用于基于连杆的机器人原型制作的连杆的演化

Emma Stensby
Norstein信息学系 奥斯陆大学奥斯陆，挪威

KaiOlav Ellefsen
奥斯陆大学信息学系奥斯陆，挪威

Frank Veenstra信息学系 奥斯陆大学奥斯陆，挪威

Tønnes Nygaard机械、电子与化学工程系 奥斯陆大都会大学挪威奥斯陆

Kyrre Glette*RITMO*，奥斯陆大学信息学系奥斯陆，挪威

arXiv:2305.01161v1 [cs.RO] 2 May 2023

摘要——**机器人系统的原型制作是一个耗时的过程。然而，计算机辅助设计可以显著加快这一过程。质量多样性进化方法在优化性能的同时追求新颖性，并可用于生成多样化设计的集合。这个设计库可作为引导设计人员并启动快速原型制作过程的工具。本文在基于机械连杆的机器人领域探讨了这一思路。这些机器人可作为快速原型制作的良好测试平台，因为它们能够快速改造，从而迅速迭代设计。我们比较了三种用于优化二维机械连杆的进化算法：1）标准进化算法，2）多目标算法 NSGA-II，以及 3）质量-多样性算法 MAP-Elites。随后，通过原型制作过程，将部分进化得到的连杆机构应用于实体六足机器人，并在两种不同的地面上进行测试。我们发现，除标准进化算法外，所有受测方法都能找到可生成与指定目标路径相似路径的机械连杆机构。然而，在原型制作过程中，将连杆长度作为行为描述符的质量-多样性方法最为实用。这是因为质量-多样性方法提供了更多相似设计供选择，而且可以通过行为描述符限制搜索范围，从而生成适合在我们的六足机器人平台上建造的连杆机构。**

索引词—**机器人、机械连杆机构、进化算法、质量-多样性**

一、引言

机器人在日常生活中变得越来越普遍，为个人和公司自动化执行各种任务。随着设计中的机器人多样性不断增加，能够辅助并加快设计流程的工具也变得越来越重要。手动设计机器人是一项困难的任务，需要具备专业知识，而且耗时较长 [1]。加快这一过程的一种方法是将部分设计工作交由计算机完成。工程师可以从计算机提供的多种设计方案中进行选择，然后根据机器人的需求调整设计。本文研究的是如何在基于连杆的机器人领域生成这些多样化的初始设计方案。

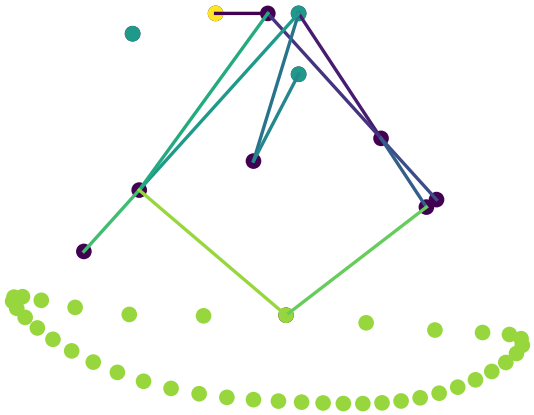


图 1. 二维连杆机构模拟器生成的连杆机构及其路径。黄色节点是电机，青绿色节点是静止的，紫色节点是可移动的。绿色点显示连杆机构的足端所沿的路径。杆件的颜色是任意的。



图 2. 用于测试进化后的机械连杆机构的六足机器人平台，以及其上进行测试的两种地面纹理。

我们认为，进化算法 [2]可以成为创建初始机器人设计的一种可行方法。进化算法在进化机器人学领域用于优化机器人形态和控制方面 [3], [4],[5]已有悠久的历史。在此背景下，一类尤其值得关注的进化算法是质量-多样性算法 [6]。这些算法在优化性能的同时，也优化多样性或新颖性，此外还

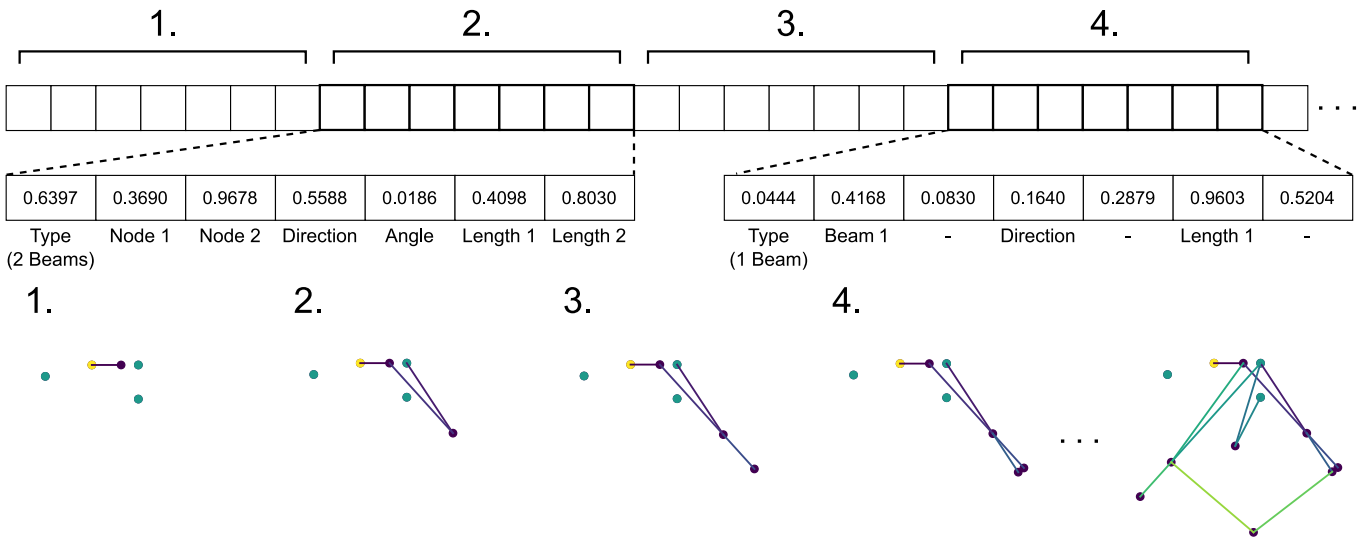


图 3. 将基因组解码为连杆机构。每七个数字组成一个区段，并解码为一个节点。第一个区段是一个特殊区段，包含相对于电机节点的三个静态节点的 x、y 坐标，以及曲柄的长度（由电机直接驱动的杆件）。第一个数字（类型）决定新节点是通过一根杆件（小于 0.25，如 4.）还是两根杆件（大于 0.25，如 2.）连接。如果节点通过一根杆件连接，则将新杆件视为现有杆件的延长部分。如果通过两根杆件连接，则新杆件可以自由转动。其余数字决定节点的属性。浮点数会经过转换，以适配相关属性的取值范围。与类型不同，其余属性中的所有取值被选中的概率相同。取值范围如下：节点： [0, 现有节点），杆件： [0, 现有杆件），方向： [0,1],角度：[0, 2*PI），长度： [可能的最短杆件长度，可能的最长杆件长度]。

经常保存已发现的解决方案库[7]。质量-多样性算法作为同时优化机器人形态和控制的方法 [5],[8], [9]，已受到广泛关注。质量-多样性算法 MAP-Elites（Multi-dimensional Archive of Phenotypic Elites） [10], 及其扩展算法（如 SAIL） [11], 此前已被用于进化模块化机器人 [12], [13], [14], ，并作为其他设计任务中原型制作的设计建议提供工具 [11], [15]。在机器人原型制作过程中，MAP-Elites 可以成为一种强大的工具。其促进多样性的解库能够确保提供各种各样的设计建议，并且可以按照易于浏览的方式进行组织。据我们所知，我们是首个将 MAP-Elites 作为设计建议提供工具，与实体机器人上的迭代原型制作过程相结合的研究。

为了能够在实体机器人上快速迭代设计，我们选择基于机械连杆的机器人作为平台。我们发现，机械连杆机构是合适的测试平台，因为通过拆开并以不同方式重新连接杆件，可以快速构造新的连杆机构，而机器人的底盘保持不变。机械连杆机构的综合研究由来已久，已有一百多年历史 [16]。许多人使用遗传算法或其他优化算法，构造能够沿特定路径运动的连杆机构，通常通过优化特定连杆机构构型中杆件的长度来实现 [17], [18], [19]。机械连杆机构已应用于许多工业领域，也用于机器人学中的运动研究 [20]。Theo Jansen 是最早使用遗传算法优化连杆式步行机器人的人之一 [21]。基于连杆的机器人只需使用少量电机，因此，如果需要比常规足式机器人更轻、更节能的机器人，它们可用于运动。不过，基于连杆机构的机器人比常规足式机器人受到更多约束。机器人建造完成后，腿部的运动轨迹通常无法改变，除非重新构建机器人，或者在连杆机构部件中预先设计了自适应修改特性 [22]。

我们测试了三种用于生成机器人机械连杆机构的进化方法。这三种进化算法分别为：1）标准进化算法[2]，2；2）多目标进化算法 NSGA-II（非支配排序遗传算法 II）[23],；以及 3）质量-多样性方法 MAP-Elites [10]。我们还测试了两种不同的适应度定义。通过比较所获得的最佳适应度以及对生成路径的定性分析，对这些方法进行评估。随后，我们通过原型制作，将部分连杆机构应用于六足机器人，并利用生成的连杆机构库辅助设计。在两种具有不同特性的地面上对机器人进行测试，并尝试评估如何利用这些连杆机构库，根据地面纹理调整连杆机构。

本研究的贡献有两方面：1）我们比较了多种用于设计机械连杆机构的进化方法，旨在找出为六足机器人进化机械腿机构最有效的方法。2）我们展示了如何将进化得到的机械连杆机构库应用于基于实体连杆机构的六足机器人平台的原型制作过程。

II. 方法

我们使用 Python 实现了一个简单的二维模拟器，用于模拟机械连杆机构，并用其评估

潜在的机器人设计[1]。我们比较了三种用于优化连杆机构的不同进化算法：1）标准进化算法 [2], 2；2）多目标进化算法 NSGA-II [23], ；以及 3）质量-多样性算法 MAP-Elites [10]。我们在实体机器人上测试了 MAP-Elites 方法生成的几个解决方案，并展示了如何在机器人系统原型设计过程中使用进化连杆机构的方案库。

A. 二维连杆机构模拟器

图 1 展示了用于机械连杆机构的二维模拟器中的一个连杆机构。一个连杆机构由一个曲柄、三个静态节点，以及若干连接在自由节点上的杆件组成，共同构成该连杆机构。三个静态节点表示连杆机构与机器人相连的位置，而曲柄是与电机相连的连杆杆件。模拟器会计算曲柄旋转一整圈时，连杆机构中每个节点所经过的路径。模拟器遍历曲柄的各个角度，并通过三角学计算连杆机构中所有节点的位置，从静态节点和曲柄节点开始，逐步向外推算。连杆机构中位置最低的节点被定义为机器人的足端，因为这个节点将与地面接触。足端决定了模拟器返回的节点路径。

B. 连杆机构编码

连杆机构通过一个取值范围为 0 到 1 的浮点数向量进行编码。该向量被划分为若干部分，第一部分描述曲柄和静态节点的位置，后续每一部分则描述一个添加到连杆机构中的节点。新添加的节点将通过一根或两根杆件连接到连杆机构上。仅通过一根杆件连接的节点始终会与另一根杆件保持静态连接，并且只会随其所连接的杆件一起旋转，以简化仿真。如果一个节点通过两根杆件连接，则这两根杆件可以自由旋转。这种简化带来了两个好处：1）连杆机构足端的位置由曲柄的位置确定，因此每次曲柄旋转时，足端始终沿着相同的路径运动；2）对于曲柄的每个位置，都可以通过三角学计算足端的位置，从而使连杆机构的仿真效率非常高。连杆机构编码的解码过程如图 3 所示。

C. 变异

在演化过程中，连杆机构会发生变异，从而生成新的连杆机构。变异作用于编码表示法——
表示连杆机构，然后根据新的编码重建连杆机构。通过添加由 $N(0,\ \sigma)$ 定义的高斯噪声来改变编码。噪声始终添加到表示梁长的值上，并以 0.2 的概率添加到其余值上。σ 是自适应的 [24]。它与基因组一起编码为一个浮点数，并通过添加由 $N(0,\ 0.1)$ 定义的高斯噪声来改变，

[1]步行机器人的代码和视频可在 https://github.com/EmmaStensby/linkage-evolution 找到。

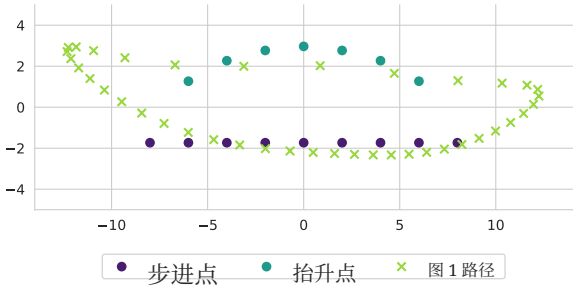


图4. 用于测量 F_p （公式1）适应度的点，以及作为参考的图1中的路径。对于所示路径， F_p 等于每个目标点到离它最近的叉形标记之间的距离之和。

概率为 0.2。将噪声添加到所有值（包括 σ）后，使用反弹法 [25]将这些值限制回 0 到 1 的范围。连杆机构中的节点数量是固定的，但参与运动机构的节点数量会有所不同。

D. 适应度函数

我们为所有方法比较了两种适应度函数。每种适应度函数都有一个供NSGA-II使用的独立多目标变体。两种适应度均根据仿真生成的足部轨迹计算。第一种适应度函数称为 F_p，用于衡量足部轨迹与由一组点定义的形状的贴合程度。第一种适应度函数的优点是，如果事先知道腿要遵循的路径，就很容易对其进行定义。第二种称为 F_{sl}，用于衡量步长和抬腿高度。它更难设计，但适用范围更广。

第一种适应度函数所用的点呈横卧D形排列。该形状底部平坦，以鼓励机器人迈步时腿端与地面接触；步线之上还设置了几个点，以鼓励腿在迈步间抬起。适应度计算如下：

$$F_p = -\sum_{i=0}^{n} D_i \tag{1}$$

其中， n 是集合中的点数， D_i 是点 i 到路径上最近点的距离。在 NSGA-II 的多目标变体中，这些点被分为两组：步进点 和抬升点，并分别测量两组点的适应度。点的位置及其划分见图 4。

第二个适应度函数用于衡量路径的抬升量和步长。步长 F_S 测量为路径中沿 y 轴方向距离路径最低点不超过 5mm 阈值的部分的长度。步长只在路径上足端运动方向与路径最低点处足端运动方向相同的部分进行测量，以避免足端沿底线来回移动。抬升量 F_l 测量为底线上的一点到沿 y 轴方向的最大距离。

其上方的一个点，此时足端正朝相反方向移动。还加入了角度误差 F_{ae} ，以避免腿的足端上下颠倒。角度误差是一个整数，等于足端朝上指向的位置数量。这些部分在 F_{sl} 中组合为

$$F_{sl} = -0.8 * F_s - 0.2 * F_l + F_{ae} \quad (2)$$

NSGA-II 使用的 F_{sl}多目标变体具有两个目标：$F_{ms} = -F_s + F_{ae}$ 和 $F_{ml} = -F_l + F_{ae}$。

许多连杆机构无法针对所有曲柄角度求解。与其移除无法完成整圈旋转的连杆机构，不如计算误差。该误差是一个整数，等于不满足连杆机构约束的曲柄位置数量。此误差会添加到所有适应度函数中，以抑制无法正常工作的连杆机构。

E. 进化算法

使用 Python 库 DEAP [26]实现了进化方法。三种方法都遵循以下通用结构：1)创建子代，2) 评估连杆机构，以及 3) 选择幸存者，并重复此过程，直到试验终止。对于这三种方法，都会从种群中选出 5000 个个人进行变异，以创建子代。对于标准进化算法和 NSGA-II，这些个人就是整个种群；而对于 MAP-Elites，则是从 MAP-Elite 保留的储备库 中随机选出的。随后评估子代连杆机构，并为下一次迭代选出幸存者。标准进化算法采用锦标赛选择，锦标赛规模为 3；NSGA-II 则采用 NSGA-II 选择 [23]。下文将结合 MAP-Elite 储备库介绍 MAP-Elites 的选择方法。参数通过手动调优确定。

F. MAP-Elites

在 MAP-Elites 中，种群被组织为一个单元格库，其中行为维度决定每个解被放入哪个单元格。每个单元格只存储一个个体，而该单元格中个体的选择采用精英策略。

我们测试了四组行为维度，以评估 MAP-Elites 方法，从而了解在原型制作时考察设计的哪些属性最有用。第一组称为 MAP-WH（MAP-宽度-高度），包含两个基于路径形状的维度：路径沿 x 轴方向的宽度，以及路径沿 y 轴方向的高度。

第二组称为 MAP-LIS（MAP-Lift-Structure），包含两个维度，一个基于连杆机构结构，另一个基于路径形状。这两个维度分别是连杆机构中梁的平均长度和路径的抬升量。路径的抬升量采用与 F_{sl} 适应度函数相同的方式计算。

第三组称为 MAP-ST（MAP-结构），包含四个维度，分别基于连杆机构的结构、梁的平均长度、机器人连接点与足端之间的最长路径、对机构功能有贡献的节点数量，以及运动节点与静止节点的比例。

最后一组称为 MAP-AU（MAP-Aurora），包含四个维度，这些维度由一个自动编码器自动定义。该自动编码器使用 AURORA（AUtonomous RObotsthatRealize their Abilities） [28]，并在整个试验过程中利用路径数据进行训练。随着自动编码器不断学习发现的路径，它会逐渐生成对可能路径搜索空间更好的表示。自动编码器有 11 个隐藏层，节点数依次为 80-64-48-32-16-4-16-32-48-64-80。层数是通过逐步增大自动编码器的规模来确定的，直到它能够重现几条测试路径为止。中间层的 4 个节点的取值决定了路径在地图中的位置。自动编码器使用 Adagrad [29] 优化器和平均绝对误差损失。自动编码器仅使用无误差的路径进行训练。为初始化地图维度，先从 5000 个随机生成的连杆机构中选取有效路径，对自动编码器训练 3000 个周期。此后，每经过 10 轮进化算法，就使用当前种群中的所有有效路径对自动编码器训练 1000 个周期。

两个二维地图的分辨率均为 100，而四维地图的分辨率为 10，因此每张地图总计有 10.000 个单元格。

G. 实体机器人

为了能够快速测试多种不同的连杆机构配置，我们使用乐高科技系列积木搭建连杆机构。这样一来，连杆机构便于重新配置，而且只需手动更换积木，就能轻松修改机构中杆件的长度。使用乐高积木的局限在于，其尺寸是固定的，因此需要选择长度最接近的积木，来近似实现演化得到的杆件长度。

机器人的底盘和间隔件使用 Fusion 360 设计，并由 HP Jet Fusion 540 和 Ultimaker 3 Extended 打印机进行 3D打印。该设计由若干通过螺钉固定在一起的小部件组成。由于塑料部件具有弹性，机器人行走时必须加装金属杆以增强刚度。组装好的机器人如图 2 所示。该设计使用的执行器为 Dynamixel MX28-AT 电机。

组装为机器人进化出的连杆机构时，制作者需要做出若干设计选择。该连杆机构是在二维环境中进行模拟的，杆件之间不会发生碰撞。实际制作时，为避免碰撞，彼此重叠的杆件需要放置在不同层，同时还要尽可能确保机构稳定。

H. 原型制作

原型制作包括搭建一台机器人，在两种质地不同的地面上测试，然后修改机器人，尝试提升其性能。我们将针对每种待比较的方法各进行一次这样的操作。我们比较使用四种 MAP-Elites 方法生成的行为库进行原型制作的效果，这些方法均采用 F_p 适应度函数演化而来。第一种地面平坦且略微湿滑，第二种地面铺有厚厚的地毯。我们预计，在地毯地面上，腿需要抬得更高，因为它.....

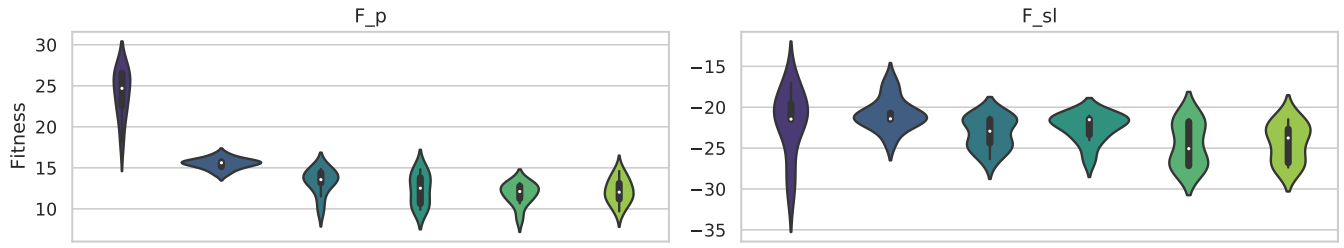


图 5. 各方法在 10 次试验中找到的最佳适应度分布。

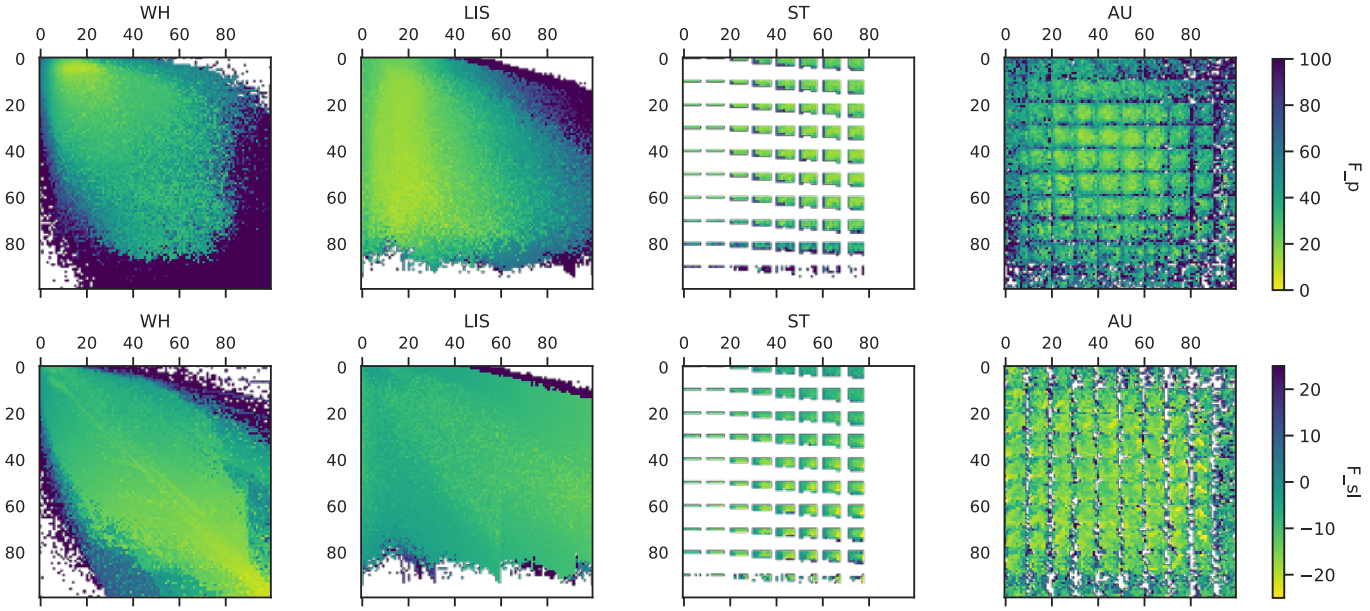


图 6. MAP-Elites方法生成的特征库。每张地图显示该方法在全部 10 次试验中于特征库每个单元格找到的最佳适应度。每个单元格的颜色表示连杆机构的适应度。

不那么平坦。地板如图 2 所示。原型制作将按以下阶段进行:

1) 根据进化得到的适应度最高的连杆机构制造机器人 2) 在两种地面上测试机器人 3) 使用进化得到的特征库修改机器人，并尝试提升其性能 4) 在两种地面上测试修改后的机器人

在阶段 3 中，为了找到更好的设计，我们将地图降采样为包含 5x5 个单元格的大小，并创建显示每个单元格中路径的地图。原始地图包含 10000 个单元格，因此降采样可确保地图足够小，让人能够查看其中展示的所有连杆机构。然后，我们从 5x5 地图中选择一个新的连杆机构，并对其进行我们认为能够提升机器人性能的修改。

III. 实验与结果

A. 二维仿真

我们对进化方法和适应度函数的每种组合分别进行了 10 次试验。每次试验运行 1 小时，在 16 个 CPU 上的实际运行时间。每次试验评估的连杆机构平均数量为 726750。

图5展示了每种方法找到的最佳适应度。使用 F_p 适应度函数时，地图方法的表现优于进化算法，且略优于 NSGA-II。使用 F_{sl} 时，所有方法的表现相近。

图6中的 MAP-Elites 解集地图展示了四种 MAP-Elites方法如何探索由其维度定义的搜索空间。这些地图展示了全部 10 次试验中每个单元格找到的最佳适应度。白色区域表示地图中未填充的部分，这主要是因为这些区域的维度描述的属性难以或无法在单个连杆机构中组合。

在图7中，我们可以看到部分试验生成的路径示例。MAP-WH 方法似乎与 F_{sl}结合时生成了更多样化的路径，而 MAP-AU 方法似乎与F_p结合时生成了更多样化的路径。除 NSGA-II 外，所有方法生成的连杆机构都过大，无法在机器人上与 F_{sl}结合成功实现。

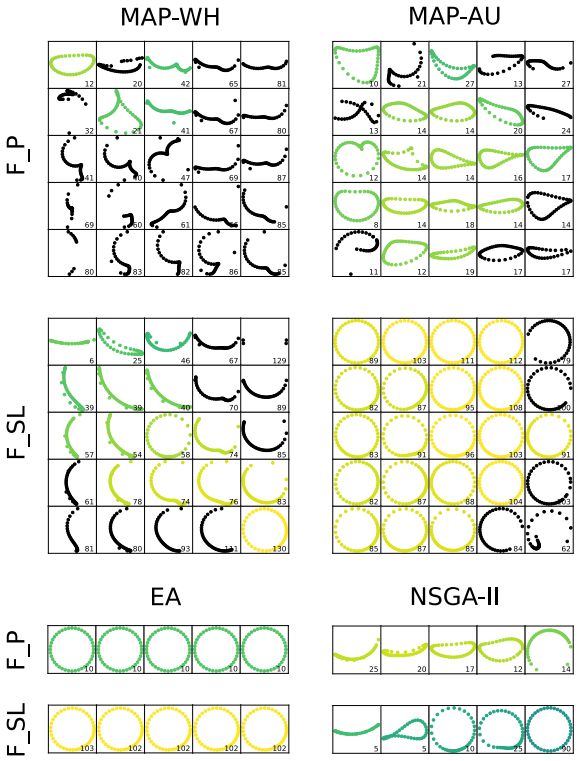


图 7. 部分方法生成的路径示例。对于 MAP-Elites，已将行为库下采样为 5x5 个单元格，并展示每个单元格中的路径。对于 NSGA-II，路径取自最终帕累托前沿。对于标准进化算法，展示了种群中适应度最高的五条路径。

B. 向现实迁移

图 8 展示了经过降采样的 5x5 地图。这些地图来自采用 F_p 适应度函数的四种不同 MAP-Elites 方法。地图中的单元格包含原始地图对应区域内适应度最高的路径。每张地图均来自相应方法的一次运行，并从四种受测方法各自的 10 次试验中随机选取。这四张地图用于原型制作过程，以选择改进后的连杆机构。

图9展示了八个经过测试的机器人。由于腿太长且脆弱，无法妥善支撑机器人机架的重量，顶排第二个机器人和第二排第一个机器人无法行走。其余机器人在两种表面上分别测试了五次，每次评估时间为10秒。人工记录了机器人在试验开始时朝向的方向上行走的距离。六个正常工作的机器人在两种地面上行走的平均距离汇总于表I。

在第一次试验中，使用 MAP-WH储备集，由适应度最佳的解生成的原始机器人在平地上朝着试验开始时面向的方向行走了 70 厘米以上。在铺有地毯的地面上，它的速度稍慢。创建修改后的连杆机构时，选择了路径更长、抬升量更大的方案，试图提高速度。然而，这种连杆机构无法产生行走步态。那里

表I 各测试机器人行走的距离。

	第一台机器人		改装后的机器人	
方法	**平地**	**地毯**	**平地**	**地毯**
MAP-WH	73.6 厘米	63.6 厘米	断裂（过长）	
MAP-LIS	断裂（过长）		40.4 厘米	48 厘米
MAP-ST	39.4 厘米	39.6 厘米	77.6 厘米	39 厘米
MAP-AU	61.8 厘米	59.4 厘米	51 厘米	50 厘米

在 MAP-WH 地图中，可供选择的有效路径并不多。这是因为宽度和高度（即该地图中的两个维度）与适应度高度相关，而适应度的目标是找到特定尺寸的路径。因此，所有有效解很可能都集中在下采样地图中的同一个方格内。

在第二次试验中，使用 MAP-LIS 动作库时，原始机器人腿部尺寸过大，无法支撑机器人机架的重量。修改后的路径采用了尺寸较小、梁更短的连杆机构，并略微提高了步高，同时仍在路径底部保留了许多采样点。这台机器人速度不快，但它是唯一一台在地毯上比在平地上行走更快的机器人。

在第三次试验中，使用 MAP-ST 动作库时，原始机器人行走得相当缓慢。原始路径的抬腿高度相当高，而且在地毯上速度没有降低。修改后的连杆机构采用了略宽的路径。修改后的机器人在平地上行走得更快，在地毯上的速度则与原始机器人相同。

在第四次也是最后一次试验中，使用 MAP-AU 动作库时，原始机器人移动得相当快，而且在两种地面上的表现相近。对于修改后的连杆机构，我们尝试选择一条底部看起来更平坦一些、抬腿高度相同的路径。然而，修改后的机器人在两种地面上的移动速度都略低于原始机器人。

IV. 讨论

在本研究中，我们比较了标准进化算法、NSGA-II 和 MAP-Elites 在生成基于连杆机构的腿部机构方面的表现。标准进化算法似乎在使用 F_p 适应度函数时收敛到了一个局部最优解，这表明在搜索这一空间时，保持一定的多样性是有益的。正如预期，NSGA-II 生成了多个连杆机构，体现了其两个适应度目标之间的权衡。

四种 MAP-Elites方法以迥然不同的方式填充了特征库，展现出各行为维度之间的差异。我们通过观察路径形状在降采样地图中的分布，对生成的连杆机构进行了定性分析。MAP-AU 生成了多样化的连杆机构，表明利用自动编码器自动定义维度，是连杆机构生成的一种可行策略。手工设计行为维度的好处在于，它能让特征库的探索变得容易得多，因为沿着地图移动时，你可以知道哪些属性会发生变化。MAP-WH 的维度仅基于路径形状，可能与适应度函数的相关性稍强；而 MAP-ST 则基于

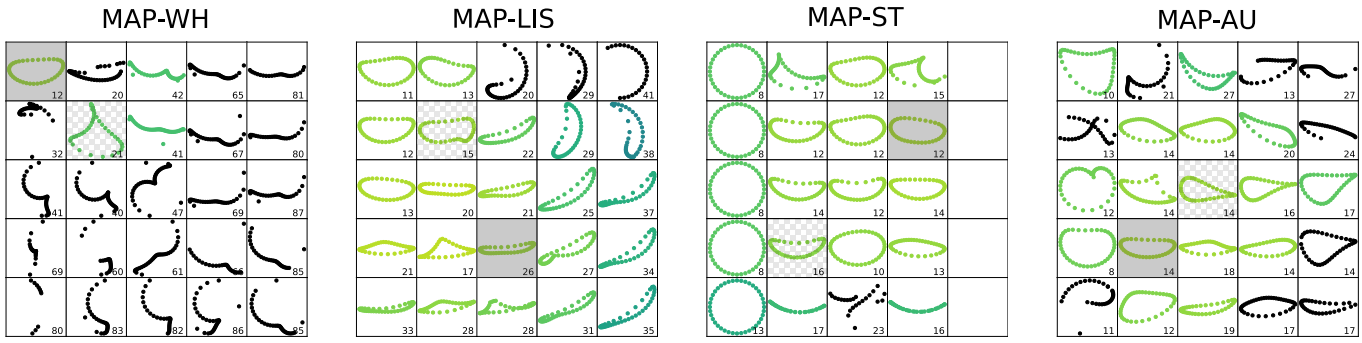


图 8。降采样地图展示了在地图不同区域中发现的连杆路径类型。路径的颜色表示其适应度。每个单元格右下角的数字表示路径的相对大小，因为路径已缩放以适应其所在单元格。四个灰色单元格展示了适应度最高的路径，四个棋盘格单元格展示了为改造后的机器人选定的路径，二者共同构成了在实体机器人上测试的八个连杆机构。

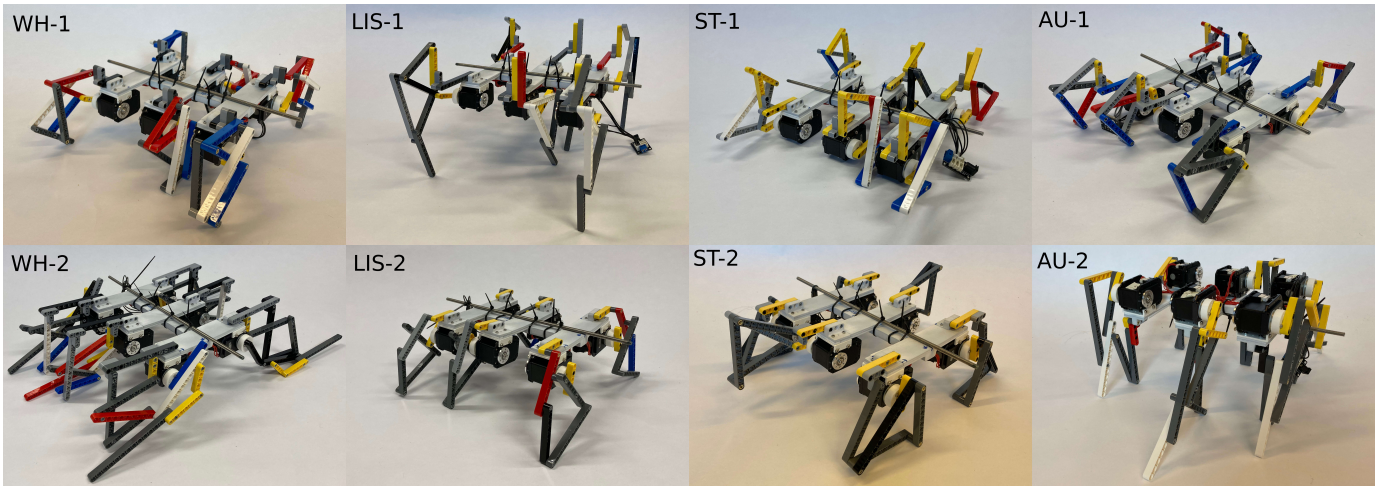


图 9。测试的八个机器人。顶行包含基于每张地图中适应度最高的连杆机构创建的原始机器人。底行包含基于降采样库制作的改造后的机器人。

仅基于连杆机构结构很难定义出一种能够填满整张地图的方式。MAP-LIS 以路径形状和连杆机构结构的组合作为其维度，看起来最有希望。

在我们比较的两个适应度函数中，F_p 产生了最适合在实体机器人上复现的路径。使用F_p找到的解决方案，其控制方式非常直观。然而，要为 F_p 创建点集，你需要确切知道想要得到什么样的足部轨迹。 F_{sl}奖励步长和抬腿高度，理论上可能产生更多样的路径，因为它为实现这些特性提供了更大的空间。然而，在我们的实验中，使用 F_{sl} 演化出的路径很快利用尺寸来产生较大的步长，因此难以在实体机器人上复现。尽管 F_p 适应度函数不如 F_{sl} 通用，但MAP-Elites方法与 F_p 的结合能够产生多样的路径。

我们研究了将 MAP-Elites 作为基于连杆的机器人的设计工具，并通过测试八台机器人进行了初步探索。要将我们的发现推广开来，还需要开展更全面的物理实验。此外，我们很可能在实验过程中逐渐熟练掌握了连杆机构的制作，这可能影响了实验结果。

由于改造后的机器人在适应度方面没有显著提升，我们无法确定哪种存档库产生的机器人适应度最高。不过，在原型制作过程中，有些存档库比其他存档库更易于使用。MAP-LIS 和 MAP-AU 的存档库最容易使用，因为其中可供选择的路径多样性更高。尤其是 MAP-LIS 存档库，选择一条改造后的路径十分容易，因为可以根据地图的尺寸推断连杆机构的长度。大型连杆机构无法制造出能够运动的机器人；而以平均梁长作为一个维度，则能够在原型制作阶段比较不同尺寸的机构。

这些连杆机构是在二维环境中进行仿真的，杆件之间没有发生碰撞。组装机器人时，我们需要决定将杆件放在哪一层，以及如何连接它们，同时保持结构稳定。对于某些连杆机构来说，这很困难，尤其是腿较长时，机构会变得脆弱。选择如何连接杆件，很可能会对机器人的速度产生重大影响

。

未来的工作可以考虑使用更坚固的材料来制作连杆机构。这样很可能能够制作出速度更快、腿更长的机器人，但每个机器人的建造时间也会增加。为了应对增加的建造时间，并使原型制作实验更加公平，可以由不同的人来建造这些机器人。另一个可以探索的方向是，通过在模拟机器人上以三维方式模拟连杆机构，将更多设计工作交由计算机完成。这样，在制作实体机器人之前，还可以测试连杆机构的组装方式以及梁之间的作用力。这也有助于提高连杆机构比较的公平性，因为这样可以避免人为错误地选择连杆机构的组装方式。

V. 结论

本文比较了几种用于设计机械连杆机构的进化方法，旨在找出进化六足机器人腿部机构的最高效方法。随后，我们通过在真实世界中测试八种进化得到的连杆机构，展示了如何将进化得到的机械连杆机构集成到基于实体连杆机构的六足机器人平台的原型制作过程中。

我们发现，在创建能够经过一组点的连杆机构方面，质量-多样性和多目标方法比标准进化算法更高效；在原型制作过程中，将连杆机构尺寸作为行为维度的 MAP-Elites 方法最为实用。我们得出结论，质量-多样性似乎是用于机器人学连杆机构生成的一种有前景的方法；而在原型制作中，反映连杆机构属性且这些属性可能在原型制作过程中发生变化的行为维度最为实用。

致谢

本研究部分由挪威研究委员会通过其卓越中心计划资助，项目编号为 262762。仿真使用了 UNINETT Sigma2 提供的资源——挪威国家高性能计算和数据存储基础设施。

参考文献

[1] T. F. Nygaard、J. Nordmoen、K. O. Ellefsen、C. P. Martin、J. Torresen 和 K. Glette，《dyret 实体测试动态机器人：现实世界进化的经验》，载于挪威人工智能学会研讨会。Springer，2019 年，第 58–68 页。[2] T. B¨ack 和 H.-P. Schwefel，《参数优化进化算法综述》，*Evolutionary Computation*，第 1 卷第 1 期，第 1–23 页，1993 年。[3] S. Doncieux、J.-B. Mouret、N. Bredeche 和 V. Padois，《进化机器人学：探索新视野》，载于进化机器人学新视野。Springer，2011 年，第 3–25 页。[4] G. S. Hornby、H. Lipson 和 J. B. Pollack，《模块化实体机器人的生成式设计系统进化》，载于国际机器人与自动化会议，*ICRA*。IEEE，2001 年，第 4146–4151 页。[5] J. Nordmoen、F. Veenstra、K. O. Ellefsen 和 K. Glette，《进化模块化机器人学中的质量与多样性》，载于 *IEEE*计算智能研讨会系列（*SSCI*）。IEEE，2020 年，第 2109–2116 页。[6] K. Chatzilygeroudis、A. Cully、V. Vassiliades 和 J.-B. Mouret，《质量-多样性优化：随机优化的新分支》，载于《黑箱优化、机器学习与无免费午餐定理》。Springer，2021 年，第 109–135 页。[7] J. Lehman 和 K. O. Stanley，《放弃目标：通过单纯搜索新颖性实现进化》，Evolutionary Computation，第 19 卷第 2 期，第 189–223 页，2011 年。[8] E. Zardini、D. Zappetti、D. Zambrano、G. Iacca 和 D. Floreano，《在软体张拉整体模块化机器人形态与控制的进化协同设计中寻求质量-多样性》，载于遗传与进化计算会议论文集，2021 年，第 189–197 页。[9] J. Lehman 和 K. O. Stanley，《通过新颖性搜索和局部竞争进化出多样化的虚拟生物》，载于遗传与进化计算会议（GECCO）。ACM，2011 年，第 211–218 页。[10] J.-B. Mouret 和 J. Clune，《通过精英映射照亮搜索空间》，arXiv 预印本 arXiv:1504.04909，2015 年。[11] A. Gaier、A. Asteroth 和 J.-B. Mouret，《通过代理辅助照亮实现数据高效的设计探索》，Evolutionary Computation，第 26 卷第 3 期，第 381–410 页，2018 年。[12] M. Yim、W.-M. Shen、B. Salemi、D. Rus、M. Moll、H. Lipson、E. Klavins 和 G. S. Chirikjian，《模块化自重构机器人系统》，IEEE《机器人与自动化杂志》，第 14 卷第 1 期，第 43–52 页，2007 年。[13] J. Nordmoen、F. Veenstra、K. O. Ellefsen 和 K. Glette，《MAP-Elites 为模块化机器人学带来强大的进化踏脚石和多样性》，《机器人与人工智能前沿》，第 8 卷，第 56 页，2021 年。[14] D. M. Bossens、J.-B. Mouret 和 D. Tarapore，《通过元进化学习行为-性能地图》，载于遗传与进化计算会议（GECCO）。ACM，2020 年，第 49–57 页。[15] A. Hagg、A. Asteroth 和 T. B¨ack，《使用质量-多样性进行原型发现》，载于自然界并行问题求解国际会议。Springer，2018 年，第 500–511 页。[16] A. Lieutier 和 J.-F. Rameau，《机械连杆设计与 NP-难问题》，《机构与机器理论》，第 82 卷，第 97–114 页，2014 年。[17] J. Cabrera、A. Simon 和 M. Prado，《使用遗传算法进行机构的最优综合》，《机构与机器理论》，第 37 卷第 10 期，第 1165–1177 页，2002 年。[18] N. N. Romero、A. Campos、D. Martins 和 R. S. Vieira，《四杆轨迹生成连杆机构最优综合的新方法》，《SN应用科学》，第 1 卷第 11 期，第 1–8 页，2019 年。[19] R. R. Bulatovi´c 和 S. R. Djordjevi´c，《轨迹生成六杆连杆机构的最优综合》，《机械科学与技术杂志》，第 26 卷第 12 期，第 4027–4040 页，2012 年。[20] S. G. Desai、A. R. Annigeri 和 A. TimmanaGouda，《一种用于步行机器的新型单自由度八连杆腿机构的分析》，《机构与机器理论》，第 140 卷，764，2019 年 747pp。– 。[21] T. Jansen，《伟大的伪装者》。010 Publishers，2007 年。[22] S. Nansai、N. Rojas、M. R. Elara 和 R. Sosa，《利用可重构连杆机构探索自适应步态模式》，载于智能机器人与系统国际会议（IROS）。IEEE，2013 年，第 4661–4668 页。[23] K. Deb、A. Pratap、S. Agarwal 和 T. Meyarivan，《一种快速且精英化的多目标遗传算法：NSGA-II》，IEEE《进化计算汇刊》，第 6 卷第 2 期，第 182–197 页，2002 年。[24] J. Teo，《用于增强实值编码遗传算法中进化搜索的自适应变异》，载于计算与信息学国际会议，2006 年，第 1–6 页。[25] J. Nordmoen、T. F. Nygaard、E. Samuelsen 和 K. Glette，《关于限制进化算法中的实值基因型》，载于进化计算应用国际会议（EvoStar 的组成部分）。Springer，2021 年，第 3–16 页。[26] F.-A. Fortin、F.-M. De Rainville、M.-A. G. Gardner、M. Parizeau 和 C. Gagn´e，《DEAP：让进化算法变得简单》，《机器学习研究杂志》，第 13 卷第 1 期，第 2171–2175 页，2012 年。[27] G. E. Hinton 和 R. R. Salakhutdinov，《使用神经网络降低数据维度》，《科学》，第 313 卷，507，2006 年 5786，504 no. pp. – 。[28] A. Cully，《通过质量-多样性和无监督描述符实现自主技能发现》，载于遗传与进化计算会议（GECCO）。ACM，2019 年，第 81–89 页。[29] J. Duchi、E. Hazan 和 Y. Singer，《用于在线学习和随机优化的自适应次梯度方法》。《机器学习研究杂志》，第 12 卷第 7 期，2011 年。