مهندسی صنایع و مدیریت شریف
دوره‌ی ۱ - ۲۶، شماره‌ی ۱، ص. ۳-۱۰

کاربرد رنگ‌آمیزی دایره‌یی گراف و الگوریتم جامعه مورچگان در حل مسئله‌ی زمان‌بندی چرخشی کارگاهی باز

محمد مدرس (استاد)
مهسا قندهاری (دانشجوی دکتری)
دانشکده‌ی مهندسی صنایع، دانشگاه صنعتی شریف

در این نوشتار مدلی برای تعیین زمان‌بندی بهینه‌ی چرخشی کارگاهی باز طراحی می‌شود. برای انجام هر عملیات چندین منبع مورد نیاز است که به‌طور هم‌زمان باید از تمامی آن‌ها استفاده شود. این مسئله از لحاظ پیچیدگی محاسباتی در رده‌ی مسائل NP-hard قرار دارد. ابتدا نشان می‌دهیم که این مسئله را می‌توان به مسئله‌ی رنگ‌آمیزی دایره‌یی رئوس یک گراف تبدیل کرد. آنگاه، الگوریتمی در چارچوب روش فراابتکاری جامعه‌ی مورچگان طراحی می‌شود که می‌تواند در حل مسائل با اندازه‌ی بزرگ‌تر مورد استفاده قرار گیرد. برای بررسی و ارزیابی کارایی الگوریتم پیشنهادی، از یک دسته مسائل با عدد رنگی دایره‌یی غیرصحیح استفاده می‌شود و در نهایت نتایج حاصله با جواب‌های به‌دست آمده از روش تحلیلی مقایسه خواهد شد.

واژگان کلیدی: زمان‌بندی، رنگ‌آمیزی دایره‌یی گراف، الگوریتم جامعه‌ی مورچگان، برنامه‌ریزی عملیاتی.

modarres@sharif.edu
ghandehary@yahoo.com

مقدمه

به‌سبب پیچیدگی، مسائل زمان‌بندی و تعیین توالی بهینه، و نیز بهبود روش‌های تقریبی برای حل این‌گونه مسائل دغدغه‌ی بسیاری از محققان در چند دهه‌ی اخیر بوده است. این مسائل به تخصیص بهینه‌ی ماشین‌ها به کارها و نیز توالی و زمان‌بندی آنها می‌پردازد.

هریک از این مسائل چندین عملیات خاص را شامل می‌شود که هرکدام از این عملیات معمولاً توسط ماشین خاص و در مدت زمان معین انجام می‌گیرد؛ اگرچه در مواردی چندین ماشین ممکن است قابلیت انجام آن را داشته باشند.

این مسائل در ادبیات به سه دسته «جریان کارگاهی[۱]»، «کار کارگاهی[۲]» و «کارگاهی باز[۳]» دسته‌بندی شده‌اند. در مسائل کارگاهی باز تقدم عملیات در کارها مطرح نیست و عملیات به هر ترتیب می‌توانند انجام شوند.

مسائل زمان‌بندی کارگاه باز عملاً کاربردهای فراوانی دارند ـ نظیر تخصیص فرکانس، تعیین توالی بهینه در تولید، یا مسائل جدول‌بندی زمانی. در ادبیات روش‌ها و رویکردهای متعددی برای حل این مسائل ـ ازجمله برنامه‌ریزی ریاضی، سیستم‌های خبره، شبکه‌های عصبی، الگوریتم ژنتیک و منطق فازی[۱] ـ به‌کارگرفته شده است. جست‌وجو در ادبیات نشان می‌دهد که رویکرد به‌کار گرفته شده در این نوشتار، یعنی مدل‌سازی مسئله‌ی زمان‌بندی کارگاه باز با استفاده از رنگ‌آمیزی دایره‌یی، رویکردی بی‌سابقه است. در تنها نوشتاری که به کاربرد رنگ‌آمیزی دایره‌یی در مسئله‌ی برنامه‌ریزی چرخشی کارگاه باز اشاره دارد[۲] نشان داده شده است که مسئله‌ی زمان‌بندی چرخشی کارگاه باز که در آن هر عملیات فقط می‌تواند یک منبع را درگیر سازد، چنانچه با سه عدد پردازش‌گر انجام گیرد از دسته مسائل NP-hard است ولی چنانچه دو پردازش‌گر در آن دخیل باشند پیچیدگی این مسائل از نوع چندجمله‌یی است. همچنین ثابت شده است که این مسئله را اگر بتوان در یک بازه زمانی به طول سه واحد برنامه‌ریزی کرد، در زمان چندجمله‌یی قابل حل است. در مدل مطروحه در نوشتار یادشده، هر عملیات لزوماً با استفاده از تنها یک منبع انجام می‌شود و از طرفی تعداد منابع محدودند. این در حالی است که در نوشتار حاضر محدودیت‌های ذکرشده ـ نظیر تعداد منابع مورد نیاز هر عملیات ـ حذف و با بهره‌گیری از مفهوم رنگ‌آمیزی دایره‌یی الگوریتمی فراابتکاری ارائه شده است.

تحقیقات و روش‌های دیگر درمورد مسئله‌ی زمان‌بندی کارگاهی با اختصار ارائه می‌شود. در کلیه‌ی مسائل عنوان‌شده، هر عملیات تنها توسط یک ماشین یا منبع پردازش می‌شود. در تحقیقی مسئله‌ی زمان‌بندی کارگاه باز با هدف کمینه‌سازی زمان انجام یک دوره، در حالتی که تعداد کارها n است بررسی شده است.[۳] در تحقیق یادشده تعداد ماشین‌ها و همچنین تعداد عملیات هر کار m فرض می‌شود و درنتیجه هر عملیات با استفاده از یک ماشین و در مدت زمان معینی انجام

تاریخ: دریافت ۱۳۸۶/۷/۱۶، داوری ۱۳۸۶/۹/۱۹، پذیرش ۱۳۸۶/۱۱/۲۴.

می‌شود. آنها توانستند الگوریتمی با پیچیدگی چندجمله‌یی برای حل مسئله با فرض این که $m = 2$ ارائه دهند، ولی نشان دادند که برای $m > 2$ این مسائل در رده‌ی NP-hard قرار می‌گیرند. پس از آن محققین دیگری تعمیمی از الگوریتم یادشده[۳] با زمان خطی پیشنهاد نمودند[۴] که در آن زمان‌های آماده‌سازی و پردازش و برداشتن قطعه از روی ماشین به‌صورت جداگانه در نظر گرفته شده است. در همین راستا الگوریتمی با زمان چندجمله‌یی برای حل این مسئله در حالتی که $m = 2$ است ارائه شده است.[۵] در ادامه‌ی این تحقیقات، الگوریتم انشعاب و تحدید برای مسئله در حالتی که تعداد ماشین‌ها m است توسعه یافت.[۶] از این الگوریتم برای اولین‌بار در حل یک سری از مسائل موجود در ادبیات (مسائل نشانه[۴]) استفاده شد. این الگوریتم با استفاده از یک روش هوشمند در انتخاب مسیر بازگشت، بهبود داده شد.[۷] الگوریتم انشعاب و تحدید دیگری نیز براساس روش‌های «انتشار اجباری[۵]» برای محدودسازی فضای جست‌وجو ارائه شده است.[۸] از سوی دیگر، محققین الگوریتمی دیگر ازنوع چندجمله‌یی برای حل مسئله با مقدار دلخواه m، و با این پیش‌فرض که «جمع زمان‌های پردازش برای یک ماشین بزرگ‌تر یا مساوی است با کل زمان‌های پردازش ضرب‌در ضریبی ثابت»، ارائه کردند.[۹] دیگر الگوریتم پیشنهاد شده در همین راستا، الگوریتمی است با زمان خطی برای حالتی که $m = 3$ است و یکی از ماشین‌ها دو ماشین دیگر را تحت سلطه قرار می‌دهد.[۱۰] برای حل این مسئله در حالتی که یک یا دو ماشین سایر ماشین‌ها را تحت سلطه قرار می‌دهند نیز، الگوریتمی برای هر مقدار دلخواه m توسعه یافته است.[۱۱]

یک الگوریتم «تجمع تطبیقی[۶]» نیز براین اساس که حالت دوماشینه به‌صورت خطی قابل حل است توسعه داده شده است.[۱۲] از سوی دیگر، الگوریتم «تعبیه سازنده[۷]» براساس تحلیلی از ساختار ترکیبات بهینه و ترتیب کارها و ماشین‌ها ارائه شد.[۱۳] همچنین دو الگوریتم ابتکاری زمان‌بندی براساس لیست زمان‌بندی کارهای اجرایی با دو تقدم برای هر عملیات و تطبیق سازنده در یک گراف دوبخشی که با رویه‌یی بهبود دهنده با جست‌وجوی محلی دنبال می‌شود پیشنهاد شده است.[۱۴] با استفاده از روش جست‌وجوی ممنوع[۸] برای حل یک سری مسائل که به‌عنوان «نشانه» در ادبیات استفاده می‌شوند بهره گرفته شده است.[۱۵] یکی از محققین با استفاده از یک رویکرد تکرارشونده و بهبوددهنده در چارچوب تجزیه‌ی Benders، روشی متشکل از الگوریتم جست‌وجوی ممنوع برای بهبود محلی و الگوریتم ژنتیک برای جست‌وجوی کل فضای اصلی را توسعه داد.[۱۶،۱۷] این مسئله به‌صورت چندمرحله‌یی در نظر گرفته شده است،[۱۸] به این صورت که هر کار شامل m مرحله و هر مرحله توسط تعدادی ماشین‌های مشابه به‌طور موازی انجام می‌شود. برای حل این مسئله، زمانی که بیشترین تعداد ماشین‌ها ۲ است، سازوکارِ ایجاد جواب توسط الگوریتم جامعه‌ی مورچگان با روش جست‌وجوی شعاعی ترکیب شده است.[۱۹]

در این نوشتار مسئله‌ی زمان‌بندی چرخشی کارگاه باز با تابع هدف کمینه‌سازی طول یک دوره در حالتی که هر عملیات می‌تواند با استفاده از چندین منبع به‌صورت هم‌زمان انجام شود، مورد بررسی قرار گرفته است. در نهایت برنامه به‌دست آمده می‌تواند با تکرار تعداد محدودی چرخه‌ی فعالیت‌ها به این مقدار کمینه دست یابد، به‌نحوی که این مقدار دست‌کم در مسائلی خاص از مقدار ارائه‌شده در مسئله‌ی زمان‌بندی کارگاه باز کم‌تر است. این مسئله در حالت‌های خاص خود، با مسائل مهمی چون مسئله‌ی تخصیص فرکانس، زمان‌بندی ترافیک، و زمان‌بندی کارگاهی رباتیک قابل انطباق است. سپس با بهره‌گیری از الگوریتم جامعه‌ی مورچگان به‌همراه یک روش بهینه‌سازی خطی، نسبت به حل تعدادی مسئله که در آنها مقدار تابع هدف در حالت معمولی و حالت چرخشی متفاوت است اقدام، و جواب‌ها با جواب‌های حاصل از روش‌های تحلیلی مقایسه شد. در ادامه‌ی این مطلب، مسئله‌ی چرخشی کارگاهی باز را با ذکر تمامی پارامترها به‌طورکامل تعریف می‌کنیم. سپس مسئله‌ی مطرح شده را با استفاده از مفهوم رنگ‌آمیزی دایره‌یی گراف در قالب یک مسئله‌ی برنامه‌ریزی مختلط مدل‌سازی می‌کنیم. در ادامه، با استفاده از رهیافت فراابتکاری مورچگان، برای حل این مسئله الگوریتمی مناسب طراحی می‌کنیم. از آنجا که این مسئله حالت کلی‌تر مسئله‌ی کارگاهی باز است و تاکنون در ادبیات مورد بررسی قرار نگرفته، نتایج به‌دست آمده از به‌کارگیری این الگوریتم برای دسته‌یی از مسائل که در آنها زمان اجرای یک دوره‌ی کامل از کارها در حالت چرخشی و غیر چرخشی یکسان نیست، در بخش بعد آمده است. پاسخ‌ها برای تمام مسائل آزموده‌شده بهینه‌اند و برای اندازه‌های بزرگ مسائل آزموده‌شده، استفاده از الگوریتم مورچگان در مقایسه با حل مسئله توسط نرم‌افزار GAMS کارایی دارد.

تعریف مسئله

چنان که اشاره شد مسئله‌ی مورد بحث در این نوشتار تعمیمی از مسئله‌ی زمان‌بندی کارگاه باز در حالت چرخشی است. به این معنی که برنامه‌ی زمان‌بندی به‌طور محدود یا نامحدود طی زمان تکرار می‌شود. در مسئله‌ی زمان‌بندی کارگاه باز، با n کار مختلف سروکار داریم. هر کار شامل تعدادی عملیات است که علی‌رغم مسئله‌ی کارگاهی بازِ متداول به چند منبع نیاز دارد به‌گونه‌یی که هر زمان که عملیات آغاز می‌شود همه‌ی منابع به‌صورت هم‌زمان درگیر می‌شوند. برای عملیات‌ها رابطه‌ی تقدم وجود ندارد. یک ماشین (منبع) در هر لحظه فقط می‌تواند توسط یک عملیات به‌کار گرفته شود. در این مسئله فرض می‌کنیم که قطع عملیات مجاز نیست و در هر زمان حداکثر یک عملیات مربوط به هر کار قابل انجام است. در این مسئله یک عملیات می‌تواند به بیش از یک کار تعلق داشته باشد. تمامی منابع در زمره‌ی تجدیدشونده هستند.

هدف تعیین توالی انجام عملیات به‌نحوی است که زمان انجام کارها در حالتی که برنامه‌ی ارائه‌شده چرخشی است کمینه شود. منظور از برنامه‌ی چرخشی آن است که برنامه‌ی ارائه‌شده در طی زمان بدون توقف تکرار می‌شود؛ در این حالت نشان می‌دهیم که متوسط زمان اجرای یک برنامه برای بیش از یک تکرار از مدت زمان انجام یک تکرار کم‌تر است. همچنین نشان می‌دهیم که تعداد تکرارها برای رسیدن به مقدار کمینه برای متوسط زمانِ انجام یک دوره محدود است.

پارامترهای مسئله

دو عملیات s, s' ناسازگار نامیده می‌شوند اگر هم‌زمان قابل اجرا نباشند؛ به‌عبارت‌دیگر متعلق به یک کار باشند یا هر دو منبعی مشترک داشته باشند. مجموعه عملیاتی که با s ناسازگارند را با $I(s)$ نشان می‌دهیم.

J: مجموعه‌ی کارها $J = \{j_1, \ldots, j_n\}$؛ S: مجموعه‌ی کل عملیات $S = \{s_1, \ldots, s_{|s|}\}$؛ R: مجموعه‌ی کل منابع تجدیدپذیر؛ $S(T) \subset S$: مجموعه‌ی عملیات مربوط به کار T؛ $S(R) \subset S$: مجموعه‌ی عملیاتی که نیاز به منبع R دارند؛ $R(s) \subset R$: مجموعه‌ی منابعی که عملیات s برای اجرا به آن نیاز دارد؛ t_s: زمان شروع عملیات s، $s \in S$.

مدل‌سازی مسئله

چنان که پیش‌تر اشاره شد، در این مسئله هدف تعیین توالی انجام عملیات به‌نحوی است که زمان انجام تمام کارها در حالتی که برنامه‌ی ارائه‌شده چرخشی باشد کمینه شود. می‌توان نشان داد که نسبت کل زمان انجام برنامه‌ها به تعداد برنامه‌ها از مدت زمان انجام یک برنامه کم‌تر یا با آن مساوی خواهد شد.

الف) مدل گراف

مسئله را با استفاده از مفاهیم رنگ‌آمیزی دایره‌یی فرموله می‌کنیم. C را دایره‌یی با طول اقلیدسی r در نظر بگیرید. یک رنگ‌آمیزی دایره‌یی با اندازه r برای گراف $G=(V,E)$ نگاشتی است که به هر رأس x از گراف G یک کمان باز $c(x)$ از C به طول یک واحد را تخصیص می‌دهد، به‌طوری که برای هر یال مثل (x,y) دو کمان متناظر با x و y هیچ اشتراکی نداشته باشند. اگر بتوان چنین نگاشتی برای گراف G پیدا کرد، آنگاه گفته می‌شود گراف G به‌صورت r دایره‌یی رنگ‌پذیر است. حداقل مقدار r را که به ازاء آن گراف G به‌صورت r دایره‌یی رنگ‌پذیر است، عدد رنگی دایره‌یی گراف G می‌نامند.

واضح است که در رنگ‌آمیزی دایره‌یی به جای این که همانند رنگ‌آمیزی بازه‌یی به هر رأس زیربازه‌یی از یک بازه به‌طول وزن آن رأس اختصاص دهیم، کمانی از یک دایره با طول اقلیدسی به اندازه‌ی وزن آن رأس اختصاص می‌دهیم به‌طوری که رئوس ناسازگار با زیربازه‌های مجزا (بدون اشتراک) متناظر شوند. پس در مسئله‌ی زمان‌بندی کارگاه باز، هر عملیات را به‌صورت یک رأس v از گراف $G=(V,E)$ در نظر می‌گیریم و بین دو رأس $v_۱, v_۲$ این گراف در صورتی یال وجود دارد که دو رأس متناظر با دو عملیات ناسازگار باشند. واضح است که هر رنگ‌آمیزی برای رئوس گرافی که به‌دست می‌آید معادل یک زمان‌بندی موجه برای مسئله‌ی زمان‌بندی کارگاه باز است.

در صورتی که زمان‌های پردازش برای عملیات‌ها همگی مساوی نباشند به هر رأس وزنی معادل با زمان پردازش عملیات متناظر با آن رأس داده می‌شود و در این صورت هر رنگ‌آمیزی برای گراف وزن‌دار معادل یک جواب موجه برای مسئله است. از آنجا که هدف مسئله پیداکردنِ برنامه‌یی با کم‌ترین زمان اجرا است، در حالتی که برنامه تکرارشدنی باشد، این برنامه معادل با یک رنگ‌آمیزی دایره‌یی به‌اندازه‌ی r در گراف $G=(V,E)$ است، به‌طوری‌که r عدد رنگی دایره‌یی گراف مورد نظر است.

ب) مدل ریاضی

در اینجا مسئله‌ی رنگ‌آمیزی دایره‌یی برای گراف $G=(V,E)$ را به‌صورت یک مدل ریاضی مطرح می‌کنیم. در مدل اول فرض شده است که مدت زمان لازم برای اجرای هر عملیات معادل یک واحد زمانی است. اگر زمان‌بندی برای انجام همه‌ی کارها در مدت زمان r قابل اجرا باشد و t_s را زمان آغاز شدن عملیات s در نظر بگیریم، مسئله‌ی برنامه‌ریزی به‌صورت برنامه‌ریزی مختلط زیر قابل طرح است.

$$Min\ \ Z=r$$
$$S.t.$$
$$۱\leq |t_s-t_{s'}|\leq r-۱,\quad s'\in I(s),\quad s\in S.$$
$$s\in S\quad ۰\leq t_s\leq r \qquad (۱)$$

با استفاده از متغیرهای صفر و ۱ این مدل قابل حل خواهد بود. در حالتی که مدت زمان انجام عملیات s_i را با w_i نشان دهیم، مدل ریاضی عبارت خواهد بود از:

$$MinZ=r$$
$$\begin{cases} r\geq t_{s_i}, & s_i\in S \\ \left.\begin{array}{l} t_{s_i}-t_{s_j}\geq w_i \\ t_{s_i}-t_{s_j}\leq r-w_j \end{array}\right\} & if, t_{s_i}\geq t_{s_j}, s_i\in I(s_j) \\ \left.\begin{array}{l} t_{s_j}-t_{s_i}\geq w_j \\ t_{s_j}-t_{s_i}\leq r-w_i \end{array}\right\} & if, t_{s_j}\geq t_{s_i}, s_i\in I(s_j) \end{cases}$$
$$(۲)$$

الگوریتم جامعه‌ی مورچگان

الف) نحوه‌ی ارائه‌ی جواب

پیش‌تر عنوان کردیم که S مجموعه‌ی تمامی عملیاتی است که برای تکمیل کارها باید انجام شود. به‌علت ماهیت مسئله‌ی زمان‌بندی کارگاه باز، هر ترتیبی برای انجام عملیات قابل قبول است. هر جواب معادل یک ترتیب از مجموعه‌ی S است به‌طوری که هر عملیات فقط و فقط یک‌بار در ترتیب آورده شود. پس اگر S دارای $|S|$ عضو باشد، هر جواب یک ترتیب از اعداد ۱ تا $|S|$ خواهد بود.

ب) طراحی الگوریتم جامعه‌ی مورچگان

الگوریتم جامعه‌ی مورچگان روشی فراابتکاری است برای یافتن جواب‌های بهینه برای مسائل بسیار سخت بهینه‌سازی ترکیبی، که ایده‌ی اصلی خود را از رفتار مورچگان برای یافتن غذا و ترشح فرومون در مسیر گرفته است. در الگوریتم جامعه‌ی مورچگان از مورچه‌های مصنوعی برای یافتن جواب‌های بهینه استفاده می‌کنیم. در الگوریتم پیشنهادی، مورچه‌ها در مسیر بین عملیات به‌صورت تصادفی و براساس میزان فرومونی که در مسیر است حرکت می‌کنند و بر پایه‌ی مطلوبیت مسیر، میزان خاصی فرومون در مسیر ترشح می‌کند. طبیعی است مسیری که تاکنون در آن فرومون بیشتری جمع شده است از مطلوبیت بیشتری برخوردار است. با توجه به ساختار جواب موجه، هر مورچه تنها یک بار می‌تواند از یک گره (عملیات) عبور کند. واضح است که تمامی مسیرهای مجاز، یال‌های گراف کاملی هستند که رئوس آن را اعضای S تشکیل می‌دهند.

ج) تعیین فرضیات الگوریتم و فهرست علائم

n_a: جمعیت جامعه‌ی مورچگان؛ S_{iter}: مجموعه جواب‌های تولیدشده در این تکرار توسط n_a مورچه که ماتریسی است شامل n ستون و حداکثر n_a سطر؛ x_{bs}: بهترین جواب به‌دست آمده تاکنون توسط الگوریتم؛ x_{rb}: بهترین جواب تولیدشده در این تکرار؛ x_k: جواب ناقص تولیدشده تا مرحله‌ی k؛ $N(x_k)$: مجموعه‌ی تمام گره‌هایی که می‌توانند به جواب x_k اضافه شوند؛ $N_{res}(x_k)$: مجموعه‌ی اصلاح‌شده‌ی $N(x_k)$؛ cf: ضریبی که میزان همگرایی جواب‌های به‌دست آمده توسط الگوریتم را نشان می‌دهد و با مقدار از قبل تعیین شده‌ی $cf-limit$ مقایسه می‌شود؛ $t(s_i,x)$: زمان شروع عملیات s_i در جواب x؛ ρ: ضریب تبخیر فرومون؛ τ: مقادیر فرومون؛ η: مقادیر کاوشی.

د) نحوه‌ی تولید جواب در الگوریتم جامعه‌ی مورچگان

زمانی که مورچه‌یی از یک گره عبور می‌کند در حرکت‌های بعد مجاز به عبور از آن گره نیست. جواب موجه مسیری است که یک مورچه می‌پیماید تا از تمامی گره‌های یک گراف کامل فقط یک بار عبور کند. مسیری که مورچه‌ی k تاکنون پیموده معادل x_k در نظر می‌گیریم. این مورچه در مرحله‌ی بعد فقط قادر به سفر به گره‌های موجود در S به‌غیر از گره‌های موجود در x_k است. $N(x_k)$ را مجموعه‌ی تمامی گره‌هایی در نظر بگیرید که می‌توانند به جواب x_k اضافه شوند.

$$N(x_k) = S|\{s_i\, exists\, in\, S\}$$

می‌توان $N(x_k)$ را به‌صورت خلاصه‌تر نوشت و از بررسی گره‌هایی (رئوسی) که مکان آنها بدیهی است، خودداری کرد.

$$N_{res}(x_k) = \{s_i | s_i \in N(x_k), I(s_i) \cap N(x_k) \neq \phi\}$$

۱. الگوریتم اصلی

قدم‌های الگوریتم عبارت‌اند از:

قدم ۱. ورود پارامترهای مسئله.

قدم ۲. $bs - update \leftarrow false, cf \leftarrow ۰, x_{bs} \leftarrow Null$.

قدم ۳. فرومون مسیرها را برای آغاز حرکت، طبق الگوریتم ۶ فرومون‌دهی کنید.

قدم ۴. شرایط خاتمه را طبق شرایط خاتمه (در انتهای الگوریتم‌های شرح داده شده) کنترل کنید. اگر شرایط برقرار نبود، به قدم بعدی بروید وگرنه به قدم ۱۸ بروید.

قدم ۵. $s_{iter} \leftarrow \phi$.

قدم ۶. برای n_a مورچه طبق الگوریتم ۲ جواب تولید کنید.

قدم ۷. مقدار مطلوبیت هریک از جواب‌های تولید شده توسط n_a مورچه را مطابق الگوریتم ۳ به‌دست آورید.

قدم ۸. x_{rb} را معادل جوابی که کم‌ترین مقدار تابع هدف (بیشترین مطلوبیت) را در این تکرار به خود اختصاص داده بگیرید.

قدم ۹. مقدار تابع هدف نظیر بهترین جواب تولیدشده در این تکرار را با مقدار تابع هدف نظیر بهترین جواب تولیدشده تاکنون مقایسه کنید و چنانچه $f(x_{bs}) > f(x_{rb})$ قرار دهید $x_{bs} \leftarrow x_{rb}$.

قدم ۱۰. مقدار فرومون در مسیرها را مطابق الگوریتم ۵ بهنگام کنید.

قدم ۱۱. ضریب همگرایی cf را مطابق فرمول زیر بهنگام کنید.

$$cf \leftarrow ۲. \left(\frac{\sum_{s_i \in S} \sum_{s_j \in Ri} \max\{\tau_{\max} - \tau_{ij}, \tau_{ij} - \tau_{\min}\}}{|T|(\tau_{\max} - \tau_{\min})} - ۰/۵ \right) \tag{۳}$$

قدم ۱۲. اگر $cf > cf\_limit$ بود به قدم بعد وگرنه به قدم ۱۷ بروید.

قدم ۱۳. اگر $bs\_update = true$ بود به قدم ۱۸ وگرنه به قدم بعد بروید.

قدم ۱۴. مقدار فرومون را طبق الگوریتم ۶ بهنگام کنید.

قدم ۱۵. جواب‌هایی را که مقادیر فرومون براساس آنها بهنگام می‌شود، تهی کنید.

قدم ۱۶. $bs\_update \leftarrow true$.

قدم ۱۷. به قدم ۴ بازگردید.

قدم ۱۸. x_{bs} را به‌عنوان بهترین جواب به‌دست آمده در نظر بگیرید.

۲. الگوریتم تولید جواب

قدم‌های الگوریتم عبارت‌اند از:

قدم ۱. $j = ۱$.

قدم ۲. مقدار جواب جزئی اولیه برای مورچه‌ی j را معادل $x_k = <۱>$ در نظر بگیرید.

قدم ۳. اگر $f(x_k) \leq f(x_{bs}) + ۱, N_{res}(x_k) \neq \phi$ به قدم بعدی وگرنه به قدم ۶ بروید.

قدم ۴. $c \in N_{res}(x_k)$ را به‌طور تصادفی و براساس احتمالات $P(c|\tau, \eta)$ تخصیص داده شده به اعضای $N_{res}(x_k)$ براساس الگوریتم ۶ انتخاب کنید.

قدم ۵. به x_k، مقدار c را اضافه کنید و به قدم ۳ بازگردید.

قدم ۶. اگر $N(x_k) \neq \phi$، تمامی اعضای آن را به هر ترتیب دلخواه به x_k اضافه کنید.

قدم ۷. اگر $j < n_a$ آنگاه $j = j + ۱$ و به قدم ۲ بازگردید وگرنه به قدم ۷ بروید.

قدم ۸. جواب‌های تولیدشده را به‌عنوان خروجی در نظر بگیرید.

۳. الگوریتم تعیین مطلوبیت جواب‌ها

قدم ۱. تقدم زمان شروع عملیات‌ها را براساس ترتیب عنوان‌شده در جواب x_k تعیین کنید. مثلاً در جواب به‌دست آمده $x_k = (s^{(۱)}, s^{(۲)}, ..., s^{(|s|)})$ تقدم زمانی به‌صورت $t(s^{(۱)}, x_k) \leq t(s^{(۲)}, x_k) \leq ... \leq t(s^{(|s|)}, x_k)$ است.

قدم ۲. مقدار تابع هدف را با حل برنامه‌ریزی زیر به‌دست آورید:

اگر زمان‌های پردازش همگی مساوی باشند:

$$MinZ = f(x)$$

$$۰ \leq t(s_i, x) \leq f(x) \ldots\ldots\ldots\ldots for..\forall s_i \in S$$

$$۱ \leq t(s_i, x) - t(s_j, x) \leq f(x) - ۱ \ldots.. for..t(s_i, x) \geq t(s_j, x)$$

درغیر این صورت:

$$MinZ = f(x)$$

$$۰ \leq t(s_i, x) \leq f(x) \ldots\ldots\ldots\ldots for..\forall s_i \in S$$

$$w_j \leq t(s_i, x) - t(s_j, x) \leq f(x) - w_i \ldots.. for..t(s_i, x) \geq t(s_j, x)$$

۴. الگوریتم بهنگام‌سازی فرومون

برای بهنگام‌سازی فرومون از مقادیر بهترین جواب به‌دست آمده تاکنون، و نیز از بهترین جواب به‌دست آمده در تکرار قبلی استفاده می‌شود. برای تمام مقدارهای $۱ \leq i, j \leq n, i \neq j$ مقادیر τ را طبق قدم‌های زیر بهنگام‌سازید:

قدم ۱. مقدار تابع $\delta(s_i, s_j, x)$ را برحسب جواب x چنین تعیین کنید:

اگر در جواب x زمان انجام s_i زودتر از زمان انجام s_j تعیین شده بود، $\delta(s_i, s_j, x) =$ ۱ وگرنه معادل با صفر در نظر بگیرید.

قدم ۲. فرومون در مسیر i به j را چنین بهنگام کنید:

$$\tau_{ij} \leftarrow \tau_{ij} + \rho(\delta(s_i, s_j, x) - \tau_{ij})$$

۵. الگوریتم تعیین مقدار احتمالات

قدم ۱. با توجه به جواب جزئی x_k برای تمامی اعضای $s_i \in N(x_k)$، زودترین زمانی که عملیات می‌تواند آغاز شود $(t_{es}(s_i, x_k))$ را بیابید. این مقدار معادل بزرگ‌ترین مقدار مربوط به مجموع زمان‌های متناظر با رئوس زیرگراف‌های کامل در $x_k + s_i$ است.

قدم ۲. مقدار کاوش (η) را چنین محاسبه کنید:

$$\eta(s_i) = \frac{\frac{1}{1+t_{es}(s_i,x^k)}}{\sum_{s_j \in N(x^k)} \frac{1}{1+t_{es}(s_j,x^k)}}, \forall s_i \in N(x^k)$$

مقدار $t_{es}(s,x)$ معادل با اندازه بزرگ‌ترین زیرگراف کامل گراف القایی $G[s,x]$ است.

قدم ۳. مقدار احتمال نظیر هر رأس در $N(x_k)$ را با توجه به η و مقدار فرومون‌ها به دست آورید:

$$TP(s_i|,\eta) = \frac{(\tau_{ij})^{\alpha}\eta(s_i)}{\sum_{s_k \in N(x^k)}(\tau_{kj})^{\alpha}\eta(s_k)}, \forall s_i \in N(x^k)$$

۶. الگوریتم مقداردهی اولیه‌ی فرومون

تمامی مقادیر فرومون‌های نظیر هر یال را مساوی یکدیگر و معادل ۰/۵ قرار دهید.

قدم پایانی

برای خاتمه باید این دو شرط برقرار باشد:

۱. $cf > cf\_limit$.

۲. $bs\_update = true$.

نتایج محاسباتی

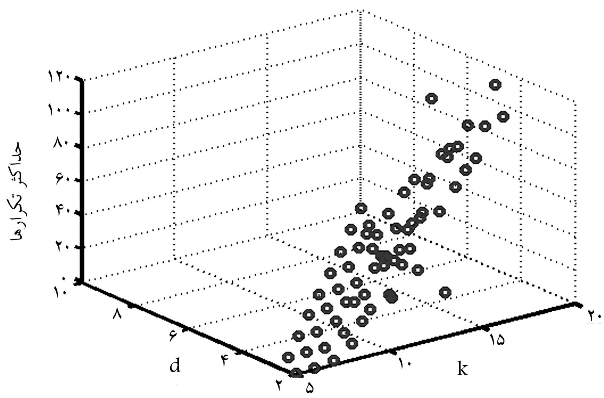
بهترین مقدار پارامتر ρ در ۱۰ اجرای مسئله برای $k < ۲۰$ و۵ $n_a =$ ، از بین مقدارهای ۰/۲، ۰/۵، ۰/۸، مقدار ۰/۲ است. از آنجا که برای مقادیر $k < ۲۰$ و برای ۵ $n_a =$ پس از تعداد قابل قبولی از تکرارها به جواب بهینه رسیدیم، تحلیلی بر روی این پارامتر انجام ندادیم. برای مقدارα (ضریب اهمیت فرومون) نتایج تحلیل در جدول ۱ آورده شده است.

جدول ۱. تحلیل نتایج براساس پارامتر α.

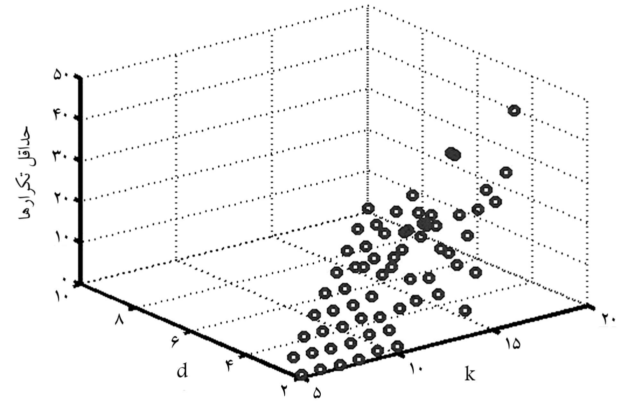
۲			۱/۵			۱			۰/۵			ضریب اهمیت فرومون		
میانگین تکرار	حداقل تکرار	حداکثر تکرار	میانگین تکرار	حداقل تکرار	حداکثر تکرار	میانگین تکرار	حداقل تکرار	حداکثر تکرار	میانگین تکرار	حداقل تکرار	حداکثر تکرار	d	k	
۳/۷	۱	۱۰	۳/۱	۱	۱۲	۳/۶	۱	۱۰	۲/۸	۱	۸	۲	۸	
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۳	۸	
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۴	۸	
۲۰/۴	۴	۴۲	۱۳/۴	۳	۳۳	۱۰/۱	۲	۳۴	۱۰/۵	۱	۳۸	۲	۱۰	
۱۲/۴	۳	۲۰	۸/۶	۲	۱۳	۷/۱	۲	۱۱	۷/۵	۳	۱۲	۳	۱۰	
۳/۷	۱	۸	۲/۵	۱	۷	۱/۸	۱	۴	۲/۱	۱	۴	۴	۱۰	
۱/۵	۱	۳	۱	۱	۱	۱	۱	۱	۱	۱	۱	۵	۱۰	
۲۰/۵	۲	۷۰	۳۲	۵	۷۸	۲۵	۲	۷۷	۲۲/۸	۳	۷۳	۳	۱۵	
۱۲/۲	۴	۷۲	۱۵	۲	۷۴	۱۱/۳	۳	۷۵	۱۳/۱	۲	۶۹	۴	۱۵	
۷/۸	۵	۱۲	۷/۱	۲	۱۹	۶/۸	۴	۱۷	۸/۷	۶	۲۱	۵	۱۵	
۶/۳	۳	۱۱	۵/۸	۲	۸	۵/۳	۲	۸	۵/۱	۱	۱۱	۶	۱۵	
۱/۳	۱	۳	۲/۱	۱	۵	۱/۱	۱	۲	۱/۵	۱	۴	۷	۱۵	
۳۰/۲	۴	۸۵	۲۶/۴	۶	۷۹	۲۷	۶	۷۵	۳۱/۵	۱۱	۸۲	۴	۱۷	
۲۵/۸	۶	۵۱	۲۱	۷	۴۰	۲۲	۸	۴۳	۲۰/۳	۵	۵۰	۵	۱۷	
۲۰/۲	۵	۳۲	۲۰/۲	۱۱	۳۱	۱۵/۸	۹	۲۹	۱۴/۸	۱۱	۲۶	۶	۱۷	
۲/۵	۱	۴	۲/۴	۲	۶	۱/۵	۱	۳	۱/۸	۱	۲	۷	۱۷	
۲/۶	۲	۵	۳/۲	۱	۷	۱	۱	۱	۲/۴	۱	۳	۸	۱۷	
۸۱/۴	۲۳	۱۵۴	۹۲/۸	۴۱	۱۷۶	۸۹	۲۴	۱۱۰	۱۰۰	۳۴	۲۱۱	۵	۲۰	
۶۸	۱۴	۸۸	۵۳	۱۱	۷۳	۶۱	۱۲	۷۹	۶۰	۸	۶۹	۶	۲۰	
۵۰/۳	۸	۶۴	۴۹/۱	۱۳	۵۹	۴۸	۲۳	۵۵	۴۳/۲	۳۱	۶۳	۷	۲۰	
۱۵/۲	۳	۴۲	۱۵/۹	۴	۳۸	۱۵/۳	۳	۳۳	۱۷/۲	۲	۴۱	۸	۲۰	
۶/۸	۲	۱۰	۵/۲	۴	۷	۴/۳	۳	۶	۶/۱	۲	۱۱	۹	۲۰	
۱/۸	۱	۳	۱/۵	۱	۲	۱/۲	۱	۲	۱/۴	۱	۳	۱۰	۲۰	

جدول ۲. تحلیل تعداد تکرارها تا رسیدن به جواب بهینه.

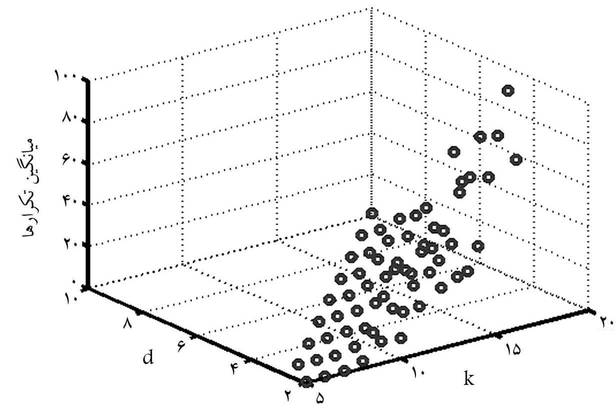
k	d	حداکثر تکرار	حداقل تکرار	میانگین تکرار	K	d	حداکثر تکرار	حداقل تکرار	میانگین تکرار
۵	۲	۱	۱	۱	۱۵	۳	۷۷	۲	۲۵
۶	۲	۱	۱	۱	۱۵	۴	۷۵	۳	۱۱٫۳
۶	۳	۱	۱	۱	۱۵	۵	۱۷	۴	۶٫۸
۷	۲	۳	۱	۱٫۲	۱۵	۶	۸	۲	۵٫۳
۷	۳	۱	۱	۱	۱۵	۷	۲	۱	۱٫۱
۸	۲	۱۰	۱	۳٫۶	۱۶	۴	۸۵	۹	۵۵
۸	۳	۱	۱	۱	۱۶	۵	۴۲	۳	۲۲٫۵
۸	۴	۱	۱	۱	۱۶	۶	۱۰	۷	۶
۹	۲	۲۸	۲	۱۰٫۹	۱۶	۷	۱۹	۲	۵٫۷
۹	۳	۷	۱	۲٫۵	۱۶	۸	۱	۱	۱
۹	۴	۱	۱	۱	۱۷	۴	۷۵	۶	۲۷
۱۰	۲	۳۴	۲	۱۰٫۱	۱۷	۵	۴۳	۸	۲۲
۱۰	۳	۱۱	۲	۷٫۱	۱۷	۶	۲۹	۹	۱۵٫۸
۱۰	۴	۴	۱	۱٫۸	۱۷	۷	۳	۱	۱٫۵
۱۰	۵	۱	۱	۱	۱۷	۸	۱	۱	۱
۱۱	۲	۷۰	۱۲	۲۳	۱۸	۴	۹۸	۲۲	۷۸
۱۱	۳	۴۷	۲	۱۶٫۳	۱۸	۵	۷۹	۱۱	۵۲٫۳
۱۱	۴	۱۳	۲	۱٫۲	۱۸	۶	۱۰۱	۵	۲۰٫۳
۱۱	۵	۲	۱	۱٫۱	۱۸	۷	۳۸	۱۵	۲۲
۱۲	۲	۹۴	۲۹	۴۳	۱۸	۸	۶	۳	۴٫۳
۱۲	۳	۴۸	۲۵	۳۱	۱۸	۹	۲	۱	۱٫۳
۱۲	۴	۴۲	۱	۱۴٫۲	۱۹	۴	۱۰۱	۴۳	۶۴
۱۲	۵	۳	۱	۱٫۸	۱۹	۵	۶۹	۲۱	۵۰
۱۲	۶	۱	۱	۱	۱۹	۶	۶۸	۱۲	۴۲
۱۳	۲	۲۷	۴۵	۳۳	۱۹	۷	۲۳	۹	۱۴
۱۳	۳	۳۳	۲۵	۲۹	۱۹	۸	۷	۲	۴٫۲
۱۳	۴	۱۰	۲	۴٫۳	۱۹	۹	۲	۱	۱٫۲
۱۳	۵	۵	۲	۳٫۱	۲۰	۵	۱۱۰	۲۴	۸۹
۱۳	۶	۱	۱	۱	۲۰	۶	۷۹	۱۲	۶۱
۱۴	۴	۳۶	۲۴	۳۱	۲۰	۷	۵۵	۲۳	۴۸
۱۴	۵	۱۹	۸	۱۶٫۸	۲۰	۸	۳۳	۳	۱۵٫۳
۱۴	۶	۲	۵	۳٫۱	۲۰	۹	۶	۳	۴٫۳
۱۴	۷	۱	۱	۱	۲۰	۱۰	۲	۱	۱٫۲



شکل ۱. نمایش حداکثر تعداد تکرارهای الگوریتم بر حسب k,d.



شکل ۲. نمایش کم‌ترین تعداد تکرارهای الگوریتم برحسب k,d.



شکل ۳. نمایش کم‌ترین تعداد تکرارهای الگوریتم برحسب k,d.

به‌علت ماهیت دایره‌یی یا سیلندری مسئله‌ی برنامه‌ریزی معرفی شده، برای آزمون روش از دسته مسائلی که دارای عدد رنگی دایره‌یی معلوم $\frac{k}{d}$ هستند، استفاده شده است. این مسائل با استفاده از تعریف زیر تولیدشده‌اند:

تعریف ۵ـ۱: برای هر دو عدد صحیح k و d که $k \geq d \geq ۱$ باشد، رنگ‌آمیزی (k, d) برای گراف G، رنگ‌آمیزی است که با رنگ‌های $\{۰, ۱, ..., k - ۱\}$ رئوس گراف را چنان رنگ می‌کند که اختلاف شماره رنگ‌های هر دو رأس مجاور بین مقادیر d و $k - d$ باشد. به عبارت دیگر:

$$(x, y) \in E(G) \Rightarrow d \leq |c(x) - c(y)| \leq k - d \qquad (۴)$$

عدد رنگی دایره‌یی در این حالت معادل $\frac{k}{d}$ است، اگر مقدار کمینه‌یی که به‌ازاء آن یک (k, d) رنگ‌آمیزی برای گراف G وجود دارد معادل $\frac{k}{d}$ باشد. برای ایجاد گرافی با عدد رنگی دایره‌یی $\frac{k}{d}$، با استفاده از تعریف بالاگراف G را با k رأس چنین تعریف می‌کنیم: هر دو رأس v_i و v_j ازگراف G مجاورند اگر $d \leq |i - j| \leq k - d$ باشد. مسئله‌ی زمان‌بندی متناظر با این گراف شامل k عملیات است $S = \{s_۱, s_۲, ..., s_k\}$ به‌طوری‌که هر دو عملیات s_i و s_j ناسازگارند اگر و تنها اگر $d \leq |i - j| \leq k - d$ باشد.

الگوریتم ذکرشده در بخش قبل با استفاده از نرم‌افزار $Matlab$.۷ برنامه‌نویسی و اجرا شده و به‌منظور نمایش چگونگی کارکرد الگوریتم، شماره‌ی تکراری را که در آن جواب بهینه یافت شده، ذکر کرده‌ایم. در تمامی مسائل حل‌شده، جواب بهینه قبل از همگرایی الگوریتم حاصل شده است. نتایج برای مقادیر $k < ۲۰$ و$n_a = ۵$ و$\rho = ۰/۲$ در جدول ۲ آمده است. این نتایج به‌صورت شماتیک در شکل‌های ۱ تا ۳ آورده شده است. چنان‌که مشاهده می‌شود کارایی الگوریتم برای حالت‌هایی که نسبت k به d کم‌تر از ۳ است، قابل ملاحظه است.

نتیجه‌گیری

در این نوشتار مسئله‌ی زمان‌بندی چرخشی یک کارگاه باز به‌منظور کمینه‌سازی زمان یک دوره بررسی شد. این مسئله با استفاده از مفهوم رنگ‌آمیزی دایره‌یی مدل‌سازی شد، و سپس الگوریتم جامعه‌ی مورچگان، به‌ویژه برای مسائل با اندازه‌ی بزرگ طراحی شد. برای بررسی کارایی این الگوریتم از یک مجموعه مسائل تولیدشده براساس مقدار تابع هدف استفاده شد. جواب‌های حاصله با جواب‌های به‌دست آمده در یک روش تحلیلی مقایسه شد. جواب‌های به‌دست آمده برای مقادیر k کم‌تر از ۲۰ همگی بهینه‌اند. ضمناً برای مقادیر بزرگ‌تر از ۱۵ نتایج حاصل نشان می‌دهد که الگوریتم پیشنهادی علاوه بر این‌که به جواب بهینه منتهی می‌شود در زمان کم‌تری (در مقایسه با نرم‌افزار GAMS) به جواب بهینه یا نزدیک به بهینه می‌رسد.

پانوشت

1. flow shop
2. job shop
3. open shop
4. benchmark programs
5. constrained propagation
6. matching aggregation
7. constructive insertion
8. tabu search

منابع

1. Jones, A., and Rabelo, L.C. *Survey of Job Shop Scheduling Techniques,* NISTR, National Institute of Standards

and Technology, Gaithersburg, MD (1998).

2. Kubale, M., and Nadolski, A. "Chromatic scheduling in a cyclic open shop", *European Journal of Operation Research* (to appear).
3. Gonzales, T., and Sahni, S. "Open shop scheduling to minimize finish time", *Journal of the Association for Computing Machinery,* **23**(4), pp. 665-679 (1976).
4. Strusevich, V.A. "Two machine open shop scheduling problem with setup, processing and removal times separated", *Computers and Operations Research,* **20**(6), pp. 597-611 (1993).
5. Pinedo, M. *Scheduling: theory, algorithms and systems,* Englewood cliffs, NJ: Prentice Hall, (1995).
6. Brucker, P.; Hurink, J.; Jurish, B., and Wostmann, B. "A branch and bound algorithms for the open shop problem", *Discrete Applied Mathematics,* **76**, pp. 43-59 (1997).
7. Gueret, C.; Jussien, N., and Prince, C. "Using intelligent backtracking to improve branch-and-bound methods: An application to open shop problems", *European Journal of Operational Research,* **127**, pp. 344-354 (2000).
8. Dorndorf, U.; Pesch. E., and Phan-huy, T. "Solving the open shop scheduling problem", *Journal of Scheduling,* **4**, pp. 157-174 (2001).
9. Fiala, T. "An algorithm for the open-shop problem", *Mathematics of Operations Research,* **8**(1), pp.100-109 (1983).
10. Adiri, I., and Aizikowitz, N. "Open shop scheduling problems with dominated machines", *Naval Research Logistics Quarterly,* **36**, pp. 273-281 (1989).
11. Strusevich, V.A. "Dominating machines in deterministic systems", In: Applications of Mathematical Methods and Computer Engineering for Solving Problems of National Economy, *Proceedings of Republican Meeting, Minsk*, pp. 151-152 (1986).
12. Rock, H., and Schmidt, G. "Machine aggregation heuristics in shop-scheduling", *Methods of Operations Research,* **45**, pp. 303-314 (1983).
13. Brasel, H.; Tautenhahn, T., and Werner, F. "Constructive heuristic algorithms for the open shop problem", *Computing*, **51**, pp. 95-110 (1993).
14. Gueret, C., and Prins, C. "Classical and new heuristics for the open-shop problem: a computational evaluation", *Europian Journal of Operatioal Research,* **107**, pp. 306-314 (1997).
15. Taillard, E. "Benchmarks for basic scheduling problems", *European Journal of Operation Research,* **6**, pp. 278-283 (1993).
16. Liaw, C.F. "An iterative improvement approach for the non preemptive open shop scheduling problem", *European Journal of Operation Research,* **111**, pp. 509-517 (1998).
17. Liaw, C.F. "A hybrid genetic algorithm for the open shop scheduling problem", *European Journal of Operation Research,* **124**, pp. 28-42 (2000).
18. Schurman, P., and Woeginger, G.J. "Approximation algorithms for the multiprocessor open shop scheduling problem", Memorandum COSOR 97-23, Eindhoven University of Technology (1997).
19. Blum, C. "Beam-ACO- hybridizing ant colony optimization with beam search: an application to open shop scheduling", *Computers and Operations Research,* **32**, pp.1565-1591 (2005).