

# 一种双向求解平行结构类问题的方法\*

胡蓬 石纯一 王克宏

(清华大学计算机系, 北京 100084)

**摘要** 平行结构类问题是一类适于分布式求解的人工智能问题, 已有的大多数求解方法均采用预期或目标来指导自底向上的问题求解, 但这些预期或目标是以局部问题求解状态为基础的, 指导性较弱. 尽管有的方法(如改进的 DVMT 结构)允许高层了解, 但未给出明确的求解算法. 本文提出一种双向求解平行结构类问题的方法, 首先根据全局问题求解状态生成预期, 指导自底向上的求解, 然后根据新产生的假设来验证和修改预期, 并用新的预期重新指导求解. 该方法不仅提高了预期的指导性, 而且使问题求解更为灵活.

**关键词** 分布式问题求解, 预期, 验证.

平行结构类问题是人工智能应用问题, 适于用分布式方法求解. 这类问题包括语音理解、车辆监控、运输调度等. 对这些问题求解方法的研究有重要的理论意义与实用价值<sup>[1-3]</sup>.

70年代末, 美国卡内基·梅隆大学在 Hearsay-Ⅱ 系统中, 首次提出黑板结构, 用机缘调度策略调用知识源, 进行语音理解<sup>[4]</sup>. 该方法是数据指导的, 即根据问题内在的多层结构, 调用与领域有关的知识源, 对初始数据(语音信号)和中间数据(称为假设, 如音节、单词等)自底向上地进行解释, 直至得出满足约束(如单词间的语言相邻约束和时间相邻约束)与度量条件(置信度最大)的最高层假设<sup>[2,3]</sup>. 在 Hearsay-Ⅱ 系统中, 也采用了自顶向下的指导, 例如, 用目标假设控制单词与词序假设的数目, 对词组假设前后相邻的单词进行预期与验证, 组成新的词组假设等. 但是, 这些预期仅根据局部假设得到, 适用面较窄(仅限于控制产生的单词与词序数目、组成新词组). 所以, Hearsay-Ⅱ 系统中的大部分问题求解过程是被动的, 当知识源条件不能满足(即触发知识源条件部分的假设未产生)时, 只能消极等待, 即使该知识源非常重要, 也无法调用其它知识源去产生激活该知识源所需要的假设.

80年代初, 美国麻省大学以车辆监控为背景, 对 Hearsay-Ⅱ 的黑板结构做了改进, 提出将目标指导与数据指导相结合的结构——DVMT<sup>[5]</sup>. 当在黑板上产生新的假设时, 并不马上形成知识源示例(待执行的知识源), 而是根据此假设生成相应目标, 然后通过目标处理与目标划分, 激活有关的动作序列实现重要的目标. 这一改进在一定的程度上弥补了 Hearsay

\* 本文 1993-03-11 收到, 1994-02-02 定稿

作者胡蓬, 1963年生, 副教授, 主要研究领域为分布式人工智能, 计算机体系结构, 知识工程. 石纯一, 1935年生, 教授, 主要研究领域为人工智能应用基础和知识工程. 王克宏, 1941年生, 副教授, 主要研究领域为知识工程, 分布式人工智能.

本文通讯联系人: 石纯一, 北京 100084, 清华大学计算机系

— I 方法中被动的问题求解的不足.

然而,Hearsay-I 系统及 80 年代的 DVMT 结构中的自顶向下指导都存在一个共同缺陷,这就是他们的目标都是根据局部假设得到的,没有利用对当前问题求解状态的宏观了解来识别潜在的解及其关系.80 年代中,美国麻省大学曾对 DVMT 结构做了进一步改进,允许基于黑板的问题求解器形成对问题求解状态的高层了解,并依此大致上预期产生潜在部分分解的重要性及代价,指导自底向上求解.但他们强调实验研究,未给出明确的算法<sup>[6]</sup>.

本文提出一种双向求解平行结构类问题的算法,根据当前问题求解状态形成自顶向下预期,指导自底向上求解,同时又用新产生的假设验证预期,对预期进行修改与扩充,继续指导求解.下一节给出平行结构类问题的描述,第 2 节详细介绍双向求解算法,第 3 节对该方法进行讨论,最后给出若干结论.

### 1 平行结构类问题

分布式问题求解系统在物理上或逻辑上分布的领域中有着广泛应用,平行结构类问题是一类重要的应用问题<sup>[3]</sup>.

平行结构类问题包括语音理解、车辆监控、运输调度、篇章理解、雷达监测、自动地震监测、空中交通控制、战场指挥和声纳解释等.这类问题有 3 个特性:

#### 1. 属于解释型问题

这类问题均可按其内在的多层结构,对初始数据自底向上地进行解释求解.

#### 2. 初始数据与中间数据是分布的

初始数据分布在某一区域内.例如车辆监控问题中的传感器信号、运输调度问题中的运输业务均在地理上分布于某一区域内.

中间数据是分布的.如车辆监控的中间结果(信号组、车辆假设等)都是地理上分布的.

#### 3. 具有平行结构

如果将初始数据所分布的区域划分为若干子区域,则对子区域内数据进行解释的子问题之间具有平行结构,即所有子问题均为解释型问题,且子问题之间无主从依赖关系.

下面,给出二维的平行结构类问题(初始数据分布在二维区域内)的描述<sup>[3]</sup>:

已知:(1) 初始数据

在二维的直角坐标系中,存在区域  $A$ (不失一般性,设为矩形)和向量集合  $E = \{e_1, e_2, \dots, e_n\}$ ,其中向量  $e_i = (x_i, y_i, ca_i^1, ca_i^2, \dots, ca_i^r, cs_i^1, cs_i^2, \dots, cs_i^s)$  ( $i = 1, 2, \dots, n$ )分布在  $A$  中, $x_i, y_i$  分别为  $e_i$  的横、纵坐标, $ca_i^1, ca_i^2, \dots, ca_i^r$  是  $e_i$  的公共属性, $cs_i^1, cs_i^2, \dots, cs_i^s$  为  $e_i$  的特殊属性.

#### (2) 多层结构

在集合  $E$  上,定义一个多层结构  $S = \{S_0, S_1, \dots, S_w\}$ ,其中  $S_i$  是第  $i$  层假设的集合, $S_i = \{S_i^1, S_i^2, \dots, S_i^j\}$ , $S_i^j$  为第  $i$  层的第  $j$  个假设,它由第  $i-1$  层或(和)低于第  $i-1$  层的假设组成. $S_0 = E$ .

#### (3) 约束

多层结构中,组成某假设的各假设之间存在公共属性约束.

#### (4) 度量

第  $i$  层的第  $j$  个假设  $S_j^i$  的度量值为  $M(S_j^i)$ .

求解:在集合  $E$  上,按照多层结构  $S$  的定义,求出满足公共属性约束条件、度量函数值最大的最高层假设(第  $w$  层).

表 1 说明了 2 个具体的平行结构类问题.

表 1 平行结构类问题示例

问题	初始数据			多层结构 $S$	约束	度量函数 $M$
	向量集合 $E$	公共属性	区域 $A$			
运输调度	运输业务	货物类型	二维市区	基本组合圈 ↓ 基本回路 ↑ 业务	1. 组成基本回路的业务的货物类型应相容 2. 组成基本组合圈的基本回路及业务的货物类型应相容	里程利用率
车场监控	传感信号	信号频率	二维地理区域	车队 车辆 信号组 信号	组成某假设的低层假设的信号频率应谐波相关	置信度

## 2 双向求解算法

### 2.1 基本算法

双向求解算法模型由自顶向下结点( $TA$ : Top-down Agent)和自底向上结点( $BA$ : Bottom-up Agent)组成.  $TA$  负责根据当前问题求解状态形成自顶向下预期集  $P_i$ , 指导  $BA$  的求解; 而  $BA$  在  $P_i$  的指导下, 自底向上地进行求解, 产生假设集  $H_{i+1}$ , 送至  $TA$ , 由其验证以前的预期集, 并通过修改与扩充, 形成新的预期集, 继续指导  $BA$  求解. 重复上述过程, 直至得到满足条件的最高层假设(图 1).

双向求解的基本算法如下:

- |   |  |
|---|--|
| <p>(1) <math>i \leftarrow 0</math></p> <p>(2) 产生预期集 <math>P_i</math>;</p> <p>(3) 发送 <math>P_i</math>; <math>P_i</math></p> <p style="text-align: center;">→</p>       | <p><math>TA</math>                      <math>BA</math></p>  |
| <p>(1) 接收 <math>P_i</math>;</p> <p>(2) 在 <math>P_i</math> 指导下产生假设集 <math>H_{i+1}</math>;</p> <p>(3) 发送 <math>H_{i+1}</math>;</p> <p style="text-align: center;">←</p> | <p>(4) 接收 <math>H_{i+1}</math>;</p> <p>(5) 根据 <math>H_{i+1}</math> 验证 <math>P_i</math>;</p> <p>如一致, 则</p> <p><math>i \leftarrow i + 1</math>, 若 <math>i \neq w</math>, 转(2);</p> <p>否则修改 <math>P_i</math>, 转(3);</p> <p>(6) 从 <math>H_w</math> 中取度量值最大者为最终解.</p> |

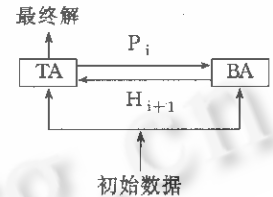


图 1 双向求解算法模型

图 1 的算法模型中,  $TA$  和  $BA$  均为逻辑结点, 由于  $TA$  与  $BA$  之间交换的预期集  $P_i$  和假设集  $H_i$  数量很大, 实际上,  $TA$  与  $BA$  宜放在同一物理结点上实现, 通过共享内存(如黑

板)来完成  $TA$  与  $BA$  之间的通信.

## 2.2 TA 的问题求解

### 2.2.1 预期的定义

$TA$  形成预期的目的是为了指导  $BA$  更有效地进行求解.  $BA$  逐层自底向上地进行问题求解, 第  $i+1$  层假设是调用有关的知识源根据第  $i$  层或低层假设组成的. 因此, 在产生第  $i+1$  层假设之前,  $TA$  应根据已知的第  $i$  层及低层假设形成预期集  $P_i$ .

#### 定义 2.1(预期集).

(1) 一个预期集是一棵树;

(2) 树中每个结点是一个预期, 由三元组  $\{C, D, S\}$  组成, 其中  $C$  表示该预期的内容,  $D$  是预期的深度,  $S$  为验证该预期的假设集合;

(3) 对根结点  $r$ , 有  $r(D)=0$ ;

(4) 对叶结点  $l$ , 有  $l(S)=\Phi$ ;

(5) 如果预期  $v$  (指结点  $v$  所代表的预期) 可以扩充为  $k$  个深一层的预期  $w_1, w_2, \dots, w_k$ , 则在预期集中,  $w_1, w_2, \dots, w_k$  是  $v$  的子结点, 且有

$$w_1(C) \supset v(C), w_2(C) \supset v(C), \dots, w_k(C) \supset v(C)$$

$$w_1(D) = w_2(D) = \dots = w_k(D) = v(D) + 1$$

$$v(S) = w_1(S) \cup w_2(S) \cup \dots \cup w_k(S)$$

( $w_i(C) \supset v(C)$ , 表明  $w_i$  的预期内容比  $v$  的更详细).

### 2.2.2 预期的形成

假定第  $i+1$  层假设由第  $j_1, j_2, \dots, j_i$  层假设组成 ( $j_1 < j_2 < \dots < j_i \leq i$ ), 则形成预期集  $P_i$  的基础是已知的假设集:  $H_{j_1} \cup H_{j_2} \cup \dots \cup H_{j_i}$ .

我们采用的预期方法是, 根据假设的属性, 用决策树分类法, 对  $H_{j_1} \cup H_{j_2} \cup \dots \cup H_{j_i}$  中的假设进行分类, 决策树中每个结点的分类结果对应一个预期, 所有预期按决策树的结构形成预期集.

决策树中采用的假设属性和决策规则同领域知识有关, 即不同的具体问题应采用不同的属性和决策规则. 例如, 运输调度问题所采用的假设属性有: 货物类型、假设的位置(坐标)和运量. 决策规则分别为: (1) 货物类型相容的假设可纳入同一类; (2) 位置相邻的假设可纳入同一类; (3) 运量相近的假可纳入同一类.

在应用决策规则时, 要使用不同的数学方法, 如规则(1)、(3)可通过预先定义“相容的货物类型集合”(CFTS)和“相近的运量范围”(CVR)来实现, 规则(2)则可先设定“类间距离度量函数”和“类间相似度”(IS), 然后用分级聚类的方法来完成<sup>[7]</sup>. 每个决策规则都至少有一个决策参数(如 CFTS, NVR, IS).

最终形成的预期集受下面的 2 个因素的影响:

(1) 决策规则的应用次序

使用决策规则的先后次序影响着预期的结果, 分辨力强的决策规则应优先采用.

(2) 预定的决策参数

决策参数影响类间重迭部分和  $BA$  将产生的第  $i+1$  层假设数目(假定形成  $P_i$ ). 决策参数越宽松(在运输调度问题中, CFTS 中元素越多、NVR 越宽、IS 越小), 类间重迭部分就越

大,BA 将产生的第  $i+1$  层假设数目就越多.

下面,通过一个例子说明在运输调度问题中,以  $H_0(=S_0=E_0)$  为基础,生成预期集  $P_0$  的过程<sup>[5,6]</sup>.

假定在某矩形区域  $A$  内分布有 I、II、III 三类 20 笔运输业务(图 2),要求根据初始数据(运输业务)形成第一个预期集  $P_0$ .

我们用“货物类型”和“位置”两个属性,按决策规则(1)、(2)来产生预期集  $P_0$ ,决策规则的应用次序为:

规则(1)→规则(2);

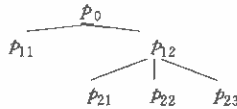
决策参数选为:  $CFTS_1 = \{ I \}, CFTS_2 = \{ II, III \}$ ;

(类型 I、III 业务相容,即所运货物可同车装载;类型 I 与 II、类型 I 与 III 的业务不相容,即这两种类型的业务所运货物不能同车运,例如煤与大米.)

类间距离度量函数选为最远距离度量;

$IS = 80\%$

形成的预期集为:



$p_0(c) = \Phi, p_0(D) = 0, p_0(s) = \Phi; p_{11}(c) = \{ \text{货物类型为 I} \}, p_{11}(D) = 1, p_{11}(s) = \Phi; p_{12}(c) = \{ \text{货物类型为 II 或 III} \}, p_{12}(D) = 1, p_{12}(s) = \Phi; p_{21}(c) = \{ \text{i. 货物类型为 II 或 III, ii. 业务位于子区域 } A_1 \text{ 内} \}; p_{21}(D) = 2, p_{21}(s) = \Phi; p_{22}(c) = \{ \text{i. 货物类型为 II 或 III, ii. 业务位于子区域 } A_2 \text{ 内} \}; p_{22}(D) = 2, p_{22}(s) = \Phi; p_{23}(c) = \{ \text{i. 货物类型为 II 或 III 类, ii. 业务位于子区域 } A_3 \text{ 内} \}; p_{23}(D) = 2, p_{23}(s) = \Phi;$

### 2.2.3 预期的验证

BA 在预期集  $P_i$  的指导下,调用知识源生成第  $i+1$  层假设,其中  $n_{i+1}$  个度量值最大者构成假设集  $H_{i+1}$ .

TA 根据假设集  $H_{i+1}$  验证预期集  $P_i$ .

验证算法如下:

- (1)  $H^* \leftarrow H_{i+1}$ ;
- (2) 取  $h, h \in H^*$ ;
- (3) 遍历  $P_i$  的叶结点  $v$ ,  
如果  $h$  满足  $v(c)$  的约束,则  $v(s) \leftarrow v(s) \cup \{h\}$ ;
- (4)  $H^* \leftarrow H^* - \{h\}$ , 如  $H^* \neq \Phi$  转(2);
- (5) 按预期集定义(定义 3.1),产生  $P_i$  中所有非叶结点  $w$  的  $w(s)$ ;
- (6) 检查验证条件  
如满足,则进一步形成预期集  $P_{i+1}$ , 如不满足,则修改预期集  $P_i$ ;
- (7) 验证结束.

(1)至(5)是为了找出验证预期集  $P_i$  的第  $i+1$  层假设,据此可检查验证条件是否满足.

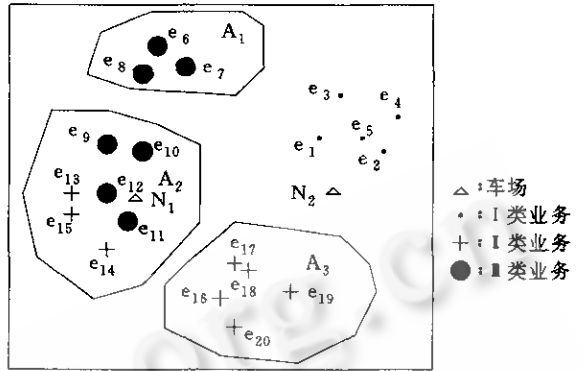


图2 运输调度中的预期

验证条件是验证算法的关键,它的定义与具体领域知识有关. 修改  $P_i$  就是调整决策规则应用次序或改变决策参数,重新形成预期集  $P_i$ .

### 2.3 BA 的问题求解

BA 在预期集的指导下,调用知识源,自底向上地进行求解. 每个知识源为一个具有部分问题求解能力的进程,由条件与动作两部分组成,当现有数据使条件满足时,可调用此知识源执行其动作(这种条件已满足,待执行的知识源称为知识源实例 KSI (Knowledge Source Instantiation)). KSI 的执行由调度器统一调用.

首先给出目标树  $GT$  (Goal Tree) 的定义:

定义 2.2. (目标树  $GT$ )

(1) 一个目标树  $GT$  是一棵树;

(2)  $GT$  中每个结点是一个目标,由二元组  $\{G, D\}$  组成,其中  $G$  表示目标的内容,  $D$  为目标的深度;

(3) 对预期集  $P_i$  中每个结点  $v$ , 在目标树  $GT$  中,有且仅有一个结点  $v'$ , 满足:

$v'(G) = \{\text{以受 } v(c) \text{ 约束的假设集为基础,产生第 } i+1 \text{ 层假设}\}$

$v'(D) = (D)$ .

并且,如果  $v$  有  $k$  个子结点  $w_1, w_2, \dots, w_k$ , 则  $v'$  也有相应的  $k$  个子结点  $w'_1, w'_2, \dots, w'_k$  满足:

$w'_1(G) = \{\text{以受 } w_1(c) \text{ 约束的假设集为基础,产生第 } i+1 \text{ 层假设}\};$

$w'_1(D) = w_1(D);$

$\dots, \dots, \dots$

$w'_k(G) = \{\text{以受 } w_k(c) \text{ 约束的假设集为基础,产生第 } i+1 \text{ 层假设}\};$

$w'_k(D) = w_k(D);$

BA 的问题求解算法如下:(假定,在  $P_i$  指导下,生成第  $i+1$  层假设)

(1)  $H \leftarrow \Phi, H_{i+1} \leftarrow \Phi;$

(2) 根据预期集  $P_i$  生成目标树  $GT;$

(3) 为  $GT$  中叶结点目标排序;

(4) 取排序第一的叶结点目标;

(5) 确定实现该目标的 KSI 的优先级;

(6) 取优先级最高的 KSI;

(7) 调用、执行上述 KSI, 如生成一个第  $i+1$  层假设  $h$ , 则  $H \leftarrow H \cup \{h\};$

(8) 取优先级次之的 KSI, 转(7). 直至实现当前目标的所有 KSI 执行完毕.

(9) 取排序次之的叶结点目标, 转(5), 直至  $GT$  中所有叶结点目标均被选择;

(10) 从  $H$  中取  $n_{i+1}$  个度量值较大者, 放入  $H_{i+1}.$

## 3 讨 论

双向求解算法具有 2 个优点:

(1) 提高了自底向上求解的效率与主动性

假定第  $i+1$  层假设由  $n$  个低层假设组成, 产生第  $i+1$  层假设时所依据的低层假设总数为  $m$ , 那么在没有预期指导的情况下, BA 找到一个度量值最大的第  $i+1$  层假设的计算复杂性为:  $CX_N = A_m^n; (A_m^n \text{ 代表从 } m \text{ 个数中取 } n \text{ 个数的排列数目})$

如果采用双向求解方法,  $TA$  先产生预期集  $P_i$ , 设  $P_i$  的叶结点预期结点分别为:  $p_1, p_2, \dots, p_k$ , 满足  $p_j(c)$  约束的假设数目为  $m_j$ , 那么,  $BA$  在  $P_i$  指导下, 找到一个度量值最大的第  $i+1$  层假设的计算复杂性为:  $CX_y = A_{m_1}^n + A_{m_2}^n + \dots + A_{m_k}^n (m_1 + m_2 + \dots + m_k = m)$

可以证明

$$A_m^n > A_{m_1}^n + A_{m_2}^n + \dots + A_{m_k}^n$$

故

$$CX_y < CX_N$$

即双向求解算法提高了  $BA$  自底向上求解的效率。

由于  $p_1, p_2, \dots, p_k$  是根据当前问题求解状态得出的, 因此,  $BA$  在这些预期指导下的求解大多数情况下是主动、有效的。

## (2) 预期指导性强、灵活

$TA$  在产生第  $i+1$  层假设之间, 根据已知的所有有关的第  $i$  层及低层假设, 形成预期集  $P_i$ , 不仅能识别出解空间中邻近区域间的关系, 而且能识别出解空间中较远区域间的关系。

$BA$  在  $P_i$  指导下, 产生假设集  $H_{i+1}$  之后, 又将  $H_{i+1}$  送至  $TA$ , 由其按  $H_{i+1}$  验证  $P_i$ , 这样能适时判断部分解 ( $H_{i+1}$ ) 的质量, 并通过修改  $P_i$  重新求解来保证部分解的质量。

本文提出的双向求解算法, 在当前全面的问题求解状态基础上, 形成预期, 指导问题求解, 并通过验证机制, 调整预期。该方法一方面提高了预期的指导性和灵活性, 另一方面也为在算法上对目标指导与数据指导相结合的求解方法做深入研究提供了基础。

当然, 双向求解算法也存在些问题。首先在有的实际应用中 (如语音理解), 难以给出宏观预期 (尤其在低层); 其次在实际运用双向求解算法时, 应在形成和验证预期的计算开销同预期对提高自底向上求解速度的作用之间权衡, 这与问题类型、规模及初始数据分布有关。

## 4 结 论

本文提出一种双向求解平行结构类问题的算法。该方法不仅提高了预期的指导性, 而且使问题求解更为灵活。该方法正在分布式运输调度系统的上实现<sup>[8]</sup>。

## 参考文献

- 1 Durfee E H *et al.* Trends in cooperative distributed problem solving. IEEE Trans. Knowl., Data Eng., 1989, 1 (1): 63-83.
- 2 Special issue on distributed sensor networks. IEEE Trans. Syst., Man, Cybern., 1991, 21(5): 1027-2101.
- 3 胡蓬, 石纯一, 苏伯琪. 平行结构类问题求解方法的研究. 软件学报, 1993, 4(2): 21-26.
- 4 Erman L D *et al.* The hearsay-II speech-understanding system: integrating knowledge to resolve uncertainty. Computing Surveys, 1980, 12(2): 213-253.
- 5 Corkill D D. A framework for organizational self-design in distributed problem solving network. Ph. D Dissertation, Univ. of Massachusetts, Feb. 1983.
- 6 Durfee E H, Lesser V R. Incremental planning to control a time-constrained, blackboard-based problem solver. COINS Tech. Rep. 87-07, Feb. 1987.
- 7 边肇祺. 模式识别. 北京: 清华大学出版社, 1988.
- 8 Su Bogong, Shi Chunyi, Wang Kehong *et al.* A distributed problem solving system for transport dispatching. Proc. Third International Conference of Industrial Engineering Applications of Artificial Intelligence and Expert Systems, Charleston, South Carolina, USA, 1990.

## AN INTEGRATED APPROACH TO FLAT—STRUCTURED PROBLEMS

Hu Peng Shi Chunyi Wang Kehong

*(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)*

**Abstract** Flat—structured Problems (FP) are an important class of Cooperative Distributed Problem Solving applications which include speech understanding, vehicle monitoring, transport dispatching and so on. So far, a number of approaches to FPs have been developed such as those in Hearsay—II and DVMT. Most of these approaches use predictions or goals to guide bottom—up problem solving. However, most predictions and goals in these approaches are based on local view of problem solving states. Although the improved architecture of DVMT allowed a high—level view, no explicit algorithm was given. This paper gives an integrated approach to FPs which makes top—down predictions from global problem solving states, guides bottom—up solving by predictions, verifies and modifies predictions by newly—created hypotheses, and guide bottom—up solving once again. This approach not only enhances the directing role of predictions obtained from global problem solving states, but also makes problem solving flexible due to the prediction verification mechanism.

**Key words** Distributed problem solving, prediction, verification.