زمان‌بندی تک‌ماشین با زمان‌های آماده‌سازی وابسته به توالی با استفاده از الگوریتم *MACO*

محمد راشکی قلعه‌نو [i]؛ سید مهدی تشکری هاشمی [ii]

چکیده

هدف از حل مسألهٔ زمان‌بندی تک‌ماشین، ارایه برنامه زمانی برای ترتیب انجام کارهاست. در مدل مورد نظر، زمان آماده‌سازی، وابسته به توالی و هدف می‌نیمم کردن مجموع دیرکردهاست. این مسأله را قبلاً رابین و همکاران (۱۹۹۵)، تان و همکاران (۲۰۰۰) و گراول و همکاران (۲۰۰۲) بررسی کرده‌اند.

در این مقاله با اصلاح و بهبود قاعده انتقال و بهنگام‌سازی محلی فرمون یالها، در الگوریتم ابتکاری بهینه‌سازی اجتماع مورچه‌ها (ACO)، (گراول و همکاران، ۲۰۰۲) و استفاده از بهبود دهنده محلی، توانایی و کارایی این الگوریتم را افزایش می‌دهیم و ویرایش جدیدی از آن به نام الگوریتم ابتکاری اصلاح شده بهینه‌سازی اجتماع مورچه‌ها (MACO) را ارایه می‌کنیم. نتایج عددی به‌دست آمده نشان می‌دهد این الگوریتم جواب‌های بهتری برای مسأله مورد نظر نسبت به دیگر روش‌های ابتکاری موجود ارایه می‌کند.

کلمات کلیدی

زمان‌بندی، الگوریتم ابتکاری، بهینه‌سازی اجتماع مورچه‌ها، تک‌ماشین، مجموع دیرکردها و زمان آماده‌سازی وابسته به توالی.

Single Machine Scheduling with Sequence Dependent Setup Times using Modified Ant Colony Optimization

M. Rashki Ghaleh no; S. Mehdi Hashemi

ABSTRACT

Single machine scheduling problem presents a time schedule for job processing sequence. In the operating situation models, setup times are sequence dependent and the objective is to minimize the total tardiness. Rubin et al. (1995), Tan et al. (2000) and Gravel et al. (2002) have presented solutions for this problem.

In this paper, by modifying and improving transition rule, local update of pheromone trail in Ant Colony Optimization (ACO) heuristic algorithm [Gravel et al. (2002)] and applying local improvement method, the abilities and proficiency of this algorithm is increased. We call this new version Modified Ant Colony Optimization (MACO) heuristic algorithm. By implementing MACO algorithm, the numerical results show that the method obtains better solutions in comparison to the other heuristic methods.

KEYWORDS

Scheduling; Heuristic algorithm; Ant Colony Optimization; Single machine; Total tardiness; Sequence dependent setup time.

[i] کارشناس‌ارشد ریاضی‌کاربردی، دانشکده ریاضی، دانشگاه سیستان وبلوچستان، زاهدان- ایران تلفن ۳۲۵۰۰۹۵(۰۵۴۱): rashki_gh@yahoo.com

[ii] دانشیار دانشکده ریاضی و علوم کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران - ایران؛ تلفن ۶۴۵۴۲۵۲۲ (۰۲۱): hashemi@aut.ac.ir

۱- مقدمه

زمان‌بندی، تخصیص منابع یا ماشین‌آلات در طول زمان، برای انجام کارهاست. مشتری‌ها نه تنها به محصولات با حجم و اندازه کوچک‌تر نیاز دارند؛ بلکه باید مدت زمان تحویل در حد امکان کوتاه‌تر شود. از طرفی تولیدکنندگان قطعات استاندارد صنعتی که در گذشته زمان آماده‌سازی وابسته به توالی را در محاسبات خود در نظر نمی‌گرفتند، اکنون در کارهای عملی خود تجدید نظر کرده‌اند. بنابراین در عمل، تحقیقات برای بهبود روش‌های موجود و ارایه روش‌های جدید، از ارزش زیادی برخوردار است.

در سال ۲۰۰۲، گراول و همکاران [٦] نشان دادند که با بکارگیری اطلاعات نگاه به جلو در الگوریتم ACO، جواب‌های به‌دست آمده با این روش در مقایسه با الگوریتم‌های بهبود دهنده تعویض جفتی تصادفی(RSPI)[۱]، ژنتیک(GA)[۲] [۱۳] سردشدن‌تدریجی(SA)[۳] [١٤] شاخه‌وکران(B&B)[٤] و الگوریتم ACO معمولی برای مجموعه مسائل آزمون رابین و همکاران [۱۳] از کیفیت بهتری برخوردار است. آنان نشان دادند که با بزرگ شدن تعداد کارها، جواب‌های به‌دست آمده با الگوریتم ACO نسبت به الگوریتم RSPI (بهترین الگوریتم نسبت به B&B ، SA و GA) بهتر می‌شود؛ اما الگوریتم ACO نسبت به مشخصه های مسائل حساسیت دارد و RSPI برای مسائل کوچک‌تر نسبت به آن بهتر عمل می‌کند. برای رفع این نقیصه، در این مقاله با اصلاح الگوریتم ACO، الگوریتمی تحت عنوان الگوریتم اصلاح شده اجتماع مورچه‌ها (MACO)[٥] را ارایه و با پیاده‌سازی آن، نتایج عددی به‌دست آمده را با نتایج تان و همکاران [۱٥] و گراول و همکاران [۱۳] مقایسه می‌کنیم.

۲- مسأله زمان‌بندی تک‌ماشین

تحویل بموقع کار به مسأله‌ای اساسی در زمینه توالی عملیات با هدف می‌نیمم کردن مجموع دیرکردها[٦] منجر می‌شود.

مشکل پرداختن به مجموع دیرکردها و نیز به اکثر معیارهای عملکرد دیگری بر اساس دیرکرد تعریف می‌شوند، از این نکته سرچشمه می‌گیرد که دیرکرد، تابع خطی از زمان تکمیل کارها نیست. در مسائل واقعی، زمان آماده‌سازی[۷] ماشین به "کار تازه تکمیل شده" و "کار در دست انجام" بستگی دارد. با اضافه شدن شرط "زمان‌های آماده‌سازی وابسته به توالی"[۸] به مسأله می‌نیمم کردن مجموع دیرکردها، ابعاد مسأله به شدت افزایش می‌یابد؛ به همین علت، محققین به بررسی اینگونه مسائل تمایل کمتری دارند. در سال ۱۹۹۰، دو و همکاران [۳] ثابت کردند مسأله زمان‌بندی تک‌ماشین با هدف می‌نیمم کردن مجموع دیرکردها، یک مسأله NP- سخت[۹] است.

با توجه به اینکه الگوریتم‌های بهینه[۱۰] برای حل اینگونه مسائل، ناکارا هستند، گروه دیگری از الگوریتم‌ها به نام الگوریتم‌های ابتکاری[۱۱] ارایه شدند که بیشتر این روش‌ها، سریع و جواب زیر بهین (در زمان معقول جواب قابل قبول) تولید می‌کنند. الگوریتم‌های ابتکاری برای حل این مسأله به سه دسته روش‌های ساختاری[۱۲]، روش‌های جستجوی محلی[۱۳] و الگوریتم‌های تجزیه[۱٤] تقسیم می‌شوند.

مسأله می‌نیمم کردن مجموع دیرکردها روی تک ماشین با زمان آماده‌سازی وابسته به توالی را می‌توان به این صورت تعریف کرد که n کار مستقل تک مرحله‌ای در زمان صفر در دسترس و به‌صورت پیوسته قابل پردازش به‌وسیلهٔ تک ماشین هستند. ماشین در هر زمان تنها یک کار را پردازش می‌کند، قطع پردازش روی یک کار مجاز نیست و زمان آماده‌سازی ماشین به نوع کار تازه تکمیل شده و کار جدید بستگی دارد.

فرض کنید (p_j) زمان لازم برای پردازش و (d_j) موعدتحویل[۱٥] کار j باشد؛ (s_{ij}) را مدت زمان لازم برای آماده‌سازی ماشین برای انجام کار j بعد از تکمیل کار i در نظر می گیریم.

زمان تکمیل کار (C_j)، زمانی است که عملیات پردازش کار j به اتمام می‌رسد و با استفاده از (۱) محاسبه می‌شود:

$$C_j = \sum_{i=1}^{j} [s_{(i-1)(i)} + p_i] \qquad (۱)$$

تأخیر[۱٦] (L_j)، مدت زمانی است که تکمیل کار j نسبت به موعد تحویل آن بیشتر وقت می‌گیرد.

$$L_j = C_j - d_j \qquad (۲)$$

دیرکرد[۱۷] (T_j)، مدت زمانی است که تکمیل کار j نسبت به موعد تحویل آن تأخیر دارد و در غیر این صورت صفر است.

$$T_j = \underset{1 \le j \le n}{Max}\{0, L_j\} \qquad (۳)$$

اگر کار j بادرنگ بعد از کار i پردازش شود، فرجه آن با استفاده از (٤) محاسبه می‌شود:

$$m_{ij} = Max\{(d_j - p_j - s_{ij}), 0\} \qquad (٤)$$

اهمیت مساله می‌نیمم کردن مجموع دیرکردها با زمان‌های آماده‌سازی وابسته به توالی روی تک‌ماشین در مسائل صنعتی در زمان‌های مختلف به‌وسیلهٔ محققین مطرح شده است؛ برای مثال پینیدو [۱۰] در یک نمونه عملی، خط تولید بسته‌های کاغذ را بررسی کرده است که در آن، زمان آماده‌سازی برای

تعویض نوع بسته‌ها متفاوت بود. کارهای مشابه‌ای را نیز، داس و همکاران [۱]، فرنکا و همکاران [۵] در کارخانه پلاستیک و در سال ۲۰۰۲ گن وهمکاران [۷] در کارخانه ریخته گری آلومنیوم انجام دادند؛ مثال‌های واقعی دیگری نیز در صنایع منسوجات، مواد دارویی، مواد شیمیایی و متالوژی بررسی شده است.

۳- الگوریتم بهینه‌سازی اجتماع مورچه‌ها(ACO)

این الگوریتم را نخستین بار در سال ۱۹۹۲ دوریگو [۲] برای حل مسائل ترکیبیاتی معرفی کرد.

در این الگوریتم از اجتماع مورچه‌ها برای ساخت جواب استفاده می‌شود؛ زیرا در یک بار اجرای الگوریتم، تعداد زیادی جواب استخراج و تورهای متنوعی به‌دست خواهد آمد و فضای بیشتری از جواب‌های شدنی جستجو می‌شود.

در طبیعت، فرمون[۱۸] ماده‌ای است که مورچه‌های واقعی در حین حرکت آن را در مسیر خود می‌ریزند و مسیر حرکت خود را مشخص می‌کنند، اگر دو مسیر متفاوت از لانه تا غذا را درنظر بگیریم، در زمان یکسان، مورچه‌ای که در مسیر کوتاه‌تر قرار دارد، تعداد دفعات بیشتری را رفت و آمد می‌کند؛ بنابراین غلظت فرمون در این مسیر بیشتر است؛ بدین ترتیب مورچه‌ها تمایل بیشتری برای حرکت در مسیر کوتاه‌تر دارند. تبخیر فرمون، از همگرایی نارس الگوریتم به جواب‌های زیر بهین و نامرغوب جلوگیری می‌کند و باعث می‌شود که الگوریتم، نواحی جدیدتری از جواب را کاوش کند. در الگوریتم ACO، ضریب تبخیر فرمون با $(0 < \rho < 1)$ نشان داده می‌شود.

اطلاعات ابتکاری در این الگوریتم، برای هر مسأله دلخواه، از روی داده‌های آن مسأله و اغلب از روی هزینه‌های تابع هدف یا تخمینی از آن محاسبه می‌شود و این اطلاعات به طور مستقیم روی ساختن جواب تأثیر گذار هستند.

در این الگوریتم، مورچه‌ها (مصنوعی) از طریق گراف ساخته شده و با استفاده از تدبیر تصمیم‌گیری تصادفی و سود بردن از اطلاعات فرمون و اطلاعات ابتکاری، برای مسأله مورد نظر به صورت افزایشی به تولید جواب اقدام می‌کنند و در حین ساختن جواب و یا در پایان هر تور مقدار فرمون را به‌روز می‌کنند. این عمل باعث می‌شود که مورچه‌هایی که بعداً وارد مرحله جستجو می‌شوند، به مورچه‌های هوشمند تبدیل شوند که قادرند مسیرهای مرغوب‌تر را شناسایی کنند. فهرست ممنوع[۱۹] مربوط به یک مورچه خاص، به مجموعه‌ای از گره‌ها[۲۰] اطلاق می‌شود که مورچه مورد نظر در تور جاری قبلاً از آن گره دیدار کرده باشد و دیگر مجاز به انتخاب آن نباشد.

یکی از مهم‌ترین مراحل الگوریتم، مسأله فرمون‌ریزی و به‌روز کردن آن است. امکان دارد نحوه فرمون‌ریزی مورچه‌ها برای حل مسائل متفاوت و مختلف باشد و انتخاب یک روش خوب برای بروز کردن فرمون یالها، باعث به‌دست آوردن جواب‌های مرغوب خواهد شد. معمولاً مقدار اولیه فرمون روی یالها عدد کوچک و دلخواه (τ_0) انتخاب می‌شود.

٤- اطلاعات نگاه به جلو در الگوریتم ACO

این روش را در سال ۲۰۰۱ گراول و همکاران [٦] در کنفرانس الگوریتم‌های ابتکاری در کشور پرتقال مطرح و با پیاده‌سازی آن در سال ۲۰۰۲ [۸] نتایج خوبی ارایه کردند.

مهم‌ترین ویژگی این روش، معرفی ماتریس‌های چندگانه بود که هرکدام با توجه به تابع هدف ساخته می‌شوند؛ ماتریس کران پایین دیرکردها در این روش معرفی شد، که ما در الگوریتم MACO از آن استفاده کرده‌ایم و بر اساس هدف می‌نیمم کردن مجموع دیرکردها به صورت زیر ساخته می‌شود.

اگر تمام کارها (گره‌های گراف) برابر N و مورچه Kام در زمان t (تعداد تکرارهای الگوریتم) در گره i باشد، آنگاه دنباله‌ای از گره‌ها وجود دارند که در فهرست ممنوع قرار گرفته‌اند (کارهایی که پردازش شده‌اند) و می‌توان آنها را به صورت $Q_i = \{Q(0), Q(1), ..., Q(i)\}$ در نظر گرفت؛ بنابراین $N-i$ گره دیگر وجود دارند که در فهرست ممنوع نیستند و می‌توان آنها را نیز بصورت $Q'_i = \{Q'(1), Q'(2), ..., Q'(N-i)\}$ درنظر گرفت.

فرض کنید j گره‌ای خارج از فهرست ممنوع و کاندیدای قرار گرفتن بعد از گره i باشد، با توجه به معلوم بودن دنباله کارهای Q_i، می‌توان مقدار واقعی دیرکرد آنها (T_{Q_i}) را محاسبه کرد؛ اما به‌خاطر اینکه ترتیب $N-i$ کار باقیمانده معلوم نیست، گراول و همکاران [۸] با استفاده از روش تان و همکاران [١٤]، کران پایینی برای مجموع دیرکردهای دنباله Q'_i معرفی کردند $(T_{Q'_i})$؛ بنابراین ماتریس با هدف می‌نیمم کردن مجموع دیرکردها به صورت $B_{ij} = T_{Q_i} + T_{Q'_i}$ محاسبه می‌شود.

۵- الگوریتم MACO در زمان‌بندی تک‌ماشین با زمان‌های آماده‌سازی وابسته به توالی

مجموعه مسائل رابین و همکاران [۱۳] سه مشخصه دارند: اختلاف زمان‌های پردازش[۲۱]، فاکتورهای تأخیر[۲۲] و بازه موعدهای تحویل[۲۳].

تان و همکاران [۱۵] با استفاده از این مسائل، الگوریتم‌های RSPI، GA، SA و B&B را برای حل مسأله می‌نیمم کردن مجموع دیرکردها روی تک ماشین با زمان آماده‌سازی وابسته

به توالی مقایسه کردند و نشان دادند که روش RSPI نسبت به روش‌های دیگر سازگاری بیشتری برای حل این مسأله دارد. به‌همین دلیل بود که گراول و همکاران [۸] الگوریتم ACO را با الگوریتم RSPI مقایسه کردند.

هرچند با بزرگ شدن تعداد کارها، الگوریتم ACO در تولید جواب‌ها، بهتر از الگوریتم RSPI عمل می‌کند، اما همان‌گونه که از شکل‌های (۲)، (۳) و (۴) مشخص است، برای مسائل ۱۵ و ۲۵ کار، الگوریتم RSPI بهتر از الگوریتم ACO عمل می‌کند و همچنین هر دو روش نسبت به مشخصه‌های مسائل حساسیت زیادی دارند. برای بهبود جواب‌های الگوریتم ACO، این الگوریتم را به صورت زیر اصلاح و پیاده‌سازی کرده‌ایم.

۵-۱- بهنگام‌سازی محلی فرمون یالها

در روش گراول و همکاران [۱۳] وقتی یک مورچه در زمان t (تعداد اجراهای الگوریتم) از یال (i,j) عبور می‌کند، فرمون روی یال طبق (۵) محاسبه می‌شود.

$$\tau_{ij}(t) = \rho \cdot \tau_{ij}(t) + (1-\rho)\cdot \tau_0 \qquad (۵)$$

در این رابطه، مقدار $\rho \cdot \tau_{ij}(t)$ تبخیر فرمون و $(1-\rho)\cdot\tau_0$ مقدار فرمون ریخته شده روی یال (i,j) به‌وسیلهٔ مورچه را مشخص می‌کند.

اشکال روش فوق این است که فرمون‌ریزی به صورت کور و به مقدار ثابت $(1-\rho)\cdot\tau_0$ صورت می‌گیرد و از اطلاعات مسأله استفاده نمی‌شود.

در الگوریتم MACO برای جلوگیری از همگرایی نارس، علاوه بر فرمون‌ریزی سراسری به‌وسیلهٔ مورچه نخبه در پایان هر دور اجرای الگوریتم (بهترین جواب)، برای رفع اشکال بالا از فرجه کارها برای فرمون‌ریزی محلی استفاده کرده‌ایم.

در این روش وقتی مورچه از یال (i,j) در زمان t عبور می‌کند، مقدار فرمون ریخته شده روی یال با استفاده از (۶) محاسبه می‌شود:

$$\tau_{ij}(t) = \rho \cdot \tau_{ij}(t) + \frac{1}{m_{ij}+10}\cdot\tau_0 \qquad (۶)$$

بنابراین هر چه مقدار فرجه کار j کمتر باشد، مقدار فرمون بیشتری روی یال (i,j) ریخته می‌شود و مورچه تمایل بیشتری برای انتخاب یال دارد و بالعکس. این کار باعث جستجو در جواب‌های با کیفیت‌تر می‌شود.

۵-۲- قاعده انتقال

در الگوریتم‌های ACO برای حل مسائل، در قاعده انتقال[۲۴]، (انتخاب گره j بعد از گره i) دو نکته کاوش در جواب‌های شدنی و ناشناخته و بهره‌برداری از اطلاعات به‌دست آمده برای یافتن جواب‌های بهتر مورد توجه قرار گرفته است.

در قاعده انتقال الگوریتم MACO علاوه بر استفاده از ماتریس زمان‌آماده‌سازی (s_{ij})[۲۵] و ماتریس فرجه (m_{ij})[۲۶] از اطلاعات نگاه‌به‌جلو (B_{ij}) استفاده کرده‌ایم. هرچند استفاده اطلاعات نگاه به جلو بار محاسباتی الگوریتم را بالا می‌برد؛ اما با توجه به استفاده از زمان‌های موعد تحویل در قاعده انتقال و همگرایی سریع به جواب‌های خوب، این زمان محاسباتی زیاد نخواهد بود.

فرض کنید در زمان t مورچه k در گره i باشد. احتمال انتخاب گره j (خارج از فهرست ممنوع) به‌وسیلهٔ این مورچه با استفاده از (۷) محاسبه می‌شود.

$$p_{ij}^{\,k}(t) = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}\cdot[\frac{1}{s_{ij}+0.9}]^{\beta}[\frac{1}{m_{ij}+0.1}]^{\delta}}{\sum_{l\notin Tabu_k}[\tau_{il}]^{\alpha}\cdot[\frac{1}{s_{il}+0.9}]^{\beta}[\frac{1}{m_{il}+0.1}]^{\delta}} \\ \qquad \dfrac{[\frac{1}{B_{ij}+0.9}]^{\phi}}{[\frac{1}{B_{il}+0.9}]^{\phi}} \quad if\ j \notin tabu_k \\ 0 \qquad\qquad Otherwise \end{cases} \qquad (۷)$$

در (۷) پارامترهای α,β,δ,ϕ نسبت اهمیت مقدار فرمون، زمان‌آماده‌سازی، فرجه و اطلاعات نگاه به جلو را تعیین می‌کند.

تنظیم این پارامترها برای ایجاد تعادل بین اطلاعات استفاده شده در قاعده انتقال بسیار مهم است و تأثیر زیادی در تولید جواب‌های خوب دارد. در این روش، مقادیر را به صورت تجربی بین ۰/۱ و۴ انتخاب و برای انتخاب گره j از قاعده انتقال زیر استفاده کرده‌ایم.

$$j = \begin{cases} \arg\ \underset{l\notin Tabu_k}{Max}\{[\tau_{il}]^{\alpha}\cdot[\frac{1}{s_{il}+0.9}]^{\beta}[\frac{1}{m_{il}+0.1}]^{\delta} \\ \qquad [\frac{1}{B_{il}+0.9}]^{\phi}\} \quad if \quad q \le q_0 \\ J \qquad\qquad Otherwise \end{cases} \qquad (۸)$$

در (۸)، J مطابق با (۷) انتخاب و q یک عدد تصادفی از بازه $[0,1]$ و پارامتر $0 < q_0 < 1$ نسبت اهمیت جستجوی جواب‌های جدید به استخراج جواب را تعیین می‌کند.

شکل (۱) : الگوریتم MACO برای مسأله زمان‌بندی روی تک ماشین با زمان‌های آماده‌سازی وابسته به توالی، با هدف می‌نیمم کردن مجموع دیرکردها.

/*STEP 1: PHEROMONE INITIALIZA TION */
For each pair of jobs (i, j) do
Set an initial value $\tau_{ij}(0) = \tau_0$
End For
Let T_+ be the best job sequence found so far and let L_+ be its total tardiness

/*STEP 2: MAIN LOOP */
For t = 1 to t_{Max} do
/*STEP 2.1: SET INITIAL JOB */
For k = 1 to m do
Set the initial job for ant k
Store this informatio n in $Tabu_k$
End For
/*STEP 2.2: BUILD A JOB SEQUENCE FOR EACH ANT */
Let T_* be the best job sequence found in the current cycle and let L_* be its total tardiness
For i = 1 to n do
For k = 1 to m do
Choose the next job j, $j \notin Tabu_k$, among the cl candidate jobs according to :

$$j = \begin{cases} \arg\max_{l \in Tabu_k} \left\{ [\tau_{il}(t)]^{\alpha} \cdot [\frac{1}{s_{il}+0.9}]^{\beta} \cdot [\frac{1}{m_{il}+0.1}]^{\delta} \cdot [\frac{1}{B_{il}+0.9}]^{\phi} \right\} & if \quad q \le q_0 \\ J & if \quad q > q_0 \end{cases}$$

Where J is chosen according to the probabilit y :

$$p^k{}_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\frac{1}{s_{ij}+0.9}]^{\beta} \cdot [\frac{1}{m_{ij}+0.1}]^{\delta} \cdot [\frac{1}{B_{ij}+0.9}]^{\phi}}{\sum_{l \in Tabu_k} [\tau_{il}(t)]^{\alpha} \cdot [\frac{1}{s_{il}+0.9}]^{\beta} \cdot [\frac{1}{m_{il}+0.1}]^{\delta} \cdot [\frac{1}{B_{il}+0.9}]^{\phi}}$$

Store this informatio n in $Tabu_k$
Local update of pheromone trail for chosen job pair (i, j) :

$$\tau_{ij}(t) = \rho_l \cdot \tau_{ij}(t) + \frac{1}{m_{ij}+10} \cdot \Delta\tau_{ij} \quad where \quad \Delta\tau_{ij} = \tau_0$$

End For
End For
/*STEP 2.3: EVALUATION OF SOLUTIONS AND IMPROVEMEN T METHOD */
For k = 1 to m do
Compute the total tardiness $L_k(t)$ for the job sequence $T_k(t)$ produced by ant k
Apply local improvemen t (RSPI) method for the sequence $T_k(t)$ and recomputed L_k
If an improved sequence is found then update T_+ and L_+
End For
/*STEP 2.4: GLOBAL UPDATE OF PHEROMONE TRAIL */
For each adjacent job pair $(i, j) \in T_*$ do
U pdate the pheromone trail according to :

$$\tau_{ij}(t+1) = \rho_g \cdot \tau_{ij}(t) + (1-\rho_g) \cdot \Delta\tau_{ij}(t) \quad where \quad \Delta\tau_{ij}(t) = 1/L_*$$

End For
Empty all $tabu_k$
End For
Print the best job sequence T_+ and its total tardiness L_+

ملاحظه می‌شود که احتمال انتخاب گره‌ای که دارای فرمون بیشتر و زمان آماده‌سازی، فرجه و اطلاعات نگاه‌به‌جلوی کمتر باشد، از سایر گره‌ها بیشتر است؛ از طرفی، در مسائل واقعی صنعتی کارهایی وجود دارند که فرجه صفر یا زمان آماده‌سازی صفر دارند؛ بنابراین برای تشویق مورچه‌ها در قاعده انتقال برای انتخاب این کارها، به صورت تجربی مقادیر ثابتی را با توجه به اهمیت عناصر ماتریس‌های استفاده شده، با آنها جمع کرده‌ایم.

۵-۳- استفاده از بهبود دهنده محلی

یکی دیگر از خصوصیات خوب الگوریتم ACO، قدرت ترکیب آن با الگوریتم‌های جستجوی محلی است؛ همان گونه که قبلاً بیان شد، روش بهبود دهنده محلی RSPI نسبت به روش‌های بهبود دهنده دیگر، برای حل مسأله مورد بحث، سازگاری بیشتری دارد. در الگوریتم MACO از الگوریتم بهبود دهنده RSPI برای بهبود جواب‌های تولید شده استفاده کرده‌ایم.

در این روش، بعد از ساختن جواب به‌وسیلهٔ هر مورچه، یالهای آن به صورت تصادفی تعویض و جواب‌های حاصل نسبت به هم سنجیده می‌شوند. همان گونه که در نتایج ارایه شده در قسمت بعد مشخص است در نتیجه استفاده از این روش در الگوریتم MACO، جواب‌های تولید شده، کیفیت بسیار خوبی نسبت به روش های ACO [۸] و RSPI [۱۳] دارند.

۵-٤- نتایج و مقایسه با روش‌های موجود دیگر

برای پیاده‌سازی الگوریتم MACO، از زبان برنامه‌نویسی *C* استفاده و هر مسأله از مجموعه مسائل آزمون رابین و همکاران [۱٦] را ۲۰ بار اجرا کرده‌ایم.

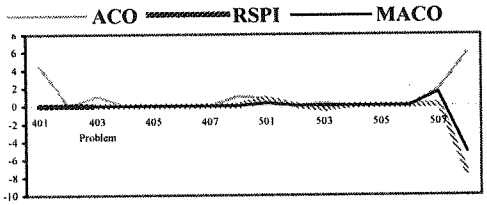
نتایج حاصل بر حسب درصد اختلاف بدترین، میانگین و بهترین نتایج به‌دست آمده نسبت به روش B&B در جدول (۱) و شکل‌های (۲)، (۳) و (۴) ارایه شده است.

نتایج ارایه شده در جدول (۱) نشان می‌دهد، در مسائل ٤۰۱، ٤۰۷ و ٤۰۸ روش MACO دقیق (واریانس برابر صفر) و نسبت به روش‌های ACO و RSPI بسیار بهتر عمل می‌کند.

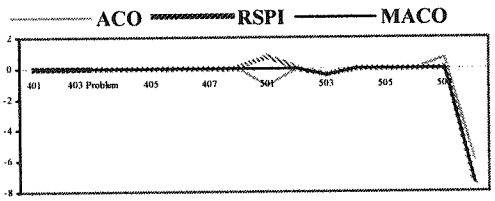
در مسأله ٤۰۳ بجز ۱ مورد از ۲۰ اجرای الگوریتم؛ که مقدار ۳٤۲٦ به‌دست آمد، در ۱۹ مورد جواب دقیق حاصل شده است.

در مسأله ٤۰٤ روش MACO بهتر از روش RSPI و همانند ACO دقیق عمل می‌کند و در بقیه مسائل این گروه، عملکرد هر سه روش یکسان است.

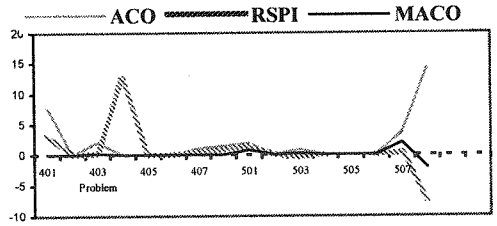
در مسائل ۲۵ کار بجز بهترین جواب مسأله ۵۰۱، الگوریتم MACO از الگوریتم ACO بهتر عمل می‌کند و همان‌گونه که ملاحظه می‌شود اختلاف بهترین جواب و بدترین جواب نیز نسبت به الگوریتم ACO، کمتر شده است. این موضوع در مسأله ۵۰۸ بیشتر به چشم می‌خورد. همان‌طور که از شکل‌های (۲)، (۳) و (۴) مشاهده می‌شود، روش MACO نسبت به دو روش دیگر، جواب‌های با کیفیت‌تر و به مشخصه‌های مسائل حساسیت کمتری دارد.



شکل (۲): درصد اختلاف میانگین نتایج الگوریتم‌های MACO، ACO و RSPI بر حسب نتایج B&B



شکل (۳): درصد اختلاف بهترین نتایج الگوریتم‌های MACO، ACO و RSPI بر حسب نتایج B&B



شکل (۴): درصد اختلاف بدترین نتایج الگوریتم‌های MACO ، ACO و RSPI بر حسب نتایج B&B

۶- نتیجه‌گیری

مسأله زمان‌بندی تک‌ماشین با زمان‌های آماده‌سازی وابسته به توالی با تابع هدف می‌نیمم کردن مجموع دیرکردها یک مسأله NP- سخت است.

الگوریتم ACO که گراول و همکاران [۸] ارایه کرده‌اند، قادر است به‌طور مؤثری با کاهش زمان‌های محاسباتی این مسائل را که کاربرد گسترده‌ای در صنعت دارند، حل کند؛ اما این مزایا برای مسائل کوچک‌تر، کمتر است و همچنین روش فوق نسبت به مشخصه‌های مسائل حساسیت زیادی دارد.

الگوریتم پیشنهادی MACO را که با اصلاح نقاط حساس

الگوریتم ACO ارایه کردیم، قادر است ، کارایی آن را گسترش دهد و با پیاده‌سازی آن نشان دادیم که نسبت به سایر روش‌های ابتکاری موجود برای مسأله مورد نظر بهتر عمل می‌کند.

جدول (۱) : مقایسه الگوریتم اصلاح شده اجتماع مورچه‌ها ، با الگوریتم‌های شاخه‌وکران ، بهبود دهنده تعویض جفتی تصادفی و اجتماع مورچه‌ها (پیاده‌سازی با زبان برنامه‌نویسی C)

مسأله	تعداد کارها	PTV	TF	DDR	B&B	بهبود دهنده تعویض جفتی تصادفی (RSPI)			الگوریتم اجتماع مورچه‌ها (ACO)			الگوریتم اصلاح شده اجتماع مورچه‌ها (MACO)			†††
						†			††			†††			
						درصد اختلاف جواب‌ها نسبت به شاخه‌وکران			درصد اختلاف جواب‌ها نسبت به شاخه‌وکران			درصداختلاف جواب‌ها نسبت به شاخه‌وکران			
						زمان‌های اجرا (ثانیه)			زمان‌های اجرا (ثانیه)			زمان‌های اجرا (ثانیه)			
						میانگین	بدترین	بهترین	میانگین	بدترین	بهترین	میانگین	بدترین	بهترین	میانگین
						جوابها	جواب	جواب	جوابها	جواب	جواب	جوابها	جواب	جواب	زمان‌های اجرا (ثانیه)
۴۰۱	۱۵	L	L	N	*۹۰	۲/۳	۰/۰	۰/۰	۴/۴	۷/۸	۰/۰	۰/۰	۰/۰	۰/۰	۰/۶۲
۴۰۲	۱۵	L	L	W	* .	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰
۴۰۳	۱۵	L	M	N	*۳۴۱۸	۰/۲	۰/۰	۰/۰	۱/۱	۲/۱	۰/۰	۰/۰	۰/۰	۰/۰	۰/۲
۴۰۴	۱۵	L	M	W	*۱۰۶۷	۱۲/۷	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰
۴۰۵	۱۵	H	L	N	* .	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	‡ .	۰/۰	۰/۰	‡ ۰/۰
۴۰۶	۱۵	H	L	W	* .	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	‡ .	۰/۰	۰/۰	‡ ۰/۰
۴۰۷	۱۵	H	M	N	*۱۸۶۱	۰/۴	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۱/۴۵	۰/۰	۰/۰	۰/۲۶
۴۰۸	۱۵	H	M	W	*۵۶۶۰	۰/۹	۰/۰	۰/۰	۱/۱	۱/۵	۰/۰	۱/۴۵	۰/۰	۰/۰	۰/۴۵
۵۰۱	۲۵	L	L	N	۲۶۴	۱/۵	۰/۸	۰/۸	۱/۹	۱/۹	۰/۸	۷/۱۵	۰/۰	۰/۳	۰/۸
۵۰۲	۲۵	L	L	W	* .	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۱۵	۰/۰	۰/۰	۰/۰
۵۰۳	۲۵	L	M	N	۳۵۱۱	-۰/۴	-۰/۴	-۰/۴	۰/۳	۰/۹	-۰/۴	۷/۸۰	-۰/۴	۰/۰	۰/۲
۵۰۴	۲۵	L	M	W	* .	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۲۰	۰/۰	۰/۰	۰/۱۱
۵۰۵	۲۵	H	L	N	* .	۰/۰	۰/۰	# _	۰/۰	۰/۰	۰/۰	۰/۱۰	۰/۰	۰/۰	۰/۱۱
۵۰۶	۲۵	H	L	W	* .	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۰	۰/۱۰	۰/۰	۰/۰	۰/۱۱
۵۰۷	۲۵	H	M	N	۷۲۲۵	۰/۰	۰/۱	۰/۲	۱/۸	۲/۷	۰/۰	۹/۸۰	۰/۰	۱/۵	۲/۰
۵۰۸	۲۵	H	M	W	۲۰۶۷	-۷/۴	-۷/۴	-۷/۴	۵/۹	۱۴/۲	-۵/۹	۸/۵۵	-۷/۴	-۵/۱	-۲/۰

* مقدار دقیق جواب. # تقسیم بر صفر و بدترین جواب ۶/۰ است. ‡ زمان هر اجرای الگوریتم (از ۲۰ اجرا) کمتر از یک ثانیه است.

DDR : بازه موعدهای تحویل.TF : فاکتورهای تأخیر. PTV : اختلاف زمان‌های پردازش. L : کم. H : زیاد. M : متوسط. N : کم‌بهنا. W : پهن.

† کامپیوتر شخصی با پردازشگر پنتیوم ۹۰ مگاهرتز. †† کامپیوتر شخصی با پردازشگر پنتیوم ۷۳۳ مگاهرتز. ††† کامپیوتر شخصی با پردازشگر پنتیوم ۱۸۰۰ مگاهرتز.

۷- مراجع

[۱] Das S.R., Gupta J.N.D., Khumawala B.M, (1995). A saving index heuristic algorithm for flowshop scheduling with sequence dependent set-up times. The Journal of the Operational Research Society, 46, 1365-1373

[۲] Dorigo M. (1992). Optimization, learning and natural algorithms, Ph.D. Thesis, Polytechnic Milan, Italy.

[۳] Du, J., Leung, J.Y., (1990). Minimizing total tardiness on one machine is NP-hard. Mathematics of Operations Research.

[۴] Emmons H., (1969). One-machine sequencing to minimize certain functions of job tardiness. Operations Research, 17, 701-715.

[۵] Franca P.M., Gendreau M., Laporte G., Muller F.M., (1996). A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times. International Journal of Production Economics, 43, 79-89 .

[۶] Gravel, M., Price, W. L. (2001). A Look-Ahead Addition to the Ant Colony Optimization Metaheuristic and its Application to an Industrial Scheduling Problem. 4th Metaheuristic International Conference. Porto, Portugal 79-84.

[۷] Gagne C.; Gravel M.; Price W., (2002); Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. European Journal of Operational Research, 143, 218–229.

[۸] Gravel M., Gagne C., Price W., (2002). Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times. Journal of Operational Research Society, 53, 895-906.

[۹] Lawler E.L., (1977). A "pseudo polynomial" algorithm for sequencing jobs to minimize total tardiness. Annals of Discrete Mathematics, 1, 331-342.

[۱۰] Pinedo, M., (1995). Scheduling theory, algorithms and systems. Prentice-Hall, NJ.

[۱۱] Potts C.N., Van Wassenhove L.N., (1982). Decomposition algorithm for the single machine total tardiness problem, Operations Research Letters, 1, 177-181.

[۱۲] Ragatz G.L., (1993). A branch-and-bound method for minimum tardiness sequencing on a single processor with sequence dependent setup times. Proceedings: Twenty-fourth Annual Meeting of the Decision Sciences Institute, 1375-1377 .

[۱۳] Rubin P.A., Ragatz G.L. (1995). *Scheduling in a sequence dependent setup environment with genetic search.* Computers and Operations Research, 22, 2, 85-99.

[۱۴] Tan K.C., Narasimhan R., (1997). *Minimizing tardiness on a single processor with sequence dependent setup time.* a simulated annealing approach, Omega, 25, 6, 619-634.

[۱۵] Tan K.C., Narasimhan R., Rubin PA., Ragatz G.L., (2000). *A comparison of four methods for minimizing total tardiness on a single processor with sequence dependent setup times.* Omega, 28, 313-326.

[۱۶] *http://mgt.bus.msu.edu/datafiles.htm*

۸- زیرنویس ها

[1] Random Search Pair wise Interchange
[2] Genetic algorithm
[3] Simulated Annealing
[4] Branch and Bound
[5] Modified ant colony optimization
[6] Total tardiness
[7] Setup time
[8] Sequence dependent setup times
[9] NP-Hard
[10] Optimal algorithms
[11] Heuristic algorithms
[12] Construction
[13] Local search methods
[14] Decomposition heuristics
[15] Due date
[16] Lateness
[17] Tardiness
[18] Pheromone
[19] Tabu list
[20] Node
[21] Processing time variance
[22] Tardiness factor
[23] Due date range
[24] Transition rule
[25] Setup times matrix
[26] Slack time matrix