中国科学: 数学　2016 年　第 46 卷　第 1 期: 1～20
SCIENTIA SINICA　Mathematica
《中国科学》杂志社
SCIENCE CHINA PRESS
自然科学基金项目进展专栏　综　述

混合整数非线性规划的算法软件及最新进展

刘明明[1], 崔春风[2], 童小娇[3][1], 戴彧虹[2]*

① 湘潭大学数学与计算科学学院, 湘潭 411105;
② 中国科学院数学与系统科学研究院, 北京 100190;
③ 湖南第一师范学院数学与计算科学学院, 长沙 410000
E-mail: mingming415114@163.com, cuichf@lsec.cc.ac.cn, xjtong-csust@hotmail.com, dyh@lsec.cc.ac.cn

收稿日期: 2014-12-29; 接受日期: 2015-06-18; 网络出版日期: 2015-12-21; * 通信作者
国家自然科学基金 (批准号: 11171095, 71371065, 11331012 和 81173633) 和国家杰出青年科学基金 (批准号: 11125107) 资助项目

摘要　混合整数非线性规划 (mixed integer nonlinear programming, MINLP) 已经渗入到了实际生活中的各个领域, 其研究有着重要的现实意义. 为有效求解不同类型的 MINLP 问题, 研究者们不断提出新的算法和有效软件. 本文致力于介绍求解 MINLP 问题的基本算法与相应的优化软件, 并介绍 MINLP 问题的研究进展.

关键词　混合整数非线性规划　分支定界　割平面　软件

MSC (2010) 主题分类　01-02, 90C11, 97N80

1　引言

科学与工程等领域中的很多优化决策问题都包括影响最终设计质量的离散变量和非线性系统. 混合整数非线性规划 (mixed integer nonlinear programming, MINLP) 就是包含这两大挑战的一类问题. 最近几十年, 应用领域对 MINLP 的需求促使其研究十分受欢迎, 其应用领域包括: 水资源管理和共享[1], 设计、组合及控制相互作用领域[2], 在不定条件下的过程组合和设计应用[3,4], 物流基地布局优化[5], 电力市场机组组合问题[6], 化工生产的计划和调度问题等. 关于 MINLP 在实际生活中的应用, 参见文献 [7]. 可见如何有效求解 MINLP 问题颇有现实意义.

MINLP 是一类包含连续与离散变量的非线性规划 (nonlinear programming, NLP) 问题. 一般情形下, MINLP 模型可以表述为以下形式:

$$\begin{aligned} Z_{\mathrm{MINLP}} = \min\ & f(x,y), \\ \text{s.t.}\ & g_j(x,y) \leqslant 0, \quad j=1,\dots,m, \\ & x \in X, \quad y \in Y \cap \mathbb{Z}^p, \end{aligned} \tag{1.1}$$

其中函数 f: $\mathbb{R}^{n\times p} \to \mathbb{R}$, g_j: $\mathbb{R}^{n\times p} \to \mathbb{R}$, n 和 p 分别是连续变量 x 和整数变量 y 的维数, X 和 Y 分别是 $\mathbb{R}^n$ 和 $\mathbb{R}^p$ 中的多面体子集, Y 是有界的. 本文假设 f 和 g 是二次连续可微分的, 但不对函数 f 和 g

英文引用格式: Liu M M, Cui C F, Tong X J, et al. Algorithms, softwares and recent developments of mixed integer nonlinear programming (in Chinese). Sci Sin Math, 2016, 46: 1–20, doi: 10.1360/N012014-00278

的凸性作过多假设. 为了更清楚表达, 定义向量值函数

$$g(x,y) = (g_1(x,y), g_2(x,y), \dots, g_m(x,y))^{\mathrm{T}},$$

下面称它为 MINLP 原问题.

MINLP 的研究可以说是在混合整数线性规划 (mixed integer linear programming, MILP) 研究成果的基础上发展起来的, 其中包括 MILP [8–12] 和 0-1 MINLP [13–15], 以及 MIQCP (mixed integer quadratic constrained programming) 等特殊形式问题. 我国学者孙小玲与李端在混合整数规划方面有一些理论与综述性文章, 如文献 [16, 17]. 但本文仅侧重介绍一般 MINLP. MINLP 是规划领域中最难求解的问题之一, 属于 NP- 难问题[18]. 由于 MINLP 含有整数变量, 使得求解 NLP 的很多优秀算法都不能直接求解 MINLP. 因此, 虽然自 20 世纪 80 年代开始 (参见文献 [9, 16]), 尤其是在越来越多实际问题中的应用, 使得 MINLP 受到了国内外很多学者的广泛关注. 但是相对于 MILP 问题, 在理论和算法上还有较大的差距, 且仍然有很多瓶颈, 如很难判断当前解是否最优. 关于 MINLP 的发展情形, 感兴趣的读者可参见文献 [19–22]. 近年来, MINLP 算法理论的研究及相关软件的开发有了很大的进展, 参见文献 [23–25].

本文致力于为求解 MINLP 问题提供强有力的工具, 即提供有助于找到高质量可行解的算法和软件. 首先, 第 2 节阐述了求解凸 MINLP 问题常用的六种确定型算法与两种启发式算法. 第 3 节介绍了求解非凸 MINLP 问题的两种工具, 以及直接求解非凸 MINLP 问题的算法. 确定型算法是将求解困难的 MINLP 问题分解为简单易处理的问题. 而启发式算法是在可接受时间与占用空间等前提下快速找到可行解, 但不保证可行解的最优性. 对于更难求解的非凸 MINLP 问题 (如其约束是双线性、三线性), 我们首先提供了两种常用处理非凸性的方法: 完全重构与凸性化. 在处理非凸 MINLP 时需要注意的是对整数变量进行松弛后的非凸 NLP. 因为在求解这类 NLP 问题时, (可能) 通常只能得到局部最优解. 经过处理的非凸 MINLP 问题就可以选择凸 MINLP 的算法进行求解. 然后给出了可以直接求解非凸 MINLP 问题的算法 (如 α- 分支定界算法 (α-branch-and-bound, α-BB)). 第 4 节分开源软件与商业软件两大类概括 MINLP 的计算软件, 包括常用的开源软件 BONMIN, COUENNE, LaGO, SCIP 及商业软件 αECP, BARON, DICOPT, KNITRO; 包括这些软件的开发者, 可以执行的算法, 软件的统一资源定位符 (uniform resource locator, 缩写为 URL), 在使用中依赖的其他软件. 最后, 第 5 节对 MINLP 发展进行简单总结并对未来研究发展趋势进行展望.

2 凸混合整数非线性规划的算法

凸 MINLP 要求目标函数及约束条件均是凸的, 即对整数变量进行松弛后得到连续凸问题. 一方面, 凸 NLP 子问题可以求得全局最优解; 另一方面, 凸约束的线性近似恰为其外逼近, 使得凸 MINLP 有较好的理论性质. 本节重点介绍求解凸 MINLP 的六种确定型算法和两种常用启发式算法. 确定型算法是将求解困难的 MINLP 问题分解为简单易处理的问题, 通常采用迭代法和分支定界两种基本思想. 启发式算法是在可接受时间、内存空间等前提下快速找到可行解, 但并不保证可行解的最优性. 事实上, 二者并非独立, 很多确定型算法往往需要利用启发式算法帮助快速找到可行解.

这里需要指明的是, 在求解 MINLP 问题的过程中, 对问题进行预处理是相当重要的. 预处理的目的是得到与原问题等价且更紧凑更容易处理的问题. 一般容易扩展到 MINLP 的预处理方法包括系数和变量的上下界的缩紧处理、删除冗余变量和冗余约束等. 在此不再详细介绍, 读者可参见文献 [19, 26].

2.1 子问题

求解 MINLP 问题的算法的基本思想是产生和改善最优解的界限. MINLP 最优解的下界 (对偶界) 由 MINLP 的松弛子问题提供. 常见的松弛策略包括对整型变量松弛到连续空间 (NLP 子问题) 和非线性约束的线性化 (MILP 子问题). 最优解的上界 (原始界) 由 MINLP 的可行解提供. 虽然不同的算法构造的子问题、提供上下界的方式不尽相同, 然而其中一些算法具有共同的基本理论, 下面进行详细描述.

2.1.1 MILP 子问题

如果 MINLP 原问题 (1.1) 的目标函数是非线性的, 它的最优解有可能是可行域凸包中的内点, 这类问题不宜直接求解. 常用的处理方法是引入辅助变量 η, 将非线性目标函数转化为线性目标函数, 并将目标函数转移到约束中, 得到与 (1.1) 等价的 MINLP 问题

$$
\begin{aligned}
Z_{\mathrm{MINLP}} = \min\ & \eta, \\
\text{s.t.}\ \ & f(x,y) \leqslant \eta, \\
& g(x,y) \leqslant 0, \\
& x \in X, \quad y \in Y \cap \mathbb{Z}^p, \quad \eta \in \mathbb{R}.
\end{aligned}
$$

求解 MINLP 的很多算法都依赖于 MINLP 的目标函数与约束函数的线性逼近. 设当前点为 $(\hat{x},\hat{y})$, 由于 f 和 g 都是凸函数, 故不等式

$$
\begin{aligned}
f(\hat{x},\hat{y}) + \bigtriangledown f(\hat{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\hat{x} \\ y-\hat{y} \end{pmatrix} \leqslant f(x,y), \\
g(\hat{x},\hat{y}) + \bigtriangledown g(\hat{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\hat{x} \\ y-\hat{y} \end{pmatrix} \leqslant g(x,y)
\end{aligned}
$$

成立. 因此, 非线性约束条件

$$
f(x,y) \leqslant \eta, \quad g(x,y) \leqslant 0
$$

在 $(\hat{x},\hat{y})$ 点处可分别松弛为线性不等式

$$
f(\hat{x},\hat{y}) + \bigtriangledown f(\hat{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\hat{x} \\ y-\hat{y} \end{pmatrix} \leqslant \eta, \tag{2.1}
$$

$$
g(\hat{x},\hat{y}) + \bigtriangledown g(\hat{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\hat{x} \\ y-\hat{y} \end{pmatrix} \leqslant 0. \tag{2.2}
$$

此时, 可得 (1.1) 的一种 MILP 子问题

$$
\begin{aligned}
Z_{\mathrm{MINLP}} = \min\ & \eta, \\
\text{s.t.}\ \ & f(\hat{x},\hat{y}) + \bigtriangledown f(\hat{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\hat{x} \\ y-\hat{y} \end{pmatrix} \leqslant \eta, \\
& g(\hat{x},\hat{y}) + \bigtriangledown g(\hat{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\hat{x} \\ y-\hat{y} \end{pmatrix} \leqslant 0, \\
& x \in X, \quad y \in Y \cap \mathbb{Z}^p, \quad \eta \in \mathbb{R}.
\end{aligned} \tag{2.3}
$$

值得指出的是, 对 $g(x,y)$ 线性化是外逼近可行域, 对 $f(x,y)$ 线性化是对目标函数进行了低估. 因此, 通常称 (2.1) 和 (2.2) 为 (1.1) 的外逼近割平面.

2.1.2 NLP 子问题

MINLP 的另一种松弛子问题是将整数变量松弛到连续空间. 假设 y_i 的每个分量满足界限 $l_i \leqslant y_i \leqslant u_i$. 为了便于描述, 令

$$I = \{1, \ldots, p\}, \quad (l_I, u_I) = \{(l_i, u_i) \mid i \in I\},$$

得 (1.1) 的一种 NLP 松弛子问题

$$\begin{aligned} Z_{\mathrm{NLPR}(l_I,u_I)} = \min\ & f(x,y), \\ \text{s.t.}\ & g(x,y) \leqslant 0, \\ & x \in X, \quad l_I \leqslant y \leqslant u_I. \end{aligned} \tag{2.4}$$

若 (l_I, u_I) 是初始问题 (1.1) 可行域的上下界限, 则求解 (2.4) 得到的 $Z_{\mathrm{NLPR}(l_I,u_I)}$ 是 Z_{MINLP} 的有效下界, 而 (1.1) 的上界由可行解提供. 在算法执行过程中, 若相信整数点 $\hat{y}$ 是原问题的一个好点, 则固定整数变量 $y = \hat{y}$, 得 NLP 子问题

$$\begin{aligned} Z_{\mathrm{NLP}(\hat{y})} = \min_x\ & f(x,\hat{y}), \\ \text{s.t.}\ & g(x,\hat{y}) \leqslant 0, \\ & x \in X. \end{aligned} \tag{2.5}$$

如果 (2.5) 是可行的, 则 (2.5) 为 (1.1) 提供上界; 否则 NLP 求解器一般会求解一个相关可行子问题. 可行子问题的一种构造方式为

$$\begin{aligned} Z_{\mathrm{NLPF}(\hat{y})} = \min\ & \sum_{i=1}^{m} \mu_i, \\ \text{s.t.}\ & g(x,\hat{y}) \leqslant \mu, \\ & \mu \geqslant 0, \\ & x \in X, \quad \mu \in \mathbb{R}^m, \end{aligned} \tag{2.6}$$

其中 μ_i 表示约束 $g_i(x,\hat{y})$ 的违反度. 一般情形下, 若 $Z_{\mathrm{NLPF}(\hat{y})} = 0$, 则 (2.6) 的解为 (1.1) 的可行解. 当子问题 (2.5) 不可行时, NLP 求解器就会返回 (2.6) 的解.

2.2 凸混合整数非线性规划问题的确定型算法

求解凸 MINLP 的算法通常采用迭代法和分支定界法 (branch-and-bound, BB) 两种思想. 迭代法是不断更新子问题和迭代点列, 直到算法收敛. 由于 NLP 子问题是凸的, 可以求得全局最优解, MILP 的 BB 算法稍加修改就可以推广到 MINLP, 并且取得了较好的数值效果. BB 算法的基本思想是生成分支定界树和上下界序列. 在树的每个节点处, 都要求解一个子问题. 对应的解将与上下界序列比较, 若不能产生更好的解, 则进行剪枝. 直到上下界相等或者没有更多的节点时, 算法终止. 本节将介绍六种主要确定型算法.

(1) 非线性规划 - 分支定界算法. 该算法于 1965 年由 Dakin [27] 首次应用到求解凸 MINLP 问题.

(2) 广义 Benders 分解算法. 该算法是 1972 年 Geoffrion [28] 首次利用非线性凸对偶理论对 Benders 分解进行推广, 将其用于求解 MINLP.

(3) 外逼近算法. 该算法于 1986 年由 Duran 和 Grossmann [29] 提出, 并且求解了一类特殊类型 MINLP 问题.

(4) 基于线性/非线性 - 分支定界算法. 该算法是 1992 年 Quesada 和 Grossmann [30] 为了改善 0-1 MINLP 问题提出的.

(5) 扩展割平面算法. 1995 年, Westerlund 和 Petterson [31] 将割平面进行扩展, 并求解了凸 MINLP 问题.

(6) 混合算法. 2008 年, Bonami 等人 [32] 提出将基于线性/非线性 - 分支定界算法与外逼近算法结合到一起形成了混合算法.

2.2.1 非线性规划 - 分支定界算法

非线性规划 - 分支定界算法 (NLP-branch-and-bound, NLP-BB) 是求解 MILP 的分支定界算法 (BB) 在 MINLP 问题中的直接推广. 1960 年, Land 和 Doig [33] 首次将 BB 算法应用到 MILP 问题. 1965 年, Dakin [27] 第一个提出 BB 算法可以应用到凸 MINLP 问题, 推动了 BB 算法的发展. 但只给出了算法理论, 没有给出计算实例. 1985 年, Gupta 和 Ravindran [34] 也将此算法应用到了凸 MINLP 问题, 并且研究了算法可行性, 给出了运算过程.

NLP-BB 算法首先要对分支定界树初始化, 一般仅包含一个根节点, 即将整数变量 $\hat{y} \in Y \cap \mathbb{Z}^p$ 松弛到 $l_I \leqslant \hat{y} \leqslant u_I$.

第 1 步 节点选择. 搜索分支定界树, 选择某一节点, 在此节点处求解 NLP 问题. 若此问题不可行, 删除此节点并重新搜索分支定界树; 否则, 设得到解 $(\hat{x}, \hat{y})$.

第 2 步 剪枝. 若在此点处的目标函数大于当前上界, 这部分可行域显然不包含最优解, 则剪枝.

第 3 步 分支. 主要对整数约束变量 $\hat{y}$ 进行检验. 若点 $\hat{y}$ 不满足整数约束条件, 不妨设 $\hat{y}_i$ 为小数, 则问题分支为左右两个子节点, 分别添加左分支约束 $y_i \leqslant \lfloor \hat{y}_i \rfloor$ 及右分支约束 $y_i \geqslant \lfloor \hat{y}_i \rfloor + 1$; 否则, 若 $\hat{y}$ 满足整数约束条件, 且目标函数值小于当前的最优值, 则更新上界, 并且剪掉目标函数值大于当前上界的分支.

第 4 步 检查分支定界树是否为空. 若分支定界树非空, 返回第 1 步; 否则, 算法终止并输出当前最优解.

NLP-BB 算法有三个关键步骤: 分支、节点选择和剪枝. 所谓的分支就是将可行域逐次分割为越来越小的子集, 通常采用的分支准则包括强分支、伪费用分支、可信性分支和混合分支等 (参见文献 [34–37]). 常用的节点选择策略包括深度优先搜索、最佳优先搜索、最佳估计搜索以及它们的结合策略 [12, 34, 35]. 剪枝就是当算法满足当前目标函数值大于当前最好上界时, 或者 NLP 子问题的解 $\hat{y}$ 恰好是整数等情形, 不需要对当前节点进一步分支时进行的操作.

NLP-BB 算法的基本思想与用 BB 算法求解 MILP 问题类似. 但是在每个节点处都要求解 NLP 问题, 需要很大的计算代价. 这也表明了求解 MINLP 比 MILP 有本质上的困难: 求解 MILP 问题时, 所有的连续松弛子问题都是线性规划 (linear programming, LP), 而 MINLP 的连续松弛子问题都是 NLP, 其求解速度和解的质量依赖于 NLP 求解.

近几十年, 在分支定界算法框架的基础上诞生了数个变体, 如空间分支定界算法 [38, 39]、分支下降算法 [40, 41] 和分支定价算法 [42]. 这些算法被应用于各个领域, 使 BB 算法得到更加广泛的利用.

2.2.2 外逼近算法

外逼近 (outer approximation, OA) 算法的理论基础是凸 MINLP 问题可以等价于一个有限维的

MILP 问题. 1986 年, Duran 和 Grossmann [29] 提出用 OA 算法求解特殊类型的 MINLP 问题, 并给出了算法的收敛性和最优性证明. 1994 年, Fletcher 和 Leyffer [43] 将 OA 算法应用于工程领域, 并给出新颖而简单的有限终止定理. 他们的研究进一步推进了 OA 算法理论发展并将其成功推广到了应用中.

OA 算法主要利用的数学工具是映射、外逼近和松弛. 它属于迭代算法, 首先在根节点求解 MINLP 的连续松弛问题 (2.4) 并初始化线性化点集 T. 它在每个迭代步都要求解 NLP 问题和 MILP 问题:

(1) 固定整数变量 $\hat{y}_k$ 得到 NLP 子问题 (2.5) 或 (2.6), 求解 NLP 问题得到连续变量 $\bar{x}_k$ 并试图更新上界.

(2) 用 $(\bar{x}_k, \hat{y}_k)$ 更新 T, 即 $T = T \cup \{(\bar{x}_k, \hat{y}_k)\}$, 构造和求解 MILP 问题

$$\begin{aligned} Z_{\mathrm{OA}} = \min\ & \eta, \\ \text{s.t.}\ & f(\bar{x},\hat{y}) + \triangledown f(\bar{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\bar{x} \\ y-\hat{y} \end{pmatrix} \leqslant \eta, \quad (\bar{x},\hat{y}) \in T, \\ & g(\bar{x},\hat{y}) + \triangledown g(\bar{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\bar{x} \\ y-\hat{y} \end{pmatrix} \leqslant 0, \quad (\bar{x},\hat{y}) \in T, \\ & x \in X, \quad y \in Y \cap \mathbb{Z}^p, \end{aligned} \tag{2.7}$$

得到整数解 $\hat{y}_{k+1}$ 并更新下界.

(3) 若上下界的差值满足给定的容忍度, 则终止; 否则, 返步 (1).

由文献 [32] 中的相关定理证明可知, 假设 (2.5) 或 (2.6) 满足 KKT 条件, 则上述算法要么收敛到 (1.1) 的最优解, 要么表明 (1.1) 不可行. 值得指出的是, 在每次迭代中, 上界和下界只是可能会得到改善. 因为只有当 (2.5) 可行并且它的最优值小于当前上界时, (1.1) 的上界才会得到更新; 否则, 若 (2.5) 不可行, 求解 (2.6) 并不能为原问题提供有效界. 另外, 由于在每一次迭代中添加新的 OA 割使 (2.7) 的定义发生了变化, 导致上一次迭代产生的解不可行, 故利用当前解对界限进行更新.

OA 算法已经用到 IDCOPT 和 BONMIN 等软件包中. 此外, OA 也可用于求解随机最短路径中的锥二次规划模型 [44], 通过一系列赋值等处理后, 数值试验结果表明 OA 算法要比标准整数规划算法好很多. 这推动了 OA 算法在理论及应用方面的进展.

2.2.3 广义 Benders 分解

广义 Benders 分解 (general Benders decomposition, GBD) 最初是 1962 年 Benders [45] 为求解 MILP 问题提出的, 其基本思想是将 MILP 模型分解为一系列 LP 模型和纯整数规划 (integer programming, IP) 模型, 并且一般只用于求解小规模问题. 1972 年, Geoffrion [28] 首次利用非线性凸对偶理论对 Benders 分解进行推广, 将其应用于 MINLP. 1995 年, Floudas [7] 详细给出了 GBD 的基本理论和过程, 进一步将 Benders 分解在求解 MILP 问题中取得的成果进行了扩展, 为 MINLP 问题的研究提供了更好的向导.

GBD 迭代算法与 OA 算法用到的数学工具相同, 只有构造 MILP 的方式不同. 它的基本思想是分解, 首先在根节点求解 MINLP 的连续松弛问题 (2.4) 并初始化线性化点集 K_1 和 K_2. 它在每个迭代步都要求解 NLP 问题和 MILP 问题:

(1) 固定整数变量 $\hat{y}_k$ 得到 NLP 子问题 (2.5) 或 (2.6), 求解 NLP 问题得到连续变量 $\bar{x}_k$ 并试图更新上界.

(2) 用 $(\bar{x}_k, \hat{y}_k)$ 更新 K_1 或 K_2, 构造和求解 MILP 问题 (2.8), 其中 MILP 子问题的更新方式如下: 如果子问题 (2.5) 是可行的, 利用目标函数及有效约束函数线性化后的有效结合, 生成最优割添加到

MILP 子问题. 对于固定的 $\hat{y}$, 令 $(\bar{x},\hat{y})$ 是 (2.5) 的最优解, 则有最优 Lagrange 乘子 $\gamma\geqslant 0$ 使得以下广义 Benders 分解对于 MINLP 问题是成立的,

$$\eta\geqslant f(\bar{x},\hat{y})+(\triangledown_y f(\bar{x},\hat{y})+\triangledown_y g(\bar{x},\hat{y})\gamma)^{\mathrm{T}}(y-\hat{y});$$

而如果 (2.5) 不可行, 利用 (2.6) 提供的解 $(\bar{x},\hat{y})$ 及 Lagrange 乘子 $\beta\geqslant 0$ 得到相应的最优割

$$\beta^{\mathrm{T}}[g(\bar{x},\hat{y})+\triangledown_y g(\bar{x},\hat{y})(y-\hat{y})]\leqslant 0.$$

这样可得到 MILP 松弛子问题

$$\begin{aligned}
Z_{\mathrm{GBD}}=\min\ &\eta,\\
\text{s.t.}\ \ &\eta\geqslant f(\bar{x},\hat{y})+(\triangledown_y f(\bar{x},\hat{y})+\triangledown_y g(\bar{x},\hat{y})\gamma)^{\mathrm{T}}(y-\hat{y}),\quad (\bar{x},\hat{y})\in K_1,\\
&\beta^{\mathrm{T}}[g(\bar{x},\hat{y})+\triangledown_y g(\bar{x},\hat{y})(y-\hat{y})]\leqslant 0,\quad (\bar{x},\hat{y})\in K_2,\\
&y\in Y\cap\mathbb{Z}^p,\quad \eta\in\mathbb{R},
\end{aligned}\tag{2.8}$$

其中 K_1 和 K_2 分别表示 (2.5) 和 (2.6) 的最优解集.

(3) 若上下界的差值满足给定的容忍度, 则终止; 否则, 返步 (1).

利用 OA 算法求解 MILP 问题得到的下界比利用 GBD 算法得到的下界的精确性要强[29], 这导致 GBD 算法总体上需要的迭代次数比 OA 算法多. 但是, 注意到子问题 (2.8) 比 (2.7) 规模小, 所以有时候用 GBD 算法比 OA 算法更方便.

如今, Abhishek 等人[46] 建议 Benders 割平面添加到分支定界算法中, 为 Benders 割平面带来了新的发展空间. 如今人们已经利用 GBD 算法求解不定性过程供应链计划中的两阶段随机线性规划问题[47], 工程设计优化中的双层规划问题[48], 更具有挑战性的过程设计和组合优化的较大规模非凸 MINLP 问题[49] 等, 不断推进 GBD 算法的研究发展.

2.2.4 基于线性/非线性 - 分支定界算法

基于线性/非线性 - 分支定界算法 (LP/NLP based branch-and-bound, LP/NLP-BB) 于 1992 年由 Quesada 和 Grossmann [30] 提出, 并证明了该方法的收敛性, 故有时也称 QG 算法. LP/NLP-BB 最初目的是改善 0-1 MINLP 问题的求解效率. 第 1 步是初始化 MILP 子问题. 在根节点处求解 MINLP 松弛问题 (2.4) 得其解为 $(\hat{x},\bar{y})$, 并初始化线性化点集

$$C=\{(\hat{x},\bar{y})\},$$

在此节点处利用外逼近思想构造 MILP 子问题. 第 2 步是利用 BB 求解 MILP. 选择分支定界树的某一节点, 求解 LP 子问题. 若 LP 不可行, 则删除此节点并重新搜索分支定界树; 否则设 $(\check{x},\hat{y})$ 为 LP 的解. 若 $\hat{y}$ 为非整数, 则对当前节点进行分支处理; 否则, 若 $\hat{y}$ 为整数, 则求解 NLP 子问题 (2.5). 如果 (2.5) 是可行的, 设 $(\bar{x},\hat{y})$ 为其最优解. 当 (2.5) 的最优函数值小于当前上界时, 更新上界. 如果 (2.5) 不可行, 则令 $(\bar{x},\hat{y})$ 为 (2.6) 的最优解. 利用 (2.5) 或 (2.6) 提供的解 $(\bar{x},\hat{y})$ 更新线性化点集

$$C=C\cup\{(\bar{x},\hat{y})\},$$

并更新 MILP 子问题. 第 3 步是检查分支定界树是否为空. 若没有要访问的节点, 则算法终止; 否则重复进行第 2 步.

LP/NLP-BB 算法已经被应用到 BONMIN 和 FiMINT 等软件包中. 它可以看成 OA 算法的延伸, 与 OA 算法的区别在于 LP/NLP-BB 求解的是一系列 LP 子问题

$$
\begin{aligned}
Z_{\mathrm{LP/NLP}} = \min\ & \eta, \\
\text{s.t.}\ & f(\bar{x},\hat{y}) + \bigtriangledown f(\bar{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\bar{x} \\ y-\hat{y} \end{pmatrix} \leqslant \eta, \quad (\bar{x},\hat{y}) \in C, \\
& g(\bar{x},\hat{y}) + \bigtriangledown g(\bar{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\bar{x} \\ y-\hat{y} \end{pmatrix} \leqslant 0, \quad (\bar{x},\hat{y}) \in C, \\
& x \in X, \quad y \in \bar{Y}, \quad \eta \in \mathbb{R}, \quad \eta \in \mathbb{R},
\end{aligned} \tag{2.9}
$$

其中 $\bar{Y}$ 是在可行域 Y 中加入分支信息, 从而避免了求解大量 MILP 子问题 (2.7), 并且不需要在所有的节点处更新线性化点集 C. LP/NLP-BB 与 NLP-BB 的不同点在于, LP/NLP-BB 并非在每个节点处求解 NLP 子问题, 而是需要求解一系列 LP 子问题. 仅当 LP 子问题得到可行整数解时, 再求解 NLP 子问题, 从而节省计算量.

2.2.5 扩展割平面法

扩展割平面法 (extended cutting plane, ECP) 是 1995 年 Westerlund 和 Petterson [31] 对 1960 年 Kelly [50] 提出的只能用于求解凸 NLP 的割平面法的扩展, 用于求解凸 MINLP 问题并给出 ECP 算法的收敛性. ECP 的基本思想是割平面, 每个迭代步仅需要求解一个 MILP 子问题

$$
\begin{aligned}
Z_{\mathrm{ECP}} = \min\ & \eta, \\
\text{s.t.}\ & f(\bar{x},\hat{y}) + \bigtriangledown f(\bar{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\bar{x} \\ y-\hat{y} \end{pmatrix} \leqslant \eta, \quad (\bar{x},\hat{y}) \in D, \\
& g_j(\bar{x},\hat{y}) + \bigtriangledown g_j(\bar{x},\hat{y})^{\mathrm{T}} \begin{pmatrix} x-\bar{x} \\ y-\hat{y} \end{pmatrix} \leqslant 0, \quad (\bar{x},\hat{y}) \in D, \quad j \in \mathcal{J}(\bar{x},\hat{y}), \\
& x \in X, \quad y \in Y \cap \mathbb{Z}^p, \quad \eta \in \mathbb{R},
\end{aligned} \tag{2.10}
$$

其中 D 为 (2.10) 的迭代解集. 如果 MILP 的解 $(\bar{x},\hat{y})$ 满足所有的线性化, 则算法终止; 否则更新迭代解集 $D = D \cup \{(\bar{x},\hat{y})\}$ 和最大违反约束指标集

$$
\mathcal{J}(\bar{x},\hat{y}) \overset{\mathrm{def}}{=} \Big\{ j \in \arg\max_{j \in J} g_j(\bar{x},\hat{y}), J = \{1,\ldots,m\} \Big\}.
$$

有的学者也考虑将所有违反约束都添加到 $\mathcal{J}(\bar{x},\hat{y})$. (2.10) 的迭代最优解为 (1.1) 的最优解提供了一非降下界序列, 有限收敛定理[31] 证明其收敛到 (1.1) 的最优解, 但是一般需要很多次迭代才能实现.

常用的商业软件 α-ECP 执行的就是 ECP 算法. 此外, ECP 算法在理论和应用中也得到了扩展. 现在 ECP 算法已经可以求解伪凸 MINLP 问题[51]. 文献 [52] 将 SHP (生成支撑超平面) 的思想融入到 ECP 算法中, 由于 SHP 算法对可行域比 ECP 算法更小, 从而加快了算法的收敛速度. 文献 [53] 详细给出了 ECP 算法在光滑和非光滑问题中的应用, 并给出了收敛性证明. 文献 [54] 利用 αECP 求解了一类非可微 MINLP, 这类问题的目标函数及约束函数是 f^0 伪凸的, 并证明了算法可以收敛到 MINLP 的全局最优解. 可见人们在努力将 ECP 应用到更复杂的 MINLP 问题, 促使对 ECP 的研究不断深入.

2.2.6 混合算法

混合算法 (Hyb) 是 2008 年由 Bonami 等人[32] 提出的. 这个混合算法是将 Quesada 和 Grossmann 提出的 LP/NLP-BB 与 OA 算法结合到一起, 本质上是对 LP/NLP-BB 进行改进. 混合算法与

LP/NLP-BB 的不同点在于, (1) 每 l 步, 在分支定界树子节点处直接求解 NLP 子问题 (2.5) 或 (2.6) 用于构造 OA 切割; (2) 限定时间 t, 对分支定界树的根节点调用 OA 算法, 求解 MILP 松弛问题和 NLP 子问题, 进行局部搜索. 如果局部枚举时间 t 是无限制的, 则此算法归结为 OA 算法. 而若 $l=1$, 即在每个节点处求解 NLP 子问题 (2.5), 这个算法就又归结为 NLP-BB.

2010 年, Abhishek 等人[46] 提出的混合算法与 Bonami 等人提出的混合算法不同点在于, 前者增加了切割管理方法. 特别地, 仅当发现 MILP 的可行整数解时 (如初始的 LP/NLP-BB 算法), NLP 子问题才能得到解决. 但是也在其他节点处添加线性化切割. 若在其他节点没有添加线性化切割, 算法就归结为 LP/NLP-BB 算法, 而如果在每个节点都添加线性化切割, 则算法归结为 ECP.

2.3 凸混合整数非线性规划的启发式算法

启发式算法可以快速找到可行解, 因此受到越来越多的关注. 很多分支定界算法中, 也考虑用启发式算法找到可行解来更新上界. 本文只对最常用的两种启发式算法: 潜水启发式算法 (diving heuristics) 和可行性泵启发式算法 (feasibility pump) 进行详细介绍, 更多其他启发式算法可参见文献 [22, 55–59].

2.3.1 潜水启发式算法

潜水启发式算法[56] 的基本原理是通过固定整数变量, 探索一条从分支定界树的根节点到叶子节点的路径, 以期待得到可行解. 最基本的方法是, 固定 NLP 松弛问题的解中离整数值最近的一个或几个变量, 然后求解得到 NLP 子问题, 持续此过程直到得到可行的整数解或不可行的 NLP 松弛问题. 不同算法执行过程中, 选择固定变量的方式、固定变量的数量和回溯 (松弛固定的变量) 概率也不尽相同. 利用此算法求解 MILP 问题与求解 MINLP 问题的不同之处在于, 前者的子问题是 LP, 当更新变量的上下界时可以用热启动快速求解, 而后者很难热启动.

Bonami 和 Goncalves[60] 采用两种不同的方式将这个算法运用于 MINLP 问题, 第一种是直接方式, 在每一次迭代中, 通过固定许多变量和回溯 (如果固定变量不可行) 限制要求解的 NLP 的数量. 第二种方式是通过固定出现在非线性目标函数和非线性约束函数的所有变量, 将原问题转化为 MILP 问题, 并利用 MILP 求解软件找到可行解. 采用固定变量的启发式思想求解 MINLP 问题的类似方法可参见文献 [59].

2.3.2 可行性泵启发式算法

可行性泵启发式算法是另一个可快速找到可行解的启发式算法. 最初是 Fischetti 等人[61] 针对 MILP 问题提出的, 后由许多学者进行改进并用到凸 MINLP 问题.

我们仅给出到 MINLP 问题的简单扩展. 可行性泵启发式算法的基本原理是产生两个迭代序列: 满足连续松弛 NLP 子问题 (2.4) 的点列 $(\bar{x}^0,\bar{y}^0),\dots,(\bar{x},\bar{y}^k)$, 但这个点列不一定满足整数约束条件; 与它相关的是整数点列 $(\hat{x}^1,\hat{y}^1),\dots,(\hat{x}^{k+1},\hat{y}^{k+1})$, 但不一定满足 (2.4) 的约束条件. 特别地, 选取 $(\bar{x}^0,\bar{y}^0)$ 是 (2.4) 问题的最优解, 并令 $\hat{x}^{k+1}=\bar{x}^k$, $\hat{y}^{k+1}$ 是 $\bar{y}^k$ 最接近的整数. 序列 $(\bar{x}^k,\bar{y}^k)$ 是满足连续约束, 并且整数变量在连续空间中距离 $\hat{y}^k$ 最近的点, 即求解以下形式的 NLP:

$$
\begin{aligned}
Z_{\mathrm{NLP}(\hat{y}^k)} = \min\ & \|y-\hat{y}^k\|_1,\\
\text{s.t.}\ & g(x,y)\leqslant 0,\\
& x\in X,\quad l_I\leqslant y\leqslant u_I.
\end{aligned}
$$

这两个序列的性质是, 在每次迭代中, $(\bar{x}^k, \bar{y}^k)$ 与 $(\hat{x}^{k+1}, \hat{y}^{k+1})$ 的距离非增. 当找到可行解或 $(\bar{x}^k, \bar{y}^k) = (\hat{x}^k, \hat{y}^k)$ 时, 迭代终止.

上述基本形式可能出现循环或者找不到整数可行解, 文献 [61] 建议用随机策略进行初始化. 文献 [62–64] 提供了几个求解 MILP 的可行性泵启发式算法变体, 而文献 [65,66] 已证实了在较短的时间内利用可行性泵启发式算法可找到 MINLP 的较好解.

3 非凸混合整数非线性规划的算法

MINLP 通常可分为凸 MINLP 和非凸 MINLP 两大类. 如果我们对 MINLP 原问题的目标函数或约束函数没有作凸性假设, 则它就是一类更具有挑战性的非凸 MINLP 问题. 凸 MINLP 理论与算法已经日渐成熟, 但很多实际问题往往是非凸的. 为了成功解决非凸 MINLP 问题, 如今越来越多的研究者开始关注. 本节首先指明凸 MINLP 问题的确定型算法只能作为非凸 MINLP 的启发式算法; 其次重点介绍两种常用处理非凸性的方法及其一些改进算法的发展情形; 最后给出可以直接求解非凸 MINLP 问题的算法.

非凸 MINLP 较之凸 MINLP 有本质困难, 求解非凸 MINLP 的 NLP 子问题很难得到全局最优解. 目前, 很多学者提出了非凸 MINLP 的全局化方法[67], 包括多重开始 (MultiStart)、变邻域搜索 (variable neighbourhood search) 和空间分支定界 (spatial branch-and-bound). 但这非本文介绍的重点, 感兴趣的读者可以参考相关文献及书籍 (参见文献 [68–71]). 一般情形下, 如果 NLP 计算软件不能保证求得 NLP 子问题的全局最优解, 用分支定界类算法中与当前上界进行比较时, 就可能将可行解, 甚至是最优解剪枝. 因此, 像凸 MINLP 问题的 NLP-BB, LP/NLP-BB 和 Hyb 这些确定型算法, 只能作为求解非凸 MINLP 问题的启发式算法. 另外一般在非凸情形下, 如 OA, GBD, ECP 使用的线性化割, 对非线性约束是无效的. 因为在非凸情形下, 线性化割不仅会切掉不可行点, 还可能切割掉部分可行点, 如图 1.

由于以上原因, 如果要求解非凸问题, 则应该谨慎使用线性化切割或接受非最优解. 因此, 要有效求解非凸 MINLP, 对非凸函数进行高效处理就显得尤为重要. 通常有两种处理非凸性的方法:

第一种方法是完全重构[72,73], 实现精确重构最常用的方法是变量符号转换. 但只在有限情形下才

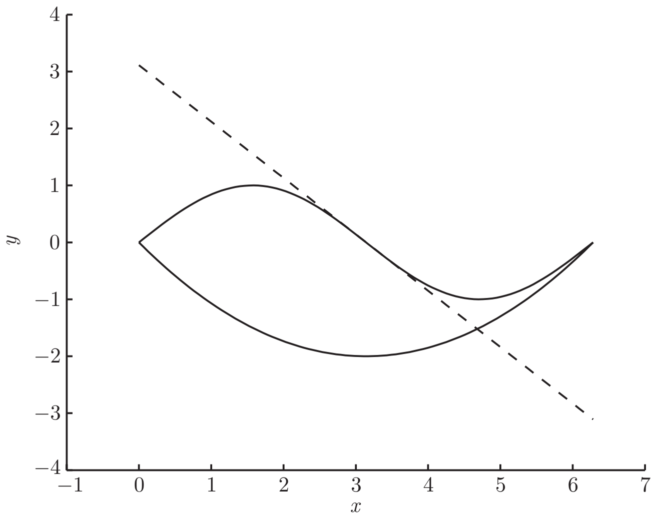

图 1 线性化割不适用于非凸可行域的情形

可以利用精确重构, 构造出与原问题完全等价的凸问题. 所谓等价, 是指满足以下描述的两个条件:

(1) 原问题所有可行点均能唯一映射到凸重构问题的可行域中;

(2) 没有不满足原问题约束条件的整数解包含在重构可行域内.

有关精确重构到标准形式的详细描述可参见文献 [39,73].

第二种方法是最普遍使用的方式, 基于非凸可行域的凸包或凸松弛[22,72]. 原问题 (1.1) 具有以下形式的松弛问题:

$$\begin{aligned} Z_{\mathrm{rMINLP}} = \min\ & \eta, \\ \text{s.t.}\ & \underline{f}(x,y) \leqslant \eta, \\ & \underline{g}(x,y) \leqslant 0, \\ & x \in X, \quad y \in Y \cap \mathbb{Z}^p, \quad \eta \in \mathbb{R}, \end{aligned} \tag{2.11}$$

其中 η 是线性目标函数的标准模型化辅助变量, $\underline{f}$: $\mathbb{R}^{n\times p} \to \mathbb{R}$ 和 $\underline{g}$: $\mathbb{R}^{n\times p} \to \mathbb{R}^m$ 是凸 (在某些情形下是线性) 函数, 在 (x,y) 的可行域内满足

$$\underline{f}(x,y) \leqslant f(x,y), \quad \underline{g}(x,y) \leqslant g(x,y).$$

大体来讲, 如果用重构方法处理复杂的非线性项, 首先应通过添加辅助变量和约束把复杂的非线性项分解为基本非线性项, 然后利用凸包/松弛对基本非线性项进行松弛处理. 用分段函数逼近原问题中的非线性函数[22,74] 是一个重要的选择. 但是, 对于不同结构的非线性函数, 一般采用不同的分段逼近方式, 如分段线性逼近一般适用于凹函数[75]. 虽然分段重构法可以扩展到非分段函数[76], 但通常还是用于分段函数 (可以表示为和的形式). 可分解函数 f 一般可以表示为

$$f = \sum_e \prod_q f_{eq}(x),$$

其中 $f_{eq}(x)$ 是单变量函数. 通过添加辅助变量及约束条件, 这些函数可以缩减重构为更简单的非线性函数, 如

$$l_{eq} = f_{eq}(x), \quad l'_e = \prod_q l_{eq}, \quad l''_q = \sum_e l'_e.$$

这种重构方法可以预先确定使用一些凸松弛已知的项, 如对于容易找到凸松弛的双线性项、三线性项或分数项[77,78].

使用凸松弛法会增大可行域: 如果 MINLP 松弛问题的最优解是原问题的可行解, 那么, 它也是原问题的全局最优解; 否则就需要精化处理, 通常采用的方法是分支. 不仅对整数变量进行分支, 还要对连续变量进行分支. 如空间分支定界算法[38,39]、分支下降算法[40,41]、αBB [79,80]、分支割平面算法[81], 利用这些算法都可以得到问题的全局最优解. 而它们的主要不同点在于选择的分支策略. 在这些算法中, 有的先对整数变量进行分支 (直到找到可行整数解), 有的先对连续变量进行分支, 或对整数变量和连续变量同时分支. 很显然这些算法一般情形下都很耗时, 并且只适用小型或中型优化问题. 但这是为得到全局最优解 (当然是在给定的容忍度内) 需要付出的代价.

从数值角度, 设计和执行全局化算法的详细步骤可参见文献 [70]. 对于具有特殊结构的非凸函数 f 和 g, 定义函数 $\underline{f}$ 和 $\underline{g}$ 的方式可参见文献 [39,78,82,83]. 需要说明的一点是, 凸 MINLP 问题的基本启发式算法也可以扩展到非凸 MINLP 问题, 只要精细化处理非凸带来的问题, 如利用稍微改进的 Feasibility Pump 可求解非凸 MINLP [84]. 事实上, 在合理时间内求得非凸 MINLP 的一个不必十

分精确但较为满意的解的启发式算法的研究成了很活跃的研究课题, 并且取得了一定成就 (参见文献 [84, 85]). 另外也可将非凸 MINLP 问题转化为为易处理的析取规划 (参见文献 [86–88]), 该方法近几年越来越受到人们的重视.

为了设计更精确更高效的算法, 越来越多的研究者开始将求解凸 MINLP 的一些确定型算法与一些智能算法或其他方法结合在一起, 并取得了一定的成功. 这些算法包括在一般 NLP-BB 算法框架内, 在根节点及某些叶子节点利用 OA 思想形成求解效率更高的混合算法[89]、遗传算法与罚函数结合[90]、模拟退火算法与确定型算法结合[91]. 虽然此类算法的研究还不成熟, 但是这使得对混合算法的研究上了新的台阶. 另外, 还有人提出了一些求解 MINLP 问题有效的改进算法, 包括析取割[92]、置信区间二次序列规划算法[93]、填充函数法[94] 等.

最后介绍直接求解非凸 MINLP 问题的全局优化算法. 可以直接求解非凸 MINLP 的全局优化算法, 包括空间分支定界算法[38,39]、分支下降算法[40,41]、αBB [79,80], 其中前两种算法比较传统. αBB 与它们相比最大的优点是不需要加入额外的辅助变量. 下面只简单介绍现在最为常用的算法 — αBB. 它是利用递归分割可行域的隐枚举全局优化算法. 最初用于求解连续二次可微的非凸 NLP 问题[95], 后来被设计为非凸 MINLP 问题的全局优化算法. αBB 与一般 BB 类似, 包含两个重要的过程: 计算子问题最优目标函数值, 分割子问题可行域. 从处理问题类型的角度来说它又可分为两类:

第一类算法是 αBB 算法的延伸, 适用于所有的整数变量都是二元变量且整数变量仅存在于线性或混合双线性项中的问题. 并且这类问题的非凸函数关于连续变量二次连续可微分. 在求解过程中对连续变量及二元变量均进行分支处理, 称为特殊结构混合整数非线性 αBB (SMIN-BB).

第二类算法保证整数变量以更一般的形式参与的更一般问题收敛到全局最优解, 但条件是它的松弛问题是二次连续的, 这类算法称为一般结构混合整数非线性 αBB (GMIN-BB).

更多直接求解非凸 MINLP 的算法可见综述性文献 [96].

4 混合整数非线性规划的计算软件

近几十年, 求解 MINLP 的计算软件也在不断地发展. 有些软件是针对凸 MINLP 开发的, 有些软件是针对非凸 MINLP 开发的 (当然它们之间有一定的交集). 本节主要介绍开源软件 BONMIN, COUENNE, LaGO, SCIP 与商业软件 αECP, BARON, DICOPT, KNITRO, 并简洁提供它们的基本信息. 可求解 MINLP 的其他开源软件有 bnb [97], MINOTAUR [98], MILANO [99] 等, 商业软件有 αBB [100], ANTIGONE [101], LindoAPI [102], MISQP [103], MINOPT [104], MIDACO [105], OQNLP [106], SBB [107] 等, 这里不再一一介绍.

4.1 开源软件

开源软件是一种源代码可以任意获取的计算机软件, 这种软件的版权持有人在软件协议的规定之下保留一部分权利并允许用户学习、修改、增进、提高这款软件的质量. 表 1 介绍了四种开源软件的基本信息. 其中, 可获界面就是计算软件所在界面, 包括 GAMS [108], AMPL [109], NEOS [110], AIMMS [111]. URL 是对可以从互联网上得到的资源的位置和访问方法的一种简洁表示, 是互联网上标准资源的地址.

BONMIN [32,112] (basic open-source nonlinear mixed integer programming) 是 Bonami 在 Carnegie Mellon 大学 Tepper 商学院时和 IBM 研究院合作, 与其合作者共同开发的, 2008 年发布了第一个版本.

表 1 四种开源软件基本信息

软件名	可获界面	URL
BONMIN	AMPL, GAMS, NEOS	https://projects.coin-or.org/Bonmin
COUENNE	AMPL, GAMS, NEOS	https://projects.coin-or.org/Couenne
LaGO	AMPL, GAMS	https://projects.coin-or.org/LaGO
SCIP	AMPL, GAMS, NEOS	http://scip. zib. de/

BONMIN 可以执行以下算法: B-BB 是 NLP-BB 算法, B-OA 基于 OA 算法, B-QG 是第 2.2.4 小节中介绍的 LP/NLP-BB, B-Hyb 是第 2.2.6 小节中介绍的的混合算法, 此外还有 B-Ecp, B-iFP 等, 详见用户手册 [112]. 以上算法都是求解凸 MINLP 的确定型算法, 非凸 MINLP 问题的启发式算法. 如果要处理非凸 MINLP, 则建议使用 B-BB, 该算法试图找到全局解, 有提高解的质量的选项. 最后, BONMIN 描述了基本启发式算法的特征, 这些启发式算法包括 Feasibility Pump, Diving Heuristic 和 RINS. 文献 [113] 提供了可用此软件求解的 MINLP 测试问题包. 在使用软件 BONMIN 求解 MINLP 的过程中可能要用到求解子问题 MILP 的软件 CBC [114] 或者 CPLEX [115], NLP 子问题的软件 IPOPT [116] 或者 Filter-SQP [117].

COUENNE [38, 118] (convex over and under envelopes for nonlinear estimation) 最初是 Belotti 等人在 Carnegie Mellon 大学 Tepper 商学院时和 IBM 合作研究中开发出的, 2008 年发布了第一个版本, 现在由 FICO 进行研究. COUENNE 基于重构的空间分支定界算法, 可以求解凸及非凸 MINLP 问题全局最优解. 与 BARON 类似, 对 MINLP 重构得到线性外部逼近. 它的主要组成部分是: (1) 在根节点处引入辅助变量对原问题重构; (2) 线性化松弛; (3) 分支规则; (4) 紧化变量上下界. 其执行的是利用 LP 进行定界的分支定界算法. 在使用软件 COUENNE 求解过程中可能要用到 CBC [114] 对 MILP 子问题进行求解, IPOPT [116] 求解 NLP 子问题.

LaGO [81] (Lagrangian global optimizer) 是由柏林 Humboldt 大学数学系的 Nowak 和 Vigerske 研究开发出来的软件包. 它能够求解凸 MINLP 和非凸 MINLP 问题. 算法几个关键步骤包括:

(1) 将原问题重构为块分离形式;

(2) 构造外逼近;

(3) 块既约算法;

(4) 分支和割平面策略.

其松弛问题是通过线性化凸函数、外逼近二次非凸函数和对非二次函数二次近似得到的. 由于 LaGO 没有原问题的代数表示, 所以不能利用重构法得到其凸松弛, 但在非二次非凸的情形下可以应用抽样法. 另外通常用割平面法改善得到的线性松弛, 利用缩小可行域法改善求解效率. LaGO 有处理一般黑箱子函数的优势, 但存在的不足之处是不能使用高级重构法和界限缩小法. 因此在使用抽样法时, 不能保证找到精确的全局最优解. 但是如今很遗憾的是, LaGO 自 2008 年来不再更新. 在使用软件 LaGO 求解 MINLP 的过程中可能要用到求解子问题 MILP 的软件 CPLEX [115], NLP 子问题的软件 IPOPT [116] 等. 值得注意的是, BONMIN, COUENNE 和 LaGo 都是 COIN-OR 的产品.

SCIP [12, 119] (solving constraint integer programs) 是 Zuse Institute Berin 优化部门及其合作伙伴开发出来的, 主要以源代码和独立二进制文件的形式获得. 它能够求得凸及非凸 MINLP 问题的全局最优解, 执行的是利用 LP 进行定界的空间分支定界算法. 与 BARON 相似, 从 MINLP 的重构中产生外部近似. 另外, SCIP 包括大邻域启发式算法及一个新的混合整数规划子问题启发式算法. 事实上,

SCIP 软件包本身包含求解 MILP 的工具, 但对于某些问题有时候可能还会用到求解线性规划强大的计算工具 CPLEX [115], 求解 NLP 子问题的软件 IPOPT [116] 等.

4.2 商业软件

商业软件是为了销售而生产, 或者服务于商业目的的计算机软件. 一般来讲, 它比开源软件更成熟, 但是使用者必须支付一定的费用. 本小节主要介绍四种商业软件的基本信息, 见表 2.

αECP [120] (alpha extended cutting plane) 是 Westerlund 及其研究团队在芬兰 Abo Akadem 大学的过程设计和系统工程试验室开发的, 它是执行第 2.2.5 小节的 ECP 算法为基础的局部 MINLP 求解软件. αECP 可以求得凸或者伪凸 MINLP 问题的全局最优解. 当 $\alpha=1$ 时, 此算法可以求解目标函数是线性函数, 约束函数是伪凸函数的 MINLP 问题[51]; 当 $\alpha\geqslant 1$ 时, 此算法可求解目标函数是线性函数, 约束函数是 f^o- 伪凸函数的 MINLP 问题[54]. 在使用软件 αECP 求解 MINLP 的过程中可能要用到求解子问题 MILP 的软件 CPLEX [115] 和 NLP 子问题的软件 CONOPT [121] 等.

BARON [41,122] (branch-and-reduce optimization navigator) 最初是 Illinois 大学 Urbana 分校化学工程系的 Sahinidis 及其团队开发的. 如今由 Carnegie Mellon 大学的 Sahinidis 和 Purdue 大学的 Tawarmalani 在研究. 它执行利用 LP 定界的空间分支定界算法, 可求解凸及非凸 MINLP 问题. 对于非凸 MINLP 问题, 其处理方法是添加辅助变量重构初始问题, 进而求得全局最优解. 它与 BB 算法的结合改善了此软件的性能, 成为当前最有效的全局求解软件之一. 它包含多个界限缩紧技术, 如 probing 和 violation transfer [22]. 从 BARON URL 给出了 1740 个 NLP/MINLP 问题测试的结果, 最新版的商业软件 BARON 14.4.0 均取得最好的计算结果. 在使用软件 BARON 求解 MINLP 的过程中可能要用到求解子问题 MILP 的软件 CPLEX [115] 等, NLP 子问题的软件 IPOPT [116], CONOPT [121] 等.

DICOPT [123] (discrete and continuous optimizer) 是由 Carnegie Mellon 大学工程研究设计中心的 Viswanathan 和 Grossmann 及其研究团队开发的. 它基于 OA 算法. 在求解非凸问题时, 依赖于等式松弛方法和增广罚函数, 即将等式松弛为不等式, 并在目标函数中加入罚因子, 但不保证求得非凸问题的全局最优解. DICOPT 适合于求解非线性函数仅涉及连续变量并且二元变量及整数变量仅存在于线性函数内的 MINLP 问题. 在使用软件 DICOPT 求解 MINLP 的过程中可能要用到求解子问题 MILP 的软件 CPLEX [115] 或者 XPRESS, NLP 子问题的软件 CONOPT [121], MINOS [124] 或者 SNOPT.

KNITRO [22,125] 是由 Ziena 优化部门研制开发的. 它最初为求解 NLP 的算法包, 自 2012 年以来加入了 MINLP 部分. 它以独立二进制文件及许多模型系统的组成部分获得. KNITRO 可以求解约束条件高达数十万的大规模 NLP 问题, 能够得到凸 MINLP 问题的全局最优解, 但要求问题的目标函数及约束函数都是光滑的. 它执行求解 NLP 最新的内点算法和积极集算法, 也可执行求解 MINLP 的利用 LP 松弛进行定界[30] 或 NLP 松弛进行定界[89] 的分支定界算法.

算法的好坏及软件的适用与否通常是由数值试验结果证明的, 而测试问题在数值试验中起着举足

表 2 商业软件基本信息

软件名	可获界面	URL
αECP	GAMS, NEOS	http://www.abo.fi/∼twesterl
BARON	AIMMS, GAMS, NEOS	http://archimedes.cheme.cmu.edu/?q=baron
DICOPT	GAMS, NEOS	http://www.gams.com/dd/docs/solvers/dicopt/index.html
KNITRO	AIMMS, AMPL, GAMS	http://www.ziena.com

轻重的作用. 文献 [108–110,113] 提供了一些 MINLP 测试问题包, 有利于研究者们开发和研究新的算法和软件. 为了证实求解 MINLP 问题优化算法的有效性, 我们可以借鉴 Spielman 和 Teng 在文献 [126] 中提出的平滑分析方法. 平滑分析法对测试集合进行扰动, 然后对其求期望并提出光滑复杂性, 最后利用概率思想对算法的光滑复杂性进行分析, 为证实算法的有效性提供新的框架.

5 总结与展望

本文主要介绍了求解 MINLP 的算法和软件及最新进展, 包括求解凸 MINLP 的六种确定型算法和两种启发式算法, 非凸 MINLP 的确定型算法和启发式算法, 以及求解 MINLP 的开源软件和商业软件. 在介绍凸 MINLP 确定型算法的过程中, 清楚呈现了 NLP 子问题与 MILP 主问题之间的关系, 有利于我们对这些算法框架的理解与掌握. 对于更具有挑战性的非凸 MINLP, 我们需要清楚地了解如何处理问题的非凸性, 设计有效的算法在合理的时间内找到高质量的可行解. 另外对 MINLP 的研究综述可参见文献 [22,96]. 总体来看, 求解一般 MINLP 问题的难点在于:

(1) 离散变量与连续变量共存;

(2) 非线性;

(3) 很难判断其解最优性 (事实上, MINLP 最优解的必要条件刻画也是很难的, 详见文献 [127]).

MINLP 研究的总体发展趋势是朝实际应用、大规模算法以及计算机科学交叉研究领域前进. 现有的算法还不能较好解决生产设备的使用计划安排、机组组合优化、通信和运输网络的拓扑优化等中的大规模 MINLP 应用问题, 解决这些应用领域中出现的 MINLP 问题具有重要的现实意义. MINLP 未来的研究方向和关键问题包括:

(1) 由于不同类型的 MINLP 问题具有自身不同的特点, 如 MIQP、MIQCP、MISDP 和 MISOCP 等. 需要根据其对应的子问题的特点设计算法, 如求解 QP 的积极集法和内点法, 求解 QCP 的罚函数、序列二次规划和半正定松弛算法, 求解 SDP 和 SOCP 的 Newton- 共轭梯度增广 Lagrangian 方法和内点法等. 如何使用或者修正现有的算法来快速求解对应的混合整数问题, 仍需要进一步理论与算法的研究.

(2) 由于 MILP 和 NLP 的研究更加成熟, 并且近几年涌现出一些新的算法, 如求解 MILP 的分支 - 割平面算法, 热启动和启发式算法, 求解 NLP 的交替方向乘子算法, 全局优化的凸逼近和模拟仿真技术等, 将为 MINLP 算法提供新的机遇.

(3) 随着并行环境的发展越来越受重视, 许多的 MIP 求解软件 (如 CPLEX、Gurobi、Xpress 和 SCIP 等) 都有自己的并行版本. 因此, 针对不同的问题, 设计高效的并行算法, 并开发出更加优秀的求解软件也是一个研究热点.

(4) 非凸 MINLP 问题的关键点是目标函数或约束函数的非凸性, 这类问题的全局优化算法受到广泛关注, 如符号重构方法、析取割平面、启发式算法、空间分支定界算法、紧化凸逼近等.

(5) 多目标 MINLP 问题的研究: 这类问题是指研究同时最优化多个目标的 MINLP 问题. 多目标优化问题的多个目标函数是相互联系, 相互冲突的, 因此通常不存在最优化每个目标函数的最优解, 再加上 MINLP 问题本身是 NP- 难的, 故这类问题会给研究者们带来更大的挑战. 目前存在的多目标优化算法大多是针对连续多目标优化的, 因此或许可以将求解连续多目标优化的算法 (如加权法和滤子法等) 通过适当改进, 扩展到 MINLP 问题.

(6) MINLP 的鲁棒优化: 这类问题的主要特点是参数具有不确定性. 此类问题在交通管理等多阶段特别是两阶段决策时经常遇到, 与随机 MINLP 问题的不同点在于, 鲁棒优化并不假设参数是随机

变量, 而通常是由某个确定性的集合来描述的. 为了尽量减少不确定性, 鲁棒优化寻求所有参数集合或误差范围内的可行解, 这就增加了求解 MINLP 的难度, 目前这一方向的研究还未取得明显的成就, 因此值得进一步探索研究.

(7) 将求解 NLP 具有快速收敛性的二次序列规划算法与确定型算法 (如分支定界算法) 以不同方式 (或者将二次序列规划算法融入到确定型算法内部, 或者利用信赖域将二次序列规划算法与确定型算法结合在一起) 结合是研究 MINLP 问题的一个重要课题. 在算法设计过程中, 可以选择改变限制使用二次序列规划算法的时间来改善整个算法的性能.

致谢 感谢两位匿名的审稿人提出的宝贵意见和建议, 使本文得到了很大的改进.

参考文献

1 Faria D C, Bagajewicz M J. A new approach for global optimization of a class of MINLP problems with applications to water management and pooling problems. AIChE J, 2012, 58: 2320–2335
2 Allgor R I, Barton P I. Mixed integer dynamic optimization. Comput Chem Eng, 1997, 21: 567–568
3 Luyben M L, Floudas C A. Analyzing the interaction of design and control, part 2: Reactor-separator-recycle systems. Comput Chem Eng, 1994, 18: 971–994
4 Adjiman C S, Androulakis I P, Floudas C A. Global optimization of MINLP problems in process synthesis and design. Comput Chem Eng, 1997, 121: 445–450
5 Shen Z H, Dong Y, Wang X X. Logistics base location optimization model based on mixed integer non-linear program. Logist Sci-Tech, 2013, 7: 89–93
6 Han D, Jian J, Yang L. Outer approximation and outer-inner approximation approaches for unit commitment problem. IEEE Trans Power Syst, 2014, 29: 505–513
7 Floudas C A. Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications. Oxford: Oxford University Press, 1995
8 Benichou M, Gauthier J M, Girodet P, et al. Experiments in mixed-integer linear programming. Math Program, 1971, 1: 76–94
9 Schrijver A. Theory of Linear and Integer Programming. New York: Wiley, 1986
10 Neumaier A, Shcherbina O. Safe bounds in linear and mixed-integer linear programming. Math Program, 2004, 99: 283–296
11 Linderoth J T, Ralphs T K. Noncommercial software for mixed-integer linear programming. In: Karlof J K, eds. Integer Programming: Theory and Practice. Boca Raton: CRC Press, 2005, 253–303
12 Achterberg T. Constraint integer programming. PhD Thesis. Berlin: Institute of Technology, 2007
13 Nemhauser G L, Wolsey L A. A recursive procedure to generate all cuts for 0-1 mixed integer programs. Math Program, 1990, 46: 379–390
14 Balas E, Ceria S, Cornuéjols G. A lift-and-project cutting plane algorithm for mixed 0-1 programs. Math Program, 1993, 58: 295–324
15 Stubbs R A, Mehrotra S. A branch-and-cut method for 0-1 mixed convex programming. Math Program, 1999, 86: 515–532
16 Sun X L, Li D. The recent developments of integer programming. J Oper Res, 2013, 18: 39–68
17 Li D, Sun X L. Nonlinear Integer Programming. New York: Springer, 2006
18 Garey M R, Johnson D. Computer and Intractability: A Guide to the Theory of NP-Completeness. New York: Freeman, 1979
19 Nemhauser G L, Wolsey L A. Integer and Combinatorial Optimization. New York: Wiley, 1988
20 Lee J, Leyffer S. Mixed Integer Nonlinear Programming. New York: Springer, 2012
21 Wolsey L A. Integer Programming. New York: Wiley, 1998
22 Belotti P, Kirches C, Leyffer S, et al. Mixed-integer nonlinear optimization. Acta Numer, 2013, 22: 1–131
23 Bussieck M R, Vigerske S. MINLP Solver Software. New York: Wiley, 2010
24 Bonami P, Kilinc M, Linderoth J. Algorithms and Software for Convex Mixed-Integer Nonlinear Programs. In: Mixed Integer Nonlinear Programming, vol. 154. New York: Springer, 2012, 1–39
25 D'Ambrosio C, Lodi A. Mixed integer nonlinear programming tools: An updated practical overview. Ann Oper Res,

2013, 204: 301–320
26 Savelsbergh M W P. Preprocessing and probing techniques for mixed integer programming problems. ORSA J Comput, 1994, 6: 445–454
27 Dakin R. A tree-search algorithm for mixed integer programming problems. Comput J, 1965, 8: 250–255
28 Geoffrion A. Generalized Benders decomposition. J Optim Theory Appl, 1972, 10: 237–260
29 Duran M, Grossmann I. An outer approximation algorithm for a class of mixed integer nonlinear problems. Math Program, 1986, 36: 307–339
30 Quesada I, Grossmann I E. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. Comput Chem Eng, 1992, 16: 937–947
31 Westerlund T, Pettersson F. A cutting plane method for solving convex MINLP problems. Comput Chem Eng, 1995, 19: 131–136
32 Bonami P, Biegler L, Conn A, et al. An algorithm framework for convex mixed integer nonlinear programs. Discrete Optim, 2008, 5: 186–204
33 Land A, Doig A. An automatic method of solving discrete programming problems. Econometrica, 1960, 28: 497–520
34 Gupta O, Ravindran V. Branch and bound experiments in convex nonlinear integer programming. Manag Sci, 1985, 31: 1533–1546
35 Linderoth J T, Savelsbergh M W P. A computational study of search strategies for mixed integer programming. INFORMS J Comput, 1999, 11: 173–187
36 Achterberg T, Koch T, Martin A. Branching rules revisited. Oper Res Lett, 2005, 33: 42–54
37 Bonami P, Lee J, Leyffer S, et al. More branch-and-bound experiments in convex nonlinear integer programming. Http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.416.7203&rep=rep1&type=pdf, 2011
38 Belotti P, Lee J, Liberti L, et al. Branching and bounds tightening techniques for non-convex MINLP. Optim Methods Softw, 2009, 24: 597–634
39 Liberti L. Reformulation and convex relaxation techniques for global optimization. PhD Thesis. London: Imperial College, 2004
40 Tawarmalani M, Sahinidis N. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Math Program, 2004, 99: 563–591
41 Sahinidis N V. BARON: A general purpose global optimization software package. J Global Optim, 1996, 8: 201–205
42 Barnhart C, Johnson E L, Nemhauser G L, et al. Branch-and-price: Column generation for solving huge integer programs. Oper Res, 1998, 46: 316–329
43 Fletcher R, Leyffer S. Solving mixed integer nonlinear programs by outer approximation. Math Program, 1994, 66: 327–349
44 Shahabi M, Unnikrishnan A, Boyles S D. An outer approximation algorithm for the robust shortest path problem. Trans Res Part E: Logist Trans Rev, 2013, 58: 52–66
45 Benders J F. Partitioning procedures for solving mixed-variables programming problems. Numer Math, 1962, 4: 238–252
46 Abhishek K, Leyffer S, Linderoth J T. FilMINT: An outer-approximation-based solver for nonlinear mixed integer programs. INFORMS J Comput, 2010, 22: 555–567
47 You F, Grossmann I E. Multicut Benders decomposition algorithm for process supply chain planning under uncertainty. Ann Oper Res, 2013, 210: 191–211
48 Mínguez R, Conejo A J, Castillo E. Optimal engineering design via Benders decomposition. Ann Oper Res, 2013, 210: 273–293
49 Li X, Chen Y, Barton P I. Nonconvex generalized Benders decomposition with piecewise convex relaxation for global optimization of integrated process design and operation problems. Ind Eng Chem Res, 2012, 51: 7287–7299
50 Kelly J E. The cutting plane method for solving convex programs. J Soc Ind Appl Math, 1960, 8: 703–712
51 Westerlund T, Porn R. Solving pseudo convex mixed integer problems by cutting plane techniques. Optim Eng, 2002, 3: 253–280
52 Da L, Cha J Z. A kind of supporting hyper plane method for solving mixed integer nonlinear programming. Syst Eng Theory Pract, 2008, 28: 82–86
53 Eronen V P. On extended cutting plane and αECP algorithms. Https://www.utu.fi/en/units/sci/units/math/Research/optimization/Documents/sovmat07062013.pdf, 2013
54 Eronen V P, Mäkelä M M, Westerlund T. Extended cutting plane method for a class of nonsmooth nonconvex MINLP problems. Optim, 2013, 63: 1–21
55 Danna E, Rothberg E, Lepape C. Exploring relaxation induced neighborhoods to improve MIP solutions. Math

Program, 2005, 102: 71–90
56 Berthold T. Heuristic algorithms in global MINLP solvers. PhD Thesis. Berlin: Techniche Universität Berlin, 2014
57 Liberti L, Maldenovi N, Nannicini G. A recipe for finding good solutions to MINLPs. Math Program Comput, 2011, 4: 349–390
58 Berthold T. Primal MINLP heuristics in a nutshell. In: Operations Research Proceedings 2013. New York: Springer, 2014, 23–28
59 Berthold T, Gleixner A. Undercover: A Primal Heuristic for MINLP Based on Sub-MIPs Generated by Set Covering. Technical Report ZIB-Report, 2012
60 Bonami P, Goncalves J. Heuristics for convex mixed integer nonlinear programs. Comput Optim Appl, 2012, 51: 729–747
61 Fischetti M, Glover F, Lodi A. The feasibility pump. Math Program, 2005, 104: 91–104
62 Achterberg T, Berthod T. Improving the Feasibility Pump. Technical Report ZIB-Report, 2005
63 Bertacco L, Fischetti M, Lodi A. A feasibility pump heuristic for general mixed-integer problems. Discrete Optim, 2007, 4: 63–76
64 Fischette M, Salvagnin D. Feasibility pump 2.0. Math Program Comput, 2009, 1: 201–222
65 Berthold L, Gleixner A. Primal Heuristics for General Mixed Integer Nonlinear Programs. Research Report IBM, 2008
66 Bonami P, Cornuéjols G, Lodi A, et al. A feasibility pump for mixed integer nonlinear programs. Math Program, 2009, 119: 331–352
67 Liberti L. Writing global optimization software. In: Global Optimization. New York: Springer, 2006, 211–262
68 Floudas C A, Gounaris C E. A review of recent advances in global optimization. J Global Optim, 2009, 45: 3–38
69 Fampa M, Lee J, Melo W. On Global Optimization with Indefinite Quadratics. Technical Report, Issac Newton Institute Preprint NI13066, 2013
70 Liberti L, Maculan N. Global Optimization: From Theory to Implementation. New York: Springer, 2006
71 Grossmann I E, Ruiz J P. Generalized disjunctive programming: A framework for formulation and alternative algorithms for MINLP optimization. In: Mixed Integer Nonlinear Programming. New York: Springer, 2012, 93–115
72 Smith E, Pantelides C. A symbolic reformulation/spatial branch and bound for the global optimization of nonconvex MINLPs. Comput Chem Eng, 1999, 23: 457–478
73 Liberti L, Cafieri S, Tarissan F. Reformulations in mathematical programming: A computational approach. In: Foundations of Computational Intelligence, vol. 3. New York: Springer, 2009, 153–234
74 D'Ambrosio C, Lodi A, Martello S. Piecewise linear approximation of functions of two nonconvex variables in MILP. Oper Res Lett, 2012, 136: 39–46
75 Bergamini M L, Grossmann I E, Scenna N, et al. An improved piecewise outer-approximation algorithm for the global optimization of MINLP methods involving concave and bilinear terms. Comput Chem Eng, 2008, 32: 477–493
76 Beale E M L, Forrest J J H. Global optimization as an extension of integer programming. In: Towards Global Optimization 2. Amsterdam: North-Holland, 1978, 131–149
77 Oktay G, Jeff L. Perspective reformulation of mixed nonlinear programs with indicators variables. Math Program, 2010, 124: 183–205
78 McCormick G. Computability of global solutions to factorable nonconvex programs: Part I — convex underestimating problems. Math Program, 1976, 10: 147–175
79 Adjiman C S, Androulakis I P, Floudas C A. Global optimization of MINLP problems. AIChE J, 2000, 46: 1769–1797
80 Adjiman C S, Androulakis I P, Floudas C A. Global optimization of MINLP problems in process synthesis and design. Comput Chem Eng, 1997, 21: 445–450
81 Nowak I, Vigerske S. LaGO: A (heuristic) branch and cut algorithm for nonconvex MINLPs. Cent Eur J Oper Res 4, 2008, 16: 127–138
82 Maranas C D. Optimization molecular design under property prediction uncertainty. AIChE J, 2008, 43: 1250–1264
83 Smith E M B, Pantelides C C. Global optimisation of nonconvex MINLPs. Comput Chem Eng, 1997, 21: 791–796
84 D'Ambrosio C, Frangioni A, Liberti L, et al. A storm of feasibility pumps for non-convex MINLP. Math Program, 2012, 136: 375–402
85 Nannicini G, Belotti P. Rounding based heuristics for nonconvex MINLP. Math Program Comput, 2012, 4: 1–31
86 Lee S, Grossmann I E. New algorithms for nonlinear generalized disjunctive programming problems. Comput Chem Eng, 2000, 24: 2125–2141
87 Raman R, Grossmann I E. Modelling and computations for logic base integer programming. Comput Chem Eng, 1994, 18: 563–578

88 Grossmann I E, Lee S. Generalized convex disjunctive and programming: Nonlinear convex hull relaxation. Comput Optim Appl, 2003, 26: 83–100

89 Wendel M, Marcia F, Fernanda R. Integrating Nonlinear Branch-and-Bound and Outer Approximation for Convex Mixed Integer Nonlinear Programming. New York: Springer, 2014

90 Li Y X, Mitsuo G. Nonlinear mixed integer programming problems using genetic algorithm and penalty function. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 4. Beijing: IEEE, 1996, 2677–2682

91 Tamaki K, Tengan T, Nakamura M. Hybrid approaches based on simulated annealing and an exact algorithm for mixed integer programming problems. In: The Third International Conference on Networking and Computing. Okinawa: IEEE, 2012, 435–440

92 Bonami P, Linderoth J T, Lodi A. Disjunctive cuts for mixed integer nonlinear programming problems. Prog Combin Optim, 2011, 18: 521–541

93 Exler O, Klaus S. A trust region SQP algorithm for mixed-integer nonlinear programming. Optim Lett, 2007, 1: 269–280

94 Shang Y L, Xu C X. Resolving communication control and management problems with modified filled function method. In: Proceedings of the 2008 ISECS International Colloquium on CCCM, 2. Guangzhou: IEEE, 2008, 612–616

95 Adjiman C S, Androulakis I P, Floudas C A. A global optimization method, αBB, for general twice-differentiable constrained NLPs. Imple Comput Results, 1998, 22: 1137–1158

96 Burer S, Letchford A N. Non-convex mixed integer nonlinear programming: A survey. Surv Oper Res Management Sci, 2012, 17: 97–106

97 Http://www.mathworks.com/matlabcentral/fileexchange/95

98 Http://wiki.mcs.anl.gov/minotaur/index.php/Main_Page

99 Http://www.pages.drexel.edu/˜hvb22/milano/

100 Androulakis I P, Maranas C D, Floudas C A. αBB: A global optimization method for general constrained nonconvex problems. J Global Optim, 1995, 7: 337–363

101 Http://helios.princeton.edu/ANTIGONE/

102 Lin Y, Schrage L. The global solver in the LINDOAPI. Optim Methods Softw, 2009, 24: 657–668

103 Exler O, Lehmann T, Schittkowski K. A comparative study of SQP-type algorithms for nonlinear and nonconvex mixed-integer optimization. Math Program Comput, 2012, 4: 383–412

104 Schweiger C A, Floudas C A. MINOPT: A modeling language and algorithmic framework for linear, mixed-integer, nonlinear, dynamic, and mixed-integer nonlinear optimization. PhD Thesis. Princeton: Princeton University, 1998

105 Schlüter M, Gerdts M, Rückmann J J. A numerical study of MIDACO on 100 MINLP benchmarks. Optim, 2012, 61: 873–900

106 Ugray Z, Lasdon L, Plummer J, et al. Scatter search and local NLP solvers: A multistart framework for global optimization. Inform J Comput, 2007, 19: 328–340

107 Http://www.gams.com/dd/docs/solvers/allsolvers.html

108 Floudas C A, et al. Http://www.gamsworld.org/minlp

109 Https://wiki.mcs.anl.gov/leyffer/images/e/e6/TACO.pdf

110 Http://www.neos-server.org/neos/

111 Bisschop J. AIMMS-optimization modeling. Http://main.aimms.com/downloads/manuals/optimization-modeling, 2006

112 Bonami P, Lee J. Bonmin users' manual. Numer Math, 2007, 4: 1–32

113 CMU-IBM open source MINLP project test set. Http://egon.cheme.cmu.edu/ibm/page.htm

114 Https://projects.coin-or.org/Cbc

115 Http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

116 Wächter A, Biegler L T. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math Program, 2006, 106: 25–57

117 Fletcher R, Leyffer S. User Manual for Filter-SQP. Numerical Analysis Report NA-181. University of Dundee, 1998

118 Belotti P. COUENNE: A user's manual. Http://www.coin-or.org/Couenne/couenne-user-manual.pdf, 2009

119 Achterberg T. SCIP: Solving constraint integer programs. Math Program Comput, 2009, 1: 1–41

120 Westerlund T, Lundqvist K. Alpha-ECP, version 5.01: An interactive MINLP-solver based on the extended cutting plane method. Technical Report. Ǎbo Akademi University, 2001

121 Drud A S. CONOPT — A large-scale GRG code. ORSA J Comput, 1994, 6: 207–216

122 Tawarmalani M, Sahinidis N V. A polyhedral branch-and-cut approach to global optimization. Math Program, 2005,

103: 225–249
123 Kocis G R, Grossmann I E. Computational experience with DICOPT: Solving MINLP in process systems engineering. Comput Chem Eng, 1989, 13: 307–315
124 Murtagh B A, Saunders M A. MINOS 5.51 User's Guide. Technical Report SOL 83-20R. Stanford University, 1983
125 KNITRO documentation: Release 9.1. Ziena Optimization LLC. Http://www.ziena.com/docs/KNITRO91_User-Manual.pdf, 2014
126 Spielman A, Teng H S. Smoothed Analysis: An attempt to explain the behavior of algorithms in practice. Commun ACM, 2009, 52: 76–84
127 Murty K G, Kabadi S N. Some NP-complete problems in quadratic and nonlinear programming. Math program, 1987, 39: 117–129

Algorithms, softwares and recent developments of mixed integer nonlinear programming

LIU MingMing, CUI ChunFeng, TONG XiaoJiao & DAI YuHong

Abstract Mixed integer nonlinear programming (MINLP) has entered into each domain of real life and its research has important practical significance. To solve different types of MINLP problems effectively, researchers have proposed algorithms and effective softwares. This paper is devoted to the basic algorithms proposed to solve the MINLP and the corresponding softwares, and introduces the progress in the research of MINLP problems.

Keywords mixed integer nonlinear programming, branch-and-bound, cutting plane, software

MSC(2010) 01-02, 90C11, 97N80

doi: 10.1360/N012014-00278