On Underwater Vehicle Routing Problem

Wojciech Bożejko(✉), Szymon Jagiełło, Michał Lower, and Czesław Smutnicki

Faculty of Electronics, Wrocław University of Technology,
Janiszewskiego 11-17, 50-372 Wrocław, Poland
{wojciech.bozejko,szymon.jagiello,michal.lower,
czeslaw.smutnicki}@pwr.edu.pl

Abstract. In the paper we consider a problem of underwater vehicle routing (also called path planning) which can be described as follows: Given positions under water have to be inspected by the mobile robot taking under consideration water currents, minimizing the total time of the inspection. Two elements of such an approach are described here: vehicle positioning method and routing algorithm. We proposed to apply tabu search and dynasearch metaheuristics as routing alogrithms. As efficient metaheuristics, proposed algorithms allows us to determine good solutions (paths) in a very short time.

1 Introduction

Underwater mobile robot which is considered in this paper is small and its dimensions are in the range of the cube $50 \times 50 \times 70$ cm. The device has a low positive buoyancy close to zero and it is equipped with three electric drive motors with the rotors. The two drive motors are used to change the direction of movement of the robot in the horizontal plane, whereas the third motor is used to immerse the robot and to change its vertical position. A permanent connection via a power cable is used to supply energy to electric motors. This means that the maximum range of the robot is associated with the position of the floating base station and the length of the connecting cable. Hence it is assumed that the maximum range of the robot is about 300 m from the base station. The base station is located on the surface of the water. Therefore, determination of its position is not a big problem using e.g. measurements by GPS (Global Positioning System) or GNSS (Global Navigation Satellite Systems). However, the satellite navigation is impossible under water.

Recognizing the problem as a vehicle routing one can observe, that mostly metaheuristics are used for its solving nowadays. Tabu Search (TS) metaheuritics was introduced by Glover [6] as an extension of classical local search methods. It explores the solution space by local search procedure with the use of neighborhoods. Most of TS efficient implementations are based on the multi-start model, a neighborhood decomposition [3] or move acceleration [4]. Congram et al. proposed so-called dynasearch neighborhood [5] based on the idea of exploration of the exponential-size neighborhood in the polynomial time. In the paper we propose to applyt the tabu search metaheuristics and tabu search with backtracking-jump as well as dynasearch to determine underwater vehicle routing path.

© Springer International Publishing Switzerland 2015
R. Moreno-Díaz et al. (Eds.): EUROCAST 2015, LNCS 9520, pp. 861–868, 2015.
DOI: 10.1007/978-3-319-27340-2_105

wojciech.bozejko@pwr.edu.pl

2 Positioning System

Due to the limited technical capabilities to determine the position of the object underwater a dead reckoning navigation, assisted with the inertial navigation system (INS) and the inertial measurement units (IMU), are often used [8]. Due to the fact that our task refers to an object moving in a short distance and in regular contact with the base station the most preferred method is the short baseline acoustic positioning system (SBL) [7,10]. The measurement of the position of the mobile robot is performed with respect to the position of the base station which position is determined from satellite navigation. It is assumed that the mobile robot moves under water at a speed up to 1 m/s. The measurement of the position of the object in the real aquatic environment must take the characteristics of the environment into account. It turns out that the most effective measuring medium in these conditions are ultrasounds. With the use of ultrasound one can determine the distance in the water, which is proportional to the distance travelled, what is more, one can also determine the speed of the object moving away, which is possible with the use of Doppler effect. The sound propagates in the water at a speed of 1490 m/s which is four times faster than in the air. Thus, the distance measurement by means of ultrasounds at the section of 300 m can be performed in 0.2 s. Assuming a maximum speed of movement of the robot at 1 m/s, it is possible to obtain the position update every 20 cm.

When making distance measurement on the basis of the average of the ultrasonic wave movement in two directions (both: from object to the reference point and from the reference point to the object) it is possible to eliminate data errors resulting from the water movement in which the measurements are taken.

2.1 Underwater Position Determination

Determination of underwater location of object in 3D space by measuring the distance from the points of reference requires designation of appropriate location of the points [2,7,10]. In determination of the robot location under water only by distance measuring, at least three such sensors are necessary. Distribution of reference points has a significant impact on the accuracy of the designated position of the underwater object. The line connecting two basis points is called the baseline (BL). Most preferred is an arrangement in which the lines from the reference points to the position of the underwater object intersect at a right angle. This means that the spacing between the reference points (BL length) should be close to twice the distance between the positioned object and the BL line. On one hand, a practical relationship can be observed here: the greater the immersion of the positioned object or the robot's work area, the greater the BL spacing between reference points. On the other hand, the specificity of the activities carried out by underwater robot determines the required measurement accuracy. Thus, the measurement system used is a compromise between technical capabilities and required expectations regarding the accuracy of the measurement. For large depths exceeding 100 m and for objects working in a large measure space there are such systems as (LBL) Long BaseLine type. In such systems BL has a

wojciech.bozejko@pwr.edu.pl

length of more than 100 m, sometimes reaching several kilometres. For distances of up to several kilometres there is a problem with the enlargement of absolute error of ultrasonic distance measurement. To increase the absolute accuracy of position measuring of the entire work space in LBL system there are often more than three points of reference fixed. In some publications, e.g. [2], LBL system is further narrowed as a measurement system based on the reference points located on the bottom of the tank. If the reference points are located on the surface of the water and their position is corrected with GPS bearing, then this type of system is called GPS intelligent buoys (GIB) system. If the required BL length does not exceed 100 m, measuring systems of this type are called Short baseline (SBL) acoustic positioning systems. The measurement principle is the same as in LBL systems. In SBL systems reference points are often placed on the hull of a ship (vessel) leading or supervising underwater research. Such placement of reference points provides a stable measurement results relative to the vessel. Rigid arrangement between the reference points is preserved, while the position of the vessel can be determined on the basis of GPS data.

Due to the fact that the measurements of the position of the underwater object are often monitored involving small vessels with limited technical possibilities related to the spacing of reference points there are (USBL) Ultra Short BaseLine type measuring systems used. In such solutions BL is small, no more than several meters and, at the same time, many times smaller than the measured distance. In such systems, there is a problem with accurate position measure based only on the distance to the reference points. Such a situation can be seen for instance when working depth of underwater robot is several times greater than the possibility of BL spacing. Therefore, the type of USBL solution is combined with other measurement systems. Location coordinates are combined with dead reckoning navigation, e.g. on the basis of determining the speed and direction of the robot movement from IMU module, based on water pressure there is the depth of immersion defined. By setting the propagation direction of the sound wave it is possible to reduce such system to one reference point located on the vessel which supervisors measurements.

2.2 Design of Measurement System

The task refers to an object that is in constant contact with the base station with the use of a cable connection. The measured object moves approximately 150 m from the vessel which supervises measurements. In the large water area such as in the sea the supervising vessel may have a length of a several dozen meters, which means that in this case the system can be classified as SBL. When using data collected from small vessels - measuring system can be categorized as USBL [2,7,10].

On the object, that is on the underwater robot there is placed a transponder emitting ultrasonic signal, while in reference points there are receivers of this signal. The underwater mobile robot will be equipped with sensors of INS and IMU type, e.g. namely in a magnetometer and a pressure sensor. On the basis of pressure measurement the depth of immersion can be determined. If the object

wojciech.bozejko@pwr.edu.pl

and reference points are in motion, direction of the object movement is known and we have an influence on the position of basis points, then, not only on the basis of the depth of immersion but also two measurements of the distance from the base station the location of an underwater object can be determined. If the base station is a vessel of a length of several meters, then ultrasonic signal receivers (reference points for measuring distances) can be placed on the bow and stern of the ship. Ultrasonic transponder is placed on the mobile robot. If on a small object reference points are located on the stern and on the bow of the vessel, a satisfactory accuracy of the measurements can be obtained when the work distance of the object is not more than twice the length of the vessel.

3 Routing Algorithm

The problem considered can be modelled as a variation of Traveling Salesman Problem (TSP) in which topology positions are modelled as a graph with weighted arcs. Weights of arcs are connected with times of travel between positions (vertexes of the graph), which are combinations of distances and influences of water currents. For the considered problem a tabu search and dynasearch metaheuristics are proposed.

Table 1. Percentage relative deviation for time of calculations 0.01 [s].

	%diff		
problem	TS	TSAB	DS
gr202	16.7	16.2	12.9
a280	22.4	22.4	21.2
pr299	24.3	24.3	21.8
bier127	6.5	6.5	5.7
pr1002	27.8	27.8	26.9
pr107	4.6	3.8	1.7
eil76	13.8	13.8	13.8
att532	28.3	28.3	26.4
gil262	30.6	30.6	19.5

	%diff		
problem	TS	TSAB	DS
berlin52	14.9	14.9	11.1
eil101	16.1	16.4	15.7
eil51	12.0	12.0	10.6
pr136	23.8	24.3	23.8
burma14	8.3	3.9	9.4
ch130	21.3	21.3	21.1
lin105	37.5	37.5	36.5
ch150	24.6	24.6	24.2
gr96	12.0	16.4	10.7

	%diff		
problem	TS	TSAB	DS
gr137	39.3	39.3	30.8
pr124	15.1	14.2	13.2
att48	13.5	13.5	5.7
ali535	29.4	29.4	25.0
gr229	25.4	25.4	21.0
pr439	22.4	22.4	19.7
pr264	17.8	17.8	14.4
gr431	29.2	29.2	24.8
gr666	23.6	23.6	21.8

	%diff		
problem	TS	TSAB	DS
Average	20.8	20.7	18.1

3.1 Tabu Search

Tabu search algorithm was proposed by Fred Glover in 1986 [6]. The idea lies in metaheuristics designed in a way so as to take local optimization methods beyond local optima. Initially, there is a move defined, which is an operation that transforms the solution S into the new solution S'. The set of all elements

wojciech.bozejko@pwr.edu.pl

Table 2. Percentage relative deviation for time of calculations 0.1 [s].

	%diff				%diff				%diff		
problem	TS	TSAB	DS	problem	TS	TSAB	DS	problem	TS	TSAB	DS
gr202	13.0	12.9	12.9	berlin52	14.9	13.5	6.6	gr137	38.8	38.8	20.4
a280	21.2	21.2	21.2	eil101	14.6	14.3	10.0	pr124	13.2	13.2	9.8
pr299	22.6	22.5	21.8	eil51	12.0	11.7	3.8	att48	13.5	13.5	3.5
bier127	5.6	5.6	5.7	pr136	20.8	20.8	23.8	ali535	27.2	27.2	23.7
pr1002	27.8	27.8	24.8	burma14	8.3	3.9	9.4	gr229	22.1	21.8	21.0
pr107	1.7	1.7	1.7	ch130	21.1	21.1	21.1	pr439	22.1	21.8	19.7
eil76	13.8	13.8	11.0	lin105	35.9	35.9	26.0	pr264	15.1	15.5	14.4
att532	28.1	28.1	25.0	ch150	24.2	24.2	24.2	gr431	28.2	28.2	24.3
gil262	23.5	23.2	19.5	gr96	10.7	10.3	10.7	gr666	23.6	23.6	21.2

	%diff		
problem	TS	TSAB	DS
Average	19.4	19.1	16.2

that can be generated from a given solution S by means of a defined move is called the neighborhood $N(S)$ of the given solution.

The primary TS mechanism consists of iterative search of neighborhood $N(S)$. The element from neighbourhood $S' \in N(S)$, for which the value of the cost function $\Theta[S']$ is the best (usually the smallest) becomes the new solution. In order to avoid getting stuck in a local optima there is an additional structure called tabu list T introduced in the algorithm, which in its simplest form contains prohibited solutions or moves. Elements from the list cannot be selected by a static (pre-set) or dynamic (variable during operations) number of iterations. The length of the list directly affects the speed of the method and its ability to exit local minima. Too short list will quickly cause 'getting stuck', whereas too long will adversely affect the speed of the algorithm. In addition, the use of too long list does not guarantee a route out of the current minimum [1].

Typically, the algorithm stops after having executed a preset number of iterations, exceeded maximum runtime, finding a solution that is sufficiently close to the lower estimate or failure to improve after the specified number of iterations or specific time. It should be noted that each TS algorithm implementation is dependent on a given problem.

3.2 Swap-Type Neighborhood

Checking the neighborhood $N(S)$ of *swap*-type consists of generating of all solutions S', which can be obtained by changing the values of the two positions p_1 and p_2 of solution S and designating for them values of the function $\Theta[S']$. The size of the neighborhood without repetitions (a number of possible generated solutions is)

$$|N| = \frac{|S|^2 - |S|}{2}, \tag{1}$$

wojciech.bozejko@pwr.edu.pl

Table 3. Percentage relative deviation for time of calculations 1 [s].

problem	%diff TS	TSAB	DS
gr202	12.9	12.9	12.9
a280	20.8	20.8	21.2
pr299	21.8	21.8	21.8
bier127	5.6	5.6	5.7
pr1002	27.5	27.5	23.9
pr107	1.7	1.7	1.7
eil76	13.8	13.8	9.7
att532	26.8	26.8	25.0
gil262	18.4	18.4	19.5

problem	%diff TS	TSAB	DS
berlin52	14.9	13.5	4.9
eil101	14.6	14.3	14.6
eil51	12.0	11.7	1.4
pr136	20.8	16.1	21.1
burma14	8.3	3.9	9.4
ch130	21.1	21.1	20.5
lin105	35.9	34.8	17.9
ch150	24.2	24.2	24.2
gr96	10.7	9.2	10.7

problem	%diff TS	TSAB	DS
gr137	38.8	35.3	20.9
pr124	13.2	13.2	13.2
att48	13.5	13.5	1.6
ali535	25.3	25.3	23.7
gr229	21.0	20.6	21.0
pr439	19.8	19.8	19.7
pr264	13.7	13.7	14.4
gr431	25.1	25.1	24.3
gr666	22.3	22.3	21.2

problem	%diff TS	TSAB	DS
Average	18.7	18.0	15.8

thus, the computational complexity of generating of all the solutions S' without designation the values of given functions is $O(n^2)$.

3.3 Tabu Search Algorithm with Back Jump Tacking

Use of the best solutions from the given neighbourhood as the start solutions results in the loss of history of previous searches. In case of 'being stuck' in the local minimum, the only thing left is to choose a new solution starting with an empty tabu list. In 1996, E. Nowicki and C. Smutnicki proposed a modification to the basic version, which enables the multiple use of neighborhoods of the selected solutions [9].

During its operation, the proposed method uses an additional tabu list L. Each time when the algorithm encounters a solution better than the previous one, it saves the current solution and T list on L list. The solution chosen in the next iteration is added to the saved T list. The elements on L list are sorted by cost function value $\Theta[S]$ [12].

In case of 'being stuck' in the local minimum, the algorithm returns to the best solution from L list. The return concerns also T list [9]. This approach ensures that the next chosen solution will be different than previously selected. In addition, in such an implementation the new solution is added to the stored T list, what allows for multiple returns, whereas the number of possible returns is equal to the length of T list minus one. After having used this limit, the solution is removed from L list. It is possible to compare the basic TS algorithm to exploration of the segment of the solution space, whereas TSAB - to the procedure of its exploration.

wojciech.bozejko@pwr.edu.pl

3.4 Dynasearch

Descent and iterative improvement algorithms are a simple and fast optimization method. The algorithm searches a better solution in the neighbourhood $N(S)$ of a given solution S. If it exists it becomes the current solution and the procedure is repeated. If there is no better solution then the algorithm terminates. An effective approach is to create a new starting solution by performing a permutation procedure on the final one and restarting the algorithm.

The quality of the obtained result depends heavily on the neighbourhood. A small one will be checked faster but will cause to terminate quickly. A big one will give better results but the computational effort will be higher. The main idea of the DS algorithm is to provide an efficient way of checking neighbourhoods which are obtained by performing series of moves instead a single one and which size is exponential [5]. The neighbourhood is explored in polynomial time by using dynamic programming techniques.

The dynasearch algorithm finds the best solution which can be obtained by using a combination of independent moves. The size of such neighbourhood for the swap move is $2^{n-1} - 1$ but the result can be obtained in $O(n^2)$ time.

4 Computational Experiments

In order to verify the proposed algorithms there were tests carried out on selected instances taken from TSPLIB [11]. The study was performed on a machine equipped with Inter i7 X980 processor, the operating system with the kernel 3.2.0.70-generic or gcc 4.6.3 compiler. During its work, the device may encounter unforeseen obstacles that prevent it to visit all the required points on the preplanned route. Such situation will require designation of a new route, which takes into account the above problem. All the computations will have to be made in very limited time, so the emphasis is placed on the quality of the solutions obtained in specific units of time: 0.01[s], 0.1[s], 1[s]. Percentage deviation between the obtained result and the best known result for each problem and average percentage deviation was calculated. The obtained results are shown in Tables 1, 2, 3.

The environment in which the device is working is diverse and vulnerable to unpredicted changes. In this situation we can not use an algorithm which gives even the optimal solution for a single problem but fails at other instances. Therefore the current research stage was focused on selecting the most universal method. The proposed procedures were tested by using 27 problem instances. The results confirmed the superiority of the TSAB method over the TS method. A very fast exploration of a large neighbourhood caused that the DS algorithm turned out to be the best choice. In this situation it will be used in further development of the project.

5 Conclusions

The paper presents tabu search and dynasearch algorithms for the underwater vehicle routing problem. Tests were carried out on literature test data taken

wojciech.bozejko@pwr.edu.pl

from the TSPLIB [11]. Computational experiments shows, that dynasearch algorithm outperforms other approaches in the matter of solutions quality as well as running time, in all time ranges (0.01, 0.1, 1 s) which makes it a very good algorithmic solution for vehicle route determination.

References

1. Abdullah, S., Ahmadi, S., Burke, E.K., Dror, M., McCollum, B.: A tabu based large neighbourhood search methodology for the capacitated examination timetabling problem. J. Oper. Res. **58**(11), 1494–1502 (2006)
2. Alcocer A., Oliveira P., Pascoal, A.: Underwater acoustic positioning systems based on buoys with GPS. In: Proceedings of the Eighth European Conference on Underwater Acoustics, 8th ECUA, Carvoeiro, 12–15 June 2006
3. Bożejko, W., Pempera, J., Smutnicki, C.: Parallel tabu search algorithm for the hybrid flow shop problem. Comput. Ind. Engi. **65**, 466–474 (2013)
4. Bożejko, W., Uchroński, M., Wodecki, M.: Parallel hybrid metaheuristics for the flexible job shop problem. Comput. Ind. Engi. **59**, 323–333 (2010)
5. Congram, R.K., Potts, C.N., Van de Velde, S.L.: An iterated dynasearch algorithm for the single maschine total weighted tardiness scheduling problem. INFORMS J. Comput. **14**(1), 52–67 (2002)
6. Glover, F.: Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. **1**(3), 533–549 (1986)
7. Hegrenaes, O., Berglund, E., Hallingstad, O.: Model-aided inertial navigation for underwater vehicles. In: IEEE International Conference on Robotics and Automation, pp. 1069–1076 (2008)
8. Kaniewski, P.: Integrated positioning system for Auv. Mol. Quantum Acoust. **26**, 115–128 (2005)
9. Nowicki, E., Smutnicki, C.: A fast taboo search algorithm for the job shop problem. Manage. Sci. **42**(6), 797–813 (1996)
10. Stutters, L., et al.: Navigation technologies for autonomous underwater vehicles. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. **38**(4), 581–589 (2008)
11. TSPLIB: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/
12. Watson, J.P., Howe, A.E., Whitley, L.D.: Deconstructing Nowicki and Smutnicki's i-TSAB tabu search algorithm for the job-shop scheduling problem. Comput. Oper. Res. **33**, 2623–2644 (2005)

wojciech.bozejko@pwr.edu.pl