

## الگوریتم فرهنگی برای شبکه توزیعی متمرکز برای تخصیص زیر پرس وجو

ناصر سیه چهره<sup>۱</sup>، مهدی رضاپور<sup>۲</sup>، محمد رضا میبیدی<sup>۳</sup>

<sup>۱</sup> دانشجوی کارشناسی ارشد مهندسی نرم افزار، دانشگاه آزاد قزوین

nasser\_S1@yahoo.com

<sup>۲</sup> دانشجوی دکترای مهندسی نرم افزار، دانشگاه صنعتی امیرکبیر

mmrezapour@aut.ac.ir

<sup>۳</sup> دانشکده مهندسی کامپیوتر و فن آوری اطلاعات، دانشگاه صنعتی امیرکبیر

mmeybodi@aut.ac.ir

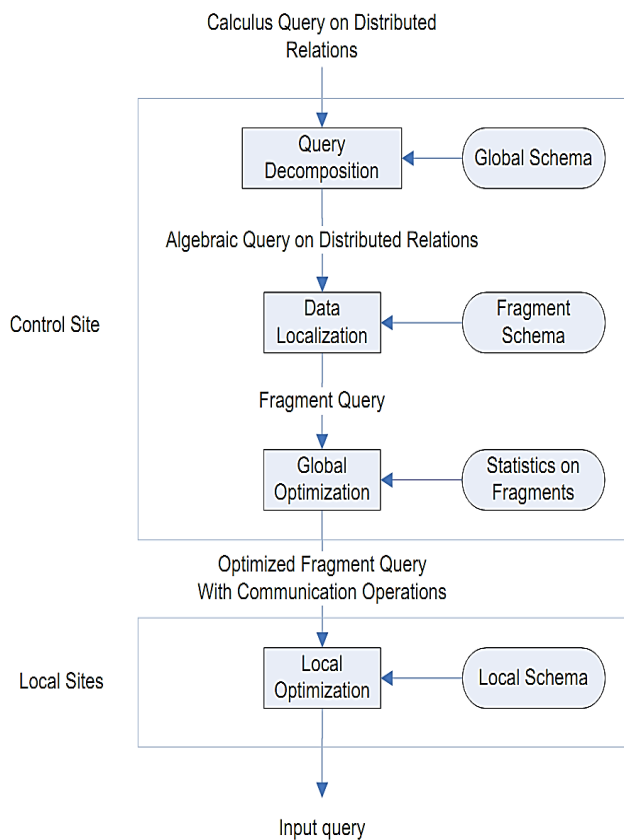
چکیده پردازش پرس و جو مفهوم مهمی در حوزه ی پایگاه داده ی توزیع شده می باشد. که در همین راستا پردازش پرس و جوی توزیع شده با توجه به اینکه هدف کم کردن یا به حداقل رساندن هزینه ی عملیاتی می باشد، فاکتور مهمی در کل کارایی سیستم پایگاه داده ی توزیع شده می باشد. ما در این مقاله از معماری شبکه توزیعی متمرکز استفاده می کنیم، براساس این نوع معماری سرور باید تصمیم گیری کند چگونه پرس وجو را شکسته و به کدام کلاینت با کمترین هزینه ی عملیاتی فرستاده و زمان پاسخ کمینه را دریافت نماید. هدف از این مقاله نحوه ی تخصیص زیر پرس وجو ها به سایت ها مختلف با شایستگی بهتری می باشد. با مدل کردن ویژگی های سایت به کروموزم ها که هر ژن کروموزم یکی از ویژگی سایت می باشد که بصورت پویا مثل ترافیک شبکه، تاخیر، پردازش محلی در حال تغییر است باعث می شود تخصیص زیرپرس وجوها هوشمندانه شود در نتیجه زمان پاسخ کمینه شود. که الگوریتم فرهنگی با روشهای شمارشی و الگوریتم ژنتیک مقایسه شده که نتایج قابل توجهی داشته است.

کلید کلیدی الگوریتم تخصیص زیر پرس و جو، الگوریتم های قطعی، الگوریتم فرهنگی، جهش، زمان پاسخ کمینه

### ۱- مقدمه

در این نوع معماری، سرورها از لحاظ عملکردی به دو بخش تقسیم می شوند، یکی سرورهای داده ای که این نوع سرور ها بیشتر در سیستم های شی گرا مورد استفاده قرار می گیرند و دیگری سرورهای تراکنشی که این نوع سرورها بیشتر در سیستم های رابطه ای مورد استفاده قرار می گیرند. در سرورهای تراکنشی که به آنها سرورهای پرس و جو هم گفته می شود روش کار بدین صورت است که درخواست های کاربران به این سیستم ها ارسال می شود و نتیجه در این سرورها تولید و به کاربر ارسال می شود. درخواست های کاربران به صورت دستورات RPC و دستورات SQL ارسال می شود. برای ارسال و دریافت نیاز به نرم افزارهایی مانند ODBC و JDBC هست که وظیفه ارتباط با سرور و ارسال پرس و جوها و دریافت نتایج را به عهده دارد. برای پیاده سازی سرورهای تراکنشی از تعدادی پردازنده حافظه مشترک دارند استفاده می شود که معمولا برای بهبود پردازنده ها از چندین مورد استفاده قرار می گیرند.

پایگاه داده ها در این نوع معماری بر حسب کاربری به دو بخش Back-end و Front-end تقسیم می شود، Back-end وظیفه بررسی و کنترل دسترسی ها، بررسی و بهینه سازی پرس و جوها و کنترل همزمانی ها و سالم بودن پایگاه داده ها را به عهده دارد، Front-end شامل ابزارهایی برای نمایش و زیبا سازی نتایج پرس و جوها مثل ابزارهای تولید فرم ها و ابزار های گزارش گیری می باشد [1]. برای ایجاد ارتباط درست میان دو قسمت فوق نیازمند یک لایه و بستر می باشیم. این بستر می تواند از دو طریق دستورات SQL و یا API ها برقرار شود. در مقایسه این نوع معماری با معماری متمرکز با استفاده از پردازنده های مرکزی می توان افزایش میزان کاربری سیستم با توجه به هزینه، آسان تر شدن گسترش و توزیع منابع، تولید واسطه های کاربر بهتر و راحت تر شدن نگهداری سیستم را نام برد.



شکل (۱): بهینه سازی پرس و جوی توزیعی

ورودی در شکل فوق یک پرس وجو بر روی داده توزیع شده در حساب رابطه ای می باشد. لایه های اصلی برای پرس وجوی توزیع شده در داخل یک توالی عملیات محلی بهینه شده قرار گرفته است، هر عامل بر روی یک پایگاه داده محلی است. این لایه ها تجزیه توابع پرس وجو، محلی کردن داده، بهینه سازی سراسری پرس وجو و بهینه سازی محلی پرس وجو را اجرا می کنند. تجزیه محلی کردن پرس وجو داده مربوط به پرس وجو باز نویسی می باشد. سه لایه اول بوسیله یک محل مرکزی انجام می شود و مورد استفاده اطلاعات سراسری است. بهینه سازی محلی توسط مکان های محلی انجام شده است.

سه نوع از انواع معمول الگوریتم ها برای بهینه سازی Join-ordering الگوریتم های قطعی، ژنتیک و تصادفی هستند [7]. زمان پردازش محلی شامل موارد زیر می باشد: نوع عملیات، اندازه ی رابطه ورودی، سرعت CPU و سرعت ورودی و خروجی سایت منتخب پس،

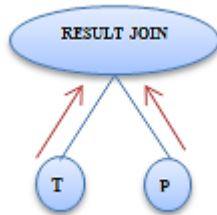
برای بهینه سازی پرس و جو در یک محیط کلاینت/سرور باید اطلاعات کافی در مورد مکان های داده ها داشته باشیم تا بتوانیم با توجه به آن بهترین روش را برای دسترسی به داده ها پیدا کنیم. مهمترین مسئله در اینجا تجزیه کردن پرس و جوها می باشد. پردازش پرس و جوی توزیع شده در سه مرحله ی فاز پردازش محلی، فاز ساده سازی و فاز پردازش انجام می گیرد، در فاز پردازش محلی گزینش ها و انتخاب انجام می گیرد و در فاز ساده سازی از یک کاهنده برای کم کردن اندازه ی رابطه ها استفاده می شود و در فاز پردازش نهایی نتایج رابطه ها در یک جا جمع شده و نتیجه ی نهایی پرس و جو ایجاد و فرستاده می شود. در اینجا به این خاطر که از موازی سازی کمتری استفاده می شود پدیده ی سربار زیاد انتقال را داریم [2]. در پردازش پرس وجوی توزیع شده، یکسری عملیات تحت عنوان عملیات عمومی نیز وجود دارد که این عملیات شامل قسمت بندی یک رابطه در داخل قطعه ها، اجتماع قطعه ها از کل رابطه و انتقال یک رابطه / قطعه از یک پایگاه داده به دیگر پایگاه داده می باشد [3]. برای ساده سازی انتقال داده و برای مرحله ی پردازش پرس و جوی توزیع شده از نیم پیوند و پیوند متوالی استفاده می شود، در نیم پیوند عملیات پیوند برای کاهش پردازش پرس و جو استفاده می شوند و با استفاده از احیا کننده ی پیوند یک پرس و جو در داخل پیوند های متوالی ترجمه می شود و هر پیوند بوسیله ی یک عملگر رابطه ای به طور محلی با توجه به محل عملگر های دیگر پیاده سازی می شود. همچنین می توان از ترکیب پیوندها و نیم پیوندها برای بهبود پردازش پرس و جوی توزیع شده استفاده کرد [4,5].

## ۲- جزئیات مدل های هزینه ی پردازش پرس وجو

Valduriez یک متدلوژی پردازش پرس وجو توزیع شده شرح داده است که در شکل ۱ نشان داده شده است [6].

ورودی در شکل فوق یک پرس وجو بر روی داده توزیع شده در حساب رابطه ای می باشد. لایه های اصلی برای پرس وجوی توزیع شده در داخل یک توالی عملیات محلی بهینه شده قرار گرفته است، هر عامل بر روی یک پایگاه داده محلی است. این لایه ها تجزیه توابع پرس وجو، محلی کردن داده، بهینه سازی سراسری پرس وجو و بهینه سازی محلی پرس وجو را اجرا می کنند. تجزیه محلی کردن پرس وجو داده مربوط به پرس وجو باز نویسی می باشد. سه لایه اول بوسیله یک محل مرکزی انجام می شود و مورد استفاده اطلاعات سراسری است. بهینه سازی محلی توسط مکان های محلی انجام شده است

برای محاسبه ی هزینه ارتباطی فرض می کنیم گرافی مانند شکل زیر داریم که در واقع موقعی هیچ رابطه ای برای اتصال وجود نداشته باشد در هر کدام از سایت ها عملیات اتصال به گره بالایی صورت می گیرد.



شکل (۲): گراف با دو گره که هر نود معرف یک سایت می باشد



شکل (۳): گراف با یک گره سمت چپ یا راست که هر نود معرف یک سایت می باشد.

= تعداد برگهای زیر درخت و جمع تمام زیر پرس و جوی های در سایت  $t$  و جمع تمام زیر پرس و جوی های در سایت  $P$

هزینه ی ارتباطی قسمت اول (وجود زیر پرس و جوی  $J$  در سایت  $P$  برای برگ اول توسط پرس و جوی  $K$ ) + (وجود زیر پرس و جوی  $J$  در سایت  $T$  برای برگ اول توسط پرس و جوی  $K$ ) جمع همه رابطه و یا نتیجه میانی  $i$  = هزینه ی ارتباطی قسمت دوم + (انتقال نتیجه میانی  $i$  توسط زیر پرس و جوی  $J$  در برگ اول و یا دوم و یا هردو در سایت  $P$ ) (نتیجه  $i$  بوسیله ی زیر پرس و جوی  $J$  در برگ اول و یا برگ دوم سایت  $P$  و یا هردو توسط پرس و جوی  $K$ ) اگر هردو پیوند قبلی و اجرای آن در یک سایت باشد رابطه ی هزینه ارتباطی قسمت دوم صفر می شود.

$$Total\ cost = \sum_j localP_j^k + COMM_j^k \quad (1)$$

که  $J$  نشان دهنده ی تعداد زیر پرس و جو ها و  $k$  نام خود پرس و جوی افزاشده می باشد.  $LocalP_j^k$  نشان دهنده ی زمان پردازش محلی  $J$  زیر پرس و جو برای پرس و جوی  $k$  می باشد.  $COMM_j^k$  یعنی زمان انتقال رابطه مورد نظر که در واقع پرس و جوی  $k$  به  $J$  زیر پرس و جو افزاشده است.

$$subQ_{jt}^k = \begin{cases} \text{صفر عدم وجود زیر پرس و جوی } J \text{ در سایت } t \text{ برای پرس و جوی } k \\ \text{یک وجود زیر پرس و جوی } J \text{ در سایت } t \text{ برای پرس و جوی } k \end{cases}$$

## ۲-۱- محاسبه ی پردازش محلی

برای محاسبه پردازش محلی، بستگی به نوع عملیات، سرعت CPU، اندازه ی رابطه و سرعت ورودی سایت منتخب می باشد. پردازش محلی از دو بخش ورودی/خروجی و پردازش تشکیل شده است که به صورت زیر محاسبه می شود.

= مجموع تمام زیر پرس و جو های سایت  $t$

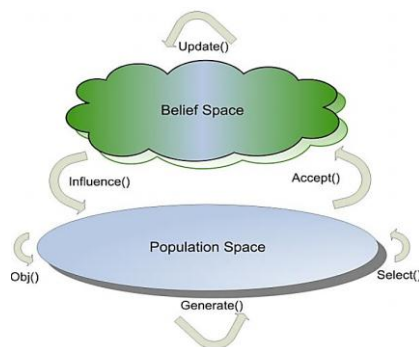
پردازش محلی (ورودی/خروجی) (تعداد بلاک رابطه ی  $i$  که توسط زیر پرس و جوی  $J$  رابطه ی  $K$ ) + (رابطه یا رابطه ها  $i$  توسط زیر پرس و جوی  $J$  برای پرس و جوی  $k$ )

= مجموع تمام زیر پرس و جو های سایت  $t$

پردازش محلی (cpu) (تعداد بلاک رابطه ی  $i$  که توسط زیر پرس و جوی  $J$  رابطه ی  $K$ ) + (رابطه یا رابطه ها  $i$  توسط زیر پرس و جوی  $J$  برای پرس و جوی  $k$ )

## ۲-۲- محاسبه ی هزینه ی ارتباطی

می باشد. فضای باور اطلاعات و تجربیات را ذخیره می کنند. که بوسیله منحصر بفردها در طول پردازش تکاملی مورد استفاده قرار می گیرد [10]. همانطور در شکل ۴ نشان داده شده، منحصر بفردها براساس تابع کارایی ارزیابی می شوند. بعضی منحصر بفردها از فضای جمعیت بوسیله تابع پذیرش انتخاب می شوند و اطلاعات فضای باور را بروز رسانی می کنند.



شکل (۴): چارت الگوریتم فرهنگی

همچنین شبه کد الگوریتم مانند زیر می باشد :

```
Begin
t=0;
Initialize Population Space POP(t);
Initialize Belief Space BLF(t);
Repeat
Evaluate Population POP(t);
Update(BLF(t), Acceptance (POP(t)));
Variation(POP(t), Influence(BLF(t)));
t=t+1;
Select POP(t) from POP(t-1);
Until termination condition achieved;
End
```

### ۳-۱- شرح الگوریتم پیشنهادی

ابتدا فضای جمعیت بصورت تصادفی مقداردهی اولیه می شود. هریک از اعضای فضای جمعیت نشان دهنده یک راه حل برای مسئله می باشد. درواقع هریک از راه حل ها یک کروموزوم نامیده می شود. کروموزوم ها براساس شایستگی آنها ارزیابی می شوند. نمایش کروموزوم ها بصورت اعداد صحیح می باشد. تعداد اعضای فضای جمعیت به تعداد عملیات ها پرس و جو و و تعداد سایت ها می باشد. پس اندازه ی فضای جمعیت به تعداد سایت ها موجود در شبکه

### ۳- زمان پاسخ

برای محاسبه ی زمان پاسخ در اینجا، در سیستم پایگاه داده ی توزیعی، تجزیه ی یک پرس و جو به یک زیر پرس و جو این امکان را می دهد که فرآیند پردازش به صورت موازی انجام شود و همچنین رابطه های میانی می توانند برای پردازش به سایت های مورد نیاز به صورت موازی فرستاده شوند. در کل دو نوع اجرای موازی وجود دارد: یکی عملیات درونی و دیگری عملیات داخلی، در عملیات درونی، بروی عملیات انتقال انفرادی خط لوله به صورت موازی صورت می گیرد که هر کدام از سایت ها به صورت موازی کار می کنند. در عملیات داخلی، چندین زیر پرس و جو در یک پرس و جو می توانند به صورت موازی اجرا شوند [8].

= زمان پاسخ زیر پرس و جو ی j توسط پرس و جو ی k

(پردازش محلی زیر پرس و جو ی j توسط پرس و جو ی k)  
+ هزینه ی ارتباطی زیر پرس و جو ی j که توسط پرس و جو ی k در برگ اول p (زمان پاسخ زیر پرس و جو ی j توسط پرس و جو ی k)

لازم به ذکر است قسمت سوم رابطه فوق به صورت تابع بازگشتی می باشد یعنی از برگ شروع کرده تا به ریشه می رسد و زمان های پاسخ را جمع می کند.

در اینجا زمان های پاسخ مختلفی به وجود می آید و حالت های مختلفی باید بررسی شود اما با رشد درجه ی موازی سازی زمان پاسخ کمینه می شود و با کمینه شدن زمان کل و با به کارگیری بهبود منابع، بهره ی ورودی سیستم افزایش پیدا می کند.

### ۴- الگوریتم فرهنگی

الگوریتم های فرهنگی توسط (Reynolds, 1994) معرفی شده است [9]. که شامل دو مولفه فضای باور و فضای جمعیت می باشد. فضای جمعیت، شامل مجموعه ای منحصر بفرد که کاندید جواب مسئله

برای طراحی فضای جمعت به تعداد سایت کروموزم وجود دارد و هر کروموزم از چهار ژن تشکیل شده است. و هر ژن به ترتیب نشان دهنده ی اولویت، تاخیر، هزینه ی ارتباطی و زمان پردازش می باشد. که یک کروموزم به صورت تصادفی تولید می شود و تشکیل جمعیت را می دهند.

### ۴-۳- طراحی فضای باور

ابتدا، یک ماتریس چهار در چهار در نظر گرفته می شود که هر سطر آن یک کروموزم می باشد و با اعداد ۱ و ۱۰۰ پر می شود سپس در طول تولید نسل براساس بهترین شایستگی کروموزم ها به روز شده می گیرد. ۲۰٪ از منحصرفردهای فضای باور، بعنوان منحصرفردهای فضای جمعیت انتخاب می شوند. در واقع این ۲۰٪ برای عمل crossover و جهش انتخاب می شوند. که در شکل زیر مشاهده می کنید.

۱	۱۰۰	۱۰۰	۱۰۰
۱	۱۰۰	۱۰۰	۱۰۰
۱	۱۰۰	۱۰۰	۱۰۰
۱	۱۰۰	۱۰۰	۱۰۰

شکل ۵- مقدار دهی اولیه ی فضای باور

### ۴-۳- طراحی تابع پذیرش

تابع پذیرش در واقع تصمیم گیری می کند که کدام منحصرفردها روی فضای باور تاثیر پیدا کنند. که در واقع درصدی از منحصرفردهای فضای جمعیت می باشند. یعنی منحصرفردهایی با شایستگی بالا جایگزین منحصرفردهایی با شایستگی پایین می شود.

### ۴-۴- طراحی تابع موثر

بعد از اینکه منحصرفردها بعد از k تا تولید نسل ارزیابی شده اند، ۱۵٪ از منحصرفردهایی از فضای باور جایگزین منحصرفردها با شایستگی پایین می شود.

### ۵-آزمایش ها

در آزمایش های بعمل آمده به این نتیجه رسیده ایم برای رسیدن به نقطه بهینه پارامتر تولید نسل باید برابر ۶۰ و الگوریتم محدودیتی در تعداد سایت ها ندارد و با کمترین زمان ممکن بهترین سایت ها را برای ارسال زیرپرس و جو در نظر می گیرد. در این نسل هم زمان پاسخ و

می باشد. عمل جهش در انتهای کار بصورت تصادفی روی یکی از ژن های کروموزم صورت می گیرد. برای Crossover کردن و تولید مثل از روش One-point بکار گرفته می شود. برای انتخاب کروموزم ها برای عمل Crossover و جهش از تابع Roulette Wheel Selection استفاده می شود. در این الگوریتم به کروموزم ها با شایستگی بالا توجه شده است به عبارت دیگر به نخبه گرایی توجه زیادی دارد. بخاطر اینکه الگوریتم زود به همگرایی برسد که باعث می شود زمان پاسخ زیرپرس وجو کمینه می شود. برای اینکار ما در این الگوریتم فشار انتخاب را زود کاهش داده ایم. بعد از مدتی تولید نسل الگوریتم به همگرایی رسیده و بهترین سایت های کاندید برای ارسال زیرپرس و جو انتخاب می شود.

Input: N site with Different Qualification

Output site with ascending priority

Method:

Do

{

Generation =0;

Generate Initial Population of site plans

Generate Initial Belief space

Compute Fitness for each Chromosome

Compute Sort to Ascending

For each  $i, j; i, j \leq 4$

Update Belief Space()

{

if Belif space[i,j]<pop[i,j]

Belif space[i,j]=pop[i,j]

if Belif space[i,j]>pop[i,j]

Belif space[i,j]=pop[i,j]

Create\_child(){

If (Belif space[i,j]<pop[i,j])

child[i,j]=random(Belif space[i,j],pop[i,j])

else if Belif space[i,j]>pop[i,j]

child[i,j]=random(Belif space[i,j], pop[i,j])

else

child[i,j]=random(Belif space [i,j],100)

}

Replace\_child();

While (n>0)

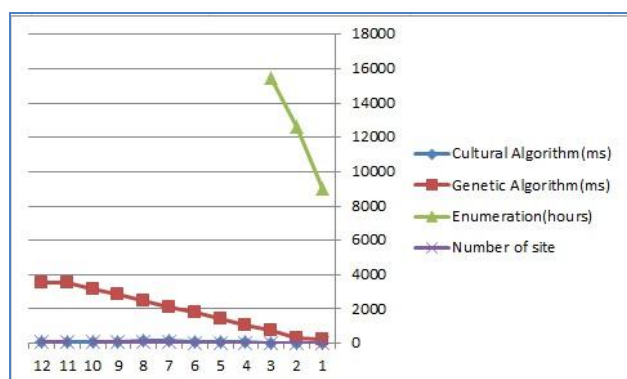
}

### ۴-۲- فضای جمعیت

Number of Site	Enumeration on (Hours)	Genetic Algorithm (Second)	Cultural Algorithm (millisecond)
3	2.5	20	3
5	Up 2.5	30	7
10	Up 43	73	24
20	----	108	73
30	----	143	64
40	----	178	100
50	----	213	186
60	----	248	170
70	----	283	107
80	----	317	77
90	----	352	94

جدول (۱): مقایسه زمان پاسخ شمارشی، الگوریتم ژنتیک و فرهنگی

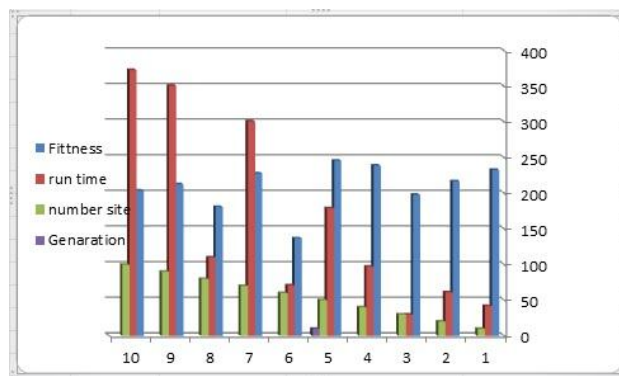
همچنین این روند بهبود زمان پاسخ در نمودار زیر نشان داده است.



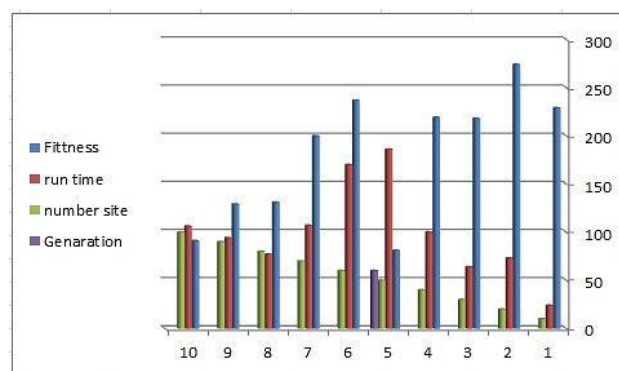
۵- نتیجه گیری  
شکل (۸): نمودار زمان پاسخ

در روش پیشنهادی برای حل مسئله نحوه تخصیص زیر پرس وجوها به سایت های مختلف در فضای شبکه می باشد. در واقع زمان پاسخ کمینه می شود. بعلاوه اینکه الگوریتم فرهنگی یک الگوریتم متاهوریستیک می باشد امکان موازی سازی در سایت مختلف را دارا می باشد. و باعث می شود زیرپرس وجوهای بیشتری را به سایت ها داده و باعث صرفه جویی در زمان پاسخ می شود. و به نسبت روش های با شمارش جامع و الگوریتم ژنتیک از کارایی و عملکرد بهتری برخوردار می باشد. ولی می توان برای بدست آوردن جواب بهینه تر می توان پارامترهای این روش را با دقت بیشتری انتخاب نمود تا به جواب با کارایی بالاتری دست پیدا کنیم.

شایستگی نیز کمینه می شود در نسل های ابتدایی روند خوبی نداشته ولی به مرور زمان روند بهتری پیدا می کند که در نمودار آمده است.



شکل (۶): جواب مسئله در نسل ابتدایی



شکل (۷): جواب مسئله در نسل پایانی

## ۶-مقایسه

در مقایسه ها بعمل آمده برای تعداد سایت های مختلف و محاسبه آن بصورت شمارشی و الگوریتم ژنتیک و الگوریتم فرهنگی زمان پاسخ کمینه برای الگوریتم فرهنگی نتیجه قابل توجهی داشته است. و حتی با افزایش تعداد سایت اختلاف سرعت زمان پاسخ بیشتر می شود که در جدول زیر مشاهده می شود.



## مراجع

- [1] Alom. B.M. Monjurul, Henskens.Frans And Hannaford.Michael, "Query Processing and Optimization in Distributed Database Systems", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.9, September 2009 .
- [2] C. Wang and M.-S. Chen, "On the Complexity of Distributed Query Optimization," IEEE vol. 8, pp. 650-662, 1994.
- [3] C. Liu and C. Yu, "Performance Issues in Distributed Query Processing," IEEE, vol. 4:8, pp. 889-905, 1993.
- [4] M. S. Chen and P. S. Yu, "Combining Join and Semi Operations for Distributed Query Processing," IEEE Trans. of Knowledge & Data Engineering, vol. 5:3, pp.534-542, 1993.
- [5] M. S. Chen and P. S. Yu, "A Graph Theoretical Approach to Determine a Join Reducer Sequences in Distributed Query Processing," IEEE Trans. of Knowledge & Data Engineering, vol. 6:1, pp. 152-165,1994.
- [6] M. Tamer Özsu, Patrick Valduriez, "Principles of Distributed Database systems, third Edition", Prentice Hall, ISBN 0-13-659707-6, 2011.
- [7] Kristina Zelenay, "Query Optimization", ETH Zürich, Seminar Algorithmen für Datenbanksysteme 2005.
- [8] Cornell, D.W. & Yu, P.S, " On optimal site assignment for relations in the distributed database environment". IEEE Transactions on Software Engineering, 15 (8),1004-1009. 1989
- [9] Reynolds, R. G., "An Introduction to Cultural Algorithms", Proceedings of the Third Annual Conference on Evolutionary Programming, San Diego, California, pp. 131-139, February 24-26, 1994.
- [10] X. Yuan, Y. Yuan. " Application of cultural algorithm to generation scheduling of hydrothermal systems", Energy Conversion and Management, vol. 47, no. 15-16, pp. 2192-2201,2006