НОВЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ

Болотов В.П.

"Следует поставить перед собой цель изыскать способ решений всех задач ...одним и притом простым способом" Даламбер

В практической деятельности каждый человек сталкивается с необходимостью принимать наилучшее (оптимальное) решение. Многие законы природы, связанные с распространением света, электричества, жидкости, газа и т.п. можно вывести на основе понятий минимизации. Так, например, **принцип Ферма** гласит: в неоднородной среде свет распространяется по такой траектории, что время, затрачиваемое им на преодоление пути от одной точки до другой, минимально.

Задачи на нахождение максимума/минимума играли важную роль в истории науки. За это время накопилось огромное число красивых, важных, ярких и интересных задач в геометрии, алгебре и физике. Решения этих задач искали крупнейшие ученые прошлых веков: Евклид, Архимед, Аполлоний, Герон, Тарталья, Торричелли, Иоган и Якоб Бернулли, Ньютон, Ферма, Эйлер и многие другие. Нахождение решений стимулировало создание таких теорий как дифференциальное и вариационное исчисление, оптимальное управление. Широкое применение подобных задач в практике связано с необходимостью разработки простых алгоритмов их решения, не требующих глубоких математических знаний.

Компьютерные технологии обеспечивают реализацию мысли Даламбера, приведенной в эпиграфе. В основу решения оптимизационных задач можно положить простую идею: *для наилучшего решения надо перебрать все возможные решения, сравнить их между собой и выбрать из них наилучшее.*

Перебора огромного количества возможных решений раньше всегда пытались избежать. Однако для ЭВМ с ее быстродействием в миллионы операций в секунду такой перебор не сложен. При этом число переборов всегда можно уменьшить, сузив перебираемую область или размерность перебираемых переменных. При такой реализации главной является формализация (алгоритмизация) на языке ЭВМ понятий: *что перебрать, как перебрать и как выбрать нужное решение.* Чаще всего перебираются одномерные или многомерные (вектора) величины, связанные между собой известными законами физики или экономики.

Пример1. В заданном наборе чисел определить наименьшее число и построить график зависимости величины числа от его номера. Пусть даны 20 чисел, значения которых в системе "Вектор" можно записать в регистры sn (n - любое целое число) вещественных чисел:

```
: s1=70. s2=50. s3= 80. s4=60. s5=10. s6=20. s7=5. s8=70. s9=90.
: s10=35. s11=15. s12=76. s13=54. s14=76. s15=125. s16=165.
: s17=66. s18=105. s19=39. s20=84.
```

Алгоритм решения основан на **переборе, сравнении и выборе** и заключается в следующем. Для сравнения сначала задаем число, заведомо больше заданных чисел (например s99=200.). Берем первое число из регистра s1 и сравниваем с начально заданным числом в s99. Если число в s1 меньше числа в s99, то число в s1 помещаем в регистр s99 (переприсваеваем). Берем второе число в s2, также сравниваем с новым числом в s99, и если, оно меньше, то меняем его на меньшее (в противном случае присвоение не выполняется) и так до тех пор, пока не переберем все числа. На последнем этапе окажется, что наименьшее число лежит в регистре s99. На языке "Калькулятор" системы "Вектор" перебор можно организовать в цикле. Организация цикла основана на рекурсивном подходе - последовательном вызове макрокоманды самой себя до тех пор пока какой-то текущий параметр не станет больше наперед заданного. В примере с числами таким параметром можно выбрать число чисел номера которых лежат в регистрах n. Наибольший номер чисел равен 20.

Ниже приведены две макрокоманды, одна из которых - головная (запускается первой) выполняет функции присвоения исходных данных и вызова МК цикла.

```
<hisl0>
: s1=70. s2=50. s3= 80. s4=60. s5=10. s6=20. s7=5. s8=70. s9=90.
```

: s10=35. s11=15. s12=76. s13=54. s14=76. s15=125. s16=165.
: s17=66. s18=105. s19=39. s20=84.
hislo1 : s99=200. n=1

<hislo1>

n > 20 ? exit
sn < s99 ? s99=sn
hislo1 : n=n+1
Наименьшее число будет лежать в s99.

Вторая часть задания - построить график зависимости значений чисел от их номера. Такой график можно построить в виде ломаных отрезков. Последовательность построения отрезков можно вставить непосредственно в текст МК.

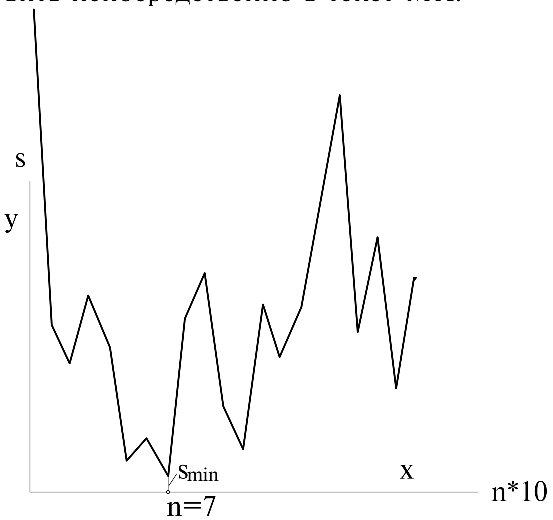

Рис.1

Пример 2. Определить маршрут движения путника по скошенному и нескошенному лугам из точки ***А*** в точку ***Б*** так, чтобы маршрут его движения был наиболее коротким и время его прохождения было наименьшим. Скорость движения по скошенному лугу равна ***v***1, по нескошенному - ***v***2.

1. ***Анализ:*** исходя из формул определяющих взаимосвязь скорости, времени и расстояния (***v=s/t***), определяем, что время нам неизвестно и его удобно выбрать в качестве минимизирующего параметра.

2.1. ***Алгоритмизация:*** общее время движения путника по лугу 1 и лугу 2 определяется суммой ***t***1 и ***t***2 :

$$t = t1 + t2 = |p - p1| \;/\; v1 + |p - p2| \;/\; v2,$$

где

$$p = (1-u)p11 + up12.$$

2.2. ***Формируем ЦФ***:

$$f(u) = \frac{\sqrt{(x-x1)^2+(y-y1)^2}}{v1} + \frac{\sqrt{(x-x2)^2+(y-y2)^2}}{v2}$$

3. ***Решение*** заключается в формировании ЦФ (построении ее графика), нахождению ее минимального значения (будет соответствовать минимальному значению времени, за которое путник пройдет весь путь) и нахождении ему маршрута движения (рис. 2).

Рассмотрим пример с нахождением кривых линий определенных свойств (задача о брахистохроне, кривой наименьшего сопротивления в заданной среде, кривой отражения и т.д.).

Обычно эти задачи решаются на основе дифференциальной оптимизации и считаются трудными. Однако умение формализовать задачу и формировать ее ЦФ делает возможным решение таких задач без привлечения дифференциальной геометрии

Пример 3. Задача о брахистохроне. Найти кривую наискорейшего спуска. На этапе анализа замечаем, что скорость движения падающих тел зависит от времени падения $v = gt^2$.

Можно в точке на горизонтали, разделяющей пространство на две части принять, что скорость в верхней части определена ***v***1($v_1 = \sqrt{2g_1y}$,), а в нижней ***v2*** ($v_2 = \sqrt{2gb}$). Таким образом, задачу свели к задаче движения по скошенному и нескошенному лугу.