فصلنامه مدیریت صنعتی دانشکده علوم انسانی دانشگاه آزاد اسلامی
واحد سنندج - سال ششم / شماره ۱۵ / بهار ۱۳۹۰

# حل مسئله زمانبندی پروژه با محدودیت منابع چند حالته (MRCPSP) با الگوریتم زنبورهای عسل

**امیر صادقی**
دانشجوی کارشناسی ارشد مهندسی صنایع دانشگاه پیام نور تهران (نویسنده مسوول) amir_sadeghi_ie@yahoo.com
**اعظم دخت صفی صمغ آبادی**
استادیار گروه مهندسی صنایع دانشگاه پیام نور تهران
**فرناز برزین پور**
استادیار گروه مهندسی صنایع دانشگاه علم و صنعت ایران

تاریخ دریافت: ۸۹/۸/۲۲ ❋ تاریخ پذیرش: ۹۰/۳/۱۱

## چکیده

مسئله زمانبندی پروژه با محدودیت منابع چندحالته[1] از جمله مسائل زمانبندی پروژه با محدودیت منابع است. در این مسئله هر پروژه از فعالیت‌هایی تشکیل شده است که بین آنها روابط پیش‌نیازی و محدودیت منابع مختلف از جمله منابع تجدید پذیر و تجدید ناپذیر وجود دارد. هر فعالیت دارای بیش از یک نوع حالت اجرا می‌باشد که هر حالت اجرا، مقدار منبع و مدت زمان انجام خاص خود را طلب می‌کند. این مسئله از جمله مسائل NP-Hard است، لذا همواره محققین در تلاش برای یافتن بهترین روش برای حل این مسائل بوده‌اند. در این پژوهش از الگوریتم زنبورهای عسل برای حل کمک گرفته شده است که تاکنون از این الگوریتم برای حل این دسته از مسائل استفاده نشده و این امر از جمله نوع‌آوریهای این پژوهش می باشد و سپس مسائل مختلف استاندارد موجود در ادبیات، با این الگوریتم مورد حل قرار گرفته و نتایج با بهترین و جدیدترین الگوریتم‌هایی که تاکنون برای این مساله مورد استفاده قرار گرفته‌اند مقایسه شده است تا عملکرد الگوریتم در حل این دسته از مسائل مورد بررسی قرار گیرد. نتایج حاکی از عملکرد موفق الگوریتم زنبورهای عسل در حل مسائل MRCPSP است.

## واژه‌های کلیدی:

زمانبندی پروژه، محدودیت منابع، چند حالته، الگوریتم زنبورهای عسل

[1]- Multi Mode Resource Constrained Project Scheduling (MRCPSP)

## ۱- مقدمه

تاریخچه مدیریت پروژه در جهان را معمولا به مدیریت پروژه‌های عظیمی همچون ساخت اهرام مصر، دیوار چین و یا بنا نهادن تخت جمشید به دستور داریوش مربوط می‌دانند اما تاریخچه مدیریت پروژه در دنیای جدید به سالهای ابتدایی دهه ۱۹۰۰ میلادی باز می‌گردد؛ امروزه با جهانی شدن تجارت، استفاده بهینه از منابع برای بقاء در صحنه‌های جهانی، امری ضروری است.از همین جهت است که مفاهیمی همچون پروژه، کنترل پروژه و زمانبندی پروژه با محدودیت منابع و ... مطرح شده است.

زمانبندی پروژه با محدودیت منابع از جمله مسائل NP-Hard است (Demeulemeester,E and Herroelen,W S.2002) . در این مسئله هر پروژه از تعدادی فعالیت تشکیل شده است، بعلاوه تعدادی منبع با ظرفیت های محدود در هر دوره زمانی وجود دارد. فعالیت ها علاوه بر اینکه نسبت به یکدیگر جهت اجرا دارای اولویت هستند، در استفاده از منابع نیز محدودیت دارند. هدف معمولا کمینه کردن زمان اتمام پروژه (Cmax) می باشد به نحوی که محدودیت های تقدمی و محدودیت منابع ارضاء شود. ساده ترین حالت این مسئله، مسئله پایه زمانبندی پروژه با محدودیت منابع است که در آن هر فعالیت تنها به یک طریق قابل انجام است و منابع نیز تنها منابع تجدیدپذیر هستند. همانطور که واضح است در عمل گاه یک فعالیت به بیش از یک روش قابل انجام است و می توان به عنوان نمونه با صرف منابع بیشتر و یا روش دیگر اجرا، آن فعالیت را به اتمام رسانید. لذا به این علت دسته‌ای دیگر از مسائل این مدل، تحت عنوان زمانبندی پروژه با محدودیت منابع چندحالته بوجود آمده است.

مسئله RCPSP چند حالته در ادبیات موضوع این مسئله به صورت MRCPSP نمایش داده می‌شود. فرض چند حالته بودن مسئله RCPSP به صورت زیر تعریف می شود: "یک فعالیت j باید در یک زمان شروع در یکی از حالات اجرایش که به صورت $1,...,M_j$ با $M_j$ حالت اجرا نمایش داده می‌شود، آغاز گردد. فعالیت با هر حالت اجرایی که شروع می‌شود باید با همان حالت هم اتمام یابد. در مدل پایه MRCPSP انقطاع جایز نیست. زمان اجرا فعالیت j در حالت m با $D_{jm}$ نمایش داده می‌شود. مقدار منبع نوع k مورد نیاز برای فعالیت j برای اجرا در حالت m با $r_{jmk}$ نمایش داده می‌شود. علاوه بر منابع تجدیدپذیر، منابع تجدیدناپذیر[1] هم اغلب در مدل‌های چندحالته به کار گرفته می‌شود. یک برنامه زمان‌بندی برای مدل MRCPSP با یک زمان شروع $S_j$ و در حالت $M_j$ برای هر فعالیت j تعیین می‌شود. مشهورترین تابع هدف در ادبیات موضوع این مسائل کمینه کردن زمان اتمام پروژه می‌باشد."واضح است که اگر تنها یک حالت برای هر فعالیت وجود داشته باشد و منبع تجدید ناپذیر وجود نداشته باشد، مسئله به مسئله RCPSP استاندارد تبدیل خواهد شد. به این نکته هم باید اشاره کرد که در صورت در نظر گرفتن منابع تجدید ناپذیر ممکن است زمان‌بندی شدنی نداشته باشیم که این امر توسط کولیش و درگزل (Kolisch, R. and Drexl, A. 1997) نشان داده شده است. مسئله بررسی وجود داشتن زمان‌بندی شدنی در این مسئله، خود یک نوع مسئله NP-Complete است. بر طبق نمادگذاری بروکر و همکاران (Brucker et al. 1999) در این حالت، پارامتر $\alpha$ با MPS شروع می‌شود.

مقالات فراوانی در این زمینه به چاپ رسیده است که می توان به موارد زیر اشاره کرد :

الکاراز و همکاران (Alcaraz et al. 2003) ، بولیمن و لکوکگ (Lecocq, H. and Bouleimen, K. 2003) ، هارتمن (Hartmann, S. 2001) ، جوزفسکا و همکاران(Jozefowska et al. 2001) ،ازدامار (Ozdamar, L. 1999) و پسچ (Pesch, E. 1999) و همچنین ورما و همکاران (Varma et al. 2007) که بر روی مدل چند حالته بدون منبع تجدید پذیر بحث کرده‌اند. مسئله چند حالته با روابط پیش نیازی کلی[2] توسط محققینی از جمله باریوس و همکاران (Barrios et al. 2009) و بروکر و کنوست (Brucker, P. and Knust, S. 2001) و کالون و همکاران (Calhoun et al. 2002) و سیدحسینی و سبزه پرور (Sabzehparvar, M. and Seyed-Hosseini, S.M. 2008) مورد بررسی قرار گرفته است. زوا و همکاران(Zhu et al. 2008) یک مسئله چندحالته با محدودیت‌های منابع کلی در نظر گرفته‌اند.

---

[1]- Non-Renewable

[2]- Generalized Precedence Relations (GPR)

نتاسمبون و راندهاوا (Nudtasomboon, N. and Randhawa, S.U. 1997) اجازه انقطاع (شکست) در فعالیت‌ها را در مسئله MRCPSP در نظر گرفتند. در مسئله MRCPSP پایه، هنگامیکه یک فعالیت برای اجرا، دوباره از سر گرفته می‌شود حالت اجرایی آن همان حالت قبل از شکست بوده و امکان تغییر حالت اجرا وجود ندارد.

کاربردی بودن این مسئله باعث توجه بیش از پیش محققین به این مسئله در سال‌های اخیر شده است. این دسته از مسائل در مهندسی صنایع به دو دلیل مورد توجه قرار گرفته‌اند: اول اینکه با توجه به شرایط متفاوت کاربردی و صنعتی از نظر تابع هدف، خصوصیات فعالیت‌ها، منابع و نوع روابط پیش‌نیازی بسیار متنوع‌اند و دوم اینکه با توجه به NP-hard بودن این مسائل محققین همواره به دنبال ارائه راه حل‌های کارآتری برای حل این دسته از مسائل بوده‌اند. مساله MRCPSP از جمله مسائل بهینه‌سازی ترکیبی[1] می‌باشد، اهدافی از قبیل بیشینه کردن نرخ ارزش فعلی [2] و اهداف کیفی و مینیمم کردن هزینه و... هم در مدلسازی این مسئله در نظر گرفته شده است.

تالبوت (Talbot, F.B. 1982) یک مدل برنامه ریزی صفر و یک برای این مسائل ایجاد کرده است. همانطور که بیان شد در این مسئله یک فعالیت در حالت‌های مختلفی قابل انجام است که هر حالت زمان انجام فعالیت و مقدار منبع‌های مورد نیاز خاص خود را می طلبد. متغیر تصمیم $X_{imt}$ این مدل بصورت زیر است :

$$X_{imt} = \begin{cases} 1 & if\ i\ \ in\ Modem\ \ finishedat\ t \\ 0 & otherwise \end{cases}$$

و مسئله به صورت زیر است:

$$\min \sum_{t=es_n}^{ls_n} t.x_{n1t}$$

S.t.

$$\sum_{m=1}^{M_i} \sum_{t=es_{ii}}^{ls_i} x_{imt} = 1 \qquad i = 1,2,\ldots,n \qquad (1)$$

$$\sum_{m=1}^{M} \sum_{t=es_i}^{ls_i} (t + d_{im}) x_{imt} \le \sum_{m=1}^{M_i} \sum_{t=es_{ii}}^{ls_j} t x_{jmt} \qquad (i,j) \in E \qquad (2)$$

$$\sum_{i=1}^{n} \sum_{m=1}^{M_i} r_{imk}^{\rho} \sum_{s=\max\{t-d_{im},es_i\}}^{\min\{t-1,ls_i\}} x_{ims} \le a_k^{\rho} \qquad for k = 1,..,K^{\rho}, t = 1,..,T \qquad (3)$$

$$\sum_{i=1}^{n} \sum_{m=1}^{M_i} r_{imk}^{\upsilon} \sum_{s=es_i}^{ls_i} x_{ims} \le \alpha_k^{\upsilon} \qquad for\ k = 1,2,\ldots,K^{\upsilon} \qquad (4)$$

$$x_{imt} \in \{0,1\} \qquad i = 1,\ldots,n\ , m = 1,2,\ldots,M_i \quad t = es_i,\ldots,ls_i \qquad (5)$$

در مدل بالا $es_i(ls_i)$ به ترتیب زودترین و دیرترین زمان شروع فعالیت i بر اساس مدلهای اجرا با کمترین زمان ممکن است. $d_{im}$ زمان انجام فعالیت در حالت اجرایی m است. $r_{imk}^{\rho}$ مقدار منبع تجدید پذیر نوع k برای انجام فعالیت Iدر حالت اجرایی m است. $\alpha_k^{\upsilon}$ سطح دسترسی منابع تجدید پذیر نوع k است. E مجموعه روابط پیش نیازی است و شبکه نیز از نوع AON است و T نیز حد بالای زمان انجام پروژه است. تابع هدف هم همان مینیمم کردن زمان اتمام پروژه است. به این نکته نیز باید اشاره شود که دو فعالیت مجازی ابتدا و انتهای پروژه با زمان انجام صفر وجود دارد که نشان دهنده شروع و پایان پروژه هستند.

محدودیت (۱) مشخص می‌کند که هر فعالیت تنها یک حالت اجرا و یک زمان شروع دارد و فعالیت با هر روش اجرایی که شروع می‌شود با همان روش اجرا باید خاتمه یابد. محدودیت (۲)، محدودیت روابط پیش‌نیازی است که مانند مدل RCPSP پایه تنها

[1]- Combinatorial Optimization Problem
[2]- Net Present Value

روابط پیش‌نیازی از نوع FS (پایان به شروع) را دارا می‌باشند. محدودیت (۳) سطح دسترسی منابع تجدیدپذیر است که معین می‌کند از این سطح تجاوزی در هر دوره صورت نگیرد و محدودیت (۴) نیز محدودیت منابع تجدید ناپذیر است و یکی از مشخصه‌های مهم مسائل MRCPSP و تفاوت عمده آن با مسائل RCPSP پایه این است که این سری از مسائل دارای این چنین منابعی می‌باشند که مقدار آنها برای کل پروژه یک مقدار معین است و در هر دوره تجدید نمی‌گردد. از جمله این منابع می-توان به بودجه پروژه و.... را اشاره کرد و در آخر مدل، محدودیت متغیر تصمیم (۵) وجود دارد که این متغیر، باینری است که در صورتی که فعالیت i در حالت اجرایی m در زمان t انجام گیرد مقدار یک و در غیر اینصورت مقدار صفر را بخود می گیرد. به این نکته نیز باید اشاره شود که مسئله MRCPSP همانند مسئله RCPSP یک مسئله NP-Hard که این امر توسط کولیش در سال ۱۹۹۵ و اسچومر در سال ۱۹۹۶ نشان داده شد. هدف از این پژوهش استفاده از الگوریتم فرابتکاری جدیدی به نام الگوریتم زنبورهای عسل به منظور دست‌یابی به نتایج بهتر در حل این مسئله کاربردی است تا بتوان این مسئله را در مسائل دنیای واقعی با بدست آوردن نتایجی مطلوب به کار برد.

## ۲- مواد و روش‌ها

این مقاله از نوع کاربردی است و روش تحقیق کمی است. الگوریتم زنبورهای عسل نتایج موفقیت آمیزی در حل مساله زمانبندی پروژه با منابع محدود تک‌حالته توسط صادقی و همکاران (Sadeghi et al., 2011) بدست آورد، لذا با فرض موثر بودن الگوریتم زنبورهای عسل در حل مسئله MRCPSP این تحقیق صورت گرفته است که نتایج بر این فرض، صحه نهاده است.
ابتدا به معرفی جامع الگوریتم زنبورهای عسل پرداخته شده است و سپس الگوریتم پیشنهادی با بکارگیری عملگرهایی که برای مسئله MRCPSP طراحی شده اند، در چارچوب کلی الگوریتم زنبورهای عسل به حل این مسئله می پردازد.
این الگوریتم مبتنی بر جمعیت به نام الگوریتم زنبور عسل (BA) نامیده شده است. این الگوریتم از رفتار زنبورهای عسل برای کاوش غذای تقلید کرده است در مدل پایه آن از جستجوی همسایگی ادغام شده با جستجوی تصادفی استفاده می کند و می توان از آن برای هر دو نوع بهینه یابی ترکیبی و بهینه یابی تابعی استفاده کرد. در بخش ۲-۳ تعدادی مسائل مبنا که کارآیی و نیرومندی این الگوریتم جدید را ثابت کرده است ارائه شده و همچنین با سایر الگوریتمهای قدیمی مقایسه شده تا کیفیت جواب این الگوریتم نمایان شود.

### - الگوریتم زنبورهای عسل

بسیاری از بهینه سازی های پیچیده چند متغیره نمی تواند مستقیما در زمانهای محاسباتی محدود بصورت چند جمله ای حل شوند. این امر موجب بوجود آمدن الگوریتمهای جستجویی شده که راه حل های نزدیک به بهینه را در زمانهای اجرایی قابل قبول پیدا کنند. برخی از سیستمهای طبیعی (کلونی حشرات اجتماعی) به ما یاد می دهند که یک سری ارگانیسمهای ساده ی خارجی قابلیت تولید سیستمهای را دارند که به ما کمک بر هم کنشهای پویا قابلیت انجام اعمال بسیار پیچیده را دارند. این الگوریتم از رفتار جستجوی زنبوران برای غذا نشات گرفته است و می توان آن را جزء "ابزارهای شبیه سازی هوشمند" طبقه‌بندی کرد. گروه زنبورها به خاطر استقلال داخلی کلونی و عملکردهای توزیع شده و سیستم درون سازمانی یکی از بهترین کلونی‌ها برای توضیح این مسائل شناخته شده است، که می توان این مدل را هم برای مسائل با عدم قطعیت نیز به کار برد(Pham et al. 2007).

الگوریتم زنبور مبتنی بر الگوریتم جستجو است که اولین بار در سال ۲۰۰۵ توسعه یافت. این الگوریتم شبیه سازی رفتار جستجوی غذای گروههای زنبور عسل است. در نسخه ابتدایی این الگوریتم، الگوریتم نوعی از جستجوی محلی انجام می دهد که با جستجوی تصادفی[1] ترکیب شده و می تواند برای بهینه سازی ترکیبی {زمانی که بخواهیم چند متغیر را همزمان بهینه کنیم.} و یا بهینه سازی تابعی به کار رود. یک کلونی زنبور عسل می تواند در مسافت زیادی و نیز در جهت های گوناگون

[1]- Random

پخش شود تا از منابع غذایی بهره برداری کند. قطعات گلدار با مقادیر زیادی نکتار و گرده که با تلاشی کم قابل جمع آوری است،به وسیله ی تعداد زیادی زنبور بازدید می شود؛ به طوری که قطعاتی از زمین که گرده یا نکتار کمتری دارد، تعداد کمتری زنبور را جلب می کند(Pham et al. 2007).

پروسه ی جستجوی غذای یک کلونی به وسیله ی زنبورهای دیده بان آغاز می شود که برای جستجوی گلزار های امید بخش {دارای امید بالا برای وجود نکتار یا گرده}فرستاده می شوند. زنبورهای دیده بان به صورت تصادفی از گلزاری به گلزار دیگر حرکت می کنند.در طول فصل برداشت محصول{گل دهی}، کلونی با آماده نگه داشتن تعدادی از جمعیت کلونی به عنوان زنبور پیشتاز (دیده بان) به جستجوی خود ادامه می دهند.هنگامی که جستجوی تمام گلزار ها پایان یافت، هر زنبور دیده بان، بالای گلزاری که اندوخته ی کیفی مطمئنی از نکتار و گرده دارد، رقص خاصی را اجرا می کند. این رقص که به نام "رقص چرخشی" (حرکتی مانند حرکت قرقره) شناخته می شود، اطلاعات مربوط به جهت تکه گلزار(نسبت به کندو)، فاصله تا گلزار و کیفیت گلزار را به زنبور های دیگر انتقال می دهد. این اطلاعات زنبور های اضافی و پیرو را به سوی گلزار می فرستد. بیشتر زنبور های پیرو به سوی گلزار هایی میروند که امید بخش تر هستند و امید بیشتری برای یافتن نکتار و گرده در آنها، وجود دارد. وقتی همه ی زنبور ها به سمت ناحیه ای مشابه بروند، دوباره به صورت کتره ای و به علت محدوده ی رقصشان در پیرامون گلزار پراکنده می شوند تا به موجب این کار سرانجام نه یک گلزار، بلکه بهترین گل های موجود درون آن تعیین موقعیت شوند(Pham et al. 2007).

الگوریتم زنبور هر نقطه را در فضای پارامتری(متشکل از پاسخ های ممکن) به عنوان منبع غذا تحت بررسی قرار می دهد." زنبورهای دیده بان" کارگزاران شبیه سازی شده به صورت کتره ای فضای پاسخ ها را ساده می کنند و به وسیله ی تابع شایستگی کیفیت موقعیت های بازدید شده را گزارش می دهند. جواب های ساده شده رتبه بندی می شوند، و دیگر "زنبورها" نیروهای تازه ای هستند که فضای پاسخ ها را در پیرامون خود برای یافتن بالاترین رتبه محل ها جستجو می کنند{که "گلزار" نامیده می شود} الگوریتم به صورت گزینشی دیگر گلزارها را برای یافتن نقطه ی بیشینه‌ی تابع شایستگی جستجو می کند. (Pham et al. 2006)

این الگوریتم نیازمند پارامترهایی می باشد که عبارتند از :

n =تعداد زنبوران پیشتاز

m = تعداد محل های انتخاب شده خارج از مناطق بررسی شده قبلی

e = تعداد بهترین محل های انتخاب شده خارج از m محل انتخاب شده

nep = e تعداد زنبوران بکار گرفته شده برای بهترین مکانهای

nsp = ( m-e محل باقیمانده) تعداد زنبوران بکار گرفته شده برای دیگر محل های انتخاب شده

ngh= اندازه اولیه منطقه ها که شامل مکان ، قسمتهای مجاور آن و معیار توقف می باشد

این الگوریتم با n سرباز دیده بان ( پیشتاز ) که به طور تصادفی در فضای جستجوی قرار دارند کار خود را آغاز می‌کند. مناسب بودن مکان‌های بررسی شده توسط سربازان دیده بان در مرحله ی ۲ ارزیابی می شود(Pham et al. 2006).

مرحله ۱ : جمعیت اولیه با راه حل های تصادفی

مرحله ۲ : ارزیابی سازگاری جمعیت

مرحله ۳ : شکل گیری جمعیت جدید

مرحله ۴ : انتخاب مکانها برای جستجوی قسمتهای مجاور

مرحله ۵ : بکار گرفتن زنبوران برای محل های انتخاب شده و ارزیابی آنها

مرحله ۶ : انتخاب مناسبترین زنبوران (زنبوران شایسته) از هر منطقه

مرحله ۷ : تعیین زنبوران باقی مانده برای جستجوی تصادفی و ارزیابی سازگاری شان

مرحله ۸ : پایان

در مرحله ۴ زنبورهایی که بیشترین صلاحیت را دارند به عنوان " زنبوران انتخابی " انتخاب می‌شوند و مکانها توسط آنها بررسی شده برای جستجوی همسایگی انتخاب می شوند، سپس در مراحل ۵ و ۶ الگوریتم جستجوها در مناطق مجاور از مکانهای انتخابی را انجام می دهند . بیشتر زنبورها برای جستجو در نزدیکی بهترین مکانها تخصیص داده می شوند . زنبورها می توانند به طور مستقیما مطابق با صلاحیت وابسته به همراه مناطقی که بررسی کرده اند انتخاب شوند . داده های صلاحیت برای تعیین احتمال زنبورهایی که انتخاب می شوند استفاده می گردند . جستجوها در مناطق همسایگی بهترین مکانهای e راه حل های بهتری را ارائه می دهند . به همراه این گشت زنی این نفرگیری تفاضلی عملیات کلیدی الگوریتم رنبورهای عسل می باشد .
اگرچه در مرحله ۶ برای هر منطقه فقط زنبورهایی با بیشترین صلاحیت به شکل جمعیت بعدی زنبوران عسل انتخاب خواهند شد، در طبیعت چنین محدودیتی وجود ندارد این محدودیت در اینجا برای کاهش تعداد نقاط کشف شده می باشد .در مرحله ۷ زنبوران باقی مانده در جمعیت بطور تصادفی در اطراف فضای جستجو برای راه حل های بالقوه تخصیص می یابند. این مراحل جداگانه هستند تا زمانیکه معیار توقف ایجاد شود و در پایان هر تکرار، کلنی دو قسمت در جمعیت جدیدش خواهد داشت. نماینده هایی از هر منطقه انتخابی و زنبوران دیده بان دیگر برای انجام جستجوهای جدید تخصیص یافته اند(Pham et al. 2006).
به منظور آزمایش الگوریتم زنبورهای عسل، در مقاله (Pham et al,2006) الگوریتم زنبورهای عسل در هشت تابع معیار بکاربرده شد و نتایج با موارد بدست آمده از الگوریتم‌های دیگر مقایسه شده اند.
اولین تابع آزمایشی Dejang بود. الگوریتم زنبور عسل موفق شد، حد مطلوبیت را ۱۲۰ بار سریعتر از الگوریتم مورچگان و ۲۰۷ مرتبه سریعتر از GA با میزان موفقیت ۱۰۰ ٪ بدست آورد. تابع بعدی Goldstein & Price بود. الگوریتم زنبور عسل تقریبا ۵ مرتبه سریع‌تر از الگوریتم مورچگان و ژنتیک به حد مطلوب رسید و با موفقیت ۱۰۰٪ همراه بود. الگوریتم در مورد تابع Branin, ۱۵٪ بهسازی در مقایسه با الگوریتم مورچگان و ۷۷ ٪ بهسازی در مقایسه با GA و همچنین ۱۰۰٪ موفق بود. (Pham et al. 2006).
تابعهای ۵ و۶ تابعهای Rosenbrack در ابعاد دو و چهار بودند. در هر دو تابع دو بعدی، الگوریتم زنبورهای عسل ۱۰۰ ٪ موفق بود. در چهار مورد بعدی به ارزیابی بیشتری برای رسیدن به حد مطلوب احتیاج بود. الگوریتم SA توانست حد مطلوبی با ۱۰ برابر ارزیابی کمتر پیدا کند اما میزان موفقیت فقط ۹۴٪ بود و الگوریتم مورچگان حد مطلوبی با ۱۰۰٪ موفقیت و ۳/۵ برابر سریعتر از الگوریتم زنبور عسل را بدست آورد (Pham et al. 2006). تابع بعدی، مدل hyper sphere شش بعدی بود. الگوریتم زنبور عسل نیمی از تعداد تابع های ارزیابی شده را در مقایسه با GA و یک سوم آنچه را که برای الگوریتم مورچگان احتیاج بوده را لازم داشت تا به جواب بهینه برسد. تابع آزمایشی ۸ ام، ده بعدی بود. الگوریتم زنبور عسل توانست ۱۰ مرتبه سریعتر از GA و ۲۵ مرتبه سریع‌تر از الگوریتم مورچگان به حد مطلوب برسد و میزان موفقیت آن هم ۱۰۰٪ بود (Pham et al. 2006).
حال پس از بررسی عملکرد موفق این الگوریتم در حل این توابع پایه، الگوریتم زنبورهای عسل برای مسئله پیشنهادی مورد استفاده قرار گرفته است.

- **الگوریتم زنبورهای عسل پیشنهادی برای مسئله MRCPSP**

در ابتدا به تنظیم پارامترهای این الگوریتم برای حل مسئله MRCPSP پرداخته شده و سپس گام‌های الگوریتم بیان شده است. همانطور که در پیش‌تر بیان شد، این الگوریتم به پارامترهایی نیاز دارد که این پارامترها از طریق حل مسائل مختلف، برای مسائل نمونه MRCPSP تنظیم شده است که در بخش نتایج بطور کامل چگونگی این تنظیمات بیان شده است.

- **گام‌های الگوریتم**

گام ۱: ایجاد یک ترتیب ممکن از انجام فعالیت‌ها با رعایت پیش‌نیازها و محدودیت منابع

در این گام بدین صورت عمل می‌گردد که ابتدا به تعداد فعالیت‌های پروژه اعداد تصادفی بین ۰ و ۱ تولید می شود سپس این اعداد مرتب گردیده و در یک لیست ذخیره می‌گردند و همچنین لیست این ترتیب مرتب شده آنها که نماینده شماره فعالیت است در لیست دیگری ذخیره می‌گردد. واضح است که ممکن است این جواب به علت رعایت نکردن پیش‌نیازها، نشدنی باشد، لذا بدین صورت عمل می‌گردد که اولین فعالیت از لیست دوم انتخاب می‌شود در صورتیکه پیش‌نیازی نداشته باشد و یا اینکه پیش-نیازهایش انجام گرفته باشند آن فعالیت انجام می‌گیرد و در غیر اینصورت به سراغ فعالیت بعدی در لیست ذکر شده درج می‌شود. پس از اینکه فعالیت انجام گرفت، زمان شروع (ST)[1] و پایان (FT)[2] و Cmax فعالیت ذخیره می‌گردد و این فعالیت از لیست پیش‌نیازها[3] و پس‌نیاز[4] سایر فعالیت‌ها، حذف می‌گردد.

گام ۲: ذخیره بهترین جواب تولید شده در گام ۱، بهترین جواب تولید شده در گام ۱ در محلی بنام BestSol ذخیره می‌گردد.

گام۳ : گام تکرار الگوریتم

تا زمانیکه معیار توقف برای مساله رعایت نشده گامهای ۴ الی ۶ تکرار می‌گردد. لازم به ذکر است که با توجه به ادبیات موضوع و مقاله‌های چاپ شده معیار توقف در سه حالت در نظر گرفته شده است که در بخش نتایج محاسباتی به آن اشاره شده است.

گام ۴ : جستجوی همسایگی و رقص زنبورهای عسل

زنبورهای پیشتاز با توجه به جواب تولید کرده که در واقع همان زمان اتمام پروژه (Cmax) می‌باشد با احتمالی که در جدول زیر آمده است به رقص می‌پردازند. میزان رقص زنبور i نیز با توجه به روابط زیر تعیین می شود و مدت زمان رقص با $d_i'$ نمایش داده می‌شود.

$$d_i' = \frac{Pf_i}{Pf_{colony}} \quad \rightarrow \quad , \quad Pf_i = \frac{1}{C_{\max}^i} \; Pf_{colony} = \frac{1}{n}\sum_{j=1}^{n} \frac{1}{C_{\max}^i}$$

در ابتدا از طریق معکوس کردن نتیجه بدست آمده از زنبورهای پیشتاز مقدار $Pf_i$ که تحت عنوان " نرخ سودآوری زنبور پیشتاز iام " تعریف شده است، بدست می‌آید. این امر واضح است که هر چه میزان Cmax کمتر باشد جواب بهتر بوده و میزان $\boldsymbol{Pf_i}$ نیز بهتر خواهد بود و سپس از میانگین تمامی $Pf_i$، $Pf_{colony}$ که تحت عنوان "متوسط نرخ سودآوری زنبورها" تعریف شده است بدست خواهد آمد و در نهایت از تقسیم $Pf_i$ بر $Pf_{colony}$ مدت زمان رقص زنبور پیشتاز iام بدست خواهد آمد. سپس بر اساس جدول زیر زنبورهای پیشتاز تصمیم به برگشت به کندو، رقصیدن و بسیج کردن زنبورهای تازه کار برای جستجوی همسایگی و یا عدم برگشت به کندو و جستجو تصادفی در سایر فضای جواب می‌گیرند.

جدول(۴): احتمال عدم رقصیدن و جستجوی تصادفی سایر نقاط (Wang et al. 2008)

| rate | $r_i$ |
|---|---|
| $Pf_i \le 0.9Pf_{colony}$ | 0.6 |
| $0.9Pf_{colony} \le Pf_i \le 095Pf_{colony}$ | 0.2 |
| $0.95Pf_{colony} \le Pf_i \le 1.15Pf_{colony}$ | 0.02 |
| $Pf_i \ge 1.15Pf_{colony}$ | 0.00 |

در واقع احتمال بازگشت به کندو و رقصیدن یک زنبور پیشتاز برای فراخواندن زنبورهای تازه کار[5] برای جستجو در همسایگی جواب بدست آمده توسط آن زنبور پیشتاز برابر $(1-r_i)\times 100\%$ است. واضح است که اگر کیفیت جواب یافته شده توسط یک زنبور پیشتاز نسبت به کل جواب‌هایی که زنبورهای پیشتاز دیگر یافته اند، ۱/۱۵ برابر باشد آن زنبور با احتمال ۱۰۰ درصد به کندو

[1]- Start Time
[2]- Finish Time
[3]- Predecessor list
[4]- Successor list
[5]- Recruit Bees

بازگشته و زنبورهای تازه‌کار را برای جستجو در همسایگی این جواب و دست یافتن به جواب بهتر در صورت امکان، بسیج می کند. از بین این نتایج بدست آمده توسط زنبورها، تعدادی به عنوان محل های انتخابی و از بین این محل‌ها نیز تعدادی به عنوان بهترین محل‌های انتخابی مشخص می گردند. لازم به ذکر است که یک نرخ همسایگی که نرخی کاهشی می‌باشد در نظر گرفته شده است که رفته رفته شعاع همسایگی را کاهش می دهد تا تاکید بر جستجو در نزدیکترین نقاط به بهترین جوابهای پیدا شده، شدت یابد.

گام ۵ : تخصیص تعداد زنبورهای تازه کار برای جستجوی همسایگی

تعداد زنبورهای تازه کاری که برای جستجوی همسایگی انتخاب می شوند نیز وابسته به کیفیت جواب و در واقع میزان رقص زنبور پیشتاز دارد که از طریق فرمول زیر محاسبه می شود :

$$n^{i}_{dance} = [d'_i . nDance0]$$

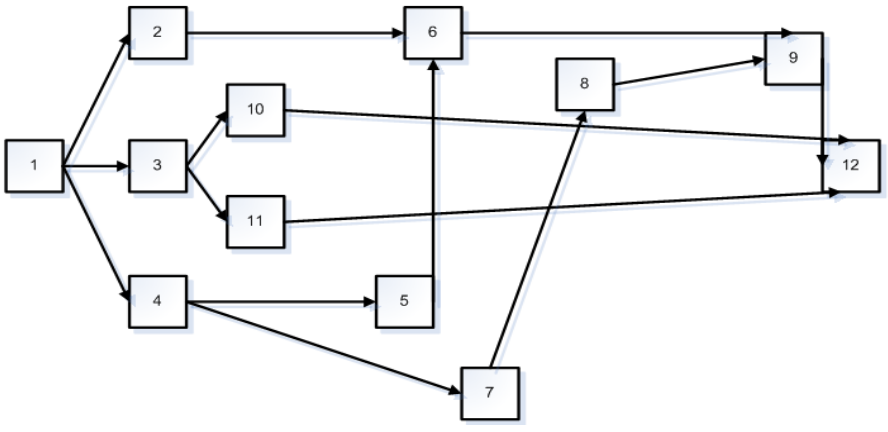
که $nDance0$ یک ضریب ثابت و $d'_i$ مدت زمان رقص زنبورعسل پیشتاز i ام است که در بالا محاسبه شده است و $n^{i}_{dance}$ ، تعداد زنبورهای تازه کار تخصیص یافته برای جستجوی همسایگی را تعیین می‌کند.

گام ۶ : ( گام پایانی ) بازگشت به گام ۲

در ادامه به منظور شفاف سازی عملکرد الگوریتم یک مثال نمونه از سری مسائل استاندارد موجود در ادبیات این موضوع تحت عنوان J104-5 که بصورت کتابخانه آنلاین با نام PSPLIB[1] در دسترس عموم قرار گرفته است، انتخاب گشته و در ادامه روند عملکرد الگوریتم در حل این مثال نمونه بطور کامل بیان شده است.

- **یک مثال عددی برای بررسی عملکرد الگوریتم زنبورهای عسل در حل مسائل MRCPSP**

برای بررسی نحوه عملکرد الگوریتم پیشنهادی یک مسئله نمونه از مسائل استاندارد این دسته از مسائل (PSPLIB) تحت عنوان J104-5 از سری مسائل MRCPSP انتخاب شده است که شبکه پروژه آن در زیر آورده شده است:




شکل (۱): شبکه مثال نمونه

در این مسئله نمونه، تعداد ۱۰ فعالیت اصلی و ۲ فعالیت مجازی (جمعاً ۱۲ فعالیت) که تمامی فعالیتها دارای ۳ مد (حالت) اجرا و فعالیت های مجازی ۱ و ۱۲ نیز دارای ۱ مد اجرا می باشند و دو منبع تجدید پذیر و دو منبع تجدید ناپذیر وجود دارد که وضعیت دسترسی این منابع بصورت زیر است :

[1]- http://129.187.106.231/psplib/

$$R_1 = 4 \qquad R_2 = 11 \qquad N_1 = 50 \qquad N_2 = 61$$

مسئله برای یک تکرار اجرا شده است و تعداد زنبورهای پیشتاز اولیه نیز ۱۰ عدد در نظر گرفته شده است لذا ۱۰ جواب اولیه خواهیم داشت که هر یک دارای یک جواب $\boldsymbol{C_{max}}$ خواهد بود. مقادیر این جوابها در جدول زیر بصورت صعودی مرتب شده اند:

جدول(۵): مقادیر تابع هدف هر زنبور پیشتاز مثال نمونه

| زنبور | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $C_{\max}$ | 32 | 37 | 38 | 41 | 42 | 50 | 50 | 50 | 50 | 53 |

همانطور که مشخص است بهترین جواب $C_{\max} = 32$ است که این جواب به عنوان بهترین جواب انتخاب شده و در محلی تحت عنوان BestSol ذخیره می گردد. روند تولید این جواب بصورت زیر بوده است :

ابتدا به تعداد فعالیت ها که در این مسئله ۱۰ فعالیت اصلی و ۲ فعالیت مجازی ( جمعاً ۱۲ فعالیت) است اعدادتصادفی بین $[0,1]$ تولید می شود و در محلی بنام q ذخیره می گردد اعداد تصادفی تولید شده برای این بهترین جواب بصورت زیر بوده است :

$$q = [0.932\ , 0.8351\ , 0.8954\ , 0.5825\ , 0.5827\ , 0.8549\ , 0.0349\ , 0.8854\ , 0.1548\ , 0.0364\ , 0.7461\ , 0.4077]$$

بعلت اینکه فعالیت ها دارای مدهای مختلف اجرا هستند به همین علت یکسری اعداد تصادفی دیگر نیز تولید و در لیستی بنام a ذخیره می شود تا بعداً برای انتخاب مد اجرا فعالیت مورد استفاده قرار می‌گیرد. اعداد تصادفی تولید شده به صورت زیر بوده است:

$$a = [0.16,\ 0.4181\ ,\ 0.333\ ,\ 0.332\ ,0.2254,\ 0.5624\ ,0.3183\ ,0.5017\ ,\ 0.125,\ 0.1125,\ 0,\ 0.3840]$$

ابتدا به روند کار برای انتخاب فعالیت ها پرداخته شده است و سپس روند انتخاب مد اجرایی بیان خواهد شد.

بعد از تولید لیست q (اعداد تصادفی) این اعداد از چپ به راست شماره های ۱ تا ۱۲ را به خود اختصاص می دهند و سپس این اعداد بصورت صعودی مرتب می شوند و در لیستی تحت عنوان L ذخیره می گردند لذا لیست L بعد از مرتب شدن بصورت زیر خواهد بود:

$$\boldsymbol{L} = [7\ ,\ 10\ ,\ 9\ ,\ 12\ ,\ 4\ ,\ 5\ ,\ 11\ ,\ 2\ ,\ 6\ ,\ 8\ ,\ 3\ ,\ 1]$$

مشخص است که این لیست و این ترتیب اجرای فعالیت‌ها بدلیل روابط پیش‌نیازی و محدودیت منابع قابل انجام نخواهد بود و ممکن است نشدنی باشد، لذا تغییراتی باید صورت گیرد تا این جواب به یک جواب شدنی تبدیل گردد برای این منظور از لیست L بترتیب از چپ به راست فعالیت ها تک تک مورد بررسی قرار می گیرند و اولین فعالیتی که پیش نیازهایش رعایت شده باشد انتخاب می گردد و در لیست جدیدی به نام $L_2$ ذخیره می گردد.و بهمین صورت این کار ادامه می یابد تا تمام فعالیتها انتخاب گردند و لیست $L_2$ کامل شود.

بر این اساس با توجه به پیش نیازها در مثال مورد بررسی از چپ به راست تنها در ابتدا فعالیت ۱ قابل انجام خواهد بود که در آخر لیست L است لذا این فعالیت انتخاب می گردد و لیست $L_2$ بصورت زیر خواهد شد :

$$L_2 = [1,0,0,0,0,0,0,0,0,0,0,0]$$

حال فعالیت ۱ از لیست پیش‌نیازها و یا پس‌نیازهای سایر فعالیت ها حذف و این لیست ها بروز می گردد و دوباره از چپ به راست از لیست L انتخاب فعالیت صورت می گیرد. با توجه به شبکه پروژه واضح است که بعد از فعالیت ۱ تنها فعالیتهای ۲و۳و۴ قابل اجرا خواهند بود که بعلت اینکه فعالیت ۴ در لیست L جلوتر از فعالیت های ۲و۳ است این فعالیت انتخاب می گردد و لیست $L_2$ بصورت زیر خواهد شد:

$$L_2 = [1,4,0,0,0,0,0,0,0,0,0,0]$$

این موارد آنقدر تکرار می گردد تا لیست $L_2$ بصورت زیر کامل می گردد.

$$L_2 = [1,4,7,5,2,\ 6\ ,\ 8\ \ ,\ 9\ ,\ 3\ ,\ 10\ ,\ 11,\ 12]$$

همانطور که مشهود است این جواب، یک جواب کاملا شدنی است. حال بعد از انتخاب فعالیت ها باید مد انجام آنها انتخاب گردد که این امر بصورت زیر انجام خواهد گرفت:

ابتدا تعداد مدهای قابل انجام برای هر فعالیت مشخص می گردد و در لیستی تحت عنوان nMode ذخیره می گردد. تعدادی از مدهای مسائل نمونه به دلیل احتیاج به منابع که سطح دسترسی بیش از میزان حداکثر در دسترس است حذف می گردد لذا این لیست برای این مسئله نمونه بصورت زیر خواهد بود :

$$nMode=[1\ ,1\ ,\ 2\ ,\ 2\ ,\ 3\ ,\ 3\ ,\ 3\ ,3\ ,\ 1\ ,\ 3\ ,\ 3\ ,1]$$

حال مد اجرا از این مدهای قابل انجام بصورت تصادفی با کمک گرفتن از اعداد تصادفی لیست a و فرمولی که در زیر آورده شده است انتخاب می گردد. مشخص است بعلت تصادفی بودن اعداد a در هر تکرار مدهای مختلفی ممکن است انتخاب گردد. فرمول مربوطه بصورت زیر است :

$$m_i = \min\ ([nMode\times(a_i)\ +1],\ nMode(i)\ )$$

این لیست یعنی $m_i$، شماره مد انتخاب شده برای فعالیت i را نشان می دهد و نام مد انتخاب شده در لیست دیگر به نام m قرار می گیرد. مقادیر $m_i$ برای هر فعالیت بصورت زیر محاسبه می گردد :

$$\begin{aligned}
a_1 = 0.16 \quad &\Rightarrow \quad \min\ ([\ 1\times 0.16+1],\ 1) = 1\\
a_2 = 0.4181 \quad &\Rightarrow \quad \min\ ([\ 1\times 0.4181+1],\ 1) = 1\\
a_3 = 0.3330 \quad &\Rightarrow \quad \min\ ([\ 2\times 0.333+1],\ 2) = 1\\
a_4 = 0.3320 \quad &\Rightarrow \quad \min\ ([\ 2\times 0.332+1],\ 2) = 1\\
a_5 = 0.2254 \quad &\Rightarrow \quad \min\ ([\ 3\times 0.2254+1],\ 3) = 1\\
a_6 = 0.5624 \quad &\Rightarrow \quad \min\ ([\ 3\times 0.5624+1],\ 3) = 2\\
a_7 = 0.3183 \quad &\Rightarrow \quad \min\ ([\ 2\times 0.3183+1],\ 3) = 1\\
a_8 = 0.5017 \quad &\Rightarrow \quad \min\ ([\ 3\times 0.5017+1],\ 3) = 2\\
a_9 = 0.0125 \quad &\Rightarrow \quad \min\ ([\ 2\times 0.0125+1],\ 1) = 1\\
a_{10} = 0.1125 \quad &\Rightarrow \quad \min\ ([\ 3\times 0.1125+1],\ 3) = 1\\
a_{11} = 0 \quad &\Rightarrow \quad \min\ ([3\times 0+1],\ 3) = 1\\
a_{12} = 0.384 \quad &\Rightarrow \quad \min\ ([\ 1\times 0.384+1],\ 1) = 1
\end{aligned}$$

لذا لیست $m_i$ به صورت زیر خواهد بود که نمایانگر شماره مد انتخاب شده از لیست مدهای شدنی است :

$$m_i = [1,\ 1\ ,1\ ,1\ \ ,1\ ,\ 2\ ,1\ ,\ 2\ \ ,1\ \ ,1\ ,1\ \ ,\ 1]$$

حال با توجه به مدهای قابل انجام که در زیر لیست شده اند مدهای انتخابی در لیست m بصورت زیر است :

$$\tilde{M} = [\ \overset{j=1}{[1]}\ \ \overset{j=2}{[\ 1\ ]}\ \ \overset{j=3}{[2,3]}\ \ \overset{j=4}{[2,3]}\ \ \overset{j=5}{[1,2,3]}\ \ \overset{j=6}{[1,2,3]}\ \ \overset{j=7}{[1,2,3]}\ \ \overset{j=8}{[1,2,3]}\ \ \overset{j=9}{[\ 1\ ]}\ \ \overset{j=10}{[1,2,3]}\ \ \overset{j=11}{[1,2,3]}\ \ \overset{j=12}{[\ 1\ ]}$$

حال m های انتخابی بصورت زیر خواهند بود :

$m_{i1}=1 \Rightarrow m_1=1$ $\quad m_{i2}=1 \Rightarrow m_2=1$ $\quad m_{i3}=1 \Rightarrow m_3=2$

$m_{i4}=1 \Rightarrow m_4=2$ $\quad m_{i5}=1 \Rightarrow m_5=1$ $\quad m_{i6}=2 \Rightarrow m_6=2$

$m_{i7}=1 \Rightarrow m_7=1$ $\quad m_{i8}=2 \Rightarrow m_8=2$ $\quad m_{i9}=1 \Rightarrow m_9=1$

$m_{i10}=1 \Rightarrow m_{10}=1$ $\quad m_{i11}=1 \Rightarrow m_{11}=1$ $\quad m_{i12}=1 \Rightarrow m_{12}=1$

در نهایت مدهای انتخابی در لیست m بدینگونه است :

$$m=[1,\ 1,\ 2,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 1,\ 1,\ 1]$$

حال بر اساس زمان های انجام فعالیت هایی که در فایل j104-5.mm در سایت PSPLIB برای هر مد موجود است مدت زمان انجام هر فعالیت و زمان شروع و پایان و منابع مصرفی بصورت زیر خواهد بود. برای انتخاب زمان شروع یک فعالیت نیز بدینگونه عمل شده است که فعالیتی که از لیست L1 برای انجام گرفتن انتخاب می گردد تمامی پیش‌نیازهایش بررسی شده و ماکزیمم زمان‌اتمام پیش‌نیازهایش به عنوان زمان شروع فعالیت انتخابی به شرط در دسترس بودن منابع مورد نیاز در نظر گرفته می شود و در صورت در دسترس نبودن منابع، زمان شروع به زودترین زمانی که منابع در دسترس باشد شیفت پیدا می کند.

$$Duration=[0,\ 7,3,\ 8,\ 5,5,1,\ 6,\ 5,\ 2,3,\ 0]$$

$$ST=[0,\ \ 14\ ,21,\ 0,9,21,8,\ 21,\ 27,\ 27,\ 29,32]$$

$$FT=[0,\ \ 21\ ,24,\ 8,14,26,9,\ 27,32,\ 29,32,32]$$

وضعیت منابع مصرفی در زمان های مختلف نیز بصورت زیر خواهد بود :

$$R_1=[\ \overset{t_1 \ldots t_{21}}{0}\quad \overset{t_{22} \ldots t_{26}}{4}\quad \overset{t_{27}}{2}\quad \overset{t_{28},t_{29}}{3}\quad \overset{t_{30},\ldots,t_{32}}{2}\ ]$$

$$R_2=[\ \overset{t_1 \ldots t_8}{10}\quad \overset{t_9}{8}\quad \overset{t_{10} \ldots t_{21}}{10}\quad \overset{t_{22},,,t_{24}}{5}\quad \overset{t_{25},\ldots,t_{27}}{0}\quad \overset{t_{28},\ldots,t_{32}}{3}\ ]$$

$$Total\ \ N_1=[\ \overset{t_1 \ldots t_8}{0}\quad \overset{t_9 \ldots t_{14}}{5}\quad \overset{t_{15} \ldots t_{21}}{9}\quad \overset{t_{22},\ldots,t_{29}}{20}\quad \overset{t_{30},\ldots,t_{32}}{27}\ ]$$

$$Total\ \ N_2=[\ \overset{t_1 \ldots t_9}{9}\quad \overset{t_{10} \ldots t_{21}}{12}\quad \overset{t_{22} \ldots t_{27}}{17}\quad \overset{t_{28},\ldots,t_{32}}{30}\ ]$$

همانطور که پیش تر بیان شد سطح دسترسی منابع بصورت زیر بوده است که در منابع تجدید پذیر تجاوزی صورت نگرفته و منابع تجدید ناپذیر باقیمانده نیز بصورت زیر است :

$$R_1=4 \qquad R_2=11 \qquad N_1=50 \qquad N_2=61$$

$$\mathrm{Re}main\ N_1=23 \qquad \mathrm{Re}main\ N_2=31$$

پس از تولید جواب های اولیه و انتخاب بهترین جواب، نوبت به بازگشت زنبورها به کندو و رقص و جستجوی همسایگی آنهاست. لذا طبق موارد بیان شده در مقاله در جدول مقادیر $Pf_i$ و $d_i'$ که همان مدت زمان رقص هر زنبور است به صورت زیر بدست خواهد آمد :

$$\rightarrow \qquad Pf_{colony}=\frac{1}{n}\sum_{j=1}^{n}\frac{1}{C_{\max}^{i}} \qquad , \qquad Pf_i=\frac{1}{C_{\max}^{i}}\ d_i'=\frac{Pf_i}{Pf_{colony}}$$

جدول (۶) : مدت زمان رقص زنبورهای پیشتاز مثال نمونه

| $C_{\max}$ | **32** | **37** | **38** | **41** | **42** | **50** | **50** | **50** | **50** | **53** |
|---|---|---|---|---|---|---|---|---|---|---|
| $Pf_i$ | 0.031 | 0.027 | 0.0256 | 0.0227 | 0.025 | 0.02 | 0.0196 | 0.0175 | 0.0161 | 0.0161 |
| $d_i'$ | 1.4359 | 1.2759 | 1.2104 | 1.0729 | 0.9441 | 0.9441 | 0.9256 | 0.8382 | 0.7614 | 0.7614 |

حال با توجه به جدول ۴ (عدم رقصیدن زنبورهای و برگشت به کندو) احتمال بازگشت هر زنبور پیشتاز به کندو و رقصیدن آن بصورت زیر خواهد بود :

جدول (۷) : احتمال بازگشت زنبورهای پیشتاز به کندو در مثال نمونه

| Bee | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| $r_0$ | 0 | 0 | 0 | 0 | 0.2 | 0.2 | 0.2 | 0.6 | 0.6 | 0.6 |
| احتمال برگشت به کندو و رقص | 1 | 1 | 1 | 1 | 0.8 | 0.8 | 0.8 | 0.4 | 0.4 | 0.4 |

واضح است که زنبورهای پیشتاز ۱ تا ۴ که دارای بهترین جوابها نیز می‌باشند با احتمال ۱۰۰٪ به کندو بازگشته و زنبورهای تازه کار را برای جستجوی همسایگی در اطراف جواب خود بسیج می کنند سایر زنبورها نیز با احتمال های بیان شده ممکن است باز گردند و یا خود به جستجوی تصادفی در سایر مکانهای جواب بپردازند. تعداد زنبورهای تازه کار برای جستجوی همسایگی در جوابهای زنبورهای پیشتازی که به کندو بازگشته اند از فرمول زیر محاسبه می گردد :

$$n\ \ follower\ Bees = \max\ (\ [D_i.nDance0],\ 1\ )$$

مقدار $\boldsymbol{nDance}0$ برابر ۳ در نظر گرفته شده است . بر این اساس تعداد زنبورهای تازه کار تخصیص یافته برای هر زنبور پیشتاز در صورتی که به کندو بازگردد به صورت زیر خواهد بود:

جدول (۸) : تعداد زنبورهای تازه کار تخصیص یافته به هر زنبور پیشتاز برای جستجوی همسایگی

| شماره زنبور پیشتاز | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| تعداد زنبورهای تازه کار تخصیص یافته | 4 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 |

برای جستجوی همسایگی از روشهایی مانند Swap و Reverse و... استفاده شده است که بصورت زیر می باشند:

در روش Swap از جواب شدنی که از تکرار قبلی بدست آمده دو تا از فعالیت ها به تصادف انتخاب شده و جایشان عوض می گردد سپس این لیست همانند روشی که در مثال گفته شد به یک لیست شدنی تبدیل می گردد. به عنوان نمونه اگر در همان مثال نمونه دو فعالیت ۶ و۱۰ بصورت تصادفی انتخاب گردند و جایشان عوض گردد لیست زیر حاصل می شود که لیستی نشدنی است.

$$NL = [1,4,7,5,2,10\ ,8\ \ ,9\ ,3\ ,6\ ,11,12\ ]$$

لذا طبق روش مثال، فعالیت ها بترتیب وارد لیست می گردند و لیست، از فعالیتی که قابل انجام نباشد می‌گذرد و روند زیر را ادامه خواهد داد تا به لیست NL2 که لیست شدنی است تبدیل گردد. (فعالیت‌های ۱۰ و ۹ و ۱۲ در لیست NL قابل انجام نیست بنابراین به آخر لیست منتقل می گردند.)

$$NL = [1,4,7,5,2,(10),8\ \ ,(9)\ ,3\ ,6\ ,11,(12)\ ]$$
$$NL = [1,4,7,5,2,8,3,6,11,(10),(9),(12)]$$

بدین صورت لیست NL2 که لیستی شدنی است بصورت زیر می‌باشد:

$$NL_2 = [1,4,7,5,2,8,3,6,11,10,9,12]$$

همانطور که در بالا بیان شد لیست NL2 یک لیست شدنی خواهد بود لذا سایر موارد طبق قبل بر روی این جواب اجرا می گردد.

برای دستور Reverse نیز دو فعالیت به تصادف انتخاب می گردد و جای این دو فعالیت عوض می گردد و سپس تمامی فعالیتهای موجود بین این دو فعالیت بصورت معکوس چیده می شوند و بعد از آن، سایر موارد برای شدنی کردن جواب انجام می-گیرد. به عنوان نمونه اگر همان جواب اولیه مثال نمونه را به عنوان بهترین جواب اولین تکرار در نظر بگیرم روند جواب دوم در اثر عمل همسایگی توسط دستور Reverse بصورت زیر خواهد بود :

$$NL = [1,4,7,5,2,6,8,9,3,10,11,12]$$

اگر به تصادف دو عدد ۴ و ۸ انتخاب گردند جواب جدید بصورت زیر خواهد بود که نشدنی است.

$$NL = [1,8,6,2,5,7,4,9,3,10,11,12]$$

روند کار برای شدنی کردن جواب بصورت زیر خواهد بود:

$$NL = [1,(8),(6),2,(5),(7),4,(9),3,10,11,(12)]$$
$$NL = [1,2,4,3,10,11,(8),(6),5,7,(9),(12)]$$
$$NL_2 = [1,2,4,3,10,11,5,7,5,8,6,9,12]$$

و بدینگونه جواب جدید تولید شده و دوباره سایر گامهای الگوریتم به اجرا در خواهد آمد.

## ۳- نتایج و بحث

گستردگی حالت‌های مختلف این مسئله با فرضیات گوناگون که هریک در شرایط عملی خاص خود صادق است و الگوریتم‌ها و روش‌های حل مختلفی که در طی زمان برای حل این حالت‌های متفاوت طراحی شده است لزوم بوجود آمدن یک دسته از مسائل استاندارد برای این حالت‌های متفاوت و بوجود آوردن امکان مقایسه الگوریتم‌های مختلف را نمایان می کند. این کار برای اولین بار توسط دیوبس [۱] صورت گرفت. او ۸۳ مثال استاندارد از مسئله زمانبندی پروژه با منابع محدود ( RCPSP) ایجاد نمود. پاترسون [۲] تعداد ۱۱ مثال نمونه و تالبوت و پاترسون [۳] تعداد ۱۰ مثال دیگر ایجاد کردند. پاترسون [۴] یک مجموعه متشکل از مسائل بالا را به همراه ۶ مثال نمونه دیگر از دیویس گرد آوری نمود و مجموعه ای متشکل از ۱۱۰ مثال نمونه استاندارد بوجود آورد. همه مسائل فوق برای حالت RCPSP تک حالته [۵] طراحی شده اند. کار مفیدی که او انجام داد این بود که با استفاده از الگوریتمهای شناخته شده در زمان خود، جواب بهینه برای کلیه مسئله های مورد نظر را با معیار کمترین زمان اتمام پروژه بدست آورد و به مجموعه اضافه نمود. مسئله های فوق به مسائل پاترسون معروف شدند و تحقیقات زیادی نظیر دمولمستر و هرولنکولیش و درگزل و بسیاری از تحقیقات دیگر بکار گرفته شدند.

در حالی که بزرگترین مزیت دسته مسائل پاترسون وجود جواب بهینه برای این مسائل و الگوریتمهای دقیق و ابتکاری متعددی که روی آنها آزمایش شده اند می باشد، این مسائل از جهت تعریف پارامترها برای آزمایش الگوریتمها تحت شرایط متفاوت

[1]- Davis,E.W
[2]- Patterson , J.H, Huber,W.D
[3]- Talbor,F.B and Patterson,J.H
[4]- Patterson , J.H
[5]- Single mode Resource Constrained Project Scheduling (RCPSP)

دارای ضعف می باشند. در طرح مسائل استاندارد نکته مهم این است که بتوان پارامترهایی را در ایجاد مسائل فوق تعیین نمود که پیچیدگی مسئله را کنترل کند.

به همین دلیل مسائل استاندارد دیگری بوجود آمدند که از جمله به کار آلوارز و تاماریت[1] می‌توان اشاره کرد که در آن یک طرح فاکتوریل کامل برای پارامترهای در نظر گرفته شده برای تولید مسائل RCPSP بکار گرفته شد و شامل ۱۴۴ مسئله می باشد.در این مسائل ۵ نوع پارامتر بطور مناسب برای ایجاد یک طرح فاکتوریل کامل تغییر می کنند و در آن مسائل با ۲۵ و ۴۹ و ۱۰۱ فعالیت در نظر گرفته شده است.

جدول (۹): تنظیم پارامتر تعداد زنبورهای پیشتاز برای الگوریتم زنبورهای عسل در ۵۰۰۰ تکرار

| شماره مسئله | تعداد زنبورهای پیشتاز | Average CPU Time | Average Cmax | شماره مسئله | تعداد زنبورهای پیشتاز | Average CPU Time | Average Cmax |
|---|---|---|---|---|---|---|---|
| | 10 | 0.19 | 19.13 | | 10 | 0.22 | 26.86 |
| | 15 | 0.26 | 19.13 | | 15 | 0.33 | 26.86 |
| **J10** | 20 | 0.43 | 19.13 | **J18** | 20 | 0.66 | 26.86 |
| | 25 | 1.12 | 19.13 | | 25 | 1.98 | 26.86 |
| | 30 | 3.26 | 19.13 | | 30 | 4.21 | 26.86 |
| | 10 | 0.19 | 21.42 | | 10 | 0.39 | 26.86 |
| | 15 | 0.27 | 21.42 | | 15 | 0.46 | 31.45 |
| **J12** | 20 | 0.45 | 21.42 | **J20** | 20 | 0.87 | 29.86 |
| | 25 | 1.5 | 21.42 | | 25 | 2.49 | 29.86 |
| | 30 | 3.26 | 21.42 | | 30 | 4.88 | 29.79 |
| | 10 | 0.21 | 24.59 | | 10 | 0.77 | 43.89 |
| | 15 | 0.3 | 21.53 | | 15 | 0.91 | 43.89 |
| **J14** | 20 | 0.51 | 21.53 | **J30** | 20 | 0.169 | 41.65 |
| | 25 | 1.63 | 24.52 | | 25 | 2.98 | 41.65 |
| | 30 | 3.49 | 24.52 | | 30 | 6.83 | 41.65 |

بعد از آن کولیش برای مسئله RCPSP تک حالته به تعداد ۴۸۰ مسئله نمونه که دارای ۳۰ فعالیت بود ایجاد کرد و برای مسئله RCPSP چند حالته به تعداد ۶۴۰ مسئله نمونه ایجاد کرد که از این تعداد تنها ۵۳۶ عدد دارای جواب موجه بودند . کلیه مسائل دارای ۱۰ فعالیت بودند. نتایج کار آنها در تحقیقات زیادی نظیر بروکر و همکاران و بسیاری از تحقیقات دیگر بکار رفته است. کولیش و اسپرچر به مسائل قبلی، مسائل جدیدی از قبیلj12,j14,…,j30 و... افزودند و به تعداد ۱۰۰۰۰ عدد مسئله نمونه ایجاد کردند.

- **تنظیم پارامترهای الگوریتم زنبورهای عسل برای حل مسائل استاندارد MRCPSP**

همانطور که در بخش ۲-۱ بیان شد این الگوریتم به پارامترهای مختلفی احتیاج دارد. برای تنظیم این پارامتر الگوریتم زنبورهای عسل مسائل مختلف استاندارد مختلف، چندین بار توسط الگوریتم با مقادیر مختلف مورد حل قرار گرفته است تا پارامتر تعداد زنبور پیشتاز مناسب برای انجام بهترین عملکرد الگوریتم مذکور بدست آید. برای این منظور مسائل سری j10 ،j12 و...وj30 هر کدام ۵ بار با یک مجموعه تعداد زنبورهای پیشتاز بین ۱۰ تا ۳۰ زنبور بصورت [10,15…,30] مورد حل قرار گفته و دو معیار مقدار میانگین تابع هدف و میانگین زمان مصرف شده CPU برای انتخاب تعداد زنبورهای پیشتاز بهینه در نظر گرفته شده است که نتایج در جدول ۹ آورده شده است که بهترین مقدارها نیز برای این پارامتر با رنگ تیره تر نمایش داده شده است. با

[1]- Alvarez-Valds,R. and Tamarit, J.M

توجه به اینکه نتایج حل مسائل J30 بدلیل پیدا نشدن جواب‌های بهینه آن تاکنون، موجود نبوده، مقایسه آن با سایر الگوریتم‌ها موجود نیست و در جداول زیر آورده نشده است.
برای سایر پارامترها از قبیل تعداد محل‌های انتخابی و بهترین محل‌ها و... تنظیم پارامتر به صورت آنلاین (بهنگام سازی پویا) از طریق جدول ۴ و روش تصادفی استفاده شده است که در مثال قابل بررسی است. این الگوریتم با Matlab Ver 2010a برنامه نویسی شده و با رایانه‌ای با مشخصات CPU 2.1 GHZ و رم 2.1 GB اجرا گشته است.

- **نتایج عملکرد الگوریتم زنبورهای عسل در حل مسائل استاندارد PSPLIB مسئله MRCPSP**

در این بخش مسائل استاندارد MRCPSP مورد بررسی قرار گرفته است که همانطور که بیان شد در کتابخانه این دسته از مسائل مجموعه های j10 و j12 و j14 و j16 و j18و j20 و j30 موجود است. مسائلی با نامهای c15 و c21 و m1 و m2 و m4 و m5 و n0 و n1 و... نیز در محموعه مسائل استاندارد MRCPSP نیز به چشم می خورند که تفاوتهایی از نظر تعداد منابع و فعالیت ها دارند . به عنوان مثال مسائل دسته C دارای ۱۸ فعالیت و ۲ منبع تجدید پذیر تجدید ناپذیر می باشند و مسائل دسته m نیز دارای ۱۸ فعالیت و ۲ منبع تجدید پذیر تجدید ناپذیر می باشند مسائل دسته n0 دارای ۱۴ فعالیت و تنها دو منبع تجدید پذیر و n1 دارای ۱۸ فعالیت و ۱ منبع تجدید ناپذیر و ۲ منبع تجدید پذیر و n3 نیز دارای ۳ منبع تجدید ناپذیر و ۲ منبع تجدید پذیر است.
در این مقاله تنها از دسته مسائل سری j برای مقایسه کمک گرفته شده است که شامل مسائل j10 و j12 و...وj20 می باشد. استفاده از سایر مسائل استاندارد MRCPSP می تواند از جمله توسعه های آتی این مقاله در نظر گرفته شود.
در ابتدا مسئله j10 مورد بررسی قرار گرفته است این مسائل دارای دو منبع تجدید پذیر و دو منبع تجدید ناپذیر هستند. از ۶۴۰ مسئله نمونه در این دسته تنها ۵۳۶ عدد از آنها دارای جواب شدنی هستند. نتایج بر حسب ۵۰۰۰ تکرار و بر اساس میانگین انحراف جواب از حد پایین مرتب شده‌اند. الگوریتم زنبورهای عسل تنها ۰.۰۵ درصد انحراف از حد پایین داشته است. برای تمامی مسائل MRCPSP سری j به جز j30 جوابهای بهینه بدست آمده است لذا حد پایین همین جواب‌های بهینه در نظر گرفته شده است.

جدول (۱۰) : مقایسه نتایج عملکرد الگوریتم رنبور عسل با سایر الگوریتمها برای مسائل j10 چندحالته

| مستله J10 الگوریتم | منبع | تعداد جوابهای شدنی | متوسط انحراف از میانگین در ۵۰۰۰ تکرار | درصد یافتن جوابهای بهینه |
|---|---|---|---|---|
| Bees algorithm | This paper | 536 | 0.05 | 98.5 |
| Differential Evolution | Damak et al. | 536 | 0.06 | 99.25 |
| Hybrid Genetic Algorithm | Lova et al. | 536 | 0.06 | 98.51 |
| PSO | Jarboui et al. | 536 | 0.09 | 99.30 |
| Scatter search | Ranjbar et al. | 536 | 0.21 | -- |
| SA | Bouleimen et al | 536 | 0.18 | --- |
| Genetic Algorithm | Jozefowska et al. | 536 | 1.16 | 96.50 |

همانطور که از نتایج بالا مشخص است الگوریتم زنبور عسل در حل مسئله j10 در مسائل MRCPSP رتبه اول را به خود اختصاص داده است و تنها ۰.۰۵ درصد انحراف از جوابهای بهینه در ۵۰۰۰ تکرار دارد الگوریتم بعدی که در رتبه دوم قرار دارد، الگوریتم ژنتیک هیبریدی است که توسط آنتونیو وهمکاران در سال ۲۰۰۹ توسعه یافته است که ۰.۰۶ انحراف داشته است.

الگوریتم جستجوی پرندگان که توسط جاربی و همکاران[1] در سال ۲۰۰۸ میلادی ایجاد شده است رتبه سوم را داراست که ۰.۰۹ درصد انحراف داشته است. الگوریتم بعدی نیز الگوریتم تکامل تفاضلی (DE) است که در سال ۲۰۰۹ توسط داماک و همکاران[2] برای حل این مسئله مورد استفاده قرار گرفته است . الگوریتم جستجوی پراکندگی نیز توسط رنجبر و همکاران در سال ۲۰۰۹ مورد استفاده قرار گرفته است که مقدار ۰.۲۱ درصد از جوابهای بهینه مسائل j10 انحراف داشته است.

دسته دیگر مسائل RCPSP چند حالته (مد)، مسائل j12 است که دارای ۱۴ فعالیت (۲ فعالیت موهمی) و دو منبع تجدید پذیر و دو منبع تجدید ناپذیر است. در مورد این مسئله الگوریتم زنبورهای عسل در رتبه پنجم نسبت به سایر الگوریتم‌های که در بخش قبل توضیح داده شد، قرار دارد. متوسط انحراف از میانگین جوابهای بهینه در ۵۰۰۰ تکرار در این مسئله ۰.۷۸ درصد بوده است. البته در دبیات این مسائل گاه از ۶۰۰۰ تکرار نیز برای مقایسه استفاده شده است

جدول (۱۱) : مقایسه نتایج عملکرد الگوریتم رنبور عسل با سایر الگوریتمها برای مسائل j12 چندحالته

| مستله J12 الگوریتم | منبع | تعداد جوابهای شدنی | متوسط انحراف از میانگین در ۵۰۰۰ تکرار | درصد یافتن جوابهای بهینه |
|---|---|---|---|---|
| Differential Evolution | Damak et al. | 547 | 0.11 | 99.30 |
| PSO | Jarboui et al. | 547 | 0.14 | 98.47 |
| SA | Bouleimen et al | 547 | 0.19 | 91.20 |
| Scatter search | Ranjbar et al. | 547 | 0.65 | -- |
| Bees algorithm | This paper | 547 | 0.78 | 88.07 |
| Genetic Algorithm | Jozefowska et al. | 547 | 1.73 | -- |

در جداول زیر، مسائل j14 و j20 قرار گرفته است که الگوریتم زنبورهای عسل در مورد این دو مسئله نیز در ۵۰۰۰ تکرار رتبه پنجم را به خود اختصاص داده است. در مورد مسائل j14 تنها ۰/۸۱ درصد انحراف از جوابهای بهینه و در مورد مسئله j20 که مسئله بزرگتری نسبت به سایر مسائل است مقدار ۳/۲۲درصد انحراف از جوابهای بهینه بدست آمده است.

جدول(۱۲) : مقایسه نتایج عملکرد الگوریتم زنبور عسل با سایر الگوریتمها برای مسائل j14 چندحالته

| مستله J14 الگوریتم | منبع | تعداد جوابهای شدنی | متوسط انحراف از میانگین در ۵۰۰۰ تکرار | درصد یافتن جوابهای بهینه |
|---|---|---|---|---|
| Hybrid Genetic Algorithm | Lova et al. | 551 | 0.14 | 92.92 |
| Scatter search | Ranjbar et al. | 551 | 0.24 | -- |
| Differential Evolution | Damak et al. | 551 | 0.34 | 97.60 |
| PSO | Jarboui et al. | 551 | 0.44 | 91.11 |
| Bees algorithm | This paper | 551 | 0.81 | 87.65 |
| SA | Bouleimen et al | 551 | 0.92 | 82.60 |
| Genetic Algorithm | Jozefowska et al. | 551 | 1.73 | -- |

[1]- Jarboui
[2]- Damak

جدول(۱۳): مقایسه نتایج عملکرد الگوریتم زنبور عسل با سایر الگوریتمها برای مسائل j20 چندحالت

| مسئله J20 / الگوریتم | منبع | تعداد جوابهای شدنی | متوسط انحراف از میانگین در ۵۰۰۰ تکرار | درصد یافتن جوابهای بهینه |
|---|---|---|---|---|
| Differential Evolution | Damak et al. | 554 | 0.70 | 91.75 |
| Hybrid Genetic Algorithm | Lova et al. | 554 | 0.87 | 80.32 |
| PSO | Jarboui et al. | 554 | 1.21 | 74.19 |
| Scatter search | Ranjbar et al. | 554 | 1.64 | -- |
| SA | Bouleimen et al. | 554 | 2.10 | 66.90 |
| Bees algorithm | This paper | 554 | 3.22 | 60.21 |
| Genetic Algorithm | Jozefowska et al. | 554 | 6.74 | -- |

- **نتیجه گیری**

در این مقاله ، الگوریتم فراابتکاری زنبورهای عسل برای حل مساله زمانبندی پروژه با محدودیت منابع چندحالته (MRCPSP) توسعه یافت که نتایج حاکی از عملکرد خوب این الگوریتم فراابتکاری در حل این سری از مسائل بود. در توسعه های آتی می توان از این الگوریتم برای حل مسائل MRCPSP در حالتی که انقطاع فعالیت ها نیز مجاز باشد استفاده کرد همچنین می توان به مسائلی از قبیل زمان بندی پروژه چندهدفه با محدودیت منابع و مسائل RCPSP چند پروژه ای اشاره کرد.

**منابع :**


1- Alcaraz ,J. Maroto, C. Ruiz, R. 2003, Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms, Journal of the Operational Research Society ,54 : 614–626.
2- Barrios, A. Ballestin, F. Valls, V. 2009, A double genetic algorithm for the MRCPSP/max, Computers and Operations Research, doi:10.1016/j.cor.
3- Bouleimen, K. Lecocq, H. 2003, A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version, European Journal of Operational Research ,149 ,2 : 268–281.
4- Brucker, P. Drexl, A. Mohring, R. Neumann, K. Pesch, E. 1999, Resource-constrained project scheduling: Notation, classification, models, and methods, European Journal of Operational Research 112, :3–41.
5- Brucker, P. Knust, S. 2001, Resource-constrained project scheduling and timetabling, Lecture Notes in Computer Science ,2079 : 277–293
6- Calhoun, K.M. Deckro, R.F. Moore, J.T. Chrissis, J.W. Hove, J.C.V. 2002, Planning and re-planning in project and production scheduling, Omega – The international Journal of Management Science ,30 ,3:155–170.
7- Damak, N. Jarboui, B. Siarry, P. Loukil, T. 2009 **,** Differential Evolution for solving multi-mode resource-constrained project scheduling problems, Computers & Operations Research,Volume 36, Issue 9, September: 2653-2659.
8- Demeulemeester,E L. Herroelen,W S.2002,Project Scheduling, Kluwer Academic Publishers: 500-526

9- Hartmann, S. 2001, Project scheduling with multiple modes: A genetic algorithm, Annals of Operations Research, 102: 111–135.
10- Jarboui, B. Damak, N. Siarry, P. Rebai, A. 2008, A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems, Applied Mathematics and Computation ,195 : 299–308.
11- Jozefowska, J. Mika, M. Rozycki, R. Waligora, G. We_glarz, J. 2001, Simulated annealing for multi-mode resource-constrained project scheduling, Annals of Operations Research , 102: 137–155.
12- Kolisch, R. Drexl, A. 1997, Local search for non-preemptive multi-mode resource constrained project scheduling, IIE Transactions, 29 : 987–999.
13- Lova, A. , Tormos, P. Cervantes, M. Barber F. 2009, An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes, Int. J. Production Economics ,117 :302–316.
14- Nudtasomboon, N. Randhawa, S.U. 1997, Resource-constrained project scheduling with renewable and non-renewable resources and time-resource tradeoffs, Computers and Industrial Engineering. 32,1 :227–242.
15- Ozdamar, L. 1999, A genetic algorithm approach to a general category project scheduling problem, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews ,29 : 44–59.
16- Pesch, E. 1999, Lower bounds in different problem classes of project schedules with resource constraints, in,We_glarz ,193: 53–76.
17- Pham, D.T. Ghanbarzadeh, A. Koç, E. Otri, S. Rahim, S. and Zaidi, M. ,2006, The Bees Algorithm – A Novel Tool for Complex Optimisation Problems, Proceedings of IPROMS, Conference:454-461.
18- Pham, D.T., Afify, Ashraf. 2007, Ebubekir Koc Manufacturing cell formation using the Bees Algorithm, IPROMS Innovative Production Machines and Systems Virtual Conference, Cardiff.
19- Ranjbar, M. de Reyck, B. Kianfar, F. 2009, A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling, European Journal of Operational Research, 193 :35–48.
20- Sabzehparvar, M. Seyed-Hosseini, S.M. 2008, A mathematical model for the multimode resource-constrained project scheduling problem with mode dependent time lags, Journal of Supercomputing , 44 , 3 : 257–273.
21- Sadeghi, A. Safi Samghabadi, A. Barzinpour, F. 2011, Solving Resource Constrained Project Scheduling with Bees Algorithm, 6th International Project Management Conference: 159-172.
22- Talbot, F.B. 1982, Resource-Constrained Project Scheduling Problem with Time-Resource Trade-Offs, The Nonpreemptive Case, Management Science, 28: 1197-1210.
23- Varma, V.A. Uzsoy, R. Pekny, J. Blau, G. 2007, Lagrangian heuristics for scheduling new product development projects in the pharmaceutical industry, Journal of Heuristics ,13 ,5: 403–433.
24- Wong, Li-Pei, Yoke, Malcolm Hean Low Chin Soon Chong ,2008, A Bee Colony Optimization Algorithm for Traveling Salesman Problem , 2008 Second Asia International Conference on Modelling Simulation : 818-823.
25- Yang, X.S. 2005. Engineering Optimizations Via Nature-Inspired Virtual Bee Algorithms, Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, LECTURE NOTES IN COMPUTER SCIENCE 3562: 317-323.

26- Zhu, G. Bard, J.F. Yu, G. 2008, A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem, INFORMS Journal on Computing, 18: 377–390.