Bulletin of Applied Mathematics and Mathematics Education
Volume 5, Number 2, October 2025 pp. 103-112
https://dx.doi.org/10.12928/bamme.v5i2.14728
e-ISSN 2776-1029, p-ISSN 2776-1002

Optimizing stock allocation and profit in MSMEs: Multiple constraints bounded Knapsack model solved using Grey Wolf Optimizer algorithm

Mufarrida Dalilah*, Hendra Cipta

Faculty of Science and Technology, Universitas Islam Negeri Sumatera Utara, Jl. Lapangan Golf, Desa Durian Jangak 20353, Medan, Indonesia

*Corresponding e-mail: mufarrida0703212062@uinsu.ac.id

ARTICLE INFO

Article History
Received 10 October 2025
Revised 27 October 2025
Accepted 27 October 2025

Keywords
GWO algorithm
Knapsack problem
MSMEs
Multiple constraints bounded
Stock optimization

How to cite this article:
Dalilah, M., & Cipta, H. (2025). Optimizing stock allocation and profit in MSMEs: Multiple constraints bounded Knapsack model solved using Grey Wolf Optimizer algorithm. *Bulletin of Applied Mathematics and Mathematics Education*, *5*(2), 103-112.

ABSTRACT

Effective inventory management is a determining factor in the probability and sustainability of micro, small, and medium enterprises (MSMEs). Adjusting the ideal stock of each product type that has to be distributed while taking perishable items, storage capacity constraints, and client demand unpredictability into account is a difficulty. Stock allocation must maximize profit while adhering to intricate constraints and particular item number limitations in the multiple-constraints Knapsack problem. This research aims to apply the Grey Wolf Optimizer (GWO) algorithm to the multiple constraints bounded Knapsack problem for optimal stock allocation while increasing profitability for MSMEs by comparing the ideal value of the simplex technique. The population parameter (Npop) and the maximum iteration (Max Iter) were the two parameters used to test the GWO method. According to sensitivity analysis, the GWO algorithm optimization study was less successful in producing the best outcomes. This resulted from a discrepancy between the simplex method's IDR 9,508,000 profit optimization and GWO's IDR 9,440,000. Nonetheless, the GWO method was almost ideal, as indicated by the deviation percentage of 0.7152%. The study highlights the applicability of metaheuristic optimization for MSME management inventory, offering a near-optimal solution with minimal deviation from analytical results. Limitations include the single-case scope and parameter sensitivity of the GWO algorithm.

Introduction

Productive companies that are controlled by individuals or corporate entities who fit the requirements of having a comparatively small workforce, a modest company size, a low turnover rate, and typically being founded with little funding make up the Micro, Small, and Medium Enterprises (MSME) sector (Munthe et al., 2023). As demonstrated by their labor absorption, foreign exchange contribution, and GDP (Gross Domestic Product) contribution, which, according to 2021 data, reached 60.5% and accounted for 96.9% of the total national labor absorption—

MSMEs are essential to the country's economy (Junaidi, 2024). However, significant obstacles are unavoidable for any organization, particularly when it comes to resource management. A company's efficacy and efficiency are significantly impacted by its resource management (Nugroho, 2021). Therefore, suboptimal inventory management directly threatens profitability because it causes excessive storage costs or insufficient storage, resulting in the loss of target markets. In addition, MSMEs also face obstacles related to storage space capacity and business capital, making stock allocation decisions a critical optimization challenge.

The problem of determining the stock composition that yields maximum profit within limited and complex resource constraints can be formulated as a combinatorial optimization problem (Iqbal et al., 2019). A model is needed that can accommodate multiple physical constraints, such as volume and weight, as well as quantity constraints per item (bounded). Therefore, this study models the challenge of inventory allocation in MSMEs as a Multiple Constraints Bounded Knapsack Problem (MCBKP). MCBKP is an appropriate model to represent the highest value load problem and is computationally classified as an NP-Hard problem (Cacchiani et al., 2022). This NP-Hard property causes exact algorithms (definite calculations) to be inefficient and impractical for real-world MSME data scenarios.

Metaheuristics have been used in a number of attempts to solve the challenging Knapsack problem. The most effective techniques for resolving a variety of practical engineering issues are metaheuristics, which draw inspiration from natural evolutionary forms, habits, and simplicity (Mirjalili et al., 2014). As researched by Sapoetra and Habibi (2023) the knapsack problem can be solved using a combination of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), which yielded a minimum error of 1.49% in 24.9 seconds. The knapsack problem in the form of interval scheduling can be solved using the Greedy and Dynamic Programming algorithms, which were researched by Prasha et al. (2024). The results show that both algorithms have their own advantages and disadvantages. Dynamic Programming provides optimal results but takes a long time compared to the Greedy Algorithm, which is faster but does not always provide optimal solutions. Wu (2023) also researched the knapsack problem in solving real goods delivery using Dynamic Programming and the Greedy Algorithm, and found that the greedy algorithm was chosen because of its efficient computation time. Finally, there is research by Nilasari (Nilasari et al., 2019) on the Knapsack problem in bamboo craft shops using the Dragonfly Optimization Algorithm (DOA) solution, which yielded less effective results but was close to the optimal value as proven by simplex method.

These algorithms still often face challenges in balancing exploration and exploitation, which can lead to premature convergence. To address this gap, this study proposes the Grey Wolf Optimizer (GWO) algorithm, a relatively new metaheuristic known for its global exploration mechanism and ease of use (Mirjalili et al., 2014). The GWO algorithm was chosen for its proven ability to handle complex search space optimization with a minimal number of parameters, promising better computational time efficiency (Faris et al., 2018).

Presenting the MCBKP model solution utilizing the GWO Algorithm to accomplish Stock Allocation and Profit Optimization in MSMEs is, thus, the primary goal of this work. The study's demonstration of GWO's efficacy as a dependable and practical computational solution in the setting of limited MSME inventory management is one of its major contributions. It is anticipated that the findings would offer suggestions for the best and most quantifiable stock allocation to increase MSME participants' profitability.

 10.12928/bamme.v5i2.14728

Method

This study is applied research that employs a quantitative approach focused on optimization modeling and computational simulation. The research aims to apply algorithmic solutions to solve multiple constraints in the bounded Knapsack problem—specifically, optimal stock allocation and profits from fruit-selling micro, small, and medium enterprises (MSMEs) in Medan Selayang District, Medan City. The best results of the simplex technique and the Grey Wolf Optimizer algorithm are compared to demonstrate the GWO algorithm's effectiveness. Secondary data are the primary data source and may be obtained indirectly through reports created by individuals or organizations, including tables, diagrams, graphs, and other visuals (Darwin et al., 2021). Daily secondary data collection includes fruit names, volume per package, cost per package, and selling price per package, as shown in Table 1.

Table 1. Data

No.	Fruit	Quantity (pack)	Volume (cm³)	Cost (IDR)	Selling Price (IDR)
1	Apple	10	24,000	400,000	513,000
2	Green Apple	2	34,560	880,000	1,100,000
3	Pear	12	60,000	255,000	323,000
4	Golden Pear	3	21,600	360,000	450,000
5	Star Fruit	2	35,530	150,000	190,000
6	Grape	2	39,990	360,000	470,000
7	Muscat	8	1,500	90,000	120,000
8	Blackcurrant	10	1,500	75,000	100,000
9	Lychee	1	28,560	252,000	315,000
10	Dragon Fruit	3	35,530	160,000	200,000
11	Avocado	15	83,979	336,000	420,000
12	Crystal Guava	4	35,530	140,000	180,000
13	Guava	3	35,530	150,000	190,000
14	Ponkam	2	35,535	380,000	477,000
15	Orange	8	211,219	790,000	980,000
16	Lemon	1	25,000	760,000	952,000
17	Kweni	3	68,726	300,000	380,000
18	Sapodilla	4	35,530	136,000	170,000
19	Snakefruit	4	35,530	140,000	180,000
20	Melon	20	2,105	35,000	45,000
21	Watermelon	80	4,210	28,000	36,000
22	Passion fruit	2	39,990	205,000	258,000
23	Pineaplle	20	2,490	24,000	30,000
24	Soursop	15	2,500	30,000	36,000
25	Papaya	25	1,200	15,000	20,000
26	Thai Mango	3	35,530	420,000	540,000
27	Udang Mango	3	68,726	500,000	630,000
28	Harum Manis Mango	6	24,333	380,000	485,000
29	Sunkist	3	25,000	450,000	568,000
30	Longan	2	28,560	280,000	390,000

Optimization algorithm

Multiple Constraints Bounded Knapsack

The Multiple Constraints Bounded Knapsack Problem (MCBKP) is a combinatorial optimization problem that is NP-hard (representing the hardest problems of Non-deterministic Polynomial (NP) but not necessarily belonging to NP itself) and appears in various applications. The constraints in the MCKP involve objects/items that have more than one dimension, such as weight, cost, and volume; sum of these dimensions cannot be greater than the storage media's capacity (Santoso et al., 2022). This indicates that MCKP is NP-complete, which supports the use of approximation algorithms (Szkaliczki, 2025). The obstacle discussed in the study is the allocation of product stock in a limited storage. The first decision variable is denoted as x_j, which represents the number of items to be allocated to storage. The maximization objective function (Z), where p_j is the net profit of each item. Here is the equation of maximum objective function:

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

This model is related to a series of constraints in the form of physical capacity constraints. There are two restrictions in this study: the total volume for each fruit (v_j) and the cost constraints (b_j). The volume capacity of each shelf in the room is $C_v = 10{,}000{,}000\ cm^3$ and the available cost is $C_M = IDR\ 35{,}000{,}000$. Furthermore, there is a stock limit constraint that limits the maximum quantity for each type of fruit. All of these constraints can be formulated as follows.

$$\sum_{j=1}^{n} v_j x_j \le C_V \tag{2}$$

$$\sum_{j=1}^{n} b_j x_j \le C_M \tag{3}$$

$$x_j \in \{0,1,\dots,m_j\}, \qquad j = 1,\dots,n \tag{4}$$

Grey Wolf Optimizer Algorithm

The Grey Wolf Optimizer (GWO) algorithm is a swarm intelligence-based metaheuristic optimization method inspired by the social hierarchy and hunting behavior of gray wolves. Gray wolves are recognized as apex predators in the wild and naturally live in pack comprises 5 to 12 wolves and maintains a strict social hierarcy: it is lead by an alpha (α) wolf, (the dominant individual, either male or female), followed by the subordinate beta (β) wolves, the third-ranking delta (δ) wolves, and finally the lowest-ranking omega (ω) wolves (Mirjalili et al., 2014). The equation model for the GWO algorithm is based on the prey attack technique (Nahor et al., 2023).

Encircling Prey

This is the equation model obtained when the grey wolf surrounds the prey:

$$\vec{D} = \left| \vec{C}.\vec{X}_p(t) - \vec{X}(t) \right| \tag{5}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{6}$$

where t is the current iteration, $\vec{X}$ is the position vector of the gray wolf, $\vec{X}_p$ is the position vector of the prey, $\vec{A}$ and $\vec{C}$ are coefficient vectors. The parameter $\vec{A}$ tends to exploit solutions in local optima, so a parameter strategy $\vec{C}$ is needed to overcome these local optima traps, especially at the end of the iteration (Ardiansyah et al., 2025) in order to obtain the best optimum region. Before proceeding, parameter ɑ must first be determined. This value is designed to decrease linearly from 2 to 0 across the iterations (Azis et al., 2025), while $r1$ and $r2$ are random numbers between [0, 1]. calculated using the following equations.

$$a(t) = 2 - \frac{2t}{T_{\max}} \tag{7}$$

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \tag{8}$$

10.12928/bamme.v5i2.14728

$$\vec{C} = 2.\vec{r}_2 \tag{9}$$

Hunting

α (alpha) serves as the hunting leader, while β (beta) and δ (delta) in pursuing the prey once it has been encircled (Qiu et al., 2024). Referring to the social hierarchy, where α (alpha) is the first-best option, β (beta) is the second, and δ (delta) is the third. The prey's escape during this process results in a shift in the gray wolf population's location. The search for the optimal hunting location is modeled in the following equation.

$$\vec{D}_\alpha = \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}\right| \text{ , } \vec{D}_\beta = \left|\vec{C}_2.\vec{X}_\beta - \vec{X}\right| \text{ , } \vec{D}_\delta = \left|\vec{C}_3.\vec{X}_\delta - \vec{X}\right| \tag{10}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.(\vec{D}_\alpha) \text{ , } \vec{X}_2 = \vec{X}_\beta - \vec{A}_2.(\vec{D}_\beta) \text{ , } \vec{X}_3 = \vec{X}_\delta - \vec{A}_3.(\vec{D}_\delta) \tag{11}$$

$$\vec{X}(\text{t}+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{12}$$

Data processing

Based on the method description, application steps of the Grey Wolf Optimizer (GWO) algorithm to solve the multiple constraints bounded Knapsack problem are outlined as follows.

After collecting the necessary data, initialized with the input variables: fruit, population (Pop), maximum iteration (Max Iter), volume (v_j), cost (b_j), and price, random numbers are subsequently generated corresponding to the number of dimensions and the population. The initial results for v_j, b_j and are verified using equations (2) and (3).

Next, the fitness value is calculated. This involves determining the total profit for each fruit quantity data point using equation (1). From the computed fitness values, the three highest are identified, which correspond to the alpha (α), beta (β), and delta (δ)wolves. These three wolves will then lead the rest of the pack in hunting the prey (optimization) (Yosviansyah & Rizal, 2024). Parameters A and C are then computed, but not before figuring out the maximum iteration value (T_{max}) to get the value of parameter α using equation (7), then finding random values between [0,1] for r_1 and r_2, and next using equations (8) and (9). The position of each wolf in the pack is then updated during the hunting process using equations (10) and (11), resulting in the updated position calculated by Equation (12). Finally, the fitness value—the most recent total profit for each fruit—is calculated, representing the current solution found by the wolf pack's pursuit. The alpha (α) value is regarded as the best optimization result. If the alpha value has not yet reached the desired solution, additional iterations are required. The process is then repeated, starting from the step of identifying the alpha, beta, and delta values until the most optimal result is achieved.

To demonstrate the effectiveness of the Grey Wolf Optimizer algorithm in solving the multiple-constraint bounded knapsack problem, a comparison is performed using the Simplex method with the aid of the Microsoft Solver Add-in. The final optimal results obtained from the GWO algorithm and the Simplex method can then be compared by calculating the percentage deviation using the following equation.

$$\%Dev = \frac{simplex - Z_i}{simplex} \times 100\% \tag{13}$$

Figure 1 details the methodology employed to solve the multiple cnstraints bounded Knapsack problem at Syariah Buah using the Grey Wolf Optimizer (GWO).

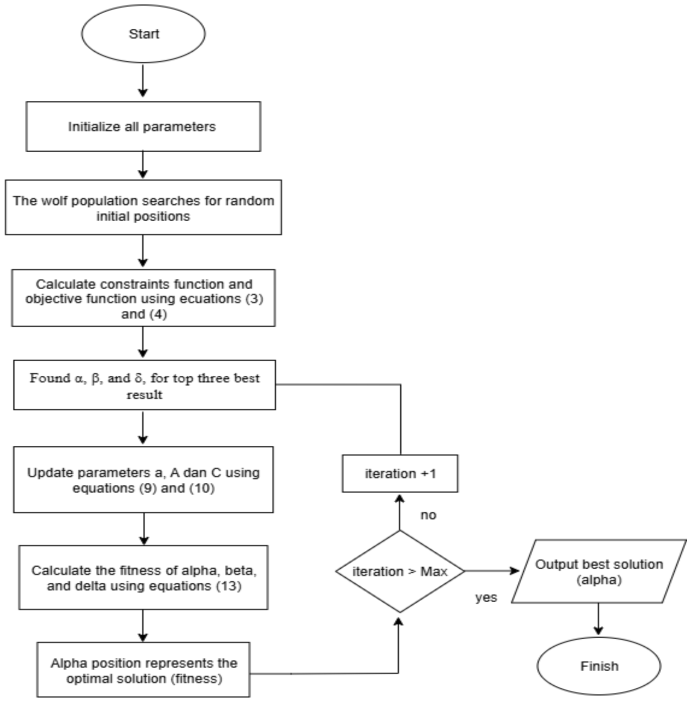

Figure 1. MCBK-GWO Flowchart

Results and discussion

The Grey Wolf Optimizer (GWO) algorithm can be applied to multiple constraints bounded knapsack problems in MSME stock and profit allocation, which provides near-optimal results. Because the data being examined is extensive, the GWO technique uses MATLAB R2025b software to execute computational simulations in its application, reducing the possibility of calculation mistakes in comparison to human computations. Because it only requires two test parameters—population parameters and maximum iteration parameters—the GWO method was selected to tackle multiple constraints bounded knapsack problem because it facilitates the conclusion of unambiguous optimization outcomes. Its complicated test parameters cause it to take a lengthy time to acquire the optimization results when compared to other metaheuristic algorithms such as Particle Swarm Optimizer (PSO), Dragonfly Algorithm Optimizer (DOA), and Genetic Algorithm (GA). The convergence graph in Figure 2. displays the ideal value that was found after evaluating the population and maximum iteration parameters.

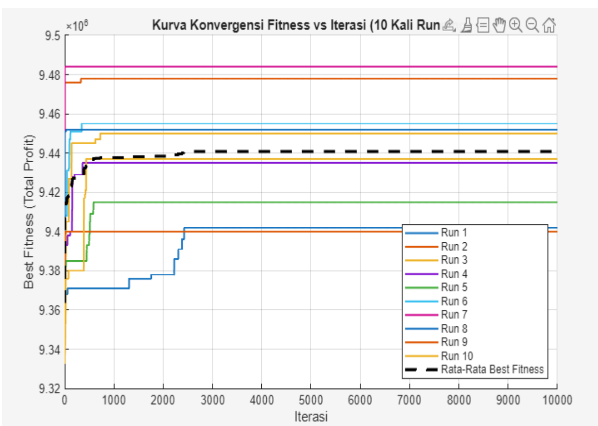

Figure 2. Plot of convergent (fitness vs iterations)

10.12928/bamme.v5i2.14728

Population parameter test

Following the setup, a simulation program code was developed to test the collected data. The testing focused on two critical parameters: population size (Npop) and maximum iterations (Max Iter). Five distinct values 30, 100, 250, 500, and 1000 were chosen for the population size comparison. For consistency, the maximum iteration (Max Iter) was fixed at 1000 across all trials. These values were taken based on sensitivity analysis and inspired by Lestasi (Lestari et al., 2019), whose research used high parameter values from large data sets to obtain the most optimal results. Each unique combination of these parameters was executed ten times; if only one calculation was performed, it would be stuck in a local optimum and the optimal result would not be achieved. The key performance indicators measured were average profit, average convergent iteration, and average computation time. These comprehensive results are summarized in Table 2. The best profit obtained on NPop 1000 was 9,484,000 with an average convergence iteration of 561. In determining fitness, convergence iteration is very important to consider because the smaller the value, the closer it is to be optimal.

Table 2. Population parameter test result

NPop	Average Profit	Best Profit	Worst Profit	Average Convergent Iteration	Average Computing Time
30	9,391,700	9,424,000	9,321,000	820.3	0.32171
100	9,445,700	9,475,000	9,363,000	710	0.99037
250	9,444,400	9,480,000	9,414,000	625.5	2.6924
500	9,447,100	9,480,000	9,393,000	530.6	5.2194
1000	9,467,700	9,484,000	9,431,000	561	10.939

Maximum iteration parameter test

Next, in the maximum iteration (Max Iter), five comparative values were used: 1000, 2000, 5000, 7000, and 10000. Similar to the population parameter, the values chosen for the maximum iteration parameter are in agreement with the sensitivity analysis. This test utilized a population parameter of 1000, as this value represented the population size that yielded the best average profit during the previous testing phase. Similar to the population parameter test, each parameter value combination was executed ten times (ten program runs). The key performance indicators measured were average profit, average convergent iteration, and average computation time. The results of this testing are presented in Table 3.

Table 3. Maximum iteration parameter test result

Max Iter	Average Profit	Best Profit	Worst Profit	Average Convergent Iteration	Average Computing Time
1000	9,391,000	9,424,000	9,399,000	668.6	5.2717
2000	9,445,700	9,475,000	9,395,000	1081.2	10.227
5000	9,444,400	9,480,000	9,426,000	2584	25.607
7000	9,447,100	9,480,000	9,445,000	3168.2	37.273
10000	9,484,000	9,484,000	9,424,000	4094.1	50.484

After concluding the tests on both the population and maximum iteration parameters, a final validation run of the GWO algorithm was performed. This ultimate test used the parameters deemed most robust: a population size of 1000 and a maximum iteration count of 10,000. To ensure the reliability of the most optimal outcome, this combination was executed ten times. Subsequently, the step requires calculating the percentage deviation to quantitatively assess how effectively the

Grey Wolf Optimizer (GWO) performs against the Multiple Constraints Bounded Knapsack Problem. This deviation analysis, presented in Table 4, utilizes Equation (14) and compares the GWO results against the known optimal profit yielded by the Simplex method--which was determined using the Microsoft Excel Solver add-in to be IDR 9,508,000-,.

Table 4. Final result with percentage deviation

No.	Cost	Best Profit	Convergent Iteration	Computing Time (sec)	Deviation (%)
1	34,999,000	9,381,000	7031	68.74	1.3357
2	35,000,000	9,439,000	8355	80.25	0.7257
3	35,000,000	9,415,000	7471	75.21	0.9781
4	34,997,000	9,386,000	8148	77.30	1.2831
5	**35,000,000**	**9,440,000**	**7732**	**74.30**	**0.7152**
6	35,000,000	9,415,000	7628	72.08	0.9781
7	34,990,000	9,412,000	6537	62.24	1.0097
8	34,999,000	9,394,000	6243	59.56	1.1990
9	34,800,000	9,340,000	6153	58.71	1.7669
10	34,999,000	9,416,000	8195	76.74	0.9676
	Average	9,403,800	73492	70.4617	1.0959

Based on the final results using a population size of 1000 and 10,000 maximum iteration, GWO algorithm achieved a solution that closely approached the theoretical optimal value. Specifically, the highest profit recorded was IDR 9,440,000 in the fifth run, yielding a percentage deviation of 0.7152%. Conversely, the smallest profit recorded was IDR 9,340,000, with a percentage deviation of 1.7669%. The average profit across all runs was found to be IDR 9,403,000 with an average convergent iteration count of 73492.

Increasing the parameters for population size and maximum iterations does not guarantee a superior optimal solution. This outcome is attributed to several factors, including the algorithm becoming trapped in a complex solution space and issues concerning the balance between exploration and exploitation. Consequently, the optimal solution is determined not solely by the quantity of the parameters but rather by the suitability of the algorithm to the specific problem.

The best-found solution by the Grey Wolf Optimizer (GWO) algorithm for the Multiple Constraints Bounded Knapsack Problem resulted in a maximum profit of IDR 9,440,000, this solution utilizes a total volume of 5,113,703 cm^3 and dictates an optimal stock of 235 packages across 29 types of fruit, detailed as follows: 10 boxes of 13.5 kg apples, 2 boxes of 18 kg green apples, 12 boxes of 17 kg pears, 3 boxes of 9 kg golden pears, 1 basket of 10 kg star fruit, 2 baskets of 6 kg grapes, 8 packages of muscat, 10 packages of blackcurrant, 1 basket of 5 kg lychees, 3 baskets of 10 kg dragon fruit, 15 baskets of 20 kg avocados, 4 baskets of 10 kg crystal guava, 3 baskets of 10 kg common guava, 1 box of 9 kg Ponkam, 3 baskets of 55 kg oranges, 1 box of 14 kg lemons, 3 baskets of 20 kg kuweni, 2 baskets of 10 kg sapodilla, 4 baskets of 10 kg snake fruit, 19 pieces of melon, 80 pieces of watermelon, 2 baskets of 6 kg passion fruit, 4 pieces of pineapple, 25 pieces of papaya, 3 baskets of 10 kg Thai mango, 3 baskets of 20 kg Udang mango, 6 boxes of 10 kg Harum Manis mango, 3 boxes of 14 kg Sunkist, and 2 baskets of 5.5 kg longan.

Conclusion

In accordance with the results and discussion presented, the Grey Wolf Optimizer (GWO) Algorithm yielded a profit of IDR 9,440,000 as the optimum solution for the Multiple Constraints Bounded Knapsack Problem with a total volume of 5,113,703 cm^3 and resulted in an optimal fruit stock of

 10.12928/bamme.v5i2.14728

235 packages covering 29 types of fruit. According to sensitivity analysis this best-found profit was achieved using final testing parameters of population sizes (Pop) of 1000 and a maximum iteration (Max Iter) of 10,000, with convergence occurring at 7732 iterations. However, the findings caution that optimality is not solely a function of increased parameter quantity. This algorithm's performance is significantly influenced by intrinsic factors, such as the risk of becoming stuck in local optima and the balance between exploration and exploitation. Thus, achieving an optimal solution relies primarily on the algorithm's structural suitability to the problem. While GWO's optimal profit showed a quantitative difference compared to the Simplex method (calculated via the Microsoft Excel Solver add-in) maximum profit of IDR 9,508,000, the result percentage deviation was sufficiently small. This outcome confirms that the GWO algorithm provides a reliable and close approximation to the theoretical optimal point. Future research should explore hybrid or adaptive versions of GWO algorithm and compare their performance with other metaheuristic algorithms on larger-scale Multiple constraints bounded knapsack problem datasets to generalize these findings.

References

Ardiansyah, Handayaningsih, S., & Fathurrizki, D. (2025). Grey Wolf Optimizer termodifikasi menggunakan Chaotic Uniform Initialization untuk estimasi Effort Cocomo. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *12*(3), 671–680. https://doi.org/10.25126/jtiik.20258901

Azis, A. I. S., Santoso, , Budy, & Jeffry, J. (2025). Penerapan Grey Wolf Optimizer dalam pelatihan multi layer Perceptron untuk menangani masalah klasifikasi dan regresi. *Advances in Computer System Innovation Journal*, *2*(3), 108–118. https://doi.org/10.51577/acsijournal.v2i3.653

Cacchiani, V., Iori, M., Locatelli, A., & Martello, S. (2022). Knapsack problems — An overview of recent advances. Part II: Multiple, multidimensional, and quadratic Knapsack problems. *Computers and Operations Research, 143*, 105693. https://doi.org/10.1016/j.cor.2021.105693

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. *Neural Computing and Applications*, *30*(2), 413–435. https://doi.org/10.1007/s00521-017-3272-5

Iqbal, M., Zarlis, M., Tulus, T., & Mawengkang, H. (2019). Pendekatan Pengembangan Metaheuristik dalam optimisasi kombinatorial. *Prosiding Seminar Nasional Riset Information Science (SENARIS)*, *1*, 1193. https://doi.org/10.30645/senaris.v1i0.135

Junaidi, M. (2024). *UMKM Hebat, Perekonomian Nasional Meningkat*. https://djpb.kemenkeu.go.id/kppn/curup/id/data-publikasi/artikel/2885-umkm-hebat,-perekonomian-nasional-meningkat.html

Lestari, V., Kamsyakawuni, A., & Santoso, K. A. (2019). Implementasi algoritma Grey Wolf Optimizer (GWO) di Toko Citra Tani Jember. *Majalah Ilmiah Matematika dan Statistika, 19*, 65-74.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Munthe, A., M. Yarham, & Siregar, R. (2023). Peranan usaha mikro kecil menengah terhadap perekonomian Indonesia. *Jurnal Ekonomi Bisnis, Manajemen dan Akuntansi*, *2*(3), 593–614. https://doi.org/10.61930/jebmak.v2i3.321

Nahor, K. E. B., Zebua, O., & Hakim, L. (2023). Penentuan lokasi dan kapasitas kapasitor bank pada jaringan distribusi penyulang americano untuk meminimalkan rugi-rugi daya dengan metode Grey Wolf Optimizer (GWO). *Electrician : Jurnal Rekayasa dan Teknologi Elektro*, *17*(1), 100–107. https://doi.org/10.23960/elc.v17n1.2376

Nilasari, L. F., Santoso, K. A., & Riski, A. (2019). Penerapan Dragonfly Optimization Algorithm (DOA) pada permasalahan multiple constraints bounded Knapsack. *Majalah Ilmiah Matematika dan Statistika, 19*(1). https://doi.org/10.19184/mims.v19i1.17264

Nugroho, A. Y. (2021). *Sistem dan Teknik Manajemen Inventory.* Gemilang Press Indonesia.

Prasha, A. A., Rachmadi, C. O., Sari, A. P., Raditya, N. G., Mutiara, S. L., & Yusuf, M. (2024). Implementasi algoritma Greedy dan dynamic programming untuk masalah penjadwalan interval dengan model Knapsack. *Format: Jurnal Ilmiah Teknik Informatika*, *13*(2), 166. https://doi.org/10.22441/format.2024.v13.i2.005

Qiu, Y., Yang, X., & Chen, S. (2024). An improved grey wolf optimization algorithm solving to functional optimization and engineering design problems. *Scientific Reports*, *14*(1), 1–24. https://doi.org/10.1038/s41598-024-64526-2

Santoso, K. A., Kurniawan, M. B., Kamsyakawuni, A., & Riski, A. (2022). Hybrid cat-particle swarm optimization algorithm on bounded Knapsack problem with multiple constraints. *Proceedings of the International Conference on Mathematics, Geometry, Statistics, and Computation (IC-MaGeStiC 2021)*, *96*, 244–248. https://doi.org/10.2991/acsr.k.220202.045

Sapoetra, Y. A., & Habibi, A. R. (2023). Multidimensional Knapsack 0-1 solution with algorithm evolution Pso-Ga. *Sinkron*, *8*(4), 2406–2413. https://doi.org/10.33395/sinkron.v8i4.12887

Szkaliczki, T. (2025). Solution methods for the multiple-choice Knapsack problem and their applications. *Mathematics*, *13*(7). https://doi.org/10.3390/math13071097

Wu, Y. (2023). Comparison of dynamic programming and greedy algorithms and the way to solve 0-1 knapsack problem. *Applied and Computational Engineering*, *5*(1), 631–636. https://doi.org/10.54254/2755-2721/5/20230666

Yosviansyah, M. N., & Rizal, Y. (2024). Optimasi persediaan bahan baku dan produksi usaha Ganepo Putri Yose dengan menggunakan algoritma Grey Wolf Optimizer. *MATHunesa: Jurnal Ilmiah Matematika*, *12*(1), 94-100.

 10.12928/bamme.v5i2.14728