

硕士学位论文

基于状态转移算法的一类动态优化方法
研究

State Transition Algorithm based Approach for a Class of
Dynamic Optimization Problems

学科专业 控制科学与工程

研究方向 智能控制与优化决策

作者姓名 黄淼

指导教师 周晓君 副教授

中南大学

2019年5月

中图分类号 TP18

学校代码 10533

UDC 621.3

学位类别 学术学位

硕士学位论文

基于状态转移算法的一类动态优化方法 研究

State Transition Algorithm based Approach for a Class of
Dynamic Optimization Problems

作者姓名： 黄淼

学科专业： 控制科学与工程

学科方向： 控制理论与控制工程

研究方向： 智能控制与优化决策

二级培养单位： 自动化学院

指导教师： 周晓君 副教授

论文答辩日期_____

答辩委员会主席_____

中南大学

2019年5月

学位论文原创性声明

本人郑重声明，所呈交的学位论文是本人在指导教师指导下进行的研究工作及取得的研究成果。尽我所知，除了论文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中南大学或其他教育机构的学位或证书而使用过的材料。与我共同工作的同志对本研究所作的贡献均已在论文中作了明确的说明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

作者签名：_____ 日期：_____年___月___日

学位论文授权使用授权书

本学位论文作者和指导教师完全了解中南大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交学位论文的复印件和电子版；本人允许本学位论文被查阅和借阅；学校可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用复印、缩印或其它手段保存和汇编本学位论文。

保密论文待解密后适应本声明。

作者签名：_____

指导教师签名_____

日期：_____年___月___日

日期：_____年___月___日

基于状态转移算法的一类动态优化方法研究

摘要: 工业生产过程通常是一个随时间连续变化的动态过程, 动态优化作为优化控制技术能有效处理其动态特性, 同时实现工业动态过程稳定运行、节能降耗与挖潜增效。由于实际工业动态优化问题通常具有非线性、多模态等特点, 传统基于梯度优化的动态优化方法易陷入局部极值, 无法找到最优操作方案。状态转移算法 (STA) 作为一种新型启发式智能优化算法, 具有结构简单、全局搜索能力性强等优点, 在处理非线性多模态问题上具有潜在优势。本文以状态转移算法为基础, 针对一类工业过程常见的优化时长固定、有约束的非线性动态优化问题求解展开研究。具体内容如下:

1) 针对如何求解非凸非线性动态优化问题的难点, 采用均匀划分控制向量参数化, 引入并改进了状态转移算法, 提出了基于变尺度状态转移算法的均匀划分动态优化方法 (udCVP-vSTA)。经典非线性动态优化问题测试表明了该方法的有效性。

2) 针对均匀划分下难以高效的获得高精度近似最优控制曲线, 提出部分网格移动的非均匀划分控制参数化策略, 并提出了加强了局部收敛能力的混合状态转移算法 (HSTA), 在此基础上, 提出了基于 HSTA 的非均匀划分动态优化方法 (ndCVP-HSTA)。实验表明了该方法较均匀分段能高效获得高精度控制曲线。

3) 针对动态优化约束处理难点, 在非均匀划分无约束动态优化方法的研究基础上, 提出了基于光滑化罚方法的约束动态优化方法。经典动态优化问题的实验测试结果表明了该约束动态优化方法的有效性。

4) 最后, 本文以湿法炼锌除铜过程优化控制为例开展动态优化应用研究, 分析其动态特性、构建动态优化问题并求解, 实验仿真结果表明本文所提除铜过程动态优化可以达到除铜过程工况稳定、节能降耗的目标。

论文共计图 38 幅, 表 18 个, 参考文献 98 篇。

关键词: 动态优化; 状态转移算法; 自适应控制参数化; 除铜过程
分类号: TP18

State Transition Algorithm based Approach for a Class of Dynamic Optimization Problems

Abstract: Dynamic optimization aims to optimize a predefined performance index for reducing production costs, improving product quality, meeting safety requirements and environmental regulations over a time interval. Since dynamic optimization problems (DOPs) arising from practical industrials are usually nonlinear, multimodal with both state and control constraints, an efficient and robust numerical dynamic optimization method is the key to realize dynamic optimization of complex industrial processes. As a heuristic intelligent optimization algorithm, state transition algorithm (STA) has advantages in dealing with DOPs because of its strong global search ability. In this paper, the dynamic optimization method based on STA is studied. The main contributions of this paper are as follows:

1) In order to solve the nonconvex DOPs which are highly nonlinear, multimodal and multidimensional, a dynamic optimization method, named udCVP-vSTA, is proposed based on the improved STA and uniform discretization-based control vector parameterization (CVP) method. Numerical experiments are provided to illustrate the effectiveness of the proposed method.

2) In order to obtain higher precision approximate optimal control curve efficiently, a dynamic optimization method, named ndCVP-HSTA, is proposed based on the un-uniform discretization-based CVP method and the proposed hybrid STA. Results show that the proposed method can not only find a better control parameterization grid, but also promotes computational efficiency.

3) In order to deal with constraints on states variable, a constrained dynamic optimization method, based on ndCVP-HSTA and smooth penalty function, is proposed. Experimental results show the effectiveness of the proposed method.

4) A dynamic optimization problem is constructed for copper removal process (CRP) to find an optimal control trajectory of zinc addition for a

given time, with considerations of production constraints. Results show that the proposed constrained dynamic optimization method can produce not only the optimal control trajectory with a qualified outlet ion concentration, but also with less zinc consumption.

There are 38 figures, 18 tables, 98 references.

Keywords: Dynamic optimization; state transition algorithm; adaptive control parameterization method; copper removal process

Classification: TP18

目 录

1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 一类动态优化问题描述.....	3
1.3 动态优化方法研究现状.....	4
1.3.1 间接动态优化方法.....	4
1.3.2 直接动态优化方法.....	5
1.4 论文研究内容与结构安排.....	9
2 基于变尺度状态转移算法的均匀划分动态优化方法.....	12
2.1 均匀划分控制向量参数化策略.....	12
2.1.1 控制向量参数化.....	12
2.1.2 均匀划分控制向量参数化.....	13
2.2 状态转移算法简介.....	15
2.2.1 状态转移算法原理.....	15
2.2.2 状态转移算法流程.....	16
2.3 变尺度状态转移算法.....	17
2.3.1 状态转移算子性能分析.....	17
2.3.2 搜索尺度自适应策略.....	20
2.3.3 变尺度状态转移算法步骤.....	21
2.4 基于变尺度状态转移算法的均匀划分动态优化方法.....	22
2.5 实验结果与讨论.....	23
2.6 本章小结.....	30
3 基于混合状态转移算法的非均匀划分动态优化方法.....	32
3.1 非均匀划分控制向量参数化策略.....	32
3.1.1 网格点移动与增删策略分析.....	33
3.1.2 基于部分移动网格优化的非均匀划分策略.....	36
3.2 混合状态转移算法.....	37
3.2.1 梯度搜索算法.....	37
3.2.2 混合状态转移算法.....	38
3.3 基于混合状态转移算法的非均匀划分动态优化方法.....	40
3.4 实验结果与讨论.....	42
3.5 本章小结.....	46

4 约束动态优化及在除铜过程优化控制中的应用.....	47
4.1 湿法炼锌除铜过程动态优化问题.....	47
4.1.1 除铜过程动态特性描述.....	47
4.1.2 除铜过程生产约束分析.....	49
4.1.3 除铜过程动态优化问题.....	50
4.2 光滑约束动态优化方法.....	52
4.2.1 一般约束动态优化问题.....	52
4.2.2 状态相关约束处理.....	54
4.2.3 基于 ndCVP-HSTA 的光滑化约束动态优化方法	56
4.3 实验结果与讨论.....	58
4.3.1 经典约束动态优化问题.....	58
4.3.2 除铜过程动态优化问题.....	61
4.4 本章小结.....	63
5 结论与展望.....	64
5.1 研究工作总结.....	64
5.2 后续工作展望.....	65
参考文献.....	66
攻读学位期间主要的研究成果.....	74
致谢.....	75

1 绪论

流程工业是制造业重要组成部分,是我国国民经济的支柱,也是我国经济持续增长、国家安全、国力强盛的重要支撑力量^[1]。我国流程工业整体实力较强,但仍面临着巨大的发展挑战^[2]。特别地,在优质资源枯竭、环境恶化及能源短缺的巨大压力下,我国工业生产陷入资源利用效率低、信息化水平不高、质量达标率低以及经济效益整体低迷等发展瓶颈,实现运行、资源和能效的三维全局优化任务紧迫而艰巨^[3]。

动态优化,作为优化技术的重要分支之一,旨在针对工业动态系统,获得其优化时间段内的最优操作轨迹,使得生产过程在安全开展的同时效益最大化,是实现生产过程稳定运行、节能降耗与挖潜增效的三维全局优化的有效手段^[4]。由于实际工业动态优化问题通常具有高维度、强非线性、时变、多约束等复杂特性,高效、高精度的动态优化方法也成为了诸多学者的研究重点。

1.1 研究背景及意义

动态优化(Dynamic optimization, DO)在工业生产过程中的重要实际应用研究意义。动态优化的典型应用领域包括:1)间歇及半间歇过程的最优控制轨迹的确定;2)连续过程的最优控制等。工业过程的动态特性既源于化学反应本身存在的动态平衡体系,也源于其实际生产过程中难以避免的操作条件的变化。一方面,从原料入口到产品出口,工业生产过程是一系列物质能量随时间连续转化与传递的物理化学反应过程,可进一步地用代数微分方程表征此动态平衡系统。另一方面,实际生产过程总是动态的,其控制变量、系统状态变量总是随时间持续改变。

动态优化问题的有效求解是动态优化从理论到应用的关键。其求解方法可划分为两类,一类为传统基于变分法的解析法^{[5][6]},此方法能求得解析解,但其需要针对每个问题进行数学推导,不利于计算机实现,相对通用性不强,在实际应用中存在不便。面对工程中的复杂动态优化问题,学者们研究了更为通用的直接动态优化方法(也称数值法)来进行求解^[7]。其中心思想为将随时间连续的无限维的优化问题用离散的方法近似为有限维优化非线性规划问题子问题(Non-linear programming, NLP),然后采用优化算法求取近似NLP问题的解,最终以分段离散的曲线逼近最优控制轨线。常用的离散策略为全部参数化和控制参数化(Control vector parameterization, CVP)方法,其中部分离散策略又拥有相较小规模因而更好求解的子问题。因此,近年来基于部分离散的动态优化问题数值解法有较为广泛研究。

动态优化方法的核心是求解其离散后的非凸 NLP 子问题，状态转移算法作为进化算法在求解此类问题上具有潜在优势。许多动态优化方法在求解离散后形成的非线性规划问题子问题时，常依赖于梯度信息进行优化，尽管其具有收敛速度快，收敛精度高的特点，当待求解动态优化问题为多峰值类型时，该类算法常常收敛到局部最优解，优化效果不佳。近年来，智能进化算法在解决各类工程实际问题上取得了较好的成效，该类算法不需要计算梯度信息，不依赖于初始点，适应性广泛，全局搜索能力较强，在处理多峰值、不连续、不含显示描述和复杂多目标等特性动态优化问题上有着独特的优势。状态转移算法（State transition algorithm, STA）是中南大学周晓君等^[8]提出的一种新型的智能型全局优化算法，该优化方法以状态空间模型为统一框架来产生候选解，并设计了包含全局与局部搜索的四种特殊的状态变换算子，因此在求解优化问题上有优良的收敛性和全局搜索能力。近年来 STA 在多个应用中展现了极佳的求解能力^{[9][10]}，在复杂问题上比粒子群算法、差分算法等经典进化算法相比有着更高的求解精度和效率^{[11][12][13][14]}，在求解非凸动态优化问题上具有潜在优势。

动态优化方法求解精度和求解效率的共同实现一直是国内外研究的前沿和难点。一方面，针对我国工业生产过程复杂多变、工况实时性要求强等特点，动态优化方法需要高效、可靠、鲁棒性强的处理具有不同系统结构的、高非线性的、多峰值的工业过程动态优化问题。另一方面，离散数值解法虽然能直接获取动态优化问题的最优解，得到相应的最优控制轨迹，但囿于其先离散后逐个计算的求解架构，其问题求解精度和求解效率两者互为矛盾，提高求解精度势必会带来需要优化的变量个数的增加，而优化变量维数的增加又定会增加优化时间和优化难度，考验优化算法优化性能。在传统的数值解法中，较为普遍的采用均匀分段方案来离散化控制网格，网格划分的疏密即直接影响了 NLP 子问题维数和非线性程度，又直接决定最终近似数值解结果逼近最优控制曲线的程度。因此，如何处理其逼近精度和计算效率之间的矛盾具有较高研究价值。

本文针对一类优化时长固定，有约束的工业动态优化问题求解方法研究。1) 针对实际工业动态优化问题的复杂性、非凸性，本文首先引入均匀划分控制参数化处理其动态特性，随后引入并改进状态转移算法，提出了参数自适应策略，提出了基于变尺度状态转移算法的均匀划分动态优化方法。2) 针对均匀划分导致的精度与计算效率的矛盾，提出了部分移动非均匀划分控制参数化方案，设计了混合状态转移算法，提出了基于混合梯度的状态转移算法的非均匀划分动态优化方法，最终能高效、准确和快速求解各类动态优化问题。3) 在此基础上，考虑到实际动态优化问题中的状态约束、控制约束以及混合状态控制约束等特点，建立了引入光滑化罚函数法，提出了基于光滑化约束动态优化方法。4) 最后，

实现了其在有色冶金过程湿法炼锌除铜过程中的应用研究,对工业生产的稳定高效高质具有重要指导意义。

1.2 一类动态优化问题描述

动态优化为减少生产消耗、满足安全要求、符合绿色生产、提升产品质量、提高生产稳定性提供了方法框架。动态优化包括离线应用,如最优控制,轨迹规划,非线性参数估计和优化反应堆和批处理过程,以及在线应用,如非线性模型预测控制,非线性状态估计和动态实时优化等。动态优化通常是基于动态模型的优化控制手段,其动态模型一般由微分代数方程(Differential algebraic equation, DAE)来表述,用来描述系统的动态特性,如质量、能量或动量的随时间或空间变化与守恒关系,如式(1-1)所示:

$$\begin{cases} \dot{\mathbf{x}}(t) = f[\mathbf{u}(t), \mathbf{x}(t), t] \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \quad (1-1)$$

其中 $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ 为 n_x 维状态变量,系统的初始状态为 \mathbf{x}_0 , $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ 为 n_u 维控制变量,即有 n_u 条控制曲线需要求解。

动态优化旨在优化一段时间内动态模型中的控制变量(操作变量),使得整个生产过程性能指标达到最优,同时需满足除了系统 DAE 内在约束外,由执行机构限制所带来的控制变量约束及由操作安全所带来的状态约束。本文仅考虑一类常见的单目标、优化时长一定、带约束的动态优化问题,其数学模型描述如下:

$$\begin{aligned} \min_{\mathbf{u}(t)} \quad & J = \Phi[\mathbf{x}(t_f)] + \int_{t_0}^{t_f} L[t, \mathbf{x}(t), \mathbf{u}(t)] dt \\ \text{s.t.} \quad & \begin{cases} \dot{\mathbf{x}}(t) = f[\mathbf{u}(t), \mathbf{x}(t), t] \\ \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{g}[\mathbf{u}(t), \mathbf{x}(t), t] \leq 0 \\ \mathbf{u}' \leq \mathbf{u}(t) \leq \mathbf{u}'' \\ t_0 \leq t \leq t_f \end{cases} \end{aligned} \quad (1-2)$$

其中,目标函数 J 为预定的性能指标,其包括了在终端时刻的终端性能指标 $\Phi[\mathbf{x}(t_f)]$ 和过程性能指标 $\int_{t_0}^{t_f} L[t, \mathbf{x}(t), \mathbf{u}(t)] dt$ (如时间最优、燃料最少、产量最优等)。控制变量必须满足 $\mathbf{u}' \leq \mathbf{u}(t) \leq \mathbf{u}''$ 。状态约束表述为 \mathbf{g} 。 $[t_0, t_f]$ 为优化控制时间区间。

控制变量 $\mathbf{u}(t)$ 是时间 t 的函数。状态变量 $\mathbf{x}(t)$ 由微分方程(1-1)随时间发生运动状态改变。控制变量 $\mathbf{u}(t)$ 即可直接影响目标函数,也间接通过系统代数方程微分方程对状态 $\mathbf{x}(t)$ 产生影响,进而间接影响目标函数。综上,动态优化旨在找

到一段时间内的最优控制变量轨迹 $u^*(t), t \in [t_0, t_f]$ ，使得预定性能指标 J 最优，且同时满足控制变量约束与状态变量约束 g 。

1.3 动态优化方法研究现状

动态优化方法是一种基于系统动态模型的优化控制方法，需要在无限维度上解决随时间连续的最优控制和相关问题。动态优化方法主要可分为间接动态优化方法、直接动态优化方法及智能动态优化方法等。

1.3.1 间接动态优化方法

最初提出的动态优化问题具有良好的解析性质，如最速下降线问题、最小面积问题等，通常，通过古典的变分法即可得到其最优解析解^[5]。变分法基于庞特里亚金最大值原理，引入 Hamilton 函数，根据一阶最优必要性条件得到两点边值问题，通过求解两点边值问题即能得到原动态优化问题严密而精确的解。如，仅考虑以上无约束的动态优化问题标准形式，首先引入 Hamilton 函数：

$$H[t, x(t), \lambda(t), u(t)] = L[t, x(t), u(t)] + \lambda(t)^T [f(t, x(t), u(t))] \quad (1-3)$$

式中 $\lambda(t)$ 是待定的 n_λ 维向量函数，现假定 $x^*(t), u^*(t)$ 为原问题解析解，则根据问题最优性必要条件可得，对于无不等式约束问题，最优性条件可表示为一组微分代数方程组，Hamilton 函数与 $x^*(t), u^*(t)$ 需要满足如下条件：

$$\begin{aligned} \dot{x}(t) &= \frac{\partial H[t, x^*(t), \lambda(t), u^*(t)]}{\partial \lambda(t)} = f[t, x^*(t), u^*(t)] \\ x^*(t_0) &= x_0 \\ \dot{\lambda}(t) &= -\frac{\partial H[t, x^*(t), \lambda(t), u^*(t)]}{\partial x(t)} \\ \lambda(t_f) &= \left. \frac{\partial \Phi[x^*(t), t]}{\partial x(t)} \right|_{t=t_f} \\ \frac{\partial H[t, x^*(t), \lambda(t), u^*(t)]}{\partial u(t)} &\equiv 0, t \in [t_0, t_f] \end{aligned} \quad (1-4)$$

式中，一般称 $\lambda(t) = (\lambda_1(t), \dots, \lambda_n(t))^T$ 为共轭或伴随向量函数。如果函数 $f(t, x(t), u(t))$ 关于系统状态变量 $x(t)$ 和控制变量 $u(t)$ 连续且二次连续可微，且 $L[t, x(t), u(t)]$ 关于所有自变量连续且二次连续可微，则可通过求解上述方程组来得到原动态优化问题的解析解，这种先找到最优解需满足的条件然后求解的方法称之为间接解法。

在 1995 年, Richard^[5]如何基于极大值原理处理具有纯状态变量约束和混合状态变量约束的最优控制问题开展研究(纯状态约束是仅用状态变量和时间表示的不等式约束, 混合约束是对可能依赖于状态变量和时间的控制变量的约束), 并证明了状态不等式约束是连续函数空间中的约束, 本质上具有无限大的性质。因此, 一个具有状态不等式约束的问题不可能在不损失信息的情况下简化为没有状态不等式约束的问题。在 2000 年, Cervantes 和 Biegler^[6]总结了有如单重打靶法和多重打靶法等的方法求得两点边值问题最优解。间接法理论基础完善, 在求解较小规模及具有特定形式的动态优化问题上拥有精确、快速的优势。在但其存在明显的局限性, 其一, 此方法需要系统变量和函数满足连续性和可微性; 其二, 若考虑约束条件, 需要引入更多的乘子和相关互补条件, 上述方程的复杂性增加, 求解将愈加困难, 甚至无法获得解析解, 以至于仍然需要另外的数值解法, 反而失去了解析解的优势。而且, 如果问题需要处理积极不等式约束, 则通常很难找到解析解^{[7][15][16][17]}; 其三, 对于高阶系统来说应用其方法求解问题是很困难的。在实际应用中, 动态优化问题通常是高维、非线性、带控制约束和状态约束的, 此类间接解法在实际应用中受到很大的限制。

1.3.2 直接动态优化方法

直接法也称为数值解法。由于决策变量是一段时间内无限维的控制变量, 因此需要在时间域上进行离散化, 将无限维的动态优化问题近似离散为有限维的优化 NLP 子问题, 然后采用传统最优化方法进行求解。根据离散程度的差别, 常用的两种离散策略为全部离散法, 即同步策略、及部分离散, 也称控制向量参数化方法 (Control Vector Parameterization, CVP)。

同步策略, 也称为全部离散化方法, 即对状态变量和控制变量同时离散化。此方法首先将优化时间区间划分为若干个子区间, 在每个子区间 (称为有限元) 上对状态变量和控制变量同时离散, 在有限元上用基函数近似状态和控制轨迹, DAE 系统将会被转化为一个 NLP 问题。这些基函数早期经常为简单的单项式函数, 后期又有了正交多项式 (如 Legendre 或 Chebyshev 多项式)。如, 利用拉格朗日多项式在每一个配置点将微分方程转化为插值代数方程, 然后将系统微分方程约束转化为配置点处的残差方程, 将代数约束转化为配置点处的代数约束, 然后构造一个以离散配置系数和有限元 (子区间) 长度作为决策变量的 NLP 问题, 利用有效的 NLP 求解器即可求解, 解到最优值后需进行仅一次微分方程数值计算即可获得最优控制轨迹。同步策略亦称为正交配置法^[18], 因为此方法选择的配置点的位置是正交多项式的根。配置点通常是第一类或第二类契比雪夫多项式的根, 或者是 Gause、Radau 或 Lobatto 正交的基点。Betts 等人^[19]用同步方法

以及网格细化策略求解力学和航空航天中地球火星转移轨道和地球月球转移轨道的案例研究。Kameswaran 等人^[20]在同步策略中使用了 Radau 配置法, 允许在每个有限元的末尾设置约束, 从而更有效地稳定高阶 DAE 系统, 同时也缓解了使用传统高斯配置法所带来的最优控制轨迹振荡的不好现象。Young^[21]回顾了整个 OC 方法, 并展示了对涉及流体的边界条件的正确处理, 证明了对通量边界条件的适当处理可以将误差减少。OC 方法直接将 DAE 系统的求解与优化问题联合起来, DAE 系统只在最优点运行一次, 因此可以避免计算中间解所需的大量计算; 此外, 他可以较好的处理约束问题, 可以保证在配置点上不违反约束。但此方法要求控制变量与状态变量在同一级别上同时离散, 当所需解精度较高时, 产生的近似 NLP 问题规模呈爆炸式增长, 需要特殊的大规模求解器, 对初始解的要求过高, 不易高效求解。

部分离散, 也就是控制向量参数化方法, 仅离散控制变量, 是动态优化方法中应用最广泛的数值计算法。1970 年, Pollard 等人^[22]率先使用分段常量 CVP 法, 获得了精馏塔的离线最优控制轨迹的近似数值解。与以上全部离散化方法类似, 此方法首先根据指定的离散化网格对优化时间段进行切分, 每一小段可用常数、多项式等基函数进行近似拟合, 则每小段控制在小时间段内轨迹连续, 且可仅由一个或多个有限维参数所表示。这样, 求解最优控制轨迹的随时间无限维问题, 近似转变为有限维的最优参数选择问题, 由此动态优化问题转化成了静态优化问题, 其近似程度取决于控制的参数化方法, 即基函数选择与时间段分配。基于控制参数化的方法有三大难点, 一、如何高效准确求得转换后非凸 NLP 问题的最优解; 二、如何确定离散网格能同时保证精度和计算效率; 三、如何满足控制变量和状态变量过程约束。

1) 在转换后 NLP 问题的求解方面, 由于 CVP 方法只离散化控制变量, 因此需要在每次迭代中将分段时间连续的控制变量代入 DAE 系统中, 方可求得系统状态变量, 才可判断其对应状态变量是否满足约束, 判断其对应目标函数好坏。由于在实际生产过程中, 动态系统通常是非线性、强耦合、多约束的系统, 其对应动态优化问题也通常是多局部极值、适应值函数曲面平坦等特点。早期主要采用数学规划算法, 例如最速下降法、共轭梯度法、高斯牛顿法、BFGS 法 (Broyden-Fletcher-Goldfarb-Shanno), 信赖法, SQP 算法 (Sequential Quadratic Programming)^[45]等, 此类算法具有收敛速度快和求解精度高的优点, 但存在难以选取的初始点和容易陷入局部极值的缺陷。2003 年, Cruz^[46]对多峰动态优化问题的研究表明, 基于梯度信息的 CVP 方法很多时候只能找到局部最优控制轨线。在实际生产过程, 动态优化问题通常具有非线性、多约束、强耦合、不连续、不含显示描述等特性, 因此通常选用直接法求解, 即将随时间连续的无穷维问题离散化为非线性

规划子问题 NLP, NLP 的求解效率和精度极大的影响了动态优化最优解的求解效率和精度。许多动态优化方法在求解离散后形成的 NLP 子问题时, 常依赖于梯度信息进行优化, 尽管其具有收敛速度快, 收敛精度高的特点, 但当待求解动态优化问题为多峰值类型时, 该类算法常常收敛到局部最优解。近年来, 智能算法在动态优化领域中有较为广泛的研究和应用。

随着人工智能学科不断发展, 涌现了一大批以遗传算法 (Genetic algorithm, GA) [47]、差分进化算法 (Differential evolution, DE) [48]、粒子群算法 (Particle swarm optimization, PSO) [49] 等为代表的智能进化算法。该类优化算法充分利用生物学、心理学等知识在优化问题求解过程中模拟人类活动或生物寻找食物的自然现象寻找最优解。与传统优化算法一致的是, 智能进化算法同样通过迭代实现寻优, 不一致的是, 此类算法不需要计算优化问题的梯度信息, 其迭代过程仅依赖于目标函数适应值, 其能够避免停滞在导数为零的局部极值点, 降低对初值解的敏感度, 在求解多峰问题时能更大概率地找到全局最优值。

在遗传算法方面: 2007 年, Pham^[51]在 GA 中引入渐进步长递减策略, 并使用因子实验和粗糙统计来分析算法参数, 并用其求解动态优化问题。Pham^[53]又于 2012 年提出了基于因子实验和模糊逻辑的 GA 参数设置方法, 旨在平衡收敛速度、鲁棒性(对每个问题的一致性性能)和通用性(对许多不同问题的适用性)。2011 年, Wongrat^[57]等人基于控制向量参数化方法和求解混合整数动态优化问题, 该问题的整数部分反映了工艺方案, 而动态部分则来源于非线性微分代数方程描述了稻谷的干燥特性, 并采用遗传算法求取其混合整数 NLP 问题, 最后应用于大米干燥过程。2013 年, 钱锋等人^[59]基于控制向量参数化方法和迭代遗传算法用来求解化工过程动态优化问题, 此算法将不可行解信息嵌入到评价函数里来解决动态优化问题约束, 此算法具有一定的鲁棒性并且适应于梯度未知情况。

在粒子群算法方面: 2012 年, 孙帆、杜文莉等^[58]提出了具有局部寻优算子的改进差分进化算法, 求解了补料分批式生化反应器的动态优化问题。2013 年, 陈旭、杜文莉等人^[60]提出了混合梯度粒子群算法来优化 NLP 子问题, 并求解了奇异控制问题、多模态问题、高维问题、以及约束问题等多种形式的动态优化问题。2014 年, 周游、刘兴高等人^[62]为了提高搜索效率和收敛速度, 引入并提出具有参数更新的 PSO, 提出了一种基于控制参数的自适应粒子群优化方法 (CP-PSO), 并求解了三种经典的动态优化问题。2016 年, 张盼盼、刘兴高^[61]等人对群智能动态优化方法进行研究, 其阐述了以细菌觅食、杂草入侵、粒子群为基础的三种智能动态优化方法, 并研究了综合了 PSO 和差分搜索算法的混合型智能动态优化算法, 此外, 还提出了基于多目标粒子群算法的动态优化方法并证明其在 Pareto 前沿面上良好的收敛性能和分布性能。

在其他智能进化算法方面: 2005 年, Faber 等^[50]利用模拟退火算法 (Simulated annealing, SA) 来进行动态优化, 成功求取非线性间歇反应器最大产量动态优化问题、简化的单压联合循环发电厂快速增加负荷的动态优化问题。2008 年, Asgari^[52]研究了区域收缩策略 (RRS), 用非线性高阶多项式逼近策略来求解动态优化问题, 其使用上一次迭代的最优解作为基准解来产生下一次迭代的收缩区域来获得更高精度, 并采用蚁群算法 (Ant Colony Optimization, ACO) 求解 NLP 子问题。2010 年, 刘宗其、钱锋等人^{[54][55][56]}利用基于知识改进文化算法对化工过程动态优化问题进行有效求解。

智能进化算法在解决各类工程实际动态优化问题上取得了较好的成效, 该类算法不需要计算梯度信息, 不要求显示描述, 因此适应性更广泛, 不依赖初始点, 更有利于处理实际多峰值、不连续的复杂动态优化问题。但进化算法求解相对耗时, 因此算法求解总体花费时间长, 效率低, 难以满足秒级实时优化要求。

2) 对于离散网格确定问题, 1996 年、1997 年, Tanartkit 等^{[23][24]}将双层优化应用到动态网格的求取中, 他们将动态优化问题转变为两层, 外层优化控制区间的长度或者时间点, 内层则在外层确定动态网格情况下, 求取最优问题的解。1997 年, Waldraff^[85]基于曲率信息来增补网格点, 确定网格点个数。1999 年, Teo 等^[25]提出了著名的 Time-Scaling 方法, 在优化控制值的同时优化每一段子区间的时间长度, 其子区间总和需为总优化长度, 在求解得到最优控制参数的同时获得此分段数下的最佳的控制网格。2003 年, Srinivasan 等^[26]研究了最优控制的子区间控制轨迹与最优必要条件之间的关系, 并将必要条件划分为约束和灵敏度两部分, 提出了基于灵敏度的自适应 CVP。1997 年, Schlege 等^[27]提出一种基于小波的自适应网络划分的 CVP 法, 在迭代过程中通过对上一步解的小波系数来调整时间网格的划分, 对某些时间网格进行细化或合并。2014 年, Assassa 等^[28]将小波自适应法用于 CVP, 并将其应用于动态不稳定的 HIPS 聚合过程。2014 年, 针对工业过程, 陈旭等人^[29]提出了新的非均匀动态离散化的动态优化框架, 此方法将离散时间长度和对应值 (采用分段常数近似) 作为决策变量同时优化, 并通过归一化对整个优化时间段进行映射, 用以避免所有时间段相加需要等于原时优化时长的约束。2016 年, 李国栋等^[30]提出了基于斜率的自适应 CVP 方法, 方法核心思想为先给定一个粗糙控制网格, 然后再根据控制节点左右斜率值增删网格节点。2016 年, 刘平等^[31]基于经验模态分解的时间网格重构 CVP 方法, 该方法对控制参数信息提取趋势项信息用以重构时间节点。2017 年, 王立伟等^[32]提出了一个新的基于灵敏度的自适应控制向量参数化方法, 并应用于生物反应过程的动态优化上。

3) 对于约束处理问题, 1971 年, Jacobson 等^{[33][34]}提出了松弛变量法, 该方

法利用松弛变量将具有状态变量不等式约束的问题转化为具有更高阶微分方程的更高维无约束问题。但该方法对控制变量有约束或状态控制耦合约束存在的情况下较难求解，当状态不等式约束多于控制变量时甚至无法用松弛变量法求得^[35]。此外，较常用的约束处理方法有外罚函数法和内点约束法^{[36][37]}，该类方法直接将不等式约束转为为离散点约束，并将约束惩罚项以某种形式增广到原目标函数。1998 年，Barton 等^[35]在论文中提出一个波动指数算法，该算法通过在每一代中不断检测初值问题求解过程中约束的积极性，利用伪导数法求解得到高阶 DAE 及其相关灵敏度。2005 年，Vassiliadis 等^[38]提出了基于分段离散有限迭代 ε -收敛策略来处理不等式路径约束，该方案将在给定容差 ε 下的有限次迭代中收敛。近期，Teo 等人通过光滑化精确罚函数方法^{[39][40][41][42][43][44]}，缓解 max 处理状态不等式约束或状态控制耦合不等式约束所带来的不连续问题，同期涌现了大量光滑化罚函数方法来处理不等式约束问题。

1.4 论文研究内容与结构安排

本研究是在国家自然科学基金项目课题“有色冶金级联反应器不确定动态优化方法”（61873285）、国家自然科学基金青年科学基金“基于分布式状态转移算法的有色冶金过程系统辨识方法研究”（61503416）、中南大学创新驱动计划“流程工业过程数据建模与分布式优化方法研究”（2018CX012）和中南大学研究生科研创新项目（2017zzts487）的资助下开展的。

动态优化是实现生产过程稳定运行、节能降耗、挖潜增效的重要手段。随着对动态优化问题的研究深入，发现实际动态优化问题主要有非线性、高维、多控制变量、多约束等特点，以 CVP 为基础的数值动态优化方法在实际问题中应用十分广泛。本论文在以 CVP 为离散方法的基础上，深入分析了影响动态优化方法效率和精度的三大难点，1) 如何求得 CVP 转换后多模态非凸 NLP 问题的最优解；2) 如何确定最优离散网格同时保证求解精度和计算效率；3) 如何满足控制变量和状态变量过程约束。考虑到状态转移算法作为新型智能算法具有良好优化性能，因此，本文针对以上三大难点问题展开基于状态转移算法的动态优化方法研究工作。研究内容主要包括：

本文针对工业生产实际动态优化问题，对一类常见的单目标优化时长固定，有约束的动态优化问题求解方法做深入研究。首先，在控制参数化离散方案下，考虑到实际动态优化问题的强非线性、多模态特点，本文首先采用均匀划分控制参数化处理其动态特性，随后引入并改进了全局智能算法状态转移算法，并提出了基于变尺度状态转移算法的动态优化方法。针对均匀划分下逼近精度和计算效率之间的矛盾，提出了具有部分移动非均匀划分方案的自适应动态优化方法；同

时,设计了混合梯度状态转移算法,最终能高效、准确和快速地求解具有不同结构的无约束动态优化问题。接下来,考虑到实际动态优化问题中的状态约束、控制约束以及混合状态控制约束等特点,引入了光滑化罚函数方法,提出了基于状态转移算法的光滑约束动态优化方法。最后,本文以湿法炼锌除铜过程最优控制问题为研究对象,建立了基于质量守恒的最小化锌粉添加量的动态优化问题模型,采用基于状态转移算法的约束动态优化方法实现有效求解,最终达到了原料节约、工况稳定的目标。本文共包括五章内容,各章节主要内容如示意图 1-1 所示。

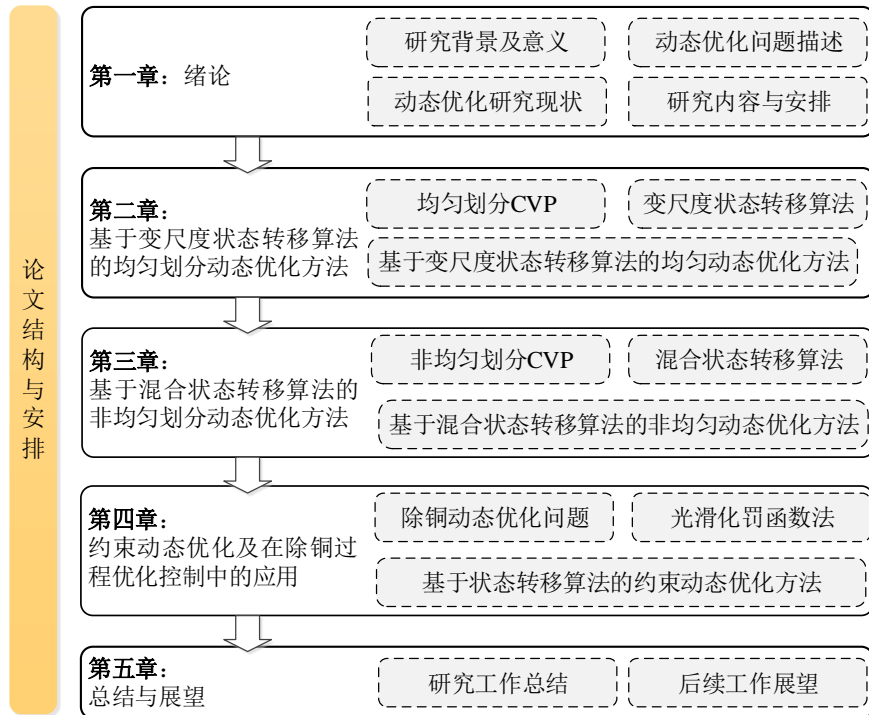


图 1-1 论文整体结构框图

第一章：绪论。介绍了动态优化的研究背景与意义,描述了动态优化问题基本概念。论述了动态优化方法的分类及其国内外研究现状,重点介绍了数值型动态优化方法和智能动态优化方法的研究现状,为后续内容开展奠定基础。

第二章：基于变尺度状态转移算法的均匀划分动态优化方法研究。本章详细阐述了 CVP 方法的基本原理,首先采用了均匀划分策略将动态问题转换为静态优化问题,随后针对其复杂、非凸、多模态特性,引入状态转移算法进行求解。其间,分析了状态转移算法原理和参数,提出了具有自适应参数策略的状态转移算法。最后,通过经典实例验证了该方法求解非线性动态优化问题的全局性。

第三章：基于混合状态转移算法的非均匀划分动态优化方法研究。本章首先深入分析了均匀离散方案下,离散程度与求解效率的矛盾问题,分析了两种非均匀离散方案优劣,提出了部分离散网格非均匀离散 CVP 方法;接着,针对算法

求解效率问题，提出了混合梯度状态转移算法（HSTA）；随后，综合以上两种方法提出了基于混合状态转移算法的非均匀划分动态优化方法（ndCVP-HSTA）；实验测试证明了该方法能高效获得高精度控制曲线。

第四章：动态优化在湿法炼锌除铜过程最优控制的研究。本章首先从工艺角度出发分析了除铜过程的动态特性。随后，根据其效用指标、执行器约束、工况约束、动态系统约束等，构建了除铜过程动态优化问题。紧接着，考虑到存在的状态变量约束，在以上无约束优化方法基础上引入光滑化罚函数法，提出了基于ndCVP-HSTA的光滑约束动态优化方法；最后，两个经典约束动态优化问题表明了方法的可行性。随后，应用该方法成功求解了除铜过程动态优化问题，实现了工艺过程工况稳定、节能降耗的目标。

第五章：结论与展望。对本论文研究工作进行简括，总结研究不足点，提出下一步研究。

2 基于变尺度状态转移算法的均匀划分动态优化方法

实际工业动态优化问题多具有非线性、多模态、不连续、不含显示描述等特点，因而在学术与工业实际上经常采用控制变量参数化（CVP），根据某种划分方案参数化控制变量，随之将原问题转变为高维非凸非线性规划（NLP）问题。状态转移算法是一种新型的智能型全局优化算法，在各个优化领域展现了较好的全局搜索能力和较快的收敛性能，在处理动态优化问题参数化后带来的非凸 NLP 上具有潜在优势。本章将介绍 CVP 的基本原理，首先采用均匀划分参数化方案，在处理非凸 NLP 问题上，介绍、分析并改进了状态转移算法，并且提出了一种基于改进状态转移算法的均匀划分无约束动态优化方法。

2.1 均匀划分控制向量参数化策略

动态优化问题需要求解在时间上连续的控制变量轨迹，由于实际工程问题复杂，往往无法求得最优解析解，因此常用控制变量参数化方法来求得数值解。CVP 方法的核心思想就是将无限维的控制变量轨迹用有限维的参数来拟合描述，将无限维最优轨迹问题转变为有限维最优参数问题或 NLP 问题。

2.1.1 控制向量参数化

CVP 通常包含两部分：1) 首先将待优化时间区间划分为若干个子区间，2) 其次在每个子区间中将控制变量以某种基函数近似。由此控制变量在优化控制区间上被划分为有限个子区间，每个子区间的时变控制曲线由有限个参数的基函数逼近。常见的基函数有常函数、线性函数、二次函数等，如图 2-1 所示。

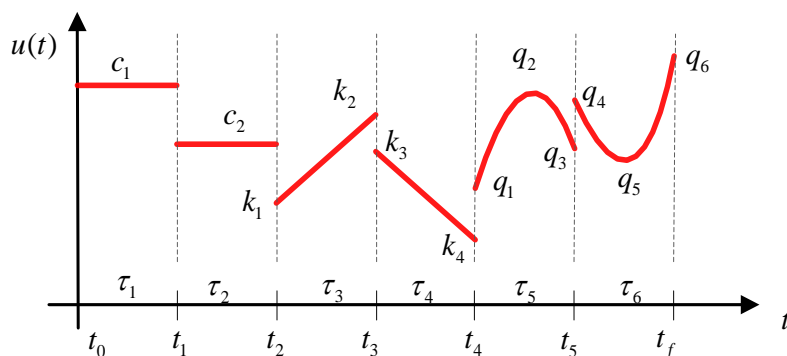


图 2-1 不同基函数下的控制向量参数化

如图 2-1 所示，在已给定离散时间区间下，即 $[t_0, t_1, t_2, t_3, t_4, t_5, t_f]$ ，不同的基函数有不同的参数表达。1) 以常函数作为基函数，则在一个子区间仅需要优化一个决策变量，如在 $[t_0, t_1]$ 时间段内的控制变量可由 c_1 表示；2) 以线性函数作为

基函数，则在一个子区间则需要优化两个决策变量，如在 $[t_2, t_3]$ 时间段内的控制变量可由 k_1 与 k_2 决定。3) 以二次函数作为基函数，则在一个子区间需要优化三个决策变量，如在 $[t_4, t_5]$ 时间段内的控制变量可由 q_1 、 q_2 与 q_3 决定。在实际使用时，用何种基函数取决于所需控制曲线的连续性、光滑性、跳变性等具体要求。

分段常数的近似策略是最基本最常用的近似策略。其一、由于计算机只能处理数字信号，且其处理有一定的精度，在此精度下，计算机的输出信号都是分段常量信号，而不能直接输出线性、二次型信号^[63]。其二，分段常数近似方案以其控制值作为决策变量，用分段恒定信号实现控制。因为实现有限长度的恒定输入值通常比实现连续变化的控制信号容易得多，因此其因简单有效而被工业和学术界普及。其三，在许多实际控制问题中（如 bang-bang 控制问题），真正的最佳控制确实是分段常数函数。其四，分段常数近似方案的其他优点包括其强大的收敛特性和处理非标准最优控制问题的灵活性^[64]。因此，本文仅考虑分段常数拟合的参数化方法。

在分段常数近似方案下，以控制向量 $u(t)$ 为例。在一定离散网格之下，时间区间 $[t_0, t_f]$ 被时间点 t_i 分为 p 个子区间 τ_i ，其中离散时间点满足如下表达式：

$$t_0 < t_1 < t_2 < t_3 < \dots < t_p = t_f. \quad (2-1)$$

离散化后，子区间内控制曲线如下所示：

$$u(t) = u_p(t) = \xi_k, t \in [t_{k-1}, t_k), k = 1, \dots, p, \quad (2-2)$$

其中 ξ_k 是第 k 段 $[t_{k-1}, t_k)$ 的控制值。则整个控制区间的近似控制曲线为：

$$u_p(t|\xi) = \sum_{k=1}^p \xi_k \chi_{[t_{k-1}, t_k)}(t), t \in [0, T), \quad (2-3)$$

$$\xi = \left[(\xi_1)^T, \dots, (\xi_p)^T \right]^T, \quad (2-4)$$

$$\chi_{[t_{k-1}, t_k)}(t) := \begin{cases} 1, & \text{if } t \in [t_{k-1}, t_k), \\ 0, & \text{if } t \notin [t_{k-1}, t_k), \end{cases} \quad (2-5)$$

可以注意到， $u_p(t|\xi)$ 在时间节点 $t = t_i, i = 1, \dots, p-1$ 上转变控制值，因此这些时间节点通常称之为切换时间点（Switching times）。

2.1.2 均匀划分控制向量参数化

每个离散网格点之间的距离可以相等，即均匀划分（uniform discretization），或不相等，即非均匀划分（un-uniform discretization），如图 2-2 所示。均匀划分方案只需要决定分段数 N 即可确定每个控制子区间的长度或离散网格点的位置，随即求取对应控制子区间内控制参数即可得到此方案下近似最优控制轨迹。但均

均匀离散化方案不适应于有奇异弧的控制曲线, 最优控制曲线拟合精度的上限较低。非均匀离散化方案同时优化切换时间点, 能在一定程度上提高解的自由度, 改进解精度, 但其将带来控制变量增多或反复精细化迭代时间长等缺点。

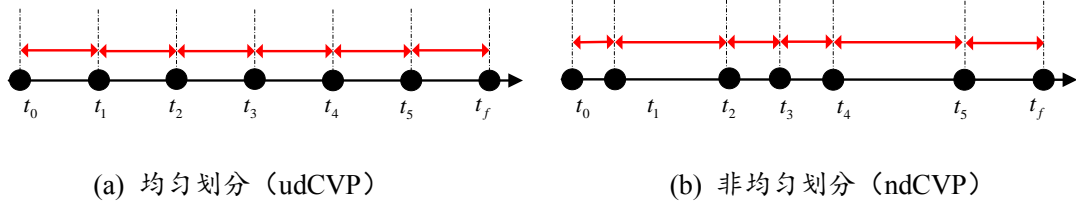


图 2-2 不同划分方案下的控制参数化时间网格

基于均匀划分的动态优化方法是所有动态数值型动态优化的基础, 一则因为均匀划分较简单基础, 便于优化; 二则实际操作方便, 便于执行。因此我们先采用基于均匀划分的动态优化方法。

本章旨在先求解无约束动态优化问题, 为接下来研究做铺垫。具体来说, 无约束指无与状态相关约束, 考虑一般无约束优化问题, 其表达式如下:

$$\begin{aligned}
 \min_{\mathbf{u}(t)} \quad & J = \Phi[\mathbf{x}(t_f)] + \int_{t_0}^{t_f} L[t, \mathbf{x}(t), \mathbf{u}(t)] dt \\
 \dot{\mathbf{x}}(t) = & f[\mathbf{u}(t), \mathbf{x}(t), t] \\
 \mathbf{x}(t_0) = & \mathbf{x}_0 \\
 \mathbf{u}^l \leq & \mathbf{u}(t) \leq \mathbf{u}^u \\
 t_0 \leq & t \leq t_f
 \end{aligned} \tag{2-6}$$

其中, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ 为 n_x 维状态变量, 系统的初始状态为 \mathbf{x}_0 , $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ 为 n_u 维控制变量; 目标函数 J 为预定的一个性能指标。控制变量必须满足 $\mathbf{u}^l \leq \mathbf{u}(t) \leq \mathbf{u}^u$; 终止时间 t_f 为固定常数。

在均匀、分段常数的近似方案下, 且假设 n_u 个控制轨迹均遵循同一个离散化方法, 则原动态优化问题目标函数可离散化为:

$$\begin{aligned}
 \min_{\xi_i} \quad & J' = \Phi[\mathbf{x}(t_f | \xi)] + \sum_{k=1}^p \int_{t_{k-1}}^{t_k} L[t, \mathbf{x}(t | \xi), \xi^k] dt \\
 \dot{\mathbf{x}}(t) = & f[\xi_i, \mathbf{x}(t), t] \\
 \mathbf{x}(t_0) = & \mathbf{x}_0 \\
 \mathbf{u}^l \leq & \xi_i \leq \mathbf{u}^u, i=1, \dots, n_u \\
 t_0 \leq & t \leq t_f
 \end{aligned} \tag{2-7}$$

其中决策变量为 $\xi_i, i=1, \dots, n_u, \xi_i, i=1, \dots, n_u$, 维数为 $n_u \times p$, 则问题转变为: 找出 n_u 个 p 维向量 ξ 使得目标函数 J' 最小, 其中决策变量仍需满足原始约束。

综上，最初无限维的动态优化问题近似为有限维的 NLP 问题，经过研究，当 p 无限大、子区间数趋近于无穷时，此近似问题最优解无限收敛于具有任何形态结构的原动态优化问题最优解，其相关证明可见参考文献[65][66]。

离散化程度决定了 NLP 最优解与原问题最优解的近似上限，那么 NLP 求解器的求解能力决定了最终数值解接近 NLP 最优解的程度。传统基于梯度的优化算法，其收敛速度快，局部收敛精度高，但该类算法常常收敛到局部最优解。近年来，智能算法在解决各类工程实际问题取得了较好的成效，该类算法适应性广泛，全局搜索能力较强等优点，在处理实际工业动态优化问题上有着独特的优势。以下介绍一种新型智能优化算法——状态转移算法。

2.2 状态转移算法简介

状态转移算法是一种新型的随机性全局优化算法^[8]，并已被证明能有效求解各类优化问题，如无约束优化问题^[67]、有约束优化问题^[68]、整数优化问题^[69]、一般离散优化问题^[70]、线性双层优化问题^[71]等，同时在多种应用中展现了较优的求解性能，如 PID 控制器整定问题^[72]，图像多阈值分割问题^[73]，湿法炼锌沉铁过程多目标优化控制问题^[74]等问题。

2.2.1 状态转移算法原理

在状态转移算法的思想中，最优化问题的解被视为一个状态，那么解的更新过程则可视作状态转移。随着迭代次数的增加，解将会由特殊设计的四个算子逐渐的转移到最优点。一般来说，状态转移算法中候选解产生的统一框架如下：

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k u_k \\ y_{k+1} = f(\mathbf{x}_{k+1}) \end{cases} \quad (2-8)$$

其中， \mathbf{x}_k 代表当前状态，对应最优化问题的具有一定维度的当前解； \mathbf{A}_k 和 \mathbf{B}_k 为状态转移矩阵，一般设计为随机矩阵，可以看成是最优化算法中的算子； u_k 为当前状态 \mathbf{x}_k 及历史状态的函数； $f(\cdot)$ 为目标函数或评价函数。

状态转移算法的优越性能主要由其特殊设计的、有一定几何意义的状态转移算子及其邻域采样决定。在连续状态转移算法中， $\mathbf{x}_k \in \mathbb{R}^n$ 是一个 n 维连续变量，在此设计了四个特殊的状态变换算子：旋转变换、平移变换、伸缩变换、坐标变换来产生候选解。

1) 旋转变换 (Rotation transformation, RT)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \frac{1}{n \|\mathbf{x}_k\|_2} \mathbf{R}_r \mathbf{x}_k \quad (2-9)$$

其中, α 为旋转因子, 是一个正常数, $\mathbf{R}_r \in \mathbb{R}^{n \times n}$ 是一个每一个元素服从 $[-1,1]$ 的均匀分布的随机矩阵。 $\|\cdot\|_2$ 表示向量的二范数, n 为维数。

2) 平移变换 (Translation transformation, TT)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta \mathbf{R}_t \frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2} \quad (2-10)$$

其中, β 为平移因子, 是一个正常数, $\mathbf{R}_t \in \mathbb{R}^{n \times n}$ 是一个每一个元素服从 $[0,1]$ 的均匀分布随机矩阵。

3) 伸缩变换 (Expansion transformation, ET)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda \mathbf{R}_e \mathbf{x}_k \quad (2-11)$$

其中, λ 为伸缩因子, 是一个正常数, $\mathbf{R}_e \in \mathbb{R}^{n \times n}$ 是一个随机对角矩阵, 它里面的每一个非零元素服从均值为 0、方差为 1 的高斯分布。

4) 坐标变换 (Axesion transformation, AT)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta \mathbf{R}_a \mathbf{x}_k \quad (2-12)$$

其中, δ 为坐标因子, 是一个正常数, $\mathbf{R}_a \in \mathbb{R}^{n \times n}$ 是一个随机对角稀疏矩阵, 它只在某个随机位置有非零元素, 其非零元素服从高斯分布。

2.2.2 状态转移算法流程

综上, STA 从当前最好解出发, 利用变换算子产生邻域、从邻域中采样形成候选解样本、选择并更新形成下一个状态最好解。其算法具体步骤如图 2-3 所示:

Step 1: 设置算法参数 $\alpha = \alpha_{\max}, \alpha_{\min}, \beta, \gamma, fc, SE$; 令 $k = 0$; 随机产生一个具有一定维度的初始解 \mathbf{Best}_0 。

Step 2: 利用 ET 变换基于 \mathbf{Best}_k 产生 SE 个候选解, 候选解与 \mathbf{Best}_k 进行贪婪更新。如果更新成功, 则利用 TT 变换基于 \mathbf{Best}_k 产生 SE 个候选解, 候选解与 \mathbf{Best}_k 进行贪婪更新;

Step 3: 利用 RT 变换基于 \mathbf{Best}_k 产生 SE 个候选解, 候选解与 \mathbf{Best}_k 进行贪婪更新。如果更新成功, 则利用 TT 变换基于 \mathbf{Best}_k 产生 SE 个候选解, 候选解与 \mathbf{Best}_k 进行贪婪更新;

Step 4: 利用 AT 变换基于 \mathbf{Best}_k 产生 SE 个候选解, 候选解与 \mathbf{Best}_k 进行贪婪更新。如果更新成功, 则利用 TT 变换基于 \mathbf{Best}_k 产生 SE 个候选解, 候选解与 \mathbf{Best}_k 进行贪婪更新;

Step 5: 置 $k = k + 1$, 如果 $\alpha < \alpha_{\min}$, 置 $\alpha = \alpha_{\min}$, 否则置 $\alpha < \alpha / fc$, 然后重返 Step2 直到终止条件满足。

其中终止条件一般为最大迭代次数。基本连续状态转移算法的默认参数如下:

表 2-1 算法默认参数

参数	值
α_{\max}	1
α_{\min}	0.0001
β	1
γ	1
δ	1
fc	2
SE	30

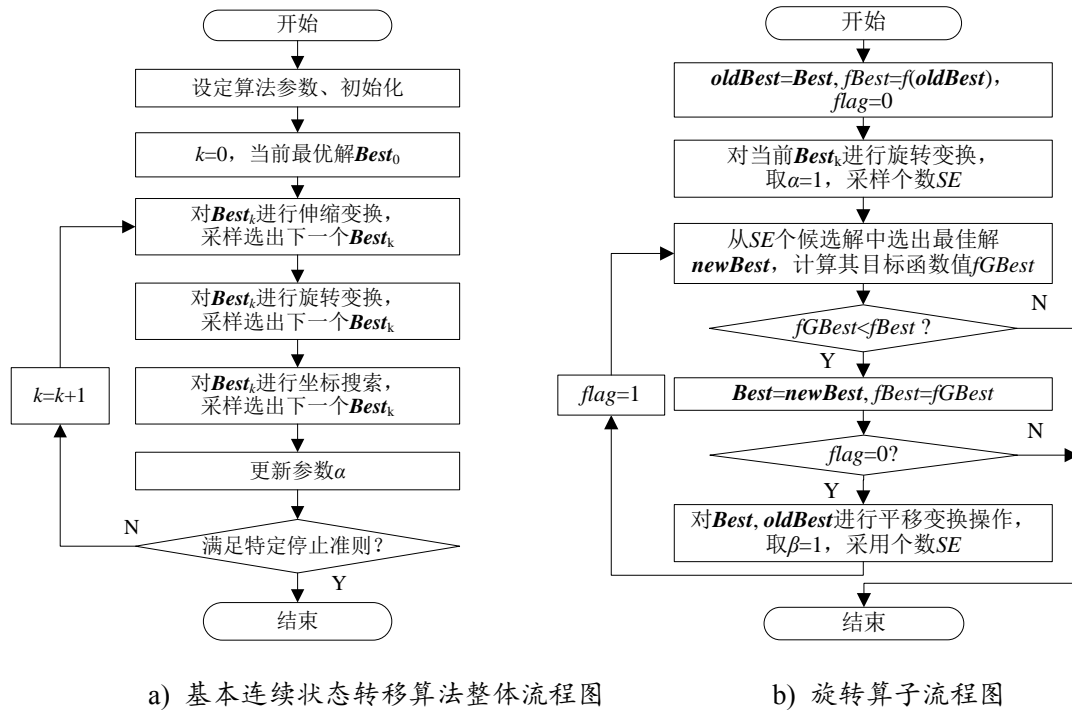


图 2-3 基本连续状态转移算法流程图

其中， $Best_k$ 为当前最优解。由此流程图以及算法步骤 Step 2、3、4 中，对当前最优解顺序执行伸缩变换与更新、旋转变换与更新、坐标变换与更新。值得注意的是，平移变换算子嵌入了伸缩、旋转、坐标变换中，实现进一步的细化搜索。右图展示旋转变换操作流程图来示意各算子使用规则。

2.3 变尺度状态转移算法

本节首先分析了状态转移算子性能及其影响因素，为了提升算子全局搜索和局部搜索性能，提出了一种变尺度状态转移算法，其能根据算法搜索性能动态更新算子参数，扩大或缩小搜索尺度。随后，提出了基于变尺度状态转移算法的动态优化方法。

2.3.1 状态转移算子性能分析

STA 为智能进化算法,在每次迭代过程中,基于当前最优解根据四个核心算子循环搜索、采样更好解,并以贪婪准则进行更新,达到一步步转移到全局最优解的目的。一个好的算法需要兼顾全局最优性能和局部最优性能,因而 STA 中的四个核心算子各有侧重点,如旋转变换算子和转移算子旨在提高算法的局部精细化搜索能力,伸缩算子和坐标算子旨在保证算法的全局搜索性能。算子功能的实现由其特殊的概率变换矩阵和参数所决定,具体性能分析如下:

1) 旋转变换 (Rotation transformation, RT)

对于旋转变换算子,假设当前状态 $\mathbf{x}_k = [1,2]$, 旋转因子 $\alpha = 1$, 利用旋转变换算子产生 $SE=2000$ 个候选解,则可产生如下图 2-4 所示的候选解邻域,其中“*”代表候选解。其特性为: a) 由旋转算子表达式可知,当前最好解确定时,最终候选解等于当前解再加增量 $\alpha' \cdot \mathbf{R}_r \cdot \hat{\mathbf{x}}_k$, 其中 $\alpha' = \alpha/n$ 为常数, $\hat{\mathbf{x}}_k = \mathbf{x}_k / |\mathbf{x}_k|$ 为 \mathbf{x}_k 的单位向量, $\mathbf{R}_r \cdot \hat{\mathbf{x}}_k$ 为 n 个以 $\hat{\mathbf{x}}_k$ 各维度值为均值的均匀分布相加, \mathbf{R}_r 中各值为 $[-1,1]$ 均匀分布,因而其各维度增量符合同一分布; b) 旋转算子可以产生一个以当前最好解为中心的立方体邻域; c) 此变换每次产生候选解范围一定。当前最好解变更为 $\mathbf{x}_k = [1,100]$ 时,所产生候选解仍在以当前最优解为中心,边长为小于 α 的超立方体内搜索的功能; d) 由此,此算子具有局部搜索功能,旨在当前最优解附近精细化搜索。

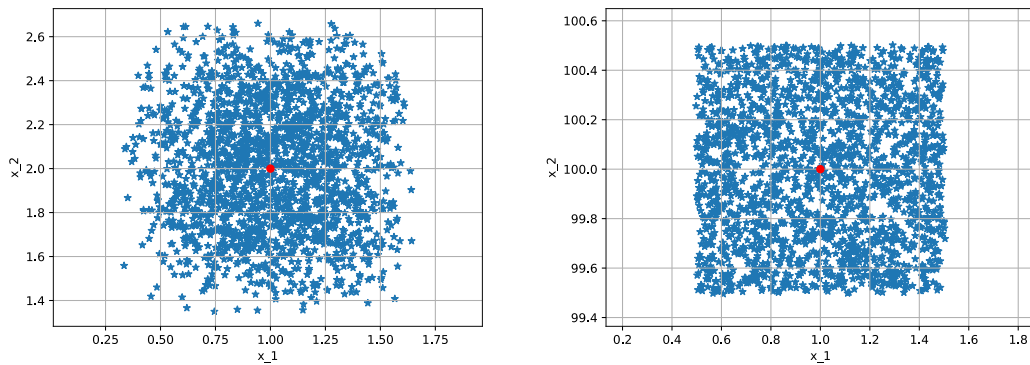


图 2-4 当前最优解为 (1, 2) (左) 和 (1, 100) (右) 旋转变换候选解邻域

2) 平移变换 (Translation transformation, TT)

对于平移变换算子,假设当前状态 $\mathbf{x}_k = [1,2]$, 之前最好解为 $\mathbf{x}_{k-1} = [0.5,0.5]$, 平移变换因子 $\beta = 1$, 利用平移变换算子产生 $SE=100$ 个候选解,则可产生如下图 2-5 所示的候选解邻域,其中“*”代表候选解。其特性为: a) 由公式可知,当之前最优解 \mathbf{x}_{k-1} 和当前最优解 \mathbf{x}_k 确定时,最终候选解等于当前解再加增量 $\beta \cdot \mathbf{R}_t \cdot \hat{\mathbf{u}}_k$, 其中 $\hat{\mathbf{u}}_k$ 为 $\mathbf{u} = \mathbf{x}_k - \mathbf{x}_{k-1}$ 的单位向量, \mathbf{R}_t 内各元素符合 $[0,1]$ 内均匀分布; b) 由此,利用平移变换算子可产生一个沿着 $\hat{\mathbf{u}}_k$ 方向,以当前最好点 \mathbf{x}_k 为起点,最大延伸范

围欧氏距离为 β 的线型候选解“邻域”；c) 此变换每次产生候选解范围一定，如之前最好解不变，当前最好解变更为 $\mathbf{x}_k = [1, 100]$ 时，所产生候选解仍限在欧氏距离为 β 的区域，但前后两次最优解可以决定候选解区域位置。d) 由此，平移变换算子旨在启发式的提高算法的局部搜索性能。

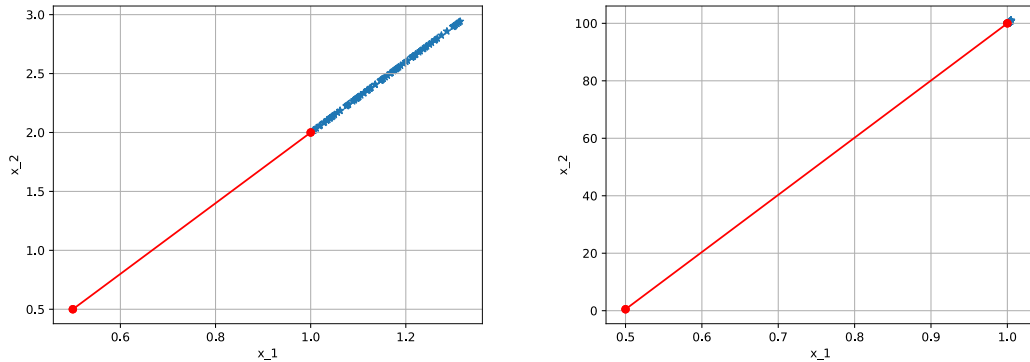


图 2-5 当前最优解为 (1, 2) (左) 和 (1, 100) (右) 下平移变换候选解邻域

3) 伸缩变换 (Expansion transformation, ET)

对于伸缩变换算子，假设当前状态 $\mathbf{x}_k = [1, 2]$ ，利用伸缩变换算子产生 $SE=2000$ 个候选解，如图 2-6 所示由“*”表示。其特性为：a) 由于最终候选解为当前解再加 $\lambda \cdot \mathbf{R}_e \cdot \mathbf{x}_k$ ，其中 λ 为常数， \mathbf{R}_e 为符合高斯分布的对角矩阵，则所加增量 $\lambda \cdot \mathbf{R}_e \mathbf{x}_k$ 在每一维上符合均值为0、方差为 $\lambda^2 \cdot \mathbf{x}_k^2$ 的高斯分布，记为 $N(0, \lambda^2 \cdot \mathbf{x}_k^2)$ 。b) 因而伸缩变换可以产生一个以当前最好解为中心的类似椭球形状邻域，新候选解 x 在每一个维度上中间值多，边缘值少。c) 此变换在概率意义上具有使得 \mathbf{x}_k 中每个元素伸缩变换到 $[-\infty, +\infty]$ 的全局搜索功能，如当前最好解 $\mathbf{x}_k = [1, 100]$ 时，利用伸缩算子，在第二维度上可大概率搜索到 $[-200, 400]$ 的区域，搜索范围受当前最优解影响较大。d) 由此，伸缩变换算子旨在提高算法的全局搜索性能。

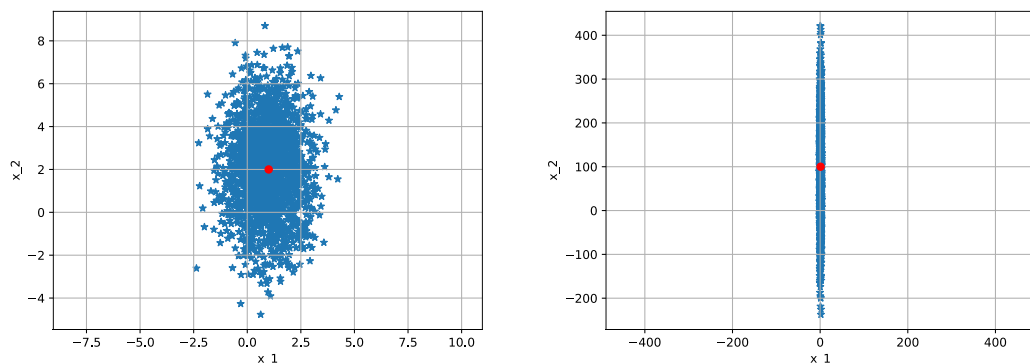


图 2-6 当前最优解为 (1, 2) (左) 和 (1, 100) (右) 下伸缩变换候选解邻域

4) 坐标变换 (Axesion transformation, AT)

对于坐标变换算子，假设当前状态 $\mathbf{x}_k = [1, 2]$ ，坐标变换因子 $\delta = 1$ ，利用平移变换算子产生 $SE = 200$ 个候选解，则可产生如下图 2-7 所示的候选解邻域，其中“*”代表候选解。其特性为：a) 由于最终候选解为当前解再加 $\delta \cdot \mathbf{R}_a \cdot \mathbf{x}_k$ ，其中 δ 为常数， \mathbf{R}_a 为符合高斯分布的对角矩阵，则所加增量 $\delta \cdot \mathbf{R}_a \cdot \mathbf{x}_k$ 仅在某一维度上有值，其余均不变，且此维上符合均值为 0、方差为 $\delta^2 \cdot \mathbf{x}_k^2$ 的高斯分布，记为 $N(0, \delta^2 \cdot \mathbf{x}_k^2)$ 。b) 因此坐标所搜算子可以产生一个以当前最好解为中心的坐标轴方向邻域，新候选解每次只在一个维度上有改变，理论上能随正态概率分布于 $[-\infty, +\infty]$ ，拥有坐标轴上的单维搜索能力。c) 如当前最好解 $\mathbf{x}_k = [1, 100]$ 时，利用伸缩算子，在某一维度上可因大方差搜索到 $-200 \sim 300$ 的区域。与伸缩变换类似，此变换搜索范围受当前最优解数值影响较大。d) 由此，坐标变换算子旨在提高算法的单维全局搜索性能。

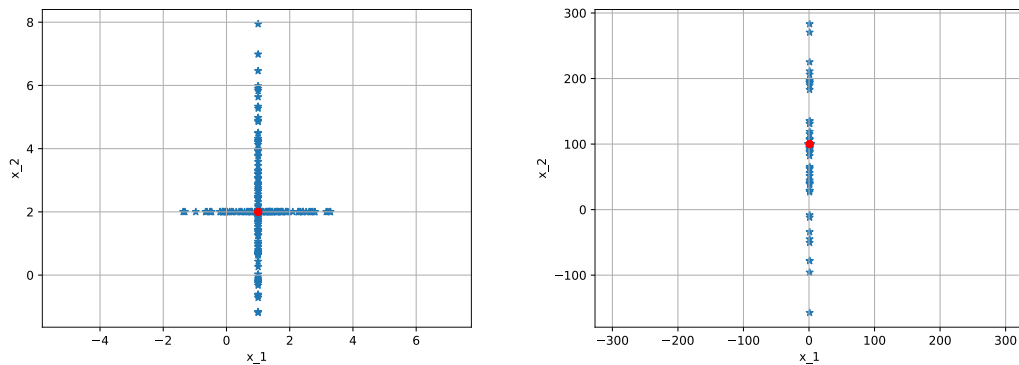


图 2-7 当前最优解为 (1, 2) (左) 和 (1, 100) (右) 下坐标变换候选解邻域

2.3.2 搜索尺度自适应策略

基本状态转移算法中，旋转变换算子和转移算子旨在提高算法的局部精细化搜索能力，伸缩算子和坐标算子旨在保证算法的全局搜索性能。伸缩变换和坐标变换能随当前最优解而获得大方差，使得有概率拓展到远离当前最优解的区域，而旋转算子和平移算子的邻域拓展范围总有限制，其范围完全受其内部算子参数旋转因子 α 、平移因子 β 决定。同 SE 个样本点下， α 、 β 增大则局部搜索尺度增大。局部搜索尺度越大，则有概率搜索到更远地方更优解，有利于加快收敛速度；局部搜索尺度越小，则有利于局部精细化搜索。

基本状态转移算法的旋转因子 α 根据迭代次数在一定范围 $[1e-4, 1]$ 内循环减小，而在整个优化过程中，平移因子 β 保持不变，限制了基本状态转移算法的局部收敛性能。 SE 一定时，若变换因子过大，则局部搜索算子较低概率产生当前最优解附近的更优解，优化后期难以收敛到局部极小点，细调难以实现；若变换因子过小，则局部搜索算子每次细调步长太小，每次迭代下降相对进步小，优化

速度慢。为了提高局部收敛能力，达到较高优化精度和较高优化速率，本节提出了一种搜索尺度自适应策略，将根据历史最优来自适应的更新局部搜索算子的范围参数。

首先，定义一个计数器 c 用来记录相对最近两次最优值（求最小值问题）的相对改进量级，如下：

$$c = \begin{cases} \max(0, c-1), & \text{if } f(\mathbf{Best}_{k-1}) - f(\mathbf{Best}_k) > \tau \\ \max(0, c+1), & \text{else} \end{cases}, \quad (2-13)$$

这里， τ 是一个指定的较小值，为 10^{-4} ， $f(\mathbf{Best}_k)$ 是第 k 次迭代的最优值。由此可知，如果两次迭代有较大改进，则 $c-1$ ，若两次迭代有较小改进则 $c+1$ 。

随后更新旋转因子 α 及平移因子 β ，如下

$$\alpha = \begin{cases} fc \times \alpha, & \text{if } c \leq C_1 \\ \frac{1}{fc} \times \alpha, & \text{if } c \geq C_2, \\ \alpha, & \text{else} \end{cases}, \quad (2-14)$$

$$\beta = \begin{cases} fc \times \beta, & \text{if } c \leq C_1 \\ \frac{1}{fc} \times \beta, & \text{if } c \geq C_2, \\ \beta, & \text{else} \end{cases}, \quad (2-15)$$

其中， C_1 和 C_2 表示计数 c 的上下阈值， fc 为衰减因子（通常取 2）。

旋转因子 α 及平移因子 β 的更新方式一致。由于算子随机性，只有在足够代数下同样现象一直发生的情况下，才改变搜索因子，改变搜索邻域范围。如果计数器小于 C_1 ，则说明一直有足够大的进步，那么搜索因子将乘以 fc ，以扩大邻域、加快优化速率；如果计数器大于 C_2 ，则说明在此搜索因子决定的搜索范围内找到更优解概率较低，则因子将除以 fc ，缩小邻域，以促进更精细化的局部搜索，直至找到足够精度下的局部极小值点。其中 C_1 和 C_2 取 4 和 9。值得注意的是，旋转因子 α 和平移因子 β 不可以无穷大或无穷小，他们均有设定范围以防止邻域过分扩张或过分细微搜索。旋转因子 α 的范围设定为 $[1e-4, 1]$ ，平移因子 β 的范围设定为 $[0.1, 10]$ ，由此可知，旋转算子仍然更倾向于细微搜索，而平移因子更保持着一定大小的邻域来维持收敛率。

2.3.3 变尺度状态转移算法步骤

具有自适应参数的变尺度状态转移算法（State transition algorithm with variable-metric strategy, vSTA）的算法步骤如下：

Step 1: 初始化：在可行域内随机产生初始解 \mathbf{Best}_0 ；设置算法参数，包括

$\alpha = \alpha_{\max}, \alpha_{\min}, \beta = \beta_{\max}, \beta_{\min}, \gamma, \delta, fc$ 和 SE ; 设置 $k = 0, c = 0$;

Step 2: 伸缩变换: 基于当前最优解 $Best_k$ 用 ET 算子产生 SE 个候选解, 根据贪婪准则更新当前最优解, 如果有当前最优解有更新, 则用 TT 算子产生 SE 个候选解并更新最优解;

Step 3: 旋转变换: 基于当前最优解 $Best_k$ 用 RT 算子产生 SE 个候选解, 根据贪婪准则更新当前最优解, 如果有当前最优解有更新, 则用 TT 算子产生 SE 个候选解并更新最优解;

Step 4: 坐标变换: 基于当前最优解 $Best_k$ 用 AT 算子产生 SE 个候选解, 根据贪婪准则更新当前最优解, 如果有当前最优解有更新, 则用 TT 算子产生 SE 个候选解并更新最优解;

Step 5: 参数更新: 利用公式(2-1)更新旋转因子 α 及平移因子 β ;

Step 6: 终止: 如果 α 及 β 超过其边界值, 则等于边界值; 如果最大迭代次数或最小相对进步等终止条件未达到, 则使 $k = k + 1$; 否则终止, $Best_k$ 则是所求问题最优解。

2.4 基于变尺度状态转移算法的均匀划分动态优化方法

由于工业生产实际动态优化问题常具有多峰值、不连续的特性。基于 CVP 离散的数值动态优化方法较解析法更适合求解此类问题。CVP 方法提供了具有一定精度的动态优化问题离散化方法, 将无限维的时变控制曲线以若干参数进行拟合, 转化为 NLP 问题。由于工业动态系统的本身耦合非线性, 离散后 NLP 子问题仍继承了甚至放大了原问题的复杂非线性多模态特性。STA 作为新型智能优化算法, 不需要计算梯度信息, 不依赖于初始点, 全局搜索能力较强, 在处理此类问题上具有一定优势, 能更有效的求解到此 NLP 问题的全局最优解。

本章提出了基于变尺度状态转移算法的均匀划分动态优化方法 (udCVP-vSTA)。本方法采用数值方法, 先用控制参数化进行均匀部分离散, 再由状态转移算法迭代求解离散后得到的非线性子问题, 从而求解出动态优化问题的最优数值解。值得注意的是, 本章采用均匀、分段常数来拟合控制曲线。

其具体步骤如下:

Step 1: 初始化: 设置离散化网格段数 p ; 设置状态转移算法参数; 设置迭代终止条件, 即最大迭代次数及最小相对改进量;

Step 2: 参数化: 由于采用均匀分段离散化方案, 将时间区间均分成 p 段, 则 $\tau_1 = \tau_2 = \dots = \tau_p$, 时间节点为 $t_k = t_0 + (t_f - t_0) / p$ 。以分段常数为基函数拟合控制曲线, 因此, 决策变量为 $\mathbf{d} = [\xi_1, \dots, \xi_k, \dots, \xi_p]^T$, 其中 ξ_k 是第 k 段 $[t_{k-1}, t_k)$ 的控制值。

Step 3: 解动态方程: 将控制轨迹代入微分方程组, 用微分方程数值求解器

(常用 ode45) 求取相应时间段内状态变量曲线, 由此可获的此控制值下动态系统状态轨迹, 即, 获得此决策变量下的动态优化问题目标函数值等。

Step 4: 优化求解: 应用 vSTA 采样并更新最优解, 求取以上 NLP 问题。达到终止准则后, 得到此近似方案下的最优控制参数值, 结合离散网格可输出最优控制轨线。

其框架如图 2-8 所示:

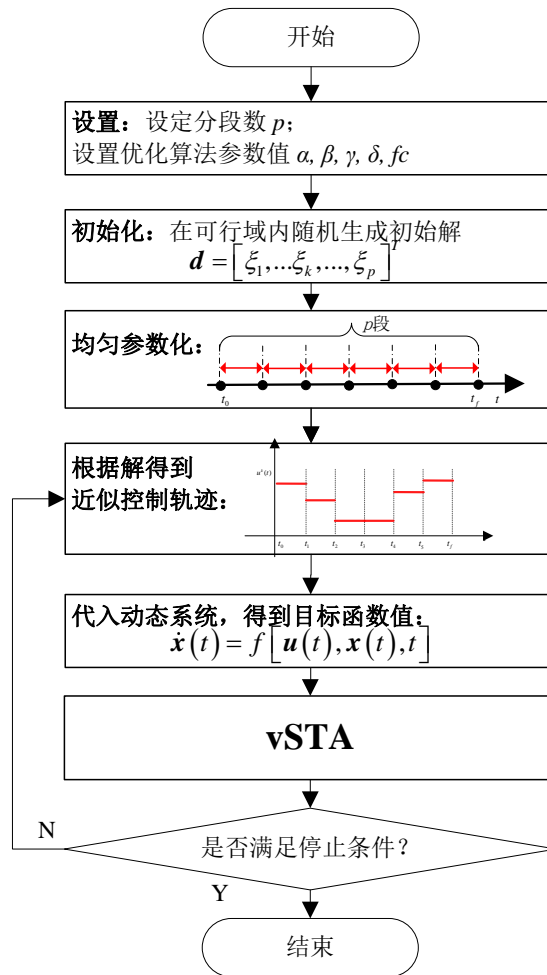


图 2-8 基于变尺度状态转移算法的均匀划分动态优化方法

2.5 实验结果与讨论

本节将介绍三个典型的具有奇异弧、多模态及切换控制等特性无约束动态优化问题, 用来验证基于状态转移算法的动态优化方法可行性和高效性。为了比较, 除了改进的状态转移算法 (vSTA), 实数编码遗传算法 (RCGA) [75], 人工蜂群算法 (ABC) [76]、自适应粒子群算法 (PSO) [77] 以及基本状态转移算法 (STA) 都将会用来求解动态优化问题。所有实现都在 MATLAB (R2016b 版本) 以及带有 3.4GHz intel i7 8G RAM 的台式电脑上运行, 选用 MATLAB 内置 ode45 函数

为微分求解器。

以下规定所有算法终止条件：启发式停止准则定义如下来检查目标函数的相对改进量：

$$\left| \frac{f(\mathbf{Best}_k) - f(\mathbf{Best}_{k-\Delta})}{f(\mathbf{Best}_k)} \right| \leq \varepsilon \quad (2-16)$$

其中 ε 为足够小的正数， Δ 指定了进步区间， $f(\mathbf{Best}_k)$ 是第 k 次迭代的最优值。除此之外，为了避免算法长时间陷入迭代，规定了最大迭代次数，当 $iter \geq \mathit{MaxIter}$ ， $\mathit{MaxIter}$ 为足够大的正数。通常 $\Delta = 20$; $\varepsilon = 10^{-6}$, $\mathit{MaxIter} = 100 \cdot \mathit{Dim}$ 。基本 STA 参数为默认参数，如表 2-2 所示。其余算法的基本参数均采用文献推荐取值或者经验取值，详细参数如下：

表 2-2 算法参数设置

算法	参数	值
vSTA	SE	10
	旋转因子 α	[1e-4,1]
	平移因子 β	[0.1,10]
	低阈值 C_1	4
	高阈值 C_2	9
GA	种群大小	20
	交叉率	0.95
	变异率	0.05, 0.1, 0.2
ABC	蜂群大小	30
	食物源数	0.5×蜂群大小
	限制数	100
PSO	粒子群大小	30
	惯性系数	[0.1,1.1]
	自调整权重	1.49
	社区调整权重	1.49
	最小邻居大小	0.25×粒群大小

将控制变量离散为 p 个区间，分别对不同 p 下的每种算法以随机初值进行 30 次独立运行，以减少算法随机性对结果的影响。采用几种常用的统计指标来评价该方法的性能，如 best 为最好值，mean 为平均值，worst 为最差值，std 为标准差，time 是 30 次独立运行的 CPU 运行时间平均值。

1) 案例一：管式反应器催化剂浓度分布问题

管式反应器催化剂浓度分布问题（Plug flow reactor catalyst blend problem）简称 PFR 问题，此问题旨在获得最佳的催化剂浓度沿管长分布曲线使得某种产物

产量最大化。在固定长度的某反应器中，发生着化学反应 $A \rightleftharpoons B \rightarrow C$ ，文献[78]中给出了描述反应物和产物浓度的系统动态方程组。将 x_a ， x_b 和 J 表示为物质 A, B, C 的摩尔分数，动态优化问题描述如下：

$$\begin{aligned} \max J &= 1 - x_A(t_f) - x_B(t_f) \\ \text{s.t.} \quad &\begin{cases} \dot{x}_A = u(k_2 x_B - k_1 x_A) \\ \dot{x}_B = u(k_1 x_A - k_2 x_B) - (1-u)k_3 x_B \\ k_1 = 1, k_2 = 10, k_3 = 1 \\ x_A(0) = 1, x_B(0) = 0 \\ 0 \leq u(t) \leq 1 \\ t_f = 12 \end{cases}, \end{aligned} \quad (2-17)$$

由以上数学描述可知，该动态优化旨在控制其中一种催化剂的浓度沿管长分布 $u(z)$ 使得最终产物 C 的浓度最大化。

在实验中，控制时间被离散化为 5、10、15、20 个子区间，在均匀常数近似方案下，控制曲线分别由 5、10、15、20 个参数组成。其每个算法独立运行 30 次的实验结果如表 2-3 所示，其最优控制曲线和状态曲线如图 2-9 所示，各文献对比结果如表 2-4 所示。

表 2-3 PFR 问题实验结果

pro.	alg.	p	best	mean	worst	std	time(s)
PFR (max)	vSTA	5	0.472599	0.472599	0.472599	1.206484e-09	5.099405
	STA		0.472599	0.472599	0.472599	1.149066e-07	7.917762
	GA		0.472599	0.472594	0.472530	1.252269e-05	11.116829
	ABC		0.472599	0.472599	0.472599	1.117669e-07	5.501882
	PSO		0.472599	0.472599	0.472599	8.557214e-09	5.763425
	vSTA	10	0.473630	0.473630	0.473630	9.716912e-08	15.329338
	STA		0.473630	0.473628	0.473625	1.294347e-06	17.288272
	GA		0.473630	0.473578	0.473441	4.568647e-05	26.030803
	ABC		0.473630	0.473630	0.473627	7.013694e-07	19.016008
	PSO		0.473630	0.473630	0.473630	1.211241e-07	16.582363
	vSTA	15	0.474531	0.474531	0.474530	1.029255e-07	33.564853
	STA		0.474530	0.474528	0.474521	1.712338e-06	38.032976
	GA		0.474531	0.474356	0.473995	1.176201e-04	40.849704
	ABC		0.474531	0.474530	0.474529	2.843568e-06	46.333104
	PSO		0.474531	0.474531	0.474530	1.827999e-07	36.512235
	vSTA	20	0.475272	0.475271	0.475270	4.846863e-07	51.646119
	STA		0.475271	0.475270	0.475270	4.419032e-07	57.225339
	GA		0.475272	0.475076	0.474530	2.203436e-04	135.502495
	ABC		0.475254	0.475268	0.475271	4.048923e-06	75.892490
	PSO		0.475272	0.475271	0.475266	1.091691e-06	61.126553

表 2-3 格列出了算法在不同分段区间时的对动态优化问题寻优结果，其中加粗部分意味着更好的结果。对于问题 1，所有的算法都能以较高的精度找到全局最优，而改进后的 STA 的执行时间在每一个分段数上均小于其余算法，表明改

进后的 STA 具有较高的效率。此外，由于改进后的 STA 拥有自适应变化的局部搜索参数，因而能在不影响全局搜索能力的同时，加强局部精细能力，在表中可体现为 LSTA 具有较好的均值、最好值、最差值和最小的标准偏差值。很明显可以看出，随控制分段数的增加，控制曲线拟合精度越高，优化算法耗时越长，但所求目标函数值精度越高。

表 2-4 PFR 问题文献对比

问题	文献	方法	段数	J
PFR (max)	Chen <i>et al.</i> [29]	HGPSO	5	0.47260
			10	0.47363
			20	0.47527
	Angira <i>et al.</i> [87]	TDE	20	0.475269
			40	0.476826
	Dadebo <i>et al.</i> [78]	IDP	20	0.475272
40			0.476946	

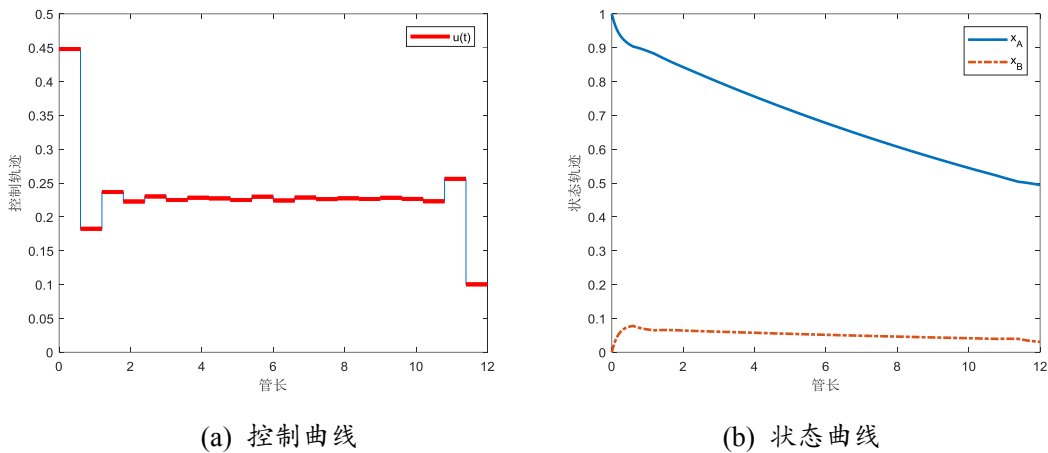


图 2-9 udCVP-vSTA 求解 PFR 问题所得最优结果 ($p=20$)

表格 2-4 为文献在分段均匀常数近似方案下的结果，IDP、TDE 以及 HGPSO 三种方法求解的最优结果。由此可知在同一分段数下 ($p=20$)，本方法所求结果与文献所求结果一致，且更高精度 ($J=0.475272$)。此外，研究也表明，在分段数较大的情况下 ($p=40$)，所求目标函数值越优 ($J=0.47695$)。

2) 案例二：CSTR 反应器冷却液问题

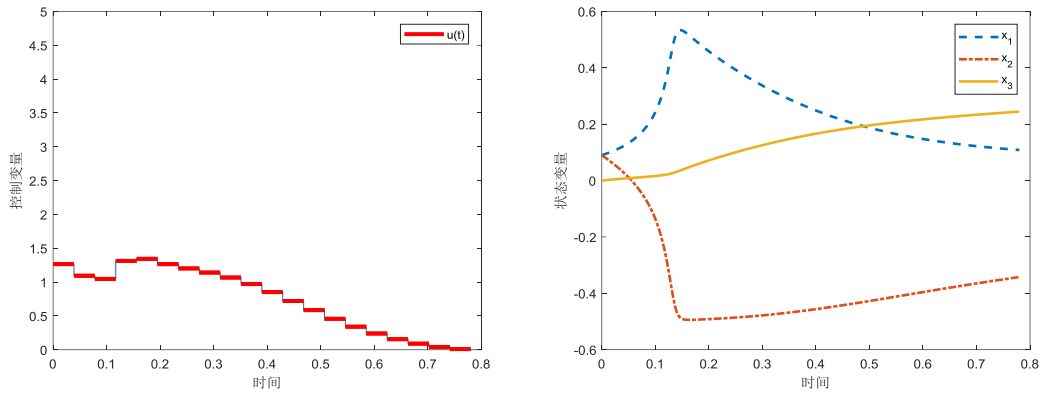
CSTR 反应器冷却液问题的非线性微分方程由其热质平衡决定。此问题动态优化模型如 (2-18) 所示，其中 $u(t)$ 为冷却液加料流率， x_1 代表无量纲稳态温度的偏差， x_2 表示与无量纲稳态浓度的偏差。此问题优化目标为获得最佳冷却液控制曲线 $u(t)$ ，在尽量少使用冷却液的情况下，使温度 x_1 和浓度 x_2 的累积变化最小。

连续搅拌反应器 (CSTR) 优化是 Cruz 研究的一个多模态动态优化问题^[79]，研究表明，用基于梯度的优化算法求解时极易陷入局部极值 0.24，其局部极值控

制曲线和状态曲线如图 2-9 所示。

$$\begin{aligned}
 & \min J = x_3(t_f) \\
 & s.t. \begin{cases} \dot{x}_1 = -(2+u)(x_1+0.25) + (x_2+0.5)e^{25x_1/x_1+2} \\ \dot{x}_2 = 0.5 - x_2 - (x_2+0.5)e^{25x_1/x_1+2} \\ \dot{x}_3 = x_1^2 + x_2^2 + 0.1u^2 \\ x(0) = [0.09, 0.09, 0] \\ 0 \leq u(t) \leq 5 \\ t_f = 0.78 \end{cases} \quad (2-18)
 \end{aligned}$$

在实验中，控制时间被离散化为 10、15、20 个子区间。其每个算法独立运行 30 次的实验结果如表 2-5 所示，其最优控制曲线和状态曲线如图 2-10 所示。



a) 局部最优控制曲线

b) 局部最优状态轨迹

图 2-10 CSTR 问题局部最优解

表 2-5 CSTR 问题实验结果

pro.	alg.	p	best	mean	worst	std	time(s)
CSTR (min)	vSTA	10	0.137258	0.137258	0.137261	8.909569e-07	20.365621
	STA		0.137258	0.137290	0.137385	2.990238e-05	24.974895
	GA		0.137259	0.137318	0.137620	1.018687e-04	53.317429
	ABC		0.137291	0.137838	0.140144	6.220728e-04	45.113475
	PSO		0.137258	0.137332	0.137994	2.244053e-04	20.716706
	vSTA	15	0.134969	0.134970	0.134971	4.285284e-07	44.395781
	STA		0.134971	0.135002	0.135079	2.440593e-05	50.962329
	GA		0.134969	0.135015	0.135279	8.640548e-05	105.488990
	ABC		0.135173	0.136301	0.140907	1.172842e-03	95.618774
	PSO		0.134969	0.134995	0.135376	8.205750e-05	49.241551
	vSTA	20	0.134156	0.134156	0.134157	4.002515e-07	75.504284
	STA		0.134160	0.134188	0.134299	3.192759e-05	85.426558
	GA		0.134157	0.134674	0.138198	9.767036e-04	120.017122
	ABC		0.134363	0.135418	0.138694	8.861394e-04	142.202570
	PSO		0.134155	0.134219	0.135289	2.532343e-04	81.210973

表格 2-5 列出了算法在不同分段区间时的动态优化问题寻优结果，其中加粗部分意味着更好的结果。我们可以看到改进后的 STA 展现出了优秀的全局搜索性能和局部精细性能。从平均执行时间的角度上看，改进的 STA 比 PSO, GA 和 ABC 算法更快。同时，由于其 30 次求解的最差值、标准差均优于其余算法，vSTA 展现了更准确且稳定的解。此外，vSTA 获得的平均值总是优于其他结果，这表明其优越的稳定性和收敛性能。因此，vSTA 的自适应变换因子不仅可以促进全局探索，还可以提高算法的求解精度和稳定性。

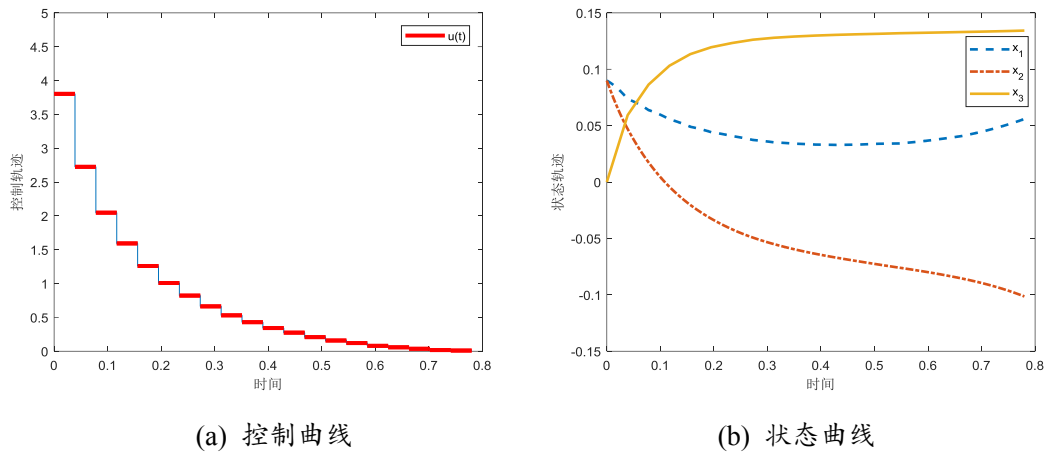


图 2-11 改进 STA 求解 CSTR 问题所得最优结果 ($p=20$)

表 2-6 CSTR 问题文献对比

问题	文献	方法	段数	J
CSTR (min)	Chen <i>et al.</i> [29]	HGPSO	5	0.14916
			20	0.13416
	Cruz <i>et al.</i> [46]	DE	13	0.13559
			IDP	13
Bayón <i>et al.</i> [88]	PMP	78	0.1334	

表格 2-6 为分段均匀常数 CVP 近似方案下，文献中 HGPSO、DE 等方法求解的最优结果。由此可知在同一分段数下 ($p=20$)，本方法所求结果与文献所求结果一致，且更高精度 ($J=0.134156$)。

此外，从表 2-6 可以看出， $p=10$ 时， $J=0.137258$ ， $p=20$ 时， $J=0.134156$ ，目标函数值下降了 0.003，但优化时长却增加了 4 倍左右。随控制分段数的增加，控制曲线拟合精度越高，优化算法耗时越长，但所求目标函数值精度越高。在分段数较大的情况下 ($p=40$)，可以达到了 $J=0.13336$ 。在 $p=78$ 时，最优目标函数值甚至可以提升到 0.1334。由此可知，在均匀常数近似的 CVP 方案下，动态优化问题求解精度和求解效率两者互为矛盾，提高求解精度势必会带来需要优化的变量个数的增加，而优化变量维数的增加又定会增加优化时间和优化难度，考验

优化算法优化性能。在高分段数下即耗费大量计算资源，还有可能不易获得更高精度的控制曲线。

3) 案例三：Nishida 问题

Nishida 问题是一个典型的 bang-bang 控制问题^[18]，其动态问题模型如公式 2-19 所示。

在实验中，控制时间被离散化为 10、15、20、30 个子区间，在均匀常数近似方案下，控制曲线分别由 10、15、20、30 个参数组成。其每个算法独立运行 30 次的实验结果如表 2-7 所示，其最优控制曲线和状态曲线如图 2-11 所示。

$$\begin{aligned} \min J &= \sum_{i=1}^4 x_i^2(t_f) \\ \text{s.t.} \begin{cases} \dot{x}_1 = -0.5x_1 + 5x_2 \\ \dot{x}_2 = -5x_1 - 0.5x_2 + u \\ \dot{x}_3 = -0.6x_3 + 10x_4 \\ \dot{x}_4 = -10x_3 - 0.6x_4 + u \\ x(0) = (10, 10, 10, 10) \\ -1 \leq u(t) \leq 1 \\ t_f = 4.2 \end{cases} \end{aligned} \quad (2-19)$$

表 2-7 Nishida 问题实验结果

pro.	alg.	p	best	mean	worst	std	time(s)
Nishida (min)	vSTA	10	1.671318	1.671318	1.671318	6.775215e-16	6.172973
	STA		1.671318	1.671318	1.671318	6.775215e-16	14.850102
	GA		1.671318	1.671320	1.671339	4.147524e-06	33.101814
	ABC		1.671318	1.671466	1.673391	4.609777e-04	15.687718
	PSO		1.671318	1.671318	1.671318	6.775215e-16	13.920187
	vSTA	15	1.234664	1.234664	1.234664	4.516810e-16	13.329868
	STA		1.234664	1.234664	1.234664	4.516810e-16	32.392231
	GA		1.234665	1.234698	1.234812	3.442074e-05	84.569890
	ABC		1.234664	1.235295	1.246156	2.121407e-03	34.367538
	PSO		1.234664	1.234664	1.234664	4.516810e-16	35.345246
	vSTA	20	1.453320	1.453343	1.453406	3.602028e-05	60.062419
	STA		1.453320	1.453342	1.453406	3.278794e-05	62.536712
	GA		1.453321	1.453439	1.453946	1.477015e-04	123.988301
	ABC		1.453321	1.454066	1.465796	2.346378e-03	66.855250
	PSO		1.453320	1.455191	1.509291	1.021788e-02	54.750744
	vSTA	30	1.148940	1.148940	1.148947	1.628926e-06	118.738517
	STA		1.148940	1.148940	1.148948	1.564072e-06	142.000548
	GA		1.149030	1.152706	1.168031	4.234865e-06	181.489771
	ABC		1.148940	1.150195	1.161845	3.120814e-3	136.556515
	PSO		1.148940	1.150092	1.157654	2.988237e-3	197.218277

表格 2-7 列出了算法在不同分段区间时的动态优化问题寻优结果，其中加粗

部分意味着更好的结果。我们可以看出两个 STA 在分段数 $p=10, 15$ 的时候都能达到很高的求解精度和稳定性，但是改进后的 STA 要更加快速。但是在 $p=20, 30$ 时，改进后的 STA 继续展现出了较高的收敛速度，但是算法稳定性却略微低于基本 STA。

表格 2-8 为分段均匀常数 CVP 近似方案下，文献中 HGPSO、牛顿法等方法求解的最优结果。由此可知在同一分段数下 ($p=15$)，本方法所求结果与文献所求结果 ($J=1.23469$ 或 $J=1.23469$) 一致，且更高精度 ($J=1.234664$)。

表 2-8 Nishida 问题文献对比

问题	文献	方法	段数	J
Nishida (max)	Chen <i>et al.</i> [29]	HGPSO	10	1.67163
			15	1.23469
	Vassiliadis <i>et al.</i> [89]	Newton 法	10	1.67163
			15	1.23469

值得注意的是，由表格 2-7 可知，对于此问题，当分段数 $p=10, 15, 20, 30$ 依次增大，最优目标函数值并不随之依次变优，分别为 1.671318, 1.234664, 1.453320, 1.148940，在分段数为 20 时最优值反而较分段数 15 的最优值有所下降，这说明拟合精度在一定程度上并不一定与分段数成正比。由图 2-11 分析可知，当均匀分段后的切换时间节点与 bang-bang 控制的切换时间节点一致时，才能有较高的拟合精度，但通常来说，一个好的近似化方案是事先未知的，在均匀分段的情况，可能需要大量的实验来反复验证，以定位到合理的切换时间点。

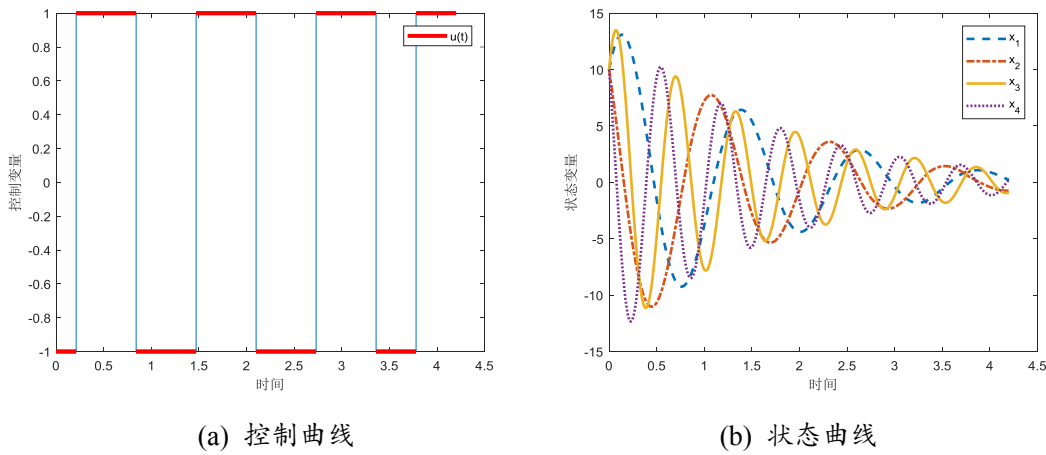


图 2-12 改进 STA 求解 Nishida 问题所得最优结果 ($p=15$)

2.6 本章小结

针对实际动态优化问题的复杂非线性特性，本节提出了基于变尺度状态转移算法的均匀划分动态优化方法 (udCVP-vSTA)。

本节首先采用均匀划分控制向量参数化方法将无限维动态优化问题近似为

有限维的 NLP 问题, 因此高效的 NLP 求解算法是动态优化方法进行离线或在线应用的关键。进一步地, 介绍了新型智能优化算法 STA, 此算法根据其拥有特殊邻域的状态变换算子来转移到下一最优解, 较梯度法有全局性好、不需要计算梯度或 Hessian 矩阵的优势。此外, 通过分别分析 STA 四个算子的全局和局部性能, 针对 STA 算子参数存在固定或无规则更新的情况, 提出了启发式参数自调节策略, 用以自适应调节局部搜索尺度; 最后将变尺度 STA 作为求解子 NLP 问题的基础算法, 建立基于变尺度 STA 的均匀划分动态优化方法。在均匀划分的 CVP 框架下, STA 用于三个经典优化问题的求解。与基本 STA 等其他智能优化算法相比, 多尺度 STA 展现出较好的全局性和局部收敛性。与其他文献结果相比, 基于 STA 的动态优化方法有着更高精度和鲁棒性。同时, 结果分析也显示了基于均匀分段的 CVP 方法在求解精度与效率上的不足, 这为后文的改进研究提供了动机。

3 基于混合状态转移算法的非均匀划分动态优化方法

工业生产过程的多变性和实时性需求, 要求动态优化方法能高效、可靠、精确及灵活地处理不同结构的实际动态优化问题。基于均匀划分控制参数化的动态优化方法具有编程易实现、实际控制器易操作的优点, 但其求解精度和求解效率两者互为矛盾: 一方面, 过于粗糙的时间网格划分将使得动态优化解近似程度低、精度差, 而提高求解精度势必会带来优化变量维数的增加, 导致参数化后得到的静态数学规划问题规模巨大甚至病态, 这给离散后的 NLP 子问题增加优化难度及计算代价。另一方面, 过细的时间网格划分将带来过参数化问题, 过参数化问题使得控制变换频繁且剧烈抖动, 不利于工厂执行器设备维护及过程工况稳定。

传统等间隔方式均匀划分的控制参数化不能快速准确的定位到最优控制切换时间节点, 难以高效高精度地获得具有奇异弧的控制曲线。因此, 本章研究一个非均匀划分策略, 能获得合理的控制时间网格来契合原最优控制曲线, 并以较少的优化参数高效获得高精度解。

3.1 非均匀划分控制向量参数化策略

传统数值解法通常采用分段均匀离散方案, 当分段数一定时, 网格划分的疏密也一定, 且其在整个优化过程中维持不变。均匀分段数的大小及网格点多少将直接影响近似化曲线逼近原始最优控制曲线的程度, 段数越少, 优化参数越少, 但控制逼近效果越差。相反, 大量的网格点将导致过度参数化、低效的控制逼近和敏感性问题。此外, 对于 bang-bang 控制问题也很难定位到最佳切换点, 导致解精度有限, 或者操作复杂。假设某 bang-bang 问题 $t_f = 5$, 其最优控制曲线如下图所示, 其最优控制曲线数学表达式为

$$u^*(t) = \begin{cases} 2, & t \in [0, 4.2) \\ 1, & t \in [4.2, 5] \end{cases}, \quad (3-1)$$

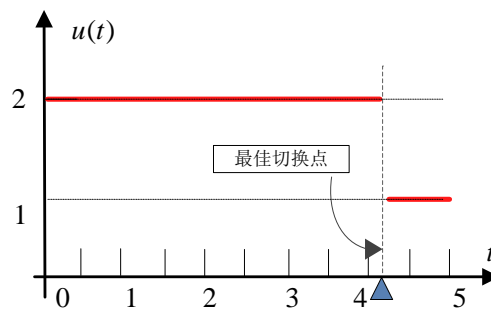


图 3-1 某 bang-bang 问题最优控制轨迹

通常来说，一个好的近似化方案是事先未知的。在均匀分段的情况下，由图可知，不管是分段数为 5 或是分段数为 10，都无法定位到最佳切换点，无法最好的拟合原始最优控制曲线。很自然的联想到，如果可以实现非均匀离散，则分段数为 2 即可完成原始最优控制曲线的拟合，因此有许多学者对非均匀离散化 CVP 方案做出研究。

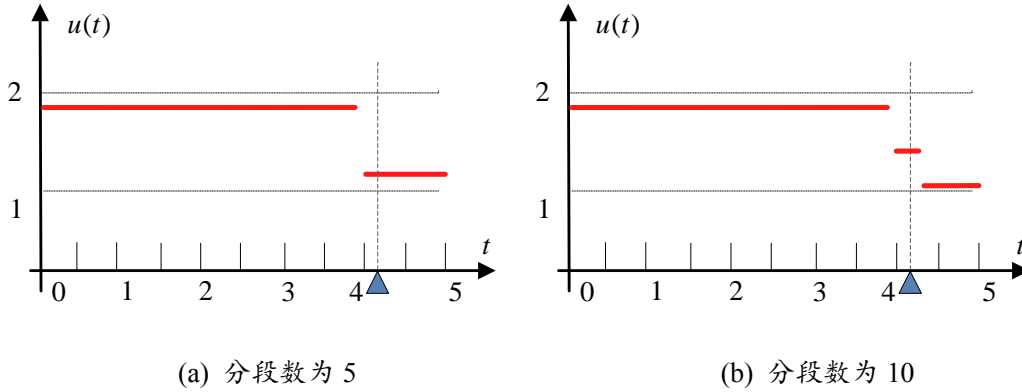


图 3-2 不同均匀分段数下的某 bang-bang 问题最佳拟合控制曲线

本节首先总结了两种非均匀离散化方法的思想——时间点移动和时间点增删，分析了两种方法的优劣，随后，提出了一种基于部分移动时间点的自适应 CVP 策略，综合了两种非均匀离散化的思想。下面具体介绍。

3.1.1 网格点移动与增删策略分析

关于如何选择最优的非均匀离散化方案已经引起了诸多学者的广泛讨论，其中大概分为两大类，第一类为离散时间点的移动（固定网格点数目），第二类为离散时间点增删（不固定网格点数目）。

1) 离散时间点移动方面：此类方法通常先确定离散时间点个数，再通过移动时间点来寻求最优离散化方案。在文献[81][82]中，为了找到最优的离散方案，除了各离散化子区间的控制参数，各个转换时间点也成为了决策变量一同优化， $\mathbf{d} = [\xi_1, \dots, \xi_p, t_1, \dots, t_p]$ ，其中 ξ_k 与 $t_k, k = 1, \dots, p$ 分别为控制值和转换时间点， p 为离散段数，此方法提高了问题优化维度，加剧了问题非线性，同时额外增加了 p 个时间点约束 $t_1 < t_2 < \dots < t_p$ 。在文献[83][84]中，Teo 等提出了 Time-scaling 方法，该方法把原始离散时间点映射到一个新的时间尺度上，并不直接优化转换时间点，而是优化时间子区间长度，则 $\mathbf{d} = [\xi_1, \dots, \xi_p, \tau_1, \dots, \tau_p]$ ，其中 $\tau_k = t_k - t_{k-1}, k = 1, \dots, p$ ，此方法将 p 个时间约束转换为一个总时长约束， $\sum_{k=1}^p \tau_k = t_f - t_0$ 。在文献[80]中，陈旭等人同样将时间子区间长度作为额外优化对象，并采用线性映射的方法进行编码，有效处理了时间节点约束，其中 $\mathbf{d} = [\xi_1, \dots, \xi_p, r_1, \dots, r_p]$ ，其中 $r_k \in [0, 1]$ ，则

每个时间子区间长度为 $t_k - t_{k-1} = \left(r_k / \sum_{k=1}^p r_k \right) (t_f - t_0)$, $k = 1, \dots, p$ 。综上, 此类方法通常先确定离散时间点数目或分段数数目, 再将各时间点位置或子时间段长度作为待优化决策变量, 与对应子区间控制参数值一同优化。

下图为离散时间点的优化, 下图为离散时间点变化后对最优控制区间的拟合情况, 原始最优控制曲线为深蓝色连续实线。

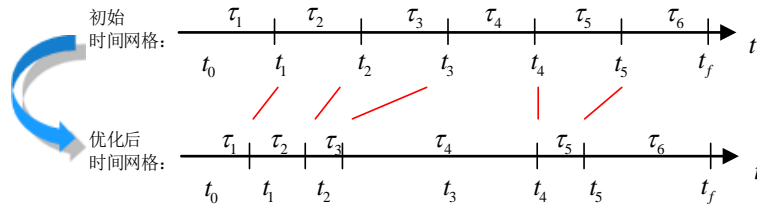


图 3-3 离散时间点移动 ($p=6$)

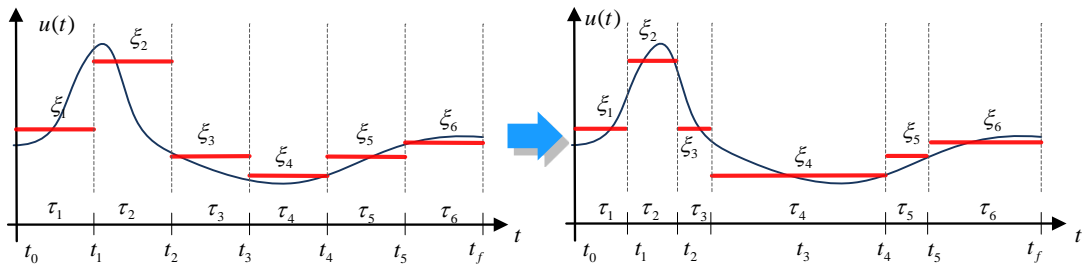


图 3-4 离散时间点移动前 (左) 与移动后 (右) 的最优控制参数图 ($p=6$)

综上, 通过离散时间点移动来寻求最优离散化方案优势在于: 1) 可以得到此分段数下最优控制结果。主要的缺点在于: 1) 得到的 NLP 问题往往是强非线性且难以解决。特别是, 使用此方法即使是线性的动态优化问题也将导致参数化后的 NLP 问题强非线性; 2) 由于离散点个数往往事先确定, 整个优化过程能到达的拟合精度也有上限; 3) 优化问题维数是同一分段数下均匀离散的两倍, 如拟将控制时间分为 40 段, 则待优化参数为 40×2 , 即 80 维, 而计算量随决策变量的数量呈多项式增长。

2) 离散时间点增删方面: 此类方法通常先给定一个粗糙控制网格, 然后以此优化结果为基础再根据某种规则增加、移除网格节点, 实现精细化的同时尽量减少分段数的剧增。单纯增加点的方法为均匀二分细化方法, 该方法从较少分段数开始优化求解, 如分段数为 4, 则下一次优化的控制网格为每一段均匀分为两段, 则分段数增至 8, 再次优化求解直到更多细化不能带来目标改进为止, 但此方法盲目增加分段数, 易导致过度参数化。Waldraff 等人^[85]检查网格点对优化问题的改进程度来移除点, 并且通过控制曲线曲率信息来确定新增点位置。Marquardt 等人先后在文献[27]和文献[28]中提出基于小波分析的离散方法, 此方法首先对在粗糙网格所得最优控制曲线进行小波分析, 根据小波因子的大小来判

断此控制曲线的平缓程度，并以此决定插入和删除点，并以之前最优控制曲线作为初始解在新的离散网格下进行优化。类似的，李国栋等人在文献[30]中提出基于斜率的自适应 CVP 方法，以两个相邻分段常数控制参数的子区间时长和参数值求得离散网格时间点的左右斜率，由此增删网格时间点。刘平等人在文献[31]中提出了基于经验模态 (EMD) 分解时间网格重构的 CVP 算法，其利用 EMD 分解上一次迭代所得控制曲线可得到节点相邻趋势信息的差值，若此差值过小，则消除此时间节点，若差值较大，则将此一时间子区间细分为四，否则仅一分为二。王立伟等人在文献[86]提出了基于敏感度的自适应 CVP 方法，若在某一时间区间内，第 $l-1$ 次和第 l 次所得的最优控制参数值的差值小于某一阈值，则此消除区间节点，若大于某一阈值则保留节点或进行时间区间二等分。

下图 3-5 为离散时间点的由第一代经插入和删除操作后得到第二代离散网格，分段数由 6 段变为 8 段，下图为离散网格变化后对最优控制区间的拟合情况，原始最优控制曲线为深蓝色连续实线。

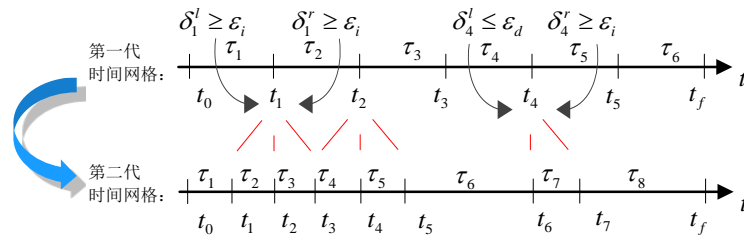


图 3-5 离散时间点增删示意图 ($p=6, p^*=8$)

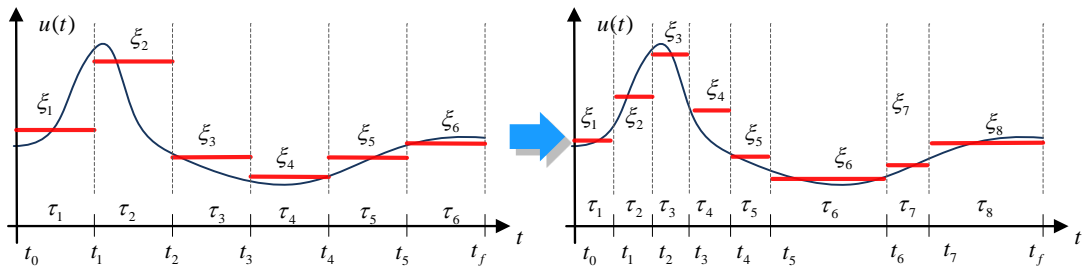


图 3-6 离散时间点增删前 (左) 与增删后 (右) 的最优控制参数图 ($p=6, p^*=8$)

综上，此类方法需要根据上一次最优控制参数结果来决定下一次优化的离散网格，不断精细化，理论上可以用相对较少的时间节点逼近原最优控制曲线。此方法优势如下：1) 每次迭代网格点都是固定的，不参与优化；2) 上一次迭代优化的最优解可作为下一次迭代的初始值，实现热启动。但此方法劣势如下：1) 下一次的离散网格完全依赖于上一次的优化结果，因此对每一次迭代的优化精度有要求，否则会合并或插入错误的网格点；2) 需要不断迭代优化，耗时长，且中间的合并与插入等操作也耗费计算资源；3) 不论是基于曲率、小波系数、斜率、模态差值或是敏感度，其均需要设定相应的上下界阈值判断是否合并、移除

或是二等分、四等分，上下阈值设定主观随意性大，较难合理适应各种控制曲线结构。

3.1.2 基于部分移动网格优化的非均匀划分策略

在分段常数近似化方案下，为获得较高的近似化精度，需要有适合其控制曲线形状的离散网格。由上一节分析可得，通过离散时间点移动来求取最佳离散化网格通常导致问题维数过高、非线性太强，从而难以求解，此外还有网格点数目固定导致的精度上限问题；通过离散时间点增删来一步一步迭代逼近最佳网格会面临效率低、主观参数过多等问题；因此，本节拟吸取离散时间点移动和离散网格点增删方法的优点，设计一种高效、易于实现、精确的非均匀离散化动态优化方法。

此方法以均等分的粗糙网格为基础，将所有时间子区间进行非均匀二分优化，用以定位最优控制切换时间点，其非均匀二等分点与所有时间子区间对应的控制参数值均为待优化变量。因为只有部分网格点在移动，因此称此策略为部分移动网格点策略。其示意图如下，其中红色坐标线为可移动时间节点，黑色坐标线为固定时间节点。第一步，在均匀离散方案下，时间区间被分成长度相等的 p 段，其时间节点为：

$$t_k = t_0 + k \cdot \frac{(t_f - t_0)}{p}, k = 0, \dots, p-1, \quad (3-2)$$

第二步，经过非均匀二分后，中间时间插入时间节点为，

$$t_k^{in} = t_k + \theta_{k+1} \cdot \frac{(t_f - t_0)}{p}, k = 0, \dots, p-1, \quad (3-3)$$

其中 $\theta = [\theta_1, \dots, \theta_p]$ 为分数时间点， $\theta_i \in [0, 1], i = 1, \dots, p$ 。

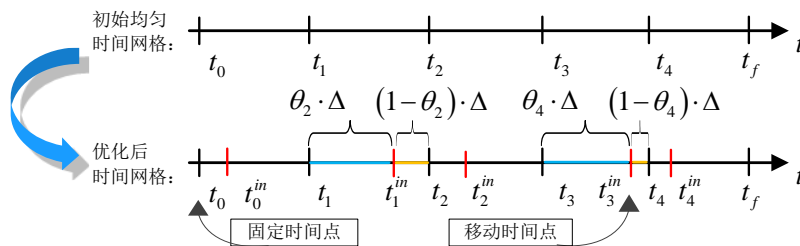


图 3-7 部分移动网格策略示意图 ($p=5, p^*=10$)

此方法特点如下：(1) 与传统网格增删法不一致的是，此处并不是考虑基于各种阈值的均匀二等分，为寻得最优的控制网格位置，仅将最佳二分点考虑为待优化变量，并引入分数编码方式，使得决策变量仅包括控制参数值 $\xi = [\xi_1, \dots, \xi_{2-p}]$

以及分数时间点 $\theta = [\theta_1, \dots, \theta_p]$, 从而决策变量为 $d = [\xi_1, \dots, \xi_{2 \cdot p}, \theta_1, \dots, \theta_p] = [\xi, \theta]$ 。由于是自适应的进行二分, 省去了传统网格增删法的反复迭代过程。(2) 与传统网格移动法不一致的是, 此方法并没有试图移动全部节点, 原有等分粗糙网格点不移动, 仅仅移动中间二分时间点, 因而所得 NLP 维数为 $3 \cdot p$ 维, 比传统网格移动法的 NLP 问题减少了 p 维决策变量。

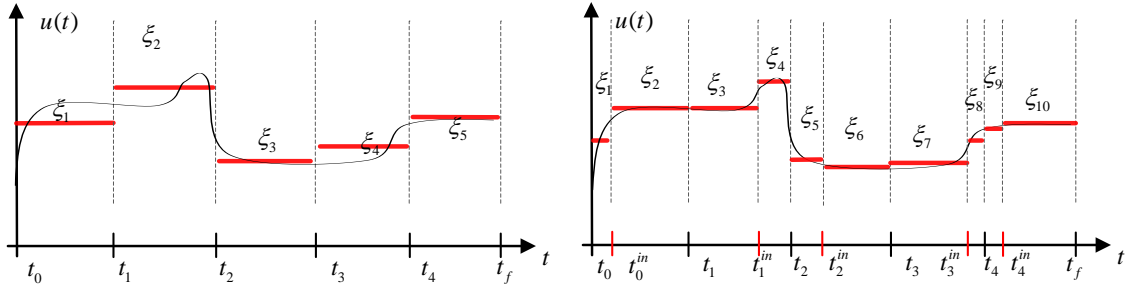


图 3-8 均匀 (左) 与部分移动 (右) 离散时间网格下的最优控制参数图

3.2 混合状态转移算法

由于还需要同步优化时间节点, NLP 问题的非线性程度得到增强, 使用不需要梯度信息、不易陷入局部的状态转移算法成为了求解此 NLP 问题很好的选择。但同时, 由于 NLP 问题的维数也相应变高。由上一章实验结果分析可知, 高维问题使得包括 STA 算法在内进化算法收敛速度低下, 无法适用于在线的动态优化及时、精确的要求。为了高效精确获得 NLP 问题的全局最优解, 本节设计一种混合梯度的状态转移算法, 平衡算法全局和局部收敛能力。

3.2.1 梯度搜索算法

基于梯度的搜索算法具有收敛速度快, 局部精细化能力强的特点。牛顿法是经典的用于求解无约束优化问题的经典算法之一, 也是许多较常用的梯度算法如 BFGS 算法、信赖域算法的基础, 其基本思想是在迭代点处对目标函数进行二次泰勒展开, 可得到如下二阶模型:

$$q(x) \approx f_k + g_k^T (x - x_k) + \frac{1}{2} (x - x_k)_k^T H_k (x - x_k)_k \quad (3-5)$$

其中 $H_k = \nabla^2 f(x_k)$, $g_k = \nabla f(x_k)$, $f_k = f(x_k)$ 。令 $s_k = x - x_k$, 当 x_k 确定时, $q(x)$ 是关于 s_k 的二次函数, 则对 s_k 求导可得极值点, 可得如下

$$H_k s_k = -g_k \quad (3-6)$$

以上线性方程的解 s_k 即为牛顿步长, $x_k + s_k$ 即为下一代点。就这样, 把二次近似模型的极值点作为新的迭代点, 不断重复直到满足停止准则。牛顿法具有局部二次收敛性, 因而局部收敛速度较快。牛顿法基本步骤如下:

- Step1: 随机初始化 x_0 , 令 $iter = 1$;
- Step2: 计算梯度 $g_{iter} = \nabla f(x_{iter})$, 若 $\|g_{iter}\| \leq \varepsilon_{opt}$, 则终止, 输出局部最优解 x_{iter} ;
- Step3: 计算 Hessian 矩阵 $H_{iter} = \nabla^2 f(x_{iter})$, 根据 $H_{iter} d = -g_{iter}$ 求步长 d_{iter} ;
- Step4: 令 $x_{iter+1} = x_{iter} + d_{iter}$, 令 $iter = iter + 1$, 重返 Step2;

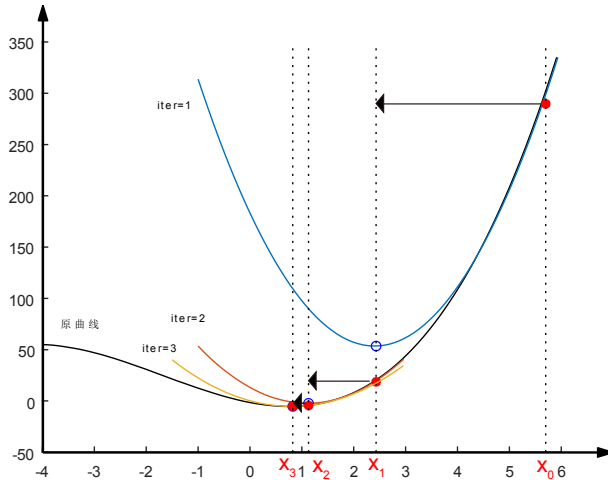


图 3-9 牛顿法迭代优化示意图

如图 3-9 所示, 优化点以 x_0 为初始点, 在此处进行二阶泰勒展开后生成蓝色二次曲线, 以此二次曲线极值解为最优解更新至 x_1 , 继续在此处进行二阶泰勒展开, 并更新至 x_2 , 同理更新至 x_3 , 经测算 $\|g_3\| \leq \varepsilon_{opt}$, 即此处梯度值小于指定精度, 则判断已经达到曲线极值点。通常 $\varepsilon_{opt} = 10^{-6}$ 。

基于梯度的优化方法, 快速收敛可能导致搜索朝着局部最优和过早解的方向发展, 特别是应用于非连续、不可微和多模态的非均匀离散下的动态优化问题时, 易过早陷入局部极值或过早终止搜索, 因此可利用其强健的局部收敛能力, 将梯度搜索算法应用于状态转移算法寻优后期。

3.2.2 混合状态转移算法

状态转移算法是一种基于随机搜索的方法, 它具有结构简单、全局搜索能力强、通用性强、鲁棒性强等特点。然而, 如图 3-10, 我们经常观察到的一个有趣的现象是, 随着迭代次数的增加, STA 的目标增量改进会迅速减少。换言之, 当当前最优解接近全局最优解时, STA 的计算将很大概率的带来冗余的计算量, 以及目标函数的微小改进。因此, 由于梯度方向的利用, 基于梯度的算法能够准确、快速地找到局部 (或全局) 最优解。因此, 为了加快局部搜索阶段的收敛速度, 本节提出了一种结合梯度优化方法的混合状态转移算法 (HSTA)。

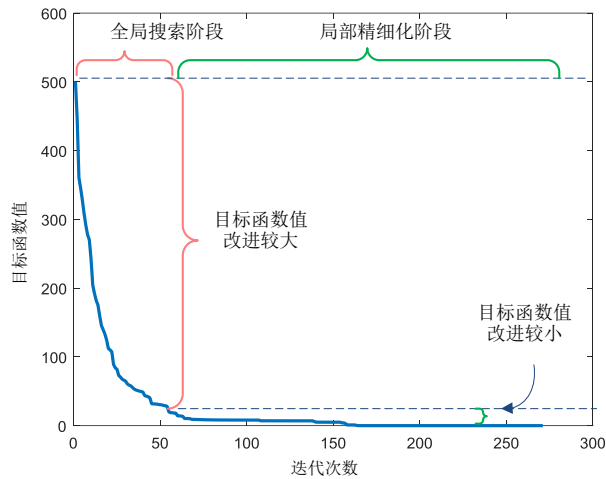


图 3-10 STA 迭代曲线与目标函数值示意图

HSTA 分为两阶段，为追求全局性搜索，在第一阶段使用全局优化算法，状态转移算法，寻求到 NLP 问题的全局最优解附近，在第二阶段使用梯度算法 BFGS 进行精细搜索，并使用前一阶段 STA 最优解作为初始解，为梯度算法提供较优初始解。其算法流程图如图 3-11 所示。

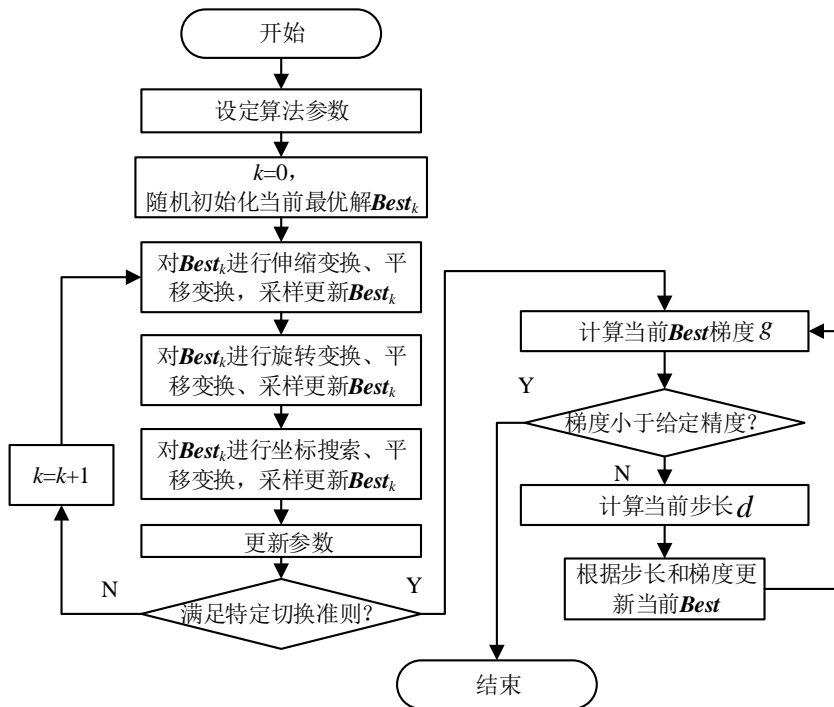


图 3-11 HSTA 算法流程图

HSTA 采用启发式准则来终止 STA 全局搜索，切换至 BFGS 局部搜索。如下，定义每次迭代间的相邻改进率为 R_{impro} ，

$$R_{impro} = \left| \frac{f(\mathbf{Best}_k) - f(\mathbf{Best}_{k-1})}{\max(1, f(\mathbf{Best}_k))} \right|, \quad (3-7)$$

其中, \mathbf{Best}_k 是第 k 次迭代的当前最优解, $f(\mathbf{Best}_k)$ 是当前最优解对应的函数值, 表示当前最好值。由于算法的随机性, 当 R_{impro} 连续 $counter$ 次以上满足 $R_{impro} \leq \varepsilon_s$ 时, 则判断 STA 搜索已经结束了全局定位阶段, 已进入了精细化搜索阶段。通常 $counter = 5; \varepsilon_s = 10^{-3}$ 。

3.3 基于混合状态转移算法的非均匀划分动态优化方法

为了在较少拟合参数的情况达到较高的曲线拟合精度, 并定位到合适的切换时间点, 本章提出了基于部分移动网格离散策略的非均匀 CVP 方法 (ndCVP); 同时, 为了能精确高效地获得全局最优轨迹, 本章使用混合状态转移算法(HSTA)求解经此离散后的强非线性 NLP 子问题。

考虑一般无约束动态优化问题, 其表达式如下:

$$\begin{aligned} \min_{\mathbf{u}(t)} \quad & J = \Phi[\mathbf{x}(t_f)] + \int_{t_0}^{t_f} L[t, \mathbf{x}(t), \mathbf{u}(t)] dt \\ \dot{\mathbf{x}}(t) = & f[\mathbf{u}(t), \mathbf{x}(t), t] \\ \mathbf{x}(t_0) = & \mathbf{x}_0 \\ \mathbf{u}^l \leq & \mathbf{u}(t) \leq \mathbf{u}^u \\ t_0 \leq & t \leq t_f \end{aligned} \quad (3-8)$$

其中, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ 为 n_x 维状态变量, 系统的初始状态为 \mathbf{x}_0 , $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ 为 n_u 维控制变量; 目标函数 J 为预定的一个性能指标。控制变量必须满足 $\mathbf{u}^l \leq \mathbf{u}(t) \leq \mathbf{u}^u$; 终止时间 t_f 为固定常数。

因此, 分段常数非均匀, 即部分移动网格近似方案下, 且假设 n_u 个控制轨迹均遵循同一个离散化方案, 则原动态优化问题目标函数可离散化为以下 NLP 问题:

$$\begin{aligned} \min_{\xi, \theta} \quad & J' = \Phi[\mathbf{x}(t_f | \xi)] + \sum_{k=1}^p \int_{t_{k-1}}^{t_k} L[t, \mathbf{x}(t | \xi), \xi^k] dt \\ \dot{\mathbf{x}}(t) = & f[\xi_i, \mathbf{x}(t), t] \\ \mathbf{x}(t_0) = & \mathbf{x}_0 \\ \mathbf{u}^l \leq & \xi_i \leq \mathbf{u}^u, i = 1, \dots, 2 \cdot p \\ 0 \leq & \theta_k \leq 1, k = 1, \dots, p \\ t_0 \leq & t \leq t_f \end{aligned} \quad (3-9)$$

其中决策变量为 $\mathbf{d} = [\xi_1, \dots, \xi_{2 \cdot p}, \theta_1, \dots, \theta_p] = [\xi, \theta]$, 维数为 $2 \cdot n_u \cdot p + p = (2n_u + 1)p$ 。

动态优化问题转变为：找出 $(2n_u + 1)p$ 维向量 $[\xi, \theta]$ 使得目标函数 J' 最小，其中决策变量中控制参数值仍需满足原始约束，新增变量 $\theta \in [0, 1]$ 。

因此，基于混合状态转移算法的非均匀划分动态优化方法框架如下图所示：

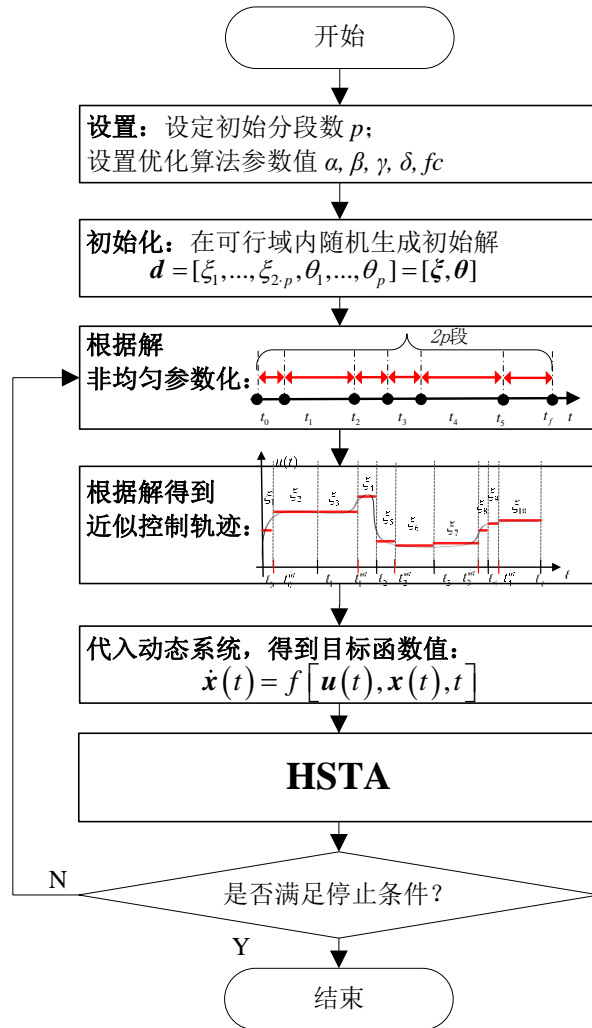


图 3-12 基于混合状态转移算法的非均匀划分动态优化方法 (ndCVP-HSTA) 流程图

下面给出此自适应动态优化方法 (ndCVP-HSTA) 的具体步骤：

Step 1: 初始化: 初始化决策变量 $d^0 = [\xi^0, \theta^0]$: 在可行域内随机产生初始解; 给定初始网格段数 p ; 设置转换容许值 ε_s ;

Step 2: 全局优化: 将原动态优化问题转换为 NLP 子问题, 以 d^0 为初始点用状态转移算法求解目标函数最优值 J ; 得到接近全局最优解 d^* , 包括最优控制参数值 ξ^* 以及最优转换时间二分点 θ^* ;

Step 3: 局部精细: 以 d^* 为初始解, 采用梯度搜索算法求解目标函数最优值 J ; 得到全局最优解 d^{**} , 因此可以得到最优控制参数值 ξ^{**} , 以及最优转换时间二分点 θ^{**} ;

Step 4: 最后根据 d^{**} 得到最优控制曲线, 包括最优离散时间点 t_k, t_k^{in} , 以及控

制参数值 ξ^{**} 。

3.4 实验结果与讨论

为了验证 ndCVP-HSTA 求解动态优化问题的有效性, 本节给出了几个经典动态优化问题的数值仿真结果。所有实验都在 MATLAB (R2017b 版本) 以及带有 3.4GHz intel i7 8G RAM 的台式电脑上运行, 选用 MATLAB 内部 ode45 函数为微分求解器。其中梯度算法 BFGS 判断结束的一阶最优性条件 $\|g\| \leq \varepsilon_{opt}$ 中, $\varepsilon_{opt} = 10^{-6}$ 。STA 算法与上一章保持一致。将控制变量离散为 p 个区间, 分别对不同 p 下的每种算法以随机初值进行 30 次独立运行, 以减少算法随机性对结果的影响。每个问题 (pro) 都采用最好值 (best)、平均值 (mean)、最差值 (worst)、标准差 (std) 和时间 (time) 等常用的统计指标来评价该方法的性能, 时间是 30 次独立运行的 CPU 运行时间平均值。

1) 案例一: 管式反应器催化剂浓度分布问题

管式反应器催化剂浓度分布问题是一个具有控制曲线奇异弧的最大化产物动态优化问题。其数学模型如式 2-17 所示。表 3-2 中展示了三种动态优化方法 ndCVP-HSTA、ndCVP-BFGS 与 udCVP-vSTA 求解此问题的对比实验数据。

如表 3-1, 为了对比 ndCVP-HSTA 的有效性, 在同一非均匀离散网格下, 分别仅采用 STA 和仅采用梯度算法对离散后的 NLP 子问题进行求解。此外, 为了阐述本节提出的部分离散网格策略 ndCVP 的有效性, 表中也列出了不同分段数下均匀离散网格的最优控制参数的目标函数值。

表 3-1 PFR 问题实验结果

pro	method	p	best	mean	worst	std	time(s)
PFR (max)	HSTA	10	0.47771202	0.47771202	0.47771202	9.639529e-13	11.537894
	BFGS		0.47771202	0.47764596	0.47672131	2.511542e-04	5.600806
	udCVP-	10	0.473630	-	-	-	15.329338
	vSTA	15	0.474531	-	-	-	33.564853
		20	0.475272	-	-	-	51.646119

由表 3-1 的上部分观察可得, 在非均匀分段的情况下, 针对同一个 NLP 子问题, HSTA 的求解能力优于单纯使用梯度算法的求解能力, 虽然其最优值相同, 但是 HSTA 求解的平均值 (0.4771202) 和最差值 (0.4771202) 均分别优于 BFGS 求解的平均值 (0.47764596) 和最差值 (0.47672131), 且 HSTA 表现出非常强的稳定性, 此性能可由标准差表示 (9.639529e-13)。由于需要同时优化控制值与时间子区间, 自变量之间耦合影响目标函数, NLP 问题非线性显著, 因 BFGS 容易过快过早收敛于局部极值, 使得 BFGS 的平均求解时长虽短于 HSTA, 但求解精

度和稳定性较差。HSTA 在前一阶段使用受初始值影响小的全局搜索算法 STA，在后一阶段使用能较快收敛到极值点的梯度算法，能较好的综合两类算法优点，在相对满意的时间内达到稳定的求解精度。

由表 3-1 的下部分观察可得，非均匀分段在 10 段的情况下的控制曲线均优于均匀分段的 10、15 及 20 段，证明此部分非均匀分段能找到合适的离散网格，定位到合理的切换点。且在相同分段数 ($p=10$) 下，ndCVP-HSTA 的消耗时间也相对优于均匀方案下基础 STA 的优化结果。

图 3-13 的图左为部分非均匀离散分 10 段下的最优控制轨迹，图右为同分段数均匀离散下的最优控制曲线。由图可知，部分离散 CVP 能较好的定位到最佳切换时间点，该问题控制轨迹可仅由三段常数组成，其最优转换时间点为[0, 0.14, 2.40, 3.03, 4.80, 5.29, 7.20, 7.77, 9.60, 11.73, 12]，具有奇异弧，最短奇异弧所占区间长度仅为 0.1363，若为均匀离散，则需要 $12/0.1363=88$ 段才能准确契合。而部分非均匀离散网格因其可移动的二分点能有机会准确的定位到最佳切换点，拟合奇异弧，最好的近似原控制曲线，得到更优的目标函数值。

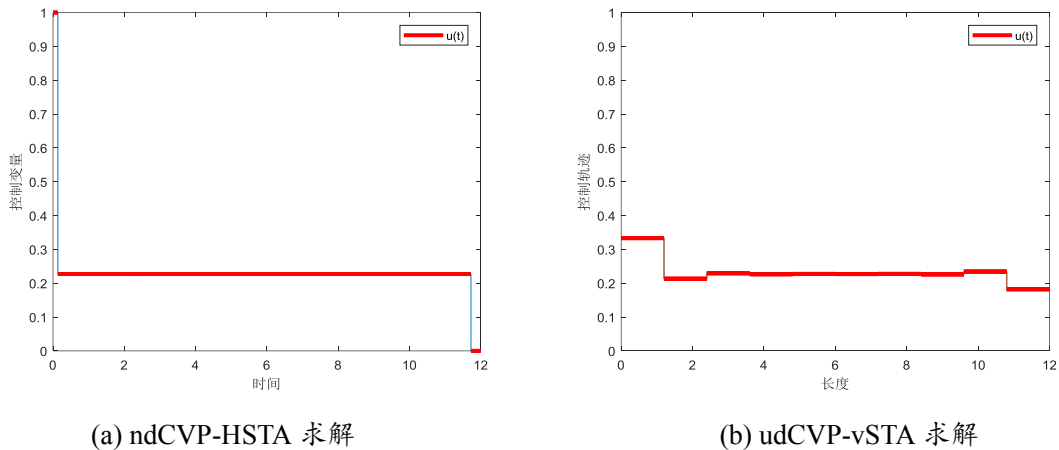


图 3-13 两种动态优化方法求解 PFR 问题最优结果 ($p=10$)

2) 案例二：CSTR 反应器冷却液问题

连续搅拌反应器 (CSTR) 优化是一个多模态动态优化问题。其数学模型如式 2-18 所示。表 3-2 中展示了三种动态优化方法 ndCVP-HSTA、ndCVP-BFGS 与 udCVP-vSTA 求解此问题的对比实验数据。

表 3-2 CSTR 问题实验结果

pro	method	p	best	mean	worst	std	time(s)
CSTR (min)	HSTA	10	0.136425	0.143688	0.244931	2.750918e-02	16.740206
	BFGS		0.136454	0.216122	0.245641	4.883403e-02	7.251487
	udCVP-	10	0.137258	-	-	-	20.365621
	vSTA	15	0.134969	-	-	-	44.395781
		20	0.134156	-	-	-	75.504284

首先, 由表 3-2 可以看出, 在非均匀分段的情况下, 针对同一个 NLP 子问题, HSTA 的求解能力优于单纯使用梯度算法的求解能力, HSTA 求解的最优值、平均值和最差值均分别优于 BFGS 求解的, 但时间上略长于梯度算法。其中有两个原因: 1) CSTR 反应器问题是个多模态动态优化问题, 梯度算法常常会陷入局部最优解 ($J=0.244$), 导致算法还未搜索到全局最优解而提前终止, 因此时间快, 但精度差; 2) 由于非均匀分段需要同时求解最优时间子区间, 动态问题的非线性和复杂性被显著增强, 因此梯度算法求解性能很大取决于初始解的好坏 (最好达到过 0.136454, 最差有 0.245641), 难以稳定的求取到最优解。3) HSTA 集成了进化算法的全局性能和梯度算法的局部性能, 定位到较好初始解后选择梯度算法跟进, 因此能够达到较高精度。

随后, 由表 3-2 下半部分可看出, 在相同的分段数 $p=10$ 下, 在非均匀分段下的控制曲线较均匀分段拥有更高的精度, 因此非均匀分段下的目标函数值 (0.136425) 较均匀分段下的目标函数值 (0.137258) 更优。进一步地, 由于梯度精细化操作的加入, 在相同段数且非均匀分段非线性更强的情况下, ndCVP-HSTA 的计算代价 (16.740206 s) 要低于 udCVP-vSTA 优化 (20.365621 s)。

图 2-14 给出了该问题在部分非均匀分 10 段下的最优控制轨迹, 该问题原最优轨迹较为光滑, 理应分段数越多越能逼近原最优控制轨迹。在相同分段数下, 部分非均匀分段方法能更好的检测到最优离散网格, 较好的拟合控制曲线的斜率变化, 从而得到更优的目标函数值。

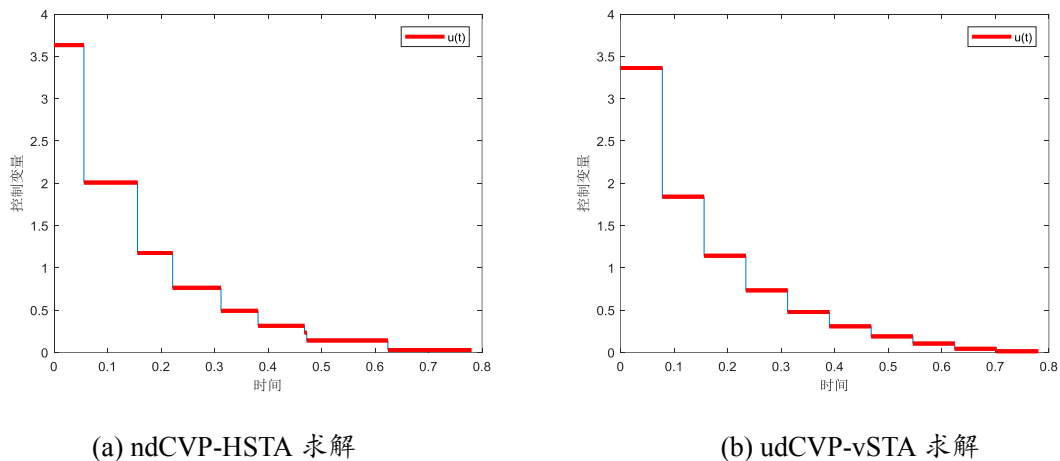


图 2-14 两种动态优化方法求解 CSTR 问题最优结果 ($p=10$)

3) 案例三: Nishida 问题

Nishida 问题是一个具有 bang-bang 控制结构的动态优化问题。其数学模型如式 2-19 所示。表 3-3 中展示了三种动态优化方法 ndCVP-HSTA、ndCVP-BFGS

与 udCVP-vSTA 求解此问题的对比实验数据。

表 3-3 Nishida 问题实验结果

pro	method	p	best	mean	worst	std	time(s)
Nishida (min)	HSTA	10	1.065413	1.091067	1.092900	6.973644e-03	11.661087
	BFGS		1.065413	1.401849	2.387988	4.396610e-01	3.156798
	udCVP-	10	1.671318	-	-	-	6.172973
	vSTA	15	1.234664	-	-	-	13.329868
		20	1.453320	-	-	-	60.062419
		30	1.148940	-	-	-	118.738517

首先，有表格 3-3 上部分可看出，在此 ndCVP 部分移动离散网格下，针对于此同一个 NLP 问题，HSTA 的求解性能，不管是平均值还是最差值都要优于仅梯度优化方法。由于 30 次独立优化初始解均随机选取，如果初始解较好，则梯度算法有机会快速定位到最优值 (1.065413)，但由于此 NLP 子问题非线性较强，梯度算法较难稳定得到最优值，甚至出现过早 (3.156798 秒) 停滞于较差局部解 (2.387988)，使得整个算法稳定性能较差，30 次运行平均值居高不下 (1.401849)。而 HSTA 充分利用梯度算法的局部快速收敛能力，首先进行全局搜索，粗略定位到全局最优解附近，再进行局部搜索提升解的精度，提升局部搜索效率。

由上一章分析可知，Nishida 问题作为 bang-bang 问题拥有自己特定的最优切换时间点。对于此问题，最优目标函数值并不随分段数增大依次变优，而是取决于控制曲线切换时间点的契合准确程度。由表 3-3 可知，非均匀分段的最优目标函数值 (1.065413) 均要优于 10 段、15 段、20 段、30 段的均匀离散下的优化情况。因为部分移动离散方法拥有更大的自由度能方便契合 bang-bang 问题的最优切换时间点。

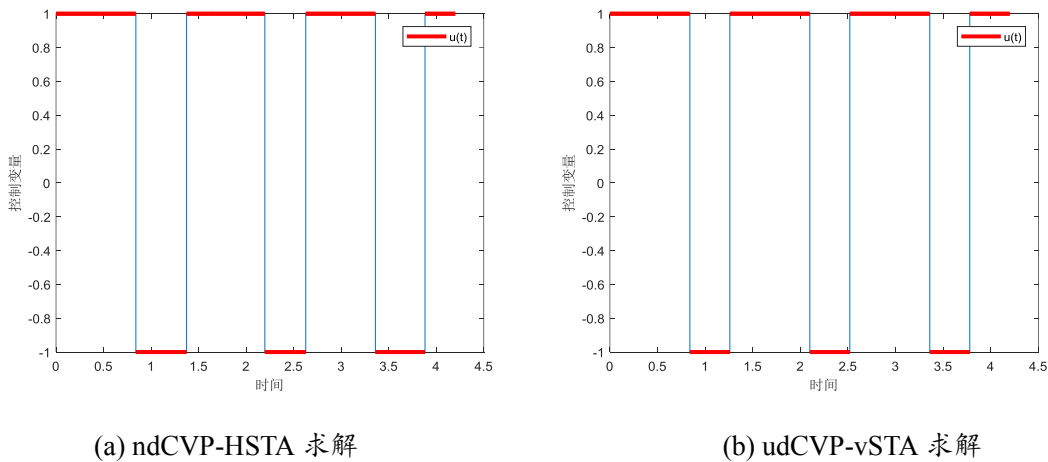


图 3-15 两种动态优化方法求解 Nishida 问题最优结果 ($p=10$)

图 3-15 的图左为非均匀离散分 10 段下的最优控制轨迹，图右为同分段数均匀离散下的最优控制曲线。由图可知，该问题为 bang-bang 控制问题，其最优控制参数值为边界值，即-1 或 1，由 7 段常数组成，部分移动离散网格下求出的最优切换时间点为 $[0, 0.83, 0.84, 1.37, 1.68, 2.20, 2.52, 2.63, 3.36, 3.88, 4.20]$ ，对应值为 $[1, 1, -1, 1, 1, -1, -1, 1, -1, 1]$ 。在相同分段数下，部分非均匀分段方法能更好的检测到最优离散网格，较好的契合控制曲线的切换时间点，从而得到更优的目标函数值。

3.5 本章小结

基于均匀划分控制参数化的数值型动态优化方法因其编程简单、工程操作方便而受到实际工程动态优化问题的广泛应用，但其问题的近似求解精度与求解效率极大的受影响于划分段数。为了提高非凸动态优化问题求解精度和效率，本章提出了基于混合状态转移算法的非均匀划分动态优化方法。

本章首先分析了传统等区间的均匀离散化方法的优缺点，阐述了非均匀控制参数化的必要性，其次分别从网格点移动、网格点增删的角度对前期学者的非均匀参数化方法思想进行总结和概括，并列出了其优势和劣势；因此提出了基于部分移动离散的自适应 CVP 策略，此方法不优化全部时间节点，仅优化二分时间区间，因此较传统网格点移动有着问题维数较低的特点，较传统网格点增删方法有着能自适应决定插入点位置的特点。此外，由于需要同时优化时间子区间长度和控制参数值，离散后 NLP 子问题趋于复杂，为了追求优化算法的高效、全局和稳定性，本章提出了 HSTA 这一两阶段算法，将全局性极强的 STA 应用于第一阶段的全局搜索，将局部搜索能力佳的梯度算法应用于后一阶段的精细搜索，使得第一阶段能为后一阶段提供较好的初始解。最后，通过结合 ndCVP 和 HSTA 算法，给出了求解动态优化问题的 ndCVP-HSTA 方法，并将此方法应用于三个动态优化问题中。实验表明，ndCVP-HSTA 能以较高的效率在较小分段数下找到令人满意的控制曲线，且结果均优于前一章的优化结果。由实验分析可得，同一分段数下，ndCVP-HSTA 在求解具有光滑控制曲线的动态优化问题上优于传统均匀分段，在求解最优控制曲线具有奇异弧或 bang-bang 结构的动态优化问题时具有显著优势。

4 约束动态优化及在除铜过程优化控制中的应用

锌是一种重要的有色金属，全世界 80%以上的锌是由湿法炼锌生产的，其主要由酸液浸出、锌液净化、电解沉积等三个部分组成^[90]，其中锌液净化阶段旨在将直接浸出液中的杂质离子，如铜、钴、镍、镉、砷等降到一定范围内，避免电解过程出现烧板等非安全生产危害。除铜工艺（Copper removal process, CRP）作为锌液净化的第一阶段，通过添加锌粉，将铜离子浓度控制在指定范围以内，在沉降铜离子的同时为下游提供适量铜离子做催化剂，由此除铜过程的浓度合格控制与锌粉添加控制尤为重要。同时，除铜工艺是固液相非催化多反应过程，是一个典型的动态过程，其各离子浓度随着反应时间的演进，在反应器进出口处有了不同程度的量变或质变。因此，为了实现除铜过程生产稳定、资源节约，本节拟对除铜过程进行动态优化，即寻求到最佳锌粉添加量的操作点或操作轨迹，在满足铜离子浓度降低至合适的生产指标的约束条件下，使得工艺过程资源消耗最小，如锌粉耗费最少。

4.1 湿法炼锌除铜过程动态优化问题

在除铜阶段，并非所有的铜都应去除，而在随后的去除过程中，需要保留少量的铜，以抑制镉的沉淀并加速钴的沉淀^[90]。在工业实践中，由于反应机理复杂，滞后的离子浓度检测机制，操作人员难以进行合理的控制来满足对过程输出的约束，实现过程稳定最优控制。实践中常用的极为保守的策略会导致锌粉消耗过多以及出口铜离子浓度过低，而添加剂数量不足则无法将杂质降低到所需水平。因此，优化除铜工艺中添加剂的控制是十分必要的。本节旨在分析湿法炼锌除铜过程，构建符合实际工况的动态优化问题，旨在保证出口铜离子浓度合格的同时减少锌粉消耗。

4.1.1 除铜过程动态特性描述

除铜过程通过向两个级联反应釜中持续添加锌粉，将铜离子从硫酸锌浸出溶液中分离出来。如图 4-1 所示，浸出液将连续地送入两个反应釜，锌粉通过传送带分别从锌粉仓中输送至两个连续釜内，再流入浓缩机进行固液分离，经过沉降的含固底流会有一部分返回#1 反应釜用作除铜反应晶种，另一部分过滤出铜渣被回收，浓密机上部流出的除铜后液送至后续除钴、镍工段。值得注意的是，第一个反应釜通常作为主反应器，负责大多数铜离子的沉积，而第二个反应釜通常作为辅助反应器，微调出口铜离子浓度至指定范围。

湿法炼锌除铜过程通过持续向连续搅拌反应釜中添加锌粉，使得入口上清液

的铜离子发生置换反应及归中反应，进而形成铜单质沉淀和氧化亚铜沉淀，降低溶液中铜离子浓度至指定范围。在除铜反应中，铜离子以铜单质和氧化亚铜两种沉淀的形式得到沉降，其化学反应方程式如下：

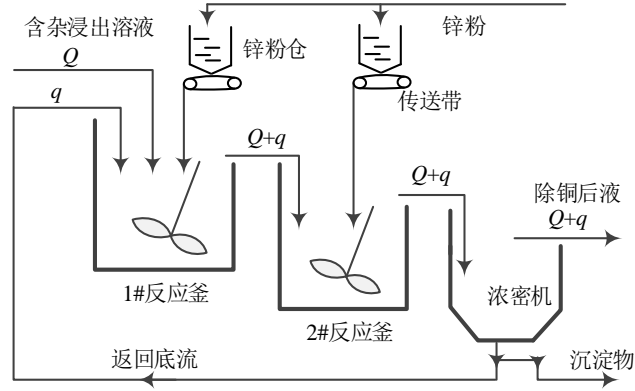
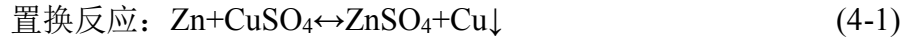


图 4-1 湿法炼锌除铜过程示意简图

在反应釜中，具有正电位的铜离子能轻易与具有负电位的锌单质发生置换反应，生成沉淀铜(4-1)。除此之外，在具有反应晶种的具有合适 pH 值的反应环境中，单质铜和二价铜离子之间也可正向进行归中反应(4-2)，从而生成沉淀氧化亚铜。

针对除铜过程，张斌等人^[92]引入 CSTR 模型，深入分析了除铜置换反应机理和除铜归中反应，给出了除铜过程竞争—连续反应模型。因此除铜过程动力学模型可由如下化学方程式来描述：

$$\begin{cases} V\dot{C}_{\text{Cu}^{2+},1} = QC_{\text{Cu}^{2+},1}^{\text{in}} - (Q+q)C_{\text{Cu}^{2+},1} - V \cdot r_{\text{Cu}^{2+},1} \\ V\dot{C}_{\text{Cu}^{2+},2} = (Q+q)C_{\text{Cu}^{2+},1} - (Q+q)C_{\text{Cu}^{2+},2} - V \cdot r_{\text{Cu}^{2+},2} \end{cases} \quad (4-3)$$

其中， $C_{\text{Cu}^{2+},i}^{\text{in}}, i=1,2$ 和 $C_{\text{Cu}^{2+},i}, i=1,2$ 分别是入口和出口铜离子浓度。 $\dot{C}_{\text{Cu}^{2+},i}, i=1,2$ 是第 i 个反应釜的离子改变速率。 V 是反应器的有效容积。 Q 和 q 分别是上清液入口流量和返回底流流量。基于竞争—连续反应机理模型，除铜过程第 i 个反应器的沉降速率 $r_{\text{Cu}^{2+},i}, i=1,2$ 可由如下表达式表示：

$$\begin{cases} r_{\text{Cu}^{2+},1} = (k_1 G_{\text{Zn},1} + k_2) V^{-1} C_{\text{Cu}^{2+},1} \\ r_{\text{Cu}^{2+},2} = (k_1 G_{\text{Zn},1} + k_3) V^{-1} C_{\text{Cu}^{2+},2} \end{cases} \quad (4-4)$$

其中， $k_i, i=1,2,3$ 为机理参数， $G_{\text{Zn},i}, i=1,2$ 分别为第 i 个反应器的锌粉添加率。

实际除铜生产过程中，可人工调节的参数包含入口溶液流量 Q 、底流流量 q 以及锌粉添加量 $G_{\text{Zn},i}, i=1,2$ 。其中，底流流量取决于除铜工艺对渣矿的质量要求，

因此在整个时间内为常数；入口流量取决于上游工艺，不属于锌液净化除铜过程优化的可控变量；除铜过程的各反应釜锌粉添加量是除铜过程中调节最频繁、影响最大的控制量，一般由专业操作员凭借经验知识根据工况手动调节。因此，动态模型中仅将锌粉添加量作为控制变量，出口铜离子浓度作为状态变量反应实际工况。为了便于表达，除铜过程的动态模型可简化为如下表达式：

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}_1 \mathbf{x} + \mathbf{A}_2 \mathbf{x}_{in} + \phi(\mathbf{x}, \mathbf{u}) = F(\mathbf{u}(t), \mathbf{x}(t), t) \\ \mathbf{A}_1 &= \begin{bmatrix} -V^{-1}(Q+q) & 0 \\ 0 & -V^{-1}(Q+q) \end{bmatrix} \\ \mathbf{A}_2 &= \begin{bmatrix} V^{-1}Q & 0 \\ 0 & V^{-1}(Q+q) \end{bmatrix} \\ \phi(\mathbf{x}, \mathbf{u}) &= \begin{bmatrix} -V^{-1}(k_1 u_1 + k_2) x_1 \\ -V^{-1}(k_1 u_1 + k_3) x_2 \end{bmatrix} \end{aligned} \quad (4-5)$$

其中，锌粉添加量为控制变量 $\mathbf{u} = [G_{Zn,1}, G_{Zn,1}]$ ，出口铜离子浓度为状态变量 $\mathbf{x} = [C_{Cu^{2+},1}, C_{Cu^{2+},2}]$ ，入口铜离子浓度为 $\mathbf{x}_{in} = [C_{Cu^{2+},1}^{in}, C_{Cu^{2+},2}^{in}]$ 。值得注意的是，第#2入口铜离子浓度即#1出口铜离子浓度，即 $C_{Cu^{2+},2}^{in} = C_{Cu^{2+},1}$ 。

4.1.2 除铜过程生产约束分析

除铜过程是湿法炼锌工艺流程中必不可少的一步，其通过向一系列连续搅拌反应釜中添加锌粉将杂质铜离子沉降到指定范围。铜离子生产约束条件的满足是至关重要的，除此之外还有由于设备要求的输入约束，以及其他考虑安全与稳定性的状态相关约束也应纳入考虑范围。

1) 输入约束

在除铜过程中，锌粉通过皮带分别传送至两个连续搅拌反应釜中。考虑到执行器设备限制，除铜过程各级反应的锌粉皮带最大添加率应在指定范围，可由如下表达式描述：

$$u_i^{\min} \leq G_{Zn,i} = u_i(t) \leq u_i^{\max}, i=1,2, \quad (4-6)$$

其中， u_i^{\min} 和 u_i^{\max} 为第*i*个反应器允许的最大锌粉添加率。

2) 生产指标约束

杂质铜离子因较低电位而成为首要去除对象，除铜过程的出口铜离子浓度过高或过低，直接影响着后续净化工序的控制与达标，从而影响整个炼锌流程安全稳定生产。出口铜离子浓度低，活化能力不足，后续除钴操作将消耗过多锌粉来沉降钴离子，除钴效率低；铜离子浓度适中，则除钴催化效果较好，除钴效率高、

效益大；铜离子浓度高，易引发析出钴返溶现象，降低除钴效率和效果；同时，过高的杂质离子浓度还会降低后续电解效率，恶化电解生产环境，增加电解能耗，甚至出现烧板现象以致停产。因此，为了便于进一步的下游处理，需要在除铜期间将铜离子浓度精确地保持在一定范围内：

$$C_{\text{Cu}^{2+},2}^{\min} \leq C_{\text{Cu}^{2+},2} = x_2(t) \leq C_{\text{Cu}^{2+},2}^{\max} \quad (4-7)$$

其中， $C_{\text{Cu}^{2+},2}^{\min}$ 和 $C_{\text{Cu}^{2+},2}^{\max}$ 为#2 反应器出口离子浓度的上下界。

3) 生产稳定约束

在除铜过程中，铜离子沉降在两个级联反应器中逐级进行。#1 反应釜负责沉淀大部分的铜离子，而#2 反应釜负责微调，其中#1 反应器的除铜率是保证除铜过程安全稳定运行的关键因素之一。#1 反应器除铜率越高，则#2 反应器入口铜离子浓度越低，则可能与#2 反应器中残留的锌粉发生反应，使得过分除铜，即出口铜离子浓度低于生产下限。相反，如果前一个反应器中铜离子除铜率过低，则后一个反应器可能没有足够的时间沉降流入的过多铜离子，使得整体除铜不充分，出口铜离子浓度偏高。因此，遵循适当的离子下降梯度很重要，这有助于降低违反过程约束的风险。

为了保证除铜过程各反应器中浓度下降的稳定性，避免产生主反应器过激调节或反应不足等现象，#1 反应器应满足如下约束：

$$R_{\text{Cu}^{2+},1}^{\min} \leq \frac{C_{\text{Cu}^{2+},1}^{\text{in}} - x_1(t)}{C_{\text{Cu}^{2+},1}^{\text{in}}} \leq R_{\text{Cu}^{2+},1}^{\max} \quad (4-8)$$

其中， $R_{\text{Cu}^{2+},1}^{\min}$ 和 $R_{\text{Cu}^{2+},1}^{\max}$ 分别是#1 除铜率的上下阈值。

4.1.3 除铜过程动态优化问题

动态优化问题一般涵盖评价指标以及约束条件两个元素。其中评价指标是衡量系统在不同控制策略下工作好坏的标准；其约束条件分为系统动态模型、系统状态相关约束、控制约束等。

通过动态特性描述和约束分析，除铜过程优化控制问题可以构建成一个典型的带状态约束的动态优化问题。此其优化目标是在一定时间内尽量减少原材料消耗，其控制变量是两个搅拌反应器中锌粉的进料速率，此外含有两个与铜离子浓度相关的状态约束：1) 由于旨在沉淀杂质铜，必须对出口铜离子浓度施加状态变量路径限制，使其在整个生产过程中始终保持一定范围内；2) 为了更好地运行和稳定生产，主反应器的铜去除率也应满足一定的路径约束。

因此，除铜过程动态优化问题的数学模型可描述如下：

$$\begin{aligned} \min_{u(t)} J(\mathbf{u}(t)) &= \int_{t_0}^{t_f} L_0(\mathbf{u}(t), \mathbf{x}(t), t) dt \\ \text{s.t.} \left\{ \begin{array}{l} \dot{\mathbf{x}} = F(\mathbf{u}(t), \mathbf{x}(t), t) \\ \mathbf{x}(t_0) = [x_1(t_0), x_2(t_0)] \\ \text{式(4-7)} \\ \text{式(4-8)} \\ u_i^{\min} \leq u_i(t) \leq u_i^{\max}, i = 1, 2 \\ t \in [t_0, t_f] \end{array} \right. \end{aligned} \quad (4-9)$$

其中， F 是动态系统微分方程约束(4-5)，用来描述此非线性系统；(4-7)和(4-8)为不等式状态变量约束； $\mathbf{u}(t)$ 表示控制变量，锌粉添加量， $\mathbf{x}(t)$ 表示状态变量，出口铜离子浓度， $\mathbf{x}(t_0)$ 是动态系统在 t_0 时刻的初始状态，优化时间段为 $[t_0, t_f]$ 。代价函数 L_0 可为两个罐子的锌粉添加总量：

$$L_0(\mathbf{u}(t)) = u_1(t) + u_2(t) \quad (4-10)$$

经过三个月的工艺观察和数据收集，除铜工艺数据如下：

表 4-1 除铜过程工业数据 (90 天)

参数	单位	值
浸出 ZnSO ₄ 入口流量, Q	m ³ /h	150-250
返回底流流量, q	m ³ /h	10-22
溶液体积, V	m ³	98-102
#1 入口铜离子浓度, $C_{\text{Cu}^{2+},1}^{\text{in}}$	g/L	1.1-2.1
#2 出口铜离子浓度理想范围, $C_{\text{Cu}^{2+},2}$	g/L	0.2-0.4
#1、#2 锌粉添加率, $G_{\text{Zn},i}, i = 1, 2$	kg/h	0-500
#1 反应器除铜率, $R_{\text{Cu},1}$	-	0.53-0.64

CRP 中出现的优化问题可以构造为动态优化问题 (DOP)，其目标是找到一个与降低生产成本相对应的时变控制轨迹，从而在给定的工艺时间内以最经济的成本获得理想的生产质量。更重要的是，优化是在存在各种生产约束的情况下进行的，即动态系统微分方程施加的固有约束、由设备要求决定的输入约束以及由安全和可操作性考虑引起的状态相关约束^[26]。

如何有效地求解动态优化问题，特别是当问题公式包含连续不等式约束时，仍然是一个挑战，主要有三个原因。1) 大多数约束动态优化问题过于复杂，难以解析求解^[44]，因此只能通过数值方法和附加参数化操作来求解。因此，求解效率和求解质量依赖于控制轨迹的参数化方案。2) 在运行时间范围内的每个点都受到约束，因此需要严格满足不可数目的常规点约束。3) 即使对于没有约束的

问题, 数值方法得到的解质量也不总是令人满意的, 因为工业过程中产生的涉及动态系统的优化问题往往因其高度非线性、多维性和多模性而困难^[87]。因此, 有效、准确地解决这些问题仍然是一个问题。

4.2 光滑约束动态优化方法

除铜过程最小锌粉添加的稳定生产优化控制问题是典型的带约束的动态优化问题。为了有效求解此类带约束动态优化问题, 本章首先分析了一般带约束动态优化问题, 在第三章无约束动态优化方法的研究基础上, 针对无穷维状态相关约束, 引入了光滑化罚函数机制, 构建了基于状态转移算法的光滑约束动态优化方法。

4.2.1 一般约束动态优化问题

不失一般性, 在本节我们先讨论符合如下标准形式的带约束、单目标、固定优化时长的动态优化问题。

$$\begin{aligned} \min_{\mathbf{u}(t)} \quad & J = \Phi[\mathbf{x}(t_f)] + \int_{t_0}^{t_f} L[t, \mathbf{x}(t), \mathbf{u}(t)] dt \\ \text{s.t.} \quad & \begin{cases} \dot{\mathbf{x}}(t) = f[\mathbf{u}(t), \mathbf{x}(t), t] \\ \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{g}[\mathbf{u}(t), \mathbf{x}(t), t] \leq 0 \\ \mathbf{u}' \leq \mathbf{u}(t) \leq \mathbf{u}'' \\ t_0 \leq t \leq t_f \end{cases} \end{aligned} \quad (4-11)$$

其中, 目标函数 J 为预定的性能指标, 其包括了在终端时刻的终端性能指标 $\Phi[\mathbf{x}(t_f)]$ 和过程性能指标 $\int_{t_0}^{t_f} L[t, \mathbf{x}(t), \mathbf{u}(t)] dt$ 。控制变量约束为 $\mathbf{u}' \leq \mathbf{u}(t) \leq \mathbf{u}''$ 。状态约束表述为 \mathbf{g} , 有 m 个。 $[t_0, t_f]$ 为优化时间区间。

动态约束优化问题中的路径约束一般可分为两类: 1) 纯控制变量约束; 2) 状态变量相关约束。这两种路径约束均与时间有关, 以下分别对此进行讨论。

针对纯控制变量约束, 由于本文采用分段常数拟合, 因此纯控制变量约束可简化为决策变量值的约束, 如图 4-2, 则可采用求解带箱型约束 (Box constraint) 优化问题的处理方法解决。典型处理方法有三角函数变换法、约束算子法^[93]: 如果控制变量受到路径约束 $\mathbf{u}' \leq \mathbf{u}(t) \leq \mathbf{u}''$, 则应用 CVP 策略后, 约束被离散化为如下形式: $\mathbf{u}' \leq \xi_i \leq \mathbf{u}''$, $i=1, 2, \dots, p$, p 为分段数量。迭代过程中对每次优化的结果都进行式 (4-12) 的约束处理: 该方法便于程序实现, 且计算量不大, 特别适用于上下限值恒定的情况。但是该方法仅能处理控制变量路径约束, 却不适用于状态变量相关约束的处理。

$$\xi_i = \begin{cases} u^l, & \xi_i \leq u^l \\ \xi_i, & u^l \leq \xi_i \leq u^u \\ u^u, & \xi_i \geq u^u \end{cases} \quad (4-12)$$

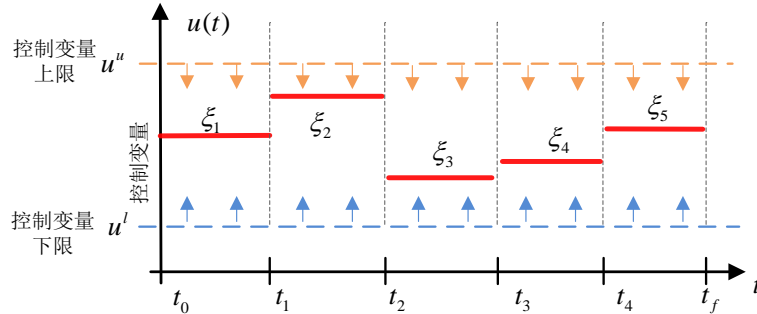


图 4-2 动态优化问题纯控制变量约束示意图

状态变量相关约束又分为两种：1) 纯状态变量约束；2) 控制状态变量混合约束。动态优化问题的一大特点是时变，即在一段时间内通过改变控制变量，控制作用在动态系统上，从而间接改变状态变量。控制变量可用分段常数值进行拟合，但状态变量曲线受控制变量以及动态系统双层影响，其状态轨迹通常连续则变化且事先未知。状态变量路径约束意味着每一个点都必须满足路径约束要求，其难点也根植于此：对于某个不等式路径约束，它在优化时间段 $[t_0, t_f]$ 内的无数点上都存在约束，且不知道在哪一点将成为积极约束。

纯状态变量约束和控制状态变量混合约束都需要在优化时间段内的无穷时间点上满足。如图 4-3 为纯状态变量约束示意图，即要求 $c_2(t) \leq x(t) \leq c_1(t)$ ， c_1 为时变的#1 状态变量约束， c_2 为时变的#2 状态变量约束。由最优控制下的状态曲线可知，其在特定的某两个时间点转换为积极约束。如图 4-4 为状态控制变量混合约束示意图，即要求 $x(t) \leq u(t)$ 。控制状态变量混合约束满足难度更大，因为控制变量需要满足其路径约束，而状态变量需要在每一个时间点相对小于控制变量，两种约束相互耦合。

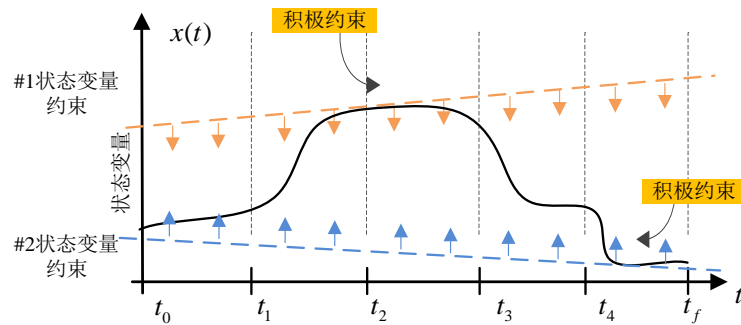


图 4-3 动态优化问题纯状态变量约束示意图

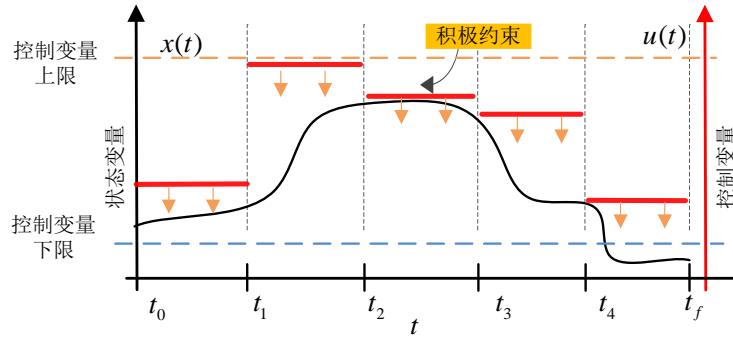


图 4-4 动态优化问题状态控制变量混合约束示意图

4.2.2 状态相关约束处理

罚函数法作为一种约束处理机制在处理约束优化问题上得到了广泛的认可^[68]。罚函数法是将约束违反度与罚因子相乘后与目标函数值相加，从而将约束问题转化为一个无约束问题。最常用的是 l_1 精确罚函数。

考虑到动态优化问题(4-9)，可以将连续时间上的不等式约束转换为下列规范的总约束违反度表达式：

$$G = \sum_{i=1}^m \int_{t_0}^{t_f} \max \{ g_i(u(t), x(t), t), 0 \} dt \quad (4-13)$$

其中， m 是约束函数的个数，将(4-13)中的约束违反度 G 作为惩罚项加在目标函数中，代价函数变为：

$$J'(u(t)) = J(u(t)) + \rho \cdot \sum_{i=1}^m \int_{t_0}^{t_f} \max \{ g_i(u(t), x(t), t), 0 \} dt \quad (4-14)$$

其中 $\rho > 0$ 为罚因子，原优化问题被转变为无约束优化问题。当 ρ 较小时，对违反约束的解的惩罚力度越小，获得原约束优化问题的可行解的速度越慢；当 ρ 越大，对违反约束的解的惩罚力度越大，获得原约束优化问题的可行解的速度越快，但转换后无约束优化问题趋于病态，不利于原问题全局最优解的搜索。值得注意的是 $\max \{ \cdot, 0 \}$ 在 $g(x) = 0$ 处是不可导的，这样使得问题难以被基于梯度的优化算法求解，也就无法利用梯度算法快速、精准的局部搜索性能。为了解决此问题，引入文献[43]提出的光滑化函数来近似此非光滑化 \max 罚函数，如下所示：

$$S(y, \alpha) = \frac{1}{2} \left[\sqrt{y^2 + 4\alpha^2} + y \right] \quad (4-15)$$

其中， α 是一个小的正数，被称为光滑因子。

光滑近似函数 $S(y, \alpha)$ 具有良好的性质：1) 对于 y 连续可导；2) 当 $\alpha \rightarrow 0^+$

时, $S(y, \alpha) \rightarrow \max\{y, 0\}$; 3) 该函数值为正, 即 $p(y, \alpha) > 0$ 恒成立。光滑近似函数 $p(y, \alpha)$ 的近似特性如图所示:

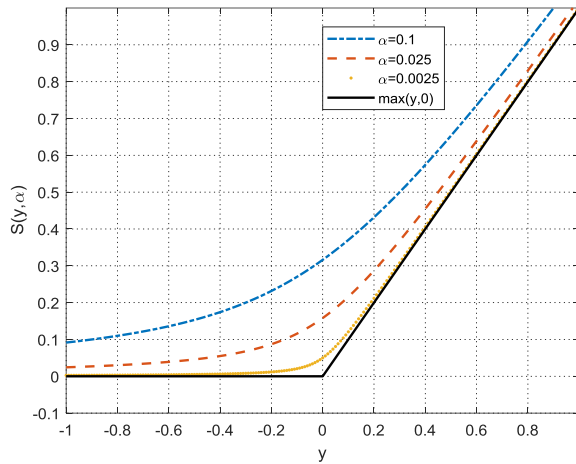


图 4-5 光滑化函数特性

从图可以很明显的看到, 在 $\max\{y, 0\}$ 及其近似函数 $S(y, \alpha)$ 之间存在着一个变宽的间隙, 其中最大的不同发生在 $y = 0$ 处。其中令 $y = g_i(u(t), x(t), t)$, 则代价函数变为:

$$J'(u(t)) = J(u(t)) + \rho \cdot \sum_{i=1}^m \int_{t_0}^{t_f} S(g_i(u(t), x(t), t), \alpha) dt \quad (4-16)$$

值得注意的是, $y = 0$ 处即为 $g_i = 0$, 也就是不等式约束变为等式约束的地方。

同时, 由图可知, 光滑因子 α 为调节近似水平的重要参数。为了获得较好的光滑化效果, α 需要取得越大越好, 但是, 为了较高精度的近似原目标函数, 特别是在约束变为积极约束时, α 需要取的足够小。换句话说, 光滑化罚函数以牺牲近似精度为前提来获得光滑化效果, 并以此来改变目标函数形态, 使得其更有利于优化算法搜索。同理, 由于惩罚因子 ρ 越大, 增广罚之后的目标函数最优解与增广前的越近似, 但是过分大的罚因子又导致增广后函数病态化。

因此, 为了方便优化且逐步提高近似精度, α 需要从指定小数值收敛到 0。同时为了方便前期搜索约束最优解区间到逐步获得可行最优解, ρ 需要从指定小数值增大到无穷。为了提高优化求解效率, 采用 α 和 ρ 同步迭代更新策略, 即 α 和 ρ 在每一次迭代中同时更新, 使得近似无约束问题最优解不断逼近原有约束问题最优解, 其更新公式如下:

$$\alpha^* = d\alpha \quad (4-17)$$

$$\rho^* = \rho/d \quad (4-18)$$

其中 $0 < d < 1$ 是一个特定的下降因子。因此, 当 $\alpha \rightarrow 0$ 的时候, 无约束问题的最

优解就是原有约束问题的最优解。

4.2.3 基于 ndCVP-HSTA 的光滑化约束动态优化方法

前两章研究了高效准确的无约束动态优化问题求解方法 (ndCVP-HSTA), 为了使带约束的动态优化问题更便于优化求解, 本节采用上节介绍的光滑化罚函数方法处理约束。

针对上述问题 (4-11), 通过引入光滑化罚函数方法, 根据当前惩罚因子 ρ 和光滑因子 α , 原始带状态相关约束的动态优化问题 (4-11) 可转变为如下无约束动态优化问题:

$$\begin{aligned} \min_{\mathbf{u}(t)} \quad & J = \Phi[\mathbf{x}(t_f)] + \sum_{k=1}^p \int_{t_{k-1}}^{t_k} L[\mathbf{u}(t), \mathbf{x}(t), t] dt \\ & + \rho \cdot \sum_{i=1}^m \int_{t_0}^{t_f} S(g_i(\mathbf{u}(t), \mathbf{x}(t), t), \alpha) dt \end{aligned} \quad (4-19)$$

$$\text{s.t.} \begin{cases} \dot{\mathbf{x}}(t) = f[\mathbf{u}(t), \mathbf{x}(t), t] \\ \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{u}^l \leq \mathbf{u}(t) \leq \mathbf{u}^u \\ t_0 \leq t \leq t_f \end{cases}$$

为在指定分段数目下高效拟合最优控制网格, 且高效且高精度的找到离散后 NLP 子问题最优解, 采用上章提出的 ndCVP-HSTA; 最终离散及光滑化罚函数处理后的 NLP 子问题如下所示:

$$\begin{aligned} \min_{\xi_i, \theta_k} \quad & J = \Phi[\mathbf{x}(t_f | \xi)] + \sum_{k=1}^p \int_{t_{k-1}}^{t_k} L[\xi, \theta, \mathbf{x}(t), t] dt \\ & + \rho \cdot \sum_{i=1}^m \int_{t_0}^{t_f} p(g_i(\xi, \theta, \mathbf{x}(t), t), \alpha) dt \end{aligned} \quad (4-20)$$

$$\text{s.t.} \begin{cases} \dot{\mathbf{x}}(t) = f[\xi, \theta, \mathbf{x}(t), t] \\ \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{u}^l \leq \xi_i \leq \mathbf{u}^u, i = 1, \dots, 2 \cdot p \\ 0 \leq \theta_k \leq 1, k = 1, \dots, p \\ t_0 \leq t \leq t_f \end{cases}$$

其中, 决策变量为 $\mathbf{d} = [\xi_1, \dots, \xi_{2 \cdot p}, \theta_1, \dots, \theta_p] = [\xi, \theta]$, 维数为 $2 \cdot n_u \cdot p + p = (2n_u + 1)p$ 。动态优化问题转变为: 找出 $(2n_u + 1)p$ 维向量 $[\xi, \theta]$ 使得目标函数 J' 最小, 其中决策变量中控制参数值仍需满足原始约束, 新增变量 $\theta \in [0, 1]$ 。

因此, 本节提出的基于混合状态转移算法的自适应约束动态优化方法步骤如下。其算法流程图如图 4-6 所示。

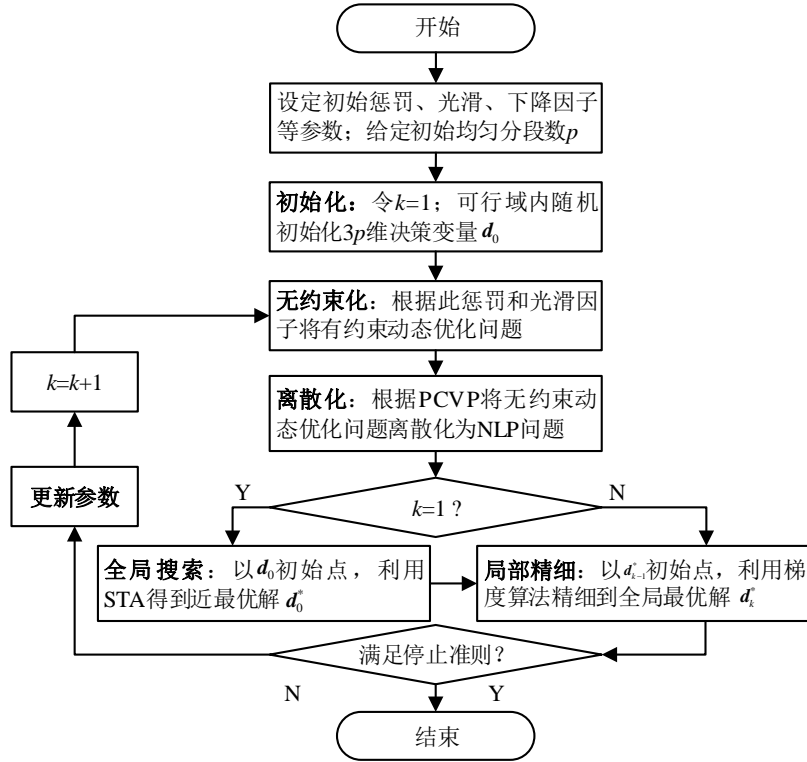


图 4-6 基于 PCVP-HSTA 的光滑化约束动态优化方法流程图

Step 1: 初始化: 设置初始惩罚因子 ρ_0 , 初始光滑化因子 α_0 , 以及下降因子 d , $0 < d < 1$; 令 $k = 1$; 给定初始网格段数 p ; 在可行域内随机初始化决策变量 $\mathbf{d}_0 = [\xi_0, \theta_0]$;

Step 2: 无约束化: 根据惩罚因子 ρ_k 和光滑化因子 α_k 将有约束动态优化问题 (4-11) 转换为无约束动态优化问题 (4-19);

Step 3: 参数化: 根据部分移动网格 CVP 策略离散化以上无约束动态优化问题, 使之成为带约束 NLP 问题 (4-20); 如果 $k = 1$, 则转向 Step 5 进行全局搜索, 否则转向 Step 6 继续局部搜索。

Step 4: 全局搜索: 以 \mathbf{d}_0 为初始解采用 STA 进行全局搜索求解得初始惩罚因子和光滑化因子下的最优解, 得到全局最优解 \mathbf{d}_0^* ,

Step 5: 精细化搜索: 以 \mathbf{d}_{k-1}^* 为初始解采用梯度算法进行局部搜索, 求解得到此惩罚因子和光滑化因子下的全局最优解 \mathbf{d}_k^* , 得到最优控制参数值 ξ_k^* 以及最优转换时间二分点 θ_k^* ;

Step 6: 更新参数: 如果满足终止条件则停止算法并输出 \mathbf{d}_k^* ; 否则, 令 $\alpha_{k+1} = d\alpha_k$; $\rho_{k+1} = \rho_k/d$; $k = k + 1$; 回到 Step 2。

值得注意的是, 其一、为提高求解效率, 下一次迭代优化的初始解都取上一次迭代优化的最优解, 使得整个过程保持热启动状态。其二、STA 仅需要定位初

始惩罚因子和光滑因子下的 NLP 子问题的全局解，随后的精细化操作均由梯度算法承担。

4.3 实验结果与讨论

在本节中，为了评估所提出的约束动态优化方法有效性，进行了两组实验：1) 一组为经典约束动态优化问题测试，分别含有纯状态变量约束或混合状态控制变量约束；2) CRP 的案例研究。所有计算都是在 Matlab (R2017b) 软件平台上进行的，该平台在具有 3.4GHz Intel i7 和 8G RAM 的台式电脑上运行。实验中设置初始惩罚因子 $\rho_0 = 1$ ，初始光滑因子 $\alpha_0 = 2.5 \times 10^{-5}$ ，下降因子 $d = 0.1$ 。PCVP-HSTA 参数同第三章。实验给出了不同迭代次数下，惩罚因子、光滑因子、总约束违反度 G 、目标函数 J 以及每段优化所用时间。

4.3.1 经典约束动态优化问题

1) 案例一：J&L 问题

J&L 问题是纯状态约束动态优化问题，由 Jacobson and Lele 提出^[33]，他包含一个纯状态变量路径约束，其路径约束可对状态变量 x_1 或 x_2 起作用，本文讨论对 x_1 有约束，其动态问题模型如下所示：

$$\begin{aligned} \min \quad & J = x_3(t_f) \\ \text{s.t.} \quad & \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_2 + u \\ \dot{x}_3 = x_1^2 + x_2^2 + 0.005u^2 \\ x_1(t) - 8(t - 0.5)^2 + 0.5 \leq 0 \\ -4 \leq u \leq 15 \\ x(0) = [0, -1, 0]^T \\ t_f = 1 \end{cases} \end{aligned} \quad (4-21)$$

表 4-2 列出了带约束 J&L 动态优化问题求解过程及结果。由表可知，在第一次迭代时，初始惩罚因子较小，惩罚力度不大，算法求得最优解处于可行域外边缘，有轻微的约束违反。在第 2 次迭代时，随着惩罚因子的加大，约束违法度直降到 0，同时，由于光滑因子减小，目标函数值得到减小。在第 3 次迭代，由于光滑因子继续减小，max 函数与光滑化函数之间的间隙趋于 0，因而所求控制曲线得到进一步的逼近原最优曲线，目标函数值得以更小。

表 4-3 列出了此问题的相关文献研究成果。数据表明，本文求解的实验结果 $J = 0.7414$ 与文献结果类似，甚至更优。对于文献[96]的实验结果 $J = 0.7366$ 本文

略差于此结果，但因为 Logsdon 实验分段数为 50，本文分段数仅为 10，更高的分段数理论上可以获得更高的最优控制轨迹拟合精度，因而获得更优的目标函数值。

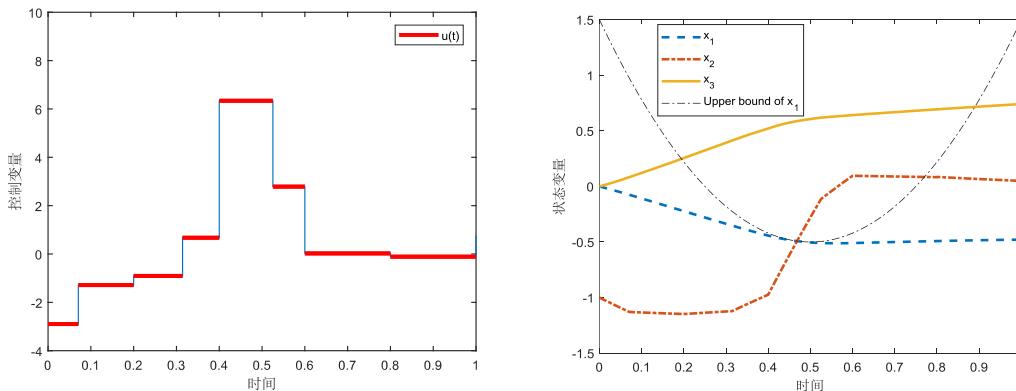
图 4-7 展示了 J&L 问题的最优控制曲线和状态轨迹。由状态轨迹可得，状态变量 x_2 一直在曲线 $8(t-0.5)^2 - 0.5$ 的下方，并在 $t = 0.5$ 的时间点左右取得积极约束。

表 4-2 Jacobson and Lele 问题实验结果

问题	段数	迭代次数	惩罚因子	光滑因子	G	J
J&L (min)	10	1	1	2.5e-4	7.7758e-04	0.7475
		2	10	2.5e-5	0	0.7417
		3	100	2.5e-6	0	0.7414

表 4-3 Jacobson and Lele 问题文献对比

问题	文献	段数	J
J&L (min)	Jacobson <i>et al.</i> [33]	-	0.75
	Feehery <i>et al.</i> [35]	-	0.75145
	Mehra <i>et al.</i> [94]	-	0.79
	Liu <i>et al.</i> [95]	30	0.7485
	Logsdon <i>et al.</i> [96]	50	0.7366



(a) 控制曲线

(b) 状态曲线

图 4-7 光滑化约束动态优化方法求解 J&L 问题所得最优结果

2) 案例二: Rayleigh 问题

Rayleigh 问题是一个典型的状态控制混合约束动态优化问题，出自文献[97]，此问题包含了三个状态变量和一个控制变量，分别有控制变量约束，以及一个与控制变量相关的混合状态约束。该问题因其控制混合状态约束而具有一定难度，其动态问题数学模型如下：

$$\begin{aligned}
 \min \quad & J = x_3(t_f) \\
 \text{s.t.} \quad & \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1 + x_2(1.4 - 0.14x_2^2) + 4u \\ \dot{x}_3 = x_1^2 + u^2 \\ x_1 + 6u \leq 0 \\ -2 \leq u \leq 2 \\ x(0) = [-5, -5, 0]^T \\ t_f = 4.5 \end{cases} \quad (4-22)
 \end{aligned}$$

表 4-4 列出了带约束 Rayleigh 问题求解过程及结果。由表可知，分段数为 10 或 20 时，随着迭代次数的增加，总约束违反度总是持续减小至 0，目标函数值则呈现先升后降的趋势。因为迭代次数增加，惩罚因子不断加倍，惩罚力度不断增大，因而最优解逐步从不可行收敛到可行域内；同时，随着光滑因子的减小，光滑化函数与 max 在积极约束处的近似拟合误差逐渐减小，求得近似解进一步逼近最优控制曲线。

表 4-4 Rayleigh 问题实验结果

问题	段数	迭代次数	惩罚因子	光滑因子	G	J
Rayleigh (min)	10	1	1	2.5e-4	0.2175	45.8796
		2	10	2.5e-5	4.5848e-5	46.0579
		3	100	2.5e-6	0	46.0579
		4	1000	2.5e-7	0	46.0575
	20	1	1	2.5e-4	0.0959	45.4812
		2	10	2.5e-5	1.6026e-05	45.5316
		3	100	2.5e-6	0	45.5290
		4	1000	2.5e-7	0	45.5285

表 4-5 列出了此问题的相关文献研究成果。数据表明，在相同分段数下，本文结果与文献研究结果类似，甚至更优。由于采用部分移动网格，曲线能在较少的分段数下更好的拟合最优切换时间点，针对此类控制轨迹，分段数为 20 的拟合精度也可以优于分段数为 30 的结果^[95]。值得注意的是，在文献^[98]十段的结果要优于本文，原因在于此文献选用了网格移动方法，在相同分段数下较部分移动网格拥有更高的灵活度，但优化问题维数较部分网格移动问题增加，其非线性更强，则计算时间复杂度更高。

表 4-5 Rayleigh 问题文献对比结果

问题	文献	段数	J
Rayleigh (min)	Loxton <i>et al.</i> [39]	10	46.09620
	Li <i>et al.</i> [98]	10	45.8048
	Liu <i>et al.</i> [95]	30	45.5716

图 4-8 展示了 Rayleigh 问题的最优控制曲线和状态轨迹。由状态轨迹可得，状态变量 x_1 一直在曲线 $-6u$ 的下方，并在初始及末尾时间点取得积极约束。

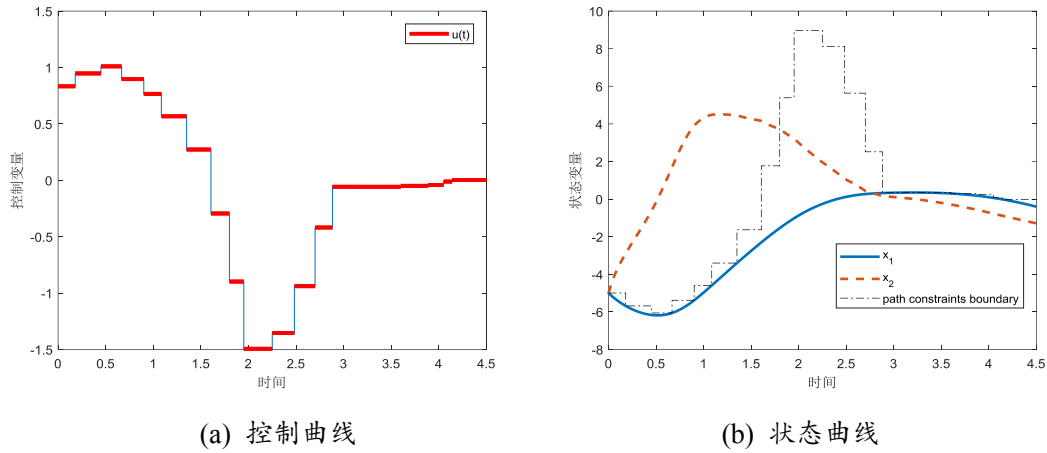


图 4-8 光滑化约束动态优化方法求解 Rayleigh 问题所得最优结果

4.3.2 除铜过程动态优化问题

此小节应用以上所提出了约束动态优化方法求解除铜过程的带状态约束动态优化问题(4-9)，其动态优化过程如图 4-9 所示。从而得到一段时间内锌粉添加的最佳操作轨迹，使得铜离子沉淀到理想的浓度范围。本节采用收集了三个月的实际除铜过程行业数据，验证了该方法的有效性。



图 4-9 除铜过程动态优化流程

湿法炼锌除铜工艺是一个长流程工艺。在实际生产过程中，溶液体积 V 、浸出液入口流量 Q 和返回底流流量 q 可以得到连续性的检测，但反应工况的关键因素，即进出口离子浓度仅能每 2 小时离线化验一次。另外，根据观察，2 小时内的溶液体积、溶液流量在 2 小时内没有明显波动。因此，我们以 2 小时为优化时间区间，即 $t_f - t_0 = 2 \text{ h}$ ，其中 V, Q, q 在 $[t_0, t_f]$ 时间段保持不变。除铜过程的动态优化可以安排为每 2 小时一次，如果存在铜离子浓度出现显著偏差，则将根据当前工况重新计算新的动态优化结果，并在接下来的 2 小时内使用。

首先展示 2 小时内的动态优化结果。 t_0 的工作条件如表 4-6 所示。值得注意的是，二号出口铜离子浓度是整个除铜过程的关键变量，必须严格控制在工业允

许范围, 即 0.2-0.4 g/L。由于生产过程存在操作、测量等误差, 也同时存在原材料成分波动等不确定因素, 为保证系统稳定性, 离子浓度实际控制范围必须从约束边界往后退一定距离, 称之为“回退”, 回退量通常取 0.05 g/L, 因此在这个动态优化过程中, #2 出口铜离子浓度应服从 0.25-0.35 g/L。

表 4-6 t_0 时刻除铜过程的工况

参数	单位	值
浸出 ZnSO ₄ 溶液入口流量, Q	m ³ /h	200
返回底流流量, q	m ³ /h	20
溶液体积, V	m ³	100
#1 入口铜离子浓度, $C_{Cu^{2+},1}^{in}$	g/L	1.7
#1 反应器初始出口铜离子浓度, $C_{Cu^{2+},1}^0$	g/L	0.7
#2 反应器初始出口铜离子浓度, $C_{Cu^{2+},2}^0$	g/L	0.32

表 4-7 列出了动态优化问题求解过程及结果。图 4-9 中, 左图为除铜过程在 t_0 时刻工况下 2 h 内最优控制曲线图, 右图为其对应最优状态曲线, 即出口离子浓度变化图。

表 4-7 除铜过程动态优化问题实验结果

问题	段数	迭代次数	惩罚因子	光滑因子	G	J
	10	1	1	2.5e-4	93.1598	319.1035
CRP		2	10	2.5e-5	1.8493	464.1158
(min)		3	100	2.5e-6	0.0001	466.4650
		4	1000	2.5e-7	0	466.3869

由表 4-7 可以很明显看出, 在初始惩罚因子和光滑化因子下, 目标函数 $J = 319.1035$ 最小, 即锌粉总添加量最少, 但总约束违反度 $G = 93.1598$, 由此表明, 惩罚力度尚且不足; 在第 2 次迭代中加大惩罚因子, 总约束违反度大幅下降, 目标函数值有所上升; 在第 3 次迭代继续加大惩罚因子, 总约束违反度下降, 目标函数微升; 在第 4 此迭代中, 罚因子继续加大, 总约束违反度降至 0, 但目标函数不升反降, 原因在于光滑因子也在同步减小, 不等式约束在积极等式约束处得到进一步逼近, 控制曲线进一步逼近原最优控制曲线。

从表 4-7 和图 4-10 可以看出, 动态优化下 2 小时内锌粉的总量(466.38 kg), 远低于实际工厂此工况下实际使用锌粉的平均量 (510.37 kg)。同时, 应用所提出的方法所得最优控制曲线, 在此控制曲线下, #2 出口铜离子浓度能平稳的处于 0.2~0.4 g/L 之间, 严格满足生产要求。

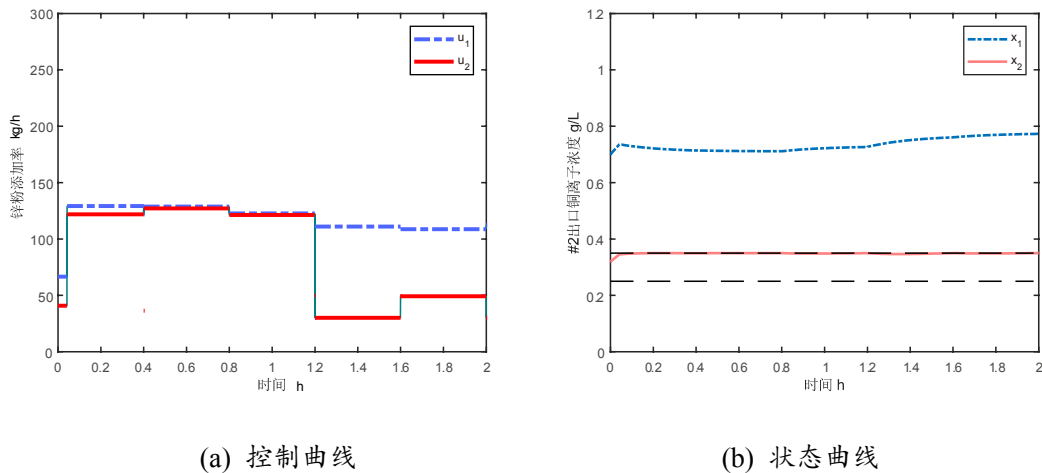


图 4-10 光滑化约束动态优化方法求解 CRP 问题所得最优结果

4.4 本章小结

除铜过程为绿色湿法炼锌工艺净化工段的首要工序，目的是通过添加锌粉将硫酸锌溶液中的杂质铜离子浓度降低到一定范围，其净化效果对炼锌过程效率提升和安全生产有着直接影响。锌粉添加为除铜工艺关键，实际生产主要依赖人工经验，存在出口铜离子浓度振荡甚至超标等现象，使得产品合格率低、锌粉消耗量大。为了满足除铜工艺各种动态物质平衡及工艺要求，同时追求最小原料消耗和出口铜离子浓度稳定，必须进行除铜过程动态优化，以实现企业高效节能生产。首先，本章分析了除铜过程的动态特性，构建动态模型，并选定合适的效用指标、考虑全面的工况约束，表征为符合实际工况的动态优化问题。随后，针对同时此类具有状态约束和控制约束的动态优化问题，引入光滑化罚函数法，提出了一种基于状态转移算法的约束动态优化方法。经典带约束动态优化问题测验结果表明，该方法能有效地求解。数值仿真也证明了该方法的有效性，优化的控制轨迹不仅可以保证出口离子浓度合格，而且可以节约锌耗。

5 结论与展望

5.1 研究工作总结

动态优化是工业过程实现节能降耗、挖潜增效的有力方法，高效、精确的动态优化方法是实现复杂工业过程动态优化的关键。本文针对一般的一类优化时间固定、有约束的工业动态优化问题，开展了相关研究。由于实际工业过程复杂多变，其动态优化问题通常具有非线性、多模态、不连续、多约束等特点，因此基于控制向量参数化的数值型动态优化方法是本文的研究重点。本文在控制变量参数化离散方法的基础上，深入分析了影响动态优化方法效率和精度的三大因素，其一、如何获得离散后多模态非凸 NLP 问题最优解；其二、如何高效的获得更高精度近似最优控制曲线；其三、如何处理带约束动态优化问题。随后，针对以上三大难点，开展具有鲁棒性和高效性动态优化方法及应用研究。本文主要研究内容有以下几个方面：

1) 提出了基于变尺度状态转移算法的均匀划分动态优化方法

针对如何获得离散后 NLP 子问题全局最优解，提出了基于变尺度状态转移算法的均匀划分动态优化方法 (udCVP-vSTA)。本文首先采用基本均匀控制参数化将动态问题转换为静态 NLP 问题，考虑到所得 NLP 问题仍然保留有原动态问题的强非线性、多模态等复杂特性，本文引入全局性智能优化算法状态转移算法，深入分析其性能，并提出了平衡局部及全局收敛能力的变尺度状态转移算法，最后采用变尺度 STA 求解 NLP 问题的基础算法，建立基于变尺度 STA 的均匀划分动态优化方法。经典非线性动态优化问题验证了此方法的优越性。

2) 提出了基于混合状态转移算法的非均匀划分动态优化方法

针对如何高效获得更高精度的近似最优控制曲线，提出了基于混合状态转移算法的非均匀划分动态优化方法 (ndCVP-HSTA)。一方面，考虑到均匀离散网格下近似精度和优化求解效率两者互为矛盾，本文总结了两类的常用的非均匀控制离散网格，提出了有部分移动网格优化策略的 ndCVP 方法，该方法能在较低维度下有较高的自由度来寻求最佳离散网格。另一方面，考虑到数值解的求取精度决定了与近似上限的接近程度，以及非均匀离散导致的 NLP 子问题维数增加、非线性增强等问题，提出了混合梯度状态转移算法 (HSTA)，该方法在搜索前期使用 STA 定位全局最优解，在搜索后期使用梯度算法精细搜索。经典非线性动态优化问题实验表明该方法有更高的效率和精度，特别是在求解 bang-bang 问题上。

3) 提出了基于 ndCVP-HSTA 的光滑约束动态优化方法

针对如何高效求解约束动态优化问题上,提出了基于状态转移算法的光滑约束动态优化方法。本文首先分析并总结了动态优化问题三类常见约束,并引入光滑化罚函数法,在以上无约束动态优化方法的研究基础上,提出了一种求解一般约束动态优化问题的基于 ndCVP-HSTA 的光滑化约束动态优化方法。两个经典带约束动态优化问题求解结果表明,该方法能高效且准确的获得完全可行的最优控制轨迹。

4) 除铜过程动态优化问题构建及求解

除铜过程优化控制是典型的动态优化问题,本文分析了除铜过程动态特性,构建动态模型,并选定合适的效用指标,考虑全面的工况约束,表征为符合实际工况的动态优化问题;随后,采用基于 ndCVP-HSTA 的约束动态优化方法求解此问题;除铜动态优化仿真结果表明,所求最优控制轨迹不仅可以使除铜过程严格满足生产约束,还可以有效节约生产成本。

5.2 后续工作展望

实现更高效高精度工业过程动态优化,还需要进一步的深入研究,总结如下:

(1) 在动态优化方法方面,部分移动网格可以继续结合网格点增删的思想,重新确定网格分段数或固定时间点,并以此分段数或离散网格进行下一次迭代优化,这样可以逐步的逼近最优控制曲线,提高解的近似精度。

(2) 在微分方程求解方面,采用替代函数方法进行局部近似计算,提高迭代求解效率。动态优化问题在离散后每一次进行迭代优化运算过程中,都需要进行微分方程的求解,直至迭代结束,当前国际上通常采用自适应变步长的数值积分方法对微分方程组进行计算,将要花费大量时间。可通过训练代理模型来替代数值微分操作,因此显著提高优化效率。

(3) 在动态优化问题构建方面,研究更具广泛意义的动态优化问题,如模型带不确定参数等。实际工业过程,如除铜过程,通常存在各种干扰等不确定因素,简单的回退操作使得控制轨迹过于保守,因此应开展研究具有不确定动态系统的动态优化问题。

(4) 开展动态优化的闭环控制方法研究。由于动态优化是一种基于模型的开环优化手段,模型与实际的不吻合极易导致求得的最优控制曲线失去最优性,甚至不再可行。因此可以考虑引入反馈思想,开展动态优化的闭环控制方法研究。

参考文献

- [1] 有色金属工业发展规划（2016-2020年）工业与信息化部, 2016年9月28日.
- [2] 丁进良, 杨翠娥, 陈远东, 等. 复杂工业过程智能优化决策系统的现状与展望[J]. 自动化学报, 2018, 44(11): 1931-1943.
- [3] 柴天佑. 复杂工业过程运行优化与反馈控制. 自动化学报, 2013, 39(11): 1744-1757.
- [4] 李国栋, 胡云卿, 刘兴高. 一种高效的快速近似控制向量参数化方法[J]. 自动化学报, 2015, 41(1): 67-74.
- [5] Hartl R F, Sethi S P, Vickson R G. A survey of the maximum principles for optimal control problems with state constraints[J]. SIAM review, 1995, 37(2): 181-218.
- [6] Cervantes A M , Biegler L T . A stable elemental decomposition for dynamic process optimization[J]. Journal of Computational and Applied Mathematics, 2000, 120(1-2):41-57.
- [7] Cots O, Gergaud J, Goubinat D. Direct and indirect methods in optimal control with state constraints and the climbing trajectory of an aircraft[J]. Optimal Control Applications and Methods, 2018, 39(1): 281-301.
- [8] Zhou X, Yang C, Gui W. State transition algorithm[J]. Journal of Industrial & Management Optimization, 2012, 8(4): 1039-1056.
- [9] Zhou X, Shi P, Lim C C, et al. A dynamic state transition algorithm with application to sensor network localization[J]. Neurocomputing, 2018, 273: 237-250.
- [10] Zhang F, Yang C, Zhou X, et al. Fractional order fuzzy PID optimal control in copper removal process of zinc hydrometallurgy[J]. Hydrometallurgy, 2018, 178: 60-76.
- [11] Zhou X, Zhou J, Yang C, et al. Set-point tracking and multi-objective optimization-based PID control for the goethite process[J]. IEEE Access, 2018, 6: 36683-36698.
- [12] Huang Z, Yang C, Zhou X, et al. A hybrid feature selection method based on binary state transition algorithm and Relief-F[J]. IEEE journal of biomedical and health informatics, 2018, doi: 10.1109/JBHI.2018.2872811.
- [13] Zhou X, Long J, Xu C, et al. An External Archive-Based Constrained State Transition Algorithm for Optimal Power Dispatch[J]. Complexity, 2019, doi: 10.1155/2019/4727168.

- [14] Zhou X, Yang K, Xie Y, et al. A novel modularity-based discrete state transition algorithm for community detection in networks[J]. *Neurocomputing*, 2019, 334: 89-99.
- [15] de Pinho M R, Ferreira M M, Ledzewicz U, et al. A model for cancer chemotherapy with state-space constraints[J]. *Nonlinear Analysis: Theory, Methods & Applications*, 2005, 63(5-7): e2591-e2602.
- [16] Sethi S P. The Maximum Principle: Pure State and Mixed Inequality Constraints[M]//*Optimal Control Theory*. Springer, Cham, 2019: 125-158.
- [17] Betts J T, Biehn N, Campbell S L. Convergence of nonconvergent IRK discretizations of optimal control problems with state inequality constraints[J]. *SIAM Journal on Scientific Computing*, 2002, 23(6): 1981-2007.
- [18] 陈珑. 工业应用中的动态优化方法研究[D]. 浙江大学控制科学与工程学系, 2010.
- [19] Biegler L T, Zavala V M. Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization[J]. *Computers & Chemical Engineering*, 2009, 33(3):575-582.
- [20] Kameswaran S , Biegler L T . Simultaneous dynamic optimization strategies: Recent advances and challenges[J]. *Computers & Chemical Engineering*, 2006, 30(10-12):1560-1575.
- [21] Young L C. Orthogonal collocation revisited[J]. *Computer Methods in Applied Mechanics and Engineering*, 2019, 345: 1033-1076.
- [22] Pollard G P, Sargent R W H. Off line computation of optimum controls for a plate distillation column[J]. *Automatica*, 1970, 6(1): 59-76.
- [23] Tanartkit P, Biegler L T. A nested, simultaneous approach for dynamic optimization problems—II: the outer problem[J]. *Computers & chemical engineering*, 1997, 21(12): 1365-1388.
- [24] Tanartkit P, Biegler L T. A nested, simultaneous approach for dynamic optimization problems—I[J]. *Computers & chemical engineering*, 1996, 20(6-7): 735-741.
- [25] Teo K L, Jennings L S, Lee H W J, et al. The control parameterization enhancing transform for constrained optimal control problems[J]. *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics*, 1999, 40(03): 314-335.

- [26] Srinivasan B, Palanki S, Bonvin D. Dynamic optimization of batch processes: I. Characterization of the nominal solution[J]. *Computers & Chemical Engineering*, 2003, 27(1): 1-26.
- [27] Schlegel M, Stockmann K, Binder T, et al. Dynamic optimization using adaptive control vector parameterization[J]. *Computers & Chemical Engineering*, 2005, 29(8): 1731-1751.
- [28] Assassa F, Marquardt W. Dynamic optimization using adaptive direct multiple shooting[J]. *Computers & Chemical Engineering*, 2014, 60: 242-259.
- [29] Chen X, Du W, Tianfield H, et al. Dynamic optimization of industrial processes with nonuniform discretization-based control vector parameterization[J]. *IEEE Transactions on Automation Science and Engineering*, 2014, 11(4): 1289-1299.
- [30] Liu P, Li G, Liu X, et al. Novel non-uniform adaptive grid refinement control parameterization approach for biochemical processes optimization[J]. *Biochemical Engineering Journal*, 2016, 111: 63-74.
- [31] Liu P, Liu X. Empirical mode decomposition-based time grid refinement optimization approach for optimal control problems[J]. *Optimization Letters*, 2017, 11(7): 1243-1256.
- [32] Wang L, Liu X, Zhang Z. A new sensitivity-based adaptive control vector parameterization approach for dynamic optimization of bioprocesses[J]. *Bioprocess and biosystems engineering*, 2017, 40(2): 181-189.
- [33] Jacobson D, Lele M. A transformation technique for optimal control problems with a state variable inequality constraint[J]. *IEEE Transactions on Automatic Control*, 1969, 14(5): 457-464.
- [34] Jacobson D H, Lele M M, Speyer J L. New necessary conditions of optimality for control problems with state-variable inequality constraints[J]. *Journal of mathematical analysis and applications*, 1971, 35(2): 255-284.
- [35] Feehery W F, Barton P I. Dynamic optimization with state variable path constraints[J]. *Computers & chemical engineering*, 1998, 22(9): 1241-1256.
- [36] Xing A Q, Wang C L. Applications of the exterior penalty method in constrained optimal control problems[J]. *Optimal Control Applications and Methods*, 1989, 10(4): 333-345.
- [37] Vassiliadis V S, Sargent R W H, Pantelides C C. Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints[J]. *Industrial & Engineering Chemistry Research*, 1994, 33(9): 2123-2133.

- [38]Chen T W C, Vassiliadis V S. Inequality path constraints in optimal control: a finite iteration ε -convergent scheme based on pointwise discretization[J]. Journal of Process Control, 2005, 15(3): 353-362.
- [39]Loxton R C, Teo K L, Rehbock V, et al. Optimal control problems with a continuous inequality constraint on the state and the control[J]. Automatica, 2009, 45(10): 2250-2257.
- [40]Li B, Yu C J, Teo K L, et al. An exact penalty function method for continuous inequality constrained optimal control problem[J]. Journal of Optimization Theory and Applications, 2011, 151(2): 260.
- [41]Jiang C, Lin Q, Yu C, et al. An exact penalty method for free terminal time optimal control problem with continuous inequality constraints[J]. Journal of Optimization Theory and Applications, 2012, 154(1): 30-53.
- [42]胡云卿, 刘兴高, 薛安克. 带不等式路径约束最优控制问题的惩罚函数法[J]. 自动化学报, 2013, 39(12): 1996-2001.
- [43]Liu X, Hu Y, Feng J, et al. A novel penalty approach for nonlinear dynamic optimization problems with inequality path constraints[J]. IEEE Transactions on automatic control, 2014, 59(10): 2863-2867.
- [44]Gao X, Zhang X, Wang Y. A simple exact penalty function method for optimal control problem with continuous inequality constraints[C]//Abstract and Applied Analysis. Hindawi, 2014, 2014.
- [45]Büsken C, Maurer H. SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control[J]. Journal of computational and applied mathematics, 2000, 120(1-2): 85-108.
- [46]Cruz I L L, Van Willigenburg L G, Van Straten G. Efficient differential evolution algorithms for multimodal optimal control problems[J]. Applied Soft Computing, 2003, 3(2): 97-122.
- [47]Davis L. Handbook of genetic algorithms[J]. 1991.
- [48]Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of global optimization, 1997, 11(4): 341-359.
- [49]Kennedy J. Particle swarm optimization[J]. Encyclopedia of machine learning, 2010: 760-766.

- [50]Faber R, Jockenhövel T, Tsatsaronis G. Dynamic optimization with simulated annealing[J]. *Computers & chemical engineering*, 2005, 29(2): 273-290.
- [51]Pham Q T. Using statistical analysis to tune an evolutionary algorithm for dynamic optimization with progressive step reduction[J]. *Computers & Chemical Engineering*, 2007, 31(11): 1475-1483.
- [52]Asgari S A, Pishvaie M R. Dynamic optimization in chemical processes using region reduction strategy and control vector parameterization with an ant colony optimization algorithm[J]. *Chemical Engineering & Technology: Industrial Chemistry - Plant Equipment - Process Engineering - Biotechnology*, 2008, 31(4): 507-512.
- [53]Pham Q T. Using fuzzy logic to tune an evolutionary algorithm for dynamic optimization of chemical processes[J]. *Computers & Chemical Engineering*, 2012, 37: 136-142.
- [54]刘宗其. 文化算法在化工动态优化中的应用研究[D]. 华东理工大学, 2011.
- [55]刘宗其, 杜文莉, 祁荣宾, 等. 基于知识改进的文化算法及其在化工动态优化中的应用[J]. *化工学报*, 2010, 61(11): 2889-2895.
- [56]彭鑫, 祁荣宾, 杜文莉, 等. 一种改进的知识进化算法及其在化工动态优化中的应用[J]. *化工学报*, 2012, 63(3): 841-850.
- [57]Wongrat W, Younes A, Elkamel A, et al. Control vector optimization and genetic algorithms for mixed-integer dynamic optimization in the synthesis of rice drying processes[J]. *Journal of the Franklin Institute*, 2011, 348(7): 1318-1338.
- [58]孙帆, 杜文莉, 钱锋. 一种改进的差分进化算法及其在补料分批式生化反应器动态优化中的应用[J]. *化工学报*, 2012, 63(11).
- [59]Qian F, Sun F, Zhong W, et al. Dynamic optimization of chemical engineering problems using a control vector parameterization method with an iterative genetic algorithm[J]. *Engineering Optimization*, 2013, 45(9): 1129-1146.
- [60]Chen X, Du W, Qi R, et al. Hybrid gradient particle swarm optimization for dynamic optimization problems of chemical processes[J]. *Asia - Pacific Journal of Chemical Engineering*, 2013, 8(5): 708-720.
- [61]张盼盼. 群智能动态优化方法研究[D]. 2016.
- [62]You Z, Liu X. Control Parameterization-Based Adaptive Particle Swarm Approach for Solving Chemical Dynamic Optimization Problems[J]. *Chemical Engineering & Technology*, 2014, 37(4):692-702.
- [63]李国栋. 基于控制向量参数化的动态优化研究[D]. 浙江大学, 2015.

- [64]Lin Q, Loxton R, Teo K L. The control parameterization method for nonlinear optimal control: a survey[J]. *Journal of Industrial and Management Optimization*, 2014, 10(1): 275-309.
- [65]Teo K L, Goh C, Wong K. A unified computational approach to optimal control problems[M]. 1991.
- [66]Loxton R, Lin Q, Rehbock V, et al. Control parameterization for optimal control problems with continuous inequality constraints: New convergence results[J]. *Numerical Algebra, Control and Optimization*, 2012, 2(3): 571-599.
- [67]Zhou X, Yang C, Gui W. A statistical study on parameter selection of operators in continuous state transition algorithm[J]. *IEEE transactions on cybernetics*, 2018, doi: 10.1109/TCYB.2018.2850350.
- [68]Han J, Yang C, Zhou X, et al. A two-stage state transition algorithm for constrained engineering optimization problems[J]. *International Journal of Control, Automation and Systems*, 2018, 16(2): 522-534.
- [69]Zhou X, Gao D Y, Yang C, et al. Discrete state transition algorithm for unconstrained integer optimization problems[J]. *Neurocomputing*, 2016, 173: 864-874.
- [70]Dong T, Yang C, Zhou X, et al. A novel discrete state transition algorithm for staff assignment problem[J]. *Control Theory & Applications*, 2016, 33(10): 1378-1388.
- [71]Huang Z , Yang C , Zhou X , et al. A Novel Cognitively Inspired State Transition Algorithm for Solving the Linear Bi-Level Programming Problem[J]. *Cognitive Computation*, 2018, 10(1).
- [72]Zhang F, Yang C, Zhou X, et al. Fractional-order PID controller tuning using continuous state transition algorithm[J]. *Neural Computing and Applications*, 2018, 29(10): 795-804.
- [73]Han J, Yang C, Zhou X, et al. A new multi-threshold image segmentation approach using state transition algorithm[J]. *Applied Mathematical Modelling*, 2017, 44: 588-601.
- [74]Han J, Yang C, Zhou X, et al. Dynamic multi-objective optimization arising in iron precipitation of zinc hydrometallurgy[J]. *Hydrometallurgy*, 2017, 173: 134-148.
- [75]Tran T D, Jin G G. Real-coded genetic algorithm benchmarked on noiseless black-box optimization testbed[C]// *Conference Companion on Genetic & Evolutionary Computation*. 2010.

- [76] Karaboga D, Ozturk C. A novel clustering approach: Artificial Bee Colony (ABC) algorithm[J]. Applied Soft Computing Journal, 2011, 11(1):652-657.
- [77] Iadevaia S, Lu Y, Morales F C, et al. Identification of Optimal Drug Combinations Targeting Cellular Networks: Integrating Phospho-Proteomics and Computational Network Analysis[J]. Cancer Research, 2010, 70(17):6704-6714.
- [78] Dadebo S A, McAuley K B. Dynamic optimization of constrained chemical engineering problems using dynamic programming[J]. Computers & chemical engineering, 1995, 19(5): 513-525.
- [79] Cruz I L L, Willigenburg L G V, Straten G V. Efficient Differential Evolution algorithms for multimodal optimal control problems[J]. Applied Soft Computing, 2003, 3(2):97-122.
- [80] 陈旭. 基于进化计算的动态多目标优化方法研究[D]. 华东理工大学, 2015.
- [81] Sarkar D, Modak J M. Optimisation of fed-batch bioreactors using genetic algorithms[J]. Chemical Engineering Science, 2003, 58(11): 2283-2296.
- [82] Sun D Y, Lin P M, Lin S P. Integrating controlled random search into the line-up competition algorithm to solve unsteady operation problems[J]. Industrial & Engineering Chemistry Research, 2008, 47(22): 8869-8887.
- [83] Teo K L, Goh C, Wong K. A unified computational approach to optimal control problems[M]. 1991.
- [84] Kapadi M D, Gudi R D. Optimal control of fed-batch fermentation involving multiple feeds using differential evolution[J]. Process Biochemistry, 2004, 39(11): 1709-1721.
- [85] Waldraff W, King R, Gilles E D. Optimal feeding strategies by adaptive mesh selection for fed-batch bioprocesses[J]. Bioprocess Engineering, 1997, 17(4): 221-227.
- [86] 王立伟. 自适应控制向量参数化动态优化研究[D]. 2017.
- [87] Angira R, Santosh A. Optimization of dynamic systems: A trigonometric differential evolution approach[J]. Computers & Chemical Engineering, 2007, 31(9): 1055-1063.
- [88] Bayón L, Grau J M, Ruiz M M, et al. Initial guess of the solution of dynamic optimization of chemical processes[J]. Journal of mathematical chemistry, 2010, 48(1): 28-37.

- [89] Vassiliadis V S , Canto E B , Banga J R . Second-Order Sensitivities of General Dynamic Systems with Application to Optimal Control Problems[J]. Chemical Engineering Science, 1999, 54(17):3851-3860.
- [90] 李若贵. 常压富氧直接浸出炼锌[J]. 中国有色冶金, 2009, 3: 12-15.
- [91] 张凤雪, 阳春华, 周晓君, 等. 基于控制周期计算的锌液净化除铜过程优化控制[J]. 控制理论与应用, 2017, 34(10): 1388-1395.
- [92] Zhang B, Yang C, Zhu H, et al. Kinetic modeling and parameter estimation for competing reactions in copper removal process from zinc sulfate solution[J]. Industrial & Engineering Chemistry Research, 2013, 52(48): 17074-17086.
- [93] 胡云卿, 刘兴高. 处理动态优化问题中控制变量路径约束的方法[J]. 自动化学报, 2013, 39(4): 440-449.
- [94] Mehra R, Davis R. A generalized gradient method for optimal control problems with inequality constraints and singular arcs[J]. IEEE Transactions on Automatic Control, 1972, 17(1): 69-79.
- [95] Liu P, Li X, Liu X, et al. An improved smoothing technique - based control vector parameterization method for optimal control problems with inequality path constraints[J]. Optimal Control Applications and Methods, 2017, 38(4): 586-600.
- [96] Logsdon J S, Biegler L T. Accurate solution of differential-algebraic optimization problems[J]. Industrial & engineering chemistry research, 1989, 28(11): 1628-1639.
- [97] Gerdts M. Global convergence of a nonsmooth Newton method for control-state constrained optimal control problems[J]. SIAM Journal on Optimization, 2008, 19(1): 326-350.
- [98] Li B , Yu C J , Teo K L , et al. An Exact Penalty Function Method for Continuous Inequality Constrained Optimal Control Problem[J]. Journal of Optimization Theory & Applications, 2011, 151(2):260-291.

攻读学位期间主要的研究成果

发表论文情况

- [1] Zhou X J, **Huang M**, Huang T W, Yang C H, Gui W H. Dynamic optimization for copper removal process with continuous production constraints [J]. IEEE Transactions on Industrial Informatics, 2019. (SCI Q1, IF:5.634, 二审)
- [2] Wang Q A, **Huang M**, Zhou X J. Feature selection in froth flotation for production condition recognition[J]. IFAC-PapersOnLine, 2018, 51(21): 123-128. (EI期刊)
- [3] **Huang M**, Zhou X J, Huang T W, Yang C H, Gui W H. Dynamic optimization based on state transition algorithm for copper removal process[J]. Neural Computing and Applications, 2017, DOI: 10.1007/s00521-017-3232-0. (SCI Q1, IF:4.215)
- [4] **Huang M**, Zhou X, Yang C, Gui W H. Dynamic optimization using control vector parameterization with state transition algorithm[C]. 2017 36th Chinese Control Conference (CCC). IEEE, 2017: 4407-4412. (EI会议)

参与科研项目情况

- [1] 2016年9月-至今, 参与国家自然科学基金青年科学基金“基于分布式状态转移算法的有色冶金过程系统辨识方法研究”(61503416)的研究工作
- [2] 2017年5月-2019年3月, 主持中南大学研究生自主探索创新项目(2017zzts487)的研究工作
- [3] 2018年1月-至今, 参与中南大学创新驱动计划“流程工业过程数据建模与分布式优化方法研究”(2018CX012)的研究工作
- [4] 2019年1月-至今, 参与国家自然科学基金面上项目“有色冶金级联反应器不确定动态优化方法”(61873285)的研究工作

获奖情况

- [1] 2019年“国家奖学金”、第十四届“中国研究生数学建模竞赛二等奖”、2019年“湖南省优秀毕业研究生”
- [2] 2019年“中南大学优秀毕业研究生”、2018年“中南大学信息学院优秀学术年会论文”、2017-2019年连续两年获得“中南大学优秀研究生”、2016-2019年连续三年获得“国家研究生学业奖学金一等奖”

致谢

时间的魅力在于他的一往无前。

特别感谢我的导师周晓君老师。周老师治学严谨、勤学敏思、细心负责、言传身教，让我度过了一个愉快而充实的三年研究生生活。在科研方面，周老师总是悉心指导，循循善诱，授之以渔，不仅激发了研究思路，同时培养了研究能力，也赋予了解决问题的勇气。

同时，由衷的感谢桂卫华老师、阳春华老师、谢永芳老师、李勇刚老师，徐德刚老师、王雅琳老师、蒋朝辉老师、王晓丽老师、唐朝晖老师、陈晓方老师、朱红求老师、罗彪老师、陈志文老师、胡文峰老师、林鹏老师、马山老师和控制工程研究所其他老师的辛勤付出，

感谢课题组韩洁师姐、张凤雪师姐、黄兆可师兄和赵婷婷师姐，感谢同门周佳佳，感谢王启安师弟、龙建朋师妹、杨珂师弟、张润东师弟、徐冲冲师弟、张云祥师弟、王湘月师弟等其他师弟师妹们，感谢七年同窗王浩，你们的鼓励与陪伴是我学习成长路上最美的风景。

感谢室友吕星晓、蒋珂、崔梦倩和胡汀师妹，每晚寝室的欢声笑语是研究生涯最好的调味剂。

感谢家人和男朋友，你们真挚且坚定的支持和鼓励，是我努力和前进的动力。

感谢中南大学，在这里我遇到了很多美好而有趣的人儿，“知行合一，经世致用”的校训我将永远铭记。

凡是过往，皆塑造自己，凡是现在，必脚踏实地，凡是未来，应秉承希望。

黄淼

2019年5月