

An Improved Grey Wolves Optimization Algorithm for Workflow Scheduling in Cloud Computing Environment

Ali Mohammadzadeh¹, Mohammad Masdari^{2*}, Farhad Soleimanian Gharehchopogh³ and Ahmad Jafarian⁴

1- Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran.

2*- Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran.

3- Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran.

4- Department of Mathematics, Urmia Branch, Islamic Azad University, Urmia, Iran.

¹ A.Mohammadzadeh@iaurmia.ac.ir, ^{2*} M.Masdari@iaurmia.ac.ir, ³ farhad@iaurmia.ac.ir
and ⁴ A.Jafarian@iaurmia.ac.ir

Corresponding author address: Mohammad Masdari, Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran.

Abstract- In this paper, an improved metaheuristic algorithm based on Grey Wolf Optimizer algorithm is proposed for solve optimization problems. In the proposed algorithm, the weakest of wolves would be excluded from the population and included with other wolves from the initial population. The choice of placed wolves would be random or fitness basis. In this algorithm, the spatial fitness of the particles is studied in each iteration, and in the case of improving the fitness basis, the wolves are moving toward the target. Otherwise, they remain in the last fit state. This algorithm is designed to improve the search performance against various issues, increase convergence speed, and avoid local optimal. Simulation has been done in Matlab software and it has been implemented with 23 different optimization mathematical functions. By examining the results and comparing the results of the results obtained from the new algorithm, Grey wolf optimizer algorithm, and several other algorithms, we conclude that by adjusting the parameters, the performed improvements have a significant effect on the performance of the algorithm on different functions.

Keywords- Evolutionary algorithm, Grey wolf optimizer, Meta-heuristic, Optimization.

ارائه یک الگوریتم بهبود یافته بهینه سازی گرگ های خاکستری برای زمان بندی جریان کار در محیط محاسبات ابری

علی محمدزاده^۱، محمد مصدری^{۲*}، فرهاد سلیمانان قره چیق^۳، احمد جعفریان^۴

۱- گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران.

۲* گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران.

۳- گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران.

۴- گروه ریاضی، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران.

¹ A.Mohammadzadeh@iaurmia.ac.ir, ^{2*} M.Masdari@iaurmia.ac.ir, ³ Farhad@iaurmia.ac.ir and ⁴ A.Jafarian@iaurmia.ac.ir

* نشانی نویسنده مسئول: محمد مصدری، ارومیه، دانشگاه آزاد اسلامی واحد ارومیه، دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر.

چکیده- در این مقاله یک الگوریتم فرا ابتکاری بهبود یافته بر اساس الگوریتم فرا ابتکاری گرگ های خاکستری به منظور حل مسائل بهینه سازی ارائه می گردد. در الگوریتم پیشنهادی ضعیف ترین گرگ ها از جمعیت حذف شده و با گرگ های دیگری از جمعیت اولیه جاگذاری می شود. انتخاب گرگ های جاگذاری شده به صورت تصادفی یا بر اساس برازندگی خواهد بود. در این الگوریتم برازندگی مکان ذرات در هر تکرار بررسی شده و در صورت بهبود برازندگی، گرگ ها به سمت هدف حرکت می کنند، در غیر این صورت در آخرین وضعیت مناسب باقی می ماندند. این الگوریتم باهدف بهبود عملکرد جستجو در مقابله با مسائل مختلف، افزایش سرعت همگرایی و جلوگیری از گیر افتادن در بهینه محلی ارائه شده است. شبیه سازی در نرم افزار متلب بر روی ۲۳ تابع استاندارد ریاضی بهینه سازی مختلف اجرا شده است. با بررسی عملکرد و مقایسه آماری نتایج به دست آمده از الگوریتم جدید با الگوریتم گرگ های خاکستری پایه و چند الگوریتم دیگر به این نتیجه می رسیم که با تنظیم مناسب پارامترها بهبودهای انجام شده تأثیر بسزایی در عملکرد الگوریتم بر روی توابع مختلف دارند.

واژه های کلیدی: الگوریتم های تکاملی، الگوریتم گرگ های خاکستری، فرا ابتکاری، بهینه سازی.

۱- مقدمه

جستجوی تصادفی کل فضای جستجو به منظور دستیابی به بخش های مطلوب تر است. مرحله بهره برداری جستجوی دقیق تر قسمت های مطلوب شناسایی شده از فضای جستجو است. یکی از مسائل چالش برانگیز روش های فرا ابتکاری، پیدا کردن موازنه مناسب بین این دو مرحله است.

در ادامه قصد بررسی اجمالی چند مورد از الگوریتم های فرا ابتکاری را که در چند دهه گذشته مطرح شده اند داریم. الگوریتم بهینه سازی ازدحام ذرات [۶] الهام گرفته شده از رفتار جمعی پرندگان است و هر ذره در آن با تغییر در مقدار سرعت و جهتش سعی در یافتن بهترین پاسخ در فضای جستجو دارد. با الهام گرفتن از تولیدمثل فاخته ها نیز الگوریتم بسیار کارای جستجوی فاخته توسط یانگ و همکاران در سال ۲۰۰۹ پیشنهاد شد [۷]. دبری بر

در چند دهه اخیر، محققان رفتارهای گروهی از موجودات را که الگوهای مناسبی برای حل مسائل بهینه سازی دارند، مدل کرده اند. این الگوریتم ها با سادگی تمام، معمولاً از رفتارهای حیوانات و پدیده های فیزیکی الهام می گیرند [۱]. در حل مسائل بهینه سازی با ابعاد بالا معمولاً الگوریتم های فرا ابتکاری، عملکرد بهتری نسبت به روش های منطقی و ریاضی دارند [۲]، به همین دلیل استفاده از آنها رو به رشد بوده و هر ساله الگوریتم های جدیدتری از قبیل نهنگ کوهان دار [۳] و سینوس کسینوس [۴] به منظور رسیدن به سرعت و دقت توسعه داده می شوند.

تقسیم فرآیند جستجو به دو مرحله شناسایی و بهره برداری، ویژگی مشترک بین روش های فرا ابتکاری است [۵]. مرحله شناسایی

در الگوریتم بهینه‌سازی سینوس و کسینوس همیشه بهترین جواب نشان‌دهنده مقصد است. بنابراین موج‌های جستجو از بهینه اصلی مسئله منحرف نمی‌شوند و الگوی نوسانی به آن اجازه می‌دهد تا فضای جستجو را در اطراف بهینه مسئله به خوبی جستجو کند و دقت مناسبی در به دست آوردن بهینه داشته باشد [۴]. رفتار جستجوگر عنکبوت را در الگوریتم بهینه‌سازی عنکبوت اجتماعی^۳ می‌توان به‌عنوان جنبش جمعی ذرات به سمت منبع غذایی توصیف کرد. عنکبوت، ارتعاشات منتشرشده در فضای جستجو را برای تعیین جهت بالقوه یک منبع غذایی دریافت و تحلیل می‌کند. هنگامی که یک عنکبوت به موقعیت جدیدی حرکت می‌کند، یک ارتعاش تولید می‌کند که در سراسر فضا منتشر می‌شود. هر لرزش عنکبوت دارای اطلاعاتی از یک عنکبوت است و دیگران این اطلاعات ارتعاش را دریافت می‌کنند [۱۴].

در سال‌های اخیر مقالات متعددی در ارتباط با گرگ‌های خاکستری [۱۵] ارائه شده است. در الگوریتم بهبودیافته جدیدی جمعیت اولیه گرگ‌ها به چندین جمعیت مستقل بر اساس ساختار اصلی الگوریتم پایه تقسیم می‌شود و بین چندین بهینه‌ساز موازی گرگ خاکستری به منظور برقراری ارتباط، استراتژی ارتباطی خاصی استفاده شده است [۱۶]. در برخی از کارها نیز از تئوری آشوب برای مقادیر تصادفی الگوریتم استفاده کرده‌اند که نتایج مناسبی نیز به دست آمده است [۱۷]. اولین استراتژی، وزن دار کردن طعمه‌ها است. در ادامه از قوانین اخترازی برای هدایت مناسب‌تر گرگ‌های خاکستری به سمت طعمه‌های مطلوب‌تر استفاده می‌کند. از این رو گرگ‌ها هم‌زمان مراحل شناسایی و بهره‌برداری را انجام می‌دهند. در استراتژی سوم ویژگی‌ها و مزایایی دو استراتژی قبل را باهم اعمال نموده‌اند. کامبوج با ترکیب الگوریتم ازدحام ذرات و گرگ‌های خاکستری توانست به نتایج بهتری برسد. در ابتدا موقعیت ذرات ازدحام توسط الگوریتم ازدحام ذرات و در گام بعدی توسط گرگ‌های خاکستری تغییر می‌یابند [۱۸]. در الگوریتم گرگ‌های خاکستری باینری در ابتدا بهترین سه پاسخ به دست آمده باینری شده و تقاطع تصادفی بر روی این سه پاسخ انجام می‌گیرد. در ادامه بر اساس تابع سیگنویید مکان ذرات تغییر پیدا می‌کند [۱۹].

در این مقاله هدف، ارائه یک الگوریتم بهبودیافته گرگ‌های خاکستری است. این الگوریتم بهبودیافته بر اساس تغییر در جمعیت گرگ‌ها و تأثیر میزان برانزندی هر گرگ است. با بهبودهای انجام گرفته شاهد نتایج خوبی نسبت به روش پایه و چندین الگوریتم دیگر بوده‌ایم. در ادامه، بخش دوم خلاصه‌ای از الگوریتم گرگ‌های خاکستری پایه است. بخش سوم الگوریتم پیشنهادی و بخش چهارم

اساس تعاملات همزیستی میان جانوران برای بقا و تولیدمثل، الگوریتمی با عنوان جستجوی ارگانسیم‌های همزیستی ارائه کرد. تعاملات همزیستی به سه نوع همکاری، هم‌سفرگی و انگلی تقسیم می‌شود [۸]. الگوریتم شکار نهنگ کوهان‌دار [۳] توسط میرجلیلی برای رسیدن به بهترین پاسخ به صورت تصادفی وارد فاز شناسایی یا بهره‌برداری می‌شود. در فاز بهره‌برداری بر اساس یک رابطه کسینوسی به پاسخ بهینه نزدیک می‌شود. در مقاله دیگری از میرجلیلی [۹] الگوریتم بهینه‌ساز چندوجهی^۱ بر اساس سه مفهوم سیاه‌چاله، سفید چاله و گرم چاله از کیهان‌شناسی ارائه شده است. از این سه مفهوم برای اکتشاف، بهره‌برداری و جستجوی محلی استفاده کرده است. الگوریتم بهینه‌سازی پروانه - شعله^۲ [۱۰] الگوریتم دیگری از میرجلیلی است. این الگوریتم مبتنی بر جمعیت است. حرکت ذرات در یک فضای یک، دو یا چندبعدی است. برای پیاده‌سازی حرکت پروانه - شعله از تابع لگاریتمی و کسینوسی استفاده شده است.

بانسال و همکاران الگوریتم میمون‌های عنکبوتی [۱۱] را از تقسیم و همجوشی ساختار اجتماعی میمون‌های عنکبوتی الهام گرفته‌اند. به منظور کاهش رقابت بر سر تغذیه، گروه‌ها به زیرگروه‌های کوچک‌تر شکسته می‌شوند. مراحل الگوریتم به این صورت است که گروه اصلی در ابتدا شروع به تغذیه و ارزیابی فاصله تا غذا می‌کند. بر اساس فاصله‌ای که تا هدف دارند اعضای گروه مکان خود را تغییر داده و دوباره فاصله از مبدأ غذا را ارزیابی می‌کنند. رهبر زیرگروه‌ها موقعیت خود را بر اساس بهترین موقعیت اعضای گروه به‌روزرسانی می‌کند و در صورتی که مکان آن به تعداد تکرار مشخصی تغییر پیدا نکند، در این صورت تمام اعضای گروه در جهات مختلف به دنبال غذا می‌گردند. رهبر عمومی همیشه موقعیت خود را به بهترین موقعیت تغییر می‌دهد ولی در صورت داشتن رکود، جمعیت را به زیرگروه‌هایی با اندازه کوچک‌تر تقسیم می‌کند. در الگوریتم مورچگان از بین چندین مسیر، مسیر با کوتاه‌ترین فاصله بر اساس میزان غلظت فرومون انتخاب می‌شود. اثر فرومون در کلونی مورچگان مانند اطلاعات عددی توزیع شده در فضای جستجو می‌باشند و مورچه‌ها تجربه خود را بر اساس آن به یکدیگر انتقال می‌دهند. مورچه‌ها برای ایجاد پاسخ از اطلاعات داخل مسئله، اثرات فرومون مصنوعی که در طول فرآیند حل مسئله در حال تغییر است و تجارب به دست آمده توسط عامل‌های جستجو استفاده می‌کنند [۱۲]. در [۱۳] الگوریتم بهینه‌سازی جدید الهام گرفته از رفتار ایستا و پویای ازدحام سنجاقک‌ها ارائه گردیده و دو فاز اصلی اکتشاف و بهره‌برداری با مدل‌سازی ارتباط اجتماعی سنجاقک‌ها در هدایت، جستجو برای غذا و اجتناب از دشمن انجام می‌گیرد.

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (۳)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (۴)$$

بردار a از ۲ تا ۰ به صورت خطی در طول دوره تکرار در هر دو فاز اکتشاف و بهره‌برداری کاهش می‌یابد. r بردار تصادفی مابین ۰ و ۱ است. با توجه به تصادفی بودن بردارهای r_1 و r_2 ، گرگ‌ها می‌توانند موقعیت خود را در داخل فضایی که طعمه را در بر گرفته به صورت تصادفی و با استفاده از روابط ۵ و ۶ تغییر دهند. همین مفهوم را می‌توان به یک فضای جستجوی n بعدی تعمیم داد. در این حالت گرگ‌های خاکستری پیرامون بهترین راه‌حل به دست آمده در ابعادی بیشتر از ابعاد مکعب حرکت می‌کنند.

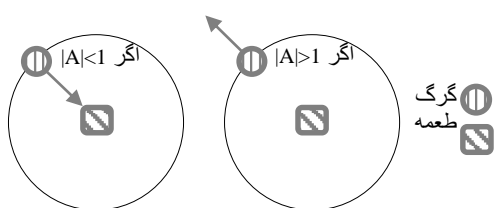
شکار گرگ‌های خاکستری اغلب توسط آلفا هدایت می‌شود. گرگ‌های بتا و دلتا نیز گاهی اوقات در شکار شرکت می‌کنند. به منظور مدل نمودن این رفتار سه مورد از بهترین راه‌حل‌های به دست آمده را ذخیره کرده و دیگر عوامل جستجو را طبق رابطه ۷ وادار می‌کنیم تا موقعیت خود را با توجه به موقعیت بهترین عوامل جستجو به‌روزرسانی نمایند.

$$\begin{aligned} \vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{aligned} \quad (۵)$$

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \end{aligned} \quad (۶)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (۷)$$

در این الگوریتم پیاده‌سازی فاز بهره‌برداری یا حمله که در صورت متوقف شدن طعمه اتفاق می‌افتد، با کاهش مقدار متغیر a از ۲ به ۱ انجام می‌شود. مقدار A نیز وابسته به a است، از این رو کاهش پیدا می‌کند. با کاهش مقدار A گرگ‌ها مجبور به حمله به سمت طعمه می‌شوند. همچنین برای جلوگیری از گیر افتادن در حداقل محلی، فاز شناسایی ارائه می‌شود. گرگ‌ها برای جستجوی شکار از یکدیگر فاصله گرفته و برای حمله به یکدیگر نزدیک شده و همکاری می‌کنند. برای شبیه‌سازی این واگرایی از بردار A با مقادیر تصادفی بزرگ‌تر از ۱ و یا کوچک‌تر از ۱- استفاده می‌کنیم. شکل ۱ بیان‌کننده این مسئله است.



شکل ۱- فاز اکتشاف در مقابل بهره‌برداری

شبیه‌سازی و نتایج را بیان خواهد نمود. در انتها بخش پنجم بیان نتیجه‌گیری و کارهای آتی است.

۲- الگوریتم فرا ابتکاری گرگ‌های خاکستری

الگوریتم گرگ‌های خاکستری با الهام از زندگی اجتماعی و شکار گرگ‌های خاکستری از چهار نوع گرگ برای شبیه‌سازی سلسله‌مراتب رهبری استفاده می‌کند. گرگ‌های خاکستری یک سلسله‌مراتب بسیار سخت اجتماعی دارند. رهبران گروه یک جنس نر و یک ماده هستند که آلفا نامیده می‌شوند. آلفا عمدتاً مسئول تصمیم‌گیری در مورد شکار، محل خواب، زمان بیداری و غیره هست. تصمیمات آلفا به گروه دیکته شده است. باین حال نوعی رفتار دموکراتیک نیز مشاهده شده است که در آن آلفا از دیگر گرگ‌های گروه پیروی می‌کند. در تجمع گروهی، کل گروه، آلفا را با نگرداشتن دم خود به سمت پایین به رسمیت می‌شناسد. با توجه به این که دستورات آلفا باید توسط گروه پیروی شود. گرگ‌های آلفا تنها مجاز به انتخاب همسر در گروه هستند. جالب توجه این است که آلفا لزوماً قوی‌ترین عضو گروه نیست، اما بهترین عضو از نظر مدیریت گروه است. این نشان می‌دهد که سازمان و نظم یک گروه از قدرت آن بسیار مهم‌تر است [۱۵].

سطح دوم در سلسله‌مراتب گرگ‌های خاکستری، بتا است. گرگ بتا، مشاور آلفا و سازمان دهنده گروه است. بتا دستورات آلفا را در سراسر گروه اجرا کرده و بازخورد آن را به آلفا ارجاع می‌دهد. پایین‌ترین رتبه را در بین گرگ‌های خاکستری، امگا دارد. امگا نقش قربانی دارد. آن‌ها آخرین گروه از گرگ‌هایی هستند که مجاز به خوردن هستند. اگر گرگی آلفا، بتا و یا امگا نباشد، زیردست (و یا دلتا در برخی منابع) نامیده می‌شود. گرگ دلتا باید به آلفا و بتا گزارش دهد، اما بر امگا تسلط دارد.

علاوه بر سلسله‌مراتب اجتماعی، شکار گرگ‌های خاکستری دارای سه مرحله ردیابی، تعقیب و نزدیک شدن به طعمه است. برای مدل‌سازی سلسله‌مراتب اجتماعی گرگ‌ها بهترین پاسخ را آلفا و از بین بهترین راه‌حل‌ها، دومین و سومین را بتا و دلتا در نظر می‌گیریم. بقیه راه‌حل‌های کاندیدا را امگا در نظر می‌گیریم. بهینه‌سازی توسط آلفا، بتا و دلتا هدایت می‌شود و گروه چهارم از این سه گروه پیروی می‌کند. مدل‌سازی رفتار محاصره گرگ‌ها از روابط ۱ و ۲ استفاده می‌کند [۱۵]:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (۱)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (۲)$$

در این روابط، t تعداد تکرار فعلی، A و C بردارهای ضریب، بردار موقعیت شکار و X بردار موقعیت یک گرگ است. برای محاسبه بردارهای A و C از روابط ۳ و ۴ استفاده می‌شود.

بیشتری گرگ‌های جاگذاری شده به صورت تصادفی و در تکرارهای آخر با احتمال بالاتری بر اساس برازندگی است. همچنین در هر تکرار برای تغییر مکان ذرات، برازندگی مکان جدید با برازندگی آخرین مکان مناسب که بیشترین برازندگی را داشته مقایسه می‌شود. در صورتی که برازندگی مکان جدید بیشتر باشد تغییر مکان انجام می‌شود و در غیر این صورت مکان با بیشترین برازندگی، مکان جدید خواهد بود. شبه کد این الگوریتم ترکیبی در جدول ۲ قابل مشاهده بوده و از این پس با نام IGWO استفاده می‌کنیم.

جدول ۲ - شبه کد الگوریتم پیشنهادی

```

1. Set the initial values of the population size  $n$ , parameter  $a$ , coefficient vectors  $A$  and  $C$ , and the maximum number of iterations  $Maxiter$ .
2. Set  $t = 0$ .
3. for ( $i = 1 : n$ ) do
4. Generate an initial population  $X_i(t)$  randomly.
5. Evaluate the fitness function of each search agent (solution)  $f(X_i)$ .
6. end for
7. Assign the values of the first, second and the third best solution  $X_\alpha, X_\beta$  and  $X_\delta$ , respectively.
8.  $r$  is random number.
9. if ( $r < 1-t/Maxiter$ )
10. Replace the values of the forth solution with Random Selected solution, respectively.
11. else Replace the values of the forth solution with  $X_\alpha$  solution, respectively.
12. end if
13. repeat
14. for ( $i = 1 : n$ ) do
15. set temp = Calculate each search agent in the population as shown in Eq. (7).
16. if ( $fitness(temp) > fitness(pBest)$ )
17. Set  $pBest = temp$ .
18. Set  $newPosition = pBest$ .
19. end if
20. Decrease the parameter  $a$  from 2 to 0.
21. Update the coefficients  $A, C$  as shown in Eq. (3) and (4), respectively.
22. Evaluate the fitness function of each search agent (vector)  $f(X_i)$ .
23. end for
24. Update the vectors  $X_\alpha, X_\beta$  and  $X_\delta$ .
25. Set  $t = t + 1$ .
26. until ( $t \geq Maxiter$ ). {Termination criteria are satisfied}
27. Produce the best solution  $X_\alpha$ .
    
```

۴- شبیه‌سازی و نتایج

شبیه‌سازی الگوریتم پیشنهادی و گرگ‌های خاکستری پایه و چندین الگوریتم جدید دیگر در نرم‌افزار متلب ۲۰۱۷ و بر روی ۲۳ تابع استاندارد ریاضی بهینه‌سازی که با عنوان CEC 2005 ارائه شده انجام گرفته است [۳، ۴، ۱۰، ۱۴، ۱۵، ۲۰-۲۸]. این توابع محک

جزء تأثیرگذار دیگر بر فرآیند شناسایی مقدار C است. مقدار این بردار عددی تصادفی در بازه $[0, 2]$ است. این مقدار تصادفی تأثیر موقعیت طعمه را در تعیین فاصله، شدت ($C > 1$) یا ضعف ($C < 1$) می‌بخشد. این بردار را نیز می‌توان به عنوان تأثیر موانعی در نظر گرفت که از نزدیک شدن به طعمه در طبیعت جلوگیری می‌کنند. شبه کد این الگوریتم در جدول ۱ ارائه شده است.

جدول ۱ - شبه کد الگوریتم گرگ‌های خاکستری

```

1. Set the initial values of the population size  $n$ , parameter  $a$ , coefficient vectors  $A$  and  $C$ , and the maximum number of iterations  $Maxiter$ .
2. Set  $t = 0$ .
3. for ( $i = 1 : n$ ) do
4. Generate an initial population  $X_i(t)$  randomly.
5. Evaluate the fitness function of each search agent (solution)  $f(X_i)$ .
6. end for
7. Assign the values of the first, second and the third best solution  $X_\alpha, X_\beta$  and  $X_\delta$ , respectively.
8. repeat
9. for ( $i = 1 : n$ ) do
10. Update each search agent in the population as shown in Eq. (7).
11. Decrease the parameter  $a$  from 2 to 0.
12. Update the coefficients  $A, C$  as shown in Eq. (3) and (4), respectively.
13. Evaluate the fitness function of each search agent (vector)  $f(X_i)$ .
14. end for
15. Update the vectors  $X_\alpha, X_\beta$  and  $X_\delta$ .
16. Set  $t = t + 1$ .
17. until ( $t \geq Maxiter$ ). {Termination criteria are satisfied}
18. Produce the best solution  $X_\alpha$ .
    
```

۳- الگوریتم پیشنهادی

در الگوریتم گرگ‌های خاکستری جمعیت به سمت پاسخ‌های بهینه یا به عبارتی گرگ‌های آلفا، بتا و دلتا حرکت می‌کنند. در هر تکرار ذرات بهینه بر اساس تابع برازندگی مشخص می‌شوند و برترین ذره آلفا نامیده می‌شود. هر بعد از مکان جدید برابر میانگین بعدهای متناظر ذرات برتر است. که به طور کامل در بخش قبل در روابط ۱ تا ۷ توضیح داده شد. جابجایی ذرات در هر مرحله بدون توجه به میزان برازندگی انجام می‌شود و به عبارتی غیر حریصانه است.

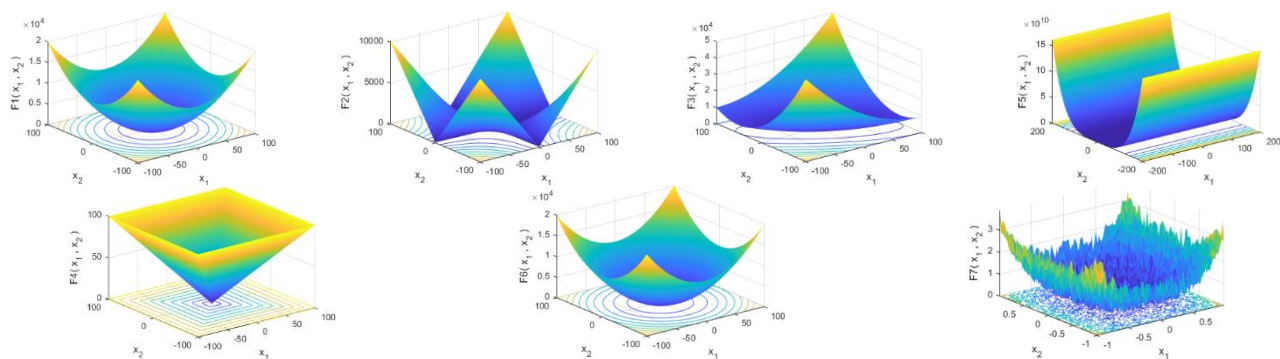
در این الگوریتم هدف، تغییر جمعیت اولیه در جهت داشتن جمعیتی با برازندگی بالاتر است. برای این منظور گرگ‌های امگا شناسایی شده و از جمعیت اولیه در هر تکرار حذف می‌شوند. برای داشتن جمعیت با برازندگی بالا به منظور همگرایی سریع‌تر به پاسخ، گرگ‌های امگا را با بهترین ذرات یا با گرگ‌های دیگری که از بین جمعیت به صورت تصادفی انتخاب می‌شوند جاگذاری می‌کنیم. ذرات تصادفی با هر درجه برازندگی می‌توانند باشند. در تکرارهای اولیه با احتمال

الگوریتم بهینه‌سازی گرگ‌های خاکستری پایه، پیشنهادی و چندین الگوریتم جدید را بر روی توابع تک‌نمایی نشان می‌دهد.

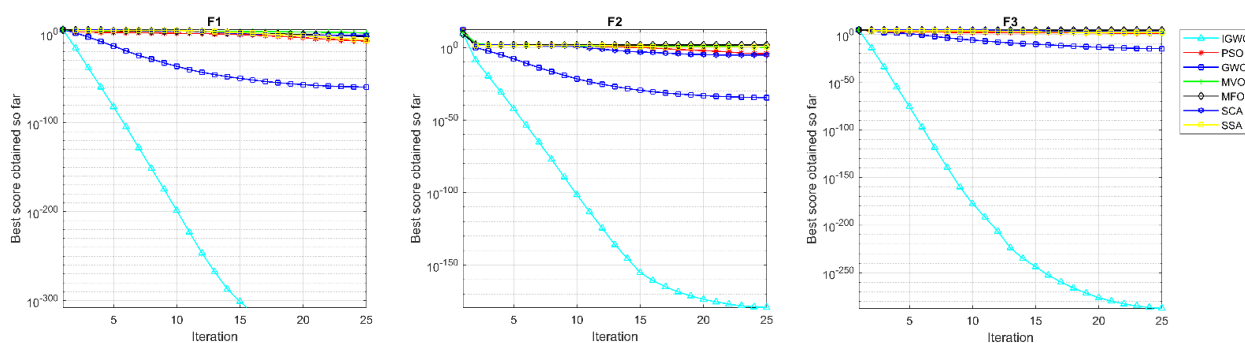
جدول ۳- توابع محک تک‌نمایی

f_{min}	محدوده	ابعاد	توابع هزینه
0		30	$f_1(x) = \sum_{i=1}^n x_i^2$
0	[-10,10]	30	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
0	[-100,100]	30	$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$
0	[-100,100]	30	$f_4(x) = \max_{1 \leq i \leq n} x_i $
0	[-30,30]	30	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$
0	[-100,100]	30	$f_6(x) = \sum_{i=1}^n [(x_i + 0.5)]^2$
0	[-1.28,1.28]	30	$f_7(x) = \sum_{i=1}^n ix_i^4 + random[0,1]$

اشکال ۳ و ۴ نیز نمودار همگرایی مربوط به بهترین پاسخ الگوریتم گرگ‌های خاکستری پایه و پیشنهادی و دیگر الگوریتم‌ها است. در اشکال ۳، ۴، ۶ و ۸ تعداد تکرارها ۴۰ برابر کمتر نمایش داده شده، از این رو تعداد تکرارها در این اشکال برابر ۲۵ شده است. بر اساس نتایج اشکال ۳ و ۴ در توابع تک‌نمایی F1 تا F4 الگوریتم بهینه‌سازی پیشنهادی دارای بهترین همگرایی است. در توابع تک‌نمایی F5 و F7 الگوریتم‌های IGWO و GWO بهترین نتایج را به دست آورده‌اند. بر اساس نتایج آماری جدول ۴ و میانگین ۱۰ تکرار، الگوریتم IGWO دارای مناسب‌ترین نتایج است. در الگوریتم پیشنهادی با توجه به حذف ذرات امگا از جمعیت که برازندگی کمتری دارند، مرحله بهره‌برداری نسبت به الگوریتم پایه بهبود چشمگیری داشته است.



شکل ۲- ترسیم توابع محک تک‌نمایی سه‌بعدی



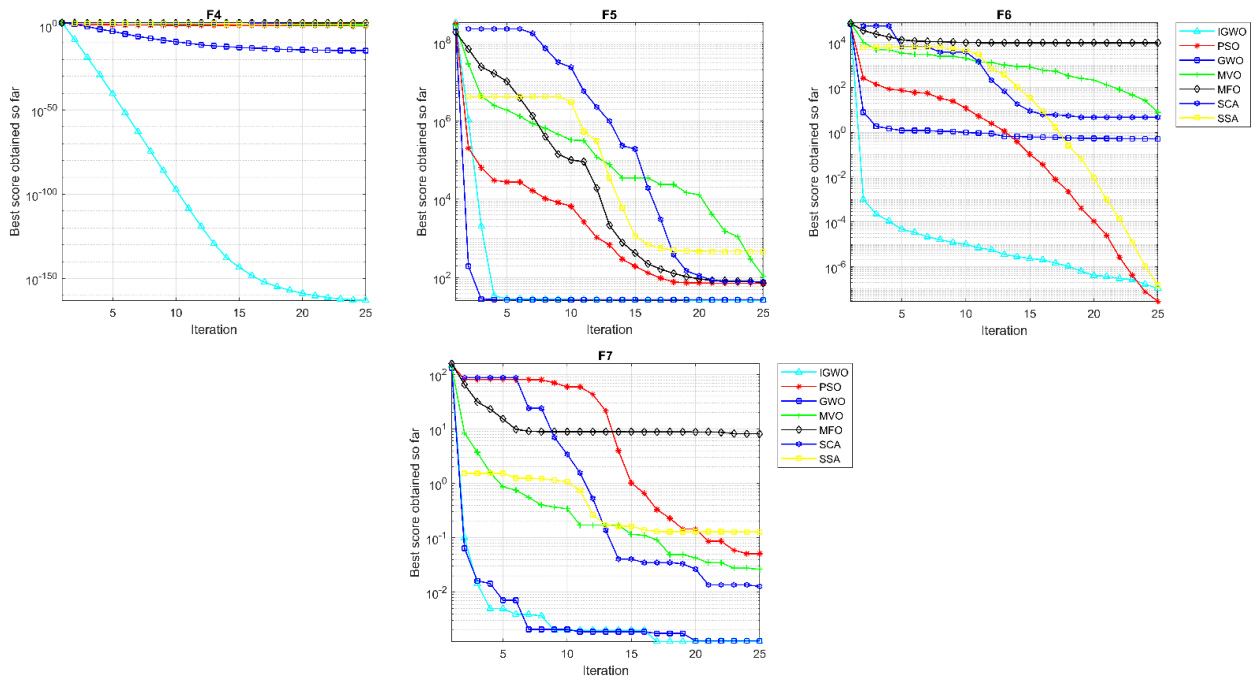
شکل ۳- مقایسه الگوریتم‌های بهینه‌سازی بر اساس توابع محک تک‌نمایی F1-F3

بدون محدودیت به صورت تک‌نمایی، چند‌نمایی و چند‌نمایی با ابعاد محدود با سطوح سختی متفاوت انتخاب شده‌اند.

بعد از ۱۰ بار اجرای الگوریتم پیشنهادی بر روی توابع مربوطه، میانگین تکرارها را به دست آورده و در جداول ۴، ۶ و ۸ نشان داده‌ایم. در هر بار اجرای کامل الگوریتم، ۱۰۰۰ بار عمل جستجو انجام می‌گیرد. اندازه جمعیت برابر ۳۰ و متناسب با آن هر پاسخ را به صورت آرایه‌ای ۳۰ تایی در نظر گرفته‌ایم. تمامی نتایج نمایش داده شده در این مقاله بر اساس قالب مورد تأیید IEEE CEC 2005 است. در جداول ۴، ۶ و ۸ نتایج بهتر با قلم ضخیم متمایز شده‌اند. این نتایج، مقایسه الگوریتم گرگ‌های خاکستری، الگوریتم ازدحام ذرات (PSO)، الگوریتم بهینه‌ساز چندوجهی (MVO) [۹]، الگوریتم بهینه‌سازی پروانه (MFO) [۱۰]، الگوریتم بهینه‌سازی سینوس کسینوس (SCA) و الگوریتم عنکبوت اجتماعی (SSA) [۱۴] است. برای شبیه‌سازی از کامپیوتری با پردازنده core i7 با توان پردازشی 2GHZ و 4GB حافظه اصلی استفاده شده است.

۴-۱- توابع محک تک‌نمایی

شکل ۲ ترسیم سه‌بعدی از این توابع محک است. همچنین توابع هزینه به همراه ابعاد، محدوده و کمترین مقدار ورودی مربوط به توابع محک تک‌نمایی در جدول ۳ نمایش داده شده است. توابع تک‌نمایی برای سنجش فرآیند بهره‌برداری مناسب هستند. جدول ۴ نتایج آماری (میانگین، میانه، انحراف معیار، کمینه و بیشینه)



شکل ۴ - مقایسه الگوریتم‌های بهینه‌سازی بر اساس توابع محک تک نمایی F4-F7

جدول ۴ - نتایج آماری توابع محک تک نمایی در ۱۰ تکرار

F1	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	1.1432e-272	1.2776e-09	3.0199e-58	0.29068	0.00059203	0.0070276	1.0044e-08
Median	3.5885e-280	4.4099e-10	1.2745e-59	0.26432	0.00010483	0.004511	9.7438e-09
STD. Dev	0	1.9562e-09	4.5196e-58	0.11126	0.00072945	0.0058338	2.3181e-09
Worst	5.716e-272	4.7281e-09	1.0303e-57	0.47761	0.001675	0.0058338	1.3886e-08
Best	4.7634e-290	1.3684e-10	3.4539e-60	0.18211	5.2987e-05	0.001594	7.6516e-09
F2	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	1.7368e-180	0.0035422	1.6342e-34	0.31232	46	7.8169e-06	0.99632
Median	8.1618e-181	9.3924e-05	1.1717e-34	0.31066	50	1.7386e-06	0.02845
STD. Dev	0	0.0076428	1.5931e-34	0.054097	5.4772	1.2459e-05	1.3695
Worst	4.5367e-180	0.017212	4.0571e-34	0.36748	50	2.9651e-05	2.8557
Best	1.0834e-183	1.2351e-05	2.6962e-35	0.22655	50	3.2317e-07	0.0055644
F3	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	1.4836e-279	14.1758	1.3129e-15	43.4899	16642.304	2216.8084	226.4108
Median	1.3522e-289	9.9435	1.1815e-16	42.9632	11223.023	2218.6933	215.4758
STD. Dev	0	9.4485	2.1058e-15	12.5752	11280.968	1493.4594	39.2889
Worst	7.2063e-279	27.8014	4.8938e-15	59.7583	33436.956	4533.294	283.4888
Best	9.126e-306	5.8634	5.5424e-19	27.4486	5570.4247	427.596	177.3234
F4	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	1.4617e-164	0.59408	1.0485e-14	1.0091	63.0638	20.6086	8.9985
Median	4.9454e-167	0.65521	9.1786e-15	0.94836	63.4631	21.4773	7.2267
STD. Dev	0	0.26777	7.1873e-15	0.36441	12.7257	11.5037	3.8875
Worst	7.2956e-164	0.90616	1.8649e-14	1.6192	79.5883	38.0445	14.6051
Best	6.9376e-170	0.21785	1.7866e-15	0.67291	43.8996	10.128	5.2437
F5	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	25.694	39.8228	27.0937	587.6235	105.6124	60.2053	264.5477
Median	25.7817	26.6348	27.1297	144.9325	114.2848	37.1477	311.1903
STD. Dev	0.14756	21.5368	0.60049	1084.6315	67.5085	42.7658	226.9828
Worst	25.8293	71.065	27.8924	2523.2951	200.7957	127.1971	497.5511
Best	25.4919	23.4629	26.1997	30.3969	29.9192	28.9504	27.6896
F6	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	9.0042e-10	3.0353e-08	0.49916	0.28208	1980.0502	4.8957	1.3281e-08
Median	7.082e-11	3.1937e-08	0.50208	0.32626	0.00023335	4.6881	1.2306e-08
STD. Dev	1.3848e-09	3.9947e-09	0.30091	0.092551	4427.5264	0.36596	3.7733e-09
Worst	3.227e-09	3.3495e-08	0.98735	0.37668	9900.2501	5.4282	1.9199e-08
Best	1.9634e-11	2.3921e-08	0.24968	0.18039	0.00012809	4.5592	9.1662e-09
F7	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	0.00082363	0.06209	0.0009236	0.019911	2.7782	0.026065	0.10834
Median	0.00069562	0.064377	0.0009241	0.017606	0.1653	0.011441	0.1118
STD. Dev	0.00027092	0.028916	0.0006354	0.0089398	3.7878	0.033008	0.044389
Worst	0.0012787	0.092623	0.002204	0.030793	8.1534	0.084938	0.16997
Best	0.00059516	0.019071	0.00051079	0.0079608	0.05778	0.0086455	0.063137

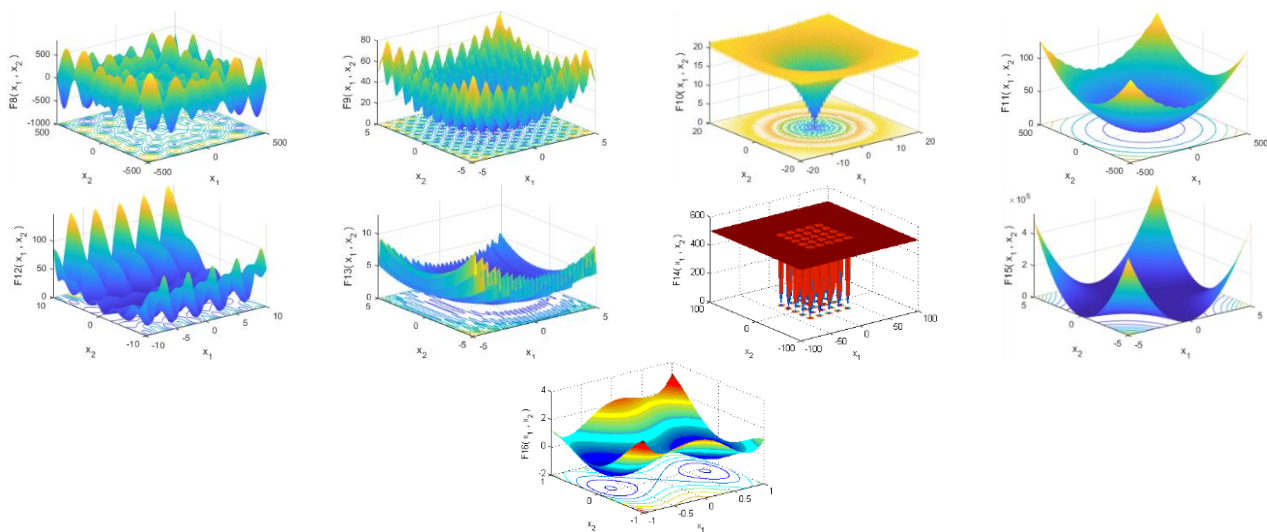
الگوریتم‌های IGWO، SSA و GWO در تابع F14 نتایج بهتری نسبت به سایر الگوریتم‌ها دارند. با در نظر گرفتن نتایج آماری جداول ۶ و ۷ الگوریتم IGWO، در توابع F8 تا F13 نتایج بهتری نسبت به سایر الگوریتم‌ها دارد. در تابع F14 الگوریتم‌های IGWO، MVO و SSA و در تابع F15 الگوریتم‌های IGWO، PSO و MFO نتایج مناسب‌تری نسبت به بقیه دارند. اکثر نتایج آماری جداول ۶ و ۷ برای تابع F16 مشابه‌اند. فقط نتایج IGWO متفاوت و نامناسب‌تر از بقیه است. با توجه به شکل ۶ و جداول ۶ و ۷ مرحله اکتشاف الگوریتم پیشنهادی بر روی اکثر توابع بهتر از سایر الگوریتم‌ها است. در الگوریتم پیشنهادی مقایسه هر پاسخ به‌دست‌آمده با بهترین پاسخ موجود، باعث بهبود مرحله اکتشاف و گریز از بهینه محلی شده است.

۴-۲- توابع محک چند نمایی

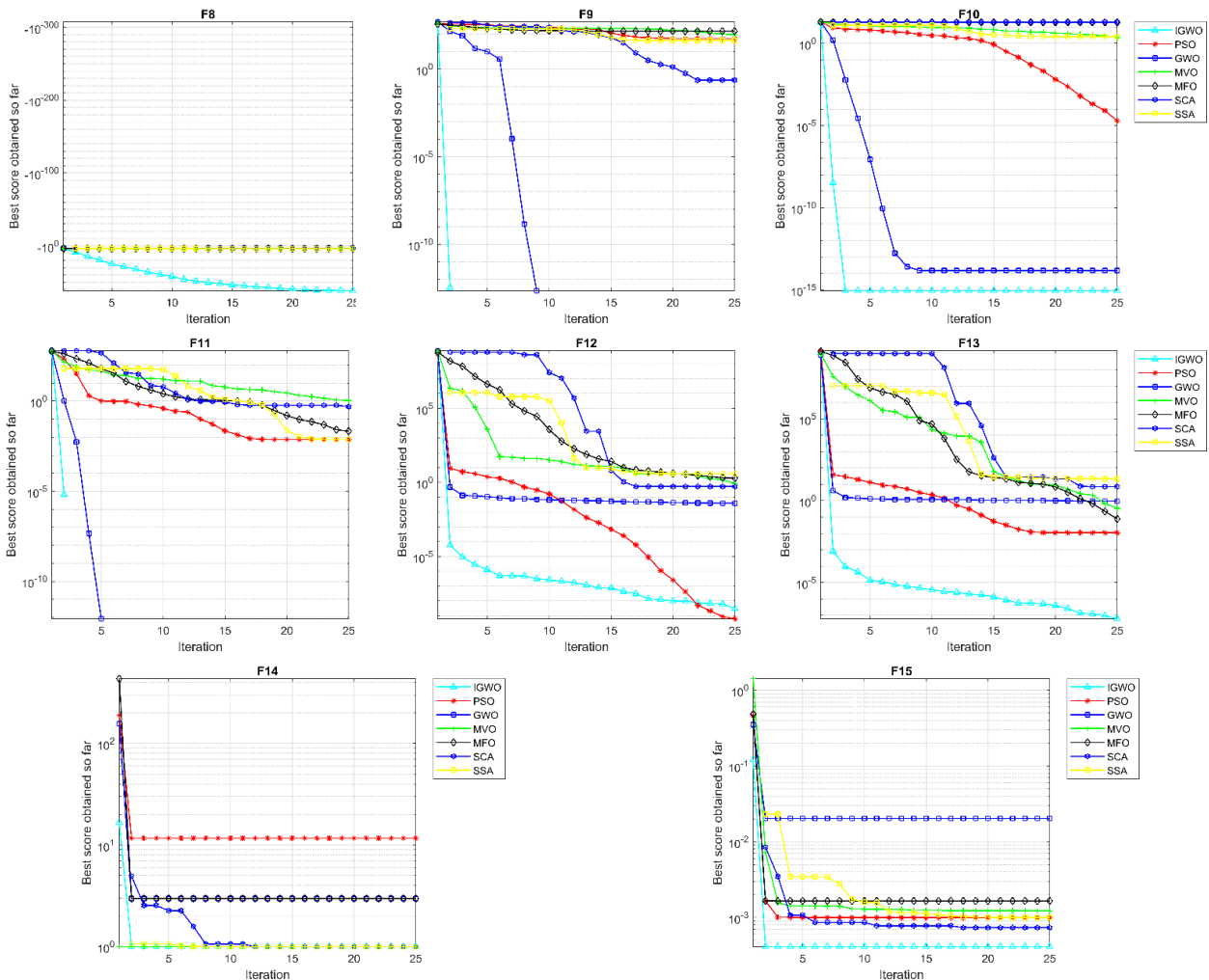
توابع هزینه مربوط به توابع محک چند نمایی به همراه ابعاد، محدوده و کمترین مقدار ورودی در جدول ۵ نمایش داده شده است. شکل ۵ ترسیم سه‌بعدی توابع محک چند نمایی است. این مسائل مرحله اکتشاف و قابلیت اجتناب از بهینه محلی الگوریتم جستجو را ارزیابی می‌نمایند. در مسائل آزمون CEC 2005 توابع F8 الی F16 چند نمایی هستند. برای بردار مکانی به طول ۳۰ نتایج میانگین الگوریتم پیشنهادی کمتر از الگوریتم پایه است. این نتایج در جداول ۶ و ۷ قابل مشاهده هستند. در توابع F8 تا F13 و تابع F15 با توجه به شکل ۶ نتایج الگوریتم پیشنهادی بهتر از بقیه الگوریتم‌ها است. همچنین

جدول ۵- توابع محک چند نمایی

توابع هزینه	ابعاد	محدوده	f_{min}
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500,500]	-418.9829×5
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	[-32,32]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i-1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100m4), y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[-50,50]	0
$f_{14}(x) = \sum_{i=1}^n \sin(x_i) \cdot \left(\sin\left(\frac{ix_i}{\pi}\right) \right)^{2m}, m = 10$	30	[0,π]	-4.687
$f_{15}(x) = \left[e^{-\sum_{i=1}^n (x_i/\beta)^{2m}} - 2e^{-\sum_{i=1}^n x_i^2} \right] \cdot \prod_{i=1}^n \cos^2 x_i, m = 5$	30	[-20,20]	-1
$f_{16}(x) = \left\{ \left[\sum_{i=1}^n \sin^2(x_i) \right] - \exp\left(-\sum_{i=1}^n x_i^2\right) \cdot \exp\left[-\sum_{i=1}^n \sin^2 \sqrt{ x_i }\right] \right\}$	30	[-10,10]	-1



شکل ۵- ترسیم توابع محک چند نمایی سه‌بعدی



شکل ۶ - مقایسه چهار الگوریتم بهینه‌سازی بر اساس توابع محک چند نمایی F8-F15

جدول ۶ - نتایج آماری توابع محک چند نمایی F8-F12 در ۱۰ تکرار

F8	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	-1.59112e+69	-6301.6911	-6270.5863	-8209.9009	-8878.2546	-3984.3398	-7527.3238
Median	-2.31021e+68	-6864.2112	-5946.916	-8358.3583	-8938.9018	-3900.5687	-7649.6914
STD. Dev	2.977547e+69	1607.7409	544.2369	643.6923	1060.3477	348.2189	421.5546
Worst	-4.02214e+61	-3656.4306	-5818.4825	-7485.2748	-7590.8205	-3688.7023	-6803.0746
Best	-6.88180e+69	-7871.1428	-7087.411	-8896.503	-10390.513	-4580.7495	-7872.7972
F9	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	4.89788e-13	50.3476	1.9867	95.8549	136.7637	12.8653	72.6318
Median	0	44.7864	0	88.7405	144.3396	0.23448	74.6218
STD. Dev	9.53878e-13	10.1594	4.4424	26.1175	39.7835	28.3851	23.9514
Worst	2.38767e-12	66.6622	9.9335	141.392	174.1879	63.6411	102.4804
Best	0	41.7885	0	75.7127	72.6317	0.00010	42.7832
F10	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	8.8818e-16	0.26809	1.5099e-14	1.5255	19.1406	12.0029	2.2529
Median	8.8818e-16	8.4004e-06	1.5099e-14	1.2452	19.0157	19.6192	2.3168
STD. Dev	0	0.59945	0	1.0424	0.43855	10.9582	0.35784
Worst	8.8818e-16	1.3404	1.5099e-14	2.9335	19.8625	20.2027	2.6605
Best	8.8818e-16	2.7453e-06	1.5099e-14	0.16281	18.7252	0.0011523	1.778
F11	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	0	0.012814	0.005156	0.6116	0.019953	0.3356	0.0073711
Median	0	0.012316	0	0.63397	0.010112	0.47904	0.0074e-07
STD. Dev	0	0.004738	0.011529	0.087518	0.022883	0.31342	0.012756
Worst	0	0.019719	0.02578	0.70357	0.059186	0.68266	0.029459
Best	0	0.007396	0	0.47179	0.002506	0.00012	5.9051e-08
F12	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	9.0208e-12	1.2482e-10	0.040851	0.83573	0.51716	0.76745	4.429
Median	1.0028e-12	6.4656e-12	0.039813	0.64241	0.11078	0.73711	4.449
STD. Dev	2.2072e-10	2.5041e-10	0.009616	0.55358	0.82855	0.23089	1.974
Worst	1.0856e-09	5.7186e-10	0.052982	1.7978	1.9479	1.0648	6.403
Best	0.4127e-12	1.756e-12	0.026605	0.40139	0.00025	0.53821	1.397

جدول ۷ - نتایج آماری توابع محک چند نمایی F13-F16 در ۱۰ تکرار

F13	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	1.5821e-08	0.0043949	0.67827	0.074108	0.33401	64.1583	4.5004
Median	1.485e-08	8.1023e-09	0.67999	0.072247	0.027678	7.1313	0.010987
STD. Dev	1.7786e-09	0.006018	0.1975	0.01091	0.71289	132.0282	10.0392
Worst	1.8283e-08	0.010987	0.92112	0.088239	1.609	300.3083	22.4591
Best	1.413e-08	4.426e-12	0.43722	0.063893	6.4104e-05	2.6367	1.1113e-09
F14	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	0.998	4.1283	6.0945	0.998	1.7916	3.3479	0.998
Median	0.998	0.998	2.9821	0.998	0.998	0.99801	0.998
STD. Dev	5.1324e-13	4.7501	4.2619	5.1305e-12	1.0867	4.2334	1.2413e-16
Worst	0.998	11.7187	10.7632	0.998	2.9821	10.7632	0.998
Best	0.998	0.998	2.9821	0.998	0.998	0.998	0.998
F15	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	0.00042689	0.00091427	0.0083302	0.0047104	0.0009383	0.0011176	0.0047331
Median	0.00041191	0.00099543	0.000309	0.00074545	0.00078266	0.0012872	0.00078331
STD. Dev	7.9762e-05	0.00016553	0.010985	0.0087539	0.00040158	0.00036756	0.0087382
Worst	0.00056196	0.0010896	0.020363	0.020363	0.0016554	0.0014578	0.020363
Best	0.00035903	0.00068315	0.0003075	0.00057748	0.00072883	0.00070591	0.00073924
F16	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
Average	-0.91872	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
Median	-0.95369	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
STD. Dev	0.07748	0	9.304e-09	1.2392e-07	0	2.493e-05	7.8505e-16
Worst	-0.8068	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
Best	-0.98672	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316

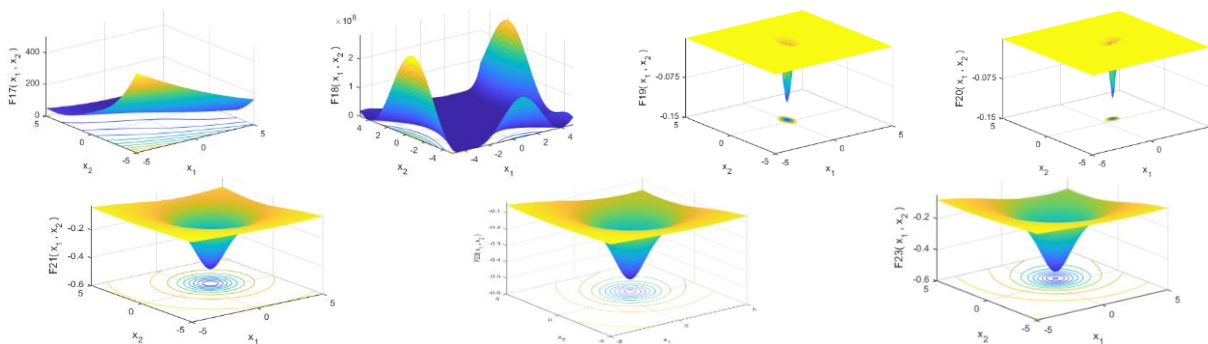
الگوریتم‌ها از نتایج آماری جدول ۹ استفاده می‌کنیم. در توابع F17 تا F20 نتایج الگوریتم‌ها مشابه یکدیگرند. در توابع F21 و F23 نتایج الگوریتم پیشنهادی بهتر است. بر روی تابع F22 نیز الگوریتم‌های IGWO و GWO دارای مناسب‌ترین نتایج هستند. در کل نتایج مراحل اکتشاف، بهره‌برداری و اجتناب از بهینه محلی الگوریتم پیشنهادی بهتر و یا برابر با سایر الگوریتم‌ها است. تغییرات اعمال شده بر روی الگوریتم پیشنهادی سبب بهبود نسبی تعادل میان مراحل اکتشاف و بهره‌برداری شده است.

۴-۳- توابع محک چند نمایی محدود

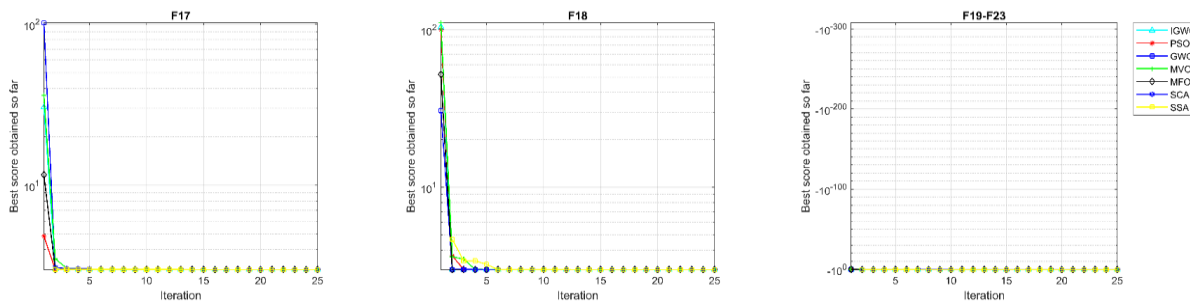
شکل ۷ ترسیم سه‌بعدی از این توابع محک است. توابع هزینه مربوط به توابع محک چند نمایی محدود به همراه ابعاد، محدوده و کمترین مقدار ورودی در جدول ۸ نمایش داده شده است. مسائل آزمون چند نمایی محدود نیز قابلیت اجتناب از گیر افتادگی در بهینه محلی و نمایانگر تعادل میان مراحل اکتشاف و بهره‌برداری است. توابع چند نمایی محدود شامل F17-F23 است. ابعاد این مسائل متفاوت است و در جدول ۸ نمایش داده شده است. شکل ۸ نتایج نزدیک الگوریتم‌ها را بر روی این توابع نشان می‌دهد. از این رو برای مقایسه دقیق‌تر

جدول ۸- توابع محک چند نمایی با ابعاد محدود

توابع هزینه	ابعاد	محدوده	f_{min}
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{\sum_{i=1}^2 (x_i - a_{ij})^6}\right)^{-1}$	2	[-65,65]	1
$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10 \left(1 - \frac{1}{8\pi}\right) \cos x_1 + 1$	2	[-5,5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
$f_{19}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	-3.86
$f_{20}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	-3.32
$f_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + C_i]^{-1}$	4	[0,10]	-10.1532
$f_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + C_i]^{-1}$	4	[0,10]	-10.4028
$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + C_i]^{-1}$	4	[0,10]	-10.5363



شکل ۷- ترسیم توابع محک چند نمایی سه‌بعدی با ابعاد محدود



شکل ۸ - مقایسه الگوریتم‌های بهینه‌سازی بر اساس توابع محک چند نمایی با ابعاد محدود

جدول ۹ - نتایج آماری توابع محک چند نمایی با ابعاد ثابت در ۱۰ تکرار

	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
F17							
Average	0.39780	0.36743	0.39789	0.39789	0.39789	0.39789	0.39789
Median	0.39789	0.36789	0.3979	0.39789	0.3977	0.39789	0.39789
STD. Dev	3.0125e-06	3.1656e-04	4.8977e-05	4.1213e-07	3.9785e-05	3.5675e-5	2.294e-14
Worst	0.3968	0.3985	0.3979	0.3978	0.3978	0.3767	0.3987
Best	0.39768	0.3887	0.3469	0.39789	0.3893	0.3245	0.39789
F18							
Average	3	3	3	3	3	3	3
Median	3	3	3	3	3	3	3
STD. Dev	4.4298e-06	7.6919e-16	8.169e-06	1.5807e-07	1.8175e-15	1.3919e-05	9.7625e-14
Worst	3	3	3	3	3	3	3
Best	3	3	3	3	3	3	3
F19							
Average	-3.8628	-3.8628	-3.8623	-3.8628	-3.8628	-3.8558	-3.8628
Median	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8548	-3.8628
STD. Dev	2.275e-06	0	0.00096394	1.8611e-07	0	0.0025014	5.8504e-14
Worst	-3.8628	-3.8628	-3.8606	-3.8628	-3.8628	-3.8545	-3.8628
Best	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8603	-3.8628
F20							
Average	-3.2982	-3.2982	-3.2147	-3.2499	-3.2507	-2.7889	-3.2227
Median	-3.322	-3.322	-3.202	-3.2027	-3.2031	-2.9918	-3.199
STD. Dev	0.053182	0.053171	0.06502	0.065855	0.06512	0.35696	0.055501
Worst	-3.2031	-3.2031	-3.1444	-3.2012	-3.2031	-2.246	-3.1959
Best	-3.322	-3.322	-3.322	-3.322	-3.322	-3.0955	-3.322
F21							
Average	-10.1528	-8.6591	-8.1136	-7.1034	-6.135	-1.5771	-8.6591
Median	-10.1528	-10.1532	-10.1523	-5.1008	-5.0552	-0.88091	-10.1532
STD. Dev	4.4087e-05	3.3408	2.7919	2.784	3.7966	1.173	3.3408
Worst	-10.1528	-2.6829	-5.0552	-5.0552	-2.6305	-0.49651	-2.6829
Best	-10.1529	-10.1532	-10.153	-10.1531	-10.1532	-3.2046	-10.1532
F22							
Average	-10.4026	-9.3481	-10.4025	-9.348	-6.6767	-3.938	-8.8727
Median	-10.4025	-10.4029	-10.4025	-10.4029	-5.1288	-4.8145	-10.4029
STD. Dev	0.00018507	2.3587	0.00027069	2.3586	3.4496	1.7875	3.4216
Worst	-10.4024	-5.1288	-10.4021	-5.1288	-3.7243	-0.90866	-2.7519
Best	-10.4029	-10.4029	-10.4028	-10.4029	-10.4029	-5.3558	-10.4029
F23							
Average	-10.5359	-9.0034	-8.9131	-9.4642	-7.856	-4.5889	-9.0034
Median	-10.5359	-10.5364	-10.5358	-10.5363	-10.5364	-5.0535	-10.5364
STD. Dev	0.00026222	3.428	3.6288	2.3974	3.6703	2.156	3.428
Worst	-10.5355	-2.8711	-2.4217	-5.1756	-3.8354	-0.94338	-2.8711
Best	-10.5362	-10.5364	-10.5363	-10.5364	-10.5364	-6.7149	-10.8711

۴-۴- مقایسه زمان اجرا

کد و افزایش پیچیدگی مراحل مختلف آن‌ها است که از قبل قابل پیش‌بینی بود. در بین تمامی الگوریتم‌ها کمترین زمان اجرا در تمامی توابع محک مربوط به PSO است. در این آزمایش پیچیدگی زمانی الگوریتم‌ها محاسبه نشده است، به این دلیل که کاهش پیچیدگی، جزء اهداف بهینه‌سازی انجام گرفته نیست [۴، ۱۰، ۱۳-۱۵، ۱۷، ۲۰، ۲۸-۳۷].

در این بخش مقایسه دیگری از لحاظ زمان اجرا انجام گرفته است. میانگین زمان اجرای الگوریتم‌ها در ۱۰ تکرار در جدول ۱۰ نمایش داده شده است. نتایج نشان‌دهنده افزایش جزئی زمان اجرا برای الگوریتم پیشنهادی است. دلیل افزایش زمان اجرا، طولانی‌تر شدن

جدول ۱۰ - مقایسه آماری میانگین زمان اجرای الگوریتم‌ها

	IGWO	PSO	GWO	MVO	MFO	SCA	SSA
F1	1.6506	0.1541	0.3519	0.5750	0.3110	0.3010	0.4197
F2	1.7147	0.1976	0.3869	0.5737	0.3393	0.3330	0.3860
F3	2.9693	1.1127	1.3121	1.4822	1.3491	1.2714	1.3685
F4	2.5242	0.1588	0.3512	0.5924	0.3149	0.3194	0.3757
F5	2.7784	0.2327	0.4126	0.6814	0.3797	0.3841	0.4378
F6	2.7423	0.1569	0.3709	0.5790	0.3095	0.2997	0.3569
F7	3.8421	0.3455	0.5802	0.7572	0.5144	0.5047	0.5608
F8	3.3348	0.2375	0.5072	0.4323	0.3801	0.3832	0.4984
F9	1.8530	0.1864	0.4060	0.6080	0.3372	0.3258	0.3952
F10	2.1258	0.2213	0.4226	0.6542	0.3772	0.3834	0.4309
F11	3.5213	0.3048	0.4519	0.7595	0.4560	0.4218	0.4749
F12	3.2034	0.8461	1.2149	1.2479	1.0077	0.9967	1.0701
F13	2.2230	0.9181	1.0692	1.3723	1.1495	1.1776	1.1708
F14	2.1259	2.0351	2.3973	2.2328	2.0853	2.0715	2.2386
F15	2.0554	0.1622	0.3761	0.3313	0.2401	0.2163	0.3614
F16	0.2730	0.1770	0.2355	0.2496	0.2535	0.1724	0.2886
F17	0.3143	0.0909	0.1839	0.3908	0.1896	0.1704	0.2882
F18	0.3592	0.0585	0.2085	0.3051	0.1581	0.1410	0.2630
F19	0.4643	0.1872	0.5344	0.3809	0.2937	0.2898	0.4105
F20	0.6813	0.1957	0.3366	0.3712	0.3303	0.2949	0.4077
F21	0.6704	0.4206	0.7996	0.5820	0.5223	0.4970	0.6303
F22	0.8783	0.6491	0.8140	1.3287	0.7469	0.6973	0.8608
F23	0.9557	0.7216	0.8556	0.8982	0.8413	0.8511	0.9627

به اجرای وظیفه باشد. اولین زمان پایان (EFT) هر گره روی پردازنده P_k از رابطه ۹ قابل محاسبه است [۳۹].

$$EFT(n_i, P_k) = AST(n_i, P_k) + W(n_i, P_k) \quad (9)$$

در رابطه ۹ متغیر W زمان موردنیاز برای پردازش وظیفه n_i روی پردازنده P_k است. زمان اتمام تمامی وظایف (makespan) برابر با زمان پایان گره خروجی گراف زمان است [۳۹]. در این مثال کاربردی هدف کاهش زمان اتمام تمامی وظایف است [۴۰-۴۵].

$$makespan = \max\{EFT(n_{exit})\} \quad (10)$$

مطابق برخی از مقالات زمان‌بندی دو نوع جریان کاری متعادل و نامتعادل داریم [۴۶، ۴۷]. Epigenomics و Montage نمونه‌هایی از این نوع جریان کاری هستند. شکل ۹ نمودار مربوط به این دو نوع جریان کاری است [۴۸]. جریان کاری Epigenomics از نوع متعادل و Montage از نوع نامتعادل است. جریان کاری متعادل دارای تعدادی خط لوله است که به سرویس‌های مشابه نیاز دارند. اما انواع مختلف خدمات را پردازش می‌کنند. جریان کاری نامتعادل پیچیده‌تر بوده و تعدادی وظایف موازی داشته که به سرویس‌های مختلفی نیاز دارند. در آزمایش انجام شده ۲۰ وظیفه بر روی ۵ ماشین با توان پردازشی یکسان اجرا شده‌اند.

۴-۵- مطالعه کاربردی: زمان‌بندی جریان کار

برای ارزیابی الگوریتم پیشنهادی مسئله جریان کاری را در محیط متلب شبیه‌سازی نموده و با الگوریتم‌های GWO و PSO مقایسه نمودیم. جریان‌های کاری مختلف تأثیرهای متفاوتی روی کارایی الگوریتم‌های زمان‌بندی دارند.

جریان کار موازی را می‌توانیم توسط یک گراف غیر حلقوی به صورت شکل ۹ نشان دهیم. گراف وظیفه $G = (N, E)$ شامل مجموعه‌ای از رئوس N و یال‌های E است. در این مسئله N مجموعه‌ای از وظایف و E مجموعه‌ای از لبه‌های موجود مابین وظایفی است که بیانگر محدودیت‌های دارای اولویت هستند. هر یال $E \in edge(i, j)$ مابین وظایف n_i و n_j نشان می‌دهد که وظیفه n_j نمی‌تواند شروع به کار نماید تا زمانی که اجرای وظیفه n_i تمام نشده باشد. وظایف بدون یال ورودی نیز وظایف شروع می‌باشند. زمان شروع واقعی (AST) برای هر گره n_i روی پردازنده P_k با رابطه ۸ محاسبه می‌شود [۳۸].

$$AST(n_i, P_k) = \max(EST(n_i, P_k), Avail(P_k)) \quad (8)$$

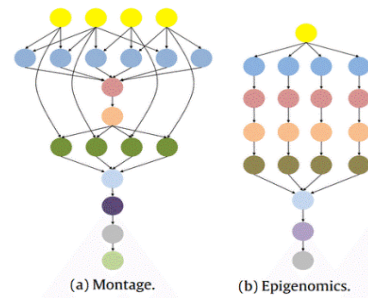
در این رابطه $EST(n_i, P_k)$ زمان شروع اجرای وظیفه n_i روی پردازنده P_k است. $Avail(P_k)$ اولین زمانی است که پردازنده P_k آماده

۵- نتیجه‌گیری و کارهای آتی

هدف این مقاله ارائه نسخه جدیدی از الگوریتم فرا ابتکاری گرگ‌های خاکستری به منظور بهینه نمودن قابلیت جستجو در شکار گرگ‌ها است. برای بررسی میزان کارایی الگوریتم پیشنهادی از ۲۳ تابع استاندارد محک IEEE CEC2005 استفاده کرده‌ایم. میزان همگرایی الگوریتم پیشنهادی در مقایسه با الگوریتم پایه و الگوریتم مورد مقایسه دیگر، بر روی برخی توابع محک بهتر است. الگوریتم پیشنهادی را به صورت محیط گسسته در مسئله زمان‌بندی جریان کاری پیاده‌سازی نموده و معیار makespan را مورد بررسی قرار داده‌ایم. نتایج نشان‌دهنده نتایج خوب برای الگوریتم پیشنهادی است. الگوریتم پیشنهادی مسیر جدیدی در راستای بهبود توانایی جستجو مبتنی بر رهبر باز کرده است. در آینده بهبودهای دیگری برای این الگوریتم در مسائل محدود و نامحدود به حالت‌های چند ازدحامی و چندهدفه مختلف می‌توان ارائه نمود.

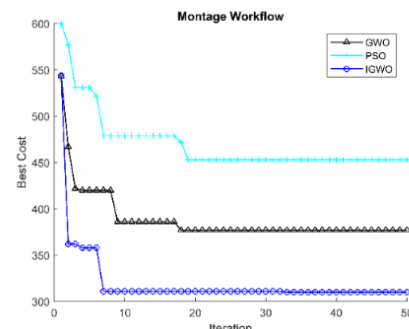
مراجع

- [1] B. Farnad, A. Jafarian, and D. Baleanu, "A new hybrid algorithm for continuous optimization problem," *Applied Mathematical Modelling*, vol. 55, pp. 652-673, 2018.
- [2] A. Aktel, B. Yagmahan, T. Ozcan, M. M. Yenisey, and E. Sansarçı, "The comparison of the metaheuristic algorithms performances on airport gate assignment problem," *Transportation Research Procedia*, vol. 22, pp. 469-478, 2017.
- [3] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.
- [4] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120-133, 2016.
- [5] A. Mozaffari, M. Emami, and A. Fathi, "A comprehensive investigation into the performance, robustness, scalability and convergence of chaos-enhanced evolutionary algorithms with boundary constraints," *Artificial Intelligence Review*, pp. 1-62, 2018.
- [6] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*: Springer, 2011, pp. 760-766.
- [7] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: approach to solve structural optimization problems," *Engineering with computers*, vol. 29, no. 1, pp. 17-35, 2013.
- [8] G. G. Tejani, V. J. Savsani, and V. K. Patel, "Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization," *Journal of Computational Design and Engineering*, vol. 3, no. 3, pp. 226-249, 2016.
- [9] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495-513, 2016.
- [10] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228-249, 2015.
- [11] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider monkey optimization algorithm for numerical optimization," *Memetic computing*, vol. 6, no. 1, pp. 31-47, 2014.
- [12] M. Dorigo and T. Stutzle, "The ant colony optimization metaheuristic: Algorithms, applications, and advances," in *Handbook of metaheuristics*: Springer, 2003, pp. 250-285.
- [13] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, and I. Aljarah, "Grasshopper optimization algorithm for multi-objective

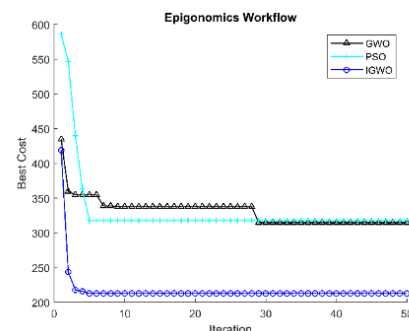


شکل ۹ - گراف بدون دور جریان‌های کاری علمی

توزیع وظایف به صورت محدود در بین ماشین‌ها می‌تواند نیازمندی‌های کاربرانی که از محیط ابری استفاده می‌کنند را برطرف نماید. با توجه به این مسئله که اکثر زمان‌بندی‌های ارائه‌شده بهترین پاسخ‌های ممکن را در هر شرایطی به دست نمی‌آورند، در این مقاله با الگوریتم بهبودیافته سعی شده تا بتوانیم مسئله زمان‌بندی را تا حدی بهبود بخشیم. الگوریتم پیشنهادی با توجه به نتایج نمایش داده‌شده دارای کمترین زمان اتمام وظایف در مقایسه با الگوریتم‌های پایه است. همان‌طور که در شکل ۱۰ قابل مشاهده است الگوریتم بهینه‌سازی پیشنهادی، بعد از چند تکرار توانسته نسبت به دیگر روش‌ها به زمان اتمام وظایف کمتری دست پیدا کند. شکل ۱۱ نتایج مربوط به اجرای الگوریتم‌ها بر روی جریان کاری Epigenomics است. نتیجه الگوریتم پیشنهادی بهتر از هر دو الگوریتم دیگر است. تغییرات اعمال‌شده بر روی الگوریتم پیشنهادی در مسئله زمان‌بندی وظایف تاثیر قابل توجهی در زمان اتمام وظایف داشته است.



شکل ۱۰ - مقایسه زمان اتمام وظایف بر روی جریان کاری Montage



شکل ۱۱ - مقایسه زمان اتمام وظایف بر روی جریان کاری Epigenomics

- Expert Systems with Applications, vol. 41, no. 5, pp. 2535-2545, 2014.
- [33] G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, "Chaotic krill herd algorithm," *Information Sciences*, vol. 274, pp. 17-34, 2014.
- [34] X.-S. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464-483, 2012.
- [35] D. Yousri, D. Allam, and M. Eteiba, "Chaotic whale optimizer variants for parameters estimation of the chaotic behavior in Permanent Magnet Synchronous Motor," *Applied Soft Computing*, vol. 74, pp. 479-503, 2019.
- [36] A. Lim, J. Lin, B. Rodrigues, and F. Xiao, "Ant colony optimization with hill climbing for the bandwidth minimization problem," *Applied Soft Computing*, vol. 6, no. 2, pp. 180-188, 2006.
- [37] Y. Xu, K. Li, J. Hu, and K. Li, "A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues," *Information Sciences*, vol. 270, pp. 255-287, 2014.
- [38] U. Schwiegelshohn, *Job Scheduling Strategies for Parallel Processing*. Springer, 2010.
- [39] S. Elsherbiny, E. Eldaydamony, M. Alrahmawy, and A. E. Reyad, "An extended Intelligent Water Drops algorithm for workflow scheduling in cloud computing environment," *Egyptian informatics journal*, vol. 19, no. 1, pp. 33-55, 2018.
- [40] I. Casas, J. Taheri, R. Ranjan, L. Wang, and A. Y. Zomaya, "GA-ETI: An enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments," *Journal of computational science*, vol. 26, pp. 318-331, 2018.
- [41] F. Abazari, M. Analoui, H. Takabi, and S. Fu, "MOWS: Multi-objective workflow scheduling in cloud computing based on heuristic algorithm," *Simulation Modelling Practice and Theory*, 2018.
- [42] E. N. Alkhanak and S. P. Lee, "A hyper-heuristic cost optimisation approach for Scientific Workflow Scheduling in cloud computing," *Future Generation Computer Systems*, vol. 86, pp. 480-506, 2018.
- [43] A. Choudhary, I. Gupta, V. Singh, and P. K. Jana, "A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing," *Future Generation Computer Systems*, vol. 83, pp. 14-26, 2018.
- [44] H. Hu et al., "Multi-objective scheduling for scientific workflow in multicloud environment," *Journal of Network and Computer Applications*, vol. 114, pp. 108-122, 2018.
- [45] H. M. Fard, R. Prodan, J. J. D. Barrionuevo, and T. Fahringer, "A multi-objective approach for workflow scheduling in heterogeneous environments," in *Cluster, Cloud and Grid Computing (CCGrid)*, 2012 12th IEEE/ACM International Symposium on, 2012, pp. 300-309: IEEE.
- [46] J. Yu, M. Kirley, and R. Buyya, "Multi-objective planning for workflow execution on grids," in *Proceedings of the 8th IEEE/ACM International conference on Grid Computing*, 2007, pp. 10-17: IEEE Computer Society.
- [47] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: a comprehensive analysis" *Journal of Network and Computer Applications*, vol. 66, pp. 64-82, 2016.
- optimization problems," *Applied Intelligence*, vol. 48, no. 4, pp. 805-820, 2018.
- [14] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163-191, 2017.
- [15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46-61, 2014.
- [16] T.-S. Pan, T.-K. Dao, and S.-C. Chu, "A communication strategy for paralleling grey wolf optimizer," in *International Conference on Genetic and Evolutionary Computing*, 2015, pp. 253-262: Springer.
- [17] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *Journal of Computational Design and Engineering*, 2017.
- [18] V. K. Kamboj, "A novel hybrid PSO-GWO approach for unit commitment problem," *Neural Computing and Applications*, vol. 27, no. 6, pp. 1643-1655, 2016.
- [19] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371-381, 2016.
- [20] A. Mukherjee and V. Mukherjee, "Chaotic krill herd algorithm for optimal reactive power dispatch considering FACTS devices," *Applied Soft Computing*, vol. 44, pp. 163-190, 2016.
- [21] J.-J. Liang, P. N. Suganthan, and K. Deb, "Novel composition test functions for numerical global optimization," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, 2005, pp. 68-75: IEEE.
- [22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Let a biogeography-based optimizer train your multi-layer perceptron," *Information Sciences*, vol. 269, pp. 188-209, 2014.
- [23] S. Mirjalili, S. M. Mirjalili, and X.-S. Yang, "Binary bat algorithm," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 663-681, 2014.
- [24] W. Guo, L. Wang, and Q. Wu, "An analysis of the migration rates for biogeography-based optimization," *Information Sciences*, vol. 254, pp. 111-140, 2014.
- [25] S. M. Mirjalili and S. Mirjalili, "Oval-shaped-hole photonic crystal waveguide design by MoMIR framework," *IEEE Photonics Technology Letters*, vol. 26, no. 24, pp. 2446-2449, 2014.
- [26] S. M. Mirjalili and S. Z. Mirjalili, "Full optimizer for designing photonic crystal waveguides: IMoMIR framework," *IEEE Photonics Technology Letters*, vol. 27, no. 16, pp. 1776-1779, 2015.
- [27] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80-98, 2015.
- [28] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, no. 2, pp. 224-232, 2014.
- [29] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831-4845, 2012.
- [30] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *Journal of Computational Design and Engineering*, vol. 5, no. 4, pp. 458-472, 2018.
- [31] S. Lukasik and S. Zak, "Firefly algorithm for continuous constrained optimization tasks," in *International conference on computational collective intelligence*, 2009, pp. 97-106: Springer.
- [32] J. Luo and M.-R. Chen, "Improved shuffled frog leaping algorithm and its multi-phase model for multi-depot vehicle routing problem,"

پاورقی‌ها:

¹ Multi-Verse Optimizer
² Moth-Flame Optimizer
³ Social Spider Algorithm

⁴ Actual Start Time
⁵ Earliest Start Time
⁶ Earliest Finish Time