

# 可视化析取时态推理器的设计与实现\*

刘越畅<sup>1,2</sup>, 林晓骏<sup>3</sup>, 汤庸<sup>2</sup>

1. 嘉应学院计算机学院, 广东 梅州 514015;
2. 华南师范大学计算机学院, 广东 广州 510631;
3. 华南理工大学计算机学院, 广东 广州 510006)

**摘要:** 时态推理是人工智能领域的重要研究方向。析取时态问题 (disjunctive temporal problem, DTP) 是一类表达力极强的定量时态模型, 被广泛应用于智能规划和调度领域的时态建模, 近年来吸引了研究人员的关注。提出 DTP 弱蕴含性和弱演化析取时态网络 (weakly evolutionary disjunctive temporal network, WEDTN) 的定义, 基于 WEDTN 设计和实现了可视化 DTP 求解器 TRSE。系统演示发现基于 WEDTN 的可视化比常见的基于搜索树的可视化更能有利于人们直观理解 DTP 的求解算法过程、挖掘和发现问题的结构特征对求解过程的影响。

**关键词:** 人工智能; 时态推理; 析取时态问题; 约束可满足问题; 弱演化析取时态网络

中图分类号: TP181 文献标志码: A 文章编号: 0529-6579 (2013) 06-0038-06

## Design and Implementation of a Visualisable Disjunctive Temporal Problem Solver

LIU Yuechang<sup>1,2</sup>, LIN Xiaojun<sup>3</sup>, TANG Yong<sup>2</sup>

1. School of Computer Science, Jiaying University, Meizhou 514015, China;
2. School of Computer Science, South China Normal University, Guangzhou 510631, China;
3. School of Computer Science, South China University of Technology, Guangzhou 510006, China)

**Abstract:** Temporal reasoning is an important research direction of Artificial Intelligence (AI). Disjunctive temporal problem (DTP) is an expressive quantitative temporal model which is widely used in automated planning and scheduling and is therefore being well studied. The definition of weak implication and weakly evolutionary disjunctive temporal network (WEDTN) is proposed. Based on the WEDTN structure a visual DTP solver TRSE is designed. The demonstration reveals that the visualization based on WEDTN better facilitates the understanding and discovery of DTP solving algorithm and the impact of structural information on that than usual search tree based visualization.

**Key words:** artificial intelligence; temporal reasoning; disjunctive temporal problem (DTP); constraint satisfaction problem (CSP); weakly evolutionary disjunctive temporal network (WEDTN)

人工智能规划 (或称自动规划: automated planning) 是人工智能领域一个重要的研究子领域。经典的智能规划依赖于几个理想化假设<sup>[1]</sup>, 如: 动作的效果是确定的, 系统是静态、完全可观察的, 动作是瞬时发生的 (动作没有持续时间), 等

等。随着解决实际问题的需要和智能规划技术研究的深入, 这些理想化假设不再是必需的。例如, 现实中的动作或者活动极少是瞬间发生的, 往往具有一定的持续时间; 另外, 动作发生的前提条件不一定非得在动作开始执行时成立, 同时动作的效果也

\* 收稿日期: 2013-07-02

基金项目: 国家自然科学基金资助项目 (61272067, 41172028); 广东省教育厅科技创新资助项目 (2012KJCX0097); 梅州市科技计划资助项目 (梅市科 2012 (14) 号); 广东省自然科学基金团队研究资助项目 (S2012030006242)

作者简介: 刘越畅 (1979 年生), 男; 研究方向: 时态表示与推理、智能规划与调度; E-mail: ycliu@jyu.edu.cn

不一定都发生在动作完成之时。再如，系统的初始条件和目标命题也可以赋以时间上的意义：初始条件命题可以发生在某个时间段内，而目标命题之间也可以有时间上的关系（时态扩展目标），等等。此外，引入时间元素还可以对经典规划进行进一步的扩展，如引入外部事件等。时态规划便是经典智能规划对命题和动作在时间的维度上进行扩展的产物。析取时态问题（Disjunctive Temporal Problem, DTP）是 Kostas Stergiou 和 Manolis Koubarakis 提出的基于时间点的定量时态模型<sup>[2]</sup>。使用 DTP 作为时态规划中的时态模型的优势是可以将规划阶段的决策推迟到调度（时态推理）阶段解决，这样平衡两方面计算量的同时提高了规划灵活性<sup>[3]</sup>。自 DTP 问题提出以来，研究者陆续提出了 DTP 求解算法（如局部搜索 DTP 算法<sup>[4]</sup>、基于电路编码的 SAT 算法<sup>[5]</sup>）、基于拓扑信息的启发式技术<sup>[6]</sup>，以及 DTP 问题的扩展（如动态 DTP<sup>[7]</sup>、基于偏好的 DTP<sup>[8]</sup>、多智能体 DTP<sup>[9]</sup>）。DTP 丰富的表达力和应用潜力使得人们对它的研究持续升温，短短时间内国际著名的人工智能研究杂志《人工智能》已有多篇专门研究 DTP 的论文诞生。

基于结构信息在 DTP 问题求解中的重要性，在析取时态网络（Disjunctive Temporal Network, DTN）<sup>[10]</sup>的基础上，本文提出了 DTP 弱蕴含性和弱演化析取时态网络（Weakly Evolutional DTN, WEDTN），设计了基于 WEDTN 的可视化 DTP 求解器 TRSE（Temporal Reasoning Solution Environment）。系统演示可以看出，与常用的基于搜索树的可视化求解系统不同，基于 WEDTN 的 DTP 求解器可视化更有利于人们直观了解 DTP 的结构特征及其与求解过程的影响。

本文组织如下：第 1 节首先给出必要的问题定义；第 2 节介绍 DTP 问题求解的基本算法及 TRSE 系统设计；第 3 节是系统演示部分；第 4 节介绍相关结论和未来工作。

### 1 问题定义

**定义 1** 一个简单时态问题（Simple Temporal Problem, STP）<sup>[11]</sup>是一个二元组  $\langle X, C \rangle$ ，其中： $X = \{x_1, x_2, \dots, x_n\}$  称为时态变量集合； $C = \{c_1, c_2, \dots, c_m\}$  是简单时态约束集合，每个  $c_i$  是形如  $x_i - x_j \leq c$  ( $x_i, x_j \in X, c \in \mathbf{Z}$ ) 的二元约束<sup>1</sup>。

由定义可知，一个 STP 与一个所有变量定义在整数域上的约束满足问题（Constraint Satisfaction Problem, CSP）是等价的。正如 CSP 可以使用约束

图来表示那样，STP 也可以使用时态约束网络来表示，下面给出具体定义。

**定义 2** 给定一个 STP  $\langle X, C \rangle$ ，其简单时态网络是一个有向边标号图  $\langle V, E, W \rangle$ ，其中： $V = \{v_i \mid x_i \in X\}$ ， $E = \{ \langle v_i, v_j \rangle \mid x_i - x_j \leq c \in C \}$ ， $W$  是一个权重函数  $W: E \rightarrow \mathbf{Z}$ ，使得：任意  $x_i - x_j \leq c \in C$  则  $W(\langle v_i, v_j \rangle) = c$ 。

**定义 3** 一个析取时态问题是一个二元组  $P = \langle X, C \rangle$ ，其中： $X = \{x_1, x_2, \dots, x_n\}$  是一个变量集合； $C = \{C_1, C_2, \dots, C_m\}$  是一个约束集合，每个  $C_i$  是一个形如  $c_{i1} \vee c_{i2} \vee \dots \vee c_{ik}$  的析取式（也称为子句），可以表示为集合  $\{c_{i1}, c_{i2}, \dots, c_{ik}\}$ ，其中的  $c_{ij}$  是一个形如  $x_u - x_v \leq c$  的简单时态约束 ( $x_u, x_v \in X; x_u, x_v, c \in \mathbf{Z}$ )。

**定义 4** 给定一个析取时态问题  $P = \langle X, C \rangle$ ，它相应的析取时态网络是一个五元组  $N = \langle V(N), E(N), L, W, PRJ \rangle$ ，其中： $V(N) = \{v_1, v_2, \dots, v_n\}$  是顶点集合； $E(N) = \{ \langle v_i, v_j, k, w \rangle \mid v_i, v_j \in V(N), k \in L, w \in W \}$  是一个四元组边集合； $L = \{k \mid C_k \in C\}$  是一个标号集合； $W$  是一个权重符号集合； $PRJ$  是个映射函数，使得： $PRJ(x_i) = v_i$  ( $x_i \in X$ )， $PRJ(c_k = x_i - x_j \leq c) = \langle v_i, v_j, k, c \rangle$ 。

由 DTN 的定义可以看出，一个 DTN 实际上是一个有向、多重标号图。

**性质 1** 给定一个 DTP  $P = \langle X, C \rangle$ ，其中： $X = \{x_1, x_2, \dots, x_n\}$ ， $C = \{C_1, C_2, \dots, C_m\}$ ， $P$  是一致的，当且仅当存在一个一致的 STP， $\langle X, C' \rangle$  ( $C' = \{c_1, c_2, \dots, c_m\}, c_i \in C_i$ )。这样的 STP 通常称为析取时态问题的解 STP。

图 1 给出了 DTP 及其 DTN 的示例<sup>[10]</sup>。

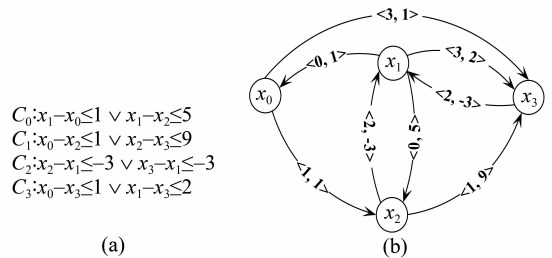


图 1 DTP 及 DTN 示例

Fig. 1 DTP and DTN

**性质 2** 给定 DTP  $P = \langle X, C \rangle$  和  $P' = \langle X', C' \rangle$ ，称  $P'$  弱蕴含  $P$ ，如果  $X = X'$ ， $C' \subset C$ ，且  $P'$  是一致的  $\Rightarrow P$  是一致的。DTN 之间的弱蕴含关系等价于它们相应的 DTP 的弱蕴含关系。

**定义 5** 给定一个初始 DTN,  $N_0$ , 和另一个 DTN,  $N_1$ ,  $N_1$  是  $N_0$  关于边  $e$  的弱演化 DTN (Weakly Evolutional DTN, WEDTN), 如果  $N_1 = N_0 - \{e' \mid N_0(L)(e') = N_0(L)(e)\}$  且  $N_1$  弱蕴含  $N_0$ .

图 2 是弱演化 DTN 序列的示意图。

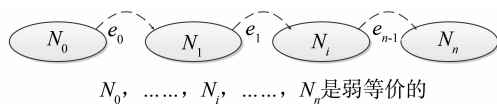


图 2 弱演化 DTN

Fig. 2 Weakly evolutionary DTN

弱演化 DTN 实际上考察了在搜索过程中 DTN 随着变量赋值的进行对原变量未取值从原 DTN 中删除后的演变过程。也就是说, DTP 求解问题变为寻找一个原问题 DTN 的弱演化 DTN 的解 STP 的过程。

## 2 算法及系统设计

### 2.1 DTP 一致性判定算法

根据性质 1, 一个 DTP 是否是一致的, 可以应用搜索算法查找解 STP 来求解。此时相当于将原问题看作一个 CSP: 每个变量对应于 DTP 中的每个析取约束, 而变量的取值范围是该析取约束中的所有析取肢 (即简单时态约束); 而约束则要求所有变量的取值构成一个一致的简单时态问题 (即解 STP)。

图 3 给出了该 CSP 求解算法的大体框架<sup>[12]</sup>。该算法每次选择一个变量 (第 2 行) 进行尝试赋值, 若当前赋值与当前部分解一致 (第 4 行) 则进行下一轮迭代 (第 5 行); 否则尝试其它的值。

#### procedure NaiveDTP

**Input:** DTP  $P = \langle X, C \rangle$ ,  $A$ —partial solution,  $U$ —unassigned meta-variables

**Output:** complete solution, or failure

1. **if**  $U = \emptyset$  **then return**  $A$ ;
2.  $C_x = \text{select\_var}(P, A, U)$ ;
3. **for** every  $c_{xi} \in \text{Dom}(C_x)$
4.     **if** consistent( $A \cup \{C_x = c_{xi}\}$ ) **then**
5.         result = NaiveDTP( $P, A \cup \{C_x = c_{xi}\}, U - \{C_x\}$ );
6.         **if** result = failure **then continue**;
7.         **else return** result;
8. **return** failure;

图 3 DTP 求解搜索算法

Fig. 3 Search algorithm for DTP solution

### 2.2 启发式 (Heuristics) 技术

DTP 的求解与 CSP 的求解一样, 实际应用中必须使用一些启发式技术, 否则必定遭遇状态爆炸的问题。当前研究文献中提出的求解 DTP 的启发式技术主要有前向检查技术、最少剩余值变量排序、语义分枝、被吸收变量移除、递增前向检查技术、基于拓扑的变量、值选择策略等。以下对这些技术的基本思想作简要介绍。

**2.2.1 前向检查 (Forward Check, FC)** 前向检查技术的核心思想在于: 在尝试赋值之前, 首先从未赋值元变量的取值域中删除那些与当前部分解不一致的元值。这种技术的优点在于可以在进行下一轮迭代之前首先发现搜索的死点 (如果前向检查过程中某个元变量的取值域为空), 从而可以提前回溯, 避免进行下一轮迭代。

**2.2.2 最少剩余值 (Minimal Remaining Values, MRV)** 最少剩余值策略的主要思想在于: 在当前节点选择变量 (FC\_ DTP 中的 select\_ var 函数) 进行当前赋值的时候, 优先选择那些具有最少候选元值的元变量。MRV 策略是一种最常使用的基于“失败优先 (FF)”原则<sup>[13]</sup>的变量选择策略。与前向检查技术配合使用时, 由于元变量的取值域随着搜索的进行不断地发生变化, 因此 MRV 实现了动态的变量选择功能。实验证明 FC 与 MRV 的组合具有显著的剪枝能力<sup>[14]</sup>。

**2.2.3 语义分枝策略 (Semantic Branching, SB)** 语义分枝策略也是 DTP 求解中的一个剪枝能力很强的启发式技术。它的核心思想是: 在针对某个变量赋值的搜索节点, 若一个值 ( $c_{ij}: x - y \leq c$ ) 尝试后到达一个死点, 该值将被抛弃并将尝试其它的值; 此时, 根据该不等式的语义, 在当前 DTP 和部分解下,  $x - y \leq c$  为假意味着  $\sim (x - y \leq c)$  即  $x - y > c$  为真, 在时态变量取值为整数的情况下有  $y - x \leq -c - 1$  为真, 将  $y - x \leq -c - 1$  显式地加入到当前部分解中并将该约束传播不会改变原问题的一致性, 并将有助于更早地剪枝。

**2.2.4 被吸收变量移除技术 (Removal of Subsumed Values, RSV)** 被吸收变量移除技术的思想是: 假设当前部分解 STP 的距离图中, 有  $x$ 、 $y$  之间的最短距离  $\text{distance}(x, y) = c$ , 如果对于某个未赋值的元变量  $C_k$  其取值域中具有一个候选元值  $c_{kj}: y - x \leq c'$  且有  $c \leq c'$ , 则称  $C_k$  被该解 STP 所吸收; 此时可直接将  $c_{kj}: y - x \leq c'$  赋值给  $C_k$  而不需要进行一致性判定和尝试其它的值 (因为此时  $c_{kj}: y - x \leq c'$  的加入不会使最短距离发生任何

变化，因此它与当前部分解 STP 必然是一致的。

2.2.5 基于拓扑的变量和值选择策略 (Topology-based Variable/Value Ordering, TVO) TVO 的基本思想是：从 DTN 中计算每条边的冲突系数，通过多种组合函数，计算每个变量的冲突系数。根据冲突系数和“失败优先”(Fail-First) 原则，优先选择那些冲突系数较大的变量和冲突系数较小的值。虽然同样是变量选择策略，实验证明 TVO 比 MRV 具有更强的剪枝能力<sup>[6]</sup>。

### 2.3 系统设计

TRSE 系统架构如图 4 所示。其中：Script 模块主要是 DTP 问题的表示脚本。该脚本传递给系统的 Model (建模) 模块，生成问题的内部表示。Model 模块将问题信息分别与 Engine 和 Visulisation 模块交互，完成问题的求解及可视化。以下分别阐述各模块的设计。

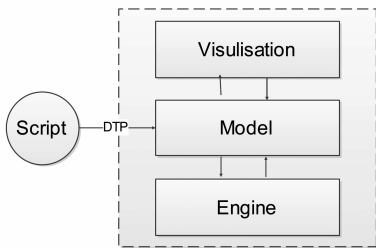


图 4 TRSE 系统架构  
Fig. 4 TRSE architecture

2.3.1 建模 (Model) 和引擎 (Engine) 模块 将 DTP 问题放在 CSP 的框架下求解最大的优势是，可以使用大量现有的 CSP 研究成果应用到 DTP 的求解中。其次，应用现有优秀的 CSP 求解框架能够使整个求解器结构清晰，易于使用和维护。为了达到这个目的，本文使用 Gecode 作为建模和求解引擎。Gecode 是瑞典皇家理工学院 Christian Schulte 开发的开放、免费、可移植、易使用、有效率的用于开发基于约束系统及应用的环境约束求解框架，它提供一个模块化和可拓展的约束求解器。图 5 展示了 Gecode 的系统架构<sup>[15]</sup>。

从 Gecode 架构设计可见，除了提供预先定义的常见类型变量（整数、实数、集合类型）及相关约束，Gecode 还提供了丰富的自定义编程 (Programming) 功能，使用户可以根据自己的需要对用户领域的约束满足问题进行灵活建模。对 DTP 的可视化约束满足建模要对其中的变量、传播器和分枝器、搜索引擎分别进行自定义编程。由于约束建

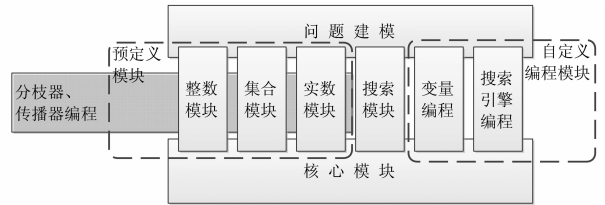


图 5 Gecode 系统架构  
Fig. 5 Gecode architecture

模可以有多种灵活的方式，对同一个 DTP，Gecode 同样提供多种建模方法。在本系统中，为方便起见，我们仍然应用整数变量类型处理。使用 Map 数据结构将每个整数值与相应的简单时态约束联系起来。

根据这一架构，在 TRSE 系统中设计了以下的分枝、传播器。

#### 1) 分枝 (Brancher) 策略。

根据以上对应用的启发式策略分析，其中的 MRV 和 TVO 分别可以作为不同的分枝器。连同最简单的变量随机选择方法，系统中设计了三种分枝器：BaseBrancher、MRVBrancher、TVOBrancher。需要注意的是，这三种分枝器是互斥的，即每次约束求解进行的时候，只能应用其中一种分枝器。

其结构如图 6。

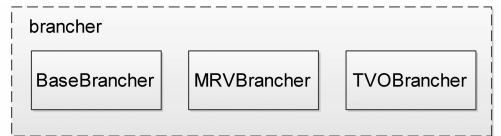


图 6 分枝器类设计  
Fig. 6 Brancher classes design

#### 2) 传播器 (Propagator)。

在 Gecode 框架中，传播器可以对两种元素进行描述：一种是约束，以判定当前值选择的一致性；另一种是约束的传播，以进行搜索的剪枝。本系统设计了以下传播器：FWPropgat/DPCPropgat/PC3Propgat 是第一种传播器，用以判断当前部分解 STP 的一致性（分别对应 FloydWarshall 算法、DPC 算法<sup>[11]</sup>、P3C 算法<sup>[16]</sup>），这三个有且仅有一个在运行中起作用；FCPropgat/RSVPropgat/SBPropgat 是第二种传播器，对应于前面描述的三种启发式：前向检查、被吸收变量移除、语义分枝，此三种传播器可以在约束求解前任意选择是否在求解中应用。传播器的设计如图 7 所示。

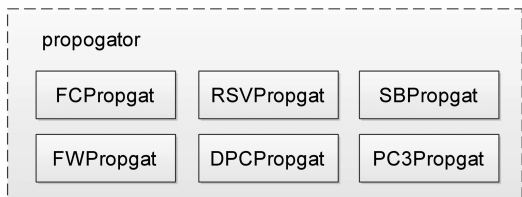


图 7 传播器类设计

Fig. 7 Propagator class design

### ③ 引擎编程 (Engine programming)

通过分枝器和传播器, 利用 Gecode 缺省的算法引擎已经能够处理大多数的约束求解及应用。在 TRSE 中, 可视化模块要求跟踪引擎的运行过程, 以便将求解过程 (主要是变量赋值信息) 进行图形化展示。因此, 需要对引擎进行变量选择并赋值后, 将该信息传递给可视化模块进行图形展示。这主要在 Status () 函数中调用 choice () 函数获得。

图 8 展示了以上设计的分枝器和传播器交替执行的迭代过程。一个基于 CSP 的 DTP 求解过程主要就是分枝和传播的迭代过程, 在这个过程中, 首先进行分枝 (即: 变量和值选择), 然后根据分枝结果进行约束的传播为下一次分枝做准备。此过程也考虑了多个分枝器和传播器的相容性问题: MRVBrancher、TVOBrancher 和 BaseBrancher 每次需要且只能选择其中之一, FCPropgat 默认使用, RSV 和 SB 传播器可同时使用, 用于 STP 简单约束传播的 P3C、DPC、FW 传播器次需要且只能选择其中之一。

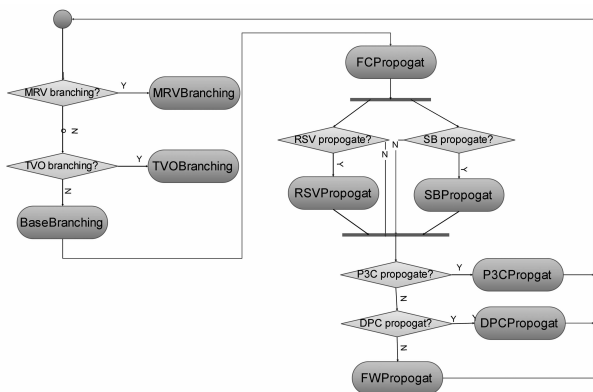


图 8 分枝器和传播器交替执行过程

Fig. 8 Iterative execution of branchers and propagators

2.3.2 可视化 (visualisation) 模块 与 Gecode 现有的 Gist 可视化模块 (仅表达搜索过程中节点选择的搜索树) 不同, TRSE 中的可视化模块使用

DTN 表达问题的拓扑结构, 并在此结构上动态展示搜索中的动态值选择过程。内部图结构使用 JGraphT<sup>[17]</sup>, 而图形展示使用开源的 JUNG 库<sup>[18]</sup>实现。

## 3 系统实现及演示

图 9 是 TRSE 系统界面。该系统分为四大模块。一是输入模块, 可以通过文本框键盘输入、选择一个 DTP 文件或者选择目录顺序求解该目录下所有 DTP 文件三种方式输入求解对象。二是选项设置模块, 用于进行求解前的设置选项 (使用的分枝器、传播器、超时设置、可视化设置)。三是可视化显示模块, 用于通过演化 DTN 动态展示求解过程。四是结果输出模块, 展示 DTP 求解结果。

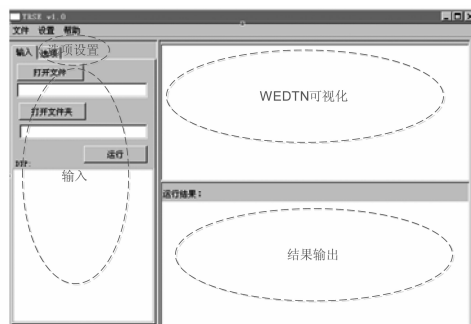


图 9 TRSE 主界面

Fig. 9 TRSE graphical user interface

示例: 假设输入 DTP:  $P = \langle X, C \rangle$ ,  $X = \{x_1, x_2, x_3, x_4, x_5\}$ ,  $C = \{C_1, C_2, C_3\}$ 。  $C_1 \equiv x_1 - x_2 \leq 2 \vee x_3 - x_2 \leq -1$ ;  $C_2 \equiv x_2 - x_3 \leq 2 \vee x_1 - x_2 \leq 4$ ;  $C_3 \equiv x_2 - x_1 \leq -3 \vee x_3 - x_1 \leq -5$ 。

选择 BaseBrancher 分枝器, 不选择除 FCPropgat 之外任何传播器进行问题求解, 可视化模块将按图 10 形式动态显示 DTN 演化过程。

## 4 结语

由于析取时态问题在智能规划和调度领域中广泛得到应用, 近年来成为该领域的研究热点, 已产生了多篇国际《人工智能》杂志上发表的研究论文。本文阐述了一个以 DTP 为模型的通用时态推理系统 TRSE 的设计和实现。TRSE 的设计充分采用软件设计模式和基于演化 DTN 的可视化设计, 运用现有的高效 CSP 求解器 Gecode 和图可视化库 JUNG, 将重点放在各种启发式技术的设计和动态可视化上。虽然 DTP 的理论研究已经取得了较大进展, 目前尚未有此类通用推理系统的研发, 基于

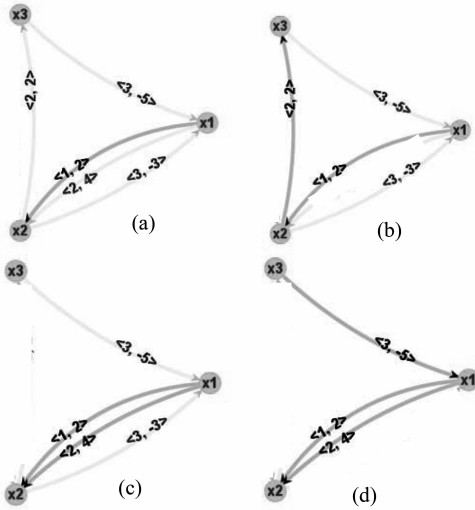


图 10 DTN 弱演化过程((a)→(b)→(c)→(d))

Fig. 10 Weak evolution process((a)→(b)→(c)→(d))

WEDTN 的可视化技术解释了 DTP 求解中的结构特性, 对于 DTP 乃至时态推理的学习、研究有重要的价值。进一步的工作将扩展对其他 DTP 求解技术的集成和已探索 DTN 和未探索 DTN 空间<sup>[12]</sup>等更细致的结构特征的展示支持。

#### 参考文献:

[1] GHALLAB M, NAU D, TRAVERSO P. Automated planning: theory and practice [M]. San Francisco: Morgan Kaufmann Publishers, 2004.

[2] STERGIIOU K, KOUBARAKIS M. Backtracking algorithms for disjunctions of temporal constraints [J]. Artificial Intelligence, 2000, 120(1): 81–117.

[3] LIU Y, FANG Y. Boost the Integration of planning and scheduling: a heuristics approach [J]. Procedia Engineering, 2012, 29: 3348–3352.

[4] MOFFITT M D, POLLACK M E. Applying local search to disjunctive temporal problems [C]//IJCAI, 2005: 242–247.

[5] BLAINE NELSON T K S K. CircuitTSAT: a solver for large instances of the disjunctive temporal problem [C]//The Eighteenth International Conference on Automated Planning and Scheduling, ICAPS 2008, Sydney, Australia, 2008: 232–239.

[6] LIU Y, JIANG Y, QIAN H. Topology-based variable ordering strategy for solving disjunctive temporal problems [C]//In 15th International Symposium on Temporal Representation and Reasoning, 2008. TIME '08, 2008: 129–136.

[7] SCHWARTZ PETER J, POLLACK MARTHA E. Two approaches to semi-dynamic disjunctive temporal problems [C]//ICAPS Workshop on Constraint Programming for Planning and Scheduling, Monterey, California, USA. June 2005.

[8] MOFFITT M D. On the modelling and optimization of preferences in constraint-based temporal reasoning [J]. Artificial Intelligence, 2011, 175(7/8): 1390–1409.

[9] BOERKOEEL JR J C, DURFEE E H. A distributed approach to summarizing spaces of multiagent schedules [C]//In Proceedings of the 26th National Conference on Artificial Intelligence (AAAI–12), Toronto, Canada, 2012: 1742–1748.

[10] 刘越畅, 姜云飞, 钱红. 基于问题结构的启发式策略在析取时态问题求解中的应用 [J]. 计算机研究与发展, 2008, 45(11): 1840–1849.

[11] DECHTER R, MEIRI I, PEARL J. Temporal constraint networks [J]. Artificial Intelligence, 1991, 49: 61–95.

[12] 刘越畅. 基于约束的时态推理和时态规划 [D]. 广州: 中山大学, 2008: 75–99.

[13] HARALICK R M, ELLIOTT G L. Increasing tree search efficiency for constraint satisfaction problems [J]. Artificial Intelligence, 1980, 14: 26–31.

[14] TSAMARDINOS IOANNIS, POLLACK MARTHA E. Efficient solution techniques for disjunctive temporal reasoning problems [J]. Artificial Intelligence, 2003, 151(1/2): 43–89.

[15] SCHULTE C, TACK G, LAGERKVIST M Z. Modeling and programming with Gecode [EB/OL]. [2010] <http://www.gecode.org/doc/4.2.0/MPG.pdf>.

[16] PLANKEN L, WEERDT M D, KROGT ROMAN VAN DER. P3c: a new algorithm for the simple temporal problem [C]//In Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 2008), 2008: 256–263.

[17] JgraphT. A free Java graph library [EB/OL]. [2013] <http://jgraph.org>.

[18] JUNG. Java universal network/graph framework [EB/OL]. [2013] <http://jung.sourceforge.net>.