probability collectives algorithm for solving knapsack problem

Mohamed Alhamrouni(1)
maalhamroni@elmergib.edu.ly

Lutfia khalifa(2)
lkalhaj@elmergib.edu.ly

(1)(2) Computer Science Dept. Faculty of Arts & Science Kasr Khiar, Elmergib University-Libya

الملخص:

تعد مشكلات التحسين التوافقي شائعة جدًا في مختلف المجالات. حيث تعمل على إيجاد أفضل حل ممكن من مجموعة محدودة من الكائنات. ببساطة، إنها عملية تحديد الحلول المثلى من مجموعة من مجموعات البيانات المتاحة لمشكلة معينة. تعتبر مشكلة الحقيبة (*KP*) مشكلة مألوفة ومدروسة في التحسين التوافقي، حيث يتم استخدامها لنمذجة المواقف الصناعية أو القرارات المالية. في هذا البحث، قمنا بتطبيق منهج الخوارزمية الجماعية الاحتمالية (*PCA*) لحل مسائل الحقيبة (*KP*)، وقد حققت أداءً عاليًا.

الكلمات المفتاحية: مشكلة الحقيبة، التحسين التوافقي، الخوارزمية الجماعية الاحتمالية.

Abstract:

Combinatorial optimization problems are very common in various fields. It involves finding the best possible solution from a finite set of objects. Simply put, it is the process of identifying the optimal solutions from a set of available data sets for a particular problem. The knapsack problem (KP) is considered a familiar and thoughtful problem in combinatorial optimization, where it is used to model industrial situations or financial decisions. In this paper, we implement the approach of a Probability collective algorithm (PCA) for solving the knapsack problems (KP), which has achieved high performance.

Keywords: knapsack problem, Combinatorial optimization, Probability collective algorithm, NP-Complete.

Introduction:

In the framework of Collective Intelligence (COIN), the Artificial Intelligence (AI) tool indicated as Probability Collectives (PC) is becoming common for modeling and controlling distributed Multi-Agent Systems (MAS) and also deep connections to Game Theory, Statistical Physics, and Optimization[1]. The PC theory first proposed by David Wolpert in 1999, considers an effectual method of sampling the probability space, changing the problem into the convex space of distribution. PC allocates probability values to each agent's moves, in contrast to stochastic approaches, For example, Genetic Algorithms (GA), Swarm Optimization and Simulated Annealing (SA), instead of deciding over the agent's moves/set of actions [15]. Every agent selects a particular action dependent on its strategy set having the highest probability and then updates its probability distribution in each iteration, which outcomes in optimizing the world utility or system objective that depends on the prior knowledge of the actions/strategies of all other agents. Thus, the process continues to find the best solution until the convergence reaches the global solution or one of the stopping criteria[2].

The most popular form of KP is the single constraint binary variant, in which we are given N items, each with a profit pi and a weight wi, with i = 1,..., N, and a knapsack capacity C. The problem is to select a subset of objects such that their weight does not exceed C while returning the highest total profit [3].

Literature Review:

Amol C. Adamuthe and his team utilized the harmony search (HS) algorithm to address both single and multi-objective knapsack issues[4]. They conducted experiments on 43 problem instances, taken from three different datasets, and were able to generate notably better results.

Their study is a significant contribution to the field of optimization since the knapsack problem is one of the most extensively researched combinatorial optimization problems. The harmony search algorithm used in their research is a heuristic optimization technique that has shown promising results in solving a variety of complex problems.

Sara Salem utilized the quadratic interpolation optimization algorithm to solve the knapsack problem with high accuracy[5]. The algorithm works by converting the continuous search space of the recently proposed quadratic interpolation optimization (QIO) into a discrete search space using various V-shaped and S-shaped transfer functions. Through the use of different instances, this study achieved the best possible results.

In a study conducted by Arish Pitchai and his team[6], they have proposed a new approach to solve the knapsack problem. They have named this approach as QWGA "Quantum Walk Genetic Algorithm." This algorithm is based on qubit representation and superposition phenomenon which are the counter-intuitive characteristics of quantum mechanics. The results of this study showed that the proposed algorithm has achieved higher performance compared to the quantum algorithm based on rotation operators.

Ravneil Nand and Priynka Sharma have developed a new approach that solves the Multi Knapsack Problem (MKP) effectively[7]. This approach combines two algorithms - the Firefly Algorithm (FA) and the Genetic Algorithm (GA) to achieve optimal results. The system they created using this approach has shown remarkable performance compared to using each algorithm separately.

Indresh Kumar Gupta has developed a new algorithm that combines the genetic algorithm (GA) and gravitational search algorithm (GSA) to solve the multidimensional knapsack problem (MDKP)[8]. In this approach, the GA is used for global search while the GSA is used for local search. The algorithm selects k% of the population after creating individuals using the GA algorithm, and the rest (100-k)% of the population is selected using the GSA algorithm. The top k% of the population is then chosen as the surviving population for the new generation of GA. The combination of GA and GSA yielded better results compared to using each algorithm separately.

Ameen Shaheen and Azzam Sleit investigated various algorithms as a potential solution for the Knapsack problem[9]. They compared the outcomes of each algorithm and determined the most efficient one. The algorithms evaluated were the genetic algorithm, branch and bound algorithm, greedy algorithm, and dynamic programming algorithm. Upon applying these algorithms to the same data and scrutinizing the results, they concluded that the genetic algorithm was the most effective in resolving the problem.

In their study, Frumen Olivas and colleagues used the fuzzy hyper-heuristics (FHH) technique to solve the knapsack problem[10]. This approach is a potent technique that combines low-level heuristics to solve optimization problems. The researchers compared a fuzzy hyper-heuristic model optimized by a genetic algorithm with three traditional selection hyper-heuristic models. All these approaches were applied to the same set of low-level heuristics. The techniques showed a high performance in solving the knapsack problem.

Shang Gao and colleagues utilized Estimation of Distribution Algorithms (EDAs) to tackle the knapsack problem[11]. The approach involves determining the probability of individual distributions in the next generation, the next generation being formed through random sampling. This technique is highly reliable and effective in solving the knapsack problem.

Yazeed Ghadi et al have developed a new approach to solve the knapsack problem[12]. The Group Counseling Optimizer (GCO) is an emerging evolutionary algorithm that simulates human behavior of counseling within a group to solve problems. GCO has been successfully applied to single and multi-objective optimization problems. In this approach, an item is either selected or dropped entirely to fill the knapsack in a way that the total weight of selected items is less than or equal to the knapsack size, and the value of all items is as significant as possible. The results of this approach have shown its efficiency in solving the knapsack problem.

Brief table of previous studies.

Table 2-1 provides a summary of previous knapsack problem studies, including researchers' names, year of publication, methods used and the results.

Table 1: a comparison of knapsack problem studies

researchers	Year	Methods used	results
Shang Gao et al.	2014	Estimation of Distribution Algorithms	This technique is highly reliable and effective in solving the knapsack problem.
Arish Pitchai et al.	2015	Quantum Walk Genetic	QWGA is better than GGA.

		Algorithm QWGA, Greedy genetic algorithm (GGA)	
Ameen Shaheen et al.	2016	genetic algorithm, branch and bound algorithm, greedy algorithm, and dynamic programming algorithm.	The genetic algorithm was the most effective in resolving the problem.
Indresh Kumar Gupta	2018	Genetic algorithm(GA) , gravitational search algorithm(GSA)	By combining these approaches, highly accurate results were achieved
Ravneil Nand et al.	2019	Multi Knapsack Problem (MKP), Firefly Algorithm (FA) + Genetic Algorithm (GA)	The FAGA model worked quite well on multidimensional knapsack problems.
Amol C. Adamuthe, et al.	2020	Harmony search (HS) algorithm	Proposed HS has enhanced speed and performance.
Frumen Olivas et al.	2020	Fuzzy- based selection hyper-heuristic approach	The proposed method has achieved better results than low-level and traditional selection hyper-heuristics.
Sara Salem	2023	quadratic interpolation optimization (QIO)	Proposed study achieved the best possible results.
Yazeed Ghadi et al.	2023	Group Counseling Optimizer (GCO)	The results of this approach have shown its efficiency in solving the knapsack problem.

Methodology:

In this part, we will introduce the knapsack problem, and then we will describe the steps of the PC algorithm used to solve the KP.

Knapsack Problem Definition:

The knapsack problem is a combinatorial problem. W represents the positive capacity of the knapsack. An individual can place a collection of x different items in the knapsack. The weight of item ‘i’ is a positive integer ‘w_i’, while the value of item ‘i’ is a positive integer p_i [13].The objective is to:

$$\sum_{i=1}^{m} p_i\, x_i$$

Subject to: $\sum_{i=1}^{m} w_i\, x_i \leq W$

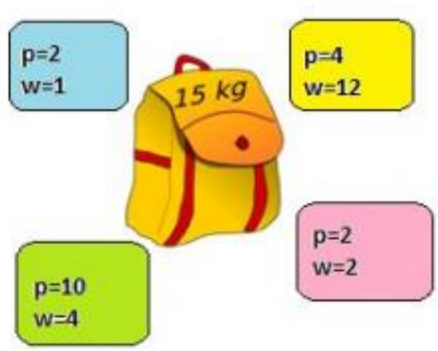


Figure 1: Describes the Knapsack Problem.

Suppose there is a knapsack with a size of 15 items and many items of varying weights and values. Within the constraints of the knapsack's capacity, we desire to maximize the worth of goods included in the knapsack. Then, items were used (1, 2, and,3). The next are their weights and values shown in Table 2:
We need to maximize the total value:

Table 2: Detail of example knapsack problem items.

item	1	2	3
values	20	30	25
weights	9	6	7

Table 3: various Solutions of the Knapsack problem

1	**2**	**3**	**Total weight**	**Total value**
0	0	0	0	0
0	0	1	7	25
0	1	0	6	30
0	1	1	13	55
1	0	0	9	20
1	0	1	16	45
1	1	0	15	50
1	1	1	22	75

$$\sum_{i=1}^{3} 20\, x_1 + 30x_2 + 25x_3 \quad x_1 \in (0,1)$$

Subject to:

$$\sum_{i=1}^{3} 9\, x_1 + 6x_2 + 7x_3 \leq 15$$

There are 2^3 =8 possible subsets of items for this issue, as shown in Table 3. Two solutions exceed the capacity of knapsack, and they are (1, 3), (1, 2, 3). The optimal value for the specific constraint (W = 15) is 50, which is reached with 2 and 3.

Probability Collectives Algorithm (PCA).

In PCA, each agent i is connected with a distinct variable x_i and is supposed to have m_i potential strategies (actions or moves) with which it is adjusting its variable. Thus, variable xi is allocated by agent i as [5]:

$$X_i = \left\{X_i^{[1]}, X_i^{[2]}, \dots\dots\dots, X_i^{[m_i]}\right\}, \quad i \in \{1,2, \dots\dots.. N\} \tag{1}$$

Each agent x_i^k 1≤ k $\geq m_i$, is a randomly selected value from $[x_i^L, x_i^H]$ using the probability distribution q (x_i) connected with agent. Each agent collects the strategy M, which is chosen randomly by other agents as:

$$Y_i^{[1]} = \left\{X_1^{[?]}, X_2^{[?]}, \ldots\ldots, X_i^{[1]}, \ldots\ldots\ldots, X_{N-1}^{[?]}, X_N^{[?]}\right\} \qquad (2)$$

The superscript [?] represents random selection, and each agent has formed one strategy set for each of the residual strategies. Thus, the set of solutions created by agent i as illustrated below:

$$\begin{aligned} Y_i^{[2]} &= \left\{X_1^{[?]}, X_2^{[?]}, \ldots\ldots, X_i^{[2]}, \ldots\ldots\ldots, X_{N-1}^{[?]}, X_N^{[?]}\right\} \\ Y_i^{[3]} &= \left\{X_1^{[?]}, X_2^{[?]}, \ldots\ldots, X_i^{[3]}, \ldots\ldots\ldots, X_{N-1}^{[?]}, X_N^{[?]}\right\} \\ Y_i^{[r]} &= \left\{X_1^{[?]}, X_2^{[?]}, \ldots\ldots, X_i^{[r]}, \ldots\ldots\ldots, X_{N-1}^{[?]}, X_N^{[?]}\right\} \\ Y_i^{[m_i]} &= \left\{X_1^{[?]}, X_2^{[?]}, \ldots\ldots, X_i^{[m_i]} \ldots\ldots, X_{N-1}^{[?]}, X_N^{[?]}\right\} \end{aligned} \qquad (3)$$

After each agent, i estimate the objective function for each combined strategy set $Y_i^{[m_i]}$ as:

$$\left[G\left(Y_i^{[1]}\right), G\left(Y_i^{[2]}\right), \ldots\ldots, G\left(Y_i^{[r]}\right), \ldots\ldots\ldots, G\left(Y_i^{[m_i]}\right)\right] \qquad (4)$$

Per agent locate the sum of the objective function for its collective strategy set to be reduced as follows:

$$\left\{\sum_{k=1}^{m_i} G(Y_1^{[k]}), \sum_{k=1}^{m_i} G(Y_2^{[k]}), \ldots\ldots \sum_{k=i}^{m_D} G(Y_D^{[r]})\right\} \qquad (5)$$

It is very difficult to locate the minimum of function $\sum_{r=1}^{m_i} G(Y_i^{[r]})^n$ because there are multiple possible local minima. For this cause, the objective function is transformed into another topological space by constructing a simple function and putting it in a new form of construction as a Homotopy Function [14].

$$J_i(q(x_i), T) = \sum_{k=1}^{m_i} G\left(Y_i^{[r]}\right) - T * E \quad , \qquad (6)$$

Each agent is associated with a uniform probability distribution $q(x_i)$, where $T \in [0, \infty)$ is a computational parameter that is called temperature. Thus, $\left(X_i^{[r]}\right)$is defined as:

$$q\left(X_i^{[r]}\right) = \frac{1}{m_i} \quad , \text{k} = 1,2,3\ldots\ldots\ldots\ldots, m_i \qquad (7)$$

Each agent i also calculates the predicted value of its objective function $\sum_{r=1}^{m_i} G\left(Y_i^{[r]}\right)$, during utilising combined product probability distribution that is created by $q\left(X_i^{[r]}\right) = \frac{1}{m_i}$, k = 1,..,m_i, and randomly sampled probabilities from distributions of different agents. That is [5]:

$$\left\{\sum_{k=1}^{1} E\left(G\left(Y_i^{[r]}\right)\right), \sum_{k=1}^{2} E\left(G\left(Y_i^{[r]}\right)\right), \ldots\ldots\ldots\ldots \sum_{k=1}^{m_D} E(G\left(Y_i^{[r]}\right))\right\} \qquad (8)$$

Currently, we require to replace E utilized in the Homotopy function and place its place a Convex function such as the Entropy Function [2].

$$S_i = -\sum_{k=1}^{m_i}\left[q\left(X_i^{[r]}\right)\log_2 q\left(X_i^{[r]}\right)\right] \quad (9)$$

Hence, the Homotopy function is reduced for each agent i as:

$$J_i\left(q\left(X_i^{[r]}\right),T\right) = \sum_{r=1}^{m_i} E\left(G(Y_i^{[r]})\right) - T * S_i$$
$$= \sum_{r=1}^{m_i} E\left(G(Y_i^{[r]})\right) - T * \left(-\sum_{r=1}^{m_i}\left[q\left(X_i^{[r]}\right)\log_2 q\left(X_i^{[r]}\right)\right]\right) \quad (10)$$

Where $T \in [0,\infty)$

Many techniques are used to update the probability of all the strategies , such as the Nearest Newton Descent Scheme (NNDS), Broden-Flectcher-Goldfarb-Shanno (BFGS) and Deterministic Annealing (DA) [6]. Thus, the NNDS will be used in this paper to update the probability for each agent i as follows:

$$q\left(X_i^{[k]}\right) \leftarrow q\left(X_i^{[k]}\right) - \propto_{step} * q\left(X_i^{[k]}\right) * K_{r\ update} \quad (11)$$

Where $K_{k\ update} = \frac{Contribution\ of\ agent\ i^k}{T} + s_i(q) + \ln\left(q\left(X_i^{[k]}\right)\right) \quad (12)$

And $Contribution\ of\ agent\ i^k = E\left(G\left(Y_i^{[k]}\right)\right)_n - \left(\sum_{r=1}^{m_i} E\left(G\left(Y_i^{[k]}\right)\right)\right)_n \quad (13)$

Where T is Boltzamann's temperature, which takes a value $\in (0,1]$, which starts from $T \gg 0\ or\ T = T_{intial}\ or\ T \to \infty$. And also, K is the number of iterations and $s_i(q)$ is the Entropy Function of the agent. Each strategy has the maximum contribution to the minimization of the expected utility function. This strategy is comprehended as a suitable combined strategy$X_i^{[fav]}$. The objective function $(G(Y^{fav})^n$ computed for all agents, where Y_{fav} is given by $Y^{fav} = \{X_1^{fav,n}, X_2^{fav,n}, .., X_{N-1}^{fav,n}, X_N^{fav,n}\}$. The PC algorithm updates the boundaries of variables Ω and Boltzmann's temperature as follows:

$$X_i^L(n+1) = (1-\lambda) * X_i^{fav} \quad , i=1\ldots N \quad (14)$$

$$X_i^H(n+1) = (1+\lambda) * X_i^{fav} , i = 1,\ldots\ldots\ldots, N \quad (15)$$

$$T_{n+1} = (1-\alpha_T) * T_n \quad (16)$$

Where $0 < \lambda < 1$the range is factor and $0 < \alpha_T < 1$ is the cooling rate. The algorithm PC continues until one of mentioned criteria is satisfied as follow

- If temperature $T \to 0$.
- If $||\ G(Y^{fav})_n - G(Y^{fav})_{n-1}|| \leq \varepsilon$ where $\varepsilon > 0$.

The PCA of knapsack problem

In this problem, we applied the PC algorithm as follows:

1. Initial parameters $(T, \lambda, \alpha_s, \alpha_T, K, M_i, Maxit, Runs)$, as well as the parameters of KP problem is as:
   N is the number of items where $N = 10, 50, 100, 150, 200$.
   W is the capacity of knapsack where W=150.
2. Initialize the weights randomly among (1:100) such as:

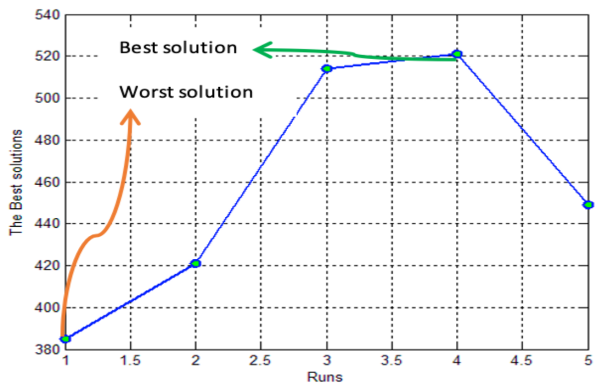
w= [10,20,50,15,60,100,3,2,18,80] when N=10.

3. Initial the strategy q to each agent i .
4. Initial items $p_i^{[m_i]}$ random such as $p_1 = \left\{2_1^{[1]}, 6_1^{[2]}, \dots \dots \dots, p_1^{[m_i]}\right\}$.
5. find a set of solution $Y_i^{[m_i]}$ then evaluate fitness function.
6. find expected function E and the homotopy function.
7. update probabilities value for k of iteration.
8. find the maximum probability.
9. evaluate objective function, and update T
10. repeat until the number of iteration $\geq Maxit$**,** and then show the results.

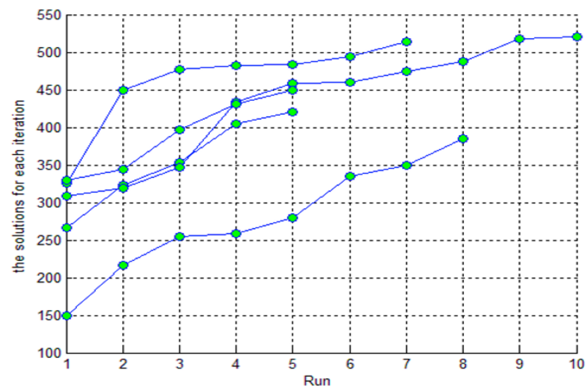
Results and Discussion:

The suggested algorithm is implemented using MATLAB programming language and tried on a computer with the following specifications: Windows 10, Intel core i5-3210M CPU 3 GHz and 4 GB RAM. We implemented the PC algorithm of the knapsack problem for different items N where these items belong to the set {10, 20, 30, 40, and 50}. Each item has N strategies, starting from uniform probability and random initial values. So, the problem is iterated 150 times over 5 runs, so the capacity of KP (W) is 150. Moreover, we took the best solutions for each run.

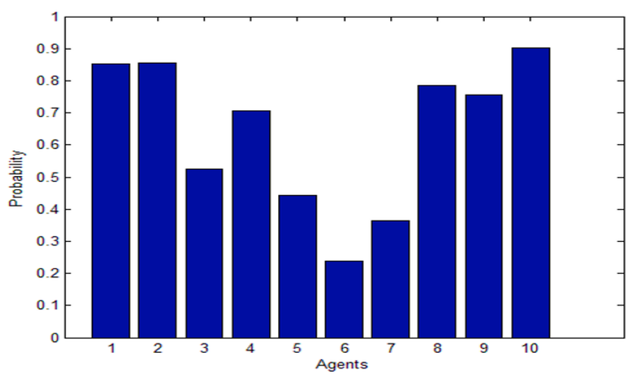
Figure (2-a) shows the performance of the probability collectives algorithm for the knapsack problem when the number of items is 10, where we get the optimal solution of about 521 and the worst solution 385 about over 4 run. Because the favorable strategy of some items is close to one, as shown in Figure (2-c). You can also see all the results for five runs in Figure (2-b).



(a) The best and worst solution over 5 runs.



(b) The solutions for each iteration



(c) Favorable strategy of 10 items

Figure 2: describes the performance of PCA for knapsack problem N=10.

To sum up, the results of the knapsack problem with different sizes of 20, 30, 40, and 50 reached the best solutions of 690, 831, 901, and 987 respectively. The worst solutions are 513,442,604 and 794 as illustrated in Table 4.

Table 4 demonstrates a summary of outcomes for the set of items belonging to {10, 20, 30, 40, and 50} and also shows the average of all results and the standard deviation over five runs. We found the mean best solutions and the standard deviation, such as mean=458 and STD 58.9152 at 22 571.830657 when N equals 10, and we also obtained the best and the worst solutions for all sizes of items. From Table 4, it is clear that the majority of results differ depending on a set of items and weights when we use the size small of items that give a convergence rate faster than other sizes, as shown in Table 4.

Table 4: The results of knapsack problem N= {10, 20, 30, 40, 50}

Number of Items	Best solution	Worst solution	mean	STD DEV	Execution Time
10	521	385	458	58.9152	571.830657 seconds
20	690	513	604.4	62.72	5569.249837 seconds
30	831	442	669.4	147.7	10432.885089 seconds
40	901	604	806.6	118.8709	17250.634935 seconds
50	987	794	898.6	73.86	22054.12301 seconds

Conclusion and Future Work

In this paper, we implemented a new metaheuristic algorithm for solving knapsack problems with various items. In addition, the capacity of the knapsack is a fixed value over five runs. The outcomes show that the PC algorithm was effective and powerful enough to solve this problem.

In the future task, we will implement the PC algorithm to solve the multi-knapsack problem with different sizes and capacities, and we will also compare the PC algorithm with various techniques, such as genetic algorithm, simulated annealing, and particle swarm intelligence to gain a better performance to solve this problem regarding the reduction of time-consuming.

References:

[1] A. J. Kulkarni, I. R. Kale, and K. Tai, “Probability collectives for solving truss structure problems,” in Proceedings of 10th World Congress on Structural and Multidisciplinary Optimization, 2013.

[2] A. J. Kulkarni and K. Tai, “Probability collectives for decentralized, distributed optimization: a collective intelligence approach,” in 2008 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2008, pp. 1271–1275.

[3] A. A. Gaivoronski, A. Lisser, R. Lopez, and H. Xu, “Knapsack problem with probability constraints,” Journal of Global Optimization, vol. 49, pp. 397–413, 2011.

[4] A. C. Adamuthe, V. N. Sale, and S. U. Mane, “Solving single and multi-objective 01 knapsack problem using harmony search algorithm,” Journal of scientific research, vol. 64, no. 1, 2020.

[5] S. Salem, “An Improved Binary Quadratic Interpolation Optimization for 0-1 Knapsack Problems,” 2023.

[6] A. Pitchai, A. V. Reddy, and N. Savarimuthu, "Quantum Walk based genetic algorithm for 0–1 quadratic knapsack problem," in 2015 International Conference on Computing and Network Communications (CoCoNet), IEEE, 2015, pp. 283–287.

[7] R. Nand and P. Sharma, "Iteration split with Firefly Algorithm and Genetic Algorithm to solve multidimensional knapsack problems," in 2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), IEEE, 2019, pp. 1–7.

[8] I. K. Gupta, "A hybrid GA-GSA algorithm to solve multidimensional knapsack problem," in 2018 4th International Conference on Recent Advances in Information Technology (RAIT), IEEE, 2018, pp. 1–6.

[9] A. Shaheen and A. Sleit, "Comparing between different approaches to solve the 0/1 Knapsack problem," International Journal of Computer Science and Network Security (IJCSNS), vol. 16, no. 7, p. 1, 2016.

[10] F. Olivas, I. Amaya, J. C. Ortiz-Bayliss, S. E. Conant-Pablos, and H. Terashima-Marin, "A Fuzzy Hyper-Heuristic Approach for the 0-1 Knapsack Problem," in 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.

[11] S. Gao, L. Qiu, and C. Cao, "Estimation of Distribution Algorithms for Knapsack Problem.," J. Softw., vol. 9, no. 1, pp. 104–110, 2014.

[12] Y. Y. Ghadi et al., "An efficient optimizer for the 0/1 knapsack problem using group counseling," PeerJ Computer Science, vol. 9, p. e1315, 2023.

[13] N. Moradi, V. Kayvanfar, and M. Rafiee, "An efficient population-based simulated annealing algorithm for 0–1 knapsack problem," Engineering with Computers, vol. 38, no. 3, pp. 2771–2790, 2022.

[14] Z. Xu, A. Unveren, and A. Acan, "Probability collectives hybridised with differential evolution for global optimisation," International Journal of Bio-Inspired Computation, vol. 8, no. 3, pp. 133–153, 2016.

[15] Kulkarni, A.J., K. Tai, and A. Abraham, Probability Collectives - A Distributed Multi-agent System Approach for Optimization. Intelligent Systems Reference Library. Vol. 86. 2015: Springer. 1-144.

Contents