بسم الله الرحمن الرحیم

چکیده:

بهینه‌سازی پرس‌وجوهای پایگاه‌دادها، یکی از مسائل پژوهشی مشکل است. روش‌های جستجوی جامع مانند: برنامه-نویسی پویا برای پرس‌وجوهای با تعداد روابط کم، مناسب هستند ولی با افزایش تعداد روابط موجود در پرس‌وجو، بدلیل نیاز به مصرف زیاد حافظه و پردازش، استفاده از این روش‌ها مناسب نخواهند بود، بنابراین لازم است که از روش‌های تصادفی و تکاملی استفاده شود. استفاده از روش‌های تکاملی بدلیل کارایی و قدرتمندی آنها، تبدیل به ناحیه پژوهشی مناسبی در زمینه بهینه‌سازی پرس‌وجو گردیده است. در این پژوهش یک الگوریتم تکاملی بنام فاخته را برای حل مساله بهینه‌سازی ترتیب اجرای عملگرهای پیوند در پرس‌وجوهای پایگاه‌داده‌ای پیشنهاد شده است. الگوریتم بهینه سازی فاخته مشهورترین پارازیت اولادی می باشد که به نوعی یک متخصص در زمینه فریب بی رحمانه می باشد. استراتژی این پرنده شامل خفیه کاری، شگفت زده کردن و سرعت عمل است. بنابراین ما در این تحقیق از الگوریتم تکاملی فاخته در فرآیند پیداکردن جواب بهینه سراسری استفاده کرده‌ایم. که این الگوریتم روی پایگاه داده های استاندارد در مقایسه با کارهای مرتبط دیگر به جواب بهینه سراسری بهتری رسیده است با توجه به عملکرد الگوریتم ژنتیک، الگوریتم فاخته در تعداد تکرار کمتر به بهترین جواب می‌رسد.

کلمات کلیدی: پایگاه‌داده رابطی– بهینه‌سازی– الگوریتم‌های تکاملی– الگوریتم فاخته

فهرست مطالب

عنوان **صفحه**

فصل اول :کلّیات پژوهش

مقدمه..... ۲

۱-۱- بیان مسئله ۲

۱-۲- اهمیت موضوع ۳

۱-۳- روش پیشنهادی ۴

۱-۴- ساختار پایان‌نامه... ..... ۵

فصل دوم :کارهای پیشین

مقدمه..... ۷

۲-۱- بررسی نوشتارهای وابسته ۷

فصل سوم : مفاهیم پایه ای

۳-۱- پایگاه‌داده ۱۱

۳-۲- مدل‌های پایگاه‌داده ۱۱

۳-۳- معماری پردازش پرس‌وجو ۱۲

۳-۴- بهینه‌ساز پرس‌وجو ۱۳

۳-۵- تولید طرح اجرای بهینه ۱۴

۳-۶- الگوریتم‌های فرامکاشفه‌ای ۱۵

۳-۶-۱- الگوریتم بهینه‌سازی ژنتیک ۱۵

۳-۶-۲- الگوریتم بهینه‌سازی فاخته ۱۶

۳-۶-۳- جزئیات الگوریتم بهینه‌سازی فاخته ۱۹

۳-۶-۴- از بین بردن فاخته‌ها..... ۲۳

۳-۶-۵- همگرایی الگوریتم..... ۲۳

۳-۶-۶- برتری الگوریتم فاخته نسبت به سایر الگوریتم‌ها ۲۴

۳-۶-۷- سوالات مهم در مورد الگوریتم فاخته ۲۵

فصل چهارم : روش پیشنهادی

۴-۱- مجموعه داده ۲۸
۴-۱-۱- شماره بندی جدول‌ها مجموعه داده ۲۸
۴-۱-۲- ماتریس زمان الحاق بین جدول‌ها ۳۰
۴-۱-۳- جدول مربوط به ارتباط‌ها ۳۱
۴-۱-۴- جدول رکوردهای مرتبط ۳۲
۴-۱-۵- جدول رکورد مرتبط برای ایجاد گراف ۳۲
۴-۱-۶- تشریح الگوریتم پیشنهادی ۳۳
۴-۱-۷- ساختار الگوریتم فاخته استفاده شده ۳۳
۴-۱-۸- باز تعریف عملگر مهاجرت ۳۵
۴-۱-۹- گسسته‌سازی برای این مسئله ۳۶
۴-۱-۱۰-تعریف تخم‌گذاری ۳۷
۴-۱-۱۱- باز تعریف عملگر مهاجرت ۳۹
۴-۱-۱۲- تابع هزینه ۴۱
۴-۲- نحوه و مشخصات پیاده‌سازی ۴۳

فصل پنجم : ارزیابی و بررسی نتایج عملی

۵-۱- ساختار ورودی مورد ارزیابی..... ۴۶
۵-۲- پیاده‌سازی الگوریتم ژنتیک و نتایج آن..... ۴۶
۵-۳- پیاده‌سازی الگوریتم فاخته و نتایج آن..... ۵۱
۵-۴- مقایسه و ارزیابی عددی دو الگوریتم ۵۶
۵-۵- تحلیل نتایج ۵۹
۵-۶- کارهای آتی ۶۰
مراجع ۶۱

فهرست شکل‌ها

عنوان شکل **صفحه**

شکل ۳-۱- معماری پرس‌وجو. ۱۲

شکل ۳-۲- روندنمای اجرای الگوریتم فاخته..... ۱۸

شکل ۳-۳- نمایش نحوه تعیین ELR و تخمگذاری در آن محدوده ۱۹

شکل ۳-۴- نمایش نحوه مهاجرت پرندگان ۲۱

شکل ۳-۵- شبه کد الگوریتم گسسته بهینه‌سازی فاخته DCOA1 ..... ۲۳

شکل ۴-۱- قسمتی از ماتریس زمانهای الحاق ۳۰

شکل ۴-۲- قسمتی از جدول ارتباط‌ها ۳۱

شکل ۴-۳- قسمتی از جدول رکوردهای مرتبط ۳۲

شکل ۴-۴- قسمتی از جدول مرتبط و رکوردهای آنها برای ایجاد گراف ۳۲

شکل ۴-۵- روندنمای اجرای پژوهش ۳۳

شکل ۴-۶- روندنمای تغییر داده‌شده الگوریتم فاخته ۳۴

شکل ۴-۷- شبه کد الگوریتم گسسته فاخته مسائل غیر جایگشتی DCOA2 ..... ۳۶

شکل ۴-۸- نمایش بردار x با بیتهای انتخاب شده ۳۸

شکل ۴-۹- اعمال جهش روش ۱ بر روی بردار x ۳۸

شکل ۴-۱۰- نمایش بردار x با بیتهای انتخاب شده ۳۸

شکل ۴-۱۱- اعمال جهش روش ۲ بر روی بردار x ۳۹

شکل ۴-۱۲- نمایش بردار x با بیتهای انتخاب شده ۳۹

شکل ۴-۱۳- اعمال جهش روش ۳ بر روی x ۳۹

شکل ۴-۱۴- نمایش ترتیب برای مثال بهینه‌ساز طرح اجرا برای ۸ جدول ۴۰

شکل ۴-۱۵- اعمال برای استخراج لیست حرکتها (مهاجرت) ۴۰

شکل ۴-۱۶- شبه کد الگوریتم گسسته فاخته برای مسائل جایگشتی DCOA3 ..... ۴۱

شکل ۴-۱۷- مثالی از ماتریس زمان و نحوه محاسبه مقدار برازش هر جواب ۴۲
شکل ۴-۱۸- یک جواب فرضی برای نشان دادن نحوه محاسبه تابع برازش ۴۲
شکل ۴-۱۹- نمونه‌ای از جواب برای الگوریتم فاخته ۴۳
شکل ۴-۲۰- نمونه‌ای از نحوه ورود اطلاعات به الگوریتم‌ها ۴۴
شکل ۵-۱- نتایج اول از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم ژنتیک استاندارد (۱۰۰ بار تکرار) ۴۸
شکل ۵-۲-نتایج دوم از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم ژنتیک استاندارد (۲۰۰ بار تکرار) ۴۸
شکل ۵-۳-نتایج سوم از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم ژنتیک استاندارد (۳۰۰ بار تکرار) ۴۹
شکل ۵-۴-نتایج چهارم از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم ژنتیک استاندارد (۴۰۰ بار تکرار) ۵۰
شکل ۵-۵-نتایج پنجم از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم ژنتیک استاندارد (۵۰۰ بار تکرار) ۵۰
شکل ۵-۶-نتایج ششم از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم ژنتیک استاندارد (۱۰۰۰ بار تکرار) ۵۱
شکل ۵-۷-نتیجه اول از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم فاخته (۱۰۰ بار تکرار) ۵۳
شکل ۵-۸-نتیجه دوم از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم فاخته (۲۰۰ بار تکرار) ۵۴
شکل ۵-۹-نتیجه سوم از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم فاخته (۳۰۰ بار تکرار) ۵۴

شکل ۵-۱۰-نتیجه پنجم از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم فاخته (۵۰۰ بار تکرار) ۵۵
شکل ۵-۱۱-نتیجه ششم از شش آزمایش با تکرارهای متفاوت برای دستور الحاق جداول ۱ تا ۳۰ با الگوریتم فاخته (۱۰۰۰ بار تکرار) ۵۶

فهرست جدول‌ها

عنوان جدول **صفحه**

جدول ۵-۱-پیکربندی پارامترهای الگوریتم ژنتیک استفاده شده.....۴۷

جدول ۵-۲-پارامترهای اجرای الگوریتم فاخته بر روی مجموعه داده. ۵۲

جدول ۵-۳-مقایسه دو الگوریتم ژنتیک و فاخته برای حل مسئله پرسوجو الحاق جدولها در بانک‌های اطلاعاتی متمرکز ۵۷

فصل اول

کلّیات پژوهش

مقدمه

مدل داده‌ای رابطه‌ای توسط کاد (Codd,1970)، معرفی شده است و در سال‌های اخیر، سیستم‌های پایگاه-داده رابطه‌ای بعنوان استانداردی در انواع کاربردهای علمی و تجاری، شناخته می‌شوند. کار بر روی عملگر پیوند، بدلیل هزینه‌های ارزیابی بالای این عملگر، هدف اولیه بهینه‌سازهای پرس‌وجوی رابطه‌ای است. در صورتی که تعداد روابط بیش از ۵ یا ۶ رابطه باشد، روش‌های جستجوی جامع، از لحاظ حافظه و زمان پرهزینه خواهند بود. پرس-وجوهای با تعداد پیوند زیاد در سیستم‌های جدید مانند سیستم‌های مدیریت پایگاه داده استنتاجی[1] سیستم های خبره،[2] سیستم‌های مدیریت پایگاه‌داده‌های مهندسی و غیره با روش‌های فرامکاشفه و تکاملی صورت می‌گیرد.

با توجه به طبیعت الگوریتم‌های تکاملی و اینکه در اکثر مواقع مقاوم و کاراتر می‌باشند و با در نظر گرفتن کارهای صورت گرفته در این زمینه، مناسب‌ترین گزینه برای حل این مسئله، استفاده از الگوریتم‌های تکاملی می‌باشد. اولین کار انجام شده بر روی مسئله بهینه‌سازی ترتیب پیوندها با استفاده از الگوریتم ژنتیکی، توسط بنت و همکارانش ارایه گردید (Horng, Kao,et al.,1994). در حالت کلی الگوریتم‌های تکاملی در مقایسه با الگوریتم برنامه‌نویسی پویا در اکثر مواقع دارای هزینه کمتری می‌باشد. از ویژگی‌های دیگر این الگوریتم، قابلیت بکارگیری آن در معماری موازی می‌باشد.

۱-۱- بیان مسئله

بهینه‌سازی پرس‌وجو، فعالیتی است که طی آن، یک طرح کارا برای اجرای پرس وجو تولید می‌گردد. طرح اجرای پرس و جو (QEP[3]) یکی از مراحل اساسی در پردازش پرس وجو می‌باشد. در این مرحله، سیستم مدیریت پایگاه‌داده از بین تعدادی طرح اجرا، بهترین طرح را بر می‌گزیند، بطوری‌که اجرای پرس‌وجوی داده شده توسط کاربر با استفاده از این طرح دارای کمترین هزینه می‌باشد. ورودی بهینه‌ساز، نمایش داخلی پرس‌وجو می‌باشد که توسط کاربر به سیستم مدیریت پایگاه‌داده، وارد شده است. هدف کلی از بهینه‌سازی پرس‌وجو، انتخاب کاراترین طرح اجرایی برای دستیابی به داده مناسب و پاسخ به پرس‌وجوی داده شده می‌باشد. به بیان دیگر، در صورتی که مجموعه همه طرح های اجرایی اختصاص داده شده برای پاسخ به پرس‌وجوی Q را با S نشان دهیم، هر عضو QEP متعلق به

[1] Deductive
[2] System Expert
[3] Query Execution Plan

مجموعه S دارای هزینه $Cost_{(\mathrm{QEP})}$ می‌باشد که این هزینه شامل زمان پردازشی و زمان ورودی /خروجی می‌باشد. هدف هر الگوریتم بهینه‌سازی یافتن عضوی مانند QEP_0 متعلق به مجموعه S می‌باشد، به نحوی که :

$$Cost_{\mathrm{QEP}} = min_{QEP \in S}\ Cost_{(\mathrm{QEP})}$$

طرح اجرایی برای پاسخ به یک پرس وجو، دنباله‌ای از عملگرهای جبر رابطه‌ای می‌باشد که بر روی روابط پایگاه داده اعمال می‌شود و جواب لازم برای آن پرس وجو را تولید می‌کند. از بین عملگرهای رابطه‌ای موجود، پردازش و بهینه‌سازی عملگر پیوند[۱] که بوسیله نماد ∞ نمایش داده می‌شود، جزو مشکل‌ترین عملگرها می‌باشد. اساساً، عملگر پیوند دو رابطه را به‌عنوان ورودی می‌گیرد و تاپل‌های آنها را یک به یک بر اساس معیار مشخصی، ترکیب کرده و یک رابطه جدید را به عنوان خروجی تولید می‌کند. از آنجا که عملگر پیوند دارای خاصیت انجمنی[۲] و جابجایی[۳] می باشد، تعداد طرح‌های اجرایی موجود برای پاسخ دهی به یک پرس وجو با افزایش تعداد پیوندهای بین روابط، به صورت نمایی رشد می‌کند. علاوه بر این موارد، معمولاً یک سیستم مدیریت پایگاه‌داده، انواع روش‌های پیاده‌سازی عملگر پیوند را برای پردازش پیوندها و انواع اندیس‌ها را برای دسترسی به روابط پشتیبانی می‌کند. بطوریکه این موارد، گزینه‌های لازم برای پاسخ دهی به یک پرس وجو را هر چه بیشتر می‌کند. گرچه تمامی طرح‌های اجرایی موجود برای پاسخ به یک پرس وجوی مشخص، دارای خروجی یکسان می‌باشند اما از آنجا که کاردینالیتی روابط میانی ایجاد شده یکسان نیستند، طرح‌های اجرایی بوجود آمده دارای هزینه متفاوتی خواهند بود. بنابراین، انتخاب ترتیب مناسب برای اجرای عمل پیوند در هزینه کلی، تاثیرگذار است. مسئله بهینه‌سازی پرس‌وجو که اغلب از آن به‌عنوان مسئله انتخاب ترتیب مناسب برای اجرای عملگرهای پیوند[۴] یاد می‌شود، یک مسئله NP-Hard می‌باشد.

۱-۲- اهمیت موضوع

با توجه به گسترش داده‌ها و افزایش تعداد آنها، ایجاد رابطه مناسب بین جدول‌های اطلاعاتی و استخراج سریع خواسته‌های کاربر از بین این اطلاعات عظیم، نیازمند ایجاد روشی سریع‌تر برای ایجاد مسیر و استخراج داده می‌باشد.

[1] Join
[2] Associative
[3] Commutative
[4] Join Ordering Problem

بهینه‌سازی پرس‌وجوهای پایگاه‌داده، یکی از مسائل تحقیقاتی مشکل است. تکنیک‌های جستجوی جامع مانند برنامه‌نویسی پویا برای پرس وجوهای با تعداد روابط کم، مناسب هستند ولی با افزایش تعداد روابط موجود در پرس‌وجو، بدلیل نیاز به مصرف زیاد حافظه و پردازش، استفاده از این روش‌ها مناسب نخواهند بود، بنابراین استفاده از روش‌های تصادفی و تکاملی پیشنهاد می‌شود. استفاده از روش‌های تکاملی بدلیل کارایی و قدرتمندی آنها، تبدیل به ناحیه تحقیقاتی مناسبی در زمینه بهینه‌سازی پرس‌وجو گردیده است.

یک مسأله بهینه‌سازی، مسأله یافتن بهترین راه حل از میان همه راه حل های عملی می‌باشد. مسأله‌های بهینه‌سازی می‌تواند به دو دسته تقسیم شود که متغیرها پیوسته یا گسسته باشند. یک مسأله بهینه‌سازی با متغیرهای گسسته به عنوان یک مسأله بهینه‌سازی ترکیبی شناخته می‌شوند.

بهینه‌سازی در سیستم‌های رابطه‌ای هم به عنوان یک چالش و هم به عنوان یک فرصت مطرح می‌گردد. چالش از آن جهت که همیشه برای رسیدن به کارایی قابل قبول در چنین سیستم‌هایی به بهینه‌سازی نیازمندیم و فرصت از آن جهت که این مساله دقیقا یکی از نقاط قوت رویکرد رابطه‌ای است زیرا عبارات رابطه‌ای به اندازه کافی در سطح معنایی بالایی قرار دارند که بهینه‌سازی به بهترین نحو در آنها قابل اعمال باشد.

هدف بهینه‌ساز، انتخاب یک استراتژی برای ارزیابی عبارت رابطه‌ای داده شده است و این روش برای بهینه سازی پرس‌وجو بسیار مناسب می‌باشد. کار بهینه ساز پرس و جو این است که یک طرح برای ارزیابی پرس و جو ارائه کند که همان نتایج پرس و جوی اولیه را با کمترین هزینه ممکن (و یا حداقل با اختلاف هزینه کم نسبت به کمترین هزینه ممکن) ایجاد کند. برای انتخاب از میان طرح‌های ارزیابی پرس و جوی متفاوت، بهینه‌ساز باید هزینه طرح‌ها را تخمین بزند زیرا محاسبه هزینه دقیق ارزیابی یک طرح معمولا بدون ارزیابی واقعی آن ممکن نیست. در عوض بهینه‌ساز با استفاده از اطلاعات آماری در مورد روابط تخمین خوبی از هزینه طرح ارائه می‌کند و عمدتا میزان دستیابی به دیسک هزینه پردازش را تعیین می‌کند.

دو عبارت جبر رابطه‌ای معادل گفته می‌شوند اگر روی هر نمونه پایگاه‌داده معتبر هر دو عبارت مجموعه تاپل‌های یکسانی تولید کنند. یک قانون هم ارزی می‌گوید که دو عبارت با شکل ظاهری متفاوت، معادلند.

۱-۳-روش پیشنهادی

در این پژوهش استفاده از یک الگوریتم تکاملی برای حل مساله بهینه‌سازی ترتیب اجرای عملگرهای پیوند

?

در پرس‌وجوهای پایگاه داده‌ای پیشنهاد شده است. جدول‌های رابطه‌ای طبق کلیدی با هم ارتباط دارند در صورتی که بین دو جدول کلید وجود نداشته باشد در واقع این دو جدول نمی‌توانند با هم در ارتباط باشند مگر به واسطه جدول دیگری. روش پیشنهادی از بهینه‌سازی الگوریتم فاخته برای جستجو در فضای حالات مسئله استفاده می‌نماید. حالات مسئله ترتیب‌های ممکن از جدول‌های مورد درخواست پرس‌وجو می‌باشد که این ترتیب‌ها با روش تکاملی فاخته ایجاد، تغییر، اصلاح، ارزیابی و تکرار می‌شود. در قسمت تولید جواب و پیشرفت به طرف جواب‌های بهینه از گسسته‌سازی ابتکاری استفاده شده‌است و در تمام مراحل تولید جواب جدید جایگشت‌ها مراعات شده‌است. در این پژوهش، نشان داده شده است که با استفاده الگوریتم فاخته در فرایند جستجو، سرعت رسیدن به جواب افزایش پیدا کرده و از بدام افتادن الگوریتم در بهینه‌های محلی جلوگیری می‌شود.

نتایج آزمایش‌ها، برتری الگوریتم فاخته را نسبت به روش‌های مبتنی بر الگوریتم ژنتیک نشان می‌دهد.

۱-۴-ساختار پایان‌نامه

این پایان‌نامه در پنج فصل تنظیم شده است. در فصل دوم تعاریفی در خصوص کارهای پیشین آورده شده است که مشتمل بر بررسی چند مقاله در راستای این پایان‌نامه می‌باشد. در فصل سوم نیز مفاهیم پایه برای موضوع پایان‌نامه ارائه شده است که این تعاریف شامل بهینه‌سازی پرس‌وجو، پایگاه داده و الگوریتم‌های تکاملی و بخصوص الگوریتم تکاملی استفاده شده در این پژوهش می‌باشد. در فصل چهارم روش پیشنهادی بکارگیری الگوریتم فاخته در بهینه‌سازی پرس‌وجوی پایگاه داده‌ای رابطه‌ای بیان شده‌است. در این فصل سیاست جذب متفاوتی برای الگوریتم فاخته شرح داده‌شده است. در فصل پنجم نتایج حاصل از ارزیابی پیاده‌سازی الگوریتم و اجرای آن برای پایگاه داده بیان‌شده است و جهت بررسی کارایی الگوریتم پیشنهادی این نتایج با نتایج حاصل از الگوریتم ژنتیک استاندارد پیاده سازی شده مقایسه شده است.

فصل دوم

کارهای پیشین

مقدمه

بهینه‌ساز پرس‌وجو از اهمیت زیادی برای پایگاه داده ارتباطی برخوردار است، مخصوصاً برای اجرای دستورات یک بهینه‌ساز پرس وجو بهترین استراتژی بر اجرای هر پرس وجو را تعیین می‌کند. بهینه ساز به عنوان مثال یک پرس و جو SQL پیچیده را انتخاب می‌کند؛ که آیا از شاخص برای یک پرس وجو مشخص استفاده کند یا نه وکدام تکنیک الحاق استفاده شود هنگامی که جداول با هم الحاق می‌شوند. بهینه‌سازهای جستجو معمولاً بر مبنای هزینه می‌باشند. در یک استراتژی بهینه‌سازی بر مبنای هزینه، طرح‌های اجرایی چندگانه ای برای یک پرس‌وجو مشخص تولید می‌شود و آنگاه یک هزینه تخمینی برای هر طرح محاسبه می‌گردد. بهینه ساز پرس‌وجو طرحی که دارای کمترین هزینهٔ تخمینی است را انتخاب می کند(DONG, LI,2005) .

۲-۱-بررسی نوشتارهای وابسته

بهینه‌سازی پرس‌وجو قبل از اجرای پرس وجو مهم است واین اصل در پایگاه‌داده رابطه‌ای و توزیع شده اهمیت زیادی دارد. در پایگاه‌های با اندازه بزرگ، مسئله بهینه‌سازی پرس‌وجو ماهیتی NP-Hard پیدا می‌کند و حل آن بسیار مشکل است. به همین دلیل به دنبال روش‌هایی می‌گردیم تا در زمان کمتر و با هزینه کمتر جوابی نزدیک به بهینه بدهند، در این راستا از الگوریتم تکاملی فرا مکاشفانه‌ای مانند ژنتیک و الگوریتم بهینه‌سازی فاخته برای محاسبه راه حل بهینه استفاده می‌کنید(Johnson, Srivatsa,2015) .

اُسزو مطرح کرده است که هدف بهینه‌سازی پرس وجو پیدا کردن ترفند اجرای نزدیک به بهینه برای پرس وجو ها است. یک ترفند اجرا برای پرس وجو ممکن است با عملوندهای جبر رابطه‌ای و ارتباط‌های اولیه برای انتقال داده توصیف شود. دراین مقاله بیان شده که انتخاب ترفند بهینه، به طور کلی نیازمند اسناد هزینه اجرای حل کننده های داوطلب برای اجرای پرس وجو می‌باشد (Özsu, Valduriez,1996).(سنگینی ترکیب I/O،CPU)

کاسمن درباره استفاده جزئی از فهرست‌ها با استفاده از بهینه‌ساز برای اجرای یک پرس وجو و اینکه در چه ترتیبی از عملگرها باید اجرا شوند بحث کرده است. بهینه‌ساز راه‌های فرعی را بر می‌شمارد، با استفاده از مدل هزینه، هزینه هر روش را تخمین می‌زندو طرحی که کمترین هزینه دارد را بر می‌گزیند(Kossmann,Stocker,2000)

ایبراراکیت مطرح کرده است که انتخاب ترفند بهینه برای یک پرس‌وجو در تعدادی از ارتباطات از نوع NP-Hard است. پرس وجوهای با تعدادجداول زیاد، هزینه‌های هنگفت بهینه‌سازی را تحمیل می‌کنند، اگرچه ماموریت

?

حقیقی بهینه‌ساز، پیدا کردن ترفند نزدیک به بهینه واجتناب از ترفندهای بد است (Ibaraki, Kameda,1984)

جی.گرایفی مطرح کرده است که اگرچه پردازش پرس وجو ها یک مسئله خوش مطالعه[1]است، معماری‌های مدرن چالش‌ها و فرصت‌های جدیدی را برای موازی سازی در همه سطوح اجرا تا سطح کار با دستگاه مطرح می کند (Graefe,2009)

ا.گرلیتز مطرح کرده است که بهینه‌سازی ممکن است به دو بخش کلی بهینه‌سازی محلی برای انتخاب ترفندهای دسترسی وبهینه‌سازی کلی در جایی که ترتیب پیوند وانتخاب پایگاه، کارهای اصلی هستند تقسیم شود. برای انتخاب پایگاه، دو گزینه اصلی وجود دارد: ترابری داده‌ها جایی که داده‌ها از پایگاه ذخیره‌سازی به پایگاه اجرای پرس وجو منتقل شده است و ترابری پرس وجو جایی که ارزیابی زیر پرس وجو به نمایندگی از پایگاه ذخیره‌سازی است (Görlitz,Thimm,et al.,2012).

بهینه‌سازی پرس‌وجوی پایگاه‌داده باتکنیک‌های جستجوی جامع مانند برنامه‌نویسی پویا برای پرس‌وجوهایی با تعداد روابط کم مناسب هستند ولی با افزایش تعداد روابط موجود در پرس وجو، به دلیل نیاز به مصرف زیاد حافظه و پردازش، استفاده از این روش‌ها مناسب نخواهد بود؛ بنابراین مجبور به استفاده از روش‌های تصادفی و تکاملی هستند. یک الگوریتم تکاملی ترکیبی برای حل مسئله بهینه‌سازی ترتیب اجرای عملگرهای پیوند در پرس وجوهای پایگاه‌داده رابطه‌ای پیشنهاد شده است. کار بر روی عملگر پیوند به دلیل هزینه‌های ارزیابی بالای این عملگر، هدف اولیه پرس وجوی رابطه‌ای است. در صورتی که تعداد روابط بیش از ۵ یا ۶ رابطه باشد تکنیک‌های جستجوی جامع، از لحاظ حافظه و زمان پر هزینه خواهد بود (Grosan,et al.,2007)

لِنزِلوت در مقاله خود درباره ترفندهای شمارش کافی، هنگام بهینه‌سازی پرس‌وجوهای پیچیده برای تعداد زیادی طرح های اجرایی بحث کرده است (Hlaing, Khine,2011). برای حل این مشکل، ترفندهای تصادفی مورد استفاده قرار گرفتند. به طور کلی، ناهماهنگی و خودمختاری منابع داده، نوع سیستم‌های ساختمان‌داده‌ها را مشخص می‌کند. برای مدیریت قابلیت‌های پرس و جوی محدود شده منابع داده، مانلِسکو اپراتور پیوند وابسته را معرفی کرد است که در حقیقت نامتقارن است (Dökeroğlu, Coşar,2012).

به هر حال سیستم‌های مدیریت پایگاه‌داده، نیازمند بکارگیری تکنیک‌های بهینه‌سازی پرس‌وجو با هزینه

[1]Well-Studied