



مرکز ملی باوردهای علمی و فناوری

سازمان بنادر و دریانوردی به عنوان تنها مرجع حاکمیتی کشور در امور بندری، دریایی و کشتی‌رانی بازرگانی به منظور ایفای نقش مرجعیت دانشی خود و در راستای تحقق راهبردهای کلان نقشه جامع علمی کشور مبنی بر "حمایت از توسعه شبکه‌های تحقیقاتی و تسهیل انتقال و انتشار دانش و سامان‌دهی علمی" از طریق "استانداردسازی و اصلاح فرایندهای تولید، ثبت، داوری و سنجش و ایجاد بانک‌های اطلاعاتی یکپارچه برای نشریات، اختراعات و اکتشافات پژوهشگران"، اقدام به ارایه این اثر در سایت SID می‌نماید.



سازمان بنادر و دریانوردی

ارایه یک الگوریتم کارا برای حل یک مدل ریاضی ارایه شده برای مساله

زمان بندی گنتری کرین ها در یک ترمینال کانتینری

محمد بزازی، کارشناس ارشد مهندسی صنایع

چکیده:

در این مقاله یک مدل برنامه ریزی صفر و یک برای مساله تخصیص و زمان بندی گنتری کرین ها که به نام QCSAP نامیده می شود، در یک ترمینال کانتینری ارایه شده است. به دست آوردن یک حل بهینه برای این گونه مسایل پیچیده در یک زمان حل قابل قبول، بسیار مشکل و به تقریب غیر ممکن است. به این دلیل یک الگوریتم ژنتیک برای مساله QCSAP ارایه شده است تا بتوان مسایل دنیای واقعی را به این طریق حل نمود. در ادامه کارایی الگوریتم ارایه شده با حل قطعی که توسط نرم افزار LINGO8 به دست آمده است (در ابعاد کوچک) مقایسه می شود که نتایج حاکی از آن است که الگوریتم توسعه داده شده کارا می باشد.

کلمات کلیدی: زمان بندی گنتری کرین ها، مساله تخصیص، بنادر کانتینری، الگوریتم

ژنتیک

۱- مقدمه :

استفاده از کانتینرها در حمل و نقل دریایی در خلال ۱۰ سال گذشته به سرعت در حال افزایش است. از این رو ترمینال‌های کانتینری که محل تخلیه و بارگیری کانتینرها از/ به کشتی‌ها می‌باشند، نقش اساسی را در صنعت حمل و نقل دریایی ایفا می‌نمایند. به طور کلی بهره‌وری یک ترمینال کانتینری را می‌توان توسط دو فاکتور نشان داد: (۱) سرعت و کیفیت عملیات تخلیه و بارگیری بر کشتی‌ها (۲) سرعت و کیفیت عملیات دریافت و تحویل کانتینرها توسط بارکش‌ها (ورود و خروج کانتینرها از/به ترمینال کانتینری). فرآیند برنامه‌ریزی عملیات کشتی‌ها شامل: برنامه‌ریزی و تخصیص اسکله، تخصیص و زمان‌بندی گنتری کرین، تخلیه و بارگیری می‌باشد. در جریان فرایند برنامه‌ریزی اسکله، زمان پهلوگیری و موقعیت پهلوگیری یک کشتی کانتینری در یک اسکله باید تعیین شود. در مساله زمان‌بندی QC توالی تخلیه و بارگیری کشتی‌ها توسط هر گنتری و هم چنین زمان‌بندی اجرای کارها تعیین می‌شود. اطلاعات ورودی برای این مساله شامل برنامه ذخیره‌سازی برای هر کشتی، زمان آماده‌سازی هر QC و یک نقشه از محوطه نگهداری کانتینرها جهت تعیین موقعیت ذخیره‌سازی هر کانتینر. سرانجام، در طول فرایند تخلیه و بارگیری، توالی تخلیه و بارگیری هر کانتینر بر اساس برنامه ارایه شده تعیین می‌شود. این مقاله روی مساله QCSAP تمرکز داشته که به قسمت دوم برنامه زمان‌بندی کشتی‌ها در ارتباط قرار می‌گیرد.

در ابتدا باید این نکته را در نظر گرفت که هدف اصلی مساله زمان‌بندی QC، مینیموم نمودن زمان تکمیل آخرین کار می‌باشد. کیم و پارک (2004) یک مدل ریاضی را برای مساله زمان‌بندی QC ارایه نمود. در این تحقیق ما مدل آن‌ها را با افزودن تخصیص QC توسعه داده و آن را QCSAP می‌نامیم. در این مدل فرض

می‌شود (به مانند بسیاری از بنادر دنیا) که جهت تاخیر در عملیات تخلیه و بارگیری مبلغی به‌عنوان جریمه که دموراژ نامیده می‌شود باید از طرف بندر به مشتریان (کشتی‌ها) پرداخت شود. هم‌چنین برای تحویل زودتر از موقع کار مبلغی به‌عنوان پاداش که دیسپاچ نامیده می‌شود از طرف مشتریان (کشتی‌ها) دریافت می‌شود. لازم به ذکر است که زمان اتمام آخرین کار روی کشتی‌ها برابر زمان تکمیل آخرین کاری است که بر کشتی‌ها انجام می‌شود. این زمان یک نقطه بسیار قابل اهمیت در بهره‌وری ترمینال است که هدف ما در این تحقیق کاهش این نقطه می‌باشد. در این تحقیق یک مدل ترکیبی برنامه‌ریزی صفرویک برای مساله فوق جهت افزایش بهره‌وری ترمینال کانتینری ارائه شده است.

به طور کلی در عملیات ترمینال‌های کانتینری مسایل تصمیم‌گیری متفاوتی مطرح می‌باشند از قبیل تخصیص اسکله، تخصیص فضای ذخیره‌سازی، زمان‌بندی گنتری کرین‌ها، زمان‌بندی جرثقیل‌های محوطه. تمامی این مسایل تصمیم‌گیری بر یکدیگر تاثیر می‌گذارند. مساله زمان‌بندی گنتری کرین‌ها یکی از مهم‌ترین این نوع مسایل است. هدف اصلی این مساله تعیین توالی عملیات‌های تخلیه و بارگیری می‌باشد به نحوی که زمان تکمیل عملیات روی کشتی مینیموم شود، لذا این نوع مسایل به مانند مسایل کلاسیک پارالل ماشین می‌باشند. اما مساله زمان‌بندی گنتری‌ها دارای خصوصیات منحصر به فردی است که آنرا از مسایل پارالل ماشین متفاوت می‌سازد. به عنوان مثال، عملیات تخلیه و بارگیری ممکن است به طور هم زمان روی یک Ship-Bay صورت گیرد. هم‌چنین زمانی که عملیات تخلیه در حال انجام می‌باشد کانتینرهای بر روی Deck باید قبل از کانتینرهایی که در Hold می‌باشند تخلیه شوند. بنابراین روابط پیش‌نیازی زیادی مابین فعالیت‌های مورد نظر وجود دارد که باید رعایت شود.

هم چنین از آن جایی که باید بین گنتری کرین ها یک حداقل فاصله برابر با یک Ship-Bay وجود داشته باشد لذا در یک زمان دوگنتری نمی توانند در منطقه ای که بسیار به هم نزدیک می باشند به طور هم زمان عملیات انجام دهند. از این رو در مساله فوق محدودیت هم زمانی نیز داریم. یعنی تعدادی کار نمی توانند به طور هم زمان انجام شوند. این دو محدودیت اصلی است که باید در مدل فوق مد نظر قرار گیرد.

۲- ادبیات موضوع :

ادبیات موضوع در زمینه ترمینال های کانتینری را می توان به صورت مسایل زیر طبقه بندی نمود :

مسایل مربوط به تخلیه و بارگیری از/ به کشتی ها- حمل و نقل کانتینرها در فضای ذخیره سازی- نخستین و یکی از مهم ترین مسایل، مساله تخصیص اسکله به کشتی ها می باشد. ایمایی و نگایوا (۱۹۹۷) یک مدل را برای می نمی م کردن مجموع زمان ایست کشتی ها در بندر جهت می نمی م کردن میزان عدم رضایت کشتی ها ارائه نمود. ایمایی و نیشیمورا و پاپادیمیتریو (۲۰۰۱) یک مدل دینامیک و استاتیک تخصیص اسکله را توسعه دادند. آن ها یک روش ابتکاری را برای حل مساله بر اساس یکنواخت سازی لاگرانژین ارائه نمودند. هم چنین آن ها در تحقیق دیگری یک مدل دینامیک برای مساله ای که دارای محدودیت های مختلف فیزیکی جهت پهلوگیری کشتی ها بود ارائه نمودند. هم چنین برای حل آن از یک الگوریتم ژنتیک استفاده نمودند. ایمایی و پاپادیمیتریو (۲۰۰۳) اولویت های خدماتی کشتی را در یک مساله تخصیص اسکله به کشتی ها تعیین نمودند. آن ها در ابتدا از یک متعادل سازی لاگرانژین استفاده نمودند، اما آن ها دریافتند که مساله متعادل سازی فوق به یک مساله QAP تبدیل شده است که

حل آن بسیار سخت می‌باشد لذا یک الگوریتم ژنتیک برای حل آن ارائه نمودند.

کیم و پارک (۲۰۰۴) یک مدل ریاضی را برای مساله زمان‌بندی QC ارائه نمودند. آن‌ها یک روش شاخه و حد را برای به دست آوردن حل بهینه به همراه یک روش حل ابتکاری که به روش GRASP معروف است ارائه نمودند. دگانزو (۱۹۸۹) اولین کسی بود که بیان نمود محدودیت طول کشتی‌ها را باید در مساله زمان‌بندی گنتری‌ها در نظر گرفت. او یک الگوریتم را برای تعیین تعداد گنتری‌هایی که به یک کشتی تخصیص می‌یابد، ارائه نمود.

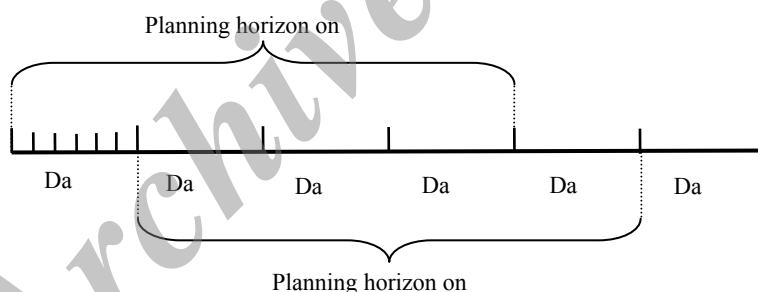
لی و کای (۱۹۹۸) این مسایل را به صورت مساله زمان‌بندی اسکله در نظر گرفتند که می‌تواند هم زمان کارهای بر روی کشتی‌های مختلف را انجام دهد. آن‌ها الگوریتم‌هایی را بر اساس روش‌های ابتکاری متفاوت برای این منظور ارائه دادند. کورادو (۲۰۰۵) در مساله زمان‌بندی گنتری‌ها، تابع هدف مساله را مینیوم ساختن زمان تکمیل کشتی‌ها و زمان بی‌کاری گنتری‌ها در نظر گرفت. وانگ و میاوو (۲۰۰۶) مساله زمان‌بندی گنتری‌ها را با محدودیت‌های عدم تداخل در نظر گرفت و یک الگوریتم ژنتیک برای آن ارائه داد. جانگ و کیم (۲۰۰۶) یک الگوریتم ژنتیک و شبیه‌سازی ذوب فلزات را برای زمان‌بندی عملیات بارگیری زمانی که چندین جرثقیل در حال عملیات روی یک کشتی می‌باشند ارائه نمود. کیا و شایان (۲۰۰۳) یک مدل شبیه‌سازی را برای تخمین و ارزیابی عملکرد یک ترمینال کانتینری با هدف متعادل‌سازی تجهیزات بندری و ظرفیت ترمینال ارائه نمودند.

ادامه مقاله به این صورت سازماندهی شده است : مساله زمان‌بندی گنتری‌کرین‌های توسعه داده شده در فصل ۲ ارائه می‌شود. الگوریتم ژنتیک توسعه داده شده در فصل

سوم ارایه شده است. نتایج محاسباتی در فصل ۴ و نتیجه گیری در فصل ۵ ارایه می شود.

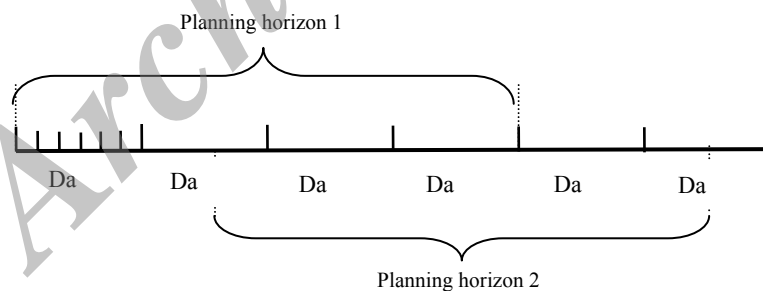
۳- مدل ریاضی توسعه داده شده (بیان مساله)

به طور کلی می توان مساله را به این شرح توصیف نمود : کشتی ها می توانند به طور هم زمان توسط چندین گنتری مورد پردازش قرار گیرند. زمان پردازش یک کشتی اگر یک تخصیص نامناسب گنتری به کشتی ها داشته باشیم، افزایش خواهد یافت. این مساله در زمان اتمام کار کشتی تاثیر مستقیم دارد. از طرفی نمی توانیم تعداد زیادی گنتری را به کشتی ها تخصیص دهیم چراکه محدودیت تعداد گنتری و عدم تداخل گنتری ها را داریم. از این رو در این تحقیق، ما تخصیص بهینه و زمان بندی بهینه جهت گنتری های تخصیص یافته به کشتی ها را در هر افق زمانی مد نظر قرار داده ایم. ژانگ و همکاران (۲۰۰۳) یک افق زمانی ثابت را برای مساله تخصیص فضای ذخیره سازی به کانتینرها به مانند شکل ۱ نشان دادند.



در بسیاری از حالات واقعی، ما با شرایط داینامیک روبرو می باشیم به نحوی که در یک بندر به طور دایمی کشتی ها در حال ورود و خروج می باشند. ما نمی توانیم فرض

کنیم که تعداد کشتی‌های فعال ثابت است زیرا بسیاری از کشتی‌ها پس از فرایندهای تخلیه و بارگیری بندر را به طول یک افق برنامه‌ریزی ترک می‌کنند و کشتی‌های دیگری جایگزین آن‌ها می‌شوند. بنابراین، نمی‌توانیم از یک برنامه زمان‌بندی ثابت برای افق زمانی خود استفاده نماییم. این برنامه فرض می‌کند که تمامی کشتی‌ها بندر را در ابتدا و انتهای افق برنامه‌ریزی ترک کرده و یا وارد می‌شوند. اما شرایط ذکر شده هرگز در شرایط واقعی اتفاق نمی‌افتد. در این حالت، در این تحقیق یک افق برنامه‌ریزی جدید (چنان که در شکل ۲ نشان داده شده است) ارائه شده است. در شکل ۲ افق زمانی ۲ بعد از ۱.۵ روز از افق زمانی ۱ و پس از ورود یک کشتی جدید به بندر آغاز می‌شود. اما افق زمانی ۳ پس از ۲ روز از افق زمانی ۲ و پس از ورود یک کشتی دیگر آغاز می‌شود. به عبارت دیگر زمان آغاز یک افق زمانی بر اساس ورود یا خروج یک کشتی از بندر می‌باشد. بنابراین، پس از آغاز هر افق زمانی جدید، یک گنتری که به کشتی A تخصیص یافته است، ممکن است در افق جدید به کشتی B تخصیص یابد (البته پس از تکمیل کلیه کارها روی کشتی A).



در زیر مدل برنامه‌ریزی صفرویک ارایه شده برای مساله QCSAP ارایه می‌شود. در این مدل گنتری‌ها به کشتی‌ها تخصیص یافته و توالی عملیات تخلیه و بارگیری گنتری‌ها روی کشتی‌ها برای هر افق زمانی تعیین می‌شود.

۲-۱- مفروضات

لازم به دقت است که برخی از مفروضات ارایه شده در زیر ممکن است با شرایط واقعی متفاوت باشد. ولی با علم به این موضوع و جهت بررسی امکان‌پذیری و توجیه‌پذیری چارچوب کلی ارایه شده آن‌ها را به طور موقت در نظر می‌گیریم:

موقعیت‌های پهلوگیری و هم‌چنین بار کاری کشتی‌ها به عنوان یک ورودی به مدل وارد می‌شود.

اطاعات مربوط به سرعت و نرخ تخلیه و بارگیری تمامی گنتری‌ها معین می‌باشد که این مهم به عنوان یک ورودی به مساله افزوده می‌شود.

فاصله ایمنی بین یک جفت گنتری تعیین شده و مد نظر قرار می‌گیرد. بنابراین تنها یک گنتری می‌تواند روی یک Ship-Bay در یک زمان فعالیت نماید.

هر کشتی به تعدادی Ship-Bay تقسیم می‌شود، هر Ship-Bay نیز به تعدادی ردیف کانتینر . Ship-Bay ها در هر کشتی دارای طول یکسانی می‌باشند. بنابراین طول یک کشتی برابر تعداد کل Ship-Bay در طول یکی از آن‌ها می‌باشد.

۲-۲- پارامترهای ورودی

K تعداد گنتری‌ها. ($K=1, 2, \dots, K$).

M تعداد کشتی‌ها ($m=1, 2, \dots, M$).

N_m تعداد فعالیت‌های روی هر کشتی (گنتری‌هایی که باید تخلیه یا بارگیری شود).

P_j^m زمان مورد نیاز برای انجام کار i روی کشتی m .

r_k اولین زمان در دسترس گنتری K ام.

l_i^m موقعیت کار i بر اساس شماره Ship-Bay روی کشتی m .

t_{ij}^{km} زمان سفر k امین QC از موقعیت l_i^m از کار i به موقعیت l_j^m از کار j روی کشتی m . t_{0j}^{km} زمان سفر از اولین موقعیت (l_k^m) از k امین QC به موقعیت (l_j^m) از کار j بر روی کشتی m . بعلاوه، t_{iT}^{km} نشان‌دهنده زمان سفر از موقعیت (l_j^m) کار j به آخرین مقصد (l_k^m) از k امین QC بر روی کشتی m می‌باشد.

H_k هزینه متغیر استفاده از K امین QC.

S_k هزینه ثابت استفاده از k امین QC.

W_m هزینه دیرکرد کشتی m .

V_m درآمد ناشی از زودکرد کشتی m .

Ψ مجموعه جفت کارهایی که نمی‌تواند به طور هم زمان انجام شوند.

Φ مجموعه جفت کارهایی که مابین آن‌ها روابط پیش‌نیازی وجود دارد.

M یک عدد مثبت بسیار بزرگ.

۲-۳- متغیرهای تصمیم

اگر k امین QC کار Z را بلافاصله پس از کار i روی کشتی m انجام

$$\left. \begin{array}{l} 1 \\ 0 \end{array} \right\} X_{ij}^{km}$$
در غیر این صورت.

اگر زمان شروع کار Z دیرتر از زمان تکمیل کار i روی کشتی m باشد.

$$\left. \begin{array}{l} 1 \\ 0 \end{array} \right\} Z_{ij}^m$$
در غیر این صورت.

اگر k امین QC به کشتی m تخصیص یابد.

$$\left. \begin{array}{l} 1 \\ 0 \end{array} \right\} Y_{km}$$
در غیر این صورت

T_m زمان دیرکرد کشتی m

E_m زمان زودکرد کشتی m

Q_k زمان تکمیل k امین QC.

C_m زمان تکمیل کشتی m

D_i^m زمان تکمیل کار i روی کشتی m

۲-۴- مدل ریاضی

$$\min Z = \sum_{m=1}^M \left(H_k Q_k + S_k \sum_{m=1}^M Y_{km} \right) + \sum_{m=1}^M W_m T_m - \sum V_m E_m$$

(1)

s.t.

$$E_m - T_m = \lambda * \sum_{i=1}^{N_m} P_i^m - C_m \quad \forall m, k$$

(2)

$$\sum_{j=1}^{N_m} X_{0j}^{km} = Y_{km} \quad \forall m, k$$

(3)

$$\sum_{j=1}^{N_m} X_{j(N_m+1)}^{km} = Y_{km} \quad \forall m, k$$

(4)

$$\sum_{k=1}^K \sum_{i=0}^{N_m} X_{ij}^{km} = 1 \quad \forall j ; j \neq i ; \forall m$$

(5)

$$\sum_{i=0}^{N_m} \sum_{\substack{j=1, \\ j \neq i}}^{N_m} X_{ij}^{km} - MY_{km} \leq 0 \quad \forall m, k$$

(6)

$$\sum_{\substack{j=1 \\ j \neq i}}^{Nm+1} X_{ij}^{km} - \sum_{\substack{j=1 \\ j \neq i}}^{N_m} X_{ji}^{km} = 0 \quad \forall i, m, k$$

(7)

$$D_i^m + t_{ij}^{km} + p_j^m - D_j^m \leq M(1 - X_{ij}^{km}) \quad \forall i, j ; j \neq i ;$$

$$\forall m, k \quad (8)$$

$$D_i^m + p_j^m \leq D_j^m, \quad \forall (i, j) \in \Phi_m,$$

$$\forall m \quad (9)$$

$$D_i^m - D_j^m + p_j^m \leq M(1 - Z_{ij}^m) \quad \forall i, j ; j \neq i ;$$

$$\forall m \quad (10)$$

$$Z_{ij}^m + Z_{ji}^m = 1, \quad \forall (i, j) \in \Psi_m, \forall m \quad (11)$$

$$\sum_{v=1}^k \sum_{\substack{u=0 \\ u \neq j}}^{N_m} X_{uj}^{vm} - \sum_{v=1}^k \sum_{\substack{u=0 \\ u \neq i}}^{N_m} X_{ui}^{vm} \leq M(Z_{ij}^m + Z_{ji}^m) \quad \forall i, j; j \neq i;$$

$$l_i < l_j; \forall m, k \quad (12)$$

$$D_j^m + t_{j(N_m+1)}^{km} - Q_k \leq M(1 - X_{j(N_m+1)}^{km}) \quad \forall j, m, k \quad (13)$$

$$r_k - D_j^m + t_{0j}^{km} + p_j^m \leq M(1 - X_{0j}^{km}) \quad \forall j; j \neq i; \quad \forall m, k \quad (14)$$

$$\sum_{l=k+1}^k \sum_{u=1}^{m-1} Y_{lu} - M(1 - Y_{km}) \leq 0 \quad \forall m, k; \quad k \leq K-1 \quad (15)$$

$$Q_k - C_m \leq M(1 - Y_{km}) \quad \forall m, k \quad (16)$$

$$X_{ij}^{km}, Z_{ij}^m, Y_{km} \in \{0, 1\} \quad \forall i, j, m, k \quad (17)$$

$$Q_k, C_m, D_j \geq 0 \quad \forall j, m, k \quad (18)$$

تابع هدف (۱) کل هزینه مدل را مینیموم می‌سازد. سه قسمت تابع هدف شامل: قسمت اول درآمد ناشی از زودکرد را اندازه می‌گیرد. قسمت دوم هزینه ثابت و قسمت سوم هزینه متغیر ناشی از دیرکرد را تعیین می‌نماید. محدودیت (۲) کل میزان دیرکرد و زودکرد را اندازه می‌گیرد. قسمت اول سمت راست محدودیت فوق وابسته به یک میزان تقریبی مربوط به زمان تحویل کارها می‌باشد که بر اساس یک توافق یا قرارداد

بین بندر و کشتی محاسبه می‌شود. محدودیت‌های (۳) و (۴) اولین و آخرین کارهای هر QC را روی هر کشتی تعیین می‌نمایند. محدودیت (۵) تضمین می‌کند که هر کار روی هر کشتی باید به وسیله یک QC تکمیل شود. محدودیت (۶) تضمین می‌کند که اگر یک QC به یک کشتی تخصیص نیابد، هیچ‌کدام از کارهای آن کشتی نمی‌تواند توسط آن QC انجام شود. محدودیت (۷) یک محدودیت تعادلی می‌باشد که تضمین می‌دارد کارها بر اساس یک توالی مشخص انجام شوند. محدودیت (۸) زمان تکمیل برای هر کار را محاسبه می‌نماید. محدودیت (۹) نشان می‌دهد که کار i باید قبل از کار j انجام پذیرد. محدودیت (۱۰) تضمین می‌نماید که چنان چه $Z_{ij}^m = 1$ عملیات روی فعالیت j بر روی کشتی m بعد از عملیات روی فعالیت i تکمیل می‌شود. محدودیت (۱۱) تضمین می‌کند که کار i و j در صورتی که عضو مجموعه مربوط به کارهای دارای روابط هم‌زمانی باشند نمی‌توانند هم‌زمان انجام شوند. با استفاده از محدودیت (۱۲) از تداخل مابین QC ها جلوگیری می‌شود. زمان تکمیل هر QC به وسیله محدودیت (۱۳) تعیین می‌شود. محدودیت (۱۴) زودترین زمان عملیات برای هر QC را محدود می‌نماید. محدودیت (۱۵) نشان می‌دهد که در تخصیص کشتی‌ها به گنتری‌ها تداخلی وجود ندارد. محدودیت (۱۶) زمان تکمیل کشتی‌ها را تعیین می‌سازد.

۳- الگوریتم ژنتیک توسعه داده شده برای حل مساله

الگوریتم ژنتیک در حل مسایل بهینه‌سازی از قبیل مسایل مربوط به توالی عملیات و زمانبندی (Holland, 1975; Goldberg, 1989) کاربرد بسیاری دارد. در واقع GA یک رویکرد فراابتکاری شناخته شده است که با الهام از سیر تکاملی طبیعت در زندگی موجودات زنده کار می‌کند. این الگوریتم در عین حال که به مفهوم حیات برای

مناسب‌ترین ساختار توجه داشته، به صورت تصادفی عمل می‌کند.
توسعه GA برای مساله مذکور شامل گام‌های زیر می‌باشد.

۳-۳-۱ طراحی کروموزوم (نمایش جواب) و جواب اولیه

گام اولیه و همین‌طور مهم‌ترین گام در اجرای الگوریتم ژنتیک نمایش جواب یا طراحی کروموزوم است. کروموزوم باید به گونه‌ای طراحی شود که در وهله اول و تا حد ممکن محدودیت‌های اساسی مساله را برآورده ساخته و مرتبه دوم انعطاف‌پذیری کافی برای اعمال عملگرهای ژنتیکی را داشته باشد. کروموزوم طراحی شده در این تحقیق هر کار در برنامه زمان بندی را به عنوان یک ژن نشان می‌دهد. در الگوریتم پیشنهادی، کروموزوم ماتریسی $V \times (A+1)$ است. که در آن V و A به ترتیب تعداد شناورها و بیش‌ترین تعداد کار روی شناورهاست. در هر ستون برای هر شناور اولین ژن (اولین سطر) بیان کننده تعداد جرثقیل‌های تخصیص یافته به آن شناور است واضح است که مجموع اعداد این سطر کوچک‌تر یا مساوی تعداد جرثقیل‌های در دسترس است. تمامی مجموعه کارهای هر شناور می‌تواند در یک رشته عمودی کد بندی شود. شکل (۳-۲) نمونه‌ای از یک کروموزوم طراحی شده برای ۳ شناور و ۶ جرثقیل است که مبین توالی کارها در شناورهاست. هم‌چنین در این شکل جرثقیل‌های تخصیص داده شده به هر شناور نیز مشخص است.

V_1	V_2	V_3
2	3	1
2	4	3
1	2	1
7	1	2
4	5	4
6	6	5
3	3	0
5	0	0
0	0	0
0	0	0

a. یک مثال از کروموزوم طراحی شده

Assigned QCs to each vessel

vessel 1	Q_1, Q_2
vessel 2	Q_3, Q_4, Q_5
vessel 3	Q_6

b. ساختار جرثقیل‌های تخصیص داده شده به شناورها

شکل **Error! No text of specified style in document.** -۱ یک مثال از

کروموزوم طراحی شده و نحوه عملکرد آن

۳-۳-۲- تولید جمعیت اولیه

این که ساختار جمعیت اولیه نقش اساسی در کارایی الگوریتم ژنتیک دارد، واقعیتی غیرقابل انکار است (Goldberg, 1989). به هر حال بیش‌تر برای اجرای الگوریتم‌های ژنتیک جمعیت اولیه را به صورت تصادفی تولید می‌کنند. برای تولید تصادفی جمعیت اولیه گام‌های زیر را در نظر می‌گیریم:

تعداد کارهای شناور i (N_i) را مشخص نموده و ژن‌هایی با مقدار صفر را به عنوان کار مجازی در نظر می‌گیریم.

اعداد غیر صفر در سطر اول کروموزوم را به عنوان تعداد جرثقیل‌های تخصیص یافته به شناورها قرار داده به نحوی که مجموع این اعداد کم‌تر یا مساوی تعداد جرثقیل‌های در دسترس باشد.

کارهای شناور i ($i=1, \dots, V$) به صورت تصادفی انتخاب کرده و به ژن‌ها در ستون مربوط به شناور i از اولین ژن تا آخرین آن اختصاص می‌دهیم.

باقیمانده ژن‌های خالی را به عنوان کار مجازی (با مقدار صفر) در نظر می‌گیریم. با توجه به توالی فعالیت‌ها در هر شناور که در هر کروموزوم نشان داده شده است، زمان بندی یک جرثقیل می‌تواند با استفاده از مراحل زیر صورت پذیرد که این مدل برگرفته و توسعه داده شده مدلی است که توسط Wang, Lee و Miao در سال ۲۰۰۶ برای یک شناور بیان شده است. هم‌چنین در این مدل مکان اولیه برای هر جرثقیل در شروع افق برنامه ریزی معلوم فرض شده است.

گام اول: با در نظر گرفتن مکان فعلی هر جرثقیل، این که هر جرثقیل کدام کار در کروموزوم را که تا کنون به هیچ جرثقیلی تخصیص نیافته، بدون تداخل با جرثقیل‌های دیگر می‌تواند برعهده گیرد، را مشخص می‌کنیم. اگر تنها یک جرثقیل در دسترس بود، کار را به همان جرثقیل تخصیص داده سپس مکان و زمان تکمیل جرثقیل‌ها و کارها به روز می‌کنیم. اگر بیش از یک جرثقیل موجود بود به گام بعدی می‌رویم.

گام دوم: زمان تکمیل کارهای تخصیص داده شده جرثقیل‌های در دسترس را با هم مقایسه داشته و کار را به جرثقیلی که زمان تکمیل کوتاه‌تری دارد اختصاص می‌دهیم سپس مکان و زمان تکمیل جرثقیل‌ها و کارها را روزآمد می‌سازیم. اگر زمان تکمیل

بیش از چند جرثقیل یکسان بود به گام ۳ می‌رویم.

گام سوم: فاصله بین این کار و جرثقیل‌های در دسترس را مقایسه نموده و کار را به جرثقیل نزدیک‌تر اختصاص می‌دهیم. مکان و زمان تکمیل جرثقیل‌ها و کارها را روزآمد می‌سازیم. اگر جرثقیل‌هایی با فواصل مساوی وجود داشتند به گام ۴ می‌رویم.

گام چهارم: کار را به جرثقیلی با عدد کوچک‌تر تخصیص می‌دهیم و سپس مکان و زمان تکمیل جرثقیل‌ها و کارها را روزآمد می‌داریم.

گام پنجم: گاه‌های ۱ تا ۴ را تا زمانی که کار تخصیص داده نشده‌ای در کروموزوم موجود باشد، ادامه می‌دهیم.

پس از تخصیص کارها به جرثقیل‌ها، محدودیت‌های مقدم و یا هم‌زمان به صورت زیر برآورده می‌شوند:

فرض می‌کنیم کار i بر کار j مقدم است، گام‌های زیر را برای تخصیص کار j به جرثقیل‌ها در نظر می‌گیریم.

اگر کار i به هیچ جرثقیلی تخصیص داده نشده باشد، مکان کار i و j را در کروموزوم جا به جا می‌کنیم. به عبارت دیگر کار i را به جای کار j تخصیص می‌دهیم.

اگر کار i به جرثقیلی اختصاص یافته باشد، با توجه به محدودیت (۳-۹) زمان تکمیل کار i را به عنوان زمان شروع کار j در نظر می‌گیریم.

فرض کنیم که کار i و j نمی‌توانند به صورت هم‌زمان انجام شوند و برای تخصیص این کارها به جرثقیل‌ها به صورت زیر عمل می‌کنیم:

اگر کار i (j) به هیچ جرثقیلی تخصیص داده نشده باشد با روش بالا کار i (j) را به

یک جرثقیل اختصاص می‌دهیم.

اگر کار i (j) پیش تر به یک جرثقیل اختصاص یافته باشد، طبق محدودیت (۳-۱۰) زمان تکمیل کار i (j) را به عنوان زمان شروع کار i (j) در نظر می گیریم. نمونه ای از تخصیص کارها به جرثقیل در هر شناور با استفاده از الگوریتم بالا در شکل (۳-۳) نشان داده شده است. در نتیجه تابع برازندگی ۲۲ کروموزوم تعریف شده در معادله (۳-۱) محاسبه شده است.

Sequence of jobs on each QC	
Q_1	2, 1, 7, 4
Q_2	6, 3, 5
Q_3	4, 2
Q_4	1
Q_5	5, 6, 3
Q_6	3, 1, 2, 4, 5

شکل. Error! No text of specified style in document. -۲- یک نمونه از

ساختار و نحوه تخصیص گنتری ها و زمان بندی آن ها در یک کروموزوم.

۳-۳-۳- طراحی عملگرهای ژنتیکی

در این تحقیق از دو عملگر تقاطعی به طور جداگانه برای دو بخش کروموزوم استفاده شده است. سطر اول کروموزوم نشان دهنده جرثقیل های تخصیص داده شده به شناورها می باشد و سایر ژن ها بیان کننده توالی کارها در شناور هستند. برای بخش اول و دوم به ترتیب از دو عملگر تقاطعی حسابی ۲۳ و عملگر extended patching استفاده شده است.

²² fitness

²³ Arithmetic Crossover

۱-۳-۳-۳ عملگر تقاطع حسابی

از آنجا که شذنی بودن فرزندان تولید شده بسیار حایز اهمیت است، از یک عملگر تقاطعی کارا به نام عملگر تقاطعی حسابی برای جستجو در فضای جواب و به دست آوردن فرزندان شذنی به طور هم زمان استفاده شده است. این عملگر با ایجاد یک ترکیبی خطی از والدین، موجب می‌شود که فرزندان به دست آمده خصیصه شذنی بودن را از والدین خود به ارث برند. به لحاظ ماهیت مساله مورد بررسی، سایر عملگرهای تقاطعی در اکثر مواقع موجب تولید جواب‌های نشذنی خواهند شد که این امر منجر به افزایش بسیار زیاد زمان محاسبات و ناکارایی الگوریتم خواهد شد. فرم کلی عملگر تقاطعی حسابی زیر می‌باشد:

$$\text{Offspring} \equiv \lambda \times \text{Parent 1} + (1-\lambda) \times \text{Parent 2}$$

λ یک عدد تصادفی تولید شده بین (۰,۵, ۱) است. بنابراین ارزش هر ژن در فرزندان تولید شده جدید به دست می‌آید. برای افزایش تاثیر والدین با fitness بهتر، آن‌ها را به عنوان Parent 1 در نظر می‌گیریم. در صورتی که والدین شذنی باشند، عملگر تقاطعی حسابی همواره تضمین می‌کند که فرزندان تولید شده نیز شذنی هستند. از آن جایی که ژن‌ها فقط مقادیر عدد صحیح را می‌پذیرند، در نتیجه باید جزو صحیح offspring ([offspring]) به عنوان خروجی مطلوب در نظر گرفته شود. بنابراین ممکن است تمامی محدودیت‌های جرثقیل‌ها برآورده نشده و ما با خطای $\varepsilon = \sum V [\text{gene}] - K \cdot \text{value}(\text{offspring})$ مواجه شویم. برای غلبه بر این مساله اگر $\varepsilon > 0$ الگوریتم بیش‌ترین مقدار را جایگزین ε می‌سازد.

کارایی عملگر حسابی تقاطعی با $\lambda = 0.7$ و تعداد ۷ جرثقیل در شکل ۵ نشان داده شده است.

$$\lambda * \begin{array}{|c|c|c|} \hline \text{parent 1} \\ \hline 2 & 4 & 1 \\ \hline \end{array} + (1-\lambda) * \begin{array}{|c|c|c|} \hline \text{parent 2} \\ \hline 3 & 2 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline \text{offspring(1)} \\ \hline 2.3 & 3.4 & 1 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|} \hline \text{offspring(1)} \\ \hline 2.3 & 3.4 & 1 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline \text{offspring(2)} \\ \hline 2 & 3 & 1 \\ \hline \end{array}$$

شکل ۳-۳-Error! No text of specified style in document. توصیفی شماتیک از

عملگر تقاطع حسابی

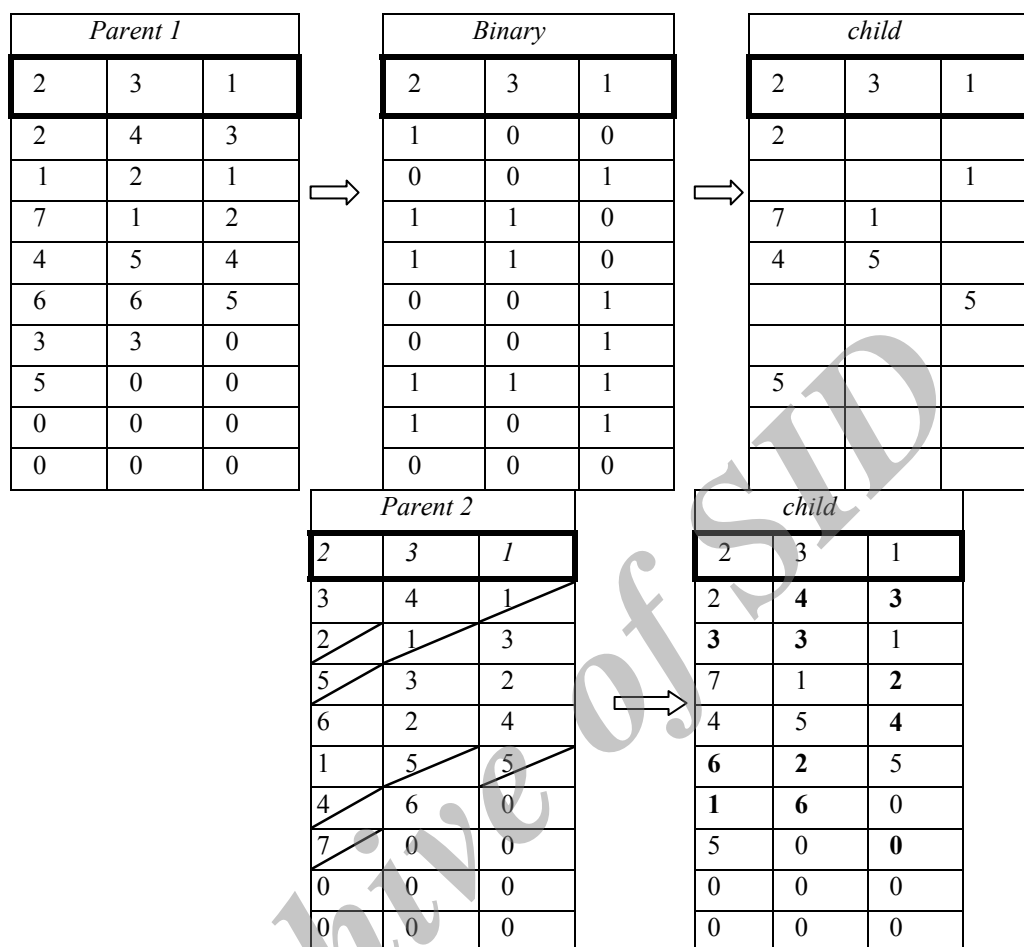
۳-۳-۲- عملگر تقاطعی جهشی توسعه یافته

عملگر تقاطعی جهشی اولین بار توسط ژانگ و چن (Chang & Gen (1997) برای حل مسایل استفاده شد. آنان این عملگر را عملگر *uniform order-based* نامیدند. این عملگر یک الگوی باینری که در آن صفر و یک‌ها کنترل کننده هستند، تولید می‌کند. با در نظر گرفتن این عملگر و یکی از والدین رشته باینری روی یکی از والدین در نظر گرفته می‌شود به نحوی که ژن‌هایی که در موقعیت یکسان هستند مستقیم بر روی ژن فرزند به صورت نظیر به نظیر منتقل می‌شوند. باقی مانده ژن‌ها در فرزند تولید شده، که در واقع مربوط به مکان‌های صفر در الگوی باینری بوده‌اند توسط ژن‌های ولاد دوم پر می‌شوند. در این تحقیق ما یک عملگر متقاطع جهشی توسعه یافته برای بخشی از کروموزوم که نشانگر توالی کارها در شناورهایست، پیشنهاد کرده‌ایم. در این عملگر یک ماتریس باینری $(V \times A)$ معرفی شده و از عملگر برای هر ستون (شناور) به طور جداگانه استفاده شده است. شکل ۶ نمونه‌ای از این عملگر تقاطعی را شرح داده است. گاه‌های اصلی در ساختار الگوریتم پیشنهادی برای این عملگر به صورت زیر است:

در ماتریس باینری به تعداد p عدد 1 را بر طبق فرمول $\lceil (V \times A) / 2 \rceil$ تولید می‌کنیم.

- ۱- به نحو تصادفی یک رشته باینری "1" with p همان گونه که در گام ۱ تعریف شده، تولید می‌کنیم.
- ۲- به طور تصادفی دو والد از جمعیت را انتخاب می‌کنیم. (به عنوان مثال والد ۱ و ۲).
- ۳- ژن‌های والد ۱ را در مکان‌هایی که ماتریس باینری دارای عدد ۱ است را در کروموزوم فرزند در همان مکان قرار می‌دهیم. جز ژن‌هایی با مقدار صفر.
- ۴- ژن‌هایی از والد ۲ که پیش‌تر از والد ۱ روی فرزند کپی شده اند را خط زده و کنار بگذارید (این عمل برای این است که از انتخاب ژن‌های تکراری جلوگیری شود). دقت کنید که asterisk ها را برای این عمل انتخاب نکنید چرا که تعداد بهینه‌گنتری‌های تخصیص یافته به هر کشتی نباید بر اساس وضعیت مشابه در والد باشد. هم چنین ژن‌های با مقدار صفر مجازی می‌باشند لذا آن‌ها را نیز نباید انتخاب کنید.
- ۵- باقی مانده ژن‌ها را با ژن‌های قطع نشده در والد ۲ با حفظ توالی ژن‌ها در والد ۲ پر می‌کنیم.

Archive of SID

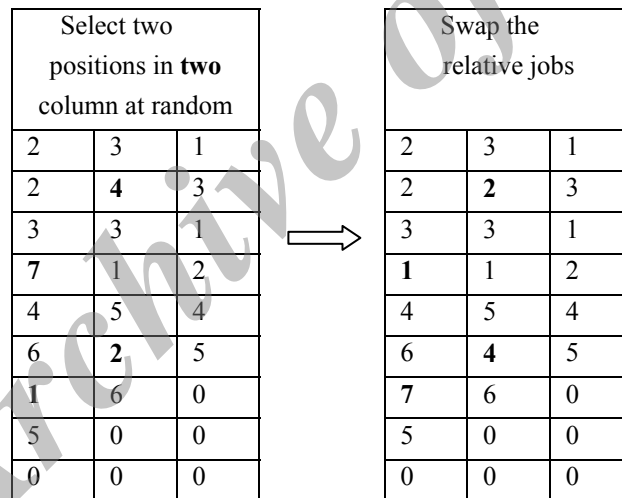


شکل ۴-Error! No text of specified style in document. ساختار عملگر تقاطعی ارایه شده

۳-۳-۳-۳- عملگر جانشینی جهشی

وظیفه اصلی عملگر جهش ایجاد تنوع در نسل‌های متوالی و بهره برداری از فضای جواب است. به طور کلی در جهش هر بیت از individuals هر فرزند در جمعیت کنترل می‌شود. و مقدار هر بیت با یک نرخ از پیش تعیین شده جا به جا می‌شود.

عملگر جهش شامل جابه جایی هر دو ژنی از کروموزوم است که به طور تصادفی انتخاب شده است (Cheng & Gen, 1997). در این عملگر یک فاکتور اولیه به نام طول جهش در نظر گرفته می شود. طول جهش نشان دهنده تعداد دفعات استفاده از این عملگر می باشد. به عبارت دیگر طول جهش در عملگر جایگزینی جهش نشان دهنده تعداد دفعات جایگزینی برای هر بار استفاده از این عملگر می باشد. به عنوان مثال طول جهش ۴ مبین این مطلب است که ۴ جفت ژن را باید دو به دو و به نحو تصادفی با هم تعویض نمود. لازم به ذکر است که انتخاب تصادفی ژن ها بر اساس یک مقدار احتمالی $p(m)$ خواهد بود. به این صورت که با احتمال p هر ژن برای جهش انتخاب شده و پس از انتخاب دو ژن آن ها با یکدیگر تعویض می شوند. این عمل به تعداد دفعاتی برابر با طول جهش تکرار می گردد. در این تحقیق طول جهش برابر ۲ در نظر گرفته شده است.



شکل ۵-Error! No text of specified style in document. یک نمایش از عملگر جهشی ارائه شده

۳-۳-۴- استراتژی انتخاب والدین

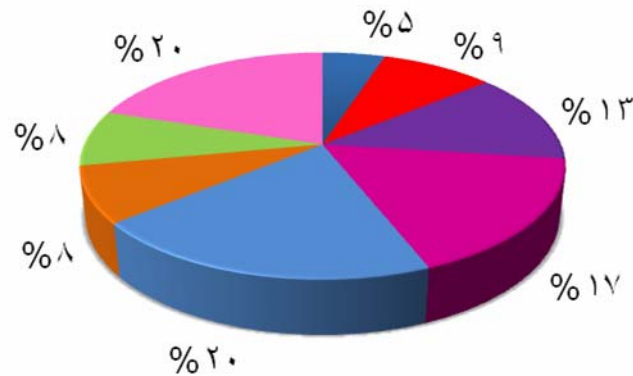
انتخاب والدین در regulating the bias در فرایند تولید مثل اهمیت زیادی دارد. استراتژی انتخاب والدین بیانگر روش انتخاب کروموزوم در جمعیت فعلی است که در واقع منشا ایجاد نسل بعدی می‌باشد. بدیهی است که حل بهتر در نسل فعلی، شانس بیش‌تری برای انتخاب به عنوان والد برای تولید فرزند دارد. معمول‌ترین روش برای انتخاب والد، استراتژی کلاسیک چرخ رولتی^{۲۴} است. در این روش از عملگری استفاده می‌شود که مبتنی بر یک احتمالی از تابع Fitness باشد. بنابراین رشته i ام در جمعیت با یک احتمالی از F_i (که مقدار تابع Fitness برای آن رشته است) انتخاب می‌شود که باید حاصل جمع این احتمال‌ها نیز برابر یک شود. احتمال رشته انتخابی i

$$P_i = \frac{F_i}{\sum_{j=1}^n F_j}$$

برابر است با: که n اندازه جمعیت است. حال اگر درصد احتمال این مقادیر را به صورت شکل (۳-۷) یک چرخ رولت نشان دهیم، آن گاه مقدار مناسب بودن جمعیت پس از n بار چرخیدن این چرخ به دست خواهد آمد. در هر زمان یک مورد از رشته‌ها توسط اشاره گر چرخ رولت انتخاب می‌شود. از آن جایی که محیط چرخ با مقادیر درصد احتمال Fitness رشته‌ها نشانه‌گذاری شده است، انتظار می‌رود که

مکانیزم چرخ به تعداد $\frac{F_i}{\bar{F}}$ کپی از رشته i ام را تولید کند.

²⁴ Roulette Wheel



شکل ۶

شکل ۶ نمونه‌ای از یک چرخ رولت است که در آن هر کروموزوم به تکه‌ای از دایره رولت تخصیص می‌یابد و سایز آن تکه با Fitness آن کروموزوم متناسب است. این چرخ به تعداد اندازه جمعیت می‌چرخد و در هر چرخش کروموزومی که نشان گر چرخ بر روی آن قرار گرفته است انتخاب گشته و به استخر فرزندان برای نسل بعد منتقل می‌شود.

۳-۳-۵- استراتژی پذیرش فرزند

در این تحقیق از استراتژی semi-greedy (حریصانه) برای پذیرش فرزند توسط عملگر ژنتیک استفاده شده است. در این استراتژی، یک فرزند ۲۵ زمانی برای نسل جدید پذیرفته می‌شود که Fitness آن از میانگین Fitness والدینش کمتر باشد. این استراتژی سبب می‌شود که الگوریتم زودتر به جواب بهینه نزدیک شود.

²⁵ offspring

۳-۳-۶- معیار توقف

جهت معیار توقف از دو استراتژی استفاده شده است :

ماکزیمم تعداد نسل‌ها.

انحراف استاندارد میزان Fitnessها در یک نسل [۱۳]

این پارامتر بر میزان نزدیکی یا تفاوت در نسل ایجاد شده با مقدار تابع هدف دلالت می‌کند.

نحوه عملکرد معیار دوم به این نحو است که اگر میزان اختلاف Fitnessها در یک نسل از یک عدد ثابت در نظر گرفته شده کوچکتر باشد، الگوریتم متوقف می‌شود. به طوری که انحراف استاندارد تفاوت Fitnessها در نسل g از رابطه زیر به دست می‌آید:

$$\sigma_g = \left[(1 / Pop_Size) \sum_{k=1}^{Pop_Size} (F_g^k - \bar{F}_g)^2 \right]^{1/2}$$

که در آن F_{gk}

میزان Fitness k امین کروموزوم نسل g می‌باشد. \bar{F}_g برابر میانگین کل Fitnessهای نسل g می‌باشد که از رابطه زیر به دست می‌آید:

$$\bar{F}_g = (1 / Pop_Size) \sum_{k=1}^{Pop_Size} F_g^k$$

بنابراین اگر $g > G_{max}$ یا $\sigma_g \leq \varepsilon$ آنگاه الگوریتم متوقف می‌شود.

۴- نتایج محاسباتی

۴-۱- اجرای مدل

در این بخش کارایی مدل پیشنهادی و الگوریتم ژنتیک توسعه داده شده با ۳۰ مثال اجرایی در اندازه‌ها مختلف نشان داده شده است. مثال‌هایی با حجم پایین با استفاده از

روش شاخه و کرانه در نرم افزار LINGO 8.0 در یک کامپیوتر شخصی ۲۶ حل شده است. همه این مثال ها هم چنین با الگوریتم ژنتیک پیشنهادی در نرم افزار MATLAB 7.2 با کامپیوتری مشابه قبل نیز حل شده است. نتایج به دست آمده از هر دو نرم افزار از لحاظ مقدار تابع هدف ۲۷ و زمان حل با یکدیگر مقایسه شده اند. از آن جایی که جواب‌های به دست آمده از طریق الگوریتم ژنتیک نزدیک به بهینه می‌باشد، برای داشتن جواب‌هایی بهتر هر مثال را ۲۰ بار با الگوریتم ژنتیک حل کرده سپس میانگین نتایج به دست آمده را به عنوان جواب نهایی در نظر گرفته‌ایم.

۴-۱-۱- مثال هایی با ابعاد کوچک

۱۰ مثال با حجم کوچک در این تحقیق حل شده است.

بر مبنای محاسبات اولیه، اندازه جمعیت، احتمال crossover، احتمال جهش و حداکثر تعداد نسل به ترتیب ۱۵۰، ۰.۲۵، ۰.۱ و ۱۰۰ در نظر گرفته شده اند. همان طور که در جدول ۴-۱ نشان داده شده است، زمان اجرا در نرم افزار LINGO و با استفاده از روش شاخه و کرانه به طور محسوسی با افزایش ابعاد مساله در حال افزایش می‌باشد. مزید بر آن و همان گونه که مشخص است توسط الگوریتم GA پیشنهادی جوابی نزدیک به بهینه را در زمانی منطقی به دست آورده‌ایم. میانگین تفاوت بین روش شاخه و کرانه (جواب بهینه قطعی) و الگوریتم ژنتیک پیشنهادی (جواب بهینه تقریبی) از نظر مقدار تابع هدف برای حجم کوچک حدود ۰.۲٪ و با انحراف معیار ۰.۶۷ می‌باشد.

²⁶ Including Two Intel® Core™2 T5600@1.83 Ghz Processors And 512 Gb Ram

²⁷ Objective Function Value (OFV)

جدول ۱- نتایج حل مساله در ابعاد کوچک

No.	Problem Information			B&B		GA		Gap (%)
	No.V	No.QC	No.job	CPU Time (Sec.)	OFV	Mean CPU Time(Sec.)	OFV	
3	2	5	3	5	62	3	62	0
5	3	5	3	12	106	5	108	1.8
2	2	5	5	23	175	8	179	2
6	3	5	5	47	185	9	188	1.6
4	2	8	5	216	150	11	153	2
9	4	8	5	1811	310	13	320	1.6
7	3	5	8	4556	325	18	332	2
8	3	8	8	4925	331	20	342	3.3
10	4	12	8	5308	342	22	354	3.2
Average								1.9

۴-۱-۲- مثال هایی با ابعاد بزرگ

در این تحقیق، ما هم چنین ۱۰ مثال در ابعاد بزرگ نیز حل کردیم. بر مبنای محاسبات اولیه، اندازه جمعیت، احتمال Crossover، احتمال جهش و حداکثر تعداد نسل به ترتیب ۳۰۰، ۰.۲، ۰.۲۵ و ۱۰۰۰ در نظر گرفته شده‌اند. برای ارزیابی کارایی الگوریتم ژنتیک پیشنهادی در حل این مثال‌ها، میانگین فاصله ایجاد شده بین نتایج حاصله برای تابع هدف که توسط روش شاخه و کرانه به دست آمده و نتایج حاصل توسط الگوریتم ژنتیک را به دست آوردیم که برابر ۳.۵٪ می‌باشد. جواب‌های حاصل از حل این مثال‌ها در جدول ۴-۲ آمده است.

جدول ۲- نتایج حل مسأله در ابعاد کوچک

No.	Problem Information			B&B		GA		Gap (%)
	No.V	No.QC	No.job	CPU Time (Sec.)	OFV	Mean CPU Time(Sec.)	OFV	
1	5	12	8	5640	381	36	392	3.3
2	3	5	12	8832	526	68	545	3.8
3	3	8	12	9205	580	76	605	4.3
4	4	5	12	10800	570	91	592	3.6
5	4	8	12	10800	622	99	648	4.1
6	4	12	12	10800	748	104	760	2.9
7	5	12	12	10800	796	108	820	3
8	3	5	16	-	-	125	1022	-
9	3	8	16	-	-	136	1036	-
10	4	5	16	-	-	141	1145	-
Average								

۵- نتیجه گیری و پیشنهادها:

در این مطالعه یک الگوریتم ژنتیک کارا برای حل مسأله زمان بندی جرثقیل‌های کانتینری ارایه شده است. برای این مسأله یک مدل برنامه‌ریزی عدد صحیح آمیخته که تعریف شده است که ضمن زمانبندی جرثقیل‌های اسکله به طور هم‌زمان جرثقیل‌ها را به شناورها نیز تخصیص ۲۸ می‌دهد. هدف در این مسأله کمینه کردن هزینه‌ها که شامل (۱) مجموع هزینه‌های ثابت و متغیر جرثقیل تخصیص داده شده به شناورها (۲) مجموع هزینه‌های حاصل از دیرکرد جرثقیل‌های تخصیص داده شده به شناورها و بیشینه کردن

مجموع عایدی‌های حاصل از انجام زودتر از موعد کارهای مربوط به شناورها در بندر می‌باشد.

به دلیل ماهیت NP-hard مدل مذکور، یک الگوریتم ژنتیک (GA) توسعه داده شد که به لحاظ نحوه نمایش جواب و طراحی عملگر قابل توجه می‌باشد. هم چنین به دلیل تعدد محدودیت‌های تساوی در مدل پیشنهادی که پیچیدگی مساله و نحوه طراحی عملگر را دوچندان می‌نماید، از عملگر تقاطع حسابی و هم چنین عملگر جهش (به اصطلاح) پله سنگی برای کاوش فضای جواب استفاده شد. مقایسه بین نتایج به دست آمده توسط GA و رویکرد شاخه و کران، مبین یک اختلاف نسبی حدود ۱٪ درصد با انحراف استاندارد می‌باشد که نتیجه بسیار مطلوبی است. علاوه بر این، با افزایش ابعاد مساله میزان اختلاف جواب‌ها افزایش نیافته و میزان اختلاف مابین نتایج حل در دو رویکرد مذکور روند صعودی را نشان نمی‌دهد.

جهت تحقیقات آینده می‌توان موارد زیر را پیشنهاد نمود:

حل مدل پیشنهادی می‌تواند توسط سایر رویکردهای فرا ابتکاری مورد بررسی قرار گیرد.

منابع و مراجع:

1. Kim K.H, and Park Y.M. A crane scheduling method for port container terminals, European Journal of Operation Research 156(2004) 752-768
2. Imai, A., Nagaiwa, K., Tat, C.W. (1997), Efficient planning of berth allocation for container terminals in Asia, Journal of Advanced Transportation 31(1), 75-94.

3. Imai, A., Nishimura, E., Papadimitriou, S. (2001), The dynamic berth allocation problem for a container port, *Transportation Research B* 35, 401-417
4. Nishimura, E., Imai, A., Papadimitriou, S. (2001), Berth allocation planning in the public berth system by genetic algorithms, *European Journal of Operational Research* 131, 282-292.
5. Imai, A., Nishimura, E. and Papadimitriou, S., Berth allocation with service priority. *Transportation Research-B*. v37 i5. 437-457.
6. Imai, A., Nishimura, E., Hattori M. and Papadimitriou S., (2007). Berth allocation at indented berths for mega-containerships, *European Journal of Operational Research* 179 (2), 579-593.
7. Daganzo CF (1989) The crane scheduling problem. *Transportation Research* 23B (3):159–175
8. Li C-L, Cai X, Lee C-Y (1998) Scheduling with multiple-job-on-one-processor pattern. *IIE Transactions* 30: 433–445
9. Kap Hwan Kim, Young-Man Park (2004), A crane scheduling method for port container terminals *European Journal of Operational Research* 156: 752–78
10. Ebru K. Bish (2003) A multiple-crane-constrained scheduling problem in a container terminal, *European Journal of Operational Research* 144: 83–107
11. Luigi Moccia, Jean-François Cordeau, Manlio Gaudioso, Gilbert Laporte (2005), A Branch-and-Cut Algorithm for the Quay Crane Scheduling Problem in a Container Terminal, *Wiley InterScience*
12. Der-Horng Lee, Hui Qiu Wang, Lixin Miao (2006) Quay crane scheduling with non-interference constraints in port container terminals, *Transportation Research Part E*

13. Sung Ho Jung · Kap Hwan Kim (2006), Load scheduling for multiple quay cranes in port container terminals, *J Intell Manuf* (2006) 17:479–492
14. R. Tavakkoli-Moghaddam and N. Safaei “An Evolutionary Algorithm for a Single-Item Resource-Constrained Aggregate Production Planning Problem, , IEEE Congress on Evolutionary Computation, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada July 16-21, 2006.
15. Zhang C., Liu J., Wan Y.W., Murty K.G and Linn R.J. Storage space allocation in container terminals, *Transportation Research Part B* 37 (2003) 883–903.
16. Goldberg, D. E., *Genetic algorithms in Search, Optimization and Machine Learning*, Addison-Wiley, Reading, MA, 1989.
17. Cheng, R. and M. Gen, “Parallel Machine Scheduling Problems Using Memetic Algorithms”, *Computers & Industrial Engineering*, Vol. 33, No. 3-4, pp. 761-764, 1997.

Archive of SID