

بهینه‌سازی مجموع جریمه‌های هزینه دیرکرد و انرژی در مسأله زمانبندی ماشین‌های موازی ناهمگن به وسیله الگوریتم ممیتیک

جواد بهنامیان^۱، امیر افسر^۲

تاریخ دریافت: ۹۶/۶/۱۳ - تاریخ پذیرش: ۹۸/۹/۴

چکیده

در مطالعات مختلف مربوط به مسائل زمانبندی، معمولاً تمرکز بر برنامه‌ریزی ماشین‌ها و تخصیص کارها به ماشین‌ها و تعیین توالی کارها، به منظور بهینه‌سازی زمان اتمام کارها، می‌باشد. با توجه به ارتباط بین اقتصاد، انرژی و نگرانی‌های زیست محیطی، مصرف انرژی یکی از موارد مهم در برنامه‌ریزی سیستم‌های مختلف می‌باشد. در این مقاله یک مسأله زمانبندی ماشین‌های موازی ناهمگن که در آن سرعت پردازش هر کار روی هر یک از ماشین‌ها قابل تنظیم است، بررسی می‌شود و از آنجا که انرژی مصرفی ماشین‌ها با سرعت پردازش آن‌ها رابطه‌ای مستقیم دارد، هدف مسأله کمینه‌سازی مجموع هزینه‌های انرژی مصرفی و جریمه دیرکرد در تحویل تقاضای مشتریان می‌باشد. به منظور بهینه‌سازی مسأله، یک الگوریتم فراابتکاری ممیتیک و یک الگوریتم فراابتکاری ژنتیک پیشنهاد شده است و در پایان نتایج بدست آمده از دو الگوریتم فراابتکاری پیشنهادی را با یکدیگر و با نتایج حاصل از خروجی نرم افزار بهینه‌سازی گمز، مقایسه و تحلیل می‌نماییم.

کلمات کلیدی زمانبندی ماشین‌های موازی ناهمگن، هزینه دیرکرد، هزینه انرژی، الگوریتم ممیتیک

* دانشیار گروه مهندسی صنایع دانشکده مهندسی، دانشگاه بوعلی سینا، همدان؛ (نویسنده مسئول)

Behnamian@basu.ac.ir

** کارشناسی ارشد گروه مهندسی صنایع دانشکده مهندسی، دانشگاه بوعلی سینا، همدان

مقدمه

هدف از زمانبندی^۱، تخصیص کارها به ماشین آلات و تعیین توالی کارهای تخصیص یافته به هر ماشین به گونه‌ای که اهداف سیستم برآورده شود، می‌باشد. اهداف متعددی برای مسائل زمانبندی در نظر گرفته می‌شوند که معمولاً کمینه‌سازی زمان اتمام کلیه کارها^۲ جزء اهداف اصلی زمانبندی، می‌باشد. با توجه به قدرت مشتریان در دنیای امروز، سعی در جلب رضایت مشتریان ایجاب می‌کند که کلیه واحدهای تولیدی و خدماتی، تقاضای آن‌ها را به موقع برآورد سازند؛ پس دیرکرد^۳ در برآورد تقاضای مشتریان هزینه‌های مختلفی را برای واحدها به وجود می‌آورد که همین موضوع باعث می‌شود واحدهای مختلف سعی داشته باشند به گونه‌ای برنامه‌ریزی کنند که هزینه‌های دیرکرد در برآورد تقاضای مشتریان را حداقل سازند و بتوانند به موقع سفارش هر مشتری را به دست او برسانند. با توجه به محدودیت منابع انرژی، برنامه‌ریزی صحیح به منظور بهینه‌سازی مصرف انرژی، می‌تواند منافع بسیاری داشته باشد. امروزه صنایع مختلف نه تنها به دنبال افزایش بهره‌وری خود می‌باشند، بلکه صرفه‌جویی در مصرف انرژی را در دستور کار خود قرار داده‌اند. پس یکی دیگر از اهداف مسائل زمانبندی، می‌تواند کاهش انرژی مصرفی باشد.

ماشین‌های موازی^۴ به عنوان یکی از زیرمجموعه‌های اصلی و پایه در فن زمانبندی، از جایگاه ویژه و مهمی برخوردارند و همواره زمانبندی این ماشین‌ها بر مبنای اهداف مختلف، مورد نظر محققین بوده است. در مسائل برنامه‌ریزی گاهی اوقات مناسب است که از چند ماشین که کار یکسانی را انجام می‌دهند، به صورت موازی استفاده کرد. در بسیاری از موارد برای از بین بردن گلوگاه‌ها می‌توان از ماشین‌های موازی استفاده کرد. حال آنکه این ماشین‌ها می‌توانند همگن و یا ناهمگن باشند. ماشین‌های همگن، مشابه یکدیگر کار می‌کنند و هر کار که به هر یک از ماشین‌ها تخصیص یابد، زمان پردازش یکسانی دارد. اما در ماشین‌های

-
1. Scheduling
 2. makespan
 3. Tardiness
 4. parallel-machine

ناهمگن، زمان پردازش هر کار روی هر ماشین، متفاوت از دیگری می‌باشد. در تحقیقات گذشته مربوط به مسائل زمانبندی ماشین‌های موازی، معمولاً سرعت پردازش ماشین‌آلات، ایستا فرض می‌شود؛ در این مقاله به بررسی ماشین‌های ناهمگن می‌پردازیم به گونه‌ای که سرعت پردازش هر کار روی هر ماشین، قابل تنظیم می‌باشد. امکان تنظیم سرعت پردازش ماشین‌آلات، باعث به وجود آمدن یک سیستم منعطف، به منظور برنامه‌ریزی ماشین‌های موازی، می‌شود. ما همواره سعی می‌کنیم مصرف انرژی را کاهش دهیم و میدانیم استفاده از سرعت پردازش بالاتر، مصرف انرژی بیشتری را در پی دارد؛ اما با توجه به اینکه سعی داریم تقاضای مشتریان را به موقع آماده‌سازی کنیم، گاهی استفاده از سرعت پردازش بالا، اقتصادی می‌باشد.

در این پژوهش به دنبال تعیین موارد زیر می‌باشیم:

الف) تخصیص کارها به ماشین‌ها

ب) تعیین توالی کارهای تخصیص یافته به هر یک از ماشین‌ها

ج) تعیین سرعت مناسب ماشین‌ها برای پردازش هر کار

دغدغه اصلی تئوری زمانبندی ماشین‌های موازی، پیدا کردن بهترین راه حل و یا نزدیکترین راه حل به راه حل بهینه به منظور تخصیص کارها به ماشین‌ها و تعیین بهترین راه حل و یا نزدیکترین راه حل به راه حل بهینه به منظور تعیین توالی کارهای تخصیص یافته به هر ماشین، می‌باشد. نمونه‌های بسیار زیادی از زمانبندی ماشین‌های موازی در زندگی روزمره ما دیده می‌شود. در بسیاری از موارد اضافه کردن یک یا چند ماشین دیگر که کاری مشابه با یکدیگر را انجام می‌دهند، باعث رفع مشکلات سیستم می‌شود. اضافه کردن تعداد مناسبی ماشین در یک سیستم تولیدی کار کارگاهی^۱ می‌تواند باعث افزایش انعطاف‌پذیری سیستم و در عین حال باعث کاهش مقدار زمان لازم برای پردازش کارها، شود (موراماتسو و همکاران، ۱۹۸۵). در یک ایستگاه پمپ بنزین استفاده همزمان از چند پمپ که همگی پمپ‌ها به یک مخزن بنزین متصل‌اند، باعث کاهش مدت زمان خدمت‌دهی می‌شود (درر و همکاران، ۱۹۸۷).

در سال ۱۹۷۷ (لنتسرا و همکاران، ۱۹۷۷) ثابت کردند که یک مسأله زمانبندی ماشین‌های موازی به منظور کمینه‌سازی هزینه‌های دیرکرد در تحویل تقاضای مشتریان، یک مسأله NP-hard می‌باشد. در سال ۱۹۷۸ برای مسأله زمانبندی ماشین‌های موازی به ازای تعداد ثابتی از ماشین‌آلات، (گری و جانسون، ۱۹۷۸) نشان دادند که این مسأله یک مسأله NP-hard می‌باشد. به این ترتیب (گری و جانسون، ۱۹۷۸) ثابت کردند که مسأله زمانبندی ماشین‌های موازی به ازای هر تعداد ماشین، حتی دو ماشین با هدف کمینه‌سازی هزینه‌های دیرکرد، یک مسأله NP-hard می‌باشد. تعدادی الگوریتم فراابتکاری شناخته شده برای حل مسائل زمانبندی ماشین‌های موازی به کار می‌روند (انگینوفی و پالوچی، ۲۰۰۷). (انگینوفی و پالوچی، ۲۰۰۷) یک مسأله زمانبندی ماشین‌های موازی با هدف کمینه کردن مجموع زمان‌های دیرکرد را بررسی و از یک رویکرد فراابتکاری ترکیبی^۱ به منظور حل مسأله مورد نظر استفاده کردند. (حسین‌زاده و کریمی، ۲۰۰۸) یک الگوریتم گروه ذرات گسسته^۲ را به منظور بهینه‌سازی مسأله زمانبندی ماشین‌های موازی به منظور کمینه‌سازی زمان اتمام کلیه کارها پیشنهاد کردند. (راجندران و سابور، ۲۰۰۴) یک الگوریتم فرا ابتکاری جست و جو ممنوع^۳ به منظور بهینه‌سازی مجموع موزون تاخیر در زمانبندی ماشین‌های موازی با در نظر گرفتن شکاف در کار، پیشنهاد دادند. (بالین، ۲۰۱۱) یک الگوریتم ژنتیک را به منظور بهینه‌سازی زمان پایان کلیه کارها در یک مسأله زمانبندی ماشین‌های موازی با ماشین‌های ناهمگن، پیشنهاد داد. (لین و همکاران، ۲۰۱۱) چند الگوریتم ابتکاری و یک الگوریتم ژنتیک را به منظور بهینه‌سازی مجموع موزون دیرکردها، مجموع موزون زمان اتمام کارها و زمان اتمام کلیه کارها در یک مسأله زمانبندی ماشین‌های موازی ناهمگن، مقایسه کرد. (رودریگز و همکاران، ۲۰۱۳) یک الگوریتم تکرار حریصانه^۴ به منظور ارائه راه حلی معقول برای ابعاد بزرگ مسأله زمانبندی ماشین‌های موازی ناهمگن با هدف کمینه‌سازی مجموع موزون زمان تکمیل کارها، پیشنهاد

-
1. hybrid metaheuristic approach
 2. discrete particle swarm optimization
 3. tabu search
 4. iterated greedy algorithm

کردند. (وردولد و هارکنز، ۲۰۰۲) دو الگوریتم فراابتکاری مبتنی بر جست و جوی همسایگی^۱ به منظور حل مسأله زمانبندی ماشین‌های موازی ناهمگن با هدف کمینه‌سازی مجموع موزون زمان تکمیل کارها، پیشنهاد کردند. (کیم و همکاران، ۲۰۰۲) یک الگوریتم شبیه‌سازی تبرید^۲ به منظور کمینه‌سازی مجموع تاخیر در یک مسأله زمانبندی ماشین‌های موازی ناهمگن، ارائه کردند. (توکلی مقدم و همکاران، ۲۰۰۹) یک الگوریتم ژنتیک به منظور کمینه‌کردن تعداد کارهایی که تحویل آن‌ها با دیرکرد روبرو می‌شود و زمان تکمیل کلیه کارها در یک مسأله زمانبندی ماشین‌های موازی ناهمگن با در نظر گرفتن زمان‌های آماده‌سازی وابسته به ماشین‌ها و وابسته به توالی، ارائه دادند. والادا و رویز، (۲۰۱۱) به منظور کمینه‌کردن زمان اتمام کلیه کارها، یک الگوریتم ژنتیک ترکیبی برای مسأله زمانبندی ماشین‌های موازی با زمان‌های آماده‌سازی وابسته به ماشین‌ها، پیشنهاد کردند. در مطالعه فانیول‌پیرو و همکاران، (۲۰۱۷) یک مسأله زمانبندی با ماشین‌های موازی ناهمگن مطرح گردید و به منظور بهینه‌سازی زمان تکمیل آخرین کار، یک الگوریتم دقیق-ابتکاری^۳ ارائه شد. (فخرزاد و رجائی، ۲۰۱۷) یک الگوریتم فراابتکاری شبیه‌سازی تبرید جهت بهینه‌سازی مسأله ماشین‌های موازی ناهمگن جهت پیاده‌سازی روش نگهداری و تعمیرات پیشگیرانه ارائه دادند.

بهبود بهره‌وری تولید، به حداقل رساندن زمان اتمام کلیه کارها، کاهش هزینه‌های تولید (هوگوین، ۲۰۰۵) و کاهش انرژی مصرفی (لیو و همکاران، ۲۰۱۳)، از جمله مهم‌ترین اهداف در برنامه‌ریزی و زمانبندی تولید، می‌باشند. مدل‌های ریاضی و الگوریتم‌های بهینه‌سازی به طور گسترده به منظور بهینه‌سازی مسائل مختلف برنامه‌ریزی و زمانبندی تولید، به کار گرفته می‌شوند (لین و لیائو، ۲۰۰۸). بهره‌وری در تولید یعنی تولید بدون اتلاف منابع. گرچه نقش مبانی اقتصاد در بهره‌وری تولید بسیار قابل توجه است، اما ملاحظات زیست محیطی در توسعه کارخانه‌ها امری ضروری می‌باشد (شروف و همکاران، ۲۰۱۳). در سال ۲۰۱۲، (دسپیس و همکاران، ۲۰۱۲) شیوه‌های صنعتی و اصول زیست محیطی را به منظور توسعه یک مفهوم

1. local search
2. simulated annealing
3. math heuristics algorithm

صحیح از اکوسیستم تولیدی، تجزیه و تحلیل کردند. بسیاری از روش‌ها و راه‌حل‌های استفاده شده برای حل مشکلات اتلاف انرژی، در طول فرآیندهای تولیدی می‌باشد (شروف و همکاران، ۲۰۱۳). در یکی از مطالعات مرتبط، (موزون و همکاران، ۲۰۰۷) به منظور به حداقل رساندن مصرف انرژی کل، پیشنهاد دادند که ماشین‌آلاتی که در برهه‌ای از زمان، بیکار می‌مانند، تا زمانی که دوباره به آن‌ها احتیاج پیدا شود، خاموش شوند. (فنگ و همکاران، ۲۰۱۱) یک برنامه‌ریزی چند هدفه عدد صحیح مختلط^۱ به منظور بهینه‌سازی زمان اتمام کلیه کارها و کمینه‌سازی مصرف انرژی، پیشنهاد کردند. (فنگ و لین، ۲۰۱۲) یک برنامه‌ریزی خطی عدد صحیح به منظور کمینه‌سازی مجموع جریمه‌های دیرکرد و هزینه انرژی پیشنهاد کردند که با استفاده از الگوریتم گروه ذرات به حل آن پرداختند. در تحقیق ژانگ و همکاران، (۲۰۱۴) مساله زمانبندی با توجه به مصرف انرژی و مباحث زیست محیطی مطرح شد و کمینه‌سازی مصرف برق و ردپای کربن^۲ با توجه به محدودیت‌های مساله، به عنوان تابع هدف در نظر گرفته شد. (شارما و همکاران، ۲۰۱۵) یک الگوریتم شبیه‌سازی تبرید را به منظور بهینه‌سازی هزینه الکتریسیته مصرفی در کارگاه ماشین‌های موازی با قابلیت تنظیم سرعت برای پردازش عملیات‌های مختلف، گسترش دادند. اخیراً، با توجه به فشار وارد بر کارخانه‌ها مبنی بر صرفه‌جویی در مصرف انرژی و همچنین در نظر گرفتن ملاحظات زیست-محیطی، زمانبندی تحت زمان مصرف انرژی (TOU)^۳، بسیار مورد توجه قرار گرفته است (ادا و همکاران، ۲۰۱۷). در پژوهش (مون و همکاران، ۲۰۱۳) یک مساله زمانبندی ماشین‌های موازی ناهمگن مطرح شد و یک الگوریتم بر پایه الگوریتم ژنتیک جهت بهینه‌سازی مجموع موزون هزینه‌های انرژی و زمان تکمیل آخرین کار، پیشنهاد گردید. (دینگ و همکاران، ۲۰۱۶) یک پژوهش مشابه را در نظر گرفتند و به منظور بهینه‌سازی هزینه‌های مصرف الکتریسیته، یک الگوریتم ابتکاری تولید ستون را پیشنهاد دادند. در مطالعه (ادا و همکاران، ۲۰۱۷) مساله زمانبندی ماشین‌های موازی ناهمگن به منظور بهینه‌سازی هزینه کل

-
1. mixed integer linear programming
 2. carbon footprint
 3. time of use

الگوریتم مصرفی در حوزه TOU با وجود امکان برش^۱ در پردازش کارها، به گونه‌ای که زمان تکمیل آخرین کار از حد بالایی تجاوز نکند، مد نظر قرار گرفت و دو الگوریتم ابتکاری جهت مسأله با ابعاد بزرگ، پیشنهاد شد.

در این پژوهش مسأله زمانبندی ماشین‌های موازی ناهمگن به گونه‌ای که مجموع هزینه‌های تاخیر در تحویل تقاضا و مجموع انرژی مصرفی حداقل شود بررسی می‌گردد. پس باید مشخص شود که هر کار به چه ماشینی تخصیص یابد و توالی کارهای تخصیص یافته به هر ماشین چگونه باشد و سرعت پردازش هر یک از ماشین‌ها برای پردازش هر کار، چقدر باشد. مشخص است که این مسأله در دسته مسائل NP-hard قرار می‌گیرد. به منظور حل مسأله، یک الگوریتم فراابتکاری ممتیک پیشنهاد می‌شود و سپس یک الگوریتم ژنتیک شبیه الگوریتم ممتیک پیشنهادی، ارائه می‌شود و در نهایت نتایج حاصل از حل مسأله به وسیله دو الگوریتم، تشریح می‌گردد و با یکدیگر مقایسه می‌شوند. از آنجا که مطالعات زمانبندی در حوزه انرژی به تازگی مورد توجه محققین قرار گرفته است و پیشینه زیادی ندارند، ارائه روش حل (الگوریتم) مناسب برای ارائه راه حل قابل قبول در زمان معقول برای این دسته از مسائل بسیار حائز اهمیت می‌باشد. در پژوهش حاضر تمرکز محقق بر ارائه راه حل جدید مبتنی بر الگوریتم ممتیک^۲ و الگوریتم ژنتیک^۳ و مقایسه و تحلیل نتایج حل می‌باشد که پیش از این برای ارائه راه حل در مسائل زمانبندی حوزه انرژی مشاهده نمی‌گردد.

ساختار این پژوهش بدین صورت است که در ادامه بخش ۱، خلاصه‌ای از تحقیقات انجام شده در حوزه مسأله مورد نظر ارائه می‌شود. در بخش ۲، شرح کامل مسأله همراه با مدل ریاضی و تعریف پارامترها و متغیرهای مربوطه آمده است. در بخش ۳، دو الگوریتم فراابتکاری ممتیک و ژنتیک به همراه یک مثال عددی جهت حل مسأله توسعه داده می‌شوند. در بخش ۴ به تجزیه و تحلیل و مقایسه نتایج حاصل از حل نمودهای مختلف از مسأله مورد نظر توسط الگوریتم‌های پیشنهادی و نرم‌افزار بهینه‌سازی گمز پرداخته می‌شود. در

1. preemption
2. memetic algorithm
3. genetic algorithm

نهایت در بخش ۵ نتیجه گیری نهایی و زمینه‌های تحقیقات آتی در این حوزه تشریح می‌شود.

تشریح مساله و مدل سازی

در نظر بگیرید n کار $\{j_1, j_2, \dots, j_n\}$ در یک کارگاه، به منظور پردازش وجود داشته باشند. به منظور پردازش کارها، m ماشین ناهمگن $\{M_1, M_2, \dots, M_m\}$ وجود دارند به گونه‌ای که انجام کارها تک مرحله‌ای می‌باشد و هر کار باید به یک ماشین تخصیص یابد و یک توالی از آن ماشین را به خود اختصاص دهد. هر کار می‌تواند روی هر یک از ماشین‌های موجود، پردازش شود. فرض می‌شود کلیه کارها در زمان صفر، در دسترس هستند و آماده پردازش می‌باشند. در ضمن فرض می‌شود زمان آماده سازی برای کارها، صفر می‌باشد و کارها بلافاصله بعد از اتمام پردازش خود، قابل تحویل هستند. به منظور پردازش هر کار، می‌توان سرعت پردازش هر یک از ماشین‌ها را به طور دلخواه، تنظیم نمود. زمان لازم برای پردازش هر یک از کارها برای وقتی که یک ماشین با سرعت استاندارد (نرمال) به پردازش آن کار بپردازد، مشخص است (p_j) . موعد تحویل هر کار (d_j) و هزینه جریمه دیرکرد در آماده سازی هر کار به ازای واحد زمان، مشخص است (w_j) . همانطور که گفته شد، به منظور پردازش هر کار، می‌توان سرعت پردازش ماشین را برای پردازش آن کار به طور دلخواه تعیین کرد. هرچه سرعت پردازش یک ماشین بیشتر باشد، انرژی مصرفی آن ماشین نیز بیشتر خواهد بود و بدیهی است که زمان لازم برای پردازش هر کار متناسب با سرعت پردازش آن کار، می‌باشد. مصرف انرژی به ازای هر سرعت در هر یک از ماشین‌ها در واحد زمان، مشخص است (e_{js}) . هدف از مسأله، تخصیص کارها به ماشین‌ها، تعیین توالی کارهای تخصیص یافته به هر یک از ماشین‌ها و تعیین سرعت مناسب ماشین‌ها برای پردازش هر کار می‌باشد به گونه‌ای که مجموع هزینه‌های تاخیر در تحویل تقاضا و مجموع انرژی مصرفی حداقل شود.

در ادامه پس معرفی مجموعه‌ها، مجموعه‌ها، پارامترها و متغیرهای تصمیم / مدل پیشنهادی ارائه خواهد شد.

مجموعه و شاخص‌ها

| | |
|---|---|
| J | مجموعه مربوط به کارها به منظور پردازش $\{j_1, j_2, \dots, j_n\}$ |
| M | مجموعه مربوط به ماشین‌ها $\{M_1, M_2, \dots, M_m\}$ |
| i | شاخص مربوط به ماشین‌ها $1 \leq i \leq m$ |
| j | شاخص مربوط به کارها $1 \leq j \leq n$ |
| l | شاخص مربوط به موقعیت (توالی) در ماشین‌ها $1 \leq l \leq n$ |
| s | شاخص مربوط به سرعت ماشین M_i ، جایی که s_i تعداد سرعت‌های مختلف ماشین i -ام است $1 \leq s \leq s_i$ |

پارامترها

| | |
|-----------|--|
| p_j | زمان لازم برای پردازش کار j تحت سرعت استاندارد |
| d_j | موعد تحویل کار j |
| w_j | جریمه دیرکرد کار j -ام در واحد زمان |
| S_{is} | s -امین سرعت پردازش ماشین M_i |
| e_{is} | هزینه انرژی مصرفی ماشین M_i در واحد زمان تحت سرعت S_{is} |
| t_{jis} | زمان لازم برای پردازش کار j روی ماشین i تحت سرعت S_{is} ($t_{jis} = p_j / S_{is}$) |

متغیرهای تصمیم

| | |
|------------|---|
| T_j | دیرکرد کار j -ام؛ اگر c_j را زمان تکمیل کار j در نظر بگیریم، داریم: $T_j = \max\{0, c_j - d_j\}$ |
| x_{jils} | اگر کار j به l -امین توالی از ماشین i تحت سرعت S_{is} تخصیص یابد، $x_{jils} = 1$ ، در غیر این صورت $x_{jils} = 0$ |
| $b_{i,l}$ | لحظه شروع یک کار در l -امین توالی ماشین M_i |

مدل سازی ریاضی مساله

بر اساس پارامترها و متغیرهای تصمیم تشریح شده، مدل ریاضی مسأله به شرح زیر می باشد. این مدل، نشان از یک برنامه ریزی خطی عدد صحیح باینری دارد:

$$\min z = \sum_{j=1}^n w_j T_j + \sum_{i=1}^m \sum_{j=1}^n \sum_{l=1}^n \sum_{s=1}^{s_i} x_{jils} t_{jis} e_{is} \quad (1)$$

S.t

$$\sum_{i=1}^m \sum_{l=1}^n \sum_{s=1}^{s_i} x_{jils} = 1 \quad \forall j \quad (2)$$

$$\sum_{j=1}^n \sum_{s=1}^{s_i} x_{jils} \leq 1 \quad \forall i, l \quad (3)$$

$$b_{i,l+1} - b_{i,l} = \sum_{j=1}^n \sum_{s=1}^{s_i} x_{jils} t_{jis} \quad \forall i, 1 \leq l \leq n-1 \quad (4)$$

$$T_j \geq b_{i,l} + x_{jils} t_{jis} - (1 - x_{jils}) M - d_j \quad \forall i, j, l, s \quad (5)$$

$$T_j \geq 0 \quad \forall j \quad (6)$$

$$x_{jils} \in \{0, 1\} \quad \forall i, j, l, s \quad (7)$$

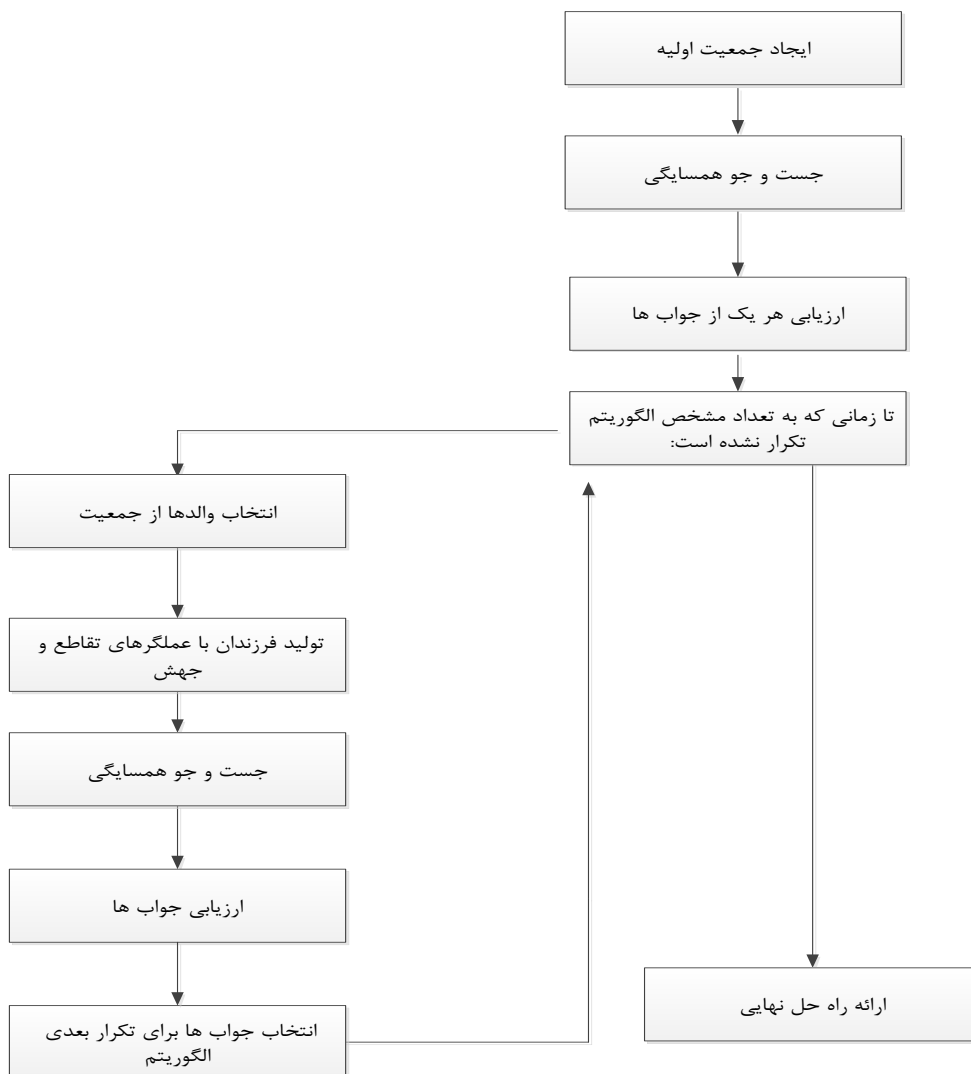
معادله (۱) تابع هدف مسأله را نشان می دهد که مجموع هزینه های جریمه دیرکرد و انرژی مصرفی را حداقل می کند. محدودیت (۲) بیان می کند هر کار باید به یک توالی از یک ماشین تخصیص یابد و تحت سرعت مشخص پردازش شود. محدودیت (۳) نشان دهنده این موضوع است که هر توالی از هر یک از ماشین ها حداکثر می تواند به یک کار اختصاص پیدا کند. محدودیت (۴) زمان شروع کار را در هر توالی از ماشین ها مشخص می کند و زمان دیرکرد کارها طبق محدودیت های (۵) و (۶) مشخص می شود. محدودیت (۷) هم طبق تعریف متغیر، بیان می کند برنامه ریزی مورد نظر، یک برنامه ریزی عدد صحیح باینری می باشد.

الگوریتم پیشنهادی

در این بخش ابتدا به تشریح الگوریتم ها و سپس به ارائه یک مثال عددی جهت بررسی و مقایسه الگوریتم ها پرداخته می شود.

الگوریتم فراابتکاری ممتیک

الگوریتم ممتیک گونه‌ای از الگوریتم‌های تکاملی است که در آن روش‌های مختلف جستجوی همسایگی با الگوریتم ژنتیک ترکیب می‌شوند تا بتوان به نتایج بهتری دست یافت. معمولاً الگوریتم‌های ژنتیکی، به دلیل بهره‌مندی از ابزارهای جست‌وجو موازی، نواحی از فضای حل که بهینه‌های محلی و سراسری در آن‌ها واقع شده است را به خوبی شناسایی می‌کنند، ولی در ادامه مسیرشان به سمت بهینه سراسری، بسیار کند عمل می‌کنند (اونگ و همکاران، ۲۰۰۶).



شکل ۱: فلوجارت الگوریتم ممتیک

از بین راهکارهای متعددی که برای برطرف ساختن مشکلات الگوریتم ژنتیک ارائه شده است استراتژی ترکیب^۱، جایگاه ویژه‌ای را به خود اختصاص داده است. استراتژی ترکیب

1. Hybrid

از به کارگیری تکنیک‌های مختلف در فرایند حل مسائل حاصل می‌گردد. الگوریتم‌های ممیتیک از مشهورترین اعضای این خانواده به شمار می‌آیند که از ترکیب الگوریتم‌های ژنتیکی با روش‌های ابتکاری جست‌وجو محلی حاصل می‌گردند. برای این منظور به ازای هر جواب که تولید می‌شود، یک جست‌وجو محلی با شعاع همسایگی از پیش تعیین شده‌ای حول آن جواب در فضای حل مسأله انجام می‌پذیرد. در الگوریتم ممیتیک به هر کروموزوم^۱ (جواب) شانس زندگی می‌دهیم و اعتقاد داریم که هر کروموزوم می‌تواند برازندگی خود را به اندازه بهترین همسایه‌اش بهبود ببخشد؛ یعنی هر جوابی که تولید شد، با استفاده از روش‌های جست‌وجو محلی، اطراف خود را جست‌وجو می‌کند و سعی می‌کند شایستگی خود را افزایش دهد. بنابراین گرچه الگوریتم ممیتیک، الگوریتم ژنتیک را پیچیده‌تر می‌کند، اما نسبت به الگوریتم ژنتیک، با استفاده از روش‌های جست‌وجو همسایگی خود، معمولاً جواب نهایی را در تعداد تکرارهای کمتری به دست می‌آورد و معمولاً به جواب‌های با کیفیت‌تری همگرا می‌شود. مطابق با توضیحات ارائه شده، ساختار کلی الگوریتم ممیتیک به صورت شکل ۱ خواهد بود.

مثال عددی

به منظور حل مدل و بررسی الگوریتم‌های پیشنهادی، چندین مثال عددی از مسأله را در نظر می‌گیریم (حل نموده‌های مختلف از مسأله) و مثال (۱) را در این قسمت به تفصیل، بررسی می‌کنیم. در این مثال مسأله زمانبندی ماشین‌های موازی ناهمگن، با در نظر گرفتن ۲۰ کار و ۳ ماشین ناهمگن با اطلاعات زیر (جدول (۱)) ارائه شده است:

1. Chromosome

جدول ۱: پارامترهای مثال (۱)

| کار | کار ۱ | کار ۲ | کار ۳ | کار ۴ | کار ۵ | کار ۶ | کار ۷ | کار ۸ | کار ۹ | کار ۱۰ | |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|
| ۴۵ | ۴۰ | ۶۲ | ۹ | ۵۱ | ۵۷ | ۱۹ | ۳۹ | ۷۳ | ۶۳ | ۹ | dj |
| ۷ | ۹ | ۱۰ | ۵ | ۹ | ۶ | ۱۳ | ۱۰ | ۳ | ۱۱ | ۷ | pj |
| ۱۵۰ | ۱۴۰ | ۱۱۰ | ۱۲۰ | ۱۱۰ | ۸۰ | ۹۰ | ۸۵ | ۱۳۰ | ۹۵ | ۱۵۰ | w j |

| کار | کار ۱ | کار ۲ | کار ۳ | کار ۴ | کار ۵ | کار ۶ | کار ۷ | کار ۸ | کار ۹ | کار ۱۰ | |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|
| ۵۱ | ۱۸ | ۴۲ | ۵۵ | ۳۱ | ۶۱ | ۴۶ | ۱۷ | ۴۹ | ۳۱ | ۱۹ | dj |
| ۹ | ۴ | ۸ | ۹ | ۱۱ | ۴ | ۱۲ | ۱۰ | ۵ | ۶ | ۹ | pj |
| ۱۱۵ | ۱۳۰ | ۷۵ | ۸۰ | ۱۰۰ | ۱۰۵ | ۱۳۵ | ۱۲۰ | ۸۰ | ۷۵ | ۲۰ | w j |

در این مثال، در مجموع ۳ سرعت برای پردازش کارها داریم:

(الف) سرعت سریع (حالت اول $s=1$)

(ب) سرعت نرمال (حالت دوم $s=2$)

(ج) سرعت کند (حالت سوم $s=3$)

اگر ماشینی روی سرعت سریع تنظیم شود، سرعت پردازش آن دو برابر سرعت نرمال می شود و اگر روی سرعت کند تنظیم شود، سرعت پردازش آن نصف سرعت نرمال می شود. ماشین M_1 را می توان روی هر ۳ سرعت برای پردازش هر یک از کارها، تنظیم کرد. ماشین M_2 را می توان روی سرعت های سریع و نرمال تنظیم کرد و ماشین M_3 را می توان روی سرعت های کند و نرمال جهت پردازش هر کار، تنظیم کرد. در ضمن می دانیم طبق جدول (۱) مدت زمان لازم برای پردازش هر کار زمانی که یک ماشین با سرعت نرمال به پردازش آن کار پردازد، مشخص است. جدول (۲) انرژی مصرفی هر یک از ماشین ها را در واحد زمان تحت سرعت های مختلف نشان می دهد (مقادیر e_{is}):

جدول ۲: مقادیر e_{is} در مثال (۱)

| <i>es-fast</i> | <i>es-normal</i> | <i>es-slow</i> | |
|----------------|------------------|----------------|---------|
| ۲۳۲ | ۱۱۳ | ۵۴ | ماشین ۱ |
| ۲۳۰ | ۱۱۴ | | ماشین ۲ |
| | ۱۱۳ | ۵۱ | ماشین ۳ |

نمایش جواب

به منظور نمایش هر جواب مسأله، از یک ماتریس با تعداد سطر برابر با تعداد کل کارها و سه ستون استفاده می‌کنیم؛ پس به منظور نمایش هر جواب مثال (۱)، از یک ماتریس با بیست سطر (تعداد کارها) و سه ستون استفاده می‌کنیم.

هر سطر ماتریس، مربوط به یک کار می‌باشد به ترتیبی که سطر اول مربوط به کار ۱، سطر دوم مربوط به کار ۲ و به همین ترتیب. ستون اول ماتریس به منظور تخصیص کارها به ماشین‌ها، ستون دوم مربوط به تعیین توالی کارهای تخصیص یافته به ماشین‌ها و ستون سوم مربوط به سرعت پردازش هر کار می‌باشد. کل درایه‌های ماتریس را اعدادی بین ۰ تا ۱۰۰ تشکیل می‌دهند. در مثال طرح شده، چنانچه عددی در ستون اول کوچکتر یا مساوی ۳۳ بود، کار مربوط به آن (سطر مربوط به آن عدد) به ماشین اول تخصیص می‌یابد (مثلا اگر عدد موجود در سطر ۱ و ستون اول برابر ۲۲ بود، کار ۱ به ماشین اول تخصیص می‌یابد) و اگر عدد بین ۳۳ و ۶۶ بود کار مربوط به آن عدد به ماشین دوم تخصیص می‌یابد و در غیر این صورت، کار مربوط به آن عدد به ماشین سوم تخصیص می‌یابد. بعد از تخصیص کارها، ستون دوم توالی کارهای تخصیص یافته به هر یک از ماشین‌ها را مشخص می‌کند؛ به نحوی که از بین کارهای تخصیص یافته به هر ماشین، هرچه عدد ستون دوم مربوط به کاری کوچکتر باشد، توالی آن کار جلوتر (زودتر) است. (مثلا اگر کار ۱ و کار ۱۳، به ماشین ۱ تخصیص یافته بودند و عدد موجود در سطر ۱ و ستون دوم ماتریس جواب، از عدد موجود در سطر ۱۳ و ستون دوم ماتریس جواب کوچکتر بود، یعنی توالی کار ۱، زودتر از کار ۱۳ است). ستون سوم مشخص کننده سرعت پردازش هر کار می‌باشد؛ به نحوی که در مثال طرح شده با مشخص بودن

کارهای تخصیص یافته به ماشین‌ها، اگر کاری طبق ستون اول ماتریس جواب، به ماشین اول تخصیص یافته باشد و عدد مربوط به ستون سوم آن کار (سطر)، کوچکتر از ۳۳ باشد، سرعت پردازش آن کار روی حالت کند تنظیم می‌شود و اگر عدد بین ۳۳ و ۶۶ باشد، سرعت پردازش آن کار روی حالت نرمال تنظیم می‌شود و در غیر این صورت، سرعت پردازش آن کار روی حالت سریع تنظیم می‌شود. اگر کاری طبق ستون اول ماتریس جواب به ماشین ۲ تخصیص یافته باشد و عدد مربوط به ستون سوم آن کار کوچکتر از ۵۰ باشد، سرعت پردازش آن کار روی نرمال و در غیر این صورت روی تند تنظیم می‌شود و در نهایت، اگر کاری طبق ستون اول ماتریس جواب به ماشین ۳ تخصیص یافته باشد و عدد مربوط به ستون سوم آن کار کوچکتر از ۵۰ باشد، سرعت پردازش آن کار روی کند و در غیر این صورت روی نرمال تنظیم می‌شود.

$$j_1 \begin{bmatrix} 78.22 & 11.63 & 24.06 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ j_{20} \end{bmatrix}_{20 \times 3}$$

شکل ۲: نمایش یک جواب از مسأله در مثال (۱)

جمعیت اولیه و ثبات جمعیت

جمعیت اولیه برابر ۵۰ در نظر گرفته می‌شود و تولید آن کاملاً تصادفی می‌باشد (تولید ۵۰ جواب اولیه تصادفی).

اندازه جمعیت در طول تکرار الگوریتم، ثابت در نظر گرفته می‌شود.

اپراتورها

تابع انتقال دقیق کروموزوم‌ها به نسل بعد^۱؛ در هر تکرار ۲۰ درصد جمعیت (معادل ۱۰ کروموزوم) مستقیماً به نسل بعدی منتقل شوند که معیار انتخاب^۲، چرخه رولت^۱ می‌باشد.

1. reproduction
2. selection

تقاطع؛ به منظور افزایش تنوع دهی، انتخاب والدین به صورت تصادفی از جمعیت نسل گذشته می‌باشد و تابع تقاطع به صورت تک نقطه^۲ عمل می‌کند. در هر تکرار ۷۶ درصد جمعیت، توسط تابع تقاطع تولید می‌شوند و معیار انتخاب جواب برای انتقال به نسل بعد از بین والدین و فرزندان، چرخه رولت می‌باشد.



شکل (۳) عملکرد عملگر تقاطع

جهش؛ به ازای هر جواب از یک نسل، یک جواب جدید توسط تابع جهش تولید می‌شود به گونه‌ای که دو سطر از هر جواب را به صورت تصادفی انتخاب کرده و آنها را با اعداد تصادفی دیگری جایگزین می‌کنیم. در نهایت از بین کل فرزندان تولید شده به وسیله جهش، دو جوابی که برازنده‌تر می‌باشند را به نسل بعد منتقل می‌کنیم (معادل ۴ درصد جمعیت در هر تکرار).

1 .roulette wheel
2 .one point

| والد | فرزند تولید شده از عملگر جهش |
|----------|------------------------------|
| 10 96 65 | 10 96 65 |
| 12 87 63 | 33 32 21 |
| 74 11 18 | 74 11 18 |
| 93 10 86 | 93 10 86 |
| 32 47 63 | 91 23 3 |

شکل (۴) عملکرد عملگر تقاطع

جست‌وجو همسایگی؛ شعاع همسایگی برابر با یک در نظر گرفته می‌شود؛ یعنی برای هر جوابی که بعد از تقاطع و جهش برای حضور در نسل بعد انتخاب شد، n همسایه در نظر گرفته می‌شود (n برابر با تعداد کارها می‌باشد) که هر جواب همسایه فقط در یک سطر با جواب فعلی تفاوت دارد. به ازای هر جواب، هر بار یک سطر آن جواب را با یک سطر تصادفی عوض می‌کنیم و این کار را برای هر یک از n سطر جواب، تکرار می‌کنیم تا n همسایه برای هر جواب به وجود آید. سپس از بین همسایه‌ها، جواب همسایه‌ای که بهترین برازندگی را دارد، جایگزین جواب فعلی می‌کنیم. در مثال (۱)، طبق روش جست‌وجو همسایگی تعریف شده، به ازای هر جواب، ۲۰ جواب همسایه به وجود می‌آید و از بین جواب‌های همسایه، جواب همسایه‌ای که بهترین برازندگی را دارد، جایگزین جواب فعلی می‌کنیم.

معیار اختتام الگوریتم

معیار اختتام الگوریتم در تمامی مثال‌ها را برابر با ۵۰۰ تکرار الگوریتم در نظر می‌گیریم.

الگوریتم ژنتیک پیشنهادی

الگوریتم ژنتیک پیشنهادی نیز همانند الگوریتم ممیتیک پیشنهاد شده می‌باشد که از آن جست-وجوهای همسایگی حذف گردیده است.

نتایج حل مثال عددی

پس از حل مثال (۱) به وسیله الگوریتم ممیتیک، در تکرار آخر جواب زیر حاصل شد:

| | | | | | | | | | | | | | | | | | | | |
|----|-----------|-----------|-----------|-----------|----------|----------|-----------|-----------|-----------|-----------|--|--|--|--|--|--|--|--|--|
| M3 | Job10 s=3 | Job12 s=3 | Job18 s=3 | Job9 s=3 | Job8 s=3 | | | | | | | | | | | | | | |
| M2 | Job3 s=1 | Job17 s=1 | Job11 s=1 | Job6 s=1 | Job4 s=1 | Job1 s=1 | Job19 s=2 | Job16 s=1 | Job20 s=2 | Job13 s=2 | | | | | | | | | |
| M1 | Job5 s=2 | Job14 s=3 | Job7 s=2 | Job15 s=2 | Job2 s=3 | | | | | | | | | | | | | | |

Fitness=17862

شکل (۵) زمانبندی مثال (۱) با الگوریتم ممیتیک

برازندگی این جواب، برابر ۱۷۸۶۲ می‌باشد. پس از حل مثال (۱) به وسیله الگوریتم ژنتیک، در تکرار آخر جواب زیر حاصل شد:

| | | | | | | | | | | | | | | | | | | | |
|----|-----------|-----------|-----------|-----------|----------|-----------|-----------|----------|-----------|-----------|----------|--|--|--|--|--|--|--|--|
| M3 | Job14 s=3 | Job1 s=3 | Job9 s=3 | Job8 s=3 | | | | | | | | | | | | | | | |
| M2 | Job11 s=2 | Job10 s=1 | Job19 s=1 | Job6 s=1 | Job7 s=2 | Job15 s=2 | Job12 s=1 | Job4 s=1 | Job18 s=1 | Job20 s=2 | Job2 s=1 | | | | | | | | |
| M1 | Job3 s=2 | Job17 s=2 | Job13 s=3 | Job16 s=2 | Job5 s=3 | | | | | | | | | | | | | | |

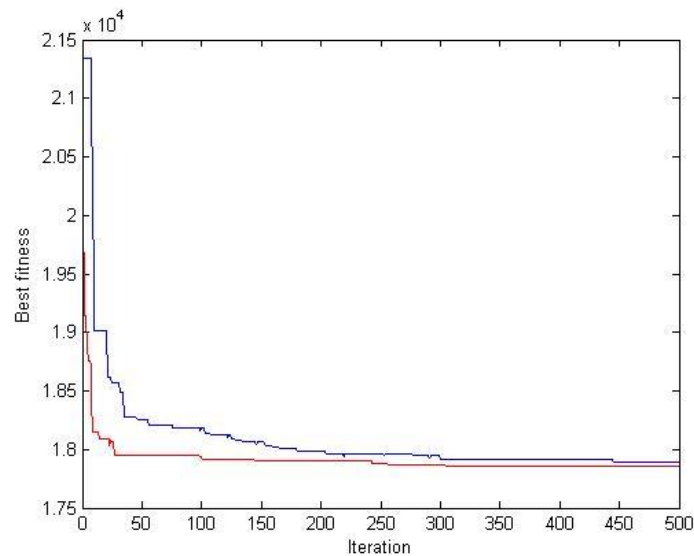
Fitness=17887

شکل (۶) زمانبندی مثال (۱) با الگوریتم ژنتیک

برازندگی این جواب، برابر ۱۷۸۸۷ می‌باشد. روند همگرایی جواب به سمت راه‌حل برتر در حل مسأله به وسیله هر یک از الگوریتم‌های پیشنهادی و مقایسه آن‌ها در بخش ۴ آمده است.

ارزیابی و مقایسه نتایج

شکل زیر، مقدار بهترین برازندگی جواب‌ها را در هر تکرار از الگوریتم‌های ممیتیک و ژنتیک در مثال (۱) در ۵۰۰ تکرار را نشان می‌دهند. خط قرمز مربوط به الگوریتم ممیتیک و خط آبی مربوط به الگوریتم ژنتیک می‌باشد.



شکل (۷) نمودار برازندگی بهترین جواب در هر تکرار الگوریتم‌ها (ممیتیک --- و ژنتیک ---)

همانگونه که انتظار داشتیم الگوریتم ممیتیک با استفاده از جست‌وجوهای همسایگی خود، در این مسأله توانست در تعداد تکرار کمتری به جواب قابل قبولی همگرا شود و در نهایت بعد از ۵۰۰ تکرار، جواب با کیفیت بالاتری را نسبت به الگوریتم ژنتیک در اختیار ما قرار می‌دهد. باید به این موضوع توجه کرد که هر تکرار الگوریتم ممیتیک، به زمان بیشتری نسبت به هر تکرار الگوریتم ژنتیک نیاز دارد؛ به طوری که ۵۰۰ تکرار الگوریتم ممیتیک برای حل مثال (۱)، ۷۹۷,۵۲ ثانیه زمان صرف کرد، در حالی که ۵۰۰ تکرار الگوریتم ژنتیک، ۱۸۸,۴۲ ثانیه زمان صرف کرد. بنابراین در هر مسأله با توجه به ابعاد و ویژگی‌های مسأله سعی می‌کنیم از الگوریتم‌های فراابتکاری استفاده کنیم. چنانچه در مثال حل شده در پژوهش فعلی دریافتیم که

معمولا الگوریتم ممیتیک در تعداد کمتری تکرار، نسبت به الگوریتم ژنتیک به سمت جوابی قابل قبول همگرا می‌شود و معمولا در نهایت جواب با کیفیت‌تری را در تعداد تکرار برابر نسبت به الگوریتم ژنتیک، در اختیار ما قرار می‌دهد؛ اما زمان لازم برای هر تکرار در الگوریتم ممیتیک بیشتر از زمان لازم برای هر تکرار الگوریتم ژنتیک می‌باشد که بستگی به ویژگی‌ها، دانش و ابعاد هر مسأله، باید تصمیم‌گیری کنیم که آیا از روش‌های مختلف جست‌وجو محلی در الگوریتم خود استفاده کنیم یا نه! اگر جواب مثبت است، چگونه و به چه میزان از این روش‌ها استفاده کنیم.

الگوریتم‌های ممیتیک و ژنتیک پیشنهاد شده برای حل مسأله در ابعاد و پارامترهای مختلف به کار رفته است. در ضمن به منظور اثبات صحت مدل و الگوریتم‌های فراابتکاری پیشنهادی، نتایج حاصل از حل کلیه مثال‌ها در نرم‌افزار بهینه‌سازی گمز، در ستون آخر جدول (۳) آورده شده است که نتایج حل ۱۳ نمود از مسأله، به شرح جدول (۳) می‌باشد (نمودهای ۱، ۳، ۴ و ۹ نمودهای ساینز کوچک و نمودهای ۲، ۷، ۱۰ و ۱۱ نمودهای ساینز متوسط و نمودهای ۵، ۶، ۸، ۱۲ و ۱۳ نمودهای ساینز بزرگ مسأله می‌باشند).

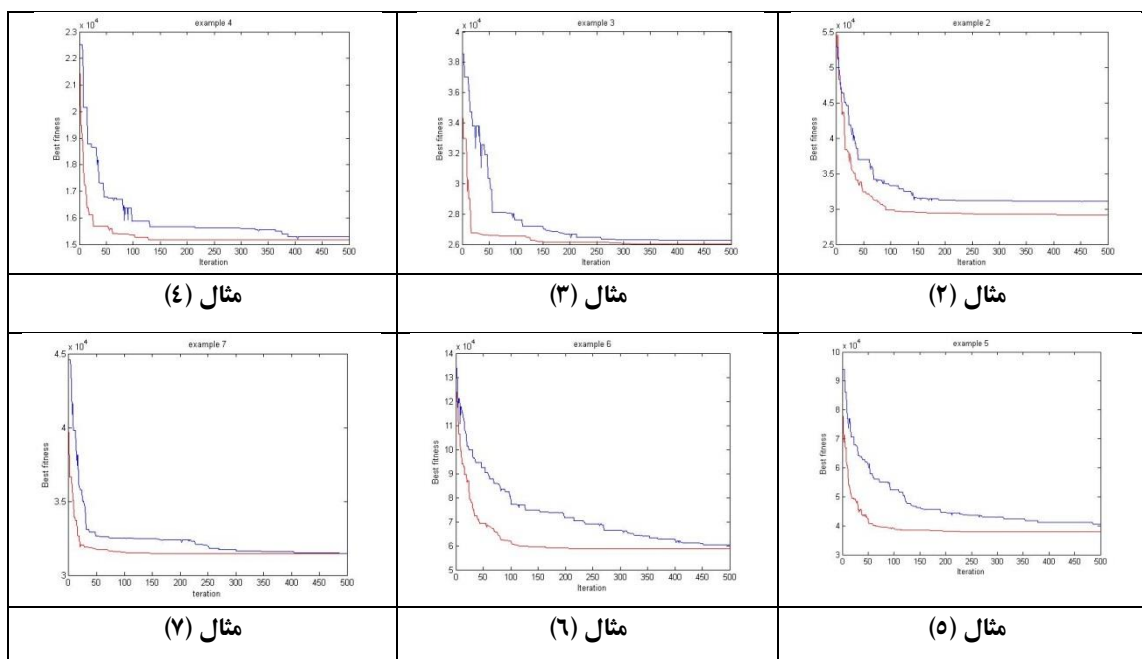
طبق جدول (۳)، مشاهده می‌شود که در تمامی مثال‌ها، الگوریتم ممیتیک پس از ۵۰۰ تکرار، به جوابی با برازندگی مطلوب‌تر نسبت به الگوریتم ژنتیک، همگرا شده است و طبق نمودارهای موجود در شکل‌های (۵) تا (۱۶) که عملکرد الگوریتم‌های پیشنهاد شده را برای حل مثال‌ها مقایسه می‌کند، مشاهده می‌شود که در هر یک از مثال‌ها، الگوریتم ممیتیک در تعداد کمتری تکرار نسبت به الگوریتم ژنتیک، به یک جواب خوب همگرا شده است؛ اما زمان لازم برای ۵۰۰ تکرار الگوریتم ممیتیک، بسیار بیشتر از زمان لازم برای ۵۰۰ تکرار الگوریتم ژنتیک (در هر مثال)، می‌باشد. در ضمن مشاهده می‌شود که برازندگی جواب نهایی حاصل از الگوریتم‌های پیشنهادی در هر یک از مثال‌ها، فاصله نزدیکی با برازندگی جواب بهینه و حد پایین مسأله مورد نظر که از خروجی نرم‌افزار گمز بدست آمده است، دارند و این بدان معناست که الگوریتم‌های فراابتکاری پیشنهادی برای این مسأله، به خوبی عمل می‌کنند و صحت مدل ایجاد شده برای مسأله و الگوریتم‌های فراابتکاری، اثبات می‌شود. ضمنا با توجه به

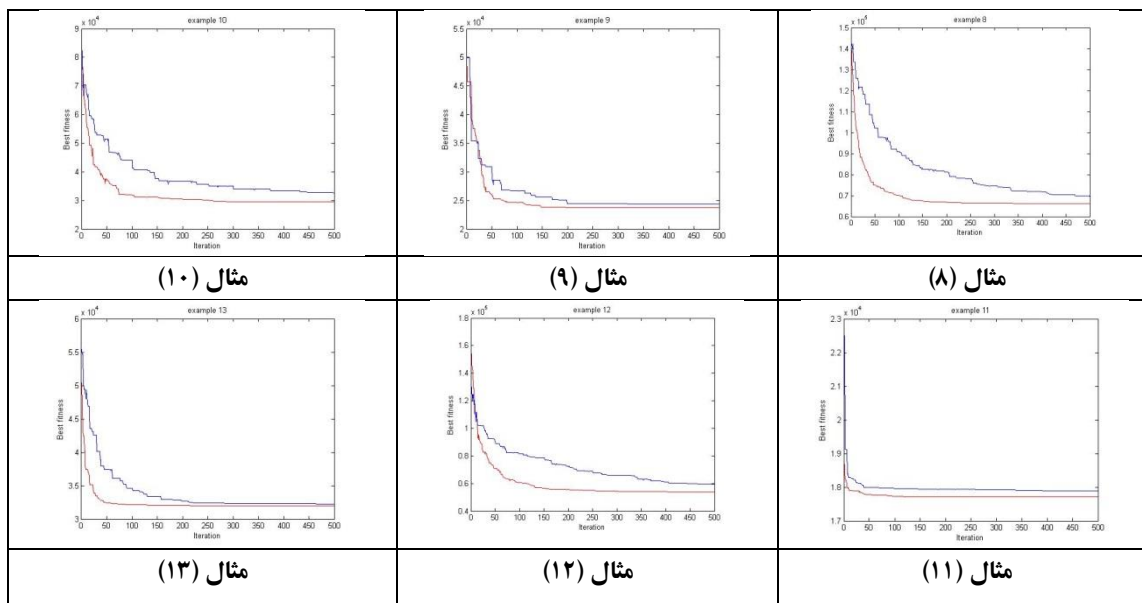
پیچیدگی مساله پژوهش حاضر، امکان رسیدن به جواب بهینه در نمودهای سائز بزرگ مساله در زمان معقول وجود ندارد.

جدول (۳) نتایج عددی مقایسات

| برازندگی جواب بهینه (خروجی گمز) | | زمان لازم برای ۵۰۰ تکرار الگوریتم | | برازندگی جواب نهایی | | تعداد ماشینها | تعداد کارها | مثالها |
|---------------------------------|---------------------|-----------------------------------|---------|---------------------|-------|---------------|-------------|---------|
| حد پایین مساله | برازندگی راه حل گمز | ژنتیک | ممتیک | ژنتیک | ممتیک | (m) | (n) | |
| ۱۷۲۴۲ | ۱۷۲۴۲ | ۱۸۸.۴۲ | ۷۹۷.۵۲ | ۱۷۸۸۷ | ۱۷۸۶۲ | ۳ | ۲۰ | مثال ۱ |
| ۲۸۴۰۱ | ۳۰۱۰۳ | ۲۶۹.۰۸ | ۱۵۲۶.۳۷ | ۳۱۰۵۶ | ۲۹۱۷۴ | ۳ | ۳۰ | مثال ۲ |
| ۲۵۰۰۱ | ۲۵۰۰۱ | ۱۸۷ | ۷۳۰.۳۵ | ۲۶۲۷۴ | ۲۶۰۱۹ | ۳ | ۱۸ | مثال ۳ |
| ۱۴۲۸۹ | ۱۴۲۸۹ | ۱۷۸.۶۳ | ۶۸۶.۲۳ | ۱۵۲۸۴ | ۱۵۱۷۶ | ۳ | ۱۵ | مثال ۴ |
| نامعلوم | نامعلوم | ۳۲۰.۵۱ | ۲۲۹۲.۹ | ۴۰۶۲۶ | ۳۷۸۲۰ | ۳ | ۴۰ | مثال ۵ |
| نامعلوم | نامعلوم | ۲۹۳.۲۲ | ۲۱۶۶.۶ | ۶۰۲۹۹ | ۵۸۷۱۶ | ۵ | ۴۲ | مثال ۶ |
| ۲۹۷۶۵ | ۲۹۹۱۴ | ۲۲۴.۹۶ | ۱۰۹۴.۱۷ | ۳۱۴۷۲ | ۳۱۴۷۵ | ۵ | ۲۴ | مثال ۷ |
| نامعلوم | نامعلوم | ۳۸۲.۳ | ۳۸۹۶.۲۲ | ۶۹۶۴۳ | ۶۶۱۳۳ | ۵ | ۶۴ | مثال ۸ |
| ۲۲۲۷۶ | ۲۲۲۷۶ | ۸۳.۲۹ | ۵۱۷.۲۶ | ۲۴۳۰۸ | ۲۳۶۹۴ | ۲ | ۱۸ | مثال ۹ |
| ۲۶۷۱۲ | ۲۷۵۴۰ | ۱۷۲.۵۲ | ۹۰۱.۶۵ | ۳۲۴۹۵ | ۲۹۴۳۶ | ۲ | ۲۸ | مثال ۱۰ |
| ۱۷۲۸۰ | ۱۷۴۲۱ | ۱۸۹.۴۱ | ۸۶۸.۴۵ | ۱۷۸۸۱ | ۱۷۷۰۴ | ۴ | ۲۰ | مثال ۱۱ |
| نامعلوم | نامعلوم | ۳۱۵.۸۱ | ۲۷۷۸.۰۸ | ۵۹۴۰۸ | ۵۳۷۴۸ | ۴ | ۵۴ | مثال ۱۲ |
| نامعلوم | نامعلوم | ۲۲۸.۹ | ۱۴۰۶.۴۲ | ۳۲۲۱۲ | ۳۱۹۴۸ | ۴ | ۳۳ | مثال ۱۳ |

معمولا در حل مسائل مختلف اگر معیار ما برای انتخاب الگوریتم مناسب، تعداد تکرار خاصی باشد، در تعداد تکرار برابر، کارایی الگوریتم ممیتیک نسبت به الگوریتم ژنتیک، بالاتر می‌باشد؛ چرا که معمولا الگوریتم ممیتیک در تعداد تکرار برابر، به جواب برازنده‌تری نسبت به الگوریتم ژنتیک، همگرا می‌شود. چنانچه معیار ما زمان در دسترس برای رسیدن به یک جواب خوب باشد، با توجه به اینکه زمان لازم برای هر تکرار الگوریتم ممیتیک از زمان لازم برای هر تکرار الگوریتم ژنتیک، بیشتر است و توجه به این موضوع که الگوریتم ممیتیک ممکن است در تعداد تکرار کمتر به جواب بهتری دست یابد (حتی در زمان برابر)، با توجه به ماهیت مسائل گوناگون و دانش موجود در هر مسأله، پیشینی می‌کنیم که کدام یک از الگوریتم‌ها، کارایی بیشتری دارد. نمودارهای زیر، عملکرد الگوریتم‌های ممیتیک و ژنتیک پیشنهاد شده را در هر یک از ۱۳ مثال حل شده، مقایسه می‌کنند:





شکل (۸) مقایسه عملکرد الگوریتم‌ها (ممتیک --- و ژنتیک ---)

نتیجه‌گیری و تحقیقات آتی

با توجه به محدودیت منابع انرژی و مسائل زیست محیطی، مصرف انرژی یکی از موارد مهم در برنامه‌ریزی سیستم‌های مختلف می‌باشد. از آنجا که در صنایع مختلف، معمولاً صرف انرژی بیشتر باعث تسریع در انجام کارها می‌گردد و به تبع هزینه‌های سیستم را افزایش می‌دهد، برنامه‌ریزی مناسب و بهینه‌سازی در حوزه مصرف انرژی می‌تواند به موفقیت یک سیستم کمک شایانی دهد. از این رو می‌توان با توجه به ماهیت مسائل، از روش‌های مختلف بهینه‌سازی به منظور بهینه‌سازی مصرف انرژی در کنار سایر اهداف سیستم، بهره برد؛ همانگونه که در مسأله موجود در این پژوهش که یک مسأله NP-hard بود، از الگوریتم فراابتکاری ممتیک جهت بهینه‌سازی انرژی مصرفی در کنار بهینه‌سازی مجموع جریمه‌های دیرکرد، برای بهینه‌سازی مدل خطی عدد صحیح باینری یک مسأله زمانبندی ماشین‌های موازی ناهمگن، استفاده شد و مشاهده شد که الگوریتم ممتیک در تعداد کمتری تکرار، نسبت به الگوریتم ژنتیک به سمت جوابی قابل قبول همگرا شد و در نهایت جواب با کیفیت‌تری را در

تعداد تکرار برابر نسبت به الگوریتم ژنتیک، در اختیار ما قرار داد؛ اما زمان لازم برای هر تکرار در الگوریتم ممتیک بیشتر از زمان لازم برای هر تکرار الگوریتم ژنتیک می‌باشد. در تحقیقات آتی سعی می‌شود مصرف انرژی را در سایر کارگاه‌های موجود در مسائل زمانبندی، دخیل کنیم و به بهینه‌سازی چندهدفه مسائل زمانبندی با در نظر گرفتن مصرف انرژی، پردازیم. در ضمن اضافه شدن مفروضاتی که مسأله ما را به مسائل دنیای واقعی نزدیک‌تر کند، مثل فرض امکان بریدگی در پردازش کارها، وجود پنجره زمانی در موعد تحویل کارها، امکان تغییر سرعت ماشین‌ها حین انجام هر کار، وجود زمان‌های آماده‌سازی وابسته به توالی و ... می‌تواند به جذابیت مسأله زمانبندی، اضافه کند.

منابع

فخرزاد، محمد باقر. رجائی، بهنام. (۱۳۹۶). نگهداری پیشگیرانه در زمانبندی ماشین‌های موازی نامرتب با احتساب اثر زوال و زمان آماده‌سازی. مجله مطالعات مدیریت صنعتی، دوره ۱۵، شماره ۱-۴۳، ۴۵.

Anghinofi, D., Paolucci, M. (2007). *Parallel machine total tardiness scheduling with a new hybrid metaheuristic approach*. Computers & Operations Research, 34(11), 3471-3490.

Balin, S. (2011). *Non-identical parallel machine scheduling using genetic algorithm*. Expert System with Applications, 38, 6814-6821.

Che, A., Zhang, S., Wu, X. (2017). *Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs*. Journal of cleaner production, 156, 688-697.

Despeisse, M., Ball, P.D., Evans, S., Levers, A. (2012). *Industrial ecology at factory level e a conceptual model*. Journal of cleaner production, 156, 688-697.

Ding, J.Y., Song, S., Zhang, R., Chiong, R. (2016). *Parallel machine scheduling under time-of-use electricity prices: new models and optimization approaches*. IEEE Transactions on Automation Science & Engineering, 13(2), 1138-1154.

Dror, M., Stern, H.I., and Lenstra, J.K. (1987), *Parallel machine scheduling: Processing rates dependent on number of jobs in operation*, Management Science, 33, 1001-1009.

Fang, K., Uhan, N., Zhao, F., Sutherland, J.W. (2011). *A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction*. Journal of Manufacturing Systems, 30, 234e240.

Fang, Lin, K-T., B.M.T. (2012). *Parallel-machine scheduling to minimize tardiness penalty and power cost*. Computers & Industrial Engineering, 64 (2013) 224-234.

Fanjul-Peyro, L., Perea, F., Ruiz, R. (2017). *Models and matheuristics for the unrelated parallel machine scheduling*

problem with additional resources. European Journal of Operational Research, 260(2), 482-493.

Garey, M.R., Johnson, D.S. (1978). "Strong" NP-completeness results: Motivation, examples, and implications. Journal of the Association for Computing Machinery, 25(3), 499-508.

Hoogeveen, H. (2005). Multicriteria scheduling. European Journal of Operational Research, 167(3), 592-623

Husseinzadeh Kashan, A. Karimi, B. (2008). A discrete particle swarm optimization algorithm for scheduling parallel machines. Computers & Industrial Engineering, 56 (2009) 216-223.

Kim, D-W., Kim, K-H., Jang, W., Chen, F. (2002). Unrelated parallel machine scheduling with setup times using simulated annealing. Robotics and Computer Integrated Manufacturing, 18, 223-231.

Lenstra, J. K., Rinnooy Kan, A. H. G., Brucker, P. (1977). Complexity of machine scheduling problems. Annals of Discrete Mathematics, 1, 343-362.

Lin, C.-H., Liao, C.-J. (2008). Makespan minimization for multiple uniform machines. Computers & Industrial Engineering, 54(4), 983-992.

Lin, Y.K., Pfund, M.E., Fowler, J.W. (2011). Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problem. Computers & Operations Research, 38(6), 901-916.

Liu, Y., Dong, H., Lohse, N., Petrovic, S., Gindy, N. (2013). An investigation into minimizing total energy consumption and total weighted tardiness in job shops. Journal of cleaner production, 65, 87-96.

Moon, J.Y., Shin, K., Park, J. (2013). Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency. International Journal of Advanced Manufacturing Technology, 68, 523-535.

Mouzon, G., Yildirim, M.B., Twomey, J. (2007). *Operational methods for minimization of energy consumption of manufacturing equipment*. International Journal of Production Research, 45, 4247e-4271.

Muramatsu, R., Ishii, K., Takahashi, K. (1985). *Some ways to increase flexibility in manufacturing systems*", International Journal of Production Research, 23, 691-703.

Ong, Y.S, Lim, M.H., Zhu, N., Wong, K.W. (2006). *Classification of adaptive memetic algorithms: a comparative study*. IEEE Transactions on Systems, Man and Cybernetics - Part B, 36(1), 141-152.

Rodriguez, F. J., Lozano, M., Blum, C., Garcia-Martinez, C. (2013). *An iterated greedy algorithm for the large-scale unrelated parallel machine scheduling problem*. Computers & Operations Research, 40, 1829-1841.

Rogendran, R., Subur, F. (2004). *Unrelated parallel machine scheduling with job splitting*. IIE Transaction, 36, 356-372.

Sharma, A., Zhao, F., Sutherland, J.W. (2015). *Econological scheduling of a manufacturing enterprise operating under a time-of-use electricity tariff*. Journal of Cleaner Production, 108, 256-270.

Shrouf, F., Ordieres-Meré, J., García-Sánchez, A., Ortega-Mier, M. (2014). *Optimizing the production scheduling of a single machine to minimize total energy consumption costs*. Journal of Cleaner Production, 67, 197-207.

Smith, J.E. (2007). *Co-evolving memetic algorithms: A review and progress report*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 37 (1), 6-17.

Tavakkoli-Moghaddam, R., Taheri, F., Bazzazi, M., Izadi, M., Sassani, F. (2009). *Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints*. Computers & Operations Research, 36, 3224-3230.

Vallada, E., Ruiz, R. (2011). *A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times*. European Journal of Operational Research, 211, 612–622.

Vredevelde, T., Hurkens, C. (2002). *Experimental comparison of approximation algorithms for scheduling unrelated parallel machines*. Informs Journal on Computing, 14(2), 175–189.

Zhang, H., Zhao, F., Fang, K., Sutherland, J.W. (2014). *Energy-conscious flow shop scheduling under time-of-use electricity tariffs*. CIRP Annals – Manufacturing Technology, 63, 37–40.