Defence Science Journal, Vol. 71, No. 6, November 2021, pp. 784-790, DOI : 10.14429/dsj.71.16174

Multiple Region Coverage Path Planning for Autonomous Underwater Vehicle

Shristi Deva Sinha

DRDO-Institute for Systems Studies and Analyses, Delhi - 110 054, India
E-mail: shristideva_issa@yahoo.com

ABSTRACT

Coverage path planning methodology for an autonomous underwater vehicle to search multiple non-overlapping regions has been proposed in the paper. The proposed methodology is based on the genetic algorithm (GA). The GA used in the proposed methodology has been tuned for the specific problem, using design of experiment on an equivalent travelling salesman problem benchmark instance. Optimality of the generated paths was analysed through simulation studies. Results indicated that the proposed methodology generated shorter paths in comparison to conventional methods.

Keywords: Autonomous underwater vehicle; Coverage path planning; Multiple region search, Genetic algorithm; Travelling salesman problem; Design of experiment; Latin square; Fractional factorial experimental design

1. INTRODUCTION

Autonomous underwater vehicles (AUVs) have been widely developed in the last few decades[1]. Their use includes persistent intelligence, surveillance, reconnaissance (ISR) operations, covert operations in confined and complex areas[2], including mine counter measure (MCM) missions. AUV carries out a given mission in a hazardous undersea environment before it runs out of battery, so mission timing is extremely critical to mission success[1]. Conventional deployment of AUVs typically involves a human user specifying a series of waypoints[3]. But, with limited battery endurance of the AUV, the path planning problem is one of the most critical problems to tackle. It is concerned with determining an optimal path from the start to destination to complete a given mission[4].

Extensive works on AUV start-to-goal (STG) path planning are available in the literature. It involves determining a path that incurs the least energy cost or length without hitting any obstacles[4]. Diverse methods like Potential field method[5], A* algorithm[6], genetic algorithm[7-8](GA), particle swarm optimisation[9] (PSO), and neural network[10] (NN) have been used for STG path planning. Another targeted research area in AUV path planning is Task Scheduling. This problem involves determining an optimal path to visit several predetermined targets, perform some tasks at the target locations, and then return home. In literature, travelling salesman problem[11] (TSP) has been used to tackle such a problem. Task scheduling is extensively used in AUV monitoring missions. Coverage path planning (CPP) is another widely studied area of AUV path planning. CPP involves determining an optimal path to exhaustively search a given area. It has wide applications in MCM, ISR and search missions. Diverse solutions to CPP[12-15] for AUV are available in the literature.

AUV search mission requires the covering of single or multiple areas. Possible scenarios are:

- Single AUV searching a single area
- Multiple AUVs searching multiple areas
- Multiple AUVs searching a single area
- Single AUV searching multiple areas

CPP has been extensively studied for the first three scenarios. Whereas, the fourth scenario has not received any attention. This lack of research could be due to the low battery endurance of the AUV. However, in recent years, high endurance AUVs are being developed. There are currently available AUVs[15] with sufficient endurance to exclusively search in multiple regions. So, path planning would be needed for efficiently conducting such missions. The problem involves the determination of an optimal AUV path covering multiple spatially distributed regions. The specific problem is not only a CPP problem but a combination of Task Scheduling and CPP. It consists of finding inter-region (connecting the regions) and intra-region (covering each region) optimal path.

A similar problem is observed for unmanned aerial vehicle (UAV) termed as the integrated travelling salesman-coverage path planning (TSP-CPP) Problem[4]. Dynamic programming (DP) based solutions[4] is proposed for the problem, but these suffer from the curse of dimensionality. The equivalent problem for AUV has enormous size due to low underwater sensor ranges. Hence, these solutions are not applicable for AUV. The paper proposes a GA based CPP methodology for a single AUV multi-region search mission. The GA used in

Received : 31 July 2020, Revised : 10 August 2021
Accepted : 06 September 2021, Online published : 22 October 2021

the methodology is tuned using a design of experiment (DoE) on a comparable TSP benchmark instance. The optimality of the path generated using the proposed methodology is analysed through simulation studies.

2. PROBLEM FORMULATION

Consider a scenario where an AUV is assigned to search n non-intersecting rectangular regions of different sizes. These regions are indexed as i=1 to n. Say, AUV is launched at a point with co-ordinates s. It exhaustively searches all the n regions and is retrieved at a point with co-ordinates d. These launch and retrieval points are indexed as i=0 and n+1, respectively. It is assumed that the AUV conducts the mission at a constant depth, so the problem is formulated in two dimensions.

Let $P_i=(p_{im})$: $m=1,2,\ldots,n_i$ be the ordered sequence of co-ordinates of the points on the i^{th} search region. The i^{th} search region is fully covered if and only if the AUV moves through the entire sequence of points of P_i in order. To find the sequence of visit to n search regions, a decision variable x_{ij} is used i,j=0, 1,2,,…,n+1 such that:

$x_{ij}= 1$ if AUV moves from i^{th} region or starting point to j^{th} region or retrieval point

$x_{ij}= 0$ Otherwise

The total distance connecting the n search regions is:

$$D_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} d\left(p_{ini}, p_{j1}\right) \tag{1}$$

where $d(a, b)$ is the Euclidean distance from location a to location b. p_{ini} and p_{j1} are the last and first point in the ordered sequence on i^{th} and j^{th} search region respectively.

The total distance within the n search regions is:

$$D_2 = \sum_{i=1}^{n}\sum_{m=1}^{n_i-1} d\left(p_{im}, p_{i(m+1)}\right) \tag{2}$$

where p_{im} and $p_{i(m+1)}$ are the consecutive points in the ordered sequence on the i^{th} search region.

The distance from the launch point to the first search region is:

$$D_3 = \sum_{i=1}^{n} x_{0i} d\left(s, p_{i1}\right) \tag{3}$$

where, p_{i1} is the first point in the ordered sequence on the i^{th} search region.

The distance from the last region to the retrieval point is:

$$D_4 = \sum_{i=1}^{n} x_{in+1} d\left(p_{ini}, d\right) \tag{4}$$

where p_{ini} is the last point in the ordered sequence on the on the i^{th} search region.

The TSP-CPP problem[4] is formulated as finding the values of P_i and x_{ij}, such that the distance z is minimised.

$$\text{Min } z=D_1+D_2+D_3+D_4 \tag{5}$$

Subject to the constraints:

$$\sum_{j=0, i\neq j}^{n+1} x_{ij} = 1, \forall i = 0 \text{ to } n+1 \tag{6}$$

$$\sum_{j=0, i\neq j}^{n+1} x_{ij} - \sum_{j=0, i\neq j}^{n+1} x_{ji} = 0, \quad \forall i = 0 \text{ to } n+1 \tag{7}$$

The solution to the TSP-CPP problem yields the optimal path for the mission. The proposed methodology for the solution to the formulated problem is explained in the next section.

3. PROPOSED METHODOLOGY

The proposed methodology consists of two steps:

- Cell decomposition of each search regions to reduce the TSP-CPP to TSP
- Determine the optimal path by solving the TSP using GA.

The proposed methodology is illustrated in Fig. 1.

Each step of the proposed methodology is explained in the following subsections.

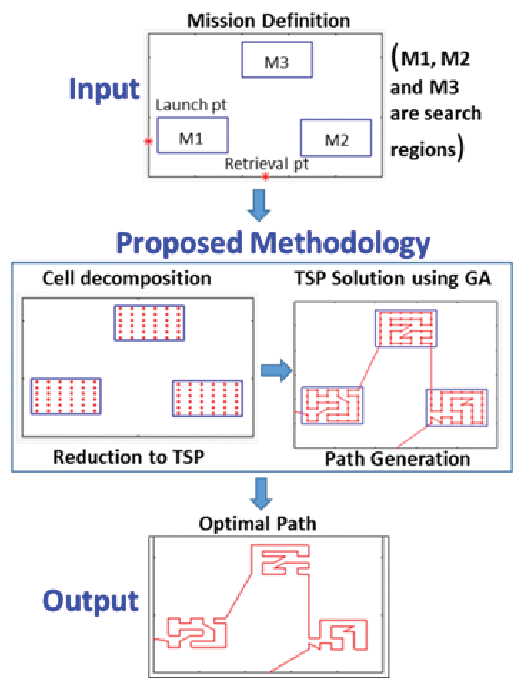

Figure 1. Proposed methodology.

3.1 Decomposition of the Search Regions

The regions are decomposed into uniform cells of size equals to (or less than) the AUV sensor field of view. So, visiting its centroid will ensure coverage of the entire cell. The field of view of the AUV sensor is assumed to be $r\times r$ square meters. Say, each region i has width w_i and height h_i. The total number of cells (k_i) in each region and size (S) of each cell are:

$$k_i = \left\lceil \frac{w_i}{r} \right\rceil \times \left\lceil \frac{h_i}{r} \right\rceil \tag{8}$$

$$S = \frac{w_i}{\left\lceil \frac{w_i}{r} \right\rceil} \times \frac{h_i}{\left\lceil \frac{h_i}{r} \right\rceil} \tag{9}$$

On cell decomposition of the search regions, this problem is reduced to visiting $k = \sum_{1}^{n} k_i$ points with additional launch and retrieval points, i.e., TSP with k+2 cities. Solving this TSP will yield the optimal path for the AUV.

However, TSP is classified as a Non-Deterministic Polynomial-time (NP) problem. To solve these problems, especially with large sizes, Meta-heuristics methods are more suitable than conventional optimisation methods[16]. GA is one of the favourite techniques used for solving TSP[16]. The reduced TSP from TSP-CPP is solved using GA in the proposed methodology.

3.2 Solution based on Genetic Algorithm

To solve the TSP, first each of the points in the problem is labelled. The centroids of the cells obtained from cell decomposition and are labelled as i =2, ..,$k+1$. The launch and retrieval points are labelled as i =1 and k+2 respectively. Then path representation is used to encode the problem as it is the most natural way to represent a path in a TSP problem[18]. Prior to solving the TSP using GA, it is to be noted that there are several operators and parameters in GA. These are listed as follows:

- Population initialisation methodology
- Population size (P)
- Crossover operator
- Probability of crossover (P_c)
- Mutation operator
- Probability of mutation (P_m)
- Fitness function
- Total number of generation (G)

The performance of GA depends on the proper selection of these operators and parameters. The main drawback of GA is that most research applying GA to solve problems do not initially investigate these factors and are usually defined in an ad-hoc fashion[16]. In the proposed methodology, the GA used for solving the TSP is tuned by identifying its optimum settings. The optimum settings are investigated using a DoE on a comparable TSP benchmark instance. The description of the DoE and the method of identification of comparable TSP benchmark instance are explained in the following subsections.

3.2.1 *Design of Experiment for Tuning Genetic Algorithm*

In literature numerous GA operators have been developed for solving the TSP. These operators are shown in Table 1.

All these numerous operators, together with the parameters, each of which can be set at numerous levels, leads to a combinatorial explosion of GA[16]. So, a proper DoE is needed to analyse these combinations. There are few works in the literature that uses DoE for investigating the settings of GA[16-17,24]. The DoE used in the paper is 3^{4-2} fractional factorial experimental (FFE) design, embedded within a full Latin Square[16] (LS).

The FFE design is applied to Initialisation, P/G, $P_{c,}$ and Fitness function. Three levels are considered for each factor to model the relationship as a quadratic. This design results in nine combinations of treatment, denoted by A, B, C, D, E, F, G, H, and I. Each combination is embedded in an LS design. The LS design is used to eliminate two nuisance sources of variability by systematically blocking in two directions. The crossover and mutation operators are two sources of variability in the GA solution and these have been used in the LS design. The random seed is a potential nuisance factor, so to analyse its effects, the experiment is replicated five times with different random seeds. The basis of the selection of this DoE is its efficiency. The main factor effects of GA can be analyse using this DoE by conducting 9×9×5=405 trials. However, the full factorial design would need 3×3×3×9×9×3×5=32,805 trials for the same analysis. (Note: P_m was found to be insignificant on initial cause-effect analysis so it is not considered as a factor in the experiment. A prescribed value of 0.5 is used in the study[16]).

This DoE is used to tune the GA by conducting experiments on TSP bench instance.

Table 1. GA operators

	Crossover operators		
1	One point[17]		*1PX*
2	Two points centre[17]		*2PCX*
3	Two points end[17]		*2PX*
4	Cycling[18]		*CX*
5	Enhanced edge recombination[19]		*EERX*
6	Edge recombination[20]		*ERX*
7	Maximal preservation[21]		*MPX*
8	Position[17]		*PBX*
9	Partially mapped[22]		*PMX*
	Mutation operators		
1	Displacement Mutation[17]		*DM*
2	Two operations adjacent swap[17]		*2OAS*
3	Two operations random swap[17]		*2ORS*
4	Three operations adjacent swap[17]		*3OAS*
5	Three operations random swap[17]		*3ORS*
6	Centre inverse[24]		*CIM*
7	Enhanced two operations random swap[24]		*E2ORS*
8	Inversion[25]		*IM*
9	Shift operation[17]		*SOM*
	Population initialisation		
1	Randomly initialise the population		*R1*
2	Add one sorted individual in the randomised population		*R2*
3	Initialise the population with chromosomes where adjacent genes are very close cities[26]		*R3*
	Fitness functions[16]		
1	$f_i = \sum_{j}^{pop} T_j - t_i$	(10)	*FF1*
2	$f_i = (T_w - T_I) + 1$	(11)	*FF2*
3	$f_i = \left[\frac{1}{(T_I + 1)}\right]$	(12)	*FF3*

where, f_i= fitness value of chromosome i

T_i= the tour distance of the chromosome i

T_w = the worst tour distance in the population.

3.2.2 *Identification of TSP Benchmark Instance*

TSP is one of the most popular combinatorial problems, and several benchmark instances of the problem and its optimal solutions are available online[27]. A comparable benchmark instance for the path planning problem is identified as follows:

The distance (*s*) to be travelled by an AUV to search a rectangular region of size $w \times h$ using the rectangular pattern is given by[4]:

$$S = \min\left(\frac{w}{r}(h-r)+(w-r),\left\lceil\frac{h}{r}\right\rceil(w-r)+(h-r)\right) \quad (13)$$

If the maximum endurance, sensor detection range, number of search areas and average speed are assumed to be one day, 400 m, 5 and 3 knots respectively. Then by eqn-(13) five regions of size 1.5 nM ×1.5 nM (approximately) can be searched within the available time. Cell decomposition of five such regions gives 180 cells i.e. the expected size of the TSP is 180 cities. So, the TSP benchmark instances with the closest number of cities: rat195 and d198 are selected for the study. These two TSP benchmark instances are used as follow:

(a) rat195: for the setting of GA parameters and operator through DoE .i.e. to tune the GA
(b) d198: for comparison of the tuned GA with other GA available in the literature.

The results and discussion are illustrated in the next section.

4. RESULTS AND DISCUSSION

Three experiments are conducted in the study:

- Experiment-A: to tune the GA
- Experiment-B: to compare the performance of the tuned GA with other tuned GA available in the literature
- Experiment-C: to compare the proposed methodology with the conventional methods.

4.1 Experiment-A

The experiment is conducted on the benchmark instance rat195. The DoE and the range of values considered for each factor are shown in Table 2.

Analysis of variance (ANOVA) is used to analyse the results obtained from five replications of the experiment. The hypothesis of interest is H_0: all main factor effects=0 against the alternative hypothesis H_1: At least one main factor effect ≠0 with α=0.01. The ANOVA table for the experimental result is shown in Table 3.

The p-values from Table 3 indicate that the null hypothesis is rejected at 0.01 significance level i.e. all the main factor effects except random (Rn) seed are statistically significant. This analysis shows that all the main factors (except Rn) need to be optimally set to tune the GA.

To identify the optimal setting of the GA, the main effect plot of all statistically significant main factors is drawn, shown in Fig. 2. Those factor levels that yield the lowest path length (marked as red boxes) give the optimal GA settings. Thus, P/G, initialisation, P_c, Crossover operator, Mutation operator and fitness function are set to 50/1000, R3, 0.9, ERX, SOM and FF2 respectively to obtain own tuned GA.

Table 2. DoE for tuning the GA

Embedded one-ninth fractional design

Combine	P/G	Ini	PC	FF
A	100/500	R1	0.1	FF1
B	100/500	R2	0.5	FF2
C	100/500	R3	0.9	FF3
D	50/1000	R1	0.9	FF2
E	50/1000	R2	0.1	FF3
F	50/1000	R3	0.5	FF1
G	500/100	R1	0.5	FF3
H	500/100	R2	0.9	FF1
I	500/100	R3	0.1	FF2

Latin square design

	1PX	2PCX	PX	CX	EERX	ERX	MPX	PBX	PMX
DM	A	I	H	G	F	E	D	C	D
2OAS	B	A	I	H	G	F	E	D	E
2ORS	C	B	A	I	H	G	F	E	F
3OAS	D	C	B	A	I	H	G	F	G
3ORS	E	D	C	B	A	I	H	G	H
CIM	F	E	D	C	B	A	I	H	I
E2ORS	G	F	E	D	C	B	A	I	A
IM	H	G	F	E	D	C	B	A	B
SOM	I	H	G	F	E	D	C	B	A

Table 3. ANOVA table for GA main factor effects

Source	SS	Dof	MS	F	P
P/G	381128004.82	2	190564002.41	74467.14	0.00
Ini	3565954453.33	2	1782977226.67	696738.18	0.00
P_C	8246964.90	2	4123482.45	1611.34	0.00
Cross	217099573.11	2	108549786.56	42418.25	0.00
Mut	7268438.26	8	908554.78	355.04	0.00
Fit	7070709.09	8	883838.64	345.38	0.00
Rn Seed	1266.52	4	316.63	0.12	0.97
ERROR	962197.07	376	2559.03		
Total	4187731607.10	404			

Experiment-B is conducted to compare the performance of this own tuned GA with other tuned GA available in the literature.

4.2 Experiment-B

In literature, various studies exist where GA is tuned to specific problems. Three tuned GA: S_1[16], S_2[29] and S_3[29] are selected for comparison with own tuned GA. The reason being: these GA are tuned considering all the operators and parameter values similar to own tuned GA. The setting of these GA and own tuned GA is shown in Table 4.

These four GA are implemented in MatLab R2010. Comparison of the results (five replications) on solving the TSP benchmark instance d198 are shown in Table 5.

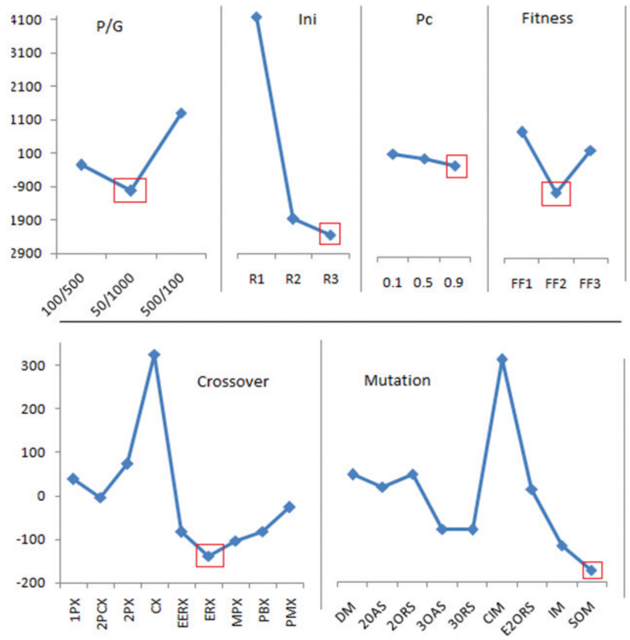

Figure 2. The main effects of GA parameters and operators.

Table 4. GA settings used for the comparison study

Settings	S_1	S_2	S_3	Own
Initialisation	R1	R1	R1	R3
P_c	1	0.3	0.9	0.9
P_m	1	0.18	0.5	0.5
Crossover operator	2PX	EERX	ERX	ERX
Mutation operator	SOM	2OAS	SOM	SOM
Fitness function	FF2	FF2	FF2	FF2

Note: P/G value for S_1, S_2 and S_3 was increased to 50/1000 for uniform comparison

Table 5. Comparison of tuned GA and others available in the literature

Path length: Optimal value=15780			
	Best	**Worst**	**Average**
Own	18842	19527	19272
S_1	45047	48968	46747
S_2	43211	45764	44092
S_3	40125	43773	42223
S_1^G	42331	46327	44531
S_2^G	38551	44538	42833
S_3^G	37741	41553	39958
Computation time (Secs)			
	Best	**Worst**	**Average**
Own	27.04	57.3	38.43
S_1	11.52	33.96	20.09
S_2	17.99	44.24	29.75
S_3	14.67	46.57	29.19
S_1^G	23.21	59.54	39.6
S_2^G	25.23	61.32	41.21
S_3^G	28.46	60.16	43.2

S_1^G, S_2^G, S_3^G are S_1, S_2, S_3 with 1500 generations

Obtained path lengths (best, average and worst) in Table 5 shows that own tuned GA gives the shortest path, nearest to the optimal value compared to S_1, S_2 and S_3. But the computation time of own tuned GA is higher than others. This may be due to the additional processing needed for population initialisation. Since, the path planning procedure is an offline pre-launch activity; the additional computation time will not affect its practical use. However, a different problem arises and demands additional analysis. There is a possibility that in similar computation time, S_1, S_2 and S_3 may generate better solutions. So, solutions were generated with S_1, S_2 and S_3 with 1500 generations (denoted by S_1^G, S_2^G and S_3^G). Table 5 shows that, despite similar computation times, path length obtained using own tuned GA are shorter than S_1^G, S_2^G and S_3^G. However, it should be noted that S_1 and S_2 were tuned for scheduling problems. S3 was tuned for TSP with a smaller number of cities. Each of these GA was tuned to problems of different nature and complexity.

This own tuned GA is used in the proposed methodology and the optimality of the generated path is studied in Experiment-C.

4.3 Experiment-C

A MatLab application was developed for painting multi-region search scenarios. Scenario painting consists of defining different launch and retrieval points and search rectangles of varying sizes. One hundred scenarios were painted in the application, with an average mission time of thirty hours. Path planning for each scenario was done using three methodologies:

i. Conventional deployment: User-defined sequence of visit to the regions and rectangular pattern[28] for region coverage
ii. Task Scheduling[5]: the sequence of visit to the regions computed using TSP on the region centroids and rectangular pattern[28] for region coverage
iii. Proposed methodology: used for path planning

Paths generated in a particular scenario are shown in Fig. 3.

In this particular scenario, the path obtained using the proposed methodology (66.75 NM) is shorter than paths obtained from conventional deployment (84.09 NM) and

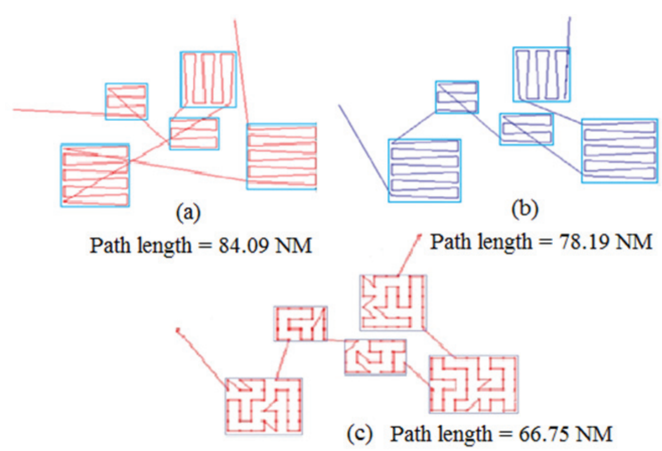

Figure 3. Path obtained by (a) Conventional deployment, (b) Task scheduling, (c) Proposed methodology.

Task scheduling (78.19NM). For overall comparison, a 99% Confidence Interval (CI) is constructed for the lengths of paths generated by the proposed methodology using five replications with different random seeds. The CI and the path lengths obtained using the methodologies for each scenario are plotted in Fig. 4.

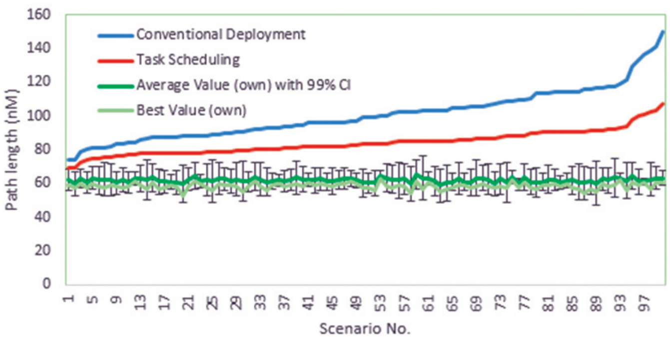

Figure 4. Comparison of the proposed and conventional methodologies.

It is observed that the CI of the path lengths of the proposed methodology is shorter in comparison to both the other methodologies. Hence, the proposed methodology generates optimal paths than the conventional methods. For better appreciation, the comparison can be done in terms of mission completion time. Assuming average AUV speed as 3 knots, the average mission completion time using the proposed methodology is four to six hours shorter than the conventional methods in a total thirty-hour mission. That is the mission completion time is one seventh to one fifth of the total mission time shorter. This saved time can be utilised to plan search on larger mission areas or add another search area to the mission of the same AUV. Hence, the results indicate that the proposed methodology could increase the operational envelope of the AUV.

5. CONCLUSION

Currently Autonomous Underwater Vehicles (AUVs) are widely used for underwater operations/missions. It has to carry out complex operations in hazardous ocean environment before it runs out of battery. So, efficiently conducting operations is critical to mission success. Path planning is used to optimise path in AUV operations. Coverage path planning (CPP) is used to generate optimal paths to cover a given area. Although CPP is extensively studied, CPP for single AUV searching multiple areas has not received any attention. In this paper, a Genetic Algorithm (GA) based methodology for multi-region search mission CPP for single AUV is proposed. Simulation study results indicate that in an average thirty hour mission, mission time using the proposed methodology is four to six hours shorter than conventional methods. This saved time can be utilised to plan search on larger mission areas or add another search area to the mission of the same AUV. Hence, the proposed methodology could increase the operational envelope of the AUV. In future work, the proposed methodology would be extended to enhance its benefits for real-world applications. The extension would consider the factors like presence of obstacles in the area of operation, and underwater current. This could be explored by modelling the ocean environment extracting the obstacle data from Electronic Navigational Charts and using the past-cast, now-cast or fore-cast undercurrent data. The proposed methodology would be used over the constructed ocean model. Future work will also focus on improving the efficiency of the proposed methodology by using parallelisation method.

REFERENCES

1. Mahmoudzadeh, S.; Powers, D.; Sammut, K. A. Lammas, A. & Yazdani, M. Optimal Route Planning with Prioritized Task Scheduling for AUV Missions. *In* 2015 IEEE International Symposium on Robotics and Intelligent Sensors, IEEE, Langkawi, Malasia, 2015. doi:10.1109/IRIS.2015.7451578
2. Hagen, P. E.; Midtgaard O. & Oistein H. Making AUVs Truly Autonomous. *In* OCEANS 2007, IEEE, Vancouver, Canada, 2007. doi: 10.1109/OCEANS.2007.4449405
3. Kruger, D.; Rustam, S.; Blum, A. & Briganti, J. Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments. *In* proceeding 2007 IEEE International Conference on Robotics and Automation, IEEE, Roma, Italy, 2007. doi: 10.1109/ROBOT.2007.364135.
4. Junfei, X.; Luis, R.; Garcia, C. & Lei, J. An Integrated Traveling Salesman and Coverage Path Planning Problem for Unmanned Aircraft Systems. *IEEE Control Systems Letters*, 2018, **3**(1), 67-72. doi:10.1109/LCSYS.2018.2851661
5. Yukiyasu, N. & Toshihiro, M. Path Planning Method based on Artificial Potential Field and Reinforcement Learning for Intervention AUVs. *In* 2019 IEEE underwater technology, IEEE, Kaohsiung, Taiwan, 2019. doi:10.1109/UT.2019.8734314.
6. Bartolome, G. & Alvarez, A. G. Path Planning of Autonomous Underwater Vehicles in Current Fields with Complex Spatial Variability: an A* Approach. *In* Proceedings of 2005 IEEE International Conference on Robotics and Automation, IEEE, Barcelona, Spain, 2005. doi: 10.11.9/ROBOT.2005.1570118
7. Jian, C.; Ye, L.; Shiqi, Z & Xiaosheng, B. Genetic-Algorithm-based Global Path Planning for AUV. *In* 2016 9th international symposium on computational intelligence and Design. IEEE, Hangzhou, China, 2016. doi.10.1109/ISCID.2016.2027.
8. Kantapon, T.; Wilsony, P. A.; Turnockz, S. R. & Alexander, B. P. Grid-Based GA Path Planning with Improved Cost Function for an Over-actuated Hover-Capable AUV. *In* 2014 IEEE/OES Autonomous Underwater Vehicles (AUV), Oxford, USA, 2014. doi:10.1109/AUV.2014.7054426
9. Ge, Y. & Rubo, Z. Path Planning of AUV in Turbulent Ocean Environments Used Adapted Inertia weight PSO. *In* 2009 5th international conference on Naural Computation, IEEE, Tianjian, China, 2009. doi:10.1109/ICNC.2009.355.

10. Shipeng, I.; Yakun, Z. A multi-AUV searching algorithm based on neuron network with obstacle. *In* 3rd international symposium on Autonomous systems, IEEE, Shangai, China, 2019 doi:10.1109/ISASS.2019.87577
11. Wenyu, C.; Meiyan, Z.; & Yahong, R. Task Assignment and Path Planning for Multiple Autonomous Underwater Vehicles Using 3D Dubins Curves. *Sensors*, 2017, **17**(7), 1607. doi:10.3390/s17071607
12. Mingzhong, Y.; Daqi, Z. & Yang, X. Complete Coverage Path Planning in an Unknown Underwater Environment Based on D-S Data Fusion Real-Time Map Building. *Int. J. Distributed Sens. Netw.*. 2012, **8**(10). doi:10.1155/2012/567959
13. Cheng, F. & Anstee, S. Coverage Path Planning for Harbour Seabed Surveys using an Autonomous Underwater Vehicle. *In* OCEANS'10 IEEE SYDNEY, Sydney, *IEEE,* 2010. doi:10.1109OCEANSSYD.2010.5603591
14. Seokhoon, Y.; & Chunming, Q. Cooperative Search and Survey Using Autonomous Underwater Vehicles (AUVs). *IEEE Trans. parallel Dist. Sys.,* 2010, **22(**3), 364-369. doi:10.1109/TPDS.2010.88
15. Williams, D. P. On Optimal AUV Track-Spacing for Underwater Mine Detection. *In* 2010 IEEE International Conference on Robotics and Automation. IEEE, Anchorage, USA, 2010, doi:10.1109/ROBOT.2010.5509435
16. Pongcharoen, P.; Warattapop, C. & Thapatsuwan, P. Exploration of Genetic Parameters and Operators through Travelling Salesman Problem. Pitsanulok, Thailand, 2007. doi: 10.2306/scienceasia1513.1874.2007.33.215
17. Murata, T. & Ishibuchi, H. Performance evaluation of genetic algorithms for flowshop scheduling problems. *In* Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE, Orlando, USA, 1994. doi:10.1109/ICEC.1994.349951
18. Oliver, I. M.; Smith, C. J. & Holland, J. R. A study of permutation crossovers on the travelling salesman problem. *In* Proceedings of the Second International Conference on Genetic Algorithms and their Applications, Cambridge Massachusetts, USA, 1987.
19. Starkweather, T.; McDaniel, S.; Mathias, K.; Whitley, D. & Whitley, C. A comparison of genetic sequencing operators. Proceedings of the Third International Conference on Genetic Algorithms, San Diego, CA, USA, 1991.
20. Whitley, D.; Starkweather, T. & Fuquay, D. Scheduling problems and the travelling salesman: the genetic edge recombination operator. *In* Proceedings of the Third International Conference on Genetic Algorithms, George Mason University, USA, 1989.
21. Muhlenbein, H.; Gorgesschleuter, M. & Kramer, O. Evolution algorithms in combinatorial optimisation. *Parallel Computing*, 1992, **7**(1), 65-85. doi:10.1016/0167-8191(88)90098-1.
22. Goldberg, D. E. & Lingle, R. Alleles, loci and the travel travelling salesman problem. *In* Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Pittsburgh, USA, 1985.
23. Chainate, W. A Hybridization of Genetic Algorithm and Neural Network for Solving the Travelling Salesman Problem. Faculty of Science, Naresuan University, Phitsanulok, 2005.
24. Tralle, D. Analysing of Genetic Operations in Genetic Algorithms Applied to Optimization of Manufacturing Systems. University of Newcastle, Newcastle, UK, 2000.
25. Goldberg, D. E. Genetic Algorithms in Search, Optimisation and Machine Learning. Addison-Wesley, Massachusetts, 1989, 372p.
26. Dao, S. D,; Kazem, A, & Marian, R. An effective genetic algorithm for large-scale travelling salesman problems, *In* Proceedings of the world congress on engineering and computer science, San Franscisco,USA, 2016.
27. www.iwr.uniheidelberg.de/iwr/comopt/soft/ TSPLIB95/ TSPLIB.html (Accessed on 3 February 2020).
28. Wagner DH, Mylander WC and Sanders TJ. Naval Operations Analysis, 3rd ed. Annapolis, Naval Institute Press, 1999, 421p.
29. Murata, T.; Ishibuchi, H. & Tanaka, H. Genetic algorithms for flowshop scheduling problems. *Computers and Industrial Eng.,* 1996, **30**(4), 1061-1071. doi:10.1016/03608352(96)00053-8.

ACKNOWLEDGEMENTS

The author would like to thank Mr S.B. Taneja, Director Institute for Systems Studies and Analyses, DRDO, for his encouragement and support in this work. The author would also like to thank Dr Vijay Rao and Ms Deepakshi Shah for their kind suggestions and guidance. The author would also thank Shri. Suneel Yadav for his support.

CONTRIBUTOR

Mr Shristi Deva Sinha obtained his MSc in Mathematics from the University of North Bengal, Darjeeling, India. He joined Institute for Systems Studies & Analyses, Delhi, in 2007 as a Scientist and has been working in the area of analysis of naval weapon systems and procedures through mathematical modelling and simulation. His areas of interest include naval systems analysis, abstract algebra, finite field theory, and cryptography.