**Annealing Simulated مؤثر برای مسئله حداکثر رساندن تأثیر در**

**شبکه های اجتماعی آنلاین**

**چکیده:**

مشکل بزرگنمایی تأثیر (IMP) یکی از شناخته شده ترین مشکلات حوزه تحقیقات شبکه های اجتماعی آنلاین است

OSN (Online Social Networks) که در سالهای اخیر مورد توجه بسیاری از محققان رشته های مختلف قرار گرفته است. یکی از دلایل این است که اگر بتوانیم کاربرانی را پیدا کنیم که بیشترین تأثیر را بر سایر کاربران داشته باشند سرعت انتشار اطلاعات در سیستم عامل افزایش می یابد. الگوریتم های سنتی مبتنی بر قانون و اکتشاف پذیر ممکن است نتوانند اطلاعات مفیدی را از این داده ها پیدا کنند زیرا داده ها به طور کلی بزرگ و پیچیده هستند. اگرچه برای حل IMP از الگوریتم های metahheuristic می توان استفاده کرد ، اما هنوز فضای زیادی برای بهبود وجود دارد. به همین دلیل یک الگوریتم کارآمد در این مقاله ارائه شده است. الگوریتم پیشنهادی ، به نام annealing شبیه سازی با پارتیشن جستجو (SASP) ، بر اساس مکانیسم پارتیشن بندی فضای جستجو برای افزایش عملکرد جستجوی بازپخت شبیه سازی شده برای IMP است. نتایج تجربی نشان می دهد که الگوریتم پیشنهادی از سایر الگوریتم های مسئله حداکثر نفوذ تأثیر در مقایسه با این مقاله از نظر کیفیت نتیجه نهایی و تعداد ارزیابی های عملکرد عینی بهتر است.

**مقدمه:**

شبکه اجتماعی آنلاین (OSN) روش دیگری برای ارتباط مردم با هم از طریق فناوری های رایانه ای و اینترنتی مدرن ، مانند شبکه حسگر بی سیم ، محاسبات ابری ، اینترنت چیزهای 2،3،4 فراهم می کند. متفاوت از نرم افزار سنتی چت ، توسعه جدید OSN قادر به ارائه خدمات فانتزی زیادی به کاربران است. در میان آنها ، یک مسئله جالب بهینه سازی در شبکه های اجتماعی آنلاین با نام مشکل حداکثر نفوذ (IMP) نامگذاری شده است. برای حل این مشکل روشی مؤثر و کارآمد مورد نیاز است زیرا می توانیم ارتباط بین کاربران ، رفتار کاربران و تأثیر کاربران را از داده های OSN دریابیم. به دلیل مسائل مربوط به حریم خصوصی ، ما نمی توانیم به تجزیه و تحلیل رفتار آنها به داده های خام برخی از کاربران دسترسی پیدا کنیم ، اما هنوز می توان علایق برخی از کاربران ناشناس را از داده های رفتار کاربر شناسایی نشده تشخیص داد. بنابراین ، روابط دوستی یا خصمانه به راحتی استخراج می شود.

شناخت علایق کاربر فقط یک مثال ساده برای نشان دادن امکانات این داده های مفید است. در حقیقت ، مطالعات متعددی سعی در ایجاد یک مکانیزم یا ابزاری قدرتمندتر برای درک رفتار کاربر برای سالها داشته است. در ، Benevenuto و همکاران. یک مدل احتمال انتقال برای درک مرحله بعدی کاربران از داده های کلیک بر روی آنها ایجاد کرده است.

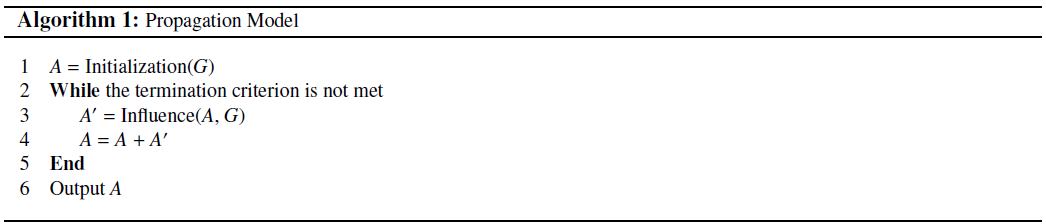
به عنوان یک مشکل بهینه سازی برای درک رفتار کاربر در OSN ، یک IMP مشهور توسط Kempe و همکارانش معرفی شدند. مشکل فرض می کند که یک نمودار G (V، E، W) مخفف OSN است و پارامترهای آن V، E، W به شرح زیر است:

V = {V1 ، V2 ،. . . ، V a مجموعه ای از گره ها است که Vi را می توان به عنوان کاربر در نظر گرفت ،

E = {E1،2 ، E1،3 ،. . . ، E1،، E2،1، E2،3،. . . E، −1 set مجموعه ای از لبه ها است ، که در آن Ei ، j می تواند به عنوان رابطه بین دو کاربر ، i j ، و

W = {W1،2 ، W1،3 ،. . . ، W1 ، ، W2،1 ، W2،3 ،. . .W ، −1 a مجموعه ای از وزن ها است ، که در آن هر Wi ، j با لبه Ei ، j E در ارتباط است.

به طور خلاصه ، هدف از این مشکل یافتن مجموعه ای از گره های A از G است که حداکثر نفوذ (A) را دارند ، جایی که A شامل <گره ها است که هرکدام نماینده یک کاربر هستند. همچنین از آنجا که مشکل NP-hard9،10 است ، چگونگی دستیابی به روشی با کارایی بالا برای یافتن گره هایی که تأثیر حداکثر را در یک زمان معقول به دست می آورند یک مسئله تحقیقاتی مهم است ، خصوصاً وقتی تعداد گره های OSN زیاد باشد.



همانطور که در خط 1 الگوریتم 1 نشان داده شده است ، مدلهای انتشار ، همانطور که در 9/10 برای اندازه گیری تأثیر A شرح داده شده است ، ابتدا فرض می کنند که مجموعه ای از گره ها از G فعال می شوند در حالی که سایرین غیرفعال خواهند ماند. سپس گره ها (یعنی کاربران) در A سعی خواهند کرد گره های همسایه غیرفعال خود را فعال کنند (یعنی تأثیر بگذارند) بر اساس قانون احتمالی داده شده همانطور که در خط 3 الگوریتم نشان داده شده است. یک انتشار

احتمال P گره ها در G. این توضیح می دهد که چگونه از این مدل احتمال استفاده کنید تا تصمیم بگیرید که آیا یک گره فعال با موفقیت بر گره های دوست غیرفعال خود تأثیر می گذارد تا آنها فعال شود یا خیر. همانطور که در خط 4 الگوریتم 1 نشان داده شده است ، گره های فعال جدید "A'" به مجموعه گره های فعال A اضافه می شوند.

این روش برای اندازه گیری حداکثر تأثیر ، از تمام کاربران (گره ها) OSN استفاده نمی کند. در عوض ، هر آزمایش فقط از بخشی از گره ها برای اندازه گیری حداکثر تأثیر A. استفاده می کند. همچنین ، بسیاری از دورها نیز باید انجام شوند ، و میانگین A به عنوان معیار نتایج استفاده می شود ، که در اینجا S تعریف می شود. مجموعه بذر A؛ من (S) تأثیر مجموعه بذر S؛ تعداد نمونه ها؛ و من (S) تأثیر بذر S در دور اول.



همانطور که قبلاً نیز اشاره کردیم ، توسعه یک روش کارایی بالا برای یافتن مجموعه "خوب" گره A در حل مسئله حداکثر نفوذ ، یک موضوع تحقیقاتی مهم است. به همین دلیل است که چن و همکاران. 11 یک الگوریتم حریص برای IMP ارائه داد ، به نام NewGreedy. زیرا ایده اصلی NewGreedy اینست که همه غیرفعال ها را بررسی کنید

گره ها در هر دور ، قادر به یافتن مجموعه ای از گره های A به سرعت می باشند. همچنین به دلیل اینکه IMP NP سخت است ، چندین مطالعه اخیر سعی در استفاده از الگوریتم های فراگرایی برای حل این مشکل دارند. یکی از الگوریتم های مشهور فراشناختی ، الگوریتم ژنتیکی (GA) ، در چندین مطالعه1213 مورد استفاده قرار گرفته است.

Guo و همکارانش با نگه داشتن تنها 50٪ کروموزومهای برتر (یعنی استراتژی انتخاب براساس مقادیر تناسب اندام این کروموزومها) یک استراتژی نخبه را برای بهبود عملکرد فرایند تکامل GA ارائه داده اند. کروموزومهای موجود در جمعیت. به عبارت دیگر ، 50٪ کروموزومها را در هر نسل از بین می برد زیرا از تناسب اندام آنها است

بدتر از 50٪ دیگر کروموزومها هستند. مشاهدات ما نشان می دهد که این استراتژی به سرعت یک راه حل "خوب" را پیدا می کند ، اما وقتی مسئله بزرگ و پیچیده باشد ، در نسل اولیه به راحتی در محلی بهینه تبدیل می شود. Tsai و همکاران برای بهبود عملکرد الگوریتم حریص و GA برای حل این مشکل. 13 با ادغام GA با الگوریتم NewGreedy به نام GNA ، الگوریتم متابوریستی ترکیبی را توسعه داد. نتایج شبیه سازی نشان می دهد که GNA قادر به یافتن نتیجه بهتر از روشهای سنتی برای حل IMP مانند NewGreedy و GA است.

**2. الگوریتم پیشنهادی**

**2.1. نشانه گذاری**

برای ساده کردن بحثی که در زیر آمده است ، از یادداشت زیر در بقیه مقاله استفاده می شود.

:Sیک راه حل برای IMP؛ یعنی ، s = {s1 ، s2 ،. . . ، sn} ، جایی که si ، iامین زیرراه حل از راه حل های موجود و n تعداد زیرراه حل های ممکن است.

Snn : تعداد راه حل های همسایه (نامزد) ایجاد شده توسط الگوریتم پیشنهادی برای راه حل های s.

Ψ: دمای فعلی الگوریتم پیشنهادی.

1,ΨminΨ : دمای اولیه و حداقل دمای الگوریتم پیشنهادی

r : کل فضای جستجو ، یعنی r = {r1 ، r2 ،. . . ، rh} ، که در آن تعداد مناطق است.

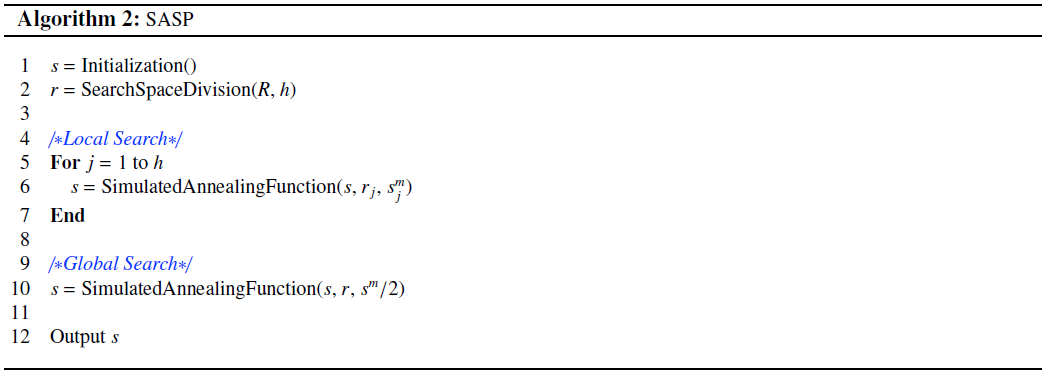
rj: a set of nodes representing a region in *r*, i.e., *r j* = {*Vh*×( *j*−1)+1, *Vh*×( *j*−1)+2, . . . , *V*\_*h* ×*j*},

sm : معیار الگوریتم پیشنهادی را برای کل فرایند جستجو ، مثلاً ، تعداد ارزیابی ها یا تعداد تکرارها متوقف کنید.

Smj : stop criterion of the proposed algorithm for the *j*-th region, which is set to *sm*/(2*h*) in this paper.

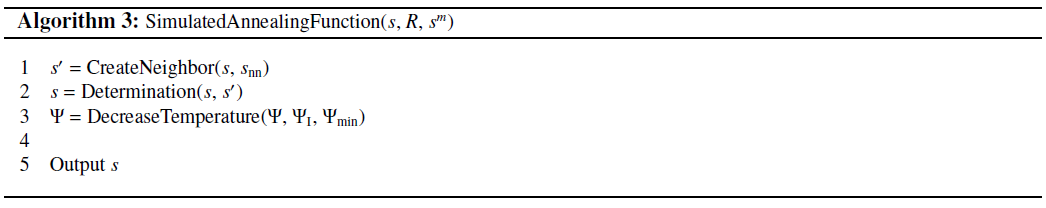
**2.2. Annealing شبیه سازی شده با بخش جستجو**

الگوریتم پیشنهادی از اقتصاد جستجو13 الهام گرفته شده است که در ابتدا فضای جستجو را به مجموعه ای از مناطق تقسیم می کند. مشخصه اصلی الگوریتم پیشنهادی این است که نه تنها فضای جستجو را به مناطق مختلف تقسیم می کند و به ترتیب از آنیل شبیه سازی شده (SA) برای جستجوی این مناطق استفاده می کند ، بلکه نتایج جستجو شده از مناطق مختلف را نیز برای بهبود بیشتر راه حل جستجو شده استفاده می کند. این تفاوت اصلی بین الگوریتم پیشنهادی با سایر الگوریتم های جستجوی تقسیم و غلبه است.



همانطور که در خط 2 الگوریتم 2 نشان داده شده است ، الگوریتم پیشنهادی پارامترها را تنظیم کرده و داده ها را ابتدا با استفاده از عملگر Initialization () وارد می کند. سپس الگوریتم پیشنهادی با استفاده از اپراتور SearchSpaceDivision () کل فضای جستجو را به مناطق h تقسیم می کند. هر منطقه با مجموعه ای از گره ها از V شبکه های اجتماعی آنلاین ، r j مرتبط است.

instance, suppose there are eight nodes in *V* (i.e., *V*={*V*1, *V*2, *V*3, *V*4, *V*5, *V*6, *V*7, *V*8}) and are divided into two regions; thus, the nodes in the first region will be {*V*1, *V*2, *V*3, *V*4}. It can be calculated as *r j* = {*Vh*×( *j*−1)+1, *Vh*×( *j*−1)+2, . . . , *V*\_*h* ×*j*}. If *h* = 2 and *j* = 1, then *r*1 = {*V*2×(1−1)+1, *V*2×(1−1)+2, . . . , *V*\_2 ×1} = {*V*1, *V*2, *V*3, *V*4}. For *j* = 2, *r*2 = {*V*5, *V*6, *V*7, *V*8} Lines 5–7 of Algorithm 2 show that the proposed algorithm will then apply the SimulatedAnnealingFunction() operator (a modified simulated annealing method), where the arguments *s*, *r j*, *smj*are, respectively, the initial solution, the region, and the stop criterion of this operator, to search each region in turn.

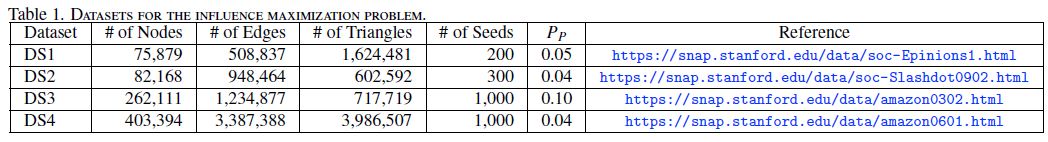


As shown in Algorithm 3, the SimulatedAnnealingFunction() operator is similar to simple simulated annealing that will create a set of neighbors. However, the search space will be restricted by the parameters *r* and *sm*; that is, the SimulatedAnnealingFunction() operator will search not the entire search space, but the restricted search space. For example, suppose the nodes {1, 2, 3, 4} are in the first region, and the nodes {5, 6, 7, 8} are in the second region. The exchange nodes of *s* will be restricted to nodes {1, 2, 3, 4} if *R* = {1, 2, 3, 4}. The Determination() operator is similar to that of simple SA, which decides whether to accept a non-improving solution or not. The probability of accepting a non-improving solution is defined as

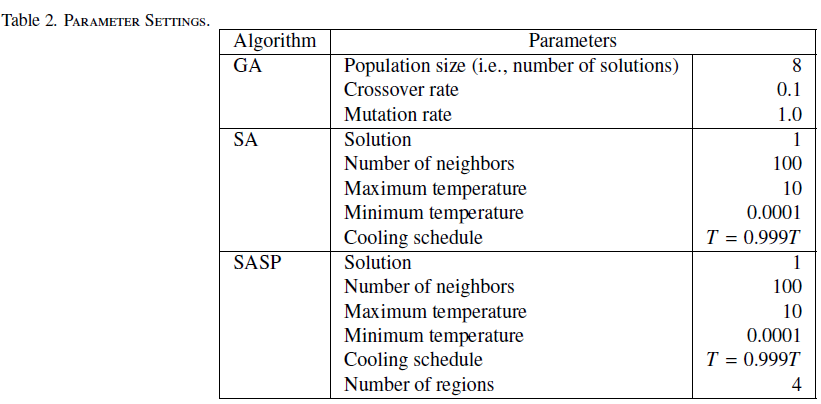


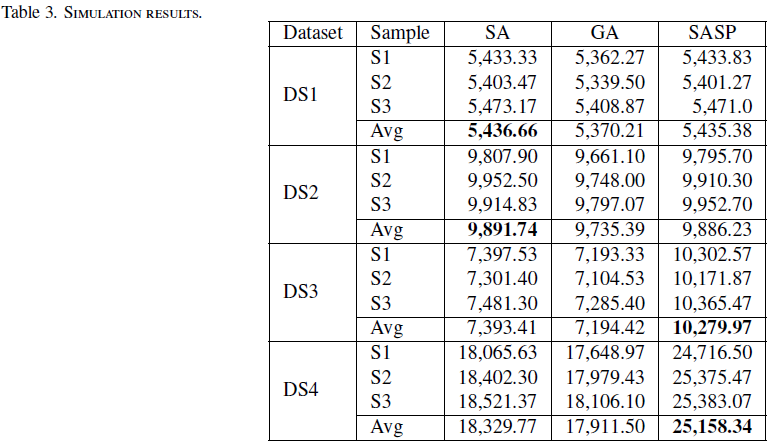
where *f* (·) denotes the evaluation function, *s* the current solution, *s*\_ the new candidate solution, and \_ the temperature. That is, if *s* satisfies the probabilistic acceptance criterion of Eq. 1 (i.e., *r* < *Pa*), where *r* is a random number in the range [0, 1). The current solution *s* and its objective value *f* (*s*) will be replaced by the new candidate solution *s*\_ and its objective *f* (*s*\_). The temperature decrease schedule of the DecreaseTemperature() operator is set to\_ = 0.999\_ as far as this paper is concerned.

**3. نتایج شبیه سازی**



در این بخش ، چندین مجموعه داده از مشکلات حداکثر نفوذ برای ارزیابی عملکرد الگوریتم پیشنهادی استفاده شده است. تجزیه و تحلیل تجربی بر روی یک کامپیوتر با پردازنده Intel i5-4590 CPU 3.30GHz و 4 گیگابایت حافظه در حال اجرا ویندوز 7 انجام شده و برنامه ها در C ++ نوشته شده اند. همانطور که در جدول 1 نشان داده شده است ، چهار مجموعه داده مشهور که شامل 75،879 گره تا 403394 گره هستند برای ارزیابی عملکرد الگوریتم پیشنهادی استفاده می شوند. ستون با عنوان "# دانه ها" تعداد گره هایی را که می خواهیم از مجموعه داده های آزمایش پیدا کنیم که بیشترین تأثیر را روی سایر گره ها در همان شبکه دارد ، نشان می دهد. ستون با عنوان "PP" نشانگر احتمال انتشار گره ها است در حالی که ستون با عنوان "مرجع" منبع این مجموعه داده ها را نشان می دهد.





هر آزمایش به مدت 30 اجرا انجام می شود و تعداد تکرارها در هر اجرا معادل 600000 تعیین می شود در حالی که تمام نتایج تجربی نشان داده شده میانگین 30 اجرا است. در جدول 2 تنظیمات پارامتر SASP ، الگوریتم ژنتیکی (GA) و بازپرداخت شبیه سازی شده (SA) ارائه شده است. طبق مشاهدات ما ، تنظیمات کلی GA (به عنوان مثال ، با نرخ متقاطع برابر با 0.6 و نرخ جهش برابر با 0.01 تعیین شده است) نتایج بدتری را برای حل این چهار مشکل حداکثر نفوذ به همراه خواهد داشت. به همین دلیل است که در این مطالعه میزان متقاطع و جهش GA برابر 0/0 و 1.0 تعیین شده است ، بدین معنی که تنها 10٪ کروموزوم ها برای متقاطع انتخاب می شوند در حالی که تمام کروموزوم ها جهش می یابند.

در جدول 3 ، S1 ، S2 و S3 نمونه هایی به طور تصادفی از مجموعه داده ها انتخاب شده است. نتایج نشان می دهد که SA نتایج بهتری را نسبت به GA و SASP به طور متوسط برای مجموعه داده های کوچک فراهم می کند. یعنی DS1 و DS2 گرچه تفاوت این نتایج معنی دار نیست. به عنوان مثال ، تفاوت بین SASP و SA در مورد DS2 در حدود 0.056 [[.0000.00056 = (9 ، 886.23 - 9891.74) / 9 ، 891.74] است. با این حال ، برای مجموعه داده های در مقیاس بزرگ ، الگوریتم پیشنهادی می تواند نتیجه بهتری نسبت به دو الگوریتم متاوریستی دیگر ارائه دهد. تفاوت بین آنها قابل توجه است. مثلا، برای DS4 ، اختلاف بین SASP و SA از نظر میانگین نتایج در حدود 37.253٪ [0.37253 = (25 ، 480.12 - 18 ، 329.77) / 18 ، 329.77] است. نتایج همچنین نشان می دهد که الگوریتم پیشنهادی برای حل مجموعه داده های مقیاس بزرگ IMP مفید است.

**4. نتیجه گیری**

در این مقاله ، ما یک الگوریتم کارآمد مبتنی بر SA را برای حل مسئله حداکثر نفوذ که فضای جستجوی هر جستجو را در طی فرایند همگرایی محدود می کند ارائه می دهیم. با استفاده از این استراتژی ، SASP قادر خواهد بود از تنوع جستجو در باریک شدن به یک جهت خاص در طی فرایند همگرایی جلوگیری کند. طراحی الگوریتم پیشنهادی ، توازن تنوع و تشدید جستجو را نیز در نظر می گیرد. یعنی پنجاه درصد برای جستجوی محلی و پنجاه درصد برای جستجوی جهانی. نتایج شبیه سازی نشان می دهد که الگوریتم پیشنهادی می تواند نتیجه بهتری را نسبت به سایر الگوریتم های متاوریستی ارزیابی شده در این مقاله ، به ویژه برای مجموعه داده های در مقیاس بزرگ ، پیدا کند. این نتایج بیشتر نشان می دهد که مقیاس پذیری SASP بهتر از سایرین است زیرا می تواند نتایج جستجو شده را قبل از پیدا کردن راه حل بهینه بهبود بخشد. از آنجا که این مطالعه امکان تقسیم فضای جستجو را نشان می دهد که ممکن است برای افزایش کارایی سایر الگوریتم های مبتنی بر الگوریتم متاوریستیستی مفید باشد ، ما سعی خواهیم کرد راه های دیگری برای تقسیم فضای جستجو برای توسعه یک کارآمدتر و کارآمدتر مبتنی بر الگوریتم SA پیدا کنیم.