数学模型的建立和选择

骆 骥
(*芜湖一中, 安徽, 241000*)

目 录

【关键字】信息原型 数学模型 建模

【摘 要】

本文主要探讨的是信息学竞赛中解题的关键：数学模型的建立和选择。首先分析了从信息原型到数学模型的重要性，提出了解题的简单过程：现实——理论——现实。然后将数学模型的建立方法分为机理分析法和统计分析法两类，并着重分析了在竞赛中常用的机理分析法建模。接着，文章又对数学模型的选择做了一些必要的概述，得出一些模型选择的原则。最后，根据整篇文章探讨的结果得出了数学模型建立和选择的一般方法。

【正 文】

一、 从信息原型到数学模型

在大千世界中，我们所面对的事物形形色色，扑朔迷离。它们都是由许多信息构成的。这些现实世界中对客观问题表面的自然语言描述，称为**信息原型**。当然，在信息学竞赛中我们所面临的问题也是信息原型。

信息原型本身是由扑朔迷离的信息构成，掩盖了其重要的属性[1]。我们因而无法直接从信息原型入手找到问题的答案。为此，我们就需要一种方法来“改造”信息原型，使之既具有原来的重要属性，也具有可研究性。

于是，我们试图将信息原型的属性一起转移到一个**模型**中。模型即是对客观问题属性的模拟。显然，这个对应出的模型可以说是信息原型的代表。我们就可以对这个模型进行研究。

用什么方法将信息原型对应到模型上去呢？我们期望运用数学方法。这样对应出的模型即具有原问题的属性又具有数学的可研究性。我们称之为**数学模型**。**数学模型**：运用数学语言对信息原型通过抽象加以翻译归纳的产物叫做数学模型。

信息原型是现实的问题，对应到的数学模型又是理论上的模型，对该模型进行研究使我们得出了现实问题的解。这就是信息学竞赛中解题的简单过程：现实——理论——现实。如图一所示：

为了能快速地从信息原型得到信息原型的解，在整个分析解题的过程中，从信息原型到数学模型的这一转变过程至关重要。根据图一，我们称该过程为**建模**：利用数学思想将信息原型转化成数学模型的过程。

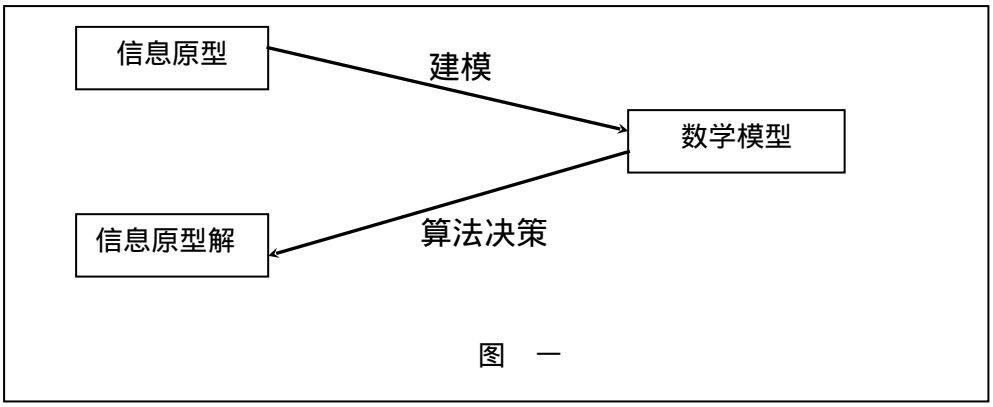


图 一

下面，我们就将讨论：如何从信息原型到数学模型。

二、 数学模型的建立

从实践中积累的经验，我们知道，建模没有固定的套路可言，方法比较多样化。但总的来说一般分为**机理分析法**和**统计分析法**两大类。

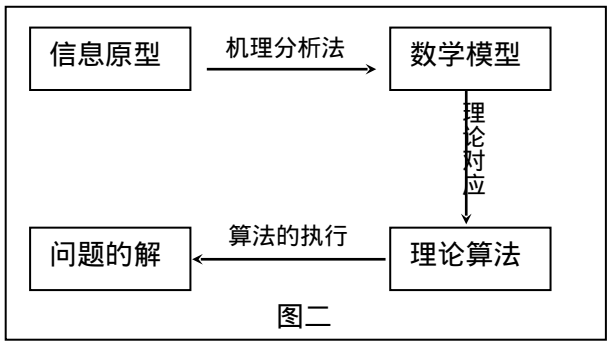
§2.1 机理分析法

【定 义】

机理分析法：根据客观事物的特性，分析其内部的机理，弄清关系，在适当抽象的条件下，利用合适的数学工具得到描述事物属性的数学模型的方法。

【图表描述】

我们在信息学竞赛中常用这种方法建立模型，然后根据所对应的算法求出解。如图二所示：



图二

由信息原型，我们运用机理分析法通过抽象建模得出数学模型，再根据得出的数学模型理论对应到算法，编程实现，通过算法的执行得到问题的解。

【分 类】

机理分析法也是多种多样的，我们在实际竞赛中往往各种机理分析方法混用，所以分类比较困难。根据建模的几个不同层次特点，一般将其分为四类：

§2.1.1 直接建模法

【诠 释】

当信息原型比较简单，其属性显而易见时，我们通常用直接建模法。该方法可以说是将信息原型的显而易见的属性按数学方法综合起来得出模型。这个得到

的数学模型针对性强，不具有普遍意义。

【分 析】

实际上，该直接建模法是一种创造。但思维比较简单，没有理论上的新发现我们称为**一级创造**。例如 NOI'97 的竞赛排名一题，信息原型的各个属性十分明确，题目本身也有一定的抽象程度，性质大都由数学语言描述，更有利于我们直接建立简单的模型解题。在竞赛中我们一般遇到的这类题目主要考察我们怎样较好地将信息原型已知的属性按数学方法综合起来。

建模方法既然有创造，就有摹仿。我们还经常摹仿已知的模型来建模。下面就要介绍两种摹仿建模的方法。

§2.1.2 套用常用模型法

【诠 释】

建模时，我们抽象数学模型的方向往往向已知的常用模型上靠拢，而且一旦符合，就直接建立已知的模型，并且完全套用其算法。

【举例分析】

下面我们就结合 NOI'99 的 01 串问题进行分析。

例 1 01 串问题[2]

这道试题信息原型很好地将数学模型遮掩了起来，使我们比较难从中理出头绪。首先还是来分析一下该题的信息原型的主要属性：

设序列 S 为问题的一个解，则其一定满足：

1. s_i=0 或 s_i=1，1<=i<=N；
2. 对于 S 的任何连续的长度为 L0 的子串 $s_j s_{j+1} \ldots s_{j+L0-1}$(1<=j<=N-L0+1)，0 的个数大于等于 A0 且小于等于 B0；
3. 对于 S 的任何连续的长度为 L1 的子串 $s_j s_{j+1} \ldots s_{j+L1-1}$(1<=j<=N-L1+1)，1 的个数大于等于 A1 且小于等于 B1。

显然，上述的属性描述运用的是自然语言，不抽象，更无法运用数学知识进行研究。我们期望将信息原型继续转化。这就运用到常用的"求部分和序列"技术。

根据一个序列 S 来构造一个序列 T，满足：

$$T_i = \begin{cases} 0 & (i = 0) \\ \sum_{j=1}^{i} s_j & (i > 0) \end{cases}$$

易知，S 和 T 是一对一的映射关系。对于序列 S，我们总结出三个条件要被满足，那么对应到序列 T，就应满足如下条件：

1. $0 \le T_i - T_{i-1} \le 1$ $(1 \le i \le N)$
2. $A0 \le L0 - (T_{j+L0} - T_j) \le B0$ $(0 \le j \le N-L0)$
3. $A1 \le T_{j+L1} - T_j \le B1$ $(0 \le j \le N-L1)$

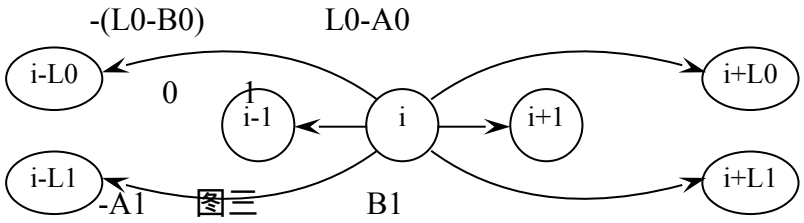
显然，同上面的自然语言描述相比，序列 T 应满足的条件充分运用了数学语言描述，比信息原型是大大抽象了。但我们仅完成了建模的第一步，对信息原型用适当的数学语言进行了描述。问题仍然难以求解。

我们面临的问题是求解不等式组。这使我们联想到 IOI97 中国组队赛的"工程

规划”一题。本题是不是求最短路径问题呢？有了这个猜测，我们不禁要往该方向上进行分析。求最短路径问题是基于图的基础上的。我们就要想方设法将信息原型对应到图论中的模型上去。（一级摹仿的体现）

注意到序列 T 应满足条件的三个不等式，每个不等式都只涉及到两个变量。这就可以对应到图中的边：两个点表示两个变量，两个点之间的边表示变量之间的关系（即不等式）。

从一个点 V_i 发出的边可能如下图三所示：



这就构造出了一个图。令 $T_0=0$，T_i=图中 V_0 到 V_i 的最短路径长度。问题就迎刃而解了。求最短路径时，有负边，我们就采用迭代法。一个图论模型就这样建立了，同时，我们也对应找到了有效算法（附程序 *Sequence.pas*）。

【小　结】

解这道题目，我们用的是“套用常用模型方法”。联想到已知的模型，从而成为我们分析的方向。然而我们在建模时一味摹仿，模型和算法都没有任何独特之处。所以，我们的这种摹仿方法还是比较低级的，称之为**一级摹仿**。

一级摹仿是我们常用的建模方法，我们运用它将信息原型和常见的已知模型对应，但这在某种程度上限制了我们的创造力，同时，建立的模型往往生搬硬套，欠缺应变改进能力。我们期望在摹仿中加入自己的创造因素，使模型具有针对信息原型的独特属性。这就是我们将要分析的第三种机理建模法：针对修改常用模型法。

§2.1.3 针对修改常用模型法

【诠　释】

一个信息原型一定有它独特的属性。所以，我们要适当修改套用的常用模型将独特的属性加入，以试图优化模型和算法[3]。

【举例分析】

下面我们来简单分析一下如何有效掌握利用信息原型的属性。

例 2 求第 k 大的数

已知 n 个数字各不相同，求其中第 k 大的数是多少？

（1<=k<=n<=10000）

这是一道简单的试题，我们完全可以套用常用的快速排序模型解决如下：对所有数字进行排序，然后取出第 k 大的数字，输出即可。

该算法的时间复杂度为 O($n\log_2 n$)

那么有没有其他方法呢？信息原型中毕竟有它独特的属性：*是求第 k 大的数，不是求全部数的有序排列*。我们将这点独特属性考虑到快速排序中去。

快速排序的基本思想关键在于不断调整使分治点左边的数不大于（或不小于）分治点，右边的数不小于（或不大于）分治点。

一般快速排序，当我们找到一个分治点 x，序列就以 x 为分治点，分成了左右两部分，我们要分别对这两部分继续进行排序。可是，本题仅仅要我们找第 k 大的数，那就意味着：

当 $x>k$ 的时候，第 k 大的数一定在左边部分，右边部分就不用继续排序了；

当 $x<k$ 的时候，第 k 大的数一定在右边部分，左边部分就不用继续排序了；

当 $x=k$ 的时候，我们就找到答案，不用继续排序了。

原来的快速排序每次要递归调用排序两个部分。而这种“变异”了的“快速排序”每次最多只要递归调用排序一个部分。

假设我们每次找到的分治点接近于中点（这可以用随机化大致保证），那么，其算法的时间代价大致为：n+n/2+n/4+n/8+……=2n。也就是说：改进后的算法时间复杂度为 O（n），比 O（$n\log_2 n$）显然降低了一级。（附程序 *sort.pas* ）

【小　结】

在上面的简单的模型设计中，我们运用了“针对修改常用模型法”，注意该方法的运用主要讲究“针对”而字，我们有效地找出信息原型的独特属性，并且有效地在算法中加以应用。

“针对修改常用模型法”不但体现了解题者将信息原型对应到常用模型的能力，更考察了解题者对信息原型本身的特殊属性的有效掌握，在某种程度上也表现出创造的意识。

所以，该摹仿方法是在一级摹仿基础上的进步，我们称之为**二级摹仿**。虽然二级摹仿参入了创造因素，但这只是局部的微小改变，摹仿终究还是摹仿。而且我们先前分析的一级创造法是简单的低级创造，有没有更高级的创造呢？有。这就是我们要分析的第四种机理建模法：综合创造法。

§2.1.4 综合创造法

【诠　释】

有很多问题我们很难用摹仿的方法解决。而信息原型的属性又不明显，很难直接建模。这时，就要运用综合创造法了。

综合创造法是根据数学理论知识，运用已知模型或方法来分析信息原型的属性，在此基础上创造出具有新意的模型或方法。该模型或方法具有很强的适用性，可以解决这一类题目。

【举例分析】

正如 NOI’97 的卫星覆盖（Cover）一题，有的选手就大胆地创造出了“离散”模型和算法。这种模型和算法在以后竞赛的某些题目中被广泛运用，例如 IOI98 的图形周长(Picture)一题。

下面我们就通过分析 Cover 一题的模型建立过程，来对综合创造法进行一些

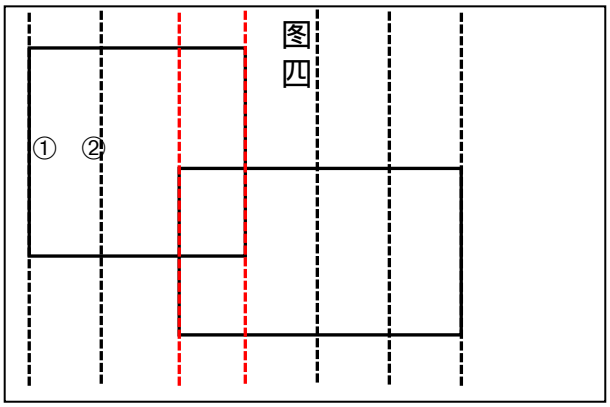
简单的了解。

例 3　NOI'97 卫星覆盖（Cover）

这是一道很好的综合题。它全面考察了选手空间想象能力、数学概念、编程技巧等各方面的综合素质。我们在这里主要分析选手大致是怎样综合创造出"离散模型"的这一片段。

Cover 一题是一道出现的比较早的静态坐标系的数据统计问题。以后的竞赛中经常出现该类题目，数据量大，维数高，则是它们的主要特征。在分析题目的过程中，我们期望能够将大规模的数据按一定原则分割，分而治之——这也是我们解决这类问题的基本思想。这主要是基于"分治法"带给我们的联想。分割问题可以使我们达到"降维"的目的（基于原有方法的启发）。

在实际分割数据的时候，我们又遇到了"怎样分割"这一难点。往往我们按照：对静态坐标系中的元进行**等分**，即在静态坐标系的某个区域内**均匀**地划分网格。这种一成不变的划分方式与问题本身的数据无关，具有盲目性，对于 Cover 一题，显然很不适用。譬如图四（Cover 为三维问题，不便作图，因此举二维的例子的横坐标划分）：



图四

图中两个矩形相互覆盖，如果我们按照盲目的等分思想分割，如虚线所示。显然，①区和②区分割开来毫无意义，即没有降低问题的维数，也没有降低我们思维的复杂度，而且还增加了分割的代价。所以，我们期望能够避免这种类型的分割。于是，我们设想这样一种分割，其分割方式与问题本身的数据有关，显然该方法不像以前的方法具有盲目性，可以说是一种智能的分割（这正是我们大胆的想象）。

两个矩形的横向关系体现在红色虚线上，我们只要按照红色虚线的划分就可以了。这个，我们称之为"离散点"。根据几何体的各个离散点来分割建立离散模型，使得各个子问题之间不再存在拓扑关系，这就是问题的离散化。

对于交错覆盖的数据统计问题，我们可以利用将问题离散化，再分割统计。这就找到了解决这类问题的新的方法：离散模型。它的本质在于：分割，使得子问题"尽量大"而且数目少，但关系简单，效率高。

【小　结】

Cover 一题的"离散模型"建立，就是我们综合创造的结果。

实际的综合创造显然比一级创造更为高级，它具有新意，有新概念，新思想，新方法。我们称为**二级创造**。对于某一个题目要创造出新的通用模型不仅要有一级、二级摹仿的功底，更要有敏锐的洞察力和非凡的创造力。

机理分析法大致分为以上四种。根据图一，对比图二发现机理分析法是简单的顺向思维。我们可不可以用逆向地思维方式来建模呢？当然可以，这就是建模的第二类方法：统计分析法。

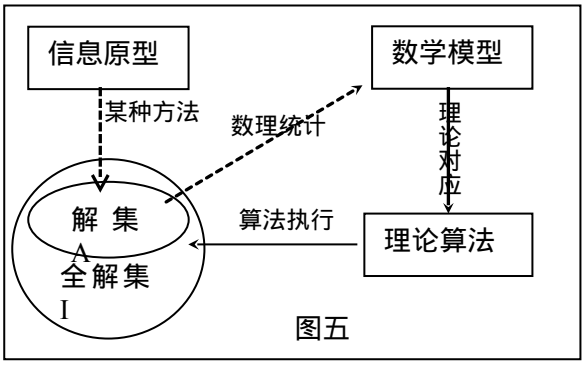
§2.2 统计分析法

【定　义】

统计分析法：我们一时得不到事物的特征机理，便通过某种方法测试得到一些数据（即问题的部分解），再利用数理统计知识对数据进行处理，从而得到最终的数学模型的方法。

【图表描述】

对应到图五中去，就是先从信息原型出发通过手工或简单的程序得到问题的部分解。然后通过对部分解的分析，运用数理统计得到信息原型的主要属性（这里的属性大部分是某些规律），从而建立数学模型，然后通过算法得出问题的全部解。



图五

统计分析法是相对于机理分析法的逆向思维。这种方法在实验性学科中应用十分广泛。

例如在研究 f(x)=sin x 图象时，我们先取一些特殊点再根据其大致走向作出函数图象。

对应到信息学竞赛中，我们一般用手工或简单的程序求出问题的部分解，然后分析出其中的规律，得出数学模型[4]。

【举例分析】

下面的 NOI'95 极值问题的模型建立，大多数选手就是运用了统计分析法，让我们来简单地分析一下。

例4　NOI'95 极值问题

m,n 为整数，且满足下列两个条件：

1　m,n ∈1, 2, 3, ……,k (1≤k≤10^9)；

2　$(n^2-mn-m^2)^2=1$。

由键盘输入 k,求一组满足上述两个条件的 m,n 并且使 m^2+n^2 的值最大。

本题是一道数学性很强的试题。信息原型本身就具有很强的抽象性。这迫使我们一开始便在抽象中分析，没有一个具体的感官概念。面对题目所给的 n，m 关系式这本已抽象的原型，我们不知怎样抽象下去得到问题的数学模型，这自然无法通过机理分析法直接推出模型或对应到常见模型上去。这时统计分析法就有了用武之地。

首先，通过手工测试，我们得出了以下一些小的数据结果：

k	2	3	4	8	16	32	64
N	2	3	3	8	13	21	55
M	1	2	2	5	8	13	34

令人惊奇的是：N，M 随着 k 的增长以斐波拉契数列形式增长！我们于是猜测：问题的解就是小于等于 k 的最大的相临两个斐波拉契数。有了问题的研究方向，我们就朝这个方向去设法证明自己的猜想。

如果我们的猜想正确，那么当(n,m)是方程$(n^2-mn-m^2)^2=1$的一组解时，根据斐波拉契数列关系，(n+m,n)也一定是方程的一组解。

若$((n+m)^2-(m+n)n-n^2)^2=1$成立则：

$=>(n^2+(m+n)n-(n+m)^2)^2=1$

$=>(n^2+mn+n^2-n^2-2mn-m^2)^2=1$

$=>(n^2-mn-m^2)^2=1$成立

以上各步可逆，所以当(n,m)是方程$(n^2-mn-m^2)^2=1$的一组解时，(n+m,n)一定是方程的另一组解[1]。（附程序 *max.pas* ）

【小　结】

原本奇妙的信息原型在简单的测试数据面前被揭开了神秘的面纱。这道题目正是统计分析法的有效运用。但要注意到，不是每个信息原型都能用统计分析法建模：因为我们有时候根本无法求出问题的部分解，或者无法根据问题的部分解用数理统计知识加以分析。所以，在图五中，这两个过程我们用的是虚线。

统计分析法虽然在近年来的竞赛中出现次数不多，但作为与机理分析法相对的一大类方法，在其他众多领域广泛运用，其应用于信息学的潜力也是巨大的。

分析完以上两种方法，我们脑海中显然已经有这么一个思想：既然有多个建模方法，建立的模型也多种多样，那如何选择呢？下面，我们就简单讨论一下如何选择数学模型。

三、　数学模型的选择

【理论描述】

信息原型就好比一个"黑匣子"，我们在里面进行盲人摸象似的探索。角度多种多样，建立数学模型还是有所区别的。如果一个信息原型能对应多个数学模型（如图六所示），这时，我们就面临抉择。

当然，我们期望所选择的模型是最好的。那么这就涉及到对模型的评价问题我们在竞赛中建立模型后要对应到算法，那么对一个数学模型评价自然要着重

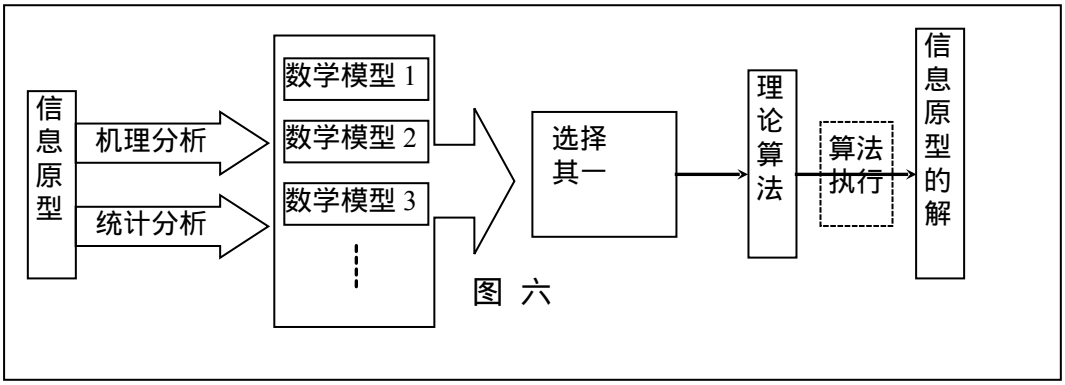


图　六

于其对应算法的好坏。

[1] 该推理证明为作者自己证明，若有错误之处，还请指正。

一般地，评价一个数学模型有以下几个原则：

1． 时间复杂度

一个好的算法一般效率比较高。在竞赛中，试题常常会做一些算法运行时间上的限制。这就要求我们所建立的数学模型对应算法的效率一定要符合要求。这也是最重要的一个原则。

2． 空间复杂度

出于计算机自身的限制，程序在运行时一般只被提供 500k 的内存空间。这也就要求我们建立模型时顾及到这一点。但对于模型对应的算法来说，并不是要求空间越低越好，只要不超过内存限制就可以了。

3． 编程复杂度

相对而言，“编程复杂度”的要求要略低一些。但是在竞赛中，如果建立的算法实现起来十分繁琐，自然不利于比赛。所以，在建立模型时（特别是在竞赛中）这点也要纳入考虑之中。

【分类分析】

面对多种数学模型的选择，大致下有两种情况：

1． 同种思想方法的不同实现方法

例如 IOI’99 的 CODES 一题，同样是动态规划的思想，却有两种不同的实现方法，这两种截然不同的动态规划算法各有千秋。

2． 不同思想方法的各种实现方法

通常来说，有向无环图中的费用流问题在理论上都可以用动态规划来解决。这就造成了可能一道题目对应两种截然不同的思想算法：动态规划和最大流。

当我们面对一道题目既可以用动态规划又可以用最大流的时候，我们一般选取动态规划算法。原因在于，最大流算法的实现是比较困难的，而且在图的建立方面要花费很大功夫，例如 IOI93 的最佳旅行路线一题，邵铮同学所出的《艺术品出售》一题，都是既可用动态规划又可用最大流解决的。

当然，这里不同的思想方法并不仅仅局限于动态规划和最大流，还包括许多其他的思想方法，例如最小生成树有有两种经典的思想方法。这还是要针对实际情况针对分析。

【小 结】

实际上，多种数学模型的选择主要考察的是选手对所要建立的模型的时空复杂度分析能力。在竞赛中，我们很少一下建立几个模型然后从中选择（因为时间不允许），而是在分析中，选择一个较好地建模方向。

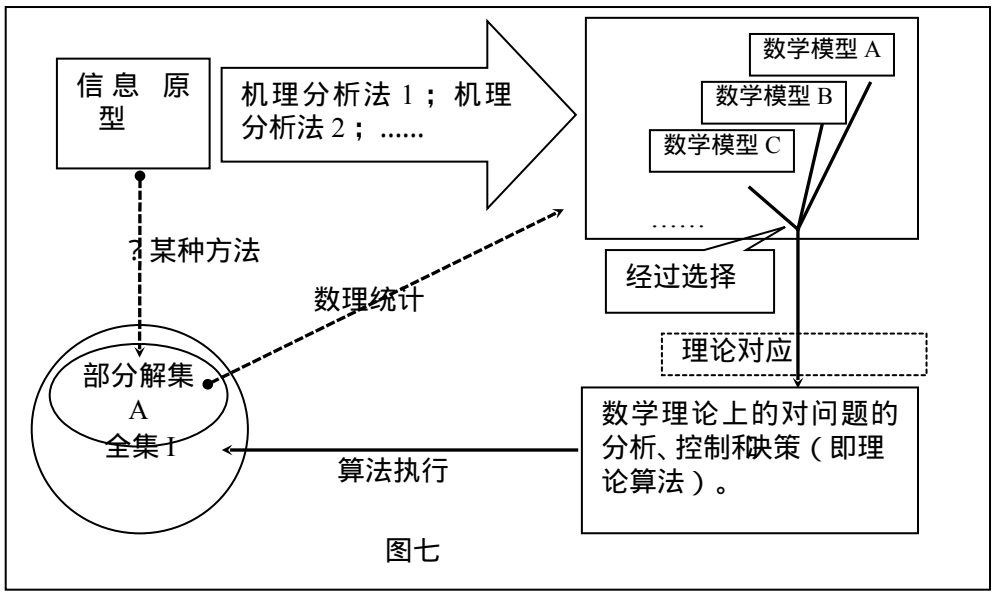
比如我们发现一道题目可能对应几种不同思想的模型，就要根据评价模型的标准来衡量一下，确定一个模型作为分析方向。这时的评价标准除了上述的时间、空间、编程三个标准外，还要加上一个**思维的复杂度**。

所谓思维的复杂度，是指思考所耗费的时间和精力。如果我们确定了一个模型作为分析的方向（没有考虑思维复杂度），从信息原型到该数学模型的建模过程却十分复杂，导致思维耗费时间长，精力多，那自然是不好的。就像上面的例子一样，面对最大流和动态规划两个模型，我们不用最大流，原因还在于其思维的复杂度。最大流所对应的图，点，边，流量，费用等等的意义确定是十分麻烦的，相反，动态规划模型我们较为熟悉，建模的思维复杂度小，因而颇受青睐。总的来说，对于多种数学模型的选择，我们遵循“边分析，边选择，四个

标准来衡量”的原则。

四、 总 结

通过上面的分析，我们大体已经得出了建立数学模型的过程。将上面几幅图结合起来，得到的图七就充分展现了建模解题的过程。我们甚至可以说，解决一个题目就是求图七中从“信息原型”到“全集 I”的“最优路径”。



图七

运用机理分析法建立一个正确高效的数学模型一般按如下步骤进行：

1． 用自己的语言复述问题

这实际上是从自然语言上对信息原型的转化，有利于我们抓住其主要属性；

2． 用数学的语言复述问题

我们抓住了信息原型的主要属性，接着用数学语言描述出来。实际上是将信息原型进行抽象，以建立适当的数学模型；

3． 联想有关的知识

事实上，当我们用数学语言描述信息原型时，如果所得结果简单明了，可以直接形成算法或构造简单模型。我们就运用机理分析法一来解决了（即一级创造）。如果不行，则需要“联想有关知识”。这实际上是往“一级摹仿”靠近。主要联想的是相似的信息原型或存在关系的已知数学模型。

4． 推出有用的事实

如果联想到的知识可以帮我们建立数学模型。那么，我们自然可以套用算法解题了。但进一步延伸下去，也就是达到“二级摹仿”。我们针对该信息原型的特有属性，得出某些“有用”的事实来改进优化模型和算法。

5． 大胆的创造

当然，通过以上四步，我们能够较好的建立一个正确高效数学模型就不错了。但通过上述四步，我们很可能在脑海中酝酿着模糊的新模型。不要埋

没它，大胆的创造，很有可能建立一种新的具有广泛适用性的模型或一套新的方法体系。这也就是我们所说的“二级创造”。

当然，还有统计分析法。一般来说，我们先用机理分析法进行分析，如果机理分析进行不下去的时候在用统计分析法。还有就是当问题的数学性强，容易找到部分简单解的情况下，我们优先考虑统计分析法。

事实上，往往我们的机理分析得出的某些结论会有效的运用于统计分析法；而统计分析法得出的某些规律也可能被运用到机理分析当中。所以，这两者并不是相对孤立的。我们在建模的时候也完全可以将两者混用。

建立完模型，我们任务才完成了一半。如果可建立的模型有多个，还要根据各个模型的时间，空间，编程三个复杂度来选择。最后还要反复不断的将整个模型完善。这才真正建立完一个准确高效的数学模型。

【附 录】

[1] 属性，即一个事物不可缺少的性质。在这里指的是决定信息原型的一切机理特征。

[2] 01 串题目

给定 7 个整数 N,A0,B0,L0,A1,B1,L1，要求设计一个 01 串 S=s1s2…si…sN，满足：

A. si=0 或 si=1，1<=i<=N；

B. 对于 S 的任何连续的长度为 L0 的子串 sjsj+1…sj+L0-1(1<=j<=N-L0+1)，0 的个数大于等于 A0 且小于等于 B0;

C. 对于 S 的任何连续的长度为 L1 的子串 sjsj+1…sj+L1-1(1<=j<=N-L1+1)，1 的个数大于等于 A1 且小于等于 B1;

输入 A0,B0,L0,A1,B1,L1 (3<=N<=1000 , 1<=A0<=B0<=L0<=N，1<=A1<=B1<=L1<=N)。

输出仅一行，若不存在满足所有条件的 01 串，则输出一个整数-1，否则输出一个满足所有条件的 01 串。

[3] 这样的优化在是在时间，空间以及编程复杂度这三者上的优化。

[4] 注意整篇论文所讨论的是模型的建立，而我们认为的所有模型对应的算法是将穷举排除在外的。因为，如果说穷举是属于统计分析法的话，那通过它求得的解集 A 则有 A=I，这就跳过了数学模型建立直接达到目的了。我们所讨论的统计分析法是基于 A ⊂ I 的前提条件下的。所以穷举以及枚举式的搜索不在讨论范围内。

【程 序】

由于篇幅所限，这里仅提供三个例题程序：NOI’99 的 01 串、求第 k 大的数，NOI’95 的极值问题。

```
{NOI’99 01 串问题}
program sequence;
 const
```

```
  infns   =     'input.txt';
  outfns  =     'output.txt';
 type
          Tarr    =     array [1..6] of integer;
          Tdata   =          array [0..1000] of Word;

 var
       p,w     :Tarr;
         {由于这题的点边很有规律，一个点最多只可能和六个点有边}
         {边上的权也都是定值，所以，我们用 w 记录边上的权，p 记录}
         {相对的点的位置}
   data    :Tdata; {data 就相当于题目分析中的部分和序列 T}
   n       :word;

procedure init; {读入过程}
 var
  inf    :text;
    a0,b0,l0,a1,b1,l1,i:Integer;
 begin
  assign(inf,infns);
  reset(inf);
    readln(inf,n,a0,b0,l0,a1,b1,l1);
    close(inf);
    p[1]:=1; p[2]:=-1; p[3]:=l0; p[4]:=-l0; p[5]:=l1; p[6]:=-l1;
    w[1]:=1; w[2]:=0;  w[3]:=l0-a0; w[4]:=-(l0-b0); w[5]:=b1; w[6]:=-a1;
    fillchar(data,sizeof(data),$a);
    data[0]:=0;
  end;

  procedure main; {迭代法求最短路径主过程}
 var
      finish          :Boolean;
      i,j             :Word;
   begin
      repeat
          finish:=True;
          for i:=0 to n-1 do
              for j:=1 to 6 do
                  if (i+p[j]<=n) and (i+p[j]>0) and (data[i]+w[j]<data[i+p[j]])
                   and (data[i]+w[j]>=0)
                       then begin finish:=false; data[i+p[j]]:=data[i]+w[j];
                           end;
          until finish;
      end;
```

```
  procedure out;  {输出过程}
        var
            i      :Word;
        begin
            assign(output,outfns);
            rewrite(output);
            if data[n]=2570   {判断无解情况}
            then writeln(-1)
                        else
            begin
                write(0);  {输出时，要将 data 转变回 01 串序列}
                for i:=1 to n-1 do
                    write(data[i]-data[i-1]);
                writeln;
            end;
            close(output);
        end;

  begin
        init;
        main;
        out;
  end.
```

{求第 k 大的数}

```
program sort;
    type
        Tdata     =     array [1..10000] of Word;
 var
        data       :Tdata;
        n,k        :Word;

 procedure out(x:Word); {输出答案}
        begin
            writeln(data[x]);
            halt;
        end;

  procedure swap(var a,b:Word);  {交换 a 和 b}
        var
            tmp        :Word;
        begin
            tmp:=a;
```

```
        a:=b;
        b:=tmp;
    end;

procedure get_mid(head,tail:integer;var x:integer); {寻找分治点 x}
    var
        i,j    :integer;
        m          :Word;
    begin
        i:=head+random(tail-head+1);
        swap(data[head],data[i]);   {随机化}

        i:=head-1; j:=tail+1; m:=data[head];
        while true do
            begin
                repeat j:=j-1;
                until data[j]<=m;
                repeat i:=i+1;
                until data[i]>=m;
                if  i>=j
                    then
                        begin
                            x:=j;
                            exit;
                        end
                    else swap(data[i],data[j]);
            end;
    end;

procedure quicksort(head,tail:integer);  {排序主程序}
    var
        x          :integer;
    begin
        if head>tail then exit;
        get_mid(head,tail,x);
        if x>k then quicksort(head,x)
                    else if x<k then quicksort(x+1,tail)
                                                    else out(x);
    end;

procedure init; {读入过程}
    var
        i     :Word;
    begin
```

```
        assign(input,'input.txt');
        reset(input);
        readln(n,k);
        for i:=1 to n do
            readln(data[i]);
        close(input);
    end;

begin
    init;
    quicksort(1,n);
end.
```

{NOI'95 极值问题}

```
program max;
 var m,n,next,k              :longint;
 begin
    write('K= '); readln(k);     {读入数值 k}
    if (k<1) or (k>1000000000) then halt;
                        {如果 k 不在要求范围内，就终止}
    m:=1; n:=1; next:=m+n;       {确定斐波拉契数列的头三项}
    while next<=k do             {顺推，求出 m,n 的最大值}
      begin
        m:=n;
        n:=next;
        next:=m+n;
      end;
  writeln('M= ',m);    writeln('N= ',n);
 end.
```

【参考书目】

1．《数学思维能力的训练》 广东人民出版社 1992

2．《IOI'98 中国集训队优秀论文集》 IOI 中国集训队《信息学奥林匹克》编辑部 1999

3．沈继红 施久玉等编著 《数学建模》 哈尔滨工程大学出版社 1998

4．王树禾 编著 《数学模型基础》 中国科学技术大学出版社 1997

5．徐利治 编著 《数学抽象与数学抽象方法》 1990