POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

TECHNOLOGIE INFORMATYCZNE W ZARZĄDZANIU SYSTEMY WSPOMAGANIA DECYZJI

pod redakcją:

Jana Studzińskiego,

Ludosława Drelichowskiego,

Olgierda Hryniewicza,

Janusza Kacprzyka

TECHNOLOGIE INFORMATYCZNE W ZARZĄDZANIU
SYSTEMY WSPOMAGANIA DECYZJI

Polska Akademia Nauk • Instytut Badań Systemowych

Seria: BADANIA SYSTEMOWE
tom 26

Redaktor naukowy:

Prof. dr hab. Jakub Gutenbaum

Warszawa 2000

TECHNOLOGIE INFORMATYCZNE W ZARZĄDZANIU SYSTEMY WSPOMAGANIA DECYZJI

pod redakcją
Jana Studzińskiego, Ludosława Drelichowskiego
Olgierda Hryniewicza i Janusza Kacprzyka

Książka zawiera wybór referatów przedstawionych na konferencji "Komputerowe systemy wielodostepne KSW'2000" w Ciechocinku w 2000 r. Konferencja pod patronatem Komitetu Badań Naukowych została zorganizowana przez Akademię Techniczno-Rolniczą w Bydgoszczy, Instytut Badań Systemowych PAN, Komisję Informatyki PAN - Oddział w Gdańsku oraz Bydgoskie Zakłady Elektromechaniczne "BELAM" S.A. w Bydgoszczy.

ISBN 83-85847-53-7
ISSN 0208-8028

Rozdział 4

Metody i algorytmy obliczeniowe w systemach komputerowych

ALGORYTMY GENETYCZNE W EWOLUCJI STRUKTUR BAZY DANYCH

Barbara Królikowska, Jerzy Marcinkiewicz
Uniwersytet Szczeciński
bkrol@uoo.univ.szczecin.pl, jmarcin@uoo.univ.szczecin.pl

Jednym z istotnych problemów projektowania baz danych jest definiowanie jej struktury fizycznej. Problem nabiera znaczenia wraz ze zwiększaniem się rozmiarów użytkowanych baz danych oraz wzrastającą zmiennością wymagań informacyjnych użytkowników baz danych. Dotychczasowe metody projektowania traktują problem definiowania struktury fizycznej w sposób marginalny. Artykuł przedstawia propozycję wykorzystania algorytmu genetycznego w definiowaniu optymalnej struktury fizycznej bazy danych. Przedstawiono sposób prezentacji problemu jako zestawu chromosomów, funkcję celu i funkcję dostosowania. W podsumowaniu wskazano na możliwości wykorzystania algorytmu symulowanego wyżarzania w definiowaniu optymalnego zestawu indeksów bazy

1. Problem projektowania i ewolucji struktur baz danych

Dotychczas rozwijane i stosowane metody projektowania baz danych koncentrują się na modelowaniu i projektowaniu struktur logicznych baz danych. Metody te często pomijają lub traktują marginalnie projektowanie struktur fizycznych baz danych. Przyczyny tego stanu rzeczy mogą być następujące:

- Struktura fizyczna bazy danych uzależniona jest od konkretnego systemu zarządzania bazą danych (SZBD) – trudno jest więc zdefiniować wyczerpujący zestaw zasad projektowania struktury fizycznej, właściwy dla SZBD różnych producentów, stosujących różne modele logiczne bazy danych.
- Problemy optymalizacji struktury fizycznej przejęło na siebie w znacznym stopniu oprogramowanie bazy danych.

Z drugiej strony większość SZBD jest konstruowana w oparciu o standardowe modele baz danych (relacyjny i obiektowy), przy czym dominujący udział w rynku mają SZBD wykorzystujące model relacyjny. Wyko-

rzystują one w znacznym stopniu ujednolicone struktury fizyczne danych. Istnieje więc możliwość sformułowania zasad i reguł definiowania i optymalizacji struktur fizycznych bazy.

Obserwuje się również zjawisko intensywnego zwiększania się rozmiarów eksploatowanych baz danych. Właściwa definicja struktury fizycznej staje się warunkiem koniecznym dla zapewnienia szybkiego dostępu (wyszukiwania) do danych. Przejawem tego zjawiska jest proponowanie przez producentów baz danych coraz bardziej wyrafinowanych form indeksowania i fizycznej organizacji tablic danych (White, 1999, Wrembel, 1997, Ooi, 1998). Ma to miejsce szczególnie w przypadku SZBD konstruowanych dla potrzeb hurtowni danych. Dotychczas administratorzy baz danych nie dysponują efektywnymi metodami i narzędziami definiowania coraz bardziej złożonych struktur fizycznych baz danych.

Innym zjawiskiem dotyczącym organizacji fizycznej baz danych jest narastająca zmienność wymagań (potrzeb) informacyjnych użytkowników baz danych. Zapewnienie właściwego poziomu gotowości informacyjnej bazy danych wymaga:

- nieustannej *modyfikacji struktur fizycznych,* – co zapewnić może satysfakcjonujące czasy wykonywania operacji na bazie danych,
- ciągłej *modyfikacji struktury logicznej* bazy danych, – co zapewniać może zaspakajanie zmieniających się potrzeb informacyjnych użytkowników bazy danych.

W efekcie można wyodrębnić dwa poziomy definiowania (optymalizacji) struktur fizycznych baz danych:

- w trakcie projektowania nowej bazy danych,
- w trakcie nieustannej modyfikacji struktur logicznych i fizycznych bazy danych.

Nieustanne rozwiązywanie problemu doboru (definicji) struktury fizycznej w odniesieniu do zmieniających się potrzeb informacyjnych użytkowników – tak w zakresie nowych danych jak i zmian kierunków zapotrzebowania na dane już występujące w bazie – wymaga wyposażenia administratora bazy danych w odpowiednie metody i narzędzia definiowania struktury fizycznej bazy danych.

Ze względu na liczbę czynników wpływających na efektywność struktury fizycznej bazy, ocenianą szybkością dostępu do danych zawartych w bazie, jak również zróżnicowane rozwiązania implementowane przez poszczególnych producentów SZBD, trudne jest wyznaczenie jednoznacznych procedur projektowania efektywnej struktury fizycznej bazy danych. Można tu raczej proponować iteracyjne poprawianie rozwiązania, w zależności od uzyskiwanych rezultatów funkcjonowania danej wersji struktury fizycznej.

Optymalna struktura będzie więc wynikiem nieustannego poszukiwania rozwiązania, a nie efektem jednorazowego działania projektowego.

Wydaje się, że skuteczną metodą dochodzenia do optymalnego rozwiązania w procesie definiowania struktury fizycznej może być technika algorytmów genetycznych.

2. Istota algorytmów genetycznych

Algorytm genetyczny jest jedną z najbardziej obiecujących strategii poszukiwania rozwiązania w zbiorze potencjalnych rozwiązań (Cytowski, 1996).

Konstrukcja algorytmu genetycznego wymaga przede wszystkim określenia *populacji początkowej* – zestawu chromosomów. Każdy z chromosomów reprezentuje jeden element zbioru rozwiązań. Gen jest atrybutem chromosomu. Klasycznie generuje się taką populację metodą losową np.: 10 chromosomów o długości 5 genów. Każdy gen otrzymuje losowo (rzut monetą) wartość *0*-reszka lub 1-orzeł (Gwiazda, 1995).

W proponowanym rozwiązaniu, przy znanym modelu logicznym bazy, do wyznaczenia chromosomów populacji początkowej można zastosować metodę deterministyczną, przyjmując binarną postać genów. Dodatkowo należy określić pozycję genu – *locus* (Goldberg, 1995) . Gen o numerze 1 w każdym chromosomie oznaczać może przykładowo klucz własny relacji w modelu relacyjnym bazy danych. *Fenotyp* (Goldberg, 1995) *w* postaci systemu kodów zawierać może parametry modelu logicznego bazy (relacje, klucze obce, indeksy).

W kolejnym kroku należy określić *funkcję przystosowania*, która będzie miarą dopasowania danego chromosomu do najlepszego rozwiązania. Określa ona, w jakim stopniu chromosom przyczynia się do rozwiązania danego problemu. Powszechnie stosuje się następujące przekształcenie kosztu w przystosowanie (Goldberg, 1995):

$$f(x) = \begin{cases} C_{max} - g(x) & \text{jeżeli } g(x) < C_{max} \\ 0 & \text{jeżeli } g(x) >= C_{max} \end{cases} \qquad (1)$$

gdzie C_{max} - max wartość funkcji g(x), która jest funkcją celu dla danego zagadnienia.

W procesie ulepszania rozwiązania stosuje się odpowiednie operatory genetyczne. Operator reprodukcji generuje następną populację. Chromosomy w tej populacji zawierają posiadają geny, których wartość jest wyznaczana losowo. Operator *krzyżowania* wymienia geny między chromosomami.

W kolejnym kroku można zastosować proces *mutacji,* który pozwala zmienić gen w wybranym chromosomie. Proces mutacji stosuje się w celu zwiększenia dopasowania chromosomu. Wybór chromosomów do procesu reprodukcji realizowany jest losowo metodą ruletki (Gwiazda, 1995) - każdemu przysługuje sektor ruletki proporcjonalnie do funkcji przystosowania.

Strukturę klasycznego algorytmu genetycznego prezentuje rys. 1.

3. Wykorzystanie algorytmu genetycznego do modelowania struktury fizycznej bazy danych.

Punktem wyjścia do definiowania struktury fizycznej bazy danych są:

- model logiczny bazy danych,
- oraz rozwiązania i ograniczenia rozwiązań narzucane przez wykorzystywany SZBD.

Do dalszych rozważań przyjmujemy założenia, że struktura logiczna bazy jest reprezentowana przez model relacyjny bazy danych uzyskany z przekształcenia modelu konceptualnego (model związków encji)[1]. Poszukuje się struktury fizycznej bazy danych o relacyjnym modelu logicznym.

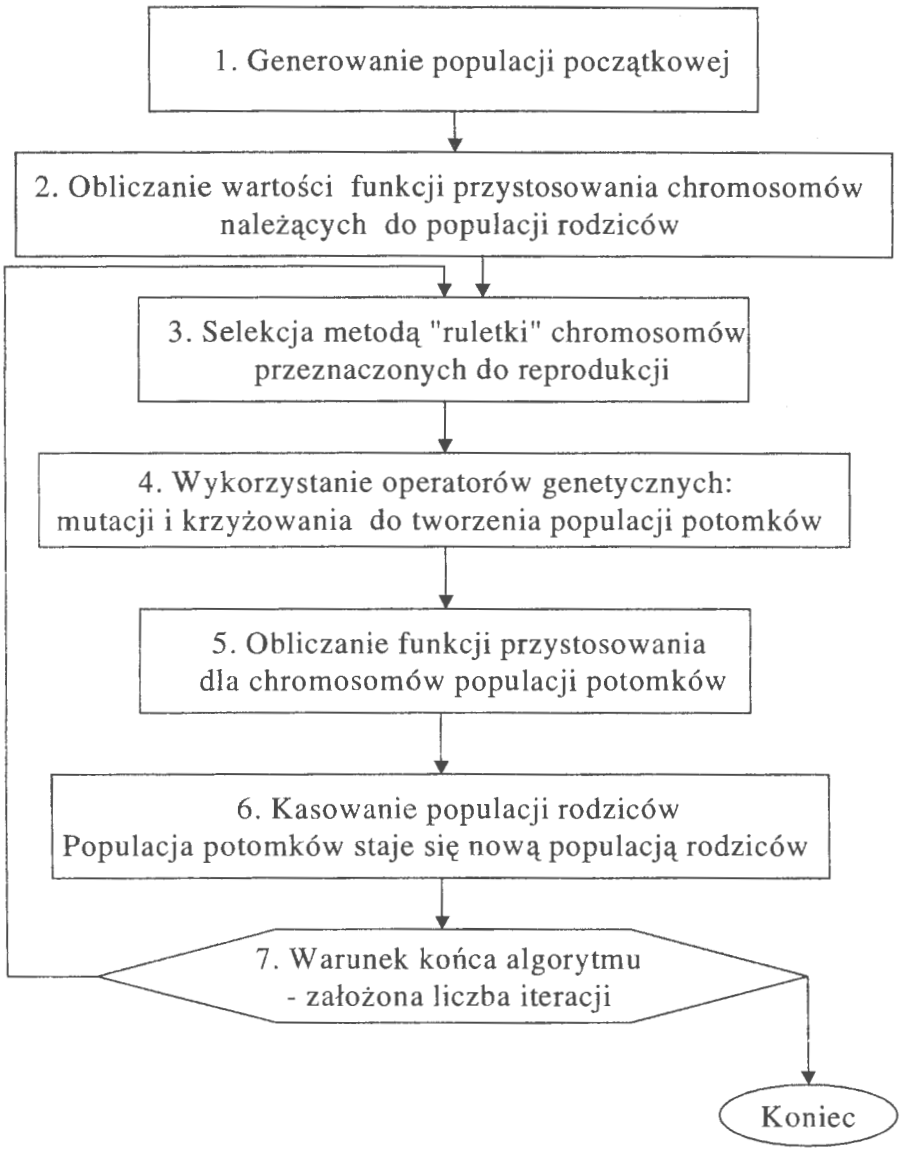
Celem wykorzystania algorytmu genetycznego jest poszukiwanie efektywnej struktury fizycznej, dysponując wyjściowym modelem logicznym bazy danych.

Parametrami poszukiwanego rozwiązania są:

- definicja tablic występujących w bazie – proces poszukiwania rozwiązania może prowadzić do łączenia lub dzielenia tablic – w stosunku do rozwiązania pierwotnego,
- rozkład atrybutów w poszczególnych tablicach – co determinuje szybkość wyszukiwania i przetwarzania danych,
- definicja indeksów i ich rodzajów dla poszczególnych tablic,
- wyznaczanie klastrów bazy danych.

---

[1] W niniejszym artykule celowo rezygnujemy z problemu definiowania struktury fizycznej dla obiektowych baz danych. Ze względu na odmienność stosowanych tam rozwiązań, zagadnienie to powinno być przedmiotem oddzielnego opracowania.



Rys. 1 Schemat klasycznego algorytmu genetycznego

Przy szukaniu lepszych rozwiązań w problemach informatycznych stosuje się często strategie przeszukiwania. Zastosowanie strategii przeszukiwania, których przykładem jest algorytm genetyczny, umożliwia analizę elementów zbioru potencjalnych rozwiązań w celu wyznaczenia tych elementów, które spełniają ograniczenia modelu. Strategie przeszukiwania wymagają opisu zadania wg następujących elementów (Cytowski, 1996):

- reprezentacja każdego z elementów przestrzeni przeszukiwania (kod elementu),
- metody obliczeniowe pozwalające wygenerować kod kolejnego elementu na podstawie kodu danego elementu,
- Metody wyboru operatorów spośród operatorów możliwych do zastosowania - strategie sterowania.

Kod elementu powinien uwzględniać strukturę zadania oraz cechy indywidualne elementu. W procesie projektowania struktury fizycznej bazy danych trudno określić metody obliczeniowe dla generowania kolejnych elementów struktury. Struktura fizyczna bazy danych, jest środowiskiem, które pozwala określić zbiór potencjalnych rozwiązań, jednak zdefiniowanie precyzyjnego algorytmu działania jest skomplikowane a wręcz niemożliwe. Podstawą projektowania jest funkcja, która określa jedynie ograniczenia dla elementów struktury – atrybutów i tablic.

Generacja populacji początkowej wymaga zdefiniowania chromosomów na podstawie znanego modelu logicznego bazy danych. W przedstawianym problemie przyjmuje się, że chromosom opisuje atrybut bazy danych (ilość chromosomów równa się ilości atrybutów w bazie danych). Wszystkie chromosomy są tej samej długości. Geny tworzą bitowy zapis cech analizowanego atrybutu. Praktycznie problem należy sprowadzić do zdefiniowania cech atrybutów, a następnie przypisać każdej z nich wartość 1-jeżeli cecha występuje, lub 0 - jeżeli nie występuje. Zbiór cech atrybutów zawiera następujące dane:

- przynależność do tablicy,
- klucz własny,
- klucz obcy,
- indeks i jego rodzaj,
- obligatoryjność wystąpienia
- rodzaje związków z innym atrybutami,
- przynależność do klastrów.

W wyniku zastosowania procesu reprodukcji, uzyskuje się kolejne populacje opisujące atrybuty, różniące się między sobą wartościami genów.

Ze względu na fakt, że szybkość uzyskiwania danych z bazy jest podstawowym kryterium oceny efektywności jej struktury fizycznej, definicję funkcji celu oparto na kryteriach oceny stosowanych w procesie wyboru strategii wykonywania zapytań na relacyjnych bazach danych (Delobel, 1989).

Jako funkcję celu optymalizacji struktury fizycznej bazy danych, przyjmuje się minimalizację kosztu wykonania kompleksowego zapytania na bazie danych. Koszt ten obejmuje przede wszystkim czas wykonywania operacji wejścia-wyjścia oraz zajętość zasobów pamięciowych komputera.

Podstawą definicji funkcji celu jest fikcyjne, kompleksowe pytanie wykorzystujące złączenia pomiędzy wszystkimi tablicami oraz wykonujące operacje selekcji na wszystkich tablicach. Definicja funkcji jest przedstawiona poniżej:

$$KD = Lstron + Lkrotek \tag{2}$$

gdzie:

Lstron – wskaźnik liczby operacji wejścia-wyjścia, mierzony liczbą stron zajmowanych przez tablice bazy,

Lkrotek – przeciętna liczba krotek przetwarzana w trakcie realizacji pytania testowego.

$$Lstron = \sum_{i=1}^{n} \frac{(Kard\ (R_i) * Latr\ (R_i))}{Wstron} \tag{3}$$

gdzie: $Kard(R_i)$ - liczba kardynalna krotek w *i-ej* tablicy,

$Latr(R_i)$ - liczba atrybutów w *i-ej* tablicy,

Wstron – Wskaźnik wielkości strony w danej bazie danych

Z kolei *Lkrotek= LKzl+Lksel+Efektind,* (4)

gdzie:

LKzl – szacunkowa liczba krotek w tablicach powstających ze złączania tablic głównych i pośrednich w operacjach złączania,

Lksel – Liczba krotek przetwarzanych w operacjach selekcji, łącznie z przetwarzanymi indeksami, przy czym dla krotek indeksów ze względu na ich minimalną wielkość, przyjmuje się przelicznik 1/3,

Efektind – efekt zastosowania indeksów

$$LKzl = \sum_{z=1,i=3}^{n-1} \frac{Kard(Z_{z-1}) + Kard(R_{i+1}) + \left|Kard(Z_z) - Kard(R_{i+1})\right|}{2} - \left|Kard(Z_z) - \frac{1}{3} Kard(R_{i+1})\right|, \quad gdzie \rightarrow Z_o = R_1\ gdy \quad z = 0 \tag{5}$$

$$LKsel = \sum_{i=1}^{n} Kard(R_i) + \sum_{j=1}^{k} Kard(Indj) \quad (6)$$

$$Efektind = \sum_{i=1}^{n} \frac{Kard(R_i)}{3 * Latr(R_i)} \quad (7)$$

Wartość maksymalna funkcji celu równa się zero (zerowy koszt wykonania zapytania).

Przy definiowaniu funkcji celu przyjęto następujące założenia:

- zakłada się typową postać schematu relacyjnej bazy danych, składającego się z tablic głównych, powiązanych z innymi tablicami co najmniej jednym powiązaniem,
- dla każdej tablicy zdefiniowano indeksy, przy czym założono że tablica może posiadać zdefiniowane indeksy dla najwyżej 1/3 swoich atrybutów,
- w każdej operacji na bazie danych, jeżeli jest to tylko możliwe, wykorzystuje się istniejące indeksy,
- w trakcie wykonywania zapytań zawsze w pierwszej kolejności wykonywane są operacji selekcji na tablicach,
- operacje złączania są wykonywane po operacji selekcji.

Proces optymalizacji struktury fizycznej w oparciu o wyżej zdefiniowany algorytm genetyczny jest przeprowadzony eksperymentalnie na przykładzie bazy danych obsługi - studiów podyplomowych. Uzyskane rezultaty będą przedstawione w następnych publikacjach z prowadzonych badań.

Proponowany algorytm genetyczny dla optymalizacji struktury fizycznej bazy danych posiada cechy klasycznego algorytmu genetycznym (rys 1). Jego zastosowanie w procesie szukania lepszych rozwiązań dla projektowanej struktury fizycznej bazy danych nie wymaga definiowania algorytmu transformacji modelu logicznego bazy danych do struktury fizycznej. Pozwala również uniknąć prostego odwzorowania modelu logicznego na strukturę fizyczną bazy danych, które nie uwzględnia w dostateczny sposób czasu dostępu do danych.

Optymalizacja w tym zagadnieniu polega na poszukiwaniu fizycznej struktury bazy o coraz krótszym czasie dostępu do danych. Otrzymana w wyniku n-iteracji populacja chromosomów reprezentujących poszczególne atrybuty oznacza fizyczną strukturę]. bazy danych, zapewniającą najszybszy dostęp do danych.

Jedną z możliwości doskonalenia prezentacji powyższego problemu w postaci algorytmu genetycznego stanowi doskonalenie funkcji celu:

- w ten sposób, żeby uwzględniała typowy zestaw żądań informacyjnych na danej bazie danych,
- zapewniała precyzyjne uwzględnienie wpływu indeksów na szybkość wykonywania typowego zestawu żądań informacyjnych.

4. Możliwości wykorzystania algorytmów genetycznych w modelowaniu baz danych

Bardzo interesującym kierunkiem wykorzystania algorytmów genetycznych może być optymalizacja systemu indeksów w bazie danych. Wynika to z ich decydującego wpływu na czas wyszukiwania danych. wymaga specjalnych.

Złożoność zagadnienia uzasadnia potrzebę zastosowania algorytmu symulowanego wyżarzania, który jest szczególnym przypadkiem algorytmu genetycznego. Definiując przestrzeń potencjalnych rozwiązań, dla każdego elementu - **x** tej przestrzeni należy wyznaczyć wartość funkcji f(x) - jakość rozwiązania dla tego elementu. Wartość funkcji f(x) jest odwrotnie proporcjonalna w do jakości rozwiązania, posiada bowiem cechy funkcji kosztowej.

Realizując algorytm symulowanego wyżarzania generuje się kolejne przybliżenia - element $\mathbf{x}_{nowy}$. Wybór przybliżenia jako nowego elementu następuje na podstawie wartości $f(x_{nowy})$, jeżeli ma wartość mniejszą od f(x). Algorytm symulowanego wyżarzania jest szczególnym przykładem algorytmu genetycznego, w którym populacja zawiera chromosom zaś operatorem genetycznym jest mutacja. Algorytm ten znalazł zastosowanie w następujących dziedzinach (Cytowski, 1996):

- projektowanie komputerowe
- przetwarzanie obrazów
- optymalizacja kombinatoryczna
- optymalizacyjne problemy sztucznej inteligencji.

Optymalizacja indeksów w bazie danych może być również interpretowana jako uczenie się. Technologia ta wykorzystuje miedzy innymi metodę *zmiany struktur danych.* Uczenie się systemu tą metodą wymaga określenia reguł powiązania operatorów genetycznych (reprodukcja, krzyżowanie, mutacja) ze strukturą danych (Cytowski, 1996). Metoda uczenia się może być szczególnie przydatna dla optymalizacji indeksów, ponieważ system wykorzystuje funkcję użyteczności. zależną od stanu.

Funkcja użyteczności uwzględnia efekty długofalowe wynikające ze zmiany środowiska w odróżnieniu od funkcji nagrody, która uwzględnia skutki natychmiastowe. Metoda uczenia się na podstawie nabytych doświadczeń:pozwoli na definiowanie najlepszych indeksów danej bazy ze względu na czas dostępu do danych, a tym samym zwiększy gotowość informacyjną bazy danych.

Bibliografia

Cytowski, J. (1996) Algorytmy genetyczne, Podstawy i zastosowania, *Akademicka Oficyna Wydawnicza* , Warszawa.

Goldberg, D. (1995) Algorytmy genetyczne i ich zastosowania, *WNT*,Warszawa.

Gwiazda, T. (1995)Algorytmy genetyczne, Wstęp do teorii, *Biblioteka Sztucznej Inteligencji*, Warszawa.

Delobel, C.,Adiba M. (1989) Relacyjne bazy danych, *WNT*, Warszawa, s. 259-291.

Gardarin, G.,Gardarin O. (1996) Le Client-Serveur, *Editions Eyrolles*, Paris, s. 106-118, 170-176.

White C.J. (1999) Sybase Adaptive Server IQ – A high Performance database for Decision Processing, *database Associates International Inc.*, January, www.dbaint.com.

Wrembel R. (1997) Nowe struktury indeksów dla magazynów danych, *III Konferencja PLOUG*, Zakopane.

Ooi, B.C., Goh C.H., Tan K.L. (1998) Indexing bitemporal databases as points, *"Information and Software technology"*, nr 40, s. 327-337.

ISSN 0208-8029
ISBN 83-85847-53-7

W celu uzyskania bliższych informacji i zakupu dodatkowych egzemplarzy prosimy o kontakt z Instytutem Badań Systemowych PAN
ul. Newelska 6, 01-447 Warszawa
tel. 837-35-78 w. 241 e-mail: bibliote@ibspan.waw.pl