

带约束多目标最优化问题的一种新的进化算法^{*}

A Novel Evolutionary Algorithm for Constrained Multiobjective Optimization Problems

刘海林^{1,2} 王宇平³ 刘永清²(广东工业大学应用数学系 广州510090)¹ (华南理工大学自动控制工程系 广州510640)²(西安电子科技大学数学科学系 西安710071)³

Abstract First, the constrained multiobjective optimization problem is transformed into an unconstrained multiobjective optimization problem by normalizing objectives and becoming constraints into additional objectives. As a result, the problem is simplified. Second, A new fitness function is defined by maximization of product of each objective of unconstrained multiobjective optimization problem with a proper weight. The way defining weights is much different from the general ways. Not only the weights defined are not limited between zero and one, but also their sum is not required to be one. The weights are only required to be positive and their product is one. Furthermore, the weights defined in this way can guide the search more effectively and can find uniformly distributed Pareto optimal solutions more easily. Based on these and combined with uniform design method, a novel evolutionary algorithm is proposed and the computer simulation is made, the results show that the proposed algorithm is effective.

Keywords Constrained optimization, Multiobjective programming, Evolutionary algorithms, Uniform design

1 引言

多目标最优化问题在科学技术、经济管理等领域大量存在,进化算法一次运算可望求出多目标最优化的许多有效解的特性,引起众多学者的研究兴趣。目前,仅在网址:www.lania.mx/~ccoello/emoo/emooibib.html上就可查到近千篇多目标进化算法方面的文献。最近几年出现的多目标进化算法,绝大多数都是研究无约束多目标最优化问题(例如见文[1,2]),而实际问题中遇到的多目标最优化往往都带有约束条件,因而处理约束就是解决现实问题的关键。正如Kalyanmoy Deb等学者在文[3]中指出:“研究者是该把注意力集中在解决约束多目标最优化问题的时候了”。

目前针对处理带约束的多目标进化算法的论文仅有少数几篇(见文[4~7])。Deb K和Srinivas N^[4,5]提出的方法是首先把所有约束和目标函数规范化,然后用罚函数的思想给各个目标函数加上比例于约束函数的惩罚项。Jimenez等在文[6]中,使用了二进制锦标赛选择,对被选择的两个解仔细进行比较,如果一个解是容许解,另一个是非容许解则选择容许解;如果两个解都是容许解,则使用小生竞技术;若两个解都是非容许解,该方法可使得距边界近的解被选择的可能性更大。

最近,Deb K等在文[7]中定义了一种约束优势原理(constraint-domination principle),该方法在处理非劣解时把容许解和非容许解进行区分。对两个解*i*和*j*,如果下列条件之一成立:

1. *i*是容许解,*j*是非容许解;
2. 解*i*和*j*都是非容许的,而解*i*比解*j*违反约束总体上程度小;
3. *i*和*j*都是容许解,而解*i*优势于解*j*;

则称解*i*约束优势于解*j*。

本章研究一般形式的约束多目标最优化问题

$$\begin{cases} \max_{x \in X} (f_1(x), f_2(x), \dots, f_m(x)) \\ s. t. \quad h_i(x) \geq 0 \quad i=1, 2, \dots, l \\ \quad \quad h_i(x) = 0 \quad i=l+1, l+2, \dots, q \end{cases} \quad (1)$$

其中, X 为 R^n 空间上的 n 维矩形域,集合

$$F = \{x | x \in X; h_i(x) \geq 0, i=1, 2, \dots, l; h_i(x) = 0, i=l+1, l+2, \dots, q\}$$

称为容许域。

定义1 设 $\bar{x} \in F$,若不 $\exists x \in F$ 使得对 $\forall i$ 有 $f_i(x) \leq f_i(\bar{x})$ 且 $\exists i_0$ 使 $f_{i_0}(x) < f_{i_0}(\bar{x})$,则称 $\bar{x}$ 为问题(1)的有效解,所有有效解的集合称为有效解集,记为 $E(f, X)$,称集合 $\{f(x); x \in E(f, X)\}$ 为问题(1)的有效界面。

定义2 设 $\bar{x} \in F$,若不 $\exists x \in F$ 使得对 $\forall i$ 有 $f_i(x) < f_i(\bar{x})$,则称 $\bar{x}$ 为问题(1)的弱有效解,所有弱有效解的集合称为弱有效解集,记为 $E_w(f, X)$,称 $\{f(x); x \in E_w(f, X)\}$ 为问题(1)的弱有效界面。

对问题(1),首先把等式约束用不等式约束代换,将约束函数作为目标函数,把约束多目标最优化转化为无约束多目标规划问题。其次,使用类似第四章中的方法定义一个同进化代数有关的变适应值函数,这样定义的适应值函数能使种群随着进化代数的增加所含的容许解也越来越多,并更易于求出均匀分布的Pareto最优解。另外,在种群中设计了一个使目标函数为非劣解的集合,该集合中的解也能随着进化代数的增加所含的容许解也越来越多。在此基础上,结合均匀设计构造了一个新的进化算法并进行了数值模拟,结果表明该方法十分有效。

2 处理约束的方法

因为问题(1)式中的等式约束

^{*} 国家自然科学基金重点基金资助(编号:69934030),陕西省自然科学基金资助(编号:2001S106),教育部留学回国人员科研启动基金资助。刘海林 博士生,副教授,研究方向:进化计算,最优化理论与算法。王宇平 博士,教授,博导,主要从事进化计算和最优化理论与算法研究。刘永清 教授,博导,研究方向为复杂化系统和系统工程。

$$h_i(x) = 0 \Leftrightarrow -h_i^2(x) \geq 0, i = l+1, l+2, \dots, q$$

这样约束多目标最优化(1)式等价与如下问题:

$$\begin{cases} \max_{x \in X} (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s. t. } h_i(x) \geq 0, i = 1, 2, \dots, l \\ -h_i^2(x) \geq 0, i = l+1, l+2, \dots, q \end{cases} \quad (2)$$

将(2)式中的 q 个约束函数 $h_i(x) (i=1, 2, \dots, l)$ 和 $-h_i^2(x) (i=l+1, l+2, \dots, q)$ 作为目标函数, 得到无约束多目标最优化问题

$$\max_{x \in X} (f_1(x), \dots, f_m(x), h_1(x), \dots, h_l(x), -h_{l+1}^2(x), \dots, -h_q^2(x)) \quad (3)$$

设对(3)式的前 m 个目标函数 $f_i(x) (i=1, 2, \dots, m)$ 极小化后分别得到最优解 x_i , 即 $f_i(x_i) = \min_{x \in X} f_i(x)$. 设各个 x_i 不全相同, 并记 $f_i^* = f_i(x_i) (i=1, 2, \dots, m)$, 令

$$g_i(x) = \frac{f_i(x) - f_i^*}{\max_{x \in X} \{f_i(x) - f_i^*\}}, i = 1, 2, \dots, m; \quad (4)$$

$$g_{l+m}(x) = \frac{h_l(x) - h_l^*}{\max_{x \in X} \{h_l(x) - h_l^*\}}, i = 1, 2, \dots, l;$$

$$g_{l+m+1}(x) = \frac{-h_{l+1}^2(x) - h_{l+1}^{*2}}{\max_{x \in X} \{-h_{l+1}^2(x) - h_{l+1}^{*2}\}}, i = l+1, l+2, \dots, q.$$

其中, 当 $i=1, 2, \dots, l$ 时, $h_i^* = \min_{x \in X} h_i(x)$; 当 $i=l+1, l+2, \dots, q$ 时, $h_i^* = \min_{x \in X} (-h_i^2(x))$. 设 $m+q=k$, 显然 $0 \leq g_i(x) \leq 1 (i=1, 2, \dots, k)$.

我们研究多目标最优化问题

$$\max_{x \in X} (g_1(x), g_2(x), \dots, g_m(x), g_{m+1}(x), \dots, g_k(x)) \quad (5)$$

设(3)、(5)式的有效解集和弱有效解集分别用 $E(f, h, X)$ 、 $E_w(f, h, X)$ 与 $E(g, X)$ 、 $E_w(g, X)$ 表示, 根据文[8]不难证明

$$E(f, h, X) = E(g, X); E_w(f, h, X) = E_w(g, X) \quad (6)$$

定义3 对多目标最优化问题(4)式, 定义集合

$\tilde{E}(g, X) = \{x: x \in F \cap E_w(g, X); \text{不} \exists \tilde{x} \in F \text{ 使得对 } i=1, 2, \dots, m \text{ 有 } g_i(\tilde{x}) \geq g_i(x) \text{ 且至少有一个不等式严格成立}\}$

$\tilde{E}_w(g, X) = \{x: x \in F \cap E_w(g, X); \text{不} \exists \tilde{x} \in F \text{ 使得对 } i=1, 2, \dots, m \text{ 有 } g_i(\tilde{x}) > g_i(x)\}$.

定理1 设集合 $E(f, X)$ 、 $E_w(f, X)$ 分别为(1)式的有效解集和弱有效解集, $\tilde{E}(g, X)$ 、 $\tilde{E}_w(g, X)$ 是定义3中给出的集合, 则

$$(a) x^* \in E(f, X) \Leftrightarrow x^* \in \tilde{E}(g, X);$$

$$(b) x^* \in E_w(f, X) \Leftrightarrow x^* \in \tilde{E}_w(g, X).$$

证明: (a)充分性: 因为 $x^* \in \tilde{E}(g, X)$, 由集合 $\tilde{E}(g, X)$ 的定义知 $x^* \in F \cap E_w(g, X)$. 用反证法, 设 $x^* \notin E(f, X)$, 则必存在某一 $\tilde{x} \in F$ 使得 $f_i(x^*) \leq f_i(\tilde{x}) (i=1, 2, \dots, m)$ 且不等式中至少有一个严格成立. 从(4)式可推出 $g_i(x^*) \leq g_i(\tilde{x}) (i=1, 2, \dots, m)$ 且不等式中至少有一个严格成立. 显然, 这与 $x^* \in \tilde{E}(g, X)$ 矛盾.

必要性: 反设 $x^* \notin E_w(g, X)$, 由(6)式 $E_w(f, h, X) = E_w(g, X)$ 得出 $x^* \notin E_w(f, h, X)$, 即存在某个 $\tilde{x} \in X$ 使得 $f_i(x^*) < f_i(\tilde{x}) (j=1, 2, \dots, m)$, $h_i(x^*) < h_i(\tilde{x}) (i=1, 2, \dots, l)$, $-h_i^2(x^*) < -h_i^2(\tilde{x}) (i=l+1, l+2, \dots, q)$. (i)若(2)式存在等式约束 ($q > 1$), 因为 $-h_i^2(\tilde{x}) > -h_i^2(x^*) = 0$ 矛盾, 所以 $x^* \in E_w(g, X)$; (ii)若(2)式不存在等式约束 ($q=1$), 则由 $h_l(\tilde{x}) > h_l(x^*) \geq 0 (i=1, 2, \dots, q)$ 得 $\tilde{x} \in F$, 再由 $f_i(x^*) < f_i(\tilde{x}) (j=1, 2, \dots, m)$ 可推出 $x^* \notin E_w(f, X)$, 这与已知条件矛盾, 所以 $x^* \in E_w$

(g, X) .

因为 $x^* \in E(f, X)$, 所以 $x^* \in F$ 且不存在 $\tilde{x} \in F$ 使得 $f_i(x^*) \leq f_i(\tilde{x}) (i=1, 2, \dots, m)$ 其中至少有一个不等式严格成立. 由前面证明了 $x^* \in E_w(g, X)$, 故 $x^* \in \tilde{E}(g, X)$.

(b)类似(a)可证明 $x^* \in E_w(f, X) \Leftrightarrow x^* \in \tilde{E}_w(g, X)$. $\square$

从定理1可以看出, 求解(1)式的有效解集 $E(f, X)$ 和弱有效解集 $E_w(f, X)$ 等价于求定义3中的集合 $\tilde{E}(g, X)$ 和集合 $\tilde{E}_w(g, X)$.

3 个体适应值的分配

根据定理1, 我们把求解(1)式的问题转化为求解无约束多目标最优化(5)在集合 $\tilde{E}(g, X)$ 中均匀分布的解的问题.

对问题(5), 我们定义个体的适应值函数为

$$G(x) = \min_{1 \leq i \leq k} \{w_i g_i(x)\} \quad (7)$$

权重确定方法如下: 令

$$\begin{cases} w_1 = \frac{1}{\cos \theta_1} \sqrt{\prod_{i=1}^{k-1} \sin^{k-i} \theta_i \cos \theta_i} \\ \dots \dots \\ w_{k-1} = \cos \theta_{k-2} / \sin \theta_{k-2} \cos \theta_{k-1} w_{k-2} \\ w_k = \cos \theta_{k-1} / \sin \theta_{k-1} w_{k-1} \end{cases} \quad (8)$$

把闭区间 $[0, \frac{\pi}{2}]$ 等分成 $M+1$ 份, 则 $[0, \frac{\pi}{2}] = \bigcup_{i=0}^M$

$[\frac{i\pi}{2(M+1)}, \frac{(i+1)\pi}{2(M+1)}]$, 令 $\theta_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{i,k-1})$, 其中 θ_{ij}

$\in [\frac{\pi}{2(M+1)}, \frac{2\pi}{2(M+1)}]$, $\dots, \frac{M\pi}{2(M+1)}$, $i=1, 2, \dots, k-1$. 这样共有 M^{k-1} 个不同的 θ_i , 代入(8)式, 可得 M^{k-1} 组权重值:

$w_{1j}, w_{2j}, \dots, w_{kj} (j=1, 2, \dots, M^{k-1})$, 显然 $w_{1j}, w_{2j}, \dots, w_{kj} = 1 (j=1, 2, \dots, M^{k-1})$ 且 $w_{ij} > 0 (i=1, 2, \dots, k-1; j=1, 2, \dots, M^{k-1})$. 把这组权重再代入(7)式, 可得到 M^{k-1} 组适应值函数

$G_j(x) = \min_{0 \leq i \leq k} \{w_{ij} g_i(x)\} (j=1, 2, \dots, M^{k-1})$. 在(5)式的目标函数空间中, 设方向角为 θ_i 的射线与问题(5)的有效界面的交点为 $g(x_j)$, 则由(8)式不难推出

$$\min_{x \in X} G_j(x) = \min_{x \in X} \max_{1 \leq i \leq k} \{w_{ij} g_i(x)\} = G_j(x_j)$$

根据 θ_i 的取法可以看出, $g(x_j) (j=1, 2, \dots, M^{k-1})$ 在(5)式的有效界面上均匀分布. 图1显示了当(5)式为两个目标函数、一个约束函数和 $M=4$ 时, $g(x_j) (j=1, 2, \dots, 16)$ 在(5)式的有效界面上分布的情形.

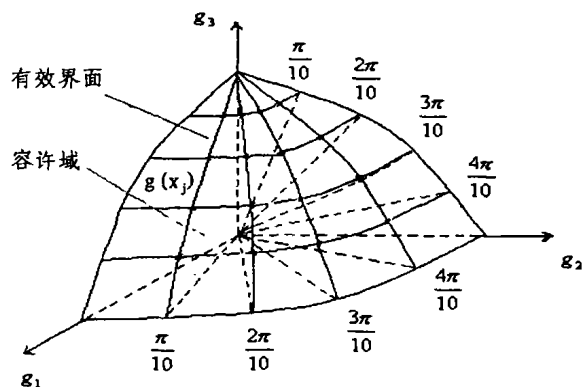


图1 $m=2, k=3, M=4$ 时, $g(x_j)$ 在(5)式的有效界面上分布的情形

当问题(5)的有效解是积极约束时, 在 $m=2, k=3, M=4$ 时如图1所示问题(5)有效界面在目标空间的 g_1, g_2 面上 ($g_3=0$), 这时按上述方法求不出问题(5)的有效解. 为解决这个

问题,在目标函数空间中,把约束函数坐标轴进行平移,即令 $\tilde{g}_i(x) = g_i(x) + a(t, v_i)$, 其中 $a(t, v_i) = v_i(1 - r_i^{(1-\frac{t}{T})^{\lambda}})$, r_i 为 $[0, 1]$ 上的一个随机数, t 表示进化代数, T 代表最大进化代数; λ 为起着 $g_i(x)$ 接近 $f_i(x)$ 快慢程度的一个参数,一般取 2 到 5 之间的值; v_i 是由在搜索空间中满足条件 $f_i(x) \geq 0$ 的点的多少来确定的一个正参数,满足条件的点越多 v_i 的值越小 ($i = m+1, m+2, \dots, k$)。设 $\tilde{g}_i(x) = g_i(x)$ ($i = 1, 2, \dots, m$), 对多目标最优化问题

$$\max_{x \in X} (\tilde{g}_1(x), \tilde{g}_2(x), \dots, \tilde{g}_m(x), \tilde{g}_{m+1}(x), \dots, \tilde{g}_k(x)) \quad (9)$$

其中不难证明它的有效解集 $E(\tilde{g}, X)$ 与 (5) 式的有效解集相同, 即 $E(\tilde{g}, X) = E(g, X)$ 。随着 t 的增加, $\tilde{g}_i(x)$ 越来越接近 $g_i(x)$, 当 $t = T$ 时, $\tilde{g}_i(x) = g_i(x)$ ($i = m+1, m+2, \dots, k$)。若我们定义问题 (9) 式个体的适应值函数为

$$\tilde{G}(x) = \min_{0 \leq i \leq k} \{w_i \tilde{g}_i(x)\} \quad (10)$$

显然, 该适应值函数随着代数 t 的增加在不断变化。当 $m = 2$, $k = 3$ 时, 如图 2 所示, 这时有望求得 (5) 式的积极约束时的有效解。

显然, 随着 M, k 的增大, M^{k-1} 以指数级增长, 这时 θ_j 的计算次数迅速增加。为减少计算次数, 我们使用均匀设计法^[9] 来确定 M 组均匀分布的 θ_j ($j = 1, 2, \dots, M$), 这样能够极大地减少运算量。

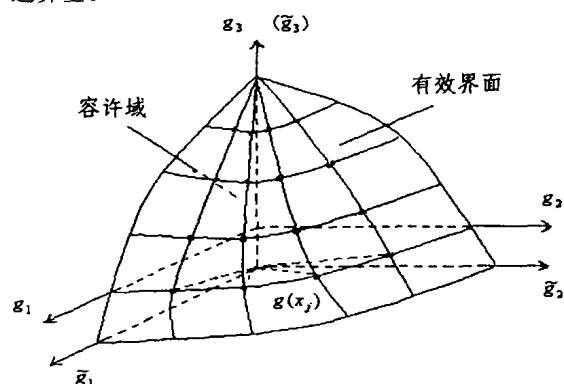


图 2 当 (5) 式的有效解为积极约束时, $g(x_j)$ 在有效界面上分布的情形

4 提出的进化算法

4.1 选择

在第 t 代, 从种群 $P(t)$ 中选择一个使 (9) 式的前 m 个目标函数非劣的点组成的集合 $\tilde{P}(t)$ 。 $\tilde{P}(t)$ 的定义为: $\forall \tilde{x} \in \tilde{P}(t)$, 不 $\exists x \in P(t)$ 使得 $\tilde{g}_i(\tilde{x}) \leq \tilde{g}_i(x)$ ($i = 1, 2, \dots, m$) 且不等式至少有一个严格成立。把集合 $\tilde{P}(t)$ 中的个体分为两类: 一类是 (1) 式可行域中的个体记为 $\tilde{P}^F(t)$, 另一类是 (1) 式非可行域中的个体记为 $\tilde{P}^F(t)$; 则 $\tilde{P}(t) = \tilde{P}^F(t) \cup \tilde{P}^F(t)$ 。在第 $t+1$ 代, 类似从种群 $P(t+1)$ 和集合 $\tilde{P}^F(t)$ 的并, 即 $P(t+1) \cup \tilde{P}^F(t)$ 集合

中选择集合 $\tilde{P}(t+1)$ 。

4.2 进化算法

在本文的进化算法中, $g_i(x)$ 中的 f_i' ($i = 1, 2, \dots, m$), h_i' ($i = 1, 2, \dots, q$) 可用比它小的估值代替。算法的初始种群、杂交、变异算子都采用文 [2] 中类似方法。算法步骤如下:

步 1 初始化种群: 产生初始种群。

步 2 选择: 用文 [2] 中的方法确定当前种群 $P(t)$, 从 $P(t)$ 中按上述方法选择集合 $\tilde{P}(t)$ 。

步 3 杂交: 除了以文 [2] 中的方式杂交外, 集合 $\tilde{P}(t)$ 中的每个个体都参与杂交。

步 4 变异: 以概率 P_m 进行变异。

步 5 终止: 当 $t = T$ 时 (T 为最大进化代数), 则结束运行, 集合 $\tilde{P}^F(T+1)$ 作为 (1) 式的有效集。否则, 转步 2。

5 数值实验

对下面的 2 个问题, 比较了本文中方法 (记为 MGA) 与 [1] 中方法 (记为 HGA) 的计算结果 (详见表 1、表 2), 2 个问题分别为:

$$\text{问题 1. } \begin{cases} \min f_1(X) = 2\sqrt{x_1} \\ \min f_2(X) = x_1(1-x_2) + 5 \\ \text{s. t. } 1 \leq x_1 \leq 4 \\ -19 \leq x_1 + x_2 \leq 14 \end{cases}, \text{ 其中 } X = (x_1, x_2)$$

问题 2. 本问题是一个著名的电子束设计 (beam design) 问题, 其数学模型如下:

$$\begin{cases} \min f_1(X) = 0.758 \{x_1(6.4 \times 10^3 - x_2^2) + (10^3 - x_1)(10^4 - x_2^2)\} \\ \min f_2(X) = 3.298 \times 10^{-5} \left\{ \left(\frac{1}{4.096 \times 10^7 - x_2^2} - \frac{1}{10^8 - x_2^2} \right) x_1^3 + \frac{10^9}{10^8 - x_2^2} \right\} \\ \text{s. t. } 0 \leq x_1 \leq \frac{180(4.096 \times 10^7 - x_2^2)}{9.78 \times 10^6} \\ 40 \leq x_2 \leq 75.2, X = (x_1, x_2) \end{cases}$$

本文算法中参数选取如下: $N = 200$, $p_m = 0.02$, $M = 13$, $\lambda = 3$ 和 $V_i = 1$ ($i = 1, 2, \dots, k$)。对 [1] 中算法, 取 $N = 200$, $p_c = 0.05$, $p_m = 0.02$, $N_{\text{elite}} = 5$, 在对每个点的局部搜索中考虑其邻域中的点数 $k = 2$, 两个算法均取 $T = 40$, 每个问题运行 5 次。在表 1、表 2 中, 将 f_2 的值域等分成 5 个区间, 分别列出了每次运行两个算法在每个区间内求出的最优解的数目, 考察在每个区间最优解的个数是否大体相等, 从而可知最优解是否分布均匀; 考察最优解的数量可知算法是否可求出足够多的最优解。从表 1 到表 2 的结果可看出: 本文中算法求出的最优解总数 nT 均比 [1] 中算法求出的多, 而使用点的数目 ns 却远远比 [1] 中使用的少, 且本文中算法求出的 Pareto 最优解能在 f_2 的值域各区间均匀分布, 从而在解空间中均匀分布; 而 [1] 中算法求出的解却不能在 f_2 的值域各区间均匀分布。从这些结果可看出, 本文中算法很有效。

(下转第 36 页)

表 1 问题 1 的结果

第 k 次运行	在 f_2 的各区间最优解的数目										nT		ns	
	[-31, -25.6]		[-25.6, -20.2]		[-20.2, -14.8]		[-14.8, -9.4]		[-9.4, -4]		MGA	HGA	MGA	HGA
	MGA	HGA	MGA	HGA	MGA	HGA	MGA	HGA	MGA	HGA				
1	43	18	32	5	37	2	40	0	34	0	186	25	4876	9145
2	51	22	23	2	34	4	31	0	27	0	166	28	3914	9176
3	50	20	31	15	33	4	25	0	35	0	174	39	3651	9139
4	34	25	41	1	23	0	37	0	22	0	157	26	3612	8975
5	28	26	36	5	34	2	21	1	18	1	137	35	3486	8917

时间与对应的 Req 持续时间一致。当编码输出信号有效时 $Ack[i]=1, Ack[i]=Req[i]$, (i 是 0 到 7 之间的数), 而 $Req[i]$ 是对应序号为 $Addr[i]$ 的输入请求。如 $Ack[3]=1$, 则 $Req[3]=1$, 此时响应第 $Addr[3]$ 号输入请求, 若 $Addr[3]=5$, 说明第 5 号输入请求 ($P_req[5]$) 获得响应。仲裁模块应送出 $Selx=B1101$ 的链路选择信号, 同时使 $Acp[5]=1$, 表示响应第 5 号输入通道的请求。

结束语 本文所提出的路由器链路调度策略和实现方法已经在 Dawning-UX8 路由器芯片上实现, 运行结果表明: 此调度策略可保证对各输入通道的公平性, 无饿死现象发生。这一实现方法工作稳定可靠, 延迟时间短, 实测的延迟时间为 2 个时钟周期。

链路调度策略对路由器的性能具有决定性影响, 合理地设计调度策略是路由器研究中的一个重要问题。利用虚通道技术提高路由器的吞吐率是目前路由器设计中广泛采用的技术, 引入虚通道后, 链路调度策略也要做相应的变化, 我们下

一步将在这方面作进一步改进。

参考文献

- 1 Pirvu M, Bhuyan L, Ni N. The Impact of link Arbitration on Switch Performance. The 5th International Symposium on HPCA, 1999
- 2 Herbordt M C, et al. Design Trade-Offs of Low-Cost Network Switches. In: IEEE Proc. of the Symposium of Massively Parallel Processing, 1998
- 3 曾嵘, 董向军, 祝明发. 蛙洞路由机置及其芯片设计. 计算机学报, 1997, 20(5), 404~411
- 4 Duato J, et al. MMR: A High-performance Multimedia Router-Architecture and Design Trade-Offs. In: IEEE Proc. of the The Fifth Intl. Symposium on HPCA, 1998
- 5 Dally W J. Virtual Channel Flow Control. IEEE TRANS. On Parallel and Distributed Systems, 1992, 3(2): 194~205

(上接第 120 页)

表 2 问题 2 的结果

第 I 次运行	在 f_2 的各区间最优解的数目 (为简单, $[a, b]$ 代表 $10^6[a, b]$)										nT		ns	
	$[2.7, 3.5]$		$[3.5, 4.3]$		$[4.3, 5.1]$		$[5.1, 5.9]$		$[5.9, 6.7]$					
	MGA	HGA	MGA	HGA	MGA	HGA	MGA	HGA	MGA	HGA	MGA	HGA	MGA	HGA
1	44	34	30	1	26	0	17	0	11	0	128	35	4619	8956
2	45	54	51	0	30	0	40	0	29	0	195	54	4186	9236
3	38	48	30	1	36	0	34	0	27	0	165	49	4023	8956
4	32	35	38	5	23	0	25	0	30	0	148	40	3982	8876
5	26	21	19	3	24	3	16	0	25	0	110	27	3654	8536

结论 本文利用新的约束处理方法和新的适应值函数构造了一个新的约束多目标优化的进化算法, 该算法搜索能力更强, 能更容易求出一组均匀分布的 Pareto 最优解。计算机模拟的结果也表明了这一点。

参考文献

- 1 Ishibuchi H, Murata T. A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Transactions on Syst. Man, Cybern. C, 1998, 28 (3): 392~403
- 2 Leung Y W, Wang Y P. Multiobjective programming using uniform design and genetic algorithm. IEEE Transactions on Syst. Man, Cybern. C, 2000, 30 (3): 293~304
- 3 Deb K, Pratab A, Meyarivan T. Constrained test problems for multi-objective evolutionary optimization. In: Zitzer E, et al. eds. First Intl. Conf. on Evolutionary Multi-Criterion Optimization, Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001. 284~298
- 4 Deb K, Pratab A, Moitra S. Mechanical component design for

multiple objectives using elitist non-dominated sorting GA. In: Schoenauer M, et al. eds. Proc. of the Parallel Problem Solving from Nature VI Conf. 2000. 859~868

- 5 Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 2 (3): 221~248
- 6 Jimenez F, Verdegay J L. Constrained multiobjective optimization by evolutionary algorithms. In: Proc. of the Intl. ICSC Symposium on Engineering of Intelligent Systems (EIS'98), 1998. 266~271
- 7 Deb K, Agrawal S, Pratab A, Meyarivan T. A first elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, In: Schoenauer M, et al. eds. Proc. of the Parallel Problem Solving from Nature VI Conf. 2000. 849~858
- 8 胡毓达. 实用多目标最优化. 上海: 上海科学技术出版社, 1990. 35~55
- 9 Fang K T. Number-theoretic Methods in Statistics. London: Chapman & Hall, 1994. 123~140