

АННОТАЦИЯ/РЕФЕРАТ

Дипломная работа: 19 с., 0 рис., 7 табл., 7 источников. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ, МАШИННОЕ ОБУЧЕНИЕ, NP-ПОЛНЫЕ ЗАДАЧИ, ЗАДАЧА О РЮКЗАКЕ, ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ
Объект исследования - задача о многомерном рюкзаке Цель работы - исследование применения генетического алгоритма к задаче о многомерном рюкзаке. В процессе работы был реализован генетический алгоритм на языке C# и его модификации. В результате исследования ...

ОГЛАВЛЕНИЕ

	Стр.
АННОТАЦИЯ/РЕФЕРАТ	5
ВВЕДЕНИЕ	7
 ГЛАВА 1 Теоретическая часть	 8
1.1 Задача о многомерном рюкзаке	8
1.2 Генетические алгоритмы	9
1.2.1 Этапы работы алгоритма	10
1.2.2 Выбор этапов	11
1.2.3 Дополнения алгоритма	12
 ГЛАВА 2 Вторая глава	 13
2.1 Наборы тестов	13
2.2 Первый набор тестов	14
2.2.1 Мутация в одной позиции	14
2.2.2 Инверсионная мутация	15
2.3 Второй набор тестов	15
ЗАКЛЮЧЕНИЕ	17
СПИСОК ЛИТЕРАТУРЫ	18
ПРИЛОЖЕНИЕ А КОД ПРОГРАММНОГО ПРОДУКТА	19

ВВЕДЕНИЕ

В ходе развития компьютерных наук человечеством встретилось с классом NP-полных задач. Такие задачи решаются алгоритмически за недетерминированное полиномиальное время, что существенно затрудняет поиск решения таких задач в приемлемые сроки. Одной из таких задач является задача о рюкзаке (Knapsack problem) и её модификации. Объектом исследования данной работы является задача о многомерном рюкзаке (Multidimensional knapsack problem). Предмет исследования - решение задачи о многомерном рюкзаке с использованием генетического алгоритма.

Многие прикладные проблемы могут быть формализованы в виде рассматриваемой задачи. Примерами таких проблем являются размещение процессоров и баз данных в системе распределенных вычислений (см. [5]), погрузка груза и контроль бюджета (см. [6]), задачи раскрой-ки (см. [7]) и др. Решение этих проблем обуславливает актуальность решения задачи о рюкзаке.

Цель исследования Для достижения цели были поставлены следующие задачи

- Исследовать
- Спроектировать
- Оценить

ГЛАВА 1

Теоретическая часть

1.1 Задача о многомерном рюкзаке

В данной работе рассматривается задача о многомерном рюкзаке (Multidimensional 0-1 knapsack problem, МКР). Эта задача является модификацией классической задачи о рюкзаке, поставленной в 19 веке Джорджем Мэттьюсоном. (см [1]) Данный же вариант задачи впервые был предложен Клиффордом Петерсеном в 1967 году.(см [2])

Постановка задач такова

Пусть существует N предметов, каждый из которых имеет стоимость c_i и размеры s_{ij} , где $i \in 1, 2, \dots, N, j \in 1, 2, \dots, M$. Пусть также существует рюкзак с ограничениями по вместимости по измерениям r_j . Требуется максимизировать сумму

$$\sum_{i=1}^N c_i x_i$$

где $x_i \in \{0, 1\}$ при условии

$$\sum_{i=1}^N s_{ij} x_i < r_j \forall j \in \{1, 2, \dots, M\} \quad (1.1)$$

И стандартная задача, и её модификация являются NP-полными задачами. Вычислительная сложность задачи такого рода при переборном решении для N предметов -

$$o(2^N)$$

, что, вкупе с NP-сложностью, делает алгоритмическое решение та-

кой задачи неэффективным для больших N . Однако такие задачи могут быть решены эвристическими алгоритмами, то есть алгоритмами, для которых их корректность строго не доказана.

1.2 Генетические алгоритмы

Генетические алгоритмы являются семейством в множестве эвристических алгоритмов. Впервые такой алгоритм был предложен А. Фразером. (см [3]) Алгоритм является итеративным. Генетический алгоритм моделирует естественные процессы эволюции популяции, а именно - мутацию и скрещивание. Решение задачи с помощью такого алгоритма требует нескольких предварительных этапов:

- Выбор кодирования генотипа.

На этом этапе нужно выбрать способ кодирования генотипа, который будет эффективен для данной задачи. Такой генотип должен однозначно моделировать сущность, рассматриваемую в задаче.

- Выбор начального приближения.

Для запуска итерационного процесса требуется создать начальное множество - пул генотипов.

- Выбор мутации.

На каждой итерации алгоритма часть пула генотипов подвергнется мутациям, то есть определенным образом изменяются их составляющие.

- Выбор механизма скрещивания (кроссинговера).

После мутации происходит создание новых генотипов из частей старых с сохранением признаков родителя. Алгоритм скрещи-

вания позволяет получить из двух родительских генотипов два различных дочерних генотипа.

- Выбор функции оценки(фитнесс-функции).

Такая функция позволяет оценивать генотипы с точки зрения их близости к оптимальному решению и отбирать из них лучшие на каждой итерации.

1.2.1 Этапы работы алгоритма

- Создается пул генотипов с использованием заданного алгоритма начального приближения
- Запускается итерационный процесс

Случайным образом выбирается часть пула, которая подвергнется мутации

Выбранная часть пула генотипов мутируется с использованием заданного алгоритма мутации

Мутировавшие генотипы замещают собой исходные в пуле, немутировавшие остаются без изменений

Из пула генотипов выбираются пары для скрещивания

Производится скрещивание с использованием заданного алгоритма

С использованием заданной функции оценки из результатов скрещивания выбираются лучшие

Если выполнено условие останова - например, достигнут предел числа итераций или известный максимум, то итерационный процесс завершается, в противном случае начинается следующая итерация.

- Результат итерационного процесса отдается пользователю

1.2.2 Выбор этапов

Наиболее естественным кодированием отдельного решения задачи о рюкзаке в генотип является бинарная последовательность длины N , состоящая из нулей и единиц. Каждый i -й элемент такой последовательности является индикатором вхождения i -го предмета в текущее решение. Такая модель требует наличия проверки коефективности генотипа - соблюдения условия [1.1](#)

Для генерации начального приближения был использован жадный алгоритм. Сначала создается генотип из единиц, соответствующий конфигурации рюкзака, в который положены все предметы. Затем в случайном порядке единицы заменяются на нули, пока полученная конфигурация не будет удовлетворять условию корректности. После этого полученный генотип мутируется с помощью текущей мутации до заполнения пула решений.

В ходе работы было реализовано несколько алгоритмов мутации и скрещивания с целью сравнения их эффективности. Были реализованы следующие алгоритмы мутации:

- Мутация в одной позиции, при которой заменяется значение в одной случайно выбранной точке генотипа.
- Инверсионная мутация, при которой половина генотипа заменяется на противоположные значения.

Были реализованы следующие алгоритмы скрещивания:

- Скрещивание по 1 точке, при котором выбирается произвольная точка в последовательности генотипа, значения до точки берутся

от первого генотипа, после - от второго.

- Скрещивание по двум точкам, при котором выбираются две различные произвольные точки, значения внутри интервала и в самих точках берутся из первого генотипа, вне интервала - из второго.
- Побитовое скрещивание, при котором значения на нечетных позициях берутся из первого генотипа, на четных - из второго.

В качестве функции оценки используется стоимость всех предметов, содержащихся в рюкзаке, соответствующем конфигурации.

1.2.3 Дополнения алгоритма

В связи со спецификой задачи в алгоритм были внесены дополнения.

Были введены проверки генотипов на корректность после мутации и скрещивания. Если генотип не удовлетворяет условию корректности, то значения начиная с первой позиции начинают зануляться до достижения генотипом корректности.

Были введены дополнительные пулы лучших конфигураций за время работы алгоритма. Такие пулы решают одновременно несколько задач

- Недопущение сильного ухудшения результатов решения вследствие случайных мутаций.
- Возможность сравнения решений после окончания работы алгоритма
- Возможность сохранения результатов при перезапуске алгоритма с другим начальным приближением. Такой перезапуск оправдан при получении генотипа - локального максимума.

ГЛАВА 2

Вторая глава

2.1 Наборы тестов

Для оценки эффективности работы алгоритма было проведено его тестирование на различных наборах тестов. Первый набор тестов взят из книги Петерсена(см. [2]) и содержит 7 задач. Условия этих задач(см таблицу 2.1) позволяют проверить эффективность простейшей версии генетического алгоритма без модификаций и оценить её эффективность.

Таблица 2.1 --- Параметры первого набора тестов

№ задачи	Размерность	Количество предметов
1	6	10
2	10	10
3	15	10
4	20	10
5	28	10
6	39	5
7	50	5

Второй набор тестов взят из статьи Чу (см [4]) и содержит в себе 30 задач с одинаковыми параметрами: размерность рюкзака равна 5, рассматривается 100 различных предметов. Для каждого набора тестов известно лучшее решение, эффективность алгоритма оценивалась по скорости поиска решения в миллисекундах и числу итераций. Для первого набора тестов рассматривались медианные значения по 10 запускам

2.2 Первый набор тестов

2.2.1 Мутация в одной позиции

Рассмотрим результаты работы алгоритма без модификаций на первом наборе тестов. Для одноточечной мутации(см таблицу 2.2) алгоритм находит целевое решение

Таблица 2.2 --- Одноточечная мутация, одноточечное скрещивание

№ теста	Итерации	Время, мс
1	6	8
2	114340	2694
3	712	18
4	4022	92
5	3049	97
6	400974	13481
7	9343432	396644

Таблица 2.3 --- Одноточечная мутация, двуточечное скрещивание

№ теста	Итерации	Время, мс
1	21	9
2	70930	1697
3	901	22
4	2092	67
5	2672	95
6	23261	818
7	4770456	199461

Таблица 2.4 --- Одноточечная мутация, побитовое скрещивание

№ теста	Итерации	Время. мс
1	3	9
2	1826334	33017
3	1455	49
4	7869	219
5	2198	74
6	289379	10013
7	348826	13940

Таблица 2.5 --- Инверсионная мутация, одноточечное скрещивание

№ теста	Итерации	Время. мс
1	3	6
2	10	2
3	58	2
4	9538	560
5	-	-
6	-	-
7	-	-

2.2.2 Инверсионная мутация

Половинная мутация в контексте данной задачи оказалась менее эффективной(см таблицу ??)

2.3 Второй набор тестов

Таблица 2.6 --- Инверсионная мутация, двуточечное скрещивание

№ теста	Итерации	Время. мс
1	2	10
2	16	1
3	356	19
4	7161	422
5	-	-
6	-	-
7	-	-

Таблица 2.7 --- Инверсионная мутация, побитовое скрещивание

№ теста	Итерации	Время. мс
1	10	10
2	23	1
3	1409	52
4	246001	12189
5	-	-
6	-	-
7	-	-

ЗАКЛЮЧЕНИЕ

В ходе работы был реализован алгоритм, решающий зада

СПИСОК ЛИТЕРАТУРЫ

1. Mathews, G. B. On the partition of numbers / G.B. Mathews / / Proceedings of the London Mathematical Society. 28: C. 486–490.
2. C.C.Petersen "Computational experience / with variants of the Balas algorithm applied to the selection of R&D projects" Management Science 13(9) (1967) 736-750.
3. Fraser Alex. Computer Models in Genetics. — New York: McGraw-Hill, 1970. — ISBN 0-07-021904-4.
4. P.C.Chu and J.E.Beasley "A genetic algorithm for the multidimensional knapsack problem-/ Journal of Heuristics, vol. 4, 1998, C. 63-86.
5. Gavish, B. and H. Pirkul. (1982). "Allocation of Databases and Processors in a Distributed Computing System." In J. Akoka (ed.) Management of Distributed Data Processing, North-Holland, c. 215–231.
6. Shih, W. (1979). "A Branch and Bound Method for the Multiconstraint Zero-One Knapsack Problem," Journal of the Operational Research Society 30, 369–378.
7. Gilmore, P.C. and R.E. Gomory. (1966). "The Theory and Computation of Knapsack Functions," Operations Research 14, 1045–1075.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММНОГО ПРОДУКТА

```
namespace Hellenist {  
    public enum Case {  
        Nominative = 1,  
        Vocative = 2,  
        Accusative = 3,  
        Genitive = 4,  
        Dative = 5  
    }  
}
```