

文章编号:1005-0523(2020)02-0056-08

改进粒子群算法的轨道列车节能控制优化

黄江平,程绍榕

(华东交通大学电气与自动化工程学院,江西 南昌 330013)

摘要:当前列车节能控制研究的优化算法存在优化效果不明显,收敛速度慢等一系列问题,提出了一种引入自适应惯性权重,同时加入具有调整能力的动态学习因子与改进速度更新公式的粒子群优化算法。一是通过惯性权重来平衡不同阶段的搜索能力,加入拥有调整能力的动态学习因子,着重加强算法后期的运算效率和收敛能力,同时引入惩罚函数,将列车运行过程中的约束条件转化为惩罚因子,提升搜索速率;二是对传统的速度更新公式进行改进,用来降低选取到不理想的粒子影响寻优结果的概率。经过 Matlab 仿真分析,与传统的列车运行优化算法相比,改进后的优化算法收敛速度更快,列车节能效果更好。

关键词:列车节能控制;粒子群;自适应惯性权重;动态调整学习因子;改进速度公式

中图分类号:U292.4+3

文献标志码:A

DOI:10.16749/j.cnki.jecjtu.2020.02.008

目前列车站间运行主要由4种状态组成:牵引、惰行、巡航和制动;因此,列车运行的节能优化问题,也通常被视为工况转换点的求解问题,尽可能在运行规定时间内减少牵引工况的能耗,逐渐成为轨道交通领域的主要研究方向。文献[1]利用惩罚函数的方法提出了无约束的节能驾驶模型,但是算法收敛时间较长,优化效果不是很明显。文献[2]在惩罚函数的基础上,利用粒子群算法代替遗传算法进行优化,调试参数少,收敛速度有了明显的提升,但是由于粒子群算法本身容易陷入局部最优,优化效果提升的不明显。文献[3]在列车运动方程的基础上,建立了列车牵引能耗的数学模型,构建了列车工况和总效率之间的径向基神经网络。文献[4]根据列车的运动情况和牵引特性曲线,采用遗传算法优化列车动车运行的惰行控制点来达到降低能耗的目的。列车运行作为一个非线性、多约束问题,智能优化算法在处理此类问题上具有明显优势。目前应用主要的智能优化算法有粒子群算法和遗传算法两大类,而文献[2]采用标准粒子群算法求取列车工况转换点,可能导致寻优陷入局部最优解和收敛速度慢的问题。

为解决上述问题,采用自适应惯性权重,动态学习因子,同时优化速度更新公式的方法来求解列车工况转换点,通过改变惯性权重,平衡不同阶段的搜索能力,同时加入可以自调整的学习因子,加强后期的运算效率和搜索能力,速度更新公式通过加入一项“参考项”,而“参考项”的参数由轮盘赌选择,从而避免不理想的粒子影响寻优结果,有利于跳出局部最优解。

1 列车控制优化问题的分析和建模

通常情况下,列车站间运行过程要按照规定的运行时刻表运行,运行时间和距离是固定的,由4种工况组成:牵引、巡航、惰行、制动^[5]。不同情况下,发动机的耗能情况如表1所示。

由上可知,要使列车实现能耗最少的目标,

表1 列车不同工况的能耗

Tab.1 Energy consumption under different operation modes for trains

列车运行工况	列车速度状态	发动机有无耗能
牵引	加速	耗能
巡航	匀速	不确定
惰行	不确定	不耗能
制动	减速	不耗能

收稿日期:2019-09-16

基金项目:江西省教育厅科技项目(GJJ160490)

作者简介:黄江平(1966—),男,教授,硕士,研究方向为高速列车操纵优化。

通讯作者:程绍榕(1996—),男,硕士研究生,研究方向为高速列车操纵优化。

尽可能的缩短牵引过程和巡航过程的能耗,但是存在规定时间,同时列车运行存在最高速度和最大加速度等限制,所以列车行驶的实际问题,转换为初始状态和终止状态已知,在规定时间内和速度限制范围内,求取不同工况下最优工况转换点的问题。

列车运行过程中,通过牛顿第二定律,得出运动学方程为

$$\frac{dv}{dt}=a=\frac{\mu_t F(v)-\mu_b B(v)-W_0(v)-G(x)}{M} \quad (1)$$

式中: v 为当前速度; a 为当前加速度; $F(v)$ 为当前时刻列车速度所对应的牵引力; $B(v)$ 为当前时刻列车速度所对应的制动力; $W_0(v)$ 为基本阻力; $G(x)$ 为坡度; M 为列车的总质量。

列车牵引力和制动力之间存在以下关系:

$$\left. \begin{aligned} 0 \leq \mu_t \leq 1 \\ 0 \leq \mu_b \leq 1 \end{aligned} \right\} \quad (2)$$

式中: μ_t 为牵引力系数; μ_b 为制动力系数。

另外,当列车在站间运行时,若运行总路程为 S ,运行总时间为 T 。则有

$$\left. \begin{aligned} v(0)=0 \\ v(S)=0 \\ t(S)-t(0)=T \end{aligned} \right\} \quad (3)$$

式中: $v(0)$ 为列车运行初速度; $v(S)$ 为列车运行末速度; $t(0)$ 为列车发车时刻; $t(S)$ 为列车到达终点时刻。

而列车在运行过程中,同时还要考虑乘客的舒适度和工况转换方面的拘束问题,故有

$$a \leq a_{\text{limit}} \quad (4)$$

式中: a_{limit} 表示乘客可以接受的最大加(减)速度。

同时列车 4 种工况间满足一定的转换规则,如表 2 所示。

“Y”表示可以直接转换,“N”表示禁止转换。

由于惰行阶段和制动阶段发动机是不耗电的^[6],所以主要在运行过程中对牵引阶段和巡航阶段的发动机能耗进行分析。同时,站间运行按照“牵引-巡航-惰行-制动”的策略进行运行^[7]。由于牵引力 F 和速度 v 的关系是根据列车特性曲线得出,并不可导,在这里,采用离散化的思想,

将列车运行区间等距划分为 $n(n=10)$ 个区间,对动力学模型进行差分计算,取时间步长 $\Delta t=1$,设定 4 种不同工况下每个时间步长 Δt 对应的加速度分别为 a_i, a_i', a_i'', a_i''' 。为了计算方便,采用延迟的思想,每一个时间步长 Δt 内加速度恒定,且在 4 种不同工况的情况下,第 i 个时间步长对应的速度分别为 v_i, v_i', v_i'', v_i''' ,对应合力分别为 c_1, c_2, c_3, c_4 ;对应的位移分别为 s_i, s_i', s_i'', s_i''' 。

根据牛顿第二定律,列车在牵引过程中单位时间步长对应的加速度 a_i 计算公式如下:

$$\left. \begin{aligned} c_1=f(v)-f_0(v)-f_w \\ a_i=\frac{c_1}{M} \end{aligned} \right\} \quad (5)$$

式中: $f(v)$ 为某一运行速度下列车的牵引力; $f_0(v)$ 为单位基本阻力; f_w 为因线路产生的附加阻力之和。

牵引过程中,速度与距离存在以下关系:

$$\left. \begin{aligned} v_{i+1}=v_i+a_i*\Delta t \\ S_{i+1}=S_i+v_i*\Delta t+a_i*\frac{\Delta t^2}{2} \end{aligned} \right\} \quad (6)$$

当列车牵引到目标速度时,进入巡航模式,巡航模式下,速度恒定,加速度为 0,在巡航过程中单位时间

表 2 工况转换拘束

Tab.2 Constraints of operation mode transition

工况	牵引	惰行	制动
牵引	Y	Y	N
惰行	Y	Y	Y
制动	N	Y	Y

步长对应的加速度 a_i' 计算公式如下:

$$\left. \begin{aligned} c_2 &= f(v) - f_0(v) - f_w = 0 \\ a_i' &= \frac{c_2}{M} \end{aligned} \right\} \quad (7)$$

巡航过程中, 速度距离关系为

$$\left. \begin{aligned} S'_{i+1} &= S_i + v_i' * \Delta t \\ v'_{i+1} &= v_i' \end{aligned} \right\} \quad (8)$$

惰行工况下, 无牵引力产生, 单位时间步长对应的加速度 a_i'' 计算公式如下:

$$\left. \begin{aligned} c_3 &= -f_0(v) - f_w \\ a_i'' &= \frac{c_3}{M} \end{aligned} \right\} \quad (9)$$

惰行过程中, 速度距离关系为

$$\left. \begin{aligned} v''_{i+1} &= v_i'' + a_i'' * \Delta t \\ S''_{i+1} &= S_i'' + v_i'' * \Delta t - a_i'' * \frac{\Delta t^2}{2} \end{aligned} \right\} \quad (10)$$

惰行到某一时刻时, 列车进入制动状态, 发动机仍然不做功, 列车在受到阻力的基础上增加了制动力, 其单位时间步长对应的加速度 a_i''' 如下:

$$\left. \begin{aligned} c_4 &= -f_0(v) - f_w - f_b \\ a_i''' &= \frac{c_4}{M} \end{aligned} \right\} \quad (11)$$

其中, f_b 表示列车的制动力。此时列车速度与距离的关系为

$$\left. \begin{aligned} v'''_{i+1} &= v_i''' + a_i''' * \Delta t \\ S'''_{i+1} &= S_i''' + v_i''' * \Delta t - a_i''' * \frac{\Delta t^2}{2} \end{aligned} \right\} \quad (12)$$

根据上述关系式, 我们可以确定列车 4 种工况下, 速度、时间、运行距离的关系。区间总路程为 S , 区间运行总时间为 T , 牵引、巡航、惰行、制动各自对应的运行距离为 s_1, s_2, s_3, s_4 。4 种工况对应运行时间为 t_1, t_2, t_3, t_4 , 在约束条件内, 存在:

$$\left. \begin{aligned} S &= s_1 + s_2 + s_3 + s_4 \\ T &= t_1 + t_2 + t_3 + t_4 \end{aligned} \right\} \quad (13)$$

固定站间运行总时间不变, 由于发动机在惰行过程和制动过程中不做功, 只需要计算牵引与巡航过程所做的功, 寻找牵引与巡航能耗最小的工况转换点。又因为牵引力做功公式为

$$E = F * S \quad (14)$$

得到在第 i 个时间步长内 Δt 所对应的牵引力 $f(v_i)$ 所做的功 ΔE_i 为

$$\Delta E_i = f(v_i) * \Delta t * v_i \quad (15)$$

将列车运行过程中每一个时间步长的牵引能耗进行累加, 得到发动机能耗公式如下:

$$E = \sum_{k=0}^{n-1} \Delta E_k = \sum_{k=0}^{n-1} f(v_k) * v_k * \Delta t \quad (16)$$

由此可确定列车运行能耗运行模型为

$$\left. \begin{aligned} \text{目标函数: } E &= \sum_{k=0}^{n-1} \Delta E_k (0 \leq k \leq n-1) \\ \text{约束条件: } 0 &\leq v_i \leq v_{\text{limit}}, 0 \leq a_i \leq a_{\text{limit}} \\ v_0 &= 0, v_n = 0 \end{aligned} \right\} \quad (17)$$

2 改进粒子群算法

2.1 粒子群算法概述

粒子群算法依靠不同粒子的相互作用寻找相关问题在整个寻优空间中的优位置,即问题的优解。在粒子群算法中,每个待解决的优化问题可能解都被视为搜索空间中存在的一个粒子。粒子群中的所有粒子都被赋予一个由被优化的函数确定的适应度值(fitness value),每个粒子具有 2 个属性,即速度与位置。在算法运行过程中,所有粒子趋向当前时刻的最优粒子的位置,并试图在可能空间中搜索全局优解。

假设在一个 N 维空间进行搜索,粒子 i 的信息可用 2 个 N 维向量来表示:第 i 个粒子的位置可表示为 $x_i=(x_{i1},x_{i2},\dots,x_{in})^T$,速度为 $v_i=(v_{i1},v_{i2},\dots,v_{in})^T$ 。在找到 2 个最优解后,粒子即可根据式(18)来更新自己的速度和位置:

$$\left. \begin{aligned} v_{id}^{k+1} &= \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (g_{id}^k - x_{id}^k) \\ x_{id}^{k+1} &= x_{id}^k + v_{id}^{k+1} \end{aligned} \right\} \quad (18)$$

式中: c_1, c_2 为学习因子,合适的 c_1, c_2 既可以加快收敛速度,又可以使结果不易陷入局部最优; r_1, r_2 是介于 $[0, 1]$ 之间的随机数; k 为迭代次数; ω 称之为惯性权重,它是一个比例因子,较大的 ω 可以加强 PSO 的全局搜索能力,而较小的 ω 则可以加强局部搜索能力,也就是说 ω 执行了全局搜索和局部搜索的平衡角色。

2.2 自适应惯性权重

在标准的粒子群算法中,使用最多的是线性惯性权重的方法,其特性为初期惯性权重较大,有利于进行全局搜索,后期惯性权重小,有利于局部寻优。它的关系式如下:

$$\omega(k) = \omega_{\max} - k(\omega_{\max} - \omega_{\min}) / T_{\max} \quad (19)$$

式中: k 为当前迭代次数; ω_{\max} 为最大惯性权重; ω_{\min} 为最小惯性权重; T_{\max} 为最大迭代次数。

但是在非线性问题的处理上,PSO 算法寻优过程十分复杂,同时存在非线性的情况,单纯依靠线性权重无法做到有效的调节,仍然难以避免陷入局部最优的情况,由此,在文献[8]的基础上加以改进,设计了一种自适应惯性权重,其组合表达式如式(20)~(22)所示:

$$\omega_t^i = k \frac{\omega_{\max} - \omega_{\min}}{\max\{\Delta x_i^t\}} \Delta x_i^t + \omega_{\min} \quad (20)$$

$$\Delta x_i = \sqrt{\sum_{d=1}^D (x_{id} - g_d)^2} \quad (21)$$

$$k = \frac{t_{\max} - t}{t_{\max}} \quad (22)$$

式中: ω_t^i 表示在第 t 次迭代中惯性权重的值,式(21)用来计算当前位置与群体最优位置间的距离,这种方法利用了粒子与群体最优位置之间的距离,对惯性权重进行动态调节,当粒子距离最优点较远时, Δx_i^t 数值较大,在不考虑 k 变化的情况下, $\frac{\Delta x_i^t}{\max\{\Delta x_i^t\}}$ 较大, ω_t^i 相对较大,较大的惯性权重保证了粒子的全局搜索能力,当 Δx_i^t 较小时,说明粒子相对靠近群体最优位置,与此对应, ω_t^i 会取一个相对较小值,这样较小的惯性权重保证了粒子的局部搜索能力。同时迭代系数 k 随着 t 的增长而不断减小,保证算法在迭代前期与迭代后期对于整体惯性权重逐渐趋于减小的趋势不变。通过引入自适应惯性权重的方法,在平衡局部搜索和全局搜索的过程中,适当降低了陷入局部最优的概率,提高了 PSO 算法的搜索准确性。

2.3 动态调整学习因子

学习因子 c_1, c_2 分别决定了粒子个体经验和群体经验对粒子运行轨迹的影响,通常情况下 $c_1=c_2$ 且为固定值,但是随着深入研究发现,学习因子的取值也可以类似惯性权重发生动态变化,根据 PSO 算法在迭代初期注重广泛搜索,迭代后期注重快速收敛的特性。在想要同时保证粒子多样性和收敛性的情况下,采用动态调整学习因子,根据文献[9]的分析, c_1, c_2 同时调整效果优于单项系数调整,在搜索的前阶段, c_1 取较大值, c_2 取较小值,侧重于增加粒子的全局搜索能力,后期则相反,侧重于增强粒子的局部搜索能力,将加速

系数表示为

$$\left. \begin{aligned} c_1(t) &= 1.2 + 1.1 \sin\left(1 - \frac{t}{T_{\max}}\right) \pi \\ c_2(t) &= 1.9 - 1.1 \sin\left(1 - \frac{t}{T_{\max}}\right) \pi \end{aligned} \right\} \quad (23)$$

式中: t 为当前迭代次数; T_{\max} 为最大迭代次数。

2.4 速度公式的改进

通常的速度更新公式, 每个粒子只通过自身最优位置和群体最优位置进行速度更新, 如果粒子本身选择位置不理想, 则会影响寻优结果, 根据社会学家 Wilson 的理论, 群体中的个体都能从其他个体的经验中受益, 其他粒子的最优位置信息同样可以起到帮助作用, 在文献[10]的基础上, 在原有速度公式上添加一项, 而 P_a 是通过轮盘赌选出的所有粒子的历史最优位置, 速度公式变更如下:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (g_{id}^k - x_{id}^k) + c_3 r_3 (P_a - x_{id}^k) \quad (24)$$

其中 $c_3 r_3$ 取 $c_1 r_1$ 和 $c_2 r_2$ 中较小值的 0.3 倍, 通过这种改动, 降低了因为选取到位置不好的粒子而影响寻优结果的概率。

3 算例仿真部分

3.1 适应度函数的确定

在列车节能控制问题中, 存在多个约束条件, 这里采用简单的惩罚函数机制, 将多个约束条件转化为惩罚因子, 从而将有约束问题转化为无约束问题, 根据提到的约束条件, 共有: 末速度为 0, 最大加速度限制, 运行时间限制, 最大速度限制以及运行距离限制。通过设置 4 个惩罚因子 $\alpha, \beta, \gamma, \lambda$, 将目标函数转换为

$$Q = \sum_{k=0}^{n-1} F_k * V_k * \Delta t + \alpha * v_n^2 + \beta * (s_n - S)^2 + \gamma * \sum_{k=0}^{n-1} [\max(0, v_k) - V_{\max}]^2 + \lambda * \sum_{k=0}^{n-1} [\max(0, a_k - a_{\max})]^2 \quad (25)$$

由于目标函数需要尽可能的取小, 所以适应度函数设置为目标函数的倒数:

$$J(X) = \frac{1}{Q} \quad (26)$$

3.2 算法流程

1) 初始化粒子的位置和速度, 根据适应度函数计算各个粒子的适应度, 同时获得粒子的个体最优位置和群体最优位置。

2) 根据式(21)计算粒子与群体最优粒子之间的距离, 同时根据式(20)更新下次迭代各粒子的惯性权重。

3) 按照新的速度更新公式(24)与动态学习因子公式(23)更新粒子的学习因子, 速度和位置, 再次计算适应度值。

4) 若当前粒子的适应度值比历史最优值好则取代历史最优值。

5) 判断是否满足迭代终止条件, 是则输出结果, 不是则返回第 2 步。

3.3 仿真分析

针对标准粒子群算法和改进粒子群算法进行对比, 选取广州地铁 8 号线中一部分区段进行仿真模拟。运行策略采用“牵引-巡航-惰行-制动”的方式。列车参数和线路参数如表(3)~(5)所示。

其中站间距离为 1 471 m, 列车的区间限速为 80 km/h, 列车最大加速度不得超过 0.8 m/s, 按照“牵引-巡航-惰行-制动”的策略运行, 粒子群规模取 $N=50$, 迭代次数为 200 次, 学习因子根据标准粒子群算法与改进粒子群算法的不同, 标准粒子群算法取固定值 $c_1=c_2=2$, 改进型粒子群算法采用动态学习因子, 同时设置 $\omega_{\max}=0.9, \omega_{\min}=0.4$, 标准粒子群算法采用线性惯性权重法, 而改进的粒子群算法采用自适应惯性权重法, 粒子最大速度不超过 0.1, 最小速度不小于 -0.1, 列车的牵引系数取 0.95, 制动系数取 0.8, 同时设置惩罚因子 $\alpha=$

1, $\beta=0.5$, $\gamma=0.05$, $\lambda=1$ 。多次运行后,分别得到位移能量对比图,位移速度对比图,适应度函数对比图如图(1)~(3)所示。

表 3 轨道列车参数
Tab.3 Parameters of rail transit trains

车型	列车编组	列车重量/t	车身长度/m	总阻力/(N/kN)
A 型	4 动 2 拖, 6 辆编组	335.4(AW2)	139.98	2.753 2+0.000 428 3

表 4 列车牵引特性和制动特性
Tab.4 Characteristics of traction and braking force of trains

V/(km/h)	0~37	43	50	55	60	65	70	75	80
F/kN	412	381	338	313	280	259	243	223	196
B/kN	381	381	381	381	381	366	342	313	300

表 5 区段线路参数
Tab.5 Parameters of rail transit section

行驶位置/m	0~150	450	750	1 050	1 200	1 350	1 471
线路坡度/‰	-3	8	-5	10	10	10.9	3
曲率半径/m	0	0	350	0	1 000	0	0

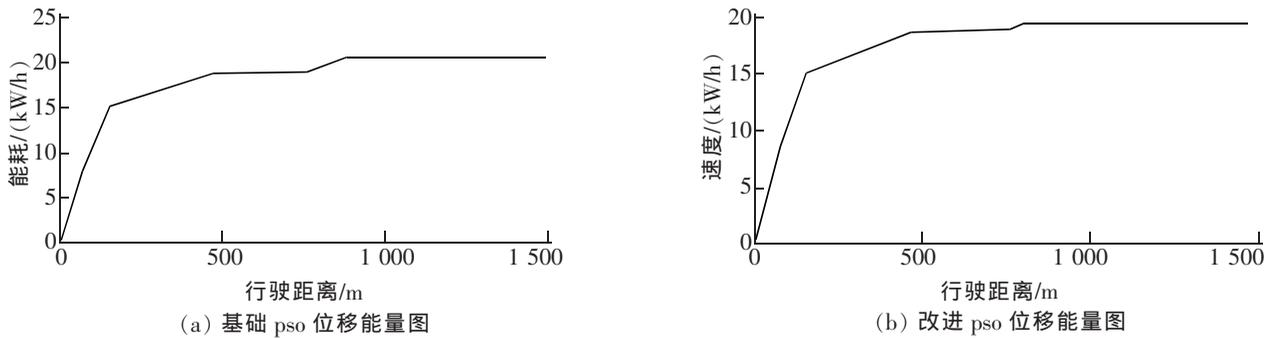


图 1 改进前后粒子群算法的位移-能量对比图
Fig.1 Comparison of the S-E curves between ordinary PSO and improved PSO

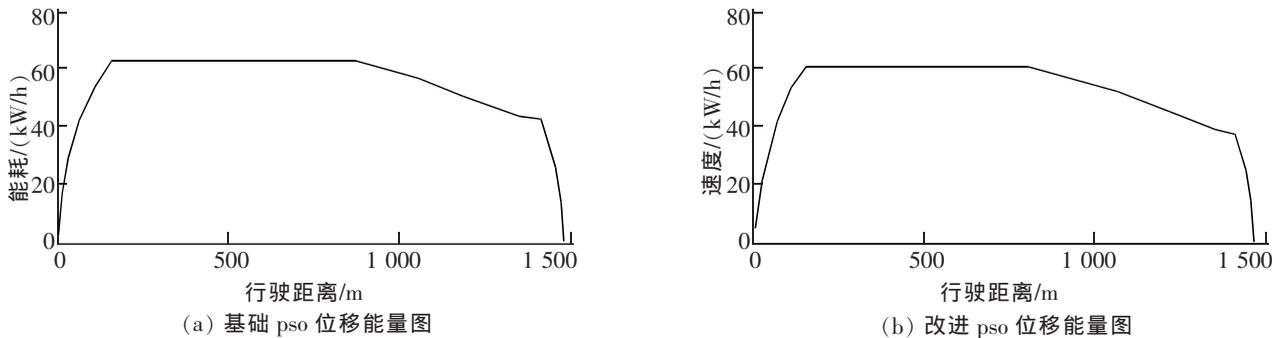


图 2 改进前后粒子群算法的位移-速度对比图
Fig.2 Comparison of the S-V curves between ordinary PSO and improved PSO

结合图 2,表(6)(7)可以发现,相较于标准型粒子群算法,改进型粒子群算法在略微增加牵引工况的情况下,有效地延长了惰行工况的运行时间。在到站时间固定不变的情况下,如图 1 所示,标准型粒子群算法优化的发动机能耗数值稳定在 20.525 kW/h 左右,而改进后的粒子群算法优化结果稳定在 19.388 kW/h 左右,相比标准粒子群算法效果提升了 5.54%左右,有效降低了能量损耗。在适应度和收敛性方面,标准型粒子群算法在 150 代左右达到收敛,而改进型粒子群算法在 120 代左右收敛就达到最优解,而适应度值也高于标准型粒子群算法,有效提升了运算的效率和精度。

表 6 2 种算法 4 种运行工况消耗时间

Tab.6 Consumption time of four operation conditions under two algorithms

方法	牵引	巡航	惰行	制动
标准粒子群算法	15	48	34	13
改进粒子群算法	16	34	48	12

表 7 2 种算法 4 种运行工况位移量

Tab.7 Displacement of four operating conditions under two algorithms

方法	牵引	巡航	惰行	制动
标准粒子群算法	121.09	790.57	481.2	79.73
改进粒子群算法	137.55	587.43	678.15	68.64

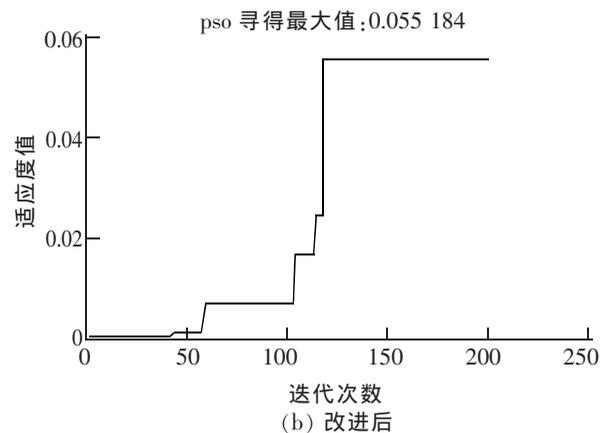
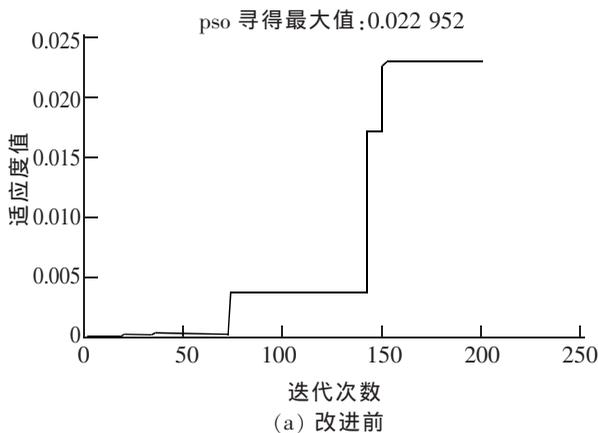


图 3 改进前后粒子群算法适应度函数对比图

Fig.3 Comparison of the fitness function between ordinary PSO and improved PSO

4 总结

针对列车运行节能控制优化这种非线性、多约束、大滞后的问题,提出了自适应惯性权重与动态学习因子相结合,同时对传统速度更新公式进行改进的方法,通过 Matlab 多次实验仿真,无论从优化效果还是收敛速度方面,改进后的粒子群算法相比于标准粒子群算法有了一定的提升,降低了列车运行过程中的能耗,给列车节能优化算法的设计提供参考。

参考文献:

- [1] 李玉生,侯忠生. 基于遗传算法的列车节能控制研究[J]. 系统仿真学报,2007,19(2):384-387.
- [2] 李焯,郭子立,郭奕杉. 基于 PSO 的轨道交通节能优化研究[J]. 控制工程,2018,25(10):1911-1915.
- [3] 李志勇,文睿,危初勇. 基于径向神经网络的机车牵引能耗计算模型[J]. 铁道学报,2011,33(9):27-30.
- [4] 王成莉. 基于遗传算法的高速列车节能运行控制研究[D]. 北京:北京交通大学,2016.
- [5] 杨辉,付雅婷. 列车运行建模与速度控制方法综述[J]. 华东交通大学学报,2018,35(5):1-8.
- [6] 张翔,徐中伟. 基于改进遗传算法的列车节能操纵方案研究[J]. 机电一体化,2017,23(1):27-30.
- [7] 樊莹莹. 一种列车运行控制优化方法研究[D]. 北京:北京交通大学,2014.
- [8] 杨轻,杨忠,许昌亮,等. 改进 PSO 算法及其无人机电力巡线规划应用[J]. 应用科技,2019,46(3):80-85.
- [9] 朱雅敏,薛鹏翔. 基于学习因子自适应改变的粒子群算法研究[J]. 陕西科技大学学报,2015,33(4):172-177.
- [10] 郭广寒,王志刚. 一种改进的粒子群算法[J]. 哈尔滨理工大学学报,2010,15(2):31-34.

Study on Energy-Saving Control of Railway Train Based on Improved PSO

Huang Jiangping, Cheng Shaorong

(School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: At present, there are a series of problems in the optimization algorithm of train energy-saving control, such as the unobvious effect of optimization and the slow convergence speed. A PSO algorithm with adaptive inertia weight and learning factor with adjustable ability and improved speed update formula is proposed. Firstly, inertia weight is used to balance the search ability in different stages, and dynamic learning factor with adjustable ability is added to enhance the operation efficiency and convergence ability of the algorithm in the later stage. Meanwhile, penalty function is introduced to transform the constraints in train operation into penalty factors, which can improve the search rate. Then, the traditional velocity updating formula is improved to reduce the probability of selecting unsatisfactory particles for the optimization results. After simulation and analysis by Matlab, compared with the traditional train operation optimization algorithm, the improved optimization algorithm converges faster and has better energy-saving effect.

Key words: train energy-saving control; particle swarm optimization (PSO); adaptive inertia weight; dynamic learning factor; improved speed update formula