天津科技大学学报
*Journal of Tianjin University of Science & Technology*
ISSN 1672-6510,CN 12-1355/N

# 《天津科技大学学报》网络首发论文

题目：面向强相关背包问题的 BPSO 算法优化
作者：赵青，刘涛，李昊
DOI：10.13364/j.issn.1672-6510.20240208
收稿日期：2024-10-16
网络首发日期：2025-06-19
引用格式：赵青，刘涛，李昊．面向强相关背包问题的 BPSO 算法优化[J/OL]．天津科技大学学报. https://doi.org/10.13364/j.issn.1672-6510.20240208

DOI:10.13364/j.issn.1672-6510.20240208

# 面向强相关背包问题的 BPSO 算法优化

赵　青，刘　涛，李　昊
(天津科技大学人工智能学院，天津 300457)

**摘　要：**二进制粒子群（BPSO）算法是解决离散二进制空间最优化问题的重要群体智能算法，而背包问题经常被用作 BPSO 算法的超高维度优化问题的测试。背包问题中的强相关背包问题，一直是这一领域的研究难点。这是因为强相关背包问题的解在解的空间范围上过于集中，导致 BPSO 算法在寻优过程中很可能忽略解集中聚集空间，导致搜索到的解与最优解相差较大。本文提出一种专门针对强相关背包问题的改进方法，即基于价值密度排序的分级竞优策略的 BPSO 算法。算法来源于分级淘汰赛的思想，增加全局搜索阶段空间上的搜索细腻程度，从而增加最优解聚集小区域被发现的可能性，避免过早陷入局部最优陷阱。实验结果表明，无论是低维度、中维度还是高维度的背包问题，均能以较高的准确率接近真实最优解，提高强相关性背包问题求解的准确性和可靠性。

**关键词：**二进制粒子群（BPSO）算法；背包问题；分级竞优策略，基于价值密度展开策略
**中图分类号：**TP301.6　　**文献标志码：**A　　**文章编号：**1672-6510（0000）00-0000-00

# Optimization of BPSO Algorithm for Strongly Correlated Knapsack Problem

ZHAO Qing，LIU Tao，LI Hao
(College of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin 300457, China)

**Abstract:** The binary particle swarm optimization (BPSO) algorithm is an important swarm intelligence algorithm for solving discrete binary space optimization problems, and the knapsack problem is often used as a test for the ultra-high dimensional optimization problems of BPSO algorithm. The strong correlation backpack problem in the backpack problem has always been a research difficulty in this field. This is because the solution of the strongly correlated knapsack problem is too concentrated in the range of the connected space, which leads to the BPSO algorithm is likely to ignore this solution concentrated aggregation space in the process of searching for the optimal, resulting in the search for the solution and the optimal solution is a large difference. This article proposes an improved method specifically for the strongly correlated knapsack problem, namely the BPSO algorithm based on value density ranking and hierarchical competitive optimization strategy. The algorithm is derived from the principle of graded elimination tournament, which increases the degree of spatial search finesse in the exploration phase, thus increasing the possibility of the optimal solution aggregating small regions to be discovered, thus avoiding falling into the local optimality trap prematurely. The experimental results show that the knapsack problem can approach the true optimal solution with high accuracy whether in low, medium or high dimensions, which improves the accuracy and reliability of solving the strong correlation knapsack problem.

**Key words:** binary particle swarm optimization（BPSO）algorithm；knapsack problem；graded competitive strategy；strategy based on value density expansion

粒子群算法（PSO）属于群体智能优化算法的一种，具有准确度高、收敛速度快等优点，而二进制粒子群（BPSO）算法是粒子群算法在二进制离散空间的扩展版本，当前在适用于高维度小样本数据的

**收稿日期：**2024-10-16；**修回日期：**2025-03-14
**基金项目：**国家自然科学基金项目(12273077)；国家重点研发计划项目(2022YFF0711500)
**作者简介：**赵　青（1983—），女，山东东营人，副教授，zhaoqing@tust.edu.cn

基于传统机器学习模型的特征选择[1]、超参数优化[2]、深度学习模型[3]的结构优化以及多领域的调度最优化问题均有广泛的应用。但是，BPSO 算法也存在一定缺陷，传统上认为它的后期局部搜索能力不足，容易陷入局部最优解。近年来，BPSO 算法的改进研究多围绕着新型传递函数的提出以及在算法的不同阶段动态调整参数平衡勘探和开采功能[4]而展开。从转换函数的角度对算法进行改进，虽然取得了一定的改进效果，但是结果仍不佳[5]。算法在前期往往在重复解空间中搜索，而在后期为了达到收敛效果，往往通过限制某些参数使搜索能力下降过快，算法容易陷入局部最优[6]。为了进一步平衡勘探和开采，避免过早陷入局部最优，一些研究采用了与其他算法相结合的复合算法，比如在参考文献中将二进制黑洞算法与 BPSO 算法相结合，在一定程度上避免了局部极小[7]；将 BPSO 算法与对抗学习、混沌映射、基于适应度的动态惯性权重和变异相结合，实现高聚类准确度并提高了 BPSO 算法的性能[8]；基于多阈值粒子群优化的关联规则挖掘模型，获得更高的适应度值[9]。GA-BPSO 算法[10]通过结合遗传算法（GA）的交叉和变异操作，增强了算法的全局搜索能力，并帮助其跳出局部最优解。这些混合算法虽然在一定程度上改进了 BPSO 算法容易陷入局部最优的情况，但也增加了模型的复杂度，而且修改后 BPSO 算法的寻优能力仍然难以达到 PSO 算法在连续空间寻优问题中的优势。Zhao 等[11]提出了一种基于速度遗留项校正的 V 型变换函数的改进方法（VCv-BPSO 算法），在价值与重量随机的低维度、中维度、高维度上的随机背包问题，4 种常见的 V 型 BPSO 算法经过该改进后取得明显效果，但是对于弱相关和强相关这一类特殊背包问题，改进结果不明显，尤其是强相关背包问题。

强相关背包问题的特点是物品的价值与重量高度相关。强相关背包问题中的物品价值通常与其重量成正比。由于价值与重量高度相关，强相关背包问题的解空间呈现出更强的非线性特性，增加了问题的求解难度，且随着问题规模的增加，解空间呈指数级增长，对算法的计　算效率和搜索能力提出更高的要求。强相关背包问题的特性使其成为测试优化算法性能的理想选择。在需要同时考虑多个相互关联的因素和限制的场景中，通过将这些场景抽象为强相关背包问题，可以使用相应的算法和技术找到最优解或近似最优解。因此，强相关性背包问题在现实中具有广泛的应用场景，尤其是在资源有限且需要最大化效益的领域，例如在资源分配、金融投资、工业生产等领域，可以显著提高经济效益和社会福祉。

经过研究发现，强相关背包问题之所以难以寻优，是因为最优解和次优解分布在整个空间的一个狭小空间内，启发式算法通常是由全局搜索逐渐过渡到局部搜索的过程，但是这个过程持续的时间短，搜索不够充分，难以发现最优解所在的狭小空间，从而导致最终结果与真实全局最优解相差较大。因此，本文在 VCv-BPSO 算法[11]的基础上提出一种基于价值密度分级竞优的 BPSO 算法，引入了“分级淘汰赛”的思想，增加全局搜索阶段空间上的搜索细腻程度，增加最优解聚集小区域被发现的可能性，从而避免过早陷入局部最优陷阱；另一方面，引入价值密度排序，目的是在搜索开始前，使解的多维空间的价值的分布存在规律性，便于启发式搜索，也使那些少量的次优解自然地聚集起来，便于全局搜索时发现，且次优解聚集的地方存在自然坡度的分布规律，使局部寻优更有效。

## 1　基于价值密度分级竞优策略的 V 型 BPSO 算法

### 1.1 传统 S 形 BPSO 算法的原理和不足分析

BPSO 根据式（1）进行速度更新后，通过 Sigmoid 函数将更新后的新速度映射到[0,1]空间，然后根据式（3）进行粒子更新。

$$V_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1 r_1 \left(p_{Best}^{t} + x_{ij}^{t}\right) + c_2 r_2 \left(g_{Best}^{t} + x_{ij}^{t}\right) \quad (1)$$

$$\text{Sigmoid}\left(v_{ij}\right) = \frac{1}{1 + \exp\left(-v_{ij}\right)} \quad (2)$$

$$x_{ij} = \begin{cases} 1, \text{如果Sigmoid}\left(v_{ij}\right) \geq rand\left(\ \right) \\ 0, \text{其他} \end{cases} \quad (3)$$

式中：$w$ 表示惯性权重，$c_1$、$c_2$ 为学习因子，$r_1$、$r_2$ 表示(0,1)之间的随机数，$p_{Best}^{t}$ 表示为某粒子在进化中寻找到的最优解，$g_{Best}^{t}$ 表示为粒子群在进化过程中寻找到的最优解，$t$ 代表第 $t$ 次迭代，$x_{ij}^{t}$ 代表粒子 $i$ 的第 $j$ 个分量在第 $t$ 次迭代的取值。

PSO 算法在变化为二进制离散空间的 BPSO 算法后，粒子按照式（3）概率进行轨迹变化。位值为 1 的概率为 $Sigmoid(v_{ij})$，而位值为 0 的概率为 $1 - sigmoid(v_{ij})$。此时，若位值已经是 0，则发生

位值跳转的概率为$sigmoid(v_{ij})$，如果位值已经是 1，则发生位置跳转的概率为$1-sigmoid(v_{ij})$。因此，可以得到位值发生跳转的绝对概率为

$$P_{跳转}=\text{Sigmoid}\left(v_{ij}\right)*\left(1-1-\text{Sigmoid}\left(v_{ij}\right)\right)+\left(1-\text{Sigmoid}\left(v_{ij}\right)\right)*\text{Sigmoid}\left(v_{ij}\right) \quad (4)$$

化简后为

$$P_{跳转}=\text{Sigmoid}\left(v_{ij}\right)*\left(1-2*\text{Sigmoid}\left(v_{ij}\right)\right) \quad (5)$$

S 型转换函数曲线如图 1 所示。原始 BPSO 算法的转换函数为 S 型，通过式（2）可知，当速度$v_{ij}$很大时，$sigmoid(v_{ij})$无限接近于 1，导致结果总为 1，过早收敛；反之，当速度无限小趋于 0 时，也容易导致结果总为 0。由图 1 可知，当速度接近 0 时，位值跳转的概率并不是逐渐减少，而是逐渐增大。这与粒子群算法的启发式策略并不相符。PSO 算法的寻优策略一般是：开始阶段采取全局搜索策略，速度较大，在全局范围内大范围地搜索优秀区域，然后逐渐由全局搜索转为局部搜索，此时应该在确定的优秀区域的局部展开细腻的局部搜索，速度应逐渐降低，以在局部范围内精细地搜索最优解，整个算法呈逐渐收敛态势。但是，从图 1 可以看出，S 型二进制粒子群算法显然违背了原始粒子群算法的原理，速度逐渐减少时，位值跳转概率却在增大，无法开展精细的局部搜索且算法难以收敛。

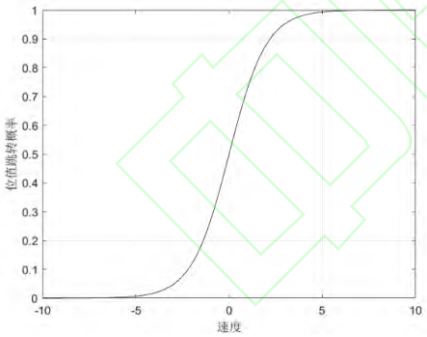



图 1　S 型转换函数曲线

**Fig. 1　S-shaped transfer function**

### 1.2　V 形 BPSO 算法的优势及强相关背包场景下的改进策略分析

V 型 BPSO 算法是在原始 S 型 BPSO 算法上提出的变种，根据具体转换函数的不同，常见的 V 形转换函数见表 1。

表 1　V 型转换函数

**Tab. 1　V-shaped Transfer function**

| VC-vg BPSO 算法 | 转换函数 |
|---|---|
| VC1-vg | $\text{Sigmoid}\left(v_{ij}\right)=\left\lvert\frac{2}{\pi}\arctan\left(\frac{\pi}{2}v_{ij}\right)\right\rvert$ |
| VC2-vg | $\text{Sigmoid}\left(v_{ij}\right)=v_{ij}^2/\left(1+v_{ij}^2\right)$ |
| VC3-vg | $\text{Sigmoid}\left(v_{ij}\right)=\left\lvert\tanh\left(v_{ij}\right)\right\rvert$ |
| VC4-vg | $\text{Sigmoid}\left(v_{ij}\right)=2/\left(1+e^{-v_{ij}}\right)-1$ |

与 S 型 BPSO 算法不同的是，V 型 BPSO 算法约定的不是粒子跳转到 1 的概率，而是粒子发生跳转的概率，公式为

$$x_{ij+1}=\begin{cases}1-x_{ij}, \text{如果} rand(\ )<\text{Sigmoid}\left(v_{ij}\right)\\ x_{ij}, \text{其他}\end{cases} \quad (6)$$

这一改变的优点是：当速度为 0 时，跳转概率为 0；当速度绝对值增大时，跳转概率逐渐增大，直至接近 1。这种趋势使算法在全局搜索阶段保持较大的跳转概率，而在局部搜索阶段逐渐收敛，保证搜索的全面性和细腻性，更符合 PSO 算法的逐步优化逻辑。V 型转换函数曲线如图 2 所示。V 型 BPSO 算法更符合原始 PSO 算法的寻优原理：当搜索进入后期时，即从全局搜索逐渐过渡为局部搜索时，速度逐渐降低，粒子变化逐渐减少。因此，V 型 BPSO 算法具有更好的后期局部搜索能力，且更容易收敛。这也是为什么本文选取 V 型 BPSO 算法进行强相关背包问题背景下的进一步改进研究的原因。

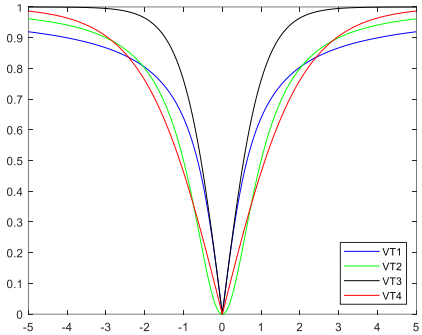



图 2　V 型转换函数曲线

**Fig. 2　V-shaped transfer function**

强相关背包问题的优秀解在解的空间里分布过于集中，而 BPSO 算法在初始撒点时可能并没有落入最优解集中的范围，导致搜索到的解是次优解而不是最优解。因此，本文在速度遗留项[11]的基础上提出改进策略，使 BPSO 算法在全局搜索阶段能够更充分的在解的空间上进行搜索，避免 BPSO 算法陷入最优解陷阱。

### 1.3　适合强相关背包问题的基于价值密度分级竞优

的 V 形 BPSO 改进方法

强相关背包问题最主要的难点在于优秀解在解空间分布上过于集中，可能只集中于整个解空间中的某一个角落，本文称之为“优秀角落”。BPSO 算法在执行过程中需要选取初始解，这些初始解很有可能没有落入优秀角落。搜索开始以后，这些初始粒子受当前搜索到的最优粒子的引导，很快由全局搜索转为局部搜索，落入某个不含次优解的地方进入局部搜索并最终收敛，导致搜索到的解与真正的全局最优解相差较大。因此，需要一种方案，在搜索的初始阶段，应该在全局范围内更广泛全面地搜索各个地区的优秀解，而不是很快地向当前全局最优粒子靠拢，从而达到尽可能发现强相关背包问题中次优解集中的小角落的目的。这个全局搜索过程要受到效率问题的制约，否则启发式算法将退化为暴力搜索法。针对以上分析，本文提出了基于分级竞优的搜索思路和基于价值密度排序的搜索空间开展策略，以在全局寻优阶段更全面地搜索优秀解，同时尽可能地减少搜索次数。

1.3.1　分级竞优策略

借用分级淘汰赛的思想，这一思路因为在早期增加了局部地区范围内的局部对比机会，没有过早地被发现的全局最优而带偏，从而可以更有机会找到各个地区的优秀解，减少优秀小角落被遗漏的概率，同时又没有进行全局范围的两两比较，没有增加过多的计算复杂度，避免退化为暴力搜索的风险。

上述思想的具体实现流程为先只有价值密度最大的 $k$ 位可动，这相当于区域比赛选拔，在经过一定的迭代次数后（一定迭代次数用 a 表示），可动位数增加至 $2k$ 位，相当于进行省级比赛选拔，继续迭代 a 次，然后不断重复上述过程直到所有位全部可动，比赛相当于进入全球比赛阶段，这个过程的示意图如图 3 所示。之后应用 VC-BPSO 算法在全部位上展开迭代，此后工作与一般 BPSO 算法一致，由全局搜索逐渐转变为局部搜索，最终收敛。

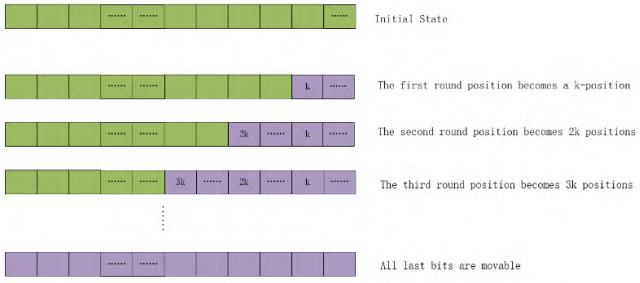



图 3　位变示意图

**Fig. 3　Diagram of displacement**

这样的搜索策略使算法在早期探索阶段能够更全面地对解进行搜索，在后期收敛阶段能够更精确地定位到局部最优或全局最优解，避免 BPSO 算法过早的陷入局部最优解。

1.3.2　基于价值密度展开策略

在开启搜索之前，本文提出先对粒子的所有位从左到右按照价值密度升序的顺序进行排列，如图 4 所示，然后在解搜索的过程中从右侧最低位（价值密度最高位）开始逐渐成为可变位，使搜索空间逐渐扩大，直至扩展到全解空间。采用这种按照价值密度由高到低的顺序（正序）逐渐展开搜索，目的是使解的多维空间价值的分布存在规律性，可以自然形成一些的“山坡”、“坡峰”、“坡谷”。其中“坡峰”是指解空间中的局部最优解或全局最优解，“坡谷”是指解空间中价值较低的区域，“山坡”是指解空间中价值逐渐上升的区域。价值密度排序使高价值密度的物品优先被选择，低价值密度的物品被优先排除，从而减少需要搜索的解空间规模，使解的总价值在搜索初期就处于较高水平。通过正序搜索，算法能够快速接近局部最优解，并在其附近进行精细搜索，避免在低价值区域浪费计算资源。按价值密度排序后，解空间的拓扑结构变得更加有序，高价值区域集中分布，便于二进制粒子群算法的启发式搜索，也使少量次优解自然聚集，便于全局搜索时发现。次优解聚集的地方存在自然坡度的分布规律，使局部寻优更有效。


图 4　价值密度示意图

**Fig. 4　Value Density Diagram**

1.3.3　算法流程：流程图及伪代码

算法伪代码见算法 1，流程图如图 5 所示。

算法 1

输入：背包问题实例背包容量 $C$、物品数量 $M$、物品的重量或体积 $W$、物品价值 $V$；粒子个数 $N$、迭代次数 $T$、惯性权重 $w$、学习因子 $c1$ 和 $c2$。

输出：全局最优解

1. 初始化

（1）参数设置：粒子数量 $N$、最大迭代次数 $T$、惯性权重 $w$、学习因子 $c1$ 和 $c2$、

（2）计算价值密度并排序

（3）初始化粒子群：每个粒子的位置$X_i$、每个粒子的速度$V_i$、粒子个体最优位置$P_i = X_i$、全局最优位置 $G$

2. 适应度函数$f(X_i)$

（1）计算总重量$\text{total_weight} = \sum_{j=1}^{M} x_{ij} \cdot w_j$

（2）If $\text{total\_weight} > C$，则$f(X_i) = 0$

（3）Else $f(X_i) = \sum_{j=1}^{M} x_{ij} \cdot v_j$

3. 主循环

for t = 1 to T do:

for 每个粒子 i = 1 to N do:

根据公式（1） 更新速度，

根据公式（2）将速度映射到 [0, 1] 区间,TF = sigmoid(Vi)

$G = k * j$(k=1,2,3.....n);

if $G > C$ ,则 $G = C$ ;

for g = 1 to G do:

if $TF >$ rand(), 更新粒子的位置，进行速度修正

end for

计算适应度$f(X_i)$;

更新个体最优 if $f(X_i) > f(P_i)$ then $P_i = X_i$;

更新全局最优 if $f(X_i) > f(G)$ then $G = X_i$;

end for

$t = t + 1$;

if $t\%a == 0$（t 达到一定次数）则 $j = j + 1$;

end for

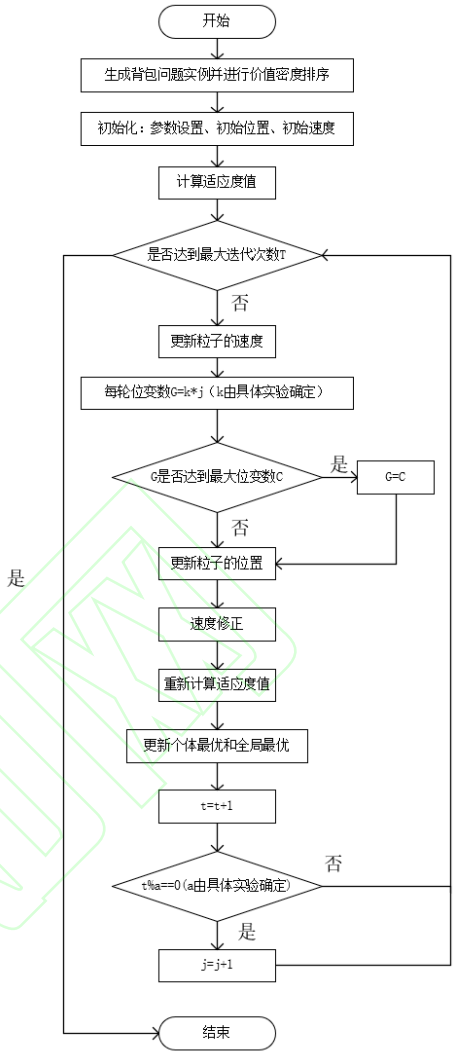



图 5　VC-vg BPSO 求解强相关背包问题的流程图

**Fig. 5　Flow chart for solving the strongly correlated knapsack problem using VC vg BPSO**

## 2　实验设计与结果分析

为验证 VC-vg BPSO 算法在解决强相关背包问题时的性能，设计 3 种不同维度（100、500、1000）的强相关背包问题，在 4 种常用的转换函数下进行实验，转换函数见表 1。实验设计对比算法分别为传统的 BPSO 算法，本文称 VT 算法；基于速度遗留项修正的 BPSO 算法，本文称 VC 算法[11]；基于价值密度乱序且采用分级竞优策略的 BPSO 算法，本文称 VC-g 算法；基于价值密度正序且采用分级竞优策略的 BPSO 算法，本文称 VC-vg 算法；基于价值密度逆序且采用分级竞优的 BPSO 算法，本文称 VC-vgr 算法。

实验环境：Win11 操作系统，Intel Core i5-13500H

CPU 处理器，16 GB 内存，MATLAB 2017A 开发环境。VT 算法的参数设置为学习因子 *c1*=*c2*=1，速度最大值$V_{max}$= 2，速度最小值$V_{min}$=-2，物体的价值 $V$ 与体积 $W$ 的关系为$\mathrm{V}=\mathrm{W}+u/8$，背包容量$\mathrm{C}=0.4*sum(\mathrm{W})$，3 种不同维度的背包实验除维度、初始粒子群大小和迭代次数不同，最主要的区别在于分级方式不同。

实验设计中求解质量的百分比（$R$），用于衡量 BPSO 算法得到的最优解（$N_{\mathrm{BPSO}}$）与动态规划法得到的最优解（$N$）之间的接近程度，公式为

$$R=\left(N_{\mathrm{BPSO}}/N\right)\times 100\% \qquad (7)$$

实验过程中参数 $w$ 选取方式为：VT 算法，$w$ 值分别选取 0.5、0.6 等恒定值以及 1.0-0.4 的线性匀速下降值；根据前人成果的经验，$w$ 大于 0.6 时 VT 算法一般不收敛。对于速度修正后的版本，$w$ 值分别测试恒取 1.0、0.99，以及 1.2-1.0、1.1-0.99 等多种匀速下降取值。经过速度修正后，并不需要依赖 $w$ 值限制粒子的后期活动能力实现收敛，算法可以成自然收敛态势，并已证明了 VC 算法的收敛性质[11]，$w$ 恒取 1 即可取得很好的解，但有时取 0.99 可以在对精度影响不大的情况下大幅度提升效率。1.2-1.0、1.2-0.99、01.1-1.0、1.1-0.99 这几种匀速下降取值有时在实验中也能取得不错的效果，因此也列入了实验范围，部分结果见表 2。

表 2 参数 *w*-转换函数 1

**Tab. 2　Parameter *w*-Transfer function 1**

| w | VT1 | | VC1 | | VC1-g | | VC1-vgr | | VC1-vg | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 全局最优解 | 百分比 | 全局最优解 | 百分比 | 全局最优解 | 百分比 | 全局最优解 | 百分比 | 全局最优解 | 百分比 |
| 0.6 | 25931.80 | 95.71% | 25303.20 | 93.39% | 25188.80 | 92.96% | 25092.80 | 92.61% | 25100.20 | 92.64% |
| 1.0-0.4 | 27206.90 | 95.42% | 27611.20 | 96.83% | 30069.40 | 94.17% | 27405.90 | 96.11% | 27908.80 | 97.88% |
| 0.99 | 25562.70 | 92.42% | 26963.60 | 97.73% | 26820.60 | 97.20% | 26672.50 | 96.68% | 27077.50 | 98.15% |
| 1.0 | 25910.40 | 93.47% | 27177.50 | 98.00% | 28095.50 | 97.33% | 26983.20 | 97.18% | 27284.40 | 98.43% |
| 1.1-0.99 | 26268.80 | 92.70% | 27538.60 | 97.18% | 27412.00 | 96.74% | 27413.90% | 96.74% | 27635.50 | 97.52% |

## 2.1　低维度背包实验设计与实验结果

基于价值密度的分级竞优策略即初始先对粒子进行价值密度排序，然后在 BPSO 算法进行迭代寻优时，先规定部分搜索范围并充分搜索，找到此范围的最优解，随后扩大搜素范围继续搜索，找到大范围的最优解，淘汰较差的次优解，保留最优解，继续扩大范围，重复上述过程，在迭代到一定次数时，开始全局搜索。在实验中，初始粒子的位变数量 $k$，相当于规定初始寻优范围，在此范围里搜索，随着迭代次数的增加，位变数量增加为 $2k$，这相当于扩大搜索范围，随着迭代寻优范围不断扩大，当粒子的全部位都变动时进入全局搜索阶段。实验设计不同的 $k$ 值表示寻优范围并验证该方法的寻优效果，位变数量按照迭代次数$\mathrm{a}=10$进行增加。在维度为 100 的低维度背包问题中，粒子群大小为 20，迭代次数为 250，以 $k$ 为 2、5、10、20、25 为例，具体数据见表 3。由表 3 可知，不同的范围扩大方式寻优效果不同，当 $k$=10 时，4 种转换函数的寻优效果最好，百分比均在 98%以上。因此，分级方式选择 $k$=10。

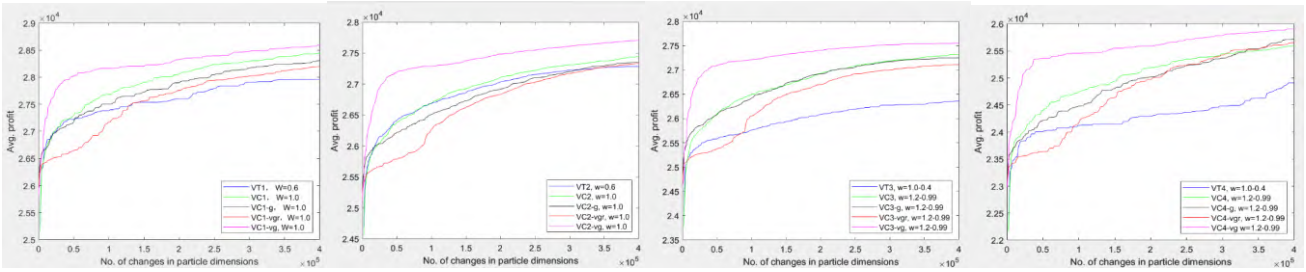
表 3　初始粒子位变数量对寻优结果的影响

**Tab. 3　Effect of initial particle variation on optimization results**

| *k* | 转换函数 1 | | 转换函数 2 | | 转换函数 3 | | 转换函数 4 | |
|---|---|---|---|---|---|---|---|---|
| | 全局最优解 | 百分比/% | 全局最优解 | 百分比/% | 全局最优解 | 百分比/% | 全局最优解 | 百分比/% |
| 2 | 27027.43 | 96.01 | 27285.77 | 97.49 | 27370.50 | 97.79 | 27378.47 | 97.82 |
| 5 | 27379.97 | 97.97 | 27288.60 | 97.73 | 27378.93 | 97.82 | 27341.17 | 97.69 |
| 10 | 27551.40 | 98.43 | 27435.20 | 98.02 | 27497.20 | 98.24 | 27439.97 | 98.04 |
| 20 | 27446.60 | 98.06 | 27302.40 | 97.55 | 27401.80 | 97.90 | 27402.57 | 97.90 |
| 25 | 27390.90 | 97.87 | 27334.70 | 97.66 | 27445.43 | 98.06 | 27371.13 | 97.79 |

确定好分级扩大方式后，在维度为 100、迭代次数为 250 的低维度背包问题中设计 5 种 BPSO 算法

在 4 种转换函数作用下的对比消融实验，结果如图 6 所示，其中横坐标为位变数量，纵坐标为全局最优解，横坐标以位变数量为 $4\times10^5$ 截图，相当于传统的 VT 算法在 200 次迭代时的位变数量，选取位变数量为横坐标更便于观察各个算法的性能和计算复杂度。粉色的线代表 VC-vg 算法，红色的线代表 VC-vgr 算法，绿色的线代表 VC 算法，黑色的线代表 VC-g 算法，蓝色的线代表传统的 VT 算法。




（a）转换函数 1　　（b）转换函数 2　　（c）转换函数 3　　（d）转换函数 4

图 6　低维度背包问题的转换函数

**Fig. 6　Transformation function for the low dimensional knapsack problem**

从图 6 可知，粉色的线在最上边且在位变为 $0.5\times10^5$ 时就与其余 4 条线有较大距离，纵坐标最优值已接近 $2.55\times10^4$，最接近全局最优解。这说明在 4 种转换函数作用下 VC-vg 算法都能以更快速度和较高的准确率接近全局最优解，其速度和准确率均优于其余 4 种 BPSO 算法。这是由于价值密度正序排序的作用是将解有规律的在解的空间上分布，同时也将少量的次优解聚集起来，使 BPSO 算法在全局搜索中更容易搜索到该区域，在进入局部搜索时寻优更高效，同时分级竞优策略也使搜索更充分，降低该区域被遗漏的风险。具体实验数据见表 4。

表 4 低维度背包实验结果

**Tab. 4　Low dimensional backpack experiment data**

| 转换函数 1 | 全局最优解 | 百分比/% | 转换函数 2 | 全局最优解 | 百分比/% | 转换函数 3 | 全局最优解 | 百分比/% | 转换函数 4 | 全局最优解 | 百分比/% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| VT1 | 27857.74 | 95.69 | VT2 | 27300.84 | 96.69 | VT3 | 26654.92 | 95.33 | VT4 | 25167.99 | 95.32 |
| VC1 | 28409.67 | 97.59 | VC2 | 27577.55 | 97.67 | VC3 | 27311.99 | 97.68 | VC4 | 25593.09 | 96.93 |
| VC1-g | 28322.34 | 97.30 | VC2-g | 27447.66 | 97.21 | VC3-g | 27258.86 | 97.45 | VC4-g | 225661.74 | 97.19 |
| VC1-vgr | 28261.21 | 97.09 | VC2-vgr | 27413.77 | 97.09 | VC3-vgr | 27183.37 | 97.22 | VC4-vgr | 25468.99 | 96.46 |
| VC1-vg | 28660.00 | 98.46 | VC2-vg | 27868.37 | 98.70 | VC3-vg | 27591.60 | 98.68 | VC4-vg | 25920.50 | 98.17 |

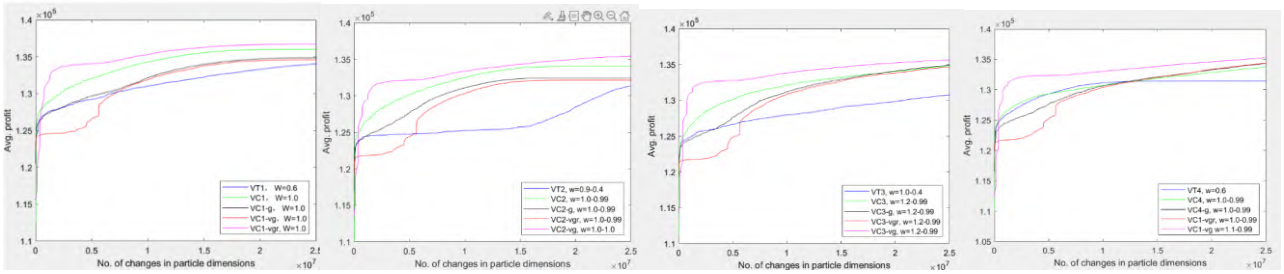
由表 4 可知，VC-vg 算法优于 VT 算法。VT 算法缺乏速度修正和排序策略，导致其在强相关背包问题中容易陷入局部最优，搜索效率较低；VC-vg 算法比 VT 算法准确率提升 1.5%~3.3%，充分说明速度修正和排序策略的重要性。同时，VC-vg 算法优于 VC 算法。VC 算法引入速度修正，但未采用价值密度排序和分级竞优策略，因此在搜索效率和准确率上略逊于 VC-vg 算法；VC-vg 算法比 VC 算法准确率提升约 1%，表明排序和分级竞优策略是算法性能的进一步提升的关键。VC-vg 算法优于 VC-g 算法和 VC-vgr 算法。VC-g 算法搜索顺序随机，无法有效利用价值密度信息，导致搜索效率低，而 VC-vgr 算法优先选择低价值密度的物品，导致背包容量利用率低，难以找到最优解。

综上所述，通过对比数据发现，经过速度修正的方法明显好于 VT 算法，而 VC-vg 算法的效果最好，在采用分级竞优的基础上，如果采用降序排序或不排序，则无法比单纯速度修正的 VC 算法取得更好的效果。由此可见，价值密度升序排序和分级竞优必须要联合使用才能在强相关背包中取得效果。

## 2.2　中维度背包实验设计与实验结果

在维度为 500 的中维度背包问题中，粒子群大小为 50，迭代次数为 1300，中维度背包问题的分级扩大方式选择 $k$=50，初始粒子位变数量为 50，迭代次数为 50 次，随后每迭代 50 次($a = 50$)增加 50 位变动，在迭代次数达到第 450 次后，粒子的全部维度都开始进行迭代寻优，中维度背包问题通过对参数 $w$ 的调整与低维度背包问题实验过程一致。图 7 为中维度背包在不同转换函数下的实验效果。由图 7 可知，粉色的线（VC-vg 算法）在最上方，其准确

度最高，而蓝色的线（VT 算法）最终都在最下方，寻优准确度最差。



（a）转换函数 1　（b）转换函数 2　（c）转换函数 3　（d）转换函数 4

图 7 中维度背包问题转换函数

**Fig. 7　Transformation function for the mid dimensional knapsack problem**

中维度背包实验数据见表 5。VC-vg 算法在 4 种转换函数的作用下百分比均在 98%以上。VC-vg 算法优于 VT 算法。4 种转换函数作用下的准确率均有所提高，这表明速度修正和排序策略在不同转换函数下均能显著提升算法性能。VC-vg 算法优于 VC 算法。VC 算法引入了速度修正，但未采用价值密度排序和分级竞优策略，因此在搜索效率和准确率上略逊于 VC-vg 算法，表明排序和分级竞优策略对算法性能的进一步提升具有重要作用。VC-vg 算法优于 VC-g 和 VC-vgr 算法。VC-g 算法和 VC-vgr 算法虽然引入了分级竞优策略，但价值密度排序是随机的、降序的，这导致在搜索最优解时效率低，因此在准确率上也低于 VC-vg 算法。

表 5 中维度背包实验数据

**Tab. 5　Mid dimensional backpack experimental data**

| 百分比/% | 全局最优解 | 百分比/% | 转换函数 2 | 全局最优解 | 百分比/% | 转换函数 3 | 全局最优解 | 百分比/% | 转换函数 4 | 全局最优解 | 百分比/% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| VT1 | 135303.05 | 97.10 | VT2 | 131598.20 | 95.70 | VT3 | 130718.30 | 95.06 | VT4 | 131424.90 | 95.57 |
| VC1 | 136009.70 | 97.60 | VC2 | 134034.15 | 97.47 | VC3 | 134919.30 | 98.11 | VC4 | 135133.45 | 98.27 |
| VC1-g | 134865.90 | 96.78 | VC2-g | 132440.75 | 96.31 | VC3-g | 134104.30 | 97.52 | VC4-g | 135096.05 | 98.24 |
| VC1-vgr | 134590.45 | 96.59 | VC2-vgr | 132136.45 | 96.09 | VC3-vgr | 133687.90 | 97.22 | VC4-vgr | 134803.10 | 98.02 |
| VC1-vg | 139554.15 | 98.16 | VC2-vg | 135733.85 | 98.70 | VC3-vg | 135665.30 | 98.68 | VC4-vg | 135676.05 | 98.67 |

综上所述，VC-vg 算法比 VT 算法的准确率提高 3%左右，比 VC 算法的百分比均高 0.5%左右，比 VC-g 的准确率提高 0.5%以上，比 VC-vgr 算法的准确率提高最少 0.5%以上，因此通过对比数据发现经过速度修正的几种方法明显好于 VT 算法，而 VC-vg 算法的效果最好，在采用分级竞优的基础上，如果采用降序排序或不排序，则无法比单纯速度修正的 VC 算法取得更好的效果。由此可见，进一步证明了价值密度升序排序和分级竞优必须要联合使用才能在强相关背包中进一步取得效果。

### 2.3　高维度背包实验设计与实验结果

在维度为 1000 的高位背包问题中，粒子群大小为 100，迭代次数为 2500，高维度背包问题的迭代打开方式为：初始位变数量为 100，迭代次数为 100 次，随后每迭代 100 次（$a = 100$），位变数量增加 100，在迭代次数达到第 900 次后，粒子的全部位都开始进行迭代寻优。图 8 为高维度背包在不同转换函数下的实验效果，高维度背包实验数据见表 6。

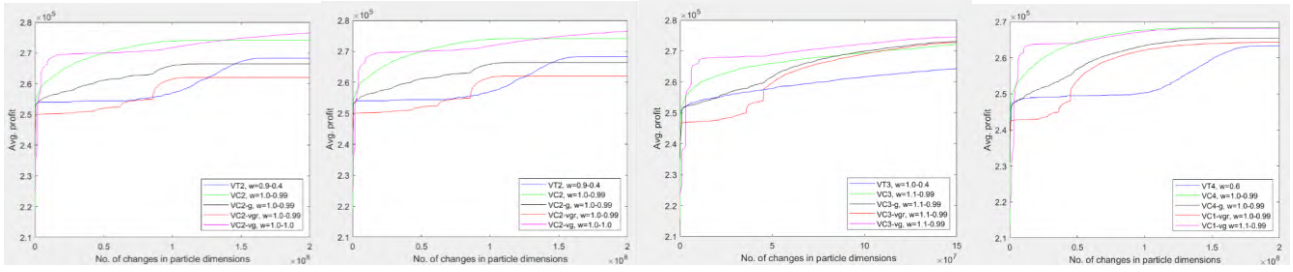


图 8 高维度背包问题转换函数

**Fig. 8　Transformation function for the high dimensional knapsack problem**

表 6　高维度背包实验数据

**Tab. 6　High dimensional backpack experimental data**

| 转换函数 1 | 全局最优解 | 百分比/% | 转换函数 2 | 全局最优解 | 百分比/% | 转换函数 3 | 全局最优解 | 百分比/% | 转换函数 4 | 全局最优解 | 百分比/% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| VT1 | 272719.80 | 96.69 | VT2 | 268346.20 | 95.59 | VT3 | 268309.00 | 96.05 | VT4 | 263320.80 | 95.78 |
| VC1 | 2272719.80 | 97.83 | VC2 | 274072.60 | 97.64 | VC3 | 272813.90 | 97.76 | VC4 | 268347.80 | 97.61 |
| VC1-g | 274452.80 | 97.31 | VC2-g | 266410.50 | 94.91 | VC3-g | 271397.90 | 97.28 | VC4-g | 265508.80 | 96.58 |
| VC1-vgr | 274394.20 | 97.29 | VC2-vgr | 262023.10 | 95.10 | VC3-vgr | 271118.00 | 97.18 | VC4-vgr | 264257.60 | 96.12 |
| VC1-vg | 276997.30 | 98.21 | VC2-vg | 276819.20 | 98.82 | VC3-vg | 275479.30 | 98.61 | VC4-vg | 268210.30 | 97.56 |

由图 8 可知，粉色的线 VC-vg 算法的寻优结果明显优于其余 4 种算法。结合表 6 数据可知，VC-vg 算法的寻优效果最好，准确度最高为 98.82%，比其余 4 种算法高 1%~3.5%。在转换函数 4 中，VC-vg 的效果虽然明显优于 VT 算法、VC-g 算法和 VC4-vgr 算法，但是却不如 VC 算法的寻优效果好。因此，对于高维度背包问题来说，在转换函数 1、2、3 中 VC-vg 算法的寻优能力要明显高于其余 4 种算法，但是在转换函数 4 中，VC 算法要略优于 VC-vg 算法。

## 3　结　论

本文聚焦于强相关背包问题的求解，深入探讨了 BPSO 算法在该问题中的优势和不足，针对强相关背包问题中最优解和次优解在空间分布上的过度集中问题，对 BPSO 算法进行了针对性的改进，引入基于价值密度的分级竞优策略，使算法在搜索过程中能够更好地平衡全局搜索和局部搜索能力，提高优势解狭小聚集空间的搜索概率，降低陷入局部最优的可能性。实验结果表明，改进后的 BPSO 算法在解决强相关背包问题上具有显著优势。后续将继续深化对 BPSO 算法的研究，探索更多针对强相关背包问题的优化策略。同时，也将关注 BPSO 算法在实际问题中的应用实践，推动其在更多领域的广泛应用和发展。

**参考文献：**


[1] LUO C, WANG S, LI T, et al. Large-scale meta-heuristic feature selection based on BPSO assisted rough hypercuboid approach[J]. IEEE Transactions on neural networks and learning systems, 2022, 34(12): 10889-10903.

[2] TANI L, VEELKEN C. Comparison of Bayesian and particle swarm algorithms for hyperparameter optimisation in machine learning applications in high energy physics[J]. Computer physics communications, 2024, 294:108955.

[3] TMAMNA J, FOURATI R, AYED E B, et al. A binary particle swarm optimization-based pruning approach for environmentally sustainable and robust CNNs[J]. Neurocomputing, 2024, 608: 128378.

[4] 袁剑锋. 基于改进二进制粒子群优化的特征选择[J]. 新型工业化,2020,10(7):21-22.

[5] 姜磊,刘建华,张冬阳,等.二进制粒子群算法中 V 型转换函数的应用分析[J].计算机应用与软件，2021, 38(4):263-270.

[6] NGUYEN B H, XUE B, ANDREAE P, et al. A new binary particle swarm optimization approach: momentum and dynamic balance between exploration and exploitation[J]. IEEE Transactions on cybernetics, 2019, 51(2): 589-603.

[7] PASHAEI E，PASHAEI E，AYDIN N. Gene selection using hybrid binary black hole algorithm and modified binary particle swarm optimization[J]. Genomics，2019，111(4): 669-686.

[8] BHARTI K K, SINGH P K. Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering[J]. Applied soft computing, 2016，43:20-34.

[9] ALJEHANI S, ALOTAIBI Y. Preserving privacy in association rule mining using multi-threshold particle swarm optimization[J]. Information sciences, 2025, 692:121673.

[10] LU D, LI W, ZHANG L, et al. Multi-objective optimization and reconstruction of distribution networks with distributed power sources based on an improved BPSO algorithm[J]. Energies, 2024, 17:4877.

[11] ZHAO Q, ZHANG C K, LI H, et al. An error discovery and correction for the family of V-shaped BPSO algorithms. [EB/OL]. [2024-06-01]. https://arxiv.org/pdf/2407.17889.