文章编号：1000-8608(2010)06-0931-06

变粒度协同进化设计算法及其在卫星舱布局设计中应用

陈　羽，滕弘飞*

（大连理工大学 机械工程学院，辽宁 大连　116024）

摘要： 采用进化算法求解复杂卫星舱布局问题时，算法容易陷入局部最优，且干涉计算复杂度高，计算耗时长. 为提高对复杂解空间的搜索能力，基于协同进化算法，将问题分解为若干子问题求解；为减少计算耗时，子问题求解时采用了一种设计变量的变粒度策略. 称上述方法为变粒度合作式协同进化算法(CCEA-CFG). 卫星舱布局优化数值实验表明，与目前常用的几种布局求解算法(遗传算法、协同进化算法以及遗传/粒子群算法(QPGP))相比，CCGA-CFG(基于GA的CCEA-CFG)具有较好的计算质量、计算效率和计算鲁棒性.

关键词： 协同进化；变粒度；布局设计

中图分类号： TP391.75；O122.4；V47　　**文献标志码：** A

0　引　言

卫星舱有效荷载的布局设计问题属于NP难的组合优化问题[1]，其解空间具有非线性、多峰的复杂形态，传统的进化计算方法求解此类规模较大的复杂系统优化设计问题，面临巨大的困难. 进化算法，如遗传算法[2]等，是目前求解复杂布局问题的主要方法，但是其每次迭代计算的适应度评价都要进行干涉量计算，干涉计算的计算复杂性很高，严重影响了随机搜索算法的效率. 另一方面，布局问题解空间多峰的性质，容易使算法陷入早熟，很难搜索到全局最优解，甚至很难获得工程可行解.

近年来，为了解决日益复杂的优化问题，协同进化算法成为一个引人关注的研究热点. 该算法采用分而治之策略，可用于求解工程系统设计问题. 协同进化算法可分为竞争式和合作式协同进化算法，本文研究后者. 1994年Potter等[3]提出了合作式协同进化算法(cooperative coevolutionary algorithm，CCEA). 研究表明，CCEA适合于多Agent学习系统[4]，可通过与小生境技术结合求解多峰优化函数[5]. 另外，CCEA在许多工程领域也得到初步应用[6,7]. 但CCEA也有自身的局限：一是优化搜索需要比一般进化算法更多的适应度评价次数，这是CCEA实际应用中的掣肘之一；二是CCEA有一定的适用范围，一般认为，CCEA适合求解分解后子系统之间耦合较弱的问题.

许多实际工程问题的适应度(评价)函数较为复杂，精确计算困难而且相当耗时，如结构设计中的有限元分析、计算流体动力学(computational fluid dynamics，CFD)等[8]. 为尽量减少求解此类问题的适应度评价次数，减少计算耗时，通常采用三类方法[9]：一是从“部分解”到“完整解”的进化，即先优化问题中最重要的部分(框架)，再逐渐把其他相对次要的部分考虑进来；二是并行计算；三是采用适应度函数的近似模型(近似模型的计算较实际模型简单). 本文研究的卫星舱复杂布局问题可进行结构性分解，符合采用从“部分解”到“完整解”进化策略(第一类方法)的条件[9]. 另一方面，对于布局问题，算法可在优化过程中根据设计变量对设计目标影响的大小，按一定顺序对设计变量进行优化，有助于提高布局问题的求解效率[10].

本文将卫星舱布局问题按物理结构分解为若干子系统，采用协同进化框架进行优化，增强对多峰解空间的寻优搜索. 与此同时，为了缓解干涉量计算带来的计算耗时问题，给出一种由粗到细的设计变量变粒度策略，将其应用于各子种群的优化搜索之中. 上述方法称为变粒度合作式协同进化算法(cooperative coevolutionary algorithm

收稿日期： 2008-03-04；　**修回日期：** 2010-04-04.
基金项目： 国家自然科学基金资助项目(50575031，60674078)；“八六三”国家高技术研究发展计划资助项目(2006AA04Z109).
作者简介： 陈　羽(1981-)，男，博士生；滕弘飞*(1936-)，男，教授，博士生导师，E-mail：tenghf@dlut.edu.cn.

with coarse-to-fine grain strategy,CCEA-CFG),旨在通过协同进化的系统分解和设计变量变粒度策略(相当于对问题进行降维处理)减少干涉计算的耗时,提高求解的效率和质量.

1 变粒度协同进化算法

1.1 协同进化算法框架

构造协同进化算法框架的初衷是求解越来越复杂的优化问题.其基本思想是,把复杂系统问题分解成规模较小,且可独立处理的子系统(问题),各子系统的设计变量没有重叠;每个子系统都采用进化算法进行并行的演化计算.各个子系统从适应度评价角度进行协同——子系统个体评价时需从其他子系统选取代表个体,组成完整解,用原问题的适应度函数进行评价.CCEA 的求解效果与问题本身的性质以及问题分解方式有直接的关系.Wiegand [11]对 CCEA 的作用机制、目标函数的可分离性,以及适应度评价时的代表个体选择方面等做了理论分析,这些分析以各种优化函数为基础,是经验性的.刘静等[12]也对协同进化算法展开了研究.然而,要严格地从数学上对 CCEA 的作用机理进行理论上的证明是相当困难的[11].目前,CCEA 的作用机理研究主要通过对优化函数的实验,再结合经验性分析(见文献[13])进行.这是有待深入研究的问题.

1.2 设计变量变粒度的思想和方法

用进化算法求解布局问题时,计算耗时的主要原因在于干涉量计算的计算复杂性高,而且进化算法的每一次适应度评价都要进行干涉量计算,从而使整个优化搜索过程的耗时长,严重影响效率.

设计变量变粒度策略是本文针对复杂布局问题设计的一种启发式方法,用以缓解因采用 CCEA 进行优化搜索带来的适应度评价次数的增加.这种启发式策略思想主要受由粗粒度到细粒度的建模方法 (coarse-to-fine grain modeling)[9]和文献[10]的启发.

本文以卫星舱布局设计为例,说明设计变量变粒度策略的具体做法.首先,计算设计变量对目标函数的影响程度大小(多学科设计优化(MDO)中[14]常用"灵敏度"表示此种影响程度).接着,对每个子系统中的设计变量按其对目标函数的影响程度大小排序,并根据经验将排序后的设计变量分为若干组(本文也称之为若干批次).优化搜索先优化由影响程度最大的一组设计变量(初始设计变量)组成的粗粒度模型,求得此粗粒度模型的优秀解,相当于原问题一个优良的"框架解";在此"框架解"的基础上,将影响程度较大的下一组设计变量添加到已有粗粒度优化模型中(参与进化计算),使模型的"粒度"细化(更为完整),再搜索得到此模型的优化解,得到一个更为充实完整的"框架解";如此反复,最终得到原问题的优化解.下面对设计变量变粒度策略的具体实现进行说明.

1.2.1 *设计变量排序* 对于一般设计问题来说,要计算设计变量对于目标函数的影响程度,数学建模困难,计算过程也较复杂[10].本文卫星舱布局问题以优化全舱质量特性为目标,可根据工程经验[15],近似用该待布物的质量 m_i 来表征该待布物设计变量对于目标函数的影响程度.

每个待布物一般有 4 个设计变量,设第 i 个待布物的设计变量为

$$u_i = \{x_i, y_i, z_i, \alpha_i\} \tag{1}$$

其中 (x_i, y_i, z_i) 和 α_i 分别为第 i 个待布物的质心坐标和方向角.本文设计变量的变粒度策略在设计变量的排序和分组时,以待布物 u_i 为单位,即假设 (x_i, y_i, z_i) 和 α_i 的灵敏度相等(都等于 m_i),这是一种近似的做法.在当前的系统分解下,对每个子系统的设计变量按所属待布物的质量 m_i 由大到小排序.对于一般设计问题,计算设计变量对于目标函数的影响程度可参考文献[16]的方法.

1.2.2 *按待布物质量排序分组(粗-细粒度模型的构造)* 如前所述,本文"粗-细"(由粗到细)粒度模型的构造是按待布物质量将待布物分为若干组(批次).分组规则如下.

(1) 确定初始待布物

质量最大的一组待布物的设计变量最先优化.根据本文实验经验(包括文后算例),当子系统待布物数量 $N_e \leqslant 10$ 时,取子系统的初始待布物数量 $N_e^{ini} = N_e \times 50\%$.

(2) 剩余待布物分组

确定了初始待布物之后,每个子系统剩余 $N_e - N_e^{ini}$ 个待布物.将子系统中剩余设计变量平均分为 t 组,那么,子系统每组有 $(N_e - N_e^{ini})/t$ 个待布物,一般取 $N_e^t \in [2,5]$;同时,考虑把质量相近的待布物尽量分在一组.t 和 N_e^t 的大小需根据实际情况调整.

由于待布物与其设计变量有对应关系(见式(1)),这样就相当于得到了各子系统的设计变量分组.

1.2.3 *变粒度规则* 本文的变粒度规则是指在

进化计算过程中，判断当前粒度模型是否已获得优化解（“框架解”），可向当前的模型中加入新的一组设计变量的准则. 在此，本文根据卫星舱布局特点，给出变粒度规则如下：

当满足以下两个条件之一时，向系统中添加一组设计变量.

(1)系统最优个体适应度满足

$$\left|\frac{F^k(\boldsymbol{X})-F^{k-m}(\boldsymbol{X})}{F^k(\boldsymbol{X})}\right|\leqslant\varepsilon_k \tag{2}$$

式中：$F^k(\boldsymbol{X})$ 及 $F^{k-m}(\boldsymbol{X})$ 分别表示系统在 k 和 $k-m$ 代时的适应度；ε_k 为预先给定值，当间隔代数 $m=200$，取 $\varepsilon_k=1/F^k(\boldsymbol{X})\times10^{-1}$.

(2) 当前进化代数与上次添加设计变量的间隔大于给定值. 为了使每次添加设计变量之后，系统有足够的时间进行进化，得到一个相对较好的“框架解”作为下一阶段进化的基础，应保证每两次添加设计变量的间隔代数足够大. 一般取间隔代数大于 500，根据实际问题和设计变量分组而定. 问题求解难度大，或者加入设计变量较多，可以考虑增大间隔代数.

2　卫星舱布局问题

卫星舱布局设计问题是指将一组待布物布置在给定的空间内，满足各种约束条件要求，并使得设计目标性能参数最优化. 本文研究一类以改善卫星全舱质量分布特性为目标，同时满足不干涉约束、静稳定性约束和动平衡度约束的卫星舱布局设计问题（它们的计算公式见文献[17]，文献[17]中的式(5a)～(5c)有误，质心计算应该包括舱体质量，即式(5a)～(5c)中的 i，应该是 $i=0$，而不是 $i=1$. 另外，算例 2 中的 m_0 值有笔误，应更正为 $m_0=576.534$ kg（文献[17]实际按此计算）），其设计实质是确定各个待布物的放置位置和方向.

卫星舱布局优化设计数学模型可表示为[17]

$$\begin{cases}\min\limits_{\boldsymbol{X}} & f(\boldsymbol{X})\\ \text{s.t.} & g_i(\boldsymbol{X})\leqslant0;\ i=1,2,\cdots,n\end{cases} \tag{3}$$

式中：$\boldsymbol{X}$ 为设计变量集合，包括每个待布物的三维坐标和方向角，即 $\boldsymbol{X}=(u_1\quad u_2\quad\cdots\quad u_N)^{\mathrm{T}}$，$N$ 为待布物数量；$f(\boldsymbol{X})$ 为目标函数；$g_i(\boldsymbol{X})(i=1,2,\cdots,n)$ 表示性能约束条件，n 为性能约束的数量. 用惩罚函数法，将带约束的卫星舱布局问题转换为无约束的优化问题，其数学模型可表示为

$$\min_{\boldsymbol{X}}F(\boldsymbol{X})=f(\boldsymbol{X})+\sum_{i=1}^{n}w_ig_i(\boldsymbol{X}) \tag{4}$$

其中 $F(\boldsymbol{X})$ 为适应度函数，w_i 为惩罚因子. 该数学模型的详细叙述见文献[17].

3　仿真实验

3.1　算例描述

本文算例引自文献[17]的算例 2. 设在卫星舱体内的 4 个承载隔板面上，要布置给定的 60 个待布物. 各待布物几何尺寸及质量见文献[17]. 其中，长方体待布物为 24 个，圆柱体待布物为 36 个. 假定各待布物为刚体，且质量分布均匀. 已知舱体半径 $R=500$ mm，中心立柱半径 $R_c=100$ mm，上、下承载隔板到底面的高度分别为 $H_2=830$ mm，$H_1=300$ mm，立柱总长 $H_t=1\ 150$ mm，承载隔板厚度 $D_t=20$ mm，舱体壁及舱内固定件总质量 576.534 kg，质心相对参考坐标系 $Oxyz$ 的坐标为(0,0,553.56) mm，卫星空舱时惯性矩阵 $\boldsymbol{I}_0$（惯性单位 kg·m^2）为

$$\boldsymbol{I}_0=\begin{pmatrix}I_x & I_{xy} & I_{xz}\\ I_{yx} & I_y & I_{yz}\\ I_{zx} & I_{zy} & I_z\end{pmatrix}=\begin{pmatrix}352.2 & 0 & 0\\ 0 & 352.2 & 0\\ 0 & 0 & 106.8\end{pmatrix}$$

对全舱整体布局技术要求如下：全舱质心位置的期望值 $x_e=0$，$y_e=0$；允许偏差值 $\Delta x_c=\Delta y_c=\pm3$ mm；舱体不平衡度的许用值 $\delta\theta_x=\delta\theta_y=\delta\theta_z=0.03$ rad. 要求各长方体待布物正交放置（即 $\alpha_i=0$ 或 $\pi/2$）.

3.2　实验设置

所有数值实验均在一台主频为 AMD 3600+，内存为 1 GB 的 PC 机上进行. 分别采用标准 GA、CCGA 以及本文的变粒度合作式协同进化算法对上述包含 60 个待布物的卫星舱布局设计问题进行 100 次随机试算. 因为本文实验在 CCEA-CFG 中采用遗传算法 GA 作为基本算法，所以下文实验将本文算法简写为 CCGA-CFG.

实验的目的如下：

(1)检验变粒度协同进化算法对求解该复杂卫星舱布局问题的有效性.

(2) 通过 CCGA-CFG 与 CCGA 的比较，检验变粒度策略在求解该复杂卫星舱布局问题中的作用和效果.

为便于比较，3 种算法的适应度函数采用同样的一组惩罚因子 (w_i)，式(4) 中：$n=7$，$w_1=8.0$，$w_2=w_3=w_4=7.0$，$w_5=w_6=w_7=100$. GA 的种群规模为 100，CCGA 和 CCGA-CFG 把问题按照上下承载隔板分为两个子系统. CCGA

和 CCGA-CFG 每个子种群的规模为 50(整个系统的规模为 100). CCGA-CFG 的变粒度规则将 60 个待布物分为两批加入优化进程中,第一批待布物共 26 个,其中包含质量较大的前 12 个长方体和前 14 个圆柱体,第二批次加入所有余下的待布物. 加入第二批待布物的时机为进化代数 $G=3\ 000$,即算法运行至 3 000 代时,加入第二批设计变量.

需要特别说明的是,为使上述几种算法的计算时间具有可比性,本文实验中几种算法的干涉量计算采用与文献[17]相同的干涉算法程序. 该程序用解析几何方法求得两正交待布物(圆柱体或长方体)在 xoy 平面上投影的干涉面积. 用干涉面积乘上"较矮"待布物的高,可得到干涉体积. 算法计算评价指标为计算成功率、计算时间和计算精度(目标函数值). 本文将计算成功率定义如下:算法在 100 次随机计算中,工程可行解的次数占算法运行总次数的百分比. 对于本算例而言,工程可行解指,干涉量小于 10 mm^3,质心偏差小于 ±3 mm,全舱三轴的不平衡度均小于 0.03 rad 的解方案. 3 种算法的数值实验统计结果见表 1(每种算法进行 100 次随机试算). 其中,总评价次数=最大进化代数×种群数×种群规模,例如:CCGA-CFG 的总评价次数为 $10\ 000\times2\times50=10^6$ 次;GA 的总评价次数为 $10\ 000\times1\times100=10^6$ 次. 3 种算法适应度进化曲线如图 1 所示. 真实计算中的适应度值为

$$F(\boldsymbol{X})=C-\left(w_8 f(\boldsymbol{X})+\sum_{i=1}^{7} w_i g_i(\boldsymbol{X})\right) \quad (5)$$

其中 C 为一个较大的常数,这里取 $C=50\ 000$,$w_8=100$. 适应度函数值越大越好.

3.3 实验分析

(1)变粒度协同进化算法的有效性

从表 1 的计算结果来看,CCGA-CFG 获得了 3 种算法中最好的设计目标函数值,计算耗时也最少(包括所有的 4 项统计指标). 因此,本文提出的变粒度合作式协同进化算法不仅能有效求解本文的卫星舱复杂布局问题,而且获得了最好的仿真实验结果.

表 1 计算结果统计

Tab. 1 Statistic of the computational results

算法	三轴转动惯量之和/(kg·m²)				计算时间/s				成功率/%	总评价次数
	最小值	平均值	最大值	标准差	最小值	平均值	最大值	标准差		
CCGA-CFG	714.60	718.70	726.86	2.29	149.47	175.52	225.96	12.77	22	10^6
CCGA	728.65	740.14	752.05	5.45	298.92	329.68	404.93	18.48	13	10^6
GA	726.36	736.94	749.20	3.50	227.08	284.36	740.51	50.13	7	10^6

从图 1 的进化曲线看,CCGA-CFG 比 GA 和 CCGA 收敛到优化解的速度更快,而且还具有良好的计算鲁棒性——100 次试算的设计目标和计算时间的标准差最小. 表 2 给出了 CCGA-CFG 的最优布局方案.

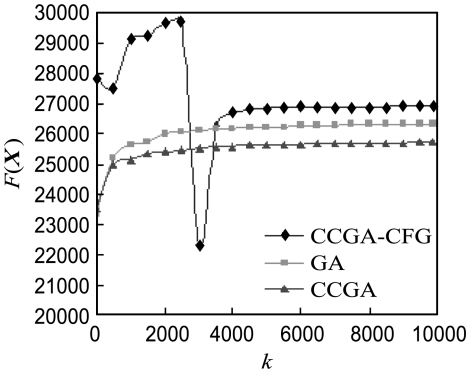

图 1 3 种算法平均适应度进化曲线

Fig. 1 Evolutionary curves of the fitness of three algorithms

(2)变粒度策略的作用

CCGA-CFG 与 CCGA 相比,不仅设计目标的最优值有提高,而且计算耗时缩减了约 47%,计算成功率也提高了 9%. 设计变量变粒度策略的效果较为明显.

从图 1 的进化曲线看,CCGA-CFG 在 3 000 代之前,仅优化第一批待布物,所以设计目标不是原问题的实际值(比实际值偏高). 第二批设计变量加入之后,CCGA-CFG 较快地收敛到较为优化的值. 可见变粒度策略提高了算法的收敛速度,而且优化效果好.

(3)与基于拟主元分析的遗传/粒子群算法(QPGP)的比较

与文献[17]中采用的基于拟主元分析的遗传/粒子群算法(QPGP)相比,变粒度协同进化算法得到的优化设计方案,在设计目标上得到了提升. CCGA-CFG 得到的设计目标的最优值和平均

表 2　CCGA-CFG 求解得到的最优布局方案

Tab. 2　The best layout scheme got by CCGA-CFG

序号	布置位置和方向			序号	布置位置和方向			序号	布置位置和方向	
	x/mm	y/mm	a_i/rad		x/mm	y/mm	a_i/rad		x/mm	y/mm
1	132.252	284.114	π/2	21	−200.148	18.919 9	0	41	−327.77	−46.521
2	−18.317 8	359.518	0	22	152.707	152.383	0	42	−202.29	360.303
3	−120.455	−326.417	0	23	2.192 38	186.731	π/2	43	160.664	261.903
4	−360.879	7.448 92	π/2	24	−360.384	21.523 1	π/2	44	360.65	−15.361
5	204.175	−165.553	0	25	−200.891	24.1158		45	86.009 7	−210.319
6	21.188 7	175.542	0	26	34.254 4	−197.265		46	−290.749	−109.100
7	225.461	−0.724 65	0	27	191.954	−56.306 5		47	−347.064	72.348 9
8	−181.593	−47.322 1	π/2	28	−67.850 6	188.141		48	−188.592	334.917
9	−13.747 7	−202.37	π/2	29	53.880 8	−381.822		49	−77.2743	−415.749
10	389.86	−161.922	π/2	30	−136.425	−205.573		50	−102.483	−172.748
11	−23.244 9	−365.404	0	31	−249.453	−139.811		51	108.874	−191.143
12	−26.111 1	224.875	π/2	32	95.547 5	128.697		52	−168.503	209.494
13	−99.281 6	−220.558	π/2	33	226.113	114.492		53	81.7116	390.347
14	−186.79	123.146	π/2	34	256.482	184.488		54	−285.948	−146.171
15	175.247	24.8101	π/2	35	−220.033	190.725		55	353.892	−150.710
16	258.122	−201.135	π/2	36	−205.397	−285.656		56	−235.529	−301.469
17	−235.097	−294.54	π/2	37	105.819	336.467		57	179.994	283.703
18	78.121 2	−371.404	0	38	−58.084 2	336.428		58	−8.05107	−314.854
19	320.878	171.011	π/2	39	148.458	−328.887		59	263.63	−267.062
20	201.31	−0.812 32	0	40	−354.525	80.802 6		60	−53.1265	336.836

值分别为为 714.60 和 718.70 kg·m^2，而 QPGP 的设计目标的最优值和平均值分别为 758.33 和 773.58 kg·m^2，分别提高了约 6%和 7%. 结果显示，QPGP 在优化搜索能力上与本文方法有一定差距，但 QPGP 的计算成功率较高.

综上，在求解本文算例的卫星舱布局问题时，(1)变粒度策略有效地缩短了计算耗时，并起到了提高求解质量的作用；(2)变粒度协同进化算法取得了 3 种算法中最好的计算结果，包括最优的设计目标值和最少的计算耗时，同时还表现出良好的鲁棒性(100 次试算，设计目标和计算时间的标准差最小)；(3)本文算法优化结果，较文献[17]中采用的 QPGP，求解质量有较大提高. CCGA-CFG 求得的最优布局方案如图 2 所示，该方案的性能指标如表 3 所示.

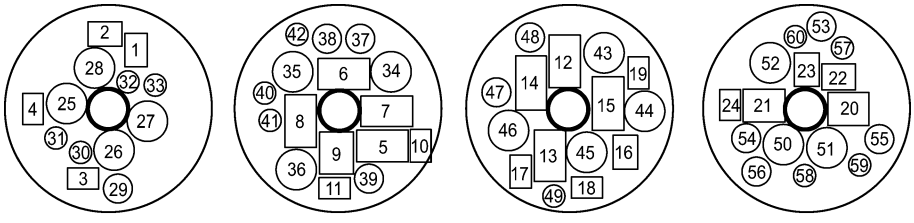

图 2　CCGA-CFG 最优布局方案二维示意图

Fig. 2　The 2-dimensional drawing of the best layout scheme got by CCGA-CFG

表 3　CCGA-CFG 最优布局方案的性能指标

Tab. 3　The performance index of the best layout scheme got by CCGA-CFG

转动惯量/(kg·m^2)			惯性夹角/rad			质心坐标/mm			干涉量/mm^3	耗时/s
x	y	z	x	y	z	x	y	z		
273.94	269.80	171.22	0	0	0.01	1.69	0.35	553.24	0	189.46

4　结　论

本文针对进化算法求解复杂卫星舱布局设计问题时存在的容易陷入局部最优、计算耗时长的问题，提出了一种变粒度合作式协同进化算法. 该方法基于协同进化算法框架，采用一种设计变量

的变粒度策略,以减少干涉量计算带来的计算消耗;同时通过"变粒度"降维达到降低问题的求解难度,提高算法对复杂问题搜索能力的目的.一个简化的卫星舱布局设计实例的数值仿真实验,验证了本文算法的有效性.

数值仿真实验结果还表明:本文的设计变量的变粒度策略不但有效缩短了CCGA的计算耗时,还提高了算法对复杂布局问题的求解质量.与CCGA、GA以及文献[17]的基于拟主元分析的遗传/粒子群算法(QPGP)相比,本文算法在求解的质量、效率和计算鲁棒性上均有所提高.

参考文献:

[1] DE BONT F M J, AARTS E H L, MEEHAN P, *et al*. Placement of shapeable blocks [J]. **Philips Journal of Research**, 1988, **43**(1):1-22

[2] 张 刚,殷国富,邓克文,等. 解空间编码遗传算法在三维布局中的应用[J]. 中国机械工程, 2006, **17**(1):79-83

[3] POTTER M A, DEJONG K A. A cooperative coevolutionary approach to function optimization [C] // **Proceedings of the Third Parallel Problem Solving from Nature**. Berlin:Springer-Verlag, 1994:249-257

[4] PANAIT L, LUKE S, WIEGAND R P. Biasing coevolutionary search for optimal multiagent behaviors [J]. **IEEE Transactions on Evolutionary Computation**, 2007, **10**(6):629-645

[5] DREZEWSKI R. A co-evolutionary multi-agent system for multi-modal function optimization [J]. **Lecture Notes in Computer Science**, 2004, **3038**:654-661

[6] 陈浩勇,王锡凡,别朝红. 协同进化算法及其在电力系统中的应用前景[J]. 电力系统自动化, 2003, **27**(23):94-100

[7] 刘 静,钟伟才,刘 芳,等. 组织协同进化分类算法用于雷达目标一维像识别[J]. 红外与毫米波学报, 2004, **23**(3):208-212

[8] JIN Y C. A comprehensive survey of fitness approximation in evolutionary computation [J]. **Soft Computing Journal**, 2005, **9**(1):3-12

[9] NAIN P K S, DEB K. A computaitonally effective multi-objective search and optimization technique using coarse-to-fine grain modeling [C] // **2002 PPSN Workshop on Evolutionary Multiobjective Optimization**. Spain:Springer, 2002:2081-2088

[10] ALADAHALLI C, CAGAN J, SHIMADA K. Objective function effect based pattern search-theoretical framework inspired by 3D component layout [J]. **Journal of Mechanical Design**, 2007, **129**(3):243-254

[11] WIEGAND R P. An analysis of cooperative coevolutionary algorithm [D]. Fairfax: George Mason University, 2003

[12] 刘 静,钟伟才,刘 芳,等. 组织协同进化分类算法[J]. 计算机学报, 2003, **26**(4):446-453

[13] POPOVICI E, DEJONG K A. The dynamics of the best individuals in co-evolution [J]. **Natural Computing**, 2006, **5**(3):229-255

[14] SOBIESZCZANSKI-SOBIESKI J, HAFKA R T. Multidisciplinary aerospace design optimization: Survey of recent development [J]. **Structural Optimization**, 1997, **14**(1):1-23

[15] LIU Z W, TENG H F. Human-algorithm-knowledge based layout design method and its application to satellite module [J]. **Computer and Industrial Engineering**, 2008, **55**(4):735-757

[16] YASSINE A, FALKENBURG D, CHELST K. Engineering design management: an information structure approach [J]. **International Journal of Production Research**, 1999, **37**(13):2957-2975

[17] ZHANG B, TENG H F, SHI Y J. Layout optimization of satellite module using soft computing method [J]. **Applied Soft Computing**, 2008, **8**(1):507-521

Coevolutionary algorithm with coarse-to-fine grain strategy and its application to layout design of satellite module

CHEN Yu, TENG Hong-fei*

(School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China)

Abstract: The main problems for evolutionary algorithm to deal with the complex layout design of satellite module include easily being trapped into local optimality, high complexity of interference computation and large amount of consuming time. To enhance the global search ability in the complex solution space, the layout design problem was decomposed and cooperative coevolutionary algorithm was employed for optimization search. Meanwhile, in order to reduce the computational time, a coarse-to-fine grain strategy of design variables is presented and applied to the co-evolutionary search of the sub-problems. The proposed algorithm was called cooperative coevolutionary algorithm with coarse-to-fine grain strategy (CCEA-CFG). The experimental results in a layout design of satellite module show that, the CCGA-CFG (CCEA-CFG based on GA) not only enhances the quality of solution, but also shortens the consuming time and is more robust, in comparison with genetic algorithm (GA), cooperative coevolutionary GA and QPGP.

Key words: coevolution; coarse-to-fine grain; layout design