دانشکده علوم ریاضی

بهینه‌سازی ترکیبیاتی

زمان‌بندی پروژه با محدودیت منابع

استاد ارجمند

آقای دکتر رضا قنبری

نگارش

مرضیه مرتضوی نژاد

زمستان ۹۰

چکیده

مسئله زمانبندی پروژه با محدودیت منابع دارای ادبیات موضوع بسیار قوی است. این مسئله به دو دلیل مورد توجه قرار گرفته است: اول اینکه این مساله با توجه به شرایط متفاوت کاربردی و صنعتی از نظر تابع هدف، خصوصیات فعالیت‌ها، منابع و نوع ارتباط بین فعالیت‌ها بسیار متنوع‌اند. از جمله کاربردهای آن می‌توان در تولید، مدیریت پروژه، برنامه‌ریزی تولید و ... اشاره کرد. دوم اینکه با توجه به $NP-hard$ بودن این مسئله روش‌های دقیق برای حل نمونه‌های بزرگ راه حل مناسبی نیست، به همین دلیل محققین همواره به دنبال ارائه راه حل‌های کاراتری برای حل آن بوده‌اند. در این گزارش، مسئله زمانبندی پروژه با محدودیت منابع با هدف کاهش مدت زمان کل پروژه معرفی شده است. در ادامه برخی از روش‌های حل آن ارائه شده‌اند،که شامل روش‌های دقیق و رویکردهای ابتکاری می‌باشند.

فهرست مطالب

فصل ۱

تعریف مسئله

۱.۱ مقدمه

مسائل زمانبندی دارای دامنه‌ی گسترده‌ای می‌باشند و مسئله‌ی زمانبندی پروژه با محدودیت منابع[۱]، نوعی از مسائل زمانبندی است که خود دامنه‌ی گسترده‌ای از مسائل دیگر زمانبندی را در بر می‌گیرد. تخصیص منابع از وظایف اصلی مدیر پروژه می‌باشد. در شرایطی که سطح منابع قابل دسترس محدود باشد، مسئله‌ی تخصیص منابع مطرح می‌شود و لازم است فعالیت‌ها به صورتی زمانبندی شوند که در هیچ شرایطی سطح مورد نیاز به منابع از میزان قابل دسترس منابع بالاتر نبوده و در عین حال پروژه در کوتاهترین زمان ممکن به انجام برسد.

به طور کلی در مسائل زمانبندی پروژه دو نوع محدودیت می‌تواند وجود داشته باشد، یکی محدودیت تقدم و تأخر و دیگری محدودیت منابع، که هر یک از آنها به نوبه‌ی خود می‌توانند شکل‌های مختلفی داشته باشند.

---

[۱] *Recource Constrained Project Scheduling Problem* (*RCPSP*)

در برخی از تحقیقات انجام شده و مدل‌های به وجود آمده برای سادگی کار یکی از این دو محدودیت حذف و یا ساده شده است. در مسئله‌ی تخصیص بهینه‌ی منابع محدود در کنترل پروژه هر دو نوع محدودیت فوق باید در نظر گرفته شوند و این باعث می‌شود که این نوع مسئله‌ی زمانبندی نسبت به بقیه مشکلتر شود. مسئله‌ی برنامه‌ریزی پروژه به منابع محدود، برای نخستین بار توسط کلی[۱] و ویست[۲] در سال ۱۹۳۶ مطرح شد و بعد از آن تحقیقات وسیعی در جنبه‌های مختلف این مسئله انجام گردید و در طول سال‌های گذشته پیشرفت‌های قابل توجهی در طراحی روش‌های کارآمد حل این مسئله انجام شده است.[۱]

۲.۱ مدل مفهومی مسئله

حالت کلی مسئله‌ی تخصیص منابع به صورت زیر است: در هر پروژه تعدادی فعالیت وابسته به هم وجود دارند که برای به پایان رساندن پروژه باید به انجام برسند. بین فعالیت‌های پروژه روابط تقدم و تأخر برقرار است. این روابط با استفاده از عوامل کنترل‌کننده‌ی، پایان به شروع[۳]، شروع به شروع[۴]، پایان به پایان[۵]، شروع به پایان[۶] تعریف می‌شوند. روابط منطقی بین فعالیت‌های یک پروژه می‌تواند در قالب ترکیبی از این عوامل کنترل‌کننده بیان شود.

شروع یک فعالیت حداقل بعد از اتمام یک فعالیت دیگر: پایان به شروع

شروع یک فعالیت حداقل بعد از شروع یک فعالیت دیگر: شروع به شروع

پایان یک فعالیت حداقل بعد از اتمام یک فعالیت دیگر: پایان به پایان

پایان یک فعالیت حداقل بعد از شروع یک فعالیت دیگر: شروع به پایان

اجرای هر فعالیت پروژه معمولاً نیازمند منابع متفاوتی از جمله زمان، سرمایه، نیروی انسانی، ابزار، مواد، انرژی و ماشین‌آلات است که بعضاً محدودند. به طور کلی منابع به تمام عناصر مورد نیاز برای تکمیل یک پروژه اطلاق می‌شود. منابع مورد استفاده در پروژه‌ها را معمولاً به دو دسته تجدیدشدنی[۷] و تجدیدنشدنی[۸] تقسیم می‌کنند. منابع تجدیدشدنی مثل ماشین، ابزار و نیروی انسانی، منابعی هستند که صرفنظر از طول پروژه در هر یک از دوره‌ها یا واحدهای زمانی پروژه ثابت باشند و تنها محدودیت آنها مقدار استفاده از آنها در طی یک

---

[۱] *Kelly* [۲] *Wiest* [۳] *Finish to Start (FS)* [۴] *Start to Start (SS)* [۵] *Finish to Finish (FF)* [۶] *Start to Finish (SF)*

[۷] *Renewable resources* [۸] *Nonrenewable resources*

دوره باشد، یعنی بعد از اتمام فعالیتی که به این منابع نیاز دارد، آزاد می‌شوند و باری انجام فعالیت‌های بعدی می‌توانند مورد استفاده قرار گیرند. در مسئله زمانبندی پروژه با منابع محدود هر کدام از منابع تجدیدشدنی به عنوان وسیله یا ابزاری در نظر گرفته می‌شود که می‌تواند حداکثر یک فعالیت را در واحد زمان انجام دهد. منابع تجدیدنشدنی، مثل سرمایه، مواد و انرژی، منابعی هستند که با انجام فعالیت‌هایی که به این منابع نیاز دارند، از مقدار کل اولیه‌ی آن کاسته می‌شود و در دوره‌های بعدی تجدید نمی‌شوند و مقدار باقیمانده می‌تواند برای مصرف فعالیت‌های بعدی مورد استفاده قرار گیرد. هدف مسئله‌ی زمانبندی پروژه با منابع محدود تعیین زمان شروع هر فعالیت به گونه‌ای است که زمان اجرای پروژه[۱] را کمینه نمایید. بدیهی است که پاسخ این سؤال نه تنها باید قیود مربوط به ارتباط منطقی فعالیت‌ها را تأمین کند، بلکه باید به محدودیت منابع نیز توجه داشته باشد.[۱] به منظور تسهیل در برنامه‌ریزی و کنترل پروژه، برنامه اجرایی پروژه را معمولاً بر روی شبکه نشان می‌دهند. شبکه‌های زمانبندی پروژه انواع مختلفی دارند. در این مساله پروژه به صورت یک شبکه فعالیت روی گره[۲] مانند گراف $G = (V, E)$ که در آن V مجموعه گره‌ها و بیانگر فعالیت‌های پروژه و E مجموعه یال‌ها و بیانگر روابط منطقی بین فعالیت‌هاست، تعریف می‌شود. فعالیت‌ها از ۱ تا n شماره‌گذاری شده‌اند.فعالیت اول و آخر را که فعالیت‌های مجازی هستند، به عنوان نقطه‌ی شروع و پایان در نظر می‌گیریم که مدت زمان انجام و نیاز به منابع آنها صفر می‌باشد.فعالیت‌ها پس از شروع تا پایان چرخه عمر خود بدون وقفه اجرا می‌شوند.از اینجا به بعد فرض می‌کنیم منابع از نوع تجدید شدنی و عامل کنترل پایان به شروع باشد. مساله $RCPS$ را می‌توان با مدل زیر نشان داد :

$$Min \quad s_n \tag{۱.۱}$$

$$st: \quad s_۱ = ۰ \tag{۲.۱}$$

$$s_i + d_i \leq s_j \quad \forall (i,j) \in E \tag{۳.۱}$$

$$\sum_{j \in A(t)} r_{jk} \leq R_k \qquad \forall t \geq ۰ \forall k = ۱, ..., m \tag{۴.۱}$$

که در آن متغیرها به صورت زیر تعریف می شوند:

n :تعداد فعالیت‌های پروژه

m : انواع منابع

d_j:مدت زمان اجرای فعالیت j

R_k:میزان موجودی منبع k

[۱] $Makespan$ [۲] $Activity\ On\ Node\ Network\ (AON)$

r_{jk}:میزان نیاز فعالیت j به منبع k

s_j: زمان شروع فعالیت j

$A(t)$:مجموعه فعالیت‌هایی که در پریود زمانی t در حال اجرا هستند

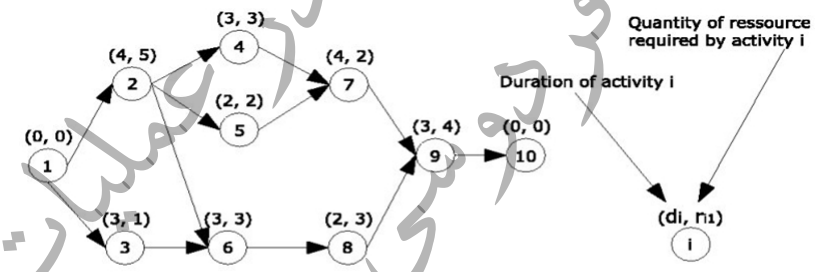
تابع هدف(۱.۱) زمان شروع فعالیت پایانی که همان زمان تکمیل پروژه است را مینیمم می کند. قید(۲.۱)زمان شروع پروژه را روز ۰ در نظر می‌گیرد .قید(۳.۱) رابطه پایان به شروع بین فعالیتj و فعالیت‌های پیش‌نیاز آن را نشان میدهد که به این دسته از محدودیت‌ها محدودیت‌های پیش نیازی[۱] .(۴.۱) نشان میدهد که میزان مصرف منبع k توسط فعالیت‌های در حال اجرا در هر پریود زمانی از میزان موجودی آن منبع بیشتر نباشد که به این دسته محدودیت‌ها محدودیت منابع[۲] می‌گویند.

یک زمانبندی [۳] برداری به صورت $S = (s_1, s_2, ..., s_n)$ است که در آن s_j زمان شروع فعالیت j را نشان میدهد.یک زمانبندی را شدنی[۴] گوییم هرگاه در (۳.۱)و (۴.۱) صدق کند.

یک ترتیب از فعالیت‌ها[۵]، یک جایگشت از $\{1, 2, ..., n\}$ است که به آن ترتیب ممکن[۶] گوییم هر گاه در شرط (۳.۱) صدق کند.

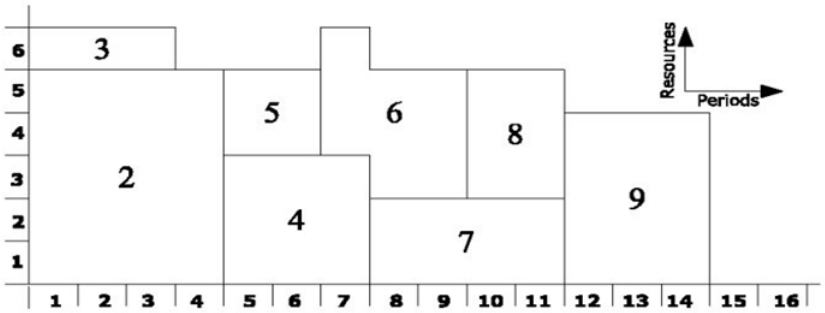
رابطه بین یک زمانبندی و یک ترتیب ممکن یک رابطه یک به یک نیست اما میتوان متناظر با هر ترتیب ممکن یک زمانبندی ارائه کرد که دارای کمترین $Makespan$ باشد.[۵]

شکل زیر یک شبکه AON و یک زمانبندی شدنی متناظر با آن را نشان می‌دهد. فرض شده است فقط یک منبع و به میزان ۶ واحد از این منبع در هر پریود موجود است.



شکل ۱.۱: شبکه AON

[۲] $resource\ constraints$ [۳] $schedule$ [۴] $feasible$ [۵] $Activity\ List$ [۶] $feasible\ Activity\ List$



شکل ۲.۱: مثالی از یک زمانبندی شدنی$S = (۰, ۰, ۰, ۴, ۴, ۶, ۷, ۹, ۱۱, ۱۴)$

۳.۱ زیر مسائل$RCPSP$

مسئله $Shop\ Scheduling$ در حالت کلی یک زیر مسئله از $RCPSP$ می‌باشد. در اینجا حالت خاصی از این مسئله تحت عنوان $Job\ Shop$ معرفی و به بیان ارتباط آن با $RCPSP$ پرداخته شده است. بعلاوه مسئله رنگ آمیزی گراف[۱] نیز به عنوان زیرمسئله‌ای از $RCPSP$ معرفی شده است. برای مشاهده زیر مسائل دیگر به مرجع [۲] مراجعه شود.

۱.۳.۱ مسئله‌ی $Job\ Shop$

ورودی‌های این مساله عبارتند از:

m ماشین:$M_۱, M_۲, ..., M_m$

n کار :$J = ۱, ۲, ..., n$

هر کار j شامل $n(j)$ عملیات مختلف است :$O_{۱j}, O_{۲j}, ..., O_{n(j)j}$

هر عملیات O_{ij} به مدت p_{ij} واحد زمان روی یک ماشین مشخص که آن را با μ_{ij} نشان می‌دهیم، $(\mu_{ij} \in$

[۱] $Graph\ Coloring$

$\{M_1, ..., M_m\}$) پردازش می‌شود.
هر ماشین در هر پریود زمانی حداکثر می‌تواند یک عملیات را پردازش کند.(۱)
بین عملیات مختلف هر کار jترتیبی به صورت داده شده وجود دارد:
$O_{1j} \to O_{2j} \to ... \to O_{n(j)j}$(۲)
در این مساله هدف پیدا کردن زمان شروع هر عملیات O_{ij} است به طوری که:
محدودیت‌های (۱) ،(۲) برقرار باشد و زمان تکمیل کارها توسط ماشین ها مینیمم شود.
این مساله را میتوان به صورت $RCPSP$ مدل کرد:
هر عملیات O_{ij} یک فعالیت (i,j) در نظر گرفته شود.در این صورت تعداد فعالیت‌ها $n(1)+n(2)+...+n(n)$ خواهد بود. به تعداد m منبع داریم که هر یک متناظر با یک ماشین است.میزان موجودی هر منبع در هر پریود زمانی برابر ۱ است.رابطه بین فعالیت‌ها نیز با استفاده از (۱) تعیین می‌شود. برای هر فعالیت (i,j): $d_{(i,j)} = p_{ij}$

$$r_{(i,j),k} = \begin{cases} 1 & \text{اگر } \mu_{ij} = k \\ 0 & \text{در غیر اینصورت} \end{cases}$$

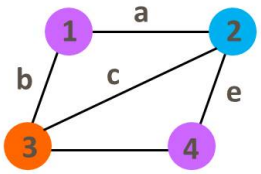
کافی است در مساله $Job\ Shop$ زمان شروع هر عملیات را زمان شروع فعالیت متناظر آن در $RCPSP$ در نظر بگیریم.[۲]

۲.۳.۱ مسئله‌ی $Graph\ Coloring$

در این مساله گراف بدون جهت $G = (V, E)$ داده شده است، هدف تعیین حداقل تعداد رنگ‌های لازم جهت رنگ‌آمیزی رئوس گراف است به طوری که هیچ دو راس مجاور همرنگ نباشند.
این مساله را می‌توان به صورت $RCPSP$ مدل کرد:
هر راس گراف یک فعالیت و هر یال یک منبع در نظر گرفته شود. موجودی هر منبع در هر واحد زمان و زمان اجرای هر فعالیت ۱ است.بین فعالیت‌ها هیچ رابطه کنترلی وجود ندارد بعلاوه میزان نیاز فعالیت j به منبع k به صورت زیر تعریف می‌شود:

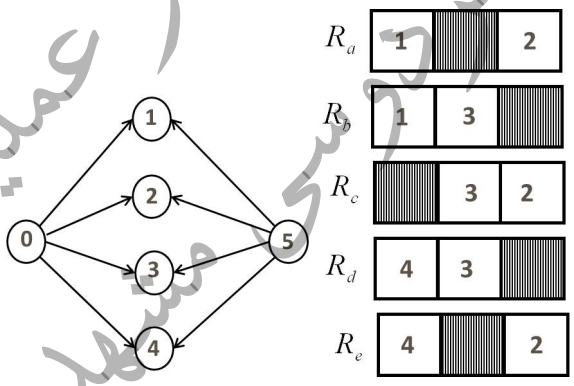
$$r_{jk} = \begin{cases} ۱ & \text{اگر راس } j \text{ مجاور یال } k \text{ باشد} \\ ۰ & \text{در غیر اینصورت} \end{cases}$$

مقدار مینیمم $Makespan$ بدست آمده برای $RCPSP$ حداقل تعداد رنگ‌ها برای رنگ آمیزی گراف را می‌دهد.در گراف داده شده زیر همانطور که دیده می شود، حداقل تعداد رنگ ها برای رنگ آمیزی گراف ۳ است.



شکل ۳.۱: گراف داده شده G

شبکه برنامه‌ریزی متناظر با این مثال به صورت زیر است، که در آن رئوس ۱و۵ متناظر با فعالیت‌های مجازی هستند.



شکل ۴.۱: یک زمانبندی با مینیمم $Makespan$ برابر است با حداقل تعداد رنگ‌های مورد نیاز برای رنگ آمیزی گراف $G = (V, E)$

همان طور که دیده می شود مقدار $Makespan$ برای زمانبندی بهینه ۳ است که برابر با حداقل تعداد رنگ‌های مورد نیاز برای رنگ‌آمیزی گراف می‌باشد.[۳]

۴.۱ کاربردهایی از $RCPSP$

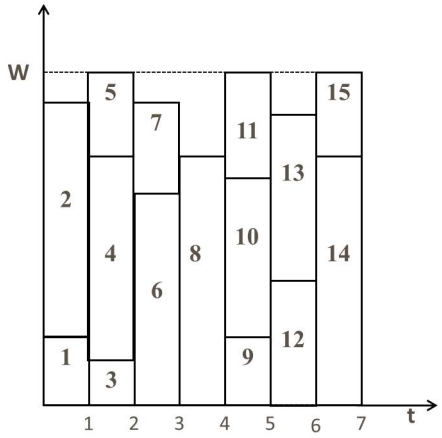
همانطور که در قبل اشاره شد مسئله $RCPS$ دارای کاربردهای وسیعی در مدیریت پروژه، زمانبندی تولید و... است. ما در اینجا به دو نمونه از کاربردهای این مسئله اشاره می‌کنیم. برای مشاهده کاربردهای بیشتری از این مسئله به مرجع [۲] مراجعه شود.

۱.۴.۱ کمینه‌سازی ضایعات برش

برخی از مواد مانند کاغذ، پارچه و...در کارخانه به صورت رل‌هایی استاندارد با طول مشخص W تولید می‌شوند.این رل‌ها متناظر با سفارشات به رل‌هایی با طول کوتاهتر برش داده می‌شوند. فرض کنید n رل با طول‌های w_i که $i = ۱, ۲, ..., n$ مورد نیاز باشد. هدف انجام برش‌ها برای تامین سفارشات است به طوریکه حداقل تعداد رل استاندارد مورد استفاده قرار گیرد. شکل زیر جوابی از مسئله‌ی ضایعات برش[۱] را در حالتی که $n = ۱۵$ نشان می‌دهد. دیده می شود که ۷ رل استاندارد در این جواب مورد استفاده قرار گرفته است.

این مسئله را می‌توان به صورت $RCPSP$ مدل کرد. مقدار تابع هدف حاصل حداقل تعداد رل‌های استاندارد را مشخص می‌کند و با توجه به زمانبندی هر فعالیت در جواب بهینه نحوه برش هر رل مشخص می‌شود.
برای این تبدیل هر رل کوتاهتر به عنوان یک فعالیت در نظر گرفته می‌شود.یک نوع منبع داریم که همان رل استاندارد است. میزان موجودی منبع برابر با طول رل یعنی W است.برای هر فعالیت زمان اجرا برابر ۱ و میزان

[۱] *A cutting stock problem*



شکل ۵.۱: یک جواب شدنی برای مسئله کمینه‌سازی ضایعات برش

منبع مصرفی برابر طول رل متناظر با آن فعالیت یعنی w_i است. بعلاوه بین فعالیت‌ها نیز هیچ عامل کنترلی وجود ندارد.[۲]

۲.۴.۱ جدول زمانبندی مدارس

در مسئله جدول زمانبندی مدارس[۱] مفروضات عبارتند از: m کلاس درس $\{C_1, C_2, ..., C_m\}$، h معلم $\{A_1, A_2, ..., A_h\}$، T پریود آموزشی $(t = 1, 2, ..., T)$ و n درس $(i = 1, 2, ..., n)$. هر درس باید توسط یک معلم و در یک کلاس تدریس شود. هر معلم A_j تنها در پریودهای خاصی در دسترس است. هدف تعیین پریودهای آموزشی برای هر درس است به طوری که:

در هر کلاس حداکثر یک درس در هر پریود تدریس شود.

هر معلم حداکثر یک درس در هر پریود ارائه دهد.

هر معلم تنها در پریودهایی که در دسترس است تدریس کند.

این مسئله را می‌توان به صورت $RCPSP$ مدل کرد. هر فعالیت متناظر با یک درس است که توسط یک معلم در یک کلاس تدریس می‌شود. بنابراین به تعداد n فعالیت و به تعداد $m+h$ منبع داریم. m منبع $1, 2, ..., m$

[۱] $High - School\ timetabling$

متناظر با کلاس‌ها و h منبع $m+۱, m+۲, ..., m+h$ متناظر با معلم‌های $A_۱, A_۲, ..., A_h$ هستند. میزان موجودی منابع عبارتند از:

$$R_k = ۱$$

$$R_{(m+j)}(t) = \begin{cases} ۱ & \text{اگر معلم } A_j \text{ در پریود } t \text{ در دسترس باشد} \\ ۰ & \text{در غیر اینصورت} \end{cases}$$

اگر فعالیت i درس اختصاص داده شده به کلاس C_l و معلم A_j باشد. میزان منابع مصرفی برای این فعالیت عبارت است از:

$$r_{ik} = \begin{cases} ۱ & \text{اگر } k = l \text{ or } k = m+j \\ ۰ & \text{در غیر اینصورت} \end{cases}$$

هدف یافتن زمانبندی شدنی برای مسئله است بطوری که $Makespan \leq T$.[۲]

فصل ۲

روش‌های حل

۱.۲ روش‌های حل $RCPSP$

دو رویکرد برای حل $RCPSP$ وجود دارد:[۴][۵]

۱.روش‌های دقیق

-روش برنامه‌ریزی ریاضی[۱][۷][۳]

-روش شاخه و کران[۲][۳]

-روش برنامه‌ریزی پویا [۳]

۲. روش‌های ابتکاری

-روش‌های ابتکاری مبتنی بر قواعد الویت‌بندی[۴][۵]

[۱] *Mathematical* [۲] *Branch & Bound* [۳] *Dynamic Programming* [۴] *Priority Rule Based*

-روش‌های فراابتکاری[۱] [۱۳]

بلازوویچ [۲] در سال ۱۸۸۳ با استفاده از مسئله *Job Shop* نشان داد که $RCPSP$ به شدت[۳] $NP-hard$ است.[۶] در نتیجه با افزایش فعالیت‌ها و انواع منابع حل آن توسط روش‌های دقیق در زمان معقول غیر ممکن است.برای پروژه‌های بزرگ روش‌های دقیق کارایی ندارند. به همین دلیل در سال‌های اخیر دامنه کاربرد روش‌های ابتکاری در حل این مساله افزایش یافته است.این روش‌ها جوابی تا حد امکان نزدیک به جواب بهینه را به دست می آورند.در ادامه به معرفی یکی از مدل‌های برنامه ریزی صفر ویک و چند الگوریتم ابتکاری می‌پردازیم.

۱.۱.۲ مدل ریاضی صفر ویک $RCPSP$

مدلی که در ادامه به معرفی آن خواهیم پرداخت یکی از اولین مدل‌های ارائه شده برای $RCPSP$ است.[۷] (پریتسکر [۴]، ۱۹۶۷) در ابتدا برای $Makespan$ یک کران بالا که آن را با T نشان می دهیم، تعیین می‌کنیم.این کران بالا را می توان توسط هر یک از روش‌های ابتکاری به دست آورد. می‌توان آن را به صورت زیر تعیین کرد:

$$T = \sum_{j \in V} d_j$$

متغیر تصمیم به صورت زیر تعریف می شود:

$$x_{it} = \begin{cases} ۱ & \text{اگر فعالیت } i \text{ در پریود } t \text{ به اتمام برسد.} \\ ۰ & \text{در غیر اینصورت} \end{cases}$$

با توجه به تعریف x_{it} مقدار $\sum_{t=۱}^{T} t x_{it}$ زمان تکمیل فعالیت i را نشان می‌دهد.مدل صفر ویک مسئله:

---

[۱] *Metaheuristic* [۲] *Blazewicz* [۳] *Strong* [۴] *Pritsker*

$$Min \qquad \sum_{t=1}^{T} t x_{nt} \qquad (۱.۲)$$

$$st: \qquad \sum_{t=0}^{T} x_{it} = 1 \qquad (۲.۲)$$

$$\sum_{t=0}^{T} t x_{jt} - d_j \geq \sum_{t=0}^{T} t x_{it} \qquad \forall (i,j) \in E \qquad (۳.۲)$$

$$\sum_{i \in V} r_{ik} \sum_{u=t}^{t+d_i-1} x_{iu} \leq R_k \qquad \forall t = 0, 1, ..., T \qquad \forall k = 1, ..., m \qquad (۴.۲)$$

$$x_{it} \in \{0, 1\} \qquad \forall t = 0, 1, ..., T \qquad \forall i \in V \qquad (۵.۲)$$

تابع هدف(۱.۲) $Makespan$ که همان زمان تکمیل فعالیت انتهای پروژه است را مینیمم می‌کند. قید (۲.۲) نشان می‌دهد هر فعالیت تا انتهای پروژه تنها یکبار می‌تواند شروع شود. قید(۳.۲) و قید (۴.۲)به ترتیب محدودیت‌های پیشنیازی و محدودیت منابع را نشان می‌دهند. تعداد متغیرهای تصمیم در این مدل nT و تعداد قیود $n + |E| + mT$ است.مقدار $|E|$ حداکثر می‌تواند $\frac{n(n-1)}{2}$ باشد. بنابراین این مدل به تعداد $O(n^2 + mT)$ قید دارد.همان طور که دیده می‌شود هر چه مقدار T کوچکتر انتخاب شود از تعداد قیود کاسته می‌شود.

۲.۱.۲ روش‌های ابتکاری مبتنی بر قواعد الویت بندی

این روش ها از دو جزء تشکیل شده اند: روش تولید برنامه زمانی[۱] (SGS) و قاعده‌ی الویت بندی. به طور کلی دو روش برای تولید برنامه زمانی شناخته شده است: روش سری[۲] و روش موازی[۳]. هر دو روش یک زمان بندی شدنی با استفاده از توسعه ی مرحله به مرحله ی زمان بندی جزئی تولید می کنند.زمان بندی جزئی، زمانبندی است که در آن تنها زیر مجموعه ای از فعالیت ها زمانبندی شده اند.فعالیت هایی که در هر مر حله برای استفاده از منابع رقابت می کنند، با استفاده از قواعد الویت بندی مرتب شده اند.قواعد الویت‌بندی به هر فعالیت مجاز برای زمانبندی یک مقدار الویت،$\nu(j)$، نسبت می‌دهد. با توجه به قاعده به کار رفته فعالیت با بیشترین یا کمترین $\nu(j)$ در الویت انتخاب قرار می‌گیرد.در این مقاله به تشریح روش سری می‌پردازیم.برای آشنایی با روش موازی به مراجع[۴][۵]مراجعه شود.در ادامه تعدادی از پر کاربردترین قواعد الویت‌بندی را معرفی می‌کنیم.

---

[۱] *Schedule GenerationScheme* [۲] *Serial* [۳] *Parallel*

الگوریتم سری

روش سری اولین بار توسط کلی در سال ۱۹۶۳ میلادی مطرح شد.[۸] این روش برای حل $RCPSP$ با n فعالیت دارای n مرحله است. در هر مر حله دو مجموعه مجزا به نام های S_g(فعالیت‌هایی که زمانبندی شده‌اند.) و D_g (فعالیت‌هایی که فعالیت پیشنیازی ندارند یا پیشنیازهای آنها در مراحل قبل زمانبندی شده‌اند.) مورد استفاده قرار می‌گیرند. در هر مرحله به کمک یک قاعده الویت‌بندی یک فعالیت از D_g انتخاب می‌شود و در زودترین زمان ممکن که به تأمین منابع مورد نیازش منجر شود، زمانبندی می‌گردد. این فعالیت از D_g حذف شده وبه S_g اضافه می‌گردد.با زمانبندی فعالیت nام اجرای الگوریتم به پایان می‌رسد.
مقدار باقی مانده‌ی منبع k در هر پریود زمانی t برابر است با:

$$\Pi R_{kt} = R_k - \sum_{j \in A(t)} r_{jk}$$

الگوریتم سری

گام آغازین:

$S_g \leftarrow \emptyset; g \leftarrow ۱$

تا زمانی که $|S_g| \leq n$ مراحل زیر را تکرار کن:

D_g و ΠR_{kt} را برای هر $t = ۱, ۲, ..., T$ و $k = ۱, ۲, ..., m$ محاسبه کن.

$j^* \leftarrow min_{j \in D_g}\{j | \nu(j) = inf_{i_g}\{\nu(j)\}\}$

$EF_{j^*} \leftarrow max\{FT_i | i \in P_{j^*}\} + d_{j^*}$

$FT_{j^*} \leftarrow min\{t | EF_{j^*} \leq t \leq LF_{j^*}, R_{j^*k} \leq \Pi R_{k\tau}, \tau = t - d_{j^*} + ۱, ..., t, \forall k\}$

$S_{g+۱} \leftarrow S_g \cup \{j^*\}$

$g \leftarrow g + ۱$

پایان

در الگوریتم فوق EF_j زودترین زمان پایان[۱] فعالیت j و LF_j دیرترین زمان پایان [۲]فعالیت j نامیده می‌شود. مقدار LF برای هرفعالیت از طریق روش مسیر بحرانی[۳] CPM محاسبه می‌شود. به این ترتیب که مقدار دیرترین زمان پایان فعالیت انتهایی پروژه برابر T در نظر گرفته می‌شود و سپس به کمک محاسبات پسرو[۴] مقدار LF سایر فعالیت ها به دست می‌آید. برای آشنایی با روش CPM به مرجع[۴] مراجعه شود.

[۱] *Earliest Finish Time (EF)* [۲] *Latest Finish Time (LF)* [۳] *Critical Path Method* [۴] *Backward*

پیچیدگی الگوریتم سری برابر با $O(n^۲m)$ است.[؟]همانطور که قبلا اشاره شد متناظر با هر ترتیب ممکن یک زمانبندی می‌توان یافت که دارای کمترین $Makespan$ باشد.این کار توسط الگوریتم سری انجام می‌شود.در این حالت با داشتن ترتیب فعالیت ها نیازی به محاسبه D_g در هر مرحله نیست. بلکه فعالیت ها به همان ترتیبی که در لیست داده شده انتخاب و زمانبندی می‌شوند.[۵]

قواعد الویت‌بندی

در زیر تعدادی از قواعد الویت‌بندی ارائه شده است.[۴]

۱ـقاعده‌ی کمترین زمان تأخیر[۱]

الویت در تخصیص منابع به فعالیت‌های بحرانی داده می‌شود. فعالیت بحرانی به فعالیتی گفته می‌شود که کمترین تأخیر در شروع آن باعث ایجاد تأخیر در زمان تکمیل پروژه شود.

۲ـقاعده دیرترین زمان پایان[۲]

الویت در این قاعده به فعالیتی داده می‌شود که دیرترین زمان پایان آن در برنامه زمانبندی پروژه، به مقطع زمانی که در آن، منابع به فعالیت‌ها اختصاص می‌یابند نزدیک‌تر باشد.

۳ـقاعده‌ی کوتاهترین زمان اجرا[۳]

هدف در این قاعده از پیش پا بر داشتن فعالیت‌هایی است که زمان اجرای کمتری دارند.

۴ـقاعده‌ی دیرترین زمان شروع[۴]

الویت به فعالیت‌هایی داده می‌شود که دیرترین زمان شروع آنها در برنامه زمانبندی پروژه، به مقطع زمانی که در آن، منابع به فعالیت‌ها اختصاص می‌یابند نزدیک‌تر باشد.

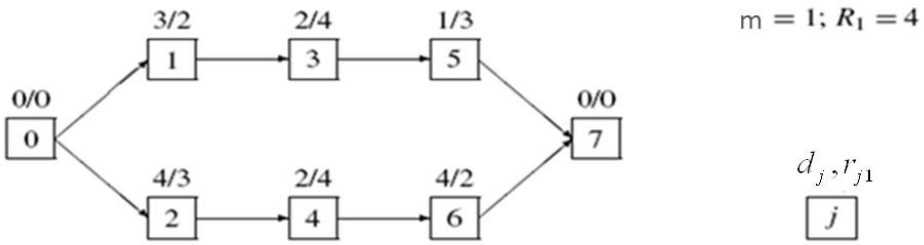
۵ـقاعده‌ی انتخاب تصادفی[۵]

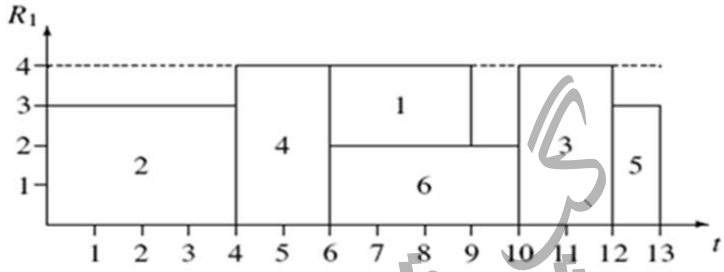
بر اساس این قاعده از میان فعالیت‌های واجد شرایط (انجام پیشنیازها و منابع به میزان کافی) یکی به تصادف انتخاب می‌شود.

---

[۱] $Minimum\ slack\ time\ (MINSK)$ [۲] $Latest\ finish\ time\ (LFT)$ [۳] $Shortest\ processing\ time\ (SPT)$

[۴] $Earliest\ start\ time\ (LST)$ [۵] $Random\ (RAN)$





g	1	2	3	4	5	6
$\mathcal{D}_g$	{1,2}	{1,4}	{1,6}	{3,6}	{3}	{5}
j	2	4	1	6	3	5

شکل ۱.۲: یک شبکه زمانبندی و یک زمانبندی شدنی که توسط روش سری با استفاده از قاعده الویت بندی LFT به دست آمده است.

۳.۱.۲ روش های فرا ابتکاری

برای مسئله $RCPS$ انواع روش‌های فراابتکاری مانند جستجوی همسایگی متغیر[۱] [۹]، آنیلینگ شبیه‌سازی شده[۲] [۱۰]، جستجوی ممنوع[۳] [۱۱] [۱۲]، الگوریتم ژنتیک[۴]، سیستم ایمنی مصنوعی[۵] [۱۴] [۱۵] و ... پیاده سازی شده‌اند. انواع روش‌های فراابتکاری برای این مسئله در مرجع [۱۳] آورده شده است. در ادامه

[۱] *Variable neighborhood search* [۲] *Simulated annealing* [۳] *Tabu search* [۴] *Genetic algorithm*

[۵] *Artificial Immune Algorithm (ASA)*

الگوریتم‌های جستجوی ممنوع و سیستم ایمنی مصنوعی برای $RCPS$ معرفی می‌شوند.

الگوریتم جستجوی ممنوع

الگوریتم جستجوی ممنوع یا به اختصار TS در اواخر دهه ۱۹۸۰ و توسط گلوور[۱] ارائه گردید. این الگوریتم احتمال قرار گرفتن در نقاط بهینه محلی[۲] را کاهش می دهد.این روش از مجموعه ای از حرکت ها[۳] برای انتقال از یک جواب به جواب دیگر واز یک تابع ارزیابی[۴] برای تعیین ارزش هر یک از حرکت ها استفاده می‌کند.شکل روش TS بسیار متنوع است و با توجه به نوع مساله از حرکت‌ها و توابع ارزیابی گوناگونی استفاده می‌شود.

[۱۱]متغیرها و پارامترهای جستجوی ممنوع

هدف از الگوریتم جستجوی ممنوع، یافتن یک ترتیب ممکن در فضای جواب (مجموعه‌ی تمام ترتیب‌های ممکن) است، به نحوی که برنامه بهینه یا شبه بهینه[۵] را ارائه نماید.درادامه به برخی از تعاریف پرداخته می شود.

یک جواب اولیه[۶] یک ترتیب ممکن است که از یکی از روش‌های ابتکاری حاصل شده است و بوسیله الگوریتم TS بهبود خواهد یافت. در اینجا روش کمترین شناوری[۷] ($MINSLK$) مورد استفاده قرار گرفته است. در این روش ابتدا زمان شناوری هر یک از فعالیت‌ها بدون در نظر گرفتن محدودیت منابع به کمک روش CPM محاسبه می‌شود.سپس منابع به فعالیت‌هایی که زودترین تاریخ انجام آنها فرا رسیده و پیشنیازهای آنها تکمیل شده است، تخصیص داده می شود. با توجه به امکان وجود فعالیت‌های متعدد با این وضعیت، الویت تخصیص منابع به فعالیت هایی داده می شود که دارای کمترین زمان شناوری می باشند.در صورتی که فعالیت‌ها در زمان شناوری نیز مشابه باشند، الویت به فعالیتی داده می‌شود که زمان اجرای کمتری دارد.

---

[۱] $Glover$ [۲] $Local\ Optimum$ [۳] $Moves$ [۴] $Evaluation\ Function$ [۵] $Optimal\ or\ near-optimal$ [۶] $Starting\ Solution$

[۷] $Minimum\ Slack$

یک حرکت به معنی انتقال از یک ترتیب ممکن به ترتیب ممکن دیگر، بوسیله جابجا کردن مکان دو فعالیت i و j می باشد. تابع هدف f برابر با $Makespan$ متناظر با یک ترتیب ممکن است. ارزش حرکت[۱] برابرتفاوت تابع هدف ایجاد شده توسط جابجایی فعالیت‌ها وتابع هدف ترتیب ممکن فعلی می باشد.اگر ارزش حرکت منفی باشد، حرکت به نام حرکت بهبود دهنده[۲] نامیده خواهد شد.

در الگوریتم جستجوی ممنوع، می‌توان تنها یک لیست ممنوعه[۳] برای ثبت حرکتهای اخیر ایجاد نمود که این لیست از پدیدار شدن مجدد یک جواب در تعداد تکرارهای معین شده، جلوگیری می‌نماید. در اینجا به دلیل تفاوت خصوصیات فعالیتها در پروژه، از دو لیست ممنوعه استفاده شده است؛ چرا که فعالیتهای پروژه قابل تفکیک به دو دسته هستند: فعالیتهای بحرانی[۴] و فعالیتهاي غیربحرانی[۵]. اگر فعالیتی میزان شناوری زمانی صفر داشته باشد، فعالیت بحرانی خواهد بود.در غیر اینصورت غیر بحرانی نامیده می‌شود.

لیست $Tabu\ ListC$ شامل فعالیت‌های بحرانی ولیست $Tabu\ ListNC$ شامل فعالیت های غیر بحرانی است.در حالتی که یک فعالیت بحرانی، در یک حرکت بسمت جلو (ابتدای پروژه) حرکت می‌کند (یعنی زودتر آغاز می شود)، می‌تواند زودتر از زمان مقرر شده تمام شود. در این حالت این فعالیت در $Tabu\ ListC$ قرار گرفته و در وضعیت ممنوعه[۶] خواهد بود و به تعداد $Tabu\ TenureC$ تکرار درآنجا خواهد ماندکه $Tabu\ TenureC$ طول لیست ممنوعه برای فعالیت‌های بحرانی است که به صورت تعداد تکرار و توسط طراح الگوریتم مشخص می‌شود. $Tabu\ ListNC$ و $Tabu\ TenureNC$ نیز به طور مشابه استفاده می‌شود، با این تفاوت که فعالیت های غیر بحرانی ممنوعه تا تعداد$Tabu\ TenureNC$ تکرار به سمت جلو حرکت نخواهند کرد.

عدد صحیح $Num\ Of\ Move$ نشانگر تعداد حرکتهای داوطلبی[۷] است که در هر تکرار، تولید و ارزیابی می شوند. این حرکتهای داوطلب در لیست داوطلب[۸] ثبت می‌شوند. هر حرکت داوطلب در لیست داوطلب بوسیله یک تابع ارزیابی، ارزیابی می‌شود. تابع ارزیابی ترکیبی از ارزش حرکت و یک تابع جریمه[۹] است. تابع جریمه وقتی به کار می رود که برخی از محدودیت‌های از پیش تعیین شده نقض شده باشند. از آنجا که در این برنامه تنها از ترتیب‌های ممکن استفاده می شود، عملاْ نیازی به تابع جریمه وجود ندارد. وقتی یک ترتیب ممکن بوسیله حرکتی که حاوی یک فعالیت ممنوعه است، ایجاد شده باشد، یک حرکت محدود شده ممنوعه[۱۰] نامیده شده و تست تنفس[۱۱] بر روی آن انجام خواهد گرفت. تست تنفس یک فاکتور مهم براي ایجاد انعطاف در الگوریتم جستجوي ممنوع است. اگر این تست ارضاء شود، وضعیت ممنوعه فعالیت در نظر

---

[۱] $Value\ of\ Move$ [۲] $Improvement\ Move$ [۳] $Tabu\ List$ [۴] $Critical\ Activity$ [۵] $Non-critical\ Activity$ [۶] $Tabu\ Status$

[۷] $Candidate\ moves$ [۸] $Candidate\ List$ [۹] $Penalty\ Function$ [۱۰] $Tabu\ Restricted\ Move$ [۱۱] $Aspiration\ Test$

گرفته نخواهد شد، در غیر اینصورت، حرکت مجاز نمی باشد. تست تنفس مورد نظر در الگوریتم حاضر بدین شکل است که اگر زمان تکمیل پروژه در ترتیب ممکن جدید، کوتاهتر از بهترین زمان یافته شده تاکنون باشد، می توان وضعیت ممنوع را در نظر نگرفت. با این توضیحات، می توان چنین گفت که حرکت مجاز[۱]، حرکتی است که محدودشده ممنوعه نبوده و یا اگر هست، تست تنفس را اضاء نماید. بهترین حرکت[۲]، حرکتی مجاز است که بیشترین ارزش را در ارزیابی بوسیله تابع ارزیابی دارا باشد. ترتیب ایجاد کننده بهترین حرکت، بعنوان جواب اولیه براي تکرار بعدی در نظر گرفته می شود. باید توجه داشت که بهترین ترتیب، لزوما بهبود در جواب فعلی نخواهد بود. در واقع الگوریتم جستجوی ممنوع به این طریق از افتادن در نقطه کمینه محلی[۳] جلوگیری می‌نماید. عملکرد الگوریتم جستجوی ممنوع به پارامترهای متعددی بستگی دارد. یکی از مهمترین پارامترها طول لیست ممنوع[۴] است. اندازه برای طول لیست ممنوعه بر پایه تست عملی قرار دارد. طول کوتاه لیست، به احتمال زیاد باعث افتادن جستجو در دور (حلقه) خواهد شد و بر عکس در صورت بزرگ بودن طول لیست، جستجو تنها به ناحیه ای کوچک محدود خواهد شد. طول لیست ممنوع برای یک الگوریتم زمانبندی عموما می تواند بصورت تابعی ساده از تعداد فعالیتها، نظیر جذر تعداد آنها، تعیین شود. پارامتر مهم دیگر، تعداد حرکتهاي داوطلبی است که در هر تکرار در نظر گرفته می‌شود. هر دو فاکتور سرعت و کیفیت جواب، می تواند از طریق اتخاذ یک استراتژی مناسب برای لیست داوطلب تاثیر بپذیرد. برای محدود کردن تعداد دفعات جستجوی الگوریتم از دو متغیر تعداد تکرار برای یافتن جواب مجاز[۵] و تعداد تکرار برای یافتن جواب[۶] استفاده شده است.

الگوریتم جستجوی ممنوع برای $RCPSP$

الگوریتم TS برای یک مساله زمانبندی پروژه با محدودیت منابع و زمان‌های مشخص فعالیت‌ها به صورت زیر پیشنهاد می‌شود:

۱. یک ترتیب ممکن را انتخاب کرده وآنرا S_i بنامید.

- جواب اولیه را با استفاده از روش $MINSLK$ تولید نمائید.
- S_b(بهترین جواب یافت شده تاکنون) را برابر ترتیب ممکن S_i قرار دهید.

[۱] *Admissible Solution* [۲] *Best Move* [۳] *Local Minimum* [۴] *Tabu Tenure* [۵] *Max Try On Admissibile*

[۶] *Max Try On Better*

۲. فعالیتها را به دو دسته بحرانی و غیربحرانی تقسیم نمائید.

- پروژه را بدون در نظر گرفتن محدودیتهاي منابع، با روش CPM زمان بندی کنید.
- فعالیتهای بحرانی و غیر بحرانی را شناسائی نمائید.

۳. مقداردهی اولیه به متغیرهای مورد استفاده در الگوریتم جستجوی ممنوع را انجام دهید.

- لیستهای ممنوع $Tabu\ ListC$ و $Tabu\ ListNC$ را خالی نمائید.
- مقادیر $Tabu\ TenureC$ و $Tabu\ TenureNC$ را مشخص کنید.
- تعداد حرکت های داوطلب در لیست داوطلب را $(Num\ Of\ Move)$ مشخص کنید.
- مقادیر لازم را به شروط توقف، یعنی $Max\ Try\ On\ Admissibile$ و $Max\ Try\ On\ Better$ بدهید.
- مقادیر اولیه متغیرهای $Not\ Find\ Admissible$ (تعداد تکرارهایی که در آنها هیچ حرکت مجازی پیدا نشده است.) و $Not\ Find\ Better$(تعداد تکرارهایی که در آنها هیچ ترتیب ممکن بهتری از S_b یافت نشده است.) را برابر صفر قرار دهید. این متغیرها در واقع شمارنده تعداد تکرارهای ناموفق اند.

۴. لیست داوطلب را که حاوی تعداد $Num\ Of\ Move$ حرکت داوطلب است، ایجاد نمائید.

- دو فعالیت از ترتیب فعلی بصورت تصادفی انتخاب می‌شوند و مکان این دو فعالیت با هم تعویض می‌گردد. اگر ترتیب ایجاد شده ممکن نیست (پیش نیازی ها را رعایت نمی کند)، زوج دیگری را انتخاب نمائید.
- این کار را آنقدر ادامه دهید که به تعداد $Num\ Of\ Move$ جابجائی پیدا شود.

۵. بهترین حرکت داوطلب مجاز را انتخاب کنید.

- متغیر های بولین $Find\ Admissible$ و $Find\ Better$ را برابر $False$ قرار دهید.
- برای هر حرکت داوطلب در لیست داوطلب، کارهای زیر را انجام دهید:
  - فرض کنید S_j ترتیب ممکن بعد از ایجاد جابجائی در S_i باشد.
  - ارزش حرکت $(f(S_i) - f(S_j))$ را محاسبه کنید.
  - اگر ارزش حرکت، از ارزش همه حرکت های مجاز پیدا شده تاکنون در لیست داوطلب، بیشتر است، آنگاه:

√ وضعیت ممنوعیت حرکت را بررسی کنید. اگر فعالیتی که بسمت انتهاي شبکه حرکت می کند،فعالیت بحرانی است و در لیست $Tabu\ ListC$ قرار دارد یا فعالیتی که بسمت ابتدای شبکه جابجا می شود، فعالیتی غیربحرانی است و در لیست $Tabu\ ListNC$ قرار دارد، در اینصورت این جابجائی، جابجائی محدود شده ممنوع است.

√ اگر جابجائی، محدود شده ممنوع نباشد، آنگاه:

- این جابجائی را بعنوان بهترین حرکت مجاز داوطلب پذیرفته و با M_b نمایش دهید.
- ارزش متغیر $Find\ Admissible$ را $True$، نمائید.
- مقدار متغیر $Not\ Find\ Admissible$ را مجدداً برابر صفر قرار دهید.

√ در غیر اینصورت:

- این جابجائی تست $Aspiration$ را خواهد گذراند، در صورتیکه $f(S_j) < f(S_i)$.
- اگر این حرکت از تست $Aspiration$ عبور کرد، آنگاه:
- این جابجائی را بعنوان بهترین حرکت مجاز داوطلب پذیرفته و با M_b نمایش دهید.
- ارزش متغیر $Find\ Admissible$ را $True$، نمائید.
- مقدار متغیر $Not\ Find\ Admissible$ را مجدداً برابر صفر قرار دهید.

۶. بهترین حرکت مجاز داوطلب را انجام دهید.

- فرض کنید S_j ترتیب ممکن پس از اعمال حرکت M_b بر روی S_i باشد.
- اگر $f(S_j) < f(S_i)$، آنگاه S_j جایگزین S_b شده و ارزش $Find\ Better$ برابر$True$ می گردد.
- اگر $Find\ Admissible$ برابر $True$ نیست، مقدار $Not\ Find\ Admissible$ را یکی افزایش دهید.
- اگر ارزش $Find\ Better$ برابر $True$ نیست، مقدار $Not\ Find\ Better$ را یکی افزایش دهید.

۷. شروط توقف را بررسی کنید.

- اگر مقدار $Not\ Find\ Admissible \geqslant Max\ Try\ On\ Admissible$ است یا $Not\ Find\ Better \geqslant Max\ Try\ On\ Better$ است، در اینصورت S_b جواب خواهد بود. در غیر اینصورت به گام ۸ بروید.

۸. محدودیتهای ممنوعه و معیار تنفس را بروز نموده و سپس به گام ۴ بروید.

- $Tabu\ ListNC$ و $Tabu\ ListC$ را بروز نمائید.

مقایسه الگوریتم جستجوی ممنوع پیشنهادی با الگوریتم آنیلینگ شبیه‌سازی شده

پترسون[۱] یک مجموعه متشکل از ۱۱۰ مثال آزمون ایجاد کرد. این مجموعه حاوی مسائلی با ۷ تا ۵۰ فعالیت که هر یک دارای ۱ تا ۳ نوع منبع تجدیدشدنی هستند، می‌باشد. تعداد ۱۰۸ پروژه از ۱۱۰ پروژه ارائه شده توسط پترسون، توسط مورس[۲] و به وسیله روش $MINSLK$ حل شده‌اند و جوابهای آن در دسترس است.
جدول ۱ نتایج حاصل از ۱۰ بار آزمایش بر روی ۱۱۰ پروژه را با استفاده از الگوریتم SA نشان می دهد. N تعداد فعالیت‌ها در هر پروژه را نشان می‌دهد و$N(T)$ نشان‌دهنده تعداد برنامه‌های تولید شده در هر درجه حرارت در فرایند آنیلینگ است.

جدول۱: نتایج حاصل از بکارگیري الگوریتم SA[۱۰]

$N(T)$	N	$۲N$	$۳N$	$۵N$	$۱۰N$
جواب‌های بالاتر از بهینه	٪۰.۸۷	٪۰.۵۵	٪۰.۴۳	٪۰.۳۳	٪۰.۲۳
میزان جواب‌های بهینه بدست آمده	٪۷۷.۰۰	٪۸۳.۶۴	٪۸۶.۱۸	٪۸۹.۱۸	٪۹۱.۹۱
تعداد جواب‌های بهینه پیدا شده در تکرارها	۴۷	۵۶	۶۲	۷۳	۸۴
زمان‌های اجرا (ثانیه)	۰.۳۱۷	۰.۶۲۸	۰.۹۱۸	۱.۵۱۳	۲.۹۷۳
انحراف معیار زمان‌های اجرا (ثانیه)	۰.۳۸۱	۰.۷۷۹	۱.۱۳۸	۱.۸۵۵	۳.۶۵۴

با فرض اینکه N تعداد کل فعالیت‌ها، مقدار $Tabu\ TenureNC$ و $Tabu\ TenureC$ برابر $\frac{\sqrt{N}}{۲}$ در نظر گرفته شده باشد. همچنین متغیر $Num\ Of\ Move$ نشانگر تعداد حرکت ها در لیست داوطلب برابر $\sqrt{N}$ لحاظ شده است. نتایج حاصله از ۱۰ بار اجرای هر یک از ۱۱۰ پروژه نمونه، که جواب اولیه آنها به روش $MINSLK$ محاسبه شده، در جدول ۲ ذکر گردیده است.

[۱] $Patterson$ [۲] $Morse$

جدول۲: نتایج حاصل از بکارگیري الگوریتم TS

$N(T)$	N	$۲N$	$۳N$	$۵N$	$۱۰N$
Max Try On Admissible	۱۰۰۰	۳۰۰۰	۶۰۰۰	۱۰۰۰۰	۲۰۰۰۰
Max Try On Better	۱۰۰	۳۰۰	۶۰۰	۱۰۰۰	۲۰۰۰
جواب‌های بالاتر از بهینه	٪۱.۴۰	٪۰.۶۹	٪۰.۴۰	٪۰.۲۹	٪۰.۱۹
میزان بهبود در زمان‌های انجام پروژه	٪۹.۲۹	٪۹.۸۹	٪۱۰.۱۳	٪۱۰.۲۳	٪۱۰.۲۳
میزان جواب‌های بهینه بدست آمده	٪۶۶.۶۴	٪۸۱.۳۶	٪۸۷.۴۶	٪۹۰.۶۴	٪۹۳.۴۳
تعداد جواب‌های بهینه پیدا شده در تکرارها	۴۳	۶۲	۷۶	۸۷	۹۵
زمان‌های اجرا (ثانیه)	۰.۲۸۳	۰.۶۸۷	۱.۱۷۳	۱.۸۵۳	۳.۳۹۶
انحراف معیار زمان‌های اجرا (ثانیه)	۰.۳۵۳	۰.۸۱۳	۱.۳۰۳	۲.۰۷۴	۳.۶۹۴

بامقایسه نتایج ذکر شده در جداول ۱ و ۲، می توان نتیجه گرفت که در شرایطی که زمان محاسباتی خیلی کوچک (متوسط ۰.۳ ثانیه) است. الگوریتم SA بهتر از الگوریتم TS عمل می‌کند. وقتی که متوسط زمان محاسباتی به بیش از ۰.۳ ثانیه می رسد، به نظر می‌رسد که الگوریتم جستجوی ممنوع در یافتن جواب های بهینه زمانهای اجرای پروژه توانمندتر است . در این آزمایش‌ها بیش از ۹۳٪ جواب‌های بهینه و در کل جواب‌های بهینه برای ۹۵ پروژه با میانگین زمان اجرای الگوریتم برابر ۳.۴ ثانیه، توسط الگوریتم جستجوی ممنوع بدست آمده است . نتایج مشابه برای الگوریتم SA ارقامی برابر دستیابی به بیش از ۹۱٪ از جواب های بهینه و یافتن جواب بهینه ۸۴ پروژه در همه تکرارها با میانگین زمانی اجرای الگوریتم برابر ۳.۰ ثانیه را نشان می‌دهد.

الگوریتم ایمنی مصنوعی

طبیعت و به طور خاص سیستم‌های بیولوژیکی همیشه به دلیل پیچیدگی، انعطاف‌پذیری و فلسفه کارکردی آنها، برای کارشناسان سحر آمیز بوده‌اند. سیستم عصبی الهام بخش تکامل تدریجی شبکه‌های عصبی مصنوعی است و در حالتی مشابه، سیستم ایمنی، موجب پیدایش یک سیستم ایمنی مصنوعی[۱] AIS شده است. بر طبق نظر کاسترو[۲] و زوبن[۳] AIS را می‌توان به عنوان چکیده‌ی سیستم های محاسباتی الهام گرفته از نظریه‌ی ایمونولوژی[۴] و اجزای آن دانست.[۱۶]
برای درک بهتری از AIS به معرفی سیستم ایمنی بدن می‌پردازیم.

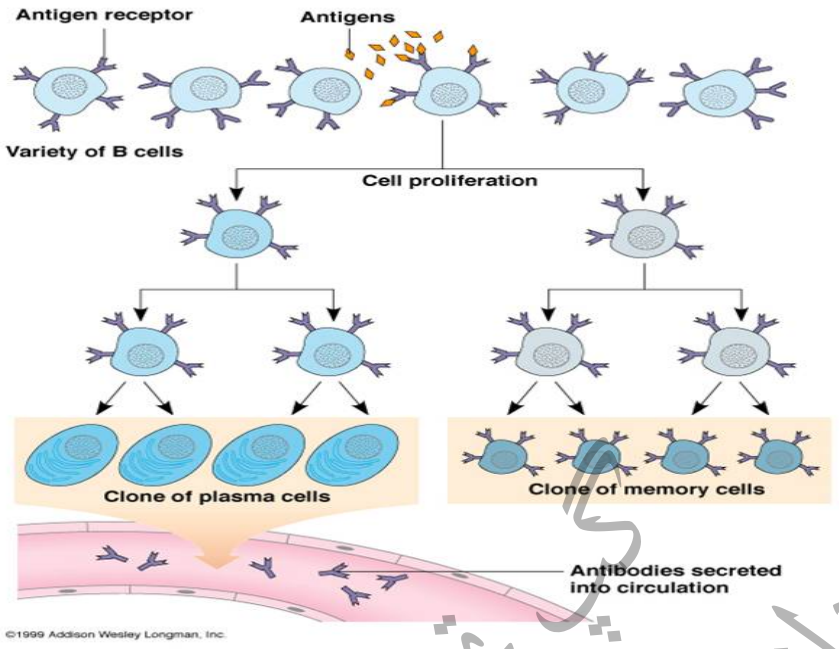
سیستم ایمنی نخاعی[۵]

ایمنی از سلول‌ها، مولکول‌ها و قوانینی تشکیل شده است که از آسیب رساندن عوامل بیماری‌زا به بدن میزبان جلوگیری می‌نماید. سطح این عوامل بیماری‌زا از گیرنده‌هایی به نام آنتی‌ژن[۶] تشکیل شده است. توانایی شناسایی این آنتی‌ژن‌ها توسط سیستم ایمنی موجب فعال شدن پاسخ ایمنی خواهد شد. یک نمونه از پاسخ‌های ایمنی می‌تواند ترشح مولکول‌های خاصی به نام آنتی‌بادی[۷] توسط یک نمونه از سلول‌های ایمنی بدن به نام $B-Cell$ باشد. این مولکول‌ها با قوانینی از پیش تعیین شده، آنتی‌ژن را شناسایی می‌نمایند و به آن متصل می‌شوند. هر سلول B توانایی تولید یک نوع آنتی‌بادی را داراست. میزان تعامل بین آنتی‌بادی و آنتی‌ژن با میزان تطابق و یا میل ترکیبی[۸] در ناحیه اتصال بین آن‌ها سنجیده می‌شود.
پس از ورود آنتی‌ژن، آن دسته از سلول‌های ایمنی که آنتی‌ژن را شناسایی کردند، شروع به تولید و تکثیر می‌نمایند. از بین سلول‌های تولید شده، دسته‌ای به عنوان سلول‌های حافظه انتخاب و نگهداری می‌شوند تا در برخوردهای بعدی سیستم ایمنی با این نوع آنتی‌ژن پاسخ ایمنی سریعتر وقویتر باشد. سلول‌های تولید شده دیگر به عنوان سلول‌های فعال وارد عمل شده و به مقابله با آنتی‌ژن می‌پردازند. در سیستم ایمنی سلول‌ها از طریق تقسیم سلولی تکثیر می‌شوند و در نتیجه سلول‌های تولید شده ساختاری مشابه سلول‌های والد خود دارند. اما هر سلول بر اساس میزان میل ترکیبی با آنتی‌ژن تحت تأثیر عملگر جهش قرار می‌گیرد. مراحل انتخاب و جهش، بلوغ میل ترکیبی[۹] نامیده می‌شود.شکل زیر نحوه‌ی تکثیر هر سلول B را نشان می‌دهد.

---

[۱] *Artificial Immune System* [۲] *Castero* [۳] *Zuben* [۴] *Immunology* [۵] *Vertebrate Immune System* [۶] *Antigen*

[۷] *Antibody* [۸] *Affinity* [۹] *Affinity muturation*



شکل ۲.۲: انتخاب کلونی

نظریه‌ی انتخاب کلونی[۱]

این نظری توسط بورنت[۲] در سال ۱۹۵۴ میلادی مطرح شد. بر طبق این نظریه تنها آن دسته از سلول‌هایی که توانایی شناسایی آنتی‌ژن را داراهستند برای تکثیر انتخاب می‌شوند. بعلاوه هر چه میزان میل ترکیبی با آنتی‌ژن بیشتر باشد، آن سلول می‌تواند فرزندان بیشتری تولید کند.از طرفی این سلول با نرخ کمتری جهش پیدا می‌کند و هر چه میل ترکیبی کمتر باشد، سلول فرزندان کمتری تولید می‌کند ولی نرخ جهش برای سلول بیشتر است به این معنی که سلول‌های فرزند تولید شده میزان میل ترکیبی بیشتری با آنتی‌ژن نسبت به والد خود دارند.[۱۶]

برای حل یک مسئله با استفاده از سیستم ایمنی مصنوعی به سه سؤال باید پاسخ داد. اول نحوه‌ی نمایش داده‌های مسئله، دوم معیار اندازه‌گیری میل ترکیبی و سوم انتخاب یک الگوریتم ایمنی مصنوعی برای حل مسئله. الگوریتم‌های مختلفی برای سیستم ایمنی مصنوعی مطرح شده است.[۱۷] در اینجا به معرفی یک نمونه از آن می‌پردازیم.

---

[۱] *Clonal Selection* [۲] *Burnet*

الگوریتم انتخاب کلونی

از جمله الگوریتم‌های ایمنی که برای مسائل بهینه سازی توسعه یافتند می‌توان به الگوریتم انتخاب کلونی اشاره کرد. این الگوریتم توسط کاسترو و زوبن در سال ۲۰۰۱ میلادی مطرح شد.[۱۶]

۱. **گام آغازین**

- انتخاب جمعیت اولیه برای سلول‌های ایمنی

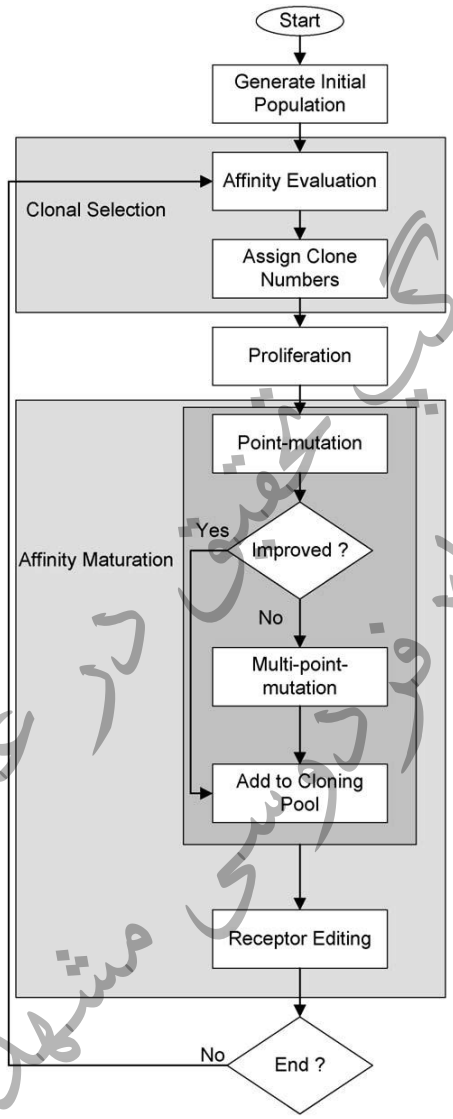
۲. **تا برقراری شرط توقف گام‌های زیر را تکرار کن**

- ارزیابی میزان میل ترکیبی
میزان میل ترکیبی هر سلول به الگوی آنتی‌ژن داده شده تعیین می‌شود.
- انتخاب کلونی
سلول‌هایی با میزان میل ترکیبی بیشتر به آنتی‌ژن انتخاب می‌شوند.هر چه میزان میل ترکیبی بیشتر بود میزان تولید کلون برای آن سلول بیشتر خواهد بود
- تغییر حالت ساختاری تحت جهش
نرخ جهش رابطه عکس با میزان شباهت با الگوی آنتی‌ژن دارد. هر چه میزان میل ترکیبی کمتر بود جهش با نرخ بیشتری انجام می‌شود وبالعکس.
- ساختن جمعیت جدید از سلول‌ها
تعدادی از سلول‌ها با میزان میل ترکیبی بیشتر جایگزین والدین می‌شوند.

الگوریتم ایمنی مصنوعی برای $RCPSP$[۱۴]

در این الگوریتم آنتی‌ژن، تابع هدف که همان کمینه کردن $Makespan$ است و آنتی‌بادی جواب کاندید شده برای $RCPSP$ است.نرخ تکثیر هر سلول متناسب با میل ترکیبی آن با آنتی‌ژن است به این معنی که هر چه آنتی بادی $Makespan$ کمتری داشته باشد، تعداد بیشتری در نسل بعد، از آن آنتی‌بادی تولید خواهد شد.

شکل زیر یک فلوچارت برای الگوریتم پیشنهادی را نشان می‌دهد. الگوریتم با یک جمعیت اولیه شروع شده و به طور تکراری، انتخاب کلونی، بلوغ میل ترکیبی و ویرایش گیرنده انجام می‌شود تا اینکه شرط توقف ارضاء شود. شرط توقف تعداد جواب‌های تولید شده در طی الگوریتم می‌باشد. سایر قسمت‌های الگوریتم در ادامه توضیح داده می‌شوند.



شکل ۳.۲: فلوچارت برای الگوریتم پیشنهادی

هر آنتی‌بادی یک ترتیب ممکن از فعالیت‌هاست.همانطور که در قبل اشاره شد به کمک الگوریتم سری با

داشتن هر آنتی‌بادی می‌توانیم جواب $RCPSP$ که یک زمانبندی شدنی S برای فعالیت‌هاست را مشخص کنیم.

۱ـتولید جمعیت اولیه

چون AIA یک الگوریتم جمعیتی است. بنابراین به یک مجوعه از جواب‌های آغازین برای شروع الگوریتم نیاز است. این جواب‌ها به طور تصادفی تولید شده‌اند به این ترتیب که برای هر آنتی‌بادی یک ترتیب ممکن (محدودیت‌های پیشنیازی و منابع برقرار باشند.) به طور تصادفی تولید می‌شود. به این دلیل که عملکرد الگوریتم و کیفیت جواب‌ها به کیفیت جواب آغازین وابسته است از یک روش بهبود دهنده روی جواب‌هایی که به طور تصادفی تولید می‌شوند استفاده شده است.این روش روس بهبود دهنده روش پیشرو و پسرو است که می‌توانید توضیحات راجع به این روش را در مرجع[۱۴] مشاهده کنید.

۲ـفرایند انتخاب کلونی

برای هر آنتی‌بادی مقدار میل ترکیبی از رابطه زیر محاسبه می‌شود.

$$Affinity(S) = \frac{۱}{Makespan(S)}$$

تعدادکلون‌های تولید شده توسط هر آنتی‌بادی بر اساس رویه زیر تعیین می‌شود:
(a) پیدا کردن ماکزیمم (Max) تابع هدف در جمعیت.
(b)محاسبه مجموع انحرافات (Δ) از بیشترین مقدار تابع هدف.
(c) معادله زیر تعداد کلون‌ها را برای هر آنتی‌بادی تعیین می‌کند.

$$Number\ of\ clones(S) = \frac{(Max - Makespan(S) + ۱)\sigma}{\Delta}$$

مقدار بزرگ برای σ ، تعدا بیشتر آنتی‌بادی‌های تولید شده در نسل جدید را باعث می‌شود و چون شرط توقف بر روی تعداد کل جواب‌های تولید شده است،σ یک فاکتور کنترل کننده‌ی تعداد تکرارها در الگوریتم است.اختلاف بین Max و مقدار تابع هدف هر جواب با مقدار ۱ جمع شده است تا شانس اندکی هم به بدترین آنتی‌بادی‌ها برای ایجاد کلون در نسل جدید داده شود.

۳-بلوغ میل ترکیبی

این مرحله توسط دو فرایند جهش تک نقطه‌ای[۱] و جهش چند نقطه‌ای[۲] انجام می‌شود. از جهش چند نقطه‌ای زمانی استفاده می‌شود که جهش تک نقطه‌ای نتواند بهبودی ایجاد کند. استفاده از این روش تضمین می‌کند سلول‌هایی با میل ترکیبی بالاتر تولید شوند بعلاوه برای سلول‌هایی که میل ترکیبی پایینی دارند، امکان جهش‌های بزرگتر ایجاد شده است تا به این ترتیب امکان فرار از دام بهینه محلی فراهم شود.

جهش تک نقطه‌ای

یک فعالیت از لیست فعالیت‌ها در ترتیب ممکن متناظر با آنتی‌بادی به تصادف انتخاب می‌شود و به مان جدیدی با شرط برقرار ماندن محدودیت‌های پیشنیازی منتقل می‌شود.

جهش چند نقطه‌ای

در این نوع جهش که بر اساس روش پیشرو - پسرو می‌باشد، برای هر آنتی‌بادی مقدار احتمالی از قبل تعیین شده است که با P_{per} نشان داده می‌شود. هر فعالیت با فعالیت بلافاصله مابعد خود در صورت برقرار بودن شرط پیشنیازی منتقل می‌شود. این رویه در زیر نشان داده شده است.در این جهش مقدار احتمال طوری محاسبه می‌شود که در گام‌های آغازین الگوریتم جهت‌های مختلفی را برای جستجو انتخاب کند و در گام های انتهایی الگوریتم به سمت بهترین جواب موجود همگرا شود. برای مشاهده نحوه‌ی تعیین مقدار احتمال به مرجع داده شده برای این الگوریتم مراجعه شود.

۴-تصحیح گیرنده

در این مرحله تعدادی از بهترین آنتی‌بادی‌های تولید شده در گام قبل با آنتی‌بادی‌هایی در نسل قبل که تمایل ترکیبی کمتری دارند جایگزین می‌شوند.
مراحل فوق تا برقرلری شرط توقف تکرار می‌شوند.

جایگاه الگوریتم ارائه شده در میان سایر روش‌ها

با توجه به بانك مسائل $PSPLIB$[۱۸]، در این قسمت به حل مسائل با تعداد فعالیتهای ۳۰، ۶۰ و ۱۲۰ مي‌پردازیم. طبق خواسته های مندرج در مقاله[۱۳] جواب‌ها بدین شکل محاسبه می‌شوند که الگوریتم باید

---

[۱] $point - mutation$ [۲] $Multipoint - Mutation$

پس از تولید تعداد مشخصی برنامه (۱۰۰۰، ۵۰۰۰ ویا ۵۰۰۰۰ برنامه)، متوقف شود و از جواب‌ها برای ارزیابی الگوریتم پیشنهادی استفاده می‌شود. تولید یك برنامه بصورت تعیین یك زمان آغاز برای هر فعالیت تعریف می‌شود. در مورد مسائل با ۳۰ فعالیت جواب بهینه موجود است ولی در مورد برنامه‌هایی با ۶۰ یا ۱۲۰ فعالیت، نتایج حاصله با حدود پایین مقایسه گشته است . در هر رده میانگین اختلاف جواب بدست آمده با حدود پایین اعلام می‌گردد. جداول ۱ تا ۳ که جواب های حاصل از راه‌حل ارائه شده را در رنکینگ کولیش و هارتمن[۱] قرار داده است، به وضوح نشان می‌دهند که عملکرد الگوریتم پیشنهادی خیلی خوب بوده است و در تمام مقایسات در میان بهترین‌ها قرار گرفته است. در این جداول الگوریتم‌ها براساس نتایجشان با تولید و اجرای ۵۰۰۰۰ برنامه مرتب شده‌اند. مرجع هر یک از الگوریتم‌های زیر را می‌توانید در مرجع [۱۳] مشاهده کنید.

جدول۱: متوسط انحراف از حل بهینه $J۳۰$

Algorithm	Reference	# of schedules		
		1000	5000	50000
Artificial Immune Algorithm	This study	0.05	0.03	0
GA, TS-path relinking	Kochetov and Stolyar	0.1	0.04	0
Scatter Search/electromagnetism	Debels et al.	0.27	0.11	0.01
GA-hybrid	Valls et al.	0.27	0.06	0.02
GA	Valls et al.	0.34	0.2	0.02
GA	Alcaraz et al.	0.25	0.06	0.03
Sampling-LFT	Tormos and Lova	0.25	0.13	0.05
TS	Nonobe and Ibaraki	0.46	0.16	0.05
Sampling-LFT	Tormos and Lova	0.3	0.16	0.07
GA	Hartmann	0.38	0.22	0.08
GA	Hartmann	0.54	0.25	0.08

جدول۲: متوسط انحراف از بهترین حل $J۶۰$

Algorithm	Reference	# of schedules		
		1000	5000	50000
Artificial Immune Algorithm	This study	11.17	10.8	10.55
Scatter Search/electromagnetism	Debels et al.	11.73	11.1	10.71
GA-hybrid	Valls et al.	11.56	11.1	10.73
GA, TS-path relinking	Kochetov and Stolyar	11.71	11.71	10.74
GA	Valls et al.	12.21	11.27	10.74
GA	Alcaraz et al.	11.89	11.19	10.84
GA	Hartmann	12.21	11.7	11.21
GA	Hartmann	12.68	11.89	11.23
Sampling-LFT	Tormos and Lova	11.88	11.62	11.36
Sampling-LFT	Tormos and Lova	12.14	11.82	11.47

[۱] $Hartmann$

جدول۳: متوسط انحراف از بهترین حل $J۱۲۰$

Algorithm	Reference	# of schedules		
		1000	5000	50000
GA-hybrid	Valls et al.	34.07	32.54	31.24
Artificial Immune Algorithm	This study	34.01	32.57	31.48
GA	Alcaraz et al.	36.53	33.91	31.49
Scatter Search/electromagnetism	Debels et al.	35.22	33.1	31.57
GA	Valls et al.	35.39	33.24	31.58
GA, TS-path relinking	Kochetov and Stolyar	34.74	33.36	32.06
Population-based	Valls et al.	35.18	34.02	32.81
GA	Hartmann	37.19	35.39	33.21
Sampling-LFT	Tormos and Lova	35.01	34.41	33.71
GA	Hartmann	39.37	36.74	34.03
Sampling-LFT	Tormos and Lova	36.24	35.56	34.77

مراجع

[۱] معرب،مژگان.برنامه ریزی پروژه با منابع محدود به کمک الگوریتم ژنتیک. پایان‌نامه دانشگاه فردوسی مشهد(۱۳۸۱).

[2] P.Brucker, S.Knust, Complex scheduling , $2^{nd}ed, Springer(2012)$.

[3] S.Demassey,E.Neron,Resource constrained-project scheduling: Models algorithma extensions and applications,WILEY(2008).

[۴] طارقیان،حامد رضا. برنامه‌ریزی و کنترل پروژه: مفاهیم و روش‌ها. مشهد:دانشگاه فردوسی مشهد(۱۳۸۸).

[5] R.Kolish, S.Hartmann,Heuristics algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis.In J.Weglarz,editor,Project

scheduling: Recent models,algorithms and applications.pages 147-178.Kluwer Academic Publisher,1999.

[6] J.Blazewicz,J.Lenstar,A.Rinnooy Kan, Scheduling subject to resource constraints: Classification and complexity,Discrete Applied Mathematics.5(1983)11-24.

[7] A.A.B.Pritsker,W.D.Watters,P.M. Wolfe,Multiproject scheduling with limited resources:A zero-one programming approach,Mnagement Science.16(1969)93-108.

[8] R.Kolish,Serial and parallel resource-constrained project scheduling methods revisited:Theory and computation,European Journal of Operational Research.90(1996)320-333.

[9] K.Fleszar,K.S. Hindi, Solving the resource constrained project scheduling problem by a variable neighbourhood search,European Journal of Operational Research.155(2004)402-413.

[10] Y. Tsai, D.D. Gemmill. Using a simulated annealing algorithm to schedule activities of resource-constrained projects, Working Paper No. 96-124, Department of Industrial and

Manufacturing Systems Engineering, Iowa State University, IA, 1996, Project Management

Journal, to appear.

[۱۱] برزین‌پور، فرناز. شاکری،آرنوش. فرزاد،احسان. حل مسئله زمانبندی پرروژه دارای محدودیت منابع با استفاده از الگوریتم جستجوی ممنوع. مجموعه مقالات سومین کنفرانس نین‌المللی مدیریت پروژه(۱۳۸۵).

[12] N.H.pan,P.W.Hsaio,K.Y.Chen,A study of project scheduling optimization using Tabu Search algorithm,Engineering Applications of Artificial Intelligence.21(200801101-1112.

[13] R.Kolish,S.Hartmann, Exprimental investigation of heuristics for resource-constrained project scheduling:an update,European Journal of Operational Research.174(2006)23-37.

[14] M.Mobini,Z.Mobini,M.Rabbani,An Artificial immune algorithm for the project scheduling problem under resource constraints,Applied soft computing.11(2011)1975-1982.

[15] R.Agarwal,M.K.Tiwari,S.K.Mukherjee,Artificial immune system based approach for solving resource constraint project scheduling problem,Int J Adv Manuf Technol.34(2007)584-593.

[16] L.N.De Castero,F.J.Von Zuben,Learning and optimization using the clonal selection principle,IEEETrans Evolutionary computation.6(2002) 239-251.

[17] L.N.De Castero,J.I.Timmis,Artificial immune systems as a novel soft computing paradigm,Soft computing.7(2003)520-544.

[18] http://www.bwl.uni-kiel.de/prod/psplib