УДК 681:658.56

БОЛЬШИЕ ДИСКРЕТНЫЕ СИСТЕМЫ ИСХОДНЫХ ДАННЫХ В ИССЛЕДОВАНИИ ОПЕРАЦИЙ

LARGE DISCRETE SYSTEMS OF INTIAL DATA IN OPERATION RESEARCH

В.В. КЛЕЙНОСОВ
V.V. KLEYNOSOV

(Российский государственный университет им. А.Н. Косыгина (Технологии. Дизайн. Искусство))
(Russian State University named after A.N. Kosygin (Technologies. Design. Art))
E-mail: office@msta.ac.ru

Постановка оптимизационных задач динамического или линейного программирования сопряжена с проблемой выбора исходных данных (ИД) для их решения. ИД должны быть достоверны, должна быть известна история их получения. Затраты на их получение не должны превышать экономического эффекта, полученного в результате решения.

Репрезентативность объема выработки ИД ограничена, с одной стороны, невозможностью получить неинтересный для практики локальный результат, с другой – возможностью все-таки решить задачу оптимизации с помощью современных ЭВМ. Вводимые в статье большие дискретные системы (БДС), возможно, способны удовлетворять всем перечисленным выше компромиссным требованиям.

Determination of optimization problems of dynamic or liner programs is related to the initial data (I.D.) selection problem for their solution. Such I.D. should be trustworthy with known history of their obtaining, and the cost of their obtaining should not exceed the economic effect, resulting from the problem solution.

Representativity of the sample scope of the I.D. is limited, on the one hand, the undesirability to obtain any local results of any interest, and on the other hand by the possibility to solve the optimization problem by means of superpower computer. The large discrete systems introduced in this article are possibly capable to satisfy all the above compromise requirements.

Ключевые слова: дискретные системы, операция, исходные данные, свойства больших систем, достоверность исходных данных, принцип соответствия, оптимальный, локальный минимум, эффективность.

Keywords: discrete system, operation, initial data, internal of big systems, reliability of initial data, principle of coordination, optimal, local minimum, effectiveness.

Вначале рассмотрим случай, когда система исходных данных (БДС) описывается с разными, но имеющими общий смысловой оттенок (время), понятиями.

Такими понятиями являются: "год", "месяц", "день", "число". Каждое последующее понятие, за исключением эквивалентных понятий "день", "число", вложено в предыдущее понятие.

Для описания понятия "год" используется четырехзначное число, номер месяца представляется одним, двумя римскими цифрами, число месяца одним, двумя арабскими цифрами от 1 до 31, дни недели – двумя буквами (Пн, Вт, Ср, Чт, Пт, Сб, Вс).

На запоминающем устройстве в электронном виде хранится большой список особо важных событий (в том числе секретных) и точные даты их свершения, например: (Вс, 22, VI, 1941).

Случайно попавший электрический сигнал частично уничтожил и перемешал имеющуюся информацию так, что сохранившаяся ее часть могла быть представлена только в виде хаотически разбросанного, далеко не полного, ряда ее отдельных оставшихся элементов, представленных ниже:

19, Пт, 6, 12, II, 2009, 1, Вс, 8, X, 26, Ср, V, 5, Пн, 2, 4, 30, 31, 2013, XII, 7, 2011, 10.

Требуется – как можно полнее восстановить потерянную информацию. Самым простым здесь может показаться метод перебора всех возможных вариантов C_{24}^4 и выбора из них тех последовательностей четырех элементов, представляющих собой год, месяц, день, число, которые совпадут с документальным их подтверждением в календаре соответствующего года (имеющегося или расчетного). Предложенная процедура имеет очень малый коэффициент эффективности. Действительно, среди четверок могут содержаться только дни ($C_7^4 = \frac{7!}{3!4!} = = \frac{5\times6\times7}{1\times2\times3} = 35$) или только числа ($C_{31}^4 = = \frac{28\times29\times30\times31}{1\times2\times3\times4} = 31465$), или только месяцы ($C_{12}^4 = \frac{12!}{8!4!} = \frac{9\times10\times11\times12}{1\times2\times3\times4} = 495$).

Для подсчета "лишней" информации в четверках, состоящих только из годов, требуется "гибкая" дополнительная информация.

Если взять информацию за какие-либо произвольно выбранные десять лет, то $C_{10}^4 = \frac{10!}{6!4!} = \frac{7\times8\times9\times10}{1\times2\times3\times4} = 210$.

Другой подсчет вариантов можно осуществить, замечая, что в каждой четверке символы г, м, д, ч могут встречаться только по разу, причем границы изменения величин ч, д, м известны априори, это 31, 7, 12. Верхняя граница L для величины г априори не известна, но может быть назначена. Если, например, положить L=10, то это количество различных четверок (г, м, д, ч) будет $10 \times 12 \times 7 \times 31 = 26040$. По сравнению с величиной $C_{24}^4 = \frac{24!}{20!4!} = = \frac{21\times22\times23\times24}{24} = 10626$ эта величина больше. Если, например, для отыскания полной одной нужной даты потребуется 10 с, то на совершение 10626 операций потребуется $\frac{106260}{3600} \approx 42$ непрерывных и часовых рабочих дня, а это около двух месяцев при рабочей неделе с двумя выходными днями.

Этот результат получен по понятным причинам для малой (24) выборки исходных данных. Можно себе представить объем работы для выборки тысячи и более исходных данных.

Можно ли сократить объем сравнений рассмотрев, например, приведенную выше выборку в 24 единицы.

Попробуем воспользоваться идеями динамического программирования, поэтапного принятия решения.

Будем рассматривать на первом этапе, независимо от других, значения величины "г", на втором – значение величины "м", на третьем – "д", на четвертом – "ч".

На первом этапе система исходных данных займет только три значения 2009, 2011, 2013, записанных по возрастанию. На втором этапе система характеризуется вектором м(II, V, X, XII), на третьем – д(Пн, Ср, Пт, Вс), на четвертом – самый длинный ч(1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 19, 26, 30, 31).

Предстоит пересмотреть $3 \times 4 \times 4 \times 14$ дат, однако все они рассматриваются на соответствующих деревьях.

(2009,II,Пн,2); (2009,II,Ср,4); (2009,II,Пт,6); (2009,II,Вс,8); (2009,V,Пн,4); (2009,V,Ср, 6); (2009,V,Пт,1); (2009,V,Пт,8); (2009,V,Вс,10); (2009,V,Вс,31); (2009,X,Пн,5); (2009,X,Пн,12); (2009,X,Пн,19); (2009,X,Пн,26); (2009,X,Ср,7); (2009,X,Пт,2); (2009,X,Пт,30); (2009,X,Вс,4); (2009,XII,Пн,7); (2009,XII,Ср,2); (2009,XII,Ср,30); (2009,XII,Пт,4); (2009,XII,Вс,6).

Все вышеописанные 24 восстановленные даты могут быть выписаны в течение часа.

Выпишем календарные данные, определяемые 2011 годом:

(2011,II,Пн,7); (2011,II,Ср,2); (2011,II,Пт,4); (2011,II,Вс,6);(2011,V,Пн,2); (2011,V,Пн,30); (2011,V,Ср,4); (2011,V,Пт,6); (2011,V,Вс,1); (2011,V,Вс,8); (2011,X,Пн,10); (2011,X,Пн,31); (2011,X,Ср,5); (2011,X,Ср,12); (2011,X,Ср,19); (2011,X,Ср,26); (2011,X,Пт,7); (2011,X,Вс,2); (2011,X,Вс,30); (2011,XII,Пн,5); (2011,XII,Пн,12); (2011,XII,Пн,19); (2011,XII,Пн,26).

Их будет 23. Как показал опыт, решение задачи на графах малоэффективно ввиду насыщенности изображения.

Таблица 1

	Пн	Ср	Пт	Вс
II				
V				
X				
XII				

В связи с этим рассмотрим метод сравнения соответствующих столбцов на матрице для 2013 г. (табл. 1).

Сравнивая текущий вектор (1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 19, 26, 30, 31) с документальными (календарными) векторами:

(4,11,18,25); (6,13,20,27); (1,8,15,22); (3,10,17,24); (6,13,20,27); (1,8,15,22,29); (4,11,18,25); (5,12,19,26); (7,14,21,28); (2,9,16,23,30); (4,11,18,25); (6,13,20,27); (2,9,16,23,30); (4,11,18,25); (6,13,20,27); (1,8,15,22,29),

можно выписать следующие даты:

(2013,II,Пн,4); (2013,II,Ср,6); (2013,II,Пт,6); (2013,II,Пт,8); (2013,II,Вс,10); (2013,V,Пн,6); (2013,V,Ср,1); (2013,V,Ср,8); (2013,V,Пт,10); (2013,V,Пт,31); (2013,V,Вс,5); (2013,V,Вс,19); (2013,V,Вс,26); (2013,X,Пн,7); (2013,X,Ср,2); (2013,X,Ср,30); (2013,X,Пт,4); (2013,X,Вс,6); (2013,XII,Пн,2); (2013,XII,Пн,30); (2013,XII,Ср,4); (2013,XII,Пт,6); (2013,XII,Вс,1); (2013,XII,Вс,8).

Их будет 24.

Таким образом, восстановлены все возможные даты (их $24 + 23 + 23 = 71$) важнейших событий.

Следует отметить, что без применения метода "Соответствия" фундаментальному документу (календарю) решение задачи восстановления было бы невозможно.

Имеется n плоских лекал произвольных форм и размеров, которые требуется уложить экономно на рулон заданной ширины H.

Для сравнения укладок между собой и выбора из них наилучшей необходимо ввести понятие эффективности укладки.

Эффективность укладки лекал на рулон определим как отношение:

$$Э = \frac{S_л}{S_{пр}},$$

где $S_л$– суммарная площадь уложенных лекал на рулон к площади $S_{пр}$ прямоугольника, описывающего произведенную укладку.

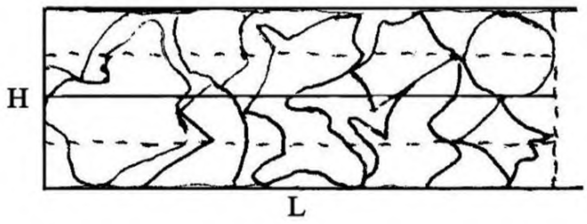

Рис. 1

Сделав укладку легко посчитать $S_{пр} = HL$, измерив L. Однако расчет $S_л$ нетривиален. Вряд ли следует рекомендовать априорное вычисление всех площадей лекал, подлежащих укладке, особенно когда n велико, лекала причудливы и очень разнообразны по форме, особенно тогда, когда лекала склонны менять свою форму во времени.

В этом случае более уместным, пожалуй, будет считаться метод равномерных сечений высоты H линиями, параллельными основанию, окаймляющему укладку прямоугольника. Можно организовать и последовательность разбиений: сначала прямоугольник разбивается на две части одной горизонтальной линией, и вычисляется сумма длин хорд пересекаемых ею лекал, затем на 4, на 8 и т.д., при этом вычисляется средняя арифметическая сумм длин хорд по каждому разбиению. Среднее арифметическое длин хорд 2n разбиения:

$$C_{2n-1} = \sum_{i=1}^{2n-1} \left(\sum_{x} (\ell_x^i)\right)/(2n-1)$$

будет стремиться при $n \to \infty$ к длине $L_л$ равновеликого прямоугольника, составленного из лекал укладки, имеющего, как видно из рис. 1, высоту H, а величина $\frac{L_л}{L_{пр}}$ будет стремиться к искомой величине Э.

При отыскании значения величины Э нужно определиться, со сколькими верными цифрами после запятой $(0 < Э < 1)$ она будет вычисляться. Вычисления C_{2n-1} прекращаются, когда все значения C_{2n-1} и C_{2n+1} полностью совпадут. Учитывая, что вычисления C_{2n+1} полностью используют значение C_{2n-1}, полученные на предыдущем шаге, поиск величины C_{2n+1} удобно проводить на компьютере, составив соответствующую программу [2].

Для очень плотных укладок вычисление легко осуществить визуально, имея раскладку, представленную на миллиметровке или на тетради в клетку.

Для одного (среднего) сечения, представленного на рис. 1, имеем:

$$\sum_x \ell_x^1 = 14 + 4 + 4 + 2 + 6 = 30, \text{ (L=55)},$$
$$Э \approx \frac{30}{55} = \frac{6}{11} \approx 0{,}54.$$

Для трех сечений (два из них отмечены пунктиром):

$$\sum_x \ell_x^3 = 13 + 4 + 7 + 8 + 9 = 41,$$
$$c_3 = \frac{30+47+41}{3} = 39\frac{1}{3} \approx 40,$$
$$Э \approx \frac{40}{55} = \frac{8}{11} \approx 0{,}74 \text{ ч.т.д.}$$

Из сказанного выше видно, что нерешаемая аналитически задача, используя неопределенность и размах исходных данных, будучи поставлена в рамки проводимой операции, приобретает совершенно иную окраску, имеющую свой специфический способ решения. Ситуация, напоминающая закон больших чисел, сама порождает простой метод решения.

ВЫВОДЫ

Специфика БДС, состоящей из четырехмерных векторов с координатами различной природы первого примера, позволила организовать несколько параллельно идущих пошаговых процесса.

Предполагаемая раскладка элементов БДС, состоящей из множества лекал различного вида и размеров на рулон, во втором примере позволила организовать на предстоящей "вытянутости" множества лекал итерационный процесс.

Суть процесса предполагает вычисление средних значений суммы длин хорд равномерных разбиений ширины лекала.

ЛИТЕРАТУРА

1. *Ефимов Н.В., Розендорн Э.Р.* Линейная алгебра и многомерная геометрия. – М.: Наука, 1970.
2. *Беллман Р.* Динамическое программирование. – М.: ИЛ. 1960.
3. *Клейносов В.В.* Оптимизация укладки лекал на рулонном материале. – М.: МГАЛП, 1994.
4. *Клейносов В.В., Кучер И.В.* Расчет эффективности раскладки материала с помощью микро-ЭВМ. – М.: МТИЛП, 1986.

REFERENCES

1. Efimov N.V., Rozendorn Je.R. Linejnaja algebra i mnogomernaja geometrija. – M.: Nauka, 1970.
2. Bellman R. Dinamicheskoe programmirovanie. – M.: IL. 1960.
3. Klejnosov V.V. Optimizacija ukladki lekal na rulonnom materiale. – M.: MGALP, 1994.
4. Klejnosov V.V., Kucher I.V. Raschet jeffektivnosti raskladki materiala s pomoshh'ju mikro-JeVM. – M.: MTILP, 1986.

Рекомендована кафедрой высшей математики. Поступила 29.03.16.