یک الگوریتم کارا برای حل مساله برنامه‌ریزی کارگاهی دو هدفه در حالت فازی با استفاده از الگوریتم ژنتیک و روش داده کاوی اطلاعات

مسعود ربانی
استادیار
گروه مهندسی صنایع، دانشکده فنی، دانشگاه تهران

رضا توکلی‌مقدم
استادیار
گروه مهندسی صنایع، دانشکده فنی، دانشگاه تهران

محمد رنجبر
دانشجوی دکتری
دانشکده مهندسی صنایع، دانشگاه صنعتی شریف

چکیده

در این مقاله یک الگوریتم کارا[1] برای حل مساله برنامه‌ریزی کارگاهی در حالت فازی[2] و با در نظر گرفتن دو هدف کمینه کردن زمان انجام کل کارها و کمینه کردن میزان تأخیر یا تحویل زودتر از موعد هر کار ارائه می‌شود. در این الگوریتم پارامترهای زمان تحویل و زمان انجام بصورت اعداد فازی مثلثی در نظر گرفته می‌شوند. در این مقاله از تکنیک داده کاوی اطلاعات[3] به عنوان یک روش کمکی در حل مسائل بهینه‌سازی ترکیبی[4] استفاده می‌شود. در این مقاله با ترکیب الگوریتم ژنتیک[5] و یک الگوریتم استنتاج صفت‌گرا[6] به عنوان یک روش داده کاوی اطلاعات به سمت جواب بهینه پیش می‌رویم. با توجه به ساختار الگوریتم طراحی شده، تمام جوابهای قابل قبول یک مساله برنامه‌ریزی کارگاهی خاص به عنوان یک پایگاه داده‌ها تلقی می‌شوند که تکنیک داده کاوی اطلاعات و الگوریتم استنتاج صفت‌گرا باید بتوانند الگوها و روابط نهفته در میان این جوابها را با توجه به خصوصیات آنها حدس زده و به الگوریتم ژنتیک به منظور یک جستجوی هدفمندتر کمک کنند. به دلیل وجود خواص توارثی در جوابهای حاصل از الگوریتم ژنتیک، قاعدتاً نسلهای آینده باید دارای خواص و الگوهای رفتاری مشابهتری نسبت به نسلهای اولیه باشند. البته فقط با استفاده از الگوریتم ژنتیک و بدون استفاده از روش داده کاوی اطلاعات نیز می‌توان به جوابهای نزدیک به بهینه و یا حتی جواب بهینه رسید، ولی تکنیک داده کاوی اطلاعات با بررسی خصوصیات برنامه‌های زمانی و توالیهای نزدیک به بهینه و کشف قوانین و روابط نهفته در میان این جوابها، عملکرد الگوریتم ژنتیک را به میزان قابل توجهی بهبود می‌بخشد.

کلمات کلیدی

مساله برنامه‌ریزی کارگاهی فازی، الگوریتم ژنتیک، داده کاوی اطلاعات، استنتاج صفت‌گرا

An Efficient Algorithm For Solving Bi-objective Fuzzy Job-Shop Scheduling Problems by Genetic Algorithms and Data Mining

M. Rabbani
Assistant Professor
Department of Industrial Engineering, Faculty of Engineering, University of Tehran

R. Tavakkoli Moghaddam
Assistant Professor
Department of Industrial Engineering, Faculty of Engineering, University of Tehran

M.Ranjbar
Ph.D Student
Department of Industrial Engineering,
Faculty of Engineering, University Technology Sharif

Abstract

This paper presents a meta-heuristic algorithm for solving bi-objective fuzzy job shop scheduling problems. These objectives are to minimize the makespan and minimize the early and late penalty. Processing time and due date are considered as fuzzy triangular numbers. This paper also introduces a novel use of data mining algorithm for solving of combinatorial optimization problems. The proposed algorithm combines genetic algorithms and an attribute-oriented induction algorithm, which is much quicker than previous methods providing the optimal solution. By considering the structure of proposed algorithm, the whole feasible solutions of a special job shop-scheduling problem are considered as a database. Attribute-oriented induction algorithm must find similar relationships and latent patterns among this database. This algorithm helps genetic algorithms to find the optimal or near-optimal solution much quicker than previous methods. Due to genetic inheritance in solutions obtained by genetic algorithms, the next generation has more similar relationships between their characteristics, operations and sequences. In this paper, three test problems are conducted in order to compare with other solutions reported in the literature and to show the efficiency of the proposed algorithm.

Keywords

Genetic algorithms, Jobshop problem, Data mining, Fuzzy jobshop sheduling

مقدمه

مساله برنامه‌ریزی کارگاهی یکی از مشهورترین و سخت‌ترین مسائل بهینه‌سازی ترکیبی است و جز مسائل NP-hard می‌باشد[۱]. این مساله از دو جنبه تحقیقاتی مدیریت تولید و بهینه‌سازی ترکیبی دارای اهمیت است و در طول سه دهه اخیر توجه بسیاری از محققان را به خود جلب کرده است، بطوریکه از سال ۱۹۶۰ به بعد به عنوان معیاری جهت ارزیابی کیفیت تکنیکهای جدید بهینه‌سازی مطرح شده است. دیویس[۷] جزء اولین کسانی است که سعی کرد مساله برنامه‌ریزی کارگاهی را بوسیله الگوریتم ژنتیک حل کند[۱]. بعد از دیویس تا به امروز نیز تلاشهای زیادی برای حل این مساله بوسیله الگوریتم ژنتیک صورت گرفته است[۱]. علاوه بر الگوریتم ژنتیک روشهای ابتکاری دیگری نیز برای حل این مساله ارائه شده‌اند که از جمله موفقترین آنها می‌توان به روشهای انتقال گلوگاه[۸][۲]، روش شبیه‌سازی بازپخت[۹][۳]، روش جستجوی محدود[۱۰][۳]، و روش جستجوی همسایگی[۱۱][۳] اشاره کرد. با وجود موفقیتهای چشم‌گیری که در استفاده از روشهای فوق برای حل مسائل بهینه‌سازی ترکیبی حاصل شده است اما بخاطر طبیعت انعطاف‌پذیر این روشها و بخصوص روش الگوریتم ژنتیک، هنوز امکان ایجاد بهبود زیادی در عملکرد این روشها وجود دارد. در این مقاله سعی شده است تا با نگاهی متفاوت و تحلیل گرانه به مساله برنامه‌ریزی کارگاهی که یکی از مسائل بهینه سازی ترکیبی می‌باشد، عملکرد الگوریتم ژنتیک را در حل این مساله بهبود بخشید. در اینجا ما با استفاده از روش داده کاوی اطلاعات خصوصیات جوابها را بررسی کرده و روابط نهفته در میان آنها را بدست می‌آوریم. داده کاوی اطلاعات روشی است جهت مدیریت بهتر پایگاه داده‌های حجیم. در این مقاله هر برنامه زمانی فعال[۱۲] که یک برنامه و توالی قابل قبول از مساله برنامه‌ریزی کارگاهی می‌باشد، بعنوان یک عضو پایگاه داده‌ها در نظر گرفته می‌شود. سپس با کلاس‌بندی[۱۳] عملیاتها و ثبت خصوصیات آنها، روابط منطقی بین خصوصیات عملیاتها و توالی آنها را از طریق یک الگوریتم استنتاج صفت‌گرا بدست می‌آوریم. قوانین مکشوفه، الگوریتم ژنتیک را در حرکت هدفمندتر و سریعتر به سمت نقطه بهینه کمک می‌کنند. این مقاله شامل پنج بخش است.

بخش اول شامل مقدمه، بخش دوم شامل بررسی مساله برنامه‌ریزی کارگاهی و بررسی حالت فازی آن، بخش سوم شامل توضیح مختصری درباره الگوریتم ژنتیک و روش داده کاوی اطلاعات، بخش چهارم شامل الگوریتم طراحی شده و بخش پنجم شامل نتایج شبیه‌سازی و مقایسه نتایج با دیگر روشها می‌باشد.

۱ـ مساله برنامه‌ریزی کارگاهی

مساله برنامه‌ریزی کارگاهی یک مساله بسیار مهم و پیچیده در بحث مدیریت تولید بهینه‌سازی ترکیبی می‌باشد. در یک مساله برنامه‌ریزی کارگاهی در حالت عمومی، مجموعه‌ای از کارها با توالیهای عملیاتی مشخص باید بر روی ماشینهای مربوطه انجام شوند.

در یک مساله برنامه‌ریزی کارگاهی، m ماشین[۱۴] و n کار[۱۵] موجود است. هر کار یا قطعه به منظور تکمیل، دارای یک فرآیند خاص می‌باشد و توالی انجام عملیاتهای مختلف به منظور تکمیل آن کار خاص، مشخص می‌باشند. هر عملیات یک کار، دارای زمان خاص خود می‌باشد. در این مساله عملیات k هر کار روی ماشین k انجام می‌شود. هدف از حل این مساله مشخص کردن توالی انجام کارها بر روی ماشینها می‌باشد بطوریکه معیار عملکرد بهینه شود.

معیار عملکرد در اینجا حداقل کردن زمان انجام کل کارها[۱۶] می‌باشد. در مساله برنامه‌ریزی کارگاهی هر کار یا هر عملیات می‌تواند یک زمان شروع خاص داشته باشد که در این مقاله زمان شروع تمام عملیاتها صفر در نظر گرفته شده است.

مساله برنامه‌ریزی کارگاهی جزء مسائل NP-Hard می‌باشد و تعداد حالتهای ممکن برای ترتیب عملیاتها بر روی ماشینها با افزایش ابعاد مساله به صورت نمایی رشد می‌کند. برای یک مساله برنامه‌ریزی کارگاهی با ۶ کار و یک ماشین تعداد حالات ممکن (قابل قبول و غیرقابل قبول) برای توالی عملیاتها برابر است با ۷۲۰= !۶ در حالیکه برای یک مساله برنامه ریزی کارگاهی با ۶ کار و ۶ ماشین تعداد حالات ممکن برابر است با ۵۱۸۴۰۰= !۶ × !۶. لذا یک جستجوگر ساده نمی‌تواند براحتی جواب بهینه را پیدا کند. الگوریتمهای جستجوگر مختلفی برای حل این مساله طراحی شده‌اند که چند مورد از آنها در بخش اول ذکر شد، ولی هنوز هیچ کدام از آنها تضمین کننده جواب بهینه نیستند.

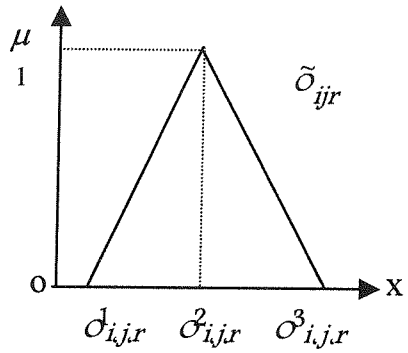
البته باید توجه داشت که در اصل تعداد برنامه‌های زمانی امکان‌پذیر مساله برنامه‌ریزی کارگاهی نامتناهی است. زیرا زمان بیکاری دلخواهی را می‌توان در فاصله میان دو عملیات مجاور با هر ماشین در نظر گرفت.

به هر حال واضح است پس از اینکه توالی عملیات هر ماشین مشخص شد، اعمال این نوع زمان بیکاری در فاصله مابین دو عملیات در هر معیار عادی عملکرد مفید نخواهد بود. به هر حال لزومی ندارد که کلیه حالات ممکن بررسی شود و می‌توان با تعریف مجموعه‌ای بنام مجموعه برنامه زمانی فعال که فقط شامل مجموعه جوابهای قابل قبول می‌باشد، تعداد حالات مورد بررسی را کاهش داد. جواب بهینه لزوماً در این مجموعه قرار دارد. الگوریتم روش ایجاد برنامه زمانی فعال[۴] در بخش ۳ این مقاله موجود می‌باشد.

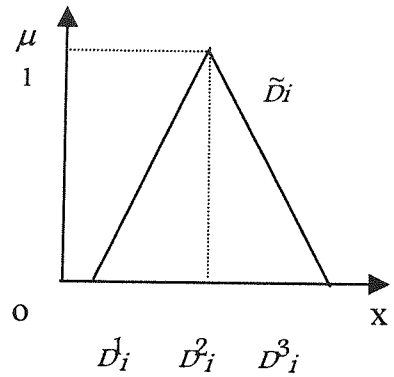
۱-۱ـ برنامه‌ریزی کارگاهی در حالت فازی

فرض کنید در یک مساله برنامه‌ریزی کارگاهی، تعداد کارها J_i (i=1, 2, ..., n) n و تعداد ماشینها M_r (r=1,2,...,m) m می‌باشد. O_{ijr} را بعنوان زمان عملیات jام کار iام روی ماشین r و D_i را به عنوان زمان تحویل کار iام در نظر بگیرید. در برنامه‌ریزی کارگاهی فازی زمان انجام هر عملیات و زمان تحویل هر کار بصورت پارامترهای فازی می‌باشند. بنا براین $\tilde{O}_{ijr}$ و $\tilde{D}_i$ بصورت اعداد فازی هستند. در اینجا این اعداد بصورت اعداد فازی مثلثی[۱۷] [۵] مطابق شکل‌های (۱) و (۲) در نظر گرفته شده‌اند. البته می‌توان زمان تحویل را به صورت یک رابطه خطی مطابق شکل (۳)، با مطلوبیت ۱ برای مقادیر کمتر از زمان تحویل نیز در نظر گرفت[۶]. ولی در اینجا به دلیل لحاظ کردن هزینه‌های انبارداری، تحویل زودتر از موعد نیز مامطلوب تلقی می‌شود و این عدد فازی نیز بصورت مثلثی در نظر گرفته شده است. در این قسمت باید توجه داشت که اعداد فازی بصورت نرمال در نظر گرفته شده‌اند و به همین دلیل حداکثر مطلوبیت ۱ در نظر می‌باشد. اگر در مساله‌ای اعداد فازی نرمال نباشند می‌توان براحتی آنها را نرمال نمود[۵].

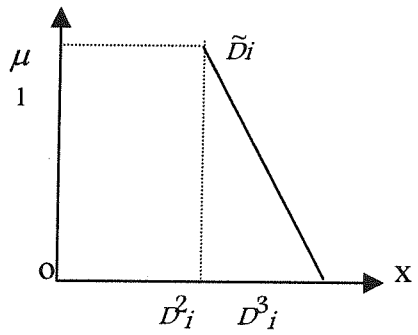
μ = تابع مطلوبیت



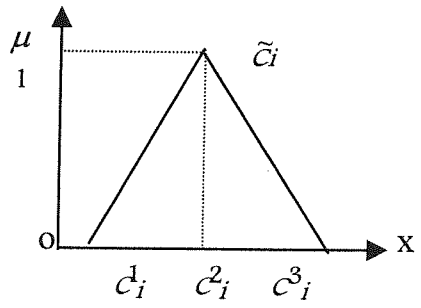
شکل (۱) زمان انجام عملیات بصورت عدد فازی مثلثی.



شکل (۲) زمان تحویل بصورت عدد فازی مثلثی.



شکل (۳) زمان تحویل بصورت عدد فازی خطی.



شکل (٤) زمان تکمیل فازی.

زمان تکمیل هر کار از جمع زمان‌های تمام عملیات آن کار بدست می‌آید. بنابراین عدد حاصل نیز یک عدد فازی خواهد بود شکل (۴).

البته باید توجه داشت که عدد حاصل لزوماً یک عدد فازی مثلثی نمی‌شود. به منظور فهم بیشتر این مطلب به مثال زیر توجه کنید:

فرض کنید $\tilde{A}$ و $\tilde{B}$ دو عدد فازی باشند، بطوریکه

$$\tilde{B} = (b^1, b^2, b^3)\ ,\ \tilde{A} = (a^1, a^2, a^3)$$
$$\tilde{A} + \tilde{B} = (a^1, a^2, a^3) + (b^1, b^2, b^3) \qquad (1)$$
$$= (a^1, b^2, a^2 + b^1, a^3, b^3)$$

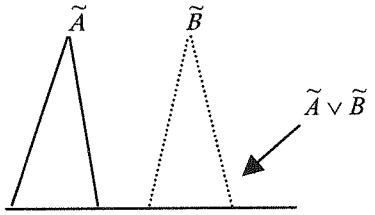
اگر تابع مطلوبیت را برای $\tilde{A}$ و $\tilde{B}$ بصورت $\mu_{\tilde{A}}$ و $\mu_{\tilde{B}}$ در نظر بگیریم، طبق اصل گسترش[18] زاده[19] برای تابع مطلوبیت $\tilde{A} + \tilde{B}$، $\mu_{\tilde{A}\vee\tilde{B}}$، خواهیم داشت:

$$\mu_{\tilde{A}\vee\tilde{B}}(z) = \sup(\min(\mu_{\tilde{A}(x)}, \mu_{\tilde{B}(x)})) \qquad (2)$$
$$Z = x \vee y$$

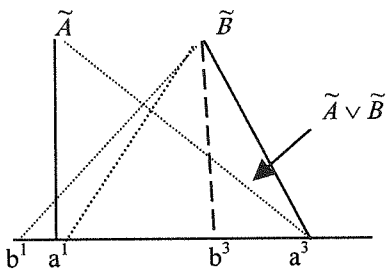
که عدد حاصل از عملکرد (max) $\vee$ براساس اصل گسترش همیشه بصورت مثلثی نیست و در واقع بصورت زیر می‌باشد ولی ما برای سادگی، این اعداد را مثلثی در نظر می‌گیریم[۵].

$$\tilde{A} \vee \tilde{B} = (a^1, a^2, a^3) \vee (b^1, b^2, b^3) \cong (a^1 \vee b^1, a^2 \vee b^2, a^3, b^3) \quad (۳)$$

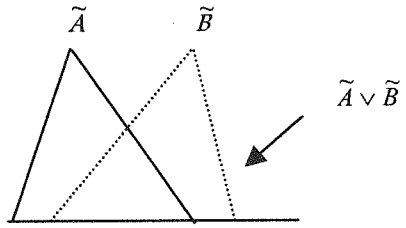
شکل‌های (۵) و (۶) و (۷) حالتهای مختلف $\tilde{A} \vee \tilde{B}$ را نشان می‌دهند.



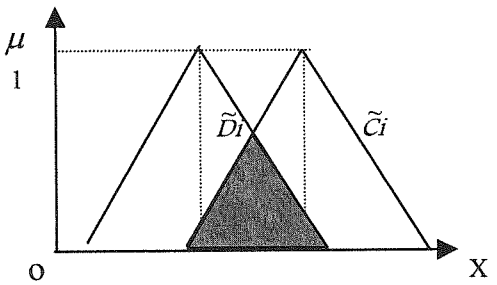
شکل (۵) $\tilde{A} \vee \tilde{B}$.



شکل (۷) $\tilde{A} \vee \tilde{B}$.



شکل (٦) $\tilde{A} \vee \tilde{B}$.



شکل (۸) تصویر تأخیر در حالت فازی.

شکل (۸) میزان اختلاف زمان تکمیل کار iام از زمان تحویل آن را نشان می‌دهد.

اگر سطح مشترک بین $\tilde{C}i$ و $\tilde{D}i$ را با AI[۲۰] نمایش دهیم $AI = S(\tilde{C}) \cap S(\tilde{D})$ و با توجه به اینکه هدف ما در اینجا کمینه‌کردن زمان کل انجام کارها و همچنین کمینه‌کردن جریمه ناشی از تأخیر و یا تحویل زودتر از موعد می‌باشد، پس می‌توان اهداف مدل را بصورت زیر بیان کرد:

$$Maximize\, Z_1 = \frac{\sum_{i=1}^{n} AI_i}{n} \quad (۴)$$

$$Maximize\, Z_2 = AI_{\min} = Min(AI_i) \quad (۵)$$

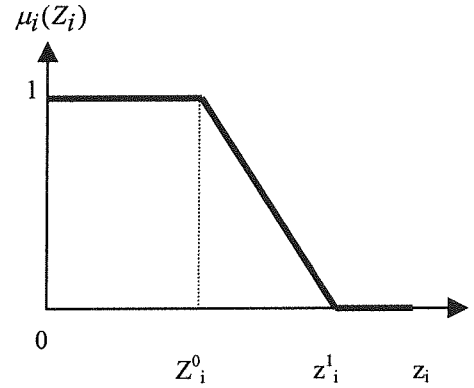
$$Maximize\, Z_3 = \tilde{C}AI_{\max} = Max(\tilde{C}_i) \quad (۶)$$

که در هر سه رابطه داریم i=1,2,...,n .

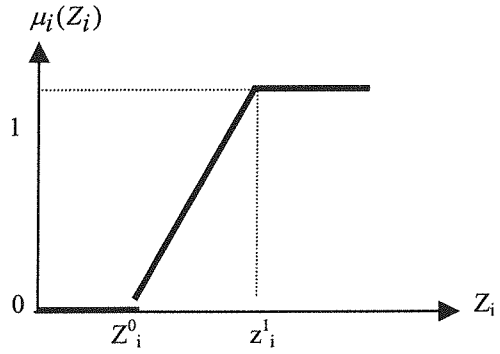
در اینجا بخاطر ساده شدن کار، تابع مطلوبیت را برای هر سه هدف بصورت خطی در نظر می‌گیریم.

$$\mu_i(z_i) = \begin{cases} 0 & if \quad Z_i \geq Z_i^1 \\ \dfrac{Z_i^1 - Z_i}{Z_i^1 - Z_i^0} & if \quad Z_i^0 \prec Z_i \prec Z_i^1 \\ 1 & if \quad Z_i \leq Z_i^0 \end{cases} \quad (۷)$$

$$\mu_i(z_i) = \begin{cases} 0 & if \quad Z_i \geq Z_i^1 \\ \dfrac{Z_i - Z_i^0}{Z_i^1 - Z_i^0} & if \quad Z_i^0 \prec Z_i \prec Z_i^1 \\ 1 & if \quad Z_i \geq Z_i^0 \end{cases} \qquad (۸)$$



شکل (۹) نمودار تابع مطلوبیت با هدف بیشینه سازی (توابع Z_1 , Z_2).



شکل (۱۰) نمودار تابع مطلوبیت با هدف کمینه‌سازی.

در اینجا چون دو هدف مد نظر است ضرایب جریمه برای تأخیر و تحویل زودتر از موعد همه مساوی ۱ در نظر گرفته شده است. این ضرایب بسته به اهمیت نسبی تأخیر و تحویل زودتر از موعد می‌توانند هر مقدار دلخواهی باشند. اهداف Z_1 و Z_2 هر دو بیانگر حداکثر کردن سطح محصور در شکل بالا می‌باشند. برای محاسبه سطح محصور بین این دو عدد فازی از طریق انتگرال باید حالات مختلف این دو عدد نسبت بهم در نظر گرفته شوند. اگر

$$AI = S(\tilde{C}) \cap S(\tilde{D}) \qquad (۹)$$

بنابراین تابع هدف کل برای یک برنامه زمانی بصورت رابطه (۱۰) تعریف می‌شود:

$$Z = Max\left(\frac{\sum_{i=1}^{n} AI_i}{n} - Makespan \right) \qquad (۱۰)$$

۱-۱-۱ـ ترتیب اعداد فازی مثلثی

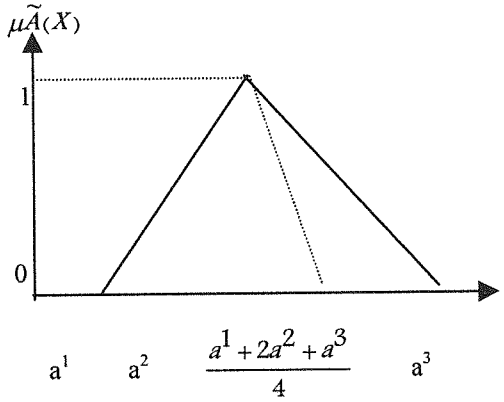
چون زمان تکمیل و همچنین زمان تحویل هر کار (قطعه) یک عدد فازی مثلثی است و از طرفی این اعداد باید با یکدیگر مقایسه شوند روش زیر جهت مقایسه اعداد فازی استفاده می‌شود[۷]:

برای مقایسه دو عدد فازی از سه معیار زیر به ترتیب استفاده می‌کنیم. معیار دوم وقتی مورد استفاده قرار می‌گیرد که در معیار اول دو عدد برابر باشند و همچنین معیار سوم نیز هنگامی مورد استفاده قرار می‌گیرد که دو عدد در دو معیار اول و دوم برابر باشند.

$$C_1(\tilde{A}) = \frac{a^1 + 2a^2 + a^3}{4} \qquad (۱۱)$$

$$C_2(\tilde{A}) = a^2 \qquad (۱۲)$$

$$C_3(\tilde{A}) = a^3 - a^1 \qquad (۱۳)$$



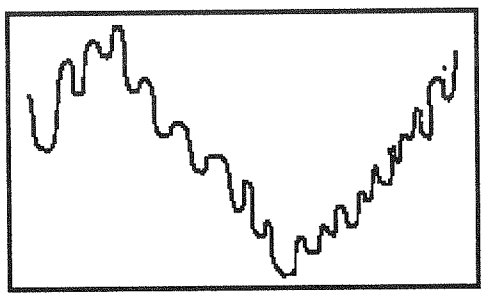
شکل (۱۱) ترتیب اعداد فازی.

۲ـ الگوریتم ژنتیک، داده کاوی اطلاعات و مساله برنامه‌ریزی کارگاهی

۲ـ۱ـ الگوریتم ژنتیک

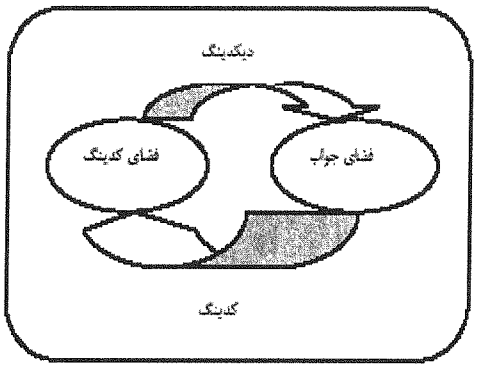
الگوریتم ژنتیک یک روش آماری برای بهینه‌سازی و جستجو است. الگوریتم ژنتیک جزئی از محاسبات تکاملی است که خود جزئی از هوش مصنوعی می‌باشد. ویژگیهای خاص این الگوریتم باعث می‌شود که نتوانیم آن را یک جستجوگر تصادفی ساده قلمداد کنیم. در واقع ایده اولیه این روش از نظریه تکاملی داروین[۲۱] الهام گرفته شده و کارکرد آن براساس ژنتیک طبیعی استوار می‌باشد[۸].

هنگام استفاده از الگوریتم ژنتیک به منظور حل یک مساله بهینه‌سازی باید فاکتورهایی نظیر فضای جستجو[۲۲]، نوع و نحوه کدگذاری[۲۳]، شکل و طول کروموزومها[۲۴]، تعداد جمعیت[۲۵] در هر نسل[۲۶]، تابع برازندگی[۲۷] و نوع و شکل عملگرهای جابجایی[۲۸] و جهشی[۲۹] را مد نظر داشت. فضای جستجو، فضایی است که ما در آن فضا به دنبال جواب بهینه می‌گردیم. شکل (۱۲) نمونه‌ای از یک فضای جستجو را نشان می‌دهد.

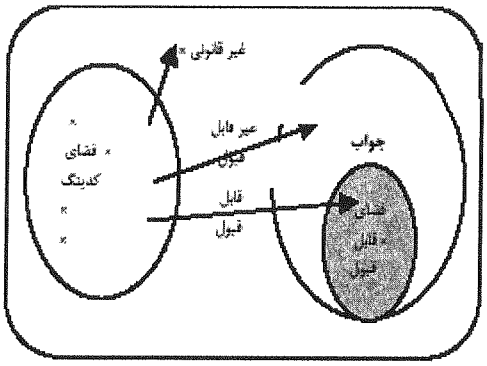


شکل (۱۲) نمونه‌ای از فضای جستجو.

در مرحله کدگذاری باید دقت شود که نوع کدگذاری باید متناسب با نوع مساله باشد و همچنین باید در مرحله کدگذاری و عکس کدگذاری[۳۰] از بروز جوابهای غیرقانونی[۳۱] و غیرقابل قبول[۳۲] جلوگیری کرد. شکلهای (۱۳) و (۱۴) رابطه بین فضای جواب و فضای کدگذاری را نشان می‌دهند.



شکل (۱۳) رابطه بین فضای جواب و فضای کدگذاری.



شکل (۱٤) رابطه بین فضای جواب و فضای کدگذاری.

شکل و طول کروموزمها در یک الگوریتم ژنتیک باید به گونه‌ای باشد که الگوریتم قادر به حل مساله با دقت مورد نظر و جستجو در فضای جواب باشد. تعداد جمعیت در هر نسل بستگی به نوع مساله دارد. تابع برازندگی در واقع میزان مطلوبیت حاصل از جوابهای یک الگوریتم ژنتیک را نشان می‌دهد. شانس انتخاب هر کروموزوم به عنوان کروموزوم والد[۳۳] برای تولید فرزندهای[۳۴] نسل آینده متناسب با مقدار برازندگی آن کروموزوم می‌باشد. همیشه بهترین کروموزوم را بدون هیچ تغییری به نسل آینده منتقل می‌کنیم. عملگرهای جابجایی و جهشی نقش بسیار تعیین کننده‌ای در عملکرد الگوریتم ژنتیک خواهند داشت. از عملگرهای جابجایی معروف[۹] می‌توان به عملگر PMX[۳۵]، عملگر OX[۳۶] و عملگر جابجایی براساس موقعیت[۳۷] اشاره کرد.

۲-۲ـ الگوریتم ژنتیک و مساله برنامه‌ریزی کارگاهی

در طول یک دهه گذشته تلاشهای فراوانی به منظور حل مسائل برنامه‌رزی کارگاهی توسط الگوریتم ژنتیک صورت گرفته است. چگونگی تطبیق الگوریتم ژنتیک با مسائل برنامه‌ریزی کارگاهی یکی از موضوعاتی است که بسیاری از محققان در حال تحقیق بر روی آن می‌باشند. در تمامی این فعالیتهای تحقیقاتی دو مساله مهم مطرح بوده است:

۱ـ چگونه یک جواب را به صورت یک کروموزوم کد کنیم بطوریکه مطمئن شویم یک کروموزوم قابل قبول ایجاد کرده‌ایم. بخاطر وجود محدودیتهای پیچیده که در ذات مساله برنامه ریزی کارگاهی وجود دارد، کدگذاری باینری ساده در چنین مواردی اصلاً عملکرد خوبی نخواهد داشت و یقیناً چنین کدگذاریی منجر به جوابهای غیرقابل قبول و یا حتی غیرقانونی (شکل (۱۴)) می‌شود.

۲ـ چگونه از طریق ترکیب روشهاس سنتی ابتکاری می‌توان عملکرد الگوریتم را افزایش داد. چون الگوریتم ژنتیک روش خیلی مناسبی برای جستجو در اطراف نقطه بهینه نیست، در نتیجه روشهای ترکیبی متعددی، از قوانین تقدمی ساده گرفته تا الگوریتمهای پیچیده انشعاب و تحدید، به منظور جبران این کمبودها پیشنهاد شده‌اند.

الگوریتمهای ژنتیکی که برای مساله برنامه ریزی کارگاهی پیشنهاد شده‌اند را می‌توان به سه دسته تقسیم کرد[۸]:

- الگوریتمهای ژنتیکی تطبیقی[۳۸]
- الگوریتمهای ژنتیکی ابتکاری[۳۹]
- الگوریتمهای ژنتیکی ترکیبی[۴۰]

در این مقاله از روش سوم استفاده شده است. این انتخاب به دلیل عملکرد موفقیت آمیزتر این نوع الگوریتمها در حل مسائل بهینه‌سازی ترکیبی می‌باشد. مهمترین خصیصه یک الگوریتم ژنتیک عملگرهای جابجایی و جهشی آن می‌باشند. در این مقاله از عملگر جابجایی PMX و عملگر جهش جانشین‌سازی[۴۱] استفاده شده است. انتخاب این عملگرها براساس سعی و خطا انجام شده است و تقریباً عملکرد اکثر عملگرها مورد بررسی قرار گرفته است. نرخ عملگر جابجای در این مقاله P_c=٪۹۵ و نرخ

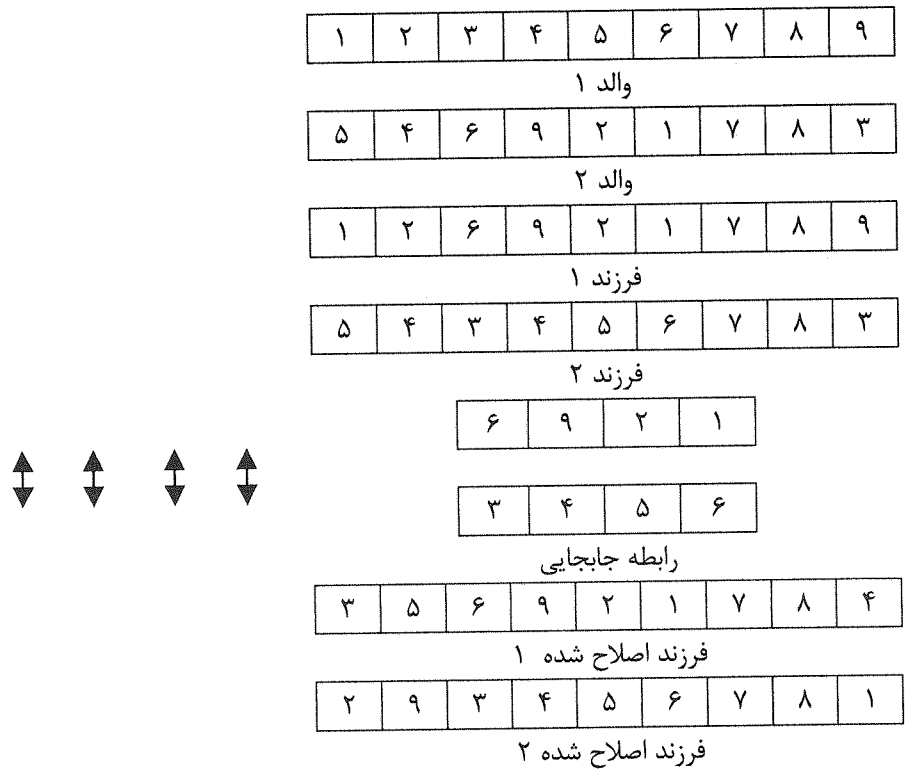
عملگر جهش P_m=٪۴ لحاظ شده است.

این اعداد نیز از روش سعی و خطا حاصل شده‌اند و ممکن است برای یک مساله بهینه‌سازی دیگر تغییر کنند.

۲-۲-۱ـ عملگر جابجایی PMX

در ایـن عملگـر پس از انتخاب دو والد، از هر کروموزوم والد دو نقطه را به تصادف انتخاب می‌کنیم. نقاط منتخب در دو والد از نظـر موقعیتـی بـاید یکسـان باشند. ابتدا زیر رشته انتخابی بین دو نقطه را در دو والد جابجا می‌کنیم. براساس این جابجایی، رابطه جابجایی را بدست آورده و برطبق این رابطه، فرزندان ایجاد شده را اصلاح می‌کنیم.

مثال:



شکل (۱۵) عملگر جابجایی PMX.

۲-۲-۲ـ عملگر جهش جانشین‌سازی

ایـن عملگـر بـرای هـر ژن کـروموزوم یک عدد تصادفی تولید می‌کند. در صورتیکه عدد تصادفی حاصل کمتر از نرخ جهش باشد آن ژن بصورت تصادفی به یکی از صورتهای ممکن دیگر تبدیل می‌شود.

۲-۳ـ داده کاوی اطلاعات[۱۰]

بـا توجـه بـه رشـد روز افـزون حجـم اطلاعـات و پایگاه‌های داده‌ها، انسان ناگزیر به استفاده از ابزارهای جدید برای تحلیل هوشـمندانه و کسب دانش مفید از این حجم وسیع از اطلاعات می‌باشد. فهم دانش[۴۲] در پایگاه داده‌ها و داده کاوی اطلاعات در پایگـاه داده‌ها جزء ابزارهای هوش مصنوعی می‌باشند که انسان را در تحلیل اطلاعات حجیم یاری می‌کنند. فهم دانش در پایگاه داده‌هـا بـه عـنوان یـک فرآیند اصلی به منظور تشخیص الگوهای معتبر، جدید، مفید و کاملاً قابل فهم می‌باشد. به عبارت دیگر فهم دانش شامل ارزیابی و تفسیر الگوها به منظور تشخیص آن چیزی است که یک دانش را شکل می‌دهد. داده کاوی اطلاعات بـه عنوان یک کاربرد از الگوریتمهای استنتاجی است که به منظور استخراج الگوها از داده‌ها در یک محدوده خاص بکار می‌رود. اکـثر الگوریـتمهای داده کاوی اطلاعات مشتق شده از روشهای یادگیری ماشین[۴۳]، تشخیص الگو[۴۴] و روشهای آماری می‌باشند.

این الگوریتمها شامل کلاس‌بندی[45]، گروه‌بندی[46] و مدلهای نموداری می‌باشند. اهداف اولیه در این الگوریتمها پیش‌بینی و توصیف است.

قبل از شروع کاوش در اطلاعات، الگوریتم کاوش باید تعیین و داده‌ها بطور مناسبی ساختار بندی شوند. در این مقاله الگوریتم استنتاج صفت‌گرا به عنوان الگوریتم کاوش انتخاب شده است.

۲-٤ـ استنتاج صفت‌گرا [۱۱]

استنتاج صفت‌گرا یک روش کاوش مجموعه‌گرا است که یک زیر مجموعه از کارهای مرتبط با هم را بر حسب صفاتشان تعمیم می‌دهد. این روش به منظور استخراج قوانین خصوصیات و قوانین کلاس‌بندی از پایگاه داده‌های رابطه‌ای از طریق بکارگیری سلسله مراتب مفاهیم در فرآیند استنتاج می‌باشد. الگوریتم استنتاج، مفاهیم سطح پایین یا همان داده‌های خام را با مفاهیم سطح بالاتر که تعمیمی از مفاهیم سطح پایینتر است جایگزین می‌کنند.

فرآیند تعمیم برای مفاهیم هر صفت دارای یک حد آستانه خاص می‌باشد. تعمیم مفاهیم باعث می‌شود که بعضی از داده‌ها دارای خصوصیات یکسان بشوند و از این رو داده‌های با خصوصیات یکسان و تکراری حذف می‌شوند. این کار باعث می‌شود که حجم اطلاعات مورد پردازش بطور قابل توجهی کاهش یابد.

یک سلسله مراتب مفاهیم در واقع یک توالی از روابطی است که میان مجموعه‌ای از مفاهیم و مفاهیم متناظر با آنها در سطوح بالاتر برقرار است. سلسله مراتب مفاهیم که ایجاد آن نیاز به داشتن دانش کافی نسبت به مساله مورد بررسی دارد، به عنوان بخش کلیدی فرآیند تعمیم در الگوریتم استنتاج صفت‌گرا می‌باشد. در یک سلسله مراتب مفاهیم، وقتی که از سطوح پایین به سمت سطوح بالا حرکت می‌کنیم، مفاهیم از حالت جزئی به سمت حالت کلی پیش می‌روند بطوریکه در بالاترین سطح کلمه "همه[6]" که بیانگر عمومی‌ترین حالت است را می‌بینیم.

۲-۵ـ داده کاوی اطلاعات و مساله برنامه‌ریزی کارگاهی

برای بررسی مساله برنامه ریزی کارگاهی بوسیله الگوریتم استنتاج صفت‌گرا اولین مرحله آماده‌سازی داده‌ها می‌باشد. در این مرحله قبل از هر چیز باید عوامل تأثیرگذار بر توالی عملیاتها در یک برنامه زمانی را تعیین کنیم. سپس باید با تعریف مفاهیم و تعیین تعداد سطوح در ساختار سلسله مراتبی هر صفت، مفاهیم را به سطوح بالاتر تعمیم دهیم. تمامی مراحل فوق‌الذکر باید توسط یک متخصص که دانش و شهود کافی نسبت به مساله برنامه‌ریزی کارگاهی دارد انجام شود.

در اینجا پنج عامل به عنوان عوامل تأثیرگذار بر توالی عملیاتها تعیین شده‌اند که عبارتند از:

- رتبه عملیات: عملیات jام کار iام چه رتبه‌ای در بین عملیاتهای کار iام دارد.
- زمان انجام عملیات
- زمان باقیمانده: عبارتست از جمع زمان انجام تمام عملیاتهای کار iام که بعد از عملیات j آن کار قرار دارند.
- بار هر ماشین: عبارتست از جمع زمان تمام عملیاتهایی که باید روی یک ماشین بدون در نظر گرفتن توالی خاصی انجام شوند.
- زمان تحویل: عبارتست از زمان تحویل کار iام

برای آشنایی با نحوه تعریف و تعمیم مفاهیم به یک مثال توجه کنید. البته اگر چه این مثال حالت فازی ندارد ولی در حالت فازی هم می‌توانیم تنها اعداد وسط را در نظر بگیریم. البته این معیار صرفاً یکی معیار ابتکاری دلخواه است و می‌توان آنرا به صورتهای دیگر نیز تغییر داد.

فرض کنید که یک مساله برنامه‌ریزی کارگاهی با ۶ کار و ۶ ماشین با مشخصات زیر موجود است:

زمان انجام =

کار۱: ۱ ۳ ۶ ۷ ۳ ۶

کار۲: ۸ ۵ ۱۰ ۱۰ ۱۰ ۴

کار۳: ۵ ۴ ۸ ۹ ۱ ۷

کار۴: ۵ ۵ ۵ ۳ ۸ ۹

کار۵: ۹ ۳ ۵ ۴ ۳ ۱

کار۶: ۳ ۳ ۹ ۱۰ ۴ ۱

ترتیب عملیات کارها بر روی ماشین‌ها:

کار۱: ۳ ۱ ۲ ۴ ۶ ۵

کار۲: ۲ ۳ ۵ ۶ ۱ ۴

کار۳: ۳ ۴ ۶ ۱ ۲ ۵

کار۴: ۲ ۱ ۳ ۴ ۵ ۶

کار۵: ۳ ۳ ۵ ۶ ۱ ۴

کار۶: ۲ ۴ ۶ ۱ ۵ ۳

زمان تحویل هر کار به ترتیب عبارتست از:

۳۰ و ۵۰ و ۳۵ و ۳۵ و ۲۵ و ۳۰

حال از روی این داده‌ها زمان باقیمانده هر عملیات و بار هر ماشین را محاسبه می‌کنیم.

بار ماشین‌ها به ترتیب عبارتست از:

۴۰ و ۲۶ و ۲۶ و ۲۲ و ۴۰ و ۴۳

زمان باقیمانده =

۲۵ ۲۲ ۱۶ ۹ ۶ ۰

۳۹ ۳۴ ۲۴ ۱۴ ۴ ۰

۲۹ ۲۵ ۱۷ ۸ ۷ ۰

۳۰ ۲۵ ۲۰ ۱۷ ۹ ۰

۱۶ ۱۳ ۸ ۴ ۱ ۰

۲۷ ۲۴ ۱۵ ۵ ۱ ۰

مشاهده می‌شود که اعداد ستون آخر همگی صفر است، این بخاطر این است که عملیاتهای آخر در هر کار پیش نیاز هیچ عملیاتی نیستند و بنابراین زمان باقیمانده برای آنها صفر می‌باشد.

تعیین رتبه عملیاتها هم کار بسیار ساده است که بصورت زیر است:

رتبه عملیاتها:

۱ ۲ ۳ ۴ ۵ ۶

۱ ۲ ۳ ۴ ۵ ۶

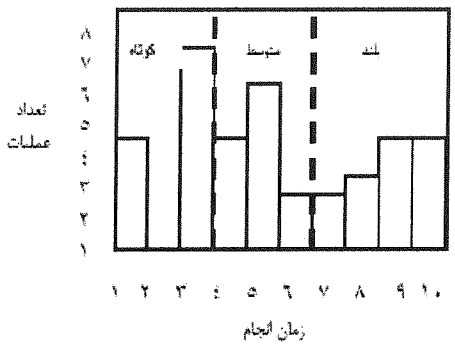
۱ ۲ ۳ ۴ ۵ ۶

۱ ۲ ۳ ۴ ۵ ۶

۱ ۲ ۳ ۴ ۵ ۶

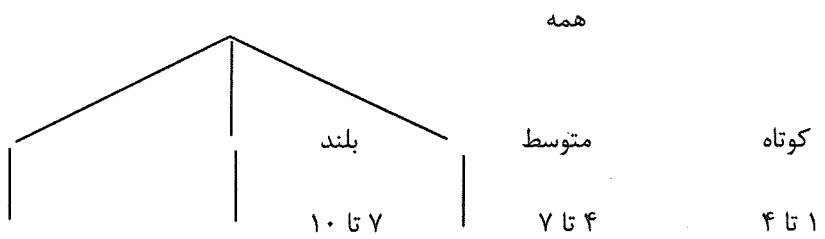
۱ ۲ ۳ ۴ ۵ ۶

حال با توجه به نمودارهای زیر می‌تواند شهود بهتری نسبت به مساله پیدا کرد.



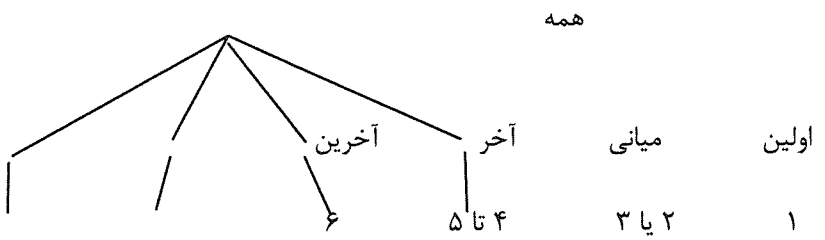
شکل (۱۶) تقسیم‌بندی زمان انجام عملیاتها.

آنچه که در نگاه اول از شکل (۱۶) می‌توان دریافت این است حداقل زمان انجام بین تمام عملیاتها برابر یک و حداکثر آنها ده می‌باشد. از طرفی دیگر این نمودار نشان‌دهنده این است که هر زمان انجام در بین عملیاتهای موجود چند بار تکرار شده است. با توجه به این نمودار ما زمان‌های انجام را به سه دسته کوتاه، متوسط و بلند تقسیم‌بندی کرده‌ایم. بنابراین اگر زمان انجام را یکی از صفات عملیاتها در نظر بگیریم "کوتاه"، "متوسط" و "بلند" سه مفهومی هستند که برای این صفت تعریف شده‌اند. در اینجا ما صفت "زمان انجام" و کلیه صفات دیگر که در ادامه مورد بررسی قرار می‌گیرند را فقط یک مرحله تعمیم داده‌ایم. نمودار تعمیم صفت "زمان انجام" بصورت زیر است:

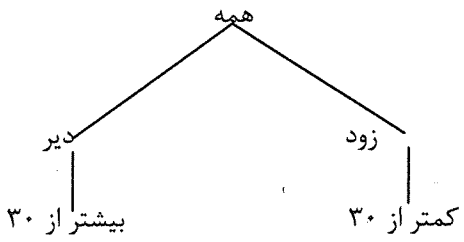


شکل (۱۷) سلسله مراتب مفاهیم صفت "زمان انجام".

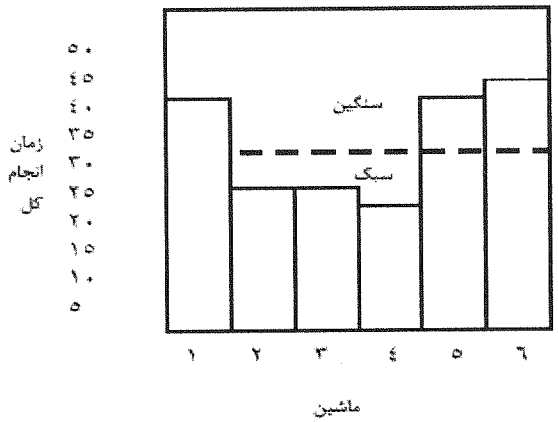
همین مراحل را برای صفات "رتبه عملیات"، "بار ماشین"، "زمان باقیمانده" و "زمان تحویل" نیز انجام می‌دهیم.



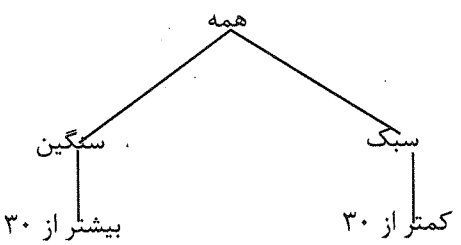
شکل (۱۸) سلسله مراتب مفاهیم صفت "رتبه عملیات".



شکل (۱۹) سلسله مراتب مفاهیم صفت "زمان تحویل".



شکل (۲۰) نمودار بار ماشین .



شکل (۲۱) سلسله مراتب مفاهیم صفت "بار ماشین".

اگـرچه تعـریف مفاهـیم و تعیین تعداد آنها یک کار اختیاری است ولی باید توجه داشت که چون اصل کار، جستجو در یک فضای بسیار پیچیده است، تعمیم زیاد ما را سر در گم می‌کند.

بـا توجه به مفاهیم تعریف شده یکسری عملیاتهای مشابه که از نظر فاکتورهای تعریف شده نزدیک به هم می‌باشند، در یک کلاس قرار می‌گیرند. در این حالت بجای سروکار داشتن با n*m حالت (تعداد کل عملیاتها) قطعاً با تعداد حالات کمتری مواجه هستیم. به عنوان مثال عملیات سوم از کار دوم و عملیات سوم از کار سوم را در نظر بگیرید:

رتبه عملیات = ۲ – زمان انجام = ۱۰ – زمان باقیمانده

= ۲۴ – بار ماشین = ۴۰ – زمان تحویل = ۵۰

رتبه عملیات = ۳ – زمان انجام = ۸ – زمان باقیمانده

= ۱۷ – بار ماشین = ۴۵ – زمان تحویل = ۳۵

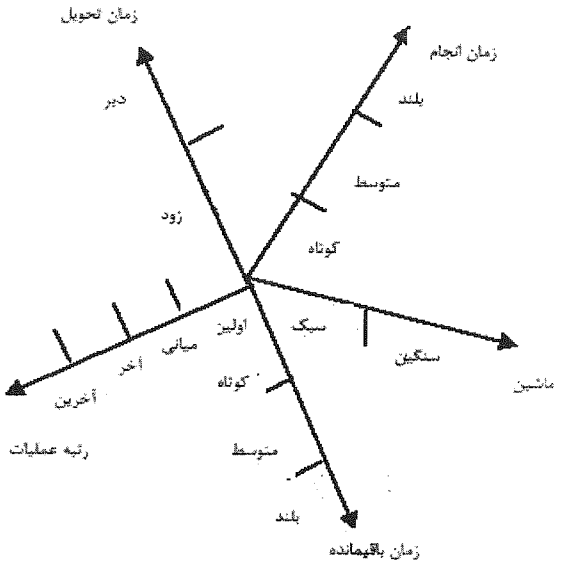
وضعیت دو عملیات فوق بعد از تعمیم:

رتبه عملیاتها = میانی – زمان انجام آنها = بلند

زمان باقیمانده = متوسط – بار ماشین = سنگین

زمان تحویل آنها = دیر

مشـاهده مـی‌شـود کـه ایـن دو عملـیات در یک کلاس قرار می‌گیرند. به طور مشابه برای همه عملیاتها، خصوصیات آنها را براسـاس مفاهیم تعریف شده مشخص می‌کنیم. با قرار دادن عملیاتهای مشابه در یک کلاس، تعداد کلاسها مشخص می‌شود. در بدتریـن حالـت تعـداد کلاسـها برابر تعداد عملیاتها یعنی n*m خواهد بود. حال برای تمام برنامه‌های نسل جاری برحسب اینکه عملـیات چـه کلاسـی دارد مشـخص می‌کنیم که هر عملیات چند بار در چه رده‌ای (بین ۱ تا n) روی ماشین مربوطه اش قرار گرفته است. برای ثبت این اطلاعات می‌توان از ساختار حجمی شکل (۲۲) کمک گرفت.



شکل (۲۲) ساختار حجمی داده‌ها براساس سلسله مراتب مفاهیم برای مساله برنامه ریزی کارگاهی.

۳ـ الگوریتم حل مساله

۳-۱ـ مفهوم شباهت

در ایـن قسـمت بـا تعـریف مفهومی بنام شباهت، مقدمات ایجاد یک جمعیت اولیه متنوع را ایجاد می‌کنیم. برای آشنایی و فهم بهتر این مفهوم به مثال زیر توجه کنید:

فـرض کنـید در یک مساله برنامه‌ریی کارگاهی با ۴ کار و ۴ ماشین دو برنامه زیر دو حالت ممکن از مجموعه حالات ممکن برای توالی عملیاتها باشند.

جدول (۱) برنامه ۱.

	کار ۱	کار ۲	کار ۳	کار ۴
ماشین ۱	۱	۲	۴	۳
ماشین ۲	۳	۱	۴	۲
ماشین ۳	۴	۳	۱	۲
ماشین ۴	۳	۲	۱	۴

جدول (۲) برنامه ۲.

	کار ۱	کار ۲	کار ۳	کار ۴
ماشین ۱	۲	۱	۴	۳
ماشین ۲	۳	۱	۴	۲
ماشین ۳	۴	۳	۱	۲
ماشین ۴	۳	۲	۴	۱

اگر بخواهیم درجه شباهت بین این دو برنامه را محاسبه کنیم به طریق زیر عمل می‌کنیم:

سـطر یک هر دو برنامه که توالی عملیاتها بر روی ماشین ۱ را نشان می‌دهند در نظر بگیرید. مشاهده می‌کنید که در برنامه ۱، در ایـن سطر، کار شماره ۱ به عنوان اولین کار برنامه‌ریزی شده است. پس می‌توان گفت که روی ماشین ۱، کار ۱ در بین ۴ کـار موجـود رتـبه اول را دارد. حـال کار شماره ۱ را روی ماشین ۱ برنامه ۲ در نظر بگیرید. مشاهده می‌کنید که کار ۱ در این بـرنامه بـه عنوان دومین کار برنامه‌ریزی شده است. کار ۱ روی ماشین ۱ در برنامه ۲ از این حیث که نسبت به کار ۳ و ۴ زودتر

برنامه‌ریزی شده است، مشابه کار ۱ روی ماشین ۱ در برنامه ۱ است، ولی از این حیث که نسبت به کار ۲ عقب‌تر است با کار ۱ روی ماشین ۱ برنامه ۱ متفاوت است. بنابراین عدد شباهت برای این دو عملیات برابر ۲ می‌باشد. حال اگر کار ۴ را روی ماشین ۱ برنامه ۱ و ۲ در نظر بگیرید طبق روش بالا مشاهده می‌شود که هر دو عملیات در موقعیت یکسانی قرار دارند، هر دو از کار ۱ و ۲ عقب‌تر و از کار ۳ جلوتر هستند. بنابراین عدد شباهت برای این عملیات به حداکثر خود یعنی ۳=۱- تعداد کارها می‌رسد. برای تمامی عملیاتها این کار را انجام می‌دهیم. برای مثال فوق محاسبات به اینگونه است:

ماشین ۱: ۱۰=۳+۳+۲+۲

ماشین ۲: ۱۲= ۳+۳+۳+۳

ماشین ۳: ۱۲= ۳+۳+۳+۳

ماشین ۴: ۱۰= ۲+۲+۳+۳

پس از اینکه جمع هر سطر را محاسبه کردیم، درجه شباهت دو برنامه را بصورت تقسیم جمع کل اعداد شباهت بر حداکثر شباهت ممکن ((۱ـ تعداد کارها)* تعداد کارها * تعداد ماشینها) محاسبه می‌کنیم. درجه شباهت دو برنامه ۱ و ۲: ۰/۹۱۷ = ۴۸/(۱۰+۱۲+۱۲+۱۰) است.

درجه شباهت برای دو برنامه کاملاً مشابه از نظر توالی "یک" است. اینکه عدد شباهت را برای ایجاد برنامه‌های متنوع چند در نظر بگیریم نیاز به سعی و خطا دارد. معمولاً درجه شباهت کمتر از ۰/۶ به ندرت اتفاق می‌افتد، مخصوصاً در برنامه‌هایی با اندازه کوچک. در این مقاله عدد ۰/۸ برای تولید یک نسل اولیه متنوع در نظر گرفته شده است.

۳-۲ـ ساختار کلی الگوریتم

ساختار کلی الگوریتم طراحی شده را می‌توان بصورت زیر تشریح کرد:

گام ۱ـ تولید نسل اولیه

۱ـ۱ـ حداکثر درجه شباهت قابل قبول = ۰/۸

۱ـ۲ـ به تعداد اندازه جمعیت اولیه برنامه زمانی تولید کنید. این برنامه‌ها باید با در نظر گرفتن شرط درجه شباهت، یعنی اینکه شباهت هیچ دو برنامه‌ای در این نسل از حداکثر مجاز بیشتر نشود، ایجاد شوند. این مجموعه برنامه زمانی را "جمعیت اولیه" می‌نامیم.

گام ۲ـ کلاس‌بندی عملیاتها

۲ـ۱ـ از روی داده‌های اولیه مساله، موارد زیر را مشخص کنید.

ـ زمان باقیمانده

ـ بار ماشین

۲ـ۲ـ مشخص کردن پارامترهایی که ممکن است در تعیین توالی عملیاتها تأثیر گذار باشند.

۲-۳ـ تعریف مفاهیم و تعیین تعداد سطوح برای هر یک از مفاهیم مرحله قبل

۲-۴ـ ایجاد یک حجم با ابعادی برابر تعداد مفاهیم تعریف شده

۲-۵ـ ایجاد ماتریس خصوصیات براساس حجم ایجاد شده. ماتریس خصوصیات مشخص می‌کند که هر عملیات در هر فاکتور مورد نظر براساس مفاهیم تعریف شده چه خصوصیت یا وضعیتی دارد. به عنوان مثال، خصوصیت زمان تحویل عملیاتی که زمان تحویل آن ۵۰ می‌باشد، خصوصیت "دیر" است.

۲-۶ـ کلاس‌بندی عملیاتها براساس مفاهیم و ماتریس خصوصیات و براساس توضیحات بخش ۳-۵

گام ۳ـ تعریف ماتریس کلاس، تعیین ابعاد آن (تعداد کارها * تعداد کلاسها) و تعیین درصدها با توجه به جوابهای حاصل از نسل اولیه. هر عدد در بازه [۱ و ۰] و در سطر pام و ستون iام ماتریس کلاس به این مفهوم است که چند درصد عملیاتهای برنامه‌های حاصل از نسل اول که در کلاس p قرار دارند، رتبه iام را در توالی ماشین مربوطه اشان دارا می‌باشند.

به منظور تعیین درصدها برای هر عملیات p از هر برنامه زمانی جمعیت اولیه مشخص می‌کنیم که این عملیات به عنوان

چندمیـن عملـیات روی ماشین مربوطه برنامه‌ریزی شده است. اگر این رتبه، رتبه i ($1 \le i \le n$) باشد، با توجه به اینکه کلاس عملـیات نـیز مشـخص است، یک واحد به تعداد عملیاتهایی که در کلاس p قرار دارند و رتبه i در توالی مربوطه اشان داشته‌اند اضافه کنید.

این کار را برای تمام عملیاتهای برنامه‌های نسل اولیه انجام می‌دهیم.

گـام ۴ـ پـس از تکمـیل ماتریس کلاس، عنصر سطر pام و ستون iام ماتریس کلاس را بر جمع مقادیر عناصر سطر pام ماتریس کلاس تقسیم می‌کنیم. عددهای حاصل همان درصدها می‌باشند.

گـام ۵ـ ماتـریس کـروموزوم را تشـکیل دهـید. تعداد سطرهای این ماتریس به اندازه جمعیت هر نسل و تعداد ستونهای آن به اندازه تعداد کارها می‌باشد. هر سطر از این ماتریس را با اعداد تصادفی بین ۱ تا n پر کنید.

گام ۶ـ تولید نسل جدید

۶-۱ـ تولید یک برنامه زمانی فعال فازی.

بـرنامه زمانـی فعـال در حالـت فازی مانند برنامه زمانی در حالت عادی است با این تفاوت که زمان انجام عملیاتها بصورت اعداد فازی مثلثی می‌باشند. الگوریتم تولید برنامه زمانی فعال در این حالت به شرح زیر است:

۶-۱-۱ـ مجموعه C را مجموعه تمام عملیاتهایی قرار دهید که پیش نیازهایشان انجام شده‌اند.

۶-۱-۲ـ از بیـن عملـیاتهای مجموعـه C، عملیاتـی را کـه دارای کوچکتریـن زمـان تکمیل می‌باشد انتخاب کنید. زمان تکمیل عملیات O_{ijr} عبارتست از جمع زمان انجام این عملیات با حداکثر زمان ۱ و زمان ۲

زمان ۱: حداکثر زمانی که تا این لحظه بر روی ماشین r برنامه‌ریزی شده است.

زمان ۲: زمان تکمیل آخرین عملیات مقدم بر این عملیات (O_{ijr} عبارتست از jامین عملیات کار iام روی ماشین rام).

۶-۱-۳ـ پـس از انـتخاب عملیات با کوچکترین زمان تکمیل، از بین عملیاتهای مجموعه C تمام عملیاتهایی که ماشین مربوطه اشان با ماشین مربوطه عملیات منتخب یکی است (m*) را در مجموعه‌ای بنام "مجموعه منتخب" قرار دهید.

(لازم بـه توضـیح اسـت کـه در تمـام مراحل الگوریتم پیشنهادی وقتی که نیاز به مقایسه بین چند عدد فازی می‌باشد از روش ترتیب اعداد فازی استفاده می‌شود)

۶-۱-۴ـ به منظور انتخاب یکی از عملیاتهای مجموعه منتخب به صورت زیر عمل می‌کنیم:

۶-۱-۴-۱ـ در این مرحله که ناچار به انتخاب بین چند عملیات هستیم به ماتریس کلاس رجوع می‌کنیم. فرض کنید قرار است که از بین s عملیات کاندیدا یکی را انتخاب و در موقعیت iام ماشین انتخاب شده در قدم ۲ تولید برنامه زمانی فعال (گام ۶-۱-۲) قـرار دهـیم. از بیـن عملـیاتها آنهایی را انتخاب می‌کنیم که با توجه به کلاس مربوطه‌اشان، عدد موجود در موقعیت iام آنها رتبه اول یا دوم را در بین اعداد آن کلاس داشته باشد.

۶-۱-۴-۲ـ اگـر بـاز هم چند عملیات دارای شرایط مرحله قبل بودند براساس اینکه در حال برنامه‌ریزی چندمین برنامه از نسل جاری هستم به سطر مربوطه از ماتریس کروموزوم رجوع می‌کنیم.

پـس از انـتخاب سـطر بـاید بـا مشخص کردن ستون مربوطه، عملیات داور را مشخص کنیم. برای مشخص کردن ستون، فرض کنـید k عملـیات روی ماشـین منتخـب (ماشـینی کـه تمام عملیاتهای کاندیدا مربوط به آن می‌باشند) برنامه‌ریزی شده است. ستون مطلوب، ستون k+۱ می‌باشد.

۶-۱-۴-۳ـ به زیر برنامه مربوط به قانون داور رجوع کرده و یکی از عملیاتها را براساس آن قانون انتخاب کنید.

قوانین داوری که در این مقاله تعریف شده‌اند به بصورت زیر می‌باشند. از بین عملیاتهای کاندیدا عملیاتی انتخاب می‌شود که به ترتیب دارای موارد زیر باشند.

الـف ـ حداقـل زمـان انجـام، ب ـ حداکثر زمان انجام، ج ـ حداقل زمان تحویل، د ـ کمترین زمان باقیمانده، ه ـ بیشترین زمان باقیمانده، و ـ حداقل کار باقیمانده، ی ـ انتخاب تصادفی

۶-۱-۴-۴ـ تا اتمام کل عملیاتها به گام (۶-۱) برگردید.

۶-۱-۴-۵ـ تا اتمام کل برنامه‌های یک نسل به گام ۶ برگردید.

۶-۱-۵ـ عملـیات انتخاب شده را از مجموعه C حذف کرده و عملیات بعدی کاری که عملیات منتخب جزئی از آن بود (کار iام)

را به مجموعه C اضافه کنید.

گام ۷ـ مانند گام ۳ با تشکیل ماتریس به نام "ماتریس سوابق"، سوابق تمام عملیاتها در برنامه‌های مختلف نسل جاری را در این ماتریس قرار دهید. سپس مشابه گام ۴ درصدها را محاسبه کنید.

گام ۸ـ هر درایه ماتریس جدید کلاس برابر است با درایه میانگین مقدار درایه قبلی و درایه نظیر در ماتریس سوابق.

گام ۹ـ مقدار تابع هدف را برای تمام برنامه‌های نسل جاری محاسبه کنید.

گام ۱۰ـ برحسب اینکه چندمین برنامه بهترین عملکرد را داشته است، شماره سطر کروموزوم مربوطه را نگاه داشته و برای تولید کروموزوم جدید که این سطر نیز جزئی از آن است به گام بعد بروید.

گام ۱۱ـ برحسب میزان تابع هدف در هر برنامه "ماتریس انتخاب" را ایجاد کنید. در ستون ۱ این ماتریس مقدار توابع هدف نسل قبل بجز بهترین برنامه را قرار داده (چون کروموزوم بهترین برنامه مستقیماً وارد نسل بعدی می‌شود) و در ستون ۲ احتمال تجمعی انتخاب آنها را قرار می‌دهیم. برنامه‌هایی که عملکرد بهتری داشته‌اند شانس انتخاب بیشتری بعنوان والد دارند.

گام ۱۲ـ تولید کروموزومهای جدید

۱۲-۱ـ هر بار با تولید ۲ عدد تصادفی و با توجه به اینکه این اعداد در چه رده‌ای از ستون ۲ ماتریس انتخاب قرار بگیرند، ۲ تا از سطرهای ماتریس کروموزوم قبلی را به عنوان کروموزومهای والد انتخاب کنید.

۱۲-۲ـ با بکار بردن عملگر جابجایی PMX و سپس عملگر جهش دو فرزند تولید کنید.

۱۲-۳ـ تا تکمیل ماتریس کروموزوم جدید به گام ۱۱ برگردید.

گام ۱۳ـ سوابق نسل جدید را همراه مقادیر تابع هدف هر برنامه ثبت کنید.

گام ۱۴ـ تا برقراری شرط توقف به گام ۶ برگردید. (تولید ۱۰۰ نسل و جمعیت هر نسل برابر ۲۰) فلوچارتهای مربوط به روش تولید برنامه زمانی فعال، روش ایجاد نسل اولیه و ساختار کلی الگوریتم در شکلهای (۲۶)، (۲۷) و (۲۸) آمده‌اند.

۳-۳ـ روش شبیه‌سازی بازپخت

در این قسمت به منظور مقایسه روش پیشنهادی با یکی دیگر از روشهای ابتکاری در حل مسائل بهینه‌سازی ترکیبی، از روش شبیه‌سازی بازپخت جهت حل مسائل نمونه استفاده می‌شود. گامهای کلی این روش در زیر آمده است:

گام ۱ـ ایجاد یک جواب اولیه X^C (تولید یک برنامه زمانی فعال). X^C بهترین جواب تکرار جاری است. دمای اولیه را $T=t_0$ در نظر بگیرید.

گام ۲ـ توالی انجام عملیاتها بر روی ماشینها را بر اساس X^C بصورت ماترسی (ماتریس A) نمایش دهید.

گام ۳ـ از ماتریس A یک ماشین خاص را بطور تصادفی انتخاب کنید. در این ماشین، دو عملیات را بطور دلخواه انتخاب کرده و توالی آنها را جابجا کنید.

گام ۴ـ براساس تغییرات اعمال شده در گام ۳، یک جواب موجه بصورت یک برنامه زمانی فعال ایجاد کرده و آن را X بنامید.

گام ۵ـ اگر حداقل مقدار AI مربوط به جواب X بزرگتر از مقدار AI مربوط به جواب X^C است، قرار دهید $X^C=X$.

گام ۶ـ اگر حداقل مقدار AI مربوط به جواب X کوچکتر از مقدار AI مربوط به جواب X^C است، با در نظر گرفتن احتمال قبولی P، قرار دهید $X^C=X$.

گام ۷ـ شماره تکرار و دما را بروز کنید.

گام ۸ـ اگر شماره تکرار به حد مشخص شده رسیده است، توقف کنید و به یک جواب بهینه تقریبی رسیده‌اید، در غیر اینصورت به گام ۲ باز گردید.

٤ـ نتایج شبیه‌سازی و مقایسه

در این قسمت سه مساله نمونه را توسط الگوریتم پیشنهادی حل کرده و جوابهای حاصل را با الگوریتم ژنتیک عادی با پارامترهای مشابه و همچنین روش شبیه‌سازی بازپخت مقایسه می‌کنیم.

٤-١ـ مساله نمونه ١ [١٢]

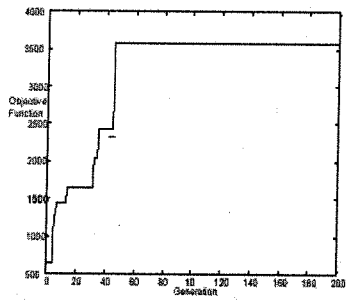
یـک مساله برنامه‌ریزی کارگاهی فازی با ۶ کار و ۶ ماشین را که اطلاعات اولیه آن در جدول (۳) آمده است، در نظر بگیرید. توالـی بهیـنه انجـام عملـیاتها روی ماشـینها طبق جدول (۴) می‌باشد. مقدار بهینه ۳۵۸۰ در تکرار ۴۷ برای این مساله بدست می‌آید (شکل ۲۳).

جدول (۳) اطلاعات مساله نمونه ۱ .

زمان تحویل	(زمان انجام عملیات) توالی						کار
(۸۸و۸۱و۷۰)	۱(۱۱و۱۰و۹)	۲(۸و۶و۴)	۳(۵و۳و۱)	۴(۵و۳و۱)	۵(۱۷و۱۴و۱۰)	۶(۱۰و۷و۵)	۱
(۸۰و۶۶و۵۰)	۲(۴و۳و۲)	۴(۱۶و۱۳و۱۰)	۶(۴و۳و۲)	۳(۱۳و۱۲و۸)	۱(۱۷و۱۳و۹)	۵(۸و۷و۶)	۲
(۹۲و۸۹و۸۵)	۴(۹و۷و۴)	۶(۱۲و۹و۶)	۲(۱۳و۱۲و۸)	۵(۱۶و۱۲و۹)	۱(۱۲و۱۱و۱۰)	۳(۶و۵و۴)	۳
(۶۰و۵۱و۴۰)	۲(۱۰و۸و۶)	۱(۷و۵و۳)	۳(۱۰و۸و۵)	۶(۸و۷و۵)	۵(۵و۴و۲)	۴(۴و۲و۱)	۴
(۹۶و۹۱و۸۳)	۳(۱۲و۱۱و۱۰)	۲(۸و۷و۴)	۶(۱۵و۱۱و۱۰)	۵(۳و۲و۱)	۱(۹و۶و۴)	۴(۱۵و۱۱و۹)	۵
(۷۸و۷۵و۷۰)	۱(۱۴و۱۳و۹)	۴(۳و۲و۱)	۶(۱۸و۱۴و۱۰)	۲(۱۱و۹و۶)	۳(۱و۲و۴)	۵(۹و۷و۶)	۶

جدول (٤) توالی بهینه مساله نمونه ۱.

ماشین ۱	۳	۵	۴	۲	۱	۶
ماشین ۲	۳	۱	۴	۵	۲	۶
ماشین ۳	۳	۴	۱	۲	۵	۶
ماشین ۴	۵	۴	۱	۲	۳	۶
ماشین ۵	۲	۴	۱	۵	۳	۶
ماشین ۶	۱	۴	۵	۲	۳	۶



شکل (۲۳) جواب بهینه مساله نمونه ۱.

٤-٢ـ مساله نمونه ٢

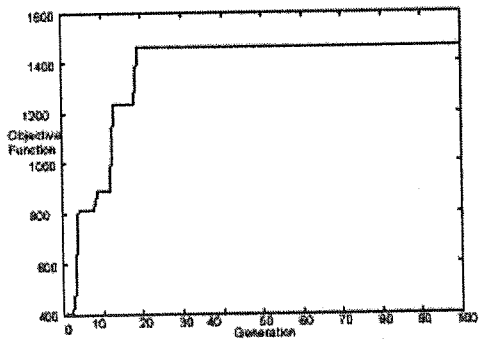
ایـن مسـاله نـیز یـک مسـاله بـرنامه‌ریزی کارگاهی فازی با ۶ کار و ۶ ماشین می‌باشد. جدول (۵) حاوی اطلاعات اولیه این مسـاله اسـت. توالـی بهیـنه انجـام عملـیاتها روی هـر ماشین طبق جدول (۶) می‌باشد. مقدار بهینه ۱۴۶۱ در تکرار ۲۰ بدست می‌آید (شکل ۲۴).

جدول (۵) اطلاعات مساله نمونه ۲ .

زمان تحویل	(زمان انجام عملیات) توالی						کار
(۱۲۱و۱۱۲و۱۰۰)	۱(۱۵و۱۱و۹)	۲(۱۷و۱۴و۱۰)	۵(۱۱و۸و۵)	۱(۱۳و۱۱و۱۰)	۳(۱۲و۹و۶)	۴(۱۷و۱۳و۹)	۱
(۷۰و۸۲و۹۱)	۶(۱۰و۷و۴)	۱(۱۷و۱۴و۱۰)	۳(۶و۵و۳)	۵(۵و۴و۳)	۲(۱۰و۸و۷)	۴(۹و۸و۵)	۲
(۶۰و۴۹و۳۵)	۶(۴و۳و۱)	۲(۶و۵و۳)	۱(۱۱و۸و۵)	۳(۶و۴و۲)	۴(۵و۴و۳)	۵(۶و۵و۳)	۳
(۱۰۲و۹۷و۹۰)	۵(۷و۵و۳)	۲(۱۳و۱۲و۱۰)	۴(۱۳و۱۲و۸)	۱(۱۷و۱۳و۹)	۳(۱۰و۸و۵)	۶(۱۴و۱۱و۸)	۴
(۸۹و۸۳و۷۵)	۴(۹و۷و۴)	۱(۷و۵و۳)	۲(۸و۶و۴)	۶(۱۷و۱۳و۱۰)	۵(۱۱و۹و۶)	۳(۱۳و۱۲و۸)	۵
(۵۹و۵۴و۵۰)	۱(۶و۴و۲)	۵(۵و۴و۳)	۳(۴و۳و۱)	۶(۱۲و۹و۶)	۴(۱۰و۹و۸)	۲(۱۳و۱۰و۸)	۶

جدول (٦) توالی بهینه مساله نمونه ۱.

ماشین ۱	۳	۱	۴	۲	۵	۶
ماشین ۲	۶	۲	۳	۵	۱	۴
ماشین ۳	۵	۳	۴	۱	۶	۲
ماشین ۴	۱	۳	۶	۲	۴	۵
ماشین ۵	۳	۵	۲	۱	۶	۴
ماشین ۶	۴	۶	۵	۳	۱	۲



شکل (۲٤) جواب بهینه مساله نمونه ۲.

٤-۳ـ مساله نمونه ۳

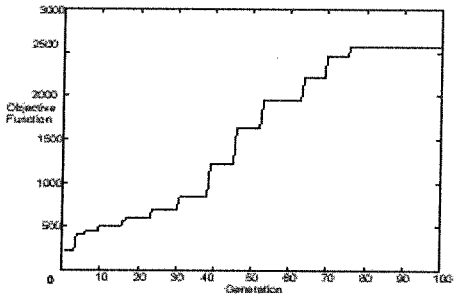
ایـن مساله نمونه، یک مساله برنامه‌ریزی کارگاهی فازی با ابعاد ۱۰×۱۰ می‌باشد. اطلاعات مربوط به این مساله در جـــدول (۷) و توالـی بهیـنه انجـام عملیاتها روی هر ماشین طبق جدول (۸) آمده است. شکل (۲۴) نشان دهنده این است الگوریتم در اجرای ۷۵ موفق به کسب جواب بهینه با مقدار ۲۵۸۳ شده است.

جدول (۷) توالی بهینه مساله نمونه ۱.

زمان تحویل	(زمان انجام عملیات) توالی						
(۱۶۳و۱۶۰و۱۵۵)	(۱۸و۱۴و۱۰) ۱۰	(۵و۳و۲) ۱	(۷و۶و۵) ۳	(۱۵و۱۲و۱۰) ۵	(۱۲و۷و۸) ۲	(۹و۸و۶) ۹	(۷و۶و۵) ۸
(۱۶۳و۱۶۰و۱۵۵)	(۱۰و۷و۴) ۴	(۶و۴و۲) ۸	(۱۶و۱۲و۸) ۹	(۱۴و۱۳و۹) ۵	(۱۲و۱۱و۷) ۷	(۹و۶و۵) ۶	(۱۶و۱۲و۹) ۰
(۱۶۳و۱۶۰و۱۵۵)	(۱۳و۱۱و۹) ۶	(۹و۷و۵) ۱	(۴و۳و۱) ۳	(۶و۴و۳) ۸	(۷و۵و۳) ۲	(۸و۷و۴) ۵	(۶و۴و۳) ۴
(۱۶۳و۱۶۰و۱۵۵)	(۵و۳و۲) ۳	(۱۰و۷و۴) ۲	(۸و۷و۴) ۹	(۱۴و۱۰و۸) ۸	(۱۰و۸و۵) ۱	(۱۱و۸و۷) ۴	(۵و۳و۱) ۶
(۱۶۳و۱۶۰و۱۵۵)	(۳و۲و۱) ۳	(۱۳و۱۱و۹) ۸	(۶و۴و۳) ۹	(۱۰و۸و۷) ۷	(۵و۳و۴) ۴	(۵و۴و۲) ۲	(۹و۷و۵) ۱
(۱۶۳و۱۶۰و۱۵۵)	(۱۰و۷و۴) ۷	(۹و۸و۵) ۱۰	(۱۲و۹و۸) ۶	(۱۵و۱۲و۱۰) ۹	(۱۶و۱۴و۵) ۵	(۹و۶و۵) ۱	(۱۸و۱۴و۱۰)
(۱۶۳و۱۶۰و۱۵۵)	(۱۳و۱۲و۹) ۴	(۱۵و۱۳و۱۰) ۳	(۱۲و۱۱و۱۰) ۲	(۱۱و۱۰و۷) ۷	(۷و۵و۴) ۹	(۸و۵و۴) ۶	(۹و۷و۵) ۱
(۱۶۳و۱۶۰و۱۵۵)	(۱۳و۹و۷) ۷	(۱۳و۱۱و۸) ۱۰	(۱۱و۹و۶) ۵	(۱۳و۱۱و۷) ۹	(۱۴و۱۱و۷) ۱	(۹و۶و۴) ۴	(۱۲و۹و۶) ۸
(۱۶۳و۱۶۰و۱۵۵)	(۶و۵و۳) ۵	(۳و۲و۱) ۸	(۱۴و۱۱و۸) ۷	(۹و۷و۴) ۱	(۱۴و۱۲و۸) ۳	(۹و۸و۶) ۱۰	(۷و۶و۴) ۴
(۱۶۳و۱۶۰و۱۵۵)	(۱۵و۱۱و۸) ۶	(۶و۴و۳) ۱۰	(۷و۵و۳) ۲	(۱۱و۱۰و۴) ۴	(۱۴و۱۱و۱۰) ۵	(۱۳و۱۱و۷) ۹	(۱۲و۹و۶) ۱

جدول (۸) توالی بهینه مساله نمونه۳.

ماشین ۱	۲	۷	۶	۱۰	۴	۹	۵	۳	۱	۸
ماشین ۲	۲	۶	۹	۸	۳	۱	۷	۵	۱۰	۴
ماشین۳	۶	۲	۱۰	۹	۸	۳	۱	۷	۵	۴
ماشین۴	۱	۹	۶	۳	۴	۵	۱۰	۸	۷	۲
ماشین ۵	۷	۴	۵	۶	۳	۱	۲	۱۰	۹	۸
ماشین ۶	۹	۱	۵	۴	۲	۷	۸	۶	۳	۱۰
ماشین ۷	۱۰	۳	۴	۱	۲	۷	۹	۵	۶	۸
ماشین ۸	۶	۱۰	۷	۱	۳	۴	۸	۹	۲	۵
ماشین ۹	۹	۳	۱	۷	۱۰	۶	۴	۵	۲	۸
ماشین ۱۰	۷	۳	۲۰	۴	۹	۵	۱۰	۶	۱	۸



شکل (۲۵) حداکثر تابع هدف در هر نسل.

۵ـ مقایسه نتایج و نتیجه‌گیری

جدول (۹) مقایسه‌ای است بین نتایج حاصل از اجرای الگوریتم برای سه مساله نمونه ذکر شده با روش شبیه‌سازی بازپخت و همچنین الگوریتم ژنتیک عادی.

جدول (۹) مقایسه نتایج در روشهای مختلف.

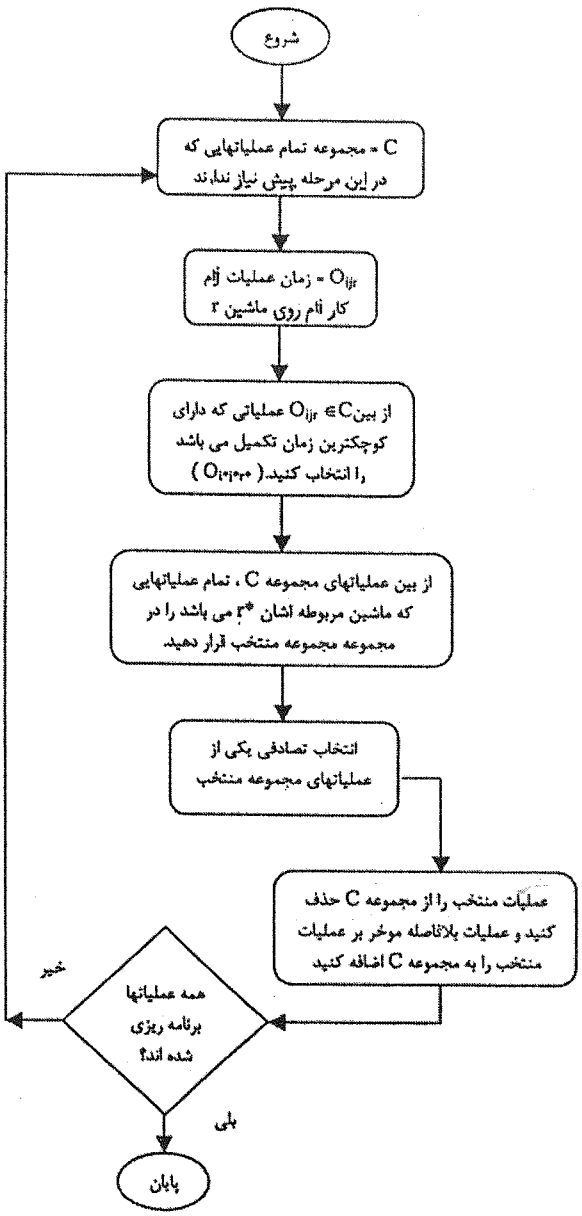
روش / مساله	الگوریتم پیشنهادی (تکرار) مقدار	الگوریتم شبیه‌سازی بازپخت (تکرار) مقدار	الگوریتم ژنتیک عادی (تکرار) مقدار
۱	۳۵۸۰(۴۷)	۳۵۱۲ (۱۱۷)	۳۵۷۲(۶۹)
۲	۱۴۶۱(۲۰)	۱۴۱۸(۸۹)	۱۴۲۸/۵(۴۹)
۳	(۷۵)		

در اینجا باید توجه داشت که تعداد جمعیت در هر نسل برای الگوریتم ژنتیک عادی نیز ۲۰ می‌باشد. منظور از الگوریتم ژنتیک عادی، الگوریتم ژنتیک با ساختار کلی و با عملگر جابجایی PMX و عملگر جهش جانشین‌سازی و با مقادیر P_c و P_m مشابه با الگوریتم پیشنهادی می‌باشد. در اینجا فاکتورهای مقایسه روشها، تعداد تکرارها و مقادیر بهینه جواب در هر مورد می‌باشند. البته باید توجه داشت که هیچ تضمینی در مورد اینکه جوابهای حاصل، جواب بهینه کلی هستند وجود ندارد. یکی دیگر از فاکتورهایی مهم که باید مورد نظر قرار بگیرد زمان حل برای هر یک زا این روشها می‌باشد. البته این فاکتور تنها زمانی صحیح است که تمامی این روشها با یک نوع کامپیوتر مشابه اجرا شوند. در هر صورت، روش پیشنهادی به دلیل محاسبات حجیم در هر مرحله نسبت به روش ژنتیک عادی و روش شبیه‌سازی باز پخت زمان بیشتری می‌برد تا به جواب بهینه یا نزدیک به بهینه برسد. ولی آنچه که مهمتر است این است که جوابهای حاصل از این روش نسبت به روشهای مذکور بهتر و قطعاً به جواب بهینه کلی نزدیکترند. با توجه به فاکتور زمان، بهتر است در مورد مسائل با ابعاد کوچک، از همان روشهای عادی استفاده شود چون الگوریتم پیشنهادی زمان بیشتری را برای رسیدن به جواب بهینه صرف می‌کند اما در مورد مسائل با ابعاد بزرگ که گاه رسیدن به جواب بهینه در آنها بسیار سخت می‌باشد، بکارگیری این الگوریتم توصیه می‌شود.

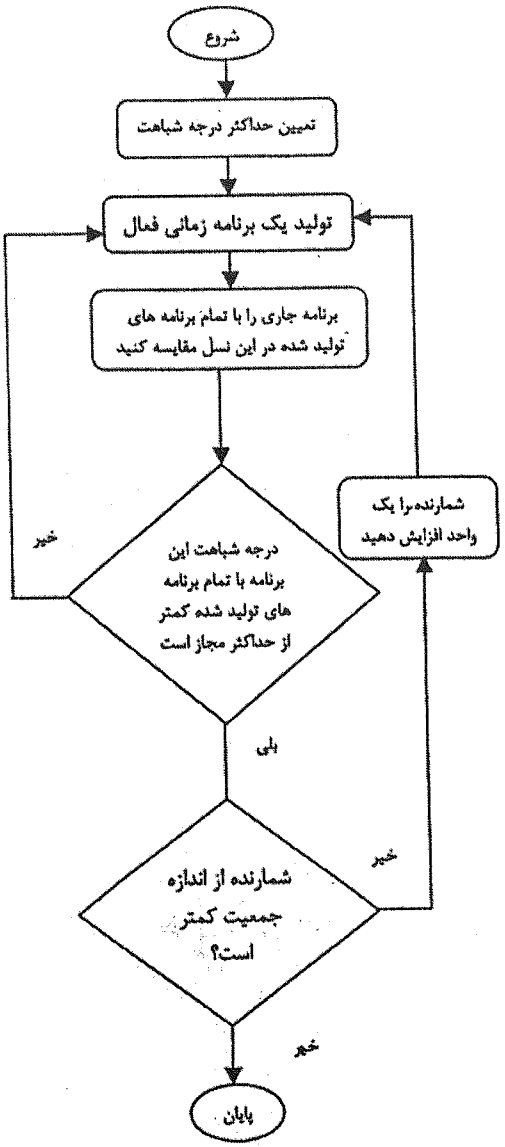
زیرنویس‌ها

1-Efficient
2- Fuzzy Job Shop Scheduling
3- Data mining
4- Combinatrial optimization
5- Genetic Algorithm
6- Attribute – Oriented induction
7- Dacis
8- Shifting Bottlenech
9- Simulated Annealing
10- Tabu Search
11- eighborhood Search
12- Active Schedule
13- Classifying
14- Machine
15- Job
16- Makespan
17- Triangular Fuzzy Number
18- Extension Principle
19- Zadeh
20-Agreement Index

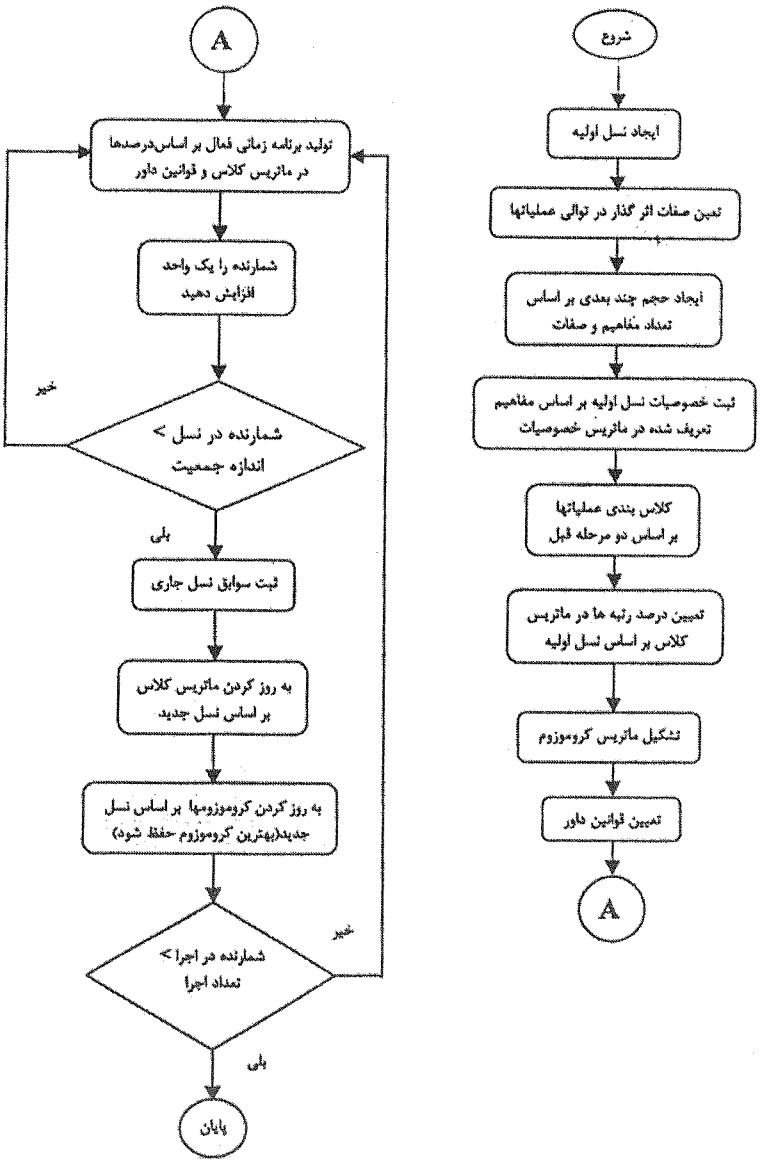
21- Darvin
22- Search Space
23- Coding
24- Chromosome
25- Population
26-Generation
27-Fittness Function
28- Crossover
29-Mutation
30- Decoding
31- Illegal
32- Infeasible
33- Parent
34- Offspring
35- Partial-mapped crossover
36- Order Crossover
37- Position based crossover
38- Adaptive genetic algorithm
39- Heuristic genetic algorithm
40- Hybrid genetic algorithm
41- Displacement
42-Knowledge Discovery
43- Machine Learning
44- Pattern recognition
45- Classifying
46- Clustering
47- Any



شکل (۲۶) فلوچارت روش تولید برنامه زمانی فعال.



شکل (۲۷) فلوچارت روش ایجاد نسل اولیه.



شکل (۲۸) ساختار کلی الگوریتم.

مراجع

[1] Cheng R, Gen M, Tsujimura Y, "A tutorial survey of job-shop scheduling problems using genetic algorithms, part II, hybrid genetic cearch strategies" International journal of computers and Industrial Engineering, 1996; 30:983-97

[2] Dileep R. Sule. "Industrial Scheduling", PWS Publishing Company, 1997.

[3] Prof. Dr. J. Kaschel, Dr. T. Teich, G. Kobernik, B.Meier, "Algorithms for the job shop Scheduling Problems-a comparison of different methods" 1999.

[4] Baker K. "Introduction to sequencing and scheduling", New York: John Wiley & Sons, 1974.

[5] H. J. Zimmermann, "Fuzzy set theory and its applications", third edition, 1996.

[6] Sakawa M, Kubota R. "Fuzzy programming for job shop scheduling", 2000.

[7] G. Bortoland, R. Degani, "A review of some methods for ranking fuzzy subsets", 1993.

[8] Davis L, editor, "Handbook of gesnetic algorithm", New York: Van Nostrand Reinhold, 1991.

[9] Gen M, Cheng R. "Genetic Algorithms and engineering design", New York: John Wiley & Sons, 1997.

[10] Jiawei Han, Micheline Kamber, "Data mining: Concepts and Techniques", 2001.

[11] David W. Cheung, H. Y. Wang, Ada W. Fu, Jiawei Han, "Efficient Rule-Based Attribute-Oriented Induction for Data Mining", 2001.

[12] Masatoshi Salawa, Tetsuya Mori, "An effective genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due date", 1999.

[13] A. Koonce, S. C. Tsai, "Using data mining to find patterns in genetic algorithms to a job shop schedule" Computers and industrial engineering, 2000.