国家出版基金项目
NATIONAL PUBLICATION FOUNDATION

“十三五”国家重点图书出版规划项目

排序与调度丛书（二期）

排序博弈

樊保强 万 龙 编著

清華大学出版社
北 京

内 容 简 介

本书全面系统地介绍了排序博弈的理论、模型和算法。全书共分为7章,第1章简要介绍排序问题的描述和表示,以及算法和计算复杂性;第2章简要介绍本书涉及的合作博弈理论;第3~7章介绍联盟排序博弈、两台机器的讨价还价问题、两代理排序的公平定价问题、时间表长和平行加工机制下的均衡分析。

本书可以作为运筹学、管理科学、应用数学和金融数学等专业的研究生教材,也可供相关领域的科研工作者参考。

图书在版编目(CIP)数据

排序博弈/樊保强,万龙编著.—北京:清华大学出版社,2024.7
(排序与调度丛书.二期)
ISBN 978-7-302-65193-2

Ⅰ.①排… Ⅱ.①樊… ②万… Ⅲ.①排序—研究 Ⅳ.①O223

中国国家版本馆CIP数据核字(2024)第034017号

责任编辑:佟丽霞 赵从棉
封面设计:常雪影
责任校对:王淑云
责任印制:宋 林

出版发行:清华大学出版社
网 址:https://www.tup.com.cn,https://www.wqxuetang.com
地 址:北京清华大学学研大厦A座 **邮 编**:100084
社 总 机:010-83470000 **邮 购**:010-62786544
投稿与读者服务:010-62776969,c-service@tup.tsinghua.edu.cn
质量反馈:010-62772015,zhiliang@tup.tsinghua.edu.cn
印 装 者:三河市龙大印装有限公司
经 销:全国新华书店
开 本:170mm×240mm **印张**:9.25 **字 数**:165千字
版 次:2024年7月第1版 **印 次**:2024年7月第1次印刷
定 价:79.00元

产品编号:098292-01

《排序与调度丛书》编辑委员会

丛书序言

我知道排序问题是从 20 世纪 50 年代出版的一本名为 *Operations Research*(《运筹学》,可能是 1957 年出版)的书开始的。书中讲到了 S. M. 约翰逊(S. M. Johnson)的同顺序两台机器的排序问题并给出了解法。约翰逊的这一结果给我留下了深刻的印象。第一,这个问题是从实际生活中来的。第二,这个问题有一定的难度,约翰逊给出了完整的解答。第三,这个问题显然包含着许多可能的推广,因此蕴含了广阔的前景。在 1960 年前后,我在《英国运筹学》(*Operational Research*,季刊,从 1978 年(第 29 卷)起改称 *Journal of the Operational Research Society*,并改为月刊)(当时这是一份带有科普性质的刊物)上看到一篇文章,内容谈到三台机器的排序问题,但只涉及四个工件如何排序。这篇文章虽然很简单,但我也从中受到一些启发。我写了一篇讲稿,在中国科学院数学研究所里做了一次通俗报告。之后我就到安徽参加"四清"工作,不意所里将这份报告打印出来并寄了几份给我,我寄了一份给华罗庚教授,他对这方面的研究给予很大的支持。这是 20 世纪 60 年代前期的事,接下来便开始了"文化大革命",倏忽十年。20 世纪 70 年代初我从"五七"干校回京,发现国外学者在排序问题方面已做了不少工作,并曾在 1966 年开了一次国际排序问题会议,出版了一本论文集 *Theory of Scheduling*(《排序理论》)。我与韩继业教授做了一些工作,也算得上是排序问题在我国的一个开始。想不到在秦裕瑗、林诒勋、唐国春以及许多教授的努力下,跟随着国际的潮流,排序问题的理论和应用在我国得到了如此蓬勃的发展,真是可喜可贺!

众所周知,在计算机如此普及的今天,一门数学分支的发展必须与生产实际相结合,才称得上走上了健康的道路。一种复杂的工具从设计到生产,一项巨大复杂的工程从开始施工到完工后的处理,无不牵涉排序问题。因此,我认为排序理论的发展是没有止境的。我很少看小说,但近来我对一本名叫《约翰·克里斯托夫》的作品很感兴趣。这是罗曼·罗兰写的一本名著,实际上它是以贝多芬为背景的一本传记体小说。这里面提到贝多芬的祖父和父亲都是宫廷乐队指挥,当贝多芬的父亲发现他在音乐方面是个天才的时候,便想将他培养成一名优秀的钢琴师,让他到各地去表演,可以名利双收,所以强迫他勤

学苦练。但贝多芬非常反感,他认为这样的作品显示不出人的气质。由于贝多芬有如此的感受,他才能谱出如《英雄交响曲》《第九交响曲》等深具人性的伟大乐章。我想数学也是一样,只有在人类生产中体现它的威力的时候,才能显示出数学这门学科的光辉,也才能显示出作为一名数学家的骄傲。

任何一门学科,尤其是一门与生产实际有密切联系的学科,在其发展初期那些引发它成长的问题必然是相互分离的,甚至是互不相干的。但只要研究继续向前发展,一些问题便会综合趋于统一,处理问题的方法也会与日俱增、深入细致,可谓根深叶茂,蔚然成林。我们这套丛书已有数册正在撰写之中,主题纷呈,蔚为壮观。相信在不久以后会有不少新的著作出现,使我们的学科呈现一片欣欣向荣、繁花似锦的局面,则是鄙人所厚望于诸君者矣。

越民义
中国科学院数学与系统科学研究院
2019 年 4 月

前　言

排序博弈是从优化的角度分析研究排序问题中存在的博弈现象，也是从博弈的观点研究排序问题。不同于经典排序所考虑的单决策者问题，排序博弈研究的是排序问题中工件或者机器属于不同决策者的博弈问题，从而所研究的内容更加贴合实际生产活动所出现的多决策者问题。例如，在某银行服务大厅内，多个客户在柜台前站成一排等待服务，每个客户有一个任务，已知每个客户的任务的服务时间，并且每个客户的费用函数是客户服务完毕时间的不减函数。多个决策者通过合作，即联合行动共同决定工件的加工顺序，能够产生节省费用。如何产生最大的节省费用，以及如何在参与合作的人中分配这些节省的费用，是这个多决策者问题需要解决的。这样的多决策者问题可以建模为一个可转移效用的排序博弈问题。

本书主要介绍各种典型的排序博弈问题模型及其理论方法。全书共分为 7 章：第 1 章简要介绍排序问题的描述和表示，以及算法和计算复杂性，包括对工件加工数据和特征、机器加工环境、加工性能指标函数的描述。第 2 章简要介绍了本书涉及的博弈理论中的成熟模型及其基本概念和基本理论，包括具有转移效用的联盟博弈、纳什讨价还价问题、公平定价问题和算法博弈论中的若干概念。第 3 章介绍联盟排序博弈，在该模型中工件属于不同的参与人，并且工件已经有一个初始的排序。参与人可以通过合作重新安排工件的加工顺序，从而产生节省费用，该模型中要解决的问题是如何在参与合作的人中分配这些节省的费用。第 4 章介绍两台机器的讨价还价问题，在该模型中机器属于不同的参与人。参与人通过协商确定工件的一个划分，使得相应的合作收益分配方案能够使所有参与人认可和接受。该章主要介绍两台机器的讨价还价问题的纳什讨价还价解。第 5 章介绍两代理排序的公平定价问题，在该模型中工件属于两个不同的参与人。基于不同的公平解概念，研究公平解存在的条件，并分析其结构性质和性能。第 6 章和第 7 章分别介绍时间表长（Makespan）和并行加工（Parallel Processing）机制下的均衡分析，此类模型研究的主要问题是纳什均衡和强均衡的存在性，并对其进行性能分析。在此类模型中工件属于不同的参与人（局中人），参与人的策略空间就是机器集，当确定了机器加工机制和费用

准则后，每个参与人在任意局势下的费用就能够被计算出来。每个参与人只是希望能减少自己的工件加工费用而不会去顾及总体目标。

本书的写作得到了唐国春先生的大力帮助和鼓励，并得到了“排序与调度丛书”编辑委员会各位老师的鼎力支持；同时也收到了排序与调度同仁和各位审稿专家的宝贵建议；清华大学出版社的编辑在本书的成稿过程中多次给予指导和建议；在本书撰写期间，作者的研究生在资料和文献整理方面给予了大力协助。在此，作者一并对上述各位表示衷心的感谢！

本书的出版得到了国家出版基金的资助。本书所涉及的部分研究成果也得到了作者主持的国家自然科学基金项目(项目号：12261039)的支持。

由于作者水平有限，本书难免存在疏漏甚至不妥之处，许多内容也有待深入研究和完善，敬请读者批评指正。

作　者

2024 年 1 月

目　录

第1章　排序论简介

排序(scheduling)论又名时间表理论,是一门源于制造和服务业,后又被广泛应用于管理科学、计算机科学和工程技术等众多领域的应用科学。排序论的研究既不同于传统的数学研究,即证明某一命题的是或非,也不同于解决某一实际问题的工作,而是从众多实际问题中提炼出某些带有普遍性的问题,然后对问题进行分析,研究它的可解性以及提出相关的算法。自创立以来,其丰硕的研究成果推动了排序论向更广阔的学科领域交叉融合,使排序论成为国内外发展迅速、研究活跃、前景诱人的学科领域之一。本章主要介绍排序问题的描述、排序问题的表示以及算法和计算复杂性的基本概念。

1.1　排序问题

排序问题是指在一定的约束条件下对工件和机器按时间进行分配和安排加工次序,使某一个或一些目标达到最优。这里将需要完成的工作、任务、被服务的对象等称为"工件",将完成工作、任务、服务所需要的资源称为"机器"。

排序在自动化学科中又称为"调度"。然而,用"排序"或"调度"来作为scheduling的中文译名都只是描述scheduling的一个侧面。scheduling既有"分配"(allocation)的作用,即把工件分配给机器以便进行加工;又有"排序"(sequencing)的功能,包括工件的次序和机器的次序这两类次序的安排;还有"调度"的效果,指把机器和工件按时间进行调度(唐国春 等,2003)。

1.1.1　排序问题的描述

用$J=\{J_1,J_2,\cdots,J_n\}$表示n个工件的集合,$M=\{M_1,M_2,\cdots,M_m\}$表示m台机器的集合,其中J_j表示第j个工件,M_i表示第i台机器,并且记p_{ij}为工件J_j在机器M_i上加工所需的时间。排序问题通常可以描述为n个工件在m台机器上进行加工,即分配工件给机器,同时安排其加工顺序。在任意时刻,每台机器只能加工一个工件,且每个工件只能在一台机器上加工。

令σ表示n个工件在m台机器上的一个排序。若同一机器在任意时刻最多只能加工一个工件,同一工件在任意时刻只能被一台机器加工,则称σ为一

个可行排序。在可行排序 σ 中，对于每一个工件 J_j，记 t 时刻完工的 J_j 的加工费用函数为 $f_j(t)$。排序的目标是使某个 $f_j(t)$ 的指标函数达到最优。

1.1.2 排序问题的表示

排序问题种类繁多，用简单明了的记号将一个排序问题表示出来，有利于在排序理论研究过程中规范表达这个排序问题以及理解该问题，有利于从事排序理论研究的专家学者之间的交流，有利于排序理论的推广、传播与应用。一般地，一个排序问题涉及工件、机器与排序指标，因此，目前国际上通用 Graham 等(1979)提出的三参数 $\alpha \mid \beta \mid \gamma$ 表示法，其中参数 α 表示“机器环境”，参数 β 表示“工件特征”，参数 γ 表示“优化目标”。

1. 机器环境的描述

关于机器环境，首先有单台机器排序问题与多台机器排序问题的区分。用 1 表示单台机器排序问题。对于多台机器排序问题，机器可分为通用平行机与专用串联机两大类。对于平行机问题，P 表示同型机问题，Q 表示同类机问题，R 表示非同类机问题；对于串联机问题，O 表示自由作业问题，F 表示流水作业问题，J 表示异序作业问题。对于多台机器排序问题，用 m 表示机器的台数。例如，$F2$ 表示 2 台机器的流水作业问题，$P2$ 与 Pm 分别表示 2 台机器的同型机问题与 m 台机器的同型机问题。如果不出现机器台数 m，则表示该多台机器排序问题所得到的研究成果或算法对任意台数机器都适用。例如，Q 就表示任意台数机器的同类机问题。

在通用平行机环境下，工件只需在其中一台机器上就可完成加工。其中：同型机是指所有机器都具有相同的加工速度；同类机具有不同的加工速度但此速度不依赖于工件；非同类机则对不同的工件具有不同的加工速度。而在专用串联机环境下，工件需要在每台机器上都进行加工。其中：在流水作业环境下，每一个工件以相同的机器次序在这些机器上进行加工；在自由作业环境下，工件依次在机器上加工的次序并不指定，可以任意；而在异序作业环境下，每一个工件以各自特定的机器次序进行加工。

2. 工件特征的描述

工件的基本特征有加工时间 p_{ij}、就绪时间 r_j、交货期 d_j、权重 w_j。由于排序论的应用领域越来越广泛，所以工件特征的描述也日益丰富。因此，工件特征还包括工件加工是否允许中断(pmtn)、工件是否具有先后约束关系(prec，tree，intree，outtree，chain)、工件加工时间可控排序(cpt)、工件可拒绝排序(rej)、成组分批排序(GT)、同时加工排序(s-batch，p-batch)、准时排序和窗时排序(jit)、资源受限排序(res)等。

p_{ij} 是指工件 J_j 在机器 M_i 上加工所需的时间，如对同型机有 $p_{ij}=p_j$。就绪时间 r_j 是指工件 J_j 可以开始加工的时间，工件就绪时间“缺省状态”是指所有的工件都同时就绪，或者认为 $r_j=0$。先后约束 $J_j \to J_k$ 表示工件 J_j 加工完后才能开始加工工件 J_k。交货期 d_j 是指工件 J_j 的按时交货时间，工件交货期“缺省状态”是指所有的工件都具有无穷大的交货期，即 $d_j=\infty$。中断是指一个工件在加工过程中，允许被别的工件抢先而中断加工，并稍后在原来的机器或在其他机器上继续加工。

3. 排序指标的描述

给定一个可行的排序 σ，令 $C_j(\sigma)$ 表示工件 J_j 的完工时间，$j=1,2,\cdots,n$，在不引起歧义的情况下，可以用 C_j 表示 $C_j(\sigma)$。排序指标一般分为两大类，一类为极小化最大费用问题；另一类为极小化总费用问题。最大费用问题包括：

(1) $C_{\max}=\max\{C_j \mid j=1,2,\cdots,n\}$——最大完工时间问题；

(2) $L_{\max}=\max\{L_j=C_j-d_j \mid j=1,2,\cdots,n\}$——最大延迟问题；

(3) $T_{\max}=\max\{T_j=\max\{0,L_j\} \mid j=1,2,\cdots,n\}$——最大延误问题。

总费用问题包括：

(1) $\sum w_jC_j$——工件总完工时间问题；

(2) $\sum w_jT_j$——工件总延误时间问题；

(3) $\sum w_jU_j$——误工工件数问题。

除上述问题之外，还存在少数考虑极大化指标的排序问题，例如第 7 章的极大化最小完工时间 $C_{\min}$。

在现代排序模型中，排序指标会增加一些其他费用。例如：工件加工时间可控问题，当工件加工时间压缩，则产生工件加工时间的压缩费用；工件可拒绝排序问题，当一个工件被拒绝加工，则产生一个惩罚费用；工件可外包排序问题，当一个工件外包，则产生一个外包费用等。

定义 1.1　对于一个排序问题，若对于任意两个可行排序 σ 和 σ'，其工件在对应可行排序中的完工时间满足 $C_j \leqslant C'_j$，$j=1,2,\cdots,n$，都有 $f(C_1,C_2,\cdots,C_n) \leqslant f(C'_1,C'_2,\cdots,C'_n)$，称排序指标 f 是正则的。

排序作为一个最优化问题，其目标通常是极小化排序指标函数。正则函数就是关于各个工件的完工时间的非降函数。容易验证上面给出的排序指标函数都属于正则函数。

三参数表示法一般情况下可以比较清晰地将一个排序问题表达清楚，但并不能将我们所讨论的问题都表示出来，有时需要给出一个必要的阐述与解释，同时新的排序问题会不断被提出来，参数会更加多种多样。

例 1.1 $1 \mid r_j \mid \sum w_j C_j$，表示单机排序问题，工件具有不同的就绪时间，排序指标为最小化加权总完工时间和。

例 1.2 $F2 \mid \text{chains} \mid C_{\max}$，表示两台机器的流水作业排序问题，工件之间具有平行链约束，排序指标为最小化工件最大完工时间。

1.2 算法和计算复杂性

优化算法是一种搜索过程或规则，通过一定的过程或规则得到问题的满意解。评价一个优化算法的优劣一方面看该算法得到的解是最优解还是近似解；另一方面看算法的时间和空间复杂性，因为算法的时间和空间复杂性对计算机的求解能力有很大影响。排序问题的时间复杂性是指求解该问题的所有算法中时间复杂性最小的算法的时间复杂性，问题的空间复杂性也可类似地定义。

1.2.1 算法及其复杂性

算法就是计算的方法之简称，它要求使用一组定义明确的规则在有限的步骤内求解某一问题。在计算机上，就是运用计算机解题的步骤或过程。在这个过程中，无论是形成解题思路还是编写程序，都是在实施某种算法。前者是推理实现的算法，后者是操作实现的算法。

对算法的分析，最基本的是对算法的复杂性进行分析，包括时间上的复杂性和空间上的复杂性。时间复杂性是指计算所需的步骤数或指令条数，空间复杂性是指计算所需的存储单元数量。在实际应用中，我们更多的是关注算法的时间复杂性。

算法的时间复杂性可以用一个变量 n 来表示，n 表示问题实例的规模，也就是该实例所需要输入数据的总量。一般在排序问题中，n 表示所要加工的总工件数。算法的时间量度记为 $T(n)=O(f(n))$，表示随问题规模 n 的增大，算法执行时间的增长率和 $f(n)$ 的增长率相同，称为算法的渐进时间复杂性，简称时间复杂性。同一算法求解同一问题的不同实例所需要的时间一般不相同，一个问题各种可能的实例中运算最慢的一种情况称为最“坏”情况或最“差”情况。一个算法在最“坏”情况下的时间复杂性称为该算法的最“坏”时间复杂性。一般情况下，时间复杂性都是指最“坏”情况下的时间复杂性。

由于算法的时间复杂性考虑的只是对于问题规模 n 的增长率，所以在难以精确计算基本操作次数的情况下，只需求出它关于 n 的增长率或阶即可。随问题规模的增大，不同的 $f(n)$ 会对 $T(n)$ 产生截然不同的效果。表 1-1 给出了不同时间复杂性算法在速度为 10^6 次/s 的计算机上求解不同规模问题所需时间的对比(马良 等，2008)。

表 1-1 不同时间复杂性算法求解不同规模问题所需时间的对比

$n/\mu s$	$\log_2 n/\mu s$	$n\log_2 n/\mu s$	$n^2/\mu s$	n^3	n^5	2^n	3^n
10	3.3	33	100	1ms	0.1s	1ms	59ms
40	5.3	213	1600	64ms	1.7min	12.7d	3855 世纪
60	5.9	354	3600	216ms	13min	366 世纪	1.3×10^{13} 世纪

一般情况下，当算法的时间复杂性 $T(n)$ 被输入规模 n 的多项式界定时，该算法为多项式时间算法，如 $T(n)$ 为 n 的对数函数或线性函数的算法，这样的算法是可接受的，也是实际有效的，因此又称为“有效算法”或“好”的算法；反之，称非多项式时间算法为指数算法，如 $T(n)$ 为 n 的指数函数或阶乘函数的算法，这样的算法大部分无法应用，没有实用价值，因此又称为“坏”的算法。

1.2.2 计算复杂性

一个最优化问题有三种提法：最优化形式、计值形式和判定形式。当讨论最优化问题的难易程度时，一般按其判定形式的复杂性对问题进行分类。一个最优化问题的判定形式可以描述为：给定任意一个最优化问题

$$\min_{x\in X} f(x),$$

问是否存在可行解 x_0，使得 $f(x_0)\leqslant L$。其中 X 为可行解集，L 为阈值。

我们把所有可用多项式时间算法解决的判定问题类称为 P 类，P 类是相对容易的判定问题类，它们有有效算法。如最大匹配问题和最小支撑树问题都是 P 类问题。还有一个重要的判定问题类是 NP 类，这类问题比较丰富。对于一个 NP 类问题，我们不要求它的每个实例都能用某个算法在多项式时间内得到解答，我们只要求：如果 x 是问题的答案为是的实例，则存在对于 x 的一个简短（其长度以 x 的长度的多项式为界）证明使得能在多项式时间内检验这个证明的真实性（Papadimitriou et al.，1988）。简而言之，以上述的最优化问题为例，给定任何一个可行解 x_0，如果存在一个多项式时间算法，该算法可以判断 x_0 是否小于或等于 L，则该问题是 NP 问题。容易证明 P$\subseteq$NP。

给定 H_1、H_2 这两个判定问题，如果存在一个多项式时间算法 α_1，将问题 H_1 的每个答案为“是”的实例 X，都转换为 H_2 问题，则把 α_1 称为 H_1 到 H_2 的多项式时间归结（Papadimitriou et al.，1988）。

对于一个判定问题 H，如果能够证明 $H\in$ NP，并且所有其他的 NP 问题都能在多项式时间内归结到 H，则称判定问题 H 是 NP-完备（NP-complete）的。NP-完备问题是 NP 类中“最难的”问题，一般认为它不存在多项式时间算法。例如，整数线性规划问题、三维匹配、点覆盖和团等问题都是 NP-完备的。

给定两个优化问题 H_1 和 H_2，若存在一个解决问题 H_2 的算法多次调用解决问题 H_1 的算法(假定解决问题 H_1 的算法复杂度为 1)，且解决问题 H_2 的算法复杂度不超过问题 H_2 某实例规模的多项式倍，则称问题 H_2 可以多项式时间图灵规约到问题 H_1。换言之，若问题 H_1 是多项式时间可解的，则问题 H_2 也是多项式时间可解的。

对于一个优化问题或判定问题 H，如果能够证明 NP 类中所有问题都可以在多项式时间内图灵规约到 H，则称 H 是 NP-难的。例如，Max-2Sat 问题和背包问题等是 NP-难的。在有的文献中，NP-完备和 NP-难的概念混用，不作严格区分。

1.2.3 排序问题的求解

对排序问题的求解主要有两个方向。一是对 H 问题，即可解问题，寻找多项式时间算法(又称有效算法)来得到问题的最优解，或者对 NP-难的在特殊情况下(如工件加工允许中断，工件的加工时间都是单位长度，工件之间有某种约束，等等)寻找有效算法，也就是研究 NP-难的可解情况；二是设计性能优良的近似算法和启发式算法。

对于使目标函数 f 为最小的优化问题，记 I 为这个优化问题的一个实例，H 为所有实例的全体；并记 $f(I)$ 为实例 I 的最优目标函数值(即最优值)，$f_H(I)$ 为利用算法 H 得到的目标函数值。如果存在一个实数 $r(r \geqslant 1)$，使得对任意 $I \in H$ 有

$$f_H(I) \leqslant rf(I),$$

则称 r 为算法 H 的一个上界。当 r 是有限数时，称算法 H 为 r 近似算法；当不能确定 r 是否有限，或能确定 r 为无穷大时，则称算法 H 为启发式算法。用近似算法和启发式算法得到的解分别称为近似解和启发式解。使上式成立的最小正数 r 称为算法的最坏情况性能比或紧界。

对于使目标函数 f 为最大的优化问题，同样可以定义算法的下界 r 满足 $0 < r \leqslant 1$，对任意 $I \in H$ 有

$$f_H(I) \geqslant rf(I),$$

而最坏情况性能比或紧界是使上式成立的最大正数 r。

无论是近似算法还是启发式算法，都是求解排序问题的方法，是方法就有好坏和优劣之分。如何衡量它的好坏呢？一般对近似算法用得较多的是最坏情况的理论分析，从而找到最小的紧界；而对启发式算法用得较多的是数值算例的计算，通过将启发式算法得到的目标函数值下界与其他算法得到的目标函数值或商业软件(如 CPLEX、Gurobi、COPT)算得的结果相比较，得出该启发式算法的性能。当然，有时也可综合使用多种方法来分析和衡量算法的性能。

第 2 章　博弈论简介

博弈论(game theory)是以数学为主要分析工具,研究在包含多个决策者或者行为主体的局势中,各决策者之间彼此存在交互性决策行为的理论。一般认为,博弈主要可以分为合作博弈和非合作博弈。非合作博弈研究人们在利益相互影响的局势中如何做出决策以使自己的收益最大,即策略选择问题。合作博弈主要研究人们达成合作的条件及如何分配合作得到的收益,以及收益分配问题。两者的区别在于相互发生作用的当事人之间有没有一个具有约束力的协议,如果有,就是合作博弈;否则,就是非合作博弈。本章主要介绍具有转移效用的联盟博弈、纳什讨价还价问题和算法博弈论。

2.1　联盟博弈

合作博弈理论主要关心的是联盟(即参与人集合,coalition),协调他们的行动并且经营他们的收益。因此,这里的核心问题是如何在组成联盟的成员之间分配他们的额外收益(或节省费用)。这个理论是基于 von Neumann 和 Morgenstern 建立的具有特征函数的合作博弈(von Neumann,Morgenstern,1944),也就是具有转移效用的合作博弈(TU-博弈)。TU-博弈产生了很多的解概念,同时也产生了一些有趣的 TU-博弈的子类。

令 N 为参与人的非空有限集合,不妨设 $N=\{1,2,\cdots,n\}$。参与人考虑不同的合作可能性,每个子集 $S\subset N$ 可看作一个联盟。集合 N 称为大联盟(grand coalition),集合 $\varnothing$ 称为空联盟(empty coalition)。用 2^N 表示 N 的所有子集组成的集合。下面简要给出具有转移效用的联盟博弈的概念。

定义 2.1　具有转移效用的联盟博弈是一序对 $\langle N,v\rangle$,其中 N 为参与人集合,$v:2^N\to\mathbb{R}$ 是满足 $v(\varnothing)=0$ 的特征函数。

实值函数 $v(S)$ 可以解释为当联盟 S 中的参与人合作时,可以获得的最大收益或可节省的最多费用,即可用于成员间分配的总收益。通常称 $\langle N,v\rangle$ 为具有特征函数 v 的博弈。

定义 2.2　在联盟博弈 $\langle N,v\rangle$ 中,如果对所有满足 $S\cap T=\varnothing$ 的 $S,T\in 2^N$,有

$$v(S\cup T)\geqslant v(S)+v(T),$$

则称联盟博弈 $\langle N,v\rangle$ 是超可加的。

如果 $S_1,S_2,\cdots,S_k$ 是两两不相交的联盟,则称 $(S_1,S_2,\cdots,S_k)$ 为 N 的一个分割。在一个超可加博弈中,对于任意分割 $(S_1,S_2,\cdots,S_k)$,都有 $v\left(\bigcup_{i=1}^{k}S_i\right)\geqslant\sum_{i=1}^{k}v(S_i)$,即 $v(N)\geqslant\sum_{i=1}^{k}v(S_i)$。特别地,$v(N)\geqslant\sum_{i=1}^{n}v(i)$。因此,对于一个满足超可加性的博弈,合作对参与人是有利的。

定义 2.3 在联盟博弈 $\langle N,v\rangle$ 中,如果对所有 $S,T\in 2^N$,有

$$v(S\cup T)+v(S\cap T)\geqslant v(S)+v(T),\tag{2.1}$$

则称联盟博弈 $\langle N,v\rangle$ 是凸的。

凸博弈还可以解释为:对于 $\forall i\in N$ 和所有 $S\subset T\subset N\backslash\{i\}$,有

$$v(S\cup\{i\})-v(S)\leqslant v(T\cup\{i\})-v(T)。\tag{2.2}$$

定义 2.4 关于博弈 $\langle N,v\rangle$,称 $v(S\cup\{i\})-v(S)$ 为参与人 i 对于联盟 S 的边际贡献,记为 $M_i(S,v)=v(S\cup\{i\})-v(S)$。

令 $\pi(N)$ 为 N 的所有排列 $\sigma:N\to\{1,2,\cdots,n\}$ 的集合。给定一个排列 σ,集合 $P(\sigma,i)=\{r\in N\mid\sigma^{-1}(i)\}$,即 $P(\sigma,i)$ 含有 σ 中所有 i 的前继。下面给出边际贡献向量的定义。

定义 2.5 关于博弈 $\langle N,v\rangle$,令 $\sigma\in\pi(N)$,边际贡献向量 $\boldsymbol{m}^\sigma(v)\in\mathbb{R}^n$,其中 $\boldsymbol{m}_i^\sigma(v)=v(P(\sigma,i)\cup\{i\})-v(P,i)$。

在不致引起混淆的情况下,称 $\boldsymbol{m}^\sigma(v)$ 为边际向量。

式(2.2)意味着参与人对某个联盟的边际贡献随着联盟规模的扩大而增加。因此,在一个凸博弈中,形成大联盟对所有参与人是有利的。

对于每个 $S\in 2^N$,用 $|S|$ 表示 S 中元素的个数,用 $\mathbf{1}^S$ 表示 S 的特征向量,其中

$$\begin{cases}1_i^S=1,i\in S\\1_i^S=0,i\in N\backslash S\end{cases}。$$

下面给出一类重要的具有转移效用的联盟博弈——均衡博弈的概念。

定义 2.6 关于映射 $\lambda:S\in 2^N\backslash\{\varnothing\}\to\mathbb{R}^+$,如果

$$\sum_{S\in 2^N\backslash\{\varnothing\}}\lambda(S)1^S=1^N,$$

其中 $\sum_{S:i\in S}\lambda(S)=1$,则称 $\lambda:S\in 2^N\backslash\{\varnothing\}\to\mathbb{R}^+$ 为均衡映射。

定义 2.7 关于联盟 B,如果存在一个均衡映射 λ 使得 $B=\{S\in 2^N\mid\lambda(S)>0\}$,则称联盟 B 是均衡的。

定义 2.8　在博弈 $\langle N,v\rangle$ 中，如果对任意的均衡映射 $\lambda: S\in 2^N\setminus\{\varnothing\}\to\mathbb{R}^+$，都有

$$\sum_{S\in 2^N\setminus\{\varnothing\}}\lambda(S)v(S)\leqslant v(N), \tag{2.3}$$

则称博弈 $\langle N,v\rangle$ 是均衡的。

具有转移效用的联盟博弈的基本问题是：如何形成大联盟 N，并且如何分配收益或节省费用 $v(N)$。解决这个问题依赖于合作博弈的解概念，例如，核心、稳定集、Shapley 值和 τ 值等。一个解概念就给出上述问题的一个答案，即当 N 中所有参与人合作时所获得的收益或节省费用，如何在参与人之间进行分配，同时这个分配要考虑参与人形成不同联盟时可能存在的潜在收益或节省费用。因此，一个解概念至少对应着一个支付向量 $\boldsymbol{x}\in\mathbb{R}^n$，其中 x_i 是分配给参与人 $i\in N$ 的支付。合作博弈的解概念一般可以分为集值解和单点解，例如，核心和稳定集是集值解，Shapley 值和 τ 值是单点解。

首先给出集值解核心的定义。

在博弈 $\langle N,v\rangle$ 中，一个支付向量 $\boldsymbol{x}\in\mathbb{R}^n$ 只有满足有效性才有可能被接受，即

$$\sum_{i\in N}x_i=v(N)。$$

如果在建议的支付向量 $\boldsymbol{x}$ 中，至少有一个参与人 $i\in N$ 的支付 x_i 满足 $x_i<v(i)$，则这样的参与人将选择不参加合作，从而导致大联盟不可能形成。因此，在博弈 $\langle N,v\rangle$ 中，若想实现支付向量 $\boldsymbol{x}$，则个体合理性必须成立，即

$$x_i\geqslant v(i)，对所有的\ i\in N。$$

Gillies 于 20 世纪 50 年代提出了核心的概念，核心的思想类似于非合作博弈的纳什均衡：如果没有支付的偏离，那么结果就是稳定的。下面给出核心的定义(Gillies，1953)。

定义 2.9　博弈 $i,j\in N$ 的核心 $C(v)$ 是一个包含所有满足有效性和联盟合理性的支付向量 $\boldsymbol{x}\in\mathbb{R}^n$ 的集合，即

(1) $\sum\limits_{i\in N}x_i=v(N)$；

(2) $\sum\limits_{i\in S}x_i\geqslant v(S)$，对所有的 $S\in 2^N\setminus\{\varnothing\}$ 成立。

联盟合理性条件 $\sum\limits_{i\in S}x_i\geqslant v(S)$ 包含了个体合理性条件。如果 $\boldsymbol{x}\in C(v)$，则当 $\boldsymbol{x}$ 作为 N 中参与人的收益支付时，不存在联盟 S 有分裂的动机，这是因为分配给 S 的总量 $\sum\limits_{i\in S}x_i\geqslant v(S)$，而 $v(S)$ 是 S 的参与人形成子联盟时可以获得的最大收益。

核心在理论上具有重要的地位，同时核心 $C(v)$ 中的元素可以通过求解线性不等式系统较容易地获得。但是核心可能是空集，还可能具有相当多的解，下面的例子说明了联盟博弈核心的这一特征。

例 2.1 （三人生产问题）假设有一项任务，三个人合作完成这项任务可以获得 1 个单位的支付，任意两个人合作完成这项任务可以获得 α 单位的支付（$\alpha \in [0,1]$），单独一个人无法完成这项任务。

我们可以把上面的生产问题建模为联盟博弈模型 $\langle N,v\rangle$，其中 $N=\{1,2,3\}$，$v(N)=1$；$v(S)=\alpha$，$|S|=2$；$v(i)=0$，$i\in N$。那么，这个博弈的核心 $C(v)$ 包含了所有支付向量 $\boldsymbol{x}=(x_1,x_2,x_3)^{\mathrm{T}}$，有 $\sum\limits_{i\in N}x_i=1$，$\sum\limits_{i\in S}x_i=\alpha$，对于任一包含两个参与人的联盟 S。显然，当 $\alpha<\dfrac{2}{3}$ 时，$C(v)$ 非空且解不唯一；当 $\alpha=\dfrac{2}{3}$ 时，$C(v)$ 包含唯一解 $\left(\dfrac{1}{3},\dfrac{1}{3},\dfrac{1}{3}\right)^{\mathrm{T}}$；当 $\alpha>\dfrac{2}{3}$ 时，$C(v)$ 是空集。

关于非空核心与均衡博弈有下面的结果（Bondareva，1963；Shapley，1967）。

定理 2.1 可转移效用的联盟博弈有非空核心当且仅当该博弈是均衡的。

证明：令 $\langle N,v\rangle$ 为一个转移效用的联盟博弈。首先，假设核心 $C(v)\neq\varnothing$，取 $x\in C(v)$。令 $\lambda:S\in 2^N\setminus\{\varnothing\}\to\mathbb{R}^+$ 是一个均衡映射，则

$$\sum_{S\in 2^N\setminus\{\varnothing\}}\lambda(S)v(S)\leqslant\sum_{S\in 2^N\setminus\{\varnothing\}}\lambda(S)x(S)=\sum_{i\in N}x_i\sum_{i\in S}\lambda(S)$$
$$=\sum_{i\in N}x_i=v(N),$$

因此，博弈 $\langle N,v\rangle$ 是均衡的。

现在假设 $\langle N,v\rangle$ 是均衡的，那么不存在均衡映射 $\lambda:S\in 2^N\setminus\{\varnothing\}\to\mathbb{R}^+$，使得 $\sum\limits_{S\in 2^N\setminus\{\varnothing\}}\lambda\{S\}v\{S\}>v\{N\}$。因此，凸集

$$\{(\mathbf{1}^N,v(N)+\varepsilon)\in\mathbb{R}^{n+1}\mid\varepsilon>0\}$$

和凸锥

$$\left\{\boldsymbol{y}\in\mathbb{R}^{n+1}\mid\boldsymbol{y}=\sum_{S\in 2^N\setminus\{\varnothing\}}\lambda(S)(\mathbf{1}^S,v(S)),\lambda(S)\geqslant 0,S\in 2^N\setminus\{\varnothing\}\right\}\tag{2.4}$$

是不相交的。事实上，如果它们是相交的，则存在均衡映射 $\lambda:S\in 2^N\setminus\{\varnothing\}\to\mathbb{R}^+$ 使得 $\sum\limits_{S\in 2^N\setminus\{\varnothing\}}\lambda(S)\mathbf{1}^S=\mathbf{1}^N$，于是 $\lambda(S)$ 是均衡映射并且有 $\sum\limits_{S\in 2^N\setminus\{\varnothing\}}\lambda(S)v(S)>v(N)$，矛盾。由凸集分离定理（见文献（Rockafellar，2017）中的定理 11.3），存在一个非零向量 $(\boldsymbol{\alpha}^N,\alpha)\in\mathbb{R}^n\times\mathbb{R}$，对于凸锥（式(2.4)）中的任一 $\boldsymbol{y}$ 和 $\forall\varepsilon>0$，有

$$(\boldsymbol{\alpha}^N, \alpha) \cdot \boldsymbol{y} \geqslant 0 > (\boldsymbol{\alpha}^N, \alpha) \cdot (\mathbf{1}^N, v(N)+\varepsilon)。 \tag{2.5}$$

既然 $(\mathbf{1}^N, v(N))$ 也在凸锥(式(2.4))中,则式(2.5)可写为

$$\boldsymbol{\alpha}^N \sum_{S \in 2^N \setminus \{\varnothing\}} \lambda(S) \mathbf{1}^S + \alpha \sum_{S \in 2^N \setminus \{\varnothing\}} \lambda(S) v(S) \geqslant 0 > \boldsymbol{\alpha}^N \mathbf{1}^N + \alpha(v(N)+\varepsilon)。$$

由定义 2.6 知 $\sum_{S:i \in S} \lambda(S)=1$, 则由上式可得

$$\boldsymbol{\alpha}^N \mathbf{1}^S + \alpha v(S) \geqslant 0 > \boldsymbol{\alpha}^N \mathbf{1}^N + \alpha(v(N)+\varepsilon),$$

所以 $\alpha < 0$。

令 $\boldsymbol{x} = -\frac{\boldsymbol{\alpha}^N}{\alpha}$。对于 $\forall S \in 2^N \setminus \{\varnothing\}$, 既然 $(\mathbf{1}^S, v(S))$ 在凸锥(式(2.4))中,由式(2.5)的左边不等式,可以得到 $\sum_{i \in S} x_i = \mathbf{1}^S x \geqslant v(S)$; 由式(2.5)的右边不等式可以得到 $v(N) \geqslant \mathbf{1}^N \boldsymbol{x} \geqslant \sum_{i \in N} x_i$。因此, $v(N) = \sum_{i \in N} x_i$, 所以有 $\boldsymbol{x} \in C(v)$。 □

凸博弈是一类常见的均衡博弈。凸博弈具有很好的性质,例如,凸博弈的核心是所有边际贡献向量的凸包(Shapley,1971;Ichiishi,1981)。

下面给出 Shapley 值和 τ 值等单点解的定义。

对于给定的联盟博弈 $\langle N, v\rangle$, Shapley 值对应于一个 $\mathbb{R}^n$ 中的支付向量。Shapley 值的第一个形式用的是博弈边际向量。

定义 2.10　对于博弈 $\langle N, v\rangle$, Shapley 值是博弈边际向量的平均值,记作 $\boldsymbol{\Phi}(v)$, 即

$$\boldsymbol{\Phi}(v) = \frac{1}{n} \sum_{\sigma \in \pi(N)} \boldsymbol{m}^{\sigma}(v), \tag{2.6}$$

其中

$$\boldsymbol{\Phi}_i(v) = \frac{1}{n!} \sum_{\sigma \in \pi(N)} [v(P(\sigma, i) \cup \{i\}) - v(P(\sigma, i))]。 \tag{2.7}$$

如果我们用 $v(S \cup \{i\}) - v(S)$ 替换式(2.7)中 $v(P(\sigma, i) \cup \{i\}) - v(P(\sigma, i))$, 其中 S 为不包含 i 的 N 的子集,那么式(2.7)可以变为

$$\boldsymbol{\Phi}_i(v) = \sum_{S:i \in S} \frac{|S|!\ (n-1-|S|!\ )}{n!}(v(S \cup \{i\}) - v(S))。 \tag{2.8}$$

其中 $|S|!$ 对应于 S 的排列数, $(n-1-|S|!)$ 对应于 $N \setminus (S \cup \{i\})$ 的排列数, $|S|!(n-1-|S|!)$ 对应于 $P(\sigma, i)$ 的排列数。注意, $\frac{|S|!(n-1-|S|!)}{n!} = \frac{1}{n}\binom{n-1}{|S|}^{-1}$, 这给出了 Shapley 值的一种概率解释。首先在 $0,1,2,\cdots,n-1$ 之间生成一个随机数,每个数被选中的概率为 $\frac{1}{n}$。

如果 r 被选中,则从所有 $N\backslash\{i\}$ 的子集中选一个基数 r 的子集,不妨设子集 S 被选中,则这样的子集有同样的概率 $\binom{n-1}{|S|}^{-1}$ 被选中。那么,给第 i 个参与人的支付为 $v(S\cup\{i\})-v(S)$,即参与人 i 在他参与的合作 S 中做出的贡献。所以,第 i 个参与人的期望支付就是 Shapley 值。

Shapley 值是 Shapley 基于公理化思想提出来的单点解概念(Shapley,1953)。在给出 Shapley 公理化描述之前需要进行以下定义:

对于所有的 $S\in N$,如果有 $v(S\cup\{i\})-v(S)=v(\{i\})$,则称参与人 i 在 $\langle N,v\rangle$ 中是虚拟的(dummy);对于所有的 $S\in N$,如果 $v(S\cup\{i\})-v(S)=v(S\cup\{j\})-v(S)$,则称参与人 i 与 j 是可以互换的(interchangeable)。

(1) 有效性(efficiency):对任意博弈 $\langle N,v\rangle$,都有 $\sum_{i\in N}\boldsymbol{\Phi}_i(v)=v(N)$。

(2) 对称性(symmetry):如果 i 与 j 是可以互换的,则 $\boldsymbol{\Phi}_i(v)=\boldsymbol{\Phi}_j(v)$。

(3) 虚拟参与人(dummy player):如果 i 是虚拟的,则 $\boldsymbol{\Phi}_i(v)=v(i)$。

(4) 可加性(additivity):如果在 N 上有两个特征函数 v 和 w,对于 $\forall i\in N$,则 $\boldsymbol{\Phi}_i(v+w)=\boldsymbol{\Phi}_i(v)+\boldsymbol{\Phi}_i(w)$。

对称性说明合作获利的分配不随每个人在合作中的记号或次序变化而变化;虚拟参与人说明如果一个成员对于任何他参与的联盟都没有贡献,则他不应当从全体合作中获利;可加性表明有多种合作时,每种合作的利益分配方式与其他合作结果无关。对于 Shapley 值的其他公理化描述可参考有关文献(Branzei et al.,2008;施锡铨,2012)。

Shapley 证明了式(2.6)确定的满足上述四条公理的解是唯一的(Shapley,1953)。

定理 2.2 Shapley 值是唯一的满足对称性、虚拟参与人、可加性与有效性的解。

对于每一个博弈 $\langle N,v\rangle$ 都存在一个唯一的 Shapley 值。因此,与核心相比,Shapley 值在实际中应用得更为普遍。但是 Shapley 值在一定场合也是有缺陷的。例如,在二人合作博弈中,由式(2.7)确定的两个参与人获得的支付是相同的。但实际上,在多数场合这个结论是不成立的,特别是一个处于优势的参与人与处于劣势的参与人合作的时候。

2.2 纳什讨价还价问题

本节从联盟博弈的角度简单介绍两人讨价还价问题及其求解过程。

讨价还价是对已有的或者合作之后能够得到的利益或节省费用的分配。如果两个参与人通过谈判形成合作,那么合作产生的效用(获得的利益或费用

节省)记为 $v(\{1,2\})$，令谈判的结果是参与人 1 获得支付 x_1，参与人 2 获得支付 x_2，则有 $x_1+x_2=v(\{1,2\})$。这样的 (x_1,x_2) 构成了一个可行配置(支付向量),所有可行的配置 (x_1,x_2) 形成了两人讨价还价问题的可行配置集，记为 F。

取 F 中的两个配置 (x_1,x_2) 和 (y_1,y_2)，令 $\theta\in[0,1]$，由前面的讨论知道 $\theta(x_1,x_2)+(1-\theta)(y_1,y_2)$ 也是一个可行配置。因此,可行配置集 F 是一个凸集。如果谈判失败,即在不合作状态下,参与人分别获得支付 $v(\{1\})$ 与 $v(\{2\})$，令 $e_1=v(\{1\})$，$e_2=v(\{2\})$，称在不合作状态下的效用配置(disagreement payoff allocation) (e_1,e_2) 为无协议点(Muthoo,1999)。e_1 和 e_2 相当于两个参与人在讨价还价过程中各自坚持的底线。考虑到博弈的个体理性性质,如果 (x_1,x_2) 是最终谈判结果,则有 $x_1\geqslant e_1$，$x_2\geqslant e_2$。

下面给出两人讨价还价问题的定义(施锡铨,2012)。

定义 2.11　两人讨价还价问题是一个序对 $\langle F,\boldsymbol{e}\rangle$，其中 F 表示 $\mathbb{R}^2$ 上的一个闭凸子集，$\boldsymbol{e}=(e_1,e_2)$ 是 $\mathbb{R}^2$ 中的一个点，且 $F\cap\{(x_1,x_2)\mid x_1\geqslant e_1, x_2\geqslant e_2\}$ 是非空有界集合。

在两人讨价还价问题中,无法达成合作时的无协议点 (e_1,e_2) 对于求解博弈的解有着至关重要的作用。但我们在讨论两人讨价还价问题时,总是假设 (e_1,e_2) 是事先知道的。

研究两人讨价还价问题,其目的是从 $F\cap\{(x_1,x_2)\mid x_1\geqslant e_1,x_2\geqslant e_2\}$ 中找出一个或若干个合理的效用配置,作为参与人的谈判结果。记这样的谈判结果为 $\varphi(F,\boldsymbol{e})$，即两人讨价还价问题的解。在研究两人讨价还价问题的解时,从不同的角度考虑会得到不同的解概念。纳什从公理的角度提出讨价还价解的概念。下面简要介绍产生纳什讨价还价解的公理系统。

令 $\boldsymbol{\varphi}(F,\boldsymbol{e})=(\varphi_1(F,\boldsymbol{e}),\varphi_2(F,\boldsymbol{e}))\in F$ 为讨价还价问题的解,其中 $\varphi_1(F,\boldsymbol{e})$ 和 $\varphi_2(F,\boldsymbol{e})$ 分别表示参与人 1 和 2 的支付。

(1) 个体合理性(individual rationality)：$\varphi_1(F,\boldsymbol{e})\geqslant v_1$，$\varphi_2(F,\boldsymbol{e})\geqslant v_2$。

(2) 帕累托有效性(Pareto efficiency)：对于 $\forall\boldsymbol{x}\in F$，如果 $\boldsymbol{x}\geqslant\boldsymbol{\varphi}(F,\boldsymbol{e})$，则有 $x_1=\varphi_1(F,\boldsymbol{e})$，$x_2=\varphi_2(F,\boldsymbol{e})$。

(3) 对称性(symmetry)：如果 $(x_1,x_2)\in F$，则有 $(x_2,x_1)\in F$，并且若 $e_1=e_2$，则 $\varphi_1(F,\boldsymbol{e})=\varphi_2(F,\boldsymbol{e})$。

(4) 无关选择的独立性(independence of irrelevant alternatives)：假设 $\langle F,v\rangle$ 和 $\langle F',v'\rangle$ 分别是两个两人讨价还价问题,其中 $v=v'$，如果 $F'\subseteq F$，并且 $\boldsymbol{\varphi}(F,\boldsymbol{e})\in F'$，那么 $\varphi(F,v)=\varphi(F',v')$。

帕累托有效性说明在 F 中找不到这样的配置 $\boldsymbol{x}$，使得任何参与人觉得 $\boldsymbol{x}$ 比 $\boldsymbol{\varphi}(F,v)$ 更令人满意。对称性的第二点说明，在讨价还价中地位相等的参与人，最终的结果应该给予他们相同的待遇。

Nash 证明了满足上述公理的两人讨价还价问题存在唯一的讨价还价解(Nash，1950，1953)。

定理 2.3 对于两人讨价还价问题 $(F,\boldsymbol{e})$，存在满足个体合理性、帕累托有效性、对称性、无关选择的独立性的唯一讨价还价解，它使纳什积达到最大的 (x_1,x_2)。也就是纳什讨价还价解是如下问题的解：

$$\boldsymbol{\varphi}(F,v) \in \arg_{\boldsymbol{x}\in F} \max_{x_1 \geqslant e_1, x_2 \geqslant e_2} (x_1 - e_1)(x_2 - e_2)。 \tag{2.9}$$

2.3 算法博弈论

算法博弈论作为计算机理论科学和博弈论的一个新的交叉研究领域，运用算法分析的理论和方法，从具体优化问题的角度在博弈论框架下进行应用建模，寻求最优解、判断不可解问题以及研究可解优化的上下限问题；讨论问题的可计算性，即参与人能否可以在多项式时间内达到一个均衡状态；同时为了令社会成员参与其中，得出的博弈解恰好符合设计者所想达到的社会选择，它也研究如何设计一个博弈形式，或者称作机制，促使参与人在自身利益驱动下选择设计者期望的策略，实现符合设计目标的系统总体均衡态。本节主要介绍本书涉及的一些概念及其定义。

一个博弈由下面三个要素刻画：一群局中人(player)；每个局中人必须选择一个策略(strategy)；所有局中人决策后形成的局势(strategy profile)，即指具有完全信息的静态博弈。根据这个模型，所有的局中人独立并且同时做出他们的决定。因此一个 n 人策略式博弈的严格定义如下。

定义 2.12 策略博弈可用一个三元组 $G=\{N,A,c\}$ 表示。其中 N 表示局中人集合；对于每个局中人 $i\in N$，有一个非空的策略空间 A_i。令 $A=\prod_{i\in N} A_i$ 为所有局势的集合；对于每个局中人 $i\in N$，有一个费用函数 c_i：$A\to\mathbb{R}$，并且 $c=\prod_{i\in N} c_i$。

定义 2.13 设三元组 $G=\{N,A,c\}$ 是一个策略式博弈，如果博弈 G 有有限个局中人并且对每个局中人而言都只有有限个策略可供选择，那么我们就称这个博弈 G 为有限策略博弈。

纳什均衡(Nash equilibrium)是博弈论中一个里程碑式的均衡概念。

定义 2.14　设三元组 $G=\{N,A,c\}$ 是一个策略式博弈，具有如下性质的一个局势 $a^* \in A$ 称为纳什均衡：对每个局中人 $i \in N$ 和每个策略 $a_i^- \in A_i$，都有 $c_i(a^*) \leqslant c_i(a_{-i}^*, a_i^-)$。这里 $a_{-i}^*=\{a_1^*, a_2^*, \cdots, a_{i-1}^*, a_{i+1}^*, \cdots, a_n^*\}$。

Aumann 最先提出强均衡（strong equilibrium）的概念（Aumann，1959）。在强均衡中，没有联盟能够偏离当前的局势使联盟中每个成员的费用均减少。

定义 2.15　设三元组 $G=\{N,A,c\}$ 是一个策略式博弈，一个局势 $a^* \in A$ 称为强均衡，若它具有如下性质：对由局中人组成的任意一个联盟 Γ 以及 Γ 的任意一个偏离 a_Γ，至少有一个局中人 $i \in \Gamma$ 满足 $c_i(a^*) \leqslant c_i(a_{-\Gamma}^*, a_\Gamma)$。

由定义 2.15 可以看出，任意强均衡都是纳什均衡，但反之不然。通常用定义在局势上的实值函数来反映博弈的社会费用（social cost）或者博弈的社会效益（social utility）。

类似于优化问题中的目标函数，所谓最优局势是社会费用最小或社会效益最大的局势。注意到纳什均衡并不总是最优局势，为了衡量均衡的效率，引入两种衡量标准无秩序代价（price of anarchy，POA）和稳定代价（price of stability，POS）。POA 是指博弈中最坏纳什均衡的目标函数值与最优值的比值，POS 是指博弈中最好纳什均衡的目标函数值与最优值的比值。类似地，可以定义强无秩序代价（strong price of anarchy，SPOA）和强稳定代价（strong price of stability，SPOS）。SPOA 是指博弈中最坏强均衡的目标函数值与最优值的比值，SPOS 则是指博弈中最好强均衡的目标函数值与最优值的比值。具体地，令 G 为一个策略式博弈，其最优准则是极小化社会费用，$N(G)$、$S(G)$ 分别表示它的所有纳什均衡、强均衡组成的集合，$C(a)$ 表示 G 中局势 a 的社会费用，C^* 表示最优社会费用。用 POA(G)、SPOA(G)、POS(G)、SPOS(G) 分别表示博弈 G 的 POA、SPOA、POS、SPOS，则

$$\mathrm{POA}(G)=\max\left\{\frac{C^N}{C^*}\middle| N \in N(G)\right\}=\frac{\max\{C^N \mid N \in N(G)\}}{C^*},$$

$$\mathrm{SPOA}(G)=\max\left\{\frac{C^N}{C^*}\middle| N \in S(G)\right\}=\frac{\max\{C^N \mid N \in S(G)\}}{C^*},$$

$$\mathrm{POS}(G)=\min\left\{\frac{C^N}{C^*}\middle| N \in N(G)\right\}=\frac{\min\{C^N \mid N \in N(G)\}}{C^*},$$

$$\mathrm{SPOS}(G)=\min\left\{\frac{C^N}{C^*}\middle| N \in S(G)\right\}=\frac{\min\{C^N \mid N \in S(G)\}}{C^*}。$$

类似地，若最优准则是极大化社会效益，则 POA(G)、SPOA(G)、POS(G)、SPOS(G) 可分别定义为

$$\mathrm{POA}(G)=\max\left\{\frac{C^*}{C^N}\,\middle|\,N\in N(G)\right\}=\frac{C^*}{\min\{C^N\mid N\in N(G)\}},$$

$$\mathrm{SPOA}(G)=\max\left\{\frac{C^*}{C^N}\,\middle|\,N\in S(G)\right\}=\frac{C^*}{\min\{C^N\mid N\in S(G)\}},$$

$$\mathrm{POS}(G)=\min\left\{\frac{C^*}{C^N}\,\middle|\,N\in N(G)\right\}=\frac{C^*}{\max\{C^N\mid N\in N(G)\}},$$

$$\mathrm{SPOS}(G)=\min\left\{\frac{C^*}{C^N}\,\middle|\,N\in S(G)\right\}=\frac{C^*}{\max\{C^N\mid N\in S(G)\}}。$$

易知 $\mathrm{POA}(G)\geqslant\mathrm{SPOA}(G)\geqslant\mathrm{SPOS}(G)\geqslant\mathrm{POS}(G)$。对算法博弈论的详细介绍可参看专著 *Algorithmic game theory*（Nisan，2007）。本书第 6 章和第 7 章介绍非合作排序博弈，它是对经典排序问题从非合作博弈的角度予以分析研究，其主要内容是对均衡效率进行分析，这也是算法博弈论的重要内容。

第 3 章　联盟排序博弈

在过去的四十多年里，产生了许多组合优化与合作博弈论的交叉研究领域。例如，分配博弈（Shapley et al.，1972），最小支撑树博弈（Granot et al.，1981），中国邮递员博弈（Granot et al.，1999）等。在排序论与合作博弈理论的交叉领域，Curiel 等最早研究了联盟排序博弈问题（Curiel et al.，1989），自此联盟排序博弈问题受到了广泛的关注，对于这一问题取得了丰富的研究成果（Curiel et al.，1994；Hamers et al.，1995；Borm et al.，2002；Hamers et al.，1999；Calleja et al.，2002；van Velzen et al.，2003）。联盟排序博弈的研究一般需要解决两个问题，一个是极小化总费用或者极大化总收益，另一个是如何在参与人之间分配节省的费用或者获得的收益。前者需要利用组合优化的理论技术方法进行处理，后者是在合作博弈理论研究范畴内解决。本章主要介绍联盟排序博弈的基本概念、模型建立以及基本联盟排序博弈问题的一些解和性质。

3.1　引言

经典排序问题是一个单决策者问题，即假设属于同一个决策者的有限多个工件在一台或者多台机器上按照顺序依次加工，这个决策者将决定工件的加工顺序从而使某排序指标达到最优。然而，在实际的生产活动中存在许多类似的多决策者问题，工件属于不同的决策者（agents），后面也称为参与人，并且工件已经有一个初始的排序。例如，在某银行服务大厅内，多个客户在柜台前站成一排等待服务，每个客户有一个任务，已知每个客户的任务的服务时间，并且每个客户的费用函数是客户服务完毕时间的不减函数。多个决策者通过合作（联盟）即联合行动共同决定工件的加工顺序，能够产生节省费用。如何产生最大的节省费用，以及如何在参与合作的参与人中分配这些节省的费用，是这个多决策者问题需要解决的。这样的多决策者问题可以建模为一个可转移效用的合作博弈问题，即联盟排序博弈。本节以一台机器为例讨论联盟排序博弈等相关概念及性质。

在讨论如何产生最大的节省费用之前，首先给出非常重要的概念，即排序局势的定义。

在一个排序局势(sequencing situation)中,有 n 个参与人 $N=\{1,2,\cdots,n\}$,每个参与人有一个工件,所有工件将在一台机器上加工,参与人 i 的工件的加工时间为 p_i。假设对于所有参与人(或工件)已存在一个初始排序 $\sigma_0:N \to \{1,2,\cdots,n\}$,$\sigma_0(i)=j$ 表示参与人 i 的工件在位置 j 加工。对于每一个参与人 $i \in N$,其加工费用函数表示为 $c_i(t)=\alpha_i t,\alpha_i>0$。即 $c_i(t)$ 表示当参与人 i 的工件在 t 时刻完工时的费用。在不致引起混淆的情况下,i 既表示第 i 个参与人,也表示第 i 个工件。

定义 3.1 一个排序局势可以表示为 $\langle N,\sigma_0,\boldsymbol{p},\boldsymbol{\alpha}\rangle$,其中 $N=\{1,2,\cdots,n\}$ 是参与人集合,$\sigma_0:N \to \{1,2,\cdots,n\}$ 是初始排序,$\boldsymbol{p}=(p_1,p_2,\cdots,p_n)\in \mathbb{R}_0^n$,$\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n)\in \mathbb{R}_0^n$。

令 $\sigma \in \pi(N)$,其中 $\pi(N)$ 表示所有工件的可能排序(排列)集合。这里假设在所有排序中,任意一对相邻加工的工件之间机器没有空闲。工件 i 在 σ 中的开始加工时间记为 $t_{\sigma,i}$,则

$$t_{\sigma,i}=\begin{cases}t_{\sigma,j}+p_j, & \sigma(i)>1\\ 0, & \sigma(i)=1\end{cases},$$

其中 j 是工件 i 的紧前工件,即 $\sigma(j)=\sigma(i)-1$。那么工件 i 在 σ 中的完工时间 $C(\sigma,i)=t_{\sigma,i}+p_i$。

对于排序局势 $\langle N,\sigma_0,\boldsymbol{p},\boldsymbol{\alpha}\rangle$,显然存在一个排序使得总费用 $\sum\limits_{i\in N}c_i(t)$ 最小,从而获得最大节省费用。这样的排序称为排序局势 $\langle N,\sigma_0,\boldsymbol{p},\boldsymbol{\alpha}\rangle$ 的最优排序,记为 $\hat{\sigma}_N \in \pi(N)$。为简便起见,排序局势 $\langle N,\sigma_0,\boldsymbol{p},\boldsymbol{\alpha}\rangle$ 的最优排序也记为 $\hat{\sigma}$。

在最优排序中,所有工件按照紧急系数 $\mu_i=\dfrac{\alpha_i}{p_i}$ 不增的顺序加工(Smith,1956)。因此,一个最优排序可以由初始排序 σ_0 连续执行以下程序获得:如果存在一对相邻加工的两个工件 i 和 j,工件 i 在 j 之前加工,且 $\mu_i<\mu_j$,则交换 i 和 j 的加工顺序。

下面利用合作博弈理论,首先基于排序局势 $\langle N,\sigma_0,\boldsymbol{p},\boldsymbol{\alpha}\rangle$ 建立联盟排序博弈模型,然后讨论一些分配规则,从而解决如何分配最大节省费用的问题。

对于排序局势 $\langle N,\sigma_0,\boldsymbol{p},\boldsymbol{\alpha}\rangle$,联盟 S 的加工费用表示为 $c_{\sigma_0}(S)=\sum\limits_{i\in S}\alpha_i C(\sigma_0,i)$,那么大联盟 N 的最大节省费用为 $c_{\sigma_0}(N)-c_{\hat{\sigma}}(N)$。

定义 3.2 如果对于所有的 $j\in N\backslash S$,有

$$P(\sigma_0,j)=P(\sigma,j),$$

则称联盟 S 在排序 σ 中是连续的(continuous)。其中 $P(\sigma,j)$ 表示排序 σ 中在工件 j 之前加工的工件集合。

联盟 S 在排序 σ 中是连续的意味着在排序 σ 中，对于任意 $i,j \in S$ 和 $k \in N$，如果 $\sigma(i) < \sigma(k) < \sigma(j)$，则有 $k \in S$，并且工件 $j \in N \backslash S$ 的开始加工时间等于它在初始排序 σ_0 中的开始加工时间。假设联盟 S 中的工件跳过 S 之外的工件进行交换是不被允许的。

下面的联盟排序博弈的定义是 Curiel 等(1989)首次给出的。

定义 3.3　关于排序局势 $\langle N,\sigma_0,\boldsymbol{p},\boldsymbol{\alpha}\rangle$，对应的联盟排序博弈表示为 $\langle N,v\rangle$，其中

$$v(S)=\max_{\sigma\in\pi(N)}\left\{\sum_{i\in S}\alpha_i(C(\sigma_0,i)-C(\sigma,i))\right\}, \tag{3.1}$$

且满足 $v(\varnothing)=0, S \in 2^N$。

在初始排序 σ_0 中，如果 i 是 j 的紧前工件，令 $g_{ij}=\max\{0,\alpha_j p_i-\alpha_i p_j\}$ 表示交换 i 和 j 的加工顺序后获得的节省费用。对于排序 σ_0 中任意连续的联盟 S，式(3.1)可以用 g_{ij} 表示如下：

$$v(S)=\sum_{i,j\in S:\sigma_0(i)<\sigma_0(j)} g_{ij} \tag{3.2}$$

对于一个在排序 σ_0 中非连续的联盟 T，其特征函数表示为

$$v(T)=\sum_{S\in T\backslash\sigma_0} v(S)。 \tag{3.3}$$

其中 $T\backslash\sigma_0$ 是联盟 T 的所有极大连续子联盟(components)的集合，一个 T 的极大连续子联盟是指在排序 σ 中，如果再增加一个 T 中的工件将不再连续的联盟。

例 3.1　令 $N=\{1,2,3\}$，$\sigma_0(i)=i, i\in N$，$\boldsymbol{p}=(2,2,1)$，$\boldsymbol{\alpha}=(4,6,5)$。计算得 $g_{12}=g_{23}=4, g_{13}=6$。则对于任意 $i\in N$，有 $v(\{i\})=0, v(\{1,2\})=v(\{2,3\})=4, v(\{1,3\})=v(\{1\})+v(\{3\})=0, v(N)=14$。

现在介绍两类与联盟排序博弈密切相关的博弈问题：Tijs 等首次提出的排列博弈(permutation games)(Tijs et al.,1984)和 Curiel 等首次提出的 σ-极大连续联盟可加博弈(σ-component additive games)(Curiel et al.,1993)。

在排列博弈中有 n 个参与人，每个参与人有一个工件和一台机器，每台机器同一时间只能加工一个工件。如果参与人 i 的工件在参与人 j 的机器上加工，那么加工费用是 α_{ij}，并且在参与人之间的补偿性支付是允许的。令 $N=\{1,2,\cdots,n\}$ 表示参与人的集合。对于任意 $S\in 2^N\backslash\{\varnothing\}$，令 $\pi(S)$ 表示 S 中参与人的所有排列集合。

定义 3.4　排列博弈是一个序对 $\langle N,v\rangle$，其中

$$v(S)=\sum_{i\in S}\alpha_{ii}-\min_{\sigma_S\in\pi(S)}\sum_{i\in S}\alpha_{i\sigma_S(i)}, \tag{3.4}$$

这里 $S\in 2^N\backslash\{\varnothing\}$ 且 $v(\varnothing)=0$。

在排列博弈中，初始的排列局势是每个参与人都要加工自己的工件。联盟 S 的最大节省费用 $v(S)$ 可以通过比较 S 中工件的最优排列和初始排列的值获得。Tijs 等(1984)给出了排列博弈的如下结果，对于该结果，Tijs 等(1984)、Curiel 和 Tijs(1986)，以及 Klijn 等(2000)分别给出了证明。

定理 3.1 排列博弈 $\langle N, v\rangle$ 是均衡的。

由定理 2.1 和定理 3.1 知排列博弈 $\langle N, v\rangle$ 的核心是非空的。下面给出 σ-极大连续联盟可加博弈的定义。

定义 3.5 称博弈 $\langle N, v\rangle$ 为 σ-极大连续联盟可加博弈，如果 $\langle N, v\rangle$ 满足以下条件：

(1) $v(\{i\})=0, i \in N$。

(2) $\langle N, v\rangle$ 满足超可加性，即对任意的 $S, T \in 2^N$，如果 $S \cap T=\varnothing$，则有 $v(S \cup T) \geqslant v(S)+v(T)$。

(3)（极大连续子联盟可加性）对于任意的 $T \in 2^N$，有 $v(T)=\sum_{S \in T\backslash\sigma_0} v(S)$，其中 $T\backslash\sigma_0$ 表示联盟 T 的所有极大连续子联盟的集合。

Curiel 等(1994)证明了 σ-极大连续联盟可加博弈有非空的核心。

定理 3.2 σ-极大连续联盟可加博弈 $\langle N, v\rangle$ 是均衡的。

证明： 令

$$\beta_i(v)=\frac{1}{2}[v(P(\sigma, i) \cup\{i\})-v(P(\sigma, i))+v(F(\sigma, i) \cup\{i\})-v(F(\sigma, i))], \tag{3.5}$$

其中，$P(\sigma, i)$ 和 $F(\sigma, i)$ 分别表示在排序 σ 中工件 i 的前继工件集合和后继工件集合。

对于任意连续联盟 $S \in 2^N$，不妨设 $S=\{i \in N \mid a \leqslant \sigma(i) \leqslant b\}$，有

$$\begin{aligned}
2 \sum_{i \in S} \beta_i(v)= & \sum_{i \in S}[v(P(\sigma, i) \cup\{i\})-v(P(\sigma, i))+v(F(\sigma, i) \cup\{i\})- \\
& v(F(\sigma, i))] \\
= & v(P(\sigma, \sigma^{-1}(b)) \cup\{\sigma^{-1}(b)\})-v(P(\sigma, \sigma^{-1}(a)))+ \\
& v(F(\sigma, \sigma^{-1}(a)) \cup\{\sigma^{-1}(a)\})-v(F(\sigma, \sigma^{-1}(b))) \\
\geqslant & 2 v(S)。
\end{aligned}$$

上式中的第二个等号可以通过消除展开后的相等项获得。不等号由 σ-极大连续联盟可加博弈的超可加性以及下面的两式获得：

$$P(\sigma, \sigma^{-1}(a)) \cup S=P(\sigma, \sigma^{-1}(b)) \cup\{\sigma^{-1}(b)\},$$

$$S \cup F(\sigma, \sigma^{-1}(b))=\{\sigma^{-1}(a)\} \cup F(\sigma, \sigma^{-1}(a))。$$

如果 $S=N$，即 $a=1$ 和 $b=n$，则有