

же содержит реализацию функционала, предоставленного администратору системы. Класс UCDAIITree являет собою визуальную часть предложенного модуля.

### Выводы

Разработан программный модуль контроля версий динамически подсоединяемых библиотек для СУИ SmartBase ITS Control. Алгоритм, который базируется на методах динамического программирования, имплементирован на удаленном сервере, отведенном под репозиторий динамически подсоединяемых библиотек, что позволяет экономить вычислительные ресурсы устройств, на которых установлен модуль мониторинга.

### Список использованных источников

1. Ролик А.И. Система управления корпоративной информационно-телекоммуникационной инфраструктурой на основе агентского подхода / А.И. Ролик, А.В. Волошин, Д.А. Галушко, П.Ф. Можаровский, А.А. Покотило // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. — К.: «ВЕК+», 2011.
2. Maier D. The complexity of some problems of subsequences and supersequences. ACM Press. – 1978. – с. 322-366

УДК 004.65

## ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ДВОХФАЗНОГО АЛГОРИТМУ ПРОЕКТУВАННЯ МУЛЬТИБАЗОВИХ СХОВИЩ ДАНИХ

Яцишин А.Ю.

*Національний технічний університет України «Київський політехнічний інститут»*

### I. Постановка проблеми

У роботах [1] та [2] розглядається задача проектування і оптимізації мультибазових сховищ даних з врахуванням структурованості даних. Метою проектування є розміщення даних у сховищі (реляційна, багатовимірна бази даних, а також XML та NoSQL ) у відповідності з їх структурованістю, метою оптимізації є застосування методик (перерозташування структурованих та слабо структурованих даних, індексація, матеріалізація, горизонтальна та вертикальна фрагментація, злиття). Для оптимізації використовується генетичний алгоритм, у даній доповіді розглядається застосування так званої адаптивності за генами.

### II. Мета роботи

Метою дослідження є розробка двофазного алгоритму проектування мультибазових сховищ даних. У даній роботі пропонуються експериментальні результати проведеного дослідження.

### III. Дослідження двофазного алгоритму

Запишемо задачу оптимізації мультибазових сховищ даних.

Нехай  $K_1, \dots, K_n$ - класи запитів до сховища даних, що надійшли до сховища даних за звітний період,  $m(S, i)$  - умова застосування компоненту штрафу по оновленню,  $f_i^s$  - доля всіх запитів вибору даних класу  $K_i$  по відношенню до всіх запитів вибору у сховищі за звітний період,  $n_i$ - розмір даних для еталонних запитів, визначається як кількість даних, отриманих у результаті виконання запитів, або, якщо запит не повертає даних, сумарну загальну площу таблиць, що беруть участь у запиті;  $t_i^s$ - час виконання еталонного запиту вибірки даних.

Необхідно оптимізувати сховище даних, тобто знайти такий стан сховища  $S$ , що наступна функція приймає максимальне значення

$$\sum_{i=1}^n \frac{n_i f_i^s}{t_i^s(S)} \rightarrow \max, \quad (1)$$

Дана оптимізація проводиться у два етапи. На першому етапі дані розміщуються у сховищі відповідно до їх структурованості, на другому – отримане сховище оптимізується відповідно до статистики виконання запитів.

У таблиці 1 показана швидкодія запитів (1) до інтеграції даних, після неї, та після оптимізації сховища.

Показники інтегральної швидкодії виконання запитів

	Мінімальна швидкодія (1)	Середня швидкодія	Максимальна швидкодія
До інтеграції даних	413.0272	502.6794	587.2584
Після інтеграції даних	421.874	517.0587	608.1161
Після оптимізації	464.9081	560.4368	668.3728

Отримані результати свідчать про зростання показника інтегральної швидкодії до 5% після інтеграції даних, та зростання цього показника ще на 10% після оптимізації сховища.

Проведемо порівняння розробленого алгоритму (2) з стандартним ГА (1) та відомими методами – гілок і границь (3) та повного перебору (4), що відображено в таблиці 2.

Бачимо, що запропонований алгоритм є найбільш ефективним серед вищезазначених і дозволяє отримати кращий результат за меншу кількість часу. Зокрема, він затрачає у 10 раз менше часу, ніж метод гілок і границь, і у 100 раз менше, ніж метод повного перебору для  $N=10$ , а на довгих хромосомах ще менше, при тому знаходячи рішення, близьке до оптимуму.

Також було проведено дослідження залежності показника інтегральної швидкодії, а також часу роботи запропонованого двофазного алгоритму від його характеристик – ймовірностей мутації, схрещування, та значення порогу адаптивності за генами, запропонованої автором.

Таблиця 2

Порівняння алгоритму з існуючими аналогами

Алгоритм	Результат				Час			
	N=10	N=20	N=30	N=60	N=10	N=20	N=30	N=60
1	1510,591	1201,374	798,3923	468,50088	5094311	3480447	6413090	123343212
2	1514,614	1201,374	806,057688	478,21268	2049749	1778678	1119716	83159767
3	1518,398	1201,374	810,061039	-	27285594	30294614	268806546	-
4	1528,853	-	-	-	379662336	-	-	-

**Адаптивність за генами.** Результат до 50% зменшення часу оптимізації сховища даних можна досягти при використанні деяких характеристик алгоритму, зокрема порогу адаптивності, який визначає, при якій мінімальній відстані між хромосомами здійснюється фіксація генів. Можливий вииграш від використання адаптивності за генами сильно залежить від оптимізаційного потенціалу області, а точніше від віддаленості оптимальних значень від початкових. На малих  $N$  доцільно вибрати малі  $\alpha < N/2$ , на великих – великі  $\alpha > N/2$ .

**Мутація.** Як впливає з проведених випробувань, імовірність мутації доцільно вибрати невелику, близько 0.15. Це пояснюється тим, що при високих імовірностях мутації збільшується генетична відстань, що приводить до більшого часу алгоритму, однак не забезпечує кращого значення функції швидкодії. В деяких випадках, зокрема при великих значеннях  $N$ , доцільно вибрати  $m > 0.8$  для швидшого дослідження області допустимих рішень.

**Кросинговер.** У результаті аналізу проведених досліджень можемо стверджувати, що для малих  $N$  доцільно вибрати малу імовірність схрещування (0.35-0.45), при великих  $N$  доцільно вибрати велику імовірність схрещування (0.8 – 0.9), хоча в окремих випадках можливе отримання схожих результатів при  $c=0.45-0.65$ .

### Висновок

У результаті проведених експериментів бачимо збільшення швидкодії сховища за рахунок використання двофазного алгоритму до 15%. Крім того, використання адаптивності за генами дозволяє отримати скорочення часу виконання алгоритму до 50 %. Це пов'язано з зменшенням часу на перепроєктування сховища даних у тих випадках, коли воно не приводить до покращення цільової функції.

### Список використаних джерел

1. Математична модель задачі проектування гібридних сховищ даних з врахуванням структур джерел даних [Текст]. Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. / Томашевський В.М., Яцишин А.Ю. – К.: Век+, – 2011. – № 53. – 211 с.
2. Яцишин А.Ю. Проектування мультибазових сховищ даних на основі двофазного алгоритму [Текст]. Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. / Томашевський В.М., Яцишин А.Ю. – К.: Век+, – 2012. – № 55. – 211 с.