همکارانش الگوریتم LUGreedy را مطرح کردند که در این الگوریتم با توجه به تصادفی بودن احتمال فعال‌سازی، عدم قطعیت را برای مسئله حداکثر سازی نفوذ در نظر می‌گیرد [۲۵]. ژیانگ و همکارانش، الگوریتم VoteRank را که مبتنی بر رأی‌دهی را مطرح کردند [۲۶]. این الگوریتم زمان اجرای مناسبی دارد. همچنین به دلیل اینکه این الگوریتم خاصیت سابماژولاریتی ندارد، تقریب بهینه را تضمین نمی‌کند. سپس، شانگ و همکارانش الگوریتم CoFIM را مطرح کردند که مبتنی بر تشخیص جامعه است [۲۷]. زمان اجرای الگوریتم CoFIM به تعداد نودهای seed وابسته است. سپس، مورنه و همکاران،

$G=(V,E)$ با مجموعه S_0 ، برای هر دو مدل LT و IC، NP-hard است [۱۰]. مسئله حداکثر سازی نفوذ یک مسئله سخت است. روش اصلی توسط کمپ و همکارانش ارائه شده است که در این روش از فرایند شبیه‌سازی مونت‌کارلو در فرایند انتشار برای محاسبه گسترش نفوذ استفاده می‌شود [۲]. مجموعه تأثیرگذار S برای R تکرار شبیه‌سازی می‌شود. بعد از پایان فرایند انتشار تعداد نودهای فعال شمرده می‌شود و سپس متوسط آن‌ها را در R تکرار محاسبه می‌شود. الگوریتم حریصانه با تعداد زیاد تکرار شبیه‌سازی مونت‌کارلو باعث دستیابی به بهترین نود نفوذپذیر در میان الگوریتم‌های مختلف شود. یک مشکل جدی برای الگوریتم حریصانه مونت‌کارلو MC-Greedy(G,K) ناکارآمدی وقت‌گیر بودن آن است. به همین دلیل لسکوک و همکارانش الگوریتم CELF را مطرح کردند که با استفاده از ارزیابی تنبل، محاسبات گسترش نفوذ را کاهش می‌دهد [۱۸]. ولی همچنان الگوریتم‌های CELF به دلیل استفاده از شبیه‌سازی مونت‌کارلو در تعداد تکرار بالا، ناکارآمد بودند. بدین ترتیب چن و همکارانش الگوریتم NewGreedyIC را مطرح کردند که با استفاده از مجموعه نودهای قابل‌دسترسی، گسترش نفوذ را برای هر نود محاسبه می‌کند. همچنین در ادامه، چن و همکاران برای بهبود زمان اجرای NewGreedyIC، الگوریتم DegreeDiscount را مطرح کردند که بعد از انتخاب نودی با بالاترین درجه به‌عنوان اولین نود Seed، برای انتخاب نودهای seed بعدی، درجه‌ی نودهایی که در همسایگی آن‌ها، seed وجود دارد، را کاهش می‌دهد [۱۹]. این الگوریتم تقریب بهینه را برای مسئله حداکثر سازی نفوذ تضمین نمی‌کند ولی زمان اجرای بسیار مناسبی به نسبت الگوریتم‌های Greedy، CELF و NewGreedyIC دارد. سپس، نارایانام و همکارانش الگوریتم SPIN را ارائه دادند که با استفاده از مقدار شیپلی، گسترش نفوذ را برای هر نود محاسبه می‌کند [۲۰]. این الگوریتم زمان اجرای مناسبی دارد ولی تقریب بهینه را در مقایسه با الگوریتم Greedy تضمین نمی‌کند. سپس، کیتساک و همکاران، الگوریتم K-core را مطرح کردند که نودهای هسته و حاشیه را در

ادامه، کیو همکارانش، الگوریتم PHG را مطرح کردند که در این الگوریتم از رویکرد مبتنی بر تشخیص جامعه استفاده می‌کند. همچنین در این الگوریتم برای محاسبات گسترش نفوذ، از الگوریتم حریمانه ویژگی گراف استفاده می‌کند [۳۶]. همچنین در این الگوریتم، میزان گسترش نفوذ و زمان اجرا، به تعداد نودهای تأثیرگذار که توسط الگوریتم حریمانه انتخاب می‌شوند، وابسته است. پس از آن، احمدی بنی و همکارانش، الگوریتم TI-SC را مطرح کردند که در این الگوریتم با استفاده از نودهای هسته در جوامع الگوریتم PHG را بهبود می‌بخشد [۱۱].

۳- روش پیشنهادی

در این مقاله الگوریتم جدیدی با نام ICIM-GREEDY ارائه شده است. در الگوریتم ICIM-GREEDY از مدل WC بهبودیافته استفاده شده است. در این الگوریتم ابتدا وزن نفوذ را محاسبه می‌شود و سپس محاسبات نفوذپذیری را بر اساس دو فاکتور مهم، قدرت نفوذ و حساسیت به نفوذ انجام می‌شود.

۳-۱- وزن نفوذ

وزن نفوذ، یک فاکتور مهم برای تشخیص نفوذ است، به همین دلیل از یک روش جدید برای محاسبه وزن نفوذ استفاده شده است. در مدل WC، وزن نفوذ، $\frac{1}{degree(v)}$ محاسبه می‌شود. اما این یک روش منطقی برای تعیین وزن نفوذ نیست به دلیل اینکه نودها دارای ویژگی‌های متفاوتی مثل ضریب خوشه‌بندی متفاوت، درجه همسایه‌های یک نود، پل محلی بودن یا در مجاورت پل محلی بودن و .. هستند، پس باید بین وزن‌های نفوذ در بین نودهای مختلف، تفاوت قائل شویم. بنابراین وزن نفوذ طبق فرمول زیر محاسبه می‌شود:

$$(۲) \quad I$$

D_u درجه نود u است و D_{II} درجه نودهای همسایه نود u است و CC_{II} ضریب خوشه‌بندی نود u است. وزن نفوذ، به این صورت تعیین می‌شود هر چه درجه همسایگان و درجه همسایگان همسایگان نود v بیشتر باشد، نودهای

الگوریتم CI که مبتنی بر محلی سازی محاسبات گسترش نفوذ بود را مطرح کردند [۲۸]. در این الگوریتم محاسبات گسترش نفوذ در دایره‌ای به شعاع L محدود می‌شود. زمان اجرای الگوریتم CI به L و تعداد نودهای seed، وابسته است. همچنین، لئو همکارانش الگوریتم LIR که مبتنی بر روش‌های اکتشافی بود را مطرح کردند [۲۹]. در الگوریتم LIR مقدار LI برای هر نود بر اساس درجه همسایگان محاسبه می‌شود و سپس، مجموعه نودهای با کمترین مقدار LI بر اساس درجه به صورت نزولی مرتب می‌شوند و k نود به عنوان seed انتخاب می‌شوند. این الگوریتم، زمان اجرای مناسبی دارد ولی تقریب بهینه را تضمین نمی‌کند. سپس، اهجام و همکارانش الگوریتم HybridRank را مطرح کردند که نودهای تأثیرگذار بر اساس مرکزیت بردار ویژه و coreness انتخاب شده‌اند [۳۰]. در این الگوریتم، در انتخاب نودهای تأثیرگذار از پدیده Rich club اجتناب می‌شود. اگرچه این الگوریتم از پدیده Rich club اجتناب می‌کند، اما نودهای تأثیرگذار گسترش نفوذ مناسبی ارائه نمی‌دهند. نگوین و همکارانش، الگوریتم probDeg را مطرح کردند که نودهای بانفوذ را با بررسی چند گام از نود و درجه همسایگان آن نود انتخاب می‌کند [۳۱]. سپس، وو و همکارانش، الگوریتم زمان خطی LAIM را مطرح کردند که این الگوریتم یک رویکرد تکراری خطی برای مسئله حداکثر تأثیرگذاری در شبکه‌های بزرگ است [۳۲]. همچنین، این الگوریتم سربار حافظه‌ی مصرفی پایینی، در شبکه‌های اجتماعی با مقیاس بزرگ دارد. سپس، بنرجی و همکارانش الگوریتم ComBIM را مطرح کردند که نودهای تأثیرگذار را با توجه به بودجه جامعه، انتخاب می‌کند [۳۳]. همچنین، این الگوریتم تقریب بهینه را تضمین نمی‌کند. در ادامه، ژی و همکارانش، الگوریتم IRR را بر مدل MBIC مطرح کردند [۳۴]. در این الگوریتم انتشار شامل دو مرحله‌ی شروع و مرجع است. این الگوریتم، گسترش نفوذ بهتری به نسبت الگوریتم DegreeDiscount دارد. روی و همکارانش، الگوریتم RNR را مطرح کردند [۳۵]. این الگوریتم، با استفاده از تأثیر همسایگان، قدرت نفوذ نود را محاسبه می‌کند. در

علاوه بر قدرت نفوذ، معیار حساسیت به نفوذ برای محاسبه نفوذ در نظر گرفته شده است. حساسیت به نفوذ مسئله‌ای بسیار مهم در حداکثر سازی نفوذ است زیرا در دنیای واقعی حساسیت به نفوذ متفاوت است ولی در تمامی کارهای قبلی حساسیت به نفوذ به صورت تصادفی تعیین شده است. برای نشان دادن اهمیت این موضوع، نگاهی به شکل ۱ داشته باشیم. همان‌طور که در این شکل می‌بینید، درجه نود ۱ و نود ۵، برابر است ولی اهمیت نود ۱ بیشتر از نود ۵ است زیرا گره‌های مهم‌تری به نود ۱ متصل هستند. ممکن است بر اساس آستانه‌ی نفوذ تصادفی، نود ۵ فعال شود که می‌تواند نودهای کمتری را در ادامه فعال کند در صورتی که با فعال شدن نود ۱ ممکن است نودهای بیشتری را نسبت به نود ۵ فعال کند. در کارهای قبلی آستانه‌ی نفوذ به صورت تصادفی تعیین می‌شود، ولی در این الگوریتم آستانه نفوذ، بر اساس اهمیت نفوذ نود و نودهای پل/پل محلی، تعداد نودهای فعال شده، طول مسیر از مبدأ و همسایگان فعال نشده تعیین می‌شود.

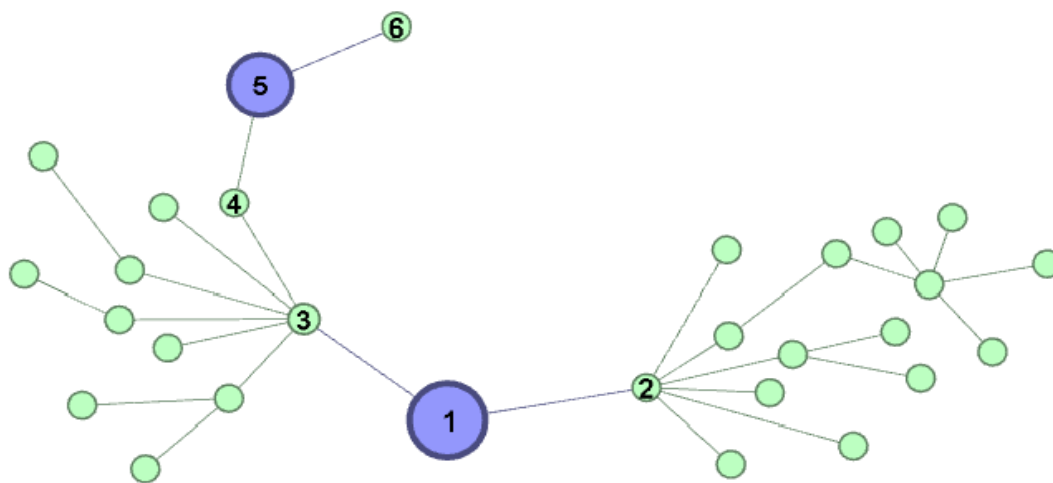
همسایه v می‌تواند نفوذ بیشتری بر روی v داشته باشند و همچنین هرچه همسایه‌های نود v دارای ضریب خوشه‌بندی کمتر باشند، تمایل بیشتری به برقراری ارتباط دارند. پس مجموع وزن نفوذ که یک گره u می‌تواند به همسایگان غیرفعال اعمال کند با استفاده از فرمول زیر محاسبه شده است:

$$(3) \quad z$$

$A(u)$ مجموعه‌ای از نودهای فعال در میان همسایگان u است. در دنیای واقعی انتشار و نفوذ تحت تأثیر تمامی همسایگان نود است و در صورتی نودی می‌تواند فعال شود که همسایگان فعال نود بتواند نود موردنظر را تحت تأثیر قرار دهند. به همین دلیل معیار $f(u)$ به صورت مجموع نفوذ همسایگان محاسبه می‌شود. بنابراین طبق رابطه‌ی زیر احتمال نفوذ به صورت زیر محاسبه می‌شود:

$$(4) \quad f$$

۲-۳- حساسیت به نفوذ



شکل ۱ - گره‌های ۱ و ۲ درجه یکسانی دارند ولی گره ۱ دارای اهمیت نفوذ بیشتری است.

سنجش میزان فرمانبرداری افراد از اقتدار^۱ در انجام کارهایی

اساس این ایده، آزمایش میلگرم است آزمایشی برای

یک نود فعال، بیشتر باشد حساسیت به نفوذ پایین‌تر است.

بنابراین برای محاسبه وزن نفوذ از رابطه‌ی زیر استفاده می‌کنیم:
 (۵)

در رابطه ۵، $NG_{inactive}$ تعداد نودهای همسایگان فعال نشده نود v است. NG_{active} تعداد نودهای فعال شده از همسایگان نود v است. NNG_{active} تعداد نودهای فعال همسایه‌ی نودهای فعال، v است. k_i درجه نودهای همسایگان نود v ، که فعال نشده‌اند، است. در واقع در این فرمول، اهمیت نودهای پل/پل محلی به نحو مناسبی در نظر گرفته شده است.

۳-۳- قدرت نفوذ

در رویکرد پیشنهادی، برای اینکه بین نفوذ نودهای مختلف تفاوتی وجود داشته باشد از مفهوم جدید توپولوژی شبکه که بر اساس تجزیه k -shell انجام خواهد شد، استفاده شده است. انتشار اطلاعات و الگوی رفتاری، وابسته به ساختار و توپولوژی شبکه است. به عنوان مثال، اگر یک hub در انتهای شبکه وجود داشته باشد، آن ممکن است حداقل گسترش را داشته باشد، در حالی که یک نود با کمترین اتصال در هسته شبکه، ممکن است گسترش قابل توجهی داشته باشد. در توپولوژی شبکه، موقعیت یک نود با استفاده از تجزیه K -Shell تعریف می‌شود. در k -shell ابتدا همه‌ی نودها با درجه $k=1$ حذف می‌شود، پس از هرس مرحله اول ممکن است گره‌هایی جدید با $k=1$ ظاهر شود. دوباره تمامی نودهای با درجه $K=1$ حذف می‌شوند. برای نودهای با درجه‌های متفاوت این روش ادامه می‌یابد. بنابراین نودهایی با درجه پایین در حاشیه و نودهایی با درجه بالا، در هسته قرار می‌گیرند. همان‌طور که در شکل ۲ قابل مشاهده است نود ۸ و ۹ درجه‌شان ۸ است ولی نود ۹ به دلیل اینکه در هسته قرار دارد، گسترش نفوذ بهتری نسبت به نود ۸ دارد. با توجه به نتایج آزمایش‌ها، زمانی که یک نود اولیه را به صورت تصادفی فعال می‌شود، نودهایی که در لایه‌هایی با k_s

مغایر با وجدان شخصی افراد بود که به ابتکار روانشناس معروف استنلی میلگرام صورت گرفت [۳۷]. آزمایش‌هایی که توسط میلگرام صورت گرفت نشان داد که حضور افراد دیگری که از فرمان‌ها پیروی نمی‌کردند، میزان فرمان‌برداری شرکت‌کننده را به نحو زیادی کاهش می‌دهد. وقتی کسانی دیگری در کنار شرکت‌کننده حضور داشتند که از اجرای فرمان‌ها سر باز می‌زدند، ۳۶ نفر از ۴۰ شرکت‌کننده از دادن ماکزیمم شوک خودداری کردند- [۳۷]. بر همین اساس، هرچه تعداد نودهای فعال شده با ایده یا محصول جدید، بیشتر باشند حساسیت به نفوذ در نود کمتر می‌شود و این باعث می‌شود ایده یا محصول جدید به راحتی پذیرفته شود.

برای محاسبه حساسیت به نفوذ فاکتورهای زیر بسیار مهم می‌باشند:

- تعداد نودهای فعال شده از همسایگان نود: هرچقدر تعداد نودهای فعال شده بیشتر باشند حساسیت به نفوذ پایین‌تر است.
- فاصله از نود مبدأ: هرچقدر فاصله نود از منبع شایعه دورتر باشد، حساسیت به نفوذ آن بیشتر می‌شود. (فاصله از نود تأثیرگذار)
- تعداد نودهای همسایگان فعال نشده از همسایگان نود: هرچقدر تعداد نودهای فعال نشده به نسبت تعداد نودهای فعال شده کمتر باشد، حساسیت به نفوذ نود کمتر می‌شود.
- تعداد نودهای فعال همسایه‌ی یک نود فعال: این فاکتور برای بیان اهمیت نودهای پل/پل محلی، مطرح شده است. زیرا این گره‌ها، نقش مهمی در فرایند انتشار اطلاعات دارند. هرچقدر تعداد نودهای فعال همسایه‌ی

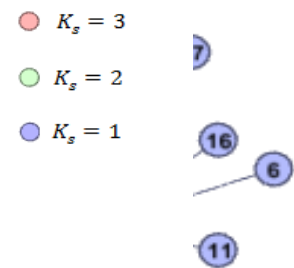
¹ authority

است. علاوه بر این، در ابتدای کار نفوذ همه نودها برابر با 1 در نظر گرفته شده است، به این معنی که اگر آن گره فعال شود، حداقل خودش را فعال کرده و یک گره فعال را نتیجه می‌دهد. مقادیر PID در یک بردار ذخیره می‌شود که قدرت نفوذ را برای هر نود مشخص می‌کند. البته برای محاسبه قدرت نفوذ می‌توان مشخصه‌های رفتاری و یا غیر ساختاری مانند سن، جنسیت و زمان تبلیغات و غیره نیز در نظر گرفت. ولی در نظر گرفتن این مشخصه‌ها در شبکه اجتماعی کمی مشکل است زیرا در اکثر مواقع به دلایلی مانند حفظ حریم خصوصی این مشخصه‌ها قابل دسترس نیستند.

بالتر قرار دارند، احتمال بیشتری دارند که فعال شوند. البته زمانی موقعیت نود مهم است که به‌عنوان یک نود منبع در نظر گرفته شود. بنابراین، نودهایی که در هسته-های داخلی (k_S بزرگ‌تری دارند) قرار دارند، نفوذ بیشتری دارند. به دلیل موارد فوق، دو فاکتور k و k_S برای محاسبه قدرت نفوذ در نظر گرفته شده است. برای هر مجموعه از نودهای تأثیرگذار S ، $\sigma(u|S) = \sigma(S \cup \{u\}) - \sigma(S)$ است. برای محاسبه قدرت نفوذ زمانی که $k > 1$ از فرمول زیر استفاده می‌شود:

$$l \quad (6)$$

در این رابطه، Count تعداد نودهای فعال شده است. تعداد نودها در هر k_S



شکل ۲- مثالی از روش k-Shell. گره ۸ و ۹ هر دو درجه‌شان ۸ است ولی گره ۹ به دلیل اینکه در هسته قرار دارد، می‌تواند گسترش نفوذ بیشتری داشته باشد.

Algorithm 1: ICIM-GREEDY(G, k): Improved Cascade model for Influence Maximization.

Input: G: Social graph, k: size seed set, R=1

1: **for** j=0 to R **do**

2: **if** (v is active.)

3: **for** 1 to H **do**

4: $\theta[e.v] = 1 - e^{-\left(\frac{\sum_{i \in NG_{inactive}, v \in A(u)} K_i}{NG_{active} \log NG_{active} * NNG_{active}}\right)}$

5: $p[e.v] = 1 - e^{-f(u)}$

6: **if** (($\theta[e.v] - p$) < 0) **then**

7: $n_a \leftarrow$ number of active node

8: count \leftarrow count + n_a

9: **end if**

10: **end for**

11: **end if**

12: **end for**

13: **return** $PID = \frac{k_s * count}{N(k_s)}$

این صورت است که برای هر یال $\bar{uv} \in E$ با احتمال $1 - p$ از G پاک می‌شود. $RanWC(G)$ نشان‌دهنده این فرایند است و $G' = RanWC(G)$ نتایج را بر روی گراف جهت‌دار نشان می‌دهد. در خط ۶-۱۱ در الگوریتم ۲ از الگوریتم کوهن برای تخمین تعداد نودهای قابل‌دسترس از هر نود استفاده شده است که در اولین گام در گراف G' تمام کامپوننت‌های قویاً متصل را محاسبه شده است. G^{*f} گراف جهت‌دار غیر مدور (DAG) است و V^* نشان‌دهنده مجموعه نودهای G^{*f} است.

الگوریتم ۱ تنها یک‌بار اجرا می‌شود تا بردار PID را مشخص کند. بردار PID قدرت نفوذ را برای هر نود ارائه می‌دهد. بردار PID به‌عنوان ورودی برای الگوریتم NewGreedyWC است. بنابراین پیچیدگی کلی الگوریتم NewGreedyWC برابر است با $O(kTn)$ بطوری که k تعداد نودهای seed است و m تعداد یال‌ها است. در واقع $T=5$ است زیرا برآورد خوبی از گسترش نفوذ را می‌دهد.

برای محاسبه بردار PID نیاز به محاسبه $N(k_s)$ و k_s است که این دو مقدار با استفاده از الگوریتم K-Shell محاسبه می‌شود [38]. پیچیدگی زمانی K-shell $O(m)$ است [38]. به دلیل اینکه به‌صورت پویا تعداد نودهای همسایه‌ی فعال، تعداد نودهای همسایه‌ی غیرفعال و تعداد نودهای همسایگان همسایگان نود فعال محاسبه می‌شود، پیچیدگی محاسبات $O(2n)$ است که

۳-۴- الگوریتم حریشانه

ما می‌توانیم به‌سادگی برای انتخاب مجموعه تأثیرگذار با اندازه k، الگوریتم ۲ را k بار فراخوانی کنیم. به دلیل اینکه الگوریتم حریشانه عمومی بسیار وقت‌گیر است، از الگوریتم NewGreedyWC استفاده شده است. در الگوریتم ICIM-GREEDY وزن نفوذ و حد آستانه به‌صورت تصادفی انتخاب نمی‌شوند به همین دلیل از شبیه‌سازی مونت‌کارلو استفاده نمی‌شود (به‌عبارتی دیگر $R=1$ است). به همین دلیل الگوریتم NewGreedyWC نسبت به الگوریتم حریشانه عمومی از لحاظ زمان اجرا بهتر می‌شود. در این الگوریتم باید توجه شود، نودهایی که گسترش نفوذ مناسبی ندارند، باعث ایجاد سربار محاسباتی می‌شوند. به همین دلیل برای کاهش سربار محاسباتی گسترش نفوذ باید نودهایی که گسترش نفوذ مناسبی ندارند در محاسبات گسترش نفوذ نادیده گرفته شوند. بدین ترتیب، برای محاسبات گسترش نفوذ، $\frac{m}{\sqrt{n}}$ نود با بالاترین درجه در نظر گرفته می‌شوند که m تعداد یال‌های گراف و n تعداد نودهای گراف است. در مدل WC گراف جهت‌دار در نظر گرفته می‌شود. اگر d_{uv} درجه v در گراف G و \bar{uv} یال در گراف باشد. در این مدل، اگر نود u فعال شده باشد، آنگاه نود v با احتمال $\frac{1}{d_v}$ در گام $i + 1$ ام، فعال می‌شود [۱۹]. در خط ۲ از الگوریتم ۲ ابتدا الگوریتم ICIM-GREEDY فراخوانی می‌شود. در خط ۴-۵ الگوریتم به

همان‌طور که در بالا اشاره شد $T=5$ است پس پیچیدگی زمانی کلی به صورت $O(kn+2m+2n)=O(kn+m)$ می‌شود.

n تعداد نودها است. پیچیدگی الگوریتم ۱، $O(2m+2n)=O(m+n)$ می‌شود. مرتبه زمانی کلی الگوریتم به صورت $O(kTn+2m+2n)$ است که

Algorithm 2 NewGreedyWC(G, k)

Input: G : Social graph, k : size seed set

Output: selected seed set

1: initialize $S = \emptyset$
 2: call algorithm ICIM-GREEDY
 3: **for** $i = 1$ to k **do**
 4: set $s_v = 0$ for all vertices.
 5: obtain $G' = \text{RanWC}(G)$
 6: compute DAG G'^* and weights $w(v^*)$ for all $v^* \in V^*$
 7: **for** $\ell = 1$ to T **do**
 8: for each $v^* \in V^*$, generate random value $X_{v^*}^\ell$ from The exponential distribution with mean $1/w(v^*)$
 9: **for each** $v^* \in V^*$, compute $Y_{v^*}^\ell = \min_{u^* \in R_{G'^*}(S^* \cup \{v^*\})} X_{v^*}^\ell$
 10: for each $v^* \in V^*$, $s_{v^*}^\ell += Y_{v^*}^\ell$
 11: **end for**
 12: for each $v \in V \setminus S$, $s_v += (T - 1)/s_{v^*}^\ell$
 13: **end for**
 15: $S = S \cup \{ \text{argmax}_{v \in V \setminus S} \{s_v\} \}$
 16: **end for**
 17: output S

ارسال کرده باشند، یال بدون جهت بین ایمیل‌ها تشکیل می‌شود [۳۹]. شبکه شامل 1k نود و 5k یال است. Netscience: این یک شبکه از همکاری‌های نویسندگان در حوزه علوم شبکه است [۳۹]. این شبکه شامل 1k نود و 2k یال است.

M-Fo115: شبکه M-Fo115 از مدل جنگل آتش با احتمال اتصال $p = 0.115$ تولید شده است که شامل 10k نود و 23k یال است [۴۰].

M-Fo120: با استفاده از مدل جنگل آتش، شبکه M-Fo120 با احتمال $p = 0.120$ که شامل 10k نود و 29k ساخته می‌شود [۴۰].

M-Fo-0.1: شبکه M-Fo-0.1 از مدل جنگل آتش با احتمال اتصال $p = 0.1$ تولید شده است که شامل 5k نود و 7k یال است [۴۰].

M-Fo-0.2: با استفاده از مدل جنگل آتش، شبکه M-Fo-0.2 با احتمال $p = 0.120$ که شامل 5k نود و 23k ساخته می‌شود [۴۰].

۴- آزمایش‌ها و ارزیابی

ما ابتدا الگوریتم ICIM-Greedy را بر روی ۸ مجموعه داده (دیتاست) مورد ارزیابی قرار می‌دهیم. دیتاست‌ها بدون جهت می‌باشند. اندازه‌ی دیتاست‌ها بزرگ و متوسط می‌باشند. هر ۸ دیتاست از سایت KONECT دانلود شده‌اند.

NetHEPT: این شبکه از بخش "High Energy Physics-Theory" است که مقالات مربوط سال ۱۹۹۱ تا ۲۰۰۳ است هر نود در شبکه یک نویسنده را ارائه می‌دهد و یال‌های بین دو نود برابر با همکاری دو نویسنده در یک مقاله است [۱۹]. این شبکه شامل 15k نود و 58k یال است.

Physics: این شبکه شامل تمامی مقالات بخش "Physics" است. در شبکه Physics نیز هر نود در شبکه یک نویسنده را ارائه می‌دهد و یال‌های بین دو نود برابر با همکاری دو نویسنده در یک مقاله است [۱۹]. این شبکه شامل 15k نود و 231k یال است.

Email: دیتاست Email شبکه‌ای از ایمیل در دانشگاه Rovira i Virgili است که اگر افراد به یکدیگر ایمیلی

جدول ۱- مشخصات دیتاست‌های مورد نظر برای آزمایش

Data set	NetHEPT	Physics	M-Fo115	M-Fo120	M-Fo-0.1	M-Fo-0.2	Email	NetScience
Node	317k	6k	10k	10k	5k	5k	1k	1k
Edge	1M	13k	23k	29k	7k	23k	5k	2k
Max Degree	343	1459	155	229	20	336	209	43
Min Degree	1	1	1	1	1	1	1	1

- Distance: یک الگوریتم اکتشافی ساده است که k نود با کوچک‌ترین میانگین فاصله را انتخاب می‌کند [۵].
- Random: الگوریتمی است که k نود را به صورت تصادفی انتخاب می‌کند [۵].
- LIR: الگوریتمی که با استفاده از توپولوژی شبکه k نود را انتخاب می‌کند [۲۹].
- ProbDeg: الگوریتمی که با تأثیر درجه همسایگان، k نود را انتخاب می‌کند [۳۱].
- TI-SC: الگوریتمی که با استفاده از تشخیص جامعه، k نود را انتخاب می‌کند [۱۱].

۴-۳- فرایند آزمایش

در این مقاله، آزمایش شبیه‌سازی توسط پایتون را بر روی سروری با پردازنده‌ی Intel Xeon E5410 و 32 گیگابایت حافظه انجام شده است که شش مجموعه داده شبکه اجتماعی و ده الگوریتم مقایسه اتخاذ شده است.

۴-۴ نتایج

برای این الگوریتم‌ها گسترش نفوذ را برای مجموعه تأثیرگذار در محدوده‌ی ۱ تا ۳۰ مقایسه می‌کنیم. همچنین زمان اجرا را برای $k=30$ مقایسه می‌کنیم.

گسترش نفوذ و زمان اجرا، نتایج گسترش نفوذ بر ۸ دیتاست بررسی شده است. اندازه‌ی مجموعه تأثیرگذار رنجی بین ۱ تا ۳۰ است. شکل‌های ۳ تا ۱۰ نتایج گسترش نفوذ را بر ۸ دیتاست با مدل WC را نشان می‌دهد و شکل ۴ زمان اجرای الگوریتم‌های متفاوت را نشان می‌دهد. در این آزمایش، هدف اصلی مقایسه الگوریتم‌های حریصانه با روش پیشنهادی ICIM-GREEDY می‌باشد زیرا روش‌هایی مانند Random و Degree به خاطر ضعف بارز در گسترش نفوذ به دلیل عدم انتخاب آگاهانه، الگوریتم‌های

۴-۱- پارامترهای مورد استفاده در آزمایش:

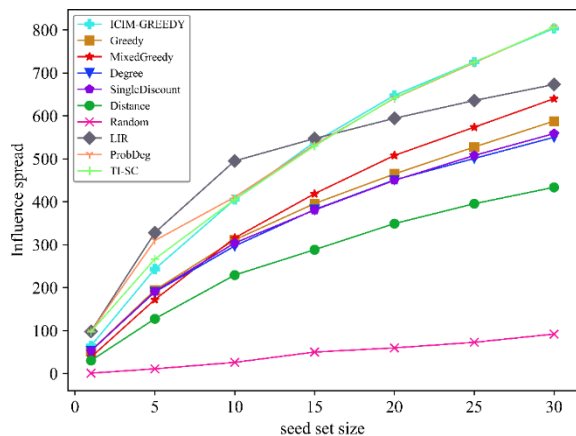
الف: تولید احتمال انتشار. به صورت کلی بر روی الگوریتم‌های مقایسه شده توزیع احتمال انتشار به صورت توزیع یکنواخت است، که از مدل زیر برای تولید این احتمال یکنواخت استفاده می‌کنیم:

ب: استفاده از مدل WC. مدل آبخاری وزن دار توسط کمپ و همکارانش در سال ۲۰۰۳ پیشنهاد شده است [۱۹]. احتمال بر روی یال‌ها به صورت $\frac{1}{\deg(v)}$ است که $\deg(v)$ درجه نود v است. گراف انتشار بر روی این مدل به صورت جهت دار است به دلیل اینکه احتمال انتشار $\frac{1}{\deg(v)}$ از یال (u,v) به یال (v,u) متفاوت است و تعداد تکرار شبیه‌سازی مونت کارلو ۲۰۰۰۰ تکرار است.

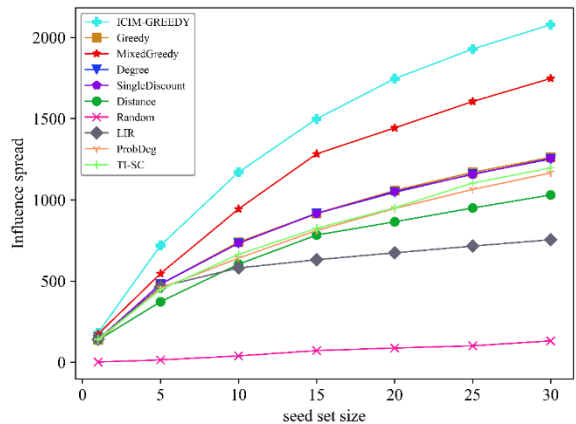
۴-۲- الگوریتم‌های مورد مقایسه

الگوریتم ICIM-GREEDY را با دو نوع از الگوریتم‌های حریصانه و هیوریستیک مقایسه کردیم. لیستی از الگوریتم‌های مقایسه شده را در زیر ارائه داده‌ایم:

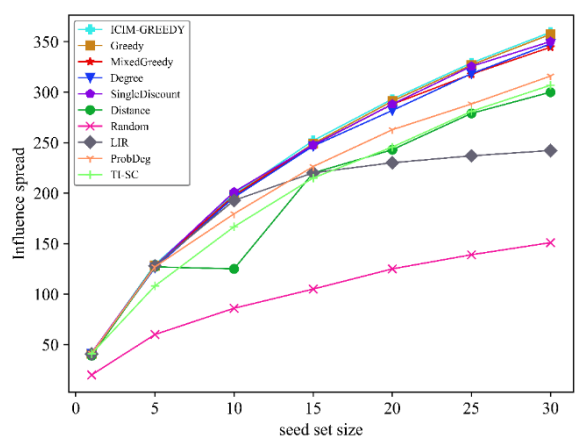
- ICIM-GREEDY: الگوریتم ۱
- Greedy: الگوریتم حریصانه معمولی که از شبیه‌سازی مونت کارلو استفاده می‌کند [۱۹].
- MixedGreedy: ترکیبی از الگوریتم حریصانه که در دور اول از NewGreedy و در دور دوم از CELF استفاده می‌کند، که $T=5$ و $R=20000$ است [۱۹].
- Degree: الگوریتم اکتشافی که k نود با بالاترین درجه را انتخاب می‌کند [۱۹].
- SingleDiscount: الگوریتم اکتشافی ساده است وقتی که هر یک از همسایه‌های که به عنوان تأثیرگذار انتخاب می‌شوند درجه کاهش آن یک است [۱۹].



شکل ۵- گسترش نفوذ بر روی الگوریتم‌های متفاوت بر روی M-Fo115

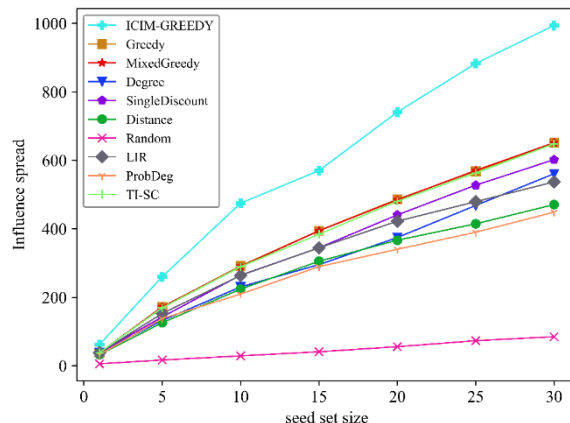


شکل ۶- گسترش نفوذ بر روی الگوریتم‌های متفاوت بر روی M-Fo120

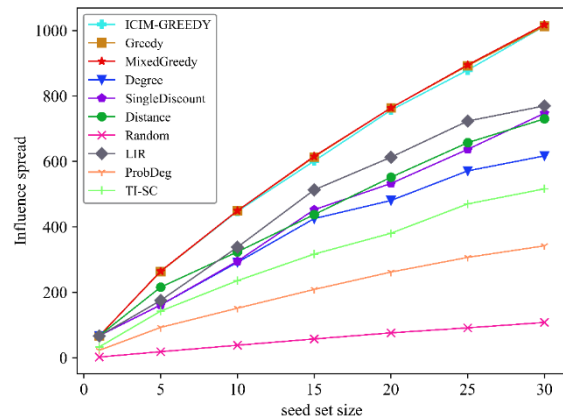


شکل ۷- گسترش نفوذ بر روی الگوریتم‌های متفاوت بر روی Email

خوبی نیستند. بر روی دیتاست NetHEPT (شکل ۳) الگوریتم ICIM-GREEDY گسترش نفوذ بهتر و چشمگیری نسبت به بقیه‌ی الگوریتم‌ها دارد و بدترین گسترش نفوذ را دارد. در دیتاست Physics (شکل ۴) الگوریتم‌های Greedy و MixedGreedy و ICIM-GREEDY گسترش نفوذ بهتری نسبت به بقیه دارند که هر سه از نظر گسترش نفوذ تقریباً یکسان و بعضاً با اختلاف بسیار ناچیز ظاهر شده‌اند. در دیتاست Physics، روش Random بدترین گسترش نفوذ را دارد. همچنین، شکل‌های ۵ تا ۱۰ نیز نشان می‌دهد که روش پیشنهادی بر روی مابقی دیتاست‌ها از نظر گسترش نفوذ، نتایج بهتری نسبت به الگوریتم‌های دیگر دارد.



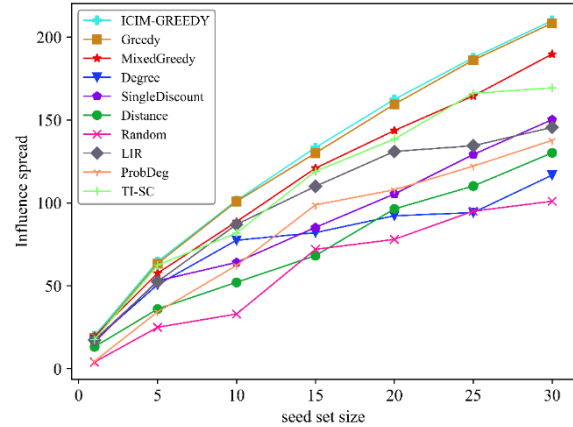
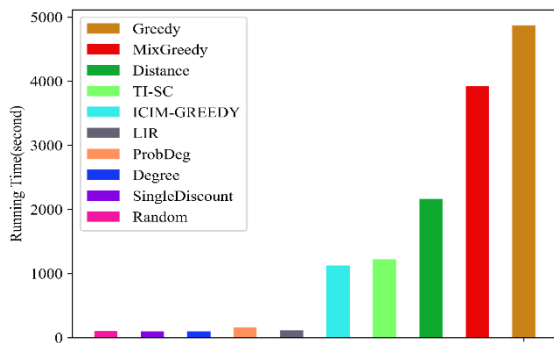
شکل ۳- گسترش نفوذ بر روی الگوریتم‌های متفاوت بر روی NetHEPT



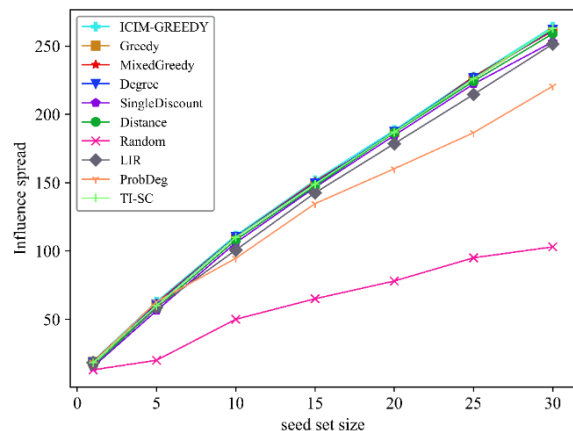
شکل ۴- گسترش نفوذ بر روی الگوریتم‌های متفاوت بر روی Physics

در شکل ۷ گسترش نفوذ بر دیتاست Email مورد بررسی قرار گرفته است که الگوریتم‌های ICIM-GREEDY و Greedy گسترش نفوذ تقریباً نزدیکی دارند و به نسبت دیگر الگوریتم‌ها گسترش نفوذ بهتری ارائه می‌دهند. البته الگوریتم ICIM-GREEDY با اختلاف بسیار کم گسترش نفوذ بهتری از الگوریتم Greedy ارائه می‌دهد. در شکل ۸، گسترش نفوذ بر دیتاست NetScience مورد بررسی قرار گرفته است که الگوریتم ICIM-GREEDY گسترش نفوذ بهتری ارائه می‌دهد. در این دیتاست الگوریتم Greedy گسترش نفوذ تقریباً نزدیکی به الگوریتم ICIM-GREEDY ارائه می‌دهد ولی گسترش نفوذ در الگوریتم ICIM-GREEDY بیشتر از الگوریتم Greedy است. به عنوان مثال در الگوریتم ICIM-Greedy گسترش نفوذ در $k=25$ و $k=30$ نسبت به الگوریتم Greedy بهتر است. در شکل ۹ و ۱۰ گسترش نفوذ بر دیتاست‌های M-Fo-0.1 و M-Fo-0.2 مورد بررسی قرار گرفته است که الگوریتم‌های ICIM-GREEDY و GREEDY گسترش نفوذ تقریباً نزدیکی دارند و به نسبت دیگر الگوریتم‌ها گسترش نفوذ بهتری ارائه می‌دهند.

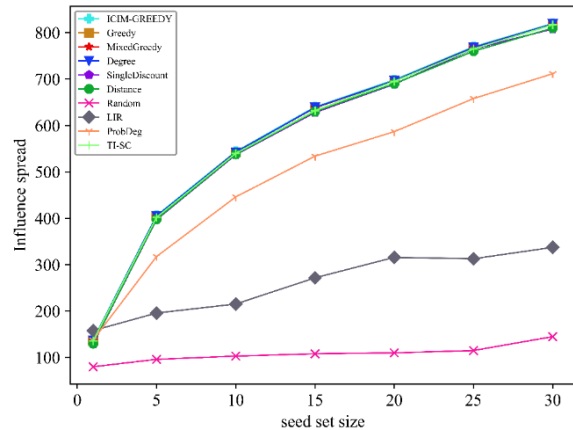
در شکل‌های ۱۱ تا ۱۸، الگوریتم‌ها از نظر زمان اجرا روی دیتاست‌های مذکور مورد آزمایش قرار گرفته‌اند. همان‌طور که در این شکل‌ها قابل‌مشاهده است، الگوریتم پیشنهادی نسبت به الگوریتم‌های حریصانه مورد مقایسه زمان اجرای کمتری دارد. البته الگوریتم‌های Degree و Random دارای کمترین مرتبه‌ی زمانی هستند ولی این الگوریتم‌ها دارای دقت بسیار پایینی در محاسبه گسترش نفوذ دارند.



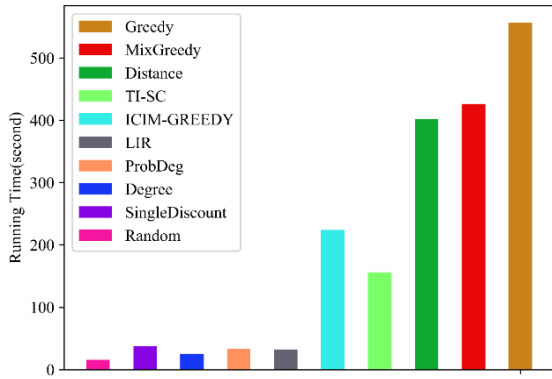
شکل ۸- گسترش نفوذ بر روی الگوریتم‌های متفاوت بر روی NetScience



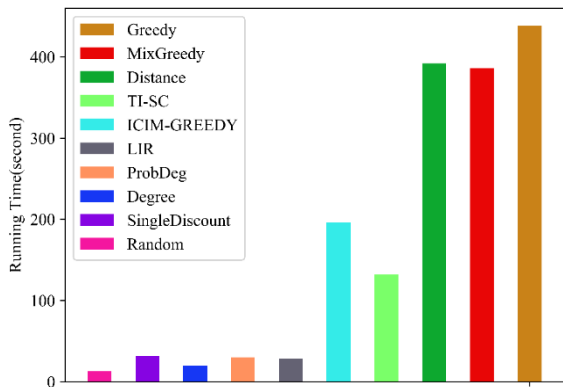
شکل ۹- گسترش نفوذ بر روی الگوریتم‌های متفاوت بر روی M-Fo-0.1



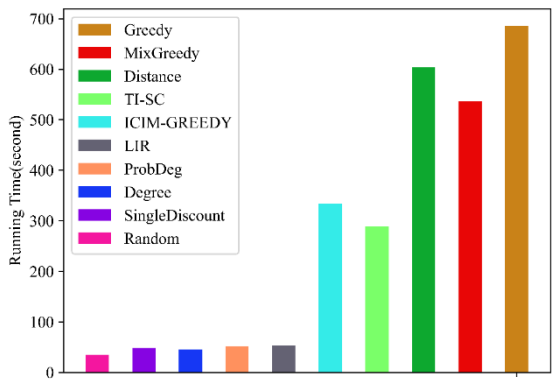
شکل ۱۰- گسترش نفوذ بر روی الگوریتم‌های متفاوت بر روی M-Fo-0.2



شکل ۱۵- زمان اجرای الگوریتم‌های متفاوت بر روی دیتاست Email

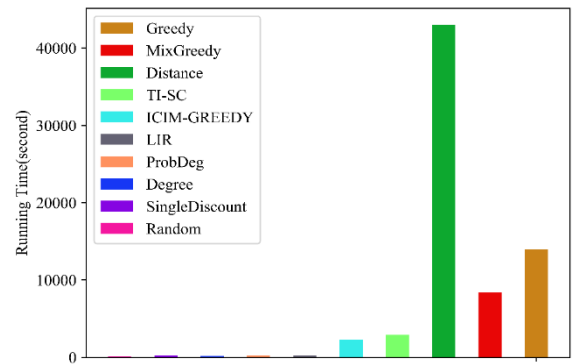


شکل ۱۶- زمان اجرای الگوریتم‌های متفاوت بر روی دیتاست Netscience

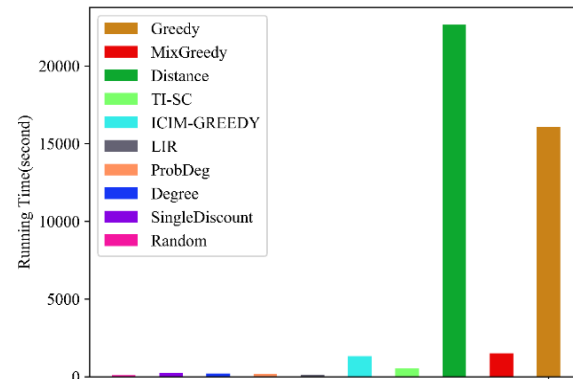


شکل ۱۷- زمان اجرای الگوریتم‌های متفاوت بر روی دیتاست M-Fo-0.1

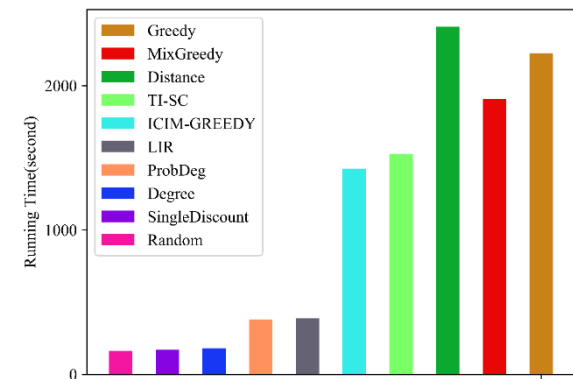
شکل ۱۱- زمان اجرای الگوریتم‌های متفاوت بر روی دیتاست NetHEPT



شکل ۱۲- زمان اجرای الگوریتم‌های متفاوت بر روی دیتاست Physics



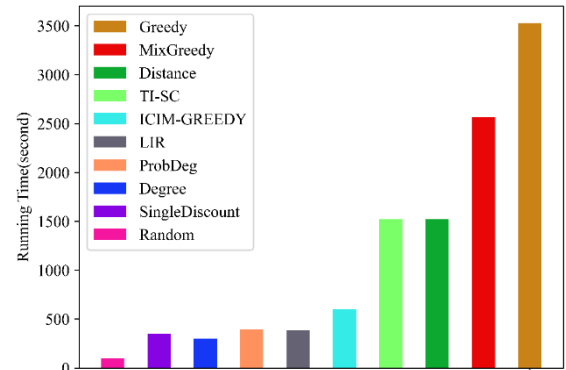
شکل ۱۳- زمان اجرای الگوریتم‌های متفاوت بر روی دیتاست M-FO115



شکل ۱۴- زمان اجرای الگوریتم‌های متفاوت بر روی دیتاست M-FO120

منابع

1. P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 57-66: ACM.
2. D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137-146: ACM.
3. S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *International workshop on web and internet economics*, 2007, pp. 306-311: Springer.
4. C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 665-674.
5. W. Chen *et al.*, "Influence maximization in social networks when negative opinions may emerge and propagate," in *Proceedings of the 2011 siam international conference on data mining*, 2011, pp. 379-390: SIAM.
6. X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model," in *Proceedings of the 2012 siam international conference on data mining*, 2012, pp. 463-474: SIAM.
7. W. Lu, W. Chen, and L. V. Lakshmanan, "From competition to complementarity: comparative influence diffusion and maximization," *arXiv preprint arXiv:1507.00317*, 2015.
8. H. Ma, Y. Zhu, D. Li, D. Kim, and J. Liang, "Improving the influence under IC-N model in social networks," *Discrete Mathematics, Algorithms and Applications*, vol. 7, no. 03, p. 1550037, 2015.
9. N. Pathak, A. Banerjee, and J. Srivastava, "A generalized linear threshold model for multiple cascades," in *2010 IEEE International Conference on Data Mining*,



شکل ۱۸- زمان اجرای الگوریتم‌های متفاوت بر روی دیتاست M-Fo-0.2

در این آزمایش، هدف اصلی مقایسه الگوریتم‌های حریصانه با ICIM-GREEDY است زیرا روش‌هایی مثل روش Degree و Random به خاطر ضعف بارز در گسترش نفوذ به دلیل عدم انتخاب آگاهانه، الگوریتم‌های خوبی نیستند و در این حوزه هم چندان مورد توجه نیستند.

۵- نتیجه‌گیری

در این مقاله بر اساس معیارهای جدید حساسیت به نفوذ و قدرت نفوذ یک الگوریتم بهینه برای مسئله حداکثر سازی نفوذ ارائه شده است. در الگوریتم ICIM-GREEDY از معیارهای که در دنیای واقعی وجود دارد، برای محاسبه حساسیت به نفوذ و قدرت نفوذ استفاده شده است. الگوریتم‌های حریصانه برای دقت بیشتر نیاز به تکرار بیشتر توسط شبیه‌سازی مونت کارلو دارند در صورتی که الگوریتم ICIM-GREEDY تنها با یک بار اجرا دارای دقت بالایی در گسترش نفوذ است به همین دلیل در این الگوریتم زمان اجرا نسبت به الگوریتم حریصانه کاهش داده شده است و دقت انتخاب مجموعه نودهای تأثیرگذار بر اساس گسترش نفوذ افزایش داده شده است. زمانی که زمان اجرا اهمیت دارد الگوریتم‌های SingleDiscount پیشنهاد می‌شود و زمانی که دقت اجرا اهمیت دارد الگوریتم‌های ICIM-GREEDY، MixedGreedy و TI-SC پیشنهاد می‌شود.

- 429: ACM.
19. W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 199-208: ACM.
20. R. Narayanam and Y. Narahari, "A shapley value-based approach to discover influential nodes in social networks," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 130-147, 2010.
21. M. Kitsak *et al.*, "Identification of influential spreaders in complex networks," *Nature physics*, vol. 6, no. 11, p. 888, 2010.
22. S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng, "Staticgreedy: solving the scalability-accuracy dilemma in influence maximization," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 509-518: ACM.
23. A. Goyal, W. Lu, and L. V. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," in *2011 IEEE 11th international conference on data mining*, 2011, pp. 211-220: IEEE.
24. Y.-C. Chen, W.-Y. Zhu, W.-C. Peng, W.-C. Lee, and S.-Y. Lee, "CIM: Community-based influence maximization in social networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 2, p. 25, 2014.
25. W. Chen, T. Lin, Z. Tan, M. Zhao, and X. Zhou, "Robust influence maximization," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 795-804: ACM.
26. J.X. Zhang, D.-B. Chen, Q. Dong, and Z.-D. Zhao, "Identifying a set of influential spreaders in complex networks," *Scientific reports*, vol. 6, p. 27823, 2016.
27. J. Shang, S. Zhou, X. Li, L. Liu, and H. Wu, "CoFIM: A community-based framework for influence maximization on large-scale networks," *Knowledge-Based Systems*, vol. 117, pp. 88-100, 2017.
28. F. Morone, B. Min, L. Bo, R. Mari, and 2010, pp. 965-970: IEEE.
10. C. Wang, W. Chen, and Y. Wang, "Scalable influence maximization for independent cascade model in large-scale social networks," *Data Mining and Knowledge Discovery*, vol. 25, no. 3, pp. 545-576, 2012.
11. H. A. Beni and A. Bouyer, "TI-SC: top-k influential nodes selection based on community detection and scoring criteria in social networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-20, 2020.
12. Y. Chen, Q. Qu, Y. Ying, H. Li, and J. Shen, "Semantics-aware influence maximization in social networks," *Information Sciences*, vol. 513, pp. 442-464, 2020.
13. Q. He *et al.*, "CAOM: A community-based approach to tackle opinion maximization for social networks," *Information Sciences*, vol. 513, pp. 252-269, 2020.
14. M. M. Keikha, M. Rahgozar, M. Asadpour, and M. F. Abdollahi, "Influence maximization across heterogeneous interconnected networks based on deep learning," *Expert Systems with Applications*, vol. 140, p. 112905, 2020.
15. J. Ko, K. Lee, K. Shin, and N. Park, "MONSTOR: An Inductive Approach for Estimating and Maximizing Influence over Unseen Social Networks," *arXiv preprint arXiv:2001.08853*, 2020.
16. J. Tang, R. Zhang, P. Wang, Z. Zhao, L. Fan, and X. Liu, "A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks," *Knowledge-Based Systems*, vol. 187, p. 104833, 2020.
17. A. Zareie, A. Sheikahmadi, and M. Jalili, "Identification of influential users in social network using gray wolf optimization algorithm," *Expert Systems with Applications*, vol. 142, p. 112971, 2020.
18. J. Leskovec *et al.*, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 420-

- 13, 2019.
34. G. Xie, Y. Chen, H. Zhang, and Y. Liu, "MBIC: A Novel Influence Propagation Model for Membership-Based Influence Maximization in Social Networks," *IEEE Access*, vol. 7, pp. 75696-75707, 2019.
35. X. Rui, F. Meng, Z. Wang, and G. Yuan, "A reversed node ranking approach for influence maximization in social networks," *Applied Intelligence*, vol. 49, no. 7, pp. 2684-2698, 2019.
36. L. Qiu, W. Jia, J. Yu, X. Fan, and W. Gao, "PHG: A Three-Phase Algorithm for Influence Maximization Based on Community Structure," *IEEE Access*, vol. 7, pp. 62511-62522, 2019.
37. S. Milgram, "Behavioral study of obedience," *The Journal of abnormal and social psychology*, vol. 67, no. 4, p. 371, 1963.
38. V. Batagelj and M. Zaversnik, "An O (m) algorithm for cores decomposition of networks," *arXiv preprint cs/0310049*, 2003.
39. J. Kunegis, "Konekt: the koblenz network collection," in *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 1343-1350.
40. A.L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509-512, 1999.
- H. A. Makse, "Collective influence algorithm to find influencers via optimal percolation in massively large social media," *Scientific reports*, vol. 6, p. 30062, 2016.
29. D. Liu, Y. Jing, J. Zhao, W. Wang, and G. Song, "A fast and efficient algorithm for mining top-k nodes in complex networks," *Scientific reports*, vol. 7, p. 43330, 2017.
30. S. Ahajjam and H. Badir, "Identification of influential spreaders in complex networks using HybridRank algorithm," *Scientific reports*, vol. 8, no. 1, p. 11932, 2018.
31. D.-L. Nguyen, T.-H. Nguyen, T.-H. Do, and M. Yoo, "Probability-based multi-hop diffusion method for influence maximization in social networks," *Wireless Personal Communications*, vol. 93, no. 4, pp. 903-916, 2017.
32. H. Wu, J. Shang, S. Zhou, Y. Feng, B. Qiang, and W. Xie, "LAIM: A linear time iterative approach for efficient influence maximization in large-scale networks," *IEEE Access*, vol. 6, pp. 44221-44234, 2018.
33. S. Banerjee, M. Jenamani, and D. K. Pratihari, "ComBIM: A community-based solution approach for the Budgeted Influence Maximization Problem," *Expert Systems with Applications*, vol. 125, pp. 1-