

决策支持系统中的建模知识表示研究^①

黄梯云,冯玉强,周宽久

(哈尔滨工业大学管理学院, 哈尔滨 150001)

摘要:提出一种基于知识的建模支持系统,该系统的知识由三部分组成:构模规则知识、变量类型知识以及数据映射知识。构模规则知识描述了决策者的构模过程;变量类型知识定义了构模规则知识中所有变量的数据类型;数据映射知识描述求解过程所需数据的来源。对于决策者输入的一个实际决策问题,系统能够根据问题描述按照构模知识自动进行求解,同时对经验进行学习,生成经验规则。

关键词:建模支持系统;模型管理系统;面向对象;模型类;机器学习

中图分类号:TP312

文献标识码:A

文章编号:1007-9807(2001)01-0045-07

0 引言

客观问题的复杂性要求决策支持系统模型库中存放大量的各种各样的模型。如果模型管理系统能够根据实际问题的需要,通过与决策者交互,自动构造模型并求解,即提供建模支持功能,无疑会减轻决策者的构模负担,这对提高决策效率起到至关重要的作用。关于自动构模,国内外许多学者做了大量有益的工作,目前研究主要集中在基于逻辑的和基于事件推理的两个方面^[1~4]。本文提出基于知识的模型自动生成,其特点是知识中的变量可以是向量或矩阵类型,并可在模型类的基础之上根据知识库中的模型操纵知识自动生成

模型并求解。

1 构模过程的图形表示

将问题求解看成是与或图的生成与搜索过程,那么构模就是图的生成与在图中选择路径的过程。对于与节点,只有其所有子节点全为可解,它才可解;对于或节点,只要有一个子节点为可解,它就可解。求解路径应该是一个有向图。

几个定义:

模型节点:模型节点对应模型类的一个对象。它的内部由输入/输出端口及连接它们的方法组成。图1是一个预测模型,其内部由两个节点和四条边组成的有向图。

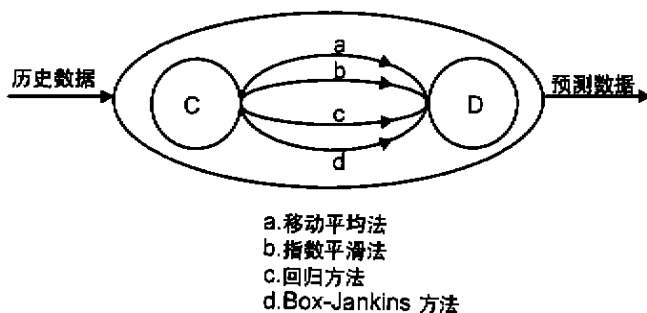


图1 具有四条边的模型节点

① 收稿日期:1999-10-06;修订日期:2000-04-26。

基金项目:国家自然科学基金资助项目(79670023)。

作者简介:黄梯云(1932-),男,上海人,教授,博士生导师。

中间节点:由其它节点的耦合而得到的节点,中间节点可以分解为一系列子节点.即决策问题可以分解成一系列的子问题(图2).

节点:叶节点是从数据库中获取数据或用户提供数据的节点,它不能继续分解.

与节点:节点N分解成一系列子问题: n_1, n_2, \dots, n_m ,当这些子问题全部可解,N称为节点.

或节点:节点N分解成一系列子问题: n_1, n_2, \dots, n_m ,当这些子问题中有一个为可解,N称为或节点.

构模过程可以由上列五种节点表示.

2 问题的提出

这里通过一个实际决策问题说明构模过程的图形表示及图形与规则表示之间的转换.

某工厂要安排下一年生产A、B、C三种产品,假设生产单位产品所需的资源消耗、资源总数、三种产品的价格、产品C的过去生产量分别存于资源消耗数据库 cons. dbf、资源数据库 totres. dbf、单位利润数据库 income. dbf 和历史产量数据库 pastd. dbf 中(参表1、2、3、4).

表1 cons. dbf

product	Resource	consum
A	设备台时(H)	30
B	设备台时(H)	80
C	设备台时(H)	50
A	材料甲(T)	4
B	材料甲(T)	5
C	材料甲(T)	2
A	材料乙(T)	3
B	材料乙(T)	2
C	材料乙(T)	2

表2 totres. dbf

source	Totam
设备台时(H)	15 000
材料甲(T)	500
材料乙(T)	300

表3 income. dbf

Product	inc(万元)
A	2
B	3
C	2.5

表4 pastd. dbf

yearlist	productc (t)
87	30
88	40
89	50
90	60
91	70
92	80
93	90
94	100
95	100

如何安排下年的生产,才能使工厂获利最多?

针对这一问题,运用运筹学可以有两种方法解决:一是利用表1至表3的数据直接利用线性规划来求解.

第2种方法是先用预测模型预测出1996年产品C的产量 X_2 ,并计算获利 Z_2 (有关变量含义参见表5),然后再利用线性规划模型计算产品A、B的产量 X_1, X_2 并计算获利 Z_1 ,最后的解是 $Z = Z_1 + Z_2, X = (X_1, X_2, X_3)^T$.

第2种比第1种方案计算的结果更可靠,因为第2种方法充分地利用了历史数据.上述是决策者在求解这类实际问题时的思维过程,计算机的自动建模过程就是有效地模拟人的求解过程,图2是该问题的求解图,它是一个与或图,并且节点之间存在顺序,对于一个根节点下的所有子节点,先计算左节点后计算右节点,因为有时右节点的计算需要左节点的值.

这样一来,对于决策者提出的决策问题的求解就转变成图的遍历过程.图的遍历首先要解决的是表示问题,表示方法直接影响系统的求解.

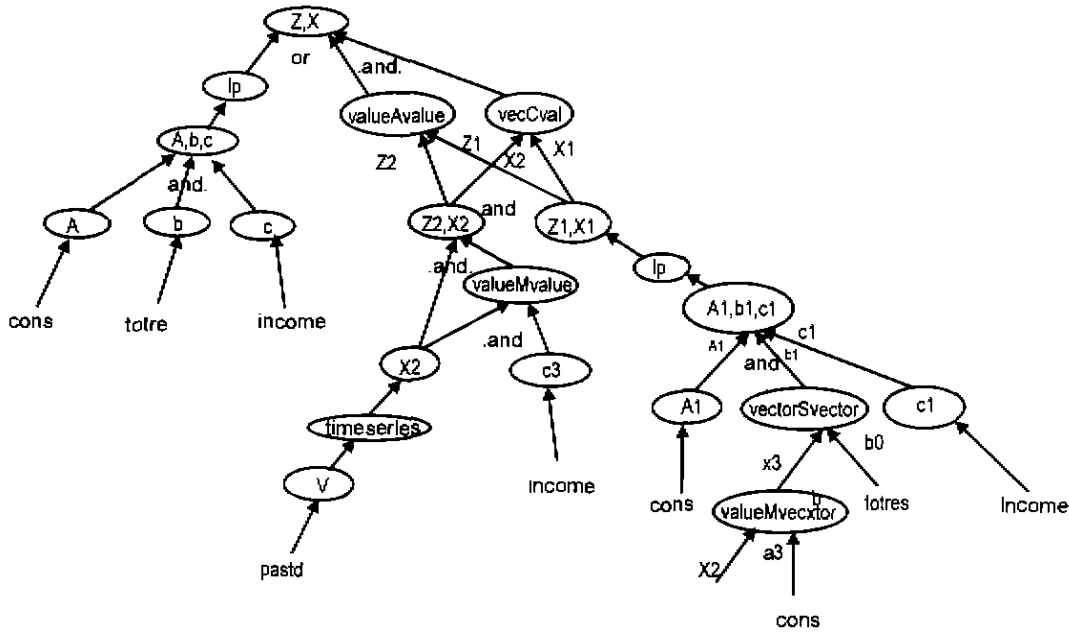


图 2 决策过程与或图

3 知识表示形式

建模支持系统中的构模知识应该由三部分组成:规则知识、变量说明和变量映射.规则知识表示问题的求解过程,变量说明表示变量类型及实际意义,变量映射表示叶节点对应变量如何从数据库中取值.

3.1 规则知识

上面问题求解过程可表示成下面的 9 条规则,这里的规则类似于 Prolog 子句,采用 Horn 子句的形式,每一个子句由结论和前提组成,只有所有前提都可解,结论才可解.变量不仅可以是简单数据类型也可以是复杂数据类型,可以取数值、字符简单类型,也可以取矩阵、向量复杂类型.与或图的中间节点、叶节点、计算节点分别用 node、leaf 和模型类名表示.对于 node 型节点,它对应的变量是由其子节点计算得到,例如规则一中的 Z、X 之值由 node(A0, b0, c0), lp([A0, b0, c0], [Z, X], []) 两个节点计算得到.对于 leaf 节点,其对应变量值从数据库提取.对于模型节点,调用相应模型,下面的 9 条规则中,lp、vecCcal、timeseries、valueMvalue、valueMvector、vectorSvector 为模型名,在节点 lp([A0, b0, c0], [Z, X], []) 中, [A0, b0, c0] 表示节点的输入端口,即它的值被传到 lp 对象的输入端口, [Z, X] 表示节点的输出端

口,lp 对象处理完消息之后,将 lp 对象的输出端口之值传到 [Z, X] 中,lp 节点的第三项 [] 为空表示采用默认方法,对于 lp 采用 simplex 方法,在实际求解时,默认方法在模型库中查找,如果不使用默认方法,就在括号中填上相应的方法名称.

- rule1: node(Z, X); -node(A0, b0, c0), lp([A0, b0, c0], [Z, X], []).
- rule2: node(Z, X); -node(Z2, X2), node(Z1, X1), valueAvalue([Z1, Z2], [Z], []), vecCval([X1, X2], [X], []).
- rule3: node(A0, b0, c0); -leaf(A0), leaf(b0), leaf(c0).
- rule4: node(Z2, X2); -node(X2), node(Z2).
- rule5: node(X2); -leaf(V), timeseries([V], [X2], []).
- rule6: node(Z2); -node(X2), leaf(c3), valueMvalue([C3, X2], [Z2], []).
- rule7: node(Z1, X1); -node(A1, b1, c1), lp([A1, b1, c1], [Z1, X1], []).
- rule8: node(A1, b1, c1); -leaf(A1), node(b1),

```

leaf(c1),
rule9: node(b1);-node(X2),
leaf(b0),
leaf(a3),
valueMvector([X2,a3],[X3],[ ]),
vectorSvector([b0,X3],[b1],[ ]).

```

规则用BNF形式表示如下:

```

<规则> ::= <结论>; - <前提>
<结论> ::= <node(变量表)>
<变量表> ::= <变量> | <变量>, <变量表>
<变量> ::= <字母> <字符串> | <字母>
<字符串> ::= <字符> <字符串> | <数字> <字符
            串> | <字符> | <数字>
<字母> ::= A|B| ...Z|a|b| ...|z
<数字> ::= 0|1|2| ...|9
<前提> ::= <谓词表>

```

```

<谓词表> ::= <node(变量表)|leaf(变量表)|
模型类名([变量表],[变量表],[字符串])|node
(变量表), <谓词表> | leaf(变量表), <谓词表>

```

3.2 变量类型说明

表5是上述规则的变量说明,变量说明知识定义了规则中用到的所有变量的数据类型及意义.这里的数据类型是指基本数据类型类,变量在求解时就成为数据类型类.类的实例即对象,它继承相应类的属性和操作,变量之间的数据传送是调用其相应的操作实现的.面向对象的多态性为实现变量之间的数据传送提供了极大的方便,在基本数据类中均定义了setdatadb,getdatadb操作用于完成对变量对象的数据设置和取值.它不仅适用于简单数据类型而且适用于复杂的象向量、矩阵等数据类型.假设A、B是nummatrixtype对象,C、D是numvectortype对象,则指令B.setdatadb(A.getdatadb())与D.setdatadb(C.getdatadb())可实现A→B和C→D的数据传送.

因此,向不同类型的对象发送相同的消息可以执行不同的操作.

3.3 数据映射知识

数据映射知识能实现leaf节点变量与数据库的连接.引入SML^[5-7]语言的索引与依赖机制

实现模型与数据库的连接;entityname、type、entity、tablename、field、dfieldset、constraint分别表示实体名称、实体类型、依赖实体名称、数据库名、字段名、依赖字段名和约束条件.在这里省去了SML语言中的复合类型实体c,而用依赖两个实体来代替.pe和a的含义表示其对应实体为基本实体和属性实体.res为pe型实体,它对应数据库totres中的resource字段,b0为a型实体,依赖于res,对应于数据库totres中的resource字段,但取值依赖于resource字段,这样res与b0就形成了依赖与对应关系.表中各实体的对应关系如图3所示.

表5 变量类型定义知识

Name	type	comment
Z	numvaluetype	工厂的总收入
X	numvectortype	A,B,C三种产品的产量
A	nummatrixtype	A,B,C三种产品消耗的各种资源
B	numvectortype	生产A,B,C的总资源
C	numvectortype	A,B,C产品单价
Z2	numvaluetype	A,B收入
X2	numvaluetype	C产品的预测产量
X1	numvectortype	A,B产品的产量
V	numvectortype	产品C的过去产量
A1	nummatrixtype	A,B各种资源消耗
b1	numvectortype	生产A,B产品的总资源
c1	numvectortype	A,B产品单价
X3	numvectortype	C产品的预测资源消耗
a3	numvectortype	产品C的各种资源消耗
C3	numvaluetype	产品C的单价
Res	labelvectortype	资源名表t
Prod	labelvectortype	产品名表
prod1	labelvectortype	产品名表
Prodc	labelvaluetype	产品名称

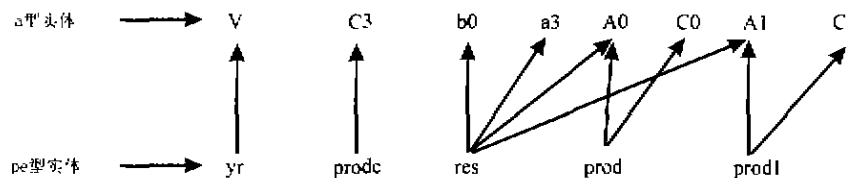


图 3 实体之间的对应和依赖关系

因此如果 yr、prodc、res、prod、prodl 的结构和值一旦确定下来以后，V、C3、b0、a3、A0、c0、A1、C1 的结构和值就确定下来了。

表 6 定义了 leaf 型节点的变量及其对应基本实体与数据库之间的映射，下列例子说明了实体之间的依赖和对应关系。

res 是 pe 型实体，从库 totres 的 resource 字段取值，

res = [设备台时(H)，材料甲(T)，材料乙(T)]

b0 是依赖于 res 的 a 型实体，从库 totres 的 totam 字段取值，依赖字段为 resource，

b0 = [80, 50, 30]

res 与 b0 之间存在一一对应关系：

res = [设备台时(H)，材料甲(T)，材料乙(T)]

b0 = [80, 50, 30]

prod 是 pe 型实体，它从库 income 的字段

product 取值，

prod = [A, B, C]

c0 是依赖于 prod 的 a 型实体，从库 income 的 inc 字段取值，依赖字段是 product，

c0 = [2, 3, 2.5]

prod 与 c0 之间存在一一对应关系：

prod = [A, B, C]
 ↓ ↓ ↓
c0 = [2, 3, 2.5]

A0 是依赖于 res 和 prod 的 a 型实体，res 和 prod 的值决定了 A0 之值，它从库 cons 的 consum 字段取值，依赖字段是 product 和 resource。

A0 与 res 和 prod 的对应关系如下：

Drod = [A, B, C]
 ↓ ↓ ↓
A0 = [2 3 3] ← [设备台时(H)]
 [4 5 2] ← [材料甲(T)] = res
 [3 2 2] ← [材料乙(T)]

表 6 数据映射知识

Entityname	type	Dentity	tablename	field	Dfieldset	constraint
Res	pe		totres	resource		
B0	a	Res	totres	totam	Resource	
Prod	pe		income	product		
C0	a	Prod	income	inc	Product	
A0	a	(res, prod)	cons	consum	(product, resource)	
Prod1	pe		income	product		product! = 'C'
C1	a	prod1	income	inc	Product	
A1	a	(res, prod1)	cons	consum	(product, resource)	
A3	a	Res	cons	consum	Resource	product = 'C'
Prodc	pe		income	product		product = 'C'
Yr	pe		pastd	yearlist		
v	a	Yr	pastd	product	Yearlist	
C3	a	Prodc	income	inc	Product	

系统提供数据映射知识的增加、删除、修改、打印等功能。

4 问题求解及模型操纵知识的学习

对于决策者提出的问题,系统遍历以该问题为根节点的有向图,根据映射知识,用数据库中的数据对叶节点赋值,然后逐层计算,最后得出根节

点的值,如图4所示。

问题求解算法的空间复杂性和时间复杂性很高,因为问题节点 $node(Z,X)$ 的子问题包括 $node$ 节点,求解算法要查找和选择 $node$ 节点对应的子问题,如果去掉子问题中的 $node$ 节点,子问题转换成由叶节点和模型节点组成,那么问题求解的速度将会提高。

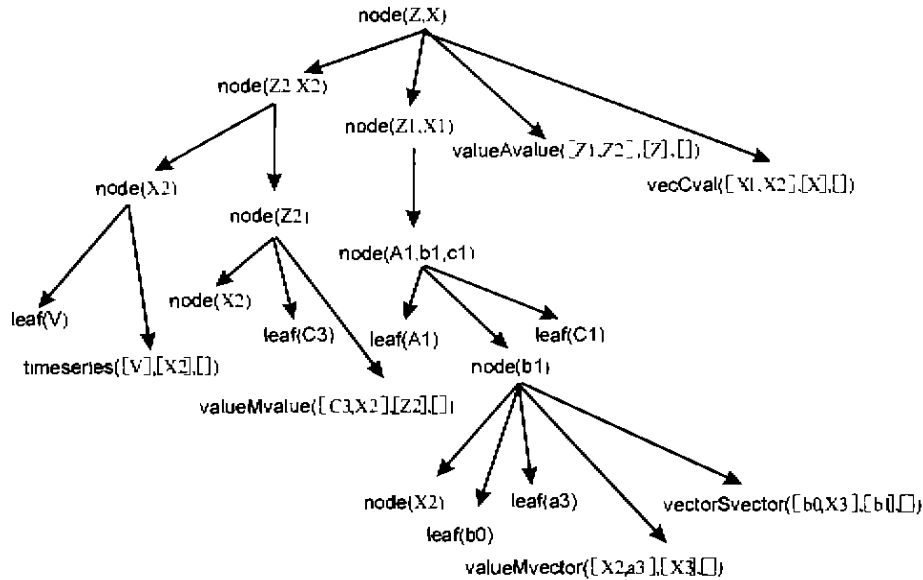


图4 $node(Z,X)$ 的求解路径

求解算法通过求解 $node(Z,X)$ 获得如下规则,从中可以看出该规则的子问题不含有 $node$ 型节点,实际运行表明使用经验规则进行问题求解,系统运行速度得到很大提高.算法的空间复杂性和时间复杂性大大降低.下面这条规则是在求解 $node(Z,X)$ 时生成的一条规则,下一次在求解同样的问题时,可以使用该条规则求解.

```
node(Z,X);leaf(A1),
leaf(V0),
timeseries([V0],[X2],[ ]),
leaf(b0),
leaf(a3),
valueMvector([X2,a3],[X3],[ ]),
vectorSvector([b0,X3],[b],[ ]),
leaf(c1),
lp([A1,b1,c1],[Z1,X1],[ ]),
```

```
leaf(C3),
valueMvalue([C3,X2],[Z2],[ ]),
valueAvalue([Z1,Z2],[Z],[ ]),
vecCval([X1,X2],[X],[ ]).
```

5 结论

本文提出基于模型类的建模支持系统的知识表示.建模支持系统支持决策者根据实际决策问题的需要输入建模知识,进行求解.建模知识由规则知识、变量类型说明知识及数据映射知识三部分组成,规则知识由中间结点、叶结点和模型节点耦合而成,模型节点对应模型库中的模型类,模型类可为基本模型类也可为复合模型类,变量映射指明规则中变量与数据库的连接,支持变量与数据库的独立.规则中的变量是基本数据类的对象,

它是数据与操作的集成,变量之间的数据传送由其操作实现.知识中的变量可以是向量或者是矩阵类型,通过数据库映射实现模型与数据库的连接,这样能够实现模型与数据的独立性,避免归结

反演推理方法造成组合爆炸的缺点,并可引入机器学习功能,当第二次处理同样类型的问题时,因省去了规则的搜索时间,运行速度可得到明显提高.

参 考 文 献:

- [1] Krishnan R. A logic modeling language for automated model construction[J]. *Decision Support Systems*, 1990, 6: 123-152
- [2] Liang Ting-Peng. Analogical reasoning and case-based learning in model management system [J]. *Decision Support Systems*, 1993, 10: 137-160
- [3] McGovern J. Theory and methodology, using case based reasoning for basis development in intelligent decision systems [J]. *European Journal of Operation Research*, 1994, 74: 40-59
- [4] Bonczek B. Representation modeling knowledge with first order predicate calculus[J]. *Operations Research*, 1987
- [5] Groffron A M. The SML language for structured modeling: level 1 and 2[J]. *Operations Research*, 1992, 40(1): 38-57
- [6] Groffron A M. The SML language for structured modeling: level 3 and 4[J]. *Operations Research*, 1992, 40(1): 58-75
- [7] Groffron A M. An introduction to structured modeling[J]. *Management Science*, 1987, 33(5): 547-588
- [8] 韩世欣,黄梯云,李一军. 基于机器学习理论的智能决策支持系统模型操纵方法的研究[J]. *决策与支持系统*, 1996, 6(1): 10-18
- [9] 王孝通,杨德礼,邓贵仕. 模型管理核心问题研究[J]. *决策与决策支持系统*, 1996, 6(2): 15-24
- [10] 陈文伟,黄金才,陈元. 决策支持系统新式结构体系[J]. *决策与决策支持系统*, 1998, 1(3): 54-60
- [11] Dolk D R. An Introduction to model integration and integrated modeling environment[J]. *Decision Support Systems*, 1993, 10: 249-254
- [12] Green B. Rule-based intelligence to support linear programming analysis[J]. *Decision Support Systems*, 1993, 9: 425-447

Study on DSS modeling knowledge representation

HUANG Ti-yun, FENG Yu-qiang, ZHOU Kuan-jiu

Management College, Harbin Institute of Technology, Harbin 150001, China

Abstract: A knowledge-based modeling support system is issued in this paper. Its knowledge consists of three parts: modeling rule knowledge, variable type knowledge and data mapping knowledge. The modeling rule knowledge represents the decision-maker's modeling process. The variable type knowledge defines data types for all variables in the modeling rule knowledge. The data mapping knowledge defines sources of the data required during solving. After the decision-maker inputs a problem, the modeling support system will solve it according to the modeling knowledge automatically and at last commit the result to the decision-maker. Meanwhile, machine learning is adopted to acquire knowledge during problem solving.

Key words: modeling support system; model management system; object-oriented; model class; machine learning