ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ РЕШЕНИЯ ЗАДАЧ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ

Нестеренков С. Н., Наливко В. Н.
Кафедра информатики, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: nsn@bsuir.by, vnalivko@mail.ru

Сформулирована задача многомерной оптимизации и предложено ее решение, базирующееся на генетическом алгоритме. Рассмотрены основные достоинства и недостатки данного подхода.

Введение

Одной из задач многомерной оптимизации является задача о рюкзаке [1]. Цель многомерной задачи о рюкзаке состоит в том, чтобы увеличить сумму значений элементов, которые должны быть выбраны из некоторого заданного набора, с учетом ограничений по нескольким ресурсам. Эта проблема широко изучалась на протяжении многих десятилетий. В основном задачу можно сформулировать следующим образом [2]:

$$max\ f\left(x_1, x_2, \ldots, x_n\right) = \sum_{j=1}^{n} p_j x_j. \qquad (1)$$

При условии:

$$\sum_{j=1}^{n} w_{ij} x_j \leq c_i, \qquad (2)$$

$$i = 1, \ldots, m; \quad x_j \epsilon \left\{0, 1\right\}; \quad j = 1, \ldots, n;$$

$$p_j > 0; \quad w_{ij} \geq 0; \quad c_i \geq 0.$$

Для корректной постановки опишем условные обозначения:

1. n – количество объектов;
2. m – количество ранцев;
3. w_{ij} - потребление ресурса i для объекта j;
4. c_i – вместимость i-го ранца;
5. p_j - прибыль;
6. x_j - переменная решения.

Многомерная задача о рюкзаке является частным случаем классической задачи о рюкзаке 0-1 и имеет более одного ограничения. Классическая задача о рюкзаке состоит в том, чтобы выбрать подмножество из бесконечного набора предметов, что повышает линейную функцию выбранных предметов в зависимости от одного ограничения неравенства. Многомерная задача о рюкзаке является особенно сложной проблемой целочисленного программирования, поскольку матрица ограничений, состоящая из w_{ij}, является плотной. С другой стороны, для многомерной задачи о рюкзаке есть реальное решение при $x_j = 0;\ j = 1, \ldots, n$, тогда как найти реальное решение может быть так же сложно, как найти оптимальное решение в общем целочисленном программировании [3].

I. Генетический алгоритм

Генетические алгоритмы, которые находят применение в биоинформатике, вычислительной науке, экономике, химии, производстве, математике, физике и других областях, являются алгоритмами поиска, основанными на естественном отборе и генетике. Эти алгоритмы принадлежат к большему классу эволюционных алгоритмов, которые генерируют решения для задач оптимизации с использованием методов, основанных на естественной эволюции: наследование, мутация, отбор и кроссовер. Можно сказать, что у самых сильных особей в популяции будет больше шансов передать свои гены следующему поколению.

В генетическом алгоритме популяция возможных решений проблемы оптимизации развивается в сторону лучших решений. Каждое возможное решение имеет набор свойств, которые могут быть видоизменены. Традиционно решения представляются в двоичном виде (0, 1), но возможны и другие кодировки.

Эволюция обычно начинается с популяции случайно сгенерированных индивидов и происходит поколениями. В каждом поколении оценивается пригодность каждого индивида в популяции, более подходящие индивиды стохастически выбираются из текущей популяции, и геном каждого индивида модифицируется для формирования новой популяции. Новая популяция затем используется в следующей итерации алгоритма. Обычно алгоритм завершается, когда было произведено максимальное количество поколений или достигнут удовлетворительный уровень пригодности популяции [4]. Многие сложные проблемы оптимизации могут быть решены или преобразованы с помощью последовательности подзадач, например, с помощью методов решения задач о рюкзаке [5].

Воспроизведение включает следующие способы:

- Чистое воспроизведение – особь копируется непосредственно в следующее поколение;
- Кроссовер – выбраны две особи, их гены в какой-то момент пересекаются, поскольку первая часть новой особи происходит от одного родителя, а последняя часть – от другого;

– Мутация – индивидуум выбран, и один бит изменен.

II. Общее описание работы алгоритма

Генетический алгоритм состоит из следующих шагов [6–11]:

1. Каждое из m-ограничений обрабатывается отдельно, и его оптимальное решение находится методом динамического программирования. Найдены общие частоты появления, которые находятся в векторах решения, затем они отсортированы в порядке убывания, и получена индексная последовательность I;
2. Первые n-элементов начальной совокупности устанавливаются таким образом, что элемент, относящийся к текущему индексу, берется до тех пор, пока он не превышает вместимость рюкзака, начиная с i-го элемента последовательности индекса $I\,(1 < m < i)$ на каждом шагу;
3. Каждое из m-ограниченйи обрабатывается отдельно, и вычисляются значения: $p_j/w_{ij}, (1 \leq i \leq m)$. Расслабленные решения каждого ограничения найдены, затем получается индексная последовательность J путем сортировки частот ввода решения каждого элемента в порядке убывания;
4. Другие n-элементов начальной совокупности устанавливаются таким образом, что элемент, относящийся к текущему индексу, берется до тех пор, пока он не превышает вместимость рюкзака, начиная с j-го элемента последовательности $J\,(1 \leq j \leq n)$;
5. Коэффициенты целевой функции p_j сортируются в порядке убывания, и получается индексная последовательность K;
6. Каждый индивид, состоящий из 2n-элементов, пересекается со всеми остальными индивидами. Если есть элемент, который можно взять для сгенерированного индивида, элемент, относящийся к текущему индексу, берется до тех пор, пока он не превышает вместимость рюкзака, начиная с первого элемента последовательности индекса $K\,(1 \leq k \leq n)$. Индивид, имеющий максимальное значение целевой функции в популяции, назначается в качестве записи;
7. Предыдущий шаг повторяется до тех пор, пока номер итерации не станет n;
8. Запись записана, и алгоритм заканчивается.

Заключение

Многомерная проблема задач о рюкзаке возникает в различных приложениях, таких как погрузка груза, упаковка в контейнеры, финансовый менеджмент, для решения задачи составления расписания пар преподавателей, причем в данную задачу можно внести еще дополнительные переменные, такие как часы занятости преподавателей другими видами деятельности, а также часы занятости студентов (в случае посещения ими занятий в других организациях) [12]. Рассмотренный алгоритм дает оптимальные решения для всех случаев. В отличие от техники классического генетического алгоритма, начальная популяция не генерируется случайным образом в этом алгоритме через шаги 1–4. Таким образом, пространство решения сканируется намного эффективнее. Кроме того, вместимость рюкзака влияет на время работы [13–16].

Список литературы

1. Абакаров А. Ш., Сушков Ю. А. Статистическое исследование одного алгоритма глобальной оптимизации. — Труды ФОРА, 2004.
2. Djannaty F., Doostar. S. A Hybrid Genetic Algorithm for the Multidimensional Knapsack Problem. International Journal Contemporary Mathematical Sciences. 2008. P. 443–456.
3. Lin E. A bibliographical survey on some well-known non-standard knapsack problems, 2004. P. 274–317.
4. Goldberg D. E. Genetic Algorithms in Search, Optimization and Machine Learning // Addison–Wesley.
5. Haul C., Voss S. Using surrogate constraints in genetic algorithms for solving multidimensional knapsack problems / Woodruff D. L. Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search // Kluwer Academic Publishers. 2002 P. 235–251.
6. Gotlieb J. On the effectivity of evolutionary algorithms for multidimensional knapsack problem, Proceedings of the 4th European Conference of Artificial Evolution. // Dunkerque, France. P. 23–27.
7. Gavish B., Pirkul H. Efficient Algorithms for Solving Multiconstraint Zero-One Knapsack Problems to Optimality // Mathematical Programming 31. P. 78–105.
8. Freville A. The multidimensional 0–1 knapsack problem. // European Journal of Operational Research 155. 2004 P. 1–21.
9. Garey M. R., Johnson D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. P. 338.
10. Kellerer H., Pferschy U., Pisinger D. Knapsack Problems. 2004 P. 546.
11. Khuri S., Back T., Heitkotter J. The Zero/One Multiple Knapsack Problem and Genetic Algorithms // ACM Symposium on Applied Computing. P. 188–193.
12. Нестеренков С. Н. Метод определения персональных весовых коэффициентов преподавателей при распределении их нагрузки / С.Н. Нестеренков // Вести Института современных знаний - 2015. N1 (62). С. 74-80.
13. Vasquez M., Yannick V. Improved results on the 0–1 multidimensional knapsack problem. // European Journal of Operational Research 165. 2005 P. 70–81.
14. Vasquez M., Hao J. K. A hybrid approach for the 0–1 multidimensional knapsack problem. // Proceedings of the Int. Joint Conference on Artificial Intelligence, Seattle, Washington. 2004 P. 328–333.
15. Raidl Gunther R., Gottlieb J. Empirical analysis of locality, heritability and heuristic bias in evolutionary algorithms: A case study for the multidimensional knapsack problem. // Evolutionary Computation Journal 13. 2005 P. 441–475.
16. Raidl Gunther R. An improved genetic algorithm for the multiconstrained 0–1 knapsack problem / Fogel D. et al., eds. // Proceedings of the 5th IEEE International Conference on Evolutionary Computation. P. 207–211.