

## مدلسازی و فهرست‌بندی در FMS با استفاده از شبکه‌های پتری و الگوریتم‌های ژنتیکی

بهزاد مشیری

استادیار گروه مهندسی برق و کامپیوتر - دانشکده فنی - دانشگاه تهران

منصور هادی‌زاده

دانشجوی فارغ‌التحصیل کارشناسی ارشد گروه مهندسی برق و کامپیوتر - دانشکده فنی - دانشگاه تهران

(تاریخ دریافت ۷۶/۷/۷، تاریخ تصویب ۷۷/۳/۳۰)

### چکیده

در این مقاله سعی خواهد شد که یک مدل جامع از یک سیستم انعطاف‌پذیر (F.M.S) به کمک شبکه پتری<sup>(۱)</sup> ارائه گردد و در ادامه به کمک الگوریتم‌های ژنتیکی روشی سریع برای یافتن یک فهرست‌بندی<sup>(۲)</sup> مناسب و بهینه ارائه گردد. در این تحقیق سعی می‌شود اولاً تمام نکات عملی موجود در F.M.S نظیر ماشین‌های CNC، ربات‌ها، حامل‌های بار نظیر<sup>(۳)</sup> AGV، نگهدارنده‌ها<sup>(۴)</sup> و همچنین مسایلی نظیر تداخل مسیر حرکت حامل‌ها و یا تداخل حرکت ربات‌ها همگی در مدلسازی مسأله مطرح شوند. ضمناً برای دستیابی به جواب‌های سریع‌تر چند روش برای فهرست‌بندی‌های محلی و نزدیک به بهینه<sup>(۵)</sup> نیز ارائه می‌شود.

کلیدواژه‌ها: سیستم‌های انعطاف‌پذیر تولید، شبکه‌های پتری، الگوریتم ژنتیک، تطابق الگو، بهینه‌سازی

### مقدمه

فرض کرده‌اند که هر ماشین امکان اجرای هر کاری را دارا می‌باشد [۷] و بعضی دیگر نیز اصولاً نگهدارنده‌ها را در نظر نداشته‌اند.

گذشته از این موضوعات تمامی کارهای ذکر شده در بالا و کارهای مشابه آنها همگی دارای یک اشکال اساسی و بسیار مهم هستند که باعث شده است مدلسازی آنها از واقعیت دور بماند و آن مسئله این است که در یک F.M.S مسأله فهرست‌بندی نه تنها به زمان اجرای سلول‌ها وابسته است، بلکه به بسیاری از روابط و توابع دیگر نیز بستگی دارد و در واقع فهرست‌بندی، تنها یافتن مسیر مناسب برای اجرای عملیات<sup>(۷)</sup> در زمان کوتاه‌ترین زمان در عین داشتن کمترین زمان در عین داشتن کمترین هزینه و تلفات و حتی داشتن بهترین کیفیت نیز می‌باشد.

اصولاً یکی از بحث‌های اساسی و مهم در سیستم‌های انعطاف‌پذیر تولید، مسأله فهرست‌بندی می‌باشد که حالت پیچیده‌ای از مسأله کلی<sup>(۶)</sup> می‌باشد که در آن هدف پیاده‌سازی  $n$  سری از عملیات بواسطه  $m$  ماشین می‌باشد [۱].

در یک سیستم انعطاف‌پذیر F.M.S مواجه با مشکلات و پیچیدگی‌های خاصی هستیم که مسأله فهرست‌بندی آن را بسیار پیچیده می‌سازد و هر چند در این مورد کارهای زیادی انجام گرفته است اما متأسفانه هیچ کدام به شکل کامل و دقیق تمامی نکات عملی موجود در یک F.M.S را مد نظر نگرفته‌اند و به عنوان مثال در بعضی از آنها اصولاً حامل‌های بار نظیر AGV در نظر گرفته نشده است [۵، ۱۱] و یا اگر مسأله AGV مد نظر بوده ربات‌ها و نحوه تداخل آنها مورد توجه نبوده است [۱۰، ۹، ۶]؛ بعضی نیز



معین بر روی مسیر انتقالی هستند و امکان انتقالات را به کمک ربات‌ها و یا روش‌های دیگر ممکن می‌سازند.

۲- نگهدارنده‌های ورودی و خروجی: که عموماً برای بالا بردن بهره‌وری و کم شدن زمان‌های بی‌کاری واحدهای نوع دوم و تنظیم زمان‌بندی‌های ما بین سلول‌ها از این نگهدارنده‌ها استفاده می‌شود. این نگهدارنده‌ها می‌توانند پلی ما بین اجزاء نوع اول و دوم نیز محسوب گردند.

#### هدف فهرست‌بندی در F.M.S

در چنین سیستمی عموماً  $m$  قطعه در دست داریم که قرار است هر کدام با کمک عملیات  $pm$  عدد از ماشین‌های موجود در کل سیستم پردازش شوند، البته باید در نظر داشت که بعضی از عملیات مربوط به هر قطعه می‌تواند توسط بیش از یک ماشین انجام گیرد.

در این حالت مسأله اصلی، مسأله فهرست‌بندی است، بدین مفهوم که مشکل یافتن  $pm$  عدد ماشین مناسب و با رعایت ترتیب صحیح برای هر یک از قطعات است تا بتوان تمام عملیات را بر روی قطعات به طور کامل انجام داد و این در حالی است که انتخاب این ماشین‌ها باید به نحوی باشد که زمان اجرای عملیات و هزینه آنها (که ممکن است توابع بسیار پیچیده‌ای از رفتار تمام سلول‌ها باشد) به طور هم زمان کمینه گردد و در ضمن باید شرایط ذیل در حین عمل فهرست‌بندی مد نظر قرار گیرد:

الف - مسأله تداخل AGV‌ها که احتمالاً دارای مسیرهای مشترک هستند باید مورد توجه قرار گیرد و همچنین مشکل تداخل ربات‌ها و یا دیگر سلول‌ها که عملکرد آنها احياناً مستقل از یکدیگر نیستند.

ب - قطعات دارای اشکال متفاوتی هستند و بنابراین نباید به جای یکدیگر استفاده شوند و در مسیرهای پردازشی یکسان قرار گیرند.

ج - حجم نگهدارنده‌ها محدود و معین است و بنابراین باید در هر لحظه ظرفیت آنها را مد نظر قرار داد.

مدلسازی سیستم انعطاف‌پذیر تولید به واسطه شبکه‌پتری

ما برای مدلسازی یک F.M.S شکل جدیدی از شبکه

در این مقاله سعی می‌شود مدلسازی در یک شکل کامل از F.M.S مطرح شود و در حین بهینه‌سازی نیز توابع هزینه و تلفات و حتی تداخل اجزاء، همگی مد نظر واقع شوند. ضمناً در این مقاله برای انجام بهینه‌سازی از الگوریتم‌های ژنتیکی<sup>(۱)</sup> GA استفاده شده است که یک روش برای دستیابی به بهینه مطلق می‌باشد. علت این قابلیت نیز آن است که این الگوریتم هیچگاه در یک بهینه محلی برای همیشه سقوط نمی‌کند.

در حقیقت GA نوعی روش جستجوی غیرخطی است که شباهت زیادی به جستجوی مبتنی بر آب دادن فولاد<sup>(۲)</sup> دارد و در آن همواره امکان فرار از هر بهینه محلی وجود دارد، در عین حال کل فرآیند در طول زمان در جهت بهینه‌تر نمودن جواب در حرکت است.

#### تعریف مسأله

#### اجزاء اصلی F.M.S

در حالت کلی می‌توان اجزاء اصلی یک F.M.S را به صورت زیر خلاصه نمود:

۱ - واحدهای پردازشی اصلی: این سری از واحدها در واقع شامل دستگاه‌هایی نظیر انواع ماشین‌های CNC مانند ماشین‌های تراش، فرز، Machine Center و حتی دستگاه‌های اندازه‌گیری و کنترل کیفیت می‌باشند که در حقیقت وظیفه اصلی عملیات اجرایی را عهده‌دار هستند و معمولاً این واحدها به عنوان پردازنده‌هایی با قابلیت استفاده مجدد<sup>(۳)</sup> مطرح می‌باشند.

۲ - واحدهای انتقال دهنده: یکی از اجزاء جداناپذیر در سیستم‌های انعطاف‌پذیر، واحدها و یا پردازنده‌هایی هستند که امکان انتقال قطعات را در بین واحدهای نوع اول امکان‌پذیر می‌نمایند. این اجزاء را می‌توان به سه دسته زیر تقسیم نمود:

الف - AGV‌ها: که وظیفه اصلی انتقالات را برعهده دارند.

ب - ربات‌ها: عموماً به منظور بارگذاری<sup>(۴)</sup> یا تخلیه نمودن<sup>(۵)</sup> قطعات از روی AGV و یا حتی انتقال مستقیم قطعات از روی واحدهای نوع اول مورد استفاده قرار می‌گیرند.

ج - ایستگاه‌های بارگذاری و تخلیه<sup>(۶)</sup>: محل‌هایی مشخص و



خود در واقع می تواند یک شبکه پتری مشتمل بر گره ها و یال های<sup>(۲)</sup> اتصال باشد.

**P(6) یا P(Place)**: نقاطی از مسیر حرکت حامل های بار را نشان می دهند که در آن نقاط یا مسیرهای عبور حامل ها با یکدیگر تلاقی دارند و یا اینکه نقاط تخلیه و یا بارگیری قطعات هستند.

در تعریف (۱) O,I,T نیز به صورت زیر بیان می شوند:

$$T = (t_1, t_2, \dots, t_n) \quad (۳)$$

که در آن  $t_i$  بیانگر یک انتقال می باشد:

$$I \text{ (Subset of ) } P(k)^*T \quad k=1, 2, 3, 4 \quad (۴)$$

که بیان تابع ورودی از  $P(k)$  به  $T$  می باشد.

$$O \text{ (Subset of ) } T^*P(k) \quad k=1, 2, 3, 4 \quad (۵)$$

که  $O$  بیانگر تابع خروجی از  $T$  به  $P(k)$  می باشد.

و در نهایت  $U$  بیانگر بک بردار دوبعدی است که از  $P(k)$  به  $M$  تعریف شده و تعداد و نوع قطعات هر گره را به صورت زیر بیان می کند:

$$U : P(k)^*S \rightarrow M^*N, \quad k = 1, 2, 3, 4 \quad (۶)$$

Where

$$S = \{s_1, s_2, \dots, s_m\}$$

$$M = \{m_1, m_2, \dots, m_n\}$$

$$N = \{1, 2, 3, \dots\}$$

که در آن  $s_i$ ها بیانگر یک مجموعه مرتب از فضای منفصل زمانی و  $m_i$ ها بیانگر نوع قطعات<sup>(۳)</sup> است که هر کدام تنها می تواند در عملیات یک فرآیند خاص شرکت نماید.

**FP**: حال اگر فرض کنیم که  $f_i = (f_1, f_2, \dots, f_n)$

مجموعه تمام عملیات مجاز  $P_i$  و  $A = (a_1, a_2, \dots, a_m)$  و همچنین  $S = (s_1, s_2, \dots, s_m)$  یک مجموعه مرتب از فضای منفصل زمانی باشد، می توان **FP** را به صورت زیر تعریف نمود:

$$FP : P^*S \rightarrow A \quad (۷)$$

**Ti**: این پارامتر، یک بردار دوبعدی است که زمان اجرای پردازش هر قطعه بر روی هر ماشین را نشان می دهد و به شکل زیر تعریف می شود:

$$TI : FP(P, S) \rightarrow [R^+] \quad (۸)$$

**CT**: اگر فرض شود که حرکت بر روی هر یک از یال های انتقالات هزینه ای را می طلبد، این هزینه را توسط این بردار نمایش می دهیم که به صورت زیر تعریف

پتری با عنوان  $PNF^{(۱)}$  تعریف نموده ایم که آن را در زیر معرفی می کنیم:

شبکه پتری مبتنی بر عملکرد  $PNF$

همانطور که قبلاً گفته شد  $PNF$  در واقع یک  $PN$  است که تعاریف هر کدام از اجزاء آن به صورت کامل تری در آمده است و در اینجا اشاره ای به این تعاریف می شود.

یک  $PNF$  به صورت کلی زیر نمایش داده می شود:

$$PNF = (P(K), T, I, O, U, FP, TI, CT, CF, PTH, FL, CR, PR, ON) \quad (۱)$$

و در آن:

$$P(K) = \{P_1(k), P_2(k), \dots, P_m(k)\} \quad k=1, 2, 3, 4 \quad (۲)$$

که در آن  $P_i$  بیانگر تمامی دستگاهها و یا سلول های مورد استفاده در سیستم و یا مفسر حالت خاصی از سیستم می باشد.  $k$  نیز در واقع نوع این سلول ها را با توجه به توصیف ذیل بیان می دارد:

**P(1) یا P(main processor)**: بیانگر ماشین هایی است

که عملیات اصلی پردازش قطعات را برعهده دارند، این اجزاء در حقیقت همان واحدهای نوع اول می باشند که در بخش ۱-۲ مطرح شدند.

**P(2) یا P(AVG)**: مبین حمل کننده های موجود در

سیستم می باشد.

**P(3) یا P(Buffer)**: یک مجموعه نگهدارنده ها را

توصیف می کند.

**P(4) یا P(Critical Area)**: این مجموعه یکی از نکات

مهمی است که در طرح ریزی شبکه بکار رفته در نظر گرفته شده است و آن مسأله تداخل عملکردی سلول ها است که می تواند به دو صورت جلوه نماید. نوع اول شامل تداخل های فیزیکی است که در اثر برخوردهای مستقیم حاصل می شود نظیر برخورد دو ربات با یکدیگر ولی نوع دوم این تداخل ها نتیجه تأثیر رفتاری یک دسته از سلول ها بر روی سلول های دیگر می باشد، تأثیر عملکردی سلول بر هزینه و یا زمان اجرای سلول دیگر از این نوع تداخل ها محسوب می شود. در این حالت ها برای نمایش نوع و شکل تداخل از این گره ها در شبکه استفاده می شود.

**P(5) یا P(Sub Node)**: نمایانگر یک زیر گره است که



$$PR = (pr_1, pr_2, \dots, pr_n) \quad (15)$$

$pr_i = 1$  if all outputs of  $i$ th node is available

$pr_i = 0$  if many outputs of  $i$ th node is available

ON: این عنصر در صورتی که یک گره دارای

محدودیت در خروج هم زمان قطعات باشد، در هر لحظه

و برای هر قطعه و در هر پردازنده، شماره نظیر یال

خروجی را تعیین می نماید. بر این اساس ON در فضای زیر

تعریف می شود:

$$ON : P^*S \rightarrow [N], \quad N = (0, 1, 2, \dots) \quad (16)$$

قانون آتش شدن<sup>(۱)</sup>

مجموعه مکان های ورودی و خروجی را برای هر

انتقال به صورت زیر تعریف می نمایم:

- مجموعه مکان های ورودی:

$$IP_i = \{P | (ti, p) \text{ (member) } I\} \quad (17)$$

- مجموعه مکان های خروجی:

$$OP_i = \{P | (ti, p) \text{ (member) } O\} \quad (18)$$

در این حالت می توان نحوه آتش شدن هر انتقال که

عامل حرکت و پویایی مدل و سرچشمه تمام عملیات است

را براساس انتقالات مجاز تعریف نمود، در حالی که یک

انتقال مجاز خود به شکل زیر تعریف می گردد:

- انتقال مجاز<sup>(۲)</sup>: یک انتقال را مجاز می نمایم اگر تمام

شرایط زیر برای آن انتقال برآورد شده باشند:

۱- تمام یال های ورودی از هر انتقال دارای یک قطعه مجاز

از گره قبلی باشند و این به معنی آن است که گره متصل به

هر یال ورودی براساس متغیر  $ON(p,s)$  دارای یک قطعه

باشد. بنابراین اگر  $T_{m,lk}$  بیانگر یال  $k$  ام از یال های ورودی

انتقال  $m$  هم باشد می توان این شرط را برای زمان  $s$  به

صورت زیر نوشت:

$$\forall k, \exists j, \quad ON(p_j, s) = T_{m,lk}, \quad p_j \text{ (member) } IP_m \quad (19)$$

۲- تمام یال های خروجی انتقال به گره هایی ختم شوند که

براساس متغیر  $CR(p,s)$  دارای جای خالی و

می گردد:

$$CT : T^*I^*O \rightarrow [R^+] \quad (9)$$

CF: این متغیر میزان هزینه اجرای هر عمل را در هر

گره نشان می دهد و به صورت زیر تعریف می شود:

$$CF : FP(P, S) \rightarrow [R^+] \quad (10)$$

PTH: اگر  $L_i = (L_1, L_2, \dots, L_h)$  مجموعه تمام

یال هایی (مسیرهایی) باشد که  $i$  ام می تواند روی آن

حرکت کند و  $L = (L_1, L_2, \dots, L_m)$  در این صورت

PTH به صورت زیر تعریف می شود:

$$PTH : P(AGV)^*S \rightarrow L \quad (11)$$

در واقع PTH نشان می دهد که تاریخچه حرکت هر

AGV در طول زمان بر روی مسیرهای مجاز چگونه بوده

است و مسیر حرکت  $i$  ام که مجموعه ای از

دوتایی های مرتب می باشد، به صورت زیر است:

$$PTH_i(\text{time}) = \{(path, time) | path \text{ (member) } L_i, time \text{ (member) } S\} \quad (12)$$

FL: یک بردار با طولی برابر تعداد تمام گره ها یعنی

$P(k)$  ها که به صورت زیر تعریف می شود:

$$FL = (r_1, r_2, \dots, r_n) \quad (13)$$

where

$r_i = 1$  if machine is Idle

$r_i = 0$  if machine is active

CR: این پارامتر نیز برداری با طول برابر تعداد گره های

موجود در سیستم می باشد که بیان می دارد کدام گره ها

اجازه ورود چند قطعه را با هم می دهند و به شکل زیر

تعریف می شود:

$$CR : P^*S \rightarrow N, \quad N = (1, 2, 3, \dots) \quad (14)$$

Where

$CR = (c_1, c_2, \dots, c_n)$

$c_i = 1$  if node has  $m$  available space.

$c_i = 0$  there is not available space.

PR: این پارامتر نیز شبیه CR یک بردار می باشد و بیان

می دارد که آیا قطعه های خروجی از یک گره می توانند در

تمام یال های خروجی گسترش یابند و یا این گسترش تنها

در بعضی از آنها انجام می گیرد. PR را می توان به صورت

زیر تعریف نمود:



همراه یک الگوریتم پرهیز از بن بست<sup>(۲)</sup> نظیر الگوریتم بانکدار<sup>(۳)</sup> به انجام رساند.

بدین وسیله می توان اطمینان یافت که اجرا در یک محیط سالم<sup>(۴)</sup> ادامه پیدا می کند. اما همان طور که انتظار می رود این روش به علت تأخیرهای بسیار زیادی که الگوریتم پرهیز از بن بست برای اجرای عملیات قطعات بوجود می آورد بهره وری خط تولید را در حین ساخت قطعات به شدت کاهش می دهد هر چند که کل عملیات فهرست بندی را تنها به ایجاد فهرست های مستقل قطعات، محدود می کند و در نتیجه زمان لازم برای فهرست بندی را نیز کاهش می دهد.

ب - در این رویکرد، تأثیرات و تداخل های (Interactions) مابین عملیات قطعات در نظر گرفته می شود و این بدان معنا است که باید بر همکنش های سلولها چه از لحاظ فیزیکی و چه از لحاظ اولویت بندی قطعات در ورود به گره ها مورد توجه قرار گیرند، در این حالت همچنین باید مسأله بن بست<sup>(۵)</sup> در ضمن فهرست بندی حل شود.

بنابراین در این مقاله سعی شده است تا فهرست بندی قطعات در دو قسمت انجام گیرد، این تفکیک اولاً به ما امکان می دهد تا بتوانیم دو رویکرد (الف) و (ب) که در بالا ذکر آنها رفت را عملی سازیم و ثانیاً در بسیاری از موارد خواهیم توانست به علت آن که فهرست بندی تک قطعات تداخل های کمی را با هم دارند بازدهی این الگوریتم دو مرحله ای را بیشتر از اجرای الگوریتم هم زمان نمایم هر چند که در بدترین حالت هر دو الگوریتم شبیه یکدیگر عمل می نمایند.

بر این اساس فهرست بند، شامل یک هسته مرکزی جستجو است که عمل فهرست بندی تک قطعات را با بهره گیری از الگوریتم ژنتیکی خود سازمانده<sup>(۶)</sup> انجام می دهد و همچنین دارای یک واحد تصحیح کننده رفتار فهرست بندی در حین ایجاد تداخل ما بین مسیرهای تک قطعات می باشد که در واقع عمل فهرست بندی مجموعه انبوهی از قطعات<sup>(۷)</sup> را برعهده دارد و در طی انجام این واحد سعی می شود تا تمامی اهداف و نیازهای مربوط به رویکرد (ب) پیاده سازی شود. این دو واحد اصلی فهرست بندی در بخش بعدی توضیح داده می شوند.

قابل دسترسی<sup>(۱)</sup> برای ورود قطعه را داشته باشند، بنابراین اگر  $T_{m,ok}$  بیانگر یال  $k$  ام از یال های خروجی انتقال  $m$  ام باشد می توان این شرط را برای زمان  $s$  به صورت زیر نوشت:

$$\forall k, \exists j, CR(p_j, s) > 0, \quad p_j \text{ (member) } O_{pm} \quad (20)$$

بر این اساس می گوئیم یک انتقال آتش می شود هر گاه آن انتقال یک انتقال مجاز باشد و طی آن آتش شدن بردار  $U$  به شکل زیر به  $U'$  تبدیل می گردد:

$$U' (P_i(K, s' = s + Dt)) = U(P_i(k, s)) \cup \{V_j, [m_j, n_j]\} \quad (21)$$

که در این رابطه متغیر  $j$  براساس شرایط (۲۰) و (۲۱) حاصل می آیند.

### فهرست بندی

به طور کلی فهرست بندی در یک محیط عملی همواره بر روی تعداد زیادی از قطعات انجام می گیرد و بنابراین می باید بهینه سازی و فهرست بندی به طور هم زمان برای همه آنها صورت پذیرد. اما این امر باعث می شود تا پیچیدگی حل مسأله بسیار زیادتر بگردد به نحوی که اگر حجم محاسبات در فهرست بندی مستقل  $m$  قطعه برابر  $P(m)$  باشد، در حالت کلی برای دستیابی به فهرست بندی بهینه تمام قطعات به طور هم زمان حجم محاسبات به صورت  $P(m)$  به توان  $m$  تعریف خواهد شد.

روشی که در این بررسی انتخاب شده است و می تواند در مواردی مشکل بالا را حل نماید، آن است که فهرست بندی هر یک از قطعات به طور مستقل و مجزا از یکدیگر انجام گیرد و در این حالت با در دست داشتن مسیر این قطعات دو رویکرد زیر در نظر گرفته می شود:

الف - در این رویکرد، فهرست بندی های مستقل و منفک قطعات، که هر کدام به طور مجزا یک فهرست بندی بهینه و یا نزدیک به بهینه می باشد، به منظور اجرا به کنترل مرکزی ارسال می گردد و برای آن که از برخوردها و حالات ناهنجار نظیر ایجاد بن بست در حین اجرای عملیات جلوگیری گردد، کنترل کننده، باید مدیریت اجرا را به



## پیاده‌سازی واحد فهرست‌بند تک قطعه‌ای به کمک الگوریتم ژنتیکی

### الگوریتم ژنتیکی خود سازمانده

فهرست‌بندی در FMS در واقع شامل دو مرحله نسبتاً مجزا است، در مرحله اول می‌باید مسیری را یافت که فعالیت‌های مورد نظر ما را طی عبور از گره‌های خاصی دنبال نماید به عبارت دیگر در این مرحله می‌باید نوعی از عملیات تطابق الگو<sup>(۱)</sup> را به انجام رساند. اما در مرحله بعد که لزوماً از لحاظ زمانی جدا از مرحله قبل انجام نمی‌گیرد، می‌باید مسیر را طوری برگزید تا مقادیر اصلی در تصمیم‌گیری نظیر کل زمان و هزینه اجرا کاهش یابد و یا متغیرهایی نظیر بهره‌وری و میزان تولید افزایش یابد. از دیدگاه دیگر در این مرحله می‌باید عملیات بهینه‌سازی<sup>(۲)</sup> را بر روی مسیرهای انتخابی اعمال نمود.

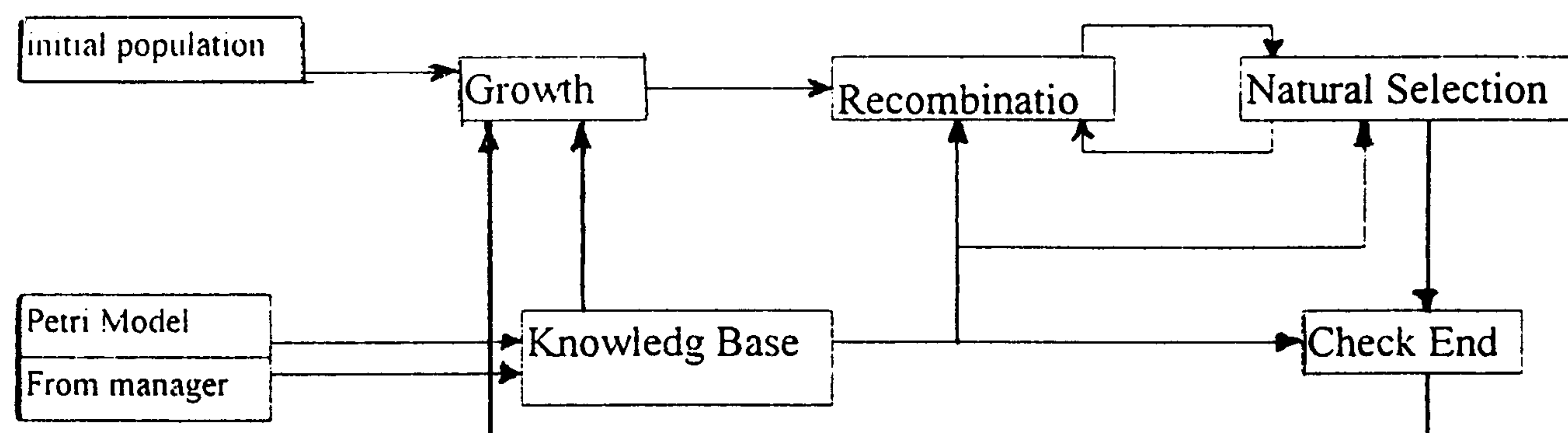
این ویژگی در فهرست‌بندی ما را ملزم می‌دارد که به دنبال یک روش کارآ و قوی در زمینه بهینه‌سازی باشیم که بتوانیم به شکلی مناسب معضلات مربوط به مسأله تطابق الگو را نیز به انجام رساند. به همین جهت در این مقاله برای پیاده‌سازی این واحد از یک الگوریتم ژنتیکی با خاصیت خود تنظیمی<sup>(۳)</sup> استفاده شده است و این به معنای آن است که برای هر قطعه، جمعیتی از مسیرها برای یافتن بهترین و مناسب‌ترین فهرست‌بندی‌ها انتخاب می‌شود اما عملیات جهش<sup>(۴)</sup> و همپوشانی<sup>(۵)</sup> مربوط به این جمعیت‌ها شدیداً متأثر از تصمیم شخصی افراد آن

انجام می‌گیرد.

به بیان دیگر در این روش سعی می‌شود با کمک اطلاعاتی که از مدل ساختاری سیستم و همچنین ترتیب نیازمندی دستگاه‌ها که از سوی طراح و یا مدیر تولید واحد صنعتی ارائه شده است یک پایگاه دانش<sup>(۶)</sup> ترتیب داده شود تا بتوان در طول فرآیند بهینه‌سازی با در دست داشتن این اطلاعات شکل و میزان عملیات نو ترکیبی<sup>(۷)</sup> و همچنین میزان و معیارهای انتخاب طبیعی را در حین جستجو به شکلی ترتیب داد تا مجموعه عملیات ضمن اعمال بهینه‌سازی، مسایل مربوط به تطابق الگو را نیز به نحو مطلوبی حل نماید. شکل (۱) واحدهای اصلی این الگوریتم را نشان می‌دهد.

فرمول‌بندی ژنتیکی و ارائه الگوریتم مربوط به واحد فهرست‌بند تک قطعه‌ای

به طور مسلم استفاده از الگوریتم‌های ژنتیکی در مورد یک کاربرد خاص نظیر زمینه مورد بحث ما، باعث خواهد شد تا علاوه بر این که در معنی و تعریف پارامترها و ابزار مورد استفاده در الگوریتم تغییراتی بوجود آید، غالباً کاربر را ملزم به ایجاد قابلیت‌های خاص و ویژه‌ای می‌کند تا بتوان با کمک آنها الگوریتم را برای کاربرد مورد نظر هر چه مناسب و منطبق‌تر نمود. این تعاریف و ویژگی‌های جدید برای کاربرد مورد نظر را می‌توان در قواعد زیر خلاصه نمود:



شکل ۱ - نمایش واحدهای اصلی الگوریتم ژنتیکی به کار گرفته شده به منظور پیاده‌سازی هسته مرکزی فهرست‌بندی، (این الگوریتم ژنتیکی دارای خاصیت خود تنظیمی است که این عمل را بر پایه معرفت موجود در پایگاه به انجام می‌رساند).



۱- هر کاراکتر در هر رشته (هر فرد از جمعیت) بیانگر یک گره و یک یال از شبکه می باشد و بنابراین هر فرد گویای یک مسیر برای انتقال مواد در شبکه می باشد.

۲- فهرست بندی برای هر قطعه همواره از یک گره آغاز شده و به یک گره پایان می پذیرد بنابراین فهرست بندی دارای نقاط آغاز و انتهای مشخصی می باشد.

۳- برای آن که بتوان با مسأله تطابق الگو هر چه بیشتر مقابله کرد و در واقع بتوان مسیرهای مناسب را هر چه زودتر به دست آورد، علاوه بر ویژگی (۴) که در زیر یاد شده است، در این کاربرد جمعیت مورد استفاده تحت دو زیر جمعیت مورد بهره برداری قرار می گیرد. یک بخش از این جمعیت حرکت خود را از گره آغازین شروع می کند که به آن جمعیت پیش رونده<sup>(۱)</sup> گفته می شود و جمعیت دیگر مسیر حرکت خود را از گره انتهایی به سمت گره ابتدایی در پیش می گیرد که به آن جمعیت پس رونده<sup>(۲)</sup> اطلاق می گردد. بنابراین، الگوریتم همچنین شامل پارامترهای کیفی است که می تواند جهت گیری افراد این جمعیت ها را به سمت یکدیگر ترغیب نماید. بدین ترتیب یک مسیر مناسب زمانی شکل خواهد گرفت که طی شرایط مجاز یکی از افراد جمعیت پیش رونده بتواند به یکی از افراد جمعیت پس رونده اتصال یابد.

۴- همان طور که در بخش اخیر یاد آور شد، الگوریتم شامل یک پایگاه معرفت کاملی است که به واحدهای دیگر در تعیین راه و نحوه اعمال تصمیم گیری در انتخاب طبیعی کمک فراوانی می نماید.

۵- محاسبه تابع تناسب<sup>(۳)</sup> براساس ترکیبی از زمان اجراء زمان انتظار، هزینه انتقالات و عملیات و در نهایت فاصله رشته تا گره هدف و یا یکی از افراد جمعیت مقابل انجام می گیرد. البته بسیار مناسب به نظر می رسد که بتوان طی آزمایشان متعدد و در یک محیط صنعتی، یک تابع هدف مناسب و کارآ که ترکیبی معین از پارامترهای مؤثر در عملکرد سیستم می باشد را معرفی نمود ولی متأسفانه این امر به علت عدم وجود امکانات و شرایط لازم به طور کامل محقق نگردیده است و در عمل ترکیب های متفاوتی از پارامترها به عنوان تابع هدف مورد بررسی بوده اند.

۶- در طی اجرای الگوریتم طول رشته ها ثابت نیست و به ازای اجرای عملیات نو ترکیبی برای چند بار یک مرحله

رشد رشته ها را به دنبال آن خواهیم داشت.

۷- شرایط پایان الگوریتم براساس تعداد نسل های ایجاد شده و رشته های حاوی مسیر مناسب بیان می گردد.

در اینجا ذکر چند نکته لازم و ضروری به نظر می رسد:  
الف- در این الگوریتم انتخاب جمعیت اولیه به صورت اتفاقی انجام نمی گیرد و در واقع تمامی افراد جمعیت به طور دقیق از همان ابتدا با مقادیر مشخصی مقداردهی می شوند (بند (۱) از الگوریتم شکل (۲)). به عبارتی همگی به گره های ابتدایی و یا انتهایی از مسیر اشاره خواهند کرد.

ب- در اینجا جهش به عنوان یک عامل گریز از بهینه های محلی مد نظر قرار گرفته است و در واقع این جهش تلاش می کند تا ما را در یافتن بهترین مسیر کمک کند.

ج- این الگوریتم عمل همپوشانی را برای هر جفت از افراد و در طول هر نسل تنها یکبار انجام می دهد و البته شاید در عمل مناسب بود که امکان همپوشانی چندتایی نیز به آنها داده می شد و بدین ترتیب شاید در رسیدن به بهینه مطلق موفق تر بودیم ولی به هر حال انجام این عمل زمان ببری بسیار را در حین فهرست بندی به خود اختصاص خواهد داد.

براساس ویژگی های ارائه شده در بالا می توان الگوریتم مورد بحث را مطابق شکل (۲) بیان نمود.

#### پیاده سازی واحد فهرست بندی قطعات انبوه

همان طور که گفته شد، هدف از این قسمت دستیابی به یک فهرست بندی است که در آن مسیر مجموعه تمامی قطعات به طور هم زمان و تنها به واسطه فهرست بندی منفرد تمام آنها حاصل می شود. به همین منظور این واحد مجبور خواهد بود نحوه اجرای عملیات فهرست بندی تمام تک قطعات را در طول زمان تحت بررسی قرار دهد و در هر لحظه تأثیرات این مسیرها را بر روی هم مورد مشاهده قرار دهد.

این تأثیرات خود شامل دو مسأله عمده یعنی تداخل<sup>(۴)</sup> و بن بست می باشند، مشکل اول باعث ایجاد افزایش زمان انتظار و بیکاری<sup>(۵)</sup> در سیستم می گردد و معضل دوم به طور کلی قسمتی و یا همه اجزا سیستم را از کار می اندازد و باعث از کار افتادن جریان انتقالات برای زمان نامحدود



۱- جمعیت‌های اولیه مربوط به افراد پیش‌رونده و پس‌رونده را انتخاب کن و به ترتیب آنها را به گره‌های آغازین انتهایی مقدار دهی نما و قرار بده  $I=0$

۲- این جمعیت‌ها را رد جهت‌های مناسب و تنها به اندازه یک طبقه (به اندازه یک گره) در جهت مقصد رشد بده به نحوی که برای این رشد گره‌ای را انتخاب نموده باشید که اولاً دارای حداقل یک مسیر برای رسیدن به گره هدف (برای جمعیت پیش‌رو گره انتهایی و برای جمعیت پس‌رو گره ابتدایی) باشد و ثانیاً در بین تمام گره‌ها، دارای هزینه و زمان عملیاتی کمتری باشد و در ضمن در پایان هر جستجو قرار بده  $I=I+1$

- حال مرحله‌های (۳) و (۴) را  $m$  بار تکرار کن :

۳- جهش را برای تمام افراد جمعیت‌های پیش‌رو و پس‌رو به ترتیب زیر انجام بده :

۲-۱- با احتمالی منطبق بر تابع توزیع احتمال  $f(x) = P \text{ mutation}^*(x-1)/(|^*|)$  که در آن  $x$  مکان هر کاراکتر از نقطه آغازین را مشخص می‌کند، یک کاراکتر را به عنوان کاراکتر جهش‌پذیر معرفی نما و قرار بده  $i=x$

۲-۳- اگر  $(i=1)$  به مرحله ۳ برو و اگر چنین نیست یک گره شانسی (Random) که البته دارای یک مسیر بین آن و گره  $i$  ام در رشته است پیدا کن (تنها با  $k$  بار جستجو) و در صورت وجود چنین گره‌ای به مرحله (۳-۳) برو و اگر یافت نشد به (۳) برگرد

۳-۳- رشته حاصله را به  $(I-i)$  بار صدا زدن به اندازه طبیعی رشد بده بنابراین رشته جدید به صورت زیر در می‌آید :

$$\text{New\_Sequence}(0, I) = \text{Old\_Sequence}(0, I-1) + \text{Random Node} + \text{Node}(i+1, I)$$

۴- همپوشانی (Crossover) را برای تمام افراد جمعیت‌های پیش‌رو و پس‌رو به ترتیب زیر انجام بده :

۴-۱- با احتمالی منطبق بر تابع توزیع احتمال  $f(x) = P \text{ mutation}^*(x-1)/(|^*|)$  که در آن  $x$  مکان هر کاراکتر از نقطه آغازین را مشخص می‌کند، یک کاراکتر را به عنوان کاراکتر جهش‌پذیر معرفی نما و قرار بده  $i=x$

۴-۲- اگر  $(i=1)$  به مرحله ۴ برو و اگر چنین نیست به طور شانسی یکی از افراد جمعیت نظیر را (برای پیش‌رو از افراد پیش‌رو و برای پس‌رو از افراد پس‌رو) انتخاب کن و در مرحله بعد به طور شانسی یکی از کاراکترها را با احتمال  $X/(|^*|)$  انتخاب کن و قرار بده  $j=x$  جال در صورتی که گره  $j$  با گره  $I-1$  مسیری دارد به مرحله (۴-۳) برو و اگر چنین نیست به مرحله ۴ برگرد.

۴-۳- حال رشته اول را با ترکیب قسمت قدیمی مربوط به خودش و قسمت جدید حاصل از فرد دوم به صورت زیر تکمیل کن.

$$\text{New\_Sequence}(0, I) = \text{Old\_Sequence}(0, I-1) + \text{Select\_Sequence}(I, I)$$

۵- در صورتی که تعداد نسل‌های ایجاد شده از مقدار ماکزیمم نسل‌های مجاز ( $O$ ) بیشتر نیست به (۲) برو و اگر چنین نیست ادامه بده :

۵-۱- اگر هنوز هیچ رشته‌ای منطبق با الگوی فعالیت خواسته شده به دست نیآورده بود، مسأله جواب نداد و به (۶) برو.

۵-۲- در صورتی که حداقل یک مسیر به دست آمده بود، کم هزینه‌ترین مسیر را به عنوان جواب اعلام کن.

۶- پایان

شکل ۲- الگوریتم فهرست‌بند تک قطعه‌ای به کمک الگوریتم ژنتیکی خود سازمانده.

### شبیه‌سازی و بحث

#### تعریف محیط آزمایش

برای انجام شبیه‌سازی‌ها از یک نمونه موردی مطابق شکل (۳) استفاده شده است و همان طور که مشاهده می‌شود این ساختار دارای ربات، AGV، انواع ماشین‌های CNC و نواحی تداخل برای AGV ها می‌باشد. برای مدلسازی نیز از دو شبکه پتری، یکی به عنوان مدل اصلی<sup>(۱)</sup>، شکل (۴)، و دیگری مدل واحدهای انتقالی<sup>(۲)</sup>،

خواهد شد. بنابراین برای مقابله با این مشکلات می‌باید ابتدا یک تعریف دقیق و ریاضی از آنها را بر روی مدل خود ارائه دهیم و سپس سعی نماییم راه‌حل‌هایی سریع برای آنها پیدا کنیم. در عمل نیز چنین کاری صورت گرفته است و نکته جالب آنکه الگوریتم‌هایی که در این زمینه ارائه شده است تنها دارای درجه محاسباتی  $(n^4)$  می‌باشد و در مقایسه با الگوریتم‌های کلاسیک و با درجه محاسباتی  $(n^n)$  بسیار کاراتر و سریع‌تر عمل می‌کنند. برای آگاهی بیشتر از جزئیات می‌توانید به [۱۵] رجوع نمایید.



البته مشخص است که اگر بخواهیم مسأله تداخل و تأثیرات متقابل را در نظر بگیریم (شیوه - ب - که در قسمت ۴ از آن یاد شد)، می باید اطلاعات بیشتری را در اختیار داشته باشیم:

جدول ۴ - نمایش زمان انتقال AGV2 بدون در نظر داشتن

## مسأله تداخل

AGV2	s1	s2	s3	s4
s1	_____	24	60	16
s2	24	_____	42	48
s3	60	42	_____	6
s4	16	48	6	_____

## نتایج شبیه سازی

در قسمت اول شبیه سازی ها مشکل بن بست مورد بررسی قرار گرفته است و برای این منظور دو راه حل بکار گرفته شده. در روش اول از یک روش پرهیز از بن بست بهره گرفته شده است و بنابراین در این قسمت تنها فهرست بندی مجزای قطعات مد نظر بوده و در روش دوم متد تصحیح و بازیابی بن بست<sup>(۲)</sup> مورد استفاده واقع شده است.

در روش اول فهرست بندی مستقل هر کدام از قطعات به منظور اجرا به کنترل مرکزی داده می شود و کنترل مرکزی در حین اجرای عملیات الگوریتمی را بر روی نحوه تخصیص دستگاهها اعمال می دارد و بدین ترتیب با متوقف کردن بعضی از فعالیتها جلوی پدید آمدن بن بست را خواهد گرفت. اما متأسفانه اعمال چنین الگوریتمی بهره وری واحد کنترل مرکزی را در این حالت تا حدود ۷۲٪ کاهش می دهد.

اما در روش دوم مشکل بن بست در حین عمل فهرست بندی حل می شود و در واقع پس از آن که فهرست بندی مستقل هر کدام از قطعات به دست آمد، وجود بن بست در اجرای هم زمان آنها مورد بررسی قرار

شکل (۵)، استفاده شده است.

در این سیستم قصد آن داریم که دو فعالیت<sup>(۱)</sup> با طرح ریزی مطابق با جدول (۱) را پیاده سازی نماییم.

جدول ۱ - نمایش فعالیت های موجود در F.M.S

Job	Type	Op1	Op2	Op3
J1	1	M3	M2/M1	M2
J2	2	M1	M2	M1

در این جدول  $M_i$  ها نمایگر ماشین ها و  $O_{pi}$  ها بیانگر مراحل کاری بر روی هر قطعه می باشد. همچنین زمان اجرای عملیات  $i$ ام، روی قطعه  $k$ ام به صورت زیر است:

جدول ۲ - نمایش زمان عملیات بر روی هر قطعه و توسط هر ماشین.

Operation	Operation Time
OP1,1,1	81
OP2,1,2	20
OP2,1,1	30
OP3,1,2	60
OP1,2,1	128
OP2,2,2	64
OP3,2,3	78

جدول ۳ - نمایش زمان انتقال AGV1 بدون در نظر داشتن

## مسأله تداخل

AGV1	s1	s2	s3	s4
s1	_____	12	30	8
s2	12	_____	21	24
s3	30	21	_____	3
s4	8	24	3	_____

و زمان انتقالات AGV ها بدون در نظر داشتن مسأله تداخل در بهترین حالت، مطابق با جدول ۳ و ۴ می باشد.

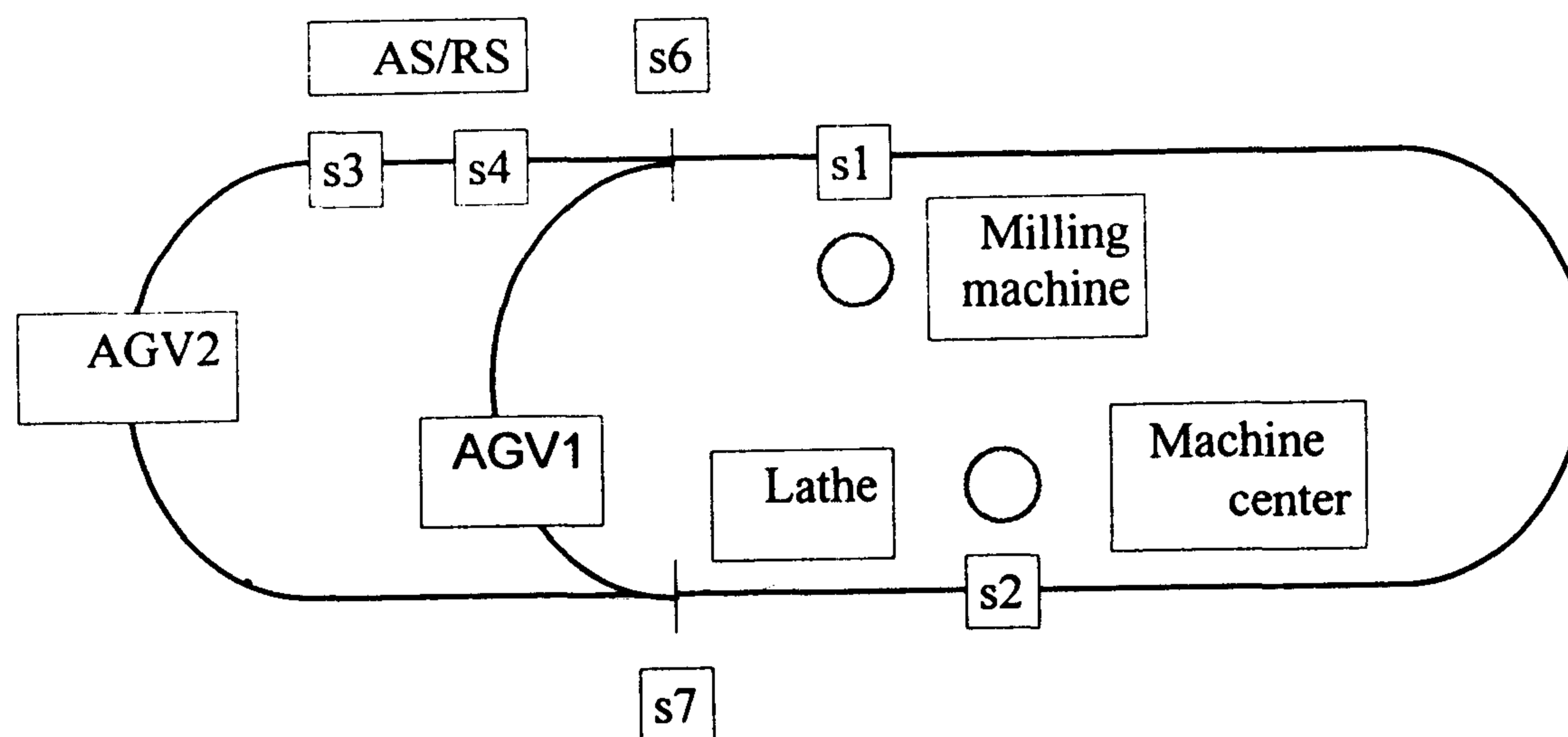


جدول ۵ - مقایسه دو روش در حل مشکل بن بست در فهرست بندی.

Metod No	Time Cycle	Cost	Idle Time	Performance	Computation Time
1	328 sec	12	109 sec	%67	80 sec
2	248 sec	15	17 sec	%93	450 sec

جدول ۶ - مقایسه روش ژنتیکی خود سازمانده و روش ژنتیکی ساده در انجام عمل فهرست بندی

Metod	Computation Time	Time Cycle	Cost
Normal GA	1800 sec	328 sec	15
Self-Organization GA	450 sec	328 sec	15



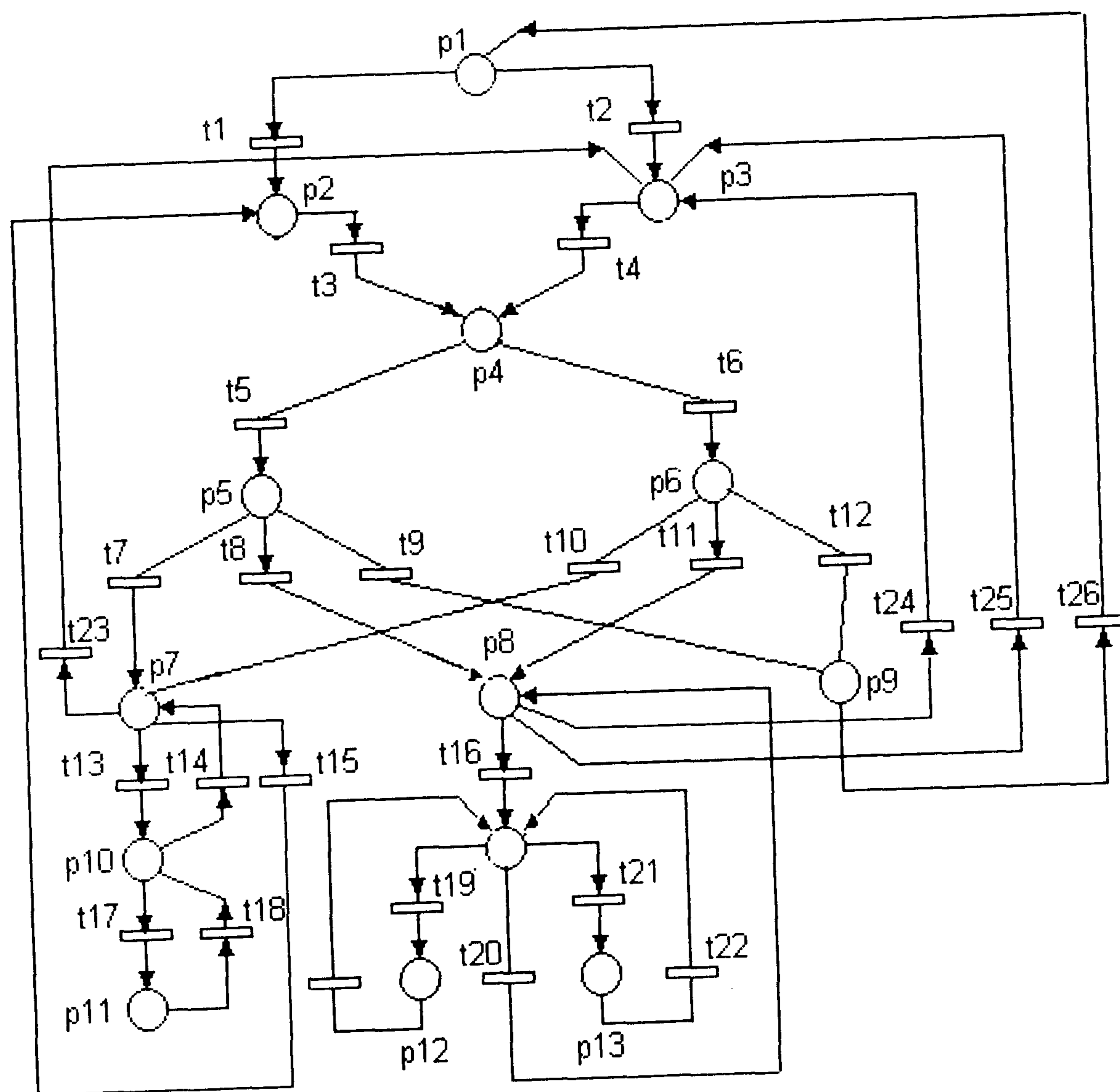
شکل ۳ - نمایش ساختار F.M.S مورد استفاده در شبیه سازی.

روش زمان اجرای فهرست بندی شدیداً افزایش می یابد (جدول ۵).

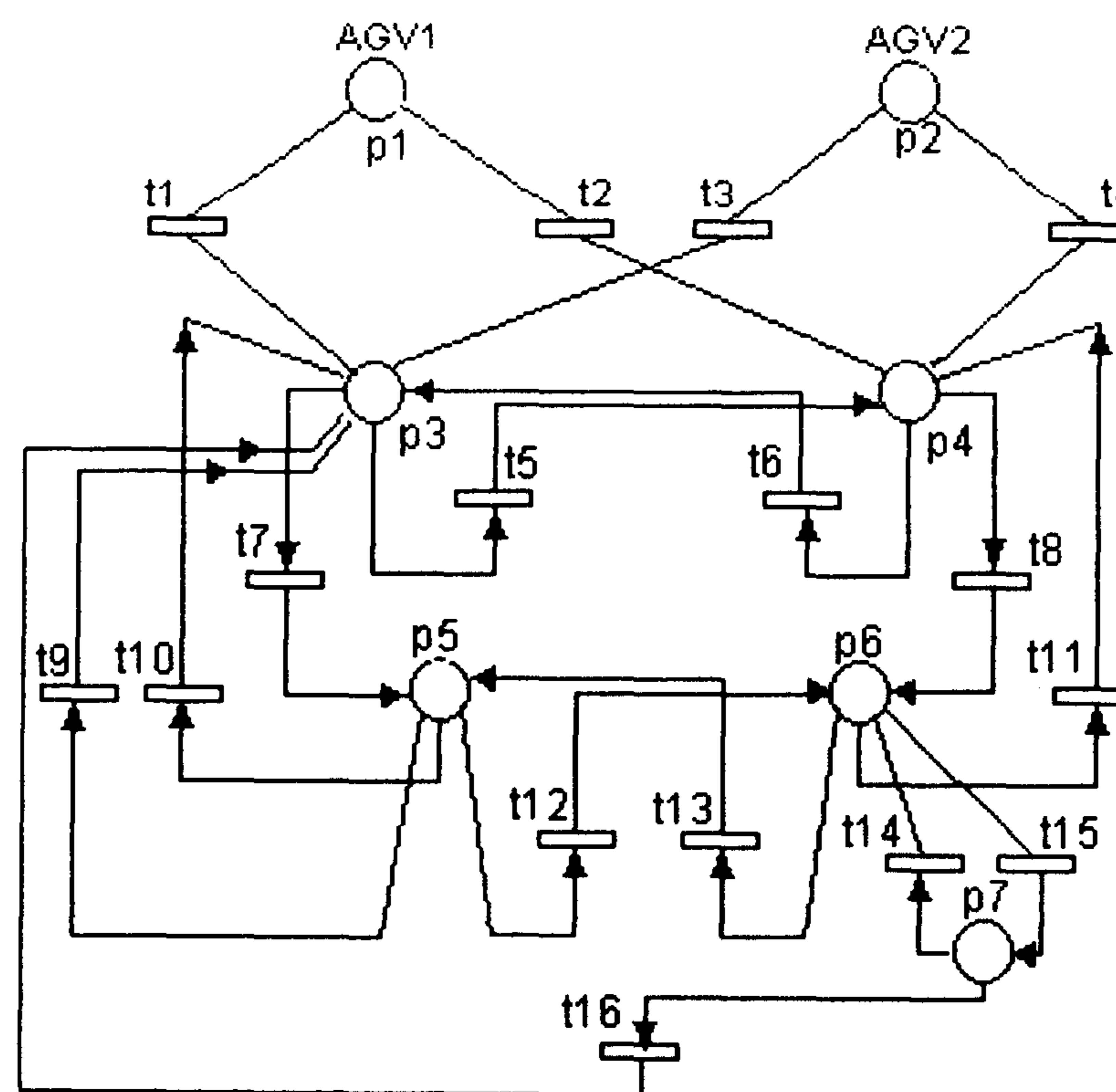
در قسمت دوم شبیه سازی ها نیز نحوه عملکرد روش جستجو مورد بررسی قرار گرفته است و به این منظور روش پیشنهادی که یک روش ژنتیکی خود سازمانده می باشد را در مقابل یک روش ژنتیکی ساده به معرض مقایسه قرار داده ایم که نتایج آن را می توان در جدول ۶ جستجو نمود.

می گیرد و راه حل هایی برای حل آن ارایه می شود. به عنوان نمونه اگر فهرست بند مستقلی که مرکزی برای دو قطعه ذکر شده در مثال بالا پیشنهاد نموده است را بر روی سیستم و بدون اعمال الگوریتم پرهیز از بن بست اجرا کنیم، زمانی که قطعه اول در M1 و قطعه دوم در M2 است یک بن بست پدید خواهد آمد در عین حال استفاده از روش دوم پیشنهاد می کند که در این حالت قطعه ۱ به s2 برود و بدین ترتیب با تغییر مسیر فهرست بندی مشکل بن بست به طور کلی منتفی می گردد. البته همان طور که مشخص است در این





شکل ۴ - نمایش عملکرد گره‌های اصلی سیستم ارایه شده در شکل (۳) به واسطه شبکه پتری PNF.



شکل ۵ - نمایش عملکرد گره‌های انتقال دهنده در سیستم ارایه شده در شکل (۳) توسط شبکه پتری PNF.



## نتیجه گیری

می شود را نیز پیش بینی می نماید و به ازای هر برخورد راه حل مناسبی در پیش می گیرد.

واحد فهرست بندی ارایه شده، خود از دو قسمت مجزا تشکیل یافته. واحد اول که وظیفه فهرست بندی تک قطعات را برعهده دارد از یک الگوریتم ژنتیکی خود سازمانده استفاده می نماید که به همراه اعمال ترفندهای خاص دیگر می تواند مسأله بهینه سازی و تطابق الگو را در مسیریابی قطعات به خوبی انجام دهد.

واحد دوم نیز که براساس یک سری از نتایج ریاضی حاصل از مدل، بنا شده است به همراه یک بخش شبیه ساز، می تواند طی یک شبیه سازی دقیق در تمام لحظات، شکل برخوردها را در سیستم بازشناسی نماید و به ازای هر کدام سیاست خاصی را برای رفع آن اجرا نماید. نتیجه عملی این قسمت از تحقیق تا ارایه یک روش کلی و البته سریع برای فهرست بندی مجموعه ای از قطعات در هر سیستم دلخواهی می باشد که می تواند توسط شبکه پتری PNF مدل سازی شود.

بنابراین در نهایت این مقاله سعی نموده است تا یک روش فهرست بندی بهینه و البته به صورت Off Line را ارایه دهد.

در پایان نیز ضمن تشکر از نظرات ارزنده داوران محترم جهت پر بار نمودن نکات علمی این مقاله، امید است در آینده با استفاده از این نظرات در تحقیقات آتی نتایج قابل توجهی حاصل گردد.

مقاله حاضر توجه خود را بر روی دو موضوع اساسی در حوزه سیستم های انعطاف پذیر، یعنی مدلسازی و فهرست بندی این سیستم ها معطوف نموده است. بنابراین نتایج حاصل از این بررسی را می توان در دو قسمت به شکل زیر بیان نمود؛

۱- در حوزه مدلسازی، مقاله دست به ارایه یک مدل جدید براساس شبکه های پتری برای مدلسازی سیستم های انعطاف پذیر نموده است. این شبکه قادر است تمامی اجزا و لوازمات یک سیستم FMS نظیر ماشین های CNC، ربات ها، AGV ها، بافرها و همچنین مسایلی نظیر تداخل مسیر حرکت AGV ها و یا تداخل حرکت ربات ها را به طور کامل و دقیق تعریف نماید. بنابراین محصول این بخش از تحقیق ارایه یک مدل کارا و عملی است که می تواند سیستم های انعطاف پذیر واقعی را به طور کامل مدل سازی نماید.

۲- در بحث فهرست بندی نیز برخلاف بسیاری از کارهایی که در این حوزه انجام شده است، فهرست بندی ارایه شده، امکان مسیریابی مجموعه انبوهی از قطعات (حالت Lot Size) را داراست و قادر می باشد تا فهرست بندی مجموعه متنابهی از انواع قطعات و با تعداد زیاد را به طور مجزا و دقیق برای هر کدام انجام دهد. در این راه نیز علاوه بر ارایه یک فهرست بندی بهینه از نظر هزینه و زمان اجرا، تمامی تداخل ها و بن بست ها که در سیستم باعث ایجاد تلفات

## مراجع

- 1 - Daeyoung Chung, and Chankwon Park. (1996). "Developing a shop floor scheduling and control software for an FMS." *Computer Ind. Engineering*, 30(3), 557-567.
- 2 - Hu, G. H., Wang, Y. S. and Ioh, H. T. (1995). "An FMS scheduling and control decision support based on generalized stochastic petri net." *Int. J. Adv. Manuf. Technol.*, 10(1), 52-58.
- 3 - Paul Fredman. (1991). "Time, petri nets and robotics." *IEEE Transaction on Robotics and Automation*, 7(4).
- 4 - Rober-Y, AL-JAAR, ALANA. Desorchers. (1990). "Performance evaluation of automated manufacturing system



- using generalized stochastic petri nets." *IEEE Transaction on Robotics and Automation*, 6(6), 621-639.
- 5 - Zhou, Q. Wang, M. (1995). "Generation of optimal control policy for flexible manufacturing cells : A petri net approach." *Int. J. Adv. Manuf. Technol.*, 10(1), 59-65.
- 6 - Fujimoto, H. Yasuda, K. (1995). "Applications of genetic algorithm and simulation to dispatching rule-based FMS scheduling." *IEEE International Conference on Robotics and Automation*, (1), 195.
- 7 - Huang, B. and Zhang, B. (1994). "A new scheduling model based on extended petri net-TERM net." *IEEE International Conference on Robotics and Automation*, (1), 495-500.
- 8 - Jong-Kun, L. (1996). "Time composition rules for time petri nets." *The Fourth International Conference on ICVRCV*, (2), 714-718.
- 9 - Fanti, M. P. (1993). "Two methods for real-time routing selection in flexible manufacturing system." *International Conference on Robotics and Automation*, (2), 1158.
- 10 - Cheng, C. W. (1994). "Petri-Net based modeling and scheduling of a flexible manufacturing system." *International Conference on Robotics and Automation*, 513.
- 11 - Kimm, G. H. and Lee, G. S. G. (1995). "Genetic reinforcement learning approach to the machine scheduling problem." *International Conference on Robotics and Automation*, (1), 195.
- 12 - Amar, A. D. (1997). "Costs in design of scheduling algorithms : A study on branch-and-bound methodology." *Computer Ind. Engng*, 32(1), 129-138.
- 13 - Viswanadham, N. and Narahari, Y. (1990). "Deadlock prevention and deadlock avoidance in flexible manufacturing system using petri net models." *IEEE Transaction on Robotics and Automation*, 6(6).
- 14 - Wysk, R. A., Yang, N. S. and Joshi, S. (1991). "Detection of deadlock in flexible manufacturing cells." *IEEE Transaction on Robotics and Automation*, 7(6).

۱۵ - هادی زاده، م. "بهبود رفتار مدیریتی سیستم‌های انعطاف پذیر تولید به کمک روش‌های هوشمند ترکیبی." پایان نامه کارشناسی ارشد، دانشگاه تهران، دانشکده فنی، گروه مهندسی برق و کامپیوتر، (۱۳۷۶).

	کلیدواژه‌ها
1 - Petri Net	شبکه پتری
2 - Scheduling	فهرست بندی
3 - Automated Guided Vehicle	حامل های بار
4 - Buffers	نگهدارنده ها
5 - Sub-optimal	نزدیک به بهینه
6 - Machine Scheduling Problem	حالت پیچیده ای از مساله کلی
7 - Simulating Arealing	آب دادن فولاد
8 - Reusable	استفاده مجدد
9 - Load / Unload Stations	ایستگاههای بارگذاری و تخلیه



---

10 - Petri Net based on Function	شکل جدیدی از شبکه پتری
11 - Link	یال اتصال
12 - Enabled Transition	انتقال مجاز
13 - Deadlock Avoidance	پرهیز از بن بست
14 - Banker's Algorithm	الگوریتم بانکدار
15 - Self Organization Genetic Algorithm	الگوریتم ژنتیکی خود سازمانده
16 - Lot Size	انبوهی از قطعات
17 - Pattern Matching	تطابق الگو
18 - Self Organization	خاصیت خودتنظیمی
19 - Mutation	جهش
20 - Crrossover	همپوشانی
21 - Knowledge Base	پایگاه دانش
22 - Recombination	عملیات نوترکیبی
23 - Idle Time	زمان انتظار و بیکاری
24 - Process Flow Model	عنوان مدل اصلی
25 - Transition Model	مدل واحدهای انتقالی
26 - Deadlock Recovery	تصحیح و بازیابی بن بست

---