

doi: 10.3969/j.issn.1006-1576.2012.01.007

基于点图的作战任务时间协同一致性检验方法

姚剑, 朱延广, 曹星平, 王维平

(国防科学技术大学信息系统与管理学院, 长沙 410073)

摘要: 针对作战计划制定过程中任务时间协同一致性的问题, 提出基于点图推断并校验时间的方法。构建描述作战任务时间协同关系的点图模型, 通过顶点聚合运算、顶点折叠运算和顶点时间运算 3 种基本运算进行检验, 给出了作战任务时间协同一致性检验方法, 并结合测试案例对方法的有效性进行验证。验证结果表明, 该方法能有效地检验作战任务时间协同的一致性。

关键词: 作战任务; 时间协同一致性; 点图; 时间推理

中图分类号: TJ06 **文献标志码:** A

Verify the Operational Task Temporal Collaboration Consistency Based on Point Graph

Yao Jian, Zhu Yanguang, Cao Xingping, Wang Weiping

(School of Information System & Management, National University of Defense Technology, Changsha 410073, China)

Abstract: The task of verifying the operational task temporal collaboration consistency in military planning is researched based on the point-graph (PG), and a method based on PG to cope with temporal reasoning is designed. Build a PG model to describe operational task temporal collaboration, verify the operational task temporal by point-unification, point-folding and point-time calculation, and a consistency verifying algorithm is given. Finally, an example is present to illustrate the feasibility and effectiveness of above method.

Key words: operational task; temporal collaboration consistency; point graph; temporal reasoning

0 引言

作战计划制定过程中, 军事决策人员经常需要指定作战任务之间的时间协同关系^[1]。时间协同是以时间作为基本参数, 通过对时间的规定来实现作战任务在特定时刻协调一致, 控制各种相关任务互不干扰、连续、紧凑、有序地进行。因此, 对作战任务的时间协同一致性进行检验^[2]是作战计划制定过程中需要解决的问题。

作战任务时间协同一致性分析是一类典型的时间推理问题, Allen 提出的时段时序逻辑 (interval temporal logic, ITL)^[3]被广泛应用于时间推理的时间信息表达, 而后 Allen 又进一步扩充了时间点与时段关系, 提出点-时段时序逻辑 (point-interval temporal logic, PITL) 并详细地描述了 PITL 使用的基本时序结构。传统 PITL 中时间点和时段信息是独立存在的, 因此 Zaidi 等人将时间的点描述与时段描述结合起来, 对点 PITL 方法进行了扩展^[4]。扩展的 PITL 在时间点与时段的时序关系描述方面具有很大的伸缩性, 既可以分别进行定性和定量时间

推理, 也可以将二者结合起来。Zaidi 等人基于图论提出了一种点图 (point graph, PG)^[5]用于描述 PITL。时间推理引擎是基于点图推断并校验时间系统中时段与时间点所呈现的时序关系。因此, 笔者基于点图给出作战任务时间协同一致性检验算法, 并通过一个测试案例对其有效性进行验证。

1 点图模型介绍

1.1 点图的数学描述

点图是 Zaidi 等人在 Allen 的时段时序逻辑理论基础上, 建立的一种用于描述任务之间时序关系的加权有向图^[3]。标准的点图可以表示为以下四元组:

$$PG = \langle V, E, T, D \rangle$$

其中: V 表示顶点的集合, $\forall v \in V$ 表示时间轴上的一个时间点; E 表示有向边的集合, $\forall e_{ij} \in E$ 表示时间点 v_i 和 v_j 之间的先后关系为 $v_i < v_j$; T 表示顶点对应的的时间取值集合, 表示为 $T = \text{Time}[v]$, 如果存在顶点 $v \in V$ 的时间取值未知, 那么表示为 $\text{Time}[v] = \text{null}$; D 表示有向边对应的的时间间隔取值集

收稿日期: 2011-08-13; 修回日期: 2011-09-13

基金项目: 面向体系对抗效能仿真优化的认知演化算法研究 (61074107); 基于因果关系建模的应急方案有效性评估与优化方法 (91024015)

作者简介: 姚剑 (1988—), 男, 硕士, 从事体系工程与体系仿真研究。

合, 表示为 $D=length[e]$, 显然 $Length[e_{ij}] = Time[v_j] - Time[v_i]$ 。如果存在有向边 $e \in E$ 对应的时间间隔未知, 那么表示为 $Length[e] = null$ 。

点图中出度大于 1 的顶点称为分支顶点, 入度大于 1 的顶点称为汇合顶点。

一个典型点图如图 1。

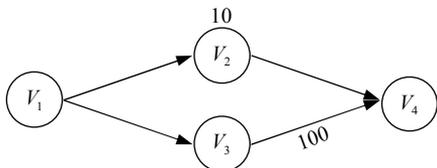


图 1 点图示例

如图 1 所示, 顶点 v_2 的时间取值为 10, 其余顶点的时间取值为 $null$; 顶点 v_3 和 v_4 的时间间隔取值为 100, 其余顶点间的时间间隔为 $null$ 。顶点 v_1 为分支顶点, v_2 和 v_3 称为分支顶点 v_1 的子节点; 顶点 v_4 为汇合顶点, v_2 和 v_3 称为汇合顶点 v_4 的父节点。

时序关系一致性判定标准为^[5]:

1) 任意顶点 $v \in V$ 对应的时间取值 $Time[v]$ 是唯一的。任意边 $e \in E$ 的长度 $Length[e]$ 是唯一的。

2) 任意 2 个顶点 v_i 和 v_j 之间的对应的时间关系由于数据不足无法判断或者根据已知数据可推理确定为 $Time[v_i] = Time[v_j]$ 、 $Time[v_i] < Time[v_j]$ 和 $Time[v_i] > Time[v_j]$ 3 种关系之一。

1.2 点图的基本运算

1.2.1 顶点聚合运算

1) 单顶点聚合

对于任意 $v_i, v_j \in V$, 如果 $Time[v_i] = Time[v_j]$ 或 $Length[e_{ij}] = Length[e_{ji}] = 0$, 那么 v_i 和 v_j 可以合并为复合顶点 $P = [v_i; v_j]$, 并且称 v_i 和 v_j 包含于复合节点 P , 记为 $v_i \in P, v_j \in P$ 。

2) 复合顶点聚合

对于任意复合顶点 $P_i = [v_{i_1}; v_{i_2}; \dots; v_{i_n}]$ 和复合顶点 $P_j = [v_{j_1}; v_{j_2}; \dots; v_{j_m}]$, 如果存在 $v_i \in P_i$ 和 $v_j \in P_j$ 满足 $Time[v_i] = Time[v_j]$ 或 $Length[e_{ij}] = Length[e_{ji}] = 0$, 那么 P_i 和 P_j 可以合并为复合顶点 $P = [P_i; P_j]$ 。

1.2.2 顶点折叠运算

1) 分支顶点折叠

设顶点 $v_i, v_j, v_k \in V$, v_j 和 v_k 是 v_i 的子顶点, 且

$Length[e_{ij}] \neq null$, $Length[e_{ik}] \neq null$, 那么 v_i 为可折叠的分支顶点。

如果 $Length[e_{ij}] < Length[e_{ik}]$, 那么删除点图中的有向边 e_{ik} ; 如果存在 e_{jk} , 并且 $Length[e_{jk}] = null$, 那么 $Length[e_{jk}] = Length[e_{ij}] - Length[e_{ik}]$, 如果 $Length[e_{jk}] \neq null \wedge Length[e_{jk}] \neq Length[e_{ij}] - Length[e_{ik}]$ 顶点折叠运算发现错误, $Length[e_{jk}]$ 取值不唯一; 如果不存在边 e_{jk} , 那么在点图上添加边 e_{jk} , 令 $Length[e_{jk}] = Length[e_{ij}] - Length[e_{ik}]$ 。

如果 $Length[e_{ij}] = Length[e_{ik}]$, 并且点图中不存在 v_k 和 v_j 之间的有向边, 或者存在有向边并且两点间的时间间隔为 $null$, 那么 v_k 和 v_j 之合并为复合顶点 $P = [v_k; v_j]$, 边 e_{ik} 和 e_{ij} 在点图上合并为边 e_{iP} , 令 $Length[e_{iP}] = Length[e_{ik}] = Length[e_{ij}]$, 否则分支顶点折叠无法进行, 顶点折叠运算发现错误。

2) 汇合顶点折叠

设顶点 $v_i, v_j, v_k \in V$, v_j 和 v_k 是 v_i 的父顶点。

$Length[e_{ji}] \neq null$, $Length[e_{ki}] \neq null$, 那么 v_i 为可折叠的汇合顶点。

如果 $Length[e_{ji}] < Length[e_{ki}]$, 那么删除点图中的有向边 e_{ki} ; 如果存在 e_{kj} , 并且 $Length[e_{kj}] = null$, 那么 $Length[e_{kj}] = Length[e_{ki}] - Length[e_{ji}]$, 如果 $Length[e_{kj}] \neq null \wedge Length[e_{kj}] \neq Length[e_{ki}] - Length[e_{ji}]$ 顶点折叠运算发现错误, $Length[e_{kj}]$ 取值不唯一; 如果不存在边 e_{kj} , 那么在点图上添加边 e_{kj} , 令 $Length[e_{kj}] = Length[e_{ki}] - Length[e_{ji}]$ 。

如果 $Length[e_{ji}] = Length[e_{ki}]$, 并且点图中不存在 v_k 和 v_j 之间的有向边, 或者存在有向边并且两点间的时间间隔为 $null$, 那么 v_k 和 v_j 合并为复合顶点 $P = [v_k; v_j]$, 边 e_{ki} 和 e_{ji} 在点图上合并为边 e_{Pi} , 令 $Length[e_{Pi}] = Length[e_{ki}] = Length[e_{ji}]$ 。否则分支顶点折叠无法进行, 顶点折叠运算发现错误。

1.2.3 顶点时间运算

对于任意 $v_i, v_j \in V$, $e_{ij} \in E$, 如果已给定 $Time[v_i]$ 、 $Time[v_j]$ 和 $Length[e_{ij}]$ 3 个数据之中任意 2 个, 则根据公式 $Length[e_{ij}] = Time[v_j] - Time[v_i]$ 计算另外一个数据。

如果 $Time[v_i]$ 、 $Time[v_j]$ 和 $Length[e_{ij}]$ 3 个数据都

已给定, 那么判断 $\text{Length}[e_{ij}] = \text{Time}[v_j] - \text{Time}[v_i]$ 是否成立, 若等式不成立, 说明 $\text{Length}[e_{ij}]$ 取值不唯一, 时序关系不满足一致性。

2 作战任务时间协同一致性检验方法

2.1 作战任务时间协同关系表示

定义 1: 作战任务的时间表示

作战任务的时间表示为 $\text{task} = [ts, te]$, 其中 ts 和 te 分别表示任务的开始和结束时间。如果任务的开始时间未知, 那么 $ts = \text{null}$; 如果任务的结束时间未知, 那么 $te = \text{null}$ 。

定义 2: 作战任务的定性关系

对于作战任务 $\text{task}_1 = [ts_1, te_1]$ 和 $\text{task}_2 = [ts_2, te_2]$, 两者之间的定性关系表示为 $\langle \text{task}_1, \text{task}_2 \rangle$, 其中 $\langle \text{task}_1, \text{task}_2 \rangle$ 属于关系集合 R 。

$$R = \{ \text{before}, \text{meet}, \text{overlaps}, \text{starts}, \text{during}, \text{finishes}, \text{equals} \}$$

作战任务之间的定性关系描述如表 1。

表 1 作战任务之间的定性关系描述

关系	描述
$\langle \text{task}_1, \text{task}_2 \rangle = \text{before}$	$te_1 < ts_2$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{meet}$	$te_1 = ts_2$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{overlaps}$	$ts_1 < ts_2; ts_2 < te_1; te_1 < te_2$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{starts}$	$ts_1 = ts_2; te_1 < te_2$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{during}$	$ts_1 > ts_2; te_1 < te_2$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{finishes}$	$ts_1 > ts_2; te_1 = te_2$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{equals}$	$ts_1 = ts_2; te_1 = te_2$

定义 3: 作战任务的定量关系

对于作战任务 $\text{task}_1 = [ts_1, te_1]$ 和 $\text{task}_2 = [ts_2, te_2]$, 两者之间的定性关系为 $\langle \text{task}_1, \text{task}_2 \rangle$, 那么两者之间的定量关系表示为 $\text{Value}[\langle \text{task}_1, \text{task}_2 \rangle] = [u_{ss}, u_{se}, u_{es}, u_{ee}]$ 。

其中, $u_{ss} = ts_2 - ts_1$, $u_{se} = ts_2 - te_1$, $u_{es} = te_2 - ts_1$, $u_{ee} = te_2 - te_1$ 。如果 $u_{ss}, u_{se}, u_{es}, u_{ee} = \text{null}$, 那么分别表示对应的取值未知。

表 1 中作战任务之间的定性关系, 可以定量地表示为表 2。

表 2 作战任务之间的定量关系描述

关系	描述
$\langle \text{task}_1, \text{task}_2 \rangle = \text{before}$	$[\text{null}, u_{se}, \text{null}, \text{null}]$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{meet}$	$[\text{null}, 0, \text{null}, \text{null}]$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{overlaps}$	$[u_{ss}, -u_{se}, \text{null}, u_{ee}]$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{starts}$	$[0, \text{null}, \text{null}, u_{ee}]$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{during}$	$[-u_{ss}, \text{null}, \text{null}, u_{ee}]$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{finishes}$	$[-u_{ss}, \text{null}, \text{null}, 0]$
$\langle \text{task}_1, \text{task}_2 \rangle = \text{equals}$	$[0, \text{null}, \text{null}, 0]$

2.2 作战任务时间协同关系点图建模

n 个作战任务之间的时间协同关系可以表示为一个点图模型 $PG_{\text{task}} = \langle V_t, E_t, T_t, D_t \rangle$ 。

其中, V_t 表示顶点的集合, 任意 $v_k \in V_t, v_{n+k} \in V_t$ 分别表示作战任务 k 的开始和结束时间; E_t 表示表示有向边的集合, 对于任意作战任务 k , 显然 $e_{k, n+k} \in E_t$; T_t 表示顶点对应的的时间取值集合, 如果作战任务 k 的开始或者结束时间已知, 那么 $\text{Time}[v_k]$ 或者 $\text{Time}[v_{n+k}]$ 的取值存在, 否则为 null ; D_t 表示有向边对应的的时间点间隔集合, 如果作战任务 k 的开始和结束时间已知, 那么 $\text{Length}[e_{k, n+k}]$ 的取值存在, 否则为 null 。

对于任意 2 个作战任务 task_i 和 task_j 。

如果 $\langle \text{task}_i, \text{task}_j \rangle = \text{before}$, 那么 $e_{n+i, j} \in E_t$, 并且

$$\text{Length}[e_{n+i, j}] = u_{se};$$

如果 $\langle \text{task}_i, \text{task}_j \rangle = \text{meet}$, 那么顶点 v_{n+i} 和 v_j 聚合为 $[v_{n+i}; v_j]$;

如果 $\langle \text{task}_i, \text{task}_j \rangle = \text{overlaps}$, 那么 $e_{i, j} \in E_t$, $e_{j, n+i} \in E_t$, $e_{n+i, n+j} \in E_t$, 并且 $\text{Length}[e_{i, j}] = u_{ss}$, $\text{Length}[e_{j, n+i}] = -u_{se}$, $\text{Length}[e_{n+i, n+j}] = u_{ee}$;

如果 $\langle \text{task}_i, \text{task}_j \rangle = \text{starts}$, 那么顶点 v_i 和 v_j 聚合为 $[v_i; v_j]$, $e_{n+i, n+j} \in E_t$, 并且 $\text{Length}[e_{n+i, n+j}] = u_{ee}$;

如果 $\langle \text{task}_i, \text{task}_j \rangle = \text{during}$, 那么 $e_{j, i} \in E_t$, $e_{n+i, n+j} \in E_t$, 并且 $\text{Length}[e_{j, i}] = -u_{ss}$, $\text{Length}[e_{n+i, n+j}] = u_{ee}$;

如果 $\langle \text{task}_i, \text{task}_j \rangle = \text{finishes}$, 那么顶点 v_{n+i} 和 v_{n+j} 聚合为 $[v_{n+i}; v_{n+j}]$, $e_{j, i} \in E_t$, 并且 $\text{Length}[e_{j, i}] = -u_{ss}$;

如果 $\langle \text{task}_i, \text{task}_j \rangle = \text{equals}$, 那么顶点 v_i 和 v_j 聚合为 $[v_i; v_j]$, 顶点 v_{n+i} 和 v_{n+j} 聚合为 $[v_{n+i}; v_{n+j}]$ 。

2.3 时间协同一致性检验算法

作战任务时间协同关系的不一致情况及其在点图模型的表现形式:

1) 任务的给定数据不一致

作战任务的时间数据包括开始时刻、结束时刻和持续时间, 在计划中多次被赋值且数值不同, 则作战计划必然存在错误。

在点图模型表现为顶点时间运算或顶点折叠运算出现错误, 即根据已给定的数据进行运算发现存

在某些边的长度值不唯一, 从而时序关系不满足一致性标准。

2) 任务之间时序逻辑错误

作战任务之间在计划中出现时间逻辑错误, 计划中的两个时刻经过计算推理在时间轴上先后顺序不一致。

可采用 S-Invariant^[6]检测点图中环路, 如果点图中存在环路, 在环路中至少存在 2 个顶点 v_1 和 v_2 , 在数值可推断同时满足不等式 $\text{Time}[v_i] < \text{Time}[v_j]$ 和 $\text{Time}[v_i] > \text{Time}[v_j]$, 故时序关系不满足一致性。

3) 作战计划时序关系无可行解

作战计划中的某些时序关系在逻辑上成立, 但在实际数值上无法满足, 如图 2。

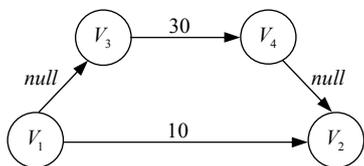


图 2 不存在可行解的点图路径

可采用 FPSO 和 FPSI 算法^[7]搜索点图中的路径, 建立路径等式并进行判断。如图 2, 分支顶点 v_1 和汇合顶点 v_2 存在 2 条路径, 建立路径等式为:

$\text{Length}[e_{12}] = \text{Length}[e_{13}] + \text{Length}[e_{34}] + \text{Length}[e_{42}]$, 根据已给定的数据 $\text{Length}[e_{12}] = 10$, $\text{Length}[e_{34}] = 30$ 可得 $\text{Length}[e_{13}] + \text{Length}[e_{42}] = -20$, 故 $\text{Length}[e_{13}]$ 和 $\text{Length}[e_{42}]$ 不存在可行解, 故时序关系不满足一致性。

基于以上 3 个不一致判别准则, 给出时间协同一致性检验算法如表 3。

表 3 时序关系一致性检验算法

算法 1: 时序关系一致性检验算法
1: 建立作战行动时序关系的点图模型;
2: 顶点集合 V 进行聚合运算;
3: 点图环路检测, 如果环路存在, 那么时序关系不满足一致性, 终止算法, 否则进入第 4 步;
4: 搜索点图中符合时间运算条件的顶点集合 V_c , 对任意 $v \in V_c$ 进行时间运算, 如果运算发现时序关系不满足一致性, 终止算法, 否则重复搜索和运算直到 $V_c = \emptyset$, 进入第 5 步;
5: 搜索点图中可折叠的顶点集合 V_f , 对任意 $v \in V_f$ 进行折叠运算, 如果运算发现错误, 则时序关系不满足一致性, 终止算法, 否则重复搜索和运算直到 $V_f = \emptyset$, 进入第 6 步;
6: 点图环路检测, 如果存在环路, 那么时序关系不满足一致性, 终止算法, 否则进入第 7 步;
7: 搜索点图中分支顶点和汇合顶点之间的所有路径, 建立路径等式, 如果存在路径等式无可行解则时序关系不满足一致性, 否则时序关系满足一致性。算法终止。

算法结果:

1) 如果任务时间协同关系满足一致性, 在数据充足时, 点图最终简化为一条直线。数据不足时, 结果为存在分支顶点和汇合顶点的点图并且点图上所有的路径等式存在可行解。

2) 如果任务时间协同关系存在不一致, 算法检测出不一致时序关系, 点图运算停止在发现不一致关系的状态。

3 测试案例

下面以机群空袭敌方导弹基地^[8]为例, 对算法进行演示和验证。

作战任务的基本信息如表 4。

表 4 作战任务的基本属性

任务编号	持续时间	描述
$task_1$	null	预警机对空域预警
$task_2$	30	干扰和掩护编队 1 飞赴干扰空域
$task_3$	null	干扰编队对敌电磁空间进行干扰
$task_4$	25	攻击和掩护编队 2 飞赴突防空域
$task_5$	10	攻击编队突防攻击敌导弹基地
$task_6$	30	攻击、掩护和干扰编队集合返航

任务时间协同的基本原则是空袭全程在预警机指挥下进行; 攻击编队和掩护编队在干扰编队对敌实施电磁干扰后进行突防攻击; 干扰编队返航时间在其他编队完成攻击任务之后。

基于上述时间协同原则, 指定作战任务之间的时间协同关系如表 5。

表 5 作战任务的时序关系

时序关系的定性描述	时序关系的定量描述
$\langle task_1, task_2 \rangle = starts$	$[0, null, null, null]$
$\langle task_2, task_4 \rangle = overlaps$	$[10, null, null, null]$
$\langle task_2, task_3 \rangle = meet$	$[null, 0, null, null]$
$\langle task_5, task_3 \rangle = finishes$	$[-5, null, null, 0]$
$\langle task_3, task_6 \rangle = before$	$[null, 5, null, null]$
$\langle task_6, task_1 \rangle = finishes$	$[null, null, null, 0]$
$\langle task_4, task_5 \rangle = before$	$[null, 3, null, null]$

根据表 5 所示的作战任务间的时序关系, 建立时序关系的点图模型如图 3。

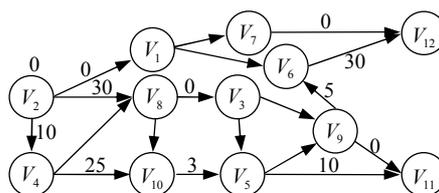


图 3 作战任务时序关系的点图表示

时序关系一致性检验算法对图 3 所示的点图进行检测得到图 4。

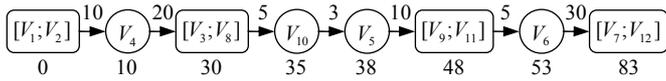


图 4 对图 3 进行一致性检测的结果点图

如图 4 所示, 检测算法的结果: 任务时序点图模型图 3 最终化简为一条直线。

上述检测算法的结果说明表 4 和表 5 所描述的作战任务时序满足一致性。

下面在表 4 和表 5 描述的任务属性和时序关系基础上添加若干语句, 以演示算法对不一致信息的检测能力。

1) 在表 5 描述的作战任务时序关系中增加一个定性关系 $(task_5, task_3) = before$, 则初始的时序关系点图模型如图 5。

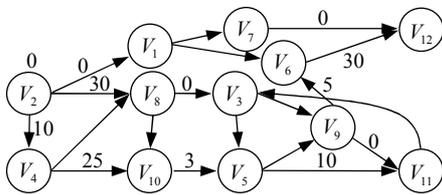


图 5 增加定性关系的点图

对图 5 进行一致性检验, 算法完成顶点聚合运算后, 发现节点 $[V_3;V_8]$ 、 $[V_9;V_{11}]$ 、 $[V_{10}]$ 和 $[V_5]$ 组成环路如图 6 所示, 说明指定任务关系出现时序逻辑错误, 不满足一致性。

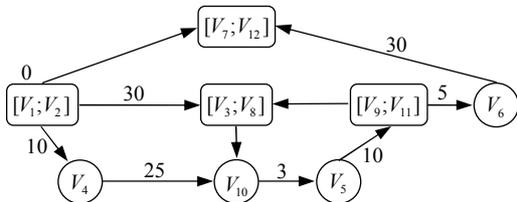


图 6 存在环路的点图

2) 在表 4 描述的作战任务基本属性中增加作战任务 $task_3$ 的持续时间为 15, 则初始时序关系的点图模型如图 7。

对图 7 进行一致性检验, 分支顶点 V_{10} 折叠运算由于 $Length[V_{10}, V_9; V_{11}] \neq Length[V_{10}, V_5] + Length[V_5, V_9; V_{11}]$ 无法进行, 算法发现错误而终止, 如图 8, 说明任务的给定数据无法满足一致性。

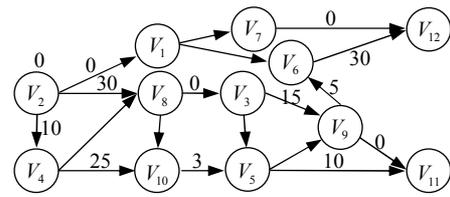


图 7 增加持续时间的点图

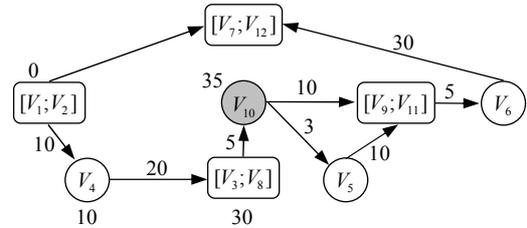


图 8 折叠运算发现错误

4 结束语

测试案例的算法演示和分析表明, 作战任务时间协同一致性检验方法能够有效地检验作战任务时间协同的一致性。

参考文献:

- [1] 曹淑信. 信息火力战[M]. 北京: 国防大学出版社, 2006.
- [2] 汤罗浩, 朱承, 罗江锋. 基于 STN 的时间冲突检测与消解[J]. 计算机工程, 2011, 37(3): 251-253, 256.
- [3] Allen J F, Ferguson G. Actions and events in interval temporal logic [J]. Journal Logic and Computation, 1994, 4(5): 531-579.
- [4] Abbas K. Zaidi, Lee W. Wagenhals. Planning temporal events using point-interval logic [J]. Mathematical and Computer Modeling, 2006(43): 1229-1253.
- [5] Abbas K. Zaidi, Mashhood Ishaque, Alexander H. Levis. Project Management Using Point Graphs[J]. Systems Engineering, 2009, 12(1): 36-54.
- [6] Wilson P M H. Some Geometry and Combinatorics for the S-Invariant of Ternary Cubics [J]. Experimental Mathematics, 2006, 15(4): 479-490.
- [7] Abbas K. Zaidi, Alexander H. Levis. Rule Decomposition and Validation for Distributed Decision Making[C]. Washington DC: Proc. 1st Int'l Symposium on Command and Control Research and Technology, 1995: 210-217.
- [8] 胡文远. 空地对抗仿真系统研究[D]. 北京: 北京航空航天大学, 1999.