فصلنامه مدیریت صنعتی دانشکده علوم انسانی، دانشگاه آزاد اسلامی
واحد سنندج – سال سیزدهم/ شماره ۴۳ / بهار ۱۳۹۷

# ارائه مساله طرح تاسیسات پویا با روش‌های فراابتکاری تک جواب و مبتنی بر جمعیت

**محمد مهدی کرم‌پور**
کارشناسی ارشد مهندسی صنایع، دانشکده فنی مهندسی، گروه مهندسی صنایع، دانشگاه علم و فناوری مازندران، بهشهر
**مصطفی حاجی آقائی کشتلی** (نویسنده مسؤول)
استادیار مهندسی صنایع، دانشکده فنی و مهندسی، گروه مهندسی صنایع، دانشگاه علم و فناوری مازندران، بهشهر
**Email: mostafahaji@mazust.ac.ir**

تاریخ دریافت: ۹۵/۴/۹ * تاریخ پذیرش: ۹۶/۱۱/۸

## چکیده

در جوامع مدرن امروزی، مراکز تولیدی باید قادر به ارائه پاسخی صریح، دقیق، فوری به تغییرات تقاضا و تبع آن تغییرات در محصول باشند، بنابراین در این مقاله در پی مطرح نمودن مساله چیدمان و بازآرایی تاسیسات مراکز تولیدی و خدماتی هستیم به‌گونه‌ای که مجموع هزینه‌های حمل و نقل مواد و بازآرایی خطوط تولیدی به کمترین میزان خود برسد. یک طرح بهینه باعث انتقال بهینه و کارا مواد بین تاسیسات و کاهش فرآیند کار و هزینه انبارداری می‌شود. برای مراکز تولیدی، هزینه حمل مواد مهمترین شاخص برای تعیین بهره‌وری یک طرح می‌باشد. این موضوع به عنوان مساله طرح تاسیسات پویا (DFLP) مطرح گردیده است. در این مقاله، DFLP با پنج روش فراابتکاری شامل ژنتیک (GA)، شبیه‌سازی تبرید (SA)، جستجوی ممنوع (TS)، ازدحام ذرات (PSO) و رقابت استعماری (ICA) مدل‌سازی شده است. روش تاگوچی جهت انتخاب پارامترهای بهینه به کار برده شده است. نتایج حاصله به خوبی نمایانگر آن‌اند که روش‌های فراابتکاری، راه‌کارهایی بهینه و موثر برای غلبه بر مساله طرح تاسیسات مراکز تولیدی و خدماتی ارائه می‌دهند. نتایج مدل‌سازی حاکی از آن هستند که الگوریتم رقابت استعماری در مقام مقایسه با دیگر روش‌ها، جواب‌های بهینه بهتری را در این مساله با محدودیت‌های مفروض در معیارهای مختلف ارائه می‌دهد.

**کلمات کلیدی:** الگوریتم رقابت استعماری، الگوریتم های فراابتکاری، روش تاگوچی، مساله طرح تاسیسات پویا.

## ۱- مقدمه

امروزه اهمیت مسائل جانمایی بر کسی پوشیده نیست. بسیاری از مراکز کارگاهی و صنعتی هر روزه با ورود تجهیزات و تکنولوژی جدید با این مساله رو به رو خواهند بود. از این رو، مساله طرح تاسیسات استاتیک[1] کاراترین ترتیب بخش‌ها در یک مرکز را مشخص می‌نماید. این مرکز می‌تواند کارخانه‌ها، ساختمان‌های اداری و یا مراکز خدماتی باشد. یک طرح کارا به دیگر عملیات‌های مبتنی بر جریان کار جهت اجرای مناسب کمک می‌کند. برای ساخت تاسیسات، هزینه نقل و انتقال مواد با اهمیت‌ترین معیار برای تعیین کارایی و بهره‌وری یک چیدمان است که اغلب ۲۰- ۲۵ ٪ کل هزینه‌های عملیاتی و ۱۵- ۷۰ ٪ از کل هزینه های ساخت یک کالا به آن تعلق دارد (Tompkins, White, Bozer, Frazelle, Tanchoco, & Trevino, 1996). سطوح توان عملیاتی، کیفیت و موجودی نیز به طور غیرمستقیم با جریان مواد مرتبط هستند. هزینه حمل مواد، بر اساس جریان مواد میان بخش‌ها و فواصل بین موقعیت بخش‌ها تعیین می‌گردد. اگر این انتقال مواد برای مدت زمان طولانی تغییر نکند، مساله طرح تاسیسات استاتیک می‌تواند برای تعیین بهترین طرح استقرار مورد استفاده قرارگیرد.

هنگامی که جریان مواد میان بخش‌ها در طول افق برنامه‌ریزی شروع به تغییر کند، مساله طرح تاسیسات از حالت ایستا به حالت پویا تغییر می‌یابد که آن را مساله طرح تاسیسات پویا[2] می‌نامند. اساس DFLP بر تغییرات جریان استوار است که ممکن است در آینده رخ دهد. افق برنامه‌ریزی (دورنمای آینده) می‌تواند به چندین بازه زمانی، هفته، ماه و یا سال، تقسیم شود. جریانی از داده برای هر دوره پیش‌بینی شده و فرض بر این است که جریان داده در طول دوره ثابت باقی می‌ماند، بنابراین مساله طرح تاسیسات را می‌توان برای هر دوره به عنوان یک SFLP در نظر گرفت و آن را به طور مستقل حل نمود.

اگر هزینه‌های انتقال مواد بسیار بزرگتر از هزینه های بازآرایی باشد،DFLP می‌تواند به صورت پی‌درپی به عنوان یک سری از SFLPها یا مسائل تخصیص مضاعف[3] حل شود. به عبارت دیگر، چیدمان برای دوره اول می‌تواند با حل QAP (SFLP) با استفاده از داده‌های تکی برای دوره اول به دست آید و سپس چیدمان برای دوره دوم را می‌توان با حل QAP (SFLP) با استفاده از داده‌های تکی برای دوره دوم به دست آورد. همچنین، اگر هزینه‌های بازآرایی بسیار بزرگتر از هزینه‌های انتقال مواد باشند، DFLP می‌تواند به عنوان یک سری از SFLPها (QAPها) حل شود. طرح به دست آمده برای دوره اول می‌تواند برای همه دوره ها اختصاص یابد، و کل هزینه حمل مواد به دست آید. علاوه بر این، طرح به دست آمده برای دوره دوم را نیز می‌توان به تمام دوره‌ها اختصاص داد، و کل هزینه حمل و انتقال مواد را به دست آورد. از این رو، تفاوت در هزینه های مادی حمل و نقل و بازآرایی به اندازه کافی کوچک هستند، به طوری که راه حل بهینه برای DFLP می‌تواند توسط روش‌های فوق به دست آید.

این مقاله سعی در حل مساله طرح تاسیسات پویا به کمک پنج الگوریتم فراابتکاری ژنتیک[4] ، شبیه‌سازی تبرید[5] ، جستجوی ممنوع[6] ، ازدحام ذرات[7] و رقابت استعماری[8] دارد، تا در نهایت بهترین استراتژی حل برای این مساله را بدست آورد.

ب) پیشینه پژوهش

واضح است که کارایی یک الگوریتم به شدت به پارامترهای آن وابسته می‌باشد، به گونه‌ای که پارامترهای مختلف ممکن است پاسخ‌هایی با کیفیت‌هایی کاملا متفاوت تولید کنند. همچنین، اگر تنظیم پارامترها به شکل درستی صورت نگیرد الگوریتم قادر به دست‌یابی به جواب‌هایی با کیفیت قابل قبول نمی‌باشد (Hajiaghaei-Keshteli, & Sajadifar, 2010). از این رو در این مقاله برای تنظیم کردن پارامترهای هر یک از الگوریتم‌های پیشنهادی از روش تاگوچی استفاده می‌گردد.

در مقایسه با SFLP، DFLP بسیار جدید است، روزنبلات (Rosenblatt, 1986) اولین کسی است که به‌طور کامل DFLP را معرفی نموده و یک الگوریتم برنامه‌ریزی پویا برای حل آن ارائه کرده است. همانند SFLP محاسبات DFLP برای مسائل

---

[1] Static facility layout problem (SFLP)
[2] Dynamic facility layout problem (DFLP)
[3] Quadratic assignment problem (QAP)
[4] Genetic Algorithm
[5] Simulated Annealing
[6] Tabu Search
[7] Particle Swarm Optimization
[8] Imperialist Competitive Algorithm

بزرگ نیاز به روش های فراابتکاری یا ابتکاری دارد، به عبارت دیگر، از آنجاییکه DFLP یک NP-hard می‌باشد، تعداد راه‌حل‌های ممکن یا طرح‌های چیدمان برای یک نمونه با N بخش و T دوره، $N!^{T}$ است، که تنها مسائل با سایز کوچک می‌توانند دریک زمان محاسباتی منطقی حل شوند.

اکثر روش‌های حل برایDFLP با استفاده از روش‌های ابتکاری صورت می‌گیرد. لاکسونن (Lacksonen TA, 1993) یک الگوریتم برای حل DFLP با تغییر مناطق هر بخش با استفاده از طرح پیوسته چیدمان ارائه کرده است. روزنبلات (Rosenblatt, 1986) دو هیوریستیک را که در هر دو آنها از برنامه‌ریزی پویا استفاده می‌شود را پیشنهاد داده‌اند. با این حال، اربن (Urban, 1993) هیوریستیک تعویض زوجی تندترین شیب را برای حلDFLP معرفی نموده، که شبیه بهCRAFT ارائه شده توسط آمور و بوفا (Armour, & Buffa 1963) برای SFLP است. تفاوت این دو هیوریستیک در هزینه‌های بازآرایی و مفهوم پنجره‌های پیش بینی است. پنجره پیش‌بینی، تعدادی از دوره‌های متوالی در جریان داده‌های جمع‌آوری شده برای بهبود یک طرح اولیه در یک دوره معین با به‌کارگیری هیوریستیک‌های زوجی با شیب تند می‌باشد. تعداد پنجره‌های پیش‌بینی برابر با تعداد دوره‌های زمانی (T) است. به‌ویژه، هیوریستیک اربن (Urban, 1993) با یک حل اولیه برای دوره ۱ و پنجره پیش‌بینی روی یک تنظیم می‌شود (بعنوان مثال m=1). بنابراین، جریان داده برای دوره ۱ و برای بهبود چیدمان اولیه از CRAFT استفاده می‌گردد. این چیدمان بهبودیافته در دوره ۱ بعنوان یک چیدمان اولیه برای دوره ۲ و نیزCRAFT و جریان داده در دوره ۲ برای تولید یک چیدمان بهبودیافته برای دوره ۳ مورد استفاده قرار می‌گیرند. این فرایند تا رسیدن به یک چیدمان بهبودیافته برای تمام دوره‌ها ادامه می‌یابد. بعنوان یک نتیجه، برای پنجره‌های پیش‌بینی T، T جواب بدست می‌آید و از بین آنها بهترین جواب انتخاب می‌شود.

چندین الگوریتم ابتکاری پیشنهادی برای DFLP نیز وجود دارد. کانوی و ونکاتارامان (Conway, & Venkataramanan, 1994)، DFLP را با الگوریتم ژنتیک ساده حل کرده‌اند. بالاکریشنان و چنگ (Balakrishnan, & Cheng, 2000) یک الگوریتم ژنتیک حلقه‌های تودرتو[9] را برای حلDFLP ارائه داده‌اند. در حلقه درونی، اپراتورهای ژنتیک به منظور بهبود برخی از جواب‌ها در جمعیت استفاده می‌شوند. حلقه خارجی شامل دوره‌های معین جایگزینی تعدادی از جواب‌های نامطلوب در جامعه است که تضمین می‌کند حلقه داخلی با جمعیت های مختلف کار می‌کند. این اقدام فضای جستجو را گسترش داده و منجر به راه‌حل‌هایی با کیفیت بالا می‌شود. کاکو و مازولا (Kaku, & Mazzola, 1997)، یک الگوریتم جستجوی ممنوع برایDFLP ارائه داده‌اند. بالاکریشنان و همکاران (Balakrishnan, Cheng, & Conway, 2000) دو الگوریتم ابتکاری پیشنهاد داده‌اند که الگوریتم ابتکاری مبادله زوجی تندترین شیب اربن را بهبود بخشیده‌اند. اولین الگوریتم ابتکاری، از الگوریتم ابتکاری اربن، برای تولید راه‌حل می‌کند که این راه‌حل‌های تولید شده برای هر پنجره پیش بینی، با استفاده از الگوریتم ابتکاری مبادله زوجی بازگشت به عقب، بهبود یافته و در نهایت بهترین راه‌حل انتخاب می‌شود. روش ابتکاری دوم به این صورت است که الگوریتم ابتکاری اربن را با برنامه‌ریزی پویا ترکیب می‌کند. بالاکریشنان و همکاران (Balakrishnan, Cheng, Conway, & Lau, 2003) یک الگوریتم ژنتیک ترکیبی برای DFLP، که از یک عملگر تقاطع[10] ابتکاری بر اساس برنامه‌نویسی پویا استفاده می کند، ارائه داده‌اند. ابتدا، هر طرح چند دوره‌ای در داده‌های والد، به تعدادی طرح تک دوره‌ای، تجزیه می‌شود. سپس، با حذف طرح‌های تکراری یا کپی شده، بهترین ترکیب در میان تمام طرح‌بندی‌ها با استفاده از برنامه‌ریزی پویا به دست می‌آید.

لاکسونن و انسکور (Lacksonen, & Enscore, 1993) پنج الگوریتم اصلاحی برای حلDFLP بکار گرفته‌اند. این پنج الگوریتم براساس ۱. روش ابتکاری برنامه‌ریزی پویای ارائه شده توسط روزنبلات (Rosenblatt, 1986)، ۲. الگوریتم شاخه و کران برایQAP توسط پاردالوس و کروز (Pardalos, & Crouse, 1989)، ۳. الگوریتم صفحه برش برایQAP توسط بورکارد و بونیگر (Burkard, & Bonniger, 1983)، ۴. برش درخت توسط گوموری و هو (Gomory, & Hu, 1961)، و ۵. کرافت (CRAFT) برای QAP توسعه یافته توسط آمور و بافا (Armour, & Buffa 1963)، بوده است. همچنین،

[9] Nested loops genetic algorithm (NLGA)
[10] Crossover

بالاکریشنان و چنگ (Balakrishnan, & Cheng, 2000) الگوریتم ژنتیک ارائه شده توسط کانوی و ون کاتارامانان (Conway, & Venkataramanan, 1994) را برای حل DFLP بهبود داده‌اند. بایکاسوگلو و جیندی (Baykasoglu, & Gindy, 2001) و بالاکریشنان و همکاران (Balakrishnan, Cheng, Conway, & Lau, 2003) به ترتیب روش ابتکاری شبیه‌سازی‌شده تبرید و الگوریتم ژنتیک ترکیبی برای DFLP را پیشنهاد کرده‌اند. اخیرا، ارل و همکاران (Erel, Ghosh, & Simon, 2003) یک رویکرد سه فازی برای حل DFLP معرفی نموده‌اند. در وهله اول، مجموعه‌ای از چیدمان‌های "خوب" با ترکیب داده‌های جریان از T دوره با استفاده از یک طرح وزن‌دهی ایجاد شده است و برای حل SFLP در هر دوره از داده‌های جریان ترکیب‌شده استفاده می‌شود. در وهله دوم، مجموعه‌ای از راه‌حل‌هایی که در مرحله اول و با برنامه‌ریزی پویا به دست آمده‌اند، برای رسیدن به راه‌حل‌هایی برای DFLP استفاده می‌شوند. در پایان، راه حل بدست آمده در مرحله دوم با استفاده از یک استراتژی تبادل زوجی به صورت تصادفی بهبود می یابد. همچنین، مک‌کندال و شانگ (McKendall, & Shang, 2006) سیستم‌های ترکیبی را برای حل DFLP ارائه کرده اند.

در ادامه مقاله در بخش ۲ روش شناسی پژوهش متشکل از بیان مساله، چهارچوب مفهومی و روش‌های پیشنهادی ارائه می‌شود. در بخش ۳ به انجام آزمایش‌های موردنظر و نتایج بدست آمده به همراه پیشنهادهایی برای تحقیقات آتی پرداخته می‌شود. درنهایت در بخش ۴ فهرست منابع استفاده شده ارائه می‌گردد.

## ۲- روش‌شناسی

تحقیق حاضر با هدف ارائه بهترین الگوریتم به منظور حل مساله طرح تاسیسات پویا انجام‌شده است. می‌توان بیان داشت که روش های فراابتکاری و ابتکاری می توانند برای حل مسائل طرح تاسیسات پویا بسیار کارا باشد. در این تحقیق نه تنها از الگوریتم‌های قدرتمند تک جواب بهره گرفته شده، بلکه از روش های مبتنی بر جمعیت با استفاده از رویکردهای نوین نیز استفاده گردیده است، تا با مقایسه روش‌های گوناگون کاراترین روش‌ها مشخص گردند.

الف) مدل مفهومی

مفروضات برای DFLP استاندارد (McKendall, & Shang, 2006) به صورت زیر توضیح داده شده‌اند:

- جریان بین بخش‌ها در تاسیسات، پویا و قطعی می‌باشد.
- افق برنامه‌ریزی به T دوره تقسیم شده است.
- در هر دوره زمانی، هر یک از تاسیسات تنها باید در یک محل قرار داده شوند.
- در هر دوره زمانی، جریان مواد بین هر جفت از تاسیسات از قبل مشخص می‌باشد و در طول دوره تغییر نمی‌کند.

هدف، به دست آوردن طرح چیدمانی است که مجموع هزینه‌های جابجایی مواد و بازآرایی را به حداقل برساند(به عنوان مثال، طرحی برای همه دوره‌ها).

$$\min z = \sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N} A_{t,i,j,l} \times Y_{t,i,j,l} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} c_{t,i,j,k,l} \times X_{t,i,j} \times X_{t,k,l} \qquad (۱)$$

Subject to.

$$\sum_{j=1}^{N} X_{t,i,j} = 1 \qquad i = 1,2,...,\mathrm{N}; \qquad t = 1,2,...,T \qquad (۲)$$

$$\sum_{i=1}^{N} X_{t,i,j} = 1 \qquad i = 1,2,...,\mathrm{N}; \qquad t = 1,2,...,T \qquad (۳)$$

$$Y_{t,i,j,l} = X_{t-1,i,j} \times X_{t,i,l} \qquad i,j,l = 1,2,...,N; \qquad t = 2,...,T \qquad (۴)$$

$$X_{t,i,j} \in \{0,1\} \qquad i,j = 1,2,...,N; \qquad t = 1,2,...,T \qquad (۵)$$

$$Y_{t,i,j,l} \in \{0,1\} \qquad i,j,l = 1,2,...,N; \qquad t = 2,...,T \qquad (۶)$$

$N$: تعداد بخش‌ها/مکان‌های تاسیسات

$T$: تعداد دوره‌های زمانی مورد نظر

$i, j, k, l$: شاخص برای بخش‌ها/مکان‌های تاسیسات

$t$: شاخص برای دوره‌های زمانی موردنظر

$A_{t,i,j,l}$ : هزینه انتقال بخش i از مکان j به l در دوره t

$C_{t,i,j,k,l}$ : هزینه جریان مواد در دوره t، بین بخش i که در مکان j قرار گرفته و بخش k که در مکان l می‌باشد.

متغیرهای تصمیم مدل (باینری):

$X_{t,i,j}$ : برابر با ۱ خواهد بود اگر بخش i در دوره t به مکان j تخصیص یابد، در غیراینصورت برابر با ۰ می‌باشد.

$Y_{t,i,j,l}$ : برابر با ۱ خواهد بود اگر بخش i از مکان j به l، در ابتدای دوره t، انتقال یابد و در غیراینصورت برابر با ۰ است.

تابع هدف (۱) برای کمینه کردن مجموع هزینه‌های بازآرایی و جابه‌جایی مواد است. مجموعه محدودیت (۲) تضمین می‌کند که هر مکان تنها به یک بخش در هر دوره تخصیص می‌یابد، و مجموعه محدودیت (۳) تضمین می‌کند که دقیقا یک بخش به هر مکان در هر دوره اختصاص داده شود. در مجموعه محدودیت (۴) اگر یک بخش بین مکان‌ها در دوره‌های متوالی جابه‌جا شود با اضافه کردن هزینه‌های بازآرایی، به هزینه جابه‌جایی مواد کمک می‌کند(در جلوگیری از بالا رفتن آن موثر می‌باشد). در نهایت، مجموعه محدودیت‌های (۵) و (۶) متغیرهای تصمیم مدل را تعیین می‌کنند.

به منظور پیاده‌سازی این مساله، پنج روش الگوریتم‌های فراابتکاری استفاده کرده است. به منظور جستجوی فضاهای شدنی این روش‌ها در مسائل NP-hard برای دست‌یابی به پاسخ‌های بهینه می‌باشند. الگوریتم‌های بهینه‌سازی در دو دسته دقیق[11] و تقریبی[12] جای می‌گیرند، بطوری که زمان اجرای آن‌ها متناسب با ابعاد مسائل به صورت نمایی افزایش می‌یابد و ممکن است هیچ‌گاه به پاسخ نرسند. برای غلبه بر این مشکل الگوریتم‌های ابتکاری ارائه شدند که مشکل عمده آن‌ها گیر افتادنشان در فضاهای بهینه محلی[13] است، در واقع الگوریتم‌های فراابتکاری راه حلی برای فرار از این نقاط با الهام گرفتن از پدیده‌های طبیعی و روزمره زندگی انسان هستند.

الگوریتم‌های فراابتکاری را می‌توان در دو دسته جمعیت محور مانند، الگوریتم ژنتیک (GA) و ازدحام ذرات (PSO) و الگوریتم رقابت استعماری یا نقطه محور مانند شبیه سازی تبرید (SA) و جستجوی ممنوع (TS) طبقه‌بندی کرد.

ب) الگوریتم ژنتیک

بر پایه نقش ژنتیک در طبیعت و تکامل طبیعی جانداران، جان هالند (۱۹۷۵) گونه منحصر به فردی از الگوریتم‌های تکاملی را در اوایل دهه ی ۷۰ میلادی به نام الگوریتم ژنتیک ارائه داد. پس از مشخص نمودن جواب اولیه باید با استفاده از دو عملگر تقاطع و جهش[14] اقدام به ایجاد کروموزوم‌های جدید موسوم به فرزند شود. سپس با عمل ارزیابی (برازش)، مهمترین بحث در فرآیند انتخاب، بهترین کروموزوم ها انتخاب می‌شوند. بر این اساس، پس از تکرار چند نسل، بهترین نسل که همان پاسخ بهینه مساله است، ایجاد خواهد شد. شبه کد الگوریتم ژنتیک در ادامه آمده است.

[11] Exact
[12] Approximate
[13] Local optimum
[14] Mutation

```
Initialize the parameters.
while (t<maximum number of iteration)
    Select the two solutions by roulette wheel selection.
    Perform crossover on the two mentioned solutions.
    Perform mutation on the percent of the new generation.
    Merge the next generation.
    t=t+1
endwhile
```

شکل شماره (۱): شبه کد الگوریتم ژنتیک

پ) الگوریتم رقابت استعماری

الگوریتم رقابت استعماری نه از پدیده‌ای اجتماعی اقتباس گشته است. به طور خاص این الگوریتم به پروسه استعمار به عنوان مرحله‌ای از تکامل سیاسی – اجتماعی نوع بشر نگریسته و با مدل‌سازی ریاضی این پدیده تاریخی از آن به عنوان منشا الهام یک الگوریتم بهینه‌سازی استفاده می‌کند. به منظور توضیح بیشتر، شبه–کد الگوریتم رقابت استعماری در شکل (۲) آورده شده است.

```
Generate a population randomly named as countries.
Initialize empires and their colonies.
while (t<maximum number of iteration)
    for each empire
        move the colony toward the relevant empires.
        Compute the cost of assimilated country.
        if the cost of new country is better than empire
       Exchange the empire and its country.
    endif
        Pick the weakest colony form the weakest empire.
        Assign it to the empire that has most likelihood to posse it.
    endfor
    if there is an emperialist has no colonies.
      Eliminate this empire.
    endif
```

```
        Update the new generation.
        T=t+1;
Endwhile
```

شکل شماره (۲): شبه کد الگوریتم رقابت استعماری

ت) الگوریتم ازدحام ذرات

نخستین بار کندی و ابرهارت در دهه ۹۰ میلادی با الهام از حرکات پرندگان و با کشف رابطه منطقی میان تغییر جهت و سرعت پرندگان و علم به اشتراک‌گذاری اطلاعات میان مجموعه خودشان (پرندگان) و با کمک علم فیزیک روشی پیشنهادی با عنوان ازدحام ذرات را ارئه نمودند. برای اطلاعات بیشتر در مورد مراحل الگوریتم در شکل (۳) شبه کد الگوریتم آمده است.

```
Set the parameters.
Generate initial particles (P).
T1=clock;
it=1;
while (t< maximum time of simulation)
    for each particle p in P
        fp=f(p); /*evaluate the particles*/
        if fp is the better than pbest
            pbest=p;
        endif
    endfor
    gbest= best p in P;
    for each particle p in P
        v= w*v+c1*rand*(pbest-p)+c2*rand*(gbest-p);
        p=p+v;
    endfor
    w=w*;
    it=it+1;
    T2=clock;
    t=etime(T2,T1);
endwhile
return gbest
```

شکل شماره (۳): شبه کد الگوریتم هوش تجمعی ذرات

ث) الگوریتم شبیه سازی تبرید

الگوریتم فراابتکاری شبیه سازی تبرید (SA) را نخستین بارکپاتریک و همکارانش در دهه ۸۰ میلادی به جوامع علمی ارائه کردند. الگوریتم شبیه‌سازی تبرید جزء روش‌های جستجوی نقطه‌ای (همسایگی) است که به دلیل پذیرفتن پاسخ های غیر بهبوددهنده تابع هدف، بر خلاف سایر روش‌های جستجوی نقطه‌ای به نقطه شروع (راه حل اولیه) بستگی ندارد و می‌تواند از دام بهینه‌های محلی تا حد زیادی بگریزد.

نحوه کار این الگوریتم بدین شکل است که نخست یک جواب شدنی اولیه به دست می‌آید، پس از آن برای تولید جواب‌های همسایگی از عملگر جهش استفاده می‌شود و در ادامه با بکارگیری تابع احتمال بولتزمان $p = e^{\frac{-\Delta E}{T}}$ برای پذیرش پاسخ های بدتر نسبت به پاسخ‌های پیشین و خروج از بهینه محلی بهره می‌برد. سپس با الگویی خاص پارامتر دما را در هر تکرار کاهش داده تا شرایط اختمام و توقف الگوریتم مهیا شود. شرایط خاتمه گوناگونی می‌توان برای یک الگوریتم شبیه‌سازی تبرید به کار گرفت که از جمله آن‌ها می توان به تعداد دفعات کاهش سطح دما، تعداد دفعات متوالی که در تابع هدف بهبودی مشاهده نشود، رسیدن به دمای پایانی مورد نظر و تعداد تکرار الگوریتم اشاره کرد. شبه کد الگوریتم تبرید در شکل (۴) آورده شده است.

```
Select a random solution X*.
Initialize the parameters.
while (t< maximum number of iteration)
    sub=0;
    while (sub< maximum number of sub-iteration)
      Create a neighbor of this solution.
      if the function value of the new solution is better than prior
        Replace the new solution as old solution.
      else
        Calculate δ, δ = |f_old − f_new|.
          if rand< exp(−δ/T)
            Replace the new solution.
          endif
      endif
    sub=sub+1;
    endwhile
    Update T.
    Update the X* if there is better solution.
    t=t+1;
endwhile
return X*
```

شکل شماره (۴): شبه کد الگوریتم تبرید شبیه‌سازی

ث) الگوریتم جستجوی ممنوع

الگوریتم جستجوی ممنوع یک الگوریتم بهینه‌سازی فراابتکاری است که نخستین بار در دهه ۸۰ میلادی توسط گلوور معرفی گردید. برای رسیدن به یک پاسخ بهینه در مسائل بهینه‌سازی با الگوریتم جستجوی ممنوع ابتدا از یک جواب اولیه شروع به حرکت کرده و سپس الگوریتم بهترین جواب همسایه را از میان همسایه‌های جواب فعلی انتخاب می‌کند، چنانچه این جواب در فهرست ممنوع نباشد، الگوریتم به جواب همسایه‌ای که ایجاد شده طبق پارامترهای خود حرکت می‌کند، در غیراین‌صورت

الگوریتم با بهره‌مندی از معیاری به نام معیار تنفس[15] اقدام به بهبود راه‌حل‌ها می‌کند. بر اساس معیار تنفس اگر جواب همسایه از بهترین جواب یافت شده تا کنون بهتر باشد، الگوریتم به آن حرکت خواهد کرد، حتی اگر آن جواب در فهرست ممنوع باشد که این خود یکی از ابزارهای این الگوریتم در کنار فهرست ممنوع برای فرار از بهینه محلی می‌باشد. شروط خاتمه گوناگونی نظیر تعداد تکرار، زمان سپری شده از شروع جستجوی الگوریتم، عدم بهبود جواب‌ها و یا همگرایی به پاسخ بهینه را می‌توان برای الگوریتم جستجوی ممنوع در نظر گرفت. شبه کد الگوریتم جست و جوی ممنوع در شکل (۵) آورده شده است.

Choose an initial solution $X$ in S.
Set X*=X.
*while* ($t$< maximum number of iteration)
    Generate a subset $V^*$ of solution in $N_{(X,t)}$ such that either one of Tabu conditions is violated or at least one of the aspiration conditions holds.
    Choose a best $Y$ in $V^*$ and set $X = Y$.
    *if* f($X$)<f(X*)
      Set X*=X.
    *endif*
    Update Tabu and aspirations conditions.
    $t$=$t$+1
*endwhile*
return X*

شکل شماره (۵): شبه کد الگوریتم جست و جوی ممنوع

## ۳- نتایج و بحث

الف) نمونه‌ها

اندازه مسائل پیشنهادی از مقاله‌ی مرتبط (Pourvaziri, & Naderi, 2014) در شش اندازه مختلف الهام گرفته شده است. مقدار تقریبی پارامترها با استفاده از مقالات مرتبط در جدول (۱) خلاصه گردیده است.

جدول شماره (۱): اندازه‌های مسائل آزمایشی و مقادیر پارامترها

| شماره مساله | اندازه مشخصه N | اندازه مشخصه T | توزیع پارامترها |
|---|---|---|---|
| ۱ | ۶ | ۵ | U(۳۰۰ – ۱۴۳۰) |
| ۲ | ۶ | ۱۰ | U(۲۵۰ – ۱۲۰۰) |
| ۳ | ۱۵ | ۵ | U(۲۰۰ – ۸۰۰) |
| ۴ | ۱۵ | ۱۰ | U(۱۵۰ – ۶۰۰) |
| ۵ | ۳۰ | ۵ | U(۱۰۰ – ۴۰۰) |
| ۶ | ۳۰ | ۱۰ | U(۵۰ – ۲۰۰) |

ب) تنظیم پارامترها به روش تاگوچی[16]

پر واضح است که کارایی یک الگوریتم شدیدا به پارامترهای آن وابسته می‌باشد، به‌طوریکه پارامترهای گوناگون ممکن است پاسخ‌هایی با کیفیت‌های کاملا متفاوت تولید کنند. همچنین، اگر پارامترها به گونه‌ای صحیح تنظیم نشوند الگوریتم توانایی لازم برای دست‌یابی به جواب‌هایی با کیفیت قابل قبول ندارد. به‌منظور تنظیم کردن پارامترهای الگوریتم‌ها، مساله‌ای با سایز متوسط

[15] Aspiration
[16] Taguchi

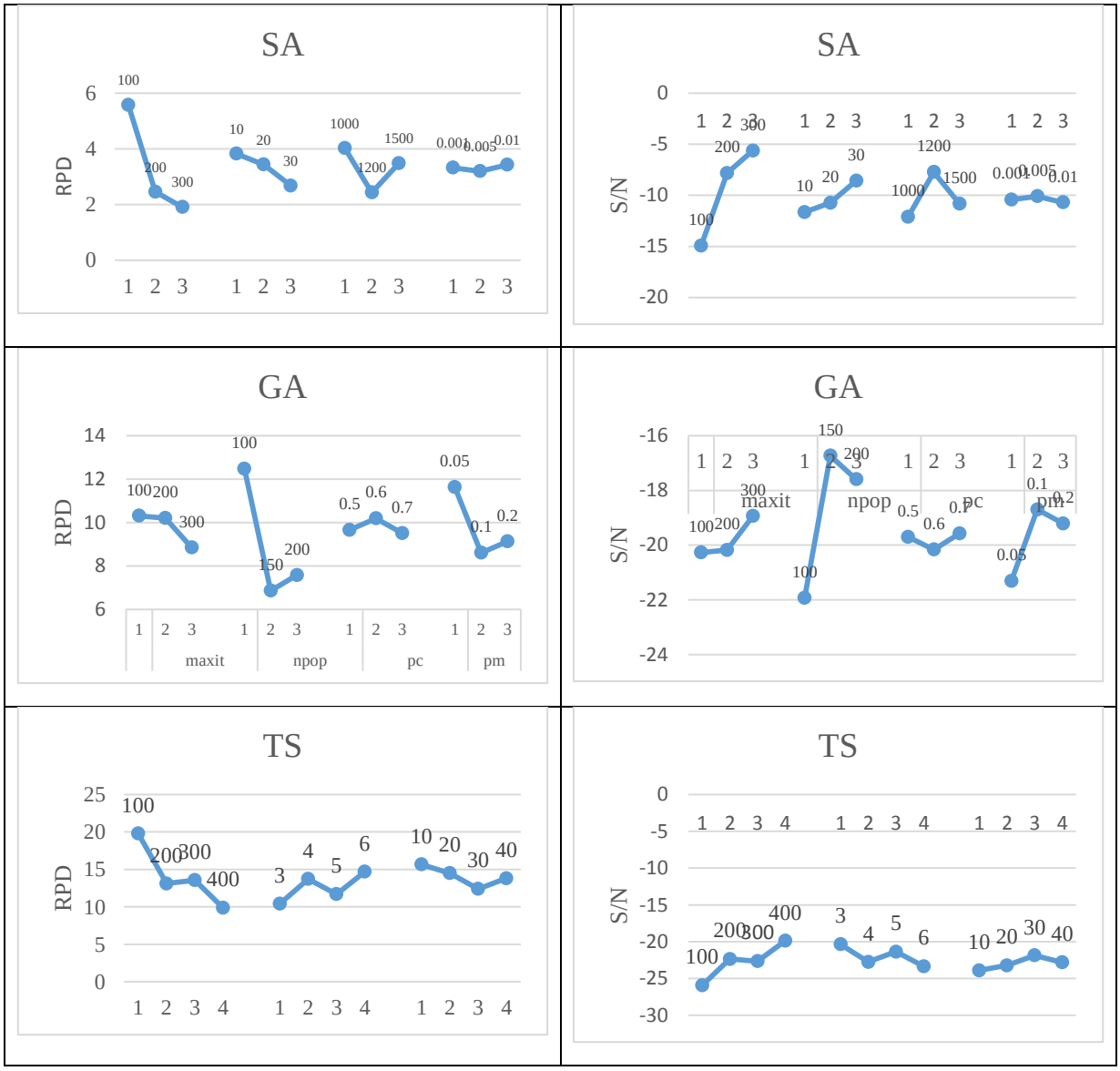
انتخاب کرده و بهترین مجموعه پارامترها تخمین زده می‌شوند. برای هر پارامتر الگوریتم چند سطح با توجه به اهمیت تنظـیم آن پیشنهاد می شود. مقادیر ارائه شده برای الگوریتم ها در جدول (۲) آمده است. معیارهای پراکندگی جواب های الگوریتم از بهترین جواب[17] و سیگنال پاسخ هر سطح[18] برای به دست آوردن بهترین سطح پارامترها، ارزیابی شده که نتایج آن در شکل (۶) برای هر الگوریتم آورده شده است. برای مطالعه بیشتر می توان به مقـالات حـاجی‌آقـایی و همکـاران (Hajiaghaei-Keshteli, & Sajadifar, 2010; Hajiaghaei-Keshteli, Sajadifar, & Haji, 2011) مراجعه کرد.

جدول شماره (۲): مقادیر پیشنهادی برای الگوریتم‌ها

| الگوریتم | پارامتر | پارامتر سطح اول | پارامتر سطح دوم | پارامتر سطح سوم | پارامتر سطح چهارم |
|---|---|---|---|---|---|
| GA | Maximum number of iteration=Maxit | ۱۰۰ | ۲۰۰ | ۳۰۰ | - |
| | Number of initial population=nPop | ۱۰۰ | ۱۵۰ | ۲۰۰ | - |
| | Percent of crossover=pc | ۰/۵ | ۰/۶ | ۰/۷ | - |
| | Percent of mutation=pm | ۰/۰۵ | ۰/۱ | ۰/۲ | - |
| SA | Maximum number of iteration=Maxit | ۱۰۰ | ۲۰۰ | ۳۰۰ | - |
| | Maximum number of sub-iteration=Sub-it | ۱۰ | ۲۰ | ۳۰ | - |
| | Temperature =T | ۱۰۰۰ | ۱۲۰۰ | ۱۵۰۰ | - |
| | Rate of reduction=R | ۰/۰۰۱ | ۰/۰۰۵ | ۰/۰۱ | - |
| TS | Maximum number of iteration=Maxit | ۱۰۰ | ۲۰۰ | ۳۰۰ | ۴۰۰ |
| | Number of neighbour=NN | ۳ | ۴ | ۵ | ۶ |
| | Tabu list=TL | ۱۰ | ۲۰ | ۳۰ | ۴۰ |
| ICA | Maximum number of iteration=Maxit | ۱۰۰ | ۲۰۰ | ۳۰۰ | - |
| | Number of initial population=nPop | ۱۰۰ | ۱۵۰ | ۲۰۰ | - |
| | Number of empires=Nemp | ۵ | ۸ | ۱۰ | - |
| | Assimilation rate=As | ۰/۳ | ۰/۵ | ۰/۶ | - |
| | Betta=b | ۲ | ۴ | ۵ | - |
| | Zetta=z | ۰/۰۵ | ۰/۱ | ۰/۱۵ | - |

[17] RPD (Relative of Percent Deviation)
[18] S/N (Sense to Noise)

| | | | | | |
|---|---|---|---|---|---|
| PSO | Maximum number of iteration=Maxit | ۱۰۰ | ۲۰۰ | ۳۰۰ | - |
| | Number of initial population=nPop | ۱۰۰ | ۱۵۰ | ۲۰۰ | - |
| | Individual learning=c1 | ۱/۲ | ۱/۵ | ۲ | - |
| | Social learning=c2 | ۱/۲ | ۱/۵ | ۲ | - |
| | Inertia weight=w | ۰/۸ | ۰/۹ | ۱ | - |




شکل شماره (۶): مقادیر پراکندگی انحراف پارامترهای الگوریتم‌ها و نرخ پاسخ آنها

برای الگوریتم ژنتیک و شبیه‌سازی تبرید از جدول L۹ استفاده شده و برای الگوریتم رقابت استعماری و ازدحام ذرات از طرح ۲۷ L و در نهایت برای الگوریتم جستجوی ممنوع طرح ۱۶ L پیشنهاد گردیده است. جدول (۳) مقادیر نهایی الگوریتم‌ها برای حل در مساله نشان می‌دهد.

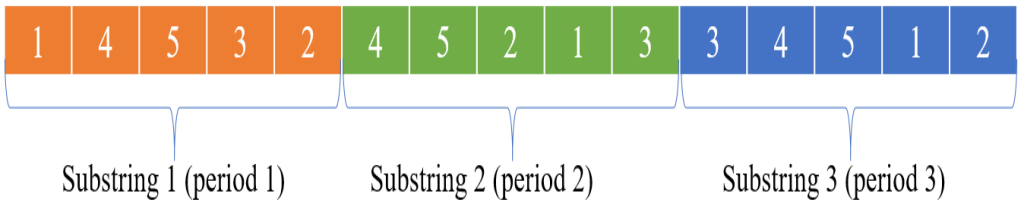
جدول شماره (۳): مقادیر نهایی برای پارامترهای الگوریتم‌ها

| الگوریتم | پارامتر | بهترین مقدار |
|---|---|---|
| GA | Maximum number of iteration=Maxit | ۲۰۰ |
| | Number of initial population=nPop | ۲۰۰ |
| | Percent of crossover=pc | ۰/۷ |
| | Percent of mutation=pm | ۰/۱ |
| SA | Maximum number of iteration=Maxit | ۳۰۰ |
| | Maximum number of sub-iteration=Sub-it | ۲۰ |
| | Temperature =T | ۱۲۰۰ |
| | Rate of reduction=R | ۰/۰۰۵ |
| TS | Maximum number of iteration=Maxit | ۴۰۰ |
| | Number of neighbour=NN | ۶ |
| | Tabu list=TL | ۴۰ |
| ICA | Maximum number of iteration=Maxit | ۳۰۰ |

| | | |
|---|---|---|
| | Number of initial population=nPop | ۲۰۰ |
| | Number of empires=Nemp | ۱۰ |
| | Assimilation rate=As | ۰/۶ |
| | Betta=b | ۵ |
| | Zetta=z | ۰/۱ |
| PSO | Maximum number of iteration=Maxit | ۳۰۰ |
| | Number of initial population=nPop | ۱۵۰ |
| | Individual learning=c1 | ۱/۲ |
| | Social learning=c2 | ۲ |
| | Inertia weight=w | ۱ |

پ) مقایسه روش‌ها

نخستین مرحله در حل مدل، ایجاد ارتباط میان مدل مساله با ساختار الگوریتم‌های فراابتکاری برای ایجاد یک راه ارتباطی میان مساله و فضای شدنی است، که در آن تکامل به وقوع می‌پیوندد. عملا، باید راهی به منظور نمایش کروموزوم‌های شدنی برگزیده شود. معمولا، کروموزوم‌های شدنی، در مسائل بهینه‌سازی ترکیبی (گسسته) می‌توانند رشته‌هایی از ارقام صحیح، در مساله‌های پیوسته برداری رشته‌هایی از ارقام حقیقی، در مساله‌های بولین رشته‌هایی از اعداد زوجی و چنانچه نیاز باشد، ترکیبی از نمایشها را در بربگیرند. لذا گزینش یک روش نمایش متناسب، یکی از با اهمیت‌ترین بخش‌های طراحی یک الگوریتم است. بیشتر الگوریتم‌های فراابتکاری از رویه‌های تصادفی به منظور ایجاد مجموعه جواب‌های اولیه استفاده می‌کنند. نحوه نمایش جواب در این مقاله در شکل (۷) آورده شده است. جدول(۳) حل مساله DFLPبا اندازه‌های مختلف را توسط پنج الگوریتم مذکور نشان می‌دهد. نتایج حاصله نشان از برتری الگوریتم رقابت استعماری و هوش تجمعی ذرات با افزایش اندازه مساله است. به صورت مفصل تر، الگوریتم ژنتیک و تبرید شبیه‌سازی شده تنها در مسائل یک و چهار به ترتیب به مقادیر بهتری دست پیدا کرده‌اند. در مجموع مسائل الگوریتم رقابت استعماری عملکرد مناسب‌تری داشته است. لازم به ذکر است هر الگوریتم با پارامترهای بهینه، ۳۰ مرتبه مورد اجرا قرار گرفته است.



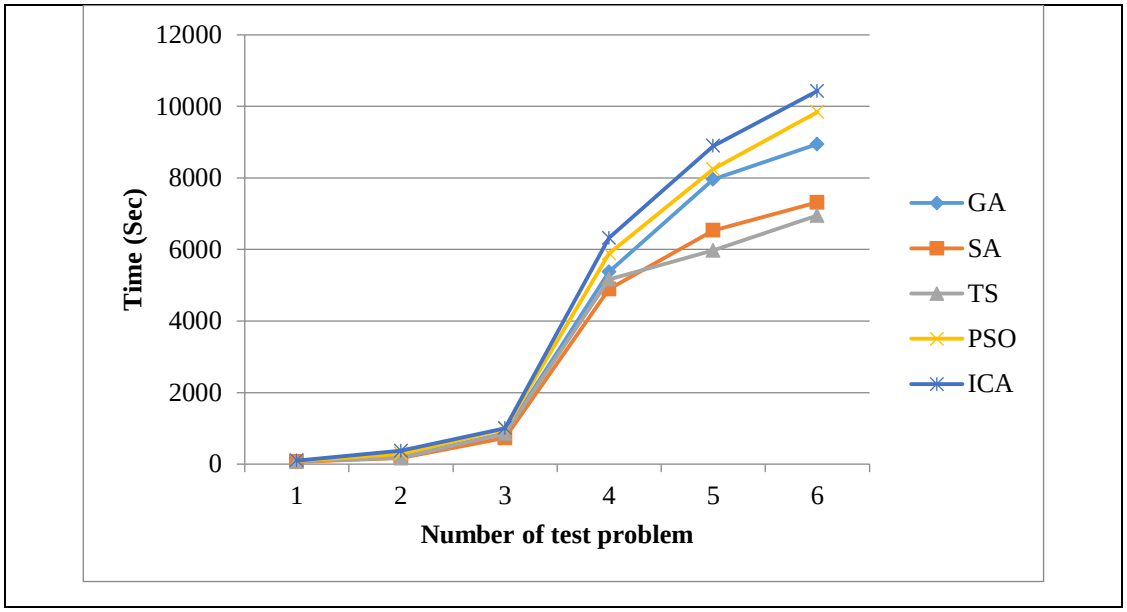

شکل شماره (۷): نمایش جواب الگوریتم‌ها برای ۵ جانمایی در سه دوره

جدول شماره (۴): نتایج محاسباتی برای مساله DFLP با سایزهای مختلف

| شماره مساله | الگوریتم GA | الگوریتم SA | الگوریتم TS | الگوریتم ICA | الگوریتم PSO |
|---|---|---|---|---|---|
| ۱ | ۱۰۵۶۴۲/۹ | ۱۱۵۰۲۲ | ۱۲۲۸۰۹/۶ | 113946 | ۱۱۰۹۱۸/۴ |
| ۲ | ۴۳۵۶۲۵/۳ | ۴۴۶۳۱۵ | ۴۴۴۲۸۴/۴ | ۴۳۴۰۷۲/۱ | ۴۴۴۲۵۱ |
| ۳ | ۴۵۹۴۰۶/۳ | ۴۵۸۷۳۶ | ۴۴۴۲۶۷/۴ | ۴۶۱۴۱۳/۱ | ۴۴۰۰۶۰/۹ |
| ۴ | ۷۶۹۸۵۹/۴ | ۷۶۴۵۳۱ | ۷۶۹۷۶۱/۳۳۳۳ | ۷۶۴۵۷۶/۱۴۸۱ | ۷۶۷۷۹۷/۵۱۸۵ |
| ۵ | ۹۰۴۸۵۳ | ۸۹۶۳۹۴ | ۹۱۲۶۳۳/۴ | ۸۶۸۴۵۰/۳ | ۸۹۲۸۴۵/۴۸۱۵ |

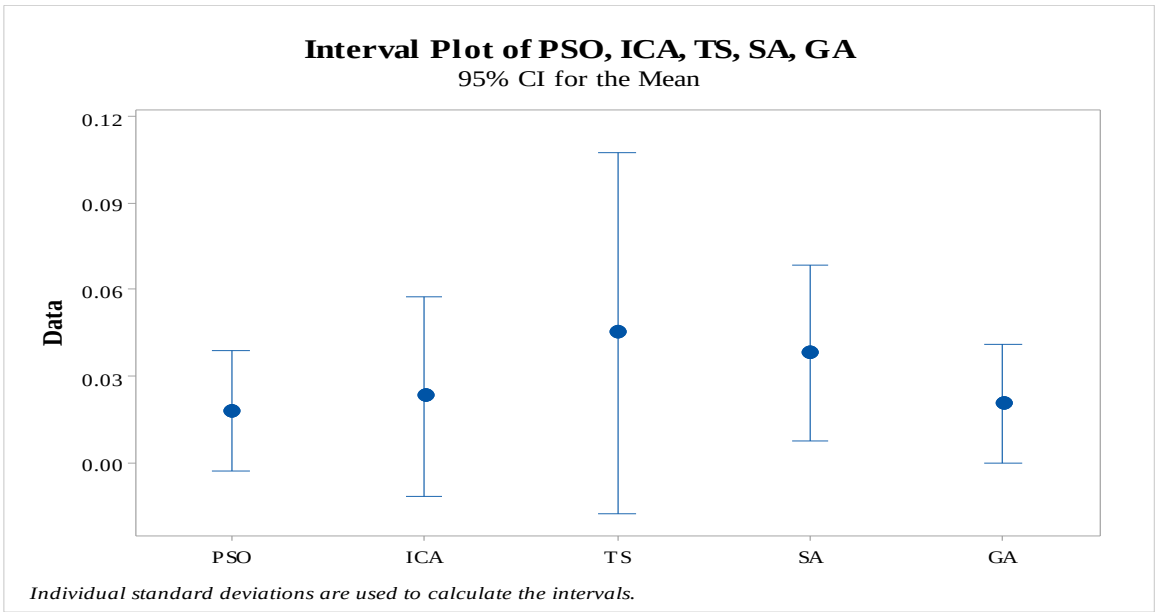
| | | | | | |
|---|---|---|---|---|---|
| ۶ | ۱۲۴۱۷۷۷ | ۱۲۵۳۰۰۷ | ۱۲۲۹۶۳۲ | ۱۲۲۲۰۱۵/۷۰۴ | ۱۲۱۰۱۹۵/۲۹۶ |
| میانگین | ۶۵۲۸۶۰/۶۵ | ۶۵۵۶۶۷/۵ | ۶۵۳۸۹۸/۰۲۲۲ | ۶۴۴۰۷۸/۸۹۲ | ۶۴۴۳۴۴/۷۶۶ |

شکل (۸) بیانگر رفتار زمان حل الگوریتم‌ها نسبت به سایز مساله می باشد. همان طور که می‌بینیم الگوریتم رقابت استعماری با افزایش اندازه مساله، نیاز به زمان بیشتری برای حل دارد.




شکل شماره (۸): زمان حل الگوریتم‌ها در مسائل مختلف طراحی شده

و درنهایت همان طور که در شکل (۹) مشخص است با استفاده از تحلیل واریانس، جواب الگوریتم‌ها را برای ۳۰ بار اجرا، بین سایز مساله و کیفیت جواب الگوریتم‌ها، به تصویر کشیده‌ایم. نتایج نشان دهنده اختلاف آشکار میان کیفیت جواب‌های الگوریتم است. بر این اساس، الگوریتم تجمعی ذرات نسبت به سایر روش ها میانگین کمتری دارد و الگوریتم جستجوی ممنوع انحراف معیار زیادی نسبت به سایر الگوریتم‌ها داراست.




شکل شماره (۹): کیفیت جواب الگوریتم‌ها

ت) نتیجه‌گیری و پیشنهادات آتی

در این مقاله، دو دسته از الگوریتم‌های فراابتکاری جمعیت محور و نقطه‌ای، شامل الگوریتم فراابتکاری ژنتیک، شبیه‌سازی تبرید، جستجوی ممنوع، ازدحام ذرات و رقابت استعماری برای حل DFLP بکار گرفته شده است. کیفیت پاسخ‌های ارائه شده توسط الگوریتم‌ها تا حد زیادی به تنظیم اولیه پارامترهای هر الگوریتم بستگی دارد. در این مقاله با استفاده از تکنیک تاگوچی، تنظیم پارامتر الگوریتم‌ها بگونه‌ای پیاده‌سازی شد که هر یک از الگوریتم‌ها در بهترین شرایط خود، از منظر هم‌خوانی پارامترها، اجرا شوند و بهترین پاسخ را با توجه به محدودیتهایشان ارائه کنند. بعد از مقایسه جواب‌های بدست آمده از اجرای هر یک از پنج الگوریتم متوجه شدیم که الگوریتم‌های جمعیت محور در مجموع جواب‌های بهینه بهتر، با زمان حل قابل قبول‌تری را ارائه می‌دهند و دارای عملکرد مناسبی می‌باشند. در ابتدا برای مسائل کوچک الگوریتم‌های موجود همگی پاسخ‌های قابل قبولی را ارائه می‌دادند، اما با افزایش فضای جستجو به مرور اختلاف کارایی الگوریتم‌ها و برتری نسبی الگوریتم‌های جمعیت محور نسبت به دیگر الگوریتم‌ها مشخص شد، اگر چه کارایی الگوریتم‌ها زمانی‌که اندازه مسائل بزرگتر شد، کاهش یافت؛ اما مشخص شد که بین الگوریتم های بکار گرفته شده برای حل مدل مفروض، الگوریتم رقابت استعماری می‌تواند این افت را بهتر از الگوریتم‌های دیگر کاهش دهد و به‌طور استوارتری عمل کند، به عبارت دیگر، با افزایش اندازه مساله و فضای جستجو، کارایی الگوریتم بیشتر نمایان خواهد شد. همچنین با گرفتن میانگین از پاسخ‌های ارائه شده توسط هر یک از الگوریتم‌های پنج گانه نتیجه مشابهی در برتری الگوریتم رقابت استعماری برای حل مدل DFLP حاصل شد. از پیشنهادات زیر می‌توان برای مطالعات آینده استفاده نمود:

- توسعه تکنیکی ترکیبی برای حل DFLP، که ترکیب دو یا چند متاهیوریستیک را در نظر بگیرد.
- هزینه بازآرایی بعنوان یک تابع هدف جداگانه در نظر گرفته شود؛ بنابراین موردی چند هدفه برای مساله طرح تاسیسات پویا (MODFLP) را می‌توان مدل سازی و حل کرد.
- طراحی الگوریتمی با تعداد پارامترهای کمتر.
- ارائه یک راه‌حل جدید برای تولید همسایگی و یا اپراتورهای جهش بطوریکه بتواند به منظور بهبود اثربخشی، از نظر کیفیت راه‌حل و زمان محاسباتی، استفاده شود.

## ۴- منابع

1. Armour, G. C. (1963). A heuristic algorithm and simulation approach to relative location of facilities. *Management Science* , 9(2), 294-309.
2. Balakrishnan J, &. C. (2000). Genetic search and the dynamic layout problem. *Computers & Operations Research* , 27(6):587–593.
3. Balakrishnan, J. C. (2000). An improved pair-wise exchange heuristic for the dynamic plant layout problem. *International Journal of Production Research* , 38(13), 3067-3077.
4. Balakrishnan, J. C. (2003). A hybrid genetic algorithm for the dynamic plant layout problem. *International Journal of Production Economics* , 86(2), 107-120.
5. BaykasogluA, &. G. (2001). A simulated annealing algorithm for dynamic facility layout problem. . *Computers & Operations Research* , 28(14):1403–26.
6. Burkard RE, &. B. (1983). A heuristic for quadratic Boolean problems with application to quadratic assignment problem. *European Journal of Operational Research* , 13:374–86.
7. Conway, D. G. (1994). Genetic search and the dynamic facility layout problem. *Computers & Operations Research* , 21(8), 955-960.
8. Erel E, G. J. (2003). New heuristic for the dynamic layout problem. . *Journal of the Operational Research Society* , 54:1275–82.
9. Gomory RE, &. H. (1961). Multi-terminal network flows. *SIAM Journal* , 9:551–70.
10. Hajiaghaei-Keshteli, M. &. (2010). Deriving the cost function for a class of three-echelon inventory system with N-retailers and one-for-one ordering policy. *The International Journal of Advanced Manufacturing Technology* , 50(1-4), 343.
11. Hajiaghaei-Keshteli, M. S. (2011). Determination of the economical policy of a three-echelon inventory system with (R, Q) ordering policy and information sharing. *The International Journal of Advanced Manufacturing Technology* , 55(5-8), 831-841.
12. Kaku, B. K. (1997). A tabu-search heuristic for the dynamic plant layout problem. *INFORMS Journal on Computing* , 9(4), 374-384.
13. Lacksonen TA, &. E. (1993). Quadratic assignment algorithms for the dynamic layout problem. *International Journal of Production Research* , 31(3):503–17.
14. McKendall, A. R. (2006). Hybrid ant systems for the dynamic facility layout problem. *Computers & Operations Research* , 33(3), 790-803. .
15. Pardalos PH, &. C. (1989). Proceedings of the 1989 supercomputer conference. In *A parallel algorithm for the QAP*. (pp. p. 351–60.). New York: ACM Press.
16. Pourvaziri, H. &. (2014). A hybrid multi-population genetic algorithm for the dynamic facility layout problem. *Applied Soft Computing* , 24, 457-469.
17. Rosenblatt, M. J. (1986). The dynamics of plant layout. *Management Science* , 32(1), 76-86.
18. Tompkins, J. A. (1996). *Facilities Planning.* 2 nd edn(New York: Wiley).
19. Urban, T. (1993). A heuristic for the dynamic facility layout problem. *IIE Transactions* , 25(4):57–63.