

模型法求规划的研究^{*}

陈舒 戈也挺 骆斌 陈世福

(南京大学计算机软件新技术国家重点实验室 南京210093)

摘要 模型法求规划是一种通用的规划方法。它从给定问题的模型中抽取出规划。本文综述了模型法求规划的基本方法。首先介绍了 SATPLAN 和 CSP,以讨论模型法求规划的通用框架。然后介绍了新近开发的模型法求规划系统, BLACKBOX 和 GP-CSP。在将模型法与演绎法和 CBP 法进行比较后,给出了模型法求规划的未来研究方向。

关键词 规划,模型法求规划

Research on Model-Based Planning

CHEN Shu GE Ye-Ting LUO Bin CHEN Shi-Fu

(State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract Model-based planning is a general planning method. It extracts a plan from the model of the given problem. This paper presents a survey of model-based planning. At first, it introduces SATPLAN and CSP to discuss the general frameworks of model-based planning. Then it introduces newly developed model-based planning systems, BLACKBOX and GP-CSP. After comparing model-based planning with planning as deduction and CBP, it gives some remarks on the future of model-based planning.

Keywords Planning, Model-based planning

1 引言

在人工智能领域,规划问题是一个研究重点。规划问题有专用和通用两类解决方案。专用解决方案指为特定问题设计专用的算法,比如,我们可以针对“汉诺塔”问题编写算法。这类专用算法通常效率较高,但只能针对某个问题。因此我们希望能找到一种通用的规划求解方法。STRIPS 系统就是早期著名的通用规划系统^[1]。

通用的规划方法可以分为演绎法和模型法。演绎法与定理证明类似,它将规划目标视为需要证明的定理,如果定理可被证明,则定理的推导过程对应于一个规划。STRIPS 系统即使用演绎法。模型法求规划则是求满足约束条件的理论的模型,如模型存在,即可从该模型中抽取出规划。因为在一阶逻辑中,定理证明是半可判定的,而求模型是不可判定的,所以人们曾经认为演绎法较模型法更加容易求解。20世纪90年代的研究表明,对命题逻辑而言演绎法和模型法在问题求解的难易度上没有太大区别,而且模型法还有诸多优点。随着求模型算法的不断发展,人们发现使用模型法可以获得较高的求解效率。模型法成为通用规划求解的主要方案。

本文试图对模型法求规划进行总结。首先通过 SATPLAN^[12]和 CSP^[34]介绍模型法求规划的基本方法;然后介绍从 SATPLAN 发展的 BLACKBOX^[16]和从 CSP 发展的 GP-CSP^[28];第4节比较模型法(以 SATPLAN 为代表)与演绎法(以 STRIPS 为代表);第5节比较模型法和另一类规划求解方式 Case-based Planning^[24];最后是小结。

2 SATPLAN 和 CSP

初期的模型法以 SATPLAN 和 CSP 为代表。以下举例说明。

例:如图1,假设初始状态下箱 A 在箱 B 上,箱 B 和箱 C

在桌上,箱 A 和箱 C 上没有其它箱子,目标是要在两个动作后 C 在 B 上。

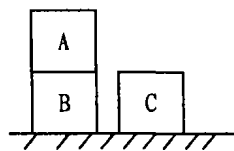


图1

2.1 SATPLAN

SATPLAN 是基于行动理论的^[20,35]。采用 SATPLAN 需要按行动理论的要求给出观察公理、效应公理等。本例中,变量 x, y, z 代表各个物体,其取值范围为 $\{A, B, C, TABLE\}$;时间变量 i 的取值范围为 $\{1, 2, \dots\}$,且 $i+1$ 指紧接在 i 时刻后的那个时刻。谓词 $on(x, y, i)$ 指在 i 时刻物体 x 放在物体 y 上; $clear(x, i)$ 表示在 i 时刻物体 x 上没有其他物体; $move(x, y, z, i)$ 表示在时刻 i 将物体 x 从物体 y 上移动到物体 z 上。

一个可能的描述如下:

$$on(A, B, 1) \wedge on(B, TABLE, 1) \wedge on(C, TABLE, 1) \wedge clear(A, 1) \wedge clear(C, 1) \wedge on(C, B, 3) \quad (1)$$

$$\forall x, y, z, i (on(x, y, i) \wedge clear(x, i) \wedge clear(z, i) \wedge move(x, y, z, i) \supset on(x, z, i+1) \wedge clear(y, i+1) \wedge clear(x, i+1)) \quad (2)$$

$$\forall x, y, z, i (move(x, y, z, i) \supset (on(x, y, i) \wedge clear(x, i) \wedge clear(z, i))) \quad (3)$$

$$\forall x, z1, y, y1, z, z1, i ((x \neq x1 \vee y \neq y1 \vee z \neq z1) \supset \neg move(x, y, z, i) \vee \neg move(x1, y1, z1, i)) \quad (4)$$

$$\forall i < 3 (\exists x, y, z (move(x, y, z, i))) \quad (5)$$

所有公式分为四类:

1) 初始状态和目标状态,如式(1),对应于行动理论的观察公理。

2) 动作的前提和结果描述,对应于行动理论的效应公理。式(2)是对 $move$ 动作的结果的描述;式(3)是对动作前提条

^{*} 本文得到国家自然科学基金(60003010)资助。陈舒 硕士,研究方向为人工智能与非经典逻辑。戈也挺 硕士,研究方向为人工智能与非经典逻辑。骆斌 博士,副教授,研究方向为人工智能,智能数据库。陈世福 教授,博士生导师,研究方向为人工智能。

件的描述。

3)其他约束条件。式(4)保证在某一时刻至多只有一个动作执行。当然,我们也可以附加定义两个动作是否互斥,并规定,两个动作可以同时执行当且仅当这两个动作不互斥。

4)“空动作”,如式(5)。“空动作”在任一时刻都出现,属于系统的约束条件。

对问题描述完整后,可以利用任何一种 SAT 算法求解满足公式集的模型,若模型存在即可得到相应的规划。如上例中存在模型式(6),根据它可以得到规划式(7)。

```
{ on(A,B,1),on(B, TABLE,1),on(C, TABLE,1),clear(A,1),clear(C,1),clear(TABLE,1),
move(A,B, TABLE,1),on(A, TABLE,2),on(B, TABLE,2),on(C,
TABLE,2),
clear(A,2),clear(B,2),clear(C,2),clear(TABLE,2),move(C,
TABLE,B,2),
on(C,B,3),on(B, TABLE,3),on(A, TABLE,3),clear(A,3),clear(C,3),clear(TABLE,3)} (6)
move(A,B, TABLE,1) ^ move(C, TABLE,B,2) (7)
```

2.2 CSP(Constraint Satisfaction Problem)

一个 CSP 问题三元组 $\langle X, D, C \rangle$ 表示。其中, $X = \{X_1, X_2, X_3, \dots\}$ 为变量集; $D = \{D_1, D_2, D_3, \dots\}$, 每个 D_i 对应于变量 X_i 的有限的取值范围,如变量 X_1 的取值范围为0和1,则它对应的值域 $D_1 = \{0, 1\}$; $C = \{c_1, c_2, c_3, \dots\}$ 为约束集,每个约束 c_i 是 X 的子集上的一个关系,用来说明对于某一组变量哪些元组的取值是允许的,如 $X_1 < X_2$ 。CSP 的解则是满足约束集的所有变量的一个解。

例如,上例可表示为 $\langle X, D, C \rangle$, 其中 $X = \{A_1, B_1, C_1, A_2, B_2, C_2, A_3, B_3, C_3\}$, 代表箱子 A、B、C 在 1, 2, 3 时刻的状态; 集合 $D = \{D_{A_1}, D_{B_1}, D_{C_1}, D_{A_2}, D_{B_2}, D_{C_2}, D_{A_3}, D_{B_3}, D_{C_3}\}$, 其中 $D_i = \{A, B, C, D\}$, 代表箱子 A、B、C 和桌子 D, $A_1 = B$ 表示箱子 A 在 1 时刻放在箱子 B 的上面; 约束集 $C = \{C_1, C_2, C_3, \dots\}$ 由以下约束组成:

- (8) $(A_1 = B) \wedge (B_1 = D) \wedge (C_1 = D)$
- (9) $C_3 = B$
- (10) $\forall t, X. X_t \neq X$
- (11) $\forall t, X, Y, Z. (X_t = Y) \wedge (X_t = Z) \Rightarrow Y = Z$
- (12) $\forall t, X, Y, Z. (X_t = Z) \wedge (Y_t = Z) \wedge (Z \neq D) \Rightarrow X = Y$
- (13) $\forall t, X, Y, Z. (X_t = Y) \wedge (X_{t+1} = Z) \Rightarrow (\neg \exists K (K_t = X)) \wedge ((\neg \exists K (K_t = Z)) \vee (X = D))$

以上约束可以分为状态约束、步长约束、资源容量约束等。式(8)和(9)是关于初始状态和目标状态(假定步长为2)的约束; 状态约束式(10)表示一个箱子不可能放在自己的上面, 式(11)表示一个箱子在某个时刻不可能放在两个不同物体上; 步长约束限定该规划的步数, 即 $t = 1, 2, 3$; 资源容量约束式(12)要求所有的箱子上至多只能(直接)放一个箱子; 动作描述式(13)指搬箱子动作发生的前提条件。这些约束可以是对同一个约束条件不同角度的表述, 不同表述产生的冗余有时可以提高求模型的速度。

在 CSP 中, 若模型存在即可得到相应的规划。如上例可求的模型式(14), 并根据该模型得到规划式(15)(此处引入 SATPLAN 中 move 动作的定义来表示所得的规划)。

```
(A1=B) ^ (B1=D) ^ (C1=D) ^ (A2=D) ^ (B2=D) ^ (C2=D)
^ (A3=D) ^ (B3=D) ^ (C3=B) (14)
move(A,B, TABLE,1) ^ move(C, TABLE,B,2) (15)
```

须指出, 对同一个规划问题, 变量的不同设定会导致不同的求解效率。在 GP-CSP 中, 搬箱子问题的变量以动作为值域^[28], 因此不需要在求得模型后再次计算对应的规划。

2.3 SATPLAN 和 CSP 的比较

SATPLAN 是基于行动理论的, 理论研究较多^[35], 描述问题较 CSP 清晰简洁。例如在上面的例子里, 若要描述同一时刻只能有一个动作, 我们采用的这种 CSP 描述方式就不能

给出直观的约束表达。而 CSP 在规划问题发生变化时比 SATPLAN 有更强的适应能力。在 SATPLAN 中, 为避免不同动作的相互冲突, 要求同一时刻只能有一个动作执行; 而 CSP 中没有这样的强制要求, 因为它的一些静态约束可以保证动作间没有相互的冲突。

3 BLACKBOX 和 GP-CSP

为进一步提高效率, H. Kautz 和 B. Selman 提出 BLACKBOX^[16], M. Do 和 S. Kambhampati 提出 GP-CSP^[28]。它们都是将已有的规划方式和 Graphplan^[3]相结合的产物。

3.1 Graphplan

Graphplan 从 STRIPS 描述出发构筑规划图。规划图由命题层(proposition level)、动作层(action level)和它们中存在的联系构成, 如图2。其中奇数层为命题层, 是该时刻可能为真的命题的集合, 第一层命题层表示初始状态; 偶数层为动作层, 是在前一层的命题成立时可能执行的动作的集合; 用箭头表示联系, 包括同一命题层的不同命题之间的互斥(箭头1)、某个命题是下一层某动作的前提(箭头2)、某个动作产生的命题(箭头3), 以及同一动作层的不同动作之间互斥(箭头4)。

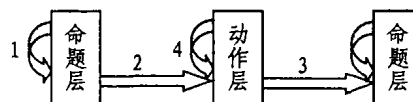


图2

规划的搜索以规划图为基础, 具体方式如下: (1)以初始状态构成第一层命题层; (2)查找当前最后一层命题层(第 i 层)是否包含目标状态, 若有, 从目标状态出发查找从初始状态到达目标状态的路径, 一旦找到这样的路径即可获得相应的规划并结束算法; (3)若不包含目标状态或是没有找到路径, 则根据第 i 层构筑第 $i+1$ 层(动作层)和第 $i+2$ 层(命题层), 并返回到第(2)步。

Graphplan 在构筑规划图时效率较高, 同时, 规划图中包含较多互斥信息, 可以在搜索时减少搜索空间^[10]。

3.2 BLACKBOX

BLACKBOX 结合了 SATPLAN 和 Graphplan 的方法。虽然, SATPLAN 基于求模型的思想, 而 Graphplan 使用 STRIPS 系统的实现思想, 属于演绎规划的范畴。但是, 两者都将求规划的步骤分为两步: 第一步, 构建一个命题结构(prepositional structure); 第二步, 基于所构建的命题结构进行搜索。SATPLAN 中以输入的公理集来构建 CNF, Graphplan 中以 STRIPS 模式的操作集为输入构建规划图。Kautz 和 Selman 提出规划图可以转化为 CNF 形式, 并由此得到一种新的规划求解器——BLACKBOX。BLACKBOX 的求解过程如下: (1)将 STRIPS 形式表示的规划问题转换为步长 k 的规划图, 并将其中的互斥关系一并计算出来; (2)把规划图转化为 CNF; (3)用通用的 CNF 简化算法对之化简; (4)利用现有的一个搜索算法求解模型; (5)如果找到了模型, 就将这个模型转化为相应的规划, 如果没有, 则将步长 k 加 1, 并返回到第(1)步。另外, “BLACKBOX”可以使用多种搜索算法求模型。比如, 可以直接对规划图使用 Graphplan, 或者在 CNF 的基础上运行 Walksat^[5], 若是预定时间内没有求出结果, 还可以再改用另一种求模型的工具, 比如 satz^[7]。

3.3 GP-CSP

M. Do 和 S. Kambhampati 发现 CSP 和 Graphplan 之间的转换关系并提出另一种求规划的方式:GP-CSP。在 GP-CSP 中,规划图被转化为 CSP 形式,然后采用 CSP 的求解方式寻找满足约束的模型。在 GP-CSP 中,变量的取值域为各个动作,各个时刻的状态隐含在该时刻可以执行的动作中;在 SATPLAN 中,状态和动作是分离的,因此它求出模型后需要计算模型所对应的动作序列。

3.4 BLACKBOX 与 GP-CSP 的比较

目前的研究一般以实验数据作为不同实现方式间比较的依据。表1、表2分别列出了 BLACKBOX 和 GP-CSP 的一些实验数据。

表1 BLACKBOX 与 graphplan、SATPLAN 求解效率的比较^[16]

Problem	BLACKBOX		graphplan	SATPLAN		
	walksat	satz		create	walksat	satz
rocket. a	3. 2sec	5sec	3. 4min	42sec	0. 02sec	0. 3sec
rocket. b	2. 5sec	10sec	8. 8min	41sec	0. 04sec	0. 3sec
log. a	7. 4sec	5sec	31. 5min	1. 2min	2sec	1. 7min
log. b	1. 7min	7sec	12. 7min	1. 3min	3sec	0. 6sec
log. c	14. 9min	9sec	>24hour	1. 7min	2sec	4sec
log. d	>24hour	52sec	>24hour	3. 5min	7sec	1. 8hour

表2 GP-CSP 与 graphplan、BLACKBOX 求解效率的比较^[28]

Problem	GP-CSP		graphplan		BLACKBOX	
	time	mem	time	mem	time	mem
log-rocket-a	9. 25sec	5M	68sec	61M	8. 88sec	70M
log-rocket-b	19. 42sec	5M	130sec	95M	11. 74sec	70M
log-a	16. 19sec	5M	1771sec	177M	7. 05sec	72M
log-b	2898sec	6M	787sec	80M	16. 13sec	79M
log-c	>3hour	-	>3hours	-	1190sec	84M

表1是 BLACKBOX 和 graphplan、SATPLAN 的比较,测试平台为194 MHz SGI Challenge server。Problem 列出一组基准规划问题(benchmark planning problem);其余各列分别给出 BLACKBOX、graphplan 和 SATPLAN 在不同算法(walksat 或 satz)下使用的时间,其中 SATPLAN 又将构建 CNF 和对 CNF 求解的时间分列。实验数据表明,BLACKBOX 的求解速度比其它两者提高了至少10倍。同时,SATPLAN 的时间主要消耗在构造 CNF 上,其模型搜索算法效率较高。这说明由规划图转换得到 CNF 是 BLACKBOX 成功的原因之一。

表2是 GP-CSP 和 graphplan、BLACKBOX 的比较,测试平台为 SUN Ultra5, 256M RAM machine。其中 GP-CSP 使用 GAC-CBJ 求模型;BLACKBOX 使用 satz 求模型。对每一种规划方法分列求解时间和空间消耗。实验数据表明:GP-CSP 的求解速度比 graphplan 提高10倍以上,比 BLACKBOX 稍逊,但它的空间消耗仅为 BLACKBOX 的十分之一。

SATPLAN(CSP)在构造 CNF(CSP 形式)时效率一般,而它们的模型搜索算法效率较高^[9]。Graphplan 则在构筑规划图时效率较高,而且规划图中包含的互斥信息对求模型的算法来说可以减少搜索空间,加快搜索效率。因此,一旦发现从规划图到 CNF(CSP 形式)的转换关系,就可以利用 graphplan 来构筑规划图,而后再将之转换为 CNF(CSP 形式)并利用通用的高效模型求解算法求出模型,从而获得相应规

划。

4 模型法和演绎法比较

在模型法求规划中,上层描述和底层算法相分离。因此,模型搜索算法的研究可以独立进行。现在已经找到许多高效的求模型算法,如 GSAT^[4],walksat^[5]。同时,模型法的上层描述能力较演绎法更强,具体表现在以下三方面:

(1)演绎法中描述并发动作比较困难;模型法中可以通过良好的定义支持同一时刻有多个动作执行,如 CSP。

(2)模型法求规划中可以很容易地描述不允许某个动作发生的状态。在搬箱子问题中,假定初始条件增加一条:第一个动作时不允许把箱 A 从箱 B 搬到箱 C。在演绎法中可以描述为式(16)。如将式(16)直接加入,那么若 $move(A, B, C, 1)$ 为真,则 $\neg move(A, B, C, 1)$ 为假,从而式(16)的左部为假。此时,可以推出任意值!可见,演绎法求规划比较适用于证真的情况而无法对否定情况给出简单恰当的描述。而在模型法求规划中(以 SATPLAN 为例),可以直接加入一个初始条件,如式(17)所示。

$$\exists x1, x2, y1, y2, z1, z2 (on(A, B, 1) \wedge on(B, TABLE, 1) \wedge on(C, TABLE, 1) \wedge clear(A, 1) \wedge clear(C, 1) \wedge \neg move(A, B, C, 1) \wedge move(x1, y1, z1, 1) \wedge move(x2, y2, z2, 2) \supset on(C, B, 3)) \quad (16)$$

$$on(A, B, 1) \wedge on(B, TABLE, 1) \wedge on(C, TABLE, 1) \wedge clear(A, 1) \wedge clear(C, 1) \wedge \neg move(A, B, C, 1) \wedge on(C, B, 3) \quad (17)$$

$$\forall i < N. \neg on(A, B, i) \vee \neg on(A, TABLE, i) \quad (18)$$

(3)模型法求规划有较强的描述中间状态的能力。在搬箱子问题中,假定希望在任意时刻箱 A 必须放在箱 B 或桌子上,我们可以直接引入对它的描述,如式(18)所示。在 STRIPS 系统中就很难加入这样的限制条件^[5]。因为这样的限制条件在搬箱子问题中是作为所求规划的一部分出现的,在证明过程中却不能将之加入要证明的公式中。

另一方面,演绎法求解时不需要指定步长,步长只是在深度遍历时记录的遍历深度,可以作为结束遍历的标志之一。而在模型法中需要指定规划的步长,若不指明,需要将之看作若干个不同步长的规划问题。例如在 CSP 中,假设步长为 k 来构造三元组(X, D, C),并对之求模型,若没有满足约束条件的模型,将步长 k 加1重新构造三元组。

5 模型法和 CBP 比较

Case-based Planning(以下简称 CBP)^[23,24]是求解规划的另一种方法:它根据已有的历史数据获得可能的规划。它的求解过程如下:将以往的规划求解情况作为历史数据统一进行管理;对于一个新的规划问题,首先将它按照既定的规则与历史数据相匹配,将匹配所得的规划问题的解作为当前问题的“解”;此时,需要判断所得的规划是否有效,若无效,则对它进行修正;求得规划之后,将这次求规划的结果也作为历史数据的一部分整理并存储下来。

和 CBP 相比,模型法求规划无法重用以往的求解信息。在模型法中通常从随机的状态开始搜索模型,因此初始的状态往往只能满足部分约束;在 CBP 中首先根据旧有的信息找到相近的规划,这个规划一般可以满足大部分的约束条件。在模型法中如果对规划问题描述的不够确切,比如缺少部分信息,就不能得到正确的规划;而 CBP 中却可能可以,因为它的历史数据可能对类似的规划问题有完整的描述,从而以之作为初始状态重用的规划可能是正确的。

但 CBP 存在它固有的问题。如何判断规划问题和历史数据中的哪些规划相匹配,以及如何通过对现有规划的修正得

到当前问题的规划是十分困难的:当历史数据增多后,如何快速找到合适的匹配成为难题,因为不当的匹配所得到的规划可能导致修正工作无法得到正确规划;对规划的修正则更加困难,在 CHARADE 中甚至将修正工作交由用户完成。

在模型法求规划中,可以通过求取所有模型的方式获得问题的全部规划,但在 CBP 中就不能保证求出所有的规划。而在效率问题上,CBP 对于某个问题的求解效率和它的历史数据密切相关,并且 CBP 对大多数常见问题有较高的效率^[24]。

小结 模型法作为通用规划求解的主要方案,发展至今已取得较大成果。同时,由于通用规划技术与定理证明、逻辑程序设计等研究方向都存在密切的联系,模型法的发展在一定程度上也有助于相关领域的发展。

模型法可能的研究方向有两个:其一,提高算法的求解效率,务求在较短的时间内求得有效的规划;其二,扩展系统的描述能力。以往的研究工作侧重于提高求解效率:或是寻找新的高效算法,如 GSAT^[4,14];或是利用不同表达形式在不同求解阶段的优势,如 BLACKBOX、GP-CSP、C-SATPLAN^[25]和 Petriplan^[11];也有一些扩展描述能力的尝试,如 Temporal Constraint^[18,22]涉及多个动作的并发执行和不同动作间相互影响的问题。下一步研究工作的侧重点可能如下:其一,进一步扩展系统的描述能力。在现实问题中动作或许有两种或两种以上可能的执行结果,如抛硬币。即使不同执行结果出现的概率是已知的,现有的方式仍然无法描述这样的动作;其二,在现实问题中,问题的描述往往是够完整的。而引入 CBP 的思想,利用以往求解信息,当问题相关信息不完整或不确定的时候,也可能求解模型。

参 考 文 献

- Nareyek A. AI Planning in a Constraint Programming Framework. In: Hommel, G. ed. Communication-Based Systems, Kluwer Academic Publishers. 163~178
- Mali A D, Kambhampati S. On the utility of Plan-space (Causal) Encodings. In: Proc. of AAAI, 1999. 557~563
- Blum A L, Frust M L. Fast Planning Through Planning Graph Analysis. In: Proc. of Artificial Intelligence 90, 1997. 281~300
- Selman B, Levesque H, Mitchell D. A New Method for Solving Hard Satisfiability Problem. In: Proc. of AAAI, 1992. 440~446
- Selman B, Kautz H, Cohen B. Noise strategies for Improving local search. In: Proc. of AAAI, 1994. 337~343
- Bonet B, Geffner H. Planning as Heuristic Search: New Results. In: Proc. of European Conf. on Planning 1999. 360~372
- Li C M, Anbulagan. Heuristics based on unit propagation for satisfiability problems. In: Proc. of IJCAI (1), 1997. 366~371
- Knoblock C A. An Analysis of ABSTRIPS. In: Proc. of AAAI, 1992. 126~135
- Weld D S. Recent Advances In AI Planning. In Proceeding of AI Magazine, 1999, 20(2): 93~123
- Mitchell D, Selman B, Levesque H. Hard and Easy Distributions of SAT Problems. In: Proc. of AAAI, 1992. 459~465
- Silva F, Castilho M A, Künzle L A. Petriplan: a new algorithm for plan generation (Preliminary report). In: Proc. of IBERAMIA-SBIA, 2000. 86~95
- Kautz H, Selman B. Planning as Satisfiability. In: Proc. of ECAI, 1992. 359~363
- Kautz H, Selman B. Pushing the Envelope: Planning, Propositional logic, and Stochastic Search. In: Proc. of AAAI 2, 1996. 1194~1201
- Kautz H, Selman B. Domain-Independent Extensions to GSAT: Solving Large Structured Satisfiability Problems. In: Proc. of IJCAI, 1993. 290~295
- Kautz H, Mcallester D, Selman B. Encoding Plans in Propositional Logic. In: Proc. of KR, 1996. 374~384
- Kautz H, Selman B. Unifying SAT-based and Graph-based Planning. In: Proc. of IJCAI, 1999. 318~325
- Kautz H, Walser J P. State-space Planning by Integer Optimization. In: Proc. of AAAI, 1999. 526~533
- Allen J F. Planning as Temporal Reasoning. In: Proc. of KR, 1991. 3~14
- Frank J, Jónsson A K, Morris P. On Reformulating Planning As Dynamic Constraint Satisfaction. In: Proc. of SARA, 2000. 271~280
- McCarthy J, Hayes P J. Some Philosophical Problem From the Standpoint of Artificial Intelligence. In: Proc. of Machine Intelligence 4, 1969. <http://www-formal.stanford.edu/jmc/mccha-y69.pdf>
- McCarthy J. Elaboration Tolerance. <http://www.formal.stanford.edu/jmc/elaboration.pdf>.
- Rives J, Sánchez J L, Pereira R. A Temporal Constraint Satisfaction Problem-Solver. <http://www.ilog.com/products/optimization/tech/research/indra.pdf>.
- Hammond K J. Case-based Planning: A Framework for Planning from Experience. In: Proceeding of Cognitive Science, 1990, 14 (3): 385~443
- Spalazzi L. A Survey on Case-Based Planning. In Proceeding of Artificial Intelligence Review, 2001, 16(1): 3~36
- Baiocchi M, Marcugini S, Milani A. C-SATPLAN: a SATPlan-based tool for planning with constraints. AIPS'98 Workshop on Planning as Combinatorial Search, Pittsburgh, USA, June 7, 1998. <http://www.dipmat.unipg.it/~milani/planning/publ/aips98CSAT.ps.gz>
- Cadoli M. A Survey of Complexity Results for Planning. In: Proc. of IPW, 1993. 131~145
- Ernst M D, Millstein T D, Weld D S. Automatic SAT-Compilation of Planning Problems. In: Proc. of IJCAI, 1997. 1169~1176
- Do M B, Kambhampati S. Planning as Constraint Satisfaction: Solving the planning graph by compiling it into CSP. In Proceeding of Artificial Intelligence, 2001, 132(2): 151~182
- Thagard P, Verbeurgt K. Coherence as Constraint Satisfaction. In Proceeding of Cognitive Science, 1998, 22(1): 1~24
- van Beek P, Chen Xinguang. CPlan: A Constraint Programming Approach to Planning. In: Proc. of AAAI, 1999. 585~590.
- Yang Qiang. Intelligent Planning. Springer-Verlag, 1997. ISBN 3-540-61901-1
- Bergmann R. Efficient Retrieval of Abstract Cases for Case-Based Planning. In: Proc. of SARA, 1998. 9~18
- Bergmann R, Wilke W. Building and Refining Abstract Planning Cases by Changing of Representation Language. In: Proc. of JAIR 3, 1995. 53~118
- Ruttkey Z. Constraint Satisfaction - a Survey. In: Proc. of CWI Quarterly 11, 1998. 123~161
- 戈也挺, 朱朝晖, 陈世福. 行动推理中若干问题的研究. 计算机科学, 2000, 27(3): 85~89