زمانبندی دسته‌ای در محیط جریان کاری منعطف

محمدرضا امین‌ناصری*، عیسی نخعی** و محمدعلی بهشتی‌نیا***
دانشکدهٔ فنی و مهندسی، گروه مهندسی صنایع، دانشگاه تربیت مدرس

(دریافت مقاله: ۸۴/۷/۱۳– دریافت نسخه نهایی: ۸۶//۲۳)

چکیده – این مقاله به بررسی مسئله زمانبندی دسته‌ای در محیط جریان کاری منعطف می‌پردازد. در این مقاله فرض می‌شود که برخی از ماشینها قابلیت پردازش همزمان چند کار را دارند. این مسئله در صنایع مختلفی از قبیل صنایع تولید فنر، سیم و صنایع اتومبیل‌سازی کاربرد دارد. ابتدا مدل ریاضی عدد صحیح مختلط مسئله بیان می‌شود و سپس NP-Hard بودن مسئله نشان داده می‌شود. سه الگوریتم ابتکاری به‌نامهای H1، H2، H3 به‌منظور حل مسئله و یک کران پایین به‌منظور مقایسه الگوریتمها توسعه داده می‌شود. در انتها نیز به مقایسه الگوریتمهای ارائه شده با یکدیگر پرداخته می‌شود. نتایج نشان می‌دهد که الگوریتم H3 نسبت به سایر الگوریتمها نتایج بهتری را می‌دهد.

واژگان کلیدی : زمانبندی- جریان کاری منعطف- توالی عملیات- دسته - الگوریتم ابتکاری

Batch Scheduling in Flexible Flow Shop

M.R. Amin Naseri, I. Nakhaee, and M. A. Beheshti Nia
Department of Industn Engineering, Faculty of Engineening, Tarbial Modarres University

Abstract: *In this paper, the problem of batch scheduling in a flexible flow shop environment is studied. It is assumed that machines in some stages are able to process a number of jobs simultaneously. The applications of this problem can be found in various industries including spring and wire manufacturing and in auto industry. A mixed integer programming formulation of the problem is presented and it is shown that the problem is NP-Hard. Three heuristics will then be developed to solve the problem and a lower bound is also developed for evaluating the performance of the proposed heuristics. Results show that heuristic H3 gives better results compared to the others.*

Keywords: *Scheduling, Flexible flow shop, Sequencing, Batch, Heuristics.*

* – دانشیار　　** – استادیار　　*** – دانشجوی دکتری

فهرست علائم

c_{ji}	زمان تکمیل کار i در مرحله j	Q	یک عدد حقیقی مثبت بسیار بزرگ
c_{max}	حداکثر زمان تکمیل کارها در مرحله آخر	u_{jk}	ظرفیت انجام همزمان کارها برای ماشین kام در مرحله j
$d_{j,k,b}$	زمان تکمیل bامین دسته روی ماشین k، در مرحله j	v_{jk}	سرعت ماشین kام در مرحله j
i	شاخص کار، i=1,2,…,n	x_{jki}	اگر در مرحله j (به شرطی که ظرفیت ماشین آلات آن برابر یک باشد) کار i به ماشین k تخصیص یابد برابر یک و در غیر این‌صورت برابر صفر است
j	شاخص مرحله، j=1,2,…,m	y_{jil}	اگر در مرحله j (به شرطی که ظرفیت ماشین آلات آن برابر یک باشد) کار i قبل از کار l قرار گیرد برابر یک و در غیر این‌صورت برابر صفر است
k	شاخص ماشین، $\forall j, k = 1,2,...,l_j$	$z_{j,k,i,b}$	اگر در مرحله j (به شرطی که ظرفیت ماشین آلات آن بیشتر از یک باشد) کار i به bامین دسته[i] از ماشین k تخصیص یابد برابر یک و در غیر این‌صورت برابر صفر است
lj	تعداد ماشینها در مرحله j		
m	تعداد مراحل		
n	تعداد کارها		
p_{ij}	واحدهای زمانی مورد نیاز به‌منظور پردازش کار i در مرحله j		
p_{ijk}	مدت زمان لازم برای پردازش کار i در مرحله j، در صورت تخصیص به ماشین kام		

۱- مقدمه

این مقاله به بررسی مسئله زمانبندی دسته‌ای در محیط جریان کاری منعطف[۱] می پردازد. در این مقاله فرض می‌شود که برخی از ماشینها قابلیت پردازش همزمان چند کار را دارند و امکان دسته بندی کارها، به‌منظور پردازش سریعتر و ارزانتر آنها وجود دارد. امروزه در اکثر محیطهای تولیدی ایستگاههایی وجود دارند که در آنها چند کار به‌طور همزمان می‌توانند روی یک ماشین مورد پردازش قرار گیرند. کوره‌های پخت و عملیات گرمایی، سند بلاست، شات بلاست ، ریخته گری (قالبهای خوشه‌ای) و ... از جمله ایستگاههایی هستند که در آنها چند کار به‌طور همزمان می‌توانند روی یک ماشین مورد پردازش قرار گیرند.

مسئله جریان کاری منعطف اولین بار توسط سالوادر [۱] مطرح شد. لین و ژانگ [۲] در مقاله‌ای، تحقیقاتی که تا سال ۱۹۹۹ در زمینه مسئله جریان کاری منعطف انجام شده‌اند را بررسی کرده و آنها را از سه دیدگاه مختلف، تقسیم‌بندی کرده‌اند. تحقیقات از دیدگاه تعداد مراحل در نظر گرفته شده در محیط زمانبندی، به سه دسته‌ی دو مرحله‌ای، سه مرحله‌ای و k مرحله‌ای ($k\geq 3$) دسته‌بندی شده‌اند. از دیدگاه توابع هدف در نظر گرفته شده، مسائل به سه دسته تک معیاره، دو معیاره و چند معیاره دسته‌بندی شده‌اند. از دیدگاه روشهای حل مسئله نیز، دو دسته‌ی شاخه و حد[۲] و الگوریتمهای ابتکاری[۳] در نظر گرفته شده‌اند.

تا قبل از سال ۱۹۹۹ بجز تحقیقاتی نظیر تحقیق کوچر و موردز [۳]، اکثر تحقیقات روی مسئله جریان کاری منعطف، بدون در نظر گرفتن محدودیتهای خاصی متمرکز شده‌اند. در تحقیقات بعد از سال ۱۹۹۹، عمدتا محدودیتهای متفاوتی نیز در نظر گرفته شده‌اند.

ساویک [۴] یک روش دقیق را بر اساس مدلسازی عدد صحیح مختلط[۴]، برای حل مسئله جریان کاری منعطف با بافرهای میانی محدود و زمانبندی گروهی ارائه داده است. کیپرس و کولاماس [۵] یک الگوریتم ابتکاری برای مسئله جریان کاری منعطف با تابع هدف جمع وزنی زمانهای تکمیل کارها[۵] ارائه

داده‌اند. الگوریتم آنها مسئله مورد نظر را به یک مسئله ماشینهای موازی تبدیل کرده و پس از به‌دست آوردن توالی کارها، مجددا آن را به یک مسئله جریان کاری منعطف تبدیل می‌کند. کرز و اسکین [۶] به بررسی مسئله جریان کاری منعطف با در نظر گرفتن زمانهای آماده سازی وابسته به توالی[۶] و مجاز بودن پرش کارها از برخی مراحل پرداخته و سه دسته از الگوریتمهای ابتکاری را برای حل مسئله مورد نظر ارائه داده‌اند. الگوریتمهای دسته اول بر پایه تخصیص ساده کارها به ماشینها و بدون توجه به زمانهای پردازش آنها استوارند. الگوریتمهای دسته دوم بر پایه روشهای حل مسئله فروشنده دوره گرد[۷] و الگوریتمهای دسته سوم بر پایه قاعده جانسون[۸] استوارند. کرز و اسکین [۷] مجددا به بررسی همین مسئله پرداخته و پس از مدلسازی عدد صحیح مختلط مسئله مذکور و ارائه یک کران پایین، یک الگوریتم ژنتیک به‌نام RKGA را نیز برای آن ارائه داده‌اند. اسرو و پاترنینا [۸] به بررسی مسئله جریان کاری منعطف با تابع هدف کمینه کردن حداکثر زمان تکمیل کارها پرداخته‌اند. الگوریتم آنها بر پایه نظریه محدودیتها[۹] استوار است.

اُگاز و همکاران [۹] به بررسی مسئله جریان کاری منعطف با تابع هدف کمینه کردن حداکثر زمان تکمیل کارها پرداخته‌اند. آنها فرض کرده‌اند که برای هر عملیات در هر مرحله، ممکن است چند ماشین مورد نیاز باشد. آنها یک روش جستجو ممنوع[۱۰] را برای حل مسئله ارائه داده‌اند.

لوجندران و همکاران [۱۰] به بررسی مسئله جریان کاری منعطف با در نظر گرفتن زمانبندی گروهی[۱۱] و تابع هدف کمینه کردن حداکثر زمان تکمیل کارها پرداخته‌اند. در مسئله آنها فرض شده است که کارهای هر گروه می‌توانند روی چند ماشین مورد پردازش قرار گیرند که در این‌صورت به ازای هر ماشین، یک زمان آماده‌سازی جداگانه نیاز است. اما اگر تمام کارهای یک گروه روی یک ماشین مورد پردازش قرار گیرند، تنها یک زمان آماده‌سازی برای آنها کافی است.

کیپرسز و کولاماس [۱۱] به بررسی مسئله جریان کاری منعطف با در نظر گرفتن تفاوت سرعت ماشینها و تابع هدف کمینه کردن حداکثر زمان تکمیل کارها پرداخته‌اند. در تحقیق آنها فرض شده است که یک توالی از قبل تعیین شده از کارها وجود دارد و روش ارائه شده توسط آنها تنها به زمانبندی توالی داده شده می‌پردازد. لوجندران و همکاران [۱۲] به بررسی زمانبندی گروهی در محیط جریان کاری منعطف با زمانهای آماده‌سازی وابسته به توالی و تابع هدف کمینه ساختن بازه ساخت[۱۲] پرداخته‌اند. آنها سه الگوریتم مختلف بر پایه‌ی الگوریتم جستجو ممنوع را برای حل مسئله ارائه داده‌اند.

زندیه و همکاران [۱۳] به بررسی زمانبندی در محیط جریان کاری ترکیبی با در نظر گرفتن زمانهای آماده سازی وابسته به توالی با تابع هدف کمینه ساختن بازه ساخت پرداخته‌اند. آنها یک الگوریتم سیستم ایمنی[۱۳] را برای حل مسئله مورد نظر ارائه داده‌اند. جانیاک و همکاران [۱۴] به بررسی مسئله زمانبندی در محیط جریان کاری ترکیبی با در نظر گرفتن زمان آماده سازی برای کارها پرداخته‌اند. تابع هدف مسئله آنها شامل سه قسمت می‌شود. این سه قسمت شامل جمع وزنی زمانهای زودکرد[۱۴]، جمع وزنی زمانهای دیرکرد[۱۵] و جمع وزنی زمانهای انتظار[۱۶] می‌شود. آنها به‌منظور حل مسئله ۳ الگوریتم سازنده و ۳ الگوریتم فراابتکاری بر پایه تکنیکهای جستجو ممنوع و شبیه‌سازی بازپخت ارائه داده‌اند.

شیا و همکاران [۱۵] به بررسی مسئله زمانبندی در محیط جریان کاری منعطف با تابع هدف کمینه ساختن جمع وزنی زمان تکمیل کارها پرداخته‌اند. آنها حالتی را بررسی کرده‌اند که زمان پردازش عملیات متعلق به کارهای مشابه یکسان است. آنها برای حل مسئله، یک الگوریتم ژنتیک ترکیبی را بر پایه الگوریتمهای ژنتیک سازنده[۱۷] ارائه داده‌اند.

با توجه به تحقیقاتی که روی مسئله مذکور صورت گرفته است، از نظر روش حل مسئله، به‌طور کلی تا کنون سه دسته از روشهای حل برای مسئله ارائه شده‌اند که شامل روشهای دقیق از قبیل برنامه ریزی ریاضی و الگوریتم شاخه و حد، روشهای فرا ابتکاری[۱۸] مانند شبیه سازی بازپخت[۱۹]، الگوریتم ژنتیک[۲۰] و الگوریتم جستجو ممنوع و روشهای ابتکاری، سه گروه مزبور را تشکیل می‌دهند.

در مورد مسئله زمانبندی دسته‌ای نیز تحقیقات مختلفی صورت پذیرفته است. مونما و پاتس[۱۶] زمانبندی دسته‌ای را با توابع هدف مختلف روی محیط تک ماشینی مورد بررسی قرار داده‌اند. واخاریا و چانگ [۱۷] یک روش شبیه سازی بازپخت را برای مسئله زمانبندی دسته‌ای با تابع هدف کمینه کردن حداکثر زمان تکمیل کارها، در محیط جریان کاری ارائه کرده‌اند. در الگوریتم آنها توالی کارها در تمام مراحل یکسان فرض شده است. مونما و پاتس [۱۸] یک الگوریتم تخمینی[۲۱] را برای مسئله زمانبندی دسته‌ای با مجاز بودن قطع کارها[۲۲]، در محیط ماشینهای موازی ارائه داده‌اند. پاتس و ون‌واسنهوف [۱۹] تحقیقات مختلفی را که تا سال ۱۹۹۲ روی مسائل زمانبندی دسته‌ای انجام شده بود را مورد بررسی قرار داده‌اند. پس از آنها وبستر و بیکر[۲۰] و پاتس و کوالیو [۲۱] این بررسی را توسعه داده‌اند.

در ادامه این مقاله، در بخش دو به تعریف مسئله می‌پردازیم. در بخش سوم به ارائه روشهای مختلف برای حل مسئله مورد نظر پرداخته می‌شود و یک کران پایین برای مسئله توسعه داده می‌شود. بخش چهارم به بیان نتایج محاسباتی به‌دست آمده از مقایسه و بررسی روشهای مختلف حل مسئله اختصاص یافته است و در بخش پایانی نتیجه‌گیری و زمینه‌های تحقیقات آتی ارائه می‌شود.

۲- تعریف مسئله

در این تحقیق به زمانبندی دسته‌ای در محیط جریان کاری منعطف پرداخته می‌شود. در برخی از ایستگاهها امکان انجام همزمان چند کار توسط ماشینها وجود دارد و بنابراین کارها دسته‌بندی می‌شوند. در محیط جریان کاری منعطف فرض می‌شود، تعداد n کار وجود دارند که باید در m مرحله روی آنها پردازش انجام پذیرد. در هر مرحله مانند مرحله j تعداد l_j ماشین وجود دارد. مسیر انجام پردازش برای تمام کارها یکسان است و هر کار باید از تمام مراحل بگذرد و در هر مرحله تنها باید روی یکی از ماشینها پردازش شود. همچنین زمانهای آماده‌سازی[۲۳] ماشین آلات، حمل و نقل بین مراحل و آماده بودن[۲۴]، برای تمام کارها برابر صفر فرض شده و قطع کارها جایز نیست. ماشینها نیز دچار خرابی نمی‌شوند ولی ممکن است در برخی از مواقع دچار بیکاری شوند.

در مسئله مورد بررسی در این مقاله علاوه بر فرضیات محیط جریان کاری منعطف فرض می‌شود که ماشینهای هر مرحله کاملا مشابه هم بوده و فقط ممکن است از لحاظ سرعت انجام کار با یکدیگر تفاوت داشته باشند. همچنین فرض می‌شود مدت زمان لازم برای پردازش کار i در مرحله j، در صورت تخصیص به ماشین kام که با p_{ijk} نشان داده می‌شود از معادله زیر به‌دست می‌آید.

$$p_{ijk} = \frac{p_{ij}}{v_{jk}} | \forall i = 1,2,...,n$$

این سرعت بیانگر این مطلب است که ماشین kام در مرحله jام، می‌تواند v_{jk} واحد زمانی مورد نیاز برای پردازش کارها را در یک واحد زمان انجام دهد. استفاده از این تعریف برای سرعت ماشین‌آلات توسط کیپرسز و کولاماس[۱۱] پیشنهاد شده است. علاوه بر این فرض می‌شود که در برخی از مراحل، ماشینها می‌توانند چند کار را همزمان مورد پردازش قرار دهند که در این حالت ظرفیت ماشینها ممکن است متفاوت باشد (این ظرفیت برای ایستگاههای معمولی برابر یک کار فرض می‌شود). در مراحلی که دارای ماشینهایی با ظرفیت بیشتر از یک هستند، ممکن است پردازش کاری به پایان برسد اما باید منتظر بماند تا کلیه‌ی کارهای متعلق به آن دسته تکمیل شوند. در این حالت زمان تکمیل کارهای متعلق به هر دسته یکسان و برابر طولانیترین زمان تکمیل کارهای متعلق به آن دسته است. تابع هدف مسئله، کمینه ساختن حداکثر زمانهای تکمیل کارها در مرحله آخر (c_{max}) است.

۳- فرمولبندی مسئله

در این قسمت مدل ریاضی عدد صحیح مختلط مسئله مورد بررسی را ارائه می‌دهیم. مدل ریاضی عدد صحیح مختلط مسئله مورد بررسی، در زیر آورده شده است.

$$\text{Min } c_{max}$$

subject to:

$$\sum_{k=1}^{l_j} x_{jki} = 1 \qquad j = 1,2,\dots,m \mid u_{jk} = 1, \forall k = 1,2,\dots,l_j \quad i = 1,2,\dots,n \tag{۱}$$

$$\sum_{k=1}^{l_j} \sum_{b=1}^{\left\lceil \frac{n}{u_{j,k}} \right\rceil + 1} z_{j,k,i,b} = 1 \qquad j = 1,2,\dots,m \mid \exists u_{jk} > 1, \text{for } k = 1,2,\dots,l_j \quad i = 1,2,\dots,n \tag{۲}$$

$$\sum_{i=1}^{n} z_{j,k,i,b} \le u_{jk} \qquad j = 1,2,\dots,m \mid \exists u_{jk} > 1, \text{for } k = 1,2,\dots,l_j \quad k = 1,2,\dots,l_j \quad b = 1,2,\dots,\left\lceil \frac{n}{u_{j,k}} \right\rceil + 1 \tag{۳}$$

$$c_{ji} \ge c_{(j-1)i} + \frac{p_{ji}}{v_{jk}} - Q \times (1 - x_{jki}) \qquad j = 1,\dots,m \mid u_{jk} = 1, \forall k = 1,2,\dots,l_j \quad i = 1,2,\dots,n \quad k = 1,2,\dots,l_j \tag{۴}$$

$$d_{j,k,b} \ge c_{(j-1),i} + \frac{p_{ji}}{v_{jk}} - Q \times (1 - z_{j,k,i,b}) \qquad j = 1,\dots,m \mid \exists u_{jk} > 1, \text{for } k = 1,2,\dots,l_j \quad i = 1,2,\dots,n \quad k = 1,2,\dots,l_j \quad b = 1,2,\dots,\left\lceil \frac{n}{u_{j,k}} \right\rceil + 1 \tag{۵}$$

$$c_{0,i} = 0 \qquad i = 1,\dots,n \tag{۶}$$

$$c_{ji} + Q \times (2 + y_{jil} - x_{jki} - x_{jkl}) \ge c_{jl} + \frac{p_{ji}}{v_{jk}}$$
$$c_{jl} + Q \times (3 - y_{jil} - x_{jki} - x_{jkl}) \ge c_{ji} + \frac{p_{jl}}{v_{jk}}$$
$$j = 1,\dots,m \mid u_{jk} = 1, \forall k = 1,2,\dots,l_j \quad i,l = 1,2,\dots,n \quad i < l \quad k = 1,2,\dots,l_j \tag{۷}$$

$$c_{j,i} \ge d_{j,k,b-1} + \frac{P_{ji}}{v_{jk}} - Q \times (1 - z_{j,k,i,b}) \qquad j = 1,2,\dots,m \mid \exists u_{jk} > 1, \text{for } k = 1,2,\dots,l_j \quad i = 1,2,\dots,n \quad k = 1,2,\dots,l_j \quad b = 1,2,\dots,\left\lceil \frac{n}{u_{j,k}} \right\rceil + 1 \tag{۸}$$

$$C_{max} \ge C_{mi} \qquad i = 1,2,\dots,n \tag{۹}$$

$$d_{j,k,b} \ge d_{j,k,(b-1)} \qquad j = 1,2,\dots,m \mid \exists u_{jk} > 1, \text{for } k = 1,2,\dots,l_j \quad k = 1,2,\dots,l_j \quad b = 1,2,\dots,\left\lceil \frac{n}{u_{j,k}} \right\rceil + 1 \tag{۱۰}$$

مجموعه محدودیتهای (۱) تضمین می‌کند که برای مراحل با ماشینهای دارای ظرفیت یک، هر کار فقط به یک ماشین تخصیص می‌یابد. مجموعه محدودیتهای (۲) تضمین می‌کند که برای مراحل دارای ماشینی با ظرفیت بیشتر از یک، هر کار دقیقا به یک ماشین و به یک دسته از آن ماشین اختصاص می‌یابد. مجموعه محدودیتهای (۳) ظرفیت هر ماشین را برای مراحل دارای ماشینی با ظرفیت بیشتر از یک در نظر می‌گیرد. مجموعه محدودیتهای (۴) ارتباط بین تاریخ تکمیل هرکار با تاریخ تکمیلش در مرحله قبل را برای مراحل با ماشینهای دارای ظرفیت یک در نظر می‌گیرد. مجموعه محدودیتهای (۵) همین ارتباط را برای مراحل دارای ماشینی با ظرفیت بیشتر از یک می‌گیرد. مجموعه محدودیتهای (۶) زمان تکمیل هر کار در مرحله صفرم را برابر صفر در نظر می‌گیرد. مجموعه محدودیتهای (۷) ارتباط بین زمان تکمیل کارهای درون یک مرحله نسبت به هم را برای مراحل با ماشینهای دارای ظرفیت یک در نظر می‌گیرد. مجموعه محدودیتهای (۸) همین ارتباط را برای مراحل دارای ماشینی با ظرفیت بیشتر از یک در نظر می‌گیرد. مجموعه محدودیتهای (۹) c_{max} را با توجه به زمان تکمیل کارها در مرحله آخر تعیین می‌کند. مجموعه محدودیتهای (۱۰) بین تاریخ تکمیل دسته‌های متعلق به یک

ماشین، برای مراحل دارای ماشینی با ظرفیت بیشتر از یک، اتصال برقرار می‌کند.

به‌منظور کم کردن تعداد محدودیتها و متغیرها، در برخی از ایستگاهها که ظرفیت ماشینها برابر یک است، نوع محدودیتها تغییر پیدا کرده‌است. این امر اگر چه تنوع محدودیتها و متغیرها را افزایش می‌دهد، اما تعداد آنها را کاهش می‌دهد. به‌طور مثال اگر یک مسئله جریان کاری ترکیبی با دو مرحله عملیاتی که در مرحله اول ۲ ماشین که ظرفیت هر دو آنها سه کار است و در مرحله دوم ۲ ماشین با ظرفیت یک وجود داشته باشد، آن‌گاه به‌منظور برقراری ارتباط بین تاریخ تکمیل هرکار در مرحله دوم با تاریخ تکمیلش در مرحله اول می‌توان از مجموعه محدودیتهای (۵) استفاده کرد که در این حالت ۶۰ محدودیت باید در نظر گرفته شود. اما اگر به‌دلیل اینکه تمام ماشینهای مرحله دوم دارای ظرفیت برابر یک هستند از مجموعه محدودیتهای (۴) استفاده شود، آن‌گاه تعداد ۱۰ محدودیت باید در نظر گرفته شود.

گری و همکاران [۲۲] نشان دادند که پیچیدگی مسئله $F3//c_{max}$ از نوع NP-Hard است و از آنجا که مسئله مورد بررسی در این تحقیق، تعمیم مسئله مذکور است، بنابراین حداقل از نوع NP-Hard خواهد بود. در نتیجه پیدا کردن جواب بهینه در زمان معقول برای مسئله ناممکن است و باید از راه حلهای ابتکاری برای حل آن استفاده کرد. به‌علت اینکه مسئله مورد بررسی در این تحقیق تعمیمی از مسئله ماشینهای موازی نیز است، همین نتیجه را می‌توان از NP-Hard بودن پیچیدگی مسئله $P//c_{max}$ نیز به‌دست آورد [۲۳].

۴- توسعه الگوریتمهای ابتکاری برای حل مسئله

در این بخش سه الگوریتم ابتکاری به منظور حل مسئله توسعه داده می‌شوند. الگوریتم ابتکاری اول، H1، از یک توالی پایه برای زمانبندی استفاده می‌کند. الگوریتم ابتکاری دوم، H2، بر پایه نظریه محدودیتها و الگوریتم ابتکاری سوم، H3، نیز بر پایه الگوریتمهای مربوط به مسئله ماشینهای موازی[۲۵] استوار هستند.

برای حل مسئله مورد بررسی می‌توان آن را به دو زیر مسئله تعیین توالی کارها در هر مرحله و تخصیص کارها بر اساس توالی به‌دست آمده، به ماشینهای آن مرحله تقسیم نمود. به‌عبارت دیگر مسئله اصلی به دو زیر مسئله تعیین توالی کارها در هر مرحله و زمانبندی بر اساس توالی به‌دست آمده تبدیل می‌شود.

زمانبندی، عمل تخصیص کارها به ماشینها و تعیین زمان شروع و تکمیل برای هر کار است. فرض کنید یک توالی معین از کارها وجود دارد و می‌خواهیم کارها را به ترتیب توالی به ماشینها تخصیص دهیم. در این قسمت دو حالت زمانبندی پیشرو[۲۶] و پسرو[۲۷] را بیان می‌کنیم.

الف- زمانبندی پیشرو

در این زمانبندی نحوه تخصیص کارها به ماشینها به صورتی است که در هر مرحله، بر اساس توالی آن مرحله، کار به ماشینی تخصیص می‌یابد که زودترین زمان تکمیل را برای آن کار در آن مرحله (مستقل از سایر کارها)، به‌وجود آورد. این امر با توجه به زمان آماده بودن ماشینها در آن مرحله انجام می‌پذیرد. در این حالت ممکن است پردازش برخی از کارهای متعلق به یک دسته به اتمام برسد، اما این کارها باید منتظر بمانند تا کلیه کارهای متعلق به آن دسته تکمیل شوند. در واقع زمان تکمیل کارهای متعلق به یک دسته، زمانی است که کلیه کارهای متعلق به آن دسته تکمیل می‌شوند. وقتی که تعداد کارهای تخصیص یافته روی هر ماشین به مضرب صحیحی از ظرفیت آن ماشین رسید، یک دسته از کارها تشکیل می‌شود که زمان تکمیل آن دسته برابر طولانیترین زمان تکمیل کارهای متعلق به آن است. تعداد کارهای متعلق به یک دسته روی هر ماشین به اندازه ظرفیت آن ماشین است، مگر اینکه تخصیص کلیه کارها به اتمام رسیده باشد. در این حالت ممکن است دسته آخر روی هر ماشین، تعداد کار کمتری را نسبت به ظرفیت آن ماشین شامل باشد. زمان تکمیل کلیه کارهای متعلق به یک دسته یکسان و برابر زمان تکمیل آن دسته فرض می‌شود.

در هر مرحله زمان شروع هر کار، از معادله زیر به‌دست می‌آید.

$$S_{jki} = \max\{av_{jk}, c_{j-1,i}\} \quad (۱۱)$$

که S_{jki} بیانگر زمان شروع کار i در مرحله j، در صورت تخصیص به ماشین kام است. همچنین $c_{j-1,i}$ برابر زمان تکمیل کار i در مرحله j-1 و av_{jk} برابر زمان آماده بودن ماشین kام در مرحله jام است.

ب- زمانبندی پسرو

در زمانبندی پسرو، هر کار در هر مرحله بر اساس توالی پایه آن مرحله و با توجه به زمان آماده بودن ماشینها، بصورت رو به عقب به ماشینی اختصاص می‌یابد که مستقل از سایر کارها، دیرترین زمان شروع را برای آن کار موجب شود.

وقتی که تعداد کارهای تخصیص یافته روی هر ماشین به مضرب صحیحی از ظرفیت آن ماشین رسید، یک دسته از کارها تشکیل می‌شود که زمان شروع آن دسته برابر بزرگترین زمان شروع کارهای متعلق به آن فرض می‌شود. تعداد کارهای متعلق به یک دسته روی یک ماشین به اندازه ظرفیت آن ماشین است مگر اینکه تخصیص کلیه کارها به اتمام رسیده باشد. در این حالت، در هر ماشین ممکن است تعداد کارهای دسته اول، از ظرفیت آن ماشین کمتر باشد. زمان شروع کلیه کارهای متعلق به یک دسته یکسان و برابر زمان شروع آن دسته فرض می‌شود.

پس از بیان زمانبندیهای پیشرو و پسرو، به بیان الگوریتمهای ابتکاری می‌پردازیم.

۴-۱- الگوریتم ابتکاری H1

این الگوریتم مسئله را به m مسئله ماشینهای موازی تبدیل می‌کند. این الگوریتم پس از تعیین توالی کارها، برای تخصیص کارها به ماشینها از زمانبندی پیشرو استفاده می کند.

گام صفر:

- برای هر مرحله شاخص ρ را از طریق معادله زیر حساب کنید.

$$\rho(j) = \sum_{k=1}^{l_j} v_{jk} * u_{jk}, \qquad \forall j = 1,2,\dots,m \quad (۱۲)$$

- برای هر کار یک زمان پردازش مجازی را از طریق معادله $\widetilde{p}_i = \sum_{j=1}^{m} \frac{p_{ij}}{\rho(j)}, \forall i = 1,2,\dots,n$ حساب کنید.

گام یک: قرار دهید j=1.

گام دو: اگر j=1 کارها را بر اساس زمان پردازش مجازیشان، به‌ترتیب صعودی مرتب کنید.

در غیر این‌صورت:

کارها را بر اساس زمان تکمیلشان در مرحله قبل از j، به‌ترتیب صعودی مرتب کنید. در صورت تساوی زمان تکمیل کارها در مرحله قبلی، کار با زمان پردازش مجازی کمتر را در اولویت قرار دهید. توالی به‌دست آمده را توالی پایه بنامید.

گام سه: قرار دهید i=1.

گام چهار: قرار دهید k=1 و min=∞.

گام پنج: کار iام از توالی پایه را به مکان kام از توالی میانی تخصیص دهید. بر اساس توالی میانی به‌دست آمده، عملیات زمانبندی را انجام داده و مقدار c_{max} حاصل از این زمانبندی را به‌دست آورید.

گام شش: اگر $min > c_{max}$، آن‌گاه

قرار دهید $min = c_{max}$ و توالی میانی مزبور را ذخیره کنید.

گام هفت: k=k+1.

گام هشت: اگر k>i به گام نه بروید، در غیر این‌صورت با حذف کار iام از توالی میانی، مجددا به گام پنج بروید.

گام نه: توالی میانی را که در گام شش ذخیره کرده‌اید را در نظر گرفته و قرار دهید i=i+1.

گام ده:اگر i>n به گام یازده بروید و در غیر این‌صورت به گام چهار بروید.

گام یازده: آخرین توالی را که در گام شش ذخیره کرده‌اید را به‌عنوان توالی مرحله j در نظر بگیرید و قرار دهید j=j+1.

گام دوازده: اگرj>m به گام سیزده بروید، در غیر این‌صورت به گام دو بروید.

گام سیزده: الگوریتم را خاتمه دهید.

۴-۲- الگوریتم ابتکاری H2

الگوریتم ابتکاری دوم، از ایده مربوط به نظریه محدودیتها الهام گرفته شده است. در این الگوریتم ابتدا ایستگاه گلوگاه شناسایی می‌شود و سپس توالی کارها برای مراحل قبل از ایستگاه گلوگاه با زمانبندی پسرو مشخص می‌شوند.

پس از تعیین توالی کارها برای ایستگاههای قبل از ایستگاه گلوگاه، به زمانبندی پیشرو پرداخته می‌شود. توالی کارها برای ایستگاه گلوگاه و ایستگاههای بعد از آن، به‌ترتیب صعودی از زمان تکمیل کارها در ایستگاه بلافاصله قبل از آن، به‌دست می‌آید.

گام صفر: برای هر مرحله شاخص ρ را از طریق معادله $\rho(j)=\sum_{k=1}^{l_j} v_{jk} * u_{jk}, \forall j=1,2,...,m$ محاسبه کرده و سپس بر اساس آن برای هر مرحله شاخص indicator را از طریق معادله زیر حساب کنید.

$$\text{Indicator}(j)=\sum_{i=1}^{n}\frac{p_{ij}}{\rho(j)}, \qquad \forall j=1,2,...,m \qquad (۱۳)$$

گام یک: مرحله‌ای که بزرگترین مقدار indicator را دارد به‌عنوان گلوگاه در نظر بگیرید و شماره آن را در متغیر bottleneck ذخیره کنید.

گام دو: برای هر کار یک زمان پردازش مجازی را بر اساس معادله زیر به‌دست آورید.

$$\tilde{p}_i=\sum_{j=\text{bottleneck}}^{m}\frac{p_{ij}}{\rho(j)}, \qquad \forall i=1,2,...,n \qquad (۱۴)$$

گام سه: قرار دهید: j=bottleneck-1

گام چهار: اگر j=bottleneck-1 آن‌گاه کارها را بر اساس زمان پردازش مجازی آنها به ترتیب نزولی مرتب کنید.

در غیر این‌صورت:

کارها را بر اساس زمان شروعشان در مرحله بلافاصله بعد از مرحله j، به ترتیب نزولی مرتب کنید. در صورت تساوی زمان شروع بین چند کار، اولویت را به کار دارای زمان پردازش مجازی بزرگتر بدهید.

گام پنج: در مرحله j کارها را به‌صورت رو به عقب (بر اساس توالی به‌دست آمده در گام چهار و زمان شروعشان در مرحله j+1)، به ماشینی تخصیص دهید که مستقل از سایر کارها، دیرترین زمان شروع برای هر یک از آنها به‌دست آید (زمانبندی پسرو). در صورتی که در مرحله j=bottleneck-1 هستید زمان آماده بودن ماشینها در مرحله بلافاصله بعد را برابر یک عدد بزرگ مانند T فرض کنید.

گام شش: j=j-1

گام هفت: اگر j=0 به گام هشت بروید، در غیر این‌صورت به گام چهار بروید.

گام هشت: قرار دهید j=1

گام نه: اگر j<bottleneck

کارها را بر اساس زمان شروعشان در مرحله j مرتب کنید. در صورت تساوی زمان شروع بین چند کار، اولویت را به کار دارای زمان پردازش مجازی کوچکتر بدهید.

در غیر این‌صورت:

کارها را بر اساس زمان تکمیلشان در مرحله j-1 مرتب کنید. در صورت تساوی زمان تکمیل بین چند کار، اولویت را به کار دارای زمان پردازش مجازی کوچکتر بدهید.

گام ده: کارها را به‌صورت رو به جلو و بر اساس توالی به‌دست آمده در گام نه، به ماشینی اختصاص دهید که مستقل از سایر کارها، زودترین زمان تکمیل را برای آن کار به‌وجود آورد (زمانبندی پیشرو).

گام یازده: قرار دهید: j=j+1

گام دوازده: اگر j>m آن‌گاه به گام سیزده بروید، در غیر این‌صورت به گام نه بروید.

گام سیزده: الگوریتم را خاتمه دهید.

۴- ۳- الگوریتم ابتکاری H3

این الگوریتم مسئله را به m مسئله ماشینهای موازی تبدیل می‌کند. الگوریتم H3 ابتدا برای هر کار یک زمان پردازش مجازی را محاسبه کرده و بر اساس آن توالی کارها در مرحله

اول را تعیین می‌کند. الگوریتم سپس بر مبنای توالی به‌دست آمده، به انجام زمانبندی پیشرو می‌پردازد. در ادامه به بیان دقیقتر الگوریتم H3 می‌پردازیم:

گام صفر:

- برای هر مرحله شاخص ρ را از معادله $\rho(j)=\sum_{k=1}^{l_j} v_{jk} * u_{jk}, \forall j=1,2,\dots,m$ به‌دست آورید.
- برای هر کار یک زمان پردازش مجازی را از معادله $\tilde{p}_i=\sum_{j=1}^{m}\frac{p_{ij}}{\rho(j)}$ حساب کنید.

گام یک: قرار دهید j=1

گام دو: اگر j=1 کارها را بر اساس زمان پردازش مجازیشان به‌ترتیب صعودی مرتب کنید.

در غیر این‌صورت:

کارها را بر اساس زمان تکمیلشان در مرحله قبل به‌ترتیب صعودی مرتب کنید. در صورت تساوی زمان تکمیل بین چند کار، اولویت را به کار دارای زمان پردازش مجازی کوچکتر بدهید.

گام سه: در مرحله j کارها را به‌ترتیب توالی به‌دست آمده در گام دو و با توجه به زمانهای آماده بودن ماشین‌آلات، به ماشینی تخصیص دهید که زودترین زمان تکمیل را برای آن کار–مستقل از سایر کارها– به‌وجود آورد.

گام چهار: j=j+1

گام پنج: اگر j>m به گام شش بروید.

در غیر این‌صورت:

به گام دو بروید.

گام شش: الگوریتم را خاتمه دهید.

۴- ۴- توسعه یک کران پایین

از آنجایی که مدل مطرح شده در این مقاله تاکنون در ادبیات موضوع مطرح نشده است و از سوی دیگر بعلت NP-Hard بودن مسئله، پیدا کردن جواب بهینه در زمان معقول برای مسائل با اندازه متوسط و بزرگ ناممکن است، بنابراین مبنایی برای مقایسه کارایی الگوریتمهای ارائه شده در دست نیست. در نتیجه برای مقایسه کارایی الگوریتمها، داشتن یک کران پایین می‌تواند مبنای مقایسه مناسبی باشد. بنابراین در این بخش یک کران پایین به این منظور توسعه داده شده است که به بیان آن می‌پردازیم.

$$LB=\max_{\substack{\forall i=1,\dots,n\\ \forall j=1,\dots,m}}\{lb(j),, lb'(i)\} \tag{۱۵}$$

که

$$lb(j)=\sum_{h=1}^{j-1}\frac{\min\limits_{i=1,\dots,n}\{p_{ih}\}}{\max\limits_{k=1,\dots,l_h}\{v_{hk}\}}+\max\left\{\frac{\sum\limits_{i=1}^{n}p_{ij}}{\sum\limits_{k=1}^{l_j}v_{jk}*u_{jk}},\frac{\max\limits_{i=1,\dots,n}\{p_{ij}\}}{\max\limits_{k=1,\dots,l_j}\{v_{jk}\}}\right\}+\sum_{h=j+1}^{m}\frac{\min\limits_{i=1,\dots,n}\{p_{ih}\}}{\max\limits_{k=1,\dots,l_h}\{v_{hk}\}},\forall j=1,\dots,m \tag{۱۶}$$

و

$$lb'(i)=\sum_{j=1}^{m}\frac{p_{ij}}{\max\limits_{k=1,\dots,l_j}\{v_{jk}\}},\forall i=1,\dots,n \tag{۱۷}$$

به‌منظور توضیح، فرض کنید در مرحله‌ای مانند j قرار داریم. اولین کاری که در این مرحله تکمیل می‌شود را k_1 و آخرین کار را k_n می‌نامیم. اولین جمله از سمت چپ معادله مربوط به lb(j) یک کران پایین برای حداقل زمان ممکنی است که طول می‌کشد تا کار k_1 به مرحله j برسد. جمله دوم آن بیانگر یک کران پایین برای زمان تکمیل کار k_n، در مرحله jام است. جمله سوم معادله مربوط به lb(j) نیز بیانگر یک کران پایین برای مدت زمانی است که طول می‌کشد تا کار k_n پس از خروج از مرحله jام، کاملا تکمیل شود. در نتیجه مجموع این سه جمله یک کران پایین برای تابع هدف کمینه کردن حداکثر زمان تکمیل کارها را می‌دهد. اما چون می‌توان به جای مرحله j هر مرحله دلخواه دیگر را در نظر گرفت، بنابراین m کران پایین به‌دست می‌آید.

همچنین هر یک از مجموعه کرانهای مربوط به lb'(i)

جدول ۱- پارامترهای مسائل مختلف

پارامتر مسئله	پایین	متوسط	بالا
تعداد کارها	۱۰	۵۰	۱۰۰
تعداد ایستگاهها	۲	۱۰	۲۰
تعداد ماشینها در هر مرحله	۱	U[1,4]	U[2,8]
زمانهای پردازش	U[12,18]		U[8,22]
سرعت ماشینها		U[1,4]	
ظرفیت ماشینها	U[1,5]		U[5,10]

بیانگر یک کران پایین برای مدت زمان تکمیل یک کار دلخواه است و بدیهی است که حداکثر زمان تکمیل کارها (c_{max})، بزرگتر یا مساوی زمان تکمیل هر یک از کارهاست. در نتیجه n کران پایین دیگر نیز به‌دست می‌آید. لذا تعداد (m+n) کران پایین برای مسئله داریم و بدیهی است که بزرگترین آنها، به‌عنوان بهترین کران پایین برای مسئله انتخاب شود.

۵- نتایج محاسباتی

در این بخش ابتدا مسائل تصادفی با اندازه‌های مختلف را مورد بررسی قرار داده و سپس به مقایسه سه الگوریتم ابتکاری H1، H2 و H3 با یکدیگر می‌پردازیم.

۵-۱- تولید داده‌های تصادفی برای مسئله

در این قسمت نحوه تولید مسائل تصادفی مختلف ارائه می‌شود. شش پارامتر از مسئله انتخاب شده و برای هر کدام از پارامترها سطوحی چون بالا، متوسط و پایین - به‌صورت نشان داده شده در جدول (۱) تعریف شده است.

فرض شده است که در هر ایستگاه با احتمال ۰/۷ تمام ماشینها دارای ظرفیت برابر یک و با احتمال ۰/۳ دارای ظرفیتهای متفاوت‌اند. این ظرفیتهای متفاوت بر اساس حالات معرفی شده در جدول (۱) مشخص می‌شوند. تعداد ترکیبات مختلفی که توسط مقادیر مختلف این ۶ پارامتر می‌توان به‌دست آورد برابر ۱۰۸ (۲×۱×۲×۳×۳×۳) نوع مسئله است.

از هر نوع مسئله، ده مثال به‌صورت تصادفی ایجاد شده و در داخل یک فایل متنی قرار داده شده است (جمعا ۱۰۸۰ مسئله). کلیه برنامه‌های رایانه‌ای این تحقیق با زبان برنامه نویسی Visual Basic6 نوشته و توسط یک کامپیوتر P IV با پردازنده ۲۴۰۰ مگا هرتز اجرا شده اند.

۵-۲- مقایسه الگوریتمهای ابتکاری

پس از تهیه مسایل مختلف، هر یک از مسایل به ترتیب از داخل فایل متنی ذکر شده در قسمت ۴- ۱ خوانده شده و توسط الگوریتمهای ابتکاری H1، H2 و H3 حل شده‌اند. به‌عبارت دیگر هر یک از الگوریتمهای ابتکاری، ۱۰۸۰ بار اجرا[۲۸] شده‌اند. نتایج به‌دست آمده از اجرای این سه الگوریتم ابتکاری در جدول (۲) نشان داده شده است. معیار انحراف نسبی برای هر یک از جوابها، از معادله زیر به‌دست می‌آید.

$$\text{انحراف نسبی}=\frac{(\text{کران پایین}-\text{جواب به‌دست آمده از الگوریتم ابتکاری})}{\text{کران پایین}} \qquad (۱۸)$$

نتایج نشان می‌دهد که از نظر زمانهای حل، الگوریتمهای ابتکاری H2 و H3 تفاوت زیادی با هم ندارند، اما الگوریتم ابتکاری H1 در حل مسایل، به‌طور نسبی زمانبر نشان می‌دهد. الگوریتم ابتکاری H3 از نظر معیارهای میانگین جوابها و حداکثر جوابهای به‌دست آمده از حل مسائل مختلف، نتایج

جدول ۲- نتایج به‌دست آمده از اجرای سه الگوریتم ابتکاری

میانگین انحراف نسبی	میانگین زمانهای حل (بر حسب ثانیه)	پایینترین تابع هدف	بالاترین تابع هدف	میانگین جوابها	الگوریتم
۰/۴۳۱	۹/۲۱	۱۰/۵	۴۱۷۲/۷۵	۶۳۳/۵۵	الگوریتم ابتکاری H1
۰/۲۷۴	۰/۰۴۴	۱۰/۵	۲۴۹۷/۰۸	۵۶۱/۱۱	الگوریتم ابتکاری H2
۰/۲۳۸	۰/۰۱۷	۱۰/۵	۲۲۹۵/۸۳	۵۴۲/۳۱	الگوریتم ابتکاری H3

بهتری را نسبت به سایر الگوریتمها می‌دهد. این در حالی است که الگوریتم ابتکاری H1 از لحاظ این معیارها، بدترین نتایج را می‌دهد.

۶- خلاصه و نتیجه‌گیری

در این مقاله به بررسی مسئله زمانبندی دسته‌ای در محیط جریان کاری منعطف با تابع هدف کمینه کردن حداکثر زمان تکمیل کارها پرداخته شد. در این مقاله فرض شده است که برخی از ماشینها قابلیت انجام همزمان چند کار را دارند. ایستگاههایی مثل کوره اعم از کوره‌های پخت و عملیات گرماتی، سند بلاست، شات بلاست، ریخته‌گری (قالبهای خوشه‌ای) و ... از جمله ایستگاههایی هستند که در آنها چند کار به‌طور همزمان می‌توانند روی یک ماشین مورد پردازش قرار گیرند. ابتدا مدل ریاضی عدد صحیح مختلط مسئله مذکور ارائه شد و نشان داده شد که مسئله مذکور حداقل از نوع NP-Hard است. برای حل مسئله، آن را به دو زیر مسئله تعیین توالی کارها در هر مرحله و تخصیص کارها به ماشینها بر اساس توالی آن مرحله تقسیم کردیم. سپس سه الگوریتم ابتکاری به‌نامهای H1، H2 و H3 به‌منظور حل مسئله توسعه داده شد. الگوریتم H1 از یک توالی پایه برای زمانبندی استفاده می‌کند. الگوریتم H2 بر پایه نظریه محدودیتها و الگوریتم H3 نیز بر پایه الگوریتمهای مربوط به مسئله ماشینهای موازی استوارند. به‌منظور مقایسه کارایی الگوریتمها، کران پایینی برای مسئله مذکور توسعه داده شد. هر یک از الگوریتمها روی تعداد زیادی از مسائل تصادفی اجرا شدند که نتایج محاسباتی به‌دست آمده نشان دهنده برتری الگوریتم ابتکاری H3، از نظر معیارهای میانگین جوابها، میانگین انحراف نسبی از کران پایین و بزرگترین مقدار برای تابع هدف، نسبت به سایر الگوریتمهای ابتکاری دارد. ارائه الگوریتمهای ابتکاری و فوق ابتکاری[۲۹] مختلف با مکانیزمهای مختلف برای تعیین توالی کارها در هر مرحله و یا تخصیص به ماشینها، می‌تواند به‌عنوان زمینه‌ای برای انجام تحقیقات بیشتر روی این مسئله در نظر گرفته شود. در نظر گرفتن توابع هدف مختلف یا در نظر گرفتن دو یا چند تابع هدف به‌صورت همزمان و توسعه الگوریتمهای ارائه شده برای حل آنها می‌تواند زمینه‌ای دیگر برای تحقیقات آتی باشد. همچنین اضافه کردن محدودیتهایی مانند وجود حالت مونتاژ و یا ترکیب محیطهای زمانبندی مختلف، که باعث عملیترشدن و نزدیکتر شدن به مسائل زمانبندی واقعی می‌شوند و توسعه الگوریتمهای ارائه شده برای حل آنها می‌تواند به‌عنوان زمینه‌ای دیگر برای تحقیقات آتی در نظر گرفته شود.

واژه نامه

1. flexible flow shop
2. branch and bound
3. heuristic
4. mixed integer programming
5. total weighted completion times
6. sequence dependent
7. traveling sales man
8. johnson rule
9. theory of constraint
10. tabu search
11. group scheduling
12. make span
13. immune algorithm
14. total weighted earliness times
15. total weighted tardiness times

16. total weighted waiting times
17. constructive genetic algorithm
18. meta heuristic
19. simulated annealing
20. genetic algorithm
21. approximation algorithm
22. preemption
23. set up time
24. ready time
25. parallel machines
26. forward
27. backward
28. run
29. meta heuristic

مراجع

1. Salvador, M., “A Solution to a Special Case of .Flow Shop Scheduling Problems,” In: Elmaghraby, S.E. (Ed.), *Symposium on the Theory of Scheduling and its Applications,* Springer, Berlin, pp. 83–91, 1973.
2. Linn, R., and Zhang, W., “Hybrid Flow Shop Scheduling: a Survey,” *Computers and Industrial Engineering*, Vol. 37, pp. 57-61, 1999.
3. Kochhar, S., and Mords, J. T., “Heuristic Methods for Flexible Flow Line Scheduling,” *Journal of Manual System*, Vol. 6, pp. 299-314 , 1987.
4. Sawik, T., “An Exact Approach for Batch Scheduling in Flexible Flow Lines with Limited Intermediate Buffers,” *Mathematical and Computer Modeling*, Vol. 36, pp. 461-471, 2002.
5. Kyparisis, G. J., and Koulamas, C., “A Note on Weighted Completion Time Minimization in a Flexible Flow Shop,” *Operations Research Letters*, Vol. 29, pp. 5–11, 2001.
6. Kurz, M. E., and Askin, R. G., “Comparing Scheduling Rules for Flexible Flow Lines,” *International Journal of Production Economics*, Vol. 85, pp. 371–388, 2003.
7. Kurz, M. E. and Askin, R. G., “Scheduling Flexible Flow Lines with Sequence-Dependent Setup Times,” *European Journal of Operational Research*, Vol. 159, pp. 66–82, 2004.
8. Acero-Dominguez, M. J., and Paternina-Arboleda, C. D., “Scheduling Jobs on a K-Stage Flexible Flow Shop Using a TOC-Based (bottleneck) Procedure,” *Systems and Information Engineering Design Symposium*, pp. 295-298, 2004.
9. Oguz, C., Zinder, Y., Do, V. H., Janiak, A., and Lichtenstein, M., “Hybrid Flow Shop Scheduling Problems with Multiprocessor Task Systems,” *European Journal of Operational Research*, Vol. 152, pp. 115–131, 2004.
10. Logendran, R., Carson, S., and Hanson, E., “Group Scheduling in Flexible Flow Shop,” *International Journal of Production Economics*, Vol. 96, pp. 143-155, 2005.
11. Kyparisis, G. J., and Koulamas, C., “Flexible Flow Shop Scheduling with Uniform Parallel Machines,” *European Journal of Operational Research*, Vol. 168, pp. 1-13, 2006.
12. Logendran, R., Deszoeke, P., and Barnard, F., “Sequence-Dependent Group Scheduling Problems in Flexible Flow Shops,” International Journal of Production Economics, Vol. 102, pp. 66-86, 2006.
13. Zandieh, M. ,Fatemi Ghomi, S.M.T., and Moattar Husseini, S.M., “An Immune Algorithm Approach to Hybrid Flow Shops Scheduling with Sequence-Dependent Setup Times,” *Applied Mathematics and Computation*, Vol. 180, pp. 111–127, 2006.
14. Janiak, A., Kozan, E., Lichtenstein, M., and Oguz, C., “Metaheuristic approaches to the Hybrid Flow Shop Scheduling Problem with a Cost-Related Criterion,” *International Journal of Production Economics*,Vol. 105, pp. 407–424, 2007.
15. Shiau, D., Cheng, S., and Huang, Y., “Proportionate Flexible Flow Shop Scheduling Via a Hybrid Constructive Genetic Algorithm,” *Expert Systems with Applications,* In Press.
16. Monma, C. L., and Potts, C. N., “On the Complexity of Scheduling with Batch Setup Times,” *Operations Research*, Vol. 37, pp. 798-804, 1989.
17. Vakharia, A.J., and Chang, Y.-L., “A Simulated Annealing Approach to Scheduling a Manufacturing Cell,” Naval Research Logistics, Vol. 37, pp. 559-577, 1990.
18. Monma, C. L., and Potts, C. N., “Analysis of Heuristics for Preemptive Parallel Machine Scheduling with Batch Setup Times,” *Operations Research*, Vol. 41, pp. 981-993, 1993.
19. Potts, C. N., and Van Wassenhove, L. N., “Integrating Scheduling with Batching and Lot-Sizing: A Review of Algorithms and Complexity,” *Journal of the Operational Research Society*, Vol. 43, pp. 395-406, 1992.
20. Webster, S. T., and Baker, K. R. “Scheduling Groups of Jobs on a Single Machine,” *Operations Research*, Vol. 43, pp. 692-703, 1995.
21. Potts, C. N., and Kovalyov, M. Y., “Scheduling with Batching: A Review,” European *Journal of Operational Research*, Vol. 120, pp. 228-249, 2000.
22. Garey, M. R., Johnson, D. S., and Sethi, R., “The Complexity of Flow Shop and Job Shop Scheduling,” *Mathematics of Operation Research*, Vol. 1, pp. 117–129, 1976.
23. Garey, M. R., and Johnson, D. S., “Strongly NP-Completeness Results: Motivation Examples and Implication,” *Journal of Association for Computing Machinery*, Vol. 25, pp. 499–508, 1978.