STUDIA INFORMATICA 2011
Volume 32 Number 2A (96)

Daniel KOSTRZEWA, Henryk JOSIŃSKI
Politechnika Śląska, Instytut Informatyki

# OCENA JAKOŚCI STRATEGII EKSPLORACJI PRZESTRZENI POSZUKIWAŃ DLA PROBLEMU OKREŚLENIA KOLEJNOŚCI REALIZACJI ZŁĄCZEŃ

**Streszczenie**. Artykuł stanowi próbę oceny jakości autorskiej strategii eksploracji przestrzeni poszukiwań dla problemu określenia kolejności realizacji złączeń w zapytaniu adresowanym do bazy danych. Jakość strategii zostanie oceniona przez porównanie czasów wykonania zapytania, uzyskanych w systemie SQL Server 2008: z jednej strony przez realizację planu wygenerowanego za pomocą algorytmu IWO, z drugiej zaś – przez realizację planu wyznaczonego przez moduł optymalizacyjny systemu zarządzania bazą danych.

**Słowa kluczowe**: eksploracja przestrzeni poszukiwań, określenie kolejności realizacji złączeń, algorytm IWO

# VERIFICATION OF THE SEARCH SPACE EXPLORATION STRATEGY BASED ON THE SOLUTIONS OF THE JOIN ORDERING PROBLEM

**Summary**. The paper addresses the problem of quality estimation of the search space exploration strategy. The strategy is used to find a satisfying solution to the join ordering problem, which constitutes a crucial part of the database query optimization task. The method of strategy verification is based on the comparison of the execution time for the solution produced by the IWO algorithm with the execution time for the solution determined by the SQL Server 2008 optimizer.

**Keywords**: exploration of the search space, join ordering, the IWO algorithm

## 1. Wstęp

Wyznaczenie porządku realizacji złączeń w zapytaniu adresowanym do bazy danych stanowi bardzo istotną część procesu optymalizacji zapytania, zarówno dla przypadku danych scentralizowanych, jak i rozproszonych. Literatura związana z tym tematem jest bardzo bogata

– kolejność wykonania złączeń określają między innymi metody [12, 15, 16]: Minimum Selectivity, Top-Down, KBZ, Relational Difference Calculus oraz AB. W tym celu wykorzystywano również metaheurystyki kombinatoryczne: algorytm symulowanego wyżarzania [2, 12, 15, 16], algorytm iteracyjnego poprawiania [2, 12, 15, 16], algorytm z listą tabu [12, 13], a także metody hybrydowe [2, 7, 9, 12] i algorytmy genetyczne [12, 15, 16] (ze względu na ograniczenie miejsca zrezygnowano z umieszczenia w bibliografii innych pozycji).

Autorzy niniejszego artykułu przedstawili w pracach [3, 4, 5, 6] rezultaty zastosowania algorytmu ewolucyjnego oraz zmodyfikowanej wersji algorytmu IWO (ang. *Invasive Weed Optimization*), w której pierwotną strategię penetracji przestrzeni poszukiwań zastępowano wariantami hybrydowej metody autorskiej. W kolejnej fazie badań postanowiono porównać wyniki metody autorskiej z rezultatami generowanymi przez moduły optymalizatorów komercyjnych systemów zarządzania bazami danych. Niniejszy artykuł służy prezentacji rezultatów eksperymentów przeprowadzonych dla danych scentralizowanych z wykorzystaniem systemu SQL Server 2008.

Przedstawienie form reprezentacji pojedynczego rozwiązania rozważanego problemu optymalizacyjnego stanowi treść punktu 2. Punkt 3. będzie poświęcony opisowi algorytmu IWO, ze szczególnym uwzględnieniem zaproponowanej hybrydowej metody penetracji przestrzeni poszukiwań. Sposób przeprowadzenia eksperymentów oraz ich rezultaty zostaną omówione w punkcie 4. Wnioski sformułowane w punkcie 5. tworzyć będą formę podsumowania.

## 2. Analiza problemu optymalizacyjnego

Poszukiwany porządek realizacji złączeń (w przypadku danych rozproszonych uzupełniony o wskazanie miejsc ich realizacji) ma zapewnić, że wynik zapytania dotrze do użytkownika w jak najkrótszym czasie. Funkcję celu, dla rozważanego problemu, stanowi szacunkowy koszt realizacji ciągu złączeń. Definiujący go model obliczeniowy opisano w pracach [4, 5]. Zawiera on formuły umożliwiające oszacowanie liczebności zbioru wynikowego operacji złączenia i rozmiaru rekordu w zbiorze wynikowym oraz kosztu operacji złączenia przy zastosowaniu algorytmu *pętli zagnieżdżonych* (ang. *nested loops join*) [11] i kosztu operacji przesyłu danych (w przypadku danych rozproszonych).

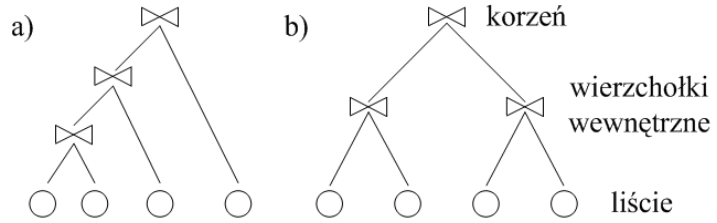
### 2.1. Reprezentacja pojedynczego rozwiązania

Przestrzeń poszukiwań tworzą porządki (plany) realizacji zadanego ciągu złączeń, które często przedstawiane są w postaci *drzew realizacji złączeń* (ang. *join processing tree*) [15, 16]. W drzewie realizacji złączeń wyróżnia się następujące elementy (rys. 1):

- *liście* przedstawiające zbiory rekordów,
- *wierzchołki wewnętrzne*, które symbolizują operacje złączenia; wierzchołek nazywany *korzeniem* jest elementem, któremu przypisano złączenie realizowane jako ostatnie; drzewo o *n* liściach ma *n*-1 wierzchołków wewnętrznych,
- *krawędzie* łączące każdy wierzchołek wewnętrzny z dwoma niżej położonymi elementami drzewa (wierzchołkami wewnętrznymi lub liśćmi); każda taka trójka elementów wyraża pojedynczą operację złączenia i jej argumenty. Krawędzie ilustrują przepływ danych od poziomu liści aż do korzenia.

Układ krawędzi, wierzchołków wewnętrznych i liści decyduje o kształcie drzewa. Jeśli do każdego wierzchołka wewnętrznego prowadzi co najmniej jedna krawędź wychodząca z liścia, to drzewo charakteryzuje się *sekwencyjnym układem wierzchołków* (rys. 1a). Wystąpienie co najmniej jednego wierzchołka, do którego prowadzą krawędzie wychodzące z wierzchołków wewnętrznych, a nie z liści, oznacza możliwość równoczesnej realizacji złączeń, reprezentowanych przez te dwa wierzchołki. Drzewo o takiej własności nosi nazwę drzewa *o równoległym układzie wierzchołków* (rys. 1b).

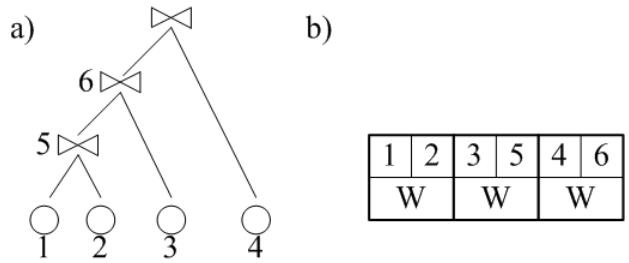
Z przeprowadzonych eksperymentów wynika, że w przypadku systemu SQL Server 2008 pod uwagę brane są wyłącznie drzewa o sekwencyjnym układzie wierzchołków; ich zatem będą dotyczyć dalsze rozważania.




Rys. 1. Kształt drzewa realizacji złączeń o układzie wierzchołków: a) sekwencyjnym, b) równoległym
Fig. 1. Shape of a join processing tree: a) sequential, b) parallel

## 2.2. Odwzorowanie drzewa realizacji złączeń w osobnika populacji chwastów

W algorytmie IWO pojedyncze rozwiązanie problemu optymalizacyjnego jest zapisane w osobniku populacji chwastów. Sposób odwzorowania drzewa realizacji złączeń w osobnika przedstawiono na rys. 2.




Rys. 2. Reprezentacje pojedynczego rozwiązania: a) drzewo realizacji złączeń, b) osobnik
Fig. 2. Representations of a single solution: a) a join processing tree, b) an individual

Należy zauważyć, że w zaproponowanej postaci osobnika uwzględniono możliwość reprezentacji drzewa realizacji złączeń dla danych rozproszonych. Pojedyncza operacja złączenia jest bowiem zapisana jako trójka, nazywana *genem*, na którą składają się numery zbiorów rekordów, stanowiące argumenty złączenia oraz numer węzła sieci (stacji roboczej), w której dojdzie do realizacji złączenia (na rys. 2b oznaczony symbolem *W*). Układ genów odpowiada kolejności wykonywania poszczególnych złączeń. Zbiory rekordów, wymienionych w zapytaniu, otrzymały numery od 1 do 4, natomiast zbiorom będącym wynikami kolejno realizowanych złączeń przypisywane są kolejne liczby naturalne.

Penetracja przestrzeni poszukiwań dokonywana jest przez transformacje przekształcające osobnika nazywanego macierzystym w osobnika potomnego. Dla przyjętej postaci osobnika i wobec ograniczenia do danych scentralizowanych transformacja polega na zamianie dwóch wylosowanych numerów zbiorów, z których każdy należy do innego z pary wylosowanych genów. Dzięki zamianie numerów zbiorów unika się niedopuszczalnego, wielokrotnego wystąpienia tego samego numeru zbioru, jako argumentu różnych złączeń.

Warto zaznaczyć, że omówiona postać osobnika może również reprezentować drzewa realizacji złączeń o równoległym układzie wierzchołków.

## 3. Opis algorytmu IWO

Algorytm IWO daje możliwość eksperymentowania z różnymi strategiami penetracji przestrzeni poszukiwań. W podanym pseudokodzie, w którym zastosowano konwencję terminologiczną, zgodną z „przyrodniczą” inspiracją idei algorytmu, za wyznaczenie kolejnego badanego punktu przestrzeni odpowiada wyróżniony fragment:

```
Utwórz pierwszą populację złożoną z n losowo wygenerowanych osobników.
Dla każdego osobnika:
    Oblicz wartość funkcji przystosowania.
Dopóki nie zostanie spełnione kryterium stopu algorytmu:
    Dla każdego osobnika należącego do populacji:
        Na podstawie wartości funkcji przystosowania wyznacz liczbę ziaren.
        Dla każdego ziarna:
            Określ miejsce upadku wybierając z określonym prawdopodobieństwem jedną z metod: rozwiewanie, rozsiewanie lub staczanie.
            Skonstruuj nowego osobnika zgodnie ze specyfiką wylosowanej metody.
            Oblicz wartość funkcji przystosowania nowego osobnika.
    Utwórz nową populację złożoną z n najlepiej przystosowanych osobników
    uwzględniając przy wyborze zarówno poprzednią populację, jak i nowe osobniki.
```

Charakter działania opisanego jako „Określ miejsce upadku” różni się w zależności od metody, którą wylosowano dla jego przeprowadzenia. Wartości prawdopodobieństwa wylosowania, przypisane poszczególnym metodom: rozwiewaniu, rozsiewaniu i staczaniu, stanowią parametry algorytmu.

W przypadku *rozwiewania* działanie to ma na celu obliczenie odległości miejsca upadku ziarna od jego rośliny macierzystej. Odległość ta opisana jest rozkładem *t*-Studenta o określonej liczbie stopni swobody [1], obciętym do wartości nieujemnych, przy czym wartość otrzymana z generatora liczb pseudolosowych o rozkładzie *t*-Studenta przemnażana jest przez współczynnik skalujący $\gamma_{iter}$. Współczynnik skalujący zmniejszany jest wraz z każdą kolejną iteracją algorytmu (a zatem dla każdej kolejnej populacji), zgodnie z następującą formułą:

$$\gamma_{iter} = \left( \frac{iter_{\max} - iter}{iter_{\max}} \right)^{m} \left( \gamma_{init} - \gamma_{fin} \right) + \gamma_{fin} \tag{1}$$

Symbole $\gamma_{init}$, $\gamma_{fin}$ reprezentują, odpowiednio, wartości początkową i końcową współczynnika skalującego, natomiast $m$ jest współczynnikiem modulacji nieliniowej. Aktualnie realizowana iteracja nosi numer *iter*, a łączna liczba iteracji ($iter_{max}$), równoważna łącznej liczbie populacji, może zostać zadana jako kryterium stopu algorytmu lub – jeśli kryterium stopu jest czas obliczeń – może zostać oszacowana na tej podstawie. Wynikająca z formuły (1) tendencja do stopniowego zmniejszania odległości dla kolejnych populacji zgodna jest z intencją autorów oryginalnej wersji algorytmu IWO [8, 10, 14].

Konstrukcja nowego osobnika w metodzie rozwiewania polega na wykonaniu na kopii osobnika macierzystego transformacji w liczbie równej odległości między osobnikami nowym i macierzystym.

*Rozsiewanie* to losowe rozrzucanie ziaren po całej przestrzeni. Działanie to sprowadza się zatem do utworzenia nowego osobnika w sposób losowy.

*Staczanie* opiera się na badaniu sąsiedztwa rośliny macierzystej, tworzonego przez osobniki, które różnią się od niej dokładnie jedną transformacją. Spośród określonej liczby tak zdefiniowanych sąsiadów wybiera się osobnika najlepiej przystosowanego, po czym przeprowadza się analizę jego sąsiedztwa, ponownie zmierzając do wyboru osobnika najlepiej przystosowanego. Tę procedurę powtarza się *k*-krotnie (*k* stanowi parametr metody). Nowy osobnik zostanie wyznaczony w ostatniej iteracji, jako osobnik najlepiej przystosowany. Wartość funkcji przystosowania obliczana jest jako odwrotność funkcji celu.

## 4. Badania eksperymentalne

Celem przeprowadzonych eksperymentów było porównanie czasów wykonania zapytania, uzyskanych w systemie SQL Server 2008: z jednej strony przez realizację planu skonstruowanego za pomocą algorytmu IWO, z drugiej zaś – przez realizację planu wyznaczonego przez moduł optymalizacyjny systemu zarządzania bazą danych. Wykonanie planu utworzonego przez algorytm IWO było możliwe dzięki zestawowi *wskazówek* (ang. *hints*) pozwalających użytkownikowi na wymuszenie określonych zachowań modułu optymalizacyjnego.

Wzorując się na pracy [13] skonstruowano generator zapytań testowych, a następnie, posługując się środowiskiem eksperymentalnym, opisanym w pracach [3, 4, 5], przeprowadzono serię prób, z których każda składała się z następujących etapów:

1. Generacja zapytania testowego odwołującego się do 50 tabel pojedynczej, scentralizowanej bazy danych.
2. Określenie planu realizacji zapytania testowego przy użyciu algorytmu IWO.
   Przyjęto, że minimalny czas obliczeń wynosi 5 [s], a chwilę ich zakończenia wyznacza utworzenie pierwszej populacji, którą skonstruowano po upływie tego czasu. W ten sposób uzyskano w przybliżeniu stały czas obliczeń $t_{IWO}^{opt}$. Obliczenia powtarzano 10-krotnie, otrzymując różne plany, co wynika z niedeterministycznego charakteru algorytmu IWO.
3. Wymuszenie realizacji zapytania testowego w systemie SQL Server 2008 według planu wyznaczonego w punkcie 2. i określenie jego kosztu $c_{IWO}^{opt}$.
   Zastosowanie wskazówki OPTION(FORCE ORDER) [17] pozwoliło na narzucenie takiego porządku złączeń, jaki został wyznaczony przez algorytm IWO. Ponadto, dzięki wskazówce OPTION(LOOP JOIN) wymuszono realizację złączeń zgodnie z algorytmem pętli zagnieżdżonych, czyli tym, którego użycie zakłada algorytm IWO. Ponieważ dla każdego wariantu planu otrzymanego dla danego zapytania w punkcie 2. wykonywano 10 uruchomień, rejestrowano w ten sposób 100 pomiarów czasu realizacji $t_{IWO}^{proc}$.
4. Określenie planu realizacji zapytania testowego oraz jego kosztu $c_{DBMS}^{opt}$ za pomocą metody wbudowanej w systemie SQL Server 2008, połączone z realizacją zapytania.
   10-krotne przeprowadzenie tej operacji pozwalało na zebranie serii pomiarów czasu realizacji $t_{DBMS}^{proc}$.

Należy zaznaczyć, że dla uniknięcia wpływu wcześniej wykonanych operacji na rezultat kolejnej każdorazowo opróżniano bufory pamięci cache wykonując zestaw poleceń:

```
CHECKPOINT
DBCC DROPCLEANBUFFERS.
```

Tabela 1

Wartości parametrów algorytmu

| | |
|---|---|
| Liczebność populacji | 100 |
| Maksymalna liczba ziaren | 2 |
| Liczba *k* sąsiedztw badanych podczas staczania | 3 |
| Prawdopodobieństwo rozwiewania | 0.2 |
| Prawdopodobieństwo rozsiewania | 0.75 |
| Prawdopodobieństwo staczania | 0.05 |
| Liczba stopni swobody | 2 |
| Współczynnik modulacji nieliniowej *m* | 3 |
| Początkowa wartość współczynnika skalującego $\gamma_{init}$ | 5 |
| Końcowa wartość współczynnika skalującego $\gamma_{fin}$ | 1 |

Zapytania zwracały od ok. 250 tysięcy do niemal miliona rekordów.

Eksperymenty wykonano posługując się stacją roboczą o następujących parametrach: procesor Intel Core2 Quad Q6600, 2.4GHz, pamięć RAM 2GB 800MHz. Zastosowane wartości parametrów algorytmu IWO zebrano w tabeli 1. Wartości te zostały ustalone w wyniku badań opisanych w pracy [6], jako najodpowiedniejsze dla rozważanego problemu.

Wyniki eksperymentów zebrano w tabeli 2. Dla każdego zapytania podano kolejno: koszt planu skonstruowanego przez system SQL Server 2008, wartość średnią oraz odchylenie ćwiartkowe dla kosztu 10 wariantów planów skonstruowanych przez algorytm IWO, średni czas 10 wykonań planu skonstruowanego przez system SQL Server 2008 oraz średni czas 100 wykonań wszystkich wariantów planu skonstruowanego przez algorytm IWO wraz z klasycznym współczynnikiem zmienności. Kolumna odchylenia ćwiartkowego (miary poziomu zróżnicowania badanej wielkości po odrzuceniu 25% obserwacji o wartościach najmniejszych i 25% obserwacji o wartościach największych) podkreśla duży rozstęp wartości kosztu planów konstruowanych przy użyciu algorytmu IWO. Z kolei kolumna klasycznego współczynnika zmienności „KWZ" (ilorazu odchylenia standardowego i wartości średniej) pokazuje, że rozpiętość kosztów nie przełożyła się w tak dużym stopniu na czas realizacji.

Tabela 2

Wartości kosztu i czasu realizacji planów w systemie SQL Server 2008

| Nr zapytania | Koszt planu DBMS $c_{DBMS}^{opt}$ | Koszt planu IWO $c_{IWO}^{opt}$ | | Średni czas realizacji planu DBMS $t_{DBMS}^{proc}$ [s] | Czas realizacji planu IWO $t_{IWO}^{proc}$ | |
|---|---|---|---|---|---|---|
| | | Średnia | Odch. ćw. | | Średnia [s] | KWZ [%] |
| 1 | 199 729 | 253 192 | 88 237 | 228.88 | 228.64 | 3.4 |
| 2 | 20 816.9 | 341 626 | 156 472 | 130.51 | 133.03 | 4.3 |
| 3 | 30 741.6 | 1 180 772 | 302 458 | 320.04 | 295.24 | 2.7 |
| 4 | 190 155 | 455 894 | 167 230 | 178.24 | 144.71 | 4.8 |
| 5 | 499 593 | 1 095 483 | 842 488 | 220.38 | 84.81 | 8.5 |

W przeprowadzonych eksperymentach zaobserwowano następujące przypadki:

- koszt planu skonstruowanego przez system SQL Server 2008 był znacznie niższy niż średni koszt planu wygenerowanego przez algorytm IWO, natomiast średnie czasy realizacji zapytania testowego, według każdego z obydwu planów, okazały się zbliżone (zapytania 1 i 2),
- koszt planu skonstruowanego przez system SQL Server 2008 był znacznie niższy niż średni koszt planu wygenerowanego przez algorytm IWO, natomiast relacja między średnimi czasami realizacji zapytania testowego, według każdego z obydwu planów, okazała się odwrotna i to zdecydowanie na korzyść algorytmu IWO (zapytania 3, 4 i 5).

Należy również wspomnieć o obserwacji poczynionej w trakcie analizy konstruowanych planów – rezultaty oszacowania liczebności zbiorów wynikowych poszczególnych złączeń

odbiegały znacząco od rzeczywistych liczebności, co z pewnością wpływało na wartość kosztu pojedynczego planu.

## 5. Podsumowanie

Interpretacja wyników przeprowadzonych eksperymentów wyznacza kierunek dalszych działań, które powinny zmierzać do ograniczenia rozrzutu kosztu planów generowanych za pomocą algorytmu IWO. Warto również podkreślić, że owa duża rozpiętość miała niewielkie przełożenie na czasy wykonania planów, które ponadto okazały się porównywalne, a w niektórych przypadkach znacznie krótsze od czasów realizacji planów konstruowanych przez system SQL Server 2008. Poszerzeniem możliwości porównawczych okazałoby się z pewnością uwzględnienie w modelu kosztu operacji złączenia algorytmu, opartego na funkcji mieszającej oraz metody złożonej z sortowania i scalania.

**BIBLIOGRAFIA**

1. Gajek L., Kałuszka M.: Wnioskowanie statystyczne. Modele i metody. Wydawnictwa Naukowo-Techniczne, Warszawa 1996.
2. Ioannidis Y. E., Kang Y. C.: Randomized Algorithms for Optimizing Large Join Queries. Proceedings of the ACM SIGMOD Conference on Management of Data, Atlantic City 1990.
3. Kostrzewa D., Josiński H.: Planowanie procesu scalania danych rozproszonych za pomocą algorytmu ewolucyjnego. Bazy danych. Rozwój metod i technologii. Tom 1: Architektura, metody formalne i zaawansowana analiza danych. Wydawnictwa Komunikacji i Łączności, Gliwice 2008.
4. Kostrzewa D., Josiński H.: Zastosowanie algorytmu IWO do planowania procesu scalania danych rozproszonych. Studia Informatica, Vol. 30, No. 2A (83). Wydawnictwo Politechniki Śląskiej, Gliwice 2009.
5. Kostrzewa D., Josiński H.: The Comparison of an Adapted Evolutionary Algorithm with the Invasive Weed Optimization Algorithm Based on the Problem of Predetermining the Progress of Distributed Data Merging Process. Man-Machine Interactions, Advances in Intelligent and Soft Computing, Springer-Verlag, 2009.
6. Kostrzewa D., Josiński H.: Metody przeszukiwania przestrzeni planów realizacji zapytań za pomocą algorytmu IWO. Studia Informatica, Vol. 31, No. 2A (89). Wydawnictwo Politechniki Śląskiej, Gliwice 2010.

7. Lanzelotte R. S. G., Valduriez P., Zaït M.: On the Effectiveness of Optimization Search Strategies for Parallel Execution Spaces. Proceedings of the 19th VLDB Conference, Dublin 1993.
8. Mallahzadeh A. R., Oraizi H., Davoodi-Rad Z.: Application of the Invasive Weed Optimization Technique for Antenna Configurations. Progress in Electromagnetics Research, 2008.
9. Mamaghani A. S., Asghari K., Mahmoudi F., Meybodi M. R.: A Novel Hybrid Algorithm for Join Ordering Problem in Database Queries. Proceedings of the 6th WSEAS Int. Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, Tenerife 2007.
10. Mehrabian R., Lucas C.: A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics, Vol. 1, Issue 4, 2006.
11. Mishra P., Eich M. H.: Join Processing in Relational Databases. ACM Computing Surveys, Vol. 24, No. 1, 1992.
12. Moerkotte G.: Building Query Compilers. http://pi3.informatik.uni-mannheim.de/~moer/querycompiler.pdf (draft), 2009.
13. Morzy T., Matysiak M., Salza S.: Tabu Search Optimization of Large Join Queries. Proceedings of the 4th International Conference on Extending Database Technology. (eds.): Jarke M. et al., Cambridge 1994.
14. Sepehri Rad H., Lucas C.: A Recommender System based on Invasive Weed Optimization Algorithm. IEEE Congress on Evolutionary Computation, Singapore 2007.
15. Steinbrunn M., Moerkotte G., Kemper A.: Heuristic and Randomized Optimization for the Join Ordering Problem. The VLDB Journal, Vol. 6, No. 3, 1997.
16. Steinbrunn M., Moerkotte G., Kemper A.: Optimizing Join Orders. Technical Report IP9307, Faculty of Mathematics, University of Passau 1993.
17. Tow D.: SQL. Optymalizacja. Wydawnictwo Helion, Gliwice 2004.

Recenzenci: Dr hab. inż. Andrzej Kwiecień, prof. Pol. Śląskiej
Prof. dr hab. inż. Tadeusz Morzy

Wpłynęło do Redakcji 31 stycznia 2011 r.

**Abstract**

The paper addresses the problem of quality estimation of the search space exploration strategy. The strategy is used to find a satisfying solution to the join ordering problem, which

constitutes a crucial part of the database query optimization task. The method of strategy verification is based on the comparison of the execution time for the solution produced by the IWO algorithm with the execution time for the solution determined by the SQL Server 2008 optimizer. The authors discuss representations of the single solution and describe the modified version of the IWO algorithm emphasizing features of the proposed hybrid method of the search space exploration. The results of the conducted experiments form the main topic of analysis.

**Adresy**

Daniel KOSTRZEWA: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, Daniel.Kostrzewa@polsl.pl.
Henryk JOSIŃSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, Henryk.Josinski@polsl.pl.