基于 KS 函数的协同优化算法及其应用 *

Collaborative Optimization Based on KS Function and Its Application

西北工业大学航天学院　粟　华　谷良贤　龚春林

[摘要]　多学科设计优化 (MDO) 是当前航空航天复杂系统设计研究中一个最新、最活跃的领域，针对多学科设计优化中协同算法优化过程中存在的收敛困难、计算量大、费时等问题，提出一种新的系统级约束处理方法，利用 KS 函数的凝聚特性，将系统级的多个学科约束凝聚成一个拥有光滑高阶可微的可行域单约束，来取代罚函数法中的多个约束，以缓解多约束之间的冲突并加速收敛速度。利用两个典型算例进行测试，结果表明改进方法有效。

关键词：KS 函数　协同优化方法　罚函数　约束优化

[ABSTRACT]　Recently, MDO (Multidisciplinary design optimization) is one of the latest and the most active fields in complex system design research for aerospace. Because the normal conventional collaborative optimization strategy is hard-convergent, computationally expensive and time-consuming, a new constraint method is introduced in system level. It puts several constraints in disciplines into one smooth and differentiable constraint by using the condensation characteristic of KS function, and replaces the constraints in the penalty function, then reduces the interference between the multi-restraints and accelerates the convergent speed. This method is tested by two typical examples, and the result shows that the method is effective.

Keywords: KS function　Collaborative optimization　Penalty function　Constrained optimization

航空航天系统中设计对象复杂、参与学科多、设计变量多，是一个相互耦合的非线性优化问题，一般的优化方法很难解决。针对航空航天领域中复杂的设计优化问题，提出了多学科设计优化（Multidisciplinary Design Optimization，MDO）的概念，并于 20 世纪 80 年代后期逐渐形成，目前正逐渐发展成为航空航天领域的一个重要研究方向 [1]。在可重复使用运载器、气动喷管及卫星概念设计中都得到了应用。

*　国家自然科学基金（60774087）项目资助。

协同优化算法（Collaborative Optimization，CO）是近年来兴起的一种多学科优化算法 [2-4]，该方法将复杂系统的优化设计问题分解为一个系统级优化加上多个学科级优化，从而简化学科之间的关系来处理学科耦合问题。CO 中系统级求解方法主要有：系统级敏感度方法、响应面方法、约束松弛法、线性近似子空间方法和罚函数法等。西北工业大学的李响等 [5] 提出了超球近似子空间法，国防科大的杨维维等 [6] 提出了动态罚因子算法等改进措施。但是对于学科较多、设计变量大的情况，系统级求解仍有困难，并且收敛速度慢。本文提出一种新的系统级约束处理方法，通过 KS 函数将多个学科的复杂约束凝聚为一个单约束并与罚函数算法结合，改进系统级约束处理方式。

1　协同优化

CO 是一种新型的耦合系统求解的多级优化方法，它将一个大规模的设计问题分解成多学科耦合的系统，每个学科可以单独优化而不用考虑其他学科的影响，由系统级进行统一协调，并对设计参数进行分配。

CO 的系统级优化问题表述如下：

$$\min\ f(z),$$

$$s.t.\quad J_i(z,p)=\sum_{j=1}^{h_i}(p_{ij}-z_{ij})^2=0\ (i=1,2,\cdots,n),\quad (1)$$

其中，$f(z)$ 为系统级目标函数；$\boldsymbol{z}$ 为系统级设计参数向量；$\boldsymbol{p}$ 为系统级设计状态向量，是学科级优化的设计变量最优解，由学科级传给系统级；J_i 为系统级约束，共有 n 个；h_i 为系统级分配到第 i 个学科级的设计变量个数。

CO 的学科级优化问题表述如下（以第 i 个学科为例）：

$$\begin{aligned}&\min\quad J_i(X,Q)=\sum_{j=1}^{h_i}(x_{ij}-q_{ij})^2,\\&s.t.\quad \begin{matrix}h_{i_a}(x)=0\\g_{i_b}(x)<0\end{matrix},\end{aligned}\quad (2)$$

其中，q 为学科级优化目标向量，即系统级分配下来的设计向量；x 为学科级设计参数向量；h_{i_a} 为学科 i 优化的等式约束，共 a 个；g_{i_b} 为学科 i 优化的等式约束，共 b 个。

从上面系统级约束表达式可以看到，系统级通过兼容约束 $J_i(z,p)=0$ 来协调各个学科之间的一致性，但这只是一种理想状态。在这种状态下，系统级优化问题一般不满足 Kuhn-Tucker 条件，因而是无解的。而且随着学科及其设计参数的增多，系统级设计参数成倍增长，在这种情况下还要满足系统级兼容约束 ，求解十分困难，计算负荷重、效率低。

2 系统级约束的改进

罚函数法将学科约束整合到目标函数中，从而简化了学科约束，提高了系统级处理速度，但由于在 CO 中学科级约束表达形式的特殊性，各个学科约束之间的冲突很大，罚函数法不能很好地解决这个问题。松弛算法在系统级约束中通过一个松弛量代替理想约束情况，但寻优后期系统收敛速度慢、耗时多，也没有很好地解决问题。因此本课题引入 KS 函数对学科的约束先凝聚处理，降低学科间约束的差异性，再结合罚因子使用，综合了两者的优点。

KS 函数由 G.Kreisselmeier 和 R.Steinhauser 于 1979 年提出。设有定义在 n 维欧氏空间的实值函数集合 $g_i(x)$ $(i=1,2,\cdots,n)$，其中 $x\in E^n$，在指数空间上可将该集合进行可微最大包络：

$$KS(\rho,x)=\frac{1}{\rho}\ln\left[\sum_{i=1}^{n}e^{\rho g_i(x)}\right], \tag{3}$$

式中，ρ 为一个视不同问题而给定的控制参数，$\rho>0$。ρ 越大，KS 函数越接近 $g_{\max}(x)=\max_i\{g_i(x)\}$ 形成的包络。利用 KS 函数特有的包络凝聚特性，可将多个约束形成的空间曲面围成的可行域凝聚成只由一个光滑的多维曲面围成的可行域，从而将多约束问题化简为一个单约束的问题，并通过控制参数 ρ 来调节形成的新包络曲面的精度[7-8]。

下面给出将 KS 函数和罚因子结合使用改进 CO 中系统级约束的过程。将式(1)的系统级约束代入式(3)，用 KS 函数进行凝聚变为：

$$KS(z,p,\rho)=\frac{1}{\rho}\ln\left[\sum_{i=1}^{n}e^{\rho\sum_{j=1}^{h}(p_{ij}-z_{ij})^2}\right]。 \tag{4}$$

罚因子算法将系统级中的约束条件通过 1 个惩罚因子添加到系统的目标函数中，能够灵活地处理多约束问题，由于采用罚函数简化了约束的处理过程，寻优速度比其他方法要快。采用罚因子算法处理的系统级约束表示形式为：

$$\begin{aligned}&\text{mim}\ \ f'(z)=f(z)+M\sum_{i=1}^{n}J_i^2(z)\ \ ,\\&s.t.\ z_{\min}\leqslant z\leqslant z_{\max}\ \ 。\end{aligned} \tag{5}$$

当 M 值不变时，称为静态罚因子算法；而如果通过学科间不一致约束

$$r=\sqrt{\sum(p_i-p_j)^2}\ \ (i,j=1,2,\cdots,n;i\neq j)$$

来调节 M 值，则为动态罚因子算法。

为了利用罚因子算法求解速度快的优势，本课题提出一种 KS 罚函数法，用 KS 函数凝聚后的单约束式 (4) 代替式(5)中的 $\sum_{i=1}^{n}J_i^2(z)$，得到一个新的系统级约束：

$$\begin{aligned}&\text{mim}\ f^*(z,M,\rho)=f(z)+M\frac{1}{\rho}\ln\left[\sum_{i=1}^{n}e^{\rho\sum_{j=1}^{h}(p_{ij}-z_{ij})^2}\right],\\&s.t.\ z_{\min}\leqslant z\leqslant z_{\max}\ \ 。\end{aligned} \tag{6}$$

当 M 值不变时，称为 KS 罚函数法；当 M 值随着迭代过程可变时，称为动态 KS 罚函数法。在这种情况下，由于算法采用和动态罚函数算法相似的处理机制，算法有更好的收敛速度和稳定性。

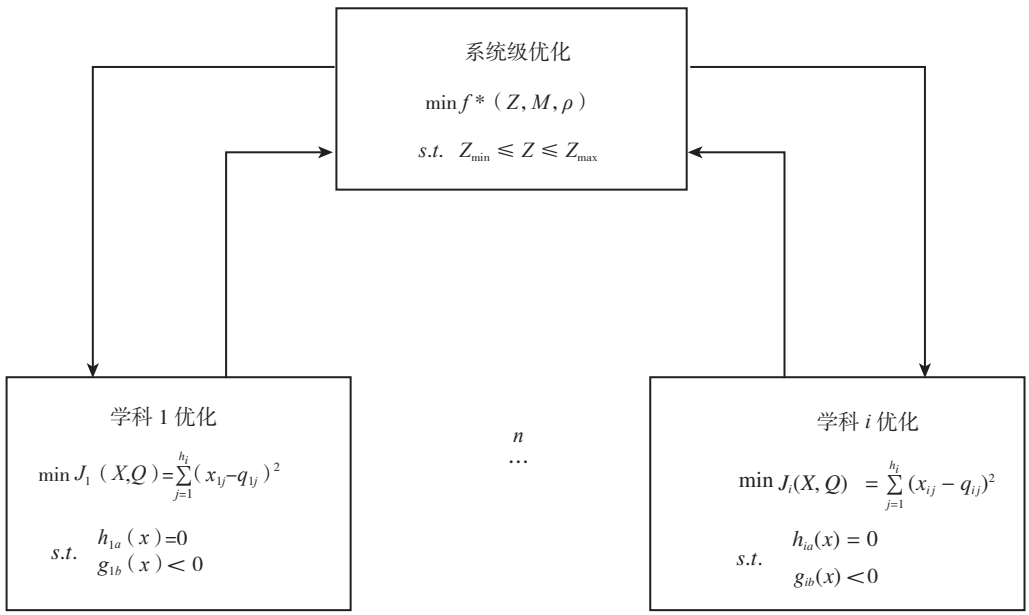

图1 CO流程框架

Fig.1 Framework of CO

改进后的 CO 框图见图 1。

改进后系统级约束的求解方式有罚因子 M 和精度 ρ 这 2 个调节参数，其中 M 主要控制约束的强弱，而 ρ 主要影响 KS 凝聚函数的近似精度。

3 算法验证

采用 2 个算例对上述改进算法进行验证，分别采用动态约束松弛法、动态罚函数法、KS 罚函数法和动态 KS 罚函数法来比较，所有程序使用 MATLAB 编程计算。

算例 1 为 1 个耦合的 2 个子空间设计问题。y_1、y_2 为 2 个子空间的耦合设计变量。

$\min\ f(x)=x^2_2+x_3+y_1+e^{-y_2}$；

子空间 1：$y_1=x^2_1+x_2+x_3-0.2y_2$ ；

子空间 2：$y_2=\sqrt{y_1}+x_1+x_3$ ；

$$g_1=\frac{y_1}{8}-1\geqslant 0, g_2=1-\frac{y_2}{10}\geqslant 0,$$

$-10\leqslant x_1\leqslant 10, 0\leqslant x_2\leqslant 10, 0\leqslant x_3\leqslant 10$。

算例 1 结果见表 1。理论全局最优解为 Z={3.03, 0, 0}，对应的目标函数值为 8.03。选取起始点 {0 0 0 0 0}，精度为 0.0001（相对误差）。

表1 算例1结果

算 法	超球近似子空间	动态罚因子	KS 罚函数	动态 KS 罚函数
x_1	3.0280	3.0154	3.0094	3.0280
x_2	0.0015	0.0671	0.1016	0.0020
x_3	0	0	0	0
y_1	7.9970	7.9642	7.9800	7.9960
y_2	5.8526	5.8268	5.8328	5.8554
F	7.99982	7.9808	7.99329	7.99892
寻优次数	36	46	54	34
计算时间	2.159149	2.690083	2.929379	1.729989

算例 2 是一个经典的工程应用减速器齿轮设计问题。其优化的数学模型如下：

$$\begin{cases}\min f(X)=0.7854x_1x_2^2(3.333x_3^2+14.933x_3-43.0934)-\\ \qquad 1.508x_1(x_6^2+x_7^2)+7.477(x_6^3+x_7^3)+\\ \qquad 0.7854(x_4x_6^2+x_5x_7^2)\\ s.t.\ g_i\leqslant 0\ (i=1,2,\cdots,11)\end{cases},$$

其中，

$$g_1=\frac{27}{x_1x_2^2x_3}-1\leqslant 0\ （齿的弯曲应力约束），$$

$$g_2=\frac{397.5}{x_1x_2^2x_3^2}-1\leqslant 0\ （齿的接触应力约束），$$

$$g_3=\frac{1.93x_4^3}{x_2x_3x_6^4}-1\leqslant 0\ （轴 1 横向变形约束），$$

$$g_4=\frac{1.93x_5^3}{x_2x_3x_7^4}-1\leqslant 0\ （轴 2 横向变形约束），$$

$$g_5=\frac{A_1}{B_1}-1000\leqslant 0\ （轴 1 的应力约束），$$

$$g_6=\frac{A_2}{B_2}-850\leqslant 0\ （轴 2 的应力约束），$$

$$g_7=x_2x_3-40\leqslant 0\ ，$$

$$g_8=x_1x_2-12\leqslant 0\ ，$$

$$g_9=-\frac{x_1}{x_2}+5\leqslant 0\ ，$$

$$g_{10}=\frac{1.5x_6+1.9}{x_4}-1\leqslant 0\ ，$$

$$g_{11}=\frac{1.1x_7+1.9}{x_5}-1\leqslant 0\ ，$$

$$\begin{cases}A_1=\left[\left(\frac{745x_4}{x_2x_3}\right)^2+16.9\times 10^6\right]^{0.5}\\ A_2=\left[\left(\frac{745x_5}{x_2x_3}\right)^2+157.5\times 10^6\right]^{0.5}\\ B_1=0.1x_6^3\\ B_2=0.1x_7^3\end{cases}。$$

变量取值范围如下：

$2.6\leqslant x_1\leqslant 3.6$ ， $0.3\leqslant x_2\leqslant 1.0$ ，

$17\leqslant x_3\leqslant 28$ ， $x_4\geqslant 7.3$，$x_5\leqslant 8.3$ ，

$2.9\leqslant x_6\leqslant 3.9$ ， $5\leqslant x_7\leqslant 5.5$ 。

其中，x_1 为齿宽系数；x_2 为齿轮模数；x_3 为小齿轮的齿数；x_4、x_5 为轴承间距；x_6、x_7 为轴承直径。该问题的最优解为：{3.5 0.7 17 7.3 7.71 3.35 5.29}，目标函数为 2994。

将轴 1、轴 2 和齿轮分解成 3 个学科，计算结果见表 2。

4 结束语

从以上 2 个算例可以看出，KS 罚函数和动态罚因子法计算性能基本一致，而加入了动态参数的动态 KS 罚函数法由于结合了罚函数和 KS 函数的优点，无论是在消耗时间上还是在计算精度上都比超球近似子空间和动态罚因子要好一些，可以很好地解决使用其他方法存在的精度不高、求解速度低的问题。这得益于 KS 函数将多个学科的约束凝聚为单个约束，并且这种约束形

（下转第 91 页）