при варируемой структуре ограничений. Если возможности объекта управления (ресурсы) не соответствуют потребностям (целям), то ЛПР в определенной мере может целенаправленно вносить структурные изменения как в сам объект, так и в его связи с внешней средой. Возникает задача целенаправленного формирования и выбора как решения модели, так и ее структурных изменений (варьирование перечня переменных, коэффициентов матрицы и правых частей ограничений).

Задача системной оптимизации может сводиться к задаче линейного программирования с ограничениями:

$$\left(A^0+\Delta A\right)x^*\le b^0+\Delta b,\quad \Delta A\in A,\quad \Delta b\in b,$$

где ΔA – вариации коэффициентов матрицы A; Δb – вариации правых частей ограничений; x^* – целевая установка (x^* не удовлетворяет (*) при ΔA=0, Δb=0).

Метод решения многокритериальной задачи определяется типом информации, получаемой от ЛПР априорно или в процессе решения, принципом оптимальности и процедурой поиска наиболее приемлемого для ЛПР решения.

Получено 21.01.2002

УДК 519.85

Н.И.САМОЙЛЕНКО, д-р техн. наук
Харьковская государственная академия городского хозяйства

ОДНОЭТАПНЫЙ ДИФФЕРЕНЦИАЛЬНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Предлагается одноэтапный метод решения задачи линейного программирования, в основу которого положен дифференциальный алгоритм. Данный метод является развитием существующего трехэтапного дифференциального алгоритма.

К задаче линейного программирования сводятся многие инженерные задачи из различных областей человеческой деятельности, в том числе многие задачи муниципальной экономики и менеджмента.

Задача линейного программирования достаточно подробно изучена. В математической литературе хорошо известны различные методы ее решения, например, *симплексный метод, метод искусственного базиса, модифицированный симплексный метод.* Каждый из этих методов разрабатывался обособлено, без привязки к общей задачи математического программирования. Реже в учебной литературе можно встретить решение задачи линейного программирования по дифференциальному алгоритму, как частное решение общей задачи математического программирования. В известных публикациях [1,2] решение по дифференциальному алгоритму распадается на три этапа: поиск опорного решения, поиск допустимого опорного решения и поиск оптимального решения. Выбор шага дифференциального алгоритма на каждом этапе рассчитывался по разным формулам.

Нами впервые публикуется дифференциальный алгоритм решения задачи линейного программирования, который состоит из одного этапа – поиска оптимального опорного решения. При этом начальное решение получают простым приемом из условия задачи.

Каноническую, или *основную* задачу линейного программирования можно записать следующим образом [3]:

$$y(\mathrm{x}) = \mathrm{c}^{\mathrm{T}}\mathrm{x} + c_0 \to \min_{\mathrm{x}\in\Omega\subset\mathrm{R}^n}, \qquad (1)$$

$$\Omega: \quad \mathrm{A}\mathrm{x} + \mathrm{b} = 0; \qquad (2)$$

$$\mathrm{x} \geq 0, \qquad (3)$$

где x – n-мерный вектор действительных переменных; c – n-мерный вектор коэффициентов оптимизируемой функции y; c_0 – свободный член оптимизируемой функции; Ω – область допустимых решений; A – матрица коэффициентов размерности $m \times n$, $m < n$; b – векторы свободных членов размерности $m \times 1$.

Дифференциальный алгоритм решения задачи математического программирования предполагает, что начальная точка приближения принадлежит области допустимых решений Ω. Произвольное разбиение переменных на зависимые $s_i \left(i = \overline{1,m}\right)$ и $t_i \left(j = \overline{1,p}\right)$ независимые не гарантирует получения допустимого опорного решения. Однако несложные преобразования исходной системы ограничений в канонической задаче линейного программирования позволяют обеспечивать неотрицательность решений на каждом шаге алгоритма.

Преобразования заключаются в следующем. Каждое ограничение-равенство, содержащее отрицательный свободной член, инвертируется, т.е. умножается на -1. Такое умножение изменяет только направление линий, соответствующих инвертируемым ограничениям-равенствам, и никоим образом не влияет на искомое решение задачи. Далее вводятся искусственные, или фиктивные неотрицательные зависимые переменные, по одному в каждое ограничение-равенство. Косая линия в обозначении фиктивной переменной используется для ее отличия от реальной зависимой переменной. Все исходные переменные задачи на первом шаге считаются независимыми.

С учетом указанных преобразований задача (1) – (3) преобразуется в задачу

$$y(\mathrm{t}) = \mathrm{c}^{\mathrm{T}}\mathrm{t} + c_0 \to \min_{\mathrm{t}\in\Omega\subset\mathrm{R}^{\mathrm{n+m}}}, \tag{4}$$

$$\Omega: \quad \$ = \mathrm{At} + \mathrm{b}; \tag{5}$$

$$\$, \mathrm{t} \geq 0. \tag{6}$$

Исходные данные, промежуточные и конечные результаты удобно представлять в виде таблицы дифференциального алгоритма.

Таблица одноэтапного дифференциального алгоритм решения задачи линейного программирования решения

	t_1	…	t_j	…	t_n	
	$\left(\frac{\delta y}{\delta t_1}\right)^{(0)} = c_1$	…	$\left(\frac{\delta y}{\delta t_j}\right)^{(0)} = c_j$	…	$\left(\frac{\delta y}{\delta t_n}\right)^{(0)} = c_n$	$y^{(0)} = c_0$
$\$_1 =$	$a_{11}^{(0)}$	…	$a_{1j}^{(0)}$	…	$a_{1n}^{(0)}$	b_1
…	…	…	…	…	…	…
$\$_i =$	$a_{i1}^{(0)}$	…	$a_{ij}^{(0)}$	…	$a_{in}^{(0)}$	b_i
…	…	…	…	…	…	…
$\$_{m} =$	$a_{m1}^{(0)}$	…	$a_{mj}^{(0)}$	…	$a_{mn}^{(0)}$	b_m

Коэффициенты $a_{ij}^{(0)}$ в таблице равны соответствующим элементам матрицы A в системе равенств (5). Таблица представляет искомое решение, если выполняется условие оптимальности

$$\left(\frac{\delta y}{\delta t_r}\right)^* \geq 0 \qquad \text{при} \qquad t_r^* = 0. \tag{7}$$

Приравнивая все независимые переменные нулю, получаем начальные значения искусственных зависимых переменных $s_i = b_i$, среди которых нет отрицательных в силу ранее проведенных преобразований. Таким образом, мы получаем первое опорное решение задачи (4)-(6):

$$\mathbf{x}^{(0)^{\mathrm{T}}} = \left[\underbrace{0 \quad 0 \quad \cdots \quad 0}_{n} \quad b_1 \quad b_2 \quad \cdots \quad b_m\right]. \tag{8}$$

Данной решение является допустимым и опорным для задачи (4)-(6), но недопустимым – для задачи (1)–(3). Решение имеет n составляющих, равных нулю, а опорное решение задачи (1)–(3) должно иметь p таких составляющих. Поэтому дифференциальный алгоритм решения задачи (4)-(6) будет прежде всего направлен на исключение искусственных переменных $\$_i \; \left(i=\overline{1,m}\right)$, точнее, на перевод их в систему независимых переменных с последующим автоматическим исключением из решения задачи. Такая процедура осуществляется, как и в общей задаче математического программирования, с помощью все тех же жордановых исключений.

Если среди коэффициентов c_j функции y имеются отрицательные составляющие, то это свидетельствует о нарушении необходимых условий минимума [1,2]. Для проведения на каждом шаге оптимизации процедуры транспозиции переменных выбирают независимую переменную, по которой нарушено необходимое условие оптимальности (7). Отрицательная производная $\left(c_j \le 0\right)$ во второй строке таблицы укажет на независимую переменную t_r, которая должна участвовать в очередной транспозиции переменных. Поскольку все независимые переменные находятся на своих нижних границах $\left(t_j = 0\right)$, то приращение варьируемой переменной на каждом шаге должно быть только положительным

$$\Delta t_r > 0 . \tag{9}$$

Условие (9) определяет выражение

$$\Delta t_r^{(k)} = \min_{\frac{\delta s_i}{\delta t_r}<0} \left\{ \frac{s_i^+ - s_i^{(k)}}{\left(\frac{\delta s_i}{\delta t_r}\right)^{(k)}} \right\} = \min_{a_{ir}<0} \left\{ \frac{-b_i^{(k)}}{a_{ir}^{(k)}} \right\} = -\frac{b_k^{(k)}}{a_{kr}^{(k)}} \tag{10}$$

в качестве критерия выбора величины приращения (длины шага) для переменной t_r. Нижний индекс свободного члена $b_k^{(k)}$ указывает на зависимую переменную s_k $\left(k \in \left\{\overline{1,m}\right\}\right)$, которая должна вместе с переменной t_r участвовать в очередной транспозиции переменных.

Транспозиция переменных осуществляется с помощью обычных жордановых исключений. При этом элементы второй строки также подвергаются вычислительным процедурам жордановых исключений на общих основаниях с остальными элементами таблицы. Последняя ячейка в этой строке показывает значение функции $y^{(k)}$ в текущей точке приближения к минимуму. Когда необходимые условия по всем переменным будут выполнены, в данной ячейке будет находиться искомый минимум функции y^{*}.

Если в процессе решения задачи все условные производные по независимым переменным окажутся одновременно больше нуля, а среди зависимых переменных будет находиться хотя бы одна искусственная переменная, то такая задача не имеет ни одного допустимого решения.

Если в процессе решения задачи окажется, что в столбце с отрицательной условной производной $\left(\frac{\delta y}{\delta t_r}\right)^{(k)}$ не окажется ни одного отрицательно элемента $a_{ir}^{(k)}$, то такая задача также не имеет решения, поскольку функция цели не ограничена снизу: ее можно уменьшать до бесконечности.

Предлагаемый одноэтапный алгоритм полностью поддерживает идеологию решения общей задачи математического программирования и в то же время по своей реализации значительно проще известного трехэтапного дифференциального алгоритма.

1.Евдокимов А.Г. Минимизация функций. – Харьков: Изд-во при Харьк. ун-те, 1977. – 160 с.

2.Евдокимов А.Г., Самойленко Н.И., Пальченко Л.А, Рябченко И.Н. Минимизация функций с применением микро- и мини-ЭВМ. Сборник задач и упражнений. – Харьков: Основа, 1993. – 256 с.

3.Акулич И.А. Математическое программирование в примерах и задачах. – М.: Высш. шк., 1986. – 319 с.

Получено 21.01.2002

УДК 628.17 (628.153) : 519.17 : 681.3

И.Л.ЯКОВИЦКИЙ

Харьковская государственная академия городского хозяйства

АЛГОРИТМ ПРИНЯТИЯ РЕШЕНИЙ В АВАРИЙНОЙ СИТУАЦИИ НА ИНЖЕНЕРНОЙ СЕТИ

Приведены характеристика аварийной ситуации и модель алгоритма принятия решения в условиях аварийной ситуации в инженерной сети, в частности в системе распределения воды.