Построение p-гиперграфов методом имитации отжига

И. Е. Наумов[1], Е. В. Хворостухина[2]

В данной работе рассматривается актуальная проблема построения гиперграфов. Для решения данной задачи использовался интеллектуальный метод оптимизации — метод имитации отжига. Была разобрана предметная область, основные детали алгоритма и спроектирована программа. Разработанный алгоритм может применяться для решения и других схожих задач.

Ключевые слова: гиперграфы, метод имитации отжига, интеллектуальные методы оптимизации

1. Введение

Гиперграф — это естественное обобщение понятия графа, ребра которых могут содержать произвольное число вершин. В работе рассматривается задача построения гиперграфов особого класса — p-гиперграфов. Это широкий и важный класс гиперграфов, поскольку их многообразие охватывает, в частности, проективные и аффинные плоскости [1], содержащие более четырёх точек (в данном случае вершинами гиперграфа служат точки плоскостей, а ребрами — соответствующие прямые). Проблема построения плоскостей и гиперграфов имеет факториальную сложность и остается до сих пор открытой. Так, в 1989 году было доказано с помощью компьютера несуществование проективной плоскости 10 порядка, что заняло около 2000 часов компьютерного времени [2]. Для решения задачи построения p-гиперграфов мы предлагаем использовать метод имитации отжига [3], позволяющий найти решение задачи поиска глобального экстремума даже для NP-полных задач за относительно небольшое число итераций и полиномиальное время.

Свое название алгоритм получил от аналогии с физическим процессом нагревания и остывания жидкостей, при котором кристаллическое

[1] *Наумов Илья Евгеньевич* — магистрант, Саратовский государственный технический университет им. Гагарина Ю.А., e-mail: ilya.naumov.99@mail.ru.

Naumov Ilia Evgenievich — graduate student, Yuri Gagarin State Technical University of Saratov

[2] *Хворостухина Екатерина Владимировна* — доцент, Саратовский государственный технический университет им. Гагарина Ю.А., e-mail: Khvorostukhina85@gmail.com.

Khvorostukhina Ekaterina Vladimirovna — associate professor, Yuri Gagarin State Technical University of Saratov

тело сначала нагревают, затем медленно охлаждают до тех пор, пока не будет сформирована наиболее эффективная кристаллическая решетка.

2. Общие сведения о p-гиперграфе

Гиперграфом $H = (V, E)$ будем называть [4] пару множеств $V = \{v_i | i \in I\}$, $E = \{e_j | j \in J\}$, где $I, J \subseteq \mathbb{N}^*$. Элементы $v \in V$ называются вершинами, а элементы $e \in E$, оответственно, ребрами (или гиперребами) и представляют собой произвольные подмножества множества V. Вершины, лежащие в одном ребре, называются смежными, и несмежными в противном случае.

Пусть p — натуральное число. Гиперграф $H = (V, E)$ будем называть p-гиперграфом [5], если он удовлетворяет следующим аксиомам:

(A1) в каждом ребре гиперграфа найдется по крайней мере $p + 1$ вершина;

(A2) любые p вершин гиперграфа содержатся точно в одном ребре;

(A3) в гиперграфе найдется p+1 несмежных вершин.

Для написания программы, осуществляющий построение p-гиперграфа, нам необходимо определиться с ключевыми моментами алгоритма: инициализация начального состояния, целевая функция, получение нового состояния.

3. Инициализация начального состояния

P-гиперграф $H(V, E)$ характеризуется числом вершин — $|V|$, а также числом p. Для проективной плоскости p всегда равно 2. В разработанной программе длина ребер гиперграфа фиксирована и равна L. Из определения следует, что $L \geq p + 1$. Число ребер гиперграфа $H(V, E)$ — $|E|$.

Число ребер, инцидентных одной вершине, одинаково для каждой из вершин и равно $|V_0|$:

$$|V_0| = \frac{|E| \cdot L}{|V|}.$$

Для инициализации начального состояния гиперграфа строится матрица A размера $|E| \times L$. Здесь строки представляют собой ребра, а столбцы — инцидентные данным ребрам вершины. Столбцы представляют из себя случайные перестановки множества номеров вершин, удовлетворяющих условию: значения в ячейках матрицы попарно различаются в пределах одной строки.

Таким образом, сформированное начальное состояние обеспечивает выполнение аксиом (A1), (A3) p-гиперграфа. После формирования начального состояния необходимо оценить его целевой функцией.

4. Целевая функция

Вследствие того, что данная функция является наиболее используемой частью алгоритма, оказывая существенное влияние на производительность, она должна требовать минимум ресурсов.

Разработанная целевая функция подсчитывает количество нарушений аксиомы (A2) гиперграфа, иными словами, аксиома нарушается, если у прямых более $p-1$ общих точек.

Алгоритм заканчивает работу, если значение целевой функции равно нулю. В этом случае результатом работы программы будет p-гиперграф, на экран будет выведена его матрица инцидентности.

5. Получение нового состояния

Для генерации новых состояний в алгоритме используются случайные перестановки с наложенными на них условиями в целях сохранения выполнимости аксиомы (A1).

В начале выбираются два различных ребра L_i и L_j ($1 \leq i, j \leq |E|$, $i \neq j$).

Затем из вершин, инцидентных данным ребрам, выбираются по одной вершине Q и P так, чтобы $Q \in L_i$, $P \in L_j$, $Q \notin L_j$, $P \notin L_i$. После соблюдения всех установленных условий вершины меняются местами: вершина Q теперь принадлежит ребру L_j, а вершина P — ребру L_i.

После этого, происходит сравнение двух состояний: того, в котором система находится сейчас (обозначим его x_k, $k \in \mathbb{N}_0$), и образованного из него нового состояния $\tilde{x}$. Если новое состояние лучше, то оно принимается за текущее. В противном случае с вероятностью $p = e^{-\Delta/T_k}$, где $\Delta = f(\tilde{x}) - f(x_k)$, оно тем не менее может быть принято за следующее состояние системы. T_k — температура системы на k-ой итерации, T_0 — начальная температуры, является гиперпараметром и задается вручную. После каждой итерации температура системы понижается: $T_{k+1} = \alpha T_k$, где α — также гиперпараметр, определяющий темп охлаждения системы. Для каждой задачи гиперпараметры подбираются индивидуально [3], вместе с тем, рекомендуемые интервалы для начальной температуры T_0 и темпа охлаждения α следующие: $T_0 \in [100; 1000]$, $\alpha \in [0.98; 0.999]$.

6. Результат

Реализация описанного выше алгоритма была осуществлена на универсальном языке программирования Python 3.9. На рисунке 1 представлен график зависимости значения целевой функции от итерации. Результатом данного прогона программы стал 3-гиперграф, состоящий из 8 вершин (vertCount=8), 14 ребер (edgeCount=14) длины 4 (lineLong = 4). При получении этого решения были использованы следующие параметры метода: значение начальной температуры $T_0 = 100$, темп охлаждения $\alpha = 0,9$. Для его построения потребовалось 5 попыток и 129 итераций. Под попыткой понимается достижение программой пункта останова: либо текущая температура достигла минимально заданной (в этом случае формируется новое начальное состояние и процесс поиска решения начинается сначала), либо искомый гиперграф был найден.

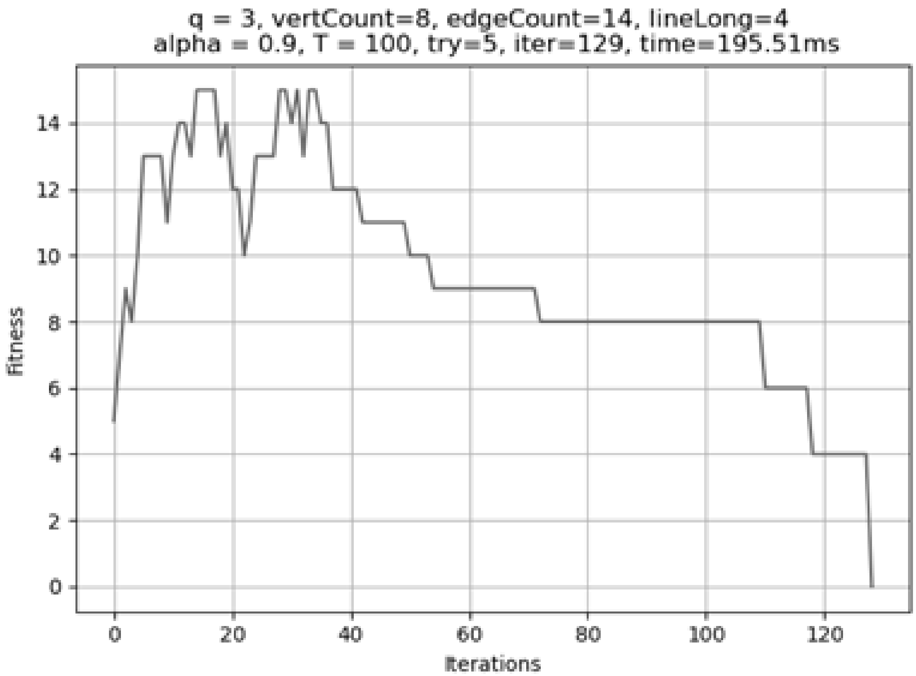

Рис. 1. График зависимости значений целевой функции от итерации.

7. Заключение

На данный момент программа способна строить p-гиперграфы невысоких порядков. Далее планируется усовершенствовать этот алгоритм, а также использовать для решения данной задачи другие алгоритмы интеллектуальной оптимизации, например, генетические алгоритмы. Разработанный нами алгоритм может быть также применен и для решения других NP-полных задач.

Список литературы

[1] Hartshorne R., *Foundations of projective geometry*, W.A. Benjamin Inc., New York, 1967.

[2] Lam C. W. H., "The Search for a Finite Projective Plane of Order 10", *The American Mathematical Monthly*, **98**:4 (1991), 305–318.

[3] Kirkpatrick S., Gelatt C. D., Vecchi M. P., "Optimization by Simulated Annealing", *Science*, **220**:4598 (1983), 671–680.

[4] Bretto A., *Hypergraph theory: an Introduction*, Springer, Berlin, 2013.

[5] Khvorostukhina E. V., Molchanov V. A., "Abstract Characterization of Input Symbol Semigroups of Universal Hypergraphic Automata", *Lobachevskii Journal of Mathematics*, **41**:2 (2020), 214–226.

P-hypergraph design with simulated annealing

Naumov I.E., Khvorostukhina E.V.

In this paper, a present-day problem is considered — the design of hypergraphs. For this purpose, the simulated annealing method was used. The subject area was reviewed along with the main parts of the algorithm and the program was designed. The developed algorithm can be used to solve other similar problems.

Keywords: hypergraphs, simulated Annealing, intelligent optimization method

References

[1] Hartshorne R., *Foundations of projective geometry*, W.A. Benjamin Inc., New York, 1967.

[2] Lam C. W. H., "The Search for a Finite Projective Plane of Order 10", *The American Mathematical Monthly*, **98**:4 (1991), 305–318.

[3] Kirkpatrick S., Gelatt C. D., Vecchi M. P., "Optimization by Simulated Annealing", *Science*, **220**:4598 (1983), 671–680.

[4] Bretto A., *Hypergraph theory: an Introduction*, Springer, Berlin, 2013.

[5] Khvorostukhina E. V., Molchanov V. A., "Abstract Characterization of Input Symbol Semigroups of Universal Hypergraphic Automata", *Lobachevskii Journal of Mathematics*, **41**:2 (2020), 214–226.