مجله علمی-پژوهشی رایانش نرم و فن‌آوری اطلاعات

جلد ۵، شماره ۲، تابستان سال ۱۳۹۵ شاپا: ۱۰۰۶-۲۳۸۳

بهبود سرعت نگهداری دید در پایگاه‌داده‌تحلیلی

عفیفه کریمی‌مصدق[۱]، نگین دانشپور [*۲]

۱- کارشناسی ارشد، گروه کامپیوتر، دانشگاه ازاد اسلامی قزوین، قزوین، ایران، Karimi.mosadegh@yahoo.com

*۲- نویسنده مسئول: استادیار، دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران، ndaneshpour@srttu.edu

چکیده- در سال‌های اخیر به‌دلیل استفاده از پایگاه‌داده‌تحلیلی موضوع مربوط به حفظ و نگهداری دید ذخیره شده مورد اهمیت است. برای دسترسی سریع به داده‌ها،پایگاه‌های داده‌تحلیلی، داده‌های لازم را از منابع مختلف جمع‌آوری کرده و به فرم دید ذخیره شده در خود ذخیره می‌کنند. این امر باعث سرعت بالای پایگاه‌داده‌تحلیلی در پاسخ به پرس‌وجوها می‌شود. وقتی داده‌ها در منابع مختلف تغییر می‌کنند، دیدهای ذخیره شده نیز باید به‌روز شوند که این امر منجر به مطرح شدن موضوع نگهداری دید می‌شود. در این زمان ارائه الگوریتم‌هایی به‌منظور نگهداری دید با هزینه بهینه، مطرح می‌شود. الگوریتمی که در این مقاله ارائه می‌شود، ترکیب یک روش ریاضی با الگوریتم فراابتکاری فرهنگی است که باعث کاهش زمان جستجو و بهینه شدن هزینه دسترسی به داده‌ها در پایگاه‌داده‌تحلیلی است. الگوریتم فرهنگی از یک فضای باور مناسب استفاده می‌کند که شامل چندین رابطه نگهداری افزایشی دید است. هر بار که بهترین پاسخ در انتهای هر نسل به‌دست آید، در فضایی به اسم فضای باور ذخیره می‌شود. آزمایش‌ها نشان می‌دهند که الگوریتم فرهنگی در مقایسه با روش‌های قبلی و الگوریتم‌های باکتریال و زنبور عسل و الگوریتم مبتنی بر تدریس و یادگیری از سرعت بالاتری به‌منظور به‌روزرسانی دید افزایشی برخوردار است.

واژه‌های کلیدی: پایگاه‌داده‌تحلیلی، الگوریتم فرهنگی، درخت دلتای بهینه، نگهداری افزایشی دید.

۱- مقدمه

پایگاه‌داده‌تحلیلی[۱]، معمولا ازپایگاه‌داده عملیاتی و یا سایر منابع داده‌ای توزیع شده سازمان‌ها و ارگان‌های متفاوت ایجاد می‌شود. پایگاه‌داده‌تحلیلی شامل داده‌هایی است که برای انجام تصمیم‌گیری‌ها و تحلیل‌ها مناسب است. پایگاه‌داده‌تحلیلی بستر مناسبی فراهم می‌آوردکه داده‌ها به‌منظور پاسخ‌گویی به پرس‌وجوهای تحلیلی به صورت بایگانی شده، سر جمع شده و سازمان‌یافته، به فرم دید ذخیره شده[۲] ذخیره شوند.

در محیط‌های پایگاه‌داده‌تحلیلی مفهوم دید ذخیره شده اهمیت بسیاری دارد. بابه‌کارگیری دید ذخیره شده درخواست‌ها می‌توانند به جای این‌که به چندین منبع دسترسی داشته باشند به‌طور مستقیم به دیدهای ذخیره شده دسترسی داشته باشند، که این باعث دسترسی سریع به داده‌ها می‌شود و سرعت را بالا می‌برد. وقتی منابع داده اصلی تغییر می‌کنند دیدهای ذخیره شده در پایگاه‌داده‌تحلیلی نیز بایدبه‌روز شوند. از آن‌جایی که افزایش به‌روزرسانی‌های دید ممکن است سربار زیادی را تحمیل کند ضروری است که به‌روزرسانی دیدها با دقت بالایی انجام شود. به همین دلیل،موضوع نگهداری دید[۳] ذخیره شده مطرح می‌شود. در بسیاری از سیستم‌ها به‌روزرسانی دیدها در زمانی اتفاق می‌افتد که سیستم بیکار است مثلا از ۱۲ شب تا ۶ صبح. پس اگر به‌روزرسانی دیدها با سرعت بالاتری انجام شود، می‌توان در زمان مشخص تعداد دیدهای بیشتری را به‌روز کرد.

دو روش برای نگهداری دید ذخیره شده وجود دارد: یکی از روش‌ها محاسبه مجدد دیدها است. این روش نیاز به ذخیره‌سازی‌های اضافی دارد و هزینه نگهداری آن بالا است، در بعضی مواقع هم به دلیل محدودیت‌های ذخیره‌سازی، غیر قابل پیاده‌سازی است. روش دیگر استفاده از تکنیک نگهداری افزایشی دید[۴] است که در سال‌های اخیر بیشتر مورد استفاده قرار می‌گیرد [۱-۹]. از آن‌جایی‌که اندازه تغییرات، کوچک‌تر از اندازه روابط اصلی و دیدها است، می‌توان نتیجه گرفت محاسبه تغییرات، بسیار ارزان‌تر

[1]Data warehouse
[2]Materialized View
[3]View Maintenance
[4]Incremental View Main tenance

www.SID.ir

Archive of SID

از انجام محاسبات مجدد است. مقدار داده افزایشی برای دیدهای یکسان که از متدهای مختلف نگهداری افزایشی استفاده می‌کنند متفاوت است و این امر منجر به هزینه‌های نگهداری دید متفاوت می‌شود.

هدف از این مقاله، ارائه الگوریتمی است که با افزایش سرعت نگهداری دید، هزینه نگهداری دید افزایشی را کاهش دهد. این الگوریتم بر پایه محاسبات هوش مصنوعی است. به این منظور، در این مقاله از الگوریتم فراابتکاری فرهنگی[1] [۱۰، ۱۱] استفاده شده تا زمان یافتن یک رابطه‌ی بهینه برای نگهداری افزایشی دیدکاهش داده شود. مشاهده می‌شود که الگوریتم پیشنهادی نسبت به الگوریتم‌های فراابتکاری مبتنی بر تدریس و یادگیری[2] و الگوریتم ژنتیک[3] و الگوریتم زنبور عسل[4] و باکتریال[5] که پیشنهاد قبلی نویسندگان این مقاله است [۱۲]، از سرعت بالاتری برخوردار است.

ساختار مقاله بدین شرح است: بخش دو، به بیان کارهای گذشته اختصاص داده شده است. در بخش سه، ابتدا عبارت نگهداری دید معرفی می‌گردد[۱۳]، سپس یک روش نگهداری دید افزایشی معرفی می‌گردد که روش دلتای بهینه نام دارد. این روش از طریق کاهش تعداد دسترسی به برخی روابط تکراری هزینه نگهداری دید را کاهش می‌دهد [۱۳]. از معایب این روش این است که، زمان زیادی را صرف پیدا کردن رابطه بهینه برای نگهداری دید افزایشی می‌کند. در نهایت در بخش چهارم روش پیشنهادی این مقاله مطرح می‌گردد. در بخش پنج نتایج شبیه‌سازی برای مقایسه و ارزیابی، آورده شده است.

۲-کارهای گذشته

ایده‌ها و الگوریتم‌های مختلفی برای نگهداری دید افزایشی وجود دارد [۲][۵][۷، ۸][۱۳- ۲۸] که هدف آن‌ها کاهش هزینه، یعنی کاهش محاسبات و افزایش سرعت نگهداری دید است.

یکی از ایده‌ها [۱۷] این است که دیدها را بر حسب اندازه تغییراتشان به‌صورت صعودی مرتب کرده و به‌صورت FIFO[6] دیدها به‌روزرسانی شوند. در این‌صورت هزینه نگهداری کاهش می‌یابد.

روش دیگر نگهداری دید استفاده از عامل‌ها است [۱۸]. عامل‌ها می‌توانند به‌صورت موازی اجرا شوند و پردازش به‌روزرسانی‌ها را به‌صورت موازی انجام دهند و این باعث می‌شود که سرعت به‌روزرسانی‌ها افزایش یابند.

از دیگر روش‌های نگهداری دید، روش‌های مبتنی بر موازنه هستند [۱۹][۲۰] [۲۱][۲۲]. این روش‌ها خطاهایی را که ممکن است در به‌روزرسانی همزمان دیدها رخ دهند از بین می‌برند.

ECA و Strobe، [۱۹][۲۰] دو روش مبتنی بر جبران هستند. تعداد پرس‌وجوهایی که الگوریتم ECA برای نگهداری دید ارسال می‌کند از مرتبه دو است، و این تعداد برای الگوریتم Strobe از مرتبه فاکتوریل است. ECA پرس‌وجوهای جبران را به منبع ارسال می‌کند و از این طریق تاثیربه‌روزرسانی‌های همزمان را که باعث خطا می‌شود برطرف می‌کند. این الگوریتم تنها برای یک منبع داده، عملکرد مناسبی دارد. الگوریتمStrobe [۲۰] مشکل روش قبل را برطرف کرده و روی چند منبع داده کاربرد دارد. این الگوریتم به‌روزرسانی‌ها را به ترتیبی که می‌رسند پردازش می‌کند. مشکل الگوریتم Strobe این است که اگر به‌روزرسانی‌های منابع ادامه داشته باشد و الگوریتم به یک حالت سکون دست پیدا نکند دید به‌روز نمی‌شود. این الگوریتم منتظر می‌ماند تا زمانی که به‌روزرسانی‌ها فروکش کنند و فقط آن زمان است که نتایج همه به‌روزرسانی‌ها در دید ادغام می‌شود. در واقع اگر هیچ دوره‌ای سیستم ساکت نباشد دید ذخیره شده به‌روز نمی‌شود.

الگوریتم SWEEP یکی دیگر از روش‌های مبتنی بر جبران است که در سال ۱۹۹۶ توسط D. Agrawal معرفی شد [۲۱]. این روش، پرس‌وجوهای نگهداری را برای هر رابطه منبع می‌فرستد حتی اگر آن‌ها درپایگاه‌داده یکسان موجود باشند.

به عبارت دیگر وقتی از الگوریتم SWEEP استفاده می‌شود باید با هر رابطه منبع مانند یک منبع جدارفتار شود که این امر باعث افزایش تعداد پرس‌وجوی نگهداری به منابع و افزایش هزینه نهایی آن می‌شود.

روش DSCM [۲۲] مشکلات روش‌های نگهداری دید مبتنی بر جبران گذشته را بر طرف می‌کند. این روش رفت‌وبرگشت بین پایگاه‌داده‌تحلیلی و منابع داده را کاهش می‌دهد و اندازه‌ی پرس‌وجوهای جبران را کاهش می‌دهد و باعث کاهش سربار نگهداری دید افزایشی می‌شود.

روش دیگری که برای نگهداری افزایشی دید مطرح شده است استفاده از روش‌های گروه‌بندی است [۱۳][۱۶]. این روش‌ها عبارت نگهداری افزایشی دید را به‌صورتی گروه‌بندی می‌کنند که دفعات دسترسی به داده کاهش یابد و کمترین هزینه را برای نگهداری

[1]Cultural
[2]TLBO
[3]Genetics Algorithm(GA)
[4]Bee Algorithm
[5]Bacterial
[6]First In First Out

www.SID.ir

افزایشی دید ایجاد کنند. یکی از روش‌های گروه‌بندی روش دلتای بهینه است [۱۳]. این روش برای یافتن رابطه بهینه برای نگهداری دید از درخت دلتای بهینه استفاده می‌کند و عبارت نگهداری دیدی را پیدا می‌کند که کمترین هزینه را داشته باشد. این موضوع هزینه نگهداری دید افزایشی را کاهش می‌دهد [۱۳]. از معایب این روش این است که برای پیدا کردن رابطه بهینه زمان زیادی را صرف می‌کند. در این مقاله از الگوریتم‌های فراابتکاری استفاده شده است و با استفاده از آن، سرعت نگهداری افزایشی دید بالا رفته است. با بالا بردن سرعت نگهداری دید می‌توان در بازه زمانی مشخص دیدهای بیشتری را به‌روز کرد. هدف این است که در زمان کوتاه‌تر عبارت نگهداری دید با هزینه کمتری را به‌دست آورد.

۳- تعاریف اولیه

در این بخش ابتدا عبارت نگهداری افزایشی دید، معرفی می‌شود [۲۹]. سپس یک روش گروه‌بندی عبارت نگهداری دید مطرح می‌شود [۱۳]. این روش با کاهش تعداد دسترسی به برخی روابط تکراری، هزینه نگهداری دید را کاهش می‌دهد اما زمان زیادی را صرف می‌کند. سپس الگوریتم فرهنگی معرفی می‌شود [۲۸] و در انتها برای مقایسه‌ی الگوریتم فرهنگی با برخی از الگوریتم‌های فراابتکاری به معرفی الگوریتم‌های زنبور عسل، باکتریال و الگوریتم مبتنی بر تدریس و یادگیری پرداخته می‌شود. در بخش بعدی، با بهره‌گیری از روش گروه‌بندی عبارت نگهداری دید، الگوریتمی برمبنای الگوریتم فرهنگی، جهت افزایش سرعت نگهداری افزایشی دید معرفی می‌شود.

عبارت نگهداری دید: در روش نگهداری افزایشی دید سیستم مجموعه‌ای از تغییرات را مدیریت می‌کند. تغییرات منابع داده در پایگاه‌داده‌تحلیلی به روش‌های مختلفی پردازش می‌شوند. فرض بر این است که دید ذخیره شده روی (SPJ[1]) تعریف شده است [۷][۱۳][۳۰].

برای دید $V=R1\infty R2\infty \ldots\ldots \infty Rn$ عبارت نگهداری ۱ برای محاسبه تغییرات دید استفاده می‌شود [۲۹]. دید فوق با V مشخص شده است و n منبع داده موجود است که با Ri نمایش داده شده‌اند و از پیوستن آن‌ها، دید مورد نظر ایجاد می‌شود.

$$\Delta v = (\Delta R1 \infty R2 \infty \ldots \infty Rn) \cup (\Delta R1 \infty \Delta R2 \infty \ldots \infty Rn) \cup \ldots \ldots \ldots (\Delta R1 \infty \Delta R2 \infty \ldots \infty \Delta Rn) \quad (۱)$$

در عبارت ۱، منظور از Δ، تغییرات است و U نیز بیانگر اجتماع نتایج است. از آن‌جایی که عبارت ۱ نیازمند $(2^n\text{-}1)$ جمله برای تغییرات دید است، یک عبارت نگهداری جدید بررسی می‌شود که فقط شامل n جمله است [۳۱].

$$\Delta v = (\Delta R1 \infty R2 \infty \ldots \infty Rn) \cup (R1' \infty \Delta R2 \infty \ldots \infty Rn) \cup \ldots (R1' \infty R2' \infty \ldots . \infty \Delta Rn) \quad (۲)$$

برای مثال تغییرات دید برای ۳ رابطه $V=R1\infty R2.\infty R$ هفت جمله دارد.

$$\Delta v = (\Delta R1 \infty R2 \infty R3) \cup (R1 \infty \Delta R2 \infty R3) \cup (R1 \infty R2 \infty \Delta R3) \cup (\Delta R1 \infty \Delta R2 \infty R3) \cup (\Delta R1 \infty R2 \infty \Delta R3) \cup (R1 \infty \Delta R2 \infty \Delta R3) \cup (\Delta R1 \infty \Delta R2 \infty \Delta R3) \quad (۳)$$

اگر از رابطه ۲ استفاده شود تغییرات دید شامل ۳ جمله است.

$$\Delta v = (\Delta R1 \infty R2 \infty R3) \cup (R1' \infty \Delta R2 \infty R3) \cup (R1' \infty R2' \infty \Delta R3) \quad (۴)$$

در عبارت ۱، $(2^n\text{-}1)$ جمله وجود دارد و هر رابطه اصلی دقیقا شامل $(2^{n-1}\text{-}1)$ جمله است. از این‌رو هر کدام از روابط اصلی باید حداقل $(2^{n-1}\text{-}1)$ بار در دسترس قرار گیرند در صورتی‌که عبارت ۲ شامل n جمله است و هر کدام از روابط اصلی فقط (n-1) بار در دسترس قرار می‌گیرند. بدیهی است تعداد زیاد دفعات دسترسی به اطلاعات، هزینه کل رابطه را افزایش می‌دهد [۱۳، ۳۲]. پس باید تا حد امکان تعداد دفعات دسترسی را کاهش داد تا متعاقب آن میزان هزینه کاهش پیدا کند. بنابراین عبارت ۲ به‌عنوان عبارت نگهداری افزایشی دید در نظر گرفته می‌شود [۳۱].

$$V = R_1 \bowtie R_2 \bowtie \cdots \bowtie R_n$$
$$\Delta V = (\Delta R_1 \bowtie R_2 \bowtie \cdots \bowtie R_n) \cup (R_1' \bowtie \Delta R_2 \bowtie \cdots \bowtie R_n) \cup \ldots (R_1' \bowtie R_2' \bowtie \cdots \bowtie \Delta R_n) \quad (۵)$$

که در آن R_i Δها تغییرات مرتبط با هر R_i است و R_i'ها تغییر یافته هر R_i می‌باشد.

هزینه یک رابطه، شامل برآیند هزینه‌های هر دو مورد می‌باشد. در این رابطه هزینه نهایی با استفاده از رابطه ۶ محاسبه می‌شود:

$$Cost_{total} = Cost_{Calc_Time} \times Cost_{run} \quad (۶)$$

که در آن $Cost_{run}$ هزینه اجرای رابطه و $Cost_{Calc_Time}$ مدت زمانی است که طول می‌کشد تا کم هزینه‌ترین رابطه پیدا شود.

روش ارزیابی دلتای بهینه: برای نگهداری افزایشی دید عبارت‌های نگهداری دید متفاوتی وجود دارد [۳۳،۱۳]. یکی از عبارت‌های نگهداری افزایشی دید عبارت ۲ است. روش ارزیابی

[1] Select Project Join

www.SID.ir

دلتای بهینه با کاهش تعداد دسترسی به عبارت ۲ هزینه نگهداری افزایشی دید را کاهش می‌دهد. در این روش [۱۳] عبارت نگهداری دید به گروه‌هایی تقسیم می‌شود، این موضوع باعث کاهش هزینه اجرای یک رابطه می‌شود.

در ادامه، روش ارزیابی دلتای بهینه، با یک مثال معرفی می‌شود. فرض می‌شود که یک رابطه با n= ۳ داریم. برای آن ΔV به‌صورت رابطه ۷ محاسبه می‌شود:

$$\Delta V = (\Delta R_1 \bowtie R_2 \bowtie R_3) \cup (R'_1 \bowtie \Delta R_2 \bowtie R_3) \cup (R'_1 \bowtie R'_2 \bowtie \Delta R_3) \quad (۷)$$

هزینه عبارت ۷ به‌صورت عبارت ۸ محاسبه می‌شود،

$$\begin{aligned} Cost(\Delta V) &= Cost(\Delta R1 \infty R2 \infty R3) + \\ &Cost(R'1 \infty \Delta R2 \infty R3) + Cost(R'1 \infty R'2 \infty \Delta R3) \\ &= Cost(|\Delta R1| + |R2| + |R3|) + \\ &Cost(|R'1| + |\Delta R2| + |R3|) + Cost(|R'1| + |R'2| + |\Delta R3|) \end{aligned} \quad (۸)$$

حال عبارت ۹ که با عبارت ۷ مشابه است بررسی می‌شود [۱۲]:

$$\Delta v = \Big(\big((\Delta R1 \infty R2) \cup (R1' \infty \Delta R2)\big) \infty R3\Big) \bigcup (R1' \infty R2' \infty \Delta R3) \quad (۹)$$

در عبارت ۹ ابتدا عبارت $(\Delta R1 \infty R2) \cup (R1' \infty \Delta R2)$ ارزیابی می‌شود و سپس نتایج آن به R3 متصل می‌شود. از این‌رو رابطه اصلی R3 فقط یک‌بار در عبارت ۹ مورد دسترسی واقع شده است در حالی‌که رابطه اصلی R3 دوبار در رابطه ۷ دسترسی شده است. بدین ترتیب اگر اندازه R3 بسیار بزرگ‌تر از اندازه‌های R1 و R2 باشد می‌توان به‌طور قابل ملاحظه‌ای هزینه محاسبه تغییرات دید را کاهش داد.

تفاوت بین رابطه‌های ۷ و ۹ در میزان هزینه اجرای آن‌ها می‌باشد. در این روش [۱۳] عبارت نگهداری افزایشی دید به گروه‌هایی تقسیم می‌شود و این تقسیم‌بندی‌ها به‌صورت بازگشتی (تا جایی که در هر گروه یک رابطه اصلی وجود داشته باشد) ادامه داده می‌شود. حال باید از بین این گروه‌بندی‌ها، گروه‌بندی انتخاب شود که هزینه‌ی نگهداری دید کمتری دارد. این کار به وسیله درخت ارزیابی دلتا انجام می‌شود.

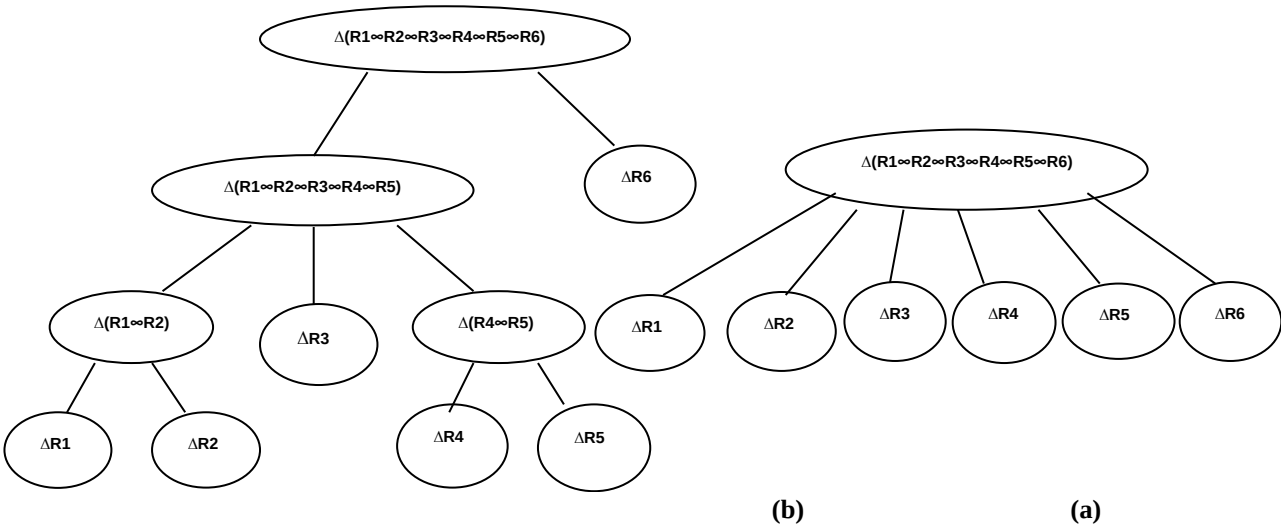
عبارت ارزیابی دلتا را می‌توان به‌صورت درخت نشان داد که به آن درخت ارزیابی دلتا گفته می‌شود. عبارت ارزیابی دلتا را برای دید $V= R1 \infty R2 \infty \dots \infty Rn$ در نظر بگیرید.

گره ریشه به‌صورت عبارت $\Delta(R1 \infty R2 \infty \dots \infty Rn)$ نشان داده می‌شود. تغییرات در هر کدام از روابط اصلی مثل $\Delta R1 \infty \Delta R2 \infty \dots \infty \Delta Rn$، تبدیل به گره‌های برگ درخت می‌شوند.

شکل ۱ دو درخت عبارت ارزیابی را برای دید $V = R1 \infty R2 \infty R3 \infty R4 \infty R5 \infty R6$ نشان می‌دهد.

این الگوریتم ابتدا درخت ارزیابی دلتای بهینه را برای رابطه اصلی می‌سازد و همان‌طور که مورد انتظار است، منحصر به فرد است. حال از بین درخت‌های ارزیابی دلتا به وسیله یک الگوریتم دینامیک درخت ارزیابی دلتای بهینه پیدا می‌شود.

سپس درخت ارزیابی دلتای بهینه در هر مرحله ایجاد می‌شود. درخت ارزیابی دلتای بهینه به این معنا است که عبارت ارزیابی دلتا مستلزم حداقل هزینه ارزیابی باشد. برای پیدا کردن درخت

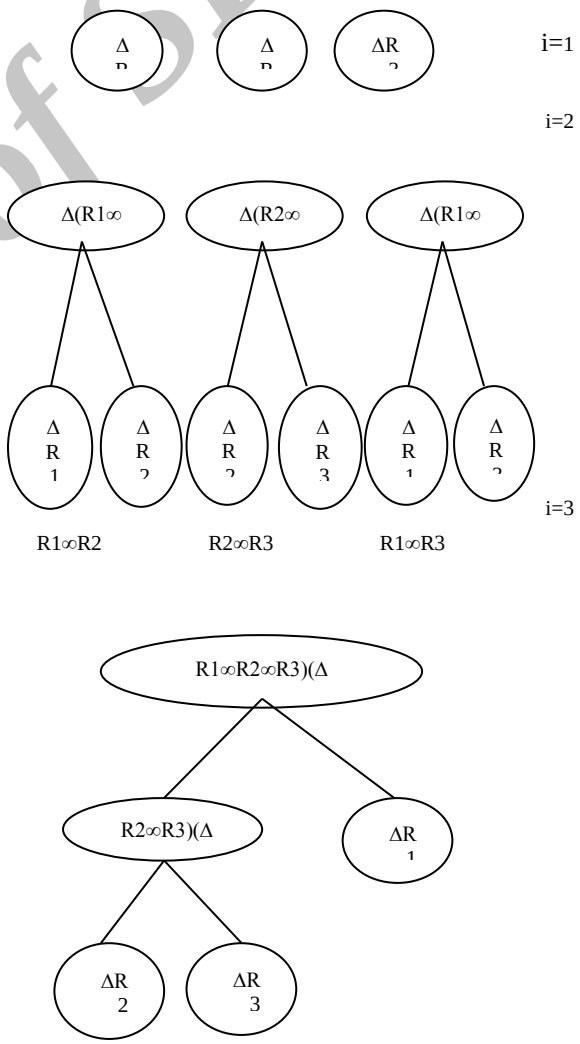


شکل ۱:درخت‌های ارزیابی دلتا[۱۲]

www.SID.ir

ارزیابی دلتای بهینه از یک الگوریتم پویا استفاده می‌شود. در اینجا از یک رویکرد پایین به بالا استفاده می‌شود. ابتدا درخت ارزیابی دلتای بهینه برای هر رابطه Ri ساخته می‌شود سپس با استفاده از نتیجه مرحله اول درخت ارزیابی دلتای بهینه برای $Ri\infty Rj(i \neq j)$ ساخته می‌شود.

در مرحله بعد با استفاده از نتیجه مرحله ۲ درخت ارزیابی دلتای بهینه برای $Ri\infty Rj\infty Rk(i \neq j \neq k)$ ساخته می‌شود شکل ۲ درخت ارزیابی دلتا را نمایش می‌دهد [۱۳]. سیستم‌های مبتنی بر این روش، ابتدا باید زمانی را صرف پیدا کردن رابطه‌ای کنند که کمترین میزان هزینه اجرا را داشته باشد. این زمان صرف شده می‌تواند در مجموع باعث افزایش برآیند هزینه‌های سیستم شود. همان‌طور که گفته شد یکی از مزیت‌های این روش، کاهش تعداد دسترسی‌ها به ترم‌های یک رابطه است.



شکل ۲: درخت ارزیابی دلتای بهینه[۱۳]

این موضوع، هزینه اجرای یک رابطه را تا حد زیادی کاهش می‌دهد. ولی یکی از نقطه ضعف‌های این روش این است که برای پیدا کردن درخت بهینه زمان زیادی صرف می‌شود. با در نظر گرفتن معایب و مزایای روش دلتای بهینه در این مقاله روشی پیشنهاد شده است که نه تنها هزینه زمانی محاسبات را کاهش می‌دهد بلکه تا میزان زیادی هزینه اجرا را با انتخاب بهترین رابطه کاهش می‌دهد. این موضوع در مجموع باعث کاهش برآیند هزینه‌های یک رابطه می‌شود.

الگوریتم فرهنگی

الگوریتم فرهنگی برای نخستین بار توسط Durham, W در سال ۱۹۹۴ معرفی شد [۲۸] و بعدها گسترش پیدا کرد و در سال‌های اخیر در مسایل مختلف بهینه سازی مورد استفاده قرار گرفته است [۱۰، ۱۱][۳۴][۳۵، ۳۷]. الگوریتم فرهنگی به‌طور کامل از روی جوامع بشری الگوبرداری شده است. به‌نحوی که بسیاری از معیارهای انسانی در گزینش نسل‌ها و یا انسان‌ها را می‌توان در این الگوریتم مشاهده کرد. در الگوریتم فرهنگی در وهله اول نیاز به یک جمعیت است. جمعیت اولیه‌ای که کار الگوریتم فرهنگی با آن آغاز می‌شود و در فضای جمعیتی[1] قرار می‌گیرد در حقیقت مشابه انسان‌های بدوی هستند. زیرا انسان‌های اولیه دارای اعتقادات و باورهای کلی هستند و هنوز ایدئولوژی در آن‌ها شکل نگرفته است.

پس برای شروع کار، یک جمعیت اولیه در نظر گرفته می‌شود با یک فضای باور[2] اولیه که دقیق و جزیی نیست. بعد از چند نسل فضای باور جزیی‌تر و دقیق‌تر می‌شود. به‌عنوان مثال جمعیت یک کشور را در نظر می‌گیریم. در یک کشور باورها و اعتقادهای مختلفی وجود دارد و اعضای جامعه بر اساس آن‌ها دسته‌بندی می‌شوند. عده‌ای باورهای مذهبی دارند پس در دسته انسان‌های مذهبی قرار می‌گیرند. عده‌ای علم محورند، در نتیجه در دسته افراد علمی قرار می‌گیرند و غیره.

با توجه به این که کدام معیار و ارزش در مساله مهم است، هدف الگوریتم مشخص می‌شود. به‌عنوان مثال، ممکن است پاسخ مساله پیدا کردن کوچک‌ترین مقدار یک تابع باشد. در آن صورت کوچک‌ترین مقدار در بین مجموعه جواب‌های ممکن برای مساله معیار است.

فرض کنید فضای باورعلمی و ارزشی باشد. همواره انسان‌ها خود را با افراد برجسته و برتر مقایسه می‌کنند و سعی دارند به آن‌ها

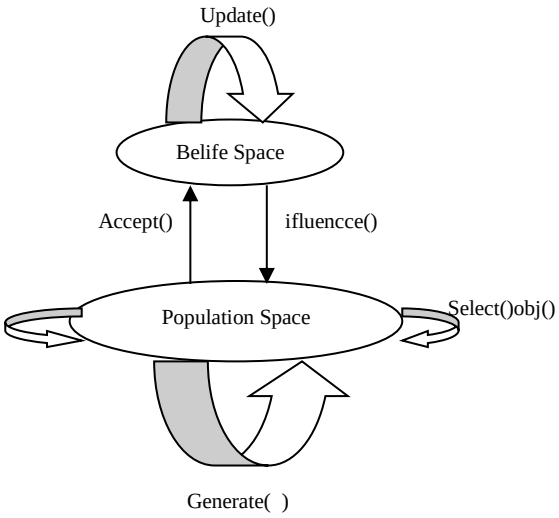
[1]Population Space
[2]Belief Space

www.SID.ir

برسند. در حقیقت آن‌ها یک معیار و یک فضای باور برای جمعیت می‌سازند. ممکن است این افراد نابغه با برخی از اعضای جمعیت ازدواج کنند که حاصل ممکن است فرزندی باشد که در آینده بر اساس معیارها در دسته افراد خاص قرار می‌گیرد یا جزء یکی از اعضای عادی جمعیت می‌شود. ولی مساله‌ای که در الگوریتم فرهنگی مورد تاکید است آن است که احتمال آن که فرزندان آن‌ها در نسل‌های آینده جزء فضای باور قرار نگیرد کم است. نسل‌ها یکی پس از دیگری به‌وجود می‌آیند و با معیارهای فضای باور مقایسه می‌شوند، اگر توانستند به آن سطح برسند با یکی از افراد فضای باور که از نظر ارزشی جایگاه کمتری دارد جابه‌جا می‌شوند در غیر این صورت باید باز هم برای رسیدن به آن تلاش کنند.

مساله جالبی که در الگوریتم فرهنگی وجود دارد، وجود عنصر خانواده و مرگ میر انسان‌ها است. در حقیقت در الگوریتم فرهنگی تعداد خانواده‌ها مشخص می‌شوند نه تعداد افراد. مثلا گفته می‌شود صد خانوار داریم که جزیی از جمیعت عمومی هستند و یک خانواده داریم که در فضای باور جا می‌گیرد. در شکل ۳ استراتژی مربوط به الگوریتم فرهنگی ارائه شده است. در ابتدا یک جمعیت اولیه به‌صورت تصادفی تولید می‌شود و در Population Space قرار می‌گیرد.

سپس هر یک از افراد جمعیت مورد ارزیابی قرار می‌گیرند. سپس از بهترین به بدترین چینش می‌شوند و تعدادی از آن‌ها توسط Accept Function انتخاب می‌شوند و در فضای باور قرار می‌گیرند. در مرحله بعدی فضای جمعیت از فضای باور توسط InfluencceFunction برای تولید جمعیت جدید استفاده می‌کند.



شکل ۳:استراتژی الگوریتم فرهنگی

الگوریتم باکتریال

امروزه بسیاری از مسایل بهینه سازی از الگوریتم باکتریال استفاده می‌کنند [۳۸]. اساس کار الگوریتم باکتریال بر این موضوع استوار است که جانداران ضعیف احتمال انقراض بیشتری نسبت به جانداران قدرتمند دارند. همین موضوع در الگوریتم ژنتیک هم لحاظ شده است. در الگوریتم ژنتیک می‌گوییم که والدینی که ارزش کمتری دارند ادامه نسل پیدا نمی‌کنند.

الگوریتم باکتریال بر اساس چهار مرحله پایه‌گذاری شده است.

حرکت

عملکرد گروهی

تولید مثل

حذف و پراکندگی

الگوریتم زنبور عسل

الگوریتم زنبور عسل [۳۹] نیز مانند تمام الگوریتم‌های فراابتکاری با یک جمعیت اولیه آغاز می‌شود. یک کلونی زنبور عسل می‌تواند در مسافت زیادی و نیز در جهت‌های گوناگون پخش شود تا از منابع غذایی بهره‌برداری کند. پروسه‌ی جستجوی غذا به وسیله زنبورهای دیده‌بان آغاز می‌شود که برای جستجوی گلزارهای امید بخش فرستاده می‌شوند. هنگامی که جستجوی تمام گلزارها پایان یافت، هر زنبور دیده‌بان، بالای گلزاری که اندوخته کیفی مطمئنی از نکتار و گرده دارد، رقص خاصی را اجرا می‌کند. این رقص که به نام رقص چرخشی شناخته می‌شود، اطلاعات مربوط به جهت تکه گلزار (نسبت به کندو)، فاصله تا گلزار و کیفیت گلزار را به زنبورهای دیگر انتقال می‌دهد. این اطلاعات زنبورهای اضافی و پیرو را به سوی گلزار می‌فرستد. وقتی همه زنبورها به سمت ناحیه‌ای مشابه بروند، دوباره به‌صورت تصادفی و به علت محدوده‌ی رقصشان در پیرامون گلزار پراکنده می‌شوند تا به موجب این کار سرانجام نه یک گلزار، بلکه بهترین گل‌های موجود درون آن تعیین موقعیت شوند.

الگوریتم مبتنی بر تدریس و یادگیری

الگوریتم مبتنی بر تدریس و یادگیری [۴۰] نیز مانند سایر الگوریتم های تکاملی یک الگوریتم برگرفته از طبیعت است و بر اساس تاثیر یک مدرس روی یادگیری کلاس کار می‌کند. جمعیت اولیه به عنوان

www.SID.ir

گروهی از دانش آموزان در نظر گرفته می‌شود. فرایند این الگوریتم به ۲ فاز تقسیم می‌شود. یک فاز مدرس که همان یادگیری از معلم است. معلم به عنوان شخصی با دانش بالا است و سعی می‌کند دانش دانش‌آموزان را بالا برده و به دانش خود نزدیک کند و دومی فاز یادگیرنده که دانش آموزان از تراکنش متقابل هم آموزش می‌بینند. یک دانش آموز می‌تواند چیزهای جدیدی را از سایر دانش آموزان یاد بگیرد به شرط این‌که سایرین دانش بالاتری نسبت به او داشته باشند.

۴- معرفی روش پیشنهادی (استفاده از الگوریتم فرهنگی به منظور افزایش سرعت نگهداری افزایشی دید)

برای دستیابی به یک روش کارآمد ابتدا باید تعداد دفعات رجوع به عبارت‌های مختلف یک رابطه، همانند روش دلتای بهینه تا حد امکان کاهش یابد. همان‌طور که پیش از این مطرح شد این کار باعث کاهش هزینه اجرای رابطه می‌شود [۱۳]. ولی یکی از مشکلات این روش، زمان زیاد برای پیدا کردن رابطه‌ای با هزینه اجرایی کم است. برای کاهش زمان جستجو و متعاقب آن کاهش هزینه زمانی، در این مقاله روش دلتای بهینه با الگوریتم فرهنگی ترکیب شده است.

در ابتدا لازم است تا جمعیت اولیه الگوریتم فرهنگی تولید شود. جمعیت اولیه الگوریتم فرهنگی شامل ترکیب‌های مختلف قابل ایجاد همانند روابط ۷ و ۹ هستند که به‌صورت تصادفی تولید شده‌اند. در مرحله بعد باید شایستگی هر یک از اعضای جمعیت مورد بررسی قرار بگیرد. سپس بهترین آن‌ها وارد فضای باور می‌شوند. در مورد مساله این مقاله، بهترین اعضای هر جمعیت، روابطی هستند که کمترین هزینه اجرایی را در پی داشته باشند. نسل بعدی الگوریتم فرهنگی با توجه به بهترین‌های فضای باور تولید می‌شوند و چرخه الگوریتم مجددا از نو آغاز می‌شود. پس از گذشت چند نسل، روابطی که در فضای باور قرار دارند کمترین میزان هزینه اجرایی را در بین جمعیت‌های تولید شده خواهند داشت. با توجه به شبیه‌سازی‌های انجام شده، مشاهده شد که حداکثر ۲۰۰ نسل برای این منظور کافی است.

با توجه به [۱۳] می‌توان نشان داد که تعداد روابطی که برای یک دید V با n رابطه می‌توان نوشت برابر است با $(۲^{n-۱} - ۱)$. در نتیجه تعداد رابطه‌هایی که در روش دلتای بهینه برای n=10 باید برای پیدا کردن رابطه‌ای با کمترین میزان هزینه بررسی شود برابر است با ۵۱۱ رابطه. از این نظر مدت زمانی که الگوریتم فرهنگی صرف یافتن رابطه‌ای با کمترین هزینه اجرایی می‌کند کمتر از روش دوم است. در شکل ۴ شبه کد الگوریتم پیشنهادی که حاصل ترکیب روش دلتای بهینه و الگوریتم فرهنگی است آورده شده است.

طبق شبه کد شکل ۴ ابتدا جمعیت اولیه الگوریتم فرهنگی مطابق با رابطه‌های نگهداری دید ایجاد می‌شود. سپس هر یک از رابطه‌های به‌دست آمده از نظر هزینه ارزیابی می‌شوند، رابطه بهینه انتخاب می‌شود و در فضای باور الگوریتم فرهنگی قرار می‌گیرد. برای انتخاب بهترین رابطه نگهداری دید، از تابع چیدن نزولی[1] استفاده می‌شود، سپس آرایه با اندیس صفر آن در فضای باور قرار می‌گیرد. این چرخه مدام تکرار می‌شود تا شرط پایان الگوریتم رخ دهد. برای پایان الگوریتم می‌توان چند شرط در نظر گرفت. یکی از شرط‌ها تولید پاسخ تکراری در چند چرخه پشت سر هم است. یکی دیگر از شرایط پایان الگوریتم می‌تواند فرارسیدن تعداد چرخه‌های از پیش تعیین شده باشد. در این مقاله از شرط پایان دوم استفاده شده است. طبق شبیه‌سازی‌های انجام شده تعداد چرخه‌ی الگوریتم پیشنهادی برابر با ۲۰۰ در نظر گرفته شده است.

```
Initial Culture _Pop_Base_Model2
Initial Belife_Space
While Condition_Algorithm
Output = culture(Culture-Pop_Base_Model2);
Output = SortMin(Output);
Belife_Space = Output(0);
Culture_Pop_Base_Model2=  Generate_Pop(Belife_Space);
End While
```

شکل ۴: شبه کد الگوریتم پیشنهادی

۵- نتایج و شبیه‌سازی

روش دلتای بهینه و الگوریتم پیشنهادی معرفی شده در این مقاله در شرایط یکسان با استفاده از C# با سیستمی با مشخصات Hard Disk ،RAM4GB ،Microsoft Windows 7 Ultimate ،750GB Intel(R) Core(TM) i5-4200U CPU 1.60GHz, 2301 Mhz, 2 Core(s), 4 Logical پیاده‌سازی شده و نتایج آن‌ها با هم مورد مقایسه قرار گرفته است. به این دلیل که در این مقاله از الگوریتم هوش مصنوعی برای بهبود روش استفاده شده است، روش پیشنهادی با الگوریتم ژنتیک، باکتریال [۱۲]، زنبور عسل و الگوریتم مبتنی بر تدریس و یادگیری نیز مقایسه شده است تا کاهش ایجاد شده در هزینه، قابل مشاهده باشد. الگورتیم ژنتیک

[1] SortMin

www.SID.ir

یک الگوریتم پرکاربرد در هر زمینه کاری است. آزمایشات نشان می‌دهد الگوریتم پیشنهادی، از الگوریتم ژنتیک بهتر عمل می‌کند.

نتایج حاصل از شبیه‌سازی به همراه پارامترهای آن در جدول ۱ ارائه شده است. این جدول دیدی را نشان می‌دهد که از ۶ رابطه تشکیل شده است. در این جدول وزن‌های هر یک از پارامترهای یک رابطه نگهداری دید، به‌عنوان مثال ارائه شده است. R، R'و ΔR هر یک اجزای یک رابطه نگهداری دید هستند. به‌عنوان مثال اندازه R_1 در ابتدا ۵۰ بوده و بعد از تغییراتی ($\Delta R1 = 10$)، اندازه‌اش به ۲۰ رسیده است ($R1' = 20$).

جدول ۱: جدول پارامترهای شبیه‌سازی

R1	R2	R3	R4	R5	R6
۵۰	۵۱	۷۸	۳۲	۱۵	۶۵
R1′	R2′	R3′	R4′	R5′	R6′
۲۰	۲۳	۵۱	۶۵	۹	۳۲
ΔR1	ΔR2	ΔR3	ΔR4	ΔR5	ΔR6
۱۰	۵۴	۲۱	۴	۸	۵۱

برای مثال هزینه نهایی یک رابطه برای $n=3$ را محاسبه می‌کنیم. $(۱ - ۲^{n-۱})$ عبارت می‌توان برای نگهداری دید افزایشی نوشت [۳۲]. برای دیدی که از ۳ رابطه اصلی تشکیل شده است ۳ عبارت نگهداری افزایشی دید وجود داردکه در زیر آمده‌اند. با توجه به جدول ۲، هزینه اجرای تغییرات دید به‌صورت رابطه ۱۰ محاسبه می‌شود.

(۱۰)

$$1)\quad \Delta V = (\Delta R_1 \bowtie R_2 \bowtie R_3) \cup (R_1' \bowtie \Delta R_2 \bowtie R_3) \cup (R_1' \bowtie R_2' \bowtie \Delta R_3)$$
$$Cost(\Delta V) = (10 + 51 + 78) + (20 + 54 + 78) + (20 + 23 + 21) = 355$$

(۱۱)

$$2)\quad \Delta V = (\Delta (R_1 \bowtie R_2) \bowtie R_3) \cup (R_1' \bowtie R_2' \bowtie \Delta R_3) = \Delta R_1 + R_2 + R_1' + \Delta R_2 + R_3 + R_1' \bowtie R_2' \bowtie \Delta R_3$$
$$Cost(\Delta V) = 10 + 51 + 20 + 54 + 78 + 20 + 23 + 21 = 277$$

(۱۲)

$$3)\quad \Delta V = (\Delta R_1 \bowtie R_2 \bowtie R_3) \cup (R_1' \bowtie \Delta (R_2 \bowtie R_3) == \Delta R_1 + R_2 + R_3 + R_1' + \Delta R_2 + R_3 + R_2' + \Delta R_3$$
$$Cost(\Delta V) = 10 + 51 + 87 + 20 + 54 + 78 + 23 + 21 = 315$$

با توجه به نتایج به‌دست آمده، رابطه ۱۱ کم هزینه‌ترین‌رابطه محسوب می‌شود. همان‌طور که مشاهده شد برای پیدا کردن کم هزینه‌ترین رابطه باید تمام روابط ممکن نوشته شود و در انتها هر کدام که کمترین مقدار را داشت انتخاب شود. برای پیدا کردن یک رابطه با کمترین مقدار ممکن است زمان زیادی صرف شود. همین موضوع هزینه نهایی را افزایش می‌دهد.

محاسبه هزینه برای یک رابطه در الگوریتم‌های پیشنهادی همانند روش دلتای بهینه است با این تفاوت که با توجه به شبیه‌سازی‌های انجام شده، یک رابطه با کمترین میزان هزینه در کمترین زمان ممکن محاسبه می‌شود.

پارامترهای مختلف در پیاده‌سازی الگوریتم فرهنگی در نظر گرفته شده است که در ادامه به توصیف آن‌ها می‌پردازیم.

تعداد جمعیت اولیه: در واقع فضای اصلی جمعیت می‌باشد و با مقداردهی اولیه کار خود را شروع می‌کند. جمعیت اولیه، درختان موجود برای هر رابطه نگهداری دید است. جمعیت اولیه ۱۰۰ در نظر گرفته شده است. در ادامه تاثیر این پارامتر بر روی سرعت نگهداری دید بررسی و مشاهده می‌شود.

تابع پذیرش: این تابع افراد شایسته در هر نسل را انتخاب کرده و وارد فضای باور می‌کند. هدف پیدا کردن کم‌ترین هزینه است. p_{accept} =۰/۰۱ در نظر گرفته شده است.

فضای باور: بهترین راه حل‌های ارایه شده در هر نسل در فضای باور ذخیره می‌شود و برای تولید نسل بعد در نظر گرفته می‌شود. در هر نسل هزینه تمام درختان محاسبه شده و کم‌هزینه‌ترین آن‌ها در فضای باور قرار می‌گیرد. در واقع در هر نسل یک درخت با کمترین هزینه در فضای باور وجود دارد که در هر نسل به‌روز می‌شود.

$S=\{MinSort\}$

تعداد نسل ها برای الگوریتم فرهنگی ۲۰۰ در نظر گرفته شده است. جدول شماره ۲ تاثیر پارامتر فضای جمعیت و تعداد نسل‌ها را بر روی سرعت نگهداری دید نشان می‌دهد.

تعداد جمعیت اولیه به‌ترتیب ۲۰، ۱۰۰و ۵۰۰ در نظر گرفته می‌شوند. با توجه به جدول ۱، در جدول ۲ هزینه نگهداری افزایشی دید الگوریتم فرهنگی نسبت به روش دلتای بهینه با تعداد جمعیت-های اولیه مختلف نشان داده شده است. درشکل ۵، سرعت نگهداری دید برای الگوریتم فرهنگی و روش ارزیابی دلتای بهینه آورده شده است.

Archive of SID

www.SID.ir

جدول ۲: مقایسه هزینه اجرای نگهداری افزایشی دید الگوریتم فرهنگی نسبت به روش ارزیابی دلتای بهینه با جمعیت اولیه متفاوت

Model (Old Method)	Culture Method(Population Space=20)	Culture Method(Population Space=100)	Culture Method(Population Space=500)
۸۲۸	۱۰۹۷	۹۰۸	۸۲۸

با توجه به جدول ۲و شکل ۵ مشاهده می شود وقتی جمعیت اولیه ۲۰ در نظر گرفته شود، سرعت نگهداری دید با استفاده از الگوریتم فرهنگی حدود ۷۰ در صد نسبت به روش ارزیابی دلتای بهینه افزایش داشته اما اندازه هزینه نیز حدود ۲۴ در صد افزایش می‌یابد. با توجه به فرمول شماره ۶ می دانیم که هزینه نهایی، برایند هزینه نگهداری افزایشی دید و سرعت نگهداری دید افزایشی است. در نتیجه برایند آنها حدود ۴۶ درصد باعث کاهش هزینه کلی می‌گردد.

با توجه به شکل ۵، اگر جمعیت اولیه ۵۰۰ در نظر گرفته شود، سرعت نگهداری افزایشی دید با استفاده از الگوریتم فرهنگی حدود ۴۵ درصد نسبت به روش ارزیابی دلتای بهینه افزایش می‌یابد.

در انتها پارامترهای الگوریتم فرهنگی را تغییر داده و جمعیت اولیه ۱۰۰ در نظرگرفته شده است. با توجه با نتایج آزمایشات نگهداری افزایشی دید با الگوریتم فرهنگی نسبت به روش ارزیابی دلتای بهینه در ۳۰ درصد مواقع حدود ۵۵ درصد و در ۷۰ درصد مواقع حدود ۶۵ درصد افزایش سرعت داشته است.

با در نظر گرفتن فرمول ۶ مشاهده می شودکه اندازه هزینه نگهداری افزایشی دید در الگوریتم فرهنگی حدود هشت در صد (۰/۰۸) افزایش داشته، در عوض سرعت نگهداری دید با استفاده از این الگوریتم بین ۵۵ تا ۶۵ در صد افزایش داشته است. برایند کاهش هزینه با استفاده از الگوریتم فرهنگی در ۶۵ در صد مواقع حدود ۵۸ درصد است. با توجه به نتایج آزمایشات مشاهده می‌شود که بیشترین برایند کاهش هزینه زمانی است که فضای جمعیت ۱۰۰ در نظر گرفته شده است.

روش پیشنهادی با برخی از الگوریتم‌های فراابتکاری مانند، الگوریتم ژنتیک، باکتریال، زنبور عسل و الگوریتم مبتنی بر تدریس و یادگیری نیز مقایسه می‌شوند.

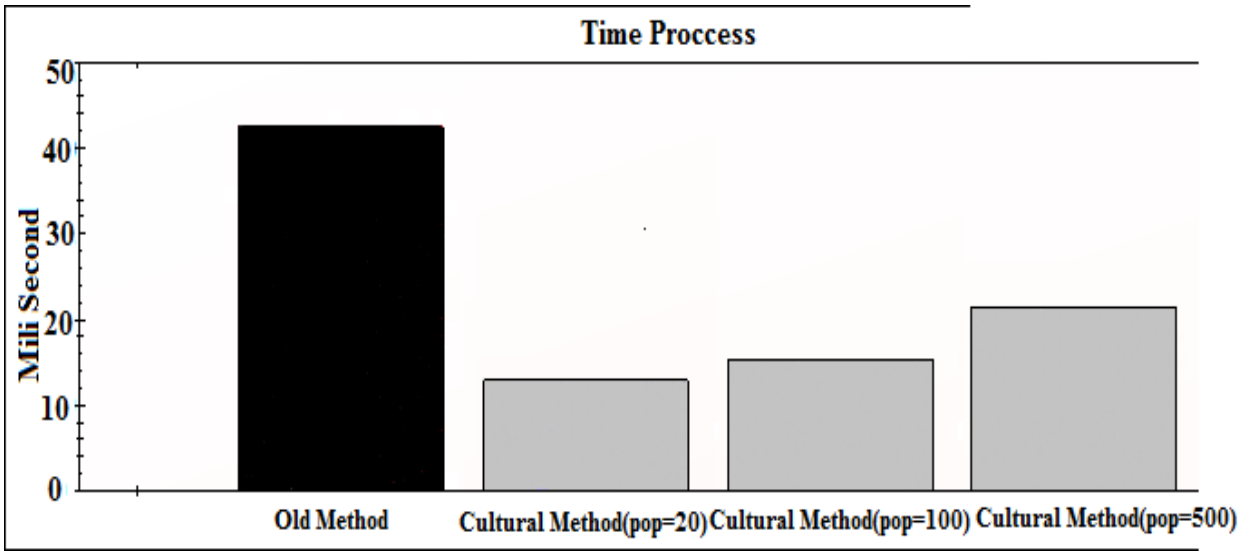
روش ارزیابی دلتای بهینه با استفاده از الگوریتم باکتریال

در [۱۲] از الگوریتم باکتریال برای بهبود هزینه نگهداری افزایشی دید استفاده شده است. الگوریتم باکتریال در بسیاری از مسایل بهینه‌سازی استفاده می‌شود [۳۸].

در ابتدا لازم است تا یک جمعیت اولیه تولید شود. جمعیت اولیه الگوریتم باکتریال شامل ترکیب‌های مختلف از عبارت‌های نگهداری دید است که به‌صورت تصادفی تولید شده اند. به هر یک از اعضای این مجموعه یک باکتری گفته می‌شود. این باکتری‌ها چند نسل را می‌گذرانند و در هر نسل چندین مرحله را طی می‌کنند و در هر مرحله تغییرمی‌کنند تا به درخت بهینه تبدیل شوند. در هر مرحله بهترین عبارت نگهداری دید انتخاب می‌شود و با مرحله قبل مقایسه می‌شود تا در نهایت چند نسل طی شود و بهترین و کم هزینه ترین عبارت نگهداری دید انتخاب شود.

روش ارزیابی دلتای بهینه با استفاده از الگوریتم زنبور عسل

الگوریتم زنبور عسل [۳۹] نیز مانند تمام الگوریتم‌های فراابتکاری با یک جمعیت اولیه آغاز می‌شود. جمعیت اولیه شامل ترکیب‌های



شکل ۵: مقایسه سرعت نگهداری افزایشی دید الگوریتم فرهنگی نسبت به روش ارزیابی دلتای بهینه با تعداد جمعیت اولیه متفاوت

www.SID.ir

مختلف قابل ایجاد برای یک عبارت نگهداری دید است که به صورت درخت ارزیابی دلتا نمایش داده می‌شود. جمعیت اولیه همان زنبورهای دیده‌بان هستند. تمام زنبورهای دیده‌بان گلزارهای اطراف را جستجو کرده و به دنبال منطقه‌ای با بیشترین گرده می‌گردند. اگر منطقه‌ای که یافته شد گرده بیشتری نسبت به منطقه قبل داشت منطقه قبل را فراموش می‌کنند و منطقه جدید را به‌خاطر می‌سپارند در غیر این‌صورت همان منطقه قبلی را به‌خاطر می‌سپارند. در انتها بقیه زنبورها به سمت بهترین منطقه حرکت می‌کنند. در واقع هر کدام از درخت‌ها درختان اطراف خود را بررسی می‌کند و به دنبال بهترین درخت می‌گردند. هر درخت از میان درختان اطرافش درختی با کمترین هزینه را انتخاب کرده و به خاطر می‌سپارد. از میان درختان جمعیت اولیه، درختی با کمترین هزینه انتخاب شده و بقیه درختان سعی می‌کنند به این درخت نزدیک شوند. چندین نسل می‌گذرد و در هر نسل هر درخت عبارت ارزیابی دلتا سعی می‌کند درخت بهینه را به خاطر بسپارد و بقیه درخت ها نیز به سمت درخت بهینه حرکت می‌کنند، تا در نهایت بهترین منطقه یعنی درختی با کمترین هزینه پیدا می‌شود.

روش ارزیابی دلتای بهینه با استفاده از الگوریتم مبتنی بر تدریس و یادگیری

در فاز معلم الگوریتم مبتنی بر تدریس و یادگیری، جمعیت اولیه که درختان موجود برای یک عبارت نگهداری دید هستند تولید شده و اندازه هزینه تمام درختان محاسبه می‌شود. بهترین و کم هزینه‌ترین درخت انتخاب می‌شود و به عنوان معلم در نظر گرفته می‌شود. حال تمام درختان بر اساس بهترین درخت تغییر داده می‌شوند و سعی می‌کنند به بهترین درخت نزدیک شوند. اگر درختان جدید از درختان قبلی هزینه کم‌تری داشتند پذیرفته می‌شوند در غیر این‌صورت درخت قبلی باقی می‌ماند. در فاز یادگیری، درختان از هم تاثیر می‌پذیرند. در اینجا فرض می‌شود هر درخت به‌صورت تصادفی با یک درخت دیگر در ارتباط است. از جمعیت تولید شده ۲ درخت انتخاب می‌شود و سعی می‌کنند به درخت با کمترین هزینه نزدیک شوند. اگر درخت تولید شده از درختان انتخابی بهتر بود پذیرفته شده و در غیر این‌صورت درختان انتخابی باقی می‌مانند.

با توجه به اندازه‌های جدول ۱ در جدول شماره ۳ هزینه نگهداری دید الگوریتم‌ها و در شکل شماره ۶ سرعت نگهداری دید الگوریتم‌ها با یکدیگر مقایسه شده است.

جدول ۳: هزینه اجرای روش دلتای بهینه، ژنتیک، باکتریال، الگوریتم مبتنی بر تدریس و یادگیری، الگوریتم زنبور عسل و الگوریتم فرهنگی

Model (old Method)	GA Method	Bacterial Method	TLBO Method	Bee Method	culture Method
۸۲۸	۹۰۸	۸۲۸	۹۱۴	۹۰۸	۹۰۸

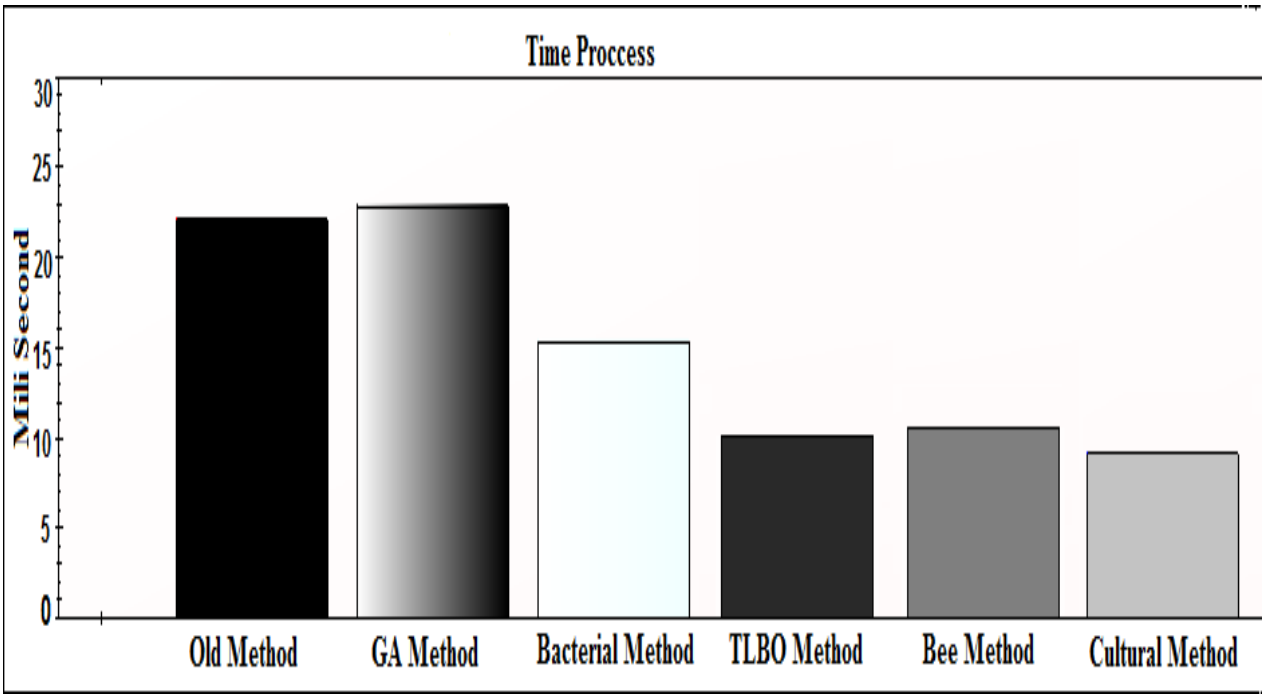
با توجه به فرمول شماره ۶ کاهش هزینه نهایی برایند کاهش هزینه و افزایش سرعت است. سعی بر این است که عبارت نگهداری دید با کمترین هزینه در زمان کوتاه تری به‌دست آید. با توجه به شکل شماره ۶ مشاهده می‌شود که الگوریتم باکتریال حدود ۳۰ درصد افزایش سرعت نسبت به روش ارزیابی دلتای بهینه دارد. الگوریتم مبتنی بر تدریس و یادگیری حدود ۵۵ درصد افزایش سرعت دارد و در عوض حدود ۹ درصد افزایش هزینه دارد، بنابراین برایند کاهش هزینه حدود ۴۶ درصد است. الگوریتم زنبور عسل حدود ۵۳ درصدافزایش سرعت و ۸ درصد افزایش هزینه دارد. برایند کاهش هزینه با استفاده از این الگوریتم حدود ۴۵ درصد است. الگوریتم فرهنگی حدود ۶۰ درصد افزایش سرعت و ۸ در صد افزایش هزینه نسبت به روش ارزیابی دلتای بهینه دارد.

برایند کاهش هزینه با استفاده از الگوریتم فرهنگی حدود ۵۲ درصد است. همان‌طور که مشاهده می‌شود الگوریتم فرهنگی بالاترین سرعت را نسبت به بقیه الگوریتم‌ها دارد.

۶- نتیجه‌گیری نهایی

پایگاه‌داده‌تحلیلی، منبع داده‌ای است که برای بهبود سرعت پاسخ‌گویی به پرس‌وجوهای تحلیلی، از منابع داده پایه ایجاد می‌شود. به این منظور، پاسخ به پرس‌وجوها در این منبع داده، ذخیره می‌شود. به این پاسخ‌های ذخیره شده، دید ذخیره شده گفته می‌شود. با توجه به تغییر داده‌های اصلی در منابع داده پایه، این دیدها نیز باید به‌روزرسانی شوند. روش‌های نگهداری دید ذخیره شده به این دلیل مطرح شده‌اند. روش دلتای بهینه، یکی از روش‌های نگهداری افزایشی دید است. این روش عبارت نگهداری دیدی را پیدا می‌کند که کمترین هزینه را دارد. اما زمان زیادی را به‌منظور جستجوی رابطه بهینه صرف می‌کند. به این دلیل، در این مقاله، برای افزایش سرعت نگهداری دید، الگوریتم فرهنگی پیشنهاد شد.این الگوریتم از یک جمعیت اولیه استفاده می‌کند و شامل مجموعه‌ای از روابط برای نگهداری افزایشی دید است. در هر

www.SID.ir



شکل ۶: مقایسه سرعت الگوریتم‌ها از سمت چپ روش دلتای بهینه، ژنتیک، الگوریتم باکتریال، الگوریتم مبتنی بر تدریس و یادگیری، الگوریتم زنبور عسل و الگوریتم فرهنگی

نسل بهترین عبارت انتخاب می‌شود و درون فضای باور قرار می‌گیرد و در نهایت بهترین و کم هزینه‌ترین عبارت تولید می‌شود.

با توجه به هزینه نهایی به دست آمده در استفاده از الگوریتم فرهنگی و مقایسه این الگوریتم با الگوریتم‌های زنبور عسل، باکتریال و الگوریتم مبتنی بر تدریس و یادگیری و همچنین با استناد به نمودارها می‌توان نتیجه گرفت که الگوریتم فرهنگی در این مقاله دارای برآیند هزینه نهایی کمتری نسبت به سایر الگوریتم‌ها است. در واقع حدود ۵۰ درصد کاهش هزینه مشاهده شد. الگوریتم پیشنهادی از نظر میزان زمان انجام محاسبات، از الگوریتم باکتریال نیز سریع‌تر است. در شکل ۶، سرعت اجرای الگوریتم‌ها در مقایسه با یکدیگر ملاحظه می‌شود. همان‌طور که مشاهده می‌شود الگوریتم ژنتیک بیشترین زمان اجرا را در بین الگوریتم‌های دیگر دارد و الگوریتم پیشنهادی (فرهنگی)، کمترین زمان اجرا را در میان سایر الگوریتم‌ها دارد.

سپاسگزاری

این پژوهش با حمایت مالی دانشگاه تربیت دبیر شهید رجایی طبق قرارداد شماره ۳۳۴۷۴ مورخ ۱۳۹۴/۱۲/۱۵ انجام گردیده است.

مراجع

[1] J. Zhou, P. Larso, G. Elmongui,"Lazy Maintenance of Materialized Views",33th International conference on Very Large data bases, Vienna, pp. 231-242, 2007.

[2] G. Luo, J.F. Naughton, C. Ellmann,M.Watzke, "A comparison of three methods for join view maintenance in parallel RDBMS", 19th International Conference on Data Engineering, pp. 177-188, 2003.

[3] X. Zhang, L. Ding, A. Rundensteiner,"Parallel multisource viewmaintenance", the VLDB Journal, 13(1): 22-48, January 2004.

[4] M. Koch, "An Applied Data Matching Methodology", Master's Thesis, University of Kaiserslautern, December 2010.

[5] T. Palpanas, R. Sidle, R. Cochrane, H. Pirahesh, "Incremental maintenance for non-distributive aggregate functions", 28th international conference on Very Large Data Bases, pp. 802-813, 2002.

[6] S. Chen, E.A. Rundensteiner, "GPIVOT: efficient incremental maintenance of complex ROLAP views", 21th international Conference on Data Engineering, pp. 552-563, 2005.

[7] Y. Zhuge, H. Garcia, J. Wiener,"Consistency algorithms for multi-source warehouse view maintenance", Journal of Distributed and Parallel Databases,6(1): 7-40, 1998.

[8] A. Behrend, T.Jörg ,"Optimized Incremental ETL Jobs for Maintaining Data Warehouses", IDEAS10, pp. 216-224,2010.

[9] H. He, , J. Xie J. Yang, H. ,"Asymmetric Batch Incremental View Maintenance",21st international conference on Data Engineering, pp. 106-117, 2005.

[10] S. Azarkasb, "Optimization Of Cultural Algorithms Structure Based On Pareto", Master's Thesis, Qazvin Islamic Azad University, Iran, 2012.

[11] S. Srinivasan, S. Ramakrishnan, "Cultural Algorithm Toolkit for Multi-objective Rule Miniing", International Journal on Computational Sciences & Applications (IJCSA) 2(4):9-23, August 2012.

[۱۲] ع. کریمی مصدق، ن. دانشپور، "کاهش هزینه نگهداری افزایشی دید پایگاه‌داده‌تحلیلی با استفاده از الگوریتم‌های فراابتکاری"، سومین کنفرانس بین المللی اطلاعات،حال و آینده۲۰۱۴.

[13] K. Yong Lee, J. Hyun Son, M. Kim, "Reducing the cost of accessing relations in incremental view maintenance", Decision Support Systems, 43(2): 512-526,2007.

[14] P.Ghosh, S. Se,"Dynamic incrementalmaintenance of materialized view based on attribute affinity",International Conference on Data Science &Engineering, pp. 12-17, 2014.

[15] N.Folkert, A.Gupta, A.Witkowski,S.Subrmanian, S. Bellamkonda, S. Shankar, T. Bozgaya, L.Sheng, "OptimisingRefresh of a Set of

Archive of SID

www.SID.ir

Materialized View", Proceedings of the 31stVLDB Conference, Trondheim, Norway, pp. 1043-1054, 2005.
[16] B. Liu, A. Rundensteiner, D. Finkel."Maintaining large update batches by restructuring and grouping", Information Systems, 32(4): 621–639, 2007.
[17] L.zhou,Q.Shi, H.Geng, "The minimum Incremental Maintenance of Materialized Views in Data Warehouse", 2nd International Asia Conference on Informatics in Control, pp. 220-223, 2010.
[18] G. C. H. Yeung,W. Gruver,"Multiagent Immediate Incremental View Maintenance for Data Warehouses",IEEE Transactions On Systems Man And Cybernetics—Part A Systems And Humans, 35(2): 305-310, 2005.
[19] Y. Zhuge, H. Garcia-Molina, J. Hammer, J. Widom,"View maintenance in a warehousing environment", SIGMOD' 95, pp. 316-327, 1995.
[20] Y. Zhuge, H. Garcia-Molina, J. Hammer, J. L. Wiener,"The strobe algorithms for multi-source warehouse consistency", Fourth International Conference on Parallel and Distributed Information Systems, pp. 146- 157, 1996.
[21] D. Agrawal, A. E. Abbadi, A Singh, T. Yurek,"Efficient view maintenance at data warehouses",SIGMOD, pp. 417-427, 1997.
[22] X.Zhang, L.Yang, D.Wang,"Incremental View Maintenance Based on Data Source Compensation in Data Warehouses", International Conference on Computer Application and System Modeling,pp. 287-291, 2010.
[23] B. Qin, S. Wang, X. Du,"EffectiveMaintenance of Materialized Views in Peer Data Management Systems", First International Conference on Semantics, Knowledge, and Grid,pp. 18, 2005.
[24] J. Zhou, P. Larson H. G. Elmongui,"Lazy Maintenance of Materialized Views", 33rd International conference on Very Large data bases, Vienna, Austria, pp. 231-242, 2007.
[25] Y. Zhuge, H. Garcia-Molina, J. Wiener,"Consistency algorithms for multi-source warehouse view maintenance", Journal of Distributed and Parallel Databases, pp. 7–40,2007.
[26] A. Behrend, T. Jörg,"Optimized Incremental ETL Jobs for Maintaining Data Warehouses", IDEAS, pp. 216-224, 2010.
[27] A. S.almazyad, m.K. Siddiqui,"Incremental View Maintenance: An Algorithmic Approach", International Journal of Electrical & Computer Sciences, 10(3): 16-21, 2014.
[28] X. Jin, H. Liao, "An incremental maintenance method for XQuery materialized view", Mechatronic Science, Electric Engineering and Computer (MEC), Jilin, pp. 797-801, 2011.
[29] T. Griffin, L. Libkin,"Incremental maintenance of views with duplicates", ACM SIGMOD Conference, pp. 328–339, 1995.
[30] K. Yi, H. Yu, J. Yang, G. Xia, Y. Chen, "Efficient maintenance of materialized top-k views", ICDE Conference, pp. 189–200, 2003.
[31] G. Luo, J.F. Naughton, C. Ellmann, M.Watzke, "A comparison of three methods for join view maintenance in parallel RDBMS", ICDE Conference, pp. 177–188, 2003.
[32] T. Jörg and S. Dessloch,"View Maintenance using Partial Deltas", BTW, LNI P – 180, Kaiserslautern, Germany, pp. 287-306, 2011.
[33] B. Qin, S. Wang, X. Du,"EffectiveMaintenance of Materialized Views in Peer Data Management Systems", First International Conference on Semantics, Knowledge, and Grid,pp:18, 2005.
[34] W. H.Durham, "Genes, Culture, andHuman Diversity", Stanford UniversityPress, Stanford, California, 1994.
[35] B.Franklin, M. Bergerman,"Cultural algorithms:concepts and experiments", Evolutionary Computation, 2: 1245–1251, 2000.
[36] S. Ahmadi,N.Forouzideh,H.Chung,R.Martin, "First Study of Fuzzy Cognitive Maps learning usingculturalalgorithm",9th ConferenceonIndustrial Electronics and Applications, pp. 2023-2028,2014.
[37] X. Benxian,C. Rongbao,L. Cheng,L. Yanhong,"Superheated steam temperature control based-on Cultural based Ant Colony optimization Algorithm for power station boiler", 32nd Chinese ControlConference(CCC), pp. 7965-7970, 2013.
[38] A. Buruzs, M. F. Hatwagner, R. C. Pozna, L. T. K6czy,"Advanced Learning of Fuzzy Cognitive Maps of Waste Management by Bacterial Algorithm", IFSA World Congress and NAFIPS Annual Meeting, pp. 890-895, 2013.
[39] Y. Yujang, H.Renjie, "A Novel Artificial Bee Colony Algorithm",Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 1:271-274, 2014.
[40] Z. Yang,K. Li, A. Foley,C. Zhang,"A new self-learning TLBOalgorithm for RBF neural modelling of batteries in electric vehicles",IEEE Congress on Evolutionary Computation (CEC), pp. 2685-2691, 2014.

Archive of SID

www.SID.ir

Improving the speed of view maintenance in data warehouses

Afifeh Karimi Mosadegh[1], Negin Daneshpour[2*]

1- Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin,Iran.
2*- Corresponding Author: Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran.
[1]karimi.mosadegh@yahoo.com. [2*] ndaneshpour@srttu.edu

Abstract- In recent years, due to the use of data warehouses, the subject of materialized view maintanance is important. For quick access to data, data warehouses collect the necessary data from various sources and stored them to form of materialized views. This leads to increased speed of responding to queries. When the data changes over different resources, the materialized views should be updated that leads to the subject of view maintanence. At this time, algorithms are presented in order to view maintanance with optimized cost. The algorithm presented in this paper is the combining of a mathematical method with Cultural meta-heuristic algorithm that leads to reduce search time and optimizes the cost of access to data in data warehouses. Cultural algorithm uses a reasonable belief space, including several incremental view maintenance relations. The best response obtained at the end of every generation is stored in the space called the belief space. The tests showe that Cultural algorithm is faster to maintain incremental views compared to previous methods and algorithms like bacterial and bees and learning tlbo algorithm.

Keywords- Data Warehouse, Cultural algorithm, Optimal delta tree, Incremental view maintenance.

Archive of SID

www.SID.ir