

УДК 007.52

О.П. Чапланов, О.Б. Чапанова

*Харківський національний університет радіоелектроніки***ЕФЕКТИВНІ МОДЕЛІ КРОСОВЕРУ ДЛЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ**

*Запропоновано модифіковану, вдосконалену модель генетичного оператора кросоверу для більш ефективного рішення задачі комівояжера.*

*кросовер, задача комівояжера, генетичний оператор, методи оптимізації*

**Вступ**

**Постановка проблеми.** Сьогодні в багатьох галузях науки та промисловості постає задача економії різноманітних ресурсів – енергетичних, паливних, транспортних, тобто оптимальних за якимось критерієм. Це великий клас задач, до якого, перш за все, можна віднести задачі оптимізації багатокстремальних функцій, комбінаторні тощо.

Одним із сучасних методів оптимізації є генетичні алгоритми. Ще у 1859 р. Ч. Дарвін виявив головний механізм розвитку: відбір у сполученні зі змінністю або, як він його називав, "спуск із модифікацією". Історія еволюційних обчислень почалася з розробки ряду різних незалежних моделей. Основними з них були генетичні алгоритми і класифікаційні системи Холанда (Holland), які отримали загальне визнання після виходу у світ книги, що стала класикою в цій області, - "Адаптація в природних і штучних системах" ("Adaptation in Natural and Artificial Systems", 1975).

Вчені, які займаються комп'ютерними дослідженнями, звернулися до теорії еволюції. 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. Його суть полягає в тому, що найбільш пристосовані особини краще виживають. Помітимо, що сам по собі природний відбір ще не забезпечує розвиток біологічного виду. Тому дуже важливо вивчити, яким чином відбувається спадкування, тобто як властивості нащадку залежать від властивостей батьків.

Генетичні алгоритми (ГА) базуються на теоретичних досягненнях синтетичної теорії еволюції, що враховує мікробіологічні механізми спадкування ознак у природних і штучних популяціях організмів, а також на накопиченому людством досвіді в селекції тварин і рослин. Генетичний алгоритм – послідовність управляючих дій і операцій, яка моделює еволюційні процеси на підставі аналогів механізмів генетичного наслідування та природного відбору. При цьому зберігається біологічна термінологія в спрощеному вигляді.

ГА працюють із сукупністю "особинь" – популяцією, кожна з яких представляє можливе вирішен-

ня даної проблеми. Кожний особень оцінюється мірою його "пристосованості" згідно з тим, наскільки "добре" відповідне йому розв'язання задачі. В природі це еквівалентно оцінці того, наскільки ефективний організм при конкуренції за ресурси. Найбільш пристосовані особні отримують можливість "відтворити" потомство за допомогою "перехресного схрещування" з іншими особнями популяції. Це призводить до появи нових особинами, які поєднують в собі деякі характеристики, які вони наслідують від батьків. Найменш пристосовані особні з меншою ймовірністю зможуть відтворити нащадків, так що ті властивості, якими вони володіють, поступово зникатимуть з популяції у процесі еволюції. Іноді відбуваються мутації, або спонтанні зміни в генах.

Таким чином, із покоління в покоління, хороші характеристики розповсюджуються по всій популяції. Схрещування найбільш пристосованих особень призводить до того, що досліджуються найбільш перспективні ділянки простору пошуку. В остаточному підсумку популяція буде сходиться до оптимального рішення задачі. Перевага ГА в тому, що він відшукує приблизно оптимальні рішення за відносно короткий час.

Методологічна основа ГА ґрунтується на гіпотезі селекції, що у самому загальному вигляді може бути сформульована так: чим вище пристосованість особи, тим вище ймовірність того, що в потомстві, отриманому з її участю, ознаки, які визначають пристосованість, будуть виражені ще сильніше. Найчастіше особини, що володіють низькою пристосованістю, не беруть участі у генерації нового покоління.

Утім, сказане справедливо не тільки для ГА, а для будь-якого чисельного методу оптимізації. Сама ідея оптимальності прийшла в науку з біології. Однак далеко не завжди ми усвідомлюємо, як багато методичних прийомів оптимального проектування мають корені в селекційній практиці і являють приклад нашого не завжди усвідомленого наслідування природі. Переконатися в справедливості сказаного не важко, якщо спробувати глянути на процедуру чисельної оптимізації через призму гіпотези селекції.

Генетичний алгоритм порівняно новітній, але не єдиний можливий спосіб рішення задач оптимізації. Традиційно обирають два основні шляхи вирішення таких задач – переборний та градієнтний.

**Мета статті.** Метою даної статті є вдосконалення існуючих сучасних методів оптимізації за допомогою саме генетичних алгоритмів, та застосування останніх для вирішення практичних задач. Провести моделювання розглянутих алгоритмів та довести доцільність використання їх для різних вхідних даних.

### Основні матеріали дослідження

Розглянемо класичну задачу комівояжера. Суть задачі полягає у знаходженні короткого шляху проходження всіх міст. При чому, кожне місто можна відвідати тільки один раз. Після цього треба повернутися до точки, з якої починалася подорож.

Генетичний алгоритм уявляє собою комбінацію переборного та градієнтного методів. Механізми кросоверу (схрещування) та мутації реалізують переборну частину, а відбір кращих рішень – градієнтний спуск. Тобто, якщо на деякій множині задана складна функція від декількох змінних, тоді генетичний алгоритм є програмою, яка за зрозумілий час знаходить точку, де значення функції знаходиться достатньо близько до максимально можливого значення. Обираючи прийнятний час розрахунку, отримуємо одне з кращих рішень, які можна отримати за цей час.

Загальна схема роботи генетичного алгоритму для задачі комівояжера, є досить стандартною: генерація популяції, відбір найкращих хромосом, операція схрещування, мутації, формування нової популяції, знов застосування генетичних операторів, представлення оптимального рішення.

Існує багато алгоритмів кросоверу саме для задачі комівояжера, бо класична схема схрещування не може бути застосована. Справа в тому, що для кожної хромосоми популяції існує ряд умов, які є обов'язковими.

Розглянемо один з алгоритмів кросоверу для такої задачі. Нехай кожна хромосома буде представлена деякою послідовністю міст, умовно визначених латинськими літерами. Для операції кросоверу обираються дві хромосоми-батьки. Ці хромосоми можуть обиратися за принципом або найбільш пристосованих хромосом (за допомогою функції пристосованості), або випадково. До речі, функція пристосованості являє собою найменшу суму відстаней в хромосомі в порядку розташування літер зліва направо. Кожна хромосома повинна включати в себе всі міста (літери), які містяться в плані маршруту комівояжера. Кожна літера в одній хромосомі може зустрічатися тільки один раз.

Таким чином, маємо хромосом-батьків,

$$P1=A \overset{|}{B} \overset{|}{E} \overset{|}{D} C$$

$$P2=B \overset{|}{D} \overset{|}{C} \overset{|}{A} E$$

у кожній хромосомі вказуємо дві точки розриву. Але, якщо число міст у хромосомі досить велике, то можна застосовувати й більш ніж дві точки розриву. Число генів батьків та нащадків повинно співпадати, тому хромосома-нащадок матиме 5 позицій для 5 міст. В хромосомі P1 знаходимо позицію, після якої зна-

ходиться перша точка розриву. Це позиція № 2, літера B. Це говорить про те, що формування нащадка почнеться з другої позиції. Далі також, у хромосомі P1 знаходимо всі літери, які знаходяться перед локусами розриву – букви B та E. У нащадка ці букви розташовуються на тих позиціях, які вони мали в хромосомі P1, наприклад

$$offspring = . B E . .$$

Наступним кроком є знаходження в хромосомі P2 літери, що знаходиться після другої точки розриву – буква A. Вона займає в нащадку наступну вільну позицію – № 4. Наступною літерою P2 є буква E, але вона вже присутня в новій хромосомі, тому продовжується пошук таких букв, які ще не були записані в хромосому – нащадок. Точкою зупинки пошуку вважається літера перед другою точкою розриву. В результаті отримаємо нову хромосому-нащадка

$$offspring = C B E A D .$$

Цей вид кросоверу називають впорядкованим, тому що літери P2 записують в тому порядку, в якому вони розташовані у хромосомі.

Для порівняння розглянемо іншу схему алгоритму кросоверу, який має назву реберний кросовер. Хромосоми-батьки можуть бути обрані так само, як і в попередньому алгоритмі. Наступною дією є пошук літери та позиції, з якої буде починатися хромосома-нащадок. З усіх можливих варіантів літер випадково обирається та, яка розпочне формування нащадка. Так само обирається й номер позиції, з якої буде відбуватися складання нащадка. Наприклад, обираємо літеру B, на позицію № 1

$$offspring = B . . . . .$$

Наступним етапом є формування таблиці зв'язків ("сусідів") для кожної літери з P1 та P2. Зв'язаними також вважаються перша і остання позиції хромосоми. В таблицю зв'язків літери записуються не повторюючись (рис. 1, а).

Далі необхідно знайти в таблиці літеру з найменшим числом зв'язків, вона займатиме наступну позицію в хромосомі. Після цього вилучаємо з таблиці ті строки, в яких були описані зв'язки літер, що вже були внесені до хромосоми, включаючи літеру, з якої починалось формування хромосоми (рис.1, б). Ця процедура повторюється доки в таблиці не залишаться літери з рівним числом зв'язків. Вони вносяться в хромосому-нащадок в той послідовності, в якій знаходяться у стовпчику таблиці (рис. 1).

A		BCE
B		AED
C		DA
D		ECB
E		BDA

а

A		BCE
<del>B</del>		<del>AED</del>
<del>C</del>		<del>DA</del>
D		ECB
E		BDA

б

Рис. 1. Таблиця зв'язків

Тоді нащадок має такий вигляд

$$offspring = B C A D E .$$

Що стосується алгоритму мутації, то її алгоритмічна схема дуже проста. В хромосомі – нащадку обирається дві точки розриву. Літера, яка знаходиться після першої точки, та літера, яка знаходиться перед другою точкою міняються місцями. Оператор мутації може бути застосований лише до нащадків. Хромосома для цієї операції обирається випадково.

### Висновки

Для порівняння дієздатності та швидкості цих алгоритмів, вони були запрограмовані на мові програмування C++ в середовищі C++Builder 6.0. Обчислювальну складність алгоритмів було оцінено за таких умов: розмір популяції дорівнював 35 хромосомам, число ітерацій також дорівнювало 35, число міст в одній хромосомі дорівнював 35. Експеримент показав, що алгоритм, де застосовувався впорядкований кросовер працює швидше ніж алгоритм з реберним кросовером. Час роботи першого алгоритму за таких умов дорівнює 921 мс, а час роботи іншого алгоритму дорівнює 1787 мс. Найбільше часу йде саме на формування таблиці зв'язків, тому на достатньо великій кількості міст доцільно використовувати впорядкований кросовер. Крім того, обчислювальна складність для цих алгоритмів пропорційно змінюється в залежності від початкових умов. Тобто, для різного числа міст у хромосомі, різниця в швидкості роботи алгоритмів залишається пропорційною зазначеним вище результатам, Окрім випадків, коли число міст не перевищує 5, в такому разі найкраще

застосовувати класичні методи пошуку оптимального рішення і не користуватися складним апаратом генетичних алгоритмів.

На сьогодні вдосконалення сучасних методів рішення задач оптимізації залишається досить актуальним питанням. За допомогою отриманих результатів можна достатньо успішно знаходити рішення для багатьох практичних задач, які відносяться до цього класу, такими є не тільки згадані транспортні задачі, а й задачі оптимальної побудови топології різноманітних мереж, економічні задачі та інші.

### Список літератури

1. *Genetic Operators, the Fitness Landscape and the Traveling Salesman Problem*, D. Whitley and K. Mathias, *Parallel Problem Solving from Nature-PPSN 2*. R. Manner and B. Manderick, eds., P. 219-228. North Holland-Elsevier, 1992.
2. Schaffer J.D., Whitley D., Eshelman L.J. *Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art//In: Procs. Of the Int. Workshop on Combinations of Genetic Algorithms and Neural Networks (Eds. L.D. Whitley, J.D. Schaffer)*. – Baltimore, Maryland, 1992. – P. 1-37.
3. Люгер Дж. *Искусственный интеллект: стратегии и методы решения сложных проблем*. – М.: Вильямс, 2003. – 340 с.
4. Руденко О.Г., Бодянский Е.В. *Основы теории искусственных нейронных сетей*. – Х.: ТЕЛТЕХ, 2002. – 230 с.

Надійшла до редколегії 13.12.2006

**Рецензент:** д-р техн. наук, проф. С.В. Бодянский, Харківський національний університет радіоелектроніки, Харків.