



ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

یک الگوریتم ژنتیک برای مسئله زمانبندی انعطاف پذیر شغلی-فروشگاه

چکیده

در این مقاله، ما یک الگوریتم ژنتیک برای مسئله برنامه ریزی انعطاف پذیر شغلی-فروشگاه (FJSP) ارائه می دهیم. این الگوریتم استراتژی های مختلف را برای تولید جمعیت اولیه، انتخاب افراد برای تولید مثل و تکثیر افراد جدید ادغام می کند. نتیجه محاسباتی نشان می دهد که ادغام راهبردهای بیشتر در یک چارچوب ژنتیکی، با توجه به دیگر الگوریتم های ژنتیک، منجر به نتایج بهتر می شود. علاوه بر این، نتایج به دست آمده کاملاً قابل مقایسه با نتایج به دست آمده توسط بهترین الگوریتم شناخته شده، بر اساس جستجوی ممنوع است. این دو نتیجه، با هم به همراه انعطاف پذیری الگوی ژنتیکی ثابت می کند که الگوریتم های ژنتیکی برای حل FJSP موثر هستند.

کلمات کلیدی: زمانبندی شغلی-فروشگاه؛ الگوریتم ژنتیک؛ سیستم های تولید انعطاف پذیر

برنامه ریزی عملیات، یکی از مهم ترین مسائل در برنامه ریزی و مدیریت فرآیندهای تولید است. یافتن بهترین برنامه زمانبندی می تواند با توجه به محیط فروشگاه، محدودیت های فرآیند و شاخص عملکرد آسان و یا بسیار دشوار باشد. یکی از سخت ترین مشکلات در این زمینه، مشکل زمانبندی کار فروشگاه (JSP) است که در آن مجموعه ای از کارها باید بر روی مجموعه ای از ماشین آلات پردازش شود که هر کار توسط یک دنباله از عملیات های پشت سر هم تشکیل شده است، هر عملیات دقیقاً نیاز به یک ماشین دارد، ماشین به طور مداوم در دسترس است و می تواند یک فرایند را در یک زمان و بدون وقفه انجام دهد. این تصمیم مربوط به چگونگی توالی عملیات در دستگاه، مانند شاخص عملکرد بهینه سازی می شود. یک شاخص نوعی معمولی برای JSP، مدت کلی برنامه زمانبندی است، به عنوان مثال، زمان مورد نیاز برای تکمیل تمام مشاغل. JSP. یک مسئله سخت NP به خوبی شناخته شده است.

مسئله زمانبندی انعطاف پذیر شغلی-فروشگاه (FJSP)، یک کلیت از JSP کلاسیک، که در آن عملیات ها مجاز به پردازش در میان مجموعه ای از دستگاه های موجود هستند. سپس بنابراین FJSP از JSP کلاسیک سخت تر است. از آن جا که سطح تصمیمات بیشتر را در کنار توالی، به عنوان مثال، مسیرهای کار، عرضه می کند. تعیین مسیرهای شغلی میانی برای تصمیم گیری، برای هر عملیات، چیزی است که دستگاه باید آن را در میان مسیرهای که در دسترس هستند، پردازش نماید.

مسئله FJSP نشان دهنده روش های ابتکاری تولید برنامه های زمانبندی خوب در زمان معقول ، به به جای جستجوی یک راه حل دقیق، همچنین برای نمونه های کوچک است. ابتکارات به طور کلی راه حل هایی را ارائه نمی دهند که مقدار تابع هدف آنها دارای فاصله تضمین شده از نقطه مطلوب باشد، اما برای بسیاری از نمونه های مسئله می توانند موثر باشند. در سال های اخیر، اتخاذ روش های فرا اکتشافی مانند شبیه سازی بازپخت، جستجوی ممنوع و الگوریتم های ژنتیکی (GAS) نسبت به الگوریتم های توزیع کلاسیک و یا روش های اکتشافی سختگیرانه (۳-۵) به نتایج بهتری منجر شده است.

در این مقاله، ما یک GA جدید را برای FJSP ارائه می دهیم که برخی از استراتژی های در حال حاضر شناخته شده در نوشته ها را بهبود می بخشد و آنها را برای پیدا کردن بهترین معیارها در هر گام الگوریتم ترکیب می کند. به طور خاص، ما این رویکرد را توسط موقعیت یابی Kacem و همکاران [۶] برای پیدا کردن تکالیف اولیه اتخاذ می کنیم که آن را با استفاده از روش مرتب سازی مجدد ماشین ها و مشاغل و توسط جستجوی مینیمم کلی در جدول نمونه ها بهبود می بخشیم. سپس، بیشتر کار باقی مانده (MWR)، بیشتر عملیات باقی مانده (MOR) و انتخاب تصادفی قواعد توزیع شغل برای ترتیب بندی عملیات ها، تولید جمعیت اولیه اتخاذ می شوند. معیار انتخاب در میان تورنمنت باینری، تورنمنت با اندازه-n و رتبه بندی خطی انتخاب می شود. اپراتورهای ژنتیکی متقاطع جهشی مختلف، برای انتساب و ترتیب بندی اتخاذ می شوند. به خصوص ما در این زمینه، یک عملگر انتساب تقابل هوشمند را معرفی می کنیم. محاسبات نشان می دهد که بهترین الگوریتم GA شناخته شده را برای FJSP انجام می دهد و اینکه کیفیت راه حل با بهترین الگوریتم شناخته شده برای FJSP ، به عنوان مثال، جستجوی ممنوع

Mastrolilli و [13] Bambardell قابل مقایسه است. الگوریتم ما ثابت می کند که هنگامی که ترکیبی از قوانین مختلف برای تولید جمعیت اولیه، انتخاب افراد و اپراتورهای تولید دوباره اتخاذ شود، چارچوب GA مناسب توسعه الگوریتم های کارآمد برای FJSP است.

این مقاله به شرح زیر سازماندهی شده است. در بخش ۱ فرمول مسئله را معرفی می کنیم و نمونه ای گویا از داده های ورودی را نشان می دهیم. در بخش ۲ ما به بررسی نوشته های مربوطه در این موضوع می پردازیم. در بخش ۳ ما الگوریتم را با جزئیات استراتژی برای تولید راه حل های اولیه، کدگذاری، عملکرد ارزیابی تناسب، معیارهای انتخاب و اپراتورهای GA اتخاذ شده برای تولید نتایج نشان می دهیم. در بخش ۴، ما یک مطالعه گسترده محاسباتی روی ۱۷۸ مشکل محک زنی را که نتایج به دست آمده را با بهترین روش شناخته شده مقایسه می کند، ارائه می دهیم. بعضی اظهارات نهایی و جهت های تحقیقاتی آینده در بخش ۵ داده شده است.

۱. FJSP

FJSP را می توان به شرح زیر بیان نمود. مجموعه $J = \{J_1, J_2, \dots, J_n\}$ از شغل های مستقل داده شده است. یک شغل J_i به واسطه به دنباله ای از $O_{i1}, O_{i2}, \dots, O_{in_i}$ عملیات که باید انجام شود، یکی پس از دیگری بر اساس این دنباله معین تشکیل می شود. این مجموعه معین $U = \{M_1, M_2, \dots, M_m\}$ از ماشین آلات است. هر عملیات O_{ij} را می توان در میان هر زیر مجموعه $U_{ij} \subseteq U$ از دستگاه های سازگار اجرا نمود. ما در صورتی دارای انعطاف پذیری جزئی هستیم که یک زیر مجموعه مناسب U_{ij} از U وجود داشته باشد، برای حداقل یک عملیات O_{ij} ، در حالی که مادارای $U_{ij} = U$ برای هر عملیات در مورد انعطاف پذیری کلی هستیم. زمان پردازش هر یک از عملیات ها وابسته به ماشین است. ما آن را با d_{ijk} زمان پردازش عملیات O_{ij} زمانی که بر روی دستگاه M_k اجرا می شود، نشان می دهیم. به عنوان مثال، هر عمل باید بدون وقفه زمانی که آغاز شد، به پایان برسد. علاوه بر این، دستگاه ها نمی تواند بیش از یک عمل را در یک زمان انجام دهند. همه شغل ها و ماشین آلات در زمان • در دسترس هستند.

مشکل، تعیین هر یک از عملیات ها به یک ماشین مناسب (مشکل مسیریابی)، و ترتیب بندی عملیات ها در ماشین آلات (مشکل توالی) به منظور به حداقل رساندن مدت کلی برنامه زمانبندی است، به عنوان مثال، زمان مورد نیاز برای تکمیل تمام مشاغل، است که به عنوان $MK = \max_i \{C_i\}$ تعریف می شود که در آن C_i زمان اتمام کار i است. با توجه به الگوی تکامل استفاده شده در الگوریتم های ژنتیک، ما به هر گونه راه حل از FJSP به عنوان یک فرد یا کروموزوم اشاره می کنیم.

داده های مسئله را می توان در یک جدول سازماندهی نمود، که در آن ردیف ها متناظر با عملیات و ستون مربوط به ماشین آلات است. نوشته های جدول ورودی، زمان پردازش، همانند مثال داده شده در جدول I. است. در این مثال، ما دارای انعطاف پذیری کلی هستیم. در یک سناریوی انعطاف پذیری نسبی، یک ورودی ∞ در جدول بدان معنی است که یک دستگاه نمی تواند عملیات مربوطه را اجرا کند، به عنوان مثال، به زیر مجموعه ای از ماشین های سازگار با آن عملیات تعلق ندارد.

جدول ۱: جدول زمان پردازش

	M_1	M_2	M_3	M_4
O_{11}	7	6	4	5
O_{12}	4	8	5	6
O_{13}	9	5	4	7
O_{21}	2	5	1	3
O_{22}	4	6	8	4
O_{23}	9	7	2	2
O_{31}	8	6	3	5
O_{32}	3	5	8	3

۲. مرور نوشته ها

برای شروع بررسی نوشته های خود، ما باید یادآوری کنیم که الگوریتم های دقیق برای حل FJSP در نمونه های کوچک موثر نیستند. در واقع، روش های دقیق بر اساس نمایش گراف فصلی از مسئله توسعه یافته است، اما آنها با بیش از ۲۰ شغل و ۱۰ دستگاه [۱] قابل استفاده نیستند. بنابراین تمرکز مرور ما تنها روی روش های اکتشافی است. اول، ما باید ادعا کنیم که برای دانش ما، هیچ الگوریتم تقریبی برای راه حل های تولید FJSP با یک فاصله تضمین شده از راه حل بهینه شناخته شده نیست. برخی نتایج شناخته شده در مورد تقریباً JSP را به راحتی قابل ارتقا به

FJSP نیست. تنها زمانی که تعداد ماشین آلات و حداکثر تعداد عملیات در هر شغل ثابت باشد، Jansen و همکاران. [۸] یک طرح زمانی تقریب خطی را ارائه داده اند.

روش های اکتشافی متعددی از جمله قوانین توزیع، جستجوی محلی و فرا اکتشاف های مانند جستجوی ممنوع، بازپخت شبیه سازی شده و الگوریتم های ژنتیک در سال های اخیر برای FJSP توسعه یافته اند. آنها را می توان به دو دسته اصلی تقسیم نمود: روش سلسله مراتبی و رویکرد یکپارچه شده. روش سلسله مراتبی، برای حل مشکل توسط تجزیه آن به دنباله ای از زیر مسئله ها، با مشکل کاهش یافته تلاش می کند. یک تجزیه معمولی، انتساب-آنگاه-توالی آمده از مشاهدات بدیهی است که زمانی که انتساب انجام می پذیرد، مسئله توالی حاصل یک JSP می شود. این رویکرد توسط [4] Chambers and Barnes, [9] Brandimarte در میان افراد دیگر پیروی شده است. آنها این مشکل انتساب را با استفاده از برخی از قوانین توزیع، و سپس JSP حاصل با استفاده از ابتکارات جستجوی مختلف ممنوع حل نموده اند.

حل روش یکپارچه شده، خیلی سخت تر است، اما در کل به نتایج بهتری می رسد که در Vaessens و همکاران [11] و Dazere و Hurink [12]. PaiiUi همکاران [۱۳] و Mastrolilli و Gambardella [3]. گزارش شده است. آنها همه یک رویکرد یکپارچه که پیشنهاد دهنده جستجوهای ممنوع مختلف برای حل این مشکل است اتخاذ نموده اند. در میان آنها، Mastrolilli و Gambardella، نتایج محاسباتی را نشان دادند که اثبات می کند که جستجوی ممنوع از هر روش اکتشافی دیگر که تا کنون توسعه یافته است، نظر زمان محاسبات و کیفیت راه حل، بهتر عمل می کند.

به تازگی، الگوریتم های ژنتیک با موفقیت برای حل FJSP اتخاذ شده اند، همانطور که توسط تعداد فزاینده ای از مقالات در مورد این موضوع ثابت شده است. مرتبط ترین آثار، Chen و همکاران. [۵]، Jia و همکاران. [۱۴]، Ho و Tay (۱۵) و، به طور خاص Kacem. و همکاران [۶] است که اساس کار ما را ارائه نموده اند. آنها، همه، رویکردهای یکپارچه هستند و برای طرح کدگذاری مختلف، تولید جمعیت اولیه، انتخاب کروموزوم و استراتژی تولید فرزندان متفاوت از یکدیگر هستند. Chen و همکاران. [۵] نمایش کروموزوم را به دو بخش تقسیم نموده اند. اولی،

سیاست مسیریابی و دومی دنباله عملیات را در هر دستگاه را تعریف می کند. Jia و همکاران. [۱۴] یک GA اصلاح شده را پیشنهاد نمودند که قادر به حل مشکلات برنامه ریزی توزیع شده است و می تواند برای FJSP اقتباس شود. Ho و Tay [۱۵] یک روش کارآمد به نام GENACE را بر اساس معماری فرهنگی تکاملی برای حل FJSP با چرخش دوباره پیشنهاد دادند. در نهایت Kacem و همکاران. [۶] از یک نمایش کروموزوم استفاده نموده اند که ترکیبی از اطلاعات مسیر یابی و تعیین توالی است و یک رویکرد را توسط موقعیت یابی برای پیدا کردن تکالیف اولیه نویدبخش توسعه می دهد. بنابراین، توزیع قوانین برای توالی عملیات اعمال می شوند. هنگامی که این جمعیت اولیه یافت شود، آنها اپراتورهای متقاطع و جهش را به طور مشترک برای تغییر تکالیف و توالی آنها اعمال می کنند که تولیدکننده فرد بهتر به عنوان نسلی است که ادامه دارد. ما این را در کار خود اتخاذ می کنیم که نشاندهنده بسیاری از گزینه های Kacem و همکاران [61]. است و برخی از بهبودها را در روش آنها نشان می دهد. علاوه بر این، آنها نتایج تجربی را تنها در برخی از موارد نمونه هدایت نمودند، و کار خود را در این مقاله با مطالعه گسترده محاسباتی کامل نمودند.

با استنتاج بررسی نوشته های ما، می خواهیم به ذکر کار مهم Tay و [Wibowo L16] بپردازیم که چهار بازنمود کروموزوم متفاوت را مقایسه می کنند که عملکرد آنها را در نمونه های مسئله ۱۰ * ۱۰ (۱۰ شغل، ۱۰ دستگاه) تست می شود. نتیجه آنها نشان می دهد که نمایش بیشتر کروموزوم توسط به هم پیوستن رشته های جزئی معنی دار تشکیل می شود که، اپراتورهای ژنتیکی مناسب تر را می توان توسعه داد، و نتایج بهتری را می توان به دست آورد. آنها همچنین برخی از اظهارات را در بهترین نرخ های اپراتورهای متقاطع و اپراتورهای جهش برای تولید فرزندان گزارش نمودند.

۳. یک GA برای FJSP

GA یک الگوریتم جستجوی محلی است که از الگوی تکامل پیروی می کند. با شروع از یک جمعیت اولیه، این الگوریتم برای اپراتورهای ژنتیکی به منظور تولید فرزندان اعمال می شود (در اصطلاحات محلی، مربوط به بررسی همسایگی است) که احتمالاً بیشتر از اجداد خود تناسب می یابد. در هر تولید، (تکرار)، هر فرد جدید (کروموزوم)

مربوط به یک راه حل است. به عنوان مثال، یک برنامه زمانی برای نمونه FJSP معین. قدرت GA با توجه به دیگر الگوریتم های جستجوی محلی با توجه به این حقیقت است که در یک چارچوب GA، استراتژی های بیشتر را می توان با هم برای پیدا کردن افراد به منظور اضافه نمودن حوزه بلوغ در مرحله اولیه جمعیت و در مرحله نسل پویا اتخاذ نمود. بنابراین فضای بیشتر متغیر جستجو را می توان در هر مرحله الگوریتم کشف نمود. ساختار کلی GA ما به شرح زیر است:

جدول ۲: رویکردی توسط موقعیت یابی (به روزسازی های بار کاری دستگاه به صورت برجسته)

	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4
O_{11}	7	6	4	5	7	6	4	5	7	6	4	5
O_{12}	4	8	5	6	4	8	9	6	4	8	9	6
O_{13}	9	5	4	7	9	5	8	7	13	5	8	7
O_{21}	2	5	1	3	2	5	5	3	6	5	5	3
O_{22}	4	6	8	4	4	6	12	4	8	6	12	4
O_{23}	9	7	2	2	9	7	6	2	13	7	6	2
O_{31}	8	6	3	5	8	6	7	5	12	6	7	5
O_{32}	3	5	8	3	3	5	12	3	7	5	12	3

۱. برنامه نویسی: ژن های کروموزوم ها، توصیف کننده انتساب عملیات ها به ماشین آلات و نظم است که در آن، آنها که در کروموزوم به نظر می رسد که دنباله ای از عملیات را توصیف می کند. هر کروموزوم نشان دهنده یک راه حل برای این مشکل است.
۲. جمعیت اولیه: کروموزوم های اولیه از ترکیب دو روش تخصیص (حداقل کلی و جایگشت تصادفی مشاغل و ماشین آلات) و ترکیبی از سه قانون توزیع (MOR, Random, MWR) برای توالی به دست می آیند.
۳. ارزیابی تناسب: مدت کلی برنامه زمانبندی برای هر کروموزوم در نسل فعلی محاسبه می شود.
۴. انتخاب: در هر تکرار، بهترین کروموزوم ها برای تولید مثل توسط یکی در بین سه روش مختلف انتخاب می شوند، یعنی تورنمنت باینری، تورنمنت با اندازه n و رتبه بندی خطی.
۵. تولید فرزندان: تولید جدید با تغییر تخصیص عملیات به ماشین ها (تقاطع انتساب، انتساب جهش، جهش هوشمند) با تغییر توالی عملیات ها (POX متقاطع و جهش PPS) به دست می آیند. این قواعد، عملی بودن افراد جدید را

حفظ می کند افراد جدید تا زمانی که به حداکثر تعداد ثابت افراد دست یابیم، در حال تولید هستند. در رویکرد سوم ما، تنها افراد جدید، حوزه جفت گیری را برای نسل بعدی، در هر گام الگوریتم تشکیل می دهند.

۶. معیار توقف: تعداد ثابت تولید به دست می آید. اگر معیار توقف برآورده شود، این الگوریتم به پایان می رسد و بهترین کروموزوم به همراه برنامه زمانبندی مربوطه به عنوان خروجی داده می شود. در غیر این صورت، این الگوریتم دوباره مراحل ۳-۵ را تکرار می کند.

برای توصیف GA ما، ما جزئیات را در ادامه مراحل مختلف و یا الگوریتم ارائه می دهیم. به طور خاص، ما در بخش ۳,۱ چگونه تولید جمعیت اولیه را توصیف می کنیم؛ در بخش ۳,۲ طرح برنامه نویسی در حال حاضر اتخاذ شده را ارائه می دهیم؛ در بخش ۳,۳ ما تابع ارزیابی تناسب را توصیف می کنیم؛ در بخش ۳,۴ گزارش استراتژی ها برای انتخاب افراد به منظور تولید مثل فراهم شده است و در نهایت، در بخش ۳,۵ ما اپراتورهایی همانند فرموله کردن و متقاطع، برای تغییر تخصیص و تعیین توالی عملیات را توصیف می کنیم.

3.1. جمعیت اولیه

جمعیت اولیه به دنبال رویکرد موقعیت یابی توسط Kacem و همکاران ایجاد می شود. [۶]. این رویکرد، بار پردازش و بار کاری ماشین آلات، یعنی مجموع زمان پردازش این عملیات اختصاص داده شده به هر دستگاه را در نظر می گیرد. این روش شامل پیدا کردن برای هر عمل، هر دستگاه با حداقل زمان پردازش، تثبیت آن انتساب، و اضافه نمودن این زمان به هر ورودی پس از آن در همان ستون (ماشین بار کاری به روز رسانی)، همانطور که در جدول (۲) نشان داده شده است، می باشد که در آن مقادیر ضخیم با به روز رسانی بار کاری مطابقت دارد. از آنجا که این روش به شدت وابسته به نظمی است که در آن عملیات و ماشین در جدول، معین هستند. ما کمی آن را در دو روش زیر تغییر می دهیم:

- AssignmentRule1 جستجو برای حداقل کلی در جدول زمان پردازش.
- AssignmentRule2 به طور تصادفی، جایگشت مشاغل و ماشین آلات در جدول.

جدول ۳ انتسابات اولیه توسط AssignmentRule (به روز سازی های بار کاری ماشین به صورت برجسته)

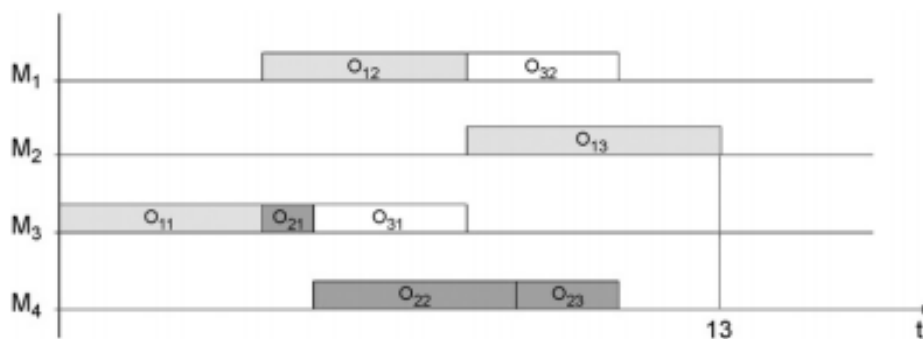
	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4
O_{11}	7	6	4	5	7	6	5	5	7	6	5	7	7	6	4	5
O_{12}	4	8	5	6	4	8	6	6	4	8	6	8	4	8	5	6
O_{13}	9	5	4	7	9	5	5	7	9	5	5	9	9	5	4	7
O_{21}	2	5	1	3	2	5	1	3	2	5	1	5	2	5	1	3
O_{22}	4	6	8	4	4	6	9	4	4	6	9	6	...	4	6	8
O_{23}	9	7	2	2	9	7	3	2	9	7	3	2	9	7	2	2
O_{31}	8	6	3	5	8	6	4	5	8	6	4	7	8	6	3	5
O_{32}	3	5	8	3	3	5	9	3	3	5	9	5	3	5	8	3

جدول ۴ انتسابات اولیه توسط AssignmentRule2 (به روز سازی های بار کاری ماشین به صورت برجسته)

	M_4	M_1	M_3	M_2	M_4	M_1	M_3	M_2	M_4	M_1	M_3	M_2	M_4	M_1	M_3	M_2
O_{31}	5	8	3	6	5	8	3	6	5	8	3	6	5	8	3	6
O_{32}	3	3	8	5	3	3	11	5	3	3	11	5	3	3	8	5
O_{11}	5	7	4	6	5	7	7	6	8	7	7	6	5	7	4	6
O_{12}	6	4	5	8	6	4	8	8	9	4	8	8	6	4	5	8
O_{13}	7	9	4	5	7	9	7	5	10	9	7	5	...	7	9	4
O_{21}	3	2	1	5	3	2	4	5	6	2	4	5	3	2	1	5
O_{22}	4	4	8	6	4	4	11	6	7	4	11	6	4	4	8	6
O_{23}	2	9	2	7	2	9	5	7	5	9	5	7	2	9	2	7

AssignmentRule شروع را از عملی پیش بینی می کند که مربوط به حداقل کلی در جدول است. به عنوان یک نتیجه، به روز رسانی بار کاری ماشین در هر عملیات دیگر انجام می شود، یعنی، هر ورودی دیگر در همان ستون. AssignmentRule2 به طور تصادفی مشاغل و ماشین آلات را قبل از اعمال روش توسط موقعیت یابی پیش بینی می کند. مزیت استفاده از AssignmentRule2 اینست که تکالیف اولیه متفاوت را در اجراهای متفاوت از الگوریتم با کاوش بهتر در فضای جستجو می یابد. پذیرش ترکیبی از این دو قانون، مجموعه اولیه از تکالیف را

تولید می کند. برای مثال، ۱۰ درصد از جمعیت اولیه می تواند توسط قانون ۱ و ۹۰٪ توسط قانون ۲ تولید شود. یکی از اهداف تحقیق ما، در واقع، پیدا کردن یک تنظیم قوی از این ترکیب برای موارد مختلف کلاس های مختلف مسئله. به عنوان مثال نشان دادن اینکه چگونه AssignmentRule1 و AssignmentRule2 کار می کنند در جداول ۳ و ۴ به ترتیب (مقادیر ضخیم، گزارش به روز رسانی حجم کار ماشین است) نشان داده شده است. در هر دو نمونه، آخرین جدول نشان داده شده، نشان دهنده تکالیف نهایی به دست آمده است.



شکل ۱. نمودار Gantt

هنگامی که تکالیف حل و فصل شوند، ما باید تعیین کنیم که چگونه توالی عملیات ها در دستگاه صورت می گیرد. بدیهی است، توالی در صورتی امکان پذیر است که اولویت را در میان عملیات همان کار محدود می کند، به عنوان

مثال، عملیات $O_{i,j+1}$ نمی تواند قبل از عملیات $O_{i,j}$ پردازش شود.

در GA ما، توالی تکالیف اولیه توسط ترکیبی از سه قوانین توزیع شناخته شده به دست می آید:

(a) انتخاب تصادفی یک شغل (تصادفی)؛

(b) بیشتر کار باقی مانده (MWR)؛

(c) بیشترین تعداد از عملیات باقی مانده (MOR).

به عنوان مثال اولیه، کروموزوم ها را می توان با قاعده (a) 40 درصد توسط قانون (b) ۴۰ درصد توسط قانون (c) تولید نمود. باز هم، پیدا کردن یک تنظیم محکم و یا ترکیبی از این سه قانون برای طبقات مختلف از موارد مشکل، یکی از اهداف تحقیقات ما است. به عنوان مثال، با استفاده از قانون MWR برای تکالیف جدول ۳، ما راه حل اولیه

زیر را به دست می آوریم، که مدت کلی برنامه زمانبندی برابر با ۱۳ است، و با استفاده از نمودار گانت در شکل ۱ ترسیم می شود.

$$S = (O_{11}, M_3), (O_{12}, M_1), (O_{21}, M_3), (O_{22}, M_4), (O_{31}, M_3), (O_{13}, M_2), (O_{32}, M_1), (O_{23}, M_4).$$

3.2. برنامه نویسی

به منظور اجرای GA ما، ما نیاز به نشان دادن یک برنامه زمانی نمادین، به عنوان مثال، با یک رشته داریم. ما از نمایش فهرست کار توالی پیشنهاد شده توسط [6] Kacemerar، استفاده می کنیم که در آن یک رشته سه تایی (I, j, k) یکی برای هر عمل تشکیل می شود، که در آن

- I ، کاری است که عمل به آن تعلق دارد؛
- J ، تعداد پیشرونده که آن عمل در کار i است؛
- K ، ماشین اختصاص داده شده به این عملیات است.

طول رشته برابر با تعداد کل عملیات است. در مثال نشان داده شده در شکل ۱، راه حل

$$S = (O_{11}, M_3), (O_{12}, M_1), (O_{21}, M_3), (O_{22}, M_4), (O_{31}, M_3), (O_{13}, M_2), (O_{32}, M_1), (O_{23}, M_4)$$

توسط رشته

(1,1,3)	(1,2,1)	(2,1,3)	(2,2,4)	(3,1,3)	(1,3,2)	(3,2,1)	(2,3,4)
---------	---------	---------	---------	---------	---------	---------	---------

 نمایش داده می شود.

3.3 ارزیابی تناسب

تابع ارزیابی تناسب برای کروموزوم ها، با مدت کلی برنامه زمانبندی راه حل که آنها نشان می دهند همزمان می شود. بنابراین، از آنجایی که ما در جستجوی راه حل های ROR با مقادیر پایین تر از مدت کلی برنامه زمانبندی هستیم، تکامل ژنتیکی را با کروموزوم ها با تناسب پایین تر ترجیح داده می شود. برای هر نسل، همه کروموزوم ها بررسی می شوند. و بهترین فرد ثبت شده است.

3.4 انتخاب

مرحله انتخاب، مسئول انتخاب کروموزوم ها برای تولید مثل است. در روش ما، معیار استفاده شده برای انتخاب کروموزوم هایی که باید در حوزه بلوغ گنجانده شود را می توان در میان سه روش انتخاب به خوبی شناخته شده در نوشته های GA انتخاب نمود: تورنمنت باینری، تورنمنت با اندازه و رتبه بندی خطی.

- تورنمنت دودویی: دو نفر به طور تصادفی از افراد انتخاب می شوند و بهترین آنها را برای تولید مثل انتخاب می شود.
- تورنمنت با اندازه n : فرد برای تولید مثل که در میان یک عدد تصادفی از افراد انتخاب می شود.
- رتبه بندی خطی: افراد با توجه به تناسب خود طبقه بندی می شوند و یک رتبه $r_i \in \{1, \dots, N\}$ به هر یک از آنها اختصاص داده می شود. که در آن N اندازه جمعیت است. بهترین فرد به رتبه N می رسد در حالی که بدترین رتبه برابر ۱ می شود. بنابراین،

$$p_i = \frac{2r_i}{N(N+1)}, \quad i = 1, \dots, N$$

احتمال انتخاب فرد i ام در رتبه سفارش است.

از حوزه جفت گیری، جفت کروموزوم ها به طور تصادفی برای تولید مثل انتخاب می شوند. در رویکرد ما، حوزه جفت گیری به طور کامل در هر تکرار تمدید می شود، و سپس معیار انتخاب شده باید تکرار شود تا زمانی که تعداد افراد در حوزه جفت گیری برابر با اندازه جمعیت است. تجربه محاسباتی نشان میدهد که تورنمنت دودویی نتایج بهتری را در میان سه روش مختلف ارائه می دهد.

3.5 تولید فرزندان

هنگامی که کروموزوم ها برای تولید مثل انتخاب می شوند، اپراتورهای متقاطع و ژنتیکی برای تولید فرزندان داده می شوند. اپراتور متقاطع برای جفت کروموزوم ها اعمال می شود، در حالی که اپراتور جهش برای افراد تک می شوند. ما بین دو نوع از اپراتورها تمایز قائل می شویم:

- انتساب اپراتورها،
- تعیین توالی

تنها اپراتورهای انتساب، ویژگی انتساب کروموزوم ها را تغییر می دهند، یعنی توالی عملیات در فرزندان حفظ می شود. انتساب متقاطع، فرزندان را با تبادل انتساب زیر مجموعه ای از عملیات بین این دو والدین تولید می کند. جهش انتساب تنها انتساب را برای یک عملیات تک در والدین تنها مبادله می کند. در هر دو مورد، عملیات هایی که باید تبادل شود، به طور مستقل انتخاب می شوند. در جهش هوشمند، ما یک عملیات را بر روی دستگاه با حداکثر بار کاری انتخاب می کنیم و آن را به ماشین حجم کار مینیمم در صورت سازگاری، اختصاص می دهیم.

اپراتورهای ترتیب بندی، تنها ترتیب و توالی عملیات را در کروموزوم های والدین تغییر می دهد، یعنی، انتساب عملیات به دستگاه ها در فرزندان حفظ می شود. در استفاده از اپراتورهای توالی، ما باید به محدودیت های اولویت در میا عملیات همان کار توجه کنیم. اتخاذ یک الگوریتم اصلاح کننده برای تغییر غیرعملی فرزندان بسیار وقت گیر است، و از این رو طراحی اپراتورها به گونه ای که محدودیت های اولویت نقض نشود، بهتر است. همانند kacem همکاران [6]. A1، اپراتورهای تقاطع حفظ اولویت مبتنی بر ترتیب (POX) و جهش جابجایی حفظ اولویت Lee (PPS) و همکاران. [۱۷۱] اتخاذ می شوند.

POX، دو فرزند را با شروع از دو والدین تولید می کند. POX یک عملیات را از والدین اول انتخاب می کند، در اولین فرزند، تمام عملیات های کاری را کپی می کند که عملیات انتخاب شده متعلق به آن است، سپس این فرد جدید را کامل با عملیات باقی مانده، با همان ترتیبی که در والدین دوم ظاهر می شود، تکمیل می کند. این روند متقارن برای والدین و فرزند دوم تکرار می شود. توجه داشته باشید که POX محدودیت های توالی را حفظ می کند. PPS، یک عمل را از یک کروموزوم والدین انتخاب می کند و آن را به یک موقعیت دیگری حرکت می دهد در حالیکه مراقب محدودیت های مقدم که برای آن عملیات است. اپراتور PPS تنها در صورتی که اعمال می شود که جهش کیفیت راه حل را بهبود بخشد.

فاز تولید فرزندان زمانی پایان می یابد که به حداکثر تعداد افراد در حوزه جفت گیری برسند، یعنی، نسل جدید یافت می شود. این الگوریتم زمانی به پایان می رسد که به حداکثر تعداد تولید می رسد، و بهترین فردی، همراه با برنامه مربوطه، به عنوان خروجی داده می شود.

4 نتیجه محاسباتی

- رویه GA شرح داده شده در بخش 3 در پردازنده 8 MHz پنتیوم IV. پیاده سازی می شود و در میزان زیادی از نمونه های مسئله نوشته ها. (<http://www.idsia.ch/monaldo>) آزمایش می شود. بهترین نتایج پس از پنج اجرا از جمعیت های مختلف اولیه انتخاب می شوند. مجموعه های مختلفی از موارد مشکل در نظر گرفته می شوند.
1. Brandimarte [12]: مجموعه داده ها شامل مسائلی با تعداد شغل ها در گستره 10 تا 20، تعداد ماشین آلات اعم از 4 تا 5 و تعدادی از عملیات برای هر شغل در محدوده 5 تا 15 می شود.
2. Dauzere-Peres و Pauli [12]: مجموعه داده ها شامل 18 مسئله آزمون با شمار شغل ها اعم از 10 تا 20، تعداد ماشین آلات در محدوده 5 تا 10 و تعدادی از عملیات ها برای هر یک از شغل های مختلف از 5 تا 25 می شود.
3. Barnes و Chambers [4]: مجموعه داده ها شامل 21 مسئله به دست آمده از کلاسیک mtIO به عنوان مثال JSP؛ Fisher و Thompson و نمونه های [181] la40. la24 مطرح شده توسط Lawrence [19] می شود. شمار F مشاغل در محدوده 10 تا 15 و تعداد ماشین ها در محدوده از 11 تا 18 است
4. Hurink 4. و همکاران. (13): مجموعه داده ها شامل 129 مسئله آزمون ایجاد شده از 43 نمونه کلاسیک JSP است. آنها مسئله آزمون را به سه زیر مجموعه، EDala، RDma و VData. با توجه به تعداد متوسط از جایگزین ماشین آلات برای هر عمل تقسیم نمودند. شمار مشاغل در محدوده 6 تا 30 و تعداد ماشین ها در محدوده 5 تا 15. در آزمایش ما، ما مقادیر مختلف را برای یک لیست از پارامترهای الگوریتم تست می کنیم. و تجربه محاسباتی ثابت می کند که انتخاب تورنمنت باینری و مقادیر زیر بیشتر موثر هستند:
- جمعیت کلی: 5000؛
 - تعداد تولید: 1000؛
 - نرخ تکالیف اولیه با روش حداقل کلی: 10 درصد.
 - سرعت را از تکالیف اولیه با استفاده از روش تصادفی: 90٪؛

- نرخ توالی های اولیه با قاعده تصادفی: ۲۰٪؛
- نرخ توالی اولیه با قاعده MWR: ۴۰٪؛
- نرخ توالی اولیه با قاعده MOR: ۴۰٪؛
- احتمال متقاطع POX: ۴۵ درصد؛
- احتمال انتساب متقاطع: ۴۵٪؛
- احتمال جهش PPS: ۲٪؛
- احتمال جهش انتساب: ۲٪؛
- احتمال جهش هوشمند تخصیص: ۶٪.

جدول ۵، مقایسه GA ما برای الگوریتم های پیشنهاد شده توسط Chen و همکاران [5]، Ho و Tay [15] و Jia و همکاران [14] برای موارد مسئله از [9] Brandimarte است. ستون اول، گزارش نام مثال، ستون دوم و سوم، گزارش تعداد شغل و تعداد ماشین آلات برای هر نمونه، ستون چهارم، گزارش حد پایین (۳) شناخته شده، ستون پنجم، گزارش GA بهترین مدت کلی برنامه زمانبندی ما در پنج اجرا است. ستون های باقی مانده، گزارش بهترین نتایج از سه الگوریتمی است که ما به همراه هم با انحراف نسبی با توجه به الگوریتم مقایسه می کنیم. انحراف نسبی به صورت زیر تعریف می شود

$$dev = [(MK_{comp} - MK_{GA}) / MK_{comp}] \times 100\%,$$

که در آن MK_{GA} ، مدت کلی برنامه زمانبندی به دست آمده توسط الگوریتم ما و MK_{comp} مدت کلی برنامه زمانبندی از الگوریتمی است که ما با آن مقایسه می کنیم. نتایج نشان می دهد که الگوریتم ما، سه الگوریتم دیگر ژنتیک را انجام می دهد. در جدول، نتایج گزارش شده در ستون ۶ و ۱۰ توسط پیاده سازی الگوریتم Chen همکاران [۵]، و با تطبیق این الگوریتم FJSP و Jia به دست آمده است.

در جدول شماره ۶، بهترین نتایج خود را در طول پنج اجرای GA با بهترین نتایج جستجوی ممنوع Mastrolilli و [3] Gambardella در ۱۰ نمونه FJSP از Brandimarte مقایسه می کنیم. ستون هفتم، گزارش انحراف

نسبی الگوریتم ما با توجه به آنها است مقادیر LB، درون پرانتز، مطلوب هستند. نتایج نشان می دهد که کیفیت راه حل ها قابل مقایسه است.

جدول ۷، نتایج محاسباتی در چهار کلاس نمونه را نشان می دهد.

جدول ۵: مقایسه با دیگر GAها در نمونه های FJSP از Brandimarte

Name	<i>n</i>	<i>M</i>	LB	GA	Chen	<i>dev</i> (%)	GENACE	<i>dev</i> (%)	Jia	<i>dev</i> (%)
Mk01	10	6	36	40	40	0	41	+2.44	40	0
Mk02	10	6	24	26	29	+10.34	29	+10.34	28	7.14
Mk03	15	8	204	204	204	0	204	0	204	0
Mk04	15	8	48	60	63	+4.76	67	+10.45	61	1.64
Mk05	15	4	168	173	181	+4.42	176	+1.70	176	1.70
Mk06	10	15	33	63	60	-5.0	68	+7.35	62	-1.61
Mk07	20	5	133	139	148	+6.08	148	+3.38	145	4.14
Mk08	20	10	523	523	523	0	523	0	523	0
Mk09	20	10	299	311	308	-0.97	328	+5.18	310	-0.32
Mk10	20	15	165	212	212	0	231	+8.23	216	1.85
Average improvement						+1.96		+5.18		+1.45

جدول ۶: مقایسه با جستجوی ممنوع Mastrolli and Gambardella در ۱۰ نمونه FJSP از Brandiamrte

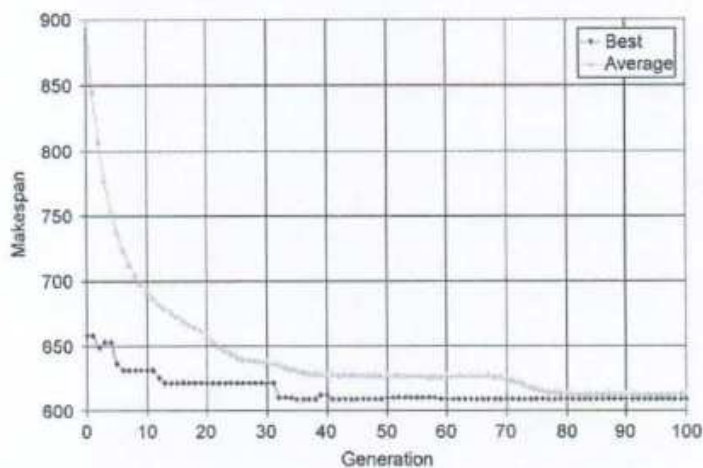
Name	<i>n</i>	<i>M</i>	LB	GA	M.G.	<i>dev</i> (%)
Mk01	10	6	36	40	40	0
Mk02	10	6	24	26	26	0
Mk03	15	8	(204)	204	204	0
Mk04	15	8	48	60	60	0
Mk05	15	4	168	173	173	0
Mk06	10	15	33	63	58	-8.62
Mk07	20	5	133	139	144	+3.47
Mk08	20	10	(523)	523	523	0
Mk09	20	10	299	311	307	-0.64
Mk10	20	15	165	212	198	-7.07

جدول ۷: خطای میانگین نسبی (MRE) روی بهترین مرز پایینی شناخته شده

Data set	Num.	Alt.	GA (%)	Chen (%)	Jia (%)
Brandimarte	10	2.59	17.53	19.55	19.11
Dauzère-Pérés and Paulli	18	2.49	7.63	7.91	10.62
Barnes and Chambers	21	1.18	29.56	38.64	29.75
Hurink EData	43	1.15	6.00	5.59	9.01
Hurink RData	43	2	4.42	4.41	8.34
Hurink VData	43	4.31	2.04	2.59	3.24

ستون اول، گزارش مجموعه ای از داده ها، ستون دوم تعداد نمونه برای هر کلاس، ستون سوم، متوسط تعداد ماشین های جایگزین در عمل است. سه ستون بعدی، گزارش خطای میانگین نسبی از (MRE) بهترین راه حل به دست آمده توسط GA، چن و همکاران. [۵] و جیا و همکاران. [۱۴]، به ترتیب با توجه به شناخته شده ترین کران پایین است. خطای نسبی (RE) % به صورت $RE = [(MK - LB) / LB \times 100]\%$ تعریف می شود که در آن MK، بهترین مدت کلی برنامه زمانبندی به دست آمده توسط الگوریتم گزارش شده و LB به عنوان بهترین مرز پایینی شناخته شده است. این جدول نشان می دهد که الگوریتم ما با یک درجه بالاتر از انعطاف پذیری (Hurink VOata) قوی تر است. علاوه بر این، نتیجه نشان می دهد که الگوریتم ما بهتر از دو تا دیگر از الگوریتم های ژنتیکی است.

در شکل ۲ کاهش مدت کلی برنامه زمانبندی متوسط و بهترین اجرا در مدت کلی برنامه زمانبندی با پنج اجرا، برای کلاس مسئله آزمون la01 با ۱۰ شغل و ۵ ماشین آلات آمده است [۱۹]. الگوریتم ما مدت کلی برنامه زمانبندی متوسط را بسیار سریع را بهبود می بخشد و بهترین مدت کلی برنامه زمانبندی، برابر با ۶۰۹، پس از ۳۵ نسل به دست می آید و مطلوب است.



شکل ۲. کاهش طول کلی برنامه زمانبندی

5 نتیجه گیری و تحقیقات آینده

در این کار ما یک الگوریتم ژنتیک (GA) را برای مشکل زمانبندی انعطاف پذیر شغلی- فروشگاه (FJSP) توسعه داده ایم. مطالعات گسترده محاسباتی نشان می دهد که الگوریتم ما نسبت به GA های شناخته شده دیگر برای مشکل مشابه، دارای نتایج بهتری و قابل مقایسه با بهترین الگوریتم شناخته شده تا کنون است. به عنوان یک نتیجه، چارچوب GA به طور موثر، الگوریتمی کارآمد برای FJSP، در هنگام یکپارچه سازی استراتژی های مختلف برای انتخاب و تولید مثل، و همچنین روشی برای پیدا کردن یک جمعیت اولیه است. در آینده، تحقیق در مورد مسائل زیر جالب خواهد بود:

- استفاده از ترکیبی از معیارهای مختلف انتخاب برای انتخاب بهترین فرد از یک نسل از کروموزوم؛
- حفظ بهترین افراد از حوزه جفت گیری در یک نسل پی در پی از کروموزوم ها؛
- بهبود عملکرد جهش هوشمند با انتخاب یک عملیات بحرانی، به جای هر عملیات، در بحرانی ترین ماشین؛
- در نظر گرفتن سایر شاخص های عملکرد و محدودیت های فرایند.

References

- [1] Pinedo M. Scheduling: theory, algorithms and systems. Englewood cliffs, NJ: Prentice-Hall; 2002.
- [2] Garey MR, Johnson DS, Sethi R. The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research* 1976;1:117-29.
- [3] Mastrolilli M, Gambardella LM. Effective neighbourhood functions for the flexible job shop problem. *Journal of Scheduling* 1996;3:3-20.
- [4] Barnes JW, Chambers JB. Flexible Job Shop Scheduling by tabu search. Graduate program in operations research and industrial engineering. Technical Report ORP 9609, University of Texas, Austin; 1996. (<http://www.cs.utexas.edu/users/jbc/>).
- [5] Chen H, Ihlow J, Lehmann C. A genetic algorithm for flexible Job-shop scheduling. In: *IEEE international conference on robotics and automation*, Detroit; 1999. p. 1120-5.
- [6] Kacem I, Hammadi S, Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 2002;32(1):1-13.
- [7] Feige U., Scheideler C. Improved bounds for acyclic job shop scheduling. In: *Proceedings of the 30th annual ACM symposium on the theory of computing (STOC '98)*; 1998. p. 624-33.
- [8] Jansen K, Mastrolilli M, Solis-Oba R. Approximation algorithms for Flexible Job Shop Problems. In: *Lecture notes in computer science*, vol. 1776. *Proceedings of the fourth Latin American symposium on theoretical informatics*; Berlin: Springer. 2000. p. 68-77.
- [9] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research* 1993;41:157-83.
- [10] Paulli J. A hierarchical approach for the FMS scheduling problem. *European Journal of Operational Research* 1995;86(1):32-42.
- [11] Vaessens RJM, Aarts EHL, Lenstra JK. Job Shop Scheduling by local search. *COSOR Memorandum 94-05*. Eindhoven University; 1994.
- [12] Dauzère-Pérès S, Paulli J. An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. *Annals of Operations Research* 1997;70:281-306.
- [13] Hurink J, Jurish B, Thole M. Tabu search for the job shop scheduling problem with multi-purpose machines. *OR-Spektrum* 1994;15:205-15.
- [14] Jia HZ, Nee AYC, Fuh JYH, Zhang YF. A modified genetic algorithm for distributed scheduling problems. *International Journal of Intelligent Manufacturing* 2003;14:351-62.
- [15] Ho NB, Tay JC. GENACE: an efficient cultural algorithm for solving the Flexible Job-Shop Problem. *IEEE international conference on robotics and automation* 2004;1759-66.
- [16] Tay JC, Wibowo D. An Effective Chromosome Representation for evolving flexible job shop schedules, *GECCO 2004*. In: *Lecture notes in computer science*, vol. 3103. Berlin: Springer; 2004. p. 210-21.
- [17] Lee KM, Yamakawa T, Lee KM. A genetic algorithm for general machine scheduling problems. *International Journal of Knowledge-Based Electronic* 1998;2:60-6.
- [18] Fisher H, Thompson L. Probabilistic learning combinations of local job shop scheduling rules. In: Muth JF, Thompson GL, editors. *Industrial scheduling*. Englewood Cliffs, NJ: Prentice-Hall; 1968. p. 225-51.
- [19] Lawrence S. An experimental investigation of heuristic scheduling techniques. In: *Supplement to resource constrained project scheduling*, GSIA. Pittsburgh, PA: Carnegie Mellon University; 1984.



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