

粒子群算法中随机数参数的设置与实验分析

刘志雄^{1,2}, 梁 华³

(1. 武汉科技大学 机械自动化学院, 湖北 武汉 430081; 2. 天津港(集团)有限公司 博士后科研工作站, 天津 300461;

3. 重庆工商大学 计算机与信息工程学院, 重庆 400067)

摘要: 粒子群算法的相关参数, 对粒子群算法的优化性能有着重要影响, 本文针对粒子群算法模型中随机数参数的设置问题展开实验分析. 首先, 由于各种高级程序语言的结构不同, 在粒子群算法的实现程序中, 对速度更新公式内同一个粒子速度向量, 其各个分量的随机数参数的设置各不相同. 其次, 根据连续函数优化问题和作业车间调度问题中的典型测试算例, 以及对于设备拥有量参数优化问题的计算, 表明在粒子群算法中设置不同的随机数参数将对粒子群算法的优化性能产生较大影响, 并且, 对一个粒子速度向量中的不同分量所对应的随机数参数, 如果设置相同的值, 可以有效地提高粒子群算法的优化效率.

关键词: 粒子群算法; 随机数; 参数设置; 调度; 优化

中图分类号: TP39 文献标识码: A

Parameter setting and experimental analysis of the random number in particle swarm optimization algorithm

LIU Zhi-xiong^{1,2}, LIANG Hua³

(1. College of Machinery and Automation, Wuhan University of Science and Technology, Wuhan Hubei 430081, China;

2. Postdoctoral Research Center, Tianjin Port (Group) Co LTD, Tianjin 300461, China;

3. School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing 400067, China)

Abstract: The parameters in particle swarm optimization have important effect on the optimization performance. The parameter setting of the random number in the particle swarm optimization model is analyzed by the experiments. First, because of different structures in different high-level languages, we find that in the program of particle swarm optimization algorithm, different components of a velocity vector may have different parameter settings for the corresponding random number in the particle velocity updating equation. Next, in continuous function optimization and benchmark tests of Job Shop scheduling, as well as the computation of the equipment-possession-quantity parameter optimization model, all results indicate that different parameter settings for the random number may cause significantly different effects on the optimization performance of particle swarm optimization algorithm. Furthermore, it is also found that the optimization efficiency of a particle swarm optimization algorithm can be obviously improved if the corresponding random number of different components of a velocity vector is set to the same value.

Key words: particle swarm optimization algorithm; random number; parameter setting; scheduling; optimization

1 引言(Introduction)

粒子群算法(particle swarm optimization, PSO)是1995年由美国的Kennedy博士和Eberhart博士^[1]提出的一种基于群智能的优化算法, 由于该算法是一种模拟鸟群觅食的仿生算法, 并且有一组显式的数学计算模型, 操作和实施简单. 粒子群算法提出后, 在连续空间和离散空间优化问题中得到了广泛的应用, 其算法的良好操作性和优化性能均得到了检

验. 同时, 相关学者对粒子群算法模型得到不断的改进^[2~4], 模型参数的设置也得到广泛的分析和探讨研究^[5~8].

粒子群算法中的参数主要包括惯性权重、加速因子、收敛因子、种群数量等, 大量实验分析说明, 这些参数对粒子群算法的优化性能影响很大. 不同类型的优化问题, 粒子群算法的参数设置也不尽相同; 同一优化问题, 不同粒子群算法模型, 模型中的

收稿日期: 2009-12-29; 收修改稿日期: 2010-05-16.

基金项目: 国家自然科学基金资助项目(70801047); 中国博士后科研基金资助项目(20090450769); 湖北省自然科学基金资助项目(2009CDB108); 湖北省教育厅科研项目(Q20101115).

参数设置也可能有差异.

在粒子群算法的粒子速度更新公式中,除了惯性权重和加速因子之外,还有一个参数——随机数参数,是指0至1之间的随机数.在编写粒子群算法的程序时,这个参数一般由编程语言的相应随机数函数产生.由于不同高级编程语言的语言结构和编程方法有差异,因此,编写粒子群算法程序时,对于同一速度向量的不同分量,在速度更新公式中,对应的随机数参数可以设置为相同值或者不同值.在粒子群算法的相关研究中,随机数参数上述设置问题几乎被研究者们忽视,因而这种随机数的不同设置对粒子群算法优化性能的影响,到目前为止还缺乏相关分析.

本文首先从不同高级程序语言在实现粒子群算法时的结构区别出发,对粒子群算法中随机数参数的设置问题进行分析;然后针对随机数参数的不同设置,采用粒子群算法对连续函数优化问题和作业车间(job shop)调度问题中的若干测试算例及设备拥有量模型的参数优化问题进行优化.通过面向连续空间和离散空间两大类优化问题的实验研究,对粒子群算法中随机数的参数设置问题展开探讨分析.

2 粒子群算法(Particle swarm optimization algorithm)

粒子群算法是模拟鸟群觅食的过程,采用速度—位置模型进行搜索.在粒子群算法中,每个优化问题的解都是搜索空间的一只鸟,称之为“粒子”,所有粒子都有一个由被优化的函数决定的适应值,每个粒子还有一个速度决定它们搜索的方向和距离,所有粒子通过追随当前的最优粒子在解空间中进行搜索.最优粒子包含两个部分:个体最优粒子和全局最优粒子.个体最优粒子是指每个粒子本身在迭代过程中所找到的最优解,称为个体最优解;而全局最优粒子则是指当前粒子群中的最优粒子,称为全局最优解.粒子群中的每个粒子通过追随个体最优粒子和全局最优粒子进行搜索.假设粒子群中的第*i*个粒子在*D*维空间中的位置表示为一个*n*维向量 $x_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{iD})$,速度 $v_i = (v_{i1}, v_{i2}, \dots, v_{ij}, \dots, v_{iD})$ 决定了粒子在搜索空间单次迭代的位移.当前粒子群中的个体最优粒子表示为 $p_i = (p_{i1}, p_{i2}, \dots, p_{ij}, \dots, p_{iD})$,全局最优粒子表示为 $g = (g_1, g_2, \dots, g_j, \dots, g_D)$,粒子群中的所有粒子根据式(1)(2)来更新其速度和位置:

$$v_{ij} = \omega v_{ij} + c_1 r_1 (p_{ij} - x_{ij}) + c_2 r_2 (g_j - x_{ij}), \quad (1)$$

$$x_{ij} = x_{ij} + v_{ij}. \quad (2)$$

式(1)中, r_1 和 r_2 为(0,1)之间的随机数, c_1 和 c_2 为

学习因子,一般取 $c_1 = c_2 = 2$, ω 为惯性权重, ω 取较大值粒子群算法具有较强的全局搜索能力, ω 取较小值粒子群算法倾向于局部搜索.速度 v_i 可以设定在 v_{\max} 和 v_{\min} 之间,当超过这个区间时,就取对应的上下限值. x_i 位置也可设定在一定的 $[x_{\min}, x_{\max}]$ 区间范围内.由式(1)(2)所组成的粒子群算法一般称为基本粒子群算法(简称PSO).在此基础上,如果将惯性权重 ω 的取值根据迭代次数依次递减,此时粒子群算法称为惯性权重线性递减粒子群算法.一般情况下,将 ω 的初始值设置为0.9,然后按照迭代次数线性递减到0.4.惯性权重线性递减粒子群算法是粒子群算法研究领域一种比较常用的算法模型,并且通过研究发现,粒子群模型的选择与随机数参数的设置问题相关性不大.因此,本文将采用上述惯性权重线性递减粒子群算法对不同问题进行优化.

3 随机数参数的设置分析(Analysis of the random number parameter setting)

在粒子群算法速度更新式(1)中,随机数参数 r_1 和 r_2 一般是由高级程序语言所对应的随机函数产生.

如果对于速度向量 v_i 的两个不同分量 v_{ic} 和 v_{id} ,其在粒子速度更新式(1)对应的随机数分别用 r_{1c} 和 r_{2c} 以及 r_{1d} 和 r_{2d} 表示,那么,在粒子群算法的程序实现过程中,由于高级程序语言的不同结构,或者同一种程序不同的粒子群算法编程实现过程,可能导致两个不同分量 v_{ic} 和 v_{id} 所对应的随机数存在不同的设置.即可以设置为 $r_{1c} \neq r_{1d}$, $r_{2c} \neq r_{2d}$;也可以设置为 $r_{1c} = r_{1d}$, $r_{2c} = r_{2d}$.

因此,本文以Visual Basic和MATLAB两种不同高级语言为例,分析粒子群算法的不同程序实现过程以及随机数参数的不同设置问题.

3.1 Visual Basic的粒子群算法程序设计(Particle swarm optimization algorithm programming for Visual Basic)

假设优化问题是一个*D*维空间问题(即每个粒子速度和位置向量都包含*D*个分量),种群数量为POPSIZE,那么,使用Visual Basic编写粒子群算法时,一般采用双层循环语句来实现.对粒子速度和位置进行更新时,其代码是由种群数量的外循环和问题维数的内循环两个循环代码组成,具体可以描述如下(在Visual Basic程序设计语言中,随机数由rnd函数产生):

```
For i = 1 to POPSIZE  种群数量外循环
  For j = 1 to D      问题维数内循环
     $v_{ij} = \omega * v_{ij} + c_1 * \text{rnd} * (p_{ij} - x_{ij}) +$ 
     $c_2 * \text{rnd} * (g_j - x_{ij})$ 
     $x_{ij} = x_{ij} + v_{ij}$ 
```

Next j

Next i

上述Visual Basic的代码中, 对于同一个粒子速度向量 v_i 的每个分量 v_{ij} , 所对应的由rnd函数产生的随机数参数均不同, 即随机数参数被设置为不同的值。当然, 上述代码描述也可以描述如下:

For $i = 1$ to POPSIZE 种群数量外循环

$r_1 = \text{rnd}$

$r_2 = \text{rnd}$

For $j = 1$ to D 问题维数内循环

$v_{ij} = \omega * v_{ij} + c_1 * r_1 * (p_{ij} - x_{ij}) +$

$c_2 * r_2 * (g_j - x_{ij})$

$x_{ij} = x_{ij} + v_{ij}$

Next j

Next i

很明显, 此时上述Visual Basic的第二种代码中, 对于同一速度向量 v_i 中的每个分量 v_{ij} , 所对应的随机数(由 r_1 或者 r_2 产生)将是相同的值。因此, 使用Visual Basic来设计粒子群算法时, 可以有2种不同的格式。

然而, 对于Visual Basic程序语言的使用者来说, 绝大多数程序设计者由于编程习惯或者喜好, 更倾向于前面一种代码格式, 即将随机数参数设置为不同值。第一种Visual Basic代码描述方式可以说是绝大多数程序设计者的一种默认方式。

3.2 MATLAB的粒子群算法程序设计(Particle swarm optimization algorithm programming for MATLAB)

MATLAB是一种基于矩阵运算的科学计算软件, 除了具有强大的科学计算功能外, 其本身也具有程序设计功能。MATLAB通常采用矩阵运算编程, 如果用MATLAB代替Visual Basic来编写粒子群算法的程序, 就采用单层循环语句来实现, 那么粒子群算法的程序结构会发生一些变化, 粒子速度和位置更新公式可以描述如下(在MATLAB程序设计语言中, 随机数由rand函数产生):

For $i = 1$ to POPSIZE 种群数量外循环

$v[i] = \omega * v[i] + c_1 * \text{rand} * (p[i] - x[i]) +$

$c_2 * \text{rand} * (g - x[i])$

$x[i] = x[i] + v[i]$

Next i

上述代码描述中, 整个速度更新公式都是用矩阵向量表示的。此时, $v[i]$ 中所包含的分量 $v[i][j]$ 都被隐藏, 即问题维数的内循环被隐藏了。可以看出, 对于同一个 $v[i]$ 中的每个分量 $v[i][j]$, 对应的由MATLAB随机数函数rand所产生的随机数都是

相同值。

这里暂且将这种代码描述方式称为MATLAB的矩阵操作方式。此时的MATLAB矩阵操作方式就和Visual Basic的第二种代码描述方式在粒子群算法的程序设计过程和结果方面是一致的。

与Visual Basic程序设计语言一样, MATLAB在程序设计时除了可以使用矩阵外, 同样可以使用数组来表示矩阵。因此, 如果使用数据来表示矩阵, 那么粒子群算法中粒子速度和位置更新公式的MATLAB程序代码可以描述如下:

For $i = 1$ to POPSIZE 种群数量外循环

For $j = 1$ to D 问题维数内循环

$v[i][j] = \omega * v[i][j] + c_1 * \text{rand} * (p[i][j] -$

$x[i][j]) + c_2 * \text{rand} * (g[j] - x[i][j])$

$x[i][j] = x[i][j] + v[i][j]$

Next j

Next i

可以看出, 此时MATLAB程序代码描述和Visual Basic的第一种程序代码描述没有区别, 并且, 此时对于同一个粒子速度 $v[i]$ 中的每个分量 $v[i][j]$, 对应的由MATLAB随机数函数rand所产生的随机数都是不同值。这里暂且将这种代码描述方式称为MATLAB的数组操作方式。

因此, 对于MATLAB来说, 针对粒子群算法的程序设计也可以包含2种不同的结构。然而, 对于已经熟悉使用MATLAB和喜好矩阵运算的程序设计者来说, 他们更愿意使用MATLAB的前一种代码描述方式, 即矩阵操作方式。而对于不熟悉MATLAB和矩阵运算的程序设计者而言, 在他们使用MATLAB编程时, 可能很容易将其他程序设计语言的风格和习惯, 如Visual Basic, 引入到MATLAB的程序设计中, 此时, 就他们更容易和倾向于使用MATLAB的第二种代码描述方式, 即数组操作方式。

综上所述, 通过Visual Basic和MATLAB两种不同程序语言的粒子群算法实现过程可以看出, 粒子群算法的随机数参数设置问题在程序实现过程中是实际存在的。从上述分析结果看, 同一种程序设计语言, 粒子群算法的实现过程可以存在两种不同的程序设计结果。正因为于此, 粒子群算法中随机数参数的设置问题往往被高级程序设计语言本身的结构所掩盖而被忽视。

4 计算结果与分析(Calculation results and analysis)

为了进一步分析随机数参数的不同设置对优化结果的影响, 这里分别针对连续函数优化问题

和调度问题(job shop调度)中的几个典型算例, 基于MATLAB语言编写程序, 分别采用MATLAB的2种不同粒子群算法程序设计方式, 即矩阵操作方式和数组操作方式, 也就是说将随机数参数分别设置为相同值和不同值. 这里粒子群算法采用前文所述的惯性权重线性递减粒子群模型.

4.1 连续函数优化问题的计算结果(Calculation results of continuous optimization problem)

这里主要针对Sphere, Rosenbrock和Rastrigin等3个测试函数进行优化计算, 为了分析粒子种群数量、函数维数与随机数参数设置的相互关系, 将粒子种群数量分别取值20, 40, 60, 80, 100, 即

$$\text{POPSIZE} = \{20, 40, 60, 80, 100\},$$

函数的维数设定为 $D = \{2, 4, 6, 8, 10, 20\}$. 不同种群数量和函数维数情况下, 针对几个算例, 粒子群算法都独立运行20次, 每次优化运行最大迭代次数为1000次. 由于3个测试函数的全局最小值均为0, 因此同时设定函数值小于0.00001时运行终止.

4.1.1 Sphere函数的计算结果(Calculation results of Sphere function)

Sphere函数如式(3)所示, 是一个单峰函数, 当 $x_i = 0$, 函数有全局唯一最小值 $f_1 = 0$.

$$f_1 = \sum_{i=1}^D x_i^2. \quad (3)$$

Sphere函数的计算结果如图1和图2所示, 其中: 图1为种群数量为20时, 粒子群算法在矩阵操作方式和数组操作方式下, 20次运行找到最优值的最小迭代次数和平均迭代次数; 而图2则是种群数量为100时的Sphere函数的计算结果.

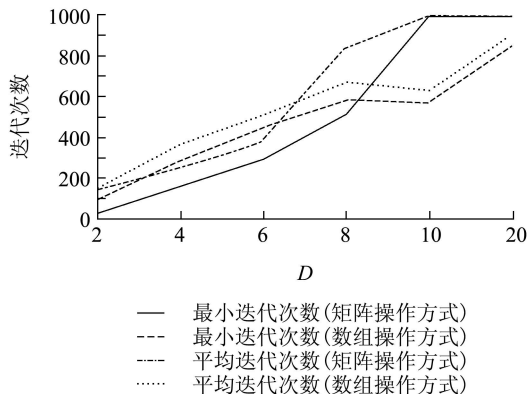


图1 种群数量为20时Sphere函数的计算结果

Fig. 1 Computational results of Sphere function as swarm size is 20

在图1中, 当函数维数 D 小于10时, 矩阵操作方式

下, 粒子群算法找到最优值的最小迭代次数要好于数组操作方式; 而当函数维数 D 小于8时, 矩阵操作方式下, 粒子群算法找到最优值的平均迭代次数要好于数组操作方式. 当函数维数增加时, 矩阵操作方式下粒子群算法的计算结果则比数组操作方式差, 这主要是由于函数维数的增加使得计算复杂度增加. 从图2的计算结果看, 当种群数量为100时, 上述情况可以发生改变, 无论是找到最优值的最小迭代次数还是平均迭代次数, 采用矩阵操作方式的粒子群算法均能取得比数组操作方式要好的计算结果.

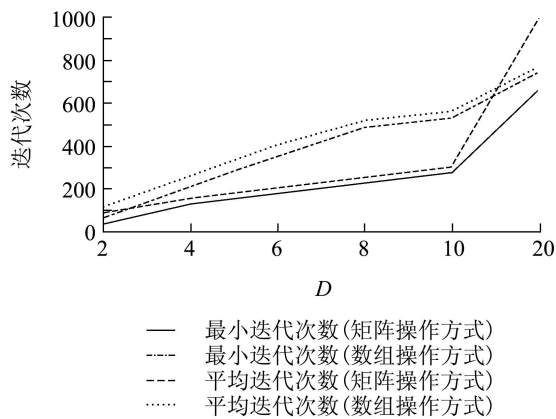


图2 种群数量为100时Sphere函数的计算结果

Fig. 2 Computational results of Sphere function as swarm size is 100

图3为矩阵操作方式下, 粒子群算法运行20次找到最优值的平均迭代次数. 从图3看, 在相同函数维数的情况下, 随着种群数量的增加, 矩阵操作方式下的粒子群算法找到最优值的平均迭代次数是逐渐递减的, 说明种群数量的增加能有效改善矩阵操作方式下粒子群算法的优化性能.

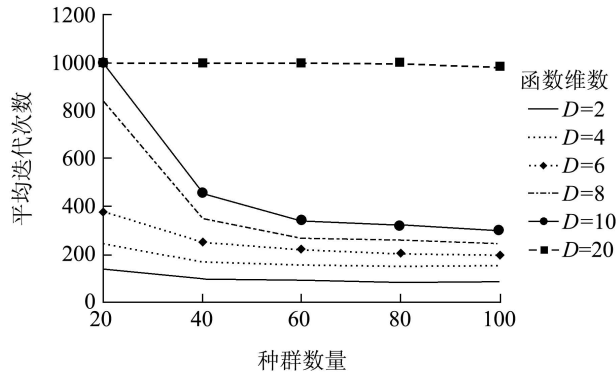


图3 矩阵操作方式Sphere函数的平均迭代次数

Fig. 3 Average iteration times for Sphere function on matrix operation

表1为种群数量为100时, Sphere函数的函数值计算结果, 可以看出, 两种不同操作方式下, 函数值的

计算结果相差不大.

表 1 种群数量为100时Sphere函数的函数值

Table 1 The fitness of Sphere function as swarm size is 100

| 维数 | 最小函数值 | | 平均函数值 | |
|----|----------|----------|----------|----------|
| | 矩阵操作方式 | 数组操作方式 | 矩阵操作方式 | 数组操作方式 |
| 2 | 2.74E-07 | 5.01E-07 | 4.29E-06 | 5.07E-06 |
| 4 | 2.28E-06 | 1.06E-06 | 7.72E-06 | 6.82E-06 |
| 6 | 2.90E-06 | 3.47E-06 | 7.84E-06 | 7.79E-06 |
| 8 | 5.88E-06 | 5.76E-06 | 8.74E-06 | 8.25E-06 |
| 10 | 5.45E-06 | 5.29E-06 | 8.63E-06 | 8.31E-06 |
| 20 | 9.99E-06 | 6.87E-06 | 1.73E-04 | 8.71E-06 |

基于上述计算结果,可以看出,对于Sphere函数,采用矩阵操作方式的粒子群算法能够取得更好的计算结果,即当随机性参数设置为相同值时,粒子群算法具有更好的优化性能.

4.1.2 Rosenbrock函数的计算结果(Calculation results of Rosenbrock function)

Rosenbrock函数如式(4)所示,也是一个单峰函数,当 $x_i = 1$,函数有全局唯一最小值 $f_2 = 0$. Rosenbrock函数是一个病态函数,极难最小化,寻优容易陷入局部解.

$$f_2 = \sum_{i=1}^D (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2). \quad (4)$$

Rosenbrock函数的计算结果如图4和图5所示,其中:图4为种群数量为20时,粒子群算法在矩阵操作方式和数组操作方式下,20次运行找到最优值的最小迭代次数和平均迭代次数;图5则是种群数量为100时的计算结果.

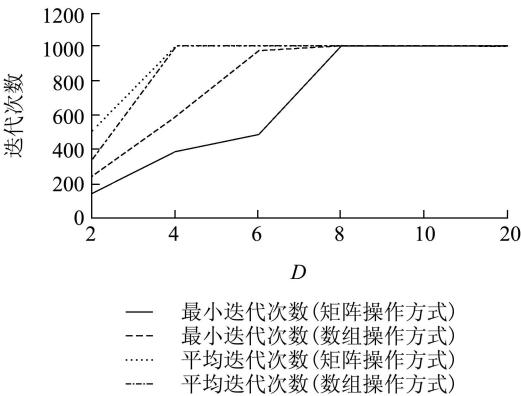


图 4 种群数量为20时Rosenbrock函数的计算结果
Fig. 4 Computational results of Rosenbrock function as swarm size is 20

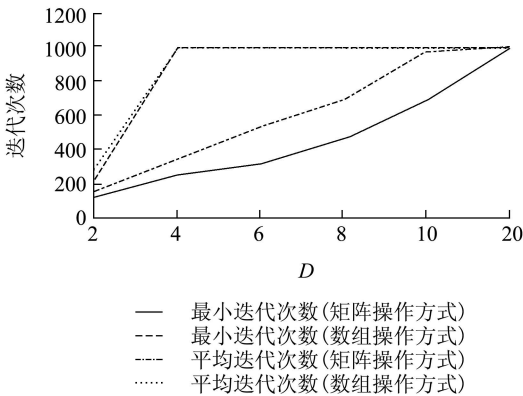


图 5 种群数量为100时Rosenbrock函数的计算结果
Fig. 5 Computational results of Rosenbrock function as swarm size is 100

从图4看,对于Rosenbrock函数,当函数维数 D 小于8时,在矩阵操作方式下,粒子群算法20次独立运行找到最优值的最小迭代次数和平均迭代次数均明显好于数组操作方式;当函数维数 D 大于或者等于8时,矩阵操作方式下粒子群算法的计算结果要比数组操作方式差.但是,从图5的计算结果看,当种群数量为100时,矩阵操作方式下粒子群算法的计算结果均比数组操作方式好.因此,随着种群数量的增加,矩阵操作方式下粒子群算法的优化性能得到更为明显的改善.

表2为种群数量为100时,Rosenbrock函数的函数值计算结果,可以看出,矩阵操作方式下粒子群算法找到的最小函数值和平均函数值均要好于数组操作方式.

表 2 种群数量为100时Rosenbrock函数的函数值
Table 2 The fitness of Rosenbrock function as swarm size is 100

| 维数 | 最小函数值 | | 平均函数值 | |
|----|----------|----------|----------|----------|
| | 矩阵操作方式 | 数组操作方式 | 矩阵操作方式 | 数组操作方式 |
| 2 | 6.58E-09 | 2.77E-07 | 5.90E-06 | 5.23E-06 |
| 4 | 1.00E-06 | 0.001196 | 0.185078 | 1.884827 |
| 6 | 3.65E-06 | 0.002952 | 0.993491 | 2.067699 |
| 8 | 7.10E-06 | 0.000630 | 0.798183 | 4.758523 |
| 10 | 9.58E-06 | 0.311682 | 2.800104 | 4.787292 |

结合上述分析,利用粒子群算法优化Rosenbrock函数时,采用矩阵操作方式的程序设计方法,即将随机数参数设置为相同值时,粒子群算法能够取得相对较好的优化结果.

4.1.3 Rastrigin函数的计算结果(Calculation results of Rastrigin function)

Rastrigin函数如式(5)所示,是一个多峰函数,当 $x_i = 0$,函数有全局唯一最小值 $f_3 = 0$,在全局最小

值周围有多个局部最小值.

$$f_3 = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10). \quad (5)$$

图6是种群数量为20时, 粒子群算法优化Rastrigin函数的计算结果, 而图7则是种群数量为100时的计算结果.

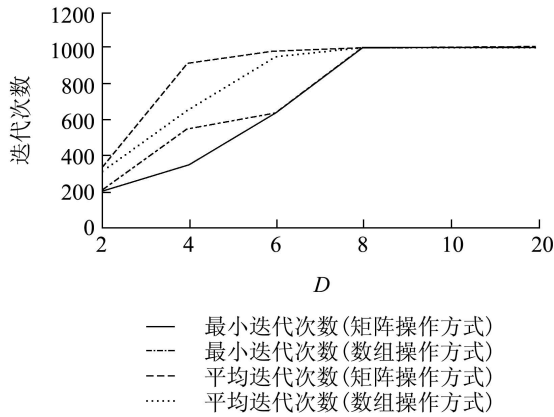


图6 种群数量为20时Rastrigin函数的计算结果
Fig. 6 Computational results of Rastrigin function as swarm size is 20

在图6中, 当函数维数 D 小于6时, 矩阵操作方式下找到最优值的最小迭代次数要好于数组操作方式, 但随着函数维数的增加, 这种优势就发生了逆转. 同时, 随着函数维数的增加, 矩阵操作方式下找到最优函数值的平均迭代次数普遍比数组操作方式差.

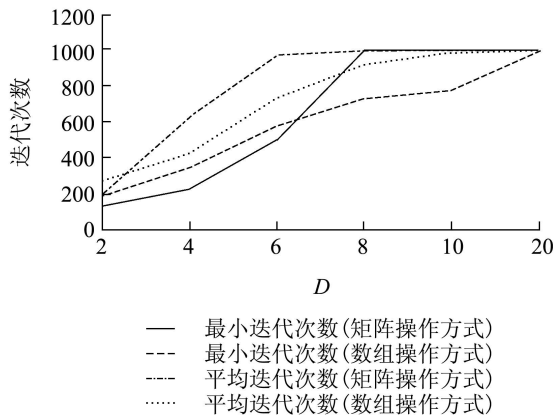


图7 种群数量为100时Rastrigin函数的计算结果
Fig. 7 Computational results of Rastrigin function as swarm size is 100

从图7看, 随着函数维数的增加, 种群数量的增加也不能有效改善矩阵操作方式下粒子群算法的优化结果, 反而使得数组操作方式下粒子群算法的优化性能得到改善.

从表3, Rastrigin函数的函数值计算结果看, 矩阵操作方式下的函数值普遍大于数组操作方式. 因此,

对于Rastrigin函数, 基于数组操作方式的粒子群算法具有相对更好的优化性能.

表3 种群数量为100时Rastrigin函数的函数值

Table 3 The fitness of Rastrigin function as swarm size is 100

| 维数 | 最小函数值 | | 平均函数值 | |
|----|----------|----------|----------|----------|
| | 矩阵操作方式 | 数组操作方式 | 矩阵操作方式 | 数组操作方式 |
| 2 | 7.39E-07 | 7.13E-07 | 4.64E-06 | 5.09E-06 |
| 4 | 2.12E-06 | 1.14E-06 | 4.48E-01 | 6.71E-06 |
| 6 | 9.31E-06 | 1.79E-06 | 2.69E+00 | 9.95E-02 |
| 8 | 9.95E-01 | 5.16E-06 | 5.72E+00 | 5.97E-01 |
| 10 | 3.98E+00 | 7.66E-06 | 1.15E+01 | 1.64E+00 |
| 20 | 2.19E+01 | 7.96E+00 | 6.46E+01 | 1.50E+01 |

综合上述几个连续函数的计算结果分析, 3个算例中, Sphere函数和Rosenbrock函数属于单峰函数, 求解复杂性要比Rastrigin多峰函数小, 因而, 在粒子群算法的程序实现过程中, 将随机数参数设置为相同值时, 粒子群算法能够具有更好的优化性能.

4.2 调度问题的计算结果(Calculation results of scheduling problem)

调度问题属于典型的离散空间优化问题, 前面已经针对连续空间优化问题中粒子群算法的随机数参数设置进行了分析, 因此, 这里再针对调度问题的粒子群算法随机数参数设置进行探讨.

这里以车间作业调度(job shop scheduling)问题为对象展开分析, 分别针对几个典型算例(benchmark problem)进行计算. 同样采用惯性权重线性递减粒子群算法, 粒子种群数量为40, 最大迭代次数为300. 每个算例, 粒子群算法均独立运行20次, 计算结果如表4所示. 其中, 表中最大偏差表示20次独立运行找到的最大函数值与该算例已知最优值之间的偏差, 最小偏差表示20次独立运行找到的最小函数值与该算例已知最优值之间的偏差, 平均偏差表示20次独立运行的平均函数值与该算例已知最优值之间的偏差.

job shop调度问题是一类比较复杂的组合优化问题, 其全局最优值附近有很多局部最优值, 也属于一类多峰函数. 从表4的计算结果看, 矩阵操作方式下的粒子群算法优化性能要比数组操作方式下的粒子群算法差. 进一步计算发现, 通过增加种群数量也不能改善矩阵操作方式下粒子群算法的计算结果. 因此, 对于job shop调度问题, 当随机数参数设置不同时, 粒子群算法的优化性能要好些.

表 4 车间作业调度问题的计算结果

Table 4 Computational results for job shop scheduling problem

| 问题 | 已知最优值 | 数组操作方式 | | | 矩阵操作方式 | | |
|------|-------|--------|--------|--------|--------|--------|--------|
| | | 最大偏差/% | 最小偏差/% | 平均偏差/% | 最大偏差/% | 最小偏差/% | 平均偏差/% |
| FT6 | 55 | 0 | 0 | 0 | 5.45 | 0 | 1 |
| FT10 | 930 | 22.8 | 6.45 | 13.46 | 27.31 | 16.67 | 20.85 |
| FT20 | 1165 | 21.97 | 8.93 | 14.55 | 28.67 | 14.16 | 21.88 |
| LA1 | 666 | 3.153 | 0 | 0.345 | 6.76 | 0 | 3.23 |
| LA6 | 926 | 0 | 0 | 0 | 2.7 | 0 | 0.42 |
| ABZ5 | 1234 | 10.373 | 3.728 | 7.164 | 14.42 | 7.13 | 10.66 |
| ABZ6 | 943 | 10.392 | 3.712 | 6.766 | 14.63 | 4.24 | 10.93 |

5 设备拥有量参数优化模型的计算结果
(Calculation results of equipment possession quantity parameter optimization model)

前面所述计算结果均是针对一些典型算例进行计算, 下面将针对设备拥有量计算模型这个实际应用算例, 对粒子群算法的随机数参数的设置问题进行计算和分析。

5.1 设备拥有量参数优化模型(Equipment possession quantity parameter optimization model)

在企业中, 企业所拥有的设备总数称为设备拥有量, 为完成生产任务而运行的设备台数称为设备出勤台数。设备拥有量的数量应该大于设备出勤台数, 但设备拥有量的数量不能太大, 否则就会造成设备的闲置和设备资金的浪费。同时, 在生产过程中, 并非所有设备都能保证100%的设备完好率, 由于设备维修和保养, 因而不能保证所有设备都能够100%出勤。

设备拥有量与设备出勤台数以及设备完好率之间存在一定的数学映射关系, 在一定设备拥有量数量和设备完好率的情况下, 能够唯一确定设备出勤台数的数量^[9]。即设备拥有量 n 与设备最佳出勤台数 m^* , 设备完好率 p 之间存在着——对应的单值函数关系 $n = f(m^*, p)$ 。假设设备拥有量的取值范围为 $n = \{21, 22, \dots, 89, 90\}$, 设备完好率的取值范围为 $p = \{0.6, 0.62, 0.64, \dots, 0.94, 0.96, 0.98\}$, 那么由上述——对应关系可以得到如表5所示的1420组 (n, m^*, p) 数据。

通过分析, 可以建立设备拥有量、设备出勤台数和设备完好率之间的数学关系, 即建立设备拥有量数学模型^[10], 如式(6)。

$$n = (1 + m^*)(1 + x_1 \cdot p + x_2 \cdot p^2 + x_3 \cdot p^3 + x_4 \cdot p^4 + x_5 \cdot p^5). \quad (6)$$

为了得到式(6)中的参数 $(x_1, x_2, x_3, x_4, x_5)$, 采用最小二乘法来建立设备拥有量的参数优化模型, 如式(7)所示:

$$f(x_1, x_2, x_3, x_4, x_5) = \sum_{k=1}^{1420} (n_k - (1 + m_k^*)(1 + x_1 \cdot p_k + x_2 \cdot p_k^2 + x_3 \cdot p_k^3 + x_4 \cdot p_k^4 + x_5 \cdot p_k^5))^2. \quad (7)$$

表 5 设备拥有量、完好率和出勤台数对应关系

Table 5 Relationship among equipment possession quantity, the ratio of equipment in good condition and equipment quantity on duty

| 设备完好率 $p/\%$ | 设备拥有量 $n/\text{台}$ | 设备最佳出勤台数 $m^*/\text{台}$ |
|-----------------|-----------------------|----------------------------|
| 60 | 20 | 10 |
| 60 | 21 | 10 |
| \vdots | \vdots | \vdots |
| 62 | 24 | 12 |
| 62 | 25 | 13 |
| \vdots | \vdots | \vdots |
| 98 | 89 | 84 |
| 98 | 90 | 85 |

5.2 优化结果(Optimization results)

仍然采用惯性权重线性递减粒子群算法, 此时函数的维数是一定的, 即 $D = 5$, 粒子种群数量 $\text{POPSIZE} = \{20, 40, 60\}$ 。当 $(x_1, x_2, x_3, x_4, x_5) = (22.1128, -78.7699, 110.5318, -71.4151, 17.5549)$, 式(7)有最小值276.88566508。不同种群数量下, 粒子群算法均独立运行20次, 计算结果如表6所示。

从表6的计算结果看, 无论是最小迭代次数和平均迭代次数, 还是优化后的最小函数值和平均函数值, 在矩阵操作方式下, 粒子群算法的优化

性能明显比数组方式好得多, 即当随机数参数设置相同时, 粒子群算法能够很有效地优化设备拥

有量模型. 在数组操作方式下, 粒子群算法却无法找到设备拥有量参数优化模型的最小值.

表6 设备拥有量优化模型的计算结果

Table 6 Computational results for equipment possession quantity optimization model

| 种群数量 | 最小迭代次数 | | 平均迭代次数 | | 最小函数值 | | 平均函数值 | |
|------|--------|--------|--------|--------|-------------|------------|-----------|------------|
| | 矩阵操作方式 | 数组操作方式 | 矩阵操作方式 | 数组操作方式 | 矩阵操作方式 | 数组操作方式 | 矩阵操作方式 | 数组操作方式 |
| 20 | 686 | 1000 | 902.55 | 1000 | 276.8856651 | 1057.83018 | 278.55661 | 3395.9298 |
| 40 | 580 | 1000 | 622.35 | 1000 | 276.8856651 | 1345.53615 | 276.88567 | 2069.91634 |
| 60 | 535 | 1000 | 571.95 | 1000 | 276.8856651 | 767.49423 | 276.88567 | 1689.75796 |

6 结论(Conclusion)

本文针对粒子群算法粒子速度更新公式中的随机数参数的设置问题展开分析, 在分析了不同程序设计语言结构所造成的粒子群算法的程序设计差异, 从而导致随机数参数在设置上产生差异. 并且针对随机数参数的设置问题, 对连续函数优化问题和调度问题中的几个典型算例以及设备拥有量参数优化模型采用粒子群算法进行优化计算, 计算结果说明对于部分优化问题, 对于同一粒子速度向量中的各个分量, 在速度更新公式中所对应的随机数设置相同时, 粒子群算法有着更好的优化性能, 因而, 随机数参数的设置直接影响着粒子群算法的优化性能和优化效率.

参考文献(References):

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization[C] // *Proceedings of International Conference on Neural Networks*. New York: IEEE, 1995: 1942 – 1948.
- [2] EBERHART R C, SHI Y H. Particle swarm optimization: development, applications and resources[C] // *Proceedings of the Congress on Evolutionary Computation*. Piscataway: IEEE, 2001: 81 – 86.
- [3] 巩敦卫, 张勇, 张建化, 等. 新型粒子群优化算法[J]. 控制理论与应用, 2008, 25(1): 111 – 114.
(GONG Dunwei, ZHANG yong, ZHANG Jianhua, et al. Novel particle swarm optimization algorithm[J]. *Control Theory & Applications*, 2008, 25(1): 111 – 114.)
- [4] 冯远静, 俞立, 冯祖仁. 采样粒子群优化模型及其动力学行为分析[J]. 控制理论与应用, 2009, 26(1): 28 – 34.

(FENG Yuanjing, YU li, FENG Zuren. Sample particle swarm optimization and its dynamic behavior[J]. *Control Theory & Applications*, 2009, 26(1): 28 – 34.)

- [5] SHI Y H, EBERHART R C. Parameter selection in particle swarm optimization[C] // *Proceeding of the 7th Annual Conference on Evolutionary Programming*. Berlin: Springer-Verlag, 1998: 591 – 600.
- [6] SHI Y H, EBERHART R C. Empirical study of particle swarm optimization[C] // *Proceedings of Congress on Evolutionary Computation*. Piscataway: IEEE, 1999: 1945 – 1950.
- [7] EBERHART R C, SHI Y H. Comparing inertia weights and constriction factors in particle swarm optimization[C] // *Proceedings of Congress on Evolutionary Computation*. New York: IEEE, 2000: 84 – 88.
- [8] 王维博, 林川, 郑永康. 粒子群算法中参数的实验与分析[J]. 西华大学学报(自然科学版), 2008, 27(1): 76 – 80.
(WANG Weibo, LIN Chuan, ZHENG Yongkang. Experiment and analysis of parameters in particle swarm optimization[J]. *Journal of Xihua University(Natural Science)*, 2008, 27(1): 76 – 80.)
- [9] 陈传诗, 李德源. 现代设备管理[M]. 北京: 人民交通出版社. 1988.
(CHEN Chuanshi, LI Deyuan. *Modern Equipment Management*[M]. Beijing: China Communications Press. 1988.)
- [10] 刘志雄. 设备投资决策模式和方法研究[D]. 武汉: 武汉理工大学. 1999.
(LIU Zhixiong. *Study of equipment investment decision mode and method*[D]. Wuhan: Wuhan University of Science and Technology, 1999.)

作者简介:

刘志雄 (1975—), 男, 副教授, 博士, 目前研究方向为生产调度及其智能优化方法, E-mail: lzx_brad@126.com;

梁华 (1975—), 女, 讲师, 目前研究方向为计算机集成制造, E-mail: ygxmonkey@126.com.