



ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

# الگوریتم ژنتیک چند هدفی و کاربردهای آن در برنامه زمانبندی Flowshop

## چکیده

در این مقاله الگوریتم ژنتیک چند هدفی ارائه می کنیم و آن را به برنامه زمانبندی flowshop اعمال می کنیم. ویژگی های مشخصه الگوریتم ما رویه انتخاب آن و استراتژی نگهداری نوابغ آن است. رویه انتخاب در الگوریتم ژنتیک چند هدفی ما افراد را برای عملیات متقاطع برگزیده بر مبنای مجموع وزن دهی شده توابع چند هدفی با وزن های مختلف انتخاب می کند. استراتژی حفظ نوابغ در الگوریتم ما از راه حل های چندنخبه ای به جای انتخاب یک نخبه منفرد استفاده می کند. یعنی، تعداد مشخصی از افراد از مجموعه آزمایشی از راه حل های بهینه Pareto انتخاب شده اند و در نسل افراد نابغه بعدی جایگزین شده اند. برای نشان دادن اینکه مشی ما می تواند مسائل بهینه سازی چند هدفی را با نماهای Pareto کاو کنترل و اداره نماید، ما الگوریتم ژنتیکی پیشنهادی را به یک مسئله بهینه سازی تابع دو هدفی با نمای Pareto کو اعمال می کنیم. در نهایت، عملکرد الگوریتم ژنتیک چند هدفی ما با اعمال آن به مسئله برنامه زمانبندی Flowshop آزموده شده است: برای به حداقل رساندن زمان اتمام آخرین کار و به حداقل رساندن و به حداقل رساندن تأخیر مجموع. علاوه بر این ما الگوریتم خود را به مسأله برنامه زمانبندی flowshop با سه هدف اعمال می کنیم: به حداقل رساندن زمان اتمام آخرین کار، به حداقل رساندن دیرکرد مجموع، و به حداقل رسان زمان روند (گردش) مجموع.

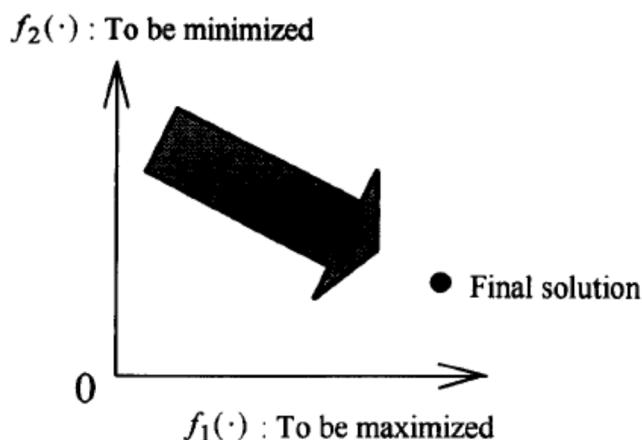
## مقدمه

مسائل برنامه زمانبندی flowshop یکی از معروفترین مشکلات در حوزه برنامه زمانبندی هستند. از کار جانشون به بعد هدف به حداقل رساندن زمان اتمام آخرین کار به عنوان شرایط زمانبندی flowshop بکار رفته است. مشی های اکتشافی مختلف و همچنین تکنیک های بهینه سازی برای به حداقل رساندن زمان اتمام آخرین کار پیشنهاد شده اند. در حالیکه این مطالعات هدف منفردی را بهبود می دهند، مسائل جهان واقعی متعددی شامل چندین هدف می شوند. اخیراً چند محقق مسائل زمانبندی flowshop چند هدفی را حل کرده اند. به عنوان مثال، Ho و

Chan روش اکتشافی برای زمانبندی flowshop با دو شرط ارائه کرده اند، Gangadharan و همکارانش زمانبندی اکتشافی سخت شبیه سازی شده را برای زمانبندی flowshop با دو شرط ارائه کرده اند، و Morizawa و همکارانش روش نمونه گیری تصادفی پیچیده ای برای مسائل چند منظوره (هدفی) ارائه کرده اند.

الگوریتم های ژنتیک عمدتاً برای مسائل بهینه سازی تک هدفی به کار رفته اند. وقتی الگوریتم ژنتیک تک هدفی را به مسأله بهینه سازی چند هدفی اعمال می کنیم باید چندین تابع هدف را در یک تابع تناسب عددی تلفیق کنیم. اگر وزن ثابتی به هر یک از چند تابع هدف برای تلفیق آنها اختصاص دهیم، جهت جستجو در الگوریتم ژنتیک در فضای هدف چند بعدی، به گونه ای که در شکل 1 نشان داده شده است ثابت می ماند. در شکل 1،  $f_1(\cdot)$  یک تابع هدف است که باید بیشینه شود و  $f_2(\cdot)$  در حال بیشینه شدن است. چرخه بسته موجود در شکل 1 نشان دهنده راه حل نهایی با الگوریتم ژنتیک تک هدفی است.

پس از کار Schaffer برخی مطالعات سعی در طراحی الگوریتم های ژنتیک چند منظوره کرده اند. Schaffer الگوریتم ژنتیک ارزیابی شده برداری (VEGA) را برای یافتن راه حل های بهینه Pareto مسائل بهینه سازی چند هدفی پیشنهاد کرده است. در VEGA ی Schaffer گروهی به زیرگروه های مجزا تقسیم شده بود، سپس هر زیر گروه توسط تابع هدف خود اداره و کنترل گشته بود. اگرچه Schaffer نتایج موفقیت آمیزی از کار خود گزارش کرده است، اما به نظر می رسد روش او تنها قادر به یافتن راه حل های فوق العاده در دیدگاه های Pareto می باشد، درست همان طور که در شکل 2 نشان داده است چون جهات تحقیق او موازی با محورهای فضای هدف هستند. Schaffer دو روش برای بهبود مشی خود در مقاله اش ارائه کرده است. یکی از این روش ها ارائه اولویت انتخاب اکتشافی برای افراد چیره نشده در هر نسل است. دیگری پیوند زدن میان گونه ها با اضافه کردن چند انتخاب جفت است.

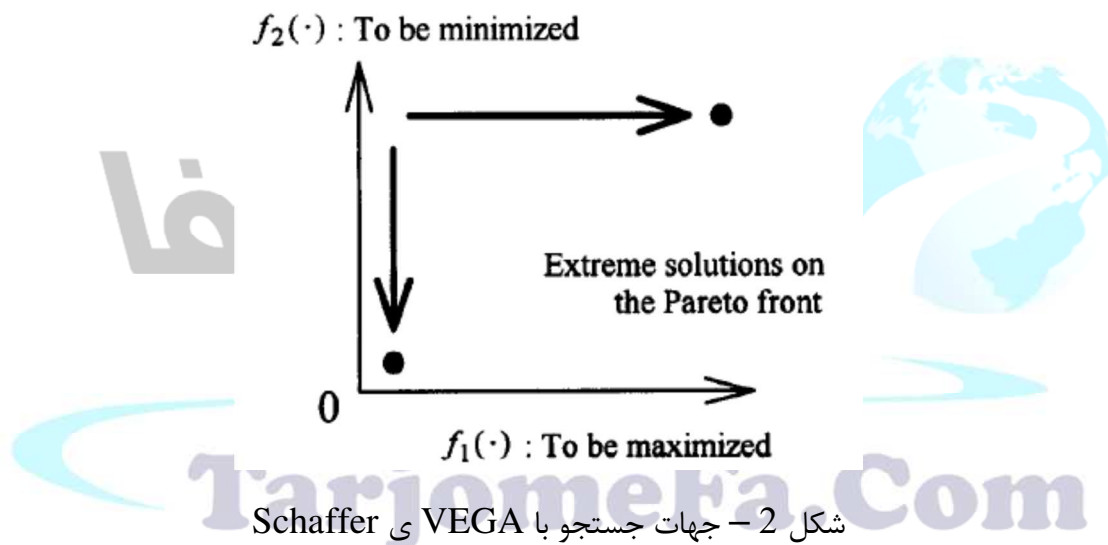


شکل 1 - جهت جستجو بواسطه GA با تابع تناسب تلفیق شده

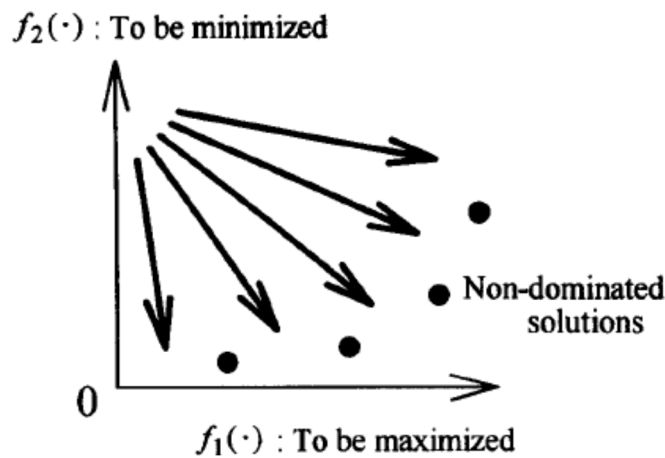
نکته مسائل بهینه سازی چند هدفی این است که چگونه تمام تبادلات ممکن میان چند تابع هدف را بیابیم که معمولاً با هم تناقض دارند. چون انتخاب راه حل منفردی برای ی مسأله بهینه سازی چند هدفی بدون تعاملات تکراری با ایجاد کننده جهت مشکل است، مشی معمول و کلی نشان دادن مجموعه راه حل های بهینه Pareto برای تصمیم گیرنده است. پس از آن تصمیم گیرنده می تواند هر یک از راه حل های بهینه Pareto را انتخاب کند. برای درک تمام راه حل های بهینه Pareto با الگوریتم های ژنتیک، انواع مختلفی از افراد را باید در هر نسل نگاه داشت. اخیراً Gen و همکارانش الگوریتم ژنتیکی برای حل یک مسأله حمل و نقل دو ظری ارائه داده اند، Tamaki و همکارانش الگوریتم ژنتیکی برای مسائل زمانبندی با چند شرط پیشنهاد داده اند، و Horn و همکارانش الگوریتم ژنتیک Pareto ی پسرفت را با مشارکت دادن مفهوم تفوق Pareto در رویه انتخاب و اعمال فشار عقب نشینی برای پشت سر گذاشتن گروه در بخش های پیشروی Pareto ارائه داده اند.

در این مقاله کا الگوریتم ژنتیک چند منظوره را با جهات تحقیق مختلف، به گونه ای که در شکل 3 نشان داده شده است ارائه می کنیم. دو ویژگی مشخصه در الگوریتم ما وجود دارد. یکی از آنها رویه انتخاب است. در رویه انتخاب، الگوریتم ژنتیک چند منظوره ما از مجموعه وزن دهی شده چند تابع هدف برای تلفیق آنها در یک تابع تناسب عددی استفاده می کند. وزن های الصاق شده به چند تابع هدف ثابت نیستند، بلکه بصورت تصادفی به هر انتخاب اختصاص یافته اند. بنابراین جهت جستجو در الگوریتم ژنتیک چند هدفی ما ثابت نیست. تعداد مشخصی از افراد در این

مجموعه به نسل بعدی که افراد نابغه هستند، مواردی به ارث می دهند. برای نشان دادن اینکه روش ما می تواند مسائل بهینه سازی چند هدفی را با دیدهای پیشروی Pareto کاو کنترل نماید ما الگوریتم ژنتیک پیشنهادی خود را به مسأله بهینه سازی تابع دو هدفی با دیدهای Pareto ی کاو اعمال می کنیم. عملکرد الگوریتم ژنتیک چند هدفی ما با اعمال آن به مسأله زمانبندی flowshop با دو هدف آزموده است: برای به حداقل رساندن زمان اتمام آخرین کار و به حداقل رساندن تأخیر کل. علاوه بر این ما الگوریتم خود را به مسأله زمانبندی flowshop با سه هدف اعمال کرده ایم: برای به حداقل رساندن زمان اتمام آخرین کار، به حداقل رساندن تأخیر مجموع، و برای به حداقل رساندن flowtime مجموع.



شکل 2 - جهات جستجو با VEGA ی Schaffer



شکل 3 - جهات جستجو توسط الگوریتم ژنتیک چند هدفی

## 2- عملیات ژنتیکی در الگوریتم ژنتیکی چند هدفی

### 1-2- رویه انتخاب

یکی از ساده ترین روش ها برای تلفیق چندین تابع هد در یک راه حل تناسب عددی مشی مجموع وزن دهی شده زیذ است (ما فرض را بر این می گذاریم که تمام توابع هدف باید بیشینه شوند):

$$f(\mathbf{x}) = w_1 \cdot f_1(\mathbf{x}) + \dots + w_i \cdot f_i(\mathbf{x}) + \dots + w_n \cdot f_n(\mathbf{x}), \quad (1)$$

که در آن  $\mathbf{x}$  یک رشته است (یعنی، راه حل)،  $f(\mathbf{x})$  تابع تناسب تلفیقی است،  $f_i(\mathbf{x})$  آمین تابع هدف است،  $w_i$  وزن ثابت برای  $f_i(\mathbf{x})$  است و  $n$  تعداد توابع هدف است.

اگر ما از مجموع وزن دهی شده در (1) به همراه  $w_i$  ها با وزن های ثابت استفاده کنیم جهت جستجو در الگوریتم های ژنتیک به گونه ای که در شکل 1 نشان داده شده است ثابت می شود. بنابراین ما ایده استفاده از وزن های متغیر را در رویه انتخاب پیشنهاد می کنیم. هدف این ایده تحقق یافتن جهات جستجوی مختلف در شکل 3 برای جستجوی راه حل بهینه Pareto است. به عنوان مثال، می توانیم وزن های متغیر را به صورت زیر به یک مسأله بهینه سازی دو نظوره اختصاص دهیم:

$$f(\mathbf{x}) = w_1 \cdot f_1(\mathbf{x}) + w_2 \cdot f_2(\mathbf{x}), \quad (2)$$

$$\begin{cases} w_1 = (i - 1) / (N_{\text{selection}} - 1), & i = 1, 2, \dots, N_{\text{selection}}, \\ w_2 = 1 - w_1, \end{cases} \quad (3)$$

که در آن  $N_{\text{selection}}$  تعداد انتخاب ها در هر نسل است. شمای وزن دهی در (2) - (3) بدین معناست که الگوریتم ژنتیک چند منظوره ما از وزن های مختلفی برای انتخاب هر زوج رشته والد استفاده کرده است. چون زوج های  $N_{\text{selection}}$  رشته های والد باید در هر نسل انتخاب شوند، زوج های  $N_{\text{selection}}$  با وزن های مختلف در (3) اختصاص یافته اند. در (3) مقادیر  $w_1$  و  $w_2$  بصورت زوج در بازه بسته  $[0, 1]$  توزیع شده اند. در شبیه سازی کامپیوتری در این مقاله، ما از شمای وزن دهی شده در (2)-(3) برای مسائل دو هدفی استفاده کرده ایم.

بطور کلی، می‌توانیم به صورت تصادفی ارزش هر وزن را تعیین کنیم. برای یک مسئله بهینه‌سازی چند هدفی با  $n$  تابع هدف، وقتی هر زوج رشته والد برای عملیات متقاطع انتخاب شده‌اند می‌توانیم عدد واقعی تصادفی به هر وزن به شرح زیر اختصاص دهیم.

$$w_i = \frac{rnd_i}{\sum_{j=1}^n rnd_j}, \quad i = 1, 2, \dots, n, \quad (4)$$

که در آن  $rnd_i$  و  $rnd_j$  اعداد صحیح تصادفی غیر منفی هستند (یا اعداد حقیقی تصادفی غیر منفی). از (4)، می‌توان مشاهده کرد که  $n$  عدد تصادفی واقعی تولید شده برای وزن‌های  $w_i$  زمانی مجموع وزن‌دهی شده را در (1) محاسبه می‌کند که هر زوج از رشته‌های والد انتخاب شده باشند. این رویه  $n_{\text{selection}}$  مرتبه در هر نسل برای انتخاب  $n_{\text{selection}}$  زوج از رشته‌های والد برای عملیات متقاطع تکرار شده است. در شبیه‌سازی‌های کامپیوتری برای مسئله سه هدفی، ما از شمای وزن‌دهی در (1) و (4) استفاده کرده‌ایم. مجموع وزن‌دهی شده  $f(x)$  در (1) و (2) برای تعیین احتمال انتخاب هر رشته در الگوریتم ژنتیک چند هدفی ما استفاده شده‌اند. چون وزن‌های  $w_i$  ثابت نیستند بلکه متغیرند، احتمال انتخاب هر رشته نیز حتی در نسل منفرد متغیر است. ایت به جهات تحقیق مختلف در الگوریتم ژنتیک چند هدفی ما تحقق می‌بخشد.

## 2-2- استراتژی حفظ نوابغ

در مسائل بهینه‌سازی چند هدفی، راه حل با بهترین مقدار برای هر هدف را می‌توان به عنوان فرد نابغه در نظر گرفت. بنابراین ما  $n$  فرد نابغه برای مسئله  $n$  هدفه داریم. فکر این مورد طبیعی است که چنین راه‌حلی باید برای نسل بعد در الگوریتم‌های ژنتیک حفظ شوند.

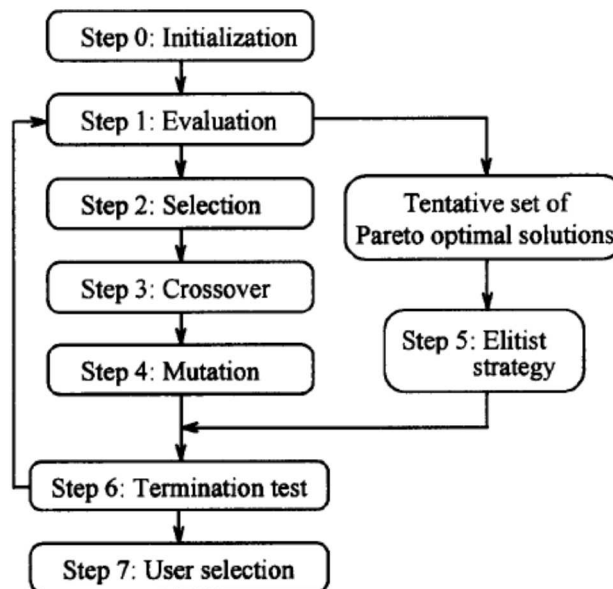
در عرض اجرای الگوریتم ژنتیک چند هدفی ما، مجموعه آزمایشی از راه‌حل‌های بهینه Pareto ذخیره شده است و در هر نسل بروز رسانی شده است. علاوه بر این ما تعداد معینی از افراد را نگه می‌داریم که بطور تصادفی از مجموعه آزمایشی راه‌حل‌های بهینه Pareto علاوه بر  $n$  فرد نابغه با توجه به  $n$  هدف انتخاب شده‌اند. یعنی، چندین راه‌حل

بهینه Pareto به عنوان افراد نابغ در الگوریتم ژنتیک چند هدفی ما استفاده شده اند. این استراتژی حفظ نابغه روی نگهداری انواع مختلفی در هر گروه در الگوریتم ژنتیک چند هدفی ما تأثیر می گذارد.

در شبیه سازی کامپیوتری، سه راه حل به عنوان افراد نابغه برای مسأله زمانبندی flowshop دو هدفی انتخاب شده اند: دو فرد نابغه با توجه به دو هدف و یک فرد انتخاب شده بصورت تصادفی از مجموعه راه حل بهینه Pareto ی آزمایشی. بعنوان یک متغیر، ما می توانیم الگوریتم ژنتیک چند هدفی را با انتخاب تصادفی تعداد معینی از افراد نابغه از مسأله زمانبندی طراحی کنیم. در شبیه سازی کامپیوتری برای یک مسأله زمانبندی flowshop سه هدفی، سه راه حلی که بصورت تصادفی از مجموعه آمایشی از راه حل های بهینه Pareto انتخاب شده بودند، به نسل بعدی منتقل شدند (به ارث داده شدند).

### 3-2- الگوریتم

عملیات ژنتیک زیر برای تولید و کنترل (اداره) یک گروه (به عنوان نمونه، مجموعه ای از رشته ها) در الگوریتم ژنتیک چند هدفی ما بکار گرفته شده اند (به شکل 4 مراجعه نمایید).



شکل 4 - طرح کلی الگوریتم ژنتیک چند هدفی ما

مرحله 0 (تخصیص اولیه): ایاد یک گروه اولیه حاوی  $N_{pop}$  رشته که در آن  $N_{pop}$  تعداد رشته ها در هر گروه است.



مرحله 1 (ارزیابی): محاسبه مقادیر توابع هدف برای رشته هایوزن دهی شده. بروز رسانی مجموعه آزمایشی از راه حل های بهینه Pareto.

مرحله 2 (انتخاب): محاسبه مقدار تناسب  $f(x)$  هر رشته با استفاده از وزن های تعریف شده در (3) و (4). انتخاب زوجی از رشته ها از گروه فعلی طبق احتمال انتخاب زیر. احتمال انتخاب  $P(x)$  رشته  $x$  در گروه  $\Psi$  به صورت زیر اختصاص یافته است:

$$P(x) = \frac{f(x) - f_{\min}(\Psi)}{\sum_{x \in \Psi} \{f(x) - f_{\min}(\Psi)\}}, \quad (5)$$

که در آن

$$f_{\min}(\Psi) = \min\{f(x) | x \in \Psi\}. \quad (6)$$

این مرحله  $N_{\text{selection}}$  مرتبه تکرار شده تا منشأ  $N_{\text{pop}}$  بواسطه عملیات متقاطع در مرحله 3 ایجاد شود. مقادیر وزن های  $w_i$  در (3) یا (4) در هر یک از  $N_{\text{selection}}$  تکرار اختصاص یافته اند.

مرحله 3 (متقاطع): برای هر زوج منتخب، اعمال عملیات متقاطع برای ایجاد یک منشأ با احتمال متقاطع  $P_e$ .  $N_{\text{pop}}$  رشته باید با عملیات متقاطع ایجاد شوند.

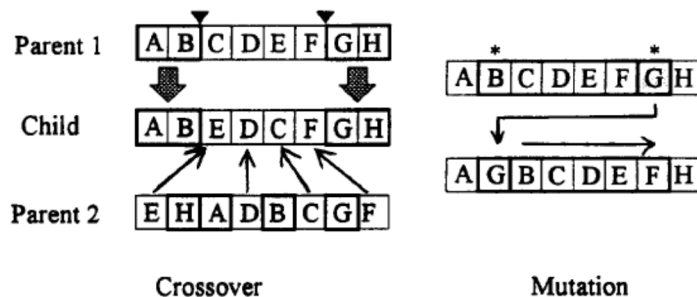
مرحله 4 (جهش): برای هر رشته تولید شده با عملیات متقاطع، اعمال عملیات جهش با احتمال جهش تخصیص یافته  $P_m$ .

مرحله 5 (استراتژی هوشمندانه): حذف رشته های  $N_{\text{elite}}$  بصورت تصادفی از رشته های  $N_{\text{pop}}$  تولید شده با عملیات بالا، و اضافه کردن همان تعداد رشته از مجموعه آزمایشی راه حل های بهینه Pareto به گروه فعلی.

مرحله 6 (تست خاتمه): اگر شرایط توقف از پیش تعیین شده برقرار نشده است، به مرحله 1 بازگردید.

مرحله 7 (انتخاب کاربر): الگوریتم ژنتیک چند هدفی مجموعه نهایی از راه حل های بهینه Pareto را به تصمیم گیرنده نشان می دهد. راه حل منفرد (یعنی، راه حل نهایی) بواسطه اولویت تصمیم گیرنده انتخاب می شود.

در این مقاله، ما از روش متقاطع دو نقطه ای و جهش چرخش که در شکل 5 برای زمانبندی flowshop نشان داده شده است استفاده کرده ایم.



شکل 5 - عملگرهای ژنتیکی

ما الگوریتم ژنتیک چند هدفی را به مسأله تست با اعمال کرده ایم که با دو هدف زیر همراه است که باید کمینه شوند:

$$\text{Minimize } f_1(\mathbf{x}) = 2\sqrt{x_1}, \quad (7)$$

$$\text{Minimize } f_2(\mathbf{x}) = x_1(1 - x_2) + 5, \quad (8)$$

مربوط به

$$\begin{cases} 1 \leq x_1 \leq 4, \\ 1 \leq x_2 \leq 2. \end{cases} \quad (9)$$

با جایگزین کردن (7) در معادله (8)، ما به رابطه میان  $f_1(\mathbf{x})$  و  $f_2(\mathbf{x})$  به صورت زیر می رسمیم:

$$f_2(\mathbf{x}) = \frac{1 - x_2}{4} \cdot \{f_1(\mathbf{x})\}^2 + 5. \quad (10)$$

وقتی  $x_2=2$  باشد، معادله (10) پیشروی Pareto این مسأله را ارائه می کند. این پیشروی Pareto شکل کاو را در فضای هدف به گونه ای که در شکل 6 نشان داده شده تشکیل می دهد.

در الگوریتم ژنتیک چند هدفی ما، تابع تناسب  $f(\mathbf{x})$  به شرح زیر اختصاص یافته است.

$$f(\mathbf{x}) = -w_1 \cdot f_1(\mathbf{x}) - w_2 \cdot f_2(\mathbf{x}). \quad (11)$$

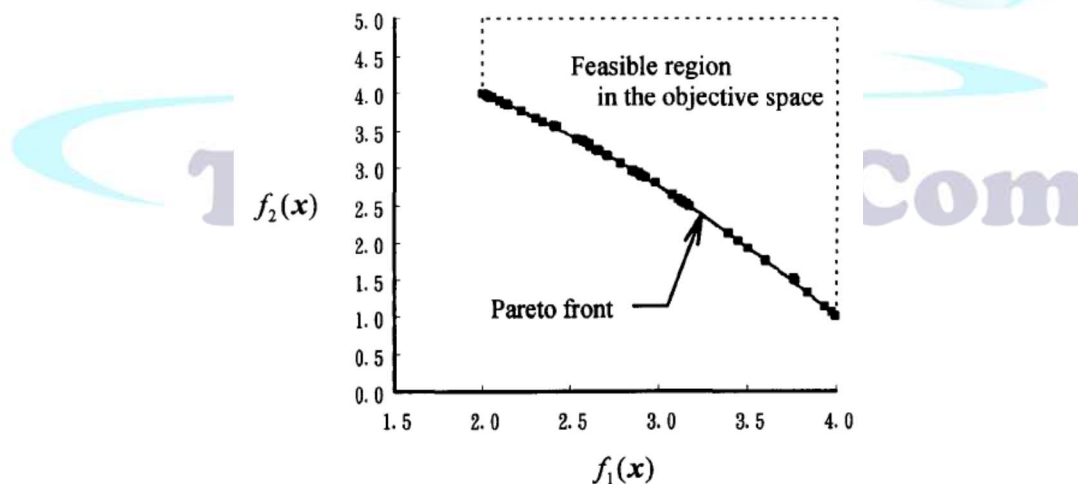
چون  $f_1(x)$  و  $f_2(x)$  باید کمینه شوند، علامت منفی "-" به هر وزن در (11) الصاق شده است.

در الگوریتم ژنتیک چند هدفی ما، تعداد راه حل ها در هر گروه (یعنی، اندازه گروه) به صورت  $N_{pop}=100$  اختصاص یافته است، دو برخورد نقطه ای با احتمال برخورد  $0/9$  بکار رفته است، احتمال جهش به صورت  $P_m=0.01$  اختصاص یافته است، و تعداد افراد نابغه بصورت  $N_{elite}=5$  اختصاص یافته است (پنج راه حل هوشمندانه بصورت تصادفی از یک مجموعه آزمایشی از راه حل های بهینه انتخاب شده اند). در شکل 6، راه حل های نهایی با الگوریتم ژنتیک چند هدفی ما را پس از 20 نسل نشان می دهد. از شکل 6، می توان مشاهده کرد که الگوریتم ژنتیک چند هدفی می تواند راه حل های متعددی در پیشروی Pareto کاو بیابد.

علاوه بر این ما الگوریتم ژنتیک تک موضوعی را اعمال کرده ایم که در آن وزن های  $w_1$  و  $w_2$  به شرح زیر ثابت شده

اند:

$$w_1:w_2 = 100:1, 50:1, 20:1, 15:1, 10:1, 5:1, 2:1, \\ 1:1, 1:2, 1:5, 1:10, 1:15, 1:20, 1:50, 1:100.$$



شکل 6- نتایج شبیه سازی برای مسأله پیشروی Pareto کاو

الگوریتم ژنتیک تک هدفی با این 15 مقدار وزنی مختلف تنها دو راه حل Pareto پیدا کرده است:

$(f_1, f_2) = (2, 4)$  و  $(f_1, f_2) = (4, 1)$ . از این نتایج شبیه سازی، می توان مشاهده کرد که الگوریتم ژنتیک تک

هدفی با وزن های ثابت نمی واند هیچ گونه پیشروی Parto کاوی بیابد، حتی اگر مقادیر وزنی مختلفی به کار گرفته شده باشند.

#### 4- نتایج شبیه سازی برای زمانبندی flowshop

در این بخش، ما کارایی الگوریتم ژنتیک چند هدفی خود را با شبیه سازی های کامپیوتری در مسأله زمانبندی flowshop با دو هدف نشان می دهیم: برای کمنه کردن زمان اتمام آخرین کار و برای به حداقل رساندن تأخیر مجموعی علاوه بر این الگوریتم خود را برای مسأله زمانبندی flowshop با سه هدف اعمال می کنیم: برای به حداقل رساندن زمان اتمام آخرین کار، برای به حداقل رساندن تأخیر کل، و برای به حداقل رساندن flowtime مجموع.

مسائل زمانبندی flowshop یکی از شناخته شده ترین مسائل در حوزه زمانبندی است. فرضیات کلی مسائل زمانبندی flowshop را می توان به شرح زیر نوشت (به کار Dudek و همکارانش مراجعه کنید). کارها در چندین مرحله پشت سر هم در حال پردازش هستند. یک ماشین در هر مرحله وجود دارد. ماشین ها بصورت مداوم در دسترسند. یک کار در یک ماشین در یک زمان بدون اولویت پردازش شده اند، و یک ماشین بیش از یک کار را در یک زمان پردازش نمی کند. در این مقاله، ما فرض را بر این می گذاریم که  $n$  کار به همان ترتیب روی  $n$  ماشین در حال اجرا هستند. این بدین معناست که زمانبندی flowshop مایک مسأله ترتیب بندی  $n$  کاری است. در این مقاله، ترتیب  $n$  کار با رشته  $\mathbf{x} = (x_1, \dots, x_k, \dots, x_n)$  نشان داده شده که در آن  $x_k$ ،  $K$  امین کاری است که باید روی  $m$  ماشین پردازش شود.

#### 4-1- مسأله زمانبندی flowshop دو هدفی

در شبیه سازی های کامپیوتری، ما از دو هدف به عنوان شرایط زمانبندی استفاده کرده ایم؛ برای به حداقل رساندن زمان اتمام آخرین کار و برای به حداقل رساندن تأخیر مجموعی. ما تاریخ مقرر هر کار توسط رویه زیر:

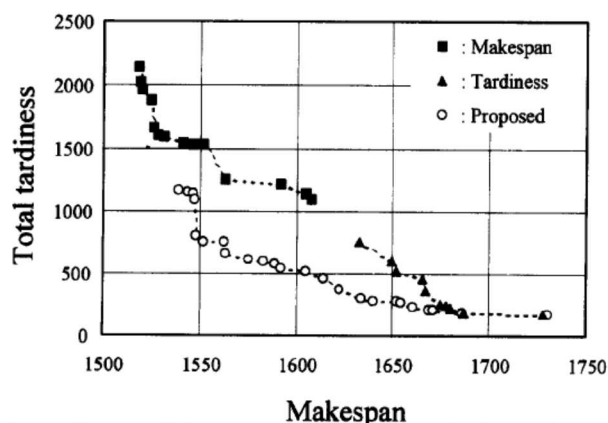
مرحله 1: تولید تصادفی اولویت  $n$  کار.

مرحله 2: محاسبه زمان تکمیل  $C_j$  هر کار،  $j = 1, 2, \dots, n$ .

مرحله 3: اضافه کردن عدد تصادفی صحیح  $rnd_j$  در بازه بسته  $[100, -100]$  به هر  $C_j$ . یعنی، زمان مقرر  $J$  امین کار  
 $d_j = C_j + rnd_j$  است.

می دانیم که هیچ همبستگی میان دو هدف وجود ندارد: زمان اتمام آخرین کار و تأخیر مجموع. در الگوریتم ژنتیک  
 چند هدفی ما، تابع تناسب  $f(x)$  را می توان به صورت زیر نوشت:

$$f(x) = -W_{Makespan} \cdot Makespan(x) - W_{Tardiness} \cdot Tardiness(x), \quad (12)$$



شکل 7 - مقایسه الگوریتم ژنتیک چند هدفی با دو آزمایش الگوریتم ژنتیک تک هدفی

که در آن  $Makespan(x)$  زمان اتمام آخرین کار وقتی است که  $n$  کار به ترتیب  $x$  در حال پردازشند،  
 $Tardiness(x)$  تأخیر کل است، و  $W_{Makespan}$  و  $W_{Tardiness}$  به ترتیب وزن های متغیر غیر منفی برای  
 $Makespan(x)$  و  $Tardiness(x)$  هستند. در شبیه سازی های کامپیوتری،  $W_{Makespan}$  و  $W_{Tardiness}$  به همان  
 شیوه (3) تخصیص یافته اند. چون  $Makespan(x)$  و  $Tardiness(x)$  باید کمینه شوند، علامت منفی به هر وزن  
 در (12) الصاق شده است.

در مسأله تست، ما مسأله زمانبندی flowshop را با 20 کار و 10 ماشین با تخصیص تصادفی زمان پردازش هر کار  
 در هر ماشین به صورت عدد صحیحی در وقفه بسته [99 و 1] تولید کرده ایم. در شبیه سازی کامپیوتری، ما از  
 تخصیص های پارامتری زیر استفاده کرده ایم.

$$N_{pop}=10$$

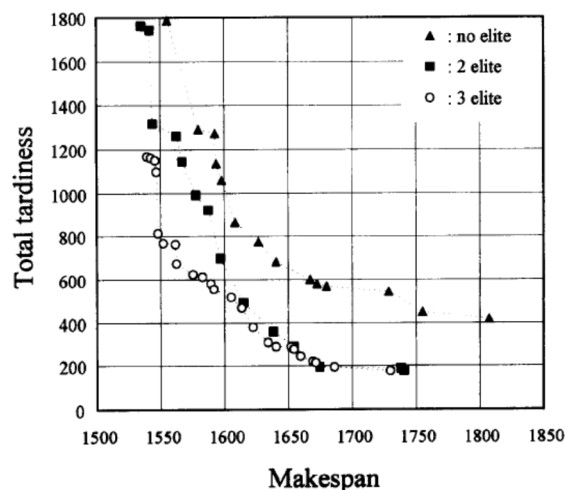
$$P_c=1.0$$

احتمال دگرگونی:  $P_m=1.0$  به ازای هر رشته

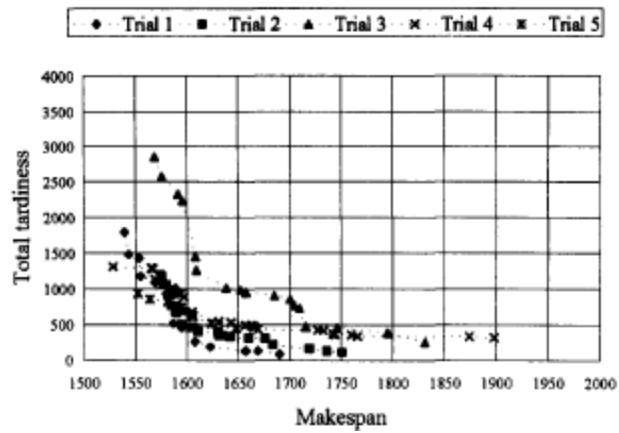
راه حل های تفوق نیافته ای که با الگوریتم پیشنهادی حاصل شده زانند با  $\circ$  در شكا 7 نشان داده شده اند که در آن محورهای افقی و عمودی به ترتیب زمان اتمام آخرین کار و تأخیر مجموع هستند. در شکل 7، راه حل های تفوق نیافته بدست آمده با دو آزمایش الگوریتم ژنتیک تک هدفی با  $\blacksquare$  (بدست آمده با الگوریتم ژنتیک برای به حداقل رساندن زمان اتمام آخرین کار) و  $\blacktriangle$  (بدست آمده با الگوریتم ژنتیک برای به حداقل رساندن تأخیر مجموع) هستند. در الگوریتم ژنتیک تک هدفی تابع تناسب معادل با مقدار تابع هدف آن با علامت منفی بود. مجموعه آزمایشی راه حل های بهینه Pareto در الگوریتم ژنتیک تک هدفی ذخیره و بروز رسانی شده اند.

برای مقایسه الگوریتم پیشنهادی با دو آزمایش الگوریتم ژنتیک تک هدفی تحت همان بار محاسباتی، ما تعداد ارزیابی های تابع تناسب را در الگوریتم ژنتیک چند هدفی به مقدار 100000 تنظیم می کنیم در هر آزمایش الگوریتم ژنتیک تک هدفی به آن مقدار 50000 اختصاص می دهیم. بنابراین 100000 راه حل در هر مشی ارزیابی شده اند.

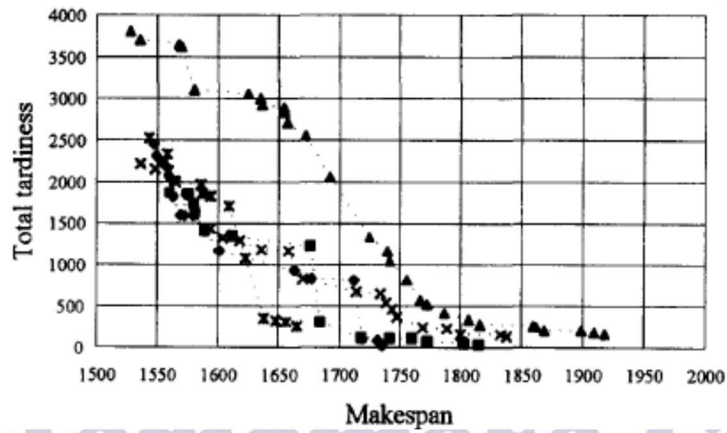
از شکل 7 می توان مشاهده کرد که راه حل های تفوق نیافته حاصل شده با الگوریتم ژنتیک چند هدفی بر مجموعه راه حل های تفوق نیافته بدست آمده با الگوریتم ژنتیک تک هدفی برتری دارند. این به دلیل آن است که راه حل های متعددی که با  $\blacksquare$  و  $\blacktriangle$  نشان داده شده اند بر راه حل هایی که با  $\circ$  نشان داده شده اند برتری نترند. این نشان دهنده عملکرد بالای الگوریتم ژنتیک چند هدفی ماست.



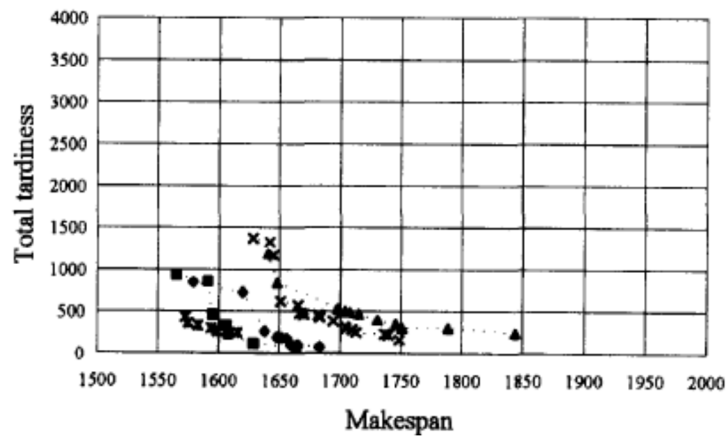
شکل 8 - تأثیر تعداد افراد نابغه



الف) الگوریتم ژنتیک چند هدفی



ب) Schaffer و VEGA



ج) الگوریتم ژنتیک تک هدفی

شکل 9 - مقایسه الگوریتم ژنتیک چند هدفی با VEGA ی Schaffer و الگوریتم ژنتیک تک هدفی

کارایی استراتژی حفظ نابغه در شکل 8 نشان داده شده است، در شکل 8، هیچ نابغه ای، 2 نابغه و 3 نابغه به ترتیب یعنی هیچ فرد نابغه ای، دو فرد نابغه و سه فرد نابغه برای گروه بعدی در الگوریتم ژنتیک چند هدفی حفظ شده اند. در الگوریتم 2 نابغه ای، تنها افراد نابغه با توجه به دو تابع هدف حفظ شده اند. از طرف دیگر، فرد انتخاب شده بصورت تصادفی از مجموعه آزمایشی از راه حل های بهینه Pareto علاوه بر این دو فرد نابغه در الگوریتم 3 نابغه ای حفظ شده است.

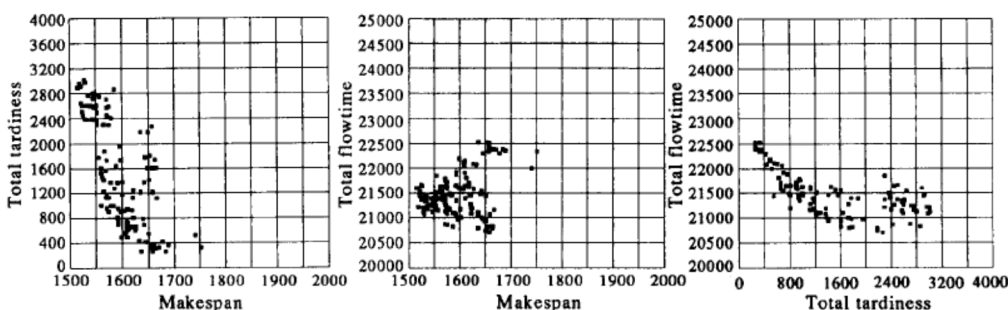
جدول 1 - زمان CPU میانگین استفاده شده توسط هر الگوریتم برای یافتن راه حل های بهینه Pareto در شکل

9

Algorithm	Proposed algorithm	Schaffer's VEGA	SOGA
CPU time (s)	3.08	3.00	2.48

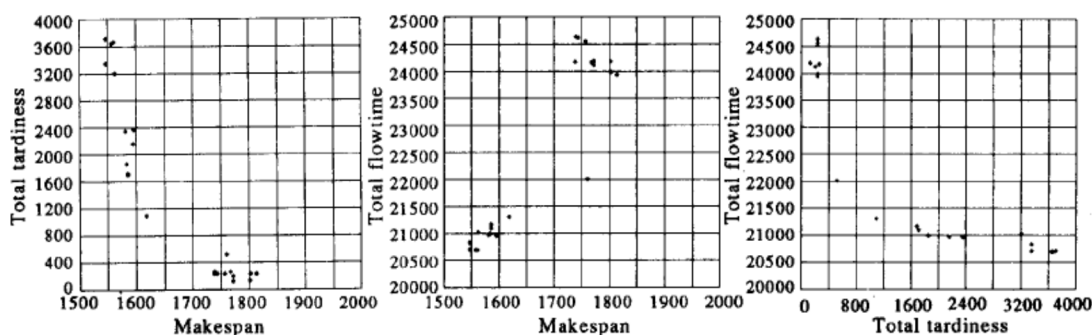
از شکل 8 می توان مشاهده کرد که روش 3 نابغه ای راه حل های بهتری نسبت به روش 2 نابغه ای و بدون هیچ نابغه ای می یابد. این بدین معناست که استراتژی حفظ نابغه ما تأثیری روی عملکرد الگوریتم ژنتیک چند هدفی دارد.

علاوه بر این ما VEGA ی Schaffer و نسخه دیگر الگوریتم ژنتیک ت هدفی را به همان مسأله زمانبندی flowshop اعمال کرده ایم. در الگوریتم ژنتیک تک هدفی، ما از وزن های  $W_{Makespan}=5$  و  $W_{Tardiness}=2$  برای محاسبه مقدار تناسب استفاده کرده ایم. مجموعه آزمایشی از راه حل های بهینه Pareto در الگوریتم ژنتیک موجود در الگوریتم ژنتیک چند هدفی ذخیره سازی و بروز رسانی شده اند.

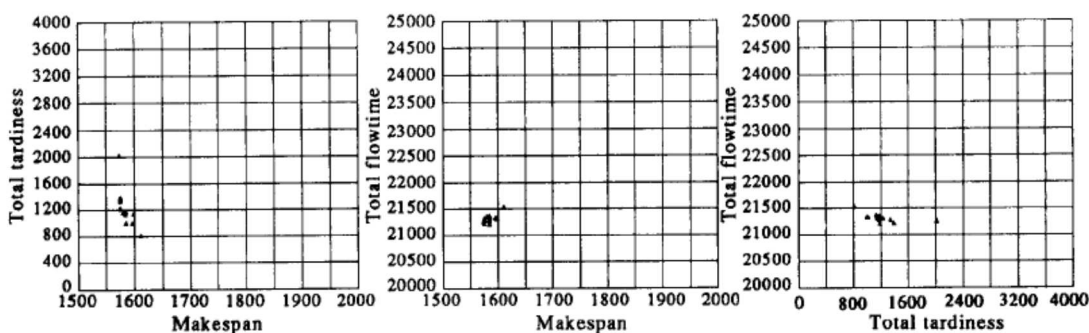




(الف) الگوریتم ژنتیک چند هدفی



(ب) Schaffer VEGA



(ج) الگوریتم ژنتیک تک هدفی

شکل 10 مقایسه الگوریتم ژنتیک چند هدفی با Schaffer VEGA و الگوریتم ژنتیک تک هدفی

به عنوان شرایط توقف، ما از تعداد مجموع ارزیابی های رشته ها استفاده کرده ایم. وقتی 100000 راه حل در هر الگوریتم ارزیابی شده اند، الگوریتم خاتمه یافته است. نتایج شبیه سازی با الگوریتم ژنتیک پیشنهادی، VEGA و Schaffer الگوریتم ژنتیک تک هدفی به ترتیب در شکل های 9 (الف)، (ب) و (ج) نشان داده شده اند. ما هر الگوریتم را پنج مرتبه به همان مسأله زمانبندی flowshop اعمال کرده ایم. از شکل 9، می توان مشاهده کرد که راه حل های بهتری با الگوریتم چند هدفی بدست آمده اند چون راه حل های متعدد بدست آمده با Schaffer VEGA در شکل 9 (ب) بر راه حل های الگوریتم ژنتیک چند هدفی موجود در شکل 9 (الف) تفوق یافته اند. الگوریتم ژنتیک تک هدفی می تواند راه حل های بهتری نسبت به الگوریتم ژنتیک چند هدفی ما بیابد. زمان CPU میانگین استفاده شده با هر الگوریتم برای یافتن راه حل های موجود در شکل 9 در جدول 1 نشان داده

شده اند. می توان از جدول 1 مشاهده کرد که زمان محاسبه مشابهی توسط هر الگوریتم در شبیه سازی های کامپیوتری صرف شده است.

## 2-4- مسأله زمانبندی flowshop سه هدفی

ما علاوه بر این الگوریتم ژنتیک چند هدفی خود را به مسأله زمانبندی Flowshop با سه هدف اعمال کرده ایم. : به حداقل رساندن زمان اتمام آخرین کار، به حداقل رساندن دیرکرد مجموع، و به حداقل رسان زمان روند (گردش) مجموع (یعنی، مجموع زمان تکمیل در تمام کارها). ما از همان پارامترهای استفاده شده برای مسأله زمانبندی flowshop دو هدفی استفاده کرده ایم. چون نشان دادن راه حل ها در فضای هدف سه بعدی مشکل است، ما نتایج شبیه سازی را با ارائه آن ها در فضای هدف دو بعدی نشان می دهیم. شکل 10 نتایج شبیه سازی حاصل شده با الگوریتم ژنتیک چند هدفی ما، Schaffer VEGA و الگوریتم ژنتیک تک هدفی را نشان می دهد. در الگوریتم ژنتیک تک هدفی، ما از وزن های ثابت  $W_{Makespan}=5$  و  $W_{tardiness}=2$  و  $W_{Flowtime}=1$  برای محاسبه مقدار تناسب استفاده کرده ایم. از شکل 10 می توان مشاهده کرد که الگوریتم ژنتیک چند هدفی ما می تواند مجموعه راه حل های بهتری بیابد.

## 5- نتیجه گیری

در این مقاله، ما چارچوب الگوریتم های ژنتیک را برای مسائل بهینه سازی چند هدفی پیشنهاد کردیم. روش ما دو ویژگی مشخصه داشت. یک ویژگی این است که وزن های استفاده شده برای تلفیق چندین هدف در یک تابع تناسب عددی بهصورت تصادفی برای هر انتخاب تخصیص یافته اند. یعنی، وزن ها ثابت نیستند بلکه در الگوریتم ژنتیک چند هدفی ما متغیرند. ویژگی مشخصه دیگر این است که چند فرد نابغه انتخاب شده از مجموعه آزمایشی از راه حل های بهینه Pareto به نسل بعدی ارث داده شده اند. با شبیه سازی های کامپیوتری، نشان داده ایم که الگوریتم ژنتیک چند هدفی ما می تواند راه حل های بهینه Pareto کاو را بیابد، و می توانیم نشان دهیم که الگوریتم ژنتیک چند هدفی می تواند راه حل های بهتری نسبت به VEGA که توسط Schaffer ارائه شده و الگوریتم ژنتیک تک هدفی بیابد.

## REFERENCES

1. S. M. Johnson. Optimal two- and three-stage production schedules with setup times included. *Naval Res. Logistics Q.* **1(1)**, 61–68 (1954).
2. D. G. Dannenbring. An evaluation of flowshop sequencing heuristics. *Mgmt Sci.* **23**, 1174–1182 (1977).
3. M. Nawaz Jr, E. E. Encscore and I. Ham. A heuristic algorithm for  $m$ -machine,  $n$ -job flowshop sequencing problem. *OMEGA* **11**, 91–98 (1983).
4. I. H. Osman and C. N. Potts. Simulated annealing for permutation flow-shop scheduling. *OMEGA* **17(6)**, 551–557 (1989).
5. M. Widmer and A. Hertz. A new heuristic method for the flowshop sequencing problem. *Europ. J. Opnl Res.* **41(2)**, 186–193 (1990).
6. E. Ignall and L. E. Schrage. Application of branch- and bound technique to some flow shop problems. *Ops Res.* **13(3)**, 400–412 (1965).
7. Z. Lomnicki. A branch- and -bound algorithm for the exact solution of the three-machine scheduling program. *Opnl Res. Q.* **16(1)**, 89–107 (1965).
8. J. C. Ho and Y.-L. Chang. A new heuristic for the  $n$ -job,  $m$ -machine flowshop problem. *Europ. J. Opnl Res.* **52**, 194–202 (1991).
9. R. Gangadharan and C. Rajendran. A simulated annealing heuristic for scheduling in a flowshop with bicriteria. *Proc. 16th Int. Conf. on Computers Ind. Engng* 345–348, 7–9 Mar. (1994).
10. K. Morizawa, T. Ono, H. Nagasawa and N. Nishiyama. An interactive approach for searching a preferred schedule. *J. Japan Ind. Mgmt Assoc.* **44(4)**, 277–283 (1993, in Japanese).
11. J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. *Proc. 1st ICGA*, pp. 93–100 (1985).
12. M. Gen, K. Ida, E. Kono and Y. Li. Solving bicriteria solid transportation problem by genetic algorithm. *Proc. 16th Int. Conf. on Computers Ind. Engng*, 572–575, 7–9 Mar. (1994).
13. H. Tamaki, M. Mori, M. Araki, Y. Mishima and H. Ogai. Multi-criteria optimization by genetic algorithms: a case of scheduling in hot rolling process. *Proc. APORS'94*, 374–381, 26–29 July (1994).
14. J. Horn, N. Nafpliotis and D. E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. *Proc. 1st CIEC*, pp. 82–87 (1994).
15. T. Murata, H. Ishibuchi and H. Tanaka. Genetic algorithms for flowshop scheduling problems. *Computers ind. Engng* (to appear).
16. R. A. Dudek, S. S. Panwalkar and M. L. Smith. The lessons of flowshop scheduling research. *Ops Res.* **40(1)**, 7–13 (1992).

The logo for TarjomeFa.Com features a stylized blue and white graphic on the left, resembling a wave or a stylized letter 'T'. To the right of this graphic, the text 'TarjomeFa.Com' is written in a large, bold, blue serif font. The background of the entire page is white with a thin black border.

TarjomeFa.Com

برای خرید فرمت ورد این ترجمه، بدون واتر مارک، اینجا کلیک نمائید.



این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

لیست مقالات ترجمه شده ✓

لیست مقالات ترجمه شده رایگان ✓

لیست جدیدترین مقالات انگلیسی ISI ✓

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی