



Flowing Water Algorithm: A New Approach for Combinatorial Optimization Problems

Xuepeng Liu¹, Qing Wang^{1,2,*}, An-Da Li¹

¹School of Management, Tianjin University of Commerce, Tianjin, China

²Management Innovation and Evaluation Research Center, Tianjin University of Commerce, Tianjin, China

Email address:

wangqingtjcu@163.com (Qing Wang)

*Corresponding author

To cite this article:

Xuepeng Liu, Qing Wang, An-Da Li. Flowing Water Algorithm: A New Approach for Combinatorial Optimization Problems. *American Journal of Mechanical and Industrial Engineering*. Vol. 7, No. 3, 2022, pp. 45-52. doi: 10.11648/j.ajmie.20220703.12

Received: July 11, 2022; **Accepted:** August 16, 2022; **Published:** August 24, 2022

Abstract: Being greatly inspired by the natural flowing regulation of water, we propose a new meta-heuristic algorithm — *Flowing Water Algorithm* (FWA) for the solution of combinatorial optimization problems (COPs). Since the solution space of COPs is multidimensional, complex and has many local extreme values, according to our proposed method, it appears to be similar to an endless hilly area with mountains, valleys and plateaus. The downward-flowing water in such area finds its way to the lowest point in the hill. Water always flows downward and eventually converges at the lowest place without any outside intervention except for gravity. Such a flowing course can be deemed as a process for the water to seek for the lowest point. The proposed algorithm is derived from such a water flow process. This algorithm combines a local search strategy with a population-based search strategy to improve both local and global search abilities. Four operators, including the local search, water overflow, drilling water tunnel and evaporation-rain are included in FWA, making this algorithm successfully perform tabu search, positive feedback, “survival of the fittest”, and local optimum escape. Two examples of its application in the traveling salesman problem (TSP) show that FWA outperforms the benchmark methods for both solution quality and convergence speed.

Keywords: Combinatorial Optimization Problems, Heuristics, Flowing Water Algorithm, Artificial Water

1. Introduction

Combinatorial optimization problems (COPs) are widely found in practical applications such as scheduling, routing, placement, investment, configuration, design and insurance. Extensive studies have been carried out to investigate the methods for solving COPs. The currently available methods can be simply classified into two categories: exact approaches and non-exact approaches [1]. The exact approaches, such as enumeration, branch and bound, branch and cut, and dynamic programming, seek optimal solutions through exact computations. However, many combinatorial optimization problems are often NP-hard, and their search space (i.e., the number of possible choices) is characterized by the “combination explosion” because of the increase in the problem size. Thus, the exact approaches cannot obtain the optimal solutions of the combinatorial optimization problems within polynomial time. In this case, non-exact approaches

such as heuristic algorithms have been proposed to search for good solutions which are near-optimal within a reasonable computational time [2].

Heuristic algorithms are very popular approximate approaches used in practice. Heuristic algorithms cannot offer a convergence guarantee but can flexibly make use of the special properties of the search space to exploit the near-optimal solutions [2]. Inspired by the surrounding world (physics, nature, biology, etc.), researchers have proposed various intelligent heuristic algorithms [3], such as genetic algorithms (GA) [4], simulated annealing (SA) [5], Tabu search (TS) [6], ant colony optimization (ACO) [7] and particle swarm optimization (PSO) [8]. These techniques are widely applied to search for the satisfactory albeit not optimal solutions of combinatorial optimization problems within a limited time. Specifically, heuristic algorithms can be classified into two categories: tailored-heuristics and meta-heuristics. The standard tailored heuristics are designed to solve specific problems and tailored to fit the inherent

characteristics of the problem at hand. Generally, tailored heuristics are regarded to have little value in other problems, so most researchers pay close attention to meta-heuristics which can be readily adapted for many types of problems. Meta-heuristics are designed to be sufficiently flexible to solve as many different combinatorial optimization problems as possible. The user only needs to change the neighborhood structures when applying the method to another problem domain. Many well-known meta-heuristic methods have demonstrated their usefulness and efficiency such as SA [9], TS [10], greedy search [11], GA [12], ACO [13], and PSO [14].

Local search and population search are used as the two basic principles in the development of meta-heuristics [15]. Local search methods, such as SA and TS, are directed to explore the neighborhood solution space intensively for new better feasible solutions. Population search methods, such as GAs, ACO algorithms and PSO, integrate "self-adaptation operation" with "co-operation", enabling the individuals to evolve through iterations according to certain rules. "Self-adaptation operation" denotes the individual's independent evolution path. "Co-operation" means that the individuals synergistically evolve by exchanging information with each other. To date, many meta-heuristic algorithms have been introduced in the literature based on different philosophies of intensification and diversification [16]. Though the abovementioned meta-heuristics algorithms for combinatorial optimization problems have their own advantages, they also have disadvantages. This has motivated the ongoing exploration of new, better algorithms.

Inspired by the phenomenon of flowing water in nature, we propose a new meta-heuristic algorithm —Flowing Water Algorithm (FWA) integrating two main principles: local search and population search. This algorithm integrates the advantages of the above algorithms.

The remainder of the paper is organized as follows. Section 2 gives a brief description of COPs, especially the traveling salesman problems (TSP). In section 3, this paper systematically presents the proposed FWA, including the insights from the natural flowing water phenomenon, FWA's characteristics and the detailed algorithm operation. In section 4, the advantages of FWA are demonstrated by its application to the TSP and the obtained results are compared to those obtained using some well-known meta-heuristic algorithms. Finally, conclusions are drawn in the last section of this paper.

2. Combinatorial Optimization Problem Description

Combinatorial optimization problems can be formulated as follows. Given a solution $s = \{m_{i1}, m_{i2}, \dots, m_{in}\}$, (i_1, i_2, \dots, i_n) as a permutation of set $\{1, 2, \dots, n\}$. Let S be the space of the solutions and n be the problem size. Furthermore, let $f(s)$ be the optimality function that seeks the global optimal solution s^* , thus $s^* = \arg\{\min_{s \in S} f(s)\}$. Obviously, the solutions structure S and the optimality function $f(s)$ depend on the characteristics of the specific combinatorial optimization problems.

Without the loss of generality, in this paper we study the *Flowing Water Algorithm* by using the example of the well-known TSP as the benchmark of the combinatorial optimization problems. TSP can be described as follows. Assume n cities, $c_i (i = 1, 2, \dots, n)$ is used to construct a complete weighted undirected graph, i.e., $G = (C, E)$. Therefore, $C(G) = \{c_1, c_2, \dots, c_n\}$ is called the set of cities of graph G and $E(G) = \{e_{ij} | (i, j) \in \{1, 2, \dots, n\}, i \neq j\}$ is called the edge-set of graph G . Furthermore, the distances between every two cities i and j are already known and are denoted as d_{ij} . It is required in TSP that the salesman should travel through all cities in the city set without any repetition by taking the shortest path among all feasible paths. Thus, the mathematical model of this problem is given as:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n \quad \sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad (1) \\ & x_{ij} \in \{0, 1\} \quad i \neq j \end{aligned}$$

where x_{ij} denotes whether the edge e_{ij} is passed through. If yes, then it is marked as 1; if no, then it is marked as 0.

3. Flowing Water Algorithm

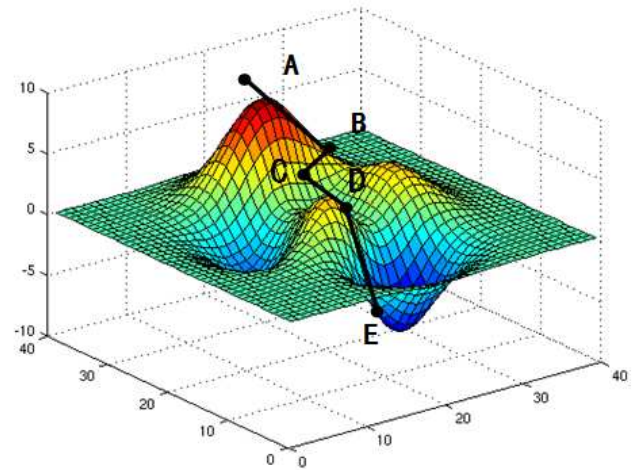


Figure 1. 3D plot of water flowing process.

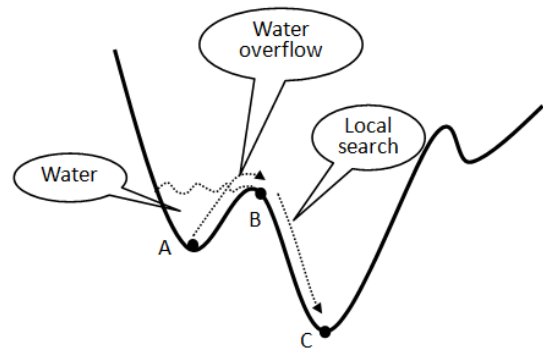


Figure 2. Pictorial representation of water overflowing.

Being greatly inspired by the natural flowing regulation of water, we propose a new meta-heuristic algorithm —*Flowing*

Water Algorithm (FWA) to obtain the near-optimal solutions of COPs. Since the solution space of COPs is multidimensional, complex and has many local extreme values, it appears to be similar to an endless hilly area with mountains, valleys and plateaus. The downward-flowing water in such area finds its way to the lowest point in the hill. Water always flows downward and eventually converges at the lowest place without any outside intervention except for gravity. Such a flowing course can be deemed as a process for the water to seek for the lowest point as shown in Figure 1. A water source from the initial position A flows through the anchor point positions (local minima) B, C and D, and eventually reaches the lowest point E. The proposed algorithm is derived from such a water flow process. The position points that the water flows through map the feasible solutions of COPs. Specifically, flowing water possesses the following characteristics:

Local search of flowing water. Water always flows downward and keeps searching for the next better or lower positions locally, which implies its nature of local search. Consequently, it always chooses the lowest and shortest possible path during its flow. Additionally, water possesses the characteristics of tabu search because it never flows inversely.

Overflowing of water. As shown in Figure 2, the flowing water stagnates at the locally good position A that it has just found. However, with the increase of the flow, the water level rises, and the water eventually overflows from an appropriate local suboptimal point B toward the global optimal point C. From the view of the optimization algorithms, such a feature of the flowing water reflects its ability to escape from the local minimum.

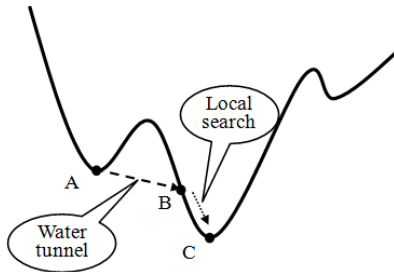


Figure 3. Pictorial representation of drilling water tunnel.

Drilling water tunnel. When water flows downward to the next position, the lower the position is, the greater the kinetic energy converted from the gravitational potential energy of the water. In an extreme case, if the gravitational potential energy is sufficiently high, water can chronically wear away the boundary limits to form a water tunnel which directs the following water to an even lower position. In practice, people often lead water to a lower position by artificially drilling a water tunnel. Similarly, as shown in Figure 3, when the flowing water stagnates at a locally good position A, we can drill a water tunnel to divert its path to another position B which is closer to the global optimum position C. The above phenomenon provides inspiration for solving COPs by using a “drilling water tunnel” to artificially help the searching process to jump out of the local minimum and accelerate the

convergence to the globally optimal solution.

Evaporation-rain. Competition is common in natural systems. Compared to the flowing water with the large water volume in the low position, the flowing water with a small water volume in the high position evaporates much more easily into vapor. Such water vapor usually rains randomly under certain conditions and thus some new flowing waters i.e. new positions (new positions denote new solutions) are generated. From the viewpoint of the optimization algorithms, evaporation shows the flowing water system’s feature of “survival of the fittest” which eliminates the relatively bad solutions. Additionally, rain diversifies the flowing waters population (new solutions) and endows the system with the capability of searching for global optima.

Convergence of water flow. In nature, flowing water tends to continually select the lower paths locally and may continuously converge to larger rivers in lower positions during the flow process and eventually find its way into the sea. Therefore, the lower paths attract more flowing water and correspondingly the water volume in these paths is larger. In other words, the paths which have accumulated more water lead the directions of the following flowing water. The above phenomenon reflects the strong positive feedback mechanism of water volume which directs the flowing water to choose the lower (optimum) paths toward the lowest position with greater probability.

3.1. Proposed Algorithm

Inspired by the natural flowing water phenomenon, the *Flowing Water Algorithm* (FWA) for solving COPs is proposed. This analogy endows our algorithm with advantageous characteristics of strong global search and local search capabilities, tabu search, positive feedback and the ability to escape from local minima. In FWA, we develop *artificial water* as an optimization tool. Compared to the natural flowing water, the artificial water has three additional characteristics as described below:

Countable. Each *artificial water* is a numerable and discrete “point” (its location denotes a solution of COPs) with some amount of flow volume which is inversely proportional to the altitude of the point.

Communication abilities. The individuals of the *artificial waters* population can exchange the information about the current global optimal point.

Memory abilities. Each *artificial water* can remember the information about all the points it has flown through including the current global optimal point.

The simple paradigm of the FWA is displayed in Figure 4. Without the loss of generality, the detailed design and corresponding operation of the FWA are given as follows:

A. Encoding

In FWA, each position that the *artificial waters* flow through denotes a single feasible solution. Different COPs usually need different encoding patterns. Taking the TSP as an example, a feasible solution to the problem is a path by which a salesman can travel through all cities without any repetition and the encoding pattern is a permutation of the cities set

$$C(G) = \{c_1, c_2, \dots, c_n\}.$$

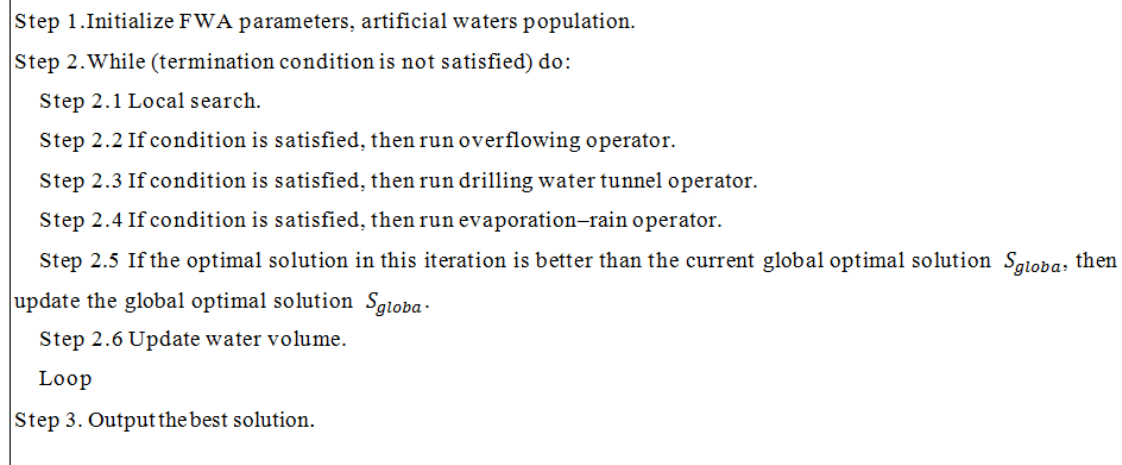


Figure 4. Paradigm of FWA.

B. Initialization

We comprehensively adopt the stochastic rule and the heuristic rule to generate the flowing artificial waters population, i.e., the initial feasible solutions. The stochastic method generates a part of the feasible solutions population randomly without any rules, thus it guarantees the diversity and randomness of the solutions population. Meanwhile, the heuristic method generates the rest of the initial solutions population according to heuristic rules, guaranteeing the goodness of the solutions population. In nature, water always flows downward and chooses the shortest possible path during its flow. Therefore, the initialization stage adopts the drop-altitude rule that the lower and shorter paths are more probable to be chosen by the artificial waters.

C. Local search operator

The local search capability of FWA enables the artificial water to constantly exploit the next better positions in their neighborhood. Each artificial water independently starts the local search at an initial feasible solution in its corresponding neighborhood of solutions obtained by making small changes. A better solution is found to replace the current solution. Moreover, artificial waters possess a memory ability which ensures that the previously visited solutions are not searched again. In this work, the local search proceeds by exchanging the positions of two codes within one-step ($k = 1$) distance in the encoding of a solution, as illustrated in Figure 5. For example, there exists a one-step ($k = 1$) distance between code a and code b in the encoding pattern and a new solution is obtained by swapping the positions of the two codes as shown in Figure 5.

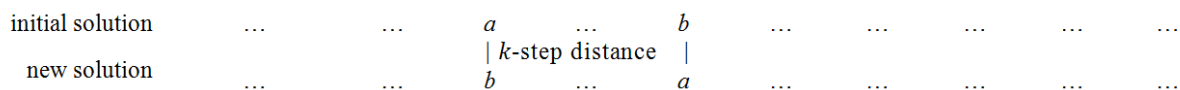


Figure 5. Pictorial representation of local search by exchanging points positions.

Generally, the local search operator is combined with the overflowing water operator. The detailed illustration of the use of the local search operator with the overflowing operation is given in Figure 6.

D. Overflowing operator

Overflowing operator enables the artificial water to escape from the current local optima. It forces the local search to proceed by positioning the artificial water at a suboptimum solution position. The placement of artificial water at a suboptimum solution forces it to escape from the current position and proceed to the local search; this is called the overflowing operator. In the case of Figure 2, the artificial water becomes trapped in the local optima A. Overflowing operator forces it to overflow from a suboptimum solution B which is obtained by modifying the neighborhood of the current solution A. This may deteriorate the objective function value but can definitely enable the artificial water to jump out

of the local minimum and to come closer to the global optima C. Specifically, the overflowing water operator has various tactics, and here we give one paradigm of the overflowing water operator for TSP, which is shown in step 4 of Figure 6.

E. Drilling water tunnel operator

To enable the artificial waters escape from the local optima and speed up the convergence to the current global optimal solution, the FWA employs the drilling water tunnel operator. In this work, we propose the following strategy for the TSP problem. First, a local optimum search for a better solution is performed with a certain number of steps from the current position to the current global best position (global optima). Specifically, firstly a comparing point in the encoding scheme is chosen and the local optimum and global optimum are compared, and then the local optimum swaps another code with the comparing point to make the comparing point have the same value as the global optima. Taking the comparing

point in Figure 7 as an example, code b in the encoding scheme of the original local optimal solution is different from code c in the encoding scheme of the current global optimal

solution. Thus, in order to approach the current global optimal solution, code c and code b in the encoding scheme of the local optimal solution are swapped, generating a new solution.

```

1. Given an artificial water  $i$  an initial solution  $s_i^0$ , set the step size  $k = 1$ .
2. While (termination condition is not satisfied) do:
    2.1 Construct the neighborhood  $N^k(s_i^0)$  of the solution  $s_i^0$  according to the step size  $k$ .
    2.2 If there exists a solution  $s'_i = \min(N^k(s_i^0))$  to be better than the solution  $s_i^0$ , then set
         $s_i^0 = s'_i$ ,  $k = k + 1$ , circles program from step 2.
    Else
        2.2.1 If  $k \geq 2$ , then let  $s''_i = \min(N^k(s_i^0))$ .
            If  $s'_i < s_i^0 < s''_i$ ,
                then set  $s_i^0 = s''_i$ , set  $k = 1$ , circles program from step 2 (This
                subsection runs overflowing operator).
            Else set  $k = k + 1$ , circles program from step 2.
        End if
        2.2.2 Else set  $k = k + 1$ , circles program from step 2.
        2.2.3 End if
    End if
End While
3. Output the best solution

```

Figure 6. Paradigm of local search operator and overflowing operator.

original solution	c	...	b
global optimal solution	c
new solution	b	...	c

Figure 7. Pictorial representation of drilling water tunnel operator.

F. Evaporation-rain operator

The evaporation-rain operator integrates evaporation and rain sub-operations. The inferior artificial waters (solutions) are evaporated and transformed into vapor in a certain proportion by the artificial evaporation and then the artificial rain transforms the vapor completely into the same amount of new artificial waters (solutions) as the evaporated artificial waters. The new solutions are randomly generated to diversify the flowing waters population. Specifically, the rain operator generates the new artificial waters according to the drop-altitude rule and the water volume rule, as illustrated below (see section 3.2 and 3.3).

3.2. Solution Generation

The feasible solutions of *artificial waters* in the *initialization* stage and the *artificial rain* stage are generated by successively choosing a series of fragment paths according to the *drop-altitude rule* and the positive feedback mechanism of *water volume rule*. The shorter paths at the same drop in altitude are chosen with higher probability by *artificial waters*. This demonstrates the principle of local optimization. Meanwhile, the larger the water volume of a path is, the more

likely is this path to be selected by the *artificial waters*, which implies the positive feedback mechanism of the water volume and stronger cooperation among all flowing *artificial waters*. This demonstrates the global optimization principle. For TSP, at the t th iteration, the *artificial water* k from point i selects path (i, j) as the next path to the point j with the probability of $p_{ij}^k(t)$ given by

$$p_{ij}^k(t) = \begin{cases} \frac{\lambda(w'_{ij}(t)) + (1-\lambda)a'_{ij}}{\sum_{j \in N_i^k} (\lambda(w'_{ij}(t)) + (1-\lambda)a'_{ij})}, & \text{if } j \in N_i^k \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where N_i^k denotes the collection of nodes that the artificial water k can choose to move to in the next step. For TPS, N_i^k consists of all unvisited nodes that are directly connected to node i in figure $G = (C, E)$. $a'_{ij} = 1 - a_{ij}^*$ is the path distance contribution parameter, and $a_{ij}^* = (a_{ij} - \min(a_{ij})) / (\max(a_{ij}) - \min(a_{ij}))$ denotes the normalized distance a_{ij} of path (i, j) . $w'_{ij}(t) = (w_{ij}(t) - \min(w_{ij}(t))) / (\max(w_{ij}(t)) - \min(w_{ij}(t)))$ is the water volume

contribution parameter calculated as the normalized value of $w_{ij}(t)$; and $w_{ij}(t)$ denotes the water volume of path (i, j) in the t th iteration. λ denotes the parameter that control the relative importance of the water volume.

3.3. Principle of Updating Water Volume

Simulating the natural convergence and evaporation regulation of flowing water, the FWA updates the water volume after each iteration. Specifically, after the t th iteration, the water volume $w_{ij}(t+1)$ of path (i, j) is updated as

$$w_{ij}(t+1) = w_{ij}(t) - \tau w_{ij}(t) + \sum_{k=1}^m \Delta w_{ij}^k(t), \forall (i, j) \in A \quad (3)$$

where $\tau \in (0, 1]$ denotes the evaporation ratio of the water volume.

After the t th iteration, the water volume of the *artificial water* k is denoted as $c^k(t)$ and the corresponding altitude is denoted as $d^k(t)$. These are inversely correlated as described by

$$c^k(t) = Q/d^k(t) \quad (4)$$

where Q is a constant. After the t th iteration, the correspondingly water volume increment $\Delta w_{ij}^k(t)$ on any fragment path (i, j) flowed through by the *artificial water* k is identical to $c^k(t)$, so $\Delta w_{ij}^k(t) = c^k(t)$.

4. Computational Experiments

Our FWA is elaborately devised based on the natural regulation of flowing water and is characterized by strong global and local search capabilities, tabu search, positive feedback and the ability to escape from local premature. To demonstrate its utility and effectiveness, this section will show the examples of its application to two standard TSPs with test data sets Eil51 and CH130.

The standard TSP Eil51 has about fifty-one cities and to date, the best solution to this problem is known as 426. In our experiment, the parameters for this problem are set as below: the number of flowing *artificial waters* population $m = 100$; the number of maximum iterations $NC_{max} = 100$; the evaporation proportion of the waters population $evap = 0.8$; the probability of drilling water tunnels $cave = 0.2$; the weight of water volume $\lambda = 0.5$; the evaporation ratio of water volume $\tau = 0.1$.

We ran FWA twenty times using the parameters specified above, and the best result is achieved as 428.9816 which is very close to the known best value of 426. Correspondingly, the best tour is (44, 17, 37, 15, 45, 33, 39, 10, 49, 5, 38, 11, 32, 1, 22, 2, 16, 50, 9, 30, 34, 21, 29, 20, 35, 36, 3, 28, 31, 8, 26, 7, 43, 24, 23, 48, 6, 27, 51, 46, 12, 47, 4, 18, 14, 25, 13, 41, 40, 19, 42), where the numbers represent the codes of the cities and their sequence reflects the travelling schedule. This tour can be observed in Figure 8 which gives the locations of all cities. Figure 9 illustrates the processes of the iterations and the lower line and the upper line respectively

stand for the best tour length value and the average tour length value. Compared to the results obtained by SA, TS, GAs, ACO and PSO methods reported in the literature [17] that ran for the same times with the same number of maximum iterations as that in our experiment, it can be obviously seen from Table 1 that the results obtained by FWA are much closer to the already-known best solution to this problem than those of the other meta-heuristics algorithms. Furthermore, the FWA achieves these results using many fewer iterations.

Table 1. Comparisons of various meta-heuristic algorithms on eil51.

algorithm	Best optima	Worst optima	Average iterations
AA	450.9815	455.8169	123
GA	467.9805	490.1634	259
SA	439.3299	444.6047	172
TS	491.8765	544.4865	277
PSO	456.5339	487.7408	264
FWA	428.9816	438.5315	46

To evaluate the effectiveness of FWA in solving large-scale combinatorial optimization problems, this paper uses the standard TSP test data CH130 with approximately 130 cities which has the exact solution of 6110. In our experiment, the parameters for this problem are set as follows: the number of flowing waters population $m = 100$; the number of maximum iterations $NC_{max} = 500$; the evaporation proportion of the waters population $evap = 0.8$; the probability of drilling water tunnels $cave = 0.2$; the weight of water volume $\lambda = 0.5$; the evaporation ratio of water volume $\tau = 0.1$.

We ran FWA twenty times using the parameters above, obtaining the best result of 6188.9484 which is very close to the exact solution of 6110. Correspondingly, the best tour is (39, 71, 50, 130, 50, 2, 118, 80, 46, 20, 93, 37, 22, 47, 40, 23, 122, 55, 60, 51, 42, 44, 4, 35, 54, 17, 34, 31, 27, 19, 100, 116, 24, 15, 29, 95, 79, 12, 87, 81, 103, 77, 94, 89, 110, 98, 68, 48, 25, 113, 32, 36, 84, 119, 111, 123, 101, 82, 57, 9, 56, 65, 52, 75, 74, 99, 73, 92, 38, 106, 58, 49, 53, 120, 72, 91, 6, 102, 10, 14, 96, 67, 13, 33, 21, 18, 108, 8, 126, 114, 3, 83, 76, 109, 61, 129, 124, 64, 69, 86, 88, 26, 97, 70, 63, 43, 104, 107, 127, 1, 41), where the number represent the codes of the cities and their sequence reflects the travelling schedules. This tour can be observed in Figure 10 which gives the locations of all cities. Figure 11 illustrates the processes of iterations and the lower line and the upper line, respectively, represent the best tour length value and the average tour length value.

Compared to the results obtained using the SA, TS, GAs, ACO and PSO methods as reported in the literature [17], where the same times with the same number of maximum iterations were use as in our experiment, the results obtained by the FWA are much closer to the exact solution of 6110 than those obtained by the other meta-heuristics algorithms as shown in Table 2. Additionally, the FWA requires relatively fewer iterations.

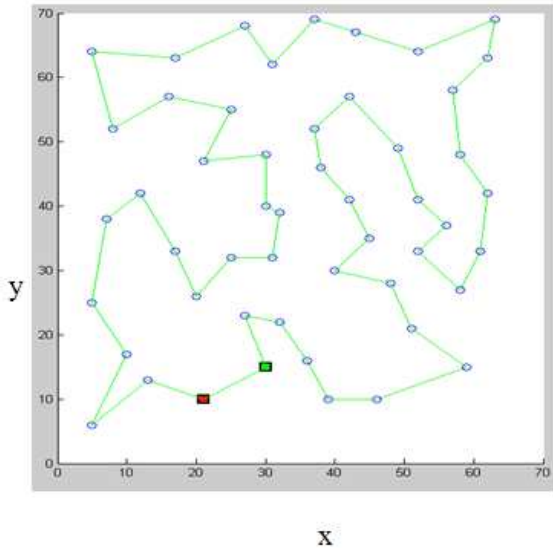


Figure 8. The optimal travelling path for Eil51.

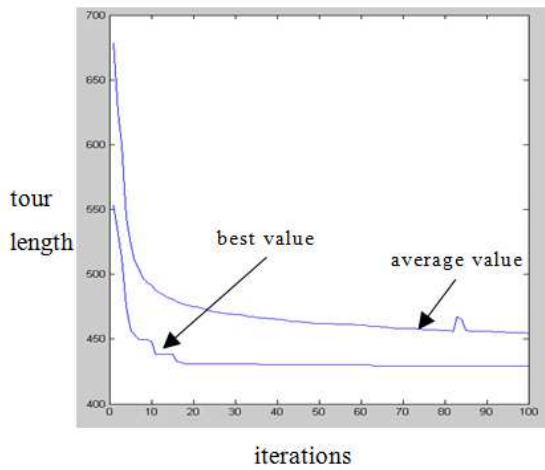


Figure 9. Iteration process of solving Eil51.

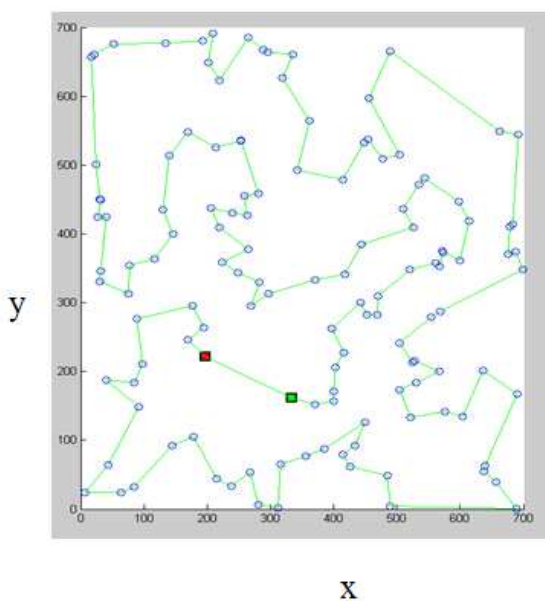


Figure 10. The optimal travelling path for CH130.

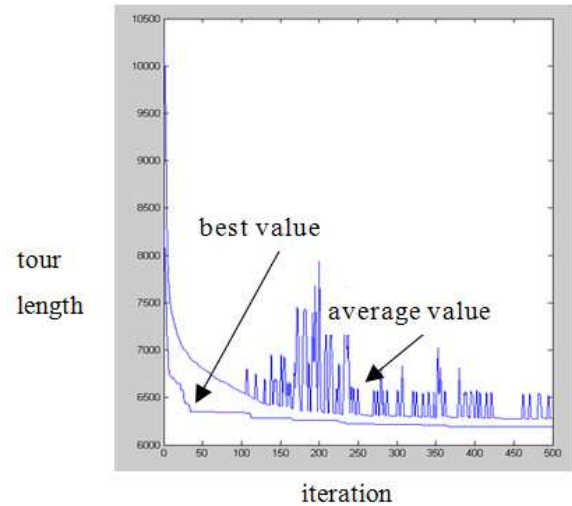


Figure 11. Iteration process of solving CH130.

Table 2. Comparisons of various meta-heuristic algorithms for CH130.

algorithm	Best optima	Worst optima	Average iterations
AA	6417.9875	6430.9665	151
GA	7118.2572	7322.3878	432
SA	6350.4431	6471.0394	266
TS	10516.1795	12258.2495	354
PSO	8134.7704	8234.6089	407
FWA	6188.9484	6258.2782	346

5. Conclusion

This paper proposes the novel meta-heuristic algorithm, FWA, inspired by the natural phenomenon of flowing water, for the solution of COPs. FWA integrates four operators including the local search operator, overflowing operator, drilling water tunnel operator and evaporation-rain operator. It displays six search properties, including (i) combination of local search and population search, (ii) feature of tabu search strategy because water never flows inversely, (iii) ability of escaping from local convergence by using the overflowing water operator, (iv) acceleration of the convergence speed to the global optimum by using the drilling water tunnel operator; (v) ability to speed up the search for global optimal solutions with the evaporation-rain operator, which embodies the idea of “survival of the fittest” and diversifies the flowing waters population; (vi) allowing water volume positive feedback for a more efficient and effective search by the synergy of *artificial waters*. We adopt FWA for solving TSP and test its efficiency using two representative TSP cases. The results show that FWA outperforms the benchmark meta-heuristic algorithms with respect to the quality of final solutions, number of iterations and convergence speed.

The research on the FWA in this paper is currently only in the preliminary stage, and many further studies should be carried out for the characteristics of this algorithm in the future, such as the astringency, theoretical basis, the sensitivity and the impact that the parameters have on the optimization. Given that the FWA displays its remarkable superiority in optimization of combinatorial optimization problems, it can

be anticipated that this algorithm possesses broad prospects of application, requiring ongoing explorations and in-depth investigations in future studies.

Conflicts of Interest

The authors declare that they have no competing interests.

Funding

The publication of this study was supported by the Research Project of Tianjin Education Commission for the project entitled "Retailer purchasing decision-making under the interactive effect of supply interruption risk and strategic customer behavior", with the Grant NO. 2017SK071.

References

- [1] M. Randall, D. Abramson, A general meta-Heuristic based solver for combinatorial optimisation problems, *Computational Optimization & Applications*, 20 (2) (2001) 185-210.
- [2] G. A. Kochenberger, F. Glover, B. Alidaee, C. Rego, A unified modeling and solution framework for combinatorial optimization problems, *Or Spectrum*, 26 (2) (2004) 237-250.
- [3] Z. Michalewicz, D. B. Fogel, *How to solve it: Modern heuristics*, Springer Science & Business Media, 2013.
- [4] J. H. Holland, *Adaptation in natural and artificial systems*, MIT Press, 1992.
- [5] S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies, *Journal of Statistical Physics*, 34 (5-6) (1984) 975-986.
- [6] F. Glover, Future paths for integer programming and links to artificial intelligence, *Computers & Operations Research*, 13 (5) (1986) 533-549.
- [7] M. Dorigo, V. Maniezzo, A. Coloni, Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society*, 26 (1) (1996) 29.
- [8] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory. In *Micro Machine and Human Science, MHS'95, Proceedings of the Sixth International Symposium on IEEE* (1995) 39-43.
- [9] P. Tian, J. Ma, D. M. Zhang, Application of the simulated annealing algorithm to the combinatorial optimisation problem with permutation property: An investigation of generation mechanism, *European Journal of Operational Research*, 118 (1) (1999) 81-94.
- [10] M. Obaidullah, G. N. Khan, Application mapping to mesh NoCs using a Tabu-search based swarm optimization, *Microprocessors & Microsystems*, 55 (2017) 13-25.
- [11] L. Fanjul-Peyro, R. Ruiz, Iterated greedy local search methods for unrelated parallel machine scheduling, *European Journal of Operational Research*, 207 (1) (2010) 55-69.
- [12] S. Sasikala, S. A. alias Balamurugan, S. Geetha, A novel adaptive feature selector for supervised classification, *Information Processing Letters*, 117 (2017) 25-34.
- [13] R. Saini, N. Anand, A multi-objective ant colony system algorithm for virtual machine placement, *International Journal of Engineering Research & Applications*, 7 (1) (2017) 95-97.
- [14] C. Jing, X. Wang, Identification of Hammerstein systems with continuous nonlinearity, *Information Processing Letters*, 115 (11) (2015) 822-827.
- [15] A. Hertz, M. Widmer, Guidelines for the use of meta-heuristics in combinatorial optimization, *European Journal of Operational Research*, 151 (2) (2003) 247-252.
- [16] A. H. Halim, I. Ismail, Combinatorial optimization: Comparison of heuristic algorithms in travelling salesman problem, *Archives of Computational Methods in Engineering*, (2017) 1-14. <https://doi.org/10.1007/s11831-017-9247-y>.
- [17] J. Shen, K. Wang, The light ray particle swarm optimization for solving the traveling salesman problem, *CAAI Transactions on Intelligent Systems*, 7 (2012) 174-182. (In Chinese).