فصلنامه علمی ـ پژوهشی مطالعات مدیریت صنعتی سال دهم، شماره ۲۷، زمستان ۹۱
صفحات ۳۹ تا ٦۲

رویکرد الگوریتم فرا ابتکاری کلونی زنبور عسل مصنوعی برای تعیین مکان بهینه سوئیچ‌ها در شبکه ارتباطی تلفن همراه

سید محمد علی خاتمی فیروزآبادی
امین وفادار نیکجو

چکیده

در این تحقیق برای حل مسئله‌ی تخصیص سلول به سوئیچ (CTSAP)، از الگوریتم فراابتکاری کلونی زنبور عسل مصنوعی (ABC) استفاده شده است. هدف مسئله، تخصیص بهینه سلول‌ها به سوئیچ‌ها با حداقل هزینه است. در این تحقیق هزینه از دو جزء تشکیل یافته است. یکی هزینه‌ی تعویض‌ها که مربوط به دو سوئیچ است و دیگری هزینه‌ی اتصال می‌باشد. ظرفیت پاسخگویی تماس هر سوئیچ نیز محدود است و فرض می‌شود همه‌ی سوئیچ‌ها ظرفیت برابری داشته باشند. در مدل این پژوهش هر سلول باید فقط و فقط تنها به یک سوئیچ متصل گردد (single homed). مدل ریاضی این تحقیق، غیرخطی صفر و یک است.

کد رایانه‌ای الگوریتم با نرم‌افزار MATLAB 7.8.0 نوشته شده است. پس از تعیین مقادیر پارامترهای مدل و تأیید صحت عملکرد کد و تنظیم پارامترهای کنترل، کارایی الگوریتم با ایجاد مسائل آزمایشی، با یکی از بهترین الگوریتم‌های فراابتکاری در حل مسئله CTSAP یعنی الگوریتم بهینه سازی کلونی مورچگان (ACO) مقایسه شده است و نتایج نشان می‌دهد که الگوریتم ABC در قیاس با ACO عملکرد رضایت‌بخشی دارد.

واژگان کلیدی: مسئله تخصیص سلول به سوئیچ، الگوریتم فراابتکاری، شبکه‌های تلفن همراه، الگوریتم کلونی زنبور عسل مصنوعی

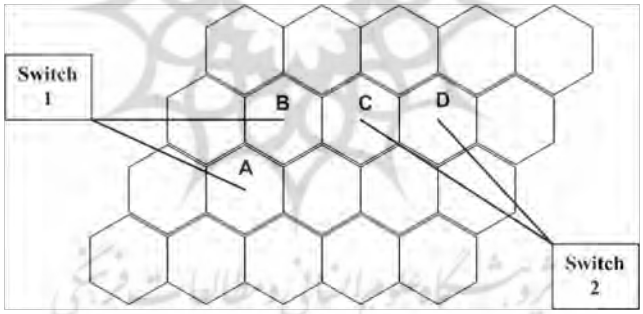
استادیار گروه مدیریت صنعتی دانشگاه علامه طباطبائی، نویسنده مسئول Email: smakhf@hotmail.com
کارشناس ارشد مدیریت صنعتی، دانشگاه علامه طباطبائی

تاریخ دریافت: ۹۰/۱۰/۳ **تاریخ پذیرش: ۹۱/۹/۱**

مقدمه

زندگی در عصر ارتباطات و اطلاعات بدون استفاده از ابزارهای مرتبط و مناسب آن مشکل و حتی غیرممکن است؛ یکی از این ابزارهای ارتباطی تلفن همراه است. امروزه با گسترش استفاده از تلفن همراه وجود سیستم های خدمات دهنده‌ی ارتباطی مناسب که بتواند رضایت مشتریان را کسب و حفظ نماید ضروری است. یکی از راه های پاسخگویی به این نیاز رو به افزایش مشتریان خدمات تلفن همراه، طراحی مناسب شبکه‌های تلفن همراه است. در این نوع طراحی‌ها مسائلی نظیر تعیین مکان تجهیزات مثل آنتن‌های BTS[1]، سوئیچ[2]ها و ... وجود دارد.

در شبکه‌های تلفن همراه، آنتن‌های BTS وجود دارند که شامل تجهیزاتی برای انتقال و دریافت سیگنال‌های رادیویی و رمزگذاری و رمزگشایی ارتباطات با سوئیچ‌ها یا همان BSC[3]ها هستند. هر منطقه‌ی تحت پوشش یک BTS با شکل یک شش ضلعی سلول[4] نامیده می‌شود. وظیفه‌ی هر سوئیچ، مدیریت سیگنال‌های رادیویی یک یا چند سلول می‌باشد مثل تخصیص فرکانس یا تعویض سیگنال‌ها بین BTSها (handoff) [۱].



شکل ۱. تخصیص سلول‌ها به سوئیچ‌ها

در شکل ۱ سلول‌های A و B به سوئیچ ۱ متصل هستند و سلول‌های C و D به سوئیچ ۲ اتصال دارند. تصور کنید مشترکی در حال حاضر با شخصی در حال مکالمه است و تماس از طریق سلول B و سوئیچ ۱ انتقال می‌یابد. همچنین فرض کنید قدرت-های سیگنال در سلول‌های A و C به حد مشخصی رسیده‌اند که مورد توجه قرار گیرند.

1- Base Transceiver Station
2- switch
3- Base Station Controller
4- cell

از آنجایی که سلول C مجاور سلول B می‌باشد، اطلاعات قدرت سیگنال از سلول C به سوئیچ ۱ از طریق شبکه‌ی سیگنال‌دهی داده می‌شود (در شکل ۱ نشان داده نشده است). از این رو سوئیچ ۱ قادر است تمامی اطلاعات قدرت سیگنال در مورد این تماس را دریافت کند و می‌تواند متوجه شود که کدام سلول دارای بالاترین قدرت سیگنال است. اگر مشترک از سلول B به سلول A حرکت کند، سوئیچ ۱، تعویض سیگنال را برای این تماس انجام می‌دهد. این تعویض نسبتاً ساده است و شامل هیچگونه بهنگام سازی مکانی در پایگاه‌های داده که موقعیت مشترک شبکه را ثبت می‌کنند نمی‌شود. حال فرض کنید که مشترک از سلول B به سلول C حرکت کند. در این صورت تعویض شامل انجام یک داده‌ساماندهی نسبتاً پیچیده بین سوئیچ‌های ۱ و ۲ می‌شود. افزون بر آن، مکان مشترک شبکه در پایگاه‌های داده بایستی بهنگام شود. از اینرو دو نوع تعویض وجود دارد، یکی شامل تنها یک سوئیچ می‌شود و دیگری که شامل دو سوئیچ می‌گردد [۲].

با ارجاع به شکل ۱، فرض کنید که می‌دانیم فرکانس تعویض‌ها بین سلول‌های A و B خیلی بالا است و بین B و C چندان بالا نیست. در این صورت این فکر مناسب خواهد بود که سلول‌های A و B به سوئیچ یکسانی (در صورت امکان) متصل باشند تا اینکه هزینه تعویض‌ها کاهش یابد. این انگیزه‌ی ابتدایی برای مسئله‌ی تخصیص سلول می‌باشد [۲].

از جمله اولین نفراتی که بر روی مسئله تخصیص سلول به سوئیچ کار نمودند *مرچانت و سنگوپتا*، (۱۹۹۵) بودند که دو نوع هزینه را در نظر گرفته‌اند یکی هزینه‌ی تعویض بین سلول‌ها و دیگری هزینه‌ی اتصال (cabling یا trunking) بین سلول و سوئیچ متناظرش [۲]. محدودیت موجود نیز حجم تماسی است که هر سوئیچ می‌تواند مدیریت نماید. این مسئله دقیقاً به شکل یک مسئله برنامه‌ریزی عدد صحیح فرمول بندی شده است و همچنین الگوریتمی ابتکاری برای آن ارائه شده که نشان می‌دهد بسیار خوب عمل می‌نماید.

از جمله الگوریتم‌های فراابتکاری که در گذشته برای حل مسئله تخصیص سلول به سوئیچ مطرح شده می‌توان به این موارد اشاره نمود: الگوریتم‌های ژنتیک ترکیبی که مدل پیشنهادی بر مبنای مسئله p-median دارای ظرفیت ارائه گردیده است و مسئله به

دو زیرمسئله تقسیم گردیده است: انتخاب بهترین مجموعه‌ی سوئیچ‌ها و مسئله‌ی تخصیص ترمینال به منظور ارزیابی هر انتخاب از سوئیچ‌ها. الگوریتم‌های ژنتیک ترکیبی نیز برای حل مسئله توسط یک الگوریتم ژنتیک متعارف با جستجویی محدود و چندین الگوریتم ابتکاری جستجوی محلی شکل داده شده‌اند. در واقع تمامی روش‌های ابتکاری ارائه شده شامل یک الگوریتم ژنتیک معمولی برای انتخاب گره‌ی قرارگیری سوئیچ در شبکه بودند که با ترکیب با روش‌های ابتکاری محلی امکان تعیین مقدار تابع هدف متناظر فراهم می‌شد. آزمایشهای انجام شده، نشان دادند که روش‌های ابتکاری ژنتیک ترکیبی عملکرد خوبی برای حل مسئله مکان‌یابی سوئیچ[1] داشتند. [۱]، تکنیک بهینه‌سازی ذرات پراکنده صفر و یک اصلاح شده[2] در آن، توابع هزینه بر سه دسته‌ی هزینه‌ی اتصال کل، هزینه تعویض کل وهزینه سوئیچ کل در نظر گرفته شده‌اند. نتایج آزمایشها مقایسه MBPSO با ACO نتایج بهتری را برای MBPSO در قالب صحت و زمان حل برای ترکیبات زیادی از پارامترها نشان داد[۳]، مطالعه‌ی تطبیقی سه الگوریتم ابتکاری ـ جستجوی ممنوع[3]، شبیه سازی تبرید[4] و الگوریتم‌های ژنتیک موازی با مهاجرت[5] هزینه‌ی حاصله توسط این سه الگوریتم بسیار نزدیک به هم می‌باشد. این سه الگوریتم نتایج بهتری در مقایسه با الگوریتم ژنتیک استاندارد فراهم می‌کنند و همیشه جواب‌های شدنی می‌یابند. از نظر زمان‌های پردازش جستجوی ممنوع سریعترین روش است. نهایتاً، نتایج با کران پایینی از بهینه کلی مقایسه شدند که بر اثربخشی این سه الگوریتم برای این مسئله NP-hard تأکید دارند [۴]، رویکردهای تکاملی برای حل مسئله، مثل الگوریتم‌های ژنتیک کلاسیک، الگوریتم‌های ممتیک[6] و رویکرد چند جمعیتی در این پژوهش نتایج محاسباتی حاصله از آزمایشها گسترده بر اثربخشی این رویکرد جهت ایجاد جواب‌های خوب برای مسائل دارای اندازه مشخص تأکید دارند. در واقع در این پژوهش رویکردی تکاملی (الگوریتم‌های ممتیک) برای حل مسئله ارائه گردید. در هر سری از آزمایشها، الگوریتم‌های ممتیک بهبودی در زمان‌های پردازش و در هزینه‌ها در مقایسه با الگوریتم ژنتیک استاندارد حاصل کردند.

1- Switch Location Problem (SLP)
2- Modified Binary Particle Swarm Optimization (MBPSO)
3- Tabu Search (TS)
4- Simulated Annealing (SA)
5- Parallel Genetic Algorithms with Migrations (PGAM)
6- Memetic Algorithm (MA)

در نهایت نیز، نتایج حاصله با نتایج جستجوی ممنوع و شبیه‌سازی تبرید مقایسه گردید. الگوریتم‌های ممتیک، جستجوی ممنوع و شبیه‌سازی تبرید جواب‌هایی شدنی با هزینه بسیار نزدیک به هم ایجاد نمودند. در کل، نتایج حاصله توسط الگوریتم‌های ممتیک بهتر از نتایج جستجوی ممنوع و شبیه‌سازی تبرید می‌باشند [۵]، الگوریتم‌های ممتیک، در این پژوهش الگوریتم‌های ممتیک مختلفی برای حل مسئله ارائه شدند که نتایج حاصله، کارایی و اثربخشی الگوریتم ممتیک را برای ایجاد جواب‌های خوب در شبکه‌های تلفن همراه اندازه متوسط و بزرگ، در قیاس با الگوریتم ژنتیک استاندارد و دیگر روش‌های ابتکاری معروف در ادبیات موضوع تضمین می‌نماید [۶]، بهینه سازی کلونی مورچگان[1]در این تحقیق الگوریتمی براساس رویکرد ACO برای حل مسئله تخصیص سلول ارائه گردیده است. پژوهشگران مسئله را به صورت مسئله‌ی انطباق در یک گراف دو بخشی جهت‌دار موزون مدل نمودند بطوریکه مورچه‌های مصنوعی می‌توانند مسیرهایی را که متناظر جواب‌های شدنی بر روی گراف است، بسازند. رفتار مورچه‌ها به وسیله‌ی آزمایش نتایج محاسباتی الگوریتم ACO تحت تنظیمات پارامتر مختلف مورد کاوش و تحلیل قرار گرفت. عملکردهای الگوریتم ACO و چندین روش ابتکاری و فراابتکاری شناخته شده در ادبیات موضوع به طور تجربی مطالعه شد. نتایج آزمایشها نشان می‌دهد که الگوریتم ACO پیشنهادی، در مقایسه با اکثر روش‌های ابتکاری و فراابتکاری موجود از لحاظ کیفیت جواب و زمان اجرا، رویکردی رقابتی و اثربخش است و نتایج رضایت‌بخشی ایجاد می‌کند. اگرچه ACO زمان طولانی‌تری در مقایسه با سه روش ابتکاری H, H-II , H-IV می‌گیرد ولی میتواند جواب‌های نزدیک به هم بهتری بیابد. در واقع، زمان موردنیاز ACO هنگام مقایسه با زمان مورد نیاز توسط الگوریتم‌های دقیق، منطقی به نظر می‌رسد. در مقایسه با دیگر روش‌های فراابتکاری مثل شبیه‌سازی تبرید، ممتیک و جستجوی ممنوع، ACO هم از لحاظ کیفیت جواب و هم زمان اجرا در وضعیت بهتری قرار دارد [۷]، جستجوی ممنوع، در این پژوهش در ابتدا یک تساوی ریاضیاتی جدید بین مسئله تخصیص و مسئله مکان‌یابی دارای p قطب ثابت مشهور برقرار کردند و از آن به عنوان پایه‌ای برای روش ابتکاری، جواب‌های ابتدایی تولید نمودند. جهت ارزیابی عملکرد

1- Ant Colony Optimization (ACO)

این رویکرد، دو حد پایین برای بهینه‌ی کلی تعریف شد که به عنوان مرجع برای ارزیابی کیفیت جواب‌های کسب شده به کار رفتند. آزمایشها محاسباتی نشان می‌دهند که این رویکرد معمولاً جواب‌های بهتری مخصوصاً برای شبکه‌های دارای مقیاس بزرگ با بیش از ۵۰ سلول و ۳ سوئیچ در مقایسه با دیگر روش‌های ابتکاری یافت شده در ادبیات ایجاد می‌کند [۸]،ترکیب تکنیک بهینه سازی محلی k-opt با ACO ، در مقایسه با تحقیقات گذشته ، این روش با در نظر گرفتن کیفیت جواب‌های ایجاد شده و هم زمان اجرای الگوریتم‌ها، نسبتاً کارا است. جواب‌های ایجاد شده توسط بهترین گونه این الگوریتم، تنها ۱٪ هزینه‌اش بیشتر از روش‌هایی است که بهترین جواب‌ها را ارائه نموده‌اند امّا جواب‌های این الگوریتم دارای زمان‌های اجرای کوتاه تری هستند [۹]، رویکرد آمیخته‌ای از جستجوی محلی تکراری (2-opt و 3-opt) و شبیه‌سازی تبرید، در این مطالعه روش‌های جستجوی محلی 2-opt و 3-opt به کار گرفته شده‌اند تا عملکرد زمان پردازش الگوریتم ترکیبی ارائه شده را بهبود دهند. نتایج تجربی نشان می‌دهند که روش ابتکاری ارائه شده نتایجی نزدیک به جواب بهینه همان طور که انتظار می‌رفت، تولید می‌کنند [۱۰] و رویکردی ترکیبی از یک شبکه عصبی Hopfield صفر و یک با الگوریتم ژنتیک، در آن شبکه‌ی Hopfield[1] محدودیت‌های مسئله را حل می‌کند و جواب شدنی ایجاد می‌کند و سپس یک الگوریتم ژنتیک به دنبال جواب‌های با کیفیت بالا با حداقل هزینه‌های تعویض و اتصال (کابل کشی) می‌گردد. تازگی این رویکرد ترکیب HNN و الگوریتم ژنتیک[2] و شکل دادن الگوریتم ترکیبی HNN-GA می‌باشد. دو الگوریتم ژنتیک با روش کدگذاری مختلف ارائه شده‌اند تا با HNN ترکیب گردند. این رویکرد بر روی مجموعه‌ای از نمونه مسائل تخصیص مورد آزمایش قرار گرفت و نتایج با GA دارای تابع جریمه و الگوریتم ممتیک شکل داده شده توسط یک GA و روش ابتکاری جستجوی ممنوع مورد مقایسه قرار گرفت. در نتیجه این الگوریتم قادر بود جواب‌های خیلی خوبی از نظر تعداد جواب‌های شدنی ایجاد شده و حداقل هزینه‌ی یافت شده در نمونه‌های آزمایشی در نظر گرفته شده، بدست آورد و نیز قادر به بهبود نتایج حاصله توسط دیگر تکنیک‌های در نظر گرفته شده بود [۱۱].

1- Hopfield Neural Network (HNN)
2- Genetic Algorithm (GA)

از آنجایی که استفاده از سیستم ارتباطی تلفن همراه امروزه بسیار رواج یافته است، ضروری است که بتوان سیستمی را ارائه نمود که هزینه‌ی کمتری را برای ارائه دهنده‌ی این نوع خدمات در بر داشته باشد و نیز رضایت مشتریان را افزایش دهد یکی از راه‌های دستیابی به این هدف استقرار سوئیچ‌ها در شبکه تلفن همراه با حداقل هزینه است که با حل مدل ریاضی مربوطه به وسیله الگوریتم ارائه شده در این پژوهش سعی به انجام این مهم داریم.

در تحقیق حاضر از روش فراابتکاری الگوریتم کلونی زنبور عسل مصنوعی[1] استفاده می‌شود از میان الگوریتم‌های فراابتکاری نسبتاً نوینی که تاکنون برای حل CTSAP استفاده شده است می‌توان PSO و ACO را نام برد و از آنجایی که الگوریتم ABC یکی از الگوریتم‌هایی است که در سال ۲۰۰۵ توسط کارابوگا[2] از دانشکده مهندسی کامپیوتر دانشگاه Erciyes ترکیه ابداع گردیده و در مجموعه‌ی الگوریتم‌های بهینه‌سازی بر مبنای جمعیت[3] قرار دارد و همانطور که از مطالعه‌ی ادبیات تحقیق نیز مشاهده می‌شود اکثر روش‌های حل استفاده شده از مجموعه SOA هستند مثل GA، ACO و PSO از این رو برای حل این مسئله، ABC مدنظر قرار گرفت. دلیل اینکه چرا از میان الگوریتم‌های مختلف زنبور عسل، ABC انتخاب گردید این است که از سادگی و در عین حال شهرت و کاربرد خوبی در میان الگوریتم‌های زنبور برخوردار است. بر این نکته تأکید می‌شود که کارایی یک الگوریتم بستگی بسیاری با نوع و ساختار مسئله دارد و در نتیجه الگوریتمی بسیار موفق در مسئله‌ای خاص می‌تواند در مسئله‌ای دیگر کارایی ضعیفی داشته و بهبود کارایی حاصله، کار بسیار مشکلی باشد پس زمینه‌ی الگوریتم‌های فراابتکاری بسیار تجربی بوده و نمی‌توان به قطعیت قبل از انجام آزمایشها تجربی برتری الگوریتمی خاص را برای حل مسئله‌ای خاص اعلام داشت.

به طور خلاصه می‌توان دلایل زیر را برای انتخاب ABC ذکر کرد:

۱. با توجه به کارایی خوب الگوریتم ACO ارائه شده توسط جیان شیو، لین و هسیائو (۲۰۰۶) برای حل CTSAP[4]، ABC نیز که مانند ACO جزء الگوریتم‌های بهینه‌سازی

1- Artificial Bee Colony (ABC)
2- Karaboga
3- Swarm-based Optimization Algorithms (SOA)
4- Cell To Switch Assignment Problem (CTSAP)

بر مبنای جمعیت است و نیز مانند آن از رفتار زندگی حشره‌ای اجتماعی الهام گرفته شده است، نیز می‌توانست کارایی مطلوبی داشته باشد (قرار گرفتن در مجموعه SOAها)
۲. عدم به کارگیری الگوریتم‌های مختلف زنبور عسل در حل CTSAP تاکنون (کاربردی جدید).
۳. ABC در مقایسه با دیگر الگوریتم‌های به کار گرفته شده برای حل CTSAP، نسبتاً جدید است (نوین بودن).
۴. سادگی منطق به کار رفته در ABC (سادگی پیاده سازی).
بقیه بخش‌های مقاله بدین شکل است که در بخش ۲ مسئله و مدل ریاضی آن معرفی و در بخش ۳ رویکرد ABC برای مسئله مورد نظر توضیح داده می‌شوند. پیاده سازی الگوریتم ABC در نمونه مسائل آزمایشی در بخش ۴ مطرح شده است و در بخش ۵ به ارائه مقایسات محاسباتی و ارزیابی کارایی الگوریتم ارائه شده پرداخته و نهایتاً در بخش ۶ نتایج و پیشنهادات برای تحقیقات آتی عرضه می‌شوند.

تعریف مسئله

مسئله‌ای که مدل ریاضی آن در این تحقیق به کمک ABC حل می‌شود، مسئله‌ی تخصیص سلول به سوئیچ می‌باشد که از نوع مسائل مکان‌یابی P-median دارای ظرفیت یا CPMP[1] می باشد. در این مسئله گروهی از سلول‌ها و گروهی از سوئیچ‌ها را داریم که مکانشان مشخص است. مسئله تخصیص سلول‌ها به سوئیچ‌ها به طریقی بهینه است یعنی تمام تلاش خود را انجام می‌دهیم تا معیار هزینه حداقل گردد. هزینه نیز از دو جزء تشکیل یافته است. یکی هزینه‌ی تعویض‌ها که مربوط به دو سوئیچ است و دیگری هزینه‌ی اتصال. ظرفیت پاسخگویی تماس هر سوئیچ نیز محدود است و همه‌ی سوئیچ‌ها دارای ظرفیت برابری هستند. در مدل این پژوهش هر سلول بایستی تنها به یک سوئیچ متصل گردد (single homed)؛ در دسته‌ای دیگر از مسائل که به آنها dual homed گفته می‌شود الگوهای تماس در زمان‌های مختلف روز می‌توانند متفاوت باشند یعنی یک سلول ممکن است به منظور صرفه‌جویی، در صبح به یک سوئیچ خاص اتصال یابد و در غروب به یکی دیگر متصل شود [۲].

1- Capacitated P-Median Problem

مدل ریاضی [۲]

مکان سلول‌ها و تعداد آنها (n) و نیز تعداد سوئیچ‌ها (m) از قبل مشخص است.

مدل ریاضی مسئله‌ی تخصیص سلول به سوئیچ مورد نظر غیرخطی صفر و یک می‌باشد. که این متغیر صفر و یک به صورت (۱) تعریف می‌گردد. این مدل ریاضی قبلاً در منبع [۲] معرفی شده و مدل جدیدی توسط مؤلفین ارائه نشده است زیرا مدل یک مدل واقعی و کاربردی در مخابرات می‌باشد و نتیجه این پژوهش قابلیت بهره برداری در صنعت مربوطه را دارد.

$$x_{ik} = \begin{cases} 1 & \text{وقتی سلول } i \text{ به سوئیچ } k \text{ تخصیص یابد} \\ 0 & \text{در غیر اینصورت} \end{cases} \qquad (۱)$$

به دلیل آنکه هر سلول بایستی تنها به یک سوئیچ متصل گردد:

$$\sum_{k=1}^{m} x_{ik} = 1 \qquad i = 1, \dots, n \qquad (۲)$$

محدودیت (۳) مربوط به محدودیت ظرفیت هر سوئیچ است که مقدار مشخصی تماس یا به عبارتی سلول را می‌تواند حمایت کند که M_k همان حداکثر تماس‌هایی است که سوئیچ k می‌تواند در واحد زمان پاسخگو باشد (که برای همه سوئیچ‌ها برابر است) و λ_i تعداد تماس‌ها یا ارتباطاتی که سلول i در واحد زمان دارد.

$$\sum_{i=1}^{n} \lambda_i x_{ik} \le M_k \qquad k = 1, \dots, m \qquad (۳)$$

محدودیت (۴) برای لحاظ کردن هزینه‌ی تعویض است. زمانی که دو سلول مختلف به یک سوئیچ خاص متصل شوند، آنگاه y_{ij} مقدار یک می‌گیرد و در غیر اینصورت صفر است.

$$y_{ij} = \sum_{k=1}^{m} x_{ik} x_{jk} \qquad i, j = 1, \dots, \mathrm{n}\ ,\ i \neq j \qquad (۴)$$

اگر y_{ij} مقدار یک بگیرد، هزینه‌ی تعویض در تابع هدف یعنی (۵) صفر می‌شود که در آن h_{ij} [۱] هزینه‌ی تعویض بین سلول i وسلول j می‌باشد.

$$\sum_{i=1}^{n} \sum_{j=1}^{n} h_{ij}(1 - y_{ij}) \qquad (۵)$$

هزینه اتصال نیز طبق (۶) محاسبه می‌شود که در آن c_{ik} [۲]، هزینه‌ی اتصال بین سلول i و سوئیچ k به ازای هر کیلومتر و d_{ik} ، فاصله‌ی بین سلول i و سوئیچ k می‌باشد.

1- handoff cost
2- cabling cost (or trunking cost)

$$\sum_{i=1}^{n}\sum_{k=1}^{m} c_{ik}d_{ik}x_{ik} \quad (۶)$$

در نهایت تابع هدف مدل ریاضی مسئله CTSAP به صورت حداقل سازی (۷) خواهد بود.

$$\sum_{i=1}^{n}\sum_{k=1}^{m} c_{ik}d_{ik}x_{ik} + \sum_{i=1}^{n}\sum_{j=1}^{n} h_{ij}(1-y_{ij}) \quad (۷)$$

رویکرد ABC

نمایش جواب‌ها

برای نمایش جواب‌ها در این الگوریتم از روش کلید تصادفی[1] استفاده شده است. یعنی جواب‌ها به صورت یک ماتریس $(1 \times n)$ می‌باشد که n تعداد سلول‌ها است و هر یک از درایه‌ها عددی اعشاری است که قسمت صحیح آن نشان دهنده‌ی شماره‌ی سوئیچی است که سلول موردنظر با آن در ارتباط است و از بین درایه‌هایی که اعداد صحیح یکسانی دارند، موردی که عدد اعشاری کوچکتری دارد، محل قرارگیری سوئیچ می‌باشد. این طرز نمایش جواب‌ها تاکنون در پژوهشی برای مسئله تخصیص سلول به سوئیچ مشاهده نشده و برای اولین بار بدین منظور ارائه می‌شود تا بتوان از الگوریتم ABC در جهت حل مسئله موردنظر و رسیدن به جواب‌های بهتر استفاده مؤثرتر نمود و در واقع مختص این الگوریتم ارائه گشته است.

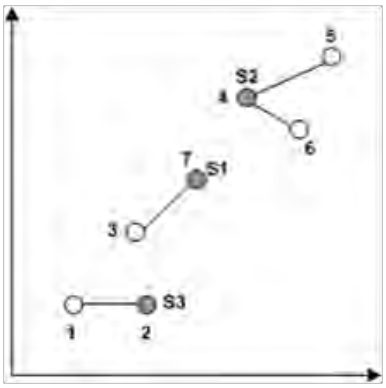
مثلاً به جواب زیر که برای ۷ سلول ($n=۷$) و ۳ سوئیچ ($m=۳$) است دقت کنید:

(۳/۴۵ ۳/۱۵ ۱/۴۱ ۲/۱۲ ۲/۴۹ ۲/۹۰ ۱/۲۳)

سلول‌های ۱ و ۲ به سوئیچ ۳ تخصیص یافته‌اند و سوئیچ ۳ بر روی سلول ۲ قرار دارد. سلول‌های ۳ و ۷ به سوئیچ ۱ تخصیص یافته‌اند و سوئیچ ۱ بر روی سلول ۷ قرار دارد. سلول‌های ۴، ۵ و ۶ به سوئیچ ۲ اختصاص یافته‌اند و سوئیچ ۲ بر روی سلول ۴ مستقر شده است.

شکل ۲ نحوه تخصیص سلول‌ها به سوئیچ‌ها و محل قرارگیری سوئیچ‌ها را برای جواب فوق با فرض نمودن مختصات دلخواه برای سلول‌ها نشان می‌دهد.

1- random key



شکل ۲. مثالی از تخصیص سلول‌ها به سوئیچ‌ها

شرح الگوریتم کلونی زنبور عسل مصنوعی (ABC)

الگوریتم ABC توسط *کارابوگا* برای بهینه‌سازی توابع ریاضی معرفی گردید. هر جواب (یعنی یک مکان در فضای جستجو) نشان دهنده‌ی یک منطقه‌ی غذایی بالقوه می‌باشد و کیفیت جواب معادل است با کیفیت آن منبع غذا. عامل‌ها (زنبورهای عسل مصنوعی) به جستجو و بهره‌برداری از منابع غذایی در فضای جستجو می‌پردازند [۱۲]. ABC از سه نوع عامل استفاده می‌کند: زنبورهای کارگر[1] (EB)، زنبورهای ناظر[2] (OB) و زنبورهای کاشف[3]. زنبورهای EB با جواب‌های فعلی الگوریتم در ارتباط هستند. در هر گام الگوریتم، یک EB تلاش می‌کند تا جوابی را که ارائه می‌نماید، به وسیله‌ی گام جستجوی محلی بهبود دهد، بعد از آن تلاش خواهد نمود تا زنبورهای OB را برای مکان فعلی خود به کار بگیرد. زنبورهای OB از میان مکان‌های بهبود یافته طبق کیفیت آن مکان‌ها انتخاب می‌کنند یعنی جواب‌های بهتر، زنبورهای OB بیشتری به خود جذب می‌کنند. اگر زنبور OB که به کار گماشته شده بود، قادر بود که مکان بهتری را بیابد، زنبور EB مکان خود را بهنگام می‌کند در غیر این صورت در مکان فعلی خود باقی می‌ماند. افزون بر آن، یک زنبور EB مکان خود را ترک خواهد کرد اگر قادر نبوده باشد مکان خود را در تعداد معینی گام بهبود دهد (به این تعداد گام، «حد[4]» گفته می‌شود). اگر یک زنبور EB مکان خود را ترک کند به زنبور کاشف تبدیل می‌شود

1- Employed Bees
2- Onlooker Bees
3- scouts
4- limit

یعنی مکانی جدید به طور تصادفی در فضای جستجو انتخاب می‌کند [۱۲].

گام‌های اصلی الگوریتم به قرار زیر می‌باشند [۱۳]:

۱. ایجاد جمعیت اولیه

۲. تکرار

۳. استقرار زنبورهای EB بر روی منابع غذایی خود

۴. استقرار زنبورهای OB بر روی منابع غذایی بر حسب مقدار شهد آنها

۵. فرستادن زنبورهای کاشف به فضای جستجو برای یافتن منابع غذایی جدید

۶. به یاد سپردن بهترین منبع غذایی یافت شده تاکنون

۷. تا شرایط توقف برقرار شوند.

در گام اول، ABC جمعیتی اولیه به طور تصادفی تولید می‌کند یعنی SN جواب ایجاد می‌کند که SN بیانگر تعداد زنبورهای EB یا OB می‌باشد. هر جواب x_i که $i = 1,2,\dots,SN$ می‌باشد، یک بردار D بعدی است که D تعداد پارامترهای بهینه سازی است. بعد از این مرحله، جمعیت موقعیت‌ها (جواب‌ها) در معرض سیکل‌های تکراری $C = 1,2,\dots,MCN$ فرآیندهای جستجوی زنبورهای EB، OB و زنبورهای کاشف قرار می‌گیرند [۱۳].

یک زنبور OB مصنوعی منبع غذا را بسته به مقدار احتمال مرتبط با آن منبع غذا انتخاب می‌کند، p_i از رابطه‌ی (۸) محاسبه می‌گردد:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad (۸)$$

که fit_i مقدار برازندگی جواب i می‌باشد که نسبتی است از حجم شهد آن منبع غذا در مکان i و SN تعداد منابع غذا می‌باشد که برابراست با تعداد زنبورهای EB یا OB [۱۳].

ABC به منظور ایجاد یک موقعیت غذای کاندید از مکان غذای قدیمی، از رابطه‌ی (۹) بهره می‌برد.

$$V_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \qquad (۹)$$

$k \in \{1,2,\dots,SN\}$ و $j \in \{1,2,\dots,D\}$ به طور تصادفی انتخاب می‌گردند. اگرچه k به طور تصادفی تعیین می‌گردد ولی بایستی با i متفاوت باشد. ϕ_{ij} عددی تصادفی بین $[-1,1]$ می‌باشد. اگر مقدار پارامتر تولیدی از این طریق از حد از پیش تعیین شده‌اش فراتر رود، پارامتر می‌تواند مقدار قابل پذیرشی به خود بگیرد مثلاً می‌تواند مقدار حد را به خود بگیرد [۱۳].

فرض کنید منبع ترک شده x_i باشد و $j \in \{1,2,\dots,D\}$ در آن صورت زنبورهای کاشف منبع غذای جدیدی را که جایگزین x_i می‌شود را می‌یابند. این عمل می‌تواند طبق معادله (۱۰) تعریف گردد [۱۳].

$$x_i^j = x_{\min}^j + \text{rand}[0,1](x_{\max}^j - x_{\min}^j) \qquad (۱۰)$$

شبه کد الگوریتم ABC به قرار زیر است [۱۴]:

۱. بارگذاری نمونه‌های آزمایشی
۲. ایجاد جواب اولیه x_i که $i = 1,2,\dots,SN$
۳. ارزیابی برازندگی (fit_i) جمعیت
۴. سیکل را برابر ۱ قرار بده
۵. تکرار
۶. **برای** هر زنبور کارگر {جوابی جدید طبق (۹) ایجاد کن، مقدار fit_i را محاسبه و فرآیند انتخاب را اعمال کن}
۷. مقدار احتمال p_i برای جواب‌ها را طبق (۸) محاسبه کن
۸. برای هر زنبور ناظر {جوابی را طبق p_i انتخاب و جواب جدیدی تولید کن و مقدار fit_i آن را محاسبه و فرآیند انتخاب را اعمال کن}
۹. اگر جواب ترک‌شده‌ای برای زنبور کاشف وجود دارد،
آن را با جواب جدیدی که به طور تصادفی طبق (۱۰) تولید می‌شود جایگزین کن
۱۰. بهترین جواب یافت شده را ذخیره کن
۱۱. به مقدار قبلی سیکل یکی اضافه کن
۱۲. تا سیکل برابر MCN گردد

در شکل ۳، فلوچارت نحوه‌ی عملکرد الگوریتم ABC به کارگرفته شده توسط محققین پژوهش حاضر، ارائه شده است.

پیاده سازی الگوریتم ABC

تولید مقادیر برای پارامترهای مدل ریاضی

برای اینکه بتوان مدل را در نمونه‌هایی آزمایشی حل نمود بایستی برای پارامترهای مدل ریاضی، مقادیری مشخص گردند. که این مقادیر براساس مقاله [۸]، انتخاب

گردیده‌اند که به طور خلاصه در جدول ۱ آورده شده‌اند.

جدول۱. پارامترهای مدل و مقدار آنها

پارامتر مدل	مقدار
$\boldsymbol{\lambda_i}$	Gamma(1,1)
$\boldsymbol{M_k}$	$\frac{\mathbf{1}}{\mathbf{m}}\left(\mathbf{1}+\frac{\boldsymbol{K}}{\mathbf{100}}\right)\sum_{\mathbf{i-1}}^{\mathbf{n}}\boldsymbol{\lambda_i}$
$\boldsymbol{h_{ij}}$	$\boldsymbol{\lambda_i \times r_{ij}}$
$\boldsymbol{r_{ij}}$	[0,1]
$\boldsymbol{c_{ik}}$	1

در تعیین مقدار M_k ، برای تضمین ۱۰ الی ۵۰ درصد ظرفیت اضافی برای هر سوئیچ، K به طور یکنواخت بین ۱۰ و ۵۰ انتخاب می‌شود. در تعیین مقدار h_{ij} ، به r_{ij} احتمال تعویض[1] گفته می‌شود و طبیعتاً عددی بین صفر و یک است. مثلاً برای تعیین r_{ij} اگر از سلول i کلاً به ۴ سلول دیگر بتوان رفت (یعنی هزینه تعویض از i تا هر کدام از این چهار سلول، صفر نیست)، ۵ عدد تصادفی بین صفر و یک ایجاد می‌شود و برای اینکه مجموع اعداد یا احتمالات یک شود، هر کدام تقسیم بر مجموع می‌شوند و j امین احتمال (j می‌تواند ۱ تا ۴ باشد) به عنوان احتمال تعویض بین سلول i و j (r_{ij}) انتخاب می‌شود. دقت شود پنجمین عدد احتمال عدم تعویض بین دو سلول i و j می‌باشد. مثلاً در شکل ۲ هزینه‌ی تعویض ارتباط بین سلول‌های (۴، ۵ و ۶) ، (۳ و ۷) ، (۱ و ۲) صفر می‌باشد.

پرتال جامع علوم انسانی

1- handoff probability

شکل ۳. فلوچارت نحوه‌ی عملکرد الگوریتم ABC ارائه شده

تنظیم پارامترهای کنترل (عملگرهای) الگوریتم ABC

با بررسی‌های تجربی به عمل آمده برای سه پارامتر *SN*، *MCN* و *LS* سطوح طبق جدول ۲ در نظر گرفته شد و برای انتخاب مقدار مناسب برای هر پارامتر آزمایش عاملی کامل[1] بر روی دو نمونه مسئله انجام شد.

1- full factorial experiment

جدول۲. سطوح تنظیم پارامتر

جواب‌ها[1] *(SN)*	۲۰	٤۰	٦۰
حداکثر تعداد سیکل[2] *(MCN)*	۱۰۰	۲۰۰	۳۰۰
جستجوی محلی[3] *(LS)*	۵	۱۰	۲۵

جدول ۳. ۲۷ سناریو آزمایش عاملی کامل

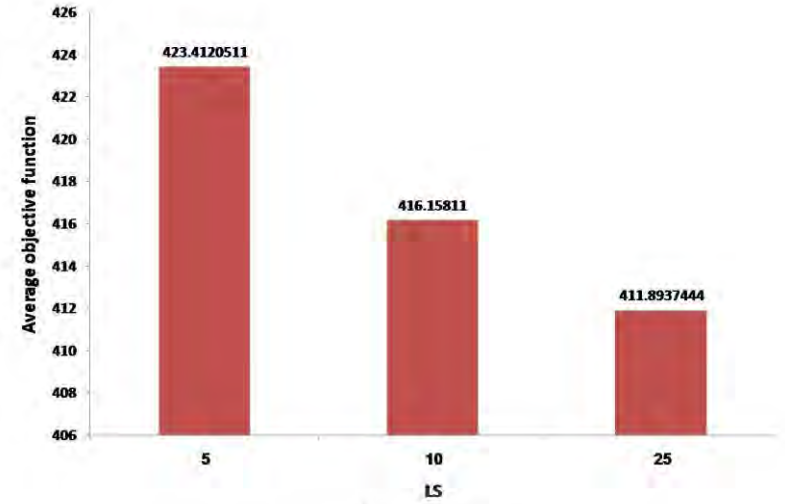
سناریو	SN	MCN	LS	سناریو	SN	MCN	LS
۱	۲۰	۱۰۰	۵	۱۵	٤۰	۲۰۰	۲۵
۲	۲۰	۱۰۰	۱۰	۱٦	٤۰	۳۰۰	۵
۳	۲۰	۱۰۰	۲۵	۱۷	٤۰	۳۰۰	۱۰
٤	۲۰	۲۰۰	۵	۱۸	٤۰	۳۰۰	۲۵
۵	۲۰	۲۰۰	۱۰	۱۹	٦۰	۱۰۰	۵
٦	۲۰	۲۰۰	۲۵	۲۰	٦۰	۱۰۰	۱۰
۷	۲۰	۳۰۰	۵	۲۱	٦۰	۱۰۰	۲۵
۸	۲۰	۳۰۰	۱۰	۲۲	٦۰	۲۰۰	۵
۹	۲۰	۳۰۰	۲۵	۲۳	٦۰	۲۰۰	۱۰
۱۰	٤۰	۱۰۰	۵	۲٤	٦۰	۲۰۰	۲۵
۱۱	٤۰	۱۰۰	۱۰	۲۵	٦۰	۳۰۰	۵
۱۲	٤۰	۱۰۰	۲۵	۲٦	٦۰	۳۰۰	۱۰
۱۳	٤۰	۲۰۰	۵	۲۷	٦۰	۳۰۰	۲۵
۱٤	٤۰	۲۰۰	۱۰				

جدول ٤. نمونه مسائل در نظر گرفته شده برای تنظیم پارامتر

شماره	(n) تعداد سلول‌ها	(m) تعداد سوئیچ‌ها	شبکه
۱	۲۰	۵	۱۰۰ × ۱۰۰
۲	٤۰	٦	۱۰۰ × ۱۰۰

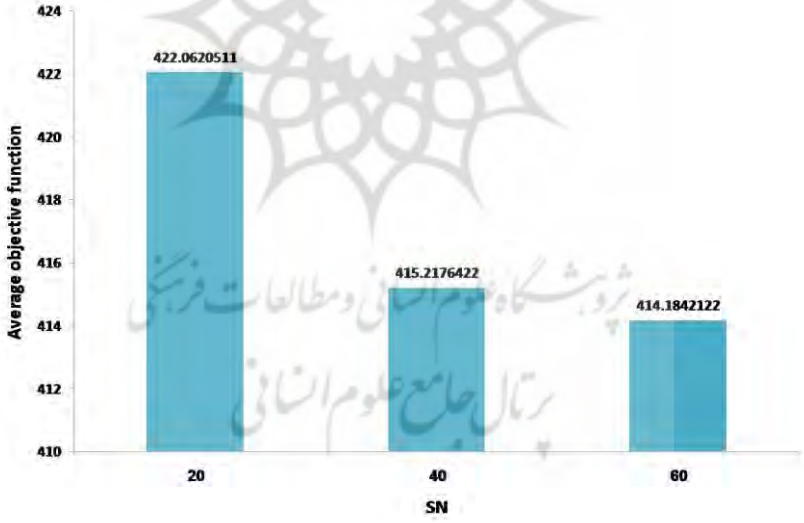
هر یک از مسائل جدول ۴ را با در نظر گرفتن هر یک از ۲۷ سناریوی جدول ۳ مورد ۵ بار اجرا قرار داده و نتایج ثبت گردید. متوسط مقادیر تابع هدف در سناریوهایی که LS=5، LS=10 و LS=25 است، بدست آورده می‌شود که در شکل ۴ قابل مشاهده است.

1- Solution
2- Maximun Cycle Number
3- Local Search



شکل ٤. متوسط مقدار تابع هدف به ازای *LS* های مختلف

متوسط مقادیر تابع هدف در سناریوهایی که *SN*=20، *SN*=40 و *SN*=60 است، بدست آورده می‌شود که در شکل ۵ قابل مشاهده است.

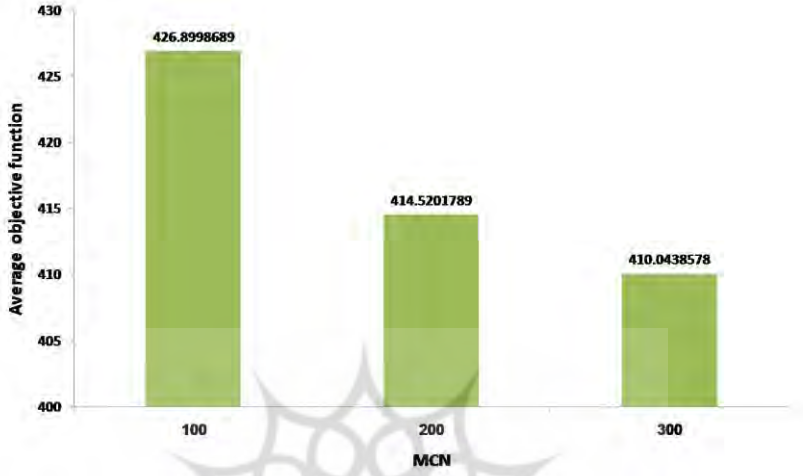


شکل ۵. متوسط مقدار تابع هدف به ازای *SN* های مختلف

با توجه به شکل ۵، از آنجایی که تغییر از ۴۰ به ۶۰ برای SN بهبود اندکی در مقدار میانگین تابع هدف ایجاد کرد و با در نظر گرفتن زمان حل بیشتری که نیاز دارد،

مقدار ۴۰ برای SN انتخاب شد.

متوسط مقادیر تابع هدف در سناریوهایی که MCN=100، MCN=200 و MCN=300 است، بدست آورده می‌شود که در شکل ۶ قابل مشاهده است.



شکل٦. متوسط مقدار تابع هدف به ازای MCN های مختلف

به دلیل اینکه سومین سطح MCN، زمان محاسباتی را افزایش می‌دهد و تفاوت متوسط مقدار تابع هدف در سطوح دوم و سوم چندان چشمگیر نیست، مقدار ۲۰۰ انتخاب گردید. در نتیجه مقادیر جدول ۵ انتخاب می گردند.

جدول ۵. مقادیر منتخب پس از تنظیم پارامتر

SN	MCN	LS
٤٠	۲۰۰	۲۵

ارزیابی کارایی

از آنجایی که الگوریتم ACO ارائه شده توسط *جیان شیو، لین و هسیائو* (۲۰۰۶) [۷]، برای حل مسئله تخصیص سلول از کارایی مناسبی در قیاس با ۶ الگوریتم دیگر (H, H-II , H-IV ، شبیه‌سازی تبرید، ممتیک و جستجوی ممنوع) برخوردار بود و در کمتر مقاله‌ای چنین مقایسه گسترده‌ای انجام شده بود، برای مقایسه با ABC انتخاب گردید.

از دو جهت مقدار تابع هدف (objective function) و سرعت الگوریتم یا زمان پردازش

(CPU time) این مقایسه کارایی انجام می‌شود. بدین منظور ۹ مسئله آزمایشی طراحی گردید که ۳ مسئله کوچک، ۳ مسئله متوسط و ۳ مسئله بزرگ هستند (جدول ۶). نتایج تمامی اجراهای کدهای نرم‌افزار MATLAB 7.8.0 (R2009a) این دو الگوریتم بر روی رایانه‌ای قابل حمل با مشخصات زیر انجام گرفت:

MS-Windows XP Professional (SP3)
Pentium(R) Dual-Core
CPU T4200@ 2.00 GHz 1.20 GHz
2.96 GB of RAM

جدول ٦. مسائل آزمایشی طراحی شده

اندازه	مسئله	سلول‌ها (n)	سوئیچ‌ها (m)	شبکه
کوچک	۱	۷	۳	۱۰۰ × ۱۰۰
	۲	۱۳	٤	۱۰۰ × ۱۰۰
	۳	۲۰	٥	۱۰۰ × ۱۰۰
متوسط	٤	٤٠	٦	۱۰۰ × ۱۰۰
	٥	٥٠	۷	۱۰۰ × ۱۰۰
	٦	٦٠	۸	۱۰۰ × ۱۰۰
بزرگ	۷	۸٠	۱٠	۲۰۰ × ۲۰۰
	۸	۱٠٠	۱۲	۲۰۰ × ۲۰۰
	۹	۱۲٠	۱٤	۲۰۰ × ۲۰۰

اعداد دو ستون ABC و ACO جدول ۷ میانگین ۵ بار اجرا می‌باشد که تمامی اجراها نیز با پارامترهای مدل یکسان انجام شدند. ستون RPD[1] درصد انحراف نسبی برای مقادیر تابع هدف و زمان پردازش ACO در مقایسه با ABC را نشان می‌دهد که از رابطه $\frac{(ACO-ABC)}{ABC} \times 100$ قابل محاسبه می‌باشد. اگر ۰ باشد یعنی اختلافی بین مقدار یا زمان ABC و ACO وجود ندارد و اگر ۱ باشد یعنی ACO ۰/۰۱ برابر ABC یا ۱ ٪ بزرگتر (بدتر) از ABC است و به همین میزان ABC بهتر است. اگر منفی باشد یعنی ACO بهتر است. اعداد ستون‌های ABC و ACO مربوط به زمان پردازش، میانگین ۵ بار ثبت اولین زمانی است که بهترین جواب حاصل شده است.

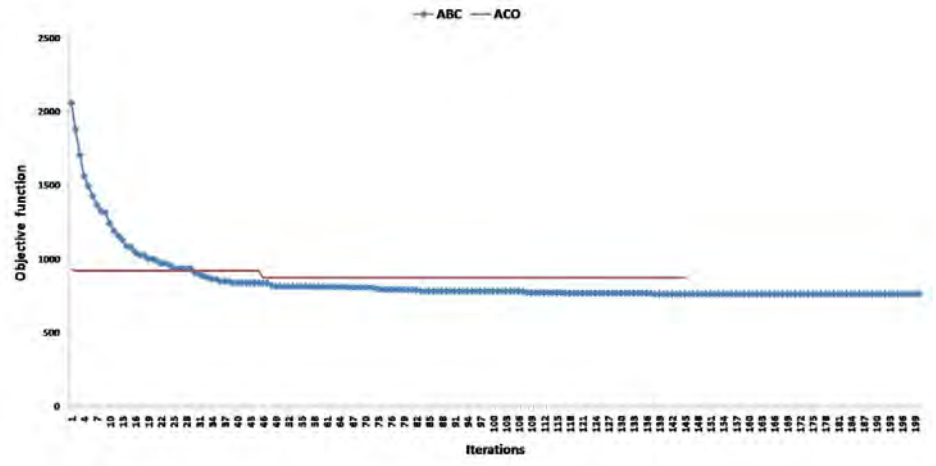
طبق جدول ۷، الگوریتم ABC در ۸ مسئله از ۹ مسئله آزمایشی طراحی شده دارای مقدار تابع هدف بهتری (کمتری) در مقایسه با ACO می‌باشد و در آن ۱ مسئله نیز که

1- Relative Percent Deviation

ACO مقدار تابع هدف بهتری بدست آورده تفاوت بسیار ناچیز است. زمان پردازش (زمانی که بهترین جواب حاصل شده است) نیز میانگین ۵ بار اجرا است. که از این لحاظ الگوریتم ACO بهتر عمل کرده است ولی در تنظیم پارامتر سعی بر آن بوده که تا جایی که ممکن است مقدار تابع هدف کاهش یابد و در واقع مقدار تابع هدف در اولویت توجه بوده است. از طرف دیگر الگوریتم ACO در ابتدا با جوابی نسبتاً خوب تمامی مسائل را شروع به حل می‌کند ولی در ادامه نمی‌تواند جواب را بهبود زیادی بدهد ولی ABC برعکس ACO در ابتدا از جواب آغازین خوبی شروع نمی‌کند ولی در ادامه می‌تواند بهبودهای تدریجی را ایجاد کند که نتیجتاً نیازمند زمان بیشتری برای رسیدن به جواب‌های بهتر می‌باشد. که می‌توان در شکل ۷ که مربوط به مسئله‌ای در اندازه‌ی متوسط با $m = ۸$ و $n = ۶۰$ است، این مطلب را مشاهده کرد. می‌توان طوری پارامترهای الگوریتم را تنظیم نمود تا تعادلی بین مقدار تابع هدف و زمان پردازش ایجاد گردد ولی در مسائل فراابتکاری ابتدا مقدار تابع هدف مورد مقایسه قرار گرفته و در صورت نتیجه نگرفتن به زمان پردازش رجوع می‌شود.

جدول ۷. نتایج مقایسه ABC و ACO

مسئله	تابع هدف			زمان پردازش (ثانیه)		
	ABC	ACO	RPD (%)	ABC	ACO	RPD (%)
۱	٦٨/٤٣	٦٧/٥٩	-١/٢٣	٣٨٥/٧٦	٩/٤٧	-٩٧/٥٤
۲	١٨٠/٩٧	٢١٨/٧٣	٢٠/٨٧	٥٥٠/٧٣	٢٥/٥٧	-٩٥/٣٦
۳	٢٥١/٨٨	٢٦٤/٤٩	٥	٩٣٢/١٣	٤٨/٨٢	-٩٤/٧٦
٤	٥٦١/٣٤	٦٩٦/٠٢	٢٤	٢٢٣٠/٤١	٩٦/٤٨	-٩٥/٦٧
٥	٦٦٧/١٨	٧٢٢/٧٩	٨/٣٤	٣١٨٦/٣٢	٨٢/٩٤	-٩٧/٤٠
٦	٧٥٢/١٨	٨٧٧/٧٣	١٦/٧	٣٥٩٧/١٢	٤٧/٥١	-٩٨/٦٨
۷	١٧٨٥/٤٨	٢١٥٦/٩٨	٢٠/٨١	٥٦١٧/٨٠	٢٤٦/٧١	-٩٥/٦١
۸	٢٠٣١/٤	٢٥٠٤/٩٦	٢٣/٣١	٧٤٤٠/٦٧	٤٧٨/٤٠	-٩٣/٥٧
۹	٢٣٢٨/٧٤	٢٩٢٦/٧٢	٢٥/٦٨	٩٦٥٩/٧٥	٦١١/٤٧	-٩٣/٦٧



شکل ۷. نمودار همگرایی برای $m = ۸$ و $n = ۶۰$

نتیجه‌گیری و پیشنهادها

در این تحقیق برای حل مسئله‌ی تخصیص سلول به سوئیچ که در شبکه‌های تلفن همراه کاربرد دارد و از نظر پیچیدگی ثابت شده است که جزء مسائل NP-hard است (روش برنامه‌ریزی عدد صحیح برای یافتن جواب هنگامی که تعداد سلول‌ها بیش از ۳۵ می‌گردد ناتوان است) از الگوریتم فراابتکاری کلونی زنبور عسل مصنوعی استفاده شد. کد رایانه‌ای الگوریتم با نرم‌افزار MATLAB 7.8.0 (R2009a) نوشته شد پس از تخمین پارامترهای مدل و تأیید صحت عملکرد کد و تنظیم پارامترهای کنترلی الگوریتم از طریق ایجاد ۹ مسئله آزمایشی در اندازه‌های کوچک، متوسط و بزرگ، کارایی الگوریتم از لحاظ مقدار تابع هدف در درجه‌ی اول و زمان پردازش در درجه دوم با یکی از بهترین الگوریتم‌های فراابتکاری ارائه شده برای مسئله CTSAP یعنی الگوریتم بهینه سازی کلونی مورچگان مورد ارزیابی قرار گرفت. به دلیل آنکه الگوریتم ACO ارائه شده توسط جیان شیو، لین و هسیائو (۲۰۰۶) [۷]، برای حل مسئله تخصیص سلول از کارایی مناسبی در قیاس با ۶ الگوریتم دیگر (H, H-II , H-IV، شبیه‌سازی تبرید، ممتیک و جستجوی ممنوع) برخوردار بود برای مقایسه با ABC انتخاب گردید و نتایج نشان داد که الگوریتم ABC در قیاس با ACO عملکرد رضایت بخشی دارد.

از جمله پیشنهادهایی که می‌توان برای تحقیقات آتی ارائه کرد عبارتند از:

۱. توجه بیشتر به ساختار و ماهیت مدل ریاضی در طراحی الگوریتم ABC جهت دستیابی به الگوریتمی کاراتر.

۲. توسعه مدل ریاضی مسئله با توجه به نیازهای خاص شبکه تلفن همراه محلی مثل در نظر گرفتن سلول‌ها به صورت dual homed (هر سلول می تواند به دو سوئیچ متصل گردد) و حل آن جهت رفع کمبودها یا بهبود شبکه.

۳. استفاده از الگوریتم ABC ارائه شده در شبکه مخابرات ایران و به کارگیری داده‌های واقعی

۴. تعیین مقادیر پارامترهای مدل ریاضی مسئله براساس روشی جدیدتر و مناسب‌تر با استفاده از طراحی آزمایشها[1].

۵. استفاده از مسائل نمونه استفاده شده توسط الگوریتم مورد مقایسه (ACO) در مقاله مربوطه، و همچنین استفاده از تحلیل‌های گسترده‌تر آماری جهت مقایسه‌ای عادلانه‌تر.

۶. مقایسه‌ی گسترده‌تر الگوریتم ABC با بعضی از الگوریتم‌های فراابتکاری برتر دیگر در حل مسئله CTSAP و ارائه نتیجه‌ای جامع‌تر.

۷. حل CTSAP با الگوریتم‌های نوین دیگر مثل رقابت استعماری، الگوریتم جریان آب و ... و ارائه نتایج.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی

1- DOE

منابع

1. salcedo-Sanz, Sancho; Jose, A. Portilla-Figueras; Emilio G. Ortiz-Garcia; Angel M. Perez-Bellido; Christopher Thraves; Antonio Fernandez-Anta; Xin Yao, **Optimal switch location in mobile communication networks using hybrid genetic algorithms**. Applied Soft Computing, 8, 2008, 1486-1497.
2. Merchant, Arif and Bhaskar Sengupta, **Assignment of Cells to Switches in PCS Networks**. IEEE/ACM Transactions on networking, vol. 3, no. 5, 1995, 521-526.
3. Udgata, Siba K.; U. Anuradha; G. Pavan Kumar; Gauri K. Udgata. **Assignment of Cells to Switches in a Cellular Mobile Environment using Swarm Intelligence**. IEEE computer society, 2008, 189-194.
4. Quintero, Alejandro; Samuel Pierre, **Assigning cells to switches in cellular mobile networks: a comparative study**. Computer Communications, 26, 2003, 950-960.
5. Quintero, Alejandro; Samuel Pierre, **Evolutionary approach to optimize the assignment of cells to switches in personal communication networks**. Computer Communications, 26, 2003, 927-938.
6. Quintero, Alejandro; Samuel Pierre, **Sequential and multi-population memetic algorithms for assigning cells to switches in mobile networks**. Computer Networks, 43, 2003, 247-261.
7. Jian Shyu, Shxyong; B.M.T. Lin; Tsung-Shen Hsiao, **Ant colony optimization for the cell assignment problem in PCS networks**. Computers & Operations Research, 33, 2006, 1713-1740.
8. Pierre, Samuel; Fabien Houeto, **A tabu search approach for assigning cells to switches in cellular mobile networks**. Computer Communications, 25, 2002, 464-477.
9. Fournier, Joseph R.L.; Samuel Pierre, **Assigning cells to switches in mobile networks using an ant colony optimization heuristic**. Computer Communications, 28, 2005, 65-73.
10. Rajalakshmi, K.; Prakash Kumar; Hima M. Bindu, **Hybridizing Iterative Local Search Algorithm for Assigning Cells to Switch in Cellular Mobile Network**. International Journal of Soft Computing, 5(1), 2010, 7-12.
11. Salcedo-Sanz, Sancho; Xin Yao, **Assignment of cells to switches in a cellular mobile network using a hybrid Hopfield network-genetic algorithm approach**. Applied Soft Computing, 8, 2008, 216-224.
12. Panigrahi, Bijaya Ketan; Yuhui Shi; Meng-Hiot Lim, **Handbook of Swarm Intelligence.** Springer-Verlag Berlin Heidelberg, 2011.
13. Karaboga, Dervis; Bahriye Akay, **A comparative study of Artificial Bee Colony algorithm**. Applied Mathematics and Computation**,** 214, 2009, 108-132.
14. Karaboga, Dervis; Celal Ozturk, **A novel clustering approach: Artificial Bee Colony (ABC) algorithm.** Applied Soft Computing, 11, 2011, 652-657.