arXiv:2603.01393v1 [cs.CC] 2 Mar 2026

Hotaru Beam 的 NP完全性与物理零知识证明

Taisei Otsuji、Peter Fulla、Takuro Fukunaga

日本东京中央大学理工学部otsuji0024@gmail.com,
{fulla,fukunaga}@ise.chuo-u.ac.jp

摘要

Hotaru Beam 是一种逻辑谜题，其目标是仅绘制具有指定起点和弯折次数的线段，从而将放置在网格上的圆连接起来。零知识证明是一种通信协议，允许一方在不实际泄露某项信息的情况下，说服另一方自己掌握该信息。我们证明 Hotaru Beam 是 NP 完全的，并提出一种物理零知识证明（即可用实物实现），用以证明某人知道该谜题的一个解。

1 引言

零知识证明（ZKP）是一种由两名玩家——证明者和验证者——之间进行的通信协议。通过零知识证明，证明者可以说服验证者相信自己掌握某项信息，却不实际透露该信息。Goldwasser、Micali 和 Rackoff 提出了零知识证明的概念 [4]。此后，零知识证明一直被视为密码学和复杂性理论中的重要概念。

物理零知识证明是指人们使用日常物品即可执行的零知识证明。Gradwohl 等人 [5] 指出，即使非专业人士也能轻松理解物理零知识证明，从而提高协议的可实施性和可靠性。近几十年来，针对谜题的物理零知识证明一直是活跃的研究领域。这些谜题包括数独 [5], 、Nurikabe 和 Hitori [8], 、数回 [6], 、数链 [9], 以及桥梁谜题 [10]。Goldreich 等人[3] 证明，NP 中的任何问题都可以构造出零知识证明。虽然这一发现可用于为这些谜题构造零知识证明，但与 Goldreich 等人提出的方法相比，针对特定谜题设计零知识证明有望更加简单、高效。

Hotaru Beam 是一种逻辑谜题，由日本谜题公司 Nikoli 创作（参见维基百科词条 [11]）。代表萤火虫 （日语称“hotaru”）的圆圈被放置在网格上。谜题要求通过画线（代表光光束 ）连接所有萤火虫，并遵循若干规则，例如按指定次数转弯。

本文的主要贡献是开发一种使用卡片的物理零知识证明，用以证明某人知道 Hotaru Beam 谜题的一个解。尽管连通性约束是若干其他类型谜题（例如数链或桥）所共有的特征，且这些谜题已有物理零知识证明 [9, 10], ，但将弯折次数约束整合进来带来了独特的挑战。为解决这一问题，我们提出了两个创新想法。首先，我们提出一种在揭示弯折次数的同时实现光束的协议。其次，我们提出一种在实现光束的整个过程中跟踪萤火虫之间连接的协议。我们相信，这些想法可为设计针对各种其他几何对象的零知识证明提供有价值的启示。

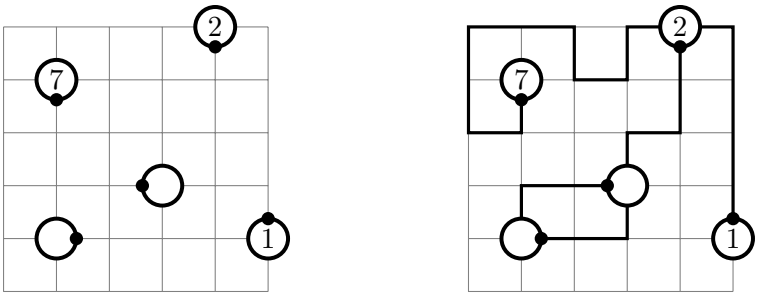


图 1：Hotaru Beam 的一个实例（左）及其一个解（右）

此外，我们证明了 Hotaru Beam 的 NP完全性。我们的证明通过对平面单调3-SAT问题 [1]进行归约来完成。

本文其余部分结构如下：第 2 节给出 Hotaru Beam 的形式化定义。第 3 节证明 Hotaru Beam 的 NP完全性。第 4 节介绍物理零知识证明的基本概念，第 5 节则详述我们为 Hotaru Beam 设计的协议。最后，第 6 节给出结语。

2 Hotaru Beam 的规则

Hotaru Beam 在矩形网格上进行。若干称为萤火虫的圆被放置在网格的交叉点上。对于每只萤火虫，在其边界与网格线的某个交叉点上放置一个点，圆内也可能有一个数字。图 1 的左侧展示了 Hotaru Beam 的一个实例。

任务是从每只萤火虫画出一条光束。光束从该点出发，沿网格线延伸，并在到达某只萤火虫的边界时停止（可能与其出发时所在的边界相同）。所画光束必须满足以下约束。

- 光束既不相交也不分叉；因此，一条光束不能终止于萤火虫的点，因为另一条光束（或同一条光束）会从该点出发。
- 萤火虫内部的数字规定了其光束的弯折次数。我们称之为弯曲约束。如果萤火虫内部没有数字，则弯折次数不受限制。
- 所有萤火虫必须通过光束最终连接成一个单一连通分量。我们称之为连通性约束。

我们将一条光束中的每一条单独直线段称为一条线段。满足这些约束的光束集合称为一个解。解的示例见 图 1 的右侧。

3 NP完全性

在本节中，我们证明如下定理。

定理1 它是 *NP* 完全的，即判定 *Hotaru Beam* 的一个实例是否存在解。

显然，给定一个解，我们可以在多项式时间内检查其有效性，因此该问题属于 NP。我们通过从平面单调3-SAT问题归约来证明其 NP 完全性 [1]。

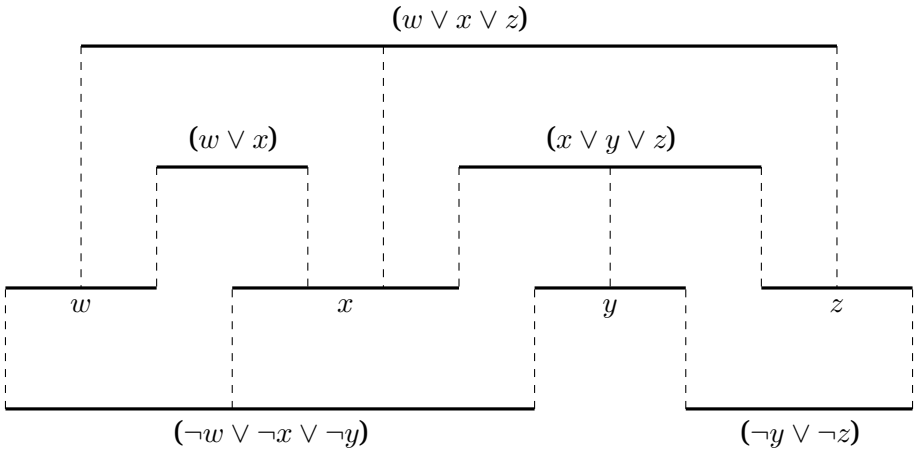


图 2：平面单调3-SAT问题的一个实例（为提高图示清晰度，垂直线段以虚线绘制）

3.1 平面单调 3-SAT

众所周知，3-SAT 问题由一个定义在一组布尔变量 $\{x_1, \ldots, x_n\}$ 上的合取范式公式 $C_1 \wedge \ldots \wedge C_m$ 给出，其中每个子句 C_i 都是至多三个文字的析取（即一个变量或其否定）。如果一个子句仅由正文字组成，我们称其为正 子句；类似地，如果一个子句仅由负文字组成，我们称其为负 子句。如果公式中的每个子句都是正子句或负子句，则称该公式为单调 公式。

平面单调3-SAT问题是3-SAT 问题的一个特例，其输入是以平面嵌入表示的单调公式。更具体地说，变量和子句分别用一条水平线段表示。变量对应的线段位于 x 轴上，而正（负）子句对应的线段位于 x 轴上方（下方）。变量出现在子句中的情况用连接这两者的一条竖直线段表示。除此之外，线段之间不允许相交。（示例见图 2。）

已知平面单调3-SAT问题是 NP 完全的 [1]。

3.2 归约

不失一般性，我们可以假设线段端点的坐标为整数，且取值范围长度为多项式级（相对于 m 和 n）。这样，在构造 Hotaru Beam 实例的网格时，我们只需照搬这些线段的位置即可。

我们用图 3 所示的构件替换变量线段，用图 4 所示的构件替换正子句线段，并用类似的构件（垂直翻转）替换负子句线段。阴影萤火虫的数量和位置经过选择，使其与垂直线段的交点相匹配。因此，对于每条垂直线段，都可以从相应的子句构件发射一束光束，使其抵达变量构件。注意，每个变量构件都会发射一束光束，从而将其连接到右侧的下一个变量。为了将最右侧的变量连接到最左侧的变量，我们添加一串弯折数为 0 的萤火虫，使其避开子句构件（例如，位于所有正子句构件的上方）。

如果两个交点距离过近，可能没有足够的空间展开构件。不过，这个问题很容易避免，例如可以假设

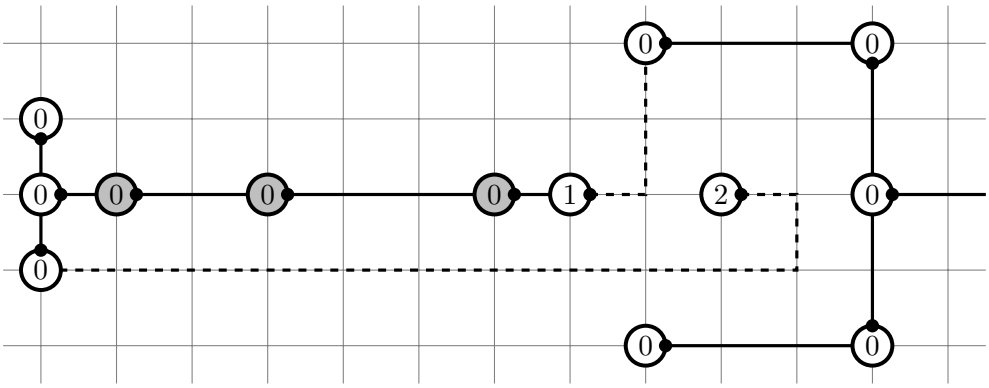


图 3：模拟变量的构件。弯折数为 0 的萤火虫发出的光束（用实线绘制）在任何解中都是固定的。注意，弯折数为 1 和 2 的萤火虫发出的光束只有两种绘制方式。用虚线绘制的方案对应于将变量设为真（另一种情况是对称的）。带阴影的萤火虫充当变量所出现的各个子句中的光束的潜在入口点。在本例中，只有正子句（置于上方）能够与该变量连接。

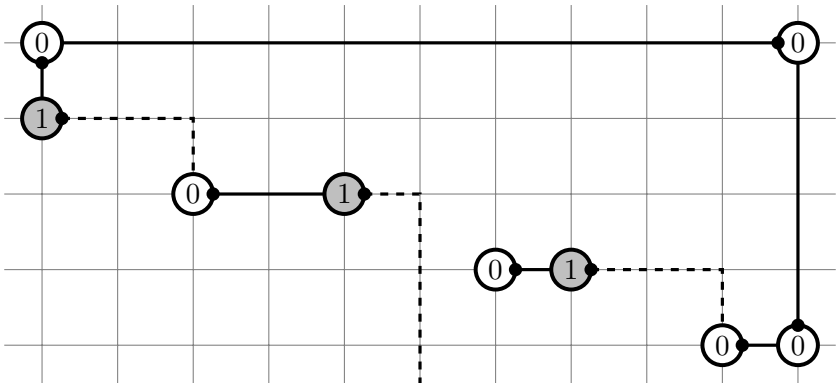


图 4：模拟正子句的构件。带阴影的萤火虫表示文字；从这些萤火虫发射的一组可能的光束用虚线绘制。在本例中，只有第二个文字会与模拟其对应变量的构件连接。

t 的坐标　　线段端点都是某个（足够大的）整数常数的倍数。

不难看出，所构造的 Hotaru Beam 实例有解，当且仅当给定的公式是可满足的。首先，考虑一个满足该公式的变量赋值。对于每个变量构件，我们根据赋予变量的值选择光束的绘制方式：当变量值为真（假）时，使其能够与上方（下方）的带阴影萤火虫连接。对于子句构件中每个带阴影的萤火虫（表示一个文字），如果所赋的值满足该文字，我们就绘制其光束，使其离开该构件并抵达对应的变量；否则，就绘制其光束，使其连接到同一子句构件的下一部分。注意，变量构件形成一个环，因此其中的所有萤火虫都彼此连接。由于该变量赋值满足公式，对于每个子句，至少有一个满足的文字，因此子句构件中的所有萤火虫最终都与变量构件中的萤火虫连接。

对于另一个方向，考虑 Hotaru Beam 实例的一个解。我们根据变量构件中光束绘制的选择来定义变量赋值（与上文相同）。

由于所有萤火虫都必须连通，因此每个子句构件至少有一束光束射出。根据构造，这样的光束只能由一只被着色的萤火虫（代表一个文字）射出，并且相应的变量必须赋值为使该文字成立。因此，该赋值满足该公式。

4 物理零知识证明的预备知识

4.1 物理零知识证明的定义

零知识证明是一种供两名玩家进行通信的协议：一名证明者（我们称他为 Peter）和一名验证者（我们称她为 Vera）。在我们的情形中，两名玩家都会看到一个 Hotaru Beam 实例 I，而 Peter 还会拿到 I 的一个解。他的目标是在不泄露任何相关信息的情况下，说服 Vera 相信他知道一个解。

玩家按照协议规定交换一系列消息。最后，Vera 会接受或拒绝 Peter 的主张。该协议必须满足以下三项性质：

- 完备性：如果 Peter 知道一个解，那么 Vera 就会接受这一主张。
- 可靠性：如果 Peter 不知道一个解，那么 Vera 就会拒绝这一主张。
- 零知识性：除了 Peter 拥有一个解这一事实之外，Vera 不会从通信中获得任何额外信息。

一般来说，玩家的行为可以随机化，并且可靠性条件会被放宽，使得 Vera 可能以很小的概率接受一个假声明。然而，我们的协议是确定性的，不允许出错。

确立零知识性质的一种常见方法是提供一个算法（称为模拟器），该算法在无法获知解的情况下，生成一系列与协议真实运行期间所交换的消息不可区分的消息。在我们的情形中，模拟器的存在将平凡地由每条消息都仅直接依赖于实例 I 这一事实得出。

至于物理零知识证明，该协议可以由人使用日常物品来执行。我们提出一种针对 Hotaru Beam 的基于卡片的物理零知识证明。一张卡片的正面要么显示图案（♣ ♠ ♡ ♢），要么显示数值（1，2，…），而所有卡片的背面彼此无法区分。只有当卡片正面朝上时，Vera 才能看到卡片的正面。然而，Peter 可以自由检查任何卡片，因此我们不妨假定他能看到所有卡片的正面。为清晰起见，我们将背面朝上的牌表示为 ♣ ♠ ♡ ♢ 1，2，…。

玩家按照我们的协议操作卡片：他们可以重新排列已经摆放好的卡片、将卡片翻面、丢弃卡片或加入新卡片。Vera 负责核验她能看到的卡片正面是否显示了协议要求的图案或数值。例如，当 Peter 被要求从一堆面朝下的卡片中丢弃一张 ♡ 卡片时，他会先让 Vera 确认卡片正面，然后才真正将其移除。

4.2 基本扑克牌操作协议

在本小节中，我们介绍几种基本的扑克牌操作协议，它们作为子程序用于我们的零知识证明中。

4.2.1 牌堆移位洗牌

牌堆移位洗牌协议（由 Nishimura 等人 [7] 提出）接收一组大小相同的背面朝上的牌堆 P_0，P_1，……，P_{k-1}，然后按一个随机均匀选取的量 x 进行循环移位，移位量从 $\{0, 1, \ldots, k-1\}$ 中选取。换句话说，得到的序列为 P_x，$P_{(x+1) \bmod k}$，……，$P_{(x+k-1) \bmod k}$，原序列的任意循环移位出现的概率都相同。玩家不会得知 x 的值。

为了在实际操作中执行该协议，双方玩家轮流将序列任意移位，直到无人能够实际追踪总移位量为止。在我们的零知识证明中，洗牌是唯一会产生随机性的环节（更准确地说，是玩家对卡片状态的认知丢失）。

4.2.2 牌堆选择

牌堆选择 协议（由 Dumas 等人 [2] 以卡片选择 协议的形式提出）允许 Peter 对从序列中选出的一叠牌执行操作，而不向 Vera 暴露选中了哪一叠。给定一列背面朝上的牌堆（每堆大小相同），协议按以下步骤进行。

1. Peter 将一张♠卡片放在他选中的牌堆上，并在其余每堆上各放一张 ♣ 卡片，以此标记选中的牌堆。（Vera 看不到放置的卡片正面。）2. Vera 将一张 ♡ 卡片放在序列中的第一堆上，并在其余每堆上各放一张 ♢ 卡片，以此标记第一堆。随后将这些卡片翻至背面朝上。3. 对序列进行一次牌堆移位洗牌。Peter 通过找到♠卡片来识别他选中的牌堆。他移除顶部的两张卡片（这两张卡片是在前两步中放置的），对该牌堆剩余的卡片执行预定操作，然后将那两张卡片放回顶部。再次对序列进行牌堆移位洗牌。4. Vera 将每堆最上方的卡片翻至正面朝上，进行循环移位，使标有♡卡片的牌堆回到原来的位置（序列中的第一堆），然后移除每堆顶部的两张卡片。

请注意，零知识性质得以满足，因为 Vera 可观察到的结果（操作是对哪一堆执行的，以及 ♡ 卡片的位置）均匀分布。

4.2.3 可逆牌堆移位洗牌

该协议扩展了堆选择协议，允许 Peter 对序列中选定的若干卡片堆执行多项操作。Vera 会知道这些牌堆的相对位置（即它们沿环排列时彼此之间的距离），但不知道它们在输入序列中的绝对位置。换言之，Peter 可以对经过牌堆移位洗牌的序列执行操作，然后撤销洗牌（使 Vera 相信序列已恢复原始顺序）。

其实现实际上与牌堆选择协议相同。尽管始终使用这一（更通用的）版本也是可行的，但调用特殊情况（即选择单个牌堆，而不是执行可逆洗牌）有时会更直观。

4.2.4 证明集合成员关系

该协议可用于使 Vera 相信，背面朝上的牌 c 的正面属于某个集合 S，而不会透露实际的图案/数值。

将一列带有正面 S 的 $|S|$ 张新卡片正面朝上摆放。Vera 确认其正面后，将卡片翻至背面朝上，并进行牌堆移位洗牌。Peter 将卡片 c 与序列中正面与 c 相同的卡片交换。再对该序列进行一次牌堆移位洗牌，然后将卡片翻至正面朝上。最后，Vera 确认序列中的卡片正面仍为 S。

4.3 逻辑值的表示

我们用卡片的有序对表示逻辑值: 🂣 🂱 对应于真和 🂱 🂣 对应于假。为方便起见，我们将表示真（假）的一对记作T (F)。虽然看起来像一张卡片，但请注意，它实际上是一个有序对。

我们还引入一种协议，使 Peter 能够将两对背面朝上的逻辑卡片对 b_1 和 b_2 替换为它们的析取 $b_1 \vee b_2$，同时不向 Vera 泄露任何有关这些值的信息。协议步骤如下。

1. 首先，对 b_1 和 b_2 进行随机洗牌，使玩家无法区分它们。然后，Peter查看卡片的正面，并从 b_1 和 b_2中选择一个。如果 b_1或 b_2 中有一个表示真，则Peter选择它。如果 b_1 和 b_2 都表示真，或者都表示假，则Peter任意选择其中一个。所选的一对被放在第一行。在整个过程中，卡片的正面始终保持隐藏，从而防止Vera确定选择了哪一对。

2. 准备另一对卡片🂡 🂨，并将其放在第二行。在卡片翻转为背面朝下之前，Vera确认卡片的正面。

3. 该p ile-shifting洗牌得以应用，此时将垂直排列的卡片视为一个pi le.

4. 翻转第一行的卡片。任一玩家复制第一行中相同的逻辑牌对，并将其放在第三行。换言之，第一行被复制到第三行。双方玩家都可以确认第一行和第三行表示相同的逻辑值。

5. 将第一行和第三行中的卡片翻回正面朝下后，对所有行进行牌堆移位洗牌，以打乱各列。

6. 翻开第二行中的卡片，并移动牌堆，使第二行显示🂡 🂨 此时，第一行和第三行与原来的第一行相同。

7. 最后，输出第一行和第三行中的卡片对。

如果 b_1 和 b_2 都为假，那么步骤 1 中的第一行也表示假。因此，在这种情况下，输出的两组卡片对都为假。如果 b_1 或 b_2 为真，且证明者在步骤 1 中将其放入第一行，那么输出的两组卡片对都为真。在这些情况下，该过程实际上是将输入的逻辑卡片对替换为表示其析取的卡片对。

另一方面，当 b_1 和 b_2 中恰好有一个为真时，验证者无法察觉证明者是否将假卡片对放在第一行，这不符合步骤 1 中规定的行为，

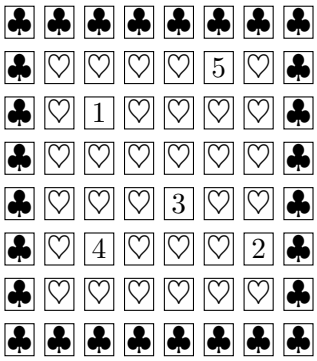


图 5：图 1 中实例的卡片表示

导致用假卡片对替换 b_1 和 b_2。然而，在我们的 Hotaru beam 零知识证明中使用时，这种差异并不会造成问题，因为这种行为只会使证明者处于不利地位。

该协议的零知识性显而易见，因为验证者的观察与证明者的行为无关。

5 Hotarubeam 的物理零知识证明

在本文余下部分中，我们用 n 表示萤火虫的数量，用 h 表示棋盘的高度（即纵轴上的网格点数量），用 w 表示棋盘的宽度（即横轴上的网格点数量）。我们还任意确定萤火虫的排列顺序。

在我们的协议中，Peter 将使用卡片对他的解进行编码。他将从第 i 个萤火虫开始，依次嵌入光束，直到 $i = 1, \ldots, n$。每次嵌入后的状态都通过两种结构记录：棋盘和连接表，下面将对此进行介绍。

5.1 棋盘

棋盘最初通过按图 5 所示摆放卡片来表示。形式化地说，我们摆放与网格点对应的 $w \times h$张卡片；包含萤火虫的网格点用相应的数字卡表示（ i第只萤火虫对应i ），其余网格点则用 $\boxed{\heartsuit}$张卡片表示。随后，用由 $\boxed{\clubsuit}$张卡片组成的边界围住这些 $w \times h$ 张卡片。我们使用$\boxed{\heartsuit}$ 张卡片表示某个网格点仍可供光束通过；在协议过程中，其中一些卡片将被替换为 $\boxed{\clubsuit}$ 张卡片，因为我们会依次嵌入解中的光束。

我们将这些卡片统称为棋盘。Vera 确认棋盘的初始排列后，所有卡片都会被翻面朝下。

5.2 连接表

除了棋盘之外，我们的协议还维护一个用于编码萤火虫之间连接关系的卡片集合，我们称之为连接表（参见图 6）。它由 n 列组成，每只萤火虫对应一列。 i列由一张 i 卡片和 n对逻辑卡片组成。起初，我们在第T 对中使用 i，其余卡片对则使用F 。

1	2	3	4	5		
T	F	F	F	F		
F	T	F	F	F		
F	F	T	F	F		
F	F	F	T	F		
F	F	F	F	T		

图 6：5 只萤火虫情况下连接表的初始状态

Vera 确认连接表的初始排列后，所有卡片都会被翻面朝下。

设 G 是一个具有 n 个顶点（代表萤火虫）的无向图，其边对应于截至该时刻已嵌入的光束。整个协议过程中，以下不变量始终成立：如果连接表第 j 个逻辑牌对在第 i 列中的值为 T，那么 Vera 可以确定萤火虫 i 和 j 属于 G的同一个连通分量。 □

5.3 嵌入光束段

本小节将介绍线段嵌入协议。Peter 将使用该协议，通过嵌入其解的一部分（一个光束段）来更新棋盘。该协议的输入是一组由 k 张卡片组成的序列（棋盘的一行或一列）。其中一张卡片正面朝上（我们称之为起始卡），其余卡片均背面朝上。Peter 可以选择一个值 l，并替换 l张与起始卡相邻的连续卡片，前提是这些卡片都是♡ 卡片。示例见图7。

我们使用该协议的若干变体，它们在替换卡片的限制以及向 Vera 揭示的信息方面有所不同。首先介绍最基本的版本；其他变体的实现细节将在本节末尾讨论。

在基本设定中，Peter 必须选择一个值 $l \geq 1$。Vera 不会得知确切的值，但可以确信该值满足要求。（在某些变体中，Peter 也可以选择 $l = 0$，而 Vera 无法分辨两者。）Peter 可以选择替换起始位置左侧或右侧的卡片；不过，Vera 会得知这一选择。（在某些变体中，这一选择会对她保密。）从与起始位置相邻的卡片开始，Peter 将前 $l-1$ ♡ 张卡片替换为 ♣ 张卡片，并将最后一张 ♡ 卡片替换为一张 ♢ 卡片。（在某些变体中，最后一张♡ 卡片也会替换为一张 ♣ 卡片。在这种设定下，要求最后一张 ♡ 卡片之后紧接着一张数字卡。Vera 会知道这一点，但不会得知该数字卡的值。）

假设 Peter 选择了一个值 $l \geq 1$，并希望替换起始位置右侧的卡片（替换左侧卡片的情况与此对称）。首先，我们构造一个由 k 张卡片组成的辅助序列，称为遮罩序列（见图9中的底行）。构造方法如下：

1. 玩家按照图 8 第一行的方式摆出一串新卡片。Vera 确认卡片正面后，将这些卡片全部翻至背面朝上。2. 使用牌堆移位洗牌协议对这串卡片进行洗牌。随后，Peter 再进行一次循环移位，使 ♢ 卡片移至从左数第 l 个位置，如图 8 第二行所示。

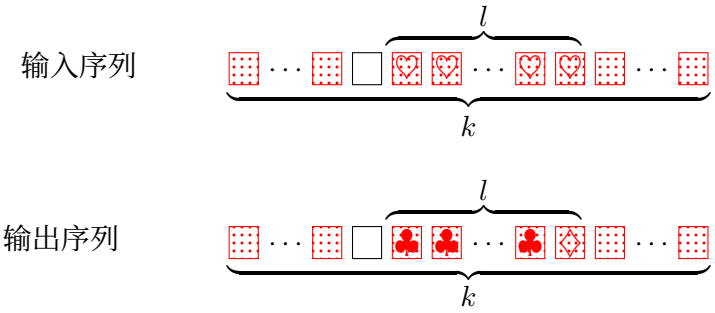


图7：线段嵌入协议的输入和输出序列

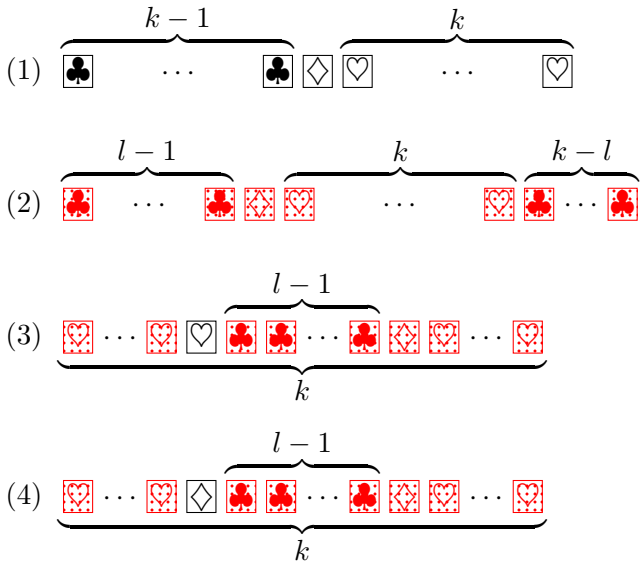


图 8：逐步构造掩码序列

3. 丢弃第 k 张卡片右侧的卡片，然后对剩余序列进行牌堆移位洗牌。Peter 将♢卡片翻至正面朝上，向 Vera 展示它仍在序列中。然后他将其翻回背面朝上，再进行一次牌堆移位洗牌，并将 ♢ 卡片左侧第一张 ♡ 卡片翻至正面朝上，如图 8 第三行所示。

4. Peter 将正面朝上的 ♡ 卡片替换为一张 ♢ 卡片。然后，他向 Vera 证明（使用集合成员关系证明协议），右侧的下一张卡片是 ♣ 卡片或 ♢ 卡片。

请注意，Vera 可以确定所得的掩码序列对于某个 $l \geq 1$ 而言与图 8 最后一行所示相同，但她无法获知关于 l 的值的任何其他信息。

掩码序列被循环移位并放置在输入序列下方，使正面朝上的 ♢ 卡片与起始卡对齐（见图9）。考虑任意一列正面朝下的列（一对垂直对齐的卡片），并注意它必须至少包含一张 ♡ 卡片。对于每一个这样的列，Peter将该对卡片洗牌，并从中取出一张♡ 卡片。他允许Vera核验被取出卡片的正面，然后将其丢弃。剩余的卡片被放回顶行。最后，与起始卡对齐的 ♢卡片也被丢弃。结果如图7所示。

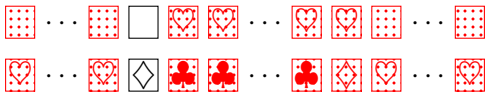


图9：构造掩码序列后卡片的排列

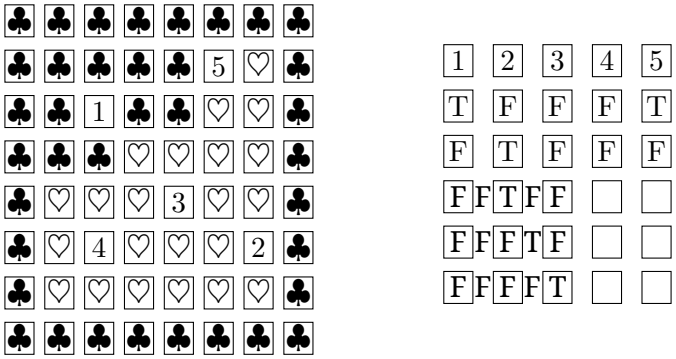


图10：将从第 1 只萤火虫到第 5 只萤火虫的光束嵌入棋盘（左）和连接表（右）

最后，我们讨论在我们的协议中使用的嵌入一段光束的另外几种变体。

- 隐藏替换的方向Peter 准备两个版本的掩码序列：一个用于替换起点左侧的卡片，另一个用于替换起点右侧的卡片。然后，他秘密选择所需的版本（进行洗牌），并用它进行替换。另一个版本则被丢弃。

- 允许 $l \geq 0$如果 $l = 0$，实际上并不会发生替换。这可以通过一个由 $k - 1$ ♡张卡片和一张 ♢卡片组成的掩码序列来模拟。为了使这种情况无法被区分，Peter 还会为某些 $l' \geq 1$ 准备一条多余的掩码序列，并秘密选择所需的那一条。

- 嵌入最后一条线段光束的最后一条线段终止于一只萤火虫，因此第 l 张卡片是数字卡，而不是♡卡片。Peter 不会从该列揭示并弃掉一张 ♡卡片，而是将这张面朝下的数字卡收起来。如下一节所述，Peter 可以在不透露数字卡数值的情况下，让 Vera 相信这确实是一张数字卡。

5.4 嵌入光束

假设 Peter 想要嵌入一条光束，该光束从第 s 只萤火虫出发，终止于第 t 只萤火虫。这需要修改棋盘（将光束经过的网格点上的 ♡张卡片替换为 ♣张卡片），并修改连接表（将第 s 列中的第 t 个逻辑牌对设为 T）。示例见图10。

如果光束完全不允许弯曲（即第 s 只萤火虫的内部为 0，或其点旁边的网格点包含另一只萤火虫），那么它在任何解中的绘制方式都相同。因此，没有必要隐藏其绘制图案，

Peter 可以进行必要的修改，让 Vera 看到这些图案。在下文中，我们假设这种特殊情况不适用。

首先，Peter 复制连接表中第 s 列的逻辑卡片对（不向 Vera 揭示实际值）。我们将复制得到的序列记为 C。

嵌入光束可以通过将其拆分为若干直线部分（线段），并迭代应用线段嵌入协议来完成。我们先考虑待嵌入的光束具有预定弯折次数 $b \geq 1$ 的情况。我们在弯折处将光束拆分，从而得到 $b+1$ 条线段。

假设第一条线段的方向是水平的，另一种情况与此类似（注意，方向由点的位置决定，因此 Vera 知道该方向）。Peter 将对包含第 s 只萤火虫的那一行棋盘应用线段嵌入协议，以此嵌入这条线段。他将 s 张卡片翻至正面朝上，将其指定为起始卡，将 l 设为线段的长度（$l \geq 1$），并根据点的位置公开选择替换方向（向左或向右）。应用该协议时，他用 ♣ 张卡片替换前 $l-1$ ♡ 张卡片，并用一张 ♢ 卡片替换最后一张 ♡ 卡片。这张 ♢ 卡片在嵌入下一条线段时充当起始卡。

其余线段的方向在垂直和水平方向之间交替。对于垂直（水平）线段，Peter 使用堆选择协议拿起棋盘中包含该 ♢ 卡的 X 列（行），且不向 Vera 泄露其位置。随后，他对 X 应用可逆牌堆移位洗牌，并将 ♢ 卡翻至正面。他令 l 等于待嵌入线段的长度（$l \geq 1$），秘密选择替换方向，并应用相应的线段嵌入协议变体。然后，他撤销对 X 执行的洗牌，并将其放回棋盘。

嵌入最后一条线段时，Peter 令 l 等于其长度减一（$l \geq 0$），并将所有 ♡ 卡替换为 ♣ 卡。我们将紧跟在最后一张被替换的卡之后的那张卡记为 f （实际上它是t）。Peter 需要向 Vera 证明 f 是数字卡，但不能透露它的数值或位置。这可以参照集合成员关系证明协议来完成：Peter 使用堆选择协议拿起连接表的第 t列，并将其第一行中的 t 与 f交换。对于该列中的每个逻辑牌对 p，他对 p 及其在 C中对应的逻辑牌对 p' 应用逻辑值复制协议。如果$p =$ 为 F，他就将 p 替换为 p'；否则，将 p' 替换为 p。换言之，操作完成后，p等于两个原始逻辑值的析取。这确保第 t列中的第 s个逻辑牌对被设为 T，因为第 s列中的第 s个逻辑牌对初始化时就被设为 T。将第 t列放回连接表后，Peter 让 Vera 确认顶行仍包含卡片 1、.....、 n。现在 Vera 可以确信 f确实是数字卡，而且 Peter 已在适当的列上执行了逻辑值复制协议。尽管她并未获知这些值及其替换方式，连接表的不变量仍得以维持，因为涉及的萤火虫（s和 t）刚刚由一束光束连接起来。

如果实例没有规定光束弯折的次数，就必须对 Vera 保密。Peter 通过添加长度为 0 的虚拟 线段，使线段数量变为 $w \cdot h$。在这种情况下，他会使用线段嵌入协议的变体，以允许 $l = 0$。请注意，即使光束完全没有弯折，也必须至少将其分成两段（非虚拟线段），因为我们会以特殊方式处理第一段和最后一段。

5.5 协议概述

首先，按照第 5.1 节和第 5.2 节所述，摆放棋盘和连接表卡片；Vera 确认卡片正面后，将所有卡片翻面朝下。然后，Peter 依次嵌入

从 s号萤火虫出发的光束，适用于 $s = 1, \ldots\ldots, n$。最后，他通过反复执行以下操作，确保连接表中的所有逻辑卡片对都为T 。

假设连接表中的两列（例如 i 和 j）在某一行（例如 k）中都有一张 T 逻辑卡片对。根据不变量，这意味着萤火虫 i和 j 属于 G的同一个连通分量，因此 Peter 可以像嵌入光束的最后一段时那样，用这两列的逐行析取替换这两列。

使用堆选择协议选取参与操作的列。由于图 G 是连通的，通过至多进行 T 次此操作，就能将连接表中的所有逻辑卡片对变成 n^2 。（Peter 总会恰好进行 n^2 次操作，以隐藏所需的数量。）最后，Vera 将连接表中的卡片翻至正面朝上，并检查卡片正面。根据不变量，这使她确信连通性约束已得到满足。

6 结论

我们证明了 Hotaru Beam 的 NP完全性，并为其提供了一种物理零知识证明。我们的零知识证明包含两个新颖的思路：线段嵌入协议和连接表。我们认为，这些思路可能对其他协议有所帮助，尤其是用于网格棋盘上的谜题以及证明特定图存在的协议。

致谢

本研究得到了 JSPS KAKENHI 项目（编号：JP20H05965、JP23K21646 和 JP21K11759）的资助。

参考文献

[1] Mark de Berg与Amirali Khosravi，平面线段的最优二叉空间划分，《国际计算几何与应用杂志》**22**（2012），第3期，187–206。[2] Jean-Guillaume Dumas、Pascal Lafourcade、Daiki Miyahara、Takaaki Mizuki、Tatsuya Sasaki与Hideaki Sone，*Norinori*的交互式物理零知识证明，计算与组合数学——第25届国际会议，COCOON 2019，中国西安，2019年7月29日至31日，会议录，《计算机科学讲义》，第11653卷，施普林格，2019，第166–177页。[3] Oded Goldreich、Silvio Micali与Avi Wigderson，对*NP*中所有语言仅证明其有效性的证明具有零知识证明系统，《ACM学报》**38**（1991），第3期，691–729。[4] Shafi Goldwasser、Silvio Micali与Charles Rackoff，交互式证明系统的知识复杂性（扩展摘要），第17届年度ACM计算理论研讨会会议录，1985年5月6日至8日，美国罗德岛州普罗维登斯，美国计算机协会，1985，第291–304页。[5] Ronen Gradwohl、Moni Naor、Benny Pinkas与Guy N. Rothblum，数独谜题解的密码学与物理零知识证明系统，《计算系统理论》**44**（2009），第2期，245–268。

[6] Pascal Lafourcade、Daiki Miyahara、Takaaki Mizuki、Tatsuya Sasaki与Hideaki Sone，数回的物理零知识证明：如何进行物理保拓扑计算，信息安全实践与经验——第15届国际会议，ISPEC 2019，马来西亚吉隆坡，2019年11月26日至28日，会议录，《计算机科学讲义》，第11879卷，施普林格，2019，第135–151页。[7] Akihiro Nishimura、Yu-ichi Hayashi、Takaaki Mizuki与Hideaki Sone，基于卡片的协议的移堆洗牌，电子、通信与计算机科学基础学会志 **101-A**（2018），第9期，1494–1502。[8] L´eo Robert、Daiki Miyahara、Pascal Lafourcade与Takaaki Mizuki，连通性的交互式实体零知识证明：在*Nurikabe*与*Hitori*中的应用，欧洲可计算性会议，施普林格，2021，第373–384页。[9] Suthee Ruangwises与Toshiya Itoh，数链谜题与k条顶点不相交路径问题的物理零知识证明，新一代计算 **39**（2021），第1期，3–17。[10] Suthee Ruangwises与Toshiya Itoh，连通生成子图的实体零知识证明：在桥梁谜题及其他问题中的应用，非常规计算与自然计算——第19届国际会议，UCNC 2021，芬兰埃斯波，2021年10月18日至22日，会议录，《计算机科学讲义》，第12984卷，施普林格，2021，第149–163页。

[11] 维基百科：*Hotaru Beam*，https://en.wikipedia.org/wiki/Hotaru_Beam。