基于 LLM 的多智能体化学过程优化方法

Tong Zeng,† Srivathsan Badrinarayanan,† Janghoon Ock,† Cheng-Kai Lai,† and Amir Barati Farimani*,‡,¶,§

†*Department of Chemical Engineering, Carnegie Mellon University, PA 15213, USA*

‡*Department of Mechanical Engineering, Carnegie Mellon University, PA 15213, USA*

¶*Department of Biomedical Engineering, Carnegie Mellon University, PA 15213, USA*

§*Machine Learning Department, Carnegie Mellon University, PA 15213, USA*

E-mail: barati@cmu.edu

arxiv:2506.20921v1 中译本

摘要

化学过程优化对于最大化生产效率和经济性能至关重要。传统方法，包括基于梯度的求解器、进化算法以及参数网格搜索，在操作约束不明确或不可用时变得不切实际，要求工程师依赖主观启发式方法来估计可行的参数范围。

为了解决这一约束定义瓶颈，我们提出了一种多智能体框架，该框架由大型语言模型（LLM）智能体组成，能够自主地从最小的过程描述中推断操作约束，然后使用推断出的约束进行协同优化指导。

我们的基于 AutoGen 的代理框架采用了 OpenAI 的 o3 模型，并配备了专门用于生成约束、参数验证、模拟执行和优化引导的智能体。

通过两个阶段——利用嵌入式领域知识自主生成约束，随后进行迭代多智能体优化——该框架消除了对预定义操作范围的需求。

在成本、产率以及产率与成本比率指标上，在水解烷基化过程上的验证表明，该框架不仅展示了与传统优化方法相当的性能，而且实现了更好的计算效率，需要更少的迭代次数就能收敛。

我们的方法在 20 分钟内完成收敛，相比网格搜索实现了 31 倍的速度提升。

除了计算效率之外，该框架指导下的推理搜索展示出了复杂的过程理解能力，能够正确识别效用权衡，并应用基于领域的启发式方法。

这种方法对于操作约束定义不明确或不可用的优化场景具有显著潜力，特别是新兴过程和改造应用。

介绍

化学过程的高效优化是流程系统工程中的一个核心目标，直接关系到生产效率和经济表现[1]。传统的优化方法包括基于梯度的求解器，如 Interior Point OPTimizer（IPOPT）[2] 和 Sparse Nonlinear

OPTimizer（SNOPT）[3]，这些依赖于目标函数的基本平滑性和可微性[4]。当这些要求无法满足时，替代方法如进化算法[5] 和无导数方法[6] 提供了可行的解决方案。此外，穷举搜索技术如网格搜索系统地评估了设计空间内所有离散参数值的所有组合[7]。

然而，这些优化方法需要一个完整且定义良好的设计空间，并描述操作约束条件。这一要求使得在可行参数范围或过程操作条件不完整或不可用的情况下应用这些方法变得困难。在这种情况下，工程师通常依赖启发式方法或领域知识来估计可行范围[8,9]。这些启发式方法高度主观，并可能导致变化的优化结果。数学仿真平台如 IDAES[10] 和建模库如 Pyomo[11] 提供了灵活的环境来构建详细的流程模型，但仍依赖用户在运行优化前定义变量约束和平滑的目标函数。这一约束定义瓶颈显著限制了此类优化技术的应用，特别是对于非专家用户或操作配置尚未确立的新工艺。

大型语言模型（LLMs）的最新进展提供了可能的方法来解决这些限制。LLMs 在复杂问题解决任务[12,13] 中表现出强大的推理能力，并通过少量示例学习[14] 有效地适应以前未见过的领域。最值得注意的是，最近的研究表明 LLMs 可以自主进行科学研究，从假设生成和实验设计到数据分析和手稿编写[15]，展示了它们在独立科学推理和发现方面的潜力。

这些通用能力已经转化为化学工程应用方面的显著进展。在合成规划中，LLMs 根据战略需求[16] 评估合成路线，表明传统上需要多年培训的专业知识可以通过自然语言界面变得更加易于获取。对于材料发现，基于 LLM 的代理如 ChemCrow[17] 已成功使用自动化实验室平台规划和执行合成。对于控制系统设计[18–22]，LLMs 使得在无需专门编程技能的情况下实现更灵活的过程控制方法成为可能。

然而，大多数现有的努力集中在孤立的决策步骤或下游控制上，对将大型语言模型整合到稳态模拟或过程优化管道中的探索有限，利用协作多代理大型语言系统自动生成稳定状态过程优化约束条件的研究也基本未被探索。这项工作通过引入一个自主的、多代理框架来解决这一空白，该框架结合了语言模型推理与过程模拟以实现操作条件优化。我们解决了在可行的操作范围未知或定义不清时自动生成约束条件的挑战。

我们的方法利用了 AutoGen 平台[23] 与 OpenAI 的 o3 模型[24] 创建了一个多代理系统，在该系统中，语言模型代理合作生成现实的约束并迭代优化工艺参数。框架在实际化学流程图上进行了评估——使用 IDAES[10] 模拟的脱烷基化（HDA）过程，并从三个指标进行评估：成本、产率和产率与成本之比。我们选择了 IPOPT 和网格搜索作为代表性的基准：IPOPT 作为一种广泛用于化学工程领域的先进梯度优化求解器，以及网格搜索作为一种无梯度的穷举方法，用作全局优化性能的基础线。结果显示了相当的性能，同时减少了迭代次数和计算时间。这种方法可以显著降低工艺优化的专业门槛，并在工业环境中更广泛地应用先进的优化技术，特别是在新兴工艺或对现有操作进行改造的情况下，这些情况下运行约束条件特征不明显。

方法论

框架架构

我们提出了一种使用 AutoGen 的 GroupChat 架构的多智能体交互系统，以解决复杂优化任务中单个智能体方法的基本局限性。各个智能体保持针对特定功能的专注提示，防止当冗长的多功能提示淹没 LLM 注意力机制时发生的上下文稀释。每个智能体的专业角色使得可以直接调用函数，而不是解析复杂的输出，从而降低实现复杂性和错误的可能性。模块化设计允许独立调试和修改优化组件，这对于迭代算法开发和部署灵活性至关重要。

我们的框架由五个专门的 LLM 代理组成，其中每个“代理”代表一个具有特定提示和功能调用能力的专用途 LLM 实例，这些能力和提示是为特定优化任务设计的。该系统通过两个阶段运行：约束生成，随后在协作多代理环境中进行迭代优化，如图 1所示。在整个过程中，“自主性”指的是每个代理仅使用嵌入式领域知识、生成的约束和共享的优化历史记录，在无人干预的情况下推理、行动和迭代的能力。

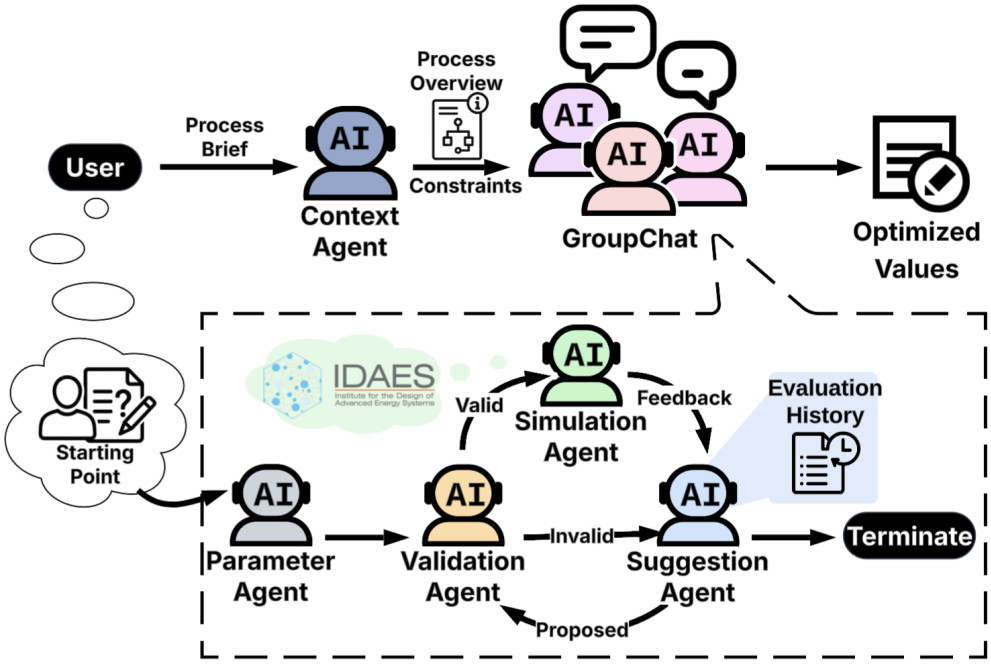


图 1: 基于 LLM 的优化框架架构，展示了代理工作流程。ContextAgent 使用领域知识从基本过程信息生成约束和详细过程概述，而 GroupChat 环境中的专业代理处理参数引入、验证、优化建议以及通过迭代细化直到达到最优解的 IDAES 基础模拟评估。

第一阶段涉及**上下文代理**独立工作，从给定的描述中推断出实际变量范围并生成过程上下文，使用嵌入式的领域知识。该代理为每个决策变量推断约束范围作为下限和上限，并创建一个详细的过程

概述文本，以提供对后续优化代理的上下文理解（由 ContextAgent 生成的详细过程概述见支持信息）。这些生成的约束条件和过程概述构成了随后优化阶段的基础。

第二阶段使用 GroupChat 环境支持四个专门代理在优化过程中进行协调交互和共享内存。通过参数提议、验证、模拟和细化的迭代周期，这些代理共同导航由约束定义的参数空间以达到最优解。

代理角色和功能

在 GroupChat 架构中，每个代理维护专门的职责以协调实现自主优化。优化过程通过特定顺序中的协同代理交互进行操作：

参数代理通过接收用户提供的参数值对来启动过程作为初始起点。这些初步猜测可以是任意的，并且可能违反可行范围，因为用户通常缺乏关于先前由 ContextAgent 生成的约束范围的知识。ValidationAgent 将识别任何约束违规并重定向到 SuggestionAgent 进行纠正。

验证代理担任关键的第一个检查点，评估每个参数集与生成的约束条件，并提供有关所提议值是否可接受的反馈，或者确定哪些特定参数需要调整以避免违反约束。当检测到无效输入时，系统会自动重定向至 SuggestionAgent 进行修订，创建一个验证-修正循环，确保所有参数集保持在可行范围内。

模拟代理在参数通过验证后执行过程评估。该代理运行一个自定义的目标函数，与预定义的基于 Python 的 IDAES 仿真模型接口（详细的 IDAES 实现、热力学属性和动力学参数在支持信息中提供），计算每个参数配置的成本、产量和产率成本比等绩效指标。性能结果报告回 SuggestionAgent，包括任何模拟收敛问题。

建议代理担任优化引擎的角色，保持对优化历史的全面记录。每次迭代后，当前参数值和相应的模拟结果都会被添加到代理的记忆中，直接访问有效边界和历史试验数据。这种积累的知识使代理能够根据观察到的趋势和约束违反情况来改进并提出未来的参数集，完成迭代优化循环。

代理行为通过精心设计的提示来控制，这些提示定义了角色、目标和输出格式。ContextAgent 提示集成了工程领域知识和格式要求，从过程描述中生成现实的操作约束条件，而 SuggestionAgent 提示则实现了基于记忆的优化逻辑以及对约束违规的处理。完整的提示模板在支持信息中提供，展示了将过程工程启发式方法与结构化的输出要求相结合，从而实现自主约束生成和迭代参数精炼的能力。

优化工作流程

迭代优化过程通过一个精确定义的代理交互序列运行，该序列确保系统参数精炼。代理之间的互动通过定制的选择器函数（详细算法见支持信息）进行协调，根据当前发言者和消息内容决定发言顺序，防止冗余通信并确保逻辑工作流程的推进。

核心优化循环遵循以下顺序：ParameterAgent 引入值 →，ValidationAgent 检查可行性 →，SimulationAgent 评估性能 →，SuggestionAgent 分析结果并提出改进措施 →，ValidationAgent 验证新提案。此循环自主重复，直到 SuggestionAgent 基于连续迭代中逐渐减少的性能提升检测到收敛。

此结构化工作流设计解决了自主优化中的三个关键挑战：（1）通过强制验证检查点执行约束，（2）通过记忆引导的建议进行系统性探索，以及（3）通过趋势分析实现自主终止。

与需要数学导数的基于梯度的方法或评估固定离散化的网格搜索不同，此基于代理的方法利用推理智能导航参数空间，根据约束违规和性能反馈调整搜索策略。

该框架的效率源于这种推理引导方法，在 20 至 45 次迭代中即可收敛，相比之下，穷尽方法需要数千次迭代，同时通过系统性约束执行和趋势分析保持解的质量。

框架的自主性通过 SuggestionAgent 根据模拟历史检测到收益递减时自我决定终止来体现。这种方法反映了代理在防止不必要的计算的同时适应不同问题特征的能力，确保了计算效率而不牺牲解的质量。

评估指标

模拟代理使用三个指标来评估过程性能，这些指标捕获了化学过程优化的主要目标：运营成本、产品收率和收率与成本比。

年度运营成本作为主要的经济指标，通过优化公用设施管理（包括运行单元的供暖成本和制冷成本）来最小化能源消耗：

$$\text{Operating Cost} = \text{Heating Cost} + \text{Cooling Cost} \quad [\$/\text{yr}] \tag{1}$$

其中详细的供热和制冷成本计算及其公用设施系数在支持信息中提供。

年度产品产量代表生产效率，通过优化绝对生产率来最大化生产能力：

$$\text{Yield} = F_{\text{benzene,reactor}} \times 3600 \times 24 \times 365 \quad [\text{mol/yr}] \tag{2}$$

产率与成本比率提供了一个综合指标，平衡了这些相互竞争的目标，使得能够在最大化产出和最小化支出之间评估权衡：

$$\text{Yield-to-Cost Ratio} = \frac{\text{Annual Yield}}{\text{Operating Cost}} \quad [\text{mol}/\$] \tag{3}$$

这些指标被选中以涵盖工业优化的重点领域：经济表现（成本）、生产效率（产量）以及整体过程

效率（产量与成本比率），允许在多个性能维度上直接对比传统优化方法。

结果与讨论

案例研究验证

为了验证我们多智能体框架的约束生成和优化能力，我们使用 HDA 过程来展示其性能，这展示了我们的方法解决的受限优化挑战。该过程通过连接单元操作将甲苯转化为苯，但如图 2所示，仅提供了基本信息：进料条件、转化率和设备配置。没有指定操作温度范围、压力限制或可行参数边界。这个过程代表了一个现实的工业场景，在这种情况下详细的操作约束通常不可用，使其成为自主约束生成的理想测试案例。当尝试将传统方法应用于此过程时，优化挑战变得明显。

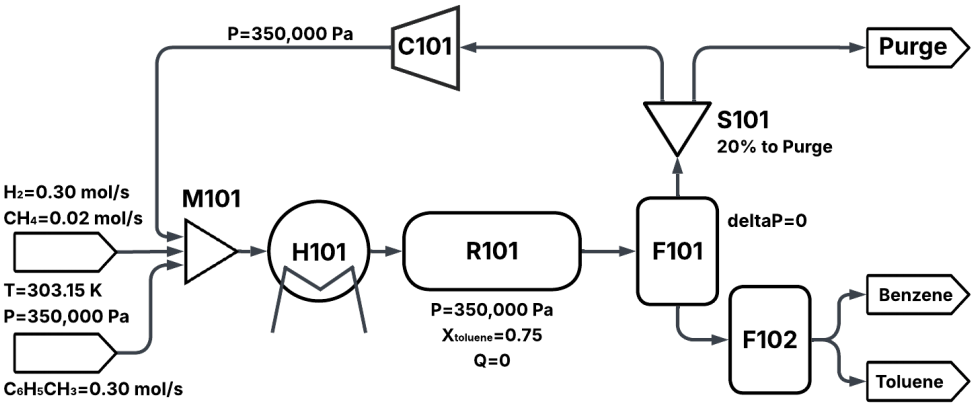


图 2: 流表用于 IDAES 模拟和优化研究的脱烷基化（HDA）过程。该过程包括混合器 M101、加热器 H101、反应器 R101、闪蒸分离器 F101 和 F102、分离器 S101 以及压缩机 C101。进料流包括氢气（0.30 mol/s）、甲烷（0.02 mol/s）和甲苯（0.30 mol/s），温度为 303.15 K，压力为 350,000 Pa。反应器在 350,000 Pa 的压力下运行，并将甲苯转化率达到 75%，随后的分离单元生产出苯和甲苯的产品流，其中 20%的顶部流通过分离器 S101 排出。

传统优化方法如 IPOPT 或网格搜索无法在没有显式约束集的情况下直接应用。这一限制展示了我们的 ContextAgent 的实际需求，它使用嵌入的过程知识推断变量边界。鉴于这种约束生成能力，我们将优化问题表述如下：

$$\text{optimize } \phi(x) \quad \text{subject to} \quad x_i^L \leq x_i \leq x_i^U, \quad \forall i = 1, \ldots, 4 \tag{4}$$

其中 $\phi(x)$ 表示由 IDAES 模拟评估的目标函数（例如产量、成本或产率与成本比），而 x^L 和 x^U 是从 LLM 生成的领域知识中得出的变量边界。向量 $x = (x_1, x_2, x_3, x_4)$ 代表四个关键决策变量：H101 出口温度、F101 出口温度、F102 出口温度以及 F102 压降。IDAES 模型的实现使用理想气液平衡关系进行

热力学计算。反应器以固定的 75%甲苯转化率运行，采用化学计量反应约束（完整的工艺描述、进料规格和目标函数定义在支持信息中提供）。

约束生成质量

为了评估自主约束生成的可靠性和一致性，使用相同的提示在五个独立试验中对 ContextAgent 的约束生成能力进行了评估，以确保具有代表性的约束生成并尽量减少随机方差。表 1展示了各个试验的结果以及关键单元操作 H101、F101 和 F102 的操作条件下推断出的平均约束范围。最右边一列显示的平均范围用于后续优化性能比较。

表 1: 上下文代理约束生成结果及五次试验的平均范围

单位变量	试验 1	试验 2	试验 3	试验 4	试验 5	平均范围
H101 Temp (K)	[870, 950]	[873, 973]	[820, 980]	[800, 950]	[773, 1023]	[827.2, 975.2]
F101 Temp (K)	[310, 330]	[300, 340]	[290, 390]	[330, 420]	[298, 368]	[305.6, 369.6]
F102 Temp (K)	[300, 340]	[330, 380]	[280, 340]	[300, 360]	[323, 413]	[306.6, 366.6]
F102 ΔP (Pa)	[-200000, -50000]	[-200000, -50000]	[-240000, -10000]	[-200000, -10000]	[-220000, -20000]	[-216000, -28000]

正如预期，由于 LLM 响应的随机性，各次试验中的约束生成显示出变异性，详情见支持信息。对于关键 H101 反应器温度，边界范围在试验中从 [773, 1023] K 变化到 [870, 950] K，范围宽度从 80 K 变化到 250 K。闪蒸分离器温度也显示出类似的变异性，F101 的边界范围为 [290, 390] K 至 [330, 420] K，F102 的边界范围为 [280, 340] K 到 [323, 413] K。压力降约束则更加一致，所有试验生成了相似的下限（-240000 至-200000 Pa），但上限变化（-10000 至-50000 Pa）。虽然这种变异性可能一开始看起来令人担忧，但实际上它展示了框架强大的约束生成能力。关键见解不是完美的一致性，而是始终能够生成可行的工业界限，这使得优化能够在传统方法由于缺少约束信息而无法应用时继续进行。

重要的是，尽管在不同试验中存在这种可观察的变化，ContextAgent 一致生成了现实且符合工程要求的边界。对于关键的 H101 反应器温度，生成的平均范围为 [827.2, 975.2] K，与文献报道的工业实践中 [773, 977] K[8,25] 范围有很好的对齐。尽管在文献中不易获得闪蒸分离器温度和压力降的具体工业范围，代理生成的反应器约束与既定工业实践之间的紧密对应表明了该代理利用仅有的基本工艺信息和嵌入领域的知识提取符合现场操作条件的能力。

性能比较

优化结果：为了评估我们方法的有效性，我们将 LLM 框架的优化性能与 IPOPT（详细算法描述见支持信息）和使用平均 ContextAgent 生成约束条件的网格搜索基准进行了比较。网格搜索在每个变量上实现了 10 个离散点，总共有 10,000 种参数组合（10Œ10Œ10Œ10）跨四个决策变量进行评估。表 2 总结了根据之前定义的三个评价指标的优化结果。

表 2: IPOPT、LLM 和网格搜索优化结果的比较。

度量	方法	H101 Temp (K)	F101 Temp (K)	F102 Temp (K)	ΔP F102 (Pa)	值
Cost ($/yr)[a]	LLM	830.0	325.0	360.0	-180000	5.792×10^5
	IPOPT	827.2	305.6	306.6	-28314	5.660×10^5
	Grid Search	827.2	305.6	306.6	-48444	5.660×10^5
Yield (mol/yr)[b]	LLM	975.0	369.5	366.0	-180000	1.644×10^7
	IPOPT	857.5	369.6	344.4	-130340	1.565×10^7
	Grid Search	958.8	369.6	339.9	-89333	1.750×10^7
Yield/Cost (摩尔/量子)[b]	LLM	827.2	369.0	306.7	-28000	17.23
	IPOPT	827.2	369.6	306.6	-121039	17.62
	Grid Search	827.2	369.6	306.6	-68889	17.62

[a] 越低越好。[b] 较高越好。

性能分析：上述结果显示在所有三项指标中具有竞争力的表现。网格搜索始终能够找到最佳解决方案，得益于对离散参数空间的全面评估。然而，基于 LLM 的优化框架相比于传统方法表现出了远优越的优化性能。表 3a显示了每种优化方法跨不同指标所需的迭代次数。

表 3: 优化性能比较

(a) 收敛速度（迭代至收敛的次数）

度量	大语言模型	IPOPT
Cost ($/yr)	21	35
Yield (mol/yr)	26	81
Yield/Cost (摩尔/米^3)	43	46

(b) 优化 墙钟时间（小时）

度量	大语言模型	网格搜索
Cost ($/yr)	0.17	10.50
Yield (mol/yr)	0.20	10.50
Yield/Cost (摩尔/千克水)	0.33	10.50

LLM 框架始终需要的迭代次数少于 IPOPT。尤为显著的是，在产量优化方面，IPOPT 需要 81 次迭代，而 LLM 方法仅需 26 次迭代，减少了三倍。与穷举方法相比，时间效率的提高更为显著。表 3b 显示了 LLM 框架在所有指标下的优化时间不到 20 分钟，相比之下，网格搜索固定需要 10.5 小时，速度提高了三十倍。由于设计原因，网格搜索必须评估所有 10,000 种参数组合，而通过智能搜索策略调整，LLM 框架在大约 20-45 次迭代内收敛。

尽管需要显著较少的迭代和计算时间，LLM 框架实现了具有竞争力的解决方案质量。IPOPT 在成本最小化和产量与成本比率优化方面略胜一筹，而 LLM 框架在产量最大化方面优于 IPOPT。这些微小的性能差异证明了该框架对于约束限制优化问题的可行性，特别是考虑到其显著的计算优势。框架平衡解决方案质量和计算效率的能力使其特别适用于需要快速优化或计算资源有限的情景。

所有计算基准测试均在配备 Apple M1 芯片（8 核，16GB 内存）的 MacBook Pro 上进行。

推理引导的参数探索：为了考察代理的内部决策逻辑，我们进行了一个独立的额外试验，并放宽了冗长程度的限制。为了最小化成本，代理展现了如下所示的复杂过程理解能力：

> 保持反应器负荷在允许范围内——接近最低的 *H101* 温度 ($\approx 830K$) 可以减少炉子工作量和燃料使用。将两个闪蒸罐推向其范围内的较暖端；更暖的冷凝器意味着需要较少的制冷/

盐水工作量。放松第二次闪蒸的压力降至机械最小值 *(–30* 千帕*)*。节流在本质上是“免费”的，因此选择最小的 ΔP 可以避免产生过多的蒸汽，否则这些蒸汽将不得不在下游被再压缩或冷凝。

这一推理展示了三种关键能力：正确识别加热和冷却公用设施之间的权衡，理解操作条件对能耗的影响，以及应用领域相关的启发式方法进行参数选择。代理在调整参数时能够阐述过程工程原理的能力使这种方法区别于传统的数值优化器。虽然这种局部成本最小化策略可能不会直接导致全局最优解，但它提供了可以通过后续迭代进一步精炼的有效搜索方向。

表 4展示了代理的推理如何转化为成本最小化过程中的系统参数调整。优化过程始于用户任意提供的初始猜测值，这些值可能会违反自主生成的约束条件。如图所示，初始 H101 温度为 600 K，这违反了平均生成范围 [827.2, 975.2] K，导致状态显示“无效”，直到验证代理-建议代理循环在第 1 次迭代时产生第一个可行解。进展揭示了代理系统地测试不同操作策略并根据性能反馈进行调整的能力。早期参数探索显示出不同的成本表现，因为代理正在测试不同的操作策略，而后续的迭代则展示了框架向更具有成本效益的操作条件收敛的趋势。到第 10 次迭代时，代理识别出最佳反应器温度为 830 K，这显著降低了运行成本，随后的迭代主要集中在微调次要变量，同时保持该温度设定。

表 4: 系统参数探索在运营成本最小化过程中的应用。

迭代	**H101 Temp (K)**	**F101 Temp (K)**	**F102 Temp (K)**	**F102 ΔP (Pa)**	**成本 ($/年)**
User Initial Guess	600.0	325.0	375.0	-240000	Invalid
1	840.0	325.0	360.0	-180000	601715
2	830.0	330.0	363.0	-160000	617356
6	840.0	330.0	357.0	-175000	632627
10	830.0	325.0	360.0	-180000	579186
⋮	⋮	⋮	⋮	⋮	⋮

方法论优势：大型语言模型方法不要求目标函数可微或连续，这可能使其适用于具有不连续或不可微目标的优化问题。当考虑问题维度时，其扩展性优势尤为明显。网格搜索随着决策变量数量的增长计算需求呈指数级增加，在我们的四变量实现中，即使每个变量只使用了 10 个离散点，也需要超过 10 小时的实际运行时间。虽然大型语言模型框架在问题复杂度上的可扩展性仍有待验证，但其当前效率表明它可能在需要智能引导参数探索的复杂优化场景中具有潜在优势。

结论

本工作证明了大型语言模型代理可以自主解决限制传统优化方法的约束定义瓶颈。通过从最小的过程描述生成工程适用的操作范围并在这些范围内迭代优化，我们的多代理框架能够在之前需要大量

领域专业知识或手动指定约束的情景中实现过程优化。

框架的主要创新在于自主约束生成与协作多智能体优化的结合。通过利用嵌入式领域知识来为单元操作生成工程适用的操作界限，该系统能够在传统方法无法直接应用的情况下实现优化。结合处理非可微目标函数的能力，这一功能使得在传统基于梯度的方法可能遇到困难的地方进行优化成为可能。在 HDA 过程中的验证显示，在多个目标下与 IPOPT 和网格搜索相比具有竞争力的性能，并且通过在不到 20 分钟内达到收敛，显著提高了计算效率，而相比之下网格搜索需要超过 10 小时。值得注意的是，框架的推理能力使其区别于传统的优化器，智能体通过明确识别权衡并采用基于领域的参数选择策略展现了对过程的深刻理解。

本研究为自主过程优化奠定了基础，这可能显著降低专业知识壁垒，并促进高级优化技术的广泛应用。在 HDA 工艺中展示的可行性证明了多代理方法的概念验证，开启了向自主过程建模和流程图合成扩展功能的道路。进步的关键机遇包括与工艺数据库集成以生成更好的约束条件，以及开发结合大语言模型推理与传统优化方法的方法，以提高稳健性。

代码可用性

本研究中使用的所有代码均可在以下位置公开获取：

`https://github.com/tongzeng24/ProcessAgent`

Supporting Information Available

支持信息包括多代理算法规范、完整的代理提示模板、ContextAgent 过程理解演示、约束生成方法比较，以及详细的 HDA 过程实现，包括饲料规格、具有效用成本系数的经济目标函数公式化、操作约束和 IDAES 建模细节。

作者贡献

T.Z., S.B., J.O., C.L 和 A.B.F. 提出了该项目。T.Z. 开发了核心多智能体优化框架，并领导了手稿的准备。S.B. 为约束生成模块的发展做出了贡献，改进了核心优化算法，并协助撰写手稿。J.O. 在整体框架和优化策略的设计上提供了关键的概念输入。C.L. 贡献于基于 IDAES 模拟的实现和验证。所有作者审阅并批准了最终的手稿。A.B.F. 监督了该项目。

References

(1) Biegler, L. T. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*; SIAM: Philadelphia, 2010.

(2) Wächter, A.; Biegler, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* **2006**, *106*, 25–57.

(3) Gill, P. E.; Murray, W.; Saunders, M. A. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review* **2005**, *47*, 99–131.

(4) Biegler, L. T.; Grossmann, I. E. Retrospective on optimization. *Computers & Chemical Engineering* **2004**, *28*, 1169–1192.

(5) Gross, B.; Roosen, P. Total process optimization in chemical engineering with evolutionary algorithms. *Computers & Chemical Engineering* **1998**, *22*, S229–S236.

(6) Rios, L. M.; Sahinidis, N. V. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization* **2013**, *56*, 1247–1293.

(7) Grossmann, I. Challenges in the Application of Mathematical Programming in the Enterprise-wide Optimization of Process Industries. *Theoretical Foundations of Chemical Engineering* **2014**, *48*, 555–573.

(8) Douglas, J. M. *Conceptual Design of Chemical Processes*; McGraw-Hill: New York, 1988.

(9) Himmelblau, D. M.; Riggs, J. B. *Basic Principles and Calculations in Chemical Engineering*, subsequent ed.; Prentice Hall: Upper Saddle River, NJ, 2005.

(10) Lee, A.; Ghouse, J. H.; Eslick, J. C.; Laird, C. D.; Siirola, J. D.; Zamarripa, M. A.; Gunter, D.; Shinn, J. H.; Dowling, A. W.; Bhattacharyya, D.; Biegler, L. T.; Burgard, A. P.; Miller, D. C. The IDAES process modeling framework and model library—Flexibility for process simulation and optimization. *Journal of Advanced Manufacturing and Processing* **2021**, *3*, e10095.

(11) Bynum, M. L.; Hackebeil, G. A.; Hart, W. E.; Laird, C. D.; Nicholson, B. L.; Siirola, J. D.; Watson, J.-P.; Woodruff, D. L. *Pyomo – Optimization Modeling in Python*, 3rd ed.; Springer Optimization and Its Applications; Springer Cham, 2021.

(12) Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; others Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* **2022**, *35*, 24824–24837.

(13) Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; others Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*, 1877–1901.

(14) Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; others Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* **2022**, *35*, 27730–27744.

(15) Lu, C.; Lu, C.; Lange, R. T.; Foerster, J.; Clune, J.; Ha, D. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. 2024; `https://arxiv.org/abs/2408.06292`.

(16) Bran, A. M.; Neukomm, T. A.; Armstrong, D.; Jonev, Z.; Schwaller, P. Chemical reasoning in LLMs unlocks steerable synthesis planning and reaction mechanism elucidation. 2025; `https://arxiv.org/abs/2503.08537`, arXiv:2503.08537.

(17) Bran, M.; Cox, A.; Schilter, S.; others Augmenting large language models with chemistry tools. *Nature Machine Intelligence* **2024**, *6*, 525–535.

(18) Guo, X.; Keivan, D.; Syed, U.; Qin, L.; Zhang, H.; Dullerud, G.; Seiler, P.; Hu, B. ControlAgent: Automating Control System Design via Novel Integration of LLM Agents and Domain Expertise. 2024; `https://arxiv.org/abs/2410.19811`.

(19) Vyas, J.; Mercangöz, M. Autonomous Industrial Control using an Agentic Framework with Large Language Models. 2024; `https://arxiv.org/abs/2411.05904`.

(20) Schweidtmann, A. Generative artificial intelligence in chemical engineering. *Nature Chemical Engineering* **2024**, *1*, 193.

(21) Balhorn, L.; Caballero, M.; Schweidtmann, A. Toward autocorrection of chemical process flowsheets using large language models. Proceedings of the 34th European Symposium on Computer Aided Process Engineering. 2024; pp 3109–3114.

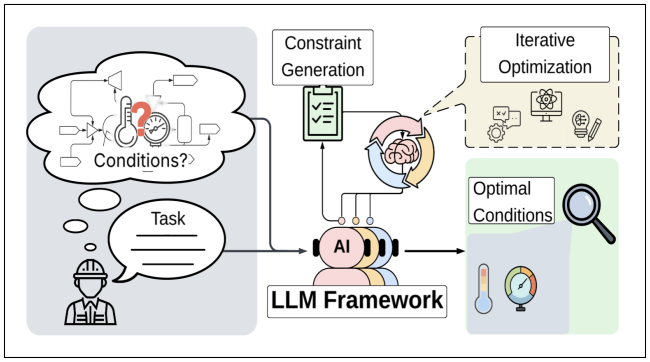
(22) Hirtreiter, E.; Schulze Balhorn, L.; Schweidtmann, A. M. Toward automatic generation of control structures for process flow diagrams with large language models. *AIChE Journal* **2024**, *70*, e18259.

(23) Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E. E.; Jiang, L.; Zhang, X.; Zhang, S.; Awadallah, A.; White, R. W.; Burger, D.; Wang, C. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. COLM 2024. 2024.

(24) OpenAI O3 and O4-mini System Card. `https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf`, 2025; Large language model.

(25) Eldood, A. A. M. Selective Steam Dealkylation of Alkyl Aromatic Hydrocarbons. M.Sc. thesis, University of Calgary, Calgary, Alberta, Canada, 2008; Master's thesis.

TOC Graphic


Conditions?
Task
Constraint Generation
Iterative Optimization
AI
LLM Framework
Optimal Conditions