



ارائه یک الگوریتم ترکیبی برای زمان بندی وظایف جریان کاری علمی روی بستر ابر محاسباتی با محدودیت مهلت زمانی و هدف کاهش هزینه اجرا

ملیحه حریری^{۱*}، مصطفی نوری بایگی^۲ و سعید ابریشمی^۳

^۱ دانشجوی کارشناسی ارشد مهندسی کامپیوتر دانشگاه فردوسی مشهد، malihe.hariri@mail.um.ac.ir

^۲ استادیار دانشگاه فردوسی مشهد، nouribaygi@um.ac.ir

^۳ استادیار دانشگاه فردوسی مشهد، s-abrishami@um.ac.ir

چکیده: جریان‌های کاری علمی^۱ برای پردازش داده‌های عظیم و تحلیل‌ها و شبیه‌سازی‌های پیچیده استفاده می‌شوند، در نتیجه نیازمند منابع محاسباتی قدرتمندی هستند که بتوانند نتایج موردنظر را در زمان قابل قبول و با هزینه مطلوب تولید کنند. به این منظور از منابع زیرساخت‌های توزیع‌شده‌ای چون ابر محاسباتی به دلیل مزایایی از جمله دسترسی به منابع مجازی^۲، نامحدود^۳ و کشسان^۴ برای اجرای جریان‌های کاری، استفاده می‌شود و وظایف جریان‌های کاری برای اجرا روی منابع محاسباتی نگاشت می‌شود. برای نگاشت وظایف به منابع محاسباتی، مسئله به صورت مسئله زمان‌بندی مدل می‌شود؛ اما مسئله زمان‌بندی با وجود محدودیت‌ها و معیارهای مختلف یک مسئله *NP-hard* است. به همین جهت الگوریتم‌های مختلفی برای حل مسئله زمان‌بندی در زمان چندجمله‌ای ارائه شده‌است. الگوریتم ارائه‌شده در این پژوهش یک الگوریتم ترکیبی مبتنی بر مدل ریاضی است که مسئله زمان‌بندی را با شکستن مسئله به زیرمسئله‌های کوچک‌تر و زمان‌بندی بسته‌هایی از وظایف و مدل کردن مسئله با استفاده از یک مدل ریاضی خطی - صحیح انجام می‌دهد. مزیت این روش کاهش هزینه اجرای وظایف در یک مهلت زمانی مشخص نسبت به الگوریتم‌های زمان‌بندی ارائه‌شده است.

کلید واژه‌ها: زمان‌بندی وظایف، الگوریتم ترکیبی، مدل ریاضی، مهلت زمانی، هزینه

اهداف و معیارهای مختلفی در نظر گرفته می‌شود که در اغلب مسائل زمان و هزینه به عنوان دو معیار مهم در نظر گرفته شده‌اند. انتخاب منابع سریع‌تر هزینه زیادی را بر کاربر تحمیل می‌کند و استفاده از منابع کندتر با وجود اینکه هزینه را کاهش می‌دهد ولی قادر به اجرای وظایف در زمان ارائه شده نیست، بنابراین باید بین این دو توازن برقرار شود. الگوریتم ارائه شده در این مقاله به چالش‌های ذکر شده به خوبی پاسخ می‌دهد. نوآوری‌های ارائه شده در این الگوریتم را می‌توان به صورت زیر بیان کرد:

۱. ارائه یک روش مناسب برای توزیع مهلت زمانی با استفاده از مدل خطی برای فراهم‌آوری منابع مناسب
۲. ارائه یک الگوریتم زمان‌بندی با مدل خطی-صحیح برای زمان‌بندی با کیفیت بیش‌تر روی منابع

۱- مقدمه

جریان‌های کاری مجموعه‌ای از وظایف محاسباتی هستند که بین وظایف آن‌ها وابستگی‌های داده‌ای وجود دارد. پردازش و اجرای جریان‌های کاری در سال‌های اخیر روی بستر ابر محاسباتی مورد توجه محققان زیادی قرار گرفته است. برای زمان‌بندی جریان‌های کاری الگوریتم‌های مختلفی ارائه شده است که زمان‌بندی وظایف را روی منابع ابر محاسباتی به گونه‌ای انجام می‌دهند که نیازهای عملیاتی و غیرعملیاتی مطرح شده در مسأله برآورده شود.

در مسائل زمان‌بندی یک مسأله مهم انتخاب نوع و تعداد منابع مناسب برای زمان‌بندی وظایف است. انتخاب منابع مناسب تأثیر زیادی در عملکرد الگوریتم زمان‌بندی دارد. در مسائل زمان‌بندی

^۱ Workflow

^۲ Virtualized

^۳ Infinite

^۴ Elastic

فراهم شده

کوله‌پشتی فراهم می‌شود و زمان‌بندی وظایف روی منابع فراهم شده انجام می‌شود.

۲- کارهای پیشین

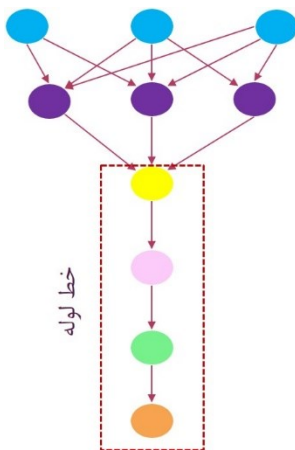
در [9] Rodriguez و همکاران یک الگوریتم ترکیبی زمان‌بندی و فراهم‌آوری منابع ارائه داده‌اند، با این هدف که زمان اجرای جریان کاری کمینه شود و زمان‌بندی با بودجه موردنظر به پایان برسد. در مرحله ایستا بودجه روی وظایف جریان کاری توزیع می‌شود و منابع موردنیاز با استفاده از یک مدل ریاضی فراهم می‌شوند و زمان‌بندی وظایف آماده روی منابع فراهم شده انجام می‌شود.

برای مسأله زمان‌بندی دسته‌بندی‌های مختلفی ارائه شده است که در [1]، [2]، [3]، [4] و [5] مسأله زمان‌بندی از منظرهای مختلف مورد بررسی قرار گرفته است. در [1] مسأله زمان‌بندی را از دیدگاه‌های مختلفی مورد بررسی قرار داده است. از نظر معیارهای زمان‌بندی می‌توان مقالات را به چندین بخش تقسیم کرد. اغلب الگوریتم‌های زمان‌بندی، دو فاکتور زمان اجرا و هزینه اجرای جریان کاری روی منابع محاسباتی را در نظر می‌گیرند و تلاش می‌کنند با در نظر گرفتن یک معیار به‌عنوان محدودیت مسأله زمان‌بندی، معیار دیگر را بهینه کنند؛ اما علاوه بر این دو معیار، معیارهای دیگری از جمله قابلیت اعتماد، مصرف انرژی، امنیت، بهره‌وری منابع و بارکاری نیز در نظر گرفته می‌شود.

۳- سیستم پیشنهادی

۳-۱- پیش‌نیازها

خط لوله یک ساختار توپولوژیکی رایج در جریان‌های کاری علمی است و شامل گروهی از وظایف است که بین آن‌ها یک ارتباط یک‌به‌یک وجود دارد. به‌صورت رسمی‌تر یک خط لوله به صورت $T_p = \{t_1, t_2, \dots, t_n\}$ و $n \geq 2$ تعریف می‌شود که بین دو وظیفه t_i و t_{i+1} یک یال وجود دارد. اولین وظیفه در خط لوله ممکن است بیش‌تر از یک پدر داشته باشد ولی حتماً باید یک فرزند داشته باشد. بقیه وظایف خط لوله باید یک پدر و یک فرزند داشته باشند. در شکل ۱ خط لوله یک جریان کاری نمونه نشان داده شده است.



شکل ۱ خط لوله یک جریان کاری نمونه

۳-۲- مدل سامانه و فرضیات مسأله

جریان کاری به‌صورت یک گراف جهت‌دار بدون دور مدل می‌شود که $T = \{t_1, t_2, \dots, t_n\}$ شامل مجموعه وظایف جریان کاری و

از نظر نگاشت وظایف جریان‌های کاری به منابع ابر محاسباتی، می‌توان کارهای انجام‌شده را به سه صورت ایستا، پویا و ترکیبی تقسیم‌بندی کرد. در این کارها با توجه به سیاست مسأله، نگاشت وظایف به منابع به سه روش متفاوت صورت می‌گیرد.

در [6] Moa و همکاران یک الگوریتم پویا برای زمان‌بندی وظایف جریان‌های کاری روی بستر ابر عمومی با محدودیت مهلت زمانی و هدف کاهش هزینه اجرا ارائه داده‌اند. در این کار برای بسته‌های وظایف آماده منابعی فراهم می‌شود که اجرای آن‌ها را قبل از مهلت زمانی و با کمترین هزینه به پایان برسانند.

در [7] Abrishami و همکاران دو الگوریتم برای زمان‌بندی جریان‌های کاری در بستر ابر محاسباتی با هدف کاهش هزینه اجرا و محدودیت مهلت زمانی ارائه داده‌اند. الگوریتم IC-PCP مسیر بحرانی جریان کاری را روی ارزان‌ترین منبعی که قادر باشد همه وظایف را قبل از دیرترین زمان اتمام آن به پایان برساند، زمان‌بندی می‌کند. در الگوریتم IC-PCPD2 هر وظیفه روی ارزان‌ترین منبعی که قادر باشد قبل از مهلت زمانی آن به پایان برساند، زمان‌بندی می‌شود.

در [8] Rodriguez و همکاران یک الگوریتم ابتکاری با زمان‌بندی ترکیبی برای جریان‌های کاری علمی در زیرساخت ابر عمومی ارائه داده‌اند. در این کار در مرحله ایستا مهلت زمانی روی وظایف جریان کاری توزیع می‌شود و در مرحله پویا منابع موردنیاز برای زمان‌بندی وظایف آماده با استفاده از الگوریتم



الگوریتم ۱: پیدا کردن خط لوله‌ها

```

1. procedure findPipeline(Task t, Pipeline p, List<Task> ischeck){
2. if ( t.getChildList().size() > 1 OR t.getChildList().size() == 0
   OR t.getChildList().get(0).getParentList().size() > 1) {
3.   if (!pipeline.isEmpty())
4.     t.putInPipeline(pipeline);
5.   return pipeline;
6. }
7. t.putInPipeline(pipeline);
8. return findPipeline(t.getChildList().get(0), pipeline, ischeck);

```

خط لوله‌های جریان کاری در الگوریتم ۱ نشان داده شده است.

الگوریتم ۱ پیدا کردن خط لوله‌های جریان کاری

تشکیل خط لوله‌ها به این صورت است که ابتدا وظایف به ترتیب توپولوژیکی مرتب می‌شوند، سپس خط لوله‌ها ساخته می‌شوند. برای هر وظیفه مرتبی که پردازش نشده است الگوریتم تلاش می‌کند خط لوله‌ای بسازد که با این وظیفه شروع شود. بهترین حالت بازگشت وقتی اتفاق می‌افتد که وظیفه هیچ فرزندی نداشته باشد یا بیش از یک فرزند داشته باشد و یا چندین پدر داشته باشد. حالت بازگشتی زمانی اتفاق می‌افتد که وظیفه فقط یک فرزند و فقط یک پدر داشته باشد. در این حالت وظیفه به خط لوله اضافه می‌شود و بازگشت با فرزند آن ادامه پیدا می‌کند. وقتی خط لوله شناسایی شد و بازگشت به پایان رسید، این کار برای وظیفه پردازش نشده بعدی تکرار می‌شود تا زمانی که هیچ وظیفه پردازش نشده‌ای باقی نماند.

در مرحله بعد به هر وظیفه در جریان کاری یک مهلت زمانی تخصیص داده می‌شود. مهلت زمانی تخصیص داده شده به وظایف در الگوریتم زمان بندی و هزینه اجرا تأثیر زیادی دارد. به این منظور روش توزیع مهلت زمانی ارائه شده در مقاله [8] را با استفاده از مدل ریاضی خطی بهبود بخشیدیم.

الگوریتم توزیع مهلت زمانی وظایف به صورت زیر عمل می‌کند: برای توزیع مهلت زمانی وظایف، ابتدا زودترین زمانی که وظایف می‌توانند به پایان برسند، روی سریع‌ترین ماشین موجود مطابق $eft_t = \max_{p \in \text{parents}}\{eft_p\} + Runtime_t^{fastestVm}$

محاسبه می‌شود. سپس مقدار بیشینه زودترین زمان پایان بین تمام وظایف مطابق $eft_{max} = \max_{t \in W}\{eft_t\}$ محاسبه می‌شود و میزان زمان اضافی از مهلت زمانی جریان کاری

E مجموعه یال‌های این گراف است. یک یال $e_{ij} = (t_i, t_{i+1})$ بین دو وظیفه t_i و t_{i+1} جود دارد، اگر بین این دو وظیفه وابستگی داده‌ای وجود داشته باشد. در نتیجه وظیفه فرزند تا قبل از اجرای کامل وظایف پدرانش نمی‌تواند شروع به اجرا کند.

فراهم‌کننده منابع، انواع مختلفی از منابع $VMT = \{vmt_1, vmt_2, \dots, vmt_n\}$ با هزینه‌های اجاره متفاوت و پیکربندی‌های مختلف را پیشنهاد می‌کند. در این مقاله منابع فراهم شده از فراهم‌کننده گوگل با دوره‌های زمانی یک دقیقه‌ای استفاده شده است و به ازای استفاده جزئی بیش از دوره زمانی، هزینه این زمان، به هزینه کلی اضافه می‌شود.

ظرفیت ذخیره‌ساز فرض شده در این مسأله بی‌نهایت در نظر گرفته شده است. داده‌های ورودی و خروجی هر یک از وظایف روی ذخیره‌ساز ذخیره می‌شوند و زمان انتقال داده از ذخیره‌ساز، در زمان اجرای وظیفه و زمان اجرای منبع در نظر گرفته شده است. همچنین از هزینه مالی استفاده از ذخیره‌ساز صرف نظر می‌شود. زمان فراهم‌آوری منابع در قالب هزینه در الگوریتم در نظر گرفته شده است. به این صورت که الگوریتم به نسبت افزایش سرعت منبع انتخابی به ارزان‌ترین منبع، هزینه مازاد باید پرداخت کند. همچنین منابع زمانی که به ۵ ثانیه آخر دوره زمانی خود می‌رسند در صورتی که هیچ وظیفه‌ای برای اجرا وجود نداشته باشد، خاموش می‌شوند.

۳-۳- روش پیشنهادی

در این مقاله، یک الگوریتم ترکیبی برای زمان بندی وظایف جریان کاری علمی، ارائه شده است که در واقع بهبودی روی روش ارائه شده در [8] است. روش پیشنهادی این مقاله از دو مرحله ایستا و پویا تشکیل شده است. در مرحله ایستا جریان کاری پیش پردازش می‌شود و ساختار خط لوله‌های جریان کاری مشخص می‌شود و برای هر وظیفه با استفاده از یک روش بهبود یافته با استفاده از مدل ریاضی خطی، مهلت زمانی وظایف محاسبه می‌شود. در مرحله پویا زمان بندی بسته‌های وظایف آماده (خط لوله‌های آماده) با استفاده از یک روش فراهم‌آوری منابع با مدل برنامه ریزی خطی - صحیح انجام خواهد شد. در ادامه به تشریح هر دو مرحله پرداخته شده است.

۱-۱-۱- مرحله ایستا

در مرحله ایستا ابتدا ساختار خط لوله‌های جریان کاری مطابق تعریف ارائه شده در ۳-۱- مشخص می‌شود. نحوه تشکیل



می‌شود. برای زمان‌بندی وظایف داخل صف اجرا، ابتدا وظایف به بسته‌هایی از وظایف آماده یا بسته‌هایی از خط لوله‌های آماده، دسته‌بندی می‌شوند. اگر منابعی از قبل وجود داشته باشند که بخشی از دوره زمانی آن‌ها استفاده شده‌است و هم‌اکنون بی‌کار هستند، به‌اندازه تعداد وظایفی از بسته که منبع می‌تواند در مهلت زمانی مشخص شده و قبل از اتمام دوره زمانی‌اش اجرا کند، برای زمان‌بندی به منبع تخصیص داده می‌شود و برای مابقی وظایف باقی‌مانده در بسته، با استفاده از یک طرح فراهم‌آوری، منابع موردنیاز برای اجرای آن‌ها تعیین می‌شود.

برای فراهم‌کردن منابع موردنیاز برای وظایف باقی‌مانده از یک مدل ریاضی خطی-صحیح استفاده شده است که نوع و تعداد منابع موردنیاز برای اجرای بسته‌های وظایف (بسته‌های خط لوله) را به‌گونه‌ای تعیین می‌کند، که هزینه اجرای این بسته از وظایف کمینه باشد و قبل از مهلت زمانی وظایف به پایان برسد.
جدول ۳: ویژگی‌های هر منبع

C_{vmt}	هزینه منبع در هر دوره پرداختی
p_t^{vmt}	زمان پردازش وظیفه‌ای با بزرگ‌ترین زمان پردازش روی منبع با نوع vmt

جدول ۴: متغیرهای مسأله

C_{bot}	هزینه اجرای یک بسته از وظایف
$Cost_{vm}$	هزینه استفاده از منبع برحسب تعداد دوره‌های زمانی مصرف‌شده
$N_{vmt,k}$	متغیر صحیح نشان‌دهنده تعداد وظایف داده‌شده به k امین نمونه نوع vmt
$U_{vmt,k}$	مدت‌زمانی که k امین نمونه از نوع vmt برای اجرای وظایف استفاده کرده است
$P_{vmt,k}$	متغیر صحیح نشان‌دهنده تعداد دوره‌های زمانی اجاره‌شده برای k امین نمونه نوع vmt
$L_{vmt,k}$	متغیر باینری که در صورتی که k امین نمونه از نوع vmt اجاره‌شده باشد مقدار ۱ می‌گیرد. در غیر این صورت مقدار ۰ دارد
n	تعداد وظایف موجود در بسته
δ_{bot}	مهلت زمانی تخصیص داده‌شده به بسته وظایف (بسته خط لوله‌ها)
IntTol	معیار توقف حل‌کننده

$sparetime = \delta_t - eft_{max}$ به‌دست می‌آید. حال با استفاده از یک مدل ریاضی، زمان اضافی به وظایف جریان کاری با استفاده از فرمول‌های ۱ تا ۴ تخصیص داده می‌شود. در ادامه متغیرها و پارامترهای مدل ریاضی در جدول ۱ و جدول ۲ آورده شده‌است.
جدول ۱ متغیرهای مدل ریاضی

$spare_i$	میزان زمان تخصیص داده شده به هر سطح از جریان کاری
T	متغیر تعریف شده برای تبدیل مدل به خطی

جدول ۲ پارامترهای ورودی

$\max t_{il}$	بیشترین زمان اجرا بین وظایف یک سطح از جریان کاری
$\sum t_{il}$	مجموع زمان اجرای همه وظایف جریان کاری
$SpareTime_w$	زمان اضافی باقی‌مانده از جریان کاری

- ۱ $Max T$
- ۲ $\frac{\max t_i + spare_i}{\sum t_{il}} \geq T \quad \forall l \in level$
- ۳ $\sum spare_i \leq sparetime_w \quad \forall l \in level$
- ۴ $spare_i \geq 0$

در این مدل، تابع هدف بیشینه کردن مقدار $\frac{\max t_i + spare_i}{\sum t_{il}}$ به ازای هر سطح از جریان کاری است. با این محدودیت که میزان زمان اضافی داده شده به هر سطح گراف کمتر از میزان زمان اضافی کل جریان کاری باشد. بعد از حل مدل، مقدار $spare_i$ برای هر سطح از جریان کاری به‌دست می‌آید و به تمام وظایف هر سطح از گراف جریان کاری تخصیص داده می‌شود. سپس مهلت زمانی δ_t داده‌شده به هریک از وظایف جریان کاری به صورت فرمول ۵ محاسبه می‌شود.

$$\delta_t = \max_{p \in t.parents} \{eft_p\} + sparetime_t + runtime_t^{fastestVm} \quad 5$$

۱-۲-۱- مرحله پویا

در این مرحله، زمان‌بندی وظایف آماده در صف اجرا آغاز



فراهم‌آوری منابع به‌صورت RP_{vmt} تعیین می‌کند از هر نوع منبع VMT چه تعداد باید فراهم شود و هر منبع قادر به زمان‌بندی چند وظیفه است. در الگوریتم شبه کد الگوریتم ترکیبی پیشنهادی ارائه شده است.

```

1. for(all bot in BOT)
2.   if(bot.size() > 1)
3.     if(idleVmList.isEpmty())
4.       MilpResult = Milp.solveMilp(bot, deadlineBot);
5.       for(all milpresult =(vmType,numVm,NTask)
6.         MyTask t = getTaskFromBot();
7.         submitTaskVm(t, vm);
8.       end for
9.     end if
10.    else
11.      MyVm vm = idleVmList.get();
12.      if(vm.NTask != 0)
13.        submitTaskVm(t, vm);
14.      end if
15.    end else
16.    end if
17.  else
18.    if(idleVmList.isEpmty())
19.      MyVM vm = provisionsingleBot(bot);
20.      MyTask t = getTaskFromBot();
21.      submitTaskVm(t, vm);
22.    end if
23.  else
24.    MyVm vm = idleVmList.get();
25.    if(vm.NTask!=0)
26.      submitTaskVm(t, vm);
27.    end if
28.  end else
29. end for

```

الگوریتم ۲ زمان‌بندی وظایف جریان کاری

۴- ارزیابی کارایی

در شبیه‌سازی‌های صورت گرفته شده از جریان‌های کاری علمی Montage و Epigenomics و Sipht و Ligo استفاده شده‌است و جریان‌های کاری با تعداد تقریباً ۱۰۰۰ وظیفه در نظر گرفته شده‌است، تخمین اندازه وظایف ۱۰٪ دقیق نیست و در محاسبه اندازه وظایف تغییرات حدود ۱۰٪ با توزیع یکنواخت برای نرمال‌سازی زمان اجرای وظایف لحاظ می‌گردد.

در این مقاله برای هر جریان کاری ۷ مهلت زمانی در نظر گرفته شده‌است. سخت‌ترین مهلت زمانی δw_1 برابر مجموع زمان اجرای وظایف روی مسیر بحرانی و زمانی است که برای خواندن و نوشتن از ذخیره‌ساز نیاز است. مابقی مهلت‌های زمانی به صورت فرمول‌های ۱۴ تا ۲۰ محاسبه می‌شود.

τ	دوره زمانی هر منبع محاسباتی (ثانیه)
C_{vmt}	هزینه اجرای یک دوره زمانی منبع نوع vmt

مدل ریاضی ارائه شده، نوع و تعداد منابع برای اجرای بسته‌های وظایف را به گونه‌ای انتخاب می‌کند که هزینه اجرای این بسته از وظایف روی منابع موجود کمینه شود و اجرای وظایف قبل از مهلت زمانی داده شده به پایان برسد. مدت زمانی که k امین نمونه از نوع vmt مشغول اجرا بوده، به صورت فرمول $U_{vmt,k} = p_t^{vmt} * N_{vmt,k}$ محاسبه می‌شود. هزینه هر نوع منبع استفاده شده بر اساس تعداد دوره‌های زمانی اجرا شده توسط منبع و هزینه ثابت منبع به ازای یک دوره زمانی به صورت فرمول $Cost_{vm} = p_{vmt,k} * C_{vmt}$ محاسبه می‌شود. بنابراین مسأله را می‌توان به صورت زیر فرموله کرد:

$$Min C_{bot} \quad 6$$

$$C_{bot} - cost_{vm} \geq 0 \quad 7$$

$$\sum_{j \in VMT} \sum_{k \in vm_j} N_{vmt,k} = n \quad 8$$

$$N_{vmt,k} \leq n * l_{vmt,k} \quad 9$$

$$\frac{U_{vmt,k}}{\tau} \leq P_{vmt,k} \quad 10$$

$$N_{vmt,k} \geq L_{vmt,k} \quad 11$$

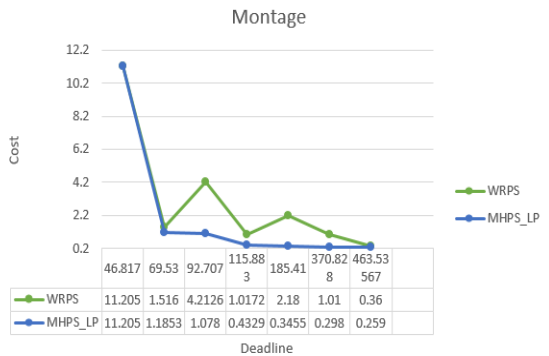
$$\frac{U_{vmt,k}}{\tau} + (1 - IntTol) \geq P_{vmt,k} \quad 12$$

$$U_{vmt,k} \leq \delta_{bot} \quad 13$$

تابع هدف مسأله، کمینه کردن هزینه اجرای بسته وظایف است که در فرمول ۶ محاسبه شده است. محدودیت ۷ نشان می‌دهد که هزینه اجرای این بسته از وظایف باید نسبت به هزینه‌ی اجرای منبع در دوره‌های زمانی کمینه شود. محدودیت ۸ تضمین می‌کند، همه وظایف بسته روی منابع پردازش می‌شوند محدودیت ۹ و ۱۱ تضمین می‌کند به هر منبع فراهم شده، حتماً وظیفه‌ای برای اجرا داده خواهد شد. محدودیت ۱۰ و ۱۲ تعداد دوره‌های زمانی استفاده شده منبع را با محاسبه مدت زمان اجرای منبع و با گرد کردن به نزدیک‌ترین عدد صحیح بزرگ‌تر به دست می‌آورد. محدودیت ۱۳ تضمین می‌کند، مدت زمان اجرای وظیفه روی منبع زمان‌بندی شده باید کمتر از مهلت زمانی داده شده باشد. بعد از حل مسأله متغیر $N_{vmt,k}$ تبدیل به یک طرح

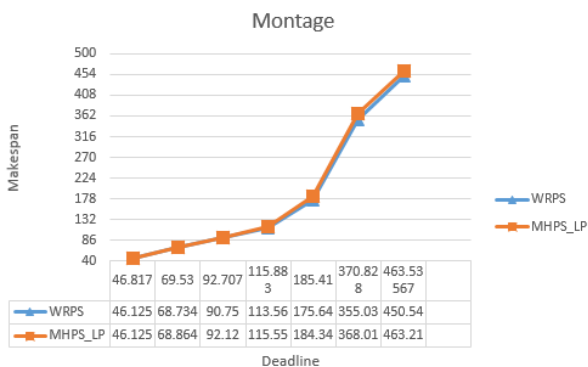


در این جریان کاری هزینه و زمان اجرا روی الگوریتم پیشنهادی و الگوریتم مورد مقایسه در شکل ۲ و شکل ۳ قابل مشاهده است. در الگوریتم WRPS با افزایش مهلت زمانی در برخی موارد هزینه افزایش پیدا کرده است که به دلیل توزیع نامناسب مهلت زمانی با روش pspare است، که روند کاهش هزینه اجرای جریان های کاری در الگوریتم پیشنهادی مقاله قابل مشاهده است.



شکل ۲: هزینه اجرای جریان کاری Montage روی دو الگوریتم WRPS و MHPSLP

در نمودار زمان اجرای الگوریتم ها روی جریان کاری Montage نشان داده شده است. در الگوریتم MHPSLP، برای کم کردن هزینه بهترین استفاده از زمان موجود را نسبت به الگوریتم مورد مقایسه داشته اند. این بهبود در مهلت های زمانی بالاتر در الگوریتم پیشنهادی ما قابل مشاهده است.



شکل ۳: زمان اجرای جریان کاری Montage روی دو الگوریتم WRPS و MHPSLP

• جریان کاری Epigenomics

$$\delta w_{int} = \delta w_1 / 2 \quad 14$$

$$\delta w_2 = \delta w_1 + \delta w_{int} \quad 15$$

$$\delta w_3 = \delta w_2 + \delta w_{int} \quad 16$$

$$\delta w_4 = \delta w_3 + \delta w_{int} \quad 17$$

$$\delta w_5 = 4 * \delta w_1 \quad 18$$

$$\delta w_7 = 8 * \delta w_1 \quad 19$$

$$\delta w_7 = 10 * \delta w_1 \quad 20$$

تابع در نظر گرفته شده برای اجرای جریان های کاری از منابع فراهم کننده گوگل با دوره زمانی ۶۰ ثانیه فرض شده است و ۴ نوع از این منابع در نظر گرفته شده است که در جدول قابل مشاهده است. همچنین ذخیره ساز با بیشترین سرعت خواندن و نوشتن فرض شده است.

نام	حافظه	Mips	قیمت بر هزینه
N1-standard-1	3.75 GB	2.75*2750	\$0.00105000
N1-standard-2	7.5 GB	5.5*2750	\$0.00210525
N1-standard-4	15 GB	11*2750	\$0.004221026
N1-standard-8	30 GB	22*2750	\$0.00846315

روش پیشنهادی این مقاله الگوریتم MHPSLP است که با کار ارائه شده در [8] از نظر زمان و هزینه مورد ارزیابی قرار گرفت است.

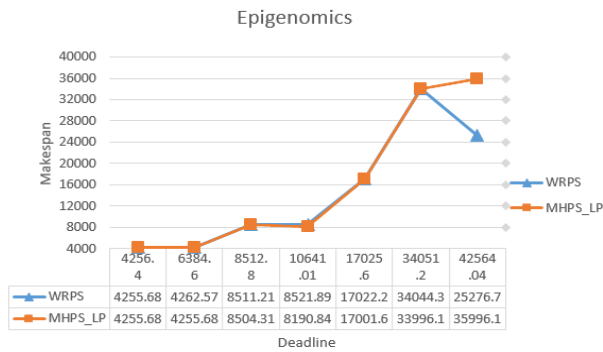
۴-۱- نتایج آزمایش ها و تحلیل ها

تأثیر مهلت زمانی روی هزینه و زمان اجرای ۴ جریان کاری در نمودارهای زیر بررسی شده است. در تمام نمودارهای ارزیابی عملکرد الگوریتم با افزایش مهلت زمانی از سخت ترین تا سست ترین نشان داده شده است.

• جریان کاری Montage



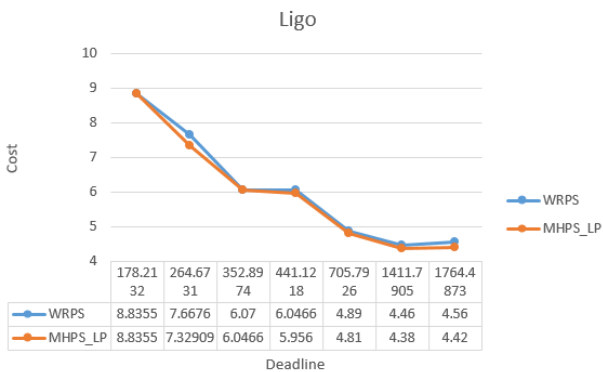
همین شکل دیده می شود.



شکل ۵: زمان اجرای جریان کاری Epigenomics روی دو الگوریتم WRPS و MHPSLP

• جریان کاری Ligo

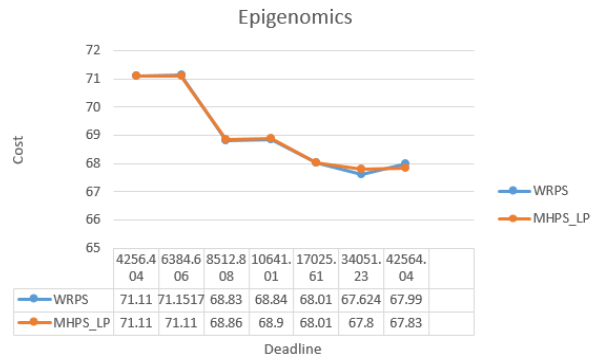
در این جریان کاری هزینه اجرا و زمان اجرای جریان کاری در نمودارهای آمده است. در این جریان کاری هزینه اجرا در الگوریتم MHPSLP کاهش یافته است، در نتیجه منابع کمتری فراهم می شود و منابع می توانند چندین وظیفه را زمان بندی و به ترتیب اجرا کنند. در صورتی که تعداد منابع فراهم شده در الگوریتم WRPS بیش تر است و به علت مهلت زمانی کمی که به وظایف با زمان اجرای بالا داده است مجبور می شود از منابع سریع تر برای اجرا استفاده کند، در نتیجه هزینه بالاتر می رود.



شکل ۶: هزینه اجرای جریان کاری Ligo روی دو الگوریتم WRPS و MHPSLP

نتایج ارزیابی زمان اجرای الگوریتم ها روی جریان کاری Ligo نشان می دهد عملکرد الگوریتم MHPSLP در استفاده از زمان باقی مانده بهتر از الگوریتم های دیگر است. مخصوصاً در مهلت های زمانی بالاتر این تفاوت واضح تر می شود که نشان دهنده عملکرد بهتر الگوریتم در مهلت های زمانی بالاتر است.

هزینه و زمان اجرا روی الگوریتم پیشنهادی و الگوریتم مورد مقایسه در شکل ۴ و شکل ۵ نشان داده شده است. در الگوریتم WRPS در برخی مهلت های زمانی هزینه افزایش یافته است که به دلیل توزیع نادرست مقدار زمانی اضافی جریان کاری بین وظایف است. در این صورت منابع مطابق با نیاز فراهم نمی شوند و منجر به افزایش هزینه خواهد شد. در الگوریتم MHPSLP این مشکل برطرف شده است، اما در مهلت زمانی سوم تا پنجم هزینه نسبت به الگوریتم مورد مقایسه بیشتر شده است.



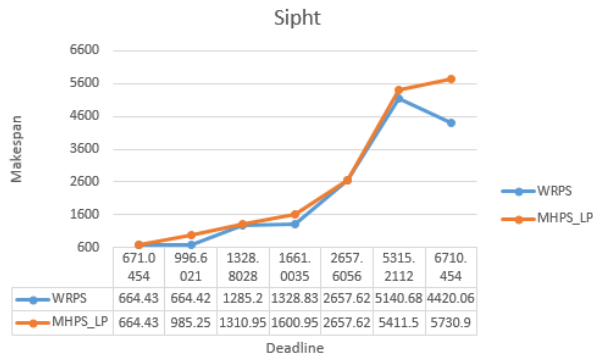
شکل ۴: هزینه اجرای جریان کاری Epigenomics روی دو الگوریتم WRPS و MHPSLP

زمان اجرای الگوریتم ها روی جریان کاری Epigenomics در شکل ۵ نشان داده شده است. در الگوریتم WRPS زمان اجرای جریان کاری در مهلت های زمانی سوم تا پنجم کاهش یافته است و این به دلیل ساختار این جریان کاری و فرضی است که برای فراهم آوری منابع بسته های خط لوله های وظایف در نظر گرفته شده است.

ما در این مسأله فرض کردیم که اندازه زمان اجرای همه خط لوله ها به اندازه خط لوله ای است که بیش ترین زمان اجرا را دارد، در نتیجه منابع می توانند تعداد کمتری از این خط لوله ها را زمان بندی کنند و تعداد منابع فراهم شده بیشتر می شود و از طرفی به دلیل زمان اجرای بالای این خط لوله ها الگوریتم منابع سریع تر را برای اجرای وظایف در مهلت زمانی تعیین شده انتخاب می کند و به همین دلیل اجرای وظایف را زودتر به پایان می رساند و الگوریتم تلاش می کند برای این وظایف منابع سریع تر فراهم کند تا اجرای آن ها را زودتر به پایان برساند و از طرفی طبق فرض، برای فراهم آوری منابع خط لوله ها نیز منابع سریع تر انتخاب می شود که این دو باعث می شود الگوریتم از زمان باقی مانده تا مهلت زمانی به خوبی استفاده نکند و زمان اجرا کاهش می یابد و از طرفی هزینه نیز افزایش پیدا می کند. این مشکلات در الگوریتم MHPSLP در همین مهلت های زمانی و به



وظایف با زمان اجرای بالا وجود دارد که الگوریتم WRPS توجهی به اهمیت آن نمی‌کند و ارزان‌ترین منابع را برای آن انتخاب می‌کند. در نتیجه منجر به افزایش زمان اجرای آن می‌شود. در الگوریتم MHPSLP بهبود زمان اجرا مشهود است.



شکل ۹: زمان اجرای جریان کاری Sipt روی دو الگوریتم WRPS و MHPSLP

۵- نتیجه‌گیری

در این مقاله یک الگوریتم ترکیبی بهبود یافته برای زمان بندی وظایف جریان های کاری در ابر محاسباتی ارائه شده است. در این الگوریتم هدف کمینه کردن هزینه اجرای وظایف روی منابع ابر محاسباتی است، به این شرط که اجرای همه وظایف جریان کاری در مهلت زمانی تعیین شده به پایان برسد. بنابراین مهلت زمانی تخصیص داده شده به وظایف تأثیر زیادی روی عملکرد الگوریتم زمان بندی خواهد داشت.

الگوریتم توزیع مهلت زمانی در [8] مهلت زمانی وظایف را به درستی توزیع نمی‌کند. از طرفی به دلیل در نظر گرفتن زمان فراهم‌آوری منابع، زمان زیادی را از دست می‌دهد. به همین دلیل الگوریتم زمان بندی WRPS قادر به اجرای موفق و کامل همه وظایف برخی از جریان‌های کاری مثل Sipt و Ligo در سخت‌ترین مهلت زمانی نیست. از طرفی این توزیع نامناسب مهلت زمانی باعث افزایش هزینه اجاره منابع در برخی موارد شده است که به صورت نمونه می‌توان به جریان کاری Sipt اشاره کرد که در بخش قبل شرح داده شد. در اغلب جریان‌های کاری، کار ارائه شده ما عملکرد بهتری نسبت به الگوریتم WRPS داشته است. جز در مواردی که بسته به ساختار جریان کاری تغییراتی خلاف انتظار رخ داده است، در کل می‌توان گفت مدل ریاضی منابع را هوشمندانه‌تر انتخاب می‌کند و توزیع درست مهلت زمانی روی وظایف جریان کاری الگوریتم را در انتخاب منابع درست هدایت



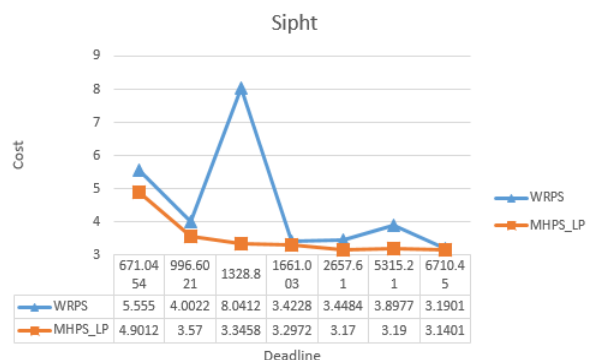
شکل ۷: زمان اجرای جریان کاری Ligo روی دو الگوریتم WRPS و MHPSLP

• جریان کاری Sipt

در این جریان کاری هزینه اجرا و زمان اجرای جریان کاری در شکل ۸ و شکل ۹ نشان داده شده است. در این جریان کاری هزینه اجرا اهمیت توزیع مهلت زمانی روی وظایف را در این جریان کاری به وضوح بیشتری نسبت به جریان‌های کاری بررسی شده نشان می‌دهد و آن هم به دلیل ساختار متفاوت این جریان کاری و تفاوت زمان اجرای بسته‌های وظایف است.

در الگوریتم WRPS مقدار هزینه با افزایش مهلت زمانی از مهلت زمانی دوم به مهلت زمانی سوم به میزان زیادی افزایش پیدا می‌کند. به این علت که این الگوریتم مهلت زمانی زیادی به وظایف با زمان اجرای کم داده می‌دهد. بنابراین الگوریتم برای اجرای وظایف بزرگ‌تر مجبور است منابع سریع‌تر و بیشتری فراهم کند که هزینه اجرا را خیلی بالاتر می‌برد.

در الگوریتم MHPSLP هزینه به ویژه در مهلت زمانی دوم به بعد بهبود بیشتری پیدا کرده است و حتی در سخت‌ترین مهلت زمانی نیز قادر به کاهش هزینه اجرا شده است و با افزایش مهلت زمانی این کاهش هزینه قابل مشاهده است.



شکل ۸: هزینه اجرای جریان کاری Sipt روی دو الگوریتم WRPS و MHPSLP

نتایج ارزیابی زمان اجرای الگوریتم‌ها روی جریان کاری Sipt در شکل ۹ نشان داده شده است. در این جریان کاری یک بسته از



- clouds,” *Futur. Gener. Comput. Syst.*, vol. 82, pp. 1–11, 2018.
- [6] M. Mao and M. Humphrey, “Auto-scaling to minimize cost and meet application deadlines in cloud workflows,” *Proc. 2011 Int. Conf. High Perform. Comput. Networking, Storage Anal. - SC '11*, p. 1, 2011.
- [7] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, “Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 158–169, 2013.
- [8] M. A. Rodriguez and R. Buyya, “A responsive knapsack-based algorithm for resource provisioning and scheduling of scientific workflows in clouds,” in *Proceedings of the International Conference on Parallel Processing*, 2015, vol. 2015-Decem, pp. 839–848.
- [9] M. A. Rodriguez and R. Buyya, “39 Budget-Driven Resource Provisioning and Scheduling of Scientific Workflow in IaaS Clouds with Fine-Grained Billing Periods,” vol. 9, no. 4, pp. 1–22, 2017.
- می‌کند. در آینده در نظر داریم مشکلات پیش‌رو را برطرف کنیم و
فرضیات را به‌گونه‌ای تغییر دهیم که این مشکلات برطرف شود.
- مراجع**
- [1] M. A. Rodriguez and R. Buyya, “A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments,” *Concurr. Comput.*, vol. 29, no. 8, pp. 1–32, 2017.
- [2] S. Smachat and K. Viriyapant, “Taxonomies of workflow scheduling problem and techniques in the cloud,” *Futur. Gener. Comput. Syst.*, vol. 52, pp. 1–12, 2015.
- [3] E. N. Alkhanak, S. P. Lee, and S. U. R. Khan, “Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities,” *Futur. Gener. Comput. Syst.*, vol. 50, pp. 3–21, 2015.
- [4] F. Wu, Q. Wu, and Y. Tan, “Workflow scheduling in cloud: a survey,” *J. Supercomput.*, vol. 71, no. 9, 2015.
- [5] L. Thai, B. Varghese, and A. Barker, “A survey and taxonomy of resource optimisation for executing bag-of-task applications on public