

موقعیت‌یابی در اینترنت اشیا با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات مرتبه‌کسری آشوبی

علی ابراهیمی^۱، احمد حاجی پور^{۲*} و حمیدرضا توکلی^۳

اطلاعات مقاله	چکیده
دریافت مقاله: ۱۳۹۸/۰۲/۲۹ پذیرش مقاله: ۱۳۹۸/۰۶/۱۳	موقعیت‌یابی دقیق و سریع در اینترنت اشیا یک نیاز اساسی در گسترش استفاده از برنامه‌های کاربردی برای این شبکه‌هاست. به دلیل محدود بودن توان محاسباتی گره‌های حسگر و محدود بودن ظرفیت انرژی این گره‌ها، کاهش پیچیدگی محاسباتی و همچنین کاهش سربار ارتباطی در الگوریتم‌های موقعیت‌یابی برای این شبکه‌ها، اهمیت حیاتی دارد. در این مقاله، الگوریتمی مبتنی بر بهینه‌سازی ازدحام ذرات (PSO) به منظور موقعیت‌یابی دقیق و سریع گره‌های حسگر در اینترنت اشیا ارائه می‌شود. به‌طور کلی، قرار گرفتن در نقاط بهینه محلی و نرخ هم‌گرایی آهسته، دو ضعف اساسی الگوریتم PSO استاندارد است. در الگوریتم پیشنهاد شده با به‌کارگیری نظریه آشوب، از قرار گرفتن در نقاط بهینه محلی جلوگیری شده، هم‌گرایی کلی الگوریتم بهبود می‌یابد. به‌علاوه با بهره‌گرفتن از مشتقات مرتبه کسری، نرخ هم‌گرایی ذرات به سمت جواب بهینه تسریع می‌شود. ارزیابی کارایی الگوریتم با اجرای آن بر روی توابع آزمون خاص انجام می‌شود. نتایج شبیه‌سازی، بهبود کارایی الگوریتم پیشنهادی در موقعیت‌یابی گره‌های حسگر در اینترنت اشیا را نشان می‌دهند.
واژگان کلیدی: بهینه‌سازی ازدحام ذرات، مشتقات کسری، آشوب، اینترنت اشیا، تعیین موقعیت گره‌های حسگر.	

۱- مقدمه

اینترنت اشیا (IoT)، مجموعه‌ای از تجهیزات الکترونیکی مانند حسگرهاست که قابلیت اتصال به شبکه در آن‌ها وجود دارد، به‌طوری که این تجهیزات قادر به جمع‌آوری و تبادل اطلاعات با یکدیگر هستند [۱ و ۲]. کاربردهای بسیاری را می‌توان برای اینترنت اشیا نام برد که به‌عنوان نمونه به ردیابی هدف، نظارت بر خانه‌ها و شهرها، اتوماسیون، نظارت بر سلامت، مدیریت حمل‌ونقل و محیط زیست اشاره می‌شود [۳]. در تمام کاربردهای فوق، گره‌های حسگر به طور مداوم، بر محیط اطراف نظارت کرده، داده‌های حس شده را بر اساس نیاز، جمع‌آوری و ارسال می‌کنند. تعیین سریع محل دقیق گره‌های حسگر، پیش‌نیاز این

کاربردهاست؛ زیرا تفکیک داده‌های حس شده و استفاده از اطلاعات به‌دست‌آمده از گره‌ها بدون اطلاعات مکان حسگرها بسیار دشوار است [۴]. علاوه بر این، تعیین دقیق موقعیت حسگرها می‌تواند شرایط انجام مسیریابی جغرافیایی [۵]، تشخیص نفوذ [۶]، نظارت بر ترافیک [۷] و... را نیز فراهم کند.

روش معمول برای تعیین موقعیت، استفاده از سیستم موقعیت‌یاب جهانی است [۸]؛ اما این روش محدودیت‌هایی را به همراه دارد؛ به‌عنوان مثال اگر حسگرها در محیط‌های شهری یا محیط‌های داخلی مستقر شده باشند، به دلیل اینکه در این محیط‌ها سیگنال‌های دریافت شده از ماهواره‌ها به‌شدت تضعیف می‌شوند، تعیین موقعیت گره‌های حسگر

* پست الکترونیک نویسنده مسئول: a.hajipour@hsu.ac.ir

۱. کارشناسی ارشد، دانشکده برق و کامپیوتر، دانشگاه حکیم سبزواری

۲. استادیار، دانشکده برق و کامپیوتر، دانشگاه حکیم سبزواری

۳. استادیار، دانشکده برق و کامپیوتر، دانشگاه حکیم سبزواری

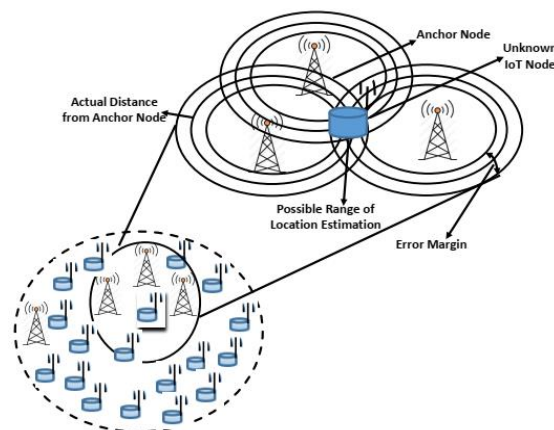
الگوریتم‌های الهام‌گرفته از بیوفیزیک، الگوریتم‌های کارآمد محاسباتی هستند که از آن‌ها به‌طور گسترده برای حل مسائل بهینه‌سازی استفاده می‌شود. یک الگوریتم جست‌وجوی مؤثر زیست‌محیطی برای تعیین مکان گره‌های حسگر در شبکه‌های حسگر بی‌سیم در [۱۸] پیشنهاد شده است که در آن، نویسندگان عملکرد الگوریتم جست‌وجوی فاخته را در کاهش میانگین خطای مکان‌یابی و افزایش نرخ هم‌گرایی نشان داده‌اند. در میان الگوریتم‌های زیستی، از الگوریتم بهینه‌سازی ازدحام ذرات (PSO) به دلیل سادگی و سهولت اجرا بیشتر استفاده شده است [۱۹-۲۵].

به‌طور کلی قرار گرفتن در نقاط بهینه محلی و همچنین نرخ هم‌گرایی آهسته، دو ضعف اصلی الگوریتم بهینه‌سازی ازدحام ذرات استاندارد است. روش‌های ترکیبی مختلفی برای جلوگیری از قرار گرفتن در نقاط بهینه محلی ارائه شده است. در [۲۶] با ترکیب الگوریتم بهینه‌سازی ازدحام ذرات و الگوریتم تبرید عملکرد الگوریتم بهینه‌سازی ازدحام ذرات به‌منظور جلوگیری از قرار گرفتن در نقاط بهینه محلی بهبود داده شده است.

در بسیاری از روش‌های پیشنهادی، نویسندگان تنها به یکی از این ضعف‌ها پرداخته‌اند و برطرف کردن هم‌زمان این دو نقطه ضعف، کمتر مورد توجه قرار گرفته است.

در این تحقیق، از دو ابزار به‌طور هم‌زمان برای کاهش نقاط ضعف الگوریتم بهینه‌سازی ازدحام ذرات استاندارد استفاده می‌شود. در این روش که بهینه‌سازی ازدحام ذرات مرتبه‌کسری آشوبی (FCPSO) نام‌گذاری می‌شود، با به‌کارگیری آشوب، هم‌گرایی کلی الگوریتم بهبود داده شده، همچنین از قرار گرفتن در نقاط بهینه محلی جلوگیری می‌گردد. به‌علاوه، از مشتقات مرتبه‌کسری برای تسریع نرخ هم‌گرایی ذرات به سمت جواب کلی بهینه استفاده می‌شود.

این مقاله به شرح زیر سازماندهی می‌شود: بخش دوم به معرفی الگوریتم بهینه‌سازی ازدحام ذرات مرتبه‌کسری آشوبی می‌پردازد. در بخش سوم، عملکرد الگوریتم پیشنهادی توسط ۵ تابع آزمون سنجیده می‌شود. بخش چهارم، به نحوه تعیین موقعیت گره‌های حسگر در اینترنت اشیا و شبیه‌سازی و بیان نتایج حاصل از الگوریتم‌ها می‌پردازد. بخش پایانی به جمع‌بندی مقاله اختصاص می‌یابد.



بسیار دشوار است [۹]. همچنین تجهیز گره‌های حسگر برای استفاده از سیستم موقعیت‌یابی جهانی، باعث افزایش هزینه شبکه می‌شود که پیاده‌سازی آن را دشوار می‌کند. اگر مختصات یک گره حسگر در اینترنت اشیا معلوم نباشد، در صورتی که این گره در برد فرستندگی حداقل سه گره دیگر که اطلاعات مکانی خود را دارند (گره‌های لنگر) باشد، موقعیت گره مذکور قابل تعیین خواهد بود (شکل ۱).

در این روش، تعیین موقعیت یک گره حسگر، شامل دو مرحله است: در مرحله اول که فاز دامنه نامیده می‌شود، یک گره حسگر در اینترنت اشیا برآورد فاصله خود را بر اساس شدت سیگنال دریافتی، زمان ورود سیگنال دریافتی، فاصله زمانی ورود سیگنال دریافتی و... برآورد می‌کند [۱۰]. نتایج به‌دست‌آمده در طی فاز دامنه، متأثر از عامل نویز بوده، ممکن است خطا داشته باشد [۱۱]. در مرحله دوم که فاز تخمین نامیده می‌شود، تعیین موقعیت گره حسگر در اینترنت اشیا با استفاده از اطلاعات فاز اول و با استفاده از الگوریتم‌های بهینه‌سازی رایج معمولی مانند حل یک مجموعه معادلات یا با استفاده از الگوریتم‌های بهینه‌سازی که خطای موقعیت‌یابی را به حداقل می‌رسانند، انجام می‌گیرد. الگوریتم‌های موقعیت‌یابی بسیاری به‌منظور افزایش دقت موقعیت‌یابی و افزایش طول عمر گره‌های حسگر پیشنهاد شده‌اند [۱۲-۱۷]. در مراجع فوق، نویسندگان پیشنهاد داده‌اند موقعیت‌یابی گره‌های حسگر شامل دو مرحله است: مرحله تقسیم‌بندی و مرحله اصلاح مکان‌یابی. در مرحله تقسیم‌بندی، ابتدا منطقه هدف به قسمت‌های کوچک تقسیم می‌شود. سپس در مرحله اصلاح مکان‌یابی، با استفاده از یک الگوریتم و اطلاعات گره‌های مرجع، موقعیتی دقیق‌تر برای گره‌ها مشخص می‌شود.

طبیعی r معرف بعد فضای جست‌وجوی مسئله است. متغیرهای $v[t]$ و $x[t]$ به ترتیب بردارهای سرعت و موقعیت ذره در تکرار t ام و $\chi_1[t]$ و $\chi_2[t]$ به ترتیب مؤلفه‌های بهترین موقعیت شناختی و بهترین موقعیت اجتماعی هستند.

۲-۲- الگوریتم ازدحام ذرات آشوبی

الگوریتم ازدحام ذرات آشوبی، ترکیبی از الگوریتم بهینه‌سازی ازدحام ذرات استاندارد و یک نگاشت آشوبی است که در آن، این نگاشت، در فرایند مقداردهی اولیه و همچنین فرایند تکامل الگوریتم بهینه‌سازی ازدحام ذرات اثرگذاری دارد [۲۸-۳۱]. آشوب، پدیده‌ای در سیستم‌های غیرخطی است که شامل بی‌شمار حرکت دوره‌ای ناپایدار است. به عبارت دیگر، آشوب، یک رفتار تصادفی‌گونه در یک سیستم غیرخطی معین است. یک نگاشت آشوبی، یک سیستم دینامیکی گسسته زمان [۳۲] مطابق زیر است:

$$x_k = f(x_{k-1}), \quad k = 1, 2, 3, \dots \quad (2)$$

از نگاشت آشوبی می‌توان برای تولید دنباله‌های اعداد استفاده کرد. این دنباله‌ها که به دنباله‌های آشوبی شناخته می‌شوند، مشخصات نگاشت آشوبی مانند تصادفی بودن، مستقل از زمان بودن و منظم بودن را دارا هستند و بر این اساس، هیچ حالتی در آن‌ها دوباره تکرار نمی‌شود. دنباله‌های آشوبی به‌عنوان دنباله‌های تصادفی در نظر گرفته شده، به‌عنوان پارامتر تصادفی در الگوریتم بهینه‌سازی ازدحام ذرات آشوبی به کار گرفته می‌شوند.

به این ترتیب، دنباله‌های آشوبی، ابزاری مناسب برای کنترل الگوریتم بهینه‌سازی ازدحام ذرات هستند و از قرار گرفتن الگوریتم در یک نقطه بهینه محلی در هنگام فرایند جست‌وجو و همچنین از پدیده هم‌گرایی زود هنگام که از مشکلات الگوریتم استاندارد است، جلوگیری می‌کنند. تاکنون نگاشت‌های آشوبی بسیاری برای بهبود الگوریتم بهینه‌سازی ازدحام ذرات معرفی شده‌اند [۳۳ و ۳۴]. برای الگوریتم پیشنهاد شده در این مقاله، از نگاشت آشوبی لجستیک برای تولید یکنواخت ذرات استفاده می‌شود تا کیفیت تولید جمعیت اولیه بهبود یابد و از قرار گرفتن در نقاط بهینه محلی جلوگیری شود. در شبه کد زیر، تولید جمعیت اولیه با استفاده از نگاشت آشوبی (لجستیک) آورده شده است:

۲- الگوریتم ازدحام ذرات مرتبه کسری آشوبی

در این بخش، ابتدا الگوریتم ازدحام ذرات بیان شده، سپس الگوریتم ازدحام ذرات آشوبی ارائه می‌گردد و در نهایت، با استفاده از تئوری مشتقات مرتبه کسری، الگوریتم ازدحام ذرات مرتبه کسری فرمول‌بندی می‌شود.

۲-۱- الگوریتم ازدحام ذرات

الگوریتم بهینه‌سازی ازدحام ذرات استاندارد، یک روش بهینه‌سازی ذرات است که در ابتدا توسط کندی و ابراهارت [۲۷] پیشنهاد شده است. این الگوریتم با مجموعه‌ای از راه‌حل‌های تصادفی شروع می‌شود و با به‌روزرسانی نسل‌ها، جواب بهینه را جست‌وجو می‌کند. دو فرایند یادگیری در بهینه‌سازی ازدحام ذرات وجود دارد: فرایند یادگیری شناختی که مبتنی بر تاریخچه و گذشته هر ذره است. فرایند یادگیری اجتماعی که مبتنی بر تاریخچه و گذشته همه ذرات است و از مبادله اطلاعات میان تمام ذرات موجود در جمعیت حاصل می‌شود. ذرات در فضای جست‌وجوی چندبعدی جابه‌جا می‌شوند. در حین حرکت، هر ذره موقعیت خود را با توجه به تجربه خود و تجربه ذرات همسایه به‌روزرسانی می‌کند و پس از دریافت اطلاعات مربوط به بهترین موقعیت خود و بهترین موقعیت همسایگان، بر اساس یک تابع ریاضی، موقعیت خود را به‌روزرسانی می‌کند. هر ذره، مختصات خود را در فضای حل مسئله جست‌وجو می‌کند که بهترین جوابی را که تاکنون به دست آمده، به‌عنوان بهترین مکان ذره (pbest) در نظر می‌گیرد. در همین حال، همه ذرات به سوی بهترین ذره‌ها (gbest) حرکت می‌کنند. در هر تکرار، بهینه‌سازی شامل تغییرات سرعت هر ذره به سمت مکان‌های pbest و gbest است. مدل ریاضی حرکت ذرات به صورت روابط (۱) و (۲) است:

$$\begin{aligned} v[t+1] &= wv[t] + \rho_1 r_1 (\chi_1[t] - x[t]) + \rho_2 r_2 (\chi_2[t] - x[t]) \\ x[t+1] &= x[t] + v[t+1] \end{aligned} \quad (1)$$

که در آن، پارامترهای w ، ρ_1 و ρ_2 وزن‌هایی حقیقی و مثبت هستند که اثر اینرسی را نشان می‌دهند. بهترین مؤلفه شناختی (بهترین مکان ذره) و بهترین مؤلفه اجتماعی (بهترین مکان ذره‌ها) در هنگام تعیین سرعت جدید $v[t+1]$ برای یک ذره مشخص است. ضرایب r_1 و r_2 بردارهای تصادفی r بعدی هستند که در آن هر مؤلفه یک متغیر تصادفی با توزیع یکنواخت بین صفر و یک بوده، عدد

پایداری الگوریتم بهینه‌سازی ازدحام ذرات با اعمال مشتقات مرتبه کسری در [۳۶] ثابت شده است. شبه کد الگوریتم بهینه‌سازی ازدحام ذرات مرتبه کسری آشوبی به صورت زیر است:

Algorithm 2: Pseudo-code of FCPSO

- 1: Initialize ρ_1, ρ_2, w and maximum number of iteration t_{max} .
- 2: Initialize $X_{min}, X_{max}, V_{min}$ and V_{max}
- 3: Initialize α and h
- 4: Generate the particles for HFPSO (Total Number of particles is N).
- 6: For each particle: initialize velocity (V) as zero and position (X) by logistic map then evaluate objective function.
- 7: Update personal best and global best
- 8: $k = 1$
- 9: For $t = 1$ to t_{max}
- 10: For each particle do:
- 11: calculate:

$$V(t+1) = (\alpha + h^\alpha w)V(t) + h^\alpha \rho_1 r_1 (\chi_1(t) - X(t)) + h^\alpha \rho_2 r_2 (\chi_2(t) - X(t))$$
- 12: Apply Velocity Limits.
- 13: calculate: $x(t+1) = \alpha x(t) + \frac{1}{h^\alpha} v(t+1)$
- 14: Apply Position Limits.
- 15: Evaluate objective function.
- 16: Update personal best and global best.
- 17: End For.
- 18: End For.
- 19: Declare global best.

نکته مهم در تعیین مقدار بهینه مرتبه مشتق کسری، این است که این مقدار، وابسته به مسئله بهینه‌سازی است و برای هر مسئله، مقدار مشخصی دارد. با توجه به این نکته، در ادامه، شبه کد تعیین مقدار بهینه مشتق مرتبه کسری برای الگوریتم FCPSO ارائه شده است.

Algorithm 3: Pseudo-code for determination of optimal fractional order value

1. Begin
2. Initialize $\alpha = 0$
3. Run FCPSO algorithm
4. Declare global best as best response
5. $\alpha = \alpha + 0.1$
6. Run FCPSO algorithm
7. If global best better than best response, declare global best as best response and α as best α
8. If $\alpha < 1$ go to 5
10. Declare best response and best α
11. END

۳- نتایج شبیه‌سازی الگوریتم‌های بهینه‌سازی ازدحام ذرات بر روی توابع آزمون

برای نشان دادن کارایی الگوریتم ازدحام ذرات مرتبه کسری آشوبی، نتایج حاصل از آن با الگوریتم‌های ازدحام ذرات

Algorithm 1: Pseudo-code of Initialization by Logistic Map

1. Begin
2. Randomly initialize chaotic variables
3. While (number of maximal iterations is not met)
4. If chaotic variable plunges into the fixed points or the small periodic cycles
5. Implement a very small positive random perturbation (0,0,25,0.5,0.75,1)
6. Map them by (0.1,0.26,0.51,0.76,0.9)
7. Else
8. Update the variables by $x_{n+1} = \mu x_n (1 - x_n), n = 1, 2, 3, \dots$ directly
9. End
10. Next generation until stopping criterion
11. Remap the chaotic variables into the optimization problem space

۲-۳ الگوریتم ازدحام ذرات مرتبه کسری آشوبی

پس از تولید جمعیت اولیه در الگوریتم، باید موقعیت و سرعت حرکت ذرات به‌روزرسانی شود. در این مقاله، از روش مشتقات مرتبه کسری برای مدل‌سازی حرکت ذرات استفاده می‌شود. به‌طور کلی، معادلات الگوریتم بهینه‌سازی ازدحام ذرات در شکل گسسته به صورت زیر است:

$$v[t+1] = wv[t] + \sum_{i=1}^2 \rho_i r_i (\chi_i[t] - x[t]), \quad (4)$$

$$x[t+1] = v[t]$$

و شکل پیوسته معادلات فوق به صورت زیر است:

$$\frac{d}{dt} v[t] = wv[t] + \sum_{i=1}^2 \rho_i r_i (\chi_i[t] - x[t]), \quad (5)$$

$$\frac{d}{dt} x[t] = v[t]$$

از آنجایی که ذرات به صورت زیگ‌زاگ حرکت می‌کنند، به دلیل پیچیدگی حرکت ذرات و وابستگی حرکت آن‌ها به موقعیت قبلی، از مشتقات مرتبه کسری که توصیف بهتری از گذشته بلندمدت حرکت ذرات دارند، استفاده می‌شود [۳۶]. معادلات فوق با استفاده از مدل مشتق مرتبه کسری کاپوتو به صورت زیر نوشته می‌شوند:

$${}^c D^\alpha v[t] = wv[t] + \sum_{i=1}^2 \rho_i r_i (\chi_i[t] - x[t]), \quad (6)$$

$${}^c D^\alpha x[t] = v[t]$$

با گسسته‌سازی مشتق مرتبه کسری کاپوتو برای v و x ، سرعت و مکان ذرات بر اساس مرتبه مشتق کسری (α) به صورت زیر است:

$$v(t+1) = (\alpha + h^\alpha w)v(t) + h^\alpha \rho_1 r_1 (\chi_1(t) - x(t)) + h^\alpha \rho_2 r_2 (\chi_2(t) - x(t)) \quad (7)$$

$$x(t+1) = \alpha x(t) + \frac{1}{h^\alpha} v(t+1)$$

نتایج شبیه‌سازی تولید جمعیت اولیه به کمک آشوب با توابع هدف Sphere، Rastrigin، Ackley، Rosenbrock و Schwefel و مدل کردن سرعت و مکان ذرات با به‌کارگیری مشتق مرتبه کسری و به‌ازای مرتبه‌های کسری مختلف در جدول‌های ۲، ۳، ۴، ۵ و ۶ آورده شده است. همچنین شکل‌های (۲) تا (۹) ارزیابی توابع Sphere و Rastrigin را با سه الگوریتم به‌ازای مرتبه‌های مختلف مشتق کسری نشان می‌دهند.

استاندارد و ازدحام ذرات مرتبه کسری مقایسه شده است. تمام شبیه‌سازی‌ها در سیستم عامل ویندوز ۱۰ با استفاده از پردازنده‌های چهارهسته‌ای، فرکانس کاری ۳/۲ گیگاهرتز و با ظرفیت حافظه چهار گیگابایت انجام می‌شود. شبیه‌سازی در نرم‌افزار متلب ۲۰۱۶ انجام می‌گیرد. برای بررسی کارایی الگوریتم پیشنهادی، از پنج تابع آزمون [۳۷] استفاده شده است که معادلات آن‌ها و محدوده جست‌وجو برای آن‌ها در جدول ۱ آورده شده است. در همه توابع، نقطه هدف، نقطه بهینه صفر و ابعاد همه توابع $r = 50$ است.

جدول ۱: توابع آزمون

نام تابع	رابطه	محدوده جست‌وجو
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	$-\infty \leq x_i \leq \infty$ $1 \leq i \leq n$
Rastrigin	$f(x) = An \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)]$	$-5.12 \leq x, y \leq 5.12$
Ackley	$f(x, y) = -20 \exp[-0.2 \sqrt{0.5(x^2 + y^2)}] - \exp[0.5(\cos 2\pi x + \cos 2\pi y)] + 20$	$-5 \leq x, y \leq 5$
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-\infty \leq x_i \leq \infty$ $1 \leq i \leq n$
Schwefel	$f(x) = 418.9829 - \sum_{i=1}^d x_i \sin(\sqrt{ x_i })$	$x \in [-500, 500]$

جدول ۲: نتیجه اجرای الگوریتم‌های ازدحام مرتبه کسری و مرتبه کسری آشوبی در تابع Rastrigin به‌ازای مرتبه‌های مختلف مشتق (مقدار تابع هدف و زمان اجرای این تابع با الگوریتم بهینه‌سازی ازدحام ذرات استاندارد به ترتیب ۰/۹۹۵۳ و ۰/۱۰۹۱۴۷ است)

FCPSO		FPSO		مرتبه مشتق کسری
زمان اجرا	مقدار تابع هدف	زمان اجرا	مقدار تابع هدف	
۰/۱۰۱۲۳۱	۱/۴۶۸۲e-۵	۰/۰۸۹۲۲۸	۱/۰۱۳۷	۰/۲
۰/۰۹۶۹۴۰	۰/۰۰۳۴	۰/۰۹۶۲۲۱	۰/۹۸۰۹	۰/۴
۰/۱۰۵۹۹۰	۲/۴۸۴۴e-۶	۰/۱۰۹۷۷۷	۰/۰۵۲۷	۰/۶
۰/۰۸۴۰۷۶	۰/۰۲۱۷	۰/۰۹۹۳۹۱	۳/۹۸۷۵	۰/۸

جدول ۳: نتیجه اجرای الگوریتم‌های ازدحام مرتبه کسری و مرتبه کسری آشوبی در تابع Spher به‌ازای مرتبه‌های مختلف مشتق (مقدار تابع هدف و زمان اجرای این تابع با الگوریتم بهینه‌سازی ازدحام ذرات استاندارد به ترتیب ۶/۲۲۷۹e-۳۹ و ۳/۵۸۳۰۶۵ است)

FCPSO		FPSO		مرتبه مشتق کسری
زمان اجرا	مقدار تابع هدف	زمان اجرا	مقدار تابع هدف	
۰/۲۳۶۰۱۴	۳/۷۲۱۱e-۵۷	۰/۴۷۲۷۹۵	۱/۱۰۶۵e-۴۰	۰/۲
۰/۰۸۱۹۰۳	۶/۴۲۸۵e-۴۹	۰/۰۹۶۶۲۷	۷/۳۶۳۱e-۳۷	۰/۴
۰/۰۹۳۲۷۳	۵/۵۴۰۶e-۴۰	۰/۰۶۸۴۱۵	۱/۲۲۹۵e-۳۴	۰/۶
۰/۱۱۲۴۲۶	۶/۳۰۰۹e-۴۰	۰/۰۶۶۶۷۶	۰/۰۰۱۵	۰/۸

جدول ۴: نتیجه اجرای الگوریتم‌های ازدحام مرتبه کسری و مرتبه کسری آشوبی در تابع Ackley به‌ازای مرتبه مختلف مشتق (مقدار تابع هدف و زمان اجرای این تابع با الگوریتم بهینه‌سازی ازدحام ذرات استاندارد به‌ترتیب ۱۵-۲/۷۱۹۹e و ۰/۱۱۳۳۹۲ است)

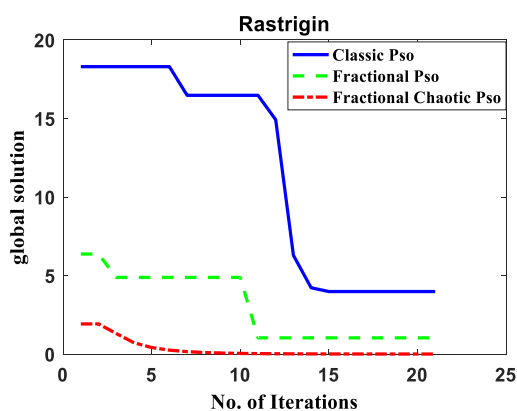
FCPSO		FPSO		مرتبه مشتق کسری
زمان اجرا	مقدار تابع هدف	زمان اجرا	مقدار تابع هدف	
۰/۰۹۶۲۴۳	۳/۶۷۷۵e-۱۶	۰/۰۶۸۲۳۹	۴/۹۲۴۵e-۱۰	۰/۲
۰/۰۸۳۸۴۳	۳/۵۳۳۴e-۱۷	۰/۰۶۴۳۵۲	۱/۰۴۹۴e-۱۶	۰/۴
۰/۰۸۶۸۰۸	۱/۷۸۶۴e-۱۶	۰/۰۶۴۹۳۵	۹/۰۷۱۷e-۱۴	۰/۶
۰/۰۷۵۶۵۴	۳/۵۳۶۱e-۱۴	۰/۰۵۵۷۸۸	۰/۰۰۱۲	۰/۸

جدول ۵: نتیجه اجرای الگوریتم‌های ازدحام مرتبه کسری و مرتبه کسری آشوبی در تابع Rosenbrock به‌ازای مرتبه مختلف مشتق (مقدار تابع هدف و زمان اجرای این تابع با الگوریتم بهینه‌سازی ازدحام ذرات استاندارد به‌ترتیب ۱۸/۹۲۴۰ و ۰/۱۴۵۹۹۷ است)

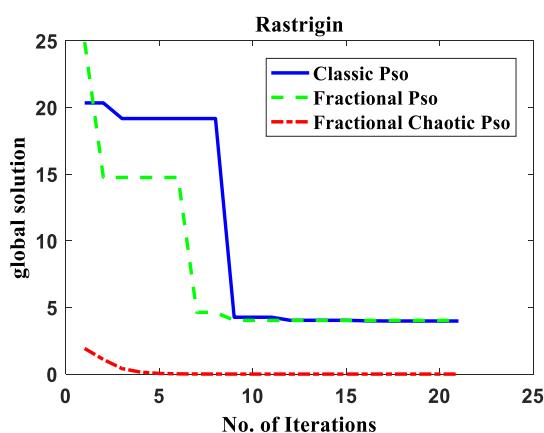
FCPSO		FPSO		مرتبه مشتق کسری
زمان اجرا	مقدار تابع هدف	زمان اجرا	مقدار تابع هدف	
۰/۰۹۴۴۳۵	۰/۱e-۱۱	۰/۰۵۹۳۲۷	۳/۵۵۶۸e-۹	۰/۲
۰/۰۹۲۶۶۹	۰/۱e-۱۱	۰/۰۵۹۶۸۳	۸/۸۴۰۵e-۶	۰/۴
۰/۱۰۰۸۱۶	۰/۱e-۱۱	۰/۰۶۴۳۹۹	۶/۷۲۱۶e-۴	۰/۶
۰/۰۹۲۳۲۴	۰/۱e-۱۱	۰/۰۵۹۴۳۷	۳/۱۲۱۱e-۴	۰/۸

جدول ۶: نتیجه اجرای الگوریتم‌های ازدحام مرتبه کسری و مرتبه کسری آشوبی در تابع Schwefel به‌ازای مرتبه مختلف مشتق (مقدار تابع هدف و زمان اجرای این تابع با الگوریتم بهینه‌سازی ازدحام ذرات استاندارد به‌ترتیب ۴۲۰/۹۶۸۷ و ۰/۱۰۰۵۷۲ است)

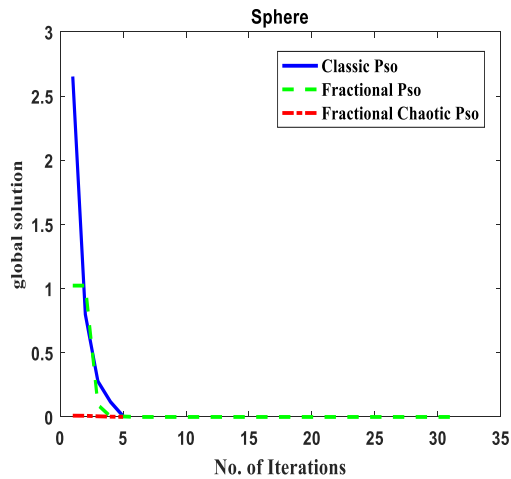
FCPSO		FPSO		مرتبه مشتق کسری
زمان اجرا	مقدار تابع هدف	زمان اجرا	مقدار تابع هدف	
۰/۰۷۸۲۴۵	۴۲۱/۰۵۴e-۱۶	۰/۰۶۱۸۸۸	۷/۲۷۰۳e-۱۰	۰/۲
۰/۰۸۷۵۰۱	۴۲۱/۱۴۶e-۱۵	۰/۰۶۴۹۱۳	۱/۰۴۴۱e-۶	۰/۴
۰/۰۸۷۹۹۵	۴۲۰/۹۲۸۳e-۱۵	۰/۰۶۹۷۵۸	۳/۳۰۶۱e-۵	۰/۶
۰/۰۸۵۵۲۲	۳۰۲/۵۲۷۹e-۱۵	۰/۰۶۲۵۷۷	۰/۳۴e-۵	۰/۸



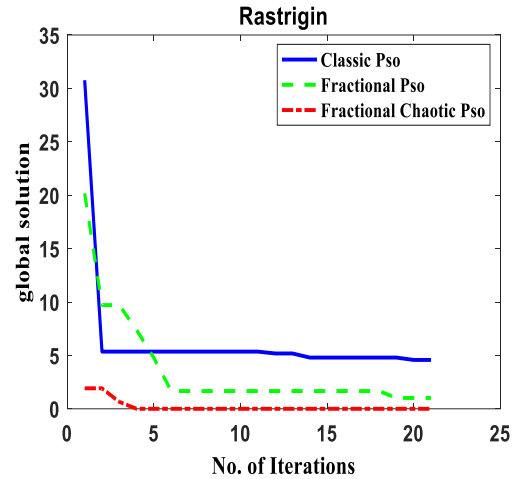
شکل ۳: ارزیابی تابع هدف Rastrigin با سه الگوریتم به‌ازای $\alpha = 0.4$



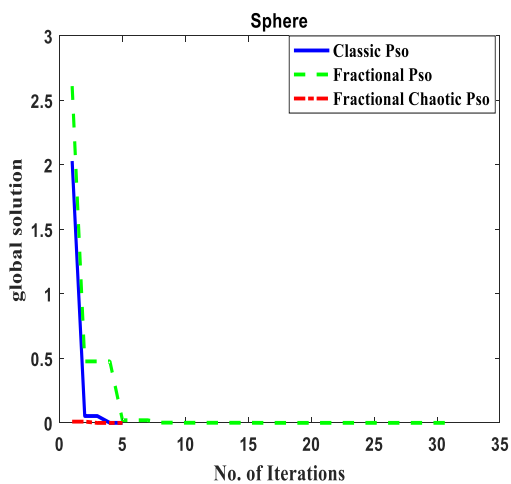
شکل ۲: ارزیابی تابع هدف Rastrigin با سه الگوریتم به‌ازای $\alpha = 0.2$



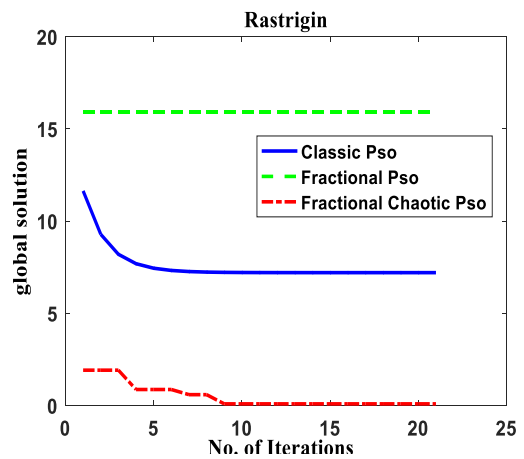
شکل ۷: ارزیابی تابع هدف Sphere با سه الگوریتم به‌ازای $\alpha = 0.4$



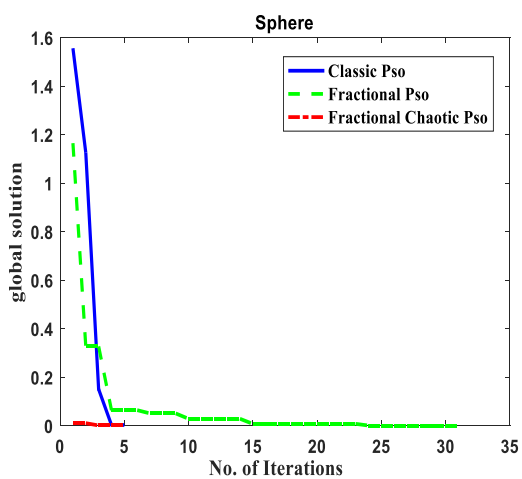
شکل ۴: ارزیابی تابع هدف Rastrigin با سه الگوریتم به‌ازای $\alpha = 0.6$



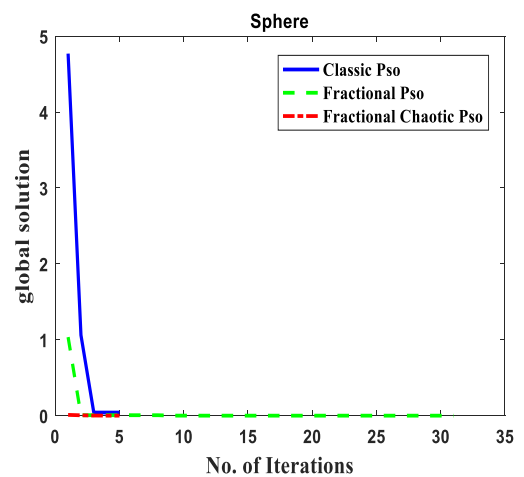
شکل ۸: ارزیابی تابع هدف Sphere با سه الگوریتم به‌ازای $\alpha = 0.6$



شکل ۵: ارزیابی تابع هدف Rastrigin با سه الگوریتم به‌ازای $\alpha = 0.8$



شکل ۹: ارزیابی تابع هدف Sphere با سه الگوریتم به‌ازای $\alpha = 0.8$



شکل ۶: ارزیابی تابع هدف Sphere با سه الگوریتم به‌ازای $\alpha = 0.2$

بوده، در غیر این صورت، این گره قابل مکان‌یابی نیست. با توجه به پیاده‌سازی ساده، پیچیدگی سخت‌افزاری کم و همچنین هزینه پایین، در این مقاله رویکرد فاصله‌یابی مبتنی بر RSSI برای تخمین فاصله بین گره‌های همسایه در نظر گرفته شده است. اندازه‌گیری فاصله در RSSI را می‌توان بدون مصرف انرژی اضافی انجام داد که این روش برای گره‌های حسگر در اینترنت اشیا که انرژی محدود دارند، مفید خواهد بود. در این روش با فرض اینکه توان سیگنال ارسالی P_{Tx} باشد و مدل افت مسیر و ضریب افت مسیر μ مشخص باشند، گیرنده می‌تواند از توان سیگنال دریافتی P_{Rcvd} برای محاسبه فاصله \hat{d}_{nm} بین خودش و فرستنده، از رابطه زیر استفاده کند:

$$\hat{d}_{nm} = \sqrt{\frac{\mu c P_{Tx}}{P_{Rcvd}}}$$

البته اشکال این روش آن است که مقادیر اندازه‌گیری‌شده از روش RSSI ثابت نیستند. حتی اگر فرستنده و گیرنده نیز ثابت باشند و حرکت نکنند، مقدار توان دریافتی، پیوسته در حال تغییر است. این مسئله که به دلیل پدیده محوشدگی^۱ و تغییر مشخصات کانال مخابراتی رخ می‌دهد، خطای فاصله‌یابی نامیده می‌شود.

اگر d_{nm} فاصله واقعی بین گره حسگر n ام با مختصات مکانی نامعلوم (x_n, y_n) و گره لنگر m ام با مختصات مکانی (x_m, y_m) باشد، می‌توان آن را از رابطه اقلیدسی زیر به دست آورد:

$$d_{nm} = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2}$$

همان‌طور که در بالا توضیح داده شد، با توجه به خطای فاصله‌یابی، فاصله واقعی d_{nm} معمولاً با فاصله اندازه‌گیری‌شده \hat{d}_{nm} متفاوت است. ما اثر خطای فاصله‌یابی را به‌عنوان یک نویز گوسی در نظر گرفته، آن را با P_n نشان می‌دهیم. پس داریم: $\hat{d}_{nm} = d_{nm} + P_n$.

۴. تابع هزینه یا هدف که عبارت است از میانگین مربع خطای فاصله‌یابی بین گره لنگر و گره حسگر با مختصات نامعلوم، به صورت زیر تعریف می‌شود و با $f(x_n, y_n)$ نمایش داده می‌شود:

$$f(x_n, y_n) = \frac{1}{M} \sum_{m=1}^M (\sqrt{(x_n - x_m)^2 + (y_n - y_m)^2} - \hat{d}_{nm})^2$$

به‌طوری که $M \geq 3$ نشان‌دهنده تعداد گره‌های لنگر است

همان‌طور که در شکل‌های فوق و نیز در نتایج جداول دیده می‌شود، در بعضی موارد، نتایج حاصل از الگوریتم بهبودیافته با مشتقات کسری نسبت به الگوریتم استاندارد ضعیف‌تر بوده، وابستگی به مقدار α یعنی مرتبه مشتق کسری، کاملاً مشهود است. برای مقادیر α برابر با $0/2$ و $0/8$ نتایج به‌دست‌آمده از الگوریتم بهینه‌سازی ازدحام ذرات استاندارد هم بدتر است و باید α برابر $0/6$ انتخاب شود تا نتایج بهتری حاصل گردد. ارائه الگوریتم بهینه‌سازی مرتبه کسری مبتنی بر الگوریتم ازدحام ذرات که به‌ازای تمام مقادیر مشتق مرتبه کسری، عملکردی بهتر نسبت به الگوریتم ازدحام ذرات استاندارد نشان دهد، جزء پژوهش‌های آتی مورد اشاره در مرجع [۳۶] است. با توجه به نتایج به‌دست‌آمده، الگوریتم پیشنهادی ازدحام ذرات مرتبه کسری آشوبی به‌ازای هر مقدار مشتق مرتبه کسری عملکرد بهتری نسبت به دو الگوریتم دیگر نشان می‌دهد که یکی از مهم‌ترین دستاوردهای این الگوریتم پیشنهادی است.

۴- فرایند مکان‌یابی گره حسگر در اینترنت اشیا با استفاده از الگوریتم‌های ازدحام ذرات

در این بخش، ابتدا فرایند مکان‌یابی گره حسگر در اینترنت اشیا فرمول‌بندی می‌شود و سپس نتایج شبیه‌سازی با سه الگوریتم ازدحام ذرات مقایسه می‌گردد.

۴-۱- فرایند مکان‌یابی گره حسگر در اینترنت اشیا

در این مقاله، منظور از تعیین موقعیت محل حسگرها در اینترنت اشیا، محاسبه دقیق مختصات نامعلوم N گره حسگر بر اساس مختصات معلوم M گره لنگر در مختصات دوبعدی است.

به‌طور کلی فرایند مکان‌یابی گره‌های حسگر شامل مراحل زیر می‌شود:

۱. M گره لنگر و N گره حسگر با مختصات نامعلوم به صورت تصادفی در میدان دوبعدی حسگری پخش شده‌اند که برد مخابراتی هر گره حسگر، R فرض می‌شود.

۲. گره‌های لنگر، مختصات مکانی خود را می‌دانند و این مختصات را به‌طور مداوم و به صورت همه‌پخشی ارسال می‌کنند.

۳. یک گره حسگر با مختصات نامعلوم در صورتی که در برد فرستندگی حداقل سه گره لنگر باشد، قابل موقعیت‌یابی

^۱. Fading

مکانی گره حسگر که از اجرای الگوریتم پیشنهادی به دست می‌آید باشد، داریم:

$$E_{ALE} = \frac{\sum_{n=1}^N \sqrt{(X_n - x_n)^2 + (Y_n - y_n)^2}}{N}$$

باید خاطرنشان کرد که هدف اصلی الگوریتم مکان‌یابی پیشنهادی، کاهش مقدار میانگین خطای موقعیت‌یابی، یعنی E_{ALE} است. شکل (۱۰) فلوجارت الگوریتم مکان‌یابی را نشان می‌دهد.

۲-۴- نتایج شبیه‌سازی مکان‌یابی با سه الگوریتم ازدحام ذرات

برای سنجش کارایی الگوریتم پیشنهادی، نتایج حاصل از اجرای الگوریتم پیشنهادی با دو الگوریتم ازدحام ذرات استاندارد و ازدحام ذرات مرتبه کسری مقایسه شده است. تمام شبیه‌سازی‌ها در سیستم عامل ویندوز ۱۰ با استفاده از پردازنده‌های چهارهسته‌ای و فرکانس ۳/۲ گیگاهرتز و ظرفیت حافظه چهار گیگابایت انجام شده و برای شبیه‌سازی، از نرم‌افزار متلب ۲۰۱۶ استفاده گردیده است. همچنین پارامترهای شبیه‌سازی، مطابق با جدول ۷ است.

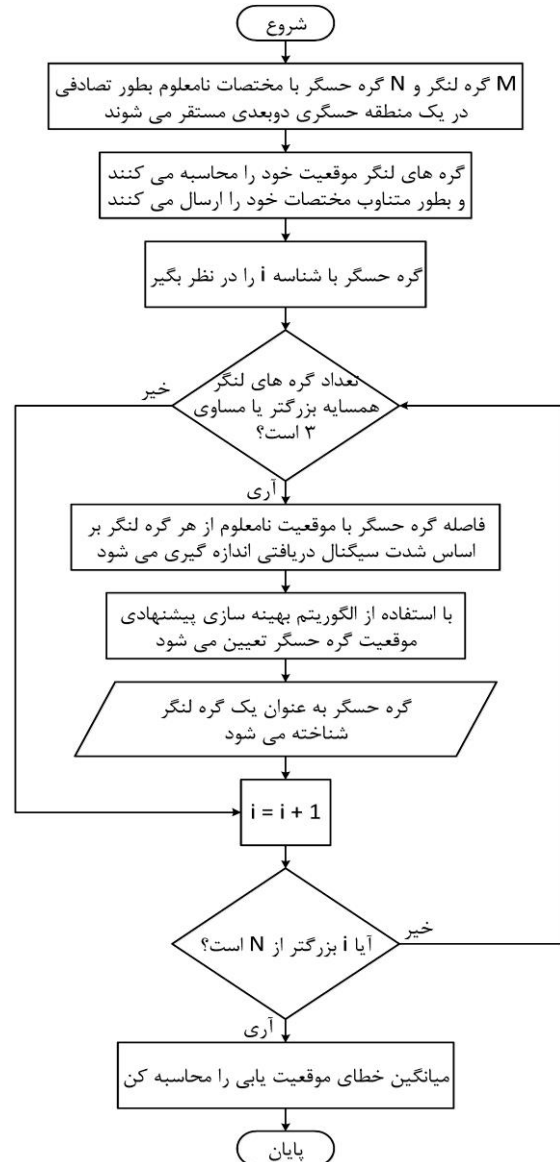
جدول ۷: پارامترهای شبیه‌سازی مکان‌یابی گره حسگر در

اینترنت اشیا

مقدار	پارامترها
۱۰۰ × ۱۰۰	اندازه میدان حسگر (مترمربع)
۱۰	تعداد گره‌های لنگر (M)
۵۰	تعداد گره‌ها با مختصات نامعلوم (N)
۲۵ متر	برد ارسال (R)
۲	درصد نویز (Pn)
۱۵۰	حداکثر تکرار (tmax)
۰/۷	ضریب لختی (w)

در این مقایسه، محل استقرار گره‌های لنگر برای هر سه الگوریتم، یکسان در نظر گرفته شده است. تخمین محل گره‌های حسگر با استفاده از الگوریتم پیشنهادی و مقایسه موقعیت تخمین زده شده با موقعیت واقعی این گره‌های حسگر در شکل (۱۱) نشان داده شده است. همچنین در شکل (۱۲) میزان خطا در تعیین محل گره‌های حسگر، با استفاده از الگوریتم پیشنهادی نشان داده شده است. برای مقایسه خطای موقعیت‌یابی گره‌های حسگر با استفاده از الگوریتم پیشنهادی و دیگر الگوریتم‌ها، در شکل (۱۳)

که گره حسگر با مختصات نامعلوم در برد رادیویی آن‌ها قرار دارد. با اجرای الگوریتم بهینه‌سازی پیشنهادی تابع هزینه $f(x_n, y_n)$ کاهش یافته، مختصات نامعلوم گره حسگر n ام به دست می‌آید.



شکل ۱۰: فلوجارت الگوریتم موقعیت‌یابی گره‌های حسگر در اینترنت اشیا

۵. در پایان هر بار تکرار الگوریتم پیشنهادی، یک گره حسگر که مختصات مکانی آن مشخص شده است، می‌تواند به‌عنوان یک گره لنگر جدید، در تکرار بعدی مورد استفاده قرار گیرد.

۶. مراحل ۲ الی ۴ تا یافتن مختصات مکانی همه گره‌های حسگر تکرار می‌شود.

۷. محاسبه میانگین خطای مکان‌یابی E_{ALE} : اگر (X_n, Y_n) مختصات مکانی واقعی حسگر باشد و (x_n, y_n) مختصات

در جدول ۸ زمان محاسبه و متوسط خطای موقعیت‌یابی الگوریتم‌ها با هم مقایسه می‌شود.

جدول ۸: نتایج شبیه‌سازی مکان‌یابی گره حسگر در اینترنت اشیا با استفاده سه الگوریتم بهینه‌سازی

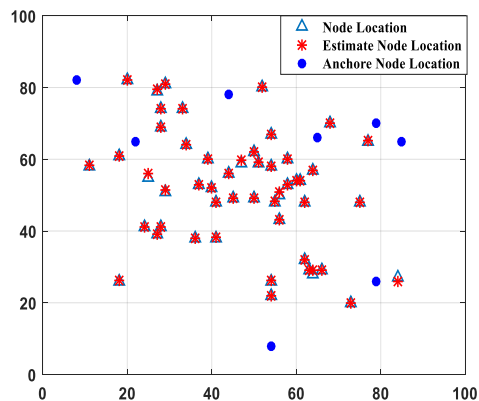
نام الگوریتم	متوسط خطای موقعیت‌یابی	زمان محاسبه
ازدحام ذرات	۰/۰۹۳۶	۱۴۳/۳۵۲۸۷
ازدحام ذرات مرتبه کسری	۰/۲۹۱۱	۷۸/۶۳۲۵۴
ازدحام ذرات مرتبه کسری آشوبی	۰/۰۰۲۳	۴۸/۶۰۵۸۶

مشاهده می‌شود که میزان خطا در الگوریتم بهینه‌سازی مرتبه کسری نسبت به دو الگوریتم دیگر بیشتر است که به دلیل وابستگی زیاد این الگوریتم به مشتق مرتبه کسری (α) است. همچنین از نظر زمان اجرای الگوریتم، بیشترین زمان اجرا مربوط به الگوریتم ازدحام ذرات استاندارد است. می‌توان به راحتی مشاهده کرد که میزان متوسط خطای موقعیت‌یابی با استفاده از الگوریتم پیشنهادی به صورت قابل ملاحظه‌ای نسبت به دو الگوریتم دیگر کمتر است. همچنین عملکرد این الگوریتم نسبت به دو الگوریتم دیگر به‌ازای تمام مقادیر مشتق مرتبه کسری بهتر بوده، به طوری که خطای موقعیت‌یابی، کاهش چشمگیری یافته است. همچنین سرعت هم‌گرایی افزایش یافته است که در نتیجه، زمان صرف‌شده برای اجرای الگوریتم کاهش می‌یابد.

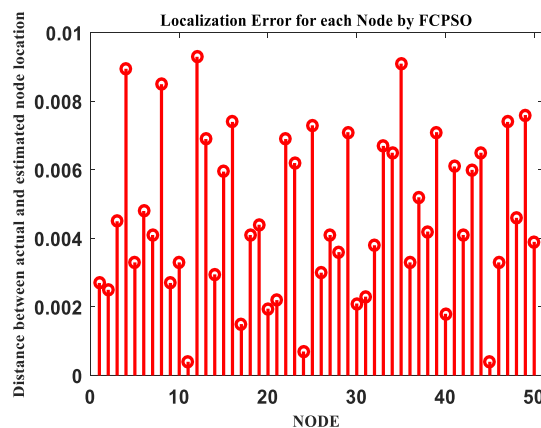
۵- نتیجه‌گیری

کاربردهای جدید و نوظهور اینترنت اشیا به تعیین دقیق موقعیت گره‌های حسگر در حداقل زمان ممکن وابسته است. در این مقاله با استفاده از الگوریتم بهینه‌سازی پیشنهادی، خطای موقعیت‌یابی گره‌های حسگر به‌طور قابل توجهی کاهش یافته است. همچنین در الگوریتم پیشنهادی که نتیجه به‌کارگیری هم‌زمان مشتقات کسری و نظریه آشوب است، نرخ هم‌گرایی افزایش یافته، از قرار گرفتن در نقاط بهینه محلی جلوگیری می‌شود. ویژگی بارز این روش آن است که علاوه بر حفظ پایداری، به‌ازای مقادیر مختلف مشتق مرتبه کسری نسبت به الگوریتم‌های مشابه، نرخ هم‌گرایی بهتری دارد. نتایج شبیه‌سازی، کارایی الگوریتم پیشنهادی را نشان داده است.

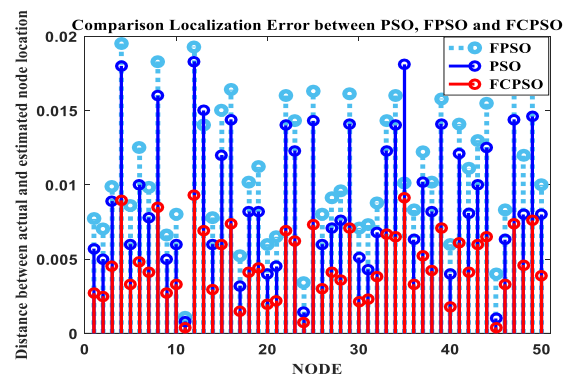
خطای موقعیت‌یابی گره‌های حسگر با استفاده از دو الگوریتم دیگر آورده شده است. همان طور که در شکل‌ها به خوبی می‌توان مشاهده کرد، مقدار خطای موقعیت‌یابی با الگوریتم پیشنهادی از دیگر الگوریتم‌ها کمتر است.



شکل ۱۱: تخمین محل گره‌های حسگر در اینترنت اشیا با استفاده از الگوریتم پیشنهادی و مقایسه با موقعیت واقعی این گره‌های حسگر



شکل ۱۲: تخمین محل گره‌های حسگر با استفاده از الگوریتم پیشنهادی و مقایسه با موقعیت واقعی این گره‌های حسگر



شکل ۱۳: مقایسه میزان خطای موقعیت‌یابی محل گره‌های حسگر با استفاده از سه الگوریتم ازدحام ذرات استاندارد، ازدحام ذرات مرتبه کسری و ازدحام ذرات مرتبه کسری آشوبی

مراجع

- [1] C-W. Tsai, C-F. Lai and A.V. Vasilakos, "Future internet of things:open issues and challenges", *Wireless Networks*, Vol. 20, No. 8, 2014, pp. 2201–2217.
- [2] S. Hasan and E. Curry, "Thingsonomy: Tackling variety in internet ofthings events", *IEEE Internet Computing*, Vol. 19, No. 2, 2015, pp. 10–18.
- [3] X. Li, R. Lu, X. Liang, X. Shen, J. Chen and X. Lin, "Smart community:an internet of things application", *IEEE Communications Magazine*, Vol. 49, No. 11, 2011, pp. 68–75.
- [4] Z. Chen, F. Xia, T. Huang, F. Bu and H. Wang, "A localization methodfor the internet of things", *The Journal of Supercomputing*, Vol. 63, No. 3, 2013, pp. 657–674.
- [5] S. Cirani, L. Davoli, G. Ferrari, R. Leone, P. Medagliani, M. Picone and L. Veltri, "A scalable and self-configuring architecture for servicediscovery in the internet of things", *IEEE Internet of Things Journal*, Vol. 1, No. 5, 2014, pp. 508–521.
- [6] S. Raza, L. Wallgren and T. Voigt, "Svelte: Real-time intrusion detectionin the internet of things", *Ad hoc networks*, Vol. 11, No. 8, 2013, pp. 2661–2674.
- [7] L. Zhou and H.-C. Chao, "Multimedia traffic security architecture for the internet of things", *IEEE Network*, Vol. 25, No. 3, 2011, pp. 35–40.
- [8] J. Hightower and G. Borriello, "A survey and taxonomy of location systems for ubiquitous computing", *IEEE computer*, Vol. 34, No. 8, 2001, pp. 57–66.
- [9] N. Bulusu, J. Heidemann and D. Estrin, "Gps-less low-cost outdoor localization for very small devices", *IEEE personal communications*, Vol. 7, No. 5, 2000, pp. 28–34.
- [10] S. Pandey and S. Varma, "A range based localization system in multihop wireless sensor networks: A distributed cooperative approach", *Wireless Personal Communications*, Vol. 86, No. 2, pp. 2016, 615–634.
- [11] M. Aziz, M.-H. Tayarani-N and M.R. Meybodi, "A two-objective memetic approach for the node localization problem in wireless sensor networks", *Genetic Programming and Evolvable Machines*, Vol. 17, No. 4, 2016, pp. 321–358.
- [12] M.X. Cheng and W.B. Wu, "A model-free localization method for sensor networks with sparse anchors", in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, pp. 1–7.
- [13] A. Pal, "Localization algorithms in wireless sensor networks: Current approaches and future challenges", *Network protocols and algorithms*, Vol. 2, No. 1, 2010, pp. 45–73.
- [14] G. Mao, B. Fidan and B. D. Anderson, "Wireless sensor network localization techniques", *Computer networks*, Vol. 51, No. 10, 2007, pp. 2529–2553.
- [15] S. Sivakumar and R. Venkatesan, "Meta-heuristic approaches for minimizing error in localization of wireless sensor networks", *Applied SoftComputing*, Vol. 36, 2015, pp. 506–518.
- [16] D. Miorandi, S. Sicari, F. De Pellegrini and I. Chlamtac, "Internet ofthings: Vision, applications and research challenges", *Ad Hoc Networks*, Vol. 10, No. 7, 2012, pp. 1497–1516.
- [17] J. Cota-Ruiz, P. Rivas-Perea, E. Sifuentes and R. Gonzalez-Landaeta, "A recursive shortest path routing algorithm with application for wireless sensor network localization", *IEEE Sensors Journal*, Vol. 16, No. 11, 2016, pp. 4631–4637.
- [18] J. Cheng and L. Xia, "An effective cuckoo search algorithm for node localization in wireless sensor network", *Sensors*, Vol. 16, No. 9, 2016, p. 1390.
- [19] V. Nagireddy, P Parwekar and Tk. Mishra, "Comparative Analysis of PSO-SGO Algorithms for Localization in Wireless Sensor Networks", in *Information Systems Design and Intelligent Applications*, 2019, pp. 401-409.
- [20] K. Hu, X. Song, Z. Sun, H. Luo and Z. Guo, "Localization Based on MAP and PSO for Drifting-Restricted Underwater Acoustic Sensor Networks", *Sensors*, Vol. 19, No. 1, 2019, pp. 71-79.
- [21] I.F.M. Zain and S.Y. Shin, "Distributed localization for wireless sensor networks using binary particle swarm optimization (bpso)", in 2014.

- [22] Y. Xuerong, C. Hao, L. Huimin, C. Xinjun and Y. Jiabin, "Multi-Objective Optimization Design for Electromagnetic Devices With Permanent Magnet Based on Approximation Model and Distributed Cooperative Particle Swarm Optimization Algorithm", *IEEE Transactions on Magnetics* Ma, Vol. 54, No.3, 2018, pp.1-5.
- [23] X. Wang, G. Wang and Y. Wu, "An Adaptive Particle Swarm Optimization for Underwater Target Tracking in Forward Looking Sonar Image Sequences", *IEEE Access*, Vol. 6, No. 1, 2018, pp.33-43.
- [۲۴] حسین شریف‌زاده و نیما امجدی، «توزیع بهینه توان راکتیو با استفاده از الگوریتم بهینه‌سازی دسته ذرات»، *مجله مدل‌سازی در مهندسی*، دوره ۴، شماره ۱۸، ۱۳۸۸، صفحه ۶۷-۷۳.
- [۲۵] روح‌الله مقصودی، یعقوب حیدری و بهزاد مشیری، «یک تحلیل مقایسه‌ای از الگوریتم‌های هوش جمعی کلونی زنبور مصنوعی و بهینه‌سازی گروهی ذرات در طراحی یک کنترل‌کننده PID فازی کسری و پیاده‌سازی آن بر روی موتور DC»، *مجله مدل‌سازی در مهندسی*، دوره ۱۱، شماره ۳۵، ۱۳۹۲، صفحه ۱۱-۲۳.
- [۲۶] ابراهیم اسدی گنگرج، فاطمه بزرگ‌نژاد و محمدمهدی پایدار، «توسعه روش‌های فراابتکاری برای حل مسئله زمان‌بندی نیروی انسانی در محیط جریان کارگاهی»، *مجله مدل‌سازی در مهندسی*، دوره ۱۶، شماره ۵۴، ۱۳۹۷، صفحه ۲۸۳-۲۹۳.
- [27] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", In *Micro Machine and Human Science, Proceedings of the 6th International Symposium on*. IEEE, 1995, pp. 39-43.
- [28] S. Poursiah Navi, E. Toreini, M. Mehrnejad and S. Kazem, "Analysis of The Usage of Chaotic Theory in Data Clustering Using Particle Swarm Optimization", *Indian Journal of Science*, Vol 4, No. 3, 2014, pp. 335-353.
- [29] W. Liu, N. Luo, G. Pan and A. Ouyang, "Chaos particle swarm optimization algorithm for optimization problems", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 32, No. 11, 2018.
- [30] B. Liu, L. Wang, Y.-H. J, F. Tang and D.-X. Huang, "Improved particle swarm optimization combined with chaos", *Chaos, Solitons & Fractals*, Vol. 25, No. 5, 2005, pp. 1261-1271.
- [31] X. Tang, L. Zhuang and Ch. Jiang, "Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization", *Expert Systems with Applications*, Vol. 36, No. 9, 2009, pp. 11853-11857.
- [32] T. Zhu T, B. Li and H. Zheng, "Optimization for Power System of Electric Vehicle Based on CPSO", *SAE Technical Paper*, 2019.
- [33] D. Tian and T. Zhao, "Particle swarm optimization based on tent map and logistic map", *Journal of Shaanxi University of Science and Technology*, Vol. 28, 2010, pp. 17-23.
- [34] D. Tian, "Particle swarm optimization with chaos-based initialization for numerical optimization." *Intelligent Automation & Soft Computing*, Vol. 12, No.6, 2017, pp. 1-12.
- [35] J. Sabatier, O.P. Agrawal and J.A. Tenreiro Machado, *Advances in fractional calculus*, Springer, 2007.
- [36] M. Couceiro and S. Sivasundaram, "Novel fractional order particle swarm optimization", *Applied Mathematics and Computation*, Vol. 283, 2016, pp. 36-54.
- [37] M. Gang, Z. Wei, C.A. Xiaolin, "novel particle swarm optimization algorithm based on particle migration", *Applied Mathematics and Computation*, Vol. 5, No. 218, 2016, pp. 6620-6626.