

حل مسئله‌ی تسطیح منابع پروژه با استفاده از یک الگوریتم شبیه‌سازی تبرید کارا

فریبرز جولای* (استاد)

دانشکده مهندسی صنایع، دانشگاه تهران

محسن باقری (استادیار)

دانشکده‌ی مهندسی صنایع، دانشگاه صنعتی سجاد مشهد

مهندسی صنایع و مدیریت شریف، زمستان ۱۳۹۵ (۳۲-۱) شماره ۲/۲، ص. ۳۵-۴۵

در مسئله‌ی تسطیح منابع الگوی استفاده از منابع طی زمان مورد توجه قرار می‌گیرد و سعی می‌شود یک زمان‌بندی مبنا، که سطح استفاده از منابع در آن تا حد ممکن یکنواخت باشد، ارائه شود. این مسئله از جمله مسائل کلاسیک در حوزه‌ی زمان‌بندی پروژه است که برای آن راه‌حل‌های کارایی اندکی در ادبیات موضوع می‌توان یافت. در این نوشتار با استفاده از شبیه‌سازی تبرید اقدام به حل این مسئله می‌کنیم. شیوه‌ی نمایش جواب و تولید همسایگی براساس ویژگی‌هایی است که در این نوشتار بررسی و اثبات شده است. کارایی الگوریتم شبیه‌سازی تبرید از طریق مقایسه با سه روش ابتکاری دیگر و نیز با مقایسه با جواب‌های مدل ریاضی صفر- یک غیرخطی که با نرم‌افزار لینگو حل شده است، مورد بررسی قرار گرفته است. در این بررسی‌ها مجموعه‌ی متنوعی از مسائل آزمایشی مد نظر قرار داشته است. نتایج محاسباتی نشان می‌دهد، الگوریتم ارائه شده، نسبت به سه روش ابتکاری برتری دارد و نتایج قابل رقابتی را با جواب‌های مدل ریاضی به دست می‌دهد.

واژگان کلیدی: زمان‌بندی پروژه، مسئله‌ی تسطیح منابع، روش‌های فراابتکاری، الگوریتم شبیه‌سازی تبرید.

fjolai@ut.ac.ir
m.bagheri@sadjad.ac.ir

۱. مقدمه

تعمیراتی و پرسنل کلیدی نیروگاه‌های اتمی تولید برق،^[۶] در پروژه‌های طراحی و ایجاد واحدهای صنعتی^[۷] و پروژه‌های حوزه فناوری اطلاعات،^[۸] در ادبیات موضوع مشاهده می‌شود.

برخلاف مسائل تخصیص منابع که در آن زمان‌بندی فعالیت‌های پروژه با در نظر گرفتن محدودیت دسترسی به منابع هدف قرار می‌گیرد، در مسئله‌ی تسطیح منابع سعی بر آن است تا فعالیت‌ها به‌گونه‌ی زمان‌بندی شود که نوسانات سطح منابع کمیته باشد. در این دسته از مسائل الگوی استفاده از منابع اهمیت دارد و هدف رسیدن به نوعی زمان‌بندی است که پروفایل منابع آن تا حد ممکن بدون نوسان باشد. مسئله‌ی تخصیص منابع در دسته‌بندی دمیولیمیستر و هرولن^[۱] به صورت $m, \delta, cpm | level$ نمایش داده می‌شود.

۱.۱. روش‌های حل دقیق

۱.۱.۱. شمارش ضمنی

محققین بر مبنای شمارش ضمنی تمامی حالات ممکن برای زمان شروع فعالیت‌ها روش دقیق را ارائه کرده‌اند.^{[۹] و [۱۰]} بدین منظور مجموع مربع نوسانات منابع در دوره‌های متوالی^[۹] و نیز مجموع قدرمطلق تفاوت سطح منبع مورد استفاده با مقدار هدف^[۱۰] مد نظر قرار گرفته است. در روش یونس و سعد^[۱۱] ابتدا تمام فعالیت‌های بحرانی زمان‌بندی، و سپس تمام حالت‌های ممکن برای سایر فعالیت‌ها در نظر گرفته می‌شود. در این روش اگرچه جواب بهینه حاصل می‌شود، حل مسائل با اندازه بیش از ۱۰ فعالیت به‌ویژه وقتی شناوری فعالیت‌ها زیاد باشد، بسیار زمان‌بر و پیچیده می‌شود.

از آنجا که نوسانات منابع در مورد نیروی انسانی به معنای استخدام و اخراج، و در تجهیزات به معنای اجاره کردن و بازپس دادن است در اغلب پروژه‌ها این نوسانات نامطلوب است. مسئله‌ی تسطیح منابع از آن روی اهمیت می‌یابد که سعی در کمیته‌سازی نوسانات منابع انسانی و تجهیزات در طی زمان دارد. به‌ویژه در پروژه‌هایی که بی‌کار ماندن منابع (انسانی یا تجهیزات) یا بالا بودن بار کاری منابع هزینه‌بر است، این مسئله مهم تلقی می‌شود. علی‌رغم اهمیت ویژه‌ی این مسئله، در بسته‌های نرم‌افزاری معروف روشی مناسب برای تسطیح منابع وجود ندارد.^[۱۲] مثال‌هایی از کاربردهای عملی این مسئله در پروژه‌های راه‌سازی و ساختمان‌سازی،^[۲-۵] در زمان‌بندی پرسنل

* نویسنده مسئول

تاریخ دریافت: ۱۳۹۳/۷/۵، اصلاحیه ۱۳۹۳/۱۲/۲۴، پذیرش ۱۳۹۴/۴/۱۰.

عمده‌ترین ضعف روش‌های شمارش ضمنی، محدودیت آن‌ها برای حل مسائلی با اندازه‌های متوسط و بزرگ است. در روش‌های شج داده شده مسائلی با حجم بیشینه ۱۰ فعالیت به‌صورت بهینه حل شده‌اند.

۲.۱.۱. مدل‌سازی ریاضی

در برخی از منابع، حل دقیق مسئله‌ی زمان‌بندی از طریق مدل‌سازی ریاضی برای حالت خاصی از مسئله (کمینه‌سازی اوج بار کاری منابع) مطرح شده است.^[۱۱] در این تحقیق مثال‌های محدودی با بیشینه ۱۵ فعالیت به صورت بهینه حل شده است. بدین منظور یک مدل ریاضی برنامه‌ریزی مختلط غیرخطی برای مسئله‌ی تسطیح منابع ارائه شده^[۶] که آن را به مدلی خطی تبدیل می‌کند. در مدل مورد بررسی آن‌ها تابع هدف می‌تواند مجموع مربع نوسانات، یا مجموع قدرمطلق نوسانات باشد. در این مدل‌ها روابط پیش‌نیازی کمینه و بیشینه میان فعالیت‌ها وجود دارد. آن‌ها برای نخستین بار مسائلی با حجم بیشینه ۵۰ فعالیت با روابط پیش‌نیازی کمینه‌یی و بیشینه‌یی را به صورت بهینه حل کرده‌اند. در بررسی تأثیر شکست فعالیت‌ها روی بهبود تابع هدف در مسئله‌ی تسطیح منابع،^[۱۲] یک مدل برنامه‌ریزی خطی صفر و ۱ ارائه شده، و نیز یک مثال در دو حالت امکان شکست و عدم شکست حل شده است. حجم مسائلی که توسط این روش‌ها حل شده‌اند، تا ۵۰ فعالیت نیز افزایش یافته است، اما در این مدل‌ها روابط پیش‌نیازی بیشینه‌یی باعث محدودتر شدن فضای شمارش می‌شود.

۳.۱.۱. شاخه و حد

برای مسئله‌ی تسطیح منابع با پنجره‌های زمانی (روابط پیش‌نیازی کمینه‌یی و بیشینه‌یی)، روش شاخه و حد بر مبنای زمان شروع فعالیت‌ها و میزان شناوری باقی‌مانده‌ی فعالیت‌ها ارائه شده است.^[۱۳] در این روش مثال‌های با حجم بیشینه ۲۰ فعالیت به صورت بهینه حل شده است. برای مسئله‌ی تسطیح منابع با وجود پنجره‌های زمانی، ویژگی‌هایی از این مسئله ارائه شده^[۸] که براساس آن و با استفاده از الگوریتم شاخه و حد، حل دقیقی برای آن ارائه می‌شود. ویژگی مورد بررسی باعث تعریف مفهومی به نام درخت کمینه‌ی T شکل شده است که در شاخه زدن در الگوریتم شاخه و حد مورد استفاده قرار گرفته است. اندازه مسائل حل شده بیشینه ۲۰ است.

۲.۱. روش‌های حل ابتکاری

۱.۲.۱. الگوریتم‌های ابتکاری

محققین روشی تکراری بر مبنای جابه‌جایی فعالیت‌ها در حدود شناوریشان می‌پردازند.^[۱۴] این روش از جمله روش‌های معروف در حوزه‌ی زمان‌بندی پروژه است.^[۱] هریس^۱ برای مسئله‌ی تسطیح منابع با یک منبع تجدیدپذیر، روشی ابتکاری بر مبنای آنچه روش بسته‌بندی نامیده، ارائه می‌کند.^[۱۵]

این روش سعی در کنار هم قرار دادن فعالیت‌ها دارد به‌گونه‌یی که منابع مورد نیاز آن‌ها در پروفایل منابع به شکل مستطیل کامل درآید. سپس با جابه‌جایی بسته‌های تشکیل شده تابع هدف بهبود می‌یابد. آن‌ها در این تحقیق مثالی را به تفصیل با این روش حل کرده‌اند. تکموتو و همکاران^[۷] به مدل‌سازی مسئله با استفاده از برنامه‌ریزی کوادراتیک می‌پردازند و الگوریتمی ابتکاری بر مبنای انتخاب متغیرهای محوری برای حل مسائل با حجم بالا (بیش از ۱۰۰۰ فعالیت) ارائه می‌کنند و به حل مسئله‌یی واقعی در حوزه‌ی احداث واحدهای صنعتی می‌پردازند. برینکمن^۲ و نیومن^[۱۶] برای نخستین بار مسئله‌ی تسطیح منابع با پنجره‌های زمانی را مطرح می‌کنند و روشی ابتکاری بر مبنای لیست فعالیت‌ها برای حل ارائه می‌دهند. نیومن و همکاران نیز برای مسئله‌ی تسطیح منابع با پنجره‌های زمانی و محدودیت منابع، روشی ابتکاری براساس لیست اولویت ارائه کرده‌اند.^[۱۷، ۱۸] در روش مورد بررسی لیست اولویت برای فعالیت‌های غیربهرانی تشکیل می‌شود که هنگام زمان‌بندی به زمان‌بندی ناقصی که فعالیت‌های بحرانی در آن زمان‌بندی شده‌اند، اضافه می‌شود.

۲.۲.۱. الگوریتم‌های فراابتکاری

لو^۳ و همکاران^[۱۹] با استفاده از الگوریتم ژنتیک به حل مسئله‌ی تسطیح منابع می‌پردازند و براساس الگوریتم ارائه شده یک سیستم پشتیبان تصمیم‌گیری توسط واسط کاربر برای استفاده‌ی مدیران ساخت طراحی می‌کنند. در الگوریتم ژنتیک ارائه شده، شیوه‌ی نمایش جواب‌ها به صورت بردار زمان شروع فعالیت‌ها است. در این تحقیق کارایی الگوریتم از طریق مقایسه‌ی جواب یک مسئله (با ۱۱ فعالیت) با الگوریتم هریس^[۱۴] بررسی شده است. لو و همکاران^[۲۰] به مدل‌سازی مسئله‌ی تسطیح منابع در محیط فازی می‌پردازند و با استفاده از الگوریتم ژنتیک، براساس میزان دقت اطلاعات ورودی به ارائه‌ی جواب برای پروژه‌های ساخت می‌پردازند. جرجیوس^۴ و همکاران^[۲۱] برای مسئله‌ی تسطیح منابع با وجود محدودیت منابع روشی براساس الگوریتم شبیه‌سازی ترید استفاده کرده‌اند. در الگوریتم ارائه شده جواب‌ها به صورت لیست اولویت فعالیت‌ها نمایش داده می‌شود و همسایگی‌ها به صورت تعویض جفتی بر مبنای مشخصات شناوری‌ها صورت می‌گیرد. هاشمی دولابی و همکاران^[۲] به مدل‌سازی مسئله‌ی تسطیح منابع با امکان در نظر گرفتن شکست فعالیت‌ها می‌پردازند. آن‌ها برای حل مسئله از الگوریتم ژنتیک به همراه یک الگوریتم جست‌وجوی محلی و اصلاح زمان‌بندی، استفاده کرده‌اند. جواب‌های حاصل از الگوریتم ژنتیک بهبود قابل توجهی نسبت به الگوریتم زودترین زمان شروع دارد. شیوه‌ی نمایش و ایجاد همسایگی در الگوریتم ژنتیک ارائه شده رعایت پیش‌نیازی میان فعالیت‌ها را تضمین نمی‌کند. آن‌ها مثالی از کاربرد مدل در یک پروژه‌ی ساخت تونل ارائه می‌کنند. هون‌جون و ال‌ریز^[۴] دو هدف زمان اتمام پروژه و مسطح کردن منابع را در نظر گرفته‌اند و یک زمان‌بندی کاربردی از طریق محیط نرم‌افزاری کاربرپسند ارائه کرده‌اند. آن‌ها از الگوریتم ژنتیک چندهدفه برای حل مدل استفاده کرده‌اند.

خلاصه‌ی بررسی ادبیات موضوع در جدول ۱ آمده است، اگرچه برخی از محققین^[۲-۱۷، ۲۰] در امر مدل‌سازی نوآوری داشته‌اند و رویکردهای عملی را مورد توجه قرار داده‌اند، همگی فاقد الگوریتم قابل اطمینانی هستند که کارایی آن در مجموعه‌ی وسیعی از مسائل آزمایشی مورد سنجش قرار گرفته باشد. همچنین روش‌های حل دقیق تنها قادر به حل مسائلی با حجم کم و با وجود محدودیت‌های خاص -- مانند روابط پیش‌نیازی کمینه‌یی و بیشینه‌یی، یا امکان شکست فعالیت‌ها -- هستند.

در این نوشتار ضمن ارائه‌ی الگوریتم حل براساس روش فراابتکاری شبیه‌سازی ترید، برای مسئله‌ی تسطیح منابع دارای روابط پیش‌نیازی کمینه‌یی، به بررسی این الگوریتم در مجموعه‌ی وسیعی از مسائل آزمایشی می‌پردازیم. ساختار تحقیق بدین صورت است: در بخش دوم مدل مفهومی تسطیح منابع

جدول ۱. خلاصه‌ی تحقیقات انجام شده در زمینه‌ی زمان بندی پروژه با هدف تسطیح منابع.

مرجع	تابع هدف	ویژگی خاص	الگوریتم (دقیق یا ابتکاری؟)	حجم حل شده (بهینه)
[۱۳]	عمومی	-	ابتکاری (جابه‌جایی براساس شناوری و تکرار شونده)	-
[۹]	مجموع مربع نوسانات	-	دقیق (شمارش ضمنی)	۱۰
[۱۱]	اوج بارکاری منابع	در نظر گرفتن پیک استفاده از منابع	دقیق (برنامه‌ریزی ریاضی)	۱۵
[۱۴]	مجموع مربع نوسانات	-	ابتکاری (بر مبنای روش بسته‌بندی)	-
[۷]	مجموع مربع نوسانات	-	ابتکاری	-
[۱۵]	مجموع مربع نوسانات	وجود روابط پیش‌نیازی کمیته‌یی و بیشینه‌یی (پنجره زمانی)	ابتکاری (بر مبنای لیست اولویت فعالیت‌ها)	-
[۱۰]	مجموع قدرمطلق تفاوت از مقدار هدف	-	دقیق (شمارش ضمنی)	۱۰
[۲۰]	مجموع مربع تفاوت از مقدار هدف	-	ابتکاری (الگوریتم ژنتیک)	-
[۱۷، ۱۶]	مجموع مربع نوسانات و مجموع قدرمطلق نوسانات	وجود روابط پیش‌نیازی کمیته‌یی و بیشینه‌یی (پنجره زمانی)	ابتکاری (بر مبنای لیست اولویت فعالیت‌ها)	-
[۱۶]	مجموع مربع نوسانات	-	ابتکاری (الگوریتم ژنتیک)	-
[۱۲]	مجموع مربع نوسانات و مجموع قدرمطلق نوسانات	وجود روابط پیش‌نیازی کمیته‌یی و بیشینه‌یی (پنجره زمانی)	دقیق (شاخه و حد)	۲۰
[۱۸]	مجموع مربع تفاوت از مقدار هدف	امکان شکست فعالیت‌ها	دقیق (برنامه‌ریزی ریاضی)	۲۰
[۲۱]	مجموع مربع نوسانات	در نظر گرفتن محدودیت برای منابع	ابتکاری (شبیه‌سازی تبرید)	-
[۵]	هزینه شکست فعالیت‌ها	امکان شکست فعالیت‌ها	ابتکاری (گرافیکی در صفحات گسترده)	-
[۳]	مجموع مربع نوسانات	امکان شکست فعالیت‌ها	ابتکاری (الگوریتم ژنتیک)	-
[۲]	دو هدف: زمان اتمام پروژه و مجموع مربع نوسانات	-	ابتکاری (الگوریتم ژنتیک چند هدفه)	-
[۸]	مجموع مربع نوسانات و مجموع قدرمطلق نوسانات	وجود روابط پیش‌نیازی کمیته‌یی و بیشینه‌یی (پنجره زمانی)	دقیق (شاخه و حد)	۲۰
[۶]	مجموع مربع نوسانات و مجموع قدرمطلق نوسانات	وجود روابط پیش‌نیازی کمیته‌یی و بیشینه‌یی (پنجره زمانی)	دقیق (برنامه‌ریزی خطی)	۵۰

ارائه می‌شود، و اثبات ویژگی‌هایی از این مسئله در بخش سوم ارائه می‌شود. در بخش چهارم نیز نحوه‌ی نمایش جواب و شیوه‌ی تعیین پارامترهای الگوریتم شبیه‌سازی تبرید آمده است. در بخش پنجم نتایج محاسباتی ارائه خواهد شد و در نهایت نتیجه‌گیری و ارائه‌ی پیشنهادات برای تحقیقات آتی در بخش ششم ارائه می‌شود.

$$\min z = \sum_{k=1}^m C_k \sum_{t=0}^{S_n} (R_k(S^o, t) - R_k(S^o, t-1))^2 \quad (1)$$

st :

$$S_j^o \geq S_i^o + d_i \quad \forall (i, j) \in A \quad (2)$$

$$S_i^o = 0 \quad (3)$$

$$S_n^o \leq \delta_n \quad (4)$$

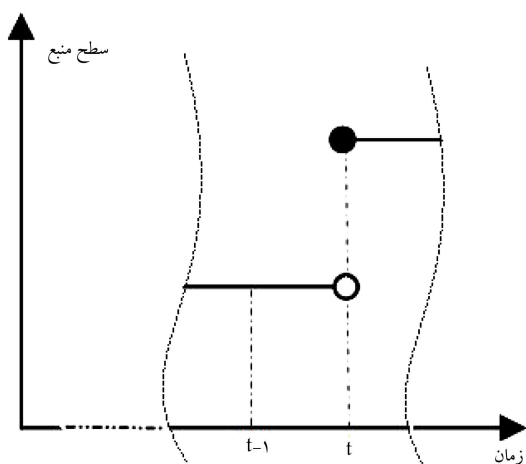
کمیته‌بودن تمامی این معیارها به معنای مسطح‌تر بودن پروفایل منابع است. در روابط فوق a_k را می‌توان به عنوان سطح هدف منبع k ام، متوسط سطح منبع k ام یا مقدار در دسترس منبع k ام تعبیر کرد. محدودیت ۲ نمایانگر محدودیت پیش‌نیازی میان فعالیت‌های پروژه است و این واقعیت را بیان می‌کند که اگر فعالیت i پیش‌نیاز فعالیت j است آنگاه شروع فعالیت j باید پس از پایان i باشد. زمان شروع پروژه (شروع فعالیت موهومی اول) صفر لحاظ می‌شود (رابطه‌ی ۳). محدودیت ۴ این واقعیت

۲. مسئله‌ی تسطیح منابع

مدل مفهومی مسئله‌ی تسطیح منابع با استفاده از نمادهای جدول ۲، به صورت روابط ۱ تا ۴ قابل نمایش است. در تابع هدف ۱ معیار عملکرد (PM) زمان بندی که مجموع وزن دار مربع نوسانات پروفایل منابع را در دوره‌های متوالی محاسبه می‌کند، کمیته شده است. در رابطه‌ی ۱ فرض بر این است که: $R_k(S^o, -1) = 0$. معیار عملکرد را می‌توان به صورت‌های دیگری چون مجموع موزون مربع سطح منابع مورد استفاده: $\sum_{k=1}^m C_k \sum_{t=0}^{S_n} (R_k(S^o, t))^2$ یا مجموع موزون قدرمطلق اختلاف سطح منابع مورد استفاده با سطح مورد نظر به صورت $\sum_{k=1}^m C_k \sum_{t=0}^{S_n} |R_k(S^o, t) - a_k|$

جدول ۲. شرح نمادهای مدل.

نماد	شرح
$G(N, A)$	شبکه‌ی غیرچرخه‌یی AON شامل مجموعه گره‌ها و مجموعه بردارها.
A	مجموعه روابط پیش‌نیازی شبکه: اگر $(i, j) \in A$ باشد، فعالیت i پیش‌نیاز j است.
N	مجموعه گره‌های شبکه که شامل n فعالیت است. در مجموعه‌ی $\{1, 2, \dots, n\}$ فعالیت‌های ۱ و n به ترتیب نشان‌دهنده‌ی فعالیت‌های موهومی شروع و پایان‌اند.
m	تعداد انواع منابع تجدیدپذیر.
d_i	مدت زمان انجام فعالیت i : $d_i \in \mathbb{Z}_{\geq 0}$.
C_k	ضریب (اهمیت) منبع k ام در محاسبه‌ی معیار عملکرد.
r_{ik}	مقدار منبع مورد نیاز k ام ($k = \{1, 2, \dots, m\}$) برای انجام فعالیت i : $r_{ik} \in \mathbb{Z}_{\geq 0}$.
S_i^0	زمان شروع فعالیت i در زمان‌بندی مبنا.
S^0	زمان‌بندی مبنا که توسط بردار شروع فعالیت‌های پروژه نمایش داده می‌شود: $S^0 = (S_1^0, S_2^0, \dots, S_n^0)$.
δ_n	زمان تحویل پروژه که زمان ختم پروژه از آن نباید بیشتر شود.
$R_k(S, t)$	مقدار منبع k ام مورد نیاز در زمان t در زمان‌بندی S .
$R_k(S, \circ)$	پروفایل منبع k ام در زمان‌بندی S : سطح احتیاجات منابع در زمان‌بندی S در طی زمان.



شکل ۱. پیوستگی راست پروفایل منابع.

به صورت مجموع مربع نوسانات منابع در دوره‌های متوالی باشد، این معیار به صورت $PM = \sum_{j=1}^J (p(t_j))^2$ قابل بازنویسی است. در شکل ۲ نقاط جهش پروفایل منابع در یک زمان‌بندی ناکامل نشان داده شده است. در این شکل‌ها همچنین تأثیر اضافه‌کردن یک فعالیت بر نقاط جهش نشان داده شده است.

۳.۳. ویژگی ۱: تأثیر جابه‌جایی یک فعالیت بر نقاط جهش پروفایل منابع

با تغییر زمان شروع فعالیت i از نقطه‌ی زمانی t' به نقطه‌ی زمانی t'' نقاط جهش تنها در نقاط شروع و پایان فعالیت تغییر می‌کنند و سایر نقاط جهش بدون تغییر باقی می‌مانند. تغییر در مقدار جهش در نقطه‌ی شروع جدید به صورت $P_{new}(t') = P_{old}(t') - r_i$ و $P_{new}(t'') = P_{old}(t'') + r_i$ اگر معیار سنجش عملکرد،

را بیان می‌کند که زمان ختم پروژه نباید از مهلت تحویل پروژه بیشتر شود در غیر این صورت زمان‌بندی غیر موجه خواهد شد.

۳. تعاریف، ویژگی‌ها و نتایج

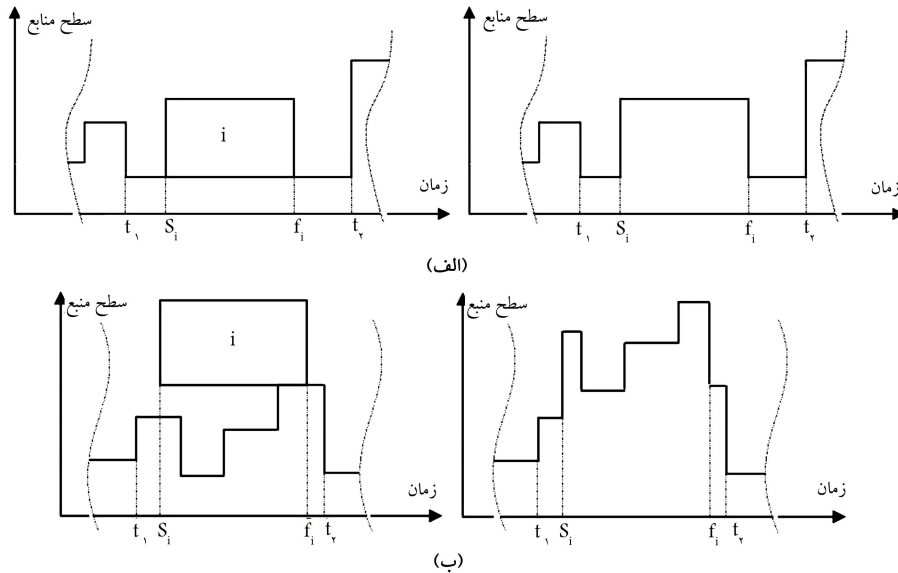
در این بخش تعاریف و ویژگی‌هایی در مسئله‌ی تسطیح منابع ارائه می‌شود که براساس نتایج آن می‌توان فضای جست‌وجوی گسسته در الگوریتم فراابتکاری شبیه‌سازی تبرید را بسیار محدودتر کرد. شیوه‌ی نمایش جواب و تولید همسایگی در الگوریتم شبیه‌سازی تبرید ارائه شده در قسمت ۴ براساس نتایجی است که در این بخش برای حذف جواب‌های مغلوب گرفته می‌شود. این موضوع چنان که در قسمت ۵ آورده شده، در سرعت و عملکرد الگوریتم تأثیر قابل توجهی دارد.

۱.۳. پیوستگی راست تابع پروفایل منابع

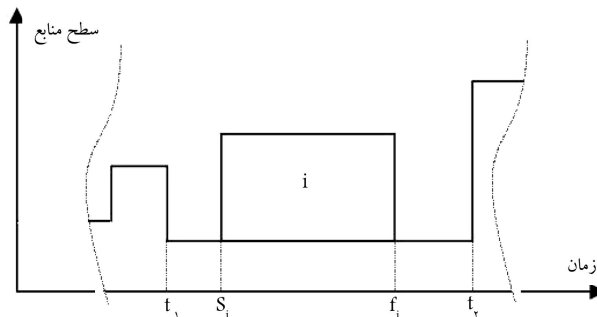
در زمان‌بندی S که پروفایل منبع (سطح منبع) آن با $(R_k(S, t), 0 \leq t \leq \delta_n)$ نمایش داده می‌شود، پروفایل منبع یک تابع پله‌یی و با پیوستگی راست است (شکل ۱).

۲.۳. نقاط جهش در پروفایل منابع

نقطه‌ی زمانی t یک نقطه‌ی جهش در پروفایل منابع است، مشروط بر آن که $R_k(S, t) \neq R_k(S, t-1)$. توجه داریم که یک نقطه‌ی جهش در پروفایل منبع، نقطه‌ی شروع (یا پایان) یک یا چند فعالیت است. مقدار جهش در نقطه‌ی t را با $p(t)$ نمایش داده و به صورت $p(t) = R_k(S, t) - R_k(S, t-1)$ تعریف می‌کنیم. $p(t)$ می‌تواند مقداری مثبت یا منفی داشته باشد. اگر تعداد نقاط جهش پروفایل منابع در زمان‌بندی S برابر با J باشد، زمان‌های نقطه‌ی جهش را با $T_1 \leq T_2 \leq \dots \leq T_J$ می‌دهیم. اگر معیار سنجش عملکرد،



شکل ۲. نقاط جهش پروفایل.



شکل ۳. نقاط t_1 و t_2 برای فعالیت i .

با توجه به این که $p_{old}(t_1) < 0$ و $p_{old}(t_1 + d_i) = 0$ است، ΔPM همواره مقداری منفی خواهد بود. این بدان معناست که جابه‌جایی انجام شده (آغاز فعالیت بلافاصله پس از اتمام یک یا چند فعالیت) منجر به بهبود تابع هدف می‌شود. در رابطه‌ی ۵ اگر $t' = s_i$ و $t'' = t_2 - d_i$ باشد و $p(t') = p(f_i - d_i) = -r_i$ باشد داریم:

$$\Delta PM = 4r_i^2 + 2r_i[p_{old}(t_2 - d_i) + p_{old}(f_i) - p_{old}(s_i) - p_{old}(t_2)]$$

از آنجا که $p_{old}(f_i) = -r_i$ و $p_{old}(s_i) = r_i$ داریم:

$$\begin{aligned} \Delta PM &= 4r_i^2 + 2r_i[p_{old}(t_2 - d_i) - 2r_i - p_{old}(t_2) - p_{old}(t_2)] \\ &= 2r_i[p_{old}(t_2 - d_i) - p_{old}(t_2)] \end{aligned}$$

با توجه به این که $p_{old}(t_2) > 0$ و $p_{old}(t_2 - d_i) = 0$ است، ΔPM همواره مقداری منفی خواهد بود. این بدان معناست که جابه‌جایی انجام شده (پایان فعالیت بلافاصله پیش از شروع یک یا چند فعالیت) منجر به بهبود تابع هدف می‌شود.

۱.۴.۳. نتیجه‌ی ۲

براساس ویژگی ۲ می‌توان نتیجه گرفت که در یک زمان بندی، اگر زمان شروع فعالیت i در حالت‌هایی مانند شکل ۳ را برابر با زمان پایان یک یا چند فعالیت قرار دهیم، یا زمان پایان فعالیت i را برابر با زمان شروع یک یا چند فعالیت قرار دهیم، معیار

نقطه‌ی پایان جدید و قدیم به صورت $P_{new}(t' + d_i) = P_{old}(t' + d_i) + r_i$ و $P_{new}(t'' + d_i) = P_{old}(t'' + d_i) - r_i$ است.

۱.۳.۳. نتیجه ۱

اگر معیار سنجش عملکرد، به صورت مجموع مربع نوسانات منابع در دوره‌های متوالی باشد $PM = \sum_{t=0}^{\delta_n} (p(t))^2$ ، تغییر ناشی از این جابه‌جایی چنین محاسبه می‌شود:

$$\begin{aligned} \Delta PM &= \sum_{t=0}^{\delta_n} [(p_{new}(t))^2 - (p_{old}(t))^2] \\ &= [(p_{old}(t') - r_i)^2 - (p_{old}(t'))^2] \\ &\quad + [(p_{old}(t'') + r_i)^2 - (p_{old}(t''))^2] \\ &\quad + [(p_{old}(t' + d_i) + r_i)^2 - (p_{old}(t' + d_i))^2] \\ &\quad + [(p_{old}(t'' + d_i) - r_i)^2 - (p_{old}(t'' + d_i))^2] \\ \Delta PM &= 4r_i^2 + 2r_i[p_{old}(t'') + p_{old}(t' + d_i) - p_{old}(t') \\ &\quad - p_{old}(t'' + d_i)] \end{aligned} \quad (5)$$

۴.۳. ویژگی ۲: نقاط مغلوب در جابه‌جایی یک فعالیت

اگر فعالیت i در زمان بندی ناکامل S بتواند بین دو نقطه‌ی t_1 و t_2 (دو نقطه جهش متوالی) مطابق شکل ۳ جابه‌جا شود، بهترین زمان بندی برای فعالیت i یکی از دو حالت $s_i = t_1$ و یا $f_i = t_2$ است. در رابطه‌ی ۵ اگر $t' = s_i$ و $t'' = t_1$ باشد داریم:

$$\Delta PM = 4r_i^2 + 2r_i[p_{old}(t_1) + p_{old}(f_i) - p_{old}(s_i) - p_{old}(t_1 + d_i)]$$

با توجه به این که $p_{old}(f_i) = r_i$ و $p_{old}(s_i) = r_i$ داریم:

$$\begin{aligned} \Delta PM &= 4r_i^2 + 2r_i[p_{old}(t_1) - 2r_i - p_{old}(t_1 + d_i)] \\ &= 2r_i[p_{old}(t_1) - p_{old}(t_1 + d_i)] \end{aligned}$$

عملکرد بهبود می‌یابد. از این رو می‌توان با اضافه کردن روابط پیش‌نیازی موقت اضافه و زمان‌بندی کردن فعالیت‌ها در زودترین و نیز دیرترین زمان ممکن به ترتیب دو حالت فوق را تحقق بخشید. از این نتیجه در سازوکار تولید همسایگی در الگوریتم شبیه‌سازی تبرید استفاده کرده‌ام. چگونگی به‌کارگیری این سازوکار در ادامه شرح داده شده است.

۴. الگوریتم شبیه‌سازی تبرید

براساس ویژگی‌های بررسی شده در قسمت قبل، با اضافه کردن فعالیت‌های پیش‌نیازی (موقت) اضافی و زمان‌بندی فعالیت‌ها در زودترین و نیز در دیرترین زمان ممکن، کوشیده‌ام تا زمان‌بندی‌هایی با معیار عملکرد مناسب به دست آوریم. با این‌کار زمان شروع یک فعالیت همواره بلافاصله پس از پایان فعالیت‌های پیش‌نیازش (در حالت زمان‌بندی در زودترین زمان)، یا پایان فعالیت بلافاصله پیش از شروع فعالیت‌های پس‌نیازش (در حالت زمان‌بندی در دیرترین زمان) قرار خواهد داشت که مطابق نتایج ذکر شده از لحاظ کردن گزینه‌های مغلوب جلوگیری می‌کند. علاوه بر این در هر تکرار الگوریتم شبیه‌سازی تبرید سرعت محاسبات با توجه به سادگی محاسبات رفت و برگشت مربوط به زمان‌بندی زودترین زمان (ES) و دیرترین زمان (LS)، به مقدار قابل توجهی افزایش خواهد یافت.

۱.۴. نمایش جواب در الگوریتم شبیه‌سازی تبرید

همسایگی‌ها به صورت ماتریس بالامتثلی روابط پیش‌نیاز موقت $A_{n \times n}^{temp}$ نمایش داده می‌شوند. هر عنصر باینری a_{ij}^{temp} از ماتریس، اگر فعالیت i پیش‌نیاز مستقیم j باشد، مقدار ۱ را اخذ می‌کند و در غیر این صورت صفر است.

با داشتن ماتریس روابط پیش‌نیازی اولیه پروژه (دائمی) $A_{n \times n}^{pern}$ و ماتریس روابط اضافی موقت $A_{n \times n}^{temp}$ ماتریس جدید تأثیرگذار $A_{n \times n}^{eff}$ به دست می‌آید که برای هر عنصر ماتریس $A_{n \times n}^{eff}$ داریم:

$$a_{ij}^{eff} = \max \{a_{ij}^{temp}, a_{ij}^{pern}\}$$

که با استفاده از آن زمان‌بندی ES و زمان‌بندی LS را انجام داده و دو زمان‌بندی با دو مقدار برای معیار عملکرد به دست می‌آید و زمان‌بندی با معیار عملکرد بهتر به عنوان جواب مربوط به همسایگی ایجاد شده در نظر گرفته می‌شود. گزینه‌های ممکن برای ایجاد ماتریس $A_{n \times n}^{temp}$ با استفاده از الگوریتم وارشال - فلوید^[۱] و ماتریس دورترین فاصله $A_{n \times n}^{lp}$ به دست می‌آید.

۱.۱.۴. الگوریتم دورترین فاصله وارشال - فلوید

با استفاده از این الگوریتم می‌توان ماتریس $(n \times n)$ دورترین فاصله $\Pi = [\pi_{ij}]$ را به دست آورد. هر عنصر π_{ij} این ماتریس نشان‌دهنده طولانی‌ترین فاصله میان فعالیت i و فعالیت j است. با تعریف ماتریس $\Pi^{(1)}$ به صورت:

$$\pi_{ij}^{(1)} = \begin{cases} 0 & \text{if } i = j \\ l_{ij} & \forall (i, j) \in E \\ -\infty & \text{O.W} \end{cases}$$

و رابطه‌ی تکراری:

$$\pi_{ij}^{(k+1)} = \max \left\{ \pi_{ij}^{(k)}, \pi_{ik}^{(k)} + \pi_{kj}^{(k)} \right\}$$

می‌توان $\Pi = \Pi^{(n+1)} = [\pi_{ij}^{(n+1)}]$ را به دست آورد. در این رابطه l_{ij} (فاصله میان گره i و گره j) به صورت $l_{ij} = d_i$ تعریف می‌شود. اگر در ماتریس Π عنصر π_{ij} برابر با $-\infty$ باشد به این معنی است که هیچ‌گونه رابطه‌ی پیش‌نیازی میان فعالیت i و فعالیت j (مستقیم یا غیر مستقیم) وجود ندارد. به عنوان مثال شبکه‌ی با $n = ۱۲$ فعالیت با شبکه‌ی پیش‌نیازی به صورت شکل ۴الف دارای ماتریس روابط پیش‌نیازی اولیه‌ی پروژه (دائمی) $A_{n \times n}^{pern}$ به صورت شکل ۵الف خواهد بود. در شبکه‌ی نشان داده شده به عنوان مثال، فعالیت ۲ پیش‌نیاز مستقیم فعالیت ۸ است، بنابراین عنصر باینری $a_{28}^{pern} = ۱$ خواهد بود. با در نظر گرفتن این روابط پیش‌نیازی و در نظر گرفتن زمان تحویل $\delta_n = ۳۲$ ، زمان‌بندی ES و LS معیارهای عملکرد مربوط به این دو زمان‌بندی به صورت $PM_{ES} = ۱۷۰$ و $PM_{LS} = ۲۹۶$ خواهد شد که پروفایل منابع مربوط به این دو زمان‌بندی در شکل ۶ نمایش داده شده است.

با استفاده از الگوریتم وارشال - فلوید ماتریس طولانی‌ترین مسیر $A_{n \times n}^{lp}$ به صورت شکل ۵ب خواهد بود. در ماتریس فوق اگر $a_{ij} = -\infty$ باشد، بدان معناست که هیچ مسیری از فعالیت i به فعالیت j وجود ندارد (فعالیت i چه به صورت مستقیم و چه به صورت غیر مستقیم پیش‌نیاز فعالیت j نیست). به عبارت دیگر اضافه کردن رابطه‌ی پیش‌نیازی میان فعالیت i و فعالیت j می‌تواند یک گزینه برای ایجاد همسایگی در الگوریتم شبیه‌سازی تبرید باشد. اگر در یک همسایگی، پیش‌نیازی‌های موقت به صورت زیر ایجاد شود:

پیش‌نیازی اضافه ۱: فعالیت ۲ پیش‌نیاز فعالیت ۳.

پیش‌نیازی اضافه ۲: فعالیت ۸ پیش‌نیاز فعالیت ۹.

پیش‌نیازی اضافه ۳: فعالیت ۹ پیش‌نیاز فعالیت ۱۰.

پیش‌نیازی اضافه ۴: فعالیت ۱۰ پیش‌نیاز فعالیت ۱۱.

با اضافه کردن روابط پیش‌نیازی فوق، ماتریس روابط پیش‌نیازی موقت $A_{n \times n}^{temp}$ به صورت شکل ۵ج خواهد شد که در نتیجه ماتریس روابط پیش‌نیازی تأثیرگذار $A_{n \times n}^{eff}$ به صورت شکل ۵د خواهد بود. شبکه‌ی فعالیت‌ها با در نظر گرفتن روابط پیش‌نیازی جدید به صورت شکل ۴ب است. با اضافه کردن روابط پیش‌نیازی ذکر شده، ماتریس پیش‌نیازی تأثیرگذار $A_{n \times n}^{eff}$ حاصل می‌شود که با زمان‌بندی ES و LS دو معیار عملکرد به صورت $PM_{ES} = ۲۳۸$ و $PM_{LE} = ۷۰$ حاصل می‌شود. پروفایل منابع مربوط به این زمان‌بندی‌ها در شکل ۷ به نمایش درآمده است. ملاحظه می‌شود با در نظر گرفتن روابط پیش‌نیازی اضافه ذکر شده، معیار عملکرد از $PM = ۱۷۰$ (شکل ۶الف) به $PM = ۷۰$ (شکل ۷ب) بهبود می‌یابد.

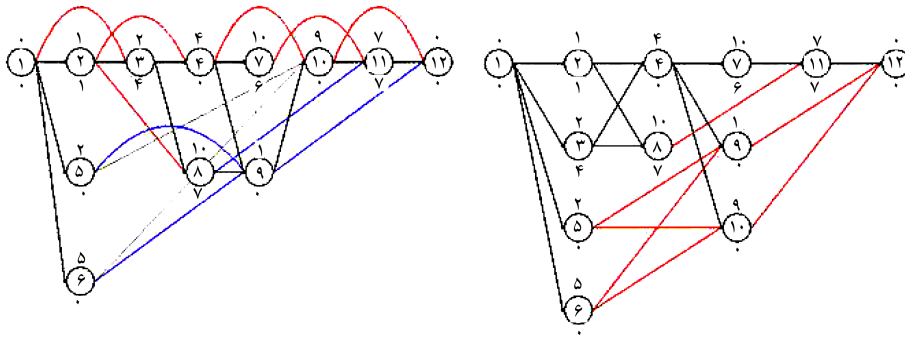
۲.۴. تنظیم پارامترها

۱.۲.۴. تعداد پیش‌نیازی‌های اضافه

بیشینه تعداد پیش‌نیازی‌های اضافی $Max.p$ ، در الگوریتم پیشنهادی به صورت تابعی از تعداد فعالیت‌ها n ، ضریب فشردگی زمان تحویل λ و نیز ضریب پیچیدگی شبکه OS ، براساس رابطه زیر تعیین می‌شود، c_1 ضریبی است که در تنظیم پارامترها مناسب‌ترین مقدار آن تعیین می‌شود.

$$Max.p = c_1 \times [10\lambda - 9] \times \left[\frac{n-2}{10} \right] \times \left[\frac{1.5}{OS} - 1 \right]$$

زمان تحویل δ_n براساس رابطه‌ی $\delta_n = \lambda ES_n$ تعیین می‌شود، که در آن λ ضریب فشردگی زمان تحویل و ES_n زودترین زمان پایان پروژه است. انتظار می‌رود با



الف) بدون در نظر گرفتن روابط پیش نیازی اضافه؛
 ب) با در نظر گرفتن روابط پیش نیازی اضافه.

شکل ۴. شبکه‌ی فعالیت‌ها.

	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۲		۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰
۳			۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۴				۰	۰	۰	۰	۰	۰	۰	۰	۰
۵					۰	۰	۰	۰	۰	۰	۰	۰
۶						۰	۰	۰	۰	۰	۰	۰
۷							۰	۰	۰	۰	۰	۰
۸								۰	۱	۰	۰	۰
۹									۰	۱	۰	۰
۱۰										۰	۱	۰
۱۱											۰	۱
۱۲												۰

ج) ماتریس روابط پیش نیازی موقت $A_{n \times n}^{temp}$

	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
۱	۰	۱	۱	۰	۱	۱	۰	۰	۰	۰	۰	۰
۲		۰	۰	۱	۰	۰	۰	۱	۰	۰	۰	۰
۳			۰	۱	۰	۰	۰	۱	۰	۰	۰	۰
۴				۰	۰	۰	۱	۰	۱	۱	۰	۰
۵					۰	۰	۰	۰	۱	۱	۰	۰
۶						۰	۰	۰	۱	۱	۰	۰
۷							۰	۰	۰	۰	۰	۰
۸								۰	۰	۱	۰	۰
۹									۰	۰	۰	۱
۱۰										۰	۰	۱
۱۱											۰	۱
۱۲												۰

الف) ماتریس روابط پیش نیازی اولیه (دائمی)؛

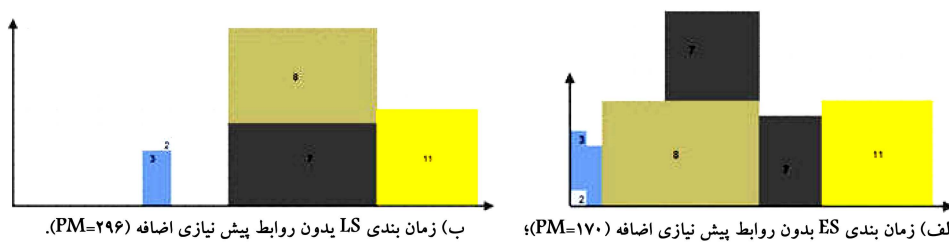
	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
۱	۰	۱	۱	۰	۱	۱	۰	۰	۰	۰	۰	۰
۲		۰	۱	۱	۰	۰	۰	۱	۰	۰	۰	۰
۳			۰	۱	۰	۰	۰	۱	۰	۰	۰	۰
۴				۰	۰	۰	۱	۰	۱	۱	۰	۰
۵					۰	۰	۰	۰	۱	۱	۰	۰
۶						۰	۰	۰	۱	۱	۰	۰
۷							۰	۰	۰	۰	۱	۰
۸								۰	۱	۰	۱	۰
۹									۰	۱	۰	۱
۱۰										۰	۱	۱
۱۱											۰	۱
۱۲												۰

د) ماتریس روابط پیش نیازی جدید $A_{n \times n}^{eff}$.

	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
۱	۰	۰	۰	۲	۰	۰	۶	۲	۶	۶	۱۶	۲۳
۲		۰	$-\infty$	۱	$-\infty$	$-\infty$	۵	۱	۵	۵	۱۵	۲۲
۳			۰	۲	$-\infty$	$-\infty$	۶	۲	۶	۶	۱۶	۲۳
۴				۰	$-\infty$	$-\infty$	۴	$-\infty$	۴	۴	۱۴	۲۱
۵					۰	$-\infty$	$-\infty$	$-\infty$	۲	۲	$-\infty$	۱۱
۶						۰	$-\infty$	$-\infty$	۵	۵	$-\infty$	۱۴
۷							۰	$-\infty$	$-\infty$	$-\infty$	۱۰	۱۷
۸								۰	$-\infty$	$-\infty$	۱۰	۱۷
۹									۰	$-\infty$	$-\infty$	۱
۱۰										۰	$-\infty$	۹
۱۱											۰	۷
۱۲												۰

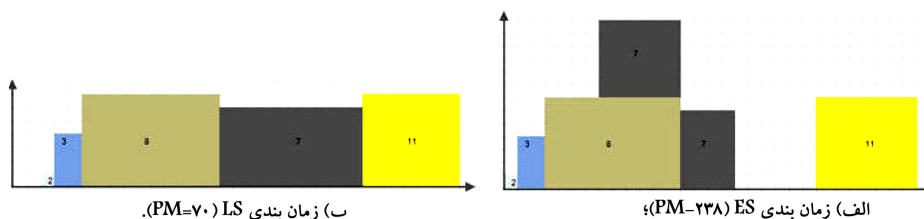
ب) ماتریس طولانی‌ترین مسیر $A_{n \times n}^{lp}$ ؛

شکل ۵. ماتریس‌های مورد استفاده در نمایش جواب.



الف) زمان بندی ES بدون روابط پیش نیازی اضافه (PM=۱۷۰)؛
 ب) زمان بندی LS بدون روابط پیش نیازی اضافه (PM=۲۹۶).

شکل ۶. زمان بندی بدون روابط پیش نیازی اضافه.



شکل ۷. زمان بندی با روابط پیش‌نیازی اضافه.

جدول ۳. مقادیر پارامترها.

۳۲	۲۲	۱۲	تعداد فعالیت‌ها
۱/۵	۳	۴	$c_1 \in \{0.5, 1, \dots, 5\}$
0.82	0.84	0.96	$\alpha \in \{0.80, 0.82, \dots, 0.98\}$
۲	۲	۲	$c_2 \in \{1, 2, \dots, 10\}$
۲۵۰۰	۱۵۰۰	۵۰۰	شرط توقف

الگوریتم براساس تعداد کل همسایگی‌های بررسی شده تعریف می‌شود که در قسمت تنظیم پارامترها مقدار مناسب آن با توجه به تعداد فعالیت‌ها تعیین شده است.

۵.۲.۴. تنظیم پارامترها

پارامترهای الگوریتم، براساس آزمایشات عملی کامل صورت گرفته روی مجموعه‌ی ۴۸۰ مسئله‌ی آزمایشی (متشکل از مسائلی با ۱۲، ۲۲ و ۳۲ فعالیت با تعداد ۱ الی ۴ منبع) تعیین شده است. نتایج حاصل از اجرای الگوریتم روی مجموعه مسائل آزمایشی در سطوح مختلف در جدول ۳ خلاصه شده است. مقادیر تنظیم شده برای پارامترها براساس تعداد فعالیت‌های شبکه ارائه شده است.

۵. نتایج محاسباتی

جواب‌های الگوریتم شبیه‌سازی تیرید ابتدا با جواب‌های مدل ریاضی صفر و ۱ که توسط یکی از نرم‌افزارهای حل مدل‌های ریاضی حل شده، مقایسه شده است و سپس با جواب‌های سه روش ابتکاری مقایسه صورت گرفته است. کلیه‌ی الگوریتم‌ها با ++C کد شده‌اند و توسط رایانه‌ی بی با مشخصات ۲٫۳۰GHz، ۴GB RAM، @ ۱۵-۲۴۱۰M CPU اجرا شده‌اند و نیز از نسخه‌ی پیشرفته ۹٫۷ LINGO برای حل مدل‌های ریاضی استفاده شده است.

۱.۱.۵. مقایسه‌ی حل شبیه‌سازی تیرید با جواب‌های حاصل از حل

لینگو (مدل برنامه‌ریزی صفر و ۱)

ابتدا یک مدل برنامه‌ریزی صفر و ۱ غیرخطی برای مسئله‌ی تسطیح منابع ارائه و سپس مدل ریاضی برای مجموعه‌ی مسائل آزمایشی حل می‌شود.

۱.۱.۵. مدل برنامه‌ریزی صفر و ۱

در این مدل متغیر صفر و ۱ x_{it} ، نشان‌دهنده‌ی آن است که آیا فعالیت i در زمان t شروع می‌شود یا خیر. اگر فعالیت i در زمان t شروع شود، $x_{it} = 1$ است و در غیر این صورت $x_{it} = 0$ است. در این مدل مقدار مجموع (موزون) مربع نوسانات

اضافه کردن پیش‌نیازی‌های اضافه، زمان ختم پروژه و در نتیجه احتمال غیر موجه شدن جواب، افزایش یابد. بنابراین λ با $Max.p$ رابطه‌ی مستقیم دارد. تعداد پیش‌نیازی‌ها در شبکه‌های دارای OS بالا به خودی خود زیاد است و گزینه‌ها برای افزودن پیش‌نیازی‌های جدیدتر کم‌تر است. همچنین در شبکه‌های دارای OS بالا تعداد مسیرهای بحرانی بیشتر است و با اضافه کردن پیش‌نیازی‌های اضافه احتمال غیر موجه شدن جواب بیشتر خواهد بود. به همین دلیل در رابطه‌ی فوق $Max.p$ رابطه‌ی معکوس دارد با OS ؛ به‌عنوان مثال، بیشترین تعداد روابط پیش‌نیازی اضافه همزمان در مسئله‌ی با $n = 12$ و $\lambda = 1.0$ و $OS = 0.75$ براساس رابطه‌ی فوق برابر خواهد بود با $c_1 = Max.p$ و در مسئله‌ی با $n = 32$ و $\lambda = 1.5$ و $OS = 0.5$ بیشترین تعداد پیش‌نیازی‌های اضافه برابر $Max.p = 36c_1$ خواهد بود. به این معنا که در هر بار ایجاد همسایگی، بیشینه $36c_1$ رابطه پیش‌نیازی اضافه به صورت تصادفی ایجاد و به پیش‌نیازی‌های مسئله‌ی اصلی اضافه می‌شود.

۲.۲.۴. پیش‌پردازش

برای جلوگیری از غیر موجه شدن جواب‌ها، روابط اضافی را، که به‌تنهایی زمان اتمام پروژه را آن‌قدر افزایش می‌دهند که جواب غیر موجه می‌شود، با یک لیست ممنوع دائمی کنترل می‌کنیم. لیست ممنوع شامل روابط پیش‌نیازی اضافه‌ی است که به تنهایی منجر به غیر موجه شدن جواب می‌شود. الگوریتم شبیه‌سازی تیرید در پیش‌پردازش، روابط پیش‌نیازی اضافی ممنوع را تعیین، و آنها را از لیست گزینه‌های ممکن برای ایجاد همسایگی حذف می‌کند. پس از حذف گزینه‌های ممنوع، تعداد کل گزینه‌های ممکن برای پیش‌نیازی‌های اضافه را با np نمایش می‌دهیم.

۳.۲.۴. ایجاد همسایگی

در هر بار ایجاد همسایگی، مقدار تصادفی nn (عدد صحیح بین ۱ تا $Max.p$) تعیین می‌شود. از میان np پیش‌نیازی موجود در ماتریس $A_{n \times n}^{temp}$ (که می‌تواند مقادیر صفر یا ۱ را داشته باشند)، nn گزینه به تصادف انتخاب می‌شوند. هر یک از گزینه‌های انتخاب شده به‌صورت تصادفی به مقدار متمم‌اش تغییر می‌کند. احتمال این تغییر در هر بار ایجاد همسایگی، براساس آزمایشات انجام گرفته برابر $p = 0.1$ تعیین شده است.

۴.۲.۴. دمای اولیه، کاهش دما و شرط توقف

دمای اولیه به‌گونه‌ی تعیین شده که اولین همسایگی‌ها با احتمال نسبتاً بالایی پذیرفته می‌شود. T_0 به‌صورت ضربی از مقدار تابع هدف اولین جواب تصادفی sf_{nt} تعیین می‌شود: $T_0 = c_2 \times sf_{nt}$. مناسب‌ترین مقدار عددی برای ضریب c_2 نیز در تنظیم پارامترها تعیین می‌شود.

در الگوریتم شبیه‌سازی تیرید پیشنهادی، دمای هر مرحله براساس دمای مرحله‌ی قبل با سازوکار کاهش دمای هندسی $T_i = \alpha \times T_{i-1}$ تعیین می‌شود. مقدار ضریب کاهش دمای α در قسمت تنظیم پارامترها تعیین می‌شود. شرط توقف

جدول ۴. تعداد مسائل آزمایشی با توجه به تعداد منابع و فعالیت‌ها.

	۱۲ فعالیت	۲۲ فعالیت	۳۲ فعالیت	جمع
۱ منبع	۲۰	۱۰	۵	۳۵
۲ منبع	۱۶	۸	۴	۲۸
۳ منبع	۱۲	۶	۳	۲۱
۴ منبع	۸	۴	۲	۱۴
جمع	۵۶	۲۸	۱۴	۹۸

منابع در دوره‌های متوالی کمینه می‌شود.

$$\min \sum_{k=1}^m \sum_{t=0}^{\delta_n} C_k \left(\sum_{i=1}^n r_{ik} \left(\sum_{q=\max\{t-d_i+1, EST_i\}}^{\min\{t, LST_i\}} x_{iq} - \sum_{q=\max\{t-d_i, EST_i\}}^{\min\{t-1, LST_i\}} x_{iq} \right) \right)^2$$

st :

$$\sum_{t=EST_i}^{LST_i} x_{it} = 1 \quad i = 1, 2, \dots, n$$

$$\sum_{t=EST_i}^{LST_i} t.x_{it} + d_i \leq \sum_{t=EST_j}^{LST_j} t.x_{jt} \quad (i, j) \in A$$

$$x_{it} \in \{0, 1\}, \quad i = 1, 2, \dots, n \quad t = EST_i, \dots, LST_i$$

عبارت $\sum_{q=\max\{t-d_i+1, EST_i\}}^{\min\{t, LST_i\}} x_{iq}$ که مقدار صفر یا ۱ اخذ می‌کند نشان می‌دهد که آیا فعالیت i در زمان t که $t = 0, \dots, \delta_n$ فعال است یا خیر. توجه داریم که اگر فعالیت i در زمان t آغاز شود، این فعالیت در بازه زمانی $(t, t + d_i - 1)$ فعال (در حال انجام) خواهد بود؛ به عبارت دیگر فعالیت i زمانی در بازه t فعال است که شروع آن در فاصله‌ی $1 + t - d_i$ تا $q = t$ واقع شود. با لحاظ کردن زودترین و دیرترین زمان شروع فعالیت i حدود بالا و پایین مجموع فوق $q = \max\{t - d_i + 1, LST_i\}$ تا $q = \min\{t, LST_i\}$ خواهد بود و در نتیجه $\sum_{i=1}^n r_{ik} x_{iq} = u_{kt}$ نشان‌دهنده‌ی میزان منبع k ام مورد استفاده در دوره زمانی t ام است. محدودیت اول بیان می‌دارد که یک فعالیت تنها یک بار شروع می‌شود و محدودیت دوم مربوط به پیش‌نیازهاست. در

محدودیت‌های ذکر شده، عبارت $\sum_{t=EST_i}^{LST_i} t.x_{it}$ نشان دهنده‌ی زمان شروع فعالیت i است.

به منظور مقایسه، جواب‌های حاصل از حل ۹۸ مسئله با نرم‌افزار لینگو و الگوریتم شبیه‌سازی تبرید با یکدیگر مقایسه شده است. ویژگی ۹۸ مسئله‌ی مورد بررسی با جزئیات بیشتر در جدول ۴ به نمایش درآمده است. همچنین خلاصه‌ی نتایج حاصل از مقایسه‌ی دو الگوریتم در جدول ۵ آورده شده است. با توجه به غیرخطی بودن تابع هدف و زیاد بود تعداد متغیرهای صفر و ۱، محدودیتی از نظر مدت زمان حل برای لینگو در نظر گرفته نشده است. در عمده‌ی این مسائل نرم‌افزار لینگو پس از اتمام عملیات حل گزارشی مبتنی بر حصول جوابی محلی، بهترین جواب حاصله در مدت حل و شرط توقف داخلی خود را گزارش کرده است.

با بررسی این دو جدول می‌توان بیان کرد که زمان حل الگوریتم شبیه‌سازی تبرید به مراتب کم‌تر از زمان حل لینگو است. این تفاوت آن‌قدر بالاست که عملکرد بعضاً ضعیف شبیه‌سازی تبرید در بعضی مسائل قابل چشم‌پوشی باشد. با افزایش تعداد فعالیت‌ها عملکرد شبیه‌سازی تبرید نسبت به لینگو بهتر می‌شود، به گونه‌ی که درصد جواب‌های بهتر این الگوریتم از ۱۶٪ در $n = ۱۲$ به ۵۷٪ در $n = ۳۲$ افزایش می‌یابد. با توجه به عملکرد بهتر شبیه‌سازی تبرید در مسائل با حجم بالا (چه از نظر معیار عملکرد تسطیح و چه از نظر زمانی) و نیز با توجه با این که در مسائل با حجم کم، جواب شبیه‌سازی تبرید به‌طور نسبی قابل رقابت با لینگو است، کارایی بالاتر الگوریتم شبیه‌سازی تبرید قابل پذیرش است. کارایی الگوریتم شبیه‌سازی تبرید همچنین در قسمت بعد از طریق مقایسه با الگوریتم‌های ابتکاری مورد بررسی قرار گرفته است.

۲.۵. مقایسه‌ی الگوریتم‌های ابتکاری

سه الگوریتم ابتکاری بررسی شده است: ۱. الگوریتم برگس و کیلبرو^[۱۳] که به صورت تکرار شونده و بازگشتی و براساس یک لیست اولویت به زمان‌بندی فعالیت‌ها می‌پردازد (BK)؛ ۲. الگوریتم ساده و سریع زمان‌بندی در زودترین زمان ممکن (ES)؛ ۳. در نهایت الگوریتم زمان‌بندی در دیرترین زمان ممکن (LS).

۱.۲.۵. تولید مسائل آزمایشی

در تولید مسائل آزمایشی در حوزه‌ی زمان‌بندی پروژه الگوریتم‌های متعددی معرفی شده که براساس شاخص‌های مختلفی براساس زمان، شبکه، و نیازمندی‌های منبع یا به صورت ترکیبی به تولید مسائل آزمایشی می‌پردازند. در این تحقیق از الگوریتم RanGen^[۱۲] استفاده شده است. نرم‌افزار RanGen تولیدکننده‌ی شبکه‌های AON است که به صورت تصادفی شبکه‌هایی با شاخص‌های دلخواه تولید می‌کند.

جدول ۵. خلاصه نتایج حل مسائل توسط شبیه‌سازی تبرید و لینگو.

تعداد فعالیت‌ها	۱۲	۲۲	۳۲	کل
متوسط معیار عملکرد SA	۱۷۸۴٫۰۹	۲۸۱۴٫۴۶	۴۵۳۶٫۴۳	۲۴۷۱٫۶۷
متوسط معیار عملکرد Lingo	۱۷۳۹٫۵۷	۲۷۵۶٫۰۴	۴۶۴۰٫۵۰	۲۴۴۴٫۴۱
تعداد جواب‌های بهتر SA	۹	۱۰	۸	۲۷
تعداد جواب‌های بهتر Lingo	۲۴	۱۳	۳	۴۰
تعداد جواب‌های برابر	۲۳	۵	۳	۳۱
متوسط زمان Lingo (ms)	۴۰۱۰۸۲۱٫۷۱	۱۷۴۶۶۰۴۵٫۴۳	۵۳۰۸۴۹۲۱٫۱۴	۱۴۸۶۵۷۵۶٫۹۸
متوسط زمان SA (ms)	۰٫۹۷	۱۳٫۶۲	۶۶٫۸۵	۱۳٫۹۹
درصد جواب‌های بهتر SA	۱۶٫۱	۳۵٫۷	۵۷٫۱	۲۷٫۶

مقادیر در نظر گرفته شده برای شاخص‌ها در جدول ۶ آمده است. در این جدول n نشان‌دهنده‌ی تعداد فعالیت‌ها، OS نشان‌دهنده‌ی ضریب پیچیدگی شبکه است که نزدیک بودن آن به ۱ باعث می‌شود شبکه به حالت سری نزدیک‌تر شود، و RF نشان‌دهنده‌ی ضریب منابع است و نزدیک بودن آن به مقدار ۱ بیان می‌دارد که فعالیت‌ها نیازمند انواع بیشتری از منابع موجود هستند و RC نشان‌دهنده‌ی ضریب محدودیت منابع است. بیشتر بودن این شاخص بیان‌گر این است که فعالیت‌ها مقدار بیشتری از منابع موجود را طلب می‌کنند. همچنین m تعداد انواع منابع را نمایش می‌دهد. تعداد لیست‌های اولویت مورد بررسی در الگوریتم برگس $nlist$ را مناسب با اندازه شبکه تعریف کرده‌ایم بدین صورت که $nlist = 20$ برای $n = 12$ و $nlist = 50$ برای $n = 22$ و $nlist = 100$ برای $n = 30$ در نظر گرفته شده است. برای هر یک از ترکیب‌های فوق ۱۰ مسئله‌ی آزمایشی ایجاد شده که در مجموع $960 = 10 \times 2^2 \times 3 \times 4$ مسئله‌ی متنوع در مجموعه مسائل آزمایشی ایجاد شده است.

با توجه به نتایج نمایش داده شده در جدول ۷ و شکل ۸، ملاحظه می‌شود الگوریتم شبیه‌سازی تیرید نسبت به روش برگس (چه به لحاظ زمان حل و چه از نظر مقدار جواب) برتر است. همچنین این روش از نظر مقدار جواب (معیار عملکرد)

جدول ۶. شاخص‌های مربوط به مسائل آزمایشی.

شاخص	تعداد حالت‌ها	مقادیر
N	۳	{۱۲, ۲۲, ۳۲}
OS	۲	{۰٫۵, ۰٫۷۵}
RF	۲	{۰٫۵, ۰٫۷۵}
RC	۲	{۰٫۵, ۰٫۷۵}
m	۴	{۱, ۲, ۳, ۴}

۶. نتیجه‌گیری و پیشنهادات

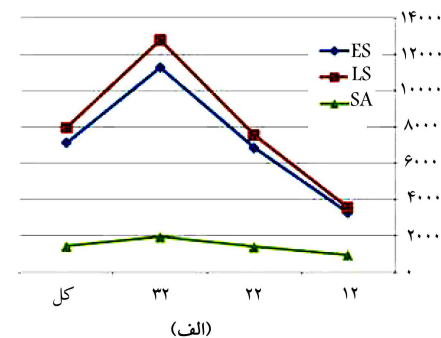
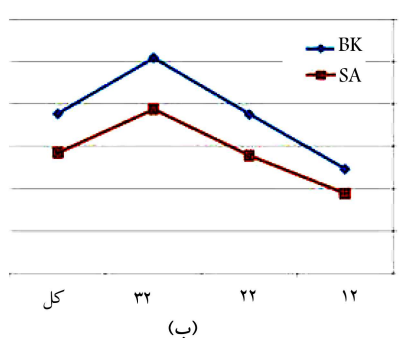
در مسئله‌ی تسطیح منابع سعی بر آن است که فعالیت‌ها به‌گونه‌ی زمان‌بندی شود که پروفایل منابع، که نشان‌دهنده‌ی میزان استفاده از منابع تجدیدپذیر در طی زمان است، تا حد ممکن مسطح باشد. در این نوشتار ضمن مرور کامل ادبیات موضوع، این نتیجه حاصل شد که روش‌های ابتکاری که کارایی آن‌ها بررسی شده باشد برای مسئله‌ی تسطیح منابع به‌ندرت یافت می‌شود.

در این نوشتار با استفاده از شبیه‌سازی تیرید اقدام به حل این مسئله می‌کنیم. برای نمایش جواب از بردارهای صفر و ۱، که به‌گونه‌ی مؤثر پیش‌نیازهای موقت ایجاد می‌کنند، استفاده کرده‌ایم. براساس ویژگی‌های اثبات شده، معیار عملکرد را از طریق بردارهای پیش‌نیاز جدید و زمان‌بندی در زودترین زمان و در دیرترین زمان بهبود می‌دهیم.

کارایی الگوریتم شبیه‌سازی تیرید طی مقایسه‌ی جواب‌های مدل ریاضی صفر و ۱ و نیز با سه روش ابتکاری ES ، LS و BK مورد سنجش قرار گرفته است. برای بررسی، مجموعه‌ی متنوعی از مسائل آزمایشی تولید شده است. نتایج محاسباتی

جدول ۷. معیار عملکرد و زمان الگوریتم شبیه‌سازی تیرید و سه الگوریتم ابتکاری.

الگوریتم	۱۲ فعالیت		۲۲ فعالیت		۳۲ فعالیت		کل
	معیار عملکرد	زمان	معیار عملکرد	زمان	معیار عملکرد	زمان	
ES	۳۲۷۳	۰٫۰۰	۶۸۳۹٫۱	۰٫۰۰	۱۱۲۶۹٫۴	۰٫۰۰	۷۱۲۷٫۲
LS	۳۵۳۱٫۶	۰٫۰۰	۷۵۲۶٫۹	۰٫۰۰	۱۲۷۷۴٫۶	۰٫۰۰	۷۹۴۴٫۴
BK	۱۲۳۲٫۲	۰٫۴۸	۱۸۷۹٫۱	۸٫۴۹	۲۵۴۸٫۵	۴۹٫۶۷	۱۸۸۶٫۶
SA	۹۳۸٫۵	۰٫۰۹	۱۳۸۶٫۷	۱٫۳۳	۱۹۳۷٫۹	۶٫۶۵	۱۴۲۱



شکل ۸. معیار عملکرد الگوریتم شبیه‌سازی تیرید و سه الگوریتم ابتکاری.

زمانی و نیز امکان وجود محدودیت منابع، مدل‌ها را به مسائل دنیای واقعی نزدیک‌تر می‌کند و همچنین بررسی کارایی بسیاری از الگوریتم‌های فراابتکاری در حل مسئله‌ی تسطیح منابع روی مجموعه وسیعی از مسائل آزمایشی، از جمله نقص‌های ادبیات موضوع است.

نشان می‌دهد، الگوریتم ارائه شده نسبت به سه روش ابتکاری برتری دارد و نتایج قابل رقابتی را با جواب‌های مدل ریاضی به دست می‌دهد. به‌عنوان پیشنهاد برای تحقیقات آتی به نظر می‌رسد در نظر گرفتن مسئله‌ی تسطیح منابع با ویژگی‌هایی چون امکان شکست فعالیت‌ها، امکان وجود پنجره‌های

پانویس‌ها

1. Harris
2. Brinkmann
3. Leu
4. Georgios

منابع (References)

1. Demeulemeester, E.L. and Herroelen, W.S., *Project Scheduling—A Research Handbook*, Kluwer Academic Publishers, Boston (2002).
2. Kastora, A. and Sirakoulis, K. "The effectiveness of resource levelling tools for resource constraint project scheduling problem", *International Journal of Project Management*, **27**(5), pp. 493-500 (2009).
3. Hashemi Doulabi, H., Seifi, A. and Shariat, S. "Efficient hybrid genetic algorithm for resource leveling via activity splitting", *Journal of Construction Engineering and Management*, **137**(2), pp. 137-146 (2011).
4. Heon, J.D. and El-Rayes, K. "Multi objective optimization of resource leveling and allocation during construction scheduling", *Journal of Construction Engineering and Management*, **137**(12), pp. 1080-1088 (2011).
5. Hariga, M. and El-Sayegh, S. "Cost optimization model for the multi-resource leveling problem with allowed activity splitting", *Journal of Construction Engineering and Management*, **137**(1), pp. 56-64 (2011).
6. Rieck, J., Zimmermann, J. and Gather, T. "Mixed-integer linear programming for resource leveling problems", *European Journal of Operational Research*, **221**(1), pp. 27-37 (2012).
7. Takamoto, M., Yamada, N., Kobayashi, Y. and Nonaka, H. "Zero-one quadratic programming algorithm for resource leveling for manufacturing process schedules", *Systems and Computers in Japan*, **26**, pp. 68-76 (1995).
8. Gather, T., Zimmermann, J. and Bartels, J.H. "Exact methods for the resource levelling problem", *Journal of Scheduling*, **14**(6), pp. 557-569 (2011).
9. Ahuja, H.N., *Construction Performance Control by Networks*, John Wiley, New York (1976).
10. Younis, M.A. and Saad, B. "Optimal resource leveling of multi-resource projects", *Computers and Industrial Engineering*, **31**, pp. 1-4 (1996).
11. Easa, S.M. "Resource leveling in construction by optimization", *Journal of Construction Engineering and Management*, **115**, pp. 302-316 (1989).
12. Son, J. and Mattila, K. "Binary resource leveling model: Activity splitting allowed", *Journal of Construction Engineering and Management*, **130**(6), pp. 887-894 (2004).
13. Neumann, K., Schwindt, C. and Zimmermann, J., *Project Scheduling with Time Windows and Scarce Resources*, Springer, Berlin (2003).
14. Burgess, A.R. and Killebrew, J.B. "Variation in activity level on a cyclic arrow diagram", *The Journal of Industrial Engineering*, **13**, pp. 76-83 (1962).
15. Harris, R.B. "Packing method for resource leveling (pack)", *Journal of Construction Engineering and Management*, **116**, pp. 39-43 (1990).
16. Brinkmann, K. and Neumann, K. "Heuristic procedures for resource-constrained project scheduling with minimal and maximal time lags: The resource-levelling and minimum project-duration problems", *Journal of Decision Systems*, **5**, pp. 129-156 (1996).
17. Neumann, K. "Scheduling of projects with stochastic evolution structure", In: Weglarz, J. (Ed.), *Project Scheduling—Recent Models, Algorithms and Applications*, Kluwer Academic publishers, Boston, pp. 309-332 (1999).
18. Neumann, K. and Zimmermann, J. "Procedures for resource levelling and net present value problems in project scheduling with general temporal and resource constraints", *European Journal of Operational Research*, **127**, pp. 425-443 (2000).
19. Leu, S.S., Yang, C.H. and Huang, J.C. "Resource leveling in construction by genetic algorithm based optimization and its decision support application", *Automation in Construction*, **10**, pp. 27-41 (2000).
20. Leu, S.S., Chen, A.T. and Yang, C.H. "A fuzzy optimal model for construction resource leveling scheduling", *Canadian Journal of Civil Engineering*, **26**(6), pp. 673-684 (1999).
21. Georgios, K.K. and Konstantinos, P.A. "A simulated annealing hyperheuristic for construction resource leveling", *Construction Management and Economics*, **28**(2), pp. 163-175 (2010).
22. Demeulemeester, E., Vanhoucke, M. and Herroelen, W. "RanGen: A random network generator for activity-on-the-node networks", *Journal of Scheduling*, **6**, pp. 17-38 (2003).