

协同优化方法与并行子空间方法的评估与比较

王晓青

北京宇航系统工程研究所, 北京 100076

E-mail: sttg@163.com

收稿日期: 2008-08-23; 接受日期: 2008-10-09

摘要 多学科优化设计技术是航空航天领域内近年来发展起来的一种新的总体优化设计技术, 致力于解决复杂系统的耦合设计问题, 以获得系统的整体最优. 协同优化方法和并行子空间方法是多学科优化设计技术中的 2 种主要方法, 本文从计算效率、计算精度、学科自治、工程实用性等方面, 结合 3 个实例, 对 2 种方法进行深入的分析与评估. 分析结果表明: 并行子空间方法在计算效率、学科自治以及工程实用性方面比协同优化方法略胜一筹.

关键词

多学科优化设计
协同优化方法
并行子空间方法

1 引言

航天运载器是由多个学科组成的复杂大系统. 当前航天领域激烈竞争使得对运载器的设计越来越高, 各个学科间的相互影响变得更加紧密和重要, 单个学科的优化设计已越来越不能满足总体设计的要求, 多个学科或全系统的一体化优化设计的需求越来越迫切. 但由于飞行器系统比较复杂, 不同的分系统之间相互耦合, 真正实现飞行器全系统一体化优化设计需要考虑飞行器的所有分系统的设计和不同分系统之间的耦合关系, 极大地增加飞行器总体优化设计模型的复杂程度, 增加了优化设计的计算量. 为了有效提高设计效率, 节省各个学科精确分析时的计算开销, 寻求系统的最优方案, 各种多学科优化方法逐步发展起来.

多学科优化方法(multidiscipline design optimization, MDO)主要是针对多学科优化设计过程的复杂性特点所采取的一系列方法和数学手段^[1~4]. MDO方法与传统意义上的寻优算法不同. 传统寻优算法属于优化理论的研究领域, 而MDO方法是从设计问题本身入手, 从设计计算结构、信息组织的角度来研究问题, 是在具体寻优算法的基础上提出一套设计计

算框架, 该计算框架将设计对象各学科的知识与这些具体的寻优算法结合起来, 形成一套有效的解决复杂对象的优化求解方法. 按照优化层次上的分解方式可将多学科优化设计方法分为 2 种: 单级优化方法和两级优化方法.

单级优化方法^[2]将各学科的所有设计变量和约束都集成到系统级进行优化, 计算效率不高, 随着问题规模的扩大, 计算量就会超线性的增加, 不利于学科自治, 不适用于工程实用阶段. 近年来, 两级多学科优化设计方法逐步成为MDO方法的研究热点.

两级优化方法又称分解方法, 是后期发展起来的 MDO 方法. 它将系统的优化设计问题分为两级: 一个系统级和并行的多个学科级, 目前有代表性的几种两级优化方法包括: 并行子空间方法(concurrent subspace optimization, CSSO)、协同优化方法(collaborative optimization, CO)和 BLISS 方法(bi-level integrated system synthesis, BLISS). 其中, CSSO 方法与 CO 方法以其本身所具备的优点成为当前研究的热点, 但是, 国内研究重点在于这 2 种方法本身的特点以及对这 2 种方法的改进, 对于 2 种方法之间的比较以及 2 种方法处理具体问题时的计算效率以及工程实用性方面研究较少, 本文结合具体实例, 对 2 种方法

进行深入的比较与评估。

2 方法描述

2.1 并行子空间方法

1988年, Sobieski^[5]提出CSSO方法, CSSO采用学科精确分析模型与系统级采用学科近似模型的方式, 将多学科优化设计问题进行分解. 典型的多学科优化设计问题可以用下式描述

$$\begin{aligned} \min \quad & f(z), \\ \text{s.t.} \quad & \begin{cases} g(z) \leq 0, \\ h(z) = 0, \\ S_i(y_i; x, x_i, y_{ki}) = 0, \\ z_L \leq z \leq z_U, \quad z = \{x_{loc}, x_{sh}\}, \end{cases} \end{aligned} \quad (1)$$

式中, z 为设计变量, 包括全局设计变量 x_{sh} 和局部设计变量 x_{loc} ; $g(z)$ 为学科约束; $h(z)$ 为等式约束; $S_i(y_i; x, x_i, y_{ki})$ 为学科分析, 学科分析就是求解学科状态方程的过程, 即: $y_i = CA_i(x, x_i, y_{ki}) = S_i^{-1}(0; x, x_i, y_{ki})$, 其中 y_{ki} 表示其他学科 k ($1 \leq k \leq n, n$ 为学科总数) 到学科 i ($1 \leq i \leq n, n$ 为学科总数) 的耦合状态变量, 且 $k \neq i$, “;” 表示只有 y_i 是未知量, 表示学科状态变量; z_L, z_U 为设计变量上下限。

采用CSSO方法可以将(1)式的多学科优化设计问题分解为学科级优化问题和系统级优化问题, 学科级优化模型如(2)式, 系统级的优化模型如(3)式。

学科级优化模型.

$$\begin{aligned} \min \quad & f(z_i, z_0, x_i, x_0, y_i(z_i, z_0, x_i, x_0, \tilde{y}_j), \tilde{y}_j), \\ \text{s.t.} \quad & g(z_i, z_0, x_i, x_0, y_i(z_i, z_0, x_i, x_0, \tilde{y}_j), \tilde{y}_j) \leq 0, \end{aligned} \quad (2)$$

式中, f 代表目标函数; g 代表学科级约束; z_i, x_i 分别代表学科“ i ”的全局变量和局部变量; \tilde{y}_j 表示非局部耦合状态变量的响应面模型. z_0, x_0 分别表示在优化过程中保持为常数的变量。

系统级优化模型.

$$\begin{aligned} \min \quad & f(z, x, \tilde{y}), \\ \text{s.t.} \quad & g(z, x, \tilde{y}) \leq 0, \end{aligned} \quad (3)$$

式中, f 代表目标函数; g 代表系统级约束; z, x 分别代表系统级全局变量和局部变量; \tilde{y} 代表全局耦合状态变量的响应面模型. CSSO方法的计算框架如图1. 在图1的一个迭代过程中先执行流程1, 再执行流程2.

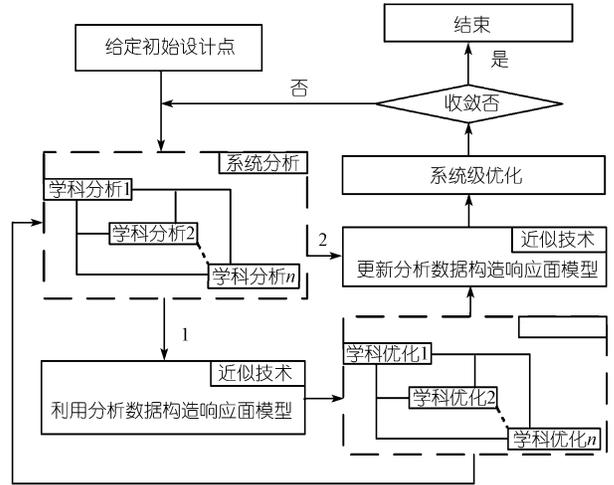


图1 基于响应面方法的CSSO方法框架

CSSO 的特点可总结如下。

1) 在并行子空间优化中, 系统精确分析在优化过程之外完成, 系统级优化在响应面近似模型的基础上进行. 这是CSSO方法降低了计算量的一个重要原因。

2) 本学科的学科分析采用精确分析模型, 来自其他学科的状态变量均采用近似模型来表达, 近似模型不仅简化了计算量, 而且是各个学科间进行信息交换的纽带, 随着优化迭代过程的展开, 近似模型的精确度不断提高, 正是这点保证了并行子空间算法的精度和可靠的收敛性。

CSSO方法是一种适用于大型工程系统分布式设计的MDO方法. 将学科精确分析模型与近似模型相结合, 精确模型使各个学科专家可以充分利用本学科固有资源, 充分提高了各个学科的自治性, 实现各个学科独立并行的优化设计, 近似模型将各个学科之间联系起来, 对学科之间复杂的耦合关系进行了有效的处理. 因此, CSSO方法是工程上最有潜力的MDO方法之一。

2.2 协同优化方法

CO方法最早是1994年由斯坦福大学以Kroo^[6]教授为首的团队在一致性约束方法基础上提出的. CO通过学科级优化, 采用松弛因子实现系统级协调的方式, 将多学科优化设计问题分解为学科级优化和系统级优化, 学科级优化模型如(4)式, 系统级的优化

模型如(5)式.

学科级优化模型.

$$\begin{aligned} \min J_i &= \sum (z_i - z_i^t)^2 + \sum (x_i - x_i^t)^2 + \sum (y_i - y_i^t)^2, \\ \text{s.t. } g(x_i, z_i, y_i(x_i, y_j, z_i)) &\leq 0, \end{aligned} \quad (4)$$

式中, z 表示全局设计变量; x 表示学科设计变量(也称为局部设计变量); y 为状态变量; g 为学科约束条件; 下标 i, j 分别代表第 i, j 个学科, 且 $i \neq j$, 角标“ t ”表示系统级传给学科级的目标值, 在学科级优化中保持为常数.

系统级优化模型.

$$\begin{aligned} \min f(z, y, x), \\ \text{s.t. } J_i^* = \sum (z - z_i^*)^2 + \sum (x - x_i^*)^2 + \sum (y - y_i^*)^2 &\leq \varepsilon, \end{aligned} \quad (5)$$

式中, 角标“*”表示各个学科传给系统级的最优值, 系统级的设计变量包括 3 部分: 全局设计变量 z 、耦合状态变量 y 以及系统目标函数的显性设计变量 x_{iobj} . J 为一致性约束; ε 为松弛因子, 由于优化过程中系统级优化与学科级优化的结果总会存在一些误差, 要想等式约束严格成立是不可能的, 因此 CO 方法通过松弛因子将一致性约束的等式约束转化为不等式约束, 在满足一定精度的情况下, 构成系统级的一致性约束. 图 2 给出 CO 方法的框架图.

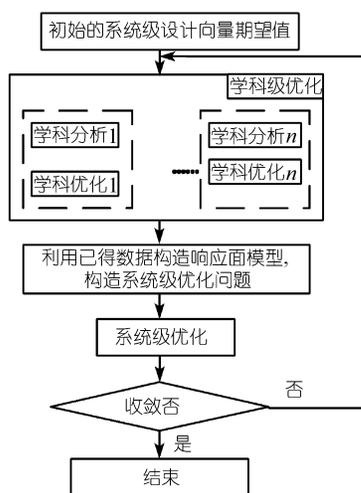


图 2 协同优化方法框架图

CO 的特点可总结如下.

1) 各个学科级并行设计优化, 系统级通过一致

性约束进行协调优化.

2) 所有的状态向量当作设计向量, 设计变量的维数增多.

由于协同优化独特的计算结构, 一般情况下, 系统级设计者和学科级设计者之间的信息传递要经过多次循环迭代过程才可能收敛. 在计算收敛前, 来自各学科的设计向量最优解是不同的, 即学科间存在不一致, 系统级设计者根据来自各学科级的返回信息, 按照一定的规则协调学科间的不一致, 同时使得原问题目标函数最优. CO方法本质上就是系统级协调优化方法.

3 实例分析

为了对这 2 种方法进行更进一步的深入分析, 评价 2 种方法的计算效率, 选用文献以及 NASA Langely 研究中心的经典算例^[7-10], 即数值测试问题(analytic test)、丙烷燃烧问题(combustion of propane)和减速器问题(speed reducer), 进行 CO 和 CSSO 方法比较测试, 这 3 个算例不仅能够充分体现 MDO 问题的多学科耦合的特点, 而且计算简便, 不至于花费太高的计算代价, 可获得便于使用的学科分析代码, 对 CO 方法和 CSSO 方法的比较测试分析研究具有典型代表意义.

算例 1 数值测试问题(analytic test).

数值测试(analytic test)是一个多学科优化方法评估的经典实例, 虽然实例模型简单, 却代表多学科优化设计的所有特性, 包括 2 个学科. 每个学科各有一个状态耦合变量. 2 个学科之间存在着非线性耦合, 各个学科和系统之间也存在着耦合, 这一算例囊括了多学科优化设计的所有术语和技术, 因此在进行方法评价时首先选用该算例, 算例数学描述为

$$\begin{aligned} \min f &= x_2^2 + x_3 + y_1 + e^{-y_2}, \\ \text{s.t. } c_1 &= \frac{y_1}{8} - 1 \geq 0, \\ c_2 &= 1 - \frac{y_2}{10} \geq 0, \\ y_1 &= x_1^2 + x_2 + x_3 - 0.2y_2, \\ y_2 &= \sqrt{y_1 + x_1 + x_3}, \end{aligned} \quad (6)$$

其中, $Y = \{y_1, y_2\}$ 为状态耦合变量; $X = \{x_1, x_2, x_3\}$ 为设计变量; y_1, y_2 分别代表学科状态变量; 等式表示 2 个学科的学科分析, 以下分别简称为学科 1, 学科 2;

c_1, c_2 分别为约束条件. 显而易见, 学科 1 和学科 2 之间有耦合, 按照图 1 计算流程对原问题进行分解计算.

为了便于比较, 2 种方法框架的系统级优化均采用可行方向法(CONMIN), 学科级均采用序列二次规划法(NLPQL). 计算初值取为 $(x_1, x_2, x_3) = (0, 5.0, 5.0)$.

采用 CO 方法计算后系统级目标函数收敛过程如图 3, 采用 CSSO 方法得到系统级目标函数收敛过程如图 4, 从图 4 中可见 CO 方法系统级迭代了 80 多次, 而 CSSO 方法只需迭代 30 多次就可以收敛. 系统级运行时间记录为表 1. 从图中曲线和表中数据明显可以看出算例 1 采用 CO 方法系统级的收敛速度不如 CSSO 方法系统级的收敛速度快.

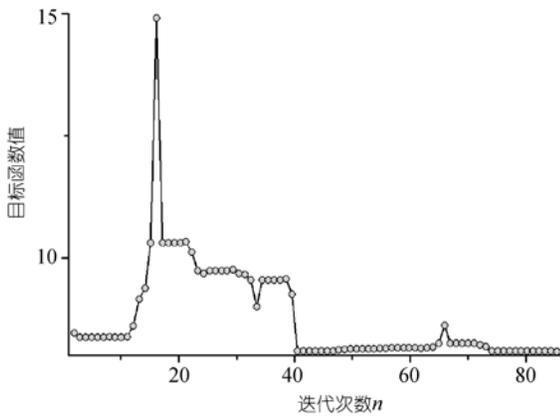


图 3 CO 系统级目标函数迭代过程

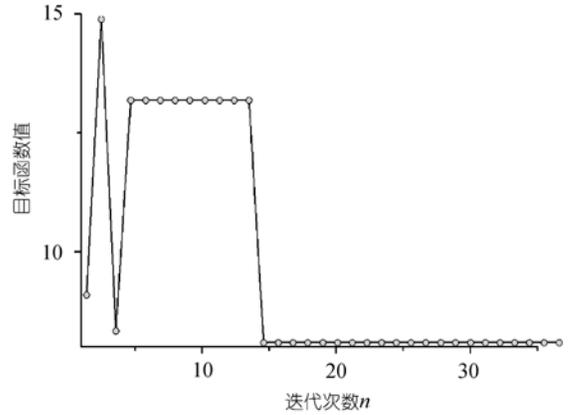


图 4 CSSO 系统级目标函数迭代过程

MDO 计算中, 耗时最长、占用资源最多的就是各个学科分析, 如气动专业的 CFD 计算, 往往需要花费很长的时间计算一个状态, 在 MDO 设计中, 要设法降低学科分析的次数, 减少每个学科迭代的次数. 各个学科迭代次数越小, 学科分析次数越少, 则耗用计算资源越少, 计算效率越高. 算例 1 的 2 个学科级迭代次数见表 2, 从表 2 中数据可见, 算例 1 采用 CO 方法的学科分析迭代次数高于 CSSO 学科分析迭代次数. CSSO 方法的计算效率比 CO 方法的计算效率高.

为了比较这 2 种方法的计算精度, 将文献提供参考值做为精确值, 对 2 种方法设计变量优化解的计算结果进行处理, 整理为表 3, 从表 3 中数据可见, 算例 1

表 1 CO 方法与 CSSO 方法系统级运行时间

序号	名称	CO 算法系统级运行时间/s	CSSO 算法系统级运行时间/s
1	数值测试问题	669	9
2	丙烷燃烧问题	21930	4959
3	减速器问题	252	30

表 2 CO 方法与 CSSO 方法学科级迭代次数比较

序号	名称	CO 迭代次数/次			CSSO 迭代次数/次		
		学科 1	学科 2	学科 3	学科 1	学科 2	学科 3
1	数值测试问题	215	70	/	76	86	/
2	丙烷燃烧问题	4785	81210	25471	2916	4069	4300
3	减速器问题	426	1480	1552	143	185	185

表 3 CO 方法与 CSSO 方法计算精度比较

序号	名称	CO 算法结果误差 R.E.	CSSO 算法结果误差 R.E.
1	数值测试问题	6.164×10^{-6}	7.139×10^{-6}
2	丙烷燃烧问题	3.390×10^{-3}	4.751×10^{-8}
3	减速器问题	3.764×10^{-4}	3.176×10^{-6}

采用 CO 方法的计算精度比采用 CSSO 方法的计算精度高.

算例 2 丙烷燃烧问题(combustion of propane).

丙烷燃烧(combustion of propane)是一个丙烷在空气中燃烧的化学平衡问题, 数学模型描述为

$$\begin{aligned}
 \min \quad & f_2 + f_6 + f_7 + f_9, \\
 \text{s.t.} \quad & f_2(x) \geq 0, \quad f_6(x) \geq 0, \\
 & f_7(x) \geq 0, \quad f_9(x) \geq 0, \\
 & f_2(x) = 2x_1 + x_2 + x_4 + x_7 + x_8 + x_9 + 2x_{10} - R, \\
 & f_6(x) = K_6 x_2^{1/2} x_4^{1/2} - x_1^{1/2} x_6 \left(\frac{p}{x_{11}}\right)^{1/2}, \\
 & f_7(x) = K_7 x_1^{1/2} x_2^{1/2} - x_4^{1/2} x_7 \left(\frac{p}{x_{11}}\right)^{1/2}, \\
 & f_9(x) = K_9 x_1 x_3^{1/2} - x_4 x_9 \left(\frac{p}{x_{11}}\right)^{1/2}.
 \end{aligned} \tag{7}$$

有 8 个约束, 11 个设计变量, 前 10 个设计变量为每摩尔丙烷燃烧产生的各产物的摩尔数, 第 11 个变量是用于简化计算所引入的未知量, 是其余 10 个未知量的和. 这 11 个设计变量可细化为 4 个全局设计变量和 7 个学科设计变量(局部变量), 该问题的固定参数有气压 p 、空气与燃料的比率 R 和测量数据 K_i , $i = 5, 6, \dots, 10$. 对方法进行测试时, 各固定参数采用文献[9,10]中给定的值, 分别为

$$\begin{aligned}
 R &= 10, p = 40, K_5 = 1.0, K_6 = 1.0, K_7 = 1.0, \\
 K_8 &= 0.1, K_9 = 1.0, K_{10} = 0.1.
 \end{aligned}$$

(7)式中的 $x_2, x_4, x_5, x_8, x_9, x_{10}, x_{11}$ 均为耦合变量, 满足下列状态方程

$$\begin{cases}
 f_1(x) = x_1 + x_4 - 3 = 0, \\
 f_3(x) = 2x_2 + 2x_5 + x_6 + x_7 - 8 = 0, \\
 f_4(x) = 2x_3 + x_9 - 4R = 0, \\
 f_5(x) = K_5 x_2 x_4 - x_1 x_5 - 3 = 0, \\
 f_8(x) = K_8 x_1 - x_4 x_8 \left(\frac{p}{x_{11}}\right) = 0, \\
 f_{10}(x) = K_{10} x_1^2 - x_4^2 x_{10} \left(\frac{p}{x_{11}}\right) = 0, \\
 f_{11}(x) = x_{11} - \sum_{j=1}^{10} x_j = 0.
 \end{cases} \tag{8}$$

设计变量最优解的参考值为 $(x_1, x_2, x_3, x_4, x_5,$

$x_6, x_7, x_8, x_9, x_{10}, x_{11}) = (1.378887, 1.872886, 18.426810, 1.621113, 0.0498605, 1.094798, 0.931314, 0.167939354, 3.14638, 0.05198618, 28.741974)$, 目标函数值为 0. 需要对原问题进行分解计算, 将原优化模型分解为系统级优化和学科级优化.

2 种方法的系统级优化均采用可行方向法(CONMIN), 学科级均采用序列二次规划法(NLPQL). 计算初值均取为 $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}) = (2.0, 1.0, 20.0, 1.0, 1.0, 0, 0, 1.0, 1.0, 1.0, 28.0)$.

CO 方法和 CSSO 方法的系统级运行时间记录为表 1. 从表 1 中数据明显可以看出算例 2 采用 CO 方法系统级的收敛速度不如采用 CSSO 方法系统级的收敛速度快.

算例 2 的两个学科级迭代次数如表 2, 从表 2 中数据可见, 本算例 CO 方法的学科分析迭代次数高于 CSSO 学科分析迭代次数.

为了比较这 2 种方法的计算精度, 将文献提供参考值做为精确值, 对 2 种方法设计变量优化解的计算结果进行处理, 整理为表 3, 从表 3 中数据可见, 本算例 CO 方法的计算精度不如 CSSO 的计算精度高.

算例 3 减速器问题(speed reducer).

这是一个减速器齿轮箱的算例, 减速器多学科设计优化的目标是在满足齿轮箱中的转轴和齿轮大量约束的同时, 使得齿轮箱重量最轻(体积最小). 减速器通常安装在轻型飞机发动机和螺旋桨之间, 用于调节两部件的转动速度, 使其保持最大工作效率的转动速度. 该优化问题有 7 个设计变量, 分别为齿面宽度 x_1 , 齿轮模数 x_2 , 小齿轮齿数 x_3 , 大小齿轮之间的轴承间距 x_4, x_5 , 大小齿轮轴的轴直径 x_6, x_7 , 如图 5. 需要满足的约束条件包括: 齿轮的最大弯曲应力不能超过规定值、齿轮最大接触应力不能超过规定值、大小齿轮轴的最大横向挠度需满足设计值、大小

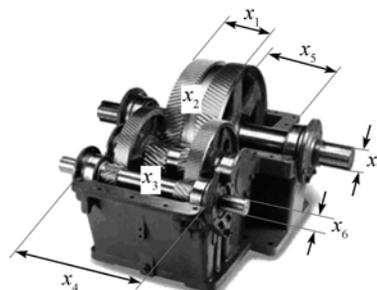


图 5 减速器设计变量

齿轮的轴内最大应力值不能超过规定值、齿轮大小满足尺寸和空间的限制约束、齿轮轴满足轴尺寸限制以及满足设计变量上下限约束等。

考虑各种约束条件以及设计变量上下限的约束条件, 将变速器多学科优化的优化模型整理为下式

$$\begin{aligned} \min \quad & C_{f1}x_1x_2^2(C_{f2}x_3^2+C_{f3}x_3-C_{f4})-C_{f5}(x_6^2+x_7^2)x_1+ \\ & C_{f6}(x_6^3+x_7^3)+C_{f1}(x_4x_6^2+x_5x_7^2), \\ \text{s.t.} \quad & 1-\frac{x_1}{C_{g9}x_2} \geq 0, 1-\frac{x_1}{C_{g11}} \geq 0, \\ & g_1 = \frac{C_{g1}}{x_2^2x_3} \leq 1, g_2 = \frac{C_{g2}}{x_2^2x_3^2} \leq 1, \\ & g_3 = \frac{C_{g3}x_4^3}{x_2x_3x_6^4} \leq 1, g_4 = \frac{C_{g4}x_5^3}{x_2x_3x_7^4} \leq 1, \\ & g_5 = \frac{\sqrt{\left(\frac{C_{A12}x_4}{x_2x_3}\right)^2 + C_{A1}}}{C_{g5}C_Bx_6^3} \leq 1.0, \\ & g_6 = \frac{\sqrt{\left(\frac{C_{A12}x_5}{x_2x_3}\right)^2 + C_{A2}}}{C_{g6}C_Bx_7^3} \leq 1.0, \\ & g_7 = \frac{x_2x_3}{C_{g7}} \leq 1.0, g_8 = \frac{C_{g8}x_2}{x_1} \leq 1.0, \\ & g_9 = \frac{x_1}{C_{g9}x_2} \leq 1.0, g_{10} = \frac{C_{g10}x_6 + C_{g105}}{x_4} \leq 1.0, \\ & g_{11} = \frac{C_{g11}x_7 + C_{g105}}{x_5} \leq 1.0, \\ & 2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, \\ & 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, \\ & 7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, \\ & 5.0 \leq x_7 \leq 5.5, \end{aligned} \tag{9}$$

其中, g_1 为齿轮的最大弯曲应力; g_2 为齿轮最大接触应力; g_3, g_4 为轴的最大横向挠度; g_5, g_6 为轴内最大应力; g_7, g_8 和 g_9 为尺寸和空间限制约束; g_{10}, g_{11} 为轴尺寸计算的经验公式。

优化模型中各系数分别取值如下

$$\begin{aligned} C_{f1}=0.7854, C_{f4}=43.0934, C_{f2}=3.3333, C_{f3}=14.9334, \\ C_{f6}=7.477, C_{g1}=27.0, C_{g2}=397.5, C_{g3}=1.93, \\ C_{g4}=1.93, C_{g5}=1100.0, C_{A12}=745.0, C_{A1}=0.169 \times 10^8, \end{aligned}$$

$$C_B = 0.1, C_{A2} = 0.1575 \times 10^9, C_{g6} = 850.0.$$

设计变量最优解的参考值为 $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)=(3.5, 0.7, 17.0, 7.3, 7.7153199, 3.35, 5.287)$, 目标函数值为2994.355026. 采用CO与CSSO方法将原优化问题分解为系统级优化与学科级优化。

2种方法的系统级优化均采用可行方向法(CONMIN), 学科级均采用序列二次规划法(NLPQL). 计算初值均取为 $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)=(3.5, 0.7, 17.0, 7.30, 7.715, 3.35, 5.287)$.

CO方法系统级目标函数迭代历史曲线如图6, CSSO方法系统级目标函数迭代历史曲线如图7. 从图6和7中可见CO方法系统级迭代了50多次, 而CSSO方法只需迭代40多次. 系统级运行时间记录为表1. 从图6和7中曲线和表1中数据明显可以看出算例CO方法系统级收敛速度不如CSSO方法系统级收敛速度快。

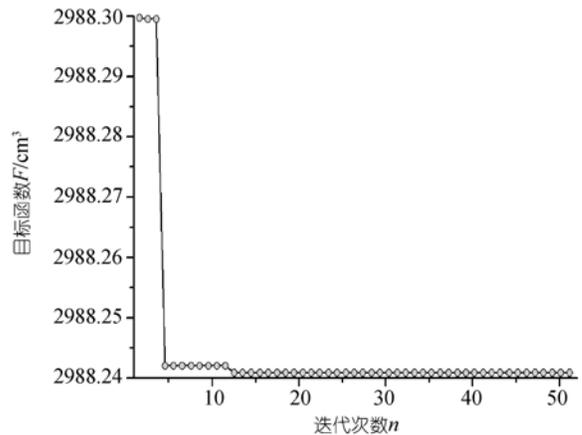


图6 CO方法系统级目标函数迭代过程图

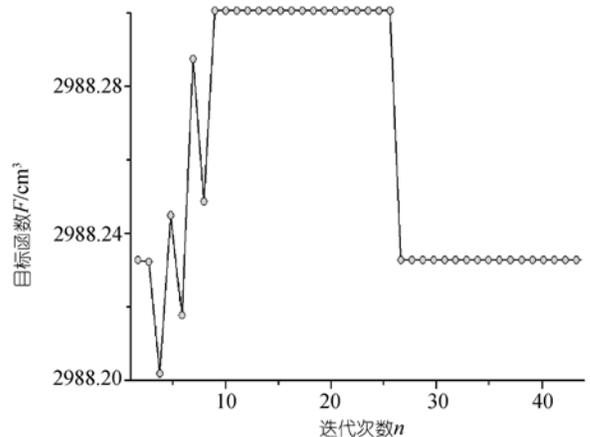


图7 CSSO方法系统级目标函数迭代过程图

学科级迭代次数对于方法的计算效率影响很大, 学科级迭代次数越多, 方法的计算效率越低. 2 个学科级迭代次数如 2, 从表 2 中数据可见, 本算例 CO 方法的学科分析迭代次数高于 CSSO 学科分析迭代次数.

为了比较这 2 种方法的计算精度, 将文献提供参考值做为精确值, 对 2 种方法设计变量优化解的计算结果进行处理, 整理为表 3, 从表 3 中数据可见, 本算例 CO 方法的计算精度不如 CSSO 的计算精度高.

表 3 中, 结果误差 R.E. 用下式计算

$$\text{R.E.} = \sqrt{\sum_{i=1}^k (x_i - x_{i\text{exact}})^2}, \quad (10)$$

式中, k 为设计变量的总数, $x_{i\text{exact}}$ 表示设计变量的精确解. x_i 为不同算法设计变量优化结果.

4 结论

通过以上分析可以得到如下结论.

1) 从提高计算效率来讲, CSSO 方法的计算效率高.

相对于协同优化方法而言, 并行子空间方法大大减少了学科分析迭代的次数, 从而大大减少了总的计算量. 并行子空间方法的计算效率高于协同方法的计算效率.

2) 从提高学科自治来讲, CSSO 方法提高了各个学科的自治性.

CSSO 方法将学科精确分析模型与近似模型相结合, 精确模型使各个学科专家可以充分利用本学科

固有资源, 充分提高了各个学科的自治性, 实现各个学科独立并行的优化设计.

3) 从计算精度来看, 不同算例 2 种方法的精度不同.

通过上面的算例发现, 第 1 个算例 CSSO 方法计算的精度不如 CO 方法计算的精度高, 后 2 个算例 CSSO 方法计算的精度高于 CO 方法计算的精度. 可见不同情况, 2 种方法的计算精度需要具体情况具体分析. CSSO 方法采用学科精确模型与近似模型相结合的方式, 随着数据量的增加和迭代过程的收敛, 响应面近似模型不断地逼近原有的学科分析精确模型. 可以提高 CSSO 方法计算的精度.

4) 从工程实用性方面讲, CSSO 方法的工程实用性强.

方案设计阶段要求“高效、方便、准确”等要求. 并行子空间方法的计算效率满足方案设计阶段的“高效”要求; 学科自治方面的优势, 可以充分利用原有的设计资源, 增强设计阶段的继承性, 满足“方便”的设计要求; 采用精确分析模型与近似模型结合的形式, 提高了分析的精度, 满足“准确”的设计要求, 所有这些决定了 CSSO 方法具有较强的工程实用性.

可见 CSSO 方法在计算效率、学科自治以及工程实用性方面都比 CO 方法略胜一筹. 在多学科优化设计过程中, 从工程应用性方面来讲, 可以首先考虑采用 CSSO 方法.

参考文献

- 1 Natalia M. Alexandrov, srinivas kodiyaalam. In: Initial Results of An MDO Method Evaluation Study. AIAA-98-4884
- 2 Hulme K F, Bloebaum C L. A comparison of solution strategies for simulation-based multidisciplinary design optimization. AIAA-98-4977
- 3 Tedford N P, Joaquim R R A. Martins, comparison of MDO architectures within a universal framework. AIAA 2006-1617
- 4 Perez R E, Liu H H T. Evaluation of multidisciplinary optimization approaches for aircraft conceptual design. AIAA 2004-4537
- 5 Sobieszcanski-Sobieski J. A step from hierarchic to non-hierarchic systems. AIAA CP-3031, 1989
- 6 Kroo I, Altus S, Braun R, et al. Multidisciplinary optimization methods for aircraft preliminary design. AIAA 94-4325
- 7 Sellar R S, Batill S M, Renaud J E. Response surface based, concurrent subspace optimization for multidisciplinary system design. AIAA 96-0714
- 8 <http://mdob.larc.nasa.gov/>
- 9 Padula S L, Alexandrov N, Green L L. MDO test suite at NASA langley research center. AIAA Paper No. 96-4028
- 10 Tedford N P, Joaquim R R A. Martins, comparison of MDO architectures within a universal framework. AIAA 2006-1617