УДК 519.854

С.В. ЧУПОВ

НОВЫЕ ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ НА ОСНОВЕ ЛЕКСИКОГРАФИЧЕСКОГО ПОИСКА

Аннотация. Предложены новые подходы к решению задач дискретного программирования на основе поиска лексикографического упорядочения векторов, при котором оптимальное решение задачи либо совпадает с лексикографическим экстремумом множества допустимых решений задачи, либо находится достаточно близко от него в лексикографическом смысле. Описаны обобщенная схема такого лексикографического поиска и возможности для ее модификации. Проиллюстрированы значительные преимущества в эффективности работы данного подхода по сравнению со стандартным алгоритмом лексикографического поиска.

Ключевые слова: лексикографический порядок, лексикографический максимум, задача дискретного программирования, алгоритм лексикографического поиска.

ВВЕДЕНИЕ

Практические задачи оптимизации требуют эффективных алгоритмов для отыскания их решений. В современных условиях в связи с быстрым развитием средств вычислительной техники и разнообразных технологий, в частности технологии параллельных вычислений, возникает необходимость в разработке новых алгоритмов и методов, позволяющих за приемлемое время получать оптимальные или близкие к ним решения. Это особенно актуально вследствие значительного роста размерности реальных прикладных задач. В данной работе такие методы и алгоритмы используются в целях повышения эффективности алгоритма лексикографического поиска решения дискретных задач оптимизации [1, 2].

ПОСТАНОВКА ЗАДАЧИ. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Рассмотрим задачу дискретного программирования: максимизировать

$$x_0 = f_0(x) \tag{1}$$

при условиях

$$f(x) \le b, \tag{2}$$

$$x \in D, \tag{3}$$

где $f: R^n \to R^m$, $b \in R^m$, $D = D_1 \times D_2 \times \ldots \times D_n$ — дискретное множество, $D_j \subset [\alpha_j, \beta_j] \subset R$, $j = 1, 2, \ldots, n$, — дискретные множества. Если $f_0(x)$ и векторная функция $f(x)$ линейны, а каждое из множеств D_j, $j = 1, 2, \ldots, n$, — подмножество целых чисел, то задача (1)–(3) является задачей целочисленного линейного программирования. Множество решений, удовлетворяющих условиям (2), (3), обозначим X^D. Полагаем, что каждая переменная задачи ограничена как снизу, так и сверху, т.е. множество X^D ограничено. Определим лексикографическое упорядочение векторов следующим образом: вектор $x \in R^n$ лексикографически положительный, $x >^L 0$ ($>^L$ — знак отношения «лексикографически больше»), если первая по порядку отличная от нуля координата этого вектора положительна; вектор $x \in R^n$ лексикографически больше вектора $y \in R^n$, $x >^L y$, если вектор $(x - y)$ лексикографически положительный, $x - y >^L 0$. Равенство векторов определяется стандартным образом. При таком упорядочении любые два вектора одной размерности сравнимы между собой. На основе лексикографического упорядочения векторов формулируется понятие лексикографического максимума множества. Решение x^L из некоторого множества $G \subset R^n$ называется лексикографическим максимумом этого множества, $x^L = \max^L G$, если любое решение из множества G лексикографически не больше x^L, т.е. для каждого допустимого вектора x из множества G выполняется условие $x \le^L x^L$ ($\le^L$ — знак отношения «лексикографически не больше»). Аналогично вводится понятие лексикографического минимума множества.

Рассмотрим множество

$$\{(x_0, x) \in R^{n+1} | \, x \in X^D, \; f_0(x) - x_0 \ge 0\}. \tag{4}$$

Теорема 1. Лексикографический максимум (x_0^*, x^*) множества (4) определяет вектор x^* — оптимальное решение задачи (1)–(3).

Доказательство. Рассмотрим задачу оптимизации: максимизировать

$$z = x_0 \tag{5}$$

при условиях

$$f_0(x) - x_0 \ge 0, \tag{6}$$

$$x \in X^D. \tag{7}$$

Нетрудно видеть, что задачи (1)–(3) и (5)–(7) эквивалентны. Пусть (x_0^*, x^*) — лексикографический максимум множества (4). Тогда произвольное решение (x_0, x), которое принадлежит множеству (4), лексикографически не больше решения (x_0^*, x^*), поэтому $x_0^* \ge x_0$. Поскольку значение x_0 в задаче (5)–(7) максимизируется, точка (x_0^*, x^*) — оптимальное решение задачи (5)–(7). Вследствие того, что задачи (1)–(3) и (5)–(7) эквивалентны, вектор x^* — оптимальное решение задачи (1)–(3).

ДЕТЕРМИНИРОВАННЫЙ АЛГОРИТМ ЛЕКСИКОГРАФИЧЕСКОГО ПОИСКА

Используя понятие лексикографического максимума множества и теорему 1, процесс нахождения оптимального решения задачи (1)–(3) можно свести к отысканию лексикографических максимумов последовательности множеств $X^0, X^1, \ldots, X^k, \ldots$, где $X^0 = X^D$, $X^k = \{x \in X^D | \, x \le^L x^{k-1}, \; f_0(x) \ge x_0^{k-1} + 1\}$, $k = 1, 2, \ldots$, $x^k = \max^L X^k$, $x_0^k = f_0(x^k)$, $k = 0, 1, \ldots$ На таком определении множеств базируется алгоритм последовательного лексикографического поиска решения задачи (1)–(3) [2, 7]. Если $X^0 \ne \varnothing$, то в результате реализации описанной схемы последовательного лексикографического поиска строится лексикографически убывающая последовательность допустимых решений

$$x^0 >^L x^1 >^L \ldots >^L x^k >^L \ldots, \tag{8}$$

которой соответствует возрастающая последовательность $x_0^0 > x_0^1 > \ldots > x_0^k > \ldots$ значений целевой функции (1). Процесс решения задачи заканчивается, как только на некотором шаге $k+1$ $(k \ge 0)$ получим $X^{k+1} = \varnothing$. Это условие фактически означает, что среди допустимых решений, лексикографически меньших x^k, не существует лучших по значению целевой функции. Поскольку каждая точка x^i, $i \in \{0, 1, \ldots, k\}$, является лексикографическим максимумом соответствующего множества X^i, вектор x^k будет решением задачи (1)–(3). Конечность поиска обеспечивается ограниченностью допустимого множества X^D. Описанный алгоритм лексикографического поиска достаточно прост, если имеется возможность быстро определить лексикографический максимум множества X^D или некоторого его подмножества $\{x \in X^D \mid x \leq^L \bar{x}\}$. Однако при отыскании отдельных допустимых решений последовательности (8) процесс поиска может занимать длительное время. Каждое допустимое решение x^k, $k \geq 1$, определяется в результате построения лексикографически убывающей последовательности точек

$$x^{k-1} = y^0 >^L z^1 >^L y^1 >^L \ldots >^L z^r >^L y^r >^L \ldots, \qquad (9)$$

где $z^r = \max^L \{x \in D \mid x <^L y^{r-1}, f_0(x) > x_0^{k-1}\}$, $y^r = \max^L \{x \in X^D \mid x <^L z^r\}$. Очевидно, $x^k = z^r$, если $z^r \in X^D$, и $x^k = y^r$, если $f_0(y^r) > x_0^{k-1}$. В процессе построения такой последовательности гарантируется, что каждая точка просматривается только один раз и не будет пропущено ни одного подходящего решения. Таким образом, оптимальное решение также не будет пропущено. Описанный алгоритм лексикографического поиска решения задачи (1)–(3) и соответствующий метод назовем детерминированными [6]. Далее под подходящим решением будем понимать допустимое решение, для которого значение целевой функции не меньше (для задачи максимизации) или не больше (для задачи минимизации) уже достигнутого рекордного значения.

АНАЛИЗ ЭФФЕКТИВНОСТИ ДЕТЕРМИНИРОВАННОГО АЛГОРИТМА

Эффективность детерминированного алгоритма существенно зависит от количества точек в последовательности (9). Как правило, их число значительно возрастает с увеличением номера шага k, что снижает эффективность поиска. Это особенно заметно при решении задач, число переменных в которых больше 80.

В качестве иллюстрации продемонстрируем результаты решения тестовых задач о многомерном ранце с булевыми переменными стандартным детерминированным алгоритмом лексикографического поиска при различных значениях количества переменных и разных структурах матрицы ограничений задачи. На рис. 1 изображены три кривые, соответствующие разным структурам матрицы ограничений задачи. Кривая *1* соответствует структуре матрицы ограничений, для которой доля единичных значений в оптимальном решении составляет около 25 % общего числа переменных, кривая *2* — 50 % , кривая *3* — 75 %. Процентное отношение количества ненулевых значений в решении к общему числу переменных назовем плотностью значений решения. Последовательно решались серии по пять задач с количеством ограничений, равным 20, и количеством переменных 20, 25, 30, 35, 40, 45, 50, 55, 60. После решения отдельной серии подсчитывалось среднее время его получения. Отметим, что время решения задачи — это время, затраченное на анализ и получение всех допустимых решений x^k, $k \geq 0$, последовательности (8).

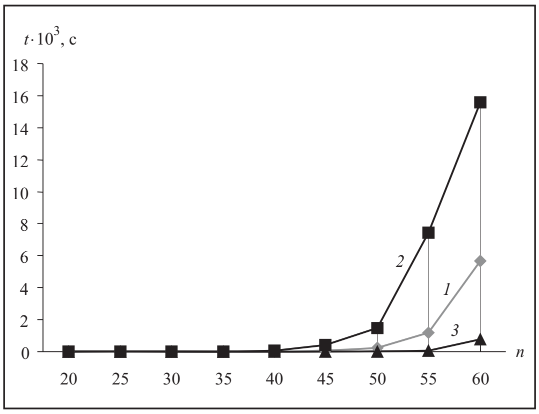

Рис. 1. Графики зависимости размерности от времени решения задачи (*1* — 25 %, *2* — 50 %, *3* — 75 %)

Анализируя представленные результаты, можно сделать вывод, что, во-первых, эффективность работы детерминированного алгоритма лексикографического поиска в значительной степени зависит от структуры матрицы ограничений задачи. Легко видеть, что чем больше допустимых решений содержит множество X^D, тем больше нужно времени на их анализ. При плотности значений 75 % общее число допустимых пересматриваемых решений существенно меньше, чем соответствующее число при плотности 50 %, а при плотности 25 % оно лишь в несколько раз больше, чем при плотности 75 %, и значительно меньше, чем соответствующее количество при плотности 50 %. Во-вторых, при увеличении размерности задачи, в частности числа переменных, с определенного момента «экспоненциально» возрастает время получения решения. Из приведенных результатов видно, что при небольшом числе переменных ($n < 50$) и соответствующей структуре ограничений (плотность в пределах (0 %, 20 %] и [70 %, 100 %)) детерминированный алгоритм лексикографического поиска достаточно эффективен. Современные практические задачи включают сотни и тысячи переменных, поэтому явное использование детерминированного алгоритма для задач большой размерности малоэффективно. Требуется модификация детерминированных схем лексикографического поиска, чтобы за приемлемое время получать подходящие результаты.

ПОИСК В РАЗЛИЧНЫХ ЛЕКСИКОГРАФИЧЕСКИХ ПОРЯДКАХ

Рассмотрим некоторые возможности, позволяющие повысить эффективность процесса поиска оптимального решения, основанного на лексикографическом упорядочении векторов. Отметим, что лексикографическое упорядочение как полный порядок определяет однозначное направление движения (движение в порядке лексикографического убывания решений, например при построении лексикографически убывающей последовательности решений (8), или движение в порядке лексикографического возрастания), в результате которого можно достичь решения задачи (1)–(3). Выбирая различные лексикографические порядки, получаем разные начальные точки, начиная с которых можно осуществить движение к оптимальному решению. Каждая такая точка является лексикографическим максимумом множества X^D в выбранном лексикографическом порядке. Очевидно, что чем лексикографически ближе по отношению к оптимальному решению расположена начальная точка, тем быстрее будет получено решение задачи. Предположим, что выбрано четыре разных лексикографических порядка: $order_1$, $order_2$, $order_3$, $order_4$ и $x^1 = \max^L_{order_1} X^D$, $x^2 = \max^L_{order_2} X^D$, $x^3 = \max^L_{order_3} X^D$, $x^4 = \max^L_{order_4} X^D$ — соответствующие лексикографические максимумы множества X^D в них, x^* — оптимальное решение. Пусть по отношению к x^* решения x^1, x^2, x^3, x^4 схематически размещены так, как показано на рис. 2. Штриховыми линиями показаны направления лексикографического убывания для каждого порядка. Нетрудно заметить, что в лексикографическом порядке $order_4$ решение x^4 находится ближе всего к x^* по сравнению с другими решениями в соответствующих лексикографических порядках, т.е. $(x^4 - x^*)_{order_4} = \min^L \{(x^i - x^*)_{order_i}, i = 1, 2, 3, 4\}$. Таким образом, это решение следует выбрать в качестве начальной точки для алгоритма лексикографического поиска, который будет проводиться в порядке $order_4$.

На рис 3. приведены графики двух процессов получения оптимального решения для одной конкретной задачи о многомерном ранце с булевыми переменными, осуществленных в двух разных лексикографических порядках. При этом оптимальное решение в порядке $order_1$ получено за 0.002 с, для анализа всех подходящих решений потребовалось 11.610 с; в порядке $order_2$ оптимальное решение получено за 93.693 с, анализ всех подходящих решений занял 583.990 с. Разница в эффективности очевидна.

Одной из возможностей, позволящей значительно повысить эффективность алгоритма лексикографического поиска, является удачный выбор лексикографического порядка, в котором будет осуществляться движение к оптимальному решению. Его делают один раз в начале алгоритма или постепенно уточняют лексикографический порядок в процессе поиска решения задачи. Это один из возможных путей построения различных эффективных методов лексикографического поиска. Он требует отдельного исследования. Важным является также вопрос оценки близости имеющегося лексикографического максимума, полученного в данном лексикографическом порядке, по отношению к оптимальному решению в этом же порядке, которое неизвестно. От корректной оценки зависит эффективность метода поиска, основанного на лексикографическом упорядочении.

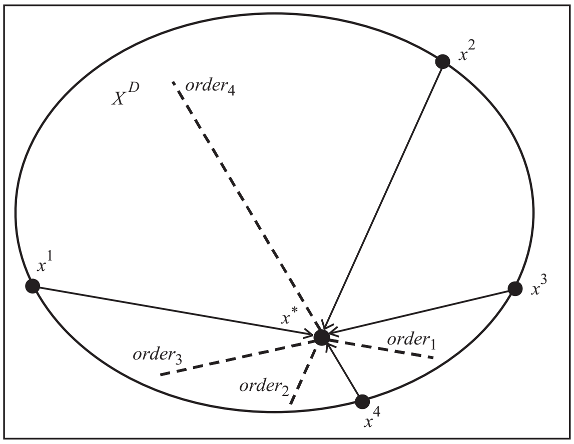

Рис. 2. Схема размещения начальных точек в разных порядках

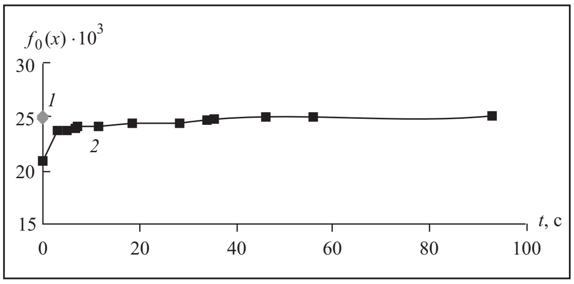

Рис. 3. Графики получения оптимального решения в разных порядках (*1* — $order_1$, *2* — $order_2$)

ВЫБОР НАПРАВЛЕНИЯ ЛЕКСИКОГРАФИЧЕСКОГО ДВИЖЕНИЯ

Дополнительного исследования требуют вопросы, каким образом осуществлять поиск в новом лексикографическом порядке по сравнению с предыдущими и каким должно быть его направление: движение в порядке лексикографического убывания или возрастания. При этом желательно не пересматривать решения, пересмотренные при использовании предыдущих порядков, причем движение следует осуществлять в лексикографическом направлении нахождения оптимального решения. Если за начальную точку движения в новом лексикографическом порядке выбрать последнюю подходящую точку, полученную в результате поиска в предыдущем порядке, то возникает некоторая неопределенность, поскольку не известно, с какой стороны расположена оптимальная точка по отношению к текущей в новом лексикографическом порядке: она может быть лексикографически меньше или больше текущей. Можно параллельно двигаться в обоих лексикографических направлениях, но при этом на последующих шагах количество одновременных поисков в разных направлениях может значительно возрасти и в результате существенно ухудшить общую эффективность поиска. Если по определенным правилам выбрать только одно из направлений движения, то возникает вероятность движения в ложном направлении, что, возможно, сделает поиск бесперспективным. Пример такой ситуации иллюстрируется на рис. 4. Пусть в процессе работы алгоритма лексикографического поиска используется три лексикографических порядка: $order_1$, $order_2$, $order_3$ и $x^0_{order_1}$, $x^0_{order_2}$, $x^0_{order_3}$ — соответствующие лексикогра-

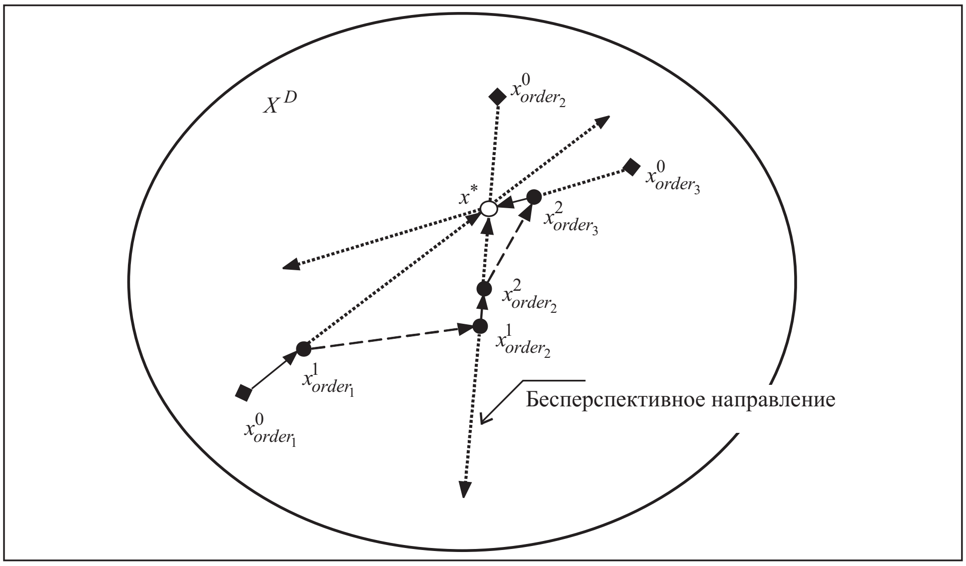

Рис. 4. Схема процесса поиска в разных лексикографических порядках

фические максимумы множества X^D в этих порядках. Поиск начинается с решения $x^0_{order_1}$, движение осуществляется в направлении лексикографического убывания в текущем порядке и определяется лучшее решение $x^1_{order_1}$ $(f_0(x^1_{order_1}) > f_0(x^0_{order_1}))$. Переходим к поиску в лексикографическом порядке $order_2$. Преобразуем решение $x^1_{order_1}$ в $x^1_{order_2}$, при этом $f_0(x^1_{order_1}) =$ $= f_0(x^1_{order_2})$. Однако возникает вопрос, в каком лексикографическом направлении двигаться. Если дальше двигаться, как на предыдущем шаге, то попадем на бесперспективное направление, поскольку оптимальное решение в порядке $order_2$ лексикографически больше, чем $x^1_{order_2}$, а движение осуществляется в направлении лексикографического убывания. Если двигаться в направлении лексикографического возрастания, то в результате получим лучшее по значению целевой функции решение $x^2_{order_2}$. Далее переходим к поиску в порядке $order_3$, получаем решение $x^2_{order_3}$ и, двигаясь в направлении лексикографического убывания, получаем оптимальное решение x^*.

Чтобы избежать такой неопределенности, т.е. при произвольном лексикографическом порядке двигаться всегда в одном направлении, можно в качестве начальной точки движения выбирать не последнюю полученную точку в предыдущем порядке, а последнюю допустимую точку, полученную в данном лексикографическом порядке. Вначале, когда данный лексикографический порядок еще не использовался, такой точкой может быть лексикографический максимум множества X^D в этом порядке. На последующих шагах, когда в процессе работы алгоритма вернемся к текущему порядку, это будет последняя допустимая точка, полученная в результате лексикографического поиска в данном порядке.

ИСПОЛЬЗОВАНИЕ ПРЯМЫХ ЛЕКСИКОГРАФИЧЕСКИХ ОГРАНИЧЕНИЙ

Очевидно, что при таком подходе можно получить решения, уже проанализированные в предыдущих лексикографических порядках. Для того чтобы избежать подобной ситуации, можно с помощью прямых лексикографических ограничений исключать из дальнейшего просмотра решения, просмотренные на предыдущих шагах. В дальнейшем лексикографическое неравенство $x \le^L \hat{x}$ ($x \ge^L \hat{x}$, $x <^L \hat{x}$, $x >^L \hat{x}$) в данном лексикографическом порядке, где $\hat{x}$ — фиксированный вектор, будем называть прямым лексикографическим ограничением. Чтобы уточнить, в каком именно лексикографическом порядке определено данное лексикографическое ограничение, используем запись $x \le^L_{order_k} \hat{x}$ (или сокращенно $x \le^L_k \hat{x}$), где $order_k$ — лексикографический порядок, в котором проводится сравнение векторов. Обозначим $w^k_{order_k}$ последнюю допустимую точку, просмотренную в порядке $order_k$ на шаге $k = 0, 1, \ldots$ Для того чтобы отбросить просмотренные решения, нужно в стандартный детерминированный алгоритм лексикографического поиска внести определенные изменения. А именно, в структуру множества $X^k = \{x \in X^D \mid x \le^L x^{k-1}, f_0(x) \ge x_0^{k-1} + 1\}$, на котором осуществляется поиск лексикографического максимума на текущем шаге, добавить группу прямых лексикографических ограничений, обеспечивающих отсев решений, просмотренных на предыдущих шагах. Предположим, что поиск лексикографического максимума множества X^k проводится в порядке $order_k$, а $order_j$, $j = 0, 1, \ldots, k-1$, — лексикографические порядки, в которых уже выполнялся поиск. Таким образом, множество X^k определим как $X^k = \{x \in X^D \mid x <^L_k x^{k-1}, x <^L_j w^j_{order_j}, j = 0, 1, \ldots, k-1, f_0(x) \ge \hat{x}_0 + 1\}$, где $\hat{x}_0$ — достигнутое рекордное значение целевой функции задачи.

ЛЕКСИКОГРАФИЧЕСКИЙ ПОРЯДОК И ФУНКЦИИ РАССТОЯНИЯ

Во многих современных алгоритмах поиска оптимального решения задач оптимизации используются различные функции расстояния $d(x, y)$ между двумя решениями: x и y. Во-первых, такие функции позволяют разбить начальную область поиска на меньшие области, которые, возможно, не пересекаются, и проводить поиск отдельно в каждой из них. Во-вторых, использование функций расстояния позволяет не допустить проведения поиска в тех областях, которые при определенных условиях считаются бесперспективными или уже были просмотрены. При этом, если точка x — центр области радиуса r, то точка y принадлежит этой области при выполнении условия $d(x, y) \le r$. Лексикографический порядок при некоторых условиях также дает возможность определить функцию расстояния. Например, если векторы x и y булевы, то расстояние между ними в данном лексикографическом порядке, где координаты отсортированы по важности в порядке их естественной нумерации, можно задать функцией $d(x, y) = \left| \sum_{j=1}^{n} 2^{n-j} (x_j - y_j) \right|$. Но в большинстве случаев не обязательно знать точное значение расстояния между двумя решениями, достаточно установить, принадлежит ли некоторое решение определенной области или нет. Для этой цели удобно использовать прямые лексикографические ограничения (односторонние или двусторонние).

СВОЙСТВА ПРЯМЫХ ЛЕКСИКОГРАФИЧЕСКИХ ОГРАНИЧЕНИЙ

Определим некоторые свойства, связанные с прямыми лексикографическими ограничениями.

Свойство 1. Пусть $X^1 = \{x \in D \mid x \le^L_1 x^i, i = 1, \ldots, t\}$ и $X^2 = \{x \in D \mid x \le^L_1 \tilde{x}\}$, где $\tilde{x} = \min^L \{x^1, x^2, \ldots, x^t\}$, знак $\le^L_1$ — лексикографическое отношение в порядке $order_1$. Тогда $X^1 = X^2$.

Доказательство. Справедливость свойства следует из определения лексикографического минимума множества и того факта, что в множестве X^1 все прямые лексикографические ограничения используют одно и то же лексикографическое отношение $\leq_1^L$.

Отметим, что один лексикографический порядок можно получить из другого определенной перегруппировкой переменных задачи, которая установит новые приоритеты для переменных. Этого можно достичь с помощью матриц перестановок. Пусть P — матрица перестановки такая, что позволяет вектор x, определенный в порядке $order_2$, рассматривать в порядке $order_1$. Тогда $P(x)$ обозначает тот же вектор x, но определенный в порядке $order_1$.

Свойство 2. Пусть $X^1 = \{x \in D \mid x \leq_1^L \hat{x}^1\}$, $X^2 = \{x \in D \mid x \leq_2^L \hat{x}^2\}$ и $X^{12} = \{x \in D \mid x \leq_1^L \hat{x}^1,\ x \leq_2^L \hat{x}^2\}$. Тогда $X^{12} = X^1 \cap P(X^2)$, где $P(X^2) = \{y \in D \mid y = P(x),\ x \leq_2^L \hat{x}^2\}$.

Доказательство. Множество $P(X^2)$ можно рассматривать как образ множества X^2 при отображении P. Другими словами, множество X^2 в порядке $order_2$ будет множеством $P(X^2)$ в порядке $order_1$. Поскольку $X^{12} = X^1 \cap X^2$, в порядке $order_1$ имеем $X^{12} = X^1 \cap P(X^2)$.

Проиллюстрируем последнее свойство на примере. Пусть множество D представляет собой множество всех булевых векторов размерности 4 ($D = B^4$). Лексикографический порядок $order_1$ определяет приоритеты переменных как $\{1, 2, 3, 4\}$, порядок $order_2$ — $\{2, 4, 1, 3\}$. Рассмотрим два прямых лексикографических ограничения: $x \leq_1^L x^1$, где $x^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$; $x \leq_2^L x^2$, где $x^2 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$. Тогда

$X^1 = \{x \in D \mid x \leq_1^L x^1\} = \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0' & 0' & 1' & 1' & 0' & 0' & 1' & 1' & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix} \right\}$ в лексикографическом порядке $order_1$ и $X^2 = \{x \in D \mid x \leq_2^L x^2\} = \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0' & 0' & 1' & 1' & 0' & 0' & 1' & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{matrix} \right\}$ в порядке $order_2$, но

$P(X^2) = \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0' & 0' & 0' & 0' & 0' & 0' & 0' & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{matrix} \right\}$ в порядке $order_1$. Таким образом, если считать, что в результате лексикографического поиска в порядках $order_1$ и $order_2$ были просмотрены решения, не удовлетворяющие указанным прямым лексикографическим ограничениям, то дальнейшему просмотру в порядке $order_1$ подлежат только точки из множества $X^1 \cap P(X^2) = \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0' & 0' & 0' & 0' & 0 \\ 0 & 1 & 0 & 1 & 0 \end{matrix} \right\}$.

ОБОБЩЕННЫЙ АЛГОРИТМ ЛЕКСИКОГРАФИЧЕСКОГО ПОИСКА

Учитывая приведенные выше рассуждения, можем построить новый обобщенный алгоритм лексикографического поиска решения задачи (1)–(3). Для этого нужно определить: правило, согласно которому будет осуществляться движение от одного решения к другому в данном лексикографическом порядке; правило построения или выбора нового направления движения (выбора или построения нового лексикографического порядка); условия, определяющие необходимость изменения лексикографического направления движения и прекращения процесса вычислений. На рис. 5 приведена структурная схема, в которой используются следующие обозначения: $y = ALexMove(x, order)$ — алгоритм лексикографического поиска, начинающий свою работу с решения x в порядке $order$; y — допустимое решение, полученное в результате поиска, с которого поиск можно продолжать на следующих шагах; $order = GetOrder(XSet, OrderSet)$ — правило, по которому определяется новый лексикографический порядок $order$; $x = GetSolution(XSet, order)$ — правило, позволяющее получить последнее допустимое решение, просмотренное при поиске в порядке $order$; $OrderSet$ — множество лексикографических порядков, использованных на предыдущих шагах; $XSet$ — множество решений, на которых заканчивались предыдущие шаги; $CanFinished(x)$ — условие завершения поиска; $CanOrderChanged(x, order)$ — условие смены лексикографического порядка, если текущее решение — x, а текущий порядок — $order$; x^* — лучшее решение из найденных на данный момент; $f^* = f_0(x^*)$; k — номер шага алгоритма; $\max_k^L X$ — лексикографический максимум множества X в порядке $order_k$.

$$
\begin{array}{l}
k = 0;\ OrderSet = \{order_k\};\ x^k = \max_k^L X^D;\\
f^k = f_0(x^k);\ x^* = x^k;\ f^* = f^k;\ XSet = \varnothing;\\
\text{while } (!CanFinished(x^*))\{\\
\quad x^k = AlexMove(x^k, order_k);\\
\quad \text{if } \ (f_0(x^k) > f^*)\{\\
\quad x^* = x^k;\ f^* = f^k;\\
\quad \}\\
\quad \text{if } (CanOrderChanged(x^k, order_k))\{\\
\quad XSet = XSet \cup \{x^k\};\\
\quad k = k + 1;\\
\quad order_k = GetOrder(XSet, OrderSet);\\
\qquad \text{if } \ (order_k \in OrderSet)\{\\
\qquad x^k = GetSolution(XSet, order_k);\\
\qquad \}\\
\qquad \text{else } \{\\
\qquad x^k = \max_k^L \{x \in X^D \mid x <_t^L x^t,\ \forall x^t \in XSet\};\\
\qquad OrderSet = OrderSet \cup \{order_k\};\\
\qquad \}\\
\quad f^k = f_0(x^k);\\
\quad \}\\
\}\\
x^{\text{opt}} = x^*;\ f^{\text{opt}} = f^*;
\end{array}
$$

Рис. 5. Структурная схема обобщенного алгоритма лексикографического поиска

Пусть $y = ALexMove(x, order)$ — стандартный детерминированный алгоритм лексикографического поиска. В процессе его работы строится лексикографически убывающая последовательность всех подходящих решений. Итак, для произвольного лексикографического порядка $order$ получим: $x >_{order}^L \ldots >_{order}^L y$. Из того, что допустимое множество дискретно и ограничено, следует, что для каждого лексикографического порядка $order$ необходимо только конечное число шагов использования детерминированного алгоритма $AlexMove$ для просмотра всех подходящих решений на лексикографическом промежутке $\min_{order}^L X^D \le_{order}^L x \le_{order}^L \max_{order}^L X^D$, среди которых найдется и оптимальное решение. Для каждого нового лексикографического порядка $order$ мощность множества еще не просмотренных решений каждый раз уменьшается за счет выбора начальной точки $x_{order}^0 = \max_{order}^L \{x \in X^D \mid x <_t^L x^t \ \forall x^t \in XSet\}$. На определенном шаге мощность множества $\{x \in X^D \mid x <_t^L x^t \ \forall x^t \in XSet\}$ уменьшится настолько, что для просмотра всех решений достаточно осуществить их полный перебор или придем к ситуации, когда это множество окажется пустым. Отсюда следует, что количество используемых в алгоритме разных лексикографических порядков конечно. В то же время можно обеспечить конечность числа разных лексикографических порядков, использующихся в обобщенном алгоритме лексикографического поиска, если предварительно зафиксировать их максимальное количество. Из приведенных выше рассуждений сформулируем следующее утверждение.

Теорема 2. Если $ALexMove(x, order)$ — детерминированный алгоритм лексикографического поиска, решение x^{opt}, полученное в результате применения обобщенного алгоритма лексикографического поиска, является решением задачи (1)–(3).

ВОЗМОЖНОСТИ СТОХАСТИЧЕСКОГО ЛЕКСИКОГРАФИЧЕСКОГО ПОИСКА

Выбор различных способов построения нового лексикографического порядка и начальной точки на каждом шаге, а также правил определения лексикографического направления движения в данном порядке позволяет строить разнообразные методы рассмотренного выше обобщенного поиска. Более того, если в алгоритме *ALexMove* отказаться от построения лексикографически монотонной последовательности решений, то, хотя не будет гарантии просмотра всех подходящих решений, появится возможность создавать более гибкие схемы перебора решений в заданном лексикографическом порядке. С учетом того, что оптимальное решение в каждом лексикографическом порядке *order* содержится между лексикографическим минимумом и максимумом множества X^D, т.е. $\min^L_{order} X^D \le^L_{order} x^{\text{opt}} \le^L_{order} \max^L_{order} X^D$, можно, например, определить следующие общие схемы стохастического лексикографического поиска в данном порядке *order*.

1. Напомним, что на каждом k-м шаге ($k \ge 1$) детерминированного алгоритма лексикографического поиска осуществляется поиск лексикографического максимума множества $X^k = \{x \in X^D \,|\, x \le^L x^{k-1},\ f_0(x) \ge \hat{f}\}$. В результате строится последовательность решений (9), в которой для каждого $r = 0, 1, \ldots$ множество $\{x \in X^k \,|\, y^{r+1} <^L x <^L y^r\}$ в данном порядке *order* пусто. Это, в свою очередь, гарантирует, что не будет пропущено ни одного подходящего решения. Но на некоторых шагах количество членов последовательности (9) может быть очень велико. Пусть для определенного $\tilde{y}^r <^L y^r$ есть возможность вычислять вероятность $P(\tilde{y}^r)$ того, что множество $\{x \in X^k \,|\, \tilde{y}^r <^L x <^L y^r\}$ в данном порядке *order* пусто. Это позволяет строить не всю последовательность (9), а только ее часть, что существенно уменьшит количество ее членов, а следовательно, и общее время получения оптимального решения задачи.

2. Предположим, что с определенной вероятностью можно выяснить, с какой стороны от оптимального в данном лексикографическом порядке расположено решение, получаемое на очередном шаге. Таким образом, на каждом k-м шаге осуществляется процесс случайного перебора допустимых решений на лексикографическом отрезке $x^k_{\min} \le^L_{order} x \le^L_{order} x^k_{\max}$. По завершении очередного просмотра получаем допустимое решение x^k из этого промежутка. Определяем вероятность $P(x^k, order)$ того, что полученное решение x^k лексикографически больше, чем оптимальное. Если $|P(x^k, order) - 0.5| \le \varepsilon$, то отрезок не изменяется; если $P(x^k, order) > 0.5 + \varepsilon$, то на следующих шагах поиска в порядке *order* используем лексикографический промежуток $x^k_{\min} \le^L_{order} x \le^L_{order} x^k$; если $P(x^k, order) < 0.5 - \varepsilon$, то в порядке *order* будем использовать лексикографический отрезок $x^k \le^L_{order} x \le^L_{order} x^k_{\max}$.

3. Пусть известен механизм, позволяющий с заданной вероятностью выяснить, насколько близко в лексикографическом смысле находится оптимальное решение по отношению к определенному лексикографическому промежутку, или, другими словами, вычислить вероятность того, что данный лексикографи-

```
Step=1  RecordF=3735  at  0:00:00.001
Step=1  RecordF=3749  at  0:00:00.036
Step=1  RecordF=3756  at  0:00:00.037
Step=4  RecordF=3759  at  0:00:01.743
Step=9  RecordF=3766  at  0:00:12.385
** Solve: Record=3766 and feasible.**
```

Рис. 6. Результат работы стохастического алгоритма лексикографического поиска

```
Step=0  RecordF=3735  Etaps=0  at  0:00:00.000
Step=1  RecordF=3748  Etaps=1  at  0:00:00.002
Step=2  RecordF=3753  Etaps=89  at  0:00:00.003
Step=3  RecordF=3757  Etaps=1858  at  0:00:00.007
Step=4  RecordF=3761  Etaps=1082885  at  0:00:02.308
Step=5  RecordF=3761  Etaps=20334824735 at 12:12:16.305
```

Рис. 7. Результат работы детерминированного алгоритма лексикографического поиска

ческий промежуток содержит оптимальное решение. Тогда случайным образом определяются лексикографические границы поиска, осуществляется процесс случайного перебора допустимых решений в этих пределах и с помощью указанного механизма вычисляется вероятность нахождения оптимального варианта на этом промежутке. В результате принимается решение о возможности использования данного промежутка на следующих шагах или исключения его из дальнейшего поиска.

Одна из реализаций последней рассмотренной общей схемы заключается в следующем [3]. На каждом шаге согласно указанному механизму фиксируется решение s^k, которое является нижней границей поиска, и предпринимается попытка за определенное число испытаний найти допустимое решение множества $\{x \in X^D \mid s^k \leq^L x <^L x^k, f_0(x) > f_0(x^k)\}$, где x^k — лучшее допустимое решение, полученное на предыдущем шаге. В процессе испытания $t = 1, 2, \ldots, t_{\max}$ случайным образом строится решение $z^t \in \{x \in B^n \mid s^k \leq^L x <^L x^k, f_0(x) > f_0(x^k)\}$. Оно уточняется путем отыскания допустимого решения $y^t = \max^L \{x \in X^D \mid s^k \leq^L x <^L z^t\}$.

Проиллюстрируем эффективность использования такой схемы стохастического лексикографического поиска для алгоритма *ALexMove* по сравнению со стандартным детерминированным алгоритмом лексикографического поиска. Решалась первая тестовая задача из набора задач о многомерном ранце, предложенного Ф. Гловером, в которой количество ограничений — 15, переменных — 100, рекордное значение — 3766. Процесс решения задачи алгоритмом стохастического лексикографического поиска изображен на рис. 6, стандартным детерминированным алгоритмом лексикографического поиска — на рис 7.

Из приведенных результатов видно, что оптимальное решение методом стохастического лексикографического поиска, построенного по обобщенному алгоритму, представленному на рис. 5, получено за 12 с. В то же время стандартный детерминированный алгоритм лексикографического поиска осуществлялся в одном фиксированном лексикографическом порядке и был остановлен после более чем 12 ч работы, так и не достигнув оптимального решения. Это свидетельствует о значительном преимуществе, которое можно получить, если в стандартных детерминированных схемах применить механизмы случайного поиска.

ЗАКЛЮЧЕНИЕ

Лексикографическое упорядочение вариантов позволяет организовывать их целенаправленный просмотр. Осуществляя поиск в различных лексикографических порядках и выбирая разные схемы лексикографического перебора вариантов в заданном порядке с учетом характерных особенностей задачи, можно строить эффективные методы лексикографического поиска для отыскания оптимальных решений современных прикладных задач оптимизации.

СПИСОК ЛИТЕРАТУРЫ

1. Сергиенко И.В., Шило В.П. Задачи дискретной оптимизации. Проблемы, методы решения, исследования. — К.: Наук. думка, 2003. — 261 с.
2. Червак Ю.Ю. Оптимізація. Непокращуваний вибір. — Ужгород: Ужгород. нац. ун-т, 2002. — 312 с.
3. Чупов С.В. Стохастичний алгоритм лексикографічного пошуку розв'язку булевої задачі про ранець // Праці VII міжнар. школи-семінару «Теорія прийняття рішень» (29 вересня – 4 жовтня 2014 р., м. Ужгород). — Ужгород: Ужгород. нац. ун-т, 2014. — С. 263, 264.
4. Чупов С.В. Структурні властивості множини, заданої в різних порядках // Праці IV міжнар. школи-семінару «Теорія прийняття рішень» (29 вересня – 4 жовтня 2008 р., м. Ужгород). — Ужгород: Ужгород. нац. ун-т, 2008. — С. 162.
5. Чупов С.В. Лексикографічний максимум опуклих компактів та алгебраїчні операції над ними // Праці міжнар. конф. «Прогнозування та прийняття рішень в умовах невизначеності» (PDMU – 2001) (11–14 вересня 2001 р., Київ). — К.: Вид.-поліграф. центр «Київський університет», 2001. — С. 138, 139.
6. Чупов С.В. Алгоритм пошуку цілочислового лексикографічного максимуму множини // Наук. вісник Ужгород. держ. ун-ту: Сер. Математика. — Ужгород, 1997. — Вип. 2. — С. 122, 123.
7. Червак Ю.Ю., Гренджа В.І., Чупов С.В. Лексикографічна оптимізація в дискретному програмуванні // Міжнар. конф. «Питання оптимізації обчислень» (Київ, 6–9 жовтня 1997 р.): Праці НАН України. — Київ: Ін-т кібернетики ім. В.М. Глушкова, 1997. — С. 317–319.

Надійшла до редакції 10.11.2015

С.В. Чупов

НОВІ ПІДХОДИ ДО РОЗВ'ЯЗАННЯ ЗАДАЧ ДИСКРЕТНОГО ПРОГРАМУВАННЯ НА ОСНОВІ ЛЕКСИКОГРАФІЧНОГО ПОШУКУ

Анотація. Запропоновано нові підходи до розв'язання задач дискретного програмування на основі пошуку лексикографічного впорядкування векторів, при якому оптимальний розв'язок задачі або збігається з лексикографічним екстремумом множини допустимих розв'язків задачі, або знаходиться достатньо близько від нього в лексикографічному сенсі. Описано узагальнену схему такого лексикографічного пошуку та можливості для її модифікації. Проілюстровано значні переваги в ефективності роботи цього підходу в порівнянні з стандартним алгоритмом лексикографічного пошуку.

Ключові слова: лексикографічний порядок, лексикографічний максимум, задача дискретного програмування, алгоритм лексикографічного пошуку.

S.V. Chupov

NEW APPROACHES TO SOLVING DISCRETE PROGRAMMING PROBLEMS BASED ON LEXICOGRAPHIC SEARCH

Abstract. The author proposes new approaches to solving discrete programming problems based on the search for lexicographical ordering of vectors, such that the optimal problem solution either coincides with the lexicographic extremum of the feasible set of problem solutions or is close enough to it in the lexicographic sense. The general scheme of such lexicographic search and the possibilities for its modification are described. Significant advantages in the efficiency of this approach compared with the the standard lexicographic search algorithm are illustrated.

Keywords: lexicographical ordering, lexicographic maximum, discrete programming problem, lexicographic search algorithm.

Чупов Сергей Викторович,
кандидат физ.-мат. наук, доцент Ужгородского национального университета,
e-mail: sergey.chupov@gmail.com.