مسئله سرمایه گذاری در کنترل پروژه

شهرام شادخ - دانشگاه صنعتی شریف

چکیده:

در این نوشته مسئله سرمایه گذاری در منابع در برنامه ریزی پروژه(RIP) مورد بررسی قرار می گیرد. این مسئله شکل خاص مسئله تخصیص منابع در کنترل پروژه است. معروفترین مسئله از این دست مسئله تخصیص منابع محدود می باشد که در این زمینه مقالات زیادی منتشر شده است. تاکنون در مورد مسئله RIP به دلیل شکل پیچیده آن کارهای زیادی انجام نشده است. از جمله کارهای صورت گرفته در این زمینه می توان به کار آکپان (۱۹۹۷) و یا کار کیمز (۲۰۰۱) اشاره نمود.

در این نوشته یک الگوریتم ژنتیک برای حل مسئله RIP ارائه شده و نتیجه آن با کارهای انجام گرفته مقایسه شده است. برای انجام مقایسه از ۲۸۸۰ مسئله استاندارد که در سایت PSBLIB دانشگاه کیل المان مبنای مقایسه الگوریتمهای ارائه شده در زمینه مسائل تخصیص منابع در کنترل پروژه می باشد استفاده شده است.

در الگوریتم ژنتیک ارائه شده از روشهای جستجوی موضعی برای بهبود کارایی روش استفاده گردیده است. نتایج نشان می دهد که تقریبا در تمامی مسائل آزمایش شده الگوریتم ژنتیک از روشهای موجود بهتر عمل می کند.

کلمات کلیدی:

کنترل پروژه، سرمایه گذاری در منابع، الگوریتم ژنتیک

۱) مقدمه :

مسائل برنامه ریزی دارای دامنه گسترده أی میباشند و مسئله تخصیص منابع محدود و نامحدود دربرنامه ریزی وکنترل پروژه نوعی از مسئله برنامه ریزی است که خود دامنه گسترده أی از مسائل دیگر بر نامه ریزی را نیز در بر می گیرد. برای مثال میتوان نشان داد که مسئله (Job- Shop) حالت خاصی از مسئله تخصیص منابع دربرنامه ریزی وکنترل پروژه میباشد[Baker] .

در حالیکه در روشهای سنتی نظیر CPM فرض بر این است که منابع بصورت نامحدود وجود دارند در دیدگاههای جدید سعی بر اینستکه فرض منطقی منابع محدود در نظر گرفته شود. در مورد اولین کارها در این زمینه میتوان از کار کلی[Kelly] نام برد که در آن برنامه ریزی پروژه با منابع محدود (RCPSP)[۱] معرفی و بررسی شده است.

بطور کلی در مسائل برنامه ریزی دو نوع محدودیت میتواند وجود داشته باشد، یکی محدودیت تقدم و تاخر ودیگری محدودیت منابع، که هریک از آنها به نوبه خود میتوانند شکلهای مختلفی داشته باشند. در بسیاری از تحقیقات انجام شده ومدلهای بوجود آمده برای سادگی کار یا یکی از این دو نوع محدودیت حذف ویا ساده شده است[Cheng]. در مسئله تخصیص منابع محدود در کنترل پروژه ساختار مسئله طوری است که هر دو نوع محدودیت فوق باید در نظر گرفته شود واین مطلب باعث میشود که این نوع مسئله برنامه ریزی نسبت به بقیه مسائل برنامه ریزی مشکل تر شود.

حالت کلی مسئله تخصیص منابع در کنترل پروژه عبارت است از:

تعدادی فعالیت که در آنها محدودیت تقدم وتاخر باید رعایت شود و فعالیتها میتوانند به روشهای مختلف انجام گیرند که هر روش انجام برای یک فعالیت زمان خاص خود و مقدار لازم استفاده از منابع خاص خود را می طلبد، (فرضیات مختلفی میتواند روی منابع یا محدودیتهای تقدم وتاخر وجود

[۱] Resource Constraint Project Sscheduling Problem

داشته باشد.) و مسئله عبارت از پیدا کردن یک برنامه تخصیص منابع به فعالیتها است طوریکه محدودیتهای منابع وفعالیتها برآورده شود و تابع هدف مورد نظر بهینه گردد.

با توجه به نتایج بدست آمده توسط بلزویک[Blazewics] به سادگی میتوان نشان داد که مسئله تخصیص منابع در کنترل پروژه از نوع NP-hard است. بنابراین برای مسائل بزرگ (در اینجا شاید حتی تا ۱۰۰ فعالیت و ۲ تا ۳ محدودیت) الگوریتمهای بهینه ساز در زمان معقول قادر به حل مسئله نمیباشند، وحتی روشهایی که فقط یک جواب قابل قبول را می دهند مفید محسوب میشوند. به همین دلیل در سالهای اخیر دامنه کاربرد روشهای هیورستیک در حل این نوع از مسائل روبه افزایش است. در ۳۰ سال اخیر دامنه گسترده أی از روشهای هیورستیک ایجاد شده ومورد آزمایش قرار گرفته است. بیشتر این روشها براساس اولویت دهی به فعالیتها باتوجه به قوانین مختلف اولویت[2] دهی شکل گرفته است.

بدلیل ساختار پیچیده مسئله و اینکه بعضی از مسائل برنامه ریزی پروژه در قالب مسائل ترکیبی[3] می گنجند، در سالهای اخیر تمایل محققین برای استفاده از الگوریتم ژنتیک افزایش یافته است. الگوریتم ژنتیک یک روش جستجوی تصادفی است که در سال ۱۹۷۰ در آمریکا توسط هلند[Holland] ارائه شد. اساس این الگوریتم فرآیند تحول طبیعی میباشد. در الگوریتم ژنتیک افراد جامعه جوابهای مسئله میباشند که بطور مناسب کد بندی شده اند. این جامعه جوابها با استفاده از عملگرهای تحول[4] طوری هدایت می شود که فرد مناسب تری در جامعه بوجود آید.

در زمینه استفاده از الگوریتم ژنتیک در برنامه ریزی و تخصیص منابع نتایج خوبی تجربه شده است. برای مثال میتوان به الگوریتم ژنتیک هارتمن[Hartmmen] اشاره نمود که برای مسئله تخصیص منابع محدود در حالتیکه برای هر فعالیت چند روش اجرا میتواند وجود داشته باشد طرح شده است. او با استفاده از یک طرح مناسب در پارامتر های الگوریتم ژنتیک و با استفاده از یک روش جستجوی موضعی نتایجی را بدست آورد که در مورد این نوع مسئله روش او از سایر روشها پیشی گرفت[Hartmann & Kolisch].

۲) مسئله سرمایه گذاری در منابع (RIP):

n فعالیت باید انجام شوند، فعالیتهای شماره 0 و n+1 طوری مجازی در نظر گرفته می شوند که تنها فعالیت شروع 0 باشد و تنها فعالیت ختم n+1 باشد. مجموعهٔ این فعالیتها را با V نمایش می دهیم در اینصورت مجموعهٔ V را دارای n+2 عضو در نظر می گیریم. همچنین فرضیات زیر را نیز درنظر می گیریم:

۱) فعالیتها قابل شکسته شدن نیستند.

۲) کلیه پارامترهای مدل از نوع قطعی می باشند و در مورد هیچ پارامتری تصادفی بودن وجود ندارد.

۳) مقدار سطوح منابع در طول پروژه ثابت است.

۴) مدت زمان انجام پروژه از پیش تعیین شده است و آنرا T می نامیم.

سایر پارا متر ها به قرار زیر است:

K : مجموعهٔ منابع قابل تجدید.

ρ : تعداد اعضای مجموعه منابع قابل تجدید. $n(K) = \rho$

S_j : زمان شروع فعالیت j.

$S = (S_0,...,S_{n+1})$: بردار زمانهای شروع فعالیتها.

C_k: هزینه هر واحد منبع قابل تجدید kام $k \in K$

R_k: سطح منبع قابل تجدید kام $k \in K$

P_j: مجموعهٔ فعالیتهای پیش نیاز فعالیت j

r_{kj}: مقداری از منبع قابل تجدید k که توسط فعالیت j در هر واحد زمان استفاده شود.

y_{it}: مقدار ۱ را می گیرد اگر فعالیت i در زمان t شروع شود و در غیر اینصورت مقدار صفر را می گیرد. $t \in \{0,...,T\}$

درنتیجه مدل زیر را خواهیم داشت:

---

[2] Periority Rules
[3] Combinatorial
[4] Evolutionary Operations

$$\text{Min} \sum_{k=1}^{\rho} C_k R_k$$

$$S_0 = 0$$
$$S_{n+1} \le T$$
$$\sum_{\forall t} y_{it} = 1$$
$$\sum_{\forall t} t \times y_{jt} \ge D_i + \sum_{\forall t} t \times y_{it} \quad \forall i \in P_j$$
$$\sum_{i} \sum_{u=t-D_i+1}^{t} r_{ki} \times y_{iu} \le R_k \quad \forall t$$
$$y_{it} \in \{0,1\} \quad i \in \{1,..,n\} \ \forall t$$
$$R_k \ge 0 \ k \in K$$

در حالتی که تأخیر زمانی وجود ندارد می توان به الگوریتم آکپان[Akpan, E. O. P] اشاره نمود. الگوریتم آکپان بر اساس روش کمترین دیرترین زمان اتمام[۵]، فعالیتها را برنامه ریزی می کند. اصول کار این روش اینستکه ابتدا فعالیتها را در زودترین زمان برنامه ریزی می کند و سطح منابع را در این حالت در حالت ماکزیمم در نظر می گیرد. سپس با تعریف یک معیار کارآیی برای منابع سعی بر این دارد که منبعی را کاهش دهد که اولاً بیشترین مقدار را از هزینه کاهش دهد و ثانیاً کمترین افزایش را در کل طول پروژه ایجاد کند. منبع انتخاب شده را یک واحد کاهش می دهد و فعالیتها را براساس روش دیرترین زمان اتمام برنامه ریزی می کند. در روش دیرترین زمان اتمام بر اسا س حداکثر زمان اتمام پروژه دیرترین زمان اتمام(یا دیرترین زمان شروع) هر فعالیت بدست می آید، سپس کلیه فعالیتهایی که پیشنیازهای آنها برنامه ریزی شده اند در نظر گرفته می شوند[۶] و بر اساس یک معیار از بین آنها یک فعالیت برای برنامه ریزی انتخاب می شود. در اینجا می توان از روشهای سری[۷] و موازی[۸] برای ایجاد برنامه (SGS) استفاده نمود. سپس دوباره لیست فعالیتهایی که کلیه پیش نیاز های آنها رعایت شده است در نظر گرفته می شود و از لیست فوق فعالیتی برای برنامه ریزی انتخاب می شود، الگوریتم تا جایی ادامه می یابد که تمام فعالیتها برنامه ریزی شده باشند. الگوریتم هنگامی پایان می یابد که با کاهش هریک از منابع دیگر نتوان پروژه را در زمان موردنظر پایان داد.

در واقع این روش در هر دور از اجرای الگوریتم یک مسئله تخصیص منابع محدود(RCPSP) ساده را حل می کند و برای این منظور از روش کمترین دیرترین زمان اتمام استفاده می کند.در این روش واضح است که حتی هنگامی که الگوریتم پایان می یابد شاید بتوان با تغییر زمانبندی فعالیتها هزینه ها را کاهش داد.

آسمند بعنوان پایان نامه دوره فوق لیسانس خود الگوریتم آکپان را با معیارهای مختلفی برای انتخاب منبعی که باید کاهش یابد آزمایش نمود و این معیارهای مختلف را بایکدیگر مقایسه کرد.

۳) طرح یک الگوریتم ژنتیک برای مسئله RIP:

۳-۱) کلیات مدل:

برای دستیابی به یک الگوریتم ژنتیک مناسب یکی از مهمترین اقدامات طرح یک کروموزوم مناسب است (ژنوتایپ). و سپس استخراج جواب مسئله از این کروموزوم (فنوتایپ). در مسئله RIP باید زمان شروع هریک از فعالیتها و همینطور سطح هریک از منابع تعیین گردد. بنابراین علاوه بر اینکه مانند RCPSP نیاز به تعیین زمان اجرای هر فعالیت می باشد متغیرهای تصمیم جدید دیگری که همان سطح منابع می باشند به مسئله اضافه شده اند. در مدل در نظر گرفته شده سطح منابع می تواند اعداد حقیقی باشد.

[۵] Least Latest Finish Time.
[۶] Eligible Activity Set
[۷] Parallel Schedule Generation Scheme.
[۸] Serial Schedule Generation Scheme.

همچنین در مورد الگوریتم ژنتیک تجربه نشان داده است که اگر این الگوریتم همراه با یک روش جستجو و به سازی همراه باشد کارایی بیشتری خواهد داشت. در الگوریتم ارائه شده دو روش جستجو ی موضعی[9] در نظر گرفته است. در الگوریتم ژنتیکی که ارائه می شود می توان تعداد نسل های موردنیاز برای تولید را انتخاب نمود.

نسل اولیه بصورت تصادفی ایجاد می‌گردد و مقدار تطابق هریک از اعضای آن محاسبه می گردد.سپس این نسل به صورت زوج زوج جدا شده و بین هر دو زوج با احتمال P_{cr} عمل تقاطع صورت می گیرد و از هر عمل تقاطع دو کروموزوم جدید ایجاد می گردد. تطابق این کروموزوم ها محاسبه می شود و کروموزوم های جدید ایجاد شده و کروموزوم های والد آنها برای انتخاب با هم در نظر گرفته میشوند.

قبل از اینکه از بین این چهار کروموزوم ۲ عدد از آنها انتخاب شود، ابتدا برای هر یک از آنها با احتمال P_{mu} جهش صورت میگیرد. سپس برای هریک با احتمال P_{se} (احتمال انجام جستجوی موضعی) جستجوی موضعی انجام میشود.

برای انتخاب از بین کروموزوم های فرزند و والد، فرض کنیم والد اول را P1 بنامیم و والد دوم را P2 بنامیم و فرزندها را CH1 و CH2 نام دهیم. اگرعدم تطابق کروموزوم I را با f(I) نشان دهیم . ابتدا کروموزوم های P1 و CH1 را در نظر می گیریم و عدد تصادفیr را در بازه صفر ویک تولید می کنیم.

اگر $(f(P1)+f(CH1))\times r \le f(P1)$ بودCH1 جانشین P1 خواهد شد و در غیر این صورت CH1نسل را ترک خواهد کرد. عین این موضوع در مورد فرزند دوم و والد دوم وجود دارد، یعنی اگر $(f(P2)+f(CH2))\times r \le f(P2)$ بودCH2 جانشین P2 خواهد شد و در غیر این صورت CH2 نسل را ترک خواهد کرد.

در مورد عمل جهش نیز همین روش برای جایگزینی وجود دارد. عمل جهش صورت می گیرد و یک کروموزوم جهش یافته ایجاد می گردد. سپس یک مقایسه بین تطابق کروموزوم جهش یافته و کروموزوم اولیه انجام می گردد و با احتمالی متناسب با تطابق کروموزوم اولیه یا کروموزوم جهش یافته انتخاب می شود. اگر تعداد اعضای هر نسل را POP در نظر بگیریم در هر عمل تقاطع و نیز در هر عمل جهش عدد فوق ثابت می ماند و نهایتاً کروموزوم های بوجود آمده جایگزین کروموزوم هایی می شوند که آنها را بوجود آورده اند. برای اینکه ممکن است در ایجاد نسل توسط تقاطع یا جهش با شروع از یک نسل اولیه بعضی از ویژگیهای مطلوب دیده نشود لازم است که تغییرات تصادفی در افراد جامعه وجود داشته باشد بنابراین در هر نسل تعدادی کروموزوم مهاجر که به تصادف ایجاد شده اند به جامعه وارد می شوند و به تصادف جایگزین افراد جامعه می شوند. برای این کروموزوم های تصادفی مقدار تطابق محاسبه می گردد . اگر P جامعهٔ فعلی باشد و $I \in P$ و new کروموزوم جدید باشد یکی از افراد جامعه به تصادف انتخاب شده و با احتمال $P_{leave}(I,new)$ کروموزوم جدیدجایگزین آن می گردد و اگر $f(.)$ مقدار عدم تطابق باشد مقدار $P_{leave}(I,new)$ بصورت زیر بدست می آید:

$$P_{leave}(I,new)=\frac{f(I)}{f(I)+f(new)}$$

یک عدد تصادفی ایجاد می شود مانند rand. اگر rand < $P_{leave}(I,new)$ کروموزوم I از جامعه خارج می شود و new جای آنرا می گیرد و در غیر اینصورت new به عنوان مهاجر پذیرفته نمی شود. بنابر این هر چقدر عدم تطابق $f(new)$ بیشتر باشد احتمال پذیرفته نشدن کروموزوم new بیشتر است. تعداد اعضای جامعه POP و تعداد نسل هایی که می خواهیم تولید کنیم در هر اجرا قابل تنظیم می باشد. در هر نسل برای کلیه اعضای جامعه این کار صورت می گیرد.

۳-۲) کروموزوم ها و تطابق[10]:

۳-۲-۱) کلیات، طرح کروموزوم و تابع تطابق:

هر فرد از جامعه دارای کروموزومی میباشد که ازدو قسمت تشکیل شده است، یکی ترتیب برنامه ریزی فعالیتها و دیگری سطح هریک از منابع. قسمت اول را قسمت ترتیب می نامیم و قسمت دوم را قسمت سطح منابع می نامیم. در این صورت یک کروموزوم به شکل زیر تعریف می شود:

j_u^I : عنصر u ام قسمت ترتیب فرد I ام جامعه که شماره یکی از فعالیتهای پروژه است که در مکان u ام قسمت ترتیب کروموزوم قرار دارد.

R_v^I : سطح منبع قابل تجدید v ام در فرد I ام جامعه که یک عدد حقیقی میباشد.

فرد I ام بصورت زیر خواهد بود:

[9] Local Search
[10] Fitness

$$I = ((j_1^I,...,j_n^I),(R_1^I,...,R_\rho^I))$$

ترتیب فعالیتها از چپ به راست $(j_1^I,...,j_n^I)$ طوری در نظر گرفته می شود که کلیه فعالیتهای پیش نیاز یک فعالیت، سمت چپ آن فعالیت قرار گرفته باشند.یعنی اگر a و b دو فعالیت در پروژه باشند:

$$a \in P_b \ , j_k^I = a, j_l^I = b \ \Rightarrow k < l$$

مقدار سطح منابع نیز بصورت تصادفی تعیین می گردد.

در واقع هر کروموزوم (genotype) می تواند تنها یک برنامه (Phonotype) ایجاد کند. هنگام تبدیل جنوتایپ به فنوتایپ از روش تولید برنامه سری[11] استفاده می شود. و زمانهای شروع $S_1,...,S_n$ تعیین می گردد. روش تولید برنامه سری برنامه های فعال بوجود می آورد.[12] و برای معیارهای منظم کارآیی[13] جواب بهینه حتماً در بین برنامه های فعال می باشد[Baker].

معیار تابع هدف مسئله RIP منظم نیست(زیرا ممکن است یکی از زمانهای اتمام فعالیتها در برنامه کاهش یابد ولی هزینه منابع افزایش یابد) ولی هنگامی که در کروموزوم سطح منابع مشخص باشد و بخواهیم یک برنامه بدست آوریم که زمان کل آن از زمان حداکثر ممکن بیشتر نشود می توان یک مسئله RCPS را حل نمود که دارای معیار منظم کارایی می باشد[Baker]. یعنی جواب حداقل زمان اتمام پروژه با منابع مشخص در درون جواب های فعال می باشد و روش ایجاد برنامه سری جوابهای فعال را می دهد.

باید توجه کرد که با داشتن کروموزومها به شکل فوق برنامه های تولید شده از نظر تقدم و تأخر فعالیتها موجه می باشند ولی ممکن است زمان پروژه از حداکثر زمان مجاز بیشتر شود که در اینصورت از نظر حد اکثر زمان اتمام غیر موجه است.

در اینجا می توان گفت که الگوریتم فوق این امکان را فراهم می کند که مسائل سرمایه گذاری در منابع را که در آن جریمه دیرکرد به ازای هر واحد زمان برای کل پروژه وجود دارد را نیز بتوانیم حل کنیم. به هر حال اگر C_d مقدار جریمه دیرکرد پروژه در هر واحد زمان باشد. زمان اتمام پروژه $S_n + D_n$ می باشد و مقدار کل جریمه دیرکرد $C_d \times (S_n + D_n - T)$ است اگر $S_n + D_n$ > T باشد.

در اینجا عدم تطابق جمع هزینه منابع و هزینه دیرکرد در صورت تاخیر در نظر گرفته می شود. اگر تاخیر وجود داشته باشد:

$$f(I) = \sum_{k=1}^{m} C_k R_k^I + C_d \times (S_n^I + D_n^I - T)$$

در غیر اینصورت:

$$f(I) = \sum_{k=1}^{m} C_k R_k^I$$

واضح است که اگر بخواهیم اصلاً دیرکرد وجود نداشته باشد می توان مقدار C_d را نسبت به C_k بزرگ در نظر گرفت.

اما نکته قابل توجه اینستکه کروموزومهای غیرقابل قبول به دلیل تأخیر ممکن است دارای بعضی ساختارهای مناسب باشند و اگر آنها را به دلیل بزرگ در نظر گرفتن اصلاً وارد نسل‌های الگوریتم نکنیم ممکن است به ساختارهای مناسب فوق به ندرت برخورد کنیم. برای سود بردن از ساختارهای مناسب کروموزومهایی که دارای تأخیر زمانی هستند و بطور موازی حذف آنها از دامنهٔ جوابهای موجه از یک عمل سردسازی تدریجی[14] استفاده شده است. برای این منظور ابتدا مقدار C_d زیاد در نظر گرفته نمی شود با گذشت هر نسل مقدار C_d به اندازهٔ Δ_{inc} افزایش می یابد. به این ترتیب هم از ساختارهای احتمالی مناسب در کروموزوم های غیرقابل قبول استفاده می شود و هم در انتهای اجرای الگوریتم این کروموزومها تا جای ممکن از داخل نسل حذف می شوند.

چند کروموزوم ترتیبی ممکن است بعد از رمز گشایی منجر به یک جواب در برنامه شوند یعنی در طرح فوق چند کروموزوم متفاوت ممکن است برنامه های مشابه بدهند (یعنی چند جنوتایپ یک فنوتایپ را بدهند) یعنی رابطه کروموزومها و برنامه ناشی از آنها یک به n است. برای یک مثال ساده شبکه گره‌ای زیر را در نظر بگیرید(شکل ۱):

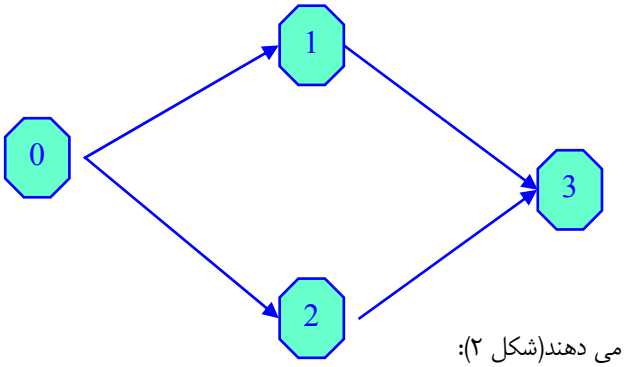
---

[11] Serial SGS

[12] – یک برنامه را فعال می‌گویند اگر هیچ فعالیتی از آن نتواند به سمت چپ (روی محور زمان) حرکت داده شود مگراینکه حداقل یک فعالیت دیگر مجبور شود به سمت راست حرکت کند.

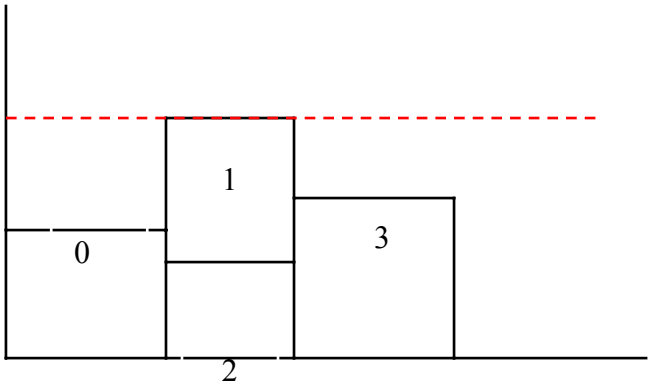
[13]- Regular measure of Performance.

[14]- Anealing



(شکل ۱)
شبکه گره ای فعالیتها

کروموزومهای (0,1,2,3) و (0,2,1,3) هر دو جواب شکل زیر را می دهند(شکل ۲):



(شکل ۲): نمودار مصرف منابع برای کروموزومها

۳-۲-۲) ایجاد یک کروموزوم به تصادف:

برای تولید تصادفی کروموزومها چون نمی دانیم کدام ترتیب مناسب تر است بهتر است کروموزومهای تصادفی با توزیع تقریباً یکنواخت از کل فضای جواب موجود بیایند تا یک جستجوی کلی در کل فضای جواب صورت گرفته باشد و امکان حضور نقاط مختلف فضای جواب بطور همگن وجود داشته باشد.

برای بوجودآمدن یک ترتیب تصادفی می توان به صورت زیر عمل نمود:

- لیست فعالیتها را actirity-list بنامید و فعالیت ها را طوری اسم گذاری کنید که اگر $i \in P_j$ بود آنگاه i < j باشد.
- مجموعهٔ فعالیتها را V در نظر بگیرید و لیست فعالیتهایی بنام activity-list بسازید. که در ابتدا شامل تمام فعالیتهای V باشد($V=\{0,1,\ldots,n,n+1\}$).
- مجموعهٔ فعالیتهایی که ترتیب آنها مشخص شده است را S_List بنامید. در ابتدای کار این مجموعه خالی است.
- لیست ترتیبی فعالیتها را Per بنامید که این لیست یک n+1 تایی مرتب است که همان قسمت اول کروموزوم را تشکیل می دهد و در ابتدای کار خالی است.
- مجموعهٔ دیگری نیز تشکیل دهید که در آن کلیه فعالیتها در هر مرحله قرار خواهد گرفت که پیش نیازهای آنها در مجموعهٔ S_List قرار دارد. این مجموعه مجموعه فعالیتهای مجاز برای برنامه ریزی است(ESA)[۱۵].

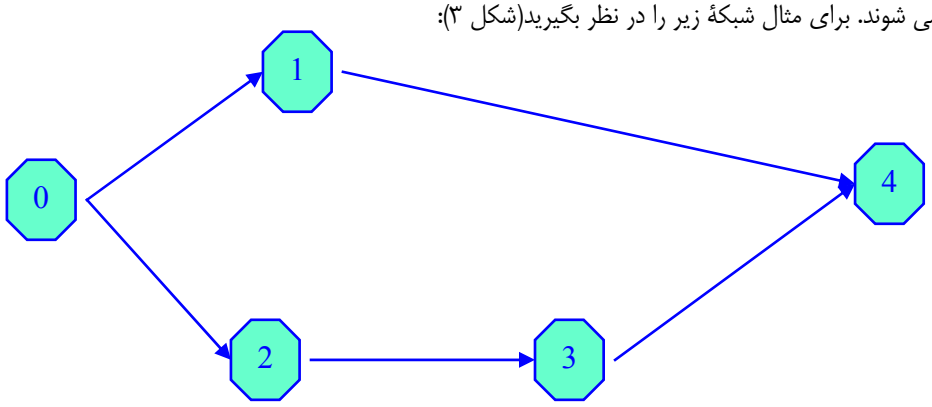
الگوریتم تشکیل ترتیب تصادفی بصورت زیر است:

۱) در نظر بگیرید $Per(a_0,\ldots,a_{n+1}) = (NULL,\ldots,NULL)$ و i = 1 و j = 1
۲) فعالیت 0 را از مجموعهٔ actirity-list خارج کنید و به مجموعهٔ EAS اضافه کنید.

[۱۵] Eligible Activity Set

۳) اگر EAS خالی باشد الگوریتم خاتمه یافته است. در غیراینصورت به قدم ۴ بروید.
۴) یک عدد صحیح با توزیع یکنواخت گسسته بین 1 و j تولید کنید و آنرا r بنامید.
۵) r امین فعالیت داخل EAS را از لیست فوق خارج کنید و آنرا در مکان i ام لیست Per قرار دهید و نام این فعالیت را job بگذارید.
۶)در نظر بگیرید i = i + 1 و j = j –1.
۷) کلیه فعالیتهای پس نیاز[۱۶] فعالیتjob را در نظر بگیرید اگر k عدد از آنها فعالیتهایی باشند که هیچ پیش نیازی در actirity-list نداشته باشند کلیه آنها را از لیست فوق خارج کنید و به EAS اضافه کنید.
۸) در نظر بگیرید j = j + k و به قدم ۴ بروید.

با این روش یک ترتیب تصادفی از فعالیتها بدست می‌آید. اما باید دقت کرد که با درنظرگرفتن انتخاب فعالیتها با توزیع یکنواخت گسسته از مجموعه EAS ترتیبهای ممکن با توزیع یکنواخت انتخاب نمی شوند. برای مثال شبکهٔ زیر را در نظر بگیرید(شکل ۳):



(شکل۳) : یک شبکه گره ای از فعالیتها

ابتدا فعالیت 0 در Per قرار می گیرد. سپس فعالیتهای 1 و 2 در EAS هستند.
اگر P_j^i احتمال قرارگرفتن فعالیت j در مکان i ام باشد خواهیم داشت:

$$P_1^1 = \frac{1}{2} \quad , P_1^2 = \frac{1}{2} * \frac{1}{2} = \frac{1}{4} \quad , P_1^3 = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$$

مشاهده می شود که فعالیت ۱ بیشتر تمایل دارد که زودتر برنامه ریزی شود در صورت اینکه امکان این وجود دارد که در مکانهای ۱ و ۲و ۳ قرار بگیرد.برای اینکه به یکنواخت بودن توزیع قرارگیری هر فعالیت در مکانهای ممکن خودش نزدیک تر شویم می توان برای انتخاب فعالیتها از داخل EAS وزن قرار داد.در اینصورت وزنهای فعالیتها بصورت زیر تعیین می شود:
w_i : وزن فعالیت i. اگر وزن فعالیت j ، w_j باشد و $i \in P_j$ آنگاه داریم:

$$w_i = \underset{\forall j | i \in P_j}{Max}\{w_j + 1\}$$

.

اولین وزن را به n امین فعالیت نسبت می دهیم و آنرا صفر می گذاریم .
برای شبکه مثال بالا خواهیم داشت:

$w_0 = 3$, $w_1 = w_3 = 1$, $w_2 = 2$, $w_4 = 0$

بنابراین احتمال قرارگیری فعالیت ۱ بعداز ایجاد توازن در وزن های انتخابی بصورت زیر خواهد بود:

$$P_1^1 = \frac{1}{1+2} = \frac{1}{3} \quad , P_1^2 = \frac{2}{3} * \frac{1}{2} = \frac{1}{3} \quad , P_1^3 = 1 - (P_1^1 + P_1^2) = \frac{1}{3}$$

[۱۶] Sucessor

همانطور که ملاحظه می‌شود، در این مثال کلیه جوابهای ممکن با توزیع یکنواخت انتخاب می شوند.

در حالت کلی ممکن است بعد از وزن دادن فعالیتها به روش بالا باز هم توزیع انتخاب کروموزومها یکنواخت نشود ولی با این کار به توزیع یکنواخت نزدیک تر می شویم. در اینصورت اگر در EAS به تعداد u فعالیت وجود داشته باشد و وزنهای آنها را به ترتیب w(1) تا w(u) بنامیم، احتمال انتخاب فعالیت j مطابق قدم ۴ الگوریتم بصورت زیر است:

$$w(j) \Big/ \sum_{i=1}^{u} w(i)$$

بعداز ایجاد رشته کروموزوم در قسمت ترتیب فعالیتها باید قسمت سطح منابع را نیز به تصادف ایجاد نمود. برای این کار در مورد هر منبع سطح در دسترس به تصادف انتخاب می شود. اما دامنهٔ انتخاب سطح منبع برای منبع kام از یک حد پایین تا یک حد بالا خواهد بود. هر چقدر دامنه آن حدود کوچکتر باشد برای رسیدن به جواب بهینه مناسب تر است.

یک حد پایین که می توان برای سطح هر منبع در نظر گرفت این است که برای هر منبع مقدار سطح حداقل آن نباید از مقدار موردنیاز آن منبع برای هریک از فعالیتهای پروژه کمتر باشد. اگر این سطح حد پایین را $\underline{R_k}$ در نظر بگیریم داریم:

$$\underline{R_k} = \underset{\forall j}{Max}\{r_{jk}\}$$

با کمی دقت می توان یک حد پایین بهتر نیزبدست آورد. اگرکل مقدار منبع موردنیاز k ام از رابطه زیر بدست آید:

$$\sum_{j=0}^{n+1} r_{jk} \times D_j$$

یک حد پایین بهتر بصورت زیر خواهد بود:

$$\underline{R_k} = Max\{\frac{\sum_{\forall j} r_{jk} \times D_j}{Min\{T, \sum_{\forall j} D_j\}}, \underset{\forall j}{Max}\{r_{jk}\}\} \qquad (A)$$

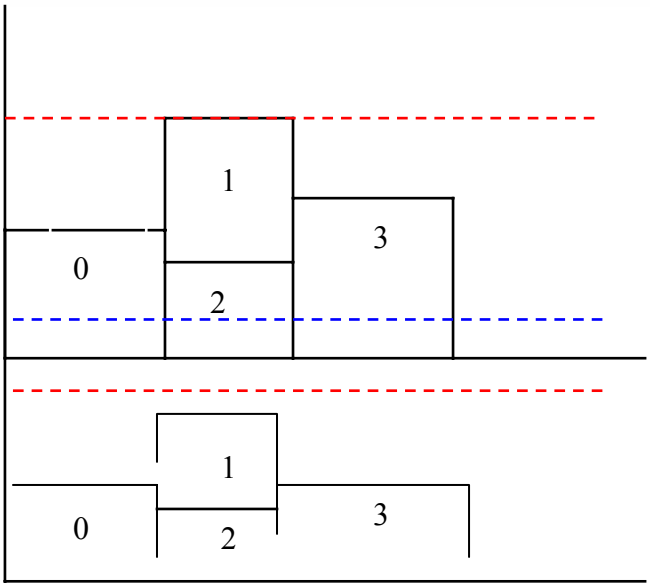
یک حد بالا برای منبع k می تواند جمع مقدار موردنیاز منبع k برای همهٔ فعالیتها باشد. اگر حد بالا را با $\overline{R_k}$ نمایش دهیم داریم:

$$\overline{R_k} = \sum_{\forall j} r_{jk}$$

ولی می توان برنامه زودترین زمان اجرای فعالیتها را در نظر گرفت که یک برنامه موجه است. و مقدار حداکثر استفاده منبع k ام در این برنامه یک حد بالای بهتر خواهد بود.

برای هر منبع k یک عدد بطور تصادف از بازهٔ $[\underline{R_k}, \overline{R_k}]$ انتخاب می شود و برای سطح منبع kام در نظر گرفته می شود. بعد از انجام روش ایجاد برنامه سری می‌توانیم در بعضی از موارد سطوح منابع را بدون تغییر ترتیب فعالیتها کاهش داد. چون سطح منابع به تصادف انتخاب می‌شود. در بعضی از مواقع ممکن است سطح یک منبع در یک برنامه از حداکثر موردنیاز در آن برنامه بیشتر در نظر گرفته شده باشد در اینصورت سطوح منابع طوری کاهش می‌یابد که با حداکثر موردنیاز در آن برنامه تطابق داشته باشد.

برای مثال ممکن است یک کروموزوم تصادفی با روش ایجاد برنامه سری برای منابع ۱ و ۲ بصورت زیر نتیجه دهد(شکل ۴):



(شکل ۴): ایجاد سطح منابع بصورت تصادفی و اصلاح آن

مشخص است که سطح منبع ۲ که به تصادف انتخاب شده است را می‌توان از R_2 به R_2^1 کاهش داد.

۳-۳)طرح تقاطع[۱۷]:

برای تقاطع دو کروموزوم در قسمتی که ترتیب فعالیتها واقع است از تقاطع‌های زیر استفاده می‌شود:

۳-۳-۱) تقاطع یک نقطه‌ای:

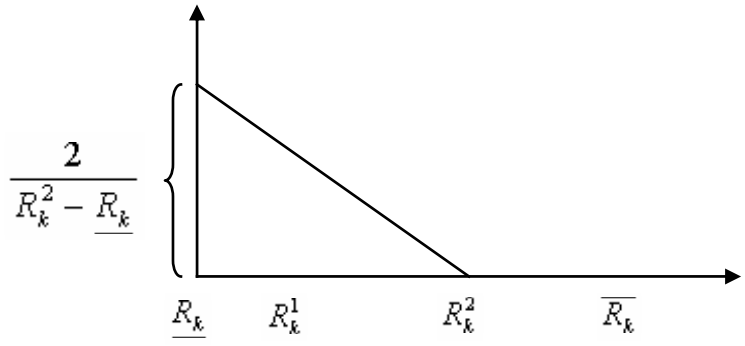
والد ها را P1 و P2 بنامید. در قسمتی از کروموزوم‌ها که ترتیب فعالیتها وجود دارد و دارای n خانه می‌باشد عددی به تصادف از بین اعداد ۱ تا n انتخاب می‌شود. آنرا C_r بنامیم.

برای فرزند اول خانه‌های C_r تا n از کروموزوم P1 عیناً در فرزند کپی می‌شود. و برای خانه‌های ۱ تاC_r-1 فعالیتهای باقی مانده در P1 به ترتیبی که در کروموزوم P2 هستند کپی می‌شود. برای فرزند ۲ ابتدا خانه‌ها از P2 کپی می‌شوند و بعد به ترتیب P1.

برای مثال دو کروموزوم 6 2 4 3 5 1 و 5 6 1 4 3 2 را در نظر بگیرید و فرض کنیم C_r = 4 باشد. در اینصورت فرزند ۱ بصورت 5 2 1 4 3 6 است. برای تعیین سطح منابع از مقدارعدم تطابق استفاده می‌شود.

اگر عدم تطابق والد ۱ را f_1 و عدم تطابق والد ۲ را f_2 بنامیم و $f_1 \le f_2$ باشد و سطح منبع k برای والد ۱ برابر R_k^1 و برای والد ۲، R_k^2 باشد و اگر $R_k^1 \le R_k^2$ باشد سطح منبع فرزند آنها بطور تصادفی از یک توزیع مثلثی بدست می‌آید. توزیع مثلثی طوری در نظر گرفته می شود که سطح منبع تمایل بیشتری داشته باشد که به سمت سطح منبع کروموزومی از والد خود حرکت کند که عدم تطابق کمتری دارد.

توزیع مثلثی فوق بصورت زیر است(شکل ۵):

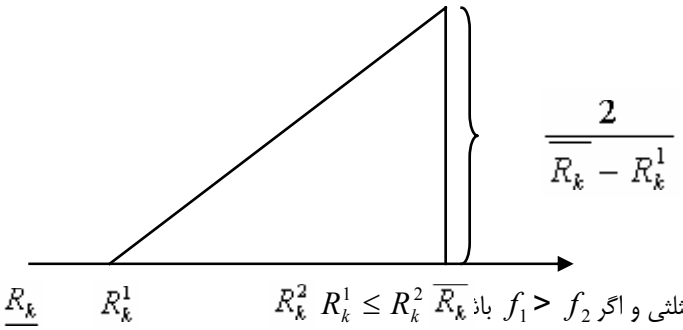


(شکل ۵): توزیع مثلثی اگر $R_k^1 \le R_k^2$ و $f_1 \le f_2$

[۱۷] Crossover design

$$f_X(x)=\frac{2\times(R_k^2-x)}{(R_k^2-\underline{R_k})^2} \qquad \underline{R_k}\le x\le R_k^2$$

و اگر $f_1 > f_2$ باشد و $R_k^1 \le R_k^2$ باشد، توزیع مثلثی زیر برای تعیین سطح منبع بکار گرفته می شود(شکل ۶):



(شکل ۶): توزیع مثلثی و اگر $f_1 > f_2$ باشد $R_k^1 \le R_k^2$ $\overline{R_k}$

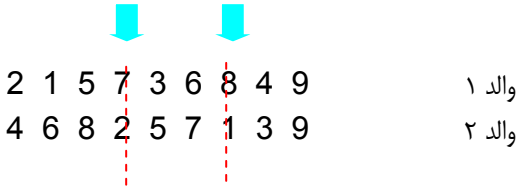
$$f_X(x)=\frac{2\times(x-R_k^2)}{(\overline{R_k}-R_k^2)^2} \qquad R_k^1\le x\le \overline{R_k}$$

۳-۳-۲) تقاطع دو نقطه‌ای:

برای تقاطع قسمت ترتیب فعالیتها دو عدد تصادفی بین ۱ تا n ایجاد می‌گردد. (آنها را C_{r_1} و C_{r_2} بنامیم) برای فرزند اول خانه‌های C_{r_1} تا C_{r_2} مستقیماً از والد ۱ در خانه‌های مشابه آن کپی می‌شود. در خانه‌های ۱ تا C_{r_1} -1 همان فعالیتهای خانه‌های ۱ تا C_{r_1} -1 والد ۱ با ترتیب قرارگیری آنها در والد ۲ قرار می‌گیرد و در خانه‌های C_{r_2} +1 تا J نیز همان فعالیتهای والد ۱ با ترتیب قرارگیری آنها در والد ۲ قرار می‌گیردبرای فرزند ۲ برعکس خانه‌های C_{r_1} تا C_{r_2} از والد ۲ کپی می‌شود.

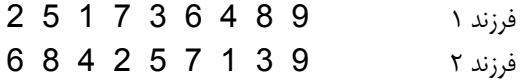
وموارد مشابه فرزند ۱ روی فرزند ۲ نیز صورت می گیرد.

برای مثال دو والد زیر را در نظر بگیرید(شکل ۷):



(شکل ۷): والدها

باتوجه به نقاط شکست فرزندان زیر را خواهیم داشت(شکل ۸):



(شکل ۸): فرزندان

با توجه به اینکه روابط تقدم و تأخر در بین فعالیتها وجود دارد و اینکه ممکن است چند کروموزوم هنگامی که به روش ایجاد برنامه سری تبدیل به یک برنامه می‌شوند همانند هم باشند روش تقاطع دونقطه‌ای برای هنگامی که تعداد فعالیتها زیاد نمی‌باشد (تا ۳۰ فعالیت) معمولاً منجر به فرزندان جدید نمی‌شود. یا اگر فرزند جدیدی تولید شود تشابه خیلی زیادی با یکی از والدها دارد (مثلاً جای دو عدد از فعالیتها در رشته با یکدیگر عوض می‌شود) بنابراین برای این موارد بهتر است از روش تقاطع یک نقطه‌ای استفاده شود. از طرفی اگر مقدار فعالیتها خیلی زیاد باشد (از حدود ۱۰۰ عدد به بالا) روش

یک نقطه‌ای منجر به تولید فرزندانی می‌شود که ممکن است تفاوت زیادی با والدهای خود داشته باشند یا مشابه حالتی خواهد بود که به تصادف یک کروموزوم تولید شده است. در اینصورت برای اینکه خصوصیات والدهای آنها در فرزندان باقی بماند و از حالت تصادفی بودن خارج شویم بهتر است از تقاطع دونقطه‌ای استفاده شود.

در تقاطع دونقطه‌ای برای تولید سطح منابع دقیقاً مانند تقاطع یک نقطه‌ای عمل می‌شود.

۳-۴) جهش[۱۸]:

یک کروموزوم با احتمال P_{mu} جهش می‌یابد. در کروموزومی که می‌خواهد جهش یابد یک عدد تصادفی در فاصلهٔ ۱ تا n انتخاب می‌شود. اگر آنرا a بنامیم. در روی قسمت ترتیب فعالیتهای کروموزوم فعالیت واقع در مکان a در نظر گرفته می‌شود.

در روی رشتهٔ فوق مکانهای b و c به ترتیب مکان آخرین پیش‌نیاز فعالیت موردنظر روی رشتهٔ کروموزوم و مکان اولین پس‌نیاز فعالیت واقع در مکان a می‌باشد.

یک عدد تصادفی بین b+1 و c+1 تولید می‌شود. اگر آنرا d بنامیم و d < a فعالیت موردنظر به مکان d منتقل می‌شود و کلیه فعالیتهای واقع در مکانهای d تا a-1 یک واحد به سمت راست شیفت پیدا می‌کنند. برای مثال رشتهٔ زیر را در نظر بگیرید: 8 6 4 2 7 3 5 1 10 9 11. اگر a = 6 باشد و پیش‌نیازهای فعالیت 4، فعالیتهای 1 و 3 باشند و پس‌نیازهای آن 11 و 10 باشند یک عدد تصادفی بین 4 و 8 تولید می‌شود. اگر این عدد 4 باشد کروموزوم زیر را خواهیم داشت: 10 9 11 8 6 2 7 4 3 5 1.

اگر d < a باشد فعالیت مورد نظر به مکان d منتقل می شود و فعالیتهای واقع در مکانهای a + 1 تا d یک واحد به چپ حرکت می‌کنند.

برای جهش در سطح منابع یک منبع به تصادف انتخاب می‌شود و یک واحد از آن کاسته می‌شود. کروموزوم جهش یافته با احتمالی متناسب با عدم تطابق آن جایگزین کروموزوم اولیه می‌شود.

۳-۵) روش جستجوی موضعی[۱۹]:

در اینجا دو روش جستجوی موضعی بکار گرفته شده است.

۳-۵-۱) روش ۱:

برای یک کروموزوم دلخواه تعاریف زیر را در نظر بگیریم:

– لیست فعالیتهای مؤثر: برای هر منبع k لیست فعالیتهای مؤثر فعالیتهایی را شامل می‌شود که در نقطه‌ای از زمان که منبع موردنظر در حال استفادهٔ ماکزیمم است در حال اجرا می‌باشد. نام این لیست را برای منبع k ، acttive_List[k] بگذاریم و فعالیتهای داخل لیست را فعالیتهای موثر می نامیم.

– اولین زمان ممکن: برای هر فعالیت مؤثر و برای یک منبع موردنظر اولین زمان ممکن زمانی است که با شیفت فعالیت موردنظر به سمت راست اولین زمانی باشد که بدون در نظر گرفتن روابط تقدم و تأخر در نقطهٔ شروع جدید فعالیت، کل استفاده از منبع از سطح فعلی استفاده کمتر شود یا مساوی سطح فعلی شود.

_حرکت مؤثر: حرکت یک فعالیت مؤثر به اولین زمان ممکن.

قدمهای الگوریتم بصورت زیر است:

۱) در نظر بگیرید k = 1
۲) برای منبع k لیست فعالیتهای مؤثر را تشکیل دهید. و در نظر بگیرید i = 1
۳) i امین فعالیت لیست فعالیتهای مؤثر را در نظر بگیرید.
۴) فعالیت i را بدون در نظر گرفتن تقدم و تأخر یک حرکت مؤثر دهید.
۵) کروموزوم جدیدی به شکل زیر بسازید.

[۱۸] Mutation
[۱۹] Local Search

الف) کلیه فعالیتهایی که قبل از فعالیت i (بعد از حرکت مؤثر) شروع می‌شوند را در نظر گرفته و از آنها تمامی فعالیتهایی که چه بطور مستقیم و چه بطور غیرمستقیم پس‌نیازi نمی‌باشند را با همان ترتیب کروموزوم اولیه در کروموزوم جدید کپی کنید.

ب) فعالیت i را در خانه بعدی کپی کنید.

ج) فعالیت‌های باقی مانده را در مکانهای باقی مانده به ترتیب کروموزوم اولیه کپی کنید.

۶) یک واحد از سطح منبع k کم کنید.

۷) تطابق کروموزوم جدید را حساب کنید.

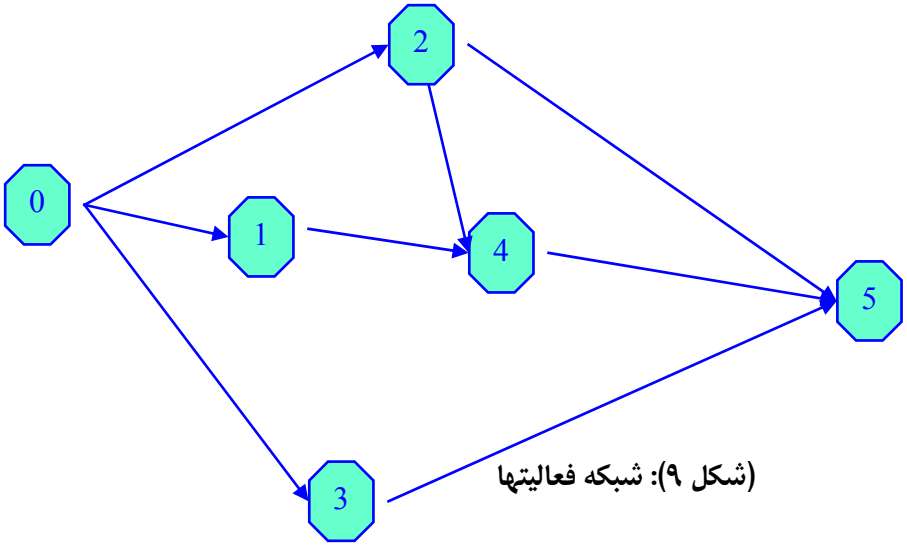
۸) اگر تطابق بهتر شد، کروموزوم جدید جایگزین کروموزوم قدیم می‌شود و به قدم ۱ بروید در غیر اینصورت

۹) در نظر بگیرید i = i + 1 و واحد کم شده از منبع k را دوباره اضافه کنید و به ۹ بروید.

۱۰) اگر i از تعداد فعالیتهای لیست فعالیتهای مؤثر بیشتر شد به قدم ۱۰ بروید ودر غیر اینصورت به قدم ۴ بروید.

۱۱) در نظر بگیرید k = k + 1 ، اگر k از تعداد منابع بیشتر شد الگوریتم خاتمه یافته و در غیر اینصورت به قدم ۲ بروید.

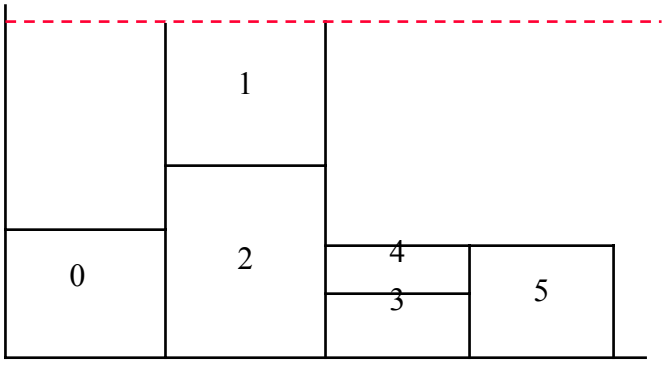
برای مثال شبکه زیر(شکل ۹) و کروموزوم مربوط به آن (شکل ۱۰) و برنامه تولید شده ازآنرا (شکل ۱۱) در نظر بگیرید:



(شکل ۹): شبکه فعالیتها

0 2 1 4 3 5

(شکل ۱۰): کروموزوم

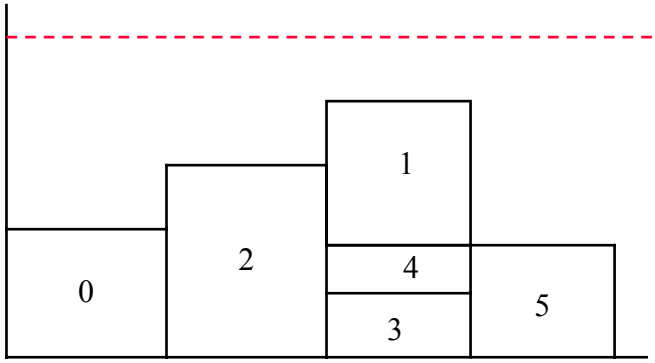


(شکل ۱۱): نمودار مصرف منبع

لیست فعالیتهای مؤثر: active_list = { 1, 2 }

اگر فعالیت ۱ اول انتخاب شود این فعالیت می‌تواند در زمان ۲ شروع شود.

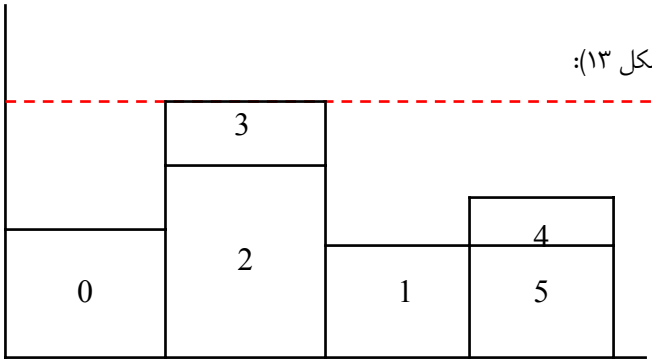
زیرا در اینصورت سطح منبع بدون در نظر گرفتن تقدم و تأخر می تواند کاهش یابد(شکل ۱۲):



(شکل ۱۲): نمودار مصرف منبع بعد از جابجایی فعالیت ۱ و قبل از انجام ایجاد کروموزوم تغییر یافته

بنابر این یک کروموزوم ممکن با انتقال این فعالیت به زمان ۲ بدست می آید. کروموزوم جدید بصورت 0 2 3 1 4 5 است.

و برنامه بوجود آمده از آن بصورت زیر است(شکل ۱۳):



(شکل ۱۳)

۳-۵-۲) روش ۲:

در این روش از ایدهٔ بیان شده در جهش استفاده می‌شود.

تعریف جهش (i,q) : یک جابجایی در ترتیب اجزای کروموزوم طوریکه:

حرکت فعالیت i در یک کروموزوم از مکان خودش به مکان q ام طوریکه اگر مکان فعالیت i ، a باشد و اگر q<a باشد، فعالیتهای مکانهای q تا a-1 یک واحد به راست حرکت می‌کنند و اگر q > a باشد فعالیتهای مکانهای a+1 تا q یک واحد به چپ حرکت می‌کنند.

– تعریف مکان حداقل فعالیت : در یک کروموزوم مکان حداقل L(i) یک خانه بعد از مکان آخرین پیش‌نیاز مستقیم i است اگر از چپ به راست روی رشته کروموزوم حرکت کنیم.

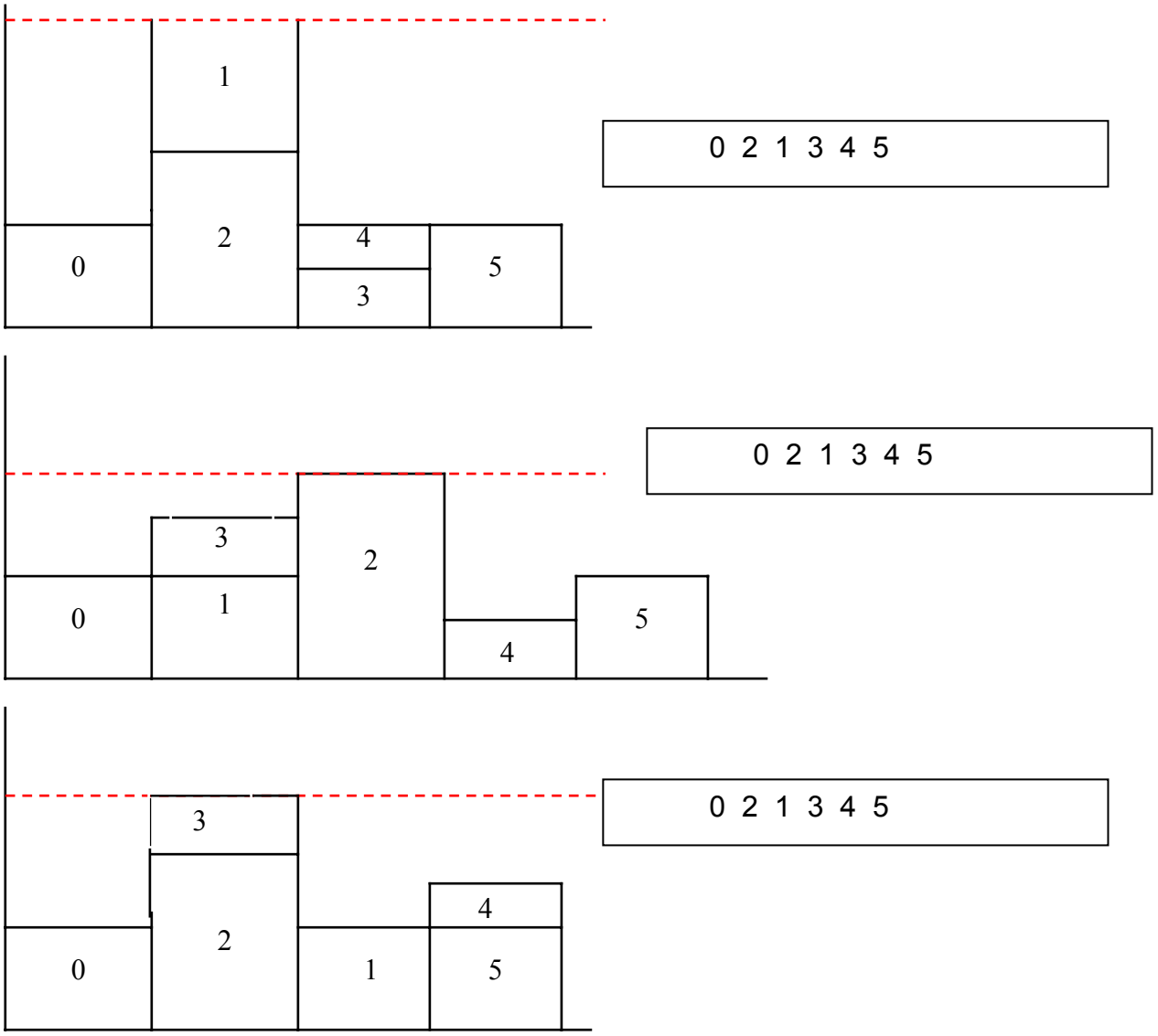
– تعریف مکان حداکثر فعالیت : در یک کروموزوم مکان حداکثر U(i) یک خانه قبل از مکان اولین پس‌نیاز مستقیم i است اگر از چپ به راست روی رشته کروموزوم حرکت کنیم.

قدمهای روش فوق به قرار زیر است:

۱) در نظر بگیرید k = 1 و اولین منبع را در نظر بگیرید.

۲) لیست فعالیتهای مؤثر را تشکیل دهید و در نظر بگیرید i = 1.

۳) i امین فعالیت مؤثر را در نظر بگیرید.
۴) مقادیر L(i) و U(i) را بدست آورید و قرار دهید j = L(i).
۵) یک جهش (i,j) انجام دهید.
۶) یک واحد از سطح منبع k کم کنید.
۷) تطابق کروموزوم بوجود آمده را بدست آورید.
۸) اگر کروموزوم جدید بهتر بود آنرا جایگزین کروموزوم قدیم کنید و به ۱ بروید. در غیر اینصورت در نظر بگیرید j = j + 1 و واحد کم شده از منبع k را دوباره اضافه کنید و به ۹ بروید.
۹) اگر j > U(i) بود در نظر بگیرید: i = i + 1
۱۰) اگر i از تعداد فعالیتهای مؤثر بیشتر شد در نظر بگیرید k = k + 1 و به ۱۱ بروید. در غیر اینصورت به ۳ بروید.
۱۱) اگر k از تعداد منابع بیشتر شد الگوریتم خاتمه یافته در غیر اینصورت به ۲ بروید.
اگر مثال قبل را در نظر بگیرید به ترتیب کروموزوم‌های زیر را بدست می‌آوریم (شکل ۱۴):



(شکل ۱۴): نمودار های بدست آمده از مراحل مختلف الگوریتم

۴) انجام آزمایشات روی الگوریتم ژنتیک و مقایسه نتایج:

برای مسئله RIP در حالتی که تأخیر زمانی حداکثر وجود نداشته باشد بدلیل اینکه تحقیقات زیادی روی این مسئله انجام نشده است، مسائل با جـواب بهینه وجود ندارد. از جمله تحقیقاتی که در این زمینه اخیراً صورت گرفته توسط کیمز (Kimms) بوده که روی بدست آوردن حدود پایین و بالا برای این مسئله کار کرده است.

کیمز مدل برنامه‌ریزی ریاضی مسئله RIP (فصل ۶-۲) را با استفاده از دو روش بررسی کرده است. روش اول با استفاده از ساده‌سازی لاگرانژ[۲۰] بود. در این روش محدودیت‌های مسئله را با اعمال ضریب لاگرانژ به تابع هدف برده و از روش جستجوی گرادیان حدود بالا و پایین را بدست آورده است.

روش دوم روش ایجاد ستون[۲۱] می‌باشد. با استفاده از این روش نیز حدود بالا و پایین را برای مسئله فوق بدست آورده است.

او روشهای خود را روی مسائل استاندارد تخصیص منابع که در سایت اینترنتیPSBLIB [RainerKolisch(1996)]در حالت یک روش اجرا آزمایش کرده است. برای آزمایش مسئله RIP روی مسائل با ۳۰ فعالیت وبا یک روش اجرا(موجود در فایل فشرده شده30.SM.RCP در سایتPSBLIB) او اعداد مربوط به سطح منابع که در فایل فوق قرار دارد را نادیده گرفت (زیرا این اعداد در مسئله سرمایه گذاری در منابع کاربرد ندارند) و برای هزینه منابع از بین اعداد ۱ تا ۱۰برای هریک از چهار منبع یک عدد را به تصادف انتخاب نموده است. خوشبختانه این اعداد برای انجام آزمایش الگوریتم ژنتیک اکنون در دسترس است. او نتایج خود را نیز در مقاله ارائه نموده است. و کار آیی حدود بالا وپایین بدست آمده از روشهای خود را نسبت به حدود ساده بدست آمده در بخش ۳-۲-۱ که نام آنها را UB_0برای حد بالا و LB_0 برای حد پایین می گذاریم بر حسب درصد محاسبه کرده است.

اگر UB , LB به ترتیب حد بالا و حد پایین بدست آمده از روش کیمز باشد میزانی که روش او این حدود را بهبود بخشیده توسط فرمول زیر به درصد محاسبه نموده است:

در صد بهبود حد بالا $\frac{UB_0 - UB}{UB_0} \times 100$

در صد بهبود حد پایین $\frac{LB - LB_0}{LB_0} \times 100$

چون الگوریتم ژنتیک جوابهای موجه تولید می‌کند برای مقایسه الگوریتم ژنتیک عدد بدست آمده از این الگوریتم را نیز مطابق فرمولها بدست آمده به عنوان بهبود حد بالا با اعداد کیمز مقایسه می کنیم.

مسائل فوق همگی توسط برنامه PROGEN تولید شده اند. این برنامه که توسط کولیش بوجود آمده است قادر است به روش سیستماتیک مسائل برنامه ریزی پروژه را بوجود آورد. این نرم افزارباتوجه به اینکه قابلیت این را دارد که پارامتر های مختلف که شدت پیچیدگی مسئله را تعیین می کند تعریف کند اجازه می دهد که مسائل برنامه ریزی پروژه باتوجه به پارامتر های موجود به صورت طرح فاکتوریل کامل بوجود بیاید. او سه دسته پارامتر برای تولید مسائل بوجود آورده است. پارامتر های ثابت، پارامتر های پایه و پارامتر های متغیر. مسائل استاندارد موجود در سایت PROGEN که اخیرا مرجع بسیار مناسبی برای ارزیابی الگوریتمهای برنامه ریزی پروژه شده اند از ترکیبات مختلف این پارامتر ها استفاده کرده اند.

در مسائل مورد بررسی ما پارامترهای زیر مقادیر مختلف گرفته اند:

۱) NC[۲۲] : پیچیدگی شبکه نشان دهنده متوسط تعداد بردار های غیر تکراری گذرنـده از یـک گـره در شـبکه پروژه شامل گره های مجازی در شبکه گره ای.

۲) RF_k[۲۳] : متوسط مقداری که از منبع k استفاده می شود.

۳) RS_k : نشان دهنده مقدار در دسترس بودن منبع k برای مسائل تخصیص منابع محدود. مقدار صـفر را مـی گیرد اگر منبع k به اندازه ای باید باشند که فعالیت با بیشترین مقدار لازم برای استفاده نیاز دارد و مقدار یـک را می گیرد برای سطح منبع در برنامه زود ترین زمان اجرای ممکن.

در PSBLIB دسته مسائلی که توسط کیمز برای آزمایش انتخاب شده است مسائل با ۳۰ فعالیت و ۴ منبع قابل تجدید می باشد. در این مسائل ترکیب پارامتر های فوق به صورت زیر است:

---

[۲۰] Lagrangian Relaxation Method
[۲۱] Column Generation
[۲۲] Network Complexity
[۲۳] Resource Factor

فاکتور	سطح پارامتر			
NC	۱٫۵	۱٫۸	۲٫۱	
RF	۰٫۲۵	۰٫۵	۰٫۷۵	۱
RS	۰٫۲	۰٫۵	۰٫۷	۱

در اینجا ۴۸ = ۳×۴×۴ دسته پارامتر برای تولید مسائل تصادفی وجود دارد. در مسائل استاندارد در سایت PSBLIB از هر یک از این دسته ها ۱۰ مسئله به تصادف تولید شده است. در اینجا این نکته قابل توجه است که پارامتر RS برای مسئله RIP کاربردی ندارد. در عوض در مقایسه مسئله RIP و RCPSP مسائل تخصیص منابع محدود سعی بر این دارد که پروژه در زود ترین زمان پایان یابد و با هزینه منابع کاری نداریم در صورتیکه در مسئله سرمایه گذاری در منابع پروژه باید در زمان مشخص تمام شود(یا در صورت تاخیر جریمه پرداخت شود) و میزان هزینه برای هر منبع قابل تجدید باید مشخص باشد.

کیمز در مورد کل زمان ممکن برای پروژه پارامتر θ را تعریف کرده است. اگر زود ترین زمان اتمام پروژه را EF بنامیم برای θ مقادیر ۱٫۵ ، ۱٫۴ ، ۱٫۳ ، ۱٫۲ ، ۱٫۱ ، ۱ را در نظر گرفته و مدت مجاز اجرای پروژه را $\theta \times$ EF در نظر گرفته است. بنابر این کل مسائل ممکن برای بررسی۲۸۸۰=۱۰×۶×۴۸ مسئله است. از طرفی برای هزینه منابع در هر دسته مسئله و برای هر منبع عددی به تصادف از بازه ۰ تا ۱۰ انتخاب کرده است(که البته خود این اعداد در مقاله او وجود ندارد). سپس او بعضی از این مسائل را بدلیل اینکه با استفاده از روش تولید ستون در زمان معقول قابل حل نبودند را حذف نمود. وتنها دسته ای از آنها را که توسط روش تولید ستون در زمان معقول قابل حل بودند در نظر گرفت.

تعداد مسائلی که در هر دسته از پارامتر ها توسط او حل شده است در جدول ۱ آمده است:

جدول ۱: تعداد مسائل حل شده در هر دسته از پارامترتوسط کیمز

n = 30	θ	1	1.1	1.2	1.3	1.4	1.5
NC = 1.5	RF = .25	40	40	40	40	40	40
	RF = .5	37	26	11	22	27	34
	RF = .75	28	8	2	7	12	13
	RF = 1	21	7	3	2	5	8
NC = 1.8	RF = .25	40	40	40	40	40	40
	RF = .5	40	26	17	23	31	38
	RF = .75	24	7	3	12	17	23
	RF = 1	27	8	1	0	2	9
NC = 2.1	RF = .25	40	40	40	40	40	40
	RF = .5	40	29	20	30	36	39
	RF = .75	34	10	1	5	16	22
	RF = 1	32	4	0	2	3	7

متوسط مقدار بهبود بدست آمده توسط روشهای ساده سازی لاگرانژ و ایجاد ستون در جداول ۲ و ۳ آمده است:

جدول ۲: متوسط مقدار بهبود بدست آمده توسط روش ساده سازی لاگرانژ

n = 30	θ	1	1.1	1.2	1.3	1.4	1.5
NC = 1.5	RF = .25	27.93	29.76	28.77	26.11	24.8	25.18
	RF = .5	23.21	28.98	30.68	33.49	33.73	36.42
	RF = .75	17.44	29.8	43.88	35.87	35.65	37.1
	RF = 1	17.07	23.98	28.67	24.92	26.8	34.78
NC = 1.8	RF = .25	22.05	24.87	25.38	24.56	21.52	22.33
	RF = .5	19.78	24.81	28.42	27.86	30.1	31.46
	RF = .75	15	15.43	30.12	33.93	33.26	33.83
	RF = 1	9.45	17.12	23.68	---	26.02	31.54
NC = 2.1	RF = .25	19.04	24.68	23.8	22.3	20.28	21.13
	RF = .5	17.35	23.8	26.92	30.24	28.34	29.56
	RF = .75	9.47	16.14	28.5	25	26.9	28.29
	RF = 1	12.07	20.76	---	23.7	22	24.49

جدول ۳: متوسط مقدار بهبود بدست آمده توسط روش ایجاد ستون

n = 30	θ	1	1.1	1.2	1.3	1.4	1.5
NC = 1.5	RF = .25	22.32	20.94	16.81	15.32	12.94	11.79
	RF = .5	26.13	30.93	28.32	27.43	25.95	27.42
	RF = .75	27.28	33.8	35.86	34.33	33.27	33.12
	RF = 1	28.48	34.12	38.04	19.8	27.79	33.89
NC = 1.8	RF = .25	17.27	16.75	15.78	13.42	10.48	10.5
	RF = .5	22.68	27.44	25.15	21.93	22.86	23.41
	RF = .75	25.35	23.89	29.44	30.93	28.24	26.81
	RF = 1	21.19	27.17	26.87	---	32	35.55
NC = 2.1	RF = .25	14.69	16.07	11.52	9.76	9.15	8.39
	RF = .5	19.91	26.51	24.91	23.09	20.75	19.81
	RF = .75	19.02	25.42	9.28	22.87	22.77	24.06
	RF = 1	24.88	28.83	---	25.87	30.53	32.86

برای تست الگوریتم ژنتیک مسائل فوق باهمان پارامتر های هزینه کیمز در نظر گرفته شده اند. کلیه 2880 مسئله فوق توسط این الگوریتم حل شده است و جدول مقدار بهبود این الگوریتم مشابه جداول بالا در زیر آمده است (جدول ۴).

در اجرای الگوریتم ژنتیک برای اینکه قادر بوده باشیم که در زمان معقول همه مسائل فوق را حل کنیم و در عین حال برای نشان دادن قدرت این روش سعی شده است که پارامتر های الگوریتم ژنتیک طوری انتخاب شود که حل هر مسئله حداکثر ۷ ثانیه زمان ببرد.

در اینجا پارامتر ها به قرار زیر است:

تعداد نسلها: ۵۰۰ نسل

احتمال جهش ۰٫۵

احتمال تقاطع: ۰٫۹

احتمال جستجو: ۰٫۱

همانطور که از جدول ملاحظه می شود به ازای کلیه مقادیر الگوریتم ژنتیک بهتر عمل می کند.

جدول ۴: متوسط بهبود UB با استفاده از الگوریتم ژنتیک.

n = 30	θ	1	1.1	1.2	1.3	1.4	1.5
NC = 1.5	RF = .25	32.06	39.43	41.85	43.14	43.79	44.12
	RF = .5	29.06	37.42	42.98	47.24	51.27	53.99
	RF = .75	31.38	38.19	43.4	47.95	51.27	54.53
	RF = 1	33.42	39.3	44.26	48.49	51.89	54.7
NC = 1.8	RF = .25	25.72	34.9	39.15	40.92	41.34	41.58
	RF = .5	26.55	36	41.79	46.59	49.94	52.7
	RF = .75	29.56	36.77	41.55	45.76	49.14	52.12
	RF = 1	26.08	33.82	39.32	43.45	47.23	50.56
NC = 2.1	RF = .25	21.9	33.55	37.44	38.89	39.1	39.4
	RF = .5	25.44	34.4	40.73	45.64	49.07	52.12
	RF = .75	23.5	31.97	37.89	41.94	45.91	49.06
	RF = 1	26.52	34.15	39.63	43.83	47.22	50.19

مقایسه جدول ۴ و جداول ۲ و ۳ نشان می دهد که برای تمام موارد بجز برای حالت NC = 1.5 و RF = 0.75 و θ = 1.2 روش الگوریتم ژنتیک بطور قابل ملاحظه ای از روشهای کیمز بهتر عمل می کند. برای حالت بالا درصد بهبود برای الگوریتم ژنتیک برابر است با 43.4 در صورتیکه برای روش ایجاد ستون این مقدار 35.86 است و برای روش لاگرانژ برابر است با 43.88. البته برای این حالت هر دو روش کیمز متوسط درصد بهبود را تنها برای ۲ مسئله در نظر گرفته اند.

۵) نتیجه:

با توجه به مقایسه الگوریتم ژنتیک و روشهای ارائه شده توسط کیمز و با توجه به اینکه در ارائه نتایج الگوریتم ژنتیک هیچگونه تنظیمی روی پارامتر های این الگوریتم صورت نگرفته است، امکان این وجود دارد تابا آزمایش روشهای دیگری برای طرح تقاطع و یا جهش و انجام تنظیمات مناسب روی پارامتر های الگوریتم ژنتیک، نتایج پربار تری بدست آید که می تواند عنوان تحقیقات بعدی قرار گیرد.

۶) مراجع:

1. Akpan, E.O.P. (1997). Optimal resource determination for project scheduling. Production Planning and Control, 8(5): 462-468.
2. Baker, K.(1974). Introduction to sequencing and scheduling. *John Wiely and Sons, Inc.*
3. Blazewics, J. (1978). Complexity of computer scheduling algorithms under resource constraints, in Proceedings, First Meeting of the AFCET on Applied Mathematics. Palaiseau, Poland. 169-178.
4. Cheng, R., and Gen M. (1997). Genetic algorithms and engineering design. John Wiely and Sons, Inc.
5. Drexl, A., Kimms, A., (2001), Optimization Guided Lower & Upper Bounds for the Resource Investment Problem. Journal of the Operational Re search Society Vol 52, PP. 340-351.
6. Hartmann. (1997). A competitive genetic algorithm for resource constrained project scheduling. Technical report 451, Manuscripte aus den Instituten fur Betriebswritschaftslehre der Universitat Kiel. *Naval Research Logistics.*
7. Holland, J.(1975). Adaptation in Natural and Artificial Systems. The university of Michigan Press.
8. Kelly, J. (1963). The critical path method: resource planning and scheduling. In Muth and Thompson, 347-365.
9. Kolisch, R., Spercher A., Drexl A. (1992). Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems. Institut fur Betriebswritschaftslehre, Universitat zu Kiel.
10. Mohring, R.H. (1984). Minimizing costs of resource requirements in project networks subject to a fix completion time. *Operations Research,* 32: 89-120.