مجله مهندسی مکانیک مدرس، مرداد 1394، دوره 15، شماره 5، ص‌ص 319-328

ماهنامه علمی پژوهشی

مهندسی مکانیک مدرس

mme.modares.ac.ir

بهینه‌سازی فرایند تطابق چندین ابر نقاط با استفاده از الگوریتم جدید ترکیبی جستجوی گرانشی و نلدر- مید

محمد خسروی[1]، خلیل خلیلی[2*]، حسین امیرآبادی[3]

1- دانشجوی دکترا، مهندسی مکانیک، دانشگاه بیرجند، بیرجند

2- دانشیار، مهندسی مکانیک، دانشگاه بیرجند، بیرجند

3- استادیار، مهندسی مکانیک، دانشگاه بیرجند، بیرجند

* بیرجند، صندوق پستی 97175/615، kkhalili@birjand.ac.ir

اطلاعات مقاله

مقاله پژوهشی کامل

دریافت: 11 دی 1393

پذیرش: 12 اسفند 1393

ارائه در سایت: 19 فروردین 1394

کلید واژگان:

تطابق

بهینه‌سازی

الگوریتم ترکیبی

الگوریتم جستجوی گرانشی

چکیده

بهینه‌سازی در بسیاری از شاخه‌های علوم کاربرد زیادی یافته است. در سال‌های اخیر، تئوری‌ها و روش‌های متعددی برای پیدا کردن پاسخ بهینه و توسعه روش‌های بهینه‌سازی ارائه شده است. الگوریتم‌های بهینه‌سازی الهام گرفته از طبیعت به عنوان روش‌های ابتکاری، راه‌حل‌های مناسبی را برای مسائل پیچیده ارائه می‌کند که یکی از کاربردهای این روش‌ها، مهندسی معکوس است. در مهندسی معکوس، هر مجموعه از نقاط اسکن شده نسبت به یک دستگاه خاص تعریف می‌شوند. در فرایند انطباق داده‌ها، مجموعه‌های این نقاط به صورت مجزا، با هم ترکیب شده و نسبت به یک سیستم مختصات واحد قرار می‌گیرند که این فرایند،تطابق ابر نقاط نامیده می‌شود. در این پژوهش، قطعه کاربردی در آزمایش تجربی توسط ماشین اندازه‌گیری مختصات در دو وضعیت بدون انتقال و انتقال یافته، اندازه‌برداری شده و فرایند تطابق ابر نقاط بر روی آن پیاده‌سازی شده است. به کمک الگوریتم‌های جستجوی گرانشی، جمعیت ذرات و ژنتیک، فرایند تطابق بهینه‌سازی شده و پارامترهای تطابق (چرخش و جابه‌جایی) به‌دست آمده است. از بین الگوریتم‌های ذکر شده، الگوریتم جستجوی گرانشی از دقت جابه‌جایی، دقت چرخشی و مقدار همگرایی بهتر و هم‌چنین زمان اجرای کمتری برخوردار بوده است. در نهایت الگوریتم ترکیبی ارائه شده است که ترکیبی از الگوریتم‌های جستجوی گرانشی و نلدر- مید می‌باشد. در الگوریتم پیشنهادی، مقادیر حدس اولیه توسط الگوریتم گرانشی به‌دست آمده و در اختیار الگوریتم نلدر- مید قرار می‌گیرد تا پاسخ دقیق به‌دست آید. الگوریتم پیشنهادی در این پژوهش، ازنظر تعداد تکرار و مقدار همگرایی بر الگوریتم‌های جستجوی گرانشی و نلدر- مید برتری دارد.

Optimization of point clouds sets registration process using a hybrid algorithm of gravitational search and nelder-mead

Mohammad Khosravi, Khalil Khalili*, Hossein Amirabadi

Department of Mechanical Engineering, University of Birjand, Birjand, Iran

* P.O.B. 97175/615 Birjand, Iran, kkhalili@birjand.ac.ir

ARTICLE INFORMATION

Original Research Paper
Received 01 January 2015
Accepted 03 March 2015
Available Online 08 April 2015

Keywords:
Registration
Optimization
Hybrid Algorithm
Gravitational Search Algorithm

ABSTRACT

Optimization has found widespread application in many branches of science. In recent years, different methods and theories have been developed to find optimal solutions. Optimization algorithms are inspired by nature as heuristics solutions to complex problems. Reverse engineering is one of the applications of optimization methods. In reverse engineering a set of scan points are defined relative to a particular coordination. In data registration process, the scanned data sets that separate and combine to a single coordinate system are called the process of registration. In this research, the applications part has been digitized by coordinate measuring machine(CMM) and the process of point clouds registration in experimental on two pieces in position (without translation and with translation case) has been implemented. Using gravitational search algorithm (GSA), particle swarm optimization (PSO) and genetic algorithm (GA) optimization process is optimized and the registration parameters (rotation and displacement) are obtained. The algorithms mentioned, GSA, the accuracy displacement, rotational accuracy and better convergence rate and the run time is less. Finally, a hybrid algorithm is proposed that is a combination of GSA, and Nelder-Mead algorithms (GSA-NM). In the proposed algorithm, the initial guess values obtained by GSA and Nelder-Mead algorithm are provided to ensure an accurate response. The proposed hybrid algorithm is superior to GSA and Nelder-Mead, in terms of the number of iterations and the amount of convergence.

1- مقدمه

در بسیاری از مسائل مهندسی، فرایند بهینه‌سازی را می‌توان به‌عنوان جزء جدانشدنی مسأله در نظر گرفت. فرایند بهینه‌سازی، به‌دست آوردن بهترین نتیجه در شرایط داده شده است و به تعبیر ریاضی هدف از بهینه‌سازی یک

برای ارجاع به این مقاله از عبارت ذیل استفاده نمایید:

Please cite this article using:

M. Khosravi, Kh. Khalili, H. Amirabadi, Optimization of point clouds sets registration process using a hybrid algorithm of gravitational search and nelder-mead, *Modares Mechanical Engineering*, Vol. 15, No. 5, pp. 319-328, 2015 (In Persian)

[Downloaded from mme.modares.ac.ir on 2024-04-18]

[DOR: 20.1001.1.10275940.1394.15.5.44.8]

تابع، پیدا کردن کمینه یا بیشینه آن تابع است. در بهینه‌سازی، پیدا کردن بهترین جواب برای یک مسأله مطرح می‌شود و لغت بهترین جواب، نشان می‌دهد که مسأله بیش از یک جواب دارد. بهترین جواب برای مسأله، به روش حل و میزان دقت آن ارتباط دارد و در مسائل بهینه‌سازی، با تغییر دادن متغیرهای ورودی، خصوصیات یک دستگاه، فرایند ریاضی و یا آزمایش تجربی می‌توان بهترین خروجی را به‌دست آورد. تابع مورد بررسی؛ تابع هدف، تابع هزینه و یا تابع برازندگی نامیده می‌شود [1].

در سال‌های اخیر برای پیدا کردن پاسخ بهینه در مسائل مهندسی، الگوریتم‌های متنوع و کارآمدی ارائه شده است. در حالت کلی، الگوی تابع بهینه‌سازی به‌صورت رابطه (1) تعریف می‌شود:

$$f(x) = f(x_1, x_2, \dots, x_n) \qquad (1)$$

پیداکردن مقادیری از متغیرهای x، بطوری که $f(x)$ را کمینه یا بیشینه کند، بهینه‌سازی نامیده می‌شود. در بهینه‌سازی می‌توان از روش‌های گوناگونی برای حل مسأله استفاده کرد. این روش‌ها را می‌توان به دو گروه عمده تقسیم‌بندی کرد [2]:

الف) روش حل تحلیلی

ب) روش حل ابتکاری[1]

در روش اول، در صورتی‌که تابع چند متغیره باشد از مشتق‌های تابع هدف و قیدها برای به‌دست آوردن مقدار بهینه استفاده می‌شود. به‌طورکلی، در روش‌های تحلیلی همواره فرض می‌شود که تابع محدب[2] است و دارای مقدار کمینه و پاسخ بهینه است. ولی در بسیاری از مسائل مهندسی، توابعی وجود دارند که محدب نیستند یا ممکن است ناپیوسته باشند؛ ازاین‌رو باید از روش‌های دیگر، مانند روش‌های ابتکاری استفاده کرد. هم‌چنین استفاده از روش‌های تحلیلی که بر مبنای روابط ریاضی و تجربی استوار است، اغلب منجر به جواب‌های بهینه نمی‌شود و کمینه‌های محلی[3] پاسخ نهایی آن‌ها هستند. دلیل آن عدم توانایی این روش‌ها، درحرکت از چند جهت به‌سوی جواب بهینه است و از آنجا که اغلب توابع دارای چند نقطه کمینه می‌باشند، این روش‌ها کارایی لازم را نخواهند داشت. بعلاوه این روش‌ها به حدس اولیه وابسته بوده و برای مسائل خاص کاربرد دارند. بنابراین در الگوریتم‌های بهینه‌سازی، تولید تصادفی جمعیت اولیه می‌تواند احتمال همگرایی را افزایش دهد [3].

با توجه به مطالب ذکر شده، روش‌های تحلیلی برای حل مسائل بهینه‌سازی همیشه دارای محدودیت‌هایی می‌باشند، لذا الگوریتم‌های بهینه‌سازی ابتکاری در سال‌های اخیر به دفعات برای بهینه‌سازی مسائل مختلف مهندسی مورد استفاده قرار گرفته‌اند. از این رو استفاده از این روش‌ها در مسائل مهندسی، باعث سادگی و سرعت بخشیدن به حل این‌گونه مسائل شده است [4].

الگوریتم‌های بهینه‌سازی الهام گرفته از طبیعت به عنوان روش‌های ابتکاری، در کنار روش‌های کلاسیک موفقیت خوبی را کسب نموده‌اند. از جمله این روش‌ها می‌توان به الگوریتم ژنتیک[4] با الهام از علم وراثت و تکامل، الگوریتم بهینه‌سازی حرکت گروهی ذرات[5] با تقلید از رفتار اجتماعی پرندگان و ماهی‌ها، الگوریتم رقابت استعماری[6] با الگوبرداری از پدیده اجتماعی- انسانی، الگوریتم مورچگان[7] با شبیه‌سازی رفتار مورچگان در یافتن کوتاه‌ترین مسیر غذا و خانه، الگوریتم گرانشی[8] با الهام از قوانین گرانش و حرکت نیوتن، الگوریتم جستجوی باکتریایی[9] با تقلید از رفتار باکتری‌ها برای جستجوی غذا و الگوریتم تبرید فلزات[10] با الهام از مشاهدات ترمودینامیک اشاره نمود.

یکی از کاربردهای روش‌های بهینه‌سازی ابتکاری، مهندسی معکوس است. در مهندسی معکوس، تعیین تبدیل[11] هندسی که یک ابر نقاط را با ابر نقاط دیگر در یک سیستم مختصات مطلق هم راستا می‌کند، فرایند تطابق دو ابر نقاط نامیده می‌شود [5]. برای فرایند تطابق، الگوریتم‌های مختلفی وجود دارد و یکی از روش‌های معمول که برای تطابق دو یا چند مجموعه نقاط مورد استفاده قرار می‌گیرد، الگوریتم تکراری نزدیک‌ترین نقطه[12] می‌باشد [6]. این الگوریتم توسط بسل و مک‌کی [7] و چن و مدیونی [8] ارائه شده است. در این روش، ابتدا دو مجموعه نقاط به صورت مدل و داده وارد الگوریتم شده و ماتریس تبدیل شامل ماتریس چرخش و بردار انتقال به‌دست می‌آید. تابع هدف این روش در رابطه (2) آورده شده است:

$$E(R,t) = \frac{1}{N}\sum_{i=1}^{N}\|m_i - (Rd_i + t)\|^2 \qquad (2)$$

در رابطه (2) مقدار N تعداد کل نقاط برای عملیات تطابق، m_i بردار هر نقطه مدل، d_i بردار هر نقطه داده، R ماتریس چرخش و t بردار جابه‌جایی دو مجموعه نقاط می‌باشند.

1-1- الگوریتم ژنتیک

الگوریتم ژنتیک به عنوان یکی از روش‌های تصادفی بهینه‌سازی توسط جان هالند [9] در سال 1967 ابداع شد. این روش بر مبنای نظریه تکامل تدریجی و ایده‌های بنیادین داروین پایه‌گذاری شده است. الگوریتم ژنتیک، اصول تنازع بقا را در بهترین جواب مسأله به کار می‌گیرد. در الگوریتم‌های تکاملی تعداد جمعیت برای شروع عملیات جستجو، بستگی به سیستم مورد نظر و قابلیت محاسباتی آن دارد و قابل تغییر است. برازندگی هر فرد نیز یک عدد حقیقی است که بر اساس تابع هدف برای هر فرد، جداگانه محاسبه شده و به عنوان معیاری برای تعیین مناسب یا نامناسب بودن آن فرد در نظر گرفته می‌شود. در هر نسل مجموعه جدیدی از تقریب‌ها، بر اساس میزان برازش[13] آن‌ها در دامنه مسأله و تکثیر با عملگرهای گرفته شده از ژنتیک طبیعی ساخته می‌شود. این فرایند در نهایت به تکامل جمعیتی از اعضا ختم می‌شود که نسبت به اعضای اولیه با محیط، سازگاری بهتری دارند. محققان زیادی در خصوص فرایند تطابق، از الگوریتم ژنتیک استفاده کرده‌اند که به برخی از آن‌ها در زیر اشاره شده است.

خسروی و همکاران [10] به کمک روش‌های نلدر- مید، تجزیه مقادیر منفرد و الگوریتم ژنتیک فرایند تطابق ابر نقاط را پیاده‌سازی کردند و با کمینه کردن تابع هدف (معادله (2))، بهترین پارامترهای تطابق (چرخش و جابه‌جایی) را به‌دست آوردند. از بین این روش‌ها نلدر- مید دارای بهترین دقت جابه‌جایی و روش تجزیه مقادیر منفرد دارای بهترین دقت چرخش می‌باشند. الگوریتم ژنتیک با مقدار جمعیت اولیه 100 و تعداد نسل 500، دقت کمتری نسبت به روش‌های نلدر- مید و تجزیه مقادیر منفرد از خود نشان داد.

شینک و هانکه [11] از اسکنر لیزری برای برداشت ابر نقاط استفاده کردند. آن‌ها فرایند تطابق ابر نقاط را به کمک الگوریتم تکراری نزدیک‌ترین

1- Heuristic Methods
2- Convex
3- Local Optima
4- Genetic Algorithm
5- Particle Swarm Optimization Algorithms
6- Imperialist Competitive Algorithm
7- Ant Colony Optimization
8- Gravitational Search Algorithm
9- Bacterial Foraging Optimization
10- Simulated Annealing
11- Transformation
12- Iterative Closest Point(ICP)
13- Fitness

[Downloaded from mme.modares.ac.ir on 2024-04-18]

[DOR: 20.1001.1.10275940.1394.15.5.44.8]

نقطه و الگوریتم ژنتیک پیاده‌سازی کردند و به فرایند تطابق کاملاً خودکار[1]دست یافتند. نتایج الگوریتم به‌دست آمده در مقایسه با روش تطابق کلاسیک، از دقت و کارآمدی بیشتری برخوردار بود.

سالومون و همکاران [12] برای تطابق تصاویر سه بعدی، از الگوریتم تکاملی تفاضلی استفاده کردند. در این الگوریتم از مقادیر حقیقی، استفاده شده که در مرحله تولید مثل با الگوریتم ژنتیک تفاوت داشته است. الگوریتم به کار گرفته شده دارای محاسبات زیادی بوده و برای تطابق حجمی که از قبل به‌صورت غیردقیق هم راستا شده‌اند، مورد استفاده قرار گرفت.

روبرتسون و فیشر [13] برای تطابق داده‌ها، الگوریتم ژنتیک را مورد استفاده قرار داده و از خطای میانگین مربعات[2] به عنوان تابع هدف بهره گرفتند. در فرایند تطابق، شش پارامتر تطابق (سه زاویه و سه جابه‌جایی) را بدون در نظر گرفتن هم‌پوشانی دو داده، بهینه‌سازی کردند.

سیلوا و همکاران [14] برای تطابق دقیق چندین تصویر با هم‌پوشانی کم، از الگوریتم ژنتیک استفاده کردند. روش جدید، خطای هم‌راستایی تصاویر را کمینه می‌کند. الگوریتم ژنتیک پیشنهادی، توانایی تطابق سطوح را بدون نیاز به پیش هم‌راستایی دارد. نتایج آزمایش، دقت روش پیشنهادی را برای تطابق تأیید می‌کند.

چاو و همکاران [15] الگوریتم ژنتیک بهبود یافته‌ای را برای تطابق سطوح آزاد ارائه دادند که 1000 بار سریع‌تر از الگوریتم ژنتیک سنتی عمل می‌کرد. فرایند تطابق، بر روی تصاویر از دو نمای مختلف پیاده‌سازی شد. دقت، سرعت و توانایی الگوریتم پیشنهادی، به وسیله آزمایش‌های تجربی واقعی مورد آزمایش قرار گرفت و زمان اجرای برنامه و همگرایی فرایند، در الگوریتم ژنتیک پیشنهادی مقایسه شده است.

2-1- الگوریتم جمعیت ذرات

الگوریتم تجمع ذرات، یک روش بهینه‌سازی تکاملی است که از رفتار اجتماعی پرندگان الهام گرفته شده است. این الگوریتم توسط جیمز کندی و راسل ابرهارت [16] در سال 1995 ارائه شد. در این روش همانند الگوریتم ژنتیک، در ابتدا یک جمعیت اولیه (تعدادی ذرات) به صورت تصادفی در فضای جستجو انتخاب و با جایگذاری در تابع هدف، مقدار جواب محاسبه می‌شود. سپس با به روز رسانی نسل‌ها، جواب بهینه در فضای مسأله به‌دست می‌آید. دراین روش هر ذره با مقدار x_i و v_i که به ترتیب موقعیت و سرعت مربوط به ذره iام است تعریف شده و مقادیر آن‌ها طبق روابط (3) و (4) به روز رسانی می‌شود.

$$v_{id}(t+1) = C_1\, v_{id}(t) + C_2 R_2\big(p_{id} - x_{id}(t)\big) + C_3 R_3\big(p_{gd} - x_{id}(t)\big) \quad (3)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (4)$$

در روابط بالا C_1، C_2 و C_3 ضرایب ثابت، R_2 و R_3 اعداد تصادفی با توزیع یکنواخت در بازه [0,1]، p_{gd} بهترین جواب در بین کل ذرات، و p_{id} بهترین جواب برای هر ذره است.

یو و وانگ [17] از الگوریتم ترکیبی جمعیت ذرات و الگوریتم تکراری نزدیک‌ترین نقطه برای فرایند تطابق دقیق، استفاده نمودند. آن‌ها مدعی شدند که این الگوریتم در فرایند تطابق دقیق، از سرعت همگرایی خوبی برخوردار می‌باشد. نتایج آزمایش‌های تجربی آن‌ها، نشان داد که الگوریتم ترکیبی جمعیت ذرات و الگوریتم تکراری نزدیک‌ترین نقطه، دارای توانایی و تطبیق بالا در پیاده‌سازی فرایند تطابق است.

3-1- الگوریتم جستجوی گرانشی

یکی از نیروهای شناخته شده در جهان، نیروی گرانش است که عالم را دور هم نگه می‌دارد [18]. وجود نیروی گرانش توسط نیوتن کشف شد. نیوتن قوانینی وضع نمود که قوانین اساسی در فیزیک هستند. یکی از این قوانین، قانون دوم نیوتن است. در فیزیک برای هر جسم سه نوع جرم قابل تعریف است. جرم گرانش فعال، جرم گرانش غیرفعال و جرم اینرسی که مقدار آن‌ها برای یک جسم با هم برابرند. با در نظر گرفتن دو جسم، نیروی گرانشی به مقدار F_{12} از جانب جسم 2 بر جسم 1 وارد شده و مقدار آن از رابطه (5) محاسبه می‌شود.

$$F_{12} = G\frac{M_{a2}M_{p1}}{R^2} \quad (5)$$

M_{p1} و M_{a2} به ترتیب جرم گرانشی غیرفعال جرم اول و جرم گرانشی فعال جرم دوم، G ثابت گرانش و R فاصله بین دو جرم می‌باشد.

درصورتی‌که جسم 1 تحت تأثیر نیروی جاذبه 2 شتابی برابر a_1 بگیرد، رابطه (6) بدست می‌آید.

$$a_1 = \frac{F_{12}}{M_{i1}} \quad (6)$$

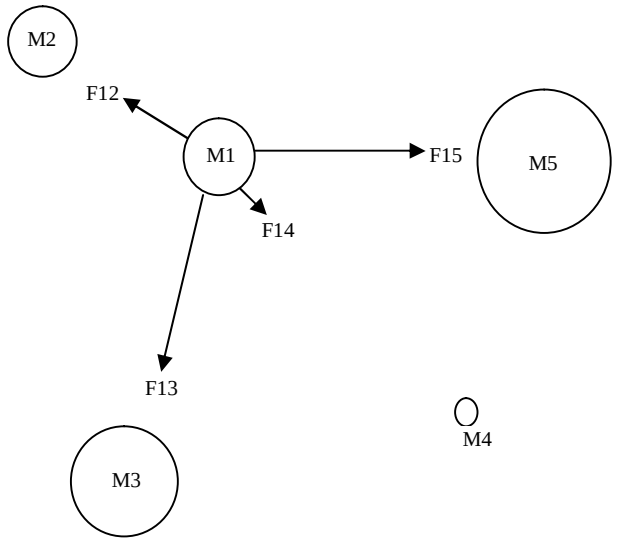
در این رابطه M_{i1} جرم اینرسی جسم اول است.

در یک سیستم شامل چند جسم، بر هر جسم از اجسام دیگر نیروی گرانشی وارد می‌شود که در نهایت جسم مورد نظر، در جهت برایند نیروها شتاب می‌گیرد و در شکل (1) نشان داده شده است.

در الگوریتم جستجوی گرانشی [19] با پیاده‌سازی قوانین گرانش و حرکت نیوتن در یک سیستم مصنوعی، پاسخ بهینه از فضای جستجو به‌دست می‌آید. الگوریتم جستجوی گرانشی شامل دو مرحله کلی زیر می‌باشد که در ادامه شرح داده شده است.

الف) تشکیل سیستم مصنوعی با زمان گسسته در محیط مسأله، موقعیت‌یابی اولیه جرم‌ها، وضع قوانین و تنظیم پارامترها؛

ب) گذر زمان، حرکت جرم‌ها و درنهایت به روز رسانی پارامترها تا شرط توقف.



شکل 1 تأثیر نیروی گرانشی بر هر جسم از طرف اجسام دیگر

1- GAReg-ISF (Genetic Algorithm Registration with Imperfect and Subdivided Features)
2- Mean Square Error (MSE)

[Downloaded from mme.modares.ac.ir on 2024-04-18]
[DOR: 20.1001.1.10275940.1394.15.5.44.8]

1-3-1- تشکیل سیستم، وضع قوانین و تنظیم پارامترها

در مرحله اول، فضای مسأله شامل یک دستگاه مختصات چند بعدی تعریف می‌شود. هر نقطه از فضا، جواب مسأله و عامل‌های جستجو، مجموعه‌ای از جرم‌ها هستند. مشخصه‌های هر جرم شامل موقعیت جرم، جرم گرانشی فعال، جرم گرانشی غیرفعال و جرم اینرسی با توجه به برازندگی عامل‌ها تعیین می‌شوند.

در مرحله دوم، قوانین حاکم بر مسأله یعنی قوانین گرانش و حرکت پیاده‌سازی می‌شوند. در قانون گرانش، هر جرم در سیستم مصنوعی تمام جرم‌ها را به سمت خود جذب می‌کند و در قانون حرکت، سرعت فعلی هر جرم با مجموع ضریبی از سرعت فعلی جرم و تغییر سرعت آن تعریف می‌شود.

در مرحله سوم، سیستم به‌صورت مجموعه‌ای از N جرم در n بعد در نظر گرفته شده که موقعیت جرم i در بعد d با x_i^d مطابق رابطه (7) بیان می‌شود.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \tag{7}$$

در لحظه شروع عامل، موقعیت‌یابی به‌صورت تصادفی انجام می‌شود.

در مرحله چهارم، مقدار نیروی وارد شده از جرم j به جرم i در زمان t و در جهت بعد d، با $F_{ij}^d(t)$ طبق رابطه (8) و مجموع نیروهای سایر جرم‌ها به جرم i در زمان t و بعد d، با $F_i^d(t)$ از رابطه (9) به‌دست می‌آید.

$$F_{ij}^d = G(t)\,\frac{M_{pi}(t)\,M_{aj}(t)}{R_{ij}(t)+\varepsilon}\,(x_j^d(t)-x_i^d(t)) \tag{8}$$

$$F_i^d = \sum_{j=1,\ j\neq i}^{N} \text{rand}_j\, F_{ij}^d(t) \tag{9}$$

در این رابطه مقدار $G(t)$، $R_{ij}(t)$، $M_{pi}(t)$، $M_{aj}(t)$ و ε به ترتیب ثابت گرانش در زمان t، فاصله بین دو جرم i و j، جرم غیرفعال i، جرم فعال j و یک مقدار بسیار کوچک می‌باشد. در رابطه (10) rand_j یک عدد تصادفی با توزیع یکنواخت، در بازه [0,1] می‌باشد و مقدار $R_{ij}(t)$ از رابطه (10) به‌دست می‌آید.

$$R_{ij}(t) = \left\|X_i(t), X_j(t)\right\|_2 \tag{10}$$

در این رابطه $X_i(t)$ و $X_j(t)$ موقعیت جرم‌ها می‌باشند.

در رابطه (9) مقدار ثابت گرانش، پارامتری مناسب برای کنترل توانایی جستجو و بهره‌وری الگوریتم به شمار می‌آید. مقدار مناسب برای ثابت گرانش، از یک مقدار اولیه بزرگ شروع شده و با گذشت زمان مقدار آن کاهش می‌یابد، در نتیجه مقدار پیشنهادی آن مطابق رابطه (11) می‌باشد.

$$G(t) = G_0 e^{-\alpha\frac{t}{T}} \tag{11}$$

در این رابطه α و G_0 ضرایب مثبت ثابت و T تعداد کل تکرار می‌باشد.

طبق قانون دوم نیوتن، مقدار شتاب جرم i در جهت بعد d و زمان t که با $a_i^d(t)$ نشان داده شده، مطابق رابطه (12) می‌باشد.

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{12}$$

در این رابطه $a_i^d(t)$، $F_i^d(t)$ و $M_{ii}(t)$ به ترتیب شتاب هر جرم در جهت بعد d، مجموع نیروی وارد شده بر هر جرم در جهت بعد d و جرم اینرسی عامل i می‌باشد. لازم به ذکر است که جرم‌های گرانشی فعال، گرانشی غیرفعال و اینرسی برای هر عامل جستجوگر مطابق رابطه (13)، برابر فرض می‌شود.

$$M_{il} = M_{al} = M_{pl} = M_l \qquad l=1,2,\dots,N \tag{13}$$

با ترکیب رابطه‌های (8)، (9) و (12) رابطه (14) به‌دست می‌آید.

$$a_i^d(t) = G(t)\sum_{j=1,\ j\neq i}^{N}\left[\text{rand}_j\,\frac{M_{aj}(t)}{R_{ij}(t)+\varepsilon}(x_j^d(t)-x_i^d(t))\right] \tag{14}$$

در مرحله پنجم، سرعت و موقعیت هر عامل در زمان بعدی و در بُعد d از روابط (15) و (16) محاسبه می‌شود.

$$v_i^d(t+1) = \text{rand}_i\ v_i^d(t) + a_i^d(t) \tag{15}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{16}$$

لازم به ذکر است با توجه به تابع هدف؛ تنظیم جرم عامل‌ها و نرمالیزه کردن آن‌ها، مطابق روابط (17) و (18) انجام می‌شود.

$$m_i(t) = \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} \tag{17}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{18}$$

در این روابط $\text{fit}_i(t)$ میزان برازندگی جرم i در زمان t، $\text{worst}(t)$ و $\text{best}(t)$ به ترتیب قوی‌ترین و ضعیف‌ترین شایستگی عامل جمعیت در زمان t بوده و مطابق روابط (19) و (20) به‌دست می‌آید.

$$\text{worst}(t) = \max_{j\in\{1,\dots,N\}} \text{fit}_j(t) \tag{19}$$

$$\text{best}(t) = \min_{j\in\{1,\dots,N\}} \text{fit}_j(t) \tag{20}$$

1-3-2- گذر زمان، حرکت جرم‌ها و به روز رسانی پارامترها

در ابتدا محیط سیستم تشکیل و پس از وضع قوانین، پارامترهای سیستم تنظیم می‌شود. هر عامل به صورت تصادفی، در یک نقطه از فضا قرار گرفته و در هر لحظه از زمان، عامل‌ها ارزیابی می‌شوند. تغییر مکان هر جرم که جواب‌های مسأله است محاسبه شده و در زمان بعد در موقعیت جدید قرار می‌گیرد و پارامترهای سیستم به روز رسانی می‌شوند. در نهایت الگوریتم می‌تواند با توجه به شرایط توقف پس از طی زمان ادامه یابد.

هدف از این پژوهش، بهینه‌سازی فرایند تطابق و کمینه کردن تابع برازندگی (رابطه (2)) به وسیله الگوریتم‌های جستجوی گرانشی، جمعیت ذرات و ژنتیک و در نهایت ارائه الگوریتم ترکیبی جدید جستجوی گرانشی و نلدر- مید است. الگوریتم ترکیبی، پارامترهای تطابق (چرخش و جابه‌جایی) را با دقت قابل قبول به‌دست آورده و در مقایسه با الگوریتم نلدر- مید از تعداد تکرار کمتری برخوردار است.

2- الگوریتم ترکیبی پیشنهادی

در سال‌های اخیر، علاقه به الگوریتم‌های ترکیبی ابتکاری به طور قابل توجهی در محدوده بهینه‌سازی افزایش یافته است. بهترین نتایج برای بسیاری از رویدادهای واقعی یا بهینه‌سازی کلاسیک، به وسیله الگوریتم‌های ترکیبی به‌دست می‌آید [20]. از آنجا که هر الگوریتم ابتکاری، یک راهبرد جستجوی متمایز دارد و ترکیب عملگرها از الگوریتم‌های مختلف، ممکن است به یک الگوریتم جدید با عملکرد بهتری نتیجه دهد، این فرایند الگوریتم ابتکاری ترکیبی نامیده می‌شود [21]. برخی از الگوریتم‌های ترکیبی ابتکاری در ادامه آمده است [22]:

الف) ترکیب الگوریتم‌های ابتکاری با یکدیگر

ب) ترکیب الگوریتم‌های ابتکاری با الگوریتم‌های بر پایه ریاضی که بیشتر در محدوده تحقیق در عملیات کاربرد دارد.

ج) ترکیب الگوریتم‌های ابتکاری با الگوریتم‌های مقید[1] که در حوزه هوش مصنوعی مورداستفاده قرار می‌گیرد.

د) ترکیب الگوریتم‌های ابتکاری با الگوریتم‌های یادگیری ماشین[2] و داده‌کاوی

محققان زیادی از الگوریتم ترکیبی استفاده کرده‌اند که به برخی از آن‌ها در زیر اشاره شده است.

لومونوسف و همکاران [23] برای تطابق دو مجموعه ابر نقاط، از روش دو

1- Constraint Programming
2- Machine Learning

[Downloaded from mme.modares.ac.ir on 2024-04-18]

[DOR: 20.1001.1.10275940.1394.15.5.44.8]

مرحله‌ای پیش تطابق[1] و تطابق نهایی استفاده کرده‌اند. در مرحله اول به کمک الگوریتم ژنتیک، پیش تطابق به‌صورت غیردقیق انجام شد. آن‌ها در مرحله دوم با استفاده از خروجی ژنتیک، تطابق دقیق را انجام دادند. برای تولید جمعیت اولیه در این روش، از مقادیر حقیقی بجای مقادیر باینری استفاده شده است که البته عملیات ژنتیک را کند می‌کند اما از روش باینری دقیق‌تر است. در این روش، بهینه‌سازی برای هفت پارامتر انجام شده که شش پارامتر مربوط به زوایا و جابه‌جایی بوده و پارامتر هفتم، هم‌پوشانی دو ابر نقاط می‌باشد. در نهایت آن‌ها به این نتیجه رسیدند که استفاده از الگوریتم ژنتیک به‌صورت منفرد، برای تطابق دقیق معقول نیست و دقت بیشتر تطابق با عملیات پیش تطابق و تطابق نهایی میسر خواهد بود.

منصور و همکاران [24] برای پیاده‌سازی فرایند تطابق ابر نقاط، از یک الگوریتم ترکیبی استفاده کردند. آن‌ها در الگوریتم پیشنهادی، از الگوریتم ژنتیک و الگوریتم شبه نیوتنی برای به‌دست آوردن شش پارامتر مجهول (چرخش و جابه‌جایی) استفاده کردند. در این الگوریتم با توجه به کاهش زمان محاسبات، یک سری نقاط خروجی یکنواخت حاصل از تطابق برای مدل نهایی به‌دست آمد. نتایج آزمایش در این الگوریتم نشان می‌دهد که الگوریتم خودکار پیشنهادی، از دقت بیشتری نسبت به الگوریتم‌های عادی برخوردار می‌باشد.

الگوریتم نلدر- مید، یک الگوریتم جستجوی غیرمرکب[2] است و با تکرار به جواب نهایی می‌رسد. در الگوریتم‌های تکرار، حدس اولیه مناسب به سرعت همگرایی و کاهش تعداد تکرار کمک زیادی می‌کند. از طرفی الگوریتم جستجوی گرانشی، الگوریتمی است که سریع به همگرایی می‌رسد و در واقع برای پیدا کردن بهترین جواب، محدوده‌ای از فضای حل مسأله را جستجو می‌نماید و در صورت ترکیب با الگوریتم نلدر- مید، شانس پیدا کردن حدس اولیه مناسب را افزایش می‌دهد.

الگوریتم ترکیبی ارائه شده مطابق شکل 2 از دو بخش تشکیل شده است. بخش اول الگوریتم جستجوی گرانشی و بخش دوم الگوریتم نلدر- مید می‌باشد که مراحل بهینه‌سازی مطابق شکل 2 نشان داده شده است.

الگوریتم ترکیبی مورد استفاده در این پژوهش قصد دارد برای یافتن حدس اولیه از مکانیزم الگوریتم جستجوی گرانشی بهره گرفته و با استفاده از جواب حاصل، پاسخ نهایی را از الگوریتم نلدر- مید به صورت دقیق به‌دست آورد.

3- توابع محک استاندارد

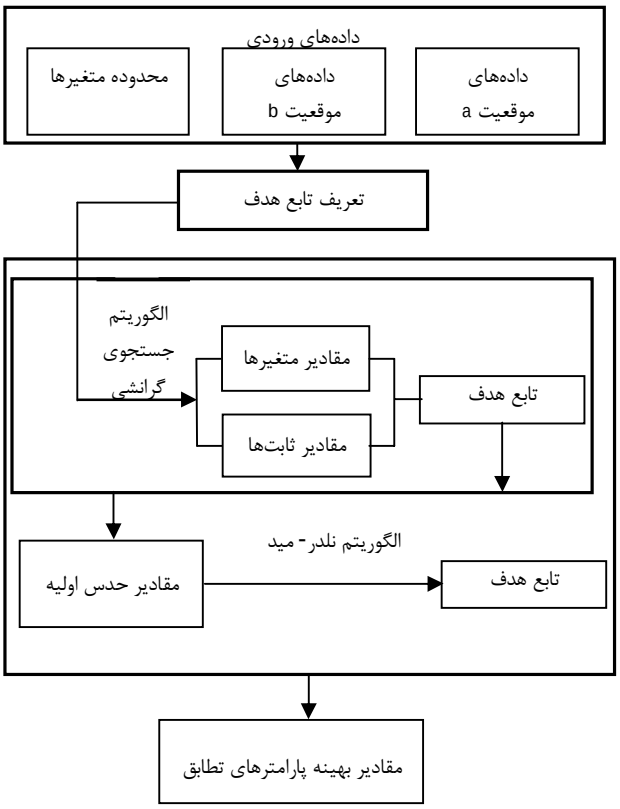
برای ارزیابی توانایی الگوریتم پیشنهادی، از کمینه کردن توابع محک استفاده شده و برخی از آن‌ها در جدول 1 آورده شده است [25]. توابع محک f_1، f_2 و f_6 دارای مقدار کمینه مطلق صفر هستند لذا سایر توابع برای تبدیل به کمینه مطلق صفر، مطابق روابط (21) تا (23) تغییر می‌کنند.

$$f_3(x)=-\cos(x_1)\cos(x_2)e^{-((x_1-\pi)^2+(x_2+\pi)^2)}+1 \quad (21)$$

$$f_4(x)=(1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2))(30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2))-3 \quad (22)$$

$$f_5(x)=\left(4-2.1x_1^2+\frac{x_1^4}{3}\right)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2+1.0316284 \quad (23)$$

الگوریتم پیشنهادی بر روی توابع ذکر شده، پیاده‌سازی شد و نتایج حاصل از اجرای 10 بار برنامه، در جدول 2 آورده شده است. در این جدول، الگوریتم پیشنهادی از صحت، دقت خوب در جواب و همگرایی مناسبی در تعداد تکرار برخوردار می‌باشد. مقادیر تعداد جرم، حداکثر تعداد تکرار، مقدار α و G_0 الگوریتم پیشنهادی، به ترتیب 30، 100، 20 و 100 تنظیم شده است. در شکل‌های 3 تا 8، تغییرات بهترین برازندگی با افزایش تعداد تکرار برای توابع f_1 تا f_6، نسبتا سریع می‌باشد.



شکل 2 نمودار روند نمای الگوریتم ترکیبی

جدول 1 توابع محک استاندارد

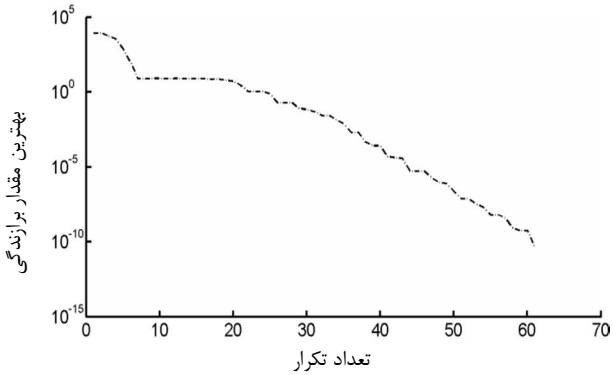
نام تابع	تابع
Baele	$f_1(x)=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1x_2)^2+(1.5-x_1+x_1x_2)^2$
Booth	$f_2(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$
Easom	$f_3(x)=-\cos(x_1)\cos(x_2)\,e^{-((x_1-\pi)^2+(x_2+\pi)^2)}$
Goldstein-price	$f_4(x)=(1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2))(30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2))-3$
Six-hump camel back	$f_5(x)=\left(4-2.1x_1^2+\frac{x_1^4}{3}\right)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$
Sphere	$f_6(x)=\sum_{i=1}^{2}x_i^2$

1- Pre-Registration
2- Simplex

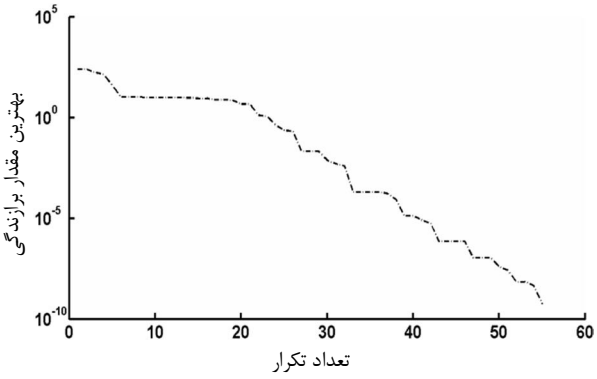
[Downloaded from mme.modares.ac.ir on 2024-04-18]
[DOR: 20.1001.1.10275940.1394.15.5.44.8]

جدول 2 نتایج حاصل از پیاده‌سازی الگوریتم پیشنهادی بر روی توابع محک

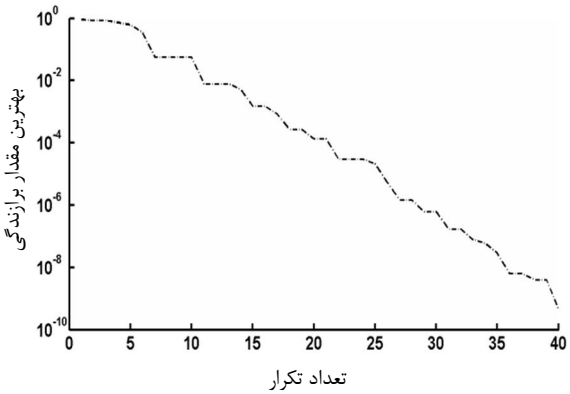
نام تابع	تعداد تکرار	مقدار متوسط همگرایی	جواب الگوریتم پیشنهادی	جواب دقیق
$f_1(x)$	60	$4/1\times10^{-11}$	3/000015 0/500003	3 0/5
$f_2(x)$	54	$5/5\times10^{-10}$	1/000017 2/999983	1 3
$f_3(x)$	39	$4/5\times10^{-10}$	3/141591 3/141610	π π
$f_4(x)$	33	$5/8\times10^{-8}$	$1/3\times10^{-5}$ -0/999989	0 -1
$f_5(x)$	42	$3/5\times10^{-8}$	-0/0897856 0/712625	-0/08984 0/71265
$f_6(x)$	52	$2/3\times10^{-10}$	$-5/3\times10^{-5}$ $-6/8\times10^{-5}$	0 0



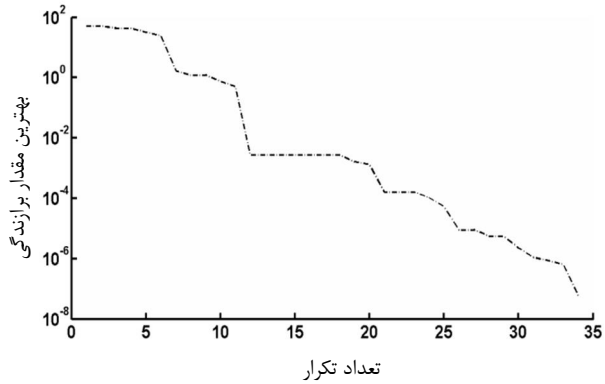
شکل 3 روند تغییر همگرایی و تعداد تکرار تابع محک f_1



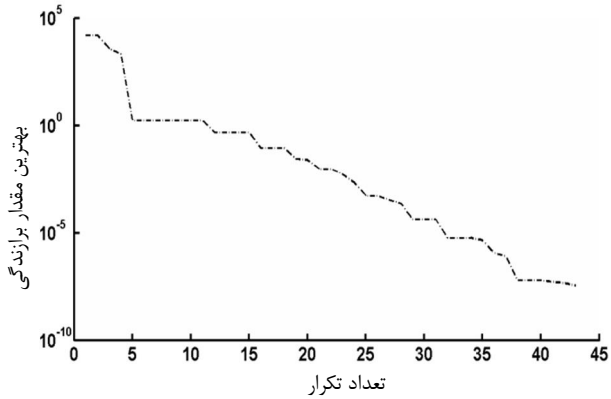
شکل 4 روند تغییر همگرایی و تعداد تکرار تابع محک f_2



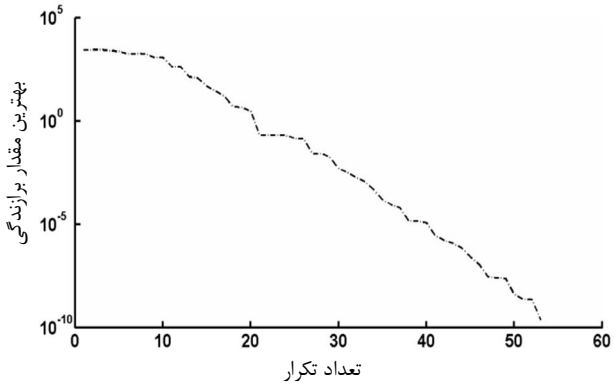
شکل 5 روند تغییر همگرایی و تعداد تکرار تابع محک f_3



شکل 6 روند تغییر همگرایی و تعداد تکرار تابع محک f_4



شکل 7 روند تغییر همگرایی و تعداد تکرار تابع محک f_5



شکل 8 روند تغییر همگرایی و تعداد تکرار تابع محک f_6

4- مطالعه موردی

روش‌های ابتکاری، راه‌حل‌های قابل قبول در زمان مناسب هستند. این روش‌ها برای حل مسائل ژرف و پیچیده در حوزه‌های مختلف، از جمله مسائل مهندسی مورد استفاده قرار می‌گیرند. یکی از مسائل مهندسی در حوزه مهندسی معکوس، فرایند تطابق ابر نقاط است. به کمک روش‌های ابتکاری، تابع هدف (رابطه (2)) کمینه شده و بهترین پارامترهای تطابق (چرخش و جابه‌جایی) به‌دست می‌آید.

اندازه برداری از روی قطعه به دلیل پیچیدگی قطعه یا برای حالتی که کورس دستگاه اجازه نمی‌دهد، ممکن است در دو یا چند مرحله برداشت نقاط انجام شده و سپس فرایند تطابق بر روی‌داده‌ها اعمال شود.

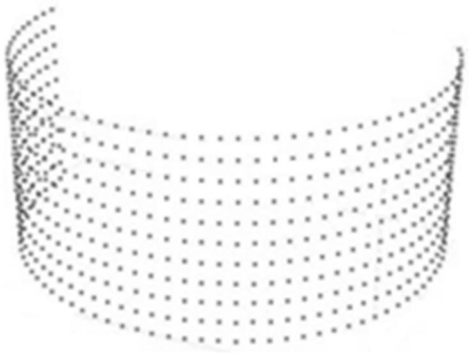
آزمایش تجربی فرایند تطابق ابر نقاط بر روی سه قطعه، شامل قطعه با هندسه منظم (نیم استوانه - شکل 9)، قطعه نمونه (شکل 10) و قطعه

[Downloaded from mme.modares.ac.ir on 2024-04-18]
[DOR: 20.1001.1.10275940.1394.15.5.44.8]

کاربردی (شکل 11) پیاده‌سازی شده که جزئیات روش برای قطعه شکل 11، به دلیل پیچیدگی بیشتر نسبت به دو قطعه دیگر در ادامه آورده شده است.

قطعه کاربردی با ابعاد کلی 60×140×300 میلی‌متر و از قسمت‌های مسطح و سطح آزاد تشکیل شده که بر روی بخش مسطح، یک سری سوراخ ماشین‌کاری شده است. سوراخ‌های بخش مرکزی قطعه، برای هم‌پوشانی دو بخش از قطعه که قرار است فرایند تطابق پیاده‌سازی شود، تعبیه شده است. تعداد این سوراخ‌ها 20 عدد می‌باشند. جنس قطعه از آلومینیوم سری 7000 اکسترود شده، انتخاب شده است تا پس از ماشین‌کاری، دارای صافی سطح قابل قبولی باشد. برای دقت عملیات ماشین‌کاری قطعه، از دستگاه فرز کنترل عددی استفاده شده است. مراحل آزمایش تجربی فرایند تطابق در ادامه آورده شده است.

مرحله اول: اندازه‌برداری نقاط از روی قطعه، در دو موقعیت بدون انتقال (چرخش و جابه‌جایی) و انتقال یافته توسط ماشین اندازه‌گیری مختصات، ساخت شرکت سوئدی جانسون[1] انجام شده است. تعداد نقاط اندازه‌برداری از روی قطعه، 390 نقطه بوده و نحوه اندازه‌برداری نقاط از روی قطعه، توسط ماشین اندازه‌گیری مختصات در شکل‌های 12 و 13 نشان داده شده است. میز گرانیتی دقیق به ابعاد 800×1000×1500 میلی‌متر و کنترلر رنی‌شاو[2] نصب شده بر روی آن، شرایط را برای دقت اسمی یک میکرون فراهم نموده است. برای آگاهی از میزان چرخش قطعه به طور تقریبی، از گیره و به طور



شکل 9 قطعه با هندسه منظم برای پیاده‌سازی فرایند تطابق

شکل 10 قطعه نمونه برای پیاده‌سازی فرایند تطابق

شکل 11 قطعه کاربردی ساخته شده برای پیاده‌سازی فرایند تطابق

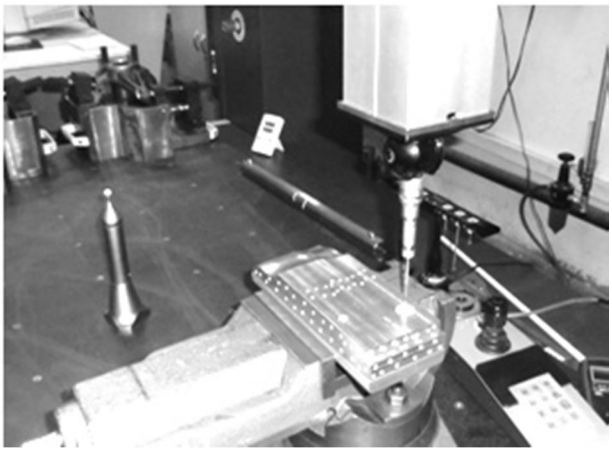
1- Johansson
2- Renishaw

دقیق توسط ماشین اندازه‌گیری مختصات استفاده شده است. مقادیر چرخش حول محورهای X، Y و Z به ترتیب 60/452، 14/863 و 30/267 درجه می‌باشند.

درصورتی‌که ابر نقاط قطعه انتخابی در حالت بدون انتقال، مطابق شکل 14 به دو بخش a و b تقسیم شود، برای پیاده‌سازی فرایند تطابق، بایستی بخش a شکل 14 (داده‌های بدون انتقال) با بخش انتقال‌یافته b، مطابق شکل 15 در نظر گرفته شوند. لذا برای داده‌های بدون انتقال a و انتقال‌یافته bبه ترتیب از مقادیر اندازه‌برداری شده در چیدمان شکل‌های 12 و 13 استفاده می‌شود. (لازم به ذکر است که بخش a و b دارای هم‌پوشانی به تعداد 20 نقطه مشترک می‌باشند).

مرحله دوم: فرایند تطابق به سه روش جستجوی گرانشی، جمعیت ذرات و ژنتیک به‌صورت جداگانه پیاده‌سازی شده است. برنامه مربوط به این الگوریتم‌ها در محیط نرم‌افزار متلب[3]، نوشته شده و در خروجی برنامه، مقادیر چرخش و جابه‌جایی (t و R) به‌دست آمده است.

مرحله سوم: در صورتی‌که ماتریس تبدیل (چرخش و جابه‌جایی) به‌دست آمده، در داده‌های قطعه پس از مراحل چرخش و جابه‌جایی تأثیر داده شود، داده‌های به‌دست آمده با داده‌های اولیه قطعه قبل از مراحل چرخش و جابه‌جایی، قابل‌مقایسه خواهد بود. اختلاف این داده‌ها، به عنوان خطای تطابق برای هر نقطه متناظر در نظر گرفته شده است.

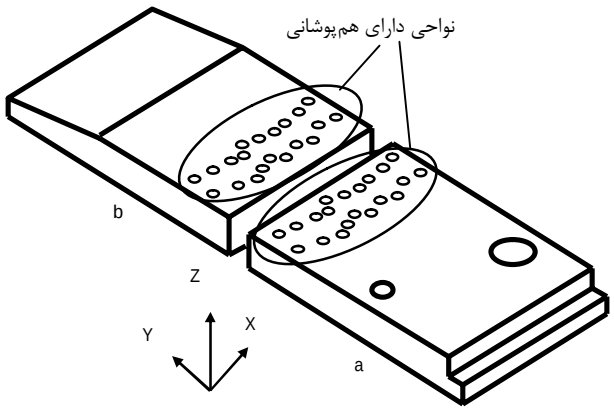


شکل 12 مرحله اندازه برداری نقاط توسط ماشین اندازه‌گیری مختصات بدون انتقال قطعه

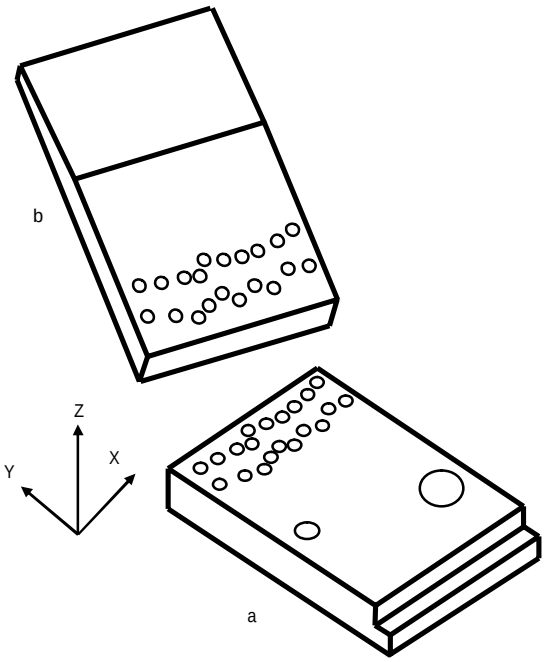
شکل 13 مرحله اندازه‌برداری نقاط توسط ماشین اندازه‌گیری مختصات در موقعیت انتقال داده شده

3- MATLAB

[Downloaded from mme.modares.ac.ir on 2024-04-18]
[DOR: 20.1001.1.10275940.1394.15.5.44.8]



شکل14 ابر نقاط بخش‌های a و b قطعه



شکل 15 ابر نقاط بخش b بعد از عملیات چرخش و جابه‌جایی

در این پژوهش، تابع هدف توسط الگوریتم‌های جستجوی گرانشی، جمعیت ذرات و ژنتیک به‌صورت جداگانه کمینه شده و پارامترهای تطابق به‌دست آمده است. سپس این الگوریتم‌ها از لحاظ دقت، میزان همگرایی و زمان اجرای برنامه با یکدیگر مقایسه شده و از مقایسه سه روش، بهترین الگوریتم ابتکاری انتخاب شده است. از آنجا که در پژوهش قبلی خسروی و همکاران [10] در پیاده‌سازی فرایند تطابق، بهترین پاسخ را از پارامترهای تطابق به کمک الگوریتم نلدر- مید به‌دست آوردند. در این تحقیق الگوریتم ترکیبی ارائه شده است که ترکیبی از الگوریتم ابتکاری انتخابی و الگوریتم نلدر- مید، می‌باشد.

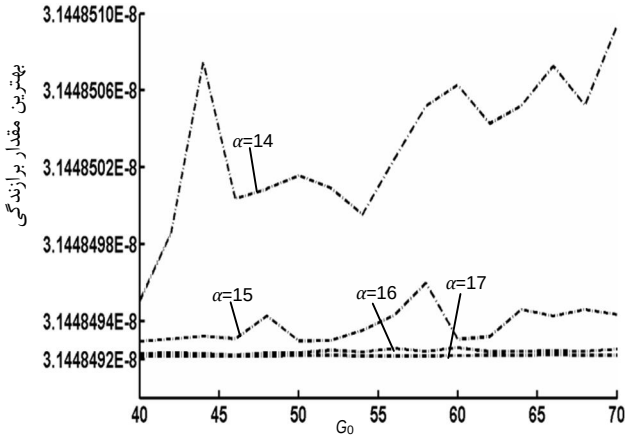
5- بحث و مقایسه روش‌های فرایند تطابق

الگوریتم‌های جستجوی گرانشی، جمعیت ذرات و ژنتیک روش‌های ابتکاری هستند که با تکرار و حدس اولیه جواب بهینه را به‌دست می‌آورند. در کلیه این روش‌ها از یک جمعیت کرومزوم، جمعیت ذرات و تعداد جرم‌ها برای رسیدن به جواب بهینه و کمینه کردن رابطه (2) استفاده شده است. مقدار جمعیت اولیه و تعداد نسل در دقت جواب تأثیر داشته و با افزایش این دو پارامتر زمان رسیدن به جواب افزایش می‌یابد. برای مقایسه بهتر نتایج با یکدیگر، جمعیت اولیه برای هر سه روش 30 و تعداد تکرار 1000 در نظر گرفته شده است.

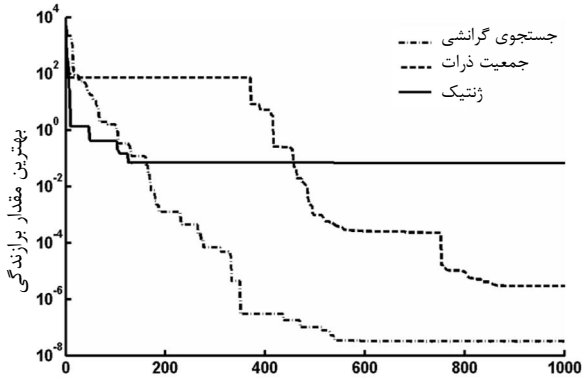
از آنجا که الگوریتم جستجوی گرانشی، دارای اشکالاتی نظیر امکان به‌دست آوردن جواب در کمینه‌های محلی و عدم حفظ تنوع جواب می‌باشد لذا با تنظیم پارامترهای جمعیت اولیه، G_0 و α، از بروز این اشکال‌ها در الگوریتم پیشنهادی جلوگیری به عمل آمده است. الگوریتم های جستجوی گرانشی و نلدر- مید به صورت منفرد دارای ایراداتی هستند، ولی درالگوریتم پیشنهادی از نقاط قوت هر دو الگوریتم، در جهت بهبود جواب و کاهش تعداد تکرار استفاده شده است.

تنظیم صحیح پارامترهای هر الگوریتم، بخش مهم مسأله بوده و میزان همگرایی را بهبود می‌بخشد و از طرفی مختص همان مسأله است. در این پژوهش مقدار نرخ جهش در روش ژنتیک 0/15، در روش جمعیت ذرات مقادیر ثابت C_2 و C_3 برابر 2 و در روش جستجوی گرانشی مقادیر ثابت G_0 و α به ترتیب 42 و 17 لحاظ شده است. با اجرای مکرر برنامه الگوریتم جستجوی گرانشی، برای مقادیر مختلف G_0 و α کمترین مقدار همگرایی و در نتیجه بهترین مقدار G_0 و α حاصل شده که نتایج در شکل 16 نشان داده شده است.

در شکل 17 بهترین برازندگی الگوریتم‌های جستجوی گرانشی، جمعیت ذرات و ژنتیک با توجه به تعداد تکرار (تعداد نسل)، مقایسه شده است. در این شکل الگوریتم جستجوی گرانشی، با درجه همگرایی بسیار خوب نسبت به الگوریتم‌های جمعیت ذرات و ژنتیک را نشان می‌دهد. نتایج حاصل از فرایند تطابق، با الگوریتم‌های ژنتیک، جمعیت ذرات و جستجوی گرانشی در جدول‌های 3 و 4 مقایسه شده است.



شکل 16 نمایش مقادیر مختلف G_0 و α با توجه به کمترین مقدار برازندگی



شکل 17 مقایسه بهترین مقدار برازندگی الگوریتم‌های جستجوی گرانشی، جمعیت ذرات و ژنتیک

[Downloaded from mme.modares.ac.ir on 2024-04-18]
[DOR: 20.1001.1.10275940.1394.15.5.44.8]

جدول 3 مقایسه زوایای چرخش در فرایند تطابق

نام روش	$R_x(\gamma)$ (deg.)	$R_y(\beta)$ (deg.)	$R_z(\alpha)$ (deg.)
ژنتیک	60/452252	14/864121	30/266055
جمعیت ذرات	60/451136	14/859786	30/269694
جستجوی گرانشی	60/451994	14/862950	30/267049
مقادیر واقعی	60/452	14/863	30/267

جدول 4 مقایسه مقادیر جابه‌جایی در فرایند تطابق

نام روش	t_x (mm)	t_y (mm)	t_z (mm)
ژنتیک	312/568741	120/960320	+0/243781
جمعیت ذرات	312/192788	120/728703	-0/042188
جستجوی گرانشی	312/234563	120/763626	-0/001765
مقادیر واقعی	312/236	120/766	0

با مقایسه مقادیر چرخش و جابه‌جایی اعمال‌شده در قطعه آزمایشی و نتایج به‌دست‌آمده از سه روش ذکرشده، مقادیر خطا در جابه‌جایی و چرخش به دست می‌آید. از روابط (24) و (25) می‌توان مقدار نهایی خطای جابه‌جایی و چرخش را به‌دست آورد.

$$E^2(R) = E^2(R_x) + E^2(R_y) + E^2(R_z) \quad (24)$$

$$E^2(t) = E^2(t_x) + E^2(t_y) + E^2(t_z) \quad (25)$$

در جدول 5 مقادیر خطای نهایی چرخش و جابه‌جایی بین الگوریتم‌های ژنتیک، جمعیت ذرات و جستجوی گرانشی مقایسه شده است. در این جدول کمترین خطای چرخش، به ترتیب مربوط به الگوریتم‌های جستجوی گرانشی، ژنتیک و جمعیت ذرات بوده و کمترین خطای جابجایی، به ترتیب مربوط به الگوریتم‌های جستجوی گرانشی، جمعیت ذرات و ژنتیک می‌باشد. در مجموع روش جستجوی گرانشی دارای کمترین خطای انتقال در مقایسه با الگوریتم‌های جمعیت ذرات و ژنتیک می‌باشد.

همان‌طور که بیان شد، قطعه آزمایشی شکل 12 به دو بخش a و b تقسیم شد و این دو بخش دارای هم‌پوشانی مراکز سوراخ‌ها به تعداد 20 سوراخ می‌باشند. بخش b پس از عملیات چرخش و جابه‌جایی در محل جدید قرار گرفت. حال اگر بخش b بر اساس مقادیر معکوس زوایا و جابه‌جایی جدول‌های 3 و 4 به موقعیت ابتدایی خود انتقال یابد، می‌توان مقدار خطای مراکز سوراخ‌های هم‌پوشانی را نیز به‌دست آورد.

به منظور بررسی زمان اجرای برنامه و میزان همگرایی هر یک از الگوریتم‌های ژنتیک، جمعیت ذرات و جستجوی گرانشی از یک محیط برنامه‌نویسی و پردازشگر واحد استفاده شده است. هر یک از برنامه‌ها به طور مجزا چندین بار در محیط متلب توسط پردازشگر هفت هسته‌ای[1] اجرا و نتایج آن در جدول 6 آورده شده است. در این جدول کمترین زمان اجرای برنامه، مربوط به الگوریتم جستجوی گرانشی می‌باشد. مقدار همگرایی در الگوریتم جمعیت ذرات، 93 برابر الگوریتم جستجوی گرانشی می‌باشد.

جدول 7 مقادیر آماره‌های خطای مراکز سوراخ‌های هم‌پوشانی در الگوریتم‌های ژنتیک، جمعیت ذرات و جستجوی گرانشی را نشان می‌دهد. در جدول 7 مقادیر $|\bar{E}|$، D_e، $\sum e^2$، σ به ترتیب مقدار متوسط بردار خطا، دامنه خطا، مجموع مربعات بردار خطا و انحراف معیار بردار خطا می‌باشند.

1- Intel(R) Core(TM) i7 2 GHZ

جدول 5 مقایسه خطای نهایی جابه‌جایی و چرخش در فرآیند تطابق

نام روش	$E(R)$ (deg.)	$E(t)$ (mm)
ژنتیک	0/001487	0/626304
جمعیت ذرات	0/004281	0/070980
جستجوی گرانشی	0/000070	0/003288

جدول 6 مقایسه زمان اجرای برنامه و همگرایی در فرآیند تطابق

نام روش	زمان اجرا (sec.)	همگرایی
ژنتیک	20/7	$6/6\times10^{-2}$
جمعیت ذرات	19/7	$2/9\times10^{-6}$
جستجوی گرانشی	18/9	$3/1\times10^{-8}$

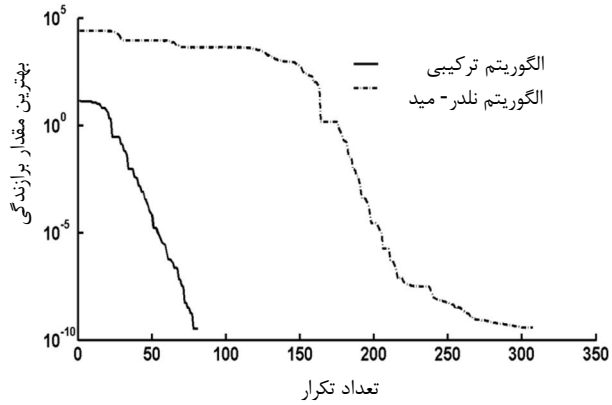
جدول 7 مقایسه خطای مراکز سوراخ‌های هم‌پوشانی در فرایند تطابق

نام روش	$\lvert\bar{E}\rvert$ (mm)	D_e (mm)	$e^2\sum$ (mm)	σ (mm)
ژنتیک	0/455315	0/001423	4/146245	0/000373
جمعیت ذرات	0/068966	0/003712	0/095175	0/000968
جستجوی گرانشی	0/003111	0/000085	0/000193	0/000023

الگوریتم نلدر- مید یک الگوریتم تکرار است و با حدس اولیه به جواب نهایی می‌رسد. در بخش اول الگوریتم ترکیبی، با تعداد تکرار پایین (تعداد تکرار 100)، جمعیت اولیه 30، G_0 به مقدار 42 و α به مقدار17، پارامترهای تطابق را نتیجه می‌دهد که مقدار حدس اولیه بخش دوم الگوریتم نلدر- مید می‌باشد. در نهایت مطابق جدول 8 الگوریتم ترکیبی با تعداد تکرار کمتر و همگرایی بهتر به جواب نهایی می‌رسد. هم چنین شکل 18 روند تغییر همگرایی و تعداد تکرار را در الگوریتم‌های ترکیبی و نلدر- مید نشان می‌دهد. در این شکل مقدار تغییرات همگرایی الگوریتم نلدر- مید تا تکرار حدودا 150، جزئی و بعد از آن به طور سریع کاهش یافته است. در صورتی که کاهش همگرایی در الگوریتم ترکیبی از تکرار اولیه به صورت سریع می‌باشد.

جدول 8 مقایسه الگوریتم نلدر- مید ترکیبی در تعداد تکرار و همگرایی

نام روش	تعداد تکرار	همگرایی
الگوریتم نلدر- مید	306	$3/7\times10^{-10}$
الگوریتم ترکیبی	80	$3/2\times10^{-10}$



شکل 18 روند تغییر همگرایی و تعداد تکرار در الگوریتم‌های ترکیبی و نلدر- مید

[Downloaded from mme.modares.ac.ir on 2024-04-18]
[DOR: 20.1001.1.10275940.1394.15.5.44.8]

6- نتیجه‌گیری

در این پژوهش برای نخستین بار از الگوریتم‌های جستجوی گرانشی و ترکیبی، برای بهینه‌سازی فرایند تطابق استفاده شده است. الگوریتم جستجوی گرانشی، در مقایسه با الگوریتم‌های جمعیت ذرات و ژنتیک دارای دقت جابه‌جایی و چرخشی بهتری می‌باشد. هم‌چنین الگوریتم جستجوی گرانشی، از همگرایی بسیار متفاوتی از الگوریتم‌های جمعیت ذرات و ژنتیک برخوردار است و زمان اجرای آن کمی بهتر از این الگوریتم‌ها است. از طرفی الگوریتم پیشنهادی در این پژوهش که ترکیبی از دو روش جستجوی گرانشی و نلدر- مید است، به مراتب از روش جستجوی گرانشی به لحاظ همگرایی و نیز از روش نلدر- مید به جهت تعداد تکرار، برتری دارد. الگوریتم ترکیبی در مقایسه با الگوریتم نلدر- مید، 13 درصد همگرایی را بهبود بخشیده و تعداد تکرار را 73 درصد کاهش داده است.

7- مراجع

[1] S. S. Rao, *Engineering optimization: theory and practice,* 4th Edition, John Wiley & Sons, 2009.
[2] J. S. Arora, *Introduction to optimum design,* Academic Press, San Diego, 2004.
[3] E. M. Beale, *Introduction to optimization,* John Wiley & Sons, 1988.
[4] A. Abraham, A. E. Hassanien, P. Siarry, A. Engelbrecht, *Foundations of computational intelligence: global optimization,* Heidelberg, Springer, 2009.
[5] Z. Xie, S. Xu, X. Li, A high-accuracy method for fine registration of overlapping point clouds, *Image and Vision Computing,* No. 28, pp. 563–570, 2010.
[6] A. Nüchter, J. Elseberg, P. Schneider, D. Paulus, Study of parameterizations for the rigid body transformations of the scan registration problem, *Computer Vision and Image Understanding,* No. 114, pp. 963–980, 2010.
[7] P.J. Besl, N.D. Mckay, A method for registration of 3-d shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 14, No. 2, pp. 239–256, 1992.
[8] Y. Chen, G. Medioni, Object modeling by registration of multiple range images, *In: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, IEEE Sacramento, California,* pp. 2724–2729, 1991.
[9] J. H. Holland, Adaption in natural and artificial systems, *Cambridge MA,* MIT Press, 1975.
[10] M. Khosravi, K. Khalili, H. Amirabadi, Employing nelder-mead method in point clouds registration prossce and comparison of the results with singular value decomposition and genetic algorithm, *Modares Mechanical Engineering,* Vol. 14, No. 9, pp. 117-124, 2014 (In Persian).
[11] S. Schenk, K. Hanke, Automatic registration of laser scanner point clouds with genetic algorithm, *Vermessung and Geoinformation,* No. 2, pp. 162-170, 2011.
[12] M. Salomon, G. Perrin, F. Heitz, Differential evolution for medical image registration, *International on Artificial Intelligence,* pp. 201–207, 2001.
[13] C. Robertson, R. Fisher, Parallel evolutionary registration of range data, *Registration Understanding,* No. 87, pp. 39–55, 2002.
[14] L. Silva, O. Bellon, K. Boyer, Enhanced, robust genetic algorithms for multiview range image registration, *Fourth International Conference on 3-D Digital Imaging and Modeling,* pp. 268–275, Banff, Canada, 2003.
[15] C. K. Chow, H. T. Tsui, T. Lee, Surface registration using a dynamic genetic algorithm, Pattern Recognition, No. 37, pp. 105–117, 2004.
[16] J. Kennedy, R. C. Eberhart, Particle swarm optimization, in *Proceedings of IEEE International Conference on Neural Networks,* Perth, Australia, Vol. 4, pp. 1942-1948, 1995.
[17] Q. Yu, K. Wang, *A point cloud registration strategy combining particle swarm optimisation and iterative closest point method,* IWAMA Trondheim, Norway, 2013.
[18] B. Schutz, *Gravity from the ground up,* Cambridge University Press, 2003.
[19] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, *Information Sciences,* No. 179, pp. 2232-2248, 2009.
[20] E. G. Talbi, A taxonomy of hybrid metaheuristics, *Journal of Heuristics,* No. 8, pp. 541–564, 2002.
[21] S. Masrom, S. Z. Z. Abidin, N. Omar, Rapid and flexible user-defined low-level hybridization for metaheuristics algorithm in software framework, *Journal of Software Engineering and Applications,* No. 5, pp. 873-882, 2012.
[22] E. G. Talbi, *Metaheuristics from design to implementation,* John Wiley & Suons, 2009.
[23] E. Lomonosov, D. Chetverikov, A. Eka´rt, Pre-registration of arbitrarily oriented 3d surfaces using a genetic algorithm, *Pattern Recognition Letters,* No. 27, pp. 1201–1208, 2006.
[24] G. Mansour, S. Mitsi, K. D. Bouzakis, D. Sagris, E. Varitis, Developed hybrid genetic algorithm for optimizing reverse engineering methods, *International Journal of Modern Manufacturing Technologies,* Vol. 2, No. 1, pp. 43–48, 2010.
[25] M. Jamil, X. S. Yang, A literature survey of benchmark functions for global optimization problems, *Int. Journal of Mathematical Modelling and Numerical Optimisation,* Vol. 4, No. 2, pp. 150–194, 2013.

[Downloaded from mme.modares.ac.ir on 2024-04-18]
[DOR: 20.1001.1.10275940.1394.15.5.44.8]