

ИНФОРМАТИКА

*Вестник Сыктывкарского университета.
Серия 1: Математика. Механика. Информатика.
Выпуск 1 (26). 2018*

УДК 517.443, 519.688

**ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ
ОТЫСКАНИЯ ОПТИМАЛЬНОЙ
ПОСЛЕДОВАТЕЛЬНОСТИ РАСКРОЯ**

В. А. Мельников

Статья описывает применение генетических алгоритмов для отыскания оптимальной последовательности раскроя фигур общего вида в двумерном пространстве. Также приводится модификация генетических алгоритмов для возможности увеличения количества генов в индивиде.

Ключевые слова: генетические алгоритмы, оптимизация, геном, индивиды.

1. Введение

Задача раскроя фигур общего вида в двумерном пространстве не решается классическими методами, применяемыми для решения задач оптимизации, основным способом отыскания оптимальной последовательности раскроя является генетический алгоритм. Также в процессе оптимизации постепенно увеличивается количество свободного места, если сохранять размер первичного индивида, то свободное место останется неиспользованным, несмотря на неразмещившиеся на листе фигуры из раскройного плана.

2. Генетические алгоритмы

Таким же образом, как нейронные сети пытаются имитировать мощь мозга, эволюционные алгоритмы (ЭА) воспроизводят биологическую эволюцию. Например, природа создала сложную структуру человеческого мозга из простейших элементов, пользуясь лишь мощью эволюции и естественного отбора. Так же как естественный отбор действует

на популяции животных, позволяя лучшим выжить и передать свои гены потомкам, так же и ЭА действует, например, на последовательности раскроя, нейронные сети, электрические контуры и так далее. Они определяют качество каждого элемента в популяции, позволяя лучшим выжить и убивая слабейших. Далее — очень важный шаг — они позволяют различным решениям «скрещиваться» между собой, порождая, возможно, более жизнеспособные решения.

Генетический алгоритм (ГА) — самый популярный из эволюционных алгоритмов. Он был изобретён Джоном Холландом в Университете Мичигана в 1975 году. По изначальной задумке, ГА использовал только бинарные числа, но для извлечения большей пользы будем пользоваться десятичными числами. Сам алгоритм состоит из нескольких частей [1]:

1. Кодирование проблемы в виде генов.
2. Выведение первоначального поколения.
3. Вычисление оценок для индивидов.
4. Скрещивание.
5. Мутация.

Рассмотрим ГА применительно к задаче раскроя [2].

2.1. Кодирование задачи

Основываясь на предложенных ранее методах представления и перемещения фигур, можно заметить, что не имеет смысла включать в геном координаты положения и угол поворота так, как они будут вычислены при расположении последовательности. Достаточно кодировать лишь саму последовательность раскроя. Отметим, что для достижения результата, в геноме индивида не должно быть повторов, соответственно, каждый ген должен быть уникален. Таким образом индивидом будет являться последовательность просто раскроя, ведь углы постановки и координаты будут восстановлены при расположении, а геном — порядковый номер фигуры в исходном наборе, это позволит избежать повторов.

2.2. Инициализация первого поколения

Первое поколение индивидов оказывает огромное влияние на результат всего ГА. Чтобы получить неплохой первичный набор, можно расположить по порядку отсортированные по площади фигуры, а остальных индивидов первого поколения получить с помощью мутации.

2.3. Оценка индивидов

Оценку результата можно проводить различными способами. Можно, к примеру, ввести некоторую оценочную функцию. Рассматривая задачу раскроя, за оценку можно взять занятую фигурами площадь. Далеко необязательно использовать какой-то один фактор, например, если индивид I характеризуется парой (n, h) , где n — число размещённых фигур, а h — высота раскроя, то можно взять за оценку данную пару. Тогда то, что индивид I_1 лучше, чем индивид I_2 , можно записать через следующее отношение ρ :

$$I_1 \rho I_2 = (n_1 > n_2 \vee (n_1 = n_2 \wedge h_1 < h_2)). \quad (1)$$

То же самое можно записать с помощью следующей функции оценки индивидов:

$$f(I) = n + h^{-1}, \quad (2)$$

тогда достаточно сравнивать числовые значения функций. Выбранная функция будет принимать большее значение при лучшем варианте раскроя. Вообще говоря, можно использовать любую функцию исходя из условий задачи.

2.4. Скрещивание

Рассмотрим скрещивание на следующем примере:

1. Выберем двух индивидов, например $[5 \ 2 \ 3 \ 7 \ 6 \ 1 \ 4]$ и $[4 \ 6 \ 2 \ 1 \ 3 \ 5 \ 7]$.
2. Случайным образом выберем часть первого родителя, которая перейдёт в потомка: $[5 \ 2 \ (3 \ 7) \ 6 \ 1 \ 4]$.
3. Удалим эти элементы из второго родителя: $[4 \ 6 \ 2 \ 1 \ (3) \ 5 \ (7)]$.
4. Первого потомка получим путём копирования генов второго родителя в первого: $[4 \ 6 \ 3 \ 7 \ 2 \ 1 \ 5]$.
5. Повторим действия 1—3, поменяв родителей местами.
6. Второго потомка получим путём копирования генов первого родителя во второго: $[5 \ 3 \ 2 \ 1 \ 7 \ 6 \ 4]$.

Известно, что в области определения функция может иметь не только абсолютные экстремумы, но и локальные [4], поэтому в процессе скрещивания стоит использовать не только самых лучших индивидов, но и достаточно хороших в целях избежания попадания на точки локального минимума. Схождение к локальному экстремуму называется

преждевременной сходимостью или сходимостью к квазиоптимальному решению [3].

На рис. 1 для некоторой функции $f : X \rightarrow Y$ продемонстрированы точки квазиоптимального и оптимального решения. На рисунке цифра 1 — квазиоптимальное решение, а 2 — оптимальное.

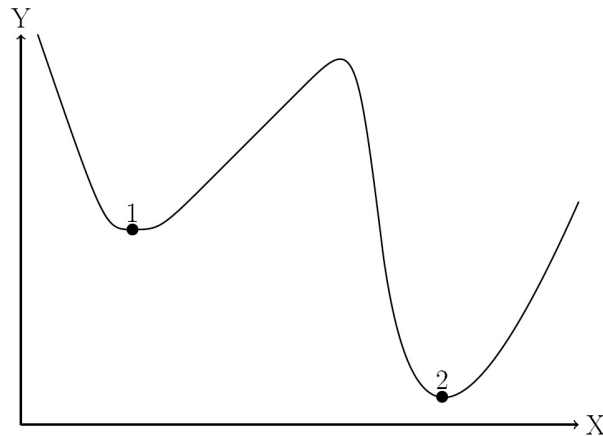


Рис. 1. Преждевременная сходимость

2.5. Мутация

Мутация происходит после скрещивания и применяется к новым индивидам, на самом деле можно применять и к появившимся ранее. В случае раскрытия в процессе мутации меняются местами два гена в последовательности. Вероятность мутации должна быть не очень высокой, приблизительно 10 %, также можно изменять экспериментально. Иногда можно переставлять не просто два гена, а целые части последовательности, но не стоит делать этого слишком часто.

2.6. Отбор

Теперь, когда для каждого индивида вычислена оценка, можно провести отбор для создания нового поколения. Самый простой вариант — взять только те индивиды, у которых достаточно хорошая оценка. В классическом варианте генетического алгоритма индивиды в новом поколении выбираются случайно, а вероятность попадания индивида в новое поколение будет пропорциональна его оценке. Например,

$$p_i = \frac{f_i}{\sum f_k} \quad (3)$$

На самом деле включать в новое поколение можно не только те индивиды, которые были выведены на данном шаге, но и те, которые уже

были ранее, ведь они могут быть не хуже, чем те, что только что появились.

Линейная зависимость от n , как в формуле (2), позволяет учитывать число генов индивида как основной фактор оценки, а высоту как дополнительный фактор для сравнения индивидов с одинаковым числом генов. Таким образом можно отбирать набор индивидов с одинаковым числом генов, отсортированных в порядке убывания высоты. Если в какой-то момент удалось увеличить геном, то отбираются только индивиды с увеличенным набором генов.

3. Численный эксперимент

Прежде чем переходить к рассмотрению входных данных, следует отметить, что для удобства работы все расчёты в программе производятся в пикселях (пкс), $1 \text{ мм} = 3,543307 \text{ пкс}$ [5].

Для проверки работоспособности и корректности алгоритма были проведены численные эксперименты на двух различных наборах данных. В обоих наборах заданы нулевое масштабирование и нулевая граница. Рассмотрим постановку задачи для первого набора данных. Имеется входной лист 1000×1000 пкс, за 10 итераций необходимо отыскать наиболее оптимальное расположение следующего набора фигур на листе:

1. «Олень» — 5 штук, с шагом 15 градусов.
2. «Обезьянка» — 5 штук, с шагом 15 градусов.
3. «Голубь» — 5 штук, с шагом 15 градусов.
4. «Лошадка» — 5 штук, с шагом 15 градусов.
5. «Ангелок» — 5 штук, с шагом 15 градусов.

Первичное решение строилось с помощью расположения фигур по убыванию площади, в него вошло 14 фигур, занятая высота на листе составила 947,115 пкс. В результате работы генетического алгоритма из первичного индивида был получен индивид, содержащий 15 фигур с занятой высотой на листе 948,115 пкс. Увеличение высоты в данном случае нельзя считать ухудшением результата, так как была расположена дополнительная фигура. Остаток из 10 фигур полностью расположился на втором раскройном листе. Высота первичного индивида для остатка фигур составила 982,065 пкс, в результате было достигнуто улучшение до 930,909 пкс. Результат работы для данного набора приведён на рис. 2.

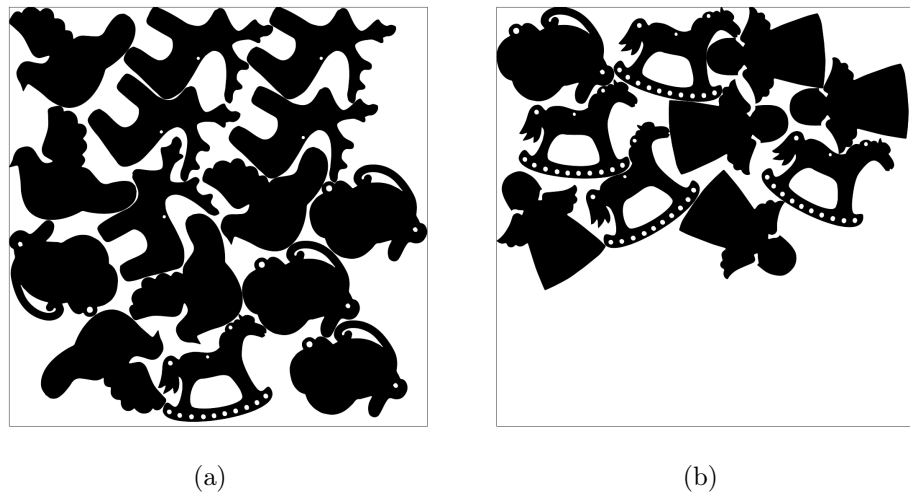


Рис. 2. Первый и второй раскройные листы первого набора фигур

Теперь рассмотрим постановку задачи для второго набора данных. Имеется входной лист 1500×1500 пкс, за 10 итераций необходимо отыскать наиболее оптимальное расположение следующего набора фигур на листе:

1. «Кольцо» — 10 штук, с шагом 15 градусов.
2. «Рамка» — 10 штук, с шагом 15 градусов.
3. «Треугольник» — 10 штук, с шагом 15 градусов.

Первичное решение так же, как и для первого набора, строилось с помощью расположения фигур по убыванию площади, в него вошли все 3 фигуры, занятая высота на листе составила 1388, 258 пкс. В результате работы алгоритма улучшения по высоте достигнуто не было, но данный набор демонстрирует важную возможность алгоритма — расположение фигуры в фигуре. Результат работы приведён на рис. 3.

4. Заключение

Основываясь на алгоритмах и результатах, описанных в статье [6], в процессе дальнейшего исследования было реализовано API на языке программирования Go для разработки систем автоматизированного раскроя.

Следует отметить чувствительность генетического алгоритма к первичному индивиду, даже различная сортировка фигур по площади приведёт к получению двух сильно отличающихся решений. Но, в отличие

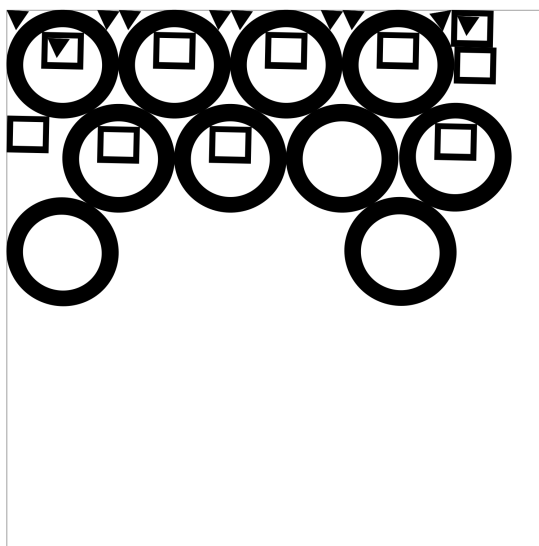


Рис. 3. Раскройный лист второго набора фигур

от человека, одну и ту же последовательность один алгоритм будет кроить одинаково. Для получения различных вариантов раскроя одной последовательности необходимо придумывать различные функции расположения и оценки полученного расположения.

Список литературы

1. MacLeod С. An Introduction to Practical Neural Networks and Genetic Algorithms For Engineers and Scientists. P. 85.
2. He Y., Liu H. Algorithm for 2D irregular-shaped nesting problem based on the NFP algorithm and lowest gravity-center principle // *Journal of Zhejiang University*. 2006. № 7. Pp. 571–574.
3. Панченко Т. В. Генетические алгоритмы / под ред. Ю. Ю. Тарасевича. Астрахань: Издательский дом «Астраханский университет», 2007. С. 16.
4. Кудрявцев Л. Д. Математический анализ. 2-е изд. М.: Высшая школа, 1973. Т. 1. 687 с.

5. Coordinate Systems, Transformations and Units [Electronic resource] / W3C. 6 мая 2017. URL: <https://www.w3.org/TR/SVG/coords.html> (дата обращения: 25.12.2017).
6. **Мельников В. А.** Методы представления фигур общего вида для задачи двумерного раскроя // *Вестник Сыктывкарского университета. Сер. 1: Математика. Механика. Информатика. 2017. Вып. 3 (24). С. 11–24.*

Summary

Melnikov V. A. Application of genetic algorithms for finding the optimal nesting sequence

This article describes usage of genetic algorithms for finding optimal sequence of nesting figures of general kind in two-dimensional space. Also the modification of algorithm for the opportunity of increasing number of gens in the individual.

Keywords: genetic algorithms, optimization, genom, individuals.

References

1. **MacLeod C.** *An Introduction to Practical Neural Networks and Genetic Algorithms For Engineers and Scientists*, p. 85.
2. **He Y., Liu H.** Algorithm for 2D irregular-shaped nesting problem based on the NFP algorithm and lowest gravity-center principle, *Journal of Zhejiang University*, 2006, № 7, pp. 571–574.
3. **Panchenko T. V.** *Geneticheskie algoritmy* (Genetic algorithms), pod red. U. U. Tarasevicha, Astrahan: Izdatelskiy dom «Astrahanskiy universitet», 2007, p. 16.
4. **Kudryavcev L. D.** *Matematicheskiy analiz* (Mathematical analysis), 2-e izd, M.: Vyshaya shkola, 1973, v. 1, 687 p.
5. Coordinate Systems, Transformations and Units [Electronic resource] / W3C. 6 мая 2017. URL: <https://www.w3.org/TR/SVG/coords.html> (date of the application: 25.12.2017).
6. **Melnikov V. A.** Metody predstavleniya figur obshchego vida dlya zadachi dvumernogo raskroya (Methods for representing figures of general kind for a two-dimensional cutting problem), *Bulletin of Syktyvkar University, Series 1: Mathematics. Mechanics. Informatics*, 2017, 3 (24), pp. 11–24.

Для цитирования: Мельников В. А. Применение генетических алгоритмов для отыскания оптимальной последовательности раскрыя // *Вестник Сыктывкарского университета. Сер. 1: Математика. Механика. Информатика. 2018. Вып. 1 (26). С. 64–72.*

For citation: Melnikov V. A. Application of genetic algorithms for finding the optimal nesting sequence, *Bulletin of Syktuykar University, Series 1: Mathematics. Mechanics. Informatics*, 2018, 1 (26), pp. 64–72.

СГУ им. Питирима Сорокина

Поступила 25.12.2017