

# یک روش ترکیبی مبتنی بر الگوریتم دلیوج و ژنتیک برای حل

## مسئله فروشنده دوره گرد

سجاد قطعی<sup>۱</sup>، محمدرضا میبیدی<sup>۲</sup>، حمید پروین<sup>۳</sup>

<sup>۱</sup>دانشکده برق، رایانه و فناوری اطلاعات، دانشگاه آزاد اسلامی قزوین

s.ghatei@qiau.ac.ir

<sup>۲</sup>دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر تهران

mmeybodi@aut.ac.ir

<sup>۳</sup>دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه علم و صنعت تهران

parvin@iust.ac.ir

چکیده - این مقاله به اثر ترکیب روش‌های بهینه‌سازی الگوریتم ژنتیک (GA) با الگوریتم دلیوج می‌پردازد. اگر چه الگوریتم‌هایی شبیه به الگوریتم ژنتیک و الگوریتم دلیوج هر یک به تنهایی قادر هستند که مسائل بهینه‌سازی را حل کنند، اما هر کدام قابلیت‌ها و ضعف‌های خاص خود را دارند. در این مقاله سعی شده است تا از مسئله واقعی فروشنده دوره گرد برای آزمون استفاده شود. نتایج الگوریتم بر دو مجموعه داده واقعی ۱۰۱ شهر و ۴۴۲ شهر حاکی از بهتر بودن نتایج ترکیبی این دو الگوریتم نسبت به هر کدام به صورت جداگانه می‌باشد.

کلید واژه - مسئله فروشنده دوره گرد، الگوریتم دلیوج اصلی، الگوریتم ژنتیک، الگوریتم ترکیبی GDA-GA

### ۱- مقدمه

شبیه‌ساز حرارتی می‌باشد الگوریتم دلیوج<sup>۱</sup> (GDA) است که در سال ۱۹۹۳ توسط دوئک<sup>۲</sup> معرفی شده است. کار این الگوریتم شبیه به رفتار یک انسانی می‌باشد که در دام سیل افتاده است، به طوری که از پایین دست‌ها دوری کرده و با افزایش سطح آب به نقاط مرتفع صعود می‌کند. در این الگوریتم نیز با تعریف یک پارامتر با عنوان "سطح آب" محدوده‌ای را برای جواب‌های بدست آمده تعیین می‌کنیم و هر بار جواب به دست آمده را با این پارامتر مورد مقایسه قرار می‌دهیم. در مسائل بهینه‌سازی در هر مرحله مقدار این پارامتر افزایش می‌یابد.

روند دیگری در روش‌های مکاشفه‌ای الگوریتم‌های تکاملی می‌باشند. الگوریتم‌های تکاملی روش‌های جستجوی تصادفی همه منظوره می‌باشند که انتخاب و تکامل طبیعی در جهان

هدف از بهینه‌سازی جستجوی مقادیری به ازای مجموعه‌ای از پارامترها است که تابع هدف مربوط به یک محدودیت مشخص را کمینه یا بیشینه کند. بهینه‌سازی در زمینه‌های فراوانی نظیر تجارت، صنعت، مهندسی و علوم کامپیوتر در سطح وسیعی کاربرد دارد [1]. هر یک از روش‌های مختلف بهینه‌سازی متناسب با یک دسته از مسائل می‌باشند. مشکل اصلی، تصمیم برای انتخاب بهترین روش بهینه‌سازی (و همچنین تنظیم پارامترهای مدل انتخابی به صورت بهینه) تقریباً غیرممکن است. به ازای تمامی مسائل بهینه‌سازی، هیچ یک از روش‌ها نسبت به سایرین از برتری مطلق برخوردار نیستند. اگرچه روش‌های متداولی موجود می‌باشند که برای مسائلی در محدوده‌های مشخص مناسب‌تر می‌باشند.

یکی از روش‌های مکاشفه‌ای که بسیار شبیه به روش

<sup>۱</sup> Great Deluge Algorithm  
<sup>۲</sup> Dueck

## ۲- الگوریتم دلیوج

الگوریتم دلیوج یک روش جامع برای حل مسائل بهینه-سازی می‌باشد که اولین بار توسط دوئک در سال ۱۹۹۳ مطرح شد [2] و ایده اصلی آن بسیار شبیه به الگوریتم شبیه‌ساز حرارتی می‌باشد [3]. در [2] این الگوریتم بخوبی برای حل مسئله فروشنده دوره‌گرد ۴۴۲ و ۵۳۲ شهر بکار برده شده است. بورک در [3,4] از این الگوریتم برای حل مسئله زمانبندی امتحانات استفاده کرده است. در [5] ناهاس و همکارانش این الگوریتم را برای حل مسئله تخصیص بافر در خطوط تولید غیرقابل اعتماد بکار برده‌اند. در [6] از روش ترکیبی دلیوج با دسته مورچگان برای حل مسئله تخصیص افزونگی استفاده شده است. در [7] با بهبود الگوریتم دلیوج میزان کارایی آن بر روی مسئله لایه‌بندی پویا سنجیده شده است. خطاب و دیگر نویسندگان در [8] از این الگوریتم برای بهینه‌سازی کارایی سیستم‌های سری-موازی سود برده‌اند. در [9] نوعی الگوریتم دلیوج اصلاح شده غیر خطی برای حل مسئله زمانبندی امتحانات ارائه شده است

در حالت کلی، این الگوریتم با یک مقدار تخمینی کار خود را شروع می‌کند و این به عنوان جواب اولیه مساله در نظر گرفته می‌شود. در مراحل بعد، الگوریتم با تولید جواب‌هایی دیگر در محدوده جواب اولیه کار خود را ادامه می‌دهد و نتیجه هر بار با یک مقدار از پیش تعیین شده که سطح آب<sup>۳</sup> (WL) خوانده می‌شود مورد مقایسه قرار می‌گیرد. اگر جواب تولیدی بزرگتر از WL بود مورد پذیرش قرار گرفته و در غیر اینصورت رد می‌شود. مقدار WL هر بار به یک مقدار مشخص (UP) افزایش می‌یابد [2] و [3]. شبه‌کد الگوریتم دلیوج در شکل ۱ مشاهده می‌شود.

```
Choose an initial configuration Old_Config
Choose WL0 and Up
For n=0 to # of iterations
    Generate a small stochastic perturbation
    New_Config of the solution
    If Fitness (New_Config) > WL
        Old_Config := New_Config
    End If
    WL = WL + Up
End For
```

شکل ۱: شبه کد الگوریتم دلیوج

برای عملگر perturbation در الگوریتم دلیوج از عملگر

زیست‌شناسی را شبیه‌سازی می‌کنند. این روش‌ها برخلاف روش‌های پیشین، بهینه‌سازی را بر اساس جمعیتی از راه‌حل‌های بالقوه (راه‌حل‌های کاندید) انجام می‌دهند.

چنانکه واضح است روش‌های بهینه‌سازی گوناگون وابسته به مسئله، کارایی‌های متفاوت دارند. از آنجایی که هر کدام از این روش‌ها مزایا و معایب خاصی دارند، با ترکیب آنها می‌توان از مزیت‌های مشترکشان بهره برد، به طوری که عیب‌های همدیگر را نیز بپوشانند. به عنوان مثال ترکیب الگوریتم دلیوج و الگوریتم‌های ژنتیک می‌تواند از ویژگی کاوش بالای الگوریتم ژنتیک و سرعت همگرایی بالای الگوریتم دلیوج بهره بگیرد. این الگوریتم‌های ترکیبی در قبال برخی مسائل کارایی بالاتری را نسبت به تک‌تک الگوریتم‌های بکار رفته در آن از خود نشان می‌دهند.

مسئله فروشنده دوره‌گرد یکی از مسائل بسیار مشکلی می‌باشد که به عنوان یک چالش برای روش‌های هوشمند مطرح بوده است. این مساله شامل مختصات چندین شهر در صفحه اقلیدسی می‌باشد که قرار است یک تور بسته با طول مینیمم که همه شهرها را به هم وصل می‌کند و تشکیل یک چندضلعی می‌دهد را بیابیم.

در این مقاله یک مدل جدید بنام GA - GDA پیشنهاد می‌گردد. در مدل پیشنهادی از هر دو خاصیت محلی و عمومی بودن الگوریتم‌ها به شکل خوب استفاده شده است. یعنی در ابتدا الگوریتم ژنتیک با گام‌های بزرگ به جستجو پرداخته و بعد از به دست آوردن یک سری نتایج اولیه نزدیک به بهینه، بهترین نتایج به عنوان ورودی به الگوریتم دلیوج سپرده می‌شود و در این مرحله این الگوریتم با گام‌های بسیار کوچک به جستجوی فضا می‌پردازد. کارایی این روش بر روی مسئله چالش برانگیز فروشنده دوره‌گرد آزمایش می‌شود.

این مقاله به صورت زیر بخش‌بندی شده است: بخش دوم مختصراً به معرفی الگوریتم دلیوج پرداخته است، در بخش سوم الگوریتم ژنتیک توضیح داده شده است، در بخش چهارم مدل پیشنهادی شرح داده می‌شود، بخش پنجم مسئله مورد ارزیابی را تشریح می‌کند. در بخش بعدی ابتدا عملگر جهش مناسب برای حل مسئله فروشنده دوره‌گرد به نام Lin2Opt توضیح داده می‌شود. در ادامه این بخش عملگر تزویج مناسب برای حل مسئله فروشنده دوره‌گرد توضیح داده می‌شود. بخش بعدی شامل آزمایشات و نتایج می‌باشد. بخش پایانی به نتیجه‌گیری می‌پردازد.

Lin2Opt استفاده شده است که در بخش ۵ بحث خواهد شد.

برای عملگر mutation در الگوریتم ژنتیک از عملگر Lin2Opt استفاده شده است که در بخش ۵ بحث خواهد شد.

### ۳- الگوریتم ژنتیک

الگوریتم ژنتیک، متداول ترین الگوریتم در بین الگوریتم-های تکاملی، همانطور که از اسمش پیداست، بر پایه نظریه تکاملی چارلز داروین یعنی بقای اصلح بنا نهاده شده است که می گوید: افرادی که به طور مؤثری برای به دست آوردن منابع رقابت می کنند، دارای شانس فزاینده ای برای تکثیر می باشند و بر اساس اصول انتخاب و تنوع مبتنی بر تصادف عمل می کنند. در ابتدا آلن تورینگ در سال ۱۹۴۸ برای اولین بار مفهوم جستجوی تکاملی (ژنتیکی) را ارائه کرد. ایده اصلی الگوریتم-های تکاملی در سال ۱۹۶۰ توسط ریچنبرگ مطرح گردید. در سال ۱۹۶۲ برمرمن با استفاده از کامپیوتر یک مساله بهینه سازی را با تکیه بر تکامل و باز ترکیبی انجام داد. اما مفهوم آن به شکل کنونی اولین بار توسط هالند ارائه شد [10].

انگیزه اصلی پیدایش الگوریتم های تکاملی مانند سایر الگوریتم های موفق دیگر، از طبیعت می باشد. همواره طبیعت به عنوان منبع الهام برای مهندسی و دانشمندان بکار رفته است که می توان ادعا نمود تمام یافته های بشر برگرفته از طبیعت می باشد. الگوریتم های تکاملی هم از این قاعده مستثنا نمی باشند. همانطور که ذکر شد، یک الگوریتم تکاملی از یک محیط شامل جمعیتی از افراد با منابع محدود تشکیل می شود، رقابت برای این منابع باعث انتخاب افرادی می شود که بهتر از بقیه با محیط تطبیق یافته اند و توانسته اند منابع بیشتری جهت ادامه حیات بدست آورند. این افراد به عنوان والد به منظور ایجاد افراد جدید از طریق آمیزش و جهش عمل می کنند. برازندگی افراد جدید ارزیابی می شود و سپس این افراد جدید به منظور بقاء به رقابت می پردازند. در طول زمان، انتخاب طبیعی باعث افزایش برازندگی متوسط جمعیت می شود. شبه کد الگوریتم ژنتیک در شکل ۲ مشاهده می شود.

### ۴- مدل پیشنهادی GDA-GA

در این مدل جدید از ترکیب الگوریتم دلیوج و الگوریتم ژنتیک استفاده شده است. هدف از پیشنهاد این مدل این بوده که با ترکیب خاصیت محلی بودن الگوریتم دلیوج و خاصیت سراسری الگوریتم بهینه سازی ژنتیک به یک موازنه در جستجو برسیم. از هر دو خاصیت محلی و عمومی بودن الگوریتم ها به شکل خوب استفاده شده است. یعنی در ابتدا الگوریتم ژنتیک با گام های بزرگ به جستجو پرداخته و بعد از بدست آوردن نتایجی نزدیک به بهینه، از آنها بعنوان ورودی الگوریتم دلیوج استفاده می شود تا در این مرحله این الگوریتم با گام های بسیار کوچک در همسایگی جوابهای تولید شده قبلی به جستجوی دقیق تر فضا بپردازد. شبه کد الگوریتم GDA-GA در شکل ۳ آمده است.

```
Choose the initial population of individuals
Evaluate the fitness of each individual in that population
Repeat on this generation until termination
  Select the best-fit individuals for reproduction
  Breed new individuals through crossover and mutation
  operations to give birth to offspring
  Evaluate the individual fitness of new individuals
  Replace least-fit population with new individuals
End Repeat
Choose the best individual as Old_Config
Choose WL0 and Up
For n=0 to # of iterations
  Generate a small stochastic perturbation New_Config
  of the solution
  If Fitness (New_Config) > WL
    Old_Config := New_Config
  End If
  WL = WL + Up
End For
```

شکل ۳: شبه کد الگوریتم GDA-GA

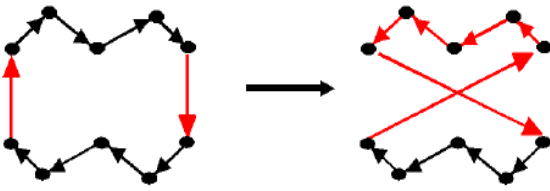
### ۵- مسئله فروشنده دوره گرد

مسئله فروشنده دوره گرد یکی از مسائل بسیار مشکلی می باشد که به عنوان یک چالش برای روش های هوشمند مطرح بوده است. این مساله شامل مختصات چندین شهر در صفحه اقلیدسی می باشد که قرار است یک تور بسته با طول مینیمم را بدهد. در این تور بسته همه شهرها به هم وصل شده و تشکیل یک چندضلعی را می دهد.

اگر C دنباله شهرها باشد، فرض کنید Length(C) طول

```
Choose the initial population of individuals
Evaluate the fitness of each individual in that population
Repeat on this generation until termination
  Select the best-fit individuals for reproduction
  Breed new individuals through crossover and mutation
  operations to give birth to offspring
  Evaluate the individual fitness of new individuals
  Replace least-fit population with new individuals
End Repeat
```

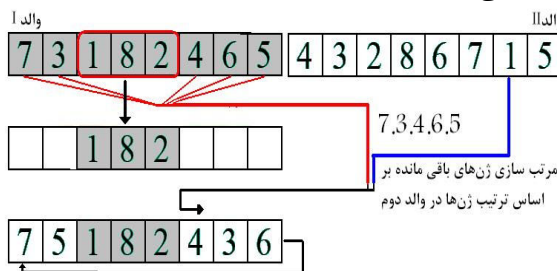
شکل ۲: شبه کد الگوریتم ژنتیک



شکل ۵: مثالی از الگوریتم Lin2Opt با ۱۰ شهر

عملگر تزویج در ژنتیک نیز با عملگرهای تزویج معمول متفاوت خواهد بود. در اینجا عملگر تقطیع دو نقطه‌ای مخصوص نمایش ترتیبی استفاده شده است.

در این عملگر همانند بازترکیبی دونقطه‌ای معمولی یک قطعه از کروموزوم‌های والدین را جدا کرده و در کروموزوم فرزندان در مکان متناظرشان کپی می‌کنیم. برای هرکدام از فرزندان، ژن‌هایی که از کروموزوم والد مربوطه کپی نشده‌اند را براساس ترتیبی که در کروموزوم والد دیگر وجود دارد به کروموزوم منتقل می‌کنیم. شکل ۶ نحوه اجرای عملگر فوق را نشان می‌دهد.



شکل ۶: مثالی از عملگر تزویج استفاده شده در ژنتیک الگوریتم‌های مورد استفاده برای نمایش ترتیبی

## ۷- نتایج شبیه سازیها

آزمایشات بر روی ۲ مجموعه داده استاندارد برای مساله فروشنده دوره گرد انجام پذیرفته است. این مسائل دارای ۱۰۱ و ۴۴۲ شهر هستند. مقدار تور کمینه در این مسائل به ترتیب ۶۴۰.۲۱۲ و ۵۰۷۸۳.۵ واحد می‌باشد. لازم بذکر است که نتایج بدست آمده، میانگین ۳۰ بار اجرای مستقل با ۱۰۰۰۰ بار تکرار می‌باشد. جدول ۱ و شکل‌های ۷ و ۸ شامل نتایج الگوریتم پیشنهادی و سایر روش‌هاست.

مسیر پیموده شده بین شهرها به ترتیب  $C$  باشد. یک تور می‌تواند یک تبدیل مانند  $C \rightarrow C\pi$  باشد. این تبدیل، جایگشتی مانند  $\pi$  را می‌گیرد و تور متناظر از  $\pi(1)$  شروع شده به  $\pi(2), \pi(3), \dots, \pi(N)$  می‌رود و نهایتاً در  $\pi(1)$  خاتمه می‌یابد که  $N$  تعداد شهرهاست. به طوری که برای هر جایگشت دیگر  $\pi'$ ،  $\text{Length}(C\pi') \geq \text{Length}(C\pi)$  می‌باشد.

دو مجموعه داده استاندارد برای مسئله فروشنده دوره گرد، مسائل ۴۴۲ و ۱۰۱ شهر گروتشل هستند. در این مجموعه داده‌ها، مختصات ۴۴۲ و ۱۰۱ شهر در صفحه اقلیدسی داده شده است. یک تور بسته با طول مینیمم که همه شهرها را به هم وصل می‌کند و تشکیل یک چندضلعی را می‌دهد خواسته شده است.

## ۶- نمایش راه حل و عملگرهای تزویج و جهش

در مسئله فروشنده دوره گرد، لزوماً از بازنمایی ترتیبی برای راه حل‌های کاندید باید استفاده کرد. از آنجایی که در نمایش ترتیبی این مسئله نمی‌توان مقدار ژن‌ها را تغییر داد بلکه فقط می‌توان محل ژن یا ترتیب ژن‌های یک کروموزوم را تغییر داد، رایج‌ترین نوع جهش، جابجا کردن دو ژن در کروموزوم می‌باشد و همچنین در نمایش ترتیبی می‌توان جهش‌های دیگری که ترتیب کروموزوم را تغییر می‌دهند استفاده کرد. البته در اینجا از عملگر جهش مخصوصی به نام Lin2Opt استفاده می‌شود که در ادامه توضیح داده خواهد شد. عملگر مولد تغییرات، perturbation، در الگوریتم دلیج نیز همین عملگر Lin2Opt در نظر گرفته شده است.

از Lin2Opt بعنوان یک روال برای ایجاد تور جایگشت جدید از تور قدیمی استفاده می‌کنیم. الگوریتم Lin2Opt در شکل ۴ آورده شده است.

Input C  
 Choose  $i, j \in C$ , where  $i < j$   
 Disconnect  $\pi(i)$  to  $\pi(i+1)$  and  $\pi(j)$  to  $\pi(j+1)$   
 Connect  $\pi(i)$  to  $\pi(j)$  and  $\pi(i+1)$  to  $\pi(j+1)$

شکل ۴: شبه کد الگوریتم Lin2Opt

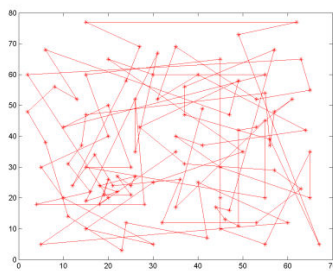
برای فهم بهتر به مثالی از اعمال الگوریتم بر روی یک تور فرضی در شکل ۵ توجه کنید.

## ۸- نتیجه گیری

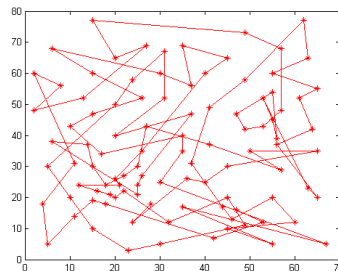
در این مقاله یک مدل ترکیبی جدید برای بهینه سازی ارائه شده است. هدف این مدل این بوده که با ترکیب خاصیت محلی بودن الگوریتم دلیوج و خاصیت سراسری الگوریتم بهینه سازی ژنتیک به یک موازنه در جستجو برسیم. از هر دو خاصیت محلی و عمومی بودن الگوریتم ها به شکل خوب استفاده شده است. یعنی در ابتدا الگوریتم ژنتیک با گام های بزرگ به جستجو پرداخته و بعد از بدست آوردن نتایج نزدیک به بهینه، آنها به عنوان ورودی، به الگوریتم دلیوج سپرده می شود تا در این مرحله این الگوریتم با گام های بسیار کوچک به جستجوی دقیق تر فضا بپردازد. این الگوریتم ها بر روی ۲ مجموعه داده از سری مسائل فروشنده دوره گرد آزمایش شده و نتایج شبیه سازی ها نشان داد که مدل ارائه شده جواب های بهتری در مقایسه با مدل اولیه ژنتیک و دلیوج استاندارد تولید می نماید.

نام الگوریتم	شهر ۱۰۱	شهر ۴۴۲
GDA	668.4631	77043
GA	694.1735	93365
GDA-GA	647.2123	61773

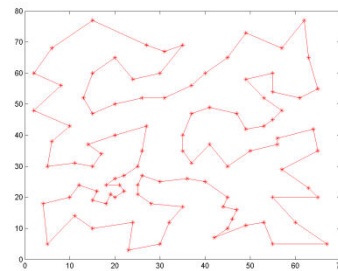
با مقایسه سطر مربوط به الگوریتم ترکیبی ژنتیک و دلیوج نسبت به الگوریتم های دیگر مشاهده می شود که به طور معمول در اکثر موارد الگوریتم پیشنهادی جواب قابل قبول به دست می دهد و این نتیجه به طور معمول دارای پایداری لازم می باشد.



(الف): پیکربندی اولیه

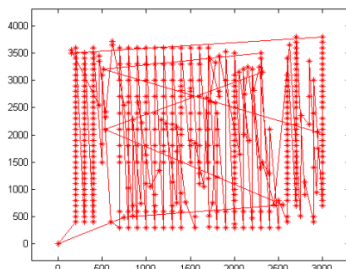


(ب): بعد از ۱۰۰۰ بار تکرار

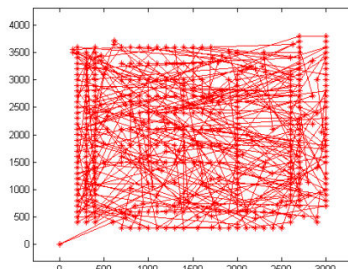


(ج): پیکربندی نهایی

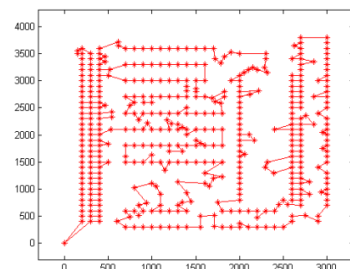
شکل ۷: نتیجه اجرای الگوریتم پیشنهادی بر روی مسئله فروشنده دوره گرد ۱۰۱ شهر



(الف): پیکربندی اولیه



(ب): بعد از ۱۰۰۰ بار تکرار



(ج): پیکربندی نهایی

شکل ۸: نتیجه اجرای الگوریتم پیشنهادی بر روی مسئله فروشنده دوره گرد ۴۴۲ شهر

- [1] Y. Liu and K. M. Passino, "Swarm Intelligence: A Survey", In Proc. of 4th International Conference of SwarmIntelligence , 2005.
- [2] G. Dueck, "New optimization heuristics: The great deluge algorithm and the record-to-record travel," *Journal of Computational Physics*, vol. 104, pp. 86–92, 1993.
- [3] E. Burke, Y. Bykov, J. Newall, and S. Petrovic, "A timepredefined approach to course timetabling," *Yugoslav Journal of Operations Research*, vol. 13(2), pp. 139–151, 2003.
- [4] Burke E, Bykov Y, Newall J, Petrovic S. A time-predefined localsearch approach to exam timetabling problems. *IIE Trans*;36(6):509–28, 2004.
- [5] Nahas N, Ait-Kadi D, Nourelfath M. "A new approach for buffer allocation in unreliable production lines", *Int J Prod Econ*;103(2):873–81, 2006.
- [6] Nahas N, Nourelfath M, Ait-Kadi D. "Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series-parallel systems", *ReliabEngSystSaf*;92(2):211–22, 2007.
- [7] Nahas N, Nourelfath M, Ait-Kadi D. "A two-phase extended great deluge algorithm for the dynamic layout problem", In: *Internationalconference on industrial engineering and systems management*, Beijing, China; 2007.
- [8] Khatab A, Ait-Kadi D, Artiba A. "Selective maintenance optimization for multi-mission series-parallel systems", *Eur J Oper Res*, Submitted, 2008.
- [9] Dario L. S. and Obit J. H., "Great Deluge with Non-linear Decay Ratefor Solving Course Timetabling Problems", *International IEEE Conference "Intelligent Systems"*, vol 4, pp.8-18, 2008.
- [10] Holland, John H, "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, 1975.