Journal of Computer Sciences

Original Research Paper

Solving School Bus Routing Problem by Intelligent Water Drops Algorithm

[1]Ameera Saleh Jaradat and [2]Mohammed Qassim Shatnawi

[1]Department of Computer Science, Faculty of Information Technology and Computer Science,
Yarmouk University, Irbid, 21163, Jordan
[2]Department of Computer Information Systems, Jordan University of Science and Technology,
P.O. Box(3030), Irbid (22110), Jordan

Article history
Received: 01-06-2017
Revised: 28-11-2017
Accepted: 30-12-2017

Corresponding Author:
Mohammed Qassim Shatnawi
Department of Computer Information Systems, Jordan University of Science and Technology, P.O. Box(3030), Irbid (22110), Jordan
Fax: + 962 (0) 2 7095123
Email: mshatnawi@just.edu.jo

Abstract: The School Bus Routing Problem (SBRP) is an essential subject in the development of transportation. Solving the problem will have a great impact on the transportation systems by improving the quality of the provided service and reducing the operations costs. SBRP tries to find an efficient school buses plan where each bus takes up students from different bus stops and transports them to their schools while fulfilling number constraints including the maximum bus capacity and the time window of a school. This study used Intelligent Water Drops (IWD) optimization approach to resolve (SBRP). IWD is a meta-heuristic swarm-based optimization technique that simulates natural water drops. The scheme of the IWD algorithm is applied here to find a reasonable solution to SBRP. The application of IWD algorithm produces satisfying results within a reasonable amount of time.

Keywords: School Bus Routing Problem (SBRP), Intelligent Water Drop (IWD), Optimization, Meta-Heuristic

Introduction

The school bus routing problem is essentially engaged in the system of the school transportation, which involves transporting students to and from the specified bus places (Schittekat *et al*., 2006). The school bus routing problem is a variation of the travelling salesman problem, generally belongs to the class of Vehicle Routing Problems (VRP) (Lim *et al*., 2011), furthermore, SBRP deals with many objectives including minimizing the transportation cost and time (Kamkar *et al*., 2010).

The most important features that need to be considered during building a solution to the SBRP are: Efficiency which involves the entire cost of running the school bus and effectiveness which deals with the satisfaction of the provided service. These features are aligned with the fairness feature which involves assigning each student a school bus. Therefore, the SPRP as an optimization problem that deals with two subject matters: The student to bus stop assignment and bus routing (Corberán *et al*., 2002).

The goal is to maximize bus utilization while reducing transportation times, given that, an on time student deliverance to the school need to be maintained. On the other hand, the SBRP has constraints that need to be considered while building the solution which include: The limitation of the traveling duration of a student in a bus (e.g., each path should not be longer than 30 min), the capacity of buses and the time window to the school (e.g., each bus has to return to the school before the classes start) (Khalid and AbdelMonaem, 2012).

The SBRP is in the class of NP-hard problems, meaning that no polynomial time algorithm has been found yet to resolve the problem. Many efforts have been made to find a solution to the SBRP using approaches such as genetic algorithm (Sghaier *et al*., 2013), Harmony search (Geem *et al*., 2005), Scatter search (Russell and Chiang, 2006) and ant colony optimization (Arias-Rojas *et al*., 2012). This study adopts the Intelligent Water Drops (IWD) algorithm to deal with SBRP. IWD is a meta-heuristic approach that has been implemented effectively on many optimization problems as the clustering problem (Hosseini, 2013) and the travelling salesman problem (Shah-Hosseini, 2009). The idea of IWD algorithm is derived from the dynamics of river schemes and is inspired by the events that happen among the water drops in rivers (Hosseini, 2013). In nature, the path that the river follows is usually created by swarms of water drops. In this algorithm the intelligent water drops IWDs collaborate together to

Science Publications

achieve more desirable solution for a given problem. During the application of the IWD technique the IWDs are formed with two major properties that are subject to change during the IWDs lifetime. These properties are: The velocity of water drop and the amount of soil it carries during its trip.

The study involves the following assumptions: First, there are a variety of students existing in diverse regions that need to go to the same school with a limited number of buses. Second, the bus is limited by a certain capacity. Third, the travelled should not exceed a certain distance. Finally, there is a time window that forces each bus to come back to the school before the classes start. Having these constraints, the goal is to locate the best possible route that minimizes the distance travelled by the buses and minimizes the total cost.

This paper is structured as follows: First, the introduction provides information about SBRP and IWD algorithm. Then, the literature review lists some of the previous work that was proposed to solve the problem. The methodology section explains the IWD approach to solve the problem, which is followed by the results obtained from the experiment. Finally, the conclusion is presented to summarize the main outcomes of this research.

Literature Review

The previous work made on school bus routing and scheduling are very different in terms of objectives, constraints and assumptions and thus also in terms of solution techniques.

Spasovic *et al*. (2001) provided guidelines to schools and bus operators for the purpose of developing an efficient school bus routes. The guidelines were emerged upon performing a case study that involved Riverdale, New Jersey elementary School. The study applied three methods including the time saving heuristic method, ROUTER and the sweep method. The capacity, road length and total tour time were the main constraints in the problem. The results derived from the time saving heuristic and the ROUTER methods were extremely near and more valuable than those derived from the sweep method. Though the heuristic method cost an enormous amount of time. But, considering all of the various constraints, the heuristic method was counted as the most accurate one. However, the ROUTER method takes into account for quick adaptations for alterations in the student mobility rate throughout the year. So, for city districts that have one school or one bus station, the ROUTER methods maintain the optimum results.

Li and Fu (2002) formulated the bus routing problem as a multi-objective combinatorial optimization problem through a case study. The main objectives were minimizing the total obligatory count of buses, the total travel time that the students spend at pick-up points and the total bus travel time, in addition to harmonizing the loads and travel times among the buses. Li and Fu (2002) developed a heuristic system that was tested against data from a kindergarten in Hong Kong. The reported results showed a reduction of 29% in whole travelling times when compared to the existing system.

Geem *et al*. (2005) developed a heuristic that utilizes the music inspired Harmony Search (HS) algorithm. The objectives include minimizing the overall travel time of the provided buses as well as the travel time of the entire buses, subject to the limitations of the capacity of each bus and time window. The experiment designed a system consisting of one school, one bus and 10 bus stops. The HS algorithm was applied and the results were compared to results from other techniques like the Genetic Algorithm (GA). The results showed that the HS algorithm was able find good solutions that are in some cases better than genetic algorithm.

Russell and Chiang (2006) proposed a scatter search algorithm for the purpose of solving the Vehicle Routing Problem in the Company of time Windows and Split Deliveries (VRPTWSD). The scatter search is an example of evolutionary algorithms that generates a new solution upon operating on a set of initial reference solutions. The process involves providing means of solutions combination, variation and escalation of the search process in order to find the optimal rout. Russell and Chiang (2006) applied the scatter search algorithm on Solomon’s problem set consisting of 100 clients with Euclidean distance and a varying proportion of time windows according to the provided problem.

Bektas and Elmasta (2007) and Fugenschuh (2009) used integer programming as an approach to resolve School Bus Routing Problem (SBRP). The main objective was to reduce the entire cost of students’ transportation to and from the school, taking in consideration the capacity and distance. The results of applying the integer programming formulation were a total of around 28% to 32% reduction of the overall cost when matched up to the existing direction-finding method.

Arias-Rojas *et al*. (2012) proposed a heuristic that uses Ant Colony Optimization (ACO). The heuristic was implemented using actual facts from the school. The objective was to extend bus usage and to cut down travel periods for students, considering the delivery of students without delay as a constraint. Arias-Rojas implemented "cluster-first route-second" strategy which consists of grouping the students into clusters and then, solving each cluster by finding a rout. The results of running the algorithm and selecting the best routs shows an average reduction in distance travelled by 8.3% in the morning and 21.4% in the afternoon which was compiled to a total reduction of 30 min in average for each bus. The main challenge in this approach is speeding up the algorithm while reducing travel time. Sayyah *et al*. (2016) developed a new version of ant colony optimization that uses three search methods, which effectively outperform the existing ones.

Abounacer *et al*. (2009) introduces the Professional Staff Transportation Problem (PSTP) which is basically a variation of the (SBRP). PSTP aims to transport the staff of one or several companies using minimum total cost in view of the quality of service offered. The authors adopted Ant Colony Optimization (ACO) and Genetic Algorithm (GA) to formulate a meta-heuristic approach for solving the PSTP with the objectives of minimizing the transportation cost and optimizing the quality of service. The experimental results showed that applying ACO at first, plays an important role in generating efficient solutions.

Sghaier *et al*. (2013) used the Genetic approach to solve the (SBRP) with the introduction of new genetic operators for the purpose of enhancing the performance of the genetic algorithm. The objectives include minimizing the cost of the transportation by cutting down the overall traveled distance and the amount of bus operations. The application of the algorithm on real cases showed a 20% reduction of the number of buses used and 5% reduction of the traveled distance. The main inadequacy on this approach is in the absence of the bus capacity constraints.

Methodology

We now give an explanation of our proposed methodology but first we will give a brief presentation of the IWD algorithm. Then, apply the proposed IWD algorithm on the School Bus Routing Problem (SBRP)

Intelligent Water Drops Algorithm

The IWD algorithm was first introduced by Shah-Hosseini (2009). The IWD algorithm is an evolutionary algorithm rooted in the driving of river schemes and activities that develop between the water drops in waterway. The IWD algorithm was successfully implemented on different complex optimization problems. Shah-Hosseini (2009) tested the IWD algorithm on travelling salesman problem and obtained near-optimal solutions. Later, Hosseini implemented the IWD algorithm on different hard problems like the multiple knapsack problem (Hosseini, 2009) and the clustering problem (Hosseini, 2013). Harish *et al*. (2014) productively replaced IWD algorithm with the famous Genetic Algorithm (GA) for various experimental functions to gain multimodal peaks by showing less convergence time with IWD. Abdulelah *et al*. (2015) adopted the IWD algorithm for optimizing COQUAMO, which is a reproduction used to approximate the superiority of the software project. The results of this adaptation acquire more reasonable evaluation of software development quality.

In the IWD approach, IWDs have two key properties that control the process of finding their best possible pathway to their target (Hosseini, 2007). These properties are the velocity of the water drop and the amount of soil it carries while moving. Both properties change during the runtime of the algorithm. In IWD algorithm, less soil indicates more velocity that’s why IWDs choose path with less soil on its links.

An IWD moves from a starting place to a ending. The speed and the amount of soil of the IWD are initially zero. During its travel, the water drop moves in the surroundings from which it takes away some soil and it may obtain some speed. From its existing location to its subsequent location, the speed of IWD grows a nonlinear, inversely relative amount to the soil level between two positions. Therefore, an IWD in a pathway with a smaller amount soil becomes faster than an IWD in a pathway with extra soil. The mechanism in IWD is to choose the trail to its subsequent position. In this mechanism, the IWD favors pathways holding little soil levels rather than pathways having high soil levels. This behavior of path election is carried out by establishing a uniform random distribution on the soils of the obtainable paths. Then, the probability of the next step to be selected is contrarily correlated to the soil level of the reachable paths. Therefore, paths with lower soil levels have a higher opportunity to be chosen by the IWD.

The IWD algorithm represents the SBRP in the form of a graph $G(N, E)$. N corresponds to the node set, namely places. E corresponds to the edge set, namely distances between stops. Then, each IWD goes into creating its way out stepwise by moving between the stops until the IWD eventually achieves a solution. Algorithm iteration is accomplished when all IWDs have carried out their solutions. Upon the iteration completion, the iteration-best solution, which is referred to as T^{IB}, is predicted. Then, it is used to upgrade the total-best solution, which is referred to as T^{TB}. The quantity of soil on the lines of T^{IB} is decreased depending on the superiority of the solution. After that, the algorithm proceeds with one more iteration using new IWDs still with the matching soils on the edges of the graph and the entire procedure is carried over. The algorithm terminates when it hits the upper limit number of iterations $iter_{\max}$ or the T^{TB} arrives at the anticipated quality.

Next, we specify the major steps of the IWD algorithm as explained by Hosseini (2013).

Step 1: Initialize static parameters. Static parameters are unchangeable during the runtime of the algorithm. The graph $G(N, E)$ that describes the question is provided to the algorithm

The maximum number of iterations $iter_{\max}$ is assigned by the operator. The iteration counter $iter_{count}$ is set to zero. The number of water drops N_{IWD} is set to a non-negative integer number, which is generally assigned the count of nodes N_c of the graph. For updating the velocity, the parameters a_v, b_v and c_v are used. For soil updating, the parameters a_s, b_s and c_s are used. To update local soil and the global soil, the parameters ρ_n and ρ_{IWD} are used respectively, where ρ_n and ρ_{IWD} are small positive numbers whose value range between 0 and 1. The parameter $Inir_{soil}$ is the initial soil on each path. The amount of soil in the path connecting two nodes i and j is set by $soil(e_{i,j}) = Init_{Soil}$. The parameter $Init_{Vel}$ is the initial velocity of each IWD. These parameters are specified by the user and should be adjusted empirically in relevance to the operation.

Step 2: Initialize dynamic parameters. Dynamic parameters change at each iteration of the algorithm. Each IWD is assigned a list containing the nodes that have been visited *VC*(*IWD*), this list is to begin with empty: *VC*(*IWD*) = {}. The velocity of each IWD must have an $Init_{Vel}$. Initially, *soil*(*IWD*) is set to 0

Step 3: Distribute the IWDs at random over the nodes of the graph

Step 4: Update the list of visited nodes for each IWD by adding the recently visited nodes

Step 5: Reiterate Steps 5.1 to 5.4 for IWDs with partial resolutions

Step 5.1: For the IWD existing in node *i*, the following node *j* is selected, which does not rupture any of the constraints of the problem and not a member of the list of the visited nodes *VC*(*IWD*), by means of the function $p_i^{IWD}(j)$ in Equation 1:

$$p_i^{IWD}(j) = \frac{f(soil(e_{i,j}))}{\sum_{k \notin vc} f(soil(e_{i,j})} \quad (1)$$

f(), is defined as:

$$f\left(soil\left(e_{i,j}\right)\right) = \frac{1}{\varepsilon_s + g(soil(e_{i,j}))} \quad (2)$$

where, ε_s is a small absolute number to avoid the possibility of division by zero, here $\varepsilon_s = 0.0001$:

$$g\left(soil\left(e_{i,j}\right)\right) = \begin{cases} soil\left(e_{i,j}\right) if \min_{l \notin vc(IWD)} soil\left(e_{i,l}\right) \geq 0 \\ soil\left(e_{i,j}\right) - \min_{l \notin vc(IWD)} otherwise \end{cases} \quad (3)$$

Next, insert the recently visited node *j* to *VC*(*IWD*).

Step 5.2: Update the velocity for every IWD hopping from node *i* to node *j* using Equation 4:

$$vel^{IWD}(t+1) = vel^{IWD}(t) + \frac{a_v}{b^v + c_v.soil^2(e_{i,j})} \quad (4)$$

where, $vel^{IWD}(t+1)$ is the restructured velocity of the IWD.

Step 5.3: Calculate the $\Delta soil(i, j)$ that the IWD carries from the path for the IWD stepping from node i to node j by Equation 5:

$$\Delta soil(i,j) = \frac{a_s}{b_s + c_s.time^2(i,j;vel^{IWD}(t+1))} \quad (5)$$

where, *time*() is determined as:

$$time\left(i,j;vel^{IWD}(t+1)\right) = \frac{HUD(j)}{vel^{IWD}(t+1)} \quad (6)$$

where, *HUD*(*j*) is the heuristic function that scales the undesirability of the *IWD* to jump from node *i* to *j*. *HUD*(*j*) is determined fittingly for the given problem.

Step 5.4: During node visiting and edge selection, the IWD updates the soil it carries $soil^{IWD}$ and the soil of the edge $soil(e_{ij})$ by equation Equation 7 and 8:

$$soil(e_{ij}) = 1 + \rho_n.soil(e_{ij}) - (\rho_n.\Delta soil(e_{ij})) \quad (7)$$

$$soil^{IWD} = soil^{IWD} + \Delta soil(e_{ij}) \quad (8)$$

Step 6: After all IWDs construct there solution, get the iteration-best solution T^{IB} from all the solutions T^{IWD} produced by the IWDs using Equation 9:

$$T^{IB} = agr\max_{\forall T^{IWD}} q\left(T^{IWD}\right) \quad (9)$$

where, *q*() is the quality measure function of the provided solution.

Step 7. Fill in the amount of soils on the edges that outline the recent iteration-best solution T^{IB} by Equation 10:

$$soil\left(e_{ij}\right) = \left(1 - \rho_{IWD}\right).soil\left(e_{ij}\right) - \rho_{IWD}.\frac{1}{(N_c - 1)}.soil_{IB}^{IWD}\forall_{e_{ij}} \in T^{IB} \quad (10)$$

where, N_c represents the counted number of nodes in the solution T^{IB}.

Step 8. Update the total best solution T^{TB} by the existing iteration-best solution T^{IB} using Equation 11:

$$T^{TB} = \begin{cases} T^{TB}, & q(T^{TB}) \geq q(T^{IB}) \\ T^{IB}, & otherwise \end{cases} \quad (11)$$

Step 9: Increase the iteration counter $iter_{count}$ by one. Subsequently, jump to Step 2 if $iter_{count}$ is less than $iter_{\max}$

The previously mentioned steps are repeated until either the algorithm achieves the maximum number of iterations, or T^{IB} remains the same for a consecutive number of iterations.

IWD Algorithm for SBRP

In SBRP, a map of stops is provided and it is constrained to visit the entire stops one after another, this is necessary to complete the tour in a way that every stop is visited only once excluding the first one, as it is visited two times. The objective of this work is to locate a tour having a minimum total length amongst all available tours intended for the provided map. A map for the school bus route is interpreted by graph $G(N, E)$ where N is the node set representing the bus stops and E is the edge set representing the connections and weighted with the distances between the stops. The graph has n bus stops.

In this study, the graph representing the bus stops is a complete one; which means that, there exist a direct path between each bus stop and another. The graph in consideration is weighted with the distance. Therefore, given a graph $G(N, E)$, a solution to the SBRP is an ordered set of n different bus stops.

SBRP resolution for n-stop problem can be interpreted as a tour $T = (S_1, S_2, S_3,\ldots,S_n)$. The bus travels from stop S_1 to S_2, then from S_2 to S_3 and continues this until it reaches S_n. Upon reaching S_n, the bus proceeds to the first stop S_1 and the tour length, $TL()$ is computed by: $TL(S_1, S_2, S_3\ldots S_n) = \Sigma d(S_i, S_{i+1})$, where $S_{n+1}= S_1$. Here, we use the Euclidean distance function to calculate the distance between two stops. The objective of an optimization algorithm for SBRP is to locate the minimum length tour $T^* = \left(S_1^*, S_2^*, \ldots S_n^*\right)$ over all potential tours:

$$TL(T^*) \le TL(S_1, S_2, S_3 \ldots S_n)\ \textit{for every tour} \\ = (S_1, S_{2,} S_3, \ldots, S_n)$$

where, TL is the whole length for a specified tour and T^* is the best possible tour.

So as to employ the IWD approach for the SBRP, the School Bus Route (SBR) is anticipated as a complete weighted graph $G(N, E)$. The weight on every relation of edge set E represents a measure for the soil that the water drop carries while moving between each end of that link. As the IWD travels among nodes of graph, it changes the amount of soil on these links. The nodes of the graph represent the stops of SBR. Each holds the physical position of the bus stop it represents. Starting from an arbitrary node, an IWD moves to new nodes through the edges of the graph up till it proceeds to the starting point. While moving through links, the IWD alters the amount of soil of every edge.

The constraint on the SBRP is not to visit the bus stop more than once. In order to satisfy this constraint, the IWD next possible stop is selected from the stops that do not exist in the list of the nodes visited by IWD, $VC(IWD)$.

The time it takes the IWD to bypass bus stop i to bus stop j, corresponds to the function $HUD(i, j)$:

$$HUD(i, j) = \|s(i) - s(j)\|$$

where, $s(k)$ stands for the two dimensional positional vector for the bus stop k and the formula $\|s(i) - s(j)\|$ represents the Euclidean measure.

The local function $HUD(i, j)$ scales how it is undesirable for an IWD to travel from bus stop i to bus stop j. The undesirability value depends on the distance. When the distance between bus i and j is small, $HUD(i, j)$ grows to be large and $HUD(i, j)$ turns out small when the span between bus i and j is big.

To facilitate finding preferable tours with anticipation to avoid local optimums, we will use an adjustment that forces the IWD to get reinitialized after reaching a constant number. The pathways of the total-best solutions T^{TB} are assigned less soil than the other pathways:

$$soil(i, j) = \begin{cases} a_1 \mu_1\, Init_{Soil} & \textit{for every}(i, j) \textit{in } T^{IB} \\ Init_{Soil} & \textit{otherwise} \end{cases}$$

where, a_1 is a small positive number, the number chosen here is 0.10 and μ_1 is a random number, which is drawn from a uniform distribution in the interval [0, 1]. Accordingly, *IWD* chooses paths of T^{TB} with less soil on.

Now we summarize our approach into four phases: Initialization phase, construction phase, reinforcement phase and termination phase. The major steps of the IWD-algorithm are shown in Fig. 1:

1. Initialization phase: The major purpose of this phase is the formulation of the problem
   - The problem formulation consists of initializing both the static and dynamic parameters. The initial values for these parameters appears in Table 1
   - Set the the quality of the total-best solution T^{TB} to unfavorable amount. Here $q(T^{TB}) = \infty$
   - We have represented the bus stop graph G by a two dimentional matrix:

     $$G_{ij} = \begin{cases} 1, & \textit{if } (i, j) \in E \\ 0, & \textit{otherwise} \end{cases}$$

   - We also use a two dimentional distance matrix D_{ij} where the the edges are wighted with the distance between i^{th} and j^{th} bus stops
   - The vector tour is defined to contain the sequence of bus stops where each bus stops at during their tour
   - The soil is a two dimensional array initialized by the soil initial value

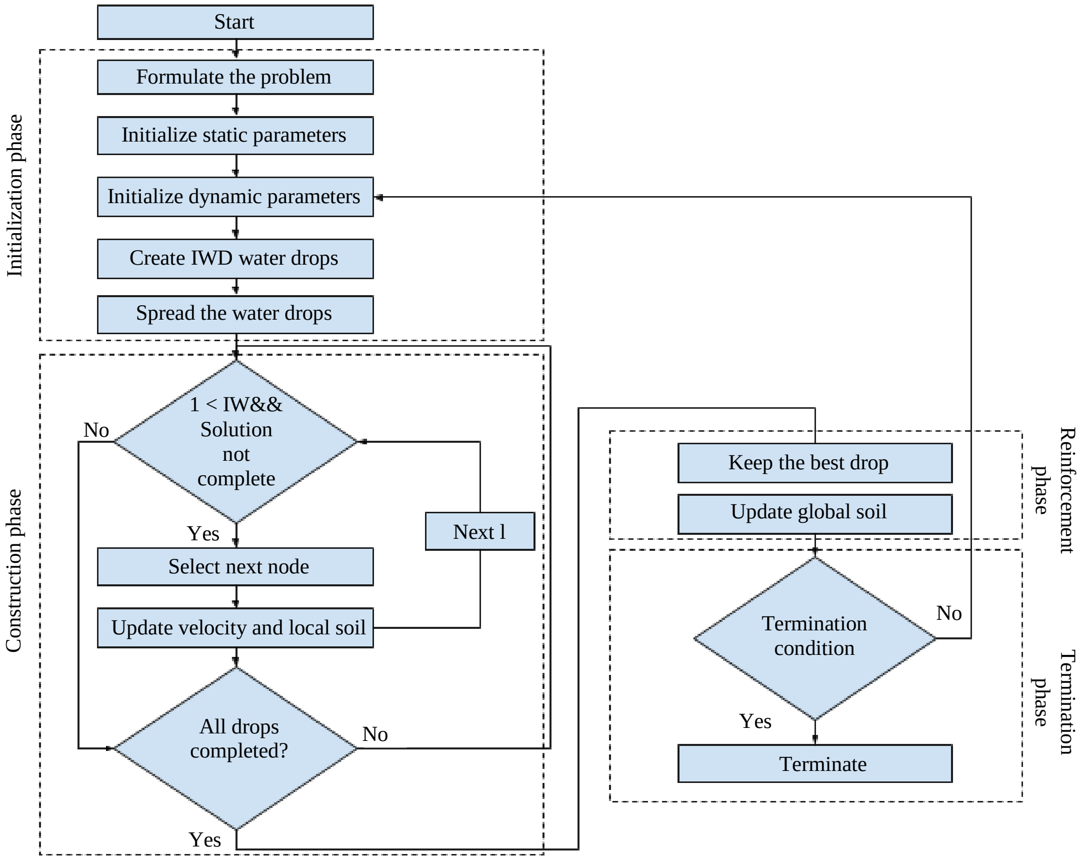

Fig. 1: A detailed flowchart of the steps of the proposed IWD algorithm

Table 1: Parameters of the Iwd experiments

Parameter	Value
a_v	1
b_v	0.01
c_v	1
a_s	1
b_s	0.01
c_s	1
$iter_{\max}$	1000
$Init_{soil}$	10000
$Init_{vel}$	200
ρ_{IWD}	0.9
ρ_n	0.9
N_{IWD}	Number of nodes in G

2. The construction phase: In this phase we have constructed the solution by implementing the steps of the IWD algorithm:

    - As mentioned earlier, the school bus rout is an undirected weighted graph $G(N, E)$, where the nodes represent the stops of the school bus route and the edges weights are the amount of soil the IWD carries
    - The applied algorithm distributes IWDs randomly through the graph
    - Initially, the graph weight is filled from the initial soil amount that each IWD carries. Later, these values are changed
    - As a result of applying the algorithm steps, the IWD moves through the links which lead to the reevaluation of the link weight that the water drop moves through
    - V_c is a vector representing the nodes that the IWD has not visited yet. This vector is updated by removing the currently visited node
    - Before moving the *IWD* from node *i* to *j*:

        - V_c is checked to make sure that each stop is visited once
        - Use the probability function in Equation 2 to move the IWD between *i* and *j*
        - Update the velocity using Equation 4 for every IWD traveling from *i* to *j*
        - Measure the amount of soil from Equation 5 that the IWD carries on edge e_{ij}
        - The algorithm applies the local heuristic function *HUD*(*i*, *j*) that measures the undesirability of an IWD to move from bus stop *i* to bus stop *j*. The function basically calculates the Euclidean distance between

    node *i* and *j*. Since smaller distances are preferable, *HUD*(*i*, *j*) returns the maximum distance
- Adjust the soil *soil*(*i*, *j*) of the edge e_{ij} when the IWD moves from vertex v_i to v_j using Equation 7

- This phase is repeated until a complete solution is formed. The solution is represented by a tour which is a vector of bus stops representing the sequence of stops that the IWD travels through.
- For each tour, the algorithm calculates the tour length by applying a function *d*() that takes the tour vector as an input and returns the distance value. The function calculates the Euclidean distance between each sequence of bus stops, then sums all the calculated distances.
- After all IWDs construct their complete solutions, the algorithm selects the iteration-best solution T^{IB} from all solutions obtained by the IWDs.
- Adjust the total best-solution T^{TB} using the present T^{IB} by Equation 11:

- For the quality measure, use the distance where the smaller the distance the better the quality

3. The reinforcement phase: The purpose of this phase is to make sure that the algorithm chooses a better solution. Here, the algorithm performs the followings:

- Choose the tour with the minimum length TL
- Adjust the soil to avoid local optimums. Supsequent to each determined count of iterations the amount of soil is reinitialized one more time so that the paths of the total-best solutions T^{TB} are provided with lower amount of soil than the other paths

4. Termination phase: After performing a large number of iterations, the algorithm terminates and return the total-best solution T^{TB}

Experimental Results

This division is devoted to present the results of applying IWD algorithm. Experiments were accomplished on a computer that has Intel core I3, 3.2 GHz processor, 2 GB Ram. The following parameters need to be initialized by the beginning of IWD (Basem *et al*., 2014): Count of water drops N_{IWD}, the count of cities N_C. The count of water drops is set equal to the number of buses and the count of cities depends on the given problem. The parameters a_v = 1000, b_v = 0.01 and c_v = 1 are used for adjusting the velocity. For soil updating, the study uses parameters a_s = 1000, b_s = 0.01 and c_s = 0.01. The preliminary soil of every edge is indicated by the constant $Init_{soil}$ and the soil of the edge connecting every two bus stops is initialized by $soil_{(i,j)} = Init_{soil}$. The constant parameter $Init_{Vel}$ denotes the initial velocity of IWDs. The $Init_{soil}$ and $Init_{Vel}$ parameters are initiated by the user. In this paper, $Init_{soil}$ = 1000 and $Init_{Vel}$ = 1000.

Euclidian distance is employed as a scale for measuring the distances between the nodes. The distance between (x_i, y_i) and (x_j, y_j) is computed by the following (Hosseini, 2009; Haykin and Kosko, 2001):

$$D_{ij} = \sqrt[2]{(x_i - x_j)^2 - (y_i - y_j)^2}$$

The proposed method was tested on 5 cities (as shown in Table 2) having 10, 50, 76, 100 and 60 stops, the distances for every city between these stops are mentioned in column 3 of Table 2, the distances calculated after implementing our suggested method once and after ten iterations using only one bus. Table 2 shows the reduction in the travelled distance after one iteration of implementing the method; this is reduced even more after 10 iterations.

Figure 2 shows the Convergence curves of two implementations of the IWD algorithm, which clarifies the reduction in the calculated distance after performing ten iterations of the algorithm.

The method also has been tested with $iter_{\max}$ = 10 (as shown in Table 3) for the same cities but this time considering two buses for collecting students. It can be seen from the results that the distance travelled in every city is reduced when the number of buses increased.

Figure 3 shows a curve indicating the time it took the algorithm to find the solution for forty different executions of the application scoring a median of 52.712108ms.

Table 2: Comparison between near-optimal length and IWD

City	Stops	Distance	IWD after one iteration	IWD after ten iteration
1	10	235.6 m	228.8 m	167.9 m
2	50	1313.4 m	953.2 m	600.8 m
3	76	1974.9 m	1704,8 m	990.9 m
4	100	191371.2 m	180031.9 m	129960.8 m
5	60	2989.8 m	2359.2 m	1607.89 m

Table 3: Comparison between near-optimal length and IWD using two buses with 10 iterations.

City	Stops	Distance	Distance after using IWD for bus number 1	Distance after using IWD for bus number 2
1	10	235.6 m	106.9 m	114.1 m
2	50	1313.4 m	953.2m	600.8 m
3	76	1974.9 m	1001.8 m	994.4 m
4	100	191371.2 m	67906.01 m	73254.80 m
5	60	2989.8 m	1192.5 m	1026.1m

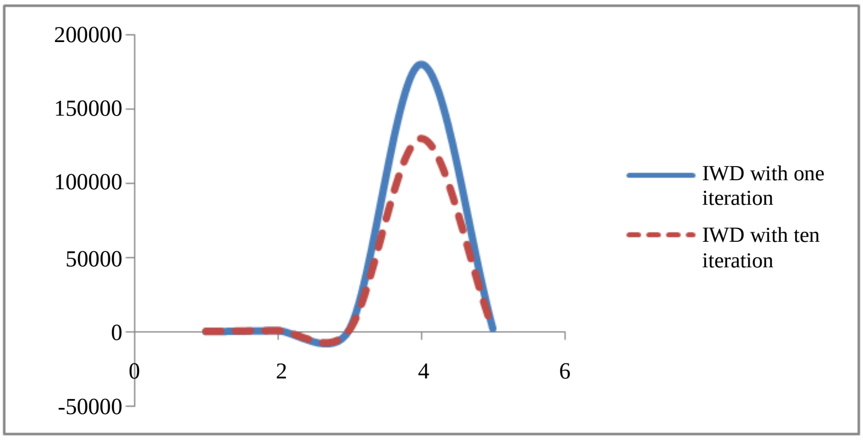

Fig. 2: Convergence curves of two runs of the IWD algorithm. The resulting distance of IWD with one iteration (solid) and ten iterations (dotted)

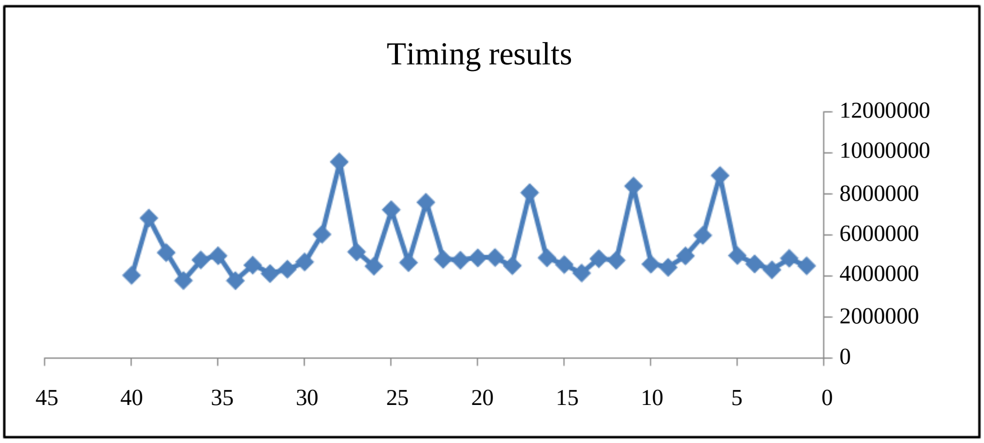

Fig. 3: An illustration of the timing results showing the time it took the IWD algorithm to find the solution for forty different runs of the algorithm

Conclusion

The IWD algorithm belongs to the class of nature inspired Algorithms, these algorithms imitate the natural act of swarms in nature to resolve problems. In the School Bus Routing Problem (SBRP), it is important to provide a solution that minimizes the total cost of transportation. In this study, we proposed an answer to the school bus routing problem through adapting the IWD approach. The motivation of the IWD algorithm launches from the behavior of the water drops as they flood in rivers.

The application of the IWD algorithm to solve the SBRP builds a graph of bus stops and links between these stops. Then, the heuristic spread the IWDs on the stops and the IWDs collaborate to accomplish a superior answer for the problem, by selecting the path with the minimum cost among all possible ones. According to experiment results, it is seen that the IWD approach is capable of detecting a near optimal estimation. Though, adjustments could be applied to the standard IWD algorithm through engaging other mechanisms and formulating local heuristics that go well with the problem.

For future work, a proposal of multi-objective optimization approach is needed to further analyze the trade-off between different costs like bus travel time, time window, travelled distance and number of busses.

Acknowledgement

We would like to acknowlede the Yarmouk university and Jordan University of Science and Technology because of using their resources in accomplishing this work.

Author’s Contributions

Ameera Saleh Jaradat: Conceived of the presented idea, carried out the experiment, designed and performed the experiments, derived the models and analyzed the data, and took the lead in writing the manuscript.

Mohammed Qassim Shatnawi: Contributed to the interpretation of the results, contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript. Also contributed to the final version of the manuscript

Both authors provided critical feedback and helped shape the research, analysis and manuscript.

Ethics

This manuscript is allowed to be acceseed based on the rules of the sience publications. Readers also can inquery the authors about anything related to this publication.

References

Abdulelah, G., S.A. Saif and Z. Fayed, 2015. Intelligent Water Drops (IWD) algorithm for COQUAMO optimization. Proceedings of the World Congress on Engineering and Computer Science, (ECS’ 15), San Francisco, USA.

Abounacer, R., G. Bencheikh, J. Boukachour, B. Dkhissi and A.A. Elhilali, 2009. Population metahuristic to solve the professional staff transportation problem.

Arias-Rojas, J.S., J.F. Jiménez and J.R. Montoya-Torres, 2012. Solving of school bus routing problem by ant colony optimization. Revista EIA.

Basem, A.O., W. Li-Pei, L.P. Chee, K.T. Ahamad and A.M. Al-Betar, 2014. A modified intelligent water drops algorithm and its application to optimization problems. Expert Syst. Applic., 41: 6555-6569.

Bektas, T. and S. Elmasta, 2007. Solving school bus routing problems through integer programming. J. Operat. Res. Society, 58: 1599-1604.

Corberán, A., E. Fernández, M. Laguna and R. Martí, 2002. Heuristic solutions to the problem of routing school buses with multiple objectives. J. Operat. Res. Society, 53: 427-435.

Fugenschuh, A., 2009. Solving school bus routing problems with integer programming. Eur. J. Operat. Res., 193: 867-884.

Geem, Z., K.S. Lee and Y. Park, 2005. Application of harmony search to vehicle routing. Am. J. Applied Sci.

Harish, Y., B. Venkateshwarlu and C. Jada, 2014. Intelligent water drops algorithm for multimodal spaces. Proceedings of the International Conference on Swarm, Evolutionary and Memetic Computing, (EMC’ 14), Springer, Cham.

Haykin, S. and B. Kosko, 2001. Intelligent Signal Processing. 1st Edn., Wiley, New York,

Hosseini, H., 2007. Problem solving by intelligent water Drops. Proceedings of the IEEE Congress on Evolutionary Computation, Sept. 25-28, IEEE Xplore Press, Singapore. DOI: 10.1109/CEC.2007.4424885

Hosseini, H., 2009. Optimization with the Nature-Inspired Intelligent Water Drops Algorithm. In: Evolutionary Computation, Wellington P.D.S., (Ed.), Austria: I-Tech, Vienna, pp: 297-320.

Hosseini, H., 2013. Improving K-means clustering algorithm with the intelligent water drops algorithm. Int. J. Data Min. Model. Manage.
ISBN-10: 0780360109, pp: 608.

Kamkar, I., T. R.M. Akbarzadeh andY. Mahdi, 2010. Intelligent water drops a new optimization algorithm for solving the vehicle routing problem. Proceedings of the IEEE International Conference on Systems Man and Cybernetics, Oct. 10-13, IEEE Xplore Press, Istanbul. DOI: 10.1109/ICSMC.2010.5642405

Khalid, A. and F. AbdelMonaem, 2012. A novel GIS-based decision-making framework for the school bus routing problem. Geo-spatial In. Sci., 15: 51-59. DOI: 10.1080/10095020.2012.708151

Li, L. and Z. Fu, 2002. The school bus routing problem a case study. J. Operat. Res. Society, 53: 552-558.

Lim, Y., P. Hong, R. Ramli and R. Khalid, 2011. An improved tabu search for solving symmetric traveling salesman problems. Proceedings of the IEEE Colloquium on Humanities, Science and Engineering, Dec. 5-6, IEEE Xplore Press, Penang, pp: 851-854. DOI: 10.1109/CHUSER.2011.6163857

Russell, R.A. and W. Chiang, 2006. Scatter search for the vehicle routing problem with time windows. Eur. J. Operat Res., 169: 606-622.

Sayyah, M., H. Larki and M. Yousefikhoshbakht, 2016. Solving the vehicle routing problem with simultaneous pickup and delivery by an effective ant colony optimization. J. Industrial Eng. Manage. Stud., 3: 15-38.

Schittekat, P., M. Sevaux and K. Sorensen, 2006. A mathematical formulation for a school bus routing problem.

Sghaier, S., N. Guedria and R. Mraihi, 2013. Solving school bus routing problem with genetic algorithm. Proceedings of the IEEE International Conference on Advanced Logistics and Transport, (ALT, 13), pp: 7-12.

Shah-Hosseini, H., 2009. The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm. Inspired Computation.

Spasovic, L., C. Kelnhofer-Feeley and S. Chien, 2001. A methodology for evaluating of school busrouting - A case study of riverdale, New Jersey. Trans. Res. Board,Washington, D.C..