

# الگوریتم جستجوی گرانشی چندهدفه مبتنی بر مرتب‌سازی جبهه‌های مغلوب‌نشده

محمد امیر عباسیان<sup>۱</sup>، حسین نظام آبادی پور<sup>۲</sup>

۱- بخش مهندسی برق، دانشگاه شهید باهنر کرمان، کرمان، ایران. [m.amir.abbasian@gmail.com](mailto:m.amir.abbasian@gmail.com)

۲- بخش مهندسی برق، دانشگاه شهید باهنر کرمان، کرمان، ایران. [nezam@mail.uk.ac.ir](mailto:nezam@mail.uk.ac.ir)

**چکیده:** در سال‌های اخیر روش‌های فراوانی بر اساس الگوریتم‌های بهینه‌سازی ابتکاری برای بهینه‌سازی مسائل چندهدفه ارائه شده است. هر چند این روش‌ها ضمانتی بر یافتن جبهه بهینه پرتو ندارند ولی برای یافتن تقریب نسبتاً خوبی از این مجموعه تلاش می‌کنند. در این مقاله، روشی بر اساس الگوریتم جستجوی گرانشی (GSA) برای بهینه‌سازی مسائل چندهدفه بر مبنای مفاهیم پرتو ارائه شده که موسوم به الگوریتم جستجوی گرانشی چندهدفه با جبهه‌های مغلوب نشده (NF-MOGSA) است. در روش پیشنهادی از مفاهیم جبهه‌های مغلوب نشده برای تخصیص شایستگی و فاصله ازدحامی راه‌حل‌ها برای حفظ تنوع جمعیت استفاده شده است. برای ارزیابی الگوریتم پیشنهادی، مجموعه‌ای از آزمایش‌ها انجام و در آن الگوریتم پیشنهادی به توابع محک استاندارد SCH, KUR, FON, POL, ZDT1, ZDT2, ZDT3 و ZDT6 اعمال شد. نتایج به دست آمده بر روی توابع محک استاندارد و مقایسه با الگوریتم‌های مطرح در این زمینه، نشان دهنده مناسب بودن روش پیشنهادی در حل مسائل بهینه‌سازی چندهدفه است.

**واژه‌های کلیدی:** مسائل بهینه‌سازی چندهدفه، الگوریتم جستجوی گرانشی، تخصیص شایستگی، حفظ تنوع و نخبه گرایی.

## Multi Objective Gravitational Search Algorithm Using Non-dominated Fronts

Mohammad Amir Abbasian, Hossein Nezamabadi-pour

Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran.

**Abstract-** Nowadays, many techniques based on the heuristic optimization algorithms have been presented for multi-objective optimization problems. Although these techniques don't guarantee to find the Pareto optimal front, they quest to achieve a good approximation of this set. In this paper, a technique based on gravitational search algorithm (GSA) is presented for optimization of multi-objective problem named multi-objective optimization gravitational search with non-dominated fronts (NFMOGSA). In the proposed method, the non-dominated front and the crowding distance of solutions are used to fitness assignment and diversity preservation. To evaluate the proposed algorithm, some experiments are done and the NFMOGSA is applied to the standard benchmark functions, SCH, KUR, FON, POL, ZDT1, ZDT2, ZDT3 and ZDT6. The results of applying NFMOGSA on the standard benchmark functions and comparison with other well known multi-objective optimization techniques confirm the efficiency and effectiveness of our technique in the multi-objective optimization problems.

**Keywords-** Multi-objective optimization problems, Gravitational search algorithm (GSA), Fitness assignment, Diversity preservation and Elitism.

تاریخ ارسال مقاله:

تاریخ اصلاح مقاله: ۹۱/۱/۲۰

تاریخ پذیرش مقاله: ۹۱/۱/۳۰

نام نویسنده مسئول: دکتر حسین نظام‌آبادی پور

نشانی نویسنده مسئول: کرمان، میدان پژوهش، دانشگاه شهید باهنر کرمان، بخش مهندسی برق.

## ۱- مقدمه

بسیاری از مسائل جهان حقیقی را می‌توان به صورت مسائل بهینه‌سازی با اهداف چندگانه در نظر گرفت که شامل اهدافی مغایر<sup>۱</sup>، در فضای جستجوی<sup>۲</sup> بسیار بزرگ و پیچیده هستند [۱]. روش‌های بهینه‌سازی متعارف، مثل روش‌های مبتنی بر گرادیان<sup>۳</sup> و سیمپلکس<sup>۴</sup> و نیز روش‌های ابتکاری مثل پخت شبیه‌سازی شده<sup>۵</sup> به دلیل این‌که در اصل برای حل مسائل چندهدفه طراحی نشده‌اند، برای بهینه‌سازی این مسائل کارآمد نیستند [۲].

در مسائل بهینه‌سازی تک هدفه<sup>۶</sup> (SOP) راه‌حل (پاسخ/جواب) بهینه معمولاً به وضوح قابل تعریف است. اما در مسائل بهینه‌سازی چندهدفه (MOP)، ممکن است برای یک هدف، یک راه‌حل بهترین جواب باشد در حالی که برای اهداف دیگر جواب مناسبی نباشد. بنابراین در MOP به جای یک جواب بهینه واحد، مجموعه‌ای از جواب‌ها وجود دارد که در مصالحه<sup>۷</sup> با یکدیگرند. از خصوصیات این مجموعه از جواب‌های بهینه این است که قابل قیاس با یکدیگر نبوده و چنانچه همه‌ی اهداف در نظر گرفته شوند، نسبت به یکدیگر برتری ندارند. هدف از بهینه‌سازی چندهدفه، یافتن مجموعه‌ای از چنین جواب‌هایی است که به آن، مجموعه بهینه پرتو می‌گویند [۳]. البته تعیین مجموعه بهینه پرتو به صورت کامل امری غیر ممکن است. برای بسیاری از مسائل، به‌خصوص مسائل بهینه‌سازی ترکیباتی یافتن مجموعه جواب‌های بهینه پرتو از لحاظ محاسباتی ناممکن است. بنابراین، یک راه‌حل شدنی برای بهینه‌سازی مسائل چندهدفه تخمین مجموعه راه‌حل بهینه است [۴].

در الگوریتم‌های چندهدفه ابتکاری اصولی پایه وجود دارد که همواره به عنوان دغدغه‌ای برای طراحی این روش‌ها و الگوریتم‌ها بوده است. مفاهیمی همچون [۳، ۵]:

- تخصیص شایستگی: چگونه تخصیص شایستگی و تحقق عملکرد انتخاب انجام شود تا جستجو به سمت مجموعه بهینه پرتو هدایت شود.
- حفظ تنوع: چگونه تنوع جمعیت حفظ شود تا از هم‌گرایی زودرس<sup>۸</sup> جلوگیری شده و رسیدن به یک توزیع خوب و مجموعه مغلوب‌نشده گسترده، ضمانت شود.
- نخبه گرایی: چگونه از گم شدن جواب‌های مغلوب‌نشده در روند تکامل جمعیت جلوگیری شود.

در سال‌های اخیر روش‌های زیادی برای بهینه‌سازی مسائل چندهدفه ارائه شده است. معروف‌ترین و پرکاربردترین این روش‌ها بر اساس تکامل طبیعی است. در این روش‌ها به طور تصادفی  $N$  فرد به عنوان جمعیت اولیه ایجاد شده و شایستگی آن‌ها ارزیابی می‌شود. سپس، افراد بر اساس شایستگی‌شان انتخاب شده و وارد حوضچه ازدواج می‌شوند و به طور اتفاقی بر روی آن‌ها تغییراتی داده می‌شود تا جمعیت فرزند تولید گردد. این تغییرات به کمک دو عملگر همبری (ترکیب) و جهش انجام می‌گیرند. پس از ایجاد جمعیت فرزند در صورتی‌که شرط توقف برقرار نشده باشد، از جمعیت فرزند برای ایجاد جمعیت بعدی استفاده می‌شود و این روند تکرار می‌شود. استفاده از راهبردهای مختلف برای اجرای اصول بیان شده، سبب ایجاد الگوریتم‌های متعددی شده است. به طور مثال در روش SPEA2 [۶] که نسخه بهبود یافته SPEA [۷] است، تخصیص شایستگی بر اساس قدرت مغلوب کنندگی راه‌حل‌ها انجام می‌شود. ضمن آن‌که از یک معیار فاصله که تراکم راه‌حل‌ها را در فضای جستجو یا اهداف می‌سنجد برای ایجاد تنوع در جمعیت استفاده می‌شود. به کارگیری آرشیو خارجی برای حفظ راه‌حل‌های مناسب به دست آمده تا لحظه جاری از دیگر سیاست‌های این روش است.

روش دیگر NSGA2 [۸]، نسخه اصلاح شده NSGA [۹] است. در این روش جبهه‌های مغلوب‌نشده جمعیت رتبه‌بندی<sup>۹</sup> شده و با توجه به رتبه جبهه برای هر راه‌حل شایستگی تخصیص داده می‌شود. همچنین از فاصله ازدحامی راه‌حل‌ها در هر جبهه به عنوان معیاری برای ایجاد تنوع در زمان انتخاب، استفاده می‌شود. هر چند در این روش از آرشیو خارجی استفاده نشده است، اما با ترکیب نسل والد و فرزند و انتخاب بهترین‌ها از میان آن‌ها به نوعی نخبه‌گرایی صورت گرفته است. از روش‌های دیگر می‌توان به VEGA [۱۰]، MOGA [۱۱]، RWGA [۱۲]، NPGA [۱۳]، PAES [۱۴]، RDGA [۱۵]، MOEA با دو آرشیو [۱۶]، OWMOSAGE [۱۷] و DMOEA [۱۸] اشاره کرد.

از روش‌های دیگر برای بهینه‌سازی چندهدفه، روش بهینه‌سازی جمعیت ذرات است که به وسیله کندی و ابره‌ارت در سال ۱۹۹۵ ارائه شد [۱۹]. در این روش، حرکت جمعیت به سمت امیدوارکننده‌ترین ناحیه از فضای جستجو هدایت می‌شود و هر ذره نماینده یک راه‌حل است که دارای موقعیت و سرعت در فضای جستجو است. در هر تکرار سرعت و موقعیت با توجه به بهترین موقعیت به دست آمده تا لحظه جاری برای هر ذره و نیز بهترین موقعیت کلی، به روز می‌شود. الگوریتم TVMOPSO [۲۰] از روش‌های معروفی است که بر اساس PSO در

جواب‌های خوب از آرشیو حذف شوند. همچنین استفاده از عملگر وراثتی جهش، الگوریتم را از ذات گرانشی‌اش دور ساخته است. روش پیشنهاد شده در مرجع [۲۷] اگر چه توانایی خوبی در آشکارسازی جبهه پرتو دارد و نشان داد که الگوریتم گرانشی قادر به حل مسائل بهینه‌سازی چندهدفه است اما در قیاس با کارهای روز در این زمینه قدری ضعیف‌تر است. این موضوع در آزمایش‌های انجام شده در این مقاله به اثبات می‌رسد.

از این رو، در این مقاله روشی برای بهینه‌سازی مسائل چندهدفه بر مبنای الگوریتم گرانشی ارائه شده که از یک طرف کلیه ساختارهای حاکم بر الگوریتم گرانشی را حفظ کرده و از طرف دیگر قابل رقابت با سایر الگوریتم‌های مطرح در زمینه بهینه‌سازی چندهدفه است. برای این منظور در نسخه پیشنهادی از ابزاری چون مرتب‌سازی سریع جبهه‌های مغلوب نشده، فاصله ازدحامی و تخصیص جرم برای هدایت عامل‌ها به سمت جبهه بهینه پرتو استفاده شده است.

در ادامه این مقاله و در بخش دوم برخی مفاهیم پایه درباره بهینه‌سازی چندهدفه آورده شده است. سپس در بخش سوم الگوریتم جستجوی گرانشی پایه بیان شده و در بخش چهارم روش پیشنهادی برای مسائل بهینه‌سازی چندهدفه با استفاده از الگوریتم جستجوی گرانشی موسوم به الگوریتم جستجوی گرانشی چندهدفه با جبهه‌های مغلوب‌نشده<sup>۱۱</sup> (NFMOGSA) آمده است. در بخش پنجم نتایج به دست آمده از NFMOGSA بروی توابع محک با روش‌های SPEA2 [۶]، NSGA2 [۸]، TVMOPSO [۲۰] و MOGSA [۲۷] مقایسه شده است و آخرین بخش به جمع‌بندی این مقاله می‌پردازد.

## ۲- بهینه‌سازی چندهدفه

در این بخش به برخی مفاهیم پایه درباره مسئله بهینه‌سازی چندهدفه اشاره شده است [۳].

تعریف ۱- (مسئله بهینه‌سازی چندهدفه). یک مسئله بهینه‌سازی چندهدفه (MOP)، شامل مجموعه‌ای  $n$  پارامتری (متغیرهای تصمیم‌گیری)، از  $K$  تابع هدف و مجموعه‌ای از  $m$  قید است که اهداف و قیود، توابعی از متغیرهای تصمیم‌گیری هستند. هدف از بهینه‌سازی چندهدفه عبارت است از (به عنوان مثال بیشینه‌سازی):

$$\begin{aligned} \text{maximize} \quad & y = f(x) = (f_1(x), f_2(x), \dots, f_K(x)) \\ \text{subject to} \quad & e(x) = (e_1(x), e_2(x), \dots, e_l(x)) \leq 0 \\ \text{where:} \quad & x = (x_1, x_2, \dots, x_n) \in X \\ & y = (y_1, y_2, \dots, y_n) \in Y \end{aligned} \quad (1)$$

زمینه بهینه‌سازی چندهدفه ارائه شده است. در این روش، تخصیص شایستگی بر اساس تراکم راه‌حل‌ها در اطراف راه‌حل مغلوب‌نشده انجام می‌پذیرد و پارامتر تراکم راه‌حل‌ها، ضمانت‌کننده حفظ تنوع در این روش است. استفاده از آرشیو خارجی برای نخبه‌گرایی از دیگر ویژگی‌های این الگوریتم است.

روش دیگری که بر اساس PSO ارائه شده، روش PAPSO است [۲۱] که از آرشیو خارجی برای نخبه‌گرایی و از فاصله بین راه‌حل‌ها برای ایجاد تنوع استفاده می‌کند. همچنین از مفهوم غلبگی پرتو به عنوان معیاری برای تخصیص شایستگی سود می‌برد. از دیگر روش‌ها در این زمینه می‌توان به MSPSO [۲۲]، PO-MOPSO [۲۳] و DCMOPSO [۲۴] اشاره کرد. برای یک بررسی کامل از روش‌های موجود در زمینه بهینه‌سازی چند هدفه می‌توان به مراجع [۲]، [۴]، [۵] و [۲۵] مراجعه کرد.

یکی از روش‌های جدید در زمینه بهینه‌سازی تک هدفه، الگوریتم جستجوی گرانشی (GSA) است که در سال ۲۰۰۹ به وسیله راشدی و همکاران ارائه شد [۲۶]. این روش از قوانین جاذبه و حرکت نیوتن برای جستجو استفاده می‌کند. در این الگوریتم، ذرات به عنوان عامل‌های جستجوکننده در نظر گرفته می‌شوند و جرم آن‌ها متناسب با عملکردشان محاسبه می‌شود. همچنین از نیروی جاذبه به عنوان ابزاری برای تبادل اطلاعات استفاده می‌شود. براساس نیروی جاذبه، تمام عامل‌ها یکدیگر را جذب می‌کنند و این نیرو سبب حرکت کلی عامل‌ها به سمت عامل‌هایی با جرم بزرگ‌تر می‌شوند. به مرور زمان امیدواریم که اجرام به‌وسیله سنگین‌ترین جرم که همان جواب بهینه است جذب شوند [۲۶].

در مرجع [۲۷]، شیوه جدیدی برای بهینه‌سازی چندهدفه بر مبنای GSA به نام الگوریتم جستجوی گرانشی چندهدفه<sup>۱۰</sup> (MOGSA) ارائه شد که عملکرد نسبتاً قابل قبولی براساس همگرایی، تنوع و یافتن راه‌حل‌های حدی دارد. این الگوریتم اولین الگوریتم بهینه‌سازی چندهدفه است که بر پایه الگوریتم جستجوی گرانشی استوار است. در این روش، فرآیند تخصیص شایستگی و استفاده از آرشیو خارجی، بر اساس روش SPEA2 است. اجرای این دو فرآیند، سبب شده تا ضمن افزایش پیچیدگی این روش، زمان اجرای الگوریتم نیز افزایش یابد. در این الگوریتم، در تکرارهای پایانی زمانی که اعضای آرشیو و جمعیت به یکدیگر نزدیک می‌شوند، فاصله اعضا به یکدیگر کم شده و همسایگان اعضای جمعیت و آرشیو تقریباً فاصله یکسانی با هم پیدا می‌کنند. در این حالت انتخاب  $k$  امین نزدیک‌ترین همسایه پیچیده‌تر شده و ممکن است

و مفهوم جرم ارائه شد. این الگوریتم بر اساس گرانش نیوتن است. کارآمدی این الگوریتم در بهینه‌سازی توابع محک استاندارد، برای بهینه‌سازی مسائل تک هدفه به اثبات رسیده است [۲۶].

در این الگوریتم، عوامل به عنوان اجسام در نظر گرفته شده و عملکردشان به وسیله جرم آن‌ها اندازه‌گیری می‌شود. همه این اجسام به وسیله نیروی گرانشی همدیگر را جذب می‌کنند. این نیرو، حرکتی کلی را برای تمام اجسام به سمت اجسام با جرم سنگین‌تر سبب می‌شود. از این رو، اجرام با استفاده از دریافت اطلاعات از طریق نیروی گرانش با یکدیگر همیاری دارند.

الگوریتم جستجوی گرانشی، به عنوان یک سیستم مجزا از اجسام، شبیه یک جهان مصنوعی کوچک در نظر گرفته می‌شود که اجرام آن از قوانین گرانش و حرکت نیوتن پیروی می‌کنند.

در سیستمی متشکل از مجموعه ای با  $N$  عامل (جرم) موقعیت  $i$  امین عامل به صورت زیر تعریف می‌شود:

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \text{ for } i=1, 2, \dots, N \quad (6)$$

که  $x_i^d$  نشان دهنده موقعیت  $i$  امین عامل در  $d$  امین بعد است. مقدار نیروی وارده بر جرم  $i$  از طرف جرم  $j$  در زمان  $t$  عبارت است از:

$$F_{ij}^t(t) = G(t) \frac{M_{aj} \times M_{pi}}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (7)$$

$M_{aj}$  جرم گرانشی اکتیو عامل  $j$  ام،  $M_{pi}$  جرم گرانشی پسیو عامل  $i$  ام و  $G(t)$  ثابت گرانش در لحظه  $t$ ، یک مقدار ثابت کوچک و  $R_{ij}(t)$  فاصله اقلیدسی میان دو عامل  $i$  و  $j$  است:

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2, \quad (8)$$

برای ایجاد خاصیت اتفاقی در الگوریتم، فرض شده است که کل نیرویی که بر عامل  $i$  ام در بعد  $d$  اثر می‌گذارد، مجموع وزندار شده تصادفی از  $d$  امین مؤلفه نیروهای اعمال شده از طرف سایر عوامل است:

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t), \quad (9)$$

که  $rand_j$  یک عدد تصادفی با تابع توزیع یکنواخت در بازه  $[0,1]$  است. بنابر قانون حرکت، شتاب عامل  $i$  در لحظه  $t$  و بعد  $d$  خواهد بود:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}, \quad (10)$$

در این رابطه،  $M_{ii}(t)$  جرم اینرسی عامل  $i$  ام است. بنابراین، موقعیت و سرعت  $i$  امین ذره به صورت زیر محاسبه می‌شود:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t), \quad (11)$$

در این رابطه،  $x$  بردار تصمیم‌گیری<sup>۱۲</sup>،  $y$  بردار هدف<sup>۱۳</sup>،  $X$  فضای تصمیم<sup>۱۴</sup> و  $Y$  فضای هدف<sup>۱۵</sup> است و قیود  $e(x) \leq 0$  مجموعه جواب‌های شدنی را مشخص می‌کنند.

تعریف ۲- (مجموعه جواب‌های شدنی<sup>۱۶</sup>). مجموعه شدنی  $x_f$  به صورت مجموعه‌ای از بردارهای تصمیم‌گیری  $x$  تعریف می‌شوند که قید  $e(x)$  را برآورده می‌سازند.

$$x_f = \{x \in X | e(x) \leq 0\} \quad (2)$$

تصویر  $x_f$ ، ارائه کننده نواحی شدنی در فضای هدف می‌باشد که به صورت زیر مشخص می‌شود:

$$Y_f = f(X_f) = \bigcup_{x \in X_f} \{f(x)\} \quad (3)$$

تعریف ۳- (غلبگی پرتو<sup>۱۷</sup>). برای هر یک از دو بردار تصمیم  $a$  و  $b$  داریم:

$$\bullet \quad a \succ b \text{ (بر } b \text{ غلبه می‌کند)}^{18}, \text{ اگر و تنها اگر } f(a) \succ f(b)$$

$$\bullet \quad a \geq b \text{ (به طور ضعیفی بر } b \text{ غلبه می‌کند)}^{19}, \text{ اگر و تنها اگر } f(a) \geq f(b)$$

$$\bullet \quad a \sim b \text{ (نسبت به } b \text{ بی تفاوت است)}^{20}, \text{ اگر و تنها اگر } f(a) \not\geq f(b) \wedge f(b) \not\geq f(a)$$

تعریف ۴- (بهینگی پرتو<sup>۲۱</sup>). به بردار تصمیم  $x \in X_f$  مغلوب نشده گویند، اگر و تنها اگر، با توجه به مجموعه  $A \subseteq X_f$ :

$$\nexists a \in A : a \succ x \quad (4)$$

تعریف ۵- (مجموعه‌ها و جبهه‌های مغلوب نشده).  $A \subseteq X_f$  قرار دهید. تابع  $p(A)$  مجموعه بردارهای تصمیم‌گیری مغلوب نشده در  $A$  را می‌دهد:

$$p(A) = \{a \in A | a \text{ is nondominated regarding } A\} \quad (5)$$

با توجه به  $A$  مجموعه  $p(A)$ ، مجموعه مغلوب نشده و مجموعه متناظر بردارهای هدف  $f(p(A))$  جبهه مغلوب نشده مربوط به  $A$  می‌باشد. همچنین مجموعه  $p(X_f) = X_p$  را مجموعه بهینه پرتو و مجموعه  $Y_p = f(X_p)$  را جبهه بهینه پرتو می‌نامند.

### ۳- الگوریتم جستجوی گرانشی (GSA)

الگوریتم جستجوی گرانشی از جدیدترین الگوریتم‌های ابتکاری هوش جمعی است که به وسیله راشدی، نظام‌آبادی‌پور و سریزدی در سال ۲۰۰۹ برای بهینه‌سازی مسائل تک هدفه (SOP)، با الهام از قانون جاذبه

۴. به روز رسانی  $G(t)$ ,  $best(t)$ ,  $worst(t)$ ,  $M_i(t)$  برای  $i=1,2,...,N$
۵. محاسبه نیروی کلی در ابعاد مختلف
۶. محاسبه شتاب و سرعت
۷. به روز رسانی موقعیت عوامل
۸. تکرار مراحل ۳ تا ۷ در صورت فرا نرسیدن شرط توقف
۹. پایان

#### ۴- الگوریتم جستجوی گرانشی چندهدفه پیشنهادی با جبهه‌های مغلوب نشده (NFMOGSA)

در الگوریتم پیشنهاد شده از روش مرتب‌سازی سریع جبهه‌های مغلوب نشده که در روش NSGA2 [۸] آمده برای تخصیص شایستگی و از روش فاصله ازدحامی برای ایجاد تنوع در جمعیت استفاده شده است. همچنین برای جلوگیری از نابودی راه‌حل‌های خوب در هر تکرار، موقعیت عامل‌ها قبل و پس از به روز رسانی، با هم ترکیب شده و سپس  $N$  تا از بهترین راه‌حل‌ها برای ادامه کار الگوریتم انتخاب می‌شود. بدین ترتیب راه‌حل‌های مناسب در طول تکرارها حفظ خواهند شد.

##### ۴-۱- روش مرتب‌سازی سریع جبهه‌های مغلوب‌نشده

در روش مرتب‌سازی سریع برای هر راه‌حل درون جمعیت دو پارامتر محاسبه می‌شود: الف) رتبه مغلوب شدگی  $n_p$ ، که عبارت است از تعداد جواب‌هایی که بر حل  $p$  ام غلبه کرده‌اند؛ ب)  $s_p$ ، که عبارت است از مجموعه جواب‌هایی که جواب  $p$  بر آن‌ها غلبه کرده است. تمام جواب‌هایی که در اولین جبهه مغلوب نشدگی قرار دارند دارای رتبه مغلوب شدگی ( $n_p$ ) صفر هستند. حال برای هر حل  $p$  با  $n_p=0$  هر عضو  $q$  متعلق به مجموعه  $S_p$  متناظر با آن راه‌حل مشاهده شده و از رتبه مغلوب شدگی آن یک واحد کم می‌شود. در این حالت، هر عضو  $q$  که مقدار مغلوب شدگی ( $n_p$ ) آن صفر گردید، درون مجموعه جداگانه  $Q$  قرار داده می‌شود. این اعضا متعلق به دومین جبهه مغلوب نشده هستند. این روند با هر عضو درون  $Q$  ادامه یافته تا تمام جبهه‌های مغلوب نشده مشخص گردند.

##### ۴-۲- فاصله ازدحامی

برای تخمین چگالی جواب‌های اطراف یک جواب مشخص در جمعیت، میانگین فاصله دو نقطه در طرفین آن جواب در راستای هر بعد از اهداف،

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1), \quad (12)$$

که  $rand_i$  یک عدد تصادفی یکنواخت در بازه  $[0,1]$  است که برای ایجاد خاصیت تصادفی بودن برای جستجو، از آن استفاده می‌شود.

مقدار ثابت گرانش،  $G$ ، در ابتدا مقدار دهی اولیه می‌شود و با گذشت زمان از مقدار آن کاسته می‌شود. این امر برای کنترل درستی جستجو مفید است. به عبارت دیگر،  $G$  تابعی با مقدار اولیه  $G_0$  و زمان  $t$  است:

$$G(t) = G(G_0, t). \quad (13)$$

با توجه به رابطه ۷ در می‌یابیم که جرم سنگین‌تر به معنای عامل مؤثرتر است. این بدان معنی است که عوامل بهتر قدرت جذب بالاتری دارند و آهسته‌تر حرکت می‌کنند. با فرض تساوی جرم گرانشی و جرم اینرسی، مقادیر اجرام با نداشت شایستگی محاسبه می‌شود. جرم گرانشی و جرم اینرسی با روابط زیر به روز می‌شوند:

$$M_{ai} = M_{pi} = M_{ii} = M_i, \quad i=1,2,...,N, \quad (14)$$

$$M_i(t) = \frac{fit_i(t) - worst(t)}{\sum_{j=1}^N (fit_j(t) - worst(t))}, \quad (15)$$

$fit_i(t)$  نشان دهنده مقدار شایستگی عامل  $i$  در زمان  $t$  و  $worst(t)$  (برای مسأله بیشینه‌سازی) عبارت است از:

$$worst(t) = \min_{j \in \{1,...,N\}} fit_j(t). \quad (16)$$

برای اجتناب از افتادن در بهینه محلی، الگوریتم بایستی در ابتدا قدرت کاوش بالایی داشته باشد. با گذشت زمان به قدرت بهره‌وری افزوده و از قدرت کاوش کاسته می‌شود. برای بهبود عملکرد GSA در کنترل کاوش و بهره‌وری، تنها به  $Kbest$  عامل برتر (جرم بزرگتر) اجازه داده می‌شود که بر دیگران نیرو وارد کنند.  $Kbest$  تابعی از زمان است با مقدار اولیه  $K_0$  که با گذشت زمان کاهش می‌یابد.  $K_0$  به گونه‌ای تنظیم می‌شود که در شروع همه عوامل در اعمال نیرو به یکدیگر شرکت کنند. این پارامتر با گذشت زمان به طور خطی کاهش می‌یابد و در نهایت تنها یک عامل نیرویش را به دیگران اعمال خواهد کرد. بنابر این رابطه (۹) به صورت زیر اصلاح می‌شود:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(t), \quad (17)$$

مراحل مختلف الگوریتم عبارتند از:

۱. تعیین فضای جستجو
۲. مقدار دهی اولیه عامل‌ها به صورت تصادفی
۳. ارزیابی شایستگی عوامل

بدون نیاز به پارامتر اضافی که لازم باشد به وسیله کاربر تعریف شود، محاسبه می‌شود.

لازم به ذکر است که برای جواب‌هایی که دارای موقعیت یکسانی در فضای هدف هستند، مقدار فاصله ازدحامی آن‌ها یکسان در نظر گرفته نمی‌شود. برای تعیین مقدار فاصله ازدحامی چنین راه‌حل‌هایی ابتدا فاصله ازدحامی برای همه راه‌حل‌ها حساب می‌شود. کم‌ترین مقدار فاصله ازدحامی در هر جبهه به عنوان  $min_{dis}$  در نظر گرفته می‌شود. برای هر جبهه  $z$ ام به طور جداگانه، راه‌حل‌هایی که دارای موقعیت یکسان در فضای هدف هستند بر اساس مقدار فاصله ازدحامی‌شان به صورت نزولی مرتب می‌شوند. مقدار فاصله ازدحامی جدید برای این راه‌حل‌ها طبق روابط (۲۲ و ۲۳) محاسبه می‌شود.

$$CD_{new}(X_1) = \min_{dis} - \varepsilon, \quad (22)$$

$$CD_{new}(X_{i+1}) = CD_{new}(X_i) - \varepsilon, \quad (23)$$

For  $i=2,3,\dots,L_{new}-1$

در این روابط،  $CD_{new}(X_i)$  مقدار فاصله ازدحامی جدید برای  $i$  امین راه‌حل تکراری در  $z$  امین جبهه پس از مرتب سازی است. همچنین  $L_{new}$  تعداد این راه‌حل‌ها در جبهه  $z$ ام و  $\varepsilon$  مقدار مثبت کوچکی است. در صورتی که راه‌حل‌هایی وجود داشته باشند که بیش‌تر از دو بار در فضای هدف تکرار شوند، روند کاهش میزان فاصله ازدحامی آن‌ها مانند روند فوق مجدداً تکرار می‌شود.

### ۳-۴- تخصیص شایستگی

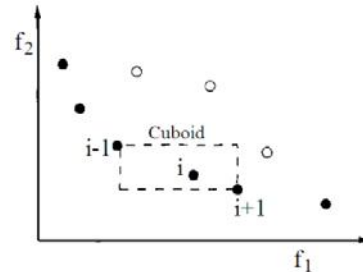
به هر عامل دو مقدار برای شایستگی اختصاص می‌دهیم. مقدار اول عبارت است از رتبه جبهه مغلوب نشده که هر عامل در آن قرار دارد. دومین مقدار، شامل فاصله ازدحامی است که هر عامل نسبت به سایر عوامل موجود در جبهه مغلوب نشده مربوط به خود دارد.

### ۴-۴- انتخاب Kbest

از میان جمعیت، تعداد  $kbest$  عامل از بهترین‌های جمعیت به عنوان مجموعه  $Kbest$  انتخاب می‌شوند. در شروع الگوریتم تعداد  $Kbest$  برابر تعداد کل جمعیت است که با افزایش تکرارها این تعداد به دو درصد از تعداد کل عامل‌های جمعیت کاهش می‌یابد.

برای انتخاب  $Kbest$  ابتدا اولین مقدار شایستگی هر عامل، یعنی شماره جبهه مغلوب نشده آن عامل مورد بررسی قرار می‌گیرد و عواملی که دارای کم‌ترین مقدار هستند، انتخاب می‌شوند تا  $kbest$  امین فرد نیز انتخاب گردد. اگر شماره جبهه  $kbest$  امین فرد با شماره جبهه  $kbest+1$

در فضای هدف محاسبه می‌شود. کمیت  $i_{distance}$  که به نام فاصله ازدحامی شناخته می‌شود در واقع به عنوان تخمینی از مکعب شکل یافته با نزدیکترین همسایگان آن به عنوان رئوسش می‌باشد. جواب با مقدار فاصله ازدحامی کم‌تر درون ناحیه شلوغ تری نسبت به سایر جواب‌هاست (شکل ۱).



شکل (۱): فاصله ازدحامی راه‌حل  $i$  ام [۸]

برای محاسبه فاصله ازدحامی، ابتدا جمعیت بر اساس هر مقدار تابع هدف و به صورت صعودی مرتب شده، سپس برای جواب‌های کناری هر تابع هدف (اولین و آخرین راه‌حل در هر بعد) مقدار فاصله معادل با بی‌نهایت (مقداری بزرگ) قرار داده می‌شود. برای سایر جواب‌های موجود میانی مقدار فاصله ازدحامی برابر با اختلاف نرمالیزه شده مقدار تابع هدف دو همسایه کناری آن در نظر گرفته می‌شود. مقدار فاصله ازدحامی نهایی برابر با مجموع مقدار فاصله ازدحامی به دست آمده در هر بعد می‌شود. برای مسأله‌ای با  $K$  تابع هدف که  $F_1, F_2, \dots, F_R$  جبهه‌های مغلوب نشده آن هستند فاصله ازدحامی بدین صورت محاسبه می‌شود:

$$CD_k(X[1,k]) = A; k = 1, 2, \dots, K \quad (18)$$

$$CD_k(X[L,k]) = A; k = 1, 2, \dots, K \quad (19)$$

$$\text{where } A \succ \max_{i=2,\dots,L-1} \{CD_k(X[i,k])\}$$

$L = |F_j|$ ، تعداد راه‌حل‌ها در جبهه  $z$ ام،  $X[i,k]$  بیانیگر  $i$  امین راه‌حل در لیست مرتب شده، با توجه به تابع هدف  $k$  ام است. برای  $i=1, 2, \dots, L-1$

$$CD_k(X[i,k]) = \frac{Z_k(X[i+1,k]) - Z_k(X[i-1,k])}{Z_k^{\max} - Z_k^{\min}} \quad (20)$$

$$CD(X) = \sum_K CD_k(X) \quad (21)$$

در رابطه (۲۰)،  $Z_k(X)$  بیانگر مقدار تابع هدف  $k$  ام به ازای ورودی یا راه حل  $X$  است. همچنین  $Z_k^{\min}$  و  $Z_k^{\max}$  به ترتیب مقادیر بیشینه و کمینه برای تابع هدف  $k$  ام را نشان می‌دهد. مهم‌ترین مزیت روش فاصله ازدحامی آن است که اندازه‌گیری چگالی جمعیت در اطراف یک جواب

جمعیت برابر ۱۰۰ و تعداد اعضای آرشیو خارجی برای SPEA2، TVMOPSO و MOGSA برابر با ۵۰ در نظر گرفته شده است. نرخ همبری برابر ۰/۹ و احتمال جهش برابر ۰/۱ قرار داده شده است [۶، ۸، ۲۰ و ۲۷]. همچنین به طور خاص برای TVMOPSO،  $c_{1i}=2.5$ ،  $c_{2i}=2.5$ ،  $w_1=0.7$ ،  $w_2=0.4$  و  $b=5$  می‌باشد [۲۰]. برای هر الگوریتم از بهترین پارامترهایی که ارائه شده، استفاده شده است.

#### ۱-۵- توابع محک استاندارد

برای آزمودن کارایی الگوریتم پیشنهادی، NF-MOGSA، و مقایسه آن با سایر روش‌ها از توابع محک استاندارد همچون SCH<sup>[۲۳]</sup> [۱۰]، KUR<sup>[۲۴]</sup> [۲۸]، FON<sup>[۲۵]</sup> [۲۹]، POL<sup>[۲۶]</sup> [۳۰]، و نیز توابع محک ZDT1، ZDT2، ZDT3 و ZDT6 [۸] استفاده شده است. این توابع، مجموعه راه‌حل‌های بهینه پرتو را شامل می‌شوند که به کمک آن‌ها می‌توان جنبه‌های مختلف کارایی الگوریتم را بررسی کرد.

#### ۲-۵- معیار کارایی

قبل از پرداختن به معیارهای کارایی، لازم است به این نکته اشاره شود که به طور کلی در طراحی الگوریتم‌ها برای بهینه‌سازی چند هدفه باید چند نکته مهم مد نظر قرار گیرد که عبارتند از [۳]: الف) فاصله جبهه مغلوب نشده تا جبهه بهینه پرتو، باید به حداقل برسد. ب) راه حل‌های پیدا شده باید دارای توزیع مناسبی باشند. ج) گستردگی جبهه مغلوب نشده نهایی، باید به حداکثر برسد. یعنی برای هر هدف، دامنه گسترده‌ای از مقادیر، باید تحت پوشش راه حل‌های مغلوب نشده قرار بگیرد. از این رو معیارهای کارایی باید به گونه‌ای انتخاب شوند که اهداف فوق الذکر را قابل سنجش نمایند. بر همین مبنا در این مقاله از سه معیار برای محاسبه کارایی الگوریتم پیشنهادی و مقایسه با سایر الگوریتم‌ها در حل مسائل بهینه‌سازی چندهدفه استفاده شده است. این معیارها قابلیت ارزیابی افراد مغلوب‌نشده در مقایسه با جبهه بهینه پرتو را بر حسب اهداف سه گانه فوق به ما می‌دهند.

الف) فاصله نسلی<sup>[۲۸]</sup> (GD): مفهوم فاصله نسلی به‌وسیله فان ولدویزن<sup>[۲۹]</sup> و لامونت<sup>[۳۰]</sup> بیان گردید [۳۱]. فاصله نسلی روشی برای تخمین فاصله اقلیدسی میان اعضای موجود در جبهه مغلوب‌نشده و نزدیک‌ترین عضو در جبهه بهینه پرتو است که بدین صورت تعریف می‌شود:

امین فرد مغایر با هم باشد فرآیند انتخاب  $Kbest$  پایان می‌پذیرد. در غیر اینصورت تمام عواملی که در جبهه مغلوب نشده یکسانی با عامل  $kbest$  ام هستند بر اساس فاصله ازدحامی آن‌ها انتخاب می‌شوند.

#### ۴-۵- مراحل مختلف الگوریتم پیشنهادی NFMOGSA

۱. ایجاد جمعیت اولیه  $N$  تایی به طور تصادفی
  ۲. تخصیص شایستگی به عوامل، با توجه به موقعیتشان در فضای هدف
  ۳. انتخاب  $kbest$  عامل از میان جمعیت برای اثر گذاری بر روی اعضای جمعیت
  ۴. محاسبه جرم اعضای  $Kbest$  (رابطه (۱۵))
  ۵. محاسبه شتاب افراد جمعیت (طبق رابطه (۲۵))
- $$a_i^d(t) = \sum_{j \in Kbest} G(t) \frac{M_j(t) \times (x_j^d(t) - x_i^d(t))}{R_{ij}(t) + \varepsilon} \times rand_j \quad (24)$$
۶. به روز رسانی سرعت و موقعیت عامل‌ها ( روابط (۱۱) و (۱۲))
  ۷. در نظر گرفتن موقعیت عامل‌ها قبل و پس از به روز رسانی و قرار دادن آن‌ها در مجموعه جدید  $R$
  ۸. تخصیص شایستگی به عوامل مجموعه  $R$  با توجه به موقعیت آن‌ها در فضای هدف
  ۹. انتخاب  $N$  عامل برتر از میان مجموعه  $R$  (مطابق با روش انتخاب  $Kbest$ )، برای تکرار بعدی
  ۱۰. اگر شرط توقف فرا رسیده عوامل به روز شده به عنوان جواب پایانی قرار بده، در غیر این صورت به مرحله ۲ برو.

#### ۵- ارزیابی و نتایج عملی

برای بررسی کارآمدی الگوریتم پیشنهادی، از چندین تابع محک استاندارد استفاده شده است که معمولاً در مباحث بهینه‌سازی چندهدفه به کار گرفته می‌شوند. این توابع دوهدفه بوده و هیچ قیدی بر روی آن‌ها نیست. از سوی دیگر نتایج الگوریتم پیشنهادی با الگوریتم‌های شناخته شده در بهینه‌سازی چندهدفه مانند SPEA2 [۶]، NSGA2 [۸]، TVMOPSO [۲۰] و همچنین MOGSA [۲۷] مقایسه می‌شود. برای مقایسه الگوریتم‌ها از مجموعه‌ای از معیارهای استاندارد که در این زمینه ارائه شده‌اند، استفاده می‌شود. در آزمایش‌های انجام شده، تعداد افراد

همانگونه که نتایج ارائه شده در جدول ۱ بیان می‌دارد، برای توابع محک SCH، FON و POL روش پیشنهادی در تمام معیارها، عملکردی عمدتاً بهتر و در بعضی موارد برابر نسبت به سایر روش‌ها داشته است.

از سوی دیگر، روش پیشنهادی برای تابع KUR از نظر فاصله نسلی ( $GD$ ) و معیار فاصله ( $SP$ ) عملکرد بهتری نسبت به روش‌های SPEA2، TVMOPSO و MOGSA داشته و عملکرد تقریباً برابری با روش NSGA2 داشته است به گونه ای که تفاوت این دو روش تنها به میزان ناچیز ۰/۰۰۱۹ و ۰/۰۰۱۱ به ترتیب برای این دو معیار بوده است. همچنین از لحاظ بیشینه گستردگی ( $MS$ )، عملکرد بسیار بهتری نسبت به روش‌های TVMOPSO و MOGSA به دست آورده است و تنها به میزان ۰/۰۰۴۲ و ۰/۰۰۲۶ عملکرد ضعیف‌تری نسبت به روش‌های NSGA2 و SPEA2 داشته است.

برای تابع محک ZDT1 روش پیشنهادی از نظر معیار فاصله ( $SP$ ) و فاصله نسلی ( $GD$ ) عملکرد بهتری نسبت به سایر روش‌ها داشته و از لحاظ معیار بیشینه گستردگی ( $MS$ ) به میزان ۰/۰۰۹۲ نسبت به روش‌های SPEA2، NSGA2 و ۰/۰۰۸۸ نسبت به روش TVMOPSO عملکرد نامناسب‌تری داشته است.

برای معیار فاصله نسلی ( $GD$ )، روش پیشنهادی NFMOGSA تنها در مقایسه با روش NSGA2 به میزان ۰/۰۰۰۳ عملکرد بدتری برای تابع محک ZDT2 داشته و این در حالی است که برای تابع محک ZDT3 روش پیشنهادی بهتر از سایر روش‌ها بوده است. همچنین برای تابع محک ZDT6 روش NFMOGSA به میزان ۰/۰۰۳۰، ۰/۰۰۲۷ و ۰/۰۰۱۱ ضعیف‌تر از روش‌های SPEA2، NSGA2 و TVMOPSO می‌باشد.

در خصوص معیار فاصله ( $SP$ )، برای توابع محک ZDT3، ZDT2 و ZDT6 همواره روش پیشنهادی بهتر بوده و تنها روش NSGA2 به ترتیب به میزان ۰/۰۰۰۶، ۰/۰۰۱۳ و ۰/۰۰۰۴ بهتر از روش پیشنهادی بوده است.

از نظر بیشینه گستردگی ( $MS$ )، روش پیشنهادی برای توابع محک ZDT1، ZDT2، ZDT3 و ZDT6 عملکرد بهتری نسبت به روش MOGSA داشته است. لیکن در مقایسه با روش‌های SPEA2 و NSGA2 به ترتیب به میزان ۰/۰۰۹۲، ۰/۰۰۷۲۰، ۰/۰۰۰۸ و ۰/۰۰۴۸۲ و در مقایسه با روش TVMOPSO به ترتیب به میزان ۰/۰۰۸۸، ۰/۰۰۰۶۷ و ۰/۰۰۴۸۲ عملکرد نامناسب‌تری از خود نشان داده است.

$$GD = \frac{\sqrt{\sum_{i=1}^l d_i^2}}{l}, \quad (25)$$

که  $l$  تعداد اعضای جبهه مغلوب‌نشده و  $d_i$  فاصله اقلیدسی بین  $i$  امین جواب مغلوب‌نشده و نزدیک‌ترین عضو مجموعه بهینه پرتو است. اگر  $GD=0$  باشد نشان‌دهنده آنست که تمام عناصر یافت شده در مجموعه مغلوب‌نشده متعلق به جبهه بهینه پرتو هستند.

ب) بیشینه گستردگی<sup>۳۱</sup> ( $MS$ ): این معیار بدین صورت اندازه‌گیری می‌شود:

$$MS = \sqrt{\frac{1}{K} \sum_{i=1}^K \left\{ \frac{\min(f_i^{\max}, F_i^{\max}) - \max(f_i^{\min}, F_i^{\min})}{F_i^{\max} - F_i^{\min}} \right\}^2} \quad (26)$$

که  $K$  تعداد اهداف،  $f_i^{\min}$  و  $f_i^{\max}$  مقادیر بیشینه و کمینه  $i$  امین هدف در مجموعه جواب‌های مغلوب‌نشده یافت شده است. همچنین  $F_i^{\min}$  و  $F_i^{\max}$  مقادیر بیشینه و کمینه  $i$  امین هدف در مجموعه بهینه پرتو می‌باشد [۳۲]. در صورتی که جبهه مغلوب‌نشده راه‌حل‌های مرزی در جبهه بهینه پرتو را شامل شود،  $MS=1$  می‌شود.

ج) معیار فاصله<sup>۳۳</sup> ( $SP$ ): هدف از معیار فاصله [۳۳]، تشخیص توزیع بردارها در سراسر مجموعه جواب‌های مغلوب‌نشده است.

$$s = \sqrt{\frac{1}{l-1} \sum_{i=1}^l (\bar{d} - d_i)^2} \quad (27)$$

$$d_i = \min_j \left( \sum_{k=1}^K |f_k^i - f_k^j| \right), i, j = 1, \dots, l \quad (28)$$

$l$  تعداد جواب‌های مغلوب‌نشده تولیدی به‌وسیله الگوریتم؛  $K$  تعداد اهداف و  $\bar{d}$  میانگین همه  $d_i$  ها می‌باشد. مقدار صفر نشان‌دهنده این است که همه جواب‌های مغلوب‌نشده یافت شده دارای فاصله یکسانی هستند.

### ۳-۵- نتایج آزمایش‌ها

نتایج به دست آمده از معیارهای کارایی مربوط به روش پیشنهادی و سایر روش‌های مطرح در بهینه‌سازی چندهدفه، بر روی توابع محک مختلف ZDT3، ZDT2، ZDT1، SCH، POL، FON، KUR، SPEA2 و ZDT6 در جدول (۱) ارائه شده است. همچنین نتایج مربوط به مقایسه جبهه پرتو به‌دست آمده به‌وسیله کلیه روش‌ها با جبهه پرتو واقعی مربوط به توابع محک POL و ZDT3 در شکل‌های (۲ و ۳) آورده شده است.



به دست آمده در تکرار جاری با راه‌حل‌های مربوط به تکرار قبل و انتخاب راه‌حل‌های شایسته‌تر استفاده شد. نتایج به دست آمده بروی توابع محک استاندارد ZDT3, ZDT2, ZDT1, FON, POL, KUR, SCH و ZDT6 و مقایسه راه‌حل‌های به دست آمده در فضای هدف با دیگر روش‌های بهینه‌سازی چندهدفه مطرح نشان‌دهنده کارایی روش پیشنهادی در حل مسائل بهینه‌سازی چندهدفه است. در کارهای آتی برای افزایش دقت الگوریتم بر نحوه انتخاب عوامل تاثیر گذار اجرام با استفاده از روش‌های خوشه‌بندی تمرکز خواهد شد.

### سپاسگزاری

برخود لازم می‌دانیم از داوران محترم به خاطر ارائه رهنمودهای ارزنده‌ی شان که باعث افزایش کیفیت مقاله شد، تشکر و قدردانی کنیم.

### مراجع

- [1] V. L. Huang, S. Z. Zhao, R. Mallipeddi and P. N. Suganthan, "Multi-objective Optimization Using Self-adaptive Differential Evolution Algorithm", IEEE Transactions On Evolutionary Computation, pp. 190 – 194, 2009.
- [2] C. M. Fonseca, P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization", International Journal of Evolutionary Computation, Vol. 3, pp. 1–16, 1995.
- [3] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications", Ph.D. thesis, Shaker Verlag, Aachen, Germany, 1999.
- [4] A. Konak, D. W. Coit and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial", Reliability Engineering and System Safety, Vol. 91, No. 9, pp. 992–1007, 2006.
- [5] E. Zitzler, M. Laumanns and S. Bleuler, "A Tutorial on Evolutionary Multiobjective Optimization", In: Gandibleux X., (eds.): Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems, Vol. 535, Springer, 2004.
- [6] E. Zitzler, M. Laumanns and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization", in Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), Int. Center Numerical Methods Eng. (CIMNE), pp. 95–100, 2002.
- [7] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach", IEEE Transactions On Evolutionary Computation, Vol. 3, No. 4, pp. 257–271, 1999.
- [8] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", IEEE

علاوه بر سه معیار فوق الذکر که از آن‌ها در مقایسه روش‌ها استفاده شد، زمان رسیدن به پاسخ نیز برای الگوریتم‌های مختلف به عنوان یک معیار محاسبه و در جدول (۱) گزارش شده است. در مقایسه الگوریتم‌ها به این نکته توجه شده است که تعداد ارزیابی توابع برای همه آن‌ها یکسان در نظر گرفته شود. اگر از این زاویه به موضوع پرداخته شود کلیه روش‌ها زمان تقریباً مساوی برای رسیدن به پاسخ دارند چون که اغلب زمان صرف محاسبه توابع هدف خواهد شد. این موضوع به خصوص در مسائل عملی مشهود خواهد بود زیرا که در این مسائل ارزیابی مدل مهندسی زمانبر است و زمان صرف شده توسط الگوریتم در سایر قسمت‌ها تقریباً قابل اغماض است. از این رو در اینجا برای مقایسه الگوریتم‌ها، متوسط زمانی روی اجراهای مختلف به گونه‌ای منظور شده است که الگوریتم به جبهه پرتو همگرا شده باشد. نتایج این مقایسه در جدول ۱ بر حسب ثانیه آمده است. بررسی نتایج نشان می‌دهد که در بین الگوریتم‌های مورد مقایسه الگوریتم پیشنهادی بعد از الگوریتم TVMOPSO در مقام دوم قرار دارد.

با توجه به نتایج به دست آمده، روش پیشنهادی نسبت به روش قبلی در زمینه بهینه‌سازی چندهدفه بر مبنای الگوریتم جستجوی گرانشی، MOGSA، در تمام موارد عملکرد بسیار بهتری داشته است. همچنین نسبت به روش TVMOPSO تنها از لحاظ بیشینه گستردگی ( $MS$ ) و برای توابع محک خانواده ZDT عملکرد بهتری به دست نیاورده ولی در سایر موارد بسیار کارآمدتر ظاهر شده است. در مقایسه با روش‌های SPEA2 و NSGA2 نیز، روش ارائه شده در اکثر موارد جوابی بهتر یا مساوی داده است. در مواردی هم که روش NFMOGSA نتوانسته بهتر از این دو روش یا سایر روش‌ها عمل کند میزان خطا بسیار ناچیز و در همه موارد به جز بیشینه گستردگی ( $MS$ ) و تنها برای تابع محک ZDT6 زیر 0.0093 بوده که قابل اغماض می‌باشد. بنابراین، می‌توان نتیجه گرفت که الگوریتم ارائه شده در حل مسائل چند هدفه موثر بوده و قابل رقابت با الگوریتم‌های مطرح در این زمینه است.

### ۶- نتیجه‌گیری و راه کارهای آتی

در این مقاله روشی بر اساس الگوریتم جستجوی گرانشی (GSA) برای بهینه‌سازی مسائل چندهدفه ارائه شده است. در این روش، استفاده از جبهه‌های مغلوب‌نشده برای تخصیص شایستگی و فاصله ازدحامی راه‌حل‌ها به عنوان معیاری برای ایجاد تنوع در بین راه‌حل‌های انتخابی پیشنهاد شده است. همچنین به منظور نخبه‌گرایی، از مقایسه راه‌حل‌های

- [22] Y. Wang and Y. Yang, "An interactive multi-swarm PSO for multiobjective optimization problems", *Expert Systems with Applications*, 2008.
- [23] Y. Wang and Y. Yang, "Particle swarm optimization with preference order ranking for multi-objective optimization", *Information Sciences*, Vol. 179, No.12, pp. 1944-1959, 2009.
- [24] H. Wang and F. Qian, "Improved PSO-based Multi-Objective Optimization using Inertia Weight and Acceleration Coefficients Dynamic Changing, Crowding and Mutation", *Proceedings of the 7th World Congress on Intelligent Control and Automation*, Chongqing, China, pp. 4479 - 4484, 2008.
- [25] A.Zhou, B.-Y.Qu, H.Li, S.-Z.Zhao, P.N.Suganthan, and Q.Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art", *Swarm and Evolutionary Computation*, vol. 1, No. 1, pp. 32-49, March 2011.
- [26] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "A gravitational search algorithm", *Information Science, Special Section on High Order Fuzzy Sets*, Vol. 179, No. 13, pp. 2232-2248, 2009.
- [۲۷] ز. بنی‌اسدی، ح.نظام‌آبادی‌پور و م. مغفوری فرسنگی، ارائه یک الگوریتم جستجوی گرانشی چند هدفه در حل مسائل عددی استاندارد و جایابی بهینه جبران کننده توان راکتیو، مجله علمی- پژوهشی سیستم‌های هوشمند در مهندسی برق، سال اول، شماره اول، صفحات ۵۹-۷۸، زمستان ۱۳۸۹.
- [28] F. Kursawe, "A variant of evolution strategies for vector optimization", in *Parallel Problem Solving from Nature*, Vol. 496, pp. 193-197, 1990.
- [29] C.M. Fonseca and P.J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms-Part II: Application example", *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 28, pp. 38-47, 1998.
- [30] C. Poloni, "Hybrid GA for multiobjective aerodynamic shape optimization", in *Genetic Algorithms in Engineering and Computer Science*, pp. 397-414, 1997.
- [31] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm research: A history and analysis", *Technical Report: TR-98-03*, 1998.
- [32] Tsung-Ying Sun and Sheng-Ta Hsieh, "Particle Swarm Optimizer for Multi-objective Problems based on Proportional Distribution and Cross-Over Operation", *IEEE International Conference on Systems, Man and Cybernetics*, pp. 2658-2663, 2008.
- [33] J.R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization", *Master's Thesis*, Cambridge, Massachusetts, 1995.
- [34] K. Deb and S. Jain, "Running Performance Metrics For Evolutionary Multi-Objective Optimization", *Technical Report KanGAL, Report Number 2002004*, Indian Institute of Technology Kanpur, India, May 2002.
- Transactions On Evolutionary Computation, Vol. 6, No. 2, pp. 182-197, 2002.
- [9] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms", in *genetic algorithms Evolutionary Computation*, Vol. 2, No. 3, pp. 221-248, 1994.
- [10] J.D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms", In: *Proceedings of the international conference on genetic algorithm and their applications*, pp. 93-100, 1985.
- [11] C. M. Fonseca and P. J. Fleming, "Multiobjective genetic algorithms", In: *IEE colloquium on 'Genetic Algorithms for Control Systems Engineering'*, (Digest No. 1993/130), London, UK: IEE, pp. 6/1-6/5, 1993.
- [12] T. Murata and H. Ishibuchi, "MOGA: multi-objective genetic algorithms", In: *Proceedings of the 1995 IEEE international conference on evolutionary computation*, Vol. 31, No. 8, pp. 1185-1192, 1995.
- [13] J. Horn, N. Nafpliotis and D. E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization", In *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE World Congress on Computational Computation, Vol. 1, pp. 82-87, 1994.
- [14] J. D. Knowles and D. W. Corne, "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimization", In *Congress on Evolutionary Computation (CEC99)*, Vol. 1, pp. 98-105, 1999.
- [15] H. Lu, G. Yen, "Rank-density-based multiobjective genetic algorithm and benchmark test function study", *IEEE Trans Evol Comput.*, Vol. 7, No. 4, pp. 325-43, 2003.
- [16] K. Praditwong, X. Yao, "A New Multi-objective Evolutionary Optimisation Algorithm: The Two-Archive Algorithm", *International Conference on Computational Intelligence and Security*, Vol. 1, pp. 286 - 291, 2006.
- [17] V. L. Huang, S. Z. Zhao, R. Mallipeddi and P. N. Suganthan, "Multi-objective Optimization Using Self-adaptive Differential Evolution Algorithm", In *Congress on Evolutionary Computation*, pp. 190 -194, 2009.
- [18] H. Liu, X. Li and Y. Chen, "Multi-objective Evolutionary Algorithm Based on Dynamical Crossover and Mutation Probability", *International Conference on Computational Intelligence and Security*, Vol. 1, pp. 150 - 155, 2008.
- [19] J. Kennedy and R.C. Eberhart, "Particle swarm optimization", in: *Proceedings of the Conference on Neural Networks*, IEEE Perth, Australia, pp. 1942-1948, 1995.
- [20] P. K. Tripathi , S. Bandyopadhyay and S. K. Pal, "Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients", *Information Sciences*, Vol. 177, pp. 5033-5049, 2007.
- [21] D. Lei, "A Pareto archive particle swarm optimization for multi-objective job shop scheduling", *Computers & Industrial Engineering*, Vol. 54, No. 4, pp. 960-971, 2008.

---

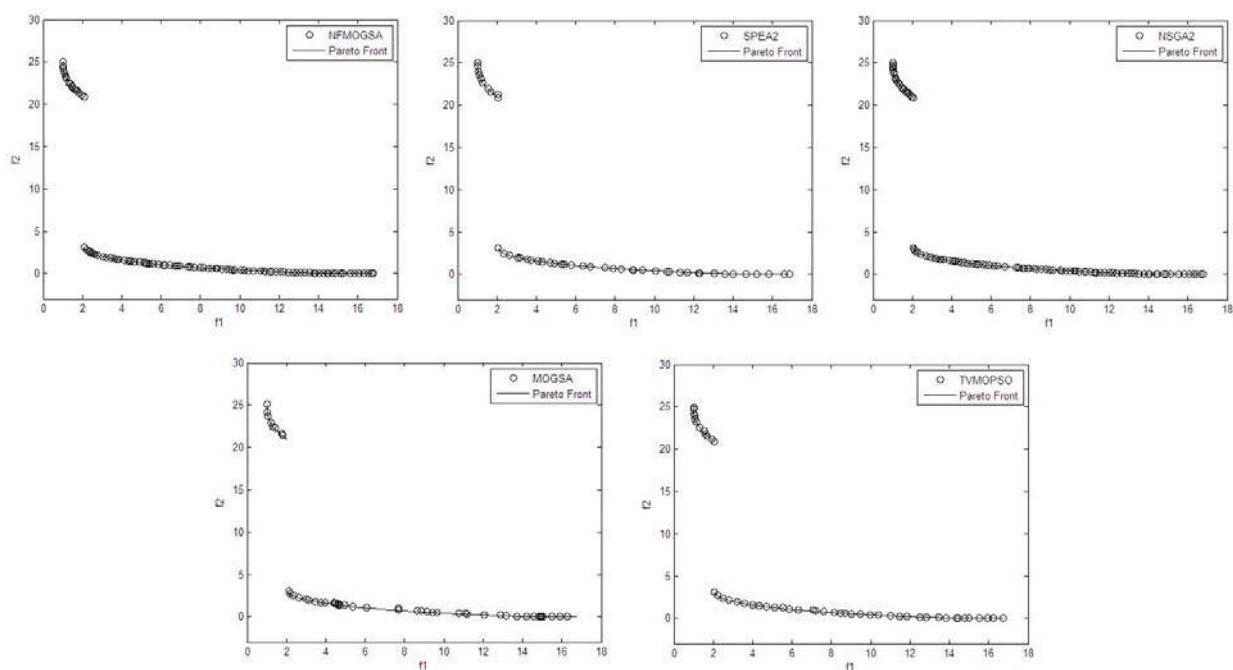
### زیر نویس ها

---

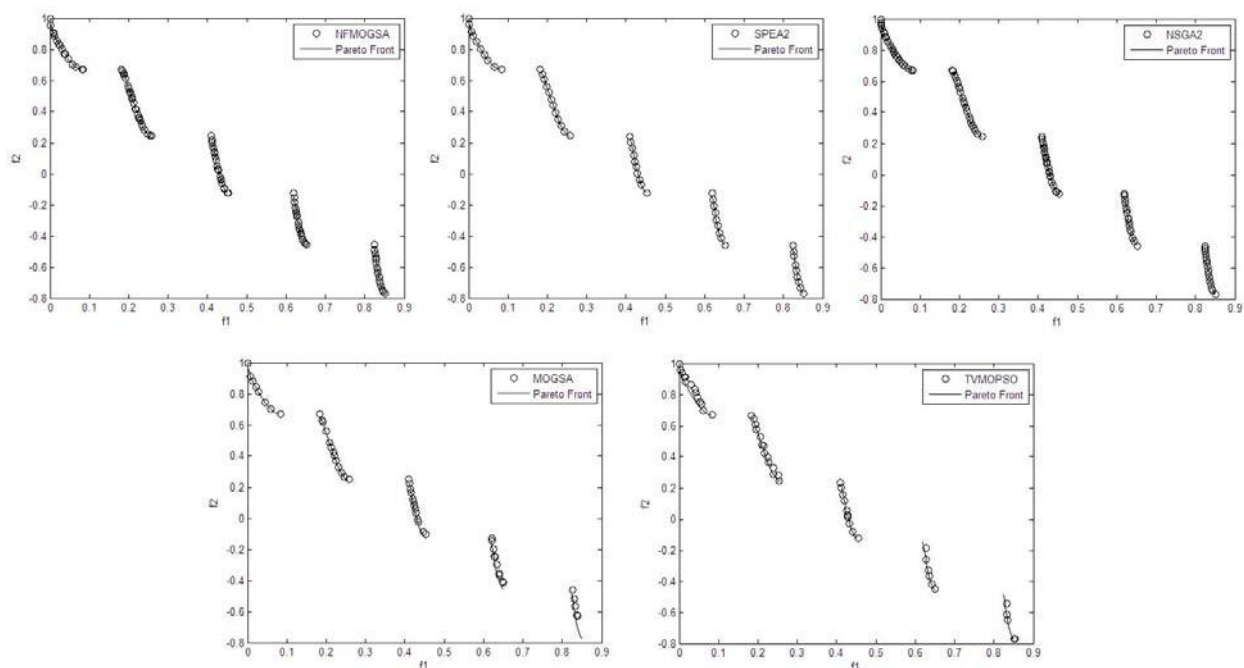
- <sup>1</sup> Conflicting objective
- <sup>2</sup> Search space
- <sup>3</sup> Gradient based
- <sup>4</sup> Simplex based
- <sup>5</sup> Simulated annealing
- <sup>6</sup> Single objective Problem (SOP)
- <sup>7</sup> Trade-off
- <sup>8</sup> Premature convergence
- <sup>9</sup> Ranking
- <sup>10</sup> Multi Objective Gravitational Search Algorithm (MOGSA)
- <sup>11</sup> MOGSA Using Non-dominated Front (NF-MOGSA)
- <sup>12</sup> Decision vector
- <sup>13</sup> Objective vector
- <sup>14</sup> Decision space
- <sup>15</sup> Objective space
- <sup>16</sup> Feasible set
- <sup>17</sup> Pareto dominance
- <sup>18</sup> a dominates b
- <sup>19</sup> a weakly dominates b
- <sup>20</sup> a is indifferent to b
- <sup>21</sup> Pareto optimality
- <sup>23</sup> Schaffer' s study
- <sup>24</sup> Kursawe's study
- <sup>25</sup> Fonseca and Fleming's study
- <sup>26</sup> Poloni's study
- <sup>28</sup> Generational Distance
- <sup>29</sup> Van Veldhuizen
- <sup>30</sup> Lamont
- <sup>31</sup> Maximum Spread
- <sup>32</sup> Spacing Metric

جدول (۱): معیارهای کارایی استفاده شده برای بررسی الگوریتم‌ها

|      |          | SPEA2  | NSGA2  | TVMOPSO | MOGSA    | NF-MOGSA |
|------|----------|--------|--------|---------|----------|----------|
| KUR  | GD       | ۰.۰۰۶۰ | ۰.۰۰۳۳ | ۰.۰۱۰۲  | ۰.۰۰۷۷   | ۰.۰۰۵۲   |
|      | SP       | ۰.۱۳۹۰ | ۰.۰۰۹۶ | ۰.۰۴۵۹  | ۰.۰۲۷۷   | ۰.۰۱۰۷   |
|      | MS       | ۰.۹۹۷۹ | ۰.۹۹۹۵ | ۰.۹۸۷۵  | ۰.۹۸۰۳   | ۰.۹۹۵۳   |
|      | Time (s) | ۹۶.۸۵  | ۵۸.۲۸  | ۱۰.۷۳   | ۷۵.۶۸    | ۵۲.۲۸    |
| SCH  | GD       | ۰.۰۰۱۳ | ۰.۰۰۰۹ | ۰.۰۰۱۲  | ۰.۰۰۱۴   | ۰.۰۰۰۹   |
|      | SP       | ۰.۰۷۰۴ | ۰.۰۰۷۴ | ۰.۰۱۷۸  | ۰.۰۸۳۶   | ۰.۰۰۵۳   |
|      | MS       | ۱.۰۰۰۰ | ۱.۰۰۰۰ | ۱.۰۰۰۰  | ۰.۹۸۱۸   | ۱.۰۰۰۰   |
|      | Time (s) | ۷۳.۰۰  | ۶۲.۲۵  | ۲۲.۸۴   | ۱۵۲.۱۲   | ۶۱.۳۱    |
| FON  | GD       | ۰.۰۰۰۴ | ۰.۰۰۰۲ | ۰.۰۰۰۴  | ۰.۰۰۰۱   | ۰.۰۰۰۱   |
|      | SP       | ۰.۰۱۴۵ | ۰.۰۰۱۲ | ۰.۰۰۱۳  | ۰.۰۰۳۳   | ۰.۰۰۰۸   |
|      | MS       | ۱.۰۰۰۰ | ۱.۰۰۰۰ | ۰.۹۹۹۶  | ۰.۹۹۹۹   | ۱.۰۰۰۰   |
|      | Time (s) | ۱۱۹.۴۲ | ۶۰.۴۵  | ۱۰۱.۱۴  | ۱۲۸.۱۷   | ۶۰.۲۸    |
| POL  | GD       | ۱.۴۹۲۴ | ۱.۰۳۹۰ | ۱.۴۶۹۲  | ۱.۴۴۲۹   | ۱.۰۳۰۷   |
|      | SP       | ۰.۲۱۳۰ | ۰.۰۲۴۷ | ۰.۰۷۰۹  | ۰.۰۴۱۵   | ۰.۰۱۵۷   |
|      | MS       | ۱.۰۰۰۰ | ۰.۹۹۹۵ | ۰.۹۹۸۵  | ۰.۹۸۴۶   | ۱.۰۰۰۰   |
|      | Time (s) | ۹۱.۸۴  | ۶۱.۱۵  | ۲۸.۵۰   | ۹۴.۶۸    | ۵۷.۲۰    |
| ZDT1 | GD       | ۰.۰۰۰۶ | ۰.۰۰۰۵ | ۰.۰۰۱۳  | ۰.۰۰۲۰   | ۰.۰۰۰۵   |
|      | SP       | ۰.۰۱۵۶ | ۰.۰۰۰۷ | ۰.۰۰۴۱  | ۰.۰۰۷۶   | ۰.۰۰۰۵   |
|      | MS       | ۱.۰۰۰۰ | ۱.۰۰۰۰ | ۰.۹۹۹۶  | ۰.۹۸۴۴   | ۰.۹۹۰۸   |
|      | Time (s) | ۲۰۹.۳۱ | ۶۴.۹۶  | ۱۲۸.۰۳  | ۹۹.۴۸    | ۵۹.۴۵    |
| ZDT2 | GD       | ۰.۰۰۰۷ | ۰.۰۰۰۴ | ۰.۰۰۱۴  | ۰.۰۰۱۶   | ۰.۰۰۰۷   |
|      | SP       | ۰.۰۱۶۷ | ۰.۰۰۰۱ | ۰.۰۰۲۶  | ۰.۰۰۳۳   | ۰.۰۰۰۷   |
|      | MS       | ۱.۰۰۰۰ | ۱.۰۰۰۰ | ۰.۹۹۹۵  | ۰.۹۵۴۸   | ۰.۹۹۲۸   |
|      | Time (s) | ۵۲۵.۰۰ | ۶۵.۵۷  | ۱۹۱.۱۵  | ۲۸۰.۷۵   | ۵۷.۶۸    |
| ZDT3 | GD       | ۰.۰۰۱۲ | ۰.۰۰۰۷ | ۰.۰۰۱۲  | ۰.۰۰۱۴   | ۰.۰۰۰۷   |
|      | SP       | ۰.۰۰۵۹ | ۰.۰۰۰۲ | ۰.۰۰۹۴  | ۰.۰۱۳۶   | ۰.۰۰۱۵   |
|      | MS       | ۱.۰۰۰۰ | ۱.۰۰۰۰ | ۱.۰۰۰۰  | ۰.۹۵۱۳   | ۰.۹۹۹۲   |
|      | Time (s) | ۲۰۸.۶۷ | ۷۶.۵۴  | ۱۲۴.۴۸  | ۳۶۸.۹۳   | ۱۱۵.۴۲   |
| ZDT6 | GD       | ۰.۰۰۰۵ | ۰.۰۰۰۸ | ۰.۰۰۲۴  | ۰.۰۰۴۸   | ۰.۰۰۳۵   |
|      | SP       | ۰.۰۱۳۲ | ۰.۰۰۰۳ | ۰.۰۰۲۳  | ۰.۰۰۸۱   | ۰.۰۰۰۷   |
|      | MS       | ۱.۰۰۰۰ | ۱.۰۰۰۰ | ۱.۰۰۰۰  | ۰.۹۴۵۲   | ۰.۹۵۱۸   |
|      | Time (s) | ۴۲۲.۲۹ | ۴۷۲.۷۰ | ۳۹۵.۰۰  | ۱۸۲.۲۰۳۱ | ۱۹۷.۱۷   |



شکل (۲): نتایج به دست آمده به وسیله روش‌های SPEA2، NSGA2، MOGSA، و NF-MOGSA بر روی تابع محک POL



شکل (۳): نتایج به دست آمده به وسیله روش‌های SPEA2، NSGA2، MOGSA، و NF-MOGSA بر روی تابع محک ZDT3