فصلنامه علمی – پژوهشی مطالعات مدیریت صنعتی – سال دوازدهم، شماره ۳۲، بهار ۹۳
صفحات ۶۶ – ۴۹

ترکیب الگوریتم بهینه‌سازی ازدحام ذرات و تکنیک شبیه‌سازی در حل مسئله جانمایی پویای تسهیلات

پرهام عظیمی * – **محمد علی زارع شوریجه** **

(تاریخ دریافت : ۹۰/۱۲/۲ – تاریخ پذیرش : ۹۱/۴/۲۱)

چکیده

امروزه یکی از ضروریات رویارویی با تغییرات مستمر در بازار مصرفی، استفاده از طرح جانمایی پویای تسهیلات است. مسئله جانمایی پویای تسهیلات [۱] (DFLP)، مسئله‌ای بهینه‌سازی است که در جستجوی مناسب‌ترین موقعیت تسهیلات تولیدی برای افقی چنددوره‌ای است، به‌گونه‌ای که مجموع کل هزینه‌های طرح جانمایی کمینه گردد. این پژوهش، برای نخستین بار، رویکرد شبیه‌سازی و الگوریتم فراابتکاری بهینه‌سازی ازدحام ذرات [۲] (PSO) را برای حل DFLP ترکیب کرده و با اعمال اصلاحاتی در الگوریتم ازدحام ذرات در فضای صفر و یک [۳] (BPSO)، الگوریتمی جدید به نام الگوریتم بهینه‌سازی ازدحام ذرات گمارش شده [٤] (APSOA) را ارائه داده‌است، که نتیجه آن اثربخشی بیشتر PSO در حل DFLP است. نتایج به‌دست‌آمده حاصل از آزمون APSOA روی مسائل مختلف DFLP با اندازه‌های کوچک، متوسط و بزرگ، حکایت از عملکرد بسیار خوب آن دارد.

کلیدواژه‌ها: جانمایی تسهیلات، مسئله جانمایی پویای تسهیلات (DFLP)، الگوریتم بهینه‌سازی ازدحام ذرات گمارش شده (APSOA)، شبیه‌سازی.

---

*نویسنده مسئول، استادیار، دانشکده مهندسی صنایع و مکانیک، واحد قزوین، دانشگاه آزاد اسلامی، قزوین، ایران.
p.azimi@qiau.ac.ir
**کارشناس ارشد مدیریت صنعتی گرایش تحقیق در عملیات، دانشگاه علامه طباطبائی، تهران، ایران.

1. Dynamic facility layout problem
2. Particle swarm optimization
3. Binary particle swarm optimization
4. Assigned particle swarm optimization algorithm

مقدمه

در اقتصاد امروز جهان، کارخانجات تولیدی می‌بایست قادر به پاسخ‌گویی سریع به تغییرات در تقاضا و سبد محصول باشند. این تغییرات موجب تغییر در جریان مواد بین ماشین‌ها، بخش‌ها[1] و تسهیلات[2] مختلف تولیدی می‌گردد. تغییر در جریان مواد می‌تواند تغییر درآرایش و جانمایی تسهیلات را به مدیران یا مهندسان تولید تحمیل کند (مک کندال، شانگ، و کاپوسامی، ۲۰۰۶). آن‌ها تقاضا و جریان مواد در دوره‌های آینده را پیش‌بینی می‌کنند. با استفاده از نتایج این پیش‌بینی‌ها می‌توان طرح‌های مختلف جانمایی تا چند دوره دیگر را نیز برنامه‌ریزی نمود. طرح جانمایی انتخاب شده برای یک شرکت، روابط بین فعالیت‌هایی را که همراه با انتقال مواد هستند مشخص می‌کند. بنابراین جانمایی تسهیلات و فعالیت‌های مربوط به جابه‌جایی مواد کاملاً با هم مرتبط‌اند و مستقیماً بر یکدیگر اثر می‌گذارند و هرگز نمی‌توان یکی از این دو را بدون درنظرگرفتن دیگری مورد تحقیق و بررسی قرار داد(دریرا، پیروال، و ونکاتارامانان، ۱۹۹۴).

رُزِنبِلات[3] (۱۹۸۶) اولین کسی بود که از الگوریتم برنامه‌ریزی پویا[4] (DP) برای حل بهینه DFLP استفاده کرد. بالاکریشنان، جیکوبز، و ونکاتارامانان (۱۹۹۲) یک محدودیت بودجه برای هزینه‌های بازآرایی تسهیلات درنظر می‌گیرد. لکسُنِن و انسکور[5](۱۹۹۳) سه الگوریتم صفحات برشی، شاخه و کران و برنامه‌ریزی پویا را برای حل DFLP پیشنهاد می‌کنند. اِربن[6] (۱۹۹۳) از معاوضه دوبه‌دو[7] و بالاکریشنان، چنگ و کانوی[8] (۲۰۰۰) نیز از الگوریتمی مرکب مرکب از معاوضه دوبه‌دو بهبودیافته و DP برای حل DFLP بهره می‌گیرند. کونوای و ونکاتارامانان (۱۹۹۴) و همچنین و کالو و مازولا (۱۹۹۷) به‌ترتیب از روش جستجوی ژنتیک و جستجوی ممنوع[9] (TS) برای DFLP استفاده می‌کنند. بالاکریشنان و چنگ (۲۰۰۰) الگوریتم ژنتیک[10] (GA) بهبودیافته‌ای با حلقهٔ تودرتو[11] ارائه می‌کنند. بالاکریشنان و همکاران

1. Department
2. Facilities
3. Rosenblatt
4. Dynamic programming
5. Lacksonen & Enscore
6. Urban
7. Pairwise exchange
8. Balakrishnan, Cheng & Conway
9. Tabu search
10. Genetic Algorithm
11. Nested loop

(۲۰۰۳) نیز الگوریتم ژنتیک مرکبی ارائه می‌کنند که از DP برای تقاطع، ازCRAF[1] برای ایجاد جهش و از روش اِربن برای ایجاد جمعیت اولیه استفاده می‌نمایند. مک‌کندال و هاکوبیان[2] (۲۰۱۰) و بایکاسگلو و گیندی[3] (۲۰۰۱) به‌ترتیب از الگوریتم جستجوی ممنوع و شبیه‌سازی تبرید[4] (SA) بهره می‌گیرند. مک‌کندال، شانگ و کاپوسامی (۲۰۰۶) نیز از تنظیمات و مفروضات دیگری برای کاربرد SA بهره می‌گیرد. آنان SA-I و SA-II را ارائه می‌کنند. SA-I از رویکرد سازگاری مستقیم و SA-II از رویکردی مانند SA-I همراه با استراتژی نگاه به جلو و نگاه به عقب[5] استفاده می‌کنند به‌طوری‌که نتایج این تحقیق در حال حاضر ارائه‌دهنده بهترین جواب‌های DFLP تاکنون است. دریکی از آخرین مطالعات انجام گرفته در حوزه ادبیات DFLP، در سال ۲۰۱۱، از رویکرد استوار[6] در حل مسائل DFLP استفاده می‌شود (پیلای، هوناگوند، و کریشنان، ۲۰۱۱). مک‌کندال و شانگ[7] (۲۰۰۶) نیز اولین بار و پس از آن بیکازوگلو، درلی و سابونچو(۲۰۰۶) و نیز نینگ، لام (۲۰۱۰) از بهینه سازی کلونی مورچگان[8] (ACO) برای حل DFLP کمک می‌گیرند.

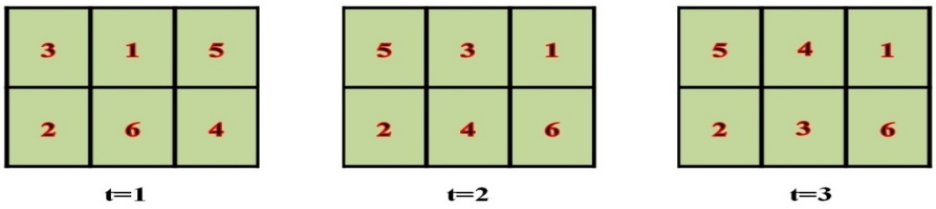
یکی از روش‌های تکاملی و فراابتکاری جدید، الگوریتم بهینه‌سازی ازدحام ذرات[9] (PSO) می‌باشد که در حل بسیاری از مسائل *NP-Hard* و پیچیده استفاده شده‌است (اوربان، ۱۹۹۳). دراین تحقیق سعی می‌شود برای اولین بار مسئله DFLP با ترکیب دو رویکرد شبیه‌سازی[10] و الگوریتم PSO حل شود. در ادامه این مقاله به‌ترتیب به بیان مسئله، تشریح الگوریتم PSO و الگوریتم پیشنهادی، یافته‌های حاصل از آزمون آن، نتیجه‌گیری و پیشنهادات پرداخته می‌شود.

بیان مسئله

جانمایی تسهیلات، آرایشی است از هر چیز مورد نیاز برای تولید کالا یا ارائه خدمات. تسهیل (تسهیلات) موجودیتی است که انجام کاری را امکان‌پذیر یا آسان‌تر می‌کند، که می‌تواند یک ماشین، ابزار، ایستگاه کاری، سلول تولیدی، بخش، انبار و غیره باشد(دریرا، پیروال و هاجری

1. Computerized Relative Allocation of Facilities Technique
2. McKendall & Hakobyan
3. Baykasoğlu & Gindy
4. Simulated annealing
5. Look-ahead/look-back strategy
6. Robust approach
7. McKendall & Shang
8. Ant colony optimization
9. Particle Swarm Optimization
10. Simulation

گابوج، ۲۰۰۷). شایان ذکر است که در این مقاله ماشین‌آلات تولیدی به‌عنوان تسهیلات درنظر گرفته می‌شوند. درصورتی‌که این مسئله تنها برای یک دوره تولید مورد بررسی قرار گیرد آن را «مسئله جانمایی ایستای تسهیلات[1]» (SFLP) می‌نامند. زمانی‌که جریان مواد بین ماشین‌ها، درطی یک افق برنامه‌ریزی تغییر می‌کند، این مسئله تحت‌عنوان «مسئله جانمایی پویای تسهیلات[2]» (DFLP) شناخته می‌شود. افق برنامه‌ریزی به تعدادی دوره که می‌توانند هفته، ماه یا سال باشند، تقسیم می‌شود و جریان داده‌های مربوط به هردوره پیش‌بینی می‌شود و فرض می‌گردد که جریان داده‌ها در طول هردوره ثابت می‌ماند (شکل(۱)). در این حالت برای هر دوره یک طرح جانمایی مناسب پیشنهاد می‌گردد. هدف یک مسئله جانمایی پویا می‌تواند تعیین یک طرح جانمایی برای هر دوره در افق برنامه‌ریزی باشد، طوری‌که مجموع هزینه‌های جابه‌جایی مواد برای تمام دوره‌ها و نیز مجموع هزینه‌های چینش یا آرایش مجدد[3] تسهیلات بین دوره‌های مختلف افق برنامه‌ریزی حداقل گردد. یک طرح جانمایی خوب می‌تواند تا ۵۰٪ از هزینه‌های عملیاتی را بکاهد. برای تسهیلات تولیدی، هزینه جابه‌جایی مواد[4] مهمترین شاخص برای تعیین اثربخشی یک جانمایی است زیرا این هزینه ۵۰-۲۰ درصد کل هزینه‌های عملیاتی و ۷۵-۱۵ درصد کل هزینه‌های تولید یک محصول را شامل می‌شود (تامپکینزو همکاران، ۱۹۹۶).



شکل (۱): یک طرح جانمایی پویا با سه دوره و شش مکان

مسائل جانمایی، به جز نمونه‌های کوچک آن، اغلب در زمره مسائل *NP-hard* وپیچیده هستند(دریرا، پیروال و هاجری گابوج، ۲۰۰۷). پارامترهای این مدل به ترتیب شامل N تعداد ماشین‌ها، T تعداد دوره‌ها، A_{tijl} هزینه انتقال ماشین i از مکان j به مکان l در دوره t (به طوریکه $A_{tiil}=0$) و C_{tijkl} هزینه جابه‌جایی مواد در دوره t بین ماشین i قرار گرفته در مکان j و ماشین k قرار گرفته در مکان l است.

1. Static Facility Layout Problem
2. Dynamic Facility Layout Problem
3. Rearrangement
4. Material Handling Cost

$$\min \quad \sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N} A_{tijl}Y_{tijl} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} C_{tijkl}X_{tij}X_{tkl} \qquad (۱)$$

S.t.

$$\sum_{j=1}^{N} X_{tij} = 1 \qquad i = 1, \dots, N \qquad t = 1, \dots, T \qquad (۲)$$

$$\sum_{i=1}^{N} X_{tij} = 1 \qquad j = 1, \dots, N \qquad t = 1, \dots, T \qquad (۳)$$

$$Y_{tijl} = X_{(t-1)ij}X_{til} \qquad i, j, l = 1, \dots, N \qquad t = 2, \dots, T \qquad (۴)$$

$$X_{tij} \in \{0,1\} \qquad i, j = 1, \dots, N \qquad t = 1, \dots, T \qquad (۵)$$

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

متغیرهای تصمیم این مسئله از نوع صفر و یک و تابع هدف مسئله غیرخطی است. اگر ماشین i در دوره t به مکان jاختصاص یابد مقدار متغیر Xtij برابر ۱ و درغیراین‌صورت صفر خواهد بود. همچنین اگر در آغاز دوره t، ماشین i از مکان j به مکان l منتقل شود، مقدار متغیر Ytijl برابر ۱ و درغیراین‌صورت صفر خواهد بود. مجموعه محدودیت اول و دوم به ترتیب تضمین می‌کنند که هر مکان تنها به یک ماشین و هر ماشین تنها به یک مکان در هر دوره تخصیص یابد. محدودیت (۴) نشان‌دهنده تغییر مکان یک ماشین از یک دوره به دوره بعد است که می‌توان متغیر Ytijl در گزاره نخست تابع هدف را با مقادیر نظیر آن در محدودیت (۴) جایگزین نمود و این محدودیت را حذف نمود.

الگوریتم بهینه‌سازی ازدحام ذرات (PSO) و الگوریتم پیشنهادی

الگوریتم PSO یک تکنیک بهینه‌سازی تصادفی الهام گرفته شده از رفتار گله پرندگان و ماهی‌ها است (کندی و ابرهارت، ۱۹۹۵). در اصطلاح این اجتماعات جانوری را ازدحام[1] و

1. Swarm

هریک از اعضای این اجتماعات جانوری را ذره[1] گویند (تالبی، ۲۰۰۹). یک ازدحام شامل M ذره می‌شود که در یک فضای D بعدی در حال پروازند. موقعیت هر ذره i یک جواب کاندید برای مسئله در تکرار tام الگوریتم است که با بردار xit=(xi1t, xi2t,…, xiDt) در فضای جواب نشان داده می‌شود. هر ذره موقعیت خود، xit را براساس دو عامل، به سوی بهینه کلی تغییر می‌دهد: بهترین موقعیت خود تا تکرار فعلی، pit=(pi1t,pi2t,…,piDt) و بهترین موقعیتی که تاکنون به وسیله یکی از ذرات کل ازدحام بدست‌آمده‌است، pgt=(pg1t,pg2t,…,pgDt). vit=(vi1t,vi2t,…,viDt) بردار سرعت ذره i در تکرار tام الگوریتم و pbesti وgbest مقادیر تابع هدف (یا همان برازش[2]) بردارهای pit وpgt را نشان می‌دهند (کاکو و مازولا، ۱۹۹۷)، (لکسونن و انسکور، ۱۹۹۳) و (نینگ، لام و لام ، ۲۰۱۰). در تکرار tام موقعیت و سرعت ذره iام طبق روابط زیر به‌روزرسانی می‌شوند:

$$x_{id}^{t} = x_{id}^{t-1} + v_{id}^{t} \qquad i = 1,2,\dots,M \text{ and } d = 1,2,\dots,D \qquad (۷)$$

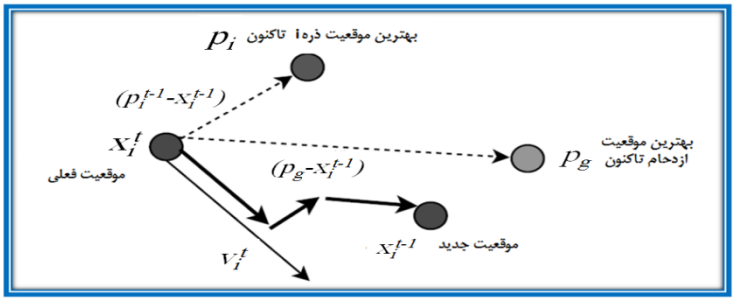
$$v_{id}^{t} = wv_{id}^{t-1} + c_1 r_1 (p_{id}^{t-1} + x_{id}^{t-1}) + c_2 r_2 (p_{gd} + x_{id}^{t-1}) \qquad (۸)$$

$$i = 1,2,\dots,M \text{ and } d = 1,2,\dots,D$$

این روابط برای تمام D بعد هر ذره بایستی بدست‌آیند. در رابطه (۸) c1 و c2 دو عدد مثبت حقیقی‌اند که ثابت‌های شتاب[3] یا فاکتورهای یادگیری[4] نامیده می‌شوند. این دو مقدار ثابت به‌ترتیب نشان‌دهنده تمایل یک ذره به سوی موقعیت pbest خود و gbest ازدحام می‌باشد. پارامترهای r1 وr2 تضمین می‌کنند که الگوریتم تصادفی است، زیرا این دو پارامتر، اعدادی مستقل و تصادفی در بازه [0,1] هستند و برای تنوع‌بخشی به جمعیت مورد استفاده قرار می‌گیرند (کندی و ابهارت، ۱۹۹۵)، (اولاریو، زومایا، ۲۰۰۶) و (رکه-لوپز و همکاران، ۲۰۰۹). w وزن اینرسی[5] نامیده می‌شود که نشان‌دهنده تأثیر سرعت پیشین ذره بر سرعت فعلی آن است و معمولاً در فاصله صفر و یک اختیار می‌شود و در حقیقت موازنه‌ای بین تنوع جستجو[6] و تشدید جستجو[7] برقرار می‌کند (تالبی، ۲۰۰۹). هر چه w بزرگتر باشد تنوع جستجو

1. Particle
2. Fitness
3. Acceleration constants
4. Learning factors
5. Inertia weight
6. Global exploration
7. Exploitation

بیشتر می‌شود و برعکس (تالبی، ۲۰۰۹). شکل (۲) نحوه به‌روزرسانی موقعیت ذره نمونه i را نشان می‌دهد.



شکل (۲): به‌روزرسانی سرعت و موقعیت ذره در PSO

برای ممانعت از رشد بسیار شدید سرعت و عدم‌خروج ذرات از فضای جستجو، هر یک از مؤلفه های بردار سرعت محدود به بازه $[v_{min},v_{max}]$ می‌شوند. معمولاً $v_{min}=-v_{max}$ و v_{max} را برابر ۴ درنظر می‌گیرند (کندی، ۱۹۹۷). درصورتی که مقدار تابع هدف جواب بدست‌آمده (ذره) در تکرار فعلی بهتر از مقدار *pbest* ذره در تکرار قبلی باشد *pbest* و بردار متناظر آن یعنی p_i^t وبه همین ترتیب اگر *pbest* جدید ذره‌ای بهتر از *gbest* فعلی باشد، *gbest* و بردار متناظر آن یعنی p_g به‌روز رسانی می‌گردند. برای استفاده PSO در فضای گسسته، مقدار سرعت بدست‌آمده از رابطه (۸)، در رابطه زیر قرار می‌گیرد و به احتمالی بین ۰ و ۱ تبدیل می‌شود(کندی و ابرهارت، ۱۹۹۷):

$$Sig(v_{id}^t)=\frac{1}{(1+e^{(v_{id}^t)})} \qquad (۹)$$

سپس هر بعد به‌دست‌آمده از رابطه (۹) طبق رابطه (۱۰) یک بعد (یا یک بیت[1]) از بردار موقعیت جدید، xidt را می‌سازد (ρ_{id} عددی تصادفی در بازه [0,1] است) (لازینکا، ۲۰۰۹):

$$x_{id}^t=\begin{cases}1 & \rho_{id}<Sig(v_{id}^t)\\ 0 & otherwise\end{cases} \qquad (۱۰)$$

1. Bit

در این تحقیق تعداد ذرات جمعیت نخستین، M، ۲۰ ذره است. از طرفی واضح است که تعداد جواب‌ها یا طرح‌های جانمایی ممکن برای یک مسئله با N ماشین و T دوره (N!)T می‌باشد. بنابراین برای تولید یک جمعیت تصادفی نخستین، به کمک نرم‌افزار شبیه‌سازی پرسرعت ED [1] 8.0. ، یک کارخانه فرضی با N ماشین و T دوره را مدل‌سازی و ۱۰۰۰ جواب ممکن به صورت تصادفی تولید می‌گردد و با محاسبه مقدار تابع هدف این جواب‌ها در نهایت، ۲۰ جواب برتر به عنوان جمعیت اولیه وارد الگوریتم PSO می‌شوند. این روش موجب به‌دست‌آوردن حد بالای مناسبی برای ورود به مرحله استفاده از PSO خواهد شد.

جواب‌های موجه یک مسئله DFLP ترکیب بسیار خاصی از اعداد صفر و یک هستند به گونه‌ای که هر مکان تنها قابل تخصیص به یک ماشین و هر ماشین تنها قابل تخصیص به یک مکان است. یک مسئله DFLP با T دوره و N ماشین (با توجه به برنامه‌ریزی ریاضی آن که قبلاً ارائه گردید) دارای 2T.N محدودیت و T.N2 متغیر تصمیم (و به عبارت دیگر T.N2 بُعد یا مؤلفه در هر جواب) می‌باشد که تنها TN متغیر از این تعداد اجازه دارند طبق محدودیت‌های سخت تحمیلی ذکر شده در بالا، مقدار ۱ را اختیار کنند و مابقی صفر هستند. بنابراین برای تولید هر جواب در عین حالی که می‌بایست الگوریتم ماهیت تصادفی خود را حفظ کند، نباید از محدودیت‌های سخت تحمیلی اشاره شده بر جواب‌ها نیز تخطئ کند. برای سازگار نمودن الگوریتم PSO صفر و یک با DFLP، اصلاحاتی بر مدل PSO صفر و یک موجود در ادبیات پیشنهاد می‌گردد. مدل PSO اصلاح‌شده در این تحقیق، از سوی محققان، الگوریتم بهینه‌سازی ازدحام ذرات گمارش شده (APSOA) نام نهاده می‌شود. APSOA از تلفیق مفاهیم جستجوی تصادفی PSO و مفاهیمی مشابه مفاهیم سنتی تخصیص استفاده می‌کند.

در APSOA برای تعیین هر مؤلفه از بردار موقعیت ذره، به جای استفاده از رابطه (۱۰)، از مفاهیمی مشابه مفاهیم سنتی تخصیص استفاده می‌گردد. در ادامه با ارائه یک مثال نحوه عملکرد این مدل تشریح می‌گردد. ب‌عنوان مثال، مسئله‌ای با ۴ ماشین و ۲ دوره مفروض است، دراین‌صورت، تعداد کل ابعاد هر جواب موجه این مسئله ۲(۴)۲ یعنی ۳۲ است. فرض کنید مقدار خروجی این ۳۲ مؤلفه برای تابع Sig(vidt) به صورت ماتریس‌های زیر می‌باشد. $Sig\left((v_{id}^{t}):,:,n\right)$, n=1, 2, …,T همان مقادیر خروجی Sig(vit) در دوره n است.

1. Enterprise Dynamic

$$\mathrm{Sig}\left((v_{id}^{t}):,:,1\right)=\begin{bmatrix}0.1734 & 0.0605 & 0.6569 & 0.0155\\ 0.3909 & 0.3993 & 0.6280 & 0.9841\\ 0.8314 & 0.5269 & 0.2920 & 0.1672\\ 0.8034 & 0.4168 & 0.4317 & 0.1062\end{bmatrix} \quad (۱۱)$$

$$\mathrm{Sig}\left((v_{id}^{t}):,:,2\right)=\begin{bmatrix}0.3724 & 0.9516 & 0.2691 & 0.4177\\ 0.1981 & 0.9203 & 0.4228 & 0.9831\\ 0.4897 & 0.0527 & 0.5479 & 0.3015\\ 0.3395 & 0.7379 & 0.9427 & 0.7011\end{bmatrix} \quad (۱۲)$$

حال به‌گونه‌ای می‌بایست این احتمالات به اعداد صفر و یک تبدیل گردند که اولاً از محدودیت‌های DFLP برای تولید جواب موجه تخطئ نکنند و دیگر آن‌که، مؤلفه‌هایی که بزرگترند و به عبارت دیگر از سرعت بیشتری برخوردار بوده‌اند با احتمال بیشتری مقدار ۱ را اختیار کنند. بدین منظور در هر ماتریس (یعنی در خروجی Sig(vit) مربوط به هر دوره) ابتدا بزرگترین مقدار احتمال را یافته و به جای آن مقدار ۱ قرار داده می‌شود و بقیه عناصر سطر و ستون آن مؤلفه صفر می‌شوند. سپس از بین مؤلفه‌های بدون تغییر باقیمانده ماتریس به‌روزشده جدید (یعنی ماتریسی که یکی از مؤلفه‌های آن ۱ و برخی صفر شده‌اند و برخی نیز هنوز تغییر نکرده‌اند) بار دیگر بزرگترین مقدار انتخاب و فرایند فوق بر روی آن اجرا می‌شود.

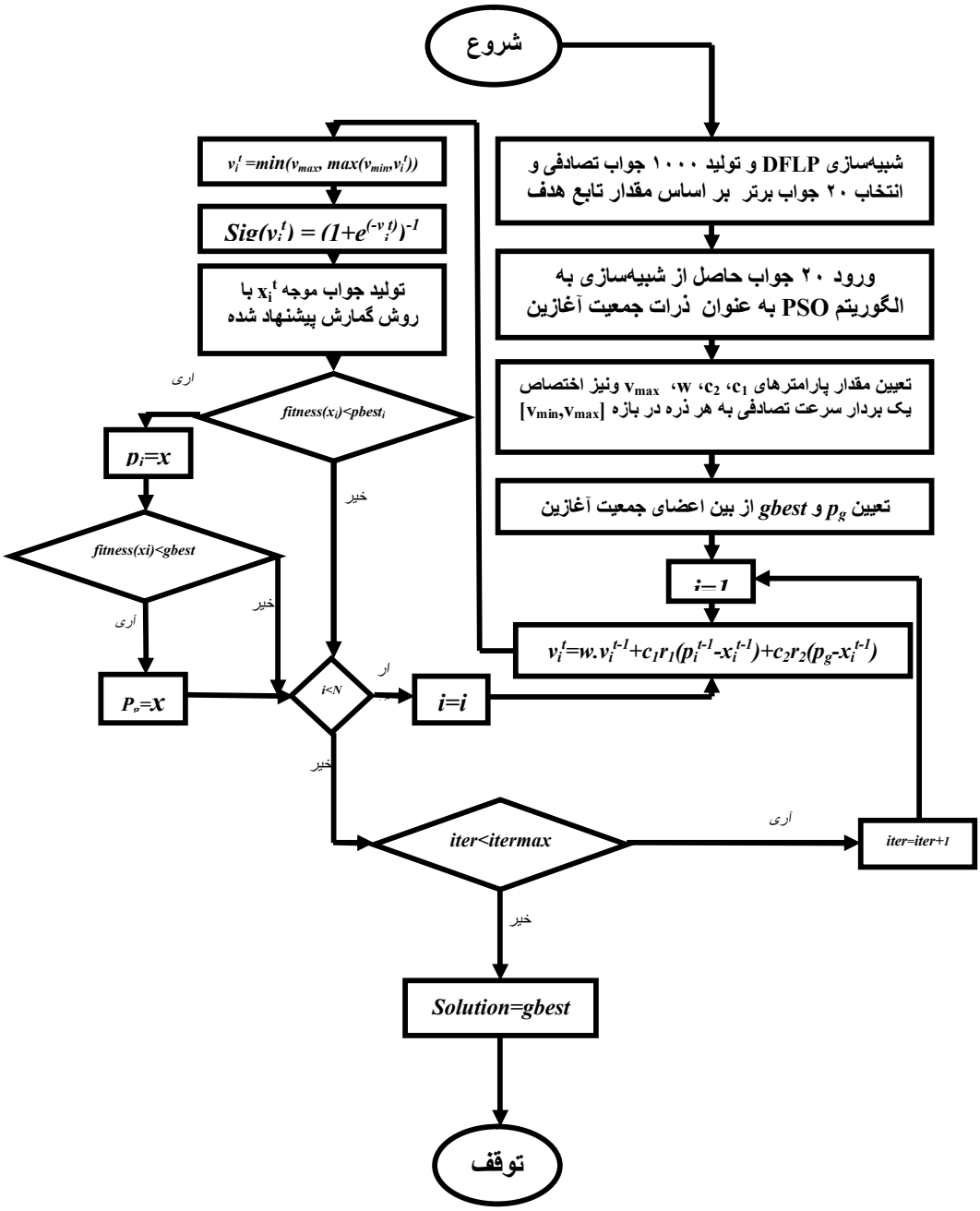
$$\mathrm{Sig}\left((v_{id}^{t}):,:,1\right)=\begin{bmatrix}0.1734 & 0.0605 & 0.6569 & 0.0155\\ 0.3909 & 0.3993 & 0.6280 & \mathbf{0.9841}\\ 0.8314 & 0.5269 & 0.2920 & 0.1672\\ 0.8034 & 0.4168 & 0.4317 & 0.1062\end{bmatrix}$$

$$\mathrm{Sig}\left((v_{id}^{t}):,:,1\right)=\begin{bmatrix}0.1734 & 0.0605 & 0.6569 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}\\ 0.8314 & 0.5269 & 0.2920 & \mathbf{0}\\ 0.8034 & 0.4168 & 0.4317 & \mathbf{0}\end{bmatrix} \quad (۱۳)$$

$$\mathrm{Sig}\left((v_{id}^{t}):,:,1\right)=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix} \quad (۱۴)$$

این عمل آن‌قدر تکرار می‌شود تا تمامی عناصر Sig(vidt) در تمام دوره‌ها با توجه به مقدارهای احتمالی اولیه آن‌ها به مقدارهای صفر و یک تبدیل گردند. عدد ۱ موجود در سطر i و ستون j این

ماتریس (۱۴) نشان‌دهنده تخصیص ماشین موجود در سطر iام به مکان jام در دوره ۱ است. ابتکار صورت گرفته در این تحقیق برای سازگار نمودن PSO صفر و یک و مسئله DFLP بسیار ضروری و مفید است. از مزایای مدل پیشنهادی APSOA این است که کلیه جواب‌های تولیدی به این روش موجه خواهند بود. شایان ذکر است، در صورت عدم استفاده از این روش زمان فوق‌العاده زیادی (گاه چندین هفته) برای تولید یک جواب موجه مورد نیاز است. گام‌های الگوریتم APSOA پیشنهادی عبارتند از: گام (۱) شروع الگوریتم با تولید ذرات جمعیت آغازین با استفاده از شبیه‌سازی؛ گام (۲) تعیین پارامترهای c1، c2، w و Vmax ؛ گام (۳) انتخاب pbest برای هر ذره و سپس gbest ازدحام؛ گام (۴) به‌روزرسانی بردار سرعت ذره با استفاده از رابطه (۸)؛ گام (۵) به‌دست‌آوردن هر بعد از ماتریس احتمالات با استفاده از رابطه (۹)؛ گام (۶) استفاده از مدل پیشنهادی برای تولید جواب موجه xi ؛ گام (۷) محاسبه مقدار تابع هدف xi و به‌روزرسانی pbest و gbest در صورت نیاز؛ گام (۸) اگر شاخص توقف محقق نگردید، رفتن به گام (۴) و در غیراین‌صورت رفتن به گام (۹) ؛ گام (۹) نمایش بهترین جواب بدست آمده (شکل (۳)). شایان ذکر درالگوریتم پیشنهادی برای رهایی از بند نقاط بهینه محلی، بعد از هر ۷۰ تکرار، با حفظ بهترین جواب به‌دست‌آمده تا آن تکرار (آخرین gbest)، با یک اجرای مجدد شبیه‌سازی، ۱۹ ذره جدید جایگزین ذرات جمعیت پیشین می‌شود که با بهترین ذره کنونی از جمعیت قبلی (pg) جمعاً ۲۰ ذره جدید را تشکیل می‌دهند. این روش، تأثیر بسیار جدی بر کیفیت جواب‌های به‌دست‌آمده دارد. در این تحقیق پارامترهای الگوریتم PSO، یعنی c1، c2، w و Vmax به ترتیب ۱، ۰/۵، ۰/۶ و ۴ درنظر گرفته شده‌اند. از آنجا که جواب‌های نهایی الگوریتم تحت‌تأثیر پارامترهای PSO می‌باشند بنابراین برای تخمین پارامترهای آن ابتدا با جایگشت‌های مختلف از مقادیر مختلف c1، c2، w ، از کوچک تا بزرگ، مقادیر تابع هدف به ازای هر جایگشت به‌دست آورده می‌شود و سپس به کمک رگرسیون یک خط با متغیرهای مستقل c1، c2، w و متغیر وابسته «مقدار تابع هدف» برازش می‌گردد و در نهایت بیشینه این خط با توجه به محدودیت‌های مربوط به c1، c2، w در نرم‌افزار Lingo حل‌شده و مقادیر تخمینی پارامترها بدست می‌آید. در این پژوهش از نرم‌افزار MATLAB برای محاسبه مقادیر تابع هدف و نیز کدنویسی الگوریتم PSO استفاده می‌شود.



شکل (۳): فلوچارت الگوریتم پیشنهادی

یافته‌های پژوهش

در حل مسائل DFLP، محققان از داده‌های یکسانی استفاده می‌کنند که اولین بار توسط بالاکریشنان و چنگ (۲۰۰۰) ارائه شدند. به‌طور کلی این داده‌ها شامل داده‌های جریان بین تسهیلات، هزینه‌های جابه‌جایی هر ماشین و نیز فواصل بین مکان‌های مختلف جهت چیدمان برای ۴۸ مسئله نمونه با اندازه‌های مختلف ۶، ۱۵ و۳۰ ماشین با افق‌های زمانی ۵ و ۱۰ دوره است. شایان ذکر است پرداختن به مسئله DFLP همراه با فرضیاتی است: (۱) فرض شده‌است که ابعاد مکان‌ها (ماشین‌ها) ثابت و غیرقابل تغییر است، (۲) هزینه‌های جابه‌جایی تسهیلات و نیز هزینه‌های انتقال مواد در طول یک دوره ثابت هستند، (۳) فرض شده‌است ارزش زمانی پول ثابت یا از قبل در هزینه‌های جابه‌جایی دوره‌های مختلف دیده شده‌است، (۴) تعداد تسهیلات و مکان‌های تخصیصی برابرند و (۵) نهایتاً اینکه مسئله بدون محدودیت بودجه‌ای فرض شده‌است.

تحقیقات زیر ازمهمترین ومعتبرترین مطالعات انجام گرفته روی DFLP به شمار می‌آیند که برای مقایسه نتایج این تحقیق و تشخیص میزان کارآمدی الگوریتم پیشنهادی استفاده می‌شوند: الگوریتم ژنتیک (کنوی و ونکاتارامانان، ۱۹۹۴)، الگوریتم ژنتیک با حلقه تودرتو (بالاکریشنان و چنگ، ۲۰۰۰)، شبیه‌سازی تبرید (بایکاسوگلو و جیندی، ۲۰۰۱)، الگوریتم ژنتیک توأم با DP (بالاکریشنان و همکاران، ۲۰۰۳) ، الگوریتم GADP با رشته آغازین تصادفی(بالاکریشنان و همکاران، ۲۰۰۳)، الگوریتم GADP با تولید رشته به‌وسیله الگوریتم معاوضه دوبه‌دوی اِربن (بالاکریشنان و همکاران، ۲۰۰۳)، الگوریتم بهینه‌سازی کلونی مورچگان (بایکاسوگلو، درلی و سابونچو، ۲۰۰۶)، SA اصلاح‌شده(مک‌کندال و شانگ، ۲۰۰۶)، SA استوار (پیلای،هوناگوند و کریشنان،۲۰۱۱).

در هریک از مسائل، جواب حاصل از شبیه‌سازی که جواب آغازین الگوریتم پیشنهادی محسوب می‌شود طی تکرارهای متعدد به تدریج بهبود می‌یابد و به بهترین جواب به‌دست‌آمده همگرا می‌گردد. در شکل(۴) برای مشاهده روند هم‌گرایی جواب در طی تکرارهای متوالی، نمودار هم‌گرایی برای ۶ مسئله یعنی مسائل شماره ۱، ۱۰، ۲۴، ۳۲، ۴۰ و ۴۸ (دو مسئله با اندازه

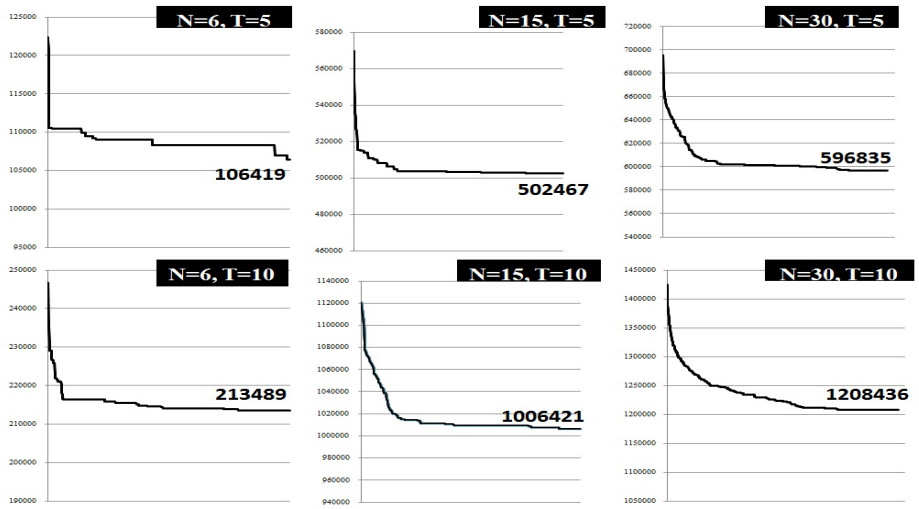
کوچک، دو مسئله با اندازه متوسط و دو مسئله با اندازه بزرگ) از بین ۴۸ مسئله نمونه، رسم گردیده‌است. محور عمودی نشان‌دهنده مقدار تابع هدف و محور افقی نشان‌دهنده تکرارهای الگوریتم است. درجداول (۱) و(۲) نیز بهترین جواب به‌دست‌آمده از بین تمام الگوریتم‌ها تاکنون، به صورت برجسته و ایتالیک مشخص شده‌است. نتایج تطبیقی نشان می‌دهند که جواب‌های الگوریتم پیشنهادی *APSOA* کاملاً رضایت‌بخش و قابل رقابت با سایر الگوریتم‌ها است و تقریباً در تمامی مسائل حل‌شده توسط *APSOA*، جواب‌های به‌دست‌آمده بهتر از CVGA، NLGA ، SA، ACO و SA استوار هستند. در سال ۱۹۸۶ جواب بهینه برای مسئله نخست (۶ ماشین و ۵ دوره) توسط الگوریتم دقیق برنامه‌ریزی پویا (روزنبلات، ۱۹۸۶) مقدار ۱۰۶۴۱۹ به‌دست‌آمده‌است که APSOA توانسته‌است جواب بهینه این مسئله را نیز بیابد. محققان امیدوارند با انجام تحقیقات آتی بیشتر واصلاحات و ابتکارات جدید در الگوریتم APSOA، جواب‌های به‌دست‌آمده از SA اصلاح‌شده را بهبود دهند. در تحقیق حاضر از یک رایانه شخصی با پردازشگر Intel(R) ,Core(TM) i5 ,2.40 GHz با RAM: 2.67 GB استفاده می‌شود. در جدول(۳) میانگین زمان‌های محاسباتی برحسب دقیقه برای مسائل نمونه نمایش داده شده‌است. در اکثر مطالعات صورت گرفته در زمینه DFLP توجه کمتری به مقوله زمان شده‌است و تأکید محققان بر کیفیت جواب‌های نهایی بوده‌است.

جدول (۱): نتایج تطبیقی مسائل جانمایی پویا با اندازه کوچک و متوسط(۶ و۱۵مکان (ماشین))

Problem size	N=6, T =5							
Problem #	P1	P2	P3	P4	P5	P6	P7	P8
APSOA	***106,419***	***104,834***	***104,320***	106,515	***105,628***	104,053	***106,439***	***103,771***
CVGA (Adaptive)	108,976	105,170	104,520	106,719	***105,628***	105,606	***106,439***	104,485
NLGA (Adaptive)	***106,419***	***104,834***	***104,320***	106,515	***105,628***	104,053	106,978	***103,771***
GADP (Adaptive)	***106,419***	***104,834***	104,529	106,583	***105,628***	104,315	106,447	***103,771***
SA (Adaptive)	107,249	105,170	104,800	106,515	106,282	***103,985***	106,447	***103,771***
ACO (Adaptive)	***106,419***	***104,834***	***104,320***	106,509	***105,628***	104,053	***106,439***	***103,771***
Modified SA (Adaptive)	***106,419***	***104,834***	***104,320***	***106,399***	***105,628***	***103,985***	***106,439***	***103,771***
Robust SA (Adaptive)	***106,419***	105,731	107,650	108,260	108,188	107,765	108,114	107,248
Best cost	106,419	104,834	104,320	106,399	105,628	103,985	106,439	103,771
Problem size	N=6, T =10							
Problem #	P9	P10	P11	P12	P13	P14	P15	P16
APSOA	215,171	213,489	208,315	212,767	211,160	211,013	215,493	214,313
CVGA (Adaptive)	218,407	215,623	211,028	217,493	215,363	215,564	220,529	216,291
NLGA (Adaptive)	214,397	212,138	208,453	212,953	211,575	210,801	215,685	214,657
GADP (Adaptive)	***214,313***	***212,134***	***207,987***	212,741	210,944	210,000	215,452	***212,588***
SA (Adaptive)	215,200	214,713	208,351	213,331	213,812	211,213	215,630	214,513
ACO (Adaptive)	217,251	216,055	208,185	212,951	211,076	210,277	215,504	214,621
Modified SA (Adaptive)	***214,313***	***212,134***	***207,987***	***212,530***	***210,906***	***209,932***	***214,252***	***212,588***
Robust SA (Adaptive)	220,776	217,412	219,024	217,350	217,142	217,397	219,788	220,144
Best cost	214,313	212,134	207,987	212,530	210,906	209,932	214,252	212,588
Problem size	N=15, T =5							
Problem #	P17	P18	P19	P20	P21	P22	P23	P24
APSOA	498,897	499,374	503,421	501,012	500,331	493,476	496,722	502,467
CVGA (Adaptive)	504,759	514,718	516,063	508,532	515,599	509,384	512,508	514,839
NLGA (Adaptive)	511,854	507,694	518,461	514,242	512,834	513,763	512,722	521,116
GADP(R) (Adaptive)	493,707	494,476	506,684	500,826	502,409	497,382	494,316	500,779
GADP(U) (Adaptive)	484,090	485,352	489,898	484,625	489,885	488,640	489,378	500,779
SA (Adaptive)	501,447	506,236	512,886	504,956	509,636	508,215	508,848	512,320
ACO (Adaptive)	501,447	506,236	512,886	504,956	509,636	508,215	508,848	512,320
Modified SA (Adaptive)	***480,453***	***484,761***	***488,748***	***484,405***	***487,882***	***487,147***	***486,779***	***490,812***
Robust SA (Adaptive)	506,847	500,284	508,011	503,699	502,622	499,891	502,919	507,970
Best cost	480,453	484,761	488,748	484,405	487,882	487,147	486,779	490,812
Problem size	N=15, T =10							
Problem #	P25	P26	P27	P28	P29	P30	P31	P32
APSOA	1,002,912	1,006,421	1,010,987	999,364	1,004,861	1,002,898	1,007,378	1,008,510
CVGA (Adaptive)	1,055,536	1,061,940	1,073,603	1,060,034	1,064,692	1,066,370	1,066,617	1,068,216
NLGA (Adaptive)	1,047,596	1,037,580	1,056,185	1,026,789	1,033,591	1,028,606	1,043,823	1,048,853
GADP(R) (Adaptive)	1,004,806	1,006,790	1,012,482	1,001,795	1,005,988	1,002,871	1,019,645	1,010,772
GADP(U) (Adaptive)	987,887	980,638	985,886	976,025	982,778	973,912	982,872	987,789
SA (Adaptive)	1,017,741	1,016,567	1,021,075	1,007,713	1,010,822	1,007,210	1,013,315	1,019,092
ACO (Adaptive)	1,017,741	1,016,567	1,021,075	1,007,713	1,010,822	1,007,210	1,013,315	1,019,092
Modified SA (Adaptive)	***979,468***	***978,065***	***982,396***	***972,797***	***977,188***	***967,617***	***979,114***	***983,672***
Robust SA (Adaptive)	1,059,100	1,022,447	1,068,402	1,054,997	1,051,395	1,057,543	1,037,066	1,040,450
Best cost	979,468	978,065	982,396	972,797	977,188	967,617	979,114	983,672

جدول (۲): نتایج تطبیقی حاصل از مسائل جانمایی پویا با اندازه بزرگ (۳۰ مکان (ماشین))

Problem size	N=30, T =5							
#Problem	P33	P34	P35	P36	P37	P38	P39	P40
APSOA	578,746	576,301	586,831	583,007	570,813	572,098	569,898	596,835
CVGA (Adaptive)	632,737	647,585	642,295	634,626	639,693	637,620	640,482	635,776
NLGA (Adaptive)	611,794	611,873	611,664	611,766	604,564	606,010	607,134	620,183
GADP(R) (Adaptive)	603,339	589,834	592,475	586,064	580,624	587,797	588,347	590,451
GADP(U) (Adaptive)	578,689	572,232	578,527	572,057	559,777	566,792	567,873	575,720
SA (Adaptive)	604,408	604,370	603,867	596,901	591,988	599,862	600,670	610,474
ACO (Adaptive)	604,408	604,370	603,867	596,901	591,988	599,862	600,670	610,474
Modified SA (Adaptive)	***576,039***	***568,095***	***573,739***	***566,248***	***558,460***	***566,077***	***567,131***	***573,755***
Robust SA (Adaptive)	579,704	576,350	586,831	584,318	570,492	572,782	571,703	596,835
Best cost	576,039	568,095	573,739	566,248	558,460	566,077	567,131	573,755
Problem size	N=30, T =10							
Problem#	P41	P42	P43	P44	P45	P46	P47	P48
APSOA	1,193,876	1,182,286	1,196,981	1,187,153	1,177,201	1,178,309	1,196,364	1,208,436
CVGA (Adaptive)	1,362,513	1,379,640	1,365,024	1,367,130	1,356,860	1,372,513	1,382,799	1,383,610
NLGA (Adaptive)	1,228,411	1,231,978	1,231,829	1,227,413	1,215,256	1,221,356	1,212,273	1,245,423
GADP(R) (Adaptive)	1,194,084	1,199,001	1,197,253	1,184,422	1,179,673	1,178,091	1,186,145	1,208,436
GADP(U) (Adaptive)	1,169,474	1,168,878	1,166,366	1,154,192	1,133,561	1,145,000	1,145,927	1,168,657
SA (Adaptive)	1,223,124	1,231,151	1,230,520	1,200,613	1,210,892	1,221,356	1,212,273	1,231,408
ACO (Adaptive)	1,223,124	1,231,151	1,230,520	1,200,613	1,210,892	1,239,255	1,248,309	1,231,408
Modified SA (Adaptive)	***1,163,222***	***1,161,521***	***1,156,918***	***1,145,918***	***1,126,432***	***1,145,146***	***1,140,744***	***1,161,437***
Robust SA (Adaptive)	1,172,691	1,182,286	1,188,620	1,198,487	1,198,674	1,202,033	1,210,573	1,209,088
Best cost	1,163,222	1,161,521	1,156,918	1,145,918	1,126,432	1,145,146	1,140,744	1,161,437



شکل (۴): نمودارهای هم‌گرایی جواب برای مسائل شماره ۱، ۱۰، ۲۴، ۳۲، ۴۰ و ۴۸

جدول (۳): میانگین زمان‌های محاسباتی مسائل نمونه

میانگین زمان اجرا	N	T	مسائل	میانگین زمان
	6	5	P1 to P8	9.3
	6	10	P9 to P16	17.9
	15	5	P17 to P24	99.3
	15	10	P25 to P32	173.8
	30	5	P33 to P40	210.0
	30	10	P41 to P48	398.9

نتیجه‌گیری و پیشنهادها

در تحقیق حاضر، برای نخستین‌بار دو رویکرد شبیه‌سازی و بهینه‌سازی ازدحام ذرات برای مسئله پیچیده جانمایی پویای تسهیلات ترکیب شد و الگوریتم ابتکاری جدیدی به نام *APSOA* برای حل DFLP پیشنهاد گردید. تحقیقات نشان داد، استفاده از *APSOA* در مقایسه با BPSO معمولی بسیار اثربخش‌تر و زمان تولید جواب‌های موجه را فوق‌العاده کاهش می‌دهد. الگوریتم پیشنهادی روی داده‌های مسائل نمونه موجود در ادبیات DFLP آزمون گردید و نتایج تطبیقی یافته‌ها نشان از عملکرد بسیار خوب *APSOA* داشت. جهت انجام تحقیقات آتی مسیرهای پژوهشی زیر به محققان علاقمند پیشنهاد می‌گردد: دخالت دادن فاصله مکان اولیه و مکان جدید ماشین در محاسبات هزینه جابه‌جایی تسهیلات، برای واقعی‌ترشدن مدل، استفاده از PSO در حل DFLP با محدودیت بودجه، حل DFLP را با درنظرگرفتن ارزش زمانی پول در یک اقتصاد تورمی، استفاده از رویکرد برنامه‌ریزی چندهدفه برای تحقق همزمان چند هدف (کمی یا کیفی)، تحلیل حساسیت بین ابعاد مختلف مسئله و الگوریتم‌های موردنظر، استفاده از PSO در حل DFLP با تسهیلات غیرهم‌اندازه و استفاده از رویکردهای فراابتکاری ترکیبی.

منابع

Balakrishnan, J., & Cheng, C. H. (2000). Genetic search and the dynamic layout problem. *Computers & Operations Research*, 27(6), 587–593.

Balakrishnan, J., Cheng, C. H., & Conway, D. G. (2000). An improved pair-wise exchange heuristic for the dynamic plant layout problem. *International Journal of Production Research*, 38(13), 3067–3077.

Balakrishnan, J., Cheng, C. H., Conway, D. G., & Lau, C. M. (2003). A hybrid genetic algorithm for the dynamic plant layout problem. *International Journal of Production Economics*, 86(2), 107–120.

Balakrishnan, J., Robert Jacobs, F., &Venkataramanan, M. A. (1992). Solutions for the constrained dynamic facility layout problem. *European Journal of Operational Research*, 57(2), 280–286.

Baykasoglu, A., & Gindy, N. N. Z. (2001). A simulated annealing algorithm for dynamic layout problem. *Computers & Operations Research*, 28(14), 1403–1426.

Baykasoglu, A., Dereli, T., & Sabuncu, I. (2006). An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. *Omega*, 34(4), 385–396.

Conway, D. G., & Venkataramanan, M. A. (1994). Genetic search and the dynamic facility layout problem. *Computers and Operations Research*, 21(8), 955–960.

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31, 255–267.

Kaku, B.K. & Mazzola, J.B. (1997). A Tabu-search heuristic for the dynamic plant layout problem. *INFORMS Journal on Computing,* 9 (4), 374-384.

Kennedy, J. (1997). The particle swann: social adaptation of knowledge. *International Conference on Evolutionary Computation (Indianapolis, Indiana)*, IEEE Service Center, Piscataw;ty, NJ, 303-308.

Kennedy, J., & Eberhart, R. (1995b). A new optimizer using particle swarm theory. *In Sixth international symposium on micro machine and human science*, IEEE, 39–43.

Kennedy, J., & Eberhart, R. C. (1995a). Particle swarm optimization. *Proc. IEEE Intl. Conf. on Neural Networks (Perth, Australia)*, IEEE Service Center, Piscataway, NJ, IV, 1942-1948.

Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm, systems, man, cybernetics, computational cybernetics and simulation. *IEEE International Conference*, 5(12–15), 4104–4108.

Lacksonen, T. A., & Enscore, E. E. (1993). Quadratic assignment algorithms for the dynamic layout problem. *International Journal of Production Research*, 31, 503–517.

Lazinica, A. (2009). Particle swarm optimization. Austria. In-Tech.

McKendall, A. R. & Hakobyan, A. (2010). Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research*, 201, 171–182.
McKendall, A. R. & Shang, J. (2006). Hybrid ant systems for the dynamic facility layout problem. *Computers & Operations Research*, 33,790–803.
McKendall, A. R., Shang, J., & Kuppusamy, S. (2006). Simulated annealing heuristics for the dynamic facility layout problem. *Computers& Operations Research*, 33(8), 2431–2444.
Ning, X. Lam, K.C. & Lam, M. C. K. (2010). Dynamic construction site layout planning using max-min ant system. *Automation in Construction* ,19, 55–65.
Olariu, S. & Zomaya, A. Y. (2006). Handbook of bio-inspired algorithms and applications. Chapman & Hall/CRC, Taylor & Francis Group.
Pillai, V. M., Hunagund, I. B., & Krishnan, K. K. (2011). Design of robust layout for dynamic plant layout problems. *Computers & Industrial Engineering*, (In press), doi:10.1016/j.cie.2011.05.014.
Reche-Lopez, P., Ruiz-Reyes, N., Galan, G. S. & Jurado, F. (2009). Comparison of metaheuristic techniques to determine optimal placement of biomass power plants. *Energy Conversion and Management*, 50, 2020–2028.
Rosenblatt, M. J. (1986). The dynamics of plant layout. *Management Science*, 32(1), 76–86.
Talbi. E.G. (2009). Metaheuristics, from design to implementation. John Wiley & Sons, Inc.
Tompkins, J. A., White, J. A., Bozer, Y. A., Frazelle, E. H., Tanchoco, J. M., & Trevino, J. (1996). Facilities planning. New York: Wiley.
Urban, T.L. (1993). A heuristic for the dynamic facility layout problem. *IIE Transactions*, 25(4), 57-63.