Advances and Applications in Mathematical Sciences
Volume 17, Issue 1, November 2017, Pages 95-111

VORTEX SEARCH ALGORITHM FOR SOLVING SET COVERING PROBLEM IN WIRELESS SENSOR NETWORK

M. RAJESWARI[1], J. AMUDHAVEL[2] and P. DHAVACHELVAN[3]

[1,3]Department of Computer Science
Pondicherry University
Puducherry, India
E-mail: raji.rajeswari18@gmail.com
dhavachelvan@gmail.com

[2]Department of CSE
KL University
Andhra Pradesh, India
E-mail: info.amudhavel@gmail.com

Abstract

This work describes Vortex Search optimization algorithm for solving set covering problem in wireless sensor networks. The coverage problem is a combinatorial optimization problem, which is *NP*-hard and modeled by the bipartite graph. The algorithm was tested on randomly generated problems containing up to 1000 rows and 1000 columns. The performance of VSA in solving set covering problem was evaluated on the larger set of generated problems. The computational results showed that VSA is capable of produced highly productive solutions for the set covering problem.

1. Introduction

Wireless Sensor Networks (WSN) has wide-range potential applications towards the researchers in last decades. WSN provide remotely interaction with physical world with the new class of computer systems in the broad sense. This network transforms and manages many applications like homes, environment, and industrial factories [1]. In WSN, the data for processing can

---

2010 Mathematics Subject Classification: 68N13, 97N50.

Keywords: vortex search algorithm, evolutionary algorithm, set covering problem, wireless sensor network, optimization problem.

Received March 10, 2017; Accepted July 20, 2017

be done event-driven and on-demand process. In event drive, the process is activated by more than one or one sensor nodes and in on-demand. The reporting is done from sensor nodes and monitoring station. In the literature, the important problem addressed over the decade is set covering problem. Generally in wireless sensor network, the communication takes place in a medium which is shared by other sensor nodes and so interference may occur. The transmission range occurs over more than one base station, to overcome this problem segmentation of frequency spectrum into channels are assigned to the base station to avoid interference. This segmentation prevents interferences and no two base stations will overlap in the same transmission range.

In wireless ad-hoc communication, the network is highly reliable and the information is updated at regular intervals of time with wireless communication. The WSN consists of sensor nodes which are used to collect the larger amount of data in the dynamic environment.

The size of the sensor nodes are tiny, low battery, minimum cost and randomly node deployment. The main issue to be noted in the wireless environment is Coverage problem [8]. The wireless network does not hold coverage ability in the dynamic environment. When the network topology fails it leads to the poor quality of network coverage.

The interference at sensor depends upon the transmission range and the number of base stations it is covered. The number of base station covering the sensor node is considered to be a lower bound for the number of channels and the number of channels gets reduced. The coverage properties of the sensors and the transmission range of a base station depend on the position, obstacles and transmission power. To minimize the interference, the base station covering the sensor nodes is minimal. The availability of the wireless network will be achieved if the sensor nodes are covered by at least one of the base station.

In this work, the proposed algorithm helps to use set covering problem in WSN by deploying sensor nodes in its optimal position. Here, we use Vortex Search Algorithm (VSA) for handling set covering related issues. VSA is the new single solution based metaheuristic approach for solving global optimization problems. This algorithm is studied under the group of search algorithms which comprise of pattern search and random search algorithms. The random search is generally to be as fixed step size random search, which is iteratively moved towards better positions in the solution search space. The pattern search is similar to the random walk with step size. VSA is inspired by the social behaviour of vortex pattern, which is created by the vertical flow of the stirred fluids in the search behaviour. This approach balances the process of both exploitation and exploration behaviour of the search [2]. This algorithm model the vortex pattern for the search behaviour by using an adaptive step size adjustment methodology. Initially, VSA behaves as the explorative process and the exploration search ability of the algorithm is increased. This algorithm converges towards the near optimal solution and work in an exploitative manner. This approach tunes the current solution towards an optimal solution.

In this work, the authors have proposed VSA to set covering problem in the wireless sensor networks. The objective of the problem is to find the minimum cost coverage set which is cardinality to the union of sensor nodes. Use of Vortex algorithm to solve minimum set coverage in WSN reduces the interference of transmission range in the search path. The remainder sector of the work is organized as follows. Section 2 summarizes the related work on set covering problem in various environments and solving them using the evolutionary algorithm. Sections 3 explain in detail about the problem formulation of the set covering problem. Section 4 describes the flow of VSA. Section 5 shows the empirical result on the performance of VSA. Finally, section 6 concludes and provides future enhancements of the work.

2. Literature Survey

VSA is a single-solution based optimization algorithm which is used to find the better position in its neighbourhood search for each iteration. To avoid getting trapped in local optima, the authors used modified vortex search, the number of parallel vortices are considered as parents' vortex and many child vortexes instead of having the single vortex in the algorithm [3]. In each iteration, the best child and parent vortex are found to obtain the global best in the population.

Doğan and Ayhan [4] proposed vortex algorithm for analog filter group delay optimization. In this methodology, the number of all-pass filters is cascaded to a Chebyshev low-filter. The algorithm is applied to achieve the optimum parameters of the all-filters.

The selection of passive components in the analog active filter design is crucial. VSA is used to find the near optimum selection of passive components in the analog filter. The problem design becomes complicated as the number of possible passive components combinations increases and leads to infeasible in the exhaustive search. The intelligent VSA is used to obtain a fast and near optimum selection of passive components [5]. The performance of the algorithm is compared with other evolutionary algorithms like particle swarm optimization and artificial bee colony algorithm and harmony search algorithm.

Doğan [6] proposed modified VSA, which improves the candidate solution. The original VSA uses Gaussian distribution function at each iteration to generate the candidate solution around the current best solution. This methodology leads to increase in the trap of local optima, the number of local optima points to generate the candidate solutions leads to the trap of local minimum. The authors of this work proposed an adaptive step-size adjustment to replace Gaussian distribution function and to escape from local minima.

Huang et al. [7] proposed gradient-based approximation in VSA to the optimization of the fort of KCS container ship. This proposed vortex search based on gradient approximation shows faster convergence than VSA to trap local minimum and converge towards the global optimum. This proposed algorithm has achieved a significant resistance reduction towards the optimal solution.

Various meta-heuristic approaches is used in VANET to solve coverage and energy related issues in [17-20]. Optimization approach is used in web service based on QoS metrics [21-23]. Bio-inspired algorithm Directed Bee colony optimization algorithm is used to solve scheduling problem [24].

3. Problem Definition

The Set Covering problem (SCP) is the NP-hard problem [9] with computational complexity theory. The SCP is the fundamental problem in the division of covering problems. Given a universe set of elements U and consists of the family of sets S of v sets, $U = S_1, S_2, \ldots, S_v$ of subsets of U, the union of all subsets equals to the universe $S_j \subseteq U$, $j = 1, 2, \ldots, v$. The objective of SCP is to find the smallest sub-collection of S subsets whose union is equaled to the universe U. The SCP aims to find the minimum cardinality such that $\bigcup_{j \in J} S_j = U$. The elements of the Uare called points and given $J \subseteq \{1, 2, \ldots, v\}$ is a point which is to be covered if it belongs to the $\bigcup_{j \in J} S_j$. The SCP is to find the set which uses minimum covering sets. The minimum cost of SCP, each subset S_j, $1 \le j \le v$ has c_j cost coverage and the problem is to find the minimum cost coverage subset, $J \subseteq \{1, 2, \ldots, v\}$. Each point is covered and $\sum_{j \in J} c_j$ is minimized.

For incidence matrix, M of a set of covering problem is formulated as follows. There are $|U|$ rows in matrix M, each point of $u_i \in U$ and v columns in M, one for each subset S_j. The intersection of the i^{th} row and j^{th} column of the matrix M, entry e_{ij}.

$$e_{ij} = \begin{cases} 1, & u_i \text{ in set } S_j \\ 0, & \text{Otherwise} \end{cases}. \tag{1}$$

The objective of SCP is to find the minimum set coverage set, can be formulated as

Minimize

$$\mathbb{Z} = \sum_{j=1}^{v} c_j u_j. \tag{2}$$

Subject to

$$\sum_{j=1}^{v} e_{ij} u_j \ge 1, \; \forall i = \{1, 2, 3, \ldots, r\} \tag{3}$$

$$u_j \in \{0, 1\}, \; \forall j = \{1, 2, 3, \ldots, v\}, \tag{4}$$

where c_j is the cost of the j^{th} column, u_j is the decision variable. The each row in the matrix should be covered by at least one column is ensured by

Equation (3). $r \times v$ matrix $M = e_{ij}$ is the constant coefficient matrix, the elements can be found using Equation (1). Equation (4) is the integrity constraint, the value of u_j can be found.

4. Vortex Search Algorithm

VSA is the meta-heuristic algorithm inspired by the vertical flow of stirred fluids. The natural search behaviour of the vortex algorithm is generated by the vortex pattern using step-size adjustment method. This algorithm provides the balance among exploitation and exploration strategy. In VSA, the candidate solution is generated using a Gaussian distribution function at each iteration, which provides simplicity. To avoid local optima struck, the adaptation of step-size adjustment method is used in VSA. This methodology increases the convergence rate towards obtaining the optimal solution. Figure 1 shows the illustrative search process of VSA, the centre point is chosen and the best solution is chosen from the centre point. In the next iteration, the best solution of the previous iteration is considered as the centre point and search around it. The process continuous for a maximum number of iterations and the best solution is obtained using VSA.

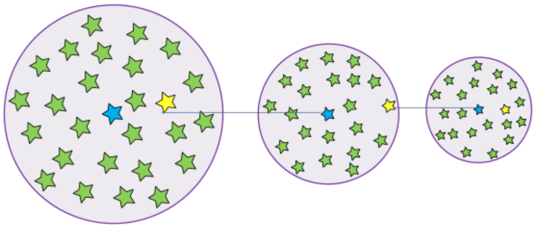


Figure 1. Illustrative search process of Vortex search.

4.1. Generation of Initial Solution

The pattern in VSA is modelled by a number of nested circles in a two-dimensional optimization problem. The outer circle of the vortex is centred first on the search space and initial centre can be calculated using

$$\vartheta_0 = \frac{ul + ll}{2}, \tag{5}$$

where ul and ll are upper and lower limit of $d \times 1$ vectors in d dimensional space.

4.2. Generation of Candidate Solution

The neighbour solution $C_t(s)$, are randomly generated around the initial center ϑ_0 using Gaussian distribution function in the d dimensional space for the iteration t.

$$C_0(s) = \{S_1, S_2, \ldots, S_k\},\ k = 1, 2, \ldots, n, \tag{6}$$

where n is the total number of candidate solutions. The Gaussian distribution function is represented as

$$p(\chi \mid \vartheta, \Sigma) = \frac{1}{\sqrt{(2\pi)^d \mid \Sigma \mid}} \exp\left\{-\frac{1}{2}(\chi - \vartheta)^T \Sigma^{-1}(\chi - \vartheta)\right\}, \tag{7}$$

where χ is the $d \times 1$ vector random variable, Σ is the covariance matrix. The value of the Σ can be computed using Equation (4)

$$\Sigma = \sigma^2 \cdot [\mathrm{I}]_{d \times d}, \tag{8}$$

where σ is the variance of the distribution and I is the identity matrix $d \times d$. The initial standard deviation σ_0 can be calculated using Equation (5)

$$\sigma_0 = \frac{\max(\mathrm{ul}) - \min(\mathrm{ll})}{2}. \tag{9}$$

In a two-dimensional problem, the initial radius value r_0 of the outer circle is taken from variance σ_0.

4.3. Selection Phase

The solution $S' \in C_0(S)$ is selected and memorized from candidate initial solution C_0 to replace ϑ_0. Before selection phase, the search boundaries are ensured for the candidate solutions. The search boundaries can be represented using Equation (6)

$$S_k^i = \begin{cases} \mathrm{rand.}(\mathrm{ul}^{\mathrm{i}} - \mathrm{ll}^{\mathrm{i}}) + \mathrm{ll}^{\mathrm{i}},\ S_k^i < \mathrm{ll}^{\mathrm{i}} \\ S_k^i,\ \mathrm{ll}^{\mathrm{i}} \le S_k^i \le \mathrm{ul}^{\mathrm{i}} \\ \mathrm{rand.}(\mathrm{ul}^{\mathrm{i}} - \mathrm{ll}^{\mathrm{i}}) + \mathrm{ll}^{\mathrm{i}},\ S_k^i > \mathrm{ul}^{\mathrm{i}} \end{cases}, \tag{10}$$

where $k = 1, 2, \ldots, n$ and $i = 1, 2, \ldots, d$ and rand is the uniformly distributed random number. S' is the second circle's centre, the effective radius r_1 in the generation process is reduced in the new circle. The new set of the candidate solution around the new centre is generated. If the value of selection solution is found to be best, it is assigned as the new best solution and memorized. The best solution of the third circle's centre is assigned and memorized for the best solution.

4.4. Radius Decrement Process

The radius of the search process is tuned using inverse incomplete gamma function at each iteration.

$$\gamma(\chi, z) = \int_0^{\chi} e^{-t} t^{z-1} dt, \ z > 0 \tag{11}$$

where $z > 0$ is the shape parameter and $\chi \geq 0$ is the random variable.

The incomplete gamma function, its complementary $\Gamma(\chi, z)$ is represented by

$$\Gamma(\chi, z) = \int_0^{\chi} e^{-t} t^{z-1} dt, \ z > 0 \tag{12}$$

$$\gamma(\chi, z) + \Gamma(\chi, z) = \Gamma(z), \tag{13}$$

where $\Gamma(z)$ is the gamma function and t represents the iteration index. The value of the z is computed by

$$z_t = z_0 - \frac{1}{\text{Max\_Iter}} \tag{14}$$

$$r_0 = \sigma_0 \cdot \frac{1}{\chi} \cdot \text{gammaincinv}\,(\chi, z_0) \tag{15}$$

$$r_t = \sigma_0 \cdot \frac{1}{\chi} \cdot \text{gammaincinv}\,(\chi, z_t). \tag{16}$$

The general workflow of VSA is shown in Figure 2. The pseudo code of VSA is shown in Figure 3.

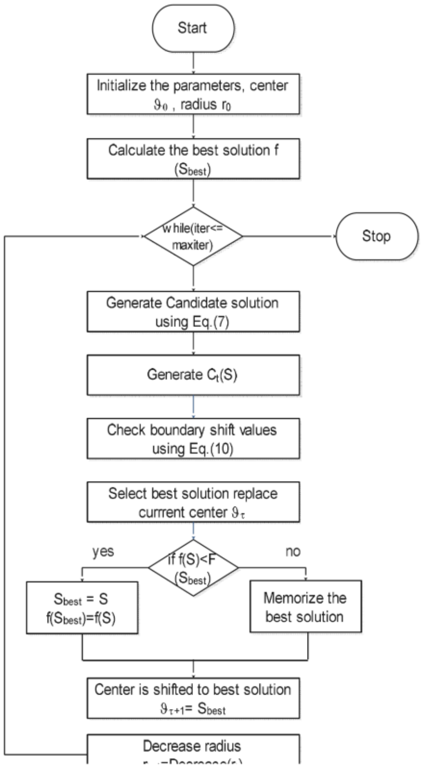
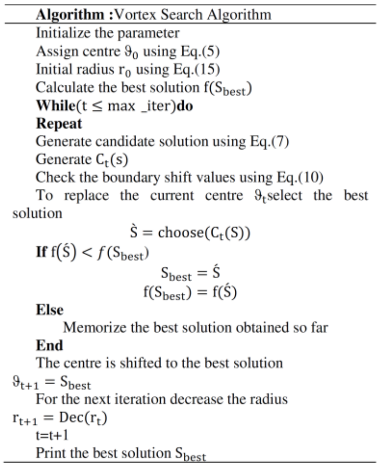


Figure 2. Flowchart of VSA.



```
Algorithm :Vortex Search Algorithm
Initialize the parameter
Assign centre $\vartheta_0$ using Eq.(5)
Initial radius $r_0$ using Eq.(15)
Calculate the best solution $f(S_{best})$
While($t \leq$ max _iter)do
Repeat
Generate candidate solution using Eq.(7)
Generate $C_t(s)$
Check the boundary shift values using Eq.(10)
To replace the current centre $\vartheta_t$select the best solution
        $\grave{S} = \text{choose}(C_t(S))$
If $f(\acute{S}) < f(S_{best})$
        $S_{best} = \acute{S}$
        $f(S_{best}) = f(\acute{S})$
Else
    Memorize the best solution obtained so far
End
The centre is shifted to the best solution
$\vartheta_{t+1} = S_{best}$
For the next iteration decrease the radius
$r_{t+1} = Dec(r_t)$
t=t+1
Print the best solution $S_{best}$
```

Figure 3. Pseudo code of VSA.

5. Experimental Analysis

The proposed methodology for solving SCP optimization problem in Wireless Sensor Networks has been implemented in MATLAB 8.3 in the system configuration of Intel Core i7 Processor with 3.2GHz speed and 4GB RAM. For testing the efficiency of proposed algorithm, testbed designs are prepared. The parameter values were adjusted in an experimental way. The parameter settings for the experimental environment are described in table 1.

Table 1. Configuration Parameter For Experimental Evaluation.

Type	**Method**
Number of Vortexes	200
Maximum Iterations	1000
Initialization Technique	Binary
Termination Condition	Maximum Iterations
Limit	50
Maximum number of columns	0.5% in SCP instance
Maximum number of columns to eliminated	1.2% in SCP instance

These parameter settings had shown good results and Vortex search is tested on 45 standard non-unicost SCP instances which are available in OR library at http://people.brunel.ac.uk/~mastjjb/jeb/info.html. The characteristic of each instance are described in Table 2, each contains 5 problems and shows non-zero entries in the matrix.

Table 2. Details of Instances.

Instance	**#of instances**	**Range**	***r***	***v***	**Optimal solution**
4	10	[1, 100]	200	2000	Known
5	10	[1, 100]	200	2000	Known
6	5	[1, 100]	200	2000	Known
A	5	[1, 100]	300	3000	Known
B	5	[1, 100]	300	3000	Known
C	5	[1, 100]	400	4000	Known
D	5	[1, 100]	400	4000	Known

The proposed algorithm to solve SCP is compared with recent works solving SCP with ABC algorithm [11], Ant cover with local search algorithm [12], genetic algorithm [13], Cutting planes [14], hybrid heuristics algorithm [15] and Lagrangian heuristic algorithm [16]. Table III shows the detailed result comparison of the previous algorithm with VSA.

5.1. Performance Metrics

5.1.1. Average Error Rate

Average Error rate is the difference between the known optimal value and the average best value obtained. The error rate can be calculated using Equation (13)

$$\text{Error rate} = \sum_{i=1}^{n} \frac{\text{optimal}_{\text{Instance}} - \text{Average} - \text{best value}_i}{\text{optimal}_{\text{Instance}}} \tag{17}$$

5.1.2. Convergence Rate

Convergence rate is the percentage of the average of the convergence rate of solutions. The average convergence rate can be calculated using Equation (14)

$$\text{Convergence} = \sum_{i=1}^{n} 1 - \frac{\text{Average} - \text{best}_i - \text{optimal}_{\text{Instance}}}{\text{optimal}_{\text{Instance}}} * 100. \tag{18}$$

5.1.3. Computational Time

The computational time is defined as the total time taken to complete the runtime of the proposed algorithm. The best solution is not set as an epoch value for of algorithm since this evaluation is purely based on random location.

$$\text{Computatio nal Time} = \text{Run time of proposed algorithm.} \tag{19}$$

The results obtained on solving SCP using VSA and compared with other algorithms are shown in Table III. The performance is compared with previous methods and the numbers in the table refer the best value obtained in solving SCP using corresponding algorithms. The computational analysis with respect to the performance metrics is shown in Figures 4-6.

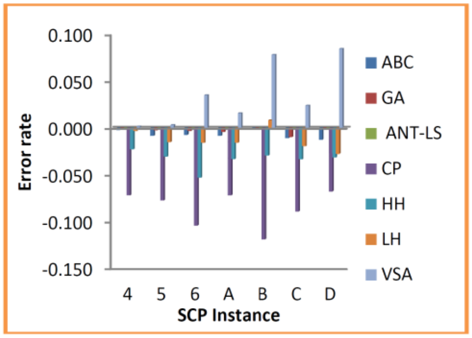


Figure 4. Average Error rate.

Table 3. Experimental Value-Optimal Value.

Instance	Optimum	ABC	GA	ANT-LS	CP	HH	LH	VSA
4.1	429	431	430.7	429	449	433	429	428.98
4.2	512	512	512	512	545	527	512	512
4.3	516	516	516	516	551	534	519	516
4.4	494	494	495.8	494	532	504	499	493.93
4.5	512	512	512	512	528	518	512	512
4.6	560	561	560	560	609	568	560	557.17
4.7	430	430	431	430	450	442	434	429.97
4.8	492	495	492.1	492	512	508	490	488.61
4.9	641	645	645	641	711	664	645	638.29
4.1	514	514	514	514	590	519	516	513.42
5.1	253	255	253	253	268	268	259	251.16
5.2	302	312	304	302	326	317	310	299.62
5.3	226	229	228	226	246	230	229	225.98
5.4	242	242	243.4	242	256	246	245	240.48
5.5	211	211	211	211	226	212	211	210.96
5.6	213	213	213	213	236	216	213	212.48
5.7	293	298	293	293	311	298	299	291.74
5.8	288	288	289	288	323	305	298	286.88

5.9	279	280	279	279	297	286	279	278.98
5.1	265	267	265	265	281	276	270	264.97
6.1	138	142	138	138	149	142	140	133.09
6.2	146	146	146.8	146	162	158	154	140.36
6.3	145	146	145	145	160	149	142	139.85
6.4	131	131	131	131	138	132	130	128.89
6.5	161	161	162.7	161	189	180	167	153.01
A.1	253	256	254.2	253	267	260	256	246.63
A.2	252	254	253	252	281	263	259	247.25
A.3	232	234	234.9	232.8	244	244	235	227.8
A.4	234	234	234	234	257	241	239	231.12
A.5	236	239	236	236	245	239	237	234.85
B.1	69	69	69	69	82	72	68	64.38
B.2	76	76	76	76	91	80	73	69.21
B.3	80	80	80	80	87	82	80	74.08
B.4	79	79	79	79	82	81	80	71.11
B.5	72	72	72	72	78	72	72	67.6
C.1	227	232	229.4	227	242	235	230	223.7
C.2	219	219	220	219	240	224	223	212.71
C.3	243	245	247.4	243	266	256	251	234.42
C.4	219	224	221.1	219	247	227	224	213.63
C.5	215	215	216.1	215	228	219	217	211.44
D.1	60	60	60	60	63	62	61	55.22
D.2	66	67	66	66	72	68	68	59.15
D.3	72	73	72.6	72	78	74	75	64.97
D.4	62	63	62	62	65	64	64	55.66
D.5	61	62	61	61	65	63	62	58.56

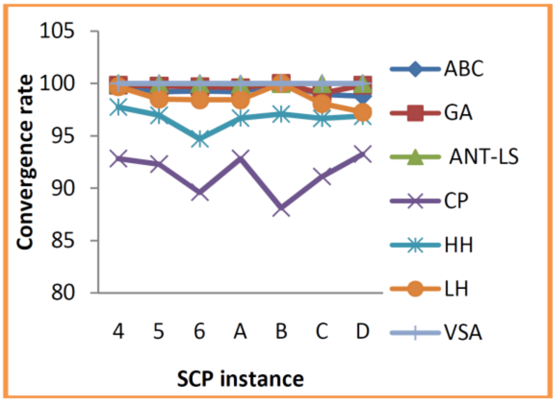


Figure 5. Average Convergence rate.

The evaluation based on error rate shows that VSA yields lesser error rate compared to other competitor methods. 100% of all the instances have achieved lesser error rate when compared with other algorithms. Some of the instances have achieved lesser value than the optimal value specified in the dataset. The error rate obtained by using VSA with other competitor algorithm is shown in figure 4.

The computation based on convergence rate proves VSA achieved 100% convergence rate on all the instances of the SCP. The convergence rate obtained using vortex search and other competitor algorithms are shown in figure 5. The computational time taken to solve SCP is less when compared to other algorithms is shown in figure 6.

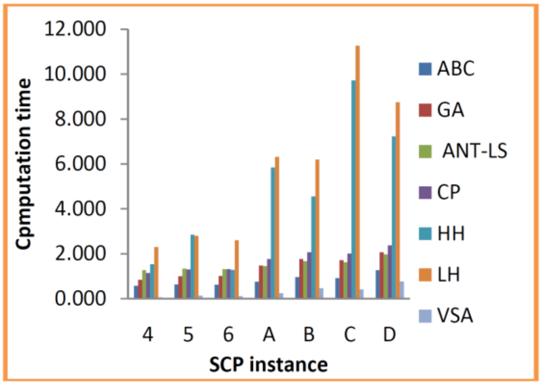


Figure 6. Computational time.

6. Conclusion

In this paper, the author has presented VSA for solving SCP. The performance analysis of various instances in SCP is taken and the results produced good quality. Experiments showed interesting performance with respect to error rate, convergence and computational time for achieving the best solution. From the computational results, VSA is capable of generating optimal solutions for small scale problem instances as well as for larger scale instances.

Acknowledgement

This work is a part of the Research Projects sponsored by the Major Project Scheme, UGC, India, Reference Nos: F.No./2014-15/NFO-2014-15-OBC-PON-3843/(SA-III/WEBSITE), dated March 2015. The authors would like to express their thanks for the financial supports offered by the Sponsored Agency.

References

[1] Cardei, Mihaela and Jie Wu, Energy-efficient coverage problems in wireless ad-hoc sensor networks, Computer Communications 29(4) (2006), 413-420.

[2] Doğan, Berat and Tamer Ölmez, A new metaheuristic for numerical function optimization: VSA, Information Sciences 293 (2015), 125-145.

[3] Sajedi, Hedieh and Seyedeh Fatemeh Razavi, MVSA: Multiple VSA, In Computational Intelligence and Informatics (CINTI), 2016 IEEE 17th International Symposium on, pp. 000169-000174. IEEE, 2016.

[4] Doğan, Berat and Ayhan Yüksel, Analog filter group delay optimization using the VSA, In Signal Processing and Communications Applications Conference (SIU), 2015 23th, pp. 288-291. IEEE, 2015.

[5] Doğan, Berat and Tamer Ölmez, Vortex search algorithm for the analog active filter component selection problem, AEU-International Journal of Electronics and Communications 69(9) (2015), 1243-1253.

[6] Doğan and Berat, A Modified VSA for Numerical Function Optimization, arXiv preprint arXiv:1606.02710 (2016).

[7] Huang, Yujia, Guoxiang Hou, Xide Cheng, Baiwei Feng, Liang Gao and Mi Xiao, A new VSA with gradient-based approximation for optimization of the fore part of KCS container ship, Journal of Marine Science and Technology: 1-11.

[8] Johnson and S. David, Approximation algorithms for combinatorial problems, Journal of Computer and System Sciences 9(3) (1974), 256-278.

[9] D. S. Johnson, The NP-completeness column: An ongoing gulde, Journal of Algorithms 3(4) (1982), 381-395.

[10] Doğan Berat and Tamer Ölmez, A new metaheuristic for numerical function optimization: VSA, Information Sciences 293 (2015), 125-145.

[11] Crawford, Broderick, Ricardo Soto, Rodrigo Cuesta and Fernando Paredes, Application of the artificial bee colony algorithm for solving the set covering problem, The Scientific World Journal 2014 (2014).

[12] Ren, Zhi-Gang, Zu-Ren Feng, Liang-Jun Ke and Zhao-Jun Zhang, New ideas for applying ant colony optimization to the set covering problem, Computers and Industrial.

[13] Beasley, E. John and Paul C. Chu, A genetic algorithm for the set covering problem, European journal of operational research 94(2) (1996), 392-404.

[14] Balas, Egon and Ho. Andrew, Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study, In Combinatorial Optimization, pp. 37-60. Springer Berlin Heidelberg, 1980.

[15] Vasko, J. Francis and George R. Wilson, Hybrid heuristics for minimum cardinality set covering problems, Naval Research Logistics Quarterly 33(2) (1986), 241-249.

[16] Beasley and E. John A lagrangian heuristic for set-covering problems, Naval Research Logistics (NRL) 37(1) (1990), 151-164.

[17] J. Amudhavel, K. Prem Kumar, Cindu Jayachandrameena, Abinaya, Shanmugapriya, S. Jaiganesh, S. Sampath Kumar and T. Vengattaraman, An robust recursive ant colony optimization strategy in VANET for accident avoidance (RACO-VANET), In Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference on, 1-6. IEEE, 2015.

[18] D. Saravanan, V. Agalya, J. Amudhavel and S. Janakiraman, A brief survey on performance analysis and routing strategies on Vanets, Indian Journal of Science and Technology 9(11) (2016).

[19] J. Amudhavel, V. Brindha, B. Anantharaj, P. Karthikeyan, B. Bhuvaneswari, M. Vasanthi, D. Nivetha and D. Vinodha, A survey on intrusion detection system: State of the art review, Indian Journal of Science and Technology 9(11) (2016).

[20] S. Jaiganesh, S. Jarina, J. Amudhavel, K. Premkumar, S. Sampathkumar and T. Vengattaraman, Performance analysis of collision avoidance frame works in VANETS, Indian Journal of Science and Technology 9(11) (2016).

[21] J. Amudhavel, U. Prabu, P. Inbavalli, N. Moganarangan, V. Ravishankar, R. Baskaran and P. Dhavachelvan, Survey and Analysis of web service Composition Strategies: A state of art performance study, Indian Journal of Science and Technology 9(11) (2016).

[22] M. Rajeswari, G. Sambasivam, N. Balaji, M. S. Saleem Basha, T. Vengattaraman and P. Dhavachelvan, Appraisal and analysis on various web service composition approaches based on QoS factors, Journal of King Saud University-Computer and Information Sciences 26(1) (2014), 143-152.

[23] J. Amudhavel, U. Prabu, P. Dhavachelvan, N. Moganarangan, V. Ravishankar and R. Baskaran, A comprehensive analysis and performance assessment on QoS aware web service selection algorithms, In Communication Technologies (GCCT), 2015 Global Conference on, pp. 846-849. IEEE, 2015.

[24] M. Rajeswari, J. Amudhavel, Sujatha Pothula and P. Dhavachelvan, Directed Bee Colony Optimization Algorithm to Solve the Nurse Rostering Problem, Computational Intelligence and Neuroscience, vol. 2017, Article ID 6563498, 26 pages, 2017. doi:10.1155/2017/6563498.