

Comparison of Intelligent Water Drops and Backtrack Algorithm in Solving Travelling Salesman Problem

Indra Maryati*, Gunawan**

* Department of Information Technology, Sekolah Tinggi Teknik Surabaya

** Department of Electrical Engineering, Faculty of Industrial Technology, Institut Teknologi Sepuluh Nopember

Keywords:

Intelligent Water Drops
Meta-heuristic Algorithm
Swarm Intelligent Algorithm
Optimization Algorithm
Travelling Salesman Problem

ABSTRACT

Travelling Salesman Problem (TSP) is a classical problem to find the best and minimum solution for a salesman to travel all of the city and return to the start point of his journey. With the problem, there are several ways to get the best solution using different algorithms. Backtrack algorithm is one of algorithm that can be used to find a solution in TSP. Backtrack algorithm has weakness in processing time/duration, whether the solution is always optimal. The new algorithm, called Intelligent Water Drops (IWD), is one of meta-heuristic algorithm that inspired from the motion of water flow in a river. IWD also an optimization algorithm, this algorithm will search for the optimal path or solution for the problem. IWD is chosen because it has better performance in time/duration rather than Backtrack algorithm in finding the solution. In this paper will explain how the workings of IWD and compare with Backtrack to clarify the algorithm. This algorithm shows a magnificent result to solve TSP with minimum cost and time.

*Copyright © 2013 Information Systems International Conference.
All rights reserved.*

Corresponding Author:

Indra Maryati,
Departement of Information Technology
Sekolah Tinggi Teknik Surabaya,
Jalan Ngagel Jaya Tengah 73-77, Surabaya, Indonesia.
Email: maryati@stts.edu

1. INTRODUCTION

Travelling Salesman Problem, called TSP, can be defined as the problem of determining a route of traveling salesman to visit each city exactly once and returns to the origin city, with the assumption that all distance or cost of travel between the cities is known, and each city have to be visited only once in the salesman' route. The purpose of modeling this problem is to facilitate the optimal route search, in this case is minimal in terms of distance or travel cost of the tour, which visited all the cities and returns to the origin city.

TSP was first formulated as a mathematical problem in 1930 and has since become a very popular optimization problem. But the real problem has existed since the 19th century. Reference to this problem has been around since 1832 in which a guide book for salesmen, have been published in Germany. This book contains five important that visiting these cities around Germany and Switzerland. These routes can be used by salesmen to get around efficiently in dozens of cities.

TSP is often used in testing the effectiveness of an algorithm for optimization. However, each algorithm has its advantages and disadvantages like Backtrack algorithm has its disadvantages in time/duration to find a solution. Algorithms that used to search for optimal solutions inspired by the natural state belonging to the meta-heuristic algorithms. The most commonly used algorithms for optimization problem is Genetic algorithm, however, in this paper will use other meta-heuristic algorithms, namely Intelligent Water Drops (IWD). IWD is an algorithm that is inspired by the movement of water in the river [1].

As the IWD is one of meta-heuristic and swarm intelligence algorithms, it can find the solution faster than Backtrack algorithm but IWD may not find the optimal result in like Backtrack algorithm if it implements in different cases or problems. In this research will implement IWD algorithm in solving TSP and also it can prove that IWD algorithm can used as one of algorithm to find solution faster than Backtrack

algorithm. With IWD, there will be more variance of meta-heuristic algorithms that can be used by students to search solutions besides Genetic algorithm.

2. LITERATURE REVIEW

This research is based on some other literature that proposed by other researcher. Backtrack algorithm, one of heuristic algorithms, is the algorithm in finding optimal solution. As the development of research method and others factors, meta-heuristics algorithm was introduced to improve heuristics algorithm. IWD as one of meta-heuristics algorithm and Swarm Intelligence algorithms that based on the state of nature of moving water in the river.

2.1. Natural Water Drops

IWD algorithm is one of the Swarm Intelligence algorithms, where the algorithm is based on the state of nature of the movement of water in the river. Moving water on a river comes from a source and will be heading to a destination. Water traveling from the source to the destination on the river is not always smooth or straight but has turns. Thus the river has multiple lines, where each line has a beginning and an end. On the riverbed which the water flow through is also contains soil, where the amount of soil on the riverbed will affect the selection of the path to be chosen by the water. Usually the water will tend to choose the path of the river that has less amount of soil.

If the water flow through a river, soil on the riverbed will come up along with the flowing water. So that the flowing water in a river contains a number of soil. Everything that moves must have speed. Likewise, the moving water, the water velocity will affect the amount of soil taken on the riverbed. River with less soil will cause the water velocity increases, instead of river with more soil will cause the water velocity decreases. The water velocity will also affect the amount of soil taken in from the riverbed. Water that has more velocity will collect more soil on the riverbed, instead of water that has less velocity will only collect less soil.

From the observation of natural water in the river, the flowing water has 3 properties, i.e. the amount of soil that it brings, the velocity of the water itself, and the city that has been visited. River has only one property, i.e. the amount of soil on the riverbed. If the water has reached the goal, it will be selected the fastest path to reach the goal. The fastest path is the path that has the shortest distance traversed by water.

2.2. Intelligent Water Drops

The IWDs algorithm is based on the natural water drops in the rivers as explain before. Flowing waters will denote as intelligent water drops in this algorithm. Each of the intelligent water drops will have amount of soil which is denoted as $SOIL_{IWD}$, velocity which is denoted as VEL_{IWD} , and the visited city which is denoted as VC_{IWD} . Path in a river also contains some soil in the riverbed, this soil denoted as $SOIL[i,j]$ where i is start of a path and j is the end of a path.

IWD algorithm need graph as the input problem, this means the problem have to represent in graph. In a graph, city will represent as the vertices while path represent as the edges. Amount of soil ($SOIL[i,j]$) in a path is initialize with initial value that has been given by user. Amount of soil ($SOIL_{IWD}$) and velocity (VEL_{IWD}) in each IWD also will initialize by user. User also will initialize the number of IWD that will use in the algorithm, which is usually same with the number of city in the problem.

Iteration in IWD algorithm is defined by user, which in every iteration each IWD will have a tour to visit all of the city (each IWD will have their own iteration). Below is the IWD algorithm that specified by Hamed Shah-Hosseini [3] and the block diagram for TSP.

1. Initialization of static parameters.
2. Initialization of dynamic parameters.
3. Spread the IWDs randomly on the nodes of the graph as their first visited nodes.
4. Add the node just visited to visited node list of each IWD.
5. Repeat steps 5.1 to 5.4 for those IWD with partial solution.
 - 5.1 For the IWD residing in node i , choose the next node j , which does not violate any constraints of the problem and is not in the visited node list of the IWD.
 - 5.2 For each IWD moving from node i to node j , update its velocity.
 - 5.3 Compute the soil.
 - 5.4 Update the path soil and IWD soil.
6. Find the current iteration-best solution.
7. Update soils on the path of current iteration-best solution.
8. Update the total-best solution by the current iteration-best solution if the current iteration-best solution is better.
9. Increment the iteration counter.
10. Stops with total-best solution

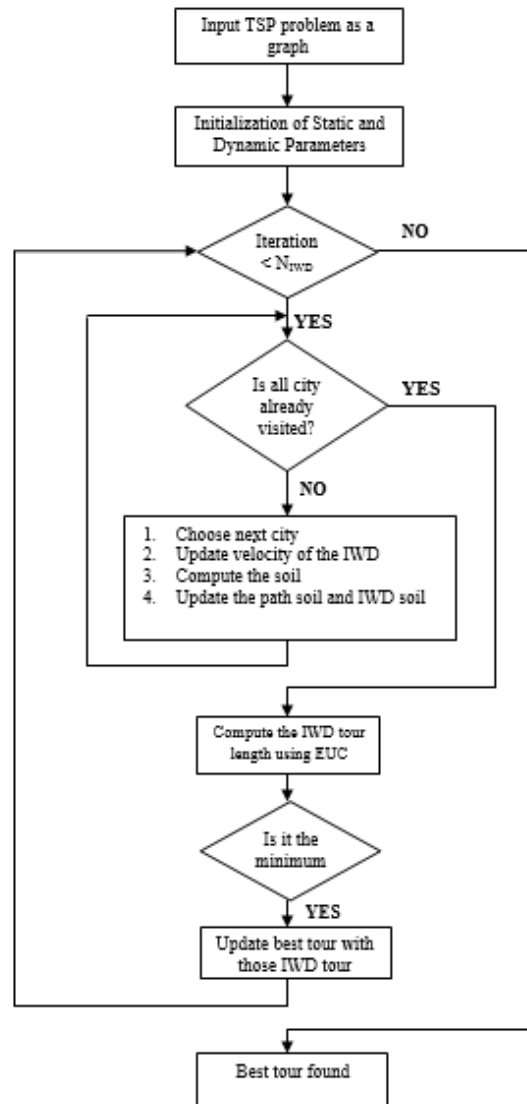


Figure 1. Block diagram of IWD algorithm in solving TSP problem

2.3. Backtrack Algorithm

Backtrack algorithm is one of search algorithm that give promising solution which is optimal solution. However, this algorithm has the disadvantage that in time/duration to obtain an optimal solution. The approach of this algorithm is try all the alternative choices. Below is the Backtrack algorithm for TSP [4].

```

1. Function TSP_Backtrack (A, l, lengthSoFar, minCost)
2. n ← length[A] //number of elements in the Array A
3. if l = n
4.   then minCost ← min(minCost, lengthSoFar + distance[A[n], A[1]])
5.   else for i ← l+1 to n
6.     do Swap A[l+1] and A[i] //select A[i] as the next city
7.     newLength ← lengthSoFar + distance[A[l], A[l+1]]
8.     if newLength ≥ minCost //this will never be a better solution
9.       then skip //prune
10. else minCost ← min(minCost, TSP_Backtrack(A, l+1, newLength, minCost))
11.   Swap A[l+1] and A[i] // undo the selection
12. return minCost
  
```

2.4. TSP Library

To measure the performance of a method or algorithm, required a lot of problems that can be accessed by the public. The problem commonly called benchmark problems or *instances* or *dataset*.

Algorithm researchers around the world will be using the same problems to compare the performance of the algorithms they created.

For the Travelling Salesman Problem, the problems can be found in TSPLIB, a library consisting of many problems related to TSP [2]. In TSPLIB library, cost in every edge is mention in the dataset and usually use Euclidean Distance 2 dimension (EUC_2D). Mathematical formula to calculate the distance to the Euclidean Distance method can be seen below.

$$Cost = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \dots\dots\dots (eq. 1)$$

Which, X_1 and Y_1 is coordinate 1st node and X_2 and Y_2 is coordinate 2nd node. Below is an example of TSPLIB dataset.

```
NAME : TSP-n8
TYPE : TSP
DIMENSION : 8
EDGE_WEIGHT_TYPE : EUC_2D
NODE_COORD_SECTION
1 50 50
2 70 90
3 90 80
4 85 25
5 75 10
6 53 2
7 8 48
8 8 80
EOF
```

Annotations for the dataset format:

- NAME : TSP-n8 → Problem name
- TYPE : TSP → Problem type
- DIMENSION : 8 → Number of city
- EDGE_WEIGHT_TYPE : EUC_2D → Edge weight type
- NODE_COORD_SECTION (with coordinates) → City coordinate
Format : <no> <x> <y>
- EOF → End of dataset

3. RESEARCH METHOD

The research was begun by studying Travelling Salesman Problem and represents it into graph. By using the graph, IWD will solve the TSP and give the orders of the visited city as an output and time in solving the problem will show.

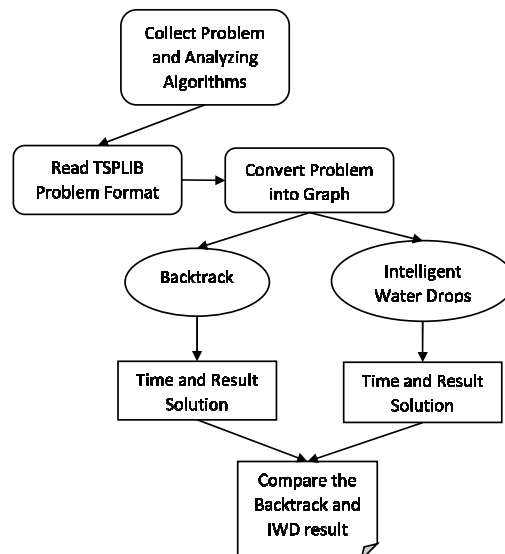


Figure 2. Block diagram of Research Methods

In Backtrack algorithm, the algorithm search the next city based on the distance from one city to the next city until all cities was visited. The distance is got by using eq.2 or Euclidean Distance. Backtrack algorithm will conduct or try all route/tour possibilities to find the shortest route by compare it with the last or the best route that have been travelled.

TSP problem that define in TSPLIB library is represented in form of a graph. This is because IWD algorithm need graph as the input problem. First the IWD will spread randomly in one of the cities, and then the IWD choose the next city based on probability that get from equation found in [1]. After next city is found the amount of soil and velocity is recalculate to get the actual amount. This process will repeated until all the cities was visited and in the last process the complete route/tour will compare with the best tour gets in every iteration.

As the final steps, comparison will be done by comparing both of the result tour. Time/duration and the route is used as the comparison factors. The time/duration factor represents in hours, minutes, seconds, and milliseconds while route represents in list of the visited city from start city and return to start city again.

4. RESULTS AND ANALYSIS

Program is made to compare backtrack algorithm with IWD algorithm. The program is made using C# language. As the input, first program reads the TSP problem from TSPLIB dataset, explained in chapter 2.4, which is already modified by choosing only several cities randomly. And as the result, the program will give out best tour, cost for the tour, and time taken to find the tour solution. Here are the comparison results in a problem Travelling Salesman Problem.

Table 1. The Result of Backtrack and IWD Algorithm in Solving TSP

Number of City	Differences	Backtrack	IWD, With initial value: Water Drops: 15; Velocity: 100; Soil: 10000; Tour: 10
5	Best Tour	0, 1, 2, 4, 3, 0	0, 3, 4, 2, 1, 0
	Cost	84	84
	Time	00:00:00.0020013	00:00:00.1310859
6	Best Tour	0, 1, 2, 4, 5, 3, 0	0, 1, 2, 4, 5, 3, 0
	Cost	140	140
	Time	00:00:00.0010004	00:00:00.0170117
7	Best Tour	0, 1, 6, 4, 5, 3, 2, 0	0, 1, 6, 4, 5, 3, 2, 0
	Cost	198	198
	Time	00:00:00.0010015	00:00:00.0180156
8	Best Tour	0, 1, 6, 2, 5, 4, 3, 7, 0	0, 7, 3, 4, 5, 2, 6, 1, 0
	Cost	152	152
	Time	00:00:00.0070038	00:00:00.0220134
9	Best Tour	0, 3, 5, 2, 8, 1, 7, 4, 6, 0	0, 6, 4, 7, 1, 8, 2, 5, 3, 0
	Cost	9990	9990
	Time	00:00:00.0290209	00:00:00.0260155
10	Best Tour	0, 4, 3, 5, 2, 8, 1, 6, 7, 9, 0	0, 4, 3, 5, 2, 8, 1, 6, 7, 9, 0
	Cost	330	330
	Time	00:00:00.2421638	00:00:00.0300231
11	Best Tour	0, 6, 2, 10, 1, 3, 5, 4, 8, 9, 7, 0	0, 7, 9, 8, 4, 5, 3, 1, 10, 2, 6, 0
	Cost	192	192
	Time	00:00:02.5066696	00:00:00.0270189
12	Best Tour	0, 1, 2, 8, 7, 9, 11, 5, 6, 4, 10, 3, 0	0, 1, 2, 8, 7, 9, 11, 5, 6, 4, 10, 3, 0
	Cost	169	169
	Time	00:00:28.4319543	00:00:00.0390263
13	Best Tour	0, 4, 5, 1, 3, 8, 7, 10, 12, 11, 6, 2, 9, 0	0, 4, 5, 1, 3, 8, 7, 10, 12, 11, 6, 2, 9, 0
	Cost	235	235
	Time	00:05:49.5100063	00:00:00.0290197
14	Best Tour	0, 5, 4, 10, 1, 12, 13, 9, 11, 3, 7, 2, 6, 8, 0	0, 8, 6, 2, 7, 3, 11, 9, 13, 12, 1, 10, 4, 5, 0
	Cost	182	182
	Time	01:17:58.9770850	00:00:00.0310186
51	Best Tour	xxxxx	0, 7, 25, 30, 27, 2, 35, 34, 19, 15, 49, 8, 48, 4, 36, 16, 3, 46, 11, 45, 50, 37, 10, 31, 26, 5, 47, 22, 6, 42, 23, 13, 24, 17, 12, 40, 39, 18, 41, 43, 14, 44, 32, 38, 9, 29, 33, 20, 28, 1, 21, 0
	Cost	xxxxx	450
	Time	xxxxx	00:00:00.2141409

Backtrack algorithm always finds the best tour, seen in its travelling cost. The best tour is got by computing the distance travelled from start city and return to start city again. From the given result in the table above, shown that Backtrack algorithm is good while processing problems with few cities which is below 10 cities. After the problem increases in the number of cities, the process took very long time/duration to get the best tour. This is because of combinations of possibilities route for the algorithm is getting bigger.

In other hands, IWD algorithm has the ability to produce the best tour faster than Backtrack algorithm while facing the increasing numbers of the cities. IWD algorithm also gives out same route like in Backtrack algorithm but in descending orders. IWD algorithm is shown has better performance in time/duration from 10 cities and above.

From the given result in the table above, IWD algorithm will give the best time taken to find a solution if the problem consists of many cities to visit and same best tour or route like in Backtrack, shown in problem that have 11 cities. In every iteration to find the next city to visit, Backtrack algorithm tried all of the

city that never visited before one by one. This is affect the time taken by Backtrack algorithm to find the best solution. The IWD best tour route and travelling cost is same as in Backtrack.



5. CONCLUSION

In this paper, IWD algorithm is compared with Backtrack algorithm to benchmark the performance of IWD. IWD algorithm has better performance in time/duration rather than Backtrack algorithm in finding the optimal solution as the numbers of cities growth. IWD algorithm sometimes cannot find the optimal solution; this is because of the random probability equation [1] that used to find the next city to visit. Comparison using Backtrack algorithm is the best way to test performance of new algorithm, like Intelligent Water Drops.

REFERENCES

- [1] H. Shah-Hosseini, "Problem solving by intelligent water drops," *Proceedings of IEEE Congress on Evolutionary Computation (CEC) Institute of Electrical and Electronics Engineers Computer Society*, pp. 3226-3231, 2007.
- [2] G. Reinelt, Universität Heidelberg, "TSPLIB," <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>, 1997.
- [3] H. Shah-Hosseini, "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm," *International Journal Bio-Inspired Computation*, vol. 1, no. 1/2, pp.71–79, 2009.
- [4] -, http://www.win.tue.nl/~mdberg/Onderwijs/Algoritmiek_Material/backtracking.pdf, pp. 2-5.

BIBLIOGRAPHY OF AUTHORS

	<p>Indra Maryati received the S.Kom degree in Computer Science in 2010 from Sekolah Tinggi Teknik Surabaya, Indonesia. Currently she is pursuing a master's degree in Information Technology at Sekolah Tinggi Teknik Surabaya. Additionally, she is working as a Junior Lecturer at Sekolah Tinggi Teknik Surabaya, in Indonesia. Her research interests include Database Technology, Human Computer Interaction, Graph Visualization, and Intelligent Water Drops algorithm.</p>
	<p>Gunawan received the S.Kom degree in Computer Science in 1991 and M.Kom in Information Technology in 2005 from Sekolah Tinggi Teknik Surabaya, Indonesia. Currently he is pursuing a doctoral degree at Institut Teknologi Sepuluh Nopember, in Surabaya, Indonesia. He is currently working as Vice Dean of Academic Affairs and a Senior Lecturer at Sekolah Tinggi Teknik Surabaya. His research interests include Natural Language Processing (especially in Bahasa Indonesia), Artificial Intelligence, Data and Web Mining, Neural Network, Fuzzy, and Evolutionary Algorithm such as Genetic Algorithm.</p>