

考虑多种训练方式的自适应最有价值球员算法

王 宁*, 刘 勇

(上海理工大学 管理学院, 上海 200093)

(* 通信作者电子邮箱 806929086@qq.com)

摘 要:最有价值球员算法(MVPA)是一种模拟体育比赛的新型智能优化算法,为解决其寻优精度低和收敛速度慢等问题,提出一种考虑多种训练方式的自适应最有价值球员算法(ACMTM-MVPA)。MVPA的初始化方式单一,随机性和盲目性强,降低了算法的收敛速度和寻求精度。为了增强初始化球员的水平,提高初始球队的整体实力,在MVPA的竞争阶段之前加入了训练阶段,并在训练阶段使用邻域搜索算法以及混沌序列和反向学习算法来训练和筛选球员;为了增强球员的自我探索能力以及向最佳球员学习的能力,使球员具有争夺最有价值球员奖杯的资格,在队伍竞争阶段加入了自适应的球员进化因子。对15个标准函数的测试结果表明,ACMTM-MVPA与MVPA、粒子群优化(PSO)算法和遗传算法(GA)相比,在寻优精度和收敛速度上更有优势。最后给出了ACMTM-MVPA在暴雨强度公式参数优化中的应用实例,结果显示,该算法明显优于自适应光学优化算法、传统回归法与优选回归法。

关键词:最有价值球员算法;训练阶段;自适应;球员进化因子;暴雨强度公式

中图分类号:TP301.6 **文献标志码:**A

Adaptive most valuable player algorithm considering multiple training methods

WANG Ning*, LIU Yong

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: The Most Valuable Player Algorithm (MVPA) is a new intelligent optimization algorithm that simulates sports competitions. It has the problems of low precision and slow convergence. An adaptive most valuable player algorithm considering multiple training methods (ACMTM-MVPA) was proposed to solve these problems. MVPA has a single initialization method, which is random and blind, reducing the convergence speed and accuracy of the algorithm. In order to enhance the level of the initial player and improve the overall strength of the initial team, the training phase was added before the competition phase of MVPA, and the neighborhood search algorithm and chaotic sequence and reverse learning algorithms were used to train and screen players; in order to enhance the player's ability to self-explore and learn from the best player to make the player have the qualification to compete for the most valuable player trophy, an adaptive player evolution factor was added during the team competition phase. Experimental results on 15 benchmark functions show that the proposed algorithm outperforms MVPA, Particle Swarm Optimization (PSO) algorithm and Genetic Algorithm (GA) in optimization accuracy and convergence speed. Finally, an application example of ACMTM-MVPA in parameter optimization of storm intensity formula was given. The results show that this proposed algorithm is superior to accelerated genetic algorithm, traditional regression method and preferred regression method.

Key words: Most Valuable Player Algorithm (MVPA); training phase; adaptive; player evolution factor; storm intensity formula

0 引言

智能优化算法是受人类智能、生物群体社会性或自然现象规律的启发,模仿其规律所提出来的求解问题的算法,为许多传统优化技术很难解决的组合优化问题提供了切实有效的解决办法^[1]。智能优化算法因其全局优化性能较好,可以进行并行计算和通用性强等特点,已广泛应用于系统控制、人工智能、生产调度、计算机工程等领域,展示出了其强大的优势,并且具有更好的应用前景。

最有价值球员算法(Most Valuable Player Algorithm, MVPA)是Boucekara^[2]于2017年提出的新型智能优化算法。该算法受到体育比赛的启发,球员们为了赢得联赛冠军而进行队伍竞争,并且他们单独进行竞争以赢得MVP奖杯。每个球员即代表一个潜在的解,通过竞争阶段来不断提高球员的水平,从而产生一个MVP,即对应问题的最优解。

MVPA具有结构简单、参数设置少、收敛速度较快等优点,已经应用到降低圆形天线阵列的最大旁瓣电平优化问

收稿日期:2019-10-26;修回日期:2019-12-20;录用日期:2019-12-23。

基金项目:教育部人文社会科学研究规划基金资助项目(16YJA630037);上海市“科技创新行动计划”软科学研究重点项目(17692109400, 18692110500);上海市社科规划课题(2019BGL014);上海理工大学人文社科“攀登计划”项目(SK17PB04)。

作者简介:王宁(1994—),男,安徽蚌埠人,硕士研究生,主要研究方向:智能优化、系统工程;刘勇(1982—),男,江苏金湖人,副教授,博士,主要研究方向:智能优化、系统工程。

题^[3]、压力容器设计问题^[4]等实际问题中;然而,MVPA在求解复杂优化问题时存在寻优精度低、收敛速度慢等缺陷。为了提高算法的优化性能,文献[4]将教与学优化算法引入MVPA中,算法设计了两种训练模式来分别提高球员的能力和队伍的凝聚力,提高了算法的收敛速度和搜索精度。

本文分析了MVPA寻优精度低的原因,并从球员初始化方式出发,提出了考虑多种训练方式的自适应最有价值球员算法(Adaptive Most Valuable Player Algorithm Considering Multiple Training Methods, ACMTM-MVPA)。球员除了通过个体竞争和队伍竞争提升技能之外,还考虑球员为了进入球队而付出的努力和训练对提高球员技能的重要性,最后使用一系列标准测试函数和求解暴雨强度公式参数优化问题对ACMTM-MVPA的有效性进行了验证。

1 最有价值球员算法

在MVPA中,将每个球员看作一个潜在的可行解,并将球员分配到不同的队伍中去,通过个体竞争和队伍竞争来使球员的能力不断提高,并最终产生一个MVP球员,即算法的最优解。算法可以分为个体竞争和队伍竞争两个阶段,个体竞争意味着球员之间的竞争,队伍竞争即队伍之间的比赛。球员数量即群体的大小,球员的各种能力就是球员的维度,适应度函数就是球员的水平,适应度越好,说明水平越高。

1.1 初始化

首先在搜索空间内随机生成 $PlayersSize$ 个球员,并随机分成 $TeamsSize$ 个队伍。队伍生成方法^[2]如下:

$$nP_1 = \text{ceil}\left(\frac{PlayersSize}{TeamsSize}\right) \quad (1)$$

$$nP_2 = nP_1 - 1 \quad (2)$$

$$nT_1 = PlayersSize - nP_2 \times TeamsSize \quad (3)$$

$$nT_2 = TeamsSize - nT_1 \quad (4)$$

其中:ceil为向上取整函数; nT_1 表示第一部分的队伍数量, nP_1 表示第一部分每个队伍的队员数量; nT_2 表示第二部分的队伍数量, nP_2 表示第二部分每个队伍的队员数量。

1.2 竞争阶段

在完成初始化之后,算法进入竞争阶段。在该阶段中,球员通过队内竞争来提高自身能力,同时球队为了提升其实力与其他队伍进行队间竞争。下文给出球员之间的个体竞争以及球队之间的队伍竞争的具体方法。

1.2.1 个体竞争

在个体竞争阶段,每个球员都致力于成为所在队伍中的最佳球员($FranchisePlayer$)和联盟最有价值球员(MVP)。为此,球员努力去提高自身能力,并与所在队伍中最佳球员和联盟最有价值球员作比较。因此,每个球员按照如式(5)^[2]更新:

$$TEAM_i = TEAM_i + rand \times (FranchisePlayer_i - TEAM_i) + 2 \times rand \times (MVP - TEAM_i) \quad (5)$$

其中: $TEAM_i$ 表示第*i*个队伍中队员; $rand$ 是 $[0,1]$ 区间服从均匀分布的常数; $FranchisePlayer_i$ 表示第*i*个队伍中的最佳球员; MVP 表示联盟最有价值球员。

1.2.2 队伍竞争

在球员完成个体竞争之后,球队之间进行队伍竞争。在这个阶段, $TEAM_i$ 和 $TEAM_j$ 两个队伍相互比赛。按以下机制

选取获胜队伍。

以最小化问题为例,首先按照式(6)标准化队伍的适应度^[2]。

$$fitnessN(TEAM_i) = fitness(TEAM_i) - \min(fitness(ALL Teams)) \quad (6)$$

其中: $fitnessN(TEAM_i)$ 是第*i*个队伍未标准化的适应度值; $\min(fitness(ALL Teams))$ 是全联盟中最小的适应度值。 $TEAM_i$ 打败 $TEAM_j$ 的概率^[2]如式(7)所示:

$$Pr\{TEAM_i \text{ beats } TEAM_j\} = 1 - \frac{(fitnessN(TEAM_i))^k}{(fitnessN(TEAM_i))^k + (fitnessN(TEAM_j))^k} \quad (7)$$

其中:文献[2]中的*k*取1; $fitnessN(TEAM_i)$ 为第*i*个队伍标准化后的适应度值; Pr 表示获胜概率。

根据式(6)~(7),获胜概率分为不同和相同两种情况。

如果两个队伍的获胜概率不同,则会生成一个 $[0,1]$ 区间的随机数,如果这个随机数大于较高的获胜概率,则获胜概率较高的队伍获胜;反之,则获胜概率较低的队伍获胜。

如果两个队伍的适应度相同,则会得到相同的获胜概率。因此,还需要再生成一个 $[0,1]$ 区间的随机数,若这个随机数高于0.5,第一个队伍获胜;否则,第二个队伍获胜。

当球队 $TEAM_i$ 获胜时,则 $TEAM_i$ 中队员按如式(8)^[2]更新:

$$TEAM_i = TEAM_i + rand \times (TEAM_i - FranchisePlayer_j) \quad (8)$$

其中: $TEAM_i$ 表示第*i*个队伍中队员; $rand$ 表示 $[0,1]$ 内均匀分布的随机数; $FranchisePlayer_j$ 表示第*j*个队伍中的最佳球员。

当球队 $TEAM_i$ 获胜时,该队伍中队员按如式(9)^[2]更新:

$$TEAM_i = TEAM_i + rand \times (FranchisePlayer_j - TEAM_i) \quad (9)$$

2 考虑多种训练方式的自适应MVPA

在现实的球赛中,每一个球队都想挑选实力强劲的球员,以此来增强球队的实力,从而提高球队在联赛中的地位。但是MVPA采用随机的方式初始化球员,并且球队选拔球员具有较强的随机性和盲目性,因此,本文考虑加入训练阶段。训练阶段的加入就是通过多种训练方式来增强球员的能力。在训练阶段中,本文加入了邻域搜索算法和混沌序列与反向学习算法。邻域搜索算法主要是让想将实力较弱的球员培养出实力强劲的球员,通过局部搜索的方式,让球员能快速靠近较优解;混沌序列和反向学习算法主要是淘汰实力弱的球员,筛选出实力强劲的球员。同时,可以提高球员的训练速度和效率,从而有助于提高算法的收敛速率,通过球队的整体平均水平和优秀球员的平均水平决定是选择邻域搜索算法还是混沌序列和反向学习算法或保持不变。

在现实的联赛中,比赛初期的球队之间的实力差距较大,球员水平也参差不齐,在球队比赛的过程中,即使是对方球队中的最佳球员,也不一定有很多值得学习的地方,但随着比赛的不断推进,实力弱的球队以及球员可能会被不断地淘汰或者是提高,到比赛后期球队和球员的实力都会越来越强,对于球员来说,赢的球队中的最佳球员一定有很多值得球员学习的地方,也能通过学习提高自身的能力和水平。因此,在比赛初期,球员更多地需要自我探索,而在比赛后期,双方球队和球员的实力都很强,球员都可以从对方最佳球员的身上学习到很多技巧和方法。基于此,本文考虑在队伍竞争阶段球队

输了比赛后的更新公式中加入自适应球员进化因子,在迭代前期,自适应球员进化因子较大,增强算法的全局搜索能力;在迭代后期,自适应球员进化因子较小,增强算法的局部寻优能力。

2.1 训练阶段

在MVPA中,球员在初始化后被随机分配至各个球队。而在实际的过程中,球员必须达到一定的水平,满足球队的要求才能进入球队。因此,球员要通过训练不断提高自身能力。借鉴篮球联赛中球员加入队伍之前的海选过程,为提高球员的质量和球队的整体水平,在MVPA的基础上,引入训练阶段来提高算法的寻优精度和收敛速度。

训练阶段加入在竞争阶段之前,这样球员提高自身能力除了通过个体竞争和队伍竞争之外,还有球员为了通过进入球队而进行的努力和训练。在训练阶段,球员为了进入理想的球队,会不断训练来提高自己的能力。在训练过程中,球员可能会出现以下三种情况:

1)当球员水平高于优秀球员的平均水平时,球员觉得自身能力很强,且自己的训练方法很有效,则会一直按照自己的方法进行训练。

2)当球员水平高于所有球员的平均水平但低于优秀球员的平均水平时,球员觉得自身能力还需要加强,且自己的训练方法有一定的效果,则会微调训练方法以获得更好的效果。

3)当球员水平不高于所有球员的整体平均水平时,球员觉得自身能力不行,且训练方法没有效果,则会很大程度地调整自己的训练方法,以获得意想不到的良好效果。

设定 f_i 表示当前球员的适应度值, $f_{\min}$ 表示当前所有球员的最小适应度值, f_{avg} 表示当前所有球员的平均适应度值,将适应度值优于 f_{avg} 的适应度值求平均得到 f_{avg}' 。

当 $f_i \leq f_{\text{avg}}'$,此时球员水平高于优秀球员的平均水平,球员的能力比较强,可能不会做出改变,因此,这里考虑球员保持不变,如式(10)所示:

$$Player_i^{\text{new}} = Player_i^{\text{old}} \quad (10)$$

当 $f_{\text{avg}}' < f_i < f_{\text{avg}}$,球员对自己的训练方法进行局部调整,局部搜索策略是以每个当前解 $Player_i$ 为中心,在其周围通过式(11)产生一个较优的解:

$$Player_i^{\text{new}} = Player_i^{\text{old}} + (Player_i^{\min} + Player_i^{\max}) \times (r_1 - r_2) \times \varphi \quad (11)$$

其中: r_1, r_2 为(0,1)内随机数; φ 为微调因子; $Player_i^{\min}$ 和 $Player_i^{\max}$ 分别为第 i 个球员的下界和上界。

当 $f_i \geq f_{\text{avg}}$ 时,球员可能会在很大程度上调整自己的训练方法,本文采取混沌序列和反向学习的方法来更新球员,先用Cat混沌序列产生一个解,然后再使用式(12)产生相对应的反向解。

$$Player_i^{\text{new}} = (Player_i^{\min} + Player_i^{\max}) - Player_i^{\text{old}} \quad (12)$$

混沌现象^[5]是在非线性动力系统中表现的确定性、类随机的过程,这种过程既非周期又不收敛,混沌理论看似混乱但其内在结构十分精致,具有随机性、规律性和遍历性等特点,已经逐渐作为智能优化算法的优化机制得到广泛应用。基于不同映射的混沌序列的发生器会产生具有不同概率密度分布的混沌序列,这种分布特性的差异会影响算法的效率。Cat映射具有良好的遍历均匀分布特性,当最优解均匀分布于自变量区间或靠近自变量区间的中间位置时,有助于提升搜索效

率。其动力学方程如下:

$$\begin{cases} x_{n+1} = (x_n + y_n) \bmod 1 \\ y_{n+1} = (x_n + 2y_n) \bmod 1 \end{cases} \quad (13)$$

反向学习策略(Opposition-Based Learning, OBL)是由Tizhoosh^[6]在2005年首次提出的一种新技术。文献[7]证明了反向学习算法的学习速度快,而且优化能力更强,文献[8]表明通过反向学习来进行种群的初始化,更有助于提高算法的收敛速率。其基本思想为设 $X = (x_1, x_2, \dots, x_D)$ 表示 D 维空间中的一个可行解,其对应的反向解 $X' = (x'_1, x'_2, \dots, x'_D)$ 可定义为:

$$x'_j = a_j + b_j - x_j; \quad x_j \in [a_j, b_j] \quad (14)$$

其中: a_j, b_j 分别为 x_j 的下、上边界。由于反向解有50%以上的可能性比原解更优,因此,若将反向解和原解进行合并,并选择其中较优的解,则获取最优解的概率将大大提高。

2.2 自适应球员进化因子

为了获得更好的效果,MVPA需要平衡好全局搜索和局部搜索能力:在算法初期,进化因子 μ 较大,让球员有更多的自我提升机会,从而扩大球员的探索区域;在算法后期,进化因子 μ 较小,让球员向优秀球员学习,使得球员能够达到最佳状态;同时,在算法初期,球员应该有较强的自我学习能力,不断去探索。随着算法的迭代,球员不能增强向优秀球员的学习能力,对应的局部开发能力逐渐增强。

根据以上准则,本文对式(8)进行了调整,添加了自适应的球员进化因子,如式(15)~(16)所示:

$$TEAM_i = \mu \times TEAM_i + \text{rand} \times (TEAM_i - \text{FranchisePlayer}_i) \quad (15)$$

$$\mu = 1 - \frac{1}{\exp\left[\left(1 - \frac{ITER}{MAXITER}\right)^2\right]} \quad (16)$$

式中:球员进化因子 $\mu \in [0, 1]$, $MAXITER$ 为最大迭代次数, $ITER$ 为当前迭代次数。由式(16)可知,在算法迭代初期, $\mu \approx 1$,此时式(15)中,球员的提升主要来源于对自身的探索开发,具有较强的全局搜索能力;随着算法的不断迭代, μ 不断减小,此时球员应向最佳球员学习,吸收最佳球员的优良增强自己,不断提升,此时,算法的局部开发能力增强,从而提高算法的收敛速度和寻优精度。

综上所述,ACMTM-MVPA的基本流程如下:

步骤1 初始化阶段,在搜索空间内,随机生成 $PlayersSize$ 个球员,并计算出所有球员的适应度值,以及所有球员的平均适应度值和高于平均值的球员的平均值。

步骤2 训练阶段,按照式(10)~(12)生成一个球员数量为 $PlayersSize$ 的群体。然后这些球员被随机分配组成队伍数量为 $TeamsSize$ 的球队。通过评估这些球员的能力,选出MVP(联盟最有价值球员)和每个球队 $FranchisePlayer$ (每个队伍中的最佳球员)。

步骤3 个体竞争阶段,在该阶段每个球队都要进行个体竞争并按照式(5)更新选定球队的每个球员对应的解。

步骤4 队伍竞争阶段,按照式(6)标准化球队的适应度。

步骤5 按照式(7)计算队伍的获胜概率。

步骤6 根据获胜机制按照式(15)或式(9)更新球队中队员的技能。

步骤7 检查是否满足算法停止准则,若满足,则输出当前最优解并结束算法;否则转至步骤2。

3 数值实验

3.1 参数设置

为验证新算法的性能,选取了15个测试函数作为实验对象,这些函数的函数名、表达式、维数、搜索范围和理论最优值如表1所示。为了检验新算法的合理性和有效性,将提出的ACMTM-MVPA与MVPA、粒子群优化(Particle Swarm Optimization, PSO)算法和遗传算法(Genetic Algorithm, GA)进行比较实验。

本文的实验环境为:计算机CPU为i5-3210M,内存为

4 GB,操作系统Windows 7,编程软件Matlab 2016b。ACMTM-MVPA的参数设置为:球员数量 $PlayersSize = 100$,队伍数量 $TeamsSize = 20$,最大迭代次数 $MAXITER = 300$, $\varphi = 0.001$,最大适应度评价次数为 3×10^4 。MVPA的参数设置为:球员数量 $PlayersSize = 100$,队伍数量 $TeamsSize = 20$,最大迭代次数 $MAXITER = 300$,最大适应度评价次数为 3×10^4 。PSO算法的参数设置为:群体规模 $N = 30$,最大迭代次数 $T = 300$,权重的上下界为 $w_{\max} = 0.9$, $w_{\min} = 0.4$,学习因子 $c_1 = c_2 = 2$,最大适应度评价次数为 3×10^4 。GA的参数设置为:种群规模 $PopSize = 100$,最大迭代次数 $MaxIteration = 300$,交叉概率 $P_c = 0.6$,变异概率 $P_m = 0.01$,最大适应度评价次数为 3×10^4 。

表1 测试函数
Tab. 1 Test functions

函数名	表达式	维数	搜索范围	最优解
f_1	$f_1(x) = \sum_{i=1}^D x_i^2$	500	$[-5, 12, 5, 12]$	0
f_2	$f_2(x) = \left(\sum_{i=1}^D x_i + \left(\sum_{i=1}^D x_i \right)^{1/n} \right)^2$	500	$[0, 10]$	0
f_3	$f_3(x) = [x_i^2 + 2x_{i+1}^2 - 0.3\cos(2\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7]$	500	$[-15, 15]$	0
f_4	$f_4(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \cos\left(\frac{x_i}{\sqrt{i}}\right)$	500	$[-600, 600]$	0
f_5	$f_5(x) = \sum_{i=1}^D [x_i^6 \times (\sin(1/x_i) + 2)]$	500	$[-1, 1]$	0
f_6	$f_6(x) = -20 \times e^{-0.2 \times \sqrt{\frac{\sum x_i^2}{D}}} - e^{\frac{\sum \cos(2\pi x_i)}{D}} + 20 + e$	500	$[-32, 32]$	0
f_7	$f_7(x) = \left[\frac{1}{D} \sum_{i=1}^D x_i - \left(\prod_{i=1}^D x_i \right)^{1/D} \right]^2$	500	$[-1, 1]$	0
f_8	$f_8(x) = \sum_{i=1}^D x_i - \left(\prod_{i=1}^D x_i \right)$	500	$[-10, 10]$	0
f_9	$f_9(x) = \sum_{i=1}^D (ix_i^2)$	500	$[-10, 10]$	0
f_{10}	$f_{10}(x) = \sum_{i=1}^D x_i^{i+1} $	500	$[-1, 1]$	0
f_{11}	$f_{11}(x) = \sum_{i=1}^D x_i / e^{\sum \sin(x_i^2)}$	500	$[-2\pi, 2\pi]$	0
f_{12}	$f_{12}(x) = \sum_{i=1}^D x_i^2 + \left(\frac{1}{2} \sum_{i=1}^D ix_i \right)^2 + \left(\frac{1}{2} \sum_{i=1}^D ix_i \right)^4$	500	$[-5, 10]$	0
f_{13}	$f_{13}(x) = 1 - e^{\left(-0.5 \sum_{i=1}^D (x_i)^2 \right)}$	500	$[-100, 100]$	0
f_{14}	$f_{14}(x) = \sum_{i=1}^{D/4} ((x_{4i-2} + 10x_{4i})^2 + 5(x_{4i-1} + x_{4i})^2 + (x_{4i-2} + x_{4i-1})^4 + 10(x_{4i-3} + x_{4i})^4)$	500	$[-100, 100]$	0
f_{15}	$f_{15}(x) = \sum_{i=1}^D \left((10^6)^{\frac{i-1}{n-1}} \times x_i^2 \right)$	500	$[-100, 100]$	0

3.2 算法的寻优精度及计算成本比较

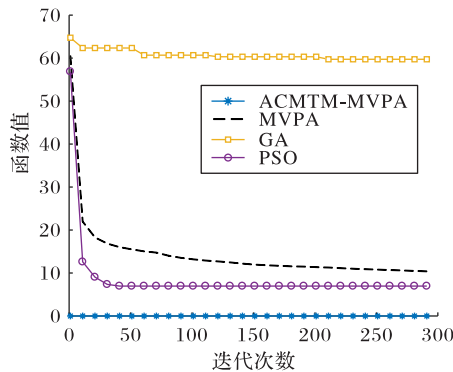
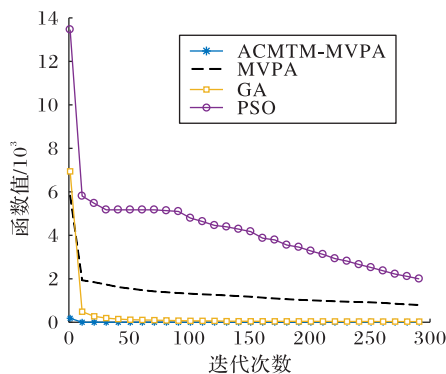
本节评估在相同的最大适应度评价次数下,算法的寻优精度以及平均运行时间。对于每个测试函数,ACMTM-MVPA、MVPA、PSO和GA均独立运行30次,结果如表2所示,其中:Best表示最优值,Mean表示均值,Worst表示最差值,Sd表示标准差,Meantime表示平均运行时间(单位:s)。

由表2可知,对于 $f_2(x)$ 、 $f_3(x)$ 、 $f_4(x)$ 、 $f_5(x)$ 、 $f_7(x)$ 、 $f_9(x)$ 、 $f_{10}(x)$ 、 $f_{13}(x)$,本文算法均达到了最优解,其他函数的优化结果也都要优于MVPA、PSO算法和GA算法,同时通过不同测试函数在测试结果的标准差可知,ACMTM-MVPA具有较好的鲁棒性。

在算法运行时间方面,ACMTM-MVPA在竞争阶段之前添

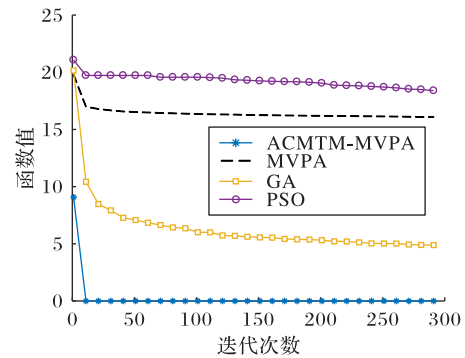
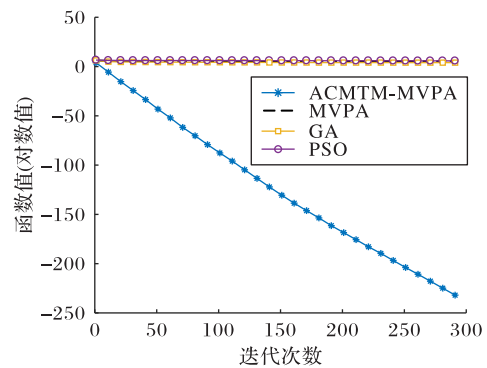
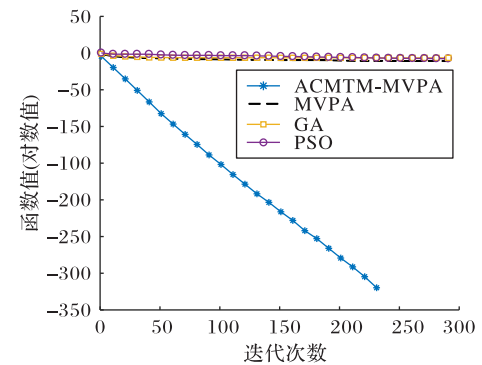
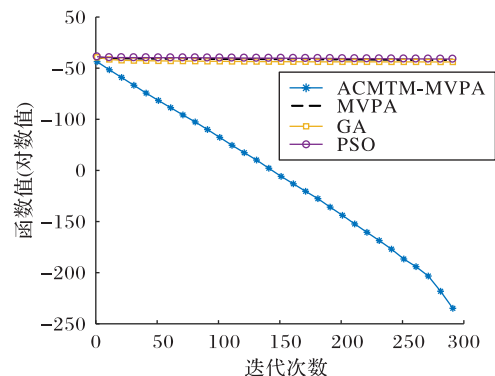
加了训练阶段,使得算法在前期可以搜索到更多的可行解,同时增加了算法的运行时间;自适应球员进化因子的添加,一方面能够让算法避免过早陷入局部最优解,另一方面,也会抑制算法的收敛速度,并增加了算法的运行时间。虽然训练阶段和自适应球员进化因子的添加增加了算法的运行时间,但极大地提高了算法的优化精度,而且运行时间的增加在可接受的范围内。

为了更直观地显示 ACMTM-MVPA 的优化精度和收敛速度,图1~6展示了函数 $f_2(x)$ 、 $f_4(x)$ 、 $f_6(x)$ 、 $f_9(x)$ 、 $f_{10}(x)$ 、 $f_{14}(x)$ 的收敛曲线(部分函数纵坐标的函数值取以10为底的对数)。从图中可以看出,与 MVPA、PSO 算法和 GA 算法相比,ACMTM-MVPA 不仅优化精度高而且收敛速度快。对于 ACMTM-MVPA 来说,6个测试函数都在50次迭代次数以内就收敛到一个较优解,由图1、图2、图5可知, $f_2(x)$ 、 $f_4(x)$ 、 $f_{10}(x)$ 均收敛到相应的理论最优值。可以得出,ACMTM-MVPA 对复杂函数的优化具有十分显著的效果。

图1 $f_2(x)$ 进化曲线Fig. 1 Evolution curve of $f_2(x)$ 图2 $f_4(x)$ 进化曲线Fig. 2 Evolution curve of $f_4(x)$

3.3 算法的收敛速度比较

为了测试 ACMTM-MVPA 的收敛速度,设定各个函数的优化精度为 $10E-15$,函数的维数为500,设定适应度函数最大评价次数为 3×10^4 ,独立运行20次,比较搜索到给定精度需要的最小评价次数 Mini_FEs、平均评价次数 Mean_FEs 以及达到优化精度的成功率 SR^[9],测试结果如表3所示。成功率指的是寻优达到给定精度的次数与总次数的比值,反映了算法的稳健性;函数平均评价次数反映算法的收敛速度和计算代价。

图3 $f_6(x)$ 进化曲线Fig. 3 Evolution curve of $f_6(x)$ 图4 $f_9(x)$ 进化曲线Fig. 4 Evolution curve of $f_9(x)$ 图5 $f_{10}(x)$ 进化曲线Fig. 5 Evolution curve of $f_{10}(x)$ 图6 $f_{14}(x)$ 进化曲线Fig. 6 Evolution curve of $f_{14}(x)$

由表3可知,ACMTM-MVPA 搜索到给定精度所需的评价次数的最小值、平均值在大多数函数测试中都是最小的,在同

样的精度条件下,评价次数越少,说明算法的搜索速度越快。尽管 ACMTM-MVPA 在处理函数 f_{11} 时的搜索速度跟其他三个算法一样,但是在其他 14 个函数上都优于 MVPA、PSO 算法和 GA 算法。此外,成功率为 0 表明,对于 20 次独立运行,在规定

的迭代次数下,算法不能达到给定的优化精度;成功率为 1 表明,对于所有 20 次独立运行,在规定的迭代次数下,算法都能达到给定的优化精度。表 3 结果表明,本文算法相较于其他算法具有更好的收敛速度和稳健性。

表 2 实验结果对比(500 维)

Tab. 2 Comparison of experimental results(500 dimensions)

函数	结果	ACMTM-MVPA	MVPA	PSO	GA
$f_1(x)$	Best	8.901E-244	80.278 3	277.156	4.931 17
	Mean	5.457E-241	137.658	372.977	5.772 39
	Worst	2.678E-240	228.047	447.668	6.613 39
	Sd	0	43.029 4	47.594 8	0.467 652
	Meantime/s	5.244 73	5.451 09	5.315 56	1.178 99
$f_2(x)$	Best	0	4.741 26	2.073 6	55.972 2
	Mean	0	6.110 19	6.158 04	57.712 8
	Worst	0	8.577 57	9	59.393 2
	Sd	0	0.900 431	2.865 31	1.033 26
	Meantime/s	5.494 55	5.213 52	3.403 82	1.466
$f_3(x)$	Best	0	2 299.84	7 128.59	499.513
	Mean	0	3 036.67	9 331.11	524.605
	Worst	0	4 485.56	12 028.1	559.478
	Sd	0	635.32	1 428.89	16.635 2
	Meantime/s	6.272 35	7.756 62	30.301 7	2.054 06
$f_4(x)$	Best	0	305.954	952.997	17.053 9
	Mean	0	481.93	1 271.98	20.380 9
	Worst	0	691.174	1 593.56	23.452 3
	Sd	0	107.21	178.573	1.489 69
	Meantime/s	6.981 38	6.646 25	4.216 73	1.827 36
$f_5(x)$	Best	0	0.032 46	0.564 57	1.075E-06
	Mean	0	0.091 69	1.315 08	1.833E-06
	Worst	0	0.165 10	4.276 91	2.883E-06
	Sd	0	0.034 58	0.950 16	4.611E-07
	Meantime/s	12.280 8	6.554 82	3.677 09	2.203 3
$f_6(x)$	Best	8.881E-16	14.686 1	14.574 9	4.305 42
	Mean	8.881E-16	15.612 6	15.967 8	4.425 12
	Worst	8.881E-16	16.574 6	17.782 9	4.572 75
	Sd	0	0.496 718	0.774 051	0.072 80
	Meantime/s	5.960 36	6.298 57	6.661 57	1.263 89
$f_7(x)$	Best	0	0.093 60	0.072 31	0.004 777
	Mean	0	1.059 31	0.118 09	0.006 923
	Worst	0	1.369 02	0.221 42	0.008 805
	Sd	0	0.413 18	0.042 87	0.001 120
	Meantime/s	5.168 55	4.709 84	4.869 66	0.758 329
$f_8(x)$	Best	0	0	1.054E-135	0
	Mean	0	0	1.203E-76	0
	Worst	0	0	2.406E-75	0
	Sd	0	0	5.380E-76	0
	Meantime/s	5.139 34	4.831 5	7.876 86	0.715 078
$f_9(x)$	Best	3.133E-244	34 309.3	290 540	4 526.23
	Mean	8.677E-239	64 552.5	379 469	5 313.81
	Worst	3.358E-238	106 521	565 145	5 849.19
	Sd	0	17 755.6	66 506.4	355.865
	Meantime/s	4.890 38	4.499 99	3.702 25	0.653 673
$f_{10}(x)$	Best	0	5.208E-18	5.054E-10	6.093E-09
	Mean	0	1.515E-11	4.884E-08	1.314E-07
	Worst	0	1.962E-10	2.265E-07	4.542E-07
	Sd	0	4.543E-11	6.804E-08	1.052E-07
	Meantime/s	11.037	7.160 95	3.092 4	3.155 88

续表					
函数	结果	ACMTM-MVPA	MVPA	PSO	GA
$f_{11}(x)$	Best	0	5.462E-136	2.923E-62	1.523E-103
	Mean	4.464E-216	1.718E-136	7.430E-46	9.567E-97
	Worst	4.465E-215	5.462E-136	7.187E-45	9.055E-96
	Sd	0	1.718E-136	2.265E-45	2.849E-96
	Meantime/s	6.48426	6.22983	3.54206	0.965432
$f_{12}(x)$	Best	0	3657.76	44866	9.638E+19
	Mean	1.849E-175	4420.91	109198	1.221E+20
	Worst	1.849E-174	4889.16	301547	1.499E+20
	Sd	0	340.303	75754.5	1.548E+19
	Meantime/s	4.96958	4.60198	3.58518	0.629629
$f_{13}(x)$	Best	0	1	1	1
	Mean	0	1	1	1
	Worst	0	1	1	1
	Sd	0	0	0	0
	Meantime/s	5.71041	4.73424	5.34137	0.617086
$f_{14}(x)$	Best	0	5.476E+07	1.43017E+11	862200
	Mean	7.704E-229	9.107E+07	1.49035E+11	1.146E+06
	Worst	4.033E-228	1.432E+08	1.59055E+11	1.454E+06
	Sd	0	2.785E+07	5.82392E+09	222340
	Meantime/s	9.81748	7.16818	8.33096	1.95029
$f_{15}(x)$	Best	0	3.393E+08	5.257E+09	1.076E+08
	Mean	1.373E-228	5.152E+08	7.099E+09	1.614E+08
	Worst	9.827E-228	6.766E+08	9.286E+09	1.985E+08
	Sd	0	1.132E+08	1.096E+09	2.588E+07
	Meantime/s	14.0602	7.194877	3.19886	3.24696

表 3 算法评价次数比较

Tab. 3 Comparison of algorithmic evaluation times

函数	精度	指标	算法				函数	精度	指标	算法			
			ACMTM-MVPA	MVPA	PSO	GA				ACMTM-MVPA	MVPA	PSO	GA
$f_1(x)$	1.00E-15	Mini_FEs	2 100	30 000	30 000	30 000	$f_9(x)$	1.00E-15	Mini_FEs	2 500	30 000	30 000	30 000
		Mean_FEs	2 215	30 000	30 000	30 000			Mean_FEs	2 595	30 000	30 000	30 000
		SR/%	100	0	0	0			SR/%	100	0	0	0
$f_2(x)$	1.00E-15	Mini_FEs	500	30 000	30 000	30 000	$f_{10}(x)$	1.00E-15	Mini_FEs	700	24 300	30 000	30 000
		Mean_FEs	500	30 000	30 000	30 000			Mean_FEs	830	28 860	30 000	30 000
		SR/%	100	0	0	0			SR/%	100	20	0	0
$f_3(x)$	1.00E-15	Mini_FEs	2 300	30 000	30 000	30 000	$f_{11}(x)$	1.00E-15	Mini_FEs	100	100	100	100
		Mean_FEs	2 475	30 000	30 000	30 000			Mean_FEs	100	100	100	100
		SR/%	100	0	0	0			SR/%	100	100	100	100
$f_4(x)$	1.00E-15	Mini_FEs	2 300	30 000	30 000	30 000	$f_{12}(x)$	1.00E-15	Mini_FEs	3 400	30 000	30 000	30 000
		Mean_FEs	2 425	30 000	30 000	30 000			Mean_FEs	3 755	30 000	30 000	30 000
		SR/%	100	0	0	0			SR/%	100	0	0	0
$f_5(x)$	1.00E-15	Mini_FEs	500	30 000	30 000	30 000	$f_{13}(x)$	1.00E-15	Mini_FEs	800	30 000	30 000	30 000
		Mean_FEs	670	30 000	30 000	30 000			Mean_FEs	1 085	30 000	30 000	30 000
		SR/%	100	0	0	0			SR/%	100	0	0	0
$f_6(x)$	1.00E-15	Mini_FEs	3 900	30 000	30 000	30 000	$f_{14}(x)$	1.00E-15	Mini_FEs	2 600	30 000	30 000	30 000
		Mean_FEs	4 100	30 000	30 000	30 000			Mean_FEs	2 715	30 000	30 000	30 000
		SR/%	100	0	0	0			SR/%	100	0	0	0
$f_7(x)$	1.00E-15	Mini_FEs	200	30 000	30 000	30 000	$f_{15}(x)$	1.00E-15	Mini_FEs	3 000	30 000	30 000	30 000
		Mean_FEs	210	30 000	30 000	30 000			Mean_FEs	3 175	30 000	30 000	30 000
		SR/%	100	0	0	0			SR/%	100	0	0	0
$f_8(x)$	1.00E-15	Mini_FEs	100	1 700	25 100	30 000							
		Mean_FEs	100	2 220	28 090	30 000							
		SR/%	100	100	70	0							

4 ACMTM-MVPA 在暴雨强度公式中的应用

随着经济社会的发展,中国步入城镇化快速发展的阶段,城镇化率已由 1978 年的 17.9% 增加到 2018 年的 59.58%。在全球气候变化与快速城镇化背景下,中国城市洪涝灾害日益严重。城市洪涝灾害已成为制约当地社会经济发展的突出问题^[10-11]。暴雨强度公式是确定城市防洪除涝或者排水工程设计标准的重要依据,其选择的合理性直接影响工程的规模和效益,也直接影响着工程投资建设进度^[12]。

根据我国现行的排水规范,一个重现期的暴雨强度公式^[12]形式为:

$$h=A/(t+B)^n$$

(17)

式中: h 为暴雨强度(mm/s); t 为降雨历时(min); A 、 B 、 n 为常数。这是常用的一种洪水灾害危险性分析模型,已经被广泛用于给水排水设计等洪水灾害管理中,由公式可知,该式为超定非线性方程,因此公式的参数优化问题实际上是一个非线性优化问题。然而,传统的优化方法不仅计算量大、通用性差,且所求结果也往往不太理想,本文提出将 ACMTM-MVPA 应用于暴雨强度公式参数优化中,以期获得良好的效果。

表 4 不同方法对各重现期暴雨强度公式拟合效果的比较

Tab. 5 Fitting effect comparison of storm intensity formula for each reproducing period by different methods

方法	参数最优值			最小残差平方和 Q	重现期/a	方法	参数最优值			最小残差平方和 Q	重现期/a
	A	B	n				A	B	n		
ACMTM-MVPA	12.721	6.399	0.528	0.039 80	100	ACMTM-MVPA	10.467	7.580	0.599	0.007 40	5
自适应光学优化算法	12.893	6.548	0.527	0.039 80	100	自适应光学优化算法	10.269	7.422	0.594	0.007 40	5
传统法	17.700	10.000	0.601	0.046 24	100	传统法	13.400	10.000	0.653	0.008 96	5
优选法	14.690	7.630	0.560	0.045 61	100	优选法	11.870	8.600	0.630	0.008 44	5
ACMTM-MVPA	12.332	6.724	0.543	0.025 30	50	ACMTM-MVPA	10.198	8.356	0.643	0.003 00	2
自适应光学优化算法	12.573	6.910	0.547	0.025 30	50	自适应光学优化算法	10.042	8.217	0.639	0.003 00	2
传统法	16.800	10.000	0.611	0.029 80	50	传统法	13.400	10.000	0.684	0.113 18	2
优选法	14.230	7.940	0.580	0.029 96	50	优选法	11.470	9.290	0.670	0.003 40	2
ACMTM-MVPA	11.452	6.924	0.559	0.017 70	20	ACMTM-MVPA	10.318	9.162	0.693	0.001 10	1
自适应光学优化算法	11.533	6.969	0.560	0.017 70	20	自适应光学优化算法	10.211	9.075	0.690	0.001 10	1
传统法	15.300	10.000	0.623	0.020 83	20	传统法	11.700	10.000	0.724	0.001 52	1
优选法	13.140	8.090	0.590	0.019 55	20	优选法	11.390	9.900	0.720	0.001 52	1
ACMTM-MVPA	11.382	7.724	0.584	0.012 80	10						
自适应光学优化算法	11.217	7.527	0.581	0.012 80	10						
传统法	14.400	10.000	0.636	0.031 66	10						
优选法	12.890	8.680	0.610	0.032 60	10						

5 结语

本文针对 MVPA 在求解复杂优化问题时寻优精度低、收敛速度慢的缺陷,提出一种考虑多种训练方式的自适应最有价值球员算法。在竞争阶段之前添加了训练阶段,来提高球员的能力和球队的整体实力,从而提高最优解的质量;在队伍竞争阶段,引入自适应球员进化因子,均衡了算法全局开发和局部搜索能力,从而有效地提高算法的寻优精度和收敛速度。选取了 15 个测试函数进行数值实验,结果表明,与 MVPA、PSO 算法和 GA 相比,本文提出的 ACMTM-MVPA 在求解复杂优化问题时不仅收敛速度快,而且优化精度高。同时,将算法应用于暴雨强度公式参数优化中,并与传统回归法、优选回归

根据文献[13]附表中不同重现期暴雨强度与降雨历时的数据。用 ACMTM-MVPA 来优化式(18)中的参数 A 、 B 、 n 使下式优化准则函数极小化^[13]:

$$\min Q=\sum_{z=1}^m\left(\frac{A}{\left(t_z+B\right)^n}-i_z\right)^2$$

(18)

其中:参数 A 、 B 、 n 的变化范围分别是 $[0,30]$ 、 $[0,100]$ 、 $[0,2]$,算法的迭代次数 $T=1\,000$ 。为了方便比较,同时给出了文献[14]中提供的使用传统回归法(简称传统法)、优选回归法(简称优选法)以及文献[15]中自适应光学优化算法所得的拟合结果,表 4 给出了不同方法对各重现期暴雨强度公式拟合效果的比较。由表 4 可知,在相同的数据和标准情况下,传统法和优选法的优化结果明显弱于 ACMTM-MVPA 和自适应光学优化算法,虽然自适应光学优化算法和 ACMTM-MVPA 的优化结果相近,但由文献[15]可知,自适应光学优化算法的迭代次数为 5 000,本文中 ACMTM-MVPA 的迭代次数只有 1 000 次,由此可知,ACMTM-MVPA 具有较快的优化速率和较高的拟合精度。同时,ACMTM-MVPA 具有参数设置简单、搜索精度高以及不易陷入局部最优解的特点,可尝试广泛应用于不同的复杂优化模型。

法和自适应光学优化算法进行比较,结果表明,ACMTM-MVPA 对暴雨强度公式参数优化中取得了最好的拟合效果,说明该算法的应用是成功的,可以考虑推广至其他复杂优化问题的应用。

参考文献 (References)

[1]

于坤杰,王昕,王振雷.基于反馈的精英教学优化算法[J].自动化学报,2014,40(9):1976-1983.(YU K J, WANG X, WANG Z L. Elitist teaching-learning-based optimization algorithm based on feedback [J]. Acta Automatica Sinica, 2014,40(9):1976-1983.)

[2]

BOUCHEKARA H R E H. Most valuable player algorithm: a novel optimization algorithm inspired from sport [J]. Operational Research, 2017, 80:1-57.

- [3] BOUCHEKARA H R E-H, ORLANDI A, AL-QDAH M, et al. Most valuable player algorithm for circular antenna arrays optimization to maximum sidelobe levels reduction [J]. IEEE Transactions on Electromagnetic Compatibility, 2018, 60(6): 1655-1661.
- [4] LIU X, LUO Q, WANG D, et al. An improved most valuable player algorithm with twice training mechanism [C]// Proceedings of the 2005 International Conference on Intelligent Computing, LNCS 10954. Cham: Springer, 2018: 854-865.
- [5] 徐辰华, 李成县, 喻昕, 等. 基于 Cat 混沌与高斯变异的改进灰狼优化算法[J]. 计算机工程与应用, 2017, 53(4): 1-9, 50. (XU C H, LI C X, YU X, et al. Improved grey wolf optimization algorithm based on chaotic Cat mapping and Gaussian mutation [J]. Computer Engineering and Applications, 2017, 53(4): 1-9, 50.)
- [6] TIZHOOSH H. Opposition-based learning: a new scheme for machine intelligence [C]// Proceedings of the 2005 International Conference on Computational Intelligence for Modeling Control and Automation. Piscataway: IEEE, 2005: 695-701.
- [7] 张斌, 李延晖, 郭昊. 基于反向学习的跨种群差分进化算法[J]. 计算机应用, 2017, 37(4): 1093-1099. (ZHANG B, LI Y H, GUO H. Cross-population differential evolution algorithm based on opposition-based learning [J]. Journal of Computer Applications. 2017, 37(4): 1093-1099.)
- [8] 王燕. 反向粒子群算法理论及其应用研究[D]. 西安: 西安工程大学, 2011: 11-12. (WANG Y. The research of opposition-based particle swarm optimization and its application [D]. Xi'an: Xi'an Polytechnic University, 2011: 11-12.)
- [9] 王皓, 欧阳海滨, 高立群. 一种改进的全局粒子群优化算法[J]. 控制与决策, 2016, 31(7): 1161-1168. (WANG H, OUYANG H B, GAO L Q. An improved global particle swarm optimization [J]. Control and Decision, 2016, 31(7): 1161-1168.)
- [10] 张建云, 王银堂, 贺瑞敏, 等. 中国城市洪涝问题及成因分析[J]. 水科学进展, 2016, 27(4): 485-491. (ZHANG J Y, WANG Y T, HE R M, et al. Discussion on the urban flood and waterlogging and causes analysis in China [J]. Advances in Water Science, 2016, 27(4): 485-491.)
- [11] 熊立华, 闫磊, 李凌琪, 等. 变化环境对城市暴雨及排水系统影响研究进展[J]. 水科学进展, 2017, 28(6): 930-942. (XIONG L H, YAN L, LI L Q, et al. Advances in analysis of impacts of changing environments on extreme urban rainfall and drainage infrastructure [J]. Advances in Water Science, 2017, 28(6): 930-942.)
- [12] 张子贤, 孙光东, 孙建印, 等. 城市暴雨强度公式拟合方法研究[J]. 水利学报, 2013, 44(11): 1263-1271. (ZHANG Z X, SUN G D, SUN J Y, et al. Study on fitting methods for urban storm intensity formula [J]. Journal of Hydraulic Engineering, 2013, 44(11): 1263-1271.)
- [13] 邹长武, 熊建秋, 李祚泳. 改进的蚂蚁算法及其在暴雨强度公式参数优化中的应用[J]. 四川大学学报, 2005, 37(5): 9-13. (ZOU C W, XIONG J Q, LI Z Y. Improved ant colony algorithm and its application to parameters optimization in storm intensity formula [J]. Journal of Sichuan University, 2005, 37(5): 9-13.)
- [14] 杨开, 程晓如. 暴雨强度公式中系数 B 统计算法一例[J]. 人民长江, 1996, 27(3): 16-22. (YANG K, CHENG X R. An example of coefficient B statistical algorithm in storm intensity formula [J]. Yangtze River, 1996, 27(3): 16-22.)
- [15] 王金叶. 改进光学优化算法及其应用研究[D]. 上海: 上海理工大学, 2017: 40-46. (WANG J Y. The study of optics optimization algorithm and its applications [D]. Shanghai: University of Shanghai for Science and Technology, 2017: 40-46.)

This work is partially supported by the Ministry of Education Human and Social Science Research Planning Fund Project (16YJA630037), the Key Project of Soft Science Research in Shanghai Science and Technology Innovation Action Plan (17692109400, 18692110500), the Shanghai Social Science Planning Project (2019BGL014), the Shanghai University of Technology Humanities and Social Sciences "Climbing Plan" Project (SK17PB04).

WANG Ning, born in 1994, M. S. candidate. His research interests include intelligent optimization, system engineering.

LIU Yong, born in 1982, Ph. D., associate professor. His research interests include intelligent optimization, system engineering.