A Study on the Greedy Repair Strategy for Particle Swarm Optimization in the Discounted {0-1} Knapsack Problem (Postprint)

Authors: Dai Zuhua, Zhou Bin, Long Yujing, Wang Zongquan

Date: 2022-04-07T15:01:57+00:00

Abstract

When swarm intelligence heuristic algorithms solve the Discounted {0-1} Knapsack Problem (D{0-1}KP), some repair and optimization strategy is required to transform abnormal encoding individuals into encoding individuals that satisfy the solution constraints, in order to improve solution efficiency and quality. Based on the concept of item set value density, and taking the Particle Swarm Optimization (PSO) algorithm as an example, a set of item set-based greedy repair and optimization methods (Group Greedy Repair and Optimization Algorithm, GGROA) is proposed, and the PSO-GGRDKP algorithm (PSO based GGROA for solving D{0-1}KP) is further constructed to investigate the feasibility and performance of the GGROA method. PSO-NGROADKP (PSO based NGROA for solving D{0-1}KP) and PSO-GRDKP (PSO based GROA for solving D{0-1}KP) are particle swarm algorithms based on the item greedy repair and optimization method. Experimental results on the D{0-1}KP standard dataset demonstrate that, compared with PSO-NGROADKP and PSO-GRDKP, the PSO-GGRDKP algorithm has a slightly higher solution error rate, but its time performance is improved by 13.8% and 12.9%, respectively.

Full Text

Preamble

Greedy Repair Strategy for Particle Swarm Optimization in Discounted {0-1} Knapsack Problem

Dai Zuhua, Zhou Bin, Long Yujing, Wang Zongquan
(College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China)

Abstract: When swarm intelligence heuristic algorithms solve the Discounted {0-1} Knapsack Problem (D{0-1}KP), repair and optimization strategies are required to convert abnormal coding individuals into feasible solutions that satisfy constraint conditions, thereby improving solution efficiency and quality. Building upon the concept of group value density, this paper proposes a set of group-based greedy repair and optimization methods (GGROA) using the Particle Swarm Optimization (PSO) algorithm as a case study. We further construct the PSO-GGRDKP algorithm (PSO based GGROA for solving D{0-1}KP) to investigate the feasibility and performance of the GGROA approach.

PSO-NGROADKP (PSO based NGROA for solving D{0-1}KP) and PSO-GRDKP (PSO based GROA for solving D{0-1}KP) are PSO algorithms based on item-level greedy repair and optimization methods. Experimental results on standard D{0-1}KP datasets demonstrate that compared with PSO-NGROADKP and PSO-GRDKP, the PSO-GGRDKP algorithm exhibits slightly higher solution error rates but achieves time performance improvements of 13.8% and 12.9%, respectively.

Keywords: Discounted {0-1} Knapsack Problem; heuristic algorithm; particle swarm optimization algorithm; abnormal coding individual; greedy repair and optimization; D{0-1}KP dataset

0 Introduction

The {0-1} Knapsack Problem ({0-1}KP) is a fundamental NP-complete problem in computer science and a classic combinatorial optimization problem with widespread applications in commerce, economics, management, and security. Over the decades since its introduction, {0-1}KP has been extensively studied, giving rise to two major algorithmic categories: exact algorithms and approximation algorithms. Exact algorithms include dynamic programming, backtracking, and branch-and-bound methods, while approximation algorithms primarily encompass randomized algorithms, approximation schemes, and bio-inspired algorithms. In 2005, Guder first proposed the Discounted {0-1} Knapsack Problem (D{0-1}KP) as an extension of {0-1}KP. This problem serves as a classical mathematical model for characterizing commercial phenomena such as discount sales and bundled promotions, providing scientific computational foundations for merchants to design marketing strategies and for consumers to make purchasing decisions.

A D{0-1}KP instance consists of a set of item groups, with each group containing three items available for selection. Definition 1 presents the first mathematical model of D{0-1}KP.

Definition 1. First Mathematical Model of D{0-1}KP [3]: Given n item groups and a knapsack with capacity C, each group I_i comprises three items. Let w_{3i+j} and p_{3i+j} denote the weight and value coefficients of item j in group

i, respectively, where $j \in \{0, 1, 2\}$. The value density of each item is defined as $e_{3i+j} = p_{3i+j}/w_{3i+j}$. At most one item can be selected from each group for inclusion in the knapsack. Under the constraint that the total weight does not exceed capacity C, the objective is to select items from the given groups to maximize the sum of their value coefficients.

All parameters are positive integers, with $i \in \{0, 1, \ldots, n-1\}$. D{0-1}KP is a special integer programming problem. Define decision variables $x_{3i+j} \in \{0, 1\}$, where $x_{3i+j} = 1$ indicates that item j in group i is included in the knapsack, and $x_{3i+j} = 0$ otherwise. For a decision vector $\mathbf{x} = (x_0, x_1, \ldots, x_{3n-1})$, the integer programming formulation of D{0-1}KP is:

$$\max \quad f(\mathbf{x}) = \sum_{i=0}^{n-1} (p_{3i}x_{3i} + p_{3i+1}x_{3i+1} + p_{3i+2}x_{3i+2}) \tag{1}$$

$$\text{s.t.} \quad \sum_{i=0}^{n-1} (w_{3i}x_{3i} + w_{3i+1}x_{3i+1} + w_{3i+2}x_{3i+2}) \leq C \tag{2}$$

$$x_{3i} + x_{3i+1} + x_{3i+2} \leq 1, \quad i = 0, 1, \ldots, n-1 \tag{3}$$

$$x_{3i+j} \in \{0, 1\}, \quad i = 0, 1, \ldots, n-1, \ j = 0, 1, 2 \tag{4}$$

Each item group in D{0-1}KP has four possible selection states. When the knapsack capacity and item value coefficients have large ranges, determining the constraint conditions for not selecting a particular item becomes challenging, making D{0-1}KP algorithmically more difficult than {0-1}KP. Building upon {0-1}KP research, scholars have investigated various algorithms for D{0-1}KP. The Basic Dynamic Programming (BDP) algorithm [5] provides exact solutions for small-scale instances. In 2016, He Yichao et al. [4] constructed a dynamic programming objective function based on the principle of "minimizing total weight for a given sum of selected item values," proposing a New Exact algorithm for D{0-1}KP (NE-DKP). When the knapsack capacity exceeds the cumulative sum of the third-item values across all groups, NE-DKP outperforms BDP. However, dynamic programming algorithms have pseudo-polynomial time complexity and are generally unsuitable for large-scale D{0-1}KP instances.

Swarm intelligence heuristic algorithms have become mainstream approaches for solving D{0-1}KP in recent years, demonstrating outstanding performance advantages for large-scale instances. When designing heuristic algorithms using the first mathematical model of D{0-1}KP, decision vectors $\mathbf{x}$ with different value combinations correspond to different individuals. Although this binary encoding facilitates the implementation of evolutionary operators, the probability of abnormal coding individuals (i.e., individuals whose encodings do not correspond to feasible solutions) in the search space is at least $1 - (4/7)^n$ due to constraints (2) and (3). Consequently, repair and optimization strategies are necessary to convert abnormal coding individuals into feasible solutions, thereby improving solution efficiency and quality. Michalewicz [6], He Yichao [3,4], and other scholars have proposed greedy repair and optimization strategies that effectively eliminate abnormal coding individuals. In their study of D{0-1}KP particle swarm algorithms, He Yichao et al. [4] proved that only four possible relationships exist among the value-to-weight ratios of three items within a given group. Leveraging this property, they proposed the GR-DKP (Greedy Repair Algorithm for D{0-1}KP) and designed the PSO-GRDKP algorithm (PSO based Greedy Repair Algorithm for solving D{0-1}KP) [4]. In their research on the First Genetic Algorithm (FirEGA) for D{0-1}KP [3], they also proposed the GROA (Greedy Repair and Optimization Algorithm). Both GROA and GR-DKP select items greedily based on non-increasing item value density, choosing the item with maximum value density when multiple items in a group are selected. Yang Yang et al. further optimized GROA, constructing the NGROA (New Greedy Repair and Optimization Algorithm) [7], which also selects items based on non-increasing item value density but chooses the item with maximum value in the group when multiple items are selected. Their research applying NGROA to NFirEGA (New First Genetic Algorithm) demonstrated improved solution quality for D{0-1}KP. More recently, literature [8] defined the concept of group value density and proposed a greedy selection strategy based on non-increasing group value density, selecting the item with maximum value density that satisfies solution constraints when multiple items in a group are chosen.

The time complexity of the aforementioned greedy repair and optimization algorithms [3-8] is $O(n \log n)$. Most researchers have adopted the GROA algorithm from [3] to design various heuristic evolutionary algorithms for D{0-1}KP, including differential evolution (DE) [9], Mutated Double Codes Binary Bat Algorithm (MDBBA) [10], Monarch Butterfly Optimization with Differential Evolution (DEMBO) [11], Moth Search Algorithm (MS) [12], Lagrange Interpolation based Learning Monkey Algorithm (LSTMA) [13], Ring Theory Based Evolutionary Algorithm (RTEA) [14], and Discrete Hybrid Teaching-Learning based Optimization Algorithm (HTLBO) [15]. These studies typically use FirEGA [3] and basic heuristic algorithms as baselines, comparing solution quality and convergence speed to demonstrate algorithmic performance.

In research employing the first mathematical model and GROA algorithm for D{0-1}KP heuristics [10,11,14], experimental results in [10] show that on UDKP and SDKP instances, the Double Codes Binary Bat Algorithm (DBBA) exhibits worse solution quality than FirEGA, while on IDKP and WDKP instances, DBBA performs better. The Monarch Butterfly Optimization (MBO) algorithm yields significantly worse results than FirEGA across best, mean, and worst evaluation metrics [11]. The PSO-GRDKP algorithm demonstrates superior solution accuracy and stability compared to FirEGA [14]. These findings indicate that different heuristic algorithms employing the same class of greedy repair and optimization methods exhibit varying performance. Based on the group value density concept [8], this paper constructs a set of group-based greedy repair and optimization methods and investigates their feasibility and performance, further discussing the adaptability relationship between greedy repair methods and data instance types. To avoid algorithmic differences confounding the research, we use the particle swarm algorithm as a case study to construct D{0-1}KP solution algorithms.

1.1 Binary Particle Swarm Optimization Algorithm

Kennedy and Eberhart proposed the standard Particle Swarm Optimization (PSO) algorithm in 1995 through their study of bird flock foraging behavior [16]. Binary PSO (BPSO) is a discrete-space variant for binary optimization problems [17]. Bansal and Deep employed particle velocity as the probability of selecting an item as 1 or 0 in {0-1}KP, proposing a Modified Binary PSO (MBPSO) algorithm for knapsack problems [18]. The MBPSO algorithm operates as follows: assume a D-dimensional search space, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ represents the position of the i-th particle in the swarm, $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ denotes its personal best position, and $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ is the corresponding velocity vector. Let $\mathbf{p}_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ represent the global best solution found by the swarm during evolution. Personal and global bests are determined by the fitness function. The particle evolution equations are:

$$v_{ij}^{t+1} = v_{ij}^t + c_1 r_1 (p_{ij} - x_{ij}^t) + c_2 r_2 (p_{gj} - x_{ij}^t) \quad (4)$$

$$\text{sig}(v_{ij}^{t+1}) = \frac{1}{1 + e^{-v_{ij}{}^{t+1}}} \quad (5)$$

$$x_{ij}^{t+1} = \begin{cases} 1, & \text{if } r_3 < \text{sig}(v_{ij}^{t+1}) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where t indicates the iteration number, v_{ij}^t is the velocity of particle i in dimension j at iteration t, and x_{ij}^t is the corresponding position. Equation (4) constitutes the evolutionary dynamics of the swarm, comprising three components: v_{ij}^t represents the inertial potential from the particle's history; $c_1 r_1 (p_{ij} - x_{ij}^t)$ is the attractive potential from the particle's personal best, representing "individual cognition"; and $c_2 r_2 (p_{gj} - x_{ij}^t)$ is the attractive potential from the swarm's global best, representing "social cognition" [19]. The learning factors c_1 and c_2 serve as weighting coefficients for individual and social cognition, influencing the degree of learning from personal and social information during evolution and affecting the particle's ability to identify optimization objectives. The parameters r_1, r_2, r_3 are random numbers in $(0, 1)$.

The PSO-GRDKP algorithm [4] is a PSO-based solver for D{0-1}KP with search space dimension $D = 3n$. The fitness function corresponds to the total value of items in the knapsack, with learning factors $c_1 = c_2 = 2$ and time complexity $O(nT)$, where T is the number of iterations. This algorithm employs a greedy repair and optimization operator to correct and optimize infeasible particles in the population, effectively increasing the proportion of high-quality individuals and enhancing the algorithm's search capability.

1.2 Greedy Repair and Optimization Algorithm for Infeasible Individuals

He Yichao et al. proposed two greedy repair algorithms for D{0-1}KP: GROA [3] and GR-DKP [4]. GR-DKP was specifically developed for PSO, and its pseudocode is presented in Algorithm 1.

Algorithm 1: GR-DKP
Input: Value vector $\mathbf{p} = [p_0, \ldots, p_{3n-1}]$, weight vector $\mathbf{w} = [w_0, \ldots, w_{3n-1}]$; knapsack capacity C; array H storing item indices sorted by non-increasing value density; particle $\mathbf{x}$ to be repaired.
Output: Repaired feasible particle $\mathbf{x}$ and fitness value $f(\mathbf{x})$.

The algorithm proceeds with initialization, repair operations (steps 2-4), and optimization (step 5), with overall time complexity $O(n \log n)$. If step 2 of Algorithm 1 is modified to item $= \arg\max_{j\in\{0,1,2\}}\{p_{3i+j}\}$, the algorithm evolves into NGROA [7].

2.1 Group-Based Greedy Strategy for Particle Repair and Optimization

Existing greedy repair strategies for infeasible particles in D{0-1}KP [3,4,7] sort items by value density during preprocessing. This paper introduces the group value density R_i and sorts data in non-increasing order of R_i at the group level before selecting items. Literature [8] presents three algorithms based on this concept.

Algorithm 2 performs repair operations (steps 2-5) and optimization (step 6) on particle $\mathbf{x}$, with time complexity $O(n \log n)$.

2.2 D{0-1}KP Particle Swarm Optimization Algorithm Based on GGROA

In constructing the PSO-GRDKP algorithm [4], the authors employed the basic discrete particle evolution equations (4)-(6) with learning factors $c_1 = c_2 = 2$. To avoid confounding effects from evolutionary formulas and parameters and to facilitate comparative analysis under uniform conditions, we adopt the same scheme to construct PSO-GGRDKP (PSO based GGROA for solving D{0-1}KP), with pseudocode in Algorithm 3. The parameters m and z correspond to the group value density calculation functions and item selection strategies in Algorithm 2.

Algorithm 3: PSO-GGRDKP
Input: Value vector $\mathbf{p}$, weight vector $\mathbf{w}$, knapsack capacity C, population size N, maximum iterations T, learning constants c_1, c_2, GGROA parameters m, z.
Output: Best particle $\mathbf{g}_{best}$ and approximate optimal solution value fitness($\mathbf{g}_{best}$).

The algorithm initializes the population, then iteratively updates velocities and positions using equations (4)-(6), applying the GGROA repair operator to infeasible particles. The time complexity is $O(nT + n \log n)$.

It is evident that when $m = 1$, the algorithm is equivalent to the GR-DKP [4] strategy. Combining NGROA concepts, we define eight D{0-1}KP group-based greedy repair and optimization methods (GGROA) by pairing four group value density calculations (equations (7)-(10)) with two item selection strategies (equations (11)-(12)), as shown in Table 1.

Table 1. GGROA Strategies for D{0-1}KP in PSO Algorithm

Group Value Density	Item Selection Strategy	GGROA
$R_i^{(1)}$	Strategy 1	(1,1)
$R_i^{(1)}$	Strategy 2	(1,2)
$R_i^{(2)}$	Strategy 1	(2,1)
$R_i^{(2)}$	Strategy 2	(2,2)
$R_i^{(3)}$	Strategy 1	(3,1)
$R_i^{(3)}$	Strategy 2	(3,2)
$R_i^{(4)}$	Strategy 1	(4,1)
$R_i^{(4)}$	Strategy 2	(4,2)

The pseudocode for GGROA is presented in Algorithm 2, where parameter $m \in \{1, 2, 3, 4\}$ corresponds to the four group value density functions (equations (7)-(10)), and parameter $z \in \{1, 2\}$ corresponds to the two item selection strategies (equations (11)-(12)). The time complexity of PSO-GGRDKP is $O(nT+n \log n)$.

3.1 Experimental Design

The D{0-1}KP dataset serves as a standard benchmark for evaluating algorithmic performance [3], comprising four instance types: Inverse Strongly Correlated D{0-1}KP (IDKP), Strongly Correlated D{0-1}KP (SDKP), Weakly Correlated D{0-1}KP (WDKP), and Uncorrelated D{0-1}KP (UDKP). Each type includes instances with scales $n \in \{100, 300, 500, 800, 1000\}$. To validate the performance characteristics of various greedy repair strategies for D{0-1}KP, we conduct two experiments. Experiment 1 examines eight group-based greedy repair strategies using PSO to investigate their feasibility, performance, and adaptability to different data instance types. Experiment 2 evaluates the performance characteristics of three typical infeasible individual repair methods: PSO-GRDKP [4], PSO-NGROADKP, and PSO-GGRDKP. Both experiments analyze algorithmic performance through solution time and computational results. All experiments run on a ThinkStation P330 with an Intel® Core™ i7-8700 CPU (3.2 GHz), 16 GB DDR4 memory, NVIDIA P2200 GPU, Microsoft Windows 10 Education Edition, and Python 3.6.

Let opt denote the optimal solution obtained via basic dynamic programming for each D{0-1}KP instance. The PSO algorithm uses a population size of 200, maximum iterations equal to the number of item groups n, and runs independently 20 times per instance. We report the maximum (best), mean, and minimum (worst) approximate solution values, along with average computation time T.

3.2 Experiment 1: Data and Analysis

Table 2 lists the IDs corresponding to the eight group-based greedy repair strategies.

Table 2. GGROA Strategies and Their IDs for D{0-1}KP

ID	Strategy
1	(1,1)
2	(1,2)
3	(2,1)
4	(2,2)
5	(3,1)
6	(3,2)
7	(4,1)
8	(4,2)

Table 3 presents experimental results for the eight greedy repair methods on four instance types with $n = 900$, where subscripts in PSO-GGRDKP names correspond to Table 2 IDs. Box plots for eight instances (IDKP3, IDKP6, SDKP3, SDKP6, UDKP3, UDKP6, WDKP3, WDKP6) are shown in Figure 1.

Table 3. PSO-GGRDKP Experimental Results on Standard D{0-1}KP Instances

Instance (Optimal)	GGRDKP1	GGRDKP2	GGRDKP3	GGRDKP4	GGRDKP5	GGRDKP6	GGRDKP7	GGRDKP8
IDKP3 (234804)	…	…	…	…	…	…	…	…
IDKP6 (452463)	…	…	…	…	…	…	…	…
SDKP3 (238248)	…	…	…	…	…	…	…	…
SDKP6 (466097)	…	…	…	…	…	…	…	…
WDKP3 (256616)	…	…	…	…	…	…	…	…

Instance (Optimal)	GGRDKP1	GGRDKP2	GGRDKP3	GGRDKP4	GGRDKP5	GGRDKP6	GGRDKP7	GGRDKP8
WDKP6 (466050)	...	...	...	...	...	...	...	...
UDKP3 (184006)	...	...	...	...	...	...	...	...
UDKP6 (536578)	...	...	...	...	...	...	...	...

Results demonstrate that all eight group-based greedy repair strategies are feasible for PSO, though significant performance differences exist across strategies on the same instance type, and the same strategy performs differently across instance types. For each data type, we identify the top three strategies by mean solution quality: IDKP favors PSO-GGRDKP1 and PSO-GGRDKP2; SDKP favors PSO-GGRDKP1 and PSO-GGRDKP5; WDKP favors PSO-GGRDKP1 and PSO-GGRDKP8; UDKP favors PSO-GGRDKP2 and PSO-GGRDKP6. Considering average computation time T, we select the most time-efficient algorithms: PSO-GGRDKP2 for IDKP, PSO-GGRDKP5 for SDKP, PSO-GGRDKP8 for WDKP, and PSO-GGRDKP2 for UDKP.

3.3 Experiment 2: Data and Analysis

Table 4 presents experimental data for PSO-GRDKP, PSO-NGROADKP, and PSO-GGRDKP. For PSO-GGRDKP, we select the best-performing strategies identified in Experiment 1 for each data type.

Table 4. Experimental Results of PSO-GGRDKP, PSO-NGROADKP, and PSO-GRDKP

Dataset	PSO-GGRDKP	PSO-NGROADKP	PSO-GRDKP
	mean	worst	mean
IDKP1-10	...	...	...
SDKP1-10	...	...	...
WDKP1-10	...	...	...
UDKP1-10	...	...	...

The three greedy repair strategies exhibit varying solution quality and time performance. Define the solution error rate as $\mathrm{ERR} = 1 - \mathrm{mean}/\mathrm{opt}$. The average error rate across all instances, $\mathrm{ERR}_{\mathrm{AVE}}$, indicates algorithm performance (lower values are better). Table 5 summarizes the average error rates and computation times for the three algorithms across four data types.

Table 5. Average Error Rates and Computation Times of Three PSO Algorithms

Algorithm	IDKP ERR_{AVE}	IDKP T_{AVE}	SDKP ERR_{AVE}	SDKP T_{AVE}	WDKP ERR_{AVE}	WDKP T_{AVE}	UDKP ERR_{AVE}	UDKP T_{AVE}
PSO-GGRDKP	…	…	…	…	…	…	…	…
PSO-NGROADKP	…	…	…	…	…	…	…	…
PSO-GRDKP	…	…	…	…	…	…	…	…

Results show consistent trends across instance types: IDKP yields the lowest error rates, while UDKP produces the highest. Among the three algorithms, PSO-NGROADKP achieves the lowest average error rate (0.2%), while PSO-GGRDKP's average error rate is slightly higher at 0.7% compared to PSO-NGROADKP and 0.4% compared to PSO-GRDKP. However, PSO-GGRDKP significantly outperforms both algorithms in time performance, achieving 12.9% improvement over PSO-GRDKP and 13.8% over PSO-NGROADKP. The time performance trends vary across instance types: PSO-GGRDKP performs best on WDKP, PSO-NGROADKP on SDKP, and PSO-GRDKP on IDKP.

Further investigation reveals that solution error rates correlate strongly with data characteristics. Table 6 presents statistical measures for each instance type: correlation coefficient ρ between item values and weights, mean item value density μ, and standard deviation σ.

Table 6. Statistical Values of D{0-1}KP Instances

Dataset	ρ	μ	σ
IDKP	…	…	…
SDKP	…	…	…
WDKP	…	…	…
UDKP	…	…	…

Analysis reveals that except for SDKP, the error rates across IDKP, WDKP, and UDKP correlate with ρ—larger correlation coefficients yield smaller error rates. Interestingly, SDKP's correlation coefficient is slightly smaller than IDKP's but larger than WDKP's, yet its error rate exceeds WDKP's. Moreover, SDKP and UDKP show large differences in ρ but similar error rates. Table 6 shows the ordering $\mu_{IDKP} < \mu_{WDKP} < \mu_{SDKP} < \mu_{UDKP}$ and $\sigma_{IDKP} < \sigma_{WDKP} < \sigma_{SDKP} < \sigma_{UDKP}$, which aligns with the observed error rate trends. This indicates that PSO performance with greedy repair strategies for D{0-1}KP is highly correlated with data characteristics such as mean and standard deviation of item value density, explaining the similar performance on SDKP and UDKP.

4 Conclusion

This paper defines the group value density concept for D{0-1}KP and constructs the Group Greedy Repair and Optimization Algorithm (GGROA) with eight combined strategies. Using PSO as a case study, we develop the PSO-GGRDKP algorithm to investigate GGROA's feasibility and performance for solving D{0-1}KP. Experimental results on four D{0-1}KP instance types demonstrate:

1. The optimal GGROA operators for each instance type are: IDKP—PSO-GGRDKP2, SDKP—PSO-GGRDKP5, WDKP—PSO-GGRDKP8, and UDKP—PSO-GGRDKP2.
2. Compared with PSO-NGROADKP and PSO-GRDKP, PSO-GGRDKP exhibits average error rates 0.7% and 0.4% higher, respectively, while improving time performance by 13.8% and 12.9%.
3. The error rates of all three algorithms correlate strongly with data characteristics including correlation coefficients, mean value density, and value density standard deviation.

These results confirm that PSO-GGRDKP significantly improves algorithmic time efficiency for D{0-1}KP at the cost of slightly higher solution error rates. Inspired by research showing performance variations when applying GROA across different heuristic algorithms, we infer that GGROA's performance may also vary across algorithmic frameworks, which warrants future investigation. Additionally, recent work [20] proposed a group-based encoding scheme for D{0-1}KP in PSO, suggesting a promising direction for further PSO-GGRDKP enhancement.

References

[1] Guder J. Discounted knapsack problems for pairs of items [D]. Nuremberg: University of Erlangen-Nurnberg, 2005.

[2] Guldan B. Heuristic and exact algorithms for discounted knapsack problems [D]. Nuremberg: University of Erlangen-Nuremberg, 2007: 1-10.

[3] He Yichao, Wang Xizhao, Li Wenbin, et al. Research on genetic algorithms for discounted{0-1}knapsack problem [J]. China Journal of Computer, 2016, 39(12): 2614-2630.

[4] He Y C, Wang X Z, He Y L, et al. Exact and approximate algorithms for discounted{0-1}knapsack problem [J]. Information Sciences, 2016, 369(11): 634-647.

[5] Rong A, Figueira J R, Klamroth K. Dynamic programming based algorithms for the discounted{0-1}knapsack problem [J]. Applied Mathematics & Computation, 2012, 218(12): 6921–6933.

[6] Michalewicz Z, Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems [J]. Evolutionary Computation, 1996, 4(1): 1-32.

[7] Yang Yang, Pan Dazhi, He Yichao. Improved repair strategy genetic algorithm solve discount{0-1}knapsack problem [J]. Computer Engineering and Applications, 2018, 54(21): 37-42.

[8] Wilbaut C, Todosijevic R, H Anafi S, et al. Heuristic and exact fixation-based approaches for the discounted 0-1 knapsack problem [J]. arXiv preprint, 2021, arXiv: 2106.03438.

[9] Zhu H, He Y C, Wang X Z, et al. Discrete differential evolutions for the discounted{0–1}knapsack problem [J]. International Journal of Bio-Inspired Computation, 2017, 10(4): 219-238.

[10] Wu Congcong, He Yichao, Chen Yiying, et al. Mutated bat algorithm for solving discounted{0-1}knapsack problem [J]. Journal of Computer Applications, 2017, 37(005): 1292-1299.

[11] Feng Yanhong, Yang Juan, He Yichao, et al. Monarch butterfly optimization algorithm with differential evolution for the discounted{0-1}knapsack problem [J]. Acta Electronica Sinica, 2018, 46(6): 1343-1350.

[12] Feng Y, Wang G G. Binary moth search algorithm for discounted{0-1}knapsack problem [J]. IEEE Access, 2018, 6(99): 10708-10719.

[13] Xu Xiaoping, Xu Li, Wang Feng, Liu Long. Learning monkey algorithm based on Lagrange interpolation to solve discounted{0-1}knapsack problem [J]. Journal of Computer Application, 2020, 40(11): 19-24.

[14] He Y C, Wang X Z, Gao S G. Ring theory-based evolutionary algorithm and its application to D{0-1}KP [J]. Applied Soft Computing, 2019, 77(4): 714-722.

[15] Wu C C, Zhao J L, Feng Y H, et al. Solving discounted{0-1}knapsack problems by a discrete hybrid teaching-learning-based optimization algorithm [J]. Applied Intelligence, 2020, 50(12): 1872-1888.

[16] Kennedy J, Eberhart R C. Particle swarm optimization [C]//Proceedings of IEEE International Conference On Neural Networks. Perth, Australia: IEEE, 1995: 1942-1948.

[17] Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm [C]//IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, 12-15 Oct. 1997. IEEE, 1997, 5(1): 4104–4108.

[18] Bansal J C, Deep K. A modified binary particle swarm optimization for Knapsack problems [J]. Applied Mathematics and Computation, 2012, 218(22): 11042-11061.

[19] Ma Rongyong, Yang Leilei, Zhang Zhichao. Analysis the Characteristic of C1, C2 based on the PSO of Iterative Shift and Trajectory of Particle [J]. Mathematical Computation, 2013, 2(4): 109-115.

[20] Truong T K. Different transfer functions for binary particle swarm optimization with a new encoding scheme for discounted{0-1}knapsack problem [J]. Mathematical Problems in Engineering, 2021, 2021(4): 1-15.

Note: Figure translations are in progress. See original paper for figures.