

## تعیین محدوده‌نهایی معادن روباز با استفاده از الگوریتم زنبور عسل

ابراهیم نوروزی قالینی<sup>۱</sup>، مجید عطایی پور<sup>۲\*</sup>

<sup>۱</sup>کارشناس ارشد استخراج معدن، دانشگاه صنعتی امیرکبیر، Ebrahim.noroozi@aut.ac.ir  
<sup>۲</sup>دانشیار دانشکده مهندسی معدن و متالورژی، دانشگاه صنعتی امیرکبیر، map60@aut.ac.ir

(دریافت: ۱۳۹۷-۰۲-۳۰، پذیرش: ۱۳۹۷-۰۹-۱۴)

### چکیده

در گذشته برای تعیین محدوده نهایی معدن از روش‌های گوناگونی استفاده شده است. از جمله این روش‌ها می‌توان به روش‌های مبتنی بر هوش مصنوعی مانند الگوریتم‌های فراکاوشی ژنتیک، مورچگان و رقابت استعماری اشاره کرد. الگوریتم زنبور عسل یکی از الگوریتم‌های قدرتمند فراکاوشی است که از زندگی جمعی زنبورها الهام گرفته شده است. در این مقاله ابتدا یک مثال فرضی از زندگی زنبورها برای یافتن منبعی با بیشترین میزان شهد با این الگوریتم شرح داده شد، سپس روشی بر اساس الگوریتم زنبور عسل برای تعیین محدوده نهایی معدن پیشنهاد و برای بررسی عملکرد آن ابتدا یک مثال دو بعدی به صورت مرحله به مرحله توضیح داده شد. این مثال نشان داد مواردی را که مخروط شناور قادر به یافتن جواب بهینه برای تعیین محدوده نهایی معدن نیست، می‌توان با الگوریتم زنبور عسل به خوبی حل کرد. سپس از این الگوریتم برای تعیین محدوده نهایی در معدن سونگون با بلوک  $120 \times 100 \times 45$  استفاده شد. برای اعتبارسنجی پاسخ مساله تعیین محدوده نهایی با استفاده از الگوریتم زنبور عسل، از نظریه‌گراف و مخروط شناور استفاده شد. نتایج نشان می‌دهد که اختلاف نقدینگی محدوده به دست آمده از طریق الگوریتم زنبور عسل حدود  $12/3$  درصد از محدوده مخروط شناور بیشتر و تنها  $1/6$  درصد از محدوده نظریه‌گراف کمتر است.

### واژه‌های کلیدی

محدوده نهایی معدن، الگوریتم زنبور عسل، نظریه‌گراف، مخروط شناور.

## ۱- مقدمه

گراف<sup>۷</sup> [۴]، سیمپلکس دوگان<sup>۸</sup> [۵]، الگوریتم حمل و نقل<sup>۹</sup> [۶] و جریان شبکه<sup>۱۰</sup> [۷] اشاره کرد. برای رفع این محدودیت محققان از سال‌ها پیش به دنبال یافتن روش‌هایی برای حل مسایل با زمان کوتاه‌تر بوده‌اند که الگوریتم‌های جستجوگر نتیجه پژوهش‌ها در این حوزه است. با این الگوریتم‌ها مسایل در زمانی معقول و با دقت قابل قبول حل می‌شوند [۸، ۹]. الگوریتم‌های جستجوگر را می‌توان به دو دسته الگوریتم‌های کاوشی<sup>۱۱</sup> و فراکاوشی<sup>۱۲</sup> تقسیم کرد. از جمله الگوریتم‌های کاوشی می‌توان به مخروط شناور<sup>۱۳</sup> [۱۰]، مخروط شناور II [۱۱]، کروبوفا<sup>۱۴</sup> [۱۲] و کروبوفا اصلاح شده [۱۳] اشاره کرد. از جمله الگوریتم‌های فراکاوشی استفاده شده برای تعیین محدوده نهایی معدن می‌توان الگوریتم‌های ژنتیک<sup>۱۵</sup> [۱۴]، کلونی مورچگان<sup>۱۶</sup> [۱۵] و رقابت استعماری<sup>۱۷</sup> [۱۶] را نام برد.

در سال‌های اخیر استفاده از هوش مصنوعی<sup>۱۸</sup> در بسیاری از مسایل مهندسی متداول شده است. الگوریتم‌های بهینه‌سازی فراکاوشی زیر شاخه هوش مصنوعی‌اند و الگوریتم زنبور عسل یکی از الگوریتم‌های مبتنی بر هوش جمعی<sup>۱۹</sup> است. الگوریتم‌های کلونی زنبور مصنوعی<sup>۲۰</sup> [۱۷]، بهینه‌سازی کلونی زنبور<sup>۲۱</sup> [۱۸]، بهینه‌سازی ازدحام زنبور<sup>۲۲</sup> [۱۹]، زنبور مجازی<sup>۲۳</sup> [۲۰] و الگوریتم زنبورها<sup>۲۴</sup> [۲۱] از جمله الگوریتم‌های الهام گرفته از زندگی اجتماعی زنبورهای عسل‌اند. الگوریتم بهینه‌سازی جفت‌گیری زنبور عسل<sup>۲۵</sup> [۲۲] نیز بر اساس رفتار جفت‌گیری زنبورها توسعه یافته است. الگوریتم ABC اولین بار توسط کارابوگا<sup>۲۶</sup> آرایه شد [۱۷]. در این مقاله از این الگوریتم برای تعیین محدوده نهایی معدن استفاده شده است. پس از اواخر زیادی برای بهبود این الگوریتم تلاش کردند که از نسخه‌های بهبود یافته آن می‌توان به ABC-G-ABC I-ABC

روش‌های استخراج معدن به دو دسته کلی روش‌های استخراج روباز و زیرزمینی تقسیم می‌شوند. روش‌های نواری، کواری و روباز از مهم‌ترین روش‌های استخراج معادن روبازند. پس از اکتشاف یک ذخیره در صورت انتخاب روش استخراج روباز برای بهره‌برداری از آن، باید محدوده نهایی معدن<sup>۱</sup> تعیین شود. هدف از طراحی محدوده نهایی معدن تعیین پارامترهایی مانند میزان گسترش طولی، عرضی و عمقی معدن، مسیرهای دسترسی به ماده معدنی، محل دپوی باطله، محل احداث تاسیسات سطحی، نسبت باطله‌برداری، عمر معدن، میزان ذخیره قابل استخراج به روش روباز، میزان باطله‌برداری و نحوه برنامه‌ریزی تولید است.

برای تعیین محدوده نهایی معدن می‌توان از روش‌های دستی یا رایانه‌ای استفاده کرد. در روش دستی از مفهوم ارزش سربه‌سری و صفر کردن ارزش در مرزهای محدوده نهایی استفاده می‌شود. طراحی محدوده نهایی با روش‌های دستی زمان‌بر است و طراحی به این روش نیازمند مهندسین و طراحان باتجربه است. همچنین این روش تنها در کانسارهای کوچک قابل کاربرد است. روش‌های کامپیوتری بر استفاده از مدل بلوکی مبتنی است. در این روش‌ها ابتدا کانسار به بلوک‌های کوچک‌تر تقسیم شده و سپس عیار بلوک‌ها با یکی از روش‌های تخمین عیار مانند زمین آمار یا عکس فاصله تخمین زده می‌شود. پس از تهیه مدل بلوکی عیاری<sup>۲</sup> با استفاده از قیمت ماده معدنی و هزینه‌ها مدل بلوکی اقتصادی<sup>۳</sup> معدن تهیه می‌شود. در مدل بلوکی اقتصادی ارزش بلوک‌ها ممکن است مثبت، منفی یا صفر باشد و ارزش بلوک‌های هوا در بالای توپوگرافی صفر در نظر گرفته می‌شود. محدوده بهینه<sup>۴</sup> واقعی، محدوده‌ای است که سود حاصل از آن محدوده اولاً مثبت است و ثانیاً از بین مثبت‌های ممکن، بیشینه مقدار را دارد. روش‌های کامپیوتری به دو دسته کلی روش‌های مبتنی بر منطق ریاضی<sup>۵</sup> و روش‌های جستجوگر تقسیم می‌شوند. روش‌های مبتنی بر منطق ریاضی همواره قادر به یافتن محدوده بهینه‌اند اما در حل مسایل بزرگ محدودیت دارند و برای حل مسایل بزرگ به زمان بسیار طولانی نیاز دارند. از جمله روش‌های ریاضی می‌توان به برنامه‌ریزی پویا<sup>۶</sup> [۳-۱]، لچ و گروسمن مبتنی بر نظریه

7 Graph theory

8 Dual simplex

9 Transportation algorithm

10 Maximum flow

11 Heuristic

12 Meta- heuristic

13 Moving cone

14 Korobov

15 Genetic algorithm

16 Ant colony algorithm

17 Imperialist competitive algorithm

18 Artificial intelligence

19 Swarm intelligence

20 Artificial bee colony (ABC)

21 Bee colony optimization

22 Bee swarm optimization

23 Virtual beeaalgorithm

24 Bees algorithm

25 Honey-bees mating optimization

26 Karaboga

1 Ultimate pit limit

2 Grade block model

3 Economic block model

4 Optimum

5 Rigorous

6 Dynamic programming

را دارد. اعضای یک کلونی زنبور عسل ممکن است در مسافت‌های زیاد و در جهت‌های گوناگونی پخش شوند و از منابع غذایی بهره‌برداری کنند. هر کلونی زنبور از سه دسته زنبور تشکیل می‌شود که هر یک از این دسته‌ها وظایفی معینی را بر عهده دارند. زنبورهای پیش‌آهنگ<sup>۶</sup> اولین دسته‌اند که حدود ۵ تا ۱۰ درصد کلونی را تشکیل می‌دهند. وظیفه زنبورهای پیش‌آهنگ کشف منابع جدید است. این زنبورها به صورت تصادفی در محیط اطراف کندو پخش می‌شوند و به جستجوی غذا می‌پردازند. زنبورهای پیش‌آهنگ پس از برگشت به لانه در یک رقص چرخشی<sup>۷</sup> اطلاعات خود را با سایر زنبورهای درون کندو به اشتراک می‌گذارند و تعدادی از آن‌ها را برای بهره‌برداری از منبع کشف شده استخدام می‌کنند. زنبورهای کارگر<sup>۸</sup> که حدود نیمی از یک کندو را به خود اختصاص می‌دهند، دومین دسته را تشکیل می‌دهند. وظیفه این دسته از زنبورها استخراج منابع غذایی از پیش مشخص شده است. زنبورهای ناظر<sup>۹</sup> سومین دسته از زنبورهای درون یک کندو را تشکیل می‌دهند. این زنبورها در درون کندو منتظر سایر زنبورها می‌مانند و پس از تبادل اطلاعات در رقص چرخشی یک منبع غذایی را که غنی‌تر از سایر منابع است، انتخاب و از آن بهره‌برداری می‌کنند. این زنبورها بر اساس شایستگی منابع غذایی به آن‌ها اختصاص می‌یابند و منابع شایسته‌تر زنبور بیشتری را جذب می‌کنند. به بیان ساده‌تر زنبور عسل ناظر زنبوری است که در منطقه رقص منتظر سایر زنبورها برای کسب اطلاعات در مورد منابع غذایی می‌ماند و در نهایت یک منبع را انتخاب می‌کند. زنبور عسل کارگر زنبوری است که به طرف منابع غذایی از پیش مشخص شده می‌رود و زنبور عسل پیش‌آهنگ زنبوری است که به صورت تصادفی به جستجوی غذا می‌پردازد.

### ۳- الگوریتم زنبور عسل

در الگوریتم زنبور عسل، ابتدا نیمی از جمعیت زنبورها کارگر و نیم دیگر غیر کارگرند. برای هر منبع غذایی فقط یک زنبور عسل کارگر وجود دارد، بنابراین تعداد زنبورهای کارگر با تعداد منابع غذایی اطراف کندو برابر است. زنبورهای کارگری که منابع آن‌ها به وسیله زنبورهای ناظر و دیگر زنبورهای کارگر تهیه شده باشد به زنبورهای پیش‌آهنگ تبدیل می‌شوند. گام‌های اصلی الگوریتم زنبور به شرح زیر است [۴۲]:

SA, PABC, CABC اشاره کرد [۲۳-۲۷]. برخی از نویسندگان برای افزایش کارایی الگوریتم زنبور عسل از تلفیق آن با سایر الگوریتم‌ها استفاده کرده‌اند. از جمله این تلاش‌ها می‌توان به تلفیق الگوریتم زنبور عسل با الگوریتم‌های سیمپلکس<sup>۱</sup> [۲۸]، ژنتیک [۲۹]، مورچگان [۳۰] و تکامل تفاضلی<sup>۲</sup> [۳۱] اشاره کرد. همچنین تلاش‌هایی برای افزایش سرعت همگرایی<sup>۳</sup> الگوریتم در حل مسایل بزرگ انجام شده است [۳۲, ۳۳]. بر اساس یک تحقیق در سال ۲۰۱۴ از بین مقالات ارایه شده در مورد الگوریتم زنبور، ۵۸ درصد موارد با الگوریتم کلونی زنبور مصنوعی سروکار داشته‌اند [۳۴].

از الگوریتم زنبور عسل در حل بسیاری از مسایل مهندسی، صنعتی و ریاضی مانند بهینه‌سازی مکان حفر چاه‌ها در مخازن نفتی [۳۵]، بهینه‌سازی تخلیه آب از سدها [۳۶]، خوشه‌بندی داده‌ها [۳۷]، زمان‌بندی کار ماشین‌ها [۳۸]، خطایابی تصادفی نیروگاه اتمی [۳۹] و از تلفیق آن با شبکه‌های عصبی مصنوعی در پیش‌بینی فشار انتهای چاه بهمراه شبکه [۴۰] استفاده شده است. از جمله معهود موارد کاربرد این الگوریتم در معدن، پیش‌بینی و بهینه‌سازی عقب‌زدگی<sup>۴</sup> ناشی از انفجار در معادناست [۴۱].

در این مقاله کاربرد الگوریتم زنبور عسل برای حل مساله تعیین محدوده‌نهایی تشریح شده است. پس از شرح کلی بر الگوریتم زنبور عسل و حل یک مثال فرضی از زندگی زنبورها با استفاده از آن، مراحل الگوریتم برای حل مساله محدوده‌نهایی تشریح و یک روندنما<sup>۵</sup> برای تعیین محدوده‌نهایی معدن ارایه شده است. سپس روند حل الگوریتم بر روی یک مدل دو بعدی نشان داده شده و برای اعتبارسنجی عملکرد الگوریتم در مقیاس واقعی، الگوریتم در محیط نرم‌افزار Matlab بر روی یک معدن سونگون پیاده‌سازی و نتایج آن با نتایج به دست آمده از نظریه گراف مقایسه شده است.

### ۲- زندگی زنبورها در طبیعت

زنبورها به صورت جمعی زندگی می‌کنند. هر یک از این حشرات به تنهایی یک جزو ساده است که در کنار هم یک کلونی زنبور را تشکیل می‌دهند که رفتاری منسجم و پیچیده دارد. این رفتار منسجم و پیچیده باعث ایجاد سیستمی یکپارچه می‌شود که توانایی کشف و بهره‌برداری از شهد گل‌ها

<sup>1</sup>simplex

<sup>2</sup>Differential evolution

<sup>3</sup>Convergence

<sup>4</sup>Back break

<sup>5</sup>Flowchart

<sup>6</sup>Scout bee

<sup>7</sup>Waggle dance

<sup>8</sup>Employed bee

<sup>9</sup>Onlooker bee

عدم بهبود جواب در چرخه است. یک زنبور به صورت تصادفی یک اصلاح بر روی مکان منبع غذایی (جواب) موجود در حافظه خود ایجاد می‌کند و میزان شهد (مقدار شایستگی) منبع جدید (جواب جدید) را تحت آزمایش قرار می‌دهد. در مورد زنبورهای واقعی، برای تولید جواب جدید از اطلاعات موجود و بصری زنبور استفاده می‌شود اما در مورد زنبورهای مصنوعی از هر اطلاعاتی برای مقایسه استفاده نمی‌شود و بهبود فقط به صورت تصادفی با استفاده از رابطه ۲ و یکی دیگر از جواب‌ها انجام می‌گیرد. در صورتی که مقدار شهد منبع جدید بیشتر از منبع قبلی ذخیره شده در حافظه زنبور عسل باشد، موقعیت جدید حفظ و موقعیت قبلی فراموش و در غیر این صورت موقعیت قبلی نگهداری می‌شود. زنبورهای کارگر پس از تکمیل فرآیند جستجو، اطلاعات شهد منابع غذایی (راه‌حل) و اطلاعات مربوط به موقعیت آن‌ها را با زنبورهای ناظر در منطقه رقص به اشتراک می‌گذارند. یک زنبور ناظر با ارزیابی اطلاعات شهد گرفته شده از همه زنبورهای کارگر، یک منبع غذایی را با احتمال مربوط به مقدار شهد آن منبع انتخاب می‌کند. زنبورکارگر نیز اصلاحی را در موقعیت (راه‌حل) موجود در حافظه خود انجام می‌دهد و مقدار شهد منبع داوطلب (راه‌حل) را بررسی می‌کند. اگر میزان شهد بیشتر از منبع قبلی باشد آن را جایگزین و منبع پیشین را فراموش می‌کند. زنبور ناظر یک منبع غذایی را با توجه به مقدار احتمال مرتبط با آن منبع غذایی که از رابطه ۱ به دست می‌آید، انتخاب می‌کند [۴۲].

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (1)$$

که در آن:

$p_i$  احتمال انتخاب زنبور  $i$

$fit_i$  میزان شایستگی راه‌حل  $i$

$SN$  تعداد منابع غذایی

در این الگوریتم، زنبورهای کارگر اطلاعات خود را با زنبورهای ناظر به اشتراک می‌گذارند و زنبورهای ناظر با استفاده از این اطلاعات یک منبع غذایی جدید را در اطراف منبع غذایی قبل انتخاب می‌کنند و برای تولید منبع غذایی جدید در اطراف منبع فعلی از رابطه ۲ استفاده می‌شود [۴۲].

$$v_i = x_i + \varphi_i (x_i - x_k) \quad k \in \{1, 2, \dots, BN\} \& k \neq i \quad (2)$$

که در آن:

$v_i$  جواب جدیدی در اطراف جواب  $x_i$

$x_i$  جواب  $i$  برای مساله بهینه‌سازی

- مقداردهی اولیه
- تکرار (تا رسیدن به وضعیت دلخواه)
- ارسال زنبورهای کارگر به منابع غذایی
- ارسال زنبورهای ناظر به منابع غذایی
- ارسال زنبورهای پیش‌آهنگ برای جستجوی منابع غذایی جدید

در مرحله مقداردهی اولیه مجموعه‌ای از منابع غذایی به صورت تصادفی به وسیله زنبورهای کارگر انتخاب و مقدار شهد آن‌ها اندازه‌گیری می‌شود. زنبورهای کارگر پس از برگشت به کندو، اطلاعات حاصل از منابع غذایی را در منطقه رقص اطراف کندو با زنبورهای ناظر به اشتراک می‌گذارند. در مرحله دوم هر زنبور کارگر پس از تبادل اطلاعات، به منبع غذایی که در چرخه قبل مشاهده کرده است باز می‌گردد و یک منبع غذایی جدید را در اطراف منبع غذایی موجود انتخاب می‌کند. در مرحله سوم زنبورهای ناظر یک منبع غذایی را بر اساس اطلاعات به اشتراک گذاشته شده به وسیله زنبورهای کارگر در منطقه رقص، انتخاب می‌کنند. با افزایش میزان شهد منبع غذایی، احتمال انتخاب این منبع، توسط زنبورهای ناظر نیز افزایش می‌یابد. بنابراین زنبورهای رقصان با میزان شهد بیشتر، زنبورهای ناظر بیشتری را برای بهره‌برداری از آن منابع استخدام می‌کنند. یک زنبور ناظر پس از انتخاب یک منطقه برای بهره‌برداری، به سوی آن حرکت می‌کند و پس از رسیدن به منطقه بر اساس اطلاعات بصری یک منبع غذایی جدید را در اطراف آن منبع انتخاب می‌کند. اطلاعات بصری بر مقایسه مکان منابع غذایی مبتنی است. در مرحله چهارم منابع غذایی که شهد آن‌ها به وسیله سایر زنبورها تهی شده است، به صورت تصادفی توسط زنبورهای جستجوگر جایگزین می‌شوند. در این الگوریتم تعداد زنبورهای کارگر و زنبورهای ناظر در یک کندو برابر خواهد بود. در الگوریتم زنبور عسل، مکان منبع غذایی نشان‌دهنده یک جواب ممکن برای مساله بهینه‌سازی است، همچنین میزان شهد منبع غذایی نشان‌دهنده کیفیت (شایستگی) آن است. تعداد زنبورهای کارگر و یا ناظر برابر با تعداد جمعیت راه‌حل‌ها است. در مرحله ابتدایی تعداد  $SN$  راه‌حل به صورت تصادفی در فضای جواب تولید می‌شود. هر راه‌حل یک (منبع غذایی)  $x_i = (1, 2, \dots, SN)$  است که یک بردار  $D$  بعدی است، و  $D$  تعداد پارامترهای بهینه‌سازی است. پس از مقداردهی اولیه چرخه تکرارها شروع می‌شود. از مقداردهی اولیه چرخه تکرارها شروع می‌شود.  $C = (1, 2, \dots, C_{max})$  فرآیند چرخه برای زنبورها است،  $C$  تعداد

- یک فرآیند انتخاب تصادفی که به وسیله زنبور پیش‌آهنگ انجام می‌شود.

در الگوریتم زنبور عسل سه پارامتر کنترلی تعداد منابع غذایی ( $SN$ ) که با تعداد زنبورهای کارگر یا زنبورهای ناظر برابر است، تعداد عدم موفقیت در بهبود جواب ( $limit$ ) و حداکثر تعداد چرخه ( $MCN^1$ ) وجود دارد.

در مورد زنبور عسل واقعی سرعت استخدام زنبورهای ناظر پارامتری برای نشان دادن سرعت کشف و بهره‌برداری از منابع غذایی جدید است. به طور مشابه استخدام زنبورهای مصنوعی جدید سرعت اکتشاف راه‌حل‌های ممکن و با کیفیت خوب را نشان می‌دهد. بقا و پیشرفت کلونی زنبور عسل به کشف سریع و استفاده کارآمد از بهترین منابع غذایی بستگی دارد. همچنین، راه حل درست مسایل دشوار مهندسی، به‌ویژه مسایلی که باید در زمان واقعی حل شود، به کشف سریع راه‌حل‌های خوب وابسته است. در یک فرآیند جستجوی قوی، فرآیندهای اکتشاف و بهره‌برداری باید باهم انجام شود. در الگوریتم زنبور عسل، در حالی که زنبورهای کارگر و ناظر فرآیند بهره‌برداری در فضای جستجو را انجام می‌دهند، زنبورهای پیش‌آهنگ فرآیند اکتشاف را کنترل می‌کنند.

### ۳-۱- مثال

برای تشریح بیشتر الگوریتم زنبور عسل، یک مثال از زندگی زنبورها با استفاده از این الگوریتم حل می‌شود. در این مساله هدف یافتن نقطه‌ای از طبیعت با بیشینه میزان شهد استفاده شده است. به عنوان مثال در یک محدوده انتخاب شده از طبیعت، میزان شهد گل‌ها با استفاده از رابطه ۳ قابل محاسبه است. در این مثال محدوده جستجو  $\alpha = [0,3]$  و  $\beta = [0,5]$  در نظر گرفته شده و نمودار میزان شهد در این بازه در شکل ۱ نشان داده شده است. حل این مساله به صورت مرحله به مرحله مطابق با آنچه در الگوریتم توضیح داده شد، ارایه می‌شود.

$$V = -\sin(\alpha) \cdot \cos(\beta) \quad (3)$$

که در آن:

$\alpha$  عددی برحسب رادیان در بازه  $[0,3]$

$\beta$  عددی برحسب رادیان در بازه  $[0,5]$

$V$  میزان شهد هر منبع مفروض

$K$  شماره یکی دیگر از جواب‌ها که به صورت تصادفی انتخاب می‌شود و مقدار آن باید از  $i$  متمایز باشد.

$\varphi_i$  یک بردار  $D$  عضوی است که هر کدام از درایه‌های آن عددی تصادفی بین  $[-1, 1]$  است و تولید موقعیت منبع غذایی همسایه در اطراف  $x_i$  را کنترل می‌کند.

$x_k$  جوابی تصادفی از بین  $SN$  جواب برای مساله که با استفاده از آن جوابی جدید در اطراف  $x_i$  ایجاد می‌شود.

رابطه ۲ نشان می‌دهد با کاهش تفاوت بین  $x_i$  و  $x_k$  تغییرات در موقعیت  $x_i$  کاهش می‌یابد. با توجه به مکانیزم انتخاب حریصانه (انتخاب جواب با ارزش بیشتر از بین دو جواب) به تدریج جواب‌ها به بیشینه ارزش همگرا شده و فاصله بین جواب‌ها کاهش پیدا می‌کند.

منبع غذایی که شهد آن به وسیله زنبورها تهی شده است، توسط زنبورهای پیش‌آهنگ با یک منبع ماده غذایی جدید جایگزین می‌شود. در الگوریتم زنبور عسل این عمل با تولید جواب جدید به صورت تصادفی شبیه‌سازی شده و جایگزین منبع تهی شده می‌شود. در این الگوریتم اگر یک جواب پس از یک تعداد تکرار از پیش تعیین شده بهبود نیابد، به عنوان منبع غذایی ترک شده تلقی می‌شود.

پس از ایجاد هر جواب جدید  $v_i$  شایستگی این جواب توسط زنبور مصنوعی محاسبه شده و اگر شایستگی آن برابر یا بیشتر از جواب  $x_i$  باشد، جایگزین آن می‌شود. به عبارت دیگر، عمل انتخاب بین منابع غذایی قبلی و فعلی با یک مکانیزم انتخاب حریصانه انجام می‌شود. الگوریتم زنبور عسل در حقیقت چهار فرآیند انتخاب مختلف را به کار می‌گیرد:

- یک فرآیند انتخاب سراسری که به وسیله زنبور ناظر برای کشف مناطق امیدبخش از آن استفاده می‌شود، این فرآیند با رابطه ۱ توضیح داده می‌شود.

- یک فرآیند انتخاب محلی بر اساس اطلاعات آن ناحیه برای زنبورهای کارگر و ناظر (برای زنبورهای واقعی اطلاعاتی مانند رنگ، شکل و عطر گل) که از آن برای تعیین یک منبع غذایی در اطراف منبع غذایی قبل استفاده می‌شود و بر اساس رابطه ۲ بیان می‌شود.

- فرآیند انتخاب حریصانه محلی که به وسیله تمام زنبورها انجام می‌شود و طی آن اگر مقدار شهد منبع کاندیدا بیشتر از منبع فعلی باشد، زنبور منبع فعلی را فراموش کرده و منبع کاندیدا را حفظ می‌کند و در غیر این صورت، زنبور منبع فعلی را در حافظه نگه می‌دارد.

<sup>1</sup> Maximum cycle number

مقاله برای ضرب درایه‌های متناظر از عملگر استفاده شده است.

$$V_1 = \begin{pmatrix} 1.5 \\ 0.25 \end{pmatrix} + \begin{pmatrix} 0.15 \\ -0.75 \end{pmatrix} \times \left( \begin{pmatrix} 1.5 \\ 0.25 \end{pmatrix} - \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix} \right) = \begin{pmatrix} 1.35 \\ 3.27 \end{pmatrix}$$

$$V = -\sin(1.35) \cdot \cos(3.27) = 0.9677$$

چون میزان شهد منبع ( $v_1$ ) بیشتر از منبع قبل ( $x_1$ ) است منبع جدید جایگزین منبع قبل می‌شود. بنابراین:

$$x_1 = \begin{pmatrix} 1.35 \\ 3.27 \end{pmatrix} V_1 = 0.9677$$

$$C_1 = 0$$

برای منبع دوم نیز به همین شکل عمل می‌شود:

$$V_2 = \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix} + \begin{pmatrix} -0.32 \\ -0.61 \end{pmatrix} \times \left( \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix} - \begin{pmatrix} 0.72 \\ 3.41 \end{pmatrix} \right) = \begin{pmatrix} 1.93 \\ 3.84 \end{pmatrix}$$

$$V = -\sin(1.93) \cdot \cos(3.84) = 0.7170$$

چون میزان شهد منبع جدید ( $v_2$ ) بیشتر از منبع قبل ( $x_2$ ) است منبع جدید جایگزین منبع قبل می‌شود. بنابراین:

$$x_2 = \begin{pmatrix} 1.93 \\ 3.84 \end{pmatrix} V_2 = 0.7170 \quad C_2 = 0$$

همچنین برای منبع سوم:

$$V_3 = \begin{pmatrix} 0.72 \\ 3.41 \end{pmatrix} + \begin{pmatrix} 0.22 \\ 0.94 \end{pmatrix} \times \left( \begin{pmatrix} 0.72 \\ 3.41 \end{pmatrix} - \begin{pmatrix} 1.93 \\ 3.84 \end{pmatrix} \right) = \begin{pmatrix} 0.45 \\ 3 \end{pmatrix}$$

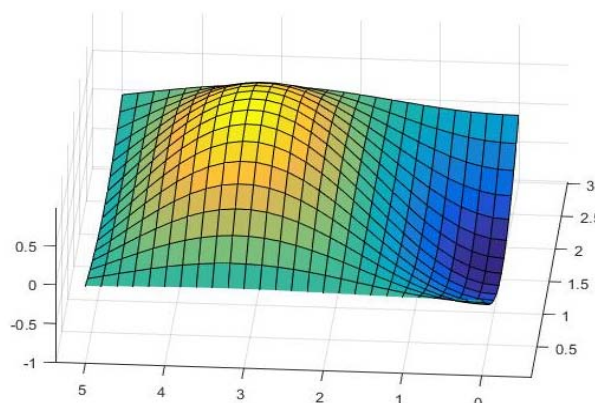
$$V = -\sin(0.45) \cdot \cos(3) = 0.4306$$

به دلیل عدم بهبود ارزش منبع جدید، این منبع جایگزین منبع قبل نمی‌شود. همچنین این منبع غذایی جریمه شده و یک واحد به شمارنده عدم افزایش شهد منبع افزودده می‌شود. بنابراین:

$$x_2 = \begin{pmatrix} 0.72 \\ 3.41 \end{pmatrix} V_2 = 0.6358 \quad C_2 = 1$$

### ۳-۱-۳- مرحله سوم

در این مرحله احتمال دریافت زنبور به وسیله هر منبع، با استفاده از رابطه ۱ محاسبه می‌شود. برای محاسبه احتمال دریافت زنبور ابتدا لازم است میزان شایستگی منبع با استفاده از رابطه ۴ به دست آید. با توجه به میزان شهد و با محاسبه احتمال دریافت زنبور، به تعدادی از منابع زنبور اختصاص داده می‌شود. این بدین معنا است که با توجه به احتمال انتخاب هر کدام از منابع، یکی از آن‌ها انتخاب می‌شود. اگر میزان شهد یک منبع بیشتر باشد احتمال انتخاب آن به وسیله زنبورهای ناظر بیشتر است. در این مرحله ممکن است تمام زنبورهای



شکل ۱- نمودار پراکندگی شهد بر روی محدوده.

### ۳-۱-۱- مرحله اول

در مرحله اول منابع اولیه ایجاد می‌شود. منابع اولیه مختصات نقاطی تصادفی بر روی محدوده مورد بررسی است. بنابراین هر منبع اولیه برداری با دو مولفه مختصاتی در جهت افقی ( $\alpha$ ) و عمودی ( $\beta$ ) است. برای این مثال سه منبع اولیه در نظر گرفته می‌شود که هر کدام از آن‌ها نقطه‌ای را بر روی سطح زمین و در درون محدوده مورد بررسی بازنمایی می‌کنند. برای این سه منبع اولیه به شرح روابط ۴ تا ۶ است:

$$x_1 = \begin{pmatrix} 1.5 \\ 0.25 \end{pmatrix} V_1 = -\sin(1.5) \cdot \cos(0.25) = -9665$$

$$x_2 = \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix} V_2 = -\sin(2.5) \cdot \cos(4.5) = 0.2670$$

$$x_3 = \begin{pmatrix} 0.72 \\ 3.41 \end{pmatrix} V_3 = -\sin(0.72) \cdot \cos(3.41) = 0.6358$$

### ۳-۱-۲- مرحله دوم

به تعداد منابع، زنبورهای کارگر وجود دارد. در این مرحله هر یک از این زنبورهای کارگر منبع غذایی جدیدی را در اطراف منبع غذایی موجود جستجو می‌کند. این عمل در الگوریتم با استفاده از رابطه ۲ به صورت تصادفی انجام می‌شود. پس از کشف منبع جدید میزان شهد این منبع محاسبه می‌شود. اگر میزان شهد منبع جدید بیش از منبع پیشین باشد، منبع جدید جایگزین جواب قبل می‌شود. اگر میزان شهد این منبع کمتر باشد، منبع باید جریمه شود. جریمه بدین معنی است که یک واحد به شمارنده  $C$  که برای هر منبع غذایی ایجاد شده است، اضافه می‌شود. اگر شمارنده جریمه یک منبع غذایی از حد از پیش تعیین شده (در این مثال ۵) فراتر رود، این منبع با منبعی تصادفی در محدوده جواب جایگزین می‌شود. در این

$$v = -\sin(1.80) \cdot \cos(3.06) = 0.9704$$

به دلیل افزایش میزان شهد منبع جدید این منبع جایگزین منبع قبل می‌شود. بنابراین:

$$x_1 \begin{pmatrix} 1.80 \\ 3.06 \end{pmatrix} v_1 = 0.9704 \rightarrow c_1 = 0$$

دومین منبع غذایی در اطراف منبع ۳ ایجاد خواهد شد. زنبور ناظر به صورت تصادفی منبع غذایی دو را انتخاب کرده و با استفاده از آن در اطراف منبع غذایی سه، یک منبع جدید را برای بهره‌برداری انتخاب می‌کند. در این صورت:

$$v_3 = \begin{pmatrix} 0.72 \\ 3.41 \end{pmatrix} + \begin{pmatrix} -0.92 \\ 0.34 \end{pmatrix} \cdot \left( \begin{pmatrix} 0.72 \\ 3.41 \end{pmatrix} - \begin{pmatrix} 1.80 \\ 3.06 \end{pmatrix} \right) = \begin{pmatrix} 1.71 \\ 3.53 \end{pmatrix}$$

$$V = -\sin(1.71) \cdot \cos(3.53) = 0.9165$$

به دلیل بهبود جواب این منبع جایگزین منبع قبل می‌شود. بنابراین:

$$x_3 = \begin{pmatrix} 1.71 \\ 3.53 \end{pmatrix} v_3 = 0.9165 \rightarrow c_2 = 0$$

سومین منبع غذایی نیز در اطراف جواب ۳ ایجاد می‌شود. اگر زنبور ناظر با استفاده از منبع غذایی ۲، منبع غذایی جدید را در اطراف منبع غذایی ۳ انتخاب کند. در این صورت:

$$v_3 = \begin{pmatrix} 1.71 \\ 3.53 \end{pmatrix} + \begin{pmatrix} 0.41 \\ 0.18 \end{pmatrix} \cdot \left( \begin{pmatrix} 1.71 \\ 3.53 \end{pmatrix} - \begin{pmatrix} 1.93 \\ 3.84 \end{pmatrix} \right) = \begin{pmatrix} 1.62 \\ 3.47 \end{pmatrix}$$

$$v = -\sin(1.62) \cdot \cos(3.47) = 0.9441$$

به دلیل بهبود جواب، این منبع جایگزین منبع قبل می‌شود. بنابراین:

$$x_3 = \begin{pmatrix} 1.62 \\ 3.47 \end{pmatrix} \rightarrow v_3 = 0.9441 \rightarrow c_3 = 1$$

پس از ایجاد سه منبع غذایی جدید این مرحله تمام می‌شود.

### ۳-۱-۴- مرحله چهارم

در مرحله چهارم منابع غذایی تهی شده که تعداد عدم افزایش شهد در اطراف آن‌ها به  $C_{max}$  رسیده است، با منابع غذایی تصادفی جدید جایگزین می‌شوند. در این مرحله شرط پایان الگوریتم نیز بررسی می‌شود. در این مثال شرط پایان الگوریتم ۱۰۰ تکرار در نظر گرفته شده است. الگوریتم در صورت ارضای شرط پایان متوقف می‌شود و در غیر این صورت به مرحله دوم باز می‌گردد. برای رسیدن به بیشینه میزان شهد

ناظر با توجه به میزان شهد به یک منبع اختصاص یابد. سپس زنبورها با استفاده از رابطه ۲ و یکی دیگر از منابع، یک منبع جدید را در اطراف منبع مورد نظر انتخاب می‌کنند. اگر این منبع نسبت به منبع قبل شهد بیشتری داشته باشد، جایگزین منبع قبل می‌شود و در غیر این صورت به این منبع جریمه تعلق می‌گیرد.

برای تعیین شایستگی جواب‌های این مساله رابطه ۴ پیشنهاد شده است. بنابراین شایستگی هر منبع با استفاده از رابطه ۴ محاسبه می‌شود:

$$fit_i = V_i + 1 \quad (4)$$

$$fit_1 = value_1 + 1 = 0.9677 + 1 = 1.9677$$

$$fit_2 = value_2 + 1 = 0.7170 + 1 = 1.7170$$

$$fit_3 = value_3 + 1 = 0.6358 + 1 = 1.6358$$

پس از محاسبه شایستگی هر منبع، احتمال انتخاب آن به وسیله زنبور ناظر نیز محاسبه می‌شود:

$$p_1 = \frac{fit_1}{\sum_{n=1}^3 fit_n} = \frac{1.9677}{1.9677 + 1.7170 + 1.6358} = 0.37$$

$$p_2 = \frac{fit_2}{\sum_{n=1}^3 fit_n} = \frac{1.7170}{1.9677 + 1.7170 + 1.6358} = 0.32$$

$$p_3 = \frac{fit_3}{\sum_{n=1}^3 fit_n} = \frac{1.6358}{1.9677 + 1.7170 + 1.6358} = 0.31$$

اکنون برای انتخاب منبع به وسیله زنبور از احتمال تجمعی منابع استفاده می‌شود. اگر احتمال تجمعی با  $P$  نشان داده شود، در این صورت  $P_1$ ،  $P_2$  و  $P_3$  به شرح زیر است:

$$p_1 = 0.37 \rightarrow p_2 = 0.69 \rightarrow p_3 = 1$$

برای انتخاب منابع به وسیله زنبورها، به تعداد منابع غذایی موجود، اعداد تصادفی بین صفر و ۱ تولید می‌شود. اگر سه عدد تصادفی ۰٫۸۵، ۰٫۹۳ و ۰٫۱۲ تولید شده باشد. یکی از این اعداد به احتمال تجمعی منبع ۱ و دو عدد دیگر به احتمال تجمعی منبع ۳ نزدیک‌تر است. بنابراین با انتخاب منابع ۱ و ۳ به وسیله زنبورها، در اطراف منبع ۱ یک منبع جدید و در اطراف منبع ۳ دو منبع جدید انتخاب می‌شود. اگر در اطراف منبع ۱ به صورت تصادفی منبع ۲ به عنوان منبع جدید انتخاب شده باشد. در این صورت:

$$v_1 = \begin{pmatrix} 1.35 \\ 3.27 \end{pmatrix} + \begin{pmatrix} -0.78 \\ 0.36 \end{pmatrix} \cdot \left( \begin{pmatrix} 1.35 \\ 3.27 \end{pmatrix} - \begin{pmatrix} 1.93 \\ 3.84 \end{pmatrix} \right) = \begin{pmatrix} 1.80 \\ 3.06 \end{pmatrix}$$

صفر را بپذیرد. هر یک از بلوک‌ها ممکن است درون محدوده نهایی معدن باشند که در این صورت درایه آن‌ها عدد یک، یا خارج از محدوده نهایی معدن باشند که در این صورت درایه مربوط به آن‌ها عدد صفر خواهد بود. همچنین برای هر جواب یک ماتریس صفر  $(A)L \times M$  (تعداد سطرها و  $M$  تعداد ستون‌های مدل بلوکی) ایجاد می‌شود. درایه‌های این ماتریس با استفاده از درایه‌های متناظر در آرایه جواب مقارنه می‌شود. بدیهی است که اگر بلوک  $a$  در درون مخروط استخراجی بلوک  $b$  قرار داشته باشد، در صورت استخراج بلوک  $b$  بلوک  $a$  نیز باید استخراج شود. به عبارت دیگر برای پایداری شیب باید رابطه  $l$  برای  $l=(2,3,\dots,L)$  و  $m=(2,3,\dots,M-1)$  برقرار باشد.

$$\text{if } A(l,m)=1 \text{ then } A_{l-1,m-1} + A_{l-1,m} + A_{l-1,m+1} = 3 \quad (5)$$

برای محاسبه ارزش هر جواب از رابطه ۶ استفاده می‌شود.

$$V_i = \sum_{l=1}^L \sum_{m=1}^M A_i(l,m) \times BEV(l,m) \quad (6)$$

که در آن:

$L$  تعداد سطرهای مدل بلوکی

$M$  تعداد ستون‌های مدل بلوکی

$BEV$  ارزش اقتصادی بلوک

۴-۱-۱- مراحل تعیین محدوده نهایی معدن با استفاده از الگوریتم زنبور

۴-۱-۱-۱- مرحله اول

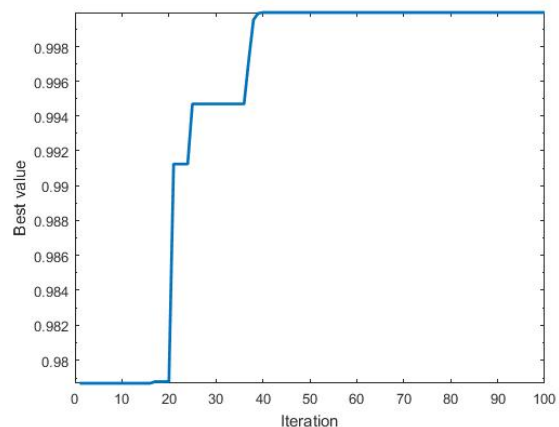
اولین مرحله از الگوریتم زنبور عسل ایجاد جواب‌های اولیه است. برای ایجاد جواب‌های اولیه  $SN$  (تعداد جواب‌های اولیه) آرایه که هر آرایه  $D$  (تعداد بلوک‌ها یا متغیرهایی که باید برای آن‌ها تصمیم‌گیری شود) درایه دارد، ایجاد می‌شود. درایه‌های هر آرایه به تصادف با مقادیر صفر و یک پر می‌شود. برای حل سریع‌تر مساله می‌توان محدوده نهایی حاصل از مخروط شناور را به عنوان یکی از جواب‌های اولیه در نظر گرفت و سایر جواب‌ها را با ایجاد تغییرات تصادفی بر روی جواب حاصل از مخروط شناور ایجاد کرد که با این عمل سرعت رسیدن به جواب بهینه افزایش می‌یابد، سپس ماتریس  $A$  با استفاده از رابطه ۵ برای هر یک از جواب‌های اولیه محاسبه و ارزش هر جواب با استفاده از رابطه ۶ محاسبه می‌شود.

۴-۱-۲- مرحله دوم

به تعداد منابع، زنبورهای کارگر مصنوعی ایجاد می‌شود. هر یک از این زنبورهای مصنوعی با استفاده از رابطه ۷ که برای

سایر تکرارهای الگوریتم تا ارضای شرط پایان مشابه آن‌چه در تکرار اول گفته شد انجام می‌شود.

در این مثال تنها یک تکرار از الگوریتم شرح داده شد. برای حل مثال بیان شده تکرارها به صورت متوالی انجام می‌پذیرد تا شرط پایان الگوریتم ارضا شود. این مثال در محیط نرم‌افزار متلب اجرا شد و جواب نهایی (1.53, 3.15) به دست آمد. در شکل ۲ بهبود جواب با افزایش تکرارها برای این مثال نشان داده شده است.



شکل ۲- بهبود جواب با افزایش تکرارها

همان‌طور که قبلاً بیان شد و در مثال بالا نیز مشاهده می‌شود، در تکرارهای متوالی الگوریتم جواب‌های مساله به مرور به یکدیگر نزدیک و با کم شدن تفاضل مختصات جواب‌ها و کوچک‌تر شدن گام‌های الگوریتم، به مقدار بیشینه همگرا می‌شوند.

۴- تعریف مساله تعیین محدوده نهایی با استفاده از الگوریتم زنبور عسل

تعیین محدوده نهایی معدن به معنی تصمیم‌گیری در مورد استخراج یا عدم استخراج هر یک از بلوک‌های مدل بلوکی اقتصادی معدن است. برای حل این مساله تصمیم می‌توان از متغیرهای دودویی<sup>۱</sup> استفاده کرد، به طوری که متغیر مربوط به هر بلوک مفروضاً اگر بلوک استخراج شود برابر یک و در غیر این صورت برابر صفر شود. برای حل این مساله از الگوریتم بهینه‌سازی زنبور عسل استفاده می‌شود. برای تعیین محدوده نهایی ابتدا آرایه‌ای که تعداد درایه‌های آن برابر تعداد بلوک‌های درون مدل بلوکی است، ایجاد می‌شود. هر درایه از این آرایه نشان‌دهنده یک متغیر دودویی برای یک بلوک است که بسته به استخراج یا عدم استخراج بلوک ممکن است مقادیر یک یا

<sup>۱</sup> Binary



مطلوب رسید، در صورتی که اگر تمام زمان حل مساله صرف جواب‌های نسبتاً خوب شود، این احتمال وجود دارد که مساله در تله بهینه موضعی بیفتد.

پس از محاسبه شایستگی جواب، احتمال دریافت زنبور به وسیله هر جواب، با استفاده از رابطه ۱ محاسبه می‌شود و با توجه به آن شایستگی و این احتمال به هر جواب تعدادی زنبور اختصاص داده می‌شود. اگر شایستگی یک جواب بیشتر باشد احتمال انتخاب شدن آن برای جذب زنبور جدید بیشتر خواهد بود و حتی ممکن است همه زنبورهای ناظر به یک جواب اختصاص یابد. سپس با استفاده از رابطه ۲ جواب جدیدی ایجاد می‌شود. اگر ارزش این جواب نسبت به جواب قبلی بالاتر باشد، جواب جدید جایگزین جواب قبل می‌شود و در غیر این صورت به این جواب جریمه تعلق می‌گیرد.

#### ۴-۱-۴- مرحله چهارم

در این مرحله جواب‌هایی که مقدار جریمه آن‌ها به  $C_{max}$  رسیده است، با جواب‌های تصادفی جدید جایگزین می‌شوند، مقدار شمارنده  $C$  مربوط به جواب جایگزین شده صفر می‌شود. در نهایت، شروط پایان الگوریتم بررسی می‌شود. در صورت ارضای این شروط الگوریتم متوقف می‌شود و در غیر این صورت کنترل الگوریتم به مرحله دوم بازمی‌گردد.

روندنمای تعیین محدوده نهایی معدن با استفاده از الگوریتم زنبور عسل در شکل ۴ نشان داده شده است.

#### ۴-۲- حل مثال دو بعدی

##### ۴-۲-۱- مرحله اول

در مدل بلوکی دو بعدی شکل ۳ برای تعیین محدوده نهایی با استفاده از الگوریتم زنبور عسل، ابتدا تعدادی جواب اولیه تولید می‌شود و سپس برای هر کدام از جواب‌ها آرایه‌ای که تعداد درایه‌های آن برابر تعداد بلوک‌های مثبت درون مدل بلوکی اقتصادی است، ایجاد می‌شود. با توجه به این که در مدل شکل ۳ تعداد بلوک‌های مثبت برابر ۳ است، هر کدام از آرایه‌ها ۳ درایه خواهند داشت. به هر کدام از این درایه‌ها در صورتی که بلوک مثبت متناظر درایه در درون محدوده نهایی جواب قرار داشته باشد عدد ۱ و در غیر این صورت عدد صفر اختصاص می‌یابد. علاوه بر این برای هر جواب یک ماتریس صفر  $A$  نیز تولید می‌شود که ابعاد آن مشابه مدل بلوکی است (برای این مثال  $3 \times 5$ ) و درایه‌های آن برای بلوک‌های درون محدوده نهایی جواب با عدد ۱ و برای سایر بلوک‌ها با عدد صفر پر می‌شود.

حل مسایل با متغیرهای دودویی در این مقاله آرایه شده است، جواب جدیدی را ایجاد می‌کنند.

$$v_{i,j} = x_{ij} + \text{round}(\varphi_{i,j}(x_{ij} - x_{k,j})) \quad (7)$$

که در آن:

$j = [1, 2, \dots, D]$  شماره بلوک در آرایه جواب

$x_{i,j}$ ،  $x_{k,j}$  و  $v_{i,j}$  متغیرهای دودویی

تابع round عدد صحیح بودن  $v_{i,j}$  را تضمین می‌کند و برای صفر یا یک بودن  $v_{i,j}$  لازم است  $\varphi$  یک عدد نامشبت باشد. به تجربه معلوم شده است که برای متغیرهای صفر و یک بهترین بازه تغییرات  $\varphi$  بازه  $[-0.65, 0]$  است.

پس از ایجاد جواب جدید ارزش این جواب با استفاده از رابطه ۶ محاسبه می‌شود. اگر ارزش جواب جدید برابر جواب قبل یا بیشتر از آن باشد، جواب جدید جایگزین جواب قبلی می‌شود، در غیر این صورت این جواب باید جریمه شود. برای جریمه کردن، یک واحد به شمارنده  $C$  ایجاد شده برای آن جواب اضافه می‌شود. اگر شمارنده جریمه یک جواب به یک حد از پیش تعیین شده برسد، این جواب با جواب تصادفی جدیدی در محدوده جواب جایگزین می‌شود.

#### ۴-۱-۳- مرحله سوم

در این مرحله ابتدا میزان شایستگی هر جواب محاسبه می‌شود. در این تحقیق رابطه ۸ برای تعیین شایستگی هر جواب پیشنهاد شده است.

$$fit_i = \begin{cases} value_i + \varepsilon & \text{if } value_i \geq 0 \\ |value_i| & \text{if } -1 < value_i < 0 \\ \frac{1}{|value_i|} & \text{if } value_i \leq -1 \end{cases} \quad (8)$$

که در آن:

شایستگی منبع  $i$

val ارزش جواب  $i$  (ارزش پیت تشکیل شده به ازای جواب  $i$ )

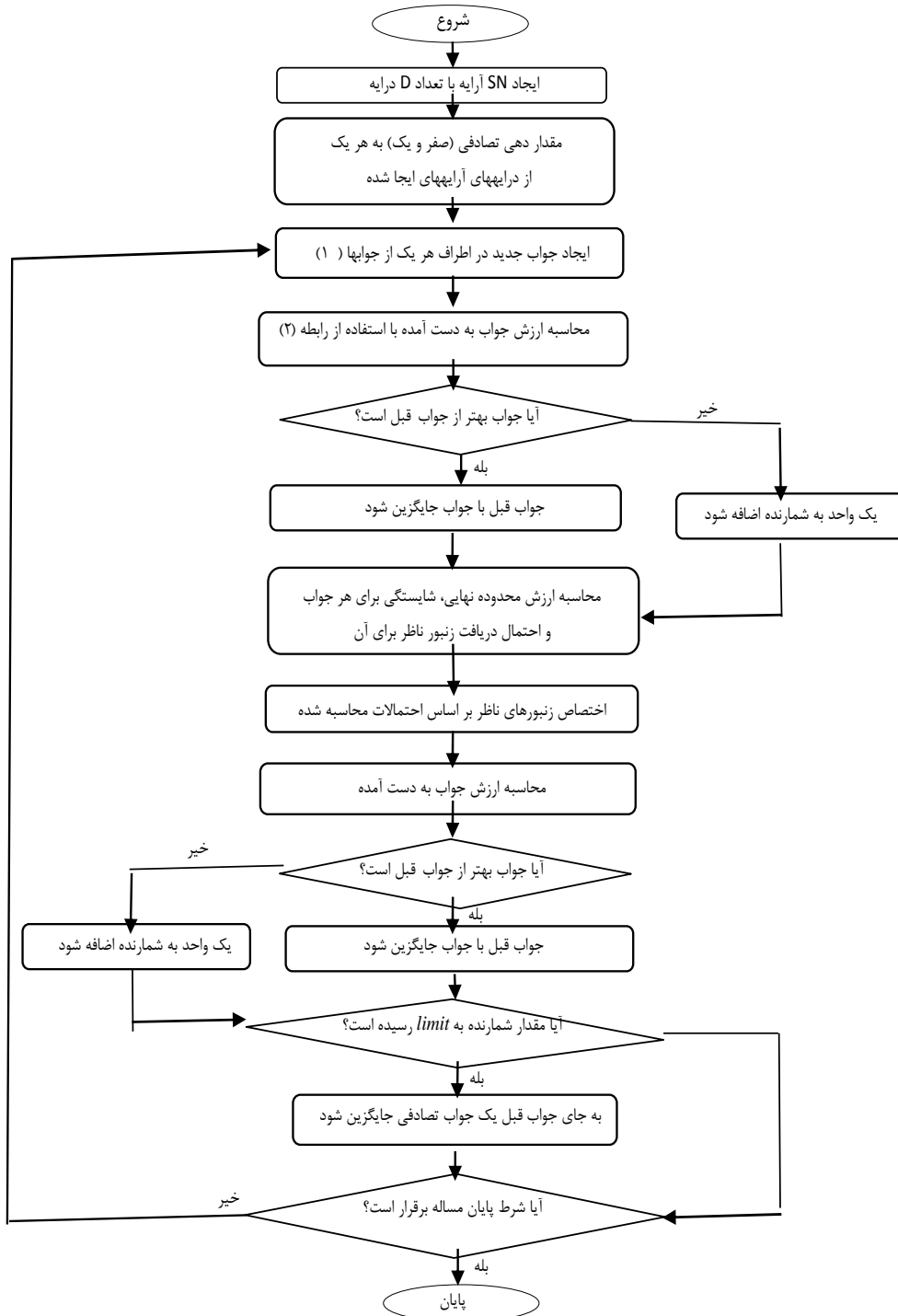
یک عدد بسیار کوچک

در صورت بزرگ بودن قدر مطلق ارقام مربوط به ارزش جواب، آن را بر یک عدد مثبت تقسیم می‌کنند. بهتر است این عدد طوری انتخاب شود که با تقسیم ارزش پیت بر آن اعداد تک رقمی بزرگتر از دو به دست آید تا در صورت منفی بودن ارزش جواب، احتمال انتخاب آن به سمت صفر میل نکند چرا که در الگوریتم زنبور عسل به جواب‌های بد نیز امکان بهبود داده می‌شود. چه بسا بتوان با بهبود یک جواب بد به نتیجه

|    |    |    |    |    |
|----|----|----|----|----|
| -۱ | -۱ | -۵ | -۱ | -۱ |
| -۲ | ۶  | -۴ | ۶  | -۲ |
| -۳ | -۳ | ۳  | -۳ | -۳ |

شکل ۳- مدل بلوکی دو بعدی اقتصادی معدن (BEV).

در این مثال اگر دو جواب اولیه به دست آید، در جواب اول ممکن است محدوده نهایی به دست آمده با استفاده از مخروط شناور باشد (شکل ۵). محدوده نهایی مربوط به این جواب ( $A_1$ ) در شکل ۶ نشان داده شده است. آرایه مربوط به این جواب نیز در ستون چهارم جدول ۱ دیده می شود.



شکل ۴- روند نمای تعیین محدوده نهایی معدن با الگوریتم زنبور.

ارزش محدوده نهایی جواب دو نیز با استفاده از رابطه ۶ محاسبه می‌شود:

$$V_2 = \sum_{i=1}^3 \sum_{m=1}^5 A_2(l, m) \times BEV(l, m) = -1$$

۴-۲-۲- مرحله دوم

در این مرحله با استفاده از رابطه ۷ برای هر یک از جواب‌های مساله، جواب جدیدی ایجاد می‌شود. برای ایجاد جواب‌های جدید می‌توان تعداد بلوک‌هایی که تغییرات بر روی آن‌ها اعمال می‌شود، تعیین کرد. همچنین بلوک‌هایی تغییرات باید بر روی آن‌ها اعمال شود، به صورت تصادفی تعیین می‌شود. در این مثال، در جواب اول یک بلوک که به تصادف بلوک متناظر درایه یک است، برای تغییر انتخاب شده است:

$$V_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \text{round} \left[ \begin{pmatrix} 0.54 \\ 0 \\ 0 \end{pmatrix} \times \left( \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right) \right] = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

جواب به دست آمده بدین معنی است که در جواب جدید باید بلوک‌های متناظر درایه‌های دو (بلوک واقع در سطر ۲ و ستون ۴) و سه (بلوک واقع در سطر ۳ و ستون ۳) در درون محدوده نهایی قرار گیرند. محدوده در برگرفته این دو بلوک مثبت همان محدوده نهایی پیشین یعنی شکل ۶ خواهد بود و بهبودی حاصل نخواهد شد. به دلیل عدم بهبود جواب در این مرحله، جریمه‌ای برای جواب یک در نظر گرفته شده و یک واحد به شمارنده این جواب افزوده خواهد شد و  $C_1$  به یک افزایش خواهد یافت.

$$x_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \rightarrow v_1 = 2 \rightarrow C_1 = 1$$

برای جواب دوم نیز یک بلوک که به تصادف بلوک متناظر درایه‌سپه‌است، برای تغییر انتخاب شده است:

$$V_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \text{round} \left[ \begin{pmatrix} 0 \\ 0 \\ -0.25 \end{pmatrix} \times \left( \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right) \right] = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

در این جواب نیز بهبودی حاصل نشده و جواب دوم نیز باید جریمه شده و یک واحد به شمارنده به آن اضافه شود. بنابراین:

$$x_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \rightarrow v_2 = 1 \rightarrow C_2 = 1$$

|    |    |    |    |    |
|----|----|----|----|----|
| -۱ | -۱ | -۵ | -۱ | -۱ |
| -۲ | ۶  | -۴ | ۶  | -۲ |
| -۳ | -۳ | ۳  | -۳ | -۳ |

شکل ۵- محدوده نهایی مربوط به جواب ۱

جدول ۱- وضعیت بلوک‌های مثبت مدل در جواب ۱

| شماره بلوک | شماره ردیف در مدل بلوکی | شماره ستون در مدل بلوکی | درون یا بیرن از محدوده نهایی |
|------------|-------------------------|-------------------------|------------------------------|
| ۱          | ۲                       | ۲                       | ۱                            |
| ۲          | ۲                       | ۴                       | ۱                            |
| ۳          | ۳                       | ۳                       | ۱                            |

|   |   |   |   |   |
|---|---|---|---|---|
| ۱ | ۱ | ۱ | ۱ | ۱ |
| ۰ | ۱ | ۱ | ۱ | ۰ |
| ۰ | ۰ | ۱ | ۰ | ۰ |

شکل ۶- محدوده نهایی ایجاد شده برای جواب ۱

درایه جوابیک:

$$x_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

ارزش محدوده نهایی جواب یک نیز با استفاده از رابطه ۶ محاسبه می‌شود:

$$V_1 = \sum_{i=1}^3 \sum_{m=1}^5 A_1(l, m) \times BEV(l, m) = 2$$

اکنون با اعمال تغییراتی در جواب یک جواب دوم انتخاب می‌شود. اگر جواب دوم طوری انتخاب شود که محدوده نهایی آن ( $A_2$ ) به صورت شکل ۷ باشد. آرایه مربوط به این جواب نیز در ستون چهارم جدول ۲ دیده می‌شود.

جدول ۲- وضعیت بلوک‌های مثبت مدل در جواب ۲

| شماره بلوک | شماره ردیف در مدل بلوکی | شماره ستون در مدل بلوکی | درون یا بیرن از محدوده نهایی |
|------------|-------------------------|-------------------------|------------------------------|
| ۱          | ۲                       | ۲                       | ۰                            |
| ۲          | ۲                       | ۴                       | ۱                            |
| ۳          | ۳                       | ۳                       | ۰                            |

|   |   |   |   |   |
|---|---|---|---|---|
| ۰ | ۰ | ۱ | ۱ | ۱ |
| ۰ | ۰ | ۰ | ۱ | ۰ |
| ۰ | ۰ | ۰ | ۰ | ۰ |

شکل ۷- محدوده نهایی ایجاد شده برای جواب ۲

درایه جواب دو:

$$x_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

|   |   |   |   |   |
|---|---|---|---|---|
| ۱ | ۱ | ۱ | ۱ | ۱ |
| ۰ | ۱ | ۰ | ۱ | ۰ |
| ۰ | ۰ | ۰ | ۰ | ۰ |

شکل ۸- محدوده نهایی ایجاد شده برای جواب ۲

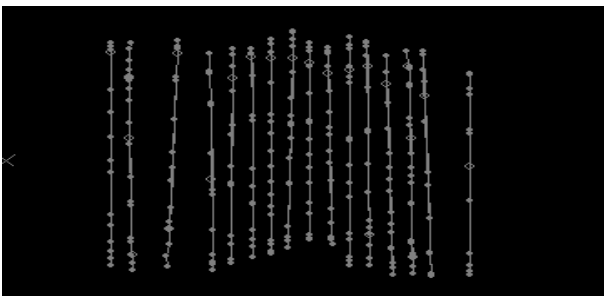
## ۴-۲-۴- مرحله چهارم

در این مرحله مقدار شمارنده جواب‌ها بررسی می‌شود. چون مقدار این شمارنده‌ها به مقدار از پیش تعیین شده که در این مثال ۵ در نظر گرفته شده نرسیده است، هیچ یک از جواب‌ها با جواب تصادفی جدید جایگزین نمی‌شود. در نهایت شروط پایان الگوریتم بررسی می‌شود. در این مثال برای پایان اجرای الگوریتم دو شرط در نظر گرفته شده است. شرط اول بار ۳۰ تکرار و شرط دوم عدم بهبود جواب در ۵ تکرار متوالی است. اگر یکی از این شروط برقرار باشد الگوریتم متوقف می‌شود و در غیر این صورت کنترل الگوریتم به مرحله دوم بازمی‌گردد.

در این جا فقط یک تکرار از الگوریتم شرح داده شده است و این عملیات تا رسیدن به یکی از شرایط توقف الگوریتم ادامه می‌یابد.

## ۴-۳- تعیین محدوده نهایی سه بعدی

تعیین محدوده نهایی معدن به کمک کامپیوتر معمولاً با استفاده از مدل بلوکی انجام می‌شود. در این پژوهش برای امکان‌سنجی کاربرد الگوریتم در مسایل سه‌بعدی و بزرگ مقیاس، مدل بلوکی معدن سونگون در نرم‌افزار Datamine ساخته شد. این معدن دارای ۱۶۰ گمانه اکتشافی است که با استفاده از این گمانه‌ها ۱۸ مقطع برای معدن رسم شد (شکل ۹).



شکل ۹- مقاطع رسم شده برای معدن سونگون

برای ساخت مدل بلوکی عیاری ابعاد بلوک‌ها ۲۵×۲۵×۲۵ متر در نظر گرفته شد و مدل بلوکی با ۱۲۰×۱۰۰×۴۵ بلوک ساخته شد. پس از ساخت مدل بلوکی و تعیین چگالی آن با استفاده از کریجینگ عیار بلوک‌ها تخمین زده شد (شکل ۱۰).

## ۴-۲-۳- مرحله سوم

در این مرحله پس از محاسبه شایستگی جواب‌های جدید با استفاده از رابطه ۸ و به دست آوردن احتمال اختصاص آن‌ها به هر یک از جواب‌های قبل با استفاده از رابطه ۱، جواب‌های جدید اختصاص داده می‌شود:

$$fit_1 = 2$$

$$fit_2 = \frac{1}{|-1|} = 1$$

$$p_1 = \frac{fit_1}{fit_1 + fit_2} = \frac{2}{2+1} = 0.67$$

$$p_2 = \frac{cost_2}{fit_1 + fit_2} = \frac{1}{2+1} = 0.33$$

پس از محاسبه احتمال تجمعی که برای جواب یک ۰/۶۷ و برای جواب دو ۱ به دست می‌آید، عددی تصادفی بین صفر و یک تولید می‌شود. این عدد تصادفی اگر بین صفر و ۰/۶۷ باشد جواب اول و اگر بین ۰/۶۷ و یک باشد جواب دو برای اختصاص جواب جدید انتخاب می‌شود. اگر به صورت تصادفی عددهای ۰/۴۱ و ۰/۸۹ به دست آمده باشد، به هر یک از جواب‌ها یک جواب جدید اختصاص داده می‌شود و مانند مرحله دوم اقدام می‌شود.

$$v_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + round \left[ \begin{pmatrix} 0 \\ 0 \\ -0.21 \end{pmatrix} \times \left( \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right) \right] = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + round \left[ \begin{pmatrix} 0.51 \\ 0 \\ 0 \end{pmatrix} \times \left( \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right) \right] = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

به علت عدم بهبود جواب اول جواب جدید جایگزین آن نشده و با جریمه شدن این جواب یک واحد به شمارنده آن افزوده می‌شود اما ارزش جواب دوم ۳ است که نسبت به ارزش قبلی یعنی ۱- بالاتر است، بنابراین این جواب جایگزین جواب پیشین می‌شود. در نتیجه:

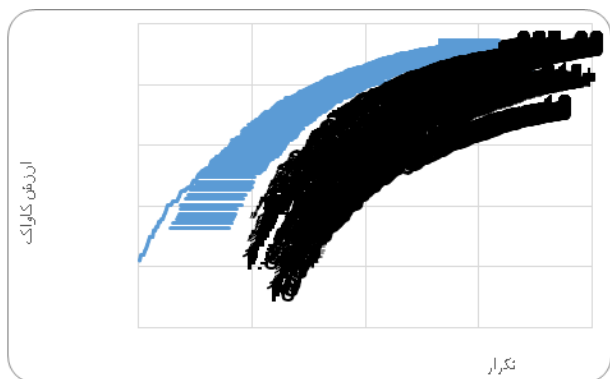
$$x_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \rightarrow V_1 = 2 \rightarrow C_1 = 2$$

$$x_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \rightarrow V_2 = 3 \rightarrow C_2 = 1$$

ماتریس صفر جواب جدید جایگزین شده جواب دو قبل در

شکل ۸ دیده می‌شود.

نرم افزار متلب الگوریتم زنبور عسل برای تعیین محدوده نهایی معدن صورت بندی و بر روی مدل بلوکی معدن فرضی اجرا شد. با اجرای الگوریتم زنبور عسل میزان نقدینگی ناشی از استخراج محدوده نهایی ۲۱,۱۸۴ میلیارد دلار به دست آمد که نسبت به نقدینگی حاصل از محدوده به دست آمده با استفاده از الگوریتم مخروط شناور (۱۸,۵۷۲ میلیارد دلار) ۱۲,۳ درصد بیشتر است.



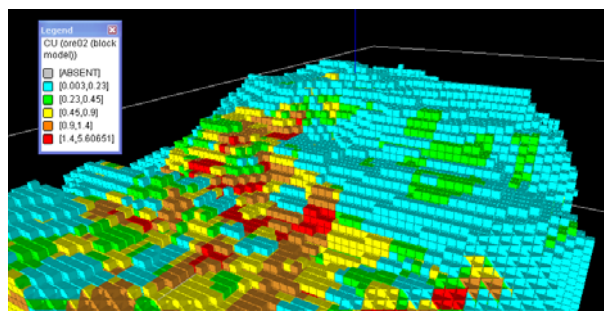
شکل ۱۱- بهبود ارزش محدوده نهایی با افزایش تکرارها

#### ۵- اعتبارسنجی

یکی از روش های تعیین محدوده نهایی معدن استفاده از نظریه گراف لرج و گروسمن است. از نظر ریاضی اثبات شده است که این الگوریتم منجر به تعیین محدوده نهایی واقعی معدن می شود. در این مقاله برای اعتبارسنجی تعیین محدوده نهایی با استفاده از الگوریتم زنبور عسل، از نظریه گراف و نرم افزار NPV Scheduler که محدوده نهایی را براساس این نظریه تعیین می کند، استفاده شد. نقدینگی حاصل از محدوده نهایی تعیین شده با استفاده از نظریه گراف ۲۱,۵۲۸ میلیارد دلار به دست آمد که در مقایسه با نقدینگی حاصل از محدوده نهایی به دست آمده با الگوریتم زنبور عسل فقط ۱,۶ درصد بیشتر است. در شکل های ۱۲ و ۱۳ به ترتیب مقطعی از محدوده نهایی معدن با استفاده از الگوریتم زنبور عسل و نظریه گراف نمایش داده شده است.



شکل ۱۲- محدوده نهایی به دست آمده با استفاده از الگوریتم زنبور عسل



شکل ۱۰- مدل بلوکی عیاری معدن سونگون

پس از ساخته شدن مدل بلوکی عیاری یک خروجی اکسل از آن تهیه شد. در مرحله بعد با فراخوانی مدل بلوکی عیاری معدن به نرم افزار Matlab با استفاده از پارامترهای اقتصادی جدول ۳، مدل بلوکی اقتصادی با استفاده از رابطه های ۹ و ۱۰ ساخته شد [۴۳].

جدول ۳- پارامترهای اقتصادی

| پارامتر            | هزینه و قیمت       |
|--------------------|--------------------|
| قیمت مس            | ۷۱۰۰ دلار بر تن مس |
| هزینه معدنکاری     | ۲/۵ دلار بر تن سنگ |
| هزینه فرآوری       | ۱۰ تن کانسنگ       |
| هزینه ذوب و پالایش | ۱۰۰ تن فلز         |

(۹)

$$v = [(p-r) * g * y - m - c] Q$$

(۱۰)

$$v = \begin{cases} v & \text{if } (p-r) * g * y > c \\ -mQ & \text{if } (p-r) * g * y < c \end{cases}$$

در این روابط

$v$  ارزش بلوک

$p$  قیمت

$r$  هزینه ذوب تصفیه و فروش یک تن فلز

$g$  عیار بلوک

$y$  بازیابی فلز

$m$  هزینه استخراج یک تن سنگ

$c$  هزینه فرآوری یک تن کانسنگ

$Q$  تناژ بلوک

برای حل این مدل سه بعدی نیز مشابه آنچه در حالت دوبعدی توضیح داده شد، عمل می شود. برای این منظور در

*Erzmetall*, vol. 37, no. 1984, pp. 139-44, 1984.

3. J. Yamatomi, G. Mogi, A. Akaike, and U. Yamaguchi, "Selective extraction dynamic cone algorithm for three-dimensional open pit designs," in *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, 1996, vol. 5, no. 33, p. 220A.
4. H. Lerchs and G. FI, "Optimum design of open-pit mines," in *Operations Research*, 1964, vol. 12, pp. B59-&: INST OPERATIONS RESEARCH MANAGEMENT SCIENCES 901 ELKRIDGE LANDING RD, STE 400, LINTHICUM HTS, MD 21090-2909.
5. R. Underwood and B. Tolwinski, "A mathematical programming viewpoint for solving the ultimate pit problem," *European Journal of Operational Research*, vol. 107, no. 1, pp. 96-107, 1998.
6. P. Huttagosol and R. Cameron, "A computer design of ultimate pit limit by using transportation algorithm," in *Proceedings of 23rd APCOM symp., Tucson, Arizona*, 1992, pp. 443-460.
7. T. Johnson, "A comparative study of methods for determining ultimate open-pit mining limits," *11th Conference on Application of Computer Methods in the Mineral Industries, University of Arizona*, 1973.
8. R. Khalokakaie, P. Dowd, and R. Fowell, "A Windows program for optimal open pit design with variable slope angles," *International Journal of Surface Mining, Reclamation and Environment*, vol. 14, no. 4, pp. 261-275, 2000.
9. Y. C. Kim, "Ultimate pit limit design methodologies using computer models—The state of the art," *Mining engineering*, vol. 30, no. 10, pp. 1454-1459, 1978.
10. T. R. Carlson, J. D. Erickson, D. O'Brain, and M. T. Pana, "Computer techniques in mine planning," *Mining Engineering*, vol. 18, no. 5, pp. 53-56, 1966.
11. E. Wright, "MOVING CONE II-A simple algorithm for optimum pit limits design," in *Proceedings of the 28th Symposium on the application of computers and operations research in the mineral industries (APCOM), (Colorado USA)*, 1999, pp. 367-374.



شکل ۱۳-محدوده نهایی به دست آمده با استفاده از نظریه گراف

#### ۶- نتیجه گیری

پس از اکتشاف یک کانسار در صورتی که روش استخراج روباز برای بهره‌برداری از آن انتخاب شده باشد، لازم است تا محدوده نهایی معدن تعیین شود. از اهداف طراحی محدوده نهایی یک معدن، تعیین پارامترهایی مانند میزان گسترش طولی، عرضی و عمقی معدن، مسیرهای دسترسی به ماده معدنی، محل دپوی باطله، محل تاسیسات سطحی، نسبت باطله‌برداری، عمر معدن، میزان ذخیره قابل استخراج به روش روباز، میزان باطله‌برداری و نحوه برنامه‌ریزی تولید است. تا به حال روش‌های گوناگونی برای تعیین محدوده نهایی معدن پیشنهاد شده است. در این پژوهش برای تعیین این محدوده یک مدل برنامه‌ریزی صفر و یک پیشنهاد شده و برای حل مدل از الگوریتم زنبور عسل کمک گرفته شده است. برای ارزیابی کارایی این الگوریتم در تعیین محدوده نهایی معدن، مراحل الگوریتم در یک مثال دوبعدی به صورت کامل شرح داده شد. سپس این الگوریتم برای تعیین محدوده نهایی یک معدن سونگونیا  $45 \times 100 \times 120$  بلوک به کار گرفته شد. برای اعتبارسنجی عملکرد این الگوریتم، محدوده نهایی معدن فرضی با استفاده از روش مخروط شناور و نظریه‌گراف تعیین شد. نتایج اعتبارسنجی نشان می‌دهد که اختلاف نقدینگی محدوده به دست آمده از طریق الگوریتم زنبور عسل حدود  $3/12$  درصد از محدوده مخروط شناور بیشتر و تنها  $6/1$  درصد از محدوده نظریه‌گراف کمتر است.

#### منابع و مراجع

1. E. Koenigsberg, "The optimum contours of an open pit mine: An application of dynamic programming," *17th Application of Computers and Operations Research in the Mineral Industry*, pp. 274-287, 1982.
2. F. Wilke and E. Wright, "Determining the optimal ultimate pit design for hard rock open pit mines using dynamic programming,"

22. O. B. Haddad, A. Afshar, and M. A. Mariño, "Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization," *water resources management*, vol. 20, no. 5, pp. 661-680, 2006.
23. G. Li, P. Niu, and X. Xiao, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization," *Applied soft computing*, vol. 12, no. 1, pp. 320-332, 2012.
24. G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied mathematics and computation*, vol. 217, no. 7, pp. 3166-3173, 2010.
25. X. Xu and X. Lei, "Multiple sequence alignment based on abc\_sa," in *International Conference on Artificial Intelligence and Computational Intelligence*, 2010, pp. 98-105: Springer.
26. H. Narasimhan, "Parallel artificial bee colony (PABC) algorithm," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 306-311: IEEE.
27. W. Zou, Y. Zhu, H. Chen, and X. Sui, "A clustering approach using cooperative artificial bee colony algorithm," *Discrete dynamics in nature and society*, vol. 2010, 2010.
28. F. Kang, J.-J. Li, and Q. Xu, "Hybrid simplex artificial bee colony algorithm and its application in material dynamic parameter back analysis of concrete dams," *Journal of Hydraulic Engineering*, vol. 40, no. 6, pp. 736-742, 2009.
29. H. Zhao, Z. Pei, J. Jiang, R. Guan, C. Wang, and X. Shi, "A hybrid swarm intelligent method based on genetic algorithm and artificial bee colony," *Advances in Swarm Intelligence*, pp. 558-565, 2010.
30. X. Shi, Y. Li, H. Li, R. Guan, L. Wang, and Y. Liang, "An integrated algorithm based on artificial bee colony and particle swarm optimization," in *Natural Computation (ICNC), 2010 Sixth International Conference on*, 2010, vol. 5, pp. 2586-2590: IEEE.
31. W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871-882, 2011.
12. M. David, P. Dowd, and S. Korobov, "Forecasting departure from planning in open pit design and grade control," in *12th Symposium on the application of computers and operations research in the mineral industries (APCOM)*, 1974, vol. 2, pp. F131-F142.
13. P. Dowd, "Open-pit optimization. Pt. 1: optimal open-pit design," *Transactions of the Institution of Mining and Metallurgy. Section A. Mining Industry*, vol. 102, 1993.
14. B. Denby and D. Schofield, "Open-pit design and scheduling by use of genetic algorithms," *Transactions of the Institution of Mining and Metallurgy. Section A. Mining Industry*, vol. 103, 1994.
۱۵. ع. عظیمی, "تعیین محدوده نهایی معدن با استفاده از الگوریتم کلونی مورچگان," پایان نامه کارشناسی ارشد دانشگاه صنعتی شاهرود, ۱۳۸۹.
۱۶. ش. ا. رافعی, "بهینه سازی محدوده نهایی و برنامه ریزی تولید در معادن روباز با استفاده از الگوریتم رقابت استعماری," پایان نامه کارشناسی ارشد دانشگاه صنعتی شاهرود, ۱۳۹۴.
17. D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department 2005.
18. D. Teodorovic and M. Dell'Orco, "Bee colony optimization—a cooperative learning approach to complex transportation problems," *Advanced OR and AI methods in transportation*, pp. 51-60, 2005.
19. H. Drias, S. Sadeg, and S. Yahi, "Cooperative bees swarm for solving the maximum weighted satisfiability problem," *Computational intelligence and bioinspired systems*, pp. 417-448, 2005.
20. X.-S. Yang, "Engineering optimizations via nature-inspired virtual bee algorithms," *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, pp. 317-323, 2005.
21. D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm—A novel tool for complex optimisation," in *Intelligent Production Machines and Systems-2nd I\* PROMS Virtual International Conference (3-14 July 2006)*, 2011: sn.

38. F. J. Rodriguez, C. García-Martínez, C. Blum, and M. Lozano, "An artificial bee colony algorithm for the unrelated parallel machines scheduling problem," in *International Conference on Parallel Problem Solving from Nature*, 2012, pp. 143-152: Springer.
39. I. M. S. de Oliveira, R. Schirru, and J. de Medeiros, "On the performance of an artificial bee colony optimization algorithm applied to the accident diagnosis in a pwr nuclear power plant," in *2009 international nuclear Atlantic conference (INAC 2009)*, 2009.
40. R. Irani and R. Nasimi, "Application of artificial bee colony-based neural network in bottom hole pressure prediction in underbalanced drilling," *Journal of Petroleum Science and Engineering*, vol. 78, no. 1, pp. 6-12, 2011.
41. E. Ebrahimi, M. Monjezi, M. R. Khalesi, and D. J. Armaghani, "Prediction and optimization of back-break and rock fragmentation using an artificial neural network and a bee colony algorithm," *Bulletin of Engineering Geology and the Environment*, vol. 75, no. 1, pp. 27-36, 2016.
42. D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459-471, 2007.
43. F. D. Castillo and R. Dimitrakopoulos, "Joint effect of commodity price and geological uncertainty over the life of mine and ultimate pit limit," *Mining Technology*, vol. 123, no. 4, pp. 207-219, 2014.
32. A. Banharnsakun, T. Achalakul, and B. Sirinaovakul, "Artificial bee colony algorithm on distributed environments," in *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*, 2010, pp. 13-18: IEEE.
33. A. Basturk and R. Akay, "Parallel implementation of synchronous type artificial bee colony algorithm for global optimization," *Journal of Optimization Theory and Applications*, vol. 155, no. 3, pp. 1095-1104, 2012.
34. D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21-57, 2014.
35. B. Nozohour-leilabady and B. Fazelabdolabadi, "On the application of artificial bee colony (ABC) algorithm for optimization of well placements in fractured reservoirs; efficiency comparison with the particle swarm optimization (PSO) methodology," *Petroleum*, vol. 2, no. 1, pp. 79-89, 2016.
36. A. Ahmad, S. F. M. Razali, Z. S. Mohamed, and A. El-shafie, "The Application of Artificial Bee Colony and Gravitational Search Algorithm in Reservoir Optimization," *Water Resources Management*, vol. 30, no. 7, pp. 2497-2516, 2016.
37. C. Zhang, D. Ouyang, and J. Ning, "An artificial bee colony approach for clustering," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4761-4767, 2010.