Why the Intelligent Water Drops Cannot Be Considered as a Novel Algorithm

Christian Leonardo Camacho-Villalón(✉), Marco Dorigo, and Thomas Stützle

IRIDIA, Université Libre de Bruxelles, Brussels, Belgium
{ccamacho,mdorigo,stuetzle}@ulb.ac.be

Abstract. In this paper we show that intelligent water drops (IWD), a swarm intelligence based approach to discrete optimization proposed by Shah-Hosseini in 2007, is a particular instantiation of the ant colony optimization (ACO) metaheuristic. To do so, in the paper, we identify the components of IWD and place them into the ACO metaheuristic framework. We show therefore that there was no need for a new natural metaphor. We also discuss that the proposed metaphor does not bring any novel insight into the algorithmic optimization process used by IWD.

Keywords: Intelligent water drops · Ant colony optimization Novel algorithm

1 Introduction

Recently, many so-called *novel* approaches to stochastic optimization based on a natural metaphor have been proposed in the literature. Unfortunately, as also discussed in [29], such natural metaphors are often unnecessary or even misleading. For example, stochastic optimization algorithms based on diverse metaphors such as spiders [8], whales [22], grey wolves [23], birds [2], and so on, have been proposed and published in the literature. However, the real value of using a metaphor is often unclear. In some rare cases such as for harmony search [32] and black holes [24], it has been formally shown that the *novel* algorithm is just a re-formulation, using different terms, of an already well-known algorithm. In general, however, it remains challenging to understand whether the *novel* algorithms are indeed new or not.

We believe that the usage of such new metaphors should be limited to the cases in which they are indeed useful to express a new concept. This means that (i) it should not be possible to express the same algorithmic ideas using the terminology of already existing algorithms, and (ii) the inspiring metaphor should bring some new concepts that are related to the optimization process proposed. Unfortunately, this is often not the case. In our research we intend to examine a number of such *novel* nature-inspired algorithms to understand if they meet the two above-mentioned requirements and therefore deserve to be considered

M. Dorigo et al. (Eds.): ANTS 2018, LNCS 11172, pp. 302–314, 2018.
https://doi.org/10.1007/978-3-030-00533-7_24

novel. In this particular paper, we study the intelligent water drops (IWD) algorithm and its relation to the well-known ant colony optimization metaheuristic [16]. To do so, we first briefly present the ACO metaheuristic (Sect. 2) and the IWD algorithm (Sect. 3), highlighting their constituent components. Then, in Sect. 4, we perform a component-by-component comparison between ACO and IWD and show that IWD is indeed a particular case of ACO and that therefore it was not necessary to introduce a new terminology. We also discuss the fact that the inspiring metaphor does not bring any concepts that are related to the optimization process proposed. Therefore, the proposed IWD algorithm does not meet the two conditions set out in points (i) and (ii) above. Accordingly, we conclude that there is no need for an IWD algorithm and that adding it as a new tool to the optimization tool set is unnecessary and misleading.

2 Ant Colony Optimization

Ant colony optimization (ACO) is a metaheuristic that was first proposed in the early '90s [10,13,14]. The original source of inspiration was the foraging behavior of *Argentine* ants as described in a seminal paper by Deneubourg et al. [9]. In [9], it was shown that ants can find a shortest path between their nest and a food source by depositing pheromones on the ground and by choosing their way using a stochastic rule biased by pheromone intensity. In an analogous way, Dorigo et al. [10,13,14] showed that artificial agents, also called *artificial ants*, that

- move on a graph representation of a discrete optimization problem, where a path on the graph corresponds to a problem solution,
- deposit virtual pheromones on the graph edges, and
- use pheromones to bias the construction of random paths on the graph,

can find high quality solutions by letting their stochastic solution construction routine be biased by the value of virtual pheromones.

After the publication of the seminal algorithm in [13–15], many variants and improvements have been proposed [1,3–7,12,15,17,18,20,28,30]. Most of this work has been summarized in a book [16] where ACO is described as a constructive population-based metaheuristic comprising three main algorithmic components: (i) *stochastic solution construction*; (ii) *daemon actions*; and (iii) a *pheromone update procedure*.

One iteration of the ACO metaheuristic can be described as follows. First, every ant constructs a solution using a *stochastic solution construction* mechanism that iteratively selects solution components to add to the partial solution under construction. Once all ants have completed their solutions, an optional procedure called *daemon action* can be applied.[1] Finally, a *pheromone update procedure* modifies the pheromone trails.[2] Several iterations are executed until

[1] Daemon actions, for example, perform a local search procedure to improve an ant's solution or deposit an additional amount of pheromone on some solution components.

[2] In some ACO implementations, the *pheromone update procedure* can be interleaved with the solution construction (e.g., [12,17]), an example being the *local pheromone update procedure* that is implemented in ACS [12].

Algorithm 1. ACO metaheuristic

Set initial parameters
while termination condition not met **do**
  repeat
    Apply *stochastic solution construction*
    % solution components are iteratively added to a partial solution using a stochastic selection rule biased by artificial pheromones
    Apply *local pheromone update procedure* % optional
  until construction process is completed
  Apply *daemon actions* % optional
  Apply *pheromone update procedure*
end while
Return best solution

a termination condition is verified. An algorithmic outline of the ACO metaheuristic is shown in Algorithm 1.

Artificial pheromones, indicated by τ, are numerical values given to each of the solution components in the search space. They are iteratively modified by ants in order to bias the selection of solution components. Pheromone values can increase due to ants depositing pheromones (positive feedback) or decrease through evaporation (negative feedback). ACO algorithms also use heuristic information, indicated by η, to bias the solution construction process.

In Table 1 we summarize all the most important ACO algorithms. They differ in the way in which stochastic solution construction and pheromone update are implemented.

3 The Intelligent Water Drops Algorithm

The intelligent water drops (IWD) algorithm, published first by Shah-Hosseini in 2007 [25], was proposed as a *novel* nature-inspired algorithm for combinatorial optimization problems. This algorithm is explained using a metaphor in which water streams are seen as groups of individual particles (water drops) moving in discrete steps.

In the words of the author:

> *In the water drops of a river, the gravitational force of the earth provides the tendency for flowing toward the destination ... It is assumed that each water drop flowing in a river can carry an amount of soil. The amount of soil of the water drop increases while the soil of the riverbed decreases. In fact, some amount of soil of the river bed is removed by the water drop and is added to the soil of the water drop.*
> [26, pp. 195]
> *A water drop has also a velocity and this velocity plays an important role in the removing of soil from the bed of the rivers ... The faster water drops are assumed to gather more soil than others.*
> [26, pp. 196]

In computational terms, the intelligent water drops:

– move on a graph representation of a discrete optimization problem, where a path on the graph corresponds to a problem solution,
– modify the amount of soil on the graph edges as a function of their velocity,
– use soil amount to bias the construction of random paths on the graph.

Shah-Hosseini [25–27] has described IWD as a constructive population-based algorithm composed of three algorithmic components: (i) *stochastic solution construction*; (ii) *local soil update procedure*; and (iii) *global soil update procedure*.

One iteration of the IWD algorithm consists of the following steps. First, each water drop constructs a solution using a *stochastic solution construction* mechanism biased by the amount of soil associated to the solution components, so that components with lower soil values have a higher probability to be chosen. After a solution component is selected, a *local soil update procedure* performs two actions: (i) it decreases the soil in the solution component, which is, according to the metaphor, removed by the water drop, and (ii) it increases the soil in the water drop, which indicates that it has been loaded into the water drop. For this procedure to take place, each water drop keeps a record of its own velocity and soil gathered during the iteration. After each water drop has built a complete solution, a *global soil update procedure* updates the soil values using the *iteration-best* water drop (i.e., the water drop that built the best solution in the current iteration). Several iterations are performed before a termination criterion is met and the algorithm stops. Algorithm 2 depicts this process.

Algorithm 2. Intelligent water drops algorithm

```
1: Set initial parameters
2: while termination condition not met do
3:    repeat
4:       Apply stochastic solution construction
         % solution components are iteratively added to a partial solution using a
         stochastic selection rule biased on amount of soil
5:       Apply local soil update procedure
6:    until construction process is completed
7:    Apply global soil update procedure
8: end while
9: Return best solution
```

It is clear that in the IWD algorithm the *soil* variable plays the same role as *pheromone* in ACO: it represents the numerical information given to the solution components in order to bias their selection during the stochastic construction process. Differently from artificial ants in ACO, water drops have associated a velocity variable. The velocity is an independent property of each water drop, that is, for different solutions constructed different velocities are obtained. When one iteration starts, all water drops have the same initial velocity; however, the velocity of a water drop is updated as a function of the soil found in the edges it traverses while building a solution. The value of soil loaded in the water drops is non-linearly proportional to the *heuristic undesirability*,[3] that is, the inverse of the time needed for the water drops to move from one solution component to another.

4 Discussion

Algorithms 1 and 2 show the general structure of the ACO metaheuristic and of the IWD algorithm. Both are composed of the following three main algorithmic components:

– a *stochastic solution construction* mechanism to iteratively construct solutions biased by a quantity (pheromone/soil) associated to solution components,
– a *local update procedure* to improve the search interleaving the construction mechanism with a local update of pheromone/soil,
– a *global update procedure* to give a positive feedback via modifications of the pheromone/soil associated to specific solutions.

In this section, we present a detailed analysis of the two approaches comparing their algorithmic components in order to clarify if IWD is in fact a new algorithm and deserves to be called a novel approach or should rather be considered a variant of ACO. To this purpose, in Table 1 we schematically present the algorithmic components proposed in some of the best-known ACO variants: Ant System (AS) [13–15], Ant System with Q-learning (Ant-Q) [17], MAX-MIN Ant System (MMAS) [31], Ant Colony System (ACS) [12], Approximate Nondeterministic Tree-Search procedure (ANTS) [20]; and in IWD.

One difference between IWD and ACO is that in ACO pheromone values are always positive, while in IWD the value of soil progressively becomes negative. Unlike ACO pheromones, in IWD the soil is gradually removed by the water drops, which implies that additional mechanisms have to be introduced to manage negative and positive soil values as well as to avoid a possible division by zero.

Another difference is that IWD constructs solutions biased solely by the values of *soil*; that is, no problem-specific information is used to bias solution construction, as opposed to what is done in ACO with *heuristic information*.[4]

[3] The author calls *heuristic undesirability* the inverse of the *heuristic information* used in ACO. For example, in the travelling salesman problem the ACO *heuristic information* is commonly defined as $\eta_{ij} = 1/d_{ij}$, where d_{ij} refers to the distance between city i and city j. In IWD, the *heuristic undesirability* is, for the same problem, defined as $HUD_{ij} = d_{ij}$.

[4] The usage of heuristic information is a way to integrate problem-specific information in the stochastic solution construction procedure so as to stochastically favor solution components of lower cost.

As it is shown in the following, IWD's *local soil update* and *global soil update* are special cases of the components used to update pheromones in ACO. However, the function of these components in IWD is different from their typical function in ACO. The *local soil update* procedure is the most different due to the introduction of the water drop velocity and the soil removed from the riverbed, this latter computed using the linear motion equations and the heuristic undesirability.

4.1 Stochastic Solution Construction

Ants construct solutions adding new solution components with a probability computed using a *transition rule* (see second column of Table 1), that is, a function of the pheromone values and of the heuristic information. The transition rule not only states which information will be used by ants to choose the next solution component, but also how the relative importance of such information will be weighted.

The *stochastic solution construction* mechanism used in IWD is a particular case of the *random proportional rule* of AS proposed in [15], in which the parameters τ and η are weighted using $\alpha = -1$ and $\beta = 0$.

Equations 1 and 2 show the *transition rules* in AS and IWD respectively:

$$p_j^{ant} = \frac{[\tau_j]^\alpha \cdot [\eta_j]^\beta}{\sum\limits_{h \in N^f} [\tau_h]^\alpha \cdot [\eta_h]^\beta} \tag{1}$$

$$p_j^{iwd} = \frac{\frac{1}{\epsilon + g(soil_j)}}{\sum\limits_{h \in N^f} \left(\frac{1}{\epsilon + g(soil_h)}\right)} \tag{2}$$

where N^f is the set of feasible solution components and j is one solution component in the search space. The parameter ϵ is a small positive constant added to avoid a possible division by zero in Eq. 2.

From the equations, it can be seen that IWD uses a transition rule that includes only the information given by the *soil* (i.e., *heuristic information* is not used) and that $1/soil$ is used so as to favor solution components with a low soil level (as opposed to ACO variants which favor solution components with a high pheromone level).

Additionally, because the value of soil can become negative, IWD applies a function g to the value of soil to keep it positive in Eq. 2:

$$g(soil_j) = \begin{cases} soil_j & \text{if } \min\limits_{h \in N^f} soil_j \geq 0 \\ soil_j - \min\limits_{h \in N^f} soil_j & \text{otherwise} \end{cases} \tag{3}$$

Table 1. Main algorithmic components used in AS, ANT-Q, MMAS, ANTS, ACS and IWD. We do not show here the *daemon actions* component commonly integrated in ACO implementations. *Daemon actions* are optional problem-specific operations; for example, the application of a local search procedure. In the table, s^{k} is the solution built by ant k, s^{best} is the solution built by the ant that built the best solution, and iwd^{best} is the solution built by the water drop that built the best solution.

Algorithm	Stoch. solution constr.: Transition rule	Local update procedure	Global update procedure
AS	random proportional rule $\left[\frac{[\tau_j]^\alpha\cdot[\eta_j]^\beta}{\sum_{h\in N^f}[\tau_h]^\alpha\cdot[\eta_h]^\beta}\right]$, where N^f is the set of feasible component	ant density $[\tau_j + Q_1]$, ant quantity $[\tau_j + (Q_2 \cdot \eta_j)]$, where Q_1 and Q_2 are constants	ant cycle $\left[(1-\rho)\cdot\tau_j + \sum_{k=1}^{\text{ants}} \Delta\tau_j^k\right]$, where $\Delta\tau_j^k$ is defined as: $\begin{cases} F(k) & \text{if } j \in s^{\mathrm{k}} \\ 0 & \text{otherwise} \end{cases}$
ANT-Q	pseudo-random rule $\begin{cases} \arg\max_{h\in N^f}\{\tau_h^\alpha\cdot\eta_h^\beta\} & \text{if } q \le q_0 \\ S & \text{otherwise} \end{cases}$ where S is random value from a probability distribution given by τ_h^α and η_h^β, and q is random value from a uniform probability distribution	AQ-values learning rule $\left[(1-\alpha)\cdot\tau_j + \alpha\cdot[\Delta\tau_j + \gamma\cdot\max_{h\in N^f}\tau_h]\right]$, where $\Delta\tau_j$ is defined as: $\frac{W}{cost}$ where W is a constant and $cost$ is the solution cost	delayed reinforcement $\left[(1-\alpha)\cdot\tau_j + \alpha\cdot(\Delta\tau_j^{\mathrm{best}} + \gamma\cdot\max_{h\in N^f}\tau_h)\right]$, where $\Delta\tau_j^{\mathrm{best}}$ is defined as: $\begin{cases} F(s^{\mathrm{best}}) & \text{if } j \in s^{\mathrm{best}} \\ 0 & \text{otherwise} \end{cases}$
MMAS	random proportional rule [same as AS]	–	pheromones update rule $\left[\max\{\tau_{min}, \min\{\tau_{max}, (1-\rho)\cdot\tau_j + \Delta\tau_j^{\mathrm{best}}\}\}\right]$, where $\Delta\tau_j^{\mathrm{best}}$ is defined as: $\begin{cases} F(s^{\mathrm{best}}) & \text{if } j \in s^{\mathrm{best}} \\ 0 & \text{otherwise} \end{cases}$
ANTS	additive random proportional rule $\left[\frac{\alpha\cdot\tau_j + [1-\alpha]\cdot\eta_j}{\sum_{h\in N^f}\alpha\cdot\tau_h + [1-\alpha]\cdot\eta_h}\right]$	–	trail update $\left[\tau_j + \Delta\tau_j^k\right]$, where $\Delta\tau_j^k$ is defined as: $\left[\tau_0\cdot(1 - \frac{z_{curr} - \mathrm{LB}}{\overline{z} - \mathrm{LB}})\right]$, where $\overline{z}$ is the average cost of the solutions, z_{curr} is the current cost of the solution and LB is a lower bound for the problem
ACS	pseudo-random proportional rule $\begin{cases} \arg\max_{h\in N^f}\{\tau_h^\alpha\cdot\eta_h^\beta\} & \text{if } q \le q_0 \\ \text{[same as AS]} & \text{otherwise} \end{cases}$ where q is defined as in Ant-Q	local pheromone update $\left[(1-\varphi)\cdot\tau_j + \varphi\cdot\tau_0\right]$, where τ_0 is the pheromone lower bound	offline pheromones update $\begin{cases} (1-\rho)\cdot\tau_j + \rho\cdot\Delta\tau_j^{\mathrm{best}} & \text{if } j \in s^{\mathrm{best}} \\ \tau_j & \text{otherwise} \end{cases}$, where $\Delta\tau_j^{\mathrm{best}} = F(s^{\mathrm{best}})$
IWD	random selection $\left[\frac{\frac{1}{\epsilon + g(soil_j)}}{\sum_{h\in N^f}(\frac{1}{\epsilon + g(soil_h)})}\right]$	local soil update $\left[(1-\varphi)\cdot soil_j - \varphi\cdot\Delta soil_j\right]$, where $\Delta soil_j$ is defined in Equations 6,7 and 8	global soil update $\begin{cases} (1+\rho)\cdot soil_j - \rho\cdot\Delta soil_j^{\mathrm{best}} & \text{if } j \in iwd^{\mathrm{best}} \\ soil_j & \text{otherwise} \end{cases}$, where $\Delta soil_j^{\mathrm{best}} = F(iwd^{\mathrm{best}})$

4.2 Local Update Procedure

The *local pheromone update procedure* allows the artificial ants to give a negative or a positive feedback to other ants while constructing solutions[5] so as to avoid stagnation[6]. ACO variants implementing the idea of local negative feedback are, for example, Ant-Q [17] and ACS [12]. In the Ant-Q algorithm, pheromones are called AQ-values and the goal of the artificial ants is to learn these values (see AQ-values learning rule in Table 1) so that they can probabilistically favor better solution components.

IWD implements a variant of the *AQ-values learning rule* of Ant-Q, where parameter γ is set to $\gamma = 0$, and $\Delta soil_j$ is defined differently from $\Delta\tau_j$ (see Eqs. 6, 7 and 8). In fact, $\Delta soil_j$ is the only real difference between IWD and what had already been proposed in the context of the ACO metaheuristic. The implementation of this component in Ant-Q and IWD is shown in Eqs. 4 and 5, respectively:

$$\tau_j = (1-\alpha)\cdot\tau_j + \alpha\cdot\left[\Delta_{\tau_j} + \gamma\cdot\max_{h\in N^f}\tau_h\right] \tag{4}$$

$$soil_j = (1-\varphi)\cdot soil_j - \varphi\cdot\Delta soil_j \tag{5}$$

Equation 4 interpolates between the current pheromone value τ_j and the maximum pheromone over the possible next components; it simulates the change in the amount of pheromone due to evaporation and ant deposit. In Ant-Q, Equation 4 is used for both local and global reinforcement, the former applied after a solution component is selected during the solution construction, and the latter applied after the construction process finishes and all solutions are completed. However, in most Ant-Q implementations Δ_{τ_j} is defined as zero for the local pheromone update and as $1/cost^{best}$ for the global pheromone update [11,17,19].

Parameters α and γ are the learning step and the discount factor, respectively. The values chosen for these two parameters can favor the exploration or the exploitation behavior of the algorithm. The application of Eq. 4 can either enhance or reduce the exploration capabilities of Ant-Q by slightly reducing or increasing (depending on the values of γ and $\max\tau_h$) the pheromones. In a later ACO variant, ACS, a similar idea was proposed where $\gamma\cdot\max\tau_h$ was replaced by a small constant τ_0.

On the other hand, Eq. 5 intends to model the erosion of soil by water drops. In the metaphor of the IWD algorithm, water drops remove part of the soil every time a solution component is added. In practice, the *local soil update* procedure slightly increases the probability of one solution component to be selected by other water drops (in IWD lower soil values are preferred), thus implementing a

[5] Note that the idea of giving a positive feedback during the construction process was explored in some of the first ACO variants: *ant quantity* and *ant density* [10,13] However, these variants were abandoned many years ago because of their inferior performance compared with other ACO variants.

[6] Stagnation happens when the pheromones trails converge and all ants construct the same solutions over and over again.

form of positive feedback. The amount of soil removed by a water drop, $\Delta soil_j$, is computed using the *linear motion equations* of physics. As said in Sect. 3, different water drops have different velocities. The initial water drops velocity is a user selected parameter and its value should be selected empirically by running experiments on the considered problem. In fact, its value can greatly vary from problem to problem; for example, in [27], where the traveling salesman problem is considered, the water drops initial velocity is set 200, while in [26], where the problem considered is the multidimensional knapsack problem, the water drops initial velocity is set to 4. Once a solution component j has been added, the velocity of a water drop vel^{iwd} is updated according to

$$vel^{\text{iwd}} = vel^{\text{iwd}} + \frac{a_v}{b_v + c_v \times [soil_j]^2} \tag{6}$$

where a_v, b_v, c_v and $soil_j$ are also user selected parameters. The *time* required by the water drop to move from the current solution component to the next one is computed dividing the *heuristic undesirability* (HUD_j) by the water drop's new velocity. HUD_j represents the *distance* in the linear motion equation $V = d/t$, which, hence, becomes

$$time^{\text{iwd}} = \frac{HUD_j}{vel^{\text{iwd}}} \tag{7}$$

Finally, the amount of soil to be removed and to be loaded into the water drop is a function of the *time* taken by the water drop to move between the two solution components:

$$\Delta soil_j = \frac{a_s}{b_s + c_s \times [time^{iwd}]^2} \tag{8}$$

$\Delta soil_j$ tends to be larger for solution components with lower soil values or for those with small *heuristic undesirability*. In Eqs. 6 and 8, parameters b_v and b_s are used to avoid a possible division by zero. Typical values for the user selected parameters in the above equations are $a_v = 1, b_v = 0.01, c_v = 1$, $a_s = 1, b_s = 0.01, c_s = 1$, and for the initial value of soil 10 000 [27].

The velocity can be seen as an indicator of the quality of the partial solution constructed so far, that is, faster water drops have traversed edges with lower soil. However, putting the desirability of a solution component in terms of the velocity (quality of a partial solution) and of the heuristic information, as is defined for $\Delta soil_j$, is rather similar to the abandoned idea of *ant quantity* (see AS local update procedure in Table 1). Moreover, the *local soil update* component cannot be explained in terms of the inspiring metaphor. For example, if soil is removed, it is unclear why then the new amount of soil is computed by an equation such as Eq. 5 that uses a decay factor φ (and not simply by subtracting $\Delta soil_j$ from the current soil value). Additionally, the metaphor of water drops acting as individual particles removing the soil in the riverbeds is unrealistic, as water in a river should rather be seen as a moving fluid.

4.3 Global Update Procedure

The *global pheromone update procedure* in ACO is performed at the end of an iteration once all solutions have been completed. The main goal of this procedure is to give a *positive feedback* to the solution components included in a set of solutions that is used to deposit pheromones; common choices in ACO algorithms are using the *iteration-best* or *global-best* solution, but other options have been examined. Solution components that receive a higher amount of pheromone will have a higher probability of being selected by other ants in the next iterations.

The *global soil update* is a special case of the *offline pheromone update* in ACS, in which the parameter ρ has a range defined in the interval $[-1, 0]$, differently from its typical range defined in $(0, 1]$. Eqs. 9 and 10 show the definition of this component in ACO and IWD[7] respectively:

$$\tau_j = \begin{cases} (1-\rho) \cdot \tau_j + \rho \cdot \Delta\tau_j^{best} & \text{if } j \in s^{\text{best}} \\ \tau_j & \text{otherwise} \end{cases} \tag{9}$$

$$soil_j = \begin{cases} (1+\rho) \cdot soil_j - \rho \cdot \Delta soil_j^{best} & \text{if } j \in iwd^{\text{best}} \\ soil_j & \text{otherwise} \end{cases} \tag{10}$$

where the parameter $\Delta\tau_j^{best}$ is commonly defined as the inverse of the total cost of the solution $(1/cost^{best})$, while $\Delta soil_j^{best}$ is proportional to the soil gathered by the best water drop divided by the number of solution components $(soil^{best}/N^{best})$.

The *global soil update procedure*, as defined in [27], has two different outcomes depending on the value of $soil_j$ in the solution component. Let us consider the first summand in the first case in Eq. 10, $(1 + \rho) \cdot soil_j$. It is easy to see that if $soil_j > 0$, the resulting value of the first summand will be positive and therefore it will contribute with a *negative feedback* to the solution component. In the opposite case, when $soil_j < 0$, the product $(1 + \rho) \cdot soil_j$ will be negative and therefore the first summand will contribute with a *positive feedback* to the solution component. In other words, the first summand can either increase the value of soil if $soil_j > 0$, or decrease it if $soil_j < 0$. Regarding the second summand in the first case of Eq. 10, $-\rho \cdot \Delta soil_j^{best}$, the value of $\Delta soil_j^{best}$ is defined as always positive (see Eq. 8) and as we have it multiplied by $-\rho$, the result of this second summand will always be negative.

5 Conclusions

As the IWD algorithm, there are many other algorithms published as *novel* nature-inspired approaches in the metaheuristics literature. In fact, the already large number of these so-called *novel* approaches has made the selection of optimization algorithms troublesome, specially for those who use them for specific application problems and do not necessarily have a deep knowledge in the field

[7] There are two versions of this component in IWD. In [25], the first article proposing IWD, ρ was defined in the range $[0, 1]$ (just as in Eq. 9). However, in a later publication [27], the range of ρ was changed to $[-1, 0]$, leading to a somewhat different behavior of the update as explained here.

of metaheuristics. The very few existing, rigorous analyses of *novel* algorithms have shown in some selected cases that either (i) they simply re-use ideas proposed in the past [24,32], or that (ii) the scientific rationale behind the source of inspiration is incongruous and questionable [21,29].

In this paper, we contribute to such rigorous analyses by examining in more detail the Intelligent Water Drops (IWD) algorithm. In particular, we have shown that the algorithmic components proposed in IWD are not new and that they mainly have been proposed in the context of ant colony optimization (ACO) often already one or two decades earlier. More concretely, we found that the *stochastic construction mechanism* of IWD is a special case of the *random proportional rule* proposed in AS, the very first ACO algorithm. The *local soil update* component is a slight variant of the *AQ-values learning rule* that was proposed in the Ant-Q algorithm, a predecessor of ACS. The only, small, difference of IWD with earlier ACO algorithms is the definition of the $\Delta soil_j$ term in the local soil update; unfortunately, the rationale behind the definition of $\Delta soil_j$ and the definition of the *local soil update* component cannot be explained in terms of the source of inspiration of IWD. Finally, the *global update procedure* is a special case of the *offline pheromone update* proposed in ACS.

If we reconsider the two main criteria we have defined in the introduction, namely the fact that (i) it should not be possible to express the same algorithmic ideas using the terminology of already existing algorithms, and (ii) the inspiring metaphor should bring some new concepts that are related to the optimization process proposed, we can summarize the analysis of our article by saying that the IWD algorithm fails on both criteria.

Acknowledgments. Marco Dorigo and Thomas Stützle acknowledge support from the Belgian F.R.S.-FNRS, of which they are Research Directors.

References

1. Alaya, I., Solnon, C., Ghédira, K.: Ant colony optimization for multi-objective optimization problems. In: 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), vol. 1, pp. 450–457. IEEE Computer Society Press, Los Alamitos, CA (2007)
2. Askarzadeh, A.: Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. Commun. Nonlinear Sci. Numer. Simul. **19**(4), 1213–1228 (2014)
3. Birattari, M., Balaprakash, P., Dorigo, M.: The ACO/F-RACE algorithm for combinatorial optimization under uncertainty. In: Doerner, K.F., Gendreau, M., Greistorfer, P., Gutjahr, W.J., Hartl, R.F., Reimann, M. (eds.) Metaheuristics - Progress in Complex Systems Optimization, Operations Research/Computer Science Interfaces Series, vol. 39, pp. 189–203. Springer, New York (2006). https://doi.org/10.1007/978-0-387-71921-4
4. Blum, C.: Beam-ACO–hybridizing ant colony optimization with beam search: an application to open shop scheduling. Comput. Oper. Res. **32**(6), 1565–1591 (2005)
5. Blum, C., Dorigo, M.: The hyper-cube framework for ant colony optimization. IEEE Trans. Syst. Man, Cybern. - Part B **34**(2), 1161–1172 (2004)

6. Bullnheimer, B., Hartl, R.F., Strauss, C.: An improved ant system algorithm for the vehicle routing problem. Ann. Oper. Res. **89**, 319–328 (1999)
7. Cordón, O., de Viana, I.F., Herrera, F., Moreno, L.: A new ACO model integrating evolutionary computation concepts: the best-worst ant system. In: Dorigo, M., et al. (eds.) Abstract Proceedings of ANTS 2000 - From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms, pp. 22–29. IRIDIA, Université Libre de Bruxelles, Belgium, 7–9 September 2000
8. Cuevas, E., Miguel, C., Zaldívar, D., Pérez-Cisneros, M.: A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Syst. Appl. **40**(16), 6374–6384 (2013)
9. Deneubourg, J.L., Aron, S., Goss, S., Pasteels, J.M.: The self-organizing exploratory pattern of the Argentine Ant. J. Insect Behav. **3**(2), 159–168 (1990). https://doi.org/10.1007/BF01417909
10. Dorigo, M.: Optimization, Learning and Natural Algorithms. Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy (1992). (in Italian)
11. Dorigo, M., Gambardella, L.M.: A study of some properties of Ant-Q. In: Voigt, H.-M., Ebeling, W., Rechenberg, I., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 656–665. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61723-X_1029
12. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. **1**(1), 53–66 (1997)
13. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System: An autocatalytic optimizing process. Technical report, 91–016 Revised, Dipartimento di Elettronica, Politecnico di Milano, Italy (1991)
14. Dorigo, M., Maniezzo, V., Colorni, A.: Positive feedback as a search strategy. Technical report, 91–016, Dipartimento di Elettronica, Politecnico di Milano, Italy (1991)
15. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. - Part B **26**(1), 29–41 (1996)
16. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge, MA (2004)
17. Gambardella, L.M., Dorigo, M.: Ant-Q: a reinforcement learning approach to the traveling salesman problem. In: Proceedings of the Twelfth International Conference on Machine Learning, ML 1995, pp. 252–260. Morgan Kaufmann Publishers, Palo Alto (1995)
18. Guntsch, M., Middendorf, M.: A population based approach for ACO. In: Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R. (eds.) EvoWorkshops 2002. LNCS, vol. 2279, pp. 72–81. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46004-7_8
19. Machado, L., Schirru, R.: The Ant-Q algorithm applied to the nuclear reload problem. Ann. Nucl. Energy **29**(12), 1455–1470 (2002)
20. Maniezzo, V.: Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. INFORMS J. Comput. **11**(4), 358–369 (1999)
21. Melvin, G., Dodd, T.J., Groß, R.: Why 'GSA: a gravitational search algorithm' is not genuinely based on the law of gravity. Natural Comput. **11**(4), 719–720 (2012)
22. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016)
23. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)

24. Piotrowski, A.P., Napiorkowski, J.J., Rowinski, P.M.: How novel is the "novel" black hole optimization approach? Inf. Sci. **267**, 191–200 (2014)
25. Shah-Hosseini, H.: Problem solving by intelligent water drops. In: Proceedings of the 2007 Congress on Evolutionary Computation, CEC 2007, pp. 3226–3231. IEEE Press, Piscataway (2007)
26. Shah-Hosseini, H.: Intelligent water drops algorithm: a new optimization method for solving the multiple knapsack problem. Int. J. Intell. Comput. Cybern. **1**(2), 193–212 (2008)
27. Shah-Hosseini, H.: The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. Int. J. Bio-Inspired Comput. **1**(1–2), 71–79 (2009)
28. Socha, K., Dorigo, M.: Ant colony optimization for continuous domains. Eur. J. Oper. Res. **185**(3), 1155–1173 (2008). https://doi.org/10.1016/j.ejor.2006.06.046
29. Sörensen, K.: Metaheuristics–the metaphor exposed. Int. Trans. Oper. Res. **22**(1), 3–18 (2015). https://doi.org/10.1111/itor.12001
30. Stützle, T., Hoos, H.H.: The $\mathcal{MAX}-\mathcal{MIN}$ and local search for the traveling salesman problem. In: Bäck, T., Michalewicz, Z., Yao, X. (eds.) Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, ICEC 1997, pp. 309–314. IEEE Press, Piscataway (1997)
31. Stützle, T., Hoos, H.H.: $\mathcal{MAX}--\mathcal{MIN}$ Ant system. Future Gener. Comput. Syst. **16**(8), 889–914 (2000)
32. Weyland, D.: A rigorous analysis of the harmony search algorithm: how the research community can be misled by a "novel" methodology. Int. J. Appl. Metaheuristic Comput. **12**(2), 50–60 (2010)