AIP Conference Proceedings

scitation.org/journal/apc

Volume 2540

The 3rd International Conference on Science, Mathematics, Environment, and Education

Flexibility in Research and Innovation on Science, Mathematics, Environment, and education for sustainable development

Surakarta, Indonesia • 27–28 July 2021

Editors • Nurma Yunita Indriyanti and Meida Wulan Sari

ICoSMEE

THE 3RD INTERNATIONAL CONFERENCE ON SCIENCE, MATHEMATICS, ENVIRONMENT, AND EDUCATION

AIP Publishing

Preface: Flexibility in Research and Innovation on Science, Mathematics, Environment, and Education for Sustainable Development

Cite as: AIP Conference Proceedings **2540**, 010001 (2023); https://doi.org/10.1063/12.0012091
Published Online: 27 January 2023

Preface: Flexibility in Research and Innovation on Science, Mathematics, Environment, and education for sustainable development

This year we present the conference theme '**Flexibility in Research and Innovation on Science, Mathematics, Environment, and education for sustainable development**'. This theme is closely related to the 21st century and the Society 5.0 issues. This conference aims to provide a forum to share and discuss ideas and new development in science, math, environment and education, as well as novel ideas in a holistic manner to address the challenge and issues on sustainable future. According to the theme, this conference encompasses a wide spectrum of topics including the state-of-art and recent trend in different innovative research in science, mathematics, environment and education that will be covered through plenary and parallel session. Due to the pandemic, we conducted this year conference in online mode.

This second International Conference on Science, Mathematics, Environment and Education (ICoSMEE) aims at bringing together researchers, educators, scientists, and scholar students in the area of Science, Mathematics, Environment and Education to exchange and share their experiences, ideas, and findings and to discuss practical challenges encountered and the solutions to develop humanity and the quality of human life in a sustainable manner. The conference was held in July 27-28, 2021. Hundred of mathematics and science education experts and practitioners joined the event. There were six keynote speakers who came from Indonesia, United States, United Kingdom, Taiwan, and Japan. The presenters of parallel session were from Indonesia, Malaysia, Brunei Darussalam, and Hungary. There are 214 articles were presented in the conference and 162 articles are selected to be publishing in the present conference proceeding.

The Editors,

Dr.paed. Nurma Yunita Indriyanti
Meida Wulan Sari, M.Pd.

COMMITTEE

International Advisory Commitee
Prof. Dr. Hans-Dieter Barke
Prof Bob Bucat
Prof. Lilia Halim
Prof. Dr. Karl-Heinz Van Pee
Heri Setya Kusuma, Ph.D

STEERING COMMITTEE
Dr. Mardiyana, M.Si
Prof. Slamet Subiyantoro, M.Si
Dr. Dewi Kusumawardani, M.Si
Dr. Djono, M.Pd

ORGANIZING COMMITTEE

Chairman : Dr. Sri Budiawanti, S.Si.,M.Si.

The 3rd International Conference on Science, Mathematics, Environment, and Education
AIP Conf. Proc. 2540, 010001-1–010001-3; https://doi.org/10.1063/12.0012091
Published by AIP Publishing. 978-0-7354-4309-9/$30.00

Deputy Chairman	:	Dr. Triyanto, S.Si, M.Si
Secretary	:	Febriani Sarwendah Asri Nugraheni, M.Pd Fairusy Fitria Haryani, S.Pd, M.Si Riki Andriatna, S.Pd., M.Pd.
		Dr. Yulianto Agung Rezeki, S.Pd, M.Si
Treasury	:	Nurmiyati, S.Pd, M.Si
		Shofi Hikmatuz Zahroh, S.Pd., M.Pd.
		Bachtiar Arddun
		Hilmi Uulul Azmi
		Nathania Almira Chiesa
		Niken Renawati
Website and Publication	:	Dr. Budi Usodo, M.Pd
		Chandra Adi Prabowo, S.Pd, M.Pd
		Dimas Gilang Ramadhani, S.Pd
		Ario Wiraya, S.Si., M.Sc.
Event	:	Dr. Sri Yamtinah, S.Pd., M.Pd
		Isma Aziz Fakhrudin, M.Pd
		Lina Mahardiani, S.T., M.M., M.Sc., Ph.D.
		Elvin Yusliana Ekawati, S.Pd, M.Pd
		Dr. Muzzazinah, M.Si
		Azizah Nurul I
		Hayu Diah Cahyani
		Sri Lestari
		Shofi Nurtaqia
Paper and Proceeding	:	Dr.paed Nurma Yunita Indriyanti, M.Si, M.Sc
		Dewanto Harjunowibowo, S.Si., M.Sc.
		Meida Wulan Sari, M.Pd.
		Dr. Farida Nurhasanah, M.Pd
		Dr. Umi Fatmawati, S.Pd M.Si
		Murni Ramli, S.P, M.Si.,Ed.D
		Dr.rer.nat. Sri Mulyani, M.Si
		M. Ahsanul Taqwim
		Annisa Widyastuti
		Ellis Octavia
		Yuniar Mindiarti

		Lintang Rahmayanti
		Siti Nur Afifah
		Tyas Dwi
Food and Beverage	:	Yesiana Arimurti, S.Si., M.Sc.
		Kistantia Elok Mumpuni, S.Pd., M.Pd.
		Fadhilah Hanum
		Khoirul Annam
		Syahreza Figo
Webinar Management	:	Bayu Antrakusuma, M.Pd
		Alanindra Saputra, S.Pd., M.Sc.

Application of combined GSA&sCSO algorithm to modified bounded knapsack with multiple constraints problem against uncertain coefficient

Cite as: AIP Conference Proceedings **2540**, 080011 (2023); https://doi.org/10.1063/5.0107520
Published Online: 27 January 2023

Ingka Maris, Agustina Pradjaningsih and Kiswara Agung Santoso

Application of Combined GSA&sCSO Algorithm to Modified Bounded Knapsack with Multiple Constraints Problem against Uncertain Coefficient

Ingka Maris[1, a)], Agustina Pradjaningsih[2], and Kiswara Agung Santoso[2]

[1]*Master Program of Mathematics Departement, Faculty of Science, University of Jember*
Jl. Kalimantan No. 37 Kampus Tegal Boto, Jember, Indonesia
[2]*Mathematics Departement, Faculty of Science, University of Jember*
Jl. Kalimantan No. 37 Kampus Tegal Boto, Jember, Indonesia

[a)]Corresponding author: ingkamaris@gmail.com

Abstract. Optimization problems are interest and common problems that are often encountered in life. Optimization can be applied to solve various problems, for example development, government, business, social, economic and something related to the limitation of resource capacity. The most frequently encountered, optimization is often used to find the best solution, that is maximizing profits or minimizing production costs. One of the optimization problems that often occurs is the knapsack problem. There are several types of knapsack problems, one of which is Modified Bounded Knapsack with Multiple Constraints (MBKMC) problem. In popular mathematical studies, metaheuristic algorithms are very often used to solve optimization problems. In this paper the authors did not only use one algorithm, but implemented two metaheuristic algorithms which were combined into one, namely the Gravitational Search Algorithm (GSA) and the Cat Swarm Optimization (CSO) algorithm. The combined algorithm uses the entire GSA algorithm mechanism which is added with the CSO algorithm seeking mode to become the GSA&sCSO algorithm. The author uses the GSA&sCSO algorithm to solve the MBKMC problem of uncertain coefficient. Based on the results of this research, the GSA&sCSO algorithm produces a better solution (higher profit) than the GSA algorithm and the CSO algorithm and earn a better advantage in accordance with the knapsack capacity. In addition, the uncertain coefficient greatly affects the solution obtained, i.e if there is a change of the coefficient, then the solution also changes.

INTRODUCTION

In industrial world, there are so many problems to solve. Purchasing is one of the most important of those problems. In this purchasing problem, industries have to manage all items that should be chosen to buy. This problem arises when the industries have some obstacles, such as weight capacity, space capacities, capital, etc. Every industri is required to be able to solve this problem in order to get maximum profits. The selection of items must be done as well as possible so that the industry does not lose. To meet these objectives, the industry can solve the problem by optimizing it. In mathematics, this problem is commenly known as knapsack problem.

Knapsack problem is defined as an optimization problem in the selection of items to be included in a knapsack [1]. However, the knapsack has limited weight and space size so that the total weight and total size of the selected items should not exceed the capacity of the media or knapsack. There are several types of knapsack problems on mathematics, namely the 0-1 knapsack problem, bounded knapsack problem and unbounded knapsack problem [2]. The knapsack problem also has several variations, namely single objective knapsack, multi objective knapsack, multidimensional knapsack, multiple constraints knapsack and quadratic knapsack [3]. There are some research that discuss the combination of the type and variation of knapsack problem, such as multiple constraints 0-1 knapsack problem [4][5], multidimensional 0-1 knapsack problem [6], multidimensional 0-1 knapsack problem with multiple

The 3rd International Conference on Science, Mathematics, Environment, and Education
AIP Conf. Proc. 2540, 080011-1–080011-9; https://doi.org/10.1063/5.0107520
Published by AIP Publishing. 978-0-7354-4309-9/$30.00

constraints [7][8], multidimensional bounded knapsack problem [9][10] and multiple constraints bounded knapsack problem [11].

The solution for the knapsack optimization problems and its types described above can be found with several mathematic methods or algorithms. Many scientific research use metaheuristic algorithms because they can be an efficient way to produce the optimal solution [12]. Metaheuristic algorithms are the most common algorithms used to solve problems that based on biology, physics or ethology [13]. There are so many metaheuristic algorithms. Some of them used in the previous optimization research, including GSA: A Gravitational Search Algorithm [14] and Cat Swarm Optimization (CSO) algorithm [15].

Based on the description above, the authors are interested in solving knapsack problem using metaheuristic algorithm. In daily life, it is often found that the industry must determine the minimum and maximum amount of items to be chosen. Therefore, this problem can be classified into modified bounded knapsack problem. Since there are several constraints, which are weight, space, and capital, so this problem is also classied into knapsack problem with multiple constraints. Thus, in this reseach, the author are interested in solving modified bounded knapsack problem with multiple constraints (MBKMC). In order to solve the problem, we purpose a combination (hybrid) algorithm of GSA and CSO. In the proposed hybrid algorithm, the authors use the entire GSA mechanism, while the CSO algorithm mechanism used is only the seeking mode to balance the exploration and exploitation capabilities of the GSA and CSO hybrid algorithms. The aim of this research is to analyze the performance of the porposed algorithm to solve the MBKMC. In addition, based on the real condition, the cost of items can change anytime that makes the coefficient uncertain. Hence, this research also aims to analyze the changes of the optimal value of the MBKMC problem in the case of uncertain coefficient. To be able to fullfil the aims of this study, the author create a simulation program using MATLAB to help the computational experiment. In this reseach, we use some simulation data of MBKMC that randomly generated according to real condition of purchasing problem.

MODIFIED BOUNDED KNAPSACK WITH MULTIPLE CONSTRAINTS PROBLEM

The MBKMC problem is a knapsack problem that has a minimum and maximum amount of items available and more than one problem. Limits on the amaount of items have a purpose, such as the minimum amaount of items (l_b) to get maximum profit and the maximum amount of items (u_b) so as not to exceed media capacity or costs. MBKMC has several constraints including weight capacity, space and the cost or capital provided. MBKMC problems can be formulated in the following equation:

$$\text{Maximize} \quad Z = \sum_{j=1}^{n} p_j x_j \tag{1}$$

$$\text{Subject to} \quad \sum_{j=1}^{n} w_j x_j \leq C \tag{2}$$

$$\sum_{j=1}^{n} v_j x_j \leq S \tag{3}$$

$$\sum_{j=1}^{n} b_j x_j \leq M \tag{4}$$

$$x_j \in \mathbb{Z}, \;\; lb_j \leq x_j \leq ub_j, \;\; j = 1, 2, \dots, n \tag{5}$$

Equation (1) maximizes the objective function (Z) in the form of the total profit p_j with the quantity of items of type j of x_j. There are three constraints on the MBKMC problem, namely weight (w_j), volume (v_j) and cost (b_j). The amount of constraints must not exceed weight capacity (C), volume capacity (S) and cost or capital (M).

THE PROPOSED ALGORITHM

To obtain maximum profits with optimal solutions, in this paper, the authors use two algorithms, namely GSA and CSO. Then the two algorithms are combined into one called GSA&sCSO. The combination of the two algorithms is expected to get a more optimal solution than the compiler algorithms, namely GSA and CSO. An explanation of some of these algorithms will be explained below.

The GSA Algorithm

Gravitational Search Algorithm (GSA) is an optimization algorithm based on the law of gravity. In this algorithm, the agent is used to become an object and its performance is measured by mass. All objects attract each other by the force of gravity and cause a global movement of the heavier mass. Heavy masses that correspond to solutions will move slower than light ones [14]. The position of the mass corresponds to the solution of the problem which shows that gravity and mass are inertia with the time interval of the mass being attracted by the heaviest mass. The mass will show the optimal solution in the search space [16].

GSA algorithm steps:

a. Initialization early
The GSA algorithm has an initial N (agent) solution with m random dimensions. Agent positions can be calculated using the following Equation (6):

$$X_i = \left(x_i^1, \dots, x_i^d, \dots, x_i^n\right), i = 1,2, \dots, N \quad (6)$$

x_i^d is the position of the i agent in the d dimension.

b. Evaluate fitness and save the best and worst agents.

c. Update $G(t)$, best (t), worst (t) and the mass of inertia $M_i(t)$ for $i = 1,2, \dots, N$.
$G(t)$ is the gravitational constant at wich t can be calculated using the Equation (7):

$$G(t) = G_0 e^{-\alpha \frac{t}{T}} \quad (7)$$

G_0 and α are constant and T is the maximum iteration value. In the mass $M_i(t)$ of each agent there are the best (t) and worst (t) agent which are determined by the fitness value and are calculated using the following Equation (8) and Equation (9):

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (8)$$

$$M_i(t) = \frac{m_{i(t)}}{\sum_{j=1}^{N} m_j(t)} \quad (9)$$

d. Total force calculation
Each agent F has the total force in each iteration using the following Equation (10):

$$F_i^d(t) = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} \left(x_j^d(t) - x_i^d(t)\right) \quad (10)$$

ε is a small constant and a gravitational constant. $R_{ij}(t)$ is the euclid distance between agents which can be calculated using the Equation (11):

$$R_{ij}(t) = \left\| x_i(t), x_j(t) \right\|_2 \quad (11)$$

e. Acceleration (α) and velocity (v) calculation
Next calculate the speed and acceleration using the Equation (12) and Equation (13):

$$\alpha_i^d = \frac{\sum_{j=1, j \neq i}^{N} rand_j F_{ij}^d(t)}{M_i(t)} \quad (12)$$

$$v_i^d(t+1) = rand_0 X v_i^d(t) + \alpha_i^d(t) \quad (13)$$

f. Update agent position
The last stage is updating the agent position with the Equation and the procedure is repeated to the maximum iteration limt [6].

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (14)$$

Or we can calculate acceleration using Equation (15) [17]:

$$\alpha_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j \times M_j(t) \frac{G(t)}{R_{ij}(t) + \varepsilon} \left(x_j^d(t) - x_i^d(t)\right) \quad (15)$$

The CSO Algorithm

The Cat Swarm Optimization (CSO) algorithm was created based on the observations made on the behavior of a group of cats [7]. Cats have the ability to both hunt and be aware of their surroundings. The CSO algorithm uses cats and their behavior to solve optimization problems and is divided into 2 steps, namely seeking mode and tracing mode. Seeking mode is the stage where the cat looks around and devises a strategy. Tracing mode is a stage that describes a cat while following or tracking prey.

The first step of the CSO algorithm is to determine the number of individual cats. There are M dimensions, the speed of each dimension, the fitness value and the flag in each cat's position. The final solution of the algorithm is the best position of all cats saved during iteration.

The steps used in the CSO algorithm are as follows:

a. Initial position and velocity of N cats in search space.

b. Evaluate the fitness value and save the best position as x_{best}.

c. Divide the cats into seeking and tracing mode.

d. If cat is in seeking mode, create new candidate position using Equation (16), and then choose one to replace the current position using Roulette Wheel.

$$x'_{j,d} = x_{j,d} \pm SRD \cdot r \cdot x_{j,d} \tag{16}$$

where r is random number and $x_{j,d}$, $x_{j,d}{}'$ are current and new value of dimension d respectively.

e. If cat is in tracing mode, update the velocity and the position using Equation (17) and Equation (18).

$$v_{k,d}(t+1) = v_{k,d}(t) + r_1 c_1 \left(x_{best,d}(t) - x_{k,d}(t)\right), \quad d = 1, 2, \ldots, D \tag{17}$$

$$x_{k,d}(t+1) = x_{k,d}(t) + v_{k,d}(t+1) \tag{18}$$

where c_1 is acceleration coefficient; r_1 is a random number; $v_{k,d}(t)$, $v_{k,d}(t+1)$, respectively, are current and new velocity; and $x_{k,d}(t)$, $x_{k,d}(t+1)$ are current and new position respectively.

f. Merge the cats from seeking and tracing mode.
g. Evaluate the fitness value and update the best position.
h. Check the termination criterion. If the criterion is reached, then the algorithm is stopped, but if else, back to step c.

The GSA&sCSO Algorithm

The GSA&sCSO hybrid algorithm is a new algorithm that consists of a combination of two algorithms, namely the GSA algorithm and the CSO algorithm. The GSA and CSO algorithms are metaheuristic algorithms which are then combined into one. In the GSA&sCSO algorithm, authors use the entire GSA algorithm mechanism and the CSO seeking algorithm mode to obtain the optimal solution. The GSA algorithm has good global capabilities, while the CSO seeking mode algorithm has good local capabilities. This makes the authors combine the two algorithms. The combination of the two algorithms aims to get an algorithm that is better than the composing algorithm and to get the optimal solution. The following are the steps used in the GSA&sCSO hybrid algorithm:

a. Generate a random position of N agents (X) and initial velocity (V).
b. Change the position (X) to the positional (Y) form using the following Equation (19):

$$y_{k,d} = \left[x_{k,d} \times (ub_d - lb_d)\right] + lb_d, \quad d = 1, 2, \ldots, D \tag{19}$$

c. Check the constraints of each solution by Equation (2)-(5). All solutions obtained must meet the constraints. If the solution obtained exceeds the limit value of the constraint, a penalty will be made to meet the constraints using the Equation (20):

$$x_{k,d} = x_{k,d} - \frac{1}{ub_d - lb_d} \tag{20}$$

d. Evaluate the fitness function. This step is used to determine the quality of each solution.
e. Initial best storage solution and initial worst. The best and worst values are determined by the fitness value and are used to calculate the mass of inertia.
f. Calculate the mass of inertia. The agent acceleration can be calculated using the heaviest inertia mass because it is the most efficient.
g. Calculate the euclid distance between agents. Agent acceleration is also calculated using the euclideal distance between agents.
h. Calculate acceleration and velocity. Acceleration is used to update the speed, while velocity is used to update the agent's position.
i. Update agent position. New agent positions are used to obtain new solution values.
j. Seeking mode. The steps in seeking mode can be seen in Equation (16).
k. Examination of termination criteria. Maximum iteration is used for dismissal criteria.

COMPUTATIONAL RESULT

Experimental Design

In this study, the authors randomly generated 20 MBKMC of data with some addition rules as follows. (1) The number of items used is 30, 40, 50, 60, 70, 80, 90, 100, 110 and 120. (2) The minimal number and maximal number of each item were randomly generated in [1, 20]. And, (3) Each item has weight, volume, cost, and profit that were randomly generated according to real conditions. Those items in the data will be placed into the knapsack according to the knapsack capacity and produce optimal solutions with maximum profits. The proposed algorithm has some examples using different problems. Because the problem is adjusted to everyday life, that is the purchase price of item is uncertain, each data will be divided into two types, that is min data and max data. Min data describes the minimum cost value and max data describes the maximum cost value. Some of the parameters used in finding the optimal solution in the GSA&sCSO algorithm are described in the Table 1:

TABLE 1. The Parameter of GSA&sCSO Algorithm

Data	$Npop$	G_0	α	ε	MR	SMP	CDC	SRD	c	$Imax$
MBKMC_30							0.2			
MBKMC_40							0.15			
MBKMC_50							0.12			
MBKMC_60							0.1			
MBKMC_70	100	1	1	0.0001	0.5	5	0.085	0.2	2	2000
MBKMC_80							0.075			
MBKMC_90							0.067			
MBKMC_100										
MBKMC_110							0.06			
MBKMC_120										

Result and discussion

To run the algorithm, the author uses a tool in the form of MATLAB and is operated on a personal computer with an Intel (R) Celeron (R) CPU N2840 @2.16Hz 2GB RAM. Algorithms are used to solve each data in 10 experiments. For examples of the purposes obtained from the process running a program shows in Figure (1).

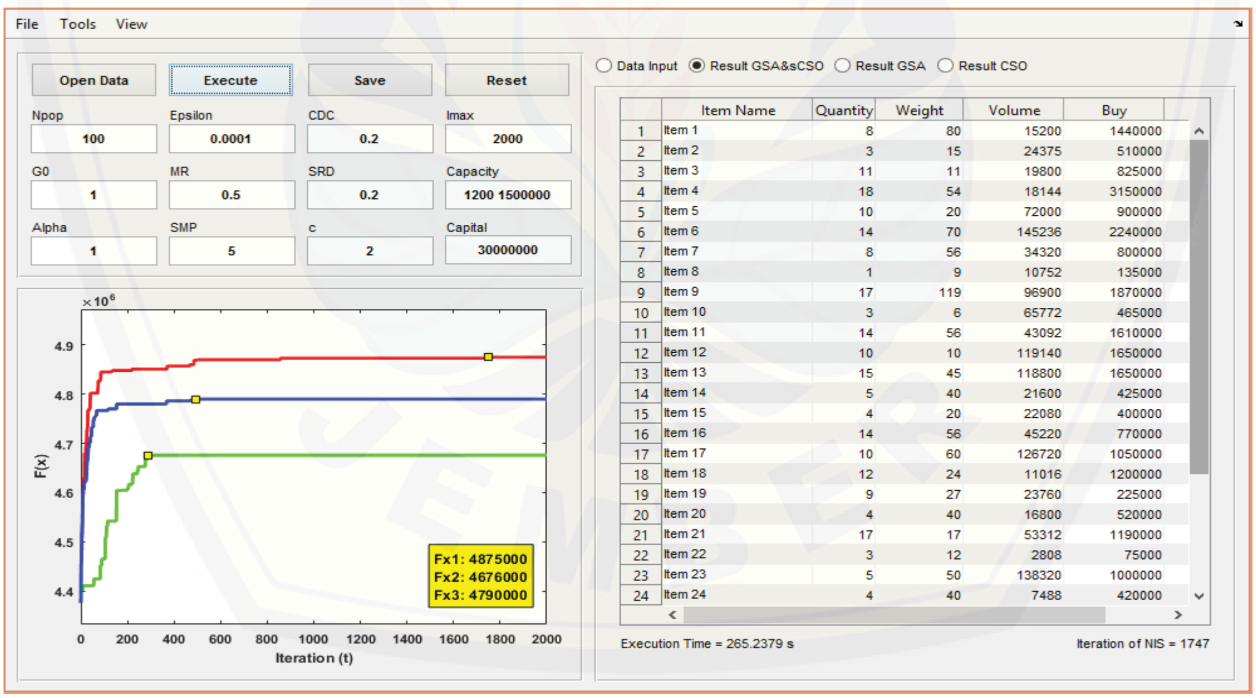

FIGURE 1. Graph of Data

Can be seen in Figure 1, the highest graph is obtained GSA&sCSO, followed CSO and GSA. The red line represents the hybrid algorithm GSA&sCSO, the blue line represents CSO and the green line represents GSA. In the yellow box, we can see the profits of using each algorithm. In addition, on the right side there is data of items that will be entered into the knapsack.

The best trial value will serve as a function of the objective. The calculation results of the GSA & sCSO algorithms are also compared with the GSA and CSO algorithms. The best final results will be shown in Table 2 and the average results will be shown in Table 3.

TABLE 2. The Best Result Obtained

Problem Set	Opt	Z		
		GSA&sCSO	GSA	CSO
MBKMC_30 max	3.995.000	3.985.000	3.857.000	3.977.000
MBKMC_30 min	4.919.000	4.907.000	4.765.000	4.867.000
MBKMC_40 max	5.037.000	5.034.000	4.807.000	5.016.000
MBKMC_40 min	6.349.000	6.344.000	6.110.000	6.297.000
MBKMC_50 max	6.950.000	6.934.000	6.684.000	6.884.000
MBKMC_50 min	8.542.000	8.526.000	8.225.000	8.450.000
MBKMC_60 max	8.537.000	8.501.000	8.200.000	8.427.000
MBKMC_60 min	10.701.000	10.633.000	10.258.000	10.504.000
MBKMC_70 max	10.143.000	10.024.000	9.551.000	9.870.000
MBKMC_70 min	12.427.000	12.255.000	11.844.000	12.188.000
MBKMC_80 max	11.043.000	10.912.000	10.211.000	10.721.000
MBKMC_80 min	13.851.000	13.702.000	12.999.000	13.446.000
MBKMC_90 max	11.843.000	11.708.000	11.112.000	11.504.000
MBKMC_90 min	14.742.000	14.588.000	13.757.000	14.364.000
MBKMC_100 max	13.849.000	13.681.000	12.959.000	13.403.000
MBKMC_100 min	17.033.000	16.830.000	15.935.000	16.563.000
MBKMC_110 max	14.492.000	14.237.000	13.139.000	13.875.000
MBKMC_110 min	17.891.000	17.566.000	16.514.000	17.214.000
MBKMC_120 max	15.112.000	14.819.000	13.397.000	14.507.000
MBKMC_120 min	18.916.000	18.528.000	16.803.000	18.061.000

TABLE 3. The Average Result Obtained

Problem Set	Opt	Z		
		GSA&sCSO	GSA	CSO
MBKMC_30 max	3.995.000	3.976.400	3.813.600	3.949.200
MBKMC_30 min	4.919.000	4.893.900	4.662.600	4.828.900
MBKMC_40 max	5.037.000	5.023.300	4.719.800	4.950.800
MBKMC_40 min	6.349.000	6.332.200	5.971.200	6.231.100
MBKMC_50 max	6.950.000	6.917.200	6.508.700	6.799.000
MBKMC_50 min	8.542.000	8.513.300	8.082.600	8.354.500
MBKMC_60 max	8.537.000	8.481.000	8.015.900	8.316.800
MBKMC_60 min	10.701.000	10.619.400	10.074.800	10.425.500
MBKMC_70 max	10.143.000	9.995.700	9.330.300	9.776.500
MBKMC_70 min	12.427.000	12.229.300	11.509.000	12.043.100
MBKMC_80 max	11.043.000	10.885.500	10.036.300	10.546.800
MBKMC_80 min	13.851.000	13.632.100	12.675.300	13.366.900
MBKMC_90 max	11.843.000	11.672.900	10.855.600	11.341.900
MBKMC_90 min	14.742.000	14.539.100	13.567.900	14.165.000

Problem Set	Opt	Z		
		GSA&sCSO	GSA	CSO
MBKMC_100 max	13.849.000	13.647.000	12.639.300	13.293.200
MBKMC_100 min	17.033.000	16.780.200	15.481.500	16.278.900
MBKMC_110 max	14.492.000	14.190.200	12.911.300	13.557.900
MBKMC_110 min	17.891.000	17.532.700	16.113.100	16.823.400
MBKMC_120 max	15.112.000	14.738.300	13.272.600	14.104.500
MBKMC_120 min	18.916.000	18.441.400	16.678.800	17.778.400

Based on the results that have been obtained in Table 2, it is known that the best results are found in the GSA&sCSO algorithm. The results of the GSA&sCSO algorithm can produce optimal values with highest profit in each data. The average results from 10 trials in each data in Table 3, show that the best result is the GSA&sCSO algorithm, followed by the CSO algorithm then the GSA algorithm. The improvement in the quality of the results of GSA combined with sCSO is very significant.

Apart from that, we can also see that the profit of each min and max data is different even though it uses the same many items. This is because the coefisien value is uncertain in the data. Min data shows a higher profit result than max data, because min data carries more items than max data. The smaller the cost of an item, the higher quantity will be transported.

To evaluate the improvement of the approach, the Equation (21) is used to calculate the precentage of deviation (PD). The Better result will be described through the smaller PD, because it is approaching the optimal value.

$$PD = \frac{Opt - Z}{Opt} \times 100\% \tag{21}$$

The optimal value (Opt) and Z denotes the result of algorithm. The percentage of deviation of the algorithms for each problem are shown in Table 4.

TABLE 4. The Percentage of Deviation

Problem Set	PD of Best Result			PD of Average Result		
	GSA&sCSO	GSA	CSO	GSA&sCSO	GSA	CSO
MBKMC_30 max	0,2503%	3,4543%	0,4506%	0,4656%	4,5407%	1,1464%
MBKMC_30 min	0,2440%	3,1307%	1,0571%	0,5103%	5,2124%	1,8317%
MBKMC_40 max	0,0596%	4,5662%	0,4169%	0,2720%	6,2974%	1,7113%
MBKMC_40 min	0,0788%	3,7644%	0,8190%	0,2646%	5,9411%	1,8570%
MBKMC_50 max	0,2302%	3,8273%	0,9496%	0,4719%	6,3496%	2,1727%
MBKMC_50 min	0,1873%	3,7111%	1,0770%	0,3360%	5,3781%	2,1950%
MBKMC_60 max	0,4217%	3,9475%	1,2885%	0,6560%	6,1040%	2,5794%
MBKMC_60 min	0,6355%	4,1398%	1,8409%	0,7625%	5,8518%	2,5745%
MBKMC_70 max	1,1732%	5,8365%	2,6915%	1,4522%	8,0124%	3,6133%
MBKMC_70 min	1,3841%	4,6914%	1,9232%	1,5909%	7,3871%	3,0892%
MBKMC_80 max	1,1863%	7,5342%	2,9159%	1,4262%	9,1162%	4,4933%
MBKMC_80 min	1,0757%	6,1512%	2,9240%	1,5804%	8,4882%	3,4951%
MBKMC_90 max	1,1399%	6,1724%	2,8625%	1,4363%	8,3374%	4,2312%
MBKMC_90 min	1,0446%	6,6816%	2,5641%	1,3763%	7,9643%	3,9140%
MBKMC_100 max	1,2131%	6,4265%	3,2204%	1,4586%	8,7349%	4,0133%
MBKMC_100 min	1,1918%	6,4463%	2,7593%	1,4842%	9,1088%	4,4273%
MBKMC_110 max	1,7596%	9,3362%	4,2575%	2,0825%	10,9074%	6,4456%
MBKMC_110 min	1,8166%	7,6966%	3,7840%	2,0027%	9,9374%	5,9672%
MBKMC_120 max	1,9389%	11,3486%	4,0034%	2,4729%	12,1718%	6,6669%
MBKMC_120 min	2,0512%	11,1704%	4,5200%	2,5090%	11,8270%	6,0140%
Average	**0,9541%**	**6,0017%**	**2,3163%**	**1,2306%**	**7,8834%**	**3,6219%**

Table 4 showed that the percentage of deviation the GSA&sCSO, GSA and CSO algorithm. With the same parameter setting, the greater size of the data (numbers of items) is possible to obtain a greater percentage of deviation. This result means that the solution is getting further from the optimal solution. That is because greater size of data has

has more possible solutions. Based on the percentage of deviation in 20 data, it can be seen that the best result is obtained by GSA&sCSO algorithm with the average equals to 0,9541% and 1,2306%.

CONCLUSIONS

In this paper, the hybrid Gravitational Search Algorithm and seeking Cat Swarm Optimization (GSA&sCSO) was combined of two metaheuristic algorithms, namely Gravitational Search Algorithm (GSA) and Cat Swarm Optimization (CSO). The GSA&sCSO algorithm has been proposed to solve Modified Bounded Knapsack Multiple Constraints (MBKMC) problem against uncertain coefficient which is the cost of each item. Based on the computational results, it is known that the GSA&sCSO algorithm provides a better solution than the GSA and the CSO. The GSA&sCSO obtained the highest profit compared to the GSA and the CSO in 20 MBKMC data. The solution obtained has been tested and produces a good PD value. Thus, the GSA&sCSO is a new hybrid algorithm that is good to use.

ACKNOWLEDGMENTS

The authors would like to say many thanks to anonymous reviewer who have contributed enormously to this research. Other than that the authors would also like to thanks Mathematics Departement, Faculty of Science, University of Jember.

REFERENCES

1. K. Diah, M. Fadhli and C. Sutanto, “Penyelesaian knapsack problem menggunakan algoritma genetika” in *Seminar Nasional Informatika 2010,* (Yogyakarta, 2010), pp. 28-33.
2. D. Pisinger,A Minimal Algorithm for the Multiple-Choise Knapsack Problem. European Journal of Operasional Research, pp. 394-410 (1995).
3. H. Kellerer, D. Pisinger and U. Pferschy, *Knapsack Problem* (Springer, Berlin, 2004).
4. A. Baihaki, Penerapan Algoritma Cuckoo Search (CS) pada Permasalahan Multiple Constraints Knapsack 0-1, (Universitas Jember, 2017).
5. Y. D. Regita, Penerapan Algoritma Elephant Herding Optimization pada Permasalahan Multiple Constraints Knapsack 0-1, (Universitas Jember, 2019).
6. L. F. M. Lopez, N. G. Blas and A. Albert, Multidimensional Knapsack Problem Optimization Using a Binary Particle Swarm Model with Genetic Operation. Soft Computing, pp. 2567-2582 (2018).
7. R. J. M. Putri, Penerapan Algoritma Modified Variable Neighborhood Search pada Multi Knapsack 0-1 Problem with Multiple Constraints, (Universitas Jember, 2018).
8. M. A. Mahendra, Penerapan Whale Optimization Algorithm pada Permasalahan Multi Knapsack Multiple Constraints, (Universitas Jember, 2018).
9. Wafirotullailiyah, Penerapan Hybrid Virus Evolutionary Genetic Algorithm (hVEGA) pada Bounded Multiple Knapsack, (Universitas Jember, 2016).
10. R. B. Kasyfi, Penerapan Algoritma Centripetal Accelerated Particle Swam Optimization pada Permasalahan Multiple Bounded Knapsack, (Universitas Jember, 2018).
11. V. Lestari, Penerapan Algoritma Gray Wolf Optimizier pada Permasalahan Multiple Constraints Bounded Knapsack, (Universitas Jember, 2019).
12. A. M. Gandomi, X. S. Yang, S. Talatahari and A. H. Alavi, Metaheuristic Algorithm in Modeling and Optimization. Metaheuristic Applications in Structures and Infrastructures, pp. 1-24 (2013).
13. I. Boussaid, J. Lepagont and P. Siarry. A survey on optimization metaheuristics. Information Sciences, pp. 82-117 (2013).
14. E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, GSA a Gravitational Search Algorithm. Information Sciences, pp. 2232-2248 (2009).
15. S. C. Chu, P. W. Tsai and J. S. Pan, *Cat Swarm Optimization LNAI 4099* (Heidelberg, Berlin, 2006)
16. R. Febriana, Aplikasi Gravitational Search Algorithm untuk penentuan Parameter Sensitivitas dan Efektifitas Glukosa pada Oral Minimal Model Termodifikasi, (FMIPA Institut Pertanian Bogor, Bogor, 2016).

17. A. A. Nagra, H. Fei, Q. Ling and S. Mehta, An Improved Hybrid Method Combining Gravitational Search Algorithm with Dynamic Multi Swarm Particle Swarm Optimization, (China, 2019), pp. 3088-3099.