

Московский государственный открытый университет
Москва, ул. П.Корчагина, 22

При построении автоматизированных систем управления часто возникает проблема выделения значимых параметров из общего потока информации, поступающего на вход системы. Алгоритм соответствующего преобразования входной информации не всегда очевиден, а иногда его построение может представлять большие трудности – например, если входные данные слабо структурированы, стандартные и теоретически обоснованные методы отсутствуют или приводят к недостаточно эффективным алгоритмам.

Одним из выходов в этой ситуации может быть применение эволюционных алгоритмов для нахождения требуемого преобразования. Всё множество потенциально возможных преобразований $X = \{x_1, x_2, \dots, x_N\}$ рассматривается как пространство поиска. Каждому преобразованию x_i приписывается показатель качества $F(x_i)$, отражающий степень его пригодности для решения поставленной задачи. Таким образом, задача синтеза преобразования сводится к задаче оптимизации целевой функции F на пространстве X .

Для решения описанной оптимизационной задачи целесообразно применять генетические алгоритмы, основной идеей которых является имитация эволюционных процессов в живой природе. При этом каждое решение (в нашем случае – каждое преобразование) x_i однозначно представляется набором элементарных количественных показателей (генов): $x_i = (x_{i1}, x_{i2}, \dots, x_{iM})$. Над случайно выбранным исходным множеством (популяцией) решений (особей) в определённом порядке проводятся операции скрещивания, мутации и отбора. В результате может быть найдена лучшая особь, представляющая собой приближённое решение оптимизационной задачи.

Основные преимущества генетических алгоритмов перед традиционными градиентными методами поиска и методами перебора состоят в том, что, в отличие от первых, не требуется линейности пространства поиска; в отличие от вторых, количество вычислений, как правило, гораздо меньше. Кроме того, не требуется априорного знания общего вида решения. Эти преимущества особенно важны в поставленных выше условиях, когда вид искомого преобразования может быть неизвестен.

Описанный подход был применён к задаче определения одного из технологических параметров по видеоинформации, поступающей в автоматизированную систему управления технологическим процессом. Видеоизображение смеси твёрдых частиц различных размеров и формы поступает на вход системы. Требуется определить средний размер (диаметр) частиц, отображаемых в кадре. Изображение снимается светочувствительной матрицей размером $L \times L$ пикселей. Глубина цветопередачи – 1 бит, то есть изображение чёрно-белое. Количество двоичных разрядов в определяемом диаметре ограничено ошибкой дискретизации изображения и равно $K = \log_2 L$. Дискретизированное изображение содержит $M = L^2$ бит. Таким образом, требуется построить преобразование, входом которого является двоичный вектор из M бит изображения, а выходом – K бит, представляющих двоичные разряды диаметра.

Такое преобразование можно рассматривать как совокупность K булевых функций M переменных. Поскольку о виде преобразования ничего не известно, его можно представить таблицами истинности этих функций, то есть двоичным вектором длиной $2^M K$.

Для оценки качества каждого преобразования x_i применяется тестовая последовательность изображений, для каждого из которых вычисляемый параметр (диаметр) заранее известен. Значение целевой функции $F(x_i)$ равно средней ошибке определения диаметра – разности между значением, вычисляемым преобразователем по предъявленному изображению, и точным значением.

Существует $T = 2^{2^M K} \approx 2^{2^{(2^2 K)}}_K$ возможных преобразований. При такой мощности пространства поиска методы перебора, очевидно, не могут быть эффективны уже для небольших значений K . Если считать, что основное время затрачивается на вычисление целевой функции $F(x_i)$, то задача перебора в данном случае вообще является НП-полной относительно K .

Для решения этой задачи был применён генетический алгоритм. В результате работы алгоритма было найдено преобразование, определяющее диаметр частиц со средней погрешностью около 3 пикселей (при размере всего изображения 32 x 32 пикселей). При этом целевая функция вычислялась всего около 70000 раз, что более чем в $2^{2^{1000}}$ раз меньше, чем для полного перебора. Преобразование получено в виде таблицы истинности и допускает не только программную, но и аппаратную реализацию. При проведении эксперимента использовалась программная библиотека GAlib (автор – М. Уолл, Массачусетский Технологический Институт).

Результаты проведённого эксперимента позволяют сделать следующие выводы:

- 1) задача синтеза преобразования двумерного изображения может быть сведена к задаче оптимизации;
- 2) для приближённого решения такой оптимизационной задачи целесообразно применение генетических алгоритмов.

Литература:

- Сиппер М., Краткое введение в генетические алгоритмы, 1996;
Даджион Д., Мерсеро М., Преждевременная сходимость в задаче составления расписания, 1996;
Бисли Д., Булл Д. Р., Мартин Р.Р., Обзор генетических алгоритмов, 1993;



USING GENETIC ALGORITHM FOR ANALYSIS OF A TWO-DIMENSIONAL IMAGE

Averyanov T.

Moscow State Open University
P. Korchagina st. 22, Moscow, Russia

In control system design a problem often arises of extracting important parameters from the input data. A corresponding transformation is not always obvious. To find it is difficult sometimes – for example, when input data is weakly structured or when standard methods are lacking or lead to inefficient algorithms. In other words, there are cases when the general form of the transformation can not be easily seen.

One of outlets is using evolutionary algorithms to find the transformation needed. All possible transformations can be considered as a search space. Every transformation is given a quality coefficient. Then the most suitable transformation is sought. In this way, the initial task of design is reduced to a conventional optimization problem, the quality coefficient being an objective function. To solve such optimization problem it is appropriate to employ genetic algorithms. As contrasted to conventional gradient methods, they do not require structured search space, and, compared to Monte-Carlo methods, they tend to be faster. These advantages make genetic algorithms most suitable for the problem put by.

This approach has been used for the following task. A camera-recorder is continuously recording an image of a conveyer belt, on which a mixture of solid objects of various shape and size lays. The image is then sampled in $L \times L$ pixels. Color depth is 1 bit. The goal is to find a way of calculating the average size of those objects (diameter). The diameter value shall be measured in pixels and shall have K bits.

The image can be represented by a binary vector of length L^2 . The total amount of all possible transformations mapping L^2 input bits onto K output bits is $2^{2^{(L^2)}K}$. Among all those transformations one (or more) may exist that carries out the task – calculating diameter based on the discretized image. A steady-state genetic algorithm had been used for finding the suitable transformations. The quality of any transformation (objective function), was estimated by an average error – the difference between the calculated diameter and the accurate value of that. The averaging had been conducted over a test set of images, which were sequentially exposed to the transformation.

As a result of the experiment, a transformation had been found which appeared to be suitable enough for the application. It was able of recognizing the diameter with an average error of 3 pixels (the total size of the image was 32×32 pixels). During the optimization the objective function was calculated about 70000 times, what is incomparably few as against exhaustive search. The software for this work used the GALib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology.

The following conclusion can be drawn from the experiment conducted: the task of the transformation synthesis can be often reduced to an optimization problem (the search of the suitable one among all possible ones) and then solved using genetic algorithms, which appear to be very appropriate for that purpose.

References:

- D. Beasley, D. Bull, R. Martin. An Overview of Genetic Algorithms. 1993.
D. Dudgeon, M. Mersereau. Multidimensional Digital Signal Processing. 1984.
M. Sipper. A Brief Introduction In Genetic Algorithms. 1996.