

7. Методи захисту банківської інформації: Навчальний посібник. / В.К. Задірака, О.С. Олексюк, М.О. Недашковський. – К.: Вища шк., 1999. – 261 с.
8. *Куканова Н.Ю.* Описание классификации угроз DSECCT // Научно-технический вестник СПбГУ ИТМО. – СПб., 2006. – Вып. 29. – С. 175–178

Поступила 1.02.2010р.

УДК 65.012.123.001.26:575(045)

Т.І. Олешко, Н.В.Ратушна

ОПТИМІЗАЦІЯ БАГАТОПАРАМЕТРИЧНИХ ФУНКЦІЙ ЗА ДОПОМОГОЮ ГЕНЕТИЧНИХ АЛГОРИТМІВ

Генетичні алгоритми є важливою складовою еволюційних методів. У світі спостерігається бум розвитку генетичних алгоритмів. Відбулись десятки міжнародних конференцій, опублікована величезна кількість літератури по генетичних алгоритмах, і інтерес до цієї тематики постійно зростає.

Генетичні алгоритми – це процедури пошуку, засновані на механізмі природного відбору і наслідування. Вони відрізняються від інших оптимізуючих і пошукових процедур такими ознаками [1]:

- обробляють не значення параметрів самої задачі, а їх закодовану форму;
- здійснюють пошук рішень, враховуючи не одиничну точку, а їх деяку популяцію;
- використовують тільки цільову функцію, а не іншу додаткову інформацію;
- використовують ймовірність, а не детерміновані правила вибору;
- у процесі еволюції кожна нова популяція залежить тільки від попередньої.

Генетичні алгоритми застосовуються до вирішення багатьох наукових і технічних проблем. Проте, можливо найпопулярніше застосування генетичних алгоритмів – оптимізація багатопараметричних функцій. Багато задач можуть бути сформульовані як пошук оптимального значення, де значення – складна функція, що залежить від певних вхідних параметрів. У деяких випадках потрібно знайти ті значення параметрів, при яких досягається найкраще значення функції. В інших випадках глобальний екстремум не потрібний – рішенням може вважатися будь-яке значення, краще за певну задану величину. У цьому випадку генетичні алгоритми – часто найкращий метод для пошуку "прийнятних" значень. Особливість генетичного алгоритму полягає в його здатності маніпулювати одночасно багатьма параметрами.

Тому не дивно, що вчені звернулися до теорії еволюції в пошуках

натхнення. Можливість того, що обчислювальна система, наділена простими механізмами змінності і відбору, могла б функціонувати за аналогією з законами еволюції в природних системах, була дуже приваблива. Ця надія стала причиною появи ряду обчислювальних систем, побудованих на принципах природного відбору.

Незважаючи на величезний інтерес до галузі еволюційного моделювання, кількість реально працюючих програм в цій галузі доволі мала. Роботи в цій галузі можна розділити на декілька великих категорій.

Перша категорія програм – пакети, що реалізують класичний генетичний алгоритм з можливим налаштуванням параметрів управління основними операторами ГА. Модель хромосоми в таких пакетах має, як правило, стандартну бінарну структуру, а функція відбору задана одним математичним виразом. Істотним недоліком цих методів є прив'язка до бінарної або числової моделі хромосоми (хромосома не може мати складної структури). До другої категорії програм по генетичних алгоритмах належать спеціалізовані програми, призначені для вирішення конкретних завдань. Ці генетичні алгоритми розроблені і оптимізовані для вирішення вузької, чітко визначеної проблеми. Третя категорія розробок по генетичних алгоритмах включає суто наукові дослідження, що полягають в дослідженні властивостей і характеристик різних генетичних алгоритмів, їх збіжності і виродженості.

Ефективність генетичного алгоритму при розв'язанні конкретної задачі залежить від багатьох чинників і, зокрема, від таких, як генетичні оператори і вибір відповідних значень параметрів, а також способу представлення рішення на хромосомі. Оптимізація цих чинників приводить до підвищення швидкості і стійкості пошуку, що істотно впливає на застосування генетичних алгоритмів.

Нині найвідомішими є 3 способи створення початкової популяції.

Стратегія “ковдри” – формування повної популяції, що містить всі можливі рішення.

Наприклад, для n -розрядної хромосоми існує 2^n варіантів рішень, які складають повну популяцію.

Стратегія “дробовика” – генерація достатньо великої випадкової множини рішень.

Стратегія “фокусування” – генерація підмножини рішень, що включає різноманіття одного рішення [2].

Перший підхід практично неможливо реалізувати, навіть для задач середньої розмірності, через величезні втрати пам'яті та процесорного часу. Також варто зауважити, що тоді початкова популяція не може розвиватися, оскільки в ній вже міститиметься найкращий розв'язок. Третій спосіб використовується тоді, коли є гіпотеза, що оптимальне рішення є варіацією відомого рішення. У цьому випадку час пошуку оптимального рішення різко зменшується, оскільки алгоритм розпочинає свою роботу в області оптимуму.

Найчастіше застосовують другий спосіб. У цьому випадку в результаті еволюції є можливість перейти в інші області пошуку, і кожна з них має порівняно невеликий простір пошуку. Загалом ефективність генетичного алгоритму, якість рішень і подальша еволюція значною мірою визначаються структурою і якістю початкової популяції. Крім того, застосовують комбінацію другого і третього способів. Спочатку визначаються особи з високим значенням цільової функції, а потім випадково формуються початкові рішення в цих областях.

Оскільки підсистема орієнтована на роботу з певною кількістю параметрів та пошуку оптимального розв'язку з великою точністю, то неможливо реалізувати першу стратегію через брак пам'яті. Третя стратегія теж не підходить, оскільки в такому разі необхідно проводити додаткові дослідження функції, що вже виходить за межі поняття генетичного алгоритму. Через великі розміри хромосоми неможливо здійснити генерацію одразу цілої хромосоми, тому генерація відбувається побітово для кожної хромосоми.

Для реалізації підсистеми побудови генетичних алгоритмів застосовують мову програмування C++. Вона дозволяє максимально компактно та ефективно оперувати даними, будувати великі абстрактні моделі та одночасно доступатися до низьких рівнів програмування, що дає змогу повністю контролювати процес реалізації та налагодження алгоритму на всіх його етапах. Вибір цієї алгоритмічної мови також пов'язаний з найефективнішою підтримкою програмних інтерфейсів, необхідних для реалізації окремих функцій майбутньої системи. Крім того, алгоритмічна мова забезпечена зручним у користуванні середовищем розроблення програм, яке дозволяє легко уникати помилок на етапі розроблення, а також містить велику довідкову систему, яка допомагає у процесі реалізації системи. І, насамкінець, мова дає змогу легко переходити на інші алгоритмічні мови різних рівнів без втрат часу на налаштування великої кількості параметрів, що економить ресурси програміста [3].

Наприкінці можна підсумувати, що генетичні алгоритми є ефективною процедурою пошуку, що конкурує з іншими процедурами. Ефективність генетичних алгоритмів сильно залежить від таких деталей, як метод кодування рішень, операторів, налаштування параметрів, окремих критеріїв успіху. Теоретична робота, відбита в літературі, присвяченої генетичним алгоритмам, не дає підстав говорити про вироблення певних строгих механізмів для чітких передбачень.

1. *Скобцов Ю.О.* Основы эволюционных вычислений: Учебн. пособие. – Донецк: ДонНТУ,
2. *М.М. Лобур, С.П. Ткаченко.* Розробка підсистеми для дослідження генетичних алгоритмів з використанням шаблонів. – Національний університет «Львівська Політехніка». – 2009.
3. *В.І. Дубровін, С.О. Суботін.* Методи оптимізації та їх застосування в задачах

Поступила 25.01.2010р.

УДК 519.711.3, 519.767.6

А. А. Верлань, О.В. Нечипоренко, Ю.О.Фуртат

ЛОГІЧНИЙ ВИВІД ТА КЕРУВАННЯ В СИСТЕМІ НЕФОРМАЛЬНОГО ПРОГРАМУВАННЯ

Розглядається задача розробки механізму нечіткого логічного виводу в системі неформального програмування. Для цієї системи конкретизуються поняття «логіко-лінгвістичне правило», «умовно нечітке висловлювання», «квантифікатор», «модифікатор». Описується модель нечіткого логічного виводу в СНЕП.

Також для системи розглядається керуюча структура. Визначаються семантичні категорії «загальне правило – окреме правило», «загальне правило – виняток з правила». Описується використання цих категорій в створенні керуючої структури системи.

The task of fuzzy inferencing mechanism development in the system of the informal programming is considered. For this system concepts «logical-linguistic rule», «conventionally fuzzy statement», «quantifier», «modifier» are specified. The model of fuzzy inferencing is described in SInP.

Also for the system a control structure is considered. Semantic categories «general rule – separate rule» «general rule – exception» are determined. The use of these categories in system's control structure creation is described.

Вступ. При розв'язуванні задач, що складно формалізуються, використовується нечітка інформація: поняття і відношення з нечіткими межами, лінгвістична інформація, і введення строгих меж або штучне введення однозначності означає огрублення початкових даних і може стати причиною отримання чіткого, але невірного результату [1].

В межах системи неформального програмування (СНЕП) для розв'язування цього класу задач розроблено механізм нечіткого логічного виводу, модель якого працює з логіко-семантичним описом предметної області і може задаватись або в явній формі (будується модель, зберігається в пам'яті ЕОМ і в при надходженні запиту дані обробляються в цій моделі), або в неявній формі.

Логіко-лінгвістичні правила. В СНЕП реалізується неявна модель нечіткого логічного виводу, модель не зберігається в пам'яті ЕОМ, запит, що надійшов, обробляється за певним алгоритмом. Така реалізація не впливає