Ткаленко О. Ю., **магистрант**
Кафедра прикладной математики и информационных систем
Одесский национальный политехнический университет

К ВОПРОСУ ОЦЕНКИ СЛОЖНОСТИ АЛГОРИТМОВ МЕТОДА ВЕТВЕЙ И ГРАНИЦ

В данной работе экспериментальным путём будет показано влияние различных способов получения оценок на вычислительную сложность алгоритмов метода ветвей и границ.

Ключевые слова: *вычислительная сложность алгоритмов, метод ветвей и границ, задача о ранце.*

Постановка проблемы и цель исследования. Скорость сходимости алгоритмов обычно оценивается числом итераций до получения результата. Однако, для различных алгоритмов одинаковой целенаправленности вычислительная сложность различная. Возможно, что большое количество итераций компенсируется простотой их выполнения. Поэтому итерационные алгоритмы, к которым относятся и алгоритм метода ветвей и границ, удобно сравнивать по самой сложности выполняемых итераций, а не только по их числу.

В целом, сложность вычислений алгоритма является функцией от размера входной и выходной информации. Однако, далеко не всегда специфическая структура решаемых задач характеризуется объёмом информации. Сам итерационный процесс предусматривается многократную обработку одних и тех же данных с небольшими изменениями. В качестве оценки сложности вычисления выбирается временная сложность, а это ни что иное, как количество выполняемых операций.

Следуя работе [1], "вычислительная сложность алгоритма – это функция, которая каждому входу размера n ставит в соответствие максимальное число операций, затрачиваемых для решения индивидуальной задачи этого размера, а именно

$$M_A(n) = \sum_{a_i \in P_n} P(a_i) M\big(A(a_i)\big),$$

где $P(a_i)$ – это вероятность появления задачи a_i, $M\big(A(a_i)\big)$ – число операций, затрачиваемых алгоритмом на решение задачи a_i, P_n – множество рассматриваемых задач размера n, $P_n = \{a_i\}_{i=1,2,\ldots}$."

Скорость сходимости различных алгоритмов метода ветвей и границ для решения задачи целочисленного линейного программирования (ЦЛП) исследовалась неоднократно (см. напр. [3]). В основном, это был подсчёт количества итераций в зависимости от изменений способа разбиения множества вариантов на подмножества. Модификация процесса последовательного построения решения [2], используемая в качестве разбиения множества вариантов, даёт хорошие результаты [3].

Признак оптимальности метода ветвей и границ выражается через оценки подмножеств вариантов. Количество просматриваемых подмножеств непосредственно зависит от точности, то есть близости к значению целевой функции оценок. Основной подход получения оценок опирается на идеи расширения-сужения множества вариантов [4]. Ослабляется система ограничений, задача приводится к проще решаемой, результат решения которой используется в качестве оценки. При решении задач ЦЛП "упрощённой" является соответствующая задача линейного программирования. Другой способ получения оценок осуществляется путём решения m нецелочисленных одномерных задач о ранце [4]. Точность полученных оценок в первом и втором случаях значительно отличаются.

Рассматривается задача ЦЛП в постановке:

$$z = max \sum_{j=1}^{n} c_j\, x_j \qquad (1)$$

$$\sum_{j=1}^{n} a_{ij}\, x_j \ \le\ b_i, i = \overline{1,m};\ 0 \ \le\ x_j \ \le 1, j = \overline{1,n} \tag{2}$$

$$x_j - \text{целое}, j = \overline{1,n}. \tag{3}$$

Задача (1)-(2) – соответствующая задача линейного программирования. Оптимальное значение z*, полученное путём её решения, является оценкой ξ множества вариантов (2)-(3). Второй способ получения оценок – это решение m ($\mathrm{i} = \overline{1,\mathrm{m}}$) задач в постановке:

$$z_i = max \sum_{j=1}^{n} c_j\, x_j$$

$$\sum_{j=1}^{n} a_{ij}\, x_j \ \le\ b_i, i = \overline{1,m};\ 0 \ \le\ x_j \ \le 1, j = \overline{1,n}.$$

Оценкой множества вариантов (2)-(3) будет

$$\xi = \min_i z_i^*.$$

Для подсчёта количества операций, первым способом алгоритм решения задачи делился на следующие модули: проверка на оптимальность опорного плана, определение вектора, вводимого в базис, вектора, выводимого из базиса, формирование разрешающей строки и столбца, пересчёт всех компонент симплекс-таблицы. Количество выполняемых операций вычисляется согласно такому полиному:

$$m(3m + 9n + 2) + \ 3n(2n + 1).$$

Количество операций вторым способом вычисляется по следующему полиному:

$$m(3n + n\log n + \ 2).$$

Модули для подсчёта операций такие: определение отношений $\mathrm{x}_\mathrm{j}^\mathrm{i} = \mathrm{c}_\mathrm{j}/\mathrm{a}_\mathrm{ij}$, $\mathrm{j} = \overline{1,\mathrm{n}}$, упорядочение $\mathrm{x}_\mathrm{j}^\mathrm{i}$ в порядке не возрастания, определение вектора решений $\mathrm{X}^\mathrm{i} = (\mathrm{x}_1^\mathrm{i}, \mathrm{x}_2^\mathrm{i}, \dots, \mathrm{x}_\mathrm{n}^\mathrm{i})$, вычисление значений целевой функции $\mathrm{z}_\mathrm{i}^\mathrm{x}$.

Результаты исследования. Для подсчёта количества итераций и операций определялись оптимальные решения многомерных задач о ранце (задачи ЦЛП со всем положительными составляющими) размерностей 5×10, 10×20, 15×25, 20×50 и 30×40. Исследуемая партия содержала по 10 примеров каждой размерности. Результаты экспериментального исследования приведены в таблице:

Таблица 1. – Количественные данные исследования

Способы оценивания / Размерность	Количество итераций		Количество операций	
	I	II	I	II
5×10	7,6	11,1	21477,8	6880,5
10×20	17,1	39,6	278150,5	144554,9
15×25	68,4	273,5	2248477,0	1697231,0
20×50	122,3	1279,4	34007915,4	1735377,7
30×40	154,1	1705,3	3683388,32	15799764,1

Выводы. Количество итераций сильно зависит от размера задачи. Второй способ получения оценок значительно увеличивает количество пересматриваемых подмножеств. Довольно слабо работает правило оценивания вариантов. Количество операций при втором способе оценивания подмножеств значительно меньше, чем при первом способе.

Как заключительное утверждение, действительно можно сказать, что "большое количество итераций компенсируется простотой их выполнения". Сложность вычислений при работе метода ветвей и границ меньше, когда используется упрощённый способ получения оценок подмножеств вариантов.

Руководитель магистерского исследования, доцент кафедры ПМИС Юхименко Б. И.

Литература

1. Анализ сложности алгоритмов [Электронный ресурс] – Режим доступа: https://pro-prof.com/archives/1660

2. Михалевич В. С. Методы последовательной оптимизации в дискретных сетевых задачах/ Михалевич В. С, Кукса А. И.// – Мо: Наука, 1983. – 208 с.

3. Юхименко Б. И. Сравнительная характеристика алгоритмов метода ветвей и границ для решения задач целочисленного линейного программирования/ Юхименко Б. И., Козина Ю. Ю.// Тр. Одес. политехн. ун-та. – 2005. – Впи.2. – с. 199-204.

4. Юхименко Б. И. Методы оптимизации. Учебное пособие. – Одесса.: Издательство ООО "НВП Интерсервис". – 2012. – 270 с.