arXiv:2304.05638v1 [cs.NE] 12 Apr 2023

基于可编程逻辑控制器的控制过程中利用进化算法进行自优化与自动代码生成

工业领域应用与过程的评估方法

1st MarlonL¨oppenberg南威斯特法伦应用科学大学自动化技术与学习系统德国 Soest，loeppenberg.marlon@fh-swf.de

2nd AndreasSchwung南威斯特法伦应用科学大学自动化技术与学习系统德国 Soest，schwung.andreas@fh-swf.de

摘要——**自动化领域的数字化转型对工业过程中的数据采集与处理提出了新的要求。必须正确解释和评估所采集数据与循环过程序列之间的逻辑关系。为解决这一问题，本文提出一种基于进化算法的新方法，用于对复杂过程的系统逻辑进行自优化。根据遗传算法结果，通过对解进行解码，推导出用于系统实现的程序代码。这通过一种灵活的系统结构实现，该结构包含上游单元、中间单元和下游单元。在中间单元中，定向学习过程与系统副本和评价函数在闭环中交互。代码生成策略由冗余、优先级、排序和性能推导构成。本文针对一个面临多目标优化问题的工业液体处理站过程，对所提出的方法进行了评估。**

索引词—**进化算法、可编程逻辑控制器代码生成、多目标优化**

I. 引言

生产工厂应用于制造业和自动化行业的许多领域。近年来，过程自动化领域经历了向数字化和智能联网转型的变革，这被称为工业 4.0 [1]。随着对自动化过程的重新诠释，越来越多的数字处理被转移到现场层级，例如用于摄像头的图像处理 [2]或用于物联网（物联网）应用的无线通信接口扩展 [3] [4]。在这一层级，来自传感器和执行器的所有数据都由可编程逻辑控制器（可编程逻辑控制器）集中处理和评估。目前，可编程逻辑控制器的代码由程序员编写，程序员需要详细了解并掌握该过程。根据待实施解决方案的规模，这项任务非常繁重且耗时。此外，所开发的程序结构具有很高的可变性，并且取决于程序员的技能。改进方法是通过自动化过程解决这些问题。

鉴于所述问题，提出了一种具有自动代码生成的自优化评估过程。适用于此实现的方法是进化算法（进化算法）。它将情境理解与基于随机和元启发式基础的不同性能目标优化相结合。为了生成求解方法，优化配置与数字孪生（数字孪生） [5] 进行周期性评估交互。该问题被视为一个多目标优化问题（MOOP） [6] ，从而在目标空间中形成帕累托前沿。通过将优化与动态系统分离，实现了解耦，并通过系统配置将该过程划分为主动部分和被动部分。系统配置的灵活接口定义决定了其与实际系统的交互方式。根据系统特定问题实现了适应度函数，以控制优化过程。还可以使用特定参数定义此函数，以影响优化过程。所提出的方法减少了机器设置时间，并自动检测和优化所有系统设置。本文的贡献可总结如下：

- 一种用于自优化和自动代码生成的系统配置，包含一个上游单元和一个下游单元。
- 展示动作空间适应在冗余优先级、序列排列和性能推导方面的可能性。
- 呈现中间单元中进化算法、仿真、评估和代码推导之间的交互。
- 在具有 MOOP 的工业控制过程环境中实现该系统配置。
- 通过系统知识将交互式进化算法适应与组合求和加权方法进行比较。

本文回顾了第二部分中的现有文献，并在第三部分概述了进化算法。在第四部分介绍了自优化与信息处理的方法。第五部分展示了结果和实施情况，并在第六部分给出了总结。

II. 相关工作

技术和科学方法的实施已经确立了新的标准。这种数字化转型也称为工业 4.0 [7]，包括去中心化、自动化、智能工厂、增强现实和虚拟现实等。物联网技术支持无线交换 [8]，以用于进一步的应用 [3] [4]。基于这些原则，人工智能（AI）在工业中的引入得以实现，并已应用于旋转机械的故障诊断 [9] 或水下航行器的路径规划 [10]等领域。与流程相关的逻辑理解正变得越来越重要。在 [11]中，采用了基于博弈论的学习方法来优化一个运行装置。该应用领域被视为进化算法研究的基本范围。尽管机器学习（ML）中的不同方法为不同问题提供了解决方案，但进化算法在这一领域尤其突出。进化算法属于随机的元启发式优化方法 [12]，，其灵感来源于生物过程和物理条件 [13]。决策策略 [14]、量子计算 [15] 以及许多其他因素的影响，促成了该领域中个体策略和方法的多种变体与特征的发展。主要分类包括遗传算法（GA），其优化基于倾向与组合；遗传编程（GP），其中会对特定结构进行调整；以及进化策略（ES），其主要基于数值向量优化 [16] [17]。该方法的实施基于遗传算法的遗传原理和进化策略的策略。进化算法的广泛应用可在多目标优化问题的应用领域中得到体现。多目标优化问题涉及对多个传递函数进行最小化或最大化，并将其作为优化目标。该方法利用这些优势，并将其应用于工业问题。由于相互关联性，通过搜索解决方案可能得到不同的结果。收敛解被描述为帕累托最优解，而这些解共同构成帕累托前沿 [6]。鉴于多个目标函数和帕累托最优解的复杂性，具体处理方式会根据优化和决策过程而有所不同。在 [18] 中进行了区分

三种偏好表达（PA）会导向不同的求解方法。偏好表达可分为先验、渐进和后验三种。先验偏好表达的应用如 [19], 所示，其中一个具有多个目标依赖关系的系统通过单一加权奖励函数持续优化。利用对系统行为的深入了解来约束解空间。渐进偏好表达也称为交互式优化，其运行方式是在持续变化的求解与决策过程中进行操作。根据关注重点，实现 [20]有不同的已知方法和策略。本文提出的方法基于最强的幸存者，可实现稳定的学习结果。本研究不采用诸如 [21] 之类的后验偏好表达，因为系统优化只会在一个子领域内进行考量。

III. 进化算法在过程优化中的应用

本文提出采用进化算法，它兼具随机优化与元启发式优化的优势。通过借鉴自然与进化过程的原理，实现持续适应与优化，如图1所示。

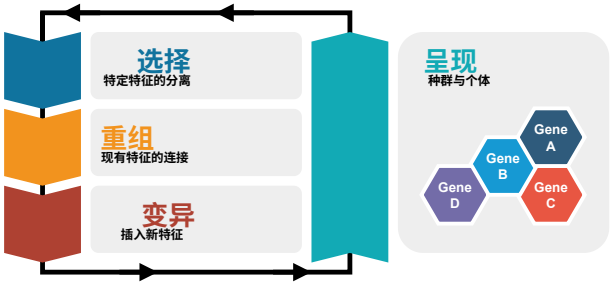

图 1. 进化算法的结构。通过选择、重组和变异进行进化的循环过程。个体的遗传特征在此过程中不断适应，并改变其在种群中的顺序。

A. 种群与个体结构

交换和修改是进化算法结构中的基本阶段。所创建的知识和理解必须集中存储。为此，将集合 P 创建为一个种群，其基数为 $|P| = p$。这代表基本单元，其中包含个体集合 p ，即集合 I，其中 $I \in \{x \mid P\ (x)\}$。每个个体都包含 $E\ (m)$ 个特定元素，以及 $T\ (n)$ 个称为基因的信息元组。在整个过程中，种群和个体的结构保持一致。

B. 选择过程

进化过程的一个重要方面是个体的选择 I。个体选择基于特定元素和元组信息进行，并定义限制顺序 R。因此，从种群中基于个体元素 $E(m)$生成一个预序（P, R）。基于最小化 $\forall\ y \in P:\ I \leq y$ 或最大化$\forall\, y \in P: I \geq y$，选择具有所需特征的特定个体 $I \in P$。被排除的个体 y从种群 P中可以在稍后阶段进行选择，前提是满足该条件。

C. 重组方法

可通过重组获取已有信息。该方法从个体专属元组 $T(n)$ 中选取优选元素 a ，如 $a \in \{x_1, x_2, \ldots, x_n\}$ 所述。所有选定元素及 a 中包含的信息会组合到新的元组 M 中，其中 $\{x|\exists\ a \in M : x \in a\}$。所选元素的信息通过新的单一结构 I_M 发现。通过以初始结构为目标，将重组个体 I_M 纳入种群 P，其中 $I_M \in P$。

D. 变异技术

可通过变异技术实现显式修改。从个体专属元组 $T(n)$ 中选出要修改的特定元素 a ，并使用 $a \in \{x_1, x_2, \ldots, x_n\}$。在 $E(m)$ 所定义的元素边界范围内，对选定元素及信息 a 进行变异，从而得到 $a \in \{x_{1\ new}, x_{2\ new}, \ldots, x_{n\ new}\}$。保留的元素与变异后的集合在 M 中组合，其中 $\{x \mid \exists\ a \in M:\ x \in a\}$。通过保留所述结构， $I_M \in P$，其中 I_M 是一个可纳入种群 P 的变异个体。

IV. 过程规范与系统概述

为解决自优化和自动代码生成问题，所提出的系统配置可分为上游单元、中间单元和下游单元。图2展示了所开发系统配置的概览及其总体布局。

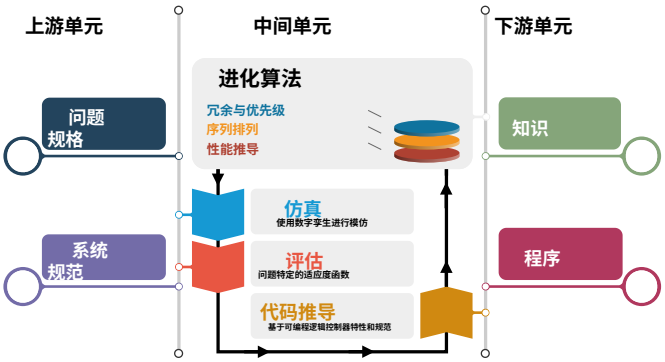

图2。所提出的自优化和自动代码生成框架示意图。串行接口将系统灵活地连接到定制的问题表示和输出表示。

所开发的系统配置在主侧具有可连接的接口，用于适应具体问题特征以及系统环境的规范，并作为上游单元使用。中间单元位于系统配置的中央，由优化算法、作为数字孪生（DT）实现的系统环境副本、带有问题专用适应度函数的评估单元以及代码推导单元组成。与之相对的是下游单元，其次侧配有系统知识分析器的输出和程序解码单元头。该配置使所提出的方法具有灵活性，并允许在存在替代起点的情况下按顺序进行扩展或修改。对系统配置交互作用的考虑是被动的，因为评估循环会与系统环境的数字孪生（DT）进行交互。根据IEC 61131、IEC61499或IEC 61512规范，将解推导为标准可编程逻辑控制器程序是可行的。优化过程结束后，所获得的知识和生成的代码可以主动用于特定系统。以这种方式创建的场景会基于评估，在仿真与观测之间被动地经历连续的周-期优化循环，随后进行主动的解输出。

A. 个体和种群的信息处理

所介绍应用的一个核心方面，是信息的编排与呈现。这是通过使用个体和种群实现的，它们在进化算法中发挥着核心作用。该表示中嵌入了影响和控制过程所需的全部系统参数、目标函数、约束函数和约束条件。对于可编程逻辑控制器，可以基于功能块、指令表、梯形图、结构化控制语言或顺序控制来实现标准化方法。遵循IEC 61131和 IEC61499的基本原则 [22] [23]。这为定义PLC程序代码提供了灵活且标准化的结构。所采取的措施支持一种跨层级方法，在这种方法中，可以实现不同的代码生成策略；在进化算法中，这些策略被定义为个体。代码结构以序列的形式加以考虑，其中各组成部分决定行为。重点在于冗余和优先级的定义。动作空间可以根据动作，定义为包含或排除相关部分。这样便可优化提供控制逻辑上下文的相关序列。另一个需要考虑的因素是序列的编排。通过记录其激活顺序中的差异，并将其与待替换的简化冗余部分进行比对，可以识别这些差异。通过识别层级循环并解析复杂结构，可以生成更高层级的序列结构。通过组合这两种变体，可以推导出对总体性能要求的另一种表示，并将其用作过程指令。还可以进一步推导出一条通用的过程指令，用于表达系统行为及其知识。这有助于更好地理解问题与解决方案之间在整体背景下的相互作用。影响个体和种群实现的可能表示取决于具体系统，并可针对该方法进行调整。在实现过程中，为个体表示定义了一种表格结构，见图 3。

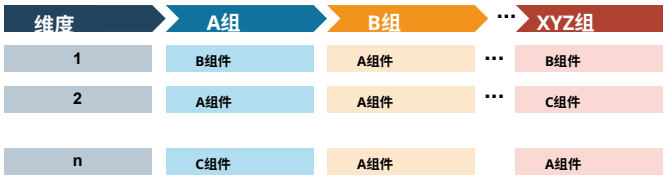

图 3. 顺序个体结构示意图。群组组件的排列方式和维度的顺序会在进化过程中发生变化。

该排列为集成组件、运行模式提供了灵活的表示方式，并清晰地呈现了依赖关系。可以更改组件组，以纳入不同的效果，也可以调整所显示维度的数量，从而改变复杂度。

B. 通过中间单元实现自优化

必须特别注意中间单元的实现。优化过程由集成的进化算法实现，该算法通过不断变化并自主调整可能的解决方案，持续探索搜索空间。为防止系统在优化过程中陷入局部极值，进化算法的功能根据以下重点进行定义。选择函数旨在从种群中选出最适合目标参数的个体。所选个体将用于变异函数。相比之下，算法会选取表现最佳个体的遗传特征并将其重新组合，以创建新的个体。为在进化过程中引入新特征，变异函数会随机选取个体的遗传特征并对其进行变异。算法结构清晰，确保了学习策略简单高效。进化算法的优化由动作空间和适应度函数的确定来定义。文中展示了动作空间的表示方法，这些方法也可扩展至其他应用。根据问题定义，必须定义适应度函数，以评估与适配后系统的数字孪生进行主动交互的效果。这代表了适配的目标，还可进一步补充特定参数。为从加权进化中生成可用解决方案，系统集成了用于生成的代码推导。所述实现以循环方式调用。每个周期结束时，根据加权评估结果对种群中的个体进行排序。优化运行的设计可依据适应度函数的质量标准来定义。所提出的中间单元通过自优化实现自动代码生成，并生成系统知识和程序代码。

V. 实验与结果

分析在如图 4 所示的液体实验室装置中进行。根据定制系统的特性，该液体实验室装置被视为一个多目标优化问题，可按照公式 1 所示的一般形式进行描述。

$$\begin{aligned} \text{min or max} \quad & f_m(\mathbf{x}), \quad m = 1, 2, \ldots, M; \\ \text{subject to} \quad & g_j > 0, \quad j = 1, 2, \ldots, J; \\ & h_k = 0, \quad k = 1, 2, \ldots, K; \\ \text{with} \quad & x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \ldots, n; \end{aligned} \tag{1}$$

考虑行为与 $\mathbf{x}$相关，$\mathbf{x}$是一个包含n 个变量的向量$\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ 以及约束函数 g_j 和 h_k，它们与 J 不等式约束和 K 等式约束相关联。变量限制发生在下边界 x_i^L和上边界 x_i^U的区域内。在这些限制下，独立的目标函数 f 形成一个变量空间 D。根据 $\mathbf{x} \in D$，变量向量由$\mathbf{x} \in D$称为可行解或不可行解。 M 多个目标函数 f_m 的结果是一个称为目标空间的多维空间 Z，其中每个可行解 f_x 都对应一个点 z，其中 $f_{\mathbf{x}} = \mathbf{z} = (z_1, z_2, \ldots, z_n)^T$ [24] [6]。由于目标函数之间存在内在冲突，不同向量之间会发生帕累托支配，从而产生 $\mathbf{x}' \preceq \mathbf{x}$ 的被支配比例。对于 $\mathbf{x} \in D$，目标函数会导出一个有效解，其中 $\{F(x) = (f_1(x)\ f_2(x) \ldots, f_m(x))\}$ 支配 $F(x')$。从集合 $\mathbf{x}$ 中的一个有效解出发，帕累托有效解可定义为集合 $\mathcal{P}$，其定义为 $\{x \in D | \neg\ \exists\ x' \in D\colon\ F(x') \preceq F(x)\}$。目标函数之间的相互依赖关系产生了若干部分可能性 $\mathcal{PF}$，称为帕累托前沿，由 $\{F(x) = (f_1(x)\ f_2(x) \ldots, f_m(x) \mid x \in \mathcal{P}\}$ [18] 描述。

A. 液体实验室装置

该系统以中央储罐 T 为核心，通过管道进行注水和排水。具体的体积流量取决于泵 P_1、 P_2 和 P_3各自的启动情况。传感器 S_1、 S_2 和 S_3可检测储罐系统的实际液位。实验室装置的运行分为不同的功能状态。实验室装置可在自动或手动模式下运行，通过开关 B_1进行选择。在自动模式下，泵根据当前液位启动，以完成注水和排水。自动运行状态由指示灯 L_1显示。在手动模式下，设备管理员可使用 B_2 和 B_3 按键分别控制注水泵和排水泵。液体实验流程由可编程逻辑控制器集中控制。程序代码按照 EN 61131 [22] 标准编写，并使用列出的传感器和执行器。

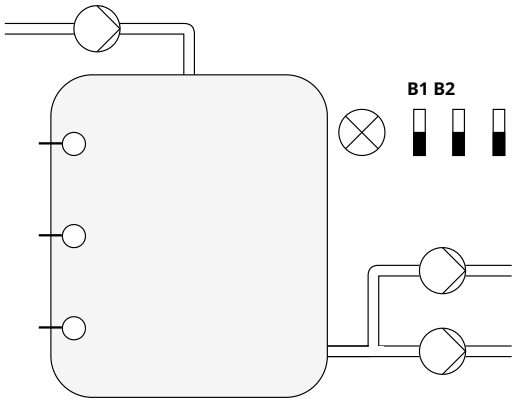

图4. 液体实验室站的示意布局，其中标出了各个执行器和传感器。水箱中的液位 T 通过泵P_1 升高，并通过泵 P_2 和泵 P_3降低。

表 I 对实验室装置的组件进行了输入和输出分类。系统状态和输出组件根据输入组件进行调整。流程优化依据最大输送量 P_{trans}、最低能耗 P_{energy} 和代码结构 P_{code} 进行。液体实验室站的功能和运行结构分为自动模式和手动模式。

表 I 液体实验室装置的组件清单。分为被动输入部件和主动输出部件。

标签	Type	Unit	描述
S_1	输入	传感器	检测液位位置 1
S_2	输入	传感器	检测液位位置 2
S_3	输入	传感器	检测液位位置 3
B_1	输入	开关	自动手动模式
B_2	输入	按钮	手动灌装
B_3	输入	按钮	手动清空
P_1	输出	Pump	活动泵 1
P_2	输出	Pump	运行中的泵 2
P_3	输出	Pump	运行中的泵 3
L_1	输出	灯	运行中的灯 1

- 泵会自动进行灌装和排空P_1、 P_2 和 P_3。其启用取决于传感器检测到的液位 S_1、 S_2 和 S_3。当 B_1 发出高电平信号时，系统处于自动模式。该模式由指示灯显示 L_1。
- 手动填充通过开关 B_1 的低电平信号和填充按钮 B_2 的高电平信号实现。这会启动泵 P_1。
- 储罐的手动排空由泵P_2 和 P_3执行。它们由来自开关 B_1 的低电平信号以及来自排空按钮 B_3的高电平信号激活。

延迟激活和停用是由系统滞后引起的。

B. 系统迁移与实现

所描述的液体实验室装置处于自动模式。优化在系统的数学和技术限制范围内进行。采用总权重法和渐进式 PA 进行预先 PA 后，求解所获得的MOOP依赖关系，其中会进行交互式优化。不满足边界条件的条件将被排除。基于液体实验室站流程的目标 P_{trans}、 P_{energy} 和P_{code}，实施该优化方法。对于先前的 PA，根据总权重法，将传输 f_1、能耗 f_2 和程序代码结构 f_3 的当前目标值组合成一个独立函数，并将其作为进化算法的适应度函数，而

$$F_{fit} = \left(\frac{1}{1+\alpha_1(P_{trans}-f_1)} + \frac{1}{1+\alpha_2 f_2} + \frac{1}{1+\alpha_3(P_{code}-f_3)}\right). \quad (2)$$

所得适应度函数 F_{fit} 可以在受参数 α_1、 α_2 和 α_3影响的最大值处进行优化。要定义适应度函数的参数，需要理解系统行为。满足这些要求的所创建个体会被存储在进化算法种群中。与这种方法相反的是对进化算法进行渐进式构建的技术实现。在这种方法中，所有具有相关基因和结果的个体都会被存储在

种群。通过对最优个体进行选择、重组和变异，持续的基因传递使整个过程不断演进并优化。

C. 性能呈现

系统组件的特定激活行为会影响液体实验室站的运行过程并改变流程。将 MOOP 转换到目标空间 D 后，便可表示耦合目标函数的影响和作用。如图 5 所示的比较展示了不同目标函数之间的依赖关系。

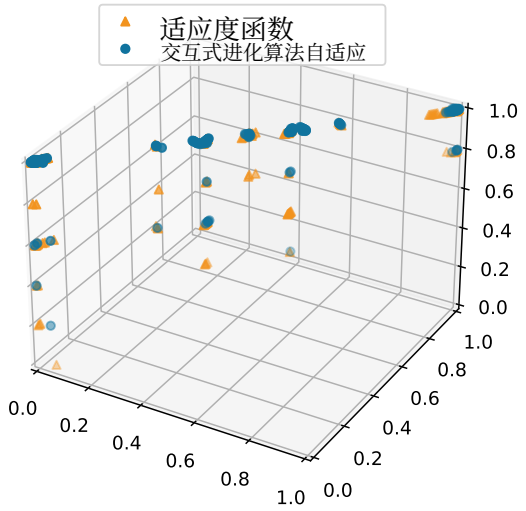

图5. 采用适应度函数的先验PA（橙色三角形）与采用遗传传递的渐进式PA（蓝色圆形）的比较。运输与能量的结合使其在目标空间中呈平行定位。

从系统和过程知识的角度来看，搜索空间中的解决方案受到适应度函数的限制。通过促进和塑造特定的解决方案，在整个优化过程中实现了持续学习方法。

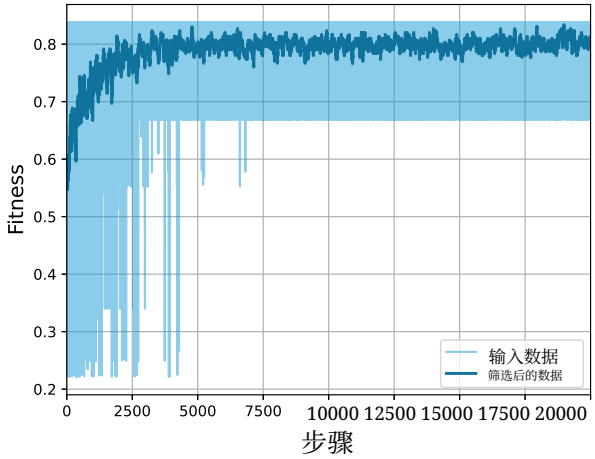

图6. 展示采用适应度函数的先验PA在经验积累下成功率不断提高。通过对不同目标函数进行加权和平衡，实现稳定状态。

D. 解码与码分

基于MOOP的性能呈现，可以从先前或渐进式PA中选择一个可能的个体解。表 II表示一个个体 I ，该个体在液体实验室装置的数字孪生上进行评估。

表 II 个体的遗传结构表示激活组成。进化算法通过选择、重组和变异修改内容与信息。

行:	输出:	输入:	算子:	取反:	模式
0	P1	S3	=	¬	A
1		B1	∧		A
2	P2	S2	=		A
3		B1	∧		A
4	P3	S1	=		A
5		B1	∧		A
6	L1	B1	=		A

液体实验室站的输出、输入、算子、取反和运行模式均以组件组的形式在表格表示中实现。通过解码单元调用各个 I 的结构化排列，即可推导出液体实验室站的过程控制系统逻辑。该结构化表示的实现方式与结构化控制语言或指令表等编程语言相同。

$$\begin{aligned} P1 &= \bar{S}3 \text{ and } B_1; \\ P2 &= S2 \text{ and } B_1; \\ P3 &= S1 \text{ and } B_1; \\ L1 &= B_1; \end{aligned} \tag{3}$$

所示解决方案表示了液体实验室装置自动运行的程序代码结构中的传感器和执行器连接。

VI. 结论

本文提出了一种具有自动代码生成功能的新型自优化方法。该方法基于进化算法，可集成到基于可编程逻辑控制器的控制流程中。文中阐释并介绍了进化算法中的选择、重组和变异。所开发的框架分为上游、中间和下游单元，并可进行专门参数化。通过与数字孪生和评价函数交互，实现持续探索。通过在可定制的结构化表示中进行冗余处理、优先级设定、排序和性能推导，对系统知识和解决方案提案进行优化。为展示现实应用中 MOOP 的复杂性，该方法在液体实验室装置中得到实现和评估。文中展示了结果和程序代码推导。

未来的工作将研究由多智能体实现的知识迁移和搜索空间规范。应使用双射功能将一对一可追溯性集成到进化过程中。通过使用神经网络表示种群，可以拓展其潜力和可能的应用。

参考文献

[1] H.Kagermann 和 W. Wahlster，“工业4.0十年”，Sci，第4 卷，第 26 期，第 26–页，2022。[2] M.U. K.Khan 和 C.-M. Kyung，“用于高速低功耗背景减除的泊松混合模型”，载于《智能传感器与系统》。查姆：SpringerInternational Publishing，2020，第 1–23页。[3] V. Chaudhary、A. Kaushik、H.Furukawa和A.Khosla，“综述——迈向由第五代人工智能和物联网驱动的、基于二维MXene和硼烯的可持续智能传感器”，ECS SensorsPlus，第1卷，第1 期，第 013601 页，2022 年 4 月。[4] M. Zarzo、A.Perles、R. Mercado和 F.-J.Garc´ıa-Diego，“使用节能型物联网无线传感器对历史建筑中的温度波动进行多变量表征”，Sensors (Basel, Switzerland)，第21 卷，第 7795 期，第 7795–页，2021。[5] M. Singh、E. Fuenmayor、E. Hinchy、Y.Qiao、N. Murray 和 D.Devine，“数字孪生：从起源到未来”，Appl.Syst.Innov，第 4 卷，第36 页，2021 年 05 月。[6] K. Deb，《使用进化算法进行多目标优化》，Wiley Interscience系统优化系列。奇切斯特 [u.a：Wiley，2002。[7] M. Ghobakhloo，“工业4.0、数字化与可持续发展机遇”，Journalof Cleaner Production，第 252 卷，第 119869 页，2020。[8] S. Li、L. D. Xu 和 S. Zhao，“物联网：综述”，Information Systems Frontiers，第 17卷，第2期，第 243–259 页，2015 年 4 月。[9] R. Liu、B.Yang、E.Zio 和 X. Chen，“用于旋转机械故障诊断的人工智能：综述”，MechanicalSystemsand Signal Processing，第 108 卷，第 33–47 页，2018。[10] A.Alvarez、A.Caiti 和 R. Onken，“可变海洋环境中自主水下航行器的进化路径规划”，IEEE J. Ocean. Eng.，第 29 卷，第 2 期，第418–429 页，2004。[11] D. Schwung、J. Reimann、A.Schwung 和 S. Ding，《智能制造系统：一种基于博弈论的方法》。Springer NatureSwitzerland AG2020，2020年 01 月。[12] H. Maier、S. Razavi、Z. Kapelan、L. Matott、J.Kasprzyk和 B. Tolson，“导论性概述：使用进化算法和其他元启发式算法进行优化”，Environmental Modelling & Software，第 114卷，第 195–213 页，2019。[13] K. Qiao、J. Liang、K.Yu、M.Yuan、B.Qu 和 C. Yue，“基于自适应资源分配的差分进化用于约束进化优化”，Knowledge-Based Systems，第 235卷，107653，2022p. 。[14] A.Kaim、B.Bartkowski、N. Lienhoop、C. Schr¨oter-Schlaack、M. Volk 和 M. Strauch，“将生物物理优化与经济偏好分析相结合用于农业土地利用分配”，EcologyandSociety，第26 卷，2021 年 03 月。[15] W. Deng、H.Liu、J.Xu、H.Zhao 和 Y. Song，“用于深度置信网络的改进量子启发式差分进化算法”，IEEETrans Instrum Meas，第69 卷，第 10 期，第 7319–7327 页，2020。[16] A. Slowik 和 H. K wasnicka，“进化算法及其在工程问题中的应用”，Neural Computing andApplications，第 32 卷，第 16期，第12 363–12379 页，2020 年 8月。[17] W. Deng、X. Zhang、Y.Zhou、Y. Liu、X.Zhou、H.Chen和 H. Zhao，“用于多目标问题的增强型快速非支配解排序遗传算法”，InformationSciences，第 585 卷，453，2022441pp。– 。[18] D.Veldhuizen 和 G. Lamont，“多目标进化算法：分析最新进展”，Evolutionary computation，第8卷，第 125–47页，2000 年02 月。[19] D. Schwung、S.Yuwono、A. Schwung 和 S. Ding，“使用可编程逻辑控制器信息增强的强化学习进行能源最优生产策略的去中心化学习”，Computers & Chemical Engineering，第 152卷，第 107382 页，2021 年 05 月。[20] W. S.Shin 和 A. Ravindran，“ 交互式多目标优化：综述 I——连续情形”，Computers&Operations Research，第18 卷，第 1 期，第97–114 页，1991。[21] D. Inghels、W. Dullaert、E.-H. Aghezzaf 和 R.Heijungs，“实现绿色废弃物材料回收与能源回收之间的最优权衡”，Waste Management，第 93卷，第 100–111页，2019。[22] “DIN EN 61131-3 可编程逻辑控制器——第 3 部分：编程语言”，2014年 6 月。[23] “工业控制系统功能块——第 1 部分：架构”，2014 年 9月。¨K. Miettinen，《非线性多目标优化》，运筹学与管理科学国际系列第 12 卷。波士顿 [24] u.a：Kluwer Academic Publishers，1999。 [