国家自然科学基金成果专著出版基金(50024036)专著
国家自然科学基金(79400011、69974033、59979024)资助

大系统试验选优理论和应用

——在复杂水利系统优化规划中的应用研究

程吉林　著

上海科学技术出版社

内 容 提 要

本书从数学模型、知识模型和试验选优理论相结合的角度出发，系统地介绍了适用于某些多维动态规划、大线性规划、复杂非线性规划和某些定性、模拟系统的试验选优方法以及在某些复杂水利系统优化规划中的应用实例。本书的理论方法和应用实例，基本上都是作者多年来的研究成果。

本书共分 6 章，除了第 1、2 章外，其他各章均有实例介绍，可供读者在研究和应用时参考。本书适合管理、水利专业的研究生和科技工作者阅读和参考。

前　言

大系统优化理论研究是系统科学领域的重要学术课题。近十年来，作者在汲取钱学森、J. S. Dyer 等人的大系统研究思想的基础上，采用数学模型、知识模型和试验选优理论相结合的方法，对某些大系统问题的优化方法进行了探索，并在某些复杂水利系统优化规划的应用中取得了较好的效果。

本书共分 6 章。第 1 章为绪论，对有关现行大系统理论和方法进行了综合评述。第 2 章介绍了大系统试验选优理论的思路并对试验选优的优良性开展了讨论。第 3 章介绍多维动态规划的广义拉氏方法和试验选优方法。对传统拉格朗日法引入松弛变量，在对松弛变量递推、检验的广义拉氏方法基础上，对拉格朗日乘子的试验选优，使多维问题转化为一维问题，从而使较多维数的某些动态规划问题的求解成为可能。这一章还给出了跨流域调水工程渠道系统优化设计的多维动态规划实例模型，及其与常规多维问题求解方法 DDDP 法的计算比较结果。第 4 章至第 6 章分别介绍了某些大线性块角结构问题、复杂非线性系统、模拟仿真、复杂定性系统的试验选优方法。对于某些耦合约束较少的大线性块角结构问题，试验选优方法同单纯形法相比将大大节省计算机(CPU)运算时间；对于某些复杂非线性问题，试验选优方法和分解聚合法相比具有同样的适用性，并且计算机运算时间大为节省。从某种意义上讲，分解聚合法为本书介绍的试验选优方法的一个特例——全面试验选优法；对于某些模拟仿真系统，试验选优方法可较好地解决某些方案组合“爆炸”问题的最优化；对于复杂定性系统，试验选优方法由于可以在选优过程中淘汰和生成方案，从而使某些复杂定性问题的优化成为可能。本书还给出了喷灌线性管网模型、地下水和地面水联合调度的复杂非线性模型、灌区优化规划模拟和定性模型的实例以及与常规方法对应的优化比较结果。

本书是作者和作者所在课题组在连续三项国家自然科学基金资助下，近十年来的研究成果的总结。在本书即将出版之际，我首先要衷心感谢我的师长武汉大学郭元裕、沈佩君、陈学敏教授和扬州大学金兆森、刘超教授，他们的指导、关心和培养使我终身难忘。我还要对我的合作伙伴扬州大学沈洁、陈平、朱春龙、吉庆丰、刘正祥和鄢碧鹏等同志多年来的工作支持和帮助表示由衷的感谢。

我衷心地欢迎读者对本书提出宝贵意见。

程吉林

2001 年 12 月于扬州大学

目　　录

第1章　绪　　论

§1.1　大系统理论的研究目的和意义

由于现代社会日趋系统化、信息化，在工程技术、社会经济和生物、生态等各个领域中均出现了许多复杂的大系统。另外，随着经济建设和科学技术的发展，也越来越要求提高决策的科学性和正确性，使原本复杂的大系统增添了更大的决策难度和复杂性，因而完善和发展大系统理论，对促进现代社会国民经济各个部门的科学管理和决策具有极其重要的社会经济价值和现实意义。

大系统可定义为：人们为了解决社会生产、科学技术等方面的某些问题，而必须处理的一系列复杂事物所组成的体系。它的特点是：①规模庞大，大系统通常包含有众多的小系统、部件、元件，占有的空间大，涉及的范围广；②结构复杂，大系统中各小系统、部件、元件之间的相互联系及其信息结构等都十分复杂；③功能综合，大系统常具有综合性的、多方面的功能与目标；④因素众多，大系统一般是多变量、多参数、多输入和多输出的系统，涉及的内部因素和外部因素众多，不仅有“物”的客观因素，还有“人”的主观因素。

大系统优化理论是20世纪70年代为求解大系统优化决策而发展起来的一门新兴学科，它综合和发展了近代控制理论、数学规划和决策论等方面的成果，不仅把复杂的工程系统作为研究对象，而且已扩展到社会经济和生态环境等系统之中。

随着人类社会的发展以及工农业和科学技术的进步，需要研究的问题越来越多、越来越复杂，所研究的优化问题“维数”越来越高，并且自然科学和社会科学的内容交织在一起。为了解决“维数”障碍问题和难以定量、决策的困难，可以从两个方面来努力：一方面发展高速度、大容量的计算机，在硬件上取得突破；另一方面提高软件技术，寻求效率高、存储少的优化技术和方法。大系统优化理论研究的就是如何提高软件的效率。

目前，大系统优化理论及其应用已在国内外受到广泛的注意与重视，具体表现在以下几个方面：①在学术活动方面，大系统理论研究及应用，已成为国际学术交流中的热门课题。从1975年至今，国际上几乎每隔二三年都要召开一次大系统理论和应用的学术交流会；②在文献资料方面，大系统理论和应用已经是国内外运筹学、系统科学与系统工程、管理工程、自动控制、水资源研究、水电能源科学等许多学科有关刊物的重要论题；③大系统研究的内容和范围不断扩大，近年来，已推广应用于各个自然科学领域和许多社会科学领域以及经济建设的各个部门，取得了可观的成果。

我国著名的系统工程专家钱学森和涂序彦等[1-7]针对大系统的复杂性，提出了许多有待解决的难题，诸如：①大系统分析设计中如何考虑人（管理、操作人员）的主观能动性；②大系统由许多分散而又联系的子系统组成，需要发展更加完善的大系统优化技术以求适用于分散性很强的大系统；③由于大系统常有许多因素具有不确定性（随机性和模糊性等），因

而，基于确定性理论的大系统优化技术是难以适应的；④大系统往往缺乏齐全的资料，是信息不足或不确定的系统，所以要求信息完备的理论和算法遇到了困难；⑤大系统数学模型往往是多维高阶的，给系统分析带来了困难。1985 年钱学森提出“大系统理论要创新”[1]，1995 年他又呼吁“要开创大型复杂巨系统的研究”[2]，并强调“在解决大系统的系统工程问题时，要注意利用不能称之为‘科学’的人的知识和经验”；“在大系统控制论中，要发展广义模型化方法，而不应局限于数学模型化，在难于或不适宜建立数学模型的场合，可以建立知识模型，引用人工智能、模糊识别、知识工程方法等，跳过数学模型的障碍，直接由知识模型转化为计算机模型，这样，既采用知识模型，又采用数学模型，两者相结合，形成广义模型，便于处理大系统的模型化问题”。

目前，在这一思想的启发下，国内外众多学者对大系统优化理论进行了探讨[8－58]，并且已将它广泛应用至工程技术、社会经济和生物生态等领域的各个部门[59－100]。大系统优化理论在现代社会国民经济各个部门的科学管理和优化决策中起着越来越重要的作用。

§1.2 多维动态规划理论与方法的研究进展

对于多维动态规划问题，由于离散型动态规划用递推方程进行择优计算时，计算机高速存储（内存）中至少须存放相邻两个阶段的各状态点的优化结果，而一个阶段的离散状态点数为

$$T_n = T_i^m$$

式中，m 为问题的维数（即状态变量数）；T_i 为 m 维空间中一个坐标（即一个状态变量）在任一阶段 i 的离散状态点数，设各个坐标离散点数都是 T_i；T_n 为在任一阶段 n 的状态点数。当维数 m 稍大时，往往会超过现有计算机的存储能力致使无法求解，或因计算时间过长而费用昂贵。这就是动态规划在广泛应用中所遇到的主要障碍，即所谓“维数灾”问题。所以，自 20 世纪 50 年代初 Bellman 提出动态规划[101－104]以来，许多学者提出了各种改进方法[105－137]来求解多维问题。由于“维数灾”主要是由离散化引起的，现有的改进方法按照其解决的途径主要可以分为以下几类。

① **降低维数 m**　　如拉格朗日乘子法（Lagranges dynamic programming，LDP）[104－106]、动态规划逐次渐近法（dynamic programming succesive approximation，DPSA）[103，108－110]、聚合库偶法（a deconposition，AD）[111－114]等。

② **减少状态离散点数 T_i**　　如状态增量动态规划法（state increment dynamic programming，SIDP）[115]、离散微分动态规划法（discrete differential dyanmic programming，DDDP）[119]、双状态动态规划法（binary state dynamic programming，BSDP）[122]等。

③ **不进行离散化**　　如微分动态规划法（differential dyanmic programming，DDP）[123]、逐级优化计算法（progressive optimality algorithm，POA）[65]等。

以下对于一些最典型的方法进行评述。

1.2.1 降低维数方法概述

1. 拉格朗日乘子法（LDP）

（1）基本方法[104，105]

拉格朗日乘子法的基本思想是:通过引入拉格朗日乘子,将一个或若干约束方程并入目标函数,仅保留较少的约束方程,从而减少了状态变量数。

为了说明怎样利用拉格朗日乘子,考虑下述一个二维问题:

$$\max Z = \sum_{s=1}^{N} \Phi_s(x_s) \tag{1-1}$$

$$s.t. \quad \sum_{s=1}^{N} h_{1s}(x_s) \leqslant b_1 \tag{1-2}$$

$$x_s \geqslant 0,\ s = 1, 2, \cdots, N$$

$$\sum_{s=1}^{N} h_{2s}(x_s) \leqslant b_2 \tag{1-3}$$

这里假设 $\Phi_s(x_s)$ 为 x_s 的非减函数,且对约束之一如(1-2)式,存在 $\sum_{s=1}^{N} h_{1s}(x_s) \leqslant \sum_{s=1}^{N} h_{1s}(x_s^*)$,$x_s^*$ 为最优解分量。

由于假设 $\Phi_s(x_s)$ 是 x_s 的非减函数,x_s 为连续变量,以及对第 1 个约束所加的条件,所以对最优解,约束(1-2)式或(1-3)式中至少有一个必须成为严格等式。现假定达到最优解时约束(1-2)式成为等式,则原问题可表述为

$$\begin{aligned} &\max Z = \sum_{s=1}^{N} \Phi_s(x_s) - \mu \sum_{s=1}^{N} h_{1s}(x_s) \\ &s.t. \quad \sum_{s=1}^{N} h_{2s}(x_s) \leqslant b_2,\ x_s \geqslant 0,\ s = 1, 2, \cdots, N \end{aligned} \tag{1-4}$$

式中,μ 为拉格朗日乘子。对一个指定的 μ 值,仅用一个状态变量,就可求解(1-4)式所表达的问题。如果把 b_2 看成一种资源量,x_s 为决策变量,并设 s 为阶段变量,λ_s 为状态变量,则可按分配问题建立状态转移方程:

$$\lambda_{s-1} = \lambda_s - h_{2s}(x_s),\ s = 2, \cdots, N \tag{1-5}$$

于是可写出递推方程

$$\begin{cases} g_s(\lambda_s) = \max\limits_{0 \leqslant x_s \leqslant \zeta_s} \{[\Phi_s(x_s) - \mu h_{1s}(x_j)] + g_{s-1}[\lambda_s - h_{2s}(x_s)]\},\ s = 2, \cdots, N \\ g_1(\lambda_s) = \max\limits_{0 \leqslant x_1 \leqslant \zeta_1} \{[\Phi_1(x_1) - \mu h_{11}(x_1)]\},\ s = 1 \end{cases} \tag{1-6}$$

其中,ζ_s 为下式的根:

$$\lambda_s - h_{2s}(\zeta_s) = 0,\ s = 2, \cdots, N \tag{1-7}$$

因此对某些固定的 μ 值,通过(1-6)式和(1-7)式,就能得到(1-4)式的最优解

$$x^* = (x_1^*, x_2^*, \cdots, x_N^*)$$

可以证明,相对于某个指定的 μ 值,可求得(1-5)式的最优解 $x^*(\mu)$,使得

$$\sum_{s=1}^{N} h_{1s}(x_s^*) = b_1 \tag{1-8}$$

即(1-2)式取等式,则此特定解 x^* 是原问题(1-1)式—(1-3)式的最优解。

还可证明,当 μ 由 0 增到∞时, $\sum_{s=1}^{N} h_s(x_s^*(\mu))$ 是单调减小的[106]。

(2) 评述

① 采用拉格朗日乘子法求解多维问题是建立在 $\Phi_s(x_s)$为 x_s 的非减函数和加入至目标函数中的约束条件在最优解时等式成立这两个条件基础上的。因此,可能遗漏使 $\sum_{s=1}^{N} h_{1s}(x_s^*) < b_1$ 成立的最优解,并且当 $\Phi_s(x_s)$不是非减函数时该方法无法应用。

② 对于 m 维动态规划问题,采用拉氏方法,必须要给出 $m-1$ 个拉氏乘子的数值,并且只有当 $m-1$ 个加入目标函数的约束条件均等式成立(即拉氏乘子为最优值)时,才能获得对应的最优决策变量。因此,拉氏乘子虽然在理论上能求解多维问题,但当 m 稍大 ($m > 4 \sim 5$) 时,一般问题的求解也是非常困难的。

③ 由求解过程可知,拉氏方法是在给定拉氏乘子后把多维问题转化为一维问题求解,但一维问题求解完成后,必须对加入目标函数中的约束进行检验,反复调整拉氏乘子,直到约束满足条件,因此该方法属于非可行算法。

④ 拉氏方法获得的最优解为全局最优解。

由于拉氏方法存在上述限制,因此在解决多维实际问题时,运用拉氏方法求解优化并不多。

2. 动态规划逐次渐近法(DPSA)

(1) 基本方法

动态规划逐次渐近法由 Bellman 提出,简称为 DPSA[104],其基本思想是,把包含若干决策变量的问题变为仅仅包含一个决策变量的若干子问题,每个子问题比原来的问题具有较少的状态变量,因而可减少高速存储量。当状态变量数和决策变量数相等时,每个子问题只有一个状态变量。

其具体步骤如下:

首先,假设一个虚拟状态序列(称为虚拟轨迹) $\{\lambda_1^0, \lambda_2^0, \cdots, \lambda_N^0\}$ 相应的决策序列(即策略)为 $\{x_1^0, x_2^0, \cdots, x_m^0\}$。

从 m 个状态变量中挑选一个状态变量如 λ_1^0(λ_1^0 为第一个状态变量),假设所有其他状态序列 $\{\lambda_s^0\}$, $s \neq 1$ 固定不变,则这个问题只有一个状态变量 λ_1^0,决策向量仍有 m 个分量。但由于 $(m-1)$ 个状态必须保持固定不变,致使决策向量还要受到这些附加约束,所以起作用的决策变量也只有一个。这样就把原来的 m 维问题简化成一维问题。对这个一维问题采用动态规划法(DP)求解,得到一个新的决策序列 $\{x_1^1, x_2^1, \cdots, x_m^1\}$ 和新的状态序列 $\{\lambda_1^1, \lambda_2^1, \cdots, \lambda_N^1\}$。

然后,另选一状态变量如 λ_2^1,重复以上最优化过程,直到每个状态变量对于这个最优化过程至少被选了一次,并且当针对任一状态变量进行最优化时都得到相同的决策序列和状态序列时,便结束计算。

(2) 评述

DPSA 是把求 m 维动态规划问题转化为求一系列一维问题，从而大大节省了计算机的存储量和计算时间。它是目前比较有效的一种降维方法，在水资源优化规划中也有过不少报道。如 Larson(1968 年)，Yeh 和 Trott(1972 年)[108]，Gilest 和 Wunderlick(1981 年)[109]将这一方法先后应用于多水库水资源优化调度。其中，Yeh 和 Trott 利用 DPSA 法成功地解决了 6 个水库联合调度的六维动态规划问题，有效地避免了“维数灾”。

但 DPSA 法不能保证在所有情况下都收敛到真正的最优解。Bellman(1957，1962 年)[102]对 DPSA 收敛性进行了讨论，但优化结果是否收敛到全局最优解并未得到证明。在实际应用中，也有不少结果确实收敛到全局最优解。因而，提高得到全局最优解的可能性是改进 DPSA 法的方向，为了改善解的最优性，可以设法选取更为合适的起始廊道。Nopmongcol 和 Askew(1979 年)[110]对 DPSA 进行了研究，将原始问题分解为若干子问题，每个子问题变成“一次 2 或 3 个变量”与“一次 1 个变量”的组合。

3. 聚合库偶法(AD)[111，112]

(1) 基本方法

聚合库偶法的基本思想是把一个 m 维问题分解成 $m-1$ 个子问题，每个子问题是由相邻的二维约束构成。这样，就把一个 m 维问题转化为 $m-1$ 个二维问题。最后一个约束和第一个约束构成第二次循环迭代的二维状态，依次类推反复迭代至最优解。

(2) 评述

AD 是 1981 年 Turgeon[112]在串联水库优化调度中提出的减维方法，它有很强的针对性。目前除了在水库优化调度的应用外，其他领域尚不多见。

1.2.2 减少状态离散点数方法概述

减少状态离散点来减轻多维动态规划的“维数灾”，是目前求解动态规划实际问题中运用最多的方法，其中最典型的有状态增量动态规划(SIDP)法、离散微分动态规划法(DDDP)和双状态动态规划法(BSDP)。

1. 状态增量动态规划法(SIDP)和离散微分动态规划法(DDDP)

(1) 基本方法

Larson(1968 年)[115]首先提出了状态增量概念。该方法用子问题的序列最优化来收敛原问题的最优解。

基于状态增量的概念，Hall(1969 年)[116]和 Trott(1971 年)[117]将 SIDP 应用于水库优化调度，Heidari(1971 年)[118]对该法进行了总结和发展，提出了 DDDP。Nopmongcol 和 Askew(1976 年)[119]研究了这两种方法，认为它们实际上是同一种方法。

SIDP 的求解步骤为：①选初始状态序列和决策序列，可根据一般经验和分析计算定出一个尽可能接近最优的决策序列，并相应得到初始状态序列；②选增量形成廊道，在该初始状态序列的上下各变动一个小范围 Δ，形成一个带状的廊道；③利用递推方程在廊道内进行寻优，找到新的状态变量序列；④反复迭代直至收敛，得到最优解。

(2) 评述

为了使上述两种法有较好的收敛性，Hall(1969 年)[120]提出了两种状态增量的方法：使每次增量 Δ 很小，但保持常量；在循环迭代初选用较大的增量 Δ，若在迭代过程中发现目标值变化很小，则逐步减小增量 Δ 来提高最优解的精度。通常，变增量法在实际应用中相当

有效。

这两种法所得到的解不能证明是全局最优解,但有可能收敛到某一局部最优解。Turgeon(1982 年)[121] 证明,如果 SIDP 中每时段状态增量相同,可能会使该法的结果不为最优解。对此,Turgeon 提出对不同时段的增量值加以调整,以提高得到全局最优解的可能性。

2. 双状态动态规划法(BSDP)[122]

(1) 基本方法

双状态动态规划由 Ozden 于 1984 年提出。它求解多维动态规划问题的思路与 DDDP 基本相同,也是由一个初始试验轨迹(又称虚拟轨迹)开始,以这个试验轨迹为基准建立状态子集,将它作为本次迭代的状态域,并用常规动态规划(DP)寻求最优轨迹(又称改善轨迹);然后再以本次迭代的最优轨迹作为下次迭代的试验轨迹,重新建立状态子集进行优化,通过多次迭代逐次逼近最优解。

(2) 评述

BSDP 在任一阶段每个坐标仅取 2 个离散值,每次迭代只包含 2^m 个网点的状态子集,因而逐阶段递推计算所需要的高速存储需要量为 2^{m+1},一次迭代的运算时间只与 2^{2m} 有关。由于各次迭代的状态子集之间相互联系,使得每次迭代所得目标函数有可能得到改善;同时在状态空间移动子集以便最终获得最优轨迹。若用常规动态规划法求解,在 m 维状态空间中每个坐标被分成 T_i 个离散值,则一个阶段的状态网点总数为 T_i^m,其运算数据存储需要量至少为 $2T_i^m$,运算时间与 T_i^{2m} 有关。BSDP 把 T_i 减少到最小值,从而大大节省了存储量和计算时间。

当阶段函数性能良好而没有尖谷时,BSDP 能非常有效地寻求到最优轨迹。否则在步长减半之前,需沿着试验轨迹周围搜寻,重新定向。该算法对阶段效益函数性质无严格要求,不强求符合诸如二次型、连续性或可微性等特性,而在凸性假设下,能保证收敛到全局最优解。BSDP 与 DDDP 的区别主要在于:对同样条件下的 m 维问题,DDDP 在每个阶段所需要的计算量是 3^{2m},运算时间是 3^{2m} 的函数;而 BSDP 在每个阶段所花费的运算时间基本上是计算量 2^{2m} 的函数。这两种算法在每个阶段的运算时间之比为 $r_m = 2^{2m}/3^{2m} \approx (0.67)^{2m}$($r_2 \approx 0.20$, $r_3 \approx 0.09$, $r_4 \approx 0.04$, $r_5 \approx 0.02$)。虽然,BSDP 法因每次迭代所取状态点数少而使迭代次数较 DDDP 法多,但总的运算时间的节省仍是非常显著的。

1.2.3 不进行离散化方法概述

对多维动态规划问题不进行离散化、直接进行求解的方法,主要有微分动态规划法(DDP)和逐级优化计算法(POA)。

1. 微分动态规划法(DDP)

在求解动态规划问题中,有些问题可以充分利用方程的结构来避免“维数灾”。这类问题是具有线性动态特性和二次特性的判别函数。如果目标函数是可分的和凸的(对于最小化问题),并且系统可以全部用一些线性的动态等式加以描述,那么决策值与状态变量将成线性关系。因此,当给定一个系统初始状态后,就能得到一个解析解,在求解过程中不会遇到‘维数灾’问题。如果目标函数不是二次的,或者系统的动态特性不能用线性表达式描述,那么可以用泰勒级数将目标函数近似展开成二次,动态特性表达式也可在

状态变量附近近似展开，这样对于非拉格朗日乘子问题也可用上述方法求解。Jacobson（1970 年）[123] 将此方法称为 DDP，它克服了“维数灾”，同时取得二次收敛。Murray（1979 年）[64] 将 DDP 法修改后应用到多水库系统，修改后的方法称为 CDDP，Yokowitz（1982 年）[124] 对其作了收敛性证明。Gjelsvik（1982 年）[125] 提出了一阶 DDP，并将其用于 5 座水库系统的优化计算。

2. 逐级优化计算法（POA）

Howson（1975）[65] 沿用动态规划的思想，在两个阶段的初末状态固定的情况下，对中间状态进行优化计算，直到完成整个周期的计算，反复迭代，直到收敛。这一方法命名为 POA，Turgeon（1981 年）[126] 对此做了进一步的研究。

POA 将多阶段优化问题分解成若干子问题，子问题之间由系统状态联系，每个子问题仅考虑相邻时段的子目标值，这样大大减少了计算所占用的内存空间。但 POA 法仅适用于初始条件和终末条件一定的最优问题。

Howson 在提出 POA 时已证明了其收敛性，也证明了解为最优解，不过这种方法对目标函数和约束条件有特定的要求。实践证明，POA 适合于如水库调度确定性入流的优化计算，它从根本上消灭了“维数灾”。

POA 对目标函数有特殊要求（即为凸性目标函数），若不满足，计算很难收敛到最优点。张勇传等（1984 年）[127] 提出了 POA 算法的改进算法 SEPOA 法。

1.2.4 本节小结

动态规划在系统优化规划与管理决策中，具有很大的灵活性和适应性。在一定条件下，动态规划方法能够保证所求得的解为全局最优解。动态规划在实际应用中特别应注意的问题是“维数灾”，由此出现了一些改进的动态规划方法，如前所述的 LDP、DPSA、AD、SIDP、DDDP、BSDP、DDP 和 POA，这些方法都能在一定程度上有效地避免“维数灾”的发生，但它们都有自身的缺陷：LDP 要求阶段函数为非减函数，且对多维的拉氏最优乘子的确定很困难；AD、DDP 和 POA 对目标函数和约束条件均有特定的要求；而 DPSA、SIDP、DDDP、BSDP 虽在一定程度上克服了“维数灾”，但若要获得全局最优解，需要一个好的初始轨迹。由于实际问题的高度复杂性，对于数十、数百乃至上千维的多维动态规划问题的求解目前尚难开展。

§1.3 大线性问题的理论与方法概述

设 $\boldsymbol{A}$ 是 $m \times n$ 矩阵，$\boldsymbol{B}$ 是 m 维向量，$\boldsymbol{C}$ 是 n 维向量，我们要求满足约束 $\boldsymbol{AX} \leqslant \boldsymbol{B}$ 的 n 维向量 $\boldsymbol{X}$，使得 $\boldsymbol{C}'\boldsymbol{X}$ 达到最大值：

$$\begin{aligned} & \max \boldsymbol{C}'\boldsymbol{X} \\ s.t. \quad & \boldsymbol{AX} \leqslant \boldsymbol{B} \\ & \boldsymbol{X} \geqslant 0 \end{aligned} \tag{1-9}$$

这就是线性规划问题。它的建模和求解与生产计划、最优控制、对策论、组合学、离散变量的最优化、计算复杂性理论和许多离散应用数学问题的研究有密切的关系。

长期以来,人们使用单纯形法及其变形来求线性规划的解[128],Klee 和 Minty[129] 举例说明了用单纯形法存在一些问题,要用指数次算术运算才能求出解来。

1979 年 Khachiyan[130] 用椭球法来求解整系数的线性规划,证明了它是个多项式算法。由于它解决了重要的理论问题,引起了人们的关注。但实算表明椭球算法远比单纯形法差。1984 年 Karmarkar[131] 提出了一种阶次比椭球法低的多项式算法,并声称这种算法在解大型问题时比单纯形算法快得多。现在,对解线性规划方法及其复杂性的研究不单是个理论问题,在实用上也极为重要,它也引起了我国运筹学工作者的兴趣。

1.3.1 Karmarkar 多项式算法概述

线性规划(LP)在近十年中最重要的发展应是 Karmarkar(1984)的多项式 LP 算法。Hooker(1986)[132] 描述了这种进展的重要性,对这种方法的优点和弱点给出了一个直观的说明,同时为读者实现这种方法提供了足够的技术细节。Karmarkar 对 LP 提出了一个新的投影标度算法(projective scalling method)[133],并声称对于大型问题他的方法比单纯形方法快 50 倍。Karmarkar 算法的主要优点是在最坏情形具有多项式运算时间,这在理论上显然优于单纯形方法,因为单纯形方法在最坏情形只具有指数运算时间。然而,尽管单纯形方法的运算时间的理论界限是指数的,但它在实际应用中运用得相当好。对于大多数线性规划来说,单纯形方法在解决实际问题时的情形比它在理论上的最坏情形要远远好得多。鉴于这些事实,一个更好的假设是,由于投影标度算法的理论界限要好得多,也许它对于实际问题会解决得很好。这方面的想法导致了对 Khachiyan(1979)[134] 提出的第一个 LP 多项式算法的广泛研究,这位前苏联数学家的成就是一个理论突破,但它要求很高的精度,对自然产生的 LP 问题,这种方法一点也不比单纯形方法要好。

Karmarkar 方法是以两个基本原理为基础:①当现行的最好已知解在由约束定义的多胞形的中心附近时,沿最速下降(对于极小化而言)的方向移动;②在每次迭代时,解空间因被变换,使得现行的最好已知解以不变的方式向多胞形的中心移动。有人[133] 已提出将来投影标度算法也许还有其他用处,比如作为非线性规划的一个子程序,由于它能够极小化一个非线性函数,对于具有特殊结构的线性规划问题,诸如各种具有附加约束的网络流问题,它可能是极其有用的;它还可能导致一种优于 Dantzig-Wolfe 分解的分解技术[135]。Karmarkar 方法的出现已引起了许多领域的学者对数学规划的大量广泛研究。

文献[136]中讨论了 Karmarkar 算法的实现,认为不可能像他所说的那样好。然而,单纯形算法经过了几十年的改进并有成熟的程序可供使用,新提出的算法难免总会有不完善之处,但它至少是第一个有实用前途的、解线性规划的多项式算法。近年来出现了许多 Karmarkar 算法的变形和完善化算法[137-144],这些方法把线性规划和非线性规划中的一些技巧和方法(如对偶变量、罚函数、牛顿法等)融合进去,使算法更趋完善,有的算法还使迭代次数的阶减少为 $O(\sqrt{m+n}L)$(m 为约束个数,n 为变量个数,L 为数据长度)次。现在,Karmarkar 算法的有效性已被确认。

1.3.2 Dantzig-Wolfe 分解方法概述

1960 年,Dantzig 与 Wolfe 在前人的工作基础上,提出了求解大型线性规划问题的著名方法——Dantzig-Wolfe 分解方法[145]。它最适合处理具有块角结构的线性规划问题:

$$
\begin{aligned}
\min Z &= \boldsymbol{C}_1\boldsymbol{X}_1 + \boldsymbol{C}_2\boldsymbol{X}_2 + \cdots + \boldsymbol{C}_N\boldsymbol{X}_N \\
s.t. \quad & \boldsymbol{A}_1\boldsymbol{X}_1 + \boldsymbol{A}_2\boldsymbol{X}_2 + \cdots + \boldsymbol{A}_N\boldsymbol{X}_N = \boldsymbol{b}_0 \\
& \boldsymbol{B}_1\boldsymbol{X}_1 = \boldsymbol{b}_1 \\
& \vdots \\
& \boldsymbol{B}_N\boldsymbol{X}_N = \boldsymbol{b}_N \\
& \boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N \geqslant 0
\end{aligned}
\tag{1-10}
$$

这个方法的基础是凸组合原理和列生成方法,它可分为二级来处理问题:第一级是子问题,第二级是主规划。Dantzig-Wolfe 分解原理是大系统优化的开拓性工作,至今仍有很强的生命力和广泛的应用,许多研究者以其基本概念为出发点,把它扩展应用于其他一些具有特殊结构的数学规划问题[146]。Dantzig-Wolfe 二级协调是通过耦合约束对应的拉氏乘子 μ 来转换的。对应于(1-10)式的分解协调结构见图 1-1。

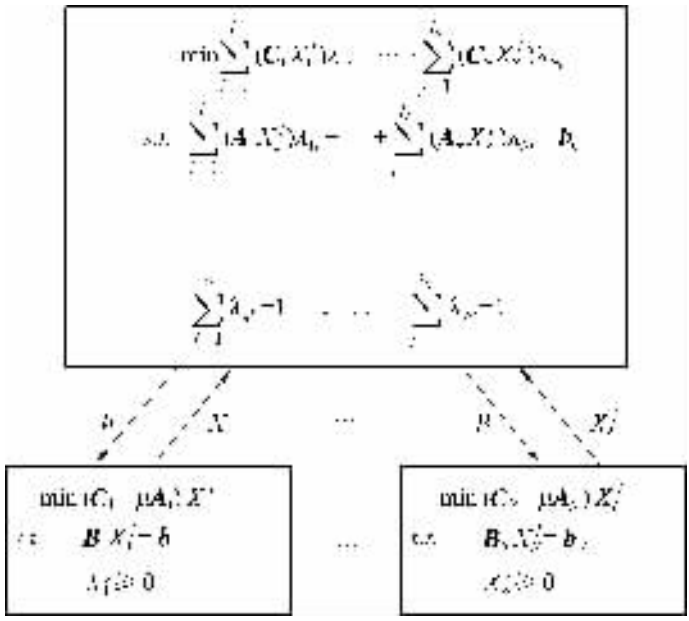

图 1-1　Dantzig-Wolfe 分解协调结构图

图中 λ_{ij} 为未知标量,根据凸多面体的性质[147],凸多面体的任何一点为凸多面体顶点上的线性组合(称凸组合),有

$$
\begin{aligned}
\boldsymbol{X}_i &= \sum_{j=1}^{l_i} \lambda_{ij} X_i^j \\
& \qquad\qquad \lambda_{ij} \geqslant 0 \\
\sum_{j=1}^{l_i} \lambda_{ij} &= 1
\end{aligned}
\tag{1-11}
$$

式中,l_i 为第 i 个凸多面体所具有的顶点数目(顶点数目一般大于空间的维数)。

一个大线性规划之所以难解,主要因为存在联系的耦合约束,从具体问题看,就是总公司下属各子公司之间有联系。所以,不能直接将它作为彼此独立的许多子问题来解。Dan-

tzig-Wolfe 分解方法把原规划分解为一个主规划和许多彼此独立的子规划,子规划按主规划给定的 μ 进行规划,求出子系统的决策变量可行解,把这些可行解汇报给主规划并用总目标来衡量是否最优,若不是最优,就重新给定 μ 值,进行反复计算。

尽管 Dantzig-Wolfe 分解原理反映的是一种数学技巧,它却揭示了一种合乎客观规律的工作方法。为了寻求最优,上级要倾听下级的意见,并根据下级的情况作出决策;下级则要坚决执行上级的指示和决策。

Dantzig-Wolfe 分解原理的意义非常重大,其思想不仅适用于广大线性规划,更重要的是也适用于大型非线性规划,目前它已作为大型非线性规划的经典方法之一[147]。

1.3.3 其他大线性方法简述

Dantzig-Wolfe 分解方法在线性规划中极具重要性,Birge(1985)提出了解块角系统的基因子分解技术的一种变形[135]。他发现 Dantzig-Wolfe 分解和基因子分解从同一路径达到最优解,并对混合这两种技术提出了一种新的分解技术:保持一个受限制的主问题以接受从子问题生成的列,不保持主问题的所有极点组的非负性(像 Dantzig-Wolfe 分解方法一样),包括一组用来在主问题的迭代过程中检验的附加约束。实质上,基因子分解方法是用来处理 Dantzig-Wolfe 分解结构的主问题数据的。

对于若干特殊的线性规划问题,Wittrock(1980)[148] 发表了关于阶梯形 LP 问题的对偶嵌套分解的文章。阶梯形 LP 问题的特点是:变量可以分在一组时间间隔上,而每个时间间隔上的变量只与相邻时间间隔上的变量有关。Wittrock 描述了一个解这种阶梯形 LP 问题的称作嵌套分解算法的特殊方法。对线性规划的对偶,该方法循环地应用 Dantzig-Wolfe 分解原理。它对每一时间间隔求解一个小的线性规划,每一间隔由决定它前面的间隔的约束来联系它后面的间隔,由增加约束来联系它前面的间隔。许多自然建立起来的问题如考虑资本预算和工厂计划中,包括了一系列必须在时间过程中按顺序作出的决策,因此阶梯形 LP 具有巨大的实际效用。还有另一个特殊的线性规划问题——Palekar(1987)[149,150] 等人为固定费用运输问题设计了一个新的有条件的惩罚,计算实例表明,这些惩罚的应用在处理困难问题时,在计数和求解时间上有可观的减少。Erenguc(1986)[151] 提出了一个解一类具有特殊结构的 LP 问题的方法。这类问题是从以下几个问题中产生的:①"公文包"选择,②广告布置的时间选择,③多阶段生产存货计划问题,④多阶段背包问题的 LP 松弛问题。他的途径包括解一系列连续的背包问题,每一个问题需要线性时间来求解,这个过程所需计算复杂性与约束矩阵的非零元素的数量成正比。

分派问题是一个具有非常特殊结构的线性规划问题,Bertsekas 对古典的分派问题提出了一个大规模的并行算法[152-155]。算法的执行过程好像进行拍卖,靠未被分派的人们同时投标争买物品来提高价格,一旦所有人投标以后,物品卖给最高的投标者。这个算法也可以解释为一个求解对偶问题的某类松弛方法。对于特殊的运算,它的最坏情形的复杂性则是 $O(MA\log(NC))$,这里 M 是节点数,A 是弧数,C 是最大绝对目标值。计算结果表明,对于大型问题,这个算法甚至在不利用并行性优势的情况下,可以与现存的方法媲美。当在并行机上执行时,算法可以很高阶地加速。

我国学者在大线性问题的研究方面也取得了一定的进展。1987 年,叶秉如发表了求解线性规划的一个新算法,他称之为最小减优率法[156],其原理与 1980 年所提出方法[157] 相

同。它是一种‘以线性规划问题约束凸集的基本特性和耦合约束对解的影响的基本法则为出发点的另一计算分解问题’的方法,先不考虑耦合约束,形成子问题,利用最小减优率的概念,从子问题的解推求有耦合约束的线性规划最优解。1986 年,何炳生[158]发表了求解线性规划的鞍点法,其原理是构造罚函数把线性问题转化为非线性问题来求解。

1.3.4 本节小结

本节主要介绍了大线性规划的各种算法。线性规划的实用算法非常重要,该问题成为各国系统工程领域的研究热点之一。近二十年来最重要的研究成果是 Karmarkar 算法,但其实用程序没有公布[159]。重要算法 Dantzig-Wolfe 提出的大线性问题分解方法在大型非线性领域取得了很大的成功,但在求解大线性问题方面并不十分理想,其他各种方法不是距实用尚有一段距离,就是算法本身有一定的限制,因此大线性问题有待于进一步研究。

§1.4 大型非线性问题的建模理论与方法概述

对于大型非线性问题,由于其复杂性,目前的建模理论和方法主要采用递阶、混合和广义模型。

1.4.1 大型非线性系统的递阶模型概述

在 Dantzig 与 Wolfe 的工作基础上,1970 年 Mesarovic 在研究互联系统的优化时,首次对大系统非线性规划分解协调技术的理论基础——拉格朗日乘子理论的基本概念作了精确的描述,并明确地提出了大系统优化的分解协调算法[160]。此后 Haimes 等人又进一步完善了这一方法[161]。就国内而言,1980 年叶秉如提出了求解线性或非线性规划的二维分解算法[157],1985 年胡振鹏提出了递阶分解聚合模型算法[162],1989 年叶秉如、董增川又在二维分解算法的基础上提出了线性规划的降维算法[14]。现在,大系统递阶模型及其优化方法已得到了长足发展和广泛应用。

1. 大系统分解-协调递阶模型

大系统分解-协调递阶模型是目前最常用的大系统优化模型,建立这一模型的基本思路是:首先,将一个复杂的大系统按照具体情况和需要划分为若干规模较小、结构比较简单的子系统;其次,采用一般的优化方法对各子系统分别择优,实现各子系统的局部最优化;然后,再根据整个大系统的总任务和总目标,修改调整各子系统的输入和决策,使各子系统‘相互协调配合’,实现大系统的全局最优化。

这种优化方法实际上是一种降维技术,即把一个具有多维问题的大系统分解为比较简单、维数较少的子系统。这种方法也是一种迭代技术,即各个子系统通过择优得到的结果还要进行反复协调修改,直到满足整个系统全局最优为止。协调通过迭代进行,以迭代换取降维,但这决不意味着增加计算时间,恰恰相反,它能大大减少计算时间。

一个大系统一般都可以分为若干层子系统,形成多层结构,在下一层进行分解,在上一层进行协调。分解的特征是暂时割裂各子系统之间的联系,让各子系统分别择优;协调的特征是恢复各子系统之间的联系和制约,以求得整个系统的最优。分解和协调在上下层之间

交替进行,这种交替递推可以扩大到更多的层次。

这种优化模型及方法既照顾了各个子系统的局部最优,也满足了整个系统全局最优的要求。实际上,没有各子系统的局部优化,决不会得到整个系统的全局最优。所以,为了得到全局最优,就得对各子系统进行协调修正,最小限度地牺牲某些子系统的局部利益。这种梯阶模型是各种大系统(包括工程和非工程的)普遍采用的结构模式,在日常生活中经常见到,只是过去没有给以定量描述而已。

通过二十多年来的研究与应用,关于非线性大系统的分解-协调国内外已提出了许多算法[163-169],它们大致可以归结为目标协调法、模型协调法和混合协调法三种基本类型。下面将以一个两层递阶系统为例,论述这种优化方法。假设所研究的两层递阶系统如图 1-2 所示,可以划分成 N 个子系统。每一个子系统的数学模型可以以其中任一子系统 i(见图1-3)为例予以描述。

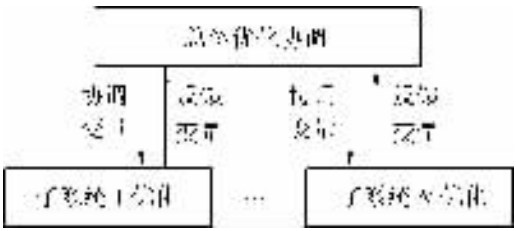

图 1-2　两层递阶结构

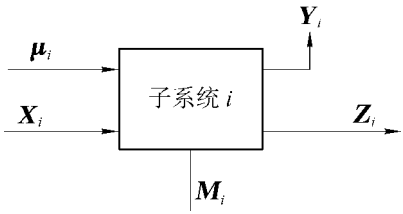

图 1-3　第 i 个子系统

设 $\boldsymbol{\mu}_i$ 既是对总系统也是对第 i 个子系统的输入(即来自总系统之外的扰动或其他无法影响的固定输入),维数为 $m_{\mu i}$;$\boldsymbol{X}_i$ 为由其他子系统提供的中间输入,维数为 m_{xi} ;$\boldsymbol{M}_i$ 为对第 i 个子系统的控制变量,维数为 m_{mi} ;$\boldsymbol{Y}_i$ 既是子系统又是总系统的输出,维数为 m_{yi} ;$\boldsymbol{Z}_i$ 只是子系统 i 的输出,它对其他子系统起输入的作用,维数为 m_{zi}。

子系统的描述方程为

$$\begin{cases} \boldsymbol{Z}_i = T_i(\boldsymbol{M}_i, \boldsymbol{X}_i) \\ \boldsymbol{Y}_i = S_i(\boldsymbol{M}_i, \boldsymbol{X}_i) \end{cases} \tag{1-12}$$

子系统之间的关联为

$$\boldsymbol{X}_i = \sum_{j=1}^{N} \boldsymbol{C}_{ij} \boldsymbol{Z}_j \tag{1-13}$$

式中,$\boldsymbol{C}_{ij}$ 为 $m_{xi} \times m_{zj}$ 的互联矩阵。

系统的目标函数为

$$\max F = \sum_{i=1}^{N} f_i(\boldsymbol{M}_i, \boldsymbol{X}_i) \tag{1-14}$$

系统的拉氏函数为

$$L = \sum_{i=1}^{N} f_i(\boldsymbol{M}_i, \boldsymbol{X}_i) + \sum_{i=1}^{N} \boldsymbol{\mu}_i'(T_i - \boldsymbol{Z}_i) + \sum_{i=1}^{N} \boldsymbol{\rho}_i'(\boldsymbol{X}_i - \sum_{j=1}^{N} \boldsymbol{C}_{ij} \boldsymbol{Z}_j) \tag{1-15}$$

式中,$\boldsymbol{\mu}_i$, $\boldsymbol{\rho}_i$ 分别为拉氏乘子向量,若假定这些等式约束是独立的,函数 f_i 和 T_i 是连续和一

阶连续可微的,则最优解应满足以下必要条件:

$$\frac{\partial L}{\partial \boldsymbol{X}_i}=\frac{\partial f_i}{\partial \boldsymbol{X}_i}+\left(\frac{\partial T_i}{\partial \boldsymbol{X}_i}\right)'\boldsymbol{\mu}_i+\boldsymbol{\rho}_i=0$$

$$\frac{\partial L}{\partial \boldsymbol{M}_i}=\frac{\partial f_i}{\partial \boldsymbol{M}_i}+\left(\frac{\partial T_i}{\partial \boldsymbol{M}_i}\right)'\boldsymbol{\mu}_i=0$$

$$\frac{\partial L}{\partial \boldsymbol{Z}_i}=-\boldsymbol{\mu}_i-\sum_{j=1}^{N}\boldsymbol{C}'_{ij}\boldsymbol{\rho}_j=0 \tag{1-16}$$

$$\frac{\partial L}{\partial \boldsymbol{\mu}_i}=T_i-\boldsymbol{Z}_i=0$$

$$\frac{\partial L}{\partial \boldsymbol{\rho}_i}=\boldsymbol{X}_i-\sum_{j=1}^{N}\boldsymbol{C}_{ij}\boldsymbol{Z}_j=0$$

这样形成的两级递阶结构的分解-协调,实际上就是在上下级之间不断交换信息的一种迭代优化过程,一般先由上级对下级发出指令(即给定协调变量值),然后各子系统经各自优化决策后向上级汇报(即送回反馈变量),从优化总系统的角度出发再向下级发出指令(即修改协调变量),如此反复迭代,最后达到整个系统的最优。在上下级(层)迭代过程中,由于所采用的协调变量不同,便得到了大系统分解-协调的三种基本算法,即目标协调法、模型协调法和混合协调法。

三种算法的分解-协调结构见图 1-4—图 1-6。

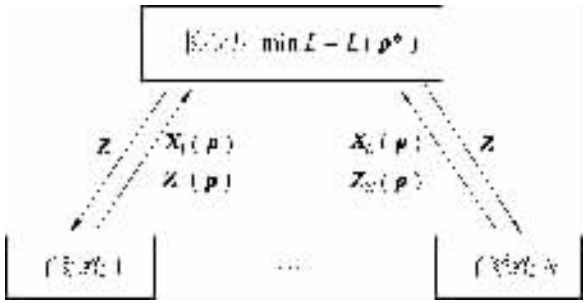

图 1-4　目标协调法递阶结构

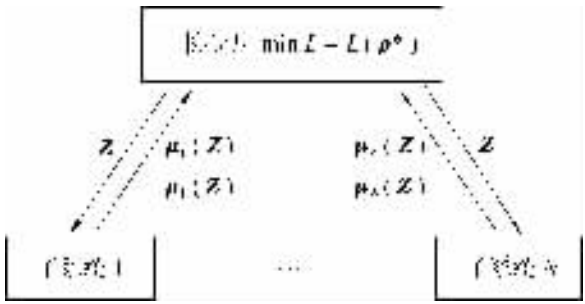

图 1-5　模型协调法递阶结构