Государственный комитет СССР
по материально-техническому снабжению
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ И ПРОЕКТНЫЙ ИНСТИТУТ
СИСТЕМ УПРАВЛЕНИЯ

НАУЧНЫЕ ОСНОВЫ И ПРОБЛЕМЫ СОЗДАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ МАТЕРИАЛЬНО-ТЕХНИЧЕСКИМ СНАБЖЕНИЕМ В ЭКОНОМИЧЕСКИХ РАЙОНАХ

СБОРНИК ТРУДОВ

Выпуск УI

Тула - 1978

М.Ш.Левин, Л.Й.Мошкович

ИНТЕРАКТИВНЫЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧИ О РЮКЗАКЕ НА ОСНОВЕ ПРИБЛИЖЕННОГО АЛГОРИТМА

Проектирование любой эффективной системы управления складскими процессами с использованием электронно-вычислительной техники с необходимостью приводит к постановке ряда оптимизационных задач, сводящихся по сути к поиску объекта в массиве данных. Такими являются задачи оптимального размещения продукции на складах, поиска продукции при комплектовании партий отгрузок, определения потребности в транспорте и ряд других.

Формализация их приводит к сложным математическим проблемам, связанным в первую очередь с необходимостью существенного учета свойства целочисленности решения. Применение общего метода динамического программирования оказывается неэффективным хотя бы ввиду большой размерности возникающей реальной задачи, что оправдывает разработку приближенных численных методов для ряда специальных задач [3].

В статье рассматривается один из таких приближенных алгоритмов, гарантирующий заданную относительную погрешность и имеющий полиноминальную трудоемкость. Предлагается также интерактивный подход, основанный на использовании эксперта в процессе решения задачи.

Рассмотрим следующую задачу целочисленного программирования:

$$\sum_{i=1}^{n} c_i x_i \to max;$$
$$\sum_{i=1}^{n} a_i x_i \le b; \qquad (I)$$
$$x_i = 0 \vee 1; \; c_i, a_i \geqslant 0; \; i = 1, \dots, n.$$

Эта "задача о рюкзаке" ([3]) возникает при решении многочисленных задач управления складскими процессами. Например, к ней сводится задача определения мест хранения продукции, позволяющих при полном опорожнивании ячеек (а это естественное требование возникает в тех случаях, когда кран-штабелер движется по "маятниковым маршрутам" [I]) накапливать на комплектовочной площадке максимально близкое к требуемому количество продукции.

Задача о рюкзаке принадлежит к классу универсальных переборных задач [4] , для которых не известно эффективных точных алгоритмов (с трудоемкостью,* полиномиально зависящей от размерности задачи). В последние годы для некоторых задач этого класса были предложены алгоритмы [9,I0,2], позволяющие получать приближенное решение, за счет снижения требований точности к которому удалось добиться уменьшения трудоемкости вычислений. Следуя [I0] , назовем алгоритм A приближенным алгоритмом задачи (I), если для произвольных исходных данных и $\forall \varepsilon \in (0,1)$ алгоритм A

---

* Под трудоемкостью понимается количество элементарных операций, необходимых для получения решения по данному алгоритму.

позволяет получить решение со значением целевой функции F такое, что

$$\frac{|F - F^*|}{F^*} \le \varepsilon , \qquad (2)$$

где F^* - значение целевой функции при точном решении.

Для задачи (I) приближенный алгоритм с трудоемкостью $O(\frac{n}{\varepsilon^2} + n \log n)$ был описан в [9] и для несколько более общего случая с трудоемкостью $O(\frac{n^3}{\varepsilon^3})$ - в [2]. Заметим, что в [10] предложено некоторое обобщение методологии построения приближенных алгоритмов.

Опишем теперь модификацию приближенного алгоритма применительно к (I). Обозначим через $x^K = (x_1, \dots, x_K)$ наборы из 0 и I длины K, через $c(x^K) = \sum_{i=1}^{K} c_i x_i$ - значение целевой функции на x^K, через $t(x^K) = \sum_{i=1}^{k} t_i x_i$ - значение ресурса на x^K.

Пусть имеется также множество $P^K = \{x^K \mid t(x^K) \le b\}$. Введем операцию усечения множества P^K. Элементы P^K распределяются на непересекающиеся подмножества так, чтобы значения целевой функции на элементах каждого подмножества были близки. Затем из каждого подмножества выбирается наиболее предпочтительный элемент.

Пусть

$$\Delta_j^K = \begin{cases} \left[0, \frac{c^K \varepsilon}{n}\right], & j = 1, \\ \left(\frac{(j-1)c^K \varepsilon}{n}, \frac{j c^K \varepsilon}{n}\right], & j = 2, \dots, N, \end{cases} \qquad (3)$$

где $c^K = \max_{x^K \in P^K} \{c(x^K)\}$, $N = [\frac{n}{\varepsilon}] + 1$.

Рассмотрим следующие шаги построения усечения множества P^n:

$$M_1^j = \{x^K \in P^K \mid c(x^K) \in \Delta_j^K\} \tag{4}$$

$$M_2^j = \{y^K \in M_1^j \mid t(y^K) \leq b\} \tag{5}$$

$$M_3^j = \{Z^K \in M_2^j \mid c(Z^K) \geq c(y^K), \forall y^K \in M_2^j\} \tag{6}$$

$$S^K = \{u^K \in P^K \mid \forall u_1^K, u_2^K \in S^K, \exists j_1, j_2, j_1 \neq j_2, u_1^K \in M_3^{j_1}, u_2^K \in M_3^{j_2}\}. \tag{7}$$

Множество S^K является усечением множества P^n. Переход от S^K к P^{K+1} определяется естественным образом:

$$P^{K+1} = \{x^{K+1} = (x_1, \dots, x_K, \delta) \mid x^K = (x_1, \dots, x_K) \in S^K, \delta = 0 \vee 1\}. \tag{8}$$

С учетом введенных обозначений и преобразований приближенный алгоритм имеет следующий вид:

Шаг 0. $S^0 = \Phi$.

Шаг K. Переход от S^{K-1} к S^K при помощи преобразований (8), (4), (5), (6), (7).

Шаг n. Переход от S^{n-1} к P^n и выбор такого $x_0^n \in P^n$, что $t(x_0^n) \leq b$ и $c(x_0^n) \geq c(y^n)$, $\forall y_n \in \{Z^n \in P^n \mid t(Z^n) \leq b\}$.

На K-м $(K=1, \dots, n)$ шаге алгоритма в решение может быть внесена погрешность (значения целевой функции), ограниченная по модулю величиной $\frac{c^K \varepsilon}{n}$. Таким образом, относительная погрешность значения целевой функции на решении x_0^n оценивается как

$$\frac{\sum_{K=1}^{n} \frac{c^K \varepsilon}{n}}{F^*} \leq \frac{\frac{F^* \varepsilon}{n} \cdot n}{F^*} = \varepsilon, \tag{9}$$

что соответствует (2). Очевидно, трудоемкость алгоритма зависит от числа промежутков Δ_j^K, а относительная погрешность – от длин промежутков. Использование в качестве требуемой для алгоритма длины промежутка, равной $\frac{F^*\varepsilon}{n}$ привело бы к значительному сокращению числа промежутков на большинстве шагов и, таким образом, уменьшило бы трудоемкость алгоритма. Так как F^* *apriori* неизвестно, то требуемую величину длины промежутка можно оценить снизу.

Например, решаем задачу эвристическим алгоритмом с трудоемкостью $O(n \log n)$ на основе лексикографического метода [6]. Полученное значение F_λ целевой функции F на решении по методу [6] позволяет оценить снизу длину промежутка величиной $d_\lambda = \frac{F_\lambda \varepsilon}{n}$. Таким образом на шаге K $(K=1,2,\dots,n)$ можно использовать промежутки

$$\bar{\Delta}_j^K = \begin{cases} [0, d_\lambda], & j=1, \\ ((j-1)d_\lambda, j d_\lambda], & j=2,\dots,N^K, \end{cases}$$

где $N^K = \left[\frac{C^K}{d_\lambda}\right] + 1,$

до тех пор, пока $C^K \leq F_\lambda$.

Подойдем теперь к рассмотренному выше алгоритму с точки зрения возможности включения в вычислительную процедуру эксперта, т.е. построения человеко-машинной, диалоговой процедуры решения. Последние несколько лет таким процедурам уделяется много внимания [5], что связано, во-первых, с неполнотой информации, существующей на этапе формализации, и, во-вторых, с чрезвычайно высокой вычислительной сложностью практических задач. На основе интуиции и опыта эксперт в процессе решения в состоянии выявлять наиболее предпочтительные направления дальнейшего решения задачи, принимать окончательное решение. Именно с целью исследования возможности построения диалоговой процедуры в K-м шаге алгоритма $(1 \leq K \leq n)$ были выделены преобразования (5) и (6).

Рассмотрим варианты включения эксперта в процедуру решения на различных шагах алгоритма. На шаге 0 возможно задание условия, при котором некоторый набор элементов исходного множества обязательно должен входить в решение задачи. В этом случае работа алгоритма начинается с некоторого непустого множества, что не влияет на постановку задачи, а только уменьшает размерность. На шаге n можно потребовать от вычислительной процедуры предоставления эксперту не одного, а ряда решений с наилучшими значениями целевой функции, что позволит осуществить анализ и выбор лучшего решения.

На других шагах алгоритма возможно включение в вычислительную процедуру следующих функций для реализации процесса диалога:

- предоставление эксперту на K-м шаге набора $\mathcal{S}^n$;
- возможность задания экспертом дополнительного условия, по которому некоторому $x^K \in \mathcal{S}^K$ на дальнейших шагах алгоритма отдается предпочтение при преобразованиях (5) и (6) независимо от значений целевых функций и ресурсов;
- возможность исключения экспертом из множества $\mathcal{S}^K$ некоторого $y^K \in \mathcal{S}^K$;

- возможность задания шагов алгоритма, на которых происходит осуществление приведенных выше функций.

Отметим, что впервые возможности учета неформализуемых ограничений в методах диалогового (полилогового) программирования описаны в [7] и [8].

Включение в вычислительную процедуру указанных выше функций, реализующих диалог, технически легко осуществимо. При этом трудоемкость изменяется не более, чем на мультипликативную константу. Такая диалоговая модель дает возможность решить задачу за время, ограниченное полиномом от размерности задачи (при фиксированном ε). Диалоговая модель позволяет эксперту получить представление о том, к каким последствиям может привести задание некоторых предпочтений между наборами элементов исходного множества. Это имеет большое значение для понимания конкретной задачи. Главным преимуществом предлагаемого диалогового подхода является то, что непосредственное включение эксперта в вычислительную процедуру предоставляет возможность учета дополнительных неформализуемых особенностей задачи.

Литература

1. Автоматизация процессов управления на универсальных базах материально-технического снабжения. М.,"Машиностроение", 1977.
2. Бабат Л.Г. Приближенное вычисление линейной функции на вершинах единичного n -мерного куба. - В кн.:Исследования по дискретной оптимизации. М., "Наука", 1976.
3. Журавлев Ю.И., Финкельштейн Ю.Ю. Сфера применения методов дискретного программирования.

В кн.: Применение исследования операций в экономике. М., "Экономика", 1977.

4. Карп Р. Сводимость комбинаторных задач. - В кн.: "Кибернетический сборник", новая серия, № 12. М., "Мир", 1975.

5. Ларичев О.И. Человеко-машинные процедуры принятия решений (обзор). "Автоматика и телемеханика", 1972, № 12.

6. Саата Т. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. М., "Мир", 1973.

7. Юдин Д.Б. Лекции по исследованию операций для студентов отделения экономической кибернетики МГУ. М., МГУ, 1977-1978.

8. Юдин Д.Б., Немировский А.С. Информационная сложность и эффективные методы решения выпуклых экстремальных задач. "Экономика и математические методы", 1975, № 3.

9. Ibarra, C. Kim, Fast Approximation Algorithms for Krapsack and Sum of Subsets Problems, J. ACM, vol 22, No 4, Oct. 1975.

10. S. Sahni. General Techniques for Combinatorial Approximations, Techical Report, 76-6, June, 1976, Institute of Technology University of Minnesota, Minneapolis.