

前优化最重要决策变量的分组,这种方法是一种基于学习的协同进化策略^[10].前面介绍的算法均是使用协同进化策略将问题分解,减少问题复杂度的方法.还有不使用维数分组策略的算法,Zhao 在 2008 年将局部搜索策略加入算法中求解大规模问题,也取得了不错的结果^[11].2015 年,Cheng 等人提出了社会学习粒子群优化算法(social learning particle swarm optimization,简称 SLPSO)^[12]和竞争学习算法(competitive swarm optimizer,简称 CSO),也可以用于求解大规模问题^[13].2017 年,Omidvar 等人提出了 DG2 算法(differential grouping)用于解决大规模的黑盒问题,通过估计舍入误差的大小来找到可靠的阈值,在效率和精度方面,DG2 在最新的大型连续优化基准套件上胜过最先进的分解方法.DG2 在目标函数中决策变量的贡献不平衡的情况下也能可靠地执行^[14].文献[15]也提供了一种分治策略的新思路,将一个子问题看做是一个计算上昂贵的问题并通过元模型来解决,提出了一种新颖的方法,即自我评估演化(self-evaluation evolution,简称 SEE),并分析其优缺点.文献[16]中提出了一种将协作共同演化方法与各种类型的模拟算法相结合的异构框架,使用协同进化方法将大问题分解为子问题以提高解决过程的效率,然后使用各种异构模拟算法来解决子问题.所提出的异构框架为每个解决方案自适应地分配不同的运算符、参数值和本地搜索算法,以有效地探索和利用给定问题实例的搜索空间.

2 大规模优化问题描述

大规模优化问题可以用公式(1)表述:

$$\begin{aligned} \min/\max F(x) &= f(x_1, x_2, \dots, x_D) \\ x_i &\in [x_{\min}, x_{\max}], i = 1, 2, \dots, D \end{aligned} \quad (1)$$

其中, $F(x)$ 表示优化问题的目标函数,在单目标优化问题中,可以理解为一个实值连续非线性目标函数从 D 维空间映射到一维适应度值 $F(x)$ ^[14]. $x_i \in [x_{\min}, x_{\max}]$ 表示边界约束,虽然也是约束条件的一种,但是一般来讲,有边界约束的问题就是无约束优化问题^[15]. x_i 表示决策变量, D 表示决策变量的个数(即优化问题的维数),并且在大规模设置中,决策变量的个数 D 一般大于 100,通常达到 1 000 维以上.本文介绍的解决方案主要针对该问题中最小化问题进行求解.

3 基于协同进化的动态多种群粒子群优化算法

针对大规模优化问题,本文提出了一种基于协同进化策略的动态多种群粒子群优化算法.对维数的动态随机分组,不仅简化了问题的复杂度,并且兼顾了决策变量之间的相互关系.对种群的分组,增加了粒子的多样性,降低了算法的收敛速度,避免了算法陷入局部最优位置.

3.1 动态多种群粒子群优化算法

不同于其他已存在的进化算法,粒子群优化算法并不需要特别大的种群规模.特别是对于简单的问题,3 个~5 个粒子的种群就可达到满意的优化结果.粒子群优化算法中除了全局优化的版本,还有局部优化的版本.全局优化版本中,粒子向当前代的全局最优学习;局部优化版本和全局优化的版本有些相似,粒子一般不向全局最优解学习,向邻域最优解学习,这种方法的优点是可以避免算法收敛过快,陷入局部次优解,增加种群的多样性.因为局部优化的版本中粒子是向其邻域最优学习的,所以定义粒子的邻域范围比较重要.为了更好地确定邻域范围,引入拓扑结构是十分必要的.有了拓扑结构的引入,实验者可以根据拓扑结构确定邻域,以便更好地找到局部最优解.这种方法在多模态优化问题上有十分优秀的效率.

在动态多种群粒子群优化算法(dynamic multi-swarm particle swarm optimizer,简称 DMS-PSO)中,可以根据邻域结构把大种群分成很多小种群,增强算法的局部优化能力^[16].为了避免此算法不收敛,小种群的信息必须保证可以互相交换,因此,这些小的种群根据不同的重组策略经常被重组.在频繁重组的过程中,这些种群之间就在不断地交换信息.将重组周期设为 R .DMS 算法的搜索方式如图 1 所示.

如图 1 所示,假设种群规模为 9,将其随机分成 3 个小种群,每个小种群中有 3 个粒子.使用粒子群算法的局部版本对 3 个小种群进行优化.在这个周期中,9 个粒子可能会收敛到 3 个局部最优位置.然后对整个种群进行

重组,这些新的小种群再开始他们的搜索,这个过程一直持续到一个停止条件被满足.通过这种随机重组的方法,小种群中的粒子的搜索空间被扩展且这些小种群更容易找到一个较好的解.

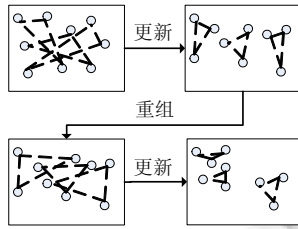


Fig.1 DMS-PSO's search
图 1 DMS-PSO 的搜索方法

3.2 动态随机协同进化策略

该策略的主要思想是将大规模复杂问题分解为一个单维或低维的简单问题解决.协同进化(cooperative coevolution,简称 CC)策略在早期阶段是静态分组,这种分组没有变化.但这种分组方法在解决不可分和部分可分的大规模问题上是无效的,而且算法的结果决定于最开始的分组情况,所以这种方法的性能非常不稳定.理论上可以将相关的决策变量分到一组,不相关的决策变量分开优化,但是在 1 000 个决策变量中,求解决策变量之间的相关性会花费巨大的评价次数和时间,并且还要保证求解必须十分准确,所以在早期求解决策变量的相关性的操作并不可行.因此,使用动态分组方法来改善这一不足之处.动态随机分组的协同进化机制如图 2 所示.

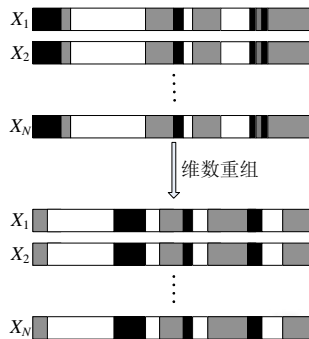


Fig.2 Mechanism of CC with dynamic random grouping
图 2 动态随机协同进化机制

如图 2 所示,种群 $pop(X_1, X_2, \dots, X_N)$,其中, N 为种群规模,首先将每个粒子的维数随机分组,例如问题维数为 20 维,随机分成 3 组,每组有 7 维,最后一组和第一组有 1 维重合,这样设计的好处是:没有对问题的维数做出硬性规定,可以测试多种维数的问题.然后对每组的维数进行更新.将种群中所有粒子的维数随机分成 3 组,如图 2 黑色部分、网状部分和白色部分所示,每组分别使用 DMS 算法进行优化,在优化时,每组迭代次数设为 $Max_oneiter$.所有的维数都优化过后,重新对维数进行随机分组,应当指出的是所有的决策变量都应该被优化.

基于协同进化策略的动态多种群粒子群优化算法的基本流程分为两部分:

首先利用协同进化策略将种群粒子的维数进行维数分组,优化问题的总维数是 D ,将维数随机分为 $group_D$ 个维数小组,每一维数小组有 Dim 维,每组覆盖的维数是 $Diml$,即 $D=group_D \times Dim$,当没有小组之间没有维数覆盖时, $Diml=0$.本文设计的协同进化策略是动态随机分组的,并没有对决策变量之间的相互关系进行求解,所以在设计维数分组策略时加上了维数之间的叠加,使维数连接更紧密.维数分组策略的伪代码见算法 1.

算法 1. 动态随机协同进化算法.

1. $pop(1:NP,1:D)$ //随机初始化种群

```

2. (best,best_val) //计算适应度值,找出当前最优解
3. While FES<0.95×Max_FES
4.   ind(1:D)←randperm(D)
5.   index(1:2D)←repmat(2D)
6.   For j←1 to group_D
7.     l←(j-1)×(Dim+Diml)+1
8.     U←j×(Dim+Diml)
9.     pop_D←pop(:index(l:u)) //对维数随机分组后的子部分
10.    pop(:index(l:u))←DMS-PSO(best,pop_D,Max_oneyter) //使用 DMS-PSO 对维数随机分组后的小
        组进行优化
11.    (best,best_val) //计算适应度值,找出当前最优解
12.  End for
13. End while
14. While FES≤Max_FES
15.   (best,best_val)←PSO(best,pop,Max_FES)
16. End while
17. 返回(best_val)

```

在上面的动态随机协同进化算法中,主要是对粒子的维数进行分组,要注意的是 $D \neq \text{group_D} \times \text{Dim}$,这样的设计是为了测试多种维数的问题.然后,对已经分好组的维数小组分别使用动态多种群粒子群优化算法进行优化,在优化每个维数小组时,需要将整个大种群随机分为 n 个小种群,每个小种群的种群规模为 m ,即 $N=n \times m$, R 表示种群重组的循环次数, Max_FES 表示最大评价次数, Max_oneyter 表示维数分组后每组优化的最大迭代次数.动态多种群粒子群优化算法的伪代码见算法 2.

算法 2. 动态多种群粒子群优化算法.

```

1. pop_D(1:N,1:Dim) //输入的维数分组后每组的种群
2. (best_D,best_val_D) //使用 gbest 代替其他没有被选中优化的维数,计算 pop_D 的最优解
3. gen=0
4. While gen≤max_oneyter
5.   index(1:N)←randperm(N)
6.   For i←1 to n
7.     l'←(i-1)×m+1
8.     u'←i×m
9.     pop_g←pop_D(index(l':u'),:) //将种群随机分成小种群
10.    找出 pop_g 的 lbest
11.    For d=1:Dim
12.       $v_{id}=w \times v_{id} + c_1 \times \text{rand}1_{id} \times (pbest_{id} - x_{id}) + c_2 \times \text{rand}2_{id} \times (lbest_{id} - x_{id})$ 
13.       $v_{id}=\min(\max(v_{id}, v_{i,\min}), v_{i,\max})$ 
14.       $x_{id}=x_{id} + v_{id}$ 
15.      If rand>0.5
16.         $x_{id}=\min(\max(x_{id}, x_{i,\min}), x_{i,\max})$ 
17.      Else
18.        If  $x_{id} < x_{i,\min}$ 
19.           $x_{id}=x_{i,\min} + (x_{i,\min} - x_{id})$ 

```

```

20.     Else
21.          $x_{id}=x_{i,max}-(x_{id}-x_{i,max})$ 
22.     End if
23. End if
24. End for
25.  $gen=gen+1$ 
26. If  $\text{mod}(gen,R)=0$ 
27.     对种群随机重新分组
28. End if
29. End for
30. End while

```

上面的 DMS 算法在进行粒子更新的时候使用的是粒子群优化算法的局部版本,如步骤 12.大规模优化问题的基准测试集中规定了搜索空间的上下限,所以粒子的位置不能超过边界 $[X_{min},X_{max}]$,而粒子的速度不能超过边界 $[V_{min},V_{max}]$.在本文的算法中,设定 $V_{max}=20\%X_{max}$.当粒子的速度超过速度边界时,选取边界作为更新速度,如步骤 13.当粒子位置超过位置边界时,有 50%的概率选择边界作为更新的位置,剩下 50%的概率超过边界的距离就返回相应的距离,如步骤 15~步骤 23.

4 实验对比与结果分析

实验选取了 DECCG,SLPSO,CSO 以及 DMS-PSO 和 DMS-CC 对比.和 DMS-PSO 对比,是为了验证新提出的策略的有效性.DECCG 在 IEEE Congress on Evolutionary Computation 中的大规模全局优化算法竞赛中取得了良好的结果,而 SLPSO 和 CSO 是最近提出的两个大规模优化算法.和这些算法对比,可以验证基于协同进化策略的动态多种群粒子群优化算法的有效性.5 个算法的参数设置如下.

- (1) DECCG:种群大小为 100,1 000 维,维数分为 10 组,每组 100 维,循环次数 $cycle=50$;
- (2) SLPSO:种群大小为 100, $\alpha=0.5,\beta=0.01$;
- (3) CSO:种群大小为 500, $\varphi=0.15$;
- (4) DMS-PSO:种群大小为 60, $c_1=c_2=1.49445,w=0.729$,种群分为 20 组,每组有 3 个粒子;
- (5) DMS-CC:种群大小为 60, $c_1=c_2=1.49445,w=0.729$,种群分为 20 组,每组有 3 个粒子,维数分为 20 组,循环次数 $Max_oneiter=100$.

上面 5 个算法分别独立运行 25 次,最大评价次数为 $3000 \times D$.表 1 给出了 5 种算法的测试结果,结果取 25 次的平均值.图 3~图 6 表示 5 种算法的 15 个测试函数的收敛图,数据取 25 次的中值.

Table 1 Experimental result

表 1 运行结果

| | DECCG | SLPSO | CSO | DMS-PSO | DMS-CC |
|-----|-------------------|-------------------|-------------------|-------------------|-------------------|
| F1 | 4.8300E-06 | 4.7800E-14 | 2.4300E-09 | 1.6627E+01 | 1.3200E+01 |
| F2 | 6.6329E+03 | 8.7723E+03 | 6.0063E+03 | 6.6366E+03 | 1.4263E+03 |
| F3 | 2.0242E+01 | 2.1589E+01 | 1.9758E+01 | 2.1403E+01 | 8.4118E+00 |
| F4 | 2.0608E+13 | 9.7449E+09 | 1.4464E+10 | 2.6721E+10 | 2.4621E+11 |
| F5 | 7.0373E+06 | 3.8196E+06 | 3.7196E+06 | 1.1432E+08 | 5.8818E+05 |
| F6 | 1.0758E+06 | 1.0602E+06 | 1.0775E+06 | 1.0500E+06 | 2.0852E+01 |
| F7 | 1.4473E+09 | 3.0408E+07 | 9.2726E+08 | 3.0853E+09 | 1.3103E+07 |
| F8 | 9.4303E+15 | 1.9075E+14 | 4.6186E+14 | 4.3620E+07 | 1.6031E+14 |
| F9 | 5.4912E+08 | 5.5739E+07 | 3.5489E+08 | 1.8714E+08 | 4.8545E+07 |
| F10 | 9.7208E+06 | 9.4291E+06 | 9.6856E+06 | 9.2307E+04 | 2.1981E+04 |
| F11 | 1.2198E+11 | 5.1581E+09 | 3.2510E+09 | 1.6834E+09 | 2.3038E+06 |
| F12 | 5.0887E+03 | 1.0395E+03 | 1.3087E+03 | 3.1743E+08 | 2.7298E+06 |
| F13 | - | - | 4.0182E+09 | 2.6749E+10 | 1.1481E+10 |
| F14 | - | - | 2.1343E+11 | 4.5106E+10 | 1.9970E+11 |
| F15 | 5.0887E+03 | 7.2472E+07 | 5.3411E+07 | 2.1614E+07 | 2.0878E+07 |

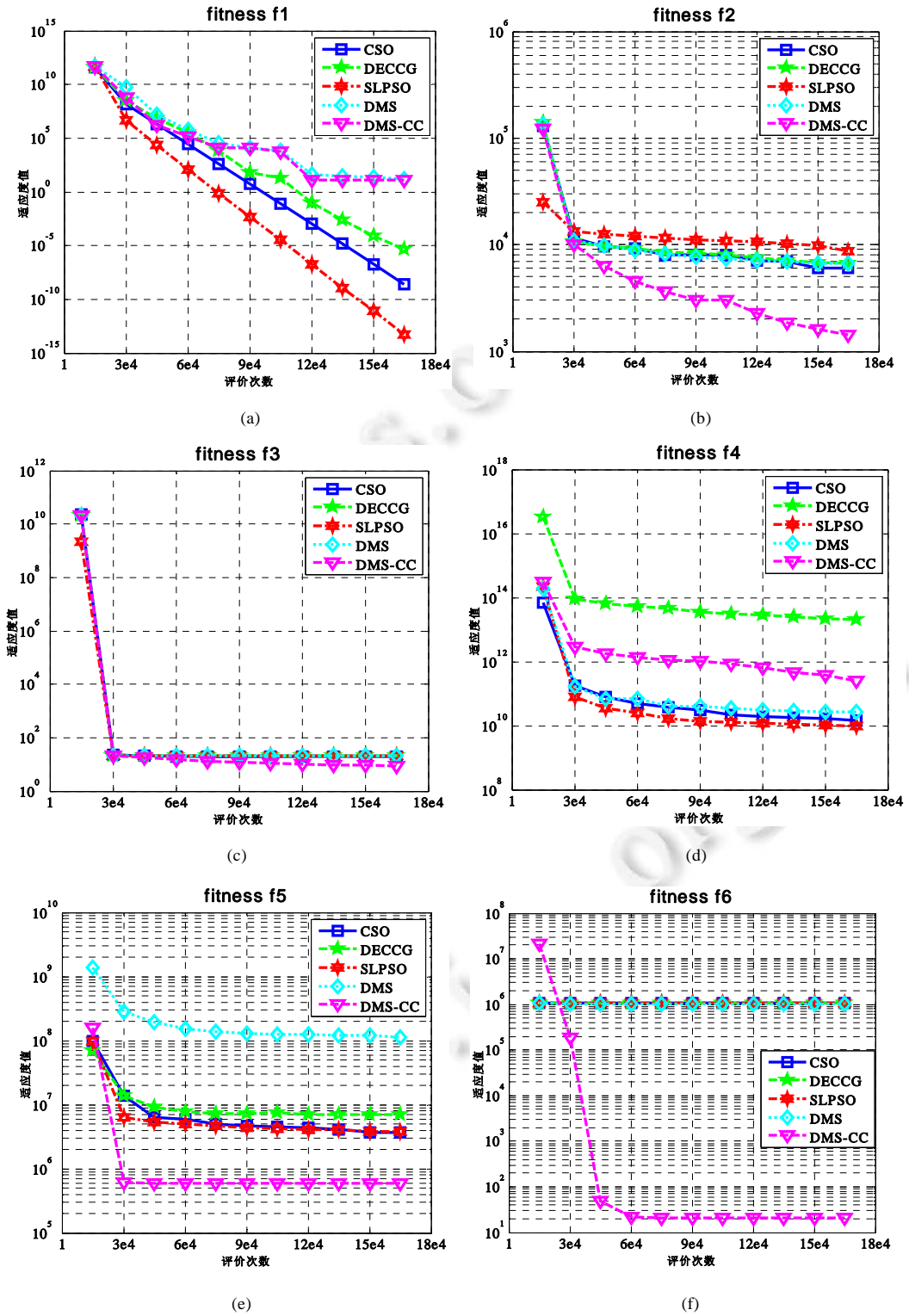


Fig.3 Convergence graph of test function F1~F6

图3 测试函数 F1~F6 的收敛图

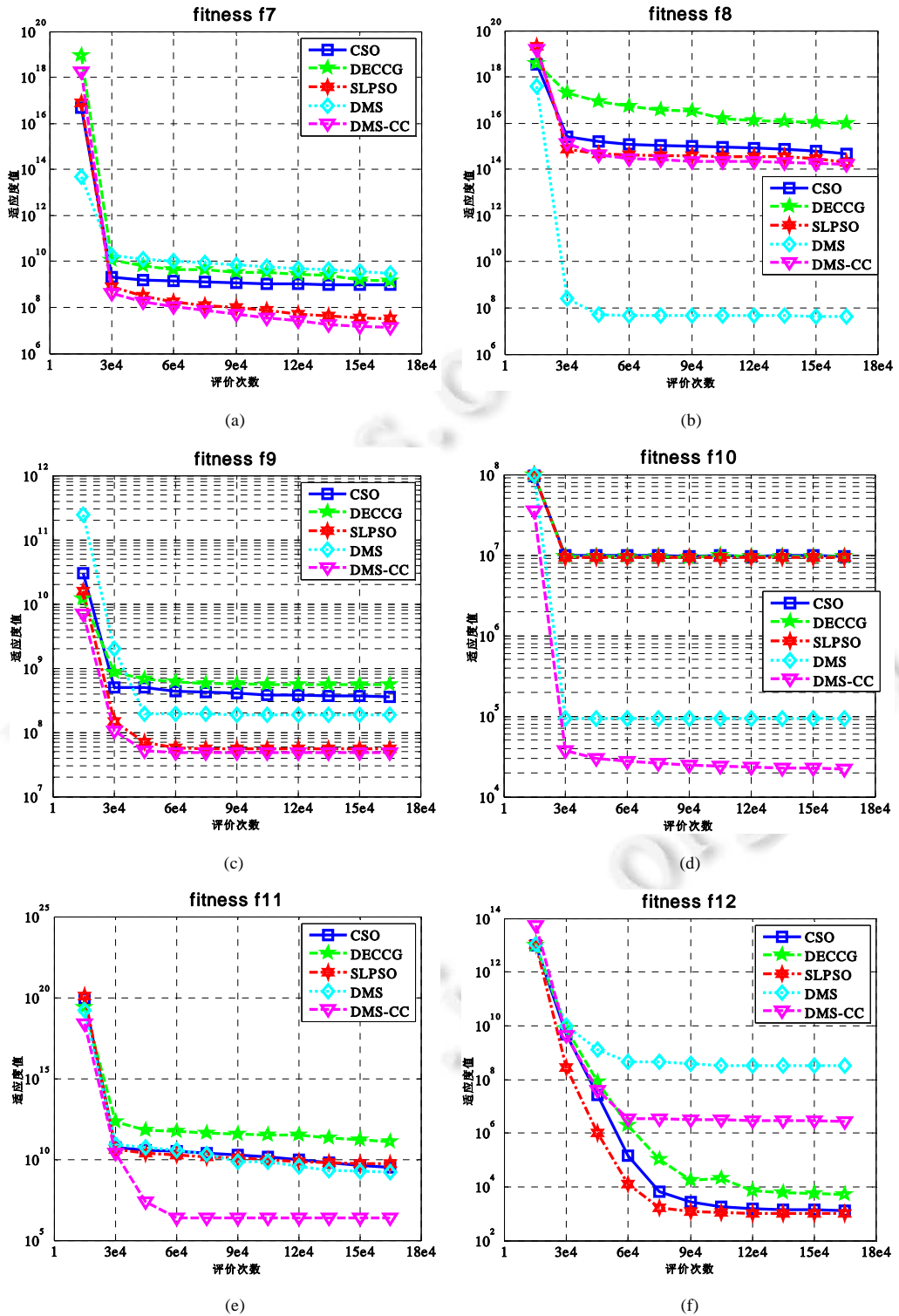


Fig.4 Convergence graph of test function F7~F12

图4 测试函数 F7~F12 的收敛图

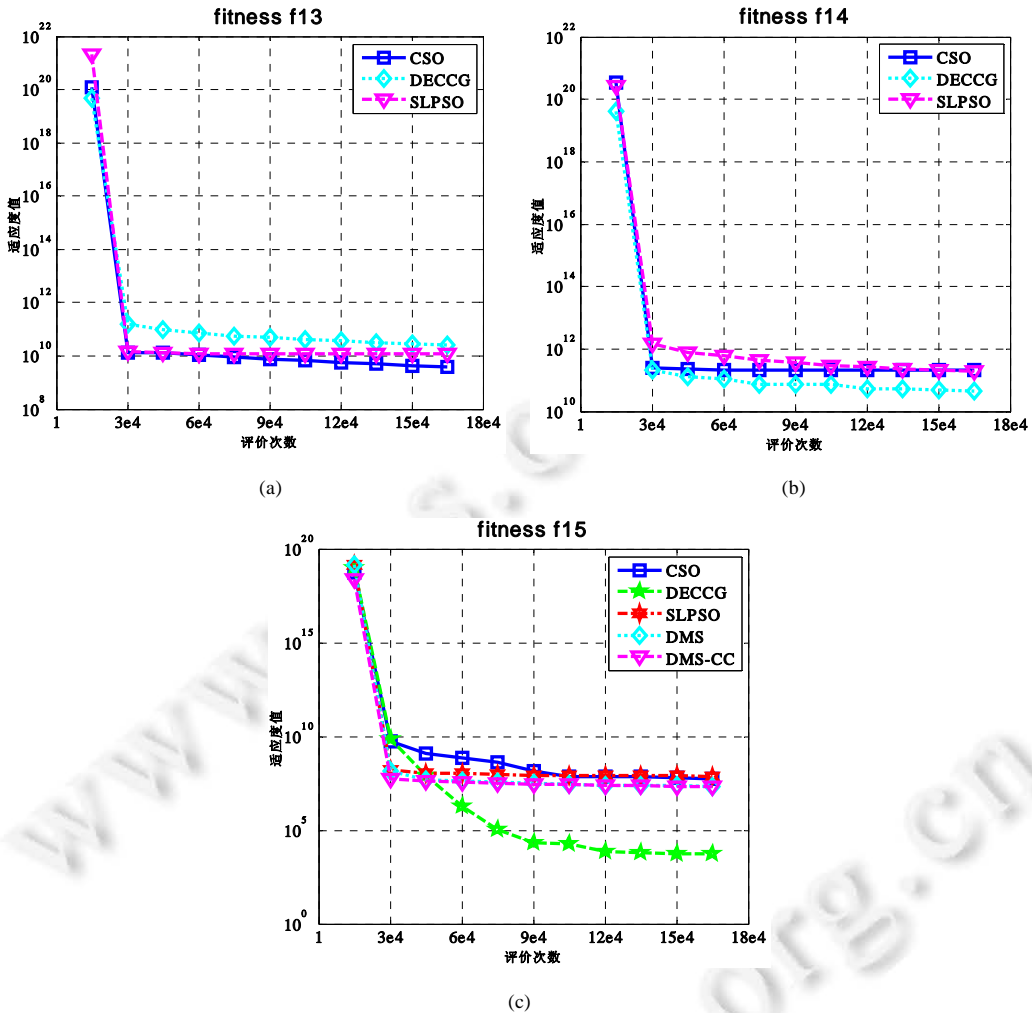


Fig.5 Convergence graph of test function F13~F15
图 5 测试函数 F13~F15 的收敛图

测试函数为 CEC2013 大规模全局优化算法测试函数集,该测试集一共有 15 个测试函数,包含了不可分、完全可分以及部分可分问题,同时包含单模和多模特性.在实验中,每个算法独立运行 25 次,算法的最大评价次数为 $3000 \times D$.除了函数 F13 和 F14 的维数设定的是 905,其他的测试函数维数设定为 1 000.因为算法 DECCG 只能优化 1 000 维的问题(分组是固定的),而算法 SLPSO 在选择学习目标时涉及到概率问题只能测试维数为 10 的倍数的测试函数,所以这两种算法均不能测试函数 F13 和 F14.

从结果我们可以看出 DMS-CC 算法在解决大规模优化问题时的有效性,其中,测试函数 F2、F3、F5、F6、F7、F9 和 F10 的结果比较好,上面的 7 个函数均为多模态问题,这是由于动态多种群的特点决定的,它将种群分成多个小种群,每个小种群在一定的搜索空间中搜索,不同于其他算法在一开始就向一个位置靠拢,在快速收敛的同时忽略了其他的位置,比较容易陷入局部最优位置.它具有更好的局部搜索能力,并且种群的多样性比较好.在测试函数 F4 中,由于 DECCG 和 DMS-CC 算法都是使用了维数分组的协同进化策略,测试函数中具有相关性的维数较多,所以具有协同进化策略的算法普遍没有一般算法的结果好.结果比较不理想的是测试函数 F1,该测试函数的决策变量之间并没有相关性并且是单模态问题,所以在评价次数一定的条件下,算法收敛能力强的算法结果比较好,由于动态多种群策略的局部搜索能力强,牺牲了部分全局搜索能力,使得算法的收敛能力

不强,所以算法 DMS-PSO 和 DMS-CC 的测试结果不好.在今后的工作中,可以加上收敛能力较强的一些全局优化策略.

5 总结与展望

本文针对大规模问题的高维特性,提出了一种有效的分组策略.将大规模复杂问题分解为一个个单维或低维的简单问题来解决,是近些年的常见策略.根据这个思想,论文提出了将动态随机分组的协同进化机制嵌入种群分组的动态多种群算法中,实现维数和种群的双分组.动态维数分组可以解决决策变量多且变量之间相互关联的问题,而动态多种群则可以解决多模态和算法收敛过快陷入局部最优的问题.最后,将论文中提出的算法和其他算法在基准测试函数集测试进行对比实验,验证结果的有效性.但是还存在如下一些问题,有待在未来的工作中得到解决.

- (1) 参数的调节:新算法使用的参数是固定的,并没有对参数进行深入讨论,增加参数的自适应机制,减小算法的复杂度;
- (2) 基于学习的协同进化策略:新算法使用了随机动态的协同进化策略,没有解决决策变量之间的相互关系.有学者提出了基于学习的协同进化策略,即:在算法的前期先判断决策变量之间的相互关系,将相关的变量分到一组.但是引入学习机制会花费巨大的评价次数,减少算法在后期优化过程中的评价次数,所以平衡前期求决策变量之间相关性的工作花费,在后期有待研究;
- (3) 实际应用:本文提出的算法在测试平台上有不错的结果,但是在实际应用中的结果并没有验证,所以算法在大规模流水线加工等实际优化问题的效果有待验证.

References:

- [1] Mahdavi S, Shiri ME, Rahnamayan S. Metaheuristics in large-scale global continues optimization: A survey. *Information Sciences*, 2015,295:407–428.
- [2] Liu JP. CMA-ES and decomposition strategy for large scale continuous optimization problem [Ph.D. Thesis]. Hefei: University of Science and Technology of China, 2014 (in Chinese with english abstract).
- [3] Potter MA. The design and analysis of a computational model of cooperative coevolution. George Mason University, 1997.
- [4] Omidvar MN, Li X, Mei Y, *et al.* Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Trans. on Evolutionary Computation*, 2014,18(3):378–393.
- [5] Potter MA, Jong KAD. A cooperative coevolutionary approach to function optimization. In: *Proc. of the Int'l Conf. on Parallel Problem Solving from Nature*. Berlin, Heidelberg: Springer-Verlag, 1994. 249–257.
- [6] Frans VDB, Engelbrecht AP. A cooperative approach to particle swarm optimization. *IEEE Trans. on Evolutionary Computation*, 2004,8(3):225–239.
- [7] Yang Z, Tang K, Yao X. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 2008,178(15): 2985–2999.
- [8] Li XD, Yao X. Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms. In: *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2009)*. IEEE, 2009. 1546–1553.
- [9] Li XD, Yao X. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans. on Evolutionary Computation*, 2012,16(2):210–224.
- [10] Mahdavi S, Rahnamayan S, Shiri ME. Multilevel framework for large-scale global optimization. In: *Proc. of the Soft Computing*. 2016. 1–30.
- [11] Zhao SZ, Liang JJ, Suganthan PN, *et al.* Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In: *Proc. of the IEEE World Congress on Computational Intelligence, IEEE Congress on Evolutionary Computation (CEC 2008)*. IEEE, 2008. 3845–3852.
- [12] Cheng R, Jin Y. A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences*, 2015, 291(6):43–60.

- [13] Cheng R, Jin Y. A competitive swarm optimizer for large scale optimization. *IEEE Trans. on Cybernetics*, 2015,45(2):191.
- [14] Omidvar MN, Yang M, Mei Y, *et al.* DG2: A faster and more accurate differential grouping for large-scale black-box optimization. *IEEE Trans. on Evolutionary Computation*, 2017,21(6):929–942.
- [15] Yang P, Tang K, Yao X. Turning high-dimensional optimization into computationally expensive optimization. *IEEE Trans. on Evolutionary Computation*, 2018,22(1):143–156.
- [16] Sabar NR, Abawajy J, Yearwood J. Heterogeneous cooperative co-evolution memetic differential evolution algorithm for big data optimization problems. *IEEE Trans. on Evolutionary Computation*, 2017,21(2):315–327.
- [17] Liang J, Song H, Qu BY, *et al.* Research on path planning problems based on improved particle swarm optimizer. *Journal of Zhengzhou University (Engineering Science)*, 2014,35(1):34–38 (in Chinese with English abstract).
- [18] Yan XF, Yan JJ, Feng Y. Gradient and particle swarm optimization based hierarchical cluster algorithm in WSN. *Journal of Zhengzhou University (Engineering Science)*, 2016,37(2):33–36 (in Chinese with English abstract).
- [19] Liang JJ, Suganthan PN. Dynamic multi-swarm particle swarm optimizer with local search. In: *Proc. of the 2005 IEEE Congress on Evolutionary Computation*. IEEE, 2005. 522–528.

附中文参考文献:

- [2] 刘金鹏.面向大规模实值优化问题的 CMA-ES 算法及其分制策略研究[博士学位论文].合肥:中国科学技术大学,2014.
- [17] 梁静,宋慧,瞿博阳,等.基于改进粒子群算法的路径优化问题研究. *郑州大学学报(工学版)*,2014,35(1):34–38.
- [18] 阎新芳,严晶晶,冯岩.WSN 中基于梯度和粒子群优化算法的分级簇算法. *郑州大学学报(工学版)*,2016,37(2):33–36.



梁静(1981—),女,河南兰考人,博士,教授,博士生导师,主要研究领域为进化计算,群集智能,多目标优化,模式识别.



于坤杰(1990—),男,博士,副教授,主要研究领域为进化计算,多目标优化,约束优化,复杂工业过程建模与优化.



刘睿(1991—),女,硕士,主要研究领域为优化算法.



瞿博阳(1984—),男,博士,副教授,主要研究领域为智能优化算法,多目标优化算法,小生境优化算法及其应用.