

基于规划匹配的 BLACKBOX 系统

黄蕾, 陈伟, 刘峰, 赵志宏

(南京大学软件学院, 南京 210093)

摘要: 提出一个基于规划匹配的 BLACKBOX 系统(CBB), 该系统将基于案例规划方法中的规划匹配思想与 BLACKBOX 方法相结合, 通过将现有问题与已有的求解记录相比较来自动确定最有效的规划工具。实验结果表明, CBB 系统对于规划工具和规划问题都有较好的自适应性和求解效率。

关键词: 规划; BLACKBOX 技术; 基于案例的规划

BLACKBOX System Based on Planning Matching

HUANG Lei, CHEN Wei, LIU Feng, ZHAO Zhi-hong

(Software Institute, Nanjing University, Nanjing 210093)

【Abstract】 This paper proposes a Case-Based BLACKBOX(CBB) system, which combines the planning matching method of Case-Based Planning (CBP) to BLACKBOX approach. When solving a new planning problem, CBB selects the best planner by matching the current problem with the existing planning records. Experimental results show that CBB can obtain a good performance without any manual pre-configuration.

【Key words】 planning; BLACKBOX thchnology; Case-Based Planning(CBP)

1 概述

规划(planning)指对预定目标制定出符合选择范围和资源限制的动作序列的过程,是人工智能研究的核心领域之一。根据求解策略的不同,规划方法可分为直接求解法和间接求解法。直接求解法是指直接寻找适用规则的方法,主要包括将规划过程转化为定理证明的演绎法^[1]和借助一定理论模型来抽取规则的模型法^[2]。而间接求解法则是通过分析历史数据中的相似的问题来获取规划方法。

文献[3]提出的 BLACKBOX 方法是一种代表性的直接求解方法。它结合了模型法和演绎法的思想,允许在一个规划问题的求解过程中分时段指定多个规划工具。这种机制使得 BLACKBOX 具有较好的灵活性,往往能够获得较高的求解效率;但由于不同规划工具在针对不同类型的规划问题时求解效率各不相同,这就要求使用者必须对各种工具的差异和所求解的规划问题有深入的了解。

与此相比,作为间接求解方法的基于案例的规划(Case-Based Planning, CBP)^[4]则在使用者对规划的了解方面没有太高的要求。对于每一个新的规划问题, CBP 方法会将其与以往的规划求解情况按照既定的规则进行匹配,将匹配所得的问题作为当前问题的“解”并在需要时进行修正。这种求解方法对使用者屏蔽了规划的求解细节,却依赖于已有的历史数据。

结合 BLACKBOX 和 CBP 的优点,本文提出 Case- Based Blackbox(CBB)方法,将 CBP 的规划匹配和重用思想应用于 BLACKBOX 系统的规划工具选择阶段,在不需要使用者预先设定的情况下,系统可根据当前时刻规划求解状态自动选择适当的规划工具,从而达到较高的求解效率。

2 CBB系统框架

图1展示了CBB系统的总体框架。CBB系统采用 STRIPS 形式的问题描述作为输入,由 Graphplan 生成规划图后,再

转化为 CNF。

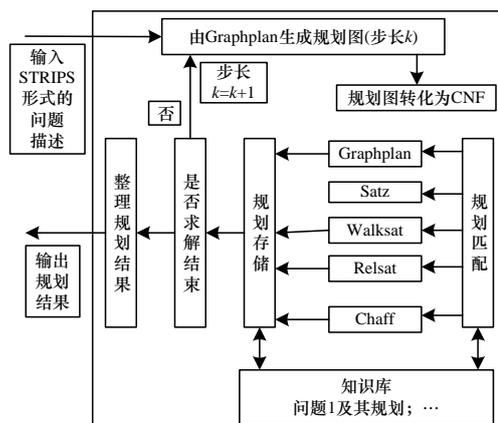


图1 CBB系统框架

与 BLACKBOX 系统相比, CBB 系统引入了规划匹配模块来代替人工对规划工具的设定,系统可根据当前规划求解状态选择合适的规划工具及调用参数。与 CBP 的规划选择机制相比, CBB 系统将匹配结果中适用于当前状态的规划工具直接运用于当前状态,而不需要通过调整相匹配的规划问题的解来适应当前的规划问题。同时, CBB 系统中知识库会由规划存储模块在当前规划求解结束后,通过记录本次规划的信息来动态更新。

3 CBB规划匹配实现

规划匹配是 CBB 系统的核心部分,用于实现根据当前状

基金项目: 江苏省高新技术基金资助项目(bg2006027)

作者简介: 黄蕾(1984-),女,硕士研究生,主研方向:人工智能,数据挖掘;陈伟,硕士研究生;刘峰,讲师、硕士;赵志宏,副教授、博士

收稿日期: 2009-07-08 **E-mail:** lei Huang.nju@gmail.com

态自动选择适当的规划来提高求解效率。CBB 系统中的规划匹配主要包括规划表示、规则匹配和匹配算法 3 个部分。

3.1 CBB 规划表示

在 CBB 系统中, 规划 PM 表示为一个四元组:

$$PM = \langle C, S, P, D \rangle \quad (1)$$

其中, C 表示不同的领域知识可能对规划效率的约束。在 CBB 系统中假设领域的名称是唯一的, 并将 C 简化为规划问题的领域名。 S 表示规划求解状态。状态 S 的基础指标包括: 规划图的步长 $step$, 动作层的变量数 act_num , 命题层的变量数 $prop_num$ 和子句数 cls_num 。在此基础上可定义状态 S 的其他指标: 规划图复杂度 $CMP_G = act_num / prop_num$, 用于描述动作作间的关联度; 规划图量级 $MGT_G = act_num + prop_num$, 用于描述规划图的节点数; 规划问题复杂度 $CMP_P = cls_num / MGT_G$, 用于描述子句间的关联度。根据上述特征, 状态 S 可表示成一个五元组:

$$S = \langle step, CMP_G, MGT_G, CMP_P, cls_num \rangle \quad (2)$$

其中, P 表示规划工具名; D 为规划工具 P 在状态 S 下的求解情况。由于规划求解存在不确定性, D 定义为三元组:

$$D = \langle expect_time, fail_time, total_num \rangle \quad (3)$$

其中, $expect_time$ 为期望时间, 简写为 ET ; $fail_time$ 为实际时间 $time \in [0.8ET, 1.2ET]$ 的求解次数, 简写为 FN ; $total_num$ 为求解总次数, 简写为 TN 。

3.2 CBB 规划匹配

规划匹配是指在知识库中找到与当前状态相似的历史记录, 并按照一定的策略选择其中一个作为本次规划求解的依据。CBB 系统选用二元组 $\langle C, S \rangle$ 作为规划匹配的特征。

在状态 S 的 5 个属性里, 对于给定的某个属性 $A \in \{cls_num, CMP_G, MGT_G, CMP_P\}$, A 在状态 S 与某历史记录 H 下的相似度为

$$sim(A) = 1 - |A_S - A_H| / A_H \quad (4)$$

其中, A_S 和 A_H 分别为属性 A 在 S 和 H 下的值。根据式(4), 状态 S 与历史记录 H 的相似度 $SIM_S(Th_{SIM})$ 为

$$SIM_S(Th_{SIM}) = \begin{cases} \sum sim(A), & \forall A, sim(A) \geq Th_{SIM} \\ -1, & \text{else} \end{cases} \quad (5)$$

其中, Th_{SIM} 为预先给定的阈值。

当前规划问题领域 C 和历史记录的领域 C_H 之间的相似度 SIM_C 定义为

$$SIM_C = \begin{cases} 1, & C = C_H \\ 0, & C \neq C_H \end{cases} \quad (6)$$

结合式(7)和式(8), 可以得到当前规划问题与历史记录 H 之间的总体相似度 $SIM_{(S,C)}$:

$$SIM_{(S,C)} = \begin{cases} SIM_S + SIM_C, & SIM_S \geq 0 \\ -1, & SIM_S < 0 \end{cases} \quad (7)$$

在实际特征匹配中, 由于 Graphplan 和其他工具在度量指标上各有侧重, 且在求解特定类型问题时(满足 $step < 10$ 且 $CMP_G < 2$ 且 $MGT_G < 5000$)有明显优势, CBB 系统针对 Graphplan 采取了特殊处理。

由于特征匹配结果对应于多个规划工具名, 需要对匹配结果进行选择。考虑到求解效率和系统求解时间稳定性, CBB 系统会选择期望时间短且置信度高的规划工具。根据式(3), 定义参考时间 DT :

$$DT = ET \times (1 + FN / TN) \quad (8)$$

CBB 系统中选用 DT 最小的规划工具。

3.3 规划匹配算法

规划匹配算法主要考虑 3 个方面: (1)由于表示状态的记录数可能达到数万条甚至数十万条, 特征匹配时必须考虑算法的效率; (2)符合特征匹配的记录数不能过多, 以免影响规划工具选择; (3)规划匹配结果需加上标记, 用于规划存储。算法 1 描述了规划匹配的算法。

算法 1 规划匹配算法

```

Input step, act_num, prop_num, cls_num, C
Output planner
Begin
Initialization: ThSIM=0.9, record_num=0;
Compute CMPG, MGTG, CMPP;
if (step<10) && (CMPG<2) && (MGTG<5000)
Return Graphplan;
while (ThSIM>0) && (record_num==0) {
Select records which satisfy SIMS(ThSIM)>0;
for (each selected record) {
Identify it with its SIMS(ThSIM);
record_num= record_num+1;
}
ThSIM= ThSIM-0.05;
}
if (record_num==0)
return Graphplan;
else {
Compute SIM(S,C)(ThSIM) of each identified record;
Compute DT for records with the maximum SIM(S,C)(ThSIM);
return the planner with the minimum DT;
}
End

```

4 CBB 规划存储实现

当前规划求解结束后, CBB 系统会将求解结果存储到知识库中。随着求解次数的增多, 不同规划工具间的差异记入知识库, 在面对新的规划问题时, 系统可以更有效地选择规划工具。

4.1 知识库组织方式

由于同一状态 S 可能对应多个规划领域和规划工具, 且规划匹配时对 S 的计算要求较高, 因此对状态 S 单独建表。如图 2 所示, 知识库将规划问题划分为 3 张表: 表 PLANNER, 表 STATUS 和表 CASE。

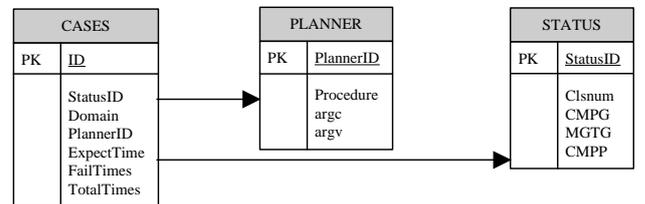


图 2 CBB 知识库设计

表 PLANNER 描述系统中现有的求解工具, 其中, PlannerID 为关键字; procedure 是该工具的实际调用函数名; argc 和 argv 存储调用函数时的可选参数。表 STATUS 描述了规划定义四元组中的状态 S ; 表 CASE 囊括了其余需要存储的信息, 其中, Domain 表示领域 C ; StatusID 和 PlannerID 为外关键字。

4.2 规划存储算法

规划存储算法的设计需要考虑 2 个方面: (1)设定并调整

规划工具在指定状态下的期望时间接近平均时间；(2)增加状态的策略需考虑对知识库记录数的影响。算法 2 描述了规划信息存储中插入新记录和修正原有记录的算法。

算法 2 规划存储算法

```

Input act_num, prop_num, cls_num, C', P', T, ThSIM
Output none
Begin
if (ThSIM≤0.8) {
  Compute SID' with formula (2);
  for (each P in Table PLANNER) {
    Insert <SID', C', P', <T, 0, 1>> to Table CASE;
  }
  else
    Insert <SID', C', P', <-T, 0, 1>> to Table CASE;
}
}
else {
if (ET<0)
  Modify the record: Domain=C, ExpectTime=T;
else if (T>=0.8ET) && (T<=1.2ET)
  Modify the record: TotalNum=TotalNum+1;
else if (FN/TN<0.5)
  Modify the record: TotalNum=TotalNum+1;
  FailNum=FailNum+1;
else
  Modify the record: TotalNum=TotalNum+1;
  FailNum=FailNum+1; ExpectTime=T;
}
}
End

```

5 实验结果

对规划系统和模型搜索算法的评价普遍以对基准规划问题(Benchmark Planning Problem, BPP)的求解为标准。本文实验为求解效率实验,说明 CBB 系统自动选择工具的效率。实验均基于 WindowsXP 下的 Cygwin 环境,硬件配置为 Pentium4 2.0 GHz 处理器和 512 MB 内存。

假定 CBB 里有 Graphplan, satz 和 chaff^[5] 3 种求解工具。令 CBB 以指定规划工具的方式将各模型搜索工具用一组规划问题(BPP 问题占 1/3)训练 3 次,以获得不同规划工具的求解记录,再按照一般求解方式对 BPP 问题求解 5 次,并取后 3 次作为结果(表 2)。其中,Time1, Time2 和 Time 3 分别是 CBB 对 BPP 问题 3 次求解的时间。AVG 是 CBB 求解规划的平均时间。MIN 列是在 BLACKBOX 中单独使用不同模型搜

索算法需要的最短时间,作为参照时间。

表 1 CBB 的求解效率

BPP	Time1	Time2	Time3	AVG	MIN
log-easy	0.19	0.19	0.19	0.19	0.15
rocket-a	1.56	1.55	1.49	1.53	1.19
rocket-b	1.77	1.74	1.83	1.78	1.75
log-a	1.23	0.92	1.21	1.12	0.81
log-b	2.76	2.98	2.61	2.78	2.52
log-c	2.79	3.69	2.95	3.14	4.21
log-d	9.83	9.84	10.91	10.19	11.68
log-d3	8.08	8.12	9.45	8.55	9.61
log-d1	17.59	18.17	18.76	18.16	19.36

从表 1 中可以发现,由于规划工具运行求解时间具有不确定性,Time1, Time2 和 Time3 略有差异,但三者的差异并不大且和参考时间相近,这表明 CBB 对当前求解状态所适用的模型搜索工具的选择基本正确。对于部分规划问题,CBB 的求解时间甚至比参考时间更短,这表明 CBB 可以通过结合不同规划工具在不同求解状态上的表现来提高求解效率。对于较复杂的规划问题,CBB 的求解效率较高,普遍优于独立使用规划工具求解。

6 结束语

CBB 系统通过在求解过程中存储当前结果来更新知识库,无需人工干预,自主选择适用的规划工具,从而获得较高的效率。实验结果表明,CBB 系统对新的规划问题和新加入的规划工具都具有很好的自适应性。CBB 系统中还有需要改进的地方。例如,如何为新加入的规划工具提供合适的训练集,如何根据不同的求解状态为同一种规划工具设定相应的参数,都值得进一步研究。

参考文献

- [1] Yang Qiang. Intelligent Planning: A Decomposition and Abstraction Based Approach[M]. Berlin, Germany: Springer, 1997.
- [2] 陈舒,戈也挺,骆斌,等. 模型法求规划的研究[J]. 计算机科学, 2004, 31(5): 138-141.
- [3] Kautz H, Selman B. Blackbox: A New Approach to the Application of Theorem Proving to Problem Solving[C]//Proc. of AIPS'98 Workshop on Planning as Combinatorial Search. PA, USA: [s. n.], 1998: 58-60.
- [4] Hammond K. Case-based Planning: Viewing Planning as a Memory Task[M]. 1st Ed. San Diego, USA: Academic Press, 1989.
- [5] Moskewicz M W, Madigan C F, Zhao Y. Chaff: Engineering an Efficient SAT Solver[C]//Proceedings of the 38th Design Automation Conference. Las Vegas, USA: [s. n.], 2001: 530-535.

编辑 索书志

(上接第 58 页)

性能上优于 PrefixSpan 算法。未来的研究问题包括如何加入有效的约束条件以及与闭合序列相结合,在改善时空性能的同时挖掘出更加有价值的序列模式。

参考文献

- [1] Agrawal R, Srikant R. Mining Sequential Patterns[C]//Proc. of the 11th Int'l Conf. on Data Engineering. Taipei, China: [s. n.], 1995: 3-14.
- [2] Srikant R, Agrawal R. Mining Sequential Patterns: Generalizations and Performance Improvements[C]//Proc. of the 5th Int'l Conf. on Extending Database Technology. Avignon, France: [s. n.], 1996: 3-17.

- [3] Zaki M J. SPADE: An Efficient Algorithm for Mining Frequent Sequences[J]. Machine Learning Journal, Special Issue on Unsupervised Learning, 2001, 42(1/2): 31-60.
- [4] Pei Jian, Han Jiawei, Mortazavi B, et al. FreeSpan: Frequent Pattern-projected Sequential Pattern Mining[C]//Proc. of the 6th Int'l Conf. on Knowledge Discovery and Data Mining. New York, USA: [s. n.], 2000: 355-359.
- [5] Pei Jian, Han Jiawei, Mortazavi-Asl B, et al. Mining Sequential Patterns by Pattern-growth: The PrefixSpan Approach[J]. IEEE Trans. on Knowledge and Data Engineering, 2004, 16(11): 1424-1440.

编辑 索书志