

改进的粒子群算法在轧制负荷分配中的优化

李荣雨 张卫杰 周志勇

(南京工业大学计算机科学与技术学院 南京 211816)

摘要 针对带钢热连轧精轧机组中负荷分配的优化问题,提出一种基于经验的自适应双层粒子群优化算法(ADLPSO-EM)。每次种群迭代后,对记忆群体通过改进的更新公式进行更新。利用改进的自适应调整惯性权重的策略充分增强种群的多样性,提高全局搜索能力。最后,在将其应用于热连轧负荷分配问题时,通过以经验法得到的值产生一个搜索邻域,并通过变邻域求出最后的负荷分配。仿真结果表明,改进的算法对负荷分配优化具有明显的效果。

关键词 负荷分配,粒子群优化,记忆群体,自适应调整,经验法,变邻域

中图分类号 TP18 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.07.037

Improved PSO Algorithm and Its Load Distribution Optimization of Hot Strip Mills

LI Rong-yu ZHANG Wei-jie ZHOU Zhi-yong

(College of Computer Science and Technology, Nanjing Tech University, Nanjing 211816, China)

Abstract Aiming at the load distribution problem of hot strip rolling, an adaptive double layer particle swarm optimization algorithm based on empirical method (ADLPSO-EM) was proposed. After each population iteration, the algorithm uses improved speed update formula to update memory swarm. At the same time, in order to improve the diversity of the population, it uses an improved adaptive adjustment strategy to update inertia weight. Finally, The initialization section of the algorithm is a changeable neighborhood based on the value obtained by the empirical method in load distribution problem. The experimental results show that the improved algorithm has a significant effect on the load distribution optimization.

Keywords Load distribution, Particle swarm optimization, Memory swarm, Adaptive adjustment, Empirical method, Changeable neighborhood

1 引言

热连轧中的负荷分配问题对提高产品质量、减少能耗具有决定性意义,因此对于负荷分配问题的优化一直都是研究热点之一。由于早期技术条件和计算水平的限制,负荷分配制定的方法主要采用基于经验的能耗曲线法^[1]。随着计算机技术的普及和最优化技术的成熟,借助计算机高效的计算能力,人们将最优化技术应用于轧钢的负荷分配中以进一步降低生产能耗和提高产品质量^[2]。20 世纪 80 年代,人工智能技术得到快速发展。由于人工智能技术所显现的优越的性能,将智能算法应用到热连轧负荷分配问题中成为研究热点之一。

粒子群优化算法^[3](Particle Swarm Optimization, PSO)是智能优化算法中重要的一个分支,其因结构易于理解且没有较多的参数需要设置等优势,在应用中得到了较多的青睐。该算法常被用于解决函数优化问题、机器学习、模糊系统控制以及众多的工程应用问题^[4]。Sun J 等人^[5]利用粒子群算法有效地解决了电力系统中的调度问题,极大地减小了系统的损耗,带来了极大的经济效益;此外,该算法还被成功应用于任务分配^[6]、无人机三维轨迹规划^[7]、射频识别网络规划^[8]等

许多大型的工程和科学领域,成为当今群体智能优化算法中重要的算法之一。为解决各种改进后的 PSO 算法仍然存留原有的缺陷问题,一些研究者从基本粒子群算法原理本身研究,通过改变算法的结构和更新机制以达到令人满意的效果。Li 等人^[9]提出了一种 SLPSO 算法,为粒子更新提供了具有 4 个状态的自适应学习框架,具有良好的全局优化性能。Wei Hong Lim 等人^[10]将粒子群分为当前群体和记忆群体,对每种群体采用不同的更新策略,极大地改善了算法的寻优质量,提高了算法的收敛效率。Chen^[11]改变了速度更新公式中全局极值更新的机制,采用了一种领导者和挑战者衰老换代的方式,即有利的领导粒子会增长担任年限,竞争者会取代领导不利的全局最优粒子,该方法有效地避免了前期全局极值对粒子群不利的影响。然而无论采用哪种改进方法,粒子群算法在实际应用中仍然存在易早熟收敛及进化后期收敛速度较慢的问题。

为了提高粒子群算法在轧制负荷分配优化中的性能,提出一种基于经验的变邻域自适应双层粒子群优化算法(ADLPSO-EM)。针对 PSO 算法的缺点,通过两种思路对粒子群算法进行了有效改进。第一种改进思路结合了参数设置

到稿日期:2017-02-28 返修日期:2017-03-30 本文受江苏省高校自然科学基金(12KJB510007)资助。

李荣雨(1977—),男,博士,副教授,主要研究方向为工业过程的优化与监控,E-mail:alleric@yeah.net(通信作者);张卫杰(1990—),男,硕士生,主要研究方向为机器学习方法在工业过程中的应用,E-mail:alleric@yeah.net。

Image Processing, 2013, 22(2): 687-699.

- [15] CHEN G Y, ZHU F Y, HENG P A. An efficient statistical method for image noise level estimation[C]// IEEE International Conference on Computer Vision. New York: IEEE Press, 2015: 477-485.
- [16] IMMERKAR J. Fast noise variance estimation[J]. Computer Vision & Image Understanding, 1996, 64(2): 300-302.
- [17] ZORAN D, WEISS Y. Scale invariance and noise in natural images[C]// IEEE International Conference on Computer Vision. New York: IEEE Press, 2009: 2209-2216.
- [18] LIU W, LIN W S. Additive white Gaussian noise level estimation in SVD domain for images[J]. IEEE Transactions on Image Processing, 2013, 22(3): 872-883.
- [19] YANG S M, TAI S C. Fast and reliable image noise estimation using hybrid approach[J]. Journal of Electronic Imaging, 2010, 19(3): 3007.
- [20] LIU X H, TANAKA M, OKUTOMI M. Noise level estimation using weak textured patches of a single noise image[C]// IEEE International Conference on Image Processing. New York: IEEE Press, 2012: 665-668.
- [21] LIU X H, TANAKA M, OKUTOMI M. Single-image noise level estimation for blind denoising[J]. IEEE Transactions on Image Processing, 2013, 22(12): 5226-5237.
- [22] ZHANG L, ZHANG L, BOVIK A C. A feature-enriched completely blind image quality evaluator[J]. IEEE Transactions on Image Processing, 2015, 24(8): 2579-2591.
- (上接第 218 页)
- 性,利用记忆群体的更新在不直接影响当前群体多样性的情况下加速了算法收敛,针对负荷分配应用,提出了以经验法为基准的变邻域算例。使用某钢厂的实际数据,利用改进的算法在 MATLAB 上对负荷分配优化问题进行仿真,通过与经验法及 ATLP SO-ELS 算法结果的对比,表明该算法对于带钢热轧连轧的负荷分配问题的优化是可行有效的,且算法的时效性有利于实时系统的在线自适应。
- ### 参 考 文 献
- [1] 孙一康. 带钢热轧连轧的模型与控制[M]. 北京:冶金工业出版社, 2002.
- [2] LI H J, XU J Z, WANG G D. Improvement on conventional load distribution algorithm in hot tandem mills[J]. Journal of Iron and Steel Research, International, 2007, 14(2): 36-41.
- [3] KENNEDY J, EBERHART R C. Particle Swarm Optimization [C]// IEEE International Conference on Neural Networks. Piscataway, 1995: 1942-1948.
- [4] KHARE A, RANGNEKAR S. A review of particle swarm optimization and its applications in Solar Photovoltaic system[J]. Applied Soft Computing, 2013, 12(5): 2997-3006.
- [5] SUN J, PALADE V, WU X J, et al. Solving the Power Economic Dispatch Problem With Generator Constraints by Random Drift Particle Swarm Optimization[J]. IEEE Transactions on Industrial Informatics, 2014, 10(1): 222-232.
- [6] HO S Y, LIN H S, LIAUH W H, et al. OP SO: Orthogonal Particle Swarm Optimization and Its Application to Task Assignment Problems[J]. IEEE Transactions on Systems Man and Cybernetics Part A: Systems and Humans, 2008, 38(2): 288-298.
- [7] FU Y G, DING M Y, ZHOU C P. Phase Angle-Encoded and Quantum-Behaved Particle Swarm Optimization Applied to Three-Dimensional Route Planning for UAV[J]. IEEE Transactions on Systems Man and Cybernetics Part A: Systems and Humans, 2012, 42(2): 511-526.
- [8] GONG Y J, SHEN M, ZHANG J. Optimizing RFID Network Planning by Using a Particle Swarm Optimization Algorithm with Redundant Reader Elimination[J]. IEEE Transactions on Industrial Informatics, 2012, 8(4): 900-912.
- [9] LI C, YANG S, NGUYEN T T. A self-learning particle swarm optimizer for global optimization problems[J]. IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics, 2012, 42(3): 627-646.
- [10] WEI H L, ISA N A M. An adaptive two-layer particle swarm optimization with elitist learning strategy [J]. Information Sciences, 2014, 273(3): 49-72.
- [11] CHEN W N, ZHANG J, LIN Y, et al. Particle Swarm Optimization With an Aging Leader and Challengers[J]. IEEE Transactions on Evolutionary Computation, 2013, 17(2): 241-258.
- [12] HAN J H, LI Z R, WEI Z C. Adaptive Particle Swarm Optimization Algorithm and Simulation[J]. Journal of System Simulation, 2006, 18(10): 2969-2971. (in Chinese)
韩江洪, 李正荣, 魏振春. 一种自适应粒子群优化算法及其仿真研究[J]. 系统仿真学报, 2006, 18(10): 2969-2971.
- [13] WEI H L, ISA N A M. Two-layer particle swarm optimization with intelligent division of labor[J]. Engineering Applications of Artificial Intelligence, 2013, 26(10): 2327-2348.
- [14] EPITROPAKISA M G, PLAGIANAKOS V P, VRAHATIS M N. Evolving cognitive and social experience in Particle Swarm Optimization through Differential Evolution: A hybrid approach [J]. Information Sciences, 2012, 216(24): 50-92.
- [15] WEI H L, ISA N A M. An adaptive two-layer particle swarm optimization with elitist learning strategy[J]. Information Sciences, 2014, 273(3): 49-72.
- [16] SHI Y, EBERHART R C. A Modified Particle Swarm Optimizer [C]// Proceedings of the IEEE Conference on Evolutionary Computation. Piscataway, 1998: 69-73.
- [17] ZHAN Z, ZHANG J, LI Y. Adaptive particle swarm optimization[J]. IEEE Transactions on Systems, Man, Cybernetics B: Cybernetics, 2009, 39(6): 1362-1381.
- [18] WANG Y, LIU J L, SUN Y K. Immune Genetic Algorithms (IGA) Based Scheduling Optimization[J]. Journal of University of Science and Technology Beijing, 2002, 24(3): 339-341. (in Chinese)
王焱, 刘景录, 孙一康. 免疫遗传算法对精轧机组负荷分配的优化[J]. 北京科技大学学报, 2002, 24(3): 339-341.

策略和种群更新机制两方面,第二种改进思路将一种全新的思想引入粒子群算法中,提出了全新的速度更新公式。本文将改进的算法应用到负荷分配问题的优化中,并通过实例仿真验证了它的有效性。

2 ADLPSO-EM 算法

2.1 基本 PSO 算法

算法首先随机初始化粒子群体,粒子群中的每个粒子 P_i 代表待优化问题的潜在解。每个粒子均有自己的状态:速度 $V_i = (v_{i,1}, v_{i,2}, \dots, v_{i,d})$ 和位置 $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$, 其中 d 表示维度(待解问题的未知数个数)。 X_i 对应的目标函数值即可作为该粒子的适应度,算法通过适应度来衡量粒子的优劣。在每一次迭代中,每个粒子都记忆着两个位置: P_i 自身目前找到的最好位置 $Pbest_i$ 和整个群体目前找到的最好位置 $Gbest$ 。每个粒子通过跟踪这两个位置来更新自己的位置和速度。在下一迭代($t+1$)中,粒子 P_i 的速度和位置更新公式如式(1)、式(2)所示。

$$V_{i,d}(t+1) = \omega \cdot V_{i,d}(t) + c_1 \cdot r_1 \cdot (Pbest_{i,d}(t) - X_{i,d}(t)) + c_2 \cdot r_2 \cdot (Gbest_d(t) - X_{i,d}(t)) \quad (1)$$

$$X_{i,d}(t+1) = X_{i,d}(t) + V_{i,d}(t+1) \quad (2)$$

其中, c_1 和 c_2 是加速因子, r_1 和 r_2 是两个(0,1)之间的随机数。

PSO 算法迭代终止条件为预先确定的最大迭代次数,或者是优化结果满足了其精度要求。算法终止时的全局极值 ($Gbest$) 即为最后的最优解。

2.2 改进的自适应惯性权重

轧制负荷分配优化问题最重要的是提高优化精度。为了避免利用粒子群算法优化该问题时发生早熟收敛,本文通过引入自适应的惯性权重^[12]来增强群体的多样性,同时对自适应发生条件进行改进。该种自适应操作是通过评价每个粒子的好坏程度,给予不同的粒子适合的惯性权重,各自完成相应的任务。该方法增强了粒子群的多样性特点,平衡了 PSO 算法的全局搜索能力及局部寻优能力。图 1 给出了粒子群分类的方法。

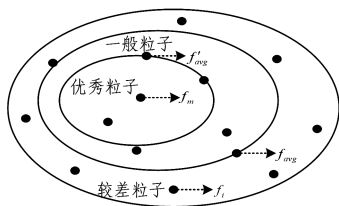


图 1 粒子分类图

Fig. 1 Particle classification

图 1 中, f_i 是粒子 P_i 的适应度值,粒子群的平均适应度值表示为 f_{avg} , f'_avg 是优于平均适应度值的粒子的平均适应度值, f_m 表示当前群体最优粒子的适应度值。本文利用文献^[12]中的计算方法设置每种粒子的惯性权重。

然而当前这种自适应操作是在算法进入早熟收敛阶段进行的,仅当在这种阶段下进行自适应操作会使得种群的多样性不能得到充分的发挥。所提算法取消了发生自适应过程的条件,使得种群进化和自适应操作同步进行,充分发挥了自适应特性,增加了群体的多样性。

2.3 记忆群体更新公式及其作用

对于粒子群算法的优化大多都是一种针对当前群体进化的改进,很少有人通过研究粒子群的记忆群体(即粒子的个体极值所组成的群体)来改善粒子群算法优化的性能。然而最新研究显示,当前群体和记忆群体在搜索过程中展现了不同的特性^[13-15]。具体来说,当前粒子群趋向于开拓搜索空间,找到最优解的大致位置;记忆粒子群趋向于探索最优解,利用双层粒子群的特性可以使算法得到更好的优化效果。

对于当前群体的更新,每个粒子通过跟踪两个极值来更新自己的位置。而记忆粒子是当前粒子的个体极值,粒子本身没有属于它的记忆粒子,它只能通过全局的记忆粒子也即全局极值来更新自己的位置。同时,记忆粒子的产生是当前粒子在更新过程中精英策略的选择,记忆本身是没有速度的。因此,记忆群体的更新公式如式(3)所示:

$$X_{i,d}(t+1) = X_{i,d}(t) + c_3 \cdot r_3 \cdot (Gbest_d(t) - X_{i,d}(t)) \quad (3)$$

由于更新后位置的好坏直接影响着我们是否确认这一次更新,因此需要对更新后的位置进行优劣性评价,通过一个中间变量存储记忆粒子更新的位置,用 $R_{i,d}^{emp}$ 替代 $X_{i,d}(t+1)$,因此式(3)变为式(4):

$$R_{i,d}^{emp} = X_{i,d}(t) + c_3 \cdot r_3 \cdot (Gbest_d(t) - X_{i,d}(t)) \quad (4)$$

其中, R_i 表示第 i 个粒子的记忆粒子位置(个体极值)。如果 $f(R_i^{emp}) < f(R_i)$,更新操作将得到确认,此次进化成功, X_i 的位置将更新为 R_i^{emp} ;否则,此次将不进行对该记忆粒子的更新。

通过 2.2 节可知,加入改进的自适应惯性权重的粒子群算法能够有效地避免局部最优的产生,从而得到全局最优解。然而在实际应用中,往往需要算法满足一定的时效性。一般的改进算法难以同时兼顾收敛精度与收敛速度,这是因为这些算法在提高收敛速度的同时往往牺牲了群体的多样性。相反,在提高收敛精度时往往需要粒子做出更多的工作,从而导致算法的收敛时间变长。

由于记忆粒子的更新是针对粒子的个体极值进行操作的,很少对当前群体产生直接的影响,因此通过记忆粒子群的更新加速算法收敛速度的操作,使当前群体的多样性仍能得到很好的保持。为了满足粒子群在实际应用中的实时性,文中将记忆群体的更新机制引入改进的自适应惯性权重粒子群算法中。

2.4 粒子的初始化及搜索区域

本文将使用基于经验分配法的改进粒子群算法对负荷分配进行优化。其基本思想是:通过经验分配法得到一组初始的压下量分配,并以该值作为基准值产生一个邻域区域,在该区域内随机初始化粒子群并运用改进的粒子群算法进行寻优。因为经验法分配的结果和理论最优分配的结果相差不大,所以将经验分配法得到的压下量作为基准值。首先算法在开始时随机初始粒子群使其位于基准值的邻域内,然后对该群体利用提出的改进粒子群算法进行寻优。利用该方式初始粒子群体,一方面要求初始值满足约束条件,能够进行优化,若不以该基准值随机初始粒子,则很多初始的粒子都不能满足约束条件;另一方面,通过以经验法得到的负荷分配值为基准而产生的随机初始粒子群,能够使得该群体大致分布于最优解的附近,加速算法收敛于全局最优。

本文针对负荷分配问题对目标函数进行优化,其优化搜索区间设置为以经验法得到的基准值的邻域范围内。利用改进的算法对其进行优化,首次优化该搜索范围时应该设置一个比较大的范围,若一次算法结束后没有得出解或解不满足要求,则其原因是优化算法没有找到最优解或者得出的解不满足约束条件,试图在该区域重新优化一次,以免错过该区域内的最优解。若仍然没有找到最优解,为了算法的效率,此时应减小该邻域范围,重新进行搜索优化。重复此过程直至找出最优解,若搜索区域变为0倍的基准值邻域,则最优解仍然为经验法得到的值,算法失败。若在某一邻域内找到满足要求的值,则停止算法搜索,输出优化后的最优解。以上算例称为变邻域的算法。

2.5 ADLPSO-EM 算法步骤

下面给出利用 ADLPSO-EM 算法求解负荷分配问题中某一目标函数的优化步骤:

- (1) 获得轧制规程中需要的各项已知参数。
- (2) 确定使用的目标优化函数。
- (3) 通过经验法计算各机架出口厚度的基准值。
- (4) 确定厚度的初始邻域范围,将其作为优化目标函数的范围,设该邻域为 $[-\delta h_i, \delta h_i]$ 。
- (5) 在该邻域内随机初始化粒子群。
- (6) 使用改进后的算法对目标函数在上述邻域内随机搜索优化。
- (7) 检验终止条件是否满足,若满足则继续,否则转向步骤(6)。
- (8) 若 $\delta=0$,则算法失败;否则检验给定的约束条件;若满足约束条件则输出结果 P_i ,否则使邻域倍数 $\delta-1$,同时转步骤(4)。

3 收敛程度的判定标准

为了正确判断当前种群的收敛程度,很多学者提出了自己的方法。SHI Y 通过迭代次数来大致反映种群的收敛程度^[16]。然而该判定标准难以正确地知道迭代次数与种群收敛程度之间的具体关系,因为对于不同问题,算法早熟收敛时迭代的次数是不一样的,使得通过迭代次数进行自适应的粒子群算法不具有普适性。为了寻找更加精确的判定标准,研究者又提出了平均适应度的判定标准。文献[12]通过统计粒子的适应度情况来判断种群的收敛程度,当检测到种群发生早熟收敛时,对惯性权重进行自适应调整,其合理性在于:当趋于收敛时,种群必将靠近最优值,使得粒子对应的函数值变小,适应度值也将变小,种群的平均适应度很好地反映了整个群体的聚集情况。但是这种判定标准在处理函数优化问题时,若该函数有多处极值点,并且恰巧种群的几个子群体分布在不同的极值点附近,此时从适应度方面来看,粒子适应度值都很小,但这几个子种群又相距很远,这种判定标准会错误地认为整个种群全部收敛于最优解附近,而事实并非如此。为了避免这种实际种群距离分散而得出错误的结论,Zhan 等人^[17]提出了一种“进化因素”的判定标准,其中粒子间平均距离的计算是核心。通过平均距离指标确实可以避免上述误导,但这种方法的计算量相当大。当问题的维度较高时,巨大

的计算量严重影响了算法的收敛效率,这对于算法应用在实时系统中是难以接受的。因此,找出一种既能正确反映种群的收敛程度,又能不增加算法的时间复杂度的收敛程度的判定标准,是一个迫切需要解决的问题。

本文提出了粒子的平均速度指标来衡量种群的收敛程度。平均速度的计算公式如下所示:

$$V_{avg}(t) = \frac{\sum_{i=1}^N \sum_{j=1}^D |v_{ij}(t)|}{N} \quad (5)$$

其中, t 为当前迭代次数, N 为种群个数, D 为粒子的维度, V_{ij} 为第 i 个粒子的第 j 维速度, V_{avg} 为种群的平均速度。

通过式(5)来判定种群的收敛程度,其原因在于:根据种群的运动趋势,群体将会靠近个体极值或者全局极值位置,此时 $Pbest-X$ 或 $Gbest-X$ 将会变得很小,根据式(1)可知,种群的速度也将随之减小。当发生收敛时,种群的所有粒子速度将达到最小。因此通过平均速度可以正确反映群体的收敛程度。

因为每次算法迭代都要求出下一步更新需要的速度,所以通过平均速度判定收敛程度的指标并不需要额外计算其他数据,仅仅是对元数据进行简单的求和及平均数的操作,算法的时间复杂度并未因此而提高。

综上,这种判定标准既能正确反映种群的收敛程度,又不会影响算法的时间复杂度,将平均速度作为判定粒子收敛程度的指标是一个很好的方法。

4 自适应惯性权重

通过第3节提出的收敛程度评价标准,算法可以很容易地对各个粒子进行收敛性评价。种群中必然会出现3种群体:收敛程度高的粒子、发散性强的粒子、介于两者之间的粒子。为了解决这两个问题,通过上述收敛程度判定标准将粒子群分为3个子群体,每个子群体采用最适合的惯性权重调整策略。具体做法如下:

假设第 i 个粒子的速度为 V_i , V_{avg} 是整个种群的平均速度, V_{min} 是种群中速度最小的一个粒子, V'_{avg} 是位于 V_{avg} 和 V_{min} 之间粒子的平均值。图1的粒子分类图就是根据上述原理得出的。

3个子种群的惯性调整策略为:

$$(1) V_{min} < V_i < V'_{avg}$$

该子群的粒子速度是整个种群中最小的,反映了该群体具有较高的收敛程度,此时子群接近于全局最优位置,把这些粒子称为优秀粒子群。为了加强该种群向全局收敛的速度,惯性权重应该被给予一个较小的值。本文根据粒子的速度值给出式(6)来调整惯性权重大小。

$$\omega = \omega - (\omega - \omega_{min}) \cdot \left| \frac{V_i - V'_{avg}}{V_{min} - V'_{avg}} \right| \quad (6)$$

由式(6)可以看出,速度越小的粒子即越优秀的粒子会被赋予一个越小的惯性权重值,很好地强化了粒子局部寻优的特点。

$$(2) V'_{avg} < V_i < V_{avg}$$

该子群的粒子速度是整个种群中速度一般的粒子,被称为一般粒子群。该种群的粒子一方面收敛程度不会太高,种

群的分散性有利于算法的全局的探索。另一方面,该种群的粒子发散性不会太强,种群的聚集性有利于算法的局部开发。因此一般粒子群能有效地平衡全局搜索与局部寻优的能力。此种类的粒子我们将不改变其惯性权重的大小即保持原先的惯性权重值。

$$(3) V_i > V_{avg}$$

该子群的粒子速度大于种群速度的平均值,这些粒子是比较发散的,称该种粒子为劣等粒子。通过劣等粒子能够间接反映粒子群的收敛性,若劣等种群的平均速度很小,则其他两个子种群的速度会更小,种群收敛程度高,因此优秀粒子群的收敛可以作为种群收敛的必要条件。为了不增加算法的额外计算量,利用式(6)中的关于优秀粒子群的收敛程度指标 $|V_{min} - V'_{avg}|$ 可以反映劣等种群的收敛情况。通过这种收敛程度指标给出劣等种群的自适应惯性权重调整公式,如式(7)所示。

$$\omega = 1.5 - \frac{1}{1 + k_1 \cdot \exp(-k_2 \cdot |V_{min} - V'_{avg}|)} \quad (7)$$

从式(7)可以看出,当种群在空间中的分布密度越大时, $|V_m - V'_{avg}|$ 会越小。通过式(7)的自适应使得 ω 的值变大,从而可以提高粒子的全局搜索能力。当群体聚集到一定程度时, ω 会被赋予一个大于 1 的数,通过一个大于 1 的 ω 使种群跳出局部最优位置。当种群较为发散时, $|V_m - V'_{avg}|$ 的值较大,由式(7)可知惯性权重将被减小,达到了增强粒子局部寻优能力的目的。其中 k_1 的大小决定了 ω 的范围, k_2 的大小控制了 ω 的变化速率。

5 负荷分配模型的建立

精轧机组的轧制规程可以分为 3 个阶段^[18],由于前期阶段带钢温度高,容易产生塑性变形,因此在轧制中把压下量的大部分安排在一、二阶段。具体地,1,2 机架作为第一阶段,该阶段需要充分利用设备的条件并且保证负荷均衡,同时考虑到带钢咬入困难和厚度的波动,第 1 机架的压力会略小于后面的机架;第二阶段包括 2,3 两个机架,因为第二阶段不用考虑咬入问题,所以 2,3 机架要尽可能地进行压下,并且保持均衡;第三阶段带钢的厚度基本接近产品要求,此时获得良好的板型是重点,该阶段的主要任务在于合理地分配压下量以获得好的带钢相对凸度。3 个目标函数如式(8)一式(10)所示:

第一阶段目标函数:

$$F_1 = \min\{(P_1 - K_1 P_2)^2\} \quad (8)$$

第二阶段目标函数:

$$F_2 = \min\{(P_2 - K_2 P_3)^2\} \quad (9)$$

第三阶段目标函数:

$$F_3 = \min\left\{\sum_{i=4}^n \left(\frac{CR_i}{h_i} - \frac{CR_n}{h_n}\right) \pm \Delta_i\right\}^2 \quad (10)$$

其中, $P_i (i=1,2,3)$ 表示第 i 机架的压力; K_1 和 K_2 为比例系数,一般 $K_1=0.9, K_2=1$; CR_i/h_i 表示第三阶段经过第 i 个机架的相对凸度, CR_n/h_n 为成品的相对凸度, Δ_i 为最优调节量。

因此最终的目标函数为:

$$F = \min\{F_1 + F_2 + \lambda F_3\} \quad (11)$$

实际生产中,各机架的负荷取值会受到设备能力和工艺流程的影响,这些影响共同构成了上述目标函数的约束条件。这些约束包括(影响较小的约束忽略不计):

$$\begin{cases} 0 \leq P_i \leq P_{max} \\ 0 \leq M_i \leq M_{max} \\ h_{i+1} \leq h_i \end{cases} \quad (12)$$

其中, P 为轧制力约束, M 为凸度约束, h 为厚度约束。

6 仿真及实例分析

为了验证本文算法的有效性,将 ADLPSO-EM 算法与标准粒子群算法(PSO)、文献[17]提出的 APSO 算法以及 Wei Hong Lim 等人提出的 ATLPSO-ELS 算法^[10]作比较。算法的参数设置如表 1 所列。

表 1 各算法参数设置

Table 1 Parameter setting for each algorithm

算法	主要参数设置
SPSO	$\omega: 0.9, c_1 = c_2 = 2.0$
APSO	$\omega: 0.9-0.4, c_1 + c_2: [3.0, 4.0]$
ATLPSO-ELS	$\omega: 0.9-0.4, c=2.0, k_1=0.4, k_2=0.8$
ADLPSO-EM	$\omega: 0.75, c_1 = c_2 = 2.0, k_1 = 1.5, k_2 = 0.3$

对这些算法进行测试时,设置种群规模 $N=30$, 维度 $D=100$, 算法的终止条件为算法的最大迭代次数达到 3000 次,每种算法对每个函数优化时重复执行 30 次。表 2 列出了 4 种算法在 4 个测试函数上的 3 个指标数据。

表 2 测试指标对比

Table 2 Comparison of test indicators

函数名	算法	MeanBst	OptV	Std
Rosenbrock	SPSO	388.774	7.0316	1.789e+003
	APSO	26.7518	26.3446	0.2449
	ATLPSO-ELS	26.0539	25.8027	0.4926
	ADLPSO-EM	25.8850	24.3011	0.4844
Griewank	SPSO	0.0138	1.756e-010	0.0169
	APSO	0	0	0
	ATLPSO-ELS	0	0	0
	ADLPSO-EM	0	0	0
Rastrigin	SPSO	46.3240	20.8941	17.0059
	APSO	0	0	0
	ATLPSO-ELS	0	0	0
	ADLPSO-EM	0	0	0
Ackley	SPSO	0.9423	4.707e-006	3.5254
	APSO	4.441e-015	4.441e-015	0
	ATLPSO-ELS	4.4208e-015	3.7109e-015	6.3157e-016
	ADLPSO-EM	4.204e-015	8.881e-016	8.862e-016

通过表 2 可以看出,ADLPSO-EM 算法在 3 个测试指标上大体显示出较好的效果。具体地,在算法收敛精度方面,ADLPSO-EM 算法的平均最优值和最优值总体上达到了 APSO 算法的效果。在 Griewank 和 Rastrigin 函数中,除 SPSO 算法外其他 3 个算法均达到了最优值 0。需要注意的是,对于 Ackley 和 Rosenbrock 函数而言,虽然都没有达到全局最优,但是 ADLPSO-EM 算法得出的结果略好于 APSO 和 ATLPSO-ELS,在 Ackley 函数中得到的最优值也比这二者好了一个数量级,这说明所提算法在求解高维多峰问题时具有较为满意的效果。在稳定性方面,ADLPSO-EM 算法、ATLPSO-ELS 算法同 APSO 算法三者都得到了较好的效果,

在 Griewank 和 Rastrigin 两个函数中标准差达到了最好值 0。总体上,对于这 4 个典型函数的测试,本文算法的优化精度均有所提升。

同时,为了验证算法的收敛速度,图 2 给出了 4 种算法对 4 个测试函数的收敛曲线图。可以看出,ADLPSO-EM 算法通过对记忆粒子群的更新,使其在复杂问题求解时能够更快地收敛。例如,从对 Rastrigin 函数优化的曲线图中可以看出,SPSO 算法在 3000 次迭代内,曲线仍没有收敛。相反,ADLPSO-EM 算法在迭代 1200 次左右时,曲线图开始趋于稳定,比 APSO 算法的收敛速度提高了 66.6%。而 ATLPSO-ELS 算法尽管在 Rosenbrock 函数和 Rastrigin 函数上与本文算法相差较少,但是从另外两个函数尤其是 Ackley 函数可以明显看出,本文算法拥有更快的收敛速度。其余收敛曲线图也都显示了 ADLPSO-EM 算法在收敛速度上的优越性。因此,通过加入记忆粒子群的更新机制,算法的收敛速度不但得到了显著提高,而且对于求解高维空间中的多峰问题,在保证良好的优化结果下,极大地提高了算法的寻优速度。在对时间要求比较高的实时系统中,其也能保证结果的精确性,因此本文所提算法是一个很好的选择。

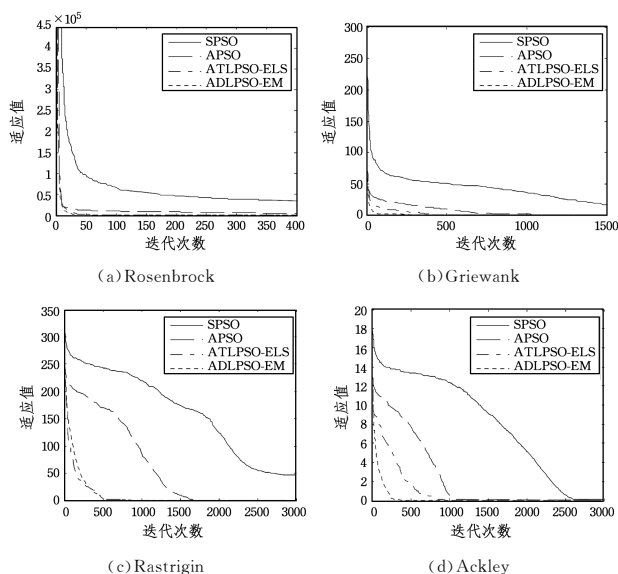


图 2 各算法的收敛曲线

Fig. 2 Convergence curve of each algorithm

为了验证 ADLPSO-EM 算法对负荷分配问题的有效性,使用了从现场采集的数据,用改进的粒子群算法 ADLPSO-EM 对热连轧中的负荷分配进行优化计算,同时与经验法分配以及 ATLPSO-ELS 算法优化的结果作对比。实验中,钢种为 Q235B 型带钢,带钢的初始宽度 $B=1520\text{mm}$,初始入口厚度 $H_0=36.3\text{mm}$,成品厚度 $H_7=5.9\text{mm}$,粗轧机出口温度 $T_{RC}=1061^\circ\text{C}$,板型要求的目标凸度 $CR_n=0.016\text{mm}$,机架数 $n=7$ 。

算法中,种群大小设置为 30 个,最大迭代次数为 3000。重复计算 20 次,取其中最好的优化结果为负荷分配设定。

根据负荷分配模型三阶段的要求,观察负载分配是否满足这些阶段的要求。通过实验得出了分配法结果比较,如表 3 所列。为了便于清晰地分析与观察表 3 中的数据,绘制出轧制力对比图与相对凸度对比图,分别如图 3 和图 4 所示。

表 3 分配法结果比较

Table 3 Comparison of distribution method results

方法	F1	F2	F3	F4	F5	F6	F7
经验法	20.93	18.30	22.56	16.67	12.01	11.33	7.96
ATLPSO-ELS	18.68	21.09	24.97	14.09	10.41	9.01	7.67
ADLPSO-EM	19.38	24.38	23.97	13.09	11.41	9.30	8.77
经验法	1.51	2.29	3.51	2.98	3.21	3.35	2.97
ATLPSO-ELS	1.55	2.71	3.45	3.02	3.11	3.16	3.02
ADLPSO-EM	1.55	2.81	3.35	3.06	3.04	3.05	3.05

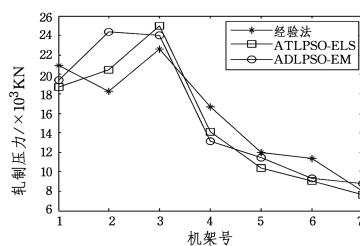


图 3 轧制力对比

Fig. 3 Rolling force contrast

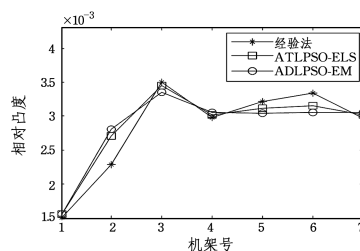


图 4 相对凸度对比

Fig. 4 Relative convexity contrast

从图 3 可以看出,ADLPSO-EM 算法分配的结果中,第一、二、三机架基本满足了本文第 5 节轧制规程的要求,具有较大的压力值,满足第一、二阶段对厚度控制的要求。3 个机架的负荷分配较均匀,并且第一机架的压力较小,使得咬入容易。而 ATLPSO-ELS 算法优化的结果显示,尽管第一机架的压力较小,满足工艺要求,但第二机架的轧制力却依然较小,与第三机架的压力相差较大,负荷分配不均,总体上不如 ADLPSO-EM 算法分配的结果。经验法的结果显示,虽然也具有压力较大的要求,但是在分配均衡上效果却不理想,并且第一机架也不能很好地满足咬入条件。在凸度方面,图 4 显示了 ADLPSO-EM 算法优化的结果,在第四、五、六、七机架中 ADLPSO-EM 优化的负荷分配使得凸度保持一致,基本维持在 3.05,表现出相当稳定的特性。而 ATLPSO-ELS 算法的优化结果尽管在第一、二、三机架中与本文算法相差不大,但是在第四、五、六、七机架中却出现了稍微的波动现象。经验法分配的波动现象更为明显,尤其对于三、四、五这 3 个轧制道次凸度的波动性很强,极差达到了 0.53。由此可知,利用 ADLPSO-EM 优化的负荷分配法更能充分发挥设备的能力以提高产品的质量。

结束语 针对带钢热连轧精轧机组中负荷分配的优化,提出了一种基于经验的改进自适应双层粒子群优化算法 (ADLPSO-EM)。利用改进的自适应策略加强了种群多样

(下转第 225 页)