

简单高效耦合策略的粒子群混合算法

李文锋^{1†}, 曹玉莲¹, 张汉²

(1. 武汉理工大学 物流工程学院, 湖北 武汉 430063;

2. 卡尔斯鲁厄理工学院 核能技术研究所, 德国 Eggenstein-Leopoldshafen 76344)

摘要: 建立了评判耦合策略优劣的定量分析方法, 发现了现有带中间启动局部搜索(local search, LS)的粒子群混合算法的不足, 进而提出一种简单高效的耦合策略. 基于该策略, 在全局性能优异的综合学习粒子群(comprehensive learning particle swarm optimizer, CLPSO)算法中引入具有快速收敛性能的传统LS方法, 提出了带LS的CLPSO混合算法(CLPSO hybrid algorithm with LS, CLPSO-LS). 以10维、30维和50维的11个标准函数, 对基于不同LS方法的4种混合算法的性能进行大量测试. 结果表明, 4种CLPSO-LS混合算法的性能均优于CLPSO算法, 验证了混合算法的有效性. 其中, 基于BFGS拟牛顿方法的混合算法的综合性能最优. 最后, 与8种先进粒子群算法的对比, 结果表明CLPSO-LS混合算法作为一种改进CLPSO算法, 其性能优于包括已有CLPSO改进算法在内的对比算法, 进一步验证了其优越性.

关键词: 定量分析; 耦合策略; 局部搜索; 粒子群优化

引用格式: 李文锋, 曹玉莲, 张汉. 简单高效耦合策略的粒子群混合算法. 控制理论与应用, 2018, 35(1): 13–23

中图分类号: TP182; TP391 **文献标识码:** A

Hybrid particle swarm optimization algorithm with simple and efficient coupling strategy

LI Wen-feng^{1†}, CAO Yu-lian¹, ZHANG Han²

(1. School of Logistics Engineering, Wuhan University of Technology, Wuhan Hubei 430063, China;

2. Institute for Nuclear and Energy Technologies, Karlsruhe Institute of Technology, Eggenstein-Leopoldshafen 76344, Germany)

Abstract: A quantitative analysis method is established to judge the pros and cons of the coupling strategy. The shortcomings of the existing hybrid particle swarm optimization algorithm with intermediate starting local search (LS) are discovered. And then a simple and efficient coupling strategy is proposed. Based on this strategy, the traditional LS method with fast convergence performance is introduced into the comprehensive learning particle swarm optimizer (CLPSO) algorithm. Then the CLPSO hybrid algorithm with LS (CLPSO-LS) is proposed. Numerous experiments are carried out to test the performance of the four different LS methods based hybrid algorithms on 10-dimensional, 30-dimensional and 50-dimensional problems of eleven benchmark functions. The results show that the performance of the four CLPSO-LS algorithms is superior to that of CLPSO algorithm, which verifies the validity of the hybrid algorithms. Among them, the performance of the BFGS quasi-Newton method based hybrid algorithm is the best. Finally, comparison results with eight advanced particle swarm optimization algorithms demonstrate that the performance of the CLPSO-LS algorithm as an improved CLPSO algorithm is superior to the compared algorithms including existing improved CLPSO algorithms, which further validates the superiority of the CLPSO-LS algorithm.

Key words: quantitative analysis; coupling strategy; local search; particle swarm optimization (PSO)

Citation: LI Wenfeng, CAO Yulian, ZHANG Han. Hybrid particle swarm optimization algorithm with simple and efficient coupling strategy. *Control Theory & Applications*, 2018, 35(1): 13–23

1 引言(Introduction)

粒子群优化(particle swarm optimization, PSO)算法是一种基于群体的元启发式进化算法^[1], 因其参数

设置少、可并行计算和寻优性能强等优点, 已被广泛应用于调度优化^[2–3]、神经网络训练^[4]和机器人控制^[5]等领域. 实际应用中, 在保持PSO算法的全局寻

收稿日期: 2017–03–17; 录用日期: 2017–09–27.

[†]通信作者. E-mail: liwf@whut.edu.cn; Tel.: +86 13986292609.

本文责任编辑: 冯祖仁.

国家自然科学基金项目(61571336, 71372202, 71672137), 河南省重大专项(151100211400)资助.

Supported by the National Natural Science Foundation of China (61571336, 71372202, 71672137) and the Major Projects of Henan Province (151100211400).

优能力的同时,需提高计算效率,而收敛速度是其关键所在。

基本PSO算法在求解多峰问题时,存在“早熟”和易陷入局部极值的难题。针对这些问题,出现了多种改进PSO算法^[6-11]。国内学者们也进行了大量PSO算法改进的研究^[12-14]。综合学习粒子群算法(comprehensive learning particle swarm optimizer, CLPSO)是其中较为成功的PSO改进算法^[11],自被提出以来,相关的理论研究^[15-17]和应用研究^[18]一直持续至今。

现有的CLPSO改进算法大多是通过改进粒子的学习策略实现的,要么通过维持种群多样性来强化探索能力,要么强化局部开发能力,但往往不能兼顾两者,甚至是以牺牲全局性能来提升局部性能,或者相反为之。针对这一难题,本研究采用传统的局部搜索方法(local search, LS)来改进CLPSO,在保持全局优势的同时,提高算法的局部收敛速度。

文献报道中已有PSO算法与LS方法结合的研究^[19-23],表明两者的混合具有重要意义。文献[19]每隔固定迭代次数,就选择部分粒子的历史最优解(记为 p_{best})执行LS。文献[20]每次迭代都对所有粒子执行LS。文献[21]中指出,如果算法本身的全局能力弱,需要高频率使用LS。以这些为代表的传统耦合策略的前提假设是认为LS具有全局加速的作用,其依据是采用LS后能得到优于当前值的“局部最优解”,影响粒子速度的更新,从而促进全局搜索。但是LS是否真的具有全局加速作用尚未被定量分析。因此,如何定量分析LS的作用,进而建立合理的耦合策略是运用该混合方法的关键。

本文建立了基于进入全局最优盆(global optimum basin, GOB)粒子比例的定量分析方法,用于评判耦合策略的好坏,通过发现中间启动LS的耦合策略的混合算法的不足,进而提出了一种基于新型耦合策略的CLPSO-LS混合算法。该策略简单高效,能充分实现CLPSO的全局能力和LS的局部收敛优势的互补。

文章组织结构如下:第2节介绍CLPSO-LS混合算法,第3节介绍标准测试函数,第4节的数值实验分析包括收敛精度分析、4种混合算法的排名分析和与其他先进PSO算法的对比分析。

2 CLPSO-LS 混合算法(CLPSO-LS hybrid algorithm)

2.1 基础知识(Fundamental knowledge)

PSO算法模拟鸟群觅食过程中个体之间的协作行为,每个粒子用速度 $v = \{v_1, v_2, \dots, v_d\}$ 和位置 $x = \{x_1, x_2, \dots, x_d\}$ 来刻画, d 代表维度。粒子记忆自身的历史位置信息,并分享当前位置信息给其他个体。 ω PSO通过惯性权重 ω_i 调节算法的全局和局部搜索能力。按照式(1)更新速度。其中: r_1, r_2 是服从均匀分布的随机数, c_1, c_2 是学习因子, i 代表粒子编号, t 是当前迭代次数, p_{best} 是个体的历史最优解, g_{best} 代表种群

最优解。

$$v_{id}^{t+1} = \omega_i \cdot v_{id}^t + c_1 \cdot r_1 \cdot (p_{bestid}^t - x_{id}^t) + c_2 \cdot r_2 \cdot (g_{best}^t - x_{id}^t). \quad (1)$$

CLPSO是基于 ω PSO改进的分维度综合学习算法^[11],其速度按照公式(2)更新。与 ω PSO不同的是,其 p_{best} 可能是种群中任意其他个体的,而不仅仅是自身的,并且同一个粒子的不同维度可能选择不同粒子的 p_{best} 。此外,CLPSO算法没有向 g_{best} 学习项。这样能有效地保证种群的多样性,探索更大的搜索空间,进而增强算法的全局搜索能力。

$$v_{id}^{t+1} = \omega_i \cdot v_{id}^t + c \cdot r_{fi(d)d}^t \cdot (p_{fi(d)d}^t - x_{id}^t), \quad (2)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \quad (3)$$

其中: $fi(d)$ 表示每个粒子各维度选择的学习对象的粒子编号。当 g_{best} 连续 m 次没有更新时,重新以锦标赛策略选择 $fi(d)$ 。粒子的位置按式(3)更新。

作为一种启发式方法,CLPSO具有很强的全局搜索能力,但针对最后阶段的开发过程相对于传统LS方法并不占优势。本文选取的4种传统LS方法,具有收敛速度快的特点,因而可以与全局搜索性能强的CLPSO算法形成优势互补。它们分别是最速下降法(steepest descent method)、DFP(Davidon-Fletcher-Powell)拟牛顿法、BFGS(Broyden-Fletcher-Goldfarb-Shanno)拟牛顿法和单纯形搜索法(Nelder-Mead)^[11]。

引入具有快速收敛性能的传统LS方法来改进全局性能优异的CLPSO算法,理论上能够进一步提高CLPSO算法的优化效率。但是,两种方法之间耦合策略的合理与否将影响到混合算法的性能。

2.2 基于RPGOB的耦合策略定量评价方法(Quantitative evaluation method for coupling strategy based on RPGOB)

现有的混合算法中常用的耦合策略是中间每隔一定的间隔代数就执行LS。这种策略的出发点是认为通过局部搜索之后得到的更优的 g_{best} 和 p_{best} 。对于PSO的全局搜索有促进作用^[21]。这是一种定性的认识,然而,中间加LS的策略是否真的具有全局加速作用?采用何种指标刻画全局收敛性能?这些问题一直没有得到澄清和有利的理论支持。本文提出一种定量方法分析这些问题。

定义进入GOB的粒子比例(ratio of particles in global optimum basin, RPGOB)这一指标,定量地评价混合算法的全局搜索速度。其合理性在于PSO算法分为前期全局探索阶段和后期局部开发阶段,而进入GOB正是两个阶段的分割点。以常用的标准多峰函数Rastrigin(记为 f_5)和不连续的Rastrigin函数(记为 f_6)为例具体说明,其全局最优解的位置已知(在 $x^* = [0, 0, \dots, 0]$ 时取得),通过导数分析,在全局最优解附近的连续区域即为GOB。计算得出函数 f_5 和 f_6 的GOB

区间均为 $[-0.49, 0.49]^D$. 则 RPGOB 指标就是进入 GOB 的粒子数与种群个体总数的比值. 在相同的计算代价下, RPGOB 越大表示对应算法的全局搜索速度越快.

运用 RPGOB 指标评价传统混合算法中采用的每隔 L 代执行一次 LS 的耦合策略的性能优劣, 包括加速全局搜索和算法收敛性能两方面. 首先, 利用基于 RPGOB 的定量分析方法讨论每隔 L 代启动 LS 是否对全局搜索有加速作用. 以 10 维的函数 f_5 测试每隔 30 代、40 代、50 代和 60 代启动 LS, 以及中间不启动 LS 的混合算法的全局搜索性能, 每次实验独立运行 25 次, 统计每次实验的 RPGOB 指标随迭代次数的变化. 为去除因 LS 占用计算资源导致全局搜索不充分而产生的影响因素, 忽略 LS 消耗的计算量, 得到每隔不同的间隔代数 L 执行一次 LS 的混合算法的 RPGOB 随迭代次数的变化曲线(图1). 图中: $L30$, $L40$, $L50$ 和 $L60$ 分别代表每隔 30, 40, 50 和 60 代执行一次 LS 的混合算法, 而不加 LS 代表的是中间不启动 LS 的混合算法. 不难看出, 不同间隔代数下的混合算法的 RPGOB 曲线几乎吻合, 只有中间不启动 LS 的算法稍微快一点. 这说明, 即使不计 LS 的计算代价, 中间启动的 LS 也不具有全局加速作用.

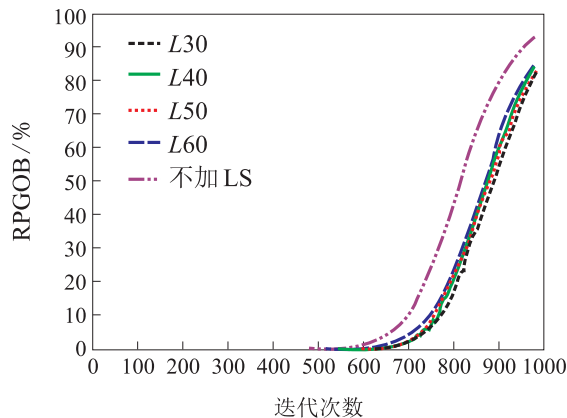


图 1 忽略 LS 计算消耗的每隔不同 L 代执行 LS 的混合算法对函数 f_5 的 RPGOB

Fig. 1 RPGOB of hybrid algorithms with LS by different L interval iterations without considering the FEs for LS on f_5

然而, 事实上 LS 会占用一定的计算量, 图2为考虑 LS 计算消耗, 每隔不同 L 代执行一次 LS 的混合算法的 RPGOB 随函数评价次数(function evaluations, FEs)的变化曲线. 对横轴在区间 $[30000, 40000]$ 、纵轴在区间 $[0, 16]$ 的部分进行了局部放大. 可以明显看出, 中间不启动 LS 的混合算法快于任何一种每隔不同 L 代启动 LS 的混合算法. 由此可见, 中间启动 LS 不但没有起到全局加速的作用, 反而因为 LS 占用计算资源, 降低了全局搜索速度.

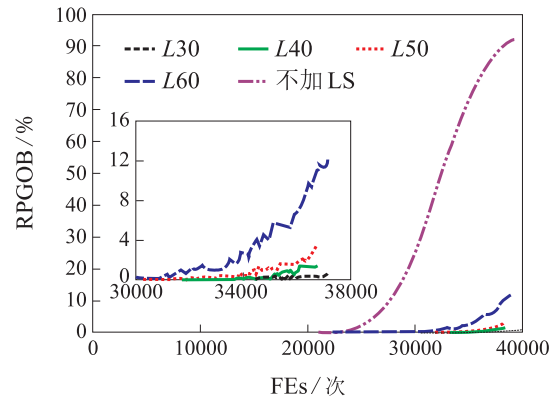


图 2 考虑 LS 计算消耗的每隔不同 L 代执行 LS 的混合算法对函数 f_5 的 RPGOB

Fig. 2 RPGOB of hybrid algorithms with LS by different L interval iterations with considering the FEs for LS on f_5

进一步讨论中间启动 LS 的耦合策略对算法收敛性能的影响. 对表1中的6个多峰函数的10维问题测试每隔30代、40代、50代和60代启动 LS 的混合算法的收敛性能, 每次实验独立运行 25 次. 表1是在 CLPSO 的迭代过程中, 每隔不同 L 代启动 LS 得到的混合算法的收敛失败率, 设定精度值大于 $1e-2$ 时视为未收敛. 从表1可知, L 越小, 即 LS 启动越频繁, 失败率越高. 这是因为频繁启动 LS 会使多个粒子聚集到同一个区域, 导致种群多样性的丧失, 引起“早熟”. 由此可见, 在 CLPSO 迭代过程中频繁启动 LS 的策略会增加算法陷入局部极小的风险.

表 1 每隔 L 代执行中间 LS 的混合算法的失败率

Table 1 Failure rate of hybrid algorithm with intermediate LS by different L interval iterations

函数	$L = 30$	$L = 40$	$L = 50$	$L = 60$
Griewanks	0.12	0.08	0.04	0.04
Rastrigin	0.76	0.6	0.24	0
不连续的Rastrigin	0.96	0.72	0.44	0.16
Weierstrass	1	1	0.48	0
偏移的Rastrigin	0.48	0	0	0
偏移不连续的Rastrigin	0.28	0.08	0	0

综上, 通过基于定量评价方法的分析, 发现传统中间启动 LS 的耦合策略并不具有加速全局搜索的作用, 反而因为 LS 占用计算资源, 使全局搜索不充分, 增加算法陷入局部极小的风险.

2.3 CLPSO-LS 混合算法简单高效的耦合策略 (CLPSO-LS hybrid algorithm with simple and efficient coupling strategy)

针对基于中间启动 LS 耦合策略下的混合算法的不足, 本文提出一种更合理的耦合策略, 即在 CLPSO 已

经进行了大量全局搜索之后,只在迭代最后对 g_{best} 执行一次LS.它不仅克服了原有耦合策略易陷入局部极小的不足,也在节约计算资源的前提下实现加速收敛的作用,简单且高效.图3是以BFGS拟牛顿方法为例的CLPSO-LS混合算法的流程.基于BFGS拟牛顿法、DFP拟牛顿法、最速下降法和单纯形搜索的CLPSO-LS混合算法分别简记为CLbfgs, CLdfp, CLsd和CLnelder.

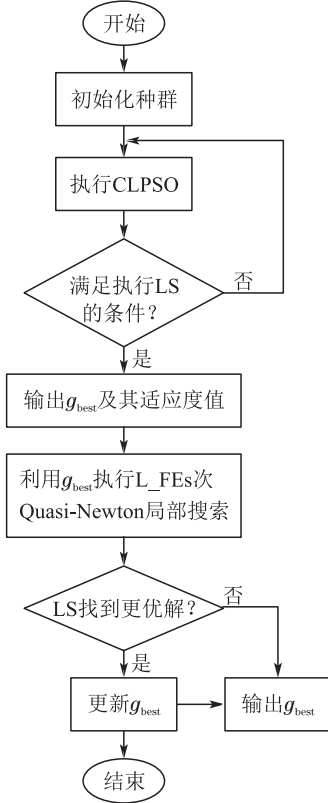


图3 CLPSO-LS混合算法的流程图

Fig. 3 The flowchart of CLPSO-LS hybrid algorithm

基于简单高效耦合策略的CLPSO-LS混合算法的基本思想是将CLPSO的全局搜索优势和LS的快速精确求解能力强结合.为了避免由于频繁启动LS导致“早熟”,同时更合理分配全局搜索和局部搜索的计算资源,不在中间启动LS.本文的策略能在保持CLPSO的全局性能优势的前提下,提高混合算法的局部收敛速度.在实际计算中,考虑到计算时间往往是有限的,在这种情况下,本文的策略可修正为在达到最大迭代次数之后,再启动LS对 g_{best} 进行搜索.

该策略一方面能在给定的实验条件下,最大限度地保持CLPSO的全局性能,另一方面是其简单且高效.它是基于对现有中间启动LS的耦合策略的混合算法中LS作用的分析提出的,具有很强的理论依据,后面将通过实验测试本文耦合策略下的CLPSO-LS混合算法的性能.

2.4 收敛性分析(Convergence analysis)

与一般的PSO类似,CLPSO-LS混合算法的搜索过程同样分为早期全局探索和后期局部开发.根据文献[11]对CLPSO的研究结果,已经充分说明了CLPSO具有强大的全局探索能力.本节只需针对CLPSO的后期局部开发的收敛速度进行理论分析,并与以DFP拟牛顿方法为代表的LS进行对比.

根据文献[25], PSO收敛速度的分析一般基于以下几点简化: 1) 一维问题, 2) 单个粒子, 3) g_{best} 和 p_{best} 固定.本文同样在这些基本假设下,对CLPSO进行收敛速度的分析. CLPSO更新关系式可以表达为式(4):

$$\begin{cases} v^{t+1} = \omega \cdot v^t + c \cdot r \cdot (p^t - x^t), \\ x^{t+1} = x^t + v^{t+1}. \end{cases} \quad (4)$$

而式(4)可以写成矩阵的形式,如式(5):

$$\begin{bmatrix} v^{t+1} \\ x^{t+1} \end{bmatrix} = \begin{bmatrix} \omega & -c \cdot r \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} v^t \\ x^t \end{bmatrix} + \begin{bmatrix} c \cdot r \cdot p \\ 0 \end{bmatrix}, \quad (5)$$

缩写为 $X^{t+1} = A \cdot X^t + b$.

当CLPSO收敛时,存在不动点 (v^*, x^*) ,使得

$$X^* = A \cdot X^* + b. \quad (6)$$

定义 $\Delta X^t = X^t - X^*$,根据式(5)和式(6),可得

$$\Delta X^{t+1} = A \cdot \Delta X^t. \quad (7)$$

对式(7)两边取范数,依矩阵相容范数的定义可得

$$\frac{\|\Delta X^{t+1}\|}{\|\Delta X^t\|} \leq \|A\|. \quad (8)$$

这说明CLPSO只有线性收敛速度,与其他PSO的收敛速度分析结果一致[25].

下面对DFP拟牛顿方法的局部收敛性能进行分析.依据文献[26],其收敛速度如式(9)所示:

$$\frac{\|x^{t+1} - x^*\|}{\|x^t - x^*\|} \leq \gamma(1+r) \|B^t - F'(x^*)\|. \quad (9)$$

又因为可证明 $\lim_{t \rightarrow \infty} \|B^t - F'(x^*)\| = 0$,故

$$\lim_{t \rightarrow \infty} \frac{\|x^{t+1} - x^*\|}{\|x^t - x^*\|} = 0, \text{ 也就是说, DFP拟牛顿方法具有}$$

局部超线性收敛速度.

综上可知,DFP拟牛顿方法的收敛速度快于CLPSO.这意味着对于局部问题,以DFP为代表的LS方法只需使用更少的FEs就能达到指定收敛精度,因而CLPSO-LS混合算法具有更强的收敛性能.

3 标准测试函数(Benchmark test functions)

为了测试本文混合算法的性能,选取11个标准函数[11]进行实验.测试函数的表达式和搜索范围如表2所示,其他具体性质参见文献[11].

表 2 标准测试函数的名称、表达式和搜索范围

Table 2 The name, formula and search range of benchmark functions

函数名称	函数表达式	搜索范围
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$
Rosenbrock	$f_2(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	$[-2.048, 2.048]^D$
Ackley	$f_3(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)) + 20 + e$	$[-32.768, 32.768]^D$
Griewanks	$f_4(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \frac{x_i}{\sqrt{i}} + 1$	$[-600, 600]^D$
Rastrigin	$f_5(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$
不连续的Rastrigin	$f_6(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10),$ $y_i = \begin{cases} x_i, & x_i < \frac{1}{2}, \\ \frac{2x_i}{2}, & x_i \geq \frac{1}{2}, \end{cases} i = 1, 2, \dots, D$	$[-5.12, 5.12]^D$
Schwefel	$f_7(x) = 418.9829 \times D - \sum_{i=1}^D x_i \sin x_i ^{\frac{1}{2}}$	$[-500, 500]^D$
Weierstrass	$f_8(x) = \sum_{i=1}^D (\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))]) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \cdot 0.5)], a = 0.5, b = 3, k_{\max} = 20$	$[-0.5, 0.5]^D$
偏移的Rastrigin	$f_9(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{\text{bias}1},$ $z = x - o, x = [x_1 \ x_2 \ \dots \ x_D], o = [o_1 \ o_2 \ \dots \ o_D]$	$[-5.12, 5.12]^D$
偏移不连续的Rastrigin	$f_{10}(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10) + f_{\text{bias}2},$ $y_i = \begin{cases} z_i, & z_i < \frac{1}{2}, \\ \frac{2z_i}{2}, & z_i \geq \frac{1}{2}, \end{cases} i = 1, 2, \dots, D,$ $z = x - o, x = [x_1 \ x_2 \ \dots \ x_D], o = [o_1 \ o_2 \ \dots \ o_D]$	$[-5.12, 5.12]^D$
偏移的Schwefel	$f_{11}(x) = 418.9829 \times D - \sum_{i=1}^D z_i \sin z_i ^{\frac{1}{2}} + f_{\text{bias}3},$ $z = x - o, x = [x_1 \ x_2 \ \dots \ x_D], o = [o_1 \ o_2 \ \dots \ o_D]$	$[-500, 500]^D$

4 数值实验(Numerical experiment)

4.1 实验参数设置(Preferences of experiment)

为了验证本文所提算法的有效性和可扩展性, 分别对11个标准测试函数的不同维度(10维、30维和50维)的问题进行实验. 第1组实验的参数设置如下: 问题维度 $D = 10$, 种群规模 $N = 40$, 最大目标函数评价次数 $\text{Max_FEs} = 4000 \times D$, LS的目标函数评价次数 $\text{L_FEs} = 70 \times D$. 第2组实验: $D = 30$, $N = 100$, $\text{Max_FEs} = 10000 \times D$, $\text{L_FEs} = 100 \times D$. 第3组

实验: $D = 50$, $N = 150$, $\text{Max_FEs} = 15000 \times D$, $\text{L_FEs} = 300 \times D$. 每次实验独立运行25次. 实验运行环境: MATLAB R2014a, Win10专业版, 64位操作系统, 8 GB内存, 2.40 GHz处理器.

4.2 CLPSO-LS的实验结果分析(Experimental results analysis of CLPSO-LS)

实验统计结果(表3)列出了4种CLPSO-LS混合算法和CLPSO算法对各测试函数的10维、30维和50维问题的计算收敛精度的中位数和均值.

表3 测试标准函数的10维、30维和50维问题的收敛精度

Table 3 Convergence precision of $D10$, $D30$ and $D50$ problems for benchmark functions

函数	算法	$D10$		$D30$		$D50$	
		Median	Mean	Median	Mean	Median	Mean
f_1	CLPSO	3.88E-11	5.09E-11	6.96E-11	7.06E-11	5.00E-12	5.25E-12
	CLbfgs	2.16E-17	2.17E-17	9.02E-18	1.28E-17	3.06E-17	2.95E-17
	CLdfp	1.90E-17	1.81E-17	2.34E-17	2.99E-17	3.90E-17	3.84E-17
	CLsd	3.06E-18	4.11E-18	5.65E-18	5.51E-18	1.86E-18	1.88E-18
	CLnelder	1.24E-12	2.96E-12	1.06E-11	1.11E-11	6.63E-13	8.62E-13
f_2	CLPSO	2.83E+00	2.93E+00	1.99E+01	2.00E+01	3.96E+01	3.97E+01
	CLbfgs	6.02E-11	5.77E-11	3.91E+00	4.18E+00	6.52E-11	6.38E-11
	CLdfp	1.44E+00	1.45E+00	1.82E+01	1.80E+01	3.70E+01	3.66E+01
	CLsd	2.28E+00	2.24E+00	1.94E+01	1.88E+01	3.90E+01	3.83E+01
	CLnelder	1.63E+00	1.65E+00	1.88E+01	1.85E+01	3.80E+01	3.78E+01
f_3	CLPSO	3.53E-06	3.30E-06	2.95E-06	2.95E-06	5.07E-07	5.06E-07
	CLbfgs	7.73E-09	7.90E-09	8.39E-09	8.68E-09	7.96E-09	7.60E-09
	CLdfp	8.81E-09	8.84E-09	8.20E-09	8.96E-09	8.05E-09	7.37E-09
	CLsd	1.92E-09	1.92E-09	4.98E-10	8.12E-10	7.82E-12	3.24E-10
	CLnelder	6.74E-07	9.34E-07	1.24E-06	1.23E-06	2.19E-07	2.32E-07
f_4	CLPSO	3.27E-04	9.95E-04	3.21E-08	4.31E-08	3.34E-10	4.87E-10
	CLbfgs	1.11E-16	2.96E-04	4.44E-16	6.08E-16	8.88E-16	8.39E-16
	CLdfp	1.11E-16	2.96E-04	1.11E-15	1.64E-15	3.89E-15	5.78E-15
	CLsd	3.47E-11	2.96E-04	6.55E-15	6.62E-15	1.57E-14	1.56E-14
	CLnelder	1.11E-05	3.15E-04	3.30E-09	6.23E-09	5.82E-11	1.00E-10
f_5	CLPSO	9.37E-06	1.80E-05	5.02E-03	5.51E-03	8.49E-02	8.66E-02
	CLbfgs	0.00E+00	9.24E-16	0.00E+00	4.97E-16	0.00E+00	1.71E-15
	CLdfp	0.00E+00	7.11E-16	0.00E+00	3.55E-16	0.00E+00	1.99E-15
	CLsd	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.42E-16
	CLnelder	3.56E-07	5.10E-07	5.83E-04	6.96E-04	1.55E-02	2.02E-02
f_6	CLPSO	9.40E-05	1.98E-04	8.99E-02	9.38E-02	9.44E-01	1.07E+00
	CLbfgs	0.00E+00	9.95E-16	0.00E+00	2.84E-16	0.00E+00	2.84E-16
	CLdfp	0.00E+00	1.35E-15	0.00E+00	4.26E-16	0.00E+00	1.42E-16
	CLsd	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	CLnelder	5.11E-06	1.18E-05	1.16E-02	1.50E-02	1.85E-01	2.08E-01
f_7	CLPSO	1.37E-08	1.84E-08	2.00E-06	2.64E-06	5.52E-06	7.82E-06
	CLbfgs	1.18E-11	1.09E-11	4.00E-11	3.71E-11	5.46E-11	5.91E-11
	CLdfp	1.18E-11	1.11E-11	3.64E-11	3.59E-11	6.18E-11	6.24E-11
	CLsd	1.36E-11	1.33E-11	4.55E-11	4.48E-11	8.00E-11	8.31E-11
	CLnelder	8.91E-11	1.92E-10	1.19E-08	2.47E-08	3.53E-10	4.37E-10
f_8	CLPSO	3.72E-05	3.78E-05	5.85E-05	5.74E-05	7.85E-06	8.55E-06
	CLbfgs	5.25E-06	1.87E-05	7.58E-06	1.03E-05	4.26E-10	2.03E-07
	CLdfp	3.42E-05	3.26E-05	2.57E-05	2.89E-05	1.67E-07	4.97E-07
	CLsd	3.88E-05	4.40E-05	4.61E-05	4.49E-05	1.08E-06	2.26E-06
	CLnelder	3.78E-05	4.04E-05	4.61E-05	4.31E-05	1.89E-06	3.15E-06
f_9	CLPSO	2.27E-06	4.47E-06	2.62E-03	3.59E-03	4.78E-02	4.55E-02
	CLbfgs	6.64E-13	6.19E-13	1.93E-12	1.87E-12	3.19E-12	3.15E-12
	CLdfp	6.64E-13	6.19E-13	1.93E-12	1.87E-12	3.19E-12	3.15E-12
	CLsd	6.63E-13	6.19E-13	1.89E-12	1.80E-12	3.15E-12	3.09E-12
	CLnelder	8.38E-11	3.66E-10	1.05E-08	1.21E-05	8.60E-08	3.87E-05
f_{10}	CLPSO	3.35E-05	5.46E-05	4.01E-02	4.58E-02	5.33E-01	5.65E-01
	CLbfgs	5.29E-13	5.36E-13	1.94E-12	1.87E-12	3.23E-12	3.19E-12
	CLdfp	5.29E-13	5.36E-13	1.94E-12	1.87E-12	3.23E-12	3.19E-12
	CLsd	5.28E-13	5.33E-13	1.90E-12	1.82E-12	3.11E-12	3.13E-12
	CLnelder	1.35E-10	7.83E-10	2.96E-07	1.06E-03	4.32E-04	9.10E-03
f_{11}	CLPSO	6.24E-10	1.16E-08	1.64E-08	2.01E-08	8.39E-09	1.24E-08
	CLbfgs	5.46E-12	1.02E-11	7.28E-11	2.49E-10	2.22E-10	2.98E-10
	CLdfp	8.19E-12	1.15E-11	9.64E-11	4.20E-10	2.33E-10	3.59E-10
	CLsd	6.37E-12	6.84E-12	6.91E-11	2.10E-10	1.56E-10	2.31E-10
	CLnelder	4.55E-12	1.22E-11	3.82E-10	6.35E-10	3.27E-10	3.58E-10

由于算法的CPU耗时取决于FEs, 算法性能的评估标准是给定 Max.FEs 下所达到的收敛精度. 对于 CLPSO-LS 混合算法, 当 CLPSO 迭代到 FEs = Max.FEs 时, 执行 LS. 而大多情况下, LS 不需要用完给定的 L.FEs 就已收敛.

4.2.1 CLPSO-LS混合算法收敛精度分析(Convergence precision analysis of CLPSO-LS hybrid algorithm)

由收敛精度的统计结果(表3)可知, 所有测试函数的10维、30维和50维问题, CLPSO-LS混合算法在相同甚至更少的FEs下, 都比CLPSO算法达到了更高的收敛精度. 下面具体分析各算法在不同函数上的优化性能.

先对10维问题的测试结果进行分析. 1) CLsd在 f_5 和 f_6 的均值为0, 说明每次测试都收敛到全局最优解. CLbfgs和CLdfp在 f_5 和 f_6 的均值, 也收敛到了很高的精度, 比CLPSO算法提高了10至11个数量级. 上述这3种混合算法在 f_5 和 f_6 的收敛精度的中位数都是0, 说明在大多数实验中, 两种拟牛顿方法也收敛到了全局最优解. 2) 对于 $f_1, f_2, f_3, f_7, f_9, f_{10}$ 和 f_{11} , 混合算法的收敛精度比CLPSO提高了3-11个数量级. 其中, CLPSO算法本身对 f_2 的收敛效果不太好, 而CLbfgs方法将其收敛精度大幅提高到了E-11这个数量级. 3) 在 f_4 和 f_8 的实验中, 本文所提算法的计算精度相比于CLPSO也提高了2-3倍.

同样地, 从表3中可看出, CLPSO-LS混合算法的性能在30维和50维问题上也有明显提高, 证明了其对高维问题的有效性.

综上, CLPSO-LS混合算法相比于CLPSO算法, 对不同维度的所有测试函数都具有明显优势, 在绝大多数函数中优势非常显著. 并且, 混合算法的优化性能在各维度间的差异较小, 具有维度扩展性与普适性.

4.2.2 4种CLPSO-LS混合算法的排名分析(Ranking analysis of the four CLPSO-LS hybrid algorithms)

分别按算法对各测试函数的收敛精度的均值和中位数进行排序(遇到相同值时取并列名次), 得出各算法在每个测试函数的排名, 再计算出算法的平均排名和最终排名. 由表4基于均值的排名的统计结果可知, CLbfgs和CLsd并列第1, 接下来依次是CLdfp和CLnelder. 拟牛顿方法在寻找可微凸函数的局部最优解方面具有很强的能力. 单纯形法的收敛速度不如其它方法快, 但其有潜力在不连续问题和连续组合优化问题中取得好的效果, 所以在实际应用中依然值得考虑.

表 4 基于收敛精度的均值的算法排名

Table 4 Ranking based on the mean of convergence precision

函数	CLPSO	CLbfgs	CLdfp	CLsd	CLnelder
f_1-D10	5	3	2	1	4
f_2-D10	5	1	2	4	3
f_3-D10	5	2	3	1	4
f_4-D10	5	1	1	1	4
f_5-D10	5	3	2	1	4
f_6-D10	5	2	3	1	4
f_7-D10	5	1	2	3	4
f_8-D10	3	1	2	5	4
f_9-D10	5	3	2	1	4
$f_{10}-D10$	5	2	3	1	4
$f_{11}-D10$	5	2	3	1	4
f_1-D30	5	2	3	1	4
f_2-D30	5	1	2	4	3
f_3-D30	5	2	3	1	4
f_4-D30	5	1	2	3	4
f_5-D30	5	3	2	1	4
f_6-D30	5	2	3	1	4
f_7-D30	5	2	1	3	4
f_8-D30	5	1	2	4	3
f_9-D30	5	2	3	1	4
$f_{10}-D30$	5	2	3	1	4
$f_{11}-D30$	5	2	3	1	4
f_1-D50	5	2	3	1	4
f_2-D50	5	1	2	4	3
f_3-D50	5	3	2	1	4
f_4-D50	5	1	2	3	4
f_5-D50	5	2	3	1	4
f_6-D50	5	3	2	1	4
f_7-D50	5	1	2	3	4
f_8-D50	5	1	2	3	4
f_9-D50	5	2	2	1	4
$f_{10}-D50$	5	2	2	1	4
$f_{11}-D50$	5	2	4	1	3
平均排名	4.9394	1.8485	2.3636	1.8485	3.8485
最终排名	5	1	3	1	4

为了充分进行对比, 本文也采用了中位数进行排名(表5). 得出CLbfgs最优, 且4种混合算法都优于CLPSO算法. 综合两种排名方式, 得出具有超线性收敛速度的CLbfgs算法的性能最优, 可推荐作为后续研究的首选方法.

表 5 基于中位数的算法排名

Table 5 Ranking based on the median of convergence precision

度量指标	CLPSO	CLbfgs	CLdfp	CLsd	CLnelder
平均排名	4.9394	1.5455	1.9394	1.9697	3.7879
最终排名	5	1	2	3	4

4.3 与其他改进PSO算法的对比(Comparison with other improved PSO algorithms)

4.3.1 收敛精度分析(Convergence precision analysis)

为了进一步验证本文所提算法的优越性,与另外8种优秀的改进PSO算法进行对比,包括CSO^[6], SLPSO^[7], FIPS^[8], FDR-PSO^[9], CPSO-H^[10], DNLPSO^[15], HCLPSO^[16]和DMS-L-PSO^[19]. 其中DNLPSO和HCLPSO是基于CLPSO改进的算法. DMS-L-PSO是采用了不同于本文耦合策略的基于LS的动态多子群PSO算法. 出于统一性的考虑,同样对上述11个标准函数的10维、30维和50维问题进行测试,参数设置与前面的实验一致.

平均收敛精度如表6所示. 粗体显示的是各函数的

最优结果. CLbfgs算法在10维的 f_2, f_4, f_5, f_6, f_7 和 f_{10} , 30维的 $f_5, f_6, f_7, f_9, f_{10}$ 和 f_{11} 以及50维的 $f_2, f_5, f_6, f_7, f_9, f_{10}$ 和 f_{11} 中超过其他所有算法,尤其在复杂多峰问题中表现出很强的优势. 在部分函数中,其他算法略优于CLbfgs. 对单峰函数 f_1 和简单多峰函数 f_3 , SLPSO算法和CSO算法在3个维度测试中都分别取得最优. CSO算法和DMS-L-PSO算法对10维 f_8 收敛到全局最优, CSO算法和SLPSO算法对30维和50维的 f_8 收敛到全局最优. 在10维的 f_{11} 中HCLPSO略优于本文算法. 对于10维 f_2, f_5 和 f_6 , CLPSO本不是最优的,但在加入LS之后, CLbfgs混合算法的性能在这3个函数上明显超过了其他所有算法. 在30维的 f_6, f_9 和 f_{10} 中,以及50维 f_2, f_5, f_6, f_7, f_9 和 f_{10} 中,也有同样的结论.

表6 各种改进PSO算法的收敛精度的均值

Table 6 The mean of convergence precision of improved PSO algorithms

函数		CSO	SLPSO	FIPS	FDR-PSO	CPSO-H	DMS-L-PSO	DNLPSO	HCLPSO	CLPSO	CLbfgs
D10	f_1	7.4E-79	2.4E-87	1.2E-08	6.4E-52	3.8E-06	3.8E-38	1.9E-01	2.0E-22	5.1E-11	2.2E-17
	f_2	6.1E+00	5.3E+00	5.1E+00	4.0E-01	2.4E+00	6.0E-11	5.0E+00	3.8E+00	2.9E+00	5.8E-11
	f_3	3.0E-15	6.6E-02	4.6E-05	5.4E-15	6.3E-04	3.7E-15	1.5E+00	6.1E-12	3.3E-06	7.9E-09
	f_4	2.1E-02	1.6E-02	1.3E-01	7.1E-02	4.8E-02	1.0E-02	3.8E-01	2.3E-02	1.0E-03	3.0E-04
	f_5	4.5E+00	6.1E+00	5.5E+00	3.7E+00	1.4E-06	5.3E-15	1.0E+01	1.0E-04	1.8E-05	9.2E-16
	f_6	6.5E+00	8.1E+00	7.1E+00	3.8E+00	8.7E-07	3.2E-01	9.0E+00	4.3E-02	2.0E-04	1.0E-15
	f_7	5.6E+02	6.3E+02	8.3E-05	7.1E+02	3.8E+01	1.6E+02	8.1E+02	1.4E+01	1.8E-08	1.1E-11
	f_8	0.0E+00	2.3E-06	2.6E-03	1.7E-04	9.3E-03	0.0E+00	1.7E+00	2.6E-13	3.8E-05	1.9E-05
	f_9	4.7E+00	6.2E+00	4.6E+00	4.0E+00	4.0E-02	1.9E-13	1.1E+01	4.5E-05	4.5E-06	6.2E-13
	f_{10}	8.0E+00	9.9E+00	6.2E+00	4.0E+00	8.0E-02	3.9E-01	1.2E+01	4.0E-02	5.5E-05	5.4E-13
	f_{11}	2.1E+02	2.9E+02	2.3E-06	3.0E+02	1.2E+01	1.9E+01	4.3E+02	8.0E-13	1.2E-08	1.0E-11
D30	f_1	1.2E-113	2.1E-137	4.1E-06	2.8E-96	5.2E-06	5.1E-41	4.1E+02	6.4E-40	7.1E-11	1.3E-17
	f_2	2.1E+01	1.5E+01	2.4E+01	1.3E+01	2.1E+01	7.4E-11	4.1E+01	1.8E+01	2.0E+01	4.2E+00
	f_3	3.6E-15	6.4E-15	4.5E-04	1.9E-14	5.5E-04	2.5E-14	1.9E+00	1.7E-14	3.0E-06	8.7E-09
	f_4	3.9E-04	3.9E-04	2.6E-04	1.0E-02	2.5E-02	0.0E+00	9.1E-01	4.5E-03	4.3E-08	6.1E-16
	f_5	8.4E+00	1.4E+01	8.6E+01	2.5E+01	1.9E-06	1.3E+01	7.0E+01	1.2E+00	5.5E-03	5.0E-16
	f_6	1.8E+01	2.4E+01	7.1E+01	1.3E+01	1.6E-06	1.9E+01	6.1E+01	1.8E+00	9.4E-02	2.8E-16
	f_7	1.4E+03	1.6E+03	3.4E+00	3.2E+03	3.1E-06	2.0E+03	3.3E+03	9.5E+00	2.6E-06	3.7E-11
	f_8	0.0E+00	0.0E+00	2.1E-02	3.6E-04	2.5E-02	5.7E-15	9.4E+00	2.6E-15	5.7E-05	1.0E-05
	f_9	1.0E+01	1.7E+01	6.2E+01	2.6E+01	1.4E-06	1.5E+01	8.2E+01	5.8E-01	3.6E-03	1.9E-12
	f_{10}	1.9E+01	2.8E+01	5.3E+01	1.6E+01	1.4E-06	2.2E+01	7.4E+01	1.7E+00	4.6E-02	1.9E-12
	f_{11}	3.4E+02	6.3E+02	2.1E-03	1.7E+03	1.1E-05	9.9E+01	2.7E+03	5.8E-05	2.0E-08	2.5E-10
D50	f_1	5.9E-127	7.5E-160	4.1E-04	1.8E-99	4.6E-06	9.5E-40	1.6E+01	4.2E-46	5.3E-12	3.0E-17
	f_2	3.6E+01	2.5E+01	4.4E+01	2.7E+01	3.2E+01	7.2E-11	5.7E+01	3.2E+01	4.0E+01	6.4E-11
	f_3	3.6E-15	7.0E-15	3.6E-03	3.5E-14	4.4E-04	7.5E-14	1.7E+00	3.2E-14	5.1E-07	7.6E-09
	f_4	0.0E+00	3.0E-04	7.2E-04	4.0E-03	3.9E-02	0.0E+00	2.2E+00	2.1E-03	4.9E-10	8.4E-16
	f_5	1.0E+01	1.6E+02	2.3E+02	5.3E+01	1.4E-06	2.9E+01	1.1E+02	3.0E+00	8.7E-02	1.7E-15
	f_6	1.3E+02	3.0E+02	2.1E+02	2.5E+01	1.5E-06	3.9E+01	1.2E+02	3.5E+00	1.1E+00	2.8E-16
	f_7	2.2E+03	2.3E+03	4.7E+02	6.2E+03	3.8E-06	6.4E+03	7.6E+03	2.4E+01	7.8E-06	5.9E-11
	f_8	0.0E+00	0.0E+00	1.1E-01	1.3E-01	3.1E-02	1.7E-12	1.3E+01	1.7E-14	8.6E-06	2.0E-07
	f_9	1.2E+01	3.5E+01	1.7E+02	6.3E+01	1.1E-06	3.8E+01	1.6E+02	2.0E+00	4.6E-02	3.2E-12
	f_{10}	3.2E+01	1.2E+02	1.7E+02	3.1E+01	8.6E-07	5.2E+01	1.7E+02	3.8E+00	5.7E-01	3.2E-12
	f_{11}	5.0E+02	1.0E+03	5.0E-01	3.4E+03	9.2E-06	1.7E+02	5.2E+03	9.1E-03	1.2E-08	3.0E-10

4.3.2 算法排名的统计分析(Statistical analysis of ranking)

依据PSO算法的收敛精度的均值进行排名(表7), 结果显示CLbfgs为第1. 接下来的3个算法, 依次为CLPSO, HCLPSO和DMS-L-PSO. 排名较靠后的另外6个算法依次为CSO, CPSO-H, SLPSO, FDR-PSO,

FIPS和DNLPSO.

为了检验各个算法的排名是否存在显著性差异, 进行统计检验. 首先采用非参数Friedman test判断这些算法之间是否在统计上不同. 然后将CLbfgs算法与其他所有算法进行非参数的Wilcoxon秩和检验, 评判CLbfgs是否显著优于其他所有算法.

表 7 基于收敛精度的均值的改进PSO算法的排名

Table 7 Ranking of PSO variants based on mean of convergence precision

度量指标	CSO	SLPSO	FIPS	FDR-PSO	CPSO-H	DMS-L-PSO	DNLPSO	HCLPSO	CLPSO	CLbfgs
得第1的次数	7	5	0	0	0	4	0	1	0	19
平均排名	4.97	6	7.576	6.212	5.121	4.576	9.667	4.212	4.152	2.424
最终排名	5	7	9	8	6	4	10	3	2	1

通过对10个算法的排名进行Friedman test, 得到 $p = 4.9733\text{E}-24 < 0.05$, 说明在显著性水平0.05下拒绝原假设(H_0 : 算法间无显著差异), 即认为这10种算法之间存在差别. 具体地, 检验本文的CLbfgs算法是否显著性地优于其他所有对比算法. 通过依次与9个对比算法的非参数Wilcoxon秩和检验, 得到表8的输出结果. 本文有33个样本, 在大样本(样本数 >30)时, 采用z检验. 标准正态分布在显著水平为0.05时, 上临界值为1.645, 下临界值为-1.645. 与9种算法检验的z统计量值均大于上临界值1.645, 所以拒绝原假设(H_0 : 检验的算法对之间无显著差异). 此外, 表格中的p值均小于0.05以及h值均为1, 也表明要拒绝原假设. 也就是说, CLbfgs 显著性地优于这9种对比PSO算法.

表 8 Wilcoxon秩和检验的结果

Table 8 The results of Wilcoxon rank sum test

算法	z统计量值	秩和	p值	h值
CSO	3.84424	1395	1.21E-04	1
SLPSO	4.66512	1460	3.08E-06	1
FIPS	6.30656	1590	2.85E-10	1
FDR-PSO	5.49923	1528	3.81E-08	1
CPSO-H	4.50203	1450.5	6.73E-06	1
DMS-L-PSO	3.8198	1395.5	1.34E-04	1
DNLPSO	7.32595	1650	2.37E-13	1
HCLPSO	3.58163	1378.5	3.41E-04	1
CLPSO	4.07286	1417	4.64E-05	1

综上, CLbfgs的计算性能显著优于其他改进PSO算法. 需要特别指出, 在33个测试中, CLbfgs算法在20个函数中性能优于采用了不同LS耦合策略的DMS-L-PSO算法, 在23个函数中性能优于CLPSO的改进算法HCLPSO, 在所有函数测试中都优于另一种CLPSO的改进算法DNLPSO. 这说明, 本文的混合算法不仅优于其他CLPSO改进算法, 同时也优于基于中间启动LS的耦合策略的其他PSO混合算法, 进一步验证了本

文耦合策略的合理性和有效性.

4.3.3 收敛曲线(Convergence curve)

算法的优化过程可直观地从收敛曲线看出. 因相似性和篇幅限制, 仅以10维函数 f_4 (图4)的进化曲线为代表进行说明. 图中收敛曲线是各算法25次运行的最优解的平均进化曲线. 迭代前期算法之间的差异不太大, 而CLPSO和CLbfgs在迭代后期优于其他算法, 尤其CLbfgs在迭代末期突然出现急剧下降, 这是传统LS方法(此处为BFGS拟牛顿方法)所起的积极加速作用. LS方法在很少的迭代步内, 把最优解的精度提高了很多. 由于LS消耗的FEs远远小于整个迭代过程的FEs, 图中未展示LS的详细进化过程, 只给出LS优化前后的最优解.

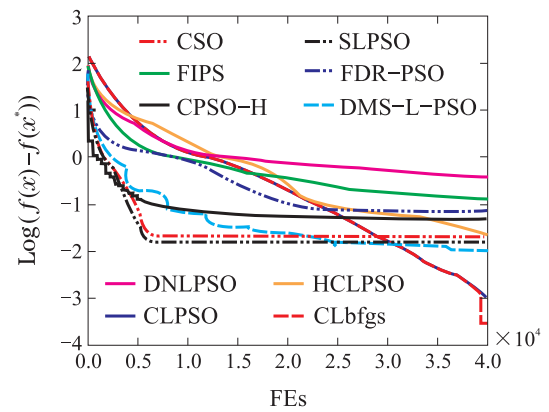


图 4 各算法对10维的函数 f_4 的收敛曲线

Fig. 4 Convergence curve of f_4 of 10D problem

对比CLbfgs与另外两种CLPSO的改进算法(HCLPSO和DNLPSO). 1) HCLPSO用两个不同的子群分别负责探索和开发, 其目的是均衡全局探索和局部开发的能力. 由于采用分群机制, HCLPSO只有部分粒子用于开发, 全局能力弱于CLPSO. 同时, CLPSO的局部搜索弱于传统的LS方法, 因此HCLPSO的计算性能弱于CLbfgs的计算性能. 2) DNLPSO为了提高CLP-

SO的局部搜索性能,在速度更新公式中同时也向 g_{best} 学习,粒子向 g_{best} 聚拢的趋势更加显著,这可能导致种群多样性降低,牺牲全局能力.因此DNLPSO只在迭代早期表现出一些优势,很快就落后于其他PSO算法.

CLbfgs作为一种基于LS的PSO算法,将其与DMS-L-PSO算法进行对比和分析.由于DMS-L-PSO在前期迭代过程中就启动LS,其局部收敛能力较强,但代价是大量的LS占用了PSO全局搜索的计算资源,同时增加了陷入局部极小的风险,可能会削弱算法的全局能力,所以其在后期表现变差.而CLbfgs的混合策略保持了CLPSO的全局性能,具有持续搜索能力.

综上,收敛曲线展现的迭代优化过程,也验证了本文简单高效耦合策略下的CLPSO-LS混合算法的合理性和优越性.

5 结论(Conclusions)

本文建立了基于RPGOB指标的定量分析方法,用于评判混合算法中耦合策略的好坏.通过这个指标对中间以不同频率启动LS的现有混合算法的分析,发现中间启动LS不仅无法起到全局加速的作用,还增加了算法陷入局部极小的可能性.在此理论分析的基础上,本文提出了一种中间不启动LS的耦合策略,只在迭代最后将具有快速收敛性能的传统LS方法引入到CLPSO算法,这样得到的CLPSO-LS混合算法能够在继承CLPSO全局优势的前提下,大幅提高算法的收敛性能,它简单且高效.

通过大量的实验测试,验证了本文提出的简单高效耦合策略下的CLPSO-LS混合算法在计算效率和收敛精度上的提升效果,说明了这种耦合策略的合理性和高效性.同时,对比分析了4种基于不同LS方法的混合算法,其中CLbfgs的性能最优.特别地,与另外8种改进PSO算法的对比实验和统计分析进一步验证了本文算法的优越性.实验结果显示,CLbfgs的综合性能优于其他所有对比算法,且统计检验结果表明本文混合算法与其他算法的差异是显著的.特别地,对比算法包含两种基于CLPSO的改进算法和一种加LS的PSO算法,说明本文的混合算法作为一种改进的CLPSO算法不仅优于其他CLPSO改进算法,本文的耦合策略也比传统的耦合策略更有效.

未来的研究工作可以从两个方面展开.一方面研究如何进一步提升CLPSO-LS混合算法的性能;另一方面将该算法应用于实际问题的求解.

参考文献(References):

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization [C] // *Proceedings of IEEE International Conference on Neural Networks*. Perth: IEEE, 1995: 1942 – 1948.
- [2] SINGH M R, MAHAPATRA S S. A quantum behaved particle swarm optimization for flexible job shop scheduling [J]. *Computers & Industrial Engineering*, 2016, 93: 36 – 44.
- [3] ZHANG Jing, WANG Wanliang, XU Xinli, et al. Hybrid particle-swarm optimization for multi-objective flexible job-shop scheduling problem [J]. *Control Theory & Applications*, 2012, 29(6): 715 – 722. (张静, 王万良, 徐新黎, 等. 混合粒子群算法求解多目标柔性作业车间调度问题 [J]. *控制理论与应用*, 2012, 29(6): 715 – 722.)
- [4] MIRJALILI S A, HASHIM S Z M, SARDROUDI H M. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm [J]. *Applied Mathematics and Computation*, 2012, 218(22): 11125 – 11137.
- [5] HUANG Kaiqi, CHEN Ronghua, DING Wensi. Crossover elite opposition-based particle swarm optimization algorithm for positioning control of rock drilling robotic drilling arm [J]. *Control Theory & Applications*, 2017, 34(3): 303 – 311. (黄开启, 陈荣华, 丁问司. 凿岩机器人钻臂定位控制交叉精英反向粒子群算法 [J]. *控制理论与应用*, 2017, 34(3): 303 – 311.)
- [6] CHENG R, JIN Y. A competitive swarm optimizer for large scale optimization [J]. *IEEE Transactions on Cybernetics*, 2015, 45(2): 191 – 204.
- [7] CHENG R, JIN Y. A social learning particle swarm optimization algorithm for scalable optimization [J]. *Information Sciences*, 2015, 291(6): 43 – 60.
- [8] MENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: simpler, maybe better [J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 204 – 210.
- [9] PERAM T, VEERAMACHANENI K, MOHAN C K. Fitness-distance-ratio based particle swarm optimization [C] // *Proceedings of IEEE on Swarm Intelligence Symposium*. Indianapolis: IEEE, 2003: 174 – 181.
- [10] VAN DEN BERGH F, ENGELBRECHT A P. A cooperative approach to particle swarm optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 225 – 239.
- [11] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(3): 281 – 295.
- [12] PAN Quanke, WANG Ling, GAO Liang. The-state-of-art of discrete particle swarm optimization algorithms [J]. *Control and Decision*, 2009, 24(10): 1441 – 1449. (潘全科, 王凌, 高亮. 离散微粒群优化算法的研究进展 [J]. *控制与决策*, 2009, 24(10): 1441 – 1449.)
- [13] CHENG Shi, SHI Yuhui. Measurement of PSO diversity based on L1 norm [J]. *Computer Science*, 2011, 38(7): 190 – 193. (程适, 史玉回. 基于L1范式的粒子群算法群体多样性研究 [J]. *计算机科学*, 2011, 38(7): 190 – 193.)
- [14] WANG Yu, ZHANG Weiguo, FU Li, et al. Nash equilibrium strategies approach for aerial combat based on elite re-election particle swarm optimization [J]. *Control Theory & Applications*, 2015, 32(7): 857 – 865. (王昱, 章卫国, 傅莉, 等. 基于精英改选机制的粒子群算法的空战纳什均衡策略逼近 [J]. *控制理论与应用*, 2015, 32(7): 857 – 865.)
- [15] NASIR M, DAS S, MAITY D, et al. A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization [J]. *Information Sciences*, 2012, 209(5): 16 – 36.
- [16] LYNN N, SUGANTHAN P N. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation [J]. *Swarm and Evolutionary Computation*, 2015, 24: 11 – 24.
- [17] LIN Guohan, ZHANG Jing, LIU Zhaohua. Immune comprehensive learning particle swarm optimization algorithm [J]. *Application Research of Computers*, 2014, 31(11): 3229 – 3232. (林国汉, 章兢, 刘朝华. 免疫综合学习粒子群优化算法 [J]. *计算机应用研究*, 2014, 31(11): 3229 – 3232.)

- [18] LIU Guohan, ZHANG Jing, LIU Zhaohua, et al. Parameter identification of PMSM using improved comprehensive learning particle swarm optimization [J]. *Electric Machines and Control*, 2015, 19(1): 51 – 57.
(林国汉, 章兢, 刘朝华, 等. 改进综合学习粒子群算法的PMSM参数辨识 [J]. 电机与控制学报, 2015, 19(1): 51 – 57.)
- [19] ZHAO S Z, LIANG J J, SUGANTHAN P N, et al. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization [C] // *2008 IEEE Congress on Evolutionary Computation*. Hong Kong: IEEE, 2008: 3845 – 3852.
- [20] QIN J, YIN Y, BAN X. A hybrid of particle swarm optimization and local search for multimodal functions [C] // *International Conference in Swarm Intelligence*. Berlin: Springer, 2010: 589 – 596.
- [21] SUDHOLT D. Parametrization and balancing local and global search [M] // FERRANTE N, CARLOS C, PABLO M. *Handbook of Memetic Algorithms — Studies in Computational Intelligence*. Heidelberg: Springer, 2012, 379: 55 – 72.
- [22] HAN F, LIU Q. An improved hybrid PSO based on ARPSO and the Quasi-Newton method [C] // *International Conference in Swarm Intelligence*. Cham: Springer, 2015: 460 – 467.
- [23] ASGARI N, RAJABI M, JAMSHIDI M, et al. A memetic algorithm for a multi-objective obnoxious waste location-routing problem: a case study [J]. *Annals of Operations Research*, 2017, 250(2): 279 – 308.
- [24] LUENBERGER D G, YE Y. *Linear and Nonlinear Programming* [M]. Fourth Edition. Switzerland: Springer, 2016: 213 – 257.
- [25] TRELEA I C. The particle swarm optimization algorithm: convergence analysis and parameter selection [J]. *Information Processing Letters*, 2003, 85(6): 317 – 325.
- [26] BROYDEN C G, DENNIS J E, MORÉ J J. On the local and super-linear convergence of quasi-Newton methods [J]. *JIMA Journal of Applied Mathematics*, 1973, 12(3): 223 – 245.

作者简介:

李文锋 (1966–), 男, 教授, 博士生导师, 目前研究方向为物联网技术与应用、优化算法和机器人技术, E-mail: liwf@whut.edu.cn;

曹玉莲 (1987–), 女, 博士研究生, 目前研究方向为群智能算法、机器学习和物流系统优化, E-mail: yulian@whut.edu.cn;

张 汉 (1987–), 男, 博士, 目前研究方向为反应堆系统优化、反应堆安全分析、启发式算法和物理热工耦合方法, E-mail: han.zhang@kit.edu.