A New Crossover Technique to Improve Genetic Algorithm and Its Application to TSP

ShamimaAkter
Department of Computer Science and Engineering
International Islamic University Chittagong
Chittagong-4318, Bangladesh
shamimaakter95@gmail.com

NazmunNahar
Department of Computer Science and Engineering
University of Chittagong
University-4331, Bangladesh
nazmun4@gmail.com

Mohammad ShahadatHossain
Department of Computer Science and Engineering
University of Chittagong
University-4331, Bangladesh
hossain_ms@cu.ac.bd

Karl Andersson
Pervasive and Mobile Computing
Luleå University of Technology
SE-931 87 Skellefteå, Sweden
karl.andersson@ltu.se

***Abstract*— Optimization problem like Travelling Salesman Problem (TSP) can be solved by applying Genetic Algorithm (GA) to obtain perfect approximation in time. In addition, TSP is considered as a NP-hard problem as well as an optimal minimization problem. Selection, crossover and mutation are the three main operators of GA. The algorithm is usually employed to find the optimal minimum total distance to visit all the nodes in a TSP. Therefore, the research presents a new crossover operator for TSP, allowing the further minimization of the total distance. The proposed crossover operator consists of two crossover point selection and new offspring creation by performing cost comparison. The computational results as well as the comparison with available well-developed crossover operators are also presented. It has been found that the new crossover operator produces better results than that of other cross-over operators.**

Keywords*— *TSP, GA, crossover operator, offspring, chromosome, substring

I.Introduction

Genetic Algorithm (GA) is one of the bio-inspired evolutionary algorithm, uses the ideas of "Normal Selection" and "Genetic Inheritance", initially proposed by John Holland [1]. GA is appropriate for the problems with NP-hard characteristics, where underlying search space cannot be well-defined. Eventually, this algorithm is used in various application domains, where there is a presence of critical number of NP-hard issues. GA explores the population of chromosomes, where each of the chromosomes represents the distinct form of candidate solution for its corresponding problem. GA consists of several operators such as selection, crossover and mutation. The selection is carried out using fitness function. These operators are applied on the candidate solutions to obtain better generation of population. Usually GA consists of five phases. 1) Selection of initial population, which represents the chromosomes and these should be chosen very carefully, 2) The fitness value of each chromosome or the candidate solution, known as individual, in the population is computed by applying fitness function, 3) The individual having highest fitness value is then chosen, 4) Crossover operator is applied to recombine the best two individuals, determined based on fitness value, allowing the generation of new offspring. and 5) Mutation operator is then operated on the new offspring to obtain genetic diversity. TSP is an example of one of the challenging research problems in the field of computing, engineering, operational research, discrete mathematics, graph theory and many others. TSP is defined as the traveling of a salesman in all the cities and return back to the initial city with the least cost. This is a NP-hard problem because time required to solve this cannot be measured in polynomial time. As an example, if we consider 10 cities, then there are 181,440 different ways of solution to the TSP. This is a notable problem which can be connected in various circumstances like programmed penetrating of printed circuit sheets and threading of sweep cells in a testable VLSI circuit [2] as well as in X-ray crystallography [3]. Over the last few decades, a remarkable attention has been given to TSP and different approaches developed to formulate a solution, by taking account of various approaches such as Cutting Planes [4], Particle Swarm Optimization [5], Simulated Annealing [6], Branch and Bound [7], Ant Colony [8, 9], Artificial Neural Networks [10], Tabu Search [11], and Genetic Algorithm [12][13][14][15]. Some of these approaches are heuristics while others are meta-heuristics. However, a comprehensive study demonstrates that GA based approaches have been successfully applied to the TSP [16]. A survey of these GA based approaches related to TSP presented in [15]. Examples of such approaches are Order Crossover with three new variations [17], parallel Genetic Algorithm with order crossover and 2-opt mutation method [18], a sequential crossover operator (SCX) [3], Triple Crossover Operator (TCO [19] and partially mapped crossover [20]. However, in the mentioned approaches [3] [19], it has been observed that some nodes are repeated again and again. Therefore, the diversity or the exploration cannot be achieved. The offspring obtained using the crossover operators [17] become similar to its parent chromosome and thus, evolution cannot be obtained. Consequently, these approaches take much time to fix the above problems. The proposed new crossover technique presented in this paper can address these limitations by minimizing the repetition of nodes. This technique can also reduce the time significantly because it considers only two crossover points and partial selection is made by cost comparison.

GA and TSP and their associated issues are presented in this section. Section II presents the literature review, while methodology will be elaborated in Section III. The results and discussion will be presented in Section IV and conclusion will be in the final section.

II. Literature Review

A lot of work has been done to formulate a solution of TSP, where most of the works emphasize on using Genetic Algorithm (GA). In most of the cases the modifications of the crossover portion of the GA have been carried out.

978-1-5386-9111-3/19/$31.00 ©2019 IEEE

A crossover operator based on cost comparison presented in [21], which creates a single site crossover point. This method is very simple with less complexity but it is difficult to obtain optimal minimum travel cost in TSP.

The partially mapped crossover (PMX) was suggested in [22]. This method chooses two points crossover operator, which sampled the parent chromosome into three substrings, where the middle substring is exchanged.
Sequential Constructive Crossover Operator (SCX) [3] generates high quality solutions to TSP because of containing a new crossover operator. SCX which builds an offspring from a couple of parents utilizing better edges based on their qualities that might be available in the parents' structure keeping up the succession of nodes in the parent chromosomes. The positive side of the SCX calculation is that it is superior to any other solution. The primary drawback of this approach is that the population size is small and they do not utilize any local search technique for improving the quality.
By using Tabu Search a crossover operator developed in [5]. This crossover operator holds the best blend of alleles and maintains a strategic distance from the presentation of most exceedingly bad mix of alleles in the offspring's. This builds the normal fitness of the population and enhances the performance of Genetic Algorithm. This crossover operator contrasted with partially resembled crossover (PMX) and superior results are created over PMX due to this operator.

Order crossover (OX) proposed in [18].It creates offsprings by picking a sub-tour through one parent and keeping the general order of bits of the other parent, which is also based on two points crossover operator.

Cycle crossover (CX) operator proposed in [23]. In this technique bits are come circularly from both parents together with their position which gives a good result in less repetition. However, it produces offspring same as the parents.

A cycle crossover operator (CX2) [24] which is the modified version of the cycle crossover operator developed in [27], generates both offspring from parents using cycle(s) till last bit. They used two different cycles to select bits from parents, as shown below.

P1= (3 4 8 2 7 1 6 5)

P2= (4 2 5 1 6 8 3 7)

After applying there CX2 operator new offsprings are given below.

O1= (4 8 6 2 5 3 1 7)

O2= (1 7 4 8 6 2 5 3)

This is interesting work because no node repetition is present but an impendence of creating the same offspring with a lot of complexity is available for the following types of parents. P1= (1 3 5 7 2 4 6 8)

P2= (2 4 6 8 1 3 5 7)

After applying CX2 operator we get new offspring as given below.

O1= (2 1 4 3 6 5 8 7)

O2= (2 1 4 3 6 5 8 7)

This is repetition of same procedure multiple times and for the large number of nodes it will perform slowly.

The operators discuss above are very popular, however they have some limitations including computational complexity, lack of variation in new offsprings, less efficient for large number of nodes and so on. In order to overcome these problems arise in previous works, a newly developed crossover operator is presented in this paper. The operator is simple enough to calculate, mainly focuses on cost comparison, and also fix the problem of repeating nodes if available. Hence, the crossover operator with other steps of genetic algorithm describe in the following section.

III. Methodology

Genetic Algorithm (GA) depends basically on copying the survival of the fittest among the species created by arbitrary changes in the quality structure of the chromosomes in the evolutionary biology as proposed in [25]. So as to take care of any genuine issue by GA, two primary prerequisites are to be fulfilled.

- The solution of a problem space can be represented by a string and
- An objective function, which determines the fitness of a solution should be characterized [3].

3.1) An Overview of GA

Genetic Algorithm (GA) consists of various phases, which are pointed below.

1. At the beginning, the population of the individual series of nodes is generated arbitrarily for the chosen TSP. At that point, the formation of a cost matrix is used to identify the way between the two cities.
2. By employing fitness function F(x) =1/f(x), the goodness value of every chromosome is required to be calculated
3. Two parent chromosomes are used to obtain the new offspring population by utilizing the crossover operator as will be discussed below.
4. Mutation can be employed if it is necessary.
5. Steps 3 and 4 can be repeated until an optimal solution is achieved.

3.2) The Proposed Method

This section presents the Chromosome design procedures, fitness function selection and the new approach of crossover operator necessary to obtain the minimum optimal distance of TSP.

- Chromosome Design

.To resolve the TSP by employing GA, initially the delineation of the population should be completed. In the population every chromosome is delineated as a series of nodes, where nodes present the cities. Having travelled all the nodes just for one time, it is essential to return to the initial node. The length of the chromosome becomes 7 while the TSP comprises 8 cities. A cost matrix is used to represent the travelling cost between the two cities. Fig 2 shows an example of chromosome for a TSP having 6 cities.

- Fitness Function

The TSP is addressed by the GA, which is employed in minimization problem. The necessary goodness function can be formulated as given in eq (1).

$$F(x) = 1/f(x) \quad (1)$$

Where, f(x) is considered as the objective function used to calculate the overall cost of a tour delineated by a string. Chromosomes are determined with a likelihood associated with their fitness value and afterward replicated into the generation to come. This procedure is carried out in the selection procedure. Exceedingly fittest chromosomes can keep losing from the best arrangement.

- Crossover operator

New solution space is generated by generating new offsprings from prior ones. This is followed by the application of the new crossover operator over the new solution space. Initially, two parents are arbitrarily chosen. Then, two crossover points are found to be fixed, one in position 3 while the other in position 7.The comparison and the selection of the fittest portion from the parents are carried out. Then in case of any replicated node is discovered from the newly created chromosome they need to be deleted. This will allow the shifting of the remaining nodes as left as possible. Then by adding the remaining nodes those are missing from the list in the right side or left side by making some comparisons.

The algorithm for this newly crossover technique is elaborated below.

Step 1: Randomly select two parents P1 and P2 and measure their cost. The cost matrix is represented in Table IV. The representation illustrates in Fig 3 where two crossover points at position 3 and position 7 are selected. Therefore, the chromosome is divided into three pattern (named first, middle and last).

Step 2: Select the first pattern from P1 which is 4-3 and these need to be added to the offspring1. Afterwards compare middle pattern from both P1 and P2; and select the one having lowest cost (e.g. from 5-7-6 and 6-1-3 we select 6-1-3 pattern which have minimum cost) and add to the offspring1. Again select the last pattern from P2 which fill the chromosome size. Again we do the same thing to create offspring2, where we select first pattern from P2, middle pattern by comparing both from P1 and P2 and select the one having minimum cost and then select last part from P1.

offspring1 remove node 3 and from offspring2 remove node 1.

Step 3: Remove the duplicate nodes, e.g. from offspring1 remove node 3 and from offspring2 remove node 1.

Step 4: Now we have to find out the missing nodes, for offspring1 node 5 is missing. To insert node 5 we have to compare with first and last node to have the minimum cost e.g. the cost of 5-4 is 65 and the cost of 2-5 is 88. Therefore, we insert node 5 before node 4. Same procedure is applied for offspring2

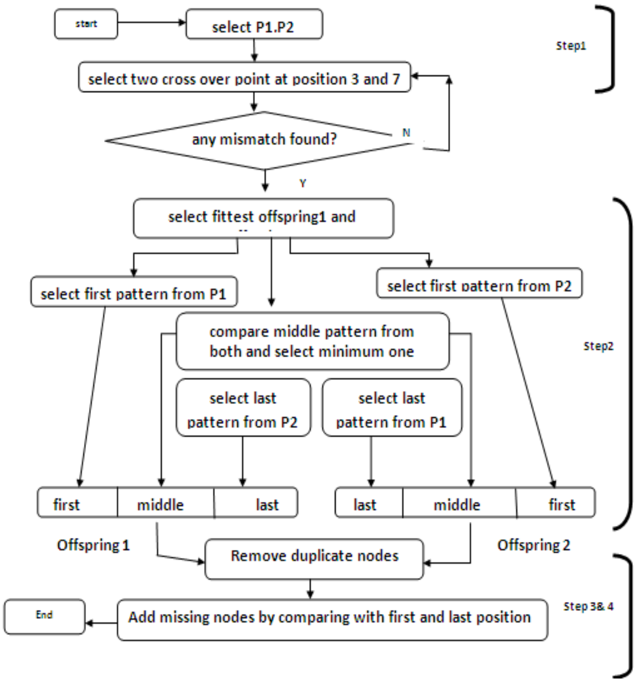
start
select P1.P2
Step1
select two cross over point at position 3 and 7
any mismatch found?
N
Y
select fittest offspring1 and
select first pattern from P1
select first pattern from P2
compare middle pattern from both and select minimum one
Step2
select last pattern from P2
select last pattern from P1
first
middle
last
last
middle
first
Offspring 1
Offspring 2
Remove duplicate nodes
End
Add missing nodes by comparing with first and last position
Step 3& 4

Fig 1:Flowchart of the algorithm

1 4 3 6 2 5

Fig 2:Representation of chromosome

4	3	5	7	6	1	2	cost=282
4	5	6	1	3	7	2	cost=329

Fig. 3. Parent chromosome representation

Table I. Partial Chromosome Representation of offspring1 and Offspring2

offspring1		**4**	**3**	**6**		**1**	**3**	**7**	**2**
offspring2		**4**	**5**	**6**		**1**	**3**	**1**	**2**

Table II. Partial chromosome representation of offsprings

offspring1	**4**	**3**	**6**	**1**	**3**	**7**	**2**
offspring2	**4**	**5**	**6**	**1**	**3**	**1**	**2**

offspring1	4	3	6	1	7	2	
offspring2	4	5	6	1	3	2	

Table III. Final chromosome representation of offsprings

offspring1	5	4	3	6	1	7	2	cost=186
offspring2	4	5	6	1	3	2	7	cost=295

Table IV. Portion of the cost matrix

Node	1	2	3	4	5	6	7
1	100	75	99	9	35	63	8
2	51	100	86	46	88	29	20
3	50	5	100	16	28	35	28
4	20	45	11	100	59	53	49
5	86	63	33	65	100	76	72
6	36	53	89	31	21	100	52
7	58	31	43	67	52	60	100

IV. Computational Result and Discussion

Visual C++ in the Intel core-i5 PC having a speed of 5GHz with a RAM of 16GB in Window 10 pro environment has been used to implement the proposed method as described in the previous section and experiment has been conducted for TSPLIB instances [26].

We perform the proposed crossover operator along with two traditional crossover operators SCX and TCO on eight benchmark instances which are taken from the TSPLIB [26]. Although the method proposed in this paper requires a number of calculations, fewer interactions are required by the which in turn reduce the cost of memory and time. The past strategies were great however our proposed method gives optimal solution in less time than the previous methods. To solve this problem using GAs, the genetic parameters are set as population size, 30; maximum generation, 10; crossover probability, 0.8; mutation probability, 0.1. The Table V reports the average time of convergence (in second) by the algorithms which is measured by the following formula-

$$\text{Excess}(\%) = \frac{\text{Solution Value} - \text{Optimal Solution Value}}{\text{Optimum Solution Value}} \times 100\%$$

In Table V, we compare our proposed method with two traditional crossover operators SCX and TCO in terms of average time. For instances ftv55, the average time of SCX and TCO is 35.11 seconds and our proposed method average time is 30.78 seconds for the same optimal solution. For almost every instances our proposed method's average time is less than the other two methods. However, the rendering of various crossover operators linked with asymmetric TSPLIB cases is illustrated in Fig. 4. The SCX operator appears possessing wide range of variations, although it is not the optimal. Although TCO operator demonstrated fewer variations, it is the worst. On the contrary, the proposed crossover operator displays the optimal result. However, it is restricted to limited range of variations and gets stuck in local minimum quickly.

V. Conclusion and Future Work

An efficient Genetic Algorithm program with having a noble crossover operator to solve TSP is proposed in this paper. An analysis among SCX, TCO and proposed crossover operator for some benchmark TSPLIB instances also presented in the result section.

The experimental results demonstrate that the proposed cross over operator outperforms other operators by providing solution in fewer iterations which reduce the time and memory significantly.

Table V. Summary of the results of different crossover operators for asymmetric TSPLIB instances

		Opt.	SCX	TCO	Proposed Crossover Operator
Instance	n	Sol.	Average	Average	Average
			Time	Time	Time
br17	17	39	0.11	0.9	0.9
ftv33	34	1286	2.25	1.98	2.26
ftv35	36	1473	9.69	6.92	7.68
ftv38	39	1530	6.89	7.12	5.59
p43	43	5620	22.98	20.22	20.45
ftv44	45	1613	19.22	15.34	15.19
ftv47	48	1776	25.99	18.10	17.49
ry48p	48	14422	25.73	27.31	27.78
ft53	53	6905	36.73	31.11	28.04
ftv55	56	1608	35.11	35.21	30.78
ftv64	65	1839	76.56	81.01	80.80
ft70	70	38673	74.19	73.99	69.49
ftv70	71	1950	58.69	58.14	58.09
kro124p	100	36230	142.02	140.01	138.89
ftv170	171	2755	259.60	248.91	248.43

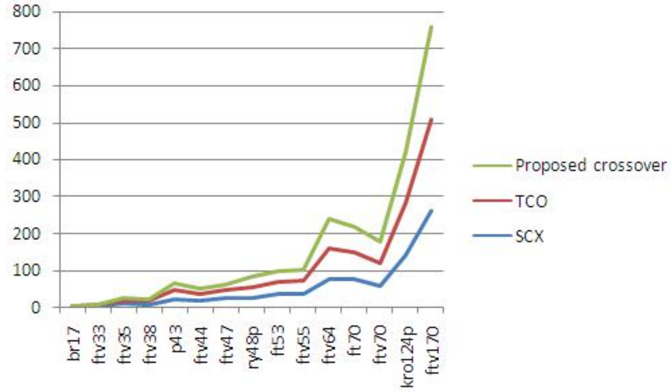

Fig. 4. Performance analysis of different crossover operators for asymmetric TSPLIB instances

In this present study, we only consider SCX and TCO to compare. The aim of the paper was to compare the performance of the various cross over operators with the proposed one. The program produced good results for various problem sizes. However, in future the algorithm will be further enhanced to address the problem of getting stuck in the local minima as well as to address different types of uncertainty found in various application domains, where belief rule based are widely used [27][28][29][30][31]. The GA with newly developed crossover operator will be used to support the learning procedures of belief rule base to obtain better accuracy in prediction. Further, a balanced algorithm allowing the optimal exploitation and exploration will be investigated.

Acknowledgement

This study was funded by Swedish Research Council under Grant 2014-4251

References

[1] Holland, John Henry. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[2] C. Ravikumar, "*Parallel techniques for solving large scale travelling salesperson problems*," Microprocessors and Microsystems, vol. 16, no. 3, pp. 149–158, 1992.

[3] H. Z. Ahmed, "*Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator*," in Proceedings of the International Journal of Biometrics & Bioinformatics (IJBB), vol. 3, p. 96, 2010.

[4] P. Miliotis, "*Using cutting planes to solve the symmetric Travelling Salesman problem*," Mathematical Programming, vol. 15, no. 1, pp. 177– 188, 1978.

[5] J. Kennedy, R. C. Eberhart, and Y. Shi, Swarm intelligence, morgankaufmann publishers, Inc., San Francisco, CA, USA, 2001.

[6] S. Kirkpatrick and G. Toulouse, "*Configuration space analysis of travelling salesman problems*," Le Journal de Physique, vol. 46, no. 8, pp. 1277–1292, 1985.

[7] S. Kirkpatrick and G. Toulouse, "*Configuration space analysis of travelling salesman problems*," Le Journal de Physique, vol. 46, no. 8, pp. 1277–1292, 1985.

[8] M. Dorigo and L. M. Gambardella, "*Ant colony system: a cooperative learning approach to the traveling salesman problem*," IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 53–66, 1997.

[9] M. Dorigo, V. Maniezzo, and A. Colorni, "*Ant system: optimization by a colony of cooperating agents,*" IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 26, no. 1, pp. 29–41, 1996.

[10] S. Bhide, N. John, and M. R. Kabuka, "*A Boolean Neural Network Approach for the Traveling Salesman Problem*," IEEE Transactions on Computers, vol. 42, no. 10, pp. 1271–1278, 1993.

[11] F. Glover, "Artificial intelligence, heuristic frameworks and tabu search," Managerial and Decision Economics, vol. 11, no. 5, pp. 365–375, 1990.

[12] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "*Genetic algorithms for the travelling salesman problem: a review of representations and operators*," Artificial Intelligence Review, vol. 13, no. 2, pp. 129–170, 1999.

[13] Michalewicz, Zbigniew. "*Evolution strategies and other methods." Genetic Algorithms+ Data Structures= Evolution Programs*. Springer, Berlin, Heidelberg, 1996. 159-177.

[14] C. Moon, J. Kim, G. Choi, and Y. Seo, "*An efficient genetic algorithm for the traveling salesman problem with precedence constraints*," European Journal of Operational Research, vol. 140, no. 3, pp. 606–617, 2002.

[15] J.Y. Potvin, "*Genetic algorithms for the traveling salesman prob*lem," Annals of Operations Research, vol. 63, pp. 339–370, 1996.

[16] Gen, M., and R. Cheng. "*Genetic algorithms and Engineering design* "John Wiley&Sons Inc." 605 Third Avenue (1997): 42-46.

[17] K. Deep and H. M. Adane, "*New variations of order crossover for travelling salesman problem*," International Journal of Combinatorial Optimization Problems and Informatics, vol. 2, no. 1, 2011.

[18] L. Davis, "Applying adaptive algorithms to epistatic domains," IJCAI, vol. 85, pp. 162–164, 1985.

[19] S. Akter, et al.,"*Using Genetic Algorithm with Triple Crossover to Solve Travelling Salesman Problem*", 1st International Conference on Machine Learning and Data Engineering, nov 20-22, 2017, Sydney.

[20] N. Kumar, R. K. Karambir et al., "*A comparative analysis of pmx, cx and ox crossover operators for solving traveling salesman problem*,"accepted on International Journal of Latest Research in Science and Technology, vol. 1, 2012,Sydney.

[21] V. Dwivedi, T. Chauhan, S. Saxena, P. Agrawal, "*Travelling Salesman Problem using Genetic Algorithm*", National Conference on Development of Reliable Information system,Techiniques and Related issues.

[22] D. Goldberg and R. Lingle, "*Alleles, Loci and the Traveling Salesman Problem,*" in Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications, vol. 1985, pp. 154–159, Los Angeles, USA.

[23] I. M. Oliver, D. J. d. Smith, and R. C. J. Holland, "*Study of permutation crossover operators on the traveling salesman problem*," in Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA, USA, 1987.

[24] K. P. Ghadle and Y. M. Muley, "*Travelling salesman problem with {MATLAB} programming,*" International Journal of Advances in Applied Mathematics and Mechanics, vol. 2, no. 3, pp. 258–266, 2015.

[25] D. E. Golberg, "*Genetic algorithms in search, optimization and machine learning, Addison-Wesley Publishing Company, 1989.*"

[26] G.Reinelt, http://www.iwr.uniheidelberg.de/group/comopt/software, october, 2018.

[27] T. Mahmud and MS Hossain," *An evidential reasoning based decision support system to support house hunting*", International Journal of Computer Applications, Vol. 57, No.21, pp 51-58, 2012.

[28] M. S. Hossain, M. S. Khalid, S. Akter, and S. Dey, "*A belief rule-based expert system to diagnose influenza,*" in 2014 9th International Forum
on Strategic Technology (IFOST), Oct 2014, pp. 113–116.

[29] R. Ul Islam, K. Andersson, and M. S. Hossain, "*A web based belief rule based expert system to predict flood,*" in Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services, ser. iiWAS '15. New York, NY, USA: ACM, 2015, pp. 3:1–3:8. [Online]. Availablehttp://doi.acm.org/10.1145/2837185.2837212.

[30] M. S. Hossain, K. Andersson, and S. Naznin, "*A belief rule based expert system to diagnose measles under uncertainty,"* in Proceedings of the 2015 International Conference on Health Informatics and Medical Systems (HIMS'15) :, 2015, pp. 17–23, godk¨and; 2015; 20150525 (karand). [Online]. Available: http://www.world-academy-of-science.org

[31] M. S. Hossain, S. Rahaman, A. Kor, K. Andersson, and C. Pattinson, "*A belief rule based expert system for datacenter pue prediction under uncertainty*," IEEE Transactions on Sustainable Computing, vol. 2, no. 2, pp. 140–153, April 2017.

.

Identify applicable funding agency here. If none, delete this text box.