

بهبود سرعت نگهداری دید در پایگاه داده تحلیلی

عفیفه کریمی مصدق^۱، نگین دانشپور^{۲*}

۱- کارشناسی ارشد، گروه کامپیوتر، دانشگاه آزاد 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 سال های اخیر به دلیل استفاده از پایگاه داده تحلیلی موضوع مربوط به حفظ و نگهداری دید ذخیره شده مورد اهمیت است. برای دسترسی سریع به داده ها، پایگاه های داده تحلیلی، داده های لازم را از منابع مختلف جمع آوری کرده و به فرم دید ذخیره شده در خود ذخیره می کنند. این امر باعث سرعت بالای پایگاه داده تحلیلی در پاسخ به پرس و جوها می شود. وقتی داده ها در منابع مختلف تغییر می کنند، دیدهای ذخیره شده نیز باید به روز شوند که این امر منجر به مطرح شدن موضوع نگهداری دید می شود. در این زمان ارائه الگوریتم هایی به منظور نگهداری دید با هزینه بهینه، مطرح می شود. الگوریتمی که در این مقاله ارائه می شود، ترکیب یک روش ریاضی با الگوریتم فرابتنکاری فرهنگی است که باعث کاهش زمان جستجو و بهینه شدن هزینه دسترسی به داده ها در پایگاه داده تحلیلی است. الگوریتم فرهنگی از یک فضای باور مناسب استفاده می کند که شامل چندین رابطه نگهداری افزایشی دید است. هر بار که بهترین پاسخ در انتهای هر نسل به دست آید، در فضایی به اسم فضای باور ذخیره می شود. آزمایش ها نشان می دهند که الگوریتم فرهنگی در مقایسه با روش های قبلی و الگوریتم های باکتریال و زنبور عسل و الگوریتم مبتنی بر تدریس و یادگیری از سرعت بالاتری به منظور به روزرسانی دید افزایشی برخوردار است.

واژه های کلیدی: پایگاه داده تحلیلی، الگوریتم فرهنگی، درخت دلتای بهینه، نگهداری افزایشی دید.

۱- مقدمه

پایگاه داده تحلیلی^۱، معمولاً از پایگاه داده عملیاتی و یا سایر منابع داده ای توزیع شده سازمان ها و ارگان های متفاوت ایجاد می شود. پایگاه داده تحلیلی شامل داده هایی است که برای انجام تصمیم گیری ها و تحلیل ها مناسب است. پایگاه داده تحلیلی بستر مناسبی فراهم می آورد که داده ها به منظور پاسخ گویی به پرس و جوی های تحلیلی به صورت بایگانی شده، سر جمع شده و سازمان یافته، به فرم دید ذخیره شده^۲ ذخیره شوند. در محیط های پایگاه داده تحلیلی مفهوم دید ذخیره شده اهمیت بسیاری دارد. بابه کارگیری دید ذخیره شده درخواست ها می توانند به جای این که به چندین منبع دسترسی داشته باشند به طور مستقیم به دیدهای ذخیره شده دسترسی داشته باشند، که این باعث دسترسی سریع به داده ها می شود و سرعت را بالا می برد. وقتی منابع داده اصلی تغییر می کنند دیدهای ذخیره شده در پایگاه داده تحلیلی نیز باید به روز شوند. از آنجایی که افزایش

به روزرسانی های دید ممکن است سربار زیادی را تحمیل کند ضروری است که به روزرسانی دیدها با دقت بالایی انجام شود. به همین دلیل، موضوع نگهداری دید^۳ ذخیره شده مطرح می شود. در بسیاری از سیستم ها به روزرسانی دیدها در زمانی اتفاق می افتد که سیستم بیکار است مثلاً از ۱۲ شب تا ۶ صبح. پس اگر به روزرسانی دیدها با سرعت بالاتری انجام شود، می توان در زمان مشخص تعداد دیدهای بیشتری را به روز کرد.

دو روش برای نگهداری دید ذخیره شده وجود دارد: یکی از روش ها محاسبه مجدد دیدها است. این روش نیاز به ذخیره سازی های اضافی دارد و هزینه نگهداری آن بالا است، در بعضی مواقع هم به دلیل محدودیت های ذخیره سازی، غیر قابل پیاده سازی است. روش دیگر استفاده از تکنیک نگهداری افزایشی دید^۴ است که در سال های اخیر بیشتر مورد استفاده قرار می گیرد [۹-۱]. از آنجایی که اندازه تغییرات، کوچک تر از اندازه روابط اصلی و دیدها است، می توان نتیجه گرفت محاسبه تغییرات، بسیار ارزان تر

³View Maintenance

⁴Incremental View Maintenance

¹Data warehouse

²Materialized View

به صورت موازی انجام دهند و این باعث می شود که سرعت به روزرسانی ها افزایش یابند.

از دیگر روش های نگهداری دید، روش های مبتنی بر موازنه هستند [۱۹][۲۰][۲۱][۲۲]. این روش ها خطاهایی را که ممکن است در

به روزرسانی همزمان دیدها رخ دهند از بین می برند.

ECA و Strobe، [۱۹][۲۰] دو روش مبتنی بر جبران هستند. تعداد پرس وجوهایی که الگوریتم ECA برای نگهداری دید ارسال می کند از مرتبه دو است، و این تعداد برای الگوریتم Strobe از مرتبه فاکتوریل است. ECA پرس وجوهایی جبران را به منبع ارسال می کند و از این طریق تاثیر به روزرسانی های همزمان را که باعث خطا می شود برطرف می کند. این الگوریتم تنها برای یک منبع داده، عملکرد مناسبی دارد. الگوریتم Strobe [۲۰] مشکل روش قبل را برطرف کرده و روی چند منبع داده کاربرد دارد. این الگوریتم به روزرسانی ها را به ترتیبی که می رسند پردازش می کند. مشکل الگوریتم Strobe این است که اگر به روزرسانی های منابع ادامه داشته باشد و الگوریتم به یک حالت سکون دست پیدا نکند دید به روز نمی شود. این الگوریتم منتظر می ماند تا زمانی که به روزرسانی ها فروکش کنند و فقط آن زمان است که نتایج همه به روزرسانی ها در دید ادغام می شود. در واقع اگر هیچ دوره ای سیستم ساکت نباشد دید ذخیره شده به روز نمی شود.

الگوریتم SWEEP یکی دیگر از روش های مبتنی بر جبران است که در سال ۱۹۹۶ توسط D. Agrawal معرفی شد [۲۱]. این روش، پرس وجوهایی نگهداری را برای هر رابطه منبع می فرستد حتی اگر آن ها در پایگاه داده یکسان موجود باشند.

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

روش DSCM [۲۲] مشکلات روش های نگهداری دید مبتنی بر جبران گذشته را بر طرف می کند. این روش رفت و برگشت بین پایگاه داده تحلیلی و منابع داده را کاهش می دهد و اندازه ی پرس وجوهایی جبران را کاهش می دهد و باعث کاهش سربار نگهداری دید افزایشی می شود.

روش دیگری که برای نگهداری افزایشی دید مطرح شده است استفاده از روش های گروه بندی است [۱۳][۱۶]. این روش ها عبارت نگهداری افزایشی دید را به صورتی گروه بندی می کنند که دفعات دسترسی به داده کاهش یابد و کمترین هزینه را برای نگهداری

از انجام محاسبات مجدد است. مقدار داده افزایشی برای دیدهای یکسان که از متدهای مختلف نگهداری افزایشی استفاده می کنند متفاوت است و این امر منجر به هزینه های نگهداری دید متفاوت می شود.

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^۱ [۱۰، ۱۱] استفاده شده تا زمان یافتن یک رابطه ی بهینه برای نگهداری افزایشی دید کاهش داده شود. مشاهده می شود که الگوریتم پیشنهادی نسبت به الگوریتم های فراابتکاری مبتنی بر تدریس و یادگیری^۲ و الگوریتم ژنتیک^۳ و الگوریتم زنبور عسل^۴ و باکتریال^۵ 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 می گردد [۱۳]، سپس یک روش نگهداری دید افزایشی معرفی می گردد که روش دلتای بهینه نام دارد. این روش از طریق کاهش تعداد دسترسی به برخی روابط تکراری هزینه نگهداری دید را کاهش می دهد [۱۳]. از معایب این روش این است که، زمان زیادی را صرف پیدا کردن رابطه بهینه برای نگهداری دید افزایشی می کند. در نهایت در بخش چهارم روش پیشنهادی این مقاله مطرح می گردد. در بخش پنج نتایج شبیه سازی برای مقایسه و ارزیابی، آورده شده است.

۲- کارهای گذشته

ایده ها و الگوریتم های مختلفی برای نگهداری دید افزایشی وجود دارد [۲][۵][۷][۸][۱۳-۲۸] که هدف آن ها کاهش هزینه، یعنی کاهش محاسبات و افزایش سرعت نگهداری دید است.

یکی از ایده ها [۱۷] این است که دیدها را بر حسب اندازه تغییراتشان به صورت صعودی مرتب کرده و به صورت FIFO^۶ دیدها به روزرسانی شوند. در این صورت هزینه نگهداری کاهش می یابد.

روش دیگر نگهداری دید استفاده از عامل ها است [۱۸]. عامل ها می توانند به صورت موازی اجرا شوند و پردازش به روزرسانی ها را

¹ Cultural

² TLBO

³ Genetics Algorithm(GA)

⁴ Bee Algorithm

⁵ Bacterial

⁶ First In First Out

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

$$\Delta v = (\Delta R_1 \otimes R_2 \otimes \dots \otimes R_n) U (R_1' \otimes \Delta R_2 \otimes \dots \otimes R_n) U \dots (R_1' \otimes R_2' \otimes \dots \otimes \Delta R_n) \quad (2)$$

برای مثال تغییرات دید برای ۳ رابطه $V = R_1 \otimes R_2 \otimes R_3$ هفت جمله دارد.

$$\Delta v = (\Delta R_1 \otimes R_2 \otimes R_3) U (R_1 \otimes \Delta R_2 \otimes R_3) U (R_1 \otimes R_2 \otimes \Delta R_3) U (\Delta R_1 \otimes \Delta R_2 \otimes R_3) U (\Delta R_1 \otimes R_2 \otimes \Delta R_3) U (R_1 \otimes \Delta R_2 \otimes \Delta R_3) U (\Delta R_1 \otimes \Delta R_2 \otimes \Delta R_3) \quad (3)$$

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

$$\Delta v = (\Delta R_1 \otimes R_2 \otimes R_3) U (R_1' \otimes \Delta R_2 \otimes R_3) U (R_1' \otimes R_2' \otimes \Delta R_3) \quad (4)$$

در عبارت ۱، $(2^n - 1)$ جمله وجود دارد و هر رابطه اصلی دقیقاً شامل $(2^{n-1} - 1)$ جمله است. از این‌رو هر کدام از روابط اصلی باید حداقل $(2^{n-1} - 1)$ بار در دسترس قرار گیرند در صورتی که 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].

$$V = R_1 \otimes R_2 \otimes \dots \otimes R_n \\ \Delta V = (\Delta R_1 \otimes R_2 \otimes \dots \otimes R_n) U (R_1' \otimes \Delta R_2 \otimes \dots \otimes R_n) U \dots (R_1' \otimes R_2' \otimes \dots \otimes \Delta R_n) \quad (5)$$

که در آن ΔR_i ها 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:

$$Cost_{total} = Cost_{Calc_Time} \times Cost_{run} \quad (6)$$

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 [۱۳، ۳۳]. یکی از عبارت‌های نگهداری افزایشی دید عبارت ۲ است. روش ارزیابی

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 بر مبنای 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 پایگاه داده تحلیلی به روش‌های مختلفی پردازش می‌شوند. فرض بر این است که دید ذخیره شده روی (SPJ) تعریف شده است [۷][۱۳][۳۰].

برای دید $V = R_1 \otimes R_2 \otimes \dots \otimes R_n$ 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 R_i نمایش داده شده‌اند و از پیوستن آن‌ها، دید مورد نظر ایجاد می‌شود.

$$\Delta v = (\Delta R_1 \otimes R_2 \otimes \dots \otimes R_n) U (\Delta R_1 \otimes \Delta R_2 \otimes \dots \otimes R_n) U \dots \dots (\Delta R_1 \otimes \Delta R_2 \otimes \dots \otimes \Delta R_n) \quad (1)$$

¹Select Project Join

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 به صورت 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 هزینه نگهداری دید کمتری دارد. این کار به وسیله درخت ارزیابی دلتا انجام می‌شود.

عبارت ارزیابی دلتا را می‌توان به صورت درخت نشان داد که به آن درخت ارزیابی دلتا گفته می‌شود. عبارت ارزیابی دلتا را برای دید $V = R_1 \bowtie R_2 \bowtie \dots \bowtie R_n$ در نظر بگیرید.

گره ریشه به صورت عبارت $\Delta(R_1 \bowtie R_2 \bowtie \dots \bowtie R_n)$ نشان داده می‌شود. تغییرات در هر کدام از روابط اصلی مثل $\Delta R_1 \bowtie \Delta R_2 \bowtie \dots \bowtie \Delta R_n$ تبدیل به گره‌های برگ درخت می‌شوند.

شکل ۱ دو درخت عبارت ارزیابی را برای دید $V = R_1 \bowtie R_2 \bowtie R_3 \bowtie R_4 \bowtie R_5 \bowtie R_6$ 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

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 $n=3$ داریم. برای آن ΔV به صورت رابطه ۷ محاسبه می‌شود:

$$\Delta V = (\Delta R_1 \bowtie R_2 \bowtie R_3) \cup (R_1' \bowtie \Delta R_2 \bowtie R_3) \cup (R_1' \bowtie R_2' \bowtie \Delta R_3) \quad (7)$$

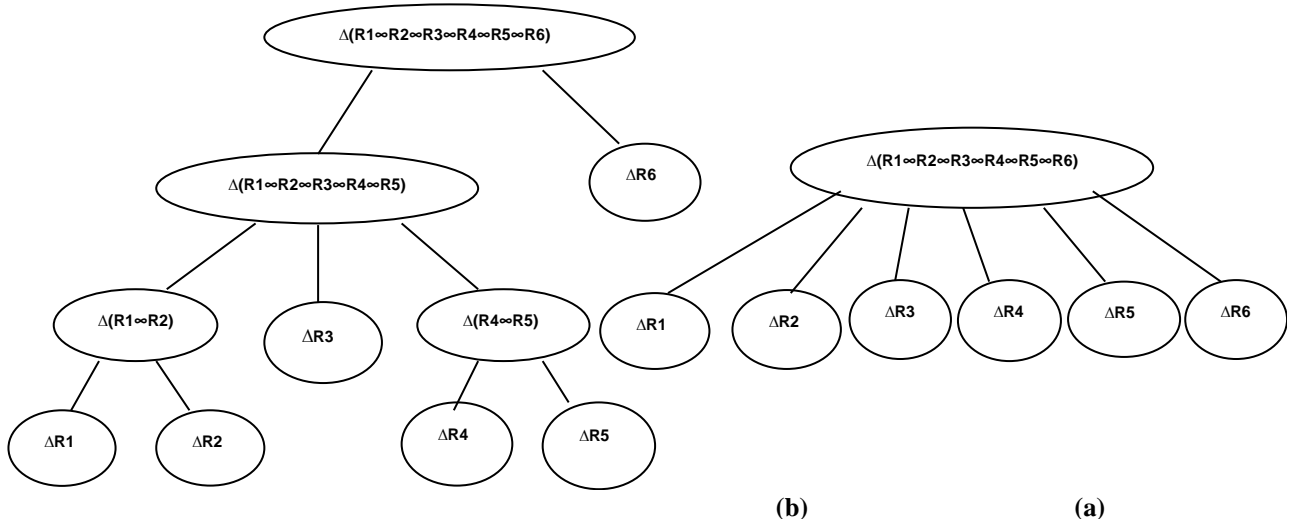
هزینه عبارت ۷ به صورت عبارت ۸ محاسبه می‌شود،

$$\begin{aligned} \text{Cost}(\Delta V) &= \text{Cost}(\Delta R_1 \bowtie R_2 \bowtie R_3) + \\ &\text{Cost}(R_1' \bowtie \Delta R_2 \bowtie R_3) + \text{Cost}(R_1' \bowtie R_2' \bowtie \Delta R_3) \\ &= \text{Cost}(|\Delta R_1| + |R_2| + |R_3|) + \\ &\text{Cost}(|R_1'| + |\Delta R_2| + |R_3|) + \text{Cost}(|R_1'| + |R_2'| + \\ &|\Delta R_3|) \end{aligned} \quad (8)$$

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

$$\Delta v = ((\Delta R_1 \bowtie R_2) \cup (R_1' \bowtie \Delta R_2)) \bowtie R_3 \cup (R_1' \bowtie R_2' \bowtie \Delta R_3) \quad (9)$$

در عبارت ۹ ابتدا عبارت $(\Delta R_1 \bowtie R_2) \cup (R_1' \bowtie \Delta R_2)$ ارزیابی می‌شود و سپس نتایج آن به R_3 متصل می‌شود. از این‌رو رابطه اصلی R_3 فقط یک‌بار در عبارت ۹ مورد دسترسی واقع شده است. در حالی که رابطه اصلی R_3 دوبار در رابطه ۷ دسترسی شده است. بدین ترتیب اگر اندازه R_3 بسیار بزرگ‌تر از اندازه‌های R_1 و R_2 باشد می‌توان به‌طور قابل ملاحظه‌ای هزینه محاسبه تغییرات دید



شکل ۱: درخت‌های ارزیابی دلتا [۱۲]

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

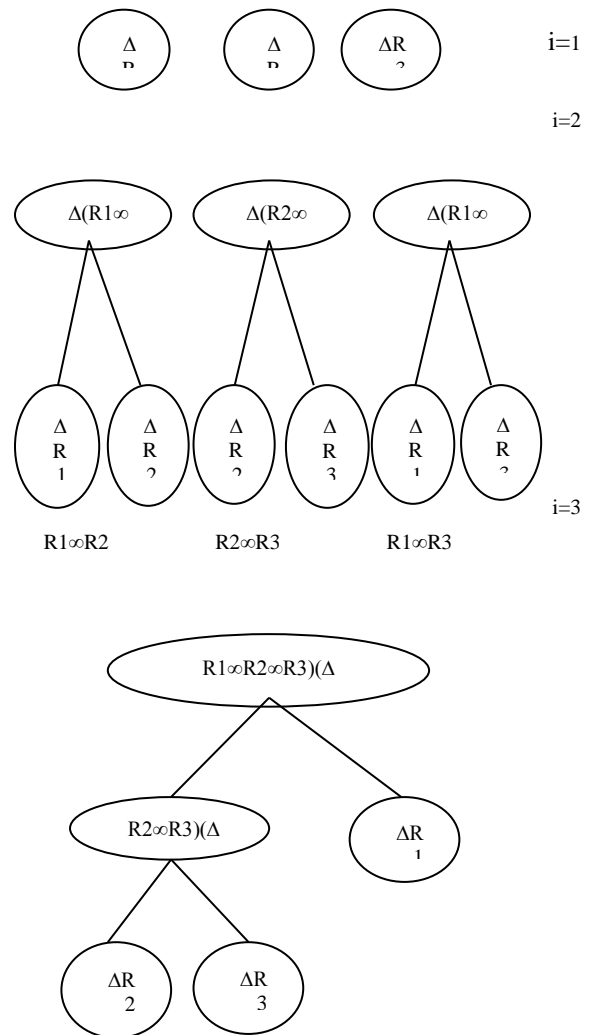
الگوریتم فرهنگی برای نخستین بار توسط Durham, W در سال ۱۹۹۴ معرفی شد [۲۸] و بعدها گسترش پیدا کرد و در سال‌های اخیر در مسایل مختلف بهینه سازی مورد استفاده قرار گرفته است [۱۰، ۱۱] [۳۴] [۳۵، ۳۷]. الگوریتم فرهنگی به‌طور کامل از روی جوامع بشری الگوبرداری شده است. به‌نحوی که بسیاری از معیارهای انسانی در گزینش 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^۱ 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^۲ اولیه که دقیق و جزئی نیست. بعد از چند نسل فضای باور جزئی‌تر 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 باور علمی و ارزشی باشد. همواره انسان‌ها خود را با افراد برجسته و برتر مقایسه می‌کنند و سعی دارند به آن‌ها

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 R_i ساخته می‌شود سپس با استفاده از نتیجه مرحله اول درخت ارزیابی دلتای بهینه برای $R_i \infty R_j (i \neq j)$ ساخته می‌شود.

در مرحله بعد با استفاده از نتیجه مرحله ۲ درخت ارزیابی دلتای بهینه برای $R_i \infty R_j \infty R_k (i \neq j \neq k)$ 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]

¹Population Space
²Belief Space

الگوریتم باکتریال

امروزه بسیاری از مسایل بهینه سازی از الگوریتم باکتریال استفاده می کنند [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 می گوئیم که والدینی که ارزش کمتری دارند ادامه نسل پیدا نمی کنند.

الگوریتم باکتریال بر اساس چهار مرحله پایه گذاری شده است.

حرکت

عملکرد گروهی

تولید مثل

حذف و پراکنندگی

الگوریتم زنبور عسل

الگوریتم زنبور عسل [۳۹] نیز مانند تمام الگوریتم های فراابتکاری با یک جمعیت اولیه آغاز می شود. یک کلونی زنبور عسل می تواند در مسافت زیادی و نیز در جهت های گوناگون پخش شود تا از منابع غذایی بهره برداری کند. پروسه ی جستجوی غذا به وسیله زنبورهای دیده بان آغاز می شود که برای جستجوی گلزارهای امید بخش فرستاده می شوند. هنگامی که جستجوی تمام گلزارها پایان یافت، هر زنبور دیده بان، بالای گلزاری که اندوخته کیفی مطمئنی از نکتار و گرده دارد، رقص خاصی را اجرا می کند. این رقص که به نام رقص چرخشی شناخته می شود، اطلاعات مربوط به جهت تکه گلزار (نسبت به کندو)، فاصله تا گلزار و کیفیت گلزار را به زنبورهای دیگر انتقال می دهد. این اطلاعات زنبورهای اضافی و پیرو را به سوی گلزار می فرستد. وقتی همه زنبورها به سمت ناحیه ای مشابه بروند، دوباره به صورت تصادفی و به علت محدوده ی رقصشان در پیرامون گلزار پراکنده می شوند تا به موجب این کار سرانجام نه یک گلزار، بلکه بهترین گل های موجود درون آن تعیین موقعیت شوند.

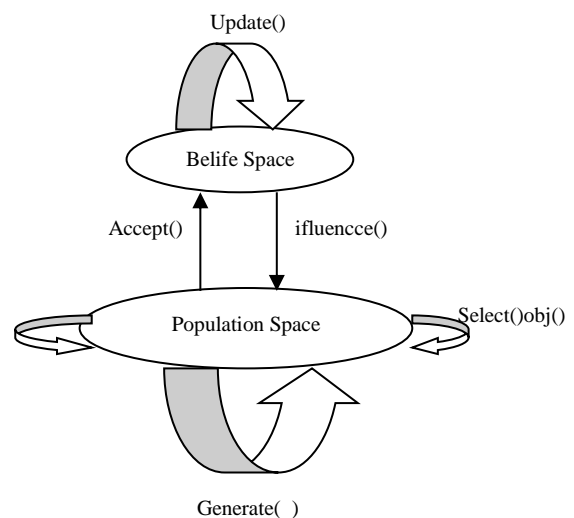
الگوریتم مبتنی بر تدریس و یادگیری

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 می کند. جمعیت اولیه به عنوان

برسند. در حقیقت آن ها یک معیار و یک فضای باور برای جمعیت می سازند. 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 می گیرد یا جزء یکی از اعضای عادی جمعیت می شود. ولی مساله ای که در الگوریتم فرهنگی مورد تاکید است آن است که احتمال آن که فرزندان آن ها در نسل های آینده جزء فضای باور قرار نگیرد کم است. نسل ها یکی پس از دیگری به وجود می آیند و با معیارهای فضای باور مقایسه می شوند، اگر توانستند به آن سطح برسند با یکی از افراد فضای باور که از نظر ارزشی جایگاه کمتری دارد جابه جا می شوند 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 انسان ها است. در حقیقت در الگوریتم فرهنگی تعداد خانواده ها مشخص می شوند نه تعداد افراد. مثلا گفته می شود صد خانوار داریم که جزیی از جمعیت عمومی هستند و یک خانواده داریم که در فضای باور جا می گیرد. در شکل ۳ استراتژی مربوط به الگوریتم فرهنگی ارائه شده است. در ابتدا یک جمعیت اولیه به صورت تصادفی تولید می شود و در Population Space قرار می گیرد.

سپس هر یک از افراد جمعیت مورد ارزیابی قرار می گیرند. سپس از بهترین به بدترین چینش می شوند و تعدادی از آن ها توسط Accept Function انتخاب می شوند و در فضای باور قرار می گیرند. در مرحله بعدی فضای جمعیت از فضای باور توسط InfluenceFunction برای تولید جمعیت جدید استفاده می کند.



شکل ۳: استراتژی 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 به دست 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^۱ 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 Intel(R) Core(TM) i5-4200U CPU 1.60GHz، 750GB 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. الگوریتم ژنتیک

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 دانش آموزان 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 این که 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 به صورت 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 مجدداً 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 $(2^{n-1} - 1)$. 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

^۱SortMin

$$\text{Cost}(\Delta V) = 10 + 51 + 87 + 20 + 54 + 78 + 23 + 21 = 315$$

با توجه به نتایج به دست آمده، رابطه ۱۱ کم هزینه ترین رابطه محسوب می شود. همان طور که مشاهده شد برای پیدا کردن کم هزینه ترین 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 می دهد.

محاسبه هزینه برای یک رابطه در الگوریتم های پیشنهادی همانند روش دلتای بهینه است با این تفاوت که با توجه به شبیه سازی های انجام شده، یک رابطه با کمترین میزان هزینه در کمترین زمان ممکن محاسبه می شود.

پارامترهای مختلف در پیاده سازی الگوریتم فرهنگی در نظر گرفته شده است که در ادامه به توصیف آن ها می پردازیم.

تعداد جمعیت اولیه: در واقع فضای اصلی جمعیت می باشد و با مقداری اولیه کار خود را شروع می کند. 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 می شود.

تابع پذیرش: این تابع افراد شایسته در هر نسل را انتخاب کرده و وارد فضای باور می کند. هدف پیدا کردن کمترین هزینه است. $p_{\text{accept}} = 0.1$ در نظر گرفته شده است.

فضای باور: بهترین راه حل های ارائه شده در هر نسل در فضای باور ذخیره می شود و برای تولید نسل بعد در نظر گرفته می شود. در هر نسل هزینه تمام درختان محاسبه شده و کم هزینه ترین آن ها در فضای باور قرار می گیرد. در واقع در هر نسل یک درخت با کمترین هزینه در فضای باور وجود دارد که هر نسل به روز می شود.

$$S = \{\text{MinSort}\}$$

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

تعداد جمعیت اولیه به ترتیب ۲۰، ۱۰۰ و ۵۰۰ در نظر گرفته می شوند. 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. در شکل ۵، سرعت نگهداری دید برای الگوریتم فرهنگی و روش ارزیابی دلتای بهینه آورده شده است.

یک الگوریتم پر کاربرد در هر زمینه کاری است. آزمایشات نشان می دهد الگوریتم پیشنهادی، از الگوریتم ژنتیک بهتر عمل می کند.

نتایج حاصل از شبیه سازی به همراه پارامترهای آن در جدول ۱ ارائه شده است. این جدول دیدی را نشان می دهد که از ۶ رابطه تشکیل شده است. در این جدول وزن های هر یک از پارامترهای یک رابطه نگهداری دید، به عنوان مثال ارائه شده است. $R, R', \Delta R$ هر یک اجزای یک رابطه نگهداری دید هستند. به عنوان مثال اندازه R_1 در ابتدا ۵۰ بوده و بعد از تغییراتی ($\Delta R_1 = 10$)، اندازه اش به ۲۰ رسیده است ($R_1' = 20$).

جدول ۱: جدول پارامترهای شبیه سازی

R1	R2	R3	R4	R5	R6
۵۰	۵۱	۷۸	۳۲	۱۵	۶۵
R1'	R2'	R3'	R4'	R5'	R6'
۲۰	۲۳	۵۱	۶۵	۹	۳۲
ΔR_1	ΔR_2	ΔR_3	ΔR_4	ΔR_5	ΔR_6
۱۰	۵۴	۲۱	۴	۸	۵۱

برای مثال هزینه نهایی یک رابطه برای $n=3$ را محاسبه می کنیم. $(2^{n-1} - 1)$ عبارت می توان برای نگهداری دید افزایشی نوشت [۳۲]. برای دیدی که از ۳ رابطه اصلی تشکیل شده است ۳ عبارت نگهداری افزایشی دید وجود دارد که در زیر آمده اند. با توجه به جدول ۲، هزینه اجرای تغییرات دید به صورت رابطه ۱۰ محاسبه می شود.

$$(10)$$

$$1) \Delta V = (\Delta R_1 \bowtie R_2 \bowtie R_3) \cup (R_1' \bowtie \Delta R_2 \bowtie R_3) \cup (R_1' \bowtie R_2' \bowtie \Delta R_3)$$

$$\text{Cost}(\Delta V) = (10 + 51 + 78) + (20 + 54 + 78) + (20 + 23 + 21) = 355$$

$$(11)$$

$$2) \Delta V = (\Delta (R_1 \bowtie R_2) \bowtie R_3) \cup (R_1' \bowtie R_2' \bowtie \Delta R_3)$$

$$= \Delta R_1 + R_2 + R_1' + \Delta R_2 + R_3 + R_1' \bowtie R_2' \bowtie \Delta R_3$$

$$\text{Cost}(\Delta V) = 10 + 51 + 20 + 54 + 78 + 20 + 23 + 21 = 277$$

$$(12)$$

$$3) \Delta V = (\Delta R_1 \bowtie R_2 \bowtie R_3) \cup (R_1' \bowtie \Delta (R_2 \bowtie R_3))$$

$$= \Delta R_1 + R_2 + R_3 + R_1' + \Delta R_2 + R_3 + R_2' + \Delta R_3$$

افزایش داشته، در عوض سرعت نگهداری دید با استفاده از این الگوریتم بین ۵۵ تا ۶۵ درصد افزایش داشته است. برآیند کاهش هزینه با استفاده از الگوریتم فرهنگی در ۶۵ درصد مواقع حدود ۵۸ درصد است. با توجه به نتایج آزمایشات مشاهده می شود که بیشترین برآیند کاهش هزینه زمانی است که فضای جمعیت ۱۰۰ در نظر گرفته شده است.

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

روش ارزیابی دلتای بهینه با استفاده از الگوریتم باکتريال

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

در ابتدا لازم است تا یک جمعیت اولیه تولید شود. جمعیت اولیه الگوریتم باکتريال شامل ترکیب‌های مختلف از عبارات نگهداری دید است که به صورت تصادفی تولید شده اند. به هر یک از اعضای این مجموعه یک باکتري گفته می‌شود. این باکتري‌ها چند نسل را می‌گذرانند و در هر نسل چندین مرحله را طی می‌کنند و در هر مرحله تغییری کنند 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

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

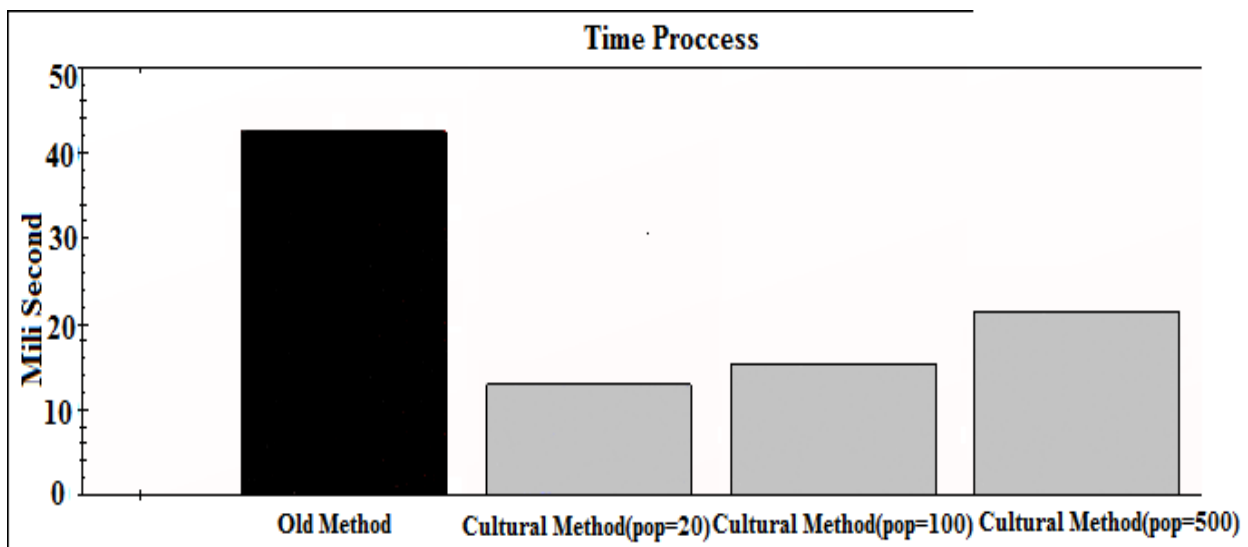
Model (Old Method)	Culture Method (Population Space=20)	Culture Method (Population Space=100)	Culture Method (Population Space=500)
۸۲۸	۱۰۹۷	۹۰۸	۸۲۸

با توجه به جدول ۲ و شکل ۵ مشاهده می شود وقتی جمعیت اولیه ۲۰ در نظر گرفته شود، سرعت نگهداری دید با استفاده از الگوریتم فرهنگی حدود ۷۰ درصد نسبت به روش ارزیابی دلتای بهینه افزایش داشته اما اندازه هزینه نیز حدود ۲۴ درصد افزایش می‌یابد. با توجه به فرمول شماره ۶ می دانیم که هزینه نهایی، برآیند هزینه نگهداری افزایشی دید و سرعت نگهداری دید افزایشی است. در نتیجه برآیند 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 نظر گرفته شده است. با توجه به 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 شود که اندازه هزینه نگهداری افزایشی دید در الگوریتم فرهنگی حدود هشت درصد (۰/۰۸)



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

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	Bacteria 1 Method	TLBO Method	Bee Method	culture Method
۸۲۸	۹۰۸	۸۲۸	۹۱۴	۹۰۸	۹۰۸

با توجه به فرمول شماره ۶ کاهش هزینه نهایی برآیند کاهش هزینه و افزایش سرعت است. سعی بر این است که عبارت نگهداری دید با کمترین هزینه در زمان کوتاه تری به دست آید. با توجه به شکل شماره ۶ مشاهده می شود 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 برآیند کاهش هزینه حدود ۴۶ درصد است. الگوریتم زنبور عسل حدود ۵۳ درصد افزایش سرعت و ۸ درصد افزایش هزینه دارد. برآیند کاهش هزینه با استفاده از این الگوریتم حدود ۴۵ درصد است. الگوریتم فرهنگی حدود ۶۰ درصد افزایش سرعت و ۸ درصد افزایش هزینه نسبت به روش ارزیابی دلتای بهینه دارد.

برآیند کاهش هزینه با استفاده از الگوریتم فرهنگی حدود ۵۲ درصد است. همان طور که مشاهده می شود الگوریتم فرهنگی بالاترین سرعت را نسبت به بقیه الگوریتم ها دارد.

۶- نتیجه گیری نهایی

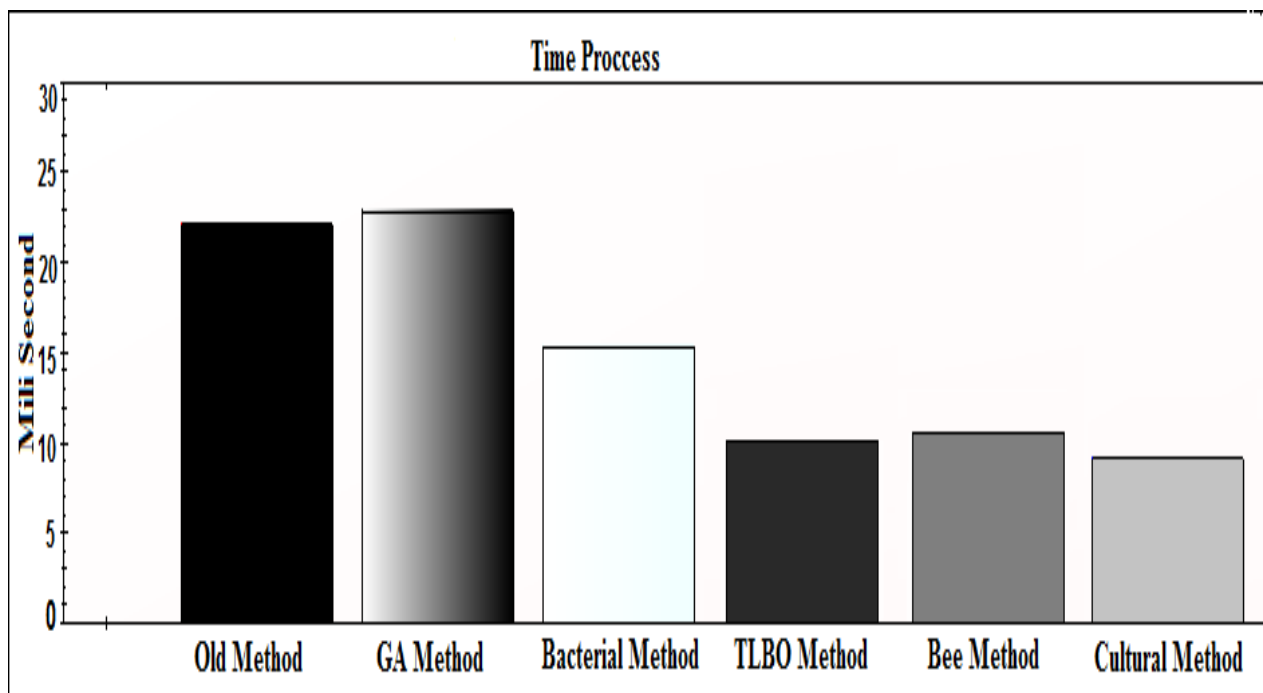
پایگاه داده تحلیلی، منبع داده ای است که برای بهبود سرعت پاسخ گویی به پرس و جوهای تحلیلی، از منابع داده پایه ایجاد می شود. به این منظور، پاسخ به پرس و جوها در این منبع داده، ذخیره می شود. به این پاسخ های ذخیره شده، دید ذخیره شده گفته می شود. با توجه به تغییر داده های اصلی در منابع داده پایه، این دیدها نیز باید به روزرسانی شوند. روش های نگهداری دید ذخیره شده به این دلیل مطرح شده اند. روش دلتای بهینه، یکی از روش های نگهداری افزایشی دید است. این روش عبارت نگهداری دیدی را پیدا می کند که کمترین هزینه را دارد. اما زمان زیادی را به منظور جستجوی رابطه بهینه صرف می کند. به این دلیل، در این مقاله، برای افزایش سرعت نگهداری دید، الگوریتم فرهنگی پیشنهاد شد. این الگوریتم از یک جمعیت اولیه استفاده می کند و شامل مجموعه ای از روابط برای نگهداری افزایشی دید است. در هر

الگوریتم زنبور عسل [۳۹] نیز مانند تمام الگوریتم های فراابتکاری با یک جمعیت اولیه آغاز می شود. جمعیت اولیه شامل ترکیب های مختلف قابل ایجاد برای یک عبارت نگهداری دید است که به صورت درخت ارزیابی دلتا نمایش داده می شود. جمعیت اولیه همان زنبورهای دیده بان هستند. تمام زنبورهای دیده بان گلزارهای اطراف را جستجو کرده و به دنبال منطقه ای با بیشترین گرده می گردند. اگر منطقه ای که یافته شد گرده بیشتری نسبت به منطقه قبل داشت منطقه قبل را فراموش می کنند و منطقه جدید را به خاطر می سپارند در غیر این صورت همان منطقه قبلی را به خاطر می سپارند. در انتها بقیه زنبورها به سمت بهترین منطقه حرکت می کنند. در واقع هر کدام از درخت ها درختان اطراف خود را بررسی می کند و به دنبال بهترین درخت می گردند. هر درخت از میان درختان اطرافش درختی با کمترین هزینه را انتخاب کرده و به خاطر می سپارد. از میان درختان جمعیت اولیه، درختی با کمترین هزینه انتخاب شده و بقیه درختان سعی می کنند به این درخت نزدیک شوند. چندین نسل می گذرد و در هر نسل هر درخت عبارت ارزیابی دلتا سعی می کند درخت بهینه را به خاطر بسپارد و بقیه درخت ها نیز به سمت درخت بهینه حرکت می کنند، تا در نهایت بهترین منطقه یعنی درختی با کمترین هزینه پیدا می شود.

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 می شود. بهترین و کم هزینه ترین درخت انتخاب می شود و به عنوان معلم در نظر گرفته می شود. حال تمام درختان بر اساس بهترین درخت تغییر داده می شوند و سعی می کنند به بهترین درخت نزدیک شوند. اگر درختان جدید از درختان قبلی هزینه کمتری داشتند پذیرفته می شوند در غیر این صورت درخت قبلی باقی می ماند. در فاز یادگیری، درختان از هم تاثیر می پذیرند. در اینجا فرض می شود هر درخت به صورت تصادفی با یک درخت دیگر در ارتباط است. از جمعیت تولید شده ۲ درخت انتخاب می شود و سعی می کنند به درخت با کمترین هزینه نزدیک شوند. اگر درخت تولید شده از درختان انتخابی بهتر بود پذیرفته شده و در غیر این صورت درختان انتخابی باقی می ماند.

با توجه به اندازه های جدول ۱ در جدول شماره ۳ هزینه نگهداری دید الگوریتم ها و در شکل شماره ۶ سرعت نگهداری دید الگوریتم ها با یکدیگر مقایسه شده است.



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

- [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 view maintenance", 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 Mining", International Journal on Computational Sciences & Applications (IJCSA) 2(4):9-23, August 2012.

[۱۲] ع. کریمی مصدق، ن. دانشپور، "کاهش هزینه نگهداری افزایشی دید

پایگاه داده تحلیلی با استفاده از الگوریتم‌های فراابتکاری"، سومین کنفرانس

بین المللی اطلاعات، حال و آینده ۲۰۱۴.

- [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 incremental maintenance 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, "Optimising Refresh of a Set of

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.

vehicles”, IEEE Congress on Evolutionary Computation (CEC), pp. 2685-2691, 2014.

- Materialized View”, Proceedings of the 31st VLDB 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, “Effective Maintenance 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.almazayad, 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. Dessoach, “View Maintenance using Partial Deltas”, *BTW, LNI P – 180, Kaiserslautern, Germany*, pp. 287-306, 2011.
- [33] B. Qin, S. Wang, X. Du, “Effective Maintenance 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, and Human Diversity”, *Stanford University Press*, 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 using cultural algorithm”, *9th Conference on Industrial 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 Control Conference (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 TLBO algorithm for RBF neural modelling of batteries in electric

Improving the speed of view maintenance in data warehouses

Afifeh Karimi Mosadegh¹, 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 Rajaei Teacher Training University, Tehran, Iran.

¹karimi.mosadegh@yahoo.com. ^{2*}ndaneshpour@srttu.edu

Abstract- In recent years, due to the use of data warehouses, the subject of materialized view maintenance 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 maintenance. At this time, algorithms are presented in order to view maintenance 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 show 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.