بسم الله الرحمن الرحيم

دانشگاه صنعتی شاهرود

دانشکده مهندسی معدن، نفت و ژئوفیزیک

گروه استخراج معدن

پایان نامه جهت اخذ مدرک کارشناسی ارشد

تعیین محدوده نهایی معادن روباز با کمک روش بهینه سازی الگوریتم کولونی مورچگان

علی عظیمی

استاد راهنما

آقای دکتر رضا خالو کاکایی

استاد مشاور

آقای دکتر جعفر فتحعلی

تابستان ۸۹

این اثر کوچک را تقدیم می‌کنم به

پدرو مادر مهربانم

تشکر و قدردانی

در ابتدا لازم می‌دانم از استاد گرانقدرم آقای دکتر کاکایی که با راهنمایی‌های ارزنده‌شان مرا در انجام این پایان نامه راهنمایی کرده‌اند تشکر کنم. همچنین از اساتید محترم دانشکده مهندسی معدن، نفت و ژئوفیزیک سپاسگذاری می‌کنم. در پایان نیز از آقای دکتر فتحعلی به دلیل مشاورات سودمندشان متشکرم.

علی عظیمی

اینجانب علی عظیمی تأیید می‌نمایم که مطالب مندرج در این پایان‌نامه نتیجه تحقیقات خودم می‌باشد و در صورت استفاده از نتایج دیگران مرجع آن را ذکر نموده‌ام.

کلیه حقوق مادی مترتب از نتایج مطالعات، آزمایشات و نوآوری ناشی از تحقیق موضوع این پایان‌نامه متعلق به دانشگاه صنعتی شاهرود می‌باشد.

ماه و سال

چکیده

قبل از استخراج معادن به روش روباز، لازم است که محدوده نهایی معدن طراحی شود. یکی از مهم‌ترین روش‌های طراحی محدوده، تعیین محدوده بهینه معدن می‌باشد. برای تخمین یا محاسبه دقیق این محدوده، نیاز به مدل بلوکی اقتصادی است که پس از اکتشاف کانسار و ایجاد یک مدل بلوکی زمین شناسی، و با درنظر گرفتن درآمدها و هزینه‌های مدل بلوکی مذکور محاسبه می‌گردد. تاکنون روش‌های بسیاری برای تعیین محدوده نهایی معادن روباز معرفی شده است. این الگوریتم‌ها را می‌توان به دو دسته ریاضی و ابتکاری تقسیم بندی نمود. روش‌های ابتکاری دارای این مزیت هستند که می‌توانند در هر نوع مدل بلوکی، طی زمانی معقول یک جواب بدست آورند، از طرف دیگر معایب این دسته از الگوریتم‌ها را می‌توان عدم بهینه بودن پاسخ یافت شده توسط الگوریتم و نبود دلیل محکم برای ارزیابی بهینه بودن جواب‌ها دانست، ازاینرو این پاسخ‌ها را باید با پاسخ‌های بدست آمده از روش‌های ریاضی مقایسه کرد. روش‌های دیگری که مورد استفاده قرار می‌گیرند، روش‌های ریاضی هستند که دارای اثبات می‌باشند و می‌توانند پاسخ‌های بهینه را پیدا کنند ولی این دسته از الگوریتم‌ها بسیار پیچیده بوده و زمان بسیار زیادی برای رسیدن به پاسخ نیاز دارند. ازاینرو در این پایان نامه مسئله تعیین محدوده نهایی معادن روباز با کمک الگوریتم کولونی مورچگان مورد بررسی قرار گرفته است. بدین منظور ابتدا یک مدل دو بعدی از مدل بلوکی برای حل توسط الگوریتم، مدلسازی شده و محدودیت‌های مسئله بر روی این مدل پیاده می‌شوند. در مراحل بعد، روش‌ها و مواردی در نظر گرفته می‌شوند که می‌توانند به حل مسئله کمک کنند. برای حل مسئله، مدل با استفاده از برنامه نویسی به زبان ویژوال بیسیک، به برنامه قابل حل توسط رایانه تبدیل شد. با پیاده‌سازی الگوریتم کولونی مورچگان بر روی مسئله مشخص شد که این الگوریتم ریاضی ابتکاری توان مناسبی در حل مسئله تعیین محدوده نهایی معادن روباز در زمانی معقول دارد، پارامترهای این الگوریتم به ابعاد مسئله حساس می‌باشند و با بزرگ شدن ابعاد مسئله، ممکن است جواب به سرعت واگرا شود. ازاینرو باید

معیارهای توقف این الگوریتم قادر باشند تا در هر مرحله از ساخت جواب‌ها، بهترین جواب یافت شده را ذخیره کنند.

کلمات کلیدی

محدوده نهایی معدن روباز، الگوریتم کولونی مورچگان، مدل بلوکی دو بعدی، مدلسازی، تابع هدف

مقالاتی که تاکنون از این پایان نامه استخراج شده‌اند:

۱. معرفی یک مکانیزم جدید در بهینه‌سازی به روش کولونی مورچگان برای استفاده در طراحی محدوده نهایی معادن روباز (علی عظیمی، رضا کاکایی)
چهلمین کنفرانس ریاضی کشور، ۲۶ الی ۲۹ مردادماه ۱۳۸۸، دانشگاه صنعتی شریف، شماره مقاله: ۵۳۴

۲. مدل‌سازی بلوک‌های اقتصادی یک معدن روباز به منظور تعیین محدوده نهایی معادن، با کمک روش بهینه‌سازی کولونی مورچگان (علی عظیمی، رضا کاکایی)
چهلمین کنفرانس ریاضی کشور، ۲۶ الی ۲۹ مردادماه ۱۳۸۸، دانشگاه صنعتی شریف، شماره مقاله: ۵۳۵

فهرست

فصل ۱ .... ۲
۱-۱ مقدمه ۲
۱-۲ هدف پایان نامه ۴
۱-۳ سازماندهی پایان نامه ۴
فصل ۲ .... ۷
۲-۱ مقدمه ۷
۲-۱-۱ طراحی محدوده نهایی معادن روباز ۸
۲-۲ روش مخروط شناور ۱۰
۲-۳ روش برنامه ریزی پویا ۱۴
۲-۴ روش کوروبوف ۱۷
۲-۵ روش اصلاح شده کوروبوف ۲۰
۲-۶ روش مخروط شناور II ۲۱
۲-۷ الگوریتم مخروط شناور II اصلاح شده ۲۵
۲-۸ اصلاحیه دوم روش مخروط شناور II ۲۸
۲-۹ الگوریتم لرچ و گرسمان ۳۱
۲-۱۰ روش پارامتری کردن ۳۳
۲-۱۱ الگوریتم ژنتیک ۳۴
۲-۱۲ روش شبکه و حداکثر جریان ۳۶
فصل ۳ .... ۳۹
۳-۱ مقدمه ۳۹
۳-۱-۱ الگوریتم ابتکاری چیست؟ ۴۰
۳-۱-۲ اصول الگوریتمهای ابتکاری ۴۰
۳-۲ دلایل نیاز به الگوریتمهای ابتکاری ۴۱
۳-۲-۱ مسائل بهینه سازی ۴۱
۳-۲-۲ پیچیدگی محاسباتی ۴۱
۳-۲-۳ تابع پیچیدگی زمانی ۴۲
۳-۳ مورچگان و محققان ۴۲
۳-۵ استفاده از رفتار مورچه‌های واقعی برای کمّی‌سازی این رفتار و حل مسئله ۴۴
۳-۶ الگوریتم کولونی مورچگان ۴۸
۳-۷ تولید راه‌حل توسط مورچه‌ها ۵۰
۳-۸ به‌هنگام‌سازی فرومون‌ها ۵۲

۳-۹ بکار بردن جستجوی محلی ۵۳
۳-۱۲ انواع الگوریتم‌های مورچگان ۵۴
۳-۱۲-۱ سیستم مورچگان و مورچه‌ی ممتاز (Ant System & elitism) ۵۴
۳-۱۲-۲ Ant-Q ۵۵
۳-۱۲-۳ سیستم کولونی مورچگان ((Ant Colony System (ACS) ۵۵
۳-۱۲-۴ سیستم کولونی مورچگان و سه برگزیده (ACS & 3-opt) ۵۷
۳-۱۲-۵ Max-Min Ant System (MMAS) ۵۷
۳-۱۳ روش جستجوی تابو ۵۸
۳-۱۴ چگونگی کارکرد الگوریتم کولونی مورچگان ۵۹
۳-۱۵ مسئله فروشنده دوره‌گرد ۶۰
فصل ۴ .... ۶۳
۴-۱ مقدمه ۶۳
۴-۲ طراحی محدوده نهایی معادن روباز با الگوریتم کولونی مورچگان ۶۳
۴-۲-۱ آماده سازی الگوریتم و مدل ۶۴
۴-۲-۲ تعریف محدودیت‌ها ۶۵
۴-۲-۲-۱ محدودیت‌های کلی ۶۵
۴-۲-۲-۲ محدودیت فنی استخراج ۶۶
۴-۲-۲-۳ محدودیت حرکت مورچه‌ها ۶۶
۴-۲-۳ قوانین حرکت مورچه‌ها ۶۹
۴-۲-۴ قوانین انتخاب مسیر و رد فرومون ۷۴
۴-۳ مثالی از روش کار ۷۵
۴-۴ یک نقص در الگوریتم کولونی مورچگان و راه برطرف کردن آن ۸۰
۴-۸ راه حلی برای طراحی معدن با شیب‌های غیر از ۴۵ درجه ۸۱
فصل ۵ .... ۸۳
۵-۱ مقدمه ۸۳
۵-۲ چگونگی کار برنامه توسعه یافته ۸۳
۵-۳ برنامه رایانه‌ای Ant Pit ۸۵
۵-۳-۱ درباره نرم افزار ۸۵
۵-۳-۲ شرح برنامه توسعه یافته ۸۵
۵-۴ روش برنامه نویسی و الزامات ۸۸
۵-۵ مقایسه نتایج الگوریتم کولونی مورچگان با روش‌های دیگر ۹۹
۵-۵-۱ مثال اول ۱۰۰

۵-۵-۲ مثال دوم ۱۰۱
۵-۵-۳ مثال سوم ۱۰۴
۵-۶ بررسی نتایج حاصله ۱۰۶
۵-۶-۱ بررسی علت واگرا شدن نتایج حاصل از الگوریتم ۱۰۶
۵-۶-۲ بررسی نتایج حاصل از مثال‌ها ۱۰۸
۵-۷ بررسی مدلسازی الگوریتم در حالت سه بعدی ۱۰۹
فصل ۶ .... ۱۱۳
۶-۱ نتیجه گیری ۱۱۳
۶-۲ پیشنهادات ۱۱۶

فهرست اشکال

شکل ۲-۱: یک مدل بلوکی که به بلوک‌های کوچکتر تقسیم بندی شده است. ۹
شکل ۲-۲: الگوریتم کلی روش مخروط شناور (کاکایی، ۱۳۸۴)... ۱۱
شکل ۲-۳: مثالی از مدل بلوکی اقتصادی. ۱۲
شکل ۲-۴: اولین بلوک مثبت واقع در سطر اول و ستون چهارم... ۱۲
شکل ۲-۵: بلوک با مخروط مثبت بعدی، بلوک (۲،۳) می‌باشد. ۱۲
شکل ۲-۶: بلوک مثبت و قابل استخراج بعدی، بلوک (۳،۴) می‌باشد. ۱۳
شکل ۲-۷: شکل نهایی پیت پس از استخراج مخروط حاصل از سه بلوک ذکر شده.................... ۱۳
شکل ۲-۸: یک مدل بلوکی اقتصادی دو بعدی فرضی.. ۱۵
شکل ۲-۹: طبقه (۰) به مدل بلوکی اولیه اضافه می‌شود. ۱۵
شکل ۲-۱۰: ارزش تجمعی بلوک‌های شکل ۲-۹، در هر بلوک نوشته می‌شود. ۱۵
شکل ۲-۱۱: ارزش تجمعی بلوک‌های شکل ۲-۱۰ برطبق رابطه (۲-۲)................................. ۱۶
شکل ۲-۱۲: محدوده نهایی برای شکل ۲-۸.. ۱۶
شکل ۲-۱۳: یک مدل بلوکی اقتصادی فرضی دو بعدی ۱۷
شکل ۲-۱۴: بلوک مثبت واقع در سطر اول و ستون دوم برداشته می‌شود. ۱۸
شکل ۲-۱۵: اختصاص مقادیر مثبت بلوک (۲،۳) به بلوک‌های منفی واقع در مخروط آن. ۱۸
شکل ۲-۱۶: اختصاص مقادیر مثبت بلوک (۲،۶) به بلوک‌های منفی واقع در مخروط آن............ ۱۸
شکل ۲-۱۷: باقی ماندن ارزش مثبت در مخروط بلوک (۲،۶) و استخراج آن......................... ۱۹
شکل ۲-۱۸: شروع از ابتدا و با مقادیر اصلی. ۱۹
شکل ۲-۱۹: ارزش صفر محدوده بهینه با حل این مدل توسط روش کوروبوف. ۲۰
شکل ۲-۲۰: اختصاص مقادیر مثبت بلوک‌های مثبت به بلوک‌های غیرمشترک_....................... ۲۱
شکل ۲-۲۱: اختصاص بقیه مقادیر بلوک‌های مثبت به بلوک‌های منفی واقع در مخروط آنها. ۲۱
شکل ۲-۲۲: الگوریتم روش مخروط شناور II. ۲۲
شکل ۲-۲۳: یک مدل بلوکی اقتصادی فرضی دو بعدی برای حل با روش مخروط شناور II. ۲۳
شکل ۲-۲۴: بلوک واقع در ردیف دوم و ستون پنجم جزء محدوده فرض می‌شود. ۲۳
شکل ۲-۲۵: بلوک واقع در ردیف دوم و ستون سوم برداشته می‌شود.................................. ۲۴
شکل ۲-۲۶: مخروط حاصل از بلوک (۳،۶) جزء محدوده معدن فرض می‌شود. ۲۴
شکل ۲-۲۷: الگوریتم روش مخروط شناور II اصلاح شده (کاکایی، ۱۳۸۵).......................... ۲۶
شکل ۲-۲۸: مدل بلوکی اقتصادی برای حل با روش مخروط شناور II اصلاح شده.................. ۲۶
شکل ۲-۲۹: مخروط حاصل از بلوک (۲،۲)، جزء محدوده فرض می‌شود.............................. ۲۷
شکل ۲-۳۰: بلوک‌های (۳،۵) و (۳،۶) برداشته می‌شوند. ۲۸
شکل ۲-۳۱: الگوریتم روش مخروط شناور II، اصلاحیه دوم (ارائه توسط آقای کاکایی). ۲۹

شکل ۲-۳۲: حل این مدل توسط روش مخروط شناور II، اصلاحیه دوم. ۲۹
شکل ۲-۳۳: محدوده بهینه متعلق به مدل بلوکی شکل ۲-۳۲. .... ۳۰
شکل ۲-۳۴: یک مدل بلوکی دو بعدی ۳۱
شکل ۲-۳۵: G (N , A)، مدل بلوکی به گرافی جهت دار تبدیل می‌شود. ۳۱
شکل ۲-۳۶: درختی که دارای دو شاخه با کلوژرهای ضعیف و قوی می‌باشد. ۳۲
شکل ۲-۳۷: پیت‌های تودرتو در روش پارامتری کردن. ۳۳
شکل ۲-۳۸: در این مدل؛ بلوک‌های دارای ماده معدنی با هاشور مشخص شده‌اند. ۳۵
شکل ۲-۳۹: یک شبکه ساده برای مدل بلوکی فرضی ارائه شده در قسمت بالای شکل. ۳۷
شکل ۳-۱: یک کولونی از مورچگان کوتاهترین مسیر را می‌یابند. ۴۴
شکل ۳-۲: فلوچارت الگوریتم کولونی مورچگان، برای یافتن جواب بهینه. ۴۵
شکل ۴-۱: در این شکل، حرکات به سمت چپ، بالا و پایین پاسخگو نیستند، ۶۷
شکل ۴-۲: محدودیت فنی استخراج ۶۸
شکل ۴-۳: باید از انحراف مورچه‌ها به سمت مناطق هاشور خورده جلوگیری کرد. ۶۸
شکل ۴-۴: یک مورچه ابتدا از سطر اول وارد مدل بلوکی می‌شود. ۶۹
شکل ۴-۵: ممکن است مورچه بلافاصله پس از ورود به مدل بلوکی از آن خارج شود ۶۹
شکل ۴-۶: پس از انتخاب هر ستون (راست)، مورچه باید به سمت چپ حرکت کند. ۶۹
شکل ۴-۷: یک مورچه نمی‌تواند در دو مرحله متوالی به سمت پایین حرکت کند. ۷۰
شکل ۴-۸: حرکت مورچه‌ها به سمت بالا، پایین و یا چپ، پس از هر حرکت به سمت چپ. ۷۰
شکل ۴-۹: انواع حرکت مورچه‌ها ۷۱
شکل ۴-۱۰: یک مدل بلوکی برای حل توسط الگوریتم کولونی مورچگان ۷۶
شکل ۴-۱۱: حالت تجمعی مدل شکل ۴-۱۰ (محدوده ممکن معدن با خطوط پررنگ). ۷۶
شکل ۴-۱۲: نتیجه جستجوی اولین مورچه تا سه مرحله از تکرار اول، با هاشور. ۷۷
شکل ۴-۱۳: موقعیت جاری جستجو ۷۸
شکل ۴-۱۴: بلوک‌های جستجو شده و وضعیت جستجو بر روی مدل بلوکی تجمعی. ۷۸
شکل ۴-۱۵: پیت جستجو شده توسط اولین مورچه از تکرار اول با ارزشی برابر با (۱-) ۷۹
شکل ۴-۱۶: روش ذخیره فرومون گذاشته شده توسط مورچه یابنده جواب ۷۹
شکل ۴-۱۷: مورچه‌ها می‌توانند با شروع از مبدا یک ستون را به تصادف انتخاب کنند. ۸۰
شکل ۴-۱۸: عدم توانایی جستجوی دو پیت کنار هم توسط الگوریتم کولونی مورچگان. ۸۱
شکل ۴-۱۹: حل مشکل شکل ۴-۱۸ با اضافه شدن یک سطر با بلوکهای دارای ارزش صفر. ۸۱
شکل ۵-۱: روندی که برنامه براساس آن کار کرده و توسعه یافته است. ۸۴
شکل ۵-۲: نمایی از حالت کلی برنامه توسعه یافته ۸۶
شکل ۵-۳: مرحله آماده سازی مدل بلوکی به منظور جلوگیری از خروج مورچه‌ها از مدل‌ها ۸۸
شکل ۵-۴: مدل بلوکی مورد استفاده برای ارزیابی تعدادی از پارامترهای الگوریتم ۹۴

شکل ۵-۵: مدل ۱ با سه ردیف و نه ستون ۱۰۰
شکل ۵-۶: مدل ۲ با پنج ردیف و دوازده ستون... ۱۰۱
شکل ۵-۷: محدوده بهینه متعلق به مدل بلوکی ارائه شده در شکل ۵-۶............................ ۱۰۲
شکل ۵-۸: نمایی از جواب برنامه توسعه یافته برای مدل ۲ .... ۱۰۴
شکل ۵-۹: مدل ۳ با سه ردیف و هفت ستون.. ۱۰۵
شکل ۵-۱۰: جواب بهینه مدل موجود در شکل ۵-۸ .... ۱۰۵
شکل ۵-۱۱: نتایج حاصل از برنامه توسعه یافته برای مدل ۳ .... ۱۰۵
شکل ۵-۱۲: پیت بهینه نهایی متعلق به مدل بلوکی موجود در شکل ۵-۴. .... ۱۰۷
شکل ۵-۱۳: اختلاف در یک بلوک با شکل ۵-۱۲، و پاسخ‌های کاملاً متفاوت. ۱۰۷
شکل ۵-۱۴: (راست) کل مسیرهای ممکن، (وسط) یکی از مسیرهایی که قبلاً طی شده_........ ۱۰۸
شکل ۵-۱۵: مسیر حرکت احتمالی یک مورچه، بلوک‌های هاشوردار خارج از مسیر جستجو....... ۱۱۰
شکل ۵-۱۶: در مسیر حرکت شکل ۵-۱۵، بلوک هاشور خورده جستجو نمی‌شود.................. ۱۱۰
شکل ۵-۱۷: محدوده نهایی بهینه احتمالی شکل ۵-۱۶. .... ۱۱۰
شکل ۵-۱۸: شباهت حرکت مورچه‌ها به مسیر موجود در مسئله‌ی فروشنده دوره گرد. ۱۱۱

فهرست جداول

جدول ۲-۱: خلاصه عملیات انجام گرفته توسط روش مخروط شناور II. ۲۵
جدول ۲-۲: خلاصه مراحل انجام گرفته توسط روش مخروط شناور II اصلاح شده. ۲۸
جدول ۲-۳: مراحل انجام گرفته برای تعیین محدوده نهایی معدن با روش مخروط شناور_ ۳۰
جدول ۵-۱: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۲، β=۰/۲،_...................... ۹۴
جدول ۵-۲: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۴، β=۰/۲،_...................... ۹۴
جدول ۵-۳: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۶، β=۰/۲،_...................... ۹۵
جدول ۵-۴: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۸، β=۰/۲،_...................... ۹۵
جدول ۵-۵: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۲، β=۰/۴،_...................... ۹۶
جدول ۵-۶: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۲، β=۰/۶،_...................... ۹۶
جدول ۵-۷: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۲، β=۰/۸،_...................... ۹۷
جدول ۵-۸: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۵، β=۰/۵،_...................... ۹۷
جدول ۵-۹: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۱، β=۰/۹،_...................... ۹۸
جدول ۵-۱۰: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۹، β=۰/۱،_...................... ۹۸
جدول ۵-۱۱: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۶، β=۰/۴،_...................... ۹۹
جدول ۵-۱۲: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۴، β=۰/۶،_...................... ۹۹
جدول ۵-۱۳: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۱، در هر تکرار ۱۰ مورچه...... ۱۰۰
جدول ۵-۱۴: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۱، در هر تکرار ۲۰ مورچه...... ۱۰۱
جدول ۵-۱۵: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۲، در هر تکرار ۱۰ مورچه...... ۱۰۲
جدول ۵-۱۶: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۲، در هر تکرار ۲۰ مورچه...... ۱۰۳
جدول ۵-۱۷: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۲، با ۱۵ تکرار_................ ۱۰۳
جدول ۵-۱۸: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۲، با ۲۰ تکرار_................ ۱۰۴
جدول ۵-۱۹: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۳، با ۱۰ تکرار و_ ۱۰۶

فصل اول

مقدمه

فصل ۱

۱-۱ مقدمه

دستیابی به تمامی پیشرفت‌های امروزه بدون وجود مواد اولیه کافی غیرممکن بود. بسیاری از دستاوردها به کمک رویکرد آزمون و خطا حاصل شده‌اند. برای رسیدن به بسیاری از اهداف علمی، دانشمندان ناچار به ابداع مسیرهای جدیدی در علم شده‌اند که شاید از ابتدا انتظار آن‌ها نمی‌رفت. در کنار همه‌ی این پیشرفت‌ها، انجام کارها ساده‌تر شد ولی همیشه این سوال وجود دارد که آیا ممکن است باز هم هزینه‌ها پایین‌تر بیایند؟

همیشه پیشرفت علم به پیچیده شدن سیستم‌ها و فرایندها منجر نشده است، بلکه در بسیاری موارد، به ساده شدن و کوتاه کردن فرایند انجامیده است. ولی همچنان تقاضا برای مواد اولیه بالاست و می‌توان گفت شاید تنها موردی که همزمان با پیشرفت تکنولوژی رشد می‌کند، تقاضا برای مواد اولیه ارزان قیمت است. از ابتدای زندگی بشر بر روی زمین این تقاضا هیچگاه کاهش نداشته است؛ حتی در زمان رکودهای بزرگ اقتصادی و صنعتی. در پی پاسخ به این نیاز، علم معدنکاری به استفاده از ماشین‌آلات بزرگ، سنگین و گران قیمت روی آورد؛ ولی باز هم کافی نبوده است. معدنکاران به استفاده از مواد منفجر شونده روی آوردند تا بلکه با هزینه‌ی اندک این مواد، مخارج عملیات استخراج را کاهش دهند. ولی باز هم امکان داشت که بخشی از مواد بدون عیار کافی استخراج شوند. برای این

منظور و برای بالا بردن دقت در امر استخراج، انواع نقشه‌ها تهیه شدند تا یاری دهنده طراحان معادن باشند. در عین حال نقشه بدون یک طرح کلی برای استفاده، بی‌فایده است. پس روش‌هایی برای طراحی ایجاد شدند. هدف این روش‌ها همواره استخراج بیشترین مقدار ماده‌ی معدنی با کمترین هزینه می‌باشد.

با بزرگ شدن ابعاد معدن و گسترش آن‌ها به عمق، یافتن یک سیستم طراحی که بتواند کار طراحی را در هر سه بُعد بطور همزمان با هم انجام دهد، تبدیل به چالشی بزرگ برای معدنکاران شد. از یک طرف به دلیل ابعاد بزرگ معدن‌ها، ممکن بود یک عملیات استخراج اقتصادی، به اشتباه غیر اقتصادی شود و از طرف دیگر راه‌حل‌های مناسبی برای طراحی بهترین محدوده نهایی وجود نداشت.

پس محققان دست بکار شدند و روش‌هایی را برای یافتن این محدوده ابداع کردند. بیشتر این روش‌ها به دلیل ابتکاری بودن نتوانستند موفقیتی بدست آورند و یا اینکه به دلیل پیچیده بودن، چندان مورد توجه قرار نگرفتند.

تنها بعضی از این روش‌ها می‌توانستند به جواب بهینه واقعی برسند، که آنها نیز علاوه بر پیچیده بودن به وقت زیادی برای حل آن هم توسط رایانه نیاز داشتند.

در چند سال اخیر برای مسائل بهینه‌سازی ریاضی، روش‌هایی معرفی شده‌اند که روش‌های ابتکاری[1] یا روش‌های تکاملی[2] نامیده می‌شوند. این روش‌ها قادرند تا در زمانی اندک به جوابی مطلوب و قابل قبول برسند. یعنی حتی اگر به دلایلی نتوانستند بهترین جواب یا جواب بهینه اصلی را بیابند، یک جواب نزدیک به حالت بهینه ارائه می‌دهند. از جمله این روش‌ها، روش‌های الگوریتم ژنتیک، الگوریتم کولونی مورچگان، روش شبیه‌سازی تبریدی و چند روش دیگر را می‌توان نام برد.

---

[1] - Heuristic algorithms
[2] - Evolution algorithms

۱-۲ هدف پایان نامه

هدف از نگارش این پایان نامه یافتن محدوده نهایی معادن روباز با بکارگیری الگوریتم کولونی مورچگان می‌باشد. از آنجا که تاکنون روش‌های مختلفی که برای تعیین کردن محدوده نهایی معادن روباز پیشنهاد شده‌اند، دارای نواقص و کاستی‌هایی می‌باشند، در این پایان نامه از روش الگوریتم کولونی مورچگان استفاده شده است.

هدف این است که دیدی کلی در رابطه با بکارگیری این روش ایجاد شود. همچنین موانع و مشکلات کاربرد این روش به منظور بررسی مسئله‌ی گفته شده نیز مورد بررسی قرار گرفته است.

در مجموع می‌توان هدف این پایان نامه را ارزیابی از کاربرد و چگونگی کاربرد روش الگوریتم کولونی مورچگان در مسائل مرتبط با معدنکاری و بخصوص مسئله یافتن محدوده نهایی معادن روباز دانست.

۱-۳ سازمان‌دهی پایان نامه

این پایان نامه که به یافتن محدوده نهایی معادن روباز با کمک الگوریتم کولونی مورچگان اختصاص دارد، به صورت زیر فصل‌بندی شده است:

در فصل اول، کلیات و هدف پایان نامه ذکر گردیده است.

در فصل دوم، تعاریف، اصطلاحات و روش‌های تعیین محدوده نهایی معادن روباز که تاکنون توسط برخی از محققان پیشنهاد شده‌اند ذکر می‌گردد. در این فصل سعی شده که تا حد ممکن هر روش با مثال آورده شود.

در فصل سوم مبانی الگوریتم کولونی مورچگان آورده شده است. از آنجا که این پایان نامه باید بتواند برای مهندسان معدن مفید واقع شود، سعی شده است تا توضیحات مربوط به این الگوریتم به سادگی بیان شوند و از توضیح ریاضی صِرف پرهیز شده است.

در فصل چهارم اصول طراحی محدوده نهایی معدن توسط الگوریتم کولونی مورچگان آورده شده است. در این فصل روش‌ها، نتایج، آثار و چگونگی فرمول‌بندی مسئله به گونه‌ای که قابل حل توسط الگوریتم مورچگان باشد ذکر شده است.

در فصل پنجم نیز برنامه‌نویسی و نتایج حاصل از آن آورده خواهند شد. در این فصل تعدادی مثال با استفاده از الگوریتم کولونی مورچگان حل شده و نتایج مثال‌ها با روش‌های دیگری نظیر روش مخروط شناور و روش برنامه ریزی پویا مقایسه شده است.

فصل ششم به نتیجه گیری از پایان نامه اختصاص خواهد داشت. در این فصل پیشنهاداتی در رابطه با کاربردهای آتی الگوریتم کولونی مورچگان نیز ارائه شده است که می‌تواند برای تحقیقات آینده مفید باشد.

فصل دوم

روش‌های تعیین محدوده نهایی معادن روباز

فصل ۲

۲-۱ مقدمه

به دلیل آنکه استخراج مواد معدنی به روش روباز به سرمایه‌گذاری و زمان بسیار زیادی نیاز دارد، طراحی صحیح و بهینه‌سازی مداوم این طراحی از اهمیت فوق‌العاده‌ای برخوردار است. پیش از شروع عملیات استخراج نیاز است تا محدوده نهایی معدن مشخص شود. این مورد از آن جهت اهمیت دارد که برای تعیین میزان ذخیره قابل استخراج و میزان باطله برداری و برنامه‌ریزی تولید، طراحی مسیرهای دسترسی، مکان‌یابی ساختمان‌ها و کارخانه‌ها و محل انباشت باطله، در دست داشتن این محدوده بسیار ضروری است. برای تعیین این محدوده، پارامترهای بسیاری مورد نیاز می‌باشند که به دلیل تعداد زیاد این پارامترها و نیاز معدن به طراحی مجدد در طول عمر خود، استفاده از رایانه ضروری می‌باشد.

در این فصل روش‌هایی که تاکنون برای تعیین و طراحی محدوده نهایی معادن روباز معرفی شده‌اند یا مورد استفاده قرار گرفته‌اند بطور مختصر مورد مرور قرار می‌گیرد. در بعضی موارد برای توضیح بیشتر روش از یک مثال نیز استفاده شده است.

۲-۱-۱ طراحی محدوده نهایی معادن روباز

محدوده نهایی معدن، محدوده‌ای است که پارامتری از آن ماکزیمم و بهینه است. برای ماکزیمم سازی از پارامتر سود و یا از NPV [1] استفاده می‌شود. در واقع طراحی محدوده نهایی یک معدن باید به منظور به حداکثر رساندن ارزش خالص فعلی صورت گیرد ولی همانگونه که توسط ویتل بیان شده است: "محدوده معدن با بیشترین ارزش خالص فعلی را نمی‌توان تعیین نمود مگر آنکه ارزش خالص فعلی بلوک‌ها تعیین شود؛ ارزش خالص فعلی بلوک‌ها را نمی‌توان تعیین نمود مگر آنکه ترتیب استخراج (برنامه ریزی تولید) آنها معلوم باشد و ترتیب استخراج بلوک‌ها را نمی‌توان تعیین کرد مگر آنکه محدوده نهایی معدن طراحی شده باشد"، (Whittle, 1988). برای رفع این مشکل پیشنهاد شده است که ابتدا محدوده نهایی با هدف حداکثر کردن سود انجام شود و سپس برنامه ریزی تولید در داخل این محدوده با هدف حداکثر کردن NPV صورت گیرد (کاکایی، ۱۳۸۴) و (Whittle, 1988).

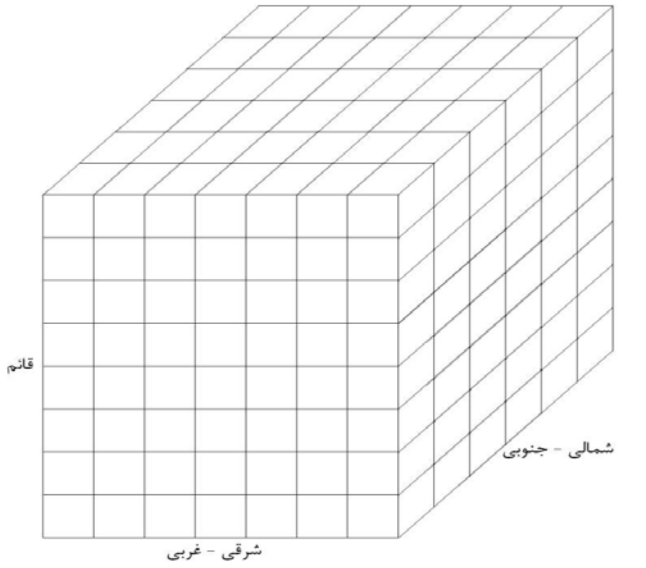
به دلیل حجم زیاد عملیات برای تعیین محدوده نهایی معدن، استفاده از رایانه امری اجتناب ناپذیر می‌باشد. برای استفاده از رایانه در امور طراحی، مفهوم عیار، ساختار زمین و پارامترهای اقتصادی باید به گونه‌ای برای رایانه تعریف شوند. استفاده از رایانه برای طراحی محدوده بهینه با ساختن یک مدل بلوکی از کانسار شروع می‌شود. برای این منظور ابتدا ذخیره را به صورت یک بلوک بزرگ به گونه‌ای درنظر می‌گیرند که تمامی نواحی ماده معدنی را دربرگیرد. در مرحله بعد این بلوک بزرگ را به بلوک‌های کوچکتر تقسیم کرده (مطابق شکل ۲-۱) و سپس برای هر بلوک، عیار ماده معدنی یا مشخصات دیگر آن با استفاده از روش‌های عکس فاصله، زمین آمار یا با استفاده از اطلاعات اکتشافی موجود تخمین زده می‌شود. از آنجاییکه ارتفاع بلوک معمولاً به اندازه ارتفاع پله درنظر گرفته می‌شود، ابعاد افقی بلوک با توجه به اطلاعات اکتشافی و روش‌های تخمین طراحی می‌شوند و برای هر بلوک یک مختصات تعیین می‌شود.

---

[1]- Net Present Value (ارزش خالص فعلی)

به یک مدل بلوکی که به هر بلوک آن مختصات بلوک و عیار ماده معدنی اختصاص داده شده است، مدل بلوکی زمین شناسی گفته می‌شود. از روی مدل بلوکی زمین شناسی، مدل بلوکی اقتصادی ساخته خواهد شد؛ یعنی برای هر بلوک، ارزش اقتصادی آن مشخص می‌شود (کاکایی، ۱۳۸۴).

در این پایان نامه هر کجا که صحبت از مدل بلوکی باشد، منظور مدل بلوکی اقتصادی است. برای مثال‌ها نیز از مدل‌های بلوکی اقتصادی آماده استفاده شده است. مدل بلوکی برای تعیین محدوده بهینه نهایی معدن و همچنین برنامه ریزی تولید مورد استفاده قرار می‌گیرد.



شکل ۲-۱: یک مدل بلوکی که به بلوکهای کوچکتر تقسیم بندی شده است.

الگوریتم‌های زیادی برای تعیین محدوده بهینه نهایی در معادن روباز ارائه شده‌اند که هدف اصلی همه این روش‌ها پیدا کردن مجموعه بلوک‌هایی است که با استخراج آنها سود بدست آمده حداکثر شود. کیم در سال ۱۹۷۸ این الگوریتم‌ها را به دو دسته بهینه و ابتکاری تقسیم‌بندی نموده است. دسته اول (الگوریتم‌های بهینه)، الگوریتم‌هایی می‌باشند که برای آنها اثبات ریاضی وجود دارد نظیر روش لرچ و گروسمن و نیز روش برنامه‌ریزی پویای دوبُعدی. دسته دوم از این روش‌ها (ابتکاری) دارای

اثبات ریاضی برای بهینه بودن جواب نبوده و حتی ممکن است نتوانند به جوابی قابل قبول دست یابند (Kim & Young, 1978).

مهم‌ترین الگوریتم‌هایی که در این پایان نامه در مورد آنها صحبت خواهد شد عبارتند از:

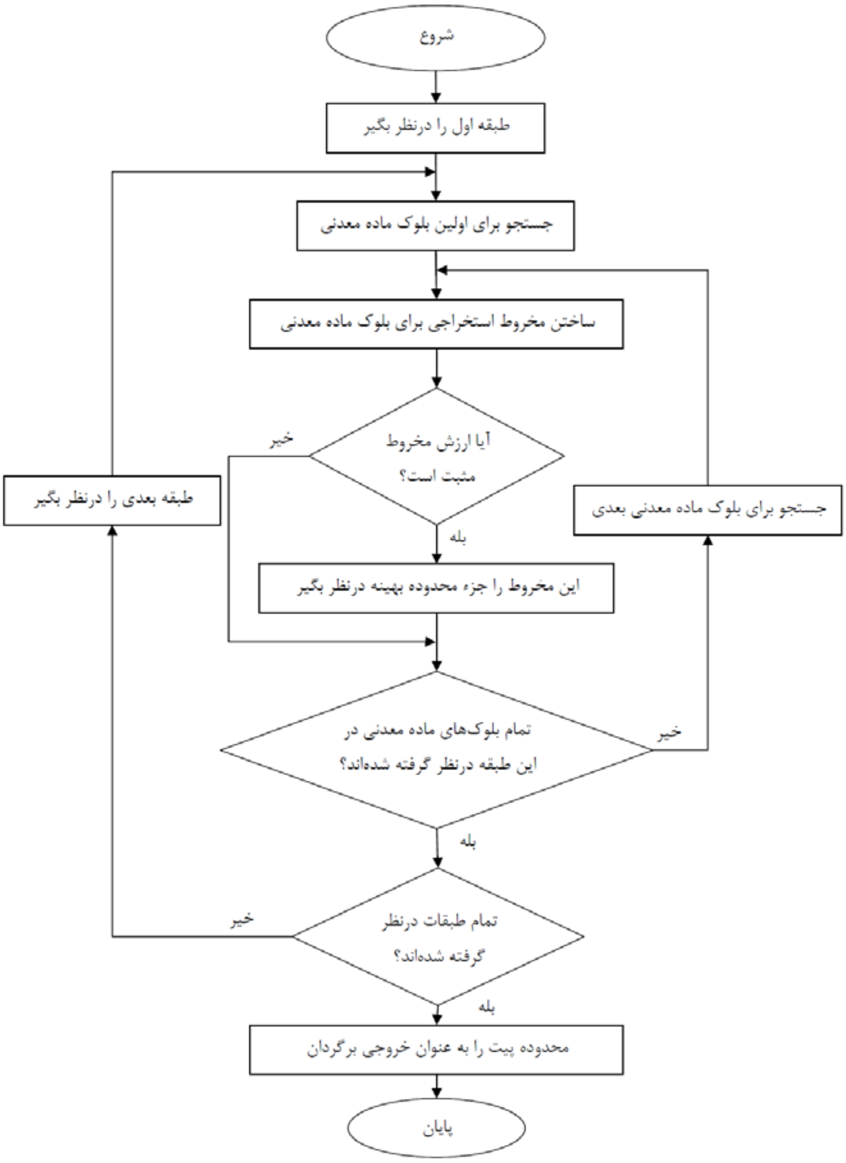
- روش مخروط شناور و تصحیح‌های آن
- روش مخروط شناور II و تصحیح‌های آن
- روش کوروبوف و تصحیح‌های آن
- روش شبکه و حداکثر جریان
- روش برنامه‌ریزی پویا
- روش لرچ و گروسمن
- روش پارامتری کردن

۲-۲ روش مخروط شناور[1]

این روش در سال ۱۹۶۶ توسط کارلسون و اریکسو ن[2] ارائه شده است (Carlson & Erickson, 1966). در این روش برای هر بلوک مثبت (ماده معدنی) یک مخروط معکوس با توجه به زاویه شیب پایداری معدن طوری ساخته می‌شود که راس آن در بلوک ماده معدنی باشد. سپس ارزش بلوک‌های واقع در داخل آن محاسبه شده و درصورتیکه مجموع این ارزش‌ها مثبت باشد تمام بلوک‌های واقع در داخل آن جزء محدوده معدن فرض می‌شوند؛ در غیر این صورت جستجو برای بلوک‌های مثبت دیگر ادامه می‌یابد (Roditis, 1993). این روش جستجوی خود را از طبقه اول ماده معدنی شروع کرده و پس از جستجوی هر طبقه به طبقه بعدی می‌رود. الگوریتم روش مخروط شناور در شکل ۲-۲ نشان داده شده است.

---

[1] - Floating Cone Method
[2] - Carlson and Ericsson



شکل ۲-۲: الگوریتم کلی روش مخروط شناور (کاکایی، ۱۳۸۴)

این روش را با یک مثال ساده بهتر می‌توان توضیح داد. به عنوان مثال مدل بلوکی شکل ۲-۳ را ملاحظه کنید. در این مدل ۴ بلوک ماده معدنی وجود دارد. الگوریتم از طبقه اول شروع و جستجو برای بلوک ماده معدنی انجام خواهد شد.

	۱	۲	۳	۴	۵	۶	۷
۱	-۱	-۱	-۱	+۲	-۱	-۱	-۱
۲	-۲	-۲	+۳	-۲	+۱	-۲	-۲
۳	-۳	-۳	-۳	+۴	-۳	-۳	-۳

شکل ۲-۳: مثالی از مدل بلوکی اقتصادی.

با شروع از سطر اول، و با حرکت از سمت چپ به راست، اولین بلوک مثبت، بلوک (۱،۴) می‌باشد (بصورت شکل ۲-۴) که با هاشور نشان داده شده است. به دلیل مثبت بودن ارزش این بلوک، جزء محدوده بهینه درنظر گرفته خواهد شد.

	۱	۲	۳	۴	۵	۶	۷
۱				+۲			
۲							
۳							

شکل ۲-۴: اولین بلوک مثبت واقع در سطر اول و ستون چهارم.

در سطر اول بلوک ماده معدنی دیگری وجود ندارد، پس الگوریتم در سطر دوم جستجوی خود را ادامه می‌دهد و به بلوک (۲،۳) می‌رسد که با فرض استخراج بلوک (۱،۴) در مرحله قبل، هم اکنون مخروط حاصل از بلوک (۲،۳) مثبت می‌باشد (با ارزش۱+). این مخروط و بلوک آن نیز جزئی از محدوده بهینه نهایی درنظر گرفته خواهند شد (شکل ۲-۵).

	۱	۲	۳	۴	۵	۶	۷
۱		-۱	-۱				
۲			+۳				
۳							

شکل ۲-۵: بلوک با مخروط مثبت بعدی، بلوک (۲،۳) می‌باشد.

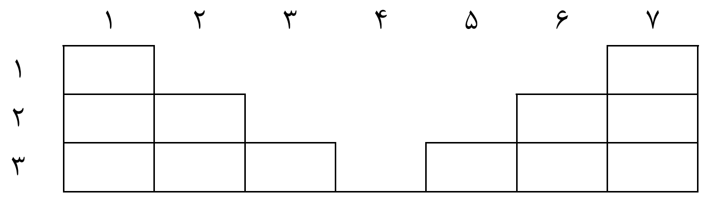
بلوک ماده معدنی بعدی، بلوک (۲،۵) می‌باشد که مخروط حاصل از آن مثبت نمی‌باشد، پس الگوریتم آن را رها خواهد کرد. در سطر سوم تنها بلوک دارای ارزش مثبت، بلوک (۳،۴) است که

دارای مخروطی با ارزش ۱+ خواهد بود (شکل ۲-۶)، پس این بلوک و مخروط حاصل از آن نیز جزء محدوده معدن فرض خواهند شد.

	۱	۲	۳	۴	۵	۶	۷
۱					-۱	-۱	
۲				-۲	+۱		
۳				+۴			

شکل ۲-۶: بلوک مثبت و قابل استخراج بعدی، بلوک (۳،۴) می‌باشد.

بلوک ماده معدنی دیگری وجود ندارد. و تا اینجا ارزش محدوده مثبت بوده است. با استخراج این سه بلوک ماده معدنی و مخروط‌های حاصل از آنها ارزش کل محدوده پیت برابر با ۴+ شده و شکل نهایی پیت به صورت شکل ۲-۷ خواهد بود.



شکل ۲-۷: شکل نهایی پیت پس از استخراج مخروط حاصل از سه بلوک ذکر شده.

با توجه به مطالب ذکر شده می‌توان مزایای روش مخروط شناور را اینگونه برشمرد (کاکایی، ۱۳۸۴):

- الگوریتمی بسیار ساده و ارزان می‌باشد؛
- نوشتن برنامه کامپیوتری برای آن بسیار ساده است؛
- زمان بسیار کوتاهی برای رسیدن به جواب نیاز دارد؛
- استفاده از شیب‌های متغیر دیواره معدن در این الگوریتم به سهولت انجام می‌گیرد.

در کنار این مزایا، روش مخروط شناور دارای معایبی نیز می‌باشد که به این صورت می‌باشند:

- در برخی از مدل‌ها قادر به بدست آوردن محدوده‌ی بهینه نیست؛

- عدم وجود اثبات ریاضی برای این روش؛
- جهت جستجو در ارزش مخروط بهینه دخیل است.

۳-۲ روش برنامه ریزی پویا[1]

روش برنامه ریزی پویا اولین بار توسط بلمن[2] در سال ۱۹۵۷ معرفی شده و یکی از پر استفاده‌ترین روش‌های تحقیق در عملیات بوده که برای حل مسائلی بکار برده می‌شود که بتوان آن را به چند مرحله تقسیم کرده و برای هر مرحله جواب بهینه پیدا نمود. این روش در سال ۱۹۶۵ توسط لرچ و گروسمن[3] برای تعیین محدوده‌ی بهینه معدن (حالت دو بعدی) بکار برده شد (Lerchs & Grossman, 1965). بعد از آن جانسون، شارپ و کانینزبرگ سعی کردند با استفاده از این روش محدوده بهینه را در حالت سه بعدی پیدا نمایند که این کوشش ناموفق بود (Johnson & Sharp, 1971).

در این روش ابتدا یک ردیف با ارزش صفر به مدل بلوکی اضافه می‌شود. سپس ارزش تجمعی ستونی بلوک‌ها محاسبه می‌شود (مطابق رابطه ۲-۱). سپس با شروع از ستون سمت چپ و انجام عملیات از طبقه بالا به طبقه پایین برای محاسبه ارزش تجمعی کلی، مطابق با رابطه ۲-۲ مجموع ارزش تجمعی بلوک با بیشترین مقدار ارزش تجمعی کلی یکی از سه بلوک واقع در ستون سمت چپ آن (ستون سمت چپ و یک طبقه بالاتر، ستون سمت چپ و در همان طبقه، ستون سمت چپ و در یک طبقه پایینتر) محاسبه می‌گردد. پس از انجام این محاسبات، با شروع از اولین ستون سمت راست در اولین سطر، فلش‌هایی به سمت بلوک‌های مجاور (در سمت چپ بلوک جاری) و با بیشترین ارزش رسم می‌شوند که با دنبال کردن این فلش‌ها محدوده بهینه بدست خواهد آمد. این روش را می‌توان با یک مثال ساده بهتر توضیح داد. مدل بلوکی شکل ۲-۸ را درنظر بگیرید.

---

[1] - Dynamic Programming
[2] - Bellman
[3] - Lerchs & Grossman

	۱	۲	۳	۴	۵	۶	۷
۱	-۱	-۱	-۳	-۱	-۱	-۱	-۱
۲	-۲	+۴	-۲	-۲	-۲	-۲	-۲
۳	-۳	-۳	-۳	-۳	+۱۲	-۳	-۳

شکل ۲-۸: یک مدل بلوکی اقتصادی دو بعدی فرضی.

گام اول: در اولین گام یک طبقه به ارزش صفر به مدل اضافه می‌شود (شکل ۲-۹).

	۱	۲	۳	۴	۵	۶	۷
۰	۰	۰	۰	۰	۰	۰	۰
۱	-۱	-۱	-۳	-۱	-۱	-۱	-۱
۲	-۲	+۴	-۲	-۲	-۲	-۲	-۲
۳	-۳	-۳	-۳	-۳	+۱۲	-۳	-۳

شکل ۲-۹: طبقه (۰) به مدل بلوکی اولیه اضافه می‌شود.

گام دوم: با شروع از طبقه اول و از سمت چپ، ارزش تجمعی بلوک‌ها برطبق رابطه ۲-۱ محاسبه می‌شوند، که در شکل ۲-۱۰ نشان داده شده است.

(رابطه ۲-۱)

$$M_{ij} = \sum_{k=0}^{i} m_{kj}$$

M_{ij} : ارزش تجمعی بلوک واقع در ردیف i و ستون j؛

m_{kj} : ارزش بلوک ردیف k و ستون j.

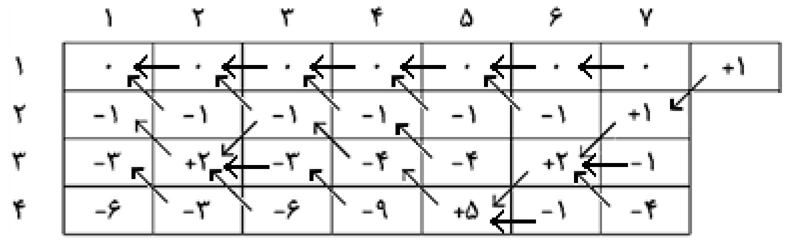
M	۱	۲	۳	۴	۵	۶	۷
۱	۰	۰	۰	۰	۰	۰	۰
۲	-۱	-۱	-۳	-۱	-۱	-۱	-۱
۳	-۳	+۳	-۵	-۳	-۳	-۳	-۳
۴	-۶	۰	-۸	-۶	+۹	-۶	-۶

شکل ۲-۱۰: ارزش تجمعی بلوک‌های شکل ۲-۹، در هر بلوک نوشته می‌شود.

گام سوم: از سمت چپ و از بالا به پایین شروع کرده و ارزش تجمعی کلی بلوک (P_{ij}) را براساس رابطه ۲-۲ محاسبه می‌شوند (مطابق با شکل ۲-۱۱).

$$P_{ij} = M_{ij} + Max\begin{cases} P_{i-1,j-1} \\ P_{i,j-1} \\ P_{i+1,j-1} \end{cases} \quad \text{(رابطه ۲-۲)}$$

سپس در تمام بلوک‌ها فلش‌هایی رسم می‌شوند که به بلوک با بیشترین ارزش در سمت چپ خود (مطابق با رابطه ۲-۲) اشاره دارند.



شکل ۲-۱۱: ارزش تجمعی بلوک‌های شکل ۲-۱۰ برطبق رابطه (۲-۲).

گام چهارم: بلوکی که دارای بیشترین ارزش تجمعی کلی و مثبت در ردیف صفر است پیدا کرده و با دنبال کردن فلش‌ها، محدوده بهینه بدست می‌آید.

پس از انجام مراحل فوق، ردیف صفر را پاک کرده و محدوده نهایی مشخص می‌شود (مطابق شکل ۲-۱۲).

	۱	۲	۳	۴	۵	۶	۷
۱							
۲	-۲						-۲
۳	-۳	-۳	-۳	-۳		-۳	-۳

شکل ۲-۱۲: محدوده نهایی برای شکل ۲-۸.

روش برنامه ریزی پویا در حالت دو بعدی به جواب بهینه می‌رسد (Wright,1986). در حالت سه بعدی درست کردن محدودیت فنی بسیار مشکل است. این روش برای روش‌های دستی و برای اینکه از مقاطع بتوان محدوده‌ی پیت را بدست آورد بکار می‌رود (کاکایی، ۱۳۸۴).

۲-۴ روش کوروبوف[1]

این روش در سال ۱۹۷۴ توسط دیوید، دود و کوروبوف[2] ارائه شد و شبیه به مخروط شناور می‌باشد.

روش کار این الگوریتم به این صورت است که برای هر بلوک ماده معدنی یک مخروط معکوس به گونه‌ای ساخته می‌شود که راس مخروط در بلوک ماده معدنی باشد. سپس در داخل هر مخروط مقادیر مثبت به مقادیر منفی اختصاص داده می‌شود تا اینکه هیچ بلوکی با ارزش منفی باقی نماند و یا اینکه تمام مقادیر مثبت اختصاص داده شود. درصورتیکه ارزش بلوکی که برای آن مخروط ساخته شده مثبت باقی بماند، تمام بلوک‌های واقع در داخل آن جزء محدوده معدن درنظر گرفته می‌شود. اگر مخروط مذکور خالی باشد، یعنی آنکه فاقد بلوکی دیگر باشد، جستجو برای بلوک‌های مثبت دیگر ادامه پیدا می‌کند؛ در غیر اینصورت الگوریتم از ابتدا با مقادیر اصلی بلوک‌ها برای بلوک‌های باقیمانده ادامه پیدا می‌کند (کاکایی، ۱۳۸۴) این الگوریتم با ذکر یک مثال بیشتر توضیح داده می‌شود؛ مدل بلوکی شکل ۲-۱۳ را درنظر بگیرید.

	۱	۲	۳	۴	۵	۶	۷
۱	-۱	+۱	-۱	-۱	-۱	-۱	-۱
۲	-۲	-۲	+۱	-۲	-۲	+۴	-۲
۳	-۳	-۳	-۳	-۳	-۳	-۳	-۳

شکل ۲-۱۳: یک مدل بلوکی اقتصادی فرضی دو بعدی

[1] - Korobov Method
[2] - David, Dowd and Korobov

در اولین سطر، بلوک (۱،۲) مثبت است و جزو محدوده بهینه درنظر گرفته می‌شود که در شکل ۲-۱۴ قسمت استخراج شده با هاشور نشان داده شده است.

	۱	۲	۳	۴	۵	۶	۷
۱	-۱		-۱	-۱	-۱	-۱	-۱
۲	-۲	-۲	+۱	-۲	-۲	+۴	-۲
۳	-۳	-۳	-۳	-۳	-۳	-۳	-۳

شکل ۲-۱۴: بلوک مثبت واقع در سطر اول و ستون دوم برداشته می‌شود.

بلوک مثبت بعدی در ردیف دوم و ستون سوم قرار دارد، با اختصاص دادن اعداد مثبت این بلوک به بلوک‌های منفی روی آن ارزش دو بلوک صفر شده و یک بلوک با ارزش منفی باقی خواهد ماند (شکل ۲-۱۵)، و الگوریتم به جستجو برای یافتن یک بلوک ماده معدنی دیگر ادامه می‌دهد.

	۱	۲	۳	۴	۵	۶	۷
۱	-۱		۰	-۱	-۱	-۱	-۱
۲	-۲	-۲	۰	-۲	-۲	+۴	-۲
۳	-۳	-۳	-۳	-۳	-۳	-۳	-۳

شکل ۲-۱۵: اختصاص مقادیر مثبت بلوک (۲،۳) به بلوک‌های منفی واقع در مخروط آن.

بلوک مثبت بعدی در ردیف دوم و ستون ششم قرار دارد که با اختصاص دادن اعداد مثبت موجود در این بلوک به بلوک‌های منفی رویی آن، باز هم ارزش بلوک مثبت باقی خواهد ماند (شکل ۲-۱۶).

	۱	۲	۳	۴	۵	۶	۷
۱			۰	-۱	۰	۰	۰
۲			۰			+۱	
۳							

شکل ۲-۱۶: اختصاص مقادیر مثبت بلوک (۲،۶) به بلوک‌های منفی واقع در مخروط آن.

به دلیل مثبت باقی ماندن ارزش این بلوک پس از اختصاص دادن مقادیر مثبت، جزء محدوده درنظر گرفته می‌شود (شکل ۲-۱۷).

	۱	۲	۳	۴	۵	۶	۷
۱			۰	-۱			
۲			۰				
۳							

شکل ۲-۱۷: باقی ماندن ارزش مثبت در مخروط بلوک (۲،۶) و استخراج آن.

به دلیل آنکه مخروط حاصل خالی نمی‌باشد، الگوریتم از ابتدا و با مقادیر اصلی شروع به جستجوی مدل بلوکی می‌نماید (شکل ۲-۱۸).

	۱	۲	۳	۴	۵	۶	۷
۱	-۱		-۱	-۱			
۲	-۲	-۲	+۱	-۲	-۲		-۲
۳	-۳	-۳	-۳	-۳	-۳	-۳	-۳

شکل ۲-۱۸: شروع از ابتدا و با مقادیر اصلی.

با اختصاص دادن مقادیر مثبت بلوک (۲،۳) به بلوک‌های رویی آن یعنی بلوک‌های (۱،۳) و (۱،۴)، ارزش بلوک (۲،۳) به همراه بلوک (۱،۳) صفر خواهد شد و ارزش بلوک (۲،۴) منفی باقی خواهد ماند. در نتیجه این بلوک جزو محدوده معدن نمی‌باشد (محدوده نهایی همانند شکل ۲-۱۸ بوده و ارزش این محدوده برابر با ۲+ می‌باشد).

روش کوروبوف تمام مزایای روش مخروط شناور را دارد، منجمله اینکه نیاز به زمان اندکی برای رسیدن به جواب دارد، اما ضعف این روش این است که در بعضی مدل‌ها محدوده‌ی بهینه با ارزش صفر یا منفی پیدا می‌کند (کاکایی، ۱۳۸۴). به عنوان مثال مدل بلوکی شکل ۲-۱۹ را درنظر بگیرید که با این روش، محدوده بهینه‌ای با ارزش صفر خواهد داشت، درصورتیکه با استفاده از روش اصلاح شده کوروبوف این مدل جوابی نخواهد داشت.

۱	۲	۳	۴	۵	۶	۷
۱	-۱	-۱	-۱	-۱	-۱	-۱
۲	-۱	-۱	-۱	-۱	-۱	-۱
۳	-۱	-۱	+۳	+۷	-۱	-۱

شکل ۲-۱۹: ارزش صفر محدوده بهینه با حل این مدل توسط روش کوروبوف.

۲-۵ روش اصلاح شده کوروبوف[1]

این روش که توسط دود و اُنور[2] پیشنهاد شده است شبیه به الگوریتم کوروبوف می‌باشد، با این تفاوت که برای اختصاص مقادیر مثبت به بلوک‌های منفی ابتدا بررسی می‌شود که آیا بلوک‌های ماده معدنی دارای بلوک‌های مشترک در مخروط‌هایشان هستند یا خیر؟ و بعد هنگام اختصاص دادن ارزش بلوک ماده معدنی (مثبت) به بلوک‌های منفی ابتدا به بلوک‌های غیر مشترک و بعد از آن به بلوک‌های مشترک اختصاص داده می‌شود (Dowd & Onur, 1993).

به عنوان مثال مدل بلوکی شکل ۲-۱۹ را درنظر بگیرید. اولین بلوک ماده معدنی، بلوک واقع در سطر سوم و ستون سوم می‌باشد، که با اختصاص دادن مقادیر مثبت آن به مقادیر منفی رویی آن (که با مخروط حاصل از بلوک (۳،۴) مشترک نیستند)، ارزش بلوک (۳،۳) برابر با ۱+ می‌شود (شکل ۲-۲۰). سپس الگوریتم کار خود را برای جستجوی یک بلوک ماده معدنی دیگر ادامه می‌دهد و به بلوک (۳،۴) رسیده و باز هم از مقادیر مثبت بلوک (۳،۴) برداشته و به مقادیر منفی بلوک‌های منفی رویی آن که با مخروط حاصل از بلوک (۳،۳) مشترک نیستند می‌دهد. در نتیجه ارزش بلوک (۳،۴) به ۵+ رسیده و دو بلوک غیرمشترک با مخروط بلوک (۳،۳) صفر می‌شوند (شکل ۲-۲۰).

[1] - Revised Korobov Method
[2] - Onur

	۱	۲	۳	۴	۵	۶
۱	۰	-۱	-۱	-۱	-۱	۰
۲	-۱	۰	-۱	-۱	۰	-۱
۳	-۱	-۱	+۱	+۵	-۱	-۱

شکل ۲-۲۰: اختصاص مقادیر مثبت بلوک‌های مثبت به بلوک‌های غیرمشترک واقع در مخروط آنها.

در این مرحله دیگر بلوک مثبتی در مدل وجود ندارد. پس مقادیر مثبت بلوک‌های (۳،۳) و (۳،۴) به بلوک‌های منفی رویی آنها اختصاص داده می‌شود، در نتیجه بلوکی با ارزش مثبت باقی نخواهد ماند (شکل ۲-۲۱)، لذا محدوده‌ی بهینه‌ای در این مدل وجود ندارد؛ درصورتیکه روش کوروبوف الگوریتم برای این مدل محدوده‌ای با ارزش منفی پیدا کرده بود.

	۱	۲	۳	۴	۵	۶
۱	۰	۰	۰	۰	۰	۰
۲	-۱	۰	۰	۰	۰	-۱
۳	-۱	-۱	۰	۰	-۱	-۱

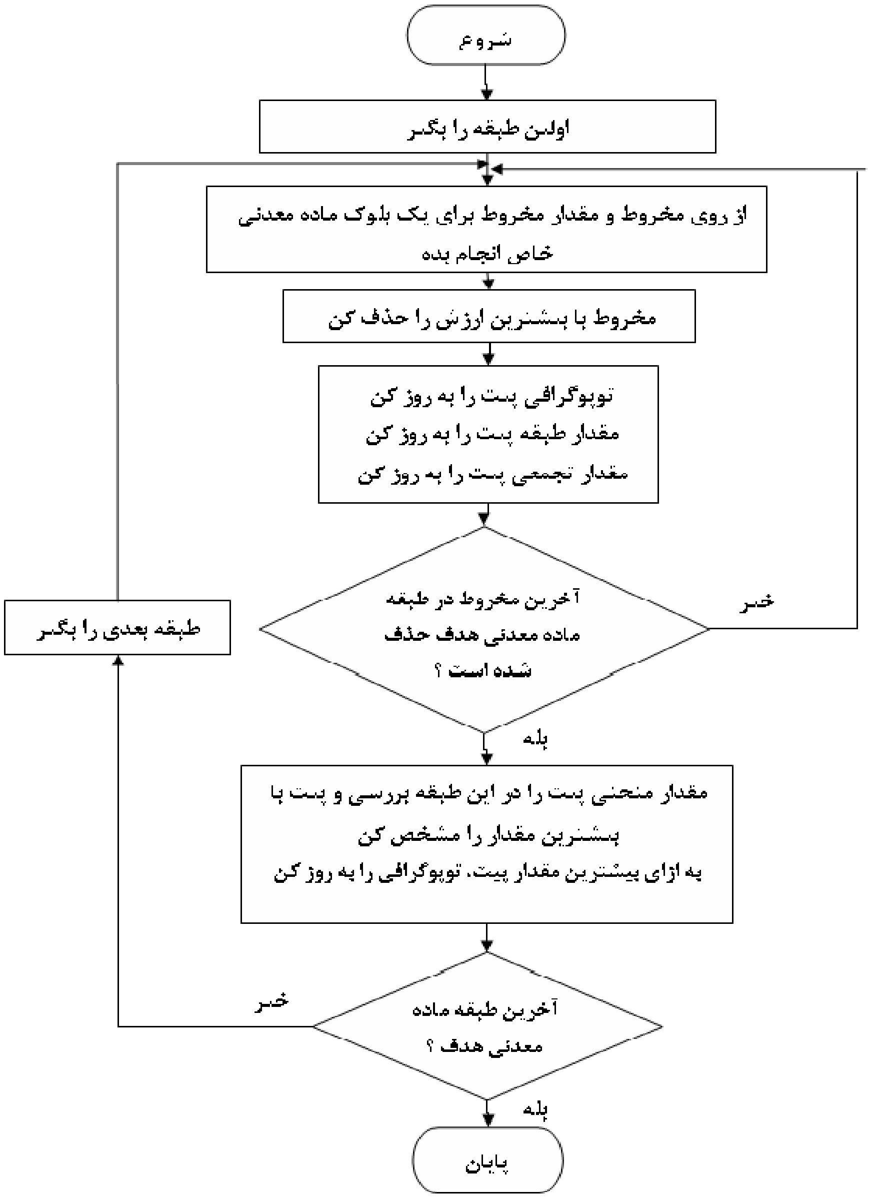
شکل ۲-۲۱: اختصاص بقیه مقادیر بلوک‌های مثبت به بلوک‌های منفی واقع در مخروط آنها.

عیب این روش این است که مخروط‌های مشترک را برای بلوک‌های موجود در یک طبقه درنظر می‌گیرد یعنی درصورتی که بلوک‌های مثبت در یک طبقه باشند. همچنین این روش علیرغم سادگی، به دلیل عدم اثبات بهینه بودن آن چندان مورد قبول قرار نگرفته است (کاکایی، ۱۳۸۴).

۲-۶ روش مخروط شناور II [1]

این روش در سال ۱۹۹۹ توسط رایت[2] و برای برطرف کردن بعضی از ضعف‌های روش مخروط شناور ارائه شد و الگوریتم آن در شکل ۲-۲۲ مشاهده می‌شود (Wright, 2000). روش کار این الگوریتم اینگونه است:

---
[1] - Floating Cone II Metod
[2] - Wright



شکل ۲-۲۲: الگوریتم روش مخروط شناور II.

۱) برای تمام بلوک‌های ماده معدنی مخروط استخراجی با توجه به زاویه شیب پایداری معدن ساخته شده و سپس ارزش آن محاسبه می‌گردد.

۲) مخروط استخراجی با بالاترین ارزش مشخص شده و آن را جزئی از محدوده معدن فرض کرده و ارزش تجمعی پیت محاسبه می‌شود.

۳) پس از اصلاح مدل بلوکی اقتصادی، همین مراحل برای بقیه بلوک‌های ماده معدنی انجام می‌شود.

پس از انجام مراحل فوق، بلوک با بالاترین ارزش تجمعی مشخص شده و محدوده بهینه برای این بلوک تعیین می‌شود. مدل بلوکی شکل ۲-۲۳ را درنظر بگیرید.

	۱	۲	۳	۴	۵	۶	۷	۸
۱	-۲	-۲	-۲	-۲	-۲	-۲	-۲	-۲
۲	-۳	-۳	+۵	-۳	+۹	-۳	-۳	-۳
۳	-۵	-۵	-۵	+۲	-۵	+۱۱	-۵	-۵

شکل ۲-۲۳: یک مدل بلوکی اقتصادی فرضی دو بعدی برای حل با روش مخروط شناور II.

در طبقه اول بلوک ماده معدنی وجود ندارد. بنابراین الگوریتم جستجوی خود را در طبقه دوم انجام می‌دهد. در طبقه دوم دو بلوک با ارزش مثبت وجود دارند که بلوک (۲،۳) دارای ارزش (۱-) و بلوک (۲،۵) دارای ارزش (۳+) است. چون بلوک (۳،۵) دارای بالاترین ارزش مخروط استخراجی است لذا این مخروط جزء محدوده فرض شده (شکل ۲-۲۴) و ارزش تجمعی معدن مطابق ردیف اول جدول ۲-۱ محاسبه می‌شود.

	۱	۲	۳	۴	۵	۶	۷	۸
۱	-۲	-۲	-۲	-۲	-۲	-۲	-۲	-۲
۲	-۳	-۳	+۵	-۳	+۹	-۳	-۳	-۳
۳	-۵	-۵	-۵	+۲	-۵	+۱۱	-۵	-۵

شکل ۲-۲۴: بلوک واقع در ردیف دوم و ستون پنجم جزء محدوده فرض می‌شود.

پس از انجام مراحل فوق و اصلاح مدل بلوکی اقتصادی، در این طبقه فقط بلوک (۳،۳) باقی می‌ماند که دارای ارزش (۱+) می‌باشد، و جزء محدوده معدن درنظر گرفته می‌شود. لذا ارزش تجمعی معدن مطابق ردیف دوم جدول ۲-۱ خواهد بود.

در طبقه دوم بلوک مثبت دیگری وجود نداشته، لذا بلوک با بالاترین ارزش تجمعی را برای این طبقه پیدا کرده که بلوک (۲،۳) دارای بیشترین ارزش تجمعی است. بنابراین دو مخروط استخراجی بلوک‌های مذکور جزء محدوده بهینه می‌باشند (شکل ۲-۲۵).

	۱	۲	۳	۴	۵	۶	۷	۸
۱	-۲						-۲	-۲
۲	-۳	-۳		-۳		-۳	-۳	-۳
۳	-۵	-۵	-۵	+۲	-۵	+۱۱	-۵	-۵

شکل ۲-۲۵: بلوک واقع در ردیف دوم و ستون سوم برداشته می‌شود.

در طبقه سوم دو بلوک ماده معدنی وجود دارد. ارزش مخروط اولین بلوک ماده معدنی، یعنی بلوک (۳،۴) برابر با (۱-) می‌شود، و ارزش بلوک (۳،۶) نیز برابر با (۱+) خواهد بود. به دلیل آنکه بلوک (۳،۶) در این طبقه دارای بیشترین ارزش مخروط استخراجی است، در نتیجه جزء محدوده معدن فرض می‌شود (شکل ۲-۲۶).

	۱	۲	۳	۴	۵	۶	۷	۸
۱	-۲						-۲	-۲
۲	-۳	-۳		-۳		-۳	-۳	-۳
۳	-۵	-۵	-۵	+۲	-۵	+۱۱	-۵	-۵

شکل ۲-۲۶: مخروط حاصل از بلوک (۳،۶) جزء محدوده معدن فرض می‌شود.

شکل محدوده نهایی به صورت شکل ۲-۲۶ (همراه با استخراج بلوک‌های هاشور خورده) می‌باشد. بطور خلاصه کلیه عملیات انجام گرفته را می‌توان در جدول ۲-۱ خلاصه کرد.

جدول ۲-۱: خلاصه عملیات انجام گرفته توسط روش مخروط شناور II.

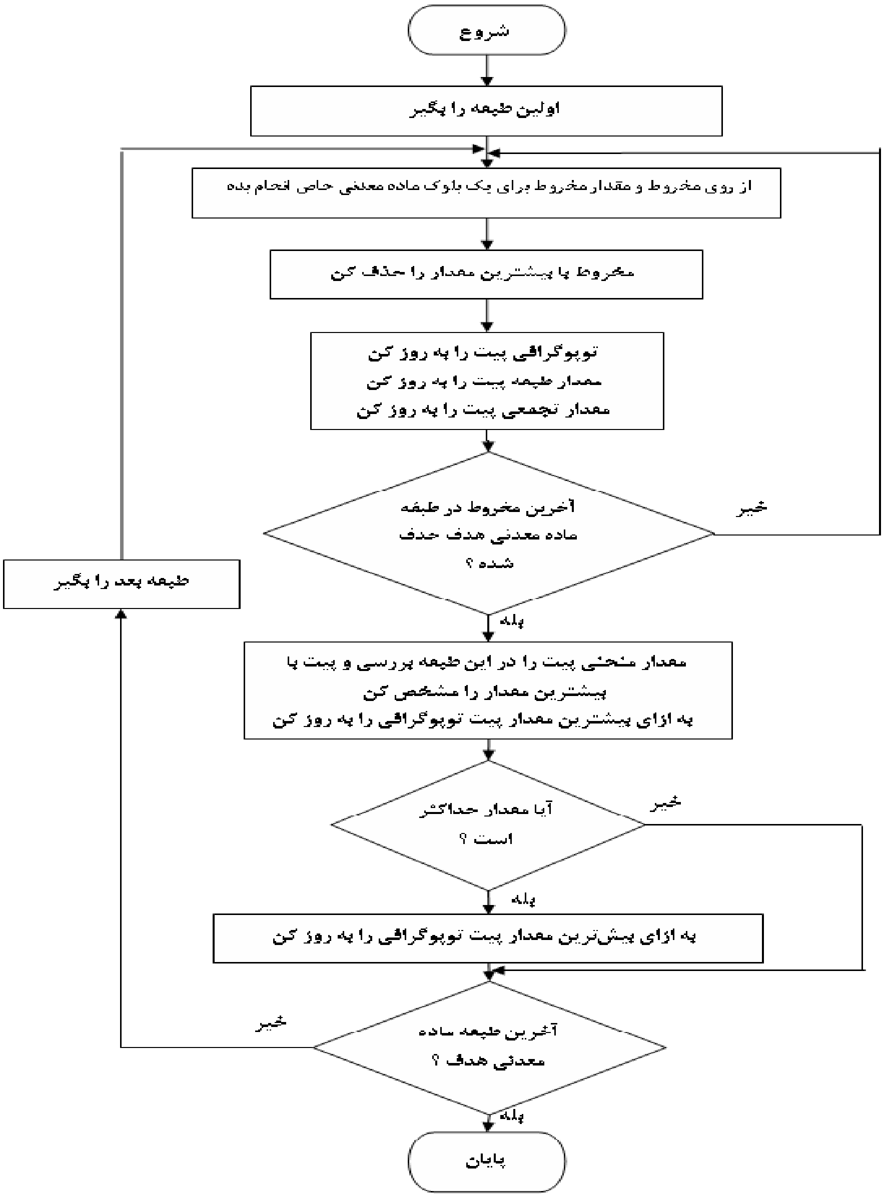
طبقه	بلوک	ارزش مخروط	ارزش تجمعی پیت	
۲	۲،۵	+۳	+۳	
۲	۲،۳	+۱	+۴	جزء محدوده
۳	۳،۶	+۱	+۵	جزء محدوه
۳	۳،۴	-۱	-۴	

و در نتیجه ارزش کلی این مدل با استفاده از الگوریتم مخروط شناور II برابر با (۵+) می‌باشد. با اینکه این روش جواب بهتری نسبت به مخروط شناور می‌دهد ولی جواب بهینه نیست، زیرا که بلوک‌ها را در یک طبقه با هم مقایسه می‌کند و نمی‌تواند بلوک‌هایی را که در طبقه‌های مختلف‌اند درنظر بگیرد. همچنین باید توجه داشت که این روش مقادیر بزرگتر را درنظر می‌گیرد و به اینکه مثبت یا منفی باشند اهمیتی نمی‌دهد، پس ممکن است که محدوده‌ی بهینه با مقدار منفی بدست آورد که اشتباه می‌باشد (کاکایی، ۱۳۸۴).

۲-۷ الگوریتم مخروط شناور II اصلاح شده[1]

این روش همانند روش مخروط شناور II می‌باشد، با این تفاوت که در این روش پس از انجام مراحل روش مخروط شناور II، بلوک با بالاترین ارزش تجمعی در هر طبقه پیدا شده و در صورتی که مقدار آن مثبت باشد، آن بلوک برداشته می‌شود و الگوریتم به طبقه بعدی می‌رود؛ ولی در صورت منفی بودن الگوریتم به طبقه بعد می‌رود و کاری به بلوک ندارد. الگوریتم این روش در شکل ۲-۲۷ آورده شده است (کاکایی، ۱۳۸۴).

[1] - Revised Floating Cone II Method



شکل ۲-۲۷: الگوریتم روش مخروط شناور II اصلاح شده (کاکایی، ۱۳۸۵)

مدل بلوکی ارائه شده در شکل ۲-۲۸ را درنظر بگیرید.

	۱	۲	۳	۴	۵	۶	۷	۸
۱	-۵	-۵	-۵	-۵	-۵	-۵	-۵	-۵
۲	-۱۰	+۳	+۲	-۱۰	-۱۰	-۱۰	-۱۰	-۱۰
۳	-۱۵	-۱۵	-۱۵	-۱۵	+۴۰	+۴۲	-۱۵	-۱۵

شکل ۲-۲۸: مدل بلوکی اقتصادی برای حل با روش مخروط شناور II اصلاح شده.

در طبقه اول بلوک ماده معدنی وجود ندارد، در نتیجه الگوریتم جستجوی خود را در طبقه دوم ادامه خواهد داد. در این طبقه بین بلوک‌های (۲،۲) با ارزش مخروط برابر با (۱۲-) و بلوک (۲،۳) با ارزش مخروط برابر با (۱۳-) بررسی انجام می‌شود (شکل ۲-۲۹). ابتدا بلوک (۲،۲) جزو محدوده بهینه فرض خواهد شد (ردیف اول جدول ۲-۲). پس از اصلاح مدل بلوکی اقتصادی، بلوک (۲،۳) دارای مخروطی با ارزش (۳-) خواهد بود که مطابق ردیف دوم جدول ۲-۲ می‌باشد. با توجه به اینکه بالاترین ارزش تجمعی پیت در طبقه دوم منفی می‌باشد، لذا هیچکدام از بلوک‌های مثبت در این طبقه به محدوده بهینه اضافه نمی‌شود.

	۱	۲	۳	۴	۵	۶	۷	۸
۱	-۵	-۵	-۵	-۵	-۵	-۵	-۵	-۵
۲	-۱۰	+۳	+۲	-۱۰	-۱۰	-۱۰	-۱۰	-۱۰
۳	-۱۵	-۱۵	-۱۵	-۱۵	+۴۰	+۴۲	-۱۵	-۱۵

شکل ۲-۲۹: مخروط حاصل از بلوک (۲،۲)، جزء محدوده فرض می‌شود.

سپس الگوریتم بلوک‌های موجود در طبقه سوم را بررسی می‌کند. در طبقه سوم دو بلوک ماده معدنی وجود دارد، بلوک (۳،۵) با ارزش مخروط برابر با (۱۵-) و بلوک (۳،۶) با ارزش مخروط برابر با (۱۳-)، بلوک (۳،۶) دارای بالاترین ارزش تجمعی (برابر ۱۲+) در طبقه سوم است و چون مقدار آن مثبت می‌باشد، لذا مخروط‌های استخراجی تا این بلوک به محدوده بهینه اضافه می‌گردد که مطابق ردیف‌های ۳ و ۴ جدول ۲-۲، شامل مخروط استخراجی بلوک‌های (۳،۵) و (۳،۶) خواهد بود. (شکل ۲-۳۰ محدوده بهینه واقعی را نشان می‌دهد.

	۱	۲	۳	۴	۵	۶	۷	۸
۱								
۲	۱۰-							۱۰-
۳	۱۵-	۱۵-	۱۵-	۱۵-			۱۵-	۱۵-

شکل ۲-۳۰: بلوک‌های (۳،۵) و (۳،۶) برداشته می‌شوند.

کل عملیات انجام گرفته را می‌توان به شکل جدول ۲-۲ خلاصه کرد.

جدول ۲-۲: خلاصه مراحل انجام گرفته توسط روش مخروط شناور II اصلاح شده.

طبقه	بلوک	ارزش مخروط	ارزش تجمعی پیت	
۲	۲،۲	۱۲-	۱۲-	
۲	۲،۳	۳-	۱۵-	
۳	۳،۵	۱۵-	۱۵-	
۳	۳،۶	۲۷+	۱۲+	جزء محدوده

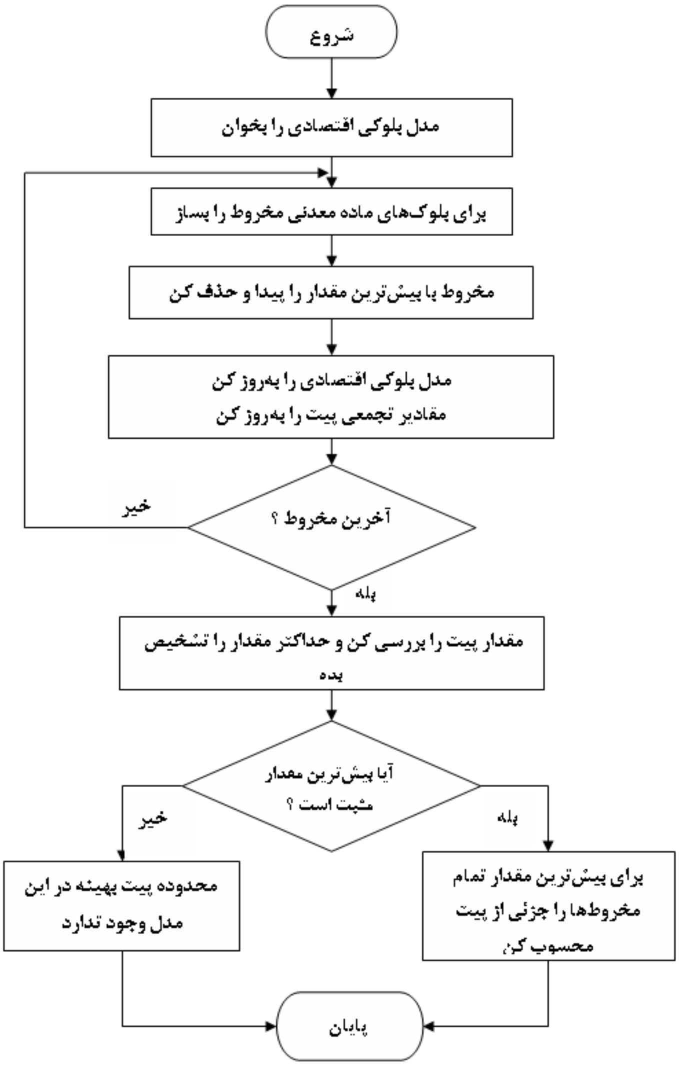
در نتیجه ارزش نهایی پیت برابر است با (۱۷+).

۲-۸ اصلاحیه دوم روش مخروط شناور II [1]

الگوریتم این روش در شکل ۲-۳۱ مشاهده می‌شود. در این روش ابتدا تمام مدل بلوکی بصورت یکجا توسط نرم‌افزار خوانده شده و برای تمام بلوک‌های ماده معدنی مخروط تشکیل می‌شود. از بین مخروط‌های تشکیل شده، هر کدام که بیشترین ارزش را دارد جزء محدوده معدن فرض کرده و دوباره کل مدل بلوکی را بدون مخروط حاصل از بلوک استخراج شده درنظر گرفته و مخروط حاصل از بلوک ماده معدنی با بیشترین ارزش جزء محدوده فرض می‌شود. در هر مرحله، ارزش‌ها را در یک جدول یادداشت کرده و در انتها با بررسی جدول، هر جا که ارزش تجمعی بیشترین مقدار را داشت، تا آنجا

[1] - Second Revise in Floating Cone Method II

عملیات استخراج انجام خواهد گرفت (کاکایی، ۱۳۸۵). با ذکر یک مثال این روش بیشتر شرح داده می‌شود.



شکل ۲-۳۱: الگوریتم روش مخروط شناور II، اصلاحیه دوم (ارائه توسط آقای کاکایی).

مدل بلوکی شکل ۲-۳۲ را درنظر بگیرید.

	۱	۲	۳	۴	۵	۶	۷	۸
۱	-۵	-۵	-۵	-۵	-۵	-۵	-۵	-۵
۲	-۱۰	+۸	-۱۰	-۱۰	-۱۰	-۱۰	+۹	-۱۰
۳	-۱۵	-۱۵	-۱۵	+۵۴	-۱۵	-۱۵	-۱۵	-۱۵

شکل ۲-۳۲: حل این مدل توسط روش مخروط شناور II، اصلاحیه دوم.

در این مدل سه بلوک ماده معدنی وجود دارد. در ابتدا ارزش مخروط استخراجی برای همه بلوک‌های ماده معدنی محاسبه شده و مهم نیست که در کجا قرار دارند. به این ترتیب، ارزش مخروط حاصل از بلوک (۲،۲) برابر با (۷-)، ارزش مخروط حاصل از بلوک (۲،۷) برابر با (۶-) و ارزش مخروط حاصل از بلوک (۳،۴) برابر با (۱-) می‌باشد. به دلیل بیشتر بودن ارزش مخروط بلوک (۳،۴)، این بلوک و مخروط آن جزء محدوده معدن فرض می‌شوند. سپس ارزش تجمعی محاسبه شده و پس از اصلاح مدل بلوکی، دو بلوک ماده معدنی باقی مانده‌اند که ارزش مخروط حاصل از بلوک (۲،۲) برابر با (۳+) و ارزش مخروط حاصل از بلوک (۲،۷) برابر با (۱-) خواهد بود. در نتیجه بلوک (۲،۲) به همراه مخروط آن به محدوده بهینه اضافه می‌گردد. پس از برداشت این بلوک، مخروط حاصل از بلوک (۲،۷) تغییری نکرده و با همان ارزش (۱-) باقی می‌ماند. کل محاسبات انجام گرفته را می‌توان بصورت جدول ۲-۳ خلاصه کرد. محدوده بهینه این مدل نیز به صورت شکل ۲-۳۳ خواهد بود.

	۱	۲	۳	۴	۵	۶	۷	۸
۱							-۵	-۵
۲	-۱۰					-۱۰	+۹	-۱۰
۳	-۱۵	-۱۵	-۱۵		-۱۵	-۱۵	-۱۵	-۱۵

شکل ۲-۳۳: محدوده بهینه متعلق به مدل بلوکی شکل ۲-۳۲.

جدول ۲-۳: مراحل انجام گرفته برای تعیین محدوده نهایی معدن با روش مخروط شناور II، اصلاحیه دوم

بلوک	ارزش مخروط	ارزش تجمعی	
۳،۴	-۱	-۱	
۲،۲	+۳	+۲	← برداشته می‌شود
۷،۲	-۱	+۱	

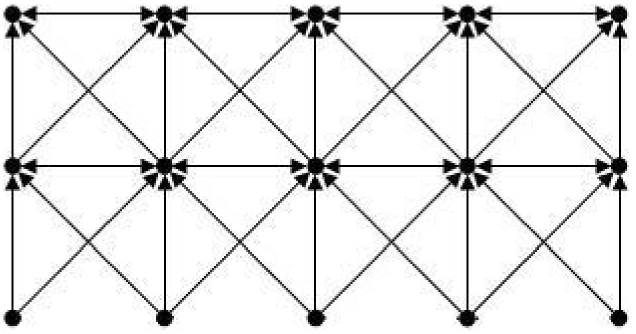
لازم به توضیح است که این روش در مقایسه با دو روش مخروط شناور و اصلاحیه اول مخروط شناور، نتایج بهتری را بدست می‌دهد.

۲-۹ الگوریتم لرچ و گرسمان[1]

این الگوریتم که در سال ۱۹۶۵ توسط لرچ و گروسمن معرفی گردید (Lerchs & Grossman, 1965)، تنها روشی است که ثابت شده در تمامی حالات و هر ترکیب از مدل بلوکی قادر به بدست آوردن جواب بهینه واقعی می‌باشد. در این روش مدل بلوکی اقتصادی به یک گراف تبدیل می‌شود که در آن هر بلوک با گره و ارتباط بین آنها با بردار بیان می‌گردد. این ارتباط (بردارها) به صورتی است که بیانگر یک پیت قابل اجرا باشد (کاکایی، ۱۳۸۴). مدل بلوکی شکل ۲-۳۴ را درنظر بگیرید.

-۲	-۲	-۲	-۲	-۲
-۲	-۲	+۴	+۵	-۲
-۳	-۳	+۷	-۳	-۳

شکل ۲-۳۴: یک مدل بلوکی دو بعدی



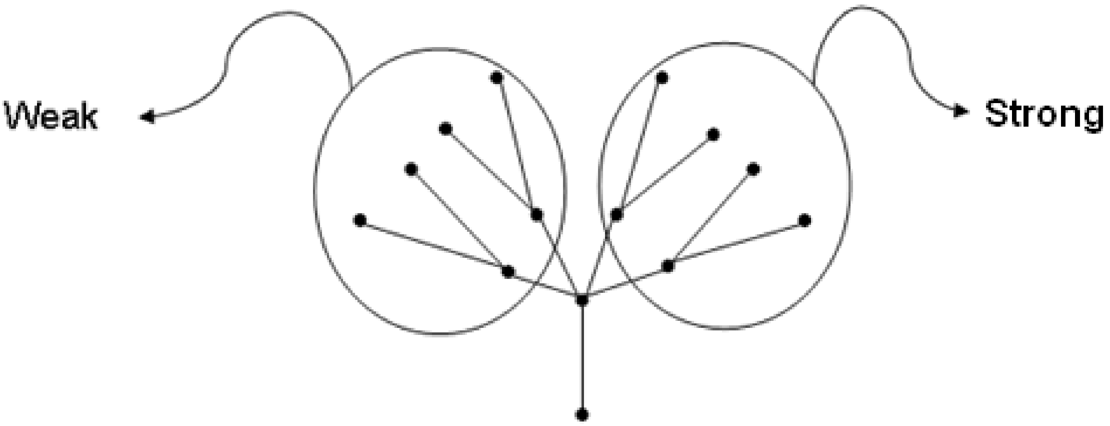
شکل ۲-۳۵: G (N , A)، مدل بلوکی به گرافی جهت دار تبدیل می‌شود.

در این روش مدل بلوکی را به گراف تبدیل کرده که خطوط جهت‌دار همان محدودیت‌های استخراج را نشان می‌دهند (همانند شکل ۲-۳۵). به هر مجموعه‌ای از گره‌ها که یک پیت را تشکیل

[1] - Lerch and Grossman algorithm (Based on Graph Theorem)

می‌دهند، کلوژر[1] گفته می‌شود. هر کلوژور دارای ارزشی است که برابر مجموع ارزش گره‌های داخلی آن است. هدف یافتن کلوژری است که بیش‌ترین مقدار را داشته باشد. پس از این مرحله گراف به صورت یک درخت درمی‌آید.

سپس درخت را به دو شاخه ضعیف و قوی تقسیم کرده که شاخه‌های قوی همان محدوده نهایی بلوک می‌باشند (شکل ۲-۳۶).



شکل ۲-۳۶: درختی که دارای دو شاخه با کلوژرهای ضعیف و قوی می‌باشد.

معایب الگوریتم لرچ و گروسمن براساس تئوری گراف عبارتند از:

- نیاز به وقت کامپیوتری زیادی دارد، در حال حاضر با توجه به پیشرفت رایانه‌ها، این عیب قابل برطرف شدن می‌باشد؛
- پیچیده و مشکل است، و تهیه برنامه رایانه‌ای با این روش از عهده یک مهندس معدن با دانش متوسط خارج است؛
- استفاده از شیب‌های متغیر در این روش بسیار مشکل است؛

و مزایای آن را نیز می‌توان اینگونه برشمرد:

- تنها روشی است که قادر می‌باشد در تمام مدل‌ها، محدوده بهینه واقعی را بدست آورد؛

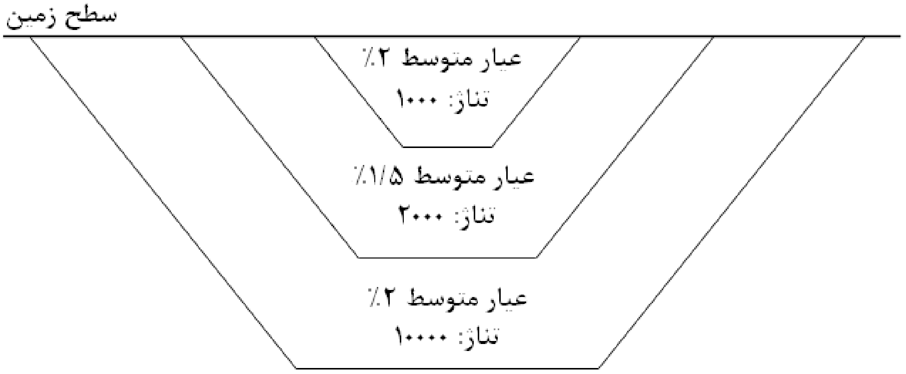
[1] - Closure

- لازم به توضیح است که بسیاری از نرم افزارهای تجاری از این روش برای محاسبه محدوده بهینه استفاده می‌کنند (کاکایی، ۱۳۸۴).

۲-۱۰ روش پارامتری کردن[1]

این روش که توسط پروفسور ماترون[2] در سال ۱۹۷۵ ارائه شده است؛ تنها مدلی می‌باشد که براساس مدل بلوکی زمین‌شناسی کار می‌کند، درصورتیکه بقیه مدل‌ها روی مدل اقتصادی کار می‌کنند. پیت حاصل از این روش تابع هزینه و درآمد معدنکاری و قیمت‌های فروش می‌باشد که ممکن است در طول عمر معدن تغییر کرده و بر این اساس محدوده معدن نیز تغییر نماید. در این روش یک سری پیت‌های تودرتو و براساس دو پارامتر عیار و تناژ و با هدف به حداکثر رساندن مقدار فلز برای مقادیر مختلف ماده معدنی طراحی می‌شود. (کاکایی، ۱۳۸۴)، (رابطه ۲-۳ و شکل ۲-۳۷).

pit = f (تناژ ، عیار) (رابطه ۲-۳)



شکل ۲-۳۷: پیت‌های تودرتو در روش پارامتری کردن.

---
1 - Parametrizing Method
2 - Matheron

در مرحله بعد پارامترهای اقتصادی روی پیت اعمال می‌شوند تا بهترین پیت بدست آید. در عمل این روش به ریاضیات قوی نیاز دارد و جواب بهینه واقعی را نیز به دست نمی‌دهد. روشی مشابه با این روش نیز اخیراً توسط چیکویسنه و همکاران معرفی شده است .(Chicoisne et al, 2009)

۲-۱۱ الگوریتم ژنتیک[1]

مبنای این روش، اصل داروین می‌باشد. در این روش به جای ارزیابی همه حالات ممکن، تعدادی از آن‌ها مورد ارزیابی قرار می‌گیرند.

به هر جواب بهینه ممکن یک کورموزم گفته می‌شود و هر کدام از بیت‌ها (که عنصری از کورموزم است) ژن نامیده می‌شود. به یک سری از کورموزم‌ها، جمعیت گفته می‌شود.

مراحل این الگوریتم را می‌توان به شرح ذیل خلاصه نمود:

گام اول: تولید جمعیت اولیه به صورت تصادفی. این جمعیت اولیه به صورت تصادفی از بین تعداد زیادی جواب انتخاب می‌شوند. ارزش محدوده (تابع برازندگی) برای این جمعیت محاسبه می‌شود.

گام دوم: انتخاب تعدادی از جمعیت اولیه برای تولید مثل. در این مرحله تعدادی از کروموزم‌ها براساس روش‌های خاصی برای تولید مثل انتخاب می‌شوند.

گام سوم: تکامل[2]. یکی از روش‌های تکامل، تبادل ژنی[3] می‌باشد. برای این کار از بین دو کرموزوم یک عدد تصادفی (بین صفر تا تعداد ژن‌ها) را انتخاب کرده و به جای این ژن‌ها دو ژن جدید دیگر قرار می‌گیرد. حاصل این کار به دست آمدن دو کورموزم جدید است.

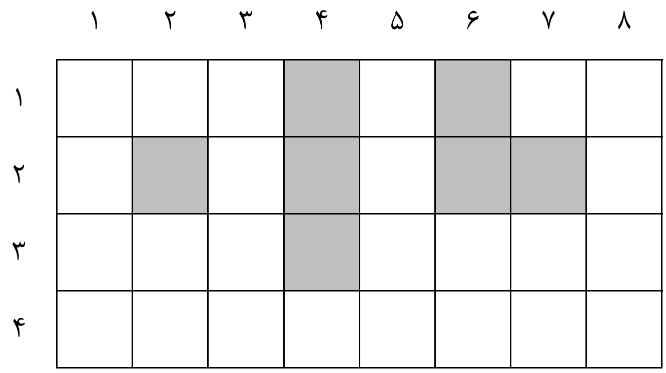
روش دیگر در تکامل، جهش ژنی[4] می‌باشد. برای این عمل، تعدادی از ژن‌های جمعیت را انتخاب کرده و سپس به صورت تصادفی آن ژن اگر ۱ بود با ۰ و اگر ۰ بود با ۱ تعویض می‌شوند.

---

[1] - Genetic Algorithm
[2] - Reproduction
[3] - Cross Over
[4] - Mutation

گام چهارم: تکرار؛ روی نسل جدید تمام کارهای بالا انجام می‌گیرند. این کار تا آنجا انجام می‌شود که هیچ بهبودی از نتایج حاصل نشود (البته بعد از هر تکرار نسل جدید از نسل قدیم بهتر می‌شود).

یک نمونه مدل‌سازی در این قسمت توضیح داده می‌شود. مدل بلوکی شکل ۲-۳۸ را درنظر بگیرید که در این مدل بلوک‌های ماده معدنی با رنگ تیره مشخص شده‌اند.



شکل ۲-۳۸: در این مدل؛ بلوک‌های دارای ماده معدنی با هاشور مشخص شده‌اند.

حال کورموزم را به صورت تعداد بلوک‌های ماده معدنی در هر ستون از مدل بلوکی تعریف کرده که در این مثال می‌تواند اعداد ۰، ۱، ۲ یا ۳ (به این معنی که از این ستون چند بلوک ماده معدنی در محدوده پیت نهایی قرار دارند) باشد؛ سپس جدول زیر تشکیل داده می‌شود (جدول احتمالات ممکن).

	ستون ۷	ستون ۶	ستون ۴	ستون ۲
هیچ‌کدام از بلوک‌ها در محدوده نهایی قرار ندارند	۰	۰	۰	۰
یکی از بلوک‌ها در محدوده نهایی قرار دارد	۱	۱	۱	۱
دو تا از بلوک‌ها در محدوده نهایی قرار دارند		۲	۲	
هر سه بلوک در محدوده نهایی قرار دارند			۳	

مزایای روش الگوریتم ژنتیک:

- به جای ارزیابی همه حالات ممکن، تعدادی از آنها ارزیابی می‌شوند؛
- بهتر شدن جواب در هر مرحله از جستجو؛
- توان بهینه سازی پارامترهای گسسته و پیوسته؛
- توان موازی سازی الگوریتم برای حل مسئله با چندین دستگاه رایانه (Haupt & Haupt, 1998).

معایب روش الگوریتم ژنتیک را نیز می‌توان اینگونه نام برد:

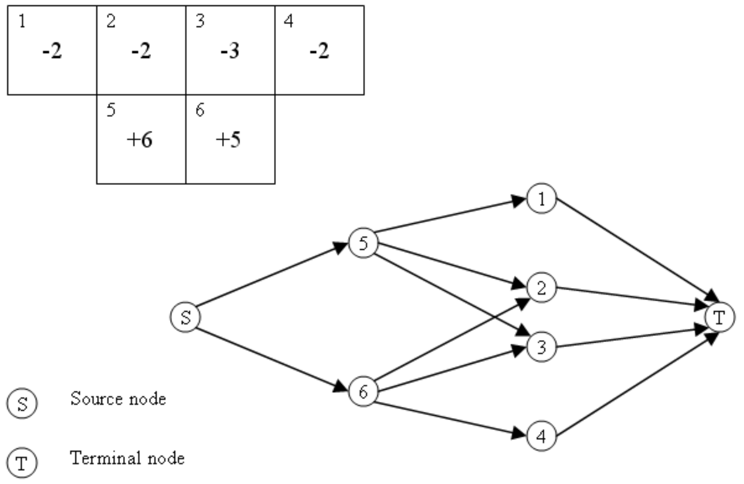
- عدم تخمین بهینه بودن جواب؛
- نیاز به مدلسازی پیچیده؛
- در بعضی حالات ممکن است مدلسازی به ایجاد کورموزم‌های بزرگ منجر شود.

۲-۱۲ روش شبکه و حداکثر جریان[1]

اولین بار جانسون[2] در سال ۱۹۶۸ روش شبکه و حداکثر جریان را برای طراحی محدوده معادن روباز بکار برد. در این روش شبکه‌ای با ترمینال‌های ورودی و خروجی مطابق شکل ۲-۳۹ ساخته می‌شود که در آن هر گره نشان‌دهنده یک بلوک بوده و بلوک‌های ماده معدنی به ترمینال ورودی و بلوک‌های باطله به ترمینال خروجی وصل شده است. بلوک‌های ماده معدنی نیز با توجه به زاویه شیب معدن طوری به بلوک‌های باطله رویی وصل می‌گردند که هر کدام نمایانگر یک پیت قابل اجرا باشند. شکل ۲-۳۹ یک شبکه ساده را نشان می‌دهد که برای یک مدل بلوکی دو بُعدی بکار رفته که در آن زاویه شیب معدن ۴۵ درجه و ابعاد بلوک‌ها یکسان فرض شده است. ظرفیت شاخه‌ها از ترمینال ورودی به بلوک‌های ماده معدنی برابر ارزش آن‌ها و ظرفیت شاخه‌ها از بلوک‌های باطله به ترمینال

[1] - Network and maximal flow techniques
[2] - Johnson

خروجی برابر قدر مطلق ارزش این بلوک‌ها در نظر گرفته می‌شود. همچنین ظرفیت شاخه‌ها از بلوک‌های ماده معدنی به بلوک‌های باطله رویی برابر بی‌نهایت منظور می‌شود. پس از ساختن شبکه، آن را برای بدست آوردن حداکثر جریان از ترمینال خروجی حل نموده که محدوده بهینه نهایی در معادن روباز را بدست می‌دهد. علیرغم بعضی از موفقیت‌های این روش، به دلایل داشتن بعضی معایب ظاهری همانند روش لرچ و گروسمن، چندان به عنوان یک روش کلی مورد قبول قرار نگرفته است (کاکایی، ۱۳۸۴).



شکل ۲-۳۹: یک شبکه ساده برای مدل بلوکی فرضی ارائه شده در قسمت بالای شکل.

فصل سوم

الگوریتم کولونی مورچگان

فصل ۳

۳-۱ مقدمه

هدف از این فصل معرفی الگوریتم کولونی مورچگان، به منظور کاربرد این الگوریتم در حل مسائل بهینه سازی می‌باشد. در این فصل سعی شده که مفاهیم الگوریتم کولونی مورچگان هر چه ساده‌تر بیان شود.

الگوریتم کولونی مورچگان جزئی از الگوریتم‌های ابتکاری و یا تکاملی می‌باشد، الگوریتم‌های تکاملی آنهایی هستند که از مشاهده رفتار، طرز عمل و کارکرد جانداران یا سیستم‌های موجود در طبیعت الهام گرفته شده‌اند، و برای مسائل روزمره قابلیت کاربرد دارند.

امروزه الگوریتم‌های تکاملی و ابتکاری مورد توجه محققان قرار گرفته‌اند و بطور وسیعی در صنایع مختلف مورد استفاده قرار می‌گیرند. یکی از این الگوریتم‌های ابتکاری، الگوریتم کولونی مورچگان می‌باشد که تاکنون از این الگوریتم برای کاربردهای بسیاری استفاده شده است نظیر حل مسائل مدیریت پروژه (Abdallah et al, 2008)، مسائل پیچیده غیر خطی (Schluter et al, 2009)، مسائل طبقه بندی در مهندسی شیمی (Shelokar et al, 2004) و بسیاری مسائل دیگر. همچنین ترکیب این الگوریتم با دیگر روش‌های فرا ابتکاری نیز نتایج رضایت بخشی را بدست داده است؛ نظیر ترکیب الگوریتم کولونی مورچگان و الگوریتم بهینه سازی توده ذرات برای رده‌بندی‌های مالی

(Marinakis et al, 2009)، ترکیب سیستم ایمنی مصنوعی و الگوریتم کولونی مورچگان برای ارزیابی بازار یک محصول خاص (Yu Chiu et al, 2009)، کاربرد الگوریتم کولونی مورچگان و روش جستجوی تابو برای بعضی مسائل خاص (Ling Huang & Jong Liao, 2008) و بسیاری از موارد دیگر.

۳-۱-۱ الگوریتم ابتکاری چیست؟

یک روش ابتکاری بخشی از یک الگوریتم بهینه سازی است که از اطلاعاتی که تاکنون توسط الگوریتم جمع آوری شده استفاده می کند تا به تصمیم گیری در مورد اینکه کدام راه حل کاندیدا باید در مرحله‌ی بعد آزمایش شود یا اینکه چگونه جواب مرحله بعدی مورد استفاده واقع شود، کمک کند. روش‌های ابتکاری معمولاً وابسته به نوع مسئله هستند و مدلسازی آنها ممکن است با یکدیگر تفاوت داشته باشد.

۳-۱-۲ اصول الگوریتم‌های ابتکاری

برای الگوریتم‌های ابتکاری قواعد و قوانینی وجود دارد که باید درنظر گرفته شوند. از آنجا که الگوریتم‌های ابتکاری برای حل مسائلی با پایه ریاضیات بکار می‌روند، هر کدام برای حل مسئله‌ای که به آن سپرده می‌شود نیاز دارد تا قوانین دقیقی را اجرا نماید. در اینجا اصول اولیه تمامی الگوریتم‌های ابتکاری ذکر می‌شود. قابل توجه است که اینها تمامی قوانین بکار رونده در الگوریتم‌ها نمی‌باشند، بلکه در تمامی اینگونه الگوریتم‌ها مشترک هستند.

- الگوریتم‌های ابتکاری دارای یک شکل کلی بوده و این وظیفه کاربر است که مسئله را به گونه‌ای فرمول بندی کند که الگوریتم ابتکاری قادر باشد بر روی مسئله پیاده سازی شود.
- هر الگوریتم ابتکاری با تعدادی تکرار سعی در بدست آوردن بهترین جواب ممکن دارد.
- معیار یا معیارهای توقفی باید در الگوریتم پیش بینی شوند.
- بعضی از الگوریتم‌های ابتکاری فقط بر روی مسائل پیوسته و بعضی دیگر فقط بر روی مسائل گسسته کار می‌کنند.

۳-۲ دلایل نیاز به الگوریتم‌های ابتکاری

۳-۲-۱ مسائل بهینه سازی

مسائل بهینه سازی دارای بیانی غالباً ساده اما راه‌حل‌هایی بسیار دشوار می‌باشند از اینرو برای حل کردن چنین مسائلی روش‌های معمول جوابگو نیستند. بسیاری از مسائلی که در عمل وجود دارند، از نوع چند جمله‌ای غیرقطعی سخت[1] به شمار می‌روند یعنی این گونه مسائل، در یک زمان محدود قابل حل نیستند. در عمل برای حل مثال‌هایی با ابعاد بزرگ استفاده از روش‌های تخمینی که بتوانند جواب‌های نزدیک به بهینه را در زمان نسبتاً کوتاهی بدهند ضروری می‌باشد.

خود مسئله‌ی بهینه سازی ترکیبی، مسئله کمینه و یا بیشینه سازی است. واژه مسئله نیز بر پرسشی کلی دلالت دارد که باید پاسخ داده شود و معمولاً چندین پارامتر و یا متغیر با مقادیر نامشخص دارد.

اگر در عمل جواب‌ها قابل دسترس نباشند، تنها راه حل این است که از بهینگی به نفع کارایی صرفنظر شود؛ به عبارت دیگر به جای یافتن جواب بهینه می‌توان جواب‌های بسیار خوب (اما نه بهینه) را جستجو کرد. الگوریتم‌های تخمین‌گر غالباً روش‌های ابتکاری نامیده می‌شوند. ویژگی این روش‌ها، یافتن جواب‌های مناسب، یعنی جواب‌های نزدیک به بهینه با هزینه محاسباتی اندک می‌باشد، بدون آنکه قادر باشند، بهینگی جواب‌های یافت شده را تضمین کنند. این دقیقاً عملی است که الگوریتم‌های ابتکاری قادر به انجام آن می‌باشند (Dorigo & Stutzle, 2004).

۳-۲-۲ پیچیدگی محاسباتی

یک روش ساده برای حل و بدست آوردن جواب‌های مسائل بهینه سازی ترکیبی، روش تحقیق فراگیر[2] است. منظور از این روش آن است که تمام جواب‌های ممکن شمارش شده و بهترین جواب

---

[1] - non-deterministic polynominal hard (NP-hard)
[2] - Exhausive search

انتخاب شود. در بسیاری از مسائل، این رویکرد به سرعت غیرممکن می‌شود. زیرا تعداد جواب‌های ممکن مسئله با افزایش ابعاد آن به صورت نمایی افزایش می‌یابد.

۳-۲-۳ تابع پیچیدگی زمانی

تابع پیچیدگی زمانی یک الگوریتم برای یک مسئله مفروض، برای هر ورودی با اندازه n، حداکثر زمانی است که الگوریتم برای یافتن جواب آن مسئله نیاز دارد.

گرچه نشان داده شده است که برخی از مسائل مهم بهینه سازی ترکیبی در زمان‌های چند جمله‌ای قابل حل هستند، اما برای اکثر اینگونه مسائل هنوز حد زمانی چند جمله‌ای یافت نشده است. در این مسائل، زمان لازم برای اجرای بهترین الگوریتم شناخته شده، با افزایش ابعاد مسئله بطور نمایی افزایش یافته و به تبع آن، زمان مورد نیاز برای یافتن یک جواب بهینه نیز به همین ترتیب افزایش می‌یابد. یکی از مثال‌های بارز در این زمینه مسئله فروشنده دوره‌گرد است که در بخش‌های بعدی توضیح داده می‌شود.

۳-۳ مورچگان و محققان

الگوریتم‌های بهینه‌سازی توسط کولونی مورچگان، بخشی از الگوریتم‌هایی هستند که از مشاهده‌ی رفتار دسته‌ها (گروه‌ها) الهام گرفته شده‌اند. الگوریتم‌های هوشمند گروهی، از موارد تکی‌ای ساخته می‌شوند که در سازمان‌دهی خودشان شرکت دارند بدون اینکه هیچگونه کنترل مرکزی بر روی اعضای گروه صورت گیرد. بهینه‌سازی توسط کولونی مورچگان از مشاهده‌ی رفتار مورچه‌های واقعی الهام گرفته است. در این فصل ابتدا تعدادی از مشاهداتی که در آزمایشات با مورچه‌های واقعی بدست آمده‌اند ارائه شده و سپس نشان داده می‌شود که چگونه این مشاهدات منبع الهام طرح ACO[1] فراابتکاری قرار گرفته‌اند (Dorigo & Socha, 2006).

[1] - Ant Colony Optimization

یکی از اولین محققانی که رفتار حشرات را بررسی کرد، حشره شناس فرانسوی "پیر پاول گراس[۱]" بود. در دهه‌های ۱۹۴۰ و ۱۹۵۰ او مشغول بررسی رفتار موریانه‌ها به ویژه گونه‌های "بِلیکوسیتِرمِس "ناتالِنسیس[۲]" و "کوبیترم‌ها[۳]" بود. او دریافت که این حشرات قادرند به آنچه که "تحریک عمده[۴]" نامیده می‌شود -مواردی که سبب فعال سازی یک واکنش ژنتیکی رمزگذاری شده می‌شوند- واکنش نشان دهند. او مشاهده کرد که اثرات این واکنش‌ها می‌تواند به عنوان یک تحریک، هم برای حشره‌ای که آن را ایجاد می‌کند و هم برای دیگر حشرات در کلونی عمل کند. گراس از کلمه‌ی استیگمرجی[۵] (نشانه‌دار کردن) برای بیان این نوع خاص از ارتباط غیر مستقیم استفاده کرد که در این ارتباط "کارگران توسط کارائی و راندمانی که بدست می‌آورند تحریک می‌شوند" (Dorigo & Socha, 2006).

دو ویژگی مهم استیگمرجی که آن را از ارتباط‌های دیگر متمایز می‌سازد به این شرح هستند:

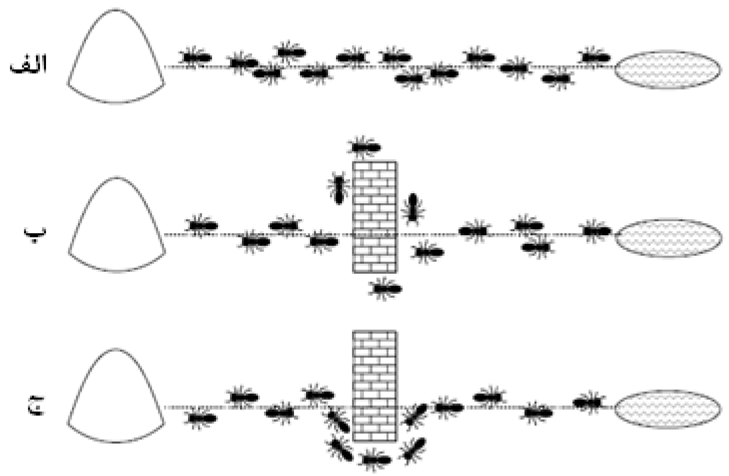
- طبیعت فیزیکی و غیرملموس اطلاعات منتشر شده توسط حشراتی که با هم ارتباط برقرار می‌کنند، که فقط به تغییر حالت فیزیکی محیطی که حشرات از آن بازدید می‌کنند ارتباط دارد (به این معنی که ارتباط بین دو یا چند مورچه و یا حشره از راه تغییر دادن محیط حاصل می‌شود)؛ و
- طبیعت محلی اطلاعات منتشر شده؛ که فقط می‌تواند در دسترس حشراتی باشد که از محل انتشار اطلاعات (و یا همسایگی بسیار نزدیک به آن) بازدید می‌کنند، بدین ترتیب این اطلاعات در دسترس کلیه حشرات کولونی قرار ندارد (Dorigo & Socha, 2006).

نمونه‌هایی از استیگمرجی را می‌توان در کولونی مورچگان مشاهده کرد. در بسیاری از گونه‌های مورچه، مورچگان در حال رفتن به سمت منبع غذا و یا در حال بازگشتن از منبع غذا به سمت لانه،

---

1 - Pieere Paul Grasse
2 - Bellicositermes natalensis
3 - Cubiterms
4 - significant stimuli
5 - Stigmergy

ماده‌ای را که *فرومون*[1] نامیده می‌شود برجای می‌گذارند. مورچه‌های دیگر قادر هستند که این فرومـون را بوکشیده، و وجود این ماده در انتخاب مسیر بعدی آنان تاثیر مـی‌گـذارد، چراکـه مورچگـان تمایـل دارند مسیر با غلظت فرومون بیشتر را دنبال کنند. فرومون گذاشته شـده بـر روی زمـین یـک مسـیر *فرومونی* را تشکیل می‌دهد که به مورچه‌ها اجازه می‌دهد تا منبـع غـذایی خـوبی را کـه پـیش از ایـن توسط مورچه‌های دیگر شناسایی شده است بیابند که در شکل ۳-۱ نیز قابل مشـاهده اسـت، در ایـن شکل، در قسمت بالا، مورچه‌ها مسیر بین لانه و غذا را بطور عادی مـی‌پیماینـد؛ در شـکل وسـط ایـن مسیر توسط مانعی مسدود شده است، بعضی از مورچه‌ها بطور تصادفی مسیر طولانی‌تر و بعضی دیگـر مسیر کوتاه‌تر را انتخاب می‌کنند؛ همانطور که در قسمت پایین شکل ملاحظه می‌شـود پـس از مـدتی تمام مورچه‌ها از مسیر کوتاه‌تر عبور می‌کنند. (Dorigo & Socha, 2006).



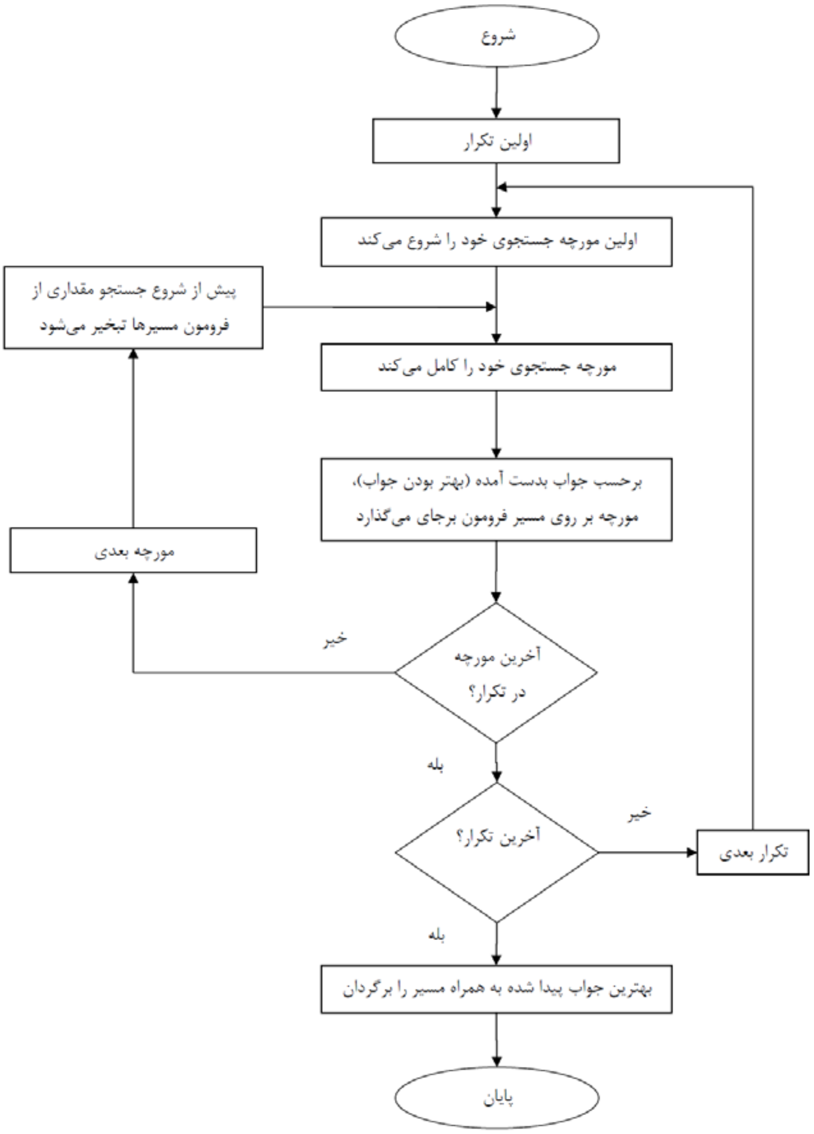
شکل ۳-۱: یک کولونی از مورچگان کوتاه‌ترین مسیررا می‌یابند.

۳-۵ استفاده از رفتار مورچه‌های واقعی برای کمّی‌سازی این رفتار و حل مسئله

با استفاده از مدلسازی رفتار مورچه‌های طبیعی می‌توان برای حل مسائل از مورچه‌های مصـنوعی استفاده کرد. برای این منظور شناخت رفتار مورچه‌ها ضروری می‌باشد. در فلوچارت شـکل ۳-۲ رفتـار

---
[1] - Fromon

مورچه‌ها برای رسیدن به یک جواب نشان داده شده است. بر اساس این فلوچارت، هر مورچـه پـس از کامل کردن مسیر خود و ساختن یک جواب بر روی مسیری که از آن عبور کرده است فرومون برجای می‌گذارد. همانطور که مشاهده می‌شود، رفتار مورچه‌ها تصادفی بـوده و ممکـن اسـت هـر مسـیری را انتخاب کنند.



شکل ۳-۲: فلوچارت الگوریتم کولونی مورچگان، برای یافتن جواب بهینه.

در دنیای واقعی و برای حل مسائل بهینه سازی مورچه‌های مصنوعی باید به گونه‌ای تعریف شوند که بتوانند همین رفتار را از خود نشان دهند. همچنین برای حل مسائل، مدلسـازی مسـئله نیـز مهـم

است. این مدلسازی باید به گونه‌ای صورت بگیرد که رفتار مورچه‌ها دچار تغییرات اساسی نشود و بتوانند بر روی مسیرهایی که از آن عبور می‌کنند فرومون برجای بگذارند و این مسیرها پس از ساخت جواب دارای معنی بوده و قابلیت بازیابی داشته باشند.

همچنین دو خصوصیت مهم از رفتار مورچه‌های طبیعی باید در مورچه‌های مصنوعی نیز وجود داشته باشد و یا مورچه‌های مصنوعی (روش جستجو) بتوانند آنها را پیاده سازی کنند. این دو خصوصیت عبارتند از:

- متغیرهای موجود در زندگی مورچه‌ها با متغیرهای مسئله هماهنگ، هم بُعد و هم وزن باشند. بطور مثال مسئله قابلیت تبدیل شدن به گراف و یا شکلی هماهنگ و قابل پیمودن را داشته باشد و مسیرهای ایجاد شده برای عبور مورچه‌ها وزن دار بوده و حداقل در یک متغیر (بطور مثال، مسافت یا هزینه) با هم اختلاف داشته باشند. (وابسته کردن متغیرهای حالت به حالت‌های مختلف مسئله).
- عامل‌ها فقط دسترسی محلی به اینگونه متغیرها دارند. منظور از عامل، همان عناصر سازنده جواب یا مورچه‌ها می‌باشند که این عامل‌ها نباید به تمامی متغیرها دسترسی داشته و یا بتوانند کل مسیر را به صورت یکجا ببینند. مورچه‌های مصنوعی که برای حل مسائل بکار می‌روند باید فقط مسیر خود را شناخته و فقط بر روی آن مسیر فرومون ریزی کنند. بطور مثال درصورتیکه مورچه‌ها بتوانند به مسیر دارای فرومونِ بیشتر دسترسی داشته باشد و قوانین حرکت تصادفی از مجموعه حرکات آنها حذف شود، در یک مسیر زیربهینه به دام می‌افتند و هیچ گاه مسیر با ارزش بهینه پیدا نمی‌شود (Dorigo & Socha, 2006).

از آنجا که ایده‌ی اصلی ACO کاملاً از زیست‌شناسی الهام گرفته است. بنابراین شباهت‌های بسیاری بین مورچه‌های واقعی و مصنوعی موجود می‌باشد. این شباهت‌ها را می‌توان اینگونه برشمرد:

هر دو مورد کولونی مورچه‌های واقعی و مصنوعی از جمعیتی از افراد تشکیل شده‌اند که برای رسیدن به یک هدف خاص، با هم کار می‌کنند. یک کولونی یک جمعیت از عامل‌های ساده (رفتار

ساده و قابل بیان به صورت ریاضی)، مستقل (نتیجه کار هر عامل مستقل از دیگری می‌باشد) و غیر همزمان (هر عامل می‌تواند کار خود را مستقل از دیگر عامل‌ها –افراد- انجام دهد) است که برای یافتن یک راه حل خوب برای مسئله‌ها همکاری می‌کنند (منظور از همکاری در کولونی، به نتیجه رساندن هدف کلی کولونی می‌باشد). در مورد مورچه‌های واقعی، مسئله یافتن غذا است، درحالی که در مورد مورچه‌های مصنوعی، یافتن راه حلی مناسب برای یک مسئله‌ی بهینه سازی داده شده می‌باشد. یک مورچه‌ی تنها (چه مورچه‌ی واقعی و چه مصنوعی) می‌تواند یک پاسخ برای مسئله‌اش بیابد؛ ولی فقط همکاری بین تعداد زیادی از آنها از طریق استیگمرجی آنها را قادر می‌سازد تا راه‌حل‌های خوب را پیدا کنند.

در مورد مورچه‌های واقعی، آنها ماده‌ی شیمیایی‌ای را برجای می‌گذارند که فرومون نامیده می‌شود و نیز فقط آن را می‌شناسند. مورچه‌های واقعی به هنگام پیمودن یک مسیر، بطور خودکار این ماده‌ی شیمیایی را بر روی زمین برجای می‌گذارند. مورچه‌های مصنوعی در دنیای مجازی زندگی می‌کنند، از این رو فقط مقادیر عددی را ویرایش می‌کنند (که برای قیاس فرومون مصنوعی نامیده می‌شود) که این موضوع بسته به حالت‌های مسئله متفاوت می‌باشد. یک توالی از مقادیر فرومون که مربوط به حالت مسئله است مسیر فرومون مصنوعی نامیده می‌شود. در ACO، مسیرهای فرومون مصنوعی به معنای شالوده‌ی ارتباطی میان مورچه‌ها می‌باشد. در مقایسه با تبخیر فرومون فیزیکی در کولونی مورچه‌های واقعی، یک مکانیزم نیز به مورچه‌های مصنوعی اجازه می‌دهد تا حافظه‌شان را فراموش کنند و بر روی جهات جستجوی محتمل جدیدی متمرکز شوند (Dorigo & Socha, 2006).

مشابه مورچه‌های واقعی، مورچه‌های مصنوعی نیز جواب‌شان را بطور مداوم با حرکت از یک جواب مسئله به دیگری می‌یابند. مورچه‌های واقعی به راحتی حرکت کرده و جهتی را براساس غلظت محلی فرومون و سیاست تصمیم‌گیری اتفاقی انتخاب می‌کنند. همچنین مورچه‌های مصنوعی راه‌حل‌ها را گام به گام و با حرکت از یک حالت موجود مسئله و تصمیم‌گیری‌های اتفاقی در هر گام تولید می‌کنند.

با این حال چند تفاوت مهم بین مورچه‌های واقعی و مصنوعی وجود دارد:

- مورچه‌های مصنوعی در یک دنیای گسسته زندگی می‌کنند؛ آنها بطور مداوم در میان یک مجموعه‌ی متناهی از حالت‌های مسئله حرکت می‌کنند.
- به‌هنگام‌سازی[1] فرومون (یعنی برجای گذاری فرومون و تبخیر آن) توسط مورچه‌های مصنوعی دقیقاً به روش مورچه‌های واقعی صورت نمی‌گیرد. بعضی اوقات به‌هنگام‌سازی فرومون فقط توسط بعضی از مورچه‌های مصنوعی صورت می‌گیرد و آن هم پس از اینکه یک راه‌حل تولید شد (Dorigo & Socha, 2006) و (Dorigo & Di Caro, 2006).

بعضی از پیاده‌سازی‌های (نسخه‌های) مورچه‌های مصنوعی از مکانیزم‌های اضافی نیز استفاده می‌کنند که در مورد مورچه‌های واقعی، عینیتی ندارد. مثال‌هایی در این زمینه شامل پیش‌بینی، جستجوی محلی، پاک کردن ردپا و غیره می‌باشد.

الگوریتم کولونی مورچگان تاکنون برای حل بسیاری از مسائل بکار رفته است. بیشتر مسائلی که تا به حال توسط این الگوریتم مورد بررسی قرار گرفته‌اند و برای آنها جواب‌هایی قابل قبول پیدا شده است دارای شکلی گراف مانند و یا جدولی بوده‌اند که الگوریتم کولونی مورچگان توانسته است با روش مخصوص به خود آنها را حل نماید. این الگوریتم اولین بار بر روی مسئله فروشنده دوره‌گرد اجرا شد که به دلیل ارتباط این مسئله با این پایان نامه، در همین فصل شرح داده می‌شود.

۳-۶ الگوریتم کولونی مورچگان

روش بهینه‌سازی توسط کولونی مورچگان توسط دوریگو[2] و همکارانش به عنوان یک روش فراابتکاری بهینه‌سازی ترکیبی بطور رسمی معرفی شده و از آن هنگام در حل مشکل بسیاری از مسائل بهینه‌سازی ترکیبی[3] (COP) مورد استفاده قرار گرفته است (Dorigo & Socha, 2006).

1 - Update
2 - Dorigo
3 - Combinatorial Optimization Problem

با یک مسئله‌ی بهینه سازی داده شده، اولین قدم برای استفاده از ACO در حل کردن مسئله، تعریف یک مدل مناسب می‌باشد. سپس این مدل برای تعریف جزء اصلی ACO، یعنی مدلسازی فرومون استفاده می‌شود (Dorigo & Socha, 2006) و (Cordon et al, 2002).

بیان صورت یک مسئله برای کاربرد الگوریتم کولونی مورچگان و یا یکی از نسخه‌های این الگوریتم عبارت است از:

فضای جستجو (به این معنی که الگوریتم باید بتواند از میان چندین پاسخ، بهترین را پیدا کند) نتیجه این عبارت این است که این الگوریتم برای مسائلی می‌تواند مورد استفاده قرار گیرد که دارای چندین جواب باشند، به این معنی که الگوریتم کولونی مورچگان به دنبال پاسخ نمی‌گردد، بلکه به دنبال بهترین پاسخ می‌گردد. نتیجه دیگری که می‌توان از این عبارت داشت این است که باید روشی را برای یافتن خود پاسخ درنظر داشت و بر اساس آن روش و کاربرد الگوریتم کولونی مورچگان و یا یکی دیگر از نسخه‌های آن، راه‌حل بهینه را یافت.

مسئله دیگری که باید به آن توجه داشت، داشتن و یا تعریف کردن محدودیت‌ها به صورت دقیق می‌باشد. این محدودیت‌ها هستند که شرایط یک مسئله و یا کیفیت یک جواب و مهمتر از همه سرعت رسیدن به جواب را تعیین می‌کنند. همچنین نکته قابل توجه در اینجا این است که تمام جواب‌های ممکن که می‌بایست در میان آنها جستجو صورت گیرد، باید محدودیت‌ها را برآورده سازند (Cordon et al, 2002).

مورد با اهمیت دیگر تابع هدف می‌باشد که از طریق الگوریتم کولونی مورچگان باید حداقل‌سازی[1] شود. درحالیکه مسئله‌ای که ما در این پایان‌نامه پیش رو داریم حداکثرسازی[2] مجموع ارزش‌ها می‌باشد، که در جای خود بحث خواهد شد.

در ACO، مورچه‌های مصنوعی به وسیله‌ی پیمودن چیزی که گراف ساختاری $G_C(V,E)$ نامیده می‌شود، راه‌حلی را برای مسئله بهینه سازی تولید می‌کنند. یک گراف ساختاری متشکل است از یک

1 - Minimizing
2 - Maximizing

مجموعه راس‌های V و یک مجموعه یال E. مورچه‌ها از یک راس به راس دیگر از طریق یال‌های گراف حرکت می‌کنند، و مرتباً یک راه‌حل جزئی را می‌سازند. به علاوه مورچه‌ها میزان معینی از فرومون را روی مولفه‌ها برجای می‌گذارند، که هم روی راس‌ها و هم روی یال‌هایی است که از آن می‌گذرند. میزان $\Delta\tau$ فرومون برجای گذاشته شده وابسته به کیفیت راه‌حل یافت شده می‌باشد. مورچه‌های بعدی از این اطلاعات فرومونی به عنوان راهنمایی به سمت ناحیه‌های امیدبخش فضای جستجو استفاده می‌کنند (Dorigo et al, 1999) و (Dorigo & Stutzle, 2004).

۳-۷ تولید راه‌حل توسط مورچه‌ها

تعداد m مورچه برای یافتن یک پاسخ بهینه، از مجموعه کل پاسخ‌های ممکن شروع به تولید پاسخ‌ها می‌کنند؛ همزمان با تولید هر پاسخ، کار ارزیابی این پاسخ نیز انجام خواهد گرفت. با تولید هر پاسخ، این پاسخ به مجموعه‌ی پاسخ‌های ممکن اضافه می‌شود. در اینجا باید دقت کرد که پاسخ‌ها با توجه به یک مجموعه به نام همسایه‌های شدنی تولید یا ساخته می‌شوند. مجموعه‌ی همسایه‌های شدنی، مجموعه‌ای از پاسخ‌هاست (و یا می‌تواند به عنوان یک محدودیت نیز درنظر گرفته شود) که بر اساس آن، جواب باید شرایطی را داشته باشد. این شرایط برای یک حالت کلی می‌توانند به این صورت بیان شوند:

- برگشتن به عقب ممنوع می‌باشد (در نتیجه فقط جواب‌ها یا گره‌هایی می‌توانند در یافتن پاسخ آتی درنظر گرفته شوند که در مسیر بازگشت قرار نداشته باشند)؛
- گذشتن از گره‌هایی که یک بار بازدید شده‌اند ممنوع می‌باشد، مجموعه‌ی گره‌هایی که پیش از این بازدید شده‌اند باید در یک لیست ممنوعه قرار گیرند و یا امکان دستیابی مورچه به این گره‌ها (یا مسیرها) غیرممکن شود.

انتخاب یک مولفه از راه‌حل بطور احتمالی و از هر مرحله‌ی تولید جواب انجام می‌گیرد. به این معنی که در هر گام برای تولید پاسخ بعدی (منظور از پاسخ بعدی، یک مولفه از پاسخ نهایی می‌باشد) این پاسخ بطور احتمالی ولی براساس قانون رسیدن به جواب انجام می‌گیرد. قوانین دقیق برای انتخاب احتمالاتی مولفه‌های راه‌حل، براساس انواع مختلف ACO متفاوت هستند. شناخته شده‌ترین قاعده متعلق به سیستم مورچه (AS) می‌باشد (رابطه ۳-۲):

$$p(c_{ij} \mid s^p) = \frac{\tau_{ij}^{\alpha} \cdot \eta(c_{ij})^{\beta}}{\sum_{c_{il} \in (s^p)} \tau_{il}^{\alpha} \cdot \eta(c_{ij})^{\beta}} \qquad \text{(رابطه ۳-۲)}$$

$$\forall c_{ij} \in N(s^p)$$

که در این رابطه:

$p(c_{ij})$: احتمال انتخاب یک مولفه،

c_{ij}: یک مولفه جواب که از ناحیه i شروع شده و به ناحیه j خاتمه می‌یابد،

s^P: مجموعه جواب‌های شدنی،

τ_{ij} مقدار فرومون مرتبط با مولفه‌ی c_{ij} است (به معنی مقدار فرومون برجای گذاشته شده بر روی هر مسیر)،

$\eta(\cdot)$ تابعی است که در هر مرحله‌ی ساخت، یک مقدار ابتکاری را به هر مولفه‌ی راه‌حل شدنی $c_{ij} \in N(s^p)$ تخصیص می‌دهد. مقادیری که به وسیله‌ی این تابع برمی‌گردند بطورکلی اطلاعات ابتکاری نامیده می‌شود (اطلاعات ابتکاری بسته به نوع مسئله می‌توانند متفاوت باشند. مثلاً درباره خود مورچه‌ها، آنها ممکن است *نزدیکترین گره بعدی* را انتخاب کنند، در اینصورت اطلاعات ابتکاری بصورت فاصله خواهد بود).

α و β پارامترهای مثبتی هستند که مقدار آن‌ها اهمیت نسبی فرومون را در مقابل اطلاعات ابتکاری نشان می‌دهد. در اینجا α و β دو پارامتر کنترل کننده‌ی اهمیت نسبی شدت مسیر $\tau_{ij}(t)$ و قابلیت بازدید (منظور از قابلیت بازدید، دور یا نزدیک بودن شهرهایی است که مورچه‌ها قصد دارند در

مرحله بعدی حرکت خود به سمت آنها بروند) η_{ij} می‌باشند. با $\alpha = 0$، فقط قابلیت بازدید شهر به حساب می‌آید یعنی در هر مرحله نزدیکترین شهر انتخاب می‌شود. در حالت عکس، با $\beta = 0$ فقط مسیرهای فرومون دارای اهمیت می‌شوند (Dorigo & Socha, 2006).

۳-۸ به‌هنگام‌سازی فرومون‌ها

در الگوریتم کولونی مورچگان، مورچه‌ها گره‌هایی را که هنگام مسیر رفت از آنها عبور کرده‌اند و هزینه یا ارزش کمان‌های طی شده (درصورتیکه گراف دارای وزن باشد) را به خاطر می‌سپارند. بنابراین آنها می‌توانند هزینه جواب‌هایی را که تولید کرده‌اند، ارزیابی کنند و این ارزیابی را برای تنظیم میزان فرومونی که در مسیر بازگشتشان از خود به جای می‌گذارند به کار برند (Mullen et al, 2009).

هدف به‌هنگام‌سازی فرومون، افزایش دادن مقادیر فرومون مرتبط با راه‌حل‌های خوب یا امیدوارکننده است، و نیز کاهش دادن مقادیر فرومون مرتبط با راه‌حل‌های بد. معمولاً این هدف ۱) با کاهش دادن تمام مقادیر فرومون در طول تبخیر فرومون و ۲) با افزایش دادن مقادیر فرومون مرتبط با یک مجموعه‌ی منتخب از راه‌حل‌های خوب یعنی S_{upd} به صورت رابطه ۳-۳ حاصل می‌شود:

$$\tau_{ij} \longleftarrow (1-\rho)\tau_{ij} + \rho \sum_{s \in S_{upd} | c_{ij} \in S} F(s)$$ (رابطه ۳-۳)

که در اینجا S_{upd} مجموعه راه‌حل‌هایی است که برای به‌هنگام‌سازی استفاده شده‌اند، $\rho \in (0,1]$ پارامتری است که نرخ تبخیر نامیده شده و $F : S \longrightarrow R_0^+$ تابعی است به این صورت:

$$f(s) < f(s') \Rightarrow F(s) \geq F(s'), \forall s \neq s' \in S$$ (رابطه ۳-۴)

$F(\cdot)$ اغلب تابع جورشدگی نامیده می‌شود (Dorigo et al, 1999) و (Mullen et al, 2009).

برای جلوگیری از همگرایی بسیار سریع الگوریتم، تبخیر فرومون امری ضروری است. این عمل، نوعی فراموشی مفید را اجرا می‌کند که توجه فرایند جستجو را به ناحیه‌ای جدید در فضای جستجو جلب می‌نماید.

تبخیر رد فرومون را می‌توان به عنوان یک مکانیزم اکتشافی نیز درنظر گرفت که از همگرایی سریع همه مورچه‌ها به سمت یک مسیر نسبتاً بهینه جلوگیری می‌کند. در حقیقت، کاهش چگالی فرومون به اکتشاف مسیرهای متفاوت در طول فرایند کمک می‌کند. در کولونی مورچه‌های واقعی نیز رد فرومون تبخیر می‌شود ولی در آنجا نقش مهمی بازی نمی‌کند. ولی در مورد مورچه‌های مصنوعی، تبخیر فرومون نقشی مهم دارد، زیرا مسائل بهینه‌سازی که مورچه‌های مصنوعی آنها را حل می‌کنند، از مسائلی که مورچه‌های واقعی می‌توانند حل کنند، بسیار پیچیده‌ترند. مکانیزمی شبیه تبخیر که به فراموش شدن خطاها و یا انتخاب‌های ضعیفی که قبلاً صورت گرفته‌اند می‌انجامد باعث بهبود مستمر در ساختار حل مسئله مدلسازی شده می‌شود و به نظر می‌رسد که این مکانیزم، برای مورچه‌های مصنوعی ضروری باشد. به علاوه، تبخیر مصنوعی فرومون وظیفه مهمی در محدود ساختن حداکثر مقدار فرومون قابل دسترسی ایفا می‌کند (Dorigo & Stutzle, 2004).

۳-۹ بکار بردن جستجوی محلی

هنگامی که راه‌حل‌ها تولید شدند، پیش از به‌هنگام‌سازی فرومون، اغلب ممکن است چند مرحله بسته به انتخاب نیاز باشد. این مراحل اغلب فعالیت‌های شیطانی (یا اهریمنی، روحی، جنی) نامیده می‌شوند، و می‌توانند برای اجرای مسئله‌ی خاصی و/یا فعالیت‌های متمرکز شده، که نمیتوانند توسط فقط یک مورچه صورت گیرد بکار می‌رود. فعالیت‌های اهریمنی را می‌توان اینگونه بیان کرد: پس از یک دور جستجوی مورچه‌ها و یافتن یک پاسخ محلی (و نه سراسری)، عاملی خارج از توان و دید مورچه‌ها و بطور مستقل از آنها جواب، مسئله یا مقدار فرومون را بررسی کرده و به صلاحدید خود آنها را تغییر می‌دهد. این روش در بعضی مسائل می‌تواند به بهتر شدن جواب و یا همگرایی سریعتر مورچه‌ها منجر شود. فعالیت‌های شیطانی‌ای که بیشتر مورد استفاده قرار می‌گیرند شامل بکار بردن جستجوی محلی برای راه‌حل ایجاد شده می‌باشد: یعنی *راه حلی که بطور محلی بهینه‌سازی شده* برای

تصمیم‌گیری در مورد اینکه *کدام فرومون به‌هنگام‌سازی شود* بکار می‌رود (Dorigo & Stutzle, 2004).

همچنین جستجوی محلی را می‌توان به این صورت تعریف کرد:

جستجو در محدوده‌ای از فضای جستجو به صورتیکه به دلایلی مانند بزرگ بودن فضای جستجو و یا وجود جواب‌های ممکن ولی زیر بهینه، فرایند جستجو به دیگر نقاط تسرّی نیابد. همچنین جستجوی محلی می‌تواند به سبب وجود فعالیت‌های اهریمنی نیز صورت گیرد (Dorigo & Stutzle, 2004).

۳-۱۲ انواع الگوریتم‌های مورچگان

از زمان معرفی سیستم مورچگان (به عنوان اولین نسخه از الگوریتم‌های ابتکاری رفتار مورچگان) در سال ۱۹۹۱، تاکنون چندین نسخه از الگوریتم‌های مورچگان، عمدتاً توسط خود دوریگو معرفی شده‌اند، عمده‌ترین تفاوت بین این الگوریتم‌های بسط داده شده در شیوه به‌روزآوری فرومون آن‌ها و نیز برخی نکات اضافی در مدیریت ردهای فرومون موجود است. الگوریتم‌های معرفی شده تا سال ۱۹۹۹ عبارتند از:

۳-۱۲-۱ سیستم مورچگان و مورچه‌ی ممتاز (Ant System & elitism)

معرفی شده در سال ۱۹۹۲، گونه‌ی اولیه‌ای از "سیستم مورچگان" است که مورچه‌های ممتاز یا برگزیده را معرفی نمود. در این نسخه، مورچه ممتاز یا بهترین مورچه (که کوتاهترین مسیر را طی کرده) مقدار زیادی فرومون برجای می‌گذارد تا احتمال رسیدن به بهترین جواب را برای دیگر مورچه‌های جستجو کننده‌ی محتمل‌ترین پاسخ افزایش دهد (Dorigo & Socha, 2006).

۳-۱۲-۲ Ant-Q

در این گونه از AS، که در سال ۱۹۹۵ معرفی شده است، قانون به‌هنگام سازی محلی از یک نوع راه‌حل از مسئله "Q-learning" الهام گرفته است. با این وجود در مقایسه با AS هیچ بهبودی را نمی‌توان اثبات کرد. در کنار آن، حتی، به نظر محققان، این الگوریتم بیشتر از یک پیش نسخه از "سیستم کولونی مورچگان" نیست (Dorigo & Socha, 2006).

۳-۱۲-۳ سیستم کولونی مورچگان ((Ant Colony System (ACS))

الگوریتم "سیستم کولونی مورچگان" (ACS) برای بهبود عملکرد نخستین الگوریتم برای مسائل با ابعاد بالاتر در سال ۱۹۹۲ معرفی شد. ACS براساس تغییراتی در AS اینگونه بیان می‌شود:

الف) ACS یک قانون گذر (انتقال) را براساس پارامتر q_0 که $(0 \le q_0 \le 1)$ معرفی می‌کند، که بین گوناگونی و تشدید، تعادل را بیان می‌کند.

مورچه‌ی k که هم اکنون روی گره i قرار دارد، گره j را براساس رابطه زیر انتخاب می‌کند:

$$j = \begin{cases} \arg\max_{u \in J_i^k} [(\tau_{iu}(t)) \cdot (\eta_{iJ})^{\beta}] & \text{if} \quad q \le q_0 \\ J & \text{if} \quad q > q_0 \end{cases} \qquad \text{(رابطه ۳-۵)}$$

در اینجا q یک متغیر تصادفی بطور یکسان توزیع شده در [0,1] است و $J \in J_i^k$ یک شهر انتخاب شده برحسب تصادف و براساس احتمال زیر می‌باشد:

$$p_{iJ}^k(t) = \frac{(\tau_{iJ}(t)) \cdot (\eta_{iJ})^{\beta}}{\sum_{l \in J_i^k} (\tau_{il}(t)) \cdot (\eta_{il})^{\beta}} \qquad \text{(رابطه ۳-۶)}$$

بنابراین برطبق پارامتر q_0 دو رفتار ممکن است: اگرکه $q > q_0$، انتخاب بر همان اساس الگوریتم AS صورت می‌گیرد و سیستم تمایل به گوناگونی دارد؛ برعکس، اگرکه $q \le q_0$ آنگاه سیستم به سمت تشدید می‌رود. در عین حال برای $q \le q_0$ الگوریتم بیشتر از اطلاعات

جمع‌آوری شده توسط سیستم استفاده کرده و نمی‌تواند یک مسیر اکتشاف نشده را انتخاب کند.

ب) مدیریت مسیرها به دو سطح تقسیم شده است: یک به‌هنگام سازی محلی و یک به‌هنگام سازی کلی. هر مورچه در زمان به‌هنگام سازی محلی و براساس رابطه زیر بر روی مسیر فرومون برجای می‌گذارد:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\tau_0$$ (رابطه ۳-۷)

که در اینجا τ_0 مقدار اولیه‌ی مسیر است. در هر بار عبور، یال‌های بازدید شده مقدار فرومون‌شان کاهش می‌یابد، که گوناگونی را با به حساب آوردن مسیرهای اکتشاف نشده تقویت می‌کند. در هر تکرار، به هنگام سازی کلی به این صورت انجام می‌گیرد:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\Delta\tau_{ij}(t)$$ (رابطه ۳-۸)

در اینجا یال‌های (i, j)، به طول بهترین گشت T^+ با طول L^+ تعلق دارند و $\Delta\tau_{ij}(t) = \frac{1}{L^+}$. در اینجا فقط بهترین مسیر به هنگام سازی می‌شود، که در عمل تشدید با انتخاب کردن بهترین جواب سهیم می‌شود.

ج) سیستم از یک لیست کاندیداها استفاده می‌کند. این لیست برای هر شهر v، نزدیک‌ترین همسایه‌ها را که براساس افزایش فاصله طبقه‌بندی شده ذخیره می‌کند. یک مورچه به یک یال به سوی یک شهر و از روی لیست فقط زمانی توجه می‌کند که اگر فقط این شهر به تازگی اکتشاف شده باشد. در حالت خاص، اگر که تمام یال‌های موجود در لیست بازدید شده باشند، انتخاب براساس قانون فرمول ۳-۴ انجام می‌گیرد. اگر نه که نزدیکترین شهری که بازدید نشده انتخاب می‌شود (Dorigo & Socha, 2006) و (Dorigo et al, 1991).

۳-۱۲-۴ سیستم کولونی مورچگان و سه برگزیده (ACS & 3-opt)

این الگوریتم که در سال ۱۹۹۶ بیان شده است، ترکیبی از ACS و یک الگوریتم جستجوی محلی از نوع سه برگزیده است. در اینجا جستجوی محلی دارای مقدار اولیه‌ای است تا جواب‌های یافته شده تابه‌حال توسط مورچگان را بهبود دهد (و بنابراین مورچگان را به نزدیکترین بهینه‌ی محلی برساند)، (Dorigo & Socha, 2006).

۳-۱۲-۵ Max-Min Ant System (MMAS)

این نوع از الگوریتم که در سال ۱۹۹۱ معرفی شده است (بطور مخفف MMAS) براساس الگوریتم AS بنا شده و بعضی تفاوت‌های قابل توجه را بیان می‌کند:

- فقط بهترین مورچه، یک مسیر فرومونی را به هنگام سازی می‌کند؛
- مقادیر فرومون مسیرها توسط τ_{min} و τ_{max} محدود شده است؛
- مسیرها دارای مقدار اولیه‌ی از فرومون برابر با τ_{max} هستند؛
- به‌هنگام سازی مسیرها بطور نسبی انجام می‌شود، قویترین مسیر کمتر از ضعیف‌ترین مسیر تقویت می‌شود؛
- مقداردهی اولیه مسیرها می‌تواند دوباره انجام شود.

بنابراین نتایج با به هنگام کردن بهترین جواب با یک فرکانس قدرتمند در حال افزایش، در طول اجرای الگوریتم، بدست می‌آید (Dorigo & Socha, 2006).

۳-۱۳ روش جستجوی تابو

یکی از روش‌هایی که در الگوریتم کولونی مورچگان کاربردهای فراوانی دارد، روش جستجوی تابو می‌باشد. معمولاً این دو روش به همراه یکدیگر برای حل مسائل بکار گرفته می‌شوند و در واقع روش جستجوی تابو تکمیل کننده بخش جستجو در الگوریتم کولونی مورچگان می‌باشد.

روش جستجو با کمک تابوها، یا بطور ساده جستجوی تابو، در سال ۱۹۸۶ توسط گلوور[1] فرمول‌بندی شد. مشخصات اصلی این روش برپایه‌ی استفاده از مکانیزم‌های الهام گرفته شده از حافظه‌ی انسان است. روش تابو، مسیری را برخلاف روش‌های دیگر مانند شبیه‌سازی تبریدی در پیش می‌گیرد که بطور کلی از حافظه استفاده نمی‌کند، و ازاینرو برای یادگیری درس‌هایی از گذشته مناسب نیست (روش شبیه‌سازی تبریدی). اصول روش جستجوی تابو ساده هستند: مشابه روش شبیه‌سازی تبریدی، روش تابو نیز در همان زمان فقط با یک "پیکربندی جاری" (در ابتدا، هر جوابی) ایفای وظیفه می‌کند، که در طول "تکرارهای" موفق نیز به‌هنگام سازی می‌شود (Glover & Lagona, 1997).

این روش در هر تکرار به به‌هنگام سازی و بکار بردن یک لیست جلوگیری کننده از حرکت‌ها نیاز دارد یعنی "لیست تابو": این لیست که این روش نیز نامش را از این لیست گرفته، شامل m حرکت $(t \longrightarrow s)$ است، که مخالف m حرکت اخیراً انجام گرفته $(s \longrightarrow t)$ می‌باشد. به این معنی که لیست تابوها شامل مجموعه‌ای از حرکت‌هایی است که تا به حال انجام گرفته و الگوریتم دیگر مجاز به انجام آنها نیست (خود به خود این کار بازگشت به عقب را غیرممکن می‌کند و نوعی نظم را به جستجو می‌بخشد) و به اصطلاح، این حرکت‌ها تابو شده‌اند؛ و یا لیستی است که الگوریتم فقط مجاز به حرکت از طریق این لیست می‌باشد و با انجام هر حرکت، یکی از مولفه‌های این لیست (که متعلق به حرکت جاری بوده است) از آن حذف می‌گردد.

[1] -Glover

ازاینرو این الگوریتم یک شکل ناقص و اولیه از حافظه را مدل‌سازی می‌کند، که حافظه‌ی کوتاه مدت جواب‌ها را تشکیل می‌دهد. دو مکانیزم دیگر به نام‌های تشدید[1] و گوناگونی[2] نیز بمنظور تجهیز کردن این الگوریتم به یک حافظه‌ی بلند مدت بکار می‌روند. تشدید شامل جستجوی اکتشافی مناطق خاص در فضای جواب است که مناطق محتمل و نویدبخش را شناسایی می‌کند. برعکس این عمل، فرایند گوناگونی بطور دوره‌ای جستجو را دوباره جهت‌دهی می‌کند تا مناطق بهینه را بیابد، یعنی آن مناطقی را که تاکنون بازدید نشده‌اند (Glover & Lagona, 1997).

در الگوریتم کولونی مورچگان، بخش‌های جستجو شده وارد لیست تابو می‌شوند تا از جستجوی دوباره این مناطق جلوگیری به عمل آید. می‌توان لزوم وجود لیست تابو را با ارائه یک یا چند محدودیت برای حرکت مورچه‌ها حذف کرد (Ling Huang & Jong Liao, 2008) و (Dorigo & Stutzle, 2004).

۳-۱۴ چگونگی کارکرد الگوریتم کولونی مورچگان

با توجه به موارد گفته شده می‌توان الگوریتم کولونی مورچگان را بطور کلی و برای حل کلیه انواع مسائل اینگونه بیان نمود:

- اولین مورچه یک جواب تولید می‌کند و برحسب جواب یافت شده روی مسیری که از آن عبور کرده است فرومون برجای می‌گذارد.
- مورچه‌های بعدی براساس رابطه (۳-۲) جستجوی خود را شروع می‌کنند و به دنبال جوابی بهتر هستند. آنها برای یافتن جواب به مسیرهای دارای فرومون نیز توجه می‌کنند.

[1] - Intensification
[2] - Diversification.

- پس از پایان یافتن جستجوی هر مورچه درصورتیکه جواب بهتر از قبل بود فرومون برجای گذاشته می‌شود. همچنین درصورتیکه پس از پایان یافتن جستجوی هر مورچه جواب بهتری یافت نشد، فرومون مسیرها برطبق رابطه (۳-۹) تبخیر می‌شود.
- در هر حالت مسیری که مورچه از آن عبور کرده است وارد لیست تابو خواهد شد.
- درصورتیکه پس از تعداد تکرارهای مشخص شده جواب بهتری یافت نشد و یا در پایان تعداد تکرارهای مشخص شده، جواب یافت شده به عنوان جواب بهینه برگشت داده می‌شود.

در این پایان نامه نیز از همین نوع کد استفاده خواهد شد. مدلسازی مسئله مورد نظر پایان نامه به همراه ذکر چند مثال در فصل ۵ به تفصیل ذکر شده است.

۳-۱۵ مسئله فروشنده دوره‌گرد[1]

یکی از مسائلی که تاکنون تمامی روش‌های ابتکاری برای حل آن آزمایش شده‌اند، مسئله فروشنده دوره‌گرد بوده است. از آنجا که الگوریتم کولونی مورچگان نیز با موفقیت برای این مسئله آزمایش شده است و نیز ارتباط این مسئله با حالت سه بعدی مسئله تعیین محدوده نهایی معادن، در اینجا بطور مختصر شرح داده شده است.

اولین مطالعات در مورد TSP و مطالعات مشابه آن به قرن ۱۹ باز می‌گردد. این مسئله از دهه ۱۹۵۰ بطور گسترده‌ای در تحقیق در عملیات و علوم رایانه مورد مطالعه قرار گرفته است و برای حل آن تکنیک‌های متعدد و متفاوتی ایجاد و استفاده شده است. تا اوایل دهه ۱۹۸۰ رویه‌های حل TSP اساساً از متدهای ابتکاری سازنده، الگوریتم‌های بهبود دهنده چرخشی و روش‌های حل دقیق مانند انشعاب و تحدید یا انشعاب و برش تشکیل شده بودند.

در اوایل دهه ۱۹۸۰، اَبَرابتکاری‌های بیشتری روی TSP آزمایش شدند. TSP اولین مسئله‌ای بود که الگوریتم شبیه سازی تبریدی برای حل آن بکار رفت (Dorigo & Stutzle, 2004).

[1] - Traveling Salesman Problem (TSP)

مسئله فروشنده دوره‌گرد، راجع به فروشنده‌ای است که از شهر محل سکونتش شروع به حرکت می‌کند و قصد دارد کوتاه‌ترین مسیر ممکن را از طریق مجموعه‌ای از شهرهای مشتری بیابد، بطوریکه هر شهر را قبل از بازگشت به شهر محل سکونتش فقط یک بار ملاقات کند. در این مسیر فروشنده از هر شهر مشتری دقیقاً یکبار عبور می‌کند و اجازه بازگشت به شهری که قبلاً از آن بازدید کرده است را ندارد. در ریاضیات برای مدلسازی این مسئله از یک گراف استفاده می‌کنند که به تعداد شهرها گره و به تعداد مسیرها کمان دارد و به هر کمان این گراف وزنی اختصاص داده می‌شود که بیانگر فاصله میان دو شهر است. این مسئله یک مسئله بهینه سازی ترکیبی چند جمله‌ای غیرقطعی سخت[1] است که تعداد زیادی از تحقیقات دانشمندان را به خود معطوف کرده است (Haupt & Haupt, 1998).

شایان ذکر است که نتایج بدست آمده از الگوریتم‌های دقیق برای حل TSP کاملاً جالب توجه هستند، بطوریکه در بهار سال ۲۰۰۲ میلادی بزرگترین مثالی که به صورت مستدل با استفاده از الگوریتم‌های دقیق به جواب بهینه رسید، تنها از ۱۵۱۱۲ شهر تشکیل شده بود. حل چنین مثال بزرگی نیازمند شبکه‌ای متشکل از ۱۱۰ پردازشگر بوده و مجموعاً زمانی معادل با ۲۲/۶ سال با استفاده از پردازشگر EV6 CPU 500 MHz را به خود اختصاص داده است (Dorigo & Stutzle, 2004).

[1] - Non-deterministic Polynominal-completeness Problem (NP-Hard)

فصل چهارم

طراحی محدوده نهایی معادن توسط الگوریتم کولونی مورچگان

فصل ۴

۴-۱ مقدمه

به منظور تعیین محدوده نهایی معادن روباز، در این پایان نامه از الگوریتم کولونی مورچگان استفاده شده است. در این فصل اصول و رویکردهایی که برای رسیدن به این هدف درنظر گرفته شده‌اند و همچنین نوع مدلسازی مسئله، تابع هدف و محدودیت‌ها ذکر می‌شوند. مدل ارائه شده و الگوریتم مدلسازی شده به همراه یک مثال نیز در این فصل ارائه خواهند شد.

۴-۲ طراحی محدوده نهایی معادن روباز با الگوریتم کولونی مورچگان

به منظور طراحی محدوده نهایی می‌بایست آشنایی مختصری با روش حل مسئله توسط الگوریتم کولونی مورچگان داشت، در زیر کد مجازی مربوط به الگوریتم کولونی مورچگان برای حل مسائل آورده شده است که در این پایان نامه نیز بر همین اساس عمل خواهد شد.

- آماده سازی الگوریتم و مدل؛
- تعریف محدودیت‌ها براساس مدلسازی صورت گرفته؛
- اولین مورچه براساس قوانین مخصوص حرکت، یک دور را بر روی مسیرها کامل می‌کند و یک جواب می‌سازد؛

- مورچه‌های بعدی بر روی مسیرها حرکت کرده و هر یک جواب خاص خود را بدست می‌آورند؛
- مورچه‌ای که بهترین جواب را بدست می‌آورد، بر روی مسیر خودش فرومون برجای می‌گذارد؛
- به عنوان دومین تکرار مورچه‌ها شروع به جستجو برای یافتن جواب می‌کنند؛
- در این تکرار، کم کم فرومون مربوط به جواب برحسب روابط موجود تبخیر می‌شود؛
- مورچه‌ای که بهترین پاسخ را بدست آورد، مسیر فرومونی را تغییر می‌دهد (به روز می‌کند)؛
- این کار آنقدر ادامه می‌یابد تا اینکه تعداد خاصی از تکرارها انجام شوند و یا در طی چند دور تکرار جواب بهتر نشود؛
- بهترین جواب بدست آمده به عنوان جواب بهینه خواهد بود؛
- بهترین مسیر بدست آمده به عنوان مسیر بهینه خواهد بود؛

۴-۲-۱ آماده سازی الگوریتم و مدل

الگوریتم کولونی مورچگان در اصل بر مبنای گراف استوار می‌باشد. در این الگوریتم، مورچگان از یک نقطه در گراف (مبدأ-لانه) شروع به حرکت کرده و به نقطه‌ی دیگری از گراف (مقصد-غذا) می‌رسند و سپس در مسیر برگشت اقدام به برجای‌گذاری فرومون می‌کنند.

از آنجا که تولید گراف در یک زبان برنامه‌نویسی کار آسانی نمی‌باشد، در هنگام تولید این برنامه رایانه‌ای از جدول به جای گراف استفاده شده است. لازم به یادآوری است که در مقاله‌ای که نویسنده این پایان‌نامه ارائه داده، یک مدل بر اساس گراف معرفی شده که در این مدل گرافی برای ارضا نمودن محدودیت‌های مسئله‌ی معادن روباز، –نظیر زاویه‌ی ۴۵ درجه‌ی دیواره‌های نهایی معدن- یال‌هایی با ارزش صفر (و نه به طول صفر) به وجود می‌آیند که در ریاضیات از آن به گراف دارای دور تعبیر می‌شود. چنین گرافی برای مسئله‌ی تعیین محدوده نهایی معادن روباز مناسب نمی‌باشد (عظیمی و کاکایی، ۱۳۸۸).

در جدول مورد استفاده در این برنامه‌ی رایانه‌ای، هر خانه جدول به شکل یک بلوک می‌باشد که در آن ارزش اقتصادی بلوک واقع شده است. محاسبات و تلاش براساس جستجوی مجموعه‌ای از بلوک‌ها استوار است که این مجموعه باید دارای بیشترین ارزش باشند.

طبق قرارداد ستون‌های جدول از راست به چپ و در جهت قائم از بالا به پایین شماره‌گذاری می‌شوند. جهت حرکت مورچه‌ها نیز از راست به چپ خواهد بود.

به عنوان جواب اولیه، یکی از بلوک‌های ماده معدنی موجود در سطح زمین (ردیف اول) انتخاب می‌شود، در صورتیکه چنین بلوکی وجود نداشت، یکی از بلوک‌های ماده معدنی در ردیف دوم و در صورت عدم وجود در ردیف سوم و الی آخر. این اولین جواب اولیه در مراحل شروع بکار جستجوی مورچه‌ها با بهترین ارزش تاکنون مقایسه شده و با آن جایگزین می‌شود.

۴-۲-۲ تعریف محدودیت‌ها

۴-۲-۲-۱ محدودیت‌های کلی

مسئله به صورت دو بُعدی بوده و جهت حرکت در مدل بلوکی از راست به چپ (و همچنین شماره‌گذاری ستون‌ها) مثبت بوده و نیز جهت حرکت در مدل بلوکی از بالا به پایین (و همچنین شماره‌گذاری سطرها یا ردیف‌ها) مثبت می‌باشد.

متغیر بیانگر ستون‌ها، i نامیده می‌شود که از ۱ تا n تغییر می‌کند و متغیر بیانگر سطرها j نامیده می‌شود که میزان آن نیز از ۱ تا m متغیر است.

مورد دیگری که باید به آن توجه داشت این است که برخلاف الگوریتم کولونی مورچگان که در آن هدف یافتن کوتاهترین مسیر می‌باشد، در این برنامه هدف یافتن بیش‌ترین ارزش است. به منظور هماهنگ کردن مدل با الگوریتم کولونی مورچگان می‌توان تمام جدول را در عدد منفی یک ضرب نمود و سپس به یافتن کمترین ارزش پرداخت. ولی از آنجا که این امر به کار با اعداد منفی منجر می‌شود، همان اعداد اصلی موجود در مدل بلوکی در محاسبات منظور شده‌اند.

۴-۲-۲-۲ محدودیت فنی استخراج

از آنجاکه در این پایان نامه شیب نهایی دیواره‌ها ۴۵ درجه درنظر گرفته شده است، ابعاد بلوک‌ها با اندازه یکسان می‌باشد (توضیح این نکته ضروری است که در این نوع مدل بندی می‌توان برای بدست آوردن شیب‌هایی به غیر از ۴۵ درجه ابعاد بلوک‌ها را در جهات مختلف متفاوت درنظر گرفت و کار را ادامه داد. این حالت از نظر مورچه‌ها تفاوتی با حالت ۴۵ درجه نخواهد داشت، زیرا این نوع مدلسازی مستقل از ابعاد بلوک‌ها می‌باشد). برای رسیدن به زاویه شیب ۴۵ درجه، مورچه‌ها می‌توانند در هر مرحله از حرکت فقط یک واحد (یک بلوک) به سمت پایین (در عمق) و یا بالا پیش بروند و پس از آن حتماً باید یک واحد به سمت چپ (جهت قراردادی در مدلسازی) حرکت کنند.

همچنین در هر مدل، ردیف‌ها و ستون‌ها به هر تعدادی که باشند، براساس محدودیت فنی یا همان محدودیت شیب ۴۵ درجه در دیواره‌های نهایی معدن، تعداد ردیف‌ها ممکن است محدود شوند.

می‌توان روابط زیر را برای تعداد ردیفها و ستون‌ها ارائه داد:

- اگر تعداد ستون‌ها فرد باشند:

تعداد ردیف‌ها = ([تعداد ستون‌ها / ۲]) + ۱

- اگر تعداد ستون‌ها زوج باشند:

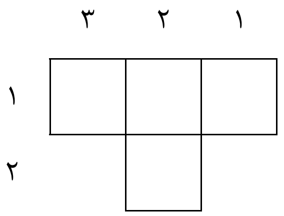
تعداد ردیف‌ها = (تعداد ستون‌ها / ۲)

که در روابط بالا علامت [...] به معنی جزء صحیح یک عبارت می‌باشد. این محاسبات خود می‌توانند به منظور بهینه‌سازی فرآیند جستجو به کار روند که در برنامه‌ی رایانه‌ای نیز گنجانده شده است.

۴-۲-۲-۳ محدودیت حرکت مورچه‌ها

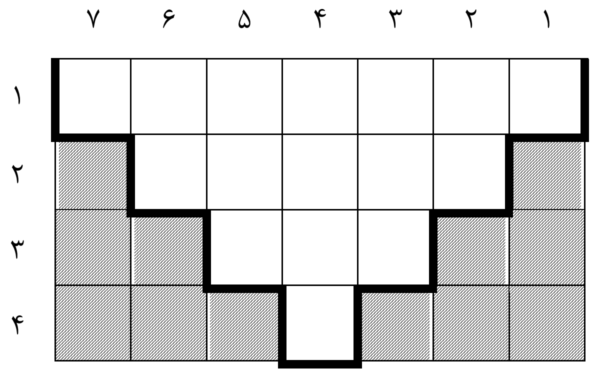
در الگوریتم درنظر گرفته شده برای یافتن یک جواب اولیه به این صورت عمل می‌شود که یک مورچه با حرکت از سطح زمین می‌تواند به سمت پایین، چپ و یا بالا حرکت کند (جهت حرکت از بالا

به پایین مثبت و از راست به چپ نیز مثبت می‌باشد). با این سه نوع حرکت، اکثر شکل‌های ممکن در یک مدل بلوکی قابلیت جستجو دارند ولی برای جستجوی شکل‌های دیگری نظیر شکل ۴-۱ می‌بایست یک نوع حرکت دیگر به مورچه‌ها اضافه شود. در این حالت و فقط با در نظر گرفتن سه نوع حرکت گفته شده، مورچه در بلوک (۲،۲) می‌ماند (به این نکته باید توجه داشت که مورچه ابتدا با شروع از بلوک (۱،۳) و با حرکت به سمت چپ بلوک (۱،۲) را جستجو می‌کند و با حرکت به سمت پایین به بلوک (۲،۲) می‌رسد؛ در چنین حالتی اگر فقط به سمت بالا حرکت کند، بسته به اینکه ارزش این بلوک (۱،۲) و بلوک قبل (۲،۲) چگونه بوده است و حرکت تصادفی بعدی به چه سمتی خواهد بود (زیرا امکان دارد که دوباره به سمت پایین باشد) حالت‌های پیچیده‌ای روی خواهد داد). به دلیل حرکت تصادفی مورچه‌ها ممکن است که بطور مداوم از بلوک (۱،۲) به سمت بلوک (۲،۲) حرکت کرده (حرکت به پایین) و پس از آن نیز حرکت آنها برعکس (حرکت به بالا) شود. برای جلوگیری از چنین مشکلی حالت چهارمی از حرکت به مورچه‌ها اضافه شد و آن توان حرکت همزمان یک واحد به سمت بالا و یک واحد به سمت چپ می‌باشد. (همچنین این حالت چهارم اضافی، از حرکت رو به عقب مورچه‌ها جلوگیری می‌کند).



شکل ۴-۱: در این شکل، حرکات به سمت چپ، بالا و پایین پاسخگو نیستند،

چنین حرکتی هم برای مدل ذکر شده در بالا و هم برای قسمت‌های ابتدایی و انتهایی مدل بلوکی و در جایی که محدودیت فنی استخراج وجود دارد قابل استفاده می‌باشد. به شکل ۴-۲ توجه کنید:

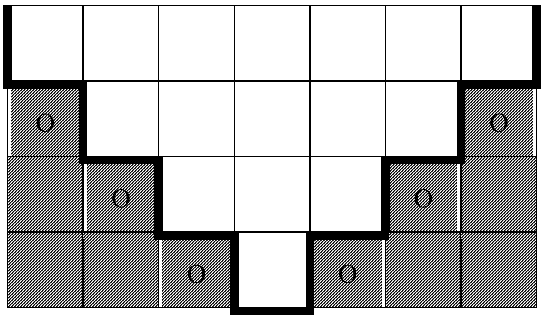


شکل ۴-۲: محدودیت فنی استخراج

در این شکل بلوک‌های (۲،۱) و (۲،۷) و (۳،۱) و (۳،۲) و (۳،۶) و (۳،۷) و ردیف چهارم به جز بلوک (۴،۴) را نمی‌توان جستجو یا استخراج کرد.

ذکر دوباره این نکته نیز ضروری به نظر می‌رسد که در این پایان نامه زاویه دیواره‌ها ۴۵ درجه درنظر گرفته شده‌اند. به همین دلیل از حرکت مورچه‌ها به نحوی که در دو مرحله متوالی به سمت بالا یا پایین حرکت کنند جلوگیری می‌شود.

همچنین هنگام جستجو باید از انحراف مورچه‌ها به سمت محدودیت فنی جلوگیری نمود، به شکل ۴-۳ توجه نمایید:



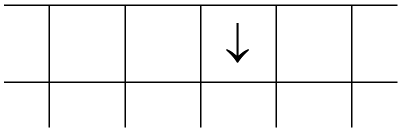
شکل ۴-۳: باید از انحراف مورچه‌ها به سمت مناطق هاشور خورده جلوگیری کرد.

در این شکل قسمت هاشور خورده باید از فرآیند جستجوی مورچه‌ها در امان باشد. برای این منظور در مدلسازی انجام شده ارزش بلوک‌هایی که در قسمت هاشور خورده با علامت دایره مشخص شده‌اند، با ارزشی برابر با یک عدد منفی بزرگ جایگزین شده است و هر مورچه پس از رسیدن به

چنین بلوک‌هایی (البته در سمت چپ مدل، چون به طور خودکار و به دلیل حرکت از راست به چپ، در سمت راست مدل مورچه‌ها نمی‌توانند وارد محدوده ممنوع شوند) حرکت خود را به سمت بالا و چپ ادامه می‌دهد.

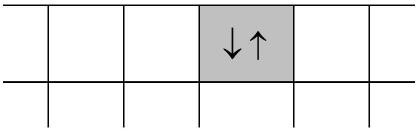
۴-۲-۳ قوانین حرکت مورچه‌ها

برای جستجوی مورچه‌ها در این بلوک‌ها قانون‌هایی ارائه شده است که در ادامه به آنها اشاره خواهد شد. هر مورچه پس از انتخاب اولین بلوک از یک ستون از مدل بلوکی (در جدول، همانند شکل ۴-۴) می‌تواند حرکت خود را شروع کند:



شکل ۴-۴: یک مورچه ابتدا از سطر اول وارد مدل بلوکی می‌شود.

و یا با بازگشت به سمت بالا از جدول خارج شده و به جستجو خاتمه دهد. در این حالت فقط یک بلوک مورد جستجو واقع شده است(شکل ۴-۵).

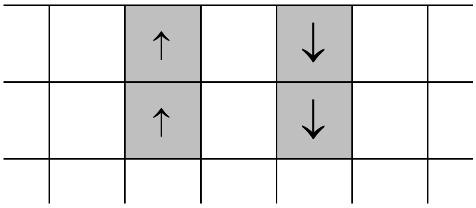


شکل ۴-۵: ممکن است مورچه بلافاصله پس از ورود به مدل بلوکی از آن خارج شود

همچنین هر مورچه می‌تواند پس از انتخاب یک ستون از مدل بلوکی به چپ حرکت نموده (شکل ۴-۶) و پس از آن به بالا یا چپ حرکت کند.

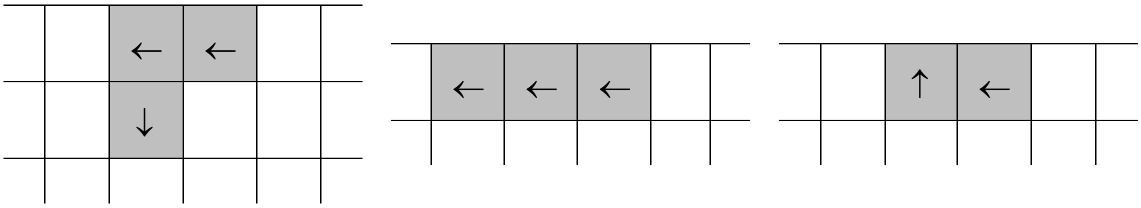
شکل ۴-۶: پس از انتخاب هر ستون (راست)، مورچه باید به سمت چپ حرکت کند.

باید توجه داشت که در هر مرحله‌ای به دلیل برآوردن محدودیت‌های فنی، یک مورچه نمی‌تواند در دو مرحله‌ی متوالی به سمت پایین یا بالا حرکت کند (همانند شکل ۴-۷):



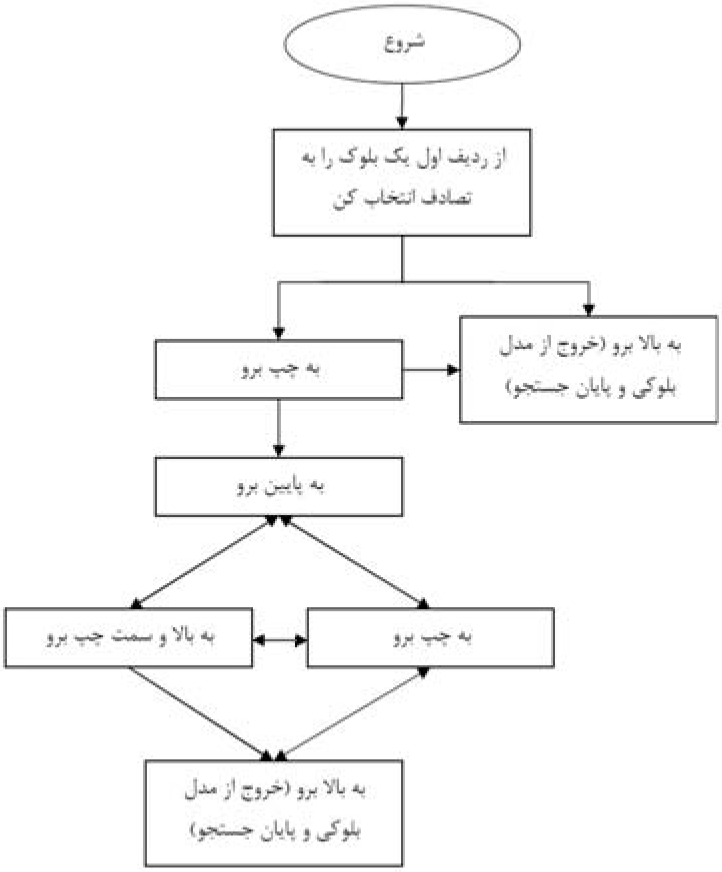
شکل ۴-۷: یک مورچه نمی‌تواند در دو مرحله متوالی به سمت پایین حرکت کند.

پس از هر حرکت به چپ، مورچه می‌تواند به سمت بالا، چپ یا پایین حرکت کند (شکل ۴-۸):



شکل ۴-۸: حرکت مورچه‌ها به سمت بالا، پایین و یا چپ، پس از هر حرکت به سمت چپ.

همانگونه که ذکر شد در برنامه توسعه یافته چهار نوع حرکت برای مورچه‌ها پیش‌بینی شده است. حرکت به سمت چپ، حرکت به سمت بالا، حرکت به سمت پایین و حرکت به سمت بالا و چپ (شکل ۴-۹ را ببینید). برای هر کدام از این حرکت‌ها، سیاست محاسبه ارزش حاصل شده تاکنون می‌تواند به گونه‌ی خاصی محاسبه شود. در مدلسازی انجام شده، ابتدا برنامه ارزش تجمعی بلوک‌های موجود بر روی هر بلوک بعلاوه‌ی ارزش آن بلوک را محاسبه می‌کند و به عنوان ارزش تجمعی بلوک‌ها نگهداری می‌کند. به این ترتیب یک مدل تجمعی با ابعاد برابر با مدل بلوکی اقتصادی حاصل می‌شود.



شکل ۴-۹: انواع حرکت مورچه‌ها

حرکت مورچه‌ها برای یافتن بهترین پاسخ بر روی این مدل بلوکی تجمعی صورت می‌گیرد. با توجه به شکل ۴-۹، بسته به حرکت یک مورچه حالت‌های مختلفی از محاسبه ارزش می‌توانند صورت گیرند:

(در تمامی موارد زیر i= شمارشگر ستون‌ها، j= شمارشگر سطرها و جهت جستجو از سمت راست به چپ می‌باشد):

حالت اول، مورچه به سمت چپ حرکت کرده است $\begin{cases} j = j \\ i = i+1 \end{cases}$ = حالت جاری

ارزش کل جستجو + ارزش بلوک در موقعیت جاری = ارزش کل جستجو

پس از هر حرکت به سمت چپ، مورچه می‌تواند یکی از حالت‌های ممکن زیر را انتخاب کند و یا یکی از حالت‌های زیر ممکن می‌باشد:

- به سمت چپ حرکت کند؛
- به پایین حرکت کند؛
- به بالا حرکت کند؛
- به بالا و سمت چپ حرکت کند؛
- در صورتیکه در سطر اول مدل بلوکی باشد با حرکت به سمت بالا از مدل خارج خواهد شد؛
- در صورتیکه در سطر اول و در آخرین ستون از مدل بلوکی باشد، باید امکان خروج مورچه از مدل را فراهم کرد؛
- درصورتیکه به محدوده مرز ممکن پیت رسیده باشد، باید با ادامه حرکات به سمت بالا و سمت چپ مورچه را از مدل بلوکی خارج ساخت؛
- درصورتیکه مورچه در آخرین ردیف از مدل بلوکی باشد، باید از حرکت مورچه به سمت پایین جلوگیری کرد.

حالت دوم: مورچه به سمت بالا حرکت کرده است. حالت جاری = $\begin{cases} j = j-1 \\ i = i \end{cases}$

ارزش کل جستجو = ارزش کل جستجو

(همانگونه که مشاهده می‌شود در این حالت ارزش کل جستجو تغییری نمی‌کند. دلیل این امر آن است که ارزش بلوک فعلی در مراحل قبلی به ارزش کل جستجو افزوده شده است).

پس از هر حرکت به سمت بالا، مورچه می‌تواند یکی از حالت‌های ممکن زیر را انتخاب کند و یا یکی از حالت‌های زیر ممکن است برای مورچه روی دهند:

- به سمت چپ حرکت کند؛
- اگر مورچه در سطر اول باشد، می‌تواند از مدل بلوکی خارج شده و جستجو را خاتمه دهد.

حالت سوم: مورچه به سمت پایین حرکت کرده است. $\begin{cases} j = j+1 \\ i = i \end{cases}$ = حالت جاری

ارزش کل جستجو + ارزش غیر تجمعی بلوک در موقعیت جاری = ارزش کل جستجو

پس از هر حرکت به سمت پایین، مورچه می‌تواند یکی از حالت‌های ممکن زیر را انتخاب کند و یا حالت‌های زیر ممکن است برای مورچه روی دهند:

- به سمت چپ حرکت کند؛
- به سمت بالا و چپ حرکت کند؛
- اگر به محدوده مرز ممکن پیت رسیده باشد، باید با ادامه حرکات به سمت بالا و سمت چپ مورچه را از مدل بلوکی خارج ساخت؛
- اگر در سمت چپ بلوک جاری، محدوده مرز ممکن پیت موجود باشد، باید با ادامه حرکات به سمت بالا و سمت چپ مورچه را از مدل بلوکی خارج ساخت.

حالت چهارم: مورچه به سمت بالا و چپ حرکت کرده است. $\begin{cases} j = j-1 \\ i = i-1 \end{cases}$ = حالت جاری

ارزش کل جستجو + ارزش بلوک در موقعیت جاری = ارزش کل جستجو

پس از هر حرکت به سمت بالا و چپ، مورچه می‌تواند یکی از حالت‌های ممکن زیر را انتخاب کند و یا حالت‌های زیر ممکن است برای مورچه روی دهند:

- به سمت چپ حرکت کند؛
- به سمت بالا حرکت کند؛
- باز هم به سمت بالا و سمت چپ برود؛
- به پایین حرکت کند؛
- اگر مورچه وارد سطر اول شد، با حرکت به سمت بالا از مدل بلوکی خارج خواهد شد؛

- اگر مورچه به ستون آخر رسید (و در نتیجه در سطر اول از مدل بلوکی بود) باید از مدل بلوکی خارج شود؛

اگر به انتهای محدوده مرز ممکن پیت رسیده باشد، باید با ادامه حرکات به سمت بالا و سمت چپ مورچه را از مدل بلوکی خارج ساخت.

۴-۲-۴ قوانین انتخاب مسیر و رد فرومون

در الگوریتم اصلی کولونی مورچگان میزان فرومون موجود روی مسیر به صورت $\tau = \frac{Q}{L}$ می‌باشد که در این رابطه τ مقدار فرومون، L کیفیت جواب حاصل شده (به معنی کم یا زیاد بودن فاصله غذا از لانه) و Q یک پارامتر ثابت می‌باشد که بر اساس آزمایش به دست می‌آید. در برنامه‌ی توسعه یافته، پارامتر میزان فرومون به صورت $\tau = Q \times L$ تعریف می‌شود که در این رابطه τ مقدار فرومون، Q یک پارامتر ثابت و L نیز (در صورت بیشتر بودن جواب یک گشت مورچه از جواب اولیه یا جواب مرحله قبل) برابر است با حاصل تفریق جواب مرحله قبل از جواب مرحله‌ی جاری. بنابراین L می‌تواند عددی بزرگتر از ۱ نیز باشد که در نتیجه پارامتر مقدار فرومون نیز می‌تواند در بعضی موارد بزرگتر از ۱ شود.

در الگوریتم اصلی کولونی مورچگان پارامتر نزدیکی به گره مجاور (در گراف) یا همان پارامتر η به صورت $\eta = \frac{1}{d}$ مطرح می‌شود که در این رابطه η پارامتر نزدیکی یا مجاورت بوده (و از نوع فاصله می‌باشد) و d فاصله‌ی موجود در روی یال‌های متصل به گرهی می‌باشد که مورچه قصد دارد در مرحله‌ی بعد از طریق آن یال به تور خود ادامه دهد. در برنامه‌ی توسعه یافته پارامتر نزدیکی به صورت تفریق ارزش بلوک‌های همسایه‌ی بلوک جاری (بسته به نوع حرکت، بلوک پایین، بلوک بالا، بلوک سمت چپ و یا بلوک بالا و سمت چپ) از ارزش بلوک جاری تعریف شده است. بر این اساس اگر حاصل این عبارت عددی مثبت باشد، مقدار η برابر ۰/۶ تعیین شده؛ اگر عددی منفی باشد، مقدار η برابر ۰/۲ تعیین شده و اگر برابر صفر شود مقدار η برابر ۰/۳ تعیین شده است.

براساس قانون گذر نسبی تصادفی در الگوریتم کولونی مورچگان، احتمال انتخاب یک مسیر (در اینجا مسیر ij) توسط مورچه به صورت رابطه ۴-۱ تعریف می‌شود:

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l}(\tau_{il})^{\alpha}(\eta_{il})^{\beta}} \quad \text{(رابطه ۴-۱)}$$

که در این رابطه با فرض اینکه مورچه بر روی گره i قرار دارد و قصد دارد به گره j برود، p_{ij} احتمال انتخاب مسیر ij توسط مورچه، τ_{ij} میزان فرومون موجود در روی مسیر ij؛ η_{ij} میزان دور یا نزدیک بودن گره بعدی به گره فعلی، τ_{il} میزان فرومون موجود بر روی مسیرهای احتمالی و ممکن که مسیر ij نیز یکی از آنهاست و η_{il} میزان دور یا نزدیک بودن گره‌های مجاور به گره فعلی است که گره j نیز جزئی از l می‌باشد. پارامترهای α و β نیز برای کم یا زیاد کردن تاثیر فرومون و توجه به ارزش بیشتر بوده و اعدادی به دلخواه و برحسب نوع مسئله می‌باشند؛ در این پایان نامه، این پارامترها برای یک مدل کوچک مورد تحلیل حساسیت قرار گرفته‌اند.

در مدلسازی انجام گرفته، احتمال انتخاب یک بلوک به صورت رابطه ۴-۲ تعریف شده است

$$p_{next_block} = \frac{(\tau_{next_block})^{\alpha}(\eta_{next_block})^{\beta}}{\sum (\tau_{neighbor_blocks})^{\alpha}(\eta_{neighbor_blocks})^{\beta}} \quad \text{(رابطه ۴-۲)}$$

که هر مورچه <u>براساس این احتمال</u> و <u>به تصادف</u> بلوک بعدی را انتخاب می‌کند.

۴-۳ مثالی از روش کار

مدل بلوکی شکل ۴-۱۰ را درنظر بگیرید.

	۱	۲	۳	۴	۵	۶	۷	۸
۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱
۲	-۲	-۲	-۲	+۳	-۲	-۲	-۲	-۲
۳	-۲	+۵	+۵	+۵	+۴	-۲	+۳	-۲
۴	-۳	-۳	+۵	+۳	+۳	+۲	-۳	-۳

شکل ۴-۱۰: یک مدل بلوکی برای حل توسط الگوریتم کولونی مورچگان

الگوریتم ابتدا این مدل بلوکی را به صورت تجمعی (همانند شکل ۴-۱۱) تبدیل می‌کند.

	۱	۲	۳	۴	۵	۶	۷	۸
۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱
۲	-۳	-۳	-۳	+۲	-۳	-۳	-۳	-۳
۳	-۵	+۲	+۲	+۷	+۱	-۵	۰	-۵
۴	-۸	-۱	+۷	+۱۰	+۴	-۳	-۳	-۸

شکل ۴-۱۱: حالت تجمعی مدل شکل ۴-۱۰ (محدوده ممکن معدن با خطوط پررنگ).

فرض کنید اولین مورچه از تکرار اول بلوک (۱،۲) را انتخاب می‌کند. در این صورت ارزش جستجو برابر با ارزش بلوک (۱-) و موقعیت این بلوک (۱،۲) ذخیره می‌شوند.

در مرحله بعد (با فرض آنکه مورچه به سمت بالا حرکت نکند و از مدل بلوکی خارج نشود) به دلیل آنکه مورچه یک واحد به سمت پایین حرکت کرده است، باید به سمت چپ حرکت کند. این مورد برای درنظر گرفتن محدودیت شیب ضروری می‌باشد. در این حالت موقعیت جاری، بلوک (۱،۳) می‌باشد. ارزش کل جستجو برابر با مجموع ارزش بلوک قبل و بلوک جاری و برابر با (۲-) می‌باشد. موقعیت بلوک جاری نیز در حافظه ذخیره خواهد شد.

برای مرحله بعد مورچه سه حرکت پیش رو دارد، حرکت به سمت بالا و خروج از مدل (در این حالت ارزش کل جستجو برابر با (۲-) خواهد بود)، حرکت به سمت چپ (که در این حالت ارزش کل جستجو برابر با (۳-) خواهد بود)، و حرکت به سمت پایین، که برای مثال این حرکت درنظر گرفته می‌شود. در این حالت ارزش کل جستجو برابر خواهد بود با (۴- = ۱+۲-۳-) و نتیجه جستجو همانند بخش هاشور خورده شکل ۴-۱۲ خواهد بود.

	۸	۷	۶	۵	۴	۳	۲	۱
۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱
۲	-۳	-۳	-۳	-۳	+۲	-۳	-۳	-۳
۳								

شکل ۴-۱۲: نتیجه جستجوی اولین مورچه تا سه مرحله از تکرار اول، با هاشور.

به دلیل حرکت رو به پایین مورچه در مرحله قبل، در مرحله جاری، مورچه فقط می‌تواند به سمت بالا، چپ و یا بالا و چپ حرکت کند. فرض کنید که مورچه به سمت چپ حرکت کرده است. در این حالت ارزش کل جستجو برابر خواهد شد با (۲- = ۴-۲+)، موقعیت جاری جستجو: بلوک (۲،۴)

در مرحله بعد مورچه می‌تواند هر چهار نوع حرکت گفته شده را داشته باشد. فرض کنید که مورچه به سمت پایین حرکت کند و به بلوک (۳،۴) برسد. در این حالت ارزش کل جستجو برابر است با (۳+ = ۲-۲-۷+). تا اینجا بلوک‌های جستجو شده به صورت شکل ۴-۱۳ می‌باشند:

	۸	۷	۶	۵	۴	۳	۲	۱
۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱
۲	-۳	-۳	-۳	-۳	+۲	-۳	-۳	-۳
۳	-۵	۰	-۵	+۱	+۷	+۲	+۲	-۵
۴	-۸	-۳	-۳	+۴	+۱۰	+۷	-۱	-۸

شکل ۴-۱۳: موقعیت جاری جستجو

به دلیل حرکت رو به پایین مورچه در مرحله پیش، در این مرحله مورچه فقط می‌تواند از بین دو حرکت به سمت چپ و به سمت بالا و چپ یکی را انتخاب کند. فرض کنید که مورچه به سمت بالا و چپ حرکت کند. موقعیت بلوک جاری برابر با (۲،۵) خواهد بود و ارزش کل جستجو برابر است با (۰ = ۳+۳-). حال مورچه می‌تواند از بین هر چهار حرکت، یکی را به تصادف انتخاب نماید. فرض کنید که باز هم به بالا و چپ برود در این حالت موقعیت جستجو در بلوک (۱،۶) قرار دارد و ارزش کل جستجو برابر است با (۱- = ۰+۱-). تا اینجا موقعیت بلوک‌های جستجو شده به صورت شکل ۴-۱۴ می‌باشد.

	۸	۷	۶	۵	۴	۳	۲	۱
۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱
۲	-۳	-۳	-۳	-۳	+۲	-۳	-۳	-۳
۳	-۵	۰	-۵	+۱	+۷	+۲	+۲	-۵
۴	-۸	-۳	-۳	+۴	+۱۰	+۷	-۱	-۸

شکل ۴-۱۴: بلوک‌های جستجو شده و وضعیت جستجو بر روی مدل بلوکی تجمعی.

به عنوان مرحله آخر، مورچه با حرکت به سمت بالا از مدل خارج خواهد شد. تا اینجا ارزش کل جستجو برابر با (۱-) بوده و شکل جستجو شده به صورت شکل ۴-۱۵ می‌باشد.

-۱				-۱
	-۲		-۵	
		+۵		

شکل ۴-۱۵: پیت جستجو شده توسط اولین مورچه از تکرار اول با ارزشی برابر با (۱-)

به دلیل آنکه در این جستجو محدوده‌ای با ارزش مثبت بدست نیامد، این مورچه نیز هیچ گونه فرومونی بر روی مسیرها برجای نخواهد گذاشت. مورچه دوم نیز جستجوی خود را شروع خواهد کرد، و در صورت یافتن جوابی مثبت، مسیرهای پاسخ یافت شده را فرومون گذاری می‌کند. این مسیر می‌تواند به صورت یک جدول به شکل ۴-۱۶ در رایانه ذخیره شود.

	۱	۲	۳	۴	۵	۶	۷	۸
۱	τ						τ	
۲		τ				τ		
۳			τ		τ			
۴				τ				

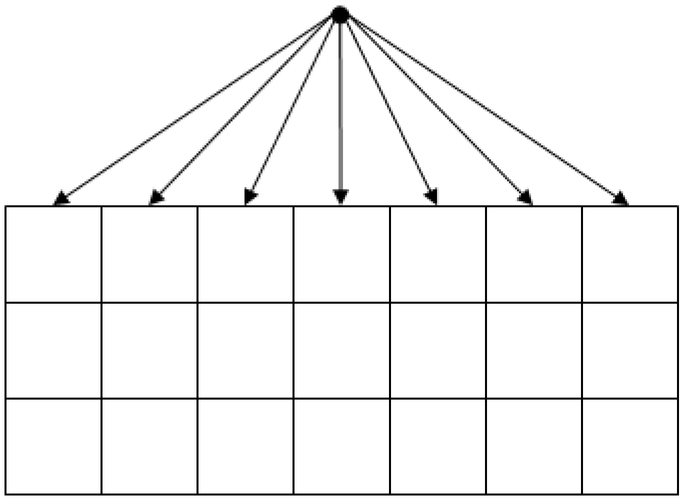
شکل ۴-۱۶: روش ذخیره فرومون گذاشته شده توسط مورچه یابنده جواب

مورچه‌های بعدی در هنگام انتخاب مسیر خود، این مقدار فرومون را براساس رابطه (۴-۱) در انتخاب خود تاثیر داده و به سمت آن بیشتر متمایل می‌شوند. پس از یافتن هر جواب بهتر از پاسخ مرحله قبل این جواب در حافظه ذخیره شده و به عنوان جواب فعلی با پاسخ‌هایی که بعداً به دست می‌آیند مقایسه خواهد شد. این کار آنقدر ادامه خواهد یافت تا آنکه یا تعداد تکرارها تمام شود و یا آنکه پس از تعداد مشخصی از تکرارها جواب بهبود نیابد. در این پایان نامه به منظور اتمام عملیات و بازگرداندن جواب یافت شده از شرط اول استفاده شده است.

۴-۴ یک نقص در الگوریتم کولونی مورچگان و راه برطرف کردن آن

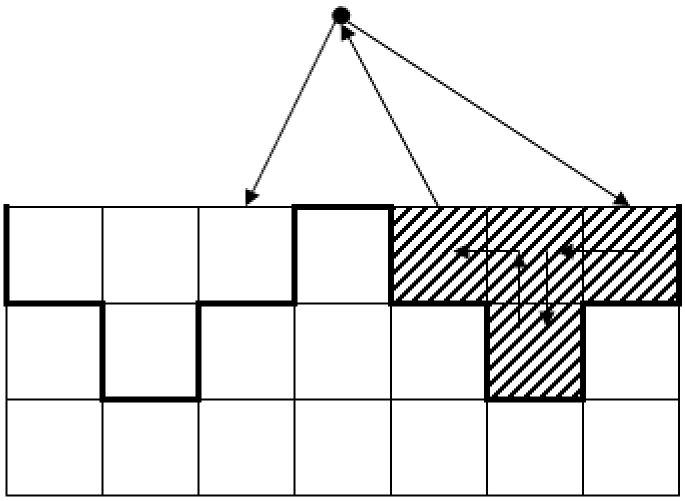
همانگونه که در بخش ۳-۵ ذکر شد، الگوریتم کولونی مورچگان بر روی گراف پیاده سازی می‌شود. در الگوریتم کولونی مورچگان، یک مورچه پیش از به پایان رساندن تور خود (کامل کردن مسیر) مجاز به برگشتن به لانه (محل شروع؛ در مسئله مورد نظر این پایان نامه، سطح زمین) نمی‌باشد.

این امر از این جهت اهمیت خواهد داشت که ممکن است مدل بهینه (پیت بهینه) از دو پیت در کنار هم تشکیل شده باشد. به شکل ۴-۱۷ توجه کنید:



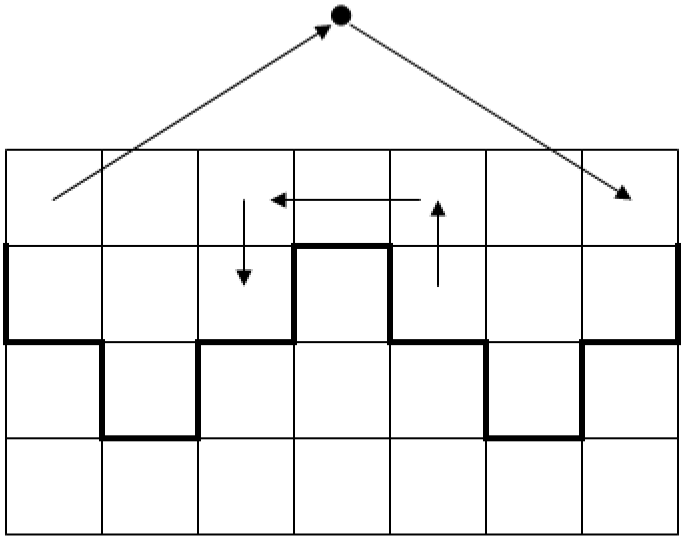
شکل ۴-۱۷: مورچه‌ها می‌توانند با شروع از مبدا یک ستون را به تصادف انتخاب کنند.

درصورتیکه مدل بهینه به صورت شکل ۴-۱۸ باشد، آنگاه الگوریتم اینگونه بیان خواهد شد که مورچه باید از سطح زمین مسیر خود را شروع کند و پس از طی ۴ بلوک اول که با هاشور مشخص شده است به نقطه شروع بازگردد و دوباره مسیر را ادامه دهد که چنین موردی در الگوریتم کولونی مورچگان ممکن نمی‌باشد (در این الگوریتم بازگشت به لانه در نیمه مسیر ممنوع می‌باشد).



شکل ۴-۱۸: عدم توانایی جستجوی دو پیت کنار هم توسط الگوریتم کولونی مورچگان.

برای حل این مشکل در این پایان نامه، در ابتدا یک سطر با ارزش صفر به بالای اولین طبقه اضافه خواهد شد که در نتیجه مورچه می‌تواند مسیر را درون این سطر و خارج از مدل بلوک اصلی و نیز بدون بازگشت به مبداء (لانه؛ نقطه شروع، سطح زمین) ادامه دهد (همانند شکل ۴-۱۹).



شکل ۴-۱۹: حل مشکل شکل ۴-۱۸ با اضافه شدن یک سطر با بلوک‌های دارای ارزش صفر.

فصل پنجم

برنامه نویسی و نتایج

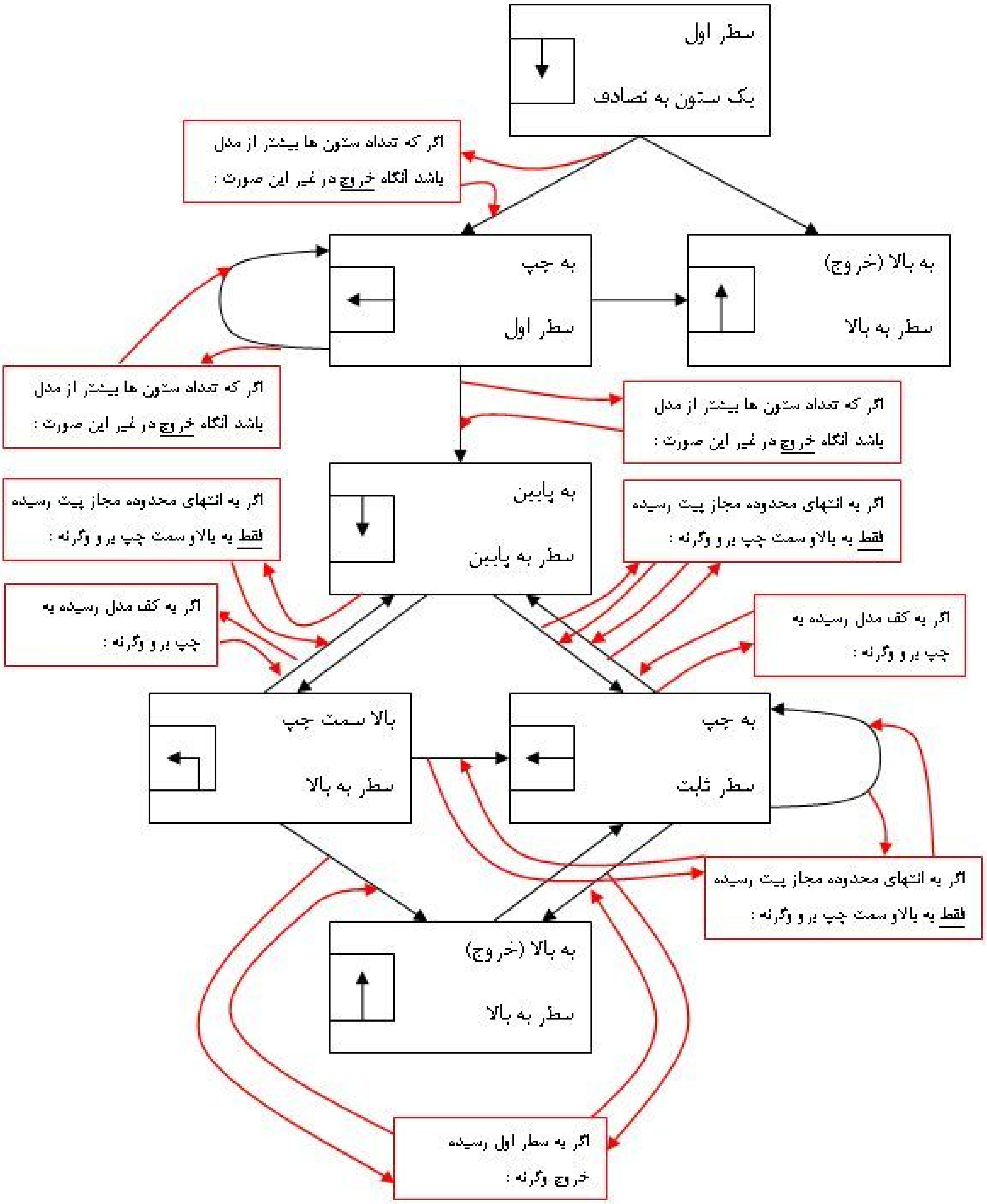
فصل ۵

۵-۱ مقدمه

در این فصل اصول برنامه رایانه‌ای توسعه یافته تشریح خواهند شد. قطعات کد نوشته شده به منظور استفاده‌های بعدی به صورت ماژولی و به زبان ساده بیان شده‌اند. همچنین برای درک بهتر چگونگی کارکرد بعضی از پارامترها از چند مدل کوچک استفاده شده است.

۵-۲ چگونگی کار برنامه توسعه یافته

برای برنامه توسعه یافته از روند ارائه شده در شکل ۵-۱ استفاده شده است. در این فلوچارت خطوط مستقیم بیانگر مراحل اصلی برنامه توسعه یافته بوده و خطوط دارای انحنا نیز محدودیت‌های جلوگیری کننده از انحراف مورچه‌ها را نشان می‌دهند.



شکل ۵-۱: روندی که برنامه براساس آن کار کرده و توسعه یافته است.

۵-۳ برنامه رایانه‌ای Ant Pit

۵-۳-۱ درباره نرم افزار

در این پایان نامه هدف تعیین محدوده نهایی معادن روباز به کمک الگوریتم کولونی مورچگان می‌باشد. به دلیل پیچیده بودن الگوریتم کولونی مورچگان، تعدد پارامترها، ویژگی غیرقطعی بودن مدل‌های بلوکی و خصلت تصادفی بودن رفتار مورچه‌ها، نرم افزاری به نام Pit Ant در محیط برنامه نویسی ویژوال بیسیک طراحی شده است. این برنامه نرم افزاری قادر است مدل‌های بلوکی اقتصادی دو بعدی را توسط الگوریتم کولونی مورچگان حل کرده و بهترین محدوده نهایی را (با توجه به محدودیت‌ها و خصوصیات یک الگوریتم ابتکاری) پیدا نماید. به منظور توسعه آتی این نرم افزار، از روش برنامه نویسی ماژولی استفاده شده است.

نرم افزار Ant Pit در محیط سیستم عامل ویندوز اجرا می‌شود. با اجرا شدن این نرم افزار، پنجره موجود در شکل ۵-۲ مشاهده خواهد شد. شرح تنظیمات موجود در این پنجره (شکل ۵-۲) در قسمت بعدی ذکر خواهند شد.

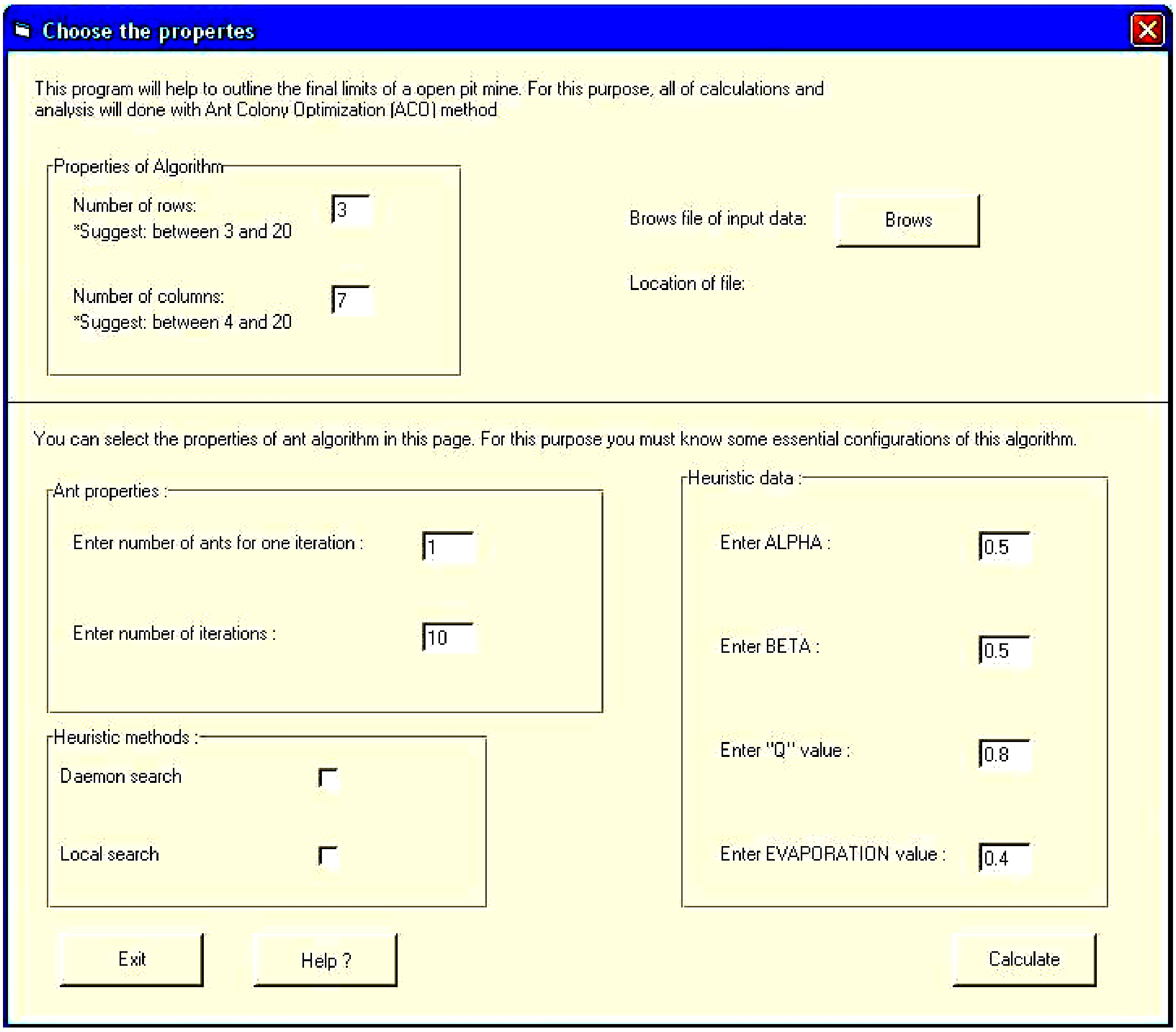
۵-۳-۲ شرح برنامه توسعه یافته

در هر قسمت از این برنامه و جلوی تنظیمات قابل دسترس آن توضیحات مختصری راجع به آن قسمت ارائه شده است.

در بالا و سمت راست این نجره دکمه Brows برای جستجو و تعیین فایل ورودی به برنامه وجود دارد. با کلیک کردن روی این دکمه، پنجره استاندارد Open متعلق به سیستم عامل ویندوز باز خواهد شد که اجازه انتخاب فایل‌های دارای پسوند (txt.) را خواهد داد. با انتخاب یک فایل، مسیر فایل ورودی در زیر دکمه نمایش داده خواهد شد.

به منظور تعیین تعداد سطرها و ستون‌های یک مدل بلوکی و یا به عبارت دیگر بخشی از مدل که کاربر قصد جستجوی آن را دارد، قابی به نام Properties of algorithm وجود دارد. در این قاب دو

گزینه قابل تنظیم وجود دارند که در بخش Number of rows کاربر تعداد سطرها و در بخش Number of columns تعداد ستون‌ها را وارد می‌کند.



شکل ۵-۲: نمایی از حالت کلی برنامه توسعه یافته

در قاب دیگری به نام Ant properties، کاربر قادر خواهد بود تا تعداد مورچه‌ها و تعداد تکرارها را به دلخواه تعیین نماید (مقادیر پیشنهادی در محل ورود داده‌ها با کلیک کردن بر روی آنها قابل تغییر خواهند بود).

به منظور دستیابی به بهترین جواب، توصیه می‌شود که برنامه چندین بار اجرا شده و مقادیر مختلفی برای تعداد مورچه‌ها و تعداد تکرارها امتحان شوند.

در پایین قاب ذکر شده، قابی به نام Heuristic methods وجود دارد که دارای دو گزینه‌ی چک شونده به نام‌های Daemon search و Local search می‌باشد. این بخش مربوط به بکارگیری روش‌های ابتکاری جستجوی اهریمنی و جستجوی محلی می‌باشد که مربوط به بخش‌های غیر اجباری (اختیاری) الگوریتم کولونی مورچگان بوده و در این پایان نامه هنوز کامل نشده‌اند.

در قسمت پایین و سمت راست پنجره شکل ۵-۲ قابی به نام Heuristic data وجود دارد که می‌توان از این بخش داده‌ها و تنظیمات مربوط به الگوریتم کولونی مورچگان را انجام داد. در این قاب بخش‌های مربوط به وارد کردن داده‌های مرتبط با پارامتر α (توان مربوط به پارامتر فرومون در رابطه قانون گذر تصادفی)، پارامتر β (توان مربوط به پارامتر مجاورت یا نزدیکی در رابطه قانون گذر تصادفی)، پارامتر Q (مربوط به محاسبه پارامتر مجاورت یا نزدیکی) و مقدار پارامتر تبخیر فرومون وجود دارند که در این پایان نامه رابطه آنها با مثال‌های متعدد نشان داده شده است.

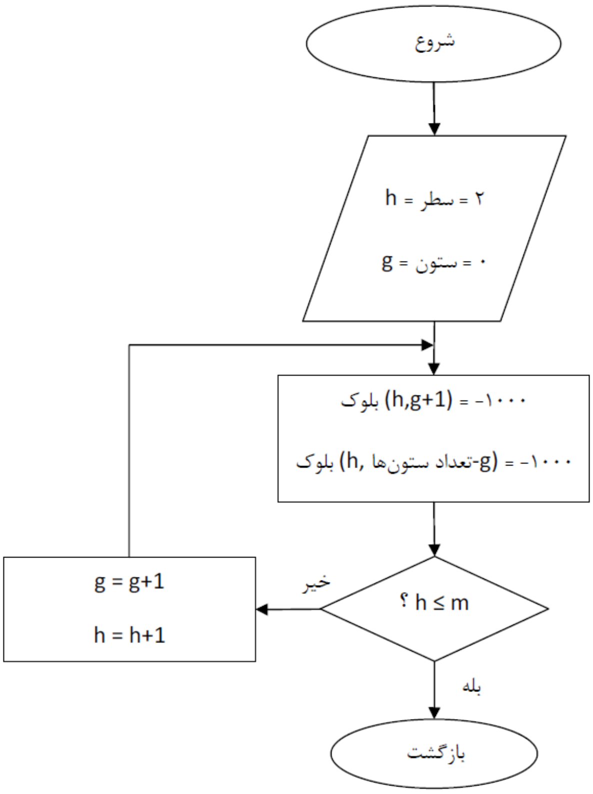
در قسمت پایین و سمت راست، دکمه‌ای به نام Calculate وجود دارد که با کلیک بر روی آن برنامه اجرا شده و در صورت موجود بودن جواب، جواب نهایی یافت شده در یک فایل متنی هم‌نام با نام فایل ورودی بر روی درایو دیسک سخت رایانه ذخیره خواهد شد. شکل نهایی جواب به صورت گرافیکی نمی‌باشد، بلکه بلوک‌هایی را که مورچه از آنها عبور کرده است نشان خواهد داد؛ بطوریکه عدد یک نشان دهنده عبور مورچه از آن مسیر و عدد صفر بیانگر عدم عبور مورچه از آن مسیر می‌باشد.

در پایین و سمت چپ پنجره شکل ۵-۲ نیز دو دکمه به نام‌های Help و Exit وجود دارند که اولی پنجره مربوط به کمک سستم عامل ویندوز را باز کرده و با کلیک بر روی دومی برنامه خاتمه می‌یابد.

۵-۴ روش برنامه نویسی و الزامات

در برنامه نوشته شده از روش ماژولی استفاده شده است به این معنی که برای هر عملیات یک تکه کد جدا نوشته شده است تا فهم، نگهداری و توسعه برنامه راحتتر صورت گیرد.

ابتدا مرزهای بزرگترین پیت ممکن به صورت شکل ۵-۳ تعیین می‌شود (m برابر با تعداد ردیف‌ها می‌باشد):



شکل ۵-۳: مرحله آماده سازی مدل بلوکی به منظور جلوگیری از خروج مورچه‌ها از مدل‌ها

با انجام این کار در هر مرحله از اجرای این فلوچارت، بزرگترین محدوده ممکن برای مدل بلوکی مشخص می‌شود. بطور مثال درصورتیکه یک مدل با سه سطر و هفت ستون داشته باشیم، در مرحله اول ابتدا بلوک‌های (۲،۱) و (۲،۷) و در مرحله بعد بلوک‌های (۳،۲) و (۳،۶) دارای ارزش ۱۰۰۰- می‌شوند.

به منظور تعیین بهترین جواب اولیه از کد زیر استفاده شده است:

بهترین جواب تاکنون = ۰

برای سطر اول تا سوم

برای ستون اول تا هفتم

اگر ارزش بلوک بزرگتر از صفر باشد آنگاه ارزش بلوک را مساوی بهترین جواب تاکنون قرار بده و خاتمه عملیات؛ وگرنه به جستجو ادامه بده

قسمت ۱

به منظور انجام اولین جستجو، الگوریتم مراحل زیر را انجام می‌دهد:

یک عدد تصادفی بین یک و تعداد ستون‌ها (عدد صحیح) تعیین کن و آن را در شماره ستون قرار بده

ارزش کل جستجو را برابر ارزش بلوک واقع در ردیف اول و ستون (عدد تصادفی) قرار بده

اگر موقعیت جاری در ستون هفتم بود خارج شو

به قسمت ۲ برو

قسمت ۲

یک عدد تصادفی بین یک و دو (عدد صحیح) تعیین کن

اگر عدد تصادفی برابر یک باشد به سمت چپ برو

اگر عدد تصادفی برابر دو باشد به سمت بالا برو (پایان)

ماژول حرکت مورچه به سمت چپ

اگر جستجو در ستون هفتم (آخر) قرار دارد، جستجو را خاتمه بده و مقادیر را برگردان

وگرنه به موقعیت افقی جستجو یک واحد اضافه کن

اگر موقعیت جاری در ستون هفتم (آخر) قرار دارد، جستجو را خاتمه بده و مقادیر را برگردان

وگرنه ارزش کل جستجو را حساب کن

یک عدد تصادفی بین ۱ و ۳ تعیین کن (عدد صحیح)

اگر عدد تصادفی برابر ۱ باشد آنگاه

اگر موقعیت جاری در ستون هفتم (آخر) قرار دارد، جستجو را خاتمه بده و مقادیر را برگردان

وگرنه اگر بلوک سمت چپ دارای ارزش ۱۰۰۰- است به ماژول بالا و سمت چپ اضطراری برو

وگرنه به سمت چپ برو

وگرنه اگر عدد تصادفی برابر ۲ باشد، آنگاه به سمت بالا برو

وگرنه اگر عدد تصادفی برابر ۳ باشد، آنگاه

اگر موقعیت جاری در ستون هفتم (آخر) قرار دارد، جستجو را خاتمه بده و مقادیر را برگردان

وگرنه اگر بلوک سمت چپ یا پایین دارای ارزش ۱۰۰۰- است، به بالا برو و مقادیر را برگردان

وگرنه به سمت پایین برو

ماژول حرکت مورچه به سمت بالا

از موقعیت قائم جستجو یک واحد کم کن

اگر سطر جاری، سطر صفر بود (خارج از مدل بلوکی) جستجو را خاتمه بده و مقادیر را برگردان

وگرنه ارزش کل جستجو را حساب کن

به چپ برو

ماژول حرکت مورچه به سمت پایین

اگر سطر آخر بود به سمت چپ برو

اگرنه به موقعیت جاری قائم جستجو یک واحد اضافه کن

ارزش کل جستجو را حساب کن

اگر ستون هفتم بود به ماژول بالا و چپ اضطراری برو

وگرنه اگر ارزش بلوک سمت چپ برابر ۱۰۰۰- بود، به ماژول بالا و چپ اضطراری برو

وگرنه یک عدد تصادفی بین ۱ و ۲ (عدد صحیح) انتخاب کن

اگر عدد تصادفی برابر ۱ بود آنگاه

اگر ارزش بلوک سمت چپ برابر ۱۰۰۰- بود به قسمت بالا و چپ اضطراری برو

وگرنه به سمت چپ برو

وگرنه اگر عدد تصادفی برابر ۲ بود، آنگاه به بالا و سمت چپ برو

ماژول حرکت مورچه به سمت بالا و چپ

از موقعیت جاری قائم جستجو یک واحد کم کن وبه موقعیت جاری افقی جستجو یک واحد اضافه کن

ارزش کل جستجو را حساب کن

اگر موقعیت جاری در سطر صفر بود آنگاه به جستجو خاتمه بده و مقادیر را برگردان

وگرنه اگر موقعیت جاری در سطر آخر و یا یک ستون اضافه‌تر از سطر آخر بود به جستجو خاتمه بده و مقادیر را برگردان

در غیر این صورت

یک عدد تصادفی بین ۱ و ۳ (عدد صحیح) تعیین کن

اگر عدد صحیح برابر ۱ بود آنگاه به سمت چپ برو

وگرنه اگر عدد تصادفی برابر ۲ بود به سمت بالا برو

وگرنه اگر عدد تصادفی برابر ۳ بود آنگاه

اگر جستجو در ستون آخر قرار داشت آنگاه به جستجو خاتمه بده و مقادیر را برگردان

درغیر این صورت به سمت پایین برو

ماژول حرکت مورچه به سمت بالا و چپ (اضطراری)

از موقعیت جاری قائم جستجو یک واحد کم کن وبه موقعیت جاری افقی جستجو یک واحد اضافه کن

اگر موقعیت جاری در سطر صفر بود به جستجو خاتمه بده و مقادیر را برگردان

در غیر این صورت ارزش کل جستجو را حساب کن

دوباره به سمت بالا و چپ اضطراری برو

قسمت ۳

حال الگوریتم باید به تعداد مورچگان این عمل را تکرار کند. یک دور از حرکت کل مورچگان به عنوان اولین تکرار از اجرای الگوریتم خواهد بود.

برای تکرار دوم به بعد

ابتدا از سطر اول یک ستون به تصادف انتخاب می‌شود و برطبق قوانین موجود و مشابه قسمت ۲، جستجو ادامه می‌یابد؛ با این تفاوت که برای انتخاب حرکت بعدی به جای استفاده از اعداد تصادفی از رابطه ۵-۱ استفاده خواهد شد:

$$P = \frac{\tau^{\alpha}.\eta^{\beta}}{\sum(\tau^{\alpha}.\eta^{\beta})} \quad \text{(رابطه ۵-۱)}$$

مقادیر η (اطلاعات ابتکاری، مجاورت، یا در این پایان نامه ارزش بیشتر) متناسب با ارزش بلوک‌های سمت چپ، بالا، پایین و بالا و سمت چپ از بلوک جاری خواهد بود و به روش زیر محاسبه می‌شود.

درصورتیکه بلوک مجاور دارای ارزش مثبت بود، مقادیر مجاورت برابر با ۰/۶، اگر بلوک مجاور دارای ارزش منفی بود، مقادیر مجاورت دارای ارزش ۰/۲ و اگر که بلوک مجاور دارای ارزش صفر بود مقدار مجاورت برابر با ۰/۳ منظور خواهد شد. این مقدار پیش از اجرای دومین دور از تکرار حرکت مورچه‌ها به هر بلوک اختصاص داده می‌شود.

همچنین مقدار τ یا پارامتر فرومون موجود بر روی مسیر به صورت رابطه ۵-۲ قابل محاسبه خواهد بود:

$$\tau = Q \times L \qquad \text{(رابطه ۵-۲)}$$

در این برنامه مقدار Q برابر با ۰/۸ قرار داده شده است که البته از صفحه اصلی برنامه قابل تغییر می‌باشد. مقدار L نیز برابر با حاصل تفریق بهترین مقدار جستجو در مرحله قبل از جواب مرحله جاری در یک تکرار می‌باشد. این مقدار نیز به مسیر مورچه‌ای که بهترین جواب را یافته، تعلق خواهد گرفت. برای اینکه رابطه‌ی انتخاب احتمالاتی بلوک‌های مجاور بعدی با ضرب صفر در صورت و مخرج مواجه نشوند به بقیه بلوک‌ها نیز مقدار فرومون ناچیز ۰/۰۵ تعلق خواهد گرفت.

نتایج اجرای برنامه بر روی مدل ارائه شده در شکل ۵-۴، به همراه تنظیمات نرم‌افزار، در جداول ۵-۱ تا ۵-۱۲ ارائه شده است. در این جداول برای هر تکرار، دو ستون وجود دارد که ستون سمت چپ نشان دهنده شماره مورچه و ستون دیگری بیانگر جواب بدست آمده می‌باشد. این مدل دارای ارزش بهینه برابر با (۴+) و پیت بهینه به صورت خطوط پررنگ می‌باشد که در شکل ۵-۴ نشان داده شده است.

-۱	+۱	-۱	-۱	-۱	-۱	-۱
-۲	-۲	+۳	-۲	-۲	+۲	-۲
-۳	-۳	-۳	-۳	+۷	-۳	-۳

شکل ۵-۴: مدل بلوکی مورد استفاده برای ارزیابی تعدادی از پارامترهای الگوریتم

جدول ۵-۱: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۲، β=۰/۲، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
+۲	۹۱	+۲	۸۱	-۱	۷۱	-۱	۶۱	-۳	۵۱	-۲	۴۱	-۱	۳۱	-۶	۲۱	۰	۱۱	-۱	۱
-۱	۹۲	-۲	۸۲	+۳	۷۲	-۵	۶۲	۰	۵۲	-۶	۴۲	-۵	۳۲	-۱	۲۲	-۷	۱۲	-۲	۲
-۲	۹۳	-۱	۸۳	-۴	۷۳	+۲	۶۳	-۴	۵۳	-۳	۴۳	-۱	۳۳	+۱	۲۳	-۱	۱۳	-۱	۳
+۲	۹۴	-۲	۸۴	+۲	۷۴	-۱	۶۴	-۷	۵۴	۰	۴۴	-۳	۳۴	-۳	۲۴	-۲	۱۴	-۳	۴
+۴	۹۵	-۲	۸۵	-۲	۷۵	۰	۶۵	-۴	۵۵	-۱	۴۵	-۱	۳۵	۰	۲۵	+۱	۱۵	۰	۵
+۴	۹۶	+۱	۸۶	-۳	۷۶	+۱	۶۶	-۱	۵۶	-۲	۴۶	-۴	۳۶	-۱	۲۶	-۴	۱۶	-۳	۶
۰	۹۷	+۳	۸۷	-۱	۷۷	-۳	۶۷	-۴	۵۷	-۲	۴۷	-۵	۳۷	-۱	۲۷	-۳	۱۷	-۱	۷
+۴	۹۸	+۲	۸۸	+۲	۷۸	-۲	۶۸	+۱	۵۸	-۱	۴۸	-۱	۳۸	-۲	۲۸	-۲	۱۸	-۲	۸
-۳	۹۹	+۱	۸۹	-۱	۷۹	-۵	۶۹	-۵	۵۹	-۳	۴۹	-۲	۳۹	۰	۲۹	-۲	۱۹	-۱	۹
+۳	۱۰۰	+۲	۹۰	-۲	۸۰	-۶	۷۰	-۲	۶۰	-۴	۵۰	-۸	۴۰	-۱	۳۰	-۲	۲۰	-۱	۱۰

پارامترهای α و β اندک (و شاید هم مساوی) متعلق به الگوریتم، دیرهنگام ولی به یک جواب مثبت همگرا خواهند شد.

جدول ۵-۲: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۴، β=۰/۲، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۲	۹۱	+۲	۸۱	+۲	۷۱	-۲	۶۱	-۱	۵۱	-۱	۴۱	+۳	۳۱	-۱	۲۱	-۱	۱۱	-۱	۱
-۳	۹۲	+۱	۸۲	-۳	۷۲	-۳	۶۲	+۱	۵۲	-۵	۴۲	۰	۳۲	-۲	۲۲	-۲	۱۲	-۱	۲
+۴	۹۳	-۱	۸۳	۰	۷۳	-۷	۶۳	-۳	۵۳	-۴	۴۳	-۲	۳۳	+۱	۲۳	-۳	۱۳	۰	۳
-۲	۹۴	+۲	۸۴	-۵	۷۴	-۵	۶۴	-۳	۵۴	-۱	۴۴	-۱	۳۴	-۵	۲۴	-۳	۱۴	-۱	۴
-۱	۹۵	-۲	۸۵	+۲	۷۵	-۱	۶۵	۰	۵۵	+۲	۴۵	-۲	۳۵	۰	۲۵	-۱	۱۵	+۱	۵
+۳	۹۶	+۲	۸۶	-۱	۷۶	-۶	۶۶	-۳	۵۶	-۲	۴۶	-۲	۳۶	-۴	۲۶	-۲	۱۶	-۱	۶
-۲	۹۷	۰	۸۷	+۲	۷۷	-۲	۶۷	۰	۵۷	-۲	۴۷	-۳	۳۷	-۱	۲۷	+۱	۱۷	-۴	۷
+۳	۹۸	-۱	۸۸	-۱	۷۸	-۳	۶۸	-۳	۵۸	-۷	۴۸	-۴	۳۸	-۴	۲۸	۰	۱۸	-۳	۸
+۴	۹۹	+۳	۸۹	+۳	۷۹	-۳	۶۹	-۲	۵۹	-۲	۴۹	+۲	۳۹	-۵	۲۹	-۴	۱۹	-۲	۹
+۴	۱۰۰	-۳	۹۰	-۱	۸۰	-۲	۷۰	-۱	۶۰	-۱	۵۰	-۳	۴۰	-۶	۳۰	۰	۲۰	-۵	۱۰

جدول ۵-۳: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۶، β=۰/۲، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۴	۹۱	-۳	۸۱	-۱	۷۱	-۵	۶۱	-۱	۵۱	-۲	۴۱	-۱	۳۱	-۲	۲۱	-۲	۱۱	-۱	۱
-۱	۹۲	-۲	۸۲	-۱	۷۲	-۱	۶۲	+۳	۵۲	-۱	۴۲	-۰	۳۲	-۱	۲۲	-۲	۱۲	-۱	۲
+۴	۹۳	-۲	۸۳	-۳	۷۳	-۸	۶۳	+۲	۵۳	-۴	۴۳	-۲	۳۳	-۳	۲۳	-۳	۱۳	-۲	۳
-۱	۹۴	-۲	۸۴	۰	۷۴	+۲	۶۴	-۲	۵۴	-۱	۴۴	-۱	۳۴	-۴	۲۴	-۷	۱۴	-۲	۴
+۳	۹۵	+۴	۸۵	+۱	۷۵	+۳	۶۵	-۱	۵۵	-۳	۴۵	+۳	۳۵	-۳	۲۵	۰	۱۵	-۲	۵
+۳	۹۶	-۳	۸۶	+۱	۷۶	-۳	۶۶	-۲	۵۶	۰	۴۶	-۲	۳۶	-۱	۲۶	-۱	۱۶	-۳	۶
۰	۹۷	+۴	۸۷	-۱	۷۷	-۲	۶۷	-۱	۵۷	-۷	۴۷	-۴	۳۷	-۲	۲۷	-۲	۱۷	-۲	۷
+۲	۹۸	-۲	۸۸	+۲	۷۸	-۳	۶۸	-۲	۵۸	-۲	۴۸	-۱	۳۸	-۲	۲۸	-۳	۱۸	-۲	۸
+۲	۹۹	+۴	۸۹	-۲	۷۹	-۱	۶۹	-۳	۵۹	-۱	۴۹	-۲	۳۹	-۱	۲۹	-۲	۱۹	-۱	۹
+۱	۱۰۰	-۲	۹۰	-۱	۸۰	۰	۷۰	۰	۶۰	-۱	۵۰	-۳	۴۰	-۲	۳۰	۰	۲۰	-۲	۱۰

پس از یافتن یک مسیر بهینه، الگوریتم دوباره مسیر بهینه را گم می‌کند.

جدول ۵-۴: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۸، β=۰/۲، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۳	۹۱	-۲	۸۱	۰	۷۱	+۳	۶۱	-۳	۵۱	+۲	۴۱	-۱	۳۱	-۱	۲۱	۰	۱۱	-۱	۱
-۲	۹۲	+۳	۸۲	-۱	۷۲	-۲	۶۲	+۱	۵۲	-۱	۴۲	-۳	۳۲	-۱	۲۲	+۱	۱۲	-۲	۲
+۴	۹۳	+۴	۸۳	-۱	۷۳	-۲	۶۳	-۴	۵۳	۰	۴۳	-۴	۳۳	-۱	۲۳	-۲	۱۳	-۱	۳
-۳	۹۴	-۲	۸۴	۰	۷۴	-۱	۶۴	-۳	۵۴	-۳	۴۴	+۱	۳۴	-۵	۲۴	-۳	۱۴	-۳	۴
-۳	۹۵	-۵	۸۵	-۱	۷۵	+۱	۶۵	+۲	۵۵	-۴	۴۵	-۲	۳۵	-۶	۲۵	-۲	۱۵	-۱	۵
-۲	۹۶	+۴	۸۶	-۱	۷۶	-۲	۶۶	-۱	۵۶	۰	۴۶	-۷	۳۶	-۱	۲۶	۰	۱۶	-۱	۶
-۲	۹۷	+۴	۸۷	-۵	۷۷	-۱	۶۷	-۲	۵۷	-۲	۴۷	+۱	۳۷	-۳	۲۷	-۳	۱۷	+۱	۷
+۱	۹۸	۰	۸۸	-۳	۷۸	-۲	۶۸	-۱	۵۸	+۱	۴۸	-۱	۳۸	-۱	۲۸	+۲	۱۸	-۲	۸
-۱	۹۹	-۳	۸۹	-۲	۷۹	-۲	۶۹	-۳	۵۹	-۱	۴۹	-۲	۳۹	-۲	۲۹	-۲	۱۹	-۳	۹
۰	۱۰۰	+۴	۹۰	-۲	۸۰	-۳	۷۰	+۱	۶۰	-۲	۵۰	۰	۴۰	-۱	۳۰	-۳	۲۰	-۲	۱۰

پس از یافتن یک مسیر بهینه، الگوریتم دوباره مسیر بهینه را گم می‌کند.

جدول ۵-۵: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۲، β=۰/۴، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۲	۹۱	+۲	۸۱	+۴	۷۱	-۳	۶۱	-۱	۵۱	-۱	۴۱	-۳	۳۱	-۲	۲۱	-۱	۱۱	-۲	۱
-۱	۹۲	-۲	۸۲	-۷	۷۲	+۲	۶۲	-۲	۵۲	+۳	۴۲	-۳	۳۲	-۴	۲۲	-۲	۱۲	-۱	۲
-۱	۹۳	-۲	۸۳	-۶	۷۳	-۵	۶۳	۰	۵۳	۰	۴۳	-۱	۳۳	-۱	۲۳	+۱	۱۳	+۱	۳
+۲	۹۴	-۱	۸۴	۰	۷۴	-۱	۶۴	-۱	۵۴	-۲	۴۴	-۱	۳۴	+۱	۲۴	-۲	۱۴	-۱	۴
+۳	۹۵	+۴	۸۵	-۲	۷۵	-۲	۶۵	+۳	۵۵	-۱	۴۵	-۳	۳۵	-۲	۲۵	۰	۱۵	-۱	۵
+۳	۹۶	-۲	۸۶	-۳	۷۶	-۱	۶۶	-۲	۵۶	-۳	۴۶	-۲	۳۶	-۳	۲۶	+۱	۱۶	-۲	۶
-۱	۹۷	-۱	۸۷	-۲	۷۷	-۲	۶۷	-۲	۵۷	-۲	۴۷	-۱	۳۷	-۱	۲۷	-۱	۱۷	-۱	۷
+۲	۹۸	+۱	۸۸	۰	۷۸	-۱	۶۸	-۲	۵۸	+۱	۴۸	+۱	۳۸	۰	۲۸	-۱	۱۸	-۱	۸
۰	۹۹	-۱	۸۹	+۳	۷۹	-۲	۶۹	+۱	۵۹	-۱	۴۹	-۱	۳۹	-۱	۲۹	-۲	۱۹	-۲	۹
-۲	۱۰۰	-۲	۹۰	-۲	۸۰	-۲	۷۰	-۱	۶۰	-۱	۵۰	-۱	۴۰	-۱	۳۰	-۱	۲۰	-۱	۱۰

جدول ۵-۶: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۲، β=۰/۶، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۱	۹۱	-۱	۸۱	-۱	۷۱	-۲	۶۱	۰	۵۱	-۱	۴۱	-۱	۳۱	-۳	۲۱	-۱	۱۱	+۱	۱
۰	۹۲	-۱	۸۲	+۱	۷۲	-۱	۶۲	-۱	۵۲	-۳	۴۲	+۳	۳۲	+۲	۲۲	-۲	۱۲	-۲	۲
+۲	۹۳	۰	۸۳	-۴	۷۳	-۳	۶۳	-۲	۵۳	-۱	۴۳	-۱	۳۳	-۲	۲۳	-۲	۱۳	-۱	۳
-۱	۹۴	۰	۸۴	-۱	۷۴	-۱	۶۴	-۱	۵۴	-۲	۴۴	+۴	۳۴	-۱	۲۴	-۱	۱۴	-۶	۴
-۱	۹۵	-۱	۸۵	+۴	۷۵	-۳	۶۵	+۲	۵۵	-۱	۴۵	۰	۳۵	-۳	۲۵	۰	۱۵	-۵	۵
-۳	۹۶	-۲	۸۶	-۲	۷۶	-۱	۶۶	-۳	۵۶	-۲	۴۶	+۴	۳۶	-۱	۲۶	-۳	۱۶	-۲	۶
-۳	۹۷	+۳	۸۷	+۴	۷۷	۰	۶۷	-۱	۵۷	-۱	۴۷	-۱	۳۷	-۲	۲۷	-۱	۱۷	-۱	۷
-۱	۹۸	-۲	۸۸	-۱	۷۸	-۱	۶۸	-۴	۵۸	-۱	۴۸	-۱	۳۸	+۲	۲۸	-۷	۱۸	-۳	۸
+۱	۹۹	-۱	۸۹	+۲	۷۹	-۲	۶۹	-۱	۵۹	+۱	۴۹	-۳	۳۹	۰	۲۹	-۱	۱۹	-۱	۹
-۲	۱۰۰	-۲	۹۰	۰	۸۰	-۱	۷۰	-۵	۶۰	-۱	۵۰	-۱	۴۰	-۱	۳۰	-۳	۲۰	-۲	۱۰

الگوریتم دو بار همگرا شده و دوباره واگرا شده است.

جدول ۵-۷: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۲، β=۰/۸، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۲	۹۱	+۳	۸۱	-۲	۷۱	+۲	۶۱	-۱	۵۱	۰	۴۱	-۳	۳۱	-۶	۲۱	۰	۱۱	-۱	۱
-۱	۹۲	-۲	۸۲	۰	۷۲	-۱	۶۲	+۱	۵۲	-۱	۴۲	-۱	۳۲	-۳	۲۲	-۱	۱۲	-۱	۲
+۳	۹۳	-۱	۸۳	-۲	۷۳	-۱	۶۳	-۱	۵۳	+۱	۴۳	-۳	۳۳	-۶	۲۳	+۱	۱۳	+۱	۳
-۳	۹۴	-۱	۸۴	-۷	۷۴	+۴	۶۴	-۴	۵۴	-۴	۴۴	+۱	۳۴	-۴	۲۴	-۲	۱۴	-۲	۴
-۲	۹۵	-۲	۸۵	۰	۷۵	+۴	۶۵	-۵	۵۵	-۲	۴۵	-۱	۳۵	۰	۲۵	-۲	۱۵	-۳	۵
۰	۹۶	-۱	۸۶	-۱	۷۶	+۳	۶۶	۰	۵۶	-۱	۴۶	-۳	۳۶	-۲	۲۶	+۲	۱۶	۰	۶
-۲	۹۷	-۱	۸۷	-۲	۷۷	-۲	۶۷	-۳	۵۷	-۳	۴۷	-۲	۳۷	+۳	۲۷	-۲	۱۷	-۱	۷
+۲	۹۸	-۲	۸۸	-۳	۷۸	+۴	۶۸	-۴	۵۸	-۱	۴۸	-۱	۳۸	-۲	۲۸	-۲	۱۸	-۲	۸
-۱	۹۹	+۲	۸۹	-۳	۷۹	-۱	۶۹	+۲	۵۹	-۱	۴۹	۰	۳۹	-۱	۲۹	-۲	۱۹	-۱	۹
-۲	۱۰۰	-۳	۹۰	-۲	۸۰	-۱	۷۰	-۱	۶۰	-۲	۵۰	-۲	۴۰	-۱	۳۰	-۱	۲۰	-۱	۱۰

الگوریتم همگرا شده و سپس دوباره واگرا شده است.

جدول ۵-۸: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۵، β=۰/۵، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۲	۹۱	-۳	۸۱	-۱	۷۱	-۱	۶۱	+۲	۵۱	-۳	۴۱	-۱	۳۱	-۳	۲۱	-۲	۱۱	-۲	۱
-۱	۹۲	-۲	۸۲	-۲	۷۲	+۴	۶۲	+۱	۵۲	-۱	۴۲	۰	۳۲	-۱	۲۲	+۱	۱۲	-۲	۲
-۱	۹۳	-۱	۸۳	-۱	۷۳	۰	۶۳	-۲	۵۳	-۲	۴۳	-۱	۳۳	+۱	۲۳	-۵	۱۳	-۲	۳
۰	۹۴	-۳	۸۴	۰	۷۴	+۴	۶۴	-۱	۵۴	۰	۴۴	+۴	۳۴	-۱	۲۴	-۲	۱۴	+۱	۴
+۳	۹۵	-۴	۸۵	-۱	۷۵	-۲	۶۵	-۳	۵۵	+۲	۴۵	+۳	۳۵	-۳	۲۵	-۷	۱۵	-۱	۵
-۲	۹۶	+۳	۸۶	+۲	۷۶	-۱	۶۶	-۴	۵۶	-۵	۴۶	+۴	۳۶	-۱	۲۶	+۲	۱۶	۰	۶
-۱	۹۷	-۲	۸۷	-۱	۷۷	-۲	۶۷	-۱	۵۷	-۱	۴۷	+۴	۳۷	-۲	۲۷	-۲	۱۷	-۳	۷
-۲	۹۸	-۲	۸۸	۰	۷۸	-۲	۶۸	-۲	۵۸	۰	۴۸	-۱	۳۸	۰	۲۸	-۳	۱۸	-۲	۸
-۳	۹۹	۰	۸۹	-۲	۷۹	+۴	۶۹	-۱	۵۹	-۳	۴۹	-۲	۳۹	-۱	۲۹	+۲	۱۹	-۱	۹
-۲	۱۰۰	-۳	۹۰	-۱	۸۰	-۳	۷۰	-۱	۶۰	-۴	۵۰	+۴	۴۰	-۳	۳۰	-۴	۲۰	-۲	۱۰

جدول ۵-۹: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۱، β=۰/۹، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۱	۹۱	-۳	۸۱	-۱	۷۱	+۱	۶۱	-۱	۵۱	+۴	۴۱	-۱	۳۱	-۲	۲۱	-۱	۱۱	-۱	۱
-۱	۹۲	-۱	۸۲	+۳	۷۲	-۲	۶۲	-۲	۵۲	۰	۴۲	-۱	۳۲	-۱	۲۲	-۱	۱۲	-۲	۲
-۴	۹۳	-۱	۸۳	+۴	۷۳	-۱	۶۳	-۲	۵۳	+۲	۴۳	-۲	۳۳	-۲	۲۳	۰	۱۳	-۲	۳
-۳	۹۴	-۲	۸۴	-۱	۷۴	۰	۶۴	-۱	۵۴	-۲	۴۴	-۱	۳۴	+۳	۲۴	-۲	۱۴	-۱	۴
-۱	۹۵	۰	۸۵	-۲	۷۵	-۱	۶۵	۰	۵۵	+۲	۴۵	-۴	۳۵	-۱	۲۵	-۱	۱۵	+۱	۵
-۱	۹۶	-۱	۸۶	-۱	۷۶	+۲	۶۶	-۱	۵۶	+۱	۴۶	۰	۳۶	-۳	۲۶	-۲	۱۶	-۲	۶
۰	۹۷	+۱	۸۷	+۴	۷۷	-۲	۶۷	-۷	۵۷	+۴	۴۷	-۱	۳۷	-۲	۲۷	-۱	۱۷	-۱	۷
-۲	۹۸	-۳	۸۸	+۲	۷۸	-۱	۶۸	-۱	۵۸	-۱	۴۸	۰	۳۸	-۱	۲۸	-۲	۱۸	-۳	۸
۰	۹۹	-۱	۸۹	۰	۷۹	+۱	۶۹	-۳	۵۹	-۱	۴۹	-۳	۳۹	+۲	۲۹	۰	۱۹	-۱	۹
+۱	۱۰۰	-۱	۹۰	-۲	۸۰	-۱	۷۰	-۱	۶۰	+۲	۵۰	۰	۴۰	-۲	۳۰	-۱	۲۰	-۱	۱۰

روند خاصی قابل مشاهده نیست.

جدول ۵-۱۰: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۹، β=۰/۱، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
۰	۹۱	-۲	۸۱	-۱	۷۱	-۳	۶۱	۰	۵۱	-۱	۴۱	+۱	۳۱	-۳	۲۱	-۴	۱۱	-۱	۱
-۱	۹۲	-۱	۸۲	+۲	۷۲	-۱	۶۲	-۲	۵۲	-۳	۴۲	-۱	۳۲	-۴	۲۲	۰	۱۲	-۲	۲
-۱	۹۳	-۳	۸۳	-۴	۷۳	-۲	۶۳	-۲	۵۳	-۱	۴۳	-۲	۳۳	-۱	۲۳	+۱	۱۳	+۴	۳
-۳	۹۴	-۲	۸۴	-۱	۷۴	-۳	۶۴	+۴	۵۴	-۱	۴۴	-۲	۳۴	-۲	۲۴	-۲	۱۴	-۲	۴
-۱	۹۵	-۳	۸۵	-۲	۷۵	-۱	۶۵	+۴	۵۵	-۱	۴۵	-۳	۳۵	-۱	۲۵	-۹	۱۵	۰	۵
-۲	۹۶	+۴	۸۶	-۳	۷۶	۰	۶۶	-۲	۵۶	-۲	۴۶	+۲	۳۶	-۲	۲۶	-۵	۱۶	-۴	۶
-۱	۹۷	+۴	۸۷	-۱	۷۷	-۲	۶۷	+۴	۵۷	-۲	۴۷	+۴	۳۷	-۱	۲۷	-۳	۱۷	+۱	۷
-۳	۹۸	+۴	۸۸	-۲	۷۸	-۱	۶۸	-۲	۵۸	+۲	۴۸	-۱	۳۸	-۲	۲۸	۰	۱۸	۰	۸
-۱	۹۹	-۱	۸۹	+۱	۷۹	-۱	۶۹	-۱	۵۹	-۲	۴۹	۰	۳۹	-۳	۲۹	-۱	۱۹	-۱	۹
-۲	۱۰۰	۰	۹۰	-۱	۸۰	+۳	۷۰	-۲	۶۰	-۱	۵۰	-۲	۴۰	+۲	۳۰	-۲	۲۰	-۲	۱۰

الگوریتم دو بار همگرا شده است.

جدول ۵-۱۱: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۶، β=۰/۴، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۱	۹۱	-۱	۸۱	+۴	۷۱	-۳	۶۱	-۱	۵۱	۰	۴۱	+۲	۳۱	۰	۲۱	-۲	۱۱	-۲	۱
-۲	۹۲	-۱	۸۲	+۴	۷۲	-۲	۶۲	+۳	۵۲	-۲	۴۲	-۱	۳۲	-۱	۲۲	-۱	۱۲	+۱	۲
+۲	۹۳	-۲	۸۳	-۴	۷۳	-۱	۶۳	-۲	۵۳	-۱	۴۳	-۲	۳۳	+۲	۲۳	-۳	۱۳	-۲	۳
۰	۹۴	۰	۸۴	+۴	۷۴	-۲	۶۴	-۱	۵۴	-۳	۴۴	+۴	۳۴	-۱	۲۴	۰	۱۴	-۲	۴
-۳	۹۵	-۵	۸۵	-۳	۷۵	-۵	۶۵	-۱	۵۵	۰	۴۵	+۴	۳۵	+۲	۲۵	-۷	۱۵	+۱	۵
-۲	۹۶	-۱	۸۶	-۱	۷۶	۰	۶۶	-۲	۵۶	-۱	۴۶	-۳	۳۶	-۴	۲۶	-۱	۱۶	-۱	۶
-۱	۹۷	+۱	۸۷	-۳	۷۷	-۳	۶۷	+۳	۵۷	-۴	۴۷	+۴	۳۷	-۱	۲۷	-۲	۱۷	-۲	۷
-۱	۹۸	-۳	۸۸	-۲	۷۸	-۳	۶۸	-۳	۵۸	-۱	۴۸	+۴	۳۸	-۱	۲۸	۰	۱۸	-۳	۸
+۱	۹۹	-۲	۸۹	-۲	۷۹	+۴	۶۹	-۲	۵۹	+۲	۴۹	-۴	۳۹	-۳	۲۹	+۲	۱۹	-۱	۹
۰	۱۰۰	-۱	۹۰	-۱	۸۰	-۱	۷۰	+۲	۶۰	-۳	۵۰	+۴	۴۰	-۵	۳۰	-۵	۲۰	+۱	۱۰

دو بار همگرا شده است و جواب‌ها دارای کیفیت خوبی هستند.

جدول ۵-۱۲: میزان فرومون صفر، میزان تبخیر فرومون صفر، α=۰/۴، β=۰/۶، تعداد مورچه‌ها=۱۰، تعداد تکرارها=۱۰

تکرار دهم		تکرار نهم		تکرار هشتم		تکرار هفتم		تکرار ششم		تکرار پنجم		تکرار چهارم		تکرار سوم		تکرار دوم		تکرار اول	
-۱	۹۱	-۱	۸۱	+۲	۷۱	-۲	۶۱	۰	۵۱	+۳	۴۱	-۴	۳۱	-۳	۲۱	-۵	۱۱	-۲	۱
-۱	۹۲	+۴	۸۲	-۳	۷۲	+۱	۶۲	۰	۵۲	-۱	۴۲	+۱	۳۲	-۲	۲۲	-۴	۱۲	+۱	۲
-۲	۹۳	+۱	۸۳	-۱	۷۳	-۳	۶۳	+۲	۵۳	-۳	۴۳	۰	۳۳	+۲	۲۳	-۳	۱۳	-۱	۳
-۱	۹۴	۰	۸۴	-۲	۷۴	۰	۶۴	-۱	۵۴	-۱	۴۴	-۱	۳۴	-۱	۲۴	+۴	۱۴	۰	۴
-۳	۹۵	+۴	۸۵	+۳	۷۵	+۱	۶۵	-۱	۵۵	-۱	۴۵	-۷	۳۵	-۲	۲۵	+۴	۱۵	-۲	۵
-۱	۹۶	-۱	۸۶	-۴	۷۶	-۳	۶۶	-۲	۵۶	+۱	۴۶	-۲	۳۶	-۱	۲۶	-۱	۱۶	-۳	۶
-۳	۹۷	+۴	۸۷	-۱	۷۷	-۱	۶۷	-۳	۵۷	-۲	۴۷	-۱	۳۷	-۳	۲۷	+۴	۱۷	-۱	۷
-۲	۹۸	-۳	۸۸	-۱	۷۸	-۲	۶۸	-۱	۵۸	-۱	۴۸	-۶	۳۸	+۱	۲۸	+۴	۱۸	-۱	۸
+۱	۹۹	-۳	۸۹	+۴	۷۹	-۳	۶۹	+۱	۵۹	۰	۴۹	+۲	۳۹	-۲	۲۹	-۳	۱۹	-۵	۹
-۲	۱۰۰	-۲	۹۰	+۳	۸۰	-۱	۷۰	-۲	۶۰	-۱	۵۰	-۱	۴۰	-۲	۳۰	+۴	۲۰	-۱	۱۰

۵-۵ مقایسه نتایج الگوریتم کولونی مورچگان با روش‌های دیگر

در این قسمت برنامه توسعه یافته برای روش الگوریتم کولونی مورچگان، با روش‌های دیگری نظیر روش مخروط شناور و برنامه ریزی پویا از طریق ذکر چند مثال مقایسه خواهد شد. هدف از این مقایسات، بررسی درستی و صحت پاسخ‌های بدست آمده از روش الگوریتم کولونی مورچگان می‌باشد.

۵-۵-۱ مثال اول

در این مثال یک مدل بلوکی دارای ۳ ردیف و ۹ ستون وجود دارد که دارای سه بلوک ماده معدنی مطابق شکل ۵-۵ می‌باشد. این مدل دارای محدوده بهینه نمی‌باشد. از حل کردن این مدل با روش‌هایی نظیر مخروط شناور و برنامه ریزی پویا نیز نتیجه‌ای بدست نخواهد آمد و این مدل فاقد محدوده بهینه می‌باشد. نتایج حاصل از حل این مدل توسط برنامه رایانه‌ای توسعه یافته در جداول ۵-۱۳ و ۵-۱۴ ذکر شده است. همانگونه که مشاهده می‌شود با تنظیمات مختلف و در ۲۵ بار اجرای برنامه، جوابی بدست نیامده است.

-۲	-۲	-۲	-۲	-۲	-۲	-۲	-۲	-۲
-۴	-۴	-۴	-۴	-۴	-۴	-۴	-۴	-۴
-۶	-۶	+۱۸	-۶	+۱۲	-۶	+۱۴	-۶	-۶

شکل ۵-۵: مدل ۱ با سه ردیف و نه ستون

جدول ۵-۱۳: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۱، با ۱۰ تکرار و در هر تکرار ۱۰ مورچه میزان فرومون=۰، میزان تبخیر فرومون=۰، α=۰/۴، β=۰/۶.

مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا
-	-	۱۹	-	-	۱۰	-	-	۱
-	-	۲۰	-	-	۱۱	-	-	۲
-	-	۲۱	-	-	۱۲	-	-	۳
-	-	۲۲	-	-	۱۳	-	-	۴
-	-	۲۳	-	-	۱۴	-	-	۵
-	-	۲۴	-	-	۱۵	-	-	۶
-	-	۲۵	-	-	۱۶	-	-	۷
			-	-	۱۷	-	-	۸
			-	-	۱۸	-	-	۹

جدول ۵-۱۴: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۱، با ۱۰ تکرار و در هر تکرار ۲۰ مورچه میزان فرومون=۰، میزان تبخیر فرومون=۰، α=۰/۴، β=۰/۶،

مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا
-	-	۱۹	-	-	۱۰	-	-	۱
-	-	۲۰	-	-	۱۱	-	-	۲
-	-	۲۱	-	-	۱۲	-	-	۳
-	-	۲۲	-	-	۱۳	-	-	۴
-	-	۲۳	-	-	۱۴	-	-	۵
-	-	۲۴	-	-	۱۵	-	-	۶
-	-	۲۵	-	-	۱۶	-	-	۷
			-	-	۱۷	-	-	۸
			-	-	۱۸	-	-	۹

۵-۵-۲ مثال دوم

در این مثال یک مدل بلوکی با ۱۲ ستون و ۵ ردیف وجود دارد (شکل ۵-۶)، این مدل دارای محدوده بهینه‌ای با ارزش (۱۳+) می‌باشد

-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶
-۶	-۶	+۱۳	-۶	+۱۴	-۶	+۱۵	-۶	+۱۵	-۶	-۶	-۶
-۸	-۸	-۸	+۱۰	-۸	-۸	+۱۰	-۸	-۸	-۸	-۸	-۸
-۸	-۸	-۸	-۸	+۱۶	-۸	+۱۶	+۶	+۶	-۸	-۸	-۸
-۱۰	-۱۰	-۱۰	-۱۰	+۲۴	-۱۰	+۲۰	-۱۰	-۱۰	-۱۰	-۱۰	-۱۰

شکل ۵-۶: مدل ۲ با پنج ردیف و دوازده ستون

از حل کردن مدل ۲ (مدل ارائه شده در شکل ۵-۶) توسط روش برنامه ریزی پویا پاسخ بهینه برابر با ۱۳+ بدست خواهد آمد. همچنین روش مخروط شناور نیز می‌تواند در این مدل پاسخ بهینه را بدست آورد. شکل محدوده جواب در شکل ۵-۷ نشان داده شده است.

-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶	-۶
-۶	-۶	-۶	+۱۵	-۶	+۱۵	-۶	+۱۴	-۶	+۱۳	-۶	-۶
-۸	-۸	-۸	-۸	-۸	+۱۰	-۸	-۸	+۱۰	-۸	-۸	-۸
-۸	-۸	-۸	+۶	+۶	+۱۶	-۸	+۱۶	-۸	-۸	-۸	-۸
-۱۰	-۱۰	-۱۰	-۱۰	-۱۰	+۲۰	-۱۰	+۲۴	-۱۰	-۱۰	-۱۰	-۱۰

شکل ۵-۷: محدوده بهینه متعلق به مدل بلوکی ارائه شده در شکل ۵-۶

با حل کردن همین مدل توسط برنامه توسعه یافته نیز جواب برابر با ۱۳+ بدست خواهد آمد. ولی همانگونه که در جداول ۵-۱۵ تا ۵-۱۸ مشاهده می‌شود، در این حالت الگوریتم برای یافتن پاسخ نیاز به تعداد مورچه‌های بیشتری دارد. جواب بدست آمده توسط الگوریتم به صورت شکل ۵-۸ می‌باشد که مقادیر صفر در این شکل نشان‌دهنده بلوک‌هایی است که در محدوده بهینه جایی ندارند و مقادیر ۱ بیانگر بلوک‌هایی است که در محدوده نهایی جای می‌گیرند. عددی که در ردیف آخر وجود دارد، جواب نهایی پیت بدست آمده می‌باشد.

جدول ۵-۱۵: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۲، با ۱۰ تکرار و در هر تکرار ۱۰ مورچه میزان تبخیر فرومون=۰/۲، α=۰/۴، β=۰/۶،

مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا
-	-	۱۹	-	-	۱۰	-	-	۱
-	-	۲۰	۴	۷	۱۱	-	-	۲
-	-	۲۱	-	-	۱۲	-	-	۳
-	-	۲۲	-	-	۱۳	-	-	۴
-	-	۲۳	-	-	۱۴	-	-	۵
-	-	۲۴	-	-	۱۵	-	-	۶
-	-	۲۵	-	-	۱۶	-	-	۷
			-	-	۱۷	-	-	۸
			-	-	۱۸	-	-	۹

جدول ۵-۱۶: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۲، با ۱۰ تکرار و در هر تکرار ۲۰ مورچه میزان تبخیر فرومون=۰/۲، α=۰/۴، β=۰/۶،

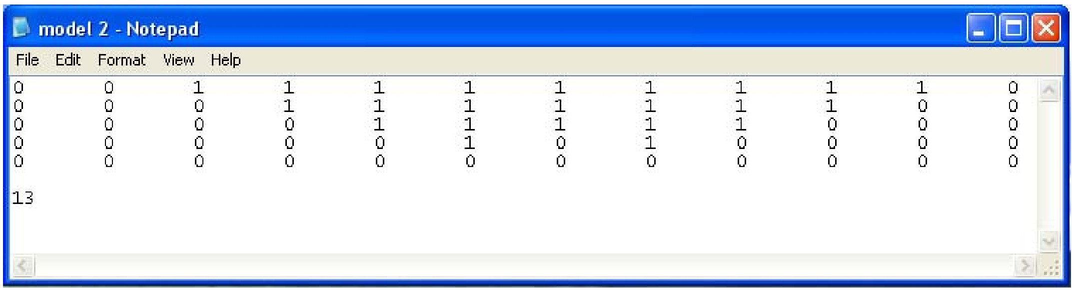
مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا
-	-	۱۹	-	-	۱۰	-	-	۱
-	-	۲۰	۶	۷	۱۱	-	-	۲
-	-	۲۱	-	-	۱۲	-	-	۳
-	-	۲۲	-	-	۱۳	-	-	۴
-	-	۲۳	-	-	۱۴	-	-	۵
-	-	۲۴	-	-	۱۵	-	-	۶
-	-	۲۵	-	-	۱۶	-	-	۷
			-	-	۱۷	-	-	۸
			۲	۷	۱۸	-	-	۹

جدول ۵-۱۷: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۲، با ۱۵ تکرار و در هر تکرار ۲۰ مورچه میزان تبخیر فرومون=۰/۲، α=۰/۴، β=۰/۶،

مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا
-	-	۱۹	-	-	۱۰	-	-	۱
۵	۵	۲۰	-	-	۱۱	-	-	۲
-	-	۲۱	-	-	۱۲	-	-	۳
-	-	۲۲	-	-	۱۳	-	-	۴
-	-	۲۳	-	-	۱۴	۵	۸	۵
-	-	۲۴	-	-	۱۵	-	-	۶
-	-	۲۵	-	-	۱۶	-	-	۷
			-	-	۱۷	-	-	۸
			-	-	۱۸	-	-	۹

جدول ۵-۱۸: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۲، با ۲۰ تکرار و در هر تکرار ۲۵ مورچه میزان تبخیر فرومون=۰/۲، α=۰/۴، β=۰/۶

اجرا	تکرار یابنده جواب بهینه	مورچه یابنده جواب بهینه	اجرا	تکرار یابنده جواب بهینه	مورچه یابنده جواب بهینه	اجرا	تکرار یابنده جواب بهینه	مورچه یابنده جواب بهینه
۱	۸	۳	۱۰	-	-	۱۹	-	-
۲	-	-	۱۱	-	-	۲۰	-	-
۳	-	-	۱۲	-	-	۲۱	۹	۶
۴	۵	۳	۱۳	-	-	۲۲	-	-
۵	-	-	۱۴	۵	۷	۲۳	-	-
۶	-	-	۱۵	۵	۸	۲۴	۳	۵
۷	۳	۹	۱۶	-	-	۲۵	۷	۷
۸	-	-	۱۷	-	-			
۹	۷	۲	۱۸	۲	۲			



شکل ۵-۸: نمایی از جواب برنامه توسعه یافته برای مدل ۲

۵-۵-۳ مثال سوم

با حل یک مدل با ۳ ردیف و ۷ ستون مشاهده خواهد شد که الگوریتم حتی با وجود تعداد اندک مورچه‌ها و تکرارها، برای این مسئله پاسخی قابل قبول ارائه می‌دهد. مدلی با چنین مشخصات در شکل ۵-۹ (مدل ۳) ارائه شده است.

-۲	-۲	-۲	-۲	-۲	-۲	-۲
-۲	-۲	-۲	+۳	-۲	-۲	-۲
-۲	-۲	+۱۰	-۲	+۱۰	-۲	-۲

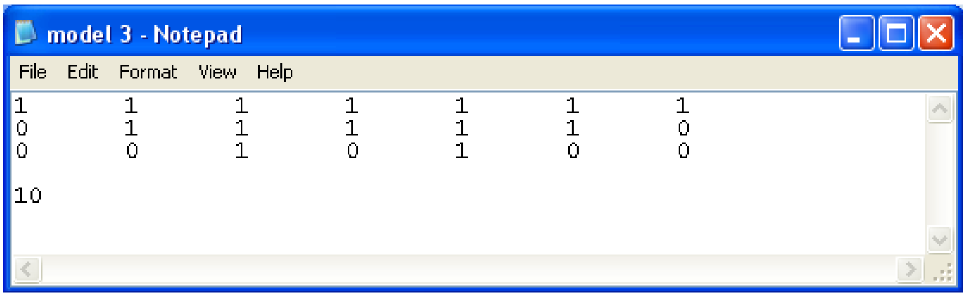
شکل ۵-۹: مدل ۳ با سه ردیف و هفت ستون

این مدل دارای جواب بهینه با ارزش برابر با ۱+ بوده که جواب بهینه این مدل که توسط روش برنامه ریزی پویا بدست آمده است در شکل ۵-۱۰ مشاهده می‌شود. شایان ذکر است که الگوریتم مخروط شناور نمی‌تواند برای این مدل محدوده نهایی بهینه را بیابد.

-۲	-۲	-۲	-۲	-۲	-۲	-۲
-۲	-۲	-۲	+۳	-۲	-۲	-۲
-۲	-۲	+۱۰	-۲	+۱۰	-۲	-۲

شکل ۵-۱۰: جواب بهینه مدل موجود در شکل ۵-۸

اولین جواب بدست آمده (حاصل از اجرای اول) برای مدل ۳ در شکل ۵-۱۱ ارائه شده است. همچنین نتایج ۲۵ بار اجرای الگوریتم در جدول ۵-۱۹ ذکر شده‌اند.



شکل ۵-۱۱: نتایج حاصل از برنامه توسعه یافته برای مدل ۳

جدول ۵-۱۹: نتایج حاصل از ۲۵ بار اجرای برنامه برای حل مدل ۳، با ۱۰ تکرار و در هر تکرار ۱۰ مورچه
میزان تبخیر فرومون=۰/۲، α=۰/۴، β=۰/۶.

مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا	مورچه یابنده جواب بهینه	تکرار یابنده جواب بهینه	اجرا
۸	۳	۱۹	۶	۷	۱۰	-	-	۱
۲	۴	۲۰	۱	۵	۱۱	۷	۶	۲
۳	۸	۲۱	۴	۳	۱۲	۷	۴	۳
-	-	۲۲	-	-	۱۳	-	-	۴
۴	۹	۲۳	۴	۷	۱۴	۲	۸	۵
۶	۸	۲۴	۲	۷۶	۱۵	۸	۳	۶
۳	۶	۲۵	۶	۶	۱۶	۴	۵	۷
			-	-	۱۷	-	-	۸
			۲	۶	۱۸	۳	۶	۹

۵-۶ بررسی نتایج حاصله

۵-۶-۱ بررسی علت واگرا شدن نتایج حاصل از الگوریتم

با بررسی مدلسازی محدوده بهینه نهایی با الگوریتم کولونی مورچگان ممکن است مسئله واگرا شدن روی دهد. بدین معنی که الگوریتم پس از یافتن بهترین جواب و مسیر بهترین جواب، از این مسیر منحرف شده و باز هم به سمت جواب‌های کاملاً غیربهینه پیش می‌رود. علت واگرا شدن الگوریتم در ادامه توضیح داده شده است:

الف- با توجه به مقادیر عددی موجود در جداول ۵-۱ تا ۵-۱۲ مشاهده می‌شود که بعضی از پاسخ‌ها به صورت عددی منفی می‌باشند ولی باید توجه داشت که ممکن است این پاسخ منفی با اندکی تغییر به جوابی صحیح تبدیل شود، به این معنی که تمام پاسخ‌های منفی به یک صورت نمی‌باشند. بطور مثال در مورد مثالی که در این پایان نامه با الگوریتم کولونی مورچگان حل شده است (شکل ۵-۴) که پاسخ بهینه به صورت کلی شکل ۵-۱۲ می‌باشد که دارای ارزش حداکثر ۴+ است:

−۱	−۱	−۱	−۱	−۱	+۱
	+۲	−۲	−۲	+۳	
		+۷			

شکل ۵-۱۲: پیت بهینه نهایی متعلق به مدل بلوکی موجود در شکل ۵-۴.

درحالیکه ممکن است یک مورچه با طی مسیر نسبتاً صحیح، حالتی به صورت شکل ۵-۱۳ را جستجو نماید که تنها در یک بلوک با بهینه‌ترین حالت موجود در شکل ۵-۱۲ تفاوت دارد ولی دارای ارزش ۳- می‌باشد.

−۱	−۱	−۱	−۱	−۱	+۱
	+۲	−۲	−۲	+۳	

شکل ۵-۱۳: اختلاف در یک بلوک با شکل ۵-۱۲، و پاسخ‌های کاملاً متفاوت.

بنابراین قانون حرکت تصادفی می‌تواند در بعضی مواقع به مسیر نسبتاً صحیح ولی پاسخ کاملاً اشتباه منجر شود.

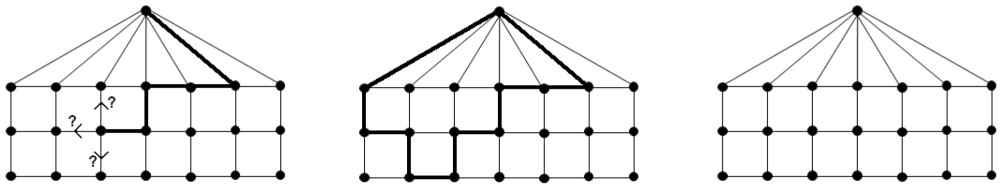
ب- همانگونه که در این مدل کوچک مشاهده می‌شود، الگوریتم قادر است یک جواب بدست آورد (در اینجا لازم است ذکر شود که الگوریتم و یا برنامه توسعه یافته باید بتواند به محض یافتن پاسخ بهینه (نسبت به بقیه پاسخ‌ها) آن را ذخیره نماید و با بقیه پاسخ‌های یافت شده مقایسه کند). ولی راهی برای فهمیدن اینکه آیا این جواب، جواب بهینه است و یا خیر وجود ندارد، برای مقابله با این مشکل می‌توان راه‌حل‌های زیر را پیشنهاد کرد:

- زیاد کردن تعداد مورچه‌ها در یک تکرار (زیاد کردن تعداد تکرارها نمی‌تواند به اندازه کافی کارساز باشد) به منظور جستجوی مناطقی دیگر؛

- پس از هر تعداد مشخص از تکرارها، مقدار فرومون گذاشته شده روی مسیرها به اندازه‌ای کاهش پیدا کند که مورچه‌ها به راحتی به سمت مسیرهای دیگر نیز متمایل شوند.

- استفاده از یک لیست تابو (با وجود اینکه این مورد شاید بتواند در مدل‌های کوچک به همگرایی سریعتر مورچه‌ها کمک کند ولی در مدل‌های بزرگ، به دو علت می‌تواند کاربرد الگوریتم کولونی مورچگان را با مشکل مواجه سازد:

- در یک مدل بزرگ، هر حرکت مورچه در یک مسیر باید با کل حرکت‌های طی شده مقایسه شود که این امر می‌تواند به کند شدن الگوریتم منجر شود.
- مسیر دقیق حرکت مورچه از قبل مشخص نیست بنابراین مقایسه این مسیر نامشخص با مسیرهای طی شده قبلی کاری نادرست است. به شکل ۵-۱۴ دقت کنید. در سمت راست این شکل حالتی که ممکن است برای مورچه پیش آید، نشان داده شده است.



شکل ۵-۱۴: (راست) کل مسیرهای ممکن، (وسط) یکی از مسیرهایی که قبلاً طی شده، (چپ) مسیر جاری حرکت

۵-۶-۲ بررسی نتایج حاصل از مثال‌ها

از مثال‌ها و مقایسه آنها با دیگر الگوریتم‌ها می‌توان نتایج زیر را دریافت:

- الگوریتم مورچه به شدت به تعداد سطرها حساس می‌باشد. این حساسیت ممکن است به دلیل نامتقارن بودن شکل مسئله باشد زیرا برخلاف عملکرد الگوریتم مورچه بر روی یک گراف، در این مسئله مورچه از یک سمت (سطح زمین) شروع به حرکت نموده و با تعدادی متناهی از حرکات به مسیر خود ادامه می‌دهد. ذکر این نکته ضروری است که در مقابل سه حرکت مختلف (حرکت به

سمت چپ، حرکت به بالا و حرکت به بالا و سمت چپ) فقط یک حرکت مورچه را به سمت عمق هدایت می‌نماید.

- این الگوریتم دارای تعداد زیادی پارامتر برای تنظیم است که ممکن است هر یک از این پارامترها برای مدلی با ابعادی خاص مناسب باشند.

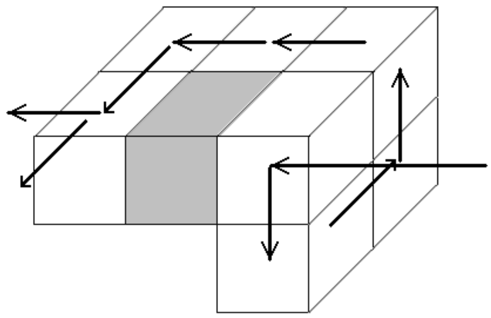
- در مدل‌هایی که بررسی شدند، پاسخ‌های بدست آمده از الگوریتم کولونی مورچگان همانند پاسخ‌های بدست آمده از روش برنامه ریزی پویا می‌باشند و جواب بهتری نسبت به مخروط شناور محاسبه می‌کند.

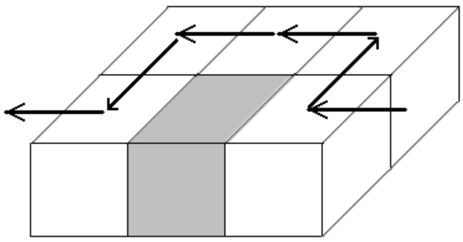
- با توجه به اینکه اثبات شده است که روش برنامه ریزی پویا در حالت دو بعدی جواب بهینه واقعی را بدست می‌آورد، لذا این الگوریتم نیز با تنظیم پارامترها قادر است محدوده بهینه واقعی را در حالت دو بعدی محاسبه نماید.

۵-۷ بررسی مدلسازی الگوریتم در حالت سه بعدی

استفاده از الگوریتم کولونی مورچگان برای حالت سه بعدی همانند حالت دو بعدی خواهد بود، با این تفاوت که حرکت مورچه‌ها برای حالت سه بعدی (محدودیت شیب) درنظر گرفته می‌شود که این امر با توجه به طبیعت الگوریتم و با عنایت به مشکلات زیر امکان پذیر نمی‌باشد.

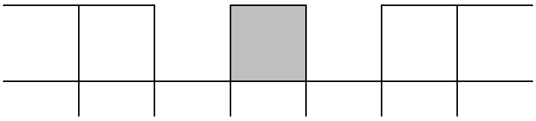
در حالت دو بعدی کلیه بلوک‌ها به صورت تجمعی محاسبه می‌شوند. مورچه در یک مدل که به صورت تجمعی است به دنبال پاسخ می‌گردد. در نتیجه کلیه‌ی بلوک‌های موجود در مسیر و بلوک‌های رویی آن را جستجو می‌نماید. ولی در یک مدل سه بعدی ممکن است که بعضی از بلوک‌ها (حتی با وجود مثبت بودن ارزش) توسط مورچه مورد جستجو واقع نشوند و در نتیجه در محدوده نهایی قرار نگیرند. به شکل‌های ۵-۱۵ توجه کنید.



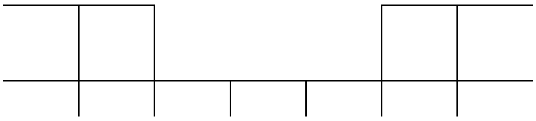


شکل ۵-۱۵: مسیر حرکت احتمالی یک مورچه، بلوک‌های هاشوردار خارج از مسیر جستجو.

با بررسی این شکل مشاهده می‌شود که ممکن است محدوده نهایی در مسیر گردش مورچه به صورت شکل ۵-۱۶ باشد، درحالیکه این احتمال وجود دارد که بلوک مشخص شده با هاشور نیز دارای ارزش مثبت بوده و محدوده نهایی اصلی به صورت شکل ۵-۱۷ باشد.

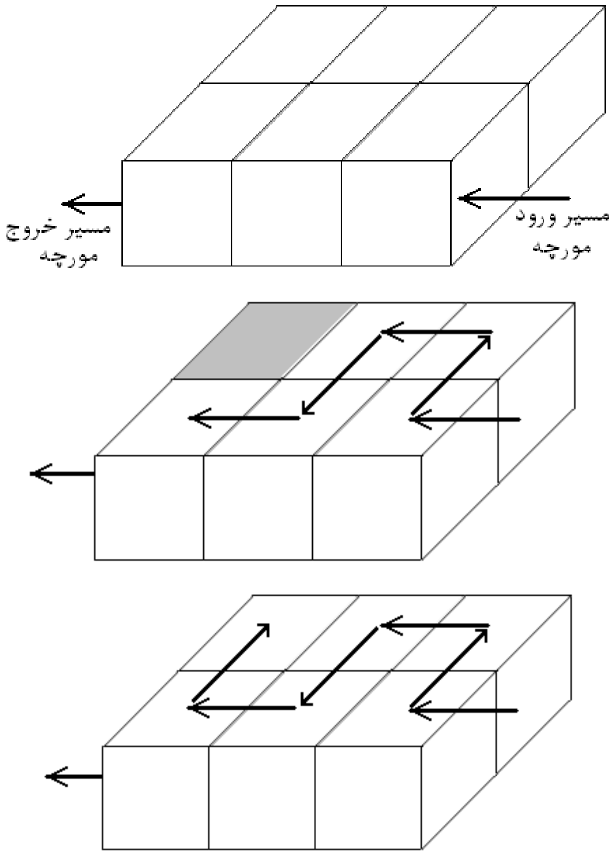


شکل ۵-۱۶: در مسیر حرکت شکل ۵-۱۵، بلوک هاشور خورده جستجو نمی‌شود.



شکل ۵-۱۷: محدوده نهایی بهینه احتمالی شکل ۵-۱۶.

در شکل ۵-۱۵، مسیر حرکت مورچه‌ها شبیه به مسئله فروشنده دوره‌گرد می‌باشد که در فصل سوم بیان شده است. برای مثال در شکل ۵-۱۸ هیچ راهی وجود ندارد که برطبق آن مورچه‌ها بتوانند هر شش بلوک را طی نمایند؛ درحالیکه از هر بلوک فقط یکبار می‌گذرند و نیز مسیر ورود و خروج در بلوک‌ها نیز یکسان است.



شکل ۵-۱۸: شباهت حرکت مورچه‌ها به مسیر موجود در مسئله‌ی فروشنده دوره گرد.

در اینجا ممکن است سوالی پیش آید که شاید مورچه‌ها بتوانند با حرکت در سه بُعد (یعنی علاوه بر حرکات گفته شده در شکل ۵-۱۸ و حرکت به عمق یا یک ردیف بالاتر) به تمام بلوک‌ها دسترسی داشته باشند ولی باز هم این احتمال وجود دارد که محدوده بهینه در همان یک سطر باشد که با حرکت مورچه‌ها به سمت عمق، چنین محدوده‌ای پیدا نخواهد شد.

فصل ششم

نتیجه گیری و پیشنهادات

فصل ۶

۶-۱ نتیجه گیری

الگوریتم کولونی مورچگان الگوریتمی است که تاکنون توانسته به حل مسائل بسیاری کمک کند. در این پایان نامه نیز مسئله تعیین محدوده نهایی معادن روباز با استفاده از این الگوریتم مورد بررسی قرار گرفته است. نتیجه کار در حالت دو بعدی بر بهینه بودن محدوده بدست آمده اشاره دارد، بطوریکه در مثال‌های حل شده، کلیه جواب‌های حاصل، منطبق بر پاسخ روش برنامه ریزی پویا می‌باشند.

با این حال در مسئله مورد بحث در این پایان نامه به نظر می‌رسد که الگوریتم کولونی مورچگان دارای نواقص و یا به عبارت بهتر ناکارامدی‌هایی باشد که می‌توانند در سه دسته طبقه‌بندی شوند.

۱- ناکارایی‌های ناشی از ذات این الگوریتم؛

۲- ناکارایی‌های ناشی از مدل‌سازی نادرست مسئله برای حل توسط الگوریتم کولونی مورچگان؛

۳- ناکارایی‌های ناشی از عدم تناسب بین مسئله و الگوریتم کولونی مورچگان.

مواردی که در این پایان نامه با آنها برخورد شده است و همچنین مواردی که می‌توانند به عنوان پیشنهاد برای کارهای بعدی مورد استفاده قرار گیرند در این فصل و در ادامه مورد بحث قرار خواهند گرفت.

۱) یکی از مشکلات موجود برای حل مسئله تعیین محدوده نهایی معادن روباز با کمک الگوریتم کولونی مورچگان، نیاز این الگوریتم به تنظیم پارامترهای فراوان می‌باشد. این پارامترها را می‌توان به

این صورت برشمرد: تعداد مورچه‌ها، تعداد تکرارها، پارامتر α (توان روی پارامتر میزان فرومون در قانون گذر تصادفی)، پارامتر β (توان روی پارامتر مجاورت در قانون گذر تصادفی)، میزان فرومون گذارده شده بر روی مسیرها، میزان تبخیر فرومون، طریقه محاسبه پارامتر مجاورت در قانون گذر تصادفی برحسب شرایط مسئله، پارامتر Q که یک ثابت برای محاسبه پارامتر مجاورت می‌باشد. علاوه بر همه اینها ممکن است که برحسب ابعاد و گستردگی مسئله، هر یک از این پارامترها که تاکنون دارای مقدار بهینه بوده‌اند؛ دیگر جوابگوی حالت جدید نباشند، که در این مسئله با آن مواجه هستیم.

در این پایان نامه مشاهده شد که برای مدل‌های با ابعاد کوچک (برای مثال مدل‌های با سه سطر) پارامترهای α و β برابر ۰/۶ و ۰/۴ و یا بالعکس جواب‌های بهتری بدست می‌دهند. پارامتر میزان فرومون برحسب پاسخ بدست آمده متغیر خواهد بود، پارامتر تبخیر فرومون نیز ثابت و برابر با ۰/۲ درنظر گرفته شد. همچنین با زیاد شدن ابعاد مسئله، تغییر دادن این پارامترها نتوانست به یافتن جواب منجر شود، بلکه زیاد شدن تعداد مورچه‌ها و تعداد تکرارها است که به یافتن جواب کمک می‌کند.

۲) هیچگونه معیاری برای مشخص بودن جواب بهینه وجود ندارد، معیارهای توقف الگوریتم را می‌توان رسیدن به یک جواب خاص (و یا بهتر شدن جواب‌ها پس از تعداد مشخصی تکرار) و یا تعداد تکرارهای از پیش تعیین شده معین کرد.

۳) الگوریتم کولونی مورچگان بر روی گراف پیاده‌سازی می‌شود. در این پایان نامه برای راحتی کار برنامه‌نویسی، هر مدل بلوکی به شکل یک آرایه (جدول) تبدیل می‌شود. این عمل ممکن است الگوریتم را از ذات خود دور سازد.

۴) در مدل‌های بلوکی بکار رفته در این پایان نامه، مدل بلوکی از سطح زمین شروع شده و به سمت عمق ادامه می‌یابد. همانگونه که پیش از این نیز ذکر شده، برای یافتن پاسخ، چهار حرکت برای مورچه‌ها پیشنهاد شده است (حرکت به سمت بالا، حرکت به سمت چپ، حرکت به سمت بالا و چپ و حرکت به سمت پایین). در اینجا مشاهده می‌شود که به ازای سه حرکت (بالا، چپ و بالا و چپ) که

تمایل به خروج مورچه از مدل بلوکی دارند فقط یک حرکت (حرکت به سمت پایین) تمایل به حرکت مورچه به سمت مناطق جدید در فضای جستجو دارد.

۵) همانگونه که در مورد (۴) نیز ذکر شد، می‌توان اینگونه بیان کرد که مورچه‌ها تمایلی به حرکت به سمت عمق ندارند. این مورد را می‌توان ناشی از نامتقارن بودن مسئله نیز دانست. برای مثال مورچه‌ای را فرض کنید که برای حرکت بر روی گراف از یک گره شروع به حرکت کرده و به یک گره دیگر می‌رسد. این مورچه برای حرکت خود دارای چندین محدودیت است، درحالیکه در این مسئله مورچه از نقطه‌ای شروع به حرکت می‌کند و برای یافتن پاسخ بهینه مجبور است تعداد بلوک‌های بیشتری را (به سمت عمق (که مساوی است با تعداد بیشتری انتخاب) که همانگونه که ذکر شد این انتخاب‌ها بیشتر به سمت خروج از مدل تمایل دارند) بپیماید.

۶) مدلسازی مسئله در حالت سه بعدی بسیار مشکل است زیرا ممکن است حالت‌های مختلفی پیش آیند که در فصل ۵ به آنها اشاره شده است.

۷) در این مسئله استفاده از لیست تابو با اشکال انجام می‌گیرد، زیرا ممکن است یک مورچه قسمتی از مسیر خود را براساس رد فرومونی بپیماید و قسمتی دیگر را بدون رد فرومونی پیمایش کند. در این حالت تنظیم اینکه آیا مورچه باید براساس رد فرومونی بقیه مسیر را بپیماید یا خیر و یا اینکه براساس لیست تابوها حرکت کند، بسیار مشکل می‌باشد.

۸) امکان واگرا شدن مورچه‌ها از مسیر بسیار زیاد است و این به دلیل حرکت تصادفی مورچه‌ها است که می‌توانند مسیر بهینه‌ای را که طی می‌کنند از نیمه راه تغییر دهند.

۹) این الگوریتم به شدت به ابعاد مسئله حساس بوده و نیاز به تنظیمات جدید دارد.

۱۰) با زیاد شدن تعداد سطرها و ستون‌ها (زیاد شدن تعداد بلوک‌ها) نمی‌توان علت گم شدن مورچه (یا به عبارت دیگر گم کردن مسیر توسط مورچه‌ها که باعث واگرا شدن جواب می‌شود) را

دریافت، با این حال با حل یک مدل کوچک می‌توان گفت که به دلیل حرکت تصادفی مورچه‌ها در هر مرحله، ممکن است که مورچه‌ها از مسیر فرومونی منحرف شوند.

به عنوان یک نتیجه کلی می‌توان موارد زیر را عنوان نمود:

همانطور که در فصل پنج بیان شد، این مسئله قابلیت تبدیل به حالت سه بعدی را ندارد. زیرا در حالت سه بعدی به مسئله فروشنده دوره گرد تبدیل خواهد شد. ایراد عمده این روش (و در کل تمام روش‌های ابتکاری) عدم توان تشخیص بهینه بودن جواب بدست آمده است که به نظر غیر قابل رفع می‌باشد و الگوریتم راهی برای فهمیدن آن ارائه نمی‌کند. با این حال ممکن است الگوریتم کولونی مورچگان بتواند در مسائلی از این دست (و بخصوص متقارن)، و با مدلسازی متفاوت عملکرد بهتری داشته باشد.

۶-۲ پیشنهادات

برای حل این مشکلات می‌توان موارد زیر را پیشنهاد کرد:

۱- افزایش تعداد مورچه‌ها و تعداد تکرارها برای یافتن پاسخ بهینه(تر)؛

۲- وزن‌دار کردن حرکت مورچه‌ها به سمت پایین؛

۳- متقارن کردن مسئله (به عنوان مثال حرکت مورچه از نقطه‌ای در داخل مدل شروع شود)؛

۴- تلاش برای بهتر مدلسازی کردن مسئله و یا مدلسازی بهتر مسئله؛

۵- می‌توان به جای استفاده از لیست تابوها از دیگر الگوریتم‌های ابتکاری استفاده کرد.

۶- پیشنهاد می‌شود که برای کارهای آتی، مدلسازی مسئله به نحو متفاوتی انجام گیرد.

۷- در آینده می‌توان برای پیش‌بینی رفتار مورچه‌ها در عمق از روش‌هایی مانند روش جستجوی تابو، روش شبیه‌سازی تبریدی و حتی نظریه بازی‌ها استفاده کرد.

منابع فارسی

- حسینی، س. م. ع.، (۱۳۸۷)، پایان نامه کارشناسی ارشد، "تعیین محدوده بهینه نهایی و برنامه‌ریزی تولید آنومالی A سنگ آهن سنگان"، دانشکده مهندسی معدن، نفت و ژئوفیزیک، دانشگاه صنعتی شاهرود.

- خالوکاکایی. رضا.، (۱۳۸۴)، "بررسی روشهای مختلف طراحی محدوده بهینه نهایی در معادن روباز"، **مجله علوم و فنون دانشگاه صنعتی شاهرود**، شماره‌های ششم و هفتم، (بهار و تابستان ۱۳۸۴)، ص ۱۹-۳۱.

- خالوکاکایی. رضا.، (۱۳۸۵)، "طراحی محدوده بهینه نهایی در معادن روباز با روش‌های اصلاح شده مخروط شناور دو"، نشریه دانشکده فنی، جلد ۱، شماره ۳، ص. ۲۹۷-۳۰۷.

- عظیمی ع.، کاکایی ر.، (۱۳۸۸)، "معرفی یک مکانیزم جدید در بهینه‌سازی به روش کولونی مورچگان برای استفاده در طراحی محدوده نهایی معادن روباز"، چهلمین کنفرانس ریاضی ایران، ۲۶ تا ۲۹ مردادماه ۱۳۸۸، دانشگاه صنعتی شریف، تهران، ایران.

- عظیمی ع.، کاکایی. ر.، (۱۳۸۸)، "مدل‌سازی بلوک‌های اقتصادی یک معدن روباز به منظور تعیین محدوده نهایی معادن، با کمک روش بهینه‌سازی کولونی مورچگان"، چهلمین کنفرانس ریاضی ایران، ۲۶ تا ۲۹ مردادماه ۱۳۸۸، دانشگاه صنعتی شریف، تهران، ایران.

- میرزایی نصیرآباد، ح.، (۱۳۸۱)، پایان نامه کارشناسی ارشد، "تعیین محدوده بهینه و برنامه ریزی تولید به کمک الگوریتم ژنتیک"، دانشکده مهندسی معدن، نفت و ژئوفیزیک، دانشگاه صنعتی شاهرود.

منابع لاتین

- Abdallah, H. M. E., Hassan T. D., Ahmed B., (2009) “Using Ant Colony Optimization Algorithm For Solving Project Management Problems”, **International Journal of Expert Systems with Applications**, Vol. 36, pp. 10004-10015.

- Akbari, A. D., Osanloo, M., Shirazi, M. A., (2008), “Determination of ultimate pit limits in open mines using real option approach”, **IUST International Journal of Engineering Science**, Vol. 19, No. 5-1, Page 23-38.

- ALFORD. C. G., and WHITTLE J., (1986), “Application of Lerchs-Grossman pit optimization to the design of open pit mines”, The AusIMM/IE Aust Newman Combined Group, Large Open Pit Mining Conference, October 1986, pp. 201-207.

- Carlson, T. R., Erickson, J. D., O’Briant D. T. and Pana, M. T., (1966), “**Computer techniques in mine planning**”, Mining engineering, Vol. 18, No. 5, pp. 53-56

- Chicoisne, R., Espinoza, D., Goycoolea, M., Moreno, E., Rubio, E., “**A new algorithm for the open-pit mine scheduling problem**”, 2009.

- Chui-Yu Chiu, I-Ting Kuo, Chia-Hao Lin, (2009), “Applying artificial immune system and ant algorithm in air-conditioner market segmentation”, **Expert Systems with Applications**, Vol. 36, pp. 4437-4442.

- Djilani. M. C., and Dowd P. A., (1994), “Optimum production scheduling in open pit mines”, LEEDS UNIVERSITY MINING ASSOCIATION JOURNAL, pp. 133-140.

- Dorigo, M., and Di Caro, G., (2006), “Ant Colony Optimization: A New Meta Heuristic”, IRIDIA.

- Dorigo, M., Di Caro, G., Gambardella, L. M., (1999), “Ant Algorithms for Discrete Optimization”, IRIDIA CP 194/6, Massachusetts Institute of Technology, Artificial Life, Vol. 5, pp. 137-172.

- Dorigo, M., Maniezzo, V., Colorni, A., (1991), “Positive Feedback as a Search Strategy”, Report n. 91-016, year 1991.

- Dorigo, M., Maniezzo, V., Colorni, A., (1996), “The Ant System: Optimization by a colony of cooperating agents”, **IEEE Transactions on Systems, Man, and Cybernetics**–Part B, Vol. 26, No. 1, pp. 1-13

- Dorigo, M., and Socha, K., (2006), “An Introduction to Ant Colony Optimization”, IRIDIA-Technical Report Series No. TR/IRIDIA/2006-010, Published as a chapter in Approximation Algorithms and Metaheuristics.

- Dorigo, M., and Stützle, T., (2004), “**Ant Colony Optimization**”, Massachusetts Institute of Technology, 2004

- Dowd. P. A., and Onur. A. H., (1993), “Open-pit optimization—part 1: optimal open-pit design”, The Institution of Mining and Metallurgy (Sect. A: Min. industry), pp. 95-104.

- Elevi. B., Dagdelen. K., Salamon. M. D. G., “Single time period production scheduling of open pit mines”, Society for mining, Metallurgy and exploration, INC., TRANSACTION VOL. 290, pp. 1888-1893.

- Glover, F., and Laguna, M., (1997), “**Tabu Search**”, Kluwer Academic Publishers, Boston.

- Hosseinzadeh Aghdam, M., Ghasem-Aghaee N., Basiri, M. E., (2009), “Text feature selection using ant colony optimization”, **Expert Systems with Applications**, Vol. 36, 2009, pp. 6843–6853, ELSEVIER

- Johnson, T. B. and Sharp , W., (1971), “a three-dimensional dynamic programming method for optimal pit design”, US Bureau of mines report on investigations, No. 7553.

- Kim, G., Young., (1978), “Ultimate Pit Limit Design Methodologies Using Computer Models – The State of the Art”, **AIME Transactions**, Vol. 264, pp. 1454-1459.

- Kuo-Ling Huang, Ching-Jong Liao, (2008), “Ant colony optimization combined with taboo search for the job shop scheduling problem”, **Computers & Operations Research**, Vol. 35, pp. 1030-1046.

- Lerchs. Helmut., and Grossman. Ingo. F., (1965), “Optimum Design of Open-Pit Mines”, Joint C.O.R.S. and O.R.S.A. Conference, Montreal, The Canadian Mining and Metallurgical Bulletin, May 1964. Transaction, C.I.M., Volume LXVII, 1965, pp. 17-24.

- Marinakis, Y., Marinaki, M., Doumpos, M., Zopounidis, C., (2009), “Ant colony and particle swarm optimization for financial classification problems”, **Expert Systems with Applications**, Vol. 36, pp. 10604-10611.

- Martin Schlüter, M., Egea, J. A., Banga, J. R., (2009) “Extended ant colony optimization for non-convex mixed integer nonlinear programming”, **Computers & Operations Research**, Vol. 36, pp. 2217-2229.

- MSDN Library, October 2000, Microsoft corporation.

- Mullen, R. J., Monekosso, D., Barman, S., Remagnino, P., (2009), “A review of ant algorithms” **Expert Systems with Applications**, Vol. 36, pp. 9608-9617.

- Onur. A. H. and Dowd. P. A., (1993), “Open-pit optimization—part 2: production scheduling and inclusion of roadways”, The Institution of Mining and Metallurgy (Sect. A: Min. industry), pp.105-113.

- Cordón, O., Herrera, F., Stützle, T., (2002), “A review on the Ant Colony Optimization Metaheuristic: Basics, Models and New Trends”, **Mathware & Soft Computing**, Vol. 9, pp. 1-35.

- Ovanic, J., and Young. Dae S., (1999), “Economic Optimization of Open Stope Geometry”, APCOM, 1999, pp. 855-862.
- Roditis. Y.S., (1993), “Beyond open pit optimization planning, scheduling and sensitivity analysis”, Society for mining, metallurgy, and exploration, INC., SME Annual Meeting, Renno, Nevada.
- Shelokar, P. S., Jayaraman, V. K., Kulkarni, B. D., (2004), “An ant colony classifier system: application to some process engineering problems”, **Computers and Chemical Engineering**, Vol. 28, pp. 1577-1584.
- WHITTLE. J., (1988), “BEYOUND OPTIMIZATION IN OPEN PIT DESIGN”, First Canadian Conference on Computer Applications in the Mineral Industry, pp. 331-337
- Wright E. Alphia., (1986), “Dynamic programming in open pit mining sequence planning-a case study”, 21st APCOM PROCEEDINGS, pp. 415-422.
- Wright E. Alphia., (2000), “MOVING CONE II- A Simple Algorithm for Optimum Pit Limits Design”, Proceeding of the 28th symposium on the application of the computer and operation research in the mineral industries (APCOM), 2000, Colorado USA, pp. 367-374.
- Wright, E. A., (1987), “The use of dynamic programming for open pit mine design: Some practical implications”, **Mining Science and Technology**, January, pp. 97-104.
- Yamatomi. J., Mogi. G., Akaike. A., and Yamaguchi. U., (1995), “Selective Extraction Dynamic Cone Algorithm for Three-Dimensional Open Pit Designs”, APCOM 1995 Conference, Brisbane, pp. 267-274.

Abstract

Before extraction of ore by open pit mining, it is necessary to determine the pit limit. Optimum open pit design is one of the most important designs in open pit mining. For estimation or assessment of this outline, economical block model is essential. This model can be achieved after exploration to ore body and creating of geological block model and applying costs and revenue of each block. A number of algorithms have been developed to determine the pit limits. These algorithms can be dividing into two groups, mathematical and heuristically methods. Heuristically methods have an advantage that can obtain solution(s) in a reasonable time. In some cases they can not produce true optimum pit limit. Mathematical methods are able to determine true optimum pit limit. but these algorithms are very complicated and need to considerable time to reach the solution(s). Hence open pit outline problem in this thesis is modeled by ant colony optimization algorithm. A 2-D model of blocks has been modeled and restrictions applied to it. At subsequent stages some techniques employed to solve the problem. A computer program developed (with Visual Basic 6® programming language) to solve the problem in a short time. With application of ant colony algorithm on problem, specified that this algorithm have a proper ability to determine the pit limit. Algorithm parameters are sensitive to dimensions of problem and with increasing of these dimensions, the answer can divergent soon. In each iteration, the best solution are kept for determination of fond optimum pit.

Key words

Final pit limit, ant colony optimization algorithm, 2-D block model, modeling, goal function

Shahrood University of Technology
Faculty of Mining, Petroleum Engineering and Geophysics

Open Pit Optimization by using of Ant Colony Optimization Method

Ali Azimi

Supervisor:

Dr. Reza Kakaie

Date: 2010, Autumn