应用数学和力学，第9卷第9期(1988年9月)
Applied Mathematics and Mechanics

应用数学和力学编委会编
重庆出版社出版

离散变量的分级优化方法及其应用*

夏德麟　周静芳

(华中工学院，1986年12月4日收到)

摘　要

在工程优化设计中，绝大多数实际问题的设计变量往往限定取离散值，为了求得问题的真正最优解，就必须采用离散变量的优化方法进行求解。本文根据离散变量数学规划的特性，提出了一种分级优化搜索算法。这种方法的基本思想是在约束集合内，寻求一可行的离散初始点，然后在该点的邻域内，进行分级寻优搜索，以求得一个改进的新离散点，随之，以该点作为初始点，重复执行分级寻优搜索过程，直至求得问题的最优解。

通过对工程实例的计算，证明本文所提出的新方法具有快速、简便的特点，能有效地应用各种工程优化设计问题。

一、引　言

工程优化设计的离散变量优化方法是数学规划中一个较难的课题，目前的研究还很不成熟。对于离散变量为整数值的整数规划方法，当前比较流行的有分枝限界法、切平面法和广义罚函数法。这些方法由于要求解一系列非线性规划子问题，故其计算量都比较大。本文作者曾提出一种分枝方向搜索法[3,4]，这种方法虽然能有效地减少计算量，但是该法与上述方法一样，也不适于求解非整数型的离散变量优化问题。为此，本文试图根据直接搜索法原理，提出一种通用的离散变量分级优化方法。

二、问题的描述与基本定义

通常，约束非线性离散变量优化问题可以表示为：

$$\begin{cases} \min\ f(x),\ x\in R^n \\ \text{s. to: } g_i(x)\leqslant 0 \quad (i=1,\ 2,\ \cdots,\ m) \\ \qquad x\in S,\ S=[S_1,\ S_2,\ \cdots,\ S_n]^T \\ \qquad x>0 \end{cases} \tag{2.1}$$

式中R^n是n维离散向量的集合，S为给定各个设计变量元素的集合，其子集$S_j(j=1,2,\cdots,n)$为一向量，它们的维数可以不同，并规定向量元素由小至大排列，而$f(x)$是非线性离散实

* 李灏推荐，

值函数，$g_i(x)$为问题的约束条件，它们是非线性实值函数。

对于这种离散变量的优化设计问题，就是要在所给定的设计变量集合S中，寻求一向量x，使其既满足约束条件，又要使目标函数值达到最小。显然，这类优化设计问题与连续变量优化设计问题有着本质上的区别，这不仅表现在它的可行解集合是有限的，而且在优化问题的基本定义和收敛准则上，也有着很大差异。基于上述原因，这类问题是不能用连续变量的优化方法进行求解的。因此，研究离散变量的优化技术是很必要的。为了便于问题的讨论，首先给出离散变量优化问题的某些基本定义与收敛准则。

1. 离散点与可行离散点

如果点x是以n维离散空间中的网格节点来定义，则点x称为离散点，如果该离散点满足所有约束条件（即$g_i(x)\leqslant 0$，$i=1,2,\cdots,m$），那么这种点称为可行离散点。显然，离散变量优化问题的目标函数值，只能在这些可行离散点上取值。

2. 可行域

在n维离散空间中，所有可行离散点所组成的集合，称为约束离散化问题的可行域，显然它是一种有限集合。

3. 可行离散点的邻域

约束离散变量优化问题的可行离散点x_i的邻域，是以相邻两离散点之间之最短距离（即$\Delta x_j=(x_{i+1})_j-(x_{i-1})_j$）所生成的区域来表示（对于二维问题而言，就是如图1所示的十字带域）。由于约束条件的限制，在该区域上的离散点，有可行的和不可行的两类。因此，在寻优过程中，只需要在该区域中的可行子域Ω中，寻求一个最佳的离散点。

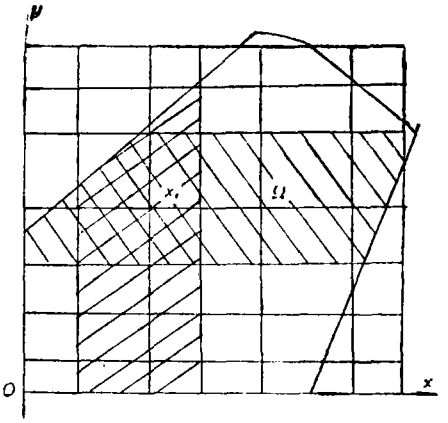


图1 x_i点的可行域

4. 寻优收敛准则

采用分级搜索方法，在离散点x_i的可行邻域Ω中寻优时，若再也找不到一个离散点x，使得$f(x)<f(x_i)$，则该离散点x_i即为问题的局部最优解，于是即结束问题的寻优工作。

本文所提出的分级搜索方法，是假定在两相邻离散点区间，目标函数是单调的，并将每一搜索过程分为三级进行。其中，第一级（称为主轴方向寻优）是以x_i为起始点，沿着平行于坐标轴的直线（其方向由$-\nabla f(x_i)$所对应的分量方向来确定）搜索，以求得该方向的最佳解，然后以该点作为新的初始点，重新进行分级搜索；当找不到这种最优解时，则转入第二级（称为顶点寻优）搜索，即在离散点x_i的相邻格子点集合寻优，若能找到一个改进的点，则以该点作为新的初始点，重新进行分级搜索；反之，转入第三级搜索（即在可行邻域Ω的边界上寻优），此时是分别以上述格子点为起始点，沿着平行坐标轴的各直线进行搜索（其方向由各格子点的方位来确定），若能求得一个改进的点，则以该点为新的初始点，再进行分级搜索；反之，则结束问题的寻优搜索，此时的初始点即为问题的局部最优解。

三、算法的基本原理

与一般的直接搜索算法类似，本文所提出的分级搜索算法，首先亦需要选取一个可行初始离散点x_d，然后按照下述迭代格式：

$$x_n=x_d^0+\lambda D \tag{3.1}$$

在可行域内进行寻优。但是在分级搜索算法的每一迭代过程中，各级的初始点与搜索方向的选取是不同的。一旦初始点与下降搜索方向确定后，则可采用离散变量的一维搜索技术来确定该方向的可行最优解。下面分别对离散变量可行初始点的确定方法，分级搜索技术与一维搜索方法等问题进行说明。

1. 可行初始离散点的确定方法

如前所述，分级搜索算法需要从一个可行初始离散点开始进行搜索，但是对于较复杂的工程实际问题，用户一般难于及时给定，故一般可任意给定一非可行点x_0，然后通过计算方法来确定一个可行离散初始点，现说明本文的方法如下。

假定对于问题(2.1)，给定x_0'使得约束 $g_j(x_0')\leqslant 0$ $(j=1,\cdots, m_1)$，$g_k(x_0')>0$ $(k=m_1+1,\cdots, m)$，则可通过

$$\left.\begin{aligned}&\min G(x)=\sum_k (g_k(x)+\varepsilon)^2 \quad (0<\varepsilon\leqslant 1)\\ &\text{s. to: } g_j(x)\leqslant 0 \qquad (j=1,2,\cdots, m)\\ &\qquad g_l(x)=-x_l\leqslant 0 \qquad (l=1,2,\cdots,n)\end{aligned}\right\} \tag{3.2}$$

求得一个满足约束的点x_c，然后在集合S中离散化，从而求得可行初始离散点。问题(3.2)的迭代搜索公式是：$x_n=x_0'+\lambda d$进行，其中搜索方向为：

$$d=\sum_k -\frac{g_k(x_0)}{\nabla g_k^T(x_0)\cdot\nabla g_k(x_0)}\nabla g_k(x_0) \quad (k=m_1,\cdots, m) \tag{3.3}$$

式中 $\nabla g_k(x_0)=\nabla g_k(x_0)/\|\nabla g_k(x_0)\|$。当按上述迭代公式在$d$方向进行一维搜索时，是以步长增量$\Delta\lambda$，来确定新点$x_n$，并检查约束 $g_j(x)\leqslant 0$，$g_l(x)=-x_l\leqslant 0$，若其中有一个违反，则重新形成$G(x)$与$g_j(x)$，继续对新的规划问题(3.2)求解，直至获得满足原问题所有约束的x_c点为止。最后，在集合S中离散化，即可求得一个可行初始离散点。

2. 分级搜索方法

分级搜索方法的基本思想如前节所述，现在来具体说明在每级搜索过程中，初始点与搜索方向的确定方法。对于第一级搜索初始点取迭代开始时，所确定的可行离散点x_d^0，而可能的下降搜索方向为：

$$\begin{aligned}&D_1=[\text{sign}(-\nabla f_1(x_d^0)),\ 0,0,\cdots,\ 0]^T,\\ &D_2=[0,\ \text{sign}(-\nabla f_2(x_d^0)),\ 0,\cdots,\ 0]^T,\\ &\cdots\cdots \qquad\qquad \cdots\cdots\\ &D_n=[0,\ 0,\ \cdots\cdots,\ \text{sign}(-\nabla f_n(x_d^0))]^T.\end{aligned}$$

式中$\nabla f_i(x_0)(i=1,2,\cdots, n)$为目标函数在点$x_0$处的梯度的第$i$个分量。显然，这些搜索方向，皆为单位向量，在寻优过程中，通常取其最速下降方向D，即

$$D=D_i|_{\min\{D_i^T\}\nabla f(x_d^0)\leqslant 0},\ (i=1,\ 2,\ \cdots,\ n) \tag{3.4}$$

作为该级的搜索方向，并按迭代公式 $x_n=x_d^0+\lambda D$ 用离散量的一维搜索算法求取该方向的可行最优点，并以该点作为新的初始点再进行分级搜索，若该可行最优点不存在，则转入第二个最速下降方向搜索寻优；若上述 n 个 D_i 方向都找不到可行最优点，则转入第二级搜索。

第二级搜索是在可行初始离散点 x_d^0 的相邻节点集合寻优。这种节点共有 $3^n-(2n+1)$ 个，但是在寻优过程中，只需要在函数下降的区域内对这种点进行比较。现在的问题是如何确定这些节点的位置。本文的方法是对每一个所给定的设计变量子集 S_j 中的分量（从小到大排序），按其自然序号来指定它，而相邻节点的序号与可行初始离散点的序号仅差一个正一（或负一），故这些相邻节点的序号向量与坐标位置，可由初始点求得。若令第 i 个节点的序号向量为 D_{pi} 则由

$$\{D_p\}^T\nabla f(x_d^0)<0 \qquad (i=1,\ 2\cdots,\ 3^n-(2n+1))$$

可以确定第 i 个节点是否为一个较好的改进点，从这些改进点选取函数值最小，并且满足约束条件的点作为第二级搜索的最优点；并以它作为新的初始点重新进行分级搜索；若这种最优点不存在，则转入第三级搜索。

对于第三级搜索是分别以可行初始离散点 x_0 的相邻节点 x_i 作为初始点，而搜索方向是在每个相邻节点序号向量的分量向量：

$$\{D_{p1}\}_i=[d_{p1},\ 0,\ 0,\ \cdots,\ 0]^T$$
$$\{D_{p2}\}_i=[0,\ d_{p2},\ 0,\ \cdots,\ 0]^T$$
$$\cdots\cdots$$
$$\{D_{pn}\}_i=[0,\ 0,\ \cdots,\ d_{pn}]^T \qquad (i=1,\ 2,\ \cdots,\ 3^n-(2n+1))$$

中选取。显然对于每一个相邻节点 x_i，这种方向共有 n 个，但是这种方向并不都是可行的。我们的任务就是要确定这些可行方向，并在其中选择一个最速下降方向。由于相邻节点可能是可行点或者非可行点（不满足约束的离散点），因此，对于可行相邻节点的下降搜索方向由下述不等式：

$$\{D_{pj}\}_i^T\nabla f(x_b)<0 \qquad (i=1,\ 2,\ \cdots,\ 3^n-(2n+1);\ j=1,\ 2,\ \cdots,\ n)$$

确定；当 x_i 为非可行点时，其下降搜索方向，由下列不等式

$$\begin{cases}\nabla g_k^T(x_i)\cdot\{D_{pj}\}_i>0 \quad (k=1,\ 2,\ \cdots,\ m_2)\\ \nabla f^T(x_0)\{x_i+\lambda\{D_{pj}\}_i-x_0\}<0\end{cases} \tag{3.5}$$
$$(i=1,\ 2,\ \cdots,\ 3^n-(2n+1),\ j=1,\ 2,\ \cdots,\ n)$$

来确定。其中 $g_k(x_i)$ 为相邻节点 x_i 所不满足的约束，第三级搜索就是在这些下降搜索上寻优，若这种最优点能找到，就以该点作为新的初始点，重新进行分层搜索；反之迭代结束，那么此时的初始点即为问题的局部最优解。

3. 离散变量的一维搜索方法

对于离散变量的优化问题，由于设计变量是在所给变量集合 S 中取值，故在一维搜索时，需要采用离散变量的一维搜索方法。本文给出一种新的算法，是基于单峰函数必存在一个最优点，而在该点的某一邻域，其对应的函数曲线可以用一二次曲线来逼近，而该二次曲线的极值点，即为一维搜索方向的最优点。因此，问题归结为如何求得这个邻域中的三点。本文的方法是分为下述三个阶段进行。在第一阶段，首先选取一个初始步长 s_0 和若干个点 x_2，

$x_3,\cdots\cdots,x_n$，使得$x_i=\mathrm{Fis}_0+a$，并令$x_1=a$为区间的起始点。显然在第一次选迭代时，一般取$a=0$，而随后则取新区间的起始点。式中Fi为Fibonacci数。当x_n的函数值大于初始点的函数值时，则该区间的长度为一定值，即$R_1=x_n-x_1$(如图2所示)，于是这些点的均方根值为：

$$x_b=\left[\frac{1}{n}\sum_{j=1}^{n}x_i^2\right]^{\frac{1}{2}},\quad f_0=f(x_0)$$

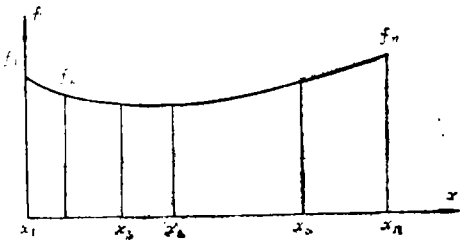


图2 一维搜索求步长

现在可在该区间的$n+1$点中取一点B，使得其函数值为最小，然后在B点两边各取一个相邻点x_A，x_B使得$x_A<x_B<x_C$。

第二阶段，就是利用上述三点及其函数值进行二次插值，求得其极小值x_m，$x_m=-2^{-1}b/a$

式中：

$$a=\frac{(x_B-x_C)f(x_A)+(x_C-x_A)f(x_B)+(x_A-x_B)f(x_C)}{(x_A-x_C)(x_A-x_B)(x_B-x_C)}$$

$$b=\frac{f(x_B)-f(x_C)}{x_B-x_C}-(x_C+x_B)a$$

若$f_B-f(x_m)<\varepsilon$，则搜索停止。反之，则在x_A，x_B，x_C，x_m中取其函数值最小的三点，重新进行二次插值，直到迭代计算满足所要求的精度为止。

第三阶段用所求得一维搜索步长求新点x_n，若该点满足约束条件，则在集合S中离散化，求得所需要的可行点。如果x_n不满足约束，则求该搜索方向与最不满足约束的交点（其方法同上，只是将函数值之绝对值最小的三点进行二次插值），然后离散化求得所需要的可行点。

4. 算法的计算流程

根据本文所提出的方法，可以编制如下计算流程（见844页）。

四、计 算 实 例

为了验证本文方法的正确性与有效性我们曾对各种工程实际问题进行试算，现给两个例子如下。

1. 船舶舱口盖的优化设计

某船的铝质箱形剖面盖如图3所示，材料的弹性模量$E=7\times10^5\mathrm{kg/cm^2}$，泊松比$\mu=0.3$，试在所给定面板厚度集合$S_1$与梁高度集合$S_2$中，选取板厚$t_f$和梁高度$h$，使箱形梁的重量为最小，并满足约束条件：

（1） 容许弯曲应力：$\bar{\sigma}=700\mathrm{kg/cm^2}$；

（2） 容许剪应力：$\bar{\tau}=450\mathrm{kg/cm^2}$；

（3） 容许挠度：$\bar{\delta}=1/400$；

（4） 面积不失稳；

（5） t_f，h为所给定的离散值(mm)，

假定：

$S_1=[2.0,\ 2.5,\ 3.0,\ 3.5,\ 4.0,\ 4.5,\ 5.0,\ 5.5,\ 6.0,\ 6.5,\ 7.0,\ 7.5,\ 8.0]^T$

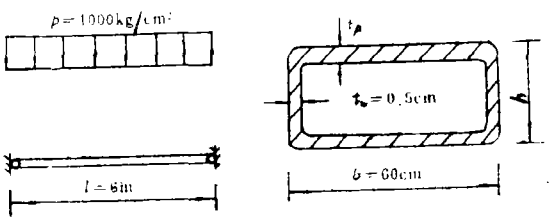
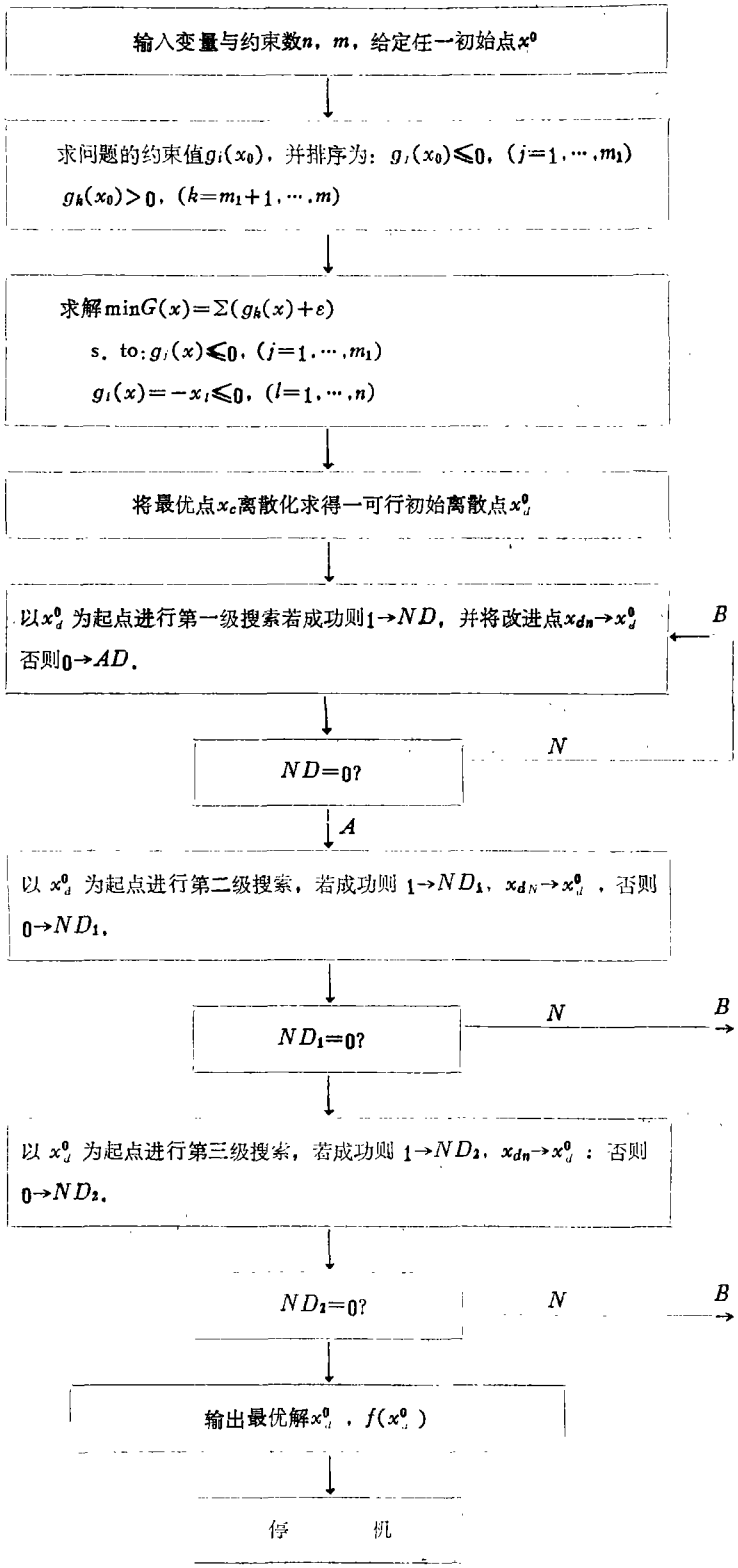


图3 船舶舱口盖


输入变量与约束数n，m，给定任一初始点x^0
求问题的约束值$g_i(x_0)$，并排序为：$g_j(x_0)\leqslant 0$，$(j=1,\cdots,m_1)$
$g_k(x_0)>0$，$(k=m_1+1,\cdots,m)$
求解$\min G(x)=\Sigma(g_k(x)+\varepsilon)$
s. to: $g_j(x)\leqslant 0$，$(j=1,\cdots,m_1)$
$g_l(x)=-x_l\leqslant 0$，$(l=1,\cdots,n)$
将最优点x_c离散化求得一可行初始离散点x_d^0
以x_d^0为起点进行第一级搜索若成功则$1\to ND$，并将改进点$x_{dn}\to x_d^0$
否则$0\to AD$.
B
N
$ND=0$?
A
以x_d^0为起点进行第二级搜索，若成功则$1\to ND_1$，$x_{dN}\to x_d^0$，否则
$0\to ND_1$.
N
B
$ND_1=0$?
以x_d^0为起点进行第三级搜索，若成功则$1\to ND_2$，$x_{dn}\to x_d^0$；否则
$0\to ND_2$.
$ND_2=0$?
N
B
输出最优解x_d^0，$f(x_d^0)$
停 机

$S_2=[200, 201, 205, 206, 207, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220]^T$

对于该问题可用下述数学形式表示：

$$\min f(x)=1200t_f+10h$$

$$\text{s.to: } g_1=\frac{45\times10^4}{7}t_f^{-3}h^{-1}-1\leqslant0,\quad g_2=\frac{225\times10^4}{7}t_f^{-3}h^{-2}-1\leqslant0$$

$$g_3=\frac{45\times10^2}{7}t_f^{-1}h^{-1}-1\leqslant0,\quad g_4=40h^{-1}-1\leqslant0,\quad t_f\in S_1,\quad h\in S_2$$

现给出计算结果如下：

表1 取$x_0=[3,215]^T$

迭代次数 \ 名称	x_d^0	$f(x_d^0)$
0	$[7.0,216.0]^T$	10559.9999
1	$[7.0,215.0]^T$	10550.0006

表2 当取$x_0=[3,219]^T$

迭代次数 \ 名称	x_d^0	$f(x_d^0)$
0	$[7.5,219]^T$	11190.2110
1	$[7.0,219]^T$	10590.3124
2	$[7.0,215]^T$	10550.0006

2. 压缩弹簧的优化设计

设有一内燃机汽门用弹簧，其最大变形量$\lambda=16.59$(mm)，工作载荷$F=680$N，工作频率$f_r=25$Hz，最高工作温度150℃，材料为50$CrVA$钢丝，要求寿命$N=10^6$(循环工作次数)。弹簧结构要求满足：钢丝的直径d，弹簧外径D，工作圈数$n(\geqslant3)$需要在规定的集合S_1，S_2，S_3中选取，使在满足约束条件下，弹簧的重量最小。

根据上述要求，该问题的数学形式可描述为：

$$\min f(x)=1.925\times10^{-5}(x_3+1.8)x_2x_1^2$$

$$\text{s.to: } g_1(x)=-404.9+3503.96x_2^{0.86}/x_1^{2.86}\leqslant0,\quad g_2(x)=-x_2/x_1+6\leqslant0$$

$$g_3(x)=\frac{(x_3+1.3)x_1+18.25}{x_2}-5.3\leqslant0,\quad g_4(x)=2.5-x_1\leqslant0$$

$$g_5(x)=x_1-9.5\leqslant0,\quad g_6(x)=30-x_2\leqslant0$$

$$g_7(x)=x_2-60\leqslant0,\quad g_8(x)=375-3.56\times10^5x_1/(x_2^2x_3)\leqslant0$$

$$g_9(x)=3-x_3\leqslant0,\quad g_{10}(x)=680-10000\times16.59x_1^4/x_3x_3^3\leqslant0$$

而给定设计变量集合分别为：

$S_1=[2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0]^T$

$S_2=[30.0, 35.5, 38.0, 40.0, 45.0, 48.0, 50.0, 55.5, 56.0, 58.5, 60.0]^T$

$S_3=[3, 4, 5, 6, 7, 8, 9]^T$

该问题的优化设计结果为(给定$x_0=[6.0, 48.0, 6.0]^T$)

迭代次数 \ 名称	x_d^0	$f(x_d^0)$
0	$[8,48,3]^T$	0.2839
1	$[7.5,48,3]^T$	0.2495
2	$[7.0,45,3]^T$	0.2037
3	$[6.5,40,3]^T$	0.1562

最优解为：

$x_{op}=[6.5, 40, 3]^T$ (mm, mm, 圈数)

$f_{op}(x_{op})=0.1562$(kg)

五、结 束 语

通过工程实例的计算表明，本文所提出的方法，原理是正确的，能有效地求解离散变量的非线性规划问题。该算法具有原理简单，计算量小的特点。

在工程实际问题中，离散变量的非线性规划问题有多种形式，即离散值可取整数值，也可取实数值，或者兼而有之；有时在设计变量中，既包含离散量，又包含有连续量，对于这些问题，都不难用本文所提出的方法得到解决。

参 考 文 献

[1] Garfinhel, R. S. and G. L. Nemhauser, *Integer Programming*, John Wiley and Sons (1972).

[2] Xia De-lin and C. L. Wang, On an approximation method of geometric programming, *Congressus Numerantium*, **34**, Canada (1982).

[3] 夏德麟，整数线性规划的一种新方法——分枝方向搜索法，应用数学和力学，**6**，3 (1985)，277—282.

[4] 夏德麟，船舶结构优化的离散变量方法，中国造船工程学会论文集 (1984).

A Discrete Nonlinear Stepping Optimization Method and Its Application

Xia De-lin Zhou Jin-fang

(*Huazhong University of Science and Technology, Wuhan*)

Abstract

Most of the practical design variables should always be discrete quantity within engineering optimization design problems. To obtain the true optimization solution, a discrete optimization method must be used. In this paper, a new method called step optimization search method is presented to solve the discrete quantity mathematic programming problems. The basic idea of this method is to find out an initial feasible point and then to search the optimum point step by step in the neighbouring region of this point so as to obtain an improved new discrete point. Respectively, the new point can be taken as initial one, and the whole process can be carried out once more until the optimum solution of the problem is obtained.

Some results of numerical examples of practical problems show that this new method can solve problems quickly and simply and can be applied in a lot of engineering design problems.