STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–8.
Published online in International Academic Press (www.IAPress.org)

Solving Multidimensional Knapsack Problem using an improved Reptile Search Algorithm

Rana Bashar Hussein [1], Zakariya Yahya Algamal [2,*]

[1] *Department of management Information systems, College of Administration and Economics, University of Mosul, Mosul, Iraq*
[2] *Department of Statistics and Informatics, University of Mosul, Mosul, Iraq*

Abstract The Multidimensional Knapsack Problem (MKP) is a well-known NP-hard combinatorial optimization problem with broad applications in management and engineering, including logistics, finance, and resource allocation. MKP involves selecting a subset of items to maximize total profit while respecting multiple resource constraints simultaneously. Traditional and nature-inspired metaheuristic algorithms have been widely used to tackle its computational complexity. This study proposes the integration of Z-shaped transfer functions into the binary Reptile Search Algorithm (RSA) to enhance its performance in solving MKP. Empirical evaluations conducted on five widely-used MKP benchmark datasets demonstrate that RSA with Z-shaped transfer functions competes favorably or surpasses other state-of-the-art transfer function variants in terms of solution quality and convergence. These results underscore the potential of Z-shaped transfer functions in improving binary metaheuristic algorithms for solving complex multidimensional combinatorial problems.

Keywords Multidimensional knapsack problem, reptile search algorithm, optimization methods, transfer functions, combinatorial optimization problem

DOI: 10.19139/soic-2310-5070-3126

1. Introduction

Combinatorial optimization is the search of an optimal object in the finite set of discrete structures. It is a key concept in operations research, in computer science, and in applied mathematics, and finds numerous applications in logistics, in telecommunications, in artificial intelligence and of particular use in chemistry and materials science. Combinatorial optimization is the solving of a problem in which the variable is discrete and (usually very large) but the objective is to be optimized. Examples of classical problems include traveling salesman problem, knapsack problem and graph coloring. The difficulty arises in the fact that there are so many possible solutions and the combinatorial explosion in the number of possible solutions that makes the total-search impractical on all but the simplest problems [1, 2].

In combinatory search work, we more typically have to deal with problems which have a finite number of results. Knapsack problems (KP) occur in most of the sciences and engineering [2, 3, 4]. The decision vectors in KP are discrete valued [5]. An attempt to take care of this difficulty is to approximate the problems by regular optimization problems (problems specifying the optimization over continuous-valued decision vectors). The MKP is a standard NP-hard multi-dimensional knapsack problem to use the discrete valued decision vectors to solve the principal optimization problem [6, 7, 8]. The MKP is a core version of the knapsack problem that takes into account more than one resource constraint. MKP have been studied due to their complex computational nature and diverse areas of applications, such as cargo loading, budget allocation, cutting stock and portfolio selection [9].

*Correspondence to: Zakariya Yahya Algamal (Email: zakariya.algamal@uomosul.edu.iq). Department of Statistics and Informatics, University of Mosul, Mosul, Iraq.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Meta-heuristic algorithms have become a high-level general purpose optimization framework which can be used to solve a vast number of problems which are complex, where it would be computationally prohibitive or even infeasible to compute an exact solution. In contrast to the problem-specific heuristics, metaheuristics are independent of a particular problem and can be adapted with little tuning to different optimization problems. They are designed to effectively search large and computationally tough solution spaces to find a near-optimal or optimal solution, by successively enhancing candidate solutions numerously using rules of thumb that are commonly inspired by natural processes or human intuition [10, 11].

MKP is an extension of the classical problem of the knapsack whereby there are several constraints on resources, and it is applicable in cargo loading problems, financial portfolio selection, production and other problems that involve allocation of resources in the real world. Officially, it aims at maximizing the total profit of a subset of items without breaching any of the several capacity restrictions. Exact branch-and-bound or integer program solver-based algorithms have run times that are NP-hard, and therefore are only able to solve small to medium sized problems. Meta-heuristic algorithms offer heuristic schemes that are powerful and balance exploration and exploitation throughout the large amount of solution space to solve large-scale MKPs. Such algorithms use a mixture of stochasticity and heuristics based on domain knowledge, and usually have a complex feasibility repair mechanism to search the multidimensional space of constraints [2, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24].

In this paper, an improved reptile search algorithm is proposed to improve multidimensional knapsack problem solving. Our proposed algorithm can efficiently exploit the strong points of reptile search algorithm in finding the best solution with high performance. The experimental results show the favorable performance of the proposed hybridization when the number of dimensions is high and the sample size is low.

2. Multidimensional knapsack problem

The MKP is a famous NP-hard combinatorial optimization problem with strong engineering backgrounds [25, 26]. The MKP can be defined as follows:

$$Maximize \quad f(d_1, d_2, ..., d_n) = \sum_{j=1}^{n} c_j d_j \tag{1}$$

$$Subject\ to \ \sum_{j=1}^{n} r_{ij} d_j \ \le \ b_i \ , i = 1, 2, ..., m \tag{2}$$

$$with \quad d_j \in \{0, 1\}\ , \ \ j = 1, 2, ..., n, \ \ c_j \succ 0, r_{ij} \ge 0, b_i \ge 0 \tag{3}$$

where n is the number of items, $m$is the number of knapsack constrains, c_j is profit of the jth item, $b_i$is the capacity of the ith knapsack and r_{ij} denotes the unit cost of the jth item on the ith knapsack. The aim for solving the MKP is to achieve a subset of items with maximum profit while taking care of constrains.

3. Binary reptile search algorithm

Metaheuristic algorithms based on nature have been widely used to address problems of complex optimization because of its simplicity and capability of avoiding local optima. Reptile Search Algorithm (RSA), proposed in 2022 [27], is a new swarm intelligence-based algorithm, which is based on hunting behavior patterns of crocodiles and applies their circling and collective hunting patterns to balance out the phases of exploration and exploitation.

RSA works with four major states related to different paths of performing hunting by dynamically altering the positions of candidate solutions in every iteration [28]. A number of modified versions have been developed to improve diversity in a population, speed of convergence, and help prevent premature stagnation.

Classical RSA is applicable on continuous search spaces only, thus directly applicable on discrete problems only. To overcome this, binary version of RSA has been designed by encoding solutions in a fashion that can support combinatorial tasks through transfer functions that take continuous position update streams and make binary decisions based on them [29]. Transfer functions often used in binary metaheuristics include S-shaped, V-shaped, or proposed Z-shaped and tent-shaped functions that are used to discrete a continuous update. The above work has established that the transfer functions play an essential part in determining the capability of the binary algorithm to traverse the search space extensively [8, 30, 32, 33, 34, 35, 36, 37, 38].

RSA has demonstrated competitive performance compared to widely used metaheuristics like Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and Ant Lion Optimizer (ALO). Its adaptability allows it to solve complex problems across continuous, discrete, and large-scale domains. Real-world applications of RSA include engineering design optimization, medical image processing (e.g., MRI segmentation), renewable energy system optimization, and hyperparameter tuning in machine learning.

In RSA, the optimization process starts with a set of candidate solution (X) as shown in eq. (1), which is generated stochastically and the best- obtained solution is considered as the nearly the optimum in each iteration.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & x_{i,j} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \tag{4}$$

Where X is asset of the candidate solutions that are generated randomly by using Eq.(2), $x_{i,j}$ denotes to the j_{th} position of the i_{th} solution, N is the number of candidate solutions, and n denotes to the dimension size of the given problem.

$$x_{ij} = rand \times (UB - LB) + LB \quad , \quad j = 1, 2, ..., n \tag{5}$$

Where rand is a random value, LB and UB denote to the lower and upper bound of the given problem, respectively.
First: Encircling phase

In this section, the exploratory behaviour (encircling) of RSA is introduced. According to the encircling behaviour, Crocodiles have two movements during the encircling are high walking and belly walk. These movements refer to different reigns, which commitment to the exploration search (globally). Crocodile movements (high and belly walking) cannot allow them to approach the target prey due to their disturbance easily, unlike another search phase (hunting phase). Hence, the exploration search discovers a wide search space; it can find the density area maybe after several endeavours. In addition, the exploration mechanisms (high and belly walking) are operated at this stage of optimization to support the other phase (hunting/exploration) in the search process through extensive and spread research.

The RSA can transfer between encircling (exploration) and hunting (exploitation) search phases, this change between various behaviors is done based on four conditions; divide the total number of iterations into four parts. The exploration mechanisms of RSA explore the search regions and approach to find a better solution based on two main search strategies (high walking strategy and belly walking strategy). This phase of searching is conditioned on two conditions. The high walking movement strategy is conditioned by $t < \frac{T}{4}$, and the belly walking movement strategy is conditioned by $t \leq 2\frac{T}{4}$ and $t > \frac{T}{4}$.

This means that this condition will be satisfied for almost the half number of exploration iterations (High walking) and another half for the Belly walking. These are two exploration search methods. Note, a stochastic scaling coefficient is examined for the element to generate more diverse-solutions and explore diverse-regions. We employed the most straightforward rule, which can mimic the encircling behavior of Crocodiles. In this paper, the position updating equations are proposed for the exploration phase as in Eq.(3).

$$x_{(i,j)}(t+1) = \begin{cases} Best_j(t) \times -\eta_{(i,j)}(t) \times \beta - R_{(i,j)}(t) \times rand, & t \leq \frac{T}{4} \\ Best_j(t) \times x_{(r_1,j)} \times ES(t) \times rand, & t \leq 2\frac{T}{4} \ \ and \ \ t > \frac{T}{4} \end{cases} \tag{6}$$

where $Best_j(t)$ is the j_{th} position in the best-obtained solution so far, r denotes to a random number between 0 and 1, t is the number of the current iteration, and T is the maximum number of iterations. $\eta(i,j)$ denotes to the hunting operator for the j_{th} position in the i_{th} solution, which is calculated using Eq.(4). β is a sensitive parameter, controls the exploration accuracy (i.e., High walking) for encircling phase over the course of iterations, which is fixed equal to 0.1. Reduce function $R_{(i,j)}$ is a value used to reduce the search area, which is calculated using Eq.(5). r_1 is a random number between $[1 \quad N]$ and $x_{(r_1,j)}$ denotes to a random position of the i_{th} solution. N is the number of the candidate solutions. Evolutionary Sense $(ES\,(t))$ is a probability ratio takes randomly decreasing values between 2 and -2 throughout the number of iterations, which is calculated using Eq.(6).

$$\eta_{(i,j)} = Best_j(t) \times P_{(i,j)} \tag{7}$$

$$R_{(i,j)} = \frac{Best_j(t) - x_{(r_2,j)}}{Best_j(t) + \varepsilon} \tag{8}$$

$$Es(t) = 2 \times r_3 \times (1 - \frac{1}{T}) \tag{9}$$

where, ε a small value and r_2 is a random number between $[1 \quad N]$. In Eq.(6), 2 is used as a correlation value to give values between 2 and 0, r_3 denotes to a random integer number between -1 and 1. $p_{(i,j)}$ is the percentage difference between the j_{th} position of the best obtained solution and the j_{th} position of the current solution, which is calculated using Eq.(7).

$$P_{(i,j)} = \alpha + \frac{x_{(i,j)} - M(x_i)}{Best_j(t) \times (UB_{(j)} - LB_{(j)} + \varepsilon} \tag{10}$$

where $M\,(x_i)$, as in Eq.(7), is the average positions of the i_{th} solution, which is calculated using Eq. (8). $UB_{(j)}$ and $LB_{(j)}$ are the upper and lower boundaries of the j_{th} position, respectively. α is a sensitive parameter, controls also the exploration accuracy (the difference between candidate solutions) for the hunting cooperation over the course of iterations, which is fixed equal to 0.1 in this paper.

$$M(x_i) = \frac{1}{n}\sum_{j=1}^{n} x_{(i,j)} \tag{11}$$

Second: Hunting phase (exploitation) In this section, the exploitative behavior (hunting) of RSA is introduced. According to the hunting behavior, Crocodiles have two strategies during the hunting are hunting coordination and cooperation. These strategies refer to different intensify techniques, which commitment to the exploitation search (locally). Crocodile strategies (hunting coordination and cooperation) allow them to approach the target prey easily due to their intensification, unlike encircling mechanisms. Hence, the exploitation search discovers the near-optimal solution, maybe after several endeavors. Besides, the exploitation mechanisms are operated at this stage of optimization to conduct an intensification search near the optimal solution and emphasized communication between them.

The exploitation mechanisms of RSA exploit the search space and approach to find the optimal solution based on using two main search strategies (i.e., (1) hunting coordination and Eq.(2) hunting cooperation), which is modelled as in Eq.(9). The searching in this phase is conditioned as the hunting coordination strategy is conditioned by $t \leq 3\frac{T}{4}$ and$t > 2\frac{T}{4}$, otherwise, the hunting cooperation strategy is performed, when $t \leq T$ and $t > 3\frac{T}{4}$. Note, stochastic coefficients are considered to generate more dense-solutions and exploit the promising regions (locally). We employed the most straightforward rule, which can mimic the hunting behavior of Crocodiles. In this paper, the following position updating equations are proposed for the exploitation phase (Eq. (9)):

$$x_{(i,j)}(t+1) = \begin{cases} Best_j(t) \times P_{(i,j)}(t) \times rand, & t \leq 3\frac{T}{4} \quad and \quad t > 2\frac{T}{4} \\ Best_j(t) - \eta_{(i,j)}(t) \times \varepsilon - R_{(i,j)}(t) \times rand, & t \leq T \quad and \quad t > 3\frac{T}{4} \end{cases} \tag{12}$$

where $Best_j(t)$ is the j_{th} position in the best-obtained solution so far, $\eta\,(i,j)$ denotes to the hunting operator for the j_{th} position in the i_{th} solution, which is calculated using Eq.(4). $P_{(i,j)}$ is the percentage difference between the j_{th} position of the best-obtained solution and the j_{th} position of the current solution, which is calculated using Eq.(7). $\eta\,(i,j)$ denotes to the hunting operator for the j_{th} position in the i_{th} solution, which is calculated using

Eq.(4). ε a small value. $R_{(i,j)}$ is a value used to reduce the search area, which is calculated using Eq. (5). In this respect, Figs. 1 and 2 show that when $t \leq \frac{T}{2}$, the encircling phase (exploration) happens, otherwise; when $t > \frac{T}{2}$, the hunting phase (exploitation) occurs to be close enough to prey when attacking.

Exploitation search mechanisms (hunting coordination and cooperation) are attempting to evade getting trapped in the local optima. These procedures assist the exploration search in determining the optimal solution and maintain the diversity over the candidate solutions. We carefully designed two parameters (i.e., β and α) to produce a stochastic value at each iteration, continue exploration not only during the first iterations but also last iterations. This part of searching is beneficial in the situation of local optima stagnation, particularly in the last iterations.

Algorithm 1 Pseudo-code of the Reptile Search Algorithm (RSA)

Initialization phase
Initialize RSA parameters α, β, etc.
Initialize the solutions' positions randomly. $X : i = 1, 2, ..., n$.
while $(t < T\ )$ **do**
Calculate the Fitness Function for the candidate solutions (X).
Find the Best solution so far.
Update the ES using Equations (6).
The beginning of the RSA
for $(i = 1\ to\ N)$ **do**
for $(j = 1\ to\ n)$**do**
Update the η, R, P and values using Equations (4), (5) and (7) respectively.
if $(t \leq \frac{T}{4})$ **then**
$x_{(i,j)}(t+1) = Best_j(t) \times -\eta_{(i,j)}(t) \times rand,$ ▷ {**High walking**}
else if $(t \leq 2\frac{T}{4}\ \ and\ \ \ t > \frac{T}{4})$ **then**
$x_{(i,j)}(t+1) = Best_j(t) \times x_{(r_1,j)} \times ES(t) \times rand,$ ▷{**Belly walking**}
else if $(t \leq 3\frac{T}{4}\ \ and\ \ \ t > 2\frac{T}{4}\ )$ **then**
$x_{(i,j)}(t+1) = Best_j(t) \times P_{(i,j)}(t) \times rand,$ ▷{**Hunting coordination**}
else
$x_{(i,j)}(t+1) = Best_j(t) \times \eta_{(i,j)}(t) \times \varepsilon - R_{(i,j)}(t) \times rand,$ ▷{**Hunting cooperation**}
end if
end for
end for

$$t = t+$$

end while
Return the best solution $(Best(X))$.

4. The proposed algorithm

The binary RSA uses the transfer function (TS) as its key aspect. The most suitable solution to the problem will involve implementation of a binary representation of 0 or 1. To be able to accomplish this objective. The transfer function is very simply and precisely calculated to give the likelihood of an element of a position vector varying between 0 and 1 (or the converse).

In this study, the Z-Shaped transfer functions of Guo, et al. [31] were modified and suggested. The mapping function that applies to these transfer functions is the asymmetric mapping function. The convergence rate is described as fast because of the effect of this asymmetric mapping function which is an absolute fulfillment in mapping the probability of the member position vector fluctuation.

One way to represent the Z-Shaped transfer function (ZTF) is as

$$T\left(x_i^k(t)\right) = \sqrt{1 - a^{x_i^k(t)}}\ , \tag{13}$$

where the number k is positive. By changing the value of k, a set of Z-shaped function families can be obtained. Table 1 illustrates the four different ZTFs.

Table 1. Z-shaped transfer functions

Name	Expression
Z_1	$T_{Z1}(x) = \sqrt{1-2^x}$
Z_2	$T_{Z2}(x) = \sqrt{1-5^x}$
Z_3	$T_{Z3}(x) = \sqrt{1-8^x}$
Z_4	$T_{Z4}(x) = \sqrt{1-20^x}$

5. Results and discussion

In this section, the performance of the proposed Z- transfer functions for solving MKP is assessed using several well-known benchmarks. The performance is compared with S-shaped transfer (STF) and V-shaped transfer (VTF) functions. The MKP instances are varying from small-scale instances to medium- and large-scale instances which are listed in Table 2.

Table 2. Characteristics of MKP instances

Instances	n	m
WEISH01	30	5
WEISH02	30	5
WEISH06	40	5
WEISH12	50	5
WEISH17	60	5
WEISH18	70	5
SENTO1	60	30
PB1	27	4
gk02	100	25
Gk03	150	25

Table 3 displays test results for MKP instances. For each instance, the Table 3 reports the best-known solutions and the average solutions found by the used transfer functions solutions over 30 runs for each instance. It is clearly seen that each transfer function attempts to approximate or match the best-known solution. The Z-shaped transfer functions (Z1–Z4) frequently reach the best-known solutions or come very close, indicating strong solution quality. While STF and VTF occasionally fall slightly short of the best-known solutions, showing comparatively weaker performance.

Z-shaped transfer functions regularly matched or closely approximated the best-known solutions for WEISH01 to WEISH18, for example Z1–Z4 all matched 4554 for WEISH01 and 4536 for WEISH02. Similar trends where Z-shaped variants matched the best-known solutions more consistently than STF and VTF in SENTO1 and PB1. While on the gk02 and gk03, Z-shaped transfer functions delivered results matching or surpassing STF and VTF, often exactly achieving the best-known solution.

Related to the effectiveness of transfer functions, Z-shaped transfer functions (Z1–Z4) show high effectiveness across all instances, yielding top-quality solutions with high consistency. On the other hand, VTF and STF sometimes close, they generally show less stable performance and slightly lower solution quality on average.

In terms of the implications on search behavior, Z-shaped transfer functions likely improve the binary RSA's ability to balance exploration and exploitation effectively in the discrete MKP solution space. This enhanced balance results in better convergence to near-optimal or optimal solutions.

Table 4 presents the average amount of time required to get the best solution in order to further demonstrate the efficacy of the method we have designed. In general trend across instances, Table 4 shows that for all MKP instances listed from WEISH01 through Gk03, the average computational times decrease consistently as we move from STF to VTF, then to the different Z-shaped transfer function variants. This trend reflects that Z-shaped transfer

Table 3. The average results for MKP instances for the used transfer functions

Instances	Best known solutions	STF	VTF	Z1	Z2	Z3	Z4
WEISH01	4554	4551	4552	4553	4554	4554	4554
WEISH02	4536	4532	4532	4536	4536	4536	4536
WEISH06	5557	5553	5551	5556	5557	5557	5557
WEISH12	6339	6338	6337	6339	6339	6339	6339
WEISH17	8633	8632	8632	8633	8633	8633	8633
WEISH18	9580	9580	9577	9580	9580	9580	9580
SENTO1	7772	7770	7771	7772	7772	7772	7772
PB1	3090	3082	3083	3090	3090	3090	3090
gk02	3958	3946	3948	3955	3958	3958	3958
Gk03	5656	5648	5647	5656	5656	5656	5656

functions generally require less computing time than STF and VTF on these problems. That is meaning STF generally exhibits the highest running times across most instances. Additionally, VTF shows slightly better (lower) times than STF but is still slower than the Z-shaped variants. Conversely, Z-shaped Transfer Functions (Z1 to Z4) consistently produce the lowest computational times, with times steadily decreasing from Z1 through Z4. Z4 often records the shortest average running time for each instance. For example, For WEISH01, STF takes 125 sec, while Z4 only takes 112 secroughly a 10% reduction. In gk02 and Gk03 instances, STF takes over 230 seconds, whereas Z4 performs notably faster at 215 and 221 seconds respectively.

Related to implications of time reduction, the reduced computational time when using Z-shaped transfer functions suggests they enable the binary RSA to converge more quickly to high-quality or best-known solutions. Improved convergence efficiency is likely due to these transfer functions better balancing exploration and exploitation in the binary search space, thus avoiding unnecessary iterations.

Table 4. The average time results in seconds for MKP instances for the used transfer functions

Instances	STF	VTF	Z1	Z2	Z3	Z4
WEISH01	125	122	121	117	114	112
WEISH02	147	140	143	138	136	134
WEISH06	136	141	132	127	125	123
WEISH12	146	139	142	135	135	131
WEISH17	180	183	176	171	169	167
WEISH18	186	188	182	177	175	173
SENTO1	196	201	192	187	185	183
PB1	205	200	201	196	194	192
gk02	232	240	228	224	221	215
Gk03	239	245	235	230	228	221

To be conclude, the results validate that integrating Z-shaped transfer functions within the binary RSA framework provides superior or at least competitive results compared to traditional STF and VTF. The consistent matching or improvement over best-known solutions underscores the practical advantage of Z-shaped transfer functions in MKP. Selecting the appropriate transfer function is crucial for binary metaheuristics solving combinatorial problems, with Z-shaped functions offering a promising direction for future research and application. In addition, the results demonstrates that Z-shaped transfer functions outperform STF and VTF in computational efficiency for binary RSA solving MKP instances. This reduction confirms the suitability of Z-shaped transfer functions as a computationally efficient and effective means of solving complex combinatorial problems like MKP.

6. Conclusion

The study confirms that the MKP, as a fundamental NP-hard combinatorial optimization problem, remains a critical challenge with extensive applications across management and engineering domains. The integration of Z-shaped transfer functions into the binary RSA has shown to significantly enhance the algorithm's capability to efficiently navigate the discrete solution space of MKP. Experimental results on five widely-used benchmark datasets demonstrate that this improved RSA variant achieves competitive or superior performance in terms of solution quality and convergence speed compared to existing state-of-the-art transfer functions. These outcomes validate the practical value of employing Z-shaped transfer functions within RSA, providing a robust and effective approach for solving complex multidimensional combinatorial problems. This approach not only advances the solution methods for MKP but also offers a promising direction for future research in binary metaheuristic algorithm design and discrete optimization.

REFERENCES

1. F. Chicano, G. Luque, Z. A. Dahi, and R. Gil-Merino, *Combinatorial optimization with quantum computers*, Engineering Optimization, vol. 57, no. 1, pp. 208–233, 2025.
2. N. Weninger and R. Fukasawa, *A fast combinatorial algorithm for the bilevel knapsack problem with interdiction constraints*, Mathematical Programming, vol. 210, no. 1, pp. 847–879, 2025.
3. M. Abdel-Basset, R. Mohamed, M. Abouhawwash, A. M. Alshamrani, A. W. Mohamed, and K. Sallam, *Binary light spectrum optimizer for knapsack problems: an improved model*, Alexandria Engineering Journal, vol. 67, pp. 609–632, 2023.
4. F. A. M. Bargarai and S. A. Saleh, *A Review Enhancing Artificial Bee Colony Algorithm for Solving the Knapsack Problem: Benchmarking, Variations, and Real-World Applications*, European Journal of Applied Science, Engineering and Technology, vol. 3, no. 2, pp. 117–128, 2025.
5. R. P. Das, J. J. Jena, S. C. Satapathy, and N. M. Hannoon, *Solving 0-1 knapsack and bin packing problem using Logical Social Group Optimization*, IEEE Access, 2025.
6. M. Abdel-Basset, R. Mohamed, K. M. Sallam, I. Alrashdi, and I. A. Hameed, *An efficient binary spider wasp optimizer for multi-dimensional knapsack instances: experimental validation and analysis*, Journal of Big Data, vol. 12, no. 1, p. 18, 2025.
7. S. Al-Shihabi, *An optimization framework for solving large scale multidemand multidimensional knapsack problem instances employing a novel core identification heuristic*, European Journal of Operational Research, vol. 320, no. 3, pp. 496–504, 2025.
8. N. A. Al-Thanoon, O. S. Qasim, and Z. Y. Algamal, *A new hybrid pigeon-inspired optimization algorithm for solving multidimensional knapsack problems*, in 2021 7th International Conference on Contemporary Information Technology and Mathematics (ICCITM), 2021: IEEE, pp. 226–229.
9. Y. He, X. Zhang, W. Li, J. Wang, and N. Li, *An efficient binary differential evolution algorithm for the multidimensional knapsack problem*, Engineering with Computers, vol. 37, no. 1, pp. 745–761, 2021.
10. M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, *Metaheuristic algorithms: A comprehensive review*, Computational intelligence for multimedia big data on the cloud with engineering applications, pp. 185–231, 2018.
11. V. Tomar, M. Bansal, and P. Singh, *Metaheuristic algorithms for optimization: A brief review*, Engineering Proceedings, vol. 59, no. 1, p. 238, 2024.
12. N. Yaghini and M. Y. S. Valizadeh, *An Artificial Bee Colony Algorithm for the Multidimensional Knapsack Problem: Using Design of Experiments for Parameter Tuning*, International Journal of Applied Metaheuristic Computing (IJAMC), vol. 15, no. 1, pp. 1–23, 2024.
13. J. Yang, Y.-H. Kim, and Y. Yoon, *A memetic algorithm with a novel repair heuristic for the multiple-choice multidimensional knapsack problem*, Mathematics, vol. 10, no. 4, p. 602, 2022.
14. J. Xu, H. Li, and M. Yin, *Finding and exploring promising search space for The 0–1 Multidimensional Knapsack Problem*, Applied Soft Computing, vol. 164, p. 111934, 2024.
15. A. Rezoug, M. Bader-El-Den, and D. Boughaci, *Application of supervised machine learning methods on the multidimensional knapsack problem*, Neural Processing Letters, vol. 54, no. 2, pp. 871–890, 2022.
16. M. P. Vasanthi, *Performance Analysis of Sequential and Parallel GRASP Algorithms for the 0-1 Multidimensional Knapsack Problem*, International Journal of Advances in Agricultural Science and Technology, vol. 8, no. 8, pp. 210–213, 2021.
17. Y. Feng and G.-G. Wang, *A binary moth search algorithm based on self-learning for multidimensional knapsack problems*, Future Generation Computer Systems, vol. 126, pp. 48–64, 2022.
18. E. Forno, R. Pignari, V. Fra, E. Macii, and G. Urgese, *Path Integral Quantum Annealing Optimizations Validated on 0-1 Multidimensional Knapsack Problem*, IEEE Transactions on Emerging Topics in Computing, 2025.
19. S. Gupta, R. Su, and S. Singh, *Diversified sine–cosine algorithm based on differential evolution for multidimensional knapsack problem*, Applied Soft Computing, vol. 130, p. 109682, 2022.
20. R. Jovanovic and S. Voß, *Matheuristic fixed set search applied to the multidimensional knapsack problem and the knapsack problem with forfeit sets*, OR Spectrum, vol. 46, no. 4, pp. 1329–1365, 2024.
21. M. I. Leghari and A. A. Shaikh, *A Practical Approach Towards Multidimensional Knapsack Problem (MKP) Using Genetic Algorithm*, Journal of Information Communication Technology (JICT), vol. 14, no. 1, 2021.
22. X. Li, W. Fang, S. Zhu, and X. Zhang, *An adaptive binary quantum-behaved particle swarm optimization algorithm for the multidimensional knapsack problem*, Swarm and Evolutionary Computation, vol. 86, p. 101494, 2024.
23. A. Mkaouar, S. Htiouech, and H. Chabchoub, *Modified artificial bee colony algorithm for multiple-choice multidimensional knapsack problem*, IEEE Access, vol. 11, pp. 45255–45269, 2023.
24. R. Nand, K. Chaudhary, and B. Sharma, *Discrete Preference based Stepping ahead Firefly Algorithm for solving Multidimensional Knapsack Problems*, IEEE Access, 2024.
25. B. Haddar, M. Khemakhem, S. Hanafi, and C. Wilbaut, *A hybrid quantum particle swarm optimization for the multidimensional knapsack problem*, Engineering Applications of Artificial Intelligence, vol. 55, pp. 1–13, 2016.

26. V. Boyer, M. Elkihel, and D. El Baz, *Heuristics for the 0–1 multidimensional knapsack problem*, European Journal of Operational Research, vol. 199, no. 3, pp. 658–664, 2009.
27. L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, *Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer*, Expert Systems with Applications, vol. 191, p. 116158, 2022.
28. Z. Elgamal, A. Q. M. Sabri, M. Tubishat, D. Tbaishat, S. N. Makhadmeh, and O. A. Alomari, *Improved reptile search optimization algorithm using chaotic map and simulated annealing for feature selection in medical field*, IEEE Access, vol. 10, pp. 51428–51446, 2022.
29. B. Ervural and H. Hakli, *A binary reptile search algorithm based on transfer functions with a new stochastic repair method for 0–1 knapsack problems*, Computers Industrial Engineering, vol. 178, p. 109080, 2023.
30. Z. A. Almishlih, O. S. Qasim, and Z. Y. Algamal, *Design and evaluation of a new tent-shaped transfer function using the Polar Lights Optimizer algorithm for feature selection*, Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, vol. 15, no. 2, pp. 27–31, 2025.
31. S.-s. Guo, J.-s. Wang, and M.-w. Guo, *Z-shaped transfer functions for binary particle swarm optimization algorithm*, Computational Intelligence Neuroscience, vol. 2020, 2020.
32. Kahya, M. A., Altamir, S. A., & Algamal, Z. Y. *Improving whale optimization algorithm for feature selection with a time-varying transfer function* , Numerical Algebra, Control and Optimization, vol. 11, no. 1 , p. 87-98 , 2020.
33. Al-Fakih, A. M., Qasim, M. K., Algamal, Z. Y., Alharthi, A. M., & Zainal-Abidin, M. H. *QSAR classification model for diverse series of antifungal agents based on binary coyote optimization algorithm* SAR and QSAR in Environmental Research, vol.34, no.4, p. 285-298, 2023.
34. Al-Fakih, A. M., Algamal, Z. Y., & Qasim, M. K. *An improved opposition-based crow search algorithm for biodegradable material classification* SAR and QSAR in Environmental Research, vol.33, no.5, p. 403-415, 2022.
35. Algamal, Z. Y., Lee, M. H., Al-Fakih, A. M., & Aziz, M. *High-dimensional QSAR modelling using penalized linear regression model with L 1/2-norm*, SAR and QSAR in Environmental Research, vol. 27, no.(9), p.703-719, 2016
36. Algamal, Z. Y., Qasim, M. K., Lee, M. H., & Ali, H. T. M. *QSAR model for predicting neuraminidase inhibitors of influenza A viruses (H1N1) based on adaptive grasshopper optimization algorithm*, SAR and QSAR in Environmental Research, vol.31, no.11, p.803-814, 2020.
37. Qasim, M. K., Algamal, Z. Y., & Ali, H. M. *A binary QSAR model for classifying neuraminidase inhibitors of influenza A viruses (H1N1) using the combined minimum redundancy maximum relevancy criterion with the sparse support vector machine*, SAR and QSAR in Environmental Research, vol. 29, no.(7), p.517-527, 2018
38. Al-Fakih, A. M., Algamal, Z. Y., Lee, M. H., Aziz, M., Ali, H. T. M. *QSAR classification model for diverse series of antifungal agents based on improved binary differential search algorithm*, SAR and QSAR in Environmental Research, vol.30, no.2, p.131-143, 2019.