

ОПТИМИЗАЦИЯ МНОГОЭКСТРЕМАЛЬНЫХ ФУНКЦИЙ БОЛЬШОГО ЧИСЛА ПАРАМЕТРОВ С ИСПОЛЬЗОВАНИЕМ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Для формально сформулированной задачи оптимизации рассмотрены основные достоинства и недостатки классических методов ее решения. Освещены основные предпосылки появления и история развития аппарата генетических алгоритмов. Проанализировано современное состояние библиографии, посвященной вопросам использования поисковых генетических алгоритмов. Рассмотрены основные идеи и принципы, лежащие в основе работы генетических алгоритмов. Выполнен обзор наиболее часто встречаемых генетических операторов: кроссинговера и мутации. Детально проанализированы базовые этапы работы классического генетического алгоритма. Рассмотрены основные способы кодирования решений в хромосомах, выбора пар особей для скрещивания, а так же представлены некоторые распространенные стратегии отбора. Сформулированы основные достоинства бинарного кодирования решений в хромосомах с применением кода Грея, а так же даны рекомендации по использованию вещественного кодирования решений в различных ситуациях. Описаны два основных класса параллельных генетических алгоритмов: «островов» и «мастер-рабочий». Приведен пример использования генетического алгоритма для оптимизации многоэкстремальной функции, зависящей от большого числа параметров. Представлены экспериментальные данные, подтверждающие преимущества использования графических процессоров для вычислений в параллельных генетических алгоритмах. Изложены рекомендации по целесообразности применения генетических алгоритмов в различных ситуациях.

Ключевые слова: оптимизация, генетический алгоритм, мягкие вычисления, генетические операторы, кодирование решений.

НОМЕНКЛАТУРА

BORF – Building and Optimization of Reconstructed Frame;

Pix – пиксель;

PIV – Particle Image Velocimetry;

ГА – генетический алгоритм;

ПГА – параллельный генетический алгоритм;

A – константа;

$[a_i, b_i]$ – диапазон действительных значений, которые может принимать оптимизируемый параметр;

dx – перемещение;

D – ограничения, выделяющие область допустимых решений;

E – функция ошибки;

g_i – целочисленное значение хромосомы;

K – количество разрядов (ген) для кодирования возможных решений в битовых строках (хромосомах);

m' – некоторое решение из области допустимых решений;

m'' – оптимальное решение;

M – пространство решений некоторой задачи;

M' – пространство допустимых решений некоторой задачи;

N – количество поколений в генетическом алгоритме;

P – популяция особей;

P_i – отдельная особь из популяции P ;

$P_{i,j}^k$ – потомок особей P_i и P_j ;

Pm_i – мутированная особь P_i ;

Q – целевая функция;

r_i – вещественное число, соответствующее закодированному в хромосоме g_i решению;

R – критерий оптимизации;

t – время работы генетического алгоритма;

V_x, V_y, V_z – проекции скорости течения на оси x, y, z ;

y – координата точки в потоке, по оси Y ;

x_i, z_i – отдельные гены в хромосомах.

ВВЕДЕНИЕ

На протяжении всей своей эволюции человечество постоянно сталкивалось с различными задачами оптимизации. Поиск кратчайшего пути, оптимальное управление запасами топлива, разработка автоматизированных систем поддержки принятия решений, выбор геометрической формы кузова автомобиля, при которой сопротивление воздуха будет наименьшим – все эти задачи тем или иным образом затрагивают теорию оптимизации. Оптимизация в широком смысле слова находит свое применение не только в науке и технике, но, так же, и в любой другой человеческой деятельности. Еще в XVIII веке великий ученый, блестящий математик и механик Леонард Эйлер сказал: «... в мире не происходит ничего, в чем не был бы виден смысл какого –нибудь максимума или минимума; поэтому нет никакого сомнения, что все явления мира с таким же успехом можно определить из причин конечных при помощи методов максимумов или минимумов, как из самих причин производящих».

Первые математические методы поиска оптимальных решений возникли уже в XVIII веке (вариационное исчисление, численные методы и др.). Однако по-настоящему широкое применение методы оптимизации получили во второй половине XX века. Этому способствовало бурное развитие ЭВМ, без использования которых было крайне сложно реализовать практическое применение математических методов оптимизации.

Одним из перспективных направлений оптимизации являются генетические алгоритмы [1]. Несмотря на то, что впервые о них заговорили более 40 лет назад, они до сих пор продолжают развиваться и находят актуальное практическое применение в самых различных областях науки и техники.

Объектом исследования является процесс решения оптимизационных задач большого числа параметров, основанный на использовании генетических алгоритмов.

Предметом исследования являются операторы, модели, методы и стратегии, используемые в ГА.

Целью данной работы является анализ развития и текущего состояния аппарата генетических алгоритмов, а так же оценка целесообразности использования ГА с бинарным кодированием для поиска глобальных максимумов и минимумов многоэкстремальной функции большого числа параметров при решении различных оптимизационных задач.

1 ПОСТАНОВКА ЗАДАЧИ

Пусть заданы $M, D: M' \subseteq M$, а также целевая функция Q , которая каждому значению $m' \in M'$ ставит в соответствие вещественное число из множества R (критерий оптимизации используется для сравнения альтернативных решений и выбора лучшего из них). Тогда модель задачи оптимизации описывается тройкой: $\langle M, D, Q \rangle$.

Требованием оптимизации является выражение:

$$Q(m') \rightarrow \min \text{ (или } Q(m') \rightarrow \max \text{)}. \quad (1)$$

Решение $m'' \in M'$, удовлетворяющее требованию оптимизации (1), называется оптимальным. Целью оптимизационной задачи является поиск оптимального решения m'' из множества всех допустимых решений. Довольно часто встречаются задачи, требующие оптимизации по нескольким критериям одновременно.

2 ЛИТЕРАТУРНЫЙ ОБЗОР

В связи с высокой распространенностью и важностью решения задач оптимизации в науке и технике, в настоящее время разработано довольно большое количество различных методов и алгоритмов поиска максимумов и минимумов. К ним можно отнести: поиск в ширину [1], метод дихотомии [1], метод градиентного спуска [2], метод золотого сечения [2] и т. д.

Все указанные выше методы оптимизации относятся к классическим. Они позволяют с высокой точностью решать большой класс задач, однако обладают и недостатками. Среди основных недостатков классических методов оптимизации следует отметить следующие:

- при оптимизации нелинейных функций возникает проблема локальных экстремумов, когда за истинное решение принимается не глобальный, а локальный экстремум;
- при увеличении числа одновременно оптимизируемых параметров резко возрастает вычислительная сложность и, как следствие, время вычислений.

Попытки преодоления указанных проблем привели к созданию теории генетических алгоритмов. Генетические алгоритмы – это поисковые алгоритмы, основанные на селекции и генетике [1]. Основателем теории генетических алгоритмов по праву считается американский

исследователь Джон Холланд, который в конце 1960-х годов предложил в качестве принципов комбинаторного перебора вариантов решения оптимизационных задач использовать методы и модели механизма развития органического мира на Земле. В 1975 году Холланд опубликовал свою самую знаменитую работу «Adaptation in Natural and Artificial Systems» [3]. В ней он впервые ввел термин «генетический алгоритм». Ученики Холланда – Кеннет Де Йонг и Дэвид Голдберг – продолжили его работу в области ГА. Наиболее известная работа Голдберга – «Genetic algorithms in search optimization and machine learning» [4].

Следует отметить, что генетические алгоритмы относятся к более общей группе методов, называемых эволюционными вычислениями [5]. В свою очередь эволюционные вычисления относятся к классу неточных, приближенных методов решения, часто называемых «мягкими вычислениями» [6].

Современная библиография по генетическим алгоритмам превышает 9000 наименований и продолжает увеличиваться. Однако работ, посвященных именно комплексному обзору теории ГА, учитывая важность и актуальность данного направления, все еще не хватает.

Одним из лучших русскоязычных современных трудов в области генетических алгоритмов является учебник Гладкова Л. А. [1]. В данном учебнике описаны фундаментальные основы ГА и эволюционного моделирования, представлены основные принципы, стратегии и концепции генетического поиска. Однако в качестве обзорного знакомства с ГА и областью их применения данный труд рассматривать не стоит, в связи с довольно большим объемом материалов, рассматриваемых в нем.

Среди наиболее известных работ в области ГА следует отметить статью Whitley D. A «Genetic algorithm tutorial» [7]. Данная работа носит обзорный характер и ее можно рекомендовать для ознакомительного знакомства с теорией генетических алгоритмов. К недостаткам этой статьи можно лишь отнести отсутствие современных примеров практического применения ГА в различных областях науки и техники (работа вышла более 20 лет назад).

В текущей работе попытаемся совершить краткий комплексный обзор основ теории оптимизации с использованием генетического поиска, и подкрепить теорию современными примерами использования аппарата ГА.

3 МАТЕРИАЛЫ И МЕТОДЫ

ГА базируются на теоретических достижениях синтетической теории эволюции и используют основные принципы эволюционной теории Ч. Дарвина [8]: наследственность, изменчивость и естественный отбор.

ГА оперируют совокупностью особей (популяцией), которые представляют собой строки (хромосомы), кодирующие одно из решений задачи. Этим ГА отличается от большинства других алгоритмов оптимизации, которые оперируют лишь с одним решением, улучшая его.

Каждая особь оценивается мерой ее «приспособленности» согласно тому, насколько «хорошо» соответствующее ей решение задачи. Для этого используется функция приспособленности [1] (целевая функция), которая выделяет наиболее приспособленные решения (в даль-

нейшем они продолжают использоваться) и наихудшие решения (удаляются из популяции и не влияют на дальнейший поиск оптимального решения). Таким образом, мы стремимся повысить приспособленность, а следовательно, и приблизиться к искомому решению.

Работа генетического алгоритма представляет собой итерационный процесс. На каждой новой итерации над текущими особями применяются различные генетические операторы, которые порождают новые особи. После этого все особи оцениваются с помощью целевой функции и наиболее приспособленные используются на следующей итерации ГА. Этот процесс продолжается до тех пор, пока не будут достигнуты желаемые результаты, либо число итераций превысит пороговое значение (пороговое значение числа итераций позволяет ограничить сверху временные рамки работы алгоритма). Так же, одним из признаков того, что следует остановить итерационный процесс, является схождение популяции (состояние популяции, когда все ее особи на протяжении нескольких поколений находятся в области некоторого экстремума и почти одинаковы). Схождение популяции обычно указывает на то, что достигнуто решение близкое к оптимальному. Обычно итоговым решением задачи служит наиболее приспособленная особь последнего поколения.

Генетические операторы являются средством отображения одного множества на другое. Они позволяют применять принципы наследственности и изменчивости к виртуальным популяциям. Все генетические операторы обладают вероятностными характеристиками, что вносит некоторую степень свободы в работу ГА. Наиболее применяемыми генетическими операторами являются оператор кроссинговера (скрещивания) и оператор мутации.

Оператор кроссинговера моделирует процесс скрещивания особей. Этот генетический оператор приводит к созданию новых особей, на основе существующих. Простейшим вариантом скрещивания является односточечный кроссинговер. Перед началом его работы, выбираются две особи P_1, P_2 из популяции P , и случайным образом определяется точка разрыва в их хромосомах. Затем хромосомы обмениваются правыми частями, в результате чего получаются новые возможные решения оптимизационной задачи:

$$P_1 = \{x_1, x_2, x_3, \dots, x_n\}, P_2 = \{z_1, z_2, z_3, \dots, z_n\},$$

$$P_{1,2}^1 = \{x_1, x_2, z_3, \dots, z_n\}, P_{1,2}^2 = \{z_1, z_2, x_3, \dots, x_n\}.$$

Рассмотренный выше оператор скрещивания не является единственным. Существует огромное число операторов кроссинговера, так как их структура в основном и определяет эффективность генетических алгоритмов. Среди наиболее известных операторов скрещивания можно отметить следующие: двухточечный оператор кроссинговера [1], циклический оператор кроссинговера [1], упорядоченный оператор кроссинговера [1], триадный кроссинговер [9], перетасовочный кроссинговер [9] и др.

Следующим, наиболее часто применяемым генетическим оператором является оператор мутации. Оператор мутации необходим для «выбивания» популяции из локального экстремума и способствует защите от преж-

девременной сходимости. Простейший оператор мутации меняет местами два случайно выбранных в хромосоме гена:

$$P_3 = \{x_1, x_2, x_3, x_4\},$$

$$Pm_3 = \{x_1, x_2, x_4, x_3\}.$$

Так же существуют и другие операторы мутации, например двухточечный оператор мутации [1].

Рассмотренные выше генетические операторы являются основными, но не единственными. К часто используемым генетическим операторам можно отнести: оператор инверсии [1], оператор транслокации [1], оператор транспозиции [1], оператор сегрегации [1] и многие другие. В настоящее время продолжают появляться новые генетические операторы.

В целом можно утверждать, что в каждом конкретном случае нужно внимательно подходить к выбору генетических операторов, используемых в ГА, так как это во многом влияет на качество и скорость получения конечного результата.

Как уже отмечалось выше, работа генетического алгоритма представляет собой повторяющийся процесс. Простейший (классический) алгоритм ГА [3] представлен на рис. 1.

В классическом ГА обычно используются два генетических оператора: кроссинговер и мутация. На первом этапе случайным образом создается популяция особей (возможных решений). Даже если все исходные особи будут иметь низкую приспособленность, в результате работы ГА должны появиться хромосомы, кодирующие более точные решения оптимизационной задачи.

Обычно для кодирования решений используется бинарный алфавит. При этом принято использовать специальный прием кодирования оптимизируемого параметра, принимающего действительные значения в некотором диапазоне $[a_i, b_i]$, в битовую строку. Интервал допустимых значений оптимизируемого параметра разбивают на участки с требуемой точностью. Для преобразования целочисленного значения хромосомы g_i из множества $\{0, \dots, 2^K\}$ в вещественное число r_i из интервала пользуются формулой:

$$r_i = g_i \cdot \frac{(b_i - a_i)}{2^K - 1} + a_i.$$

Чаще всего используется $K = 8; 16$.



Рисунок 1 – Основные этапы классического ГА

Бинарное кодирование обеспечивает лучший поиск с помощью гиперплоскостей, так как предоставляет максимальное их количество, при этом для встречаемости каждого символа в каждой позиции требуется меньший размер популяции. Следует отметить, что довольно часто ГА будет работать лучше при кодировании параметров кодом Грея [10], а не прямым бинарным кодом. Это связано с тем, что расстояние Хэмминга [10] между битовыми представлениями данных может и не отражать близость в привычном смысле – например, числа 7 и 8 различаются на 4 бита. Бинарное кодирование добавляет дополнительные разрывы, что осложняет поиск.

Однако в некоторых случаях более целесообразно применять вещественное кодирование (например, для 100 переменных, изменяющихся в интервале $[-500; 500]$, требуется найти минимум с точностью до шестого знака после запятой. В этом случае при использовании ГА с двоичным кодированием длина строки составит 3000 элементов, а пространство поиска – около 10^9 в степени 1000).

Начальный размер популяции так же может влиять на работу ГА. Большие и обладающие достаточным разнообразием представленных решений популяции дают генетическому алгоритму больший шанс скорее найти нужное решение, но вместе с тем требуют и больших вычислительных ресурсов при работе. В статье [11] подробно рассмотрен вопрос выбора размера начальной популяции, необходимой для стабильной работы ГА.

В рассматриваемом классическом ГА используется оператор кроссинговера. Существуют различные подходы к выбору особей для скрещивания. Простейший из них, когда родительские особи выбираются случайно (панмиксия). Однако обычно вероятность скрещивания лучших хромосом с худшими уменьшается на последних итерациях алгоритма. Это приводит к скрещиванию лучших хромосом между собой и закреплению желаемых признаков в хромосомах.

Часто используются и другие методы выбора пар хромосом для оператора кроссинговера. Например «близкое родство» (инбридинг), когда первая хромосома выбирается случайным образом, а вторая хромосома с большей вероятностью будет кодировать близкое решение. Так же к методам выбора особей для скрещивания можно отнести аутбридинг [9], выбор на основе кода Грея [9] и другие.

Одним из важнейших этапов ГА является этап отбора. На данном этапе выбираются особи, над которыми будут выполняться различные генетические операторы: скрещивание, мутация и т.д. Именно эти особи будут оказывать влияние на следующие поколения. В настоящее время известны различные стратегии отбора. Например, ранговый отбор, когда для каждой особи ее вероятность быть отобранной пропорциональна ее порядковому номеру в отсортированной по возрастанию приспособленности популяции. Довольно часто так же применяются турнирный отбор [9], элитный отбор [1], отбор на основе заданной шкалы [1] и др.

Рассмотренный выше классический ГА не является единственным. Существуют и другие модели ГА: *Genitor* [7], *Hybrid algorithm* [7], *CHC* [7] и т.д. В каждом конкретном случае нужно детально анализировать исходные

данные, перед выбором конкретной модели ГА. Иногда поисковые генетические алгоритмы используют совместно с классическими методами оптимизации. В таких случаях ГА позволяет приблизиться к искомым значениям, а классические методы оптимизации (метод градиентного спуска, метод покоординатного спуска и т.д.), в свою очередь, позволяют более точно их определить.

В последнее время с развитием технологий параллельных вычислений, стали появляться и широко использоваться параллельные генетические алгоритмы [1, 12]. ПГА позволяют гораздо эффективней использовать мощности современных вычислительных систем. Параллельные генетические алгоритмы условно делятся на два больших класса: «островов» и «мастер-рабочий» [12].

При использовании схемы «островов» происходит параллельное развитие нескольких независимых популяций. Через некоторое число поколений производится обмен лучшими особями. Корректная реализация генетического алгоритма класса «островов» является довольно сложной нетривиальной задачей.

В схеме «мастер-рабочий» расчет функции приспособленности каждой особи происходит параллельно, при этом реализуется один цикл развития популяций. Особенно выгодно использовать данную схему, если оценка особей вычисляется по сложным формулам.

Обычно эффективность работы различных моделей ГА принято оценивать количеством вычислений целевой функции [13]. Чем меньше потребовалось вычислений целевой функции, при поиске оптимума, тем лучше реализован ГА. Тестовые функции, на которых можно экспериментально оценить конкретную реализацию ГА, должны учитывать сильные стороны генетического алгоритма, то есть быть многоэкстремальными и многомерными. В качестве тестовых функций очень часто используют функции, предложенные одним из студентов Холланда Де Йонгом [9] (в литературе их очень часто называют *dejong1*, *dejong2*, *dejong3*, *dejong4*).

4 ЭКСПЕРИМЕНТЫ

В настоящее время существует огромное количество различных моделей и модификаций генетического алгоритма. Сравнить экспериментально, на тестовых функциях, все их сильные и слабые стороны между собой, не представляется возможным в рамках одной статьи. Поэтому, было решено в разделе эксперименты рассмотреть положительный опыт реального применения ГА для оптимизации многоэкстремальной функции в методе BORF [14], а так же нетривиальную модификацию генетического алгоритма для вычислений с использованием графических процессоров.

Метод BORF относится к классу методов трассерной визуализации PIV [15], которые позволяют экспериментально определять поля скоростей в различных потоках жидкости и газа. Данный метод пытается строить синтетическое реконструированное изображение, имеющее минимальные отличия от цифрового изображения течения, полученного естественным путем. Мерой отличия служит функция ошибки E (принцип ее вычисления детально описан в работе [14]). Функция ошибки E может иметь большое число локальных минимумов. В таком случае, использование классических методов оп-

тимізації (таких як, наприклад, метод градієнтного спуску) може привести к тому, що ми знайдемо локальний, а не глобальний екстремум. Так же функція помилки залежить від більшого числа змінних (в деяких експериментах число параметрів функції E досягло 9000). Тому, одним із способів пошуку мінімуму функції відліччя, застосовуваного в методі BORF, вважається використання генетических алгоритмів. Щоб оцінити цільовість використання ГА проводилися ряд експериментів на тестовому теченні, з різкими градієнтами швидкості, описуваному наступним чином:

$$V_x = A \tanh\left(\frac{y-100}{10}\right) + \frac{y}{450}, V_y = 0, V_z = 0. \quad (2)$$

Для розрахунку профілю швидкості тестового течення методом трасерної візуалізації в методі BORF по черзі використовувалися метод градієнтного спуску і ГА.

Довольно часто використання ГА супроводжується великими часовими витратами, особливо в тих випадках, коли розрахунок цільової функції потребує великих обчислювальних навантажень. В цьому випадку грамотне застосування паралельних обчислень може значно збільшити швидкість розрахунку. В роботі [13] автор провів експеримент, де порівняв час роботи класичного ГА і ПГА, що працює за схемою «мастер-робітник». Генетический пошук використовувався для оптимізації функції відліччя в методі BORF. При цьому ПГА працював на графічних процесорах і використовував для цього технологію Cuda. Основною перевагою використання графічних адаптерів для паралельних обчислень вважається те, що аналогічні за продуктивністю кластери або багатоядерні процесори мають в сотні разів більшу ціну. Для того, щоб проаналізувати ефективність роботи ПГА на графічних процесорах, порівнювалося час роботи генетического пошуку в методі BORF з використанням паралельних технологій і без використання.

5 РЕЗУЛЬТАТИ

На рис. 2. представлені теоретичний профіль течення (2), а так же профілі швидкості, отримані методом BORF з використанням або ГА з бінарним кодуванням рішень, або методу градієнтного спуску.

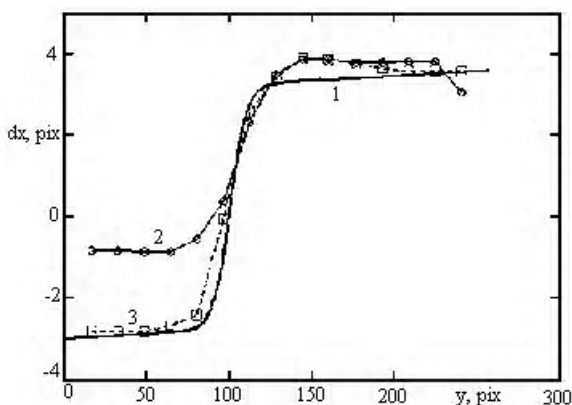


Рисунок 2 – Профілі швидкості досліджуваного течення: 1 – теоретичний профіль; 2 – профіль швидкості, побудований на основі даних, отриманих методом BORF з використанням методу градієнтного спуску; 3 – профіль швидкості, побудований на основі даних, отриманих методом BORF з використанням генетических алгоритмів

Як видно з рис. 2, використання методу градієнтного спуску в методі BORF дає менш точні результати, ніж апарат генетических алгоритмів. Це можна пояснити тим, що функція E має велике число локальних мінімумів, і метод градієнтного спуску знаходить один з таких і зупиняє свою роботу.

На рис. 3 представлений графік залежності часу роботи різних версій ГА від числа поколінь (число особей в кожному поколінні – 100).

Із графіка (рис. 3) видно, що використання технології CUDA при пошуку мінімуму функції відліччя генетическим алгоритмом дозволяє на порядки знизити час розрахунку. При цьому, слід зауважити, що з збільшенням числа поколінь час ГА, що використовує паралельні обчислення, зростає помітно повільніше, ніж час розрахунку генетического алгоритму, що не використовує паралельні обчислення.

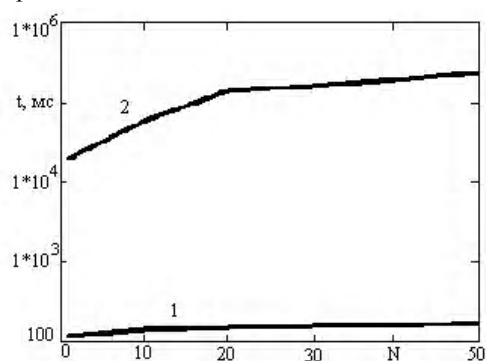


Рисунок 3 – Залежність часу роботи t від числа поколінь N ГА в задачі пошуку мінімуму функції відліччя: 1 – з використанням паралельних обчислень на графічному адаптері; 2 – без використання паралельних обчислень

6 ОБСУЖДЕНИЕ

В настоящее время ГА нашли очень широкое практическое применение абсолютно в разных отраслях науки. Генетические алгоритмы сейчас успешно используются при решении следующих проблем: оптимізація функцій, різноманітні задачі на графах (задача комівояжера, раскраска и т. д.), настройка и обучение искусственной нейронной сети, задачи компоновки, задачи составления расписаний, аппроксимация функций, некоторые задачи в области биоинформатики.

Однако, не следует использовать ГА в случае, когда необходимо найти точный глобальный оптимум. ГА не гарантирует обнаружения глобального решения за приемлемое время. ГА не гарантируют и того, что найденное решение будет оптимальным решением. Тем не менее, генетические алгоритмы применимы для поиска «достаточно хорошего» решения задачи за «достаточно короткое время».

ВЫВОДЫ

В работе проанализированы основные принципы функционирования поисковых генетических алгоритмов, представлены наиболее важные генетические операторы, модели и стратегии, применяемые в ГА. Так же рассмотрен положительный опыт практического применения ГА для оптимізації многоэкстремальных функций.

Новизна выполненного обзорного исследования состоит в комплексном подходе к анализу эффективности решения многопараметрических оптимизационных задач с использованием генетических алгоритмов, который включает характеристику операторов, моделей, методов и стратегий, используемых в ГА. Впервые показано, что классический генетический алгоритм с бинарным кодированием допускает эффективную компьютерную реализацию, учитывающую особенности архитектуры графических процессоров при их использовании в высокопроизводительных параллельных вычислениях.

Практическая значимость результатов работы заключается в систематизации, которая позволяет оценивать эффективность применения генетических алгоритмов при решении конкретных прикладных задач. Определен также доступный и экономичный способ значительного повышения производительности расчетов с использованием генетических алгоритмов на основе программно-аппаратной технологии CUDA, обеспечивающей реализацию параллельных вычислений на графических процессорах.

Одним из перспективных направлений дальнейших исследований являются распределенные генетические алгоритмы, способные эффективно выполняться параллельно на вычислительных устройствах различной архитектуры.

БЛАГОДАРНОСТИ

Работа выполнена в рамках научно-исследовательской темы кафедры информационных технологий Киевской академии водного транспорта «Моделирование сложных объектов и процессов».

СПИСОК ЛИТЕРАТУРЫ

1. Гладков Л. А. Генетические алгоритмы / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик. – М. : ФИЗМАТЛИТ, 2006. – 320 с.
2. Калиткин Н. Н. Численные методы / Н. Н. Калиткин. – М. : Наука, 1978. – 512 с.

3. Holland J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence* / J. H. Holland. – Cambridge : A Bradford Book, 1992. – 211 p.
4. Goldberg D. *Genetic Algorithms in Search, Optimization, and Machine Learning* / D. Goldberg. – Boston : Addison-Wesley Professional, 1989. – 432 p.
5. Скобцов Ю. А. Основы эволюционных вычислений / Ю. А. Скобцов. – Донецк : ДонНТУ, 2008. – 326 с.
6. Soft computing: new trends and application / [L. Fortuna, G. Rizzotto, M. Lavorgna, G. Nunnari and other]. – Heidelberg : Springer, 2001. – 267 p.
7. Whitley D. A genetic algorithm tutorial / D. Whitley // *Statistics and Computing*. – 1994. – Vol. 4. – P. 65–85. DOI: 10.1007/BF00175354.
8. Дарвин Ч. Происхождение видов путем естественного отбора / Ч. Дарвин. – СПб. : Наука, 2001. – 568 с.
9. Панченко Т. В. Генетические алгоритмы / Т. В. Панченко. – Астрахань : Астраханский университет, 2007. – 87 с.
10. Хэмминг Р. В. Теория кодирования и теория информации / Р. В. Хэмминг. – М. : Радио и связь, 1983. – 176 с.
11. Выбор размера популяции для генетического алгоритма [Электронный документ]. – Режим доступа: <http://habrahabr.ru/post/114910/>. – 25.12.2014 г. – Загл. с экрана.
12. Иванов Д. Е. Масштабируемый параллельный генетический алгоритм построения идентифицирующих последовательностей для современных многоядерных вычислительных систем / Д. Е. Иванов // *Управляющие системы и машины*. – Киев, 2011. – № 1. – С. 25–32.
13. Мочалин А. Е. Использование параллельных генетических алгоритмов в методе обработки трассерных изображений BORF / А. Е. Мочалин // *Вісник черкаського університету*. – Черкаси, 2013. – № 38(291). – С. 58–64.
14. Мочалин А. Е. Информационная технология обработки цифровых трассерных изображений на основе построения реконструированного кадра: дис. канд. техн. наук. 05.13.06. – Алчевск, 2013. – 177 с.
15. Adrian R. J. Twenty years of particle image velocimetry / R. J. Adrian // *Exp. in Fluids*. – 2005. – Vol. 39, No. 2. – P. 159–169. DOI: 10.1007/s00348-005-0991-7.

Статья поступила в редакцию 13.01.2015.

После доработки 23.02.2015.

Мочалин О. Є.

Канд. техн. наук, доцент кафедри інформаційних технологій Київської державної академії водного транспорту, Київ, Україна

ОПТИМІЗАЦІЯ БАГАТОЕКСТРЕМАЛЬНИХ ФУНКЦІЙ ВЕЛОКОЇ КІЛЬКОСТІ ПАРАМЕТРІВ З ВИКОРИСТАННЯМ ГЕНЕТИЧНИХ АЛГОРИТМІВ

Формально сформульована задача оптимізації. Розглянуті основні переваги і недоліки класичних методів оптимізації. Висвітлені основні передумови появи та історія розвитку апарату генетичних алгоритмів. Проаналізовано сучасний стан бібліографії, присвяченої питанням використання пошукових генетичних алгоритмів. Розглянуто основні ідеї та принципи, що лежать в основі роботи генетичних алгоритмів. Виконано огляд найбільш поширених генетичних операторів: кросингверу і мутації. Детально проаналізовані базові етапи роботи класичного генетичного алгоритму. Розглянуті основні способи кодування рішень в хромосомах, вибору пар особин для схрещування, а так само представлені деякі поширені стратегії відбору. Сформульовані основні переваги бінарного кодування рішень в хромосомах із застосуванням коду Грея, а так само дані рекомендації з використання дійсного кодування рішень в різних ситуаціях. Описані два основних класи паралельних генетичних алгоритмів: «острівів» і «майстер-робітник». Наведено приклад використання генетичного алгоритму для оптимізації багатоекстремальної функції, що залежить від великої кількості параметрів. Представлені експериментальні дані, що підтверджують переваги використання графічних процесорів для обчислень в паралельних генетичних алгоритмах. Викладені рекомендації щодо доцільності застосування генетичних алгоритмів у різних ситуаціях.

Ключові слова: оптимізація, генетичний алгоритм, м'які обчислення, генетичні оператори, кодування рішень.

Mochalin O. Ye.

PhD, Associate professor of department of information technology, Kyiv State Maritime Academy named after hetman Petro Konashevich-Sahaydachniy, Kyiv, Ukraine

GENETIC ALGORITHMS IN OPTIMIZATION OF MULTIEXTREMUM FUNCTIONS WITH LARGE PARAMETERS NUMBER

An optimization problem is formally formulated. The main advantages and disadvantages of classical optimization methods are considered for this problem. Basic prerequisites for the emergence and history development of the instrument of genetic algorithms are highlighted. The current state of the bibliography which is dedicated to the use of search genetic algorithms is analyzed. The basic ideas and underlying principles

of genetic algorithms functioning are considered. A review of the most commonly used genetic operators: crossover and mutation, is made. The basic steps of classical genetic algorithm operation are analyzed in detail. The coding problem of solutions in the chromosomes and the selection of individual pairs for crossbreeding are considered. Some common selection strategies are presented as well. The basic benefits of the binary coding of solutions in the chromosomes that using Gray code are formulated. The recommendations are also given on the using of solutions real coding in different situations. The two main classes of parallel genetic algorithms: «islands» and «master – slave», are described. The example of using a genetic algorithm to optimize multiextremal function depending on a large number of parameters is showed. Experimental data are presented which confirm the benefits of graphic processors using in parallel implementation of genetic algorithm. The expediency recommendations of the use of genetic algorithms in different situations are set out.

Keywords: optimization, genetic algorithm, Soft Computing, genetic operators, coding of solutions.

REFERENCES

1. Gladkov L. A., Kurejchik V. V., Kurejchik V. M. *Geneticheskie algoritmy*. Moscow, FIZMATLIT, 2006, 320 p.
2. Kalitkin N. N. *Chislennye metody*. Moscow, Nauka, 1978, 512 p.
3. Holland J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, A Bradford Book, 1992, 211 p.
4. Goldberg D. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, Addison-Wesley Professional, 1989, 432 p.
5. Skobcov Ju. A. *Osnovy jevoljucionnyh vychislenij*. Doneck: DonNTU, 2008, 326 p.
6. Fortuna L., Rizzotto G., Lavorgna M., Nunnari G., Xibilia M. G., Caponetto R. *Soft computing: new trends and application*. Heidelberg, Springer, 2001, 267 p.
7. Whitley D. A genetic algorithm tutorial, *Statistics and Computing*, 1994, Vol. 4, pp. 65–85. DOI: 10.1007/BF00175354.
8. Darwin Ch. *Proishozhdenie vidov putem estestvennogo otbora*. Sankt-Peterburg, Nauka, 2001, 568 p.
9. Panchenko T. V. *Geneticheskie algoritmy*. Astrahan', Astrahanskij universitet, 2007, 87 p.
10. Hjemming R. V. *Teoriya kodirovanija i teoriya informacii*. Moscow, Radio i svjaz', 1983, 176 p.
11. *Vybor razmera populjaccii dlja geneticheskogo algoritma* [Elektronnyj dokument]. Rezhim dostupa: <http://habrahabr.ru/post/114910/>. 25.12.2014 g. Zagl. s jekrana.
12. Ivanov D. E. *Masshtabiruemyj paralel'nyj geneticheskij algoritm postroenija identificirujushhh posledovatel'nostej dlja sovremennyh mnogojadernyh vychislitel'nyh sistem, Upravljajushhie sistemy i mashiny*. Kiev, 2011, No. 1, pp. 25–32.
13. Mochalin A. E. *Ispol'zovanie paralel'nyh geneticheskikh algoritmov v metode obrabotke trassernyh izobrazhenij BORG*, *Visnik cherkas'kogo universitetu*. Cherkasi, 2013, No. 38(291), pp. 58–64.
14. Mochalin A. E. *Informacionnaja tehnologija obrabotki cifrovych trassernyh izobrazhenij na osnove postroenija rekonstruirovannogo kadra*: dis. kand. tehn. nauk. 05.13.06. Alchevsk, 2013, 177 p.
15. Adrian R. J. *Twenty years of particle image velocimetry*, *Exp. in Fluids*, 2005, Vol. 39, No. 2, pp. 159–169. DOI: 10.1007/s00348-005-0991-7.