Journal of Theoretical and Applied Information Technology
20th January 2014. Vol. 59 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# SOLVING MAXIMAL COVERING PROBLEM USING PARTITIONED INTELLIGENT FISH ALGORITHM

[1]AMIN JULA, [2]ELANKOVAN SUNDARARAJAN, [3]NARJES KHATOON NASERI, [4]REZA ABIAT
[1, 4]Department of Computer Science, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran.
[2]Centre of Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi, 43600 Selangor.
[3]Department of Computer Science, Shoushtar Branch, Islamic Azad University, Shoushtar, Iran.
E-mail: amin.jula@gmail.com, elan@ftsm.ukm.my, {narges.naseri85, rezaabiat}@gmail.com

**ABSTRACT**

NP-Complete optimization problems are a well-known and widely used set of problems which surveyed and researched in the field of soft computing. Nowadays, because of the acceptable rate of achieving optimal or near-optimal solutions of the mentioned issues, using of nature-inspired algorithms are increasingly considered. One of the familiar problems in the field of NP problems is Maximal Covering Problem which has various applications of pure mathematics to determine the location of mobile network antennas or police stations. In this paper, we introduced a heuristic algorithm called Partitioned Artificial Intelligent Fish Algorithm which using artificial fish-search algorithm, logical partitioning of the search space of this algorithm to several sub-space and change in motor functions in fishes, deals with the suitable, innovative and fast solution of maximal covering problem. The results of implementing this algorithm and comparing it with the performance of some the best known algorithms for solving NP problems will be represented by a very good performance of the proposed algorithm.

**Keywords:** *Artificial Fish Algorithm, Maximal Covering Problem, NP-hard problem.*

## 1. INTRODUCTION

Evolutionary computation techniques, unlike the common search algorithms, works on a set of solutions in the search space and by using cooperation and competition created between solutions can quickly find the optimal solution for complex optimization problems. These techniques are mainly inspired by the process of evolution in th nature. Some of these algorithms can be named: Genetic Algorithm inspired by Genetics Science and Evolution (1975), Simulated Fusion Algorithm inspired of the observations of thermodynamics (1983), The Immune Algorithm Simulating the human immune system (1986), Population ant colony search simulating the behavior of ants in searching food (1991), optimization of particle population imitating the social behavior of birds (1995), Gravitational Attraction Search, Electro-magnetic-ike Algorithm, Imperialist Competitive Search that inspired of socio-political behavior of the countries and memetic algorithms [1-9].

Alongside evolutionary computation techniques inspired by the evolution process in nature, new set of computational techniques has been developed and used which one of the most interesting is partitioned artificial fish which has simulated the food-finding mechanism and fish moving in water [6, 7]. On the other hand, the Maximal Covering problem is one of the classic and known problems in computer science and complexity theory. This problem is one in the field of NP-hand synthetic optimization problems and it is proven that the estimated solution can be achieved in polynomial time $\ln n - \ln \ln n + \Theta(1)$[8]. This problem is also one of 21 NP-Complete raised by Karp [9] which published in 1972. The structure of the paper is as follows that the maximal covering problem is described in detail in the second part. In the third part, partitioned artificial fishes algorithm is surveyed. The Fourth part allocated to the proposed algorithm. The comparison of performance of the proposed algorithm with other optimization algorithms is in the fifth section and finally in the last part the results will be discussed.

In 2012, Yang Yu, Xin Yao and Zhi-Hua Zhou proposed a framework for evolutionary algorithms called SEIA. They tried to apply it in solving single and multi-objective NP-hard problems. In order to analyze its performance they applied SEIP to solve set covering and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Minimum k-set cover problem [10]. In 2008, Pauli Miettinen introduced Positive–Negative Partial Set Cover problem [11] which is a generalization of the Red–Blue Set Cover (RBSC) problem presented by Carr et al. [12], analyzed its complexity and application of this problem in data mining.

Two approximation algorithms are proposed in 2007 by David Peleg to apply to solving two different expansions of set covering problem called Max-Rep problem and Red-Blue Set Cover problem. The author claimed that the algorithms can be used to solve weighted variants of the respective problems [13].In 2002, a novel fuzzy simulated evolution approach proposed by Jingpeng Li and Raymond S. K. Kwan to solve non-unicost set covering problem. The proposed algorithm using seven investigated control parameters has been applied for solving a real-world transportation driver scheduling related to an industry [14].

In order to solve Set Cover Problem and escape from local minima in the process of solving, Hopfield neural network used to introduce a learning algorithm by Pei Zhang et al. in 2006 [15]. Experimental results showed that their approach could achieve proper results in comparison to most of existing approaches. A hybrid approach of genetic algorithm and a stochastic search proposed in 2009 by Yun-long Li, Xiao-min Hu and Jun Zhang to divide sensors into disjoint cover sets so that every cover set can fully cover an area and numbers of cover sets are maximized [16].

In 2010, Shin Yoo introduced a new representation of solutions named Mask-Coding for optimization of the set cover problem. Mask-Coding explores the space of the problem rather than the space of solutions. Experimental results showed that the algorithm could improve the convergence of Pareto-efficient solution set of the multi-objective set cover optimization and its diversity as well [17]. Feng Zeng et al. proposed a shortest path-based algorithm to find maximal set covering problem to use in wireless sensor networks in 2011 [18]. The experimental result showed that the algorithm has had better results in comparison to related works. Using game theory, Qiang Wang, Wenjie Yan and Yi Shen proposed a new distributed algorithm to solve SET K-COVER problem to increase the lifetime of wireless sensor networks in 2012 [19]. They claimed that on the basis of the results of real experiments on small- and large-scale wireless sensor networks, the proposed algorithm can be used in a real application environment and has good performance in convergence and coverage together.

## 2. MAXIMAL COVERING PROBLEM

The maximal covering problem is one of the classic and known problems in computer science and complexity theory. This problem is one in the field of NP-hand synthetic optimization problems and it is proven that the estimated solution can be achieved in polynomial time $\ln n - \ln\ln n + \Theta(1)$. Also, this problem is one of 21 NP-Complete raised by Carp which published in 1972. Maximal covering problem can be defined as follows:

If set of elements $\{1, 2, 3, \ldots, m\}$ that is called the universal set or reference and *n* set that make up the community of the world, are given, the maximal covering problem is finding the minimum number of sets which their community is also including all the elements if universal set. For example, if the universal set as $U = \{1, 2, 3, 4, 5\}$ and set $S = \{\{1, 2, 3\}, \{2, 3\}, \{3, 4\}, \{4, 5\}\}$ are available, with a little investigation it is found that the community of all subsets of *S* includes all elements of universal set.

Although can be held all the elements as are said, but the smallest number of sets which is called set-covering is $\text{SET - COVER} = \{\{1, 2, 3\}\{4, 5\}\}$. The more formal, suppose that *U* is reference set and *S* is a family including subsets of U, a cover, is a sub-family of sets so that their community is *U*. In presenting a maximal covering problem, the algorithm input is an ordered pair $(U, S)$ and an integer *K*; and the question is if a cover set of *K* size or less, exists or not.

In the maximal covering optimization problem, the problem is finding a covering set which formed by the minimum number of sets. The problem is a decision on a matter of finding a set covering of an NP-Complete problem and its optimization version is an NP-hand problem.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

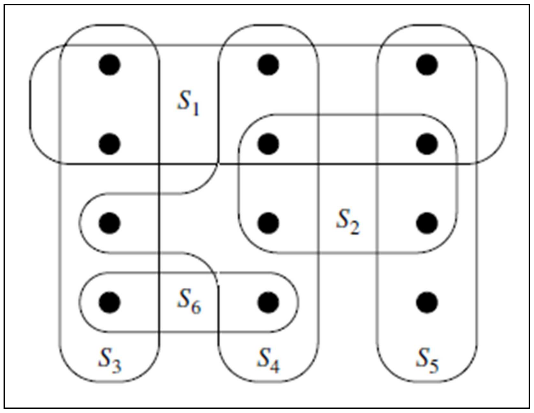


*Figure 1. A Simple Sample Of Set-Covering Problem*

Set covering problem, is the abstraction of many synthetic problems. As a simple example, if it is assumed that *X* is a set of required skills to solve a problem and there is a set of people to work on this problem and suitable, the formation of a committee including minimum possible number of people, so that for the each required skill in *X*, a member of the committee has that proficiency. In the decision version of the set covering problem, the question is whether there is a cover of K size or not? K is another parameter which is specified in the sample of the problem. Decision version of this problem is NP-complete.

As an another example, if the 12 constitutive particles of a reference set as Figure 1 are dispersed in a space and *s1* to *s6* sets be its defined subsets, then, maximal set covering equals to $\{S3, S4, S5\}$ and covering size is equal to 3.

## 3. ARTIFICIAL FISH ALGORITHM (AFA)

In the underwater world, fish can find areas with more food, that it accomplished by individual or group search of fish. So, in the underwater world, usually where there are many more fish with more food, according to this feature, artificial fish model (*AF*) has been provided with some behaviors such as: freely moving behaviors, food seeking, and group moving and following. A fictitious entity of real fish that is used in the analysis and interpretation of issues. The environment in which *AF* lives is basically the search space and domains for other *AFs*. Degree of *food density* in watery area is considered as objective function of the algorithm and shows the state of a single *AF*. *AFA* is an optimization method in which there are freely moving behavior, group moving, food seeking behavior and following behavior that by using them it is possible to explore and search the search space. During the optimization process of *AFA*, individual and environmental information is fully used for searching to reach balance; finally *AF* reaches the place in where the degree of food density and compression is in the maximum volume which means global optimal in NP-Hard problems. *AF* understands external concepts through visual which is shown in Figure 2.

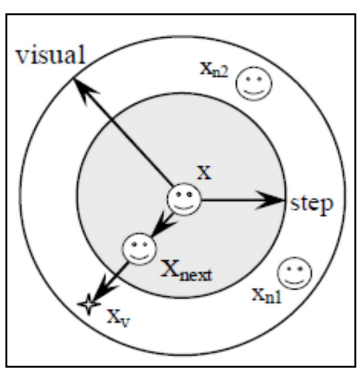


*Figure 2. Understanding Concepts Through Visual By AF*

Journal of Theoretical and Applied Information Technology
20th January 2014. Vol. 59 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

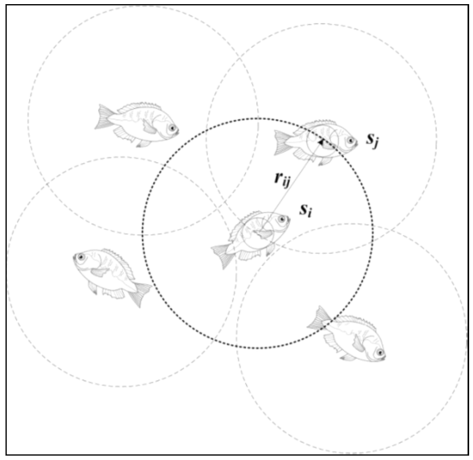

*Figure 3. Moving Fishes In The Search Space*

The current position of an artificial fish is shown by vector $X$ that is $X = (x1, x2..., xn)$. The visual is equal to the visibility of artificial fish and $X_v$ is positioned in the visual field that artificial fish wants to go there. Now if the position of $X_v$, regarding nutrient density, is better than the current position, a step forward is done in the direction of that progress which causes the change of artificial fish position from $X$ to $X_{next}$, but if the current position is better than $X_v$, so patrolling in the visibility will be continued. Step is equal to the maximal length of the moving, the distance between two artificial fish which are in $X_i$ and $X_j$ positions, is shown by $d = \left\| X_i - X_j \right\|$ that is called Euclidian distance.

A function named Rand function is used in *AF* behaviors. Rand function is the function which generates a random number in the range $(0,1]$. *AF* model is including two parts, Variables and Functions that variables including as follows: *X* is current situation of fish, step is the maximal length of the moving, visual is visible, *try_number* is the maximal number of tests and trying and $\delta$ is crowding coefficient (invoice) which $0 \le \delta \le 1$. Functions are including food seeking behavior, group moving behavior, following behavior and freely moving. Figure 3 shows a view of implementing way of artificial fish bunch algorithm.

### 3.1. Freely Moving Behavior

Fish randomly swim in the water; in fact they are looking for food or their companions in a larger range. *AF* randomly selects a position in the visual field; then it moves to that. In fact, this is the basis behavior for food seeking behavior. Freely moving behavior is modeled using Relation 1.

$$X_i^{t+1} = X_i^t + Step.rand() \qquad (1)$$

Where $X_i^t$ is the current position of fish *I* and $X_i^{t+1}$ is the next position.

### *3.2. Food Seeking Behavior*

Trending in food is an essential biological behavior. As a whole, Fish realizes food density in the water by using sight or senses. Then, decides to go there or not. If we randomly select $X_i$ in the *AF* visibility and *AF* is the current position and one state *Y=F(X)* is equal to food density (objective function value) which is shown by *FC*, *X* position is obtained by using Relation 2.

$$X_j = X_i + visual.rand() \qquad (2)$$

Now it is necessary to compare food density in $X_j$ with food density in current position. If $Y_i \le Y_j$ is a step forward to point j from current position which this moving is according to Relation 3. Otherwise other $X_j$ randomly selected and surveyed whether complies

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

movement condition or not. If not, if after *try-Number* times is able to satisfy the moving condition, freely moving behavior will be done.

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\left\| X_j - X_i^t \right\|} + Step.rand\,() \qquad (3)$$

### 3.3. Group Moving Behavior of Fish

Fish naturally gathered in groups in the process of moving that is kind of their habit which protect the group's existence of danger. $X_c$ shows the central situation and *NF* is equal to the number of neighbors in $X_c$ visibility and *N* is subtotal of the *AF*.

Now if $Y_c$ be the food density in central situation and $Y_i$ food density in the current situation, if $Y_c \geq Y_i$ it means that central position has higher food density than the current situation and is not too crowded. Then a moving occurs as a step toward a central situation.

If *nf=0* or moving condition toward to central situation can not satisfied, then food seeking behavior will occur. This behavior expressed using Relation 4.

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^t}{\left\| X_c - X_i^t \right\|} + Step.rand\,() \qquad (4)$$

When a single or a few of the fishes find food, the process of movement of a group of neighbor fishes starts and they follow the finder to quickly arrive to the found food. If $X_i$ is the current position of *AF*, neighbor of $X_i$ is searched and $Y_j$ will be reached.

If $Y_i \leq Y_j$ it means that position of $X_j$ has higher food density than the current position. So a moving occurs as a step toward $X_j$. Otherwise food seeking behavior will be done. Relation 5 is modeling the mentioned behavior. Figure 4 shows the flowchart of approach bunch of artificial fish [20].

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\left\| X_j - X_i^t \right\|} + Step.rand\,() \qquad (5)$$

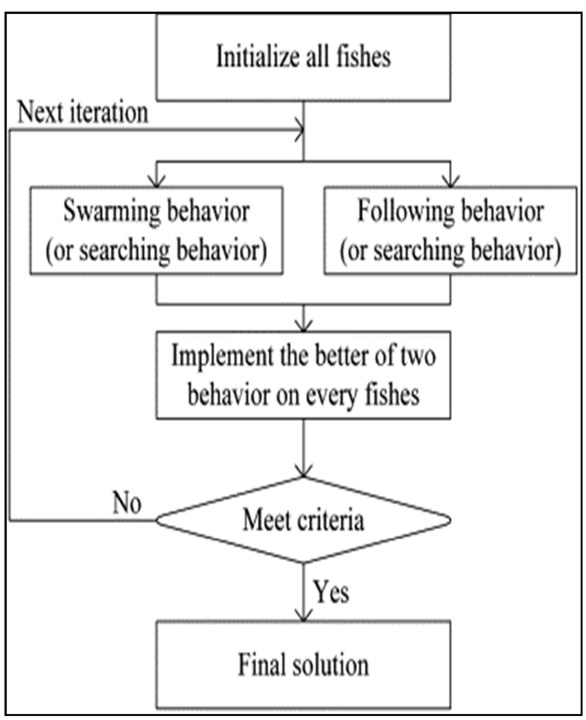


*Figure 4. Artificial Fish Flowchart*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

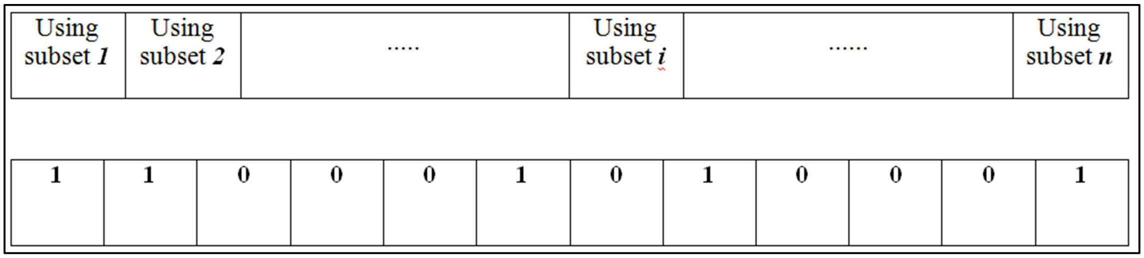

*Figure 5. Designed Pattern For Solutions And A Sample*

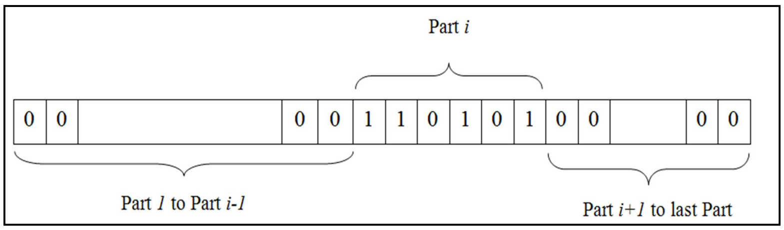

*Figure 6. A Solution In Sub-Space i*

## 4. PARTITIONED ARTIFICIAL FISH ALGORITHM

In this section Partitioned Artificial Fish Algorithm (*PAFA*) is described in details for solving the maximal covering problem. User of this algorithm should determine the number of sets and their members, and also sets of objective which has expected to be created from the initial set interred into the algorithm. It is obvious that the number of elements of initial set is smaller than or equal with the number of elements of objective set and there is not any element that is not a member of the objective set. The objective is to minimize the number of initial sets which with their community it is possible to obtain the objective set.

### 4.1. Initialization the first generation of Solutions

Maximal covering problem considered by the user which is included initial sets and objective set should be located in one file. While executing algorithm *PAFA*, these sets read from file and each set is in one dimensional array from considered values e.g. in type that is a member of algorithm class and designed for this purpose. It is worth to say that because of the number of sets is different so the size of mentioned array is different accordingly.

After this, the initial solutions that in the algorithm each of them plays roles as a fish should be generated. Every solution is an array of zero and one values that in this array if element *i* is zero means that in its generation the *i*th subset is not used and if it is one means that the *i*th subset is used. Another array named Fitness defined which preserve solutions' fitting. This array in fact is defined from a structure with two fields that first field keeps the value of competence and the second field keeps the number of initial sets which used in that solution. It is obvious that the element *i* in this array is related to solution *i*. Also this array has *n* elements so that *n* is the number of initial sets.

In this stage, initial solution created by entering random values of zero and one into solution arrays. After production, in order to calculate fitting value and achievement of the number of used subsets, each solution is sent to the fitting function. Obtained fitting values will be saved in the element concerning that solution in fitness array.

### 4.2. Calculation Function of Fitting Solution

In order to facilitate in implementation of calculation function fitting, one of the prepared classes in C#.Net named Sortedset is used. This class which has designed to create sets, eliminate duplicate values and put values into sets in the order. This design created possibility which can design fitness function easier. What should be concerned is that for solving maximal covering problem, if the value of calculated fitting for one solution be smaller and in better words near to

Journal of Theoretical and Applied Information Technology
20th January 2014. Vol. 59 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

zero – which is desirable value- is a better solution. Also among the solution with the same fitness value, the solution which has less value set used, is more favorable because the algorithm follows combinations that smaller number of the initial sets used in their production. After calculation of fitting values of all the available solutions in search space, solutions arranged according to lager fitting from smallest to largest and the first solution is chosen as the best current solution.

### 4.3. Search Space Division

In the proposed algorithm *PAFA*, the search space can divide into $2^k$ sub-spaces that $k$ should be located in $[1, \log_4^n]$. In this case initial solutions are randomly generated in created subspaces and all the defined movements in the algorithm which come in following will be occurred in sub-spaces. In this order, as shown in Figure 5, 6, each element belongs to sub-space $i$ of a solution contains a random value between 0 and 1 in elements. On the other hand, other elements will have only zero. It means that considered solution located just in one sub-space $i$.

### 4.4. Calculating Distance between Two Fishes

Distance between two solutions or in fact between two fishes in the search space or in every search subspace will be the Euclidian distance. For this purpose, it is necessary to calculate the squared value of the difference between two solutions. Then Euclidian distance of the two solutions will be obtained using the sum of all values calculated for all dimensions.

### 4.5. Food Seeking Behavior

In the proposed algorithm *PAFA*, food seeking behavior of fish $i$ be fulfilled in this order that first it is necessary to take a temporary copy of it. Then fish $i$ randomly selected fish among of 10% initial list of fishes and after naming to j, compare its fitting with that fitting. If numeric value of fitting of fish $i$ be larger than fish $j$ fitting, it means that $j$ is better solution, then fish $i$ should change its elements so that in one third of dimensions of search space its different elements in comparison to fish $j$ become equal to fish $j$ elements. It is meant that fish $i$ moved to be closer to fish $j$.

Then it is essential to calculate the new fitting of *Fish i*. If its fitting converted into a better fitting, means is smaller, so *Fish i* converted into more appropriate fish and will remain in the new place. Otherwise by using the taken copy returns to previous position.

### *4.6. Freely Moving Behavior*

Freely moving behavior which has been used in *PAFA* will run in this order which among the dimensions of a solution, one dimension will be selected randomly. Then if the value of solution in selected dimension be equals to one, it should be converted to zero; otherwise if the value of solution in selected dimension equals to zero, its value has to convert to one. It means that considered fish in one of the dimensions of search space shifted its place freely.

Freely moving qualified behavior has a lot of similarities with the mutation operator in genetic algorithms. After selection and implementation that could due to the possibility or leads to access a better solution or leads to destruction the proper procedure of moving of one fish in searching space. Therefore, time and conditions should be carefully selected to help the fishes to search the search space in a better manner.

### 4.7. Collective Moving Behavior

In collective moving function in *PAFA*, those fishes are considered as neighbors that their maximal distance from each other equals to 2. Parameter $\delta$ will also be 0.66. The function of this algorithm has been implemented so that first takes a copy of *Fish i*. So, the number of fishes that their maximal distance of *Fish i* is equal to 2, should be calculated. Then, the number of obtaining proportion against the total number of fishes will be obtained. If this value is smaller than 0.66, fish $i$ moves in each dimension with probability of 0.2 to the best fish of the group.

After that, the function of new positions will fulfilled. If the new position is better than the previous one, the fish is transmitted permanently and otherwise by using the taken copy, returns to the previous position. The described operation in the above will be done for all the fish except the best one.

### 4.8. Elitetion or preserve the best solutions

As it is mentioned all fishes for different reasons and due to different moving behavior are moving. This replacements lead to access some fishes to the best points of search space. But it is possible for a fish to leave the best found point for finding better points and moves to other places and loose it. Hence, in *PAFA* in order to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

attention to this fact, it is considered to have a function for keeping the best found points which are found by fishes in their movements. In this case, after finishing each iteration of the process of the algorithm, the best found point which is the best position will be saved and preserved using elitetion function. This operation plays an important role in achieving the best solutions in proposed algorithm.

## 5. EXPERIMENTAL RESULTS

Proposed algorithm *PAFA* is implemented in Visual Studio C#.Net 2010 environment. For investigating of the quality and manner of operations of *PAFA*, first, several different problems of maximal covering with different sizes are created. Then these problems has been solved using the proposed algorithm. Classic form of artificial fish algorithm and some other famous optimization algorithms like genetic algorithm and PSO have been implemented and their results of execution shown in designed tables.

The algorithms are executed in a PC that has utilized an Intel Core i5 processor and 8 GB RAM. Number of initialized solutions in the first iteration of the three algorithms was 400. Using the same number of first-phase solutions helped to compare process of execution of the algorithms more accurately.

More complete description of obtained results will take into consideration. Attention to this point is necessary that solving problems must be having complete solutions. It means that always can have subsets in these problems that their community results equal with reference set. Mention at this point is necessary that implementation time of four algorithms is the same. Implementing of these algorithms has finished at the end of the specified time and last obtained results investigated as solutions.

### 5.1. A problem including a reference set with 50 elements and 10 subsets

This problem is considered as a simple problem because both numbers of elements and subsets proportion to the number of elements of reference set is high. This problem has been solved by proposed artificial fish algorithms, classic artificial fish algorithm, genetic algorithm and PSO. For getting more precise results, each algorithm used more than 10 time for solving the problem and taken the average of the results. The obtained results in table 1 can be investigated. As it is shown in table 1, all algorithms are able to cover all elements of the reference set. More accurate investigation shows that PSO and proposed algorithm produced the best results because of using less number of subsets can completely cover the objective set.

### 5.2. A problem including a reference set with 500 elements and 35 subsets

This problem can be considered more difficult regarding to the pervious one; because both reference set and number of its subsets have more elements that will lead to generating more combinations. Moreover, proportion of number of elements of reference set is more than subsets which leads difficulty for solving the problem.

Like pervious problem, in order to get more precise results each algorithm used more than 10 time for solving problems and taken the average of the results. If the results of averaging were decimal, that number is round. The obtained results can observe in table 2. As it is observable in table 2, all the algorithms are succeed in covering all the elements of reference set. Obviously, only the proposed algorithm has produced the best results. Because by using less number of subsets is succeed to complete covering of reference set.

It is necessary that different combinations of subsets could cover reference set, hence that the discrepancies in obtained results could be completely natural.

*Table 1. Results Of A Problem Including A Reference Set With 50 Elements And 10 Subsets*

| PAFA | | Classic AF | | Genetic Algorithm | | PSO | |
|---|---|---|---|---|---|---|---|
| Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets |
| 0 | 5 | 0 | 7 | 0 | 7 | 0 | 5 |

Journal of Theoretical and Applied Information Technology
20th January 2014. Vol. 59 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 2. Results Of A Problem Including A Reference Set With 500 Elements And 35 Subsets*

| PAFA | | Classic AF | | Genetic Algorithm | | PSO | |
|---|---|---|---|---|---|---|---|
| Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets |
| 0 | 17 | 0 | 23 | 0 | 25 | 0 | 20 |

*Table 3. Results Of A Problem Including A Reference Set With 5000 Elements And 250 Subsets*

| PAFA | | Classic AF | | Genetic Algorithm | | PSO | |
|---|---|---|---|---|---|---|---|
| Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets |
| 0 | 87 | 10 | 128 | 12 | 160 | 0 | 113 |

*Table 4. Results Of A Problem Including A Reference Set With 5000 Elements And 250 Subsets*

| PAFA (1) | | PSO (1) | | PAFA (2) | | PSO (2) | |
|---|---|---|---|---|---|---|---|
| Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets |
| 0 | 98 | 18 | 111 | 0 | 98 | 0 | 98 |

| PAFA (3) | | PSO (3) | | PAFA (4) | | PSO (4) | |
|---|---|---|---|---|---|---|---|
| Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets |
| 0 | 98 | 30 | 100 | 18 | 111 | 18 | 111 |

| PAFA (5) | | PSO (5) | | PAFA (6) | | PSO (6) | |
|---|---|---|---|---|---|---|---|
| Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets | Number of not-covered elements | Number of selected subsets |
| 0 | 98 | 30 | 100 | 0 | 98 | 12 | 144 |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

### 5.3. A problem including reference set with 5000 elements and 250 number subset

It is obvious that this problem concerning the previous one is more complex because regarding the previous problem, the number of its reference sets and subsets are much more which leads to produce more combinations. Moreover, proportion the number of elements of the reference set is higher than the number of subsets that significantly leads to the complexity of solving problem. Like pervious problems, in order to get precious each of the algorithms used 10 times for solving problem and average of the results. If the result of averaging is decimal, that number will be rounded. The obtained results insert in table 3.

Initial investigation of the results shows that classical algorithm of artificial fish and genetic algorithm has not succeed to find a set of subsets which can cover of the elements of reference set and 10 and 12 elements has not covered, respectively. But PSO and the proposed algorithm produced the best results because they covered all the elements of reference set. Among two recent algorithms, proposed algorithm averagely covers reference set using 87 subsets against 128 subsets of PSO. It means that using *PAFA* is more suitable for solving the problem.

### 5.4. Supplementary Survey

Investigation of obtained results of implementation of different algorithms for solving different problems of maximal covering shows that in all cases proposed algorithm with attention to parallelizing and fragmentation of search space and performed changes in ways of fish movement, has succeed to get the best solutions. Certainly, in all investigated cases of PSO algorithm has the closest results and actually considered as its serious rival. For more investigation of this matter, a bulky maximal covering problem which only has one solution is designed to be solved by PSO and PAFA. Each algorithm used 6 times for solving the problem and used all the results for comparison. This problem has a reference set with 2000 elements and 400 subsets. The results are in table 4.

As is revealed in the table 4, the proposed algorithm except in one case is the only algorithm that could achieve the complete solution of the problems. Against, PSO could only find the solution in one run and in the other runs it reached solutions close to the optimal. This survey shows that using proposed algorithm is suitable with high confidence for solving maximal covering problem and expected produce and use the optimize considered solution.

## 6. CONCLUSION

In this paper, an innovative and fast algorithm called partitioned artificial intelligent fish algorithm is introduced for solving maximal covering problem which is a practical problem of NP-Complete. The results of implementation of this algorithm and comparing these results with gained results of implantation searching classical artificial fish, genetic and PSO algorithms show that logical partition and with no sense of searching space problem and performed changes in operation method moving actuators of artificial fish could have an effective role in significant improvement of operation and increasing confidence in this algorithm.

## ACKNOWLEDGMENT

This work was partly supported by the Ministry of Higher Education Malaysia [grant number UKM-TT-02-FRGS0205-2010]. The authors would like to express their gratitude for the research grant.

## REFERENCES:

[1] Tang KS, Man KF, Kwong S, He Q. Genetic algorithms and their applications. Ieee Signal Proc Mag 1996; 13: 22-37.

[2] Kirkpatrick S, Gelatto CD, Vecchi MP. Optimization by simulated annealing. Science 1983; 220: 671–680.

[3] Farmer JD, Packard NH, Perelson AS. The immune system, adaptation, and machine learning. Physica D 1986; 22: 187–204.

[4] Dorigo M, Maniezzo V, Colorni A. The ant system: Optimization by a colony of cooperating agents. Ieee T Syst Man Cy B 1996; 26: 1-13.

[5] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks; vol. 4; 1995; pp. 1942-1948.

[6] Amin Jula, Narjes hatoon Naseri, A Hybrid Genetic Algorithm-Gravitational Attraction Search algorithm (HYGAGA) to Solve Grid Task Scheduling Problem, International Conference on Soft Computing and its Applications(ICSCA'2012), Kuala Lumpur, Malaysia, 2012, pp. 158-162.

[7] Amin Jula, Narjes Khatoon Naseri, Alireza Khalilipour, Conducted Electromagnetic-Like Search Algorithm (CELA) to solve

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Graph-Coloring Problem, Science Series Data Report, vol. 4, No.1, 2012, pp. 37-45.

[8] Amin Jula, Zalinda Othman, Elankovan Sundararajan, A hybrid imperialist competitive-gravitational attraction search algorithm to optimize cloud service composition, Memetic Computing (MC), 2013 IEEE Workshop on, Singapore, 2013, pp. 37-43.

[9] Amin Jula, Narjes Khatoon Naseri, Using CMAC to obtain dynamic mutation rate in a metaheuristic memetic algorithm to solve university timetabling problem, European Journal of Scientific Research, Vol. 63, Issue 2, November 2011, pp. 172-181.

[10] Mingyan J, Yong W, Stephan P, Miguel AL, Dongfeng Y. Optimal multiuser detection with artificial fish swarm algorithm. In: ICIC 2007, CCIS 2; 2007; pp. 1084–1093.

[11] Xingwei W, Nan G, Shuxiang C, Min H. An artificial fish swarm algorithm based and abc supported qos unicast routing scheme in NGI. In: ISPA 2006 Ws, LNCS 4331; 2006; pp. 205–214.

[12] Chung CH. Recent application of the maximal covering location planning model. J Oper Res Soc 1986; 37: 735-746.

[13] Richard MK. Reducibility among combinatorial problems, 50 Years of Integer Programming 1958-2008; Springer, 2010. 219-241.

[14] Yang Y, Xin Y, Zhi-Hua Z. On the approximation ability of evolutionary optimization with application to minimum set cover. Artif Intell 2012; 180–181: 20–33.

[15] Pauli M. On the positive–negative partial set cover problem. Inform Process Lett 2008; 108: 219–221.

[16] Carr RD, Doddi S, Konjevod G, Marathe M. On the red–blue set cover problem. In: Proceeding of 11th ACM–SIAM SODA; 2000; 345–353.

[17] David P. Approximation algorithms for the label-covermax and red-blue set cover problems. Journal of Discrete Algorithms 2007; 5: 55–64.

[18] Jingpeng L, Raymond SKK. A fuzzy evolutionary approach with taguchi parameter setting for the set covering problem. In: Proceedings of 2002 Congress on Evolutionary Computation, (CEC2002); 2002; IEEE. pp. 1203-1208.

[19] Pei Z, Rong-Long W, Chong-Guang W, kozo O. An effective algorithm for the minimum set cover problem. In: Fifth International Conference on Machine Learning and Cybernetics; 2006; Dalian; pp. 3032-3035.

[20] Yun-long L, Xiao-min H, Jun Z. A new genetic algorithm for the set k-cover problem in wireless sensor networks. In: IEEE International Conference on Systems, Man, and Cybernetics; 2009; San Antonio, TX, USA; pp. 1405-1410.

[21] Shin Y. A novel mask-coding representation for set cover problems with applications in test suite minimisation. In: 2nd International Symposium on Search Based Software Engineering; 2010; pp. 19-28.

[22] Feng Z, Lan Y, Zhigang C, Huamei Q. A distributed and shortest-path-based algorithm for maximum cover sets problem in wireless sensor networks. In: International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11; 2011; pp. 1224-1228.

[23] Qiang W, Wenjie Y, Yi Shen. N-person card game approach for solving set k-cover problem in wireless sensor networks. Ieee T Instrum Meas 2012; 61: 1522-1535.

[24] Jiang M, Wang Y, Rubio F, Yuan D. Spread spectrum code estimation by artificial fish swarm algorithm. In: IEEE International Symposium on Intelligent Signal Processing (WISP); 2007; pp. 1-6.