

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه آزاد اسلامی واحد ملایر

گروه کامپیوتر

عنوان

پایان نامه ارائه شده به مدیریت تحصیلات تکمیلی به عنوان بخشی از فعالیت های

تحصیلی لازم برای اخذ درجه کارشناسی ارشد

در رشته مهندسی نرم افزار

استاد راهنما

استاد مشاور

نگارش

شهریور ۹۵

بسم الله الرحمن الرحيم

توسط:

پایان نامه

ارائه شده به مدیریت تحصیلات تکمیلی به عنوان بخشی از فعالیت های تحصیلی لازم برای اخذ درجه

کارشناسی ارشد

در رشته مهندسی نرم افزار

از

دانشگاه آزاد اسلامی واحد ملایر

ارزیابی و تصویب شده توسط کمیته پایان نامه با نمره و درجه:

دکتر نام و نام خانوادگی استاد راهنما و رئیس کمیته).....استادیار

دکتر نام و نام خانوادگی (استاد مشاور).....استادیار

دکتر نام و نام خانوادگی (استاد مدعو).....استادیار



معاونت پژوهش و فناوری

به نام خدا

نمونه اخلاق پژوهش

بیاداری از تفاوت جهان و اعتقاد به آنکه عالم محض خداست و بخوار و ناظر بر افعال انسان و به منظور پاس داشتن تمام بلند و انش و پژوهش و نظریه است به جایگاه دانشگاه و اعتدای فرهنگ و تمدن بشری و دانشمندان و اصحاب است

علمی و اندیشه‌ای دانشگاه آزاد اسلامی متعهد می‌گردد اصول زیر را در انجام فعالیت‌های پژوهشی به نظر قرار دهد

۱. اصل حقیقت‌جویی: تلاش در راستای پی‌جویی حقیقت و وفاداری به آن و دوری از حرکت پنهان سازی حقیقت.
۲. اصل رعایت حقوق: التزام به رعایت کامل حقوق پژوهشگران و پژوهشگران (انسان، حیوان و نبات و سایر مساجان حق).
۳. اصل مالکیت مادی و معنوی: تعهد کامل به حقوق مادی و معنوی دانشگاه و کلیه به کاران پژوهش.
۴. اصل منع فحش: تعهد به رعایت مصداق علی و در نظر داشتن بهر شیوه و توسعه کشور و کلیه مراحل پژوهش.
۵. اصل رعایت انصاف و انصاف: تعهد به اجتناب از حرکت جانب داری غیر علمی و مخالفت از احوال، تبعیضات و منابع در اختیار.
۶. اصل رازداری: تعهد به صیانت از اسرار و اطلاعات محرمانه افراد، سازمانها و کلیه نهادی مرتبط با تحقیق.
۷. اصل احترام: تعهد به رعایت حریم نام و حرمت و انجام تحقیقات و رعایت جانب آقا و خودداری از حرکت حرمت شکنی.
۸. اصل ترویج: تعهد به رواج دانش و ارائه نتایج تحقیقات و انتقال آن به به کاران علمی و دانشمندان به غیر از مواردی که منع قانونی دارد.
۹. اصل درست: التزام به درست جویی از حرکت رفتار غیر حرفه‌ای و اعلام موضع نسبت به کسانی که حوزه علم و پژوهش را به مثابه وسیله غیر علمی می‌آیند.



تعهد نامه اصالت رساله یا پایان نامه

اینجانب دانش آموخته مقطع کارشناسی ارشد ناپیوسته / دکترای حرفه ای / دکترای تخصصی در رشته که در تاریخ از پایان نامه/رساله خود تحت عنوان «.....»

با کسب نمره و درجه دفاع نموده ام بدینوسیله متعهد می شوم :

۱) این پایان نامه / رساله حاصل تحقیق و پژوهش انجام شده توسط اینجانب بوده و در مواردی که از دستاوردهای علمی و پژوهشی دیگران (اعم از پایان نامه ، کتاب ، مقاله و) استفاده نموده ام، مطابق ضوابط و رویه موجود، نام منبع مورد استفاده و سایر مشخصات آن را در فهرست مربوطه ذکر و درج کرده ام.

۲) این پایان نامه / رساله قبلاً برای دریافت هیچ مدرک تحصیلی (هم سطح، پایین تر یا بالاتر) در سایر دانشگاه ها و موسسات آموزش عالی ارائه نشده است.

۳) چنانچه بعد از فراغت از تحصیل، قصد استفاده و هرگونه بهره برداری اعم از چاپ کتاب، ثبت اختراع و از این پایان نامه داشته باشم، از حوزه معاونت پژوهشی واحد مجوزهای مربوطه را اخذ نمایم.

۴) چنانچه در هر مقطع زمانی خلاف موارد فوق ثابت شود، عواقب ناشی از آن را می پذیرم و واحد دانشگاهی مجاز است با اینجانب مطابق ضوابط و مقررات رفتار نموده و در صورت ابطال مدرک تحصیلی ام هیچگونه ادعایی نخواهم داشت.

نام و نام خانوادگی:

تاریخ و امضا:

اهدا

سپاسگزاری

فهرست مطالب

۱۶	مقدمه	1-1
۱۶	بیان مسئله	1-2
۲۰	اهمیت و ضرورت پژوهش	۳-۱
۲۱	اهداف تحقیق	۴-۱
۲۱	فرضیه‌های پژوهش	۵-۱
۲۲	سازمان‌دهی فصول پایاننامه	۶-۱
۲۳	فصل دوم: مبانی نظری و ادبیات پیشین	۲
۲۴	مقدمه	2-1
۲۴	مبانی نظری	۲-۲
۲۴	۲-۲-۱ رایانش ابری	
۲۶	2-2-2 مراکز داده	
۲۷	۲-۲-۳ مجازی‌سازی	
۲۹	۲-۲-۴ الگوریتم‌های بهینه‌سازی فراابتکاری	
۴۶	مرور ادبیات پیشین	2-3
۴۶	۲-۳-۱ روش‌های غیر فراابتکاری	

۲-۳-۲	روشهای فراابتکاری.....	۴۹
۲-۴	خلاصه فصل.....	۵۲
3	فصل سوم: روش پیشنهادی.....	۵۴
3-1	مقدمه.....	۵۵
۳-۲	طبقه‌بندی مسائل قرار گیری ماشین.....	۵۵
۳-۲-۱	روشهای تک هدفه.....	۵۵
۳-۲-۲	چند هدفه، پاسخ داده شده بصورت تک هدفه.....	۵۶
۳-۲-۳	روشهای چند هدفه‌ی خالص.....	۵۶
۳-۳	انواع روشهای حل مسائل بهینه‌سازی.....	۵۷
۳-۳-۱	روشهای بهینه سازی ترکیبی.....	۵۸
۳-۴	روش پیشنهادی.....	۵۹
۳-۴-۱	حل مسئله با استفاده از الگوریتم کلونی زنبور عسل.....	۵۹
۳-۴-۲	حل مسئله با استفاده از الگوریتم کلونی مورچگان.....	۶۴
۳-۴-۳	الگوریتم پیشنهادی.....	۶۹
4	فصل چهارم: اثبات و ارزیابی.....	۷۲
4-1	مقدمه.....	۷۳
۲-۴	نرم افزار و سخت‌افزارهای شبیه سازی.....	۷۳

۳-۴	معیارهای مقایسه	۷۳
۴-۴	نتایج بدست آمده	۷۴
۴-۴-۱	سناریو اول	۷۵
۴-۴-۲	سناریو دوم	۷۸
۵	فصل پنجم: نتیجه گیری	۸۲
۵-۱	مقدمه	۸۳
۵-۲	نتیجه گیری	۸۳
۵-۳	جمع بندی	۸۶
۶	فصل ششم: کارهای آتی	۸۷
۶-۱	مقدمه	۸۸
۶-۲	بحث	۸۸
6-3	پیشنهاد کارهای آینده	۸۹
7	فصل هفتم: منابع	۹۰

فهرست جداول

جدول ۱-۲ انواع الگوریتم‌های فراابتکاری.....	۳۲
جدول ۱-۳ عناصر الگوریتم زنبور عسل.....	۶۲
جدول ۱-۴ پارامترهای مربوط به الگوریتم ژنتیک.....	۷۴
جدول ۲-۴ پارامترهای مربوط به الگوریتم مورچگان.....	۷۴
جدول ۳-۴ پارامترهای مربوط به الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی.....	۷۵
جدول ۴-۴ پارامترهای مربوط به الگوریتم کلونی زنبور عسل.....	۷۵
جدول ۵-۴ نتایج بدست آمده برای سناریو اول.....	۷۵
جدول ۶-۴ نتایج بدست آمده برای سناریو دوم.....	۷۸
جدول ۱-۵ میانگین نتایج بدست آمده از ۱۰ آزمایش.....	۸۳

فهرست شکل‌ها

- شکل ۱-۲ بهینه‌سازی قرارگیری ماشین‌های مجازی..... ۱۹
- شکل ۱-۳ انواع روش‌های حل مسئله برای مسائل بهینه‌سازی..... ۵۸
- شکل ۲-۳ شمای نمایش جواب..... ۶۰
- شکل ۳-۳ فلوچارت الگوریتم زنبور عسل..... ۶۳
- شکل ۴-۳ فلوچارت الگوریتم کلونی مورچگان..... ۶۹
- شکل ۵-۳ فلوچارت روش پیشنهادی..... ۷۱
- شکل ۱-۴ مقایسه‌ی از دست رفت منابع در سناریو اول..... ۷۶
- شکل ۲-۴ مقایسه‌ی زمان اجرای الگوریتم در سناریو اول..... ۷۷
- شکل ۳-۴ تعداد سروورهای روشن در روش‌های مورد مقایسه در سناریو اول..... ۷۸
- شکل ۴-۴ مقایسه‌ی از دست رفت منابع در سناریو دوم..... ۷۹
- شکل ۵-۴ مقایسه‌ی زمان اجرای الگوریتمها در سناریو دوم..... ۸۰
- شکل ۶-۴ تعداد سروورهای روشن در روش‌های مورد مقایسه در سناریو دوم..... ۸۱
- شکل ۱-۵ مقایسه روش‌های فراابتکاری از نظر میزان از دست رفت منابع..... ۸۴
- شکل ۲-۵ مقایسه روش‌های فراابتکاری از نظر تعداد سروورهای روشن..... ۸۵

شکل ۳-۵ مقایسه روش های فراابتکاری از نظر میانگین زمان تولید مدل ۸۶

چکیده

با توسعه و فراگیر شدن اینترنت، سرویس‌های موجود در آن نیز روزبه‌روز در حال محبوب شدن هستند. یکی از محبوب‌ترین و اصلی‌ترین سرویس‌های موجود در اینترنت، رایانش ابری است. رایانش ابری مدل پردازشی جدید برای ارائه‌ی خدمات در شبکه‌های بزرگ، بخصوص شبکه‌ی اینترنت است. با پیدایش درخواست‌های زیاد برای سرویس‌های ارائه‌کننده‌ی خدمات، مراکز داده‌ی زیادی تاسیس شده‌اند. طبیعی است که بوجود آمدن مراکز داده‌ای مختلف، چالش‌هایی نیز بوجود آمده است که یکی از مهمترین این چالش‌ها نحوه‌ی قرار گیری ماشین‌های مجازی در مراکز داده با کمترین میزان از دست رفت منابع است. مسئله‌ی قرار گیری ماشین‌های مجازی در مراکز داده به نحوه‌ی که حداقل تعداد میزبان‌ها روشن بوده و از قابلیت‌های سرورهای روشن بهترین استفاده شود، تعریف می‌گردد. مسئله‌ی قرار گیری ماشین‌های مجازی در مراکز داده یک مسئله‌ی NP-سخت بوده که نمی‌توان از طریق روش‌های چند جمله‌ای به آن پاسخگویی کرد. به همین دلیل در این مطالعه، یک روش ترکیبی مبتنی بر الگوریتم مورچگان و الگوریتم کلونی زنبور عسل ارائه شد که تابع هدف آن به عنوان از دست رفت منابع معرفی گردید. در روش پیشنهادی، پاسخ‌های تولیدی از طریق الگوریتم کلونی مورچگان به الگوریتم زنبور عسل انتقال یافت و توانست پاسخ‌های بهتری را نسبت به الگوریتم مورچگان، الگوریتم زنبور عسل، الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی و همچنین الگوریتم ژنتیک ایجاد کند.

فصل اول : کلیات موضوع

۱-۱ مقدمه

در این فصل کلیات این پژوهش مورد بررسی قرار خواهد گرفت. این پژوهش جنبه‌های مختلفی از «تعریف ابر» و «مراکز داده» تا «بهینه‌سازی قرارگیری ماشین مجازی^۱ در مراکز داده» را شامل می‌شود. در ابتدای فصل به بیان مسئله‌ی موردپژوهش پرداخته خواهد شد. در ادامه اهداف تحقیق، روش‌ها و فنون اجرایی طرح و سمت و سوی آن و همچنین اهمیت و ارزش تحقیق مورد بررسی قرار می‌گیرد.

۱-۲ بیان مسئله

فناوری‌های اطلاعاتی و ارتباطی ازجمله اینترنت که امروزه جزء جدانشدنی زندگی بشر امروزی شده‌اند، در سال‌های اخیر توسعه‌ی چشمگیری داشته است. نیازهای انسان امروزی نیز متناسب با آن‌ها توسعه‌یافته است. بخصوص در سال‌های اخیر تمرکز بر روی کاهش هزینه‌ها اهمیت بسیار بالایی پیدا کرده است. راه‌حلی که امروز در عرصه فناوری برای چنین مشکلاتی پیشنهاد می‌شود فناوری بانام رایانش ابری^۲ است [1].

شبکه‌ها برای مؤسسات امکان دسترسی به انواع فناوری‌های موجود را فراهم می‌کند. یک شبکه مشارکتی^۳ (تعاونی) درواقع ماهیت سازمانی و فرایندهای تجاری، منابع و روابطی است که تلاشی مشترک را برای نیل به اهداف سازمانی را پشتیبانی می‌کند [2]. مزایای شبکه‌های مشارکتی عبارت‌اند از: سرعت، توسعه، رقابت و بهینه‌سازی منابع و نوآوری؛ بنابراین بسیاری از سازمان‌ها و مراکز تمایل به ایجاد شبکه‌های تعاونی با مراکز مشابه خوددارند تا از مزایای آن بهره گیرند.

^۱ Virtual Machine

^{۲۲} Cloud Computing

^۳ Collaborative Networks

با توجه به افزایش رشد استفاده از نرم افزارهای مبتنی بر مشتری و خدمتگذار و هزینه های مربوط به اختصاص یک سرور فیزیکی جداگانه برای هر یک از سرویس های ارائه شده، استفاده از سرورهای مجازی متداول گردید. به دلیل به وجود آمدن احتمالی ناسازگاری سرویس هایی که ارائه می شوند، در رهیافتی سنتی هر سرویس در میزبان جداگانه ای پیاده سازی می شود. در غیر این صورت و با پیش آمدن ناسازگاری، احتمال ازکارافتادن سیستم و در ادامه قطع سایر سرویس های موجود بر روی یک سرور وجود دارد [3].

در رهیافت سنتی، به دلیل افزایش سرورها، هزینه هایی مثل مصرف برق زیاد، سخت افزار اضافی، تجهیزات خنک کننده در مراکز داده، فضای لازم و نیروی انسانی که وظیفه مدیریت سیستم ها را هم از نظر نرم افزاری و هم سخت افزاری بر عهده دارند، به سازمان تحمیل می شود. مجازی سازی امکان راه اندازی چند ماشین مجازی بر روی یک ماشین میزبان فیزیکی را فراهم می کند و هر یک از این ماشین های مجازی، می توانند سیستم عاملی جداگانه داشته باشند و آن را اجرا کنند. از این رو با استفاده از مجازی سازی، می توان بر روی یک سرور و در یک زمان، تعدادی سیستم عامل را راه اندازی کرد [4].

بنابراین، استفاده از مجازی سازی در مراکز داده ی کنونی امری اجتناب ناپذیر به نظر می رسد [5]. سرورهای کنونی Amazon EC2 هم اکنون دارای هزاران ماشین مجازی در داخل سرورها هستند. هر یک از این ماشین های مجازی در دیتاسنتر، ممکن است که دارای اندازه های متفاوتی باشند. این تنوع در اندازه ممکن است که در توان پردازشی و میزان حافظه ی اصلی سیستم باشد. هر یک از سرورهای موجود در داخل دیتاسنترها، دارای ظرفیت های محدودی برای نگهداری ماشین های مجازی هستند. بنابراین انتظار می رود که چینش صحیح ماشین های مجازی در سرورها، منجر به افزایش میزان بهره وری از سرورها و در نهایت افزایش سود ارائه کنندگان سرویس گردد. قرارگیری صحیح ماشین های مجازی

در سرورها یک مسئله‌ی بهینه‌سازی از درجه‌ی Np-Hard است [4]؛ بنابراین باید بتوان با استفاده از الگوریتم‌های فراابتکاری که هدف آن‌ها بهینه‌سازی است، یک راه‌حل بهینه را برای آن‌ها ایجاد کرد. قرارگیری ماشین‌های مجازی درواقع یک فرآیند نگاشت ماشین‌های مجازی به ماشین‌های فیزیکی است. تخصیص بهینه به این دلیل مهم است که مصرف انرژی را کاهش داده و از منابع موجود استفاده بهتری دارد [۴]. از دید فراهم‌آوردندگان سرویس‌های ابری نیز، هدف حداکثر سازی میزان سود است.

یک الگوریتم ساده برای این کار به این صورت است [6]:

(۱) در ابتدا وضعیت استفاده از منابع سرور برای یک مدت طولانی رصد می‌شود تا شکل درخواست‌ها موردبررسی قرار گیرد و ماشین‌های مجازی موردنیاز را بتوانیم استخراج کنیم.

(۲) سپس یک سرور باقابلیت‌های مجازی‌سازی و ذخیره‌سازی مشترک انتخاب می‌شود.

(۳) اکنون اولین ماشین مجازی را بر روی اولین سرور قرار می‌دهیم. ماشین مجازی دوم را سعی می‌کنیم بر روی همان سرور قرار داده اما اگر گنجایش محاسباتی آن را نداشته باشد، باید یک سرور جدید را انتخاب کرده و ماشین مجازی را در آن قرارداد.

با استفاده از همین الگوریتم، بجای استفاده از ۷ سرور می‌توان سه سرور را داشت که از منابع نیز به نحو بهتری استفاده خواهد گردید (طبق شکل ۱-۱).



شکل ۱-۲ بهینه سازی قرارگیری ماشین های مجازی

در این مطالعه، هدف از الگوریتم ارائه شده حداقل سازی میزان هدر رفت منابع است. در این مطالعه، از آنجایی که با یک مسئله بهینه سازی روبرو هستیم، بهتر است از الگوریتم های فراابتکاری استفاده شود. پیش از این الگوریتم های متعددی از جمله الگوریتم مورچگان^۴ و الگوریتم زنبور عسل^۵ ارائه شده است [7, 8]. در این مطالعه قصد داریم ترکیب این دو را مورد بررسی قرار دهیم. در واقع هدف این تحقیق ارائه یک الگوریتم ترکیبی از الگوریتم زنبور عسل و الگوریتم مورچگان برای بهینه سازی قرارگیری ماشین های مجازی در مراکز داده است.

^۴ Ant Colony Algorithm

^۵ Biogeography-Based Optimization

۳-۱ اهمیت و ضرورت پژوهش

فراگیر شدن روزافزون پردازش ابری و سرویس‌های ابری در دنیای رو به رشد امروزی امری اجتناب‌ناپذیر است. در این میان، شرکت‌های بزرگ متعددی پا به عرصه‌ی سرویس‌های ابری در زمینه‌ی محاسباتی و ذخیره‌سازی عمومی نهاده‌اند که از این میان می‌توان به شرکت آمازون^۶ اشاره کرد که با سرویس EC2^۷ خود بازار رقابتی بسیار داغی را به وجود آورده است. اما باید گفت که سرویس‌های ابری عمومی تنها به شرکت‌های ارائه‌کننده‌ی ابر عمومی محدود نشده و مجموعه‌ای از ده‌ها شرکت بسیار معتبر در این بازار با یکدیگر رقابت دارند [9].

باید اذعان داشت که استفاده سروورهای گران‌قیمت برای همه‌ی افراد و یا حتی سازمان‌های بزرگ ممکن نیست. امروزه برای دستیابی به چنین زیر ساخت‌های قدرتمندی، استفاده از سرویس‌های ابری امری اجتناب‌ناپذیر به نظر می‌آید. فناوری مجازی‌سازی^۸ امری جدانشدنی از محاسبات ابری است به گونه‌ای که بر روی سروورهای موجود در مراکز داده، نیاز به ایجاد ماشین‌های مجازی با توان‌های مختلف، بسته به نیاز مشتریان است؛ بنابراین باید در داخل سروورها ماشین‌های مجازی قرار داده شود g. اکنون باید در نظر داشت که این ماشین‌های مجازی در داخل سروورها باید به گونه‌ای قرار گیرند که از توان محاسباتی سروورها حداکثر استفاده شود. این امر منجر به افزایش سود ارائه‌کنندگان سرویس خواهد بود.

^۶ Amazon

^۷ Elastic Compute Cloud

^۸ Virtualization

افزایش بهره‌وری استفاده از منابع منجر به کاهش هزینه‌های کاربران و ارائه‌کنندگان سرویس می‌شود و در نتیجه می‌تواند منجر به کاهش مصرف انرژی و کاهش تولید گازهای گلخانه‌ای گردد. موضوعی که پرداختن به آن امری داغ در مباحث مهندسی امروزی گردیده است.

۴-۱ اهداف تحقیق

تلاش اصلی این پژوهش بر ارائه یک مدل بهینه سازی کارا برای تخصیص صحیح ماشین‌های مجازی در مراکز داده است. به گونه‌ای که بتوان از زیر ساخت‌های موجود بهترین استفاده را داشت و با استفاده‌ی درست از منابع، بتوان در مصرف انرژی صرفه‌جویی و میزان تولید گاز گلخانه‌ای را کاهش داد. بطور کلی، اهداف مشخص این مطالعه عبارت‌اند از:

- ترکیب الگوریتم‌های کلونی مورچگان و کلونی زنبور عسل
- آزمایش روش‌های مختلف ترکیب از جمله، سری، موازی و درون گذاری
- آزمایش این الگوریتم ترکیبی و مقایسه با هر یک از الگوریتم‌های کلونی مورچگان و کلونی زنبور عسل

۵-۱ فرضیه‌های پژوهش

- در زیر فرضیه‌هایی در مورد مطالعه‌ی پیش رو ارائه شده است:
- الگوریتم‌های فرا ابتکاری منجر به رسیدن به جواب بهینه (زیر بهینه) در زمان کمتری می‌گردند.
 - ترکیب الگوریتم‌های فرا ابتکاری می‌تواند منجر به دستیابی به جواب‌های بهتر در زمان کمتری گردد.

- نحوه‌ی ترکیب الگوریتم‌های فراابتکاری می‌تواند بر کارایی و زمان اجرای آن‌ها تأثیرگذار باشد.

۶-۱ سازمان‌دهی فصول پایان‌نامه

فصول باقی‌مانده‌ی این پایان‌نامه به‌صورت زیر سازمان‌دهی شده است:

در فصل دوم، ما به بررسی مبانی نظری این مطالعه و ادبیات پیشین در مورد این مطالعه خواهیم پرداخت. ادبیات مشابه که با استفاده از انواع روش‌های سنتی و فراابتکاری به حل این مسئله پرداخته‌اند را مورد بررسی قرار خواهیم داد. همچنین جنبه‌های نوآوری این مطالعه نیز مورد بررسی قرار خواهد گرفت. در فصل سوم به بررسی روش پیشنهادی خواهیم پرداخت و الگوریتم خود را در غالب الگوریتم و فلوچارت نمایش خواهیم داد.

فصل چهارم به ارائه‌ی روش‌های موردنظر در تحقیق می‌پردازد و نحوه‌ی آماده‌سازی و پردازش داده‌ها و سپس پیاده‌سازی آن در محیط MATLAB خواهیم پرداخت. در این بخش به‌طور کامل پیاده‌سازی الگوریتم مورد بررسی قرار خواهد گرفت.

در فصل پنجم به بررسی تمامی نتایج به دست آمد و نتایج مقایسات خواهیم پرداخت و به شیوه‌ی بصری، مقایسات خود را ارائه خواهیم کرد. در نهایت در فصل ششم کارهای آتی را مورد بررسی قرار خواهیم داد.

۲-۱ مقدمه

با توجه به اینکه این مطالعه در زمینه بهینه سازی در مراکز داده‌ی رایانش ابری است، در ابتدای این بخش به بیان تعاریف، روش‌ها و الگوریتم‌های مرتبط با موضوع می‌پردازیم. در بخش دوم این فصل به بررسی و مرور تحقیقات انجام گرفته در زمینه بهینه‌سازی قرارگیری ماشین‌های مجازی و مهاجرت ماشین مجازی در مراکز داده می‌پردازیم تا با شناسایی نقاط قوت و ضعف هریک از آنها، به دیدی مناسب برای ارائه‌ی الگوریتم و مقایسه با آنها بپردازیم.

۲-۲ مبانی نظری

۲-۲-۱ رایانش ابری

پردازش ابری را می‌توان همانند یک چتر تصور کرد که دربرگیرنده‌ی سرویس‌های محاسباتی پیچیده‌ای است که بطور رسمی توسط فراهم‌کنندگان تجاری سرویس‌ها^۹ همانند آمازون، گوگل^{۱۰} و مایکروسافت^{۱۱} ارائه می‌شود. یکی از دلایلی که این پروژه را ابر نام‌گذاری کرده‌اند، قابلیت دسترسی به آن، همه از همه‌جای دنیا است [9].

رایانش ابری، تکنولوژی، سرویس‌ها و برنامه‌های کاربری موجود در اینترنت را گرفته و همه‌ی آنها را به یک ابری از سرویس تبدیل می‌کند. استفاده از واژه‌ی ابر، اساساً به دو مفهوم اساسی اشاره دارد [9] :

^۹ Commercial Providers

^{۱۰} Google

^{۱۱} Microsoft

- انتزاع^{۱۲}: رایانش ابری جزئیات پیاده سازی سیستم را از کاربران و توسعه دهندگان پنهان می کند. برنامه های کاربردی بر روی سیستم های فیزیکی مشخص نشده اجرا می شوند، داده ها در مکان هایی نامشخص ذخیره می گردند و این منابع ممکن است در هر کجای دنیا از طرف کاربران مورد استفاده قرار گیرد.

- مجازی سازی^{۱۳}: رایانش ابری، سیستم ها بر اساس روی هم ریزی^{۱۴} و اشتراک منابع^{۱۵} مجازی سازی می کند.

بسیاری از افرادی که در زمینه های آکادمیک و تجاری فعالیت دارند، تلاش های زیادی برای تعریف دقیق پردازش ابری نموده اند. بویا^{۱۶} و همکارانش [10] آن را به صورت زیر تعریف کرده اند: «محاسبات ابری، یک سیستم محاسباتی توزیع شده و موازی، متشکل از مجموعه ای از کامپیوترهای مجازی و متصل به یکدیگر است که به صورت پویا تخصیص پیدا کرده و به عنوان یکی از منابع محاسباتی یکپارچه بر مبنای قرارداد سطح سرویس^{۱۷}، از طریق مذاکره بین مشتری و فراهم آورنده سرویس معرفی می گردد». به عنوان تعریفی دیگر، فوررست^{۱۸} و همکارانش در [11] ادعا کرده اند که «ابر مجموعه ای از سرویس های مبتنی بر سخت افزار است که ظرفیت های محاسبه ای، شبکه ای و ذخیره سازی را فراهم می کند که در آن مدیریت این سخت افزارها در یک سطح انتزاعی بسیار بالایی قرار دارند و ظرفیت ساختارها به شدت انعطاف پذیر است.»

^{۱۲} Abstraction

^{۱۳} Virtualization

^{۱۴} Pooling

^{۱۵} Resource sharing

^{۱۶} Buyya

^{۱۷} Service-level agreements (SLA)

^{۱۸} Forrest

اصولا سرویس‌های ارائه‌شده در ابر را به سه دسته تقسیم می‌کنند:

- زیرساخت به‌عنوان سرویس^{۱۹} (IaaS): زیرساخت به‌عنوان سرویس درواقع فراهم آورنده‌ی ماشین‌های مجازی، مکان‌های ذخیره‌سازی مجازی، زیرساخت‌های مجازی و سایر سخت‌افزارها و منابع مجازی دیگر است.
- سکو به‌عنوان سرویس^{۲۰} (PaaS): سکو به‌عنوان سرویس درواقع فراهم آورنده‌ی ماشین‌های مجازی، سیستم‌های عامل، برنامه‌های کاربردی، سرویس‌ها، چارچوب‌های توسعه، تراکنش‌ها و ساختارهای کنترلی هستند.
- نرم‌افزار به‌عنوان سرویس^{۲۱} (SaaS): نرم‌افزار به‌عنوان سرویس درواقع یک محیط اجرایی کامل با نرم‌افزارهای کاربردی، مدیریت و واسط کاربری است.

۲-۲-۲ مراکز داده

یک مرکز داده عبارت است از یک ساختمان یا چندین ساختمان که به‌منظور قرارگیری زیرساخت‌های محاسباتی و ذخیره‌سازی فراهم شده است. کاربرد اصلی آن، تحویل ابزارهای موردنیاز همانند برق، سیستم خنک‌کننده، تجهیزات حمایتی و امنیتی است. تحویل انرژی ورودی و همچنین دفع گرمای تولید شده در مرکز داده، نیازمند هزینه‌های زیادی است. این هزینه‌ها بسته به اندازه‌ی مرکز داده می‌تواند بسیار متفاوت باشد [12].

^{۱۹} Infrastructure as a Service

^{۲۰} Platform as a Service

^{۲۱} Software as a Service

اندازه‌ی مراکز داده بسیار متفاوت است. دوسوم از سرورهای موجود در ایالات متحده در مراکز داده‌ای کوچک‌تر از ۵۰۰ مترمربع هستند و کمتر از ۱ مگاوات^{۲۲} برق مصرف می‌کنند [13]. بیشتر سروهای بزرگ مربوط به چندین کمپانی هستند که در کنار یکدیگر قرار گرفته‌اند.

در این مطالعه، ما الگوریتم پیشنهادی خود را بر روی مراکز داده‌هایی با اندازه‌های مختلف اجرا خواهیم کرد. بدین معنا که مرکز داده‌ی خود را با اندازه‌های مختلف از نظر تعداد پردازنده‌های رایانشی و میزان حافظه‌ی اصلی در دسترس و همچنین تعداد میزبان‌های مختلف مقایسه خواهیم کرد.

۲-۲-۳ مجازی‌سازی

در رایانش، مجازی‌سازی به معنای ایجاد یک نسخه‌ی مجازی از سکوها^{۲۳} سخت‌افزاری کامپیوتر، سیستم‌های کامپیوتری، دستگاه‌های ذخیره‌سازی و منابع شبکه‌ای است [14]. مجازی‌سازی از دهه‌ی ۱۹۶۰ به‌عنوان روشی برای تقسیم منطقی^{۲۴} منابع سیستم‌های کامپیوتری مین‌فریم^{۲۵} بین برنامه‌های کاربردی مختلف معرفی گردید. پس‌از آن معنای این عبارت توسعه پیدا کرد.

مجازی‌سازی سخت‌افزاری یا مجازی‌سازی سکو به معنای ایجاد یک ماشین مجازی است که مانند یک کامپیوتر واقعی با یک سیستم‌عامل است. نرم‌افزارهای اجرا شده بر روی این ماشین‌های مجازی از منابع سخت‌افزاری جدا هستند. برای مثال، کامپیوتری که در حال اجرای سیستم‌عامل ویندوز است، ممکن است میزبان یک ماشین مجازی باشد با سیستم‌عامل لینوکس باشد [15].

انواع مختلف مجازی‌سازی عبارت است از [9]:

^{۲۲} Mega Watt

^{۲۳} Platforms

^{۲۴} Logically

^{۲۵} Mainframe

- مجازی سازی کامل^{۲۶} - تقریباً تمامی سخت افزار تقلید شده و به نرم افزار اجازه می دهد که با نمونه ای کامل اما تقلید شده از سخت افزار در ارتباط باشد.
- مجازی سازی ناکامل^{۲۷} - بخشی از محیط مورد نظر (نه همه ی سخت افزار) تقلید می گردد. در نتیجه برخی از برنامه های مهمان ممکن است نیازمند به تغییراتی برای اجرا شدن در چنین محیطی باشند.
- مجازی سازی کمکی^{۲۸} - در این نوع مجازی سازی، محیط سخت افزاری تقلید نمی شود؛ اما برنامه های مهمان هر یک در داخل دامنه های خود اجرا می شوند، به طوری که به نظر می رسد هر یک بر روی یک سیستم عامل مجزا در حال اجرا هستند.

در این مطالعه هدف ما قرار دادن ماشین های مجازی موجود در مراکز داده هاست. از آنجایی که استفاده مستقیم از مراکز داده ها برای یک کاربر و یا شرکت وجود ندارد و هزینه ی زیادی را برای آن در بر خواهد داشت، از تکنولوژی مجازی سازی استفاده می کنند. با استفاده از مجازی سازی کامل می توان مراکز داده ی قوی را به تعداد بسیار زیادی ماشین مجازی تقسیم کرد که هر یک دارای قدرتی خاص بسته به نیاز کاربر هستند.

در این پایان نامه، مراکز داده ی خود را در شرایطی مختلف که دارای ماشین های مجازی مختلفی هستند مورد بررسی قرار خواهیم داد. همچنین قدرت این ماشین های مجازی را به صورت های مختلف در نظر خواهیم گرفت.

^{۲۶} Full virtualization

^{۲۷} Partial virtualization

^{۲۸} Paravirtualization

۴-۲-۲ الگوریتم‌های بهینه‌سازی فراابتکاری

مسئله قرارگیری ماشین‌های مجازی، یک مسئله بهینه‌سازی محسوب می‌شود، چراکه هدف یافتن حالتی بهینه برای قرارگیری ماشین‌های مجازی در داخل مراکز داده است. مسائل بهینه‌سازی مسائلی هستند که در آن‌ها یک یا چند تابع هدف به منظور کمینه یا بیشینه‌سازی با احتساب برخی محدودیت‌ها و پارامترها، تعریف می‌شوند. بهینه‌سازی سری در علم کامپیوتر و تحقیقات عملیاتی، جستجوی راه‌حل‌های تقریبی است که بهترین حالات ممکن را برای مشخصه هدف منظور می‌کنند. در حالت ایدئال، میزان تقریب برای راه‌حل‌های بهینه تا یک خطای ثابت کوچک قابل‌پذیرش است. در واقع یافتن پاسخ تا حد ممکن نزدیک به بهینه می‌تواند کفایت کند. اما گاهی فرایند جستجوی فضای مسئله به جای همگرایی به سمت راه‌حلهایی که بهینه محلی هستند همگرا می‌شود. از طرفی، در فضاهای جستجوی با ابعاد بالا امکان وجود توابع بهینه محلی زیاد است؛ بنابراین احتمال قرار گرفتن در بهینه محلی نیز در این فضاها بیشتر می‌شود [16]. مسئله بهینه‌سازی در مسائل غیرخطی و ترکیبی دشوارتر می‌شود. از روش‌های ارائه شده در حل مسائل بهینه‌سازی می‌توان به روش‌های قطعی و روش‌های احتمالی اشاره کرد. روش‌های سنتی یا قطعی در حل این مسائل تنها روابط ساده و خطی بین راه‌حل‌های موجود در فضای مسئله را می‌توانند نگاشت کنند. به علاوه، بر طیف وسیعی از مسائل از جمله مسائل چند مدل^{۲۹}، مسائل چندهدفه^{۳۰} و مسائل با تعداد زیاد محدودیت مناسبت نمی‌باشند چراکه به سمت بهینه محلی گرایش دارند. درحالی‌که، روش‌های احتمالی حل مسئله را به گونه‌ای تقریبی بر روابط غیرخطی و مبهم و پیچیده انجام می‌دهد.

بدین منظور، اخیراً الگوریتم‌های فراابتکاری‌ها برای حل مسائل بهینه‌سازی بسیار مورد استفاده قرار گرفته‌اند. چراکه این الگوریتم‌ها می‌توانند مجموعه‌ای از راه‌حل‌های بالقوه (جمعیت) را برای

^{۲۹} Multi model

^{۳۰} Multi Objective

تشخیص راه حل بهینه از طریق همکاری و رقابت میان راه حل ها در فرایندی تکراری بیابند. از جمله ی آنها الگوریتم های ^{۳۱}GA، ^{۳۲}ACO، ^{۳۳}PSO، ^{۳۴}AIA، ^{۳۵}ABC و ^{۳۶}BBO می باشند.

مذاش الهام اکثر این الگوریتم ها طبیعت است خواه اصول فیزیک باشد یا زیست شناسی و یا رفتارشناسی. این الگوریتم ها جمعیتی از عوامل هوشمند به منظور بهره برداری از هوش محاسباتی جمعی می باشند. با استفاده از این هوشمندی جمعی و استفاده از توابع شایستگی به منظور هدایت جستجو، قابلیت همگرایی به پاسخ های نزدیک به بهینه را در مسائل ترکیبی سخت دارند. فراابتکاری ها در ساختار خود دارای مؤلفه های تصادفی به منظور اکتشاف بهتر فضای جستجو می باشند. درواقع، ویژگی مشترک الگوریتم های فراابتکاری برخورداری آنها از مکانیسم فرار از بهینه محلی است. امکان حل مسائل گسسته و پیوسته را دارند. از سوی دیگر، این الگوریتم ها با مسائلی نیز مواجه اند از جمله: شامل پارامترهای متعددی هستند که برای مسئله مورد نظر باید به درستی تنظیم شوند. دیگر اینکه، در حل مسائل بهینه سازی سخت هریک به نحوی دچار همگرایی زودرس است. همگرایی زودرس ناشی از عدم تنوع در جمعیت راه حل ها است. در الگوریتم های فراابتکاری مختلف، مکانیسم های مختلف ایجاد تنوع در جمعیت راه حل ها در یک لحظه از فرایند جستجو بدین منظور تعبیه شده است. مسئله دیگر که متوجه این الگوریتم ها است، مسئله هدایت همگرایی به سمت بهینه سراسری است. بدین منظور، داشتن دانش زمینه ای در رابطه با مسئله و همچنین استفاده از مکانیسم های جستجوگری الگوریتم های مختلف به صورت مکمل می تواند به همگرایی سراسری کمک کند.

^{۳۱} Genetic Algorithm

^{۳۲} Ant Colony

^{۳۳} Particle Swarm Optimization

^{۳۴} Artificial immune algorithm

^{۳۵} artificial bee colony algorithm

^{۳۶} Biogeography-Based Optimization

الگوریتم‌های فراابتکاری به دودسته کلی روش‌های مبتنی بر یک جواب و مبتنی بر جمعیت تقسیم می‌شوند. دسته اول در طول پیشرفت فرایند جستجو یک جواب را بهبود می‌دهند، درحالی‌که در دسته دوم جمعیتی از جواب‌ها در طی فرایند جستجو مورد کنکاش قرار می‌گیرد. نمونه‌هایی از الگوریتم‌های مبتنی بر یک جواب شامل الگوریتم تبرید شبیه‌سازی‌شده، جستجوی ممنوعه^{۳۷}، جستجوی محلی تکرارشونده^{۳۸}، جستجوی همسایگی متغیر^{۳۹}، جستجوی محلی هدایت‌شده^{۴۰}، روش‌های هموار سازی، روش‌های نوسانی و جستجوی انطباقی حریصانه است. در دسته دیگر، فراابتکاری‌های مبتنی بر جمعیت خود به دودسته الگوریتم‌های تکاملی و الگوریتم‌های مبتنی بر هوش ازدحامی دسته‌بندی می‌شوند. الگوریتم‌های تکاملی برگرفته از تئوری تکاملی داروین می‌باشند، به‌گونه‌ای که یک جمعیت از جواب‌ها توسط عملگرهای بازترکیبی و جهش مورد اصلاح قرار می‌گیرند. الگوریتم‌های ژنتیک، استراتژی تکاملی^{۴۱}، برنامه‌ریزی ژنتیک، جغرافیای زیستی، تکامل تفاضلی و برنامه‌ریزی تکاملی^{۴۲} در این دسته قرار می‌گیرند. ایده الگوریتم‌های مبتنی بر هوش ازدحامی، تولید هوش محاسباتی از طریق بهره‌برداری ساده از رفتار اجتماعی بین عوامل و نه رفتار فردی تک تک عوامل است. بهینه‌سازی کلونی مورچگان، بهینه‌سازی ازدحام ذرات، الگوریتم ایمنی عصبی، بهینه‌سازی کلونی زنبور عسل مصنوعی، گروه نوازندگان^{۴۳} و رقابت استعماری^{۴۴} از جمله این الگوریتم‌ها هستند که

^{۳۷} tabu search

^{۳۸} iterated local search

^{۳۹} variable neighborhood search

^{۴۰} guided local search

^{۴۱} Evolution Strategy

^{۴۲} Evolutionary Programming

^{۴۳} Harmony Search Algorithm

^{۴۴} Imperialist Competitive Algorithm - ICA

می‌توانند برای طیف وسیعی از مسائل استفاده شوند [17]. البته دسته دوم را نیز می‌توان به نحوی تکاملی دانست چراکه در طی پیشرفت فرایند جستجوی الگوریتم، جمعیتی از جواب‌ها را به سمت پاسخ بهینه تکامل می‌دهند که البته تاکنون دسته‌بندی‌های مختلفی در ادبیات موضوع ارائه شده است. انواع الگوریتم‌های فراابتکاری در جدول ۱-۲ آمده است.

جدول ۱-۲ انواع الگوریتم‌های فراابتکاری

انواع الگوریتم‌های فراابتکاری	نام الگوریتم‌های فراابتکاری
	جستجوی محلی هدایت‌شده ^{۴۵} (GLS)
	روش جستجوی انطباقی حریصانه ^{۴۶} (GRASP)
	جستجوی محلی تکرارشونده ^{۴۷} (ILS)
الگوریتم‌های مبتنی بر یک جواب	تبرید شبیه‌سازی‌شده ^{۴۸} (SA)
	جستجوی ممنوعه ^{۴۹} (TS)
	جستجوی متغیر همسایگی ^{۵۰} (VNS)
	روش‌های هموارسازی ^{۵۱} (SM)
	روش‌های نوسانی ^{۵۲} (NM)

-
- 45. Guided Local Search
 - 46. Greedy Adaptive Search Procedure
 - 47. Iterative Local Search
 - 48. Simulated Annealing
 - 49. Tabu Search
 - 50. Variable Neighborhood Search
 - 51. Smoothing Methods
 - 52. Noisy Methods

بهینه‌سازی کلونی مورچگان^{۵۳} (ACO)

کلونی زنبورها^{۵۴} (BC)

الگوریتم ژنتیک^{۵۵} (GA)

برنامه‌ریزی تکاملی^{۵۶} (EP)

الگوریتم‌های مبتنی بر جمعیت

استراتژی تکاملی^{۵۷} (ES)

برنامه‌ریزی ژنتیک^{۵۸} (GP)

جستجوی پراکندگی^{۵۹} (SS)

بهینه‌سازی گروهی ذرات^{۶۰} (PSO)

۱-۴-۲-۲ الگوریتم کلونی زنبور عسل

هوش جمعی شاخه‌ای از پژوهش بر اساس جمعیت است که مدل‌های جمعیتی از عوامل مورد تداخل یا ازدحام که می‌توانند خودسازمان‌دهی کنند. کلونی مورچه، ازدحام پرندگان و یا زنبورها یک نمونه ساده‌ای از سیستم جمعیتی است. دیگر نمونه‌ای از هوش جمعی کلونی زنبور عسل در اطراف کندو است. هوش کلونی زنبور عسل (ABC) یک الگوریتم است که یک الگوریتم بهینه‌سازی بر اساس

53. Ant Colony Optimization

54. Bee Colony

55. Genetic Algorithm

56. Evolutionary Programming

57. Evolutionary Strategies

58. Genetic Programming

59. Scatter Search

60. Particle Swarm Optimization

رفتار هوشمندانه جمعیت زنبور عسل است. در این مطالعه، الگوریتم ABC برای بهینه‌سازی قرارگیری ماشین‌های مجازی در مرکز داده مورد استفاده قرار خواهد گرفت.

الگوریتم زنبور شامل گروهی مبتنی بر الگوریتم جستجو است که اولین بار در سال ۲۰۰۵ توسط کارابوگا^{۶۱} و همکارانش توسعه یافت؛ این الگوریتم شبیه‌سازی رفتار جستجوی غذای گروه‌های زنبور عسل است. در نسخه ابتدایی این الگوریتم، الگوریتم نوعی از جستجوی محلی انجام می‌دهد که با جستجوی تصادفی ترکیب شده و می‌تواند برای بهینه‌سازی ترکیبی یا بهینه‌سازی تابعی به کار رود [18].

یک کلونی زنبور عسل می‌تواند در مسافت زیادی و نیز در جهت‌های گوناگون پخش شود تا از منابع غذایی بهره‌برداری کند. قطعات گل‌دار با مقادیر زیادی گرده که با تلاشی کم قابل جمع‌آوری است، به وسیله‌ی تعداد زیادی زنبور بازدید می‌شود؛ به‌طوری‌که قطعاتی از زمین که گرده یا نکتار کمتری دارد، تعداد کمتری زنبور را جلب می‌کند.

پروسه‌ی جستجوی غذای یک کلونی به‌وسیله‌ی زنبورهای پیشرو^{۶۲} آغاز می‌شود که برای جستجوی گلزارهای امیدبخش فرستاده می‌شوند. زنبورهای پیشرو به صورت تصادفی از گلزاری به گلزار دیگر حرکت می‌کنند. در طول فصل برداشت محصول، کلونی با آماده نگه‌داشتن تعدادی از جمعیت کلونی به‌عنوان زنبور پیشرو به جستجوی خود ادامه می‌دهند. هنگامی که جستجوی تمام گلزارها پایان یافت، هر زنبور پیشرو، بالای گلزاری که اندوخته‌ی کیفی مطمئنی از نکتار و گرده دارد، رقص خاصی را اجرا می‌کند.

^{۶۱} Karaboga

^{۶۲} Scout bee

این رقص که به نام «رقص چرخشی»^{۶۳} شناخته می‌شود، اطلاعات مربوط به جهت تکه گلزار نسبت به کندو، فاصله تا گلزار و کیفیت گلزار را به زنبورهای دیگر انتقال می‌دهد. این اطلاعات زنبورهای اضافی و پیرو را به سوی گلزار می‌فرستد. بیشتر زنبورهای پیرو به سوی گلزارهایی می‌روند که امیدبخش‌تر هستند و امید بیشتری برای یافتن نکتار و گرده در آن‌ها، وجود دارد.

وقتی همه‌ی زنبورها به سمت ناحیه‌ای مشابه بروند، دوباره به‌صورت تصادفی و به علت محدوده‌ی رقصشان در پیرامون گلزار پراکنده می‌شوند تا به‌موجب این کار سرانجام نه یک گلزار، بلکه بهترین گل‌های موجود درون آن تعیین موقعیت شوند.

• الگوریتم

الگوریتم زنبور عسل هر نقطه را در فضای پارامتری متشکل از پاسخ‌های ممکن را به‌عنوان منبع غذا تحت بررسی قرار می‌دهد. «زنبورهای پیشرو»، کارگزاران شبیه‌سازی‌شده، به‌صورت تصادفی فضای پاسخ‌ها را ساده می‌کنند و به‌وسیله‌ی تابع شایستگی کیفیت موقعیت‌های بازدید شده را گزارش می‌دهند. جواب‌های ساده شده رتبه‌بندی می‌شوند و دیگر زنبورها نیروهای تازه‌ای هستند که فضای پاسخ‌ها را در پیرامون خود برای یافتن بالاترین رتبه محل‌ها جستجو می‌کنند (که «گلزار» نامیده می‌شود). الگوریتم به‌صورت گزینشی دیگر گلزارها را برای یافتن نقطه‌ی بیشینه‌ی تابع شایستگی جستجو می‌کند.

در الگوریتم ABC، موقعیت یک منبع غذایی یک راه‌حل مسئله بهینه‌سازی را نشان می‌دهند و مقدار شهد از منبع غذا مربوط به شایستگی راه‌حل همراه می‌شود. تعداد زنبورهای کارگر یا زنبورهای تماشاچی برابر با تعداد راه‌حل‌ها در جامعه است. در اولین قدم، ABC جمعیت اولیه را به‌صورت

^{۶۳} waggle dances

تصادفی توزیع می‌کند $P(G=0)$ راه‌حل‌های SN (مواقع منبع غذایی) که در آن SN نشان‌دهنده اندازه جمعیت است.

هر راه حل (منبع غذایی) $xi (i = 1.2.SN)$ برد D - بعدی است. در اینجا، D تعداد پارامترهای بهینه‌سازی است. پس از مقداردهی اولیه، جمعیت موقعیت‌ها (راه‌حل‌ها) در معرض تکرار چرخه است، $C_{max}, 2, \dots, C$ ؛ که C فرایندهای جستجوی زنبورهای کارگر و جستجوگر و پیشرو است.

یک زنبور کارگر مصنوعی بطور احتمالی تولید یک تغییر در موقعیت (راه‌حل) در حافظه خود برای پیدا کردن یک منبع غذایی جدید و تست میزان شهد (مقدار شایستگی) از منبع جدید (راه‌حل جدید) می‌کند. در مورد زنبور عسل واقعی، تولید منابع غذایی جدید مبتنی بر مقایسه فرآیند منابع غذایی در منطقه وابسته به اطلاعات جمع‌آوری، بصری، توسط زنبور عسل است. در این مدل، تولید موقعیت منبع جدید غذا نیز بر اساس یک فرآیند مقایسه موقعیت منبع غذایی است. باین‌حال، در این مدل، زنبورهای مصنوعی هرگونه اطلاعات در مقایسه استفاده نمی‌کنند. آن‌ها به‌طور تصادفی یک موقعیت منبع غذایی را انتخاب می‌کنند و تغییراتی را بر روی یکی از منابع موجود در حافظه خود تولید می‌کنند، به شرطی که مقدار شهد منبع جدید بیشتر از منبع قبلی حفظ شده در حافظه زنبور عسل باشد موقعیت جدید را حفظ کرده و موقعیت قبلی را فراموش می‌کند؛ در غیر این صورت او موضع قبلی را نگه می‌دارد. پس از اینکه فرایند جستجوی تمام زنبورهای کارگر تکمیل گردید، آن‌ها اطلاعات شهد از منابع غذایی (راه‌حل) و اطلاعات مربوط به موقعیت خود را با زنبورهای تماشاچی در محدوده رقص به اشتراک می‌گذارند. یک زنبور تماشاچی اطلاعات شهد گرفته شده از همه زنبورهای کارگر را ارزیابی می‌کند و یک منبع غذایی با احتمال مربوط به مقدار شهد آن انتخاب می‌شود. همین‌طور در مورد زنبور کارگر، تولید تغییراتی در موقعیت (راه‌حل) موجود در حافظه خود و مقدار شهد از منبع انتخابی

(راه حل) را چک می کند. آن شهدی که بیشتر از قبلی باشد را ارائه می دهد، زنبور عسل موقعیت جدید را حفظ می کند و قبلی را فراموش می کند. زنبور تما شاپی یک منبع غذایی را با توجه به مقدار احتمال مرتبط با آن منبع غذایی انتخاب می کند، pi که با فرمول (۱-۲) زیر محاسبه می شود:

$$pi = \frac{fit_i}{\sum_1^{SN} fit_n} \quad (1-2)$$

که در آن fit_i میزان شایستگی از راه حل i توسط زنبور کارگر آن ارزیابی شده است که ارزیابی متناسب با مقدار شهد منبع غذایی در موقعیت i ت و SN تعدادی از منابع غذایی که برابر با تعداد زنبورهای کارگر (BN) است. در این روش، زنبورهای کارگر اطلاعات خود را با زنبورهای تم اچی تبادل می کنند. ب منظور تولید یک موقعیت غذایی انتخاب شده از قبلی ، AC فرمول (۲-۲) را استفاده می کند :

$$V_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2-2)$$

که در آن $k \in \{1.2. \dots BN\}$ و $k \in \{1.2. \dots D\}$ شاخصشان به صورت تصادفی انتخاب شده است. هر چند K به صورت تصادفی تعیین شده است اما متفاوت از i است. همچنین ϕ_{ij} یک عدد تصادفی بین $[-1,1]$ است که تولید موقعیت منبع غذایی همسایه در اطراف x_{ij} را کنترل می کند ، و ییرات مقایسه ای موقعیت های غذایی همسایه توسط زنبور عسل به صورت بصری ارائه می شود. بر اساس جستجوی تصادفی ارائه شده، روش به نوعی جستجو به راه حل بهینه در فضای جستجو نزدیک می شود و گام مرحله طور تناوقی کاهش می یابد. ار پارامترهای تولید شده توسط این عملیات بیشتر از حد از پیش تعیین شده خودش باشد، پامتر را می توان به عنوان مقدار قابل قبول انتخاب کرد. منبع غذایی که

شهد آن توسط زنبورها رهاش شده با یک منبع ماده غذایی جدید توسط زنبورهای پیشرو جایگزین می شود. در الگوریتم ABC این امر با تولید موقعیت به صورت تصادفی شبیه سازی شده و جایگزین آن منبع رها شده می شود. در الگوریتم ABC، اگر یک موقعیت بیشتر از یک عدد از پیش تعیین شده بهبود نیابد آن منبع غذایی فرض شده ترک خواهد شد. پس از انتخاب هر منبع، موقعیت vi,j تولید شده و سپس توسط زنبور مصنوعی ارزیابی شد، عملکرد آن با xi,j مقایسه می شود، اگر مواد غذایی جدید برابر یا شهد بهتری از منبع قبلی داشت، آن را با قبلی در حافظه جایگزین می کند. در غیر این صورت، آن قبلی را نگه می دارد. به عبارت دیگر، یک مکانیسم انتخاب حریصانه را برای انتخاب بین منابع غذایی قبلی و فعلی را انجام می دهد. الگوریتم ABC در حقیقت چهار فرآیند مختلف انتخاب را به کار می گیرد:

۱. فرآیند انتخاب جهانی توسط زنبورهای پیشرو مصنوعی برای کشف مناطق امیدبخش انجام می شود.

۲. یک فرآیند انتخاب محلی در منطقه توسط زنبورهای کارگر مصنوعی انجام شده و پیشروها با توجه به اطلاعات محلی (در مورد زنبور عسل واقعی، این اطلاعات شامل رنگ، شکل و عطر گل) (زنبورها قادر به شناسایی نوع منبع شهد نمی شوند تا زمانی که به محل مناسب می رسند و بین منابع در حال رشد بر اساس عطر و بوی آنها تبعیض وجود دارد) برای تعیین یک همسایه منبع غذا در اطراف منبع موجود در حافظه مورد استفاده قرار می گیرند.

۳. روند انتخاب محلی به نام فرآیند انتخاب حریصانه توسط تمام زنبورها انجام می شود که در آن اگر مقدار شهد منبع کاندید بهتر از فعلی باشد، زنبور فعلی را فراموش می کند و منبع کاندید را حفظ می کند. در غیر این صورت، زنبور فعلی را در حافظه نگه می دارد.

۴. یک فرایند انتخاب تصادفی توسط زنبور پیشرو انجام می‌شود.

از توضیحات فوق روشن است که سه پارامتر کنترل وجود دارد که در ABC اصلی استفاده می‌شود:

- تعداد منابع غذایی که با تعداد زنبورهای کارگر یا زنبورهای پیشرو موجود است (SN).

- مقدار حد^{۶۴}

- حداکثر تعداد چرخه

۲-۲-۴-۲ الگوریتم بهینه‌سازی کلونی مورچگان

به‌منظور غلبه بر مشکل عدم بهینگی زیرمجموعه‌های ویژگی یافت شده توسط روش‌های سنتی، در دهه‌های اخیر الگوریتم‌های بهینه‌سازی فراابتکاری به‌کرات در تحقیقات متعدد مورد استفاده قرار گرفته‌اند. چراکه روش‌های فراابتکاری با توجه به قابلیت‌های پردازش موازی توسط جمعیتی از عوامل هوشمند، زیرمجموعه‌های کاندید را به‌عنوان راه‌حل‌های پیشنهادی هر عامل هوشمند ارائه داده و در طی یک فرایند جستجوی تکراری از طریق همکاری و رقابت میان عوامل به جستجوی هرچه بیشتر فضای مسئله در جهت یافتن پاسخ نزدیک به بهینه می‌پردازد. این روش‌ها تنها از تابع شایستگی برای هدایت جستجو استفاده می‌کنند اما به دلیل همین هوشمندی جمعی، قادر به کشف جواب‌های باکیفیت بالا و نزدیک به بهینه هستند.

یکی از شناخته‌شده‌ترین این الگوریتم‌ها در زمینه‌ی تحقیقاتی هوش ازدحامی در حل مسائل بهینه‌سازی، الگوریتم کلونی مورچگان است که همانند سایر فراابتکاری‌ها از دانش حاصل شده از

^{۶۴} The Value of limit

تکرارهای قبلی جهت همگرایی به سمت راه‌حل‌های بهتر استفاده می‌کند. بیش از این، به‌آسانی نیز پیاده‌سازی می‌شود؛ بنابراین در زمینه‌ی انتخاب ویژگی به گستردگی استفاده شده است.

ACO به دسته‌ی فراابتکاری‌های احتمالاتی مبتنی بر هوش ازدحامی تعلق دارد، الگوریتم‌هایی که می‌توانند راه‌حل‌های نزدیک به بهینه را در یک‌زمان محاسباتی معقول به وسیله هوشمندی ازدحامی از عوامل جستجوگر و همکاری میان آن‌ها و نه هوشمندی یک مغز مرکزی برای مسائل بهینه‌سازی ترکیبی سخت به دست آورند. ACO با قابلیت‌های جستجوی موازی و توزیع شده و برخورداری از مؤلفه تصادفی، قادر به اکتشاف فضای مسئله و یافتن زیرمجموعه ویژگی نزدیک به بهینه است. ماهیت پیاده‌سازی موازی و محاسبات توزیع شده‌ی آن، از همگرایی زودرس الگوریتم به سمت بهینه محلی ممانعت می‌کند [19].

بهینه‌سازی کلونی مورچگان در اوایل دهه ۱۹۹۰ توسط دوریگو برای حل مسائل بهینه‌سازی ترکیبی سخت مانند مسئله فروشنده دوره‌گرد توسعه یافت [20]. ایده ابتدایی ACO از رفتار جستجوگری مورچگان واقعی مبنی بر یافتن بهترین مسیر در بین مسیرهای مختلف از لانه به منبع غذا با وجود تمام موانع و محدودیت‌ها، الهام گرفته شده است.

اطلاعات بینایی مورچه‌ها در این مسیریابی، ماده‌ی شیمیایی به نام فرومون^{۶۵} است که مورچه‌ها مسیر عبوری خود را این‌گونه برای تکرارهای بعدی خود و برای سایر مورچه‌ها علامت‌گذاری می‌کنند. بنابراین، مورچه‌ها با استفاده از این ردگزاری می‌توانند کوتاه‌ترین مسیر میان منبع غذا و لانه را بیابند. مورچه‌های مصنوعی در ACO به‌عنوان عوامل تقلید مورچه‌های طبیعی می‌باشند. البته مورچه‌های مصنوعی شامل مزیت‌های برخورداری از حافظه کارآمد به‌منظور ذخیره‌سازی مسیرهای فرومون ریزی و امکان داشتن اطلاعات زمینه‌ای از فضای مسئله، نسبت به مورچه واقعی می‌باشند.

65 pheromone

فرایند جستجوگری مورچه‌ها این‌گونه بیان می‌شود که هر مورچه به‌تنهایی یک عمل ساده عبور از مسیر مطلوب‌تر برحسب میزان فرومون را انجام داده و بر آن فرومون می‌ریزد، اما در ادامه، این عمل همکاری کلونی مورچه‌ها به‌واسطه ارتباط غیرمستقیم آن‌ها از طریق فرومون‌ریزی می‌باشد که فرایند جستجوی فضا را به سمت مسیر بهینه هدایت می‌کند. شدت فرومون موجود بر مسیر به‌مرور زمان به‌واسطه عمل تبخیر^{۶۶}، کاهش می‌یابد. مفهوم فرومون‌ریزی و تبخیر را در بهینه‌یابی این‌گونه می‌توان بیان کرد که بر روی مسیرهای کوتاه‌تر به‌علت زمان رفت‌وآمد کمتر برای هر مورچه، میزان فرومون بیشتری قرار می‌گیرد، و بالعکس برای مسیرهای طولانی‌تر در یک بازه‌ی زمانی یکسان میزان فرومون کمتری اختصاص می‌یابد. بعلاوه مسیرهایی که ترافیک مورچه بیشتری دارد میزان فرومون بیشتری دارد. از سوی دیگر نیز، عمل تبخیر فرومون، احتمال همگرایی به مسیرهای طولانی و اشتباه را کاهش داده و از افتادن در بهینه محلی اجتناب می‌کند. و در پی آن، اکتشاف هرچه بیشتر فضاهای عبور نشده را نیز محتمل‌تر می‌سازد. اگر تبخیر وجود نداشت مسیرهای ابتدایی به حدی جذاب می‌شدند که احتمال یافتن مسیرهای جدید و بهینه به شدت کاهش می‌یافت. بنابراین، تبخیر مسیرهای طولانی‌تر و شدت بیشتر فرومون مسیرهای کوتاه‌تر، منجر به جذابیت کوتاه‌ترین مسیر می‌شود [20]. این فرایند بهره‌گیری از ارتباط غیرمستقیم بین مورچه‌ها با استفاده از ماده فرومون در جهت یافتن مسیر بهینه، «بازخورد مثبت^{۶۷}» خوانده می‌شود. «تبخیر فرومون» و «انتخاب احتمالاتی مسیرها» به مورچه‌ها امکان یافتن مسیرهای کوتاه‌تر را می‌دهد و منجر به انعطاف‌پذیری در حل مسائل بهینه‌سازی ترکیبی می‌شوند.

در کل فرایند ACO دو مرحله‌ی ایجاد راه‌حل/مسیر مطابق یک قاعده توزیع احتمالی^{۶۸} و بروز رسانی فرومون^{۶۹} بر مسیرهای عبوری هر مورچه را به صورت مکرر به‌منظور هدایت جستجو به سمت

66 evaporation

67 Positive feedback

68 probability distribution rule

69 Pheromone update

پاسخ بهینه انجام می‌دهد. از زمان معرفی ACO اصلی تاکنون اصلاحات متنوعی از ACO ارائه شده است که به‌طور کلی در این دو گام از یکدیگر متمایز می‌شوند. در ذیل تعدادی از انواع شناخته شده از الگوریتم بهینه‌سازی مورچگان معرفی می‌شود:

۱. سیستم مورچه^{۷۰} (AS):

در این روش هر مورچه پس از ایجاد مسیر کامل، یال‌های مسیر عبوری خود را برحسب هزینه پاسخی که ایجاد کرده‌اند، مطابق روابط (۳-۲) و (۴-۲) فرومون ریزی می‌کنند.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (3-2)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{J(\Psi^k)} & . \quad l_{ij} \in \Psi^k \\ 0 & . \quad l_{ij} \notin \Psi^k \end{cases} \quad (4-2)$$

در (۳-۲) پارامتر تبخیر، ρ تعداد مورچه‌ها، m مقدار فرومونی است که برحسب تابع هزینه مسیر بر روی یال i و j توسط مورچه k در تکرار t باید ریخته شود و τ_{ij} نیز مقدار فرومون روی یال i و j است. طبق این رابطه، فرومون مسیرهای عبوری مورچه‌ها برحسب مقدار هزینه هر مسیر و میزان تبخیر به‌روزرسانی می‌شود.

در (۴-۲)، J نشان‌دهنده‌ی هزینه مسیر و Ψ^k مسیر پیموده شده توسط مورچه k را نشان می‌دهد. l_{ij} یال موجود در مسیر k را نشان می‌دهد. $l_{ij} \notin \Psi^k$ بیان‌کننده این است که اگر یال در مسیر پیموده شده نباشد به‌روزرسانی فرومون برای آن انجام نمی‌شود. مطابق این رابطه نیز، برای مسیرهای طولانی فرومون کمتر و مسیرهای کوتاه فرومون بیشتر ریخته می‌شود.

۲. سیستم مورچه ممتاز^{۷۱}: (EAS)

در این روش تمام مورچه‌ها فرومون را برحسب هزینه مسیری که پیموده‌اند بر مسیر عبوری

70 ANT System (AS)

71 Elitist ANT System (EAS)

خود می‌ریزند. به علاوه، بهترین پاسخ سراسری (بهترین پاسخ تا تکرار جاری) مطابق رابطه (۵-۲) میزان فرومون اضافه‌تری نیز بر مسیر خود می‌ریزد.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k(t) + e\Delta\tau_{ij}^g(t) \quad (۵-۲)$$

$$\Delta\tau_{ij}^g(t) = \begin{cases} \frac{Q}{eJ(\Psi^k)} & . \quad l_{ij} \in \Psi \in \{\Psi^+, \Psi^*\} \\ 0 & . \quad otherwise \end{cases} \quad (۶-۲)$$

$$Q = \rho\tau_0 \quad (۷-۲)$$

e ضریب افزایش فرومون است. مطابق این رابطه بهترین پاسخ سراسری با ضریب e فرومون بیشتری دریافت می‌کند. در رابطه (۶-۲) شرط $l_{ij} \in \Psi \in \{\Psi^+, \Psi^*\}$ بیان می‌کند که تنها بهترین مورچه در تکرار جاری یا بهترین مورچه در تمام تکرارها اجازه فرومون‌ریزی دارند. رابطه (۷-۲) نیز به امکان برآورد مقدار ثابت Δ توسط ضریب تبخیر و فرومون اولیه، اشاره دارد.

۳. سیستم مورچه رتبه‌بندی شده^{۷۲}: (ASR)

در این روش، تمام راه‌حل‌های به دست آمده بر اساس طول جواب رتبه‌بندی می‌شوند و بر اساس همین رتبه‌بندی مقدار فرومون آزادسازی شده توسط آن‌ها مشخص خواهد شد و راه‌حل با طول کمتر از راه‌حل دیگر با طول بیشتر مقدار فرومون بیشتری آزاد می‌کند. رابطه (۸-۲) قاعده بروز رسانی فرومون در این روش را نشان می‌دهد.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{r=1}^{w-1} (w - r)\Delta\tau_{ij}^r(t) + w\Delta\tau_{ij}^g(t) \quad (۸-۲)$$

در رابطه (۸-۲) به تعداد w مورچه اجازه فرومون ریزی داده می‌شود. Γ رتبه ویژگی و $w-\Gamma$ وزن ریختن فرومون است که مطابق آن هرچه رتبه بهتر باشد وزن ریختن فرومون بیشتر می‌شود.

۴. سیستم مورچه حداکثر-حداقل^{۷۳}: (MMAS)

در MMAS مطابق روابط (۹-۲) و (۱۰-۲) یک مقدار کمینه و بیشینه برای فرومون تعیین کرده و فقط در هر مرحله مورچه با بهترین جواب متناسب با تابع برازش قادر به فرومونریزی است و تمام گره‌های مجاور آن به مقدار فرومون بیشینه، مقداردهی اولیه می‌شوند.

$$\tau_{ij} = [(1 - \rho)\tau_{ij} + \Delta\tau_{ij}^{best}(t)]_{\tau_{min}}^{\tau_{max}} \quad (9-2)$$

$$\Delta\tau_{ij}^{best}(t) = \begin{cases} \frac{c\tau_{ij}}{J(\Psi^k)} & l_{ij} \in \Psi^k = \Psi^+ \\ 0 & otherwise \end{cases} \quad (10-2)$$

$\Delta\tau_{ij}^{best}(t)$ بهترین پاسخ در تکرار جاری را نشان می‌دهد. در صورتی که بهترین پاسخ سراسری در این رابطه در نظر گرفته شود همگرایی سریع اتفاق می‌افتد که البته امکان قرار گرفتن در بهینه محلی را افزایش می‌دهد. اما با فرومونریزی بر بهترین پاسخ در هر تکرار، جستجوی بهتر اما زمان‌بر را منجر می‌شود. چون در این روش فرومونریزی بر روی برخی یال‌ها به‌کندی و تبخیر سریع‌تر رخ می‌دهد، میزان فرومونریزی روی یال‌ها به بازه max-min محدود شده است $\tau_{ij} \in [max.min]$ برای جلوگیری از همگرایی زودرس در تکرارهای اولیه الگوریتم و افزایش جستجوگری، مقدار اولیه فرومون به‌صورت تقریبی از τ_{max} در نظر گرفته می‌شود. از سوی دیگر، اجازه بیشتری به انتخاب یال‌های کمتر دیده شده می‌دهد.

۵. سیستم کلونی مورچگان^{۷۴}: (ACS)

قواعد بروز رسانی و تبخیر فرومون در این روش به‌صورت روابط (۱۱-۲) تا (۱۴-۲) است:

73 max-min ANT System (MMAS)

74 ANT Colony System (ACS)

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^g(t) \quad (11-2)$$

$$\Delta\tau_{ij}^g(t) = \begin{cases} \frac{\rho}{(1 - \rho)J(\Psi^+)} & , \quad l_{ij} \in \Psi^k = \Psi^+ \\ 0 & , \quad otherwise \end{cases} \quad (12-2)$$

$$\min\{\tau_{ij}, \Delta\tau_{ij}^g\} \leq (1 - \rho)\tau_{ij} + \rho.\Delta\tau_{ij}^g \leq \max\{\tau_{ij}, \Delta\tau_{ij}^g\} \quad (13-2)$$

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0 \quad (14-2)$$

این روش سه تفاوت اساسی با AS دارد:

۱. در AS ایجاد راه حل تنها با استفاده از قاعده احتمالی که تابعی از میزان فرومون و مطلوبیت اکتشافی می باشد عمل می کند. در حالی که، در روش ACS دو قاعده ایجاد راه حل حریصانه و احتمالی با تنظیم پارامتر q_0 به منظور بهره گیری از پارامترهای اکتشاف و استخراج مورد استفاده قرار می گیرد.

۲. مطابق روابط (۱۱-۲) و (۱۲-۲) با توجه به استفاده از ضریب ρ در بروز رسانی فرومون برای بهترین پاسخ، که به صورت یک ترکیب محدب از مقدار فرومون موجود و مقدار فرومونی که در این تکرار می خواهد بریزد، می باشد، به روزرسانی سخت تری دارد. بنابراین، مشکل این روش همگرایی کند آن می باشد چرا که سخت تر فرومون می ریزد یعنی کنترل می کند که با انتخاب هر یال به عنوان بهترین، میزان فرومون کمتری ریخته شود.

۳. قاعده به روزرسانی محلی فرومون در ACS به صورت رابطه (۱۳-۲) می باشد که طی آن ضریبی از مقدار فرومون اولیه نیز به یال ها افزوده می شود. بنابراین، قدرت جستجوگری الگوریتم در رفتن به مسیرهای ملاقات نشده افزایش می یابد. برخلاف MMAS که τ_{max} به عنوان فرومون اولیه منظور می شود، در این روش مقدار فرومون اولیه به مقدار کوچکی تنظیم می شود. چون ترکیب محدب است، مقدار max-min به درستی لحاظ می شود.

۳-۲ مرور ادبیات پیشین

از آنجایی که قرارگیری صحیح ماشین‌های مجازی در داخل مراکز داده‌ها از اهمیت زیادی برخوردار است، مطالعات زیادی بر روی آن انجام گرفته است. بطور کلی می‌توان این مطالعات را به دودسته‌ای استفاده از روش‌های فراابتکاری و استفاده از روش‌های غیر فراابتکاری تقسیم کرد. در ادامه به بررسی طبقه‌بندی‌شده‌ی این مطالعات می‌پردازیم.

۳-۱-۲ روش‌های غیر فراابتکاری

در روش‌های غیر فراابتکاری نویسندگان سعی کرده‌اند که بدون استفاده از الگوریتم‌های فراابتکاری، به پاسخ‌گویی به این مسئله‌ی بهینه‌سازی بپردازند.

چایسیری^{۷۵} و همکارانش یک الگوریتم قرارگیری بهینه‌ی ماشین‌های مجازی^{۷۶} (OVMP) را در [21] ارائه کردند. الگوریتم آن‌ها باهدف کاهش هزینه‌ها برای میزبانی ماشین‌های مجازی در یک محیط ابری بود. الگوریتم OVMO بر پایه‌ی تصمیم‌گیری بر روی راه‌حل بهینه‌ی برنامه‌نویسی صحیح تصادفی^{۷۷} (SIP) برای کرایه‌ی منابع از فراهم آورندگان ابری بود. کارایی الگوریتم آن‌ها توسط آزمایشات عددی و شبیه‌سازی مورد آزمون قرار گرفت. نتایج به‌وضوح نشان داد که الگوریتم پیشنهادی آن‌ها می‌تواند موجب کاهش هزینه‌های کاربران گردد.

در [22]، پیاو^{۷۸} و همکارانش برای حل این مسئله، یک روش قرارگیری و مهاجرت ماشین‌های مجازی در محیط ابری ارائه کردند. روش پیشنهادی آن‌ها، ماشین‌های مجازی را با در نظر گرفتن

^{۷۵} Chaisiri

^{۷۶} optimal virtual machine placement

^{۷۷} stochastic integer programming

^{۷۸} Piao

شرایط شبکه بین ماشین‌های فیزیکی و ذخیره‌سازی داده‌ها در مرکز داده قرار می‌داد. علاوه بر آن، روش پیشنهادی آن‌ها سناریویی را در نظر می‌گرفت که در آن شرایط ناپایدار شبکه، رفتارهای دسترسی به داده‌ها و کارایی برنامه را تغییر می‌داد. نتایج شبیه‌سازی آن‌ها در محیط کلودسیم اثبات کرد که روش پیشنهادی می‌تواند زمان اجرای وظایف را کاهش دهد.

کین‌لی^{۷۹} و همکارانش، سیاست‌هایی برای قرارگیری ماشین‌های مجازی در مراکز داده با تمرکز بر روی کاهش هزینه‌های مرتبط با الکتریسیته و در نتیجه کاهش نیاز به خنک‌کنندگی ارائه کردند. ارزیابی‌های آن‌ها نشان داد که روش پیشنهادی آن‌ها در مقایسه با روش‌های دیگر میزان الکتریسیته‌ی کمتری مصرف می‌کنند. در نتیجه‌ی مطالعات خود، آن‌ها دریافتند که با سیاست‌گذاری صحیح می‌توان در میزان برق مصرفی تا حد زیادی صرفه‌جویی کرد [23].

رحیمی‌زاده و همکارانش در [24] به مطالعه‌ی قرارگیری ماشین‌های مجازی با آگاهی از ترافیک شبکه پرداختند، با این فرض که فراهم آوردن‌گان سرویس از میزان ترافیک‌های وارد شده آگاه هستند. به این صورت که با پیش‌بینی میزان ترافیک در ساعات و روزهای مختلف، ماشین‌های مجازی در مراکز داده‌ها را سازمان‌دهی می‌کردند. آن‌ها در مطالعات خود دریافتند که کارایی هر ماشین مجازی در گرو توانایی زیرساخت برای برطرف سازی نیازهای ترافیکی است. از آنجایی که حل این مسئله یک مسئله‌ی NP-Hard است و به روش‌های دوجمله‌ای نمی‌توان به آن پاسخگویی کرد، آن‌ها چندین الگوریتم مبتنی بر هیروستیک^{۸۰} را پیشنهاد دادند.

^{۷۹} Kien Le

^{۸۰} heuristic

یوچی^{۸۱} و همکاران در [25] در ابتدا به بررسی ریسک‌های امنیتی محیط ابر بر پایه‌ی نفوذپذیری ماشین‌های مجازی و شماهای قرار دهی پرداختند. بر پایه‌ی ارزیابی‌های امنیتی، آن‌ها یک الگوریتم جدید ارائه کردند که توانایی حداقل سازی خطرات امنیتی را در بین ماشین‌های مجازی کاهش می‌داد. بر اساس نتایج آزمایش‌های آن‌ها، روش پیشنهادی آن‌ها قابلیت بهبود امنیت کل محیط ابری را دارا بود. هزینه‌های محاسباتی و هزینه‌های توسعه‌ی تکنیک آن‌ها نیز مورد بررسی، ارزیابی و امکان‌سنجی قرار گرفت.

فو^{۸۲} و همکاران [26] در مقاله‌ای با عنوان «انتخاب ماشین مجازی و قرارگیری در محیط ابر پویا» به حل مسئله‌ی بهینه‌سازی قرارگیری ماشین مجازی در محیط ابر پرداختند. آن‌ها یک سیاست قرارگیری جدید ماشین مجازی ارائه کردند که حداقل وابستگی را به ماشین‌های مجازی دیگر دارا بود. بالا بودن میزان وابستگی ماشین‌های مجازی می‌تواند منجر به افزایش پیچیدگی در عمل جابجایی ماشین‌های مجازی گردد. آن‌ها روش پیشنهادی خود را با استفاده از ابزار کلودسیم^{۸۳} شبیه‌سازی کردند و نتیجه گرفتند که مجموعه سیاست‌های در نظر گرفته شده در زمینه‌ی مصرف انرژی می‌تواند منجر به بهبود زمان انتقال ماشین‌های مجازی، سطح کیفیت سرویس و توافقات سطح سرویس گردد.

فوکوناگا^{۸۴} و همکارانش به کاهش میزان هزینه‌ی ارتباطات در بین ماشین‌های مجازی در یک شبکه‌ی مرکز داده پرداختند. آن‌ها یک مسئله‌ی بهینه‌سازی را برای یافتن بهترین قرارگیری ماشین ارائه کردند. برای حداقل سازی بار شبکه، برای هر درخواست، آن‌ها هزینه‌ی ارتباط قرارگیری ماشین‌های مجازی را در نظر گرفتند. هدف مسئله‌ی تعریف شده توسط آن‌ها، حداقل سازی هزینه‌های ارتباط بود.

۸۱

^{۸۲} Fu

^{۸۳} Cloudsim

^{۸۴} Fukunaga

بدین منظور، آن‌ها یک الگوریتم تخمینی برای این کار ارائه دادند. نتایج شبیه‌سازی آن‌ها، گواهی بر کارایی مناسب الگوریتم پیشنهادی آن‌ها بود [27].

۲-۳-۲ روش‌های فراابتکاری

از آنجایی که روش پیشنهادی ما نیز یک روش فراابتکاری است، بیشتر تمرکز خود را بر روی بررسی روش‌های فراابتکاری ارائه شده در زمینه‌ی قرارگیری ماشین مجازی در مراکز داده‌ها قرار داده‌ایم. جائو^{۸۵} و همکارانش در مقاله‌ای با عنوان «یک الگوریتم چند متغیره‌ی مورچگان برای قرارگیری ماشین‌های مجازی در محاسبات ابری» الگوریتم مورچگان را برای حل این مسئله پیشنهاد دادند. آن‌ها تابع هدف این الگوریتم را دو تابع در نظر گرفتند: از دست رفت منبع و مصرف انرژی. درواقع هدف آن‌ها کاهش هم‌زمان میزان مصرف انرژی و از دست رفت منابع بود. الگوریتم آن‌ها بر اساس برخی نمونه‌های آمده در نوشته‌های قبلی مورد آزمایش قرار گرفت. آن‌ها الگوریتم خود را پیاده‌سازی و با الگوریتم‌های مشابه از جمله الگوریتم ژنتیک مقایسه کردند. نتایج آن‌ها نشان داد که الگوریتم آن‌ها از بهبودی مناسبی برخوردار است. با آزمایشات گسترده‌ای که انجام دادند، آن‌ها دریافتند که روش آن‌ها از الگوریتم‌های دیگر پاسخ بهتری را داشته است [8].

ژنگ^{۸۶} و همکارانش در [28] در مقاله‌ای با عنوان «قرارگیری ماشین‌های مجازی بر پایه‌ی بهینه‌سازی زیست‌جغرافیایی چند هدفه»، از الگوریتم BBO برای حل این استفاده کردند. آن‌ها تعریف مسئله‌ی خود را همانند آنچه در [8] آمده است انجام داده و دو تابع هدف خود را به صورت از دست رفت منبع و مصرف انرژی در نظر گرفتند. آن‌ها الگوریتم پیشنهادی خود را برای پارامترهای مختلف

^{۸۵} Gao

^{۸۶} Zheng

انجام دادند. نتایج آن‌ها نشان داد که الگوریتم آن‌ها در بیشتر مواقع نتایج بهتری نسبت به الگوریتم ژنتیک چند هدفه دارد.

در [29]، استفانلو^{۸۷} در مقاله‌ای با عنوان «یک الگوریتم ژنتیک با کلید تصادفی برای قرارگیری ماشین‌های مجازی در مراکز داده‌ی دور از هم» تمرکز خود را بر روی قرارگیری ماشین‌های مجازی در دیتاست‌هایی که در مناطق جغرافیایی دور از هم قرار دارند، گذاشتند. آن‌ها مسئله را فرموله سازی کردند و یک الگوریتم جدید بر اساس الگوریتم ژنتیک به منظور یافتن جواب بهینه یا نزدیک به بهینه‌ی مسئله استفاده کردند. نتایج عملی نشان داد که الگوریتم ارائه شده بسیار کارا است و راه‌حل‌های ممکن را به سرعت یافته و نتایج بهتری نسبت به پاسخگوهای تجاری کنونی و الگوریتم‌های ارائه شده، به همراه دارد.

در [30]، جاشی^{۸۸} و همکاران یک روش جستجو برای بهترین قرارگیری ماشین‌های مجازی در مرکز داده با استفاده از الگوریتم فاخته^{۸۹} ارائه کردند. آن‌ها به این مسئله به عنوان یک مسئله بهینه‌سازی نگاه کردند و تابع هدف خود را برای الگوریتم فراابتکاری فاخته، از دست رفت منابع و مصرف انرژی انتخاب کردند. سپس آن‌ها الگوریتم پیشنهادی خود را در محیط متلب شبیه‌سازی کرده و با الگوریتم‌های مورچگان و ژنتیک مقایسه کردند. نتایج آن‌ها همواره منجر به بهبودی نمی‌شد، اما در شرایطی کارایی بسیار مناسبی نسبت به این دو الگوریتم از خود به نمایش گذارد.

^{۸۷} Estefanllo

^{۸۸} Joshi

^{۸۹} Cuckoo

کوتینو^{۹۰} و همکارانش در [31] در مقاله‌ای بانام "بهبود تخصیص ماشین برای کارهای علمی در مجموعه‌ی ابرهای عمومی" روشی بانام GraspCC-fed برای تولید تخمین مقدار بهینه یا نزدیک به بهینه برای تعداد ماشین‌های مجازی که باید به اجرای هر یک از کارها اختصاص داده شود، ارائه کردند. روش پیشنهادی آن‌ها، درواقع توسعه‌ای بر روی روش GRASP بود که پیش‌ازاین ارائه شده بود و بر روی ابرهای تکی و مجموعه‌ای از ابرها آزمایش شده بود. بنابراین آن‌ها به پیاده‌سازی روش پیشنهادی خود و مقایسه‌ی الگوریتم خود با GRASP پرداختند. با توجه به بهبود قابل‌توجهی که الگوریتم پیشنهادی آن‌ها دارا بود، آن‌ها عنوان داشتند که جایگزین بسیار خوبی برای الگوریتم‌های استفاده‌شده‌ی کنونی خواهد بود.

کانسال^{۹۱} و همکارانش یک تکنیک قرارگیری ماشین مجازی در رایانش ابری ارائه کردند که بر اساس الگوریتم بهینه‌سازی کرم شب‌تاب^{۹۲} بود. آن‌ها روش پیشنهادی خود را در کلودسیم پیاده‌سازی کردند و بهبودی حدود ۴۴ درصدی در میزان مصرف انرژی، کاهش ۷۲ درصدی میزان مهاجرت‌ها و کاهش ۳۴ درصدی میزان استفاده از میزبان‌ها را نتیجه گرفتند [32].

سینگ^{۹۳} و همکارانش در [33]، روشی برای قرار دادن ماشین‌های مجازی در مراکز داده با استفاده از الگوریتم کلونی زنبور عسل^{۹۴} ارائه کردند. هدف الگوریتم آن‌ها، حداقل‌سازی مصرف منابع و انرژی بود. الگوریتم آن‌ها به صورت چند هدفه ارائه شد. نتایج آزمایشات آن‌ها کارایی مناسبی از این الگوریتم فرابابتکاری برای حل این مسئله را به نمایش گذاشت.

^{۹۰} Coutinho

^{۹۱} Kansal

^{۹۲} Firefly

^{۹۳} Singh

^{۹۴} Honey Bee Colony

ونگ^{۹۵} و همکارانش در [34] الگوریتمی برای حداقل سازی میزان مصرف انرژی با استفاده از بهینه سازی ازدحام ذرات^{۹۶} برای بهینه سازی قرارگیری ماشین های مجازی در مراکز داده پرداختند. آنها برای تطابق الگوریتم PSO با مسئله ی خود، بهبودهایی را بر روی آن انجام دادند. آنها با استفاده از الگوریتم PSO بهبودیافته به یک بهبودی ۱۳ الی ۲۳ درصدی در مصرف انرژی دست یافتند.

۴-۲ خلاصه فصل

در این فصل، در ابتدا به بررسی مبانی نظری درزمینه ی قرارگیری ماشین های مجازی در مراکز داده، با استفاده از الگوریتم های فراابتکاری پرداختیم و بطور خاص به بررسی معانی و کاربردهای مراکز داده، تکنولوژی مجازی سازی و الگوریتم های فراابتکاری پرداختیم.

سپس در بخش دوم فصل به بررسی مقالات کار شده درزمینه ی قرارگیری ماشین های مجازی در مراکز داده پرداختیم و بر اساس دیدگاه این پایان نامه، آنها را از منظر حل مسئله، به دودسته ی مقالات متمرکز بر الگوریتم های فراابتکاری و غیر فراابتکاری تقسیم کردیم.

^{۹۵} Wang

^{۹۶} particle swarm optimization

۳-۱ مقدمه

قرار گیری ماشین مجازی در مراکز داده، به فرایند تخصیص ماشین های مجازی به تعدادی میزبان موجود در مرکز داده اطلاق می شود. از چنین رویه هایی برای بهینه سازی قرار گیری ماشین های مجازی در مراکز داده، کاهش میزان از دست رفت منابع و کاهش مصرف انرژی در مراکز داده استفاده می شود. بر این اساس، در این فصل با نمایش ریاضی و الگوریتم و فلوچارت روش پیشنهادی به بحث در مورد آن خواهیم پرداخت.

۳-۲ طبقه بندی مسائل قرار گیری ماشین

در این بخش روش های بهینه سازی مختلف برای این مسئله بطور دسته بندی شده مورد بررسی قرار خواهد گرفت. روش های بهینه سازی شناخته شده در این زمینه را به می توان به دسته های زیر تقسیم کرد: (۱) مسئله ی بهینه سازی تک هدفه^{۹۷}، (۲) چند هدفه ای که بصورت تک هدفه پاسخ داده شده اند^{۹۸} و (۳) روش های چند هدفه ای خالص^{۹۹}. در زیر در مورد هر یک از این روش ها به تفصیل صحبت خواهد شد.

۳-۲-۱ روش های تک هدفه

یک روش تک هدفه، بهینه سازی را تنها با در نظر گیری یک هدف انجام می دهد. بر اساس مطالعات انجام شده، عموماً مسئله ی قرار گیری ماشین در مراکز داده تنها از با یک هدف مورد بررسی قرار گرفته است به طوری که ۶۰ درصد از کل مقالات ارائه شده برای پاسخگویی به این مسئله از نوع

^{۹۷} Mono-Objective Optimization Problem (MOP)

^{۹۸} multi-objective solved as mono-objective (MAM)

^{۹۹} pure multi-objective (PMO)

تک هدفه هستند. بیش از ۷۰ درصد مقالات ارائه شده در زمینه‌ی قرار گیری ما شین‌های مجازی در مراکز داده تک هدفه دارای تابع هدف‌های مختلفی هستند [35]. بقیه موارد هم که دارای توابع یک‌سان بوده، مسئله‌ی خود را با شکلی متفاوت مدل کرده‌اند. بیشتر توابع هدف ارائه‌شده بر روی کاهش هزینه‌ها، و افزایش استفاده‌ی هرچه بیشتر از منابع تمرکز داشته‌اند. با توجه به اینکه تعداد توابع هدف ارائه‌شده دارای زمینه‌های مختلفی است، استفاده روش‌های چند هدفه معقول‌تر به نظر می‌آید.

۳-۲-۲ چند هدفه، پاسخ داده شده بصورت تک هدفه

در این روش‌ها، بهینه‌سازی چند هدفه تنها با استفاده از یک هدف انجام گرفته است. در واقع تابع هدفی در نظر گرفته خواهد شد که بصورت علنی یک هدف را داشته اما بصورت ضمنی چند هدف دیگر را در خود دارد. مشکل این روش نیاز به دانش بتلا در مورد دامنه‌ی مسئله برای ترکیب صحیح اهداف است. بر اساس مطالعات، ۳۴ درصد مطالعات انجام شده در این زمینه تعدادی بیش از یک هدف را دنبال کرده‌اند حال آنکه مسئله‌ی خود را تنها با استفاده از یک هدف نوشته‌اند [35]. در مطالعه یک هدف به عنوان عنوان ترکیب خطی از چند هدف نوشته می‌شود.

۳-۲-۳ روش‌های چند هدفه‌ی خالص

یک مسئله‌ی بهینه‌سازی چند هدفه‌ی خالص شامل مجموعه‌ای از p متغیر تصمیم‌گیری، q تابع هدف و r شرط می‌باشد. توابع هدف و شرایط، توابعی از متغیرهای تصمیم هستند. در یک فرموله سازی چند هدفه، x نشان‌دهنده‌ی بردار تصمیم بوده و y نشان‌دهنده‌ی بردار هدف است. فضای تصمیم‌گیری با X نشان داده می‌شود و فضای هدف بصورت Y نشان داده می‌شود. آنگه بر اساس [36] داریم:

Optimize:

$$y = f(x) = [f_1(x), f_w(x), \dots, f_q(x)]$$

subject to:

$$e(x) = [e_1(x). e_2(x). \dots e_r(x)] \geq 0$$

where:

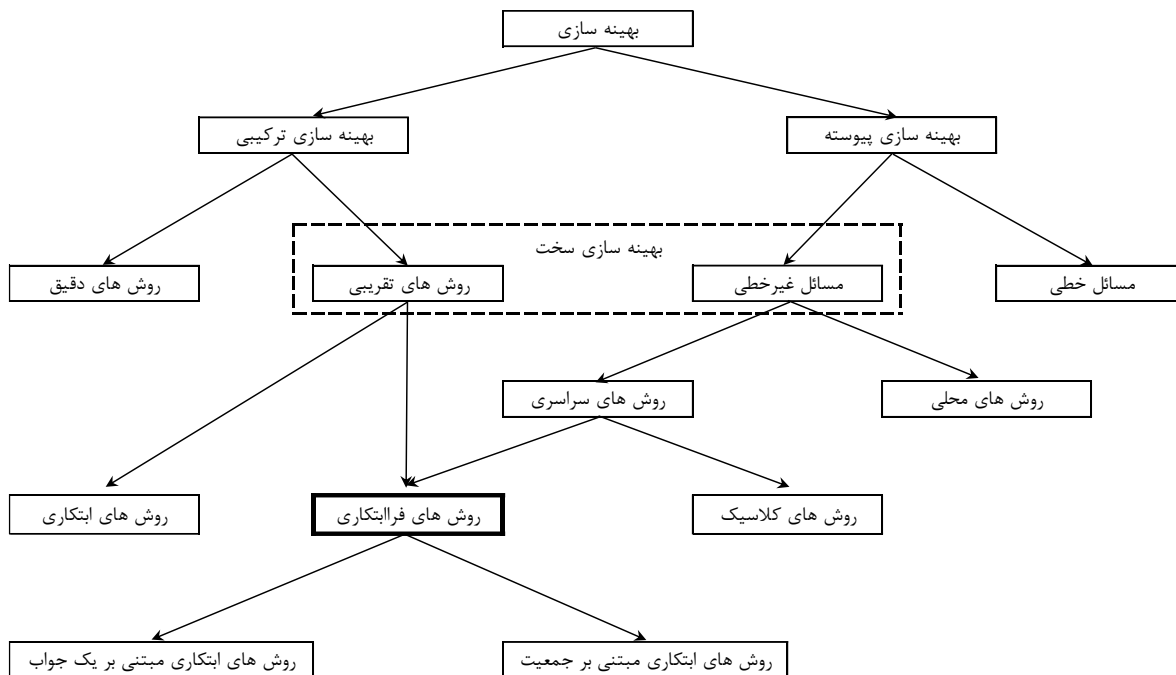
$$x = [x_1. x_2. \dots x_p] \in X$$

$$y = [y_1. y_2. \dots y_q] \in Y$$

در این مطالعه ما مسئله‌ی بهینه‌سازی خود را بصورت تک هدفه در نظر خواهیم گرفت.

۳-۳ انواع روش‌های حل مسائل بهینه‌سازی

مسائل بهینه‌سازی به دو دسته مسائل بهینه‌سازی ترکیبی و پیوسته تقسیم می‌شود. در زمینه مسائل پیوسته، روش‌های کلاسیک بسیاری برای بهینه‌سازی سراسری ارائه شده است. اما این تکنیک‌ها در صورتی که تابع هدف دارای ویژگی‌های مشخصی نظیر محدب بودن نباشد، معمولاً کارایی ندارد. روش‌های بهینه‌سازی پیوسته به دو دسته‌ی روش‌های بهینه‌سازی خطی و غیر خطی تقسیم‌بندی می‌شوند. روش‌های حل مسائل بهینه‌سازی ترکیبی را نیز می‌توان به دو قسمت الگوریتم‌های دقیق و تقریبی تقسیم نمود. بهینه‌سازی غیرخطی پیوسته و روش‌های تقریبی مسائل گسسته، جزء مسائل سخت محسوب می‌شود.



شکل ۱-۳ انواع روش های حل مسئله برای مسائل بهینه سازی

۳-۳-۱ روش های بهینه سازی ترکیبی

همان طوری که بیان شد روش های حل مسائل بهینه سازی ترکیبی را نیز می توان به دو قسمت الگوریتم های دقیق و تقریبی تقسیم نمود. الگوریتم های دقیق معمولاً برای حل مسائل با اندازه های کوچک و متوسط بکار می رود، و رسیدن به جواب بهینه را در هر یک از این مسائل تضمین می کند. ولی برای مسائل واقعی که معمولاً با ابعاد بزرگ و پیچیده هستند قابل استفاده نیستند. به عنوان مثال روش شاخه و کرانه^{۱۰۰} در این دسته از الگوریتم ها قرار دارد.

۱۰۰ . Branch & Bound (B&B)

بیشتر مسائل بهینه‌سازی ترکیبی، در کلاس NP-hard قرار می‌گیرند، که برای حل آن‌ها از روش‌های تقریبی که در یک زمان حل قابل قبول، جواب‌هایی نزدیک به بهینه را می‌دهند، استفاده می‌شوند. روش‌های دقیق به دو دسته ابتکاری و فراابتکاری تقسیم می‌شود.

الگوریتم‌های ابتکاری مختلفی برای حل مسائل بهینه‌سازی ترکیبی ابداع شده است که در آن‌ها، جواب نزدیک به بهینه تولید می‌شود. اکثر آن‌ها فقط در مورد یک مسأله‌ی به خصوص قابل کاربرد هستند. یکی از نقص‌های الگوریتم‌های ابتکاری، تولید یک جواب یا تعداد محدودی از جواب‌ها و توقف آن‌ها در بهینه‌ی محلی باکیفیت پایین است. الگوریتم‌های فراابتکاری، برای حل این مشکلات و نقص‌های روش‌های ابتکاری پیشنهاد شده‌اند. بنابر این در این مطالعه از روش فراابتکاری استفاده خواهیم کرد.

۴-۳ روش پیشنهادی

در این بخش به بررسی روش پیشنهادی خواهیم پرداخت. در ابتدا مسئله را بصورت مجزا با الگوریتم‌های مورچگان و زنبور عسل ارائه خواهیم کرد، سپس روش پیشنهادی را غالب ترکیب این دو الگوریتم ارائه می‌کنیم.

۴-۳-۱ حل مسئله با استفاده از الگوریتم کلونی زنبور عسل

الگوریتم کلونی زنبور عسل یکی از الگوریتم‌های فراابتکاری است که به تفصیل در فصل ۲ در مورد آن صحبت گردید. این الگوریتم نیاز به تولید جواب‌های اولیه دارد که به صورت تصادفی تولید می‌شود.

۱-۱-۳ نحوه نمایش جواب

در تمام الگوریتم‌های فراابتکاری، به دلیل نیاز به حل شدنی در شروع کار، لازم است پاسخ‌ها را طبق ساختار مشخصی ذخیره کنیم که به این ساختار، نحوه نمایش جواب می‌گویند. در اینجا برای نمایش جواب از ساختار آرایه‌ای استفاده می‌کنیم. بر این اساس، هر پاسخ یک آرایه‌ی ۱ در n است که در آن n تعداد ماشین‌های مجازی بوده و درایه‌ی های آن نشان‌دهنده‌ی شماره‌ی میزبانی است که ماشین مجازی به آن تعلق گرفته است. برای مثال، در شکل، ماشین مجازی شماره‌ی ۱ در میزبان شماره ۳ و ماشین مجازی شماره ۵ در میزبان شماره ۷ قرار گرفته است؛ با فرض اینکه تعداد ماشین‌های مجازی دقیقاً ۵ مورد است.

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

شکل ۳-۲ شمای نمایش جواب

۲-۱-۴ تولید جواب اولیه

در الگوریتم پیشنهادی، استراتژی تولید جواب اولیه به صورت تصادفی است. اما با توجه به نوع مسأله و محدودیت‌های موجود، بدست آوردن جواب اولیه شدنی به صورت تصادفی در یک مرحله غیر ممکن است. از اینرو ما از یک الگوریتم ابتکاری تکاملی برای تولید جواب اولیه استفاده کرده‌ایم. در هر تکرار از این الگوریتم، به ازای هر بار تولید یک پاسخ جدید، ممکن بودن پاسخ مورد بررسی قرار می‌گیرد. بر این اساس، الگوریتم ابتکاری مذکور میزان بار موجود بر روی هر یک از سرورها را از نظر مصرف پردازنده، حافظه و پهنای باند بررسی کرده و در صورتی که این مقدار از توان سرور فراتر نباشد، جواب مذکور یک جواب موجه است. در غیر این صورت باید به دنبال تولید جواب دیگری بود.

۳-۴-۱-۳ رقص چرخشی

در این مطالعه رقص چرخشی را همانند یک جهش در الگوریتم ژنتیک در نظر گرفته‌ایم با تنظیم یک شعاع r که در واقع کسری از تعداد ماشین‌های مجازی مورد مطالعه است، تعدادی ماشین مجازی از محل خود خارج شده و بر روی میزبان‌های دیگر مهاجرت خواهند کرد. این کار در واقع جستجوی همسایگی جواب‌های تولیدشده است. رقص چرخشی در واقع استراتژی تقویت^{۱۱} می‌باشد.

۳-۴-۱-۴ زنبورهای پیشرو

زنبورهای پیشرو در هر تکرار از این مطالعه به تولید جواب‌های جدید خواهند پرداخت. نحوه تولید این جواب‌های اولیه به صورت تصادفی و با توزیع نرمال خواهد بود. جستجوی زنبورهای پیشرو در واقع استراتژی تنوع بخشی^{۱۲} می‌باشد.

۳-۴-۱-۵ تابع هدف

هدف از قرار گیری صحیح ماشین‌های مجازی در مرکز داده، کاهش میزان از دست رفت منابع در مراکز داده است. این هدف بطور ضمنی اهداف دیگری همچون کاهش مصرف انرژی را نیز در خود دارا می‌باشد. از دست رفت منابع برای میزبان j ام بر اساس به صورت زیر است [28]:

$$resourceWastage(j) = \sum_{j=1}^m \left(y_j \times \frac{\left| (T_{p_j} - \sum_{i=1}^n (x_{ij} \cdot R_{p_i})) - (T_{m_j} - \sum_{i=1}^n (x_{ij} \cdot R_{m_i})) \right| + \varepsilon}{\sum_{i=1}^n (x_{ij} \cdot R_{p_i}) + \sum_{i=1}^n (x_{ij} \cdot R_{m_i})} \right)$$

که در آن y_j نشاندهنده آن است که سرور j ام در حال استفاده بوده یا خیر، R_{p_i} میزان پردازنده مصرفی توسط ماشین مجازی i ام، R_{m_i} میزان حافظه مصرفی توسط ماشین مجازی i ام،

^{۱۱} Exploitation

^{۱۲} Exploration

T_{mj} حد آستانه‌ی ماکزیمم حافظه‌ی قابل استفاده، T_{pj} حد آستانه‌ی ماکزیمم پردازنده‌ی قابل استفاده و در نهایت x_{ij} صفر است اگر و تنها اگر ماشین مجازی i ام بر روی سرور j ام قرار داشته باشد.

۳-۴-۱-۶ معیار توقف

برای توقف یک الگوریتم معیارهای مختلفی ارائه شده است. در این مطالعه به منظور ساده سازی، از تعدادی تکرار مشخص برای توقف الگوریتم استفاده شده است.

جدول ۳-۱ عناصر الگوریتم زنبورعسل

عناصر الگوریتم	استراتژی های بکار گرفته شده
جواب	یک آرایه با مقدار مناسب
جواب اولیه	جواب شدنی تصادفی با الگوریتم تکاملی تصادفی
فضای حل	جواب شدنی با توجه به محدودیت‌های مسأله
معیار ارزیابی	تابع هدف مدل
انتخاب بهترین همسایگی	بر اساس بهترین تابع هدف، لیست ممنوعه، معیار آرمانی
استراتژی تقویت	تغییر در جواب های کنونی
استراتژی تنوع بخشی	شروع مجدد با جواب تصادفی شدنی
معیار توقف	ماکزیمم تعداد تکرار

فلوچارت الگوریتم زنبورعسل در شکل ۳-۳ آمده است.



شکل ۳-۳ فلوجارت الگوریتم زنبور عسل

۳-۴-۲ حل مسئله با استفاده از الگوریتم کلونی مورچگان

الگوریتم کلونی مورچگان الگوریتم بهینه سازی دیگری است که در این مطالعه مورد استفاده قرار گرفته است. این الگوریتم برعکس الگوریتم کلونی زبور عسل، به تولید جواب های اولیه نیازی ندارد و پاسخ های اولیه را می سازد. بدیهی است که نحوه ی نمایش جواب های اولیه نیز دقیقاً مشابه با الگوریتم کلونی زبور عسل است.

۳-۴-۲-۱ تولید پاسخ ها

در الگوریتم مورچگان، پاسخ ها بصورت گام به گام تولید می شوند. بدین معنا که درایه های نشان داده شده در شکل (۳-۲) در یک حلقه با n تکرار (که n تعداد ماشین های مجازی است) تولید می شوند و به ازای اضافه شدن هر عدد به ارایه و مشخص شدن مکان قرار گیری ماشین مجازی در مرکز داده، ممکن بودن پاسخ به همانند آنچه که در بخش ۳-۴-۱-۲ توضیح داده شد، انجام می گیرد. همچنین تابع هدف این مسئله نیز همانند آنچه که در ۳-۴-۱-۵ آمده است، می باشد.

۳-۴-۲-۲ دانش زمینه ای (هیروستیک)

هیروستیک مسئله به معنای دانش اولیه ی مسئله و درک کلی از جواب های اولیه است. در صورتی که مقداری هیروستیک اولیه هو شمند انتخاب گردد، جواب ها هر چه زودتر به سمت بهینه ی سراسری هدایت می شوند. در این پایان نامه، مقدار دانش زمینه ای را بصورت ثابت و برابر ۱ قرار داده ایم تا قابلیت جستجوی محیط با شانس ثابت وجود داشته باشد.

۳-۴-۲-۳ ارائه گراف

به طور کلی، برای استفاده از الگوریتم ACO، فضای جستجوی مسئله ی انتخاب ویژگی باید توسط یک گراف وزن دار کاملاً متصل بدون جهت مشابه مسئله TSP برای کاربرد ACO ارائه شود که در آن

گره‌ها یا شهرها نشان‌دهنده‌ی ویژگی‌ها در مجموعه‌ی ویژگی اصلی هستند و یال‌های میان گره‌ها انتخاب ویژگی بعدی را نشان می‌دهد. سپس جستجو برای زیرمجموعه‌ی ویژگی بهینه به مسئله‌ی پیدا کردن مسیر با حداقل تعداد گره‌ها با توجه به یک معیار تابع تناسب مشخص، تبدیل می‌شود. این گراف به صورت $G = \langle S, E \rangle$ تعریف می‌شود طوری که $S = \{S_1, S_2, \dots, S_n\}$ مجموعه‌ی تمامی میزبان‌های اصلی و نشان‌دهنده‌ی گره‌ها در گراف می‌باشد و $E = \{(S_i, S_j) : S_i, S_j \in S\}$ نشان‌دهنده‌ی یال‌های گراف می‌باشد. وزن هر یال $(S_i, S_j) \in S$ نیز با مقدار معیار هیوریستیک (دانش زمینه‌ای) مشخص می‌شود. اکنون، سه حالت برای ارائه فضای مسئله انتخاب ویژگی برای الگوریتم ACO به وجود می‌آید: حالت گسسته، حالت باینری و حالت گسسته/باینری. در حالت گسسته، هر ویژگی به دیگری مسیر دارد و ماتریس مقادیر فرومون و مطلوبیت اکتشافی به صورت ماتریس n در n می‌باشد. در این روش، ابتدا مورچه‌ها به صورت تصادفی روی گراف پخش می‌شوند سپس هر مورچه برای ساختن پاسخ خود از "قاعده انتقال حالت" مطابق رابطه (۳-۱)، استفاده می‌کند. مسئله‌ی ما نیز به صورت گسسته خواهد بود.

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & j, l \in J_i^k \\ 0 & \text{otherwise} \end{cases} \quad (3-1)$$

P_{ij}^k احتمال قرار گرفتن مورچه k در گره j در زمان $t+1$ می‌باشد، اگر در زمان T در گره i باشد. η_{ij} هیوریستیک (بینایی) مسئله می‌باشد و τ_{ij} نیز میزان فرومون ریخته شده توسط مورچه‌ها روی یال متصل‌کننده i و j می‌باشد. پس از ایجاد پاسخ‌های هریک از مورچه‌ها، به‌روزرسانی فرومون نیز وابسته به نوع الگوریتم ACO مورد استفاده توسط هریک از روابط مربوطه انجام می‌شود.

۳-۴-۲-۴ ایجاد راه حل (قاعده انتقال حالت^{۱۰۳})؛

جستجو برای وضعیت بهینه‌ی قرارگیری ماشین‌های مجازی، هدف مورچه‌ها از سفر بر روی گراف است. ابتدا مورچه‌ها به صورت تصادفی روی گراف پخش می‌شوند. سپس هر مورچه در یک روند تکراری با اضافه کردن هر سرور به زیرمجموعه‌ی انتخاب شده، راه حل خود را ایجاد می‌کند. این کار طبق "قاعده‌ی انتقال حالت" با استفاده از قاعده‌ی حریصانه یا قاعده‌ی احتمالی که ترکیبی از اطلاعات اکتشافی و مقادیر فرومون می‌باشد، انجام می‌شود.

در قاعده‌ی حریصانه، k امین مورچه که بر روی ویژگی F_{ix} قرار دارد ویژگی بعدی F_{jy} را مطابق رابطه (۲-۳) انتخاب می‌کند:

$$jy = \begin{cases} \arg \max\{\tau_{ix,y}^\alpha [\eta_{ix,ly}]^\beta\} \cdot l \in J_l^k & \text{if } q \leq q_0 \text{ exploitation} \\ p_{ix,jy}^k & \text{otherwise exploration} \end{cases} \quad 2-3$$

که در آن J_l^k مجموعه‌ی سرورهای مجاز به انتخاب برای هر مورچه K هستند، $\tau_{ix,jy}$ شدت فرومون بر یال (ix,jy) میان دو زیرگروه S_{ix} و S_{jy} (میزان ردپای فرومونی مورچه‌ها روی یال (ix,jy)) در زمان t است که نشان‌دهنده‌ی مقدار اطلاعات مفید به دست آمده از تجربیات قبلی مورچه‌ها می‌باشد و انعکاس دهنده تمایل مورچه‌ها در انتخاب این یال در ادامه فرایند می‌باشد. $\eta_{ix,jy}$ اطلاعات هیوریستیک (بینایی مسئله) می‌باشد که مطلوبیت انتخاب یال (ix,jy) را نشان می‌دهد. این پارامتر همان‌طور که پیش‌تر نیز گفته شد، نشان‌دهنده‌ی دانش قبلی درباره‌ی مسئله است که می‌تواند مورچه‌ها را به سمت راه‌حل‌های امیدوارکننده‌تر هدایت کند. در واقع، باعث بینایی مورچه‌های مصنوعی می‌شود. هر مورچه ترجیح می‌دهد یالی را انتخاب کند که شدت فرومون بیشتر و دانش زمینه‌ای مطلوب‌تری داشته باشد.

α, β عملکرد وزنی دارند و میزان اهمیت ردپا (فرمون/تجربه مورچه‌ها) در مقابل بینایی (اطلاعات زمینه‌ای مسئله) را کنترل می‌کنند. β اهمیت نسبی مقادیر اکتشافی را نشان می‌دهد. اگر برابر صفر باشد همانند مورچه معمولی عمل می‌کند که متکی بر یادگیری می‌باشد و به تجربه مورچه‌ها اکتفا می‌کند. بنابراین، منجر به همگرایی سریع به شرایط ایستا خواهد شد و مورچه‌ها راه‌حل خوبی را پیدا نخواهند کرد. α نیز اهمیت نسبی دانش جمعی را نشان می‌دهد. در صورتی که مقدار آن برابر صفر باشد، فرایند جستجو با تکیه بر مقادیر مطلوبیت اکتشافی کاملاً حریصانه خواهد بود که در آن یادگیری معنا ندارد. از این روی، نسبت این دو پارامتر اهمیت دارد. معمولاً در هوش ازدحامی مقدار α را بیشتر از β در نظر می‌گیرند.

q یک مقدار توزیع شده یکنواخت تصادفی بین $[0, 1]$ می‌باشد. و q_0 پارامتر ثابت از پیش تعریف شده (فاکتور احتمال استخراج/اکتشاف) می‌باشد ($0 \leq q_0 \leq 1$). اگر $q \leq q_0$ باشد، مورچه‌ها ویژگی‌هایی را که حاصل مقدار اکتشافی و سطح فرمون را حداکثر کند، انتخاب خواهند کرد. در این مورد اطلاعات یاد گرفته شده از تجارب گذشته و مطلوبیت اکتشافی مسیر به شکلی حریصانه مورد استفاده قرار خواهد گرفت در غیر این صورت اگر $q \geq q_0$ باشد، طبق قاعده‌ی احتمالی گرایش به اکتشاف هرچه بیشتر فضای مسئله ادامه خواهد یافت. تنظیم q_0 می‌تواند بین اکتشاف و استخراج تعادل برقرار کند. به عبارتی دیگر، q_0 بالا احتمال انتخاب تصادفی را کم خواهد کرد. هر دو تابع قاعده انتخاب به مورچگان کمک می‌کند که امکان بررسی حالت‌های جدید را که به راه‌حل بهینه نزدیک هستند را حفظ کنند.

۳-۴-۲-۵ قاعده آپدیت فرمون

پس از اینکه تمام مورچه‌ها راه‌حل‌های خود را ایجاد کردند، میزان تناسب هر پاسخ با استفاده از یک تابع تناسب برآورد می‌شود. در این پژوهش از معیار از دست رفت منابع برای ارزیابی هر

زیرمجموعه استفاده می شود. سپس، متناسب با ارزش محاسبه شده برای هر زیرمجموعه، میزان فرومون بر مسیرهای مربوطه به روزرسانی می شود (اضافه کردن ردپا به شاخه هایی که مورچه ها اخیراً از آن ها عبور کرده اند). مقادیر فرومون با بکار بردن دو قاعده آپدیت سراسری فرومون^{۱۰۴} و آپدیت محلی فرومون^{۱۰۵} به روزرسانی می شوند. قاعده آپدیت سراسری مطابق روابط (۳-۳) و (۳-۴) می باشد:

$$\tau_{ix,jy}(t+1) = (1-\rho)\tau_{ix,jy}(t) + \sum_{k=1}^m \Delta\tau_{ix,jy}^k(t) + \rho\Delta\tau_{ix,jy}^g(t) \quad 3-3$$

$$\Delta\tau_{ix,j}^k(t) = \begin{cases} fitness(k) & \text{if } ix,jy \in subset(k) \\ 0 & \text{otherwise} \end{cases} \quad 3-4$$

که در آن m تعداد مورچه ها در هر تکرار و ρ ضریب تبخیر مسیر فرومون می باشد
 $(0 < \rho < 1)$. $\tau_{ix,jy}(t)$ و $\tau_{ix,jy}(t+1)$ به ترتیب مقادیر فرومون را بر روی یال های (ix,jy) در زمان t و در زمان $t+1$ نشان می دهد. g نشان دهنده ی بهترین مورچه در هر تکرار است. همه ی مورچه ها فرومون را مطابق با رابطه ی (۳-۳) به روزرسانی می کنند و بهترین مورچه میزان فرومون بیشتری را بر ویژگی های مربوط به بهترین راه حل می ریزد. این مقدار اضافه تر فرومون، منجر به اکتشاف مورچه ها پیرامون راه حل بهینه در تکرارهای بعدی می شود.

قاعده آپدیت محلی نیز به صورت رابطه زیر تعریف می شود:

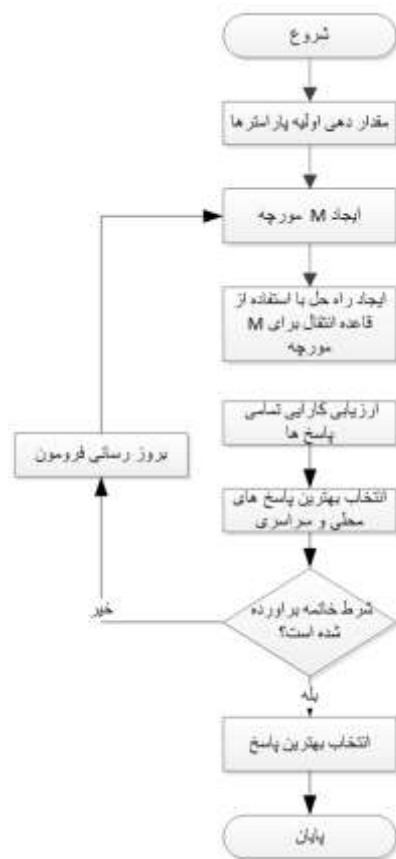
$$\tau_{zx,wy} = (1-\rho)\tau_{zx,wy} + \rho.\tau_0 \quad (3-5)$$

۳-۴-۲-۶ فلورچارت الگوریتم مورچگان

در این بخش فلورچارت مربوط به الگوریتم مورچگان ارائه شده است. فلورچارت الگوریتم مورچگتن در شکل (۴-۳) نشان داده شده است.

104 global pheromone update rule

105 local pheromone update rule



شکل ۳-۳-۴ فلوجارت الگوریتم کلونی مورچگان

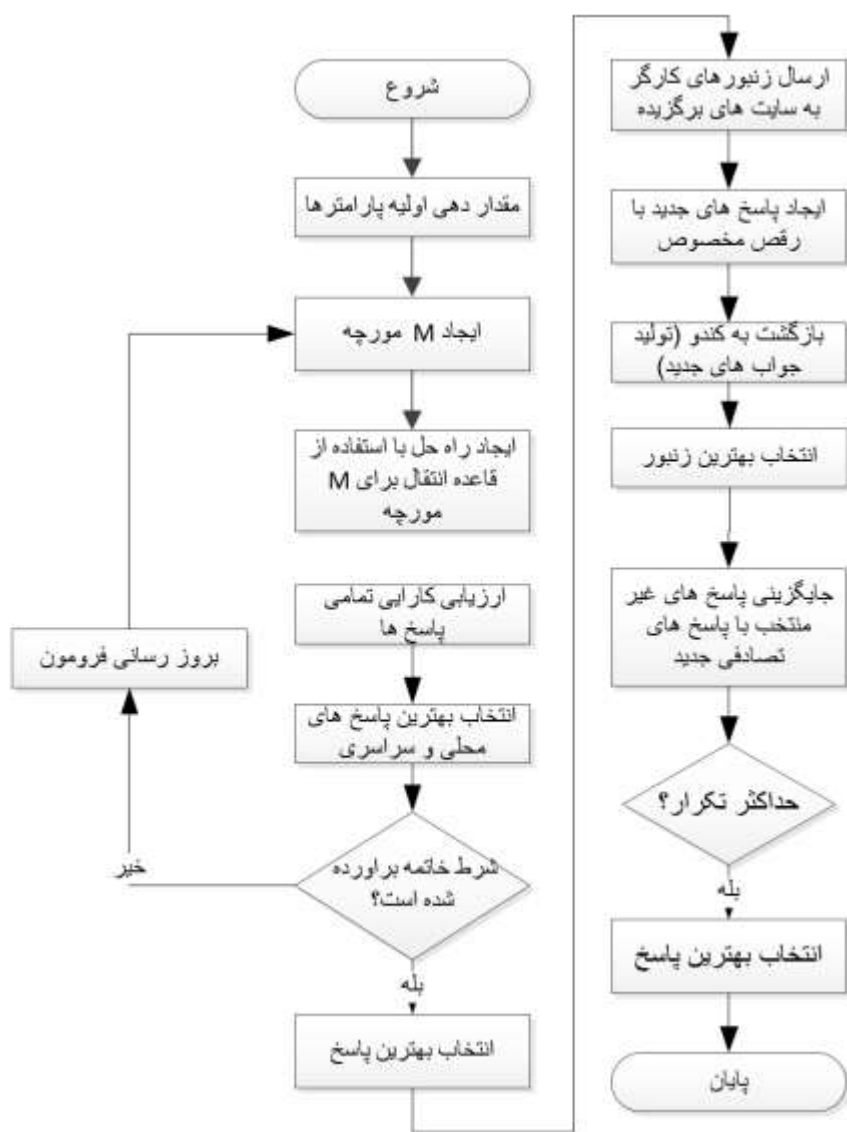
۳-۴-۳ الگوریتم پیشنهادی

الگوریتم پیشنهادی این مطالعه در واقع ترکیبی از دو الگوریتم مورچگتن و زنبور عسل می باشد. روش های مختلفی برای ترکیب الگوریتم های فراابتکاری ارائه شده است که می توان به سریال، موازی و درون گذاری اشاره کرد. در روش سریال، یکی از الگوریتم ها اجرا شده و نتایج حاصل از آن به الگوریتم دوم داده می شود تا الگوریتم دوم جواب های الگوریتم اول را بهبود دهد. چنین روشی معمولاً

در زمان‌هایی استفاده می‌شود که الگوریتم دوم پاسخ‌های اولیه تولید کرده و این پاسخ‌ها به صورت تصادفی باشد. بهتر آن است که بجای پاسخ‌های تصادفی، از پاسخ‌های تولید شده توسط الگوریتم اول استفاده گردد.

در روش موازی، هر یک از الگوریتم‌ها در یک تکرار پاسخ‌های خود را به اشتراک می‌گذارند. در واقع یک تکرار از هر یک از آن‌ها بطور همزمان اجرا شده، پاسخ‌های آن‌ها با یکدیگر ترکیب شده و به تکرار بعد منتقل می‌شود. در روش درون گذاری، یکی از الگوریتم‌ها به نحوی در داخلی دیگری قرار گرفته و بهینه‌سازی قسمتی از آن را برعهده دارد.

در این مطالعه از روش سریال استفاده شده است. بنابراین یکی از الگوریتم‌ها در امتداد دیگری قرار خواهد گرفت. باید توجه داشت که به دلیل آنکه الگوریتم کلونی زنبور عسل نیازمند پاسخ‌های تصادفی اولیه است، ترجیح ما استفاده از الگوریتم زنبور عسل در امتداد الگوریتم مورچگان است. در واقع در ابتدا الگوریتم مورچگان اجرا شده و پاسخ‌های تولید شده از آن به عنوان جواب‌های اولیه به الگوریتم زنبور عسل داده می‌شود. فلوچارت روش پیشنهادی شکل ۳-۵ نشان داده شده است.



شکل ۳-۵ فلوجارت روش پیشنهادی

۴-۱ مقدمه

در این بخش عملکرد الگوریتم ارائه شده شده در فصل قبل برای مدل تجزیه و تحلیل می‌شوند. در ادامه، ابتدا به شرایط پیاده سازی مسئله و سپس در قالب مثال عددی، نحوه عملکرد الگوریتم‌ها از نظر کارایی و اثربخشی بررسی و مقایسه شده است. در انتها عملکرد الگوریتم‌ها برای مسائل داده شده، از نظر آماری مقایسه و ارزیابی می‌شوند.

۴-۲ نرم افزار و سخت افزارهای شبیه سازی

برای انجام این مطالعه، از نرم افزار متلب نسخه ی ۲۰۱۵ استفاده شده است. همچنین این نرم افزار بر روی سیستمی با پردازنده ی Intel 6700Hq با ۴ هسته پردازنده و ۸ هسته ی مجازی با ۳۲ گیگابایت رم اجرا شده است. الگوریتم‌های ارائه شده ی در این مطالعه و الگوریتم‌های پیشین در این نرم افزار پیاده سازی شده‌اند.

۴-۳ معیارهای مقایسه

همانطور که در فصل سه نیز صحبت شد، هدف اصلی این مطالعه ارائه ی روشی برای مهاجرت ماشین‌های مجازی با تمرکز بر روی کاهش از دست رفت منابع در مراکز داده ی رایانش ابری است. بنابراین معیار اصلی آن دست رفت منابع خواهد بود. اما برای مقایسه معیارهای دیگری همچون زمان اجرای الگوریتم و تعداد سرورهای روشن نیز قرار داده شده است. بدیهی است که برای تمامی این سه معیار، مقادیر کمتر بهتر خواهد بود.

۴-۴ نتایج بدست آمده

به منظور مقایسه‌ی الگوریتم پیشنهادی با سایر الگوریتم‌ها در شرایط یک‌سان، هسته‌های تولید اعداد تصادفی را برای همگی ثابت در نظر خواهیم گرفت. در این فصل به مقایسه‌ی کارایی سه الگوریتم فراابتکاری زنبور عسل، مورچگان و ترکیبی خواهیم و همچنین الگوریتم FIFO خواهیم پرداخت و به منظور آنکه مقایسه‌ی صحیح و در شرایط یک‌سان داشته باشیم، هسته‌ی تولید اعداد تصادفی را در آنها برابر خواهیم گرفت. همچنین مقایسه‌ی آنها در دو سناریوی مختلف و با دو هسته‌ی تولید اعداد تصادفی مختلف انجام خواهد شد که در هر سناریو الگوریتم‌ها در شرایطی که دارای ۲۰ ماشین مجازی هستند، مورد بررسی قرار خواهند گرفت. همچنین در فصل بعد، مقایسه‌ی ۱۰ اجرای هر یک از این الگوریتم‌ها در مقایسه با الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی و الگوریتم ژنتیک نیز انجام خواهد شد. پارامترهای هر یک از الگوریتم‌های فراابتکاری در جداول ۴-۱ تا ۴-۴ آمده است.

جدول ۴-۱ پارامترهای مربوط به الگوریتم ژنتیک

حداکثر تکرار	تعداد جمعیت اولیه	درصد جمعیتی انتخابی برای تقاطع	درصد جمعیت انتخابی برای جهش	درصد ژن‌هایی که بر روی آنها جهش رخ می‌دهد
۵۰	۱۶	۸۰	۳۰	۱۰

جدول ۴-۲ پارامترهای مربوط به الگوریتم مورچگان

حداکثر تکرار	تعداد مورچه‌ها	نرخ بهره‌برداری / اکتشاف	مقدار آلفا	مقدار بتا	نرخ پرش فرومون
۵۰	۱۰۰	۷۰ درصد	۰,۵	۰,۵	۰,۱

جدول ۳-۴ پارامترهای مربوط به الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی

حداکثر تکرار	تعداد جمعیت اولیه	درصد ویژگی‌هایی که بر روی آن‌ها جهش رخ می‌دهد	درصد نگهداری جمعیت
۲۰۰	۵۰	۱۰	۷۰

جدول ۴-۴ پارامترهای مربوط به الگوریتم کلونی زنبور عسل

حداکثر تکرار	تعداد جمعیت مورچه پیشاهنگ	درصد مکان‌های منتخب	درصد مکان‌های نخبه
۵۰	۲۰	۵۰	۴۰

سناریو اول ۴-۴-۱

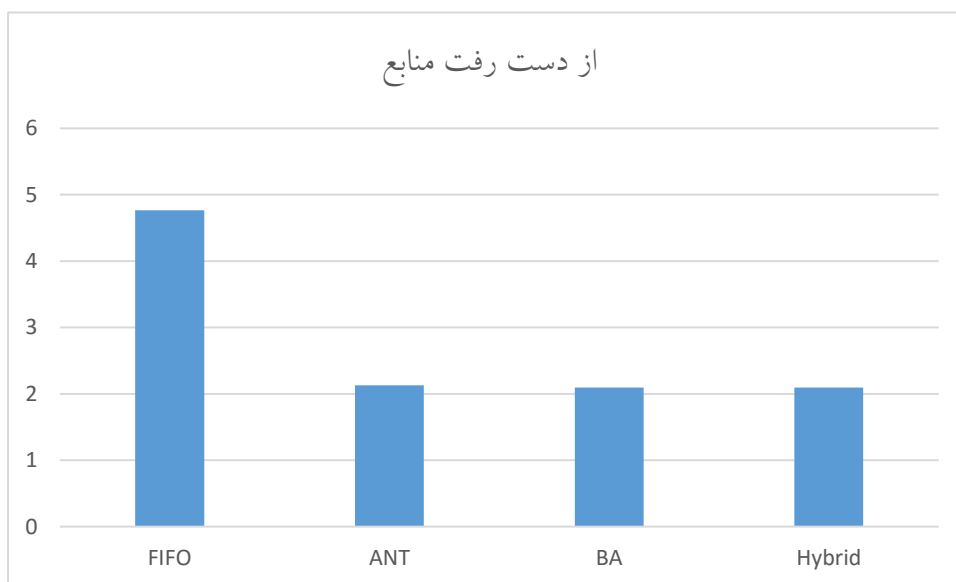
در سناریو اول، هسته‌ی تولید کننده‌ی اعداد تصادفی برابر با ۱۴ در نظر گرفته خواهد شد. این هسته بصورت تصادفی در نظر گرفته شده است. بنابراین برای اجرای هر ۴ الگوریتم، اعداد تصادفی یکسانی تولید خواهد شد که شرایط آزمایش را یکسان می‌کند. سپس هر چهار الگوریتم در شرایطی که صف ورودی ماشین‌های مجازی دارای ۲۰ ماشین مجازی هستند، آزمایش خواهد شد.

جدول ۵-۴ نتایج بدست آمده برای سناریو اول

سرورهای روشن	زمان	از دست رفت	
17	0.0781	4.7682	FIFO
6	4.9844	2.1274	ANT
6	2.6094	2.0930	BA
6	1.7969	2.0930	Hybrid

نتایج بدست آمده برای آزمایش اول در جدول ۴-۴ نشان داده شده است. با یک نگاه کلی می‌توان متوجه شد که الگوریتم ترکیبی و الگوریتم زنبور عسل با تولید پاسخ‌های حداقلی، توانسته‌اند بر رقیبان خود پیروز شوند. نمودارهای مقایسه‌ای جدول ۴-۴ در شکل‌های ۴-۱ تا ۴-۳ آمده است. هرچند روش ترکیبی پاسخی مشابه با الگوریتم زنبور عسل تولید کرده است، اما پاسخ تولید شده در یک زمان کوتاه‌تر نسبت به الگوریتم زنبور عسل تولید شده است.

همانطور که در شکل ۴-۱ می‌توان مشاهده کرد، الگوریتم ترکیبی از نظر از دست رفت انرژی دقیقاً مشابه با الگوریتم کلونی زنبور عسل بوده است. اما با مقایسه‌ی زمان تولید مدل می‌توان قدرت روش پیشنهادی را متوجه شد (شکل ۴-۲). همچنین روش ترکیبی پاسخ‌های بهتری را نسبت به الگوریتم مورچگان و همچنین الگوریتم FIFO تولید کرده است.



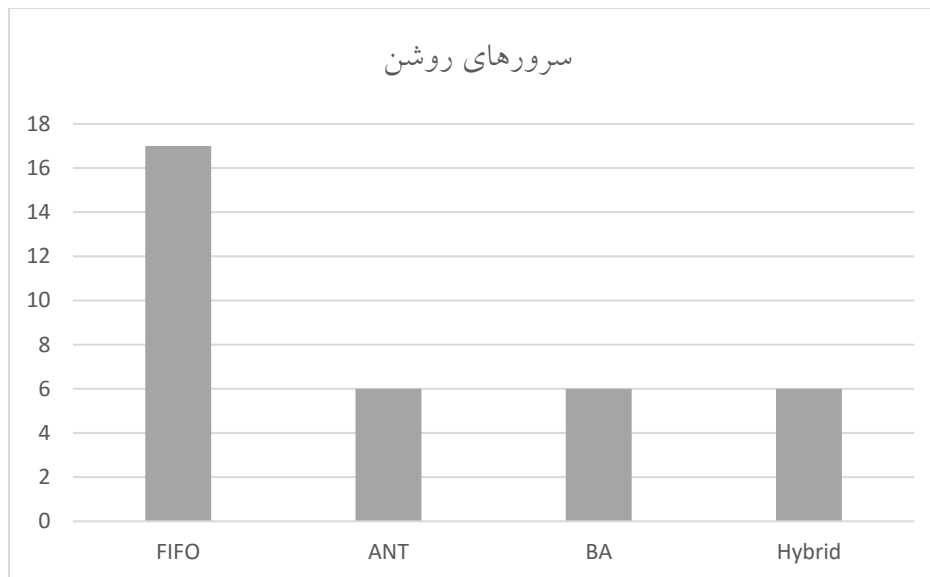
شکل ۴-۱ مقایسه‌ی از دست رفت منابع در سناریو اول



شکل ۲-۴ مقایسه‌ی زمان اجرای الگوریتم در سناریو اول

از نظر زمان تولید پاسخ، الگوریتم ترکیبی با فاصله‌ی ۲۰ درصدی بهتر از الگوریتم کلونی زنبور عسل بوده است. پس از الگوریتم زنبور عسل، الگوریتم مورچگان قرار دارد و همانطور که انتظار می‌رود FIFO با فاصله‌ای زیادی نسبت به سایر الگوریتم‌ها زمان بهتری داشته است (شکل ۲-۴)؛ اما هیچگاه نتوانسته پاسخ‌های نزدیک به الگوریتم‌های فراابتکاری موجود تولید کند.

بر اساس شکل ۳-۴، الگوریتم ترکیبی از نظر تعداد سرورهای روشن مشابه با دو الگوریتم فراابتکاری مورچگان و زنبور عسل عمل کرده است. بر این اساس، الگوریتم ترکیبی پیشنهادی توانسته است در این آزمایش بر سایر الگوریتم‌ها بصورت نسبی غلبه کند. هرچند پاسخ تولیدی آن مشابه با الگوریتم کلونی زنبور عسل بوده است، اما زمان تولید آن بسیار کمتر از آن بوده است.



شکل ۳-۴ تعداد سرورهای روشن در روش های مورد مقایسه در سناریو اول

۲-۴-۴ سناریو دوم

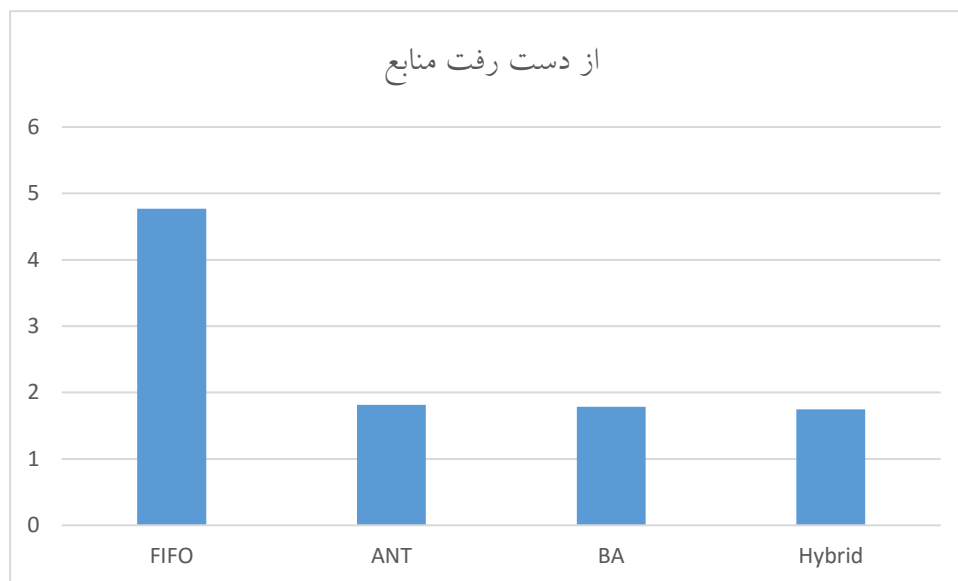
در سناریو دوم، هسته‌ی تولید کننده‌ی اعداد تصادفی برابر با ۱۰۰ در نظر گرفته خواهد شد. این عدد بصورت تصادفی انتخاب شده است. بنابراین برای اجرای هر ۴ الگوریتم، اعداد تصادفی یکسانی تولید خواهد شد که شرایط آزمایش را یکسان می‌کند. سپس هر چهار الگوریتم در شرایطی که صف ورودی ماشین‌های مجازی دارای ۲۰ ماشین مجازی هستند، آزمایش خواهد شد.

جدول ۶-۴ نتایج بدست آمده برای سناریو دوم

سرورهای روشن	زمان	از دست رفت	
FIFO	0.0781	4.7682	17
ANT	6.3906	1.8122	6
BA	2.3125	1.7847	6
Hybrid	1.85	1.7483	6

نتایج بدست آمده برای آزمایش اول در جدول ۶-۴ نشان داده شده است. با یک نگاه کلی می‌توان متوجه شد که الگوریتم ترکیبی توانسته است پاسخ بهتری را در زمان کمتر تولید کند. بر این

اساس هرچند بهبودی روش ترکیبی نسبت به الگوریتم زنبور عسل ناچیز بوده است، اما زمان تولید پاسخ در آن تا ۳۰ درصد کمتر بوده است. نمودارهای مقایسه‌ای جدول ۴-۶ در شکل‌های ۴-۴ تا ۴-۷ آمده است.



شکل ۴-۴ مقایسه‌ی از دست رفت منابع در سناریو دوم

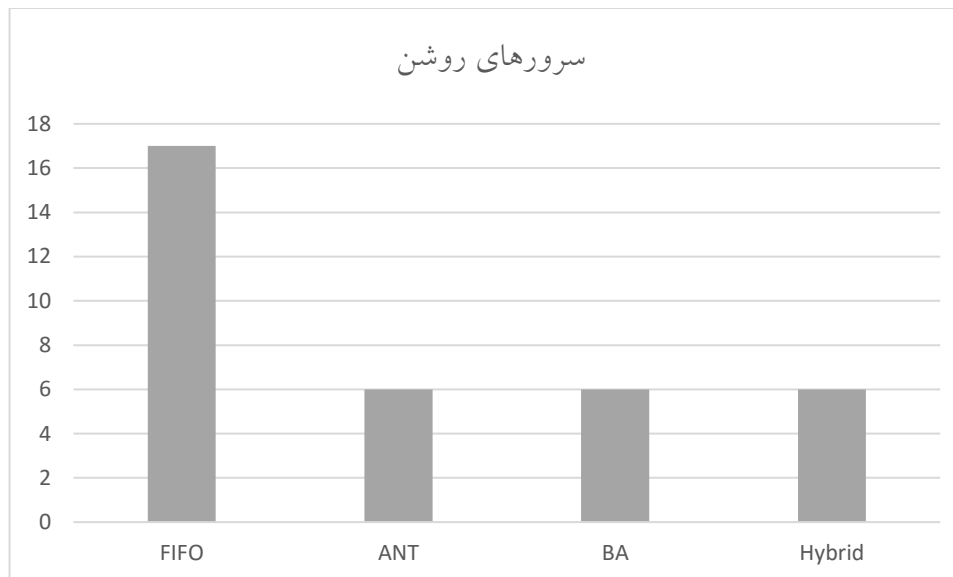
همانطور که در شکل ۴-۴ می‌توان مشاهده کرد، الگوریتم ترکیبی از نظر از دست رفت منابع با فاصله‌ی کمی از الگوریتم زنبور عسل در رتبه‌ی اول قرار گرفته است. پس از آن به ترتیب الگوریتم کلونی مورچگان و FIFO بوده‌اند.

بر اساس شکل ۴-۵، الگوریتم ترکیبی از نظر میزان زمان مصرفی در حدود ۳ دهم ثانیه از الگوریتم کلونی زنبور عسل بهتر عمل کرده و همچنین زمان بهتری نسبت به بهینه‌سازی کلونی مورچگان داشته است. طبق انتظار، الگوریتم FIFO دارای زمان اجرای بسیار پایینتری بوده است.

همچنین بر اساس آنچه که در شکل ۴-۶ نشان داده شده است، روش پیشنهادی همانند سایر روش‌های فراابتکاری حداقل تعداد سرورهای میزبان را روشن نگه داشته است و تعداد سرورهای روشن را در مقایسه با FIFO به یک سوم رسانده است.



شکل ۴-۵ مقایسه‌ی زمان اجرای الگوریتم‌ها در سناریو دوم



شکل ۴-۶ تعداد سرورهای روشن در روشن های مورد مقایسه در سناریو دوم

بر این اساس، الگوریتم ترکیبی پیشنهادی توانسته است در این آزمایش بر سایر الگوریتم‌ها تا حدودی غلبه کند. هرچند جواب‌های تولید شده از روش پیشنهادی شباهت زیادی به الگوریتم زنبور عسل داشته است، اما پاسخ را در زمان بسیار کمتری تولید می‌کند.

فصل پنجم: نتیجه گیری

۵-۱ مقدمه

در این فصل نتایج بدست آمده از فصل ۴ را به تحلیل خواهیم نشست و سپس مزایا و معیای روش پیشنهادی خود را مورد بررسی قرار خواهیم داد. برای مقایسه‌ی هرچه بهتر نتایج، در این فصل روش پیشنهادی با سایر الگوریتم‌های فراابتکاری همچون ژنتیک و بهینه‌سای مبتنی بر جغرافیای زیستی نیز مقایسه خواهد شد.

۵-۲ نتیجه گیری

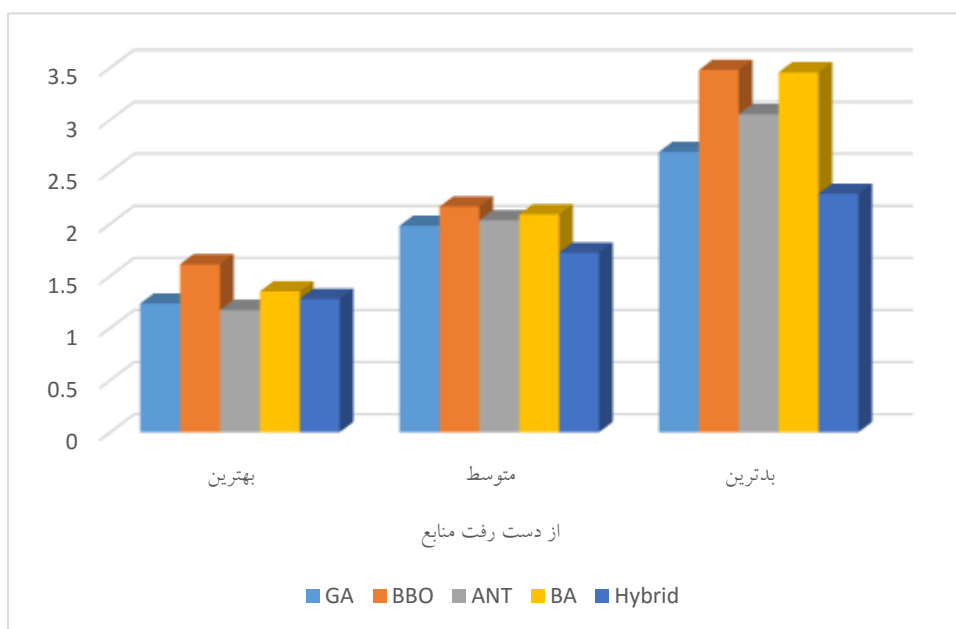
در فصل ۴ به مقایسه‌ی ریز نتایج پرداختیم. بدین منظور هسته‌های ایجاد اعداد تصادفی یکسانی را تولید کرده و در شرایطی مشابه به مقایسه‌ی آنها پرداختیم. در این فصل در این بخش به مقایسه‌ی روش پیشنهادی با الگوریتم‌های ژنتیک، بهینه‌سازی مبتنی بر جغرافیای زیستی، کلونی زنبور عسل و کلونی مورچگان برای ۱۰ اجرای متوالی خواهیم پرداخت. نتایج بدست آمده از ۱۰ اجرای متوالی در جدول ۵-۱ آمده است.

جدول ۵-۱ میانگین نتایج بدست آمده از ۱۰ آزمایش

	میانگین زمان			تعداد سروهای روشن			از دست رفت منابع	
	بدترین	متوسط	بهترین	بدترین	متوسط	بهترین	متوسط	بهترین
GA	8	6.6	6	2.6908	1.9808	1.2344		
BBO	10	8.1	7	3.4791	2.16523	1.6114		
ANT	8	7.2	6	3.0536	2.03357	1.1736		
BA	9	6.4	4	3.4563	2.09181	1.3514		
Hybrid	7	5.7	4	2.2887	1.72276	1.2792		

نتایج نشان داده شده در جدول ۵-۱ بصورت بصری در شکل‌های ۵-۱ تا ۵-۳ نشان داده شده است. بر اساس جدول ۵-۱ و آنچه که در شکل ۵-۱ نشان داده شده است، هرچند روش ترکیبی از لحاظ از دست رفت منابع در ۱۰ بار تکرار انجام شده نتوانسته است به بهترین پاسخ یعنی الگوریتم

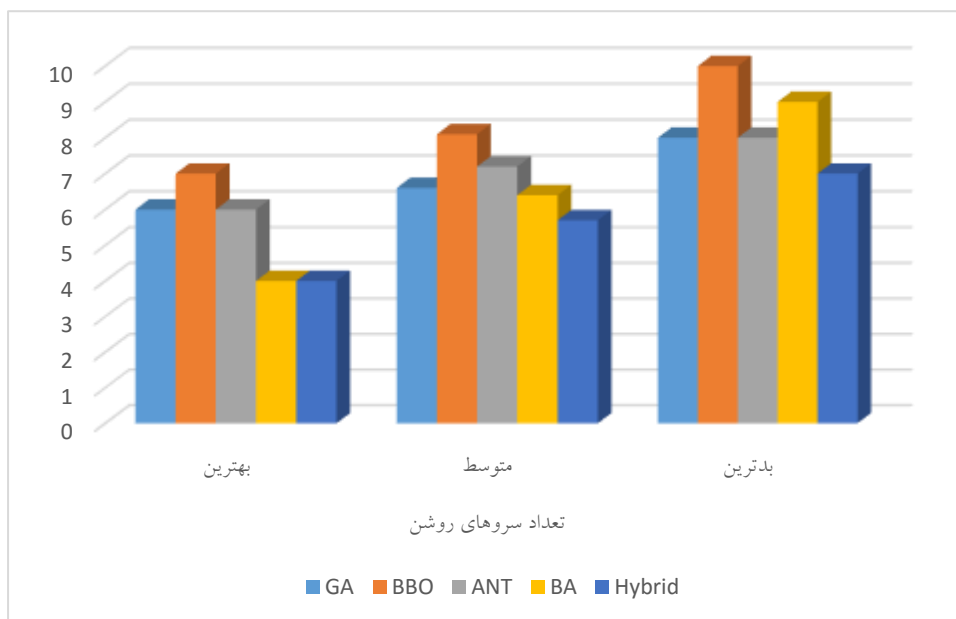
ژنتیک دست‌یابد، اما متوسط و بدترین حالت کسب شده توسط روش ترکیبی بطور ملاحظه‌ای پایین‌تری است. به عبارت دیگر روش پیشنهادی هرچند نتوانسته است بهترین پاسخ ممکن را بدست آورد، اما در ۱۰ بار تکرار انجام شده، متوسط و بدترین حالت آن کمتر بوده و در نتیجه پاسخ‌های با ثبات‌تری تولید شده است. این میزان بهبودی نسبت به سایر روش‌ها قابل توجه بوده و بنابراین می‌توان نتیجه گرفت که روش پیشنهادی از نظر رسیدن به پاسخ بهتر، توانایی بیشتری داشته است. پس از روش پیشنهادی، الگوریتم ژنتیک و کلونی مورچگان در جایگاه‌های بعدی قرار داشته و در نهایت الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی بدترین پاسخ‌ها را تولید کرده است.



شکل ۱-۵ مقایسه روش‌های فراابتکاری از نظر میزان از دست رفت منابع

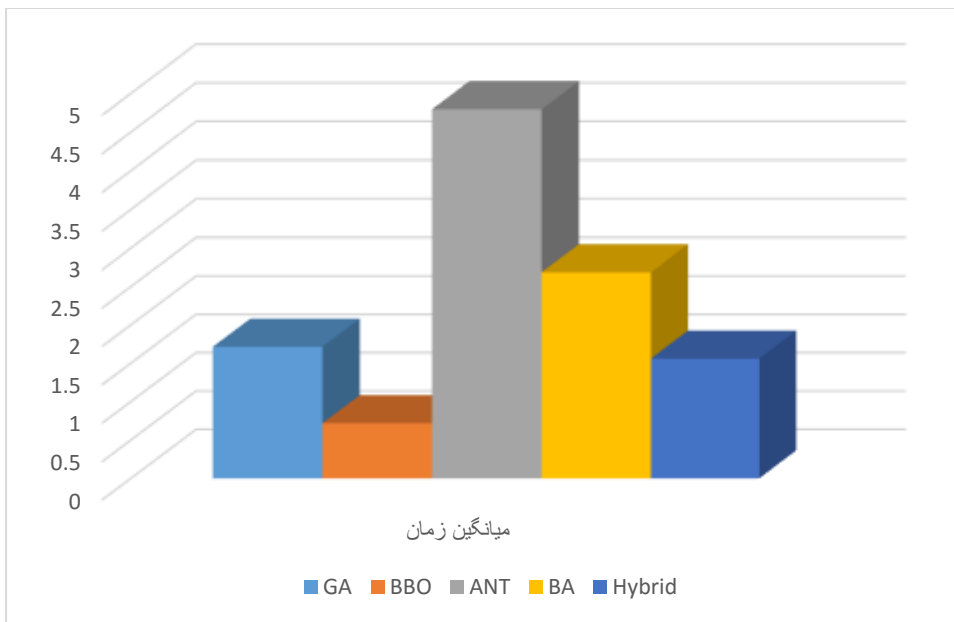
بر اساس جدول ۱-۵ و شکل ۲-۵ می‌توان نتیجه گرفت که روش پیشنهادی از نظر خاموش کردن تعداد سرورهای بلا استفاده نیز موفق بوده است. بر این اساس بهترین پاسخ تولید شده بطور مشترک با الگوریتم کلونی زنبور عسل، روش پیشنهادی بوده است که توانسته با روشن نگاه داشتن ۴ میزبان (سرور) به این پاسخ برسد. همچنین از لحاظ متوسط تعداد سرورهای روشن، این روش با

متوسط ۵,۷ سرور برای ۱۰ تکرار، کمترین میزان تعداد سرورها را داشته است. همچنین از لحاظ بدترین پاسخ تولید شده نیز، روش پیشنهادی دارای کارایی مناسبی بوده و بدترین پاسخ تولید شده از آن بطور قابل توجهی از سایر روش‌ها کمتر بوده است. بنابراین روش پیشنهادی بطور قابل ملاحظه‌ای توانسته است تعداد سرورهای روشن را بطور قابل ملاحظه‌ای کاهش دهد.



شکل ۵-۲ مقایسه روش‌های فراابتکاری از نظر تعداد سرورهای روشن

در نهایت بر اساس جدول ۵-۱ و شکل ۵-۳، زمان تولید مدل‌ها به عنوان معیاری نادقیق برای محاسبه‌ی سربار روش پیشنهادی نشان داده شده است. بر اساس شکل ۵-۳، هرچند پاسخ‌های تولیدی از الگوریتم ترکیبی در کمترین زمان ممکن تولید نشده است، اما از روش‌های کلونی زنبور عسل، کلونی مورچگان و الگوریتم ژنتیک در زمان بهتری تولید شده است و تنها زمان تولید پاسخ در آن از الگوریتم بهینه سازی مبتنی بر جغرافیای زیستی بیشتر بوده است. ولی به دلیل آنکه روش پیشنهادی پاسخ‌های بسیار بهتری را نسبت به الگوریتم بهینه سازی مبتنی بر جغرافیای زیستی تولید کرده است، می‌توان از این تفاوت چشم پوشی کرد.



شکل ۳-۵ مقایسه روش های فراابتکاری از نظر میانگین زمان تولید مدل

۳-۵ جمع بندی

نتایج نشان داده شده در فصل چهارم و همچنین نتایج نشان داده شده در بخش ۲-۵ نشان می دهد که روش پیشنهادی ترکیبی هرچند ممکن است در برخی مواقع پاسخ های بهتری نسبت به سایر الگوریتم ها تولید نکند، اما در مجموع از لحاظ از دست رفت منابع، زمان اجرا و تعداد سروهای روشن توانسته است کارایی بسیار مناسبی داشته باشد.

فصل ششم: کارهای آتی

۱-۶ مقدمه

در این فصل در ابتدا به بررسی کلی روش پیشنهادی پرداخته و مزایا و معایب آن را مورد بررسی قرار خواهیم داد، در نهایت بر اساس معایب برشمرده شده، زمینه‌های قابل کار در کارهای آتی مورد بررسی قرار خواهد گرفت.

۲-۶ بحث

در این پایان‌نامه به علت اهمیت و ضرورت بهینه‌سازی در مراکز داده‌ها به منظور کاهش از دست رفت منابع و طبع آن مصرف انرژی در آنها روشی فراابتکاری مبتنی بر ترکیبی از روش‌های فراابتکاری کلونی زنبور عسل و کلونی مورچگان پرداختیم. بر اساس اینکه کلونی زنبور عسل یک الگوریتم فراابتکاری تکاملی با نیاز به پاسخ اولیه است و الگوریتم کلونی زنبور عسل الگوریتمی با توانایی تولید پاسخ‌هاست، در این مطالعه با بکارگیری الگوریتم کلونی مورچگان برای تولید تعدادی جواب خوب و انتقال آن به الگوریتم زنبور عسل به عنوان پاسخ اولیه استفاده گردید به امید آنکه روش ترکیبی پاسخ‌های بهتری را به عنوان از دست رفت منابع تولید کند. با در نظر گرفتن معیارهای از دست رفت منابع، تعداد سرورهای روش و زمان تولید مدل، مشخص شد که روش فراابتکاری ترکیبی تولید شده، دارای پاسخ‌های بهتری نسبت به روش‌های ارائه شده رد مطالعات پیشین بوده است.

مزایای روش پیشنهادی

- سرعت مناسب
- توانایی در تولید پاسخ‌هایی مناسب از دست رفت منابع
- توانایی مناسب در حداقل سازی تعداد سرورهای روشن
- توانایی مناسب در تولید پاسخ‌های مورد نیاز در زمان کوتاه

- پاسخ‌ها تولید شده دارای ثبات بالایی است

معایب

- در برخی موارد توانایی تولید جواب بهینه را نداشته است
- بهترین پاسخ توسط این الگوریتم تولید نشده است
- برای پیاده سازی این الگوریتم نیاز به محاسبه‌ی سربار ناشی از انتقال ماشین‌های مجازی نیز باید در نظر رگفته شود.
- بر روی کاهش تعداد مهاجرت‌ها به عنوان یک هدف تمرکز نشده است.

۳-۶ پیشنهاد کارهای آینده

در روش پیشنهادی کنونی تنها بر روی مهاجرت ماشین‌های مجازی در بین میزبان‌های مختلف تمرکز شده است و بر هزینه‌ی ناشی از آنها تمرکز نشده است. در کارهای آتی سعی بر آن است که مسئله به گونه‌ای تعریف شود که کاهش تعداد مهاجرت‌های ماشین‌های مجازی نیز به عنوان هدف دیگر این مطالعه قرار داده شود. از طرف دیگر، انتقال ماشین‌های مجازی در بین میزبان‌ها دارای سربار است که این سربار لحاظ نشده است. بنابراین پیشنهاد می‌گردد که روش پیشنهادی در آینده بر روی یک تعدادی ماشین مجازی پیاده سازی شده و سربارهای ناشی از آن بر روی مراکز داده حساب گردد. همچنین می‌توان مصرف انرژی را نیز به عنوان هدف دیگر این مطالعه در آینده مورد نظر قرار داد به گونه‌ای که روش پیشنهادی در غالب یک روش چند هدفه ارائه گردد.

1. Zhang, Q., L. Cheng, and R. Boutaba, *Cloud computing: state-of-the-art and research challenges*. Journal of internet services and applications, 2010. **1**(1): p. 7-18.
2. Lai, I.K., S.K. Tam, and M.F. Chan, *Knowledge cloud system for network collaboration: A case study in medical service industry in China*. Expert Systems with applications, 2012. **39**(15): p. 12205-12212.
3. Shieh, A., et al. *Seawall: Performance Isolation for Cloud Datacenter Networks*. in *HotCloud*. 2010.
4. Goldberg, R.P., *Survey of virtual machine research*. Computer, 1974. **7**(6): p. 34-45.
5. Xing, Y. and Y. Zhan, *Virtualization and cloud computing*, in *Future Wireless Networks and Information Systems*. 2012, Springer. p. 305-312.
6. Khanna, G., et al. *Application performance management in virtualized server environments*. in *2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006*. 2006. IEEE.
7. Barlaskar, E., N.A. Singh, and Y. Jayanta, *Energy optimization methods for Virtual Machine Placement in Cloud Data Center*. ADBU Journal of Engineering Technology, 2015. **1**.
8. Gao, Y., et al., *A multi-objective ant colony system algorithm for virtual machine placement in cloud computing*. Journal of Computer and System Sciences, 2013. **79**(8): p. 1230-1242.
9. Sosinsky, B., *Cloud computing bible*. Vol. 762. 2010: John Wiley & Sons.
10. Rajkumar, B., et al., *Cloud computing and emerging IT platforms*. Future Generation Computer Systems. Elsevier Press, Inc, 2009.
11. Forrest, W. and C. Barthold, *Clearing the air on cloud computing*. Discussion Document from McKinsey and Company, 2009.

12. Barroso, L.A., J. Clidaras, and U. Hölzle, *The datacenter as a computer: An introduction to the design of warehouse-scale machines*. Synthesis lectures on computer architecture, 2013. **8**(3): p. 1-154.
13. Koomey, J., *Growth in data center electricity use 2005 to 2010*. A report by Analytical Press, completed at the request of The New York Times, 2011. **9**.
14. Uhlig, R., et al., *Intel virtualization technology*. Computer, 2005. **38**(5): p. 48-56.
15. Barham, P., et al. *Xen and the art of virtualization*. in *ACM SIGOPS Operating Systems Review*. 2003. ACM.
16. Glover, F.W. and G.A. Kochenberger, *Handbook of metaheuristics*. Vol. 57. 2006: Springer Science & Business Media.
17. Luke, S., *Essentials of metaheuristics*. 2013: Lulu Com.
18. Karaboga, D., *An idea based on honey bee swarm for numerical optimization*. 2005, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
19. Tabakhi, S., P. Moradi, and F. Akhlaghian, *An unsupervised feature selection algorithm based on ant colony optimization*. Engineering Applications of Artificial Intelligence, 2014. **32**: p. 112-123.
20. Dorigo, M. and L.M. Gambardella, *Ant colony system: a cooperative learning approach to the traveling salesman problem*. IEEE Transactions on evolutionary computation, 1997. **1**(1): p. 53-66.
21. Chaisiri, S., B.-S. Lee, and D. Niyato. *Optimal virtual machine placement across multiple cloud providers*. in *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*. 2009. IEEE.
22. Piao, J.T. and J. Yan. *A network-aware virtual machine placement and migration approach in cloud computing*. in *2010 Ninth International Conference on Grid and Cloud Computing*. 2010. IEEE.
23. Le, K., et al. *Reducing electricity cost through virtual machine placement in high performance computing clouds*. in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. 2011. ACM.
24. Ilkhechi, A.R., I. Korpeoglu, and Ö. Ulusoy, *Network-aware virtual machine placement in cloud data centers with multiple traffic-intensive components*. Computer Networks, 2015. **91**: p. 508-527.
25. Yuchi, X. and S. Shetty. *Enabling security-aware virtual machine placement in IaaS clouds*. in *Military Communications Conference, MILCOM 2015-2015 IEEE*. 2015. IEEE.

26. Fu, X. and C. Zhou, *Virtual machine selection and placement for dynamic consolidation in Cloud computing environment*. Frontiers of Computer Science, 2015. **9**(2): p. 322-330.
27. Fukunaga, T., S. Hirahara, and H. Yoshikawa. *Virtual machine placement for minimizing connection cost in data center networks*. in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2015. IEEE.
28. Zheng, Q., et al., *Virtual machine consolidated placement based on multi-objective biogeography-based optimization*. Future Generation Computer Systems, 2016. **54**: p. 95-122.
29. Stefanello, F., et al. *A Biased Random-key Genetic Algorithm for Placement of Virtual Machines across Geo-Separated Data Centers*. in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. 2015. ACM.
30. Joshi, S. and S. Kaur. *Cuckoo search approach for virtual machine consolidation in cloud data centre*. in *Computing, Communication & Automation (ICCCA), 2015 International Conference on*. 2015. IEEE.
31. Coutinho, R.d.C., et al., *Optimizing virtual machine allocation for parallel scientific workflows in federated clouds*. Future Generation Computer Systems, 2015. **46**: p. 51-68.
32. Kansal, N.J. and I. Chana, *Energy-aware Virtual Machine Migration for Cloud Computing-A Firefly Optimization Approach*. Journal of Grid Computing, 2016. **14**(2): p. 327-345.
33. Singh, A. and M. Hemalatha, *Virtual Machine Placement by Using Honey Bee Forager Algorithm in Cloud Computing*. Karpagam Journal of Computer Science, Vol7 (4), 2013: p. 209.
34. Wang, S., et al. *Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers*. in *Parallel and Distributed Systems (ICPADS), 2013 International Conference on*. 2013. IEEE.
35. Lopez-Pires, F. and B. Baran, *Virtual machine placement literature review*. arXiv preprint arXiv:1506.01509, 2015.
36. Coello, C.C., G.B. Lamont, and D.A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. 2007: Springer Science & Business Media.