

## حل مسأله تعیین موقعیت محلی با استفاده از شبکه عصبی مصنوعی چهار لایه

مهرداد کاوه<sup>۱\*</sup>، محمد سعدی مسگری<sup>۲</sup>، علی خسروی

۱- دانشجوی کارشناسی ارشد سیستم اطلاعات جغرافیایی، دانشکده مهندسی نقشه برداری، دانشگاه خواجه نصیرالدین طوسی

۲- دانشیار گروه سیستم اطلاعات مکانی، دانشکده مهندسی نقشه برداری، دانشگاه خواجه نصیرالدین طوسی

۳- دانشجوی کارشناسی ارشد ژئودزی، دانشکده مهندسی نقشه برداری، دانشگاه خواجه نصیرالدین طوسی

تاریخ دریافت مقاله: ۱۳۹۶/۱۰/۱۷ تاریخ پذیرش مقاله: ۱۳۹۷/۰۴/۰۹

### چکیده

امروزه سیستم تعیین موقعیت جهانی (GPS) در درون ساختمان‌ها و در ناحیه‌های شهری متراکم که دید مستقیم بین کاربر و ماهواره‌ها وجود ندارد، کارایی چندانی ندارد. از این رو در سال‌های اخیر سیستم تعیین موقعیت محلی (LPS) به‌طور قابل توجهی مورد استفاده قرار گرفته است. هدف اصلی از این تحقیق، ارائه یک شبکه عصبی مصنوعی چهار لایه مبتنی بر حل معادلات غیر خطی (NLANN) برای حل مسأله تعیین موقعیت محلی می‌باشد. برای ارزیابی عملکرد شبکه عصبی مصنوعی، از سه روش بهینه‌سازی گوس-نیوتون (GN)، الگوریتم ژنتیک (GA) و بهینه‌سازی ازدحام ذرات ترکیبی (HPSO) استفاده شده است. نتایج نشان می‌دهد که مدل پیشنهادی از دقت بالایی برخوردار است. دقت شبکه عصبی مصنوعی بر روی داده‌های شبیه‌سازی شده، ۰/۰۵ متر حاصل شده است. در حالی که بهترین دقت در الگوریتم‌های دیگر حدود ۰/۴۵ متر می‌باشد. در داده‌های شبکه ایستگاه‌های GPS ایتالیا، شبکه عصبی مصنوعی در مدت زمان یک دقیقه به دقت زیر ۱۰ سانتی‌متر دست یافته است. همچنین شبکه عصبی مصنوعی، در ابعاد متفاوت از منطقه مطالعاتی و نسبت سیگنال به نویزهای مختلف (SNR)، دارای دقت بهتری می‌باشد و با افزایش تعداد ایستگاه‌ها، نتایج خوبی را در زمان‌های کمتری به دست آورده است. در حالی که الگوریتم‌های دیگر از دقت خوبی برخوردار نیستند. با این حال الگوریتم HPSO از الگوریتم‌های GA و GN نتایج بهتری را به دست آورده است.

کلیدواژه‌ها: تعیین موقعیت محلی، شبکه عصبی مصنوعی مبتنی بر حل معادلات غیر خطی (NLANN)، گوس-نیوتون، الگوریتم ژنتیک و بهینه‌سازی ازدحام ذرات ترکیبی (HPSO).

\* نویسنده مکاتبه کننده: تهران، خیابان ولی عصر، تقاطع میرداماد، دانشکده نقشه برداری دانشگاه صنعتی خواجه نصیرالدین طوسی

تلفن: ۰۹۱۱۴۶۰۵۳۲۱

Email: M.kaveh11@email.kntu.ac.ir

## ۱- مقدمه

امروزه سیستم‌های ماهواره‌ای ناوبری جهانی به‌طور گسترده‌ای مورد توجه قرار گرفته‌اند [۱]. اکثر سیستم‌های تعیین موقعیت در محیط‌های داخلی ساختمان‌ها و نواحی شهری به‌خوبی عمل نمی‌کنند و پوشش آن‌ها محدود به فضاهای خارجی است [۲]. یکی از این سیستم‌های تعیین موقعیت، سیستم تعیین موقعیت جهانی<sup>۱</sup> (GPS) می‌باشد که اولین بار در ارتش آمریکا توسعه یافته است و در حال حاضر به‌طور گسترده‌ای مورد استفاده قرار می‌گیرد [۳]. GPS در فضاهای باز دارای دقت بالایی بوده ولی در تعیین موقعیت در داخل محیط‌های بسته دچار مشکل است. بنابراین سیستم تعیین موقعیت جهانی، در مواقعی که دید مستقیم بین کاربر و ماهواره‌های GPS وجود ندارد، در درون ساختمان و در ناحیه‌های شهری متراکم نمی‌تواند مورد استفاده قرار بگیرد. از این رو سیستم تعیین موقعیت محلی<sup>۲</sup> (LPS) در سال‌های اخیر به‌طور قابل توجهی توسعه یافته است [۴]. از جمله کاربردهای LPS می‌توان به تولید سرویس‌های مفید برای راهنمایی تورهای مسافرتی در فضاهای سرپوشیده از جمله موزه‌ها، فرودگاه‌ها و هتل‌ها و همین‌طور تعیین موقعیت در مغازه‌ها، بیمارستان‌ها و فضاهای باز اشاره نمود. LPS در فضاهای شهری که سیگنال از ماهواره‌ها به خوبی به کاربر نمی‌رسد، دقت بهتری را نسبت به سیستم تعیین موقعیت جهانی ارائه می‌دهد [۵].

به‌طور کلی روش‌های تعیین موقعیت به سه بخش اساسی مثلث‌بندی، آنالیز وقوع و مجاورت تقسیم می‌شوند. ساویدس<sup>۳</sup> و همکاران از روش مثلث‌بندی برای تعیین موقعیت استفاده کرده‌اند. در این تکنیک، فاصله بر پایه انتقال سیگنال بین فرستنده و گیرنده محاسبه می‌شود [۶]. آنالیز وقوع، سیگنال هدف را با پایگاه داده

بدون اینترنت مقایسه کرده تا نزدیک‌ترین محل را شناسایی کند. همچنین روش مجاورت، یک روش غیر دقیق برای تخمین مختصات نقطه مجهول بوده که موقعیت بهترین نقطه معلوم را به‌عنوان نقطه مجهول در نظر می‌گیرد [۷]. یک فاکتور مهم برای طراحی سیستم‌های مکان‌یابی، پیدا کردن مکانیسمی برای اندازه‌گیری فاصله و یا زاویه می‌باشد. چندین تکنیک در این زمینه ارائه شده است که مهم‌ترین تکنیک‌های موجود در بین این مکانیسم‌ها، زاویه رسیدن سیگنال، قدرت سیگنال دریافت شده، زمان رسیدن سیگنال<sup>۴</sup> (TOA) و اختلاف زمان رسیدن سیگنال می‌باشند [۸ و ۹]. تعیین موقعیت محلی بر پایه این تکنیک‌ها، ابتدا فواصل خط‌دار بین نقطه معلوم و مجهول را تخمین زده و سپس با استفاده از الگوریتم‌های تعیین موقعیت، مختصات نقطه مجهول را به دست می‌آورد. در سیستم‌های تعیین موقعیت محلی، دقت تعیین موقعیت و زمان اجرای الگوریتم‌ها از اهمیت بالایی برای کاربران برخوردار است؛ بنابراین بایستی از الگوریتم‌هایی استفاده شود که هم دقت تعیین موقعیت متناسب با نیاز کاربر را داشته باشند و هم از سرعت اجرای خوبی برخوردار باشند. از این رو استفاده از روش‌های هوش مصنوعی و الگوریتم‌های بهینه‌سازی برای حل معادلات غیرخطی در تعیین موقعیت محلی کارآمد خواهد بود [۱۰].

در تحقیقات زیادی از روش‌های حل دقیق برای مسأله تعیین موقعیت استفاده کرده‌اند که با استفاده از روش‌های تکراری توسعه داده شده‌اند. بیشتر این روش‌ها بر پایه مینیمم کردن تابع هدف می‌باشند که توابع اهداف در این مسائل دارای فرم غیرخطی‌اند [۱۱]. چنگ<sup>۵</sup> و همکاران از روش گوس-نیوتون<sup>۶</sup> (GN) برای تعیین موقعیت با استفاده از شبکه سنسور بی‌سیم

<sup>۴</sup> Time of Arrival<sup>۵</sup> Cheng<sup>۶</sup> Gauss – Newton<sup>۱</sup> Global Positioning System<sup>۲</sup> Local Positioning System<sup>۳</sup> Savvides

برده‌اند [۲۲]. میشر<sup>۱۰</sup> و همکاران از یک شبکه هاپفیلد<sup>۱۱</sup> با یک تابع فعال‌سازی مناسب، جهت حل کردن سیستم غیرخطی جبری از  $m$  معادله با  $n$  مجهول استفاده کرده‌اند [۲۳]. لی<sup>۱۲</sup> و همکاران، شبکه عصبی برپایه گرادیان نزولی را برای حل کردن سیستم جبری غیرخطی با سرعت همگرایی و دقت بالا مورد استفاده قرار داده‌اند [۲۴]. گروه تحقیقاتی مارگریز<sup>۱۳</sup> ریشه‌های یک معادله جبری غیر خطی را با یک شبکه عصبی برگشت‌پذیر حل کرده‌اند. مزیت اصلی روش آن‌ها استفاده از یک ساختاری است که معادلات غیر خطی را شبیه‌سازی نموده و تمام ریشه‌های آن را حل می‌کند [۲۵].

در هیچ یک از تحقیقات قبلی از شبکه عصبی حل‌کننده معادلات غیرخطی<sup>۱۴</sup> ( $NLANN$ ) در تعیین موقعیت محلی، استفاده نشده است. از این رو در این تحقیق برای ارتقای دقت تعیین موقعیت محلی و کاهش زمان محاسباتی، از شبکه عصبی مخصوص حل معادلات غیرخطی استفاده می‌شود. برای مقایسه عملکرد  $NLANN$  از مقایسه آن با سه الگوریتم گوس-نیوتون، الگوریتم ژنتیک و بهینه‌سازی ازدحام ذرات ترکیبی<sup>۱۵</sup> ( $HPSO$ ) استفاده شده است. هم‌چنین برای اندازه‌گیری فاصله، روش زمان رسیدن به سیگنال ( $TOA$ ) به کار رفته است. بخش‌های دیگر این تحقیق به شرح زیر می‌باشد؛ در بخش دوم اصول اولیه تعیین موقعیت و توابع اهداف بررسی می‌شوند. بخش سوم در ارتباط با روش انجام تحقیق و فرموله‌سازی مسأله می‌باشد. بخش چهارم شامل نتایج، مقایسه و ارزیابی الگوریتم‌ها بر روی دو مجموعه داده شبیه‌سازی و واقعی می‌باشد و در بخش آخر نتیجه‌گیری و پیشنهادها این تحقیق ارائه شده است.

استفاده کرده‌اند [۱۲]. یو<sup>۱</sup> و همکاران از مدل  $GN$  با روش لئونبرگ مارکوادر<sup>۲</sup> در تعیین موقعیت با سیگنال‌های پهنای باند بالا برای شبکه‌های سنسورهای بی‌سیم بهره برده‌اند [۱۳]. هم‌چنین در بسیاری از تحقیقات از الگوریتم‌های فراابتکاری و شبکه عصبی مصنوعی<sup>۳</sup> ( $ANN$ ) ساده برای مسائل تعیین موقعیت، به‌عنوان روش‌های بهینه‌سازی مبتنی بر هوش مصنوعی استفاده کرده‌اند [۱۴، ۱۵، ۱۶ و ۱۷]. کلارو<sup>۴</sup> و همکاران از شبکه عصبی مصنوعی برای محاسبه دقت تعیین موقعیت محلی استفاده کردند. نتایج سیستم مکان‌یابی نشان داده است که متوسط دقت تعیین موقعیت آن‌ها زیر دو متر می‌باشد [۱۸]. گروه تحقیقاتی ژو<sup>۵</sup> و همکاران، الگوریتم ژنتیک<sup>۶</sup> ( $GA$ )، بهینه‌سازی ازدحام ذرات<sup>۷</sup> ( $PSO$ ) و شبکه عصبی مصنوعی بازگشتی را برای مسأله تعیین موقعیت محلی مقایسه کرده‌اند. نتایج نشان داده است که دقت حاصل از شبکه عصبی کمی بهتر از الگوریتم‌های  $GA$  و  $PSO$  می‌باشد. هم‌چنین الگوریتم‌های  $GA$  و  $PSO$  به نسبت شبکه عصبی، زمان زیادی را برای رسیدن به جواب بهینه صرف کرده‌اند [۱۹]. هم‌چنین در بسیاری از تحقیقات، شبکه عصبی مصنوعی ساده، برای حل مسائل مختلف خطی به خوبی مورد استفاده قرار گرفته است؛ اما برای معادلات و سیستم‌های غیرخطی کمتر مورد استفاده قرار گرفته است [۲۰ و ۲۱]. ماشیا<sup>۸</sup> و سیکس<sup>۹</sup> شبکه عصبی بازگشتی چند لایه را برای حل معادلات غیرخطی به کار

<sup>۱</sup> Yu<sup>۲</sup> Levenberg Marquardt<sup>۳</sup> Artificial Neural Network<sup>۴</sup> Claro<sup>۵</sup> Zho<sup>۶</sup> Genetic Algorithm<sup>۷</sup> Particle Swarm Optimization<sup>۸</sup> Mathia<sup>۹</sup> Saeks<sup>۱۰</sup> Mishra<sup>۱۱</sup> Hopfield<sup>۱۲</sup> Li<sup>۱۳</sup> Margaris<sup>۱۴</sup> Non Linear Artificial Neural Network<sup>۱۵</sup> Hybrid Particle Swarm Optimization

## ۲- اصول اولیه تعیین موقعیت

به طور کلی تعیین موقعیت دارای سه مرحله اصلی می باشد. مرحله اول تخمین فاصله بین فرستنده و گیرنده است که در این مقاله از روش زمان رسیدن به سیگنال استفاده می شود. در مرحله دوم نوع تابع هدف برای مسأله تعیین می شود. مرحله سوم هم به دست آوردن مختصات نقطه مجهول با استفاده از الگوریتم های تعیین موقعیت می باشد.

## ۲-۱- زمان رسیدن به سیگنال

زمان رسیدن به سیگنال یکی از روش های تعیین موقعیت است که از تکنیک مثلث بندی برای مشخص کردن موقعیت کاربر استفاده می نماید. در این روش، فاصله، از مدت زمان حرکت موج بین دریافت کننده و فرستنده محاسبه می شود [۲۶]. در سیستم های تعیین موقعیت، زمان انتشار به طور مستقیم می تواند تبدیل به فاصله شود اما دریافت کننده نیاز دارد تا زمان دقیق انتشار را داشته باشد. بنابراین دقت تعیین موقعیت در این روش به خطاهای تأخیر در انتشار و دقت اندازه گیری زمان وابسته است. اگر دید مستقیم ( $LOS^1$ ) بین نقطه مجهول و نقاط معلوم وجود داشته باشد، موقعیت نقطه مجهول می تواند به دقت محاسبه شود؛ اما خط دید غیر مستقیم باعث می شود که سیگنال مسیر بلندتری را از نقطه مجهول تا نقطه معلوم طی کند و  $TOA$  اندازه گیری شده در آن عموماً خطای بزرگ تری را نسبت به سیگنال  $LOS$  دارد. در چنین شرایطی نیاز به مشخص کردن خطای  $NLOS$  و تصحیح خطای بایاس در اندازه گیری  $TOA$ ، قبل از پردازش می باشد [۲۷]. زمانی که محدوده اندازه گیری به دست آمده از زمان رسیدن دارای هیچ گونه خطایی نباشد، آن گاه فاصله بین نقاط معلوم و مجهول طبق رابطه (۱) محاسبه می شود.

$$r_{TOA} = c \Delta t_{TOA} \quad \text{رابطه (۱)}$$

که در این رابطه،  $c$  سرعت موج الکترومغناطیس،  $r_{TOA}$  محدوده و فاصله اندازه گیری شده بین فرستنده و

دریافت کننده و  $\Delta t_{TOA}$  زمان رسیدن سیگنال در دریافت کننده می باشد. اگر محدوده اندازه گیری به دست آمده از زمان رسیدن سیگنال دارای خطا باشد و همچنین خط دید مستقیم بین نقاط معلوم و مجهول وجود نداشته باشد، آنگاه محدوده اندازه گیری می تواند توسط رابطه (۲) مدل شود. در رابطه (۲)،  $m$  اندیس نقاط معلوم و  $m=1,2,\dots,M$  اندیس نمونه زمانی و  $i=0,1,\dots,(k-1)$  تعداد اندازه گیری در زمان های مختلف  $t_i$ ،  $L_m$  فاصله واقعی بین نقاط معلوم و مجهول،  $n_m$  خطای استاندارد اندازه گیری و  $NLOS_m$  خطای ناشی از دید غیر مستقیم می باشند. اگر مختصات نقطه معلوم برابر با  $(x_m, y_m)$  و مختصات نقاط مجهول هم  $(x(t_i), y(t_i))$  باشد، فاصله واقعی بین آن ها طبق رابطه (۳) محاسبه می شود. به طور کلی اگر اطلاعات نویز در دسترس نباشد، خطای استاندارد را می توان از دو روش شبیه سازی نمود. حالت اول استفاده از تابع گوسین می باشد و در حالت دوم خطای استاندارد برحسب فاصله ( $d_i$ ) مشاهده شده بین نقاط معلوم و مجهول و نسبت سیگنال به نویز ( $SNR$ ) تخمین زده می شود که در رابطه (۴) آمده است [۲۸].

## ۲-۲- توابع اهداف

هدف اصلی در این مسأله، رسیدن به دقت مناسب برای مختصات نقطه مجهول می باشد. در واقع اختلاف بین فاصله محاسبه شده و فاصله خطا دار به عنوان تابع هدف می باشد که هر قدر این مقدار کمینه شود، مختصات نقطه مجهول با دقت بهتری به دست می آید. از آنجایی که این تابع از اختلاف روابط (۱) و (۲) به دست می آید، یک مسأله غیر خطی می باشد که می توان از توابع کمترین مربعات غیر خطی<sup>۲</sup> ( $NLS$ ) و ماکزیمم احتمال<sup>۳</sup> ( $ML$ ) استفاده کرد.

<sup>۲</sup> Non-linear least squares

<sup>۳</sup> Maximum likelihood

<sup>۱</sup> Lines of Sight

$$r_{TOA} = r_m(t_i) = L_m(t_i) + n_m(t_i) + NLOS_m(t_i) \quad (2) \text{ رابطه}$$

$$L_m(t_i) = \sqrt{(x(t_i) - x_m)^2 + (y(t_i) - y_m)^2} \quad (3) \text{ رابطه}$$

$$\sigma_{\text{TOA}} = d_1 / \sqrt{\text{SNR}} \quad (4) \text{ رابطه}$$

۲-۲-۱- کمترین مربعات غیر خطی (NLS)

در این تابع هدف برای تعیین دقت مختصات نقطه مجهول از کمینه کردن رابطه (۵) استفاده شده است که در آن به دلیل در دسترس نبودن اطلاعات نویز، رابطه (۴) برای تخمین نویز سیگنال به کار رفته است.

۲-۲-۲- ماکزیمم احتمال ( $ML$ )

در این تابع هدف فرض بر این است که اطلاعات مربوط به نویز معلوم می‌باشد. در این تحقیق از توزیع خطای گوسین برای مدل‌سازی خطای استاندارد استفاده شده است. رابطه (۶) تابع هدف  $ML$  را نشان می‌دهد [۲۸].

ماکزیمم کردن رابطه (۶) هم‌ارز با کمینه کردن ترم دوم آن می‌باشد (رابطه (۷)).

$$J_{NLS,TOA}(\tilde{x}) = \sum_{l=1}^L \left( r_{TOA,l} - \sqrt{(\tilde{x} - x_l)^2 + (\tilde{y} - y_l)^2} \right)^2 = (r_{TOA} - f_{TOA}(\tilde{x}))^T (r_{TOA} - f_{TOA}(\tilde{x})) \quad (5) \text{ رابطه}$$

$$\ln(p(r_{\text{TOA}})) = \ln\left(\frac{1}{(2p)^{L/2} |C_{\text{TOA}}|^{1/2}}\right) - \frac{1}{2}(r_{\text{TOA}} - d)^T C_{\text{TOA}}^{-1} (r_{\text{TOA}} - d) \quad \text{ابطه (۶)}$$

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left( \left( \mathbf{r}_{\text{TOA}} - \mathbf{f}_{\text{TOA}}(\tilde{\mathbf{x}}) \right)^T \mathbf{C}_{\text{TOA}}^{-1} \left( \mathbf{r}_{\text{TOA}} - \mathbf{f}_{\text{TOA}}(\tilde{\mathbf{x}}) \right) \right) \quad (7)$$

### ۳- روش انجام تحقیق

همان‌طور که در بخش‌های قبلی اشاره شد به‌طور کلی دو روش برای حل کردن معادلات غیرخطی با توابع اهداف  $NLS$  و  $ML$  به‌کار رفته است. روش‌های اول مربوط به یافتن کمینه محلی است که این روش‌ها به الگوریتم‌های تکراری نیوتون منسوب می‌شوند. در این مطالعه از روش گوس-نیوتون به عنوان یکی از الگوریتم‌های تکراری نیوتون استفاده شده است. روش‌های دوم استفاده از الگوریتم‌های بهینه‌سازی مبتنی بر جستجو تصادفی همانند الگوریتم‌های ژنتیک و  $PSO$  و روش‌های هوش مصنوعی مانند شبکه عصبی مصنوعی می‌باشند. در ادامه اصول اولیه و نحوه فرموله‌سازی مسأله در الگوریتم‌ها ارائه شده‌اند.

روش‌های غیرخطی تلاش می‌کنند تا موقعیت نقطه معلوم را مستقیماً از معادلات ارائه شده، شامل تخمین زنده‌های  $ML$  و  $NLS$  به‌دست آورند. در اکثر مسائل تعیین موقعیت، اطلاعات مربوط به نویز سیگنال موجود نمی‌باشد و یا برآزش توابع برای تخمین خطای استاندارد کار دشواری است که به راحتی به‌دست نمی‌آید. زمانی که اطلاعات نویز در دسترس است، تخمین زنده  $ML$  می‌تواند به‌عنوان مدل وزن‌دار روش  $NLS$  با استفاده از کوواریانس نویز بررسی شود. در این تحقیق به دلیل عدم وجود اطلاعات نویز، از تابع هدف  $NLS$  در الگوریتم‌های ارائه شده به کار رفته است و تنها در الگوریتم  $GN$  از هر دو نوع توابع اهداف استفاده شده است.

### ۳-۱- الگوریتم گوس-نیوتون (GN)

روش GN شامل حل یک دنباله‌ای از تقریب‌های تابع بهینه غیرخطی NLS و ML است که هر کدام از آن‌ها با یک روند مستقیم یا تکراری حل می‌شوند. در مقایسه با روش نیوتون و انواع مختلف آن، روش GN برای حل توابع بهینه ML و NLS جالب توجه است؛ زیرا احتیاجی به برآورد یا تخمین مشتق مرتبه دوم تابع  $F(x)$  ندارد و بنابراین از لحاظ عددی خیلی مؤثر است. در این روش ابتدا یک موقعیت اولیه به GN داده می‌شود که همان مختصات نقطه مجهول می‌باشد. سپس الگوریتم طبق

رابطه (۸)

$$\hat{x}^{k+1} = \hat{x}^k - \mu \tilde{N}(J_{NLS, TOA}(\hat{x}^k))$$

$$\|\hat{x}^{k+1} - \hat{x}^k\|_2 \leq \epsilon E \quad \text{رابطه (۹)}$$

هدف از این بهینه‌سازی کمینه کردن دقت تعیین موقعیت محلی می‌باشد. بنابراین تابع هدف برای GA همان رابطه NLS می‌باشد. برای انتخاب والدین از روش چرخ گردان<sup>۲</sup> استفاده شده است [۳۱]. برای تولید فرزندان از والد‌های انتخاب شده، از ترکیب ریاضی<sup>۳</sup> استفاده شده است؛ به این صورت که یک بردار تصادفی بین صفر و یک ( $a$ ) انتخاب می‌شود و فرزند اول طبق رابطه (۱۰) و فرزند دوم طبق رابطه (۱۱) ایجاد می‌شوند.

همچنین برای انجام عمل جهش از توزیع‌های احتمال استفاده شده است. در رابطه (۱۲) نحوه جهش برای الگوریتم ژنتیک ارائه شده است که در آن  $\delta_i$  یکی از توزیع‌های احتمالی یکنواخت و یا نرمال می‌باشد.

### ۳-۲- الگوریتم ژنتیک (GA)

الگوریتم ژنتیک از جمله روش‌های فراابتکاری در مسائل بهینه‌سازی است که اولین بار در سال ۱۹۷۵ به‌وسیله هالند<sup>۱</sup> مطرح شده است [۲۹]. این الگوریتم یک روش مبتنی بر جمعیت می‌باشد که در هر تکرار محاسباتی روی جمعیتی از کروموزوم‌ها عمل کرده و تغییرات تصادفی بر روی آن‌ها از طریق اعمال عملگرهای انتخاب، ادغام و جهش ایجاد می‌کند [۳۰]. شکل (۱) نحوه تعریف یک کروموزوم در الگوریتم ژنتیک پیوسته را نشان می‌دهد که شامل دو ژن به‌عنوان مختصات نقطه مجهول می‌باشد. مختصات نقطه مجهول به عنوان یک جواب مسأله می‌باشد. در ادامه نحوه مدل‌سازی عملگرهای GA بیان می‌شوند.

|   |   |
|---|---|
| x | Y |
|---|---|

شکل ۱: کدگذاری یک جواب مسأله به عنوان یک کروموزوم

<sup>۲</sup> Wheel Roulette

<sup>۳</sup> Arithmetic

<sup>۱</sup> Holland

$$Offspring1 = a \times parent1 + (1 - a) \times parent2 \quad \text{رابطه (۱۰)}$$

$$Offspring2 = (1 - a) \times parent1 + a \times parent2 \quad \text{رابطه (۱۱)}$$

$$X'_i = X_i + (max_i - min_i) \times \delta_i \quad \text{رابطه (۱۲)}$$

ذرات در فضای پاسخ حرکت می کنند. سرعت ذرات طبق رابطه (۱۳) به روزرسانی می شود [۳۱]. در رابطه (۱۳)،  $V_{id}(t)$  سرعت فعلی ذره در بعد  $d$ ، ضریب اینرسی،  $brand(0, \varphi_1)$  یک عدد تصادفی بین صفر و  $P_{id}(t)$ ، بهترین تجربه شخصی ذره در بعد  $d$ ،  $P_{gd}(t)$ ،  $brand(0, \varphi_2)$  یک عدد تصادفی بین صفر و  $X_{id}(t)$ ، بهترین تجربه همسایه های ذره در بعد  $d$ ، موقعیت فعلی ذره در بعد  $d$  و  $V_{id}(t+1)$  سرعت جدید ذره در بعد  $d$  می باشد. در نهایت موقعیت جدید ذره طبق رابطه (۱۴) به روزرسانی می شود.

$$V_{id}(t+1) = \alpha V_{id}(t) + \beta brand(0, \varphi_1)(P_{id}(t) - X_{id}(t)) + \beta brand(0, \varphi_2)(P_{gd}(t) - X_{id}(t)) \quad \text{رابطه (۱۳)}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad \text{رابطه (۱۴)}$$

تولید جمعیت اولیه و شرط خاتمه نیز همانند الگوریتم ژنتیک اعمال شده است. در هر لحظه از اجرا، موقعیت جدید ذره با بهترین تجربه خود مقایسه می شود و اگر بهتر باشد به عنوان تجربه شخصی ذخیره خواهد شد. همچنین در هر نسل، ذره با بهترین تابع بهینگی به عنوان بهترین ذره عمومی و جواب مسأله انتخاب می شود.

### ۳-۴- شبکه عصبی مصنوعی به عنوان سیستم حل کننده غیر خطی

شبکه های عصبی مصنوعی از پرکاربردترین و عملی ترین روش های مدل سازی مسائل پیچیده و بزرگ که شامل صدها متغیر هستند، می باشند [۳۲، ۳۳ و ۳۴]. شبکه های عصبی مصنوعی به منظور برقراری یک

### ۳-۳- الگوریتم بهینه سازی ازدحام ذرات ترکیبی (HPSO)

الگوریتم  $PSO$  یک روش بهینه سازی سراسری است که اولین بار در سال ۱۹۹۵ توسط راسل ابرهارت و جیمز کندی بر اساس رفتار جمعی پرندگان و ماهی ها مطرح شده است [۳۰ و ۳۱].  $PSO$  همانند الگوریتم های هوش جمعی، مبتنی بر جمعیت می باشد که می تواند مسائلی که جواب آن ها در یک فضای  $n$  بعدی است را حل کند. در چنین فضایی، فرضیاتی مطرح می شود و یک سرعت ابتدایی به ذرات اختصاص داده می شود. سپس این

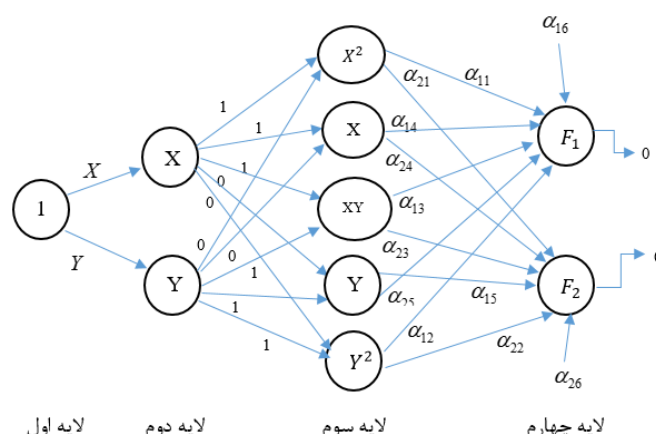
در این تحقیق از یک جستجو همسایگی برای ارتقای بهترین تجربه جهانی ذرات در الگوریتم  $PSO$  استفاده شده است؛ برای تعریف جستجوی همسایگی در مسأله حاضر از یک فضای بیضوی به مرکز ذره نخبه به عنوان فضای جستجو استفاده شده است. منظور از جستجو همسایگی، یافتن ذره ای با مختصات جدید در داخل یک بیضوی به مرکز بهترین ذره می باشد که اگر تابع بهینگی ذره جدید بهتر شده باشد، جایگزین ذره نخبه خواهد شد. هدف اصلی از این جستجو همسایگی این است که سرعت همگرایی الگوریتم افزایش پیدا کند بدون آن که جستجو تصادفی را کم نماید. تعریف یک ذره در الگوریتم  $HPSO$  همانند تعریف کروموزوم در الگوریتم  $GA$  می باشد. هم چنین تعریف تابع بهینگی،

سیستم جبری غیر خطی همانند رابطه (۱۵)، یک شبکه عصبی مصنوعی مبتنی بر حل آن‌ها مدل‌سازی شده است که ساختار آن در شکل (۲) آمده است.

$$\alpha_{11}x^2 + \alpha_{13}xy + \alpha_{12}y^2 + \alpha_{14}x + \alpha_{15}y = \alpha_{16}$$

$$\alpha_{21}x^2 + \alpha_{23}xy + \alpha_{22}y^2 + \alpha_{24}x + \alpha_{25}y = \alpha_{26}$$

رابطه (۱۵)



شکل ۲: یک شبکه عصبی حل کننده معادلات غیرخطی چهار لایه‌ای

این روش از تابع فعال‌سازی  $y = x$  برای تمام نورون‌های لایه ورودی و پنهان استفاده شده است. همچنین از تابع فعال‌سازی تانژانت هایپربولیک برای نورون‌های لایه خروجی به کار گرفته شده است. اگر ورودی تابع تانژانت هایپربولیک صفر باشد خروجی آن هم صفر خواهد بود. بنابراین اگر نورون‌های لایه دوم همان ریشه‌های معادله غیرخطی باشند، خروجی شبکه برابر با صفر خواهد بود. ترم‌های ثابت  $a_{16}$  و  $a_{26}$  به عنوان بایاس نورون خروجی می‌باشند که در واقع نقش فواصل خط‌آدار مشاهداتی مسأله را ایفا می‌کنند. زمانی  $NLANN$  می‌تواند معادلات غیر خطی مسأله را به خوبی حل کند که با الگوریتم پس انتشار خطا آموزش دیده شود. در این شبکه، شیوه آموزش طبق رابطه (۱۶) می‌باشد که بر طبق روش نزول گرادیان است و طی آن وزن‌های لایه ورودی به‌روزرسانی شده و شبکه رو به اصلاح خواهد رفت.

مدل پیشنهاد شده در شکل (۲) از یک لایه ورودی، دو لایه پنهان و یک لایه خروجی تشکیل شده است. لایه ورودی دارای یک نورون با مقدار ثابت واحد می‌باشد. لایه پنهان اول شامل دو نورون بوده که ترم‌های خطی  $X$  و  $Y$  را تولید می‌کند. همچنین در لایه پنهان دوم ترم‌های غیرخطی  $X^2$ ،  $Y^2$  و  $XY$  تولید می‌شوند و سرانجام در لایه خروجی اختلاف بین طرفین رابطه (۱۵) محاسبه می‌شود که هرچه قدر به صفر نزدیک‌تر باشد جواب مسأله دقیق‌تر خواهد بود. در شبکه عصبی پیشنهاد شده، وزن‌های ارتباط‌دهنده بین لایه دوم و سوم ثابت و برابر با صفر و یک می‌باشند. در حالی که وزن‌های ارتباط‌دهنده بین لایه سوم و چهارم برابر با ضرایب معادلات غیرخطی ارائه شده در رابطه (۱۵) می‌باشند. تنها وزن‌هایی متغیر هستند که نورون ورودی را به اولین لایه پنهان متصل می‌کنند که در واقع همان مختصات نقطه مجهول در تعیین موقعیت می‌باشند. در

شبیه‌سازی، در مرحله بعد الگوریتم‌ها بر روی داده‌های واقعی در محدوده شبکه ایستگاه‌های GPS ایتالیا با فواصل چند ده کیلومتری که دارای خطای مشاهداتی حداکثر ۱۰۰ متر می‌باشند، پیاده‌سازی شده‌اند.

جدول ۱: مختصات ایستگاه‌های پایه

| شماره ایستگاه‌های پایه | مختصات  |
|------------------------|---------|
| ایستگاه ۱              | (۰،۰)   |
| ایستگاه ۲              | (۱۰،۰)  |
| ایستگاه ۳              | (۰،۱۰)  |
| ایستگاه ۴              | (۱۰،۱۰) |

#### ۴-۱- اجرای الگوریتم‌ها بر روی داده‌های

##### شبیه‌سازی

در این قسمت هر چهار روش بر روی داده‌های شبیه‌سازی پیاده‌سازی شده‌اند. از آنجایی که دقت و زمان اجرای الگوریتم‌ها در مسأله تعیین موقعیت محلی مهم می‌باشد لذا برای الگوریتم‌های مورد استفاده در این تحقیق، این دو پارامتر مورد ارزیابی و مقایسه قرار گرفته است.

#### ۴-۱-۱- اجرای الگوریتم گوس-نیوتون بر روی

##### داده‌های شبیه‌سازی

در جدول (۲) دقت و زمان محاسبه الگوریتم GN با دو تابع هدف مختلف NLS و ML در تکرارهای متفاوت مورد بررسی قرار گرفته است. نتایج نشان داده است که این الگوریتم با تابع هدف ML دارای دقت بهتری می‌باشد. همان‌طور که ملاحظه می‌شود، GN به مقدار اولیه وابسته است و هرچه قدر مقدار اولیه به جواب مسأله نزدیک‌تر باشد الگوریتم دقت بهتری را ارائه می‌دهد. ولی با افزایش تعداد تکرار دقت نتایج تغییر چندانی نمی‌کند. در واقع در این روش دقت تعیین موقعیت به مقدار اولیه مختصات نقطه مجهول و نوع تابع هدف وابسته است و مستقل از تعداد تکرار می‌باشد.

$$lr_i = alr / (\text{norm}(F_i')^2)$$

$$d_i = \sum F_k \cdot F_i^k \quad \text{رابطه (۱۶)}$$

$$x_i = lr_i \cdot d_i$$

در این روابط  $F$  همان تابع بهینگی،  $F'$  مشتق تابع بهینگی،  $alr$  نرخ آموزش و اندیس‌های  $k$  و  $i$  به ترتیب نشان‌دهنده تعداد معادلات و تعداد تکرار می‌باشند.  $X_i$  نیز مختصات مجهول به‌روزرسانی شده است. به دلیل آن که تنها وزن‌های ارتباط‌دهنده بین لایه‌های اول و دوم متغیر می‌باشد، لذا در آموزش شبکه فقط این وزن‌ها به‌روزرسانی می‌شوند. در هر مرحله از تکرار، تابع بهینگی در شرط شبکه قرار گرفته و اگر به یک مقدار قابل قبولی رسیده باشد، شبکه متوقف می‌شود. همچنین اگر در شرط شبکه صدق نکند مقدار مختصات مجهول یا همان وزن شبکه در روند اجرای شبکه با رابطه (۱۶) به‌روز رسانی شده تا مقدار تابع بهینگی جدید تولید شود. مهم‌ترین پارامتر در قسمت به‌روزرسانی وزن شبکه، نرخ آموزش می‌باشد که در این شبکه مقدار ۰/۴ بهترین دقت تعیین موقعیت را برآورد می‌کند. با روند مراحل بالا شبکه پیشنهاد شده قادر است معادلات غیر خطی درجه دو را حل کند.

#### ۴- بحث و نتایج

همان‌طور که در بخش معرفی ذکر شده است، در این پژوهش از دو سری داده شبیه‌سازی و واقعی برای محاسبه دقت تعیین موقعیت استفاده شده است. ابتدا چهار روش بهینه‌سازی در مجموعه داده‌های شبیه‌سازی به‌کار گرفته شده‌اند. در جدول (۱) داده‌های شبیه‌سازی آمده است که شامل چهار ایستگاه معلوم در یک محدوده مربعی به ابعاد ۱۰ متر می‌باشند و هدف، به‌دست آوردن دقت تعیین موقعیت محلی یک نقطه در داخل این فضا می‌باشد. از مختصات چهار گوشه این منطقه به‌عنوان نقاط معلوم برای تعیین موقعیت استفاده شده است و میزان خطای فواصل بین نقاط مجهول و معلوم برابر با یک متر اعمال شده است. پس از کالیبراسیون پارامترهای الگوریتم‌ها با استفاده از داده

جدول ۲: کالیبره کردن تعداد تکرار در GN با روش سعی و خطا

| مختصات نقاط مجهول | تعداد تکرار | زمان اجرا (ثانیه) | تابع بهیگی $ML$ (متر) | تابع بهیگی $NLS$ (متر) |
|-------------------|-------------|-------------------|-----------------------|------------------------|
| (۳ و ۶)           | ۱           | ۰/۵۲              | ۰/۵۳                  | ۰/۶۵                   |
|                   | ۵           | ۰/۷۶              | ۰/۵۰                  | ۰/۶۴                   |
|                   | ۱۰          | ۱/۰۹              | ۰/۵۴                  | ۰/۶۳                   |
|                   | ۲۰          | ۱/۶۵              | ۰/۵۳                  | ۰/۶۸                   |
| (۴ و ۱)           | ۱           | ۰/۴۷              | ۰/۵۸                  | ۰/۷۸                   |
|                   | ۵           | ۰/۷۲              | ۰/۵۸                  | ۰/۷۲                   |
|                   | ۱۰          | ۱/۱۵              | ۰/۵۹                  | ۰/۷۷                   |
|                   | ۲۰          | ۱/۶۰              | ۰/۵۶                  | ۰/۷۵                   |
| (۸ و ۲)           | ۱           | ۰/۴۸              | ۰/۶۵                  | ۰/۸۸                   |
|                   | ۵           | ۰/۶۹              | ۰/۶۳                  | ۰/۸۲                   |
|                   | ۱۰          | ۱/۳               | ۰/۶۱                  | ۰/۸۰                   |
|                   | ۲۰          | ۱/۵۵              | ۰/۶۰                  | ۰/۸۱                   |

## ۴-۱-۲- اجرای الگوریتم ژنتیک بر روی داده‌های

## شبیه‌سازی

نتایج حاصل از اجرای  $GA$  در مجموعه داده‌های شبیه‌سازی، با استفاده از نقاط مختلف در جدول (۳) بررسی شده است. نرخ جهش  $۰/۱۸$ ، نخبه‌گرایی  $۱۱$  درصد و همچنین نرخ تقاطع بهینه  $۰/۸۶$  بهترین نتایج را تولید کرده است. تغییر تعداد جمعیت از  $۱۰۰$  به  $۱۳۰$  و سپس به  $۱۵۰$  تاثیر چندانی نمی‌گذارد. بنابراین

جمعیت اولیه  $۱۰۰$  تایی که منجر به یک پیاده‌سازی سریع‌تر می‌باشد، انتخاب شده است. تعداد تکرار کم مانند  $۵۰$ ، نتایج خوبی تولید نمی‌کند. با افزایش تعداد تکرار الگوریتم بهبود یافته است و بعد از یک حد مشخصی از تکرار، بهبود چندانی را نتیجه نمی‌دهد. در اینجا، تعداد تکرار  $۵۰۰$  انتخاب شده است. جدول (۳) مقادیر مختلف پارامترهای الگوریتم ژنتیک برای یک نقطه به مختصات (۲/۶ و ۸/۳) را نشان می‌دهد.

جدول ۳: کالیبره کردن پارامترهای  $GA$  با روش سعی و خطا بر روی داده شبیه‌سازی

| تعداد اجرا | تعداد تکرار | جمعیت اولیه | درصد نخبه‌گرایی | نرخ جهش | نرخ ترکیب | زمان اجرا (ثانیه) | تابع بهیگی (متر) |
|------------|-------------|-------------|-----------------|---------|-----------|-------------------|------------------|
| ۱          | ۵۰          | ۱۰۰         | ۱۱              | ۰/۱۸    | ۰/۸۶      | ۱/۳۱              | ۲/۴۵             |
| ۲          | ۱۰۰         | ۱۰۰         | ۱۱              | ۰/۱۸    | ۰/۸۶      | ۱/۵۴              | ۱/۹۵             |
| ۳          | ۱۵۰         | ۱۰۰         | ۱۱              | ۰/۱۸    | ۰/۸۶      | ۱/۷۹              | ۱/۴۵             |
| ۴          | ۲۰۰         | ۱۰۰         | ۱۱              | ۰/۱۸    | ۰/۸۶      | ۱/۹۸              | ۱/۱۳             |
| ۵          | ۲۵۰         | ۱۰۰         | ۱۱              | ۰/۱۸    | ۰/۸۶      | ۲/۲۳              | ۱/۰۱             |
| ۶          | ۳۰۰         | ۱۰۰         | ۱۱              | ۰/۱۸    | ۰/۸۶      | ۲/۴۶              | ۰/۹۲             |
| ۷          | ۴۰۰         | ۱۰۰         | ۱۱              | ۰/۱۸    | ۰/۸۶      | ۲/۹۱              | ۰/۹۰             |
| ۸          | ۵۰۰         | ۱۰۰         | ۱۱              | ۰/۱۸    | ۰/۸۶      | ۳/۵۱              | ۰/۸۹             |

#### ۴-۱-۳- اجرای الگوریتم بهینه سازی ازدحام

##### ذرات ترکیبی بر روی داده های شبیه سازی

نتایج حاصل از اجرای *HPSO* در مجموعه داده های شبیه سازی در جدول (۴) آمده است. نرخ حرکت اینرسی برابر با ۰/۱۲ بهترین نتایج را تولید کرده است. نرخ حرکت به سمت بهترین تجربه شخصی بین صفر و

۰/۹۵ و نرخ حرکت به سمت بهترین تجربه همسایگان هم بین صفر و ۰/۸۵ اعمال شده است. تعداد جمعیت اولیه برابر با ۱۰۰ ذره و تعداد تکرار ۵۰۰ انتخاب شده است. جدول (۴) مقادیر مختلف پارامترهای الگوریتم بهینه سازی ازدحام ذرات ترکیبی برای یک نقطه به مختصات (۲/۶ و ۸/۳) را نشان می دهد.

جدول ۴: کالیبره کردن پارامترهای الگوریتم *HPSO* با روش سعی و خطا بر روی داده شبیه سازی

| تعداد اجرا | تعداد تکرار | جمعیت اولیه | درصد حرکت اینرسی | درصد تجربه شخصی | درصد تجربه همسایگی | زمان اجرا (ثانیه) | تابع بهینگی (متر) |
|------------|-------------|-------------|------------------|-----------------|--------------------|-------------------|-------------------|
| ۱          | ۵۰          | ۱۰۰         | ۱۲               | [۰ ۰/۹۵]        | [۰ ۰/۸۵]           | ۱/۲۴              | ۲/۰۴              |
| ۲          | ۱۰۰         | ۱۰۰         | ۱۲               | [۰ ۰/۹۵]        | [۰ ۰/۸۵]           | ۱/۴۶              | ۱/۵۲              |
| ۳          | ۱۵۰         | ۱۰۰         | ۱۲               | [۰ ۰/۹۵]        | [۰ ۰/۸۵]           | ۱/۶۷              | ۱/۲۷              |
| ۴          | ۲۰۰         | ۱۰۰         | ۱۲               | [۰ ۰/۹۵]        | [۰ ۰/۸۵]           | ۱/۸۵              | ۱/۰۷              |
| ۵          | ۲۵۰         | ۱۰۰         | ۱۲               | [۰ ۰/۹۵]        | [۰ ۰/۸۵]           | ۲/۱۵              | ۰/۹۳              |
| ۶          | ۳۰۰         | ۱۰۰         | ۱۲               | [۰ ۰/۹۵]        | [۰ ۰/۸۵]           | ۲/۳۱              | ۰/۴۱              |
| ۷          | ۴۰۰         | ۱۰۰         | ۱۲               | [۰ ۰/۹۵]        | [۰ ۰/۸۵]           | ۲/۸۷              | ۰/۴۱              |
| ۸          | ۵۰۰         | ۱۰۰         | ۱۲               | [۰ ۰/۹۵]        | [۰ ۰/۸۵]           | ۳/۳۶              | ۰/۴۱              |

#### ۴-۱-۴- اجرای شبکه عصبی مصنوعی حل کننده

##### معادلات غیر خطی بر روی داده های شبیه سازی

نتایج حاصل از اجرای *NLANN* در مجموعه داده های شبیه سازی، با استفاده از پارامترها و نقاط مختلف در جدول (۵) بررسی شده است. پارامترهای مهم در شبکه عصبی شامل نرخ آموزش، تعداد تکرار، تعداد لایه ها و تعداد نورون های موجود در هر لایه می باشند. در شبکه عصبی پیشنهاد شده دو پارامتر تعداد لایه ها و تعداد نورون ها طبق ذات این نوع شبکه ثابت می باشند. این شبکه شامل چهار لایه می باشد که لایه پنهان اول دارای دو نورون و لایه پنهان دوم دارای چهار نورون هستند. پارامتر نرخ آموزش نیز با روش سعی و خطا ۰/۴ به دست آمده است که بهترین جواب ها را به دنبال داشته است. در جدول (۵) برای تکرارهای مختلف، دقت و زمان اجرای شبکه برای یک نقطه به مختصات (۲/۶ و ۸/۳) مورد بررسی قرار گرفته است. همان طور که ملاحظه می شود شبکه توانسته است نتایج خوبی را در

تکرارهای پایین ارائه دهد و از یک تکرار به بعد دقت تعیین موقعیت به صورت نوسانی می باشد. در این جا تعداد تکرار برابر با ۲۰۰ در نظر گرفته شده است.

#### ۴-۱-۵- مقایسه الگوریتم ها بر روی داده های شبیه سازی

معمولا برای مقایسه دقت الگوریتم ها در بحث های تعیین موقعیت از تابع توزیع احتمال<sup>۱</sup> (*CDF*) استفاده می شود. در شکل (۳) نمودارهای *CDF* هر چهار الگوریتم بر روی داده های شبیه سازی مقایسه شده اند. نمودار *CDF* الگوریتم *GA* در ۸۰ درصد موارد خطای کمتر از ۱/۵ متر را نتیجه داده است. الگوریتم *HPSO* در ۱۰۰ درصد موارد به دقت زیر ۲ متر و در ۶۰ درصد موارد خطای کمتر از ۱/۵ متر را نتیجه داده است. در روش گوس-نیوتون دقت زیر یک متر رسیده است. در روش گوس-نیوتون

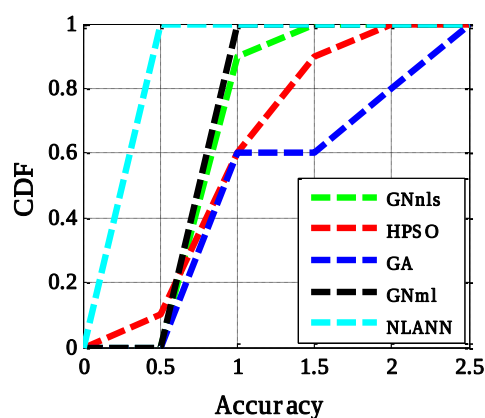
<sup>۱</sup> Cumulative distribution function

بهبود یافته است. در شبکه عصبی مصنوعی مبتنی بر حل معادلات غیرخطی در ۱۰۰ درصد موارد دقت به زیر ۰/۵ و در ۲۰ درصد موارد به دقت ۲۰ سانتی متر رسیده است که در مقایسه با سه الگوریتم دیگر، دقت تعیین موقعیت بهبود قابل ملاحظه‌ای داشته است.

با تابع بهینه کمترین مربعات در ۱۰۰ درصد موارد دقت به زیر ۱/۵ متر و در ۸۰ درصد موارد دقت به زیر یک متر رسیده است. درحالی که در روش گوس-نیوتون با تابع بهینگی بیشینه احتمال، در ۱۰۰ درصد موارد دقت به زیر یک متر و در ۶۰ درصد موارد دقت زیر ۰/۷۵ متر

جدول ۵: کالیبره کردن پارامترهای *NLANN* با روش سعی و خطا بر روی داده شبیه‌سازی

| تعداد اجرا | تعداد تکرار | نرخ آموزش | زمان اجرا (ثانیه) | تابع بهینگی (متر) |
|------------|-------------|-----------|-------------------|-------------------|
| ۱          | ۱           | ۰/۴       | ۰/۲۸              | ۲/۸۴              |
| ۲          | ۲           | ۰/۴       | ۰/۳۷              | ۱/۷۱              |
| ۳          | ۵           | ۰/۴       | ۰/۶۳              | ۰/۵۶              |
| ۴          | ۱۰          | ۰/۴       | ۱/۱۱              | ۰/۲۶              |
| ۵          | ۲۰          | ۰/۴       | ۱/۹۹              | ۰/۱۲              |
| ۶          | ۵۰          | ۰/۴       | ۴/۶۴              | ۰/۱۰              |
| ۷          | ۱۰۰         | ۰/۴       | ۸/۷۸              | ۰/۰۶              |
| ۸          | ۲۰۰         | ۰/۴       | ۱۷/۵۶             | ۰/۰۲              |



شکل ۳: مقایسه نمودارهای *CDF* بر روی داده‌های شبیه‌سازی

ایستگاه‌های معلوم باشد و یا رابطه‌ای نداشته باشد. به همین دلیل هر چهار الگوریتم با حالت‌های مختلفی از تعدادی ایستگاه‌های معلوم بر روی داده‌های واقعی از *GPS* ایتالیا مورد ارزیابی قرار گرفته‌اند.

#### ۴-۲- اجرای الگوریتم‌ها بر روی داده‌های واقعی

پس از کالیبره کردن پارامترهای الگوریتم‌ها بر روی داده‌های شبیه‌سازی، در این قسمت هر چهار روش بر روی داده‌های واقعی پیاده‌سازی شده‌اند. به‌طور کلی دقت تعیین موقعیت علاوه بر پارامترهای ذکر شده در هر یک از الگوریتم‌ها، می‌تواند وابسته به تعداد

همان طور که ملاحظه می‌شود، الگوریتم در تعداد فرستنده‌های بیشتر و با تکرارهای بالاتر، دقت بهتری را ارائه داده است. هم‌چنین با تغییر ترکیب فرستنده‌ها، نتایج در دقت برخی از نقاط متفاوت بوده است. بهترین دقت الگوریتم  $GA$ ،  $6/40$  متر می‌باشد.

#### ۴-۲-۳- اجرای الگوریتم بهینه‌سازی ازدحام ذرات ترکیبی بر روی داده‌های واقعی

جدول (۷) نتایج حاصل از  $HPSO$  بر روی شبکه مختصات  $GPS$  ایتالیا را نشان می‌دهد که در آن برای تعیین مختصات نقطه مجهول از حالت‌های مختلف تعداد ایستگاه‌های معلوم استفاده شده است. همانند الگوریتم  $GA$  در این جا نیز با افزایش تعداد فرستنده‌ها دقت بهبود یافته است. هم‌چنین ترکیب متفاوت از فرستنده‌ها در تعداد ثابت آن‌ها برای یک نقطه مجهول نتایج متفاوتی را نشان داده است. بهترین دقت الگوریتم  $HPSO$ ،  $1/10$  متر می‌باشد.

#### ۴-۲-۱- اجرای الگوریتم گوس-نیوتون بر روی داده‌های واقعی

اجرای الگوریتم  $GN$  بر روی داده‌های واقعی نتایج خوبی را حاصل نکرده است. در این الگوریتم با افزایش فواصل بین نقاط معلوم و مجهول، خطای تعیین موقعیت به صورت تصاعدی افزایش پیدا می‌کند و در مناطق بسیار بزرگ این روش بسیار نا کارآمد می‌باشد که داری خطای فاحشی است. بنابراین  $GN$  برای داده‌های  $GPS$  ایتالیا مناسب نمی‌باشد.

#### ۴-۲-۲- اجرای الگوریتم ژنتیک بر روی داده‌های واقعی

برای اجرای الگوریتم ژنتیک بر روی داده‌های واقعی از پارامترهای کالیبراسیون به دست آمده از داده‌های شبیه‌سازی استفاده شده است. جدول (۶) نتایج حاصل از  $GA$  بر روی شبکه  $GPS$  ایتالیا را نشان می‌دهد که برای تعیین مختصات نقطه مجهول از حالت‌های مختلف تعداد ایستگاه‌های معلوم استفاده شده است.

جدول ۶: زمان و دقت تعیین موقعیت  $GA$  بر روی داده‌های واقعی با فرستنده‌های متفاوت

| ۳۰ فرستنده |        | ۲۰ فرستنده |        | ۱۰ فرستنده |       | تعداد تکرار | مختصات نقاط مجهول   |
|------------|--------|------------|--------|------------|-------|-------------|---------------------|
| دقت        | زمان   | دقت        | زمان   | دقت        | زمان  |             |                     |
| ۳۲٫۸۳      | ۴۲٫۴۸  | ۵۵٫۴۳      | ۳۲٫۵۲  | ۹۵٫۱۰      | ۲۴٫۱۵ | ۱۰۰         | (۴۶۵۲۸۴۵ و ۱۲۱۷۳۹۱) |
| ۲۳٫۱۱      | ۷۱٫۸۴  | ۲۸٫۵۴      | ۵۲٫۱۷  | ۳۵٫۱۸      | ۳۱٫۱۷ | ۵۰۰         |                     |
| ۸٫۳۰       | ۱۴۳٫۱۹ | ۱۱٫۴۳      | ۱۰۲٫۴۷ | ۲۰٫۵۲      | ۷۵٫۱۴ | ۱۰۰۰        |                     |
| ۳۶٫۸۳      | ۴۶٫۴۹  | ۶۰٫۴۲      | ۳۴٫۴۳  | ۹۵٫۳۲      | ۲۶٫۵۴ | ۱۰۰         | (۴۶۳۴۸۹۹ و ۱۳۱۶۸۰۷) |
| ۲۵٫۸۴      | ۷۵٫۳۱  | ۳۴٫۵۶      | ۵۵٫۵۴  | ۳۷٫۳۷      | ۳۳٫۴۳ | ۵۰۰         |                     |
| ۱۴٫۵۹      | ۱۵۰٫۶۸ | ۱۴٫۵۲      | ۱۰۸٫۴۲ | ۲۲٫۶۶      | ۷۷٫۴۲ | ۱۰۰۰        |                     |
| ۳۰٫۵۱      | ۴۰٫۶۰  | ۵۱٫۴۳      | ۳۰٫۲۹  | ۸۹٫۱۳      | ۲۳٫۱۱ | ۱۰۰         | (۴۶۸۸۶۳۲ و ۱۳۴۳۱۸۹) |
| ۲۱٫۹۳      | ۶۹٫۸۱  | ۲۱٫۶۹      | ۴۹٫۱۳  | ۳۰٫۳۲      | ۲۹٫۱۶ | ۵۰۰         |                     |
| ۶٫۴۰       | ۱۳۵٫۹۳ | ۹٫۷۸       | ۹۸٫۳۷  | ۱۸٫۳۹      | ۷۰٫۱۸ | ۱۰۰۰        |                     |

جدول ۷: زمان و دقت تعیین موقعیت HPSO بر روی داده‌های واقعی با فرستنده‌های متفاوت

| مختصات نقاط مجهول   |  | تعداد تکرار | ۱۰ فرستنده |       | ۲۰ فرستنده |       | ۳۰ فرستنده |       |
|---------------------|--|-------------|------------|-------|------------|-------|------------|-------|
|                     |  |             | زمان       | دقت   | زمان       | دقت   | زمان       | دقت   |
| (۴۶۵۲۸۴۵ و ۱۲۱۷۳۹۱) |  | ۱۰۰         | ۱۵,۲۵      | ۹۰,۱۰ | ۲۸,۳۲      | ۵۰,۴۹ | ۳۶,۷۸      | ۲۴,۸۳ |
|                     |  | ۵۰۰         | ۲۷,۳۵      | ۲۵,۱۸ | ۴۶,۸۶      | ۲۲,۵۴ | ۶۲,۴۴      | ۱۵,۱۱ |
|                     |  | ۱۰۰۰        | ۷۰,۸۲      | ۱۵,۵۲ | ۹۴,۱۶      | ۵,۳۶  | ۱۱۲,۱۶     | ۶,۳۰  |
| (۴۶۳۴۸۹۹ و ۱۳۱۶۸۰۷) |  | ۱۰۰         | ۱۸,۹۲      | ۹۵,۷۲ | ۳۱,۶۳      | ۵۵,۶۲ | ۴۰,۸۷      | ۲۶,۴۵ |
|                     |  | ۵۰۰         | ۳۰,۶۳      | ۲۷,۳۲ | ۵۰,۱۴      | ۲۴,۴۵ | ۶۵,۱۳      | ۱۸,۳۶ |
|                     |  | ۱۰۰۰        | ۷۶,۸۲      | ۲۰,۵۱ | ۹۸,۶۳      | ۶,۵۸  | ۱۱۴,۵۴     | ۷,۱۳  |
| (۴۶۸۸۶۳۲ و ۱۳۴۳۱۸۹) |  | ۱۰۰         | ۱۴,۶۱      | ۸۵,۱۵ | ۲۷,۱۱      | ۴۵,۷۸ | ۳۴,۵۶      | ۲۰,۳۷ |
|                     |  | ۵۰۰         | ۲۵,۱۳      | ۲۲,۱۵ | ۴۳,۲۷      | ۱۷,۸۶ | ۶۰,۶۱      | ۵,۵۷  |
|                     |  | ۱۰۰۰        | ۶۷,۲۳      | ۱۳,۴۸ | ۹۰,۳۱      | ۳,۱۲  | ۱۱۰,۶۲     | ۱,۱۰  |

رسیده است. اما باز هم نتایج بهتری در تعداد فرستنده‌های بیشتر و در تکرارهای بالاتر به دست آمده‌اند. نتایج اجرای  $NLANN$  بر روی داده‌های واقعی بسیار محبوب‌تر از سه الگوریتم دیگر می‌باشد و دلیل آن هم می‌تواند فرموله‌سازی مسئله به صورت غیر خطی و آموزش شبکه برای مقادیر اولیه باشد. بهترین دقت  $NLANN$ ، ۰/۰۴ متر می‌باشد.

#### ۴-۲-۴-۴- اجرای شبکه عصبی حل کننده معادلات

##### غیر خطی بر روی داده‌های واقعی

در جدول (۸) شبکه عصبی پیشنهاد شده با ۱۰، ۲۰ و ۳۰ فرستنده مختلف برای داده‌های شبکه GPS ایتالیا مورد آنالیز قرار گرفته است. نتایج در دو پارامتر زمان آموزش و دقت مورد مقایسه قرار گرفته‌اند.  $NLANN$  در ترکیب‌های متفاوت از فرستنده‌ها، به دقت مناسبی

جدول ۸: زمان و دقت تعیین موقعیت  $NLANN$  برای داده‌های واقعی با فرستنده‌های متفاوت

| مختصات نقاط مجهول   |     | تعداد تکرار | ۱۰ فرستنده |       | ۲۰ فرستنده |       | ۳۰ فرستنده |  |
|---------------------|-----|-------------|------------|-------|------------|-------|------------|--|
| زمان                | دقت |             | زمان       | دقت   | زمان       | دقت   |            |  |
| (۱۲۱۷۳۹۱ و ۴۶۵۲۸۴۵) | ۵۰  | ۱۲,۱۷       | ۶۰,۶۸      | ۲۱,۵۰ | ۵۵,۶۰      | ۲۵,۱  | ۲۲,۸۴      |  |
|                     | ۱۰۰ | ۱۹,۳۴       | ۰,۳۸       | ۳۶,۴۸ | ۰,۴۵       | ۶۲,۷۳ | ۰,۰۴       |  |
|                     | ۲۰۰ | ۵۵,۲۶       | ۰,۰۹       | ۹۳,۰۱ | ۰,۲۱       | ۸۹,۸۳ | ۰,۰۵       |  |
| (۱۳۱۶۸۰۷ و ۴۶۳۴۸۹۹) | ۵۰  | ۱۳,۲۷       | ۵۵,۳۲      | ۲۳,۱۲ | ۵۱,۱۲      | ۲۷,۳۲ | ۲۵,۴۳      |  |
|                     | ۱۰۰ | ۲۰,۱۵       | ۰,۲۸       | ۳۷,۷۵ | ۰,۲۳       | ۶۴,۷۵ | ۰,۰۶       |  |
|                     | ۲۰۰ | ۵۸,۴۱       | ۰,۱۱       | ۹۴,۴۰ | ۰,۰۸       | ۹۱,۰۲ | ۰,۰۸       |  |
| (۱۳۴۳۱۸۹ و ۴۶۸۸۶۳۲) | ۵۰  | ۱۱,۴۹       | ۶۲,۸۴      | ۲۲,۱۳ | ۵۰,۹۵      | ۲۴,۱۴ | ۲۰,۴۳      |  |
|                     | ۱۰۰ | ۱۸,۲۸       | ۰,۳۴       | ۳۶,۱۲ | ۰,۱۸       | ۶۱,۰۳ | ۰,۰۷       |  |
|                     | ۲۰۰ | ۵۲,۵۶       | ۰,۰۸       | ۹۵,۴۱ | ۰,۱۳       | ۸۸,۹۲ | ۰,۰۴       |  |

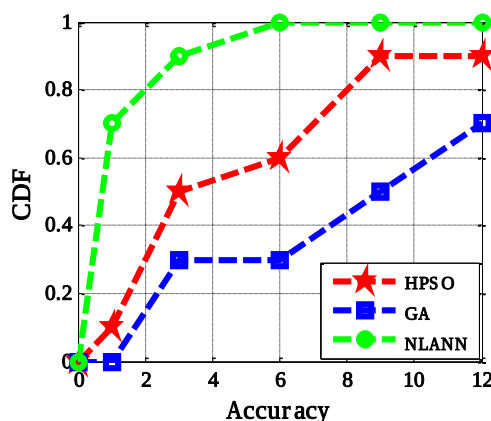
$CDF$  الگوریتم  $GA$  در ۷۰ درصد موارد خطای کمتر از ۱۲ متر و در ۳۰ درصد موارد خطای کمتر از ۶ متر را نتیجه داده است. الگوریتم  $HPSO$  در ۸۰ درصد موارد

#### ۴-۲-۵- مقایسه الگوریتم‌ها بر روی داده واقعی

در شکل (۴) نمودارهای  $CDF$  سه الگوریتم بر روی داده‌های واقعی مورد مقایسه قرار گرفته‌اند. نمودار

دقت بهتری دست یافته است که علت آن می‌تواند تعریف جستجو همسایگی برای ذره نخبه باشد. همچنین الگوریتم *HPSO* برای مسائل پیوسته، مدل‌سازی راحت و کاربردی‌تری نسبت به *GA* دارد. *NLANN* برای این مسأله دقت‌های بسیار خوبی را نسبت به سایر روش‌ها ارائه داده که علت آن می‌تواند آموزش شبکه در طی انجام محاسبات باشد.

به دقت زیر ۸ متر و در ۵۰ درصد موارد به دقت زیر ۳ متر رسیده است. در *NLANN* در ۹۰ درصد موارد دقت به زیر ۳ متر و در ۷۰ درصد موارد به زیر ۱ متر رسیده است که در مقایسه با الگوریتم‌های دیگر، دقت، بهبود قابل ملاحظه‌ای داشته است. همچنین شبکه در ۳۰ درصد موارد به دقت زیر نیم متر رسیده است که دقت خوبی می‌باشد. همان‌طور که از نتایج الگوریتم‌ها مشاهده می‌شود، الگوریتم *HPSO* به نسبت *GA* به

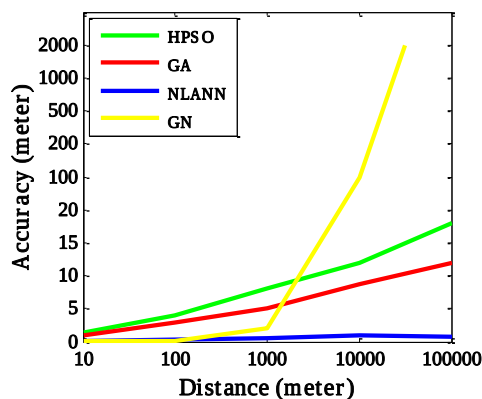


شکل ۴: مقایسه نمودارهای CDF الگوریتم‌ها برای داده‌های واقعی

*HPSO* برابر با سه متر و *GA* برابر با چهار متر است. با بزرگ‌تر شدن ابعاد منطقه دقت شبکه عصبی بسیار مناسب مانده است و الگوریتم‌های *GA* و *HPSO* دقت‌های ضعیف‌تری را ارائه داده‌اند. اما دقت *GN* با افزایش ابعاد منطقه به صورت تصاعدی افت کرده است. به طوری که در فضاهای صد هزار متری به دقت بالای دو هزار متر رسیده است. در حالی که دقت *NLANN* زیر یک متر و دقت *GA* و *HPSO* به ترتیب ۱۷ و ۱۲ متر می‌باشد.

#### ۴-۳- مقایسه دقت الگوریتم‌ها با تغییر ابعاد منطقه مطالعاتی

در این قسمت *SNR* محیط در تمام ابعاد برابر با ۳۰ در نظر گرفته شده است. همان‌طور که شکل (۵) نشان می‌دهد در فضای مربعی به طول ۱۰ متر تمام الگوریتم‌ها به دقت‌های زیر یک متر و شبکه عصبی به دقت زیر ۰/۱ متر دست یافته‌اند. در فضای ۱۰۰ متری دقت *NLANN* هم‌چنان مناسب و زیر ۰/۱ متر می‌باشد. *GN* هم دقت زیر یک متر را نتیجه داده است، اما دقت

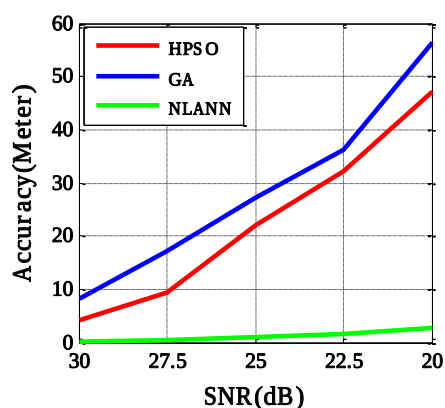


شکل ۵: مقایسه دقت تعیین موقعیت الگوریتم‌ها در ابعاد مختلف از منطقه مطالعاتی

#### ۴-۴- مقایسه دقت الگوریتم‌ها با $SNR$ های مختلف

$SNR$  همان نسبت سیگنال به نویز است که هر چقدر میزان آن بیشتر باشد، سیگنال دارای نویز کمتری بوده و خطای مشاهداتی بین فواصل کمتر است. همان‌طور که در شکل (۶) مشخص می‌باشد در حالت  $SNR$  برابر با ۳۰ که دارای کمترین خطای مشاهداتی می‌باشد، شبکه عصبی به دقت ۰/۰۸ متر،  $HPSO$  به دقت ۴/۲۳ و  $GA$  به دقت ۸/۱۸ متر دست یافته‌اند. در حالت ۲۵ دسی بل، شبکه عصبی هم‌چنان دقت مناسب زیر یک متر را

به‌دست آورده است. اما الگوریتم  $HPSO$  به دقت ۲۲/۱۲ متر و  $GA$  به ۲۷/۱۸ متر رسیده‌اند. با کاهش  $SNR$  در محیط به ۲۰ دسی بل دقت شبکه عصبی با بالاترین خطای مشاهداتی به ۲/۷ متر رسیده است. این در حالی است که دقت  $HPSO$  به ۴۷/۱۲ متر و  $GA$  به ۵۶/۳۱ متر رسیده‌اند. در واقع شبکه عصبی پیشنهاد شده در حالت بیشینه خطاهای مشاهداتی در محیط، باز هم نتایج بسیار بهتری را نسبت به الگوریتم‌های دیگر ارائه می‌دهد.



شکل ۶: مقایسه دقت تعیین موقعیت الگوریتم‌ها در  $SNR$  های مختلف

#### ۴-۵- مقایسه زمان اجرای الگوریتم‌ها

برای یک مقایسه دقیق‌تر، متوسط زمان ۳۰ اجرای مختلف از الگوریتم‌ها در نظر گرفته شده است. شرط خاتمه برای الگوریتم‌ها، رسیدن به یک مقدار مشخص از تابع بهینگی در برخی اجراهای پی‌درپی از الگوریتم‌ها می‌باشد. مقدار تابع بهینگی متناسب با هر روش برابر با بهترین دقت به‌دست آمده آن‌ها می‌باشد. متوسط زمان اجرا برای چهار الگوریتم در جدول (۹) ارائه شده است

جدول ۹: میانگین زمان اجرای الگوریتم‌ها

| الگوریتم | میانگین زمان اجرای الگوریتم‌ها برای داده‌های شبیه‌سازی (ثانیه) | میانگین زمان اجرای الگوریتم‌ها برای داده‌های واقعی (ثانیه) |
|----------|----------------------------------------------------------------|------------------------------------------------------------|
| GN       | ۲٫۹۱                                                           | ۴۳۷٫۳۱                                                     |
| GA       | ۵٫۰۲                                                           | ۲۲۶٫۸۳                                                     |
| HPSO     | ۴٫۶۲                                                           | ۱۸۰٫۳۷                                                     |
| NLANN    | ۱٫۰۷                                                           | ۱۱۲٫۲۶                                                     |

#### ۵- نتیجه‌گیری و پیشنهادات

ارائه یک مدل مناسب برای بهبود دقت تعیین موقعیت در محیط‌های بسته و شهری، از جمله اهداف مهم در تعیین موقعیت می‌باشد تا بتواند یک جایگزین مناسب برای سیستم‌های تعیین موقعیت جهانی باشد. در این پژوهش از یک شبکه عصبی مصنوعی حل‌کننده معادلات غیر خطی برای بهبود تعیین موقعیت محلی استفاده شده است. فواصل مشاهداتی برای ورودی الگوریتم‌ها از روش TOA به‌دست آمده است و از دو تابع بهینگی *NLS* و *ML* استفاده شده است. برای ارزیابی دقت و سرعت اجرای شبکه عصبی مصنوعی پیشنهاد شده، سه الگوریتم بهینه‌سازی *GN*، *GA* و *HPSO* به کار گرفته شده‌اند. الگوریتم‌ها ابتدا بر روی داده‌های شبیه‌سازی پیاده‌سازی شده‌اند و پس از اطمینان از صحت خروجی آن‌ها، در مرحله دوم بر روی داده‌های واقعی از شبکه *GPS* ایتالیا مدل‌سازی شده‌اند. نتایج نشان داده است که در هر دو نوع داده، مدل *NLANN* دقت تعیین موقعیت خوبی را در مدت زمان معقول به‌دست آورده است. در داده‌های شبیه‌سازی،

الگوریتم ژنتیک به دقت زیر ۲٫۵ متر، *HPSO* به دقت زیر دو متر، *GN* به دقت زیر ۱٫۵ متر رسیده‌اند. در حالی که *NLANN* به دقت زیر نیم متر رسیده است. همچنین در داده‌های واقعی *GA* در هفتاد درصد موارد به دقت زیر ۱۲ متر، *HPSO* در نود درصد موارد به دقت زیر ۹ متر و *NLANN* در هفتاد درصد موارد به دقت زیر یک متر رسیده است؛ در حالی که الگوریتم *GN* نتایج بسیار ضعیفی را در محدوده داده‌های واقعی ارائه داده است. با افزایش تعداد ایستگاه‌ها دقت تعیین موقعیت در تمامی الگوریتم‌ها بهبود پیدا کرده است. اما شبکه عصبی مصنوعی حتی در تعداد ایستگاه‌های کم نیز نتایج بسیار خوبی را حاصل می‌کند. بهترین دقت تعیین موقعیت برای داده‌های واقعی برابر با چهار سانتی‌متر می‌باشد که مربوط به *NLANN* با تعداد ۳۰ ایستگاه فرستنده است. برای ارزیابی دقیق‌تر الگوریتم‌ها، اجرای آن‌ها در ابعاد مختلف منطقه مطالعاتی مورد بررسی قرار گرفته‌اند. نتایج نشان داده‌اند که با افزایش ابعاد منطقه، دقت تعیین موقعیت کاهش پیدا می‌کند؛ اما همچنان دقت *NLANN* زیر یک متر

در شبکه GPS ایتالیا، نتایج متفاوت و جالبی را در دقت برخی از نقاط مجهول نشان داده است. علت این موضوع می‌تواند فاصله بین ایستگاه‌های انتخاب شده، توزیع مکانی این ایستگاه‌ها و حتی خاصیت ذاتی توابع اهداف کمترین مربعات و بیشینه احتمال باشد. زیرا در مسائل تعیین موقعیت با تابع هدف کمترین مربعات، این که کدام ایستگاه‌ها به‌عنوان نقطه معلوم و حتی نقطه مجهول انتخاب شوند، تأثیر به‌سزایی در دقت تعیین موقعیت و محاسبات سرشکنی دارند. هم‌چنین تأثیر هندسه و توزیع مکانی خاص از شبکه‌های GPS در تعیین موقعیت محلی و کشف الگوهای مکانی برای ترکیب نقاط معلوم هم نیاز به بررسی و تحقیق دارد. بنابراین شناخت این عوامل برای انتخاب نقاط معلوم و مجهول و بررسی علت تأثیر آن‌ها در تعیین موقعیت می‌تواند در موضوعات و تحقیقات آتی بررسی شوند.

بوده است. هم‌چنین الگوریتم‌ها با SNR های مختلف پیاده‌سازی شده‌اند که با کاهش SNR دقت تعیین موقعیت کم می‌شود. در مقادیر پایین SNR الگوریتم‌های GN، GA و HPSO نتایج ضعیفی را به‌دست آورده‌اند اما شبکه عصبی هم‌چنان دقت قابل قبولی دارد. البته دقت الگوریتم GN بسیار فاحش است و الگوریتم HPSO همواره دقت بهتری را نسبت به GA نشان می‌دهد که دلیل آن می‌تواند تعریف جستجو همسایگی برای ذره نخبه باشد.

در این تحقیق مدل NLANN هم از نظر دقت و زمان انجام محاسبات و هم کاربرد آن در تعیین موقعیت نقاط در فضاهای بزرگ، نسبت به سایر روش‌ها ارجح است. بنابراین کاربران شهری برای توسعه یک مدل مناسب در زمینه بهبود دقت تعیین موقعیت محلی در فضاهای شهری، می‌توانند از شبکه عصبی مصنوعی توسعه داده شده استفاده کنند.

مسئله دیگر در این تحقیق آن است که انتخاب ترکیب‌های متفاوت ۳۰ ایستگاه از ۱۴۰ ایستگاه موجود

## مراجع

- [1] R. M. Abreu, M. A. de Sousa, and M. R. Santos. A cost-effective local positioning system architecture based on TDoA. In *International Conference on Telecommunications*, Springer, Berlin, Heidelbergpp, pp 858-865, 2004.
- [2] Y. C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy characterization for metropolitan-scale Wi-Fi localization. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pp. 233-245, 2005.
- [3] H. Mehmood, and N.K. Tripathi. Cascading artificial neural networks optimized by genetic algorithms and integrated with global navigation satellite system to offer accurate ubiquitous positioning in urban environment. *Computers, Environment and Urban Systems*, 37, 35-44, 2013.
- [4] D. Madigan, E. Einahrawy, R. P. Martin, W. H. Ju, P. Krishnan, and A.S. Krishnakumar. Bayesian indoor positioning systems. *24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 2, pp. 1217-1227, 2005.
- [5] M. N. Husen, and S. Lee. Indoor human localization with orientation using WiFi fingerprinting. In *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, pp. 109, 2014.
- [6] A. Savvides, C. C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 166-179, 2001.
- [7] C. Y. Cheng. Indoor localization algorithm using clustering on signal and coordination

- pattern. *Annals of Operations Research*, 216(1), 83-99, 2014.
- [8] I. Guvenc, and C. C. Chong. A survey on TOA based wireless localization and NLOS mitigation techniques. *IEEE Communications Surveys & Tutorials*, 11(3), 107-124, 2009.
- [9] J. S. Leu, M. C. Yu, and H. J. Tzeng. Improving indoor positioning precision by using received signal strength fingerprint and footprint based on weighted ambient Wi-Fi signals. *Computer Networks*, 91, 329-340, 2015.
- [10] R. C. Eberhart, and Y. Shi. Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, Vol. 1, pp. 94-100, 2001.
- [11] A. Chehri, P. Fortier, and P.M. Tardif. UWB-based sensor networks for localization in mining environments. *Ad Hoc Networks*, 7(5), 987-1000, 2009.
- [12] B. H. Cheng, R. E. Hudson, F. Lorenzelli, L. Vandenberghe, and K. Yao. Distributed gauss-newton method for node localization in wireless sensor networks. In *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, pp. 915-919, 2005.
- [13] K. Yu, and I. Oppermann. UWB positioning for wireless embedded networks. In *Radio and Wireless Conference*, pp. 459-462, 2004.
- [14] F. Álvarez-Franco, H. González-Velasco, C. García-Orellana, M. Macías-Macías, and R. Gallardo-Caballero. Using GAs to obtain an optimal set of codes for an ultrasonic local positioning system. In *International Conference on Computer Aided Systems Theory*, Springer, Berlin, Heidelberg pp. 845-852, 2007.
- [15] H. Mehmood, and N.K. Tripathi. Optimizing artificial neural network-based indoor positioning system using genetic algorithm. *International Journal of Digital Earth*, 6(2), 158-184, 2013.
- [16] H. Zhu, B. Huang, Y. Tanabe, and T. Baba. Local Positioning with Artificial Neural Network and Time of Arrival Technique. In *2008 3rd International Conference on Innovative Computing Information and Control*, pp. 509-509, 2008.
- [17] H. Mehmood, and N. K. Tripathi. Cascading artificial neural networks optimized by genetic algorithms and integrated with global navigation satellite system to offer accurate ubiquitous positioning in urban environment. *Computers, Environment and Urban Systems*, 37, 35-44, 2013.
- [18] P. Claro, and N. B. Carvalho. Local positioning system based on artificial neural networks. In *International Conference on Artificial Neural Networks*, Springer, Berlin, Heidelberg, pp. 699-708, 2007.
- [19] S. Ngah, H. Zhu, K. T. Chen, Y. Tanabe, and T. Baba. Artificial neural network based model for local position systems, 2009.
- [20] K. G. Margaritis, M. Adamopoulos, K. Goulianas, and D. J. Evans. Artificial neural networks and iterative linear algebra methods. *PARALLEL ALGORITHM AND APPLICATIONS*, 3(1-2), 31-44, 1994.
- [21] A. Cichocki, and R. Unbehauen. Neural networks for solving systems of linear equations and related problems. *IEEE Transactions on Circuits and Systems I Fundamental Theory and Applications*, 39(2), 124-138, 1992.
- [22] K. Mathia, and R. Saeks. Solving nonlinear equations using recurrent neural networks. In *World congress on neural networks*, pp. 17-21, 1995.
- [23] D. Mishra, and P. K. Kalra. Modified Hopfield Neural Network Approach for Solving Nonlinear Algebraic Equations. *Engineering Letters*, 14(1), 2007.

- [24] G. Li, and Z. Zeng. A neural-network algorithm for solving nonlinear equation systems. In 2008 International Conference on Computational Intelligence and Security, Vol. 1, pp. 20-23, 2008.
- [25] A. Margaritis, and K. Goulianas. Finding all roots of  $2 \times 2$  nonlinear algebraic systems using back-propagation neural networks. *Neural Computing and Applications*, 21(5), 891-904, 2012.
- [26] Z. Nemec, and P. Bezousek. The Time Difference of Arrival Estimation of Wi-Fi Signals. *Radioengineering*, 17(4), 2008.
- [27] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6), 1067-1080, 2007.
- [28] R. Zekavat, and R. M. Buehrer. *Handbook of position location: Theory, practice and advances*. John Wiley & Sons, Vol. 27, 2011.
- [۲۹] M. Kaveh, M. S. Mesgari. Hospital site selection using hybrid PSO algorithm-Case study: District 2 of Tehran. *Sepehr*, pp. 7-22, 2019.
- [30] M. Kaveh, M. S. Mesgari, and R. S. Paland. Multiple criteria decision-making for hospital location-allocation based on improved genetic algorithm. *Applied Geomatics*, pp. 1-16, 2020.
- [31] M. Kaveh, and M. S. Mesgari. Improved biogeography-based optimization using migration process adjustment: An approach for location-allocation of ambulances. *Computers & Industrial Engineering*, 135, 800-813, 2019.
- [۳۲] S.M. Mosavi, M Kaveh, M Khisheh, M Aghababaei. Design and implementation a sonar data set classifier using multi-layer perceptron neural network trained by elephant herding optimization. *Darya Fonoon*, PP. 1-12, 2018.
- [33] M. R. Mosavi, M. Kaveh, M. Khishe, and M. Aghababaei. Design and Implementation a Sonar Data Set Classifier by using MLP NN Trained by Improved Biogeography-based Optimization. In *Proceedings of the Second National Conference on Marine Technology*, pp. 1-6, 2016.
- [34] M. R. Mosavi, M. Kaveh, and M. Khishe. Sonar Data Set Classification using MLP Neural Network Trained by Non-linear Migration Rates BBO. In *The Fourth Iranian Conference on Engineering Electromagnetic (ICEEM 2016)*, pp. 1-5, 2016.
- [35] M. Khishe, M. R. Mosavi, and M. Kaveh. Improved migration models of biogeography-based optimization for sonar dataset classification by using neural network. *Applied Acoustics*, 118, 15-29, 2017.
- [36] M. Kaveh, M. Khishe, and M. R. Mosavi. Design and implementation of a neighborhood search biogeography-based optimization trainer for classifying sonar dataset using multi-layer perceptron neural network. *Analog Integrated Circuits and Signal Processing*, 100(2), 405-428, 2019.
- [37] A. Lotfy, M. Kaveh, M. R. Mosavi, and A. R. Rahmati. An enhanced fuzzy controller based on improved genetic algorithm for speed control of DC motors. *Analog Integrated Circuits and Signal Processing*, 1-15, 2020.
- [38] A. Lotfy, M. Kaveh, M. R. Mosavi, and A. R. Rahmati. An Enhanced FPGA-based Implementation of Fuzzy Controller using a Personalized Microcontroller. *34<sup>th</sup> Power System Conference*, 2019.



## ***Solving the local positioning problem using a four-layer artificial neural network***

Mehrdad Kaveh <sup>1\*</sup>, Mohammad Saadi Mesgari <sup>2</sup>, Ali Khosravi<sup>3</sup>

1- M.Sc. Student of Geographic Information System in Department of Geodesy and Geomatics, College of Engineering, K.N.Toosi University of Technology.

2- Associate professor in Department of Geodesy and Geomatics, College of Engineering, K.N.Toosi University of Technology.

3- M.S.c student of Geodesy in Department of Geodesy and Geomatics, College of Engineering, K.N.Toosi University of Technology.

### **Abstract**

Today, the global positioning systems (GPS) do not work well in buildings and in dense urban areas when there is no lines of sight between the user and their satellites. Hence, the local positioning system (LPS) has been considerably used in recent years. The main purpose of this research is to provide a four-layer artificial neural network based on nonlinear system solver (NLANN) for local positioning problem. To evaluate the performance of artificial neural network, three methods of gauss-newton (GN), genetic algorithm (GA) and hybrid particle swarm optimization (HPSO) have been used. The results indicate that the proposed model has high accuracy. The accuracy of the artificial neural network on the simulated data is 0.05 m, while the best accuracy in other algorithms is about 0.45 meters. In the data of Italy's GPS network, the artificial neural network has been reached to accuracy below 10 cm in one minute. Also, artificial neural network has better accuracy in different dimensions of study area and different signal to noise ratio (SNR), and by increasing the number of stations, it has achieved good results in less time. Whereas other algorithms have not get well accuracy. However, the HPSO has better results related to GA and GN algorithms.

**Key words:** Local positioning system (LPS), artificial neural network based on nonlinear system solver (NLANN), gauss-newton (GN), genetic algorithm (GA), hybrid particle swarm optimization (HPSO).