Nemesis，一款图上的逃脱游戏

Pierre Bergé 格勒诺布尔阿尔卑斯大学、法国国家科学研究中心、
格勒诺布尔国立理工学院、LIG实验室，法国格勒诺布尔38000

Antoine Dailly
克莱蒙奥弗涅大学、法国农业、食品与环境研究院、TSCF研究组，法国克莱蒙费朗63000

Yan Gerard[1]
克莱蒙奥弗涅大学、法国国家科学研究中心、圣艾蒂安矿业学院、克莱蒙奥弗涅国立高等工程师学院、LIMOS实验室，法国克莱蒙费朗63000

arXiv:2601.13841v1 [cs.DS] 20 Jan 2026

摘要

我们定义了一种新的图上的逃脱游戏，并称之为*Nemesis*。游戏在一个图上进行，其中一部分顶点被标记为出口，两名玩家中的一名（称为逃亡者）的目标是到达其中一个出口顶点。另一名玩家即逃亡者的对手，称为*Nemesis*。她的目标是将逃亡者困在一个不包含任何出口的连通分量中。每回合，逃亡者从一个顶点移动到一个相邻顶点。随后，Nemesis 删除图中任意位置的一条边。当逃亡者到达出口，或身处不包含任何出口的连通分量时，游戏结束。对于树和最大度不超过 3 的图，Nemesis 可以在线性时间内求解。我们还证明，游戏的一个变体——Blizzard（其中只能删除与逃亡者当前位置相邻的边）——也可以在线性时间内求解。对于任意图，我们证明 Nemesis 是 PSPACE 完全的，并且在平面多重图上是 NP 困难的。我们将结果推广到相关的赶猫问题，并证明其 PSPACE 完全性。我们还证明，寻找一种基于满二叉逃脱树且其叶子为出口的策略是 NP 完全的。

2012年ACM主题分类 计算理论 → 动态图算法

关键词和短语 图、躲避与追捕博弈、PSPACE 完全性、量化 SAT、加拿大旅行者问题、猫群管理问题

数字对象唯一标识符 10.4230/LIPIcs...N

致谢 感谢 Claire Hilaire 和 Gaëtan Berthe，感谢他们与我们进行讨论并为这项工作带来了各种不同的想法。

1 引言

涅墨西斯逃脱游戏

过去二十年来，密室逃脱游戏已成为许多国家广受欢迎的娱乐形式。这类合作博弈中，参与者或团队通过解开一系列谜题，尝试逃出房间。本文提出了一个类似的框架，但场景设定在图环境中。我们将一个试图从图中逃脱的个体称为逃亡者。一些顶点被标记为出口，逃亡者的目标是到达其中任意一个。

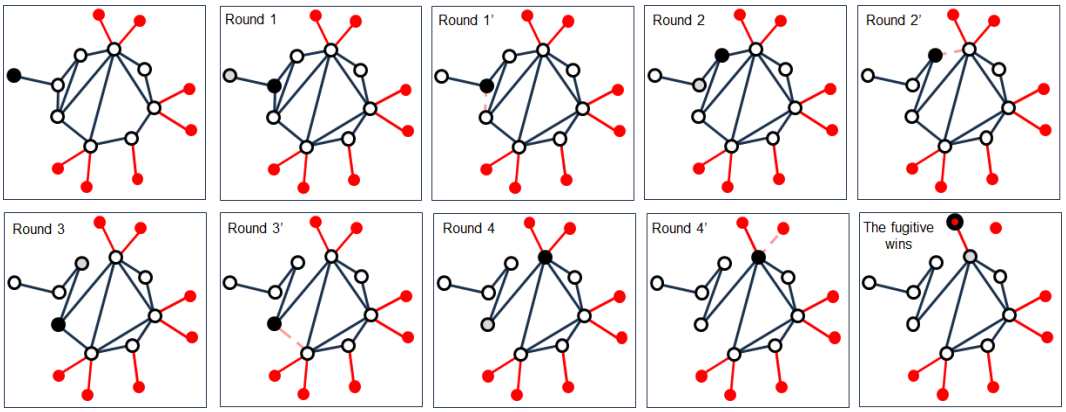
在博弈的每个回合中，逃亡者都会沿着一条边，从当前顶点移动到一个相邻顶点。如果他在任何回合成功抵达出口，就赢得博弈。在静态图中，简单的广度优先搜索（BFS）可以计算逃亡者的起始顶点到出口的最短路径，从而得出逃脱所需的最少回合数。

[1] 通讯作者

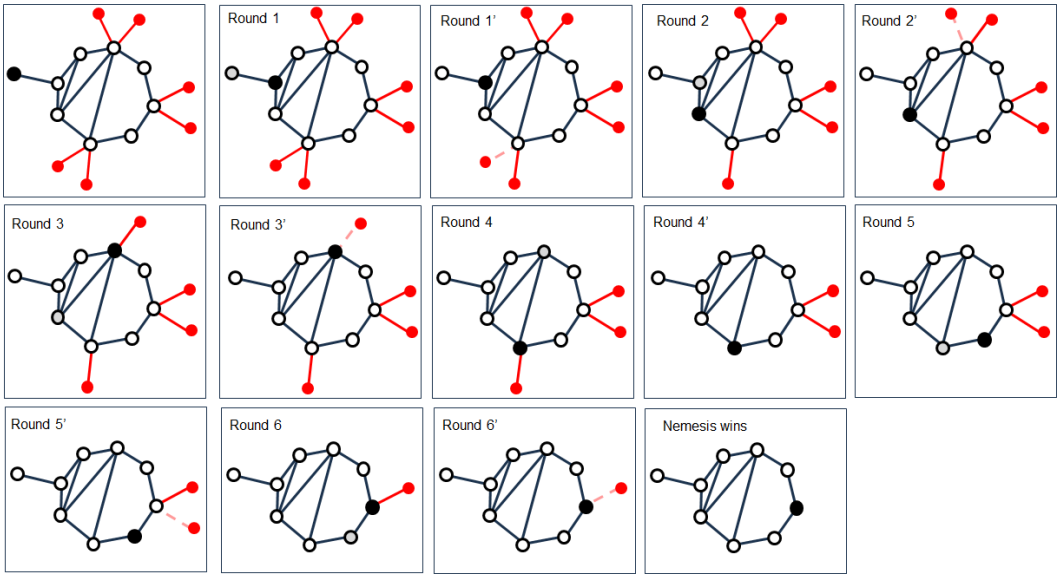
© Pierre Bergé、Antoine Dailly、Yan Gerard；依据知识共享署名 4.0 国际许可协议授权 编辑：John Q. Open 和 Joan R. Access；文章编号 N；页码 N:1–N:26 莱布尼茨国际信息学会议论文集 达格施图尔城堡—莱布尼茨信息学中心，达格施图尔出版社，德国
LIPICS

N:2 Nemesis，一款图上的逃脱游戏

这个博弈源于引入了一位拥有改变图的魔力的逃亡者对手。现在，我们定义一种在图上进行的新型双人逃脱游戏，并将其称为 *Nemesis*[2]。***Nemesis***[3] 是逃亡者对手的名字。Nemesis 以非常有限且自然的方式改变图的结构。每个回合中，在逃亡者移动之后，Nemesis 都会从图中永久移除恰好一条边。被删除的边可以位于图中的任何位置。如果 Nemesis 成功切断逃亡者所在位置与所有出口之间的所有路径，将他永远困在图中，她就赢得博弈。当逃亡者抵达出口，或 Nemesis 将他困在一个不含任何出口的连通分量中时，博弈结束（图 1 展示了两种对局）。



(a) 在上述博弈中，逃亡者获胜。



(b) 在上述博弈中，Nemesis 获胜。

图1 Nemesis 的两场博弈。出口是红色顶点。在每个回合中，位置由黑色顶点表示的逃亡者沿一条边移动，而 Nemesis 移除一条边。

尽管这一设定很简单，而且尽管人们已经研究过许多图上的追逃或阻断

[2] 在希腊神话中，Nemesis 是代表正义之怒与神罚的女神。[3] 我们在 *Nemesis* 前加上冠词 *the*，用来指代逃亡者对手；而不带冠词的 NEMESIS 则用作该博弈的名称

博弈，Nemesis 仍是一种自然的逃脱游戏，据我们所知，此前从未有人研究过。不过，Nemesis 与若干已有深入研究的问题密切相关。

相关游戏

关于图上的追捕游戏有大量文献，尤其是经典的警察与强盗问题 [27, 1, 23, 18, 10, 13], ，该问题已被证明是 PSPACE 完全问题 [24]。警察与强盗博弈和 Nemesis 之间有两个主要区别。首先，警察与强盗问题中没有出口顶点。其次，追捕者在图上移动，而在我们的设定中，Nemesis 则是删除边。

Nemesis 与守卫问题的关系更为密切，后者同样是 PSPACE 完全问题 [17, 19]。在守卫问题中，输入是一张图，其中一个顶点子集定义了一个由固定数量守卫防守的保护区域。守卫的目标是阻止入侵者进入该保护区域。如果将入侵者替换为逃亡者，并将保护区域替换为出口，那么这个框架与 Nemesis 非常相似。不过，关键区别在于，守卫在图的顶点上移动，而在我们的简单博弈中，Nemesis 会不可逆地删除边。

另一个相关问题是约翰·康威提出的天使与魔鬼问题：天使在无限二维网格上移动，而魔鬼则试图通过逐步移除（或“吃掉”）顶点来困住它 [8, 14]。已有研究表明，如果天使每回合都能移动至少距离 2 [20, 25, 29], ，天使便能获胜；此外，人们还根据天使和魔鬼各自的能力研究了若干变体 [12] ，以及在三维网格等其他图上的情形 [21, 9]。本着在图中困住移动中的玩家这一理念，赶猫问题最近在 [4] 中提出，并在 [5, 6] 中进行了研究。赶猫人并非要在无限网格中捕捉天使，而是试图困住一只在有限图中恶意游荡的猫。这个问题与 Nemesis 相似：赶猫人每次删除一条边，而猫在每回合从一条边移动到另一条边。两者的显著区别在于，猫没有可供逃脱的出口；这对于一只小猫来说或许并不令人意外，但对于一只屋顶上的猫而言却很不幸。出口顶点的缺失使赶猫问题从根本上区别于 Nemesis。在赶猫问题中，猫的目标是尽可能长时间保持自由，而赶猫人则试图尽快将其困住。关于 Nemesis 的研究成果还带来了一个推论：赶猫问题是 PSPACE 完全的。我们还注意到，2012年推出的商业桌游《无处可去》 [7], 中，两名玩家试图通过在有限的六边形网格中移除边来困住对手。

Nemesis 也可以视为一种连通性游戏，与 Hex 或广义地理游戏相近，因为一名玩家（逃亡者）试图在避开 Nemesis 删除的边的同时，连接自己的初始顶点与一个出口。不同之处在于，逃亡者必须沿着一条单一路径前进，因此其选择比不受路径结构约束的顶点选择游戏更受限制。已知许多连通性游戏都是 PSPACE 完全，例如一般图上的 Hex [16] 、平面图上的 Hex [28], ，以及 Havannah、Twixt 和 Slither [11]。

最后一个密切相关的问题是加拿大旅行者问题，由 Papadimitriou 和 Yannakakis 于 1991 [26] 提出。在这个在线问题中，旅行者希望从初始位置 s 前往已知图上的一个目标顶点 t ，但每条边都可能被雪阻断，而且旅行者只有到达边的某个端点时，才能得知该边是否可通行。设计良好策略这一任务可以建模为旅行者与风暴之间的博弈。旅行者在图中的移动方式与我们逃脱游戏中的逃亡者完全相同，而暴风雪则扮演对手的角色，决定与旅行者当前所在顶点相邻的边是否可通行。旅行者的目标是使其

行走的总长度相对于最优可用路径保持在给定的竞争比范围内。Papadimitriou 和 Yannakakis 证明了加拿大旅行者问题是 PSPACE 完全的 [26]。加拿大旅行者问题与 Nemesis 的第一个区别在于，加拿大旅行者的目标是在到达出口时保持有效的竞争比。相比之下，逃亡者的目标只是到达一个出口，因此竞争比的概念与 Nemesis 无关。另一个区别在于对手的能力：Nemesis 可以移除图中任意位置的一条边，而风暴只能影响旅行者首次到达某个位置时与该位置相连的边。最后一个区别是，雪可以在一个回合中移除多条边，而 Nemesis 一次只能移除一条边。

作为加拿大旅行者问题的一个变体，我们引入了 Nemesis 的一种变体，并将其称为 Blizzard。此时，Nemesis 化作一场暴风雪，而逃亡者则是一名寻找避难所的捕兽者。这个变体与加拿大旅行者问题的设定相似：捕兽者试图逃到避难所，而每个回合，暴风雪都会删除一条与逃亡者当前位置相邻的边（在 Nemesis 中，任何边都可以被删除；而在 Blizzard 中，我们增加了这一限制：只能删除与捕兽者相邻的边）。如果捕兽者注定会在雪中冻死，暴风雪便获胜，因为所有通往避难所的路径都已无法通行。我们证明，这一对 Nemesis 力量的限制使 Blizzard 的求解变得容易得多。

结果

Nemesis 和 Blizzard 都是零和博弈。我们要回答的核心问题是：给定逃亡者（或捕兽者）的起始位置，哪位玩家拥有获胜策略。求解该博弈，就是判定哪位玩家能够从给定的起始顶点强制获胜。

表 1 概述了我们关于 Nemesis（以及 Blizzard）的结果。这些结果分为两类：多项式时间可解情形，以及通过从计算困难问题归约而得到的困难性结果。

贯穿全文的一个关键概念是二叉逃逸树。二叉逃逸树是一棵满二叉树，其中根的度为 2，所有内部顶点的度为 3，所有叶子都是出口。逃亡者起始位置的邻域中存在二叉逃逸树，足以保证逃亡者拥有获胜策略。然而，这一条件并非普遍适用：有些实例中，即使附近不存在任何二叉逃逸树，逃亡者仍有获胜策略。

我们首先证明，对于树以及最大度为 3 的图（不含重边），二叉逃逸树的存在既是逃亡者获胜的必要条件，也是充分条件。此外，在这两类图中，均可在线性时间内检测出二叉逃逸树。因此，Nemesis 在这些图上可以高效求解。

▶**定理 1.** 对于树和最大度至多为3的图，*Nemesis* 可以在线性时间内求解。

同理，对于任意图，Blizzard 也可以使用简单图遍历算法在线性时间内求解。

▶**定理 2.** *Blizzard* 可以在线性时间内求解。

接下来，我们转而讨论困难性结果，并确立三个否定性结果。首先，我们证明，寻找二叉逃逸树的策略（它足以但并非必要地证明逃亡者拥有获胜策略）本身就是计算困难的。

	树	$\Delta \leq 3$	平面图	任意图
简单图	线性（定理 1）	线性（定理 1）	?	PSPACE 完全（定理 5 +定理 9）
多图	?	?	NP 困难(定理 4)	PSPACE 完全(定理 5)

Tab表 1 Nemesis 复杂度结果摘要 esis。

▶**定理 3**。 给定一个带有一组出口的图和一个顶点 r，判断 r 是否为二叉逃逸树的根是 *NP* 完全的，即使该图的最大度为 4。

现在，我们介绍本文的主要困难性结果，其中包括关于 Nemesis 的两项复杂度结果。首先，我们证明，在平面多重图上，该博弈是计算困难的。

▶**定理 4.** 对于平面多重图，*Nemesis* 是 *NP* 困难的。

然而，和双人游戏的常见情形一样，即使将 Nemesis 限定在平面多重图上，也尚不清楚它是否属于 NP。因此，NP 完全性并非这一问题的恰当复杂度概念。事实上，Nemesis 在平面图上的确切复杂度（即使不含重边）仍未解决，而且似乎是一个颇具挑战性的问题。

定理 4 的证明依赖于对 SAT 的一个平面变体进行归约。遗憾的是，目前我们无法将经典的 PSPACE 完全问题量化 SAT（QSAT）归约到平面多重图上的 Nemesis，因此无法证明这一受限情形下的 PSPACE 完全性。

尽管如此，我们给出了从 QSAT 到任意多重图上的 Nemesis 的归约。为模拟量词交替所需的额外构件本质上是非平面的。因此，我们只能证明任意多重图上的 PSPACE 完全性；经过进一步论证，还可证明任意简单图上的 PSPACE 完全性。

▶**定理 5.** *Nemesis* 是 *PSPACE* 完全问题。

作为定理 5 的推论，我们得到了赶猫问题的复杂度。

▶**推论 6.** 赶猫问题是 *PSPACE* 完全问题。

本文结构如下。第 2 节将对这些博弈进行形式化定义，并给出示例和基本观察。第 3 节讨论可在多项式时间内求解 Nemesis 的图类。最后，第 4 节介绍我们的困难性结果。

2 问题陈述与初步观察

我们首先对 Nemesis 进行形式化定义，并就简单情形以及出口数量相关的问题提出初步说明。

2.1 游戏

我们在图 $G = (V, E)$ 或多重图上定义 Nemesis 博弈（其中 E 是边的多重集）。部分顶点被指定为出口，其余顶点则称为常规顶点。

N:6 Nemesis，图上的逃脱游戏

Nemesis 是一种双人零和博弈。玩家1是逃亡者，玩家2是$Nemesis$。这是一种非对称博弈：逃亡者沿图移动，而 Nemesis 移除边。双方的目标也不同。逃亡者到达出口顶点即获胜，而 Nemesis 若能将逃亡者隔离在一个不含出口的连通分量中，则获胜。

博弈的一个实例由一个图或多重图 G、一组指定的出口以及一个起始顶点 s 组成，后者表示逃亡者的初始位置。不失一般性，我们可以假设出口构成一个独立集，因为出口之间的边在博弈中不起作用。

博弈按回合进行。每一回合中，逃亡者首先从当前顶点 x 移动到一个相邻顶点 y，经过一条仍存在于 E中的边 $(x,\ y)$。然后，Nemesis 从 E中移除一条边。对于多重图，如果一条边 e 的重数为$r \geq 1$，移除一条 e 会使其重数降至 $r-1$。如果 $r-1=0$，这条边就会从图中消失；否则，逃亡者仍可使用它。用博弈论的术语来说，可以把边看作具有若干生命值 （HP）。

自然可以定义博弈的若干变体。例如，Nemesis 可以在逃亡者进行第一次移动之前移除一条边；或者两名玩家同时行动，彼此都不知道对方的行动。无论具体的时序规则如何，这些变体都密切相关。本文始终采用以下约定：逃亡者先行动，从而触怒众神并使 Nemesis 开始行动。

每回合都会移除一条边，因此回合数不超过边数，博弈是有限的。Nemesis 具有以下两个基本性质。

▶ 备注 7. 如果逃亡者有获胜策略，那么他有一种获胜策略，且其中不会两次经过同一个顶点。事实上，假设某个获胜策略在第 i 回合和第 j回合之间包含一个环路。由于第 j 回合后剩余的边比第 i回合后少，逃亡者可以跳过这个环路，而不会使自己的处境变差。换句话说，加入“逃亡者不能返回先前经过的顶点”这一规则，并不会改变博弈的结果。证明中会考虑这条附加规则。因此，如果逃亡者获胜，他最多可以在 $|V|$ 回合内获胜。由此可知，在多重图中，重数大于 $|V|$ 的边不可能被 Nemesis 完全移除。这类边的表现就像具有无限重数一样，称为不可破坏的。

▶ 注 8. 在简单图中，许多实例都是平凡的。如果没有顶点与两个出口相邻，那么逃亡者总是会输；但在他一开始就在出口或出口的邻居处这两种平凡情况下除外。事实上，每当逃亡者到达一个距某个出口为 1 的顶点时，Nemesis 就可以移除通往该出口的唯一一条边，使该出口无法到达。因此，在简单图中，Nemesis 的非平凡实例必须包含与至少两个出口相邻的顶点。

2.2 练习

作为热身，我们先介绍几个游戏的小型实例。图2所示的实例中，谁能获胜？答案见脚注。[4]

[4] a–Nemesis 获胜（N），b–逃亡者获胜（F），c–N，d–N，e–F，f–N，g–F，h–F。

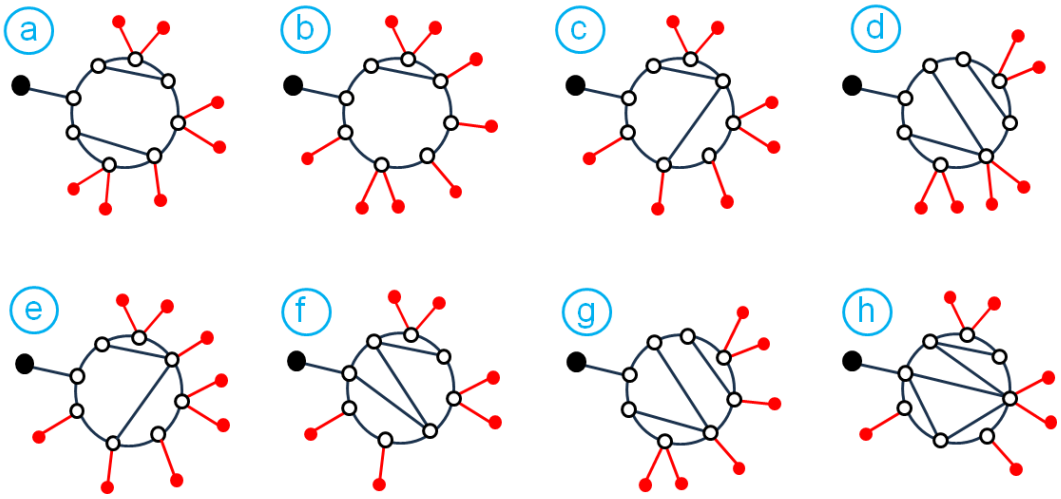


图 2 Nemesis 的八个实例。 逃亡者从黑色顶点出发，能否逃脱?

2.3 关于出口数量

现在，我们来研究出口数量如何影响 Nemesis 的复杂度。首先可以观察到，对于多重图，Nemesis 可以直接归约为只有一个出口的 Nemesis。事实上，可以将所有出口合并为一个出口顶点 t：对于每个顶点 v，将从 v 通往各个出口的边替换为一条从 v 通往 t 的边，其重数等于原有重数之和。

在后面的章节中，我们将证明，Nemesis 在多重图上是 PSPACE 完全的（定理 5）。上述归约立即表明，即使只有一个出口，Nemesis 在这种情况下仍然是 PSPACE 完全的。因此，假设 P $\neq$ PSPACE，则不存在任何以出口数量为参数的易处理算法（无论是固定参数可处理还是 XP）。

简单图的情况则不同。只有一个出口时，Nemesis 会变得平凡。实际上，有两种情况。如果起始顶点 s 与唯一的出口 t相邻，逃亡者立即获胜。否则，正如注 8 所指出的，Nemesis 有一个简单的获胜策略：每当逃亡者到达 t的一个邻居 u 时，她就移除边（u，t）。

因此，自然会问，这一观察是否可以推广到固定数量的出口。如我们现在所示，答案是否定的：即使在简单图中，将出口数量限制为两个，也已经涵盖了该博弈的全部复杂性。

▶**定理 9。** 对于简单图，存在一个从 NEMESIS 到具有两个出口的 Nemesis的多项式时间归约。

证明。 设 (G, s, X) 是 NEMESIS 的一个实例，其中 $G = (V, E)$ 是一个简单图，$s \in V$是起始顶点，且 $X \subsetneq V$ 是出口集合，其中 $s \notin X$。我们构造一个 $(G', s, \{t_1, t_2\})$ 的 具有两个出口的 Nemesis 实例，具体如下。

- G' 的顶点集为 $V\prime = V \cup Y \cup \{t_1, t_2\}$，其中 $Y = \{y_1, \ldots, y_{n+1}\}$ 是一组 $n + 1$新顶点。G' 的边集为 $E' = E \cup (X \times Y) \cup (Y \times \{t_1, t_2\})$，也就是说，$X$ 的每个顶点都与 Y的每个顶点相连，而 Y 的每个顶点都与两个出口 t_1 和t_2相连。起始位置仍为 s，出口为 t_1 和 t_2。
-

图 G' 有 $2n+3$ 个顶点，最多有 $m+(n-1)(n+1)+2(n+1)$ 条边。

现在我们证明这两个实例是等价的。首先，假设逃亡者在 (G,s,X) 中有获胜策略。根据备注 7，他至多经过 n 回合就能到达某个出口 $x\in X$。在此之前，Nemesis 至多移除了 n 条边。由于 $|Y|=n+1$，存在一个顶点 $y\in Y$，使得与 y 相接的边均未被移除。特别是，y 仍与 x、t_1 和 t_2 相邻。因此，逃亡者可以从 x 移动到 y。由于 y 有两个出口邻点，Nemesis 至多能挡住其中一个，逃亡者因此获胜。

反过来，如果 Nemesis 在 (G,s,X) 中有获胜策略，那么在 $(G',s,\{t_1,t_2\})$ 中采用相同的策略，就能阻止逃亡者到达 X 的顶点，从而阻止他到达两个出口 $\{t_1,t_2\}$ 中的任何一个。◀

定理 9 表明，在简单图中，两个出口足以刻画 Nemesis 的计算复杂性。不过需要注意的是，这一归约通常并不保持平面性。

3 多项式时间游戏

对于某些实例类别，可以利用经典的图结构高效地确定博弈的赢家。现在我们来研究几个这样的情形。

3.1 树与最大度为 3 的图

在讨论最简单的图类别之前，我们先介绍一种图简化过程。

图简化

给定一个带有出口和起始顶点 s的图，我们分两个阶段引入一种初步归约（图 3）。第一阶段是复制出口，使每个出口的度为 1。例如，图 3 中从 a) 到 b) 的转变就采用了这一操作。这种变换通过避免经过出口的无意义环来简化结构，同时保持博弈的结果不变。

第二种变换是从图中移除所有这样的顶点：如果逃亡者到达其中任何一个顶点，他就注定会输。更确切地说，我们会迭代地剪枝所有度为 1 或 2 的普通顶点（起始顶点 s 除外），因为这些顶点会成为逃亡者的陷阱。由于这些顶点在获胜策略中从不发挥作用，因此这一剪枝阶段不会改变博弈的结果。

这种剪枝可以通过广度优先搜索（广度优先搜索）来实现，并以初始待移除顶点集合进行初始化。我们还会舍弃任何不包含 s 的连通分量。总体而言，这一预处理可以在线性时间内完成。

因此，从现在起，我们假设所有输入图都已事先化简。请注意，这种化简绝不会增加任何顶点的度。

二叉逃逸树

保证逃亡者获胜的一个关键结构是二叉逃逸树。

▶ **定义 10.** 图 G （给定一组出口）的二叉逃逸树是一棵有根满二叉树（即每个内部节点恰有两个子节点的有根树），其顶点和边均属于 G，且叶子均为出口。

二叉逃逸树为逃亡者提供了一种简单的获胜策略。

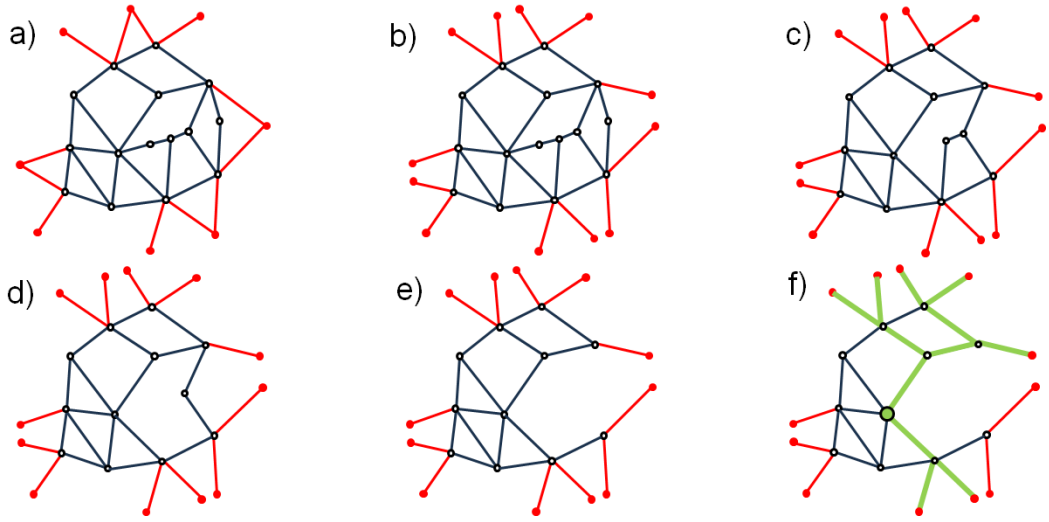


图3 输入图的化简。从 a) 到 b)，将出口复制。从 b) 到 e)，剪除度为 1 或 2 的普通顶点。在 f) 中，是一棵二叉逃逸树。

▶ **引理 11**。给定一个 *Nemesis* 实例，其图为 $G = (V、E)$，并给定一组出口和一个起始顶点 s，如果 s 或其在 G 中的某个邻居是二叉逃逸树的根，那么逃亡者就拥有获胜策略。

证明。获胜策略很简单。如果二叉逃逸树的根是 s的邻居，逃亡者就在第一回合移动到该根。否则，如果这棵树以 s为根，他就移动到它的一个子节点。

每回合，Nemesis 至多可以删除一条边。因此，从逃亡者当前所在的位置出发，当前根的两棵子树中至少有一棵保持完整。下一回合，逃亡者移动到这棵未受改变的子树的根。重复这一策略，可以确保逃亡者每次移动后都位于当前图中一棵二叉逃逸树的根上。由于二叉逃逸树的高度在每回合都严格递减，逃亡者最终会到达一个出口并赢得博弈。◀

我们将树条件定义为这样一种性质：至少有一个距离当前逃亡者位置不超过 1 的顶点，是一棵二叉逃逸树的根。引理 11 随后可以表述如下：如果满足树条件，逃亡者将获胜。

自然会有人问，这一条件是否不仅是充分条件，也是必要条件。如果确实如此，它将为逃亡者能够采取获胜策略的起始位置提供一个简单的刻画。然而，一般而言事实并非如此。实际上，存在一些实例，尽管不满足树条件，逃亡者仍然能够获胜。方格网格中就有一个反例。

方格网格

方格网格值得特别关注，因为熟悉康威天使游戏的读者可能会在 Nemesis 中认出恶魔策略的一种变体 [8, 14]。我们实例中的普通顶点集合构成一个 $m \times n$ 方格网格。每个角落顶点都连接到两个出口，而其他每个边界顶点恰好连接到一个出口。该实例及相应的策略见图 4。

N:10 Nemesis，图中的逃脱游戏

不难看出，如果逃亡者从距离最近出口至少为 6 的顶点开始，那么 Nemesis 可以通过切断角落出口来采用获胜策略。反之，如果起始顶点到边界顶点的距离至多为 5，逃亡者则可以通过抵达边界，然后沿网格移动来采用获胜策略（我们不对这一结论作形式化证明）。

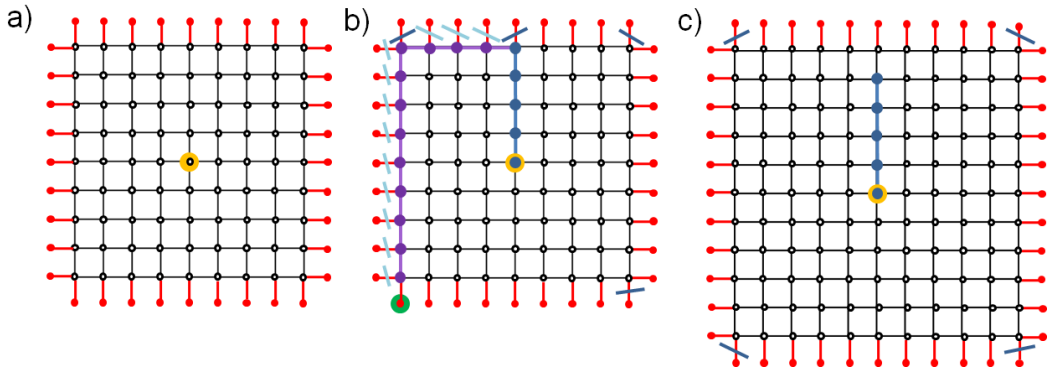


图4 方格网格上的 Nemesis。 在 a) 中，输入图的起始顶点为橙色，出口为红色。在 b) 中，起始顶点与最近出口的距离为 5，逃亡者通过在网格中移动获胜。在 c) 中，起始顶点与最近出口的距离为 6，Nemesis 通过切断每个角落的一个出口获胜。

网格的一个有趣特征是，当起始顶点 s 与出口的距离为 5 时，逃亡者拥有获胜策略，尽管 s 不满足树条件。现在我们通过矛盾来证明这一点。

假设一棵二叉逃逸树的根位于距离 s至多为 1 的顶点上。那么，从该根到出口的距离至少为 4。这个根有两个子节点，它们到出口的距离至少为 3，并且每个子节点都是一棵独立二叉逃脱树的根。这两个节点中的每一个又分别有两个子节点，它们到出口的距离至少为 2，同样作为独立二叉逃脱树的根。因此，在下一层，我们得到 8 个到出口的距离至少为 1 的节点，每个节点都是一棵二叉逃逸树的根。

然而，任何未以出口为根的二叉逃逸树，都必须包含至少一个与两个出口相邻的顶点。因此，这一构造意味着存在 8 个不同的顶点，每个顶点都与两个出口相邻。由于网格中只有 4 个这样的顶点，因此我们得出矛盾。因此， s 不满足树条件。

树条件是树和最大度为 3 的图的必要且充分条件

对于任意图，树条件是保证逃亡者获胜的充分条件，但不是必要条件。现在我们将证明，当图 G 是树，或（经过图简化后）最大度不超过 3 时，该条件也是必要条件。

▶ **引理 12.** 给定一个包含一组出口和一个图 $G = (V, E)$ 的 *Nemesis* 实例，其中该图是一棵树，或在图简化后最大度不超过 3 ，当且仅当 s或其在 G中的某个邻居是二叉逃逸树的根时，逃亡者才能从起始位置 s采用获胜策略。

证明。 注意，树或最大度数为 3 的图的简化图本身也是树或最大度数为 3 的图。图已经过简化这一条件并不会缩小引理 12 的适用范围，但

为避免在复制步骤之前人工环可能穿过初始图中的出口，这一条件是必要的。

等价性的一个方向由引理 11 给出。我们证明其逆命题，即如果不满足树条件，那么 Nemesis 就有获胜策略。

我们对图 G 的大小 $n = |V|$ 进行归纳。

我们先考虑树。归纳假设是：对于任意至多有 n 个顶点的树，当且仅当距离逃亡者当前位置不超过 1 的某个顶点是二叉逃逸树的根时，逃亡者才能获胜。

现在考虑一棵有 $n+1$ 个顶点的树，并假设逃亡者从 s 出发有获胜策略。根据备注 7，我们可以假设该策略不会两次到访同一个顶点。逃亡者第一次移动时，从 s 移动到顶点 v，并且再也不会返回 s。因此，逃亡者在以 v 为根的 G 子树中有获胜策略，该子树的大小至多为 n。

根据归纳假设，v 本身或其某个邻居是这棵子树中一棵二叉逃逸树的根。如果 v 是这样的根，那么树条件在 s处成立。否则，v 的至少一个子节点具有这一性质。

假设 v 的子节点中恰有一个 v' 是这棵子树中一棵二叉逃逸树的根。那么，逃亡者从 s 移动到 v后，Nemesis 立即可以删除边 (v, v')，从而破坏 v处的树条件。根据归纳假设，逃亡者此时没有获胜策略，这与我们的假设矛盾。因此，v 必须至少有两个子节点是二叉逃逸树的根。

由于 G 是一棵树，这两棵子树顶点不相交。因此，v 本身是一棵二叉逃逸树的根。由此，s 处通过其邻居 v满足树条件。

现在我们考虑最大度为 3 的图。证明沿用相同的归纳框架，但还需要在最后补充一个论证。

归纳假设是：对于任意最大度不超过 3 且至多有 n 个顶点的图，当且仅当满足树条件时，逃亡者才能获胜。设 G 是一个有 $n+1$ 个顶点的此类图，且 s 是逃亡者的一个获胜起始位置。

和之前一样，我们可以假设逃亡者遵循一种不经过环路的获胜策略。根据该获胜策略，设 v 是逃亡者从 s 到达的第一个顶点。第一回合结束后（逃亡者移动一次，Nemesis 删除一个顶点），剩余图的规模至多为 n，而逃亡者位于 v。根据归纳假设，v 或其某个邻居是二叉逃逸树的根。

如果 v 本身就是这样的根，那么树条件成立，证明完毕。否则，设 v' 是 v 的一个邻居，且它是二叉逃逸树的根。由于在这种情况下，v 是正则的，v 还有另外两个邻居：s、v' 和 v''（经过化简后，唯一的度为 2 的顶点可能是 s）。

Nemesis 可能在第一回合删除边（v，v'），迫使逃亡者向v''移动。如果 v'' 不是二叉逃逸树的根，那么根据归纳假设，逃亡者将输掉比赛，这与我们的假设相矛盾。因此，v' 和 v'' 都必须是二叉逃逸树的根。

如果以 v' 和 v'' 为根的两棵二叉逃逸树顶点不相交，那么 v 本身就是一棵二叉逃逸树的根，并且树条件成立。否则，这两棵树会共享顶点。现在我们证明，这种配置与逃亡者从移动（s，v）开始便拥有获胜策略这一初始假设相矛盾。

由于图的最大度为 3，且逃亡者不会重访顶点，因此逃亡者每次到达某个顶点时，与该顶点相连的一条边已经被使用过，之后不会再经过。在剩下的两条边中，Nemesis 可以删除其中一条，从而迫使逃亡者下一步的移动。如果以 v' 和 v'' 为根的两棵二叉逃逸树并非顶点不相交，

那么它们的并集就包含一个经过 v、 v' 和 v'' 的环 C，Nemesis 可以借此困住逃亡者。事实上，她每回合只需移除环 C 上唯一一条通向外部的边。由于假设图已经化简，环 C 不包含出口。因此 Nemesis 必胜，这与逃亡者最初移动到 v 后能够逃脱的假设相矛盾。由此可知，只要逃亡者在树或最大度为 3 的图中拥有获胜策略，就必须满足树条件。

◀

从 s 开始进行广度优先搜索，可以在线性时间内检查树是否满足树条件。对于最大度为 3 的图，也可以在线性时间内检查该条件，不过论证稍微复杂一些。该算法还会从起始顶点 s 进行广度优先搜索，并根据每个已访问顶点是经由 s 的哪个邻居首次到达，为其标记相应的标签。由于 $\deg(s) \leq 3$，这样的分支至多有三个。

如果一个顶点是从两个不同的分支到达的，那么这两个分支都会被禁用，因为它们必然共享一个环，不能同时属于同一棵二叉逃逸树。如果一个顶点从同一分支被到达两次，那么该分支也会被禁用。

遍历之后，当且仅当至少有两个分支仍处于启用状态时，顶点 s 才是二叉逃逸树的根。该算法正确，是因为在最大度为 3 的图中，任何包含环的分支都不可能属于二叉逃逸树。反过来，由于预先进行了化简，所有普通顶点的度都是 3。因此，任何不含环的分支都是二叉出口树。于是，存在两个处于启用状态（即不含环）的分支，就意味着 s 是二叉逃逸树的根。

3.2 Blizzard

现在我们来研究 Blizzard 的复杂性，并证明定理 2。Blizzard 博弈与 Nemesis 的不同之处在于，它限制了对手的能力。Nemesis（现在称为风暴）只能删除与逃亡者（现在称为捕兽者）当前位置相邻的边。这一限制至关重要，因为它使我们能够比在 Nemesis 中更简单地刻画捕兽者的获胜位置。

获胜位置集合

在本小节中，将时间点设定在回合中途，也就是捕兽者移动之后、风暴删除一条边之前，会更方便。

首先，我们将出口顶点集合定义为获胜位置的基础集合 W_0。显然，如果捕兽者位于出口，就意味着他已经获胜。接着，我们将至少有两条出口边的普通顶点集合定义为 W_1，也就是说，这些边通向 W_0 中的顶点。如果在回合中途，捕兽者位于这样的顶点，且这些边尚未被删除，那么风暴最多只能删除其中一条，仍会留下一条通往出口的路径。因此，捕兽者仍然能够获胜。

我们通过归纳法扩展这一构造，定义顶点集合 W_k，并称之为秩为 k的获胜位置。尽管这一术语暗示这些位置对应于捕兽者的获胜策略，但目前我们仅按以下方式用归纳法定义它们：对于 $k \geq 1$，集合 W_k 由在 $\bigcup_{i<k} W_i$中至少有两个邻居的顶点组成。归纳过程持续进行，直到不再加入新顶点，也就是说，直到某个 $W_\ell = \emptyset$。我们定义获胜位置集合 $W = \bigcup_{k<\ell} W_k$.

集合 W 可以通过广度优先搜索计算：逐步标记距离 W 为 1 的顶点，因此所需时间与图的规模成线性关系。现在我们来说明

这一术语的由来，证明 W 确实与捕兽者的获胜位置集合一致。

▶**定理 13.** 在 *Blizzard* 中，捕兽者拥有获胜策略，当且仅当其起始位置与 W 的某个顶点相邻。

证明。 在此证明中，我们仍考察一回合中途时捕兽者的位置，也就是说，在他移动之后、风暴删除一条边之前。

我们首先证明，若在一回合中途，捕兽者位于属于 W 的顶点上，那么他就拥有获胜策略。我们对秩 k 使用归纳法。假设对于某个 $d \geq 0$，对于 $\bigcup_{k\leq d} W_k$ 中的每个顶点，捕兽者从该顶点出发，都有只使用端点属于 $\bigcup_{k\leq d} W_k$ 的边的获胜策略。现在，设 $u \in W_{d+1}$。根据 W_{d+1} 的定义，顶点 u 在 $\bigcup_{k\leq d} W_k$ 中至少有两个邻居 v 和 v'。根据归纳假设，v 和 v' 对捕兽者来说都是获胜位置，且相应的获胜策略完全包含在 $\bigcup_{k\leq d} W_k$ 中。在 Blizzard 中，风暴每回合至多删除一条边，而且这条边必须与捕兽者当前所在位置相接。因此，风暴可以删除 $(u,\ v)$ 或 $(u,\ v')$，但不能两者兼删。此外，由于从 v 和 v' 出发的获胜策略都不依赖与 u 相接的边，删除与 u 相接的一条边不会影响这些策略。因此，捕兽者从 u 出发，移动到 v 或 v'。如果风暴删除通往其中一个顶点的边，捕兽者就移动到另一个顶点。无论哪种情况，他都会到达一个依据归纳假设能确保获胜的顶点。这说明，捕兽者在一回合中途位于 u 时拥有获胜策略，从而将归纳假设扩展到 $d+1$。

现在我们证明反向命题。假设在某一回合进行到一半时，捕兽者站在顶点 $u \notin W$。根据集合 W_k 的构造，这样的顶点在 W 中至多有一个邻居。因此，风暴可以删除连接 u 和 W 的唯一一条边（如果存在），从而阻止捕兽者进入 W。由于 $W_0 \subseteq W$ 和 W_0 是出口集合，捕兽者永远无法到达出口。因此，在这种情况下，风暴拥有获胜策略。◀

定理 2 指出，Blizzard 可以在线性时间内求解。这是定理 13 的推论，且获胜位置集合可以通过广度优先搜索计算得出。

4 困难性结果

本节将证明本文的困难性结果。首先，我们证明定理 3，该定理确定了判定给定顶点是否为二叉逃逸树的根这一问题的复杂度（第 4.1 节）。接着，我们给出从平面可满足性问题到平面多重图上的 Nemesis 的归约，从而证明定理 4（第 4.2 节）。最后，我们利用任意多重图作为中间引理（第 4.3 节），通过定理 5 证明任意图上的 Nemesis 是 PSPACE 完全的（第 4.4 节）。作为推论，我们还证明了赶猫问题是 PSPACE 完全的（第 4.5 节）。

4.1 二叉逃逸树搜索的 NP 完全性

下面重述的定理 3 展示了二叉逃逸树问题的复杂度差异。第 3.1 节已证明，该问题可在线性时间内求解，适用于树以及最大度为 3 的图。

▶**定理 3.** 给定一个带有一组出口的图和一个顶点 r，判断 r 是否是二叉逃逸树的根是 *NP* 完全的，即使该图的最大度为4。

证明。 该问题显然属于 NP：给定一个证书（以 r为根的子树），我们可以在多项式时间内验证它是否为二叉逃逸树。为证明其完备性，我们从线性3-SAT归约而来，其定义如下：

线性3-SAT
输入： 合取范式布尔公式 φ，其中每个子句至多与另一个子句相交；若相交，它们至多共享一个文字（称为 LSAT 公式）。**问题：** φ是否可满足?

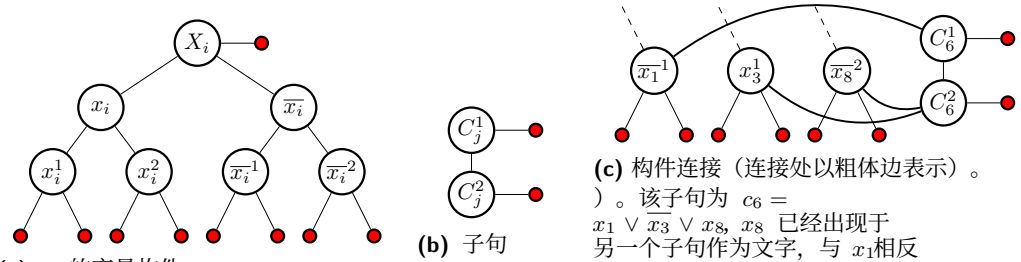
已知线性3-SAT 是 NP 完全问题 [2]。注意，在 LSAT 公式中，每个变量至多出现四次：至多两次作为正文字，至多两次作为负文字。

设 φ 是一个含有变量 x_1、……、x_n 和子句 c_1、……、c_m的 LSAT 公式。我们按如下方式从 φ 构造图 G。

变量构件。 对于每个变量 x_i，我们构造如下有根树（如图 5a 所示）。根 X_i 有一个出口子节点，以及两个子节点 x_i 和 $\overline{x_i}$。节点 x_i（分别为$\overline{x_i}$）有两个子节点 x_i^1 和 x_i^2（分别为 $\overline{x_i}^1$ 和 $\overline{x_i}^2$），每个子节点又各有两个出口子节点。该构件的设计思路是允许选择一条分支：选择 x_i（分别为 $\overline{x_i}$）分支，对应于将变量 x_i 设为真（分别为假）。

子句构件。 对于每个子句 c_j，我们构造如下简单构件：两个相邻顶点 C_j^1 和 C_j^2，每个顶点都有一个出口子节点。该构件如图 5b 所示。

构件连接。 对于每个 $1 \le j \le m$，我们将 C_j^1 与子句 c_j的第一个文字关联，将 C_j^2 与其第二个和第三个文字关联。我们按以下方式将 C_j^1 和 C_j^2 连接到 c_j 中关联文字的**相反**项：令 x_i 为 c_j 中的第一个文字；如果 x_i 已在先前的[5] 子句中出现过（这是唯一一个与 c_j 相交的子句），则将 C_j^1连接到 x_i^2，否则将 C_j^1 连接到 x_i^1。对于第二个和第三个文字以及 C_j^2，也采用同样的方法。例如，令 $c_6 = x_1 \vee x_3 \vee x_8$ 满足以下条件：x_8 是 c_6 中唯一一个曾在先前子句中出现的文字；我们将 C_6^1 连接到顶点 x_1^1，并将 C_6^2 连接到顶点 x_3^1 和 x_8^2，如图 5c 所示。



(a) x_i 的变量构件。

(b) 子句 c_j 的构件。

(c) 构件连接（连接处以粗体边表示）。）。该子句为 $c_6 = x_1 \vee \overline{x_3} \vee x_8$，$x_8$ 已经出现于另一个子句作为文字，与 x_1相反以及 $\overline{x_3}$。

图5 构件。 从线性3-SAT归约而来的变量构件和子句构件。出口顶点以红色填充。

完整构造。我们根据公式 φ 按如下方式进行。首先，根顶点 r 与顶点 X'_1、……、X'_n 构成一条线性路径，这些顶点依次相连，并且每个顶点都带有一个以 X_1、……、X_n 为根的悬挂变量构件。按照 X'_n，接下来我们有一个

[5] 在全序 c_1、……、c_m 的意义下

由顶点 C'_1、……、C'_m 构成的线性路径，这些顶点依次相连，并且每个顶点都带有一个悬挂子句构件，其中 C^1_j 与 C'_j 相连。子句构件和变量构件按照上述方式连接。最后， r 和 C'_m 有相邻的出口[6]。完整构造如图6所示。

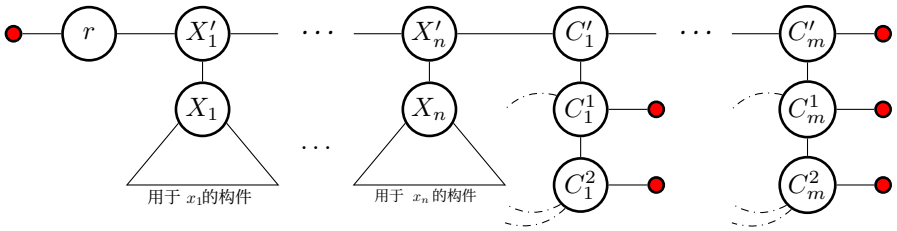


图6 构造的图。 从线性3-SAT进行的归约。变量装置如图5a所示，子句构件引出的点划线连接到变量构件中的顶点，如图5c所述。

构造的图 G的阶为 $16n+4m+3$，因此可以在多项式时间内计算出来。此外， G 的最大度为 4。现在我们证明：当且仅当 G 具有一个以 r为根的二叉逃逸树时， φ 是可满足的。

($\Rightarrow$) 选择一个满足 φ的变量赋值。由此我们推导出一个二叉逃逸树，如图7所示。该树包含由 r的出口、 r、 X'_1、……、 X'_n、 C'_1、……、 C'_m以及 C'_m的出口所诱导的路径。为了使其成为二叉树，我们将在构件中进行分叉。对于每个 X'_i，我们将顶点 X_i 及其出口子节点作为 X'_i的子节点加入该树中。如果将 x_i 设为真（分别为假），我们还将以顶点 x_i （分别为 x_i）为根的变量构件子树作为 X_i的子节点加入。对于每个 C'_j，我们将顶点 C^1_j 及其出口子节点作为 C'_j的子节点加入该树中。如果 c_j 的第一个文字被设为真，我们还将变量构件中以 C^1_j的邻居为根的子树加入。否则，我们选择 c_j 中的一个正文字（第二个或第三个），并将顶点 C^2_j 及其出口子节点，以及与子句 c_j的某个正文字关联的变量构件邻居为根的子树，作为 C^1_j的子节点加入该树。这就得到了一个二叉逃逸树：对于每个内部节点，我们处于以下三种情况之一：（1）两个出口子节点；（2）一个出口子节点 $+$ 和一个以二叉出口树为根的子节点；（3）两个以二叉逃逸树为根的子节点。

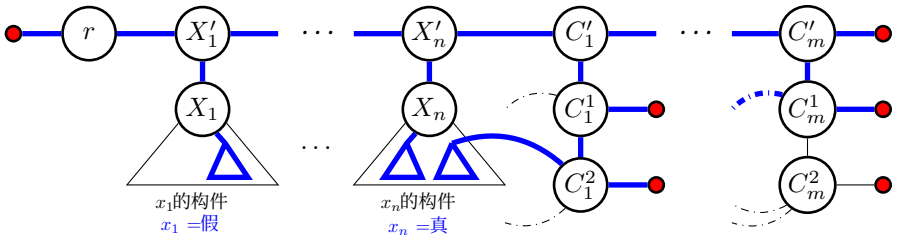


图 7 等价性示意图。 由有效变量赋值得到的二叉逃逸树的部分示意图。例如，变量 x_1 被设为假， x_n 被设为真。 c_1 的第一个文字为假，但第二个文字为真，并且等于 x_n。 c_m 的第一个文字为真。

($\Leftarrow$) 设 T 是一棵以 r 为根的二叉出口树。它满足以下性质（在证明的其余部分中，子节点 始终指 T 中的子节点）：

[6] 请注意，可以移除与 r 相邻的出口，从而将问题归约为判定是否存在一棵根与起始位置相邻的二叉逃逸树。

(i) 包含 $r, r, X'_1, \ldots, X'_n, C'_1, \ldots, C'_m$ 的出口和 C'_m 的出口的诱导路径位于 T 中。这是因为这些顶点（根 r 除外）的度恰好为 3，并且 T 是一棵满二叉树（ r 除外， T 中没有顶点的度为 2）。(ii) 该诱导路径的邻域也位于 T 中，即所有顶点 X_i 和 C_j^1。这是因为路径上的每个顶点恰有一个不在路径上的邻居，这使得作为一棵二叉树的 T 必须包含所有这些顶点。(iii) 对于每个 $1 \leq j \leq m$， C_j^1 和 C_j^2 这两个顶点中至少有一个，其子节点属于某个变量构件。否则，其中一个顶点在 T 中的度将为 1，而以 C_j^1 为根的子树就不是满二叉树，这与假设矛盾。(iv) 每个顶点 X_i 都至少有 x_i 和 x_i 之一作为子节点。事实上，在 T 中，每个顶点 X_i 都必须有两个子节点。其中一个子节点可以是其邻居出口。另一个必然是顶点 x_i 或 x_i 之一，以及它们各自的后代。(v) 只有当 ${x_i}^1$ 和 ${x_i}^2$ （分别为 x_i^1 和 x_i^2 ）不是子句构件顶点的子节点时，顶点 x_i^1 和 x_i^2 （分别为 ${x_i}^1$ 和 ${x_i}^2$）才能成为子句构件顶点的子节点。否则，如果某个子句构件在 T 中同时与分别以 x_i 和 x_i 为根的两棵子树相邻，那么根据 (iv) 中的论证， T 中会包含一个环，这与假设矛盾。

现在，我们为变量构造如下真值赋值：如果在 T 中， x_i 是 X_i 的子节点，则将变量 x_i 设为真。否则，将变量 x_i 设为假。我们声称，性质 (i)-(v) 可直接对应为：

- 每个变量都被赋为真或假：(iv)[7]。每个子句都与其至少一个文字相关联：(iii)。与子句
- 相关联的文字被设为真：否则 T 会包含一个环。一个变量不能与两个包含其相反文字
- 的子句相关联：(v)。

这些性质意味着该赋值是一致的，并且满足 φ，从而完成归约。◀

4.2 Nemesis 在平面多重图上的困难性

现在我们证明定理 4。熟悉将 QSAT 归约到 Geography [22] 或加拿大旅行者问题 [26] 的读者，或许能认出将 QSAT 归约到图问题时所采用的总体结构。这些经典归约使用一条中央主路径，并围绕它通过局部选择来编码 SAT 实例的布尔变量赋值。

平面单调直线3SAT

我们并不直接归约 QSAT，而是归约 NP 完全问题平面单调直线3SAT [15], , 记作 PMR3SAT。这些 3SAT 实例的一个关键特征是，每个子句要么是全正子句（仅包含未取反文字），要么是全负子句（仅包含取反文字）。平面单调直线3SAT 的一项重要性质是，它可以用互不相交的线段进行几何表示。

[7] 请注意，某些变量可能同时将 x_i 和 x_i 作为 X_i 的子节点；这意味着它们不用于满足子句，因此可以任意设为真或假（此处我们选择了真）。

在这种表示中，布尔变量以一系列平行于坐标轴的矩形表示，排列在水平轴上。子句则对应于平行于坐标轴的矩形：全负子句的矩形放置在半平面 $y<0$ 中，全正子句的矩形放置在半平面 $y>0$ 中。变量是否出现在某个子句中，通过连接变量矩形与该子句矩形的垂直线段表示。图 8 上半部分展示了一个 PMR3SAT 实例。我们并不依赖这些线段平行于坐标轴这一性质。

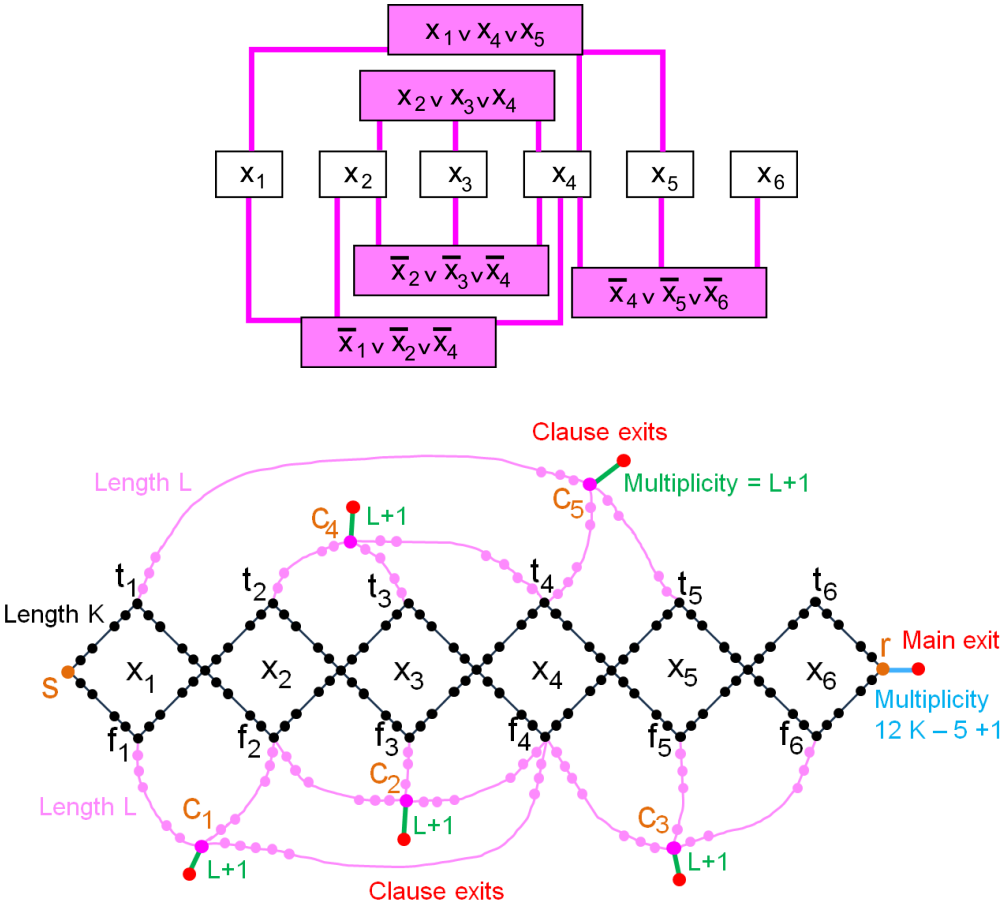


图 8 将一个 PMR3SAT 实例归约为一个 Nemesis 实例。 我们对上文表示的 PMR3SAT 实例进行归约。Nemesis 实例中的所有边均不可破坏，唯有与出口相邻的边（绿色）例外。子句出口的重数为 $L+1$，主出口的重数为 $12K-5+1$。

复仇者实例的构造

我们将介绍如何把一个 PMR3SAT 实例归约为平面多重图中的一个 Nemesis 实例。设 PMR3SAT 实例的变量数为 n，子句数为 m。构造过程见图 8。

每个布尔变量都用一个大小为 $4K$的环表示，其中 K 是一个常数，其值大于任意给定变量出现的子句数上限（大于 m 即可）。因此，PMR3SAT实例中 n 个变量的有序列表

N:18 Nemesis：图中的一款逃脱游戏

被编码为一系列环——每个变量对应一个环——每个环都通过一个中间顶点与下一个环相连。

这一归约的关键思想是：从最左侧的起始顶点 s 前往最右侧的普通顶点 r时，逃亡者必须沿每个环的一侧或另一侧穿行，因此，他从 s 到 r 的路径编码了对 n 个变量的一种赋值。为此，我们在每个环的两侧分别引入中间顶点 t_i （表示真）和 f_i （表示假）。例如，一条从 s 到 r 、依次经过顶点 t_1、 f_2、 t_3、 t_4、 f_5、 f_6 的路径，对应于这样一种赋值： x_1、 x_3 和 x_4 为真，而 x_2、 x_5 和 x_6 为假。

请注意，无论每个变量选择哪一侧，所有经过变量环的 s-r 路径长度都相同，为 $2nK$（图中，取 $n = 6$时，该长度为 $12K$）。

由于这条中心路径对归约至关重要，我们确保 Nemesis 无法破坏其结构：这条路径上的所有边都赋予了使其不可破坏的重数。下文引入的额外边也同样如此。因此，几乎所有边都是不可破坏的。只有与出口相接的边，其重数允许 Nemesis 将其删除。主出口与最右侧的普通顶点 r相连，我们将这条边的重数设为 $2nK - m + 1$。其余出口对应于子句。

每个子句都表示为一个顶点 c_i ，它通过一条重数为$L + 1$的边连接到一个出口，其中 L 的取值大于 $2nK$。每个子句顶点 c_h 也与其变量所在的环相连。

对于全正子句 $x_i \vee x_j \vee x_k$，顶点 c_h 通过长度为 L的不可破坏路径连接到 t_i、 t_j和 t_k。

对于全负子句 $x_i \vee x_j \vee \overline{x_k}$，顶点 $\overline{c_h}$ 通过长度为 L的不可破坏路径连接到 f_i、 f_j和 f_k。

L的取值大于 s 到 r的距离，这保证了这些子句路径不会在 s 和 r之间形成捷径。

它是如何运作的?

我们首先考虑从 s 到 r的直接路径。对于每个变量 x_i，路径会经过 t_i 或 f_i。不失一般性地，假设逃亡者到达了 t_i。从那里，他可以尝试沿着相应的某条子句路径到达出口。当逃亡者位于 t_i时，他到 c_h 的距离为 L，而从 c_h 出发的出口边的初始重数为$L + 1$。因此，如果 Nemesis 此前至少没有攻击过这条出口边一次，逃亡者就能到达子句出口，因为 Nemesis 已经没有足够的时间移除这条边。

这迫使 Nemesis 进行预判：在逃亡者到达 t_i之前，他必须 traverses 一条长度为 K 的路径。在这 K 回合中，Nemesis 必须攻击与 t_i 相连的所有子句的出口边。将每条此类出口边的重数减少一，是使相应子句出口变得无害的充分条件。这种归约不可避免，因为否则逃亡者就可能通过该出口逃脱。因此，对于每个从逃亡者路径可到达的子句，Nemesis 必须花费一个回合，将一条出口边的重数从 $L + 1$ 减少到 L。

PMR3SAT 与相应 Nemesis 实例之间的等价性

现在我们证明，当 PMR3SAT 实例可满足时，逃亡者沿着与满足赋值对应的路径前进，就拥有获胜策略。在这条路径上，逃亡者经过的顶点 t_i 和 f_i 与所有 m 个子句相连。因此，Nemesis 必须花费 m 个回合来阻止逃亡者从子句出口逃脱。这样一来，只剩下 $2nK - m$

个回合，Nemesis 可以在逃亡者从 s 到 r 的途中（路径长度为 $2nK$）摧毁主出口边。由于其重数设为 $2nK-m+1$，在第 $2nK-m+1$ 回合开始时（逃亡者到达 r 后），该边的重数仍为 1，因此逃亡者可以逃脱。

现在我们证明，如果 PMR3SAT 实例不可满足，那么逃亡者就无法逃脱。在这种情况下，逃亡者从 s 到 r 的路径至多经过 $m-1$ 个子句，因此 Nemesis 有足够的空闲回合来移除主出口边。

逃亡者能否选择一条更好的路径？走子句路径并无优势：如果他从 t_i 或 f_i 前往 c_h，那么 Nemesis 每回合都可以降低该子句出口边的重数。Nemesis 恰好有足够的时间在逃亡者通过出口边之前将其移除。此时逃亡者必须返回中央路径，而在此期间 Nemesis 有充足的时间摧毁主出口并获胜。

另一种可能是在变量环中折返，但这要求逃亡者两次经过某个顶点。这并无帮助：如果逃亡者确实拥有获胜策略，那么他也能拥有一套不包含环路的策略（备注 7）。

因此，无论逃亡者选择哪条路径，他都无法到达出口。

现在来验证该归约所需的时间是多项式时间。中央路径涉及的顶点数少于 $4nK$ with $K \leq m$。子句路径需要 $3(L+1)m$ 个顶点，其中 L 的数量级为 $2nK$。因此，顶点总数为$O(m^2n^2)$。编码初始 PMR3SAT 实例的 Nemesis 实例由此可在多项式时间内计算出来。

这就完成了 NP 困难性的证明，证明了定理 4。

4.3 任意多重图上 Nemesis 的 PSPACE 完全性

现在我们证明一个通往定理 5 的中间结果，即 Nemesis 在多重图上的 PSPACE 完全性。

全称玩家与存在玩家之间的 QSAT 博弈

为了确立 Nemesis 的 PSPACE困难性，我们需要将先前从 3SAT（或一般意义上的 SAT）进行的归约，转变为从量化 SAT（QSAT）进行的归约。这相当于从一个 φ 仅含存在量词的公式，

$$\exists x_1, \exists x_2, ..., \exists x_i, ..., \exists x_n, \varphi\left((x_i)_{1\leq i\leq n}\right)$$

转变为一个含有交替出现的存在量词和全称量词的公式，

$$\exists x_1, \forall x_2, ..., \exists x_{2i-1}, \forall x_{2i}, ..., \forall x_{2n}, \varphi\left((x_i)_{1\leq i\leq 2n}\right).$$

以存在量词量化的变量称为存在变量，而其他变量称为全称变量。

这种量化公式的可满足性可以看作存在玩家与全称对手之间的一场博弈：存在玩家为存在变量赋值，目标是使公式得到满足；而全称对手为全称变量赋值，目标是阻止公式得到满足。为了证明 PSPACE困难性，我们正是将这场博弈归约到 Nemesis。

电气装置及其分析

现在，我们来考察第 4.2 节中给出的 SAT 初始归约，并说明如何对其进行更新，以便归约 QSAT 博弈。核心思路是让 Nemesis

控制逃亡者在每个全称变量上的路径，从而模拟全称玩家的力量。为了解释这一构造，我们采用电学类比。

一种初步想法是在每条对应于全称变量的路径（$true$路径和 $false$ 路径）上引入一条重数为 1 的边，供 Nemesis 用作保险丝。然而，这种简单的方法可能会让 Nemesis 有能力摧毁两条并联保险丝，从而无论逃亡者如何选择，她都能获胜。因此，尽管这一想法提供了有用的直观理解，但它赋予 Nemesis 过多的力量，必须加以改进。为此，我们在全称变量的每条路径上都设置一条保险丝，并辅以两条推导路径。这样便构成了所谓的电气装置，如图9所示。

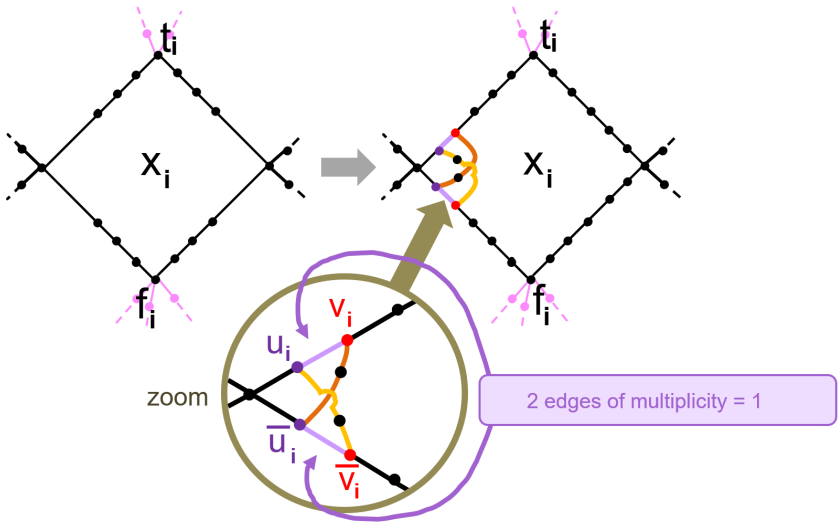


图9 用于模拟全称量词的电气装置。 左侧是用于将 3SAT 归约的初始变量构件。右侧是对其进行的更新，使 Nemesis 能够控制逃亡者所走的路径。

现在我们来详细描述这个构件。该构件由保险丝（重数为 1 的边）组成，分别放置在对应全称变量 x_i的路径的每个分支上，紧接第一条边之后。这两条保险丝分别连接 u_i 路径上的顶点 v_i 和$true$ ，以及 $false$ 路径上的顶点 u_i 和 v_i 。我们用另外两条长度为 2 的路径补全电气装置：一条从 u_i 到 v_i，另一条从 u_i 到 v_i。这两条路径上的所有边都不可破坏。

现在我们解释为什么该构件能够让 Nemesis 控制全称变量 x_i。首先回顾一下，在最初将 SAT 归约为 Nemesis 时，逃亡者到达主出口与 Nemesis 摧毁主出口边之间的竞速是十分紧迫的：如果逃亡者哪怕只输掉一个回合，他就会落败。现在回到电气装置，并且不妨设 Nemesis 已选择变量 x_i 的值为真，这意味着她打算阻止逃亡者通过 f_i。当逃亡者到达与变量 x_i 对应的环时，他的第一次移动必须前往 u_i 或 u_i。我们分析这两种情况。

- 逃亡者移动到 u_i。在这种情况下，Nemesis 不会摧毁保险丝 u_iv_i。随后，逃亡者有两种可能的移动方式。如果他前往 v_i，就等于设置了 $x_i = true$，正如 Nemesis 所预料的那样。相反，如果他朝 v_i 移动，就会比直接沿着经过 u_i 的路径晚一个回合到达 v_i

。按时到达主出口是一个苛刻的条件，因此这条绕行路线必然会导致失败。往回走也会导致失败。因此，逃亡者别无可行选择，只能按照 Nemesis 的安排，朝 t_i 前进。

- 逃亡者移动到 u_i。由于 Nemesis 已决定逃亡者必须经过t_i 而不是 f_i，她立即删除 u_i 和 v_i之间的保险丝。逃亡者不能折返，否则又会被判定失败。他唯一剩下的选择是移动到 v_i，从而朝着 t_i前进，这是 Nemesis 强加给他的路线。这条绕行路径让逃亡者损失一个回合，但 Nemesis 也花了一个回合删除保险丝。因此，这两个回合相互抵消，就时间而言，情况等同于原始 SAT 归约中的情况。

另一种可能的情况是，Nemesis 在逃亡者到达 u_i 或 u_i之前，删除与变量 x_i 相关的一条或两条保险丝。现在我们证明，她无法从这种策略中获益。Nemesis 提前花费一个回合删除至少一条保险丝，这让逃亡者可以选择使用不可破坏的绕行路径 u_i, v_i, f_i 或 u_i, v_i, t_i，而不会因此被判定失败。这使逃亡者占据优势，因此这种策略对 Nemesis 来说并非最优。

该分析表明，电气装置使 Nemesis 能够选择逃亡者是从 t_i 还是从 f_i通过；这一选择直接对应于全称变量 x_i 的取值。这种控制不会给 Nemesis 带来任何净成本。当然，她可以选择不行使这一权力，但不这样做并不会让她多获得回合来摧毁主出口边。因此，任何此类克制都毫无用处。电气装置赋予她对全称变量的控制，无法在博弈的其他地方加以利用。

多重图中的 Nemesis 的 PSPACE 困难性证明

现在我们证明，多重图中的 Nemesis 具有 PSPACE 困难性。构造用于编码 QSAT 实例的 Nemesis 图分为两步。首先，我们构造与 SAT 公式归约相对应的图，如第 4.2 节所述。其次，我们在所有全称布尔变量的入口处添加电气装置。

这种修改通过将 K 的值增加一个常数因子来增大变量环的规模，但除此之外不会影响构造的行为。

我们探讨归约的两个方向。

($\Rightarrow$) 假设该量化公式可满足。这意味着，在相应的 QSAT 博弈中，存在玩家有一个能满足所有 m 子句的获胜策略。如果逃亡者沿着变量构件做出相同的选择，他就能迫使 Nemesis 为 SAT 公式的每个 m 子句都花费一个回合来封锁子句出口。

对于全称变量，Nemesis 可以借助电气装置迫使逃亡者损失一个回合；但是，在每种这样的情况下，Nemesis 也必须花费一个回合来删除一根保险丝。因此，这些回合相互抵消，Nemesis 并未获得任何优势。由此可知，在 Nemesis 完成摧毁主出口边之前，逃亡者就能到达顶点 r 。此时，这条边的重数仍为 1，逃亡者得以逃脱并赢得博弈。换句话说，逃亡者拥有获胜策略。

($\Leftarrow$) 假设该量化公式不可满足。那么，在 QSAT 博弈中，全称玩家有一种策略，可以阻止存在玩家满足所有子句。Nemesis 采用相同的策略，阻止逃亡者接近所有 m 子句出口：必定至少有一个子句无法抵达。这个未能抵达的子句

使 Nemesis 多出一个回合，可用来摧毁主出口边。因此，当逃亡者完成从 s 到 r的路径时，通往主出口的边已无法通行。所以，Nemesis 必定赢得这场博弈。这意味着 Nemesis 有一种获胜策略。

至此，我们完成了多重图中的 Nemesis 为 PSPACE困难的证明。要证明其为 PSPACE 完全，还需说明 Nemesis 属于 PSPACE。由于 Nemesis 是一个有限的双人完全信息博弈，每个博弈局面都可以用多项式空间编码。此外，根据备注 7，如果逃亡者拥有获胜策略，那么他拥有一种不会两次经过同一顶点的策略。因此，只需考虑不重复经过顶点的对局，任何一局的长度都不超过图的顶点数。由此，可以通过对博弈树进行深度优先遍历，并根据当前玩家递归评估局面，来确定博弈的胜者。尽管这一过程可能需要指数时间，但只使用多项式空间，这表明多重图上的 Nemesis 属于 PSPACE。

结合这一结果与 PSPACE困难性，我们得出结论：多重图上的 Nemesis 是 PSPACE 完全的。

4.4 任意图上的 Nemesis 的 PSPACE 完全性

现在我们证明任意图上的 Nemesis 的 PSPACE 完全性（定理 5）。我们不从头重做证明，而是说明如何调整多重图上的 Nemesis 的 PSPACE 完全性证明，从而在简单图上得到相同的结果。

因此，我们的目标是从一个通过归约 QSAT 实例得到的复仇者实例出发，构造一个等价的复仇者实例，其底层结构是一个图（即，没有重边）。

首先，我们回顾通过 QSAT 归约得到的多重图实例的关键特征：

1. 该构造只使用四种不同的边重数。重数为 1 的边出现在电气构件中。出口边的重数为 $L+1$和 $2nK-m+1$。所有其余边均不可破坏：其重数可以设定为一个值 N，使其大于博弈的总回合数，因此 Nemesis 没有理由尝试删除这些边。这个值 N 关于原始 QSAT实例中的变量数和子句数是多项式级的。

2. 归约的一个关键性质是，逃亡者的必胜路径是紧致的。如果逃亡者在从 s 到 r的路径上哪怕只耽误一个回合，就再也无法在 Nemesis 摧毁主出口边之前到达r 。这种紧致性对于归约的正确性至关重要。

现在，我们将说明如何转换第 4.2 小节中归约后的多重图实例中的每个元素，以构造一个等价的简单图实例（转换构件如图 10 所示）。

- **顶点**。 多重图实例中的每个顶点（出口除外）都被替换为 N个副本。更具体地说，多重图中的一个顶点 u被替换为 N个顶点 u_1、.....、u_N。**不可阻断边**。 连接多重图中顶点 u 和 v 的不可阻断边，被替换为集合 $\{u_i\}^{1\leq i\leq N}$ 与集合 $\{v_j\}^{1\leq j\leq N}$之间的完全二部图。这样会产生 N^2条边，而非一条重数为 N的多重边。通过这种构造，如果逃亡者位于某个顶点 u_i，Nemesis 就无法阻止他到达 v的一个副本，因为阻断所有 N 种可能需要太多回合。这模拟了多重图实例中的不可阻断边。

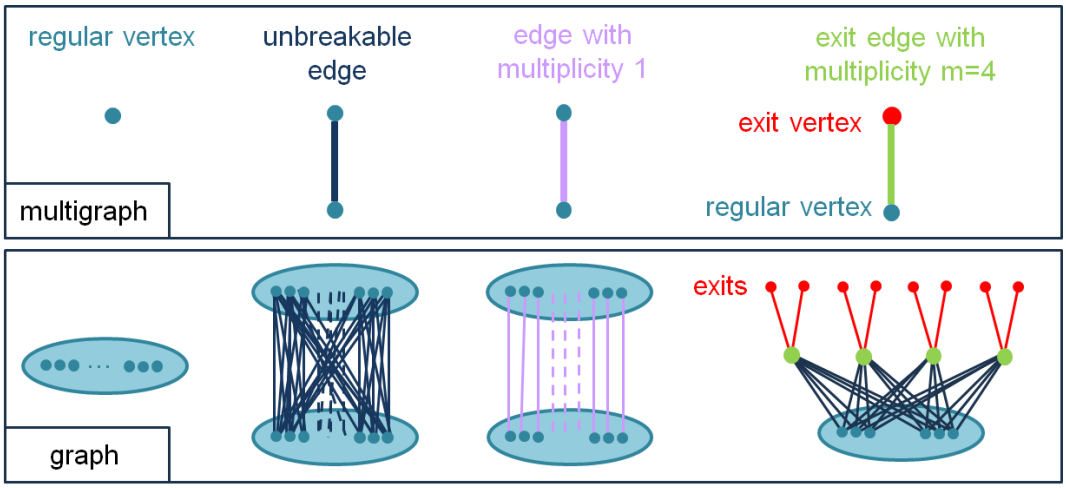


图10 用于将 Nemesis 多重图实例转换为等价 Nemesis 图实例的构件。 多重图中的每个普通顶点都会被转换为N 个该顶点的副本，其数量大于图的初始顶点数。不可阻断边会被转换为完全二部图。重数为 1 的边会被转换为匹配。重数为 k 的出口边则通过 k 个新的普通顶点进行编码（移除每个新普通顶点的一条出口边即可禁用出口）。

重数为 1的边。 多重图中连接顶点 u 和 v 的一条重数为 1 的边，被替换为一种匹配，它将每个 u_i 连接到对应的 v_i。这恰好产生 N 条简单边。

关键在于，如果逃亡者位于顶点 u_i，Nemesis 就能在一个回合内删除边 $(u_i,\ v_i)$。尽管逃亡者随后可以尝试经由 u_j抵达另一个副本v_j，Nemesis 仍可重复相同的操作。由于归约的时间安排十分紧凑，这种反复尝试必然会导致逃亡者失败。因此，当逃亡者位于 u_i时边 $(u_i,\ v_i)$ 被删除后，逃亡者必须沿另一条路径前进（在电气构件中，这对应一条不可阻断的推导路径，现在通过完全二部图实现）。

请注意，Nemesis 无法提前高效地删除这些边。如果她过早删除（$u_i,\ v_i$），逃亡者可以直接使用另一对（$u_j,\ v_j$）。这一限制并不影响归约的正确性。事实上，在多重图构造中，Nemesis 的获胜策略仅会在逃亡者位于重数为 1 的边的某个端点时删除这些边。因此，这模拟了多重图实例中重数为 1 的边。

任意重数的出口边。 为了替换连接顶点 u 与出口 t、重数为 k 的出口边，我们引入一个 出口构件。我们需要考虑到，多重图中的顶点 u 在图中由 N 个副本 u_i 表示。我们另外引入 k 个中间顶点 p_j，分别对应于 $1 \leq j \leq k$。每个顶点 p_j 恰好与两个出口相连。

该构件按如下方式模拟重数为 k 的出口边。如果 Nemesis 已经花费k 回合删除出口边，她可以通过删除与每个 p_j 相 incident 的一条出口边来停用该构件。在这种情况下，只要逃亡者到达某个 p_j，就可以立即删除剩余的出口边，从而阻止其逃脱。

另一方面，如果 Nemesis 在这个构件上使用的回合少于 k，那么

至少有一个顶点 p_j 仍保留着两条完整的出口边。当逃亡者位于某个 u_i时，他可以移动到这样的顶点 p_j，并在 Nemesis 作出反应之前逃脱。
Nemesis 可能会试图预先删除顶点 u_i 与 p_j 之间的边。不过，由于逃亡者可以从 N 个 u 副本中任选其一，他总能到达一个副本，使所有剩余的 p_j 仍然可达。因此，Nemesis 没有动机攻击这些边。

总体而言，这个 出口构件模拟了一条重数为 k的出口边。

QSAT 实例的归约分两步进行。首先，将 QSAT 实例归约为一个等价的多重图上的 Nemesis 实例。然后，将这个多重图实例转换为一个等价的简单图上的 Nemesis 实例。

通过将多重图中的每个顶点和每条多重边替换为上文所述对应的顶点构件和边构件，即可实现这种转换。每个构件都会模拟其在多重图中的对应部分的行为。

▶**引理 14**。 *将编码初始 QSAT 实例的多重图实例转换为图实例后，逃亡者或 Nemesis 存在获胜策略这一性质得以保留。*

因此，当且仅当逃亡者在多重图实例中有获胜策略时，他在生成的图实例中也有获胜策略。由于构造的图的规模相对于初始 QSAT 实例的规模是多项式级的，该归约可在多项式时间内完成。结合 Nemesis 在多重图上的 PSPACE 困难性，以及 Nemesis 属于 PSPACE 这一事实，可证明 Nemesis 在图上是 PSPACE 完全的（定理 4）。

4.5 赶猫问题的 PSPACE 完全性

赶猫问题的 PSPACE 完全性是带有两个出口的 Nemesis 的 PSPACE 完全性的推论。在赶猫问题中，逃亡者是一只猫，而 Nemesis 是牧猫者。不同于 Nemesis，其底层图没有出口。牧猫者的目标是用尽可能少的回合将猫困在单个顶点上，而猫的目标是尽可能多地保持自由。



图 11 由 无处可去 **桌游** [7]（由 Hank Atkins 创作并于 2012年 修订）似乎启发了赶猫问题 [3]。

为证明 PSPACE 完全性，我们考虑相关的判定问题：给定一个整数 k，猫是否有策略保持自由超过 k 回合？我们将 Nemesis 归约到赶猫问题。Nemesis 的一个实例由一个具有 m 条边的图 G、一组出口 E 以及一个起始顶点 s 给出。根据定理 9，我们不妨假设 $|E| = 2$。令 $\text{cat}_s(G)$ 表示猫从 s 开始时在 G 中能够保持自由的最大回合数。显然，$\text{cat}_s(G) \leq m$，因为牧猫者每回合删除一条边。

现在，我们从 G 构造一个 Cat Herding 实例 G'。图 G' 的构造方法是：在 G 的每个出口处连接一个团 K_{2m}，并将出口顶点与对应团中的一个顶点合并。每个团的规模关于 m 为多项式级，且足够大，以确保猫进入团后，即使牧猫者之前已删除团中的至多 m 条边，猫仍能自由活动超过 $2m$ 回合。

如果逃亡者在 G 中有获胜策略，那么猫就能在 G' 内到达两个团之一。进入团后，猫可以自由活动超过 $2m$ 回合，因此 $\text{cat}_s(G') > 2m$.

反过来，如果 Nemesis 在 G 中有获胜策略，那么牧猫者就能阻止猫到达任一团。这最多需要 m 回合。猫被限制在 G 后，牧猫者可以删除 G 中剩余的边，并在至多额外 m 回合内困住猫。因此，$\text{cat}_s(G') \leq 2m$.

由此可知，询问猫能否在 $2m$ 回合以上的时间内保持自由的判定问题，答案为“是”，当且仅当逃亡者在原始 Nemesis 实例中拥有获胜策略。这就证明了从 Nemesis 到 Cat Herding 的多项式时间归约。

最后，Cat Herding 是一种有限的、具有完全信息的双人博弈，且对局长度有多项式上界。因此，可以在多项式空间内判定胜者。由于 Nemesis 是 PSPACE 完全的（定理 5），我们得出结论：Cat Herding 是 PSPACE 完全的。

展望

我们介绍了 Nemesis 和 Blizzard，并指出一个颇为出人意料的事实：此前竟无人研究过这些极其简单的博弈。我们针对不同类别的图得出了一些复杂性结果，但仍有许多未解问题。能否证明 Nemesis 在平面简单图上是 PSPACE 完全的？平面图上是否存在多项式时间算法？或者，在出口数量固定的平面图上是否存在这类算法？Nemesis 和 Blizzard 的关键特点在于其简单性，这表明，理解它们的结构或许有助于深入认识其他问题，Cat Herding 问题的复杂性结果便是一个例子。最后，我们希望这些结果能够说明，图上的博弈看似简单，却能为复杂性理论提供丰富的研究空间，同时仍然妙趣横生。

参考资料

1 Martin Aigner 和 M. Fromme。警察与强盗游戏。离散应用数学，8(1):1–12，1984。2 Esther M Arkin、Aritra Banik、Paz Carmi、Gui Citovsky、Matthew J Katz、Joseph SB Mitchell 和 Marina Simakov。选择和覆盖彩色点。离散应用数学，250:75–86，2018。

N:26 Nemesis：图中的逃脱游戏

3 Rylo Ashmore。《赶猫：一种新颖的追逃博弈》。博士论文，纽芬兰纪念大学，圣让，加拿大，2025年。 4 Rylo Ashmore、Danny Dyer、Trent Marbach 和 Rebecca Milley。《切割、猫与完全图》。理论计算机科学，页码115393，2025年。 5 Rylo Ashmore、Danny Dyer 和 Rebecca Milley。《极值赶猫》，2025年。 arXiv:2505.07588。 6 Rylo Ashmore 和 Sophie Pinchinat。《在无限树上进行的赶猫游戏》。载于第45届IARCS软件技术与理论计算机科学基础年度会议（FSTTCS 2025），第10–1页。Schloss Dagstuhl–Leibniz-Zentrum für Informatik，2025年。 7 Hank Atkins。《无处可去》，2012年。 8 Elwyn R. Berlekamp、John H. Conway 和 Richard K. Guy。《数学游戏制胜之道》。学术出版社，1982 年。 9 Béla Bollobás 和 Imre Leader。《三维空间中的天使与魔鬼》。组合理论杂志 A辑，113(1):176–184，2006年。 10 Anthony Bonato 和 Richard J. Nowakowski。《图上的警察与强盗游戏》。美国数学会，2011年。 11 Édouard Bonnet 、 Florian Jamain 和 Abdallah Saffidine。《关于连通游戏的复杂性》。理论计算机科学，644:2–28，2016年。 12 Brian H Bowditch。《平面上的天使游戏》。组合学、概率与计算，16(3):345–362，2007年。 13 Nathan Bowler、Joshua Erde、Florian Lehner 和 Max Pitz。《以图的亏格界定其警察数》。SIAM离散数学杂志，35(4):2459–2489，2021年。 14 John H Conway。《天使问题》。无机会游戏，29:3–12，1996年。 15 Mark de Berg 和 Amirali Khosravi。《平面上的最优二元空间划分》。载于《计算与组合学》，第216–225页。施普林格柏林海德堡，2010年。 16 Shimon Even 和 Robert Endre Tarjan。《一个在多项式空间内完备的组合问题》。ACM学报（JACM），23(4):710–719，1976年。 17 Fedor V Fomin、Petr A Golovach、Alexander Hall、Matúš Mihalák、Elias Vicari 和 Peter Widmayer。《如何守卫一张图？》。载于《国际算法与计算研讨会》，第318–329页。施普林格，2008年。 18 Fedor V Fomin、Petr A Golovach、Jan Kratochvíl、Nicolas Nisse 和 Karol Suchan。《在图上追捕快速强盗》。理论计算机科学，411(7-9):1167–1181，2010年。 19 Fedor V Fomin、Petr A Golovach 和 Daniel Lokshtanov。《图上的守卫游戏：阻止入侵者进入！》。理论计算机科学，412(46):6484–6497，2011年。 20 Oddvar Kloster。《天使问题的一个解》。理论计算机科学，389(1-2):152–161，2007年。 21 Martin Kutz。《康威的三维天使》。理论计算机科学，349(3):443–451，2005年。 22 David Lichtenstein 和 Michael Sipser。《围棋是多项式空间困难的》。ACM学报（JACM），27(2):393–401，1980年。 23 Malaz Maamoun 和 Henry Meyniel。《关于警察与强盗游戏》。离散应用数学，17(3):307–309，1987年。 24 Marcello Mamino。《警察与强盗游戏的计算复杂性》。理论计算机科学，477:48–56，2013年。 25 András Máthé。《力量天使2获胜》。组合学、概率与计算，16(3):363–374，2007年。 26 Christos H Papadimitriou 和 Mihalis Yannakakis。《没有地图的最短路径》。理论计算机科学，84(1):127–150，1991年。 27 Alain Quillot。《图上的博弈与不动点》。博士论文，巴黎第六大学，1978年。国家博士论文。 28 Stefan Reisch。《Hex是PSPACE完全的》。信息学学报，15(2):167–191，1981年。 29 Johan Wästlund。《一个较弱的制胜天使》，2008年。