

基于 APSO 算法的多目标满意优化研究

袁德国¹ 魏子淋² 成艳荣¹ 陆廷金³

(空军工程大学工程学院 西安 710038)¹ (空军驻京西地区军代表室 北京 100074)²
(徐州空军学院 徐州 221000)³

摘要 提出了多变量系统的多参数多目标满意优化方法。将系统性能指标要求的满意设计与控制器参数优化融为一体考虑,通过设计性能指标满意度函数和系统综合满意度函数,构造出多目标系统满意优化模型,并用自适应粒子群算法实现其满意优化。仿真结果显示该方法可获得比传统粒子群算法更满意的综合性能指标,表明了该方法的有效性和实用性。

关键词 自适应粒子群算法,多目标优化,满意度,满意优化,数字滤波器

中图分类号 TH122 **文献标识码** A

Multi-objective Satisfactory Optimization Based on the APSO

YUAN De-guo¹ WEI Zi-lin² CHENG Yan-rong¹ LU Ting-jin³

(Engineering Inst., Air Force Engineering Univ., Xi'an 710038, China)¹

(The Air Force Military Representative Office of Jingxi Region, Beijing 100074, China)² (Xuzhou Air Force College, Xuzhou 221000, China)³

Abstract This paper proposed a satisfactory optimization method for the multivariable control system with multi-parameters and multi-objectives. By designing satisfactory rate functions of performance criterions and comprehensive satisfactory rate functions of the system, satisfactory optimization model of multi-objectives control system was constructed, in which satisfactory design of performance criterions and optimization of controller parameters were considered together. Then, the satisfactory optimization design procedure was given in detail by using APSO. Simulation results show that better performances can be obtained with satisfactory optimization method than those with traditional PSO, indicating that the method introduced is effective and practical.

Keywords APSO, Multi-objective optimization, Satisfactory degree, Satisfactory optimization, Digital filter

在实际工程中,人们多数情况下遇到的是多目标优化问题,对多目标优化问题进行深入分析与研究具有重要意义。传统的解决多目标优化命题的方法如牛顿法、单纯形法、拉格朗日乘子法、共扼梯度法等等,试图在求解优化问题的过程中找到问题的最优解。但是在实用中,重要的还是求有效解和弱有效解。满意优化借助几何直观构造评价函数,将多目标优化问题转化成为单目标优化问题,在满足有效解或者弱有效解的条件下,将这个解作为多目标优化问题的解,使相互矛盾的各个目标得以兼顾。

本文将自适应粒子群算法(APSO)结合到多目标满意优化中,提出一种新的多目标优化方法——MAPSO,然后把MAPSO方法应用到FIR数字滤波器的优化设计中,使多个相互矛盾的滤波器性能指标在理想的最优化目标值的某个范围做折中考虑,实现了数字滤波器参数的整体优化。

1 满意优化方法

满意优化(Satisfactory Optimization)是相对于最优化方法提出的、在人类智能准则的基础上形成的、以满意解(Satis-

fied Solution)输出为原则的一种优化方法。

1.1 满意解原则与满意优化基本流程

通过分析人类的智能活动,发现输入信息的模糊性以及输出信息的满意性是其主要的特征。人类高级智能活动的特点不再追求凡事得到最优解,而是实时地、能随机应变地求得可以解决所面临问题的满意解。正是这一条看起来十分平凡的满意解原则,却使得人类在各种智能信息处理的活动中可以极大地节省信息处理的时间以及信息存储的空间,从而提高效率。

通过对满意解原则的分析,图1给出满意优化方法的基本流程图。



图1 满意优化方法流程

1.2 满意解、满意度函数的定义

满意度和满意解一直是满意理论中研究最多、应用最广的概念。先后出现过多种满意度、满意解的定义,下面介绍几

到稿日期:2008-06-02

袁德国(1980-),男,博士研究生,研究方向为装备系统工程与智能决策,E-mail: yuandeguo@tom.com;魏子淋(1980-),男,工程师,研究方向为航空弹药;成艳荣(1980-),女,博士研究生,研究方向为装备系统工程与智能决策;陆廷金(1947-),男,教授,博士生导师,研究方向为系统工程。

种主要的定义形式。

1.2.1 基于模糊数学的定义

暨南大学的任平教授利用模糊数学的方法对满意度进行了定义。考虑论域 U 到目标值集合 V 的一个映射

$$f: U \rightarrow V$$

U 为全体可能解或可能策略的集合, V 是衡量解或策略的某种标准集。定义 V 的一个 Fuzzy 子集 \tilde{G} 为“满意”, 其隶属函数为:

$$\mu_{\tilde{G}}: V \rightarrow [0, 1]$$

令 \tilde{C} 为 U 的 Fuzzy 子集, 有

引用定义 1 对满足 $C_\lambda \neq \Phi, G_\lambda \neq \Phi$ 的任一 λ 值, $0 \leq \lambda \leq 1, M_\lambda = \{u | u \in C_\lambda, f(u) \in G_\lambda\}$ 为 f 在 \tilde{C} 上的 λ 水平的满意解集。其中, C_λ, G_λ 分别为 \tilde{C}, \tilde{G} 的 λ 水平集, 即

$$C_\lambda = \{u | u \in U, \mu_{\tilde{C}}(u) \geq \lambda\}, G_\lambda = \{v | v \in V, \mu_{\tilde{G}}(v) \geq \lambda\}$$

引用定义 2 考虑 \tilde{C} 的 Fuzzy 子集 \tilde{M} , 其隶属函数规定为:

$$\mu_{\tilde{M}}(u) = \sup_{\lambda \in M_\lambda} \lambda, 0 \leq \lambda \leq 1$$

称 \tilde{M} 为 f 在 \tilde{C} 上的满意解。 $\mu_{\tilde{C}}(u)$ 就是解 u 的满意度。

1.2.2 基于线性取值的满意度的定义

对于线性取值的满意度定义较多, 比较早的是胡思继教授, 他在探讨铁路运输计划指令性指标按纵向从上到下或从下到上进行分解的方法时, 给出了如下的定义。

引用定义 3 设 f_i 为指标数值, f_i^* 与 $f_i^{\#}$ 分别为指标的最佳解和最差解, 则指标 f_i 的满意度为

$$\alpha_i = \frac{f_i - f_i^{\#}}{f_i^* - f_i^{\#}}$$

1.2.3 基于模糊神经计算的定义

靳蕃教授从模糊神经计算智能系统输出解的一般性评定满意度的需要出发, 按照解的离散和连续性质、实时和非实时要求, 给出了几种不同的满意度表示方法。

引用定义 4 设 $\omega(q)$ 表示定义为 $q_a \leq q \leq q_b$ 的解概率密度函数, 则当获得质量为 q_i 的解 x_i 时, 满意度为

$$S(x_i) = 1 - \int_{q_{x_i}}^{q_b} \omega(q) dq$$

此满意度与时间无关, 称为静态满意度。而下面定义的用来刻画神经网络在一定时间内输出解的满意程度, 称为动态满意度。

引用定义 5 设 FNCIS 在 $t_0 \geq 0$ 时刻输出可供选择的解数为 $N(t_0)$, 在单位时间 Δt 后获得可供选择的解数为 $N(t_0 + \Delta t)$, 则该系统在 t_0 时刻输出解的动态满意度为

$$S_d(t_0) = \frac{N(t_0)}{N(t_0 + \Delta t)}$$

这几个满意度的定义反映了智能信息系统在解空间中搜索时的满意停止条件或准则, 即当搜索过程已几乎找不到解时就应该停止搜索了, 此时可认为已经找到整个解空间的满意解了。从其在 TSP 中的应用可以知道, 其满意度定义是合理可行的。

1.3 满意度函数的形式

满意度是衡量人们在某个标准下对其选择(或行为)满意程度的一种度量。可以说任何能表示解的质量的量都可用来表示满意度, 但是为了满足科研和工程计算的需要, 对一些常用的、易操作的、比较合理的表示形式进行量化的描述和研究

非常必要。

根据满意度的合成原则, 对于一般的复杂问题, 虽然有许多因素对满意度产生影响, 但我们总可以从某些简单方面开始分析, 进而可以通过一维满意度的某些合成运算来获取多维满意度。

考虑到满意度函数表示形式存在的单调性, 同时为了表达形式的一致性, 式中的满意度取值范围均为 $[0, 1]$, 在实际应用中, 满意度范围可以是用户自己定义的某个合理值。表 1 以单调减函数(戒上型)为例, 给出几种一维满意度函数的表示形式。

表 1 满意度函数的几种表示形式

降半矩形函数	降半正态函数
$s(d) = \begin{cases} 1, & d \leq a \\ 0, & d \geq a \end{cases}$	$s(d) = \begin{cases} 1, & d \leq a \\ e^{-k(d-a)^2}, & d > a \end{cases} \quad (k > 0)$
降半折线函数	降半 Γ -函数
$s(d) = \begin{cases} 1, & d \leq a_1 \\ \frac{(1-b)(a_1-d)}{a_2-a_1} + 1, & a_1 < d \leq a_2 \\ \frac{b(a_3-d)}{a_3-a_2}, & a_2 < d \leq a_3 \\ 0, & d > a_3 \end{cases}$	$s(d) = \begin{cases} 1, & d \leq a \\ e^{-k(d-a)}, & d > a \end{cases} \quad (k > 0)$

2 多目标满意优化模型

2.1 多目标满意优化问题的提出

可以将多目标最优化问题表示为:

$$\begin{aligned} & \min(f(x_1), f(x_2), \dots, f(x_m))^T, m > 1, x \in E^m \\ & s. t. \quad g_i(x) = 0, i = 1, 2, \dots, k \\ & \quad \quad h_j(x) = 0, j = 1, 2, \dots, k \end{aligned} \quad (1)$$

在实际多目标优化问题中, 要想在错综复杂的矛盾关系之间找到一个兼顾各个优化目标的绝对最优解是几乎不可能实现的。

通过本文前半部分对传统最优化与满意优化方法进行比较, 可以发现: 在多目标优化问题中, 同样存在求满意解的情况。在传统的多目标优化问题中, 有着各种意义下的最优解。但是在工程应用中, 重要的还是求有效解和弱有效解。而求有效解和弱有效解的最基本的方法是利用“评价函数法”。其基本思想是: 借助几何或是应用中的直观背景, 构造评价函数, 从而将多目标优化问题转化为单目标优化问题; 然后利用单目标优化问题的求解方法进行求解, 在满足有效解或者弱有效解的条件下, 将这个解作为多目标优化问题的解。而在这种意义下, 所求得解可以作为多目标优化问题的一种满意解。有 3 方面的因素造成了多目标优化问题的“满意性”:

(1) 由于各个目标下最优解之间的矛盾难以兼顾而无法获得绝对最优解。

(2) 对解的性能评价的满意度评价, 或者目标函数以及约束条件的模糊性。

(3) 优化计算结果的满意输出。

综上所述, 在处理多目标优化问题时, 应参照人类智能信息处理方式的特点, 以满意解输出原理为依据, 采用多目标满意优化方法进行求解。

2.2 多目标满意优化问题的描述

分析式(1), 可以将多目标优化问题 PX 描述为 $[x, Q, S]$, 其中 x 为系统的参数(变量): $x = [x_1, x_2, \dots, x_n]$ 。 Q 为系统的性能指标集: $Q = [q_1, q_2, \dots, q_n]$ 。 x, Q 的维数可以视为优化问题的复杂因素的状态。维数越高, 对应的优化问题

就越复杂。这可以将复杂的因素的状态空间分解成为若干简单的因素的状态空间,然后再利用对这些简单因素的状态空间的求解来合成复杂优化问题的解。这样就可以将系统的性能指标集 Q 分解为一个个单独的性能指标 q_i ,再利利用前面对于满意函数的设计方法,得到相应的性能指标的满意度函数 s_i 。 S 为描述系统各个性能指标 q_i 的优化结果的满意度函数集, $S=[s_1, s_2, \dots, s_m]$ 。对于任何 x ,只要确定了 $s_i(x)$, ($i=1, 2, \dots, m$),便得到了:

$$S(x)=[s_1(x), s_2(x), \dots, s_m(x)] \quad (2)$$

这时的 S 是一个 m 维的向量,它并不能直接作为此优化问题的评价标准。为了达到既不损失信息又便于进一步处理的目的,我们通过“降维”的方法对其进行变换。所谓“降维”就是将高维的状态空间 S 降为低维的状态空间,从而降低问题的复杂程度。这也就是我们解决这一类多目标优化问题时的基本思想。通过降维映射 $f: s\omega=f(S(x))=f(s_1(x), s_2(x), \dots, s_m(x))$,实现把 m 维的点(各性能指标的满意度函数值)综合为一维的点(系统的整体满意度函数值),这其中的关键在于“综合”,因此我们称映射 f 为“综合函数”。

综合函数常取如下形式:

$$f(S)=f(s_1, s_2, \dots, s_m)=\sum_{i=1}^m \omega_i s_i \quad (3)$$

其中 $\omega_i \in [0, 1]$, ($i=1, 2, \dots, m$),并且满足 $\sum_{i=1}^m \omega_i=1$ 。式(3)通常称为线性加权综合满意度函数,它反映出了综合各个性能指标的满意度后的多目标优化问题的综合满意度。从式(3)可以发现,在多目标满意优化方法中,如何将各个评价性能指标的满意度 s_k ($k=1, 2, \dots, m$)通过综合满意度函数 f 综合成反映整体优化效果的综合满意度,从而求得问题的满意解,是多目标满意优化方法的一个重要问题。

2.3 多目标满意优化模型

设优化问题 PX 的系统参数(变量)有 n 个,记为 x_1, x_2, \dots, x_n ,构成可供优化选择的参数集:

$$X=\{[x_1, x_2, \dots, x_n] | x_i \in R, i=1, 2, \dots, n\} \quad (4)$$

优化问题的解变量可以表示为 $x=[x_1, x_2, \dots, x_n] \in X$, $X \subseteq R$ 。 X 即为优化问题的解空间。其中一组参数 $x^{(j)}=[x_1^{(j)}, x_2^{(j)}, \dots, x_n^{(j)}]$,称为该参数优化问题的一个解。

设评价优化问题性能指标为 m 个,构成性能指标集 Q :

$$Q=\{(q_1, q_2, \dots, q_m) | q_i \in R, i=1, 2, \dots, m\} \quad (5)$$

对于某一个参数解 $x^{(j)}$,可以得到其性能指标 $q^{(j)}=[q_1^{(j)}, q_2^{(j)}, \dots, q_m^{(j)}]$,其中 $q_k^{(j)}$ 是每一性能指标 q_k 关于参数解 $x^{(j)}$ 的取值, $k=1, 2, \dots, m$ 。而 q_k 的取值一般和多个参数变量有关, $\varphi_k(\cdot)$ 给出了 q 关于 x 的函数关系,表示为

$$q_k=\varphi_k(x)=[\varphi_{k1}(x), \varphi_{k2}(x), \dots, \varphi_{km}(x)], k=1, 2, \dots, m \quad (6)$$

对于每一个性能指标 q_k , $k=1, 2, \dots, m$,既可以是系统的优化目标;又可以包括对系统性能指标的约束条件。 q_k 可以分别给出满意度函数 g_k ;

$$g_k: R \rightarrow [0, 1] \\ s_k=g_k(q_k), k=1, 2, \dots, m \quad (7)$$

在优化问题中, s_k 以函数值的形式反映出决策者对该项性能指标 q_k 的要求。由 s_k 构成满意度函数向量 $S=[s_1, s_2, \dots, s_m]$, $s \in [0, 1]^m$,以向量函数 $g(\cdot)$ 表示性能指标向量 q 的满意度函数,可以得到 $S=g(q)=[g_1(q_1), g_2(q_2), \dots, g_m$

$(q_m)]$ 。

综合满意度函数表示为: $f: [0, 1]^m \rightarrow [0, 1]$, 记:

$$s\omega=f(s)=f(s_1, s_2, \dots, s_m), s\omega \in [0, 1] \quad (8)$$

称为标量函数,反映出决策者对性能指标的综合评价。

由以上各式,可以得到多目标满意优化问题计算模型

$$\begin{cases} \max s\omega \\ s\omega=f(s) \\ s=g(q) \\ q=\varphi(x) \\ x \in X \subseteq R^n, q \subseteq Q \subseteq R^m \\ s \in [0, 1]^m, s\omega \in [0, 1] \end{cases} \quad (9)$$

在这个寻优计算模型中,通过对性能指标的满意度评价函数的寻优,实现系统参数的优化。系统的性能指标向量 q 反映出系统的优化期望,评价性能指标的满意度函数 g 和综合满意度函数 f 的设计,一方面反映出决策者对优化问题的意图和理解,另一方面直接影响满意优化模型的寻优结果。

3 自适应粒子群算法与多目标满意优化结合的 MAPSO 方法

PSO(Particle Swarm Optimization)算法容易实现,代码量小,从 1995 年开始,已经获得很大的发展,并已经在一些领域得到应用。它的缺点是易陷入局部极点、搜索精度不高,很多学者从事研究和改造 PSO 算法的工作。在一定的简化假设条件下,将粒子群中一个粒子的运动轨迹独立出来分析,获得了在不同参数分布情况下算法不同的稳定性。在此基础上,本文提出一种改进的自适应粒子群算法,引入“优胜劣汰”的自然发展规律,克服了原 PSO 模型易陷入局部极点的缺点。

借鉴模拟退火算法机制,在粒子群算法的初期,对整个粒子群的个体极值 $pBest$ 做评估,计算出个体平均极值 \overline{pBest} ,把个体极值 $pBest$ 劣于平均极值 \overline{pBest} 的粒子执行变异操作,其余粒子采用一个较小惯性权值,与较大的跳出局部极值的概率相对应;而在末期,整个群体采用一个较大的惯性权值,与较大的系统收敛、稳定的概率相对应。这就是本方法的自适应变异机制。自适应粒子群算法(APSO)可以表述为

$$v_{id}^{k+1}=\omega * v_{id}^k + \phi_1 * (p_{id}^k - x_{id}^k) + \phi_2 * (p_{gd}^k - x_{id}^k) \quad (10)$$

$$x_{id}^{k+1}=x_{id}^k + v_{id}^{k+1} \quad (11)$$

$$\phi_j=b_j * r_j + d_j, j=1, 2 \quad (12)$$

$$\overline{pBest}=\frac{1}{D} \sum_{i=1}^D P_{id} \quad (13)$$

其中, b_j 是值为 1.5 的常数, r_j 是在 (0, 1) 区间的均匀分布的随机数, d_j 是值为 0.5 的常数。 b_j 和 d_j 的共同作用使 ϕ_j 取值在 (0.5, 2) 的区间中。 ω 是一个线性递增的惯性权值,比如从 0.3 增加到 1.1。而 \overline{pBest} 是个体平均极值,作为是否进行变异操作的判断标准。

自适应粒子群算法的具体算法流程如下:

选定 APSO 种群规模 m ;

设 $x[i]$ 为种群中第 i 个粒子的位置;

设 $fitness[i]$ 为第 i 个粒子的适应值;

设 $v[i]$ 为第 i 个粒子的速度;

设 $gBest$ 为种群最好粒子的标号;

设 $Pbest[i]$ 为第 i 个粒子自身搜索到的最好点位置;

设 $Pbest_fitness[i]$ 为第 i 个粒子自身搜索到的最好适应值, 即 $Pbest[i]$ 处的适应值;

设 $pBest$ 为整体粒子搜索到的最好点位置的平均值;

Step1 (初始化) 对于每一个种群中的粒子 $i, i=1, 2, \dots, m$;

Step 1.1 随机初始化 $x[i]$;

Step 1.2 随机初始化 $v[i]$;

Step 1.3 计算 $fitness[i]$, 并以次初始化 $Pbest_fitness[i]$;

Step 1.4 以种群中最好适应值的粒子初始化 $gBest$;

Step 1.5 以 $x[i]$ 初始化 $Pbest[i]$;

Step 1.6 计算 $pBest$;

Step 2 循环迭代, 直到满足 APSO 终止条件;

Step 2.1 计算当前惯性权值 ω ;

Step 2.2 对每个粒子, 计算其适应值 $fitness[i]$, 若 $fitness[i] < Pbest_fitness[i]$, 则 $Pbest_fitness[i] = fitness[i]$, 且 $Pbest[i] = x[i]$;

Step 2.3 搜索 $gBest$, 若 $Pbest_fitness[i] < Pbest_fitness[gBest]$, 则 $gBest = i$;

Step 2.4 当循环迭代次数少于一定值时, 判断每个粒子是否需要变异。若 $Pbest[i] < \overline{pBest}$, 该粒子则变异操作, 重生 $x[i]$ 和 $v[i]$; 若 $Pbest[i] \geq \overline{pBest}$, 该粒子依据 (10)、(11) 和 (12) 更新 $x[i]$ 和 $v[i]$;

Step 2.5 当循环迭代次数大于一定值时, 对每个粒子, 依据 (10)、(11) 和 (12) 更新 $x[i]$ 和 $v[i]$ 。

结合 APSO 算法与多目标满意度优化, 借助应用中的直观背景, 构造满意度函数, 通过综合满意度函数, 从而将多目标优化问题转化为单目标优化问题。然后利用 APSO 算法, 对此单目标问题求解。在满足有效解的条件下, 将这个解作为多目标优化问题的解。我们称这种多目标优化为 MAPSO 方法。MAPSO 方法综合了多目标满意度优化和 APSO 算法的优点, 既使多个相互矛盾的优化目标、性能指标或者约束条件在理想的最优化目标值的某个范围得以兼顾, 又能避免寻优过程中陷入局部极值。

4 MAPSO 在 FIR 数字滤波器设计中的应用

FIR 数字滤波器的设计任务, 是要设计一个传递函数 $H(z)$, 使它的频率响应满足给定的要求。这些要求包括通带频率、阻带频率、带内的最大衰减、通带波动以及保证 $H(z)$ 具有线性行为等等。

频率抽样技术是最常用的数字滤波器设计方法。对理想滤波器 $H_d(e^{j\omega})$ 在 0 到 2π 上的 N 个等间隔频率上抽样, 对样本 $H(k)$ 内插, 得到实际响应 $H(e^{j\omega})$ 。通过这样设计所得的滤波器在通带内有较大的上冲, 在阻带内也有较大的波纹。只有改进频率采样设计的方法, 提高逼近的质量, 使得逼近的误差更小。

设 FIR 数字滤波器的频率响应是 $H_d(e^{j\omega})$, 它是频率 ω 的周期函数, 对它进行抽样, 使得每个周期有 N 个抽样值。

$$\begin{aligned} H_d(k) &= H_d(e^{j\omega}) \Big|_{\omega_k = \frac{2\pi k}{N}} = H_d(e^{j\frac{2\pi k}{N}}) \\ &= H_r(2\pi k/N) e^{j\phi(k)} \end{aligned} \quad (14)$$

其中 $H_r(2\pi k/N)$ 为滤波器的增益, $\phi(k)$ 为其滤波器的相位响应, 具体表示为:

$$H_r(2\pi k/N) = \begin{cases} H_r(0), & k=0 \\ H_r[2\pi(N-k)/N], & k=1, 2, \dots, N-1 \end{cases} \quad (15)$$

$$\phi(k) = \begin{cases} -(\frac{N-1}{2})(\frac{2\pi k}{N}), & k=0, \dots, [\frac{N-1}{2}] \\ (\frac{N-1}{2})\frac{2\pi}{N}(N-k), & k=[\frac{N-1}{2}]+1, \dots, N-1 \end{cases} \quad (16)$$

本文通过选择最佳的频率插值来改善滤波器的频率特性。过渡带上的插值不同, 效果也将不同。在频率抽样法设计中利用 MAPSO 方法来进行优化设计, 可以较容易地获得令人满意的效果, 其基本步骤如下。

(1) 选择优化参数: 过渡带的抽样点。

(2) 选择性能指标: 通带上冲的最大值、过渡带大小、阻带波纹峰值等。

(3) 根据性能指标设计各性能指标的满意度函数。

(4) 设计综合满意度函数。

(5) 以综合满意度函数作为 APSO 算法的目标函数, 进行 APSO 算法寻优。

下面通过仿真实例对 MAPSO 方法进行说明。

给定一个低通滤波器的技术指标: 通带截止频率 $\omega_c = 0.3\pi$, 抽样点数 $N=60$ 。根据设计的要求, 该数字滤波器理想的频率特性可表示为

$$|H_d(e^{j\omega})| = \begin{cases} 1, & 0 \leq \omega \leq \omega_c \\ 0, & \text{其它} \end{cases} \quad (17)$$

为了提高实际滤波器的逼近质量, 分别在 $n=8, 9$ 和 $n=53, 54$ 处插入过渡带抽样值 T_1 和 T_2 , $0 < T_1 < 1, 0 < T_2 < 1$ 。由式 (15) 得抽样幅度响应如下:

$$H_r(k) = [1, 1, 1, 1, 1, 1, 1, T_1, T_2, 0, \dots, 0, T_2, T_1, 1, 1, 1, 1, 1, 1] \quad (18)$$

MAPSO 方法对数字滤波器进行优化, 具体步骤如下:

Step 1 选择优化参数。以过渡带抽样值 T_1 和 T_2 作为优化参数, 并分别设置初值范围 $[0.5, 0.9]$ 和 $[0.01, 0.4]$ 。

Step 2 根据设计要求选择性能指标, 并设计相应的满意度函数。该 FIR 数字滤波器的性能指标如下: 通带的上冲最大值 R_p 、过渡带的大小 $belt$ 以及阻带最小衰减 A_s 。根据以上性能指标, 选择折线法满意度函数形式, 设计各性能指标满意度函数, 分别为 $[s_1, s_2, \dots, s_n] = [g_1(R_p), g_2(belt), g_n(A_s)]$, 其具体函数图如图 2 所示。

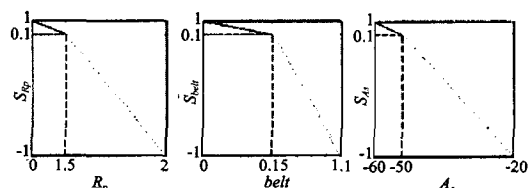


图 2 性能指标的满意度函数

Step 3 设计综合满意度函数。本例的综合满意度函数选择线性加权方式:

$$s_{\omega} = f(s) = \sum_{k=1}^m \omega_k s_k / \sum_{k=1}^m \omega_k \quad (19)$$

其中 ω_k 为关于 s_k 的权值, 用于反映设计者对性能指标 R_p , $belt$, A_s 的重视程度。为了考虑对增大阻带最小衰减 A_s 的重视, 尽量减少通带上冲和过渡带的宽度 $belt$, 取各权值系数为

$$[\omega_1, \omega_2, \omega_3] = [1, 1, 3]$$

Step 4 利用 APSO 算法优化计算。粒子种群数为 10, 进化代数为 200, 得到两个过渡带抽样值分别为 $T_1 = 0.59428, T_2 = 0.10936$, 它们对应的幅频特性曲线如图 3 所示。

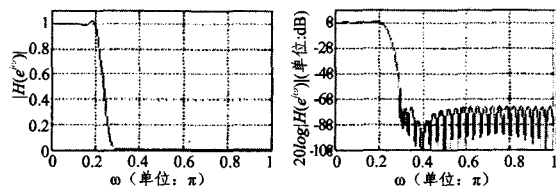


图 3 幅频特性曲线和对数幅频特性曲线

表 2 是分别采用 MAPSO 方法与 APSO 算法进行滤波器参数优化的性能值结果比较表。比较两种结果可以发现, 采用 MAPSO 方法得到的满意优化解与采用 APSO 算法得到的最优解十分接近。由于 APSO 算法只侧重阻带最小衰减 A_s 方面的寻优, 它所设计的滤波器在这方面具有的最优值, 减少了阻带纹波峰值, 比起采用 MAPSO 方法优越。但 MAPSO 方法基于多目标满意优化的思想, 不仅考虑了阻带最小衰减 A_s , 也考虑了通带上冲最大值 R_p 和过渡带 bel_t 大小, 设计所得的滤波器的通带上冲最大值较小, 过渡带也较小, 从而减少了频带边缘两采样点间的突变, 也减少了起伏振荡。

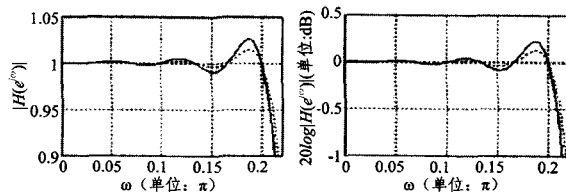
表 2 采用 MAPSO 方法和 APSO 算法优化的滤波器性能指标值比较

	进化代数	T_1	T_2	R_p (dB)	bel_t	A_s (dB)
MAPSO	200	0.59428	0.10936	0.2001	0.05500	66.7291
APSO	200	0.59433	0.10940	0.2225	0.5981	66.8495

另一方面, MAPSO 方法在设计滤波器时具有很强的灵活性。因为不同的权值系数对应不同的满意度函数, 各个权值系数的大小分别体现出设计者对各项性能指标的重视程度。因此, 设计者可以通过调节线性权值来满足对优化指标的不同要求。

继续以上述 FIR 低通滤波器为例, 来说明 MAPSO 方法的灵活性。若我们要求最大可能减少阻带纹波峰值的同时, 尽量减少通带上冲, 而过渡带的大小作次要考虑。因此, 我们对线性加权重新设计, 取为 $[3, 1, 3]$ 。利用 MAPSO 方法设计的 FIR 低通滤波器的两个过渡带抽样值分别为 $T_1 = 0.59428, T_2 = 0.10936$, 在通带处它们对应的幅频特性曲线如

图 4 所示。



(实线是线性加权调整前的结果; 虚线是线性加权调整后的结果)

图 4 通带部分的幅频特性曲线和对数幅频特性曲线

结束语 通过对多目标满意优化问题的分析, 介绍了多目标满意优化模型。考虑到自适应粒子群算法的寻优能力, 将自适应粒子群算法与多目标满意优化方法相结合, 提出一种新的针对多目标优化问题的求解方法——MAPSO 方法, 并应用在 FIR 滤波器参数优化设计中。最后通过仿真实验, 与 APSO 算法相比较, 验证了 MAPSO 方法的可行性、有效性和灵活性。

参考文献

(上接第 192 页)

[6] Yu X, Song Y. Chaos synchronization via controlling partial state of chaotic systems [J]. Int. J. Bifur. Chaos, 2001, 11(6): 1737-1741

[7] Shahverdiev E M, Sivaprakasam S, Shore K A. Lag synchronization in time-delayed systems [J]. Phys. Lett. A, 2002, 292(6): 320-324

[8] Mainieri R, Rehacek J. Projective synchronization in three-dimensional chaotic systems [J]. Phys. Rev. Lett., 1999, 82(15): 3042-3045

[9] Kim C M, Rim S H, Key W. Anti-synchronization of chaotic oscillators [J]. Phys. Lett. A, 2003, 320(1): 39-46

[1] 靳蕃. 神经计算智能基础原理与方法[M]. 成都: 西南交通大学出版社, 2000

[2] Yao X S, Huang H Z, Zhou Z R. Study on multi-objective optimization based on generalized satisfactory degree theory for transportation capability of railway-network[A]// Proceedings of ICTTS[C]. 2002; 947-952

[3] 江坤, 钱林方, 徐亚栋. 某炮口制退器的多目标优化设计[J]. 系统仿真学报, 2007, 19(15): 3478-3480

[4] 张葛祥, 金炜东, 胡来招. 多变量系统中多参数多目标满意优化研究[J]. 信息与控制, 2003, 32(6): 484-485

[5] Li Yin - ya, Sheng An - dong, Wang Yuan - gang. Satisfactory optimization of multi-objective PID controllers based on particle swarm optimization algorithm[A]// Proceedings of 2005 Chinese Control and Decision Conference[C]. 2005; 906-910

[6] 黄婉平. 自适应粒子群优化算法及其应用研究[D]. 杭州: 浙江大学, 2006; 21-32

[7] Coello C A C, Lechunga M S. MOPSO: A proposal for multiple objective particle swarm optimization[A]// Proceedings of the 2002 Congress on Evolutionary Computation, part of the 2002 IEEE World Congress on Computational Intelligence[C]. Hawaii, 2002; 1051-1056

[10] Xu D L, Li Z. Controlled projective synchronization in non-partially-linear chaotic systems [J]. Int. J. Bifur. Chaos, 2002, 12(6): 1395-1402

[11] Yan J P, Li C P. Generalized projective synchronization of chaos: the cascade synchronization approach [J]. Chaos, Solitons Fract., 2005, 30(1): 140-146

[12] Li G H, Zhou S P, Yang K. Generalized projective synchronization between two different chaotic systems using active backstepping control [J]. Phys. Lett. A, 2006, 355(4/5): 326-330

[13] Rössler O E. An equation for continuous chaos [J]. Phys. Lett. A, 1996, 57(5): 397-400