Decision Science Letters 11 (2022) 55–72

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Binary social group optimization algorithm for solving 0-1 knapsack problem

Anima Naik[a*] and Pradeep Kumar Chokkalingam[b]

[a]*Department of CSE, KL University, Hyderabad, Telangana, India*
[b]*Department of IS, HPCL, Mumbai, Maharashtra, India*

CHRONICLE

Article history:
Received May 5, 2021
Received in revised format: June 1, 2021
Accepted August 18 2021
Available online August 19, 2021

Keywords:
Combinatorial optimization problem
Meta-heuristic algorithms
0-1 knapsack
Binary algorithm
Performance

ABSTRACT

In this paper, we propose the binary version of the Social Group Optimization (BSGO) algorithm for solving the 0-1 knapsack problem. The standard Social Group Optimization (SGO) is used for continuous optimization problems. So a transformation function is used to convert the continuous values generated from SGO into binary ones. The experiments are carried out using both low-dimensional and high-dimensional knapsack problems. The results obtained by the BSGO algorithm are compared with other binary optimization algorithms. Experimental results reveal the superiority of the BSGO algorithm in achieving a high quality of solutions over different algorithms and prove that it is one of the best finding algorithms especially in high-dimensional cases.

1. Introduction

The 0-1 knapsack problem (KP) may be defined mathematically as,

$$\min \sum_{i=1}^{N} \boldsymbol{p_i x_i} \tag{1}$$

subject to

$$\sum_{i=1}^{N} \boldsymbol{w_i x_i} \leq \boldsymbol{C}, \qquad \boldsymbol{x_i} \in \{\boldsymbol{0}, \boldsymbol{1}\} \tag{2}$$

where N is the set of items are contained by KP, each item has a weight w_i and a profit p_i. C is the maximum weight capacity of knapsack and x_i is the number of copies of item i is presented in the knapsack. The 0-1 KP is an NP-hard combinatorial optimization problem. There are several methods to solve the knapsack problems and are categorized into exact algorithms and meta-heuristic algorithms. Exact algorithms such as dynamic programming and branch-and-bound can give accurate solutions. But the worst case occurs when the problem size increase. When the size of the problem increases, then the computation time increases exponentially. The meta-heuristic algorithms can give approximate solutions at reasonable times compared to exact algorithms (Martello et al., 2000). From a few decades, meta-heuristic algorithms play significant roles in solving optimization problems that imitate natural phenomena. The meta-heuristic algorithms in avoiding local optimal solutions generate multiple solutions at each run, which help to produce optimal global solutions quickly without the needs of gradient methods. Extensive works have been carried out based on meta-heuristic algorithms in recent decades to solve 0-1 KP. A modified version of ant colony optimization (ACO) is proposed to solve 0-1 KP (Shi, 2006). Particle swarm optimization (PSO) is applied to solve multidimensional KP (Kong & Tian, 2006). Improved hybrid adaptive genetic algorithm (IHAGA) (Ma & Wan, 2011) is applied to solve KP. A novel global harmony search algorithm (Zou et al., 2011), a hybrid quantum-inspired harmony search algorithm (Layeb, 2013) is applied to solve 0-1 KPs. An artificial glow-worm

* Corresponding author.
E-mail address: animanaik@klh.edu.in (A. Naik)

doi: 10.5267/j.dsl.2021.8.004

swarm optimization algorithm is applied for solving 0-1 KP (Gong et al., 2011). A novel quantum-inspired cuckoo search (Layeb, 2011), discrete binary version of the cuckoo search algorithm (Gherboudj, Layeb & Chikhi, 2012) and, an improved hybrid encoding cuckoo search algorithm (Feng, Jia & He, 2014) are proposed to solve 0-1 knapsack problems. A modified binary PSO algorithm is proposed to solve KP (Bansal & Deep, 2012). The amoeboid organism algorithm is applied to 0-1 KP (Zhang et al., 2013). Shuffled frog leaping algorithm is also applied to solve 0-1 KP (Bhattacharjee & Sarmah, 2014). An ant colony optimization algorithm is applied to solve KP (Changdar & Mahapatra, 2013). Chemical reaction optimization based on a greedy strategy (CROG) (Truong et al., 2015) is proposed to solve 0-1 KP. A simplified binary harmony search algorithm is proposed for large scale 0–1 knapsack problems (Kong et al., 2015). A hybrid algorithm based on tabu search and chemical reaction optimization is proposed to solve 0-1 KP (Yan et al., 2015). Greedy strategy based self-adaption ant colony algorithm is introduced for the 0-1 knapsack problem (Du & Zu, 2015). In addition, many algorithms have been prospered for solving 0-1 KP such as Cognitive discrete gravitational search algorithm(CDGSA) (Razavi & Sajedi, 2015), wind driven Optimization(WDO) (*Zhou* et al., 2017), greedy degree and expectation efficiency (Lv et al., 2016), improved monkey algorithm (IMA) (Zhou et al., 2016a), monogamous pairs genetic algorithm (MPGA) (Lim et al., 2016), hybrid greedy and particle swarm (GPSO) (Nguyen, Wang & Truong, 2016), Quantum inspired social evolution (QSE) algorithm (Pavithr, 2016), binary particle swarm optimization based on the surrogate information with proportional acceleration coefficients (Lin et al., 2016), complex-valued encoding bat algorithm (Zhou et al., 2016b), cohort intelligence (CI) algorithm (Kulkarni et al., 2017), Migrating birds optimization (MBO) algorithm (Ulker & Tongur, 2017), binary flower pollination algorithm (BFPA) (Abdel-Basset et al., 2018a), binary bat algorithm (BBA) (Rizk-Allah et al., 2018), Social-Spider Optimization(SSO) Algorithm [Zhao et al., 2018; Nguyen et al., 2017), binary monarch butterfly optimization(BMBO) (Feng et al., 2016a), Binary Dragonfly Algorithm(BDA) (Abdel-Basset at al., 2017), Binary Fisherman Search (BFS) algorithm (Cobos et al., 2016),elite opposition-flower pollination algorithm (EOFPA) (Abdel-Basset et al., 2018b), Opposition-based learning monarch butterfly optimization with Gaussian perturbation(OLMBO) (Feng et al., 2017). In respect of the importance of knapsack problem in practical applications, developing new algorithms to solve large-scale types of knapsack problem applications undoubtedly becomes a true challenge.

Social Group Optimization (SGO) algorithm is one of the new Metaheuristics algorithms that is inspired by the social behavior of a human to solve a complex problem (Satapathy & Naik, 2016; Naik et al., 2018). The SGO algorithm has solved different optimization problems and proved its efficiency and robustness compared to different algorithms. After that, many researchers have explored this optimization technique for solving various complex problems in different research areas. In the medical field, social group optimization supported as an automated tool to examine skin melanoma in dermoscopy images (Dey et al., 2019). SGO with Fuzzy-Tsallis entropy helps in the segmentation of ischemic stroke lesion in brain MRI (Rajinikanth & Satapathy, 2018). In the field of electrical science, a transformer fault diagnosis model is introduced using an optimal hybrid dissolved gas analysis features subset with improved social group optimization and support vector machine classifier (Fang et al., 2018). SGO algorithm is used to develop a procedure to maximize the natural frequencies of laminated composite stiffened panels. It optimizes the quantities and sizes of the stiffener to maximize the fundamental rate of the groups under certain constraints (Tran et al., 2017). In a cloud environment, adequate resource allocation and scheduling of tasks are proposed using SGO for minimizing the makespan time and maximizing throughput (Praveen et al., 2018). The circular array synthesis is performed using social group optimization. The synthesis technique employs both non-uniform amplitudes and non-uniform spacing between the elements (Chakravarthy et al., 2018). SGO is used to solve the optimized problem of economic load dispatch in the operational planning of the power system (Madhavi & Harika, 2018). SGO has been used in gray and RGB image multi-level pre-processing (Monisha et al., 2019). The application of SGO can also be shown in the wireless sensor network (Parwekar, 2018). Like this, there are so many areas where SGO performs its use in solving problems (Leena et al., 2018; Pham et al., 2018; Mani et al., 2017; Catharin et al., 2018; Rani & Suri, 2019; Naik & Satapathy, 2021; Naik et al., 2020 ; Naik et al., 2021). But, solving combinational optimization problems using SGO have not received attention yet. Due to continuous nature of SGO, it is in its infancy for solving combinatorial optimization problems, so this is the motivation behind this study. Also solving large-scale knapsack problems have not received adequate attention yet, so this can be another motivation. Solving large-scale knapsack problems to optimality undoubtedly becomes a true challenge. Therefore, in this paper we propose a novel binary social group optimization (BSGO) algorithm to solve 0-1 knapsack problems. First, the proposed algorithm is tested on different size instances KPs from the literature, after that the proposed algorithm is effectively applied for large-size problems. The experimental results demonstrated the superiority of the proposed algorithm in achieving a high quality of solutions

The rest of the paper is arranged as follows. Section 2 gives a brief description of the SGO algorithm. The proposed BSGO algorithm is explained in detail in Section.3. Simulation and experimental results are presented in Sect. 4. Finally, section 5 gives conclusions and future work.

2. Social group optimization(SGO)

The Social Group Optimization algorithm is a meta-heuristic population-based optimization algorithm inspired by the social behavior of humans while solving complex problems. Each person has their response and action to address the complex problems efficiently in life based on their traits and skills. Thus, group solving ability is more operative compared to individual expertise in exploring and manipulating individuals' characters in the group to solve a specific problem. Based on this concept, the SGO optimization technique is developed. For a detailed description of the SGO algorithm, anyone can follow the paper (Satapathy & Naik, 2016; Naik et al., 2018).

In the SGO algorithm, the definitions and concepts are considered as follows:

a) Each person represents a candidate solution empowered with some information that has an ability level to solve a problem.
b) The information (knowledge) is affected by different traits of each person with the best person's traits influence.
c) The best person signifies the best solution.
d) The best solution is to spread knowledge amongst all persons.
e) The population is corresponding to 'fitness' that is used to improve the knowledge level of entire members in the group.
f) The SGO algorithm has two stages, namely improving and acquiring stages.
g) In the group, each person's knowledge (solution) level is improved based on the best person's influence in the improving phase.
h) The best candidate solution is the one having the highest knowledge level and ability to solve the problem under concern.
i) The mutual interaction between persons in the group through the acquiring phase is improved each person's knowledge.

The preceding definitions are applied along with the assumption that the social group contains *M* persons and each person P_i is defined by $P_i = (p_{i1}, p_{i2}, p_{i3}, \dots \dots, p_{iN})$, where *i*=1,2,3,…., *M* and *N* refer to the allocated traits number to an individual to determine the individual's dimensions. In each social group, through the improving phase, the best candidate solution (*gbest*) tries to spread knowledge between all persons to assist others in improving their knowledge within the group. The best candidate solution at generation *g* for solving a minimization problem is given by:

$$[\text{minvalue, index}]=\min\{f(P_i), i = 1,2,3, \dots \dots \dots \dots, M\} \quad (3)$$
$$\text{gbest}=P(\text{index},:)$$

where f refers to the corresponding fitness value, each candidate solution gains knowledge from the group *gbest* in the improving phase. Each candidate update can be intended using the following algorithm.

Algorithm 1: The SGO Improving Phase

For i= 1: M
For j=1: N
$$Pnew_{ij} = C * P_{ij} + R * (\text{gbest}_j - P_{ij})$$
End for
End for
Accept *Pnew* if it offers superior fitness than *P*

where, C is the self-introspection parameter having 0<C< 1, whereas R is a random number specified by $R \sim U(0,1)$.

Afterward, a candidate solution of the social group interrelates with '*gbest'* of that group besides its random interaction with other candidate solutions of the group to gain knowledge during the acquiring phase. A person acquires new knowledge if the other person has more knowledge. The paramount knowledgeable person having '*gbest*' has the greatest motivation for others to learn from him/her. A person consistently will attain something new from other persons in the group, if they have more knowledge than him/her. The acquiring phase is stated as follows:

$$[\text{minvalue, index}]=\min\{f(P_i), i = 1,2,3, \dots \dots \dots \dots, M\} \quad (4)$$
$$\text{gbest}=P(\text{index},:)$$

where, P_i's are updated values at the end of improving phase. The acquiring stage algorithm is given as follows.

Algorithm 2: The SGO Acquiring Phase

For i=1: M
Chose randomly one individual P_v, where $i \neq v$
If $f(P_i) < f(P_v)$
For j=1: N
$$Pnew_{i,j} = P_{i,j} + R_1 * (P_{i,j} - P_{v,j}) + R_2 * (\text{gbest}_j - P_{i,j})$$
End for
Else
For j=1: N
$$Pnew_{i,:} = P_{i,:} + R_1 * (P_{v,:} - P_{i,:}) + R_2 * (\text{gbest}_j - P_{ij})$$
End for
End If
Accept *Pnew* if it offers superior fitness function value
End for

where, R_1 and R_2 are two independent random numbers, where each of them is $\sim U(0,1)$. These random numbers are used to affect the algorithm stochastic nature as exposed in the acquiring phase.

3. Description of proposed BSGO algorithm

The standard SGO algorithm is designed to deal with continuous problems. The knapsack problem is discrete as the problem arises in selecting or not selecting a particular item to be packed in the knapsack. One corresponds to the item being selected in the knapsack, and zero corresponds to the item being rejected out the knapsack. So to solve the 0-1 knapsack problem using the SGO algorithm, the continuous variables have to be transformed into discrete variables. For that, the sigmoid function is adapted to make SGO deal with the knapsack problem. It takes the real values generated from the standard SGO and converts them into 1(one) or 0(zero). The following steps will clarify the main concepts of BSGO. A framework of the BSGO algorithm is given in Fig.1.

Initialization

In this step, the M initial population of N dimension is generated by following algorithm 3, where N is the number of available items. Each person P_i in the population represents a solution which provides the numbers of the item available in KP.

Algorithm 3: Initialization function

```
function [P]=initialization(M,N)
for i=1 to M
    for j=1 to N
        if rand < 0.5
            Pij =0
        else
            Pij=1
        end if
    end for
end for
end function
```

Transformation

In BSGO, a transformation function is used for mapping the real-valued solutions into binary ones using following algorithm 4. To convert real-valued to a binary one, the sigmoid function is employed.

Algorithm 4. Transformation function

```
function [x]=transformation(x)
 s=1/(1+exp(-x))
    if s<rand
       x=0
    else
       x=1
    end if
end function
```

Fitness calculation

In the BSGO algorithm, the fitness function is employed for evaluating the knowledge of the person. The fitness is computed as the summation of the profits for all items selected in the knapsack. The 0-1 knapsack problem is a constrained optimization problem. So handling constrain is an integral part of calculating the fitness value of a solution because the infeasible solution may occur. In 0-1 knapsack problem, a solution will be an infeasible solution when selecting an item with maximum fitness that contradicts the constraint $\sum_{i=1}^{N} w_i x_i \leq C$. These infeasible solutions will be selected because of their maximum fitness. The fitness of a solution vector is calculated using the following algorithm 5. Using this fitness function, we determine the fitness value (maximum profit) of the solution vector if it is feasible. If a solution vector is not feasible, then first, we make infeasible to feasible and then find fitness value. To do this, we have used a repair operator (RO) function and penalty calculation (PC) function. The repair operator is a greedy type heuristic to compute the profit per unit that is defined as:

$$RO_i = \frac{P_i}{w_i} \tag{5}$$

The RO of a solution vector is calculated using the following algorithm 6, and the PC of a solution vector is computed using algorithm 7.

Algorithm 5. Fitness function

```
function [x, fitness]=fitnessfunction(x, data,capacity)
% N represents the number of items
weight=transpose(data(:,1));
profit=transpose(data(:,2));
ro= ROfunction(x, profit, weight)
 [r s]=sort(ro,'ascend');

penalty= PCfunction(x, weight, profit, capacity)
index=1;
while(penalty<0 && index<=N)
   x(s(index))=0;
   index=index+1;
   penalty=PCfunction(x, weight, profit, capacity)
end while
x;
sum=0;
    for j=1:N
        sum=sum+profit(j)*x(j);
    end
    fitness=sum;

end function
```

Algorithm 6. Finding RO

```
function [ro]=ROfunction(x, profit, weight)
% N represents the number of items
for i=1 to N
      if (x(i)≠0)
            ro(i)=profit(i)/weight(i);
      else
            ro(i)=0;
      end if
 end for
end function
```

Algorithm 7. PC function

```
function [pc]= PCfunction(x, weight, profit, capacity)
% N represents the number of items

    sum=0;
    for j=1 to N
          sum=sum+weight(j)*x(j);
    end for
    totalweight=sum;
    if  (totalweight>capacity)
          pc=capacity-totalweight;
    else
        pc=totalweight;
    end if

end function
```

Algorithm 8: The proposed BSGO

Define self introspection parameter C, Max_iter=Maximum iteration , M=pop_size, N=dimension
Initialize population P using algorithm 3
Evaluate each person P_i in the population using algorithm 5
Find the best solution f^* and best person gbest
t=1
While (t< Max_iter)
For each person P_i in the population
    Find newP_i using improving phase
    Transform real-valued to binary one using algorithm 4
    Evaluate using algorithm 5
    If newP_i is better than P_i then replace it and update population
End for
Find the best solution f^* and best person gbest
For each person P_i in the population
    Find newP_i using acquiring phase
    Transform real-valued to binary one using algorithm 4
    Evaluate using algorithm 5
    If newP_i is better than P_i then replace it and update population
End for
Find the best solution f^* and best person gbest
t=t+1
end while
return the best solution f^*

4. Simulation and experimental result

In this section, a comprehensive comparison study is conducted to investigate the optimization capability of the BSGO algorithm along with the convergence speed and numerical stability through two experiments. In the first experiment, 10 small-scale knapsack problems are considered. These small-scale problems are listed in Table A1 of Appendix A. In the second experiment, 30 large-scale knapsack problems are discussed. Information regarding these problems is given in Table A2 of Appendix A.

"*For comparing the speed of algorithms, number of iterations or generations cannot be accepted as a time measure, since algorithms perform different amounts of works in their inner loops, and also they have different population sizes. Hence, we choose max_FEs (maximum number of function evaluations) as a measure of computation time instead of generations or iterations"* (Satapathy et al., 2016) Instead of generations or iterations, max_FEs have opted as a measure of computation time. Due to the stochastic nature of meta-heuristic algorithms, we have taken different independent runs (with varying seeds of random number generator) of each algorithm and found out best, mean, median, worse, and standard deviation(SD) of function values and put in tables of the experiment. For comparing the performance of algorithms, different tests are conducted on experimental results.

4.1. Experiment 1

In this experiment the proposed BSGO algorithm is compared with different algorithms, referred from a separate research paper. For ACO (Ant Colony Optimization), SA(Simulated Annealing, GA(Genetic Algorithm), NGHS, SSO(Social-Spider Optimization) algorithm results are reported from paper (Zhao & Zhou, 2018). For CI (Cohort Intelligence), results are reported from (Kulkarni et al., 2017). For CWDO (complex-valued encoding wind-driven optimization) the results are considered from (Zhou et al., 2017). For IHS, the result is reported from (Zou et al., 2011). For PA (Parallel Approach based simulated annealing algorithm) the results are reported from (Sonuc et al., 2016). For BFSP (binary fisherman search procedure), MDSFL (modified discrete shuffled frog-leaping algorithm), SBHS(simplified binary harmony search algorithm), SLC(soccer league competition algorithm) the results are reported from paper (Cobos et al., 2016). For BDA (binary dragonfly algorithm) the results are reported from (Abdel-Basset et al., 2017). For ABHS (adaptive binary harmony search (Wang et al., 2013), IBBA-RSS (injective binary bat algorithm based rough set scheme), the results are considered from their paper. The SGO algorithm is converted to binary version algorithm by using algorithm 3 and 4 of section 3. Using algorithm 3, the SGO algorithm is initialised, and using algorithm 4, a real-valued individual is transformed to binary ones. Using algorithm 5, the fitness value of algorithm is evaluated. For comparison of the performance of algorithms, 10 small-scale 0-1knapsack problems are considered. For the SGO, the experiment is conducted in MATLAB 2016a on an Intel Core i5, 8 GB memory laptop in Windows 10 environment.

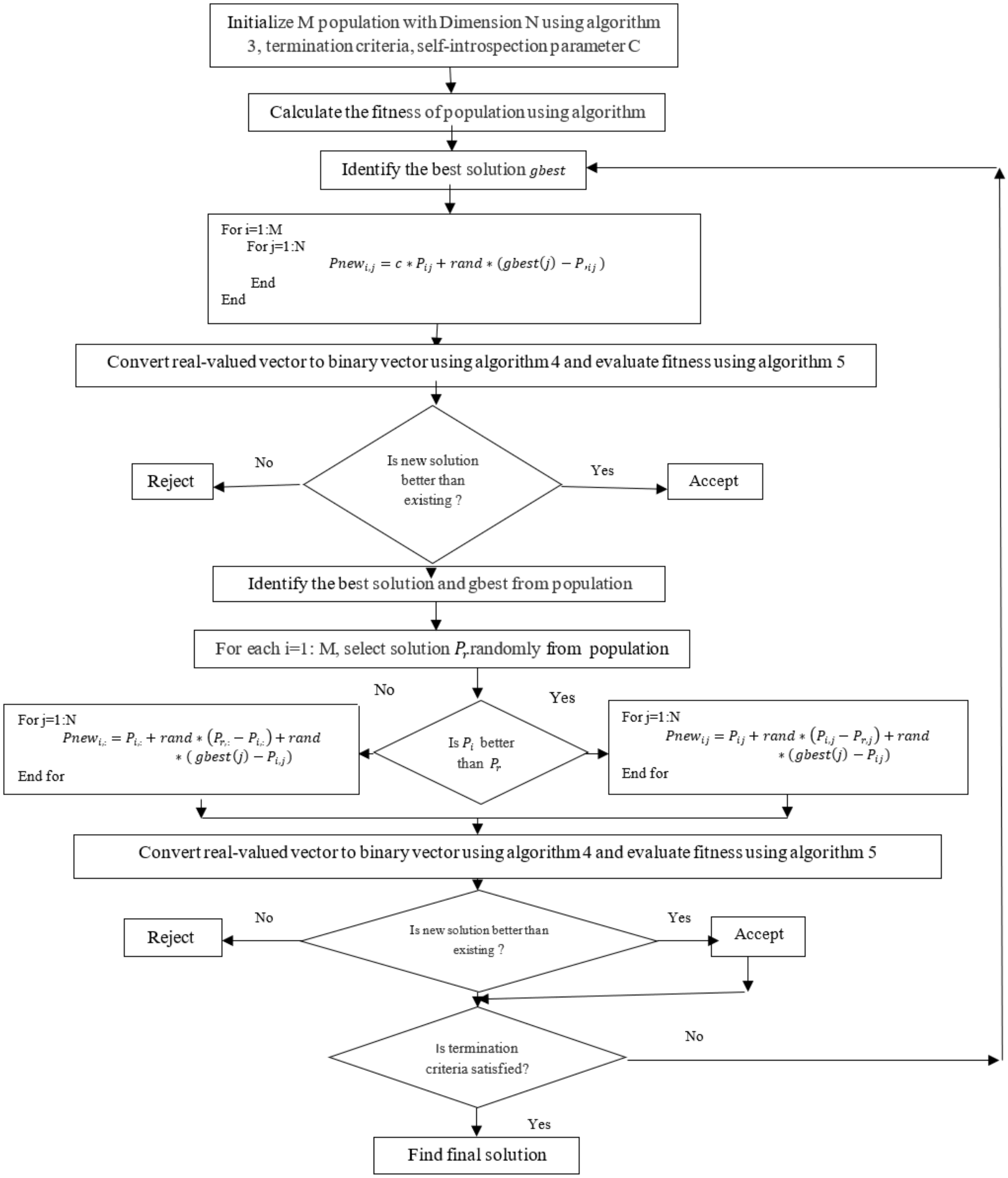


Fig. 1. Framework of BSGO algorithm

For each problem, the statistical results in terms of best value, mean value, median value, worse value and corresponding standard deviation are reported in Tables 1. The best results are highlighted in bold faces. 'NA' stands for that particular value is not available. The average error (AE) of the best profit, mean profit and worse profit for all these algorithms for all 10 0-1 KP is given Table 2. The AE can be defined mathematically as:

$$\text{AE}=\frac{1}{n}\sum_{i=1}^{n}\frac{\text{profit}_{opt}-\text{profit}_{i}}{\text{profit}_{opt}} * 100 \tag{6}$$

where n is the number of problems, $profit_{opt}$ is the optimal solution for the problem, $profit_i$ is the profit obtained by the algorithm for the best or mean or worse profit.

Table 1

Results of 10 low dimensional 0-1 KP problems, referred from different research paper

Algorithm	P1, dim=10, optimal value=295					P2, dim=20 ,optimal value=1024				
	Best	Median	Worse	Mean	Std	Best	Median	Worse	Mean	Std
ACO	295	NA	295	295	0	1024	NA	1024	1024	0
SA	295	NA	293	294.8	0.6102	1024	NA	1024	1024	0
GA	295	NA	295	295	0	1024	NA	1018	1023.8	1.0954
NGHS	295	295	295	295	0	1024	1024	1024	1024	0
SSO	295	NA	295	295	0	1024	NA	1024	1024	0
CWDO	295	NA	295	295	0	1024	NA	1024	1024	0
CI	295	NA	260	267.46	NA	1024	NA	1009	1020.55	NA
BDA	295	295	295	295	0	1024	1024	1024	1024	0
IHS	295	295	288	294.78	16	1024	1024	1024	1024	0
PA	295	295	295	295	0	1024	1024	1024	1024	0
BFSP	295	295	295	295	0	1024	1024	1024	1024	0
MDSFL	295	295	294	294.66	0.4737	1024	1024	1024	1024	0
SBHS	295	295	295	295	0	1024	1024	1018	1023.88	0.84
SLC	295	295	295	295	0	1024	1024	1024	1024	0
DHS	SLC	295	295	295	0	1024	1024	1024	1024	1024
ABHS	SLC	295	295	295	0	1024	1024	1024	1024	1024
IBBA-RSS	SLC	295	295	295	0	1024	1024	1024	1024	1024
SGO	295	295	295	295	0	1024	1024	1024	1024	0
Algorithm	P3, dim= 4,optimal value=35					P4, dim= 4,optimal value=23				
	Best	Median	Worse	Mean	Std	Best	Median	Worse	Mean	Std
ACO	35	NA	35	35	0	23	NA	23	23	0
SA	35	NA	35	35	0	23	NA	23	23	0
GA	35	NA	35	35	0	23	NA	23	23	0
NGHS	35	35	35	35	0	23	23	23	23	0
SSO	35	NA	35	35	0	23	NA	23	23	0
CWDO	35	NA	35	35	0	23	NA	23	23	0
CI	35	NA	28	34.55	NA	23	NA	16	22.6	0.64
BDA	35	35	35	35	0	23	23	23	23	0
IHS	35	35	27	34.58	1.68	23	23	23	23	0
PA	35	35	35	35	0	23	23	23	23	0
BFSP	35	35	35	35	0	23	23	23	23	0
MDSFL	35	35	35	35	0	23	23	23	23	0
SBHS	35	35	35	35	0	23	23	23	23	0
SLC	35	35	35	35	0	23	23	23	23	0
DHS	35	35	35	35	0	23	23	23	23	0
ABHS	35	35	35	35	0	23	23	23	23	0
IBBA-RSS	35	35	35	35	0	23	23	23	23	0
SGO	35	35	35	35	0	23	23	23	23	0
Algorithm	P5, dim= 15,optimal value=481.0694					P6, dim= 10,optimal value=52				
	Best	Median	Worse	Mean	Std	Best	Median	Worse	Mean	Std
ACO	481.0694	NA	481.0694	481.0694	0	52	NA	52	52	0
SA	481.0694	NA	437.9345	453.7506	21.1418	52	NA	50	51.8666	0.5474
GA	481.0694	NA	481.0694	481.0694	0	52	NA	52	52	0
NGHS	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0
SSO	481.0694	NA	481.0694	481.0694	0	52	NA	52	52	0
CWDO	481.0694	NA	481.0694	481.0694	0	52	NA	52	52	0
BDA	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0
CI	481.0694	NA	412.69	449.98	10.68	51	NA	49	50.73	0.66
IHS	481.0694	NA	437.93	478.48	10.35	50	NA	44	49.2	1.85
PA	481.0694	481.0694	437.9345	472.4424	17.25	52	52	52	52	0
BFSP	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0
MDSFL	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0
SBHS	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0
SLC	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0
DHS	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0
ABHS	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0
IBBA-RSS	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0
SGO	481.0694	481.0694	481.0694	481.0694	0	52	52	52	52	0

Table 1
Results of 10 low dimensional 0-1 KP problems, referred from different research paper (Continued)

Algorithm	P7, dim= 7,optimal value=107					P8, dim=23,optimal value=9767				
	Best	Median	Worse	Mean	Std	Best	Median	Worse	Mean	Std
ACO	107	NA	107	107	0	9765	NA	9734	9744.867	6.5796
SA	105	NA	81	84.2	7.6762	9767	NA	9762	9765.467	2.2086
GA	107	NA	105	106.9	0.3651	9767	NA	9767	9767	0
NGHS	107	107	107	107	0	9767	9767	9767	9767	0
SSO	107	NA	107	107	0	9767	NA	9767	9767	0
CWDO	107	NA	107	107	0	9767	NA	9767	9767	0
CI	105	NA	79	86.6	2.99	9759	NA	9710	9753.33	11.5
BDA	107	107	107	107	0	9767	9767	9767	9767	0
IHS	107	NA	93	103.98	4.48	9767	NA	9767	9767	0
PA	107	105	81	100.5	7.53	9767	9758	9754	9755.4	0
BFSP	107	107	107	107	0	9767	9767	9767	9767	0
MDSFL	107	107	107	107	0	9767	9767	9767	9767	0
SBHS	107	107	107	107	0	9767	9767	9767	9767	0
SLC	107	107	105	106.96	0.28	9767	9767	9767	9767	0
DHS	107	107	107	107	0	9767	9767	9767	9767	0
ABHS	107	107	107	107	0	9767	9767	9767	9767	0
IBBA-RSS	107	107	107	107	0	9767	9767	9767	9767	0
SGO	107	107	107	107	0	9767	9767	9767	9767	0
	P9,dim=5,optimal value=130					P10,dim=20, optimal value=1025				
	Best	Median	Worse	Mean	Std	Best	Median	Worse	Mean	Std
ACO	130	NA	130	130	0	1025	NA	1025	1025	0
SA	130	NA	130	130	0	1025	NA	1025	1025	0
GA	130	NA	130	130	0	1025	NA	1019	1024.4	1.8307
NGHS	130	130	130	130	0	1025	1025	1025	1025	0
SSO	130	NA	130	130	0	1025	NA	1025	1025	0
CWDO	130	NA	130	130	0	1025	NA	1025	1025	0
CI	130	N1	106	124.6	2.89	1025	NA	892	997.7	18.6
BDA	130	130	130	130	0	1025	1025	1025	1025	0
IHS	130	NA	130	130	0	1025	NA	1025	1025	0
PA	130	130	130	130	0	1025	1025	1025	1025	0
BFSP	130	130	130	130	0	1025	1025	1025	1025	0
MDSFL	130	130	130	130	0	1025	1025	1025	1025	0
SBHS	130	130	130	130	0	1025	2025	1019	1024.76	1.1758
SLC	130	130	130	130	0	1025	1025	1025	1025	0
DHS	130	130	130	130	0	1025	1025	1025	1025	0
ABHS	130	130	130	130	0	1025	1025	1025	1025	0
IBBA-RSS	130	130	130	130	0	1025	1025	1025	1025	0
SGO	130	130	130	130	0	1025	1025	1025	1025	0

Table 2
AE result of algorithms from Table 1

Algorithm	**Best**	**Mean**	**Worse**
ACO	0.0020	0.0227	0.0338
SA	0.1869	2.7327	3.7841
GA	0	0.0172	0.3040
NGHS	0	0	0
SSO	0	0	0
CWDO	0	0	0
CI	0.3874	4.7625	15.1552
BDA	0	0	0
IHS	0.3846	1.2070	6.2666
PA	0	0.7987	3.3399
BFSP	0	0	0
MDSFL	0	0.3961	0.4185
SBHS	0	0.0035	0.2050
SLC	0	0.0037	0.1869
DHS	0	0	0
ABHS	0	0	0
IBBA-RSS	0	0	0
SGO	0	0	0

From Table 2, we conclude that for all the algorithms except ACO, SA, CI, IHS, AE of best profit are zero. Similarly, except ACO, SA, GA, CI, IHS, PA, MDSFL, SBHS, SLC for all algorithms average error of mean profit is zero. NGHS, SSO, CWDO, BDA, BFSP, DHS, ABHS, IBBA-RSS, SGO algorithms have zero value for an average error of the worst profit. The obtained results from Table 1 and Table 2 we conclude that the proposed BSGO algorithm is competitive with NGHS, SSO, CWDO, BDA, BFSP, HHS, ABHS and IBBA-RSS algorithm and outperforms than ACO, SA, GA, CI, IHS, MDSFL, SBHS and SLC algorithm.

4.2. Experiment 2

In this experiment, to further prove the proficiency of the proposed BSGO algorithm, this algorithm is compared with a binary version of twelve well-known meta-heuristic optimization algorithms including ABC(Artificial Bee Colony), BA(Bat Algorithm), BH(Black Hole), CS(Cuckoo Search), DE(Differential Evolution), FLA(Frog Leaping Algorithm), GWO(Gray Wolf Optimization), HS(Harmony Search), PSO(Particle Swarm Optimization), TLBO(Teaching Learning Based Optimization), FPA(Flower Pollination Algorithm), FFA (FireFly Algorithm). We have converted these meta-heuristic algorithms to the binary version algorithm by using algorithm 3 and 4 of section 3. Using algorithm 3, we initialize each algorithm, and using algorithm 4, we transform a real-valued individual to binary ones, and using algorithm 5, we have evaluated the fitness value of each individual of algorithms. For comparison of the performance of algorithms, 30 large-scale 0-1 knapsack problems are introduced. These 30 large-scale 0-1 knapsack problems are of three categories: uncorrelated, weakly correlated and strongly correlated. Three kinds of correlation characteristics of profit and weight such as uncorrelated, weakly correlated and strongly correlated, are shown in Table A2 of Appendix A. For example, the nature of uncorrelated items, weakly correlated items and strongly correlated items of 200 dimensions are depicted in Fig. 2. Here function rand(a, b) returns a integer drawn from uniform distribution in interval [a, b], where a=10 and b=30. Here each group includes 10 large-scale 0-1 KPs problems generated randomly whose dimension is 100, 200, 300, 500, 700, 1000, 1500, 2000, 3000 and 5000 respectively. These 30 0-1 KPs are represented by ucdata1, ucdata2,...., ucdata10, wcdata1, wcdata2,...., wcdata10, scdata1, scdata2 ,..., scdata10 respectively. Additionally, the maximum capacity of knapsack equals 0.75 times of the sum of the weights. It is worth noting that these datasets are created only once using a random generator and kept constant for all experiments.

To reflect fairness of the comparison process, common control parameters such as maximum number function evaluation is deemed to be the same for all algorithms as 10,000. For that, we have considered the maximum number of iteration is 200, and population size is 25 for ABC, CS, SGO, TLBO, and for other algorithms, population size is 50 (the function is evaluated twice in each iteration for ABC, CS, SGO, and TLBO algorithm). The other algorithm-specific parameters are given in Table 3. The experiments are conducted in MATLAB 2016a on an Intel Core i5, 8 GB memory laptop in Windows 10 environment

For each problem, algorithms are run 30 times, starting from different populations randomly generated. Statistical results in terms of best value, mean value, median value, worse value and corresponding standard deviation are reported in Tables. The comparative results achieved by 13 algorithms ABC, BA, BH, CS, DE, FLA, GWO, HS, PSO, TLBO, FPA, FFA and SGO on uncorrelated, weakly correlated and strongly correlated 0-1 KPs are presented in Tables 4-6 respectively. The best results on each problem among all 13 algorithms are bolded in each table. As optimal profits are unknown AE (average error) of the best profit and/or mean profit and/or worse profit of results are not calculated. The convergence characteristics of all the algorithms in the first 50 iterations for all the uncorrelated, weakly correlated and strongly correlated 0-1 large-scale KP problems are depicted in Fig.3, Fig.4 and Fig.5 respectively.

Table 3
The setting of parameters for algorithms

Algorithm	Function Evaluation in each iteration	Other parameters
ABC	2	Number of food sources=25, maximum search time=100
PSO	1	Fully connected, cognitive parameter=2, social parameter=2, inertia of weight =0.5
DE	1	F=0.9, CR=0.1
CS	2	$P_a = 0.25$
GWO	1	The a=2-2*(iter/$iter_{max}$)
FFA	1	alpha=0.2, gamma=1.0, delta=0.97.
FLA	1	M=5, N=10
FPA	1	Switching probability P=0.8,
BA	1	alpha=0.9,gamma=0.9
TLBO	2	There is no such parameter to set value
SGO	2	There is only one parameter C called a self-introspection factor. Value of C is 0.2
BH	1	There is no parameter
HS	1	hmcr=0.9, par=0.3,bw=0.01

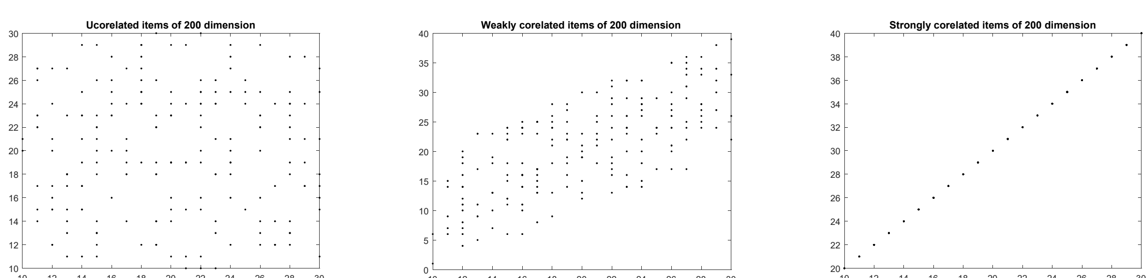


Fig. 2. Nature of uncorrelated items, weakly correlated items and strongly correlated items of 200

Table 4
Results of UC data high dimensional 01knapsack problem , 30 different runs, 200 generation

Algorithm	UCdata1, dim=100						UCdata2, dim=200					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	1632	1627	1618	1.6078e+03	6.1008	–	3278	3232	3199	3.2308e+03	24.5700	–
BA	1641	1611	1581	1.6098e+03	18.4585	–	3240	3148	3109	3.1481e+03	26.5056	–
BH	1608	1563	1550	1.5662e+03	13.7883	–	3126	3078	3052	3.0793e+03	17.8641	–
CS	1605	1592	1574	1590	8.5258	–	3203	3133	3108	3.1354e+03	22.9895	–
DE	1613	1578	1564	1.5792e+03	12.4781	–	3167	3123	3087	3.1225	20.4413	–
GWO	1604	1568	1547	1.5662e+03	13.1282	–	3174	3091	3056	3.0935e+03	24.5746	–
HS	1594	1576	1561	1.5750e+03	9.0038	–	3152	3114	3083	3.1122e+03	18.6357	–
PSO	1618	1581	1561	1.5831e+03	14.2101	–	3169	3123	3083	3.1219e+03	21.9018	–
TLBO	1607	1584	1570	1.5829e+03	8.2876	–	3165	3118	3089	3.1205e+03	20.3618	–
SGO	**1651**	**1612**	**1573**	**1.6108e+03**	18.9143		**3321**	**3232**	**3117**	**3.2231e+03**	57.7281	
FPA	1618	1579	1561	1.5786e+03	12.5574	–	3151	3104	3086	3.1075e+03	17.4726	–
FFA	1633	1614	1604	1.6161e+03	7.7029	–	3218	3179	3157	3.1792e+03	15.4393	–
Algorithm	UCdata3, dim=300						UCdata4, dim=500					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	4693	4592	4525	4.5931e+03	45.4201	–	7225	7049	6896	7.0385E+03	84.6428	–
BA	4764	4582	4537	4.5875e+03	45.4796	–	7669	7558	7454	7.5456E+03	58.4881	–
BH	4586	4471	4431	4.4826e+03	40.1561	–	7660	7380	7257	7.3950E+03	90.8084	–
CS	4615	4563	4519	4.5621e+03	28.1061	–	7662	7521	7456	7.5274E+03	55.9819	–
DE	4595	4548	4493	4.5418e+03	26.7681	–	7592	7480	7410	7.4774E+03	44.7811	–
GWO	4552	4493	4443	4.4933	29.6501	–	7544	7422	7337	7.4241E+03	52.1320	–
HS	4636	4533	4487	4.5376e+03	30.7745	–	7723	7454	7388	7.4713E+03	73.9018	–
PSO	4619	4569	4514	4.5658e+03	28.8915	–	7780	7531	7401	7.5436E+03	89.1398	–
TLBO	4628	4543	4506	4.5477e+03	24.9876	–	7637	7469	7413	7483	57.5410	–
SGO	**4871**	**4807**	**4697**	**4.7996e+03**	48.5100		**8238**	**8081**	**7844**	**8.0689E+03**	102.8319	
FPA	4601	4523	4485	4.5260e+03	28.8952	–	7586	7455	7384	7.4661E+03	53.2009	–
FFA	4664	4637	4614	4.6355E+03	15.4043	–	7700	7645	7616	7.6532E+03	27.5477	–
Algorithm	UCdata5, dim=700						UCdata6, dim=1000					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	9711	9377	9192	9.3811e+03	116.4466	–	12889	12496	12349	1.2532e+04	142.7659	–
BA	10628	10388	10337	1.0413e+04	69.7836	–	14923	14613	14506	1.4627e+04	114.2712	–
BH	10413	10220	10156	1.0237e+04	69.5761	–	14572	14359	14212	1.4362e+04	95.2886	–
CS	10586	10364	10315	1.0401e+04	77.2193	–	14692	14546	14488	1.4561e+04	58.2545	–
DE	10545	10342	10271	1.0348e+04	62.4032	–	14649	14511	14394	14500	69.6459	–
GWO	10389	10231	10164	1.0239e+04	59.1272	–	14773	14402	14312	1.4421e+04	98.1866	–
HS	10520	10321	10273	1.0340e+04	67.6216	–	14655	14449	14328	1.4466e+04	83.7556	–
PSO	10728	10465	10352	1.0478e+04	101.3282	–	15145	14862	14643	1.4849e+04	135.1496	–
TLBO	10541	10358	10274	1.0370e+04	65.5183	–	14751	14516	14400	1.4534e+04	97.5332	–
SGO	**11628**	**11373**	**10980**	**11339**	166.3424		**16397**	**16146**	**15878**	**1.6148e+04**	135.4209	
FPA	10620	10294	10222	1.0312e+04	89.0557	–	14691	14445	14349	1.4469e+04	92.9166	–
FFA	10718	10584	10527	1.0590e+04	39.5295	–	15148	14807	14732	1.4833e+04	94.4428	–
Algorithm	UCdata7, dim=1500						UCdata8, dim=2000					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	18247	17524	17307	1.7557e+04	176.4488	–	23314	22804	22527	2.2865e+04	212.1212	–
BA	21548	21181	21034	2.1204e+04	129.6044	–	28611	28257	28025	2.8246e+04	145.8881	–
BH	21051	20846	20707	2.0858e+04	89.5965	–	28117	27775	27591	2.7785e+04	136.9804	–
CS	21329	21134	21051	2.1150e+04	65.3697	–	28267	28072	27936	2.8082e+04	100.6582	–
DE	21364	21064	20937	2.1081e+04	114.3904	–	28240	27970	27812	2.7979e+04	117.3320	–
GWO	21148	20936	20802	2.0935e+04	84.5506	–	28238	27875	27749	2.7897e+04	122.0125	–
HS	21426	21020	20898	2.1050e+04	118.0167	–	28346	27864	27689	2.7870e+04	135.0396	–
PSO	22096	21718	21332	2.1713e+04	164.9957	–	29693	28954	28678	2.8989e+04	241.1206	–
TLBO	21447	21082	20949	2.1109e+04	113.0351	–	28323	28028	27788	2.8024e+04	157.4332	–
SGO	**24397**	**23928**	**23270**	**2.3876e+04**	298.9341		**32687**	**32160**	**31639**	**3.2137e+04**	264.3118	
FPA	21172	20981	20879	2.0988e+04	70.0304	–	28361	27878	27756	2.7907e+04	146.5758	–
FFA	21700	21429	21325	2.1451e+04	96.0837	–	28607	28392	28281	2.8398e+04	84.7697	–
Algorithm	UCdata9, dim=3000						UCdata10, dim=5000					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	33912	33131	32872	3.3191e+04	297.8634	–	53731	53104	52701	5.3157e+04	271.1119	–
BA	42658	41803	41498	4.1926e+04	330.8225	–	69429	68643	68110	6.8646e+04	366.4497	–
BH	41652	41139	40895	4.1160e+04	188.6589	–	67868	67240	67240	6.7300e+04	259.4373	–
CS	41727	41518	41375	4.1525e+04	95.3210	–	68475	67823	67568	6.7846e+04	216.4741	–
DE	41768	41472	41260	4.1484e+04	130.8128	–	67993	67678	67448	6.7689e+04	152.2831	–
GWO	41714	41300	41154	4.1319e+04	143.1373	–	67887	67429	67248	6.7461e+04	138.4048	–
HS	41975	41290	41092	4.1314e+04	187.1917	–	67990	67347	67087	6.7381e+04	196.6028	–
PSO	43640	43162	42790	4.3204e+04	203.8988	–	71990	71010	70500	7.0997e+04	361.0556	–
TLBO	42018	41639	41373	4.1604e+04	168.7509	–	68362	67804	67529	6.7829e+04	191.4581	–
SGO	**48920**	**48107**	**46903**	**4.8112e+04**	443.9315		**80377**	**79644**	**77802**	**7.9497e+04**	684.2918	
FPA	41836	41370	41219	4.1407e+04	135.5978	–	68188	67519	67308	6.7577e+04	229.3645	–
FFA	44658	43803	43498	4.3926e+04	330.8225	–	70429	69643	69110	6.9646e+04	366.4497	–

" –", "+" and "≈" denote that performance of algorithms are worse, better and similar to BSGO respectively

Table 5
Results of WC data high dimensional 01kanshap problem , 30 different runs, 200 generation

Algorithm	WCdata1, dim=100						WCdata2, dim=200					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	1651	1639	1631	1.6397e+03	5.4351	–	3364	3303	3286	3.3096e+03	19.2291	–
BA	1638	1612	1591	1.6107e+03	12.4428	–	3304	3254	3227	3.2581e+03	20.8143	–
BH	1604	1581	1570	1.5813e+03	8.8518	–	3233	3201	3176	3.2009e+03	16.4437	–
CS	1614	1597	1587	1.5978e+03	7.9381	–	3279	3244	3225	3.2427e+03	14.3429	–
DE	1609	1590	1582	1.5929e+03	7.0844	–	3265	3227	3214	3.2319e+03	14.6686	–
GWO	1606	1584	1571	1.5843e+03	9.6538	–	3249	3204	3181	3.2058e+03	17.1325	–
HS	1625	1587	1577	1.5885e+03	9.8076	–	3280	3215	3197	3.2196e+03	20.4984	–
PSO	1618	1597	1583	1.5971e+03	8.2334	–	3264	3237	3205	3.2363e+03	17.2230	–
TLBO	1610	1591	1584	1.5929e+03	7.8228	–	3291	3230	3208	3.2299e+03	16.4290	–
SGO	**1641**	**1609**	**1587**	**1.6086e+03**	1587		**3375**	**3321**	**3259**	**3.3215e+03**	32.4098	
FPA	1622	1590	1578	1.5918e+03	9.7651	–	3276	3229	3206	3.2307e+03	18.3343	–
FFA	1627	1615	1609	1.6158e+03	4.1992	–	3328	3282	3265	3.2854e+03	15.7841	–
Algorithm	WCdata3, dim=300						WCdata4, dim=500					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	4615	4535	4487	4.5352e+03	32.1259	–	7243	7083	6910	7071	98.5719	–
BA	4624	4551	4487	4.5474e+03	31.1012	–	7742	7579	7483	7.5901e+03	68.0767	–
BH	4494	4444	4399	4.4443e+03	25.4639	–	7519	7374	7302	7.3868e+03	60.4635	–
CS	4569	4520	4489	4.5236e+03	22.4171	–	7626	7533	7479	7.5398e+03	36.2197	–
DE	4537	4505	4466	4.5035e+03	21.6568	–	7673	7526	7426	7.5189e+03	64.6659	–
GWO	4517	4450	4414	4.4555e+03	28.0132	–	7629	7409	7347	7.4294e+03	61.4528	–
HS	4564	4492	4445	4.4905e+03	30.3198	–	7612	7518	7429	7.5071e+03	51.8442	–
PSO	4608	4504	4452	4.5046e+03	34.6008	–	7695	7559	7449	7.5639e+03	64.7003	–
TLBO	4534	4501	4465	4.5005e+03	20.0615	–	7715	7495	7445	7.5061e+03	60.6626	–
SGO	**4811**	**4702**	**4589**	**4.6923e+03**	55.3073		**8104**	**7868**	**7706**	**7.8660e+03**	101.7087	
FPA	4560	4487	4442	4.4902e+03	29.7303	–	7585	7479	7415	7.4848e+03	47.6161	–
FFA	4656	4578	4543	4.5818e+03	22.3243	–	7829	7694	7643	7.7009e+03	43.0672	–
Algorithm	WCdata5, dim=700						WCdata6, dim=1000					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	9281	8974	8817	8.9832e+03	116.1054	–	12846	12496	12148	12465	186.5553	–
BA	10279	10114	10007	1.0130e+04	76.9885	–	14791	14538	14437	1.4549e+04	82.8239	–
BH	10064	9860	9805	9.8783e+03	61.2498	–	14545	14271	14121	1.4269e+04	107.7963	–
CS	10296	10056	9979	1.0079e+04	80.1256	–	14620	14523	14393	1.4508e+04	65.7085	–
DE	10210	10022	9917	1.0032e+04	72.9858	–	14610	14396	14288	1.4409e+04	84.7410	–
GWO	10160	9950	9852	9.9545e+03	78.7951	–	14683	14322	14229	1.4352e+04	103.6125	–
HS	10126	10021	9928	1.0018e+04	57.2830	–	14645	14391	14263	1.4393e+04	92.4135	–
PSO	10359	10112	10022	1.0156e+04	102.0322	–	15211	14729	14417	1.4745e+04	190.4419	–
TLBO	10187	10045	9967	1.0053e+04	56.0603	–	14648	14453	14325	1.4452e+04	70.8422	–
SGO	**11084**	**10967**	**10722**	**1.0943e+04**	89.0441		**15990**	**15854**	**15290**	**1.5813e+04**	162.1211	
FPA	10175	10006	9923	1.0020e+04	70.5165	–	14632	14367	14295	1.4395e+04	74.7997	–
FFA	10417	10273	10193	1.0273e+04	53.6731	–	14941	14725	14636	1.4723e+04	67.7216	–
Algorithm	WCdata7, dim=1500						WCdata8, dim=2000					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	18172	17763	17476	1.7751e+04	185.2326	–	23025	22724	22427	2.2688e+04	180.0892	–
BA	21989	21533	21249	2.1527e+04	161.4532	–	28401	27996	27762	2.7997e+04	151.7021	–
BH	21360	21078	20940	2.1094e+04	103.9461	–	28015	27544	27385	2.7597e+04	152.0531	–
CS	21633	21356	21266	2.1379e+04	93.7829	–	28296	27891	27732	2.7897e+04	125.6355	–
DE	21687	21308	21159	2.1307e+04	121.7259	–	28143	27791	27638	2.7820e+04	150.7502	–
GWO	21523	21174	21029	2.1192e+04	115.7313	–	27852	27670	27485	2.7653e+04	105.7725	–
HS	21749	21312	21117	2.1312e+04	153.5591	–	28026	27732	27563	2.7744e+04	114.1172	–
PSO	22466	21896	21554	2.1897e+04	206.7262	–	29293	28714	28372	2.8728e+04	211.3318	–
TLBO	21699	21349	21185	2.1355e+04	124.6619	–	28320	27883	27719	2.7918e+04	151.2393	–
SGO	**24233**	**23731**	**23303**	**2.3699e+04**	193.9530		**32065**	**31647**	**30904**	**3.1612e+04**	259.3154	
FPA	21442	21260	21084	2.1258e+04	119.7747	–	28020	27708	27549	2.7723e+04	123.9565	–
FFA	21822	21660	21545	2.1669e+04	77.6342	–	28460	28230	28061	2.8233e+04	106.3678	–
Algorithm	WCdata9, dim=3000						WCdata10, dim=5000					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	33848	33165	32741	3.3178e+04	266.4967	–	53869	53022	52626	5.3063e+04	363.7348	–
BA	42510	41986	41620	4.1981e+04	222.1270	–	68990	68323	67944	6.8339e+04	274.9376	–
BH	41702	41232	40972	4.1260e+04	202.1042	–	67582	67278	67004	6.7274e+04	150.4867	–
CS	42314	41710	41513	4.1715e+04	170.6706	–	68377	67799	67626	6.7853e+04	207.0322	–
DE	41835	41575	41329	4.1568e+04	138.6789	–	68176	67731	67375	6.7677e+04	196.9331	–
GWO	41762	41373	41196	4.1389e+04	133.3942	–	67789	67447	67218	6.7442e+04	150.1487	–
HS	41898	41416	41200	4.1449e+04	170.2524	–	68129	67313	67118	6.7370e+04	210.9200	–
PSO	43940	43148	42677	4.3197e+04	282.2162	–	72156	71054	70159	7.0985e+04	468.9905	–
TLBO	41994	41593	41389	4.1628e+04	168.3593	–	68332	67812	67547	6.7823e+04	184.7154	–
SGO	**48463**	**47904**	**47063**	**4.7849e+04**	366.4701		**80000**	**79322**	**78473**	**7.9215e+04**	408.5615	
FPA	41902	41509	41305	4.1517e+04	159.3380	–	68124	67521	67354	6.7569e+04	180.6549	–
FFA	44510	43986	43620	4.3981e+04	222.1270	–	70990	70323	70944	6.8339e+04		–

"–", "+" and "≈" denote that performance of algorithms are worse, better and similar to BSGO respectively

Table 6
Results of SC data high dimensional 01knapsack problem , 30 different runs, 200 generation

Algorithm	SCdata1, dim=100						SCdata2, dim=200					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	2311	2292	2287	2.2938e+03	5.3439	–	4702	4679	4668	4.6794e+03	9.5581	–
BA	2312	2302	2296	2.3033e+03	4.2938	–	4702	4681	4670	4.6833e+03	9.5490	–
BH	2306	2294	2290	2.2952e+03	3.9427	–	4691	4667	4655	4.6665e+03	8.5328	–
CS	2320	2306	2301	2.3064e+03	4.7894	–	4702	4681	4673	4.6824e+03	6.4680	–
DE	2312	2302	2295	2.3025e+03	5.0702	–	4691	4679	4667	4.6775e+03	6.6006	–
GWO	2306	2297	2291	2.2967e+03	4.3482	–	4686	4668	4657	4.6677e+03	7.8227	–
HS	2311	2302	2292	2.3014e+03	4.1745	–	4698	4678	4665	4.6757e+03	8.1110	–
PSO	2312	2301	2292	2.3015e+03	4.5314	–	4692	4675	4666	4.6765e+03	7.1570	–
TLBO	2312	2302	2293	2.3025e+03	5.2572	–	4692	4672	4663	4.6741e+03	7.5263	–
SGO	**2322**	**2312**	**2312**	**2.3147e+03**	7.7901		**4739**	**4712**	**4689**	**4.7120e+03**	13.1293	
FPA	2312	2301	2295	2.3019e+03	4.3419	–	4690	4675	4668	4.6754e+03	6.0440	–
FFA	2322	2312	2311	2.3134e+03	3.0240	–	4709	4698	4689	4.6977e+03	5.0945	–

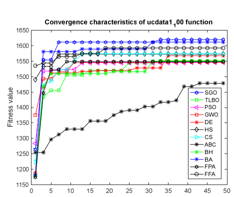
Algorithm	SCdata3, dim=300						SCdata4, dim=500					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	6883	6854	6830	6.8552e+03	15.0594	–	10878	10461	10298	1.0502e+04	169.5156	–
BA	6873	6853	6834	6.8516e+03	11.8778	–	11223	11159	11016	1.1147e+04	44.5827	–
BH	6855	6820	6799	6.8217e+03	15.6539	–	11124	10951	10806	1.0947e+04	83.8600	–
CS	6871	6852	6838	6.8521e+03	8.6969	–	11217	11129	11058	1.1132e+04	39.1228	–
DE	6884	6843	6823	6.8445e+03	13.4901	–	11199	11079	10974	1.1080e+04	61.9108	–
GWO	6874	6829	6800	6.8326e+03	17.4701	–	11159	10972	10888	1.0987e+04	64.6446	–
HS	6890	6841	6830	6.8432e+03	12.9473	–	11206	11084	10966	1.1080e+04	62.3268	–
PSO	6882	6844	6826	6.8462e+03	12.4155	–	11232	11155	10988	1.1142e+04	57.3774	–
TLBO	6881	6840	6817	6.8428e+03	17.0241	–	11162	11063	10964	1.1068e+04	57.6314	–
SGO	**6958**	**6930**	**6872**	**6.9262e+03**	21.5488		**11408**	**11347**	**11263**	**1.1339e+04**	39.3516	
FPA	6869	6838	6822	6840	11.9827	–	11238	11053	10970	1.1068e+04	80.0177	–
FFA	6892	6882	6869	6.8814e+03	6.3706	–	11228	11199	11181	1.1199e+04	11.6305	–

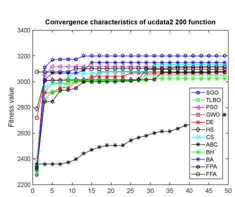
Algorithm	SCdata5, dim=700						SCdata6, dim=1000					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	14099	13640	13389	1.3675e+04	200.4054	–	19031	18520	18348	1.8594e+04	216.3066	–
BA	15579	15229	15100	1.5269e+04	136.3518	–	21918	21647	21477	2.1656e+04	122.4750	–
BH	15218	14944	14752	1.4942e+04	102.2942	–	21539	21250	21048	2.1256e+04	130.7398	–
CS	15353	15157	15037	1.5171e+04	79.7179	–	22056	21545	21377	2.1580e+04	137.6301	–
DE	15498	15121	14999	1.5147e+04	128.9765	–	21752	21488	21352	2.1503e+04	105.7471	–
GWO	15306	14970	14860	1.5011e+04	115.5403	–	21633	21301	21170	2.1334e+04	138.0600	–
HS	15349	15078	14951	1.5095e+04	106.3415	–	21672	21390	21239	2.1404e+04	110.8265	–
PSO	15566	15358	15036	1.5344e+04	146.2778	–	22441	22024	21618	2.2008e+04	202.2212	–
TLBO	15386	15130	14999	1.5154e+04	119.1638	–	21779	21539	21352	2.1522e+04	106.0562	–
SGO	**15829**	**15770**	**15665**	**1.5764e+04**	42.8715		**22845**	**22769**	**22651**	**2.2760e+04**	46.7444	
FPA	15502	15093	14978	1.5118e+04	111.4555	–	21792	21398	21197	2.1406e+04	130.1908	–
FFA	15610	15454	15370	1.5461e+04	65.7216	–	22260	21941	21807	2.1974e+04	125.0034	–

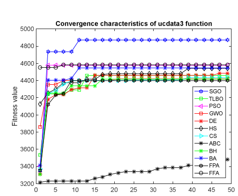
Algorithm	SCdata7, dim=1500						SCdata8, dim=2000					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	26778	26351	26044	2.6325e+04	202.0349	–	35333	34171	33673	3.4237e+04	444.9424	–
BA	32247	31867	31574	3.1864e+04	181.1630	–	43100	42295	42043	4.2360e+04	234.0165	–
BH	31978	31433	31185	3.1416e+04	177.5944	–	42248	41653	41420	4.1692e+04	205.1012	–
CS	32292	31770	31582	3.1806e+04	153.0968	–	42479	42128	41873	4.2119e+04	156.2905	–
DE	31986	31660	31427	3.1666e+04	146.1943	–	42485	42028	41774	4.2028e+04	164.2030	–
GWO	31818	31491	31318	3.1497e+04	138.0504	–	42292	41738	41524	4.1782e+04	190.6987	–
HS	31889	31668	31440	3.1661e+04	130.8470	–	42229	41846	41621	4.1851e+04	156.3408	–
PSO	33260	32720	31844	3.2627e+04	344.6677	–	44363	43403	42864	4.3426e+04	346.4391	–
TLBO	32345	31722	31575	3.1775e+04	165.7838	–	42566	42049	41809	4.2059e+04	179.2287	–
SGO	**34290**	**34162**	**33880**	**3.4143e+04**	101.0356		**45848**	**45714**	**45605**	**4.5724e+04**	59.3497	
FPA	31843	31525	31407	3.1583e+04	127.5541	–	42347	41875	41639	4.1907e+04	190.5675	–
FFA	32504	32236	32090	3.2240e+04	116.3916	–	42967	42588	42376	4.2580e+04	141.8856	–

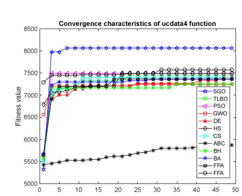
Algorithm	SCdata9, dim=3000						SCdata10, dim=5000					
	Best	Median	Worse	Mean	Std	WRS	Best	Median	Worse	Mean	Std	WRS
ABC	50291	49947	49761	4.9985e+04	202.3667	–	80331	79960	79812	80003	198.0088	–
BA	63515	62663	62501	6.2788e+04	416.9601	–	104363	104033	103500	103925	341.7287	–
BH	62159	61320	61143	6.1490e+04	400.1497	–	101830	101778	101388	1.0165e+05	202.1702	–
CS	62340	62100	62067	62163	122.2845	–	102929	102368	102291	1.0248e+05	259.6812	–
DE	62379	62168	62010	6.2195e+04	157.7317	–	102215	101956	101892	1.0199e+05	129.6430	–
GWO	61982	61815	61731	6.1830e+04	92.7723	–	102927	102157	101976	1.0230e+05	392.0825	–
HS	62050	61717	61583	6.1782e+04	178.9897	–	102316	102254	101778	1.0213e+05	224.2124	–
PSO	65135	64667	64530	6.4750e+04	239.7430	–	107616	107333	106772	1.0726e+05	374.9071	–
TLBO	62274	62107	61990	6.2116e+04	109.5276	–	102614	102480	102442	102500	68.2971	–
SGO	**68479**	**68382**	**68211**	**6.8363e+04**	115.0270		**114431**	**114315**	**114295**	**1.1433e+05**	**55.7611**	
FPA	62067	61940	61784	6.1921e+04	113.6561	–	102719	102122	101992	1.0231e+05	342.3663	–
FFA	63515	62663	62067	6.2388e+04	416.9601	–	104363	104033	103500	103925	123.7812	–

" –", "+" and "≈" denote that performance of algorithms are worse, better and similar to BSGO respectively









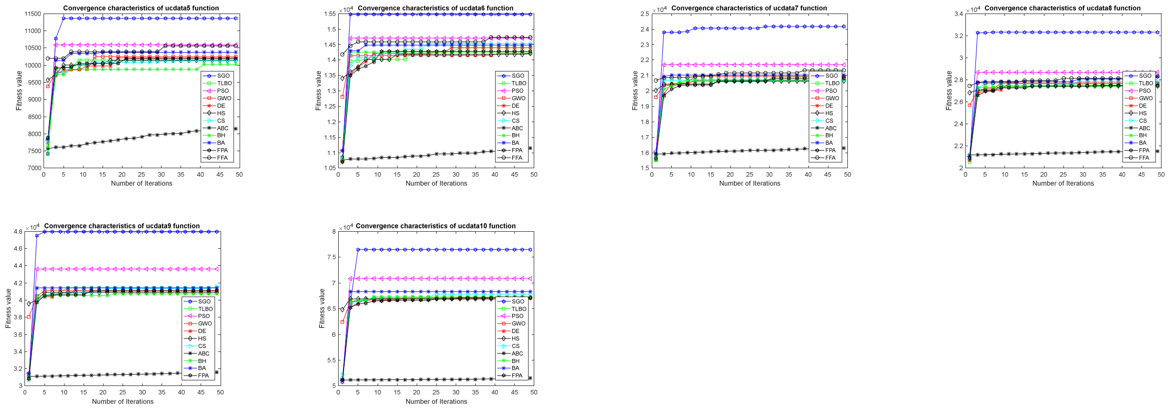


Fig. 3. Convergence characteristics of uncorrelated 0-1 KPs on first 50 iterations

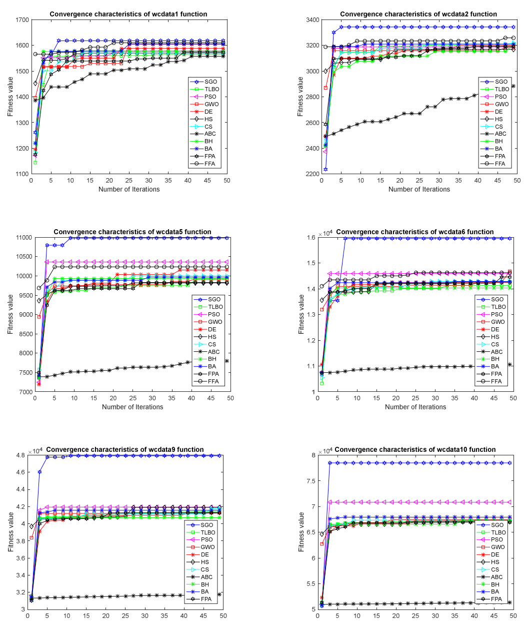
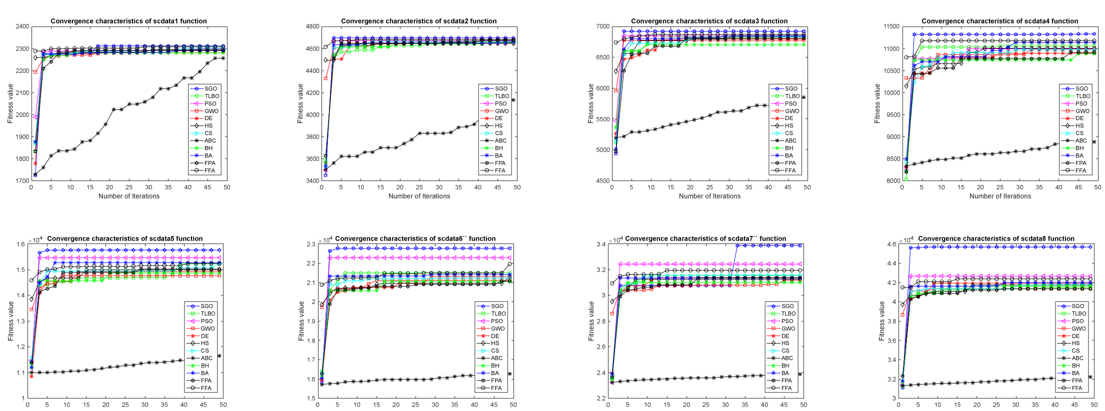


Fig. 4. Convergence characteristics of weakly correlated 0-1 KPs on first 50 iterations



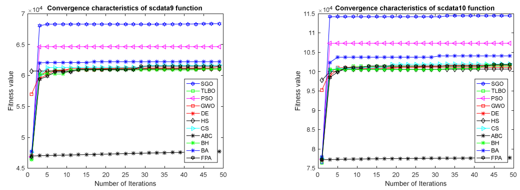


Fig. 5. Convergence characteristics of strongly correlated 0-1 KPs on first 50 iterations

Performance estimation

The performance of the proposed BSGO algorithm is investigated by using the Wilcoxon rank-sum (WRS) test for a better comparison. WRS test is a nonparametric test that utilized in a hypothesis testing situation involving a design with two samples (Derrac et al., 2011). It is a pair-wise test that aims to find out significant differences between the behaviors of two algorithms. Therefore, we have conducted the WRS test at 0.05 significance level on experimental results for proposed BSGO algorithm against the different algorithm appears in Table 4, Table 5 and Table 6 and the obtained results are reported in the last column of the same table.

Discussion

From Table 4, Table 5, Table 6, according to the WRS test it clearly indicates that the proposed BSGO algorithm achieves better results than all other algorithms for all 30 large-scale 0-1 KPs. From graphs, it is clearly visible that the proposed BSGO achieves better simulation results than all different algorithms within ten iterations itself. Consequently, the profit for each problem has improved significantly by BSGO. So it can be concluded that the proposed BSGO has some excellent characteristics of better optimization ability and good convergence as well. BSGO is a promising variant of SGO and has a strong ability to solve large-scale 0-1 KP problems than other investigated algorithms.

5. Conclusions and future work

This paper presented a novel binary social group optimization algorithm (BSGO) for solving 0/1 knapsack problems. The proposed algorithm has been extensively investigated through using 10 small-scale and 30 large-scale problems of 0–1 KP. The proposed algorithm is compared with several algorithms from the literature. Based on statistical measures, the proposed algorithm can explore better-quality solutions, and it outperforms than other compared algorithms, especially in large-scale 0-1 KPs, and comes out as one of the best algorithm. From convergence graphs, it is clearly visible that within ten iterations itself BSGO algorithm finds better results than other algorithms. The results reveal that BSGO is competent to provide very competitive results compared to other investigated algorithms. For future work, it is possible to investigate the proposed BSGO algorithm to solve different forms of KP problems like multi-dimensional KP and quadratic KP as well as various combinatorial optimization problems such as subset sum problem, quadratic assignment problem, and job shop scheduling problems.

Compliance with ethical standards

Conflict of Interest: The authors declare that they have no conflict of interest.

References

Abdel-Basset, M., El-Shahat, D., & El-Henawy, I. (2019). Solving 0–1 knapsack problem by binary flower pollination algorithm. *Neural Computing and Applications, 31*(9), 5477-5495.

Abdel-Basset, M., Luo, Q., Miao, F., & Zhou, Y. (2017). Solving 0–1 knapsack problems by binary dragonfly algorithm. *In International conference on intelligent computing (pp. 491-502).* Springer, Cham.

Abdel-Basset, M., & Zhou, Y. (2018). An elite opposition-flower pollination algorithm for a 0-1 knapsack problem. *International Journal of Bio-Inspired Computation, 11*(1), 46-53.

Bansal, J. C., & Deep, K. (2012). A modified binary particle swarm optimization for knapsack problems. *Applied Mathematics and Computation, 218*(22), 11042-11061.

Bhattacharjee, K. K., & Sarmah, S. P. (2014). Shuffled frog leaping algorithm and its application to 0/1 knapsack problem. *Applied Soft Computing, 19*, 252-263.

Catharin, A. R., Kumar, A. S., Rakshiga, M., Kumaresan, S., & Raja, N. S. M. (2018, March). Examination of Glioblastoma Images by Thresholding using Heuristic Approach. *In 2018 Fourth International Conference on Biosignals, Images and Instrumentation (ICBSII) (pp. 206-212). IEEE.*

Chakravarthy, V. S., Chowdary, P. S. R., Satpathy, S. C., Terlapu, S. K., & Anguera, J. (2018). Antenna array synthesis using social group optimization. *In Microelectronics, Electromagnetics and Telecommunications (pp. 895-905).* Springer, Singapore.

Changdar, C., Mahapatra, G. S., & Pal, R. K. (2013). An ant colony optimization approach for binary knapsack problem under fuzziness. *Applied Mathematics and Computation, 223*, 243-253.

Cobos, C., Dulcey, H., Ortega, J., Mendoza, M., & Ordoñez, A. A Binary Fisherman Search Procedure for the 0/1 Knapsack Problem. *Advances in Artificial Intelligence: Proceedings of the 17th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2016. O*. Luaces et al.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3-18.

Dey, N., Rajinikanth, V., Ashour, A. S., & Tavares, J. M. R. (2018). Social group optimization supported segmentation and evaluation of skin melanoma images. *Symmetry, 10*(2), 51.

Du, D. P., & Zu, Y. R. (2015). Greedy strategy based self-adaption ant colony algorithm for 0/1 knapsack problem. *In Ubiquitous Computing Application and Wireless Sensor (pp. 663-670).* Springer, Dordrecht.

Fang, J., Zheng, H., Liu, J., Zhao, J., Zhang, Y., & Wang, K. (2018). A transformer fault diagnosis model using an optimal hybrid dissolved gas analysis features subset with improved social group optimization-support vector machine classifier. *Energies, 11*(8), 1922.

Feng, Y., Wang, G. G., Feng, Q., & Zhao, X. J. (2014). An effective hybrid cuckoo search algorithm with improved shuffled frog leaping algorithm for 0-1 knapsack problems. *Computational Intelligence and Neuroscience*, 2014.

Feng, Y., Wang, G. G., Dong, J., & Wang, L. (2018). Opposition-based learning monarch butterfly optimization with Gaussian perturbation for large-scale 0-1 knapsack problem. *Computers & Electrical Engineering, 67*, 454-468.

Feng, Y., Wang, G. G., Deb, S., Lu, M., & Zhao, X. J. (2017). Solving 0–1 knapsack problem by a novel binary monarch butterfly optimization. *Neural Computing and Applications, 28*(7), 1619-1634.

Gherboudj, A., Layeb, A., & Chikhi, S. (2012). Solving 0-1 knapsack problems by a discrete binary version of cuckoo search algorithm. *International Journal of Bio-Inspired Computation, 4*(4), 229-236.

Gong, Q. Q., Zhou, Y. Q., & Yang, Y. (2011). Artificial glowworm swarm optimization algorithm for solving 0-1 knapsack problem. *In Advanced materials research (Vol. 143, pp. 166-171).* Trans Tech Publications Ltd.

Kong, M., & Tian, P. (2006, June). Apply the particle swarm optimization to the multidimensional knapsack problem. *In International conference on artificial intelligence and soft computing (pp. 1140-1149).* Springer, Berlin, Heidelberg.

Kong, X., Gao, L., Ouyang, H., & Li, S. (2015). A simplified binary harmony search algorithm for large scale 0–1 knapsack problems. *Expert Systems with Applications, 42*(12), 5337-5355.

Kulkarni, A. J., Krishnasamy, G., & Abraham, A. (2017). Solution to 0–1 knapsack problem using cohort intelligence algorithm. *In Cohort Intelligence: A Socio-inspired Optimization Method (pp. 55-74)*. Springer, Cham.

Layeb, A. (2011). A novel quantum inspired cuckoo search for knapsack problems. *International Journal of Bio-inspired Computation, 3*(5), 297-305.

Leena, J. G., Sundaravadivu, K., Monisha, R., & Rajinikanth, V. (2018, July). Design of Fractional-Order PI/PID Controller for SISO System Using Social-Group-Optimization. *In 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA) (pp. 1-5). IEEE.*

Lim, T. Y., Al-Betar, M. A., & Khader, A. T. (2016). Taming the 0/1 knapsack problem with monogamous pairs genetic algorithm. *Expert Systems with Applications, 54*, 241-250.

Lin, C. J., Chern, M. S., & Chih, M. (2016). A binary particle swarm optimization based on the surrogate information with proportional acceleration coefficients for the 0-1 multidimensional knapsack problem. *Journal of Industrial and Production Engineering, 33*(2), 77-102.

Lv, J., Wang, X., Huang, M., Cheng, H., & Li, F. (2016). Solving 0-1 knapsack problem by greedy degree and expectation efficiency. *Applied Soft Computing, 41*, 94-103.

Ma, Y., & Wan, J. (2011, July). Improved hybrid adaptive genetic algorithm for solving knapsack problem. *In 2011 2nd International Conference on Intelligent Control and Information Processing (Vol. 2, pp. 644-647). IEEE.*

Madhavi, G., & Harika, V. (2018). Implementation of social group optimization to economic load dispatch problem. *International Journal of Applied Engineering Research, 13*(13), 11195-11200.

Mani, M. S., Manisha, S., Thanaraj, K. P., & Rajinikanth, V. (2017, July). Automated segmentation of Giemsa stained microscopic images based on entropy value. *In 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) (pp. 1124-1128*). IEEE.

Martello, S., Pisinger, D., & Toth, P. (2000). New trends in exact algorithms for the 0–1 knapsack problem. *European Journal of Operational Research, 123*(2), 325-332.

Monisha, R., Mrinalini, R., Britto, M. N., Ramakrishnan, R., & Rajinikanth, V. (2019). Social group optimization and Shannon's function-based RGB image multi-level thresholding. *In Smart Intelligent Computing and Applications (pp. 123-132)*. Springer, Singapore.

Naik, A., & Satapathy, S. C. (2021). A comparative study of social group optimization with a few recent optimization algorithms. *Complex & Intelligent Systems, 7*(1), 249-295.

Naik, A., Satapathy, S. C., & Abraham, A. (2020). Modified Social Group Optimization—A meta-heuristic algorithm to solve short-term hydrothermal scheduling. *Applied Soft Computing, 95*, 106524.

Naik, A., Satapathy, S. C., Ashour, A. S., & Dey, N. (2018). Social group optimization for global optimization of multimodal functions and data clustering problems. *Neural Computing and Applications, 30*(1), 271-287.

Naik, A., Satapathy, S. C., & Jena, J. J. (2021). Non-dominated Sorting Social Group Optimization algorithm for multiobjective optimization. *Journal of Scientific and Industrial Research (JSIR), 80*(02), 129-136.

Nguyen, P. H., Wang, D., & Truong, T. K. (2017). A novel binary social spider algorithm for 0–1 knapsack problem. *Internatioanl Journal of Innovative Computation Information Control, 13*(6), 2039-2049.

Nguyen, P. H., Wang, D., & Truong, T. K. (2016). A new hybrid particle swarm optimization and greedy for 0-1 knapsack problem. *Indonesian Journal of Electronic Engeering Compututation Science, 1*(3), 411-418.

Parwekar, P. (2020). *SGO a new approach for energy efficient clustering in WSN. In Sensor Technology: Concepts, Methodologies, Tools, and Applications (pp. 716-734)*. IGI Global.

Pavithr, R. S. (2016). Quantum Inspired Social Evolution (QSE) algorithm for 0-1 knapsack problem. *Swarm and Evolutionary Computation, 29*, 33-46.

Pham, A. H., Vu, T. V., & Tran, T. M. (2017, August). Optimal Volume Fraction of Functionally Graded Beams with Various Shear Deformation Theories Using Social Group Optimization. *In International Conference on Advances in Computational Mechanics (pp. 395-408).* Springer, Singapore.

Praveen, S. P., Rao, K. T., & Janakiramaiah, B. (2018). Effective allocation of resources and task scheduling in cloud environment using social group optimization*. Arabian Journal for Science and Engineering, 43*(8), 4265-4272.

Rajinikanth, V., & Satapathy, S. C. (2018). Segmentation of ischemic stroke lesion in brain MRI based on social group optimization and Fuzzy-Tsallis entropy. *Arabian Journal for Science and Engineering, 43*(8), 4365-4378.

Rani, S., & Suri, B. (2019, July). Adopting social group optimization algorithm using mutation testing for test suite generation: SGO-MT. *In International Conference on Computational Science and Its Applications (pp. 520-528).* Springer, Cham.

Razavi, S. F., & Sajedi, H. (2015). Cognitive discrete gravitational search algorithm for solving 0-1 knapsack problem. *Journal of Intelligent & Fuzzy Systems, 29*(5), 2247-2258.

Rizk-Allah, R. M., & Hassanien, A. E. (2018). New binary bat algorithm for solving 0–1 knapsack problem. *Complex & Intelligent Systems, 4*(1), 31-53.

Satapathy, S., & Naik, A. (2016). Social group optimization (SGO): a new population evolutionary optimization technique. *Complex & Intelligent Systems, 2*(3), 173-203.

Shi, H. (2006, August). Solution to 0/1 knapsack problem based on improved ant colony algorithm. *In 2006 IEEE International Conference on Information Acquisition (pp. 1062-1066). IEEE.*

Sonuc, E., Sen, B., & Bayir, S. (2016). A parallel approach for solving 0/1 knapsack problem using simulated annealing algorithm on CUDA platform. *International Journal of Computer Science and Information Security, 14*(12), 1096.

Tran, M. T., Pham, H. A., Nguyen, V. L., & Trinh, A. T. (2017). Optimisation of stiffeners for maximum fundamental frequency of cross-ply laminated cylindrical panels using social group optimisation and smeared stiffener method. *Thin-Walled Structures, 120*, 172-179.

Truong, T. K., Li, K., Xu, Y., Ouyang, A., & Nguyen, T. T. (2015). Solving 0-1 knapsack problem by artificial chemical reaction optimization algorithm with a greedy strategy. *Journal of Intelligent & Fuzzy Systems, 28*(5), 2179-2186.

Ulker, E., & Tongur, V. (2017). Migrating birds optimization (MBO) algorithm to solve knapsack problem. *Procedia Computer Science, 111*, 71-76.

Wang, L., Yang, R., Xu, Y., Niu, Q., Pardalos, P. M., & Fei, M. (2013). An improved adaptive binary harmony search algorithm. *Information Sciences, 232*, 58-87.

Yan, C., Gao, S., Luo, H., & Hu, Z. (2015, June). A hybrid algorithm based on tabu search and chemical reaction optimization for 0-1 knapsack problem. *In International Conference in Swarm Intelligence (pp. 229-237). Springer*, Cham.

Zhang, X., Huang, S., Hu, Y., Zhang, Y., Mahadevan, S., & Deng, Y. (2013). Solving 0-1 knapsack problems based on amoeboid organism algorithm. *Applied Mathematics and Computation, 219*(19), 9959-9970.

Zhou, G., Zhao, R., & Zhou, Y. (2018). Solving large-scale 0-1 knapsack problem by the social-spider optimisation algorithm. *International Journal of Computing Science and Mathematics, 9*(5), 433-441.

Zhou, Y., Bao, Z., Luo, Q., & Zhang, S. (2017). A complex-valued encoding wind driven optimization for the 0-1 knapsack problem. *Applied Intelligence, 46*(3), 684-702.

Zhou, Y., Chen, X., & Zhou, G. (2016). An improved monkey algorithm for a 0-1 knapsack problem. *Applied Soft Computing, 38*, 817-830.

Zhou, Y., Li, L., & Ma, M. (2016). A complex-valued encoding bat algorithm for solving 0–1 knapsack problem. *Neural Processing Letters, 44*(2), 407-430.

Zou, D., Gao, L., Li, S., & Wu, J. (2011). Solving 0–1 knapsack problem by a novel global harmony search algorithm. *Applied Soft Computing, 11*(2), 1556-1564.

Appendix A

Table A1.
List of small-scale 0-1 knapsack problems

Problem No.	Dimension	Capacity	Weights	Profits
		First set		
P1	10	269	[95, 4, 60, 32, 23, 72, 80, 62, 65, 46]	[55, 10, 47, 5, 4, 50, 8, 61, 85, 87]
P2	20	878	[92, 4, 43, 83, 84, 68, 92, 82, 6, 44, 32, 18, 56, 83, 25, 96, 70,48, 14, 58]	[44, 46, 90, 72, 91, 40, 75, 35, 8, 54, 78, 40, 77, 15, 61, 17, 75, 29, 75, 63]
P3	4	20	[6, 5, 9, 7]	[9, 11, 13, 15]
P4	4	11	[2, 4, 6, 7]	[6, 10, 12, 13]
P5	15	375	[56.358531, 80.874050, 47.987304, 89.596240, 74.660482, 85.894345, 51.353496, 1.498459, 36.445204, 16.589862, 44.569231, 0.466933, 37.788018, 57.118442, 60.716575]	[0.125126, 19.330424, 58.500931, 35.029145, 82.284005,17.410810, 71.050142, 30.399487, 9.140294, 14.731285, 98.852504, 11.908322, 0.891140, 53.166295, 60.176397]
P6	10	60	[30, 25, 20, 18, 17, 11, 5, 2, 1, 1]	[20, 18, 17, 15, 15, 10, 5, 3, 1, 1]
P7	7	50	[31, 10, 20, 19, 4, 3, 6]	[70, 20, 39,37, 7, 5, 10]
P8	23	10000	[983, 982, 981, 980, 979, 978, 488, 976, 972, 486, 486, 972, 972, 485, 485, 969, 966, 483, 964, 963, 961, 958, 959]	[981, 980, 979, 978, 977, 976, 487, 974, 970, 485, 485, 970, 970, 484, 484, 976, 974, 482, 962, 961, 959, 958, 857]
P9	5	80	[15, 20, 17, 8, 31]	[33, 24, 36, 37, 12]
P10	20	879	[84, 83, 43, 4, 44, 6, 82, 92, 25, 83, 56, 18, 58, 14, 48, 70, 96, 32, 68, 92]	[91, 72, 90, 46, 55, 8, 35, 75, 61, 15, 77, 40, 63, 75, 29, 75, 17, 78, 40, 44]

Table A2.
Correlation characteristics of profit and weight for large-scale 0-1 knapsack problem

Correlation	Weight (w_i)	Profit(p_i))	C
Uncorrelated items	Rand(10,30)	Rand(10,30)	0.75*$\sum_{i=1}^{N} w_i$
Weakly correlated items	Rand(10,30)	Rand($w_i - 10$, w_i+10)	0.75*$\sum_{i=1}^{N} w_i$
Strongly correlated items	Rand(10,30)	w_i+10	0.75*$\sum_{i=1}^{N} w_i$