SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ

Sakarya University Journal of Science
SAUJS

ISSN 1301-4048 e-ISSN 2147-835X Period Bimonthly Founded 1997 Publisher Sakarya University
http://www.saujs.sakarya.edu.tr/

Title: Investigation of The Effect of Feeding Period in Honey Bee Algorithm

Authors: Mustafa KAYA

Recieved: 2021-12-02 00:00:00

Accepted: 2022-08-23 00:00:00

Article Type: Research Article

Volume: 26
Issue: 6
Month: December
Year: 2022
Pages: 1071-1083

How to cite
Mustafa KAYA; (2022), Investigation of The Effect of Feeding Period in Honey Bee Algorithm. Sakarya University Journal of Science, 26(6), 1071-1083, DOI: 10.16984/saufenbilder.1031673
Access link
https://dergipark.org.tr/en/pub/saufenbilder/issue/74051/1031673

Sakarya University Journal of Science 26(6), 1071-1083, 2022

Investigation of The Effect of Feeding Period in Honey Bee Algorithm

Mustafa KAYA*[1]

Abstract

In the study, it was investigated the ejaculation ability and semen quality of drones, according to feeding with pollen in different periods. In the first step of the study, 16 %, 32 %, 47 %, 63 %, 79 %, and 100 % feeding periods were applied to the drones, for investigating the effect on ejaculation ability, and the semen quality of drones was investigated. While investigating these feeding period effects “0-1”, bonded, and unbounded knapsack optimization problems were used. After the most effective feeding period was determined, this period was applied to the traveling salesman and liquid storage tank problems in the second step of the study. In the analysis of the traveling salesman problem, it was determined the shortest way between two cities. Analysis of the liquid storage tank problem, it was determined the minimum connector areas. As a result, the analysis results showed that the performance of the artificial bee colony algorithm is very good while solving too complex engineering optimization problems.

Keywords: Bee colony algorithm, feeding period effect, knapsack problems

1. INTRODUCTION

One of the most recent bee-based algorithms is the Bees Algorithm (BA). This algorithm (BA) is a population-based metaheuristic algorithm proposed by Pham et al. study [1], which is based on the behavior of honeybees that is observed when they are foraging for food. Fundamentally, the algorithm performs a kind of exploitative local or neighborhood search combined with an exploratory global search. Pham and Ghanbarzadeh [2], and solving timetabling problems (Yuce et al. [3], Abdullah et al. [4]. The artificial bee colony algorithm is one of the SI algorithms that has been developed by using waggle dance and foraging behaviors of real colonies (Lara [5]. Dongli et al. [6], proposed three modified versions of ABC in order to better-quality results for the optimization problems. In the first modification, the neighborhood structure changes in the solution updating equation of ABC, in the second modification, a new selection equation is proposed for onlooker bees in order to choose an employed bee and the last modified version of ABC is based on modifications 1, and 2. Banharnsakun et. al. [7], presented a modified method for solution update of the onlooker bees in this study. In their method, the best feasible solutions found so far are shared globally among the entire population. Thus, the new candidate solutions are more likely to be close to the current best solution. Finally, they use a more robust calculation to determine and compare the quality of alternative solutions. We empirically assess the performance of our proposed method on two sets of problems: numerical benchmark functions and image registration applications. The results demonstrate that the proposed

* Corresponding author: mkaya@aksaray.edu.tr
[1] Aksaray University,
ORCID: https://orcid.org/0000-0003-3368-0796

method can produce higher quality solutions with faster convergence than the original ABC higher-qualityt state-of-the-art ABC-based algorithm. Anuara et. al. [8], studied on artificial bee colony with the rate of change technique that models the behavior of scout bee to improve the performance of the standard ABC in terms of exploration is introduced. The technique is called artificial bee colony rate of change (ABC-ROC) because the scout bee process depends on the rate of change on the performance graph, replace the parameter limit. The performance of ABC-ROC is analysed on a set of benchmark problems and also on the effect of the parameter colony size. Furthermore, the performance of ABC-ROC is compared with the state of the art algorithms. Brajevic and Tuba [9] introduced an upgraded artificial bee colony (UABC) algorithm for constrained optimization problems. Our UABC algorithm enhances fine-tuning characteristics of the modification rate parameter and employs modified scout bee phase of the Diwold et. al. [10] studied in detail the influence of ABC's parameters on its optimization behavior. It is also investigated whether the use of OBs is always advantageous. Moreover, we propose two new variants of ABC which use new methods for the position update of the artificial bees. Extensive empirical tests were performed to compare the new variants with the standard ABC and several other metaheuristics on a set of benchmark functions. Our findings show that the ideal parameter values depend on the hardness of the optimization goal and that the standard values suggested in the literature should be applied with care. Moreover, it is shown that in some situations it is advantageous to use OBs but in others it is not. In addition, a potential problem of the ABC is identified, namely that it performs worse on many functions when the optimum is not located at the center of the search space. Finally, it is shown that the new ABC variants improve the algorithm's performance and achieve very good performance in comparison to other metaheuristics underthe standard as well as hard optimization goals. Erik Cuevas et. al. [11], explored the use of the Artificial Bee Colony (ABC) algorithm to compute threshold selection for image segmentation. ABC is an evolutionary algorithm inspired by the intelligent behavior of honey bees which has been successfully employed to solve complex optimization problems. In this approach, an image 1-D histogram is approximated through a Gaussian mixture model whose parameters are calculated by the ABC algorithm. In the model, each Gaussian function represents a pixel class and therefore a threshold point. Unlike the Expectation-Maximization (EM) algorithm, the ABC method shows fast convergence and low sensitivity to initial conditions. Remarkably, it also improves complex time-consuming computations commonly required by gradient-based methods. Experimental results over multiple images with different range of complexity validate the efficiency of the proposed technique with regard to segmentation accuracy, speed, and robustness. The paper also includes an experimental comparison to the EM and to one gradient-based method which ultimately demonstrates a better performance from the proposed algorithm. Tsai et al. [12], proposed a model based on ABC, by employing Newtonian law of universal gravitation in the onlooker bee phase of ABC. Alatas [13], proposed an ABC model that uses chaotic maps for parameter adaptation so as to prevent the ABC to get stuck local minimums. The Rosenbrock's rotational direction method which was designed to cope with specific features of "Rosenbrock's banana function" was applied to ABC in order to increase exploitation and local search abilities of the basic AB Kang et al. [14]. Karaboga and Akay [15], adapted the basic ABC for constrained optimization problems by using Deb's Rules and evaluated the performance of the adapted model on the 13 constrained optimizations in the literature. Kıran et al. [16] added directional information to ABC algorithms, instead of updating more design parameters than one. The performance of proposed approach was examined on well known 9 numerical benchmark functions and obtained results are compared with basic ABC Kırran and Gunduz. Omkar et al. [17] presented vector evaluated ABC for multi objective design optimization of laminated composite components and compared the performance of vector evaluated ABC with other swarm-based methods.

In the first step of the study, 16 %, 32 %, 47 %, 63 %, 79 %, and 100 % feeding periods applied to the drones, and effect on ejaculation ability, and semen quality of drones was investigated. While investigation these feeding period effects “0-1”, bonded, and unbounded knapsack optimization problems used.

After the most effective feeding period was determined, this period applied to the travelling salesman problem (TSP) to determine the minimum way between two cities, and the liquid storage tank problem to determine the minimum connectors areas.

2. ARTIFICIALCAL HONEY BEE COLONY

The Artificial Bee Colony (ABC) algorithm is a relatively new member of swarm intelligence and tries to model the natural behavior of real honey bees in food foraging. Honeybees use several mechanisms to optimally locate food sources and search for new ones. This makes them good candidates for developing new intelligent search algorithms.

2.1. Coding

The most significant feature that distinguishes the proposed (ABC) from other operators is the use of codes to represent the design variables. The first step in the application of the (ABC) to a problem is the determination of the most appropriate coding type. The 0-1 knapsack problem binary coding was used, but for other problems permutation coding was used.

2.2. Formation of Initial Swarm

While the initial swarm is being formed, its members must not be selected the same, since the members must be chosen randomly. The suitable selection of swarm size significantly affects the performance of the ABC algorithm.

2.3. Evaluation

The (ABC) basically finds the maximum of an unconstrained objective function. To solve a constrained objective minimization function, two transformations need to be made. The first transforms the original objective constrained function into an unconstrained objective function, using the concept of the penalty function. In the second transformation, the unconstrained objective function is transformed to the fitness function.

2.4. Selection

A sequential selection method was used. In this method, members are set in order by a linearly decreasing function. The members with the lowest fitness value are removed from the swarm in a defined ratio and members with the highest fitness values replace those removed in the same ratio.

3. APPLICATIONS

In the first step of the study, 16 %, 32 %, 47 %, 63 %, 79 %, and 100 % feeding periods applied to the “0-1”, bonded, and unbounded knapsack problems. For 0-1 knapsack problems binary coding was used, but for bounded, and unbounded problems permutation coding was used.
In the second step of the study, the most successful feeding period was applied to the traveling salesman problem.

3.1. The “0-1” Knapsack Problem

The “0-1” knapsack problem has two options. These options are 0, and 1. If good’s code become 0, this good will not become in the knapsack, else if good’s code become 1, this good will put in the knapsack. In this problem, 16 %, 32 %, 47 %, 63 %, 79 %, and 100 % feeding periods applied to the “0-1” knapsack problem.

The “0-1” knapsack problem removes the restriction of having only one of each good, but restricts the number x_i of copies of each kind of good to an integer value c_i. Mathematically the “0-1” knapsack problem can be formulated using Eq. 1.

Table 1 Items will be put in the knapsack “0-1” knapsack problem

Code	Item	Weight (kN)	Value ($)	Number
1	Map	2.5	5.225	1
2	Compass	3	7.282	1
3	water	15	1.716	1
4	Sandwich	5	1.716	1
5	Glucose	7	5.225	1
6	Tin	3	3.971	1
7	Banana	6.5	2.475	1
8	Apple	4.5	1.969	1
9	Cheese	4	7.469	1
10	Beer	4.5	3.223	1
11	Suntan cream	6.0	6.369	1
13	T-shirt	3.5	12.716	1
14	Trousers	4.0	20.218	1
15	Umbrella	2.5	7.282	1
16	Waterproof trousers	4.5	25.223	1
17	Waterproof clothes	5.5	37.719	1
18	Note-case	3	1.782	1
19	Sunglasses	2	5.225	1
20	Towel	5	3.971	1
21	Socks	2	1.716	1
22	Book	3.5	7.282	1

In this problem, a tourist wants to take a trip. He has a knapsack for carrying his goods, but knows that he can carry maximum 3 kg weight. He creates a list of what he wants to take for the trip, but the total weight of all goods will not become heavier than 3 kg. Then he add columns to his need list detailing their weights and a numerical value representing how important each good is for the trip (Table 1).

In this problem the total of the weight of the tourist’s knapsack will not exceed 3 kg and their total value should be maximized. Mathematically the “0-1” knapsack problem can be formulated using Eq. 1:

W(x)=Subject to

$$\sum_{i=1}^{n} w_i . x_i \leq W, \qquad x_i \in \{0,1\} \tag{1}$$

$$\phi(s) = W(x)(1 + KC) \tag{2}$$

K: a coefficient selected for the problem taken as 5 in this study.

In the first transformation, the constrained objective function $\phi(s)$ was transformed into an unconstrained objective function $\phi(x)$ as shown in Eq. 3.

$$\phi(x) = \sum \phi(s) / \phi(s)_{\max} \tag{3}$$

In the second transformation, the unconstrained objective function was converted to an F(s) fitness function in Eq. 4.

$$F(s) = \phi(x)_{\max} - \phi(x) \tag{4}$$

Using the “0-1” knapsack problem, the highest nectar quality was 0.756 for 79 %, the feeding period, and the average nectar quality was 0.69 for this period. The highest nectar quality was 8.62 % higher than the average nectar quality for this problem. In this analysis, 79 % of this feeding period provided 100 drones, and 16 % feeding period provided only 5 drones in this swarm (Table 2). The good's maximum price for 0.756 nectar quality was 1918.510 $, and the goods’ average price was 1818.430 $. The goods maximumm price is 5.50 % higher than their average price.

Table 2 The nectar quality obtained from different feeding periods with pollen for "0-1" knapsack proplem

	16 %	32 %	47 %	63 %	79 %	100 %
Run 1	0.572	0.745	0.723	0.756	0.615	0.745
Run 2	0.658	0.615	0.572	0.766	0.702	0.820
Run 3	0.529	0.658	0.756	0.626	0.680	0.723
Run 4	0.442	0.680	0.637	0.680	0.637	0.637
Run 5	0.723	0.734	0.658	0.745	0.723	0.766
Run 6	0.626	0.777	0.691	0.615	0.550	0.507
Run 7	0.680	0.723	0.529	0.518	0.940*	0.572
Run 8	0.399	0.637	0.734	0.777	0.615	0.626
Run 9	0.572	0.658	0.766	0.745	0.756	0.594
Run 10	0.486	0.756	61.56	0.734	0.745	0.766
Average	0.569	0.698	0.676	0.696	0.693	0.676
Best	0.723	0.734	0.658	0.745	0.940*	0.766
Price ($)	1897.1	1785.3	1867.9	1918.51*	1787.6	1695,3

3.2. The Bounded Knapsack Problem (BKP)

In this problem, 16 %, 32 %, 47 %, 63 %, 79 %, and 100 % feeding periods applied to the bounded knapsack problem.

The BKP places have an upper bound on the number of copies of each kind of goods. Mathematically, the bounded knapsack problem can be formulated using Eq. 9. In this problem, a store boss wants to carry his computer warehouse to another one. He has twenty types, and 12.815 parts (monitors, keyboards, mouse, sound cards, graphics card, TV cards, modem etc.) in his old warehouse. His computer parts total volume 35.72 m^3, but his new warehouse capacity is only 20 m^3. As a result, he can carry only 20 m^3 expensive parts to his new warehouse (Table 3).

Table 3 Items in the computer warehouse; bounded knapsack problem

	Name	Number	Volume (cm^3)	Amount ($)	Total volume (m^3)	Total Amount ($)
1	Monitor	32000	40000	225	7296000	51300
2	Keyboard	960	1200	12	288000	3600
3	Mouse	144	180	10	69120	4800
4	Sound cards	600	750	30	288000	14400
5	Graphic card	600	750	85	324000	45900
6	Modem	600	750	15	432000	10800
7	Speaker	960	1200	20	864000	18000
8	Camera	518	648	30	248640	14400
9	Hard disk	480	600	67	201600	28140
10	DVD drives	1200	1500	38	504000	15960
11	Floppy disk	480	600	20	172800	7200
12	Case	36000	45000	30	17280000	14400
13	Power	2592	3240	10	1555200	6000
14	Motherboard	2400	3000	120	1440000	72000
15	Memory	58	72	15	174000	45000
16	Scaner	8000	10000	40	2640000	13200
17	Flash drive	16	20	20	57600	72000
18	Adaptor	480	600	15	288000	9000
19	Fan	256	320	3	107520	1260
20	Ethernet	136	170	10	57120	4200

$W(x)=\sum_{i=1}^{n} w_i . x_i$ Subject to

$$\sum_{i=1}^{n} w_i . x_i \leq W, \quad x_i \in \{0, . c_i\} \tag{5}$$

In the first transformation, the constrained objective function $\phi(s)$ was transformed into an unconstrained objective function $\phi(x)$ as shown in Eq. 6.

$$\phi(x) = \sum \phi(s) / \phi(s)_{\max} \tag{6}$$

Table 4 The nectar quality obtained from different feeding period with pollen for bounded knapsack problem

	0.025 %	0.050 %	0.1 %	0.25 %	0.5 %	1 %
Run 1	0.462	0.475	0.759	0.745*	0.598	0.773
Run 2	0.551	0.553	0.874	0.672	0.579	0.518
Run 3	0.427	0.444	0.713	0.735	0.722	0.739
Run 4	0.649	0.639	0.885	0.682	0.484	0.612
Run 5	0.293	0.327	0.540	0.577	0.579	0.612
Run 6	0.293	0.327	0.540	0.577	0.579	0.612
Run 7	0.507	0.514	0.816	0.745	0.807	0.391
Run 8	0.703	0.444	0.724	0.567	0.636	0.663
Run 9	0.792	0.530	0.839	0.514	0.465	0.569
Run 10	0.605	0.600	0.598	0.593	0.437	0.450
Average	0.528	0.485	0.729	0.653	0.589	0.594
Best	0.427	0.507	0.551	0.745*	0.676	0.774
Price ($)	35712.35	32927.45	34389.43	35824.17 *	35627.17	33762.45

In the second transformation. the unconstrained objective function $\phi(x)$ was converted to an F(s) fitness function in Eq. 7.

$$F(s) = \phi(x)_{\max} - \phi(x) \tag{7}$$

Using the bounded knapsack problem. the highest nectar quality was 0.952 for 63 %. feeding period, and the average nectar quality was 0.501 for 16 % feeding period. The highest nectar quality was 16.36 % higher than the average nectar quality for this problem. In this analysis, 63 % of the feeding period provided 50 drones, and 16 % feeding period provided only 5 drones in this swarm (Table 4). For the same knapsack problem, the maximum price of 0,952 nectar quality was 34918.510 $. The computer part's maximum price was 7.71 % bigger than average price.

3.3. The Unbounded Knapsack Problem (UKP)

In this problem, 16 %, 32 %, 47 %, 63 %, 79 %, and 100 % feeding periods applied to the unbounded knapsack problem. The UKP places no upper bound on the number of copies of each kind of goods.

The unbounded knapsack problem (UKP) places no upper bound on the number of copies of each kind of good and can be formulated as above except for that, the only restriction on x_i is that it is a nonnegative integer. Mathematically the unbounded knapsack problem can be formulated using Eq. 8:

$$W(x)=\sum_{i=1}^{n} w_i . x_i$$

$$\text{Subject to } \sum_{i=1}^{n} w_i . x_i \leq W, \quad 0 \leq x_i \leq \infty \tag{8}$$

In this problem; a greengrocer wants to buy fruits with his all money. He has 100,000 USD dollars, and can buy fruits with this money. For the greengrocer fruits kind, weight, or volume does not important, because he wants to buy profitable fruits, with his all money, and want to earn maximum money from this trade (Table 5).

$$\phi(s) = W(x)(1 + KC) \tag{9}$$

K: a coefficient selected for the problem taken as 5 in this study.

In the first transformation, the constrained objective function $\phi(s)$ was transformed into an unconstrained objective function $\phi(x)$ as shown in Eq. 10.

$$\phi(x) = \sum \phi(s) / \phi(s)_{\max} \tag{10}$$

In the second transformation, the unconstrained objective function $\phi(x)$ was converted to an F(s) fitness function in Eq. 11.

$$F(s) = \phi(x)_{\max} - \phi(x) \tag{11}$$

Using the unbounded knapsack problem, the highest nectar quality was 0. 704 for a 63 % feeding period, and the average nectar quality was 0.61 for a 16 % feeding period.

Table 5 Kind of fruits in the greengrocer store; unbounded knapsack problem

	Name	Weight (kg)	Volume (m^3)	Amount ($)	Total Amount ($)
1	Apple	0weight<∞	300	1.5	1.5≤ (TA) <100.000$
2	Pear	0≤weight<∞	300	2.2	2.2≤ (TA) <100.000$
3	Orange	0≤weight<∞	300	1.6	1.6≤ (TA) <100.000$
4	Banana	0≤weight<∞	300	3.5	3.5≤ (TA) <100.000$
5	Kangerine	0≤weight<∞	300	1.6	1.6≤ (TA) <100.000$
6	Kuince	0≤weight<∞	300	1.5	1.5≤ (TA) <100.000$
7	Kherry	0≤weight<∞	300	4.2	4.2≤ (TA) <100.000$
8	Alligator pear	0≤weight<∞	300	1.3	1.3≤ (TA) <100.000$
9	Mango	0≤weight<∞	300	3.2	3.2≤ (TA) <100.000$
10	Krapes	0≤weight<∞	300	1.5	1.5≤ (TA) <100.000$
11	Blackberry	0≤weight<∞	300	2.3	2.3≤ (TA) <100.000$
12	Kiwi	0≤weight<∞	300	2.4	2.4≤ (TA) <100.000$
13	Grapefruit	0≤weight<∞	300	15.0	15.0≤ (TA) <100.000$
14	Pepino	0≤weight<∞	300	4.2	4.2≤ (TA) <100.000$
15	Peach	0≤weight<∞	300	2.5	2.5≤ (TA) <100.000$
16	Apricot	0≤weight<∞	300	2.1	2.1≤ (TA) <100.000$
17	Plum	0≤weight<∞	300	1.5	1.5≤ (TA) <100.000$
18	Pomegranate	0≤weight<∞	300	1.5	1.5≤ (TA) <100.000$
19	Raspberry	0≤weight<∞	300	3.0	3.0≤ (TA) <100.000$
20	Pineapple	0≤weight<∞	300	4.0	4.0≤ (TA) <100.000$

Fruit weight between 0 kg and ∞
Total amount (TA) between every unit amount and 100.000$

The highest nectar quality was 11.71 % higher than the lowest nectar quality for this problem. In this analysis, 63 %, of the feeding period provided only 50 drones, and 16 % feeding period provided only 5 drones in this swarm (Table 6). For the same problem, the fruit maximum price of 0.704 nectar quality was 98.700 $, and the average was 94.610 $. Fruits maximum price was 4.32 % bigger than their average price.

Table 6 The nectar quality obtained from different feeding period with pollen for unbounded knapsack problem

	0.025 %.	0.050 %.	0.1 %	0.25 %	0.5 %	1 %
Run 1	0.508	0.593	0.645	0.699	0.627	0.618
Run 2	0.606	0.691	0.742	0.604	0.607	0.414
Run 3	0.469	0.555	0.606	0.661	0.758	0.591
Run 4	0.713	0.798	0.752	0.613	0.508	0.489
Run 5	0.322	0.408	0.459	0.519	0.607	0.499
Run 6	0.323	0.408	0.459	0.519	0.607	0.467
Run 7	0.557	0.642	0.697	0.704*	0.847	0.312
Run 8	0.773	0.555	0.616	0.510	0.667	0.530
Run 9	0.871	0.662	0.713	0.462	0.488	0.455
Run 10	0.665	0.751	0.508	0.443	0.458	0.366
Average	0.580	0.606	0.619	0.598	0.617	0.474
Best	0.871	0.662	0.713	0.704*	0.488	0.455
Price($)	96143.54	89973.73	94385.19	97612.32*	93786.61	89827.72

3.4. Travelling Salesman Problem (TSP)

The travelling salesman problem (TSP) is an integer program that gives the shortest route when the distance between the pairs of points is known and each point is visited only once. It is one of the few integer algorithm-based programs described in the literature. The TSP is also a way of collecting and distributing the necessary parts of the objects in the modeling system, thus maximizing the profit while minimizing the cost using the shortest way. In this problem, a salesman wants to sell his goods in n number of cities and he wants to visit all the cities using the shortest route. Hence, the salesman has a choice of n cities of $\frac{1}{2}(n-1)!$ different routes. The salesman must travel to 32 cities, visiting each city only once, and begin by travelling from city A to city B. This means there are $8.05x10^{32}$ different routes for 32 cities (Fig. 1.).

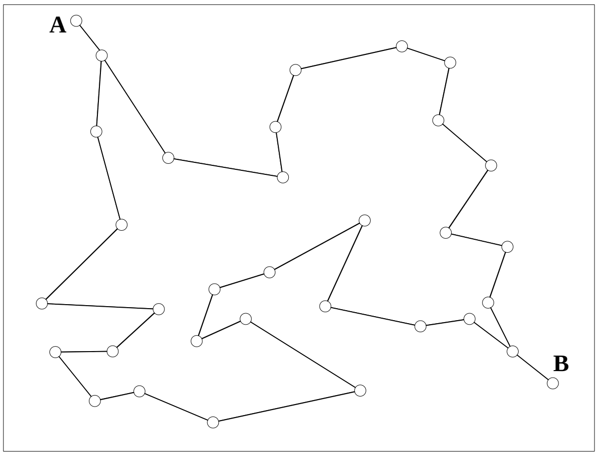

Figure 1 Optimum solution for TSP by using GA

For the TSP the objective function $W(x)$ is given in Eq. 12

$$\min W(x) = \sum_{k=1}^{nk} L_k \tag{12}$$

Where L_k is the distance between the two cities.

The constrained objective function was transformed into an unconstrained objective function $\phi(s)$ as shown in Eq. 16,

i: the number of the first city

j: the number of the second city

x_i and y_i: the coordinates of the first city

x_j and y_j: the coordinates of the second city

When the salesman goes to the target city from the starting city then;

If xi > xj

$$g(x) = \sqrt{\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right)} \tag{13}$$

Else

$$g(x) = \left[\sqrt{\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right)}\right] x100 \tag{14}$$

$$G = \sum g(x) \tag{15}$$

$$\phi(s) = W(x)(1 + KG) \tag{16}$$

K: a coefficient selected for the problem, taken as 10 in this study.

$g(x)$: a negligence coefficient thmostas calculated as follows;

In the first transformation, the constrained objective function was transformed into an unconstrained objective function $\phi(x)$ as given in Eq. 19:

$$\phi(x) = \sum \phi(s) / \phi_{\max} \tag{17}$$

In the second transformation in Eq. 20 the unconstrained objective function $\phi(x)$ was converted to the fitness function F(s)

$$F(s) = \phi_{\max} - \phi(x) \tag{18}$$

For the travelling salesman problem 79 %, feeding period was used to obtain the shortest way. Because this feeding period had used previous knapsack problems and gave the highest nectar quality. For the travelling salesman problem, the shortest way was found 2718 km, and this way 5.78 % shorter than average way length (Table 7).

Table 7 The result of runs for traveling salesman problem (km)

Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
2944	2907	2718	2865	2870	2886	2827	2923	2899	2918
Average way (km): 2885.72 km									
Minimum way (km): 2718 km									

3.5. Tsunami Force Effect to The Liquid Storage Tank Problem

The tsunami load effects should be considered in the design of vertical evacuation structures, namely; hydrostatic forces, buoyant forces, hydrodynamic forces, impulsive forces, debris impact forces, debris damming forces, uplift forces, and additional gravity loads from retained water on elevated floors Kang et al. (2011). In this problem minimum liquid storage tank twelve connectors area which connecting storage tank floor to the ground under different height and velocity tsunami waves.

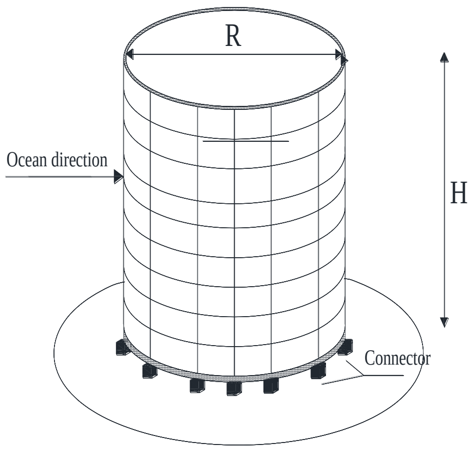

Figure 2 Liquid Storage Tank view

In this problem, it was determined the liquid storage tank's twelve connectors area under debris impact effect, and hydrostatic forces. In the problem, tsunami wave length is 7.30 m, tsunami wave speed is 30-50 km/h, tank diameter is 20 m, and tank height is 40 m (Fig.2).

3.5.1. Hydrostatic force

Hydrostatic forces occur when standing or slowly moving water encounters a structure or structural component. This force always acts perpendicular to the surface of the component of interest. It is caused by an imbalance of pressure due to a differential water depth on opposite sides of a structure or component. Hydrostatic forces may not be relevant to a structure with a finite (i.e., relatively short) breadth, around which the water can quickly flow and fill in on all sides. Hydrostatic forces are usually important for long structures such as sea walls and dikes, or for evaluation of an individual wall panel where the water level on one side differs substantially from the water level on the other side FEMA [18].

The horizontal hydrostatic force on a wall panel can be computed using Eq. 19.

$$F_h = P_c . A_w = \frac{1}{2} \rho_s . g . b . h_{max}^2 \tag{19}$$

$$M_h = F_h . \frac{d}{3} \tag{20}$$

Where P_c is the hydrostatic pressure, A_w is the wetted area of the panel, ρ_s is the fluid density including sediment, g is the gravitational acceleration, b is the breadth (width) of the wall, and h_{max} is the maximum water height above the base of the wall at the structure location.

3.5.2. Impact force

The impact force from waterborne debris (e.g., floating drift wood, boats, shipping containers, automobiles, buildings) can be a dominant cause of building damage. Unfortunately, it is difficult to estimate this force accurately. The debris impact force can be estimated using Eq. 21.

$$F_i = C_m . U_{max} \sqrt{k . m} \tag{21}$$

$$M_i = F_i . d_i \tag{22}$$

Where C_m is the added mass coefficient, U_{max} is the maximum flow velocity carrying the debris at the site, and m and k are the mass and the effective stiffness of the debris, respectively. It is recommended that the added mass coefficient be taken as C_m = 2.0. Unlike other forces, impact forces are assumed to act locally on a single member of the structure at the elevation of the water surface FEMA (2018).

$$\sum M = M_h + M_i \tag{23}$$

$$M_{net} = \sum M - \pi . r^2 . h . \gamma_{oil} . \tag{24}$$

In this problem permutation coding type was used. The connectors diameters and their codes were given in Table 8. Individual length is equal to the number of variable groups (connector number) in this problem. Randomly determined an individual was given in Table 9.

Table 8 Connector reinforcement diameters and codes

Diameter	Ø24	Ø28	Ø32	Ø36	Ø40	Ø44	Ø48	Ø52
Code	1	2	3	4	5	6	7	8

Where, A_n is the reinforcement area, ρ is the specific gravity of reinforcement, l_n is the reinforcement length. The constrained objective function was transformed into an unconstrained objective function $\phi(x)$ as expressed in Eq. 31.

Table 9 Randomly selected an individual sites codes, and connectors diameters

Number	1	2	3	4	5	6	7	8
Site code	7	5	3	8	9 6	2	7	1
Diameters	Ø48	Ø40	Ø32	Ø52	Ø44	Ø28	Ø48	Ø24

$$minW(x) = \sum_{i=1}^{n} l_n . A_n . \sigma_n \tag{25}$$

$$g_i(x) = \frac{M_{net}}{\sum_{i=1}^{n} l_n . A_n . \sigma_n} - 1 \tag{26}$$

$$\text{If} \quad g_i(x) \geq 0 \qquad c_i = g_i(x) \tag{27}$$

$$\text{If} \quad g_i(x) < 0 \qquad c_i = g_i(x).100 \tag{28}$$

$$C = \sum c_i \tag{29}$$

$$\phi(s) = W(x)(1 + KC) \tag{30}$$

K: a coefficient selected for the problem taken to be 10 in this study.

c_i: negligence coefficient and calculated as follows;

In the first transformation, the constrained objective function $\phi(s)$ was transformed to an unconstrained objective function $\phi(x)$ as expressed in Eq. 31.

$$\phi(x) = \sum \phi(s) / \phi(s)_{\max} \tag{31}$$

Where Fe_i is the required force met by reinforcements in the x or y direction in the ith zone, f_{yd} is the yield strength of reinforcement, and ns is the number of reinforcements in the ith zone.

In the 2nd transformation, the unconstrained objective function $\phi(x)$ was converted to a fitness function F(s). This transformation was achieved using the maximum of the i_{th} element of the unconstrained objective function. The fitness values of the members were calculated according to Eq. 32 as follows;

$$F(s) = \phi(x)_{\max} - \phi(x) \tag{32}$$

For the liquid storage tank problem 79 %, feeding period was used to obtain the lowest total connectors areas. Because this feeding period had used previous knapsack problems and gave the highest nectar quality.

For the liquid storage tank problem, the minimum 8 connectors area in the tank problem was 21991.8 mm^2, and connectors average area was 23508.81mm^2. Minimum area was 6.45 % less than average area (Table 10).

Table 10 Connector areas determined for liquid storage tank (mm^2)

	1	2	3	4	5	6	7	8	Total area (mm^2)
Run 1	1019.03	805.11	1018.63	805.11	1257.27	1521.03	1809.91	3216.63	11452.72
Run 2	1258.47	1258.47	617.91	1258.47	1019.83	1258.47	2464.23	2828.47	11452.72
Run 3	1521.65	1019.25	805.73	805.73	1521.65	1810.53	2124.53	3631.73	11452.72
Run 4	1020.03	1258.67	1020.03	1258.67	1020.03	1811.31	2125.31	2828.67	11452.72
Run 5	1257.95	1019.31	1019.31	805.79	1521.71	1521.71	1810.59	3217.31	11452.72
Run 6	1698.76	1421.19	794.44	1196.36	1497.57	1537.87	2640.76	3394.36	11452.72
Run 7	1259.34	807.18	807.18	1259.34	1523.1	1811.98	2125.98	3633.18	11452.72
Run 8	1021.48	1021.48	619.56	807.96	1260.12	1523.88	1812.76	2830.12	11452.72
Run 9	1523.07	807.15	807.15	1259.31	1259.31	1811.95	1811.95	3218.67	11452.72
Run 10	1258.17	1258.17	806.01	1019.53	1019.53	1521.93	2124.81	3632.01	11452.72

Average liquid storage tank connectors area

4. CONCLUSIONS

In the study, it was investigated to the effect of feeding with pollen in different periods on the ejaculation ability and semen quality of drones. In the first stage of the study “0-1”, bounded, and unbounded knapsack problems were used in this comparison. In the analysis, the maximum nectar quality was obtained from Using 79 % fertilization fault ratio for “0-1” knapsack problem, for this fertilization fault ratio drone number is 100, and this drone numbers enough for a 20.000 people swarm. The maximum nectar quality was obtained from 63 % fertilization fault ratio for bounded, and unbounded knapsack problems. For this fertilization fault ratio is 50 drones, and this drone numbers too less for a necessary drone number for a 20.000 people swarm. Using 79 % unfertilizition ratio produced successful results in the determination of the highest prices of the goods in the tourist’s knapsack, highest prices of the computer parts which the store boss carried them from old warehouse to a new warehouse, and a greengrocer wanted to buy fruits with his all money, and he won the highest money from this trade. In the second step of the study, while it was determined the travelling salesman problem, and minimum liquid storage tank connectors diameters 79 % fertilization fault ratio were used. Using 79 % fertilization ratio provided higher nectar quality than other fertilization ratios. While determining the shortest way using the same unfertilazition ratio, provided the most successful result in traveling salesman problem.

Using a 79 % fertilization ratio provided higher nectar quality than other fertilization ratios. While determining the optimum 8 connectors area for liquid storage tanks using the same unfertilazition ratio, provided the most successful result in liquid storage tank connector areas between the tank and ground. It was seen from the analysis, if the fertilization ratio increases, queen and worker bee numbers decrease, but drones number increases. Else if the fertilization fault ratio decreases queen and worker bee numbers increase, but drone number decreases. In this study, the optimum drone number was determined as 100 for this swarm.

Acknowledgments
I would like to express our great appreciation to the team during the planning and development of this research.

Funding
This study did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Authors' Contribution
The author contributed equally to the study.

The Declaration of Conflict of Interest/ Common Interest
No conflict of interest or common interest has been declared by the authors.

The Declaration of Ethics Committee Approval
This study does not require ethics committee permission or any special permission.

The Declaration of Research and Publication Ethics
The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, “The Bees Algorithm A Novel Tool For Complex Optimization Problems”, Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems (EPROMS 2006). Elsevier Science Ltd, Cardiff, UK, 2006.

[2] D. Pham, A. Ghanbarzadeh, “Multiobjective Optimization Using The Bees Algorithm”, 3rd International Virtual Conference on Intelligent Production Machines and Systems (EPROMS 2007), Whittles, Dunbeath, Scotland, pp. 111-116, 2007.

[3] B. Yuce, D. Pham D, M. Packianather, E. Mastrocinque, “An Enhancement To The Bees Algorithm With Slope Angle Computation And Hill Climbing Algorithm And Its Applications In Scheduling And Continuous Type Optimization Problem”, Production & Manufacturing Research, vol. 3, pp. 3-19, 2015.

[4] S. Abdullah, M. Alzaqebah, “A hybrid Self Adaptive Bees Algorithm For Examination Timetabling Problems”, Applied Soft Computing, vol. 13, pp. 3608-3620, 2013.

[5] C. Lara, JJ. Flores, F. Calderón, “Solving A School Timetabling Problem Using A Bee Algorithm”, Advances in Artificial Intelligence. Springer, Berlin, Heidelberg, 2008.

[6] Z. Dongli, G. Xinping, T. Yinggan, T. Yong, “Modified Artificial Bee Colony Algorithms for Numerical Optimization”, in Proc. 3rd International Workshop on Intelligent Systems and Applications, 2011, pp. 1–4, 2011.

[7] A. Banharnsakun, T. Achalakul and B. Sirinaovakul B, “The best-so-far selection in Artificial Bee Colony algorithm”, Applied Soft Computing, Vol. 11, no. 2, pp. 2888-2901, 2011.

[8] S. Anusara, A. Selamatab R. Sallehuddin, “A modified scout bee for artificial bee colony algorithm and its performance on optimization problems”, Journal of King Saud University - Computer and

Information Sciences, vol 28, no. 4, pp. Pages 395-406, 2016.

[9] I. Brajevic, M. Tuba, “An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems”, manufacturing, vol. 24, pp. 729–740, 2013.

[10] K. Diwold, A. Andrej, A. Scheidler, “Middendorf M, Performance evaluation of artificial bee colony optimization and new selection schemes”, Memetic Computing, vol. 3, pp. 149-162, 2011.

[11] E. Cuevas, F. Sención, D. Zaldivar, M. Pérez-Cisneros and H. Sossa, “A multi-threshold segmentation approach based on Artificial Bee Colony optimization”, Applied Intelligence, vol. 37, pp. 321–336, 2012.

[12] PW. Tsai, JS. Pan, BY. Liao, SC. Chu (2009), “Enhanced Artificial Bee Colony Optimization”, International Journal of Innovative Computing. Information and Control, vol. 5, pp. 5081–5092, 2009.

[13] B. Alatas, “Chaotic Bee Colony Algorithms for Global Numerical Optimization”, Expert System Application, vol. 37 (2010) pp. 5682–5687, 2010.

[14] F. Kang, J. Lie, Z. Ma, “Rosenbrock Artificial Bee Colony Algorithm for Accurate Global Optimization of Numerical Functions”, Information Sciences vol. 181, pp. 3508–3531, 2011.

[15] D. Karaboga, B. Akay, “A Modified Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Problems”, Applied. Soft Computing, vol. 11, pp. 3021–3031, 2011.

[16] MS. Kıran, M. Gündüz, “A Novel Artificial Bee Colony-Based Algorithm for Solving the Numerical Optimization Problems”, International Journal of Innovative Computing. Information and Control, vol. 9, pp. 6107–6121, 2012.

[17] SN. Omkar, J. Senthilnathproblems R. Khandelwal R, GN. Naik, S. Gopalakrishnan, “Artificial Bee Colony (ABC) for Multi-Objective Design Optimization of Composite Structures”, Applied. Soft Computing, vol. 11, pp. 489–499, 2011.

[18] FEMA P646, “Guidelines for Design of Structures for Vertical Evacuation from Tsunamis”, (2008).