МЕТОД БАЛАНСА

В.А. Камышников
Томский государственный архитектурно-строительный университет
Россия, 634003, Томск-3, пл. Соляная, 1.
E-mail: kva_son@yahoo.com

Актуальность и постановка задачи

Некоторые практические задачи оптимизации требуют получения оптимального решения, в котором искомые переменные имели бы целочисленные значения, а иногда и значения из какого-либо дискретного множества, множества не обязательно целых чисел. Такие задачи называют задачами целочисленного или дискретного программирования.

Если переменные принимают только два значения: 0 и 1, как это следует в задаче оранце, то такие задачи относят к задачам булевого программирования.

В настоящее время считается работоспособным алгоритм, если трудоемкость решения задач с его помощью растет полиномиально с ростом размера входных данных – переменных и ограничений. К сожалению, задачи линейного программирования, в т.ч. и дискретные задачи линейного программирования относятся к NP-полным задачам, которые нельзя решить никаким полиномиальным алгоритмом.

Поэтому конструирование точного метода для решения таких задач, особенно большой размерности до сих пор сталкивается с непреодолимым препятствием – длительностью решения, и до сих пор оценка эффективности для таких задач в руках вычислительной практики.

Все это свидетельствует об актуальности работ по созданию эффективных методов решения задач линейного целочисленного, в т. ч. булевого программирования.

Основные результаты и научная новизна

Метод баланса осуществляет неявный перебор решений задачи и напоминает в чем-то известный аддитивный алгоритм Балаша. Как показал массовый вычислительный эксперимент он, этот метод практически всегда дает решение, а часто и условно точное (оптимальное) решение задачи линейного целочисленного программирования с булевыми переменными за приемлемое время. Размерность, решаемой задачи определяется только емкостью жесткого диска, Удалось решать задачи (Fortran-программа) на компьютере с емкостью жесткого диска 6 GB до размерности 10000 переменных за время измеряемое минутами. И это не предел.

Надо заметить, что оптимальная точка неизвестна, поэтому для оценки точности решения использовалась точка, наилучшая для генерируемой модели. Генератор задач вначале процесса выбирает случайным образом точку, которая будет допустимой, а м. б. и оптимальной для будущей модели. В качестве цифровых значений коэффициентов целевой функции и ограничений генератор выбирает также случайные числа. Затем производится расчет правых частей ограничений. Такой порядок формирования математической модели задачи при поиске максимума целевой функции обеспечивал то, что значение целевой функции в выбранной случайно допустимой точке могло служить

нижней оценкой максимального значения целевой функции задачи на области определения.
Экспериментальная проверка алгоритма позволила получить приближенную оценку числа итераций – чаще всего число итераций не превышало 10n, где n – число переменных задачи, и зависимость числа итераций от числа переменных близка к линейной.
Подсчет арифметических и логических операций, которые необходимо выполнить при реализации одной итерации алгоритма показал, что ее трудоемкость примерно равна трудоемкости одной итерации симплекс-метода при решении такой же по размеру задачи линейного программирования.
Ниже даются таблицы результатов вычислительных экспериментов над более чем 8000 задач с булевыми переменными. Не нашлось решение 1507 задач, что составило 18,83% от общего числа решаемых задач. Среднее отклонение от предполагаемого максимума целевой функции в решенных задачах составило 1,79 единиц.
Решение находилось всегда, если коэффициенты модели были положительны или нуль, а не нашлось решение только для некоторых задач, в математической модели которых встречались отрицательные коэффициенты.
Несомненно, это свидетельствует о научной новизне полученных результатов.

METHOD OF THE BALANCE / V.A. Kamyshnikov (Tomskiy state architectural-building university, pl. Solianai, 1, Tomsk-3, 634003, Russia).
Urgency and statement of the problem
Some are a practical problems to optimization require the receptions of the optimum decision, in which sought variable had целочисленные importances, but sometimes and importances from some discrete ensemble, ensemble not without fall integer чисел. Such problems name the problem целочисленного or discrete programming.
If variable take only two importances: 0 and 1, as this follows in problem about knapsack, that such problems refer to problem of the boolean programming.
At present runnable algorithm is considered if labour content of the decision of the problems with his help grows polynomial with growing of the size input given - variable and restrictions. Regrettably, problems of the linear programming, including and discrete problems of the linear programming pertain to NP-full problem, which it is impossible solve no polynomial algorithm.
So конструирование exact method for decision of such problems, particularly big dimensionality hitherto faces with resistless obstacle - duration of the decision, and hitherto estimation to efficiency for such problems in hand computing practical persons.
All this is indicative of urgency of the work on making the efficient methods of the decision of the problems linear целочисленного, in t. ch. boolean programming.
Main results and scientific novelty
The Method of the balance realizes tacit перебор decisions of the problem and reminds in than-that known аддитивный algorithm Balasha. What has shown the mass computing experiment he, this method practically always gives the decision, but often and conditionally exact (optimum) decision

problems linear целочисленного programming with boolean variable for acceptable time.
Dimensionality solved tasks is defined only by capacity of the hard disk, Manage to solve the tasks (Fortran-program) on computer with capacity of the hard disk 6 GB before dimensionality 10000 variable for time measured minute. And this not limit.
It is Necessary to notice that optimum point unknown so for estimation of accuracy of the decision was used point best for generated to models. The Generator of the tasks in the beginning process chooses the casual image a point, which will be possible, but m. b. and optimum for future model. As digital importances factor to target function and restrictions generator chooses also casual numbers. Then calculation of the right parts of the restrictions is produced. Such order of the shaping to mathematical model of the task at searching for of the maximum to target function provided that importance of the target function in chosen accidentally possible point could serve the lower estimation of maximum importance of the target function of the task on definitional domain.
Experimental checking the algorithm has allowed to get the drawn near estimation of the number iteration - most often number iteration did not exceed 10n, where n - a number variable tasks, and dependency of the number iteration from number variable close to linear. The Count arithmetical and logical operation, which necessary to execute at realization of one iterations of the algorithm shown that her(its) labour content approximately is labour content to one iterations simplex-method at decision such on size of the task of the linear programming.
Below tables result computing experiment are given on more then 8000 tasks with Boolean variable. Was Not sewn on decision 1507 tasks that has formed 18,83% from the total number of the solved tasks. The Average detour from supposed maximum to target function in solved task has formed 1,79 units.
The Decision was found always if factors to models were positive or zero, rather then was sewn on decision for some tasks only, in mathematical model which met the negative factors. Certainly, this is indicative of scientific novelty got result.

Метод баланса. Некоторые практические задачи оптимизации требуют получения оптимального решения, в котором искомые переменные имели бы целочисленные значения, а иногда и значения из какого-либо дискретного множества, множества не обязательно целых чисел. Такие задачи называют задачами целочисленного или дискретного программирования.

Не редки случаи, когда требуется найти решение на бинарном множестве изменения переменных, т.е. когда переменные могут принимать только два значения: 0 и 1. Подобные задачи принято относить к задачам булевого программирования.

Одной из наиболее широко используемой задач булевого программирования является задача о ранце [1,2]. С ее помощью находят способы наиболее выгодного использования ресурсов.

Турист готовится к длительному переходу. Он может нести груз весом b, который может включать n типов предметов. Каждый отдельный предмет типа j весит a_j, ($j = 1,...,n$), а полезность его использования в переходе оценивается числом c_j. Необходимо определить, сколько предметов каждого типа турист должен положить в рюкзак, чтобы суммарная ценность снаряжения была максимальной. Формальная запись задачи имеет вид.

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

при ограничениях:

$$\sum_{j=1}^{n} a_{ij} x_j \le bi,\ i = 1,...,m, \qquad (2)$$

$$c_j = 1,\ a_{ji} \ge 0,\ bi > 0,\ i = 1,...,m,\ j = 1,...,n, \qquad (3)$$

$$x_j \in [0,1],\ j = 1,...,n. \qquad (4)$$

Среди методов решения подобной задачи нередко называют метод ветвей и границ и алгоритмы отсечений Гомори. Эти методы достаточно хорошо и широко описаны и часто дают решения некоторых задач.

Счетность множества решений задач дискретного программирования всегда представляло из себя соблазн их простого перебора. К сожалению полный перебор мог быть осуществлен лишь для задач малой размерности, но эта лежащая на поверхности идея способствовала появлению методов неявного перебора:

– аддитивный алгоритм Балаша;
– метод лексикографического перебора;
– метод муравьиных колоний;
– генетические алгоритмы;
– метод отжига металла;
– нейронные сети;

– метод неявного перебора по векторной решетке;

– метод неявного перебора, основанный на стратегии локального поиска и другие.

Они применяются для решения задач целочисленного программирования. Множество решений большинства задач целочисленного программирования является не только счетным, но и конечным.

Так, в комбинаторной задаче с n переменными число возможных решений равно 2^n. Следовательно, можно попытаться перебрать все эти решения, проверяя принадлежность каждого из них области допустимых решений; при утвердительном ответе – вычислить значение целевой функции и из этих значений выбрать наилучшее.

Вычислительная сложность процедуры явного перебора очевидна и хорошо исследована. Разработаны более эффективные методы перебора, позволяющие уменьшить объем вычислений либо за счет более эффективной организации работы с каждой отдельной точкой множества решений, либо за счет сокращения числа рассматриваемых точек. В этих методах активно используется информация о решаемой задаче, а также информацию, получаемая непосредственно в процессе перебора. Их называют методами неявного перебора, они отличаются друг от друга способом организации процедуры перебора, что и обычно и определяет их названия.

Эффективность того или иного алгоритма решения какой-либо задачи всегда интересует всех, кто пытается такую задачу решить. Лучшим ответом на этот вопрос была бы формула, связывающая число арифметических операций, которые необходимо проделать для получения решения и какие-то параметры задачи. Кроме того очень важным параметром эффективности алгоритмов является их конечность.

В настоящее время считается работоспособным алгоритм, если трудоемкость решения задач с его помощью растет полиномиально с ростом размера входных данных – переменных и ограничений. Алгоритм называется полиномиальным, если его временная сложность при любом n не превосходит $|p(n)|$, где p – некоторый полином, n – входная длина данных. Алгоритмы, не допускающие такой оценки, называются экспоненциальными.

Полиномиальная проверяемость означает, что существует полиномиальный алгоритм, позволяющий вычислить любое решение задачи и проверить его допустимость.

Существует кроме класса *NP* задач еще и задачи *NP*-полные. В [2] показано, что задачи линейного программирования, в т.ч. и дискретные задачи линейного программирования относятся к *NP*-полным задачам. Установлено, что *NP*-полные задачи нельзя решить никаким полиномиальным алгоритмом. Поэтому конструирование точного метода для решения таких задач может столкнуться с непреодолимым препятствием – длительностью решения.

Это обстоятельство позволяет сделать вывод, что теоретические оценки эффективности алгоритмов в настоящее время не дают возможности объективно оценивать эту эффективность, пока оценка эффективности в руках вычислительной практики, т. е. в применение более или менее удачного метода.

Излагаемый ниже метод решения задач целочисленного программирования с булевыми переменными осуществляет неявный перебор решений задачи и напоминает в чем-то известный аддитивный алгоритм Балаша и практически всегда дает решение, а часто точное решение задачи линейного целочисленного программирования с булевыми переменными.

Конечно, эта постановка задачи представляют собой довольно узкий класс задач дискретного программирования с булевыми переменными, но надежное получение решения таких задач создает предпосылки для решения более широкого класса задач булевого программирования, т.е. задач, у которых коэффициенты c_j могут отличаться от +1 как по значению, так и по знаку.

Кроме того, в процессе тестирования алгоритма удавалось иногда решить кроме подобных задач, задачи, в которых коэффициенты ai_j и bi имели знак как плюс, так и минус.

Экспериментальная проверка алгоритма позволила получить приближенную оценку числа итераций. Так при решении задач чаще всего число итераций не превышает $10n$, где n – число переменных задачи, и зависимость числа итераций от числа переменных явно ли-

нейная. Конечно, все это справедливо для задач, которые удалось с помощью алгоритма решить. Зависимость от числа ограничений тоже существует, она проявляется тоже как линейная, но с коэффициентом меньшим 10.

Подсчет арифметических и логических операций, которые необходимо выполнить при реализации одной итерации алгоритма показал, что трудоемкость одной итерации алгоритма примерно равна трудоемкости одной итерации симплекс-метода при решении задач линейного программирования.

Приведем здесь некоторые результаты решения тестовых задач, которые должны продемонстрировать неприхотливость алгоритма, его всеядность, независимость от вида ограничений и значений коэффициентов.

Цифровые значения для задач выбирались на основе генератора случайных чисел, чтобы процесс проверки имел более объективный характер.

Исследование алгоритма на основе решения множества тестовых задач проводилось в т.ч. и с целью выявления зависимости трудоемкости получения решения от объемных параметров задачи, т.е. зависимости времени решения от числа переменных и числа ограничений. Кроме того, результаты решения тестовых задач должны продемонстрировать неприхотливость алгоритма, его всеядность, независимость от вида ограничений и значений коэффициентов.

В качестве цифровых значений коэффициентов задач выбираются псевдо случайные числа с помощью генератора случайных чисел. Также с помощью генератора случайных чисел выбираются координаты допустимой точки задачи, на основе которых производится расчет правых частей ограничений, и поэтому значение целевой функции в такой точке может служить нижней оценкой максимального ее значения на области определения.

Ниже даются таблицы результатов вычислительных экспериментов над более чем 8000 задач с булевыми переменными. Не нашлось решение 1507 задач, что составило 18,83% от общего числа решаемых задач. Среднее отклонение от предполагаемого максимума целевой функции в решенных задачах составило 1,79 единиц.

Решение находилось всегда, если коэффициенты модели были положительны или нуль, а не нашлось решение только для некоторых задач, в математической модели которых встречались отрицательные коэффициенты.

Обозначения, использованные в таблице: *Iter*, *T* – среднее число итераций и среднее время (мин.), потребовавшихся алгоритму баланса для решения *L* задач размерностью *NM*; *t* – среднее время выполнения одной итерации (сек.); 2^N – число возможных решений (число полного перебора решений); % – значение *Iter* в процентах к 2^N; 10*NM* – оценка числа итераций для симплекс метода при решении задачи линейного программирования.

Сокращенная таблица результатов решения задач линейного программирования с булевыми переменными методом баланса

Число переменных	Число ограничений	Число решаемых задач:			Отклонение от контрольного значения	Среднее время решения задачи, мин.
		Решалось	Не Решено	В %		
10	100	19	0	0,00	2,58	1,71
10	1000	2	0	0,00	1,00	1,82
50	5	144	2	1,39	0,69	13,20
50	10	87	0	0,00	2,11	3,19
50	50	5	0	0,00	1,80	12,52
100	5	20	0	0,00	1,95	10,37
100	10	30	0	0,00	2,83	22,30
500	10	1	0	0,00	3,00	9,97
1000	10	1	0	0,00	3,00	5,13

В следующей таблице приведены результаты решения задач линейного программирования с булевыми переменными (2412 задач).

Затратные характеристики алгоритма баланса при решении задач линейного программирования с булевыми переменными

N	*M*	*L*	*T*	*Iter*	*t*	2^N	%	10*NM*
5	100	21	0,16	11	0,86	32	34,38	5000
7	2000	1	1,29	23	3,36	128	17,97	140000
10	100	19	1,71	42	2,45	1024	4,10	10000
10	1000	2	1,82	40	2,73	1024	3,91	100000
100	5	20	10,37	237	2,63	$>>10^{16}$	0,00	5000
100	10	30	22,30	233	5,74	$>>10^{16}$	0,00	10000
500	10	1	9,97	202	2,96	$>>10^{16}$	0,00	50000
1000	10	11	5,13	59	5,22	$>>10^{16}$	0,00	100000
10000	5	10	22,1	411	17,3	$>>10^{16}$	0,00	500000

Полученные положительные результаты решения задачи о ранце большой размерности методом баланса на однопроцессорном компьютере и удивительные по масштабности результаты решения разнообразных вычислительных задач, в том числе и задач линейного программирования [3 – 10] позволили пересмотреть вычислительный алгоритм метода баланса и подготовиться к вычислительному эксперименту. Надеюсь, что результаты тоже будут положительные.

Использованная литература

1. Акофф Р.Л. Планирование в больших экономических системах.- М.: Советское радио, 1972.

2. Вагнер Г. Основы исследования операций. Том 1,2,3.М., Мир,1972.

3. Ю. Г. Евтушенко, В. У. Малкова, А. А. Станевичюс. Параллельный поиск глобального экстремума функций многих переменных. Журнал вычислительной математики и математической физики, Т. 49, № 2, С. 255-269, 2009.

4. В. А. Гаранжа, А. И. Голиков, Ю. Г. Евтушенко, М. Х. Нгуен. Параллельная реализация метода Ньютона для решения больших задач линейного программирования. Журнал вычислительной математики и математической физики, Т. 49, № 8, С. 1369-1384,2009.

5. Ю. Г. Евтушенко, М. А. Посыпкин. Параллельные методы решения задач глобальной оптимизации. Пленарные и избранные доклады Четвертой международной конференции «Параллельные вычисления и задачи управления» М., 27-29 октября 2008 г. Институт проблем управления им. В. А. Трапезникова. PACO'2008. С. 18-39

6. А. И. Голиков, Ю. Г. Евтушенко. Нахождение проекции заданной точки на множество решений задач линейного программирования. Труды института математики и механики УрО РАН. Т. 14. № 2. С. 33-47, 2008.

7. Евтушенко Ю.Г., В. У. Малкова, А. А. Станевичюс. Распараллеливание процесса поиска глобального экстремума. Автоматика и телемеханика, № 5. С. 46-58, 2008.

8. Сальников А.М., Ярошенко Е.А., Гребенник О.С., Спиридонов С.В. Введение в параллельные вычисления. Основы программирования на языке СИ с использованием интерфейса MPI. - М.: ИПУ РАН, 2009.

9. Цымблер Н.Ю., Соколинский Л.Б. Параллельный алгоритм решения задач линейного программирования // Труды III Международной конференции "Параллельные вычисления и задачи управления" (Москва, 2-4 октября 2006 г.). [Электронное издание] -М.: Институт проблем управления РАН, 2006.

10. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: BXB-Петербург. 2002.