

协商支持系统中的模型管理框架

A Model Management Framework for Negotiation Support Systems

王 泊 史佩雯 田启家 史忠植

(中国科学院计算技术研究所智能信息处理重点实验室 北京100080)

Abstract Decision models play an important role in negotiation, and providing support to the management of these models is one of the most important functions of negotiation support systems (NSS). However, in previous research in NSS, little has been discussed about how decision models should be managed. This paper will report an ongoing research on model management in negotiation and propose a model management framework for NSS. Specifically, the paper will discuss the necessity of model management in NSS, compare it with the research in DSS, identify typical roles of model management in negotiation, and present and illustrate the model management framework.

Keywords Automated negotiation, Agent-mediated electronic commerce, Intelligent software agents, Negotiation support systems

1. 引言

协商是商业贸易交往中的重要环节。随着电子商务的快速增长,基于 Internet 的协商支持或自动协商已成为实际的需要。同时也有很多相关的工作从不同的角度研究协商和协商支持^[13]。

从决策制定的角度看,协商是由一系列决策行为构成的。协商的参与者有两方或者更多方。他们之间进行交互以求达成一致,尽可能在最后的一致协议中获得最大的利益。而典型的协商决策行为包括评估对方提出的方案,生成新的方案,冲突解决(为缩小参与者所提方案之间的距离提出的折衷方案)等等。

如何利用计算机系统提高决策的有效性和决策制定的效率,已经成为一个重要的研究课题。作为决策支持系统(DSS)的一种,协商支持系统是支持协商过程的所有计算机系统的统称。协商支持系统包括两部分:

1. 通过计算机来实现协商,包括相应的网络设施(硬件软件)、通信软件、协商协议、协商语言等等。
2. 帮助参与者做出更好的选择,包括对手建模、状态评估、规划、调度、预测等等。

显然,决策模型的管理属于第二类。在协商中数学模型占有重要的地位,尤其是在商业领域。作为决策支持系统,模型管理系统是其中不可或缺的部分。在以前的研究中,决策支持系统中的模型管理受到相当的关注,然而作为决策支持系统的一种,却很少有关于协商中的决策模型管理的研究。

在本文中,我们将讨论协商支持系统中的模型管理问题。特别是,我们将讨论决策模型在协商中的角色,及如何有效地管理。还将提出一种模型管理框架,及其对决策制定过程的支持。下面讨论协商中模型管理的必要性、扮演的角色、和必需的内容;接着提出模型管理框架的结构;然后用一个例子来说明用管理决策模型的过程。最后是结论。

2. 协商支持系统中的模型管理

首先,用一个例子来说明一下多物体协商的场景。假设买方需要买 A 和 B 两种商品。并且为了买到 A 和 B,他必须和卖方进行协商。现在假定, $A_B_Price = ¥470$, $B_B_Price =$

$¥520$;同时估计 $A_S_Price = ¥560$, $B_S_Price = ¥630$ 。在这里, B_Price 是买入的价格, S_Price 是期望的卖出价。假设买方希望购买至少8个商品 A 和最多12个商品 B。

在协商的某些阶段,卖方提出了下面两种折扣方案让买方选择:

方案1 商品 A,如果购买超过7个,可以按2.3%的折扣购买;如果购买超过13个,可以按3.2%的折扣购买。商品 B,如果购买超过5个,可以按2%的折扣购买;如果购买超过10个,可以按3%的折扣购买。

方案2 如果购买总数(A加上B)超过13个,可以按2.35%的折扣计算 A 的价格;如果购买超过20个,可以按3.3%的折扣购买 A。同时,如果购买超过15个,可以按2.25%的折扣购买 B;如果购买超过25个,可以按3.25%的折扣购买 B。

在这种情况下,购买者必须在经过认真计算后,才能做出有利的决定。也不得不借助数学模型来确定购买计划。

通过这个简单的例子可以看出,协商者在协商过程中会经常性地借助数学模型来做决策。如果协商支持系统中有数学模型管理的子系统,那对协商者来说,是非常有用的。

在决策支持系统以往的研究中,虽然没有多少涉及模型管理问题,但是其必要性和重要程度还是很受关注的。例如,在文[6]中,从基于主体(Agent)的电子商务中自动协商的角度看,多属性的决策模型的管理已经被认为是一个重要的研究课题。在一些自动协商系统或者协商决策系统中,决策模型的管理在一定程度上已经被实现。例如,MIT's Kasbah,把三个固定价格曲线在系统中预先定义好,这样所有的主体都可以直接使用。在 Bui 的协商支持系统中协商者集成了一个计算部件。

下面看看协商支持系统中模型管理应该特别重视的环节。考虑到协商中协商者可能遇到的问题,我们不难发现有以下三个问题需要解决。首先,协商者面临在有限时间内做决定的压力。也就是说协商者必须及时应对对手的决策。还有更重要的一点,和普通的决策制定相比,协商者面临的是一个更加复杂的决策环境,同时这样的环境在决策过程中是不断变化的。只有当动态协商状态以及决策模型和决策环境的关联都能够被很好地描述,决策模型才能够被有效地使用。所以如果

只是单纯地提供一组数学公式或算法,而没有任何关于它们所适用的环境的提示,也很难给协商者提供更多有用的帮助。我们认为,虽然协商支持系统中模型管理的最终目标是管理数学公式或算法,但是我们还是需要关注他们的上下文语境。特别需要强调的是,协商管理系统中的模型管理系统应该能够提供使用数学模型的支持。从在概念层分析决策环境,到在环境分析基础上的数学模型抽象,在最后数学模型的算法实现。从以上的考虑,可以总结出以下模型管理系统的主要功能:

1. 通过概念模型构建决策环境;
2. 在概念模型的基础上构建数学模型;
3. 实现模型的基本操作:添加/删除/修改/集成;
4. 帮助实现决策模型的算法;
5. 支持数学模型的基本功能:方案评估、方案生成、冲突解决、仿真等等。

要实现上述功能,需要有一个灵活的表示框架能够把不同抽象层的概念集成在一起。如何提供这样一个合适的框架是模型管理系统的主要研究课题。在决策支持系统(DSS)^[4]的研究文献中有很多描述框架的文章,如关联模型^[2]、逻辑模型^[9]、图文法^[8]、结构化建模^[5]、面向对象的方法^[10~12]。然而这些方法都有两个问题:(1)其中大部分框架仅仅涉及整个建模过程中的数学模型方面。在这些框架里,模型管理系统处理的是被数学公式形式化表述后的决策问题,而没有决策环境的理解和建模。(2)另外一些方法,像结构化建模方法,虽然也提供了概念层的决策模型管理,但是这些和一些通常的概念建模(如 OO 方法)并不完全一样。

本文提出的框架结构克服了上述两个问题,在概念层利用面向对象建模语言描述概念,通过 Petri 网来集成不同层的概念。

3. 面向对象的模型管理框架

框架分为三层:概念层、数学层、算法层。这些层之间的关系如图1所示。

每一层都会获取不同抽象程度的信息,同时在决策过程中承担不同的角色:

·概念层

—这里需要解决以下问题:(1)在协商中需要考虑哪些因素(对象、属性或决策变量)?(2)为了更好地理解要完成的任务,这些因素应该如何组织?

—概念层的目标是生成决策环境的概念模型,并且用面

向对象建模的符号系统来表示,如 UML。

—在协商过程中,概念层帮助协商者构建和理解协商环境,创建正确的数学模型,同时解释决策结果。

·数学层

—该层需要解决的问题是确定概念层提出的属性或变量之间的确切关系。

—该层输出是一组用数学公式标示的约束/关系。

—在协商中,数学层帮助协商者定义或理解属性或变量之间的确切关系。

·算法层

—该层完成数学公式的计算机实现。

—该层得到的是一组可执行的计算机程序,用程序语言或者可执行代码来表述。

—在协商中,这些是和其他协商者交互的程序,帮助协商者完成类似方案评估、方案生成、冲突解决等任务。

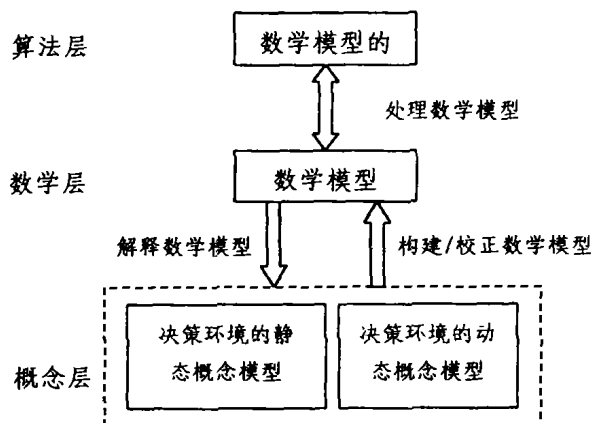


图1 三层决策模型体系结构

从协商者的角度来看,一个合适的表述框架必须要能够用一致的方式描述协商环境和数学模型,因为在协商中必须使用这些描述。而当某些模型会导致冲突的时候,需要能让协商者找到合适的模型。相对而言,算法的表达式对协商者的影响很小,因为算法通常都是可以用一种标准的方式描述,可以在后台运行的。所以在本文中涉及描述部分的时候,我们仅仅在概念层和数学层进行。

在概念层,是用面向对象的建模语言(UML^[3])来描述决策环境。协商中的常量和决策变量被视作对象的属性。数学模型是用来描述对象属性之间关系的,并用数学公式描述这种关系。

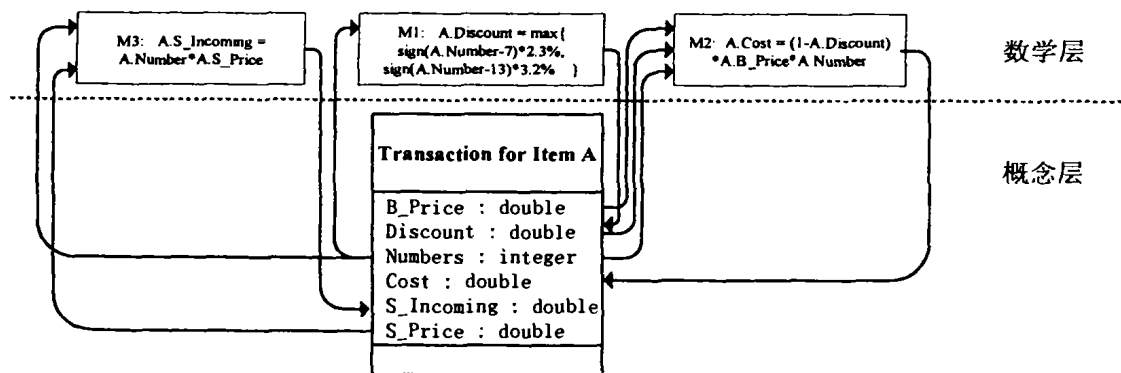


图2 交易事务 A 的协商环境和数学模型

在图2中是一个决策环境和相应的数学模型。这里,买方和卖方就商品 A 进行协商。商品 A 用一个对象——交易事务 A 来建模。事务 A 有以下属性:买入价(B_Price)、折扣(Discount)、购买数量(Number)、期待的卖出价(S_Price)、成本(Cost)和利润(S_Incoming)。其中三个数学模型定义如下:

•M1: A_Discount

$= \max \{ \text{sign}(A_Number - 7) \times 2.3\%, \text{sign}(A_Number - 13) \times 3.2\% \}$

其中: $\text{sign}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$

这就是说,如果购买超过7个但是少于13个,可以按2.3%的折扣购买;如果购买超过13个,可以按3.2%的折扣购买。Sign(x)是符号函数,正数的返回值是1,其他情况为0。

•M2: A_Cost = (1 - A_Discount)

$\times A_B_Price \times A_Number$

这是计算购买成本的公式。

•M3: AS_Incoming = A_Number $\times$ AS_Price

这个公式是解释如果按照期望的价格卖出该商品的收益。

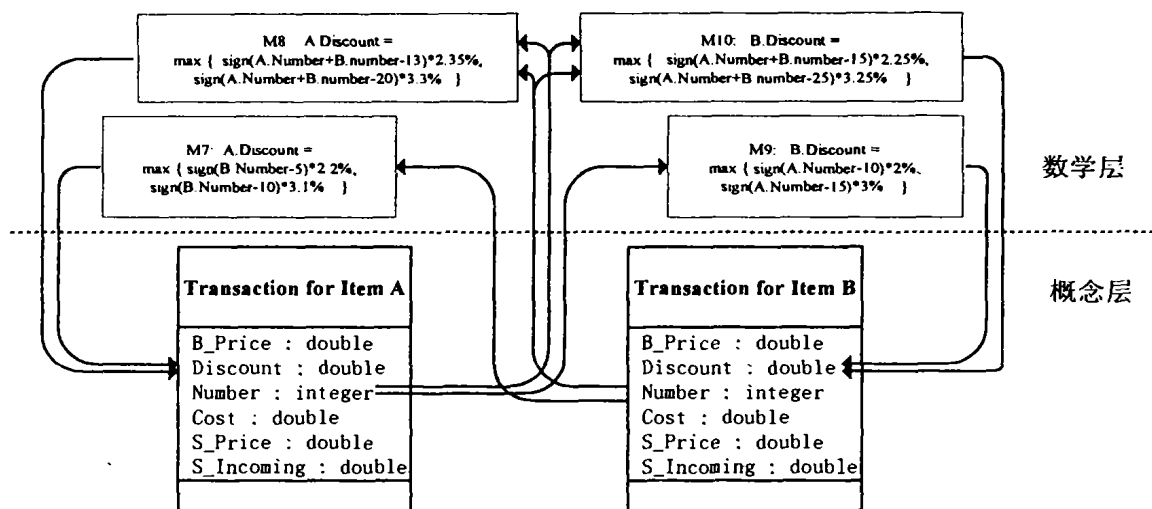


图3 两商品的协商环境和数学模型

当在现实中存在不只一种商品,一个数学模型是可以包括若干个对象的属性(图3)。以M7为例,如果你买了超过5个,但是少于10个商品B,你将获得商品A的2.2%的折扣率,同时如果你买了超过10个的商品B,商品A的折扣率可以降低到3.1%。同样,一组属性之间的关系可以有多种关系,也就是可以用多个数学模型来描述。但是在做决策时,最多只有一个模型可以被采用。对于数学模型的建立没有通用的方法。一些通用的模式可以借鉴。例如,用于创建决策模型的 action-resource 模型^[1]。如何创建 action-resource 在文[14]中有讨论。

用面向对象建模语言和数学模型的方法的优点是直观和容易理解。然而,当对象、属性和数学模型很多的时候,这种表示法将会很复杂。为了应付非常复杂的决策环境,需要引进相应的形式化工具——Petri 网^[7]。

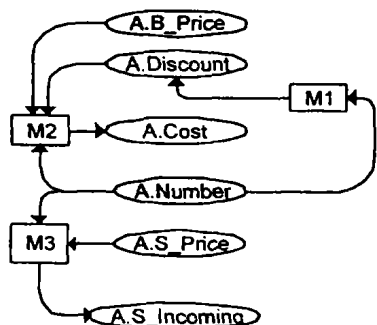


图4 图2的 Petri 网表示

Petri 网里有两个基本概念:位置(用椭圆表示,有些文献中称之为库所)和变迁(用矩形表示)。有些位置含有标记,标

记是通过变迁传递的。标记在网内流动的时候会改变整个网的状态。

用面向对象的表示方法的协商环境和数学模型,可以直接转换成 Petri 网。对象图中的属性用位置来表示,数学模型用变迁来表示。带有标记的位置是那些属性值在决策过程中确定的(图4中的 A_Number),或已分配的(图4中的 A_B_Price, A_S_Price)。图4是图2的 Petri 网的表示,带有标记的位置是灰色的。

在 Petri 网里,可以用标记的流动来模拟决策模型的执行过程。在图4中,可以看出 M1 和 M3 是可用的,因为标记作为 M2 的输入。但是由于位置 A_Discount 没有任何标记,所以 M2 是不可用的。当 M1 和 M3 执行后,标记就会流向 A_Discount 和 A_S_Incoming。也就是说 A_Discount 和 A_S_Incoming 已经被计算出来。现在, A_Discount 有了一个标记,相应地 M2 也可以计算了。当 M2 发生后,标记最后流向 A_Cost——得到了 A_Cost 的值。

和面向对象的方法相比, Petri 网缺乏结构化机制,不能把属性集中到对象中。但是, Petri 网的优势也是显然的:(1) 决策模型的静态特征通过位置和变迁的拓扑结构来表示;(2) 可以通过控制 Petri 网内标记的流向来实现决策模型的动态行为,这样有助于模型的集成。

由于篇幅所限,本文中就没有如何用 Petri 网描述决策模型做详细讨论,这里只是给出了一些简单的描述。

结论 本文提出了一种用于协商的模型管理框架。强调了管理决策模型不仅仅是数学层的问题,同时也需要从概念层考虑。这里从三个不同的抽象层来描述决策模型:概念层、

(下转第116页)

时,对 g_* 的投标将为 $\max(\sum_{(g, \pi) \in E} p(g) + k\lambda_*, \beta + \delta)$, 其中 k 是当前的输入投标, β 是 g_* 的上一次投标。

4.3 同步上升策略的平衡证明

引理2 当同步上升策略达到静态时,每一个消费 Agent 满足 λ - δ 竞争平衡的条件(条件1和2)。

证明:由于每一个消费 Agent,都是以 δ 为基数进行增加,而 δ 为一个非负数,所以显然满足定义10的条件(1)。

用 p 和 (V', E') 表示最后的价格和分配。假设定义10的条件(2)不能满足,那么可以得到:对于所有的 $c \in C$, $\sigma(c, E', p) < H_c(p) - \delta$; 用 g^* 表示消费 Agent 在价格 p 的最大盈余。

如果消费 Agent C 并没有购买任何物品,则可以得到 $\sigma(c, E', p) = 0$ 和 $p(g^*) + \delta > V_c(g^*)$, 又因为 $H_c(p) = V_c(g^*) - P(g^*)$, 上述三个式子通过变化可以得到 $\sigma(c, E', p) > H_c(p) - \delta$, 和前面的假设相矛盾,因此假设 Agent C 购买了物品 g' , 可以得到

$$V_c(g') - P(g') < V_c(g^*) - P(g^*) - \delta \quad (7)$$

用 $\hat{p}(g^*)$ 和 $\hat{p}(g')$ 表示 C 的最后的投标值,由于 C 会对 g' 给出投标值 $\hat{p}(g') + \delta$, 并且最后获得了投标,因此可以得到

$$\hat{p}(g') + \delta \geq p(g') \quad (8)$$

同时由于投标价格不会降低,可以得到

$$\hat{p}(g^*) \leq p(g^*) \quad (9)$$

将(8)和(9)代入(7),可以得到

$$V_c(g') - [\hat{p}(g') + \delta] < V_c(g^*) - [\hat{p}(g^*) + \delta]$$

显而易见, C 在此时愿意选择 g^* , 而不是 g' , 这与假设相矛盾,因此假设条件(2)不满足,不能成立。引理2得证。

引理3 当同步上升策略达到静态时,没有 inactive(非活动)状态的生产 Agent 进行价格为正的购买输入,每一个生产 Agent 满足 λ - δ 竞争平衡的条件(条件1和3)。

证明:根据投标方案,可以看出每一个生产 Agent 显而易见满足条件(1)。并且在静态的时候, π 是可行的。

用 p 和 (V', E') 表示最后的价格和分配。

如果 $H_*(p) > N_*\lambda_* + \delta$, 那么 $H_*(p) = p(g_*) - \sum_{(g, \pi) \in E} p(g) > N_*\lambda_* + \delta$, 可以得到 $p(g_*) = \sum_{(g, \pi) \in E} p(g) + N_*\lambda_* + \delta$ 。由于 π 的投标方案保证 π 的投标 g_* 不会超过 $\sum_{(g, \pi) \in E} p(g) + N_*\lambda_* + \delta$, 因此可以得到投标 g_* 满足 $\sigma(\pi, E', p) = H_*(p)$, 并且条件(3)成立。

如果 $H_*(p) \leq N_*\lambda_* + \delta$, 又由于 $\sigma(\pi, E', p) \geq 0$, 显然条件(3)也成立。这样引理3即得证。

定理2 通过同步上升策略协议,在静态时决定的价格和分配方案在有无产品提供(inactive)的生产 Agent 购买价格为正的输入的情况下,是 λ - δ 竞争平衡的。

证明:由引理可以知道 λ - δ 竞争平衡的条件对于 C 和 Π 是可以满足的。

对于 S , 根据投标方案和价格的选择, S 可以使它的盈余达到最大值,因此显然满足条件(1)和(4)。

同样由于价格的选择,条件(5)也是满足的。

而如果一个 Inactive 的生产 Agent 进行了一个正数价格的输入,则不可能达到 λ - δ 竞争平衡。定理2即得证。

结论和展望 本文针对供求关系的形成情况,在对问题进行描述,和对相关概念进行说明的基础上提出了一种策略,并进行了有效性证明。从理论上而言,这种策略是有效的,但从实际情况而言,它仍然存在许多的问题和不足,特别是在对非活动的 Agent 需要输入时,则可能出现较差的结果,我们将在后续的文章中继续进行讨论。

参考文献

- 1 Wermelinger M. Connectors for Mobile Agents. IEEE Transaction on Software Engineering, 1998, 24(5): 331~341
- 2 Pham V A, Darmouch A. Mobile Software Agents: An Overview. IEEE Communications Magazine, 1998(July): 92~102
- 3 Lange D, Oshima M. Seven good reasons for mobile agents. Communication of the ACM, 1999, 42(3)
- 4 Parkes D C. iBundle: An efficient ascending price bundle auction. In: ACM Conf. on Electronic Commerce, 1999. 148~157
- 5 Andersson A, Tenhunen M, Ygge F. Integer programming for combinatorial auction winner determination. In: Fourth Intl. Conf. on Multi-Agent Systems, 2000. 39~46
- 6 Rassenti S J, Smith V L, Bulfin R L. A combinatorial auction mechanism for airport time slot allocation. Bell Journal of Economics, 1982, 13: 402~417
- 7 Rothkopf M H, Pekec A, Harstad R M. Computationally manageable combinatorial auctions Management Science, 1998, 44: 1131~1147
- 8 Walsh W E, Wellman M P. A market protocol for decentralized task allocation. In: Third Intl. Conf. on Multi-Agent Systems, 1998. 325~332
- 9 Walsh W E, Wellman M P. Efficiency and equilibrium in task allocation economies with hierarchical dependencies. In: Sixteenth Intl. Joint Conf. on Artificial Intelligence, 1999. 520~526
- 10 Paris, France, July. 1998. 3~8
- 7 Jensen K. Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag (2nd) 1997
- 8 Jones C V, An Integrated Modeling Environment Based-on Attributed Graphs and Graph-grammars. Decision Support Systems. 1993, 10: 255~275
- 9 Kimbrough S O, Lee R M. Logic Modeling: a Tool for Management Science. Decision Support Systems, 1988, 4: 3~16
- 10 Lenard M L, An Object-Oriented Approach to Model Management. Decision Support Systems, 1993, 9: 67~73
- 11 Ma J. An Object-Oriented Framework for Model Management. Decision Support Systems, 1993, 13: 133~139
- 12 Muhanna W A. An Object-Oriented Framework for Model Management and DSS Development. Decision Support Systems, 1993, 9: 217~229
- 13 Müller H J, Principles N. In: Foundations of Distributed Artificial Intelligence (G. M. P. O'Hare and N. R. Jennings, eds.), JOHN WILEY & SONS, INC, 1996
- 14 Tian Qijia, Ma Jian, Shi Zhongzhi. Integrating Object-Oriented Analysis with Action Logic for Decision Model Building. IEEE SMC'2000, Nashville, TN, USA, Oct. 2000

(上接第109页)

数学层和算法层。在描述方面,我们采用了面向对象的方法和 Petri 网来集成决策模型的不同部分。

参考文献

- 1 Bingasioglu M, Process-Based Reconstructive Approach to Model Building. Decision Support Systems, 1994, 12: 97~113-113
- 2 Blanning R W, A Relational Theory of Model Management. In: Decision Support Systems: Theory and Applications (eds. C. W. Holsapple and A. B. Whinston), Springer, 1987
- 3 D'Souza D, Wills A. Objects, Components, and Frameworks with UML: The Catalysis Approach. Addison Wesley, 1999
- 4 Eom S B. Relationships between the decision support system sub-specialties and reference disciplines: An empirical investigation. European Journal of Operational Research, 1998, 104: 31~45
- 5 Geoffrion A M. An Introduction to Structured Modeling, Management Science, 1987, 33(5): 547~588
- 6 Guttman R H, Maes P. Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce. In: Proc. of the Second Intl. Workshop on Cooperative Information Agents (CIA'98),