

پیشرفت های مهندسی نرم افزار

الگوریتم بهینه سازی نهنگ

سید علی میرجلالی، اندرو لوییس

دانشکده ی فناوری اطلاعات و ارتباطات، دانشگاه گریفیت، پردیس ناتان، بریسبن، کوئینزلند 4111، استرالیا

کالج گریفیت، Mt Gravatt، بریسبن، کوئینزلند 4122، استرالیا

فهرست مقاله

تاریخچه ی مقاله:

رسیده شده در تاریخ 7 آگوست 2015

تجدید نظر در تاریخ 8 ژانویه 2016

پذیرفته شده در تاریخ 15 ژانویه 2016

لغات کلیدی :

بهینه سازی، محک، بهینه سازی مقید، بهینه سازی ازدحام ذرات، الگوریتم، الگوریتم ابتکاری، الگوریتم ژنتیک، بهینه سازی ساختاری

چکیده:

این مقاله ، به معرفی الگوریتم بهینه سازی جدید، فوق ابتکاری و الهام گرفته از طبیعت به نام الگوریتم بهینه سازی نهنگ (WOA) می پردازد. این روش تقلیدی از رفتار نهنگ های کوهان دار بوده و از استراتژی شکار حباب-تور الهام گرفته شده است. WOA توسط 29 مسئله ی بهینه سازی ریاضی و 6 مسئله ی طراحی ساختاری، مورد آزمایش قرار گرفته است. نتایج بهینه سازی این نکته را اثبات می کنند که الگوریتم WOA می تواند با الگوریتم های فوق ابتکاری و پیشرفته و همچنین روش های مرسوم رقابت کند. سورس کد های الگوریتم WOA در آدرس <http://www.alimirjalili.com/WOA.html> در دسترس می باشند.

©2016 Elsevier Ltd. All rights reserved.

الگوریتم های بهینه سازی فوق ابتکاری به 4 دلیل در بین مهندسی شهرت یافته اند: اول اینکه بر اساس مفاهیم ساده ای بوده و به راحتی پیاده سازی می شوند، ثانیاً به اطلاعات زیادی نیاز ندارد، ثالثاً می توانند حالت مطلوب محلی را دور بزنند و رابعاً، می توانند مسائل گوناگون با سازوکارهای متفاوت را پوشش دهند.

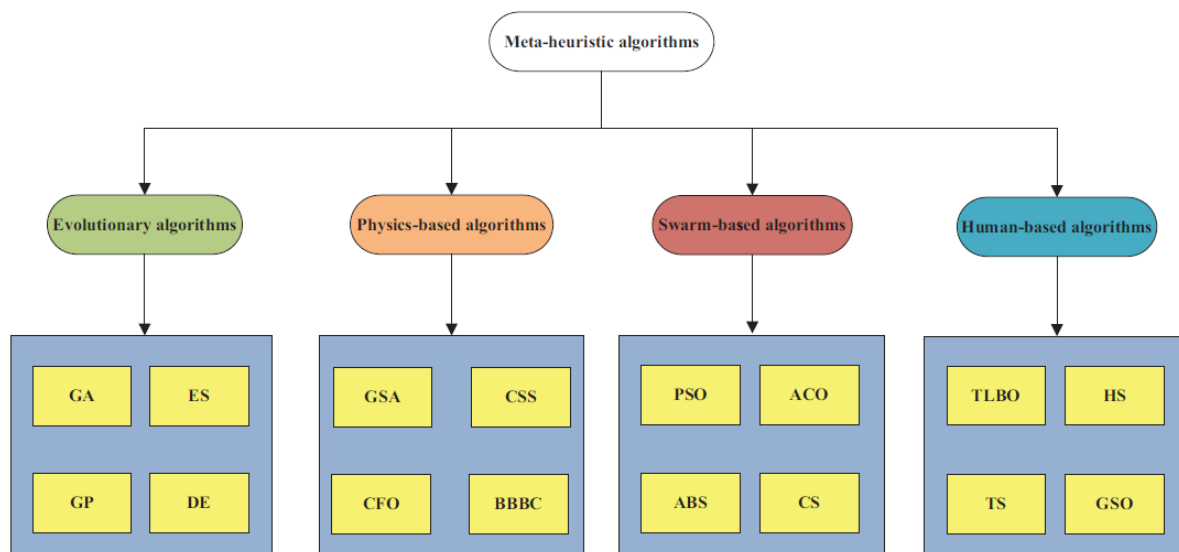
الگوریتم های فوق ابتکاری و برگرفته از طبیعت مسائل بهینه سازی را با تقلید از پدیده های فیزیکی و زیست شناسی حل می کنند. آن روش های در 3 دسته طبقه بندی می شوند (شکل 1 را ببینید): مبتنی بر تکامل، مبتنی بر فیزیک و مبتنی بر ازدحام. روش های مبتنی بر تکامل از قوانین تکامل طبیعت الهام گرفته شده اند. فرایند جستجو از یک جمعیت تولید شده به صورت

تصادفی آغاز شده که در طول نسل های متوالی نمو پیدا می کند. نقطه قوت این روش ها این است که جهت تشکیل نسل بعدی افراد، بهترین افراد دور هم جمع می شوند. این نکته به افراد کمک می کند تا در طی نسل ها، بهینه سازی شوند. محبوب ترین روش الهام گرفته شده از تکامل الگوریتم ژنتیک است که تکامل داروین را شبیه سازی می کند (1). دیگر روش های محبوب ، استراتژی تکامل (110)، یادگیری افزایشی مبتنی بر احتمالات (111)، برنامه ریزی ژنتیک (2) و بهینه ساز مبتنی بر جغرافیای زیستی می باشند.

روش های مبتنی بر فیزیک، از قوانین فیزیک حاکم در جهان تقلید می کنند. محبوب ترین این الگوریتم ها، گداختگی شبیه سازی شده (4و5) ، جستجوی محلی جاذبه ای (6)، انفجار بزرگ و بحران بزرگ (7)، الگوریتم جستجوی گرانشی، جستجوی سیستم موکل (9) بهینه سازی نیروی مرکزی (10)، الگوریتم بهینه سازی واکنشی شیمیایی مصنوعی (11)، چاله ی سیاه (12)، بهینه سازی اشعه (13)، بهینه سازی جهان کوچک (14)، الگوریتم جستجو مبتنی بر کهکشان (15) و بهینه سازی فضای خمیده (16) می باشند.

دسته ی سوم از روش های مبتنی بر طبیعت شامل روش های مبتنی بر ازدحام هستند که رفتار اجتماعی گروه هایی از حیوانات را تقلید می کنند. محبوب ترین این روش ها روش بهینه سازی ازدحام ذرات است که در ابتدا توسط کندی و ابرهارت گسترش یافت (17). PSO الهام گرفته از رفتار اجتماعی جمع شدن پرندگان است. این روش، از تعدادی از ذرات (راه حل های نامزد) که در پیرامون فضای جستجو جهت یافتن بهترین راه حل پرواز می کنند (یا همان مکان بهینه)، استفاده کرده است. ضمناً، تمام آنها در طول مسیرشان به دنبال بهترین راه حل یا همان مکان بهینه می باشند. به عبارت دیگر، ذرات بهترین راه حل های خودشان را در کنار بهترین راه حل ازدحام، در نظر می گیرند. دیگر الگوریتم مبتنی بر ازدحام، بهینه سازی کلونی مورچه است که در ابتدا توسط دوریگو و همکارانش معرفی شد (18). این الگوریتم از رفتار اجتماعی مورچه ها در کلونی ، الهام گرفته است. در حقیقت، هوش اجتماعی مورچه ها در پیدا کردن نزدیکترین مسیر از آشپانه تا منبع غذا، انگیزه ی اصلی این الگوریتم بوده است. در طی تکرار راه حل های نامزد، ماتریس فورمون شکل می گیرد.

دیگر روش های مبتنی بر ازدحام، در جدول 1 فهرست شده اند. این طبقه از روش های فوق ابتکاری به دلیل امکان رقابت با الگوریتم های مبتنی بر فیزیک و تکامل، بسیار محبوب شدند.



عموماً، الگوریتم های مبتنی بر ازدحام نسبت به الگوریتم های تکاملی، مزایایی دارند. به عنوان مثال، الگوریتم های مبتنی بر ازدحام در طی تکرار ها، از اطلاعات فضای جستجو محافظت می کنند، این در حالی است که الگوریتم های مبتنی بر تکامل به محض تشکیل جمعیت جدید، از هر اطلاعاتی صرف نظر می کنند. این روشها در مقایسه با روش های تکاملی، از عملگرهای کمتری بهره می برند (انتخاب، متقاطع، جهش،نخبه سالاری و...) و بنابراین پیاده سازی آنها ساده تر است.

باید ذکر کرد که روش های فوق ابتکاری دیگری وجود دارند که از رفتارهای انسان الهام گرفته اند. بعضی از معروفترین این روش ها عبارت است از : بهینه سازی مبتنی بر یادگیری و یاددهی (36و37)، جستجوی هارمونی (38)، جستجوی تابو (39)، بهینه ساز جستجوی گروه (42و43)، الگوریتم رقابتی امپریالیست (44)، الگوریتم لیگ قهرمانی (45و46)، الگوریتم آتش بازی (47)، بهینه سازی برخورد بدنه (48و49)، الگوریتم جستجوی داخلی (50)، الگوریتم انفجار معدن (51)، الگوریتم رقابت لیگ فوتبال (52و53)، الگوریتم بهینه سازی جستجوگر (54)، الگوریتم مبتنی بر جامعه (55)، الگوریتم بازار مبادله (56) و الگوریتم بهینه سازی گروه مشاوره (57و58).

Table 1
Swarm-based optimization algorithms developed in literature.

Algorithm	Inspiration	Year of proposal
PSO [17]	Bird flock	1995
Marriage in Honey Bees Optimization Algorithm (MBO) [19]	Honey Bees	2001
Artificial Fish-Swarm Algorithm (AFSA) [20]	Fish swarm	2003
Termite Algorithm [21]	Termite colony	2005
ACO [18]	Ant colony	2006
ABC [22]	Honey Bee	2006
Wasp Swarm Algorithm [23]	Parasitic wasp	2007
Monkey Search [24]	Monkey	2007
Wolf pack search algorithm [25]	Wolf herd	2007
Bee Collecting Pollen Algorithm (BCPA) [26]	Bees	2008
Cuckoo Search (CS) [27]	Cuckoo	2009
Dolphin Partner Optimization (DPO) [28]	Dolphin	2009
Bat-inspired Algorithm (BA) [29]	Bat herd	2010
Firefly Algorithm (FA) [30]	Firefly	2010
Hunting Search (HS) [31]	Group of animals	2010
Bird Mating Optimizer (BMO) [32]	Bird mating	2012
Krill Herd (KH) [33]	Krill herd	2012
Fruit fly Optimization Algorithm (FOA) [34]	Fruit fly	2012
Dolphin Echolocation (DE) [35]	Dolphin	2013

الگوریتم های بهینه سازی فوق ابتکاری بر مبنای جمعیت، صرف نظر از طبیعتشان خصوصیت مشترکی را دارا می باشند. فرایند جستجو به دو فاز تقسیم می شود : اکتشاف و استخراج (60،59). بهینه ساز جهت اکتشاف در سرتاسر فضای جستجو، باید شامل

عملگره‌هایی باشد : در این فاز حرکات (به معنای آشفتگی متغیر های طراحی) تا حد امکان باید تصادفی باشند. فاز استخراج به دنبال فاز اکتشاف می آید و می توان آن را به عنوان فرایند بررسی جزئی محیط (های) محتمل در فضای جستجو تعریف کرد. بنابراین استخراج به قابلیت جستجوی محلی در مناطق احتمالی فضای طراحی که در فاز اکتشاف پیدا شده اند، بستگی دارد. یافت تعادل مناسب بین اکتشاف و استخراج ، به دلیل طبیعت تصادفی فرایند بهینه سازی، چالش برانگیزترین وظیفه در مسیر توسعه ی الگوریتم فوق ابتکاری می باشد.

این مطالعه، با تقلید از رفتار شکار نهنگ های کوهان دار، الگوریتم بهینه سازی فوق ابتکاری جدیدی را به نام WOA شرح داده است. بر طبق آنچه که نویسندگان این مقاله در ذهن دارند، در زمینه ی بهینه سازی، تا کنون چنین مطالعه ای در این موضوع صورت نگرفته است. تفاوت اصلی بین اثر حاضر و مقالات منتشر شده ی اخیر توسط نویسندگان (به خصوص GWO (62)) این است که در این روش، رفتار شکار با بهترین عوامل و یا عوامل تصادفی جستجو جهت تعقیب طعمه شبیه سازی شده و همچنین با استفاده از یک مارپیچ ،شبیه سازی مکانیزم حمله ی تور-حباب نهنگ کوهان دار صورت گرفته است. کارایی الگوریتم WOA به وسیله ی حل 20 مسئله ی بهینه سازی و 6 مسئله ی بهینه سازی ساختاری ارزیابی شده است. نتایج بهینه سازی نشان داد که WOA می تواند با روش های بهینه سازی مدرن رقابت داشته باشد.

مابقی مقاله به صورت ذیل تنظیم شده است. بخش 2، الگوریتم بهینه سازی نهنگ را که در این مقاله توسعه یافته است، شرح می دهد. مسائل تست و نتایج بهینه سازی به همراه توابع ریاضی و مسائل بهینه سازی ساختاری به ترتیب در بخش 3 و 4 مطرح شده اند. در بخش 5، خلاصه ای از یافته های اصلی مقاله ارائه و مسیرهایی برای مطالعات آتی پیشنهاد شده است.



Fig. 2. Bubble-net feeding behavior of humpback whales.

2. الگوریتم بهینه سازی نهنگ (WOA)

در این بخش، در ابتدا انگیزه های به وجود آمدن این روش توضیح داده می شود. سپس مدل ریاضی ارائه می شود.

2.1 الهام

نهنگ ها موجودات جالبی هستند. آنها به عنوان عظیم ترین پستانداران در جهان شناخته می شوند. یک نهنگ بالغ می تواند تا 30 متر طول و 180 تن وزن، رشد کند. 7 گونه ی اصلی و متفاوت از این پستاندار عظیم الجثه به نام های قاتل، مینکه، سی ، گوژپشت، رایت، نهنگ سواحل اقیانوس اطلس و آبی رنگ وجود دارد. نهنگ ها اغلب به عنوان شکارچی در نظر گرفته می شوند. به دلیل تنفس آنها از سطح اقیانوس، نهنگ ها نمی خوابند. در حقیقت، تنها نیمی از مغز آنها به خواب می رود. نکته ی جالب در مورد نهنگ ها این است که آنها حیواناتی باهوش و با احساس هستند.

بر طبق اثر هاف و ون در گاش (63) ، نهنگ ها به مانند انسان ها، سلول های مشترکی به نام سلول های بنیادی در بخش های معینی از مغز خود دارند . این سلول ها مسئول قضاوت، احساسات و رفتارهای اجتماعی انسان ها می باشند. به عبارت دیگر، سلول های بنیادی، ما را از سایر موجودات متمایز می کنند. نهنگ ها دو برابر یک انسان بالغ ، دارای این سلول ها هستند و همین عامل دلیل باهوش بودن آنها می باشد. این نکته ثابت شده است که نهنگ های قادر به تفکر، یادگیری، قضاوت، برقراری ارتباط و به مانند یک انسان دچار حالت احساساتی شدن هستند ، اما به وضوح سطح هوشیاری بسیار پایین تری دارند. مشاهده شده است که نهنگ ها (اغلب نهنگ های قاتل) می توانند گویش منحصر به فرد خود را داشته باشند.

نکته ی جالب دیگر ، رفتار اجتماعی نهنگ هاست. آنها به صورت فردی و یا گروهی زندگی می کنند. هرچند که آنها اغلب در گروه ها مشاهده می شوند. بعضی از گونه های نهنگ (برای مثال نهنگ قاتل) می تواند در تمام طول زندگی خود، در یک خانواده زیست کند. یکی از بزرگ باله ترین نهنگ ها ، نهنگ گوژپشت است (*Megaptera novaeangliae*) . یک نهنگ گوژپشت بالغ تقریباً به اندازه یک اتوبوس مدرسه است. شکار مورد علاقه ی آنها ، کریل و گروه های ماهی کوچک است. در شکل 2، این پستانداران به تصویر کشیده شده اند.

جالبترین نکته در مورد نهنگ های گوژپشت روش خاص شکار آنها می باشد. این رفتار کاوشگرانه به نام روش تغذیه حباب-تور شناخته می شود (64). نهنگ های گوژپشت ترجیح می دهند تا دسته ای از کریل ها و یا ماهی های کوچک را در نزدیک سطح آب شکار کنند. این نکته مشاهده شده است که این کاوش و شکار ، با ایجاد حباب های شاخصی در امتداد یک دایره یا مسیر هایی "9 شکل" به انجام می رسد. (شکل 2) قبل از سال 2011 ، این رفتار تنها بر اساس مشاهدات سطح آب برداشت می شد. هر چند، گولدموگن و همکارانش (65) این رفتار را با استفاده از سنسورهای برچسبی مورد بررسی قرار داده اند. آنها اطلاعات 300 شکار حباب-تور را با استفاده از سنسورهای برچسبی از 9 نهنگ گوژپشت منحصر به فرد ، بدست آوردند. آنها 2 مانور را در مورد حباب کشف کردند ، "مارپیچ صعودی" و "حلقه های مضاعف". در مانور نخست ، نهنگ های گوژپشت در عمق 12 متری پایین شکار شنا کرده و سپس شروع به تولید حباب در یک مارپیچ صعودی پیرامون طعمه کرده و اقدام به شناکردن به سمت سطح می کند. مانور بعدی شامل 3 مرحله ی متفاوت است: حلقه مرجان، ضربه زدن با دم، حلقه ی دستگیری. جزئیات بیشتر در مورد این رفتار در (65) آمده است.

لازم به ذکر است که روش تغذیه حباب-تور رفتاری منحصر به فرد است و فقط در نهنگ های گوژپشت دیده می شود. در این اثر مانور تغذیه حباب-تور مارپیچی به صورت ریاضی و جهت انجام بهینه سازی، مدل شده است.

2.2 مدل ریاضی و الگوریتم بهینه سازی

در این بخش در ابتدا مدل ریاضی شکار محاصره ای، مانور تغذیه حباب-تور مارپیچی و جستجو برای شکار ارائه شده است. سپس الگوریتم WOA پیشنهاد می شود.

2.2.1 شکار محاصره ای

نهنگ های گوشت می توانند مکان شکار را شناسایی کرده و آنها را محاصره کنند. از آنجایی که مکان طراحی بهینه در فضای جستجو از راه قیاس شناخته نمی شود، الگوریتم WOA فرض می کند که بهترین راه حل نامزد حال حاضر، شکار هدف بوده و یا نزدیک به حالت مطلوب است. بعد از اینکه بهترین عامل جستجو شناسایی شد، عوامل دیگر جستجو سعی می کنند تا مکان خود را نسبت به بهترین عامل جستجو، به روزرسانی کنند. این رفتار به وسیله ی معادلات زیر بیان شده است :

$$\vec{D} = |\vec{C}\vec{X}^*(t) - \vec{X}(t)| \quad (2.1)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (2.2)$$

که در آن t تکرار جاری را نشان می دهد، A و C بردارهای ضرائب ، X^* بردار مکان بهترین راه حل بدست آمده در حال حاضر ، X بردار مکان ، $||$ قدر مطلق و \cdot ضرب نقطه ای (المان در المان) می باشد. لازم به ذکر است که در صورت وجود راه حل بهتر، X^* در هر تکرار باید بروز شود.

بردار A و C به صورت ذیل محاسبه می شود :

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (2.3)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (2.4)$$

که a به صورت خطی از مقدار 2 تا 0 و در طی تکرار ها کاهش می یابد (در هر دو فاز اکتشاف و استخراج) و r بردار تصادفی در فاصله 0 تا 1 است.

شکل 3 الف منطق پشت معادله ی 22 را برای مسئله ی 2 بعدی نشان می دهد. مکان X, Y عامل جستجو بر طبق مکان جاری بهترین ثابت (X^*, Y^*) می تواند بروز شود. مکان های متفاوت پیرامون بهترین عامل با توجه به مکان جاری و به وسیله ی تنظیم بردار های A و C بدست می آیند. مکان محتمل در حال به روز رسانی عامل جستجو در فضای 3 بعدی در شکل 3 ب رسم شده است. باید گفت که با تعریف بردار تصادفی r ، رسیدن به هر مکان در فضای جستجوی قرار گرفته در بین نقاط کلیدی ممکن بوده و در شکل 3 موجود است. بنابراین، معادله 22 به هر عامل جستجو اجازه می دهد تا مکانش را در همسایگی بهترین راه حل جاری به روز کرده و محاصره ی شکار را شبیه سازی کند.

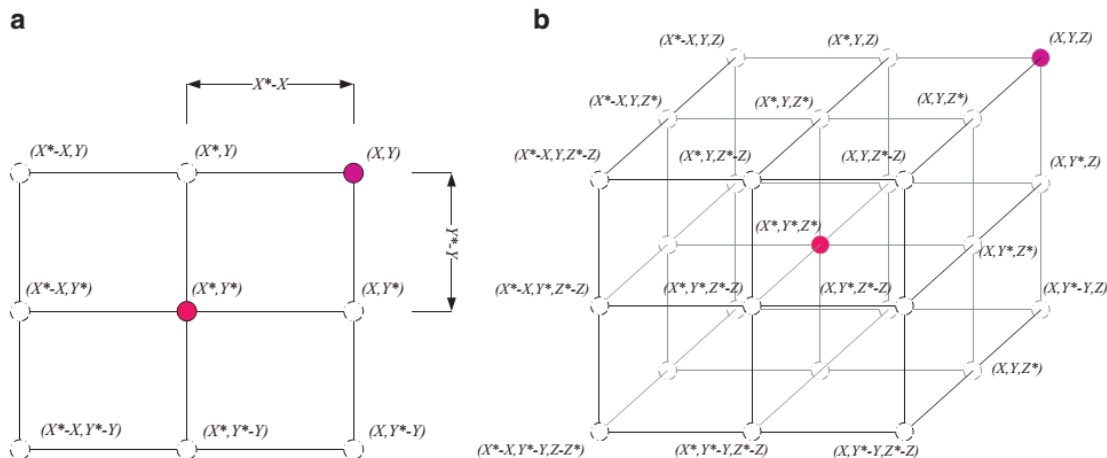


Fig. 3. 2D and 3D position vectors and their possible next locations (X^* is the best solution obtained so far).

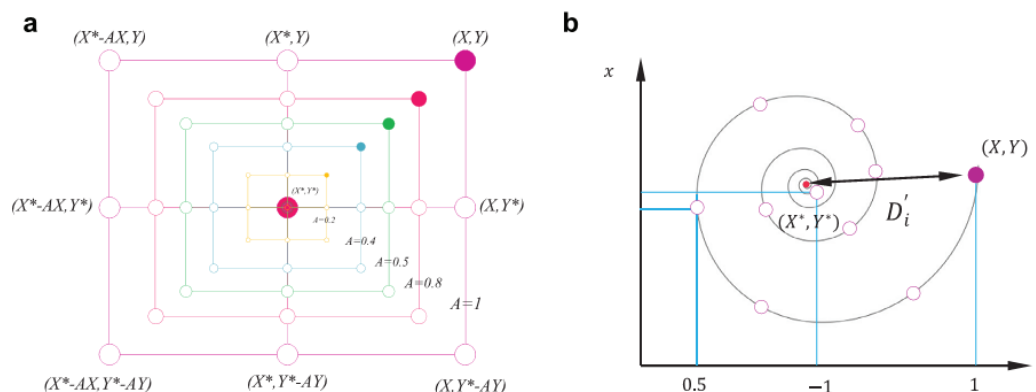


Fig. 4. Bubble-net search mechanism implemented in WOA (X^* is the best solution obtained so far): (a) shrinking encircling mechanism and (b) spiral updating position.

می توان مفهوم مشابهی را در مورد فضای جستجوی n بعدی نیز توسعه داد. عوامل جستجو به یک فوق مکعب n بعدی پیرامون بهترین راه حل بدست آمده منتقل می شوند. همانطور که در بخش قبل اشاره شد، نهنگ های گوژپشت با استراتژی حباب-تور به طعمه حمله می کنند. این روش به صورت ریاضی در زیر فرمول بندی می شود.

2.2.2 روش حمله حباب-تور (فاز استخراج)

جهت مدلسازی ریاضی رفتار حباب-تور نهنگ های گوژپشت، 2 روش به صورت زیر طراحی شده است:

1. مکانیزم محاصره ی انقباضی : این رفتار به وسیله ی افزایش مقدار a در معادله ی 2.3 حاصل می شود. به یاد داشته باشید که محدوده ی نوسان A به وسیله ی a کاهش می یابد. به عبارت دیگر، A مقداری تصادفی در فاصله ی a تا $-a$ است و a در طی تکرارها، از مقدار 2 تا 0 کاهش می یابد. با انتخاب مقادیر تصادفی A در فاصله ی 1 تا -1، می توان مکان جدید عامل جستجو را در هر کجای بین مکان اصلی عامل و مکان بهترین عامل کنونی، تعریف کرد. شکل 4، نشان دهنده ی مکان های محتمل از مختصات X, Y به X^*, Y^* بوده که در آن فضا 2 بعدی و A بین 0 تا 1 می باشد.
2. مکان در حال بروزرسانی مارپیچی

همانطور که در شکل 4 ب دیده می شود، این روش در ابتدا فاصله بین نهنگ قرار گرفته در مختصات X, Y و طعمه موجود در X^*, Y^* را محاسبه می کند. معادله ای مارپیچی بین موقعیت نهنگ و طعمه ایجاد می شود تا حرکت حلزونی شکل نهنگ گوژپشت را تقلید کند :

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (2.5)$$

در این معادله، $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ بوده و به فاصله ی آ امین نهنگ تا طعمه اشاره دارد (بهترین راه حل بدست آمده تا اینجا)، b ثابتی برای تعریف شکل مارپیچ لگاریتمی است، l عددی تصادفی بین 1 تا -1 و ضرب نقطه ای (المان در المان) می باشد.

به یاد داشته باشید که نهنگ گوژپشت، حول طعمه در امتداد یک دایره ی انقباضی و همزمان در مسیر مارپیچی شکلی به شنا در می آید. جهت مدل سازی این رفتار همزمان، فرض کرده ایم که نهنگ با احتمال 50 درصد از بین مکانیزم محاصره ی انقباضی و یا مدل مارپیچی یکی را انتخاب می کند تا موقعیت نهنگ ها در طول بهینه سازی به روز رسانی شود. مدل ریاضی بدین صورت است :

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \quad (2.6)$$

که در آن P عددی تصادفی بین 0 تا 1 است.

علاوه بر روش حباب-تور، نهنگ های گوژپشت به صورت تصادفی به دنبال طعمه می گردند. مدل ریاضی جستجو بدین صورت است.

2.2.3 جستجو برای طعمه (فاز اکتشاف)

روشی مشابه بر مبنای واریاسیون بردار A را می توان جهت جستجوی شکار (اکتشاف) به کار گرفت. در حقیقت، نهنگ های گوژپشت، بر طبق مکان یکدیگر، به صورت تصادفی به جستجو می پردازند. بنابراین، ما بردار A را با مقادیر تصادفی بزرگتر از 1 و یا کمتر از -1 به کار می گیریم تا عامل جستجو را مجبور به دور شدن از نهنگ مرجع کنیم. برخلاف فاز استخراج، ما جهت بروزرسانی موقعیت عامل جستجو در فاز اکتشاف به جای استفاده از داده های بهترین عامل جستجو، از انتخاب تصادفی عامل بهره می بریم. این مکانیزم به همراه $|\vec{A}| > 1$ بر اکتشاف تاکید دارند و به الگوریتم WOA اجازه می دهند تا جستجویی سراسری را به انجام رساند. مدل ریاضی به صورت زیر است:

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| \quad (2.7)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (2.8)$$

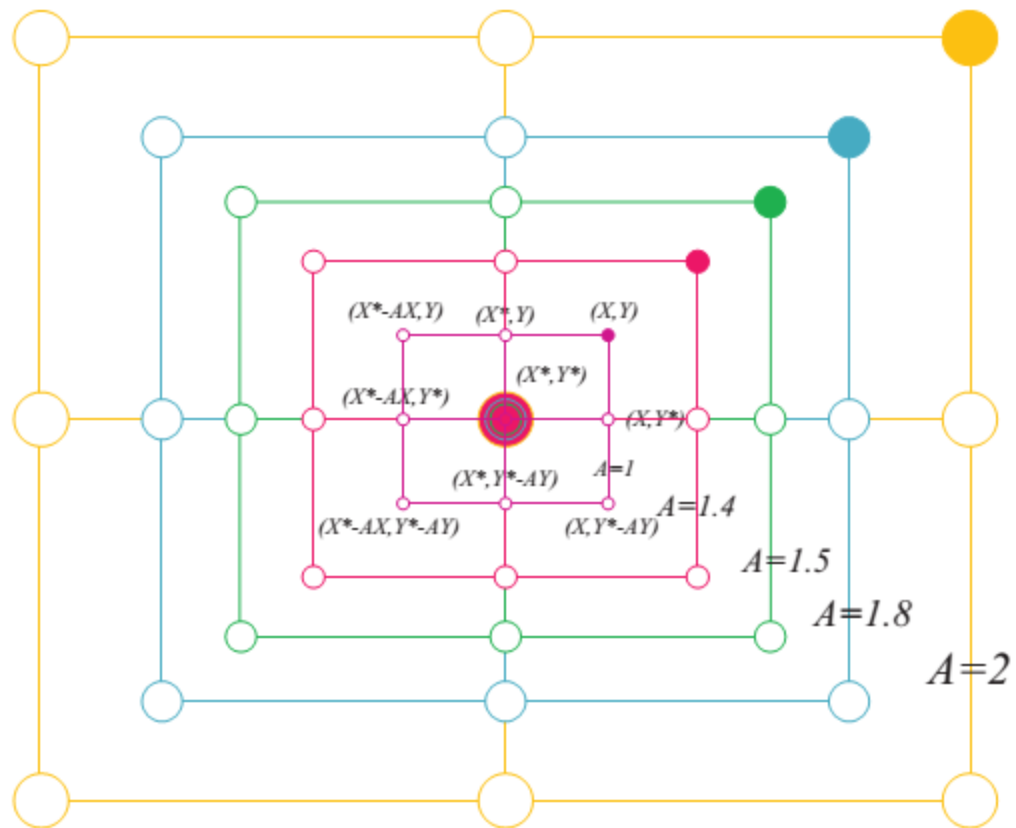


Fig. 5. Exploration mechanism implemented in WOA (X^* is a randomly chosen search agent).

```

Initialize the whales population  $X_i$  ( $i = 1, 2, \dots, n$ )
Calculate the fitness of each search agent
 $X^*$ =the best search agent
while ( $t < \text{maximum number of iterations}$ )
    for each search agent
        Update  $a$ ,  $A$ ,  $C$ ,  $l$ , and  $p$ 
        if1 ( $p < 0.5$ )
            if2 ( $|A| < 1$ )
                Update the position of the current search agent by the Eq. (2.1)
            else if2 ( $|A| \geq 1$ )
                Select a random search agent ( $X_{rand}$ )
                Update the position of the current search agent by the Eq. (2.8)
            end if2
        else if1 ( $p \geq 0.5$ )
            Update the position of the current search by the Eq. (2.5)
        end if1
    end for
    Check if any search agent goes beyond the search space and amend it
    Calculate the fitness of each search agent
    Update  $X^*$  if there is a better solution
     $t=t+1$ 
end while
return  $X^*$ 

```

Fig. 6. Pseudo-code of the WOA algorithm.

Table 2
Description of unimodal benchmark functions.

Function	V_no	Range	f_{min}
$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100,100]$	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10,10]$	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100,100]$	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	$[-100,100]$	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30,30]$	0
$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	$[-100,100]$	0
$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	30	$[-1.28,1.28]$	0

در این معادله، X_{rand} بردار موقعیت تصادفی انتخاب شده (نهنگ تصادفی) از جمعیت جاری می باشد. بعضی از موقعیت های محتمل پیرامون راه حلی خاص با شرط $A > 1$ در شکل 5 به نمایش در آمده اند.

الگوریتم WOA با مجموعه ای از راه حل های تصادفی شروع به کار می کند. در هر تکرار، عوامل جستجو موقعیت خود را با توجه به عامل جستجویی که تصادفی انتخاب شده و یا بهترین راه حل بدست آمده ی جاری ، به روز رسانی می کنند. پارامتر a جهت فراهم آوردن اکتشاف و استخراج، به ترتیب از مقدار 2 تا 0 کاهش می یابد. یک عامل جستجوی تصادفی در حالت $|A| > 1$ انتخاب می شود، این در حالی است که بهترین راه حل زمانی انتخاب می شود که جهت بروزرسانی موقعیت عوامل جستجو،

$|A| < 1$ باشد. بسته به مقدار p ، WOA این قابلیت را دارد تا بین حرکت دایروی و یا مارپیچی یکی را انتخاب کند. در نهایت، الگوریتم WOA با ارضای شرایط خاتمه، پایان می پذیرد.

شبه کد الگوریتم WOA در شکل 6 به نمایش در آمده است. از دیدگاه تئوری، WOA می تواند به عنوان بهینه سازی کلی در نظر گرفته شود، چرا که هم قابلیت استخراج و هم قابلیت اکتشاف را در خود دارد. علاوه بر این، مکانیزم فوق مکعبی ارائه شده، فضای جستجو را در همسایگی بهترین راه حل تعریف می کند و به عوامل دیگر این اجازه را می دهد تا از بهترین پیشینه ی داخل دامنه، بهره ببرند. واریاسیون تطبیقی بردار جستجوی A به الگوریتم WOA این امکان را می دهد تا به آرامی بین اکتشاف و استخراج گذر کند: با کاهش A ، بعضی از تکرار ها مربوط به اکتشاف می شوند ($|A| \geq 1$) و مابقی به استخراج بر می گردند ($|A| < 1$). نکته ی قابل ملاحظه این است که WOA تنها شامل دو پارامتر داخلی اصلی قابل تنظیم (A و C) می باشد.

اگرچه ممکن است که فرمول بندی WOA جهت بازتولید کامل رفتار نهنگ های گوژپشت شامل جهش و دیگر عملکردهای تکاملی باشد، تصمیم ما بر آن شده است که مقدار روش های ابتکاری و همچنین تعداد پارامترهای داخلی را حداقل سازیم تا یک نسخه بسیار ابتدایی از الگوریتم WOA را پیاده سازی کنیم. به هر حال، پیوند بین روش های جستجوی تکاملی ممکن است به موضوعی برای مطالعات آتی مبدل شود.

3. نتایج و بحث

کارایی عددی الگوریتم APW در این مطالعه با حل 29 مسئله ی بهینه سازی ریاضی مورد آزمایش قرار گرفته است. 23 مسئله ی اول توابع محک کلاسیکی هستند که در متون بهینه سازی (66-69) به کار رفته اند. WOA با الگوریتم های بهینه سازی مبتنی بر ازدحام و بروز مقایسه شد. جدول 2-4 خلاصه ای از مسائل محک و گزارش تابع هزینه، محدوده ی واریاسیون متغیرهای بهینه سازی و مقدار بهینه f_{min} ذکر شده در این بحث را ارائه می دهد. 6 مسئله ی تست دیگر مطرح شده در این مطالعه (جدول 5) از توابع محک مرکب موجود در جلسه مخصوص CEC 2005 بهره می برند (برای اطلاع از جزئیات توابع مرکب به مرجع 70 رجوع کنید). این توابع محک حاصل انتقال، چرخش، گسترش و یا ترکیب پیچیده ترین مسائل بهینه سازی موجود در اثر شماره 71 می باشند. به یاد داشته باشید که V_{n0} به تعداد متغیرهای طراحی در جدول 2-5 اشاره دارد. برای تمام الگوریتم ها، اندازه ی جمعیت و تعداد حداکثر تکرار به ترتیب برای 30 و 500 می باشد.

به طور کلی، توابع محک به 4 دسته تقسیم بندی می شوند: یک نمایی، چند نمایی، چند نمایی بعد ثابت و توابع مرکب. در شکل 7، نمودار های 2 بعدی مرسوم از تابع هزینه برای چند نمونه آزمایش در این مطالعه، به نمایش در آمده اند. باید یادآور شد که تفاوت بین توابع چندوجهی بعد ثابت در جدول 4 و توابع چند وجهی در جدول 3، قابلیت تعریف تعداد متغیرهای طراحی مطلوب می باشد. فرمول بندی های ریاضی از توابع تست بعد ثابت، امکان تغییر تعداد متغیرهای طراحی را به ما نمی دهد، اما آنها در مقایسه با توابع تست چند وجهی فضای جستجوی متفاوتی را برای ما در جدول 3 به ارمغان می آورند. لازم به ذکر است که توابع تست مرکب، با انتقال مقدار بهینه ی سراسری به مکان های تصادفی قبل از هر اجرا، گاهی قرار دادن مقدار بهینه سراسری بر روی مرزهای فضای جستجو و چرخش توابع با استفاده از فرمول $F(x) = f(R * x)$ که R ماتریس های چرخش عمود بدست آمده از روش سالمون، توابع آزمون چالش برانگیزی را فراهم می آورند. همانطور که در جدول 5 مشهود است، توابع تست مرکب، فضاهای جستجوی متفاوت n تابع تست اولیه دیگر را با شکل دادن و دستکاری محدوده ی پوشش آنها، ترکیب می نمایند. باید یادآور شد که σ به عنوان محدوده ی پوشش توابع تست اولیه و λ نماینگر سطح انبساط/انقباض هر تابع تست اولیه می باشد.

Table 3

Description of multimodal benchmark functions.

Function	V_no	Range	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
$F_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	$[-32, 32]$	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]$	0
$F_{12}(x) = \frac{\pi}{n} \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	30	$[-50, 50]$	0
$y_i = 1 + \frac{y_{i+1}}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$F_{13}(x) = 0.1 \{\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0

Table 4

Description of fixed-dimension multimodal benchmark functions.

Function	V_no	Range	f_{min}
$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$	2	$[-65, 65]$	1
$F_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	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
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	2	$[-5, 5]$	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	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

Table 5

Description of composite benchmark functions.

Function	V_no	Range	f_{min}
$F_{24}(CF1)$ $f_1, f_2, f_3, \dots, f_{10} = \text{Sphere Function}$ $[\phi_1, \phi_2, \phi_3, \dots, \phi_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	30	$[-5, 5]$	0
$F_{25}(CF2)$ $f_1, f_2, f_3, \dots, f_{10} = \text{Griewank's Function}$ $[\phi_1, \phi_2, \phi_3, \dots, \phi_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	30	$[-5, 5]$	0
$F_{26}(CF3)$ $f_1, f_2, f_3, \dots, f_{10} = \text{Griewank's Function}$ $[\phi_1, \phi_2, \phi_3, \dots, \phi_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [1, 1, 1, \dots, 1]$	30	$[-5, 5]$	0
$F_{27}(CF4)$ $f_1, f_2 = \text{Ackley's Function}, f_3, f_4 = \text{Rastrigin's Function}, f_5, f_6 = \text{Weierstrass Function},$ $f_7, f_8 = \text{Griewank's Function}, f_9, f_{10} = \text{Sphere Function}$ $[\phi_1, \phi_2, \phi_3, \dots, \phi_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	30	$[-5, 5]$	0
$F_{28}(CF5)$ $f_1, f_2 = \text{Rastrigin's Function}, f_3, f_4 = \text{Weierstrass Function}, f_5, f_6 = \text{Griewank's Function},$ $f_7, f_8 = \text{Ackley's Function}, f_9, f_{10} = \text{Sphere Function}$ $[\phi_1, \phi_2, \phi_3, \dots, \phi_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	30	$[-5, 5]$	0
$f_{29}(CF6)$ $f_1, f_2 = \text{Rastrigin's Function}, f_3, f_4 = \text{Weierstrass Function}, f_5, f_6 = \text{Griewank's Function}$ $f_7, f_8 = \text{Ackley's Function}, f_9, f_{10} = \text{Sphere Function}$ $[\phi_1, \phi_2, \phi_3, \dots, \phi_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [0.1 * 1/5, 0.2 * 1/5, 0.3 * 5/0.5, 0.4 * 5/0.5, 0.5 * 5/100,$ $0.6 * 5/100, 0.7 * 5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]$	30	$[-5, 5]$	0

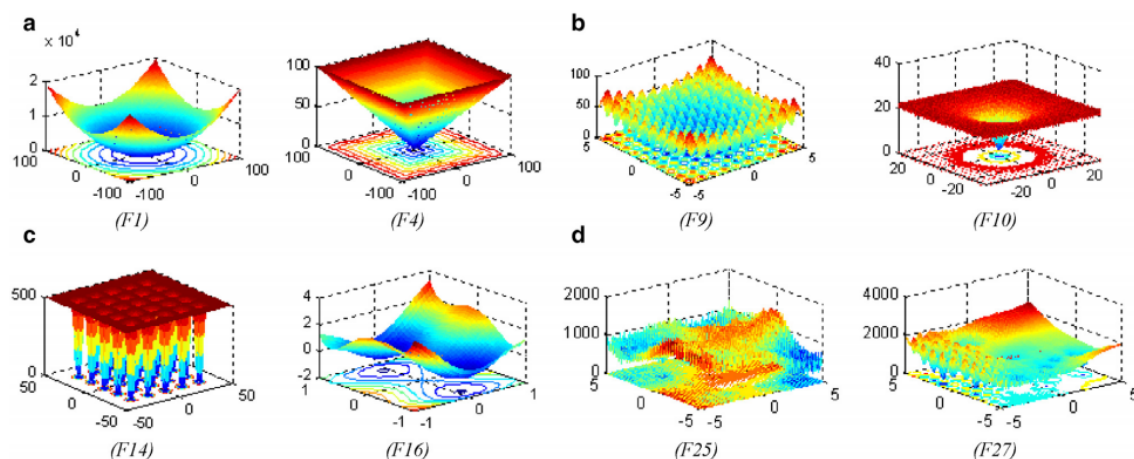


Fig. 7. Typical 2D representations of benchmark mathematical functions: (a) Unimodal functions; (b) Multimodal functions; (c) fixed-dimension multimodal functions; and (d) composite mathematical functions.

Table 6

Comparison of optimization results obtained for the unimodal, multimodal, and fixed-dimension multimodal benchmark functions.

F	WOA		PSO		GSA		DE		FEP	
	ave	std	ave	std	ave	std	ave	std	ave	std
F1	1.41E-30	4.91E-30	0.000136	0.000202	2.53E-16	9.67E-17	8.2E-14	5.9E-14	0.00057	0.00013
F2	1.06E-21	2.39E-21	0.042144	0.045421	0.055655	0.194074	1.5E-09	9.9E-10	0.0081	0.00077
F3	5.39E-07	2.93E-06	70.12562	22.11924	896.5347	318.9559	6.8E-11	7.4E-11	0.016	0.014
F4	0.072581	0.39747	1.086481	0.317039	7.35487	1.741452	0	0	0.3	0.5
F5	27.86558	0.763626	96.71832	60.11559	67.54309	62.22534	0	0	5.06	5.87
F6	3.116266	0.532429	0.000102	8.28E-05	2.5E-16	1.74E-16	0	0	0	0
F7	0.001425	0.001149	0.122854	0.044957	0.089441	0.04339	0.00463	0.0012	0.1415	0.3522
F8	-5080.76	695.7968	-4841.29	1152.814	-2821.07	493.0375	-11080.1	574.7	-12554.5	52.6
F9	0	0	46.70423	11.62938	25.96841	7.470068	69.2	38.8	0.046	0.012
F10	7.4043	9.897572	0.276015	0.50901	0.062087	0.23628	9.7E-08	4.2E-08	0.018	0.0021
F11	0.000289	0.001586	0.009215	0.007724	27.70154	5.040343	0	0	0.016	0.022
F12	0.339676	0.214864	0.006917	0.026301	1.799617	0.95114	7.9E-15	8E-15	9.2E-06	3.6E-06
F13	1.889015	0.266088	0.006675	0.008907	8.899084	7.126241	5.1E-14	4.8E-14	0.00016	0.000073
F14	2.111973	2.498594	3.627168	2.560828	5.859838	3.831299	0.998004	3.3E-16	1.22	0.56
F15	0.000572	0.000324	0.000577	0.000222	0.003673	0.001647	4.5E-14	0.00033	0.0005	0.00032
F16	-1.03163	4.2E-07	-1.03163	6.25E-16	-1.03163	4.88E-16	-1.03163	3.1E-13	-1.03	4.9E-07
F17	0.397914	2.7E-05	0.397887	0	0.397887	0	0.397887	9.9E-09	0.398	1.5E-07
F18	3	4.22E-15	3	1.33E-15	3	4.17E-15	3	2E-15	3.02	0.11
F19	-3.85616	0.002706	-3.86278	2.58E-15	-3.86278	2.29E-15	N/A	N/A	-3.86	0.000014
F20	-2.98105	0.376653	-3.26634	0.060516	-3.31778	0.023081	N/A	N/A	-3.27	0.059
F21	-7.04918	3.629551	-6.8651	3.019644	-5.95512	3.737079	-10.1532	0.0000025	-5.52	1.59
F22	-8.18178	3.829202	-8.45653	3.087094	-9.68447	2.014088	-10.4029	3.9E-07	-5.53	2.12
F23	-9.34238	2.414737	-9.95291	1.782786	-10.5364	2.6E-15	-10.5364	1.9E-07	-6.57	3.14

Table 7

Comparison of optimization results obtained for the composite benchmark functions.

F	WOA		PSO		GSA		DE		CMA-ES	
	ave	std	ave	std	ave	std	ave	std	ave	std
F24	0.568846	0.505946	100	81.65	6.63E-17	2.78E-17	6.75E-2	1.11E-1	100	188.56
F25	75.30874	43.07855	155.91	13.176	200.6202	67.72087	28.759	8.6277	161.99	151
F26	55.65147	21.87944	172.03	32.769	180	91.89366	144.41	19.401	214.06	74.181
F27	53.83778	21.621	314.3	20.066	170	82.32726	324.86	14.784	616.4	671.92
F28	77.8064	52.02346	83.45	101.11	200	47.14045	10.789	2.604	358.3	168.26
F29	57.88445	34.44601	861.42	125.81	142.0906	88.87141	490.94	39.461	900.26	8.32E-02

الگوریتم WOA برای هر تابع محک، 30 بار و با جمعیت های تصادفی متفاوت اجرا شد. نتایج آماری (به معنای میانگین تابع هزینه و انحراف معیار متناظر) در جداول 6 و 7 گزارش شده اند. WOA با PSO، GSA، DE، برنامه نویسی تکاملی سریع (FEP) و استراتژی تکاملی همراه با تطبیق ماتریس کوواریانس (CMA-ES) مقایسه شد. به یاد داشته باشید که بیشتر نتایج الگوریتم های مقایسه ای از (62) اتخاذ شده است.

3.1 تکامل قابلیت استخراج (توابع F1 تا F7)

توابع F_1 تا F_7 به دلیل داشتن یک مقدار بهینه ی کلی، تک وجهی می باشند. این توابع این امکان را بدست می دهند تا قابلیت استخراج الگوریتم های فوق ابتکاری ارزیابی شود. با توجه به جدول 6 می توان فهمید که WOA قابلیت رقابت با دیگر الگوریتم های فوق ابتکاری را دارد. به خصوص، WOA برای توابع F_1 و F_2 بهترین بهینه ساز بوده است و برای اغلب مسائل تست، حداقل در جایگاه دوم بهترین بهینه ساز قرار می گیرد. بنابراین می توان گفت که الگوریتم حاضر، استخراج بسیار خوبی را به ارمغان می آورد.

3.2 تکامل قابلیت اکتشاف (توابع F_8 تا F_{23})

برخلاف توابع تک وجهی، توابع چند وجهی شامل مقدار بهینه های داخلی بسیاری هستند که با افزایش اندازه مسئله (تعداد متغیرهای طراحی)، تعداد بهینه ها به صورت نمایی افزایش می یابد. بنابراین، این نوع از مسائل تست در مسیر ارزیابی قابلیت اکتشاف الگوریتم بهینه سازی، بسیار مفید می باشند. نتایج عنوان شده در جدول 6 برای توابع F_8 - F_{23} (یعنی همان توابع چند وجهی و چندوجهی با بعد ثابت) نشان می دهند که WOA قابلیت اکتشاف بسیار خوبی نیز دارد. در حقیقت، الگوریتم حاضر همواره کاراترین و یا الگوریتم دوم در بین بهترین الگوریتم ها جهت حل بسیاری از مسائل تست می باشد. این به دلیل مکانیزم های مجتمع اکتشاف در الگوریتم WOA است که منجر به رسیدن الگوریتم به مقدار بهینه ی سراسری می شوند.

3.3 توانایی فرار از حداقل محلی (توابع F_{24} - F_{29})

بهینه سازی توابع ریاضی مرکب، عملی به شدت چالش برانگیز است، چرا که تنها تعادلی مناسب بین اکتشاف و استخراج قابلیت اجتناب از حداقل محلی را فراهم می سازد. نتایج بهینه سازی موجود در جدول 7 نمایانگر این نکته هستند که الگوریتم WOA بهترین بهینه ساز در حل 3 مسئله ی تست بوده است و می تواند در موارد دیگر، با سایر الگوریتم ها به رقابت بپردازد. با توجه به این نکته، اثبات می شود که WOA می تواند به خوبی تعادل فازهای استخراج و اکتشاف را برقرار سازد. استراتژی تطبیقی به کار رفته جهت بروز رسانی بردار A به برقراری این تعادل کمک می کند: بعضی از تکرار ها مربوط به فاز اکتشاف ($|A| > 1$) و بقیه مربوط به فاز استخراج ($|A| \leq 1$) می باشد.

3.4 تحلیل رفتار همگرایی

مشاهده کردیم که عوامل جستجوی WOA میل دارند تا مناطق محتمل فضای طراحی را به صورت گسترده مورد جستجو قرار داده و بهترین مورد را استخراج کنند. عوامل جستجو در مراحل اولیه ی فرایند بهینه سازی به صورت ناگهانی تغییر کرده و به تدریج همگرا می شوند. بر طبق اثر برگ و همکارانش (75)، چنین رفتاری می تواند تضمین کند که یک الگوریتم مبتنی بر جمعیت در نهایت به نقطه ای در یک نقطه در فضای جستجو همگرا خواهد شد. برای چند مسئله، خم های همگرای WOA، PSO و GSA در شکل 8 مقایسه شده اند. می توان دید که WOA به تنهایی قادر به رقابت با دیگر الگوریتم های فوق ابتکاری مدرن می باشد. خمهای همگرا در WOA، PSO و GSA در شکل 8 به نمایش درآمده و نرخ همگرایی الگوریتم ها را شرح داده اند. به یاد داشته باشید که بهترین-تاکنون میانگین، اشاره به میانگین بهترین راه حل بدست آمده تا کنون در هر تکرار در طی هر سی اجرا دارد. همانطور که ممکن است در این شکل نمایان باشد، الگوریتم WOA 3 رفتار همگرایی متفاوت را در حین بهینه سازی توابع تست نشان می دهد. اولاً، همگرایی الگوریتم WOA با افزایش تعداد تکرارها، سرعت می گیرد. این به دلیل مکانیزم تطبیقی ارائه شده برای WOA می باشد که به الگوریتم کمک می کند تا به دنبال مناطق محتمل فضای جستجو در گام های ابتدایی تکرار گردد و پس از گذشت تقریباً نیمی از تکرارها،

همگرایی سریعی به سمت نقطه ی بهینه صورت می گیرد. این رفتار در F1 ، F3 ، F4 و F9 مشهود است. رفتار دوم همگرایی به سمت نقطه بهینه ای است که تنها در تکرار آخر صورت گرفته است، نظیر آنچه که در F8 و F21 مشاهده می شود. این رفتار یحتمل به دلیل شکست WOA در یافتن راه حلی مناسب برای استخراج در گام های ابتدایی تکرار و در زمان اجتناب از نقطه بهینه ی محلی است، بنابراین این الگوریتم برای پیدا کردن راه حل های مناسب در فضای جستجو جهت همگرا شدن به سمت آنها ، جستجو را ادامه می دهد.

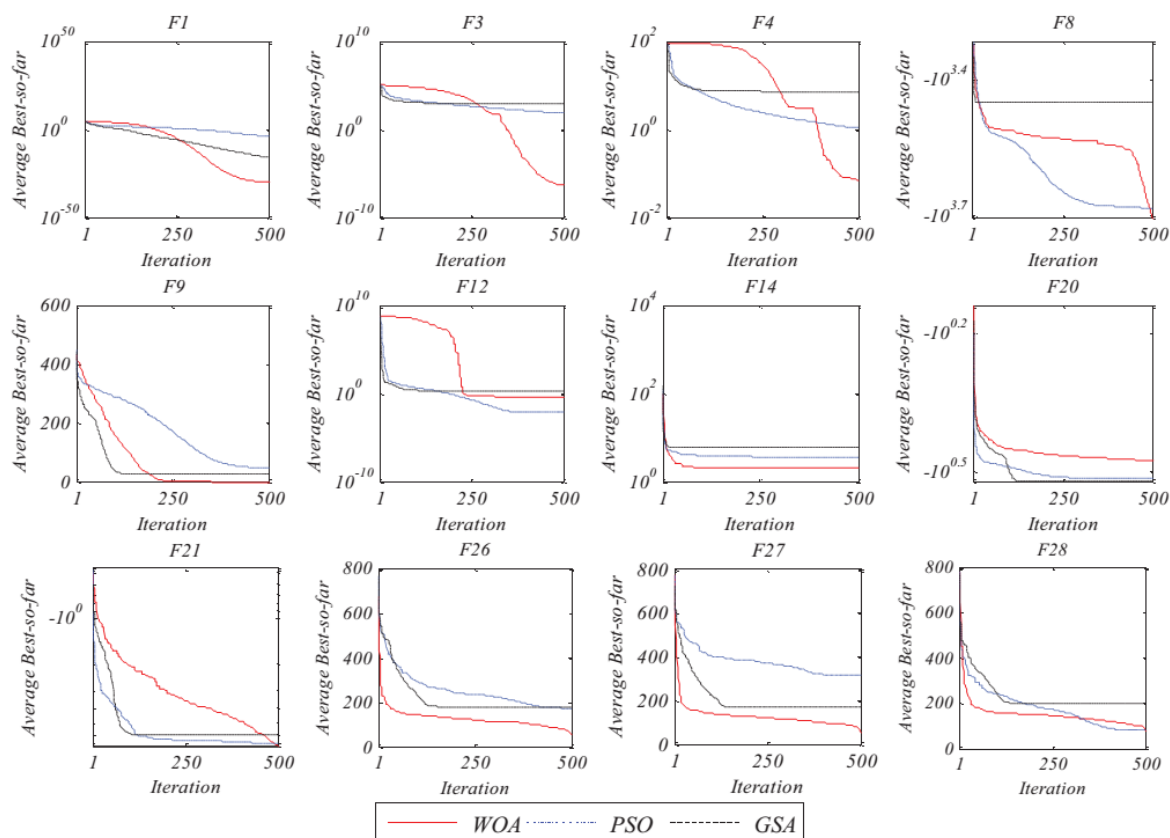


Fig. 8. Comparison of convergence curves of WOA and literature algorithms obtained in some of the benchmark problems.

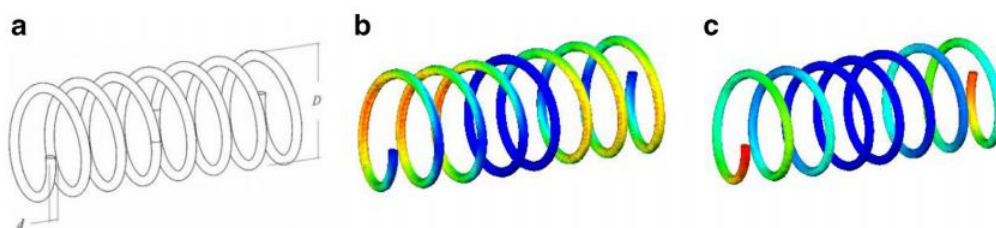


Fig. 9. (a) Schematic of the spring; (b) stress distribution evaluated at the optimum design; and (c) displacement distribution evaluated at the optimum design.

آخرین رفتار، همگرایی سریع از گام های ابتدایی تکرار هاست که در F14 ، F26 ، F27 ، F28 مشاهده می شود. از آنجایی که F26 ، F27 و F28 چالش برانگیزترین تست های این بخش هستند (توابع تست مرکب) ، این نتایج نشان می دهند که الگوریتم WOA از یک تعادل خوب بین استخراج و اکتشاف بهره می برد ، به این صورت که این تعادل موجب پیدا کردن نقطه بهینه سراسری می شود. روی هم رفته می توان گفت ، به نظر می رسد نرخ موفقیت الگوریتم WOA در حل مسائل چالش برانگیز ، بالا می باشد.

به عنوان خلاصه، نتایج این بخش خصوصیات متفاوتی از الگوریتم WOA پیشنهادی را نمایش می دهند. قابلیت بالای اکتشاف WOA، به دلیل مکانیزم بروز رسانی موقعیت نهنگ ها و با استفاده از معادله ی 2.8 می باشد. این معادله نیاز دارد تا نهنگ ها در گام های ابتدایی تکرارها به صورت تصادفی پیرامون یکدیگر حرکت کنند. به هر حال، بر طبق معادله ی 2.6 بر روی همگرایی و استخراج بالا تاکید می شود. این معادله به نهنگ ها این امکان را می دهد تا به سرعت موقعیت خودشان را پیرامون یکدیگر تغییر داده و یا در مسیری مارپیچ شکل به سمت بهترین راه بدست آمده حاضر، به حرکت درآیند. از آنجایی که این 2 فاز به صورت جداگانه و تقریباً هرکدام در نیمی از تکرارها به انجام می رسند، WOA همزمان میزان بالایی از اجتناب نسبت به بهینه ی محلی و همچنین سرعت همگرایی در طی تکرار ها را به نمایش می گذارد. به هر حال، PSO و GSA فاقد عملگرهایی برای اختصاص تعدادی از تکرار ها به اکتشاف و یا استخراج می باشند. به عبارت دیگر، PSO و GSA (و هر الگوریتم شبیه سازی دیگر) از یک فرمول برای بروز رسانی موقعیت عوامل جستجو بهره می برند که این، موجب افزایش شباهت ایستایی در نطقه بهینه ی محلی می شود. در بخش های بعد، عمل WOA با استفاده از مسائل مهندسی واقعی چالش برانگیزتری، تایید می شود.

4. WOA برای مسائل مهندسی کلاسیک

در این بخش، WOA با 6 مسئله ی طراحی مقید مهندسی مورد آزمایش قرار می گیرد: فنر منقبض / منبسط، اتصال میله، مخزن فشار، خرپا 15، 25 و 52 میله ای.

از آنجایی که مسائل این بخش قیود متفاوتی دارند، نیازمند این هستیم تا از روش دستکاری قیود استفاده کنیم. توابع جریمه ی متفاوتی در این متن وجود دارد (76): جرائم ایستا، پویا، التهاب، اطباقی، همراه با تکامل و مرگ. آخرین تابع جریمه، جریمه ی مرگ، آسانترین روش بوده که مقدار هدف بزرگی (جهت حداقل سازی) را مشخص می کند. این فرایند با استفاده از الگوریتم های ابتکاری در طول بهینه سازی، به صورت خودکار منجر به دور انداختن راه حل های غیر ممکن می شود. مزایای این روش، سادگی و هزینه ی محاسباتی کم می باشد. اگرچه، این روش اطلاعات راه حل های غیر ممکن را نادیده می گیرد، این اطلاعات ممکن است در حل مسائل با مناطق غیر عملی و تحت سلطه کمک کننده باشند. به دلیل ساده کردن موضوع، ما در این بخش برای سر و کار داشتن با قیود، الگوریتم WOA را به تابع جریمه ی پنالتی مجهز می کنیم.

Table 8

Comparison of WOA optimization results with literature for the tension/compression spring design problem.

Algorithm	Optimum variables			Optimum weight
	d	D	N	
WOA	0.051207	0.345215	12.004032	0.0126763
GSA	0.050276	0.323680	13.525410	0.0127022
PSO (Ha and Wang)	0.051728	0.357644	11.244543	0.0126747
ES (Coello and Montes)	0.051989	0.363965	10.890522	0.0126810
GA (Coello)	0.051480	0.351661	11.632201	0.0127048
RO (Kaveh and Khayatazad)	0.051370	0.349096	11.76279	0.0126788
Improved HS (Mahdavi et al.)	0.051154	0.349871	12.076432	0.0126706
DE (Huang et al.)	0.051609	0.354714	11.410831	0.0126702
Mathematical optimization (Belegundu)	0.053396	0.399180	9.1854000	0.0127303
Constraint correction (Arora)	0.050000	0.315900	14.250000	0.0128334

4.1 طراحی فنر فشرده/باز شده

هدف این مسئله ی تست، حداقل کردن وزن فنر فشرده شده/باز شده موجود در شکل 9 می باشد (77-79). طراحی بهینه، باید قیود تنش برشی، فرکانس موج و شکست را ارضا کند. 3 متغیر طراحی وجود دارد: قطر سیم (d)، قطر متوسط چنبره (D)، تعداد چنبره های فعال (N).

مسئله ی بهینه سازی به صورت ذیل فرمول بندی می شود:

Consider $\vec{x} = [x_1 \ x_2 \ x_3] = [d \ D \ N]$,

Minimize $f(\vec{x}) = (x_3 + 2)x_2x_1^2$

Subject to $g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$,

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0, \quad (5.1)$$

$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0,$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$$

Variable range $0.05 \leq x_1 \leq 2.00$,

$$0.25 \leq x_2 \leq 1.30,$$

$$2.00 \leq x_3 \leq 15.0$$

این نمونه تست با استفاده از روش های ریاضی (برای مثال، تصحیح قیود در هزینه ی ثابت (77) و توابع جریمه (78)) و یا روش های فوق ابتکاری مانند PSO ، استراتژی تکامل، GA ، جستجوی هماهنگی منظم، تکامل تفاضلی و الگوریتم بهینه سازی پرتو حل می شود.

نتایج بهینه سازی WOA در جدول 8 آمده است. استراتژی دستکاری قیود با تابع جریمه متفاوتی جهت انجام مقایسه ای عادلانه به کار رفته است (85). می توان دید که WOA از تمام الگوریتم های دیگر به جز HS و DE ، بهتر عمل کرده است.

همچنین جدول 9 شامل میانگین، انحراف معیار و تعداد تحلیل های صورت گرفته توسط 3 الگوریتم و در 30 اجرا می باشد. به یاد داشته باشد که ما از 10 عامل جستجو و حداکثر تعداد 500 تکرار برای حل این مسئله استفاده کرده ایم. این جدول نشان می دهد که WOA به طور میانگین از GSA و PSO بهتر عمل کرده و به تحلیل های کمتری نیازمند است (توابع ارزیابی).

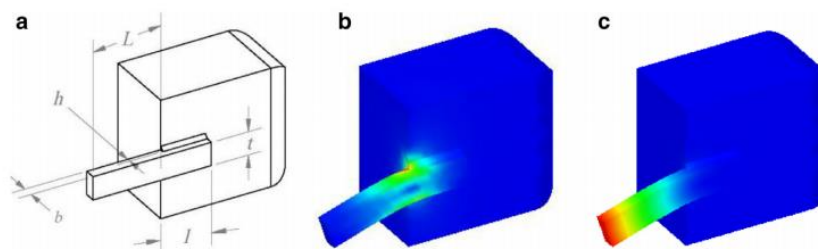


Fig. 10. Welded beam design problem: (a) Schematic of the weld; (b) Stress distribution evaluated at the optimum design; (c) Displacement distribution at the optimum design.

هدف این مسئله ی تست، حداقل کردن هزینه ی ساخت اتصال جوشکاری نشان داده شده در شکل 10 می باشد. قیود بهینه سازی عبارتند از استرس برشی (τ) ، انحنای انتهای میله (δ)، بار پیچش (Pc) و تنش خمشی در میله (σ). 4 متغیر بهینه سازی وجود دارد : ضخامت میله (h)، طول نوار کلمپ شده (l) ، ارتفاع نوار (t) و ضخامت نوار (b). فرمول ریاضی مسئله ی بهینه سازی به صورت زیر است :

Consider $\vec{x} = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b]$,

Minimize $f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$,

Subject to $g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leq 0$,

$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0$,

$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leq 0$,

$g_4(\vec{x}) = x_1 - x_4 \leq 0$,

$g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0$,

$g_6(\vec{x}) = 0.125 - x_1 \leq 0$

$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2)$
 $- 5.0 \leq 0$ (5.2)

Variable range $0.1 \leq x_1 \leq 2$,

$0.1 \leq x_2 \leq 10$,

$0.1 \leq x_3 \leq 10$,

$0.1 \leq x_4 \leq 2$

$$\begin{aligned}
\text{where } \tau(\vec{x}) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \\
\tau' &= \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P\left(L + \frac{x_2}{2}\right), \\
R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \\
J &= 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \\
\sigma(\vec{x}) &= \frac{6PL}{x_4x_3^2}, \quad \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4} \\
P_c(\vec{x}) &= \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), \\
P &= 6000 \text{ lb}, \quad L = 14 \text{ in.}, \quad \delta_{\max} = 0.25 \text{ in.}, \\
E &= 30 \times 10^6 \text{ psi}, \quad G = 12 \times 10^6 \text{ psi}, \\
\tau_{\max} &= 13,600 \text{ psi}, \quad \sigma_{\max} = 30,000 \text{ psi},
\end{aligned}$$

این مسئله ی بهینه سازی توسط Coello و Deb و با استفاده از GA حل شد، این در حالی است که Lee و Geem از HS ، مهندوی و همکارانش از HS بهبود یافته، کاوه و خیاطزاده از RO و کاوه و مهندوی از CBO برای حل این مسئله استفاده کرده اند. روش تصادفی ریچاردسون ، روش سادک، دیویدسون-فلچر-پاول و روش تقریب خطی پی در پی گریفیت و استوارت، همگی روش های ریاضی بودند که Radgsdell و Philips از آنها استفاده کرده اند. نتایج بهینه سازی در جدول 10 ، بیانگر این است که WOA در رتبه ی سوم بهترین طراحی ها قرار می گیرد.

Table 10

Comparison of WOA optimization results with literature for the welded beam design problem.

Algorithm	Optimum variables				Optimum cost
	h	l	t	b	
WOA	0.205396	3.484293	9.037426	0.206276	1.730499
GSA	0.182129	3.856979	10.00000	0.202376	1.879952
CBO	0.205722	3.47041	9.037276	0.205735	1.724663
RO	0.203687	3.528467	9.004233	0.207241	1.735344
Improved HS	0.20573	3.47049	9.03662	0.2057	1.7248
GA Coello)	N/A	N/A	N/A	N/A	1.8245
GA (Deb)	N/A	N/A	N/A	N/A	2.3800
GA (Deb)	0.2489	6.1730	8.1789	0.2533	2.4331
HS (Lee and Geem)	0.2442	6.2231	8.2915	0.2443	2.3807
Random	0.4575	4.7313	5.0853	0.6600	4.1185
Simplex	0.2792	5.6256	7.7512	0.2796	2.5307
David	0.2434	6.2552	8.2915	0.2444	2.3841
APPROX	0.2444	6.2189	8.2915	0.2444	2.3815

جدول 11 شامل نتایج آماری بعضی از الگوریتم ها ، در طول 30 اجرای مستقل می باشد. ما از 20 عامل جستجو و حداکثر 500 تکرار برای حل این مسئله استفاده کرده ایم. نکته ای که ممکن است از این جدول بدست آید، مجدداً می بینیم که WOA به طور میانگین عملکرد بهتری داشته است. علاوه بر این، این الگوریتم به حداقل تعداد تحلیل ها برای پیدا کردن بهترین طراحی بهینه نیازمند است.

Table 11

Comparison of WOA statistical results with literature for the welded beam design problem.

Algorithm	Average	Standard deviation	Function evaluation
WOA	1.7320	0.0226	9900
PSO	1.7422	0.01275	13770
GSA	3.5761	1.2874	10750

4.3 طراحی مخزن تحت فشار

در این مسئله، هدف حداقل سازی هزینه ی کلی (مواد، شکل دهی و جوشکاری) مخزن تحت فشار استوانه ای موجود در شکل 11 می باشد. هر دو انتهای مخزن بسته شده است، در حالیکه که سر آن نیمکره ی شکل است. 4 متغیر طراحی داریم : ضخامت پوسته (Ts) ، ضخامت سر (Th)، شعاع داخلی (R) و طول بخش استوانه ای بدون در نظر گرفتن سر (L). مسئله شامل 4 قید بهینه سازی می باشد و اینگونه فرمول بندی می شود :

Consider $\vec{x} = [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ T_h \ R \ L]$,
Minimize $f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2$
 $+ 3.1661x_1^2x_4 + 19.84x_1^2x_3$,
Subject to $g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0$,
 $g_2(\vec{x}) = -x_3 + 0.00954x_3 \leq 0$,
 $g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0$,
 $g_4(\vec{x}) = x_4 - 240 \leq 0$, (5.3)
Variable range $0 \leq x_1 \leq 99$,
 $0 \leq x_2 \leq 99$,
 $10 \leq x_3 \leq 200$,
 $10 \leq x_4 \leq 200$,

این مورد توسط محققین زیادی و با استفاده از روش های فوق ابتکاری (نظیر PSO ، GA ، ES ، DE ، ACO و HS بهبود یافته) و یا روش های ریاضی (افزاینده ی لاگرانژ و شاخه و کران) حل شده است.

نتایج بهینه سازی در جدول 12 مقایسه شده است. WOA ، بعد از ACO و DE سومین بهینه سازی کارا می باشد. اگرچه، دریافتیم که ACO یکی از قیود را ارضا نکرد و از آن تجاوز کرد. بنابراین، می توان این چنین عنوان کرد که WOA قادر بوده تا دومین طراحی بهینه ممکن را برای مسئله ی طراحی مخزن فشار، به ارمغان آورد.

Table 12
Comparison of WOA optimization results with literature for the pressure vessel design problem.

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
WOA	0.812500	0.437500	42.0982699	176.638998	6059.7410
Improved HS	1.125000	0.625000	58.29015	43.69268	7197.730
GSA	1.125000	0.625000	55.9886598	84.4542025	8538.8359
PSO (He and Wang)	0.812500	0.437500	42.091266	176.746500	6061.0777
GA (Coello)	0.812500	0.434500	40.323900	200.000000	6288.7445
GA (Coello and Montes)	0.812500	0.437500	42.097398	176.654050	6059.9463
GA (Deb and Gene)	0.937500	0.500000	48.329000	112.679000	6410.3811
ES (Montes and Coello)	0.812500	0.437500	42.098087	176.640518	6059.7456
DE (Huang et al.)	0.812500	0.437500	42.098411	176.637690	6059.7340
ACO (Kaveh and Talataheri)	0.812500	0.437500	42.103624	176.572656	6059.0888 (infeasible)
Lagrangian multiplier (Kannan)	1.125000	0.625000	58.291000	43.6900000	7198.0428
Branch-bound (Sandgren)	1.125000	0.625000	47.700000	117.701000	8129.1036

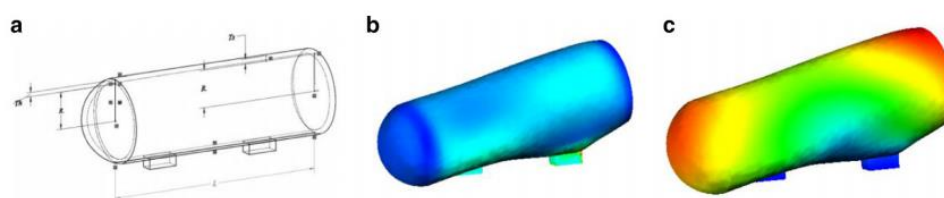


Fig. 11. Pressure vessel design problem: (a) schematic of the vessel; (b) stress distribution evaluated at the optimum design; and (c) displacement distribution evaluated at the optimum design.

Table 13

Comparison of WOA statistical results with literature for the pressure vessel design problem.

Algorithm	Average	Standard deviation	Function evaluation
WOA	6068.05	65.6519	6300
PSO	6531.10	154.3716	14790
GSA	8932.95	683.5475	7110

نتایج آماری بعضی از الگوریتم ها در حل مسئله ی طراحی فشار در جدول 13 آمده اند. ما از 20 عامل جستجو و به همراه حداکثر 500 تکرار برای حل این مسئله استفاده کرده ایم. بر طبق نتایج این جدول، یکبار دیگر می بینیم که WOA نسبت به الگوریتم های PSO و GSA نتایج بهتری را به همراه داشته است، علاوه بر این، جدول 13 نشانگر این است که WOA بهترین طراحی ممکن را با حداقل تعداد ارزیابی های عملکرد به انجام رسانیده است.

4.4 طراحی خرپای 15 میله ای

این مسئله، مسئله ای گسسته است که در آن هدف، حداقل سازی وزن خرپا می باشد. طراحی بهینه نهایی برای این مسئله باید 46 قید شامل 15 قید تنش، 15 قید فشردگی و 16 قید جابجایی را ارضا کند. همچنین با توجه به شکل 12، 8 گره و 15 میله وجود دارد، بنابراین در مجموع 15 متغیر خواهیم داشت. همچنین از این شکل در می یابیم که 3 بار، به گره های P1، P2 و P3 اعمال شده است. فزییات دیگر برای این مسئله عبارتند از :

- $\rho = 7800 \text{ kg/m}^3$
- $E = 200 \text{ GPa}$

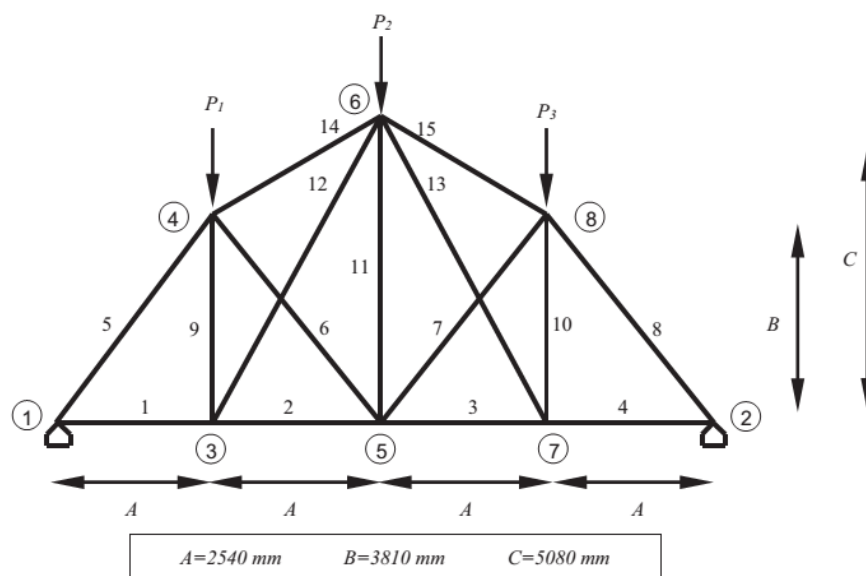


Fig. 12. Structure of a 15-bar truss.

Table 14

Comparison of WOA optimization results with literature for the 15-bar truss design problem.

Variables (mm ²)	HGA [97]	PSO [31]	PSOPC [31]	HPSO [31]	MBA [96]	SOS [98]	WOA
A1	308.6	185.9	113.2	113.2	113.2	113.2	113.2
A2	174.9	113.2	113.2	113.2	113.2	113.2	113.2
A3	338.2	143.2	113.2	113.2	113.2	113.2	113.2
A4	143.2	113.2	113.2	113.2	113.2	113.2	113.2
A5	736.7	736.7	736.7	736.7	736.7	736.7	736.7
A6	185.9	143.2	113.2	113.2	113.2	113.2	113.2
A7	265.9	113.2	113.2	113.2	113.2	113.2	113.2
A8	507.6	736.7	736.7	736.7	736.7	736.7	736.7
A9	143.2	113.2	113.2	113.2	113.2	113.2	113.2
A10	507.6	113.2	113.2	113.2	113.2	113.2	113.2
A11	279.1	113.2	113.2	113.2	113.2	113.2	113.2
A12	174.9	113.2	113.2	113.2	113.2	113.2	113.2
A13	297.1	113.2	185.9	113.2	113.2	113.2	113.2
A14	235.9	334.3	334.3	334.3	334.3	334.3	334.3
A15	265.9	334.3	334.3	334.3	334.3	334.3	334.3
Optimal weight (kg)	142.117	108.84	108.96	105.735	105.735	105.735	105.735
No. of analyses	N/A	~18,500	~16,000	7500	2000	5000	4000

- Stress limitation = 120 MPa
- Maximum stress = 115.37 MPa
- Displacement limitation = 10 mm
- Maximum displacement = 4.24 mm
- Design variabe set = $\left\{ \begin{array}{l} 113.2, 143.2, 145.9, 174.9, 185.9, \\ 235.9, 265.9, 297.1 \\ 308.6, 334.3, 338.2, 497.8, 507.6, \\ 736.7, 791.2, 1063.7 \end{array} \right\}$

این مسئله با 3 مجموعه بار متفاوت به صورت ذیل، حل شده است:

- Case 1 : $P_1 = 35 \text{ kN}$, $P_2 = 35 \text{ kN}$, $P_3 = 35 \text{ kN}$
- Case 2 : $P_1 = 35 \text{ kN}$, $P_2 = 0 \text{ kN}$, $P_3 = 35 \text{ kN}$
- Case 3 : $P_1 = 35 \text{ kN}$, $P_2 = 0 \text{ kN}$, $P_3 = 0 \text{ kN}$

ما این مسئله را با 30 عامل جستجو و در طول 500 تکرار حل کردیم، نتایج در جدول 14 آمده است. از آنجایی که این مسئله، مسئله ای گسسته است، عوامل جستجوی WOA در طول بهینه سازی به نزدیکترین عدد صحیح گرد شدند.

جدول 14 نشان می دهد که الگوریتم WOA قادر به پیدا کردن ساختار مشابهی در مقایسه با HPSO، SOS، و MBA بوده است. این نشان دهنده ی آن است که این الگوریتم می تواند به خوبی با الگوریتم های دیگر به رقابت بپردازد. علاوه بر این، تعداد ارزیابی های تابع WOA، در رتبه ی دوم پس از الگوریتم MBA قرار گرفته است.

4.5 طراحی خرپای 25 میله ای

این مسئله ، مسئله ی طراحی خرپای معروف دیگری می باشد. همانطور که ممکن است در شکل 13 دیده شود، 19 گره وجود دارند که 4 تای آنها ثابت است. 25 میله (اعضا با مناطق مقطعی) نیز وجود داشته که در 5 گروه طبقه بندی می شوند:

- Group 1 : A_1
- Group 2 : A_2, A_3, A_4, A_5
- Group 3 : A_6, A_7, A_8, A_9
- Group 4 : A_{10}, A_{11}
- Group 5 : A_{12}, A_{13}
- Group 6 : A_{14}, A_{15}, A_{17}
- Group 7 : $A_{18}, A_{19}, A_{20}, A_{21}$
- Group 8 : $A_{22}, A_{23}, A_{24}, A_{25}$

بنابراین، این مسئله 8 پارامتر دارد. دیگر فرضیات مسئله به صورت زیر است :

- $\rho = 0.0272 \text{ N/cm}^3 (0.1 \text{ lb/in.}^3)$

Table 16

Two loading conditions for the 25-bar truss design problem.

Node	Case 1			Case 2		
	P_x kips (kN)	P_y kips (kN)	P_z kips (kN)	P_x kips (kN)	P_y kips (kN)	P_z kips (kN)
1	0.0	20.0 (89)	-5.0 (22.25)	1.0(4.45)	10.0 (44.5)	-5.0 (22.25)
2	0.0	-20.0 (89)	-5.0 (22.25)	0.0	10.0 (44.5)	-5.0 (22.25)
3	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0
6	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0

این مسئله در نسخه های باینری و پیوسته حل شده است. ما برای متفاوت کردن مسئله ی بخش های قبل، از نسخه ی پیوسته ی این مسئله استفاده کرده ایم. ما این مسئله را با 30 عامل جستجو و در طول 500 تکرار حل کردیم، نتایج در جدول 17 آمده است.

جدول 17 نشان می دهد که بهترین وزن بهینه ی بدست آمده به وسیله ی روش WOA، 544.608 بوده که از دیگر الگوریتم ها بهتر است. نتایج میانگین و انحراف معیار WOA، نشان دهنده ی برتر بودن آن نسبت به سایر روش ها به جز CSP می باشد. اگرچه، اساس الگوریتم WOA به تعداد تحلیل های کمتری نسبت به CSP و جهت پیدا کردن بهترین راه حل طراحی، نیازمند است.

Table 17

Statistical comparison of WOA optimization results with literature for the 25-bar truss design problem.

Element group	Results of algorithms						
	ACO [103]	BB-BC [105]	PSO [108]	HPSACO [106]	CSS [107]	CSP [104]	WOA
A1	0.01	0.01	0.01	0.01	0.01	0.01	0.01
A2-A5	2.042	1.993	2.052	2.054	2.003	1.91	2.053
A6-A9	3.001	3.056	3.001	3.008	3.007	2.798	3.004
A10-A11	0.01	0.01	0.01	0.01	0.01	0.01	0.01
A12-A13	0.01	0.01	0.01	0.01	0.01	0.01	0.01
A14-A17	0.684	0.665	0.684	0.679	0.687	0.708	0.68
A18-A21	1.625	1.642	1.616	1.611	1.655	1.836	1.61
A22-A25	2.672	2.679	2.673	2.678	2.66	2.645	2.675
Best weight (lb)	545.03	545.16	545.21	544.99	545.10	545.09	544.608
Average weight (lb)	545.74	545.66	546.84	545.52	545.58	545.20	544.773
Std. dev. (lb)	0.94	0.491	1.478	0.315	0.412	0.487	0.498
No. of analyses	3520	12500	9596	9875	7000	17,500	9450

4.6 طراحی خریای 52 میله ای

این مسئله، مسئله ی طراحی خریای معروف دیگری است که متغیرهای آن، گسسته هستند. همانطور که در شکل 14 می بینید، 52 میله و 20 گره وجود داشته که 4 عدد از آن گره ها ثابت است. خریا 52 میله دارد، که در 12 گروه طبقه بندی می شوند :

- *Group 1* : A_1, A_2, A_3, A_4
- *Group 2* : $A_5, A_6, A_7, A_8, A_9, A_{10}$
- *Group 3* : A_{11}, A_{12}, A_{13}
- *Group 4* : $A_{14}, A_{15}, A_{16}, A_{17}$
- *Group 5* : $A_{18}, A_{19}, A_{20}, A_{21}, A_{22}, A_{23}$
- *Group 6* : A_{24}, A_{25}, A_{26}
- *Group 7* : $A_{27}, A_{28}, A_{29}, A_{30}$
- *Group 8* : $A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36}$
- *Group 9* : A_{37}, A_{38}, A_{39}

- *Group 10* : $A_{40}, A_{41}, A_{42}, A_{43}$
- *Group 11* : $A_{44}, A_{45}, A_{46}, A_{47}, A_{48}, A_{49}$
- *Group 12* : A_{50}, A_{51}, A_{52}

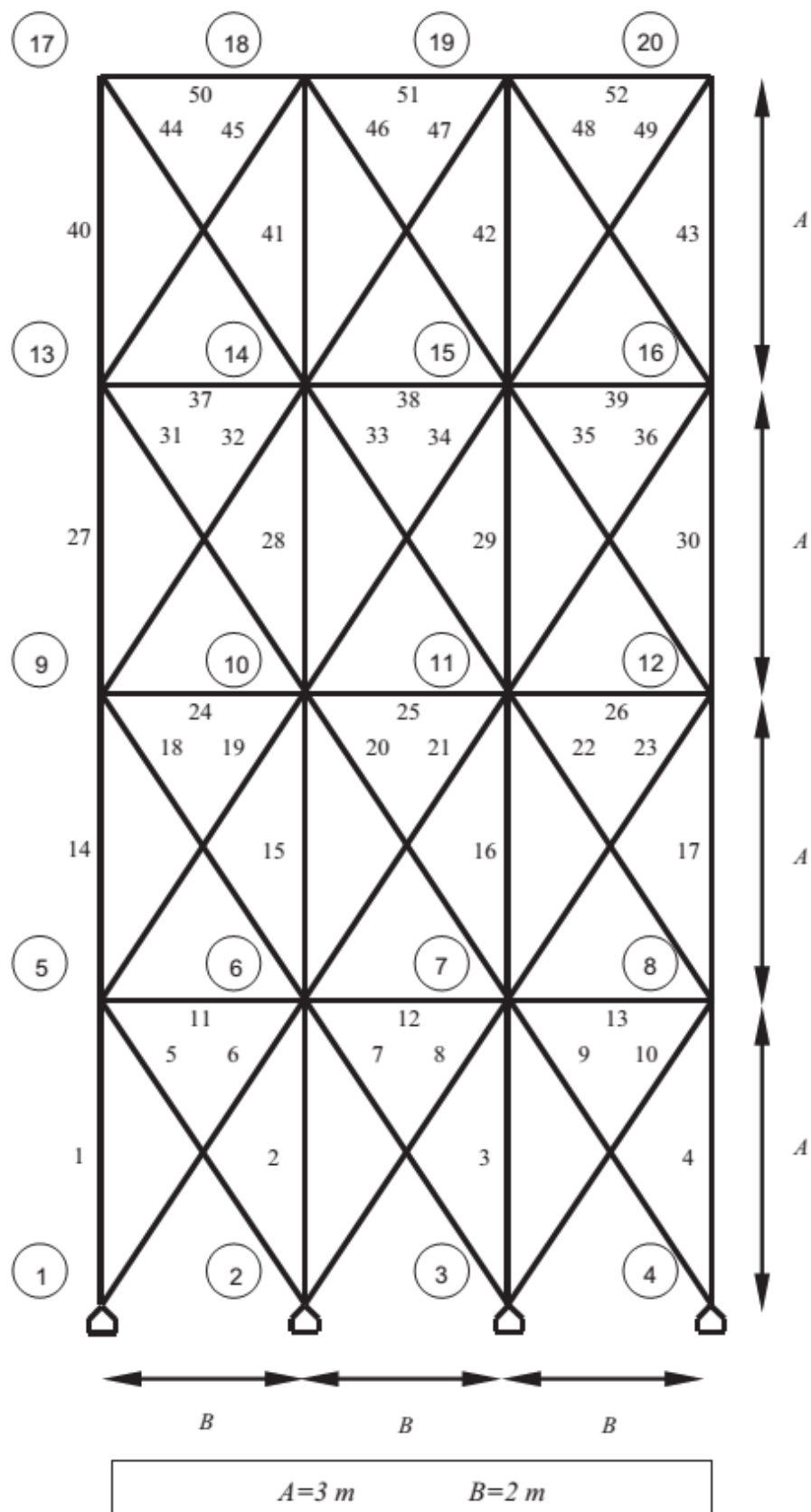


Fig. 14. Structure of a 52-bar truss.

Table 18

Available cross-section areas of the AISC norm (valid values for the parameters).

No.	in. ²	mm ²	No.	in. ²	mm ²
1	0.111	71.613	33	3.84	2477.414
2	0.141	90.968	34	3.87	2496.769
3	0.196	126.451	35	3.88	2503.221
4	0.25	161.29	36	4.18	2696.769
5	0.307	198.064	37	4.22	2722.575
6	0.391	252.258	38	4.49	2896.768
7	0.442	285.161	39	4.59	2961.284
8	0.563	363.225	40	4.8	3096.768
9	0.602	388.386	41	4.97	3206.445
10	0.766	494.193	42	5.12	3303.219
11	0.785	506.451	43	5.74	3703.218
12	0.994	641.289	44	7.22	4658.055
13	1	645.16	45	7.97	5141.925
14	1.228	792.256	46	8.53	5503.215
15	1.266	816.773	47	9.3	5999.988
16	1.457	939.998	48	10.85	6999.986
17	1.563	1008.385	49	11.5	7419.34
18	1.62	1045.159	50	13.5	8709.66
19	1.8	1161.288	51	13.9	8967.724
20	1.99	1283.868	52	14.2	9161.272
21	2.13	1374.191	53	15.5	9999.98
22	2.38	1535.481	54	16	10322.56
23	2.62	1690.319	55	16.9	10903.2
24	2.63	1696.771	56	18.8	12129.01
25	2.88	1858.061	57	19.9	12838.68
26	2.93	1890.319	58	22	14193.52
27	3.09	1993.544	59	22.9	14774.16
28	3.13	2019.351	60	24.5	15806.42
29	3.38	2180.641	61	26.5	17096.74
30	3.47	2238.705	62	28	18064.48
31	3.55	2290.318	63	30	19354.8
32	3.63	2341.931	64	33.5	21612.86

بنابراین، این مسئله 12 پارامتر دارد که باید بهینه شوند. دیگر فرضیات مسئله عبارتند از :

- $\rho = 7860.0 \text{ kg/m}^3$
- $E = 2.07e5 \text{ MPa}$
- Stress limitation = 180 MPa
- Maximum stress = 179.7652 MPa
- Design variable set is chosen from **Table 18**
- $P_k = 100 \text{ kN}$, $P_y = 200 \text{ kN}$

مجدداً ما این مسئله را با 30 عامل جستجو و در طول 500 تکرار حل کردیم. مشابه با طراحی خرابای 15 میله ای، عوامل جستجوی WOA در طول بهینه سازی به سادگی به نزدیکترین مقدار صحیح گرد می شوند. نتایج در جدول 19 خلاصه و با چندین الگوریتم مقایسه شده اند.

بر طبق جدول 19، و با بررسی نتایج موجود در این جدول، بهترین وزن بهینه ی بدست آمده توسط WOA 1902.605 بوده که همانند وزن بهینه ی روش های SOS و MBA است.

Table 19
Comparison of WOA optimization results with literature for the 52-bar truss design problem.

Variables (mm ²)	PSO [95]	PSOPC [95]	HPSO [95]	DHPSACO [109]	MBA [96]	SOS [98]	WOA
A1-A4	4658.055	5999.988	4658.055	4658.055	4658.055	4658.055	4658.055
A5-A10	1374.19	1008.38	1161.288	1161.288	1161.288	1161.288	1161.288
A11-A13	1858.06	2696.77	363.225	494.193	494.193	494.193	494.193
A14-A17	3206.44	3206.44	3303.219	3303.219	3303.219	3303.219	3303.219
A18-A23	1283.87	1161.29	940	1008.385	940	940	940
A24-A26	252.26	729.03	494.193	285.161	494.193	494.193	494.193
A27-A30	3303.22	2238.71	2238.705	2290.318	2238.705	2238.705	2238.705
A31-A36	1045.16	1008.38	1008.385	1008.385	1008.385	1008.385	1008.385
A37-A39	126.45	494.19	388.386	388.386	494.193	494.193	494.193
A40-A43	2341.93	1283.87	1283.868	1283.868	1283.868	1283.868	1283.868
A44-A49	1008.38	1161.29	1161.288	1161.288	1161.288	1161.288	1161.288
A50-A52	1045.16	494.19	792.256	506.451	494.193	494.193	494.193
Optimal weight (kg)	2230.16	2146.63	1905.495	1904.83	1902.605	1902.605	1902.605
No. of analyses	150,000	150,000	5300	11,100	5450	2350	2250

همچنین، WOA در طی حل این مسئله، به حداقل تعداد ارزیابی های تابع نیازمند است. از نتایج مشهود است که WOA، SOS، و MBA به صورت چشمگیری عملکرد بهتری را نسبت به PSO، PSOPC، HPSO و DHPSACO از خود نشان داده اند.

از آنجایی که الگوریتم WOA الگوی بهینه سازی جدیدی می باشد، مطالعه ی پارامترهای داخلی اصلی این الگوریتم نظیر تعداد عوامل جستجو، تعداد حداکثر تکرار و بردار a در معادله ی 2.3، یقیناً به محققینی که در راه اجرای این الگوریتم برای حل مسائل متفاوت قدم برداشته اند کمک می رساند. بنابراین، ما مسئله ی طراحی خرپایه 52 میله ای با تغییر این پارامتر به صورت زیر حل می نماییم :

- تعداد عوامل جستجو (n): 5, 10, 20, 40, or 80, 100
- تعداد حداکثر تکرار (t): 50, 100, 500, or 1000
- کاهش می یابد from 1 to 0, 2 to 0, or 4 to 0 به صورت خطی: α بردار

برای اینکه بتوانیم اثر این پارامترهای را بر روی کارایی الگوریتم WOA بررسی کنیم، 3 آزمایش مستقل به وسیله ی تغییر همزمان n ، t و a و α انجام می شود. مسئله ی خرپایه ی 52 میله ای 54 بار $(6*4+6*3+4*3)$ به وسیله ی WOA و با مقادیر متفاوت t ، n و a حل می شود. برای هر WOA با پارامترهای متفاوت، ما آن را به تعداد 10 مرتبه بر روی مسئله اجرا می کنیم تا بتوانیم وزن میانگین را محاسبه کرده و قابلیت اطمینان نتایج را با یکدیگر مقایسه نماییم. پس از طی تمام این مراحل، نتایج در شکل 15 به نمایش در آمده اند.

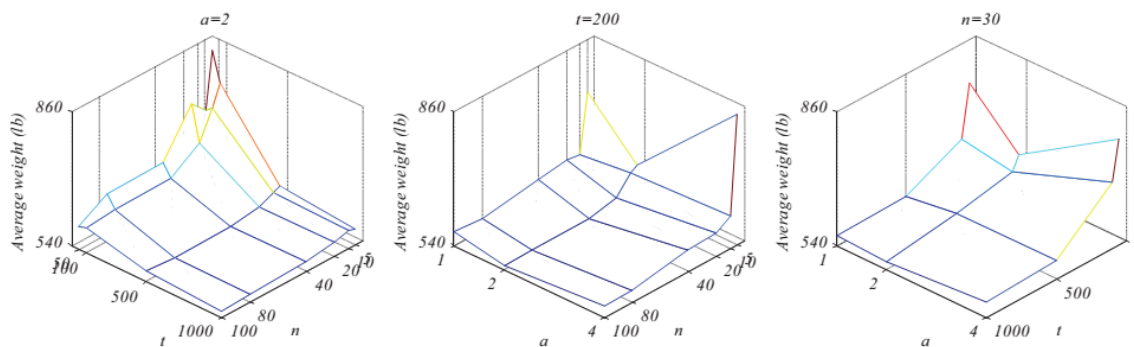


Fig. 15. Average weight obtained by WOA over 10 runs for the 52-bar truss varying n , t , and a .

این شکل نشان می دهد که بهترین مقادیر برای n و t به ترتیب برابر 100 و 1000 می باشد. این نتایج منطقی اند، چرا که با افزایش تعداد عوامل جستجو و همچنین حداکثر تکرار ها، تقریب بهتری از میزان بهینه ی سراسری بدست می آید. برای پارامتر a ، بنظر می رسد که کاهش خطی از 2 به 0 منجر به نتایج بهتری می شود. مقادیر دیگر این پارامتر نتایج بدی می دهند، زیرا آنها صرفاً-کمتر بر روی استخراج و اکتشاف تاکید دارند.

به عنوان خلاصه، نتایج مسائل طراحی ساختاری نشان می دهند که الگوریتم WOA پیشنهاد شده توان این را دارد تا در حل مسائل واقعی حتی با فضای جستجوی ناشناخته نیز موثر باشد.

5. نتیجه گیری

این مطالعه، الگوریتم بهینه سازی جدیدی را بر اساس ازدحام و با الهام از رفتار شکار نهنگ های گوژپشت ارائه داده است. روش پیشنهاد شده (به نام WOA، الگوریتم بهینه سازی نهنگ) شامل 3 عملگر برای شبیه سازی جستجوی طعمه، محاصره طعمه و رفتار تغذیه حباب-تور نهنگ های گوژپشت می باشد. مطالعه ای گسترده با استفاده از 29 تابع محک ریاضی، اکتشاف، استخراج، اجتناب از مقدار بهینه محلی و رفتار همگرایی الگوریتم ارائه شده مورد تحلیل و بررسی قرار گرفت. این چنین دریافت شد که الگوریتم WOA به تنهایی قابلیت رقابت با روش های فوق ابتکاری مدرن را دارد.

جهت جمع آوری اطلاعات بیشتری، 6 مسئله ی مهندسی ساختاری (یعنی فتر فشرده/بازشده، طراحی اتصال میله، طراحی مخزن تحت فشار و طراحی خرپایه ی 15، 25 و 52 پایه) حل شدند. WOA با بهینه سازی فوق ابتکاری، بخوبی توانایی رقابت داشته و همچنین از روش های مرسوم بالاتر است.

نسخه های باینری و چند-هدفی الگوریتم WOA، به ترتیب به نام های الگوریتم بهینه سازی نهنگ باینری و الگوریتم بهینه سازی نهنگ چند-هدفی در حال حاضر تحت توسعه می باشند.

سپاسگزاری

مفتخریم تا از انتشارات پروفیسور ا.کاوه به عنوان یکی از پیشگامان این زمینه ی بهینه سازی آماری تشکر کنیم، این آثار همواره مشوق ما بوده اند.