

假设在时刻 T 的预期收益率为 Y_0 , 此时第 i 种股票的股价和收益率为 p_i 和 y_i , 约束条件:

$$\frac{\sum_{i=1}^n 100n_i p_i y_i + y_{x_0} T - C}{T} \geq Y_0 \quad (6)$$

投资组合模型及其约束条件表示如下:

$$\min V = X' \sigma X \quad (7)$$

$$\left\{ \begin{array}{l} \frac{\sum_{i=1}^n 100n_i p_i y_i + y_{x_0} T - C}{T} \geq Y_0 \\ \sum_{i=1}^n 100n_i p_i + x_0 T \leq T \\ X_i = \frac{100n_i p_i}{T} (i = 1, 2, \dots, n) \\ \sum_{i=0}^n X_i = 1 \\ X_i \geq 0 (i = 0, 1, \dots, n) \\ n_i \in I, I \text{ 为非整数} \\ C = C_0 + \sum_{i=1}^n C_i(n_i) \end{array} \right. \quad (8)$$

3 基于天牛须搜索的粒子群优化算法

3.1 粒子群 (PSO) 算法原理

基于社会行为比拟的粒子群优化算法由 Eberhart 博士和 Kennedy 博士于 1995 年提出^[10]. 它起源于对鸟群捕食行为的研究. 基本概念是群体中的个体共享信息, 以便整个群体的运动在问题解决过程中从无序变为有序, 并最终获得问题的最佳解决方案. PSO 算法包括一些极大地影响算法性能的参数, 通常被称为探索-开发权衡: 探索是在问题空间中测试各个区域以便找到最优的能力.

PSO 算法通过设计无质量粒子来模拟鸟群中的鸟类. 粒子有两个属性: 速度 v 和位置 x , 速度代表运动速度, 位置代表运动方向. 算法中个体的极值 P_{best} 是每个粒子分别在搜索空间中搜索最佳解, 然后, 粒子与整个粒子群中的其他粒子共享个体极值 P_{best} 并找到最优个体极值, 将其作为整个粒子群的当前全局最优解 G_{best} . 粒子群中的所有粒子基于其当前的个体极值 P_{best} 和整个粒子群共享的当前全局最优解 G_{best} , 调整它们自己的速度和位置变量. 图 1 是 PSO 的流程图.

3.2 天牛须搜索 (BAS) 算法原理

天牛须搜索 (BAS) 算法基于天牛的觅食原理, 是

在 2017 年提出的一种寻找的最佳解决方案的新技术^[11,12]. 当觅食时, 天牛不知道食物的具体位置, 使用两个触角来检测食物的气味并决定其自身的方向. 具体来说, 如果甲虫左侧接收的气味强于右侧, 则甲虫向左移动, 否则向右移动. 基于这个简单的原理, 它可以很容易地找到食物. 与 PSO 类似, BAS 使用迭代方法逐步逼近最优解, 而不是了解具体公式. 在 BAS 算法中, 只有个体. 具体步骤如下:

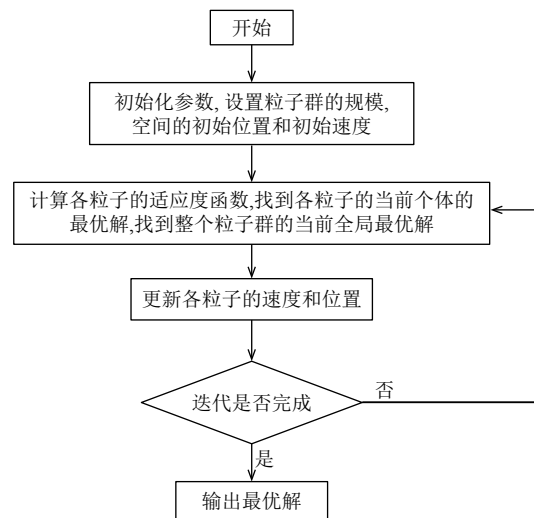


图 1 PSO 算法流程图

(1) 假设甲虫头随机向任何方向前进, 因此从右天线到左天线的矢量方向也必须是随机的. 因此, 对于 n 维空间中的优化问题, 可以生成随机向量来表示和标准化它.

$$\vec{b} = \frac{rands(k, 1)}{|rands(k, 1)|} \quad (9)$$

其中, k 是空间维度, $rands()$ 是随机函数.

(2) 左右天线之间的关系可表示为:

$$x_l - x_r = d_0 \cdot dir \quad (10)$$

x_l, x_r 可以用质心表示如下:

$$\begin{cases} x_l = x + d_0 \cdot dir / 2 \\ x_r = x - d_0 \cdot dir / 2 \end{cases} \quad (11)$$

其中, x_l 代表搜索区域的左侧, x_r 代表右侧.

(3) 确定左右触角的气味强度, 用 $f(x_l)$ 和 $f(x_r)$ 代替左右位置, $f(x)$ 是适应度函数.

(4) 为了制定检测行为指南, 本文进一步生成以下迭代模型, 通过考虑搜索行为并迭代更新甲虫的位置来检测气味.

$$x^{t+1} = x^t - \delta^t \vec{b} \text{sign}(f(x_{rt}) - f(x_{lt})) \quad (12)$$

其中, x^t 表示在甲虫的第 t 次迭代中的质心坐标, x_{lt} 表示在第 t 次迭代时的左侧天线坐标, x_{rt} 表示在第 t 次迭代时的右侧天线坐标. 第 t 次迭代的步长是 δ^t , $\text{sign}(x)$ 代表符号功能.

3.3 基于天牛须搜索的粒子群优化算法 (BSO)

(1) 基于天牛须搜索的粒子群优化算法 (BSO) 原理

BAS 算法仅针对个体, 不考虑群体之间的连接. PSO 侧重于群体对单个粒子的影响, 忽略了粒子在搜索过程中自己的判断. 因此, 本文将 BAS 和 PSO 模型集成在一起, 提出基于天牛须搜索的粒子群优化算法 (BSO). PSO 中的每个粒子都被描述为天牛并进行搜索, 天牛的初始位置和速度的过程与标准 PSO 的过程相同. 然而, 在迭代过程中, 更新天牛群位置的方式不再仅仅依赖于历史最佳解决方案和天牛个体的当前全局最优解, 而是添加了天牛天线搜索的思想, 在每次迭代中都有自己对环境空间的判断. BSO 中的个体将在每次迭代期间比较其左侧和右侧的适应度函数值, 并比较两者的更好的值, 其可用于更新天牛群的位置. 通过该方法构造的 BSO 可以很好地克服 PSO 算法导致的稳定性差, 倾向于局部最优等问题. 甲虫群位置的更新公式如下:

$$vb_i = -\delta^t \cdot \vec{b} \cdot \text{sign}(f(x_{rt}) - f(x_{lt})) \quad (13)$$

$$v_i^{k+1} = v_i^k + c_1 \cdot \text{rand} \cdot (Pb_i^k - x_i^k) + c_2 \cdot \text{rand} \cdot (Pg_i^k - x_i^k) + c_3 \cdot \text{rand} \cdot vb_i \quad (14)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (15)$$

其中, v_i^{k+1} , 表示第 k 次迭代后的第 i 个粒子的速度, x_i^{k+1} 表示第 k 次迭代后第 i 个粒子的位置, vb_i 表示 BSO 生成的更新率, c_3 是学习因子, $\text{sign}()$ 是符号函数.

(2) 算法过程

1) 初始化算法参数, 设置 PSO 的大小为 N , 学习因子为 c_1, c_2, c_3 , 惯性权重为 W , 以及每个甲虫两个天线之间的距离为 d_0 .

2) 随机初始化位置 x 和速度 v , 计算每个位置的适应度, 使用当前位置作为个体最优解 P_{best} , 最后通过比较得到当前的全局最优值 G_{best} .

3) 输入迭代:

① 随机化甲虫头部. 根据甲虫的位置计算每个甲虫的左侧距离 x_{left} 和适合度 f_{left} , 右侧距离 x_{right} 和适合度 f_{right} . 通过比较两者, 获得由群体中每个甲虫的左右

适合度生成的速度更新规则:

$$vb_i = -\delta^t \cdot \vec{b} \cdot \text{sign}(f(x_{rt}) - f(x_{lt})) \quad (16)$$

② 通过比较每个甲虫当前位置的适应度和, 得到个体最优解 P_{best} 和全局最优解 G_{best} , 通过当前的个体最优解和全局最优解生成速度更新规则.

③ 结合上述两种速度更新规则, 获取每种天线速度的当前更新规则:

$$v_i^{k+1} = v_i^k + c_1 \cdot \text{rand} \cdot (Pb_i^k - x_i^k) + c_2 \cdot \text{rand} \cdot (Pg_i^k - x_i^k) + c_3 \cdot \text{rand} \cdot vb_i \quad (17)$$

当前位置更新规则:

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (18)$$

④ 更新的学习因素和惯性权重分别为 c_1, c_2, c_3, w , 更新的个人最优解决方案和全局最优解决方案 $P_{\text{best}}, G_{\text{best}}$.

⑤ 完成迭代后, 可得到全局最优解 G_{best} , 以及与之对应的最优解位置 $f(G_{\text{best}})$.

具体算法过程如表 1 所示.

表 1 BSO 算法过程

算法	BSO 算法
输入	初始化算法参数, 获得初始个体最优解 P_{best} 和全局最优解 G_{best}
迭代	While (End condition) { 1. 计算每个甲虫的位置, 左右天线的位置的适应度 $fitness(x)$; 2. 根据计算的适应度, 生成每个甲虫的速度和位置的更新规则 3. 更新每个甲虫的速度 v 和位置 x . 更新学习因子, 惯性权重, 当前个体最优解 P_{best} 和全局最优解 G_{best} }
输出	全局最优解位置 G_{best} , 适应度函数 $f(G_{\text{best}})$

4 实证分析

为了测试 BSO 算法是否更优于解决投资组合问题, 本文将 BSO 算法与标准 PSO 算法进行了比较, 即将两者在求解投资组合时所获得的风险值与需要的迭代次数进行了对比.

4.1 数据来源和索引选择

为了测试 BSO 对人工智能投资组合模型的结果, 本文选择了 9 只人工智能概念股, 并获取股票从 2017 年 6 月 23 日到 2017 年 8 月 18 日 (9 周) 的周回报率. 计算出协方差矩阵如表 2, 可以用来衡量九种股票的风险. 收益率和股票报价分别选择周收益率和价格近似

于9周收益率和平均价格,如表2所示。

表2 所选股票的协方差矩阵

No.	1	2	3	4	5	6	7	8	9
1	10.8	0.8	3.9	2.9	-4.2	3.1	-1.0	3.9	-0.5
2	0.8	1.5	0.9	-0.2	-2.6	0.8	-1.3	0.9	0.4
3	3.9	0.9	2.2	0.9	-2.2	1.9	-1.1	2.2	0.1
4	2.9	-0.2	0.9	2.0	0.2	0.9	0.0	0.9	-0.2
5	-4.2	-2.6	-2.2	0.2	10.4	-0.8	-1.9	-2.2	-1.2
6	3.1	0.8	1.9	0.9	-0.8	2.0	-1.4	1.9	0.2
7	-1.0	-1.3	-1.1	0.0	-1.9	-1.4	13.2	-1.1	-0.1
8	3.9	0.9	2.2	0.9	-2.2	1.9	-1.1	2.2	0.1
9	-0.5	0.4	0.1	-0.2	-1.2	0.2	-0.1	0.1	2.4

数据来源: Wind 数据库

同时,无风险资产为一年期国债(收益率为3.85%),假设总共有100万资产可用于投资,并且在投资前没有购买任何证券。

4.2 参数设置

文章以交易手数向量作为粒子群和天牛群的位置变量,股票的数目为变量空间的维度数目,因此模型的空间维度数为9。PSO和BSO算法设置的粒子数和天牛数都为40,粒子群设置的迭代的次数为8000次,天牛群设置的迭代次数为4000次。PSO算法的学习因子设置为 $c_1 = 0.5, c_2 = 2$; BSO算法使用的学习因子为 $c_1 = 0.5, c_2 = 2, c_3 = 3$;两种算法的惯性权重相同,都设置为 $w = 0.9$ 。两种学习因素都是随时间变化的,并采用惩罚法来处理约束条件。

4.3 实验结果

为验证算法的性能,将总投资 T 设为100万,股价为 p_i 、收益率 y_i 设置为9只人工概念股的平均价格与周收益率(见表3)。PSO和BSO参数比较见表4。将预

期投资组合回报率分别设置为0.06、0.08和0.10。算法确定投资组合中各种股票的投资比例 x_i 及投资于无风险资产的投资比例 x_0 ,如表5所示。无风险资产的收益率为3.85%,将表2确定的协方差矩阵,表5确定的投资比例向量代入目标函数(7)及约束条件(8),计算相应的风险值,结果见表5。

表3 所选股票的周收益率和价格

股票	收益率	股价(元)
科大讯飞	0.0190	19.62
中科曙光	0.1900	24.02
浪潮软件	0.0358	25.18
机器人	0.0343	12.95
浙大网新	0.0466	24.56
海康威视	0.2288	26.66
昆仑万维	0.0188	17.25
浪潮信息	0.0422	21.50
科大智能	0.0679	23.47

数据来源: Wind 数据库

表4 PSO和BSO算法参数的比较

算法参数	PSO	BSO
粒径	40	40
迭代次数	8000	4000
学习因子	$c_1=0.5, c_2=2$	$c_1=0.5, c_2=2, c_3=3$
学习因子新方法	$c_1=0.5+2*t/t_{max}$ $c_2=2-2*t/t_{max}$	$c_1=0.5+2*t/t_{max}$ $c_2=2-2*t/t_{max}$ $c_3=3+2*t/t_{max}$
惯性权重	$w=0.9$	$w=0.9$
惯性权重新方法	$w=0.9-0.5*t/t_{max}$	$w=0.9-0.5*t/t_{max}$

表5 所选股票投资组合的投资比例

预期收益率	0.06		0.08		0.1	
	PSO	BSO	PSO	BSO	PSO	BSO
无风险资产	0.717 970	0.781 810	0.482 150	0.581 880	0.304 640	0.403 910
SHE:002230	0.005 332	0.015 247	0.018 662	0.029 391	0.037 324	0.048 184
SHA:603019	0.136 910	0.130 070	0.254 610	0.243 910	0.367 510	0.352 790
SHA:600756	0.030 511	0.000 184	0.000 000	0.000 994	0.002 347	0.000 547
SHE:300024	0.001 962	0.001 163	0.005 886	2.51E-06	0.000 000	0.000 547
SHA:600797	0.044 026	0.043 746	0.081 578	0.085 477	0.135 960	0.121 000
SHE:002415	0.002 456	0.000 702	0.061 403	6.96E-05	0.012 281	6.41E-05
SHE:300418	0.027 700	0.019 444	0.060 437	0.039 896	0.015 109	0.055 555
SHE:000977	0.005 176	8.36E-06	0.005 176	9.36E-05	0.043 136	0.000 120
SHE:300222	0.027 947	0.007 632	0.030 097	0.018 285	0.081 692	0.016 839
风险值	1.69E-05	9.75E-06	6.57E-05	3.42E-05	1.42E-04	7.35E-05
迭代时间(s)	60	40	60	40	60	40

从表5的结果可以看出,对于不同的预期收益率,BSO计算的风险值低于PSO。也就是说,BSO可以在

相同的预期回报率下获得相对更好的结果。这表明BSO算法具有更强的全局搜索能力,并且该算法更容

易找到全局最优解. 另一方面, 当预期收益率增加时, 两种算法获得的风险值增加, 无风险资产的投资比例显著下降, 高收益股票的风险值显著上升. 可以看出, 本文构建的投资组合模型符合实际市场条件. 在迭代时间上, PSO 的迭代时间为 60 s, BSO 的迭代时间为 40 s, BSO 的迭代时间更短, 更具优势.

图 2 和图 3 显示了两种算法的收敛性. 从算法收敛的角度来看, PSO 算法大约有 3000 次迭代收敛, 而 BSO 算法大约只有 1500 次. 可以看出, 与标准 PSO 相比, 在引入天牛须搜索概念后, BSO 算法每个粒子的更新方法不再仅依赖于自身的历史最优解和全局最优解, 在每次迭代期间增添粒子自己的独立判断, 这使得粒子的迭代方法更灵活, 更智能, 使得收敛所需的迭代次数更少, 结果更好.

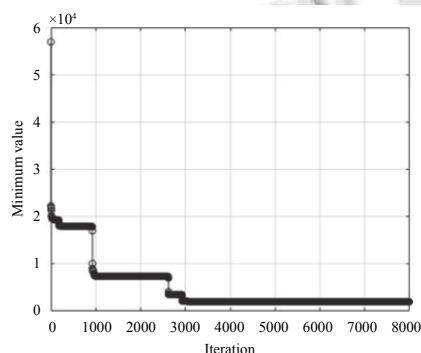


图 2 PSO 算法收敛曲线

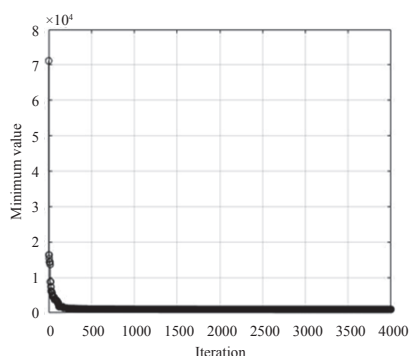


图 3 BSO 算法收敛曲线

4.4 总结

本文针对标准粒子群容易陷入局部最优和搜索精度不高的缺点, 在粒子群算法中引入了天牛须搜索概念, 提出基于甲壳虫触角搜索 (BAS) 的粒子群优化算法, 即 BSO 算法, 并将其应用在考虑现实约束条件、包含完整费用的投资组合模型求解中. 在迭代过程中, 每个粒子的更新规则源自 BAS, 更新粒子群位置的方

式不再仅仅依赖于历史最佳解决方案和粒子个体的当前全局最优解, 而是增添粒子在每次迭代中对环境空间的自身判断. 经实证分析, 对于不同的预期收益率, BSO 计算的风险值低于 PSO, 算法收敛所需的迭代次数远少于 PSO. 并且, 从迭代时间来看, BSO 所需时间仅为 PSO 的 2/3. BSO 显示出比标准 PSO 更强的全局搜索能力, 在求解多维约束问题时寻找全局最优解稳定性和有效性更佳.

参考文献

- 1 Khairalla M, Ning X, Al-Jallad N. Modelling and optimisation of effective hybridisation model for time-series data forecasting. *The Journal of Engineering*, 2018, 2018(2): 117–122. [doi: 10.1049/joe.2017.0337]
- 2 Syahputra R, Wiyagi RO, Suropto S, *et al.* A novel fuzzy approach for multi-objective optimization of distribution network configuration in complex system. *International Journal of Applied Engineering Research*, 2018, 13(2): 1120–1127.
- 3 陈炜, 张润彤, 杨玲. 基于改进粒子群算法的投资组合选择模型. *计算机科学*, 2009, 36(1): 146–147, 204. [doi: 10.3969/j.issn.1002-137X.2009.01.035]
- 4 刘晓峰, 陈通, 张连营. 基于微粒群算法的最佳证券投资组合研究. *系统管理学报*, 2008, 17(2): 221–224, 234.
- 5 杨建辉, 江文婷. 基于 PSO 的考虑完整费用的证券组合优化研究. *计算机应用研究*, 2010, 27(9): 3364–3367. [doi: 10.3969/j.issn.1001-3695.2010.09.043]
- 6 刘冬华, 甘若迅, 樊锁海, 等. 基于捕食策略的粒子群算法求解投资组合问题. *计算机工程与应用*, 2013, 49(6): 253–256, 261. [doi: 10.3778/j.issn.1002-8331.1111-0310]
- 7 吴喆珺. 改进的 PSO 算法及其在证券组合投资中的应用. *武汉职业技术学院学报*, 2014, 13(1): 36–41, 69.
- 8 朱沙, 陈臣. 一种求解基数约束投资组合优化的混合粒子群算法. *统计与决策*, 2016, (10): 64–67.
- 9 Zhu ZY, Zhang ZY, Man WS, *et al.* A new beetle antennae search algorithm for multi-objective energy management in microgrid. 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA). Wuhan, China. 2018. 1599–1603.
- 10 Eberhart R, Kennedy J. A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. Nagoya, Japan, Japan. 1995. 39–43.
- 11 Jiang XY, Li S. BAS: Beetle antennae search algorithm for optimization problems. arXiv:1710.10724, 2017.
- 12 Wang JY, Chen HX. BSAS: beetle swarm antennae search algorithm for optimization problems. arXiv:1807.10470, 2018.