

مطالعه مقایسه‌ای برای معیار دیرکرد یک مسئله تک ماشین

محمد تقی فاطمی قمی

استادیار دانشکده مهندسی صنایع دانشگاه صنعتی امیرکبیر

مرجان رضائی

دانشجوی کارشناسی ارشد مهندسی صنایع دانشگاه صنعتی شریف

اعظم ایلکی

فارغ التحصیل کارشناسی تولید صنعتی دانشکده مهندسی صنایع دانشگاه امیرکبیر

چکیده:

در این مقاله به بررسی روشهای برنامه‌ریزی یک مجموعه کار بر روی یک ماشین می‌پردازیم که هدف از آن، حداقل کردن معیار دیرکرد است. جهت حل مسئله فوق دودسته روش وجود دارد:
- روشهای بهینه‌ساز
- روشهای ابتکاری
گروه اول شامل برنامه‌ریزی پویا، روش هیبرید و روش شاخه و حد است و گروه دوم، دربرگیرنده روشهایی چون الگوریتم ویلکرسون - ایروین، فرای و همکاران، کلاین - وتورا و روش نمونه‌گیری تصادفی است. براساس برنامه‌های کامپیوتری توسعه داده شده و نتایج برگرفته از آنها نشان داده شده است که گروه دوم را می‌توان با قاطعیت بیشتر، به‌عنوان روشهای برتر معرفی نمود.

A Comparative Study for Mean Tardiness Criterion in a Single Machine Problem

M.T. Fatemi Ghomi, Ph.D.

Indust. Eng. Dept. Amirkabir Univ. of Tech.

M.Rezaei, Postgraduate student

Indust. Eng. Dept. Sharif Univ. of Tech.

A.Ilaki, B.Sc.

Indust. Eng. Dept. Amirkabir Univ. of Tech.

In this paper we examine the different methods of scheduling a given set of jobs on a single machine to minimize mean tardiness criterion. Two groups of methods have been developed to solve this problem: the optimizing methods and the heuristic methods. The first group consists of dynamic programming, hybrid algorithm, and branch & bound methods, and the second group consists of heuristics such as Wilkerson-Irwin, Fry et al, Klein-Ventura, and random sampling. Computational results based on computer programs show that the hybrid algorithm among the approaches considered in the first group, and Fry et al among the second group's procedures are the most efficient methods for mean tardiness criterion.

مقدمه :

با توجه به فرضیات مدل اصلی تک ماشین، معیار کارآئی T به شکل زیر تعریف می شود:

$$T = \sum_{i=1}^n \text{Max}(0, L_i) / N$$

که در آن، L_i مغایرت زمان تکمیل و موعد تحویل کار i می باشد $(C_i - d_i)$.

مسائل برنامه ریزی ترتیب انجام کارها گسترده و پیچیده اند. به طور کلی در مسئله ای با n کار، $n!$ طریق ممکن برای مرتب کردن آنها وجود دارد. مثلاً برای مسئله ای دارای ۴ کار، $(4!)$ یعنی ۲۴ ترتیب ممکن وجود دارد که از میان آنها باید بهترین را برگزید. ایجاد و بررسی این ۲۴ ترتیب چندان دشوار نخواهد بود، اما اگر بجای ۴ کار ۱۰ کار برای برنامه ریزی وجود داشته باشد، در آن صورت $(10!)$ یعنی ۳۶۲۸۸۰۰ حالت وجود خواهد داشت که اگر آنها را با کامپیوتری سریع بررسی کنیم که در هر ثانیه ۱۰۰۰ جواب را چک می کند، این کار به یک ساعت وقت نیاز خواهد داشت و اگر تنها یک کار به تعداد کارها افزوده شود، در آن صورت تعداد برنامه ها ۳۹۹۱۶۸۰۰ خواهد شد و کامپیوتر برای حل مسئله جدید به بیش از ۱۱ ساعت وقت نیاز خواهد داشت.

بنابراین روشهای گوناگونی برای انجام برنامه ریزی کارها به وجود آمده اند تا این محاسبات فراوان و پیچیده را کاهش و جوابهای مناسبی را به دست دهند. برخی از این روشها جواب بهینه (یعنی جوابی با حداقل دیرکرد ممکن) را مشخص می کنند و برخی دیگر که روشهای ابتکاری نامیده می شوند، ترتیبی را انتخاب می کنند که الزاماً بهینه نیست ولی حصول آن در مورد مسائل بزرگ، با صرف زمانی کمتر از روشهای گروه اول به دست می آید و به این جهت در موارد لازم به جواب بهینه

ترجیح داده می شود.

۱- روشهای بهینه ساز

۱-۱- روش برنامه ریزی پویا

این روش در سال ۱۹۵۷ توسط بلمن (Bellman) مطرح گردید و در حل بسیاری از مسائل بهینه سازی از جمله برنامه ریزی ترتیب انجام کارها به کار می رود و برای همه مسائلی که می توان آنها را به یک سری از مسائل زنجیروار تجزیه کرد، کاربرد دارد، به نحوی که جواب یک مرحله مستقیماً از مرحله پیشین بدست آید (۲ و ۱).

برنامه ریزی پویا، جزو روشهای منظورسازی (Enumeration Techniques) است بدین معنی که تمام برنامه های ممکن را بشمار آورده یا منظور می کند، سپس برنامه های غیربهینه را از میان آنها حذف کرده و جوابهای بهینه را نگه می دارد.

در سال ۱۹۶۲، هلد و کارپ، با استفاده از روش بلمن، ایده های برنامه ریزی پویا را در مسائل ترتیب انجام کارها بکار بردند. روش آنها در مورد حل مسائل تک ماشینی مصداق دارد که دارای معیار کارآئی جمع پذیری به شکل $Z = \sum g_i(C_i)$ باشد که در آن $g_i(C_i)$ تابعی افزایشی از زمانهای تکمیل است.

روش هلدوکارپ زمانی که معیار کارآئی C و \bar{F} است نیز قابل استفاده می باشد، لکن چون قاعده SPT براحتی آنها را حل می کند، لذا معقول نیست که این دو مسئله را با این روش حل نماییم. هلدوکارپ در روش خود از این مطلب سود جستند که در یک برنامه بهینه، k کار اول، ترتیب بهینه را برای حل مسئله ای شامل تنها همان k کار بوجود می آورد.

به طور خلاصه در روش برنامه ریزی پویا برای به دست آوردن ترتیب بهینه n کار، n مرحله وجود دارد که در مرحله یک با تشکیل کلیه زیرمجموعه های یک عضوی، مینیمم دیرکرد را به

دست آورده، و در مرحله ۲ با تشکیل کلیه زیرمجموعه‌های ۲ عضوی و مقایسه این مطلب که با قرار گرفتن کدام کار در ترتیب آخر با توجه به نتیجه حاصل از مجموعه‌های تک‌عضوی مرحله قبل، حداقل هزینه به دست می‌آید، و در مرحله m ام با تشکیل مجموعه n عضوی و مقایسه این مطلب که با قرار گرفتن کدام کار در ترتیب آخر این مجموعه، با توجه به نتیجه حاصل از مجموعه‌های $(n-1)$ عضوی مرحله قبل، حداقل دیرکرد به دست می‌آید، جواب بهینه با حرکت از آخرین مرحله به سمت اولین مرحله با یافتن آخرین ترتیب در هر مرحله حاصل می‌گردد. چنانچه در مسئله ترتیب انجام کارها، محدودیت‌های تقدم و تأخر نیز وارد شود، حل مسئله ساده‌تر خواهد شد، زیرا به دلیل همین محدودیتها، در هر مرحله برخی از مجموعه‌ها خودبه‌خود حذف شده و به این ترتیب محاسبات مربوطه کاهش می‌یابد. روش برنامه‌ریزی پویا بر طبق روش پیشنهادی بیکر به صورت کامپیوتری درآمده که در منابع اشاره شده در پایان مقاله، قابل دسترسی می‌باشد (۳).

۲-۱- روش هیبرید

روش هیبرید که توسط سرنینی‌واسان (Srinivasan) در سال ۱۹۷۱ ارائه شده است، از روش برنامه‌ریزی پویا نشأت گرفته و از دو خاصیت برجستگی برای مسئله دیرکرد که امونز ارائه کرده است، بهره می‌گیرد. با استفاده از این خواص در جایی که امکان پذیر باشد، روابط تقدم و تأخری بین کارها ایجاد گشته و کارهایی در اولین و آخرین محل‌های ترتیب بهینه مشخص می‌شوند. سپس کارهای باقیمانده در چارچوب برنامه‌ریزی پویا مرتب شده و به جواب نهائی که بهینه است دست می‌یابیم (۴). این روش با استفاده از سه مرحله به صورت ذیل، جواب بهینه را به دست می‌دهد:

مرحله اول: کارهایی که موعد تحویلشان از مجموع زمان انجام کارهای باقیمانده بیشتر باشد در آخرین نوبت‌ها قرار می‌گیرند. این روند تا زمانی که نتوان کاری با موعد تحویل بزرگتر از کارهای باقیمانده پیدا کرد ادامه می‌یابد.

مرحله دوم: کارهای باقیمانده از مرحله یک، به ترتیب صعودی مرتب شده و با بکارگیری قضایای ۱ و ۲ امونز و استفاده از ماتریس تقدمی M (ماتریس مربعی که مرتبه آن برابر کارهای باقیمانده از مرحله ۱ می‌باشد)، و رابطه تقدم و تأخری بین کارها بوجود می‌آید.

قضیه ۱ امونز:

برای هر دو کار i و j که $t_j < t_i$ باشد، اگر $d_j \leq \max(t_j + \sum t_k, d_j)$ (که در آن t_k مجموع زمان انجام کارهای قبل از j می‌باشد) در این صورت $m_{ij} = 1$.

قضیه ۲ امونز:

برای هر دو کار i و j که $t_j < t_i$ باشد، اگر $d_j > \max(t_j + \sum t_k, d_j)$ و $d_i + t_j \leq \sum t_k$ در این صورت $m_{ji} = 1$.

در ماتریس M تعداد ۱های هر ستون نشان دهنده تعداد کارهایی است که بایستی قبل از کار مربوط به آن ستون انجام گیرد. چنانچه ماتریس M یک $n \times n$ باشد باید ابتدا بدنبال کاری بگردیم که تعداد ۱های ستون آن برابر $n-1$ باشد و در مرحله دوم بدنبال ستون با ۱های برابر $n-2$ بگردیم. این کار تاجائی ادامه می‌یابد که نتوان کاری را به عنوان کار ماقبل آخر برای آخرین کار برنامه‌ریزی شده تخصیص داد.

مرحله سوم: برای کارهای برنامه‌ریزی نشده حاصل از مرحله ۲، روش برنامه‌ریزی پویا را بکار می‌بریم تا ترتیب بهینه برای آنها حاصل گردد.

بدین ترتیب، نتایج حاصل از مراحل ۱ تا ۳، ترتیب بهینه کلیه کارها را از آخر به اول ارائه خواهد داد. برای روش هیبرید بر طبق الگوریتم ارائه شده توسط سرنینی‌واسان، برنامه کامپیوتری بنام HYBRID تهیه گردیده، که علاقمندان می‌توانند به منبع اشاره شده در پایان مقاله مراجعه نمایند (۳).

۳-۱- روش شاخه و حد (Branch & Bound Method)

روش شاخه و حد یکی از معمولترین تکنیکهای برنامه‌ریزی است و به همراه روش برنامه‌ریزی پویا، جزو روشهای منظورسازی ضمنی (implicit enumeration) محسوب می‌شود، زیرا این روشها به جای در نظر گرفتن $n!$ حالت ممکن (که روشهای منظورسازی کامل یا صریح [Complete or explicit enumeration] چنین عمل می‌نمایند) تعدادی از ترکیبها را بررسی و تعدادی را در مراحل مختلف حل الگوریتم حذف می‌نمایند.

بررسی در روش شاخه و حد با ایجاد درخت زیرمجموعه‌های لازم شروع می‌شود، به طوری که در سطح اول هر کار به یک گره تخصیص می‌یابد، و در سطح بعدی از هر گره، کارهای باقیمانده منشعب می‌شوند و این تقسیم‌بندی ادامه می‌یابد تا جایی که با دنبال کردن شاخه‌ها از بالا به پائین بتوان به یک ترتیب شامل کلیه کارها دست یافت. برای جستجو در شاخه‌ها ابتدا بایستی یک حد پائینی برای دیرکرد به دست آورد، بدین ترتیب که با در نظر گرفتن یک گره از گره‌های سطح اول، و حرکت در جهت یکی از شاخه‌های منشعب از آن تا آخرین سطح، دیرکرد کلی ناشی از ترتیب به دست آمده محاسبه و به عنوان مبنایی جهت مقایسه در سایر شاخه‌ها بکار رود.

بنابراین در بررسی شاخه‌های دیگر چنانچه جمع دیرکرد در یک گره از دیرکرد مبنای بزرگتر یا مساوی باشد، از بررسی

گره‌های منشعب از آن صرف‌نظر و آن شاخه حذف می‌شود. در روش شاخه و حد ۲ استراتژی جستجو وجود دارد، روش جستجوی عمقی (depth first search) و روش جستجوی مرزی (border first search).

در جستجوی عمقی، اولین شاخه مورد بررسی برای به دست آوردن مبنای اولیه دیرکرد به دلخواه انتخاب می‌شود. ولی در روش جستجوی مرزی، ابتدا حدپائینی برای کلیه گره‌های سطح اول محاسبه شده و آنرا که دارای کمترین حد است در نظر می‌گیریم. سپس حدپائینی برای گره‌های منشعب از این گره محاسبه گردیده و گره با کمترین حد در نظر گرفته می‌شود. این روند ادامه می‌یابد تا آخرین سطح نیز مورد بررسی قرار گرفته و دیرکرد کلی حاصل به‌عنوان مبنا در نظر گرفته می‌شود (۲).

۲- روشهای ابتکاری

الگوریتم‌های ابتکاری زمانی که مسئله آنقدر بزرگ باشد که حل آن با روشهای بهینه‌ساز مقرون به‌صرفه نبوده، یا از نظر مدت زمان انجام غیرممکن باشد، کاربرد دارند. این الگوریتم‌ها به‌طریق غیرریاضی به‌دست آمده، و توسط قوه ابتکار و تفکر، معقول به‌نظر رسیده‌اند و در عمل ثابت شده که نسبتاً خوب هستند. روشهای ابتکاری براساس دوشبوه بوجود آمده‌اند، روشهای ابتکاری براساس الگوریتم شاخه و حد و روشهای ابتکاری براساس جستجو در همسایگی (Neighbourhood search techniques).

۱-۲- روشهای ابتکاری براساس الگوریتم شاخه و حد

در این روشها، به‌جای روش بهینه‌ساز شاخه و حد، از روش شاخه و حد بدون برگشت استفاده می‌شود. در سال ۱۹۷۰ آشور (Ashour)، روش "شناوری محدود" (Restricted floating) را ارائه نمود. در این روشی حدود پائینی در گره‌های M_1, M_2, \dots, M_n محاسبه شده و گره با کمترین حد پائینی انتخاب می‌گردد. سپس کمترین حد پائینی در زیرگره‌های آن مشخص می‌شود و بدین ترتیب این جستجو، تا یافتن برنامه کامل ادامه یافته و در گره نهایی متوقف می‌گردد. نتایج تجربی حاصل از ۱۰۰ مسئله، راندمان ۹۵٪ را به‌دست داده است که مناسب بودن روش را نشان می‌دهد (۲).

در سال ۱۹۷۶، روش جستجوی شعاعی (Beam Search Method) توسط (Lowerre) بکار گرفته شده است. این روش در هوش مصنوعی کاربرد دارد. ایده اساسی در این روش آن است که تعداد معدودی از مسیرها به‌طور موازی مورد بررسی قرار گرفته و در هر مرحله فقط بهترین جواب مسیرهای در حال جستجو نگهداری می‌شوند. سه‌طریقه در روش جستجوی شعاعی بکار می‌رود. "روش جستجوی

اولویت" که در آن از یک تابع اولویت به‌عنوان تابع ارزیابی استفاده می‌شود. "روش هزینه‌ای" که یک حد بالائی برای هزینه برنامه در نظر می‌گیرد و "روش جستجوی شعاعی تصفیه شده" که در سال ۱۹۸۹ توسط "او" و "مورتون" (Ow & Morton) ارائه گردیده که در واقع از هر دو تابع ارزیابی اولویت و ارزیابی هزینه استفاده می‌کند (۵).

۲-۲- روشهای ابتکاری جستجو در همسایگی

ایده اساسی در این روشها، به‌دست آوردن یک ترتیب اولیه و تعویض جفت‌های مجاور است به‌طوری که در ترتیب اولیه بهبود بوجود آورد. بسیاری از روشهای ابتکاری برای معیار دیرکرد، عملاً از تعویض جفت‌های مجاور استفاده می‌کنند که در ادامه به بررسی چند روش می‌پردازیم.

۱-۲-۲- روش ویلکرسون - ایروین (Wilkerson - Irwin)

روش ابتکاری ویلکرسون - ایروین در سال ۱۹۷۱ براساس تکنیک تعویض جفت‌های مجاور (Adjacent Pairwise Interchange) بوجود آمده است. کاربرد تکنیک مزبور برای بسیاری از معیارهای کارآئی منجر به دستیابی به جواب بهینه می‌شود، ولی در مورد \bar{T} الزاماً جواب بهینه را بدست نمی‌دهد و بدین سبب در گروهی از روشهای ابتکاری مطرح می‌گردد.

روش ویلکرسون - ایروین همواره با دو لیست از کارها سروکار دارد. لیست کارهای برنامه‌ریزی شده (ل.ب.ش) و لیست کارهای برنامه‌ریزی نشده (ل.ب.ن). در ل.ب.ش یک ترتیب در حال تکمیل وجود دارد که مرتباً مورد بازبینی قرار گرفته و کارهای جدید به آن اضافه می‌شود، و در ل.ب.ن کارهای باقیمانده به ترتیب زودترین موعد تحویل (EDD) قرار می‌گیرند. با اجرای مراحل مختلف الگوریتم، نهایتاً لیست اول شامل تمام کارها و لیست دوم تهی خواهد شد. در هر مرحله از اجرای الگوریتم، اولین کار برنامه‌ریزی نشده از لیست خارج شده و قاعده تصمیم‌گیری در مورد آن عملی می‌گردد. این کار، کارلولا نامیده می‌شود. بعلاوه خواهیم داشت:

$$1 = \text{آخرین کار در ل.ب.ش}$$

$$p = \text{کار لولا}$$

$$f = \text{اولین کار در ل.ب.ن}$$

$$F_i = \text{جمع زمان انجام کارهای برنامه‌ریزی شده}$$

همچنین در هر مرحله، با توجه به اینکه کارهای برنامه‌ریزی نشده به ترتیب EDD مرتب می‌شوند $d_p \leq d_r$ خواهد بود و قاعده تصمیم‌گیری برای جفت کارهای p و f بکار می‌رود. اگر $d_r + \text{Max}\{t_p, t_f\} \leq F_i + t_p$ یا $t_p \leq t_f$ باشد، در این صورت p مقدم بر f بوده و به ل.ب.ش افزوده خواهد شد. اما اگر هیچیک

از دوشروط برقرار نباشد، قاعده تصمیم‌گیری در مورد کارهای 1 و f اعمال می‌شود. یعنی اگر:

$(F_j - t_j) + \text{Max} \{t_i, t_j\} < \text{Max} \{d_i, d_j\}$ و یا $t_i < t_j$ باشد، در آن صورت f بعد از l قرار گرفته و به ل.ب.ش اضافه می‌شود.

حال اگر این قاعده نیز برقرار نباشد، یک پرش انجام می‌گیرد. یعنی کار l از ل.ب.ش خارج شده و با ترتیب EDD در ل.ب.ن قرار می‌گیرد. و قاعده تصمیم‌گیری در مورد کار f و کار ماقبل l در ل.ب.ش اعمال می‌گردد. حالت پرش بندرت روی می‌دهد ولی در صورت لزوم باید آنقدر تکرار شود تا کار l برنامه‌ریزی گردد. در برخی مواقع این روش جواب بهینه را تولید می‌کند. برای دریافتن اینکه جواب حاصل بهینه کلی نیز هست یا خیر، قاعده‌ای وجود دارد. بدین ترتیب که برای هر کار در ترتیب نهائی، یک فاصله دیرکرد (فاصله زمانی بین موعد تحویل و زمان تکمیل) تعریف می‌شود. چنانچه فواصل دیرکرد کارهای آن به ترتیب با هم همپوشانی نداشته باشد، در این صورت ترتیب حاصل بهینه است. البته این شرط برای بهینگی شرط کافی است، و ممکن است در صورت همپوشانی فواصل نیز جواب بهینه باشد، لکن راهی برای اطمینان از این موضوع وجود ندارد.

۲-۲-۲- روش فرای و همکاران

روش ابتکاری فرای و همکاران در سال ۱۹۸۹ براساس تکنیک تعویض جفتهای مجاور پدید آمده است. نتایج تجربی نشان می‌دهد که کار آئی روش با افزایش اندازه مسئله کاهش نمی‌یابد که این امر در حل مسائل بزرگ حائز اهمیت است.

با افزایش تعداد ترتیب‌های اولیه مورد بررسی، همچنین استراتژی جابجائی ترتیب کارها، در روش تعویض جفتهای مجاور می‌توان احتمال دستیابی به یک جواب خوب که به بهینه کلی نزدیک باشد را افزایش داد. فرای و همکارانش در الگوریتم ابتکاری خود از این نکته سود برده و براین اساس، سه نوع ترتیب اولیه برای مسئله بوجود آورده و سپس برای هر یک از ترتیبها، سه استراتژی (S_1 ، S_2 ، S_3) جستجو را بکار می‌گیرند. به این ترتیب ۹ جواب بهینه موضعی حاصل می‌گردد که بهترین آنها به عنوان جواب نهائی انتخاب می‌گردد. در استراتژی S_1 تعویض جفتهای مجاور از اولین محل در ترتیب اولیه آغاز شده و بسوی آخرین نوبت پیش می‌رود. زمانیکه S_1 مشخص می‌کند که دیگر ترتیب قابل جابجا شدن وجود ندارد، استراتژی S_2 تعویض جفتهای مجاور را از آخرین محل در ترتیب آغاز کرده و حرکت بسوی اولین نوبت انجام می‌گیرد. زمانی که S_2 تعیین کند که اولین جفت از کارها نباید تعویض شوند، استراتژی S_3 مورد استفاده قرار می‌گیرد. S_3 تمام جفت کارهای مجاور را قبل از جابجائی ارزیابی می‌کند و جفت کار مجاوری که در صورت تعویض بیشترین کاهش را در دیرکرد کلی بوجود آورد، شناخته و جابجا می‌کند.

سه ترتیب اولیه‌ای که بوسیله هریک از استراتژیهای فوق بررسی می‌شوند عبارتند از: زودترین موعد تحویل (EDD)، کوتاهاترین زمان انجام (SPT) و کمترین فرجه (SLK). بدین ترتیب که ابتدا کارها به ترتیب EDD مرتب می‌شوند، سپس استراتژیهای S_1 ، S_2 و S_3 بر روی آنها اعمال شده و در هر مرحله بهترین جواب نگهداری می‌شود. سپس استراتژیها در مورد کارهایی که با ترتیب‌های SPT و SLK مرتب شده‌اند اعمال می‌شود و در نهایت از میان ۹ جواب، بهترین جواب به‌عنوان ترتیب نهائی برگزیده می‌شود (۶).

با توجه به الگوریتم منظور شده برای این روش برنامه کامپیوتری به نام API تهیه گردیده که در پروژه اشاره شده در فهرست منابع قابل دسترسی می‌باشد (۳).

۳-۲- روش کلاین - ونتورا

این روش در سال ۱۹۸۹ توسط کلاین و ونتورا ارائه گردید و یک روش ابتکاری برای حداقل ساختن دیرکرد وزنی مسائل m ماشین می‌باشد. هرچند این روش برای مسائل m ماشین ابداع شده، لکن بدلیل کارا بودن روش و استفاده آن در مسائل تک ماشین، در اینجا به توضیح مختصری در باره آن می‌پردازیم (۷). این روش با در نظر گرفتن این نکته که کاهش در زمان صرف شده یک کار در سیستم منجر به کاهش مقدار دیرکرد آن می‌شود، مطرح گردیده است. تحلیل مسئله نشان می‌دهد که کاری که جریمه بزرگتری دارد باید تا آنجا که ممکن است زودتر انجام شود و کاری که موعد تحویل بزرگتری دارد دیرتر انجام گیرد. بدین منظور وزن W_i که به هر کار تخصیص داده می‌شود از رابطه $W_i = (P_i/d_i)^2$ و W_i و زمان وزنی انجام کار از رابطه $V_i = (t_i/W_i)$ بدست می‌آید که در آن P_i ، d_i و t_i به ترتیب جریمه، موعد تحویل و زمان انجام کار نام می‌باشد. در الگوریتم پیشنهادی از ترتیب صعودی کارها برحسب V_i در مراحل مختلف استفاده می‌شود تا جواب نهائی حاصل گردد.

برای الگوریتم ابتکاری کلاین - ونتورا یک برنامه کامپیوتری به نام KLEIN تهیه شده که جهت اطلاع بیشتر می‌توانید به پروژه اشاره شده در فهرست منابع این مقاله مراجعه نمایید (۳).

۱-۳-۲- روش نمونه‌گیری تصادفی

روش نمونه‌گیری تصادفی در حل مسائل برنامه‌ریزی به عنوان پلی بین روشهای ابتکاری و بهینه‌ساز شناخته شده است. استراتژی نمونه‌گیری تصادفی برخلاف بعضی از روشهای ابتکاری چون ویلکرسون - ایروین که یک ترتیب را بوجود می‌آورند و یا روشهای شاخه و حد که تمامی n! ترتیب ممکن را منظور می‌کنند، آن است که تعداد متوسطی از ترتیبها را بوجود

جدول ۱ مقایسه محاسبات لازم برای حل مسئله $n/1/A/T$ با روش منظورسازی کامل و برنامه‌ریزی پویا

| تعداد عملیات | | |
|--------------|------------------------|------------------------|
| n | منظورسازی کامل | برنامه‌ریزی پویا |
| ۴ | ۶۴۷ | ۲۳۷ |
| ۱۰ | $۲/۲۸۶ \times ۱۰^۸$ | ۳۳۷۸۹ |
| ۲۰ | $۲/۹۹۲ \times ۱۰^{۲۰}$ | $۶/۳۹۶ \times ۱۰^۷$ |
| ۴۰ | $۱/۹۸۳ \times ۱۰^{۱۵}$ | $۱/۳۵۲ \times ۱۰^{۱۴}$ |

کارها، میزان محاسبات و حافظه مورد نیاز کامپیوتر بالا خواهد بود. برای مثال برای مسائلی با ۱۰ کار بر روی کامپیوترهای PC حدود ۳ دقیقه زمان و برای ۱۵ کار به‌طور تقریبی به یکساعت زمان نیاز خواهد بود. لذا این روش برای مسائل بیش از ۱۵ کار اقتصادی نخواهد بود. علاوه بر آن به دلیل نیاز به حافظه زیاد کامپیوتر برای ذخیره کردن اطلاعات مربوط به مراحل قبل تا پایان برنامه، کارآئی این روش در مسائل با تعداد کار زیاد پائین خواهد آمد.

در روش هیبرید که رابطه میان افزایش تعداد کار با زمان حل کوچکتر از خطی است، میزان محاسبات به نحو چشمگیری کاهش می‌یابد. می‌توان گفت حل مسئله‌ای با ۱۰ کار که در روش برنامه‌ریزی پویا تقریباً به ۳ دقیقه زمان نیاز دارد، بر روی همان کامپیوتر با روش هیبرید به ۱۵ ثانیه زمان نیاز خواهد داشت. با وجود این به دلیل آنکه نمی‌توان تخمین زد که چه تعداد از کارها در مرحله اول و دوم اجرای الگوریتم برنامه‌ریزی می‌شوند، زمان اجرا برای مسائل مختلف با تعداد کار یکسان تفاوت‌های فاحشی داشته است. به‌عنوان مثال، برای مسائل با ۱۲ کار، زمان اجرا از ۱ میلی‌ثانیه تا بیش از یک دقیقه متغیر بوده است.

جدول زیر مقایسه زمانهای حل مسائل n کار را به سه روش هیبرید، برنامه‌ریزی پویا و منظورسازی کامل نشان می‌دهد. روش شاخه‌وحد به دلیل استفاده کمتر از حافظه کامپیوتر، نسبت به برنامه‌ریزی پویا از برتری قابل توجهی برخوردار است. زیرا برخلاف برنامه‌ریزی پویا که در هر مرحله اطلاعات زیادی به حافظه کامپیوتر سپرده می‌شود، و این اطلاعات تا رسیدن به جواب نهایی، جهت مقایسه بایستی در حافظه کامپیوتر نگهداری شود، در روش شاخه‌وحد علاوه بر آن که در بررسی هر مرحله اطلاعات غیرمفید حذف می‌گردد، برخی از شاخه‌ها بررسی و برخی دیگر از همان ابتدا کنار گذاشته می‌شود. میزان محاسبات و حافظه اشغالی در هر یک از دو روش

آورده و بهترین آنها را برمی‌گزیند. (یک نمونه N تائی تهیه شده و جواب مسئله در آن نمونه جستجو می‌شود). از جنبه‌های دشوار روش نمونه‌گیری آن است که چگونه یک ابزار خاص برای انجام نمونه‌گیری طراحی شود و چگونه می‌توان در مورد بهترین جواب موجود در نمونه به نتیجه رسید.

یک روش نمونه‌گیری تصادفی آن است که ابتدا همه کارها با یک قاعده (مثلاً EDD) مرتب شوند. برای تخصیص اولین محل در ترتیب، کار نوبت لازم در لیست با احتمال $P_j(1)$ انتخاب می‌شود ($j = 1, 2, \dots, n-1$). با ارائه این روند یک توزیع گسسته $P_j(k)$ در k امین مرحله بکار می‌رود.

در روش دیگر، مجموعه $P_j(k)$ ها از یک توزیع هندسی با پارامتر α برخوردارند و در این صورت:

$$P_j(k) = Q_k \alpha^k \quad \text{و} \quad j = 1, \dots, n-1$$

و Q_k یک ثابت نرمالیزه کننده است. با این ساختار، کار اول در لیست مرتب شده، دارای بیشترین احتمال انتخاب شدن است و کار دوم احتمال بزرگتر بعدی را دارد و الی آخر. این احتمالات با شکل هندسی کاهش می‌یابند ولی طبیعت این کاهش را می‌توان با انتخاب پارامتر α تعیین نمود (۱).

۳- تحلیل روشها و نتیجه‌گیری

۱-۳- روشهای بهینه‌ساز

به‌طور کلی روشهای منظورسازی، در هر مرحله از یکسری جمع و تفریق و مقایسه‌های ساده تشکیل می‌گردد. لکن آنچه می‌تواند ملاک کارآئی و مزیت یک روش بر روش دیگر قرار گیرد آن است که سرعت رسیدن به جواب بهینه در کدام روش بیشتر است. به عبارت دیگر کدام روش در حذف برنامه‌های غیرکارآ، مؤثرتر می‌باشد. به‌عنوان نمونه برای دو روش منظورسازی کامل و روش برنامه‌ریزی پویا، چنانچه تعداد عملیات ریاضی را که این دو روش برای حل مسئله $n/1/A/T$ دارند محاسبه نماییم، روشی بهتر خواهد بود که تعداد عملیات کمتری داشته باشد، چرا که مدت زمان صرف شده برای محاسبات توسط کامپیوتر با تعداد عملیات رابطه مستقیم دارد.

طبق محاسبات انجام شده در الگوریتم روشهای منظورسازی کامل و برنامه‌ریزی پویا در مسائل $n/1/A/T$ ، تعداد عملیات لازم برای روش منظورسازی کامل و روش برنامه‌ریزی پویا به ترتیب برابر با: $6n + 3(n! - 1)$ و $6n \times 2^{n-1} + 3(2^n - 1)$ می‌باشد. جدول زیر نتایج حاصل از حل مسائل ۴، ۱۰، ۲۰ و ۴۰ کار را با ۲ روش نشان می‌دهد.

روش برنامه‌ریزی پویا به دلیل آنکه از $n!$ حالت ممکنه برخی از حالات را حذف می‌کند لذا در میزان محاسبات برای جواب بهینه، صرفه‌جویی قابل توجهی را به همراه دارد. با وجود این چون محاسبات از تابع نمایی پیروی می‌کند، با افزایش تعداد

جدول ۲ مقایسه زمان حل مسائل n کار با سه روش

| متوسط زمانهای حل بر حسب میلی ثانیه | | | |
|------------------------------------|------------------|----------------|----|
| هیبرید | برنامه ریزی بویا | منظورسازی کامل | n |
| ۱۱ | ۷۰۹ | ۲۸۶۵۲ | ۸ |
| ۶۷ | ۲۴۸۱ | ۲۷۸۴۳۸ | ۹ |
| ۷۳ | ۱۱۹۳۱ | | ۱۰ |
| ۷۵ | ۴۱۴۶۱ | | ۱۱ |
| ۷۷ | ۱۴۳۵۸۰ | | ۱۲ |

جستجوی شاخه و حد یعنی جستجوی عمقی و جستجوی مرزی متفاوت می باشد. در روش جستجوی عمقی، حافظه اشغال شده کامپیوتر کمتر ولی میزان محاسبات بیشتر است. برعکس، در روش جستجوی مرزی حافظه بیشتری از کامپیوتر اشغال شده ولی به محاسبات کمتری نیاز است. با وجود این در هیچ یک از دو روش عمقی و مرزی، نمی توان تخمینی از میزان محاسبات به عمل آورد. زیرا نمی توان پیش بینی نمود شاخه اولیه انتخابی چقدر به جواب بهینه نزدیک است و چه میزان از محاسبات را حذف خواهد کرد.

روی هم رفته، برتری روش شاخه و حد بر منظورسازی کامل آشکار است. لکن به دلیل نمایی بودن رابطه اندازه کارها و زمان استفاده از این روش، در حل مسائل بزرگ توصیه نمی گردد. با توجه به مطالب عنوان شده، از میان روشهای بهینه ساز، روش هیبرید را می توان به عنوان روشی با کارایی بالاتر معرفی نمود.

۲-۳- روشهای ابتکاری

همانگونه که اشاره شد از میان روشهای ابتکاری، دسته ای

بر اساس روش شاخه و حد و گروهی بر اساس روش جستجو در همسایگی بوجود آمده اند. به دلیل آن که روشهای جستجو در همسایگی شناخته شده تر و کارآتر از گروه اول هستند، در اینجا به تحلیل و مقایسه این دسته روشها می پردازیم.

روش ویلکرسون - ایروین به طور متوسط جوابهایی با حدود ۲ تا ۳ درصد خطا می دهد، یعنی جوابهای حاصل از این روش دیرکردی ۲ تا ۳ درصد بیش از ترتیب بهینه دارند. اگر این الگوریتم، ترتیبی بوجود آورد که فواصل دیرکرد کارهای آن باهم همپوشانی نداشته باشد، در این صورت ممکن است ترتیب حاصل بهینه باشد. البته این شرط برای بهینگی شرط کافی است و چنانچه مواعدهای تحویل تقریباً در تمامی فاصله انجام کارها یعنی (F_{max} و 0) گسترده شده باشند، احتمال یافتن جواب بهینه بیشتر خواهد بود و زمانی که مواعدهای تحویل در کمتر از ۸۵٪ کل زمان انجام کارها توزیع شوند، این احتمال کمتر می شود.

روش فرای و همکاران که با ایجاد ۱۹۲ مسئله تصادفی دارای ۶، ۸، ۱۰، ۱۲ و ۱۴ کار مورد آزمایش قرار گرفته و با جوابهای بهینه آنها مقایسه گردیده حدود ۷۸/۰ درصد خطا نشان داده است، در حالی که همین ۱۹۲ مسئله با روش ویلکرسون - ایروین خطایی برابر ۹۵/۲ درصد داشته است. بنابراین برتری روش فرای و همکاران نمایان است.

روش کلاین - ونورا با ۴۰ مسئله آزمایشی مورد بررسی قرار گرفته که ۲۰ مسئله مربوط به تک ماشین و ۲۰ مسئله مربوط به m ماشین می باشد. نتایج حاصله خطایی برابر ۱۷/۰ درصد در مورد تک ماشین و ۲/۰ درصد در مورد m ماشین را نشان می دهد که خطای بسیار کمی است. اما تعداد آزمایشها برای اظهار نظر قاطع در مورد کارآئی روش کافی نمی باشد. این روش در ۷۰ درصد از موارد به جواب بهینه رسیده است. مقایسه درصد جواب بهینه با درصد خطای کوچک آن نشان می دهد که روش در موارد کمتری به جواب بهینه دست می یابد، اما جوابهای غیر بهینه آن دارای خطای اندکی هستند.

جداول ۴ و ۵ میزان خطای حاصله برای مسئله تک ماشین و

جدول ۳ مقایسه روشهای فرای و همکاران با ویلکرسون - ایروین و جوابهای بهینه

| انعطاف ناپذیری | میانگین جواب بهینه | میانگین جواب ویلکرسون - ایروین | % بزرگتر از جواب بهینه | میانگین جواب فرای و همکاران | % بزرگتر از جواب بهینه |
|----------------|--------------------|--------------------------------|------------------------|-----------------------------|------------------------|
| $R \leq 0.5$ | ۲۳/۹ | ۲۴/۳ | ۱/۲۴ | ۲۴/۰ | ۰/۱۵ |
| $R > 0.5$ | ۴۸۳/۶ | ۵۰۷/۹ | ۴/۶۶ | ۴۸۷/۷ | ۱/۳۹ |
| جمع | ۲۵۵/۱ | ۲۶۸/۳ | ۲/۹۵ | ۲۵۷/۹ | ۰/۷۸ |

جدول ۴ نتایج محاسباتی روش کلاین-ونتورا برای مسئله تک ماشین

| درصد جواب بهینه | میانگین خطا | حداقل خطا | حداکثر خطا | تعداد مسائل |
|-----------------|-------------|-----------|------------|-------------|
| ۷۰٪ | ۰/۰۱۷ | ۰/۰ | ۰/۱۱ | ۲۰ |

جدول ۵ نتایج محاسباتی روش کلاین-ونتورا برای مسئله m ماشین

| درصد جواب بهینه | میانگین خطا | حداقل خطا | حداکثر خطا | تعداد مسائل |
|-----------------|-------------|-----------|------------|-------------|
| ۶۰٪ | ۰/۰۲ | ۰/۰ | ۰/۰۸ | ۲۰ |

m ماشین با روش کلاین-ونتورا را نشان می‌دهد. در روش نمونه‌گیری به دلیل آنکه برای اجراء روش، ابتدا باید تاکتیکهای نمونه‌گیری و اندازه نمونه برگزیده شوند، بر کارآئی روش تأثیر قابل توجهی اعمال می‌گردد، زیرا به این ترتیب فرد بایستی در مورد مسائلی چون ترتیب اولیه کارها، تابع توزیع احتمال و اندازه نمونه تصمیم بگیرد. نتایج محاسباتی نشان می‌دهد که کارآئی این روش از روش ویلکرسون-ایروین

نیز کمتر است. به‌طور کلی می‌توان گفت روشهای ابتکاری که می‌توانند سریعاً جوابهای خوبی را به کمک کامپیوتر به دست دهند، بر این روش برتری محسوسی دارند. بنابراین می‌توان گفت از میان روشهای ابتکاری مسائل برنامه‌ریزی ترتیب انجام کارها، روش فرای وهمکاران به‌عنوان روشی برتر قابل ارائه و استفاده می‌باشد.

جدول ۶ بررسی روش نمونه‌گیری تصادفی با روش ویلکرسون-ایروین برای مسئله \bar{T}

| روش | میانگین نسبت | ماکزیمم نسبت | جوابهای بهینه |
|--------------------------|--------------|--------------|---------------|
| نمونه‌گیری (N = ۲۵) | ۱/۴۷ | ۴/۶۸ | ۰ |
| نمونه‌گیری (N = ۲۵۰) | ۱/۱۴ | ۱/۴۸ | ۱ |
| نمونه‌گیری (N = ۲۵۰۰) | ۱/۰۲ | ۱/۱۰ | ۹ |
| الگوریتم ویلکرسون-ایروین | ۱/۰۰۴ | ۱/۰۵ | ۱۴ |

مراجع:

- 1- Baker, K.R. *Introduction to Sequencing & Scheduling*, New York, John Wiley & Sons, 1974.
- 2- French, S., *Sequencing & Scheduling: An Introduction to the Mathematics of the Job-shop*, Chichester, England, Ellis Horwood Ltd., 1986.
- 3- ایلیکی، الف. و رضائی، م. پروژه کارشناسی مهندسی صنایع، دانشگاه امیرکبیر، نیمسال اول ۱۳۶۹
- 4- Srinivasan, V., A Hybrid Algorithm for the one-Machine Sequencing Problem to Minimize Total Tardiness, *Naval Research Logistics Quarterly*, Vol. 18, No.3, Sept. 1971, PP 317-327.
- 5- Ow, P.S., Morton, T.E., *The Single Machine Early/Tardy Problem*, *Management Science*, Vol.35, No.2, Feb. 1989, PP 177-191.
- 6- Fry, T.D., Vicens L., Macleod, K., Fernandez, S., A Heuristic Solution Procedure to Minimize \bar{T} on a Single Machine, *Journal of the Operations Research Society*, Vol.40, No.3, 1989, PP 293-297.
- 7- Klein, C.M., Ventura, J.A., A Heuristic Method for Tardiness Scheduling Problems, *Computers & Industrial Engineering*, Vol.17, No.5, 1989, PP 1-4.