www.dergipark.gov.tr
ISSN:2148-3736
El-Cezerî Fen ve Mühendislik Dergisi
Cilt: 7, No: 2, 2020 (503-512)

El-Cezerî Journal of Science and Engineering
Vol: 7, No: 2, 2020 (503-512)
DOI:10.31202/ecjse.660382

ECJSE

Research Paper / Makale

Solution And Performance Analysis Of Subset Sum Problem With A New Metaheuristic Approach

Mustafa Furkan KESKENLER[1]*, Eyüp Fahri KESKENLER[2]

[1]Ataturk University, Faculty of Engineering, Department of Computer Engineering, 25240 Erzurum/TURKEY
[2]Recep Tayyip Erdoğan University, Faculty of Engineering, Department of Electric And Electronic Engineering, 53100 Rize/TURKEY
mfkeskenler@gmail.com

Received/Geliş: 17.12.2019 **Accepted/Kabul:** 26.02.2020

Abstract: Subset sum problem was solved with two different metaheuristic approaches in the study. After these approaches, which are simulated annealing and genetic algorithms, a hybrid model of two methods was created and better results were obtained. The observed results were compared with other methods in the literature and the best time cost results were yielded owing to the hybrid algorithm developed in the study. The algorithms used gave successful results in terms of Cost values too. Performance analyses were measured on the Subset Sum Problem, defined as NP-Complete problem in computer science, with different functions used in these methods. Thus, the success of the sub-functions of the commonly used Simulated Annealing and Genetic Algorithm methods were compared and the findings were yield that could guide the researchers in other studies.

Keywords: Subset sum, simulated annealing, genetic algorithm, hybrid model, metaheuristics, optimization

Yeni Bir Metahuristik Yaklaşımla Alt Küme Toplamı Probleminin Çözümü Ve Performans Analizi

Öz: Araştırmada iki farklı metaheuristik yaklaşımla alt küme toplamı problemi çözümüne odaklanılmıştır. Benzetilmiş Tavlama ve Genetik Algoritma yaklaşımlardan sonra bu iki metottan oluşan hibrit bir model geliştirilmiş ve daha iyi sonuçlar elde edilmiştir. Gözlemlenen sonuçlar literatürdeki diğer yöntemlerle kıyaslanmış ve çalışmada geliştirilen hibrit algoritma ile en iyi zaman maliyetine sahip sonuçların elde edildiği bulunmuştur. Kullanılan algoritmalar Cost değerleri yönüyle de başarılı sonuçlar vermiştir. Bilgisayar bilimlerinde NP-Complate problem olarak tanımlanan Alt küme toplamı problemi kullanılarak bu yöntemlerde kullanılan farklı fonksiyonlarla performans analizleri yapılmıştır. Böylece yaygın olarak kullanılan Simulated Annealing and Genetic Algorithm yöntemlerine ait alt fonksiyonların başarısı kıyaslanmış ve araştırmacılar için diğer çalışmalarda yol gösterebilecek bulgular elde edilmiştir.

Anahtar Kelimeler: Alt küme toplamı, benzetilmiş tavlama, genetic algoritma, hibrit model, metasezgisel, optimizasyon

1. Introduction

The Subset Sum Problem is one of the well-known problems in algorithm theory and a good example of NP-Complete problems [1,2]. The definition of the problem is to find that the sum of any subset of the given set of minus and plus values is 0 or target [3,4]. It is quite simple to check the solution of the problem, but it is not easy to find a subset of the set of numbers that gives the above definition. A sample solution for SSP is below (Selected target is 0).

How to cite this article
Keskenler, M.F., Keskenler, E.F., "*Solution And Performance Analysis Of Subset Sum Problem With A New Metaheuristic Approach*" El-Cezerî Journal of Science and Engineering, 2020, 7 (2); 503-512.

Bu makaleye atıf yapmak için
Keskenler, M.F., Keskenler, E.F., *"Yeni Bir Metahuristik Yaklaşımla Alt Küme Toplamı Probleminin Çözümü Ve Performans Analizi"* El-Cezerî Fen ve Mühendislik Dergisi 2020, 7 (2); 503-512.

SSP solving example:

The Numbers (2000 numbers In this study)									
…	114	1453	-313	74	1033	-57	-1071	-12	…
		1	1			1	1	1	

Cost
0

If the subsets sum problem is solved with the greedy approach given in Figure 1, $O(2^N)$ becomes the runtime cost [5]. This is a very long time to solve this problem. For this reason, various dynamic programming and metaheuristic methods are used to solve this problem. Simulated annealing, genetic algorithms, and the hybrid model have been used in this study.

```
SubSetVector << Read File();
    SubSetContainer = Random(0 or 1); //initial values
    For i = 0, N   //N is number of set value
         CostVector = SubSetVector[i] x SubSetContainer[i];
BestCost = Sum(CostVector);
While(BestCost != TargerValue)
     UpdateValues (SubSetContainer);
     CostVector = SubSetVector[i] x SubSetContainer[i];
     Cost = Sum(CostVector);
     İf (Cost < BestCost)
           BestCost = Cost;
```

Figure 1. Pseudo-code of Greedy Approach for Subset Sum Problem

2. Material And Method

One of the algorithms used in computer science is the heuristic algorithms. These algorithms, which have no certainty in their operation, either do not always work with the same performance or guarantee that they always produce results, but they are still handy algorithms for optimizing the problem.

Metaheuristics algorithms are a decision mechanism that works on these heuristic algorithms. So different heuristic methods can be used for a problem. Metaheuristic algorithms are used to decide which of these heuristic methods will be chosen and what values of the selected algorithm parameters will be. Metaheuristic methods are independent of the problem. The study includes results that were obtained for Subset Sum Problem (SSP) using simulated annealing, genetic algorithm and hybrid model approaches. Linear algorithms can often have high computational costs, while heuristic algorithms are always unable to find the best way for difficult problems [6]. In this study, it has been tried to give remedial solutions for metaheuristic algorithms. The parameter values in the study were determined by running algorithms repeatedly through trial.

Measurement attributes in methods:

- A computer used for performance measurements that is 2400 MHz core 2 duo – 4 thread CPU, 8 Gb RAM, SSD.

- 2000 element cluster with values between -35000 and +35000 is used in the Performance Measurements.
- The target value is set to 0.
- The measurements were tested on different functions used in metaheuristics.

2.1. Implementation simulated annealing for SSP

Simulated Annealing is a stochastic search method [7]. It has been created by the similarity with the physical annealing process of the solids. It is based on the principle that the solids are heated and then slowly cooled. In Simulated Annealing, neighbor movements that may cause an increase in the objective function are sometimes accepted, and it makes possible to get rid of the local best spots [8]. Acceptance of this neighboring motion, which may lead to an increase in the objective function, is determined randomly. The operator that gives the probability of accepting the motion that leads to a Δ rise in the objective function is called the acceptance function. This is determined by the Boltzmann distribution function "(1)". T is the temperature in the equation [9]. Δ is the difference between the neighbor solution and the current solution “(2)”.

$$\text{F(accept)} = e^{-(\Delta / T)} \tag{1}$$

$$\Delta = \text{neighbor solution - current solution} \tag{2}$$

In simulated annealing parameters, the initial temperature is 5000 ^{o}C, cooling rate (alpha) is 0.983, freezing point (epsilon) is 0.001 and the number of iterations per temperature is 1000. The solution is performed according to the algorithm steps given in Figure 2.

```
Algorithm Simulated-Annealing for SSP
Begin
        tmp = initialTemp;
        conteiner = initialRandumValue;
        while (tmp > epsilon) do
                while (loopCriterion = False) do
                        newConteiner = PERTURB(conteiner);
                        ΔC = COST(newConteiner)
                          - COST(conteiner);
                        if (ΔC < 0) then
                                conteiner = newConteiner;
                        else if (RANDOM(0,1) < e^-(ΔC/temp)) then
                                conteiner = newConteiner;
                tmp = tmp * alpha;
End.
```

Figure 2. Pseudo-code of Simulated-Annealing for SSP

2.2. Implementation genetic algorithm for SSP

Genetic Algorithm is a method that computer science learns from natural sciences and uses to solve its problems [10]. Genetic algorithms those the basic principles have been proposed by John Holland [11] are an optimization technique based on the science of genetics. It is an algorithm developed with inspiration from biological processes such as crossing, mutation, and natural selection in the field of genetics, modeling these processes mathematically and optimizing functions [12]. GA is starting to work with a large number of solutions compared to a randomly generated initial population [13]. Thanks to the hybrid algorithm developed in this study, this initial population is aimed to consist of better individuals. Then, GA attempts to optimize solutions by using genetic operators (such as selection, crossing, mutation) [14,15].

This algorithm uses the basic three operations used in genetics. These three basic functions are listed below.

- Crossover
- Mutation
- Selection

The first two processes above are two basic processes that play a role in a genetic change. The selection between genes that change with these two basic processes (crossover and mutation) is a method that is used in genetic algorithms and provides success [16].

Tournament selection or roulette wheel selection methods have been used for the selection process.

One Point Crossover function is also used for the crossover methods. The Bit-Flip Mutation and Swap Mutation functions are used for the mutation [17]. The solution has been realized by applying the steps in Figure 3.

Genetic algorithm parameter values have been chosen as the number of generations is 1000, population size is 300, the crossover rate is 0.95, and the mutation rate is 0.35.

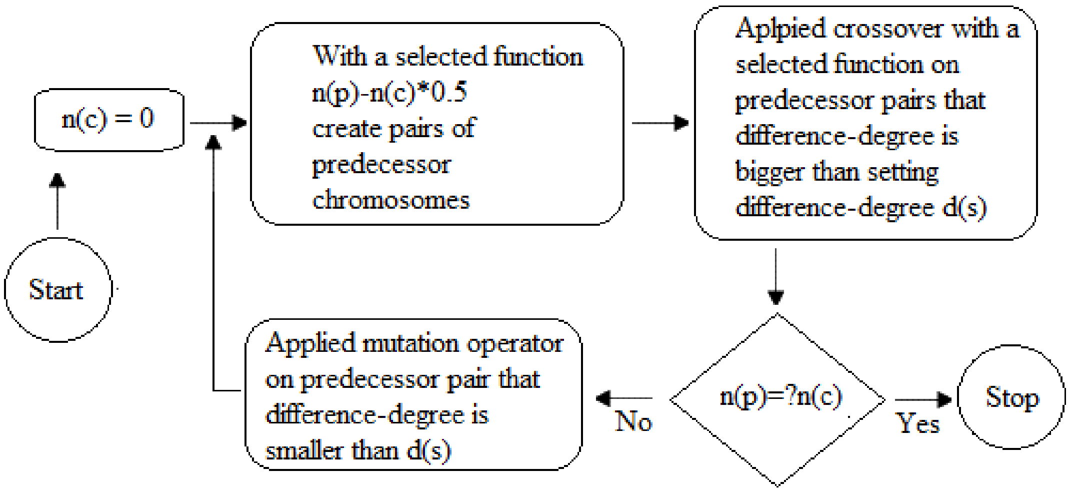

Figure 3. Genetic Algorithm Steps for SSP

2.3. Implementation hybrid model for SSP

The hybrid model is obtained by combining the Simulated Annealing and Genetic Algorithm. Simulated Annealing is used the create the initial population process for the Genetic Algorithm. The initial population values are produced at a certain level of improvement, without approaching the target value. Better results have been observed in the genetic algorithm by producing good children from better parents [18].

The parameter values for simulated annealing in the hybrid model have been selected as the initial temperature is 1000 ^{o}C, cooling rate (alpha) is 0.9, freezing point (epsilon) is 950 and the number of iterations per temperature is 15.
The parameter values for the Genetic Algorithm in the hybrid model have been chosen as the number of generations is 1000, the population size is 300, the crossover rate is 0.95, and the mutation rate is 0.35.

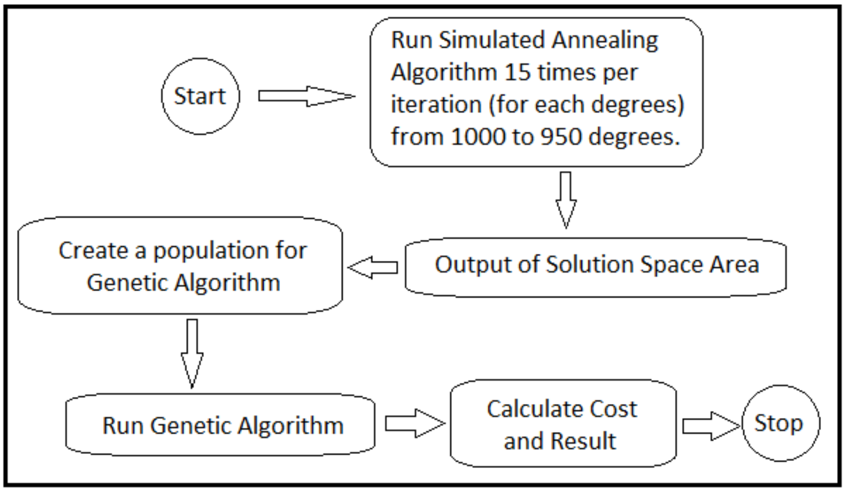

Figure 4. Hybrid algorithm model for problem

As can be seen in Figure 4, firstly the exploration process is started by running the Simulated Annealing Algorithm. But the algorithm is stopped before it goes too far. This, solution space close to the optimal solution is discovered with the very little time cost. Then the genetic algorithm starts processing by creating the population in the improved solution space obtained from the Simulated Annealing algorithm. The first population is created by this method, not randomly, and it is aimed to create good children from good parents. In this way, the genetic algorithm gives both better and faster results.

3. Results and Discussion

Results were obtained for The Subset Sum Problem using three metaheuristic methods in the study. Performance comparisons were made using different functions within each method.

As shown in Table 1, both The Bit-Flip Operator Function and the Scramble Operator Function have reached the target for the performance measurements for the simulated annealing and very good results have been obtained. Bit-Flip Operator Function is faster than Scramble Operator Function. However, it rarely happens that when they reach the target they deviate very low.

Table 1. Different function performance results for Simulated Annealing

	Used Bit-Flip Operator Function		**Used Scramble Operator Function**	
	Time (s)	**Cost**	**Time (s)**	**Cost**
Executive Time 1	33.6	0	58.1	0
Executive Time 2	109.4	0	166.2	0
Executive Time 3	32.2	-1	196.5	0
Executive Time 4	44.3	0	71.4	0
Executive Time 5	20.7	0	133.3	0
Executive Time 6	108.9	0	231.6	0
Executive Time 7	46.4	1	144.0	0
Executive Time 8	88.8	1	121.3	0
Executive Time 9	19.9	0	69.8	0
Executive Time 10	21.7	0	111.1	0
Average	**52.49**	**0.3**	**130.33**	**0**

Table 2. Different function performance results for Genetic Algorithm when Created Mating Pool Based On Roulette Wheel Selection

	Create Mating Pool Based On Roulette Wheel Selection			
	Apply Swap Mutation Operator		Apply Bit-Flip Mutation Operator	
	Apply One Point Crossover Operator		Apply One Point Crossover Operator	
	Time (s)	**Cost**	**Time (s)**	**Cost**
Executive Time 1	115.3	1	108.3	8
Executive Time 2	212.2	3	71.6	0
Executive Time 3	171.6	-1	105.7	-15
Executive Time 4	154.0	0	66.0	9
Executive Time 5	193.3	3	92.9	-3
Average	**169.28**	**1.6**	**88.9**	**7**

Table 3. Different function performance results for Genetic Algorithm when Created Mating Pool Based On Tournament Selection

	Create Mating Pool Based On Tournament Selection			
	Apply Swap Mutation Operator		Apply Bit-Flip Mutation Operator	
	Apply One Point Crossover Operator		Apply One Point Crossover Operator	
	Time (s)	**Cost**	**Time (s)**	**Cost**
Executive Time 1	32.1	-3	55.7	2
Executive Time 2	29.8	-13	72.9	11
Executive Time 3	30.3	5	110.2	0
Executive Time 4	44.9	3	11.5	-2
Executive Time 5	36.8	7	63.6	3
Average	**34.78**	**5.2**	**62.78**	**3.6**

Table 2 and Table 3 show performance results for the Genetic Algorithm. The best result for this method is obtained by using Roulette Wheel Selection, Swap Mutation Operator, and One Point Crossover Operator functions together. This is the combination that comes closest to 0 with an average value of "1.6".

For the Hybrid Model, measurements have been realized by selecting two different function combinations used in the Genetic Algorithm and the percentage of earnings compared to their previous values are explained in table 4.

The Hybrid model gains yield of both time and cost for the functions given in table 4. The reason is that the initial population members consist of better parents with Simulated Annealing. In Table 4, in the first combination, the time value decreased to 28.82 from 34.78 and the cost value decreased to 1.8 from 5.2. Thus, a profit of 17.1% for time and 65.4% for cost was obtained. In the second combination, the same good gains were obtained.

In Figure 5 and Figure 6, a comparison of the time and cost values of the algorithms is given. With the developed Hybrid Algorithm, the genetic algorithm is accelerated and the cost value is improved.

Table 4. Different function performance results and gains for Hybrid Model

	Create Mating Pool Based On Tournament Select.			**Create Mating Pool Based On Roulette Wheel Selection**		
	Apply Swap Mutation Operator			Apply Bit-Flip Mutation Operator		
	Apply One Point Crossover Operator			Apply One Point Crossover Operator		
	Time(s)	**Cost**	**Gain**	**Time(s)**	**Cost**	**Gain**
Executive Time 1	24.5	4	Time: %17.1	121.4	-2	Time: %22.9
Executive Time 2	37.6	1		53.1	3	
Executive Time 3	18.9	1		21.7	-6	
Executive Time 4	27.1	0	Cost: %65,4	83.6	-4	Cost: %57.1
Executive Time 5	36.0	-3		62.5	0	
Average	**28.82**	**1.8**		**68.46**	**3.0**	

In the solution of the Subset Sum Problem, genetic algorithms are used and 1.6 and 5.2 Cost values have been reached in this study. Oberoi et al. obtained 4.4 cost values in a similar study [19]. In the hybrid algorithm developed within the scope of the study, 1.8 cost value was obtained. The hybrid algorithm compared to the genetic algorithm reaches results in a much shorter period.

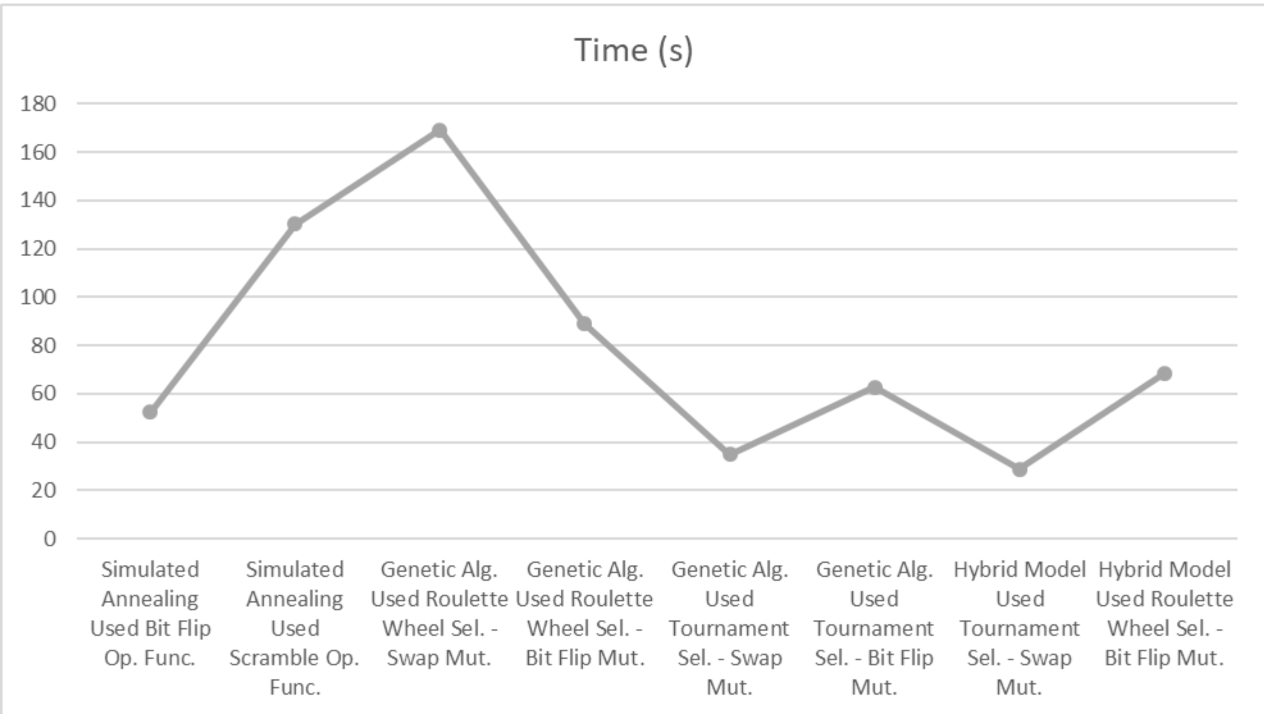

Figure 5. Comparison of time costs of algorithms

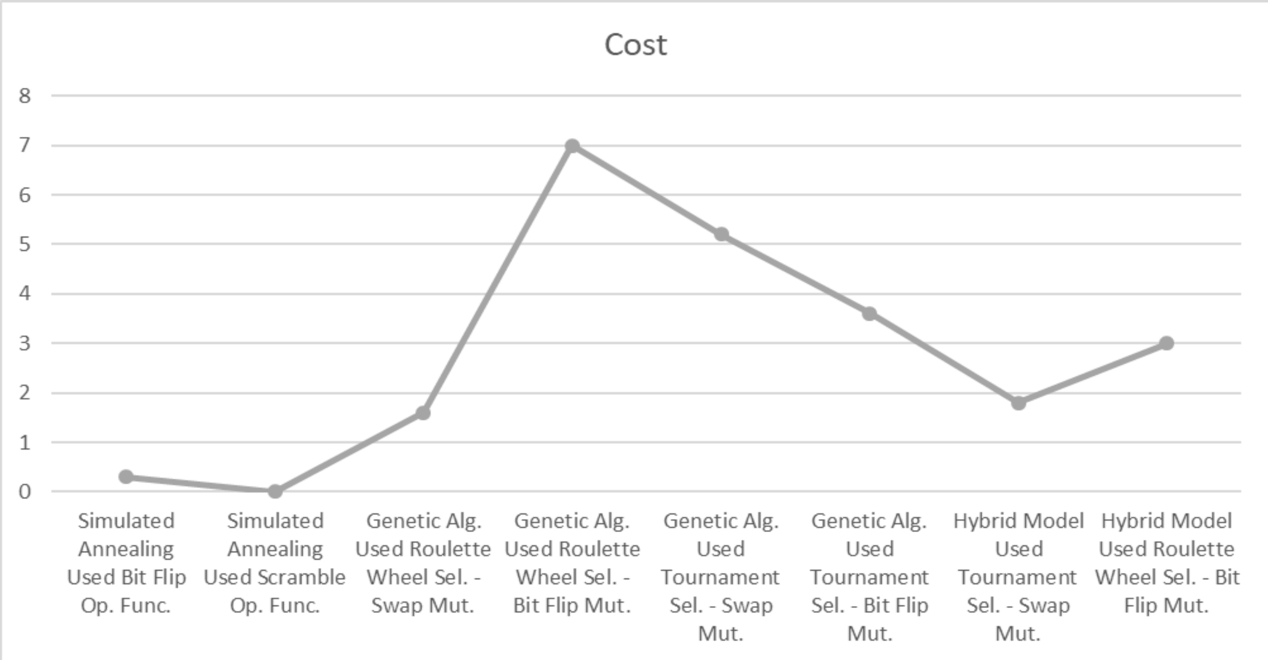

Figure 6. Comparison of cost values of algorithms

A comparison was made in Table 5 for the mean Time values (in seconds) found in the studies done by James *et al.* [20] and Verma *et al.* [21] on the Subset Sum Problem. James *et al.* made their calculations on a small problem (within 50 capacity Subsets) by using the system that has an AMD Athlon XP CPU running at 1.53 GHz, 1 GB RAM. Verma *et al.* made their calculations in Subsets with a capacity of 806 by using the system that has an i3-2120 machine with 4 GB of RAM.

Table 5. Comparison of time costs observed in this study with other studies in the literature

	Dimensions of Subsets	Algorithms	Times (in seconds)
In this study	2000	SA	52.49
		GA	88.9
		Hybrid	**28.82**
In other studies	50	Enumeration	7.10
		CKK	3.60
		SCKK	8.68
	806	Max FD	21.923
		Min FD	146.823

When the values given in Table 5 are analyzed, although the data sets smaller than the data set used in this study (2000 items) were processed, the algorithms used spent more time. The hybrid algorithm developed in this study gave the best result when the data set dimensions were taken into consideration.

4. Conclusions

In the study, three different metaheuristic methods have been applied for the subset sum problem. Creating the initial populations in the hybrid model with Simulated Annealing, where the best results are measured, is also aimed at obtaining better results for the genetic algorithm. Various functions have been used to realize performance analyzes in simulated annealing and genetic algorithms.

This study examines the analysis of these functions and methods by applying them to the Subset Sum problem and also aims to increase the accuracy of the results obtained with the developed hybrid model. Subsequent studies are intended at solving and analyzing the Subset Sum Problem using other metaheuristic methods and their hybrid models. High-Performance Computing and Parallel Programming techniques can provide useful models for improving computing times [22]. Also, the hybrid method is planned to be applied in the field of cryptography in later studies.

Similar studies in the literature were examined and the experimental results found were compared. It was observed that the hybrid algorithm developed in this study gave a successful result among these methods used different sizes of Subset datasets. These results were obtained in terms of both time and cost values too.

Acknowledgments

This work was supported by TUBITAK 2211-A Program.

References

[1]. Ramirez A. J. L., On variations of the subset sum problem, Discrete Applied Mathematics (Elsevier), 1998, 81, l-7.
[2]. Cao1 Z., Liu L., New Algorithms for Subset Sum Problem, cs-DS, 2018, 2, 1-10.
[3]. Mucha M., Nederlof J., Pawlewicz J., Węgrzycki K., Equal-Subset-Sum Faster Than the Meet-in-the-Middle, ESA, 2019, 73, 1-16.
[4]. Dietrich B. L., Escudero L. F., Chance F., Efficient reformulation for 0 – 1 programs: methods and computational results, Discrete Appl. Math., 42, 147-175.
[5]. Isa H. A., Oqeili S., Bani-Ahmad S., The Subset-Sum Problem: Revisited with an Improved Approximated Solution, International Journal of Computer Applications, 2015, 114(14), 0975 – 8887.
[6]. Elshamli A., Abdullah H., Areibi S., Genetic algorithm for dynamic path planning, Canadian Conference on Electrical and Computer Engineering, 2004, 2, 677-680.
[7]. El-Naggar K. M., AlRashidi M. R., AlHajri M. F., Al-Othman A. K., Simulated Annealing algorithm for photovoltaic parameters identification, Solar Energy (Elsevier), 2012, 86(1), 266-274.
[8]. Mamano N., Hayes W. B., SANA: simulated annealing far outperforms many other search algorithms for biological network alignment, Bioinformatics, 2017, 33, 2156-2164.
[9]. Zhu T., Chen S., Zhu W., Wang Y., Optimization of sound absorption property for polyurethane foam using adaptive simulated annealing algorithm, Journal Of Applied Polymer Science, 2018, 135, 46426.
[10]. Ren Y., Zhang C., Zhao F., Xiao H., Tian G., An asynchronous parallel disassembly planning based on genetic algorithm, European Journal Of Operational Research, 2018, 269, 647-660.
[11]. Holland J. H., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan, 1975.
[12]. Topuz R., Yıldırım M., Path Finding and Obstacle Avoidance of Autonomous Robots by Using Sonar Sensors and Heuristic Methods, El-Cezerî Journal of Science and Engineering, 2015, 3(2), 363-371.
[13]. Türken T., Pala T., Modeling and Optimization of Sour Cherry Juice Antioxidant Activity by Using Response Surface Methodology and Genetic Algorithm, El-Cezerî Journal of Science and Engineering, 2016, 3(2), 238-247.

[14]. Mazumder P., Rudnick E. M., Genetic Algorithms for VLSI Design Layout&Test Automation, Prentice Hall Inc, 1999.
[15]. Sivaram M., Krishnan B., Mohammed A., Porkodi V., Exploiting the Local Optima in Genetic Algorithm using Tabu Search, Indian Journal of Science and Technology, 2019, 12, 1-13.
[16]. Fernándeza J. R., López-Camposb J. A., Segadeb A., Vilánb J. A., A genetic algorithm for the characterization of hyperelastic materials, Applied Mathematics And Computation, 2018, 329, 239-250.
[17]. Wang R. L., Agenetic algorithm for subset sum problem, Neurocomputing, 2004, 57, 463-468.
[18]. Goldberg D. E., Genetic Algorithms in Search Optimization and Machine Learning, Addison-Wesley, Reading, 1989.
[19]. Oberoi A., Gupta J., On the Applicability of Genetic Algorithms in Subset Sum Problem, International Journal of Computer Applications, 2016, 145(9), 37-40.
[20]. James R. J. W., Storer R. H., Methods for Solving Subset Sum Problems, Lehigh University, USA, 29-38.
[21]. Verma A., Design and Evaluation of Alternate Enumeration Techniques for Subset Sum Problem, Thesis submitted in partial fulfillment of the requirements for the degree of MS (by Research) in Computer Science, International Institute of Information Technology, 2017.
[22]. Keskenler M. F., Keskenler E. F., Yoğun İşlem Yüküne Sahip Matris Çarpımı Hesaplama Sürelerinin Önbellek Kullanım Optimizasyonu ve Paralel Programlama Teknikleri Kullanılarak İyileştirilmesi, Muş Alparslan Üniversitesi Fen Bilimleri Dergisi, 2018, 6(2), 545-551.