

## دسته بندی اهداف سوناری با استفاده از روش ترکیبی ازدحام ذرات و جستجوی گرانشی

سید محمدرضا موسوی<sup>۱</sup>، محمد خویشه<sup>۲</sup>، آلاوه مریدی<sup>۳</sup>، محمدجعفر ناصری<sup>۴</sup>

m\_mosavi@iust.ac.ir

۱- استاد دانشکده برق، دانشگاه علم و صنعت ایران

۲- دانشجوی دکتری برق، دانشگاه علم و صنعت ایران

۳- دانشجوی کارشناسی ارشد برق، دانشگاه علم و صنعت ایران

۴- دانشجوی کارشناسی ارشد برق، دانشگاه علوم دریایی امام خمینی(ره)

### چکیده

با توجه به ویژگی‌های فیزیکی بسیار نزدیک اهداف واقعی و کلاتر سونار فعال، تفکیک این اهداف، از موضوعات چالش برانگیز محققان و صنعت‌گران حوزه آکوستیک است. شبکه‌های عصبی چندلایه (MLP) یکی از پرکاربردترین شبکه‌های عصبی در دسته‌بندی اهداف دنیای واقعی هستند. آموزش از مهمترین بخش‌های توسعه این نوع شبکه‌ها است که در سال‌های اخیر بسیار مورد توجه قرار گرفته است. به منظور آموزش شبکه‌های MLP از دیر باز استفاده از روش‌های بازگشتی و گرادیان نزولی مرسوم بوده است. دقت دسته‌بندی نامناسب، گیر افتادن در کمینه‌های محلی و سرعت همگرایی پایین از معایب روش‌های سنتی است. به منظور غلبه بر این معایب، در سال‌های اخیر استفاده از الگوریتم‌های ابتکاری و فرا ابتکاری بسیار مرسوم گردیده است. این مقاله برای آموزش شبکه MLP از الگوریتم بهینه‌سازی ترکیبی ازدحام ذرات و جستجوی گرانشی (PSOGSA) استفاده می‌کند. الگوریتم جستجوی گرانشی (GSA) یک روش بهینه‌سازی فرا ابتکاری جدید بر اساس فعل و انفعالات گرانش و جرم است. ثابت شده است که این الگوریتم توانایی خوبی برای جستجوی کلی دارد، اما در آخرین تکرارها دارای سرعت پایین در بهره‌برداری فضای جستجو است. با توجه به توانایی منحصر به فرد بهینه‌سازی ازدحام ذرات (PSO) در فاز بهره‌برداری، از این روش برای حل مشکل فوق استفاده می‌شود. نتایج به دست آمده نشان می‌دهد که دسته‌بندی‌کننده‌های مبتنی بر PSO، GSA و PSOGSA دادگان سونار را به ترتیب با دقت ۹۲/۷۵۰، ۹۳/۶۷۴۱ و ۹۴/۴۲۳۰۸ دسته‌بندی می‌نمایند. همچنین سرعت همگرایی الگوریتم ترکیبی نسبت به دو الگوریتم معیار ذکر شده بهتر است.

واژگان کلیدی: دسته‌بندی، سونار، کلاتر، بهینه‌سازی ازدحام ذرات، الگوریتم جستجوی گرانشی.

تاریخ دریافت مقاله: ۹۴/۰۱/۱۸

تاریخ پذیرش مقاله: ۹۴/۱۰/۲۵

## ۱- مقدمه

شبکه‌های عصبی چندلایه یکی از پرکاربردترین ابزار برای محاسبات نرم می‌باشند. با استفاده از این شبکه‌ها می‌توان مسائل غیرخطی را حل نمود. به‌طور کلی، شبکه‌های عصبی مصنوعی چندلایه به‌منظور دسته‌بندی الگو، پیش‌بینی داده و تقریب زدن توابع مورد استفاده قرار می‌گیرند [۱-۴]. دسته‌بندی الگو به معنی تقسیم‌بندی داده‌ها به دسته‌های گسسته از پیش تعریف شده است [۵]، در حالی که پیش‌بینی به معنای پیش‌بینی اتفاقات آینده بر اساس داده‌های گذشته و داده‌های کنونی است [۶]. در نهایت تقریب تابع، فرآیند مدل کردن روابط بین متغیرهای ورودی را شامل می‌شود. ثابت شده است که شبکه‌های عصبی چندلایه با یک لایه پنهان می‌توانند هر تابع گسسته و یا پیوسته را تقریب بزنند [۷]. صرف نظر از کاربردها، توانایی متمایز شبکه‌های عصبی مصنوعی چندلایه، یادگیری است [۳]. یادگیری به این معنی است که این شبکه‌ها همانند مغز انسان می‌توانند از یک تجربه یا آزمایش یاد بگیرند. این ویژگی (یادگیری) بخش ضروری همه شبکه‌های عصبی است که ممکن است به دو نوع تقسیم گردد: یادگیری با نظارت<sup>۱</sup> [۸] و یادگیری بدون نظارت<sup>۲</sup> [۹]. برای شبکه‌های عصبی مصنوعی چندلایه (در بیشتر کاربردها)، از الگوریتم‌های پس‌انتشار<sup>۳</sup> بهینه‌شده [۱۰] و یا استاندارد [۱۱]، به‌عنوان روش یادگیری استفاده می‌گردد که از خانواده یادگیری با نظارت می‌باشند. الگوریتم پس‌انتشار، بر مبنای گرادیان است که اشکالاتی همچون همگرایی آهسته [۱۲] و بکارگیری در یک محدوده کوچک [۱۳] را دارد و بنابراین برای کاربردهای عملی قابل اعتماد نیست.

هدف نهایی فرآیند یادگیری در شبکه‌های عصبی، پیدا کردن بهترین ترکیب از یال‌های وزن‌دار و بایاس<sup>۴</sup> آن‌ها است، به طوری که در آموزش شبکه و نمونه‌های آزمون، کمترین مقدار خطا را داشته باشیم. به‌رحال اغلب خطای شبکه عصبی چندلایه، برای مدت زیادی در زمان فرآیند یادگیری، بزرگ خواهد بود و الگوریتم یادگیری آن را به سمت کم شدن

هدایت می‌کند. این مسئله در فرآیندهای یادگیری مبتنی بر گرادیان، مثل الگوریتم پس‌انتشار، کاملاً مشترک است. همچنین همگرایی الگوریتم پس‌انتشار به مقدار خیلی زیاد به مقادیر اولیه نرخ یادگیری و اندازه حرکت وابسته است. مقادیر نامناسب این متغیرها می‌تواند حتی سبب واگرایی الگوریتم گردد. مطالعات بسیار زیادی برای حل این مشکل الگوریتم پس‌انتشار انجام گرفته است [۱۴]. اما بهینه‌سازی کافی بدست نیامده و هر روش فقط آثار جانبی خودش را دارد. این مقاله نشان می‌دهد که الگوریتم جستجوی ابتکاری<sup>۵</sup> و فرا ابتکاری<sup>۶</sup> می‌تواند جایگزین الگوریتم‌های یادگیری مبتنی بر گرادیان باشد [۱۵]. زیرا ماهیت تصادفی این الگوریتم‌ها اجازه می‌دهد تا درصد خطای کمتری نسبت به روش‌های مبتنی بر گرادیان داشته باشیم و همچنین احتمال گیر افتادن در کمینه‌های محلی کاهش یابد.

روش‌های بهینه‌سازی ابتکاری مختلفی برای آموزش شبکه‌های عصبی چندلایه استفاده شده است. از جمله آن‌ها می‌توان به الگوریتم تبرید تدریجی (SA) [۱۶ و ۱۷]، الگوریتم ژنتیک (GA)<sup>۸</sup> [۱۸]، بهینه‌ساز ازدحام ذرات<sup>۹</sup> (PSO) [۱۹-۲۳]، الگوریتم بهینه‌سازی مغناطیسی<sup>۱۰</sup> [۲۴]، بهینه‌ساز مبتنی بر جغرافیایی زیستی (BBO)<sup>۱۱</sup> [۱]، بهینه‌ساز گرگ خاکستری (GWO)<sup>۱۲</sup> [۲۵] و دیفرانسیل تکاملی<sup>۱۳</sup> [۲۶] اشاره کرد. با توجه به [۲۷]، برخی الگوریتم‌ها الگوریتم‌ها مانند SA و GA می‌توانند احتمال به دام افتادن در کمینه محلی را کاهش دهند، اما آن‌ها هنوز دارای نرخ همگرایی آهسته هستند. با توجه به [۱۹-۲۴]، بهینه‌ساز ازدحام ذرات یکی از کارآمدترین و عملی‌ترین الگوریتم‌های بهینه‌سازی در افزایش نرخ همگرایی است. این مقاله، به‌وسیله ترکیب الگوریتم‌های PSO و GSA سعی در غلبه بر دو مشکل ذکر شده دارد.

ادامه مقاله به شرح زیر است. بخش‌های ۲ تا ۴ به ترتیب به ارائه مقدمه‌های کوتاه درباره مفاهیم اولیه بهینه‌ساز ازدحام ذرات، الگوریتم جستجوی گرانشی و ترکیبی از این دو الگوریتم با نام (PSOGSA) می‌پردازد. بخش ۵ روش استفاده

<sup>8</sup> Genetic Algorithm (GA)

<sup>9</sup> Particle Swarm Optimization (PSO)

<sup>10</sup> Magnetic Optimization Algorithm (MOA)

<sup>11</sup> Biogeography-Based Optimizer

<sup>12</sup> Gray Wolf Optimizer

<sup>13</sup> Differential Evolution (DE)

<sup>1</sup> Supervised Learning

<sup>2</sup> Unsupervised Learning

<sup>3</sup> Back-Propagation (BP) Algorithm

<sup>4</sup> Bias

<sup>5</sup> Heuristic Optimization Methods

<sup>6</sup> Meta-Heuristic Optimization Methods

<sup>7</sup> Simulated Annealing (SA)

ذرات در یک فضای مسئله شروع می‌کند. در هر تکرار سرعت ذرات با استفاده از رابطه (۱) محاسبه می‌شود. بعد از محاسبه سرعت ذرات، موقعیت جدید ذرات با استفاده از رابطه (۲) به دست می‌آید. فرآیند تغییر موقعیت ذرات تا دستیابی به معیار نهایی ادامه دارد.

### ۳- الگوریتم جستجوی گرانشی

در سال ۲۰۰۹ راشدی و همکارانش [۲۹] یک الگوریتم ابتکاری جدید به نام الگوریتم جستجوی گرانشی (GSA) پیشنهاد کردند. هدف این الگوریتم پیدا کردن بهترین راه حل در فضای جستجوی مسئله با استفاده از قوانین فیزیکی است. نظریه فیزیکی که الگوریتم جستجوی گرانشی از آن الهام گرفته شده است، تئوری نیوتن است [۳۰] که می‌گوید: "هر ذره در جهان ذره دیگر را با یک نیرویی که به طور مستقیم با حاصل ضرب جرم آن دو ذره و به طور معکوس با مجذور فاصله بین آنها رابطه دارد، جذب می‌کند". الگوریتم جستجوی گرانشی را می‌توان به عنوان مجموعه‌ای از عوامل (راه حل‌های نامزد) در نظر گرفت که دارای جرم متناسب با مقدار خود در تابع شایستگی<sup>۵</sup> هستند. در طول اجرای الگوریتم تمام توده‌ها توسط نیروی جاذبه بین خودشان یکدیگر را جذب می‌کنند. هر چه جرم سنگین‌تر باشد، نیروی جاذبه بیشتر است. بنابراین، جرم‌های سنگین‌تر که احتمالاً نزدیک‌ترین جرم‌ها به کمینه کلی هستند، جرم‌های دیگر را متناسب با فاصله آنها جذب می‌نمایند. با توجه به [۳۱]، فرض کنید یک سیستم با  $N$  عامل وجود دارد. موقعیت هر عامل (جرم) که یک نامزد راه حل مسئله است، به صورت رابطه (۳) تعریف می‌شود:

$$X_i = (\chi_i^1, \dots, \chi_i^d, \dots, \chi_i^n), \quad i = 1, 2, \dots, N \quad (3)$$

که در آن،  $N$  مبین بعد مسئله و  $\chi_i^d$  مبین موقعیت عامل  $i$ -ام در بعد  $d$ -ام است. الگوریتم کار خود را با قرار دادن عوامل به طور تصادفی در فضای جستجو شروع می‌کند. در طول هر تکرار، نیروی گرانشی از عامل  $j$ -ام بر عامل  $i$ -ام در زمان خاص  $t$  به صورت رابطه (۴) تعریف می‌شود:

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \epsilon} \times (\chi_j^d(t) - \chi_i^d(t)) \quad (4)$$

از الگوریتم‌های ذکر شده را به عنوان الگوریتم‌های آموزشی شبکه‌های عصبی چندلایه مورد بحث قرار می‌دهد. نتایج آزمایشات و شبیه‌سازی‌ها در بخش ۶ نشان داده شده‌اند. در نهایت بخش ۷ نتیجه‌گیری مقاله را ارائه می‌دهد.

### ۲- بهینه‌ساز ازدحام ذرات (PSO)

بهینه‌ساز ازدحام ذرات یک روش محاسباتی تکاملی است که توسط کندی و ابره‌ارت<sup>۱</sup> [۲۸] ارائه شده است. بهینه‌ساز ازدحام ذرات از رفتار اجتماعی پرندگان الهام گرفته شده است. این روش از تعدادی از ذرات (راه حل‌های نامزد) که در فضای جستجو به دنبال یافتن بهترین راه حل هستند، استفاده می‌کند. در همین حال همه ذرات به سوی بهترین ذره‌ای (بهترین راه حل) که در مسیرشان وجود دارد، حرکت می‌کنند. به عبارت دیگر، ذرات بهترین راه حل‌های خود را به عنوان بهترین راه حل کلی در نظر می‌گیرند.

هر ذره در بهینه‌ساز ازدحام ذرات باید پارامترهایی مانند: موقعیت فعلی، سرعت فعلی، فاصله تا  $pbest$  و فاصله تا  $gbest$  را به منظور تغییر موقعیت خود در نظر بگیرد. مدل‌سازی ریاضی بهینه‌ساز ازدحام ذرات به صورت روابط (۱) و (۲) است:

$$v_i^{t+1} = wv_i^t + c_1 \times rand \times (pbest_i - \chi_i^t) \quad (1)$$

$$+ c_2 \times rand \times (gbest - \chi_i^t)$$

$$\chi_i^{t+1} = \chi_i^t + v_i^{t+1} \quad (2)$$

در روابط بالا،  $V_i^t$  مبین سرعت<sup>۲</sup> ذره  $i$ -ام در تکرار  $t$ -ام،  $w$  مبین تابع وزن<sup>۳</sup>،  $c_j$  مبین ضریب شتاب<sup>۴</sup> و  $rand$  نشان‌دهنده یک عدد تصادفی بین ۰ و ۱ است.  $\chi_i^t$  مبین موقعیت فعلی ذره  $i$ -ام در تکرار  $t$ -ام است. همچنین  $pbest_i$ ، بهترین راه حل عامل  $i$ -ام در تکرار  $t$ -ام است و بهترین راه‌حلی که تا کنون پیدا شده  $gbest$  است.

بخش اول رابطه (۱)،  $(wV_i^t)$ ، توانایی اکتشاف بهینه‌ساز ازدحام ذرات را فراهم می‌کند. بخش دوم و سوم به ترتیب نشان‌دهنده تفکر خصوصی و همکاری بین ذرات است. بهینه‌ساز ازدحام ذرات کار خود را با قرار دادن تصادفی

<sup>4</sup> Acceleration Coefficient

<sup>5</sup> Fitness Function

<sup>1</sup> Kennedy and Eberhart

<sup>2</sup> Velocity

<sup>3</sup> Weighting Function

یک عامل سنگین تر باشد، نیروی جاذبه بیشتر و حرکت آن آهسته تر است. جاذبه بیشتر را با توجه به قانون جاذبه بیان شده توسط رابطه (۴) و جنبش آهسته تر را با توجه قانون حرکت و رابطه (۸) می توان تعریف نمود [۲۹]. جرم تمام عوامل با استفاده از روابط زیر به روزرسانی می شود:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (11)$$

که در آن،  $fit_i(t)$  مبین ارزش شایستگی عامل  $i$ -ام در زمان  $t$   $best(t)$  مبین قوی ترین عامل در زمان  $t$  و  $worst(t)$  مبین ضعیف ترین عامل در زمان  $t$  است.  $worst(t)$  برای مسئله حداقل سازی به صورت روابط (۱۲) و (۱۳) محاسبه می شوند:

$$best(t) = \min_{j \in \{1, \dots, N\}} fit_j(t) \quad (12)$$

$$worst(t) = \max_{j \in \{1, \dots, N\}} fit_j(t) \quad (13)$$

$best(t)$  و  $worst(t)$  برای مسئله حداکثر سازی به صورت روابط (۱۴) و (۱۵) محاسبه می گردند:

$$best(t) = \max_{j \in \{1, \dots, N\}} fit_j(t) \quad (14)$$

$$worst(t) = \min_{j \in \{1, \dots, N\}} fit_j(t) \quad (15)$$

نرمال سازی جرم محاسبه شده توسط رابطه (۱۱) به صورت زیر تعریف می شود:

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (16)$$

در الگوریتم جستجوی گرانشی هر عامل یک راه حل نامزد تلقی می شود. در این الگوریتم ابتدا تمام عوامل با مقادیر تصادفی مقداردهی اولیه می گردند. پس از مقداردهی اولیه، سرعت و موقعیت همه عوامل با استفاده از روابط (۹) و (۱۱) تعریف می شوند. همچنین، پارامترهای دیگر مانند ثابت گرانش و جرم توسط روابط (۵) و (۱۱) محاسبه می گردند. در نهایت الگوریتم جستجوی گرانشی با دستیابی به معیار نهایی متوقف می شود. مراحل الگوریتم جستجوی گرانشی در شکل (۱) نشان داده شده است.

در تمام الگوریتم های مبتنی بر جمعیت که دارای رفتار اجتماعی هستند، مانند PSO و GSA، دو ویژگی ذاتی باید در نظر گرفته شود: توانایی الگوریتم برای جستجوی تمام بخش های فضای جستجو و توانایی اش در بهره برداری از بهترین راه حل.

که در آن،  $M_{ij}$  مبین جرم گرانشی فعال مربوط به عامل  $j$ -ام،  $M_{pi}$  مبین جرم گرانشی غیرفعال مربوط به عامل  $i$ -ام،  $G(t)$  مبین ثابت گرانشی در زمان  $t$ ،  $\epsilon$  یک ثابت کوچک و  $R_{ij}(t)$  مبین فاصله اقلیدسی بین دو عامل  $i$  و  $j$  است. ثابت گرانش  $G$  و فاصله اقلیدسی بین دو عامل  $i$ -ام و  $j$ -ام به ترتیب و به صورت روابط (۵) و (۶) محاسبه می شوند:

$$G(t) = G_0 \times \exp(-\alpha \times iter / \max iter) \quad (5)$$

$$R_{ij}(t) = \|X_i(t) - X_j(t)\|_2 \quad (6)$$

که در آن،  $\alpha$  مبین ضریب نزولی،  $G_0$  مبین ثابت گرانش اولیه،  $iter$  مبین تعداد تکرار فعلی و  $\max iter$  نشان دهنده حداکثر تعداد تکرار است. در یک فضای مسئله  $d$  بعدی، نیروی کلی که بر عامل  $i$ -ام اعمال می شود از رابطه (۷) محاسبه می گردد:

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad (7)$$

که در آن،  $rand_j$  نشان دهنده یک عدد تصادفی در بازه  $[0,1]$  است. با توجه به قانون حرکت، شتاب یک عامل به طور مستقیم با نیروی برآیند و به طور معکوس با جرم آن رابطه دارد. بنابراین شتاب همه عوامل به شرح زیر محاسبه می شود:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (8)$$

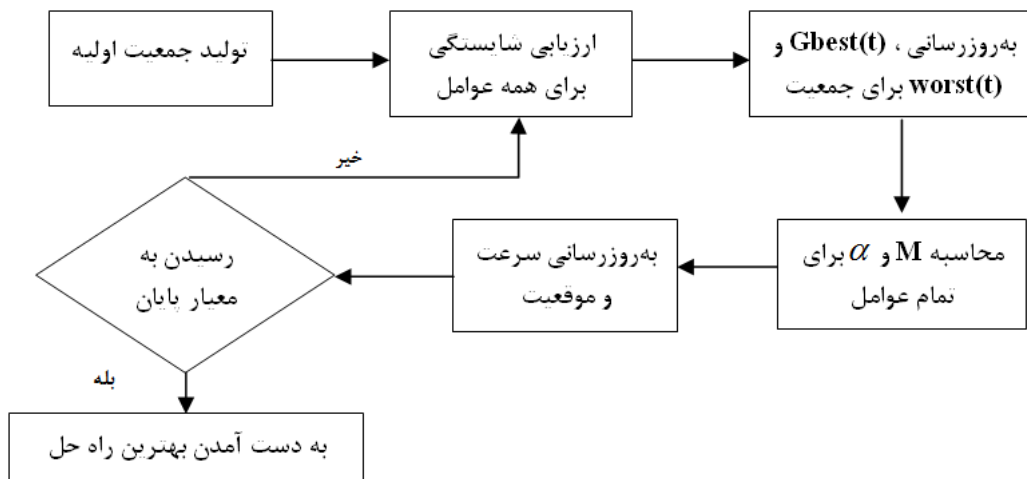
که در آن،  $d$  مبین بعد مسئله،  $t$  نشان دهنده یک زمان خاص و  $M_{ii}$  مبین جرم جسم  $i$ -ام است. سرعت و موقعیت عوامل به صورت روابط (۹) و (۱۰) محاسبه می شوند:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (9)$$

$$\chi_i^d(t+1) = \chi_i^d(t) + v_i^d(t+1) \quad (10)$$

که در آن،  $d$  مبین بعد مسئله و  $rand_i$  نشان دهنده یک عدد تصادفی در بازه  $[0,1]$  است. می توان از روابط (۹) و (۱۰) چنین استنباط کرد که سرعت فعلی یک عامل به صورت کسری از آخرین سرعت که به شتاب آن اضافه می شود، تعریف می گردد. علاوه بر این، موقعیت فعلی یک عامل برابر است با آخرین موقعیت خودش که به سرعت کنونی آن اضافه شده است.

جرم عوامل با استفاده از ارزیابی تابع شایستگی تعریف می شود. این بدین معنی است که یک عامل با سنگین ترین جرم کارآمدترین عامل است. با توجه به معادلات بالا، هر چه



شکل (۱) مراحل عمومی الگوریتم جستجوی گرانشی [۳۰].

اجتماعی ( $gbest$ ) موجود در بهینه‌ساز ازدحام ذرات را با قابلیت جستجوی محلی الگوریتم جستجوی گرانشی ترکیب می‌کند. به منظور ترکیب این الگوریتم‌ها، رابطه (۱۷) به شرح زیر پیشنهاد شده است:

$$V_i(t+1) = w \times V_i(t) + c'_1 \times rand \times ac_i(t) + c'_2 \times rand \times (gbest - X_i(t)) \quad (17)$$

که در آن،  $V_i(t)$  مبین سرعت عامل  $i$ -ام در تکرار  $t$ -ام،  $c'_1$  مبین ضریب شتاب،  $w$  مبین تابع وزن و  $rand$  نشان دهنده یک عدد تصادفی بین ۰ و ۱ است. همچنین  $ac_i(t)$  مبین شتاب عامل  $i$ -ام در تکرار  $t$ -ام و  $gbest$  بهترین راه‌حلی است که تا کنون پیدا شده است. موقعیت عوامل در هر تکرار به شرح زیر به روزرسانی می‌شود:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (18)$$

هر عامل در PSOGA به عنوان یک راه حل نامزد در نظر گرفته می‌شود. در ابتدا همه عوامل به صورت تصادفی مقداردهی اولیه می‌گردند. پس از مقداردهی اولیه نیروی گرانشی، ثابت گرانش و نیروهای حاصل در میان عوامل با استفاده از روابط (۴)، (۵) و (۷) محاسبه می‌شوند. سپس شتاب ذرات به وسیله رابطه (۸) تعریف می‌گردد. بهترین راه‌حل پیدا شده در هر تکرار باید به روز شود. پس از محاسبه شتاب و به‌روزرسانی بهترین راه‌حل، سرعت همه عوامل را می‌توان با استفاده از رابطه (۱۷) محاسبه نمود. در نهایت، موقعیت عوامل توسط رابطه (۱۸) به‌روز می‌شود. فرآیند به‌روزرسانی سرعت و موقعیت با دستیابی به یک معیار مشخص

جستجو در تمام فضای مسئله اکتشاف<sup>۱</sup> نام دارد، در حالی که همگرا شدن به بهترین راه حل در نزدیکی یک راه حل خوب بهره‌برداری<sup>۲</sup> نامیده می‌شود. یک الگوریتم مبتنی بر جمعیت باید این دو ویژگی حیاتی را به منظور تضمین پیدا کردن بهترین راه‌حل داشته باشد. در بهینه‌ساز ازدحام ذرات، توانایی اکتشاف با استفاده از  $Pbest$  و توانایی بهره‌برداری با استفاده از  $Gbest$  نمایش داده شده است. در الگوریتم جستجوی گرانشی، با انتخاب مقدار مناسب برای پارامترهای تصادفی ( $G_0$  and  $\alpha$ )، اکتشاف می‌تواند تضمین شود و حرکت آهسته عوامل سنگین‌تر می‌تواند توانایی بهره‌برداری را تضمین کند [۳۲].

راشدی و همکاران در [۲۹] یک مطالعه مقایسه‌ای بین الگوریتم جستجوی گرانشی و برخی از الگوریتم‌های بهینه‌سازی ابتکاری شناخته شده مانند بهینه‌ساز ازدحام ذرات را ارائه می‌دهند. در این مقایسه نتایج نشان می‌دهند که الگوریتم جستجوی گرانشی در زمینه بهینه‌سازی دارای توانایی است. با این حال، الگوریتم جستجوی گرانشی دارای سرعت جستجوی پایین در آخرین تکرارها است [۳۳]. در این مقاله ترکیبی از الگوریتم جستجوی گرانشی با بهینه‌ساز ازدحام ذرات با نام PSOGA، به منظور بهبود این ضعف پیشنهاد شده است.

#### ۴- الگوریتم ترکیبی PSOGA

ایده اصلی PSOGA بدین صورت است که توانایی تفکر

<sup>2</sup> Exploitation

<sup>1</sup> Exploration

FNNPSO، FNNPSOGSA و FNNPSO، معرفی و مورد بررسی قرار می‌گیرند.

#### ۵- آموزش شبکه‌های عصبی چندلایه با استفاده از الگوریتم‌های PSO، GSA و PSOGSA

به طور کلی سه روش برای آموزش شبکه‌های عصبی چندلایه با استفاده از الگوریتم‌های فرا ابتکاری وجود دارد. در روش اول الگوریتم‌های ابتکاری برای پیدا کردن ترکیبی مناسب از وزن‌ها و بایاس‌های شبکه استفاده می‌شوند، به طوری که این ترکیب حداقل خطا را برای شبکه عصبی چندلایه فراهم نماید. در روش دوم، الگوریتم‌های فرا ابتکاری به عنوان راهی برای پیدا کردن یک ساختار مناسب برای شبکه عصبی چندلایه در یک مسئله خاص به کار می‌روند. در آخرین روش، الگوریتم تکاملی، پارامترهای الگوریتم یادگیری مبتنی بر گرادینان را تنظیم می‌کند. این پارامترها عبارتند از: نرخ یادگیری و اندازه حرکت.

در مورد اول، قبل از آموزش شبکه عصبی چندلایه، ساختار شبکه ثابت می‌شود. وظیفه یک الگوریتم آموزشی پیدا کردن یک مقدار مناسب برای تمامی وزن یال‌ها و بایاس‌ها به منظور به حداقل رساندن خطای کلی شبکه عصبی چندلایه است. در مورد دوم، ساختار شبکه‌های عصبی چندلایه متفاوت است.

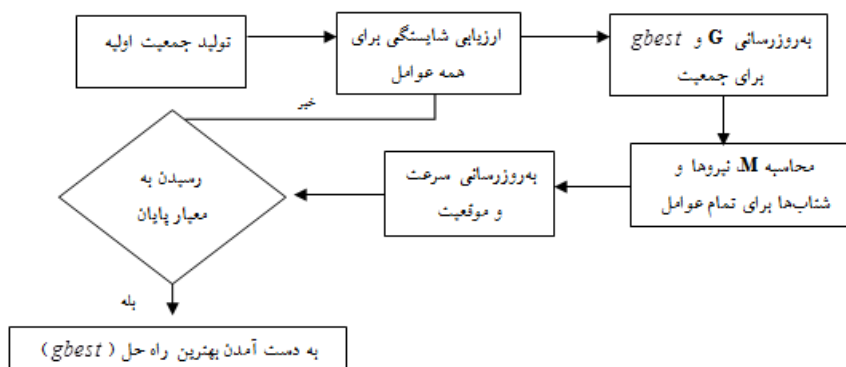
یک الگوریتم آموزشی اعمال شده به شبکه عصبی چندلایه برای تعیین بهترین ساختار برای یک مسئله خاص استفاده می‌شود. تغییر ساختار می‌تواند با دستکاری ارتباط بین نرون‌ها، تعداد لایه‌های پنهان و تعداد گره‌های پنهان در هر لایه از شبکه عصبی چندلایه انجام شود.

متوقف می‌گردد. مراحل الگوریتم PSOGSA در شکل (۲) نشان داده شده است.

برای توضیح کارآمد بودن PSOGSA، به نکات زیر اشاره می‌شود:

- در PSOGSA کیفیت راه‌حل (شایستگی) در روش به روزرسانی در نظر گرفته می‌شود.
  - عواملی که در نزدیکی راه‌حل خوب هستند، عوامل دیگری که در حال بررسی قسمت‌های مختلف فضای جستجو می‌باشند را جذب می‌کنند.
  - هنگامی که تمام عوامل در نزدیکی یک راه‌حل خوب هستند، بسیار آرام حرکت می‌نمایند. در این مورد،  $gbest$  برای بهره‌برداری از بهترین راه‌حل کلی به آنها کمک می‌کند.
  - الگوریتم PSOGSA با استفاده از حافظه، بهترین راه‌حلی که تاکنون کشف شده ( $gbest$ ) را حفظ می‌نماید تا در هر زمان در دسترس باشد.
  - هر عامل می‌تواند بهترین راه‌حل ( $gbest$ ) را مشاهده و به سمت آن میل کند.
  - با تنظیم  $C_1'$  و  $C_2'$ ، توانایی‌های جستجوی کلی و جستجوی محلی می‌تواند متعادل شود.
- نکات ذکر شده در بالا، الگوریتم PSOGSA را برای حل طیف گسترده‌ای از مسائل بهینه‌سازی، به اندازه کافی قدرتمند می‌سازد [۳۴].

در بخش بعد، مکانیزم‌های آموزش شبکه عصبی چندلایه با استفاده از PSO، GSA و PSOGSA و به ترتیب با نام‌های



شکل (۲) مراحل عمومی الگوریتم PSOGSA.

که در آن،  $w_{kj}$  مبین وزن اتصال از گره مخفی  $j$ -ام به گره خروجی  $k$ -ام و  $\theta_k$  مبین بایاس (آستانه) گره خروجی  $k$ -ام است. در نهایت، خطای یادگیری  $E$  (تابع شایستگی) به صورت زیر محاسبه می‌شود:

$$E_k = \sum_{i=1}^m (y_i^k - d_i^k)^2 \quad (21)$$

$$E = \sum_{k=1}^q \frac{E_k}{q} \quad (22)$$

که در آن،  $q$  نشان دهنده تعداد نمونه‌های آموزشی،  $d_i^k$  مبین خروجی مورد انتظار از واحد ورودی  $i$ -ام وقتی که نمونه آموزشی  $k$ -ام استفاده می‌شود و  $y_i^k$  مبین خروجی واقعی از واحد ورودی  $i$ -ام است، وقتی که نمونه آموزشی  $k$ -ام استفاده می‌گردد. بنابراین، تابع شایستگی نمونه آموزشی  $i$ -ام را می‌توان به صورت رابطه (۲۳) تعریف کرد:

$$Fitness(X_i) = E(X_i) \quad (23)$$

### ۵-۲- نمایش مسئله آموزش شبکه عصبی چندلایه توسط الگوریتم‌های PSO، GSA و PSO

(راهبرد کد کردن)

به‌طور کلی، سه روش برای نمایش وزن یال‌ها و بایاس گره‌ها وجود دارد: بردار، ماتریس و حالت دودویی [۱]. در نمایش برداری، هر عنصر با یک بردار نمایش داده می‌شود. برای آموزش یک شبکه عصبی چندلایه، باید همه وزن‌ها و بایاس‌ها معلوم باشند. در نمایش ماتریسی، هر عنصر به‌صورت یک ماتریس و برای نمایش دودویی، هر عنصر به‌صورت رشته‌ای از بیت‌های دودویی نمایش داده می‌شوند. هر کدام از این شیوه‌های نمایش، مزایا و معایب خاص خود را دارند که می‌تواند در مسائل خاص مفید واقع گردد [۳۵].

در روش نخست، تبدیل عناصر به بردار، ماتریس و یا رشته‌ای از بیت‌های دودویی آسان است، اما فرآیند بازیابی آنها پیچیده خواهد بود. به‌همین دلیل اغلب این روش در شبکه‌های عصبی ساده مورد استفاده قرار می‌گیرد. در روش دوم برای شبکه‌ها با ساختار پیچیده، بازیابی آسان‌تر از کد کردن عناصر است. این روش برای الگوریتم‌های یادگیری در شبکه‌های عصبی عمومی بسیار مناسب است. در روش سوم، نیاز است که متغیرها به‌صورت دودویی نمایش داده شوند. در این حالت وقتی که ساختار شبکه پیچیده گردد، طول هر عنصر نیز

در این بخش، به منظور طراحی دسته‌بندی کننده اهداف سوناری، در ابتدا الگوریتم‌های PSO، GSA و PSO با استفاده از روش اول به شبکه عصبی چندلایه اعمال می‌شوند. دسته‌بندی‌کننده‌های طراحی شده به‌ترتیب FNNPSO، FNNNGSA و FNNPSOGSA نام گذاری می‌گردند. توجه شود که ساختار شبکه عصبی چندلایه ثابت است. الگوریتم‌های PSO، GSA و PSO ترکیبی از وزن و بایاس‌ها که حداقل خطا را برای شبکه عصبی چندلایه باعث می‌شود، پیدا می‌کنند. به منظور طراحی FNNPSO، FNNNGSA و FNNPSOGSA عناصر اساسی تابع شایستگی و راهبرد کد کردن باید تعریف شوند.

### ۵-۱- تابع شایستگی

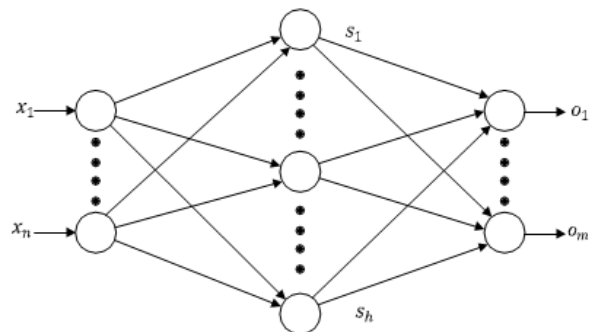
شکل (۳) شبکه عصبی با سه لایه را نشان می‌دهد (یک لایه ورودی، یک لایه پنهان و یک لایه خروجی) که در آن تعداد گره ورودی برابر  $n$ ، تعداد گره مخفی برابر  $h$  و تعداد گره خروجی برابر  $m$  است. در هر دوره از آموزش، خروجی هر گره مخفی به صورت رابطه (۱۹) محاسبه می‌شود:

$$f(s_j) = \frac{1}{\left(1 + \exp\left(-\left(\sum_{i=1}^n w_{ij} \cdot \chi_i - \theta_j\right)\right)\right)} \quad (19)$$

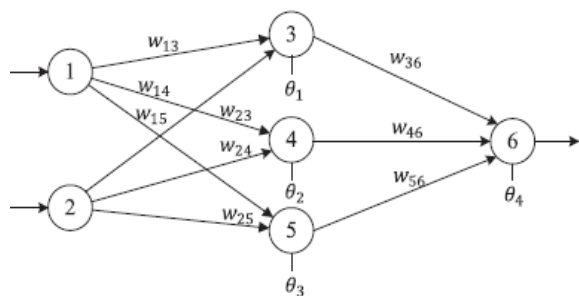
$$j = 1, 2, \dots, h$$

که در آن،  $s_j = \sum_{i=1}^n w_{ij} \cdot \chi_i - \theta_j$ ،  $n$  نشان دهنده تعداد گره ورودی،  $w_{ij}$  مبین وزن اتصال از گره  $i$ -ام در لایه ورودی به گره  $j$ -ام در لایه پنهان،  $\theta_j$  مبین بایاس (آستانه) گره  $j$ -ام و  $\chi_i$  ورودی گره  $i$ -ام است. پس از محاسبه خروجی گره‌های پنهان، خروجی نهایی را می‌توان به شرح زیر تعریف کرد:

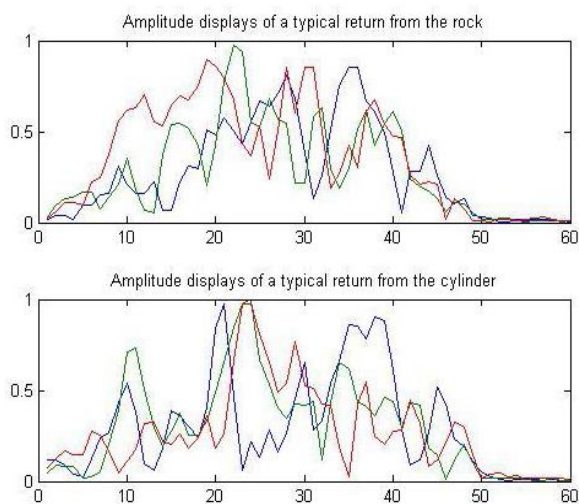
$$o_k = \sum_{j=1}^h w_{kj} f(s_j) - \theta_k \quad k = 1, 2, \dots, m \quad (20)$$



شکل (۳) ساختار شبکه عصبی سه لایه.



شکل (۴) شبکه‌ی عصبی چندلایه با ساختار ۱-۳-۲.



شکل (۵) نمایش دامنه اکوهای برگشتی از سیلندر فلزی و صخره [۳۸].

این افزونگی‌ها باعث کاهش عملکرد شبکه در دسته‌بندی اهداف و همچنین افزایش پیچیدگی زمانی شبکه می‌گردد. این مقاله با استفاده از دو روش متداول استخراج ویژگی‌های جدید (توسط روش‌های مبتنی بر گشتاورهای آماری) و انتخاب ویژگی‌های که بیشترین تأثیر را در دسته‌بندی اهداف دارند (توسط الگوریتم ژنتیک)، ابعاد دادگان ورودی را از  $D$  به  $d$  به صورتی که  $d < D$ ، تبدیل می‌نماید و بدین صورت سعی در کاهش زمان پردازش اطلاعات و افزایش دقت دسته‌بندی شبکه دارد. انتخاب مقدار  $d$  یکی دیگر از مسائل چالش برانگیز این روش است. در این مقاله برای انتخاب  $d$  از روش آزمون scree استفاده شده است.

روش [۳۸] برای انتخاب  $d$  از نمایش نزولی نمودار مقادیر ویژه استفاده می‌کند. بدین صورت که نقطه‌ای را که نمودار افت شدیدی دارد، ولی هنوز در مقادیر کوچک ثابت نشده است را به عنوان  $d$  انتخاب می‌کند. نتیجه این آزمایش در شکل (۶) نمایش داده شده است.

افزایش می‌یابد. بنابراین فرآیند کد کردن و کدگشایی خیلی پیچیده خواهد شد.

در این مقاله، چون با شبکه‌های عصبی چندلایه‌ی پیچیده سروکار نداریم، از روش برداری استفاده شده است. به منظور کاهش زمان اجرای برنامه شبکه‌های عصبی چندلایه، از جعبه ابزارهای عمومی Matlab استفاده نخواهد شد. به عنوان مثالی از این شیوه کد نویسی، بردار نهایی شبکه عصبی چند-لایه شکل (۴)، در رابطه (۲۴) آورده شده است.

$$particle = [w_{13} w_{23} w_{14} w_{24} w_{15} w_{25} w_{36} w_{46} w_{56} \theta_1 \theta_2 \theta_3 \theta_4] \quad (24)$$

## ۶- تشریح مطالب و نتایج

در این بخش الگوریتم PSO-GSA روی دادگان سونار با ابعاد کاهش یافته، اعمال می‌گردد. همچنین به منظور اطمینان از صحت عملکرد الگوریتم بهینه‌سازی PSO-GSA، مجموعه دادگان به الگوریتم‌های بهینه‌سازی ازدحام ذرات و جستجوی گرانشی اعمال و نتایج باهم مقایسه خواهند شد.

### ۶-۱- دادگان سونار

دادگان مورد استفاده در این مقاله از آزمایش Gorman و Sejnowski موجود در مراجع [۳۶ و ۳۷] استخراج شده است. در این آزمایش یک سیلندر فلزی به طول ۵ فوت و یک صخره هم‌اندازه با آن در بستر شنی دریا قرار داده شده‌اند و یک پالس چیرپ FM خطی پهن باند<sup>۱</sup> ( $ka=55/6$ ) به سمت آن‌ها فرستاده شده است. اکوهای برگشتی در فاصله ۱۰ متری از آن‌ها جمع‌آوری شده است.

بر اساس  $SNR$  اکوی دریافتی از ۱۲۰۰ اکو، ۲۰۸ اکو که  $SNR$  آن‌ها بین ۴dB تا ۱۵dB است، انتخاب شده‌اند. از این ۲۰۸ اکو، ۱۱۱ عدد مربوط به سیلندر فلزی و ۹۷ عدد مربوط به صخره هستند. شکل (۵) نمونه‌هایی از اکوهای دریافتی از صخره و سیلندر فلزی را نشان می‌دهد.

### ۶-۲- کاهش ابعاد دادگان با استفاده از الگوریتم ژنتیک و گشتاورهای آماری

با توجه به ابعاد بالای دادگان سونار (در این مقاله  $D=60$ ) و پیچیدگی الگوریتم PSO-GSA، افزونگی‌های زیادی در روند دسته‌بندی اهداف بوجود می‌آید.

<sup>1</sup> Wide-Band Linear FM



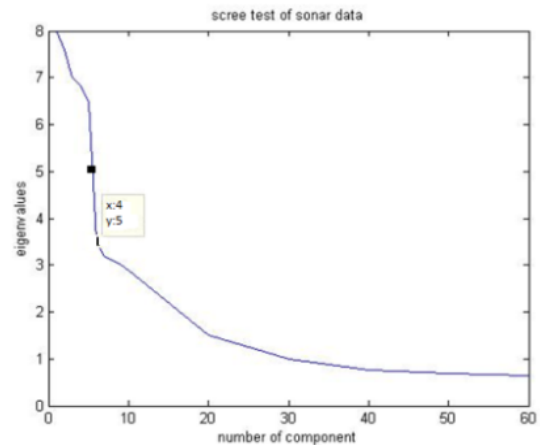
از آنجا که برای داشتن نتیجه بهینه به دو ویژگی دیگر نیز نیاز داریم، از الگوریتم ژنتیک برای انتخاب دو ویژگی مؤثرتر از بین ۶۰ ویژگی در دسترس استفاده می‌کنیم. تکنیک‌های مبتنی بر تئوری اطلاعات برای انتخاب متغیرها در پیش‌بینی سری‌های زمانی و شناسای آماری الگو بسیار پرکاربرد می‌باشند. اساس این تکنیک‌ها حداکثر سازی اطلاعات دوطرفه بین خروجی و ورودی است. بهرحال این فرآیند بدلیل محاسبه آنتروپی دارای بار محاسباتی زیادی است. برای جلوگیری از این بار محاسباتی می‌توان از روش‌های غیرمستقیم برای حداقل سازی افزونگی و حداکثر سازی اطلاعات دوطرفه استفاده کرد. با این وجود بهینه‌سازی ترکیبی (یعنی بررسی تمام متغیرها)، نیازمند بار محاسباتی بسیاری است. برای جلوگیری از این بار محاسباتی در این مقاله از الگوریتم ژنتیک استفاده شده است [۴۱].

ایده اصلی این روش انتخاب یک مجموعه ویژگی  $N_n$  عضوی بصورت  $\{n_i\}, i = (1, 2, \dots, N_n)$  است. ویژگی‌های انتخاب شده سعی در حداکثر سازی اطلاعات دوطرفه بین هدف  $I(y; y_{h_1}, \dots, y_{h_{N_n}})$  (کلاس) و خروجی‌های  $y_{h_i}$  ناشی از ویژگی  $y_{h_i}$  دارد. تعداد راه‌حل‌های این بهینه سازی ترکیبی به صورت رابطه (۲۷) است که در آن  $M$  تعداد کل ویژگی‌ها است.

$$\frac{M!}{2(M - N_n)!} \quad (27)$$

برای جلوگیری از این بار محاسباتی، بهینه‌سازی ترکیبی  $I(y; y_{h_1}, \dots, y_{h_{N_n}})$  با استفاده از الگوریتم ژنتیک انجام می‌پذیرد، با این تفاوت که عملگر ادغام<sup>۴</sup> در این الگوریتم به این صورت است که بجای ترکیب خطی والدین<sup>۵</sup>، یکی از والدین به صورت تصادفی انتخاب می‌گردد.

تابع شایستگی الگوریتم ژنتیک بکار رفته براساس حداکثر سازی ارتباط و حداقل سازی افزونگی است. این حداکثر سازی ارتباط و حداقل سازی افزونگی، با حداکثر سازی رابطه (۲۸) تحقق می‌پذیرد که در این رابطه  $\Phi = V - P$  است.  $V$  و  $P$  به ترتیب در روابط (۲۹) و (۳۰) نشان داده شده‌اند که در آن  $V$  ارتباط بین  $N_n$  ویژگی و  $P$  افزونگی بین آن‌هاست. توجه نمایید که  $V$  میانگین اطلاعات دوطرفه



شکل (۶) نتایج حاصل از آزمون scree بر روی دادگان سونار.

همان گونه که در شکل مشاهده می‌شود برای دادگان سونار،  $4 < d < 6$  می‌تواند انتخاب گردد که برای کاهش بار محاسباتی  $d=4$  انتخاب شده است. همان گونه که در مرجع [۳۹] اثبات شده است اکوهای برگشتی از اهداف، طنین، نویز و کلاتر دارای توابع چگالی احتمال<sup>۱</sup> مخصوص به خود با پارامترهای تقریباً مشخص می‌باشند.

تابع چگالی احتمال مربوط به طنین، ریلی<sup>۲</sup> و کلاتر توزیع  $K$  است، این درحالی است که تابع چگالی احتمال مربوط به هدف واقعی ریلی بعلاوه یک غالب<sup>۳</sup> است [۴۰]. با توجه به این تفاوت در توابع چگالی احتمال مربوط به اهداف واقعی و کاذب و اینکه گشتاور مرتبه سوم (Skewness) و گشتاور مرتبه چهارم (Kurtosis) با تغییر توابع چگالی احتمال تغییر می‌کنند، در این قسمت با استفاده از دادگان اصلی به جای ۶۰ بار نمونه برداری با استفاده از تابع چگالی احتمال مربوط به هر آزمایش، گشتاورهای مرتبه سوم و چهارم را محاسبه می‌کنیم و هر کدام از گشتاورها را به عنوان یک ویژگی جدید در نظر می‌گیریم. برای یک توزیع دامنه  $(X_1, \dots, X_N)$  با میانگین  $\bar{x}$  و واریانس  $\sigma$  که از رابطه (۲۵) بدست می‌آید، می‌توان گشتاورهای مرتبه سوم و چهارم را توسط رابطه (۲۶) و به ترتیب برای  $k=3, k=4$  بدست آورد.

$$\sigma^2 = (1/N) \sum_{m=1}^N (x_m - \bar{x})^2 \quad (25)$$

$$k\text{-th moment} = \sigma^k \quad (26)$$

<sup>4</sup> Crossover Operation

<sup>5</sup> Parents

<sup>1</sup> Probability Density Function

<sup>2</sup> Rayleigh

<sup>3</sup> One-Dominant-Plus-Rayleigh (1D+R)

جدول (۱) پارامترهای مورد نیاز و مقادیر اولیه.

| مقادیر                                    | پارامتر                | الگوریتم |
|---|------------------------|----------|
| ۲   | ثابت شناختی ( $C_1$ )  | PSO      |
| ۲   | ثابت اجتماعی ( $C_2$ ) |          |
| به صورت خطی از ۰/۹ تا ۰/۴<br>کاهش می‌یابد | ثابت محلی ( $w$ )      |          |
| ۲۰۸                                       | اندازه جمعیت           |          |
| اتصال کامل                                | جانمایی                |          |
| به طور تصادفی بین ۰ و ۱                   | سرعت اولیه ذرات        |          |
| ۲۰  | ضریب $\alpha$          | GSA      |
| ۱   | ثابت گرانش ( $G_0$ )   |          |
| به طور تصادفی بین ۰ و ۱                   | سرعت اولیه جرم‌ها      |          |
| ۰   | ارزش اولیه شتاب        |          |
| ۰   | ارزش اولیه جرم         |          |
| ۲۰۸                                       | اندازه جمعیت           |          |
| ۱   | $C'_1$                 | PSOGSA   |
| ۱   | $C'_2$                 |          |
| به صورت خطی از ۰/۹ تا ۰/۴<br>کاهش می‌یابد | ثابت محلی ( $w$ )      |          |
| به طور تصادفی بین ۰ و ۱                   | سرعت اولیه عوامل       |          |
| ۲۰۸                                       | اندازه جمعیت           |          |

$I(yh_i; y)$  بین دسته‌ها و خروجی‌های ناشی از ویژگی‌های انتخاب شده است و  $P$  مقدار میانگین اطلاعات دوطرفه  $I(yh_i; yh_j)$  بین خروجی‌های ویژگی‌های انتخابی است.

$$\max_{i_1 \dots i_{N_n}} \Phi \quad (28)$$

$$V = \frac{1}{N_n} \sum_{i=1}^{N_n} I(yh_i; y) \quad (29)$$

$$p = \frac{1}{N_n^2} \sum_{j=1}^{N_n} \sum_{i=1}^{N_n} I(yh_i; yh_j) \quad (30)$$

### ۳-۶- تنظیم پارامترها و انجام آزمایش

پارامترهای مورد نیاز و مقادیر اولیه در جدول (۱) آورده شده‌اند. هر شبکه، ۱۰ بار آزمایش شده است. بهترین شبکه عصبی آموزش دیده از بین ۱۰ بار اجرا، انتخاب و برای مقایسه به کار گرفته می‌شود.

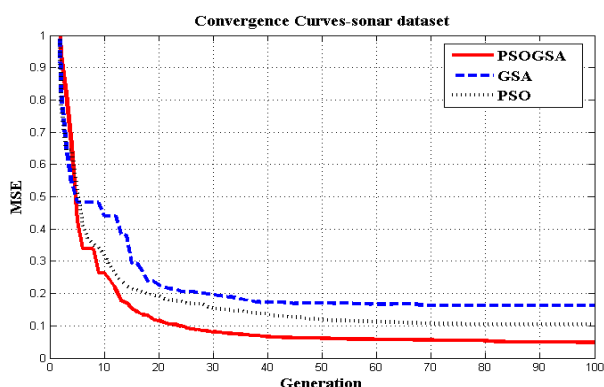
نرخ دسته‌بندی و درصد خطای آزمون دو مقایسه برای مقایسه الگوریتم‌های ذکر شده می‌باشند. برای انجام یک مقایسه نسبتاً خوب، همه الگوریتم‌ها وقتی که حداکثر تعداد تکرار به ۲۵۰ رسید، متوقف می‌شوند. در نهایت همگرایی نتایج برای انجام مقایسه‌ای جامع بررسی خواهد شد. در دسته‌بندی مجموعه‌های داده، از آن جایی که برای انتخاب تعداد گره‌های پنهان استاندارد وجود ندارد، بنابراین براساس ساختار شبکه‌های عصبی چندلایه، از پیشنهاد مطرح شده در مرجع [۴۲] و از رابطه (۱۶) استفاده خواهد شد.

$$H = 2 \times N + 1 \quad (31)$$

که در آن،  $N$  تعداد ورودی‌ها و  $H$  تعداد گره‌های پنهان را نشان می‌دهند.

از نظر آماری، الگوریتم PSOGSA به اندازه کافی توانایی جلوگیری از گیرافتادن در کمینه‌های محلی را دارد و آهنگ همگرایی آن را می‌توان در شکل (۷) مشاهده کرد.

کارایی بهتر الگوریتم PSOGSA بدلیل جستجوی همه فضای مسئله توسط الگوریتم GSA و سرعت همگرایی بالا در رسیدن به بهینه کلی با الگوریتم PSO است. به طور کلی می‌توان مشاهده کرد که FNNGSA نسبت به دسته‌بندی‌کننده‌های دیگر عملکرد خوبی ندارد. این ضعف ناشی از سرعت پایین جستجوی الگوریتم گرانشی است که توانایی بهره‌برداری FNNGSA را تحت تاثیر قرار می‌دهد.



شکل (۷) مقایسه نمودار همگرایی دسته‌بندی‌کننده‌های FNNGSA و FNNGSA.

با این حال الگوریتم جستجوی گرانشی دارای توانایی جستجوی بسیار قوی در میان تمام الگوریتم‌های تکاملی دیگر است [۲۹]. الگوریتم‌های یادگیری مورد استفاده در شبکه‌های عصبی چندلایه، نه تنها به توانایی جستجوی قوی بلکه به توانایی بهره‌برداری دقیق نیز نیاز دارند. از دقت نتایج حاصل از طبقه‌بندی توسط الگوریتم‌های آموزش شبکه عصبی چندلایه، می‌توان نتیجه گرفت که عملکرد FNNGSA بهتر از FNNGSA است. این برتری به دلیل توانایی بهره‌برداری دقیق‌تر بهینه‌ساز ازدحام ذرات است، اما این الگوریتم هنوز هم دارای مشکل به دام‌افتادن در کمینه محلی

- [2] Abedifar, V., Eshghi, M., Mirjalili S. and Mirjalili, S. M., "An Optimized Virtual Network Mapping using PSO in Cloud Computing", 21st Iranian Conference on Electrical Engineering, pp.1-6, 2013.
- [3] Nguyen, L. S., Frauendorfer, D., Mast M. S. and Gatica-Perez, D., "Hire Me: Computational Inference of Hirability in Employment Interviews based on Nonverbal Behavior", IEEE Transactions on Multimedia, Vol.16, No.4, pp.1018-1031, 2014.
- [4] Auer, P., Burgsteiner H. and Maass, W., "A Learning Rule for Very Simple Universal Approximators Consisting of a Single Layer of Perceptrons", Neural Networks, Vol.21, No.5, pp.786-795, June 2008.
- [5] Barakat, M., Lefebvre, D., Khalil, M., Druaux, F. and Mustapha, O., "Parameter Selection Algorithm with Self Adaptive Growing Neural Network Classifier for Diagnosis Issues", International Journal of Machine Learning and Cybernetics, Vol.4, No.3, pp.217-233, 2013.
- [6] Guo, Z. X., Wong, W. K. and Li, M., "Sparsely Connected Neural Network-based Time Series Forecasting", Information Sciences, Vol.193, pp.54-71, 2012.
- [7] Csáji, B. C., "Approximation with Artificial Neural Networks", Faculty of Science, Etsv Lornd University, Hungary, 2001.
- [8] Reed R. D. and Marks, R. J., "Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks", MIT Press, 1998.
- [9] Oja, E., "Unsupervised Learning in Neural Computation", Theoretical Computer Science, Vol.287, No.1, pp.187-207, 2002.
- [10] Zhang, N., "An Online Gradient Method with Momentum for Two-Layer Feedforward Neural Networks", Applied Mathematics and Computation, Vol.212, No.2, pp.488-498, 2009.
- [11] Hush D. R. and Horne, B. G., "Progress in Supervised Neural Networks", IEEE Signal Processing Magazine, Vol.10, No.1, pp.8-39, 1993.
- [12] Ng, S. C., Cheung, C. C., Leung, S. H. and Luk, A., "Fast Convergence for Back Propagation Network with Magnified Gradient Function", IEEE Joint Conference on Neural Networks, Vol.3, pp.1903-1908, 2003.
- [13] Magoulas, G. D., Vrahatis M. N. and Androulakis, G. S., "On the Alleviation of the Problem of Local Minima in Back-propagation", Nonlinear Analysis, Theory, Methods & Applications, Vol.30, No.7, pp.4545-4550, 1997.
- [14] Ho, Y. C. and Pepyne, D. L., "Simple Explanation of the No-Free-Lunch Theorem and its Implications", Journal of Optimization Theory and Applications, Vol.115, No.3, pp.549-570, 2002.
- [15] Wang, P., Yu X. and Lu, J., "Identification and Evolution of Structurally Dominant Nodes in

است. این ضعف بدان معنی است که FNNPSO عملکرد ناپایداری دارد. نتایج به دست آمده توسط FNNPSOGSA ثابت می‌کند که این الگوریتم هر دو مزیت بهره‌برداری قوی و توانایی جستجوی خوب را دارا است. به عبارت دیگر، قدرت بهینه‌ساز ازدحام ذرات و الگوریتم جستجوی گرانشی با موفقیت مورد استفاده قرار گرفته و عملکرد بسیار مناسبی را در آموزش شبکه عصبی چندلایه نشان می‌دهد. این بدین معنی است که FNNPSOGSA قادر به حل مشکل به دام افتادن در کمینه محلی است و دارای سرعت همگرایی بالایی است.

## ۷- نتیجه گیری

در این مقاله اهداف سوناری با استفاده از شبکه‌های عصبی چندلایه (MLP) دسته‌بندی گردیدند. برای آموزش دسته‌بندی‌کننده MLP از الگوریتم بهینه‌سازی ترکیبی ازدحام ذرات و جستجوی گرانشی (PSOGSA) استفاده شد. نشان داده شد که الگوریتم جستجوی گرانشی توانایی خوبی برای جستجوی کلی دارد، اما در آخرین تکرارها دارای سرعت پایین در بهره‌برداری فضای جستجو است. از طرف دیگر با توجه به توانایی منحصربه‌فرد بهینه‌ساز ازدحام ذرات (PSO) در فاز بهره‌برداری از روش ترکیبی PSOGSA برای حل مشکل فوق استفاده گردید. نتایج به دست آمده نشان داد که دسته‌بندی‌کننده‌های مبتنی بر GSA، PSO و PSOGSA دادگان سونار را به ترتیب با دقت ۹۲/۷۵۰۰، ۹۳/۶۷۴۱ و ۹۴/۴۲۳۰۸ دسته‌بندی کردند. همچنین با توجه به شکل (۷)، سرعت همگرایی الگوریتم ترکیبی نسبت به دو الگوریتم معیار ذکر شده، بهبود پیدا نمود.

با توجه به مسایل دنیای واقعی که دارای چندین مساله و پارامتر همزمان بهینه‌سازی هستند و نیز با توجه به توابع محدودکننده بسیار این مسائل، برای بهینه‌سازی آن مسائل بیشتر از الگوریتم‌های بهینه‌سازی سیستم‌های مرکب استفاده می‌شود. این نوع بهینه‌سازی برای دسته‌بندی دادگان سونار می‌تواند به‌عنوان زمینه‌ای برای پژوهش‌های بعدی باشد.

## ۸- مراجع

- [1] Mirjalili, S., Mirjalili, S. M. and Lewis, A., "Let a Biogeography-based Optimizer Train Your Multi-Layer Perceptron", Journal of Information Science, Vol.269, pp.188-209, June 2014.

- [28] Kennedy, J., "Particle Swarm Optimization", Encyclopedia of Machine Learning, pp.760-766, 2010.
- [29] Rashedi, E., Nezamabadi-pour, H. and Saryazdi, S., "GSA: A Gravitational Search Algorithm", Information Sciences, Vol.179, No.13, pp.2232-2248, 2009.
- [30] Newton, I., "In Experimental Philosophy Particular Propositions Are Inferred from the Phenomena and Afterwards Rendered General by Induction", Andrew English Translation Published, Vol.2, 1729.
- [31] Abarghouei, A. A., Ghanizadeh, A. and Shamsuddin, S. M., "Advances of Soft Computing Methods in Edge Detection", International Journal of Advance Soft Computing Application, Vol.1, No.2, pp.1-42, 2009.
- [32] Rashedi, E., Nezamabadi-pour, H. and Saryazdi, S., "BGSA: Binary Gravitational Search Algorithm", Natural Computing, Vol.9, No.3, pp.727-745, 2010.
- [33] Sinaie, S., "Solving Shortest Path Problem using Gravitational Search Algorithm and Neural Networks", Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia, M.Sc. Thesis, 2010.
- [34] Mirjalili, S. and Hashim, S. Z. M., "A New Hybrid PSO-GSA Algorithm for Function Optimization", International Conference on Computer and Information Application (ICCIA), pp.374-377, 2010.
- [35] Mirjalili, S., Hashim, S. Z. M. and Sardroudi, H. M., "Training Feedforward Neural Networks using Hybrid Particle Swarm Optimization and Gravitational Search Algorithm", Applied Mathematics and Computation, Vol.218, No.22, pp.11125-11137, 2012.
- [36] [http://archive.ics.uci.edu/ml/datasets/Connectionist+Benchmark+\(Sonar,+Mines+vs.+Rocks\)](http://archive.ics.uci.edu/ml/datasets/Connectionist+Benchmark+(Sonar,+Mines+vs.+Rocks)).
- [37] Gorman R. P. and Sejnowski, T. J., "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets", Neural Networks, Vol.1, No.1, pp.75-89, 1988.
- [38] Jade, R. K., Verma, L. K. and Verma, K., "Classification using Neural Network and Support Vector Machine for Sonar Data Set", International Journal of Computer Trends and Technology, Vol.4, No.2, pp.116-119, 2013.
- [39] Fialkowski, J. M. and Gauss, R. C., "Methods for Identifying and Controlling Sonar Clutter", IEEE Journal of Oceanic Engineering, Vol.35, No.2, pp.330-354, 2010.
- [40] Moll, C. A. M., Ainslie, M. A. and Janmaat, J., "The Most Likely Distribution of Target Echo Amplitudes", 2nd International Conference & Exhibition on Underwater Acoustic Measurements: Technologies & Results, pp.361-366, 2007.
- Protein-Protein Interaction Networks", IEEE Transactions on Biomedical Circuits and Systems, Vol.8, No.1, pp.87-97, 2014.
- [16] Shaw, S. and Kinsner, W., "Chaotic Simulated Annealing in Multilayer Feedforward Networks", Canadian Conference on Electrical and Computer Engineering, Vol.1, pp.265-269, 1996.
- [17] Chang, S. K., Mohammed, O. A. and Hahn, S. Y., "Detection of Magnetic Body using Artificial Neural Network with Modified Simulated Annealing", IEEE Transactions on Magnetics, Vol.30, No.5, pp.3644-3647, 1994.
- [18] Montana, D. J. and Davis, L., "Training Feedforward Neural Networks using Genetic Algorithms", 11th International Joint Conference on Artificial Intelligence, Vol.1, pp.762-767, 1989.
- [19] Kiranyaz, S., Ince, T., Yildirim, A. and Gabbouj, M., "Evolutionary Artificial Neural Networks by Multi-dimensional Particle Swarm", Neural Networks, Vol.22, No.10, pp.1448-1462, 2009.
- [20] Sellcs, M. and Rylander, B., "Neural Network Learning using Particle Swarm Optimization", Advances in Information Science and Soft Computing, pp.224-226, 2002.
- [21] Zhang, C., Shao, H. and Li, Y., "A New Evolved Artificial Neural Network and Its Application", Proceeding of 3rd World Congress on Intelligent Control and Automation, Vol.2, pp.1065-1068, 2000.
- [22] van den Bergh, F., Engelbrecht, A. P. and Engelbrecht, A. P., "Cooperative Learning in Neural Networks using Particle Swarm Optimizers", South African Computer Journal, No.26, pp.84-90, 2000.
- [23] Zhang, C., Shao, H. and Li, Y., "Particle Swarm Optimization for Evolving Artificial Neural Network", IEEE International Conference on Systems, Man, and Cybernetics, Vol.4, pp.2487-2490, 2000.
- [24] Mirjalili, S. and Safa Sadiq, A., "Magnetic Optimization Algorithm for Training Multilayer Perceptron", IEEE 3rd International Conference on Communication Software and Networks (ICCSN), pp.42-46, 2011.
- [25] Mirjalili, S., Mirjalili, S. M. and Lewis, A., "Grey Wolf Optimizer", Advances in Engineering Software, Vol.69, pp.46-61, 2014.
- [26] Si, T., Hazra, S. and Jana, N., "Artificial Neural Network Training using Differential Evolutionary Algorithm for Classification", The International Conference on Information Systems Design and Intelligent Applications, pp.769-778, 2012.
- [27] Zhang, J. R., Zhang, J., Lok, T. M. and Lyu, M. R., "A Hybrid Particle Swarm Optimization-back-Propagation Algorithm for Feedforward Neural Network Training", Applied Mathematics and Computation, Vol.185, No.2, pp.1026-1037, 2007.

- [41] Ludwig, O. and Nunes, U., "Novel Maximum-Margin Training Algorithms for Supervised Neural Networks", IEEE Transaction on Neural Networks, Vol.21, No.6, pp. 972-984, 2010.
- [42] Mirjalili, S., "Hybrid Particle Swarm Optimization and Gravitational Search Algorithm for Multilayer Perceptron Learning", Master, Universiti Teknologi Malaysia (UTM), 2011.