



ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

## کاربرد الگوریتم شاخه ایی و محدود به منظور حل مشکل زمان بندی جریان

### کارکردی و مقایسه آن با الگوریتم جست جوی Tabu

#### چکیده

موقعیت پیشرو در مشارکت اقتصادی بسیاری از کشورها در تصرف کارخانه های پوشاک است و آن فرصتی مغتنم برای ارتقا حوزه های این مجموعه ها می باشد. در بنگلادش، کارخانه ی پوشاک در راس سازمان قرار دارد، که نقش حیاتی در بخش اقتصادی دارد. همانگونه که تعداد job ها (کارها) و ماشین ها افزایش می یابد؛ مشکلات زمان بندی جریان کارکردی در رویکردهای صنعتی نیز با دشواری هایی مواجه است. به یک سلول جریان کارکردی منظم با چندین مرحله ی کندساز توجه کنید. اگر این چنین باشد، صاحبان صنایع منابع بیشتری برای این مراحل کندسازکننده فراهم می کنند. در این مورد حذف عامل کند ساز در بخش تولید و ارتقا بهره وری کلی صنعت، حائز اهمیت فراوانی می باشد. در این مقاله ، به روش شاخه ایی و محدود برای حل مشکل زمان بندی جریان کارکردی در ماشین های M و jobsها ی N ، پرداخته شده است. در اینجا توالی بهینه ی jobsها از طریق به حداقل رساندن زمان انقضای کلی بوسیله ی روش کمینه/ حد پایین محدودing (LB) بر مبنای الگوریتم شاخه ایی و محدود، به دست آمده است. تاثیر/ نتیجه ی این الگوریتم از طریق نمونه ی عددی نشان داده شده و همچنین جهت یافتن راه حل مناسب از یک کد ++C برای ایجاد الگوریتم استفاده شده است. پارامترهای ورودی، زمان پردازش و توالی اجرایی برای هر کار/ job در ماشین های فراهم شده هستند. این تحقیق مقادیر بهینه ی زمان کل زمان بندی را در مقایسه با روش جست و جوی Tabu در اختیار می گذارد/ تضمین می کند.

کلید واژه ها:

Flow Shop، شاخه ایی و محدود، زمان بندی، زمان کل، کد ++C، الگوریتم جست جوی Tabu

به jobs های مختلف  $n$  که باید در ماشین های  $m$  با یک ترتیب پردازش شود، توجه کنید. هر کار (job)ی، عملکردی در هر ماشین دارد و عملکرد  $i$  job در ماشین  $j$  دارای زمان پردازش  $P_{ij}$  می باشد. این مساله؛ مشکل جریان کارکردی نامیده می شود [1و5]. در یک مشکل جریان کارکردی همه ی  $jobs$ ، در همه ی ماشین ها ترتیب توالی یکسانی دارند. یک زمان بندی تبدیلی / permutation بهینه نسبت به زمان بندی جریان کارکردی کلی و بهینه؛ عملکردی بدتر و محسوسی تولید نمی کند [4]. همچنین از دیدگاه عملی، زمان بندی ها جذاب هستند زیرا آن ها برای اجرا کردن آسان تر هستند. مشکل جریان کارکردی مقدار (هارد NP) NP-hard ، به ازای  $m \geq 3$  می باشد [3,7]. راه حل های مناسب را می توان تنها از طریق روش های تعیین شماره؛ مثل شاخه ایی و محدود به دست آورد [6]. با این حال، این روش ها ممکن است تا حدودی مانع محاسبه حتی برای مسائلی با اندازه ی متوسط شود و برای مسائل بزرگ، انجام ناپذیر و سخت می شود. این موضوع باعث توسعه ی بسیاری از روش های آزمون و خطا می شود. روش آزمون و خطا های، حل مشکلات زمان بندی جریان کارکردی را می توان به دو مقوله تقسیم کرد: روش آزمون و خطا ی تولید کننده ی توالی ( sequence generating heuristics) و آزمون و خطا ی بهبود دهنده (improvement heuristics).

روش های دسته ی اول، از scratch یک زمان بندی تولید می کنند. اکثر این روش ها، برای حل مشکلات دو و سه ماشینه؛ یا ممتدند، یا بر پایه ی نظریات الگوریتم شناخته شده ی Johnson می باشند [2,4,8,9]. شروع کردن با راه حلی ایجاد شده به وسیله ی برخی روش های آزمون و خطا ی تولید کننده ی توالی و برخی آزمون و خطا ی بهبودی، طرحی برای کسب یک توالی جدید، با مقیاس های اجرایی توسعه یافته، فراهم می کند.

روش هایی از این دست شامل تکنیک های جست و جوی همجواری (neighborhood) [5] مثل تقویت کردن همزمان و جست و جوی Tabu می باشد. جست و جوی Tabu ، یک جست و جوی محلی/local است و

براساس روشی بهینه سازی شده که به طور موفقیت آمیزی برای حل بسیاری از مسائل بهینه سازی شده و ترکیبات دشوار مخصوصا در برنامه ریزی ناحیه ای ، استفاده شده است. این روش همچنین استحکام قابل توجهی نشان داده است.

## 2. پیشینه تحقیق

بسیاری از موقعیت های کاربردی و تجربی، که به طور کلی در تولید، برای رسیدن به یک زمان بندی بهینه ی jobs ها در مجموعه ی ماشین ها دغدغه هایی را به بار می آورند ، حواس محققان و مهندسان را به خود مشغول کرده است. در زمان بندی Flow Shop، هدف دستیابی به توالی jobsهایی است که وقتی در در نظم ثابتی از ماشین ها پردازش می شوند، برخی معیارهای خوبی تعریف شده را بهینه سازی کنند. در این راستا، محققان مختلفی، کارها/ تحقیقات زیادی انجام داده اند. اول از همه، Johnson شیوه ای برای به حداقل رساندن زمان کل برای n کار (job) و مشکلات زمان بندی دو ماشین ارائه داد. کار جانسون، را Ignall و Lomnicki Palmer، Hitomi و Yoshida، Heydar، Dass و Maggu، Cambell [2]، Scharge

تلفن و آدرس ایمیل نویسنده مکاتبه کننده:

Bestwick و Hastings، Nawaz et al بیشتر بسط و توسعه دادند. Sarin و Lefoka، Koulamas، Dannenbring و غیره، هم با توجه به پارامترهای مختلف (این کار را) ادامه دادند. Hitomi و Yoshida دو مرحله ی مشکل جریان کارکردی را برای به حداقل رساندن زمان کل زمانی که مجموعه ی بهینه سازی ها از زمان پردازش جدا شده بودرا مورد توجه قرار دادند. Dass و Maggu مفهوم اصلی معادل کار/ job را برای یک job block معرفی کردند. Singh T.P و Gupta Deepak ایجاد زمان بندی دو مرحله ای بهینه را مورد مطالعه قرار دادند که در آن هم زمان پردازش و هم مجموعه ی uptime با احتمالاتی از جمله معیار job block به هم ربط داده شده بود. Heydari به مشکل زمان بندی flow shop ی پرداخت که jobs ی n

پردازش شده در دو job block مجزا و در یک ردیف متشکل از یک job block که در آن ترتیب jobها تثبیت شده و job block دیگر که در آن ترتیب jobها اختیاری بود.

Lomnicki با استفاده از روش شاخه ایی و محدود مفهوم زمان بندی جریان کارکردی را ارائه کرد. به علاوه این کار توسط Ignall و Brown، Chandrasekhara، Schrage و Lomnicki با روش شاخه ایی و محدود تا مشکل زمان بندی ماشین بوسیله ی معرفی پارامترهای مختلف ، بسط و توسعه داده شد. در موقعیت های عملی، زمان های پردازش همیشه تعیین کننده نیستند، بنابراین ما احتمالات را با زمان های پردازش آن ها در رابطه با همه ی jobها در هر سه ماشین به هم ربط داده ایم. از این رو، این مسئله در این مقاله بیشتر مورد بحث قرار گرفته دارای کاربرد مهم نتایج نظری در جریان کار صنایع می باشد.

### 3. فرضیه ها

(a) هیچ جریانی مجاز نیست.

(b) هر عملی که یکبار شروع می شود بای تا تکمیل آن انجام شود.

(c) یک job یک ماهیت است، یعنی هیچ jobی را نمی توان با بیش از یک ماشین و در یک زمان پردازش کرد.

عددنویسی ها/نشان گذاری ها:

به ما job n داده شده است که باید در مشکل زمان بندی جریان کارکردی سه مرحله ای پردازش شود و ما نشان گذاری ذیل را به کار برده ایم:

$A_i$ : زمان پردازش برای job i در ماشین A

$B_i$ : زمان پردازش برای job i در ماشین B

$C_i$ : زمان پردازش برای job i در ماشین C

$P_i$ : زمان مورد انتظار برای job i در ماشین A

$P_i2$ : زمان مورد انتظار برای job i در ماشین B

$P_i3$ : زمان مورد انتظار برای job i در ماشین C

$C_{ij}$ : زمان تکمیل job i در ماشین های A, B و C

$S_0$ : توالی بهینه

$J_r$ : زمان بندی مختصر مربوط به jobهای زمان بندی r

$J_r'$ : مجموعه ی باقی مانده ی  $(n-r)$  از jobهای آزاد.

#### 4. توسعه و تکامل دقیق / ریاضی وار

توجه کنید که job n به ما می گوید  $i=1,2,3,\dots,n$  که در سه ماشین A, B و C به ترتیب ABC پردازش شده اند. یک

job i ( $i=1,2,3,\dots,n$ ) به ترتیب دارای زمان پردازش  $B_i, A_i$  و  $C_i$  در هر ماشین می باشد، با این فرض که احتمالات

مربوط به آن ها به ترتیب  $q_i, p_i$  و  $r_i$  باشد، پس  $0 \leq p_i \leq 1$  و  $\sum p_i = 1$ ،  $0 \leq q_i \leq 1$  و  $\sum q_i = 1$ ،  $0 \leq r_i \leq 1$  و  $\sum r_i = 1$  است.

الگوی ریاضی این مساله را می توان در قالب ماتریکس این گونه بیان کرد:

**گام اول:** محاسبه ی زمان پردازش مورد انتظار  $p_i1, p_i2$  و  $p_i3$  به ترتیب در ماشین های A, B و C

**گام دوم:** محاسبه کردن:

$$\text{Max}P_i1 + \text{Min}(P_i2 + p_i3) : \text{LB1} \quad (i)$$

$$\text{Min}P_i1 + \text{Max}P_i2 + \text{Min}P_i3 : \text{LB2} \quad (ii)$$

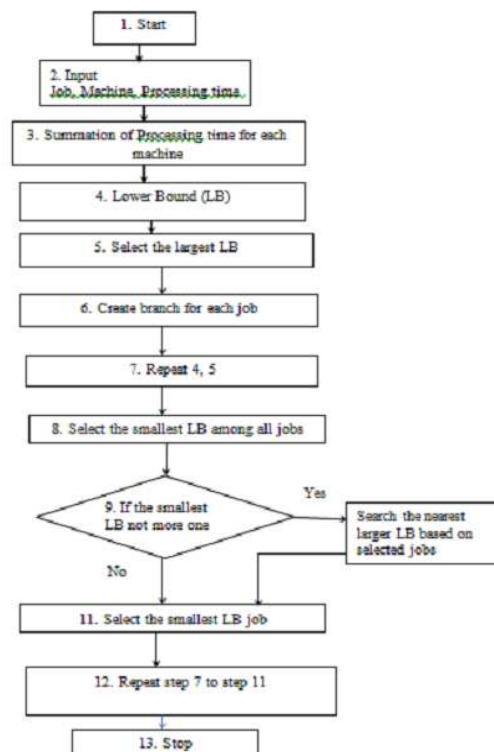
$$\text{Min}(\pi_1 + \pi_2) + \text{Max}\pi_3 : \text{LB3} \quad (\text{iii})$$

گام S, L محاسبه ی LB

گام چهارم: ادامه ی این روش تا زمانی که به انتهای نمودار درختی برسیم. در نهایت، ما به زمان بندی بهینه ی job ها دست می یابیم.

توجه کنید که در زمان بندی جریان کارکردی، سه ماشین و چهار job استفاده شده است. روش شاخه ایی و محدود برای دستیابی به مشکل زمان بندی flow shop، به کار رفته است. زمان پردازش این سه ماشین  $\pi_1$ ،  $\pi_2$  و  $\pi_3$  می باشد. به وسیله ی این روش، زمان تکمیلی را با استفاده از سه گام شاخه ایی و محدود محاسبه کنید.

#### 4.1- توسعه ی الگوریتم برای روش شاخه ایی و محدود



شکل 1: نمودار Flow ی الگوریتم شاخه ایی و محدود

## 5- تحلیل داده و محاسبه

Job\Machine	M1	M2	M3
Processing time	Pi1	Pi2	Pi3
J1	77	11	82
J2	34	92	8
J3	88	36	30
J4	1	98	9
total	200	237	129

بنابراین زمان کل بیشتر یا مساوی 237 می باشد.

$$LB1: 200 + 66 = 266$$

$$= \max P_{i1} + \min (P_{i2} + P_{i3})$$

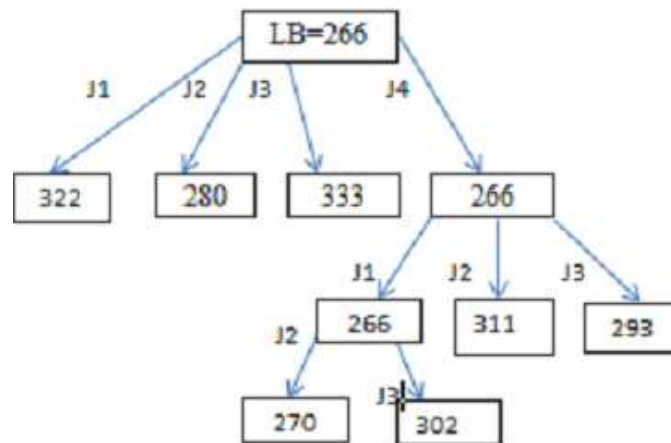
$$LB2: \min P_{i1} + \max P_{i2} + \min P_{i3}$$

$$= 1 + 237 + 8 = 246$$

$$LB3: \min (P_{i1} + P_{i2}) + \max P_{i3} = 88 + 129 = 217$$

ICMIEE-PI-140253-2

## 5.2- نمودار درختی جست و جوی شاخه ای و محدود



شکل 2: شاخه ای هایی برای توالی بهینه



### جدول 1، بررسی بین توالی قدیمی و توالی جدید

Job\Machine	M1	M2	M3
J1	77	88	170
J2	111	203	211
J3	199	239	269
J4	200	337	346

Job\Machine	M1	M2	M3
J4	1	99	108
J1	192	110	192
J2	112	204	212
J3	270	240	270

در اینجا بدون استفاده از روش شاخه ایی و محدود زمان تکمیل کلی برای توالی اولیه ی job ، 346 دقیقه می باشد.

پس از محاسبه با روش شاخه ایی و محدود ما به توالی J4-J1-J2-J3 می رسیم. زمان تکمیل کلی این توالی 270 دقیقه است. بنابراین، زمان کل کاهش یافته است. اکنون نتیجه ی این روش شاخه ایی و محدود با نتایج الگوریتم جست و جوی Tabu مقایسه می شود.

### 6- الگوریتم جست و جوی Tabu

یک رویکرد جست و جوی tabu برای حل جایگشت/تبدیل ، مشکل زمان بندی جریان کارکردی به کار می رود. پیاده سازی پیشنهادی جست و جوی tabu روش های ساده ای پیشنهاد می کند که برای ایجاد همجواری هایی از یک توالی داده شده و یک طرح ترکیبی برای تشدید و تنوع بخشیدن ، که قبلا مورد توجه قرار نگرفته است. این ویژگی های جدید منتج به اجرا می شود که انجام جست و جوی tabu ی قبلی که از مکانیسم های سادگی مقایسه استفاده می کرد، را افزایش می دهد. جست و جوی Tabu ، یک جست و جوی محلی است و مبتنی بر روشی بهینه سازی شده که به طور موفقیت آمیزی برای حل بسیاری از مسائل بهینه سازی ترکیبی و دشوار مخصوصا در برنامه ریزی ناحیه ای ،

استفاده شده است. این روش همچنین استحکام قابل توجهی نشان داده است. جست و جوی **tabu** در ساده ترین شکل آن، نیاز به اجرای ذیل دارد:

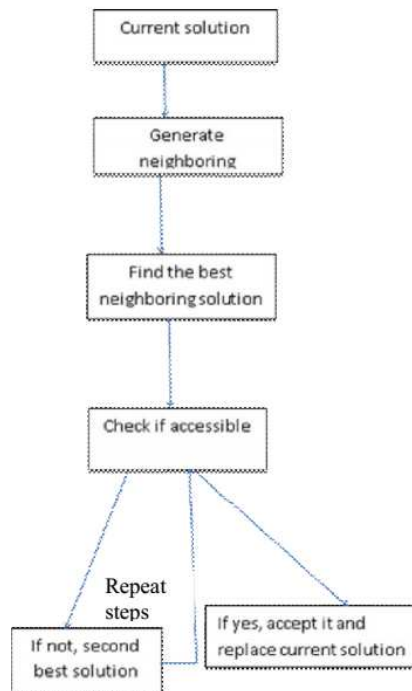
(a) توالی اولیه

(b) مکانیسم ایجاد برخی همجواری ها از توالی اخیر

(c) فهرست **Tabu**

(d) متوقف کردن معیارها

### 6.1- نمودار **Flow** ی الگوریتم پیشنهادی جست و جوی **Tabu**



### 6.2- تکمیل گام الگوریتم جست و جوی **Tabu**

(a) راه حل اولیه: در اینجا روش شاخه ایی و محدود برای به دست آوردن یک راه حل شروع اولیه به کار رفته است.

(b) ساختار همجوار: در توالی S داده شده ، ما N ها را مجموع همه ی توالی ها یی می دانیم که می توان از S به دست آورد و یکی از طرح های زیر را به کار برد:

(1) جا به جایی (الگوریتم مرتب سازی)

در توالی S داده شده، به  $i$  و  $j$  اجازه می دهد دارای دو موقعیت در توالی S باشند، با تغییر job ها در موقعیت  $i$  و  $j$  (توالی) همجوار S به دست آمده است. موقعیت های  $i$  و  $j$  را می توان با یکی از دو روش زیر تشخیص داد:

(a) موقعیت های  $i$  و  $j$  به صورت تصادفی انتخاب شده باشند؛ یا

(b) آن ها (موقعیت ها) با برخی روش های سیستماتیک شمارش شده باشند مثل جا به جایی عاقلانه ی جفت مجاور.

(2) (الگوریتم) درج سازی

در توالی S داده شده، به  $i$  و  $j$  اجازه می دهد دو موقعیت در توالی ها داشته باشند. با درج job در موقعیت  $i$  ، در موقعیت  $j$  ، (توالی) همجوار S به دست آمده است. موقعیت های  $i$  و  $j$  را می توان با یکی از دو روش زیر مشخص کرد.

(a) موقعیت های  $i$  و  $j$  به صورت تصادفی انتخاب شده باشند؛ یا

ICMIEE-PI-140253-3

(b) آن ها (موقعیت ها) با برخی روش های سیستماتیک یا نظامند مثل درج کردن هر job در هر موقعیتی شمارش شده باشند.

(c) انتخاب بهترین همجوار از فهرست موارد (کاندیدها)، تابع مورد نظر زمان کل می باشد. بنابراین، ما با مراجعه به تابع هدف، و شرایط موجود tabu به طور ساده "بهترین" را تعریف کردیم، بهترین همجوار در فهرست موارد؛ توالی ای است که کمترین زمان کل را به دست می دهد. راه های ممکن برای انتخاب بهترین همجوار شامل:

i. انتخاب اولین توالی که زمان کل را بهبود می بخشد.

ii. یافتن بهترین توالی با در نظر گرفتن زیر مجموعه ای از همجوارها

iii. جست وجوی تمام همجوارها و انتخاب بهترین توالی می باشد.

ما در الگوریتم مان، اولین گزینه را اتخاذ کردیم؛ یعنی ما همجوارها را بررسی کرده و اولین توالی ای که راه حل کنونی را اصلاح می کند/بهبود می بخشد را انتخاب کردیم. اگر عملی که راه حل را بهبود می بخشد، موجود نباشد، پس ما کل همجوارها را بررسی می کنیم.

(d) فهرست *tabu*: اندازه ی فهرست *tabu* پارامتر بسیار مهمی در الگوریتم جست و جویی *tabu* می باشد. فهرست *tabu* می تواند ثابت یا متغیر باشد.

(e) معیار *aspiration*: وقتی عمل *tabu* ی خوبی وجد داشت، به منظور در بر گرفتن فهرست *tabu*؛ ما مفهوم معیارهای *aspiration* را بیان می کنیم. ما ساده ترین شکل معیارهای *aspiration* را استفاده کردیم که به شرح زیر می باشد: یک عمل *tabu* وقتی پذیرفته می شود که آن عمل راه حلی بهتر از بهترین راه حلی که تا کنون به دست آمده را ایجاد کند.

(f) طرح تشدید و تنوع، همانگونه که قبلا ذکر شد؛ طرح تشدید غالبا حالت صفات تقویت کننده ای راه حل های خوب را به خود می گیرد در حالی که طرح تنوع عموما جست وجوی مناطقی را اجرا می کند که هنوز کشف نشده اند. روش راه اندازی مجدد تصادفی، همانگونه که از نام آن پیداست شامل راه اندازی مجدد الگوریتم با توالی ای است که به طور تصادفی ایجاد شده است. طرح دوم از ماتریکس فراوانی ای استفاده می کند که به شرح ذیل ساخته شده:

ماتریکس فراوانی یک ماتریکس  $n \times n$  می باشد که ستون هایش در موقعیت های  $n$  در یک توالی هستند و ردیف های آن  $n$  کار/*job* هستند. هر گاه که *job i* به موقعیت  $j$  برخورد می کند؛ هر ورودی  $f_{ij}$  تا یک (واحد) افزوده می شود،

توالی استفاده شده برای شروع مجدد الگوریتم تولید شده با استفاده از این ماتریکس فراوانی ، از روبه ی زیر پیروی می کند:

مرحله ی 1:  $K=1$  قرار دهیم.

مرحله ی 2: ورودی  $f_{ij}$  ماتریکس فراوانی ای را بیابیم که دارای بیشترین مقدار است. وقتی افزایشی بین دو تماس  $call$  متوالی از طرح تنوع وجود نداشته باشد، الگوریتم متوقف می شود.

مرحله ی 3: تعیین  $i$  job در موقعیت  $j$

مرحله ی 4: حذف ردیف  $i$  وستون  $j$

مرحله ی 5: تنظیم  $k=k+1$

مرحله ی 6: اگر  $k > n$  بود، متوقف کنید؛ در غیر این صورت به مرحله ی اول بروید.

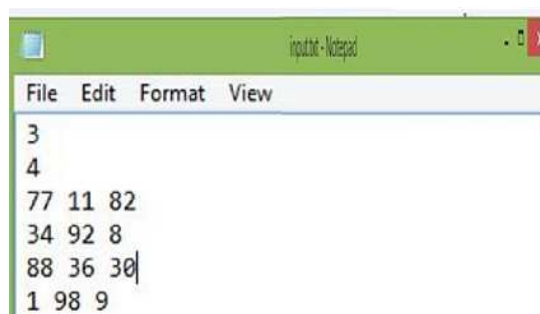
وقتی افزایشی بین دو تماس متوالی از طرح تنوع وجود ندارد، الگوریتم متوقف می شود. سایر معیارها مثل متوقف سازی را می توان از برخی مقادیر بیشینه ی تکرارها، یا متوقف سازی پس از برخی مقادیر بیشینه ی تماس های طرح تنوع، مورد استفاده قرار داد. این کار توصیف ما را از اجرا/ پیاده سازی الگوریتم جست و حوی  $tabu$  کامل می کند. در بخش بعدی ما گزارشی ارائه داده و تجارب محاسباتی خود را تحلیل می کنیم.

(g) متوقف کردن معیارها

پس از استفاده از روش شاخه ایی و محدود و روش جست و حوی  $tabu$  با نرم افزار کدینگ  $C++$  ما به زمان تکمیلی زیر دست یافتیم.

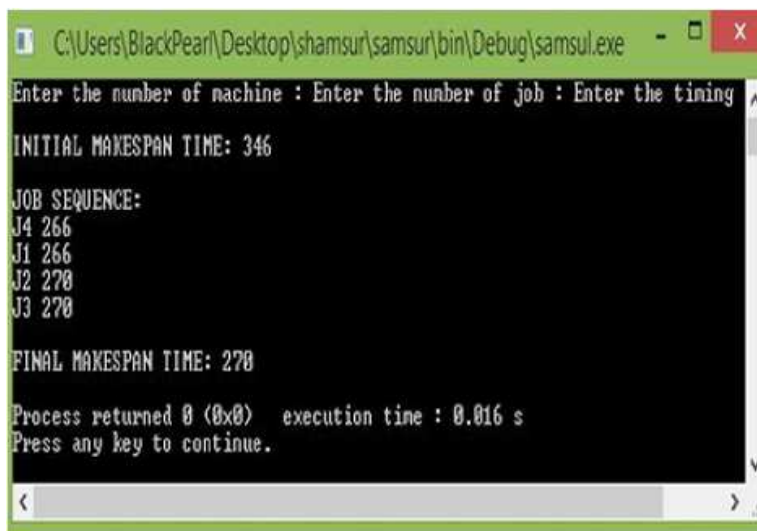
**ورودی:** در این یادداشت، ورودی ها به ترتیب و پی در پی در نظر گرفته شده اند. مانند مرحله ی اول، وارد کردن تعداد ماشین ها سپس تعداد **job** ها و در نهایت زمان پردازش **job**. وقتی همه ی اطلاعات داده شد، بعد برنامه ی **C++** را اجرا کرده و خروجی را به دست آورید.

در اینجا، هیچ ماشینی 3 و هیچ **job** ی 4 نیست. زمان پردازش سه ماشین 77، 11، 82 دقیقه است.



```
File Edit Format View
3
4
77 11 82
34 92 8
88 36 30
1 98 9
```

خروجی برای روش شاخه ایی و محدود



```
C:\Users\BlackPear\Desktop\shamsur\samsur\bin\Debug\samsul.exe -
Enter the number of machine : Enter the number of job : Enter the timing
INITIAL MAKESPAN TIME: 346
JOB SEQUENCE:
J4 266
J1 266
J2 270
J3 270
FINAL MAKESPAN TIME: 270
Process returned 0 (0x0) execution time : 0.016 s
Press any key to continue.
```

ICMIEE-PI-140253-4

```
C:\Users\BlackPearl\Desktop\shamsul\samsul1\bin\Debug\samsul1.exe
Enter number of job : Enter number of machine : Enter job to machine timing : Op
timum_sequence_is :
J4 J1 J2 J3
Optimum time is : 278
Process returned 0 (0x0)   execution time : -0.000 s
Press any key to continue.
```

## 7- نمودار Gantt

در دهه ی 1910 ، Henry Gantt نمودار Gantt را طراحی کرد. نمودار Gantt نشان دهنده ی نوعی نمودار میله ای است که زمان بندی امکان پذیری از مشکل زمان بندی را ارائه می دهد. همچنین نمودار Gantt جزئیاتی در مورد تقدم عملیات تحت job های گرفته شده در ماشین های مختلف فراهم می کند. مزیت اصلی نمودارهای Gantt این است که تجسم زمان بندی job ، برقراری ارتباط زمان بندی job با ذینفع ها ی مختلف و همچنین گروه پروژه را برای مدیر پروژه بسیار آسان می کند. نمودار Gantt برای به تصویر کشیدن برابند زمان بندی در مسائل کوچک ، متوسط مناسب است اما نشان دادن زمان بندی برای مسائلی با تعدادی کارهای بزرگ، بسیار دشوار می باشد. نمودار اندازه ی نسبی موارد کاری یا اندازه ی کلی پروژه را نشان نمی دهد. از این رو در برخی موارد مقایسه ی مقدار زمان یکسان دو پروژه بسیار سخت و دشوار می شود.

### 7.1- دلایل استفاده از نمودار Gantt

به دلیل بسیاری از مزایای ارائه شده در نمودار Gantt ، هزاران شرکت از این نمودار استفاده می کنند تا پر بازده تر شون، ارتباطات خود را ارتقا دهند، پیش بینی بلند مدت انجام دهند و نتایج را دنبال کنند. در حالی که برخی مخالفان معتقدند که آن ها اندازه ی پروژه را کاهش می دهند تا بتوانند آن را دنبال کنند. کسانی که از نمودار Gantt استفاده می کنند به مجموعه ای از مزیت های مهم توجه می کنند که شامل 5 مورد می باشد و اینج ذکر شده اند. در حالی که دلایلی برای استفاده از نمودار Gantt وجود دارد، در زیر 5 دلیل اصلی که غالبا مفید و مقرون به صرفه هستند بیان شده است:

(a) از سردر گمی در اجرا بپرهیزید. نمودار Gantt برای نگه داشتن کاربران در مسیر و ارائه ی یک جدول زمانی بصری برای شروع و خاتمه ی وظایف خاص ایجاد شده اند. تصور می شود این نمودارها با ارائه ی یک دیدگاه کلی و بصری از نقاط عطف و سایر زمان های مهم، روشی قابل فهم تر و به یاد ماندنی تر برای حفظ زمان بندی براساس وظایف و قابل ارسال بودن خواه براساس ردیابی روزانه، هفتگی، ماهانه یا سالانه بدست میدهند. نمودار زیر توانایی تجسم کسب شده در نمودار Gantt را نشان می دهد. در یک نگاه شما می توانید مشاهده کنید که مصاحبه های انجام شده 50 درصد بیشتر از آموزش ها و.. می باشد

(b) همه ی افراد را در یک صفحه نگه دارید. جایی که یک چهرچوب بصری برای کاری که باید انجام شود وجود دارد، شانس کمتری برای بدفهمی وجود دارد، به خصوص زمانی که آن کار به وظایفی بسیار پیچیده تبدیل می شود. استفاده از نمودار Gantt اجازه می دهد همه ی ذینفع ها مختلف، اطلاعات یکسان داشته باشند، درک متقابلی از انتظارات داشته باشند، و مطابق پروتکل دلخواه شان تلاش کنند.

(c) درک روابط کار. این نمودارها می توانند به وضوح نشان دهند که چگونه وظایف مختلف مرتبط هستند و شاید برای رسیدن به اهداف خود به انجام و اجرای دیگری تکیه می کنند. این روابط کاری حول محور درک زمان هر کار که بر سایر وظایف فهرست شده تاثیر می گذارد، می چرخد. این (عمل) می تواند درک بهتری از جریان کار بهینه، حداکثر بهره وری و موفقیت کلی پروژه ارائه دهد. نمودار Gantt تجسم کارهای مرتبط را بسیار آسان می سازد.



(d) منابع اختصاص یافته ی موثر. کاربران با نگاهی به آینده ی نمودار Gantt می توانند به وضوح تشخیص دهند که منابع کجا باید پیش بینی شوند، اختصاص یابند یا حداکثر استفاده از منابع کجا باید به اشتراک گذاشته شود. هر چه نمودار نزدیک تری دنبال شود، شانس بیشتری برای حفظ هزینه های پروژه در بودجه وجود دارد در حالی که همچنین اطمینان بیشتری در تکمیل به موقع پروژه وجود دارد. در شکل 3 جریان job ها بعد از زمان بندی از طریق روش شاخه ای و محدود به صورت گرافیکی نشان داده شده است.

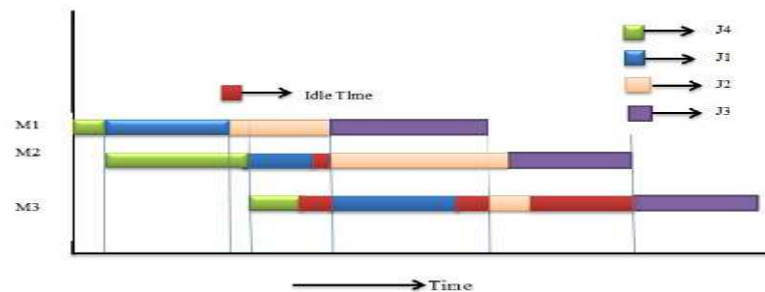


Fig 3 Gantt chart

در این تصویر می توان مشاهده کرد که همان قدر که job ها متوالی هستند، زمان بدون کار حداقل است. در اینجا رنگ قرمز ماشین آماده به کار را نشان می دهد و سایر رنگ ها زمان ماشین کاری یا زمان کاری را نشان می دهد. اگرچه این نمودار Gantt روش کار را خیلی ساده نشان می دهد اما مشکلاتی هم دارد. این روش به عنوان چاره ی کاملی برای مدیریت نا مناسب یک سازمان، طراحی نشده است. مزیت اصلی آن ؛ این است که نمودار Gantt برای بیان وابستگی های پیچیده یا پروژه های دارای تنوع پتانسیلی قابل توجه در زمان های اجرا؛ مناسب نیستند. موقعیت هایی وجود دارند که شاید سایر ابزار در واقع موثرتر باشند به ویژه سناریوهایی که یک نقطه ی عطف خاص یا یک وظیفه ی حساس به خاطر اینکه مدیر پروژه آن را شرکت نداده است، از بین می روند. سایر محدودیت ها در برگزیده ی عدم توانایی در دخیل کردن محدودیت های خاص مثل زمان، محدوده/ حوزه، و مخارج می باشد. به هر حال همه ی سازمان ها به طور کلی مزیت های نمودار Gantt در نرم افزارهای قابل اجرا را شناخته اند.

## 8- نتیجه و بحث

در این مقاله با به کار بردن الگوریتم شاخه ایی و محدود و الگوریتم جست و جوی Tabu توانستیم زمان کل (زمان اجرای کلی آخرین job) وظایف را به حداقل برسانیم. همچنین ما توانستیم زمان انتظار بین دو job که می توانند به افزایش بهره وری فلوشاپ کمک کنند را کاهش دهیم. به منظور تکمیل کارهای مذکور، اول الگوریتم شاخه ایی و محدود را به کار بردیم، سپس این الگوریتم را با الگوریتم جست و جوی Tabu مقایسه کردیم. در الگوریتم جست و جوی Tabu، ما فهرست راه حل Tabu ایجاد کردیم تا بتوانیم به راه حل های بهینه برسیم. ما اساسا روش شاخه ایی و محدود را ادامه خواهیم داد زیرا آن موثرترین روش برای زمان بندی پردازش های کلی در هر صنعت پوشاک می باشد.

اگر ما به خروجی های روش های شاخه ایی و محدود و جست و جوی Tabu توجه کنیم، به وضوح در میابیم که روش شاخه ایی و محدود بهتر از سایر روش هاست. در واقع ما در اینجا روش جست و جوی Tabu را به عنوان یک پارامتر استفاده کردیم که تاثیر روش شاخه ایی و محدود را تضمین می کند. اینجا زمان پردازش بهینه که در روش جست و جوی Tabu و روش شاخه ایی و محدود استفاده شد، 270 بار در واحد زمان است. در نهایت بهترین راه حل مناسب و بهینه ازالگوریتم شاخه ایی و محدود در ازای n تعداد job و m تعداد ماشین و از طریق مقایسه با روش جست و جوی Tabu به دست آمد.

## 9- نتیجه گیری و توصیه

در زمان بندی فلوشاپ، jobها در یک مجموعه ی منظم در در ماشین ها پردازش شده اند. مشکلات زمان بندی نواحی فلوشاپ، طبقه ی مشکلات زمان بندی با یک کارگاه یا کار گروهی، در منابع یا با سایر منابع  $1,2,...,m$  در انطباق با ترتیب های پردازش داده شده است. به ویژه حفظ جریان مداوم وظایف پردازش با حداقل زمان بدون کار و حداقل زمان انتظار مورد نظر است. برای کاهش این زمان انتظار، ما الگوریتم شاخه ایی و محدود و الگوریتم جست و جوی Tabu را در مشکل زمان بندی فلوشاپ به کار بردیم. اجرای پیشنهادی روش جست و جوی Tabu استفاده می شود اما روش های

موثر برای ایجاد همجواریها و فهرست موارد (کاندیدها) و یک طرح تشدید و تنوع ترکیبی جدید که قبلا مورد توجه قرار نگرفته بود. این ویژگی های جدید به اجرای رویکرد جست وجوی Tabu منتج می شود. نتایج بهتری به دست آمده. برای کاهش مشکل زمان بندی فلوشاپ، توصیه هایی در نظر گرفته شده:

1) در الگوریتم جست وجوی Tabu از بازیافت فهرست خودداری کنید و تا زمانی که به راه حل مورد انتظار برسیم، (فهرست) تهیه نکنید.

2) در تحقیقات بیشتر باید تلاش برای این باشد که زمان بدون کار کلی، از کل فرایند حذف شود.

3) تحقیق بهره وری کلی صنایع پوشاک را هم بهبود می بخشد.



این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

لیست مقالات ترجمه شده ✓

لیست مقالات ترجمه شده رایگان ✓

لیست جدیدترین مقالات انگلیسی ISI ✓

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی