مجله تحقیق در عملیات در کاربردهای آن، سال شانزدهم، شماره دوم (پیاپی ۶۱)، تابستان ۹۸، ص ص ۸۷-۷۷
شاپا ۷۲۸۶-۲۲۵۱

یافتن تعداد افراد با شرایط خاص در شبکه‌های اجتماعی و سازمان‌های بزرگ با استفاده از برنامه‌ریزی هم مثبت

سید سعید نبوی[۱]، محمد رضا صافی[۲*]
۱- دکتری ریاضی، دانشگاه سمنان، گروه ریاضی، سمنان، ایران
۲- استادیار، دانشگاه سمنان، گروه ریاضی، سمنان، ایران

رسید مقاله: ۱۱ تیر ۱۳۹۷
پذیرش مقاله: ۲۷ اردیبهشت ۱۳۹۸

چکیده

در شبکه‌های اجتماعی و یا سازمان‌های بزرگ، یافتن تعداد افرادی که همگی با هم رابطه‌ی معینی دارند (w^*) و یا تعداد افرادی که هیچ یک از آن‌ها رابطه‌ای معین با یکدیگر ندارند ($\overline{w^*}$)، دو مساله مورد توجه مدیران است. چنانچه افراد را رئوس و ارتباطات مذکور را یال‌های یک گراف در نظر بگیریم، w^* عدد کلیک و $\overline{w^*}$ تعداد رئوس زیرگراف مستقل آن را نشان می‌دهد. یافتن این دو عدد به حل برخی مدل‌های بهینه‌سازی منجر می‌شود. یکی از این مدل‌ها، مدل برنامه‌ریزی هم مثبت است. متاسفانه برای حل مساله‌ی هم مثبت، یک روش با زمان چندجمله‌ای وجود ندارد. در این مقاله این مسایل را به صورت مسایل هم مثبت، مدل‌سازی و برای حل آن‌ها از الگوریتم‌های ژنتیک و ازدحام ذرات استفاده می‌کنیم. همچنین کارایی روش خود را با حل دو مثال نشان می‌دهیم.

کلمات کلیدی: بهینه‌سازی در سازمان‌ها و شبکه‌های اجتماعی، برنامه‌ریزی هم مثبت، الگوریتم ژنتیک، الگوریتم ازدحام ذرات، کلیک گراف‌ها، زیرگراف مستقل.

۱ مقدمه

در شبکه‌های اجتماعی، شرکت‌ها و سازمان‌هایی که تعداد اعضای آن‌ها خیلی زیاد است مدیریت منابع انسانی از جنبه‌های مختلف، اهمیت بیش‌تری پیدا می‌کند. داشتن برخی روابط معین بین افراد در این گروه‌ها همواره اهمیت ویژه‌ای دارد. به عنوان مثال داشتن بیماری مشترک، خویشاوندی، هم شهری بودن و یا مواردی دیگر در چیدمان کارکنان در زیرمجموعه‌های مختلف سازمان و یا در امور تبلیغاتی در شبکه‌های اجتماعی دارای اهمیت زیادی است.

* عهده‌دار مکاتبات
آدرس الکترونیکی: msafi@semnan.ac.ir

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]

برخی مولفان از جنبه‌های مختلف به کاربردهای نظریه گراف در مدیریت سازمان‌ها پرداخته‌اند. آناند و کودالی [۱] برای تعیین اهداف و مسئولیت‌های افراد از روابط گراف‌ها استفاده کردند. نمونه‌هایی از کاربرد گراف در مدیریت منابع انسانی را می‌توانید در [۲] و [۳] ببینید.

برای تشریح مسایلی که در این مقاله به آن‌ها می‌پردازیم، رابطه معین R را بین کارمندان یک سازمان در نظر بگیرید. هر کارمند یک راس از گراف محسوب می‌شود. اگر دو کارمند رابطه R باهم داشته باشند بین این دو راس یال وجود دارد و در غیر این صورت یالی وجود ندارد؛ بنابراین در یک سازمان یا شبکه اجتماعی، لازم است برای هر رابطه خاص یک گراف ویژه آن رابطه درنظر گرفت.

درشبکه‌های اجتماعی مجموعه‌هایی از افراد که همگی آن‌ها با هم ارتباط معینی دارند مجموعه‌های حائز اهمیت در تصمیم‌گیری‌ها تلقی می‌شوند. این مجموعه‌ها در گراف‌ها کلیک-گراف نامیده می‌شوند. تعداد عناصر بزرگ‌ترین مجموعه‌ها از این نوع، عدد کلیک نام دارد. یافتن عدد کلیک در گراف‌ها یک مساله قدیمی و مهم به حساب می‌آید. همچنین مجموعه‌هایی از افراد که هیچ دو تای آن‌ها با یکدیگر ارتباط معینی را ندارند نیز مجموعه مهمی از نظر مدیران به حساب می‌آید. قطعا یافتن تعداد افراد بزرگ‌ترین مجموعه از این نوع نیز می‌تواند مساله مهمی باشد. این مجموعه یک گراف مستقل نامیده می‌شود. این مسایل مورد توجه محققان بوده و تعداد قابل توجهی مقاله در این زمینه وجود دارد که از روش‌های مختلف به حل آن پرداخته‌اند. به دلیل آن که مسایل بهینه‌سازی مذکور معمولا NP-hard هستند، برای حل آن‌ها از روش‌های ابتکاری استفاده می‌شود.

در میان مقالات مختلف، برخی مستقیما از الگوریتم‌های ابتکاری و اشکال بهبود یافته آن‌ها مانند ژنتیک، ازدحام ذرات، کولونی مورچه‌ها و ... استفاده کرده‌اند. به عنوان مثال، بزگان و لوچیان در [۴] با استفاده از الگوریتم ژنتیک به مساله پرداخته‌اند؛ اما طبق گفته خودشان روش آن‌ها صرفا از دیدگاه تئوری حائز اهمیت است و از نظر عملی برای گراف‌های متنوع کارامد نیست. در مرجع [۵] نویسندگان، الگوریتم ژنتیک را باروش شاخه و کران روی مساله به‌کار برده‌اند ولی نتایج عددی ارایه نداده‌اند که روش خود را از نظر زمانی کارامد نشان دهند. استفاده از کلونی مورچگان در مقاله [۶] دیده می‌شود که؛ البته روی همه گراف‌ها جواب نمی‌دهد. این کار را سلیمانی پور و همکارانش در [۷] با اضافه کردن الگوریتم ازدحام ذرات تکمیل کردند؛ اما زمان اجرای برنامه آن‌ها نسبتا بالاست.

مسایل مورد بررسی در این مقاله به مسایل برنامه‌ریزی هم مثبت تبدیل می‌شود که از نوع محدب هستند. در مراجع [۸] و [۹] در مورد حل مساله‌ی برنامه‌ریزی هم مثبت روش‌هایی ارایه شده است. در واقع مسایل گسسته مانند مسایل مورد بررسی در این مقاله به مساله‌ی هم مثبت تبدیل می‌شود که برای اطلاعات بیش‌تر می‌توانید مرجع [۱۰] را ببینید. اسپانسل و همکاران در [۸] با افراز سیمپلکس واحد و تظریف هر مرحله از این افراز، یک مساله‌ی برنامه‌ریزی خطی متناظر با آن را حل کرد و الگوریتمی‌ارایه داده‌اند که دنباله‌ای از جواب‌های ساخته شده به مقدار دقیق همگراست؛ اما اشکال اساسی در آن، این است که با افزایش تکرارها تعداد اعضای افراز زیاد شده و حل مساله‌ی برنامه‌ریزی خطی مربوطه مشکل و گاهی ناممکن می‌شود. تاناکا و یوشیز در [۹] اصلاحاتی برای کار قبلی ارایه کرده‌اند؛ اما روش آن‌ها هنوز برخی مشکلات روش قبل را دارد و در برخی موارد حتی بدتر از آن

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]

عمل می‌کند. همچنین در برخی از مراجع خود مساله‌ی هم مثبت و دوگان آن را حل کرده‌اند. بونفس و دور، در [۱۱] روشی برای حل مسایل هم مثبت ارایه داده‌اند که در آن دنباله ای از برنامه‌ریزی‌های خطی حل می‌شود تا مساله‌ی اصلی حل گردد. در [۱۲] ییلدریم نیز دنباله‌ای از برنامه‌ریزی‌های خطی را برای حل مساله می‌سازد. صافی و نبوی در [۱۳] روش ییلدریم را اصلاح نمودند و روش جدیدی برای حل این دسته از مسایل معرفی کرده‌اند. در این مقاله ما از ترکیب دو الگوریتم ژنتیک و ازدحام ذرات روی مساله‌ی هم مثبت استفاده می‌کنیم. روش ما در زمان بسیار مناسب نتیجه را معلوم می‌کند.

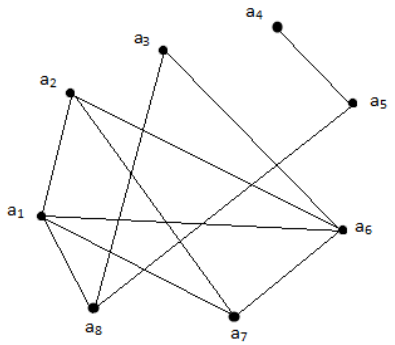
سازمان این مقاله به این شرح است: در بخش دوم برخی مقدمات لازم از گراف ها، برنامه‌ریزی هم مثبت، الگوریتم ژنتیک و الگوریتم ازدحام ذرات آورده شده است. روش استفاده از الگوریتم‌های مذکور در حل مسایل یافتن عدد کلیک و گراف پایدار در بخش سوم تشریح خواهد شد. در بخش چهارم روش‌های مطرح شده در بخش سوم را روی دو مثال پیاده‌سازی می‌کنیم.

۲ پیش‌زمینه‌های لازم

۲-۱ مقدمات لازم از نظریه گراف

یک گراف G، زوج مرتب (V,E) است که در آن V مجموعه رئوس و E مجموعه‌ی یال‌هاست که نشان‌دهنده‌ی روابط بین رئوس می‌باشد. ماتریس مجاورت گراف G ماتریسی با عناصر صفر و یک است. اگر بین راس‌های i و j یالی وجود داشته باشد، آنگاه مولفه‌ی (i,j) ماتریس برابر با یک و در غیر این صورت صفر است.

مثال ۱ فرض کنید در یک شرکت ۸ کارمند وجود دارد که راس‌های گراف هستند و با $v=\{a_1,a_2,...,a_8\}$ نشان داده شده اند. فرض کنید داشتن بیماری مشترک یک رابطه خاص در این شرکت به حساب می‌آید. هر دو نفر که بیماری مشترک دارند در شکل ۱ با یال مشترک نشان داده شده است:



شکل ۱. گراف مثال ۱ (رابطه‌ای در کارکنان یک شرکت)

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]

توجه داشته باشید که فقط یک بیماری مشترک را در نظر نگرفته‌ایم. به عنوان مثال، بیماری مشترک a_1 و a_2 ممکن است با بیماری مشترک a_2 و a_3 متفاوت باشد. به عبارت دیگر گراف مذکور خاصیت تعدی ندارد. در این مثال، ماتریس مجاورت به صورت جدول ۱ است.

حال اگر مدیر بخواهد افرادی را در یک اتاق کار قرار دهد که هیچ دو نفر آن‌ها بیماری مشترکی نداشته باشند، به دنبال مجموعه مستقل در گراف است. برخی از این مجموعه‌ها عبارتند از:

$$\{a_2, a_3, a_4\}, \{a_6, a_5, a_1\}, \{a_6, a_5, a_8\}$$

همچنین اگر بخواهد مجموعه افرادی را در نظر بگیرد که هر دو نفر آن‌ها یک بیماری مشترک با یکدیگر دارند با یک کلیک گراف مواجه هستیم. در این صورت

$$\{a_1, a_2, a_6, a_7\}, \{a_2, a_1, a_7\}, \{a_1, a_6, a_7\}, \{a_2, a_6, a_7\}$$

مثال‌هایی از این مجموعه‌ها هستند و البته $\{a_1, a_2, a_6, a_7\}$ بزرگ‌ترین آن‌هاست. عدد کلیک این گراف ۴ است؛ یعنی بزرگ‌ترین مجموعه از لحاظ تعداد از این نوع دارای ۴ عضو است.

جدول ۱. ماتریس مجاورت گراف شکل۱

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
a_1	۰	۱	۰	۰	۰	۱	۱	۱
a_2	۱	۰	۰	۰	۰	۱	۱	۰
a_3	۰	۰	۰	۰	۰	۱	۰	۱
a_4	۰	۰	۰	۰	۱	۰	۰	۰
a_5	۰	۰	۰	۱	۰	۰	۰	۱
a_6	۱	۱	۱	۰	۰	۰	۱	۰
a_7	۱	۱	۰	۰	۰	۱	۰	۰
a_8	۱	۰	۱	۰	۱	۰	۰	۰

۲-۲ مقدمات لازم از برنامه‌ریزی هم مثبت

یک مساله بهینه‌سازی در حالت کلی به صورت زیر است:

$$\begin{aligned} &\text{Min (Max)} \quad f(x) \\ &s.t. \qquad\quad x \in S \end{aligned} \tag{۱}$$

که در آن به دنبال یافتن بردار X(بردار تصمیم) هستیم که علاوه بر بهینه‌سازی $f(X)$ در مجموعه‌ی محدودیت‌های S صدق کند. در برخی مسایل، X به جای اینکه یک بردار باشد، یک ماتریس و حتی یک تابع است. همچنین S می‌تواند مجموعه‌ای از معادلات خطی و یا غیرخطی و حتی یک مخروط باشد.

تعریف ۱ : X یک ماتریس هم مثبت نامیده می‌شود هرگاه برای هر بردار نامنفی a داشته باشیم، $a^T Xa \geq 0$ مجموعه ماتریس‌های هم مثبت را با C نشان می‌دهند. چنانچه در مساله (۱) تابع f یک تابع خطی، S مجموعه‌ای حاصل از اشتراک جواب‌های نامعادلات خطی (نیم فضاها) و همچنین X ماتریسی هم مثبت باشد، مساله را یک

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]

مساله برنامه‌ریزی هم مثبت می‌نامند. در واقع یک مساله برنامه‌ریزی خطی داشته باشیم که مجهول مساله ماتریسی هم مثبت باشد.

فرض کنید A ماتریس مجاورت یک گراف باشد. I ماتریس همانی و J ماتریسی باشد که همه عناصر آن ۱ است. برای یافتن تعداد راس‌های بزرگ‌ترین مجموعه پایدار در گراف مربوطه (بیش‌ترین تعداد افرادی که هیچ دو نفر آن‌ها با یکدیگر رابطه معینی ندارند) لازم است مساله زیر حل شود [۱۴]:

$$\begin{aligned} \text{Min} \quad & y \\ s.t. \quad & y(A+I)-J \in C \\ & y \in N \end{aligned} \qquad (۲)$$

یعنی باید کم‌ترین مقدار y را طوری به‌دست آوریم که $y(A+I)J$ یک ماتریس هم مثبت باشد.

همچنین برای یافتن عدد کلیک (بیش‌ترین تعداد افرادی که هر دو نفر آن‌ها با هم رابطه معینی دارند) لازم است مساله زیر را حل کنیم [۸]

$$\begin{aligned} \text{Min} \quad & w \\ s.t. \quad & w(J-A)-J \in C \\ & w \in N \end{aligned} \qquad (۳)$$

یعنی باید کوچک‌ترین عدد طبیعی w را طوری بیابیم که برای آن $w(J-A)-J$ یک ماتریس هم مثبت باشد.

۲-۳ توصیف ساده‌ای از الگوریتم‌های ابتکاری در حل مسایل بهینه‌سازی

به طور کلی برای حل یک مساله با روش‌های تکراری دو راه کار اعمال می‌شود. یک راه کار این است که با نقطه‌ای شروع می‌کنند که در شرایط مساله صدق‌کند؛ ولی الزاما بهترین جواب مساله نیست و طی یک فرایند تکراری، گام به گام از یک نقطه به نقطه بعدی می‌روند که هم شرایط مساله را و هم مقدار تابع هدف بهتری نسبت به جواب قبلی داشته باشد.

راه کار دیگر این است که از یک نقطه X_0 شروع می‌کنند که به وضوح از بهترین جواب دارای شرایط مساله (یعنی X^*) بهتر است؛ ولی شرایط مساله را ندارد و طی یک فرایند تکراری، گام به گام از یک نقطه به نقطه بعدی می‌روند به‌طوری‌که در ضمن اینکه بهتر بودن از X^* را حفظ می‌کند، گام به گام تمام شرایط مساله را دارا شود.

در هر دو راه کار، برای تکرار گام به گام از روش‌های مختلف استفاده می‌شود. برخی از این روش‌ها برپایه روش‌های صرفا ریاضی هستند مانند: روش دو بخشی، روش نیوتن، روش گرادیان مزدوج و یا سایر روش‌های جستجو که با الگوریتم‌های ریاضی همراه هستند؛ ولی برخی دیگر برپایه آنچه در طبیعت رخ می‌دهد شکل می‌گیرند. الگوریتم‌های ژنتیک، ازدحام ذرات (پرندگان)، مورچگان، شبکه‌های عصبی و ... از این نوع هستند.

الگوریتم ژنتیک اولین بار توسط جان هنری هالند (۲۰۱۵ - ۱۹۲۹) معرفی شد. این الگوریتم برای شروع یک یا چند (معمولا ۲) ورودی می‌گیرد. از این دو ورودی که البته شرایط مساله را نیز دارند، به عنوان والدین (جمعیت اولیه) استفاده و یک یا چند خروجی به عنوان فرزند (فرزندان) تولید می کند و با استفاده از فرزند (فرزندان) و یا

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]

والدین، جواب‌های بعدی را تولید می‌کند. فرایند تولید جواب‌های (فرزندان) بعدی، با استفاده از دو عمل انجام می‌شود. یکی را تابعی به نام آمیزش (crossover) با عملی که روی والدین انجام می‌دهد و عمل دیگر را تابعی به نام جهش (motivation) انجام می‌دهد. درست مانند آنچه در بحث ژنتیک رخ می‌دهد؛ یعنی فرزندان برخی ویژگی‌های خود را مستقیما از والدین به ارث می‌برند و برخی ویژگی‌ها را که موروثی نیستند به عنوان جهش کسب می‌کنند. در ضمن تولید فرزندان، تابعی به نام برازش (fitness) بررسی خود را روی فرزندان تولید شده انجام می‌دهد و چنانچه فرزندان، شرایط مطلوب را داشته باشند، فرایند تولید را متوقف می‌کند و آخرین فرزند دارای شرایط را به عنوان بهترین گزینه معرفی می‌کند.

الگوریتم ازدحام ذرات (پرندگان) در سال ۱۹۹۵ توسط کندی و آبراهارت ابداع شد. این الگوریتم، الهام گرفته از رفتار گروهی پرندگان و ماهی‌هاست. این الگوریتم با گروهی از ذرات (نقاط) از داخل جواب ها (نقاطی که شرایط مساله را دارند؛ ولی لزوما بهترین نقطه نیستند) شروع می‌کند. هر ذره (پرنده) در هر زمان یک موقعیت و یک سرعت مربوط به آن زمان را دارد و برای حرکت از موقعیت فعلی به یک موقعیت بهتر (حرکت به سمت یک نقطه یا جواب بهتر) از دانش فردی خود (بهترین موقعیتی که تا آن لحظه داشته) و دانش دیگر ذرات گروه (بهترین موقعیت سایر نقاط در همسایگی) استفاده کرده و سرعت حرکت و مکان بعدی خود را برای رسیدن به موقعیت بهتر تعیین می‌کند. در تعیین بهترین وضعیتی که خودش و یا سایر ذرات تا لحظه کنونی داشته‌اند از تابعی به نام تابع برازش استفاده می‌کند. همچنین با همین تابع، عمل بررسی بهینگی انجام می‌شود و اگر مقدار تابع برازش در این موقعیت جدید دال بر عدم بهینگی باشد، فرایند تغییر مکان تکرار می‌گردد.

در دو الگوریتم مذکور برای توقف فرایند تکرار، معمولا شرط‌های مختلفی را به طور همزمان در نظر می‌گیرند.

۳ یافتن تعداد کارکنان خاص بین کارکنان یک شرکت

فرض کنید مدیر یک شرکت یا یک سازمان و یا در حالت کلی‌تر یک شبکه اجتماعی، رابطه خاصی را بین کارکنان و یا اعضای خود در نظر می‌گیرد. او به دنبال یافتن پاسخ دو سوال زیر است:

۱- در بزرگ‌ترین مجموعه از کارکنانش که هیچ دوتای آن‌ها با یکدیگر این رابطه را ندارند چند نفر قرار می‌گیرند؟

۲- در بزرگ‌ترین مجموعه از کارکنانش که در آن هر دو عضو آن رابطه خاص را دارند چند نفر قرار می‌گیرند؟

مدیر برای هر رابطه که از نظر او در مدیریت سازمانش دارای اهمیت است یک ماتریس مجاورت درست می‌کند و برای این ماتریس نام ویژه آن رابطه را می‌گذارد. در هر ماتریس برای هر دو نفر در مکان مربوطه یا عدد ۱ قرار می دهد یا ۰. ۱ برای وقتی دو نفر آن رابطه را دارند و در غیر این صورت ۰. برای هر فرد که به سازمان اضافه می‌شود به هر ماتریس یک سطر و یک ستون اضافه می‌گردد و رابطه آن فرد با سایرین در هر ماتریس مشخص می‌شود. همچنین در صورت خروج هر فرد از سازمان، سطر و ستون مربوط به آن فرد، از ماتریس حذف می‌شود؛ بنابراین مدیر، ماتریس مجاورت تمام رابطه‌ها را در اختیار دارد و این ماتریس‌ها با به کارگیری (خروج) هر فرد جدید (قدیم) به هنگام می‌شوند.

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]

اما برای یافتن پاسخ دو سوال مذکور لازم است مسایل (۲) و (۳) حل شود. حل این مسایل در حالت کلی و به‌صورت نظری امکان‌پذیر نیست. روش‌های عددی مختلفی برای حل این مسایل ارایه شده است. در اینجا ما از الگوریتم‌های ژنتیک و ازدحام ذرات برای حل آن‌ها استفاده می‌کنیم.

۳-۱ الگوریتم ژنتیک

حل مدل (۳) را در اینجا با الگوریتم ژنتیک شرح می دهیم؛ سپس عملیات الگوریتم ازدحام ذرات را برای آن تشریح می‌کنیم. حل مساله دیگر کاملا مشابه است. بدنه ژنتیکی برای این مساله به صورت جدول ۲ است.

جدول ۲. بدنه ژنتیکی برای مساله (۳)

جمعیت	بردارهای a با تعداد مولفه برابر با تعداد کارکنان
تابع برازش	بررسی علامت $a^T(w(J-A)-J)a$
آمیزش	ساخت چارک اول و چارک آخر مولفه‌های فرزند از ترکیب محدب مولفه‌های نظیر در والدین
جهش	ساخت ۵۰٪ میانی مولفه‌های فرزند با اعداد تصادفی

با w برابر ۲ شروع می‌کنیم (زیرا در بهترین حالت ممکن، این عدد مقدار بهینه تابع هدف است). با کمک الگوریتم مذکور سعی در یافتن بردار a می‌کنیم به طوری که $a^T(w(J-A)-J)a$ منفی شود. ابتدا دو بردار تصادفی در سایز a تولید می‌کنیم. چنانچه هیچ کدام از این دو بردار حاصل عبارت قبل را منفی نکرد، بردارهای بعدی را با استفاده از تابع آمیز در جدول ۳ تولید می‌کنیم. اگر در زمان کمتر از ۶۰ ثانیه و تعداد تکرار کم‌تر از ۱۰۰۰ برداری یافت شود که حاصل عبارت مذکور را منفی نماید w فعلی بهینه نیست و باید مقدار آن را افزایش دهیم. در غیر این صورت w فعلی کاندیدای بهینه است و آن را w^* می‌نامیم. برای چند w با مقادیر بیش‌تر از w^* فرایند را تکرار می‌کنیم اگر در هیچ‌کدام از این wها در زمان کم‌تر از ۶۰ ثانیه و یا تکرار کم‌تر از ۱۰۰۰ مقدار منفی برای $a^T(w(J-A)-J)a$ یافت نشد، مقداری بین w^* و آخرین w ماقبل آن را امتحان می‌کنیم. چون w^* باید عددی طبیعی باشد فرایند به سرعت متوقف می شود و جواب بهینه به‌دست می‌آید.

۳-۲ الگوریتم ازدحام ذرات

قسمت‌های اولیه روش در این الگوریتم کاملا شبیه قسمت‌های اولیه در الگوریتم ژنتیک است و تفاوت در نحوه تولید بردارهایی است که توسط آن‌ها وضعیت هم مثبتی $w(J-A)-J$ را بررسی می‌کنیم. بدنه الگوریتم ازدحام ذرات در جدول ۳ آمده است. در این الگوریتم برای مساله مذکور با یک مقدار دلخواه برای w و پنج بردار $a_{\cdot}^{1}, a_{\cdot}^{2}, a_{\cdot}^{3}, a_{\cdot}^{4}, a_{\cdot}^{5}$ که به طور تصادفی تولید می‌شوند شروع می‌کنیم. تعداد مولفه‌های هر کدام از این بردارها برابر تعداد کارکنان شرکت است. برای هر $1 \le l \le 5$ متناظر با هر بردار a_0^l یک بردار v_0^l به صورت تصادفی و به عنوان سرعت اولیه در نظر می‌گیریم. در هر مرحله اگر هم مثبت بودن $w(J-A)-J$ توسط بردارهای موجود مردود اعلام نشد پنج بردار جدید با استفاده از فرایند بهنگام‌سازی تولید می‌شوند.

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]

فرض کنید در مرحله i ام هستیم و بردارهای $a_i^1, a_i^2, a_i^3, a_i^4, a_i^5$ و $v_i^1, v_i^2, v_i^3, v_i^4, v_i^5$ را در اختیار داریم:

۱- اگر برای یک $1 \le l \le 5$ داشته باشیم $(a_i^l)^T (w(J-A)-J)a_i^l < 0$ در این صورت مقدار فعلی w بهینه نیست و باید با افزایش مقدار w به گام شروع الگوریتم و تولید بردارهای تصادفی برمی‌گردیم. در غیر این صورت به گام بعدی می‌رویم.

۲- قرار می‌دهیم: $a_{pbest}^l = \arg\min_i \left\{ (a_i^l)^T (w(J-A)-J)a_i^l \right\}$

۳- قرار می‌دهیم: $a_{gbest} = \arg\min_l \left\{ (a_{i-1}^l)^T (w(J-A)-J)a_{i-1}^l \right\}$

۴- قرار می‌دهیم: $v_{i+1}^l = r_l v_i^l + p_l(a_{pbest}^l - a_i^l) + q_l(a_{gbest} - a_i^l)$ که در آن اعداد مثبت r_l, p_l, q_l را به طور تصادفی تولید می‌کنیم. پس از تولید تصادفی هر سه عدد، هر یک را بر مجموع سه عدد تقسیم می‌کنیم تا از پراکندگی آن‌ها جلوگیری کنیم.

۵- قرار دهید: $a_{i+1}^l = a_i^l + v_{i+1}^l$ و به گام ۱ بر می‌گردیم.

جدول ۳. بدنه الگوریتم ازدحام ذرات

جمعیت	k بردار $a_i^1, a_i^2, ..., a_i^k$ در مرحله i ام (زمان t). تعداد مولفه هر بردار برابر با تعداد کارکنان شرکت. k بردار سرعت $v_i^1, v_i^2, ..., v_i^k$ در مرحله i ام (زمان t). برای بردارهای نظیر
تابع برازش	بررسی علامت $(a_i^l)^T (w(J-A)-J)a_i^l$ برای هر a_i^l که $1 \le l \le k$.
بهنگام سازی (وضعیت جدید ذرات در زمان بعدی)	$a_{i+1}^l = a_i^l + v_{i+1}^l$ $v_{i+1}^l = r_l v_i^l + p_l(a_{pbest}^l - a_i^l) + q_l(a_{gbest} - a_i^l)$

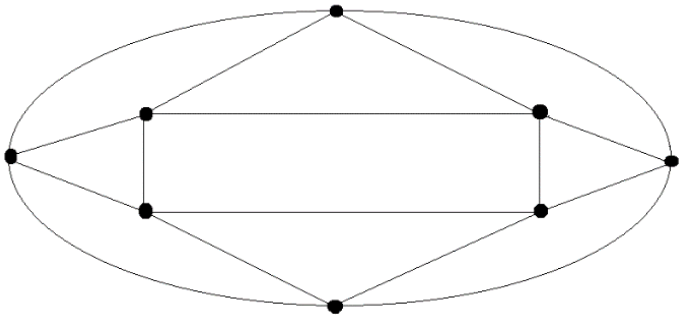
در فرایند بهنگام‌سازی a_{pbest}^l بهترین وضعیت بردار a^l در تمام تکرارهای قبلی است (که همان بهترین تجربه شخصی ذره i ام در تمام طول مسیر است). بهتر بودن به این معنی که مقدار $(a_i^l)^T (w(J-A)-J)a_i^l$ به صفر نزدیک‌تر باشد. در هر مرحله a_{gbest} بهترین بردار بین همه بردارهای مرحله قبلی است (که همان بهترین تجربه گروهی است منتهی فقط در مرحله قبلی). بهتر بودن اینجا هم به همان معنی قبلی است.

٤ دو مثال عددی

مثال ۲ شرکتی را در نظر بگیرید که ۸ کارمند دارد و رابطه معینی در آن با گراف شکل ۲ نشان داده شده است.

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]



شکل ۲. گراف مثال ۲ (یک رابطه خاص در کارکنان یک شرکت)

مدیر شرکت به دنبال یافتن پاسخ دو سوال ابتدای این بخش است. برای دو بردار ۸ مولفه‌ای به عنوان والدین به صورت زیر:

$$X=(x_1,x_2,...,x_8) \qquad Y=(y_1,y_2,...,y_8)$$

بردار فرزند $Z=(z_1,z_2,...,z_8)$ به صورت زیر تولید می‌شود:

$$z_j=\mu x_j+(1-\mu)y_j,\ j=1,2,3,6,7,8$$

که در آن μ عددی تصادفی بین صفر و یک است. اما مقادیر z_4 و z_5 را بدون توجه به والدین و صرفا به‌طور تصادفی به عنوان یک جهش ژنتیکی در فرزند، تولید می‌کنیم. در تکرارهای بعدی بین دو والد و فرزند، دو بردار را به تصادف به عنوان والدین بعدی انتخاب می‌کنیم.

جدول ٤. یافتن جواب بهینه مثال ۲ با الگوریتم ژنتیک

زمان (ثانیه)	تعداد تکرار	w	زمان (ثانیه)	تعداد تکرار	w
>۶۰	>۱۰۰۰	۳/۱	۰/۰۰۱	۱	۲
>۶۰	>۱۰۰۰	۳/۴	۰/۰۱۰	۵	۲/۶
>۶۰	>۱۰۰۰	۳/۶	۴/۰۷۶	۵۰	۲/۸
>۶۰	>۱۰۰۰	۴/۰	۶/۴۴۴	۱۸۳	۲/۹
>۶۰	>۱۰۰۰	۴/۲	>۶۰	>۱۰۰۰	۳/۰

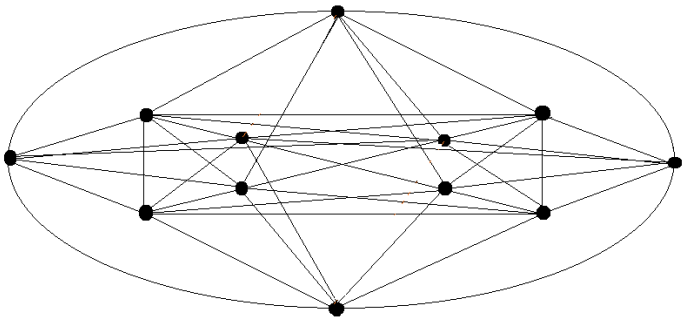
همان‌طور که در جدول ۴ دیده می‌شود برای w برابر ۲ در همان اولین تکرار و در زمان ۰/۰۰۱ ثانیه الگوریتم به ما می‌گوید این مقدار شدنی نیست. برای $w=2/6$ با ۵ تکرار و در زمان ۰/۰۱ ثانیه عدم شدنی بودن نتیجه می‌شود. دو مقدار بعدی $w=2/8$ و $w=2/9$ نیز مردود اعلام می‌شوند؛ اما برای $w=3$ قبل از شرط توقف (تکرار کم‌تر از ۱۰۰۰ و زمان کم‌تر از ۶۰ ثانیه) الگوریتم نمی‌تواند نشدنی بودن w را مردود اعلام کند. به همین دلیل مقدار w را باز هم افزایش می‌دهیم و با پنج مرتبه افزایش دیگر به عدد ۴/۲ می‌رسیم؛ ولی الگوریتم نمی‌تواند برداری پیدا کند که عدم هم مثبتی $w(J-A)-J$ را نشان دهد؛ بنابراین نتیجه می‌گیریم که افزایش w نتیجه بخش نیست. اولین عدد طبیعی برای w که الگوریتم نتوانست هم مثبتی $w(J-A)-J$ را برای آن رد کند عدد $w=3$ است که نشان می‌دهد این مقدار جواب بهینه مساله است. جواب بهینه‌ی مساله نشان می‌دهد که سه

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]

کارمند با هم ارتباط دارند و این بزرگ‌ترین زیرگراف کامل در بین این کارمندان است. حال با بررسی بین کارمندان می‌توان این مجموعه را یافت.

مثال ۳ فرض کنید گراف رابطه ای خاص در شرکتی با ۱۲ کارمند به صورت گراف شکل ۳ است. برای یافتن بیش‌ترین تعداد افرادی که همه آن‌ها دوبه دو با یکدیگر در ارتباط هستند از الگوریتم ژنتیک استفاده کرده و جدول ۵ با جواب بهینه ۴ را به‌دست می‌آوریم. جواب بهینه‌ی مساله نشان می‌دهد که چهار کارمند با هم ارتباط دارند و این بزرگ‌ترین زیرگراف کامل در بین این کارمندان است. حال با بررسی بین کارمندان می‌توان این مجموعه را یافت.



شکل ۳. گراف مثال ۳

جدول ۵. یافتن جواب بهینه مثال ۳ با الگوریتم ژنتیک

زمان (ثانیه)	تعداد تکرار	w	زمان (ثانیه)	تعداد تکرار	w
>۶۰	>۱۰۰۰	۴/۴	۲/۷۸۰	۵	۳
>۶۰	>۱۰۰۰	۴/۶	۴/۰۴۸	۲	۳/۲
>۶۰	>۱۰۰۰	۵/۰	۸/۷۸۰	۱۲۰	۳/۸
>۶۰	>۱۰۰۰	۵/۶	۱۲/۰۴۸	۱۳۱	۳/۹
>۶۰	>۱۰۰۰	۶	>۶۰	>۱۰۰۰	۴/۰

جدول ٦. یافتن جواب بهینه مثال۲ با الگوریتم (PS)

زمان (ثانیه)	تعداد تکرار	w
۰/۱۰۰	۳	۲
۱/۱۰۱	۱۰	۲/۶
۱۰/۲۸	۳۲	۲/۸
۵۳/۳۳	۹۹	۲/۹
>۶۰	>۱۰۰۰	۳/۰
>۶۰	>۱۰۰۰	۳/۱
>۶۰	>۱۰۰۰	۳/۴
>۶۰	>۱۰۰۰	۳/۶
>۶۰	>۱۰۰۰	۴/۰
>۶۰	>۱۰۰۰	۴/۲

جدول ۷. یافتن جواب بهینه مثال ۳ با الگوریتم (PS)

زمان (ثانیه)	تعداد تکرار	w
۱۰/۵۴	۱۸	۳
۱۵/۰۱	۲۰	۳/۲
۵۸/۱۸	۱۵۰	۳/۴
۶۳/۳۲	۱۸۷	۳/۹
>۶۰	>۱۰۰۰	۴/۰
>۶۰	>۱۰۰۰	۴/۴
>۶۰	>۱۰۰۰	۴/۶
>۶۰	>۱۰۰۰	۵/۰
>۶۰	>۱۰۰۰	۵/۶
>۶۰	>۱۰۰۰	۶

نتیجه استفاده از الگوریتم ازدحام ذرات (PS) برای دو مثال قبل در جدول های ۶ و ۷ آمده است. همان‌طور که

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]

مشاهده می‌کنید الگوریتم ژنتیک نسبت به الگوریتم ازدحام ذرات سرعت بالاتری دارد.

۵ نتیجه‌گیری

در این مقاله دو مساله مهم در مدیریت منابع انسانی سازمان‌ها و شبکه‌های اجتماعی را در نظر گرفتیم. یک مساله تعیین حداکثر تعداد کارکنان سازمان که برای یک رابطه معین هیچ دو نفر از آن‌ها این رابطه را ندارند. مساله دیگر تعیین حداکثر تعداد کارکنان که هر دو نفر آن‌ها این رابطه معین را با یکدیگر دارند. برای حل این مسایل، ابتدا از گراف‌ها برای بیان مساله کمک گرفتیم، سپس با استفاده از روش مطرح شده در [۸] این دو مساله را مدل‌سازی کردیم. سپس آن‌ها را با استفاده از الگوریتم‌های ژنتیک و ازدحام ذرات حل کردیم. کاربرد مدل‌های مطرح شده در [۸] دقت کار را بالا برده و کاربرد دو الگوریتم مذکور، باعث کم شدن تعداد تکرارها و بالا رفتن سرعت رسیدن به جواب شد. به طوری که تعداد تکرارهای لازم برای رسیدن به جواب، از روش ارایه شده در [۹] که بهبودی بر روش [۸] است نیز بسیار کمتر است. هر چند الگوریتم ازدحام ذرات معمولا در مسایل پیوسته سریع‌تر از الگوریتم ژنتیک به جواب می‌رسد ولی در مسایل مورد بررسی در این مقاله، به دلیل ماهیت گسسته آن، الگوریتم ژنتیک عملکرد بهتری نشان داد.

منابع

[1] Anand, G., Rambabu Kodali, (2010) A mathematical model for the evaluation of roles and esponsibilities of Human Resources in a Lean Manufacturing environment, International Journal of Human Resources Development and Management (IJHRDM), 10 (1), 63-100.

[2] Farahani, R. Z. (Ed.). (2012). Graph Theory for Operations Research and Management: Applications in Industrial Engineering: Applications in Industrial Engineering. IGI Global.

[3] Harary, F. (1959). Graph theoretic methods in the management sciences. Management science, 5(4), 387-403

[4] Bazgan, C., Luchian, H. (1995). A Genetic Algorithm for the Maximal Clique Problem. In Artificial Neural Nets and Genetic Algorithms (pp. 499-502). Springer, Vienna.

[5] Bhasin, H., Mahajan, R. (2012). Genetic algorithms based solution to maximum clique problem. International Journal on Computer Science and Engineering, 4(8), 1443.

[6] Fenet, S., Solnon, C. (2003, April). Searching for maximum cliques with ant colony optimization. In Workshops on Applications of Evolutionary Computation (pp. 236-245). Springer, Berlin, Heidelberg.

[7] Soleimani-Pouri, M., Rezvanian, A., Meybodi, M. R. (2014). An ant based particle swarm optimization algorithm for maximum clique problem in social networks. In State of the art applications of social network analysis (pp. 295-304). Springer, Cham.

[8] Sponsel, J., Bundfuss, S., Dür, M. (2012). An improved algorithm to test copositivity. Journal of Global Optimization, 52(3), 537-551.

[9] Tanaka, A., Yoshise, A. (2015). An LP-based algorithm to test copositivity. Pacific Journal of Optimization, 11(1), 101-120.

[10] Burer, S. (2009). On the copositive representation of binary and continuous nonconvex quadratic programs. Mathematical Programming, 120(2), 479-495.

[11] Bundfuss, S., Dür, M. (2009). An adaptive linear approximation algorithm for copositive programs. SIAM Journal on Optimization, 20(1), 30-53.

[12] Alper Yıldırım, E. (2012). On the accuracy of uniform polyhedral approximations of the copositive cone. Optimization methods and software, 27(1), 155-173.

[13] Safi, M., Nabavi, S.S. (2019). An Improved Method of Yildirim's Algorithm for Copositive Programming, Optimization methods and software, 21(1) 9939-23.

[14] De Klerk, E., Pasechnik, D. V. (2002). Approximation of the stability number of a graph via copositive programming. SIAM Journal on Optimization, 12(4), 875-892.

[Downloaded from mail.jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2019.16.2.8.9]