

# 改进能量谷算法及其在模糊投资组合问题中的应用

李成希 孙荣

重庆工商大学 数学与统计学院，重庆，400067；

**摘要：**能量谷算法（EVO）作为新型的智能优化算法，其拥有独特的自适应分类进化机制以及多种进化策略，使其具有优良的性能，但在实际操作中存在局部搜索能力较差，算法复杂度较高等不足，基于此，该文提出了改进能量谷算法（IMEVO）。改进方法为：（1）使用类似粒子群算法的更新公式对于满足  $(NEL < EB)$  的粒子的更新公式进行替换；（2）使用较优粒子替换最近邻粒子；（3）通过融合更新策略的方式优化了满足  $(NEL > EB)$  的粒子的更新公式。随后与能量谷算法和粒子群算法进行对比测试，结果表明，改进能量谷算法有更优良的性能表现，验证了文章提出的改进方法具有有效性和实用性。

**关键词：**能量谷算法；粒子群算法；算法改进；模糊投资组合

**DOI：**10.69979/3029-2700.25.06.041

## 引言

现实情况的各类优化问题在多种约束条件的限制下，通常是非凸且难以求解的。如何快速准确的求解这类问题，是学者们的研究热点之一。在这类问题上，元启发式算法的应用较多。其中由 Holland (1992)<sup>[1]</sup>根据自然界“物竞天择，适者生存”的规律为灵感，提出的遗传算法和由 KennedyandEberhart (1995)<sup>[2]</sup>根据鸟群搜寻食物过程的启发而提出的粒子群算法最为经典，后续学者们以此为借鉴，根据自然规律或各类行为特征提出了不同的优化算法。例如，以自然界蚂蚁搜寻食物过程为基础的蚁群算法（2022）<sup>[3]</sup>，以蝴蝶的自然觅食和交配行为为基础的蝴蝶优化算法（2023）<sup>[4]</sup>，以海鸥的迁移和攻击行为为基础的海鸥优化算法（2022）<sup>[5]</sup>，以星雀搜寻食物的过程为基础的星雀优化算法（2023）<sup>[6]</sup>，以万有引力定律为基础的引力搜索算法（2023）<sup>[7]</sup>等。而能量谷算法（EVO）是 MahdiAzi 等人（2023）<sup>[8]</sup>以宇宙空间中的粒子衰变过程为灵感，根据衰变中的物理过程所设计的新型优化算法，目前该算法的研究和应用案例较少。

由于实际问题的复杂性以及优化问题本身可能存在缺陷，算法在处理实际问题时往往会出现很多问题，这就需要对原优化算法进行调整和改进，以提升算法在处理实际问题的能力。例如，谢七月, 刘广帅等（2024）<sup>[9]</sup>将灰狼优化算法与记忆、局部搜索和进化算子相结合，增强了算法的全局性和搜索精度。王海群, 郭庆通等（2024）<sup>[10]</sup>将黄金正弦算法和动态权重引入蜣螂优化算法，

提高了算法的收敛速度。孙正阳, 杜晔（2024）<sup>[11]</sup>在萤火虫相互吸引后增加交配阶段，融合遗传算法的思想不断产生新解，提高算法脱离局部最优的能力。盛斌, 张军（2024）<sup>[12]</sup>使用鱼鹰优化算法在第一阶段的全局勘探策略替换原始麻雀算法的探索者位置更新公式，提高了算法的性能。

基于上述研究，本文通过借鉴粒子全算法的优良特性以及数学优化的思想理论，优化能量谷算法的更新流程，用以解决能量谷算法在实际测试中存在的局部搜索不足，算法复杂度较高等问题，并通过函数测试和实证研究中验证有效性和实用性。

## 1 能量谷算法

能量谷算法的灵感来自于宇宙空间中微观粒子的衰变过程。在宇宙空间中，绝大多数粒子因其具有较多的能量而处于不稳定状态，不稳定粒子倾向于通过衰变释放能量，使得粒子处于具有较低能量的稳定态。衰变过程有三类常出现的表现形态，即  $\alpha$  衰变， $\beta$  衰变以及  $\gamma$  衰变。粒子衰变过程受到诸多因素影响，通常需要考虑中子富集水平，富集界水平，粒子稳定性水平等。依据上述因素，算法拥有多种更新进化策略，以更有效的搜索到最优值。

## 2 改进能量谷算法（IMEVO）

### 2.1 算法改进

尽管能量谷算法拥有多种进化策略，但依然存在着一些问题，主要体现在三个方面：

(1) 局部搜索能力较差。满足 ( $NEL < EB$ ) 条件的粒子全部依靠随机数  $r$  进行更新，无法很好的进行局部搜索。若种群数量较低或迭代次数较短，在面对复杂问题时，收敛精度较差。

(2) 最近邻粒子的缺陷。 $X_{NG}$  需要计算各个粒子之间的欧式距离进行确定，在种群数量较多，问题维度较高的情况下，对算法的运行速度影响较大且并未产生更好的性能表现。

(3) 算法复杂度较高。满足 ( $NEL > EB$ ) 的粒子通过再分类后分别具有两种更新方式，需要分别生成两类新粒子，再通过比较判断，选择更好的粒子进入下一次迭代过程，使得算法的复杂度较高，运算时间较长。

为了克服上述提出的三点劣势，通过借鉴其他智能优化算法的优点，对 EVO 算法的流程和更新公式进行优化，具体优化方式如下。

(1) 提高局部搜索能力。粒子群算法具有优良的局部搜索能力，本文借鉴其的更新方式优化满足 ( $NEL < EB$ ) 条件的粒子，具体公式如下。

$$\begin{aligned} v_i^{new} &= wv_i^j + a_1 r_1 (gbest_i^j - x_i^j) + a_2 r_2 (cp_i^j - x_i^j) \\ X_i^{new} &= X_i^j + v_i^{new} \end{aligned}$$

(2) 使用较优粒子  $X_{BET}$  替换最近邻粒子。较优粒子由满足 ( $NEL < EB$ ) 条件的粒子群中随机抽取确定，通过随机抽取较优粒子，在降低算法复杂度的同时，利用到较优粒子中的信息。

(3) 优化算法流程。

针对满足 ( $SL > EB$ ) 的粒子的改进方式。更新思路为，首先使用全局最优粒子中的部分解替换原粒子的部分解，生成过渡粒子  $X_i^1$ ，然后在其基础上，使用较优粒子  $X_{BET}$  的部分解对其进行替换，最后生成新粒子  $X_i^{new}$ 。具体更新公式如下。

$$\begin{aligned} X_i^1 &= X_i(X_{BS}(x_i^j)) \\ X_i^{new} &= X_i^1(X_{BET}(x_i^j)) \end{aligned}$$

针对满足 ( $SL < EB$ ) 的粒子的改进方式。借鉴粒子群算法的思路，在原算法的基础上，将两类更新公式融合为一个，具体公式如下，

$$X_i^{new} = X_i^j + a_3(r_1 X_{BS}^j - r_2 X_{BET}^j) + a_4(r_1 X_{BS}^j - r_2 X_{CP}^j)$$

随后，新生成的粒子需要计算 NEL 后与原粒子比较，选择更好的粒子进入下一代，综上所述，改进能量谷算法的流程如图 1 所示。

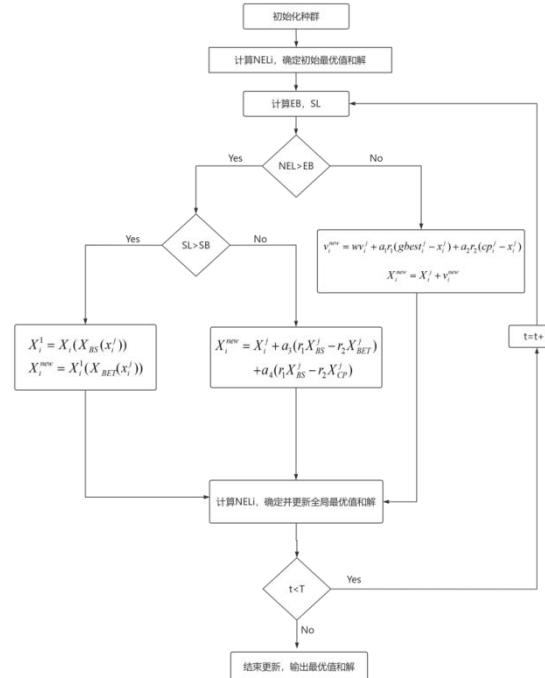


图 1 改进能量谷算法流程图

## 2.2 仿真测试

为了测试本文提出的改进能量谷算法 (IMEVO) 的性能，本文使用 6 个不同类型的测试函数进行优化测试，同时加入粒子群算法 (PSO)，能量谷算法 (EVO) 进行对比测试，使用 python 编写算法及运算代码，具体测试函数如表 1 所示。

表 1 测试函数表

函数形式	维度	取值范围	最优值
$f_1 = \sum_{i=1}^n x_i^2$	30	(-100, 100)	0
$f_2 = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	(-10, 10)	0

$f_3 = \sum_{i=1}^n ( x_i + 0.5 )^2$	30	(-30, 30)	0
$f_4 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	(-30, 30)	0
$f_5 = \sum_{i=1}^n [x_i - 10 \cos(3\pi x_i) + 10]$	30	(-5.12, 5.12)	0
$f_6 = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	(-32, 32)	0

由于所使用的算法使用随机搜索策略,为了避免单次运算的随机性,对于6个测试函数,分别使用三个算法对其独立运行30次后取平均值,用以评判算法性能,各算法初始设置参数如表2所示。

表2 算法参数设置表

ENEVO	EVO	PSO
$popsize = 100$ $n = 30$ $T = 200$ $w = 0.2$ $a_1 = a_2 = a_3 = a_4 = 1.5$	$popsize = 100$ $n = 30$ $T = 200$	$popsize = 100$ $n = 30$ $T = 200$ $w = 0.5$ $c_1 = 1$ $c_2 = 1.5$

其中,  $popsize$  为种群个数,  $n$  为函数维度,  $T$  为最大迭代次数,  $w$  为惯性权重,  $a_1$ ,  $a_2$ ,  $a_3$ ,  $a_4$  为调节因子, 通常设置为 1.5, 可根据实际情况调节,  $c_1$  为个体学习因子,  $c_2$  为社会学习因子。为了能直观观察到三类算法的对比情况,仅展示特定范围内的迭代流程,

表3 仿真测试对比表

$f_1$ 测试函数			
	IMEVO	EVO	PSO
最优值	4.03e-33	9.69e-21	5.31e-12
最劣值	4.19e-31	2.17e-15	4.77e-09
标准差	1.17e-31	6.50e-16	1.63e-09

$f_2$ 测试函数			
	IMEVO	EVO	PSO
最优值	1.29e-19	4.40e-07	0.0013
最劣值	2.10e-18	15.7757	1.6239
标准差	5.74e-19	4.6904	0.4467

$f_3$ 测试函数			
	IMEVO	EVO	PSO
最优值	1.39e-07	0.5730	2.90e-13
最劣值	6.48e-05	1.5851	3.65e-10

六个函数的具体测试结果如图2和表3所示,本文的所有运算结果仅保留四位小数。

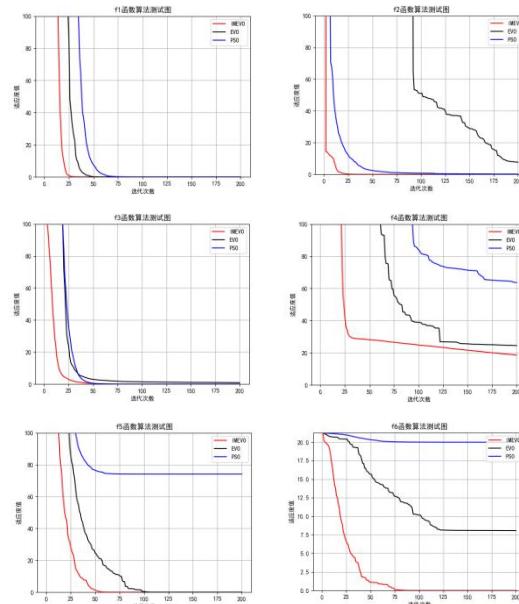


图2 仿真测试对比图

标准差	1.88e-05	0.3252	1.08e-10
$f_4$ 测试函数			
最优值	10.2810	16.1994	17.2682
最劣值	23.5064	27.5570	171.7861
标准差	3.8980	3.4668	43.5722
$f_5$ 测试函数 (当结果小于 1e-16 时, 显示为 0)			
最优值	0	0	52.7327
最劣值	0	3.38e-10	98.5006
标准差	0	1.01e-10	14.6106
$f_6$ 测试函数			
最优值	4.44e-16	5.15e-12	19.9999
最劣值	1.94e-12	20.2202	20
标准差	5.80e-13	9.8867	3.46e-07

根据图 2 和表 3 可知, 在相同种群规模和迭代次数的情况下, 本文所提出的 IMEVO 算法都有着更好的表现。在收敛速度上, IMEVO 算法相较于传统 EVO 算法和 PSO 算法有着更快的收敛速度, 能在较少的迭代次数中逼近最优解; 在搜索精度上, IMEVO 算法相较于传统 EVO 算法, 在单峰测试函数上搜索精度平均提高了一倍, 多峰测试函数上能够有效跳出局部最优解, 搜索到全局最优解, 仅在测试函数  $f_4$  上, 稍逊于 PSO 算法; 在算法稳定性方面, 对比三类算法在各类测试中的标准差指标, IMPSO 算法显著低于其他两类算法, 在搜索稳定性上有着更优的表现。

### 3 模糊投资组合实证分析

#### 3.1 模糊数理论

本文使用模糊数来表示投资活动过程中的收益和风险., 模糊数的相关理论如下。

定义 4.1 对于一般形式的梯形模糊数  $A = (a, b, \alpha, \beta)$ , 其隶属度函数  $\mu_A(x)$  具有以下形式:

$$\mu_A(x) = \begin{cases} \frac{x-(a-\alpha)}{\alpha}, & a-\alpha < x < a \\ 1, & a \leq x \leq b \\ \frac{(b+\beta)-x}{\beta}, & b < x < b+\beta \\ 0, & \text{其他} \end{cases}$$

定义 4.2 设  $A$  为模糊数,  $A$  的  $\lambda$  截集为  $A_\lambda = [a(\lambda), b(\lambda)]$   $A_\lambda = [a(\lambda), b(\lambda)]$ , 则  $A$  的可能性均值为:

$$M(A) = \int_0^1 \lambda [a(\lambda) + b(\lambda)] d\lambda = \frac{a+b}{2} + \frac{\beta-\alpha}{6}$$

定义 4.3 模糊数  $A$  的可能性方差为:

$$Var(A) = \int_0^1 \lambda [(a(\lambda) - M(A))^2 + (b(\lambda) - M(A))^2] d\lambda = \left( \frac{b-a}{2} + \frac{\alpha+\beta}{6} \right)^2 + \frac{\alpha^2 + \beta^2}{36}$$

定义 4.4 模糊数  $A$  的下半可能性方差为:

$$Var^-(A) = 2 \int_0^1 \lambda [b(\lambda) - M(A)]^2 d\lambda = \left( \frac{b-a}{2} + \frac{\alpha+\beta}{6} \right)^2 + \frac{\alpha^2}{18}$$

#### 3.2 问题描述及符号说明

假设投资者决定选择  $n$  钟风险资产和一种无风险资产, 分为  $T$  期进行进行投资, 目的是在一定的风险水平下, 使得投资者在经历  $T$  期投资后的累计财富最大化。模型中所涉及的变量符号用以下方式表示:

$$r_{it} = (a_{it}, b_{it}, \alpha_{it}, \beta_{it}) \quad i=1,2,3,\dots,n \quad t=1,2,3,\dots,T,$$

第  $i$  个资产在第  $t$  期的模糊收益率

$r_f$ : 无风险资产收益率

$R_t$ : 第  $t$  期的模糊组合收益率

$x_{it}$ : 在第  $t$  期时投资于第  $i$  个资产上的投资比例

$W_t$ : 投资着在第  $t$  期的累计财富

$Var_t$ : 第  $t$  期的模糊组合风险

$d_t$ : 在第  $t$  期投资着能接受的风险上界

$c_{it}$ : 在第  $t$  期时第  $i$  类资产投资比例变动所需要付出的交易费用率

$u_t$ : 第  $t$  期在单个资产上的投资比例上界

$l_t$ : 第  $t$  期在单个资产上的投资比例下界

#### 3.3 模糊投资组合模型

本文假定资产收益率是符合正太分布的相互独立的模糊变量, 用梯形模糊数来表示证券的预期收益率, 根据上述定理,  $n$  种资产组合也是一个模糊变量, 可以推导出第  $t$  期的组合模糊收益率为:

$$R_t = \sum_{i=1}^n x_{it} \left( \frac{a_{it} + b_{it}}{2} + \frac{\beta_{it} - \alpha_{it}}{6} \right) + (1 - \sum_{i=1}^n x_{it}) r_f - \sum_{i=1}^n c_{it} |x_{it} - x_{it-1}|$$

投资着在投资完成后的累计财富为：

$$W_T = W_1 \prod_{t=1}^T (1 + R_t) = W_1 \prod_{t=1}^T \left( 1 + \sum_{i=1}^n x_{it} \left( \frac{a_{it} + b_{it}}{2} + \frac{\beta_{it} - \alpha_{it}}{6} \right) + (1 - \sum_{i=1}^n x_{it}) r_f - \sum_{i=1}^n c_{it} |x_{it} - x_{it-1}| \right)$$

第  $t$  期的模糊组合下半方差为：

$$Var_t^- = \sum_{i=1}^n x_{it}^2 Var_t^-(r_{it}) = \sum_{i=1}^n x_{it}^2 \left( \frac{b_{it} - a_{it}}{2} + \frac{\alpha_{it} + \beta_{it}}{6} \right)^2 + \frac{\alpha_{it}^2}{18}$$

同时，在投资过程中还需要满足以下约束条件：

每一个投资周期中，投资组合的风险需要在投资者的可接受范围之内，即：

$$Var_t^- \leq d_t$$

在禁止买空和卖空的情况下，投资比例的总和应为 1，即：

$$\sum_{i=1}^n x_i = 1 \text{ 且 } x \geq 0$$

投入单一资产的比例要满足比例上下界要求，即：

$$l_t \leq x_{it} \leq u_t$$

综上所述，本文所构建的多阶段模糊均值一下半方差模型为：

$$\begin{aligned} Max W_T &= W_1 \prod_{t=1}^T \left( 1 + \sum_{i=1}^n x_{it} \left( \frac{a_{it} + b_{it}}{2} + \frac{\beta_{it} - \alpha_{it}}{6} \right) + (1 - \sum_{i=1}^n x_{it}) r_f - \sum_{i=1}^n c_{it} |x_{it} - x_{it-1}| \right) \\ &\quad \left\{ \begin{array}{l} W_{t+1} = W_t (1 + R_t) \\ Var_t^- \leq d_t \\ \sum_{i=1}^n x_{it} = 1 \\ l_t \leq x_{it} \leq u_t \end{array} \right. \end{aligned}$$

### 3.4 实证分析

为验证本文所提出的改进能量谷算法具有实用性，通过使用实际股票数据，利用上述多阶段模糊均值一下半方差模型进行实证分析。假设投资者选择上海证券交易所的 6 只股票和一种无风险资产进行投资，其中，6 只股票分别为中信证券（SH600030），浙江富润（SH600070），万通地产（SH600246），三友化工（SH600409），国药股份（SH600511），方大炭素（SH600516），同时，使用银行活期存款作为无风险资产，将上述资产依次编号为：A1, A2, A3, A4, A5, A6, A7，即  $n=7$ 。投资者选择进行 3 期投资，每个投资期持续 2 年，即  $T=3$ 。搜集上述股票 2014 年 1 月至 2020 年 1 月的周收益率数据作为样本，以两年为一次观察期，股票的详细模糊分布可见文献[13]。

模型中的参数设置如下：投资着初始财务  $W_1 = 1$ ，

银行活期存款利率为 2014 年中央银行设定的活期存款

利率  $r_f = 0.0035$ ，投资比例上界为  $u = 0.35$ ，投资

比例下界为  $l = 0.01$ ，投资组合风险  $d = 0.003$ 。买入

费用设置为  $c_{\text{买}} = 0.03741\%$ ，卖出费用设置为

$c_{\text{卖}} = 0.13741\%$ ，每一期的限制条件相同。

运用本文所设计的改进能量谷算法进行求解，为验证算法在实际问题中的性能表现，依然使用 EVO 算法和 PSO 算法进行对比实验，调整 IMEVO 算法的参数为  $w = 0.6$ ，其他算法的编码方式和参数设置均与仿真实验相同，图 3 给出三个算法独立运行 10 次后的平均效果。

表 4 三类算法运算结果对比表

投资组合模型									
	EVOPSO			EVO			PSO		
最优值	1.07651			1.07551			1.07647		
最劣值	1.07638			1.07289			1.07305		
标准差	4.79e-05			0.00093			0.001		

表 5 三类算法所计算出的最优投资比例

资产	IMEVO			EVO			PSO		
	T=1	T=2	T=3	T=1	T=2	T=3	T=1	T=2	T=3
A1	19.6%	8.0%	1.0%	17.8%	14.3%	1.0%	19.8%	11.5%	1.0%
A2	1.0%	24.1%	25.0%	1.0%	19.8%	26.0%	1.0%	24.6%	25.0%
A3	19.3%	8.1%	35.0%	22.1%	10.5%	34.8%	20.1%	7.6%	35.0%
A4	18.1%	1.9%	1.9%	22.0%	8.3%	1.8%	18.2%	1.1%	1.6%
A5	1.0%	1.0%	35.0%	1.0%	4.0%	33.1%	1.3%	1.4%	35.0%
A6	21.4%	31.8%	1.0%	17.7%	31.9%	2.3%	20.6%	31.4%	1.4%
A7	19.4%	25.1%	1.0%	18.4%	11.2%	1.0%	19.0%	22.4%	1.0%

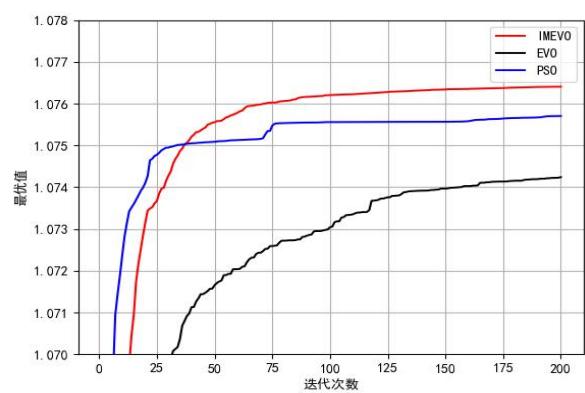


图 3 三种算法运算过程对比

根据图 3 和表 4, 表 5 可知, 在相同的风险水平下, IMEVO 算法可以搜索到使累计收益更高的资产配置比例, 同时, 在收敛速度上, IMEVO 算法在前期迭代中稍逊于 PSO 算法, 但在迭代后期时, PSO 算法陷入到了局部最优解当中, 而 IMEVO 算法能继续向全局最优解进行搜索, 而传统 EVO 算法在搜索精度和收敛速度上均劣于 IMEVO 算法。

## 4 结论

本文从能量谷算法 (EVO) 在处理优化问题中存在的局部搜索能力较差, 最近邻粒子的劣势以及更新流程的复杂性出发, 在粒子群算法和数学优化的思想上, 提出了改进能量谷算法 (IMEVO), 并通过六个经典测试函数进行对比测试, 结果表明, 改进能量谷算法在收敛速度, 搜索精度和算法稳定性上均优于传统能量谷算法, 验证了本文的改进方法具有有效性。同时, 在解决具有现实约束条件的投资组合问题中, 改进能量谷算法能更加有效的搜索到最优资产配置比例, 验证了算法在处理实际问题中具有实用性。能量谷算法作为新型的智能搜索算法之一, 目前使用其处理优化问题的研究案例较少, 本文的研究不仅优化了能量谷算法的性能, 还进一步丰富了算法的实际应用实例, 为后续应用到更多优化问题当中提供了一定的参考价值。

## 参考文献

- [1] Holland J H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence [M]. MIT press, 1992.
- [2] Kennedy J, Eberhart R. Particle swarm optimization[C]. Proceedings of ICNN'95-international conference on neural networks. IEEE, 1995, 4: 1942-1948.
- [3] 高文龙, 周天阳, 朱俊虎, 等. 基于双向蚁群算法的网络攻击路径发现方法[J]. 计算机科学, 2022, 49(S1): 516-522.
- [4] 自适应变异蝴蝶优化算法[J]. 计算机工程与科学, 2023, 45(06): 1123-1133.
- [5] 邵良杉, 闻爽爽. 基于海鸥优化算法的企业平衡运输问题研究[J]. 数据与计算发展前沿, 2022, 4(02): 121-130.
- [6] Mohamed Abdel-Basset, Reda Mohamed, Mohammad Jameel, Mohamed Abouhawwash. Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems[J]. Knowledge-Based Systems, 2023, 262.
- [7] 樊康生, 杨光永, 吴大飞, 等. 多策略融合的改进万有引力搜索算法[J]. 计算机应用研究, 2023, 40(12): 3592-3598.
- [8] Azizi M, Aickelin U, A. Khorshidi H, et al. Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization!]. Scientific Reports, 2023, 13(1): 226.
- [9] 谢七月, 刘广帅, 刘瑶, 等. 基于改进灰狼算法的定日镜场布局优化方法[J]. 太阳能学报, 2024, 45(11): 394-400.
- [10] 王海群, 郭庆通, 葛超. 基于改进蜣螂优化算法的搬运机器人路径规划[J]. 现代制造工程, 2024, (11): 7-95.
- [11] 孙正阳, 杜晔. 基于改进萤火虫算法的卫星网络路由优化方法[J/OL]. 系统工程与电子技术, 1-11.
- [12] 盛斌, 张军. 基于多策略改进麻雀算法优化DELM 的CSTR 辨识[J]. 化工设备与管道, 2024, 61(06): 7-16.
- [13] 陈思豆, 黄卓铨, 杨兴雨. 考虑限制性卖空的多期模糊投资组合优化模型[J]. 广东工业大学学报, 2021, 38(02): 39-47.

作者简介: 李成希 (2001—), 男, 重庆丰都人, 研究生, 研究方向: 投资组合、金融统计。

孙荣 (1973—), 男, 四川宣汉人, 研究生, 职称: 教授, 研究方向: 金融统计。