مجله تحقیق در عملیات در کاربردهای آن، سال نوزدهم، شماره چهارم (پیاپی ۷۵)،زمستان ۰۱، ص ص ۳۶-۱۷
شاپا ۷۲۸۶-۲۲۵۱
(مروری)

طرح تقریب دیفرانسیلی برای مساله زمانبندی روی تک ماشین با محدودیت دسترسی

مریم سیف‌الدینی[1*]
۱- استادیار، دانشگاه گیلان، گروه علوم کامپیوتر، گیلان، ایران

رسید مقاله: ۱۰ دی ۱۴۰۰
پذیرش مقاله: ۱۶ خرداد ۱۴۰۱

چکیده

مسایل زمانبندی همواره مورد توجه گسترده محققان حوزه‌های مختلف بوده است. از آنجایی که اغلب این مسایل و تقریبا همه مسایل دنیای واقعی در ردۀ مسایل NP- سخت بهینه‌سازی ترکیبیاتی و علوم کامپیوتر قرار می‌گیرند، لذا پیدا کردن راه حل مناسب، راه حلی که در زمانی معقول (زمان چندجمله‌ای) قابل اجرا باشد، دشوار است. یکی از راهکارهای مطرح‌شده برای حل این مشکل، به‌کارگیری راهکار تقریب است. طرح‌های تقریب با زمان چندجمله‌ای و طرح تقریب دیفرانسیلی که بر اساس مقایسه جواب الگوریتم مطرح‌شده برای حل مساله با جواب بهینه و جواب بدترین حالت بناشده است، در دسته روش‌های تقریب قرار می‌گیرند. کیفیت جواب الگوریتم ارایه‌شده برای یک مساله از طریق طرح تقریبی قابل ارزیابی است. این مقاله ضمن مرور طرح‌های تقریب کارا بر روی دو مساله زمانبندی کارها بر روی تک ماشین با اهداف مینیمم‌سازی ماکزیمم زمان تحویل کارها و مینیمم‌سازی مجموع وزن‌دار اتمام کارها، روش‌ها و ابزارهای لازم برای اثبات وجود یک طرح تقریبی را ارایه می‌دهد. همچنین در این مقاله با استفاده از طرح تقریب با زمان چندجمله‌ای (PTAS) یک طرح تقریب دیفرانسیلی برای مساله مینیمم‌سازی ماکزیمم زمان تحویل کارها که تا کنون مورد بررسی قرار نگرفته است، مورد مطالعه قرار گرفته است.

کلمات کلیدی: زمانبندی، طرح تقریب، الگوریتم ابتکاری، الگوریتم شبه‌چندجمله‌ای.

۱ مقدمه

مسایل زمانبندی به دلیل کاربرد زیاد در صنعت، تکنولوژی و حوزه‌های دیگر مورد توجه دانشمندان قرار دارند [۱-۴]. شروع تحقیقات در این حوزه در دهه پنجاه با انگیزه کاربردی‌بودن این مسایل در زمینه‌های علوم کامپیوتر، بهینه‌سازی ترکیبیاتی، تحقیق در عملیات و مدیریت بوده است. بسیاری از این مسایل در رده‌ی مسایل NP – سخت علوم کامپیوتر و بهینه‌سازی ترکیبیاتی قرار می‌گیرند. از این رو پیدا کردن جواب بهینه دقیق برای آنها در

* عهده‌دار مکاتبات
آدرس الکترونیکی: m.seifaddini@guilan.ac.ir

[DOI: 10.52547/jamlu.19.4.17]

زمان معقول (زمان چندجمله‌ای) امکان‌پذیر نیست. بنابراین محققان تلاش دارند با رویکردهای مناسب، روش‌های کارایی برای حل این مسایل پیدا کنند. یکی از این رویکردها ارایه راه‌حل‌هایی تقریبی و نزدیک به جواب بهینه به جای روش‌های دقیق است. این رویکردها به لحاظ زمان اجرا سریع‌ترند و در حین حال جواب‌های با کیفیتی را نیز ارایه می‌دهند.

روش‌های تقریبی در تمامی مسایل بهینه‌سازی، نظریه گراف و کلیه مسایل الگوریتم محور استفاده می‌شوند و به‌طور معمول به‌صورت الگوریتم‌های تقریب[۱] (APX)، طرح تقریب با زمان چندجمله‌ای[۲] (PTAS)، طرح تقریب با زمان کاملاً چندجمله‌ای[۳] (FPTAS) و طرح تقریب دیفرانسیلی با زمان کاملاً چندجمله‌ای[۴] (DFPTAS) مطرح می‌شوند. اساس این طرح‌ها بر مبنای مقایسه جواب الگوریتم ابتکاری[۵] مطرح شده برای حل مساله و جواب بهینه مساله است. ضمن این‌که در طرح تقریب دیفرانسیلی جواب الگوریتم ابتکاری مطرح‌شده با جواب بهینه و جواب بدترین حالت مقایسه می‌شود که نسبت به طرح‌های قبلی کاراتر است.

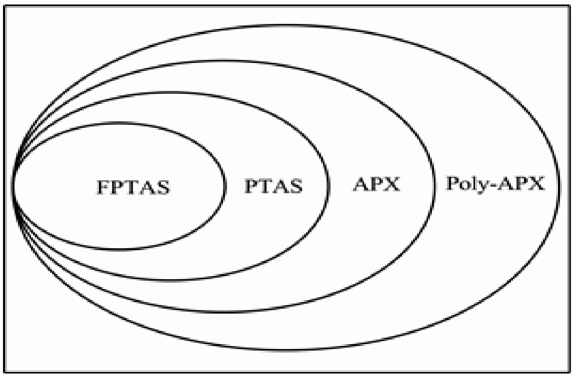
در واقع طرح تقریب دیفرانسیلی، محل قرارگیری جواب تقریبی را در بازه جواب بدترین حالت و جواب بهینه ارزیابی می‌کند. طرح تقریب دیفرانسیلی در بسیاری از مسایل بهینه‌سازی ترکیبیاتی استفاده شده است [۵]. دسته‌بندی این سه طرح مورد مطالعه در این پژوهش در شکل ۱ ارایه شده است.

مفهوم الگوریتم تقریب در شروع دهه ۱۹۷۰ توسط گری و همکاران [۶] ارایه شد. همچنین راهکار طرح‌های تقریب PTAS و FPTAS در سال ۱۹۷۸ توسط گری و جانسون [۷] ارایه شد. طرح تقریب دیفرانسیلی، DFPTAS، نیز اولین بار توسط دمانچ و همکاران [۸] ارایه شد که این طرح به دلیل این‌که فاصله جواب مساله را هم نسبت به جواب بهینه و هم نسبت به جواب بدترین حالت ارزیابی می‌کند، به نظر می‌رسد نسبت به طرح‌ها و معیارهای کلاسیک کارایی بیشتری داشته باشد. در ۱۹۸۰ دانشمندان مطالعات خود را در زمینه این طرح‌ها روی مسایل مختلف آغاز کردند و نتایج ارزشمندی را به دست آوردند.

این مقاله به مرور طرح‌های تقریب به دست آمده روی دو مساله زمانبندی خواهد پرداخت. همچنین به دلیل این‌که طرح تقریب دیفرانسیلی در مسایل زمانبندی کمتر مورد توجه قرار گرفته است این پژوهش بر این موضوع متمرکز می‌شود. وجه تمایز این مقاله با سایر پژوهش‌ها بررسی وجود طرح تقریب دیفرانسیلی مساله زمانبندی کارها بر روی تک ماشین با هدف مینیمم‌سازی ماکزیمم زمان تحویل کارها است که تا کنون مورد مطالعه قرار نگرفته است (بخش ۵).

[1] Approximation Algorithm
[2] Polynomial Time Approximation Scheme
[3] Fully Polynomial Time Approximation Scheme
[4] Differential Fully Polynomial Time Approximation Scheme
[5] Heuristic

[DOI: 10.52547/jamlu.19.4.17]



شکل ۱. دسته‌بندی طرح‌های تقریب

اهمیت درک نتایج حاصل از این طرح‌ها و چگونگی دستیابی به آنها در حل سایر مسایل زمانبندی و ارایه طرح‌های تقریب مناسب برای آنها موثر خواهد بود. براساس جستجوی نگارنده، تاکنون مقاله داخلی و خارجی به مروری جامع بر این موضوع نپرداخته است.

در بخش دوم تعاریف مربوطه ارایه می‌شود، بخش سوم این مقاله به چگونگی به‌دست آوردن طرح تقریب از یک طرف و چگونگی نشان دادن تقریب‌ناپذیری یک مساله از طرف دیگر می‌پردازد. در بخش چهارم خلاصه‌ای از نتایج به‌دست آمده بر روی دو مساله زمانبندی کارها روی یک تک ماشین با هدف مینیمم‌سازی ماکزیمم زمان تحویل[1] کارها و مینیمم‌سازی مجموع وزن‌دار زمان اتمام کارها ارایه خواهد شد. در ادامه، بخش ۵ به طرح تقریب دیفرانسیلی مساله زمانبندی کارها بر روی تک ماشین با هدف مینیمم‌سازی ماکزیمم زمان تحویل کارها تخصیص داده شده است. در پایان نیز جمع بندی و نتیجه‌گیری ارایه خواهد شد.

۲ تعاریف

به منظور ایجاد زبان مشترک و ارایه یک مبنا در ادامه به بیان چند تعریف در زمینه طرح‌های تقریب می‌پردازیم:

تعریف ۲-۱. الگوریتم A برای یک مساله مینیمم‌سازی با تابع هدف φ را یک الگوریتم $\rho-$ تقریب برای مساله π می نامیم هرگاه برای هر نمونه I از مساله رابطه $\frac{\varphi(A(I))}{\varphi(opt(I))} \le \rho$ برقرار باشد که در آن $opt(I)$ مقدار بهینه برای نمونه I است. ρ را کران بدترین حالت الگوریتم تقریب A نیز می‌نامند. مقدار ρ برای مساله مینیمم‌سازی، بزرگ‌تر یا مساوی ۱ و برابر $1+\varepsilon$ است که از نامساوی زیر حاصل می‌شود:

$$\varphi(A(I)) \le (1+\varepsilon)\big(opt(I)\big) \qquad (1)$$

و مقدار ρ برای مساله ماکزیمم‌سازی در بازه‌ی $(0,1)$است و برابر $1-\varepsilon$ است که از نامساوی $\varphi(A(I)) \ge (1-\varepsilon)(opt(I))$ نتیجه می‌شود.

تعریف ۲-۲. یک طرح تقریب برای مساله مینیمم‌سازی π، یک خانواده از الگوریتم‌های $(1+\varepsilon)$ – تقریب برای هر مقدار $\varepsilon > 0$ است.

[1] Delivery time

[DOI: 10.52547/jamlu.19.4.17]

تعریف ۲-۳. طرح تقریب را PTAS می‌گوییم هرگاه پیچیدگی زمانی آن تابعی چندجمله‌ای بر حسب اندازه ورودی باشد.

تعریف ۲-۴. طرح تقریب را FPTAS می‌گوییم هرگاه پیچیدگی زمانی آن تابعی چندجمله‌ای بر اساس اندازه ورودی و مقدار $\frac{1}{\varepsilon}$ باشد.

تعریف ۲-۵. طرح تقریب را DFPTAS گوییم هرگاه برای هر نمونه I از مساله π داشته باشیم $\varphi(A(I)) \le (opt(I)) + (1-\gamma)\varphi(wst(I))$ که در آن $\varphi(wst(I))$ مقدار جواب بدترین حالت برای نمونه I می‌باشد. γ، نسبت دیفرانسیلی نامیده می‌شود.

۳ تقریب‌پذیری و تقریب‌ناپذیری

زمانی‌که با یک مساله بهینه‌سازی ترکیبیاتی روبه‌رو می‌شویم این سوال مطرح می‌شود آیا این مساله راه‌حل دقیق با زمان معقول دارد و اگر راه‌حل بهینه در زمان چندجمله‌ای ندارد آیا می‌توان الگوریتم تقریب برای آن یافت یا خیر. به‌عبارتی آیا مساله تقریب‌پذیر است یا خیر. این بخش به چگونگی تقریب‌پذیری و تقریب‌ناپذیری یک مساله بهینه‌سازی ترکیبیاتی می‌پردازد.

۳-۱تقریب‌ناپذیری

ابتدا به چگونگی نشان دادن تقری‌ناپذیری یک مساله بهینه‌سازی به‌ویژه مساله زمانبندی پرداخته می‌شود. به عبارت بهتر، به این سوال پاسخ داده می‌شود که چگونه می‌توان نشان داد یک مساله داده شده طرح تقریب PTAS یا FPTAS ندارد؟ اطلاع از تقریب‌ناپذیری یک مساله از آنجا اهمیت دارد که زمانی بیهوده صرف‌یافتن راه حل تقریبی نشود. واقعیت این است که نشان دادن عدم وجود طرح تقریبی برای یک مساله، تکنیک‌های خاص و در بعضی موارد دشوار را می‌طلبد. به همین دلیل در اینجا به بیان تکنیک‌های تقریب‌ناپذیری اکتفاشده است که نشان می‌دهند مساله مورد نظر طرح تقریب ندارد مگر زمانی‌که P=NP [۷].

ابزار و تکنیک اصلی برای اثبات عدم FPTAS یا DFPTAS این است که نشان داده شود این مساله یک مساله قویاً NP – سخت است. مسایل قویاً NP – سخت، FPTAS ندارند مگر این که P=NP. اما مساله قویاً NP – سخت چه مساله‌ای است و چرا مسایل قویاً NP- سخت، طرح تقریب FPTAS ندارند؟ مساله قویاً NP – سخت مساله‌ای است که اگر تمامی مقادیر ورودی به صورت یکانی کدگذاری شوند، مساله همچنان NP – سخت بماند [۷].

برای درک این موضوع مساله زمانبندی ماشین‌های موازی[1] با هدف مینیمم‌سازی ماکزیمم زمان اتمام کارها را در نظر بگیرید. مسایل زمانبندی به جهت محیط سیستم زمانبندی یا همان منبع اجرای کار، محدودیت‌های واقع بر محیط، محدودیت‌های واقع بر کار و توابع هدف مختلف، تنوع گسترده‌ای دارند. به همین دلیل برای سادگی

[1] Parallel machines

[DOI: 10.52547/jamlu.19.4.17]

در ادامه از نمادگذاری رایج در حوزه زمانبندی برای نمایش مساله مورد مطالعه به صورت $\alpha|\gamma|\beta$ استفاده شده است. α محدودیت‌های واقع بر محیط ماشین، γ محدودیت‌های واقع بر کار و β، تابع هدف مورد نظر را نشان می‌دهند. جانسون و گری [۷] نشان دادند مساله زمانبندی ماشین‌های موازی با هدف مینیمم‌سازی ماکزیمم زمان اتمام کارها، قویاً NP- سخت است. فرض کنید این مساله که با $P\|C_{\max}$ نمایش داده می‌شود، با ورودی I که به صورت دودویی است یک FPTAS داشته باشد. طبق تعریف، پیچیدگی زمانی آن تابعی بر حسب $\frac{|I|^a}{\varepsilon^b}$ است (b,a مقادیر صحیح مثبت می‌باشند). فرض کنید $\varepsilon=\frac{1}{(\sum_{i=1}^{n}p_i)+1}$ (p_i، زمان اجرای کار i ام است). همچنین فرض کنید الگوریتم H الگوریتمی باشد که طرح تقریب FPTAS برروی نمونه I با ε داده شده را نتیجه دهد. پیچیدگی زمانی این الگوریتم با مقدار $\frac{|I|^a}{(\sum P_i+1)^b}$ کراندار می‌شود. اگر نمونه I به صورت یکانی کدگذاری شود این پیچیدگی می‌تواند براساس نمونه کدشده به صورت یکانی از مرتبه ($O(|I|^{a+b})$ که I به صورت یکانی است، تبدیل شود. از این رو H یک الگوریتم شبه چندجمله‌ای[1] برای مساله مورد نظر است. بنابراین، اگر بنا باشد یک FPTAS برای مساله موجود باشد، حتماً یک الگوریتم شبه‌چندجمله‌ای H برای آن موجود است. اما چون این مساله $P\|C_{\max}$ قویاً NP – سخت است، وجود الگوریتم شبه‌چندجمله‌ای برای آن سبب می‌شود که P=NP و این امکان‌پذیر نیست.

به طور کلی برای اثبات تقریب‌ناپذیری یک مساله می‌توان به قضیه‌ای از گری و جانسون [۷] رجوع کرد که به‌صورت زیر مطرح می‌شود:

قضیه۱.۱.۳. فرض کنید X یک مساله مینیمم‌سازی یا ماکزیمم‌سازی قویاً NP-سخت باشد که در شرایط زیر صادق است:

(i) همه جواب‌های شدنی برای تمامی نمونه‌های I مقادیر صحیح هستند.

(ii) چندجمله‌ای P موجود است که $opt(I)\le(P(I))$ (برای تمامی نمونه ها I که به صورت یکانی نمایش داده می‌شوند).

در این صورت مساله بهینه‌سازی X، FPTAS ندارد مگر این که P=NP.

اما چطور می‌توان نشان داد که مساله مورد نظر PTAS ندارد. یکی از تکنیک‌های قدیمی، تکنیک فاصله[2] است. این راهکار اولین بار در اواسط ۱۹۷۰ توسط سهنی و گنزالس [۹] ، گری و جانسون [۷] استفاده شد. این تکنیک در قضیه زیر ارایه شده است.

[1] Pseudo polynomial algorithm
[2] Gap technique

[DOI: 10.52547/jamlu.19.4.17]

قضیه ۲.۱.۳. فرض کنید X یک مساله تصمیم، NP – سخت باشد. همچنین فرض کنید Y یک مساله مینیمم‌سازی و Z یک زمان چندجمله‌ای قابل محاسبه برای تبدیل مجموعه نمونه‌های X به مجموعه نمونه‌های Y باشند که در شرایط زیر صادق است:

(i) هر نمونه با جواب بله از X به یک نمونه از Y با مقدار تابع هدف حداکثر a، که a عددی صحیح است نگاشته شود.

(ii) هر نمونه با جواب خیر از X به یک نمونه از Y با مقدار تابع هدف حداقل b (a<b) نگاشته شود.

(iii) در این‌صورت مساله Y الگوریتم p – تقریب با زمان چندجمله‌ای با نسبت بدترین حالت $p<\frac{b}{a}$ ندارد مگر این که P=NP. به‌عبارتی مساله Y، PTAS ندارد مگر این که P=NP.

در واقع ایده این تکنیک بر مبنای اثبات NP – سخت بودن مساله با ایجاد فاصله عمیق بین مقادیر هدف نمونه‌هایی با پاسخ خیر و مقادیر هدف نمونه‌هایی با پاسخ آری است.

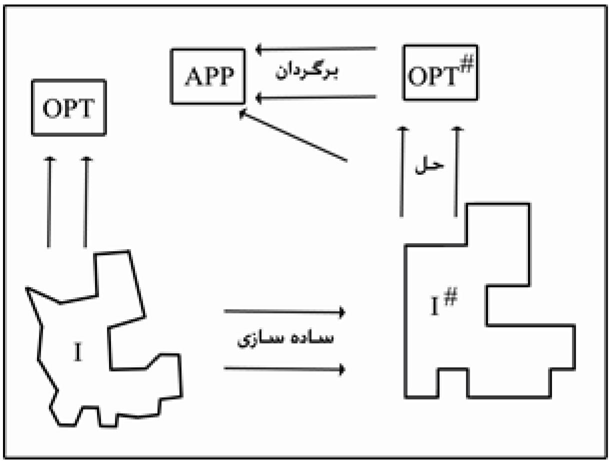
با استفاده از این تکنیک در [۱۰] ثابت شده است مساله $1|rj|\sum F_j$ که در آن $F_j = C_j - r_j$، C_j زمان اتمام کار jام و r_j زمان دسترسی به کار jام است، الگوریتم تقریب با زمان چندجمله‌ای ندارد. برای مطالعه بیشتر در مورد جزییات تقریب‌ناپذیری به [۱۱] مراجعه شود.

۲-۳ وجود طرح تقریب

پیدا کردن یک طرح تقریب برای یک مساله بهینه‌سازی از رده NP - سخت به چگونگی طراحی و آنالیز الگوریتم تقریب مربوط می‌شود. راهکارهای رسیدن به یک طرح تقریب مناسب با زمان اجرای چندجمله‌ای متفاوت هستند. دو ابزار قوی برای به‌دست آوردن طرح تقریب، تغییر در ساختار ورودی مساله و تغییر در ساختار اجرایی الگوریتم است.

تغییر در ساختار ورودی مساله: ساده‌سازی ورودی به این معنی است که نمونه سخت مساله را به یک نمونه ساده تبدیل کنیم و نشان دهیم جواب بهینه به‌دست آمده از نمونه جدید بسیار نزدیک به جواب بهینه نمونه اصلی مساله است. به‌عبارتی اگر I نمونه اصلی مساله باشد و این نمونه، نمونه‌ای دشوار برای حل باشد، در این صورت از طریق ساده‌سازی I به نمونه ساده $I^{\#}$ تبدیل شود. این ساده‌سازی به دقت تقریب وابسته است. هر چقدر ε به صفر نزدیک شود نمونه $I^{\#}$ به نمونه اولیه نزدیک‌تر خواهد شد. رعایت این نکته مهم است که این ساده‌سازی بایستی در زمان چندجمله‌ای از اندازه ورودی انجام شود و جواب بهینه $opt^{\#}$ برای نمونه ساده شده $I^{\#}$ نیز در زمان چندجمله‌ای تعیین شود. همچنین جواب بهینه opt^{*} برای نمونه ساده‌شده $I^{\#}$ قابل تبدیل و برگشت به جواب تقریبی برای نمونه اصلی باشد. فرایند ساده‌سازی در شکل ۲ به تصویر کشیده‌شده است.

[DOI: 10.52547/jamlu.19.4.17]



شکل ۲. ساده‌سازی یک نمونه دشوار

ساده‌سازی معمولا از طریق روش‌های چون گردکردن پارامترهای ورودی، ترکیب کردن، جایگزین کردن و برش انجام می‌شود.

گرد کردن: ساده‌ترین روش تغییر ساختار ورودی، گرد کردن تعدادی از ورودی‌ها است. به‌عنوان مثال می‌توان همه اعداد غیرصحیح را به نزدیک‌ترین عدد صحیح گرد کرد و یا همه ورودی‌ها را با توانی از ۲ گرد کرد.

ترکیب کردن: در این روش می‌توان نمونه‌های با اندازه کوچک را با هم ترکیب و ادغام کرد و نمونه‌هایی با اندازه بزرگ به‌دست آورد.

جایگزین کردن: در این روش می‌توان نمونه‌های شبیه به هم را با نمونه‌های دیگری جایگزین کرد. برای مثال در مساله زمانبندی اگر تعدادی کار مختلف با زمان اجرای یکسان داشتیم، می‌توان آنها را با همان تعدادکار یکسان با زمان اجرای میانگین جایگزین کرد.

برش دادن: در این روش می‌توان نمونه‌هایی با اندازه غیر متعارف را حذف کرد. به عنوان مثال در مساله زمانبندی می‌توان مجموعه‌ای از کارها با طیف وسیعی از زمان‌های اجرا را حذف کرد.

تغییر ساختار اجرای الگوریتم: یکی از راهکارهای دیگر برای یافتن طرح تقریب، تکنیک اصلاح ساختار اجرای الگوریتم است. ایده به این صورت است که یک الگوریتم دقیق اما با سرعت پایین برای حل یک مساله طراحی‌شده است. مثلا الگوریتم A، را به گونه‌ای اصلاح کنیم تا مقداری از داده‌های ورودی حذف شود و داده‌های کمتری برای پردازش باقی بمانند تا بتوان سرعت اجرای الگوریتم را بالا برد. فرض کنید نمونه I_k از مساله π را بتوان به n قسمت تقسیم کرد. $P_1,\dots,P_n$ اجزا این نمونه هستند. دنباله $P_1,\dots,P_k$ را با نمونه جدیدی با نام I_k نمایش می‌دهیم. فرض کنید χ_k مجموعه جواب‌های شدنی نمونه I_k باشد. به عبارتی χ_k مجموعه حالت‌هایی به‌صورت بردار d بعدی است که هر حالت به یک جواب شدنی مربوط می‌شود. در مرحله آغازین مجموعه جواب‌های شدنی برای نمونه I_1 را محاسبه می‌کنیم. در مرحله اجرا فرض کنید می‌خواهیم ورودی P_k را در مرحله k ام الگوریتم اجرا کنیم $(k=2,\dots,n)$. برای این کار تمامی حالت‌های ممکن و تاثیرگذار بر مقدار جواب از مرحله قبل را پیدا می‌کنیم. به‌عبارتی هر حالت از مرحله قبل و جواب‌های شدنی χ_{k-1} را انتخاب و با هم ترکیب می‌کنیم تا به جواب شدنی در مرحله k ام برسیم. بعد از n امین مرحله، الگوریتم برای هر حالت در χ_n

[DOI: 10.52547/jamlu.19.4.17]

با توجه به مقدار تابع هدف محاسبه‌شده، بهترین جواب ممکن را پیدا می‌کند. مرحله آغازین الگوریتم A زمان ثابتی را به خود اختصاص می‌دهد. از آنجایی که هر حالت در χ_{k-1} زمان ثابتی را دارد؛ لذا در k امین مرحله زمان الگوریتم متناسب با $|\chi_n|$ است. بنابراین زمان کل برابر $\sum_{k=1}^{n}\chi_k$ خواهد بود. این مقدار تابعی نمایی از اندازه ورودی مساله است؛ لذا تابع زمانی آن چندجمله‌ای نیست. پس بایستی این انباشت داده‌ها و حالت‌هایی که در روند اجرای الگوریتم به صورت نمایی تولید می‌شود را کاهش داد. به‌عبارتی الگوریتم را به‌گونه‌ای اصلاح کرد که از یک طرف زمان اجرا چندجمله‌ای را کاهش داد و از طرف دیگر جواب بهینه نزدیک به جواب بهینه قبلی پیدا کرد. اما چگونه می‌توان این کار را انجام داد؟

تصور کنید حالت‌های موجود در χ_k به‌صورت نقاط هندسی در فضای اقلیدسی d بعدی باشند. اگر چند نقطه از این فضا بسیار نزدیک به هم باشند در این صورت جواب‌های شدنی آن‌ها نیز بسیار به هم نزدیک خواهد بود. لذا می‌توان از آن‌ها صرفنظر کرد و تنها یک حالت را نگه داشت. از این‌رو، اگر حذف کردن حالت‌ها به روش مناسب و درستی انجام شود تعداد حالت ها کاهش می‌یابد و به دنبال آن پیچیدگی زمانی نیز کاهش خواهد یافت. از آنجایی که برای هر جواب شدنی حذف‌شده یک جواب شدنی نزدیک به آن وجود دارد؛ لذا جواب تقریبی بسیار خوب و با کیفیتی خواهیم داشت.

در ادامه نمونه‌ای از روش ساده‌سازی و روش اصلاح ساختار الگوریتم که برای یک مساله زمانبندی که توسط قاسم و همکاران [۱۲] انجام شده است، ارایه می‌شود.

در این مساله مجموعه‌ای از n کار $J=\{1,2,\ldots,n\}$ بر روی یک تک ماشین که در بازه $[T_1,T_2]$ در دسترس نیست، زمانبندی می‌شود به طوری که هر کار i، زمان اجرای p_i و زمان تحویل q_i را دارد. ماشین در هر زمان حداکثر یک کار می تواند انجام دهد و همه کارها در زمان شروع به کار ماشین آماده برای اجرا می‌باشند. هدف مینیمم کردن

$$\varphi_S(I)=\max_{1\le i\le n}\left(C_i(S)+q_i\right) \qquad (۲)$$

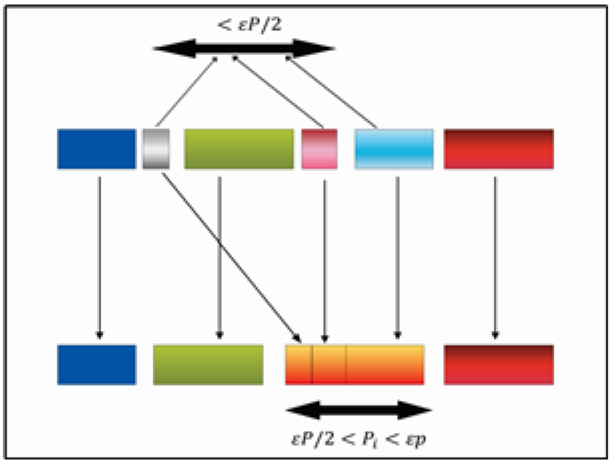
است که در آن $C_i(S)$ زمان اتمام کار i ام در زمانبندی S برای مساله P است. فرض کنید $\mathcal{P}=\sum_{i=1}^{n}p_i$.

برای به دست آوردن یک طرح تقریبی با تغییر ساختار ورودی مساله، نمونه سخت به یک نمونه ساده تبدیل شده است. در مرحله اول بازه $[0,Q]$ که $Q=max_{j\in J}\{q_j\}$ به بازه‌هایی با طول $1/\varepsilon$ تقسیم و سپس هر q_j به مقدار εQ گرد شده است. نمونه جدید I' نام دارد. از طرفی چون $\left[q_j/Q\right]Q\le q_j+Q$ و Q یک کران پایین برای جواب بهینه است، می‌توان نتیجه گرفت:

$$\varphi^*(I')\le\varphi^*(I)+\varepsilon Q\le(1+\varepsilon)\varphi^*(I) \qquad (۳)$$

این مرحله از ساده‌سازی برای به‌دست آوردن نمونه جدید در زمان $O(n)$ انجام می‌شود. دومین مرحله ساده‌سازی ورودی به این صورت است که مجموعه کارها به زیرمجموعه‌های J' تقسیم شده است. سپس

کارهای کوچک (کارهایی که طول اجرای کمتر از $\varepsilon P/۲$ دارند) در هر یک از این زیرمجموعه ها با هم ترکیب شده تا کارهای بزرگ (کارهایی که طول اجرای بزرگ‌تر از $\varepsilon P/۲$ دارند) به‌دست آید (شکل ۳). در پایان تمامی کارها بر اساس ترتیب جکسون، ترتیب نزولی زمان‌های تحویل کارها مجددا اندیس‌گذاری می‌شوند. نمونه به‌دست آمده از ساده‌سازی مرحله دوم با I'' و جواب به‌دست آمده بر اساس ترتیب جکسون برای این نمونه با $\varphi_{JS}(I'')$ نمایش داده می‌شود.



شکل ۳. تصویری از مرحله ساده‌سازی ورودی

مرحله بعد استفاده از تکنیک اصلاح ساختار اجرای الگوریتم است. ایده این روش بدین صورت است که یک الگوریتم دقیق اما با سرعت کند که برای حل مساله مطرح شده است به‌گونه‌ای اصلاح شود تا مقداری از داده‌های ورودی حذف شود و داده‌های کمتری برای پردازش باقی ماند و در نتیجه آن سرعت الگوریتم بالا رود. با به‌کارگیری رویکرد برنامه‌ریزی پویا که توسط قاسم [۱۳] مطرح شده است، طرح تقریب جدید ارایه می‌شود. این الگوریتم، الگوریتم A، [۱۳] به صورت مکرر مجموعه‌های χ_j متشکل از حالت‌های $[t,f]$ را تولید می‌کند که در آن t نشان‌دهنده زمان اتمام آخرین کار، قبل از $T_۱$ و f نشان‌دهنده ماکزیمم زمان تحویل کارها در زمانبندی مورد نظر است.

الگوریتم A

i. قرار دهید

$$\chi_۱^{\#} = \{[\circ, T_۲ + p_۱ + q_۱], [p_۱, p_۱ + q_۱]\}$$

ii. برای $j \in \{۲, ۳, \dots, \bar{n}\}$ که $\bar{n} = \min\{n, \frac{۳}{\varepsilon}\}$

$$\chi_j^{\#} = \varnothing$$

برای هر حالت $[t,f]$ در $\chi_{j-۱}^{\#}$:

۱) قرار بده

[DOI: 10.52547/jamlu.19.4.17]

$\left[t.\max\left\{f\ ,T_2+\sum_{i=1}^{j}p_i-t+q_j\right\}\right]$ در $\chi_j^{\#}$

قرار بده $\left[t+p_j\ ,max\ \left\{f\ ,t+p_j\ +q_j\right\}\right]$ در $\chi_j^{\#}$. اگر $t+p_j\le T_1$.

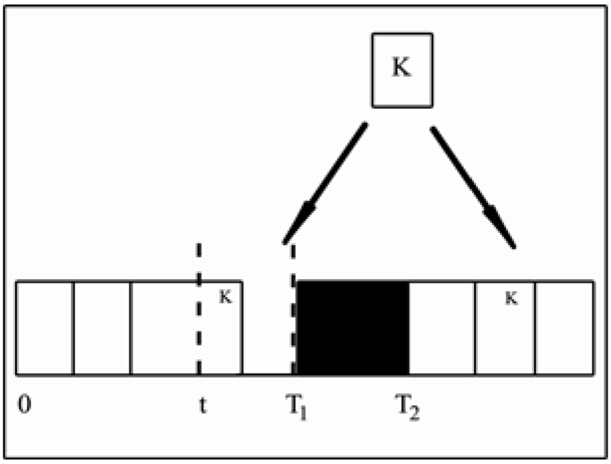
۲) حذف کن $\chi_{j-1}^{\#}$.

حالت‌های $[t,f\ ]$را با کوچک‌ترین مقدار t و f انتخاب کنید و مجموعه $\chi_j^{\#}$ متشکل از این حالت‌ها را بسازید.

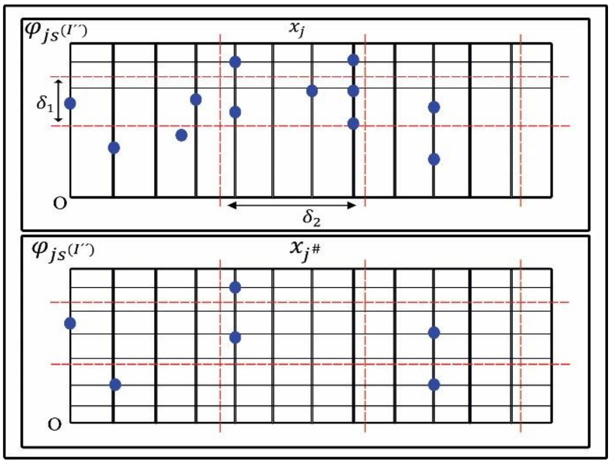
iii) محاسبه کن $\varphi_{A_\varepsilon}\ (I\ '')=min_{[t,f\ ]\in\chi_{\bar{n}}^{\#}}\ \{f\ \}$

همان‌طور که در الگوریتم A بیان شده است متناسب با هر حالت $[t,f\ ]$ دو حالت دیگر تولید می‌شود که ناشی از نحوه قرارگیری کار k ام بر روی ماشین، قبل و بعد از بازه غیرقابل دسترس است که در شکل ۴ نیز به تصویر کشیده شده است.

در شکل ۵ نشان داده شده تعداد حالت‌ها پس از تغییر ساختار الگوریتم کاهش یافته است و زمان اجرا نیز سریع‌تر شده است. زمان اجرا به صورت $O\left(\bar{n}In\bar{n}+\bar{n}^3/\varepsilon^2\right)$ به‌دست آمده است.



شکل ٤. نحوه قرارگیری کار k ام روی ماشین



شکل ٥. کاهش تعداد حالت‌ها پس از تغییر ساختار الگوریتم

٤ طرح‌های تقریب

رویکرد یافتن طرح تقریب برای سنجش کارایی الگوریتم‌های حل مسایل بهینه‌سازی ترکیبیاتی استفاده می‌شود. از میان پژوهش‌های مختلف که از این معیار استفاده کرده‌اند، می‌توان به مراجع [۵، ۱۴، ۱۵، ۱۶] اشاره کرد. از آنجایی که دو تابع هدف مینیمم‌سازی مجموع وزن دار و مینیمم‌سازی زمان تحویل کارها نسبت به سایر توابع هدف مسایل زمانبندی بسیار مورد توجه محققین قرار دارد، این بخش نتایج تقریب بر روی مسایل زمانبندی با این دو تابع هدف و محدودیت‌های زیر را ارایه می‌کند.

- محدودیت زمان دسترسی به هر کار[1]: در این محدودیت هر کار در زمانی مشخص در دسترس قرار می‌گیرد.
- محدودیت از قبل مشخص شده[2]: در این محدودیت زمان اجرای یک کار قبل از کار دیگر می‌تواند باشد.
- محدودیت دسترسی ماشین: در این محدودیت ماشین به دلایلی از جمله خرابی و تعمیر از دسترس خارج می‌شود.

٤-۱ مساله زمانبندی با هدف مینیمم‌سازی مجموع زمان وزن‌دار کارها

ابتدا به تعریف مساله مورد نظر می‌پردازیم. مجموعه‌ای از n کار به‌صورت $J_1,\dots,J_n$ در اختیار است. هر کار J_i زمان اجرای p_i و وزن نامنفی W_i را دارد. هدف این است تا این مجموعه کارها را به‌گونه‌ای روی ماشین اجرا کنیم که تابع هدف $\sum_n W_i C_i$ مینیمم شود. در اینجا C_i زمان اتمام کار J_i در زمانبندی مورد نظر است. ماشین در هر زمان یک کار می‌تواند انجام دهد. این مساله به‌صورت بهینه با رعایت قانون اسمیت [۱۷] در زمان چندجمله‌ای حل می‌شود. این قانون که به قانون WSPT معروف است بیان می‌کند که اگر کارها را بر اساس ترتیب غیرصعودی $\frac{W_j}{p_j}$ زمانبندی شود، دنباله به‌دست آمده منجر به جواب بهینه می‌شود. حال اگر محدودیت‌هایی از قبیل محدودیت کار به مساله اضافه کنیم، مساله در رده مسایل NP – سخت علوم کامپیوتر قرار می‌گیرد [۱۸]. بنابراین بایستی با طرح الگوریتم تقریب یا ارایه طرح‌های تقریبی جوابی نزدیک به جواب بهینه را پیدا کرد.

در [۱۹] نشان داده شده است نسخه بدون وزن این مساله با محدودیت زمان دسترسی برای هر کار یعنی $1|rj|\sum C_j$ به طور قوی، NP – سخت است. لذا اولین ثابت تقریب با مقدار ۲ برای این مساله در [۲۰] مطرح شد. سپس نسخه شکست کار[3] این مساله که در آن اجرای کار می‌تواند قطع شود و در زمانی دیگر ادامه پیدا کند، توسط بیکر و همکاران در [۲۱] مطرح شده و نویسندگان تلاش کردند این مساله یعنی $1|rj,Pmtn|\sum C_j$ در زمان چندجمله‌ای حل شود.

---
[1] Release date
[2] Precedence constraint
[3] Preemptive job

[DOI: 10.52547/jamlu.19.4.17]

نسخه وزن‌دار مساله، با محدودیت زمان دسترسی برای هر کار ، یعنی $1|rj|\sum W_j C_j$ توسط شالز [۲۲] و هال و همکاران [۲۳] بررسی شد و الگوریتم‌های ۳- تقریب را برای آن مطرح کردند. همچنین هال و همکاران [۲۳] مساله را با محدودیت زمان از پیش تعیین‌شده برای اجرای هر کار در نظر گرفتند و برای این مساله یعنی $1|prec|\sum W_j C_j$ الگوریتم ۲-تقریب ارایه کردند. مساله مینیمم‌سازی مجموعه وزن‌دار زمان اتمام کار با محدودیت دسترسی کار و تعیین محدودیت برای اجرای هر کار در [۲۳] مطرح شد و یک الگوریتم ۳- تقریب برای آن به‌دست آمد.

محدودیت دسترسی ماشین که به نظر می‌رسد در واقعیت بسیار رخ می‌دهد چرا که هر ماشینی می‌تواند به دلایلی از جمله خرابی و تعمیر در یک بازه مشخص و ثابتی از دسترس خارج شود، در ادبیات مختلف مورد مطالعه قرار گرفته است. ساده‌ترین حالت، مساله زمانبندی روی یک ماشین با هدف مینیمم‌سازی زمان اتمام کار و محدودیت دسترسی ماشین در یک بازه مشخص است. این مساله با $1,h_1||\sum C_j$ نمایش داده می‌شود. h_1 نماد محدودیت دسترسی ماشین در یک بازه ثابت است. در [۲۴] و [۲۵] ثابت شد که این مساله یک مساله NP – سخت است. راهکارهای مختلفی برای حل این مساله ارایه شد. در چندین تحقیق زمان اجرای بدترین حالت از روش‌های ابتکاری به‌دست آمده است. به عنوان نمونه در [۲۵] یک الگوریتم شاخه و کران برای حل این مساله ارایه شد. نسخه وزن‌دار این مساله یعنی $1,h_1||\sum C_j$، توسط قاسم و شو [۲۶] مورد مطالعه قرار گرفت و روش‌های دقیق برای حل این مساله ارایه کردند. همچنین در [۲۷] یک الگوریتم ε – تقریب با استفاده از جواب نسخه شکست کار مساله که در [۲۸] ارایه شده بود، مطرح شد. اما برای این مساله طرح‌های تقریب نیز ارایه‌شده است. در [۲۷] یک طرح تقریب با زمان چندجمله‌ای FPTAS، و پیچیدگی $O\left(n^4/\varepsilon^2\right)$ مطرح شده است. در [۲۹] یک الگوریتم ۲- تقریب با پیچیدگی $O\left(n^2\right)$ که n تعداد کارها است، ارایه شده است.

در [۳۰] یک FPTAS با پیچیدگی $O(n^2/\varepsilon^2)$ مطرح شد. همچنین یک FPTAS با زمان $O\left(n^3/\varepsilon^2\right)$ با استفاده از تکنیک اصلاح ساختار الگوریتم برنامه‌ریزی پویا برای مساله $2,h_1||\sum C_j$ در [۳۱] مطرح شد و در نسخه وزن‌دار آن در [۳۲] یعنی $2,h_1||\sum W_j C_j$ یک FPTAS با زمان $O\left(n^3/\varepsilon^2\right)$ بیان شد. براساس تحقیق نگارنده، از میان تحقیقات صورت گرفته، تنها پژوهش[۳۳] به بررسی طرح تقریب دیفرانسیلی با زمان چندجمله‌ای، DFPTAS برای مساله زمانبندی روی یک تک ماشین با تابع هدف مینیمم‌سازی زمان اتمام وزن‌دار، $1,h_1||\sum W_j C_j$، پرداخته است که این پژوهش نشان داد الگوریتم مبتنی بر قانون WSPT نمی‌تواند منجر به تقریب دیفرانسیلی برای این مساله شود، اما تغییر ساختار آن می‌تواند یک الگوریتم تقریب دیفرانسیلی با نسبت $\frac{3-\sqrt{5}}{2}$ ارایه دهد. جدول ۱ خلاصه‌ای از طرح‌های تقریب به همراه پیچیدگی زمانی آنها برای مساله زمانبندی با هدف مینیمم‌سازی مجموع زمان وزن‌دار کارها را نشان می‌دهد که در بعضی از مسایل مانند $1,h_1||\sum W_j C_j$ تلاش شده است تا پیچیدگی زمانی طرح‌های قبلی بهبود پیدا کند و طرح‌های کاراتر و سریع‌تری ارایه شود.

[DOI: 10.52547/jamlu.19.4.17]

جدول ۱. خلاصه طرح‌های تقریب برای مساله زمانبندی با هدف کمینه‌سازی مجموع زمان وزن‌دار کارها

مساله	نتیجه	مرجع
$1,h_1 \| \sum C_j$	$FPTAS \quad o\left(n^3/\varepsilon^2\right)$	[۲۸]
$1,h_1 \| \sum W_j C_j$	$FPTAS \quad o\left(n^4/\varepsilon^2\right)$	[۲۸]
$1,h_1 \| \sum W_j C_j$	$FPTAS \quad o\left(n^2/\varepsilon^2\right)$	[۳۰]
$2,h_1 \| \sum C_j$	$FPTAS \quad o\left(n^3/\varepsilon^2\right)$	[۳۱]
$2,h_1 \| \sum W_j C_j$	$FPTAS \quad o\left(n^3/\varepsilon^2\right)$	[۳۲]
$1,h_1 \| \sum W_j C_j$	DFPAS	[۳۳]

۴-۲ مساله زمانبندی با هدف مینیمم‌سازی ماکزیمم زمان تحویل

در این بخش نتایج مربوط به طرح‌های تقریب مساله زمانبندی روی تک ماشین با محدودیت دسترسی ماشین و زمان دسترسی کار با هدف مینیمم‌سازی ماکزیمم زمان تحویل ارایه می‌شود.

در مساله زمانبندی با هدف مینیمم‌سازی ماکزیمم زمان تحویل که به مینیمم‌سازی ماکزیمم تاخیر[1] نیز شهرت دارد، می‌خواهیم مجموعه‌ای از n کار $J=\{1,2,...,n\}$ را بر روی یک تک ماشین زمانبندی کنیم به‌طوری که هر کار زمان اجرای p_j و زمان تحویل q_j را دارد. ماشین در هر زمان حداکثر یک کار می‌تواند انجام دهد و همه کارها در زمان شروع به‌کار ماشین آماده برای اجرا می‌باشند. هدف، مینیمم‌سازی تابع زیر است:

$$\varphi_S(I)=\max_{1\le i\le n}\left(C_i(S)+q_i\right) \tag{۴}$$

در [۳۴] ثابت شده است این مساله با مساله مینیمم‌سازی ماکزیمم تاخیر با زمان های موعد معادل است. در سال ۱۹۸۲ قانون جکسون بر اساس ترتیب نزولی زمان‌های تحویل کارها تعریف شد(یعنی $q_1 \ge q_2 \ge \ldots \ge q_n$) و نشان داده شد این قانون منجر به حل بهینه مساله می‌شود. اما زمانی که محدودیت‌هایی از قبیل بازه دسترسی ماشین به مساله اضافه می‌شود این مساله در دسته مسایل NP- سخت قرار می‌گیرد و حل بهینه مساله در زمان چندجمله‌ای امکان‌پذیر نیست [۳۵].

مساله زمانبندی n کار بر روی یک ماشین با هدف مینیمم‌سازی ماکزیمم زمان تحویل بدون هیچ محدودیتی، $1 \| L_{max}$، نیز به‌صورت گسترده‌ای در ادبیات مربوطه مورد توجه قرار گرفته است. برای مطالعه این مساله می‌توان به [۳۸-۳۶] مراجعه کرد. اما آنچه که در واقعیت می‌تواند مورد توجه قرار گیرد عدم دسترسی به ماشین در بازه‌ای مشخص است. به هر حال ماشین می‌تواند به دلایل مختلف از جمله خرابی، تعمیر و یا موارد دیگر در بازه‌ای مشخص از دسترس خارج شود. لی [۳۷] ثابت کرد مساله زمانبندی روی یک تک ماشین تحت بازه ثابت غیرقابل دسترس و تابع هدف مینیمم‌سازی ماکزیمم تأخیر، $1,h_1 \| L_{max}$، یک مساله NP-سخت است.

[1] Lateness

[DOI: 10.52547/jamlu.19.4.17]

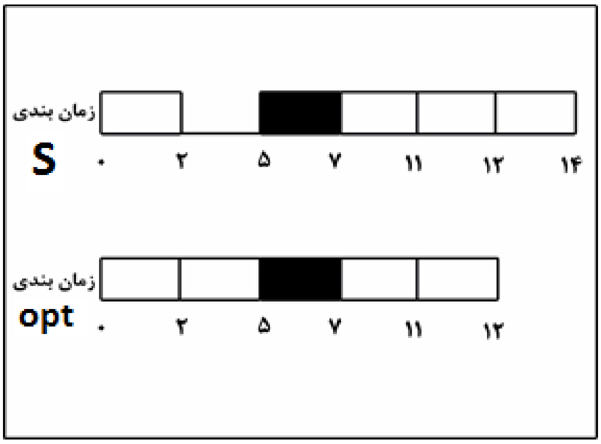
$h_{۱}$ نماد محدودیت دسترسی ماشین در یک بازه ثابت است. از این رو بررسی این مساله و ارایه راه حل برای آن بسیار مورد توجه قرار گرفته است. برای درک بهتر از این مساله مثال زیر را در نظر بگیرید.

مثال : فرض کنید داده‌های زیر از چهار کار در اختیار داریم :

$p_{۱} = ۳ , q_{۱} = ۵ , p_{۲} = ۴ ,$
$q_{۲} = ۴ , p_{۳} = ۲ , q_{۳} = ۳,$

$p_{۴} = ۱ , q_{۴} = ۱$

همچنین فرض کنید ماشین در بازه [۷ ،۵] در دسترس نیست. در این صورت یک راه حل، زمانبندی کارها براساس قانون جکسون (دنباله s در شکل ۶) با مقدار تابع هدف $\varphi(s) = ۱۶$ است که از رابطه (۱) به دست می‌آید، در حالی که جواب بهینه $\varphi^*(s) = ۱۵$ است.



شکل ٦. زمانبندی با هدف مینیمم سازی ماکزیمم زمان تحویل

لی [۳۷]، در ساده‌ترین حالت مساله، $۱||L_{max}$، قانون جکسون را ارایه کرد و نشان داد خطای جواب به‌دست آمده از به‌کارگیری این قانون نسبت به جواب بهینه از مقدار $\max\{p_i\}$ تجاوز نمی‌کند (p_i زمان اجرای کار i ام است). قاسم و ها آوری [۳۸] مساله را با محدودیت زمان‌های دسترسی برای هر کار، $۱|r_j|L_{max}$ در نظر گرفتند و نشان دادند قانون جکسون، کران بدترین حالت با مقدار ۲ را برای این مساله نتیجه می‌دهد تا این‌که یان و همکاران [۴۰] طرح تقریبی با زمان اجرای چندجمله‌ای PTAS برای این مساله پیدا کردند و قاسم [۱۳] نیز یک طرح تقریبی قوی چندجمله‌ای FPTAS برای این مساله مطرح کرد. در [۱۲] یک طرح تقریبی جدید با زمان چندجمله‌ای با استفاده از روش تغییر در ساختار ورودی مساله و سپس استفاده از روش برنامه‌ریزی پویا مطرح شد. نویسندگان تلاش کردند نمونه سخت مساله را از طریق ترکیب کارهای کوچک به یک نمونه ساده تبدیل کنند. سپس با کمک رویکرد برنامه‌ریزی پویا و تکنیک اصلاح ساختار اجرای الگوریتم، طرح تقریب مناسب را ارایه کردند. ایده این روش بدین صورت است که یک الگوریتم دقیق اما با سرعت کند که برای حل مساله مطرح شده است، به‌گونه‌ای اصلاح شود تا مقداری از داده‌های ورودی حذف شود و داده‌های کمتری برای پردازش باقی ماند تا سرعت الگوریتم بالا رود.

در [۳۹] دو طرح تقریب چندجمله‌ای برای مساله $1|r_j|L_{\max}$ طراحی شد که طرح اول با استفاده از برنامه‌ریزی پویا ارایه شد و طرح دوم بر اساس تقسیم مجموعه کارها به دو مجموعه کارهای کوچک و بزرگ و استفاده از قانون جکسون مطرح شد. در [۴۰] مساله زمانبندی روی یک تک ماشین با زمان دسترسی و زمان تحویل برای هر کار با هدف مینیمم‌سازی ماکزیمم تاخیر در نظر گرفته شد. نکته جالب در اینجا بود که چهار سناریو برای این مساله مطرح شد. در سناریوی اول برای هر کار یک مهلت مشخص تعریف شد. بدین معنی که کار بایستی قبل از زمان تعیین شده به اتمام برسد. این مساله با $1|r_j, C_{max} \le d|L_{max}$ نشان داده شد و برای آن یک طرح تقریب، PTAS ارایه شد.

نویسندگان در سناریوی دوم، مساله را با دو تابع هدف در نظر گرفتند، $1|r_j|C_{max}, L_{max}$ و یک طرح تقریب با زمان $O\left(nln(n)n^{\left(1+\frac{4}{\varepsilon}\right)}\right)$ برای مساله ارایه کردند.

در سناریوهای سوم و چهارم محدودیت دسترسی برای ماشین در نظر گرفته شد. محدودیت اول در یک بازه ثابت و مشخص است که به دلیل خرابی، نگهداری و سایر دلایل ممکن است برای یک ماشین رخ دهد. این مساله با $1, h_1|r_j|L_{max}$، نمایش داده می‌شود. محدودیت دوم مربوط به عامل[1] است که وظیفه اجرای کارها را به عهده دارد. این عامل یک بازه زمانی است که یک کار می تواند در آن اجرا شود اما نمی‌تواند شروع شود یا به اتمام برسدکه با $1, ONA|r_j|L_{max}$نشان داده می‌شود. برای هر دوی این سناریوها یکPTAS مطرح شد. معیار استفاده‌شده در این پژوهش‌ها صرفا به جواب بهینه توجه می‌کند و نمی‌تواند موقعیت جواب حاصل از الگوریتم را در مقایسه با جواب بهینه و بدترین جواب بسنجد. معیار دیگری وجود دارد که نسبت تفاوت بین مقدار جواب تولیدشده توسط الگوریتم و جواب بدترین حالت به اختلاف مقدار بهینه و مقدار بدترین حالت را در نظر می‌گیرد. این معیار به معیار نسبت تقریب دیفرانسیلی معروف است. در واقع این معیار نشان می‌دهد جواب حاصل از الگوریتم چقدر از بدترین جواب دور است و چقدر به جواب بهینه نزدیک است که در این پژوهش در بخش بعد مورد بررسی قرار می‌گیرد. خلاصه نتایج به‌دست آمده بر روی طرح‌های تقریب به‌همراه پیچیدگی زمانی آنها برای مساله زمانبندی با هدف مینیمم‌سازی ماکزیمم زمان تحویل در جدول ۲ ارایه شده است.

جدول ۲. خلاصه طرح‌های تقریب برای مساله زمانبندی با هدف کمینه‌سازی بیشینه زمان تحویل

مساله	نتیجه	مرجع
$1, h_1 \|\| L_{max}$	۲- تقریب (قانون جکسون)	[۳۷]
$1, h_1 \|\| L_{max}$	$O\left(nln(n)+n/2^{\frac{1}{\varepsilon}}\right)$, $PTAS$	[۱۶]
$1, h_1 \|\| L_{max}$	$O\left(n^3/\varepsilon^2\right)$ $FPTAS$	[۱۳]

[1] Operator Non-availability interval

[DOI: 10.52547/jamlu.19.4.17]

$۱,h_۱ \parallel L_{max}$	$O\left(nln(n)+\left(\frac{۱}{\varepsilon}\right)۲^{o\left(\frac{۱}{\varepsilon}\right)}\right),\ PTAS$	[۱۲]
$۱,h_۱ \parallel L_{max}$	$O\left(nln(n)+\min\left\{n,\frac{۳}{۴}\right\}^{۳}/\varepsilon^{۲}\right),\ FPTAS$	[۱۲]
$۱\|r_j\|L_{\max}$	$O(۱۶^{۱/\varepsilon}(\frac{n}{\varepsilon})^{۳+۴/\varepsilon}),\ PTAS$	[۳۹]
$۱\|r_j\|L_{\max}$	$O(n\log n+n(۴/\varepsilon)^{۸/\varepsilon+۲+۸/\varepsilon^{۲}}),\ PTAS$	[۳۹]
$۱,\|r_j\|L_{max}$	$O\left(nln(n)+n\left(\frac{۲}{\varepsilon}\right)^{\frac{۱۶}{\varepsilon^{۲}}+\frac{۸}{۴}}\right): PTAS$	[۴۰]
$۱,h_۱\|r_j\|L_{max}$	$O\left(۲^{\frac{۴}{\varepsilon^{۲}}}\left(\frac{۴}{\varepsilon^{۲}}\right)^{\frac{۱}{۴}}\ln(n)\left(n^{۱+\frac{۲۰}{۳۴}}+n^{۱+\frac{۱۲}{۲۴}}\right)\right),\ FPTAS$	[۴۰]

۵ طرح تقریب دیفرانسیلی

طرح تقریب دیفرانسیلی علاوه بر اندازه‌گیری فاصله جواب بهینه و جواب الگوریتم مطرح‌شده برای حل مساله مورد نظر، اختلاف جواب الگوریتم با جواب بدترین حالت را نیز اندازه می‌گیرد و بنابراین طرح کاراتری نسبت به طرح‌های قبلی است.

براساس تحقیق نگارنده، تنها پژوهش [۳۳] به بررسی تقریب دیفرانسیلی برای مساله زمانبندی روی یک تک ماشین با تابع هدف مینمم‌سازی زمان اتمام وزن‌دار پرداخته است. در [۴۱] مساله مینیم‌سازی ماکزیمم زمان تحویل روی یک تک ماشین در یک بازه غیرقابل دسترس مورد بررسی قرار گرفت. محور این بررسی وجود نسبت تقریب دیفرانسیلی به عنوان معیاری برای کارایی الگوریتم‌های تقریب بود. در این راستا، ابتدا با مطرح کردن دنباله جکسون و اعمال تغییرات بر روی آن الگوریتمی ارایه شد. این الگوریتم نتوانست در حالت کلی یک نسبت ثابت دیفرانسیلی ارایه دهد. علیرغم این‌که در بعضی حالات تقریب خوبی از جواب بهینه ارایه کردند. از این‌رو الگوریتم دیگری بایستی مطرح شود تا وجود نسبت دیفرانسیلی برای آن بررسی شود.

در این بخش به بررسی وجود تقریب دیفرانسیلی برای مساله زمانبندی روی یک تک ماشین با تابع هدف مساله مینیمم‌سازی ماکزیمم تأخیر می‌پردازیم که تا کنون مورد مطالعه قرار نگرفته است. ویژگی تابع هدف این مساله در این است که مقدار ماکزیمم تاخیر در زمان نامعلومی رخ می‌دهد. ازاین‌رو، این پیچیدگی سبب خواهد شد که این سوال مطرح شود که آیا اساسا نسبت دیفرانسیلی برای این مساله با این تابع هدف خاص به‌دست خواهد آمد یا خیر.

در این راستا، الگوریتم B بر اساس طرح تقریب زمان چندجمله‌ای (PTAS) که در [۴۲] مطرح شده، ارایه داده می‌شود. با به‌کارگیری ایده گرفته‌شده از این الگوریتم امکان ندارد که یک نسبت ثابت در حالت کلی به‌دست آورد. این الگوریتم به‌صورت زیر توصیف می‌شود:

[DOI: 10.52547/jamlu.19.4.17]

الگوریتم B

فرض کنید ε>۰ یک ثابت داده شده باشد به‌طوری که $1/\varepsilon$ یک عدد صحیح مثبت باشد.

(i) قرار دهید $G_\varepsilon = \{i | p_i > \varepsilon t_1 / 2\}$ (مجموعه کارهای بزرگ) و $G'_\varepsilon = J - G_\varepsilon$ (مجموعه کارهای کوچک).

(ii) تمام تخصیص‌های کارهای بزرگ قبل و بعد از بازه غیرقابل دسترس را درنظر بگیرید (باید توجه شود که در اینجا کران تعداد کارهای بزرگ که می‌توانند قبل از t_1 قرار گیرند، مقدار $2/\varepsilon$ است و این منجر به این خواهد شد که کران بیشترین مقدار از تخصیص‌های ممکن مقدار $n^{2/\varepsilon}$ شود).

(iii) هر تخصیص از کارهای بزرگ تولید شده در مرحله (ii) را با کارهای کوچک کامل کنید تا یک زمانبندی شدنی تولید شود. البته قانون جکسون بایستی قبل و بعد از بازه غیرقابل دسترس رعایت شود.

(iv) بهترین زمانبندی را در بین همه زمانبندی‌های تولید شده در گام (iii) انتخاب کنید. بهترین زمانبندی را با σ نمایش دهید.

با توجه به زمانبندی به‌دست آمده از الگوریتم B می‌توان نشان داد:

$$\varphi_\sigma(P) - \varphi^*(P) < \varepsilon t_1 / 2 \qquad (5)$$

علیرغم این‌که این الگوریتم یک PTAS در حالت مطلق است، نمی‌توان از وجود یک تقریب دیفرانسیل مطمئن شد. چون جمله $t_1/2$ می‌تواند از $\varphi_{wst}(P) - \varphi^*(P)$ بزرگتر شود. دو کار زیر یک مثال نقض برای رابطه $\varphi_{wst}(P) - \varphi^*(P) \ge t_1/2$ است.

به عنوان مثال فرض کنید $J = \{1,2\}$ و کارها به‌صورت زیر تعریف می‌شوند

$$q_1 = T,\ \ P_1 = 1,\ \ q_2 = 0,\ \ P_2 = T, \qquad (6)$$
$$t_2 = T + 1,\ \ t_1 = T$$

$$\varphi^*(P) = 2T + 1, \qquad (7)$$
$$\varphi_{wst}(P) = 2T + 2.$$

لذا

$$\varphi_{wst}(P) - \varphi^*(P) = 1 \qquad (8)$$

اما چگونه می‌توان الگوریتم تقریب دیفرانسیلی برای مساله ارایه داد؟ ایده اصلی بر پایه $PTAS$ طراحی‌شده برای این مساله است. ابتدا فرض زیر را انجام می‌دهیم:

$$\varphi_{WST}(P) \ge (1+\varepsilon)\varphi^*(P) \qquad (9)$$

این فرض در عمل قابل انجام است زیرا اگر برقرار نباشد بدترین حالت یک جواب خیلی خوب برای مساله است. به عبارتی اگر:

$\varphi_{WST} < (1+\varepsilon)\varphi^*(P)$ باشد هر جواب شدنی به خوبی بدترین حالت است و دیگر نیازی به تحلیل تقریب دیفرانسیلی نیست. چون بدترین حالت و جواب بهینه به هم نزدیک هستند و هر جواب شدنی قابل قبول است. لذا

[DOI: 10.52547/jamlu.19.4.17]

این فرض می‌تواند برقرار باشد. تحت این فرض مساله یک PTAS دارد. فرض کنید $\varepsilon' > ۰$، $\varepsilon' = \varepsilon^{۲}$ و $\pi_{\varepsilon'}$ جواب به‌دست آمده از $FPTAS$ مطرح‌شده در [۴۱] باشد در این صورت :

$$\begin{aligned} &\varphi_{\pi_{\varepsilon'}}(P) \le (۱+\varepsilon')\varphi^*(P) \\ &= (۱+\varepsilon^{۲})\varphi^*(P) \\ &= (۱+\varepsilon)\varphi^*(P) + (\varepsilon+\varepsilon^{۲})\varphi^*(P) \\ &= (۱+\varepsilon)\varphi^*(P) + \varepsilon\ (۱+\varepsilon)\varphi^*(P) \\ &\Rightarrow \varphi_{\pi_{\varepsilon'}}(P) \le (۱+\varepsilon)\varphi^* + \varepsilon\varphi_{WST}(P) \end{aligned} \tag{۱۰}$$

بنابراین طبق تعریف (۲-۵) می‌توان وجود طرح تقریب دیفرانسیلی را برای این مساله نتیجه گرفت.

۶ نتیجه و جمع‌بندی

در این مقاله به نتایج به‌دست آمده بر روی طرح‌های تقریب بر روی دو مساله زمانبندی با توابع هدف مینیمم‌سازی مجموع زمان وزن‌دار اتمام کارها و مینیمم‌سازی زمان تحویل کارها پرداخته شد. طرح‌های تقریب برای مسایل بهینه‌سازی ترکیبیاتی که در رده NP-سخت از دسته‌بندی مسایل علوم کامپیوتر قرار می‌گیرند بسیار راه‌گشا است. در واقع طرح تقریب کارا راهکاری برای سنجش کارایی الگوریتم ارایه‌شده برای حل مساله مورد نظر است. مساله‌ای که یافتن جواب بهینه در زمانی معقول برای آن ناممکن است. آخرین دستاوردها و طرح‌های تقریبی برای دو مساله ذکر شده در این مقاله ارایه شد. این طرح‌ها می‌تواند در آینده از نظر پیچیدگی زمان اجرا بهبود یابند و برای سایر مسایل زمانبندی همچون مسایل زمانبندی روی ماشین های موازی مورد توجه قرار گیرند.

منابع

[1] Samouei P, Fattahi P. (2017). An Analytical and comparative approach for using Metaheuristic algorithms for job shop scheduling problems. Journal of Operational Research in Its Applications.; 14 (1), 63-76.

[2] Beheshtinia M, Hassani Bidgoli M. (2017). Scheduling in Flexible Jobshop Environment Considering Assembling and Sequence Dependent Processing Times. Journal of Operational Research in Its Applications. 13 (4), 21-37.

[3] Bezdan, T., Zivkovic, M., Bacanin, N., Strumberger, I., Tuba, E., & Tuba, M. (2022). Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm. Journal of Intelligent & Fuzzy Systems, 42(1), 411-423.

[4] Martinelli, R., Mariano, F. C. M. Q., & Martins, C. B. (2022). Single machine scheduling in make to order environments: a systematic review. Computers & Industrial Engineering, 108-190.

[5] Demange, M., Paschos, V. T. (1996). On an approximation measure founded on the links between optimization and polynomial approximation theory. Theoretical Computer Science, 158(1-2), 117-141.

[6] Garey, M. R., Graham, R. L., Ullman, J. D. (1972, May). Worst-case analysis of memory allocation algorithms. In Proceedings of the fourth annual ACM symposium on Theory of computing (pp. 143-150). ACM.

[7] Garey, M. R., Johnson, D. S. (2002). Computers and intractability (Vol. 29). New York: WH freeman.

[8] Demange, M., Grisoni, P., Paschos, V. T. (1998). Differential approximation algorithms for some combinatorial optimization problems. Theoretical Computer Science, 209(1-2), 107-122.

[9] Sahni, S., Gonzalez, T. (1976). P-complete approximation problems. Journal of the ACM (JACM), 23(3), 555-565.

[10] Kellerer, H., Tautenhahn, T., Woeginger, G. (1999). Approximability and nonapproximability results for minimizing total flow time on a single machine. SIAM Journal on Computing, 28(4),

[DOI: 10.52547/jamlu.19.4.17]

1155-1166.
[11] Crescenzi, P., Kann, V. (1997, July). Approximation on the web: A compendium of NP optimization problems. In International Workshop on Randomization and Approximation Techniques in Computer Science (pp. 111-118). Springer, Berlin, Heidelberg.
[12] Kacem, I., Kellerer, H., Seifaddini, M. (2016). Efficient approximation schemes for the maximum lateness minimization on a single machine with a fixed operator or machine non-availability interval. Journal of Combinatorial Optimization, 32(3), 970-981.
[13] Kacem, I. (2009). Approximation algorithms for the makespan minimization with positive tails on a single machine with a fixed non-availability interval. Journal of Combinatorial Optimization, 17(2), 117-133.
[14] Qi, X. (2007). A note on worst-case performance of heuristics for maintenance scheduling problems. Discrete Applied Mathematics, 155(3), 416-422.
[15] Lee, C. Y. (1996). Machine scheduling with an availability constraint. Journal of global optimization, 9(3-4), 395-416.
[16] Yuan, J., Shi, L., Ou, J. (2008). Single machine scheduling with forbidden intervals and job delivery times. Asia-Pacific Journal of Operational Research, 25(03), 317-325..
[17] Smith, W. E. (1956). Various optimizers for single-stage production. Naval Research Logistics Quarterly, 3(1-2), 59-66.
[18] Lawler, E. L., Lenstra, J. K., Kan, A. H. R., & Shmoys, D. B. (1993). Sequencing and scheduling: Algorithms and complexity. Handbooks in operations research and management science, 4, 445-522.
[19] Garey, M. R. (1979). Computers and intractability: A guide to the theory of np-completeness. Revista Da Escola De Enfermagem Da USP, 44(2), 340.
[20] Phillips, C., Stein, C., Wein, J. (1998). Minimizing average completion time in the presence of release dates. Mathematical Programming, 82(1-2), 199-223.
[21] Baker, K. R. (1974). Introduction to sequencing and scheduling. John Wiley & Sons.
[22] Schulz, A. S. (1996, June). Scheduling to minimize total weighted completion time: Performance guarantees of LP-based heuristics and lower bounds. In International conference on integer programming and combinatorial optimization (pp. 301-315). Springer, Berlin, Heidelberg.
[23] Hall, L. A., Schulz, A. S., Shmoys, D. B., & Wein, J. Scheduling to minimize average completion time: O-line and on-line algorithm. Joint journal version of 12] and 18.
[24] Adiri, I., Bruno, J., Frostig, E., & Kan, A. R. (1989). Single machine flow-time scheduling with a single breakdown. Acta Informatica, 26(7), 679-696.
[25] Qi, X., Chen, T., & Tu, F. (1999). Scheduling the maintenance on a single machine. Journal of the operational Research Society, 50(10), 1071-1078.
[26] Kacem, I., Chu, C. (2008). Worst-case analysis of the WSPT and MWSPT rules for single machine scheduling with one planned setup period. European Journal of Operational Research, 3(187), 1080-1089.
[27] Kellerer, H., Strusevich, V. A. (2010). Fully polynomial approximation schemes for a symmetric quadratic knapsack problem and its scheduling applications. Algorithmica, 57(4), 769-795
[28] Wang, G., Sun, H., & Chu, C. (2005). Preemptive scheduling with availability constraints to minimize total weighted completion times. Annals of Operations Research, 133(1-4), 183-192.
[29] Kacem, I. (2008). Approximation algorithm for the weighted flow-time minimization on a single machine with a fixed non-availability interval. Computers & Industrial Engineering, 54(3), 401-410.
[30] Kacem, I., Mahjoub, A. R. (2009). Fully polynomial time approximation scheme for the weighted flow-time minimization on a single machine with a fixed non-availability interval. Computers & Industrial Engineering, 56(4), 1708-1712.
[31] Kacem, I., Lanuel, Y., Sahnoune, M. (2011). Strongly Fully Polynomial Time Approximation Scheme for the two-parallel capacitated machines scheduling problem. International Journal of Planning and Scheduling, 1(1-2), 32-41.
[32] Kacem, I., Sahnoune, M., Schmidt, G. (2017). Strongly Fully Polynomial Time Approximation Scheme for the weighted completion time minimization problem on two-parallel capacitated machines. RAIRO-Operations Research, 51(4), 1177-1188
[33] Kacem, I., Paschos, V. T. (2013). Weighted completion time minimization on a single-machine with a fixed non-availability interval: Differential approximability. Discrete Optimization, 10(1), 61-68.
[34] Dessouky, M. I., Margenthaler, C. R. (1972). The one-machine sequencing problem with early starts and due dates. AIIE Transactions, 4(3), 214-222.

[DOI: 10.52547/jamlu.19.4.17]

[35] Carlier, J. (1982). The one-machine sequencing problem. European Journal of Operational Research, 11(1), 42-47.
[36] Lee, C. Y. (1996). Machine scheduling with an availability constraint. Journal of Global Optimization, 9(3-4), 395-416.
[37] Kacem, I., Haouari, M. (2009). Approximation algorithms for single machine scheduling with one unavailability period. 4OR, 7(1), 79.
[38] Mauguière, P., Billaut, J. C., & Bouquard, J. L. (2005). New single machine and job-shop scheduling problems with availability constraints. Journal of Scheduling, 8(3), 211-231.
[39] Hall, L. A., Shmoys, D. B. (1992). Jackson's rule for single-machine scheduling: making a good heuristic better. Mathematics of Operations Research, 17(1), 22-35.
[40] Kacem, I., Kellerer, H. (2018). Approximation schemes for minimizing the maximum lateness on a single machine with release times under non-availability or deadline constraints. Algorithmica, 80(12), 3825-3843.
[41] Kacem, I., Nagih, A., Seifaddini, M. (2013). Differential approximation analysis of Jackson's rule for single-machine scheduling problem with a fixed non-availability interval. In 2013 10th IEEE International Conference on Networking, Sensing, and Control (ICNSC) (pp. 388-391). IEEE.
[42] Kacem, I., Chu, C., Souissi, A. (2008). Single-machine scheduling with an availability constraint to minimize the weighted sum of the completion times. Computers & operations research, 35(3), 827-844.

[DOI: 10.52547/jamlu.19.4.17]