مجله کنترل
ISSN (print) 2008-8345
ISSN (online) 2538-3752
جلد۱۴، شماره ۴، زمستان ۱۳۹۹، صفحه ۷۹-۶۷

ردیابی شی متحرک در ویدیو با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات به صورت فازی

مهرداد روحانی[۱]، حسن فرسی[۲]، سید حمید ظهیری ممقانی[۳]

[۱] دانشجوی دکتری مهندسی برق، گروه مخابرات، دانشگاه بیرجند، M.ruhani@birjand.ac.ir
[۲] استاد، دانشکدهٔ برق و کامپیوتر، گروه مخابرات، دانشگاه بیرجند، Hfarsi@birjand.ac.ir
[۳] استاد، دانشکدهٔ برق و کامپیوتر، گروه الکترونیک، دانشگاه بیرجند، Hzahiri@birjand.ac.ir

دریافت: ۱۳۹۸/۰۳/۰۵ ویرایش : ۱۳۹۸/۰۷/۳۰ پذیرش: ۱۳۹۸/۱۱/۱۶

چکیده: امروزه یکی از اساسی‌ترین فرآیندها به منظور ادراک محتوای ویدیو، ردیابی اشیا متحرک است که در آن فرآیند مکان‌یابی شی متحرک، در هر فریم از ویدئو صورت می‌گیرد. در یک فرآیند ردیابی لازم است، هدف توسط یک ویژگی توصیف شود. در این مقاله به‌منظور توصیف شی متحرک و رفع حساسیت به ظاهر هدف، از هیستوگرام وزن‌دار رنگ به‌عنوان ویژگی هدف متحرک استفاده شده است تا اثر پیکسل‌های لبه‌ی شی هدف در ویژگی آن کاهش داده شود. این کار حساسیت الگوریتم را به تغییر حالت، تغییر مقیاس و چرخش کاهش داده و اثر انسداد جزیی بر توصیف ویژگی هدف متحرک نیز کاهش می‌یابد. در روش پیشنهادی، از الگوریتم بهینه‌سازی ازدحام ذرات به صورت فازی شده در فرآیند جستجوی هدف استفاده می‌شود. نحوه تعیین موقعیت هدف براساس بیشینه شدن تابع شباهت و محاسبه کمترین فاصله باتاچاریا است. پارامترهای کنترلی فازی شده برای الگوریتم بهینه‌سازی ازدحام ذرات استفاده شده است که یک روش نوین را ارائه می‌کند، این کار می‌تواند هر پارامتر کنترلی را به موقع و طبق حالات مختلف هر ذره در هر نسل تنظیم و به روزرسانی کند. الگوریتم ازدحام ذرات اصلاح شده توسط ۱۱ تابع محک مورد ارزیابی قرار گرفته است. آزمایش‌ها نشان می‌دهند که الگوریتم اصلاح شده در تعداد تکرار پایین همگرایی بسیار مطلوبی را نتیجه می‌دهد. روش پیشنهادی در مقایسه با روش‌های پیشرفته و جدید به عملکرد خوبی در معیارهای دقت و نرخ موفقیت روی پایگاه داده OTB50 دست یافته است.

کلمات کلیدی: ردیابی شی متحرک، الگوریتم بهینه‌سازی ازدحام ذرات اصلاح شده، ویژگی هیستوگرام رنگی وزن‌دار.

Moving object tracking in video by using fuzzy particle swarm optimization algorithm

Mehrdad Rohani, Hasan Farsi, Seyedhamid Zahirimamaghani

Abstract: Nowadays, one of the most fundamental processes for realization video of contents is the object tracking, in which the process of location the moving object is performed in each video frame. In tracking process, the target must be described by a feature. In this paper, for the purpose of describing the target and removing the appearance sensitivity, the weighted color histogram is used as a target feature in order to reduce the effect of edge pixels on the target feature. This reduces the sensitivity of the algorithm to change deformation, scale variation and rotation, as well as the occlusion on the description of target feature. In the proposed method, particle swarm optimization algorithm has been used for search process. Maximization of the similarity function and calculating the minimum Bhattacharyya distance are used to determine target location. The fuzzy control parameters are used for the particle swarm optimization algorithm, which provides a novel method, which can regulate each control parameter and update according to the different states of each particle

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]

in each generation. The improved particle swarm algorithm is evaluated with 11 benchmark functions. The obtained results by improved algorithm show that appropriate convergence in a low number of iterations. The proposed method compared to state-of-the-art methods provides high performance in the success and precision rate on the OTB50 dataset.

Keywords: Object tracking, Improved particle swarm optimization algorithm, Weighted color histogram feature.

۱- مقدمه

در سال‌های اخیر فناوری ردیابی اشیا موضوع مهمی در بینایی ماشینی بوده است. محققین زیادی الگوریتم‌های ردیابی اشیا را بررسی کرده‌اند اما به دلیل وجود مواردی مانند پیچیدگی ظاهر شی، پیچیدگی پس‌زمینه و انسداد شی، ردیابی اشیا یک موضوع چالش برانگیز بوده است. اگرچه روش‌های پیچیده‌ی زیادی برای این امر وجود دارند اما هر یک از آن‌ها شرایط خاص و محدودیت‌های خود را دارد. ردیابی اشیا در موارد متفاوتی مانند رباتیک و نظارت بر سیستم نقلیه به صورت هوشمند [۱]، ردیابی اهداف متحرک هوایی و کاربردهای نظامی [۲] استفاده می‌شود. مسأله ردیابی همچنان با چالش‌هایی مانند تغییرات شکل هدف، تغییرات مقیاس، انسداد و چرخش روبرو است و نیازمند آن است تا با پژوهش در این حوزه روش‌های مقاوم با این چالش‌ها ارائه گردد [۳-۵].

تاکنون روش‌های متنوعی در حوزه ردیابی در ویدیو مطرح گردیده است. دسته‌ای از این روش‌های از فیلترهای مختلف برای فرآیند جستجو در یک فریم از ویدیو بهره می‌گیرند. متداول‌ترین این الگوریتم‌ها عبارتند از روش‌های ردیابی مبتنی بر فیلتر همبستگی[۱] [۸-۶]. الگوریتم‌های ردیابی مبتنی بر فیلتر همبستگی عبارتند از فیلتر همبستگی با کرنل[۲]، ردیابی توسط یادگیری[۳] [۹]، ردیاب فضای مقیاسی متمایز[4] [۱۰] و محیط زمانی_مکانی[۵] [۱۱]. به طور کلی، قائده‌ی جستجوی هدف متحرک در استفاده از فیلتر همبستگی عبارت است از این‌که برای هدف یک همبستگی در پنجره‌های مختلف تصویر محاسبه می‌شود. در همین راستا بولم[۶] روش جمع کمینه‌ی خروجی مربع خطا[۷] را براساس فیلتر تطبیقی همبستگی ارائه داد [۱۲]. این الگوریتم از اولین الگوریتم‌هایی است که فیلترهای همبستگی را معرفی کرد. پس از آن هنریکس[۸] یک فیلتر همبستگی ارائه کرد که دارای مسأله‌ی رگرسیون با ماتریس دایروی بود تا روش‌ها با کرنل‌های متفاوت را اعمال کند [۶]. سپس استفاده از کرنل‌های مختلف را توسط فیلتر هم‌بستگی بهبود بخشید [۷]. مارتین[۹] ویژگی‌های سیاه سفید در الگوریتم جمع کمینه‌ی خروجی مربع خطا را با هیستوگرام گرادیان جهت دار[۱۰] تعویض کرد و با استفاده از این ویژگی و فیلتر همبستگی، یک الگوریتم جدید ارائه کرد اما به دلیل پیچیدگی بالای این الگوریتم، بازده محاسباتی و سرعت آن پایین بود [۱۳]. به طور کلی استفاده از روش‌های مبتنی بر فیلتر در برخی از چالش‌های موجود در فرآیند ردیابی مانند تغییر شکل و چرخش‌های ناگهانی با مشکل مواجه می‌شوند به طوری که هدف را از دست داده و در برخی موارد قادر به بازیابی آن نیستند.

از آن‌جایی که در سال‌ها اخیر استفاده از روش‌های فراابتکاری در ابعاد مختلف علوم مهندسی رونق گرفته است، در حوزه‌ی ردیابی نیز به عنوان دسته‌ی دیگری از روش‌ها، از الگوریتم‌های فراابتکاری استفاده شده است. در این حوزه از الگوریتم‌های فراابتکاری به منظور فرآیند جستجو بهره می‌گیرند. الگوریتم‌های متنوعی در این حوزه وجود دارد. در سال ۲۰۱۳ مینگ لیانگ ژاو[۱۱] از الگوریتم کرم شب‌تاب[۱۲] برای جستجو درون یک فریم بهره گرفت [۱۴]. در این روش پارامترهای استفاده شده ثابت بود و تنها برای برخی از دنباله‌ها بهینه بود. این روش با تغییر زمینه شی متحرک الگوریتم با شکست مواجه می‌شد. در همان سال حسین و همکاران با استفاده از ویژگی هیستوگرام گرادیان جهتدار و الگوریتم بهینه‌سازی ازدحام ذرات روش جدیدی را برای ردیابی شی متحرک در ویدیو ارائه دادند [۱۵]. این روش نیز در حالتی که شی متحرک با چرخش همراه بود هدف خود را از دست می‌داد. پس از آن فنگ شا[۱۳] الگوریتم ازدحام ذرات جدیدی به نام CPSO[۱۴] معرفی کرد که در آن پارامتر وزن اینرسی طبق یک تابع و براساس برازندگی ذرات در هر تکرار بروزرسانی می‌شد [۱۶]. در این روش تنها سرعت عبور از پاسخ بهینه مورد بحث قرار گرفته بود و با تغییرات ناگهانی زمینه و چرخش در یک فریم، هدف قابل ردیابی نبود. چانگسوک[۱۵] نمونه‌ی دیگری از الگوریتم بهینه‌سازی ذرات به نام ADSO[۱۶] ارائه کرد که در آن موقعیت هر عامل جستجو درون یک فریم از ویدیو براساس روابط جدیدی بیان می‌شد [۱۷]. در این روابط درصدی از مکان جدید شی متحرک وابسته به مکان قبلی شی متحرک بود. با این حال تمامی پارامترهای الگوریتم بهینه‌سازی ازدحام ذرات در این روش ثابت بود.

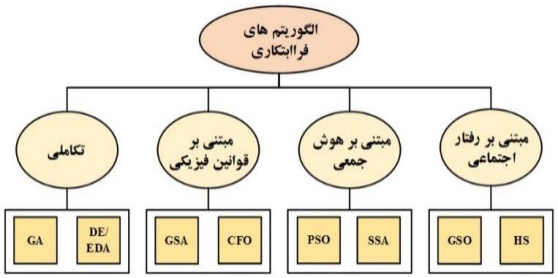
[1] Correlation Filtering
[2] Kernelized Correlation Filter
[3] Tracking-Learning-Detection
[4] Discriminative Scale Space Tracker
[5] Spatio-Temperal Context
[6] Bolme
[7] Minimum Output Sum of Squared Error
[8] Henriques
[9] Martin
[10] Histogram of Gradient
[11] Ming-Liang Gao
[12] Firefly
[13] Feng Sha
[14] Categorized Particle Swarm Optimization
[15] Changseok
[16] Adaptive Discrete Swarm Optimization

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]

به طور کلی تاکنون روش‌های متعددی برای ردیابی یک شی متحرک در ویدیو مطرح شده است که هر یک از آن‌ها دارای توانمندی‌ها و ضعف‌های متفاوتی هستند. الگوریتم بهینه‌سازی ازدحام ذرات یکی از سریعترین و ساده‌ترین الگوریتم‌هایی است که در حوزه الگوریتم‌های تکاملی معرفی شده است. از طرفی این الگوریتم دارای پارامترهای کنترلی زیادی است که کاربر را قادر می‌سازد با توجه به نیاز و کاربرد مورد نظر این پارامترها را تغییر دهد. از آن‌جایی که ماهیت ردیابی شی متحرک نیاز با یافتن پاسخ بهینه سراسری در هر فریم دارد بنابراین، با توجه به این که تغییرات محیط و پس‌زمینه و دیگر چالش‌ها در یک فریم از ویدیو وجود دارد و همچنین با توجه به ثابت بودن پارامترهای کنترلی در الگوریتم بهینه‌سازی ازدحام ذرات، ممکن است این الگوریتم در پاسخ‌های محلی گرفتار شده و فرآیند ردیابی را با شکست مواجه کند. با توجه به روش‌های ارائه شده تاکنون و ثابت بودن پارامترهای کنترلی و اهمیت مقاومت یک روش در حوزه ردیابی و پویایی آن با توجه به شرایط و چالش‌های موجود در فرآیند ردیابی، ما را برآن داشت تا در عین سادگی به طراحی مدلی از الگوریتم بهینه‌سازی ازدحام ذرات به نام FPSO[1] بپردازیم. مزیت FPSO نسبت سایر مدل‌های ارائه شده در بهینه بودن پارامترهای کنترلی در کل زمان فرآیند ردیابی است که، مهمترین عامل برای داشتن یک فرآیند ردیابی موفق و مقاوم است.

۲- الگوریتم PSO فازی شده پیشنهادی

اصطلاح متاهیوریستیک[2] اولین بار توسط گالور[3] در سال ۱۹۹۳ به هنگام معرفی روش جستجوی ممنوع[4] به کار برده شد. این روش‌ها قادرند تا از پاسخ‌های بهینه محلی[5] خارج شوند و در زمان کوتاهی به سوی پاسخ بهینه سراسری[6] حرکت کنند. عامل مهم در این روش‌ها تعادل پویا بین فرآیندهای اکتشاف[7] و استخراج[8] است. فرآیند اکتشاف قادر است تا فضای پاسخ را به خوبی جستجو کند و فرآیند استخراج، عملیات کاوش را در فضاهای محتمل‌تر انجام می‌دهد و باعث عدم اتلاف زمان در نواحی از فضای جستجو می‌شود که احتمال وجود پاسخ بهینه در آن پایین است. روش‌های فراابتکاری توسط دسته‌بندی‌های مختلفی از قبیل روش‌های مبتنی بر یک جواب و مبتنی بر جمعیت، الهام گرفته شده از طبیعت و بدون الهام از طبیعت، با حافظه و بدون حافظه و قطعی_احتمالی تقسیم‌بندی می‌شوند. برخی از روش‌ها فاقد حافظه هستند یعنی از اطلاعات به دست آمده در حین جستجو استفاده نمی‌کنند این درحالی است که برخی از روش‌های فراابتکاری از اطلاعات به دست آمده در حین جستجو بهره می‌گیرند. روش‌های مبتنی بر یک جواب در حین فرآیند جستجو یک جواب را تغییر می‌دهند در حالی که روش‌های مبتنی بر جمعیت در حین جستجو مجموعه‌ی جواب‌ها را در نظر می‌گیرند. در حالت کلی این نوع روش‌ها نسبت به روش‌های تک نقطه‌ای کندتر هستند و قادرند تا پاسخ‌های مطلوب‌تری را تولید کنند اما به دلیل پیشرفت در قدرت محاسبات کامپیوتری روش‌های مبتنی بر جمعیت دارای اهمیت بالاتری هستند که از جمله این روش‌ها می‌توان به روش‌های PSO، SSA، WOA، SCA و ... اشاره کرد [۱۸-۲۱]. تمامی این روش‌ها جزو دسته روش‌های الهام گرفته از طبیعت هستند. شکل ۱ دسته‌بندی الگوریتم‌های فراابتکاری را نمایش می‌دهد که شامل روش‌های GA[9]، DE/EDA[10]، GSA[11]، CFO[12]، PSO[13]، SSA[14]، GSO[15] و HS[16] می‌باشد.



شکل ۱: دسته‌بندی الگوریتم‌های تکاملی [۲۰].

در سال‌های اخیر الگوریتم بهینه‌سازی ازدحام ذرات[17] یک الگوریتم بهینه‌سازی بسیار پرکاربرد در حوزه بهینه‌سازی بوده است. بسیاری این الگوریتم را به صورت‌های مختلف و بسته به نوع کاربرد آن بهبود بخشیده‌اند. در سال ۱۹۹۵، دکتر کندی و ابرهارت[18]، الگوریتم بهینه‌سازی ازدحام ذرات را برای بهینه‌سازی توابع غیرخطی ارائه دادند. این الگوریتم نیاز به پارامترهایی مانند وزن اینرسی و ضریب شتاب داشت. انتخاب پارامترهای کنترلی در این الگوریتم به صورت مستقیم بر سرعت همگرایی و صحت الگوریتم اثر می‌گذاشت [۲۲]. در سال ۱۹۹۸، ابرهارت و شی[19] پارامتر جدیدی را به نام وزن اینرسی ارائه دادند. نتایج شبیه‌سازی نشان داد که این پارامتر جدید تاثیر زیادی بر بهینه‌ساز ازدحام ذرات دارد و وقتی وزن اینرسی ۰/۷۲۹۸ باشد این الگوریتم نتایج قابل قبولی را ارائه می‌داد [۲۳]. در سال ۲۰۰۶ چاترجی و سیاری[20] شیوه تنظیم کاهنده غیرخطی را بر طبق تعداد تکرارها، برای تنظیم وزن اینرسی معرفی کردند. این شیوه اجازه می‌داد تا وزن اینرسی به طور غیرخطی از مقدار بیشینه به مقدار کمینه کاهش یابد [۲۴]. در سال ۲۰۰۸، ژو[21] روش تنظیم تصادفی وزن اینرسی را برای بهبود عملکرد کلی بهینه‌سازی ارائه داد [۲۵]. در سال ۲۰۱۰ نیکنام و همکارانش الگوریتم PSO اصلاح شده‌ای را برای پخش بار اقتصادی در سیستم‌ها قدرت معرفی کرد که در آن پارامترهای کنترلی C_1 و C_2 به

[1] Fuzzy Particle Swarm Optimization
[2] Metahuristic
[3] Golver
[4] Tabu search
[5] Local
[6] Global
[7] Exploration
[8] Exploitation
[9] Genetic Algorithm
[10] Differential Evolution/Estimation of Distribution Algorithm
[11] Gravitational Search Algorithm
[12] Central Force Optimization
[13] Particle Swarm Optimization
[14] Salp Swarm Optimization
[15] Glowworm Swarm Optimization
[16] Harmony Search
[17] Particle Swarm Optimization
[18] Kennedy and Eberhart
[19] Shi and Eberhart
[20] Chatterjee and Siarr
[21] Zhu

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]

صورت فازی کنترل می‌شدند [۲۶]. تفاوت روش پیشنهادی با این روش در نوع و تعداد ورودی و خروجی‌های سیستم فازی و نحوه تعریف توابع عضویت است که روش پیشنهادی از تعداد خروجی‌های بالاتر و ورودی‌های متفاوت با این روش بهره گرفته است. در سال ۲۰۱۳ مالدونادو[1] از الگوریتم ژنتیک و الگوریتم PSO بهره گرفت تا دور موتور یک موتور DC را به صورت فازی کنترل کند [۲۷]. در سال ۲۰۱۶ توماس و همکارنش[2] مدل بهبود یافته‌ای از الگوریتم PSO ارائه دادند که در آن پارامترهای C_1 و C_2 مانند مرجع [۲۶] به صورت فازی کنترل می‌شدند با این تفاوت که این پارامترها برای هر عامل جستجو در فضای پاسخ متفاوت بود [۲۸]. در سال ۲۰۱۷ اولیواس و همکارانش[3] دو مدل فازی از الگوریتم‌های PSO و بهینه سازی زنبور عسل[4] را برای کنترل مسیر یک ربات طراحی کردند [۲۹]. در این روش پارامترهای C_1 و C_2 به صورت فازی کنترل می‌شدند با این تفاوت که، ورودی سیستم فازی پارامتر جدید بود که مشخص می‌کرد چه تعداد از عوامل جستجو نزدیک هم هستند و همچنین براساس آن پارامترهای کنترلی تعیین می‌شدند. در سال ۲۰۱۸ والدز و همکارانش[5] برای کنترل سطح آب و تعیین دمای آن توسط دو سیستم فازی را با استفاده از الگوریتم‌های PSO و الگوریتم بهینه‌سازی کلونی مورچگان[6] معرفی کردند [۳۰]. در این روش نیز پارامترهای کنترلی C_1 و C_2 در الگوریتم PSO به صورت فازی کنترل می‌شدند. در این الگوریتم یکی از ورودی‌های سیستم فازی پارامتر وزن اینرسی بود که با توجه به این که این پارامتر مقدار ثابتی در طول اجرای الگوریتم دارد بنابراین برخی از قوانین فازی بکار رفته در این روش دارای کارایی لازم نبودند.

تاکنون روش‌های متنوعی برای فازی کردن الگوریتم PSO معرفی شده است که هر کدام بسته به نوع کاربرد و هدف مورد استفاده، پارامترهای کنترلی موجود در الگوریتم را مورد تغییر و بررسی قرار داده‌اند. روش پیشنهادی معرفی شده علاوه بر پارامترهای C_1 ، C_2 و همچنین وزن اینرسی، تعداد عوامل جستجو را با توجه به روند پیشرفت الگوریتم مورد تغییر قرار داده است که یکی از پارامترهای خروجی در سیستم فازی است. روش‌های معرفی شده هر کدام از ورودی‌های متفاوتی برای سیستم فازی استفاده کرده‌اند در حالی که روش پیشنهادی به جای معرفی پارامتر جدید سعی در استفاده از پارامترهای موجود در الگوریتم PSO را داشته است تا از پیچیدگی الگوریتم کاسته شود. در ادامه به بررسی الگوریتم PSO متداول و روش پیشنهادی پرداخته شده است.

۱-۲- معرفی الگوریتم PSO متداول

الگوریتم بهینه‌سازی ازدحام ذرات از دسته الگوریتم‌های فراابتکاری است که مبتنی بر هوش جمعی بوده و الهام گرفته شده از رفتار اجتماعی پرندگان است. نتیجه‌ی مدل‌سازی این رفتار اجتماعی فرآیند جستجویی است که ذرات به سمت پاسخ بهینه میل می‌کنند. ذرات از یکدیگر می‌آموزند و برمبنای دانش بدست آمده به سمت بهترین همسایگان خود می‌روند اساس کار PSO بر این اصل استوار است که در هر لحظه هر ذره مکان خود را در فضای جستجو با توجه به بهترین مکانی که تاکنون در آن قرار گرفته است و بهترین مکانی که در کل همسایگی‌اش وجود دارد، تنظیم می‌کند. در فضای n بعدی هر ذره دارای یک موقعیت است که توسط بردار x_i به صورت زیر مشخص می‌گردد [۱۸].

$$x_i = (x_{i1}, x_{i2}, \dots, x_{in})$$

که در آن x_{in} بیانگر موقعیت ذره i ام در بعد n ام است. همچنین برای هر ذره یک بردار سرعت v_i تعریف می‌شود که به صورت زیر نمایش داده می‌شود.

$$v_i = (v_{i1}, v_{i2}, \dots, v_{in})$$

که در آن v_{in} بیانگر سرعت ذره i ام در بعد n ام است. برای هر ذره بهترین موقعیت $P_{i.best}$ در هر بعد به صورت زیر نشان داده می‌شود.

$$P_{i.best} = (p_{i1}, p_{i2}, \dots, p_{in})$$

که در آن p_{in} نشان دهنده بهترین موقعیت ذره i ام در بعد n ام است. و بهترین موقعیت ذره در بین کل ذرات به صورت زیر با $P_{g.best}$ نشان داده می‌شود.

$$P_{g.best} = (p_{g1}, p_{g2}, \dots, p_{gn})$$

سرعت و مکان هر ذره در هر تکرار با استفاده از روابط (۱) و (۲) به‌روزرسانی می‌شود [۱۸].

$$\boldsymbol{V_i(t) = w * V_i(t-1) + C_1 * rand_1 * \left(P_{i,best} - X_i(t-1)\right) + C_2 * rand_2 * \left(P_{i,g,best} - X_i(t-1)\right)} \quad (۱)$$

$$\boldsymbol{X_i(t) = X_i(t-1) + V_i(t)} \quad (۲)$$

در روابط (۱) و (۲)، $i = 1,2,\dots,Popsize$ است و $Popsize$ تعداد ذرات (عوامل جستجو) را بیان می‌کند. w ضریب وزنی اینرسی (حرکت در مسیر خودی) است که نشان دهنده میزان تأثیر بردار سرعت تکرار قبل $V_i(t-1)$ بر روی بردار سرعت در تکرار فعلی $V_i(t)$ است. $rand_1$ و $rand_2$ اعداد تصادفی با توزیع یکنواخت در بازه صفرتا یک هستند و اثر تصادفی بودن را در حرکت فردی و گروهی ذرات ایجاد می‌کنند. C_1 ضریب ثابت آموزش برای تنظیم اثر حرکت فردی است و C_2 ضریب ثابت آموزش برای تنظیم اثر حرکت گروهی است. X_i نشان دهنده بردار موقعیت هر ذره می‌باشد. ضرایب w ، C_1 و C_2 پارامترهای کنترلی الگوریتم PSO هستند که با توجه به نوع مسأله به صورت تجربی تعیین می‌گردند. به طور معمول مقدار وزن اینرسی[7] (w) در حدود مقادیر مثبت کمتر از یک در نظر گرفته می‌شود. مقادیر C_1 و C_2 نیز به طور معمول حول مقدار ۲ در نظر گرفته می‌شوند [۱۸].

[1] Maldonado
[2] Tomasz et al
[3] Olivas et al
[4] Bee Colony Optimization
[5] Valdez et al
[6] Ant Colony Optimization
[7] Inertia weight

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]

۲-۲- بررسی نقش و تأثیرات پارامترهای کنترلی الگوریتم PSO

پارامترهای کنترلی C_1 و C_2 در الگوریتم PSO نقش به سزایی در نحوه همگرا شدن دارند. C_1 پارامتر کنترلی به منظور تعیین تأثیر نقش حرکت فردی هر عامل جستجو در فضای جستجو است. نحوه عملکرد آن بستگی به مقدار آن دارد. اگر C_1 شامل مقادیر بزرگی باشد الگوریتم PSO دچار اغتشاش و آشوب می‌شود زیرا در این حالت نقش هر عامل جستجو به صورت مجزا پر رنگ شده و همگرایی با مشکل مواجه می‌شود. در این حالت ممکن است حتی با ورود به پاسخ بهینه سراسری[1] همگرایی رخ ندهد و الگوریتم از پاسخ بهینه عبور کند. مقادیر کم برای این پارامتر آشفتگی را به شدت کاهش می‌دهد و باعث می‌شود تا همگرایی به نقاط بهینه محلی[2] تسهیل شود و الگوریتم توانایی خود را برای رهایی از پاسخ‌های بهینه محلی از دست می‌دهد. در واقع C_1 نقش اکتشاف[3] را در فضای جستجو بر عهده دارد. پیشنهادات گوناگونی اعم از تغییرات خطی، انتخاب تصادفی و یا تغییرات غیرخطی برای این پارامتر ارائه شده است. استفاده از مقادیر تصادفی ممکن است در هنگام همگرایی به پاسخ‌های محلی و یا پاسخ سراسری کارآمد نباشد و همگرایی را با مشکل مواجه کند. تغییرات خطی و غیرخطی برای این پارامتر نیز ممکن است در انتهای فرآیند جستجو الگوریتم را با یک پاسخ محلی مواجه کند. گاهی ممکن است در ابتدای فرآیند جستجو و تولید جمعیت اولیه تصادفی، الگوریتم بسیار نزدیک به پاسخ سراسری باشد و انتخاب مقادیر بزرگ برای C_1 ممکن است الگوریتم را از ناحیه شامل پاسخ سراسری دور کند.

C_2 پارامتر کنترلی برای مشخص کردن نقش رهبر گروه یا همان بهترین رفتار اجتماعی است. مقادیر کوچک برای این پارامتر نقش بهترین پاسخ یافته شده در گروه را پررنگ می‌کند. به این ترتیب انتخاب مقادیر بزرگ C_2 ذرات را حول بهترین پاسخ گروه فرامی‌خواند و باعث جمع شدن ذرات در فضای جستجو حول بهترین پاسخ گروه می‌گردد. در این حالت ممکن است بهترین پاسخ گروه یک پاسخ محلی باشد و الگوریتم دچار خطا شود. انتخاب مقادیر کوچک برای این پارامتر نیز باعث بروز آشفتگی می‌شود و تأثیر رفتار فردی ذرات را در پی دارد. بنابراین C_2 نقش استخراج[4] را در فضای جستجو بر عهده دارد. برای این پارامتر نیز مانند C_1 انتخاب تصادفی، تغییرات خطی و غیرخطی پیشنهاد شده است. مشکل این روش‌ها این است که انتخاب مقادیر بزرگ در ابتدای فرآیند جستجو ممکن است باعث همگرایی محلی می‌شود. انتخاب مقادیر بزرگ در انتهای فرآیند جستجو نیز هنگامی که الگوریتم در یک پاسخ محلی به دام افتاده است، باعث همگرایی محلی می‌گردد.

سومین پارامتر کنترلی مهم در الگوریتم PSO وزن اینرسی است. زمانی که w مقدار بالایی داشته باشد، الگوریتم توانایی همگرایی سراسری بالایی دارد، درحالی‌که وزن کم باعث می‌شود تا الگوریتم به نقاط بهینه محلی همگرا شود. در بسیاری از پژوهش‌ها براساس تاثیر بالای وزن بر رفتار جستجوی الگوریتم، عملکرد بهینه‌سازی آن را با تغییر وزن بهبود بخشیده‌اند. درحال حاضر، روش‌های بهبود یافته‌ای برای تنظیم وزن ارائه شده است که می‌توان به انتخاب تصادفی وزن به جای مقدار ثابت، انتخاب وزن به صورت خطی و غیرخطی اشاره کرد. در استفاده از این روش‌ها این احتمال وجود دارد تا هنگامی که الگوریتم در ابتدا و یا انتهای فرآیند جستجو در یک نقطه بهینه محلی قرار گرفته است، در همان نقطه گرفتار شده و الگوریتم به پاسخ بهینه سراسری همگرا نشود. در فرآیند جستجوی PSO، نزدیکترین ذره به بهترین جواب، ذره بهینه نام دارد و دیگر ذرات زیربهینه هستند. باید توجه کرد که ممکن است در فرآیند نزدیک شدن ذرات به بهترین پاسخ، مقدار پاسخ ذرات زیربهینه از مقدار ذره‌ای بهینه بیشتر باشد در این حالت، ذره زیربهینه که مقدار بهتری دارد جای ذره بهینه را می‌گیرد. این وضعیت به طور مداوم اتفاق می‌افتد. وقتی ذرات از بهترین جواب دور باشند، باید فرآیند چند بار تکرار شود تا به بهترین پاسخ نزدیک شوند. به دلیل این‌که ذره بهینه به سرعت بالایی برای رسیدن به بهترین پاسخ لازم دارد، مقدار وزن باید نسبتا بالا باشد. سرعت ذرات زیربهینه با سرعت قبلی آن‌ها و فاصله آن‌ها تا ذره بهینه تعیین می‌شود بنابراین، ذرات زیربهینه هم به وزن بالا نیاز دارند. اما ممکن است در ابتدای فرآیند جستجو ذره بهینه بسیار نزدیک به پاسخ بهینه باشد. بنابراین در این حالت ممکن است ذرات زیربهینه با سرعت بالا به بهترین جواب نزدیک شوند و از ناحیه بهترین پاسخ با سرعت عبور کنند که این امر سبب می‌شود که ذرات وارد منطقه بهترین پاسخ نشوند. در این مورد، برای بهبود صحت راه حل، باید سرعت ذرات را کاهش داد به این منظور که ذرات به وزن کمتری برای نزدیک شدن به بهترین پاسخ نیاز دارند. ممکن است این دو وضعیت در طول عملکرد الگوریتم اتفاق بیافتند. روش‌های تنظیم وزن تصادفی یا ثابت مناسب این روش نیستند. روش تنظیم وزن به صورت خطی یا غیرخطی توسط تعداد تکرار ذرات از وزن متوسط استفاده می‌کند تا بدون در نظر گرفتن شرایط مختلف هر ذره در هر نسل، بهینه‌سازی به نتایج مقبولی برسد. بنابراین مقدار استاندارد قابل اطمینانی برای ارزیابی وزن‌ها وجود ندارد. اگر ذره‌ای مقدار پاسخ بالایی داشته باشد، ممکن است برای تنظیم به مقادیر مختلف وزن نیاز داشته باشد [۳۱, ۳۲].

تعداد ذرات ($Popsize$) در الگوریتم PSO با توجه به نوع مسأله تعیین می‌شود. این سوال مطرح می‌شود که اندازه جمعیت ثابت در زمان اجرا فرآیند جستجو و تعداد تکرار تا رسیدن به پاسخ بهینه موثر است؟ افزایش تعداد ذرات اولیه موجب کاهش تعداد تکرارهای لازم برای همگرا شدن الگوریتم می‌گردد زیرا فضای جستجو با افزایش جمعیت بیشتر پوشش داده می‌شود. اما کاهش در تعداد تکرارها به معنی کاهش زمان اجرای برنامه برای رسیدن به همگرایی نیست. هرچند که افزایش تعداد ذرات اولیه کاهش تعداد تکرارها را در پی دارد. اما افزایش در تعداد ذرات باعث می‌شود تا در مرحله ارزیابی ذرات زمان بیشتری صرف شود و این افزایش

[1] Global
[2] Local
[3] Exploration
[4] Exploitation

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]

در زمان ارزیابی سبب افزایش زمان اجرای الگوریتم تا رسیدن به همگرایی با وجود کاهش تعداد تکرارها، می‌گردد. پس افزایش تعداد عوامل جستجو را نمی‌توان عاملی برای کاهش زمان اجرای در نظر گرفت. همچنین برای کاهش زمان اجرای الگوریتم نمی‌توان تعداد عوامل جستجو را کاهش داد. کاهش تعداد ذرات ممکن است موجب گیر افتادن در نقاط بهینه محلی شود و الگوریتم از رسیدن به پاسخ بهینه سراسری باز خواهد ماند.

تعداد تکرار (t) پارامتر دیگری است که در زمان اجرای الگوریتم موثر است کاهش تعداد تکرار باعث کاهش زمان اجرای الگوریتم و افزایش آن سبب افزایش زمان اجرای می‌گردد. با کاهش تعداد تکرار زمان اجرای الگوریتم کاهش می‌یابد اما جواب به دست آمده، ممکن است پاسخ بهینه‌ای برای مسأله نباشد زیرا ممکن است الگوریتم به طور ناقص اجرا شود. افزایش تعداد تکرار نیز تضمینی برای رسیدن به پاسخ سراسری نیست ممکن است الگوریتم در پاسخ یک پاسخ محلی به دام افتاده و با افزایش تکرار پیشرفتی در روند فرآیند جستجو حاصل نشود.

۳-۲- روش پیشنهادی برای فازی‌سازی الگوریتم PSO با توجه به نقش پارامترهای کنترلی آن

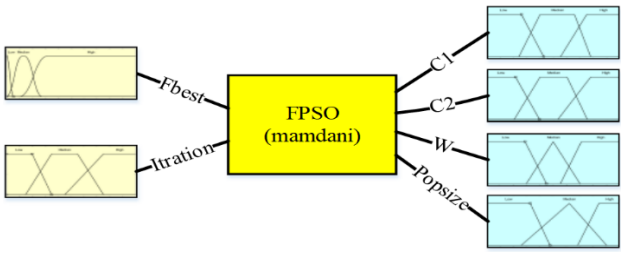
برای حل مشکل‌های بیان شده، باید یک مصامحه بین پارامترهای کنترلی در یک الگوریتم بهینه‌سازی وجود داشته باشد. به این منظور استفاده از روش‌های تصادفی، خطی و غیرخطی کارایی چندانی در روند حل مسائل بهینه‌سازی نخواهند داشت. در این مقاله روشی برای بهبود تنظیم پارامترهای کنترلی ارائه شده است تا بتوان به صورت فازی به توصیف وضعیت یک عامل جستجو پرداخته شود. روش پیشنهادی در ادامه به اختصار با FPSO نمایش داده می‌شود. وقتی میزان بهترین پاسخ در هر مرحله مقدار بالایی داشته باشد به این معنی است که ذرات در موقعیتی دور از بهترین پاسخ سراسری به صورت خوشه‌ای جمع شده‌اند و تمام جمعیت به آرامی به راه حل بهینه نزدیک می‌شود. وقتی میزان بهترین پاسخ نسبتا پایین باشد، به این معنی است که ذرات در نزدیکی راه حل بهینه نوسان دارند. این امر ما را برآن داشت تا سیستم فازی برای انتخاب مقادیر پارامترهای کنترلی طراحی نماییم. سیستم فازی[1] شده در شکل ۲ ارائه شده است که در آن ورودی بهترین پاسخ (Fbest) الگوریتم PSO و شماره تکرار جاری و خروجی آن ضرایب w ، C_1 و C_2 و $Popsize$ است که با توجه به تغییرات بهترین پاسخ و تکرار جاری برای بالا بردن دقت و سرعت همگرایی به بهترین پاسخ سراسری تغییر می‌کند. میزان وزن w در محدوده ۰/۲ تا ۰/۸ تغییر می‌کند و مقدار آن در هر مرحله به عنوان $w(t)$ بازنشانی می‌شود. مقادیر C_1 و C_2 نیز در هر مرحله براساس قوانین فازی طراحی شده، در هر مرحله به‌روز رسانی می‌گردند. اندازه $Popsize$ نیز در هر مرحله به‌روز رسانی می‌گردد. لازم به ذکر است که $Popsize$ در شروع الگوریتم مقدار داده می‌شود و میزان آن بین مقدار اولیه و صفر تغییر می‌کند. در جدول ۱ قوانین فازی استفاده شده در سیستم فازی به طور کامل ارئه شده است.

جدول ۱: قوانین فازی استفاده شده در FPSO

قوانین فازی
1. If (Fbest is High) and (Iteration is Low) then (C1 is High)(C2 is Low)(W is High)(Popsize is High)
2. If (Fbest is High) and (Iteration is Median) then (C1 is High)(C2 is Low)(W is High)(Popsize is High)
3. If (Fbest is High) and (Iteration is High) then (C1 is High)(C2 is Median)(W is Median)(Popsize is High)
4. If (Fbest is Median) and (Iteration is Low) then (C1 is High)(C2 is Low)(W is High)(Popsize is High)
5. If (Fbest is Median) and (Iteration is Median) then (C1 is High (C2 is Low)(W is Median)(Popsize is High)
6. If (Fbest is Median) and (Iteration is High) then (C1 is High)(C2 is Median)(W is Median)(Popsize is High)
7. If (Fbest is Low) and (Iteration is Low) then (C1 is Median)(C2 is High)(W is Low)(Popsize is High)
8. If (Fbest is Low) and (Iteration is Median) then (C1 is Low)(CZ is High)(W is Low)(Popsize is Median)
9. If (Fbest is Low) and (Iteration is High) then (C1 is Low)(C2 is High)(W is Low) (Popsize is Median)

۳- نحوه استفاده از روش FPSO در مسأله ردیابی هدف متحرک

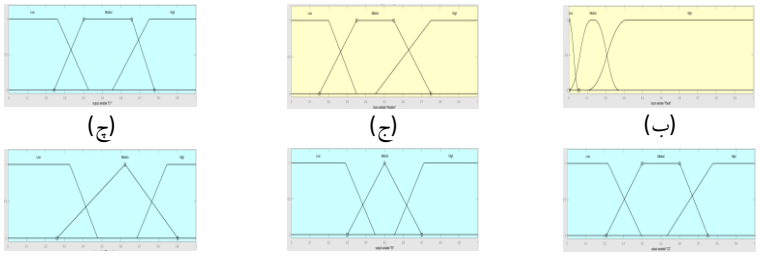
به منظور به کارگیری روش پیشنهادی در حوزه ردیابی هدف متحرک در ویدئو لازم است تا ویژگی مناسبی برای توصیف هدف متحرک در ویدئو انتخاب گردد. در این بخش ابتدا به معرفی ویژگی رنگ پرداخته شده است و سپس تابع برازندگی به کار گرفته شده برای ارزیابی هر یک از کاندیداهای تصویر که توسط الگوریتم FPSO انتخاب می‌شوند، معرفی شده است.



(الف)

[1] Fuzzy System

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]



شکل ۲: الف) سیستم فازی. ب) تابع عضویت ورودی Fbest ب) تابع عضویت ورودی Iteration. ج) تابع عضویت خروجی C1 ج) تابع عضویت خروجی C2 ج) تابع عضویت خروجی W ج) تابع عضویت خروجی Popsize.

۳-۱- ویژگی رنگ

با معرفی هیستوگرام رنگ توسط سوین و بالارد[1] در سال ۱۹۹۱، ویژگی‌های رنگی به طور گسترده در ردیابی شی استفاده شده‌اند [۳۳]. ویژگی‌های رنگی وابستگی کمتری به اندازه، جهت‌گیری و زاویه‌ی دید تصویر دارند، به همین دلیل در توصیف یک صحنه یا یک شی توانمندی بیشتری دارند. ابتدا، ویژگی‌های رنگی باید فضای رنگی مناسب را به منظور توصیف یک صحنه انتخاب کند. سپس، ویژگی‌های رنگی سنجیده می‌شوند. سرانجام، باید معیار شباهت برای سنجش شباهت رنگی بین تصاویر تعریف شود. چند روش برای توصیف ویژگی‌های رنگی از قبیل هیستوگرام‌ها، مجموعه‌های رنگ و بخش‌های رنگی وجود دارند که هیستوگرام رایج‌ترین ویژگی رنگی مورد استفاده در ردیابی می‌باشد [۳۴, ۳۵]. در این روش ناحیه‌ای از تصویر که شامل هدف[2] است با استفاده از هیستوگرام با m بین[3] در فضای رنگی RGB[4] توصیف می‌شود. تصویر مرجع توسط تابع چگالی احتمال رنگی $\{q_u\}_{u=1,2,\dots,m}$ نشان داده می‌شود که این تابع، توزیع رنگی هدف را در یک فریم[5] از ویدئو طبق رابطه (۳) تخمین می‌زند.

$$q_u = C\sum_{i=1}^{n} K\left(\left\|\frac{x-x_i}{h}\right\|^2\right)\delta[b(x_i)-u] \qquad (۳)$$

در رابطه (۳) $\{x_i\}_{i=1,2,..,n}$ مکان‌های نرمالیزه شده پیکسل در ناحیه هدف است. $b(x_i)$ مقدار اندیس پیکسل کوانتیزه شده در مکان پیکسل x_i است که مقدار پیکسل را به یک بین از هیستوگرام اختصاص می‌دهد. $u = 1,2,\dots,m$ است که m تعداد بین هیستوگرام را نشان می‌دهد. $\delta[.]$ تابع دلتای کرونکر[6] است. C ثابتی است که فرآیند نرمالیزه کردن را انجام می‌دهد تا شرط $\sum_{u=1}^{m} q_u = 1$ برآورده شود و به صورت رابطه (۴) بیان می‌شود [۳۶].

$$C = \frac{1}{\sum_{i=1}^{n} K\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} \qquad (۴)$$

برای وزن‌دهی از کرنل ایزوتروپیک $K(x)$ با پنجره‌ای به شعاع h استفاده می‌گردد. این کرنل به پیکسل‌های دورتر از مرکز ناحیه هدف که با x نمایش داده شده است، وزن‌های کمتری اختصاص می‌دهد. زیرا این نواحی گاهی دچار انسداد و تداخل با پس‌زمینه تصویر گردیده و قابلیت اطمینان کمتری دارند بنابراین استفاده از این کرنل مقاومت مدل‌سازی هدف ر ا افزایش می‌دهد. به این منظور از کرنل Epanechnikov که یک کرنل بهینه است استفاده می‌گردد. رابطه (۵) نحوه عملکرد این کرنل را نمایش می‌دهد.

$$k_E(x) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|x\|^2) & if \quad \|x\|<1 \\ 0 & otherwise \end{cases} \qquad (۵)$$

در رابطه (۵)، d نمایانگر بعد فضا و c_d برای ابعاد متفاوت مقادیر مختلف $c_1 = 2$، $c_2 = \pi$ و $c_3 = 4\pi/3$ را اخذ می‌نماید. q_u نمایش دهنده تابع چگالی احتمال مدل هدف است به طور مشابه برای مدل‌سازی کاندیداهای مختلف در یک فریم از $\{p_u(y)\}$ که نشان دهنده تابع چگالی احتمال در یک ناحیه به مرکز y است استفاده می‌شود. $p_u(y)$ به صورت رابطه (۶) بیان می‌گردد [۳۶].

$$p_u(y) = C_h\sum_{i=1}^{n} K\left(\left\|\frac{y-x_i}{h}\right\|^2\right)\delta[b(x_i)-u] \qquad (۶)$$

در رابطه (۶)، C_h ثابت نرمالیزه کننده که در آن h مقیاس هدف یا پنجره کرنل تعریف می‌گردد و به صورت رابطه (۷) نشان داده می‌شود.

$$C_h = \frac{1}{\sum_{i=1}^{n} K\left(\left\|\frac{y-x_i}{h}\right\|^2\right)} \qquad (۷)$$

به منظور مدل‌سازی هدف و مدل‌سازی کاندیداها در هر فریم از تابع چگالی احتمال با کرنل Epanechnikov استفاده می‌شود. پس از مدل‌سازی نیاز است تا مدل‌های ارائه شده با یکدیگر تطبیق یافته و مدلی که بیشترین تطبیق را در هر فریم با هدف مورد نظر دارد به عنوان هدف در فریم جاری انتخاب گردد. به این منظور لازم است از یک معیار سنجش شباهت استفاده گردد که در بخش معیار سنجش شباهت معرفی شده است.

۳-۲- تابع برازندگی

[1] Swain and Ballard
[2] Target
[3] Bin
[4] Red-Green-Blue
[5] Frame
[6] Kronecker delta function

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]

به منظور محاسبه برازندگی کاندیداهای موجود در یک تصویر و سنجش میزان شباهت از ضریب باتاچاریا[1] استفاده شده است. این ضریب میزان و درجه شباهت را برای دو هیستوگرام رنگی وزن داده شده هدف و کاندیدهای مختلف در یک فریم از ویدئو را اندازه‌گیری می‌کند. فاصله باتاچاریا به صورت رابطه (۸) بیان می‌گردد [۳۶].

$$d(y) = \sqrt{1 - \rho[p(y), q]} \quad (۸)$$

ضریب باتاچاریا در رابطه (۸) مطابق با رابطه (۹) بیان می‌گردد.

$$\rho[p(y), q] = \sum_{u=1}^{m} \sqrt{p_u(y), q_u} \quad (۹)$$

$[p(y), q]$ مقداری بین صفر تا یک دارد و زمانی که مقدار q برای هدف و $p(y)$ برای یک کاندید در یک فریم یکسان باشند مقدار $\rho[p(y), q] = 1$ است و هر چه این میزان به یک نزدیکتر باشد بیانگر شباهت بیشتر و همبستگی بالاتر است. بنابراین هر چه میزان شباهت یک کاندید به هدف مورد نظر فرآیند ردیابی بیشتر باشد مقدار ضریب باتاچاریا بیشتر و فاصله باتاچاریا کمتر است. هدف در فرآیند ردیابی کمینه‌سازی فاصله باتاچاریا می‌باشد.

٤- نتایج و پیشنهادات

در این قسمت ابتدا به معرفی توابع محک برای ارزیابی روش FPSO پرداخته شده است و سپس، نتایج حاصل از این ارزیابی در قالب شکل ۴ و جداول ۳ و ۴ ذکر شده است. در ادامه تحلیلی از این نتایج ارائه گردیده است. در قسمت دوم پایگاه داده به کار گرفته شده به منظور بررسی عملکرد روش پیشنهادی در حوزه ردیابی هدف متحرک، معرفی شده است. سپس، نتایج حاصل از این روش پیشنهادی با ۶ روش مرتبط و متداول در این حوزه مقایسه و شده و مورد تحلیل قرار گرفته است. در انتها نتیجه‌گیری و پیشنهادات روی این تحقیق ارائه گردیده است.

٤-١- توابع محک[2]

توابع محک برای مسائل بهینه‌سازی به شکل کمینه‌ساز مورد استفاده قرار می‌گیرند و به دو دسته‌ی Unimodal و Multimodal تقسیم می‌شوند. این توابع در جدول ۲ ارائه شده‌اند. Range، محدوده جستجو در فضای جستجو و Fmin نقطه بهینه را در فضای جستجو نشان می‌دهد. توابع F1 تا F6 دارای یک نقطه بهینه سراسری هستند و فرآیند استخراج را در فضای جستجو بررسی می‌کنند و عملکرد یک روش را مورد آزمایش قرار می‌دهند. توابع محک F7 تا F11 دارای چندین نقطه بهینه محلی هستند که به طور نمایی با افزایش بعد افزایش می‌یابند و به خوبی قادرندتا فرآیند اکتشاف را در فضای جستجو مورد آزمایش قرار دهند و شایستگی و توانمندی یک روش را مورد تحلیل قرار دهند. به منظور نشان دادن عملکرد روش FPSO، با یکی از جدیدترین روش‌های بهینه‌سازی فراابتکاری تحت عنوان الگویتم بهینه‌سازی سالپ (SSA) مقایسه شده است. این روش از دسته‌ی روش‌های فراابتکاری الهام گرفته شده از رفتار اجتماعی می‌باشد [۱۹].

جدول ۲: توابع محک

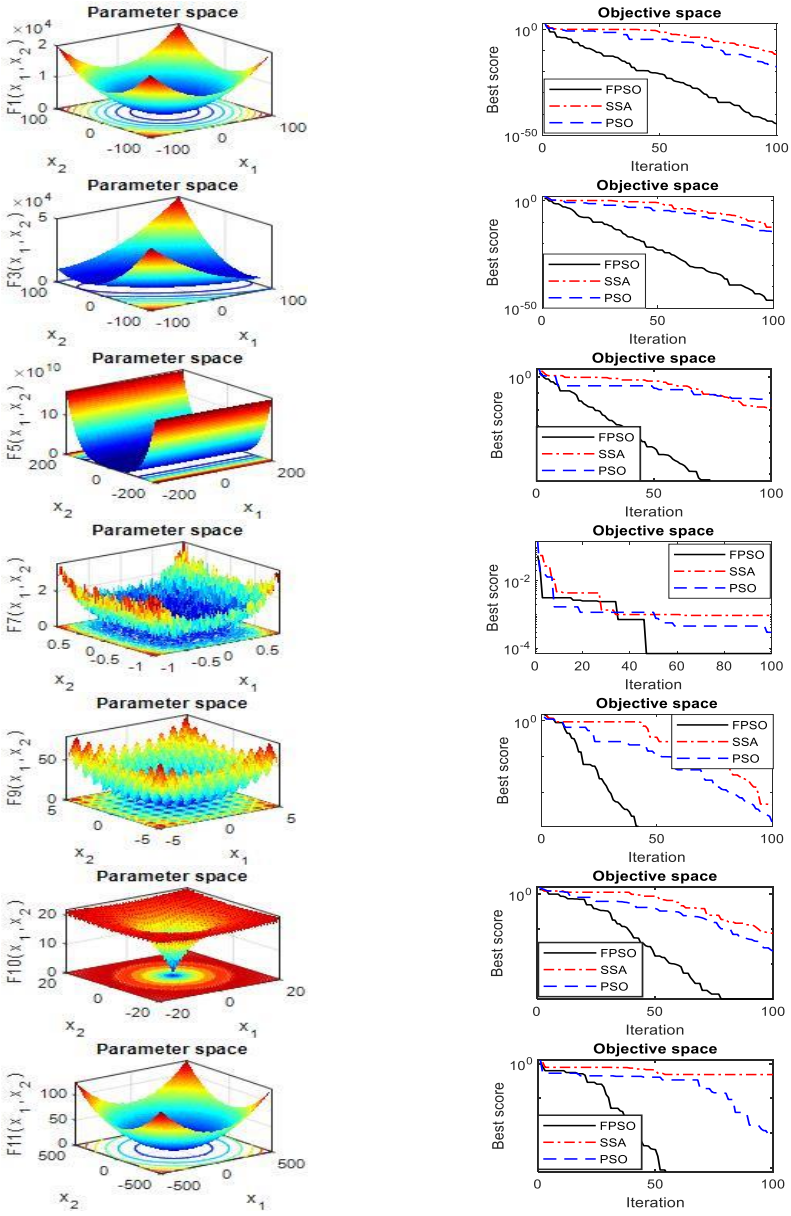
Function	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[-100,100]	0
$F_4(x) = max_i\{\lvert x_i \rvert,\ 1 \le i \le n\}$	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	[-30,30]	0
$F_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	[-100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	[-1.28,1.28]	0
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	[-500,500]	-418.98×5
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	[-32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]	0

نتایج حاصل شده از الگوریتم PSO فازی شده به FPSO نشان می‌دهد که این روش به صورت قابل ملاحظه‌ای پیشرفت داشته است. شکل ۴ شامل ۲ ستون است که ستون اول رسم توابع محک را در بازه‌های استفاده شده برای ارزیابی الگوریتم پیشنهادی نشان می‌دهد. برای ارزیابی از توابع محک F1، F3، F5، F7، F9، F10 و F11 استفاده شده است که به ترتیب در ستون اول از شکل ۴ نمایش داده شده است. همان‌طور که در شکل ۴ قابل مشاهده است، توابع محک F1 تا F6 دارای یک نقطه بهینه سراسری هستند در حالی‌که توابع F7 تا F11 دارای تعداد زیادی نقطه بهینه محلی و یک نقطه بهینه سراسری هستند. ستون دوم از شکل ۴ نمودارهای همگرایی را در تعداد ۱۰۰ تکرار برای سه روش PSO (رنگ آبی)، SSA (رنگ قرمز) و FPSO (رنگ سیاه) نمایش می‌دهد. در نمودارهای همگرایی مشاهده می‌شود که، با این‌که روش‌های PSO و SSA به نقطه بهینه یعنی صفر نزدیک شده‌اند اما روش FPSO توانسته است در تابع محک F7 در تعداد کمتر از ۵۰ تکرار به پاسخ بهینه سراسری صفر برسد که توانمندی این روش را در حل این نوع از مسائل بهینه‌سازی نشان می‌دهد این در حالی است که FPSO توانسته است در توابع محک F5 و F11 در حدود ۶۰ تکرار به پاسخ بهینه سراسری برسد که باز هم توانمندی روش را در

[1].Bhattacharyya coefficient

[2] Benchmark functions

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]



شکل ۴: توابع محک و نمودار همگرایی برای سه روش PSO، SSA و FPSO.

جدول۳: بهترین پاسخ

	F1	F3	F5	F7	F9	F10	F11
PSO	4.65E-17	4.59E-15	5.04E-07	1.97E-04	7.11E-15	4.95E-09	3.03E-011
SSA	6.94E-12	7.92E-12	1.70E-9	9.57E-05	3.14E-12	1.26E-06	0.0099
FPSO	**2.78E-45**	**4.99E-47**	**0**	**6.86E-05**	**0**	**8.88E-16**	**0**

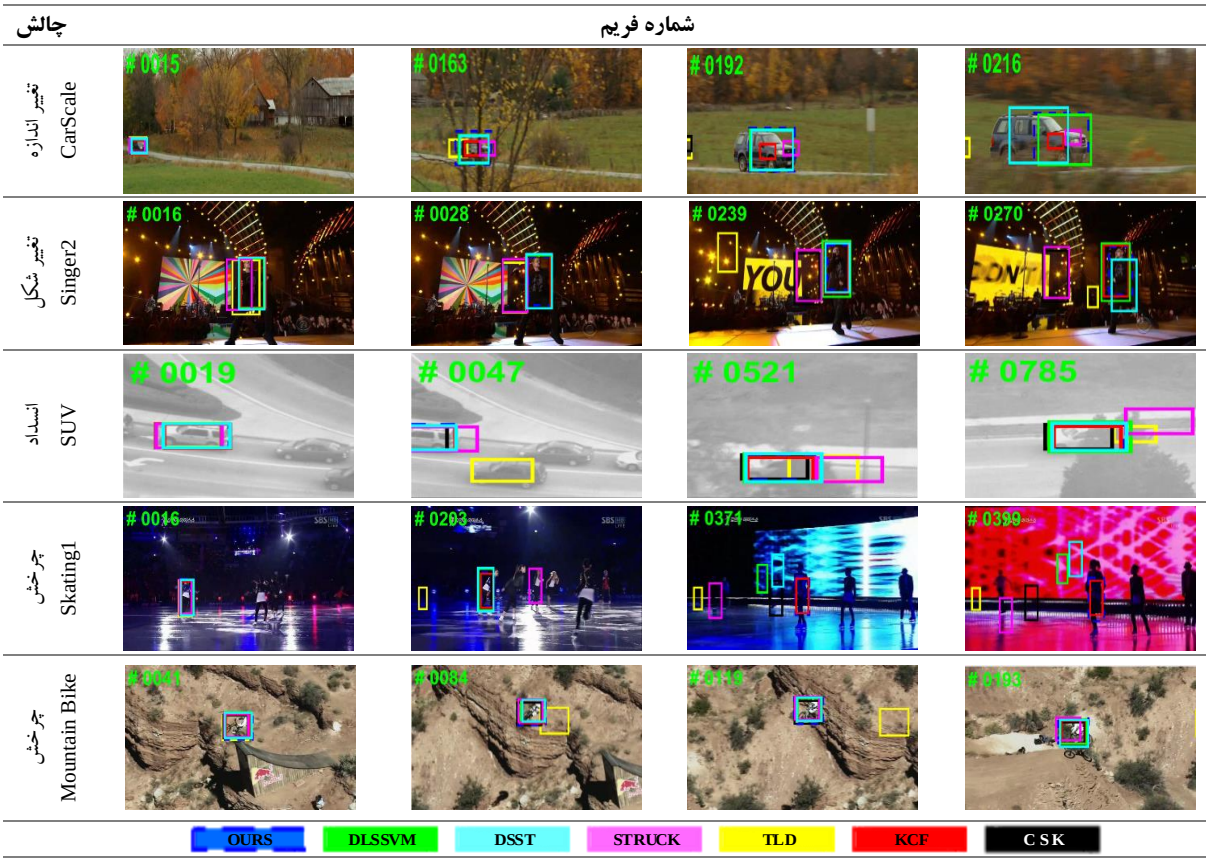
جدول۴: انحراف معیار بهترین پاسخ

	F1	F3	F5	F7	F9	F10	F11
PSO	40.5084	16.3141	8.2900	0.0191	0.4974	1.6649	0.1906
SSA	**7.2919**	5.0886	6.9000	0.0097	0.5130	1.3201	0.2107
FPSO	8.7670	**1.2328**	**2.8218**	**0.0061**	**0.1367**	**1.2196**	**0.1898**

این‌گونه مسائل نشان می‌دهد. در سایر توابع محک نیز FPSO بهتر از دو روش PSO و SSA عمل کرده است و به پاسخی بسیار نزدیک به پاسخ بهینه دست یافته است.

در جدول ۳ مقادیر بهترین پاسخ دست یافته در تعداد ۱۰۰ تکرار برای سه روش ذکر شده به صورت کمی ارائه شده است. جدول ۴ مقدار انحراف معیار را برای سه روش PSO، SSA و FPSO نشان می‌دهد. در این جدول مشاهده می‌گردد که الگوریتم SSA مقادیر انحراف معیارکمتری

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]



شکل ۵: ارزیابی ۵ دنباله ویدئویی برای ۴ چالش تغییر اندازه، تغییر شکل، انسداد و چرخش با استفاده از روش FPSO و ۶ روش مرسوم ردیابی.

نسبت به روش PSO دارد به طوری که در تابع محک F1 دارای کمترین انحراف معیار است و اختلاف انحراف معیار الگوریتم SSA با الگوریتم PSO بسیار زیاد است. در سایر توابع محک در جدول۴، الگوریتم FPSO دارای کمترین مقادیر انحراف معیار است که نتیجه آن پراکندگی کمتر ذرات در فضای جستجو است. علت این امر سرعت همگرایی بسیار بالای روش FPSO نسبت به روش‌های SSA و PSO است. همان‌طور که در نمودارها مشخص است، شیب نمودارها برای الگوریتم FPSO بسیار زیاد است که نشان می‌دهد سرعت همگرایی بالاتر از دو الگوریتم دیگر است. از آن جایی که در مسائل عملی، عموما با پاسخ‌های محلی مواجه هستیم بنابراین روش FPSO شایستگی خود را در هنگام مواجه شدن با این دسته از مسائل نشان می‌دهد.

مراحل اجرای روش پیشنهادی به صورت زیر است:

- مقادیر اولیه v_i و x_i تعیین می‌شوند.
- هدف در اولین فریم از ویدئو در چارچوب ردیابی[1] قرار گیرد.
- ویژگی شی هدف محاسبه می‌شود.
- برای تمامی فریم‌ها
- برای تمامی تکرارها
- v_i و x_i برای تمامی ذرات به روز رسانی می‌شوند.
- مقدار بهترین پاسخ در هر مرحله بازنشانی می‌شود.
- ضرایب w، C_1 و C_2 و $Popsize$ در هر مرحله بازنشانی می‌شوند.
- بهترین پاسخ با بیشترین شباهت و کمترین فاصله طبق فاصله باتاچاریا به عنوان مکان شی متحرک در فریم جاری انتخاب می‌شود.
- پایان.

۴-۲- پایگاه داده OTB50

پایگاه داده استفاده شده در این مقاله، پایگاه داده OTB50 نام دارد که از جمله پایگاه داده‌های متداول در ردیابی است [۳۷]. این پایگاه داده شامل ۵۱ دنباله تصویر ویدئویی است که هر یک از این ویدئوها شامل یک و یا چند چالش هستند. ویدئوها شامل ۱۱ چالش‌اند که می‌توان به تغییر شکل شی متحرک[2]، تغییر ابعاد[3]، چرخش[4]، انسداد[5] اشاره کرد. ارزیابی صورت گرفته روی این ۴ چالش است که شامل میزان متوسط دقت و متوسط نرخ موفقیت در مواجهه با هر یک از این ۴ چالش می‌باشد. الگوریتم پیشنهادی با روش‌های مرسوم در ردیابی مقایسه شده است که می‌توان در این حوزه

[1] Bounding Box
[2] Deformation
[3] Scale variation
[4] Rotation
[5] Occlusion

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]

به روش‌های CSK [۶]، TLD [۹]، KCF [۷]، STRUCK [۸]، DSST [۱۰] و DLSSVM [۳۸] اشاره کرد.

ارزیابی صورت گرفته در شکل ۵ برای روش FPSO و همچنین تعداد ۶ روش مرسوم در فرآیند ردیابی می‌باشد. در هر سطر آن یک چالش بررسی شده است. به این منظور از پایگاه داده OTB50 تعدادی از دنباله‌های ویدئویی انتخاب شده‌اند که چالش رخ داده در آن با شدت بیشتری اتفاق می‌افتد. سطر اول از شکل ۵ شامل تعدادی از فریم‌های دنباله‌ی CarScale است. در این دنباله تغییرات مقیاس و یا اندازه در طول دنباله، به عنوان چالش مطرح شده است. در فریم‌های اولیه تمامی ردیاب‌ها هدف را به درستی ردیابی می‌کنند اما با تغییر مقیاس هدف ردیاب TLD در فریم ۱۶۳ شروع به از دست دادن هدف می‌کند. به دنبال آن ردیاب CSK نیز هدف را از دست می‌دهد. سایر ردیاب‌ها از جمله ردیاب FPSO قادرند تا هدف را آخرین فریم حفظ کنند. در سطر دوم، دنباله Singer2 جهت بررسی چالش تغییر شکل انتخاب شده است. منظور از تغییر شکل، تغییر فرم هدف مورد نظر است. غالبا این تغییر برای اجسام غیرصلب مانند بدن انسان اتفاق می‌افتد. در دنباله Singer2، خواننده در طول ویدئو حرکت می‌کند و بدن آن دچار تغییرات ظاهری می‌گردد. البته علاوه بر آن تغییرات روشنایی پس‌زمینه در این دنباله بسیار شدید است. در این دنباله الگوریتم‌های TLD و STRUCK پس از گذشت ۲۰ فریم هدف را به کلی از دست می‌دهند. در ادامه روش CSK نیز از ردیابی هدف باز می‌ماند. در فریم‌های انتهایی روش DSST نیز از ردیابی منحرف شده است و فقط روش FPSO و روش DLSSVM قادر به حفظ هدف در چارچوب ردیابی هستند. سطر سوم دربردارنده دنباله SVU است که جهت بررسی چالش انسداد مطرح شده است. منظور از انسداد قرار گرفتن شی دیگری جلوی هدف ردیابی است به گونه‌ای که بخشی و یا تمامی ظاهر هدف پوشانده شود. در این دنباله خودرو به سرعت حرکت می‌کند و توسط درختان دچار انسداد می‌شود. ردیاب‌های TLD و CSK در این دنباله قادر به ردیابی هدف نیستند و هدف را در مواجه شدن با انسداد از دست می‌دهند. سطر چهارم دنباله Skating1 را برای چالش چرخش هدف معرفی می‌کند. در این دنباله شخص با سرعت زیاد حرکات نمایش شامل چرخش را انجام می‌دهد. در این دنباله ردیاب TLD و STRUCK هدف را در فریم‌های ابتدایی از دست می‌دهند. این در حالی است که پس از گذشت ۳۵۰ فریم ردیاب‌های DLSSVM، DSST و CSK نیز هدف را به کلی از دست می‌دهند. در این دنباله تنها روش FPSO و الگوریتم KCF قادرند تا انتهای دنباله هدف ردیابی را دنبال کنند. سطر پنجم نمونه‌ای دیگر از چرخش را طی دنباله MountainBike نشان می‌دهد. در فریم ۱۱۹ ام از این دنباله ردیاب TLD هدف را به طور کامل از دست می‌دهد این در حالی است که سایر ردیاب‌ها هدف را تا انتهای ویدئو حفظ می‌کنند.

جدول ۵: متوسط معیار دقت برای پایگاه داده OTB50

نام روش	تغییر شکل	تغییر اندازه	چرخش	انسداد
CSK [6]	۲۹/۸۰	۲۶/۵۲	۲۴/۳۱	۳۳/۳۱
TLD [9]	۳۴/۵۰	۳۴/۵۰	۳۶/۲۰	۳۸/۹۰
KCF [7]	۴۸/۶۰	۴۶/۹۰	۵۴/۹۰	۵۲/۶۰
STRUCK [8]	۴۰/۵۱	۳۹/۲۰	۳۸/۳۵	۳۹/۶۲
DSST [10]	۴۸/۸۰	۴۸/۰۰	۵۰/۹۰	۵۷/۵۰
DLSSVM [38]	۶۱/۱۱	۵۶/۹۰	۵۱/۶۰	**۶۸/۲۰**
FPSO	**۶۱/۱۵**	**۵۷/۵۰**	**۶۷/۰۴**	۶۶/۹۰

در جدول ۵ مقادیر کمی برای روش‌های مختلف برای معیار دقت نمایش داده شده است. از مقادیر ثبت شده و مقایسه آن‌ها می‌توان دریافت که روش FPSO در چالش‌های تغییر اندازه و تغییر شکل توانسته است عملکرد بهتری را نسبت به سایر روش‌ها به نمایش بگذارد درحالی‌که روش DLSSVM در چالش انسداد عملکرد بهتری را ارائه کرده است. اما روش FPSO نیز مقدار نزدیک به مقدار روش DLSSVM در چالش انسداد دارد. نتایج جدول ۵ برای سطح آستانه خطای بین صفر تا ۵۰ پیکسل بررسی شده است. منظور از سطح آستانه خطا، اختلاف فاصله‌ی مکان مرکز محاسبه شده و مرکز واقعی هدف در یک فریم است که به کلیه‌ی فریم‌ها تعمیم داده شده و متوسط گیری می‌شود. مقدار گزارش شده سطح زیر منحنی دقت برای سطح آستانه خطای صفر تا ۵۰ پیکسل بر حسب درصد محاسبه شده است.

جدول ۶: متوسط معیار نرخ موفقیت برای پایگاه داده OTB50

نام روش	تغییر شکل	تغییر اندازه	چرخش	انسداد
CSK [6]	۲۵/۲۲	۲۷/۴۲	۲۳/۱۲	۲۴/۷۳
TLD [9]	۲۸/۲۰	۳۲/۶۰	۲۸/۵۰	۲۷/۱۰
KCF [7]	۳۷/۹۰	۳۴/۳۰	۴۴/۷۰	۴۲/۶۰
STRUCK [8]	۳۲/۱۰	۳۳/۹۱	۳۰/۳۴	۲۸/۹۶
DSST [10]	۴۳/۴۰	**۴۵/۵۰**	۴۰/۸۰	۴۳/۶۰
DLSSVM [38]	۴۲/۹۰	۴۰/۵۰	۴۲/۴۰	**۴۷/۱۰**
FPSO	**۴۴/۸۰**	۴۴/۸۶	**۴۶/۵۰**	۴۶/۸۰

جدول ۶ نرخ موفقیت را به ازای ۶ روش مرسوم و روش FPSO در معیار نرخ موفقیت نمایش می‌دهد. از مقایسه نتایج به دست آمده می‌توان دریافت که روش FPSO در چالش‌های تغییر شکل و چرخش توانسته است عملکرد بهتری را نسبت به سایر روش‌ها به نمایش بگذارد. روش KCF پس از روش FPSO توانسته است تا در مواجه با چالش چرخش بهترین مقدار را کسب کند. اما روش DSST در چالش تغییر اندازه و روش DLSSVM در چالش انسداد عملکرد بهتری را ارائه کرده‌اند. با این حال روش FPSO مقادیر نزدیک به این دو روش را ثبت کرده است. مقادیر گزارش شده سطح زیر منحنی نرخ موفقیت برای سطح آستانه هم‌پوشانی صفر تا ۱۰۰ درصد است. در سطح آستانه ۱۰۰ درصد نرخ موفقیت کلیه روش‌ها

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]

نزدیک به صفر شده و در سطح آستانه‌های نزدیک به صفر نرخ موفقیت مقادیر بالایی دارند. بنابراین برای قابل اطمینان بودن یک روش باید به ازای تمامی سطوح آستانه هم‌پوشانی، نرخ موفقیت محاسبه شود و متوسط‌گیری شود و یا به عبارت دیگر سطح زیر منحنی نرخ موفقیت برای سطح آستانه هم‌پوشانی صفرتا ۱۰۰ درصد محاسبه شود.

جدول ۷ متوسط زمان اجرای هر الگوریتم را روی پایگاه داده OTB50 به ازای هر فریم بیان می‌کند. زمان اجرا بر حسب میلی‌ثانیه است. در این جدول مشاهده می‌شود که روش CSK دارای کمترین زمان اجرا است و به عبارتی سریعترین الگوریتم می‌باشد. پس از آن روش KCF دارای سرعت قابل توجه ۵/۴۰۵ میلی ثانیه است که در مقایسه با روش‌های جدیدتر و روش پیشنهادی دارای سرعت بالایی است. روش‌های DSST و DLSSVM با توجه به بالا بودن مقادیر نرخ موفقیت و دقت در ردیابی دارای سرعت‌های پایین هستند که این یک ضعف برای این روش‌ها محسوب می‌شود. این در حالی است که روش پیشنهادی با ثبت زمان ۱۶/۷۳۱ میلی ثانیه توانسته است سرعت قابل توجهی را با توجه به دقت و نرخ موفقیت بالای خود کسب کند.

۳-٤- نتیجه‌گیری و پیشنهادات

در این مقاله روش جدیدی به منظور بهبود عملکرد الگوریتم بهینه‌سازی ازدحام ذرات ارائه گردید و ایده استفاده از پارامترهای کنترلی به صورت فازی در این الگوریتم به جای استفاده از مقادیر ثابت و یا خطی و غیرخطی باعث گردید تا، علاوه بر رسیدن به پاسخ مناسب، سرعت همگرایی نیز بالاتر رود. نتایج تجربی نشان داد روش پیشنهادی در مقایسه با سایر روش‌ها ردیابی مورد مقایسه، توانسته است در ۵۱ دنباله ویدیو شامل چالش‌های متفاوت از پایگاه داده OTB50 بهترین نتایج را در معیار دقت روی چالش‌های تغییرات اندازه، تغییرات شکل و چرخش و نتیجه مطلوبی در چالش انسداد کسب کند. همچنین در معیار نرخ موفقیت در چالش‌های چرخش و تغییر شکل بهترین نتایج و در حالت تغییر اندازه و انسداد نیز نتایج مطلوبی را کسب نماید. این معیارها برتری روش پیشنهادی را نسبت به سایر روش‌های موجود نشان می‌دهد. الگوریتم PSO فازی شده علاوه بر کاربرد در زمینه ردیابی قابلیت استفاده در سایر زمینه‌های بهینه‌سازی را دارد. همان‌طور که در بخش نتایج ملاحظه شد، این روش عملکرد خوبی را در برابر توابع محک از خود نشان داد. استفاده از این روش در ردیابی باعث می‌شود تا به جای این که کلیه‌ی پنجره‌های یک تصویر اسکن شود، نقاط تصادفی از تصویر مورد بررسی قرار گرفته و برای هر نقطه تصادفی ویژگی رنگی وزن دار محاسبه شود و فاصله باتاچاریا برای هر کاندید محاسبه گردد. سپس در طول تکرارهای الگوریتم FPSO میزان این فاصله به حداقل برسد و مکان هدف در هر فریم با دقت مطلوبی شناسایی گردد. هر چند این روش هنوز نیاز به بررسی در زمینه دقت عملکرد دارد، با این حال، می‌توان ادعا کرد روش پیشنهادی ارائه شده در این مقاله می‌تواند اشیا متحرک را در یک دنباله ویدیویی به صورت مطلوبی ردیابی کند.

مراجع

[1] M. Swathy, P. Nirmala and P. Geethu, “Survey on vehicle detection and tracking techniques in video surveillance”, International Journal of Computer Applications, vol. 160, no. 7, pp. 22-28, 2017.

[2] I. Pham and M. Polasek, “Algorithm for military object detection using image data”, IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC), pp. 1-15, 2014.

[3] A. Yilmaz, O. Javed and M. Shah, “Object tracking: A survey”, Acm computing surveys (CSUR), vol. 38, no. 4, pp. 13-20, 2006.

[4] B. Zhuang, H. Lu, Z. Xiao, and D. Wang, “Visual tracking via discriminative sparse similarity map”, IEEE Transactions on Image Processing, vol. 23, no. 4, pp. 1872-1881, 2014.

[5] H. Farsi, “Improvement of minimum tracking in minimum statistics noise estimation method”, Signal Processing: An International Journal (SPIJ), vol. 4, no. 1, pp. 17-25, 2010.

[6] J. F. Henriques, R. Caseiro, P. Martins and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels”, European conference on computer vision, pp. 702-715, 2012.

[7] J. F. Henriques, R. Caseiro, P. Martins and J. Batista, “High-speed tracking with kernelized correlation filters”, IEEE Transactions on pattern analysis and machine intelligence, vol. 37, no. 3, pp. 583-596, 2015.

[8] S. Hare et al., “Struck: Structured output tracking with kernels”, IEEE Transactions on pattern analysis and machine intelligence, vol. 38, no. 10, pp. 2096-2109, 2016.

[9] Z. Kalal, K. Mikolajczyk and J. Matas, “Tracking-learning-detection”, IEEE Transactions on pattern analysis and machine intelligence, vol. 34, no. 7, pp. 1409-1422, 2012.

[0] M. Danelljan, G. Häger, F. S. Khan and M. Felsberg, “Discriminative scale space tracking,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 8, pp. 1561-1575, 2017.

[1] K. Zhang, L. Zhang, Q. Liu, D. Zhang and M. Yang, “Fast visual tracking via dense spatio-temporal context learning”, European conference on computer vision, pp. 127-141, 2014.

[2] D. S .Bolme, J. R. Beveridge, B. A. Draper and Y. M. Lui, “Visual object tracking using adaptive correlation filters”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2544-2550, 2010.

[3] M. Danelljan, G. Häger, F. Khan and M. Felsberg, “Accurate scale estimation for robust visual tracking”, British Machine Vision Conference, pp. 1-5, 2014.

[4] M. Gao, X. He, D. Luo, J. Jiang and Q. Teng, “Object tracking using firefly algorithm”, IET Computer Vision, vol. 7, no. 4, pp. 227-237, 2013.

[5] N. Hussain, A. Khan, S. G. Javed and M. Hussain, “Particle swarm optimization based object tracking using HOG features”, IEEE 9th International

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]

Conference on Emerging Technologies (ICET), pp. 1-6, 2013.

[6] F. Sha, C. Bae, G. Liu, X. Zhao, Y. Chung and W. Yeh, "A categorized particle swarm optimization for object tracking", IEEE Congress on Evolutionary Computation (CEC), pp. 2737-2744, 2015.

[7] C .Bae, K. Kang, G. Liu and Y. Chung, "A novel real time video tracking framework using adaptive discrete swarm optimization", Expert Systems with Applications, vol. 64, no. 1, pp. 385-399, 2016.

[8] J. Kennedy, "Particle swarm optimization", Encyclopedia of machine learning, pp. 760-766, 2010.

[9] S. Mirjalili, A. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems", Advances in Engineering Software, vol. 114, pp. 163-191, 2017.

[20] S. Mirjalili and A. Lewis, "The whale optimization algorithm", Advances in engineering software, vol. 95, pp. 51-67, 2016.

[21] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems", Knowledge-Based Systems, vol. 96, pp. 120-133, 2016.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization (PSO)", IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948, 1995.

[23] Y. Shi and R. Eberhart, "A modified particle swarm optimizer", IEEE international conference on evolutionary computation proceedings world congress on computational intelligence (Cat. No. 98TH8360), pp. 69-73, 1998.

[24] A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization", Computers and operations research, vol. 33, no. 3, pp. 859-871, 2006.

[25] H. Zhu, C. Zheng, X. Hu and X. Li, "Adaptive PSO using random inertia weight and its application in UAV path planning", Seventh International Symposium on Instrumentation and Control Technology: Measurement Theory and Systems and Aeronautical Equipment, vol. 7128, p. 712814-712819, 2008.

[26] T. Niknam, "A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem", Applied Energy, vol. 87, no. 1, pp. 327-339, 2010.

[27] Y. Maldonado, O. Castillo and P. Melin, "Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications", Applied Soft Computing, vol. 13, no. 1, pp. 496-508, 2013.

[28] T. Krzeszowski and K. Wiktorowicz, "Evaluation of selected fuzzy particle swarm optimization algorithms", Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 571-575, 2016.

[29] F. Olivas, L. Amador-Angulo, J. Perez, C. Caraveo, F. Valdez and O. Castillo, "Comparative study of type-2 fuzzy particle swarm, bee colony and bat algorithms in optimization of fuzzy controllers", Algorithms, vol. 10, no. 3, p. 101-128, 2017.

[30] F. Valdez, J. C. Vazquez and F. Gaxiola, "Fuzzy dynamic parameter adaptation in ACO and PSO for designing fuzzy controllers: the cases of water level and temperature control", Advances in Fuzzy Systems, vol. 2018, no. 1, pp. 1-19, 2018.

[31] B. Borowska, "Nonlinear inertia weight in particle swarm optimization", 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), vol. 1 ,pp. 296-299, 2017.

[32] C. Du, Z .Yin, Y. Zhang, J. Liu, X. Sun and Y. Zhong, "Research on active disturbance rejection control with parameter autotune mechanism for induction motors based on adaptive particle swarm optimization algorithm with dynamic inertia weight", IEEE Transactions on Power Electronics, vol. 34, no. 3, pp. 2841-2855, 2019.

[33] M. Swain and D. Ballard, "Color indexing", Computer Vision, vol. 7, no. 1, pp. 11-32, 1991.

[34] D. Zhang, J. Zhang and C. Xia, "Multi-complementary model for long-term tracking", Sensors, vol. 18, no. 2, pp. 527-552, 2018.

[35] L. Luo and X. Fan, "Immune particle filter algorithm for target tracking based on histograms of color and oriented gradient", Optical Sensing and Imaging Technology and Applications, vol. 10462, pp. 104622Q, 2017.

[36] A. Sharma, A. Malik and R. Rohilla, "A robust mean shift integrating color, GLCM based texture features and frame differencing", International Journal of Scientific and Engineering Research, vol. 7, no. 2, pp. 1386-1398, 2016.

[37] Y. Wu, J. Lim and M. Yang, "Online object tracking: A benchmark", IEEE Computer vision and pattern recognition (CVPR), pp. 2411-2418, 2013.

[38] J. Ning, J. Yang, S. Jiang, L. Zhang and M. Yang, "Object tracking via dual linear structured SVM and explicit feature map", Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4266-4274, 2016.

[Downloaded from joc.kntu.ac.ir on 2026-11-02]
[DOR: 20.1001.1.20088345.1399.14.4.11.5]
[DOI: 10.52547/joc.14.4.67]