

基于整数规划的机器人路径优化通用解决方案

Shuai D. Han Jingjin Yu

摘要：整数规划 (Integer programming IP) 已经被证明在解决许多机器人路径优化问题上的效率很高。但是整数规划的应用通常使用一种特别的、特定于问题的方式来完成。在这篇文章中，在大范围研究了基于路径的优化问题之后，我们描述了一种易于应用（简单的两步）并且对于计算时间和优化可获得性方面具有高性能的整数规划解决方案。我们通过应用于三个具有挑战性的基于路径的优化问题来演示我们的方法的通用性：多机器人路径规划问题 (MPP)，最小约束移除问题 (MCR) 以及奖励收集问题 (RCPs)。相关的实验显示这个方法能够有效地生成对于大型状态空间，复杂的限制情况以及复杂目标函数的（近似）最优解。结合 IP 方法的提出，我们引入了两种新的现实的机器人问题：多机器人最小约束移除 (MMCR) 和具有部分解的机器人路径规划问题 (MPP)，它们能够通过我们提出的 IP 流水线解决方案被快速高效地解决。

1. 引言

机器人任务和运动规划问题的研究目的是找到一条（或多条）路径，使单（或多）机器人能够优化某些累积成本或报酬。有些问题环境允许高效的基于搜索的算法，例如，通过动态规划，这些问题往往是计算上难以解决的 [1] [2]。在这些情况中，有两种被经常应用的方法：1. 设计时间多项式算法以近似计算优化解，2. 应用贪心搜索，以启发式算法为辅助。这两种方法在实际应用中都有各自的缺陷：前者并不总是确保良好的最优性，而后者在问题实例变得更大时通常无法很好地扩展。

在这篇文章中，我们描述了一种整数规划 (IP) 方法作为一个第三种通用解决方案以解决路径优化问题。构建基于整数规划的解决方案的关键是构建一个由变量和不等式组成的模型，该模型对目标问题的所有约束进行编码。对于优化路径，我们注意到，将模型构建过程划分为路径编码为第一步，然后是添加优化约束为第二步，这是很自然的。通过这个方法，我们可以很容易地解决许多独特的和具有挑战性的基于路径的优化问题，包括多机器人路径

规划问题 (MPP)，最小约束移除问题 (MCR)，和以及奖励收集问题 (RCPs)。经过广泛的评估，IP 方法通常在计算时间和解决方案的最优性方面具有竞争力。结合 IP 方式的提出，我们引入了两个新的机器人问题：多机器人最小约束移除问题 (MMCR) 和多机器人部分解路径规划问题 (MPP)。这些问题是 MCR 和 MPP 的自然衍生，它们具有实用性，但是在计算上更具有挑战性。

相应的工作：整数规划 (IP) 方法被广泛运用于从一开始就解决组合优化的挑战 [3] [4]，应用于一系列衍生的旅行商问题 (TSP) [5]、网络流的问题 [6]、多目标寻迹等 [7] 问题。更多的近期研究已经将整数规划应用于比如包括多机器人应用问题的机器人路径优化问题 [8]-[10] 以及机器人操作问题 [11] [12]。

本文由使用整数规划的一系列工作所启发和构建，IP 模型算法在多机器人路径规划 (MPP) [10] 中的应用取得了令人惊讶的初步成功，在优化 MPP 方面实现了性能飞跃。MPP 是被应用于广泛领域的重要问题，包括撤离问题 [13]，组队形成 [14] [15]，定位 [16]，microdroplet 操控 [17]，目标移动 [18]，搜寻和搜救 [19]，以及人机交互 [20]。在过去的十年中，在基于图环境的离散 MPP 问题的优化求解上已经取得了重大的进展。对这样一个离散 MPP 问题的算法解决方案是通过减少其他问题来得到解决 [10], [21], [22]。基于解耦的启发算法也被证明是有用的 [23]-[25]。和本文中被证明的部分解部分相似，一份最近的文章 [26] 给出了一个基于搜索的求解器，此求解器优化了在有限时间内到达目标的机器人数量。

为了阐述我们方法的通用性易用性，我们也研究了最小化约束移除问题 (MCR) 和一系列的机器人奖励收集问题 (RCPs)。MCR 需要寻找一条移除最少阻塞障碍物的路径，它对应于基于约束的任务和运动规划 [27] [28]，目标重组 [12] [29]，和控制策略设计 [30]。两个基于搜寻的求解器在 [1] 中被提供以拓展到有权重的障碍物 [31]。平衡两个优化问题：路径长度和计算时间的方法是存在的 [32]。最近的基

于 MCR 的研究减少了优化问题的上下限之前的差距 [33]。奖励收集问题 (RCPs) 大部分情况下是关于在不超过某些成本 (比如时间和距离) 下的奖励收集。有许多这样的问题包括境地哪旅行商问题 (TSP) [34] 和定向问题 (OP) [35]。我们关心面向更复杂的变体 [36], [37] 与非添加的优化目标相关。这样的问题模型面向信息收集任务, 比如精准农业 [38], 检测海洋的环境变量 [39], 以及基础设施检查 [40]。这些问题基本至少是 NP 困难的 [1] [35] [41]。

贡献：这篇文章主要有两个主要的贡献：

- 基于之前的研究, 我们提出了一个对于基于机器人路径优化问题的通用整数规划 (IP) 解决框架。两步的流水线框架易于应用并且经常产生高度优化的解决方案。
- 我们规划多机器人最小约束移除 (MMCR) 问题和多机器人部分解路径规划 (MPP), 分别作为 MCR 和 MPP 的实用衍生。结果表明, IP 方法可以有效地解决这些新问题。

除了主要的贡献, 这篇文章提供了许多额外的, 对于特定问题的启发式算法, 大大改善了 baseline 的 IP 形式表现; 这些启发式算法中的一部分也是通用地可应用的。除非被明确提到并引用, 这些在文中阐述的强化算法 (比如启发式算法) 也是第一次在这里被第一次提出的。然而基于 IP 的方法已经被应用于机器人, 基于我们的知识, 这篇文章是第一个总结了一个通用的 IP 框架, 它可以很容易地应用于各种基于路径的优化问题, 并提供了完整的基于模拟的实验评估。

组织结构：本篇文章的剩余部分组织如下：在第二部分, 我们系统地定义了 MPP, 多机器人 MCR 和 RCP。在第三部分, 我们概述了一般的 IP 解决方案方法, 并描述了两种经过验证的路径编码方法。在第四部分到第六部分, 我们阐述了如何为三个不同的机器人问题完成 IP 模型并且顺便引入了许多最佳实践。我们在第七部分进行总结。在附录中, 我们给出了对于整数规划实现的介绍。

2. 预备知识

A. 基于路径的优化问题

对于基于路径的优化问题, 我们采用了路径的标准图论编码。考虑一个有连接的无向图

$G(V, E)$, 顶点集合 V 和边集合 E 。对于 $v_i \in V$, 给定 v_i 的邻域为 $N(v_i) = \{v_j | (v_i, v_j) \in E\}$ 。对于一个机器人, 初始和目标顶点为 $x^I, x^G \in V$, 一条路径被定义为一列的顶点: $P = (p^0, \dots, p^T)$ 满足: (i) $p^0 = x^I$; (ii) $p^T = x^G$; (iii) $\forall 1 \leq t \leq T, p^{t-1} = p^t$ 或者 $(p^{t-1}, p^t) \in E$ 。对于 n 个具有初始和目标配置 $X^I = \{x_1^I, \dots, x_n^I\}$ 和 $X^G = \{x_1^G, \dots, x_n^G\}$ 的机器人, 路径为 $P = \{P_1, \dots, P_n\}$, 其中 $P_i = (p_i^0, \dots, p_i^T)$ 。这些路径不一定是无碰撞的。我们现在概述三种不同种类的基于路径的优化问题。

1) **多机器人路径规划 (MPP)**：其主要任务在于规划机器人到达其目标, 同时避免机器人之间的碰撞, 碰撞发生在两个机器人在某个顶点或某条边相遇。注意图论公式已经考虑了静态的障碍物。对于无碰撞的路径 $P, \forall 1 \leq t \leq T, P_i, P_j \in P$ 必须满足: (i) $p_i^t \neq p_j^t$; (ii) $(p_i^{t-1}, p_i^t) \neq (p_j^t, p_j^{t-1})$ 。MPP 的目标是最小化完成时间 T , T 为所有机器人达到目标顶点的时间。

在这篇文章中, 我们也引入了一个允许部分分解的实用 MPP 泛化问题。也就是说, 只有 $k \leq n$ 个被要求达到他们的提前明确目标的机器人。这个公式模拟了奖励递减的情况, 当达到一定数量的目标后, 报酬停止累积。在这里, 对于其他的 $(n-k)$ 个路径的最后的顶点能够在 V 中随机取得。注意路径 P 必须是无碰撞的。更通用地说, 这个问题也是更难被解决的。我们注意到这种设置与非区分机器人的情况不同, 后者要简单得多, 并且允许快速多项式时间算法。

问题 1 (泛化的时间优化 MPP)。给定 $\langle G, X^I, X^G, k \rangle$ 。寻找一个无碰撞的路径集合 P 。引导至少 k 个机器人到他们的目标并且最小化 T 。

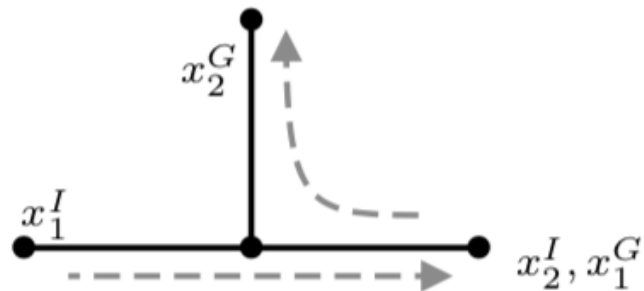


图 1: 一个 MPP 的例子, 其中 V 和 E 被涂成黑色。当 $k=n=2$ 时, 一个 3 步的无碰撞最小化完成时间解决方案需要 1 号机器人在初始位置保持不动。2 号机器人移动到中间顶点。当 $k=1$ 时, 一个 2 步的优化解决方案将 2 号机器人移动到它的目标。

2) 多机器人最小化约束移除 (MMCR) :

给定一个图 (V, E) 。给定一个障碍物 $O \subset V$ 是一个 V 中的子图。给定一个有限的障碍物集合 $\mathcal{O} = \{O_1, \dots\}$ ，多机器人最小化约束移除问题寻找一个解决方案 P 和一个需要被移动的障碍物集合 $O_r \subset O$ ，这样对于 $\mathcal{O} \setminus \mathcal{O}_r$ ， P 中的路径就不会穿越任何障碍。目标是 minimized 需要被移除的障碍物数量 $(|\mathcal{O}_r|)$ 。用更正式的语言讲：

问题 2 (MMCR)。给定 $\langle G, X^I, X^G, k, \mathcal{O} \rangle$ ，找出 P 和 \mathcal{O}_r 使得 $|\mathcal{O}_r|$ 最小，并且对于所有 $P_r \in P, O \in \mathcal{O} \setminus \mathcal{O}_r, 0 \leq t \leq T: P_t^i \notin O$

图 2 是 MMCR 的一个描述。

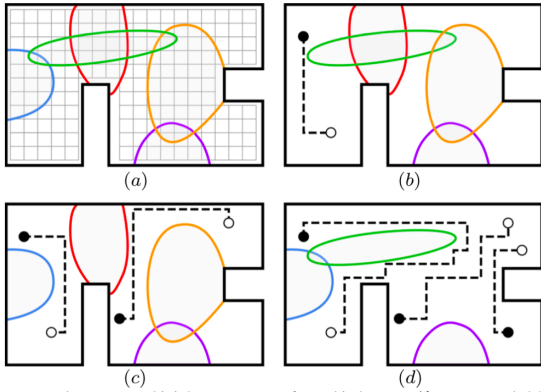


图 2: 一个 MMCR 的例子。(a) 灰色的方形网格 G 和 5 个被涂成红色蓝色绿色橙色和紫色的障碍物。为了更好的可视性， G 没有被画在其他的子图中。(b) 一个当 $n=1$ 时移动蓝色障碍物的最优解。点划线展示了一个对于机器人从初始位置 (圈) 到目标位置 (黑点) 的移动的可行的路径。一个可替代的优化算法可以用来移动绿色障碍物。(c) 对于 $n=2$ 时唯一的最优解法。(d) 当 $n=3$ 时的一个最优解法；可替代的部分解也是可行的。

3) 奖励收集问题 (RCP) :

指明 $\mathbb{R}_{\geq 0}$ 作为非负实数的集合。在一个奖励收集问题 (RCP) 中，一个机器人被安排在一个有限时间 $c^* \in \mathbb{R}_{\geq 0}$ 内在图 G 中运动，最大化其收集的奖励。这样一个设定定义了两个函数：一个成本函数 $C: P \rightarrow \mathbb{R}_{\geq 0}$ 表示已经花费的时间，和一个奖励函数 $R: P \rightarrow \mathbb{R}_{\geq 0}$ 追踪沿着 P 收集的奖励。显然，在这里时间可以被其他种类的有限资源所替代，比如燃料。

问题 3 (RCP)。给定 $\langle G, X^I, X^G, C, R, c^* \rangle$ ，找出路径 P 使得 $R(P)$ 在限制 $C(P) \leq c^*$ 下最大化 $R(P)$ 。

我们在本文中研究 RCP 的两个变体：二次相关定向问题 (QCOP) 和最优化旅行者问题 (OTP)。

在二次相关定向问题 (QCOP) 中，每个顶点 $v_i \in V$ 都关联一个奖励 $r_i \in \mathbb{R}_{\geq 0}$ 。如果 $v_i \in P$ ，不仅是 r_i 并且部分的从 $N(v_i)$ 中顶点的部分

奖励会被收集，如果 P 不包含这些顶点 (路径 P 不经过这些顶点)。将相关权重描述为 $\{w_{ij} = 1/|N(v_i)| \mid v_j \in N(v_i)\}$ ，可能从 v_i 的最大收集奖励是 $r_i + \sum_{v_j \in N(v_i)} w_{ij} r_j$ 。指定 $x_i \in \{0, 1\}$ 指明是否 $v_i \in P$ ，总共收集的奖励是

$$R_{QCOP}(P) = \sum_{i=1}^{|V|} \left(r_i x_i + \sum_{v_j \in N(v_i)} r_j w_{ij} x_i (x_j - x_i) \right) \quad (1)$$

在 QCOP 中，每条边 $(v_i, v_j) \in E$ 被关联到一个时间成本 $c_{ij} \in \mathbb{R}_{\geq 0}$ ，QCOP 的时间限制要求：

$$C_{QCOP}(P) = \sum_{i=1}^T c_{p^{t-1}, p^t} \leq c^* \quad (2)$$

在一个最优旅行者问题 (OTP) 中，在顶点 $v_i \in V$ 收集的奖励被描述为一个非递减的函数 $R_i(t_i)$ ， t_i 是机器人在顶点 v_i 处花费的时间长度。总的奖励被描述为：

$$R_{OTP}(P) = \sum_{v_i \in P} R_i(t_i) \quad (3)$$

和 QCOP 问题相似，OTP 的时间限制需要：

$$C_{OTP}(P) = \sum_{i=1}^T c_{p^{t-1}, p^t} + \sum_{v_i \in P} t_i \leq c^* \quad (4)$$

最优化 MPP、QCOP 和 OTP 已经被表明都是 NP 困难的 [2] [36] [37]。MMCR 是一个单机器人 MCR 问题的多机器人泛化，被认为是 NP 困难的。结果是 MMCR 也是在计算上难以实现的。

B. 整数规划基础

整数规划 (IP)，粗略地说，解决一类优化受整数约束的目标函数的问题。一个典型的 IP 子类是整数线性规划 (ILP)。给定一个向量 \mathbf{x} ，由整数变量组成，一个通用的 ILP 模型被描述为：

$$\text{minimize } \mathbf{c}^T \mathbf{x} \text{ subject to } \mathbf{A} \mathbf{x} \leq \mathbf{b}$$

在这里， \mathbf{c} ， \mathbf{b} 是向量，以及 \mathbf{A} 是一个矩阵。总体来说，对 \mathbf{x} 的解必须只包含整数。注意到，这个公式与等式约束兼容，因为等式约束可以解释为两个不等式约束。就如就要被阐述的，由于 IP 模型相当直接的公式化，从路径规划问题到 IP 的缩减通常不困难，在低多项式时间内实现，通过使用我们的方法。除了单纯的约简外，IP 也是一个被广泛研究的著名的基本数学问题。因此，一个 IP 模型能够经常通过使用像 Gurobi 求解器 [42]、Cplex 求解器 [43] 和 GLPK 求解器 [44] 被快速最优地解决。

注意：作为范围的一个说明，我们的工作从算法的角度研究基于 IP 和基于路径的优化问题。我们不处理执行方面的问题，如工作空间离散化、不确定性和通信问题。这些项目最终将由系统中的其他部分管理。例如，使用适当的同步和基于反馈的控制 [5]，由基于 ILP 的 MPP 求解器能够在多机器人硬件平台上轻松执行 [45]。

3. 通用方法和路径编码

我们对于基于路径优化问题的方法主要依据一个直接的 2 步的步骤。第一步，一个基本的 IP 模型被建立来确保只有可行的路径被生成。这是路径编码的步骤。比如，在一个多机器人路径规划问题的方面上，IP 模型比如生成多条路径以连接连接所需的初始和目标配置。第二步，优化条件和附加约束，比如，为了防止碰撞，这是强制的。在我们的经验中，路径编码步骤有一个有限的变化，而第二步往往有更多的变化，需要一些创造力。

在这一部分中，我们提供对于两个 IP 模型的详细描述为了进行路径编码：一个基于图的编码，与集合 E 中的初始边共同作用，以及一个时间拓展图编码，对由顶点集 V 的多个时间戳副本生成的新图上的路径进行编码。给定模型中整数变量的可行赋值与原问题中的所有可能路径对应(以一对一的方式)。优化步骤将在下一节中介绍。

A. 基于图的编码

我们定义一个通过至多一个任意的顶点的非循环路径作为路径 P 。也就是说： $\forall 1 \leq i < j \leq T, p^i \neq p^j$ 。给定 $\langle G, X^I, X^G \rangle$ ，基于图的编码引入一个二进制变量 x_{v_i, v_j} ，对于每条边 $(v_i, v_j) \in E$ ，指明是否 P 使用边 (v_i, v_j) 。下列的约束必须被满足：

$$\sum_{v_i \in N(x^I)} x_{x^I, v_i} = \sum_{v_i \in N(x^G)} x_{v_i, x^G} = 1 \quad (5)$$

$$\sum_{v_i \in N(x^I)} x_{v_i, x^I} = \sum_{v_i \in N(x^G)} x_{x^G, v_i} = 0 \quad (6)$$

$$\sum_{v_j \in N(v_i)} x_{v_i, v_j} = \sum_{v_j \in N(v_i)} x_{v_j, v_i} \leq 1, \forall v_i \in V \setminus \{x^I, x^G\} \quad (7)$$

在这里，约束(5)和(6)使 P 从 x^I 开始并且终止在 x^G 。约束(7)确保对于每个顶点，只有当且仅当一个入边被使用的情况下，一个出边才会

被使用。这避免了一条路径生成多个分支，并且保证每个顶点最多在 P 中出现一次。

从这个公式中得出的解可能会存在子循环，子循环是由和路径 P 不连接的边形成的循环（见图 3）。在 [36] 中介绍了能够通过创造整数变量 $3 \leq u_i \leq |V|$ 对每一个顶点 $v_i \in V \setminus \{x^I, x^G\}$ ，去除这些子循环，并且对每一对顶点 $v_i, v_j \in V \setminus \{x^I, x^G\}$ 增加下列的约束：

$$u_i - u_j + 1 \leq (|V| - 3)(1 - x_{v_i, v_j}). \quad (8)$$

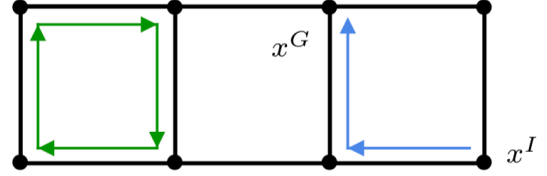


图3 在一个基于图编码公式中的对于一个子循环的表示。蓝色的线表示一个可行的路径，从 x^I 到 x^G 。绿色的循环是一个可能的非期望非连接子循环。

推论 3.1 给定 $\langle G, X^I, X^G \rangle$ ，在基于图的编码 IP 模型和所有的在图 G 中从 X^I 到 X^G 的非循环路径的解中存在一个双射关系。

证明 (\Rightarrow) 给定一个基于图的编码 IP 模型的一个可行解，一个在图 G 中从 X^I 到 X^G 的非循环路径通过跟踪正边变量直到到达 X^G 来构造得到。这个路径确保可行由于(i)通过约束(5)和(6)，路径只能从 X^I 开始终止于 X^G ，(ii)通过约束(7)和(8)，路径由 V 中的非重复顶点组成，由 E 中的边连接。

(\Leftarrow) 一个非循环的路径 $P = (p^0, \dots, p^T)$ 能够被转换为 IP 模型中的可行解，通过给相关的边变量赋值为 1 而所有其他的边变量赋值为 0。这些值满足约束(5)(6)(7)(8)因为 P 是一个从 X^I 开始到 X^G 结束的顶点序列，由边连接并且没有子循环。

当 $X^I = X^G$ ，我们将 $X^I(X^G)$ 分解为两个顶点 v_{in} 和 v_{out} 并且将他们和 $N(x^I)$ 中的所有顶点连接。一条路径能够在图中通过顶点集 $(V \setminus \{x^I\}) \cup \{v_{in}, v_{out}\}$ 被构成。结果得到的路径在新的图中是非循环的，但是在 G 中被解释为循环的路径，并且包含 $X^I(X^G)$ 。

B. 时间拓展图编码

这个编码技术使用时间扩展图而不是使用原来的图 $G(V, E)$ 。给定一个固定的时间域 $T \in \mathbb{N}$ 。我们复制 $T+1$ 个 V 的副本，取名为 V^0, \dots, V^T 。相邻的顶点在相邻时间步长被有向边连接：对于 $1 \leq t \leq T$ ，在 V^{t-1} 和 V^t 之间的

边集合是 $\{(v_i^{t-1}, v_j^t) | \forall v_i \in V, v_j \in N(v_i) \cup \{v_i\}\}$ 。然后我们增加一个有向的反馈边连接 $x^{G,T} \in V^T$ 至 $x^{I,0} \in V^0$ 。图 4 中给出了一个例子。和基于图的编码相似，我们使用一个二进制变量 x_{v_i, v_j}^t 来指示是否 (v_i^{t-1}, v_j^t) 被使用。通过将时间拓展图中 v_i 的出边（入边）表示为 $N^-(v_i)(N^+(v_i))$ ， $\forall 1 \leq t \leq T$ ，约束(5)(6)(7)被表示为：

$$\sum_{v_i \in N^+(x^G)} x_{v_i, x^G}^T = 1 \quad (9)$$

$$\sum_{v_j \in N^-(v_i)} x_{v_i, v_j}^t = \sum_{v_j \in N^+(v_i)} x_{v_j, v_i}^{t-1}, \forall v_i \in V^1 \cup \dots \cup V^T \quad (10)$$

在这个时间拓展图中的可行路径的表示是一系列的有向边，这些有向边从 $x^{I,0}$ 出发，分别经过 V^0, \dots, V^T 中的一个顶点，并且最终回到 $x^{I,0}$ 通过使用反馈边。相似地，一个 1-1 的对应解在 [5] 中存在。

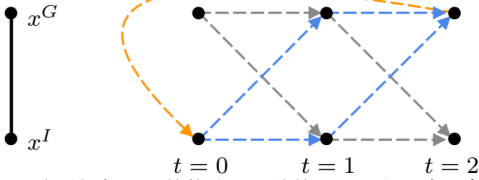


图 4 一个时间拓展图的构造。原始的图显示在左边，有两个被无向的边相连接的顶点。 $T=2$ 时的时间拓展图显示在右边。有向的边（蓝色的和灰色的虚线）被添加到 V 的三个副本中的相邻顶点。反馈边用橙色的虚线显示。灰色的边通过可达性检测被去除

推论 3.2 在可 G 中的长度小于 $T+1$ 的可行性路径和 T 步的时间拓展图中的可行性路径之间存在一个双射关系。

在一个时间拓展图编码中，生成路径的长度被时间域 T 限制。路径可以多次包含任何顶点。通过执行可达性测试，可以减少时间扩展图编码中的变量数量（见图 4），去除了从 x^I 到 x^G 不可达的边。可达性测试不影响完整性和最优性。可伸缩性还可以进一步改进通过使用 k -way split [5]，一种将问题分解成更小的子问题的“分而治之”的启发式方法。与解决单个大问题相比，解决子问题所需的时间要短得多。

C. 两种编码方式的基本拓展

两种编码都能够处理当一个机器人有多个可以选择的起点和终点的情况。在基于图的编码中，这个问题能够被解决，通过增加一个虚拟顶点，并且将其连接到所有可能的起点和终

点。在时间拓展图编码形式中，相同的目标能够被达到通过增加多个反馈边。

两种编码方式也能够被拓展到多机器人情境中。通过简单地为每个机器人的所有变量制作一个副本。然后可以在不考虑相互碰撞的情况下规划路径。正如我们之后将展示的，通过增加约束，时间拓展图编码能够生成无碰撞的路径。

在这里我们注意到我们的 IP 模型通过使用 Gurobi 求解器来解决 [42]。所有的实验在频率为 32GB 的 2133MHz，RAM 大小为 32GB 的 Intel[®] Core[™] i7-6900K CPU 上运行。

4. 多机器人路径规划的时间优化

一个高性能的时间优化 MPP IP 模型在 [5] 中被提出，通过使用时间拓展图编码。我们回顾了这个 baseline 模型，并介绍了两个用于调整状态空间的一般新方法。然后，我们介绍了一个更新的部分解的 MPP IP 模型。

基本方法 [5] 首先计算一个被低估的 T 通过引导每个机器人到它的目标，没有考虑机器人-机器人碰撞。这个目标被达成通过运行 n 个 A* 搜索并且取最大的路径长度。然后一个 IP 模型被构造并且可行性被检查：对于每个机器人 $1 \leq r \leq n$ ，一个满足约束(9)和(10)的变量集合 $\{x_{v_i, v_j}^t\}$ 被构造并且被重命名为 $\{x_{r, v_i, v_j}^t\}$ 。这个方法尝试在以下附加约束条件中寻找一个可行的变量值赋值：对于任何的 $1 \leq t \leq T$ ， $v_i \in V^T$ ，

$$\sum_{r=1}^n \sum_{v_j \in N^+(v_i)} x_{r, v_i, v_j}^t \leq 1, \quad (11)$$

$$\sum_{r=1}^n x_{r, v_i, v_j}^t + \sum_{r=1}^n x_{r, v_j, v_i}^t \leq 1, \forall v_j \in N^+(v_i) \quad (12)$$

在这里，约束(11)防止机器人同时占据同一个顶点，而约束(12)避免了 E 中边上的头对头的碰撞。通过推论 3.2，一个不可行的模型表明在最大运行时间 T 中不存在可行的解。然后， T 以 1 的步长递增，模型被重建并且被再次求解。一旦模型有一个可行解，一个有最优最大完成时间的 MPP 解被抽取。

新的有用的启发式算法。直观地，解决一个 IP 模型的时间经常和模型中的变量数量成反比。这说明计算时间可能被减少如果我们移除

V^0, \dots, V^T 中不太可能成为解决方案路径中一部分的顶点。回忆当计算被低估的最大运行时间时，对于每个机器人 r ，一个从 x_r^I 到 x_r^G 的最短路径 P_r^* 被获取。当图没有被密集占据时，我们可能通过对这些初始的候选路径做一些小的修改来找到一个解。基于这个分析，我们提出了两个新的启发式算法，基于可达性分析，来减少 IP 模型中的变量数。

- 1) **管道式邻域**：选取固定某个参数 $h_t \in \mathbb{N}$ ，对于机器人 r ，时间拓展图包括 v_i 当且仅当 v_i 和 P_r^* 某些顶点的距离在 h_t 以内。这个情况下的可达区域模仿了一个在候选路径周围的管道。启发式背后的基本原理是，一般来说，机器人的实际路径不太可能明显偏离参考路径 P_r^* 。
- 2) **可达圆**：选取固定某个参数 $h_s \in \mathbb{N}$ ，对于机器人 r ，时间拓展图包括 v_i 当且仅当 v_i 和 P_r^* 中第 $\lfloor |P_r^*|/T \rfloor$ 个元素的距离在 h_s 以内。在可达圆启发式算法背后的基本原理和管道式邻域启发式算法相似。

管道式邻域和可达圆启发式算法的影响在图 5 的左边和右边能被可视化发现。在这个例子中，我们设置可视化变量当 $t=0.4T$ 并且对于机器人 r 估计 $|P_r^*|=0.7T$ 。虚直线表示 P_r^* ，左边端点 x_r^I 和右边端点 x_r^G 。为了简明，所有其他的顶点没有展示出来。如果没有应用启发式算法， V^T 中包含整个地图中的所有顶点。通过可达性测试，一个顶点会从 V^T 被移除如果它在时间 t 内从 x^I 不可达，或者在时间 $T-t$ 内从 x^G 不可达。可达性测试移除所有不在以 x^I 和 x^G 为中心的两条弧线交点处的顶点，使 V^T 只包含在有红色边界的区域的顶点。在左边子图中的 $h_t=0.12T$ 的管道式邻域启发式算法，以及在右边子图中的 $h_s=0.18T$ 的可达圆启发式算法移除具有蓝色边框的形状中的所有顶点。被复制到 V^T 中的顶点被涂成了橙色。

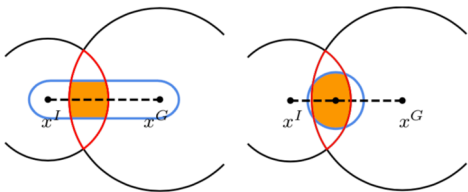


图 5 左子图为管道式邻域启发式算法的展示，右子图为可达圆启发式算法的展示。

我们观测启发式算法如何影响在 24×18 的网格上使用随机生成的测试用例的性能。如图 6 所示，变量的数量被缩减了超过 70%，当 $h_t \leq 2$ 或者 $h_s \leq 2$ 时；计算时间被缩减了

60%，当 $n \leq 60$ 时。在最好的情况下（红色曲线），计算时间的缩减超过了十倍，这是非常大的缩减。出乎我们意料的是，对于超过 60 个机器人，在 IP 模型中被缩减的变量数量不总是转化为更快的解决时间。在深入研究之后，这似乎和 Gurobi 的工作方法有关；我们能够确定我们的启发式算法干扰了 Gurobi 自己的启发式算法，当出现机器人太多的情况时。我们认为，对于所有的测试用例，实现的最优性不受新启发式算法的影响。

部分解。为了适应在问题 1 中 $k \leq n$ 个约束的情况，我们优先使用独立的路径长度来决定时间拓展参数 T 。也就是说，我们从 n 条路径中选出 k 条最短路径的最大长度 T 。对于剩下的 $(n-k)$ 个机器人，我们允许他们的在 IP 模型中的目标与他们各自指定的目标相邻。这通过在启发式算法中为这些 $(n-k)$ 个机器人增加额外的反馈边（见图 7）来实现。注意到现在的可达性测试不应该应用于这些 $(n-k)$ 个机器人。

在观测通用的 MPP 求解器时，我们使用一个 24×18 的网格，网格中有 10% 的顶点被移除来模拟为障碍物（见图 8 作为一个例子）。如图 9 所示，当我们只需要一个部分解时运行时间会被大大减少。

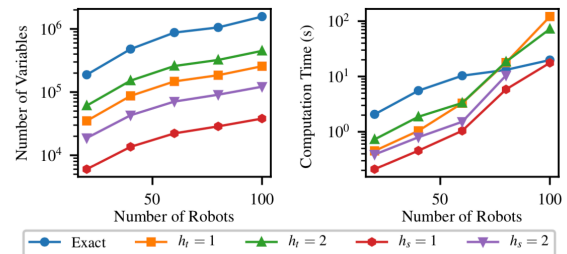


图 6 提出的 MPP 新启发式算法测试的结果。（左边）IP 模型中的顶点数与不同的 n 比较，（右边）计算时间的 IP 方法与不同的 n 的比较，具有不同的启发式参数。Exact 的条目表示原始的 IP 方法。

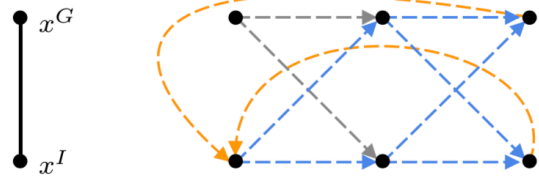


图 7 多反馈边对于和图 4 中相同的例子，使用相同的颜色和 DASH 样式进行可视化。

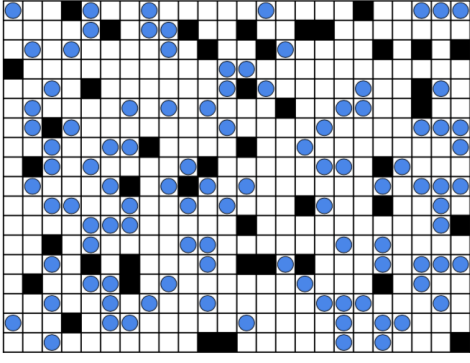


图8 一个 24×18 的网格，10%的顶点被移除（黑色的方块）以及 100 个随机分布的机器人（蓝色的圆圈）

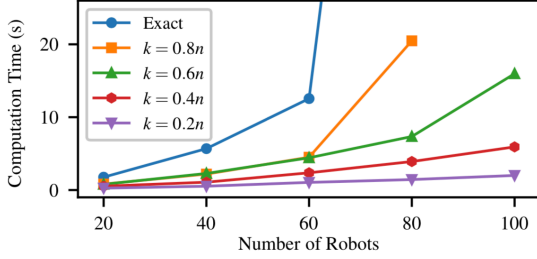


图9 MPP 部分求解器的计算时间于不同的 n 进行比较，与不同的 k 值。Exact 的曲线表示原始的 IP 方法。

支持非完全标记的问题。我们也可以采用现有的 IP 模型来解决一组机器人无法区分的问题，也就是说，他们被分配了一组没有具体个体目标的目标。解决方案中这些机器人的最终配置可以是目标的任意排列。这也被称为 k -colored 运动规划问题 [46]。为了更新这个模型，我们提出了一个有 m 个共享 m 个目标顶点的机器人的集合，我们为这个群体创建了一个 $\{x_{v_i, v_j}^t\}$ 的单个副本，然后在所有的起点终点对之间增加 m^2 个反馈边。与这些边相对应的变量的和被设置为 m 。

5. 多机器人最小约束移除 (MCR)

如在 [1] 中陈述，MCR 的状态空间能够通过建造一个建立一个反映不同组合障碍覆盖范围的图表。这里我们明确构造了图 G_{MCR}

(V_{MCR}, E_{MCR}) ，因此 V_{MCR} 中的每个部分是一个存在于同一个障碍物集合的互相连接的顶点集合。特别地， $V_i \in V_{MCR}$ 满足下列的约束：(i) $V_i \subset V_{MCR}$; (ii) $\forall v_j, v_k \in V_i, O \in \mathcal{O}, v_j \in O$ 当且仅当如果 $v_k \in O$; (iii) $\forall v_j, v_k \in V_i$ ，存在一个从 v_j 到 v_k 的路径 P ，由 V_i 中的元素组成。在这个定义之下， $\cup_{v_i \in V_{MCR}} V_i = V$ ，并且 $\forall v_i, v_j \in V_{MCR}, V_i \cap V_j = \emptyset$ 。在我们的设定中， V_{MCR} 的构造方法是：通过迭代地选择一个随机的 $v \in V$ 并且从 v 开始运行 breadth first search 搜索直到所有的叶节点与所有从 v 开始的不同的障碍物集合接

触。一个无向的边 (V_i, V_j) 被添加入边集合 E_{MCR} 若果存在 $v_i \in V_i$ 并且 $v_j \in V_j$ ，以此 $(v_i, v_j) \in E$ 。

图 10 展示了一个 G_{MCR} 的例子。

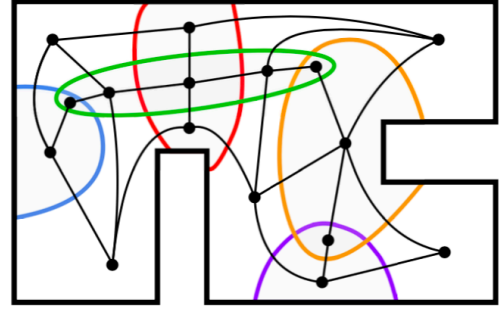


图10. 图2中实例问题的 G_{MCR} (使用黑色的点和线)。注意到每个顶点能够被看作一个在更详细的路径图 (比如 G) 中的顶点集合的高级抽象。

对于每个机器人 $1 \leq r \leq n$ ，我们将 $V_r^I \in V_{MCR}$ 描述为约束集合 x_r^I ，并且将 $V_r^G \in V_{MCR}$ 描述为约束集合 x_r^G 。我们假设 $V_r^I \neq V_r^G$ ，因为否则对于机器人 r 需要被移除的约束集合会成为无意义的 $\{O \in \mathcal{O} \mid O \cap V_r^I \neq \emptyset\}$ 。由于一些轻微的符号复用，我们将 $N(V_i)$ 描述为 V_i 的邻域。根据基于图的编码方式，对于每一个机器人 $1 \leq r \leq n$ ，一个满足约束(5)(6)(7)的变量集合 $\{x_{v_i, v_j}\}$ 被构造并且被重命名为 $\{x_{r, v_i, v_j}\}$ 。

在这个部分的剩下内容中，除非被特别提出，一条路径即指的是图 G_{MCR} 中的一条路径。我们引入二进制变量 $\{x_{v_i} \mid V_i \in V_{MCR}\}$ 来指示如果至少在 n 条路径中有一条含有 V_i ，同时，引入 $\{x_{o_i} \mid O_i \in \mathcal{O}\}$ 来指示是否 O_i 比如被移除来形成不和其他障碍物接触的路径。对于这些变量，我们有约束：

$$Lx_{v_i} \geq \sum_{r=1}^n \sum_{v_j \in N(v_i)} (x_{r, v_i, v_j} + x_{r, v_j, v_i}); \quad (13)$$

$$Lx_{o_i} \geq \sum_{V_i \in V_{MCR}, V_i \subset O_i} x_{v_i}; \quad (14)$$

在这里， L 是一个大的常数。约束(13)表明，只有在 V_i 在任意的路径中，约束(13)才会向 x_{v_i} 赋正值。约束(14)确保 $\forall V_i \in V_{MCR}$ ，如果 $x_{v_i}=1$ ，则 $\{x_{o_j} \mid O_j \in \mathcal{O}, V_i \subset O_j\}$ 全部被赋为正值。这个模型的目标是最小化 $\sum_{O_i \in \mathcal{O}} x_{o_i}$ (被移动障碍物的数量)。注意到子循环去除对于 MMCR 是不必要的因为路径中的子循环和重复元素都不会降低目标值。机器人与机器人之间的碰撞没有被考虑 (但是能够被考虑如果需要的话)。在求解了这个 IP 模型之后，被移除的

障碍物可以从这个模型中被分解出来，在这之后一个实际的机器人路径能够被简单地发现。

我们将这个 IP 方法和两个在 [1] 中的基于搜索的求解器进行比较。对于 MCR，一个在这些求解器中的搜索节点是一个 $\langle v, \mathcal{Q}_r \rangle$ 元组， v 是机器人的当前位置， \mathcal{Q}_r 是所有遇到的障碍的集合，从 v 追溯到 x^I 。然后采用不同的剪枝方法来减少搜索空间，从而产生精确算法以及贪婪算法求解器。通过维护处于搜索状态的所有机器人的位置，精确算法求解器和贪婪算法求解器都可以扩展到 MMCR。

比较结果在图 11 中被给出。左图是对于一个 100×100 的网格图中运行的 MCR，右图是对于一个 10×10 的网格图中运行的 $n=2$ 的 MMCR。所有的测试样例都是通过在网格中随机排布任意尺寸的长方体障碍物而生成的。

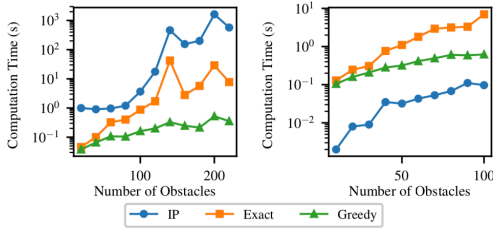


图 11. 不同的 MCR 和 MMCR 方法的计算时间，对应于障碍物的数量。左子图显示了 MCR 的结果，右图显示了 MMCR 的结果。Exact 和 Greedy 曲线分别表示精确算法和贪婪算法求解器。

我们观察到对于 MCR 精确 IP 方法比基于搜索精确离散方法运行速度慢，但是对于 MMCR，精确 IP 方法比两个基于搜索的方法都要表现优异。我们认为，这是因为在基于网格的环境中，单个机器人的问题还不够“复杂”，无法利用基于 IP 模型方法的结构优势。进一步的评估表明，我们的 IP 方法可以在 5 秒内解决 100 个机器人和 100 个障碍物的 50×50 网格上的问题，而基于搜索的求解器在这种情况下无法完成。

从图 11(左)可以看出，即使数据已经平均超过 20 个实例，计算时间也不会随着障碍物数量的增加而单调增加。这是由于 MCR 的 NP-困难，可能会有一些随机的实例是特别难解决的。但是，请注意方法之间的相对性能差异是一致的。

6. 奖励收集问题 (RCPs)

在这个部分中，我们观测一些复杂的奖励收集问题 (RCPs) 为了更深入地阐述我们的 IP 方法的灵活性。QCOP [36] 和 OTP [37] 的初始工作使用了基于图的编码。我们在这里证明，这

些问题也可以通过时间拓展图编码来解决，以获得更好的计算性能。

在 QCOP 中，起初的和目标的顶点能够从固定的集合 $X^I \subset V, X^G \subset V$ 中被自由选取。为了处理这个问题，我们创建了一个虚拟的顶点 u 并且从 u 开始到所有 V^0 的候选初始顶点增加有向的边，同样，增加从所有在 V^0, \dots, V^T 中可能的目标顶点到 u 的有向边。特别地，这些有向边被增加到时间拓展图中： $\{(u, v^0) | v \in X^I\} \cup \{(v^t, u) | v \in X^G, 0 \leq t \leq T\}$ 。对于一个固定的时间域 T ，约束(9)现在被表示为：

$$\sum_{v_i \in N^-(u)} x_{u,v_i}^0 = \sum_{t=0}^T \sum_{v_i \in N^+(u)} x_{v_i,u}^t = 1 \quad (15)$$

使用一个二进制变量 x_{v_i} 来指示是否顶点和边被使用，(1)的目标值（要被最大化）是直接被编码入 IP 模型中。时间消耗需求(2)作为一个约束被加入。

OTP 的时间拓展模型和 QCOP 相似，具有一个额外约束集 $x_{u,v_i}^0 = \sum_{t=0}^T x_{v_i,u}^t, \forall v_i \in X^I$ ，为了确保 $X^I = X^G$ ，并且构造一个额外的非整数的非负的变量集合 t_i 以指示机器人在 v_i 处的停留时间。约束 $t_i \leq L x_{v_i}$ 为了确保 t_i 是一个正值的情况仅出现在 v_i 在路径中， L 是一个大的常数。

我们使用类似于 [36] 和 [37] 的可变大小网格设置进行计算。我们使 X^I 包含两个随机选择的顶点。在 QCOP 中， $X^G = V$ ；一个随机的奖励权重 r 被赋值到每一个顶点。在 OTP 中，我们假设 $R_i(t_i)$ 是具有随机正系数的线性函数。我们从结果中可以观察到（图 12），随着问题规模的增大，时间拓展图编码总是具有竞争力的，并且表现得很好。

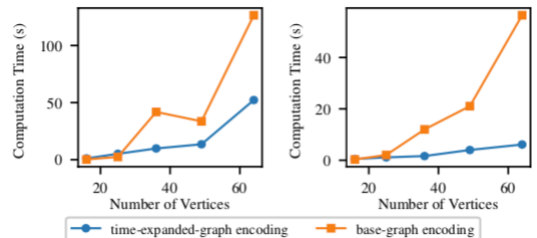


图 12. 两个路径编码方式对于不同的顶点数量 n 的计算时间：(左) QCOP 的结果，基图编码的波动主要是由于其对网格宽高比的敏感性；(右) OTP 的结果

7. 讨论和结论

在本文中，在先前研究的基础上，我们提出了一个两步整数规划(IP)方法来解决基于路

径的优化问题。该方法适用于机器人技术中涉及到过滤大量候选路径的各种计算难题。虽然使用简单, 利用高度优化的求解器的力量, IP方法带来的性能通常是有竞争力的, 甚至超过传统方法。我们指出两大优势的IP解决方案方法:(1)由于其易用性, 开发解决方案所需的时间是可以大大减少的,(2)这个方法可以提供参考的最优解, 帮助加快传统算法的设计。本文为基于路径的优化的IP模型构建提供了原理和许多最佳实践, 我们希望促进在机器人领域的从业者解决新问题时采用该方法作为首选方法。

引用文献

- [1] K. Hauser, "The minimum constraint removal problem with three robotics applications," in *The International Journal of Robotics Research*, vol. 33, no. 1, 2014, pp. 5-17.
- [2] J. Yu, "Intractability of optimal multi-robot path planning on planar graphs," in *IEEE Robotics and Automation Letters*, vol. 1, no. 1, 2016, pp. 33-40.
- [3] G. Dantzig, *Linear programming and extensions.*, Princeton university press, 2016.
- [4] G. L. Nemhauser and L. A. Wolsey, "Integer programming and combinatorial optimization," in *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, vol. 20, 1988, pp. 8-12.
- [5] J. Yu and S. M. LaValle, "Optimal multi-robot path planning on graphs: Complete algorithms and effective heuristics," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1163-1177, 2016.
- [6] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems" *Journal of the ACM (JACM)*, vol. 7, no. 4, 1960, pp. 326-329.
- [7] T. C. Hu, "Integer programming and network flows," WISCONSIN UNIV MADISON DEPT OF COMPUTER SCIENCES, Tech. Rep., 1969.
- [8] C. Morefield, "Application of 0-1 integer programming to multitarget tracking problems," *IEEE Transactions on Automatic Control*, vol. 22, no. 3, 1977, pp. 302-312.
- [9] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Control Conference (ECC), 2001 European*, IEEE, 2001, pp. 2603-2608.
- [10] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 295-310, 2005..
- [11] H. Ding, G. Reißig, D. Groß, and O. Stursberg, "Mixed-integer programming for optimal path planning of robotic manipulators," in *Automation Science and Engineering (CASE), 2011 IEEE Conference on*. IEEE, 2011, pp. 133-138.
- [12] S. D. Han, N. M. Stiffler, A. Krontiris, K. E. Bekris, and J. Yu, "Complexity results and fast methods for optimal tabletop rearrangement with overhand grasps," *The International Journal of Robotics Research*, p. 0278364918780999, 2017.
- [13] S. Rodriguez and N. M. Amato, "Behavior-based evacuation planning," in *Proceedings IEEE International Conference on Robotics & Automation*, 2010, pp. 350-355.
- [14] S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," in *Proceedings IEEE International Conference on Robotics & Automation*, 2004.
- [15] B. Smith, M. Egerstedt, and A. Howard, "Automatic generation of persistent formations for multi-agent networks under range constraints," *ACM/Springer Mobile Networks and Applications Journal*, vol. 14, no. 3, pp. 322-335, June 2009.
- [16] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325-344, June 2000.
- [17] E. J. Griffith and S. Akella, "Coordinating multiple droplets in planar array digital microfluidic systems," *International Journal of Robotics Research*, vol. 24, no. 11, pp. 933-949, 2005.
- [18] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots & Systems*, 1995, pp. 235-242.
- [19] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," in *Proceedings IEEE International Conference on Robotics & Automation*, 1997.
- [20] R. A. Knepper and D. Rus, "Pedestrian-inspired sampling-based multi-robot collision avoidance," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012, pp. 94-100.
- [21] P. Surynek, "Towards optimal cooperative path planning in hard setups through satisfiability solving," in *Proceedings 12th Pacific Rim International Conference on Artificial Intelligence*, 2012.
- [22] E. Erdem, D. G. Kisa, U. Oztok, and P. Schueller, "A general formal framework for pathfinding problems with multiple agents." in *AAAI*, 2013.
- [23] T. Standley and R. Korf, "Complete algorithms for cooperative pathfinding problems," in *Proceedings International Joint Conference on Artificial Intelligence*, 2011, pp. 668-673.
- [24] G. Wagner and H. Choset, "Subdimensional expansion for multirobot path planning," *Artificial Intelligence*, vol. 219, pp. 1-24, 2015.
- [25] E. Boyarski, A. Felner, R. Stern, G. Sharon, O. Betzalel, D. Tolpin, and E. Shimony, "Icbs: The improved conflict-based search algorithm for multi-agent pathfinding," in *Eighth Annual Symposium on Combinatorial Search*, 2015.
- [26] H. Ma, G. Wagner, A. Felner, J. Li, T. K. S. Kumar, and S. Koenig, "Multi-agent path finding with deadlines," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. International Joint Conferences on Artificial Intelligence Organization*, 7 2018, pp. 417-423. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/58>
- [27] T. Lozano-Pérez and L. P. Kaelbling, "A constraint-based method for solving sequential manipulation planning problems," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 3684-3691.
- [28] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "An incremental constraint-based framework for task and motion planning," *The International Journal of Robotics Research*, p. 0278364918761570, 2018.
- [29] A. Krontiris and K. E. Bekris, "Efficiently solving general rearrangement tasks: A fast extension primitive for an incremental sampling-based planner," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE,, 2016, p. 3924-3931.
- [30] L. I. R. Castro, P. Chaudhari, J. Tumova, S. Karaman, E. Frazzoli, and D. Rus, "Incremental sampling-based algorithm for minimum-violation motion planning," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 3217-3224.
- [31] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Ffrob: An efficient heuristic for task and motion planning," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 179-195.
- [32] A. Krontiris and K. Bekris, "Computational tradeoffs of search methods for minimum constraint removal paths," in *Eighth Annual Symposium on Combinatorial Search*, 2015.
- [33] E. Eiben, J. Gemmell, I. A. Kanj, and A. Youngdahl, "Improved results for minimum constraint removal." in *AAAI*, 2018.
- [34] E. Lawler, J. Lenstra, A. Rinnooy Kan, and D. Shmoys, "The traveling salesman problem. 1985," *John Wiley&Sons, Essex, England*, 1985.
- [35] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1-10, 2011.
- [36] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1106-1118, 2016.

- [37] J. Yu, J. Aslam, S. Karaman, and D. Rus, "Optimal tourist problem and anytime planning of trip itineraries," arXiv preprint arXiv:1409.8536, 2014.
- [38] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic uav and ugv system for precision agriculture," IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1498–1511, 2016.
- [39] K.-C. Ma, L. Liu, and G. S. Sukhatme, "An information-driven and disturbance-aware planning method for long-term ocean monitoring," in Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, 2016, pp. 2102–2108.
- [40] C. Papachristos, K. Alexis, L. R. G. Carrillo, and A. Tzes, "Distributed infrastructure inspection path planning for aerial robotics subject to time constraints," in Unmanned Aircraft Systems (ICUAS), 2016 International Conference on. IEEE, 2016, pp. 406–412.
- [41] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
- [42] I. Gurobi Optimization, Gurobi optimizer reference manual, 2014. [Online]. Available: <http://www.gurobi.com>.
- [43] Ibm ilog cplex optimization studio, Available: <https://www.ibm.com/analytics/cplex-optimizer>.
- [44] Gnu linear programming kit, [Online]. <https://www.gnu.org/software/glpk/>.
- [45] S. D. Han, E. J. Rodriguez, and J. Yu, "Sear: A polynomial-time multi-robot path planning algorithm with expected constant-factor optimality guarantee," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 1–9.
- [46] K. Solovey and D. Halperin, "k-color multi-robot motion planning," in Proceedings Workshop on Algorithmic Foundations of Robotics, 2012.

附录：

详细的 IP 模型构造实例

在这一部分中，我们将使用一些简单的问题示例提供 IP 公式的实现细节。为了更好地展示 IP 结构，我们

- 未充分应用可达性检验，且
- 没有删除多余的变量和约束。

A. 部分解的 MPP

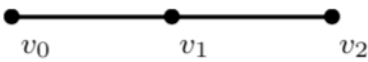


图 13. 一个简单 3 顶点 $\{v_0, v_1, v_2\}$ 的 MPP 实例。

考虑一个图 13 中的图 G ，问题中的参数 $n = 2$, $k = 1$, $x^I_1 = v_0$, $x^G_1 = v_1$, $x^I_2 = v_1$, $x^G_2 = v_2$ 。应用时间拓展图编码方法， $T = 1$ 时在我们的 IP 公式中的二进制变量为：

$$x^0_{1,v_0,v_0}, x^0_{1,v_1,v_0}, x^0_{2,v_0,v_1}, x^0_{2,v_1,v_1}, x^0_{2,v_2,v_1},$$

$$x^1_{1,v_0,v_0}, x^1_{1,v_0,v_1}, x^1_{2,v_1,v_0}, x^1_{2,v_1,v_1}, x^1_{2,v_1,v_2},$$

前五个变量描述了反馈边的使用，后五个变量描述了由 G 生成的时间拓展边的使用。

目标函数为：

$$\text{maximize } x^0_{1,v_1,v_0} + x^0_{2,v_1,v_0}$$

IP 公式的约束为：

约束(9),

$$x^0_{1,v_0,v_0} = x^1_{1,v_0,v_1} = 1, x^0_{2,v_2,v_1} = x^1_{2,v_1,v_2} = 1.$$

约束(10),

$$x^0_{1,v_0,v_0} + x^1_{1,v_0,v_1} = x^0_{1,v_0,v_0} + x^0_{1,v_1,v_0},$$

$$x^1_{2,v_1,v_2} + x^1_{2,v_1,v_1} + x^1_{2,v_1,v_0} = x^0_{2,v_2,v_1} + x^0_{2,v_1,v_1} + x^0_{2,v_0,v_1}.$$

约束(11),

$$x^0_{1,v_0,v_0} + x^0_{1,v_1,v_0} \leq 1,$$

$$x^0_{2,v_0,v_1} + x^0_{2,v_1,v_1} + x^0_{2,v_2,v_1} \leq 1,$$

$$x^1_{1,v_0,v_0} + x^1_{2,v_1,v_0} \leq 1,$$

$$x^1_{1,v_0,v_1} + x^1_{2,v_1,v_1} \leq 1,$$

$$x^1_{2,v_1,v_2} \leq 1.$$

约束(12):

$$x^0_{1,v_0,v_0} \leq 1,$$

$$x^0_{2,v_0,v_1} + x^0_{1,v_1,v_0} \leq 1,$$

$$x^0_{1,v_1,v_1} \leq 1,$$

$$x^0_{2,v_2,v_1} \leq 1,$$

$$x^1_{1,v_0,v_0} \leq 1,$$

$$x^1_{1,v_0,v_1} + x^1_{2,v_1,v_0} \leq 1,$$

$$x^1_{2,v_1,v_1} \leq 1,$$

$$x^1_{2,v_1,v_2} \leq 1,$$

注意到对于部分解，目标函数不需要被最优化；我们可以将 IP 求解的过程终止，当出现：

$$x^0_{1,v_1,v_0} + x^0_{2,v_1,v_0} \geq k = 1.$$

B. MPP 启发式算法

接着上一个例子，当我们使用管道式邻域算法， $h_t = 0$ ，变量 x^0_{2,v_0,v_1} 并且 x^1_{2,v_1,v_0} 将被置为 0，由于机器人 2 的最短路径不包含 v_0 。当可达圆 $h_s = 0$ ，另外四个变量 x^0_{1,v_0,v_0} ， x^1_{1,v_0,v_0} ， x^0_{2,v_1,v_1} ， x^1_{2,v_1,v_1} 将被置为 0。对于这个特定的问题实例，这些修改并不影响解决方案的可行性和最优性。

请注意，尽管良好的可达性测试将删除上述 6 个变量，但在较大的问题实例中，受每个启发式算法影响的变量通常不是彼此的子集。

C. MMCR

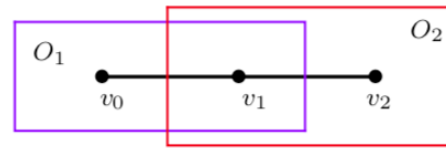


图 14. 一个简单的 3 顶点 $\{v_0, v_1, v_2\}$ 的 (M)MCR 实例以及两个障碍物 $O_1 = \{v_0, v_1\}$, $O_2 = \{v_1, v_2\}$ 。

正如图 13 所示，我们向上一个 MPP 的例子中增加两个障碍物来使之成为一个 MMCR 的例子。问题具有相同的参数 $n = 2$, $k = 1$, $x^I_1 = v_0$, $x^G_1 = v_1$, $x^I_2 = v_1$, $x^G_2 = v_2$ 。使用基于图的公式，二进制变量为：

$$x_{1,v_0,v_1}, x_{1,v_1,v_0}, x_{1,v_1,v_2}, x_{1,v_2,v_1},$$

$$x_{2,v_0,v_1}, x_{2,v_1,v_0}, x_{2,v_1,v_2}, x_{2,v_2,v_1},$$

$$x_{v_0}, x_{v_1}, x_{v_2}, x_{O_1}, x_{O_2}.$$

目标函数是最小化和路径相交的障碍物数量：

$$\text{minimize } x_{O_1} + x_{O_2}.$$

约束(5),

$$x_{1,v_0,v_1} = x_{1,v_0,v_1} + x_{1,v_2,v_1} = 1,$$

$$x_{2,v_1,v_0} + x_{2,v_1,v_2} = x_{2,v_1,v_2} = 1.$$

约束(6),

$$\begin{aligned} x_{1,v_1,v_0} &= x_{1,v_1,v_0} + x_{1,v_1,v_2} = 0, \\ x_{2,v_0,v_1} + x_{2,v_2,v_1} &= x_{2,v_2,v_1} = 0. \end{aligned}$$

约束(7),

$$\begin{aligned} x_{1,v_1,v_2} &= x_{1,v_1,v_2} \leq 1, \\ x_{2,v_0,v_1} &= x_{2,v_1,v_0} \leq 1. \end{aligned}$$

注意到约束(8)对于 MMCR 问题来说不必要因为额外的子循环不能减少目标函数值。

约束(13),

$$\begin{aligned} L x_{v_0} &\geq x_{1,v_0,v_1} + x_{1,v_1,v_0} + x_{2,v_0,v_1} + x_{2,v_1,v_0}, \\ L x_{v_1} &\geq x_{1,v_1,v_0} + x_{1,v_0,v_1} + x_{1,v_1,v_2} + x_{1,v_2,v_1} \\ &\quad + x_{2,v_1,v_0} + x_{2,v_0,v_1} + x_{2,v_2,v_1} + x_{2,v_1,v_2}, \\ L x_{v_2} &\geq x_{1,v_2,v_1} + x_{1,v_1,v_2} + x_{2,v_2,v_1} + x_{2,v_1,v_2}, \end{aligned}$$

约束(14),

$$\begin{aligned} L x_{O_1} &\geq x_{v_0} + x_{v_1}, \\ L x_{O_2} &\geq x_{v_1} + x_{v_2}, \end{aligned}$$

D.RCP

在这个部分中，我们只阐述 OTP 的实现示例，因为该公式可以粗略地看作是带有额外变量和约束的 QCOP。我们继续使用图 13 中的图。我们考虑三个顶点都和三个线性的奖励函数相关联 R_0, R_1, R_2 ；机器人从顶点 v_0 出发； (v_0, v_1) 和 (v_1, v_2) 边的成本为 c_{01} 和 c_{12} ；最大成本为 c^* ； $X^I = X^G = \{v_0, v_1\}$ 。使用时间拓展编码方式以及额外的虚拟顶点 u ， $T=2$ 时我们 IP 公式的二进制变量时：

$$x^0_{u,v_0}, x^0_{u,v_1}, x^2_{v_0,u}, x^2_{v_1,u}, x^1_{v_0,v_0}, x^1_{v_0,v_1}, x^1_{v_1,v_0}, x^1_{v_1,v_1}, x^1_{v_1,v_2}, x_{v_0}, x_{v_1}, x_{v_2}.$$

三个额外的连续变量是：

$$t_0 \geq 0, t_1 \geq 0, t_2 \geq 0.$$

目标函数是：

$$\text{maximize } R_0(t_0) + R_1(t_1) + R_2(t_2).$$

约束(4),

$$c_{01} * (x^1_{v_0,v_1} + x^1_{v_1,v_0}) + c_{12} * x^1_{v_1,v_2} + t_0 + t_1 + t_2 \leq c^*.$$

约束(15) (而不是(9))，

$$x^0_{u,v_0} + x^0_{u,v_1} = x^2_{v_0,u} + x^2_{v_1,u} = 1,$$

约束(10),

$$\begin{aligned} x^1_{v_0,v_0} + x^1_{v_0,v_1} &= x^0_{u,v_0}, \\ x^1_{v_1,v_0} + x^1_{v_1,v_1} + x^1_{v_1,v_2} &= x^0_{u,v_1}, \\ x^2_{v_0,u} &= x^1_{v_0,v_0} + x^1_{v_1,v_0}, \\ x^2_{v_1,u} &= x^1_{v_0,v_1} + x^1_{v_1,v_1}. \end{aligned}$$

约束(16),

$$\begin{aligned} x_{v_0} &\leq x^0_{u,v_0} + x^1_{v_0,v_0} + x^1_{v_1,v_0}, \\ x_{v_1} &\leq x^0_{u,v_1} + x^1_{v_0,v_1} + x^1_{v_1,v_1}, \\ x_{v_2} &\leq x^1_{v_1,v_2}. \end{aligned}$$

对于 OTP 有额外的约束：

$$\begin{aligned} x^0_{u,v_0} &= x^2_{v_0,u}, x^0_{u,v_1} = x^2_{v_1,u} \\ t_0 &< L x_{v_0}, t_1 < L x_{v_1}, t_2 < L x_{v_2}. \end{aligned}$$