

ارائه یک الگوریتم زمان‌بندی منابع در شبکه‌های Grid

مهدی مشایخی^۱، حمید شکرزاده^۲، مهدی حسنی گودرزی^۳

۱- پژوهشگاه صنعت نفت، ۲- دانشگاه آزاد واحد پردیس، ۳- دانشگاه آزاد واحد تهران جنوب

E-mail: mshayekhi@ripi.ir, shokrzadeh@gmail.com, mehdigoodarzi@azad.ac.ir

چکیده - شبکه‌های گرید امروزه به راه حل جایگزینی برای ابرکامپیوترها تبدیل شده‌اند. در حال حاضر این شبکه‌ها به طور گسترده‌ای در کشورهای توسعه یافته به منظور جلوگیری از هدر رفتن منابع و استفاده بهینه از آنها مورد استفاده قرار می‌گیرند. اساس کار این شبکه‌ها تقسیم یک کار به واحدهای کوچک و اختصاص آنها بین منابع ناهمگون متعدد برای انجام است. یکی از مهمترین مسائل موجود در این شبکه‌ها، زمان‌بندی این منابع ناهمگون می‌باشد. به عبارت دیگر زمان‌بندی در این شبکه‌ها نحوه تقسیم کار بین منابع با رعایت پارامترهای کیفیت سرویس می‌باشد. در این مقاله روشی جدید برای زمان‌بندی منابع در این شبکه‌ها ارائه شده است. مسائل مربوط به زمان‌بندی معمولاً از نوع NP بوده و در زمان عادی حل پذیر نمی‌باشند. در این مقاله با خلاصه سازی فضای حالت جواب مسئله، روشی جدید برای حل این گونه مسائل ارائه گردیده است.

کلید واژه- شبکه Grid، زمان‌بندی، کیفیت سرویس

۱ مقدمه

داده‌ای هستند. هدف اصلی این نوع گرید فراهم ساختن یک زیرساخت مناسب برای سنتز کردن و ساختن اطلاعات جدید از انباره‌های داده‌ای توزیع شده در گرید و حذف گلوگاه‌ها می‌باشد [۲]. دسته سوم گریدهای سرویسی هستند که هدف اصلی آنها فراهم کردن سرویسهایی است که یک کامپیوتر به تنهایی قادر به ارائه نمودن آن سرویسها نمی‌باشد [۳].

با توجه به گریدهای موجود (داده‌ای، محاسباتی، سرویسی) آنچه که ملموس است، مفهوم پردازش پیرنگتر از سایر موارد به نظر می‌رسد. با بهره‌وری از توان پردازشی چندین پردازنده و زمان‌های بیکار آنها می‌توان به قدرت پردازشی سریعتر از کامپیوترهای امروزی رسید. بنابراین زمان‌بندی درست درخواستهای استفاده از منابع، مخصوصاً درخواست‌های استفاده از منابع محاسباتی امری مهم به نظر می‌رسد.

۲ چالشهای مهم زمان‌بندی در گرید

مهمترین چالش در این زمینه ماهیت زمان‌بندی و توزیع درخواستها در گرید است که آنرا از روشهای زمان‌بندی ارائه

گرید یک سیستم توزیع شده در مقیاس وسیع بوده که قابل گسترش در حد گسترده‌ای به نام اینترنت است. گرید یک فناوری است که امکان اشتراک، مدیریت، هماهنگ سازی و کنترل منابع توزیع شده را فراهم می‌سازد. نظریه گرید را می‌توان ترکیبی از نظریه سیستمهای موازی و توزیعی به حساب آورد [۱] یعنی گرید هم قابلیت بهره‌وری از منابع توزیع شده و هم قابلیت پردازش موازی حجم عظیمی از اطلاعات را دارد. هدف اصلی گرید بهره‌وری کامل از تمام منابع موجود در شبکه همچون منابع محاسباتی، منابع ارتباطی، منابع ذخیره سازی، منابع نرم افزاری و غیره است.

سیستم‌های گرید بر اساس کاربرد به سه دسته تقسیم می‌شوند. دسته اول گریدهای محاسباتی هستند که هدف اصلی آنها پردازش و محاسبات در حجم بسیار بالا می‌باشد. در این نوع گریدها، برنامه‌ها باید قابلیت شکستن به واحدهای محاسباتی کوچکتر به نام کار را داشته باشند. کارها در صف سیستم قرار می‌گیرند و بعد از کشف منابع مناسب، زمان‌بندی می‌شوند و پردازش نهایی بر روی نتایج بدست آمده صورت می‌پذیرد [۲]. دسته دوم گریدهای

شدن در گرید برخوردار هستند. این کاربردها از یک سری وظیفه‌های مستقل و مجزا که به نیازهای ارتباط درون پردازشی ندارند تشکیل شده اند [۶].

۳ مدل و سیاست زمانبندی

بیشتر مدل‌های زمان‌بندی در گریدها به ۳ نوع ساختار دسته‌بندی می‌شوند [۴]: متمرکز، سلسله‌مراتبی و غیرمتمرکز.

- در ساختار متمرکز، یک زمانبند متمرکز تکی، کارها^۱ را روی منابع موجود زمانبندی می‌کند. از آنجایی که کل اطلاعات زمانبندی فقط روی یک نقطه موجود است، سیستم زمانبندی در این روش بهینه است اما در یک سیستم گرید، توسعه پذیر نیست.

- در ساختار غیرمتمرکز، درخواست کنندگان و صاحبان منابع به طور مستقل زمانبندی می‌کنند. این روش توسعه پذیر است و برای سیستم‌های گرید مناسب است.

- ساختار سلسله‌مراتبی، زمانبندها به صورت یک سلسله مراتب سازمان‌دهی می‌شوند. در سطوح بالاتر، موجودیت‌های منابع سطح بالا هستند و زیر موجودیت‌های کوچک‌تر در سطح پایین‌تر سلسله مراتب زمانبندی می‌شوند. این مدل ترکیبی از دو مدل بالا است.

کاربران برنامه‌ها را ارسال می‌کنند و با در نظر گرفتن نیازمندیهای QoS داده‌ها را وارد می‌کنند. سیستم مدیریت می‌بایست نیازمندیهای منابع را جهت اجرای برنامه تخمین بزند. ۳ سیاست برای این مسئله وجود دارد: پیش‌بینی

شده در سایر محیطها متمایز می‌کند که این موارد به طور خلاصه در زیر آمده است:

الف) خود مختاری مکان و ناهمگنی منابع عواملی هستند که باعث پیچیدگی مدیریت منابع می‌شوند. منابع سیستم‌های گرید در حوزه‌های مدیریتی جداگانه توزیع می‌شوند که باعث ناهمگنی منابع، تفاوت در کاربرد و سیاست‌های زمانبندی در مکانیسم‌های مدیریتی می‌شود [۴].

ب) با توجه به ماهیت پویا بودن گرید، چالش مهم، استفاده از یک تخمین و پیش‌بینی مناسب می‌باشد که با بکارگیری الگوریتم‌های ژنتیکی و الگوریتم‌هایی که ماهیت تخمینی دارند میسر می‌گردد.

ج) در گرید باید برای اجرای درخواست‌های بحرانی، قابلیت اعتماد بالاتری ایجاد شود؛ که این کار را می‌توان با المثنی سازی آن درخواست انجام داد تا در مهلت خواسته شده و با قابلیت اطمینان بالاتری اجابت شوند.

د) در گرید باید برای زمان‌بندی مجدد درخواست‌هایی که امکان اجابت آنها در مهلت تعیین شده نیست، منابع مناسب بعدی به صورت پویا مشخص گردند تا زمان‌بندی دوباره بدون وقفه انجام گیرد.

و) در گرید سعی بر این است که زمان‌بندی براساس پارامترهای کیفیت سرویس انجام گیرد. با پدید آمدن ایده‌های گریدهای محاسباتی، که توسط فوستر و کسلمن [۵] در سال ۱۹۹۹ مطرح گردید، یک سکوی نرم‌افزاری جدید برای اجرای کاربردها بر روی تعداد زیادی از منابع محاسباتی متجانس و توزیع شده و با عبور از حوزه‌های مدیریتی، و اعمال سیاست‌های مختلف به وجود آمد. با توجه به تعریف، یک کاربرد مجموعه‌ای از وظیفه‌ها و کارهای متعدد است و هر کار به عنوان یک واحد اتمی از محاسبات کاربرد، تعریف می‌گردد.

تمام کاربردها از پتانسیل لازم برای اجرا شدن در گرید محاسباتی برخوردار نیستند و امکان دارد هزینه و زمان اجرای برخی از کاربردها در گرید بسیار بالاتر از اجرای آن کاربرد در یک ماشین خاص باشد. با وجود این تعدادی از کاربردها وجود دارد که از پتانسیل بسیار بالایی برای اجرا

^۱ Jobs

جواب بدست آمده را با یک تکنیک خاص به نام خلاصه سازی فضای حالت جواب، مقایسه می‌کنیم.

۴-۱- طرح یک مسئله ساده [۱۳]

فرض کنیم که برای پردازش یک حجم از اطلاعات به NC میلیون سیکل CPU نیاز داریم. این مسئله باید در حداکثر T_{max} واحد زمانی حل شود همچنین هزینه پردازش ما نباید بیشتر از C_{max} دلار بشود. فرض کنید حداکثر سه منبع محاسبه برای تخصیص در اختیار داریم؛ فرض کنیم S_i ($i=1,2,3$) سرعت منبع i بر حسب میلیون سیکل بر ثانیه می‌باشد. C_i ($i=1,2,3$) برابر هزینه بر حسب دلار بر ثانیه منبع i می‌باشد. ما قادر خواهیم بود مسئله خود را به سه قسمت و موازی شکسته و حل کنیم بنابراین در مسئله ما ۷ حالت برای تخصیص منابع وجود می‌آید (۱)، (۲)، (۳)، (۱ و ۲)، (۱ و ۳)، (۲ و ۳)، (۱ و ۲ و ۳).

T را برابر زمان اجرا و C را برابر هزینه اجرای مسئله در بین منابع در نظر می‌گیریم. بعضی از حالت‌های ممکن تخصیص منابع بعلاوه تجاوز از زمان T_{max} و یا هزینه C_{max} مناسب نیستند؛ اگر فقط از یک منبع استفاده شود :

$$T = \frac{NC}{S_i} \leq T_{max} \quad (i=1,2,3) \quad (۱)$$

$$C = C_i \times \frac{NC}{S_i} \leq C_{max} \quad (i=1,2,3) \quad (۲)$$

فرمول ۱ نشان می‌دهد که زمان اجرای کل در منبع i نباید از T_{max} تجاوز کند. فرمول ۲ نیز نشان دهنده هزینه است. زمان و هزینه زمانی که از ۲ منبع i و ۱ استفاده می‌شود برابر است با :

$$T = \max \left\{ \frac{NC_i}{S_i}, \frac{NC_j}{S_j} \right\} \leq T_{max} \quad (۳)$$

$$C = c_i \times \frac{NC_i}{S_i} + c_j \times \frac{NC_j}{S_j} \leq C_{max} \quad (۴)$$

$$NC_i + NC_j = NC \quad (۵)$$

اگر از همه سه منبع استفاده کنیم :

مبتنی بر تئوری، Testcase و مبتنی بر تاریخچه است [۱۲].

- پیش بینی تئوری : نیازمندیهای برنامه کاربردی براساس تحلیل مدل برنامه نویسی برنامه و حوضه مسئله تخمین زده می‌شوند. کارایی این روش بستگی به مقدار درک مدل محاسباتی برنامه دارد.

- پیش بینی مبتنی بر تاریخچه: براساس اجراهای گذشته برنامه است. این روش مناسب برنامه‌هایی است که با زمان تغییر نمی‌کنند یا برنامه‌هایی که روی مجموعه‌ای از منابع اجرا می‌شوند که کارایی گذشته نشان‌دهنده کارایی آینده برنامه روی همه منابع است.

- پیش بینی Testcase : برنامه برای مجموعه محدودی از موارد روی ماشین‌های نماینده آزمایش می‌شود. این روش مناسب برنامه‌هایی است که مشخصات نامشخص دارند یا اینکه با زمان تغییر می‌کنند.

تاکنون روش‌های گوناگونی برای انجام زمان بندی ارائه شده است. با استفاده از تجربیات به دست آمده در مورد استفاده از الگوریتم‌های ژنتیکی در مقاله‌های ارائه شده توسط یانگ ژائو [۹] و وینچنز دی ماتینو [۱۰] و زمان بندی اکتشافی ارائه شده برای کاربردهای تشکیل شده از وظیفه‌های مجزا [۱۱] و ترکیب نتایج آنها با روش گروه بندی کارها [۷] می‌توان روش زمان بندی بهتری برای کاربردهای تشکیل شده از کارهای مجزا ارائه نمود.

شبیه سازی کارها نیز باید از استاندارد خاصی تبعیت کند به عنوان مثال هر کار باید دارای شناسه مختص به خود، اندازه فایل ورودی، فایل خروجی، مبدا ارسالی و مقصد اجرا باشد. بهتر است که کارها از مدل اطلاعاتی ارائه شده برای ارسال کار ارائه شده توسط GGF تبعیت کنند [۸].

در این مقاله مدلی بر مبنای ساختار متمرکز و با سیاستی شبیه به پیش بینی تئوری برای پشتیبانی کیفیت سرویس، ارائه می‌شود.

۴ ارائه راه حل پیشنهادی جهت زمان بندی

در این بخش در ابتدا یک مسئله ساده شده زمان بندی را مطرح می‌کنیم. در ادامه به کمک الگوریتم‌های ژنتیکی یک راه حل برای زمان بندی منابع در مسئله ارائه می‌دهیم و بعد

نشان دهنده حالت بهینه با زمان اجرای ۲۵۹۳ و هزینه ۱۳۵۲ می‌باشد دو تخصیص اول به علت تجاوز از زمان ۴۸۰۰ ثانیه حالت‌های غیر ممکن می‌باشد حالت های سوم و پنجم نیز به علت تجاوز از حداکثر هزینه ۱۵۰۰ جزء حالت‌های غیر ممکن در نظر گرفته می‌شود، اما در حالت آخر به علت استفاده از منبع سوم به اندازه ۲۲۲۲۲۲ میلیون سیکل با سرعت ۳۰۰۰ میلیون سیکل در ثانیه این نود به اندازه ۷۴۱ ثانیه (۲۸/۶٪ از زمان اجرا) مشغول بوده و سریعترین زمان اجرا را در بین حالت‌های دیگر حاصل کرده است.

جدول (۲) هزینه و زمان حل مسئله در حالت‌های مختلف

Allocation	Cycle allocation (NC _i) in millions of cycles			Execution time (sec)	Cost (\$)
	1	2	3		
(1)	10,000,000	—	—	10,000	1,000
(2)	—	10,000,000	—	5,000	1,250
(3)	—	—	10,000,000	3,333	2,000
(1, 2)	3,333,333	6,666,667	—	3,333	1,167
(1, 3)	4,800,000	—	5,200,000	4,800	1,520
(2, 3)	—	6,666,667	3,333,333	3,333	1,500
(1, 2, 3)	2,592,593	5,185,185	2,222,222	2,593	1,352

*Allocations in bold are feasible; italics are infeasible.

اگر ما به تعداد n منبع را در اختیار داشته باشیم و یک محاسبه بتواند بین هر کدام از این ۱ تا n منبع توزیع شود تمام حالت‌های ممکن برای تخصیص این منابع دارای پیچیدگی زیر می‌باشد.

$$\sum_{k=1}^n \binom{n}{k} - 1 = 2^n - 1 \quad (10)$$

بنابراین آنالیز کردن همه این حالتها به منظور حل مسئله بهینه سازی می‌تواند بسیار هزینه‌بر باشد در این حالتها برنامه ریزی منابع در گریه باید با استفاده از روشهای

$$T = \max \left\{ \frac{NC_1}{S_1}, \frac{NC_2}{S_2}, \frac{NC_3}{S_3} \right\} \leq T_{\max} \quad (6)$$

$$C = \sum_{i=1}^3 c_i \times \frac{NC_i}{S_i} \leq C_{\max} \quad (7)$$

$$\sum_{i=1}^3 NC_i = NC \quad (8)$$

اگر ما قید هزینه را نادیده بگیریم راه حلی که زمان اجرا را کمینه می‌کند می‌تواند به صورت زیر باشد:

$$N_i = NC \times \frac{S_i}{\sum_{j=1}^3 S_j} \quad (9)$$

حال بگذارید حالت کلی را در نظر بگیریم که در آن هم حداکثر هزینه و هم حداکثر زمان مطرح می‌شود یکی از روشهای حل این مسئله بهینه سازی از روش های تخصیص منابع روش «یافتن حالتی است که $T \leq T_{\max}$ و هزینه کمینه در تخصیص» می‌باشد. حالت دیگر می‌تواند روش «یافتن حالتی از تخصیص که $C \leq C_{\max}$ بوده و T را کمینه کنیم» باشد.

فرض کنید در مثال $NC = 10^7$ میلیون سیکل $T_{\max} = 4800$ و $C_{\max} = 1500$ دلار باشد. جدول ۱ مقادیر سرعت و هزینه را برای سه منبع مثال ما را نشان می‌دهد.

جدول ۱: زمان و هزینه منابع

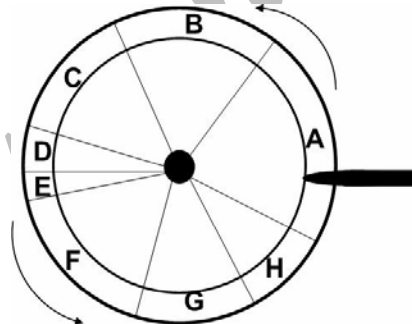
Resource	Speed (millions of cycles/sec)	Cost (dollars/sec)
۱	۱۰۰۰	۰/۱۰
۲	۲۰۰۰	۰/۲۵
۳	۳۰۰۰	۰/۶۰

برای هر امکان تخصیص، ما مسئله بهینه سازی را در حالتی که زمان T را حداقل کنیم و حالتی که فقط هزینه را $C \leq C_{\max}$ نگه داریم حل می‌کنیم. در جدول ۲ مقادیری که به صورت ایتالیک نوشته شده است حالت‌های غیر ممکن می‌باشند زیرا یا از نظر هزینه و یا زمان از حد مجاز تجاوز کرده‌اند. تخصیص‌هایی که به صورت Bold در جدول نشان داده شده حالت‌های ممکن و قابل استفاده می‌باشد خط آخر

```
double s3=3000;
double c3=0.6;
double timeThreshold = 4800;
double costThreshold = 1500;
double T = Math.Max(p.NC[0]/s1 , p.NC[1]/s2);
p.time = Math.Max( T , p.NC[2]/s3);
p.cost = (c1*(p.NC[0]/s1)) + (c2*(p.NC[1]/s2)) +
(c3*(p.NC[2]/s3));

if (p.cost>costThreshold|| p.time>timeThreshold ){
p.cost = 9999999;
p.time = 9999999;}
return p;
}
```

در ابتدا تعداد ۱۰۰ عدد از جمعیت به طور تصادفی ساخته می‌شود. در مرحله بعد در یک حلقه ۵۰ بار، هر بار دو عدد از افراد جمعیت انتخاب می‌شود تا با آمیزش این دو، دو نفر جدید تولید شوند. نحوه انتخاب دو فرد از بین جمعیت به صورت تصادفی و به کمک الگوریتم چرخ گردان مطابق شکل زیر است. هر فرد که تابع شایستگی برایش زمان اجرای کمتری (با در نظر گرفتن محدودیت هزینه) محاسبه کند، بازه زاویه‌ای بزرگتری در این دایره دارد. با چرخاندن دایره، به طور تصادفی خط نشان در یکی از بازه‌ها توقف می‌کند. به این ترتیب احتمال انتخاب شدن افرادی که دارای شایستگی بیشتری هستند در تولید نسل بعدی بیشتر است. ممکن است یک فرد چندین بار برای آمیزش و تولید نسل بعد انتخاب شود در حالی که فرد دیگری در فرایند تولید نسل بعدی اصلاً انتخاب نشده است.



شکل ۱: الگوریتم چرخ گردان. احتمال انتخاب خانه A از همه بیشتر است

عمل تولید نسل بعدی را تا جایی ادامه می‌دهیم که به یک جواب تقریباً بهینه رسیده باشیم. برای آمیزش و تولید افراد جدید از روش‌های Cross-Over و Mutation بخصوصی استفاده کرده‌ایم که از اینجا از تشریح آنها صرف‌نظر می‌کنیم. در این برنامه از تکنیک‌های ویژه‌ای به نام نخبه پروری نیز استفاده کرده ایم. به این

هوشمندانه حسی^۲ به جواب نزدیک به جواب بهینه برسیم. یک محیط واقعی بسیار پیچیده‌تر از مثال ما می‌باشد و تعداد پارامترهای منابع می‌تواند بیشتر از زمان و هزینه باشد. مسایل دیگری مانند وابستگی پروژه‌ها به هم تاخیر ارتباط برای ارسال داده‌ها و سربار هماهنگی بین آنها نیز باید مورد توجه قرار گیرد.

۴-۲- ارائه راه حل

در این بخش می‌خواهیم مسئله مطرح شده را به کمک دو روش حل کرده و نتایجشان را با هم مقایسه می‌کنیم. روش اول استفاده از تکنیک‌های مرسوم در الگوریتم‌های ژنتیک است و روش دوم، روش پیشنهادی ما یعنی رسم خلاصه فضای حالت جواب می‌باشد.

۴-۲-۱- الگوریتم‌های ژنتیکی

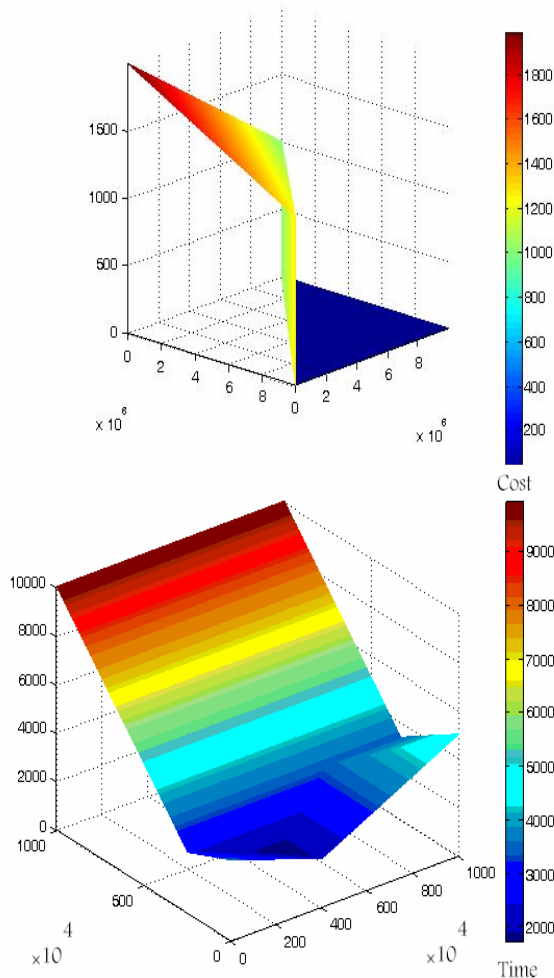
مطابق معمول همه الگوریتم‌های ژنتیکی در ابتدا باید افراد^۳ تشکیل دهنده جمعیت^۴ را مشخص کنیم. هر فرد تشکیل شده از یک ۳ تایی به شکل زیر تعریف می‌شود:
 $Individual_1 = (2222222,3333333,5555555)$
 به این معنی که ۲۲۲۲۲۲۲ سیکل پردازنده از منبع اول، ۳۳۳۳۳۳۳ سیکل پردازنده از منبع دوم و ۴۴۴۴۴۴۴ سیکل پردازنده از منبع سوم، توسط فرد شماره ۱ استفاده خواهد شد. تابع شایستگی به صورت زیر است:

public Individual FitnessFunction(Individual p)

```
{
double s1 = 1000 ;
double c1 = 0.1;
double s2=2000;
double c2=0.25;
```

² Heuristic
³ Individual
⁴ Population

در مسائلی اینچنینی عملاً محاسبه تمام حالات مسئله در زمان معقول غیر ممکن است. برای مثال ساده مسئله ما ساعتها زمان لازم است که به کمک یک کامپیوتر معمولی کل فضای حالت جواب مسئله ترسیم شود. برای همین از تکنیک الگوریتم ژنتیک در ابتدا برای حل مسئله استفاده شد.



شکل ۲: نمودار خلاصه فضای حالت هزینه (شکل بالا) و زمان (شکل پایین)

با روی هم انداختن این دو نمودار شکل ۲ و در نظر گرفتن محدودیتی که برای هزینه و زمان لازم است می‌توان مشاهده کرد که محدوده جوابهای قابل قبول در نمودار هزینه و نمودار زمان دارای همپوشانی بوده و قسمتهای مشترک جوابهای مسئله هستند (شکل ۳). مشاهده می‌کنید که جوابهای بدست آمده در این قسمت با قسمت قبل نیز مطابقت دارند.

با توجه به شکل ۳ در قسمت همپوشانی بهینه با انتخاب 200×10^4 سیکل از منبع اول (خط عمودی) و 400×10^4

معنی که افرادی که دارای بالاترین میزان شایستگی (یعنی ترکیبی از منابع که دارای کمترین زمان اجرا با وجود محدودیت هزینه) هستند، بدون آمیزش با هیچ فردی در نسل بعدی قرار می‌گیرند.

نتایج بدست آمده از اجرای الگوریتم نشان می‌دهد که در اجراهای متفاوت به طور متوسط می‌توان با در نظر گرفتن ترکیب ۳ تایی (۳۹۸۸۴۰۰، ۴۰۱۰۴۰۷، ۲۰۰۱۱۹۳) در زمان ۲۰۰۵ ثانیه و با هزینه ۱۴۹۹ دلار مسئله را حل کرد.

مقدار متوسط بدست آمده برای جواب در ۱۰۰ بار اجرای الگوریتم، با مقدار فوق تفاوت چندانی ندارد. همچنین تعداد متوسط تولید نسل برای اینکه به جواب تقریباً بهینه برسیم حدود ۵۱ بار بوده است به این معنی که ۵۱ بار کل جمعیت از نو ساخته شده است.

۴-۲-۲- حل به روش رسم خلاصه فضای حالت جواب

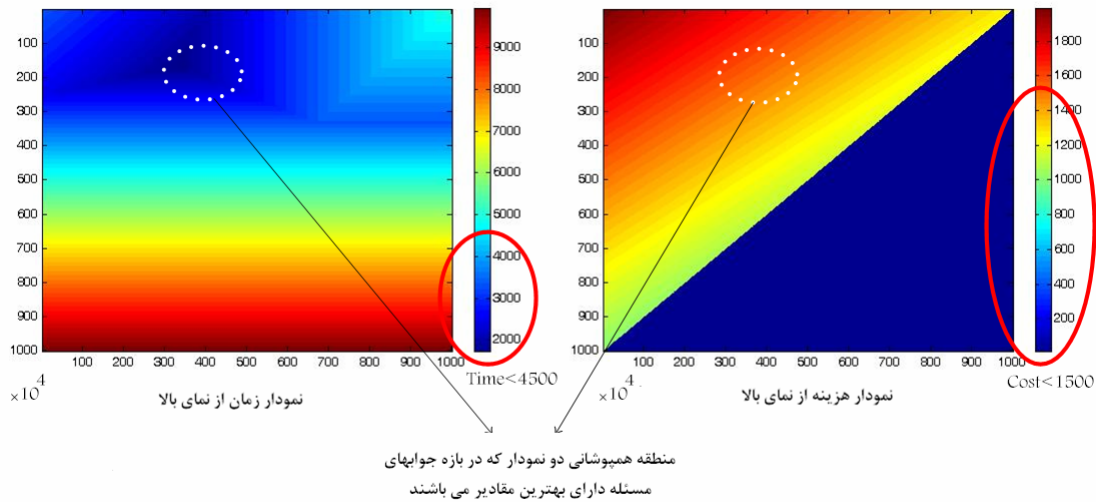
در این قسمت سعی داریم مسئله را به روشی دیگر حل کنیم و نتایج را با روش قسمت قبل مقایسه کنیم. فضای حالت جواب مسئله را در دو حالت به کمک ابزار Matlab7 رسم می‌کنیم.

حالت اول: نمودار سه بعدی که بعد اول آن میزان استفاده از منبع اول، بعد دوم میزان استفاده از منبع دوم و بعد سوم مقدار پارامتر هزینه را نشان می‌دهد. بدیهی است که مقدار استفاده منبع سوم لزومی به نمایش ندارد زیرا از مقدار اول قابل محاسبه است.

حالت دوم: نمودار سه بعدی که بعد اول آن میزان استفاده از منبع اول، بعد دوم میزان استفاده از منبع دوم و بعد سوم مقدار پارامتر زمان را نشان می‌دهد.

رسم این نمودارها به صورت خلاصه صورت می‌گیرد. به این معنی که از هر ۱۰۰۰۰ حالت (هر حالت یا جواب مسئله در نمودار معادل یک نقطه می‌باشد)، یکی برای محاسبه و رسم انتخاب می‌شود. علت اینکار اینست که محاسبه همه حالتها عملاً بسیار زمان گیر است. در حقیقت هدف اینست که فضای حالت جواب مسئله را به طور خلاصه (در زمان قابل قبول) رسم کنیم. Matlab7 این امکان را به ما می‌دهد که نمودارهای زمان گیر را در مقیاسهای بسیار کوچکتر رسم کنیم. رسم کامل فضای حالت جواب به معنای بدست آوردن کلیه جوابها بوده و در نتیجه با داشتن کلیه جوابها براحتی می‌توان جواب بهینه را بدست آورد و دیگر نیازی به روشهای هوشمند و غیره برای حل مسئله وجود ندارد. اما

سیکل از منبع دوم (خط افقی) و الباقی از منبع دیگر می‌توان مسئله را در زمان حدود ۲۰۰۰ ثانیه و هزینه ۱۵۰۰ دلار حل کرد.



شکل ۳: مشاهده قسمت همپوشانی در دو نمودار

۵ نتیجه گیری
 زیرا با خلاصه کردن فضای حالت جواب احتمال اینکه جوابهای مورد نظر ما (مقادیر کمینه) از بین رفته باشند زیاد است.

با بیشتر شدن تعداد منابع، ابعاد نمودارها و با بیشتر شدن پارامترهای کیفیت سرویس، تعداد نمودارهای مورد نیاز برای حل مسئله بیشتر می‌شود. به این ترتیب راه‌حلی توسعه پذیر جهت زمان بندی منابع در گرید بدست می‌آید که می‌تواند پارامترهای مختلف کیفیت سرویس را پشتیبانی کند.

در حالتی که مسئله خوش فرم باشد به این معنی که با خلاصه کردن فضای حالت جواب، صدمه جدی به جوابها نخورده (جوابهای بهینه از بین نرود) می‌توان از این روش برای زمان بندی استفاده نمود. در صورتی که فضای حالت جواب مسئله صاف^۵ باشد و جوابهای مسئله به صورت نقاط کمینه ناگهانی باشند، نمی‌توان از این راه حل استفاده نمود

⁵ Flat



[12] KANDAGATLA, C. Survey and Taxonomy of Grid Resource Management Systems: University of Texas, Austin, 2003.

[13] Daniel A. Menasce, Emiliano Casalicchio. "QoS in Grid Computing", IEEE INTERNET COMPUTING, AUGUST 2004

مراجع

[1] G. von Laszewski and P. Wagstorn, (2004): Tools and Environment for Parallel and Distributed Computing.

[2] Luis Ferreira, Viktors Berstis, Jonathan Armstrong, Mike Kendzierski, Andreas Neukoetter, Masanobu Takagi, Richard Bing-Wo, deeb Amir, Ryo Murakawa, Olegario Hernandez, James Magowan, Norbert Bieberstein (2003), "Introduction to Grid Computing", The IBM Redbook

[3] J. Joseph, M. Ernest, and C. Fellenstein, "Evolution of grid computing architecture and grid adoption models" IBM Systems Journal, Grid Computing, Vol. 43, No. 4, 2004.

[4] C. Kandagatla, Survey and Taxonomy of Grid Resource Management Systems, Technical Report, 2004.

[5] Foster, I. and Kesselman, K., The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1998.

[6] Rosenberg, A.L.: Optimal schedules for cycle-stealing in a network of workstations. with a bag-of-tasks workload. IEEE Transactions on Parallel Syst. 2002

[7] Nithiapidary Muthuvelu, Junyang Liu, Nay Lin Soe, Srikumar Venugopal, Anthony Sulistio, Rajkumar Buyya: A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids. ACSW Frontiers 2005: 41-48

[8] E. Stokes, L. Flon: Job Submission Information Model (JSIM), 2004, Global Grid Forum

[9] Gao, Y., Rong, H., Huang, J.Z., Adaptive grid job scheduling with genetic algorithms, Future Generation. Computer Systems 21 (1), pp. 151-161, 2005

[10] Martino, Vincenzo Di, Sub Optimal Scheduling in a Grid Using Genetic Algorithms, April 22 - 26, 2003, International Parallel and Distributed Processing

[11] C. Weng and X. Lu, "Heuristic scheduling for bag-of-tasks applications in combination with QoS in the computational grid," vol. 21, pp. 271, 2005.

Surf and download all data from SID.ir: www.SID.ir

Translate via STRS.ir: www.STRS.ir

Follow our scientific posts via our Blog: www.sid.ir/blog

Use our educational service (Courses, Workshops, Videos and etc.) via Workshop: www.sid.ir/workshop