



ПРОБЛЕМНО І ФУНКЦІОНАЛЬНО ОРІЄНТОВАНІ КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ

УДК 681.5.015

ПРОГРАМНИЙ КОМПЛЕКС МОДЕЛЮВАННЯ СКЛАДНИХ СИСТЕМ НА ОСНОВІ ІТЕРАЦІЙНИХ АЛГОРИТМІВ МГУА З МОЖЛИВІСТЮ МЕРЕЖЕВОГО ДОСТУПУ

О.С. БУЛГАКОВА, В.В. ЗОСІМОВ, В.С. СТЕПАШКО

Описано структуру спеціалізованого програмного комплексу моделювання на основі ітераційних алгоритмів методу групового урахування аргументів (МГУА) з можливістю мультидоступу через Інтернет або локальну мережу. В програмному комплексі процес моделювання реалізовано у трьох режимах діалогу — два автоматичних (стандартний та плановий) та один інтерактивний, коли можна втручатися в процес самоорганізації моделей. Програмне забезпечення працює з різними наборами даних у форматі Excel та текстовому редакторі Блокнот. Будуються моделі різної складності та структури з різним розбиттям вибірки. Найкращі моделі представляються системою для графічного і змістового аналізу та зберігаються в базі даних разом із проміжними розрахунками і результатами експериментів для подальшого застосування. Система одночасно працює з трьома базами даних: початковою, базою даних проміжних розрахунків та базою даних результатів.

ВСТУП

Прикладні задачі прийняття рішень у складних системах часто розв'язуються на основі моделей, побудованих за експериментальними даними. Однак у проблемним є як урахування всіх чинників, що впливають на ситуацію в конкретних умовах, так і складність збору достовірної інформації.

На сьогодні існує багато методів, які використовують в задачах моделювання, але не всі вони орієнтовані на побудову моделей складних систем в умовах неповноти інформації. У таких задачах переважають методи та засоби індуктивного моделювання, призначені передусім для функціонального опису вхідно-вихідних характеристик систем.

Серед різноманітних методів моделювання за експериментальними даними вирізняється метод групового урахування аргументів (МГУА), автором якого є академік О.Г. Івахненко. Цей метод дозволяє будувати моделі безпосередньо за вибіркою даних, без залучення додаткової апріорної інформації.

МГУА успішно застосовують в задачах аналізу даних і виявлення закономірностей, прогнозування і моделювання, кластеризації та розпізнавання образів. Він дає можливість автоматично знаходити взаємозалежності в даних, обирати оптимальну складність моделі. Станом на сьогодні розроблено і досліджено багато різновидів алгоритмів МГУА перебірного та ітераційно-

го типу. Перебірні алгоритми ефективні як засіб структурної ідентифікації, але лише за обмеженого числа аргументів. Ітераційні алгоритми працездатні за великої кількості аргументів, але специфіка їхньої архітектури не гарантує побудови моделі істинної структури [1]. Досі ці два класи алгоритмів розвивались без поєднання їх сильних сторін.

Нині найбільш поширеним є класичний багаторядний ітераційний алгоритм МГУА, проте він має свої істотні недоліки: можливість втрати інформативних та/або включення неінформативних аргументів, експоненційне зростання степеня полінома. Комплексне вирішення цих проблем було запропоноване в архітектурі узагальненого ітераційного алгоритму [2]. Крім того, з розвитком Інтернет-технологій з'явилася можливість використання веб-інтерфейсу як засобу забезпечення комфортної та ефективної роботи користувача.

Мета роботи — представити інструментальні засоби для практичного застосування теорії, методів, алгоритмів та інтелектуальних технологій індуктивного моделювання, прогнозування і виявлення закономірностей для інформаційної підтримки прийняття рішень у складних системах.

ХАРАКТЕРИСТИКА ІТЕРАЦІЙНИХ АЛГОРИТМІВ ІНДУКТИВНОГО МОДЕЛЮВАННЯ

Ітераційні алгоритми МГУА [1–3] розв'язують задачі побудови моделей за вибіркою $W = (X, y)$ даних n спостережень за m вхідними та однією вихідною змінними. Для програмної реалізації в системі моделювання було обрано такі різновиди алгоритмів:

- багаторядний — класичний алгоритм МГУА, де в процесі обчислень на кожній ітерації (ряді селекції) формуються проміжні частинні моделі з усіх можливих пар аргументів, якими є F кращих виходів попереднього ряду r , де F — свобода вибору рішень:

$$y_l^{r+1} = f(y_i^r, y_j^r), \quad i = 1, \dots, F; \quad j = i + 1, \dots, F; \quad C_m^2; \quad (1)$$

- релаксаційний — алгоритм, у якому пари формуються з проміжних та початкових аргументів:

$$y_l^{r+1} = f(y_i^r, x_j); \quad (2)$$

- комбінований — алгоритм, де пари формуються як з проміжних, так і з початкових аргументів, тобто він об'єднує два попередніх

$$y_l^{(r+1)} = f(y_i^r, y_j^r) \vee f(y_i^r, x_j) \quad (3)$$

- узагальнений [1] — алгоритм, де пари формуються як у комбінованому, а також застосовується комбінаторна оптимізація складності всіх частинних моделей (рис. 1):

$$y_l^{(r+1)} = f_{\text{opt}}(y_i^r, y_j^r) \vee f_{\text{opt}}(y_i^r, x_j). \quad (4)$$

Очевидно, що три попередні різновиди алгоритмів є окремими випадками узагальненого. У цьому алгоритмі комбінаторна оптимізація складності частинних моделей полягає в тому, що на кожному ряді розглядаються моделі, наприклад, такого виду (у разі лінійного частинного опису):

$$f(y_i^r, y_j^r) = a_0 d_1 + a_1 d_2 y_i^{r-1} + a_2 d_3 y_j^{r-1}, \quad (5)$$

де $d_k, k=1,2,3$ — елементи двійкового структурного вектора d , що приймають значення 1 або 0 (включення чи невключення відповідного аргумента):

$$d_k = \{0, 1\}, \quad f_{\text{opt}}(y_i^r, y_j^r) = f(y_i^r, y_j^r, d_{\text{opt}}). \quad (6)$$

Схема перебору при цьому має вигляд:

$$\begin{array}{l} 100 \rightarrow f_1 = a_0 \\ 010 \rightarrow f_2 = a_1 y_i^{r-1} \\ 001 \rightarrow f_3 = a_1 y_j^{r-1} \\ 110 \rightarrow f_4 = a_0 + a_1 y_i^{r-1} \\ 101 \rightarrow f_5 = a_0 + a_1 y_j^{r-1} \\ 011 \rightarrow f_6 = a_1 y_i^{r-1} + a_2 y_j^{r-1} \\ 111 \rightarrow f_7 = a_0 + a_1 y_i^{r-1} + a_2 y_j^{r-1} \end{array} \left. \begin{array}{l} CR_1 \\ CR_2 \\ CR_3 \\ CR_4 \\ CR_5 \\ CR_6 \\ CR_7 \end{array} \right\} \Rightarrow f_{\text{opt}}, \quad (7)$$

при цьому обирається найкращий варіант за мінімумом критерію CR , тобто оптимізується складність частинної моделі.

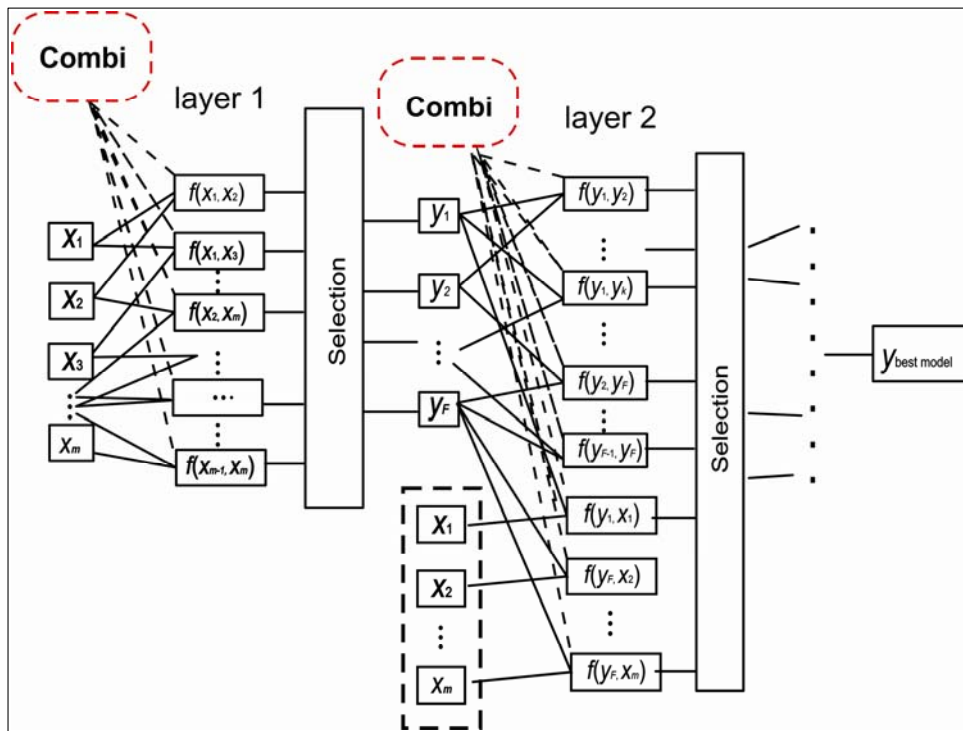


Рис. 1. Приклад схеми роботи узагальненого ітераційного алгоритму МГВА

Всі алгоритми шукають оптимальну модель як розв'язок задачі оптимізації:

$$f^* = \arg \min_{f \in \Phi} CR(y, f(X, \hat{\theta}_f)), \quad (8)$$

де CR — деякий критерій селекції, наприклад, критерій регулярності:

$$AR_{B|A} = \|y_B - \hat{y}_{B|A}\|^2 = \|y_B - X_B \hat{\theta}_A\|^2, \quad (9)$$

що ґрунтується на розбитті вибірки W на дві частини A та B обсягом n_A та n_B , $n_A + n_B = n$, де $\hat{\theta}_A$ — оцінка параметрів на підвибірці A за допомогою МНК.

ПРОГРАМНИЙ КОМПЛЕКС МОДЕЛЮВАННЯ СКЛАДНИХ СИСТЕМ ІЗ ЗАСТОСУВАННЯМ МЕРЕЖЕВИХ ТЕХНОЛОГІЙ

Нижче описано веб-технології для моделювання складних систем на основі ітераційних алгоритмів МГУА різного типу з можливістю доступу через Інтернет. Як ядро розробленої технології застосовується узагальнений ітераційний алгоритм МГУА.

Серверно-клієнтська взаємодія в програмному комплексі визначає функціональний розподіл між клієнтською та серверною частиною на так звані «операційні рівні»:

- інтерфейс користувача — веб-інтерфейс, який відповідає за представлення даних та своєчасне реагування на команди користувача;
- сервер, який відповідає як за прикладний рівень, на якому виконується обробка інформації, що поступила від користувача, так і за рівень управління даними, забезпечуючи їх зберігання та доступ до них.

Загальна функціональна схема програмного комплексу на основі ітераційних алгоритмів МГУА для моделювання складних систем із застосуванням мережових технологій представлено на рис. 2, яка складається з декількох блоків: блок зберігання даних, в якому зберігаються як вхідні та вихідні дані, так і проміжні результати; блок формування задачі, в якому можна задавати управляючі параметри; блок розв'язання задачі, в якому процес моделювання може виконуватися в трьох режимах — два автоматичних та інтерактивний. Нижче кожен з блоків розписано більш детально.

За допомогою блоку зберігання даних, маючи початкову вибірку (отриману з наявного файлу), є можливість розбивати дані на проекти, зберігати проміжні розрахунки для подальшого продовження процесу моделювання, зберігати остаточні результати розрахунків, а також використовувати отримані моделі для нових даних.

Система одночасно працює з трьома базами даних: початковою базою даних, базою даних розрахунків та базою даних результатів. Більш детально блок зберігання даних розглянуто на рис. 3.

Після того, як початкову вибірку згенеровано або отримано, переходимо до блоку формування задачі (рис. 4). На цьому етапі вибірка поділяється на дві частини — навчальну та перевірочну: на навчальній оцінюються коефіцієнти моделі, на перевірочній вибираються кращі моделі ряду за допомогою різних критеріїв (за замовчуванням — критерій регулярності). У ході генерації вибірки даних є можливість задавати: тип розбиття вибірки, рівень шуму, зовнішній критерій, вибір алгоритму моделювання.

Далі, залежно від використання різних модифікацій багаторядного алгоритму МГУА, генеруються моделі різної складності, для кожної з яких обчислюється значення критерію, за яким вони відбираються для наступного ряду (рис. 5).

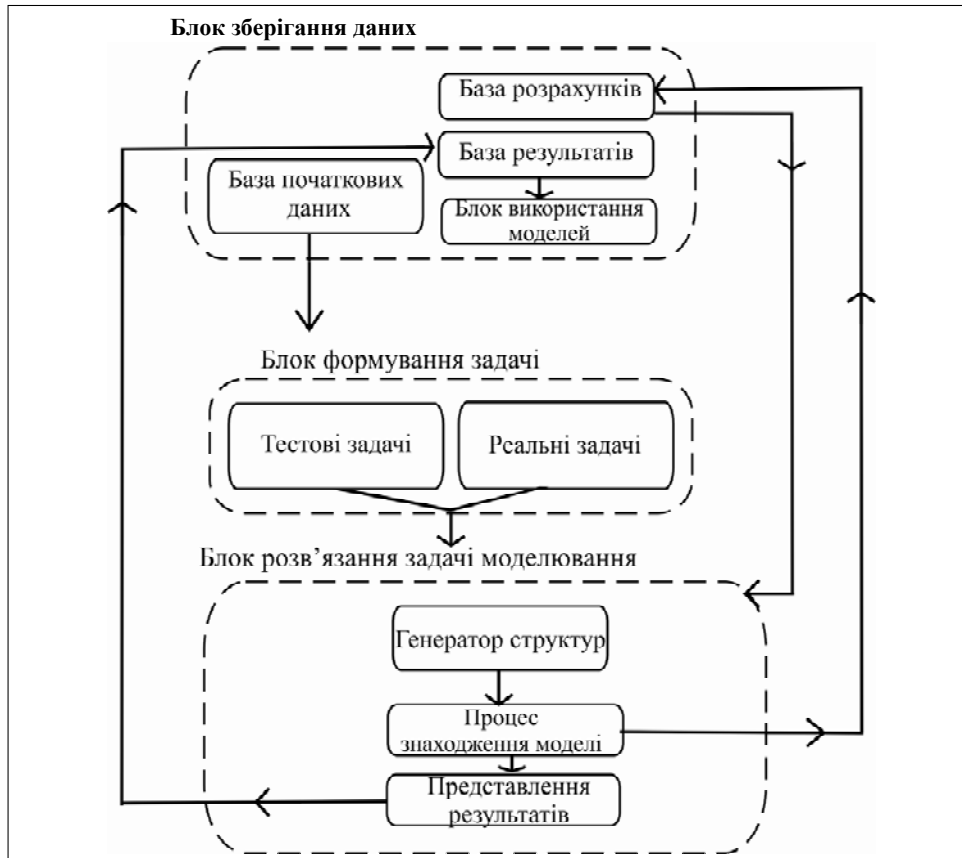


Рис. 2. Загальна функціональна схема програмного комплексу на основі ітераційних алгоритмів МГУА для моделювання складних систем із застосуванням мережесих технологій

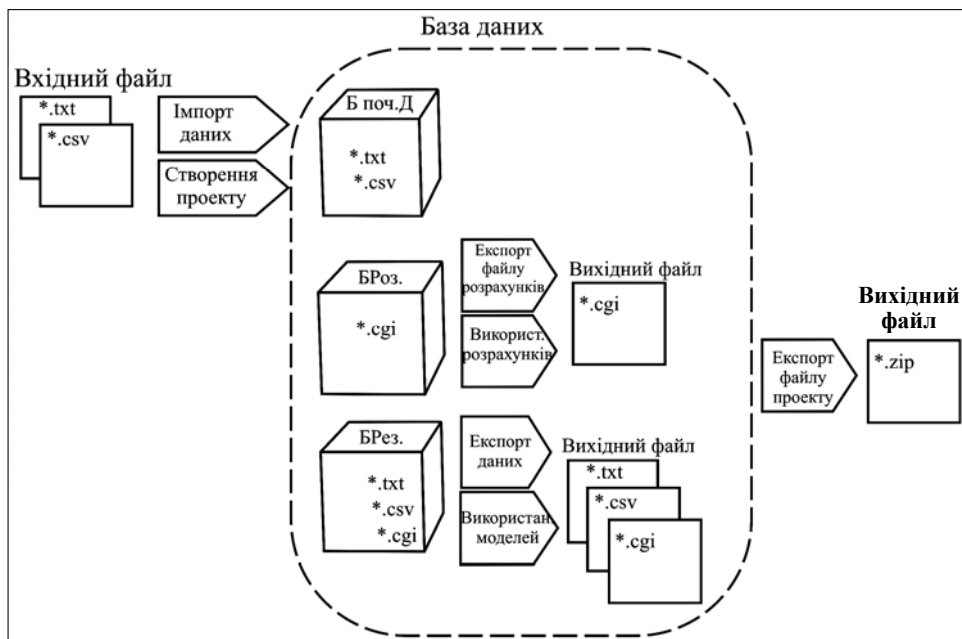


Рис. 3. Приклад блоку зберігання даних

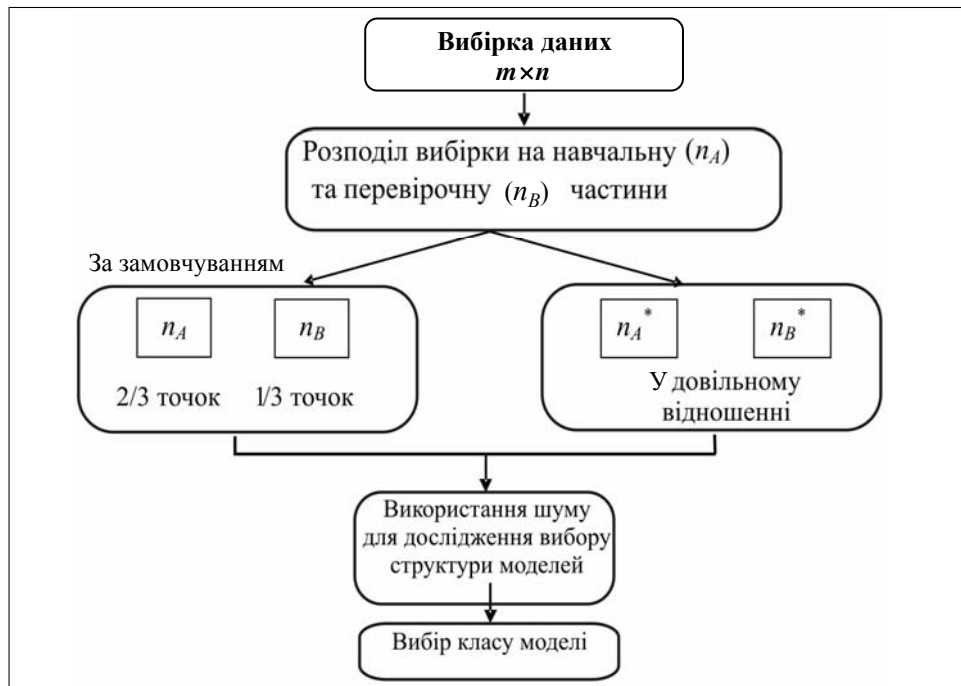


Рис. 4. Блок формування задачі моделювання

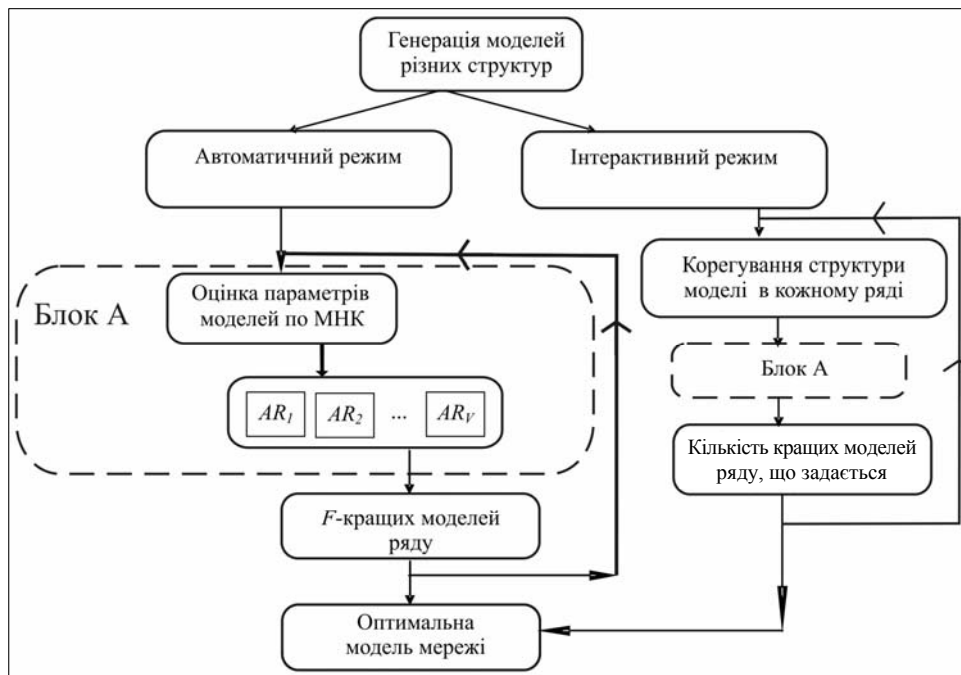


Рис. 5. Блок розв'язання задачі моделювання (А — оцінка параметрів моделей та знаходження значення критерію регулярності для кожної моделі)

У програмному комплексі процес моделювання може бути реалізовано у трьох режимах (два автоматичних та один інтерактивний):

- *автоматичний* (процес самоорганізації моделей виконується автоматично). Крім того, автоматичний режим реалізовано у двох варіантах:

– стандартний — задається вид частинного опису та свобода вибору однаковими для всіх без винятку рядів;

– планований — процес самоорганізації моделей виконується автоматично з заданим планом, тобто коли вид частинного опису та свобода вибору задаються різними для різних рядів:

- *інтерактивний* (можна безпосередньо втручатися в процес самоорганізації моделей):

- на будь-якому ряді включати або не включати модифікації;

- змінювати складність моделей частинного опису;

- вибирати різну кількість моделей, що перейдуть на наступний ряд;

- використовувати різні критерії вибору кращих моделей.

Крім того, процес самоорганізації можна зупиняти на довільному етапі обчислення, а потім у будь-який момент часу продовжити розрахунки, при цьому всі проміжні розрахунки будуть збережені. Різні аспекти організації інтерфейсу користувача та особливості його використання описано нижче.

РЕАЛІЗАЦІЯ ПРОГРАМНОГО КОМПЛЕКСУ З МОЖЛИВІСТЮ КЕРУВАННЯ ЧЕРЕЗ ВЕБ-ІНТЕФЕЙС

Основний код програми написано у процедурному стилі. Основним способом взаємодії з програмою є веб-доступ, проте з нею можна працювати й локально, запустивши одночасно декілька процесів без додаткових затрат часу. Розглянемо більш детально основні принципи організації доступу.

Принципи організації Інтернет-доступу до програми

Програму розбито на модулі, велика частина яких підтримує *cli* (інтерфейс командного рядка). Проте основним способом взаємодії з програмою є веб-доступ, для реалізації якого використовується *perl* модуль *main.cgi* на серверній стороні та *javascript* модуль *index.htm* на клієнтській стороні. Розглянемо спершу загальну схему роботи веб-додатків, які відносяться до категорії клієнт-сервер.

В ролі клієнта найчастіше виступає спеціальна програма — браузер. Браузер формує візуальне відображення даних і надає користувачу можливість взаємодії з ними використовуючи клавіатуру, мишу, тачпад та інші маніпулятори. Також браузер надає середовище для дії коду, що виконується на стороні клієнта. Найбільш універсальною і найчастіше виконуваною мовою для клієнтської сторони є *javascript*. Браузер може звертатися до веб-сервера за протоколом *HTTP*, при цьому дії користувача перетворюються на запити спеціального виду, а відповіді сервера візуалізуються. Частину перетворень браузер може робити самостійно, проте найбільші можливості досягаються взаємодією браузера та клієнтського коду сторінки, яка переглядається.

В якості сервера може виступати як безпосередньо серверна частина додатку, так і спеціальне програмне забезпечення, яке називається веб-сервером. Веб-сервер приймає запити за *HTTP*-протоколом, аналізує їх та видає відповідь самостійно або передає перетворені параметри запиту додатку, чекає його виконання і передає відповідь додатку клієнту, можливо частково його модифікуючи. Існує кілька способів взаємодії веб-сервера і серверного додатку, з них найстарішим, але, як і раніше, дуже актуальним є *cgi-common gateway interface*.

Веб-доступ до додатка можна подати як чотирикомпонентну схему (рис. 6).

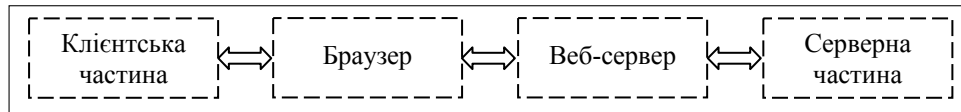


Рис. 6. Організація веб-доступу до додатка

В цій схемі браузер і веб-сервер є зовнішніми (щодо додатку) з частково замінюваними компонентами. При цьому один веб-сервер може обслуговувати одразу декілька клієнтів, тоді як один клієнт може звертатися до різних веб-серверів. Також варто зазначити, що у разі правильного підходу серверна частина додатка визначає можливості, що надаються, а клієнтська — інтерфейс. Причому клієнтська частина може бути більш ніж одного виду, тим самим дозволяючи надати користувачу можливість вибору між різними інтерфейсами до одного і того ж функціонала.

Серверну частину нашого додатка виконано на мові Perl, як і основний код. Усі взаємодії з веб-сервером зібрано в модулі `main.cgi`. Він реалізує початковий розбір і перевірку параметрів, здійснює аутентифікацію користувача і визначає доступні йому дані, вибирає відповідний для обробки запиту модуль, перетворює отримані дані у відповідну форму, забезпечує коректну обробку фатальних помилок у модулі, що викликаний. Спілкування з веб-сервером йде за протоколом `cgi`, текстова інформація для клієнтської частини передається у вигляді JSON.

Клієнтську частину виконано мовою javascript із використанням об'єктно-орієнтованого фреймворка `extJS` та концепції AJAX й вона має вигляд єдиної сторінки `index.htm`. Запити до сервера надсилаються без перезавантаження сторінки, отримані відповіді змінюють її зовнішній вигляд. У разі недоступності сервера або збою серверної частини користувачу виводяться повідомлення щодо помилок. Для зменшення кількості звернень до сервера здійснюється часткова перевірка заповнення форм і відключення частини елементів, поки не пройдено аутентифікацію, а для збереження інформації про це використовуються `cookies`. Варто зазначити, що всі перевірки на клієнтській стороні зроблено тільки для зручності, за бажання вони легко обходяться користувачем, перевірки безпеки здійснюються на серверній стороні.

Принципи організації локального доступу до програми

Доступ до Інтернету не є необхідним для роботи з комплексом, веб-сервер цілком можна запустити на машині клієнта або в межах локальної мережі.

Локальний доступ до програми реалізовано за тією ж схемою, що і веб-доступ (рис. 6). Різниця полягає в тому, що для локального доступу не потрібне підключення до мережі Інтернет. У ролі сервера може виступати будь-який комп'ютер, підключений до локальної мережі, на який встановлено відповідне програмне забезпечення.

В роботі локальної версії програми використовується той же веб-інтерфейс, що й у програмі з Інтернет-доступом. Оператор, який звик користуватися програмою з Інтернет-доступом, навіть не побачить різниці, і йому не треба буде вивчати новий інтерфейс та звикати до нього.

У комплексі реалізовано можливість ведення кількох розрахунків одночасно. Для кожного окремого розрахунку використовується окреме ядро процесора. Це дозволяє кільком операторам вести одночасні розрахунки без втрати швидкості роботи програми.

ІНТЕРФЕЙС КОМПЛЕКСУ

Розроблений інтерфейс програми являє собою сукупність засобів, за допомогою яких користувач може керувати процесом моделювання. Реалізовано такі особливості інтерфейсу: процес самоорганізації можна зупинити на будь-якому етапі обчислення, а потім у будь-який момент часу продовжити розрахунки, при цьому всі проміжні розрахунки будуть збережено; на будь-якому ряді можна включати або не включати різні модифікації, змінювати складність моделей частинного опису, вибирати різну кількість моделей, що перейдуть на наступний ряд, змінювати критерії вибору кращих моделей.

Далі розглянуто особливості інтерфейсу.

База даних

У базі даних зберігаються початкові дані, дані розрахунків та дані результатів. Маючи початкову вибірку, можна розбивати дані на проекти, зберігати проміжні розрахунки для подальшого продовження процесу моделювання, зберігати результати розрахунків, а також використовувати отримані моделі на нових даних.

Введення даних виконується поза системою, використовуючи для цього табличний редактор Excel або текстовий редактор Блокнот.

Під час завантаження даних на сервер можна створити окремий проект, до якого будуть вносити усі розрахунки та результати, пов'язані з ним (рис. 7).

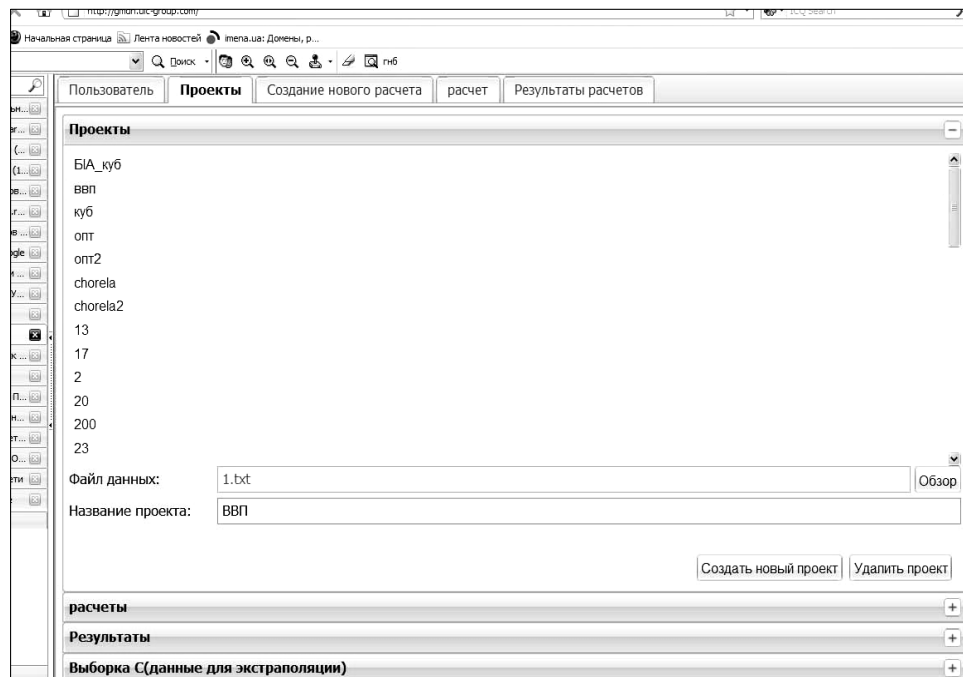


Рис. 7. Приклад веб-інтерфейсу програмного комплексу: вкладка «Проекты»

Всі розрахунки, які проводилися з даними, зберігаються у вкладці «Расчеты». Це дає змогу на будь-якому етапі зупинити процес моделювання, зберегти розрахунок на цьому етапі та продовжити процес, коли буде зручно. Бінарний файл розрахунків можна завантажити як на власний комп'ютер, так і на сервер через веб-інтерфейс для продовження розрахунків.

Для більш зручної роботи з базою розрахунків створено можливість їх копіювання. Така функція дає можливість експериментувати з файлом розрахунків, використовуючи його копію, а не оригінал.

Результати процесу моделювання зберігаються на вкладці «Результаты», де також є можливість завантажити файл результатів як на ПК, так і на сервер (таблиця).

Таблиця. Структура файлу результатів для Excel або Блокнот

1	#chaos: 0	num A: 19		MR=2	
	DM=1	IC=2			
2	#iteration: 1	num:0	add_base:0	opt:/	error: 41,530668
3	#iteration: 2	num:10	add_base:0	opt:/	error: 33,869917
4	#iteration: 3	num:10	add_base:0	opt:/	error: 31,615493
5	#iteration: 4	num:10	add_base:0	opt:/	error: 31,589807

6	Формула:				
7	$y = -121435,752 + 1,025 * X7 + 0,368 * X5 + 23,673 * X2 + 3581,541 * X3 + 0,361 * X1$ Помилка на A: 27,851 Помилка на B: 31,589 Середня помилка на A: 1,255 Середня помилка на B: 3,858				
8	Значення заданої вибірки A: 157,291 204,353 223,079... Значення заданої вибірки B: 228,414 202,363 218,522...				
9	Значення отриманої вибірки A: 156,281 205,113 223,082... Значення отриманої вибірки B: 229,412 202,472 217,112...				

У таблиці сірим кольором виділено слова, які не пишуться в файлі результату. Значення слів, написаних англійською мовою, таке:

- chaos — шум;
- num A — кількість рядків (точок) у вибірці A;
- iteration — номер ітерації;
- num — кількість відібраних моделей ряду, які перейдуть на наступний ряд (якщо 0, то переходять усі частинні моделі);
 - add_base — додавання початкового базису (0 — без додавання, 1 — з додаванням);
 - opt (optimization) — оптимізація (l — комбінаторна оптимізація лінійного частинного опису, sq — комбінаторна оптимізація квадратичного частинного опису);
 - error — помилка ряду;
 - DM, IC, MR — ідентифікатори, що використовуються для кодування ітераційних алгоритмів. Наприклад, DM = 1, IC = 2, MR = 2 відповідає релаксаційному ітераційно-комбінаторному алгоритму зі стандартним автоматичним режимом.

На вкладку «Выборка С» завантажуються дані для перевірки отриманої моделі на додатковій екзаменаційній частині даних.

Блок формування задачі

На цьому етапі відбувається процес формування задачі:

- вибір модифікації алгоритму;
- розбиття вибірки: за замовчуванням вибірку буде поділено у пропорції 2/3 (навчальна підвибірка А) та 1/3 (перевірочна підвибірка В). Якщо вказати частку точок для А в інтервалі від 0 до 1, то для В залишиться решта точок;
- додавання шуму на вектор значень вихідної змінної. Шум задається в інтервалі від 0 до 1, де 0 — нульовий рівень шуму, а 1 — 100% шуму;
- призначення критерію, за яким будуть відбиратися кращі моделі ряду. Алгоритм може змінювати свою структуру за заданим планом, наприклад, запис sq 1 20 означає:
 - sq — комбінаторна оптимізація квадратичного частинного опису (lin — лінійного);
 - 1 — з додаванням початкового базису (0 — без додавання);
 - 20 — свобода вибору $F = 20$.

Кожен рядок вказує номер ряду (ітерації) алгоритму, останній рядок дублюється до кінця процесу. Якщо останній рядок буде містити літеру s, то це означає зупинку процесу, який можна буде продовжити в інтерактивному режимі. Після того як задачу сформульовано, слід натиснути кнопку «Создать». Як тільки розрахунки закінчатся, користувача буде повідомлено у відповідному діалоговому вікні.

Модуль розрахунків

Далі, залежно від використання різних варіантів оптимізації (які було обрано на попередньому етапі), генеруються моделі різної складності, для кожної з яких обчислюється значення критерію, за яким вони відбираються для наступного ряду (рис. 8).

На цьому етапі можна продовжувати розрахунки в інтерактивному режимі, коли в процес побудови моделей можна безпосередньо втручатися: продовжити обчислення, в якому на будь-якому ряді можна включати або не включати запропоновані модифікації; змінювати складність моделей частинного опису; вибирати різну кількість моделей, що перейдуть на наступний ряд; використовувати різні критерії вибору кращих моделей.

Така структурна побудова дає змогу експериментувати з вхідними даними на кожному етапі, тим самим у режимі онлайн змінюючи структуру алгоритму.

В цьому модулі відображаються як графічно, так і чисельно значення мінімальних помилок попередніх та наступних рядів. Також тут можна побачити помилки всіх розв'язків конкретного ряду. Зберегти обрану модель можна, натиснувши на знак «плюс», після чого з'явиться вікно «сохранение расчетов», у якому слід задати ім'я отриманого результату, а також додати коментар.

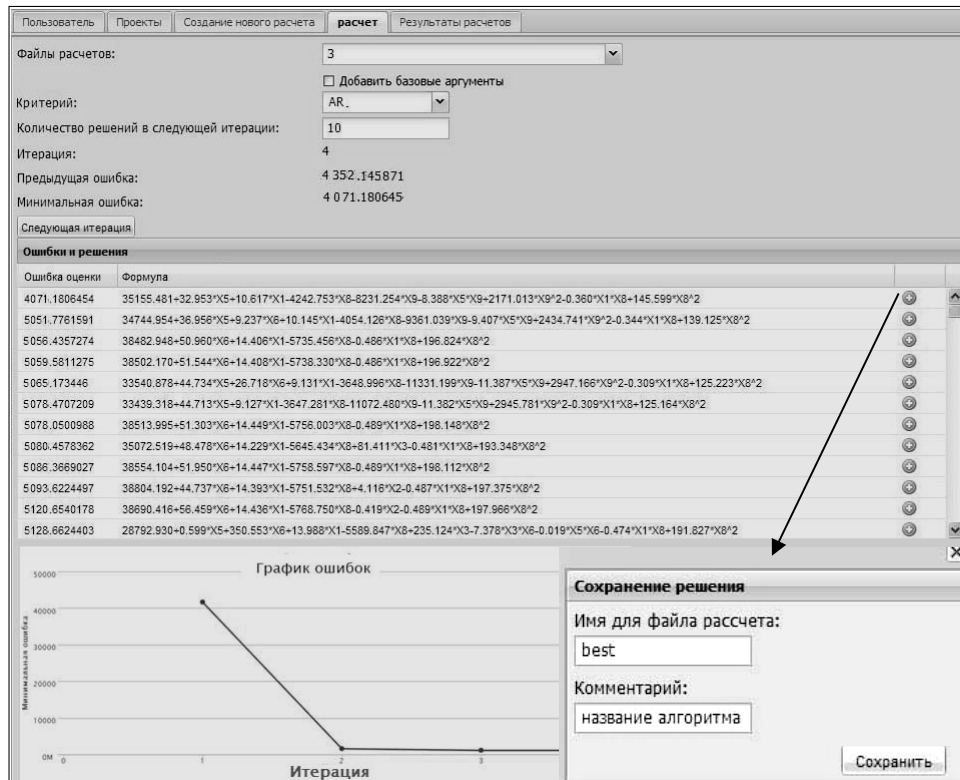


Рис. 8. Пример модулю розрахунків

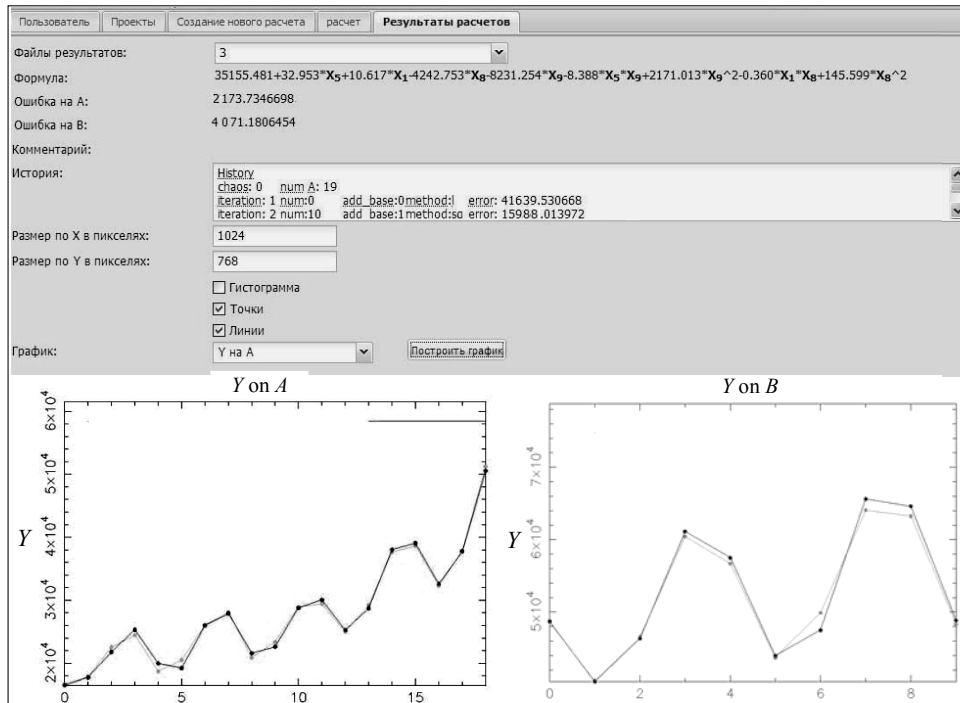


Рис. 9. Пример модулю результатів

Після того, як процес моделювання завершено, переходимо на вкладку «Результаты расчетов» (рис. 9), де можна побачити:

- формулу, яка відображає залежність вхідних змінних від вихідної;
- значення критерію на підвибірках А та В;
- середнє значення критерію на підвибірках А та В;
- історію процесу моделювання: кількість ітерацій, кількість моделей ряду, похибка ряду, модифікація алгоритму.

Крім того, можна побудувати графіки функцій, абсолютних помилок на різних підвибірках у вигляді гістограм, ліній та точок.

ВИСНОВКИ

Описане програмне забезпечення дозволяє працювати з різними наборами даних, виконувати сплановані обчислювальні експерименти [3] та розв'язувати практичні задачі моделювання. Програма має зручний інтерфейс для роботи з даними та різними варіантами ітераційних алгоритмів МГУА. Моделі, які побудовано, надаються системою для графічного і змістовного аналізу та зберігаються в базі даних для подальшого застосування. Реалізовано онлайн-систему, в якій процес моделювання може бути виконаний в автоматичному та інтерактивному режимах з Інтернет-доступом.

Описані в статті засоби автоматизованої технології побудови моделей складних систем на основі узагальненого ітераційного алгоритму МГУА застосовувались як для порівняльних досліджень ефективності різних алгоритмів за допомогою обчислювальних експериментів, так і для розв'язання реальних задач моделювання екологічних та економічних об'єктів і систем.

ЛІТЕРАТУРА

1. *Ивахненко А.Г., Юрачковский Ю.П.* Моделирование сложных систем по экспериментальным данным. — М.: Радио и связь, 1986. — 118 с.
2. *Степашко В.С., Булгакова О.С., Зосімов В.В.* Гібридні алгоритми самоорганізації моделей для прогнозування складних процесів // Індуктивне моделювання складних систем. Збірник праць, випуск 2. — Київ: МННЦ ІТС, 2010. — С. 236–246.
3. *Булгакова О.С., Степашко В.С.* Порівняльний аналіз ефективності ітераційних алгоритмів МГУА за допомогою обчислювальних експериментів / Булгакова О.С // Вісник ЧДТУ. — № 1. — Черкаси: Вид-во ЧДТУ, 2011. — С. 41–44.

Надійшла 22.05.2012