Опубликовано в материалах 2-й межвузовской научной конференции по проблемам информатики СПИСОК-2011, с. 336-338.

М. В. Буздалов
Санкт-Петербургский государственный университет информационных технологий, механики и оптики

Генерация тестов для олимпиадных задач по программированию с использованием эволюционных стратегий

Введение

В мире проводится большое число олимпиад по программированию. Они способствуют выявлению талантливых программистов среди школьников и студентов. Среди них можно отметить международную студенческую олимпиаду по программированию International Collegiate Programming Contest [1], проводимую Association for Computing Machinery, с развитой сетью отборочных соревнований, международную олимпиаду школьников по информатике [2], соревнования, проводимые компанией TopCoder [3], интернет-олимпиады по информатике и программированию [4] и многие другие.

На олимпиадах по программированию предлагается решить одну или несколько задач. Решением задачи является программа, написанная на одном из разрешенных языков программирования.

Программа тестируется на наборе тестов, неизвестных участникам. На работу программы накладываются определенные ограничения, такие как

максимальное время выполнения и максимальный объем используемой памяти.

Решение считается прошедшим определенный тест, если оно при работе с ним не нарушило ограничений, завершилось без ошибок, и его ответ признан правильным. О конкретных видах задач и ограничениях можно прочитать, например, на сайте олимпиады [5].

Цель работы

Создание тестов для олимпиадных задач является сложным творческим процессом. В большинстве случаев, тесты создаются вручную или с помощью программ, генерирующих тесты по некоторому шаблону. Для некоторых задач такой способ генерации тестов приводит к тому, что набор тестов оказывается слабым, и в результате засчитывается множество неэффективных решений.

Цель работы – генерация тестов, на которых неэффективные решения работают как можно дольше. Качество таких тестов может выражаться количественно, что позволяет использовать для поиска качественных тестов методы оптимизации. В качестве метода оптимизации в работе используется (1+1)-эволюционная стратегия [6].

Метод генерации тестов рассматривается на примере задачи “Work for Robots” [7], размещенной на сервере Timus Online Judge [8].

Описание предлагаемого подхода

Для генерации теста выбирается решение, против которого нужно сгенерировать этот тест. При успешной генерации теста, им может быть «покрыт» целый класс решений, таким образом, при разумном выборе решений, можно сгенерировать достаточно полный набор тестов.

Исходный код решения модифицируется, чтобы, помимо ответа на решаемую задачу, решение вычисляло функцию приспособленности теста. Функция приспособленности может быть как числом, так и более сложным объектом, например, вектором чисел, который может сравниваться с другим таким вектором лексикографически. Такой подход обладает большой гибкостью, так как позволяет описывать достаточно разнообразные функции приспособленности, которые, в свою очередь, могут ускорить процесс нахождения оптимума.

Для генерации теста используется (1+1)-эволюционная стратегия, особью которой является тест, а функция приспособленности этого теста вычисляется при запуске на нем модифицированного решения. Для задачи “Work for Robots”, используемой для демонстрации подхода, тест задается симметричной матрицей булевых значений размером 50×50 – матрицей смежности графа, который содержится в тесте.

Используется два оператора мутации. Первый из них изменяет содержимое случайно выбранной ячейки матрицы (и симметричной ей ячейки) на противоположное, второй изменяет 10, 100 или 1000 таких ячеек. Операторы мутации применяются по очереди.

Результаты

До применения описанного подхода, для задачи “Work for Robots” было засчитано 86 решений. Тесты генерировались против восьми из этих решений. По итогам генерации было выбрано пять тестов, которые оказались трудными для семи из восьми выбранных решений. Эти тесты были добавлены в набор тестов на сервер Timus Online Judge. 45 из имевшихся решений не прошли новые

тесты. Этот результат показывает высокую эффективность описанного метода.

Литература

1. ACM International Collegiate Programming Contest. http://en.wikipedia.org/wiki/ACM_ICPC.
2. International Olympiad in Informatics. http://www.ioinformatics.org.
3. TopCoder. http://www.topcoder.com/tc.
4. Интернет-олимпиады по информатике. http://neerc.ifmo.ru/school/io/.
5. Правила проведения полуфинала NEERC. http://neerc.ifmo.ru/information/contest-rules.html.
6. *Bäck T., Hoffmeister F., Schwefel H.-P.* A Survey of Evolution Strategies. In Proceedings of the Fourth International Conference on Genetic Algorithms. 1991. pp. 2–9.
7. Задача “Work for Robots”. http://acm.timus.ru/problem.aspx?num=1695
8. Timus Online Judge. Архив задач с проверяющей системой. http://acm.timus.ru