Swarm Intelligence (2019) 13:173–192
https://doi.org/10.1007/s11721-019-00165-y

The intelligent water drops algorithm: why it cannot be considered a novel algorithm

A brief discussion on the use of metaphors in optimization

Christian Leonardo Camacho-Villalón[1] · Marco Dorigo[1] · Thomas Stützle[1]

Received: 16 December 2018 / Accepted: 2 May 2019 / Published online: 14 May 2019

Abstract
In this article, we rigorously analyze the intelligent water drops (IWD) algorithm, a metaphor-based approach for the approximate solution of discrete optimization problems proposed by Shah-Hosseini (in: Proceedings of the 2007 congress on evolutionary computation (CEC 2007), IEEE Press, Piscataway, NJ, pp 3226–3231, 2007). We demonstrate that all main algorithmic components of IWD are simplifications or special cases of ant colony optimization (ACO), and therefore, IWD is simply a particular instantiation of ACO. We show that the natural metaphor of *"water drops flowing in rivers removing the soil from the riverbed"*, the source of inspiration of IWD, is unnecessary, misleading and based on unconvincing assumptions of river dynamics and soil erosion that lack a real scientific rationale. We carry out a detailed review of modifications and extensions proposed to IWD since its first publication in 2007. We find that research on IWD is for the most part misguided and that the vast majority of the ideas explored in the literature on IWD have been studied many years before in the context of ACO. Finally, we discuss the use of natural metaphors as a source of inspiration for optimization algorithms, which has become an extremely popular trend in the last 15 years, and propose some criteria to limit their usage to the cases in which the metaphor is indeed useful.

Keywords Intelligent water drops · Ant colony optimization · Novel algorithm · Metaphor-based algorithm

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11721-019-00165-y) contains supplementary material, which is available to authorized users.

✉ Christian Leonardo Camacho-Villalón
ccamacho@ulb.ac.be

Marco Dorigo
mdorigo@ulb.ac.be

Thomas Stützle
stuetzle@ulb.ac.be

[1] IRIDIA, Université Libre de Bruxelles, Brussels, Belgium

1 Introduction

Over the last 15 years, many so-called *novel* approaches to stochastic optimization, developed using metaphors from natural and artificial systems, have been proposed in the literature. Most of these approaches belong to one of the three categories, depending on the source of inspiration: (i) algorithms using metaphors from living organism, such as *spiders* (Cuevas et al. 2013), *fireflies* (Yang 2009) *whales* (Mirjalili and Lewis 2016), *gray wolves* (Mirjalili et al. 2014), *birds* (Askarzadeh 2014), *bacteria* (Passino 2002), *flower pollination* (Yang 2012), and *invasive weed* (Mehrabian and Lucas 2006); (ii) algorithms using metaphors from physical, chemical or other natural phenomena, such as *biogeography* (Simon 2008), *electromagnetism* (Birbil and Fang 2003), *black holes* (Hatamlou 2013), *river erosion* (Shah-Hosseini 2007), *the big bang* (Erol and Eksin 2006), *electricity* (Kaveh and Talatahari 2010), *water cycle* (Eskandar et al. 2012), and *wheel spinning* (Joslin and Clements 1999); and (iii) algorithms using metaphors from different aspects human behavior, such as *musicians* (Geem et al. 2001), *political imperialism* (Atashpaz-Gargari and Lucas 2007), *teachers* (Rao et al. 2011), *social behavior* (Ray and Liew 2003), *interior design and decoration* (Gandomi 2014), and so on (see Campelo 2017) . However, the real value of using such metaphors in optimization algorithms is often unclear and has recently prompted a number of criticisms (e.g., see Sörensen 2015).

One initial criticism is that, in the absence of a common framework to classify metaphor-based algorithms using standard optimization and computational terminology, the mainstream approach has been to differentiate them in terms of the inspiring metaphor and terminology employed, which has shown to be very ineffective to pinpoint real differences at an algorithmic level. In fact, very often the metaphoric terms used by the authors hide the fact that there are no algorithmic differences with previously published work. There are a few papers, in which this has been shown explicitly. For example, Weyland (2010) has shown that harmony search is a simplification of evolutionary strategies; Piotrowski et al. (2014) have shown that black holes optimization is a variant of particle swarm optimization; and Simon et al. (2011) have shown that biogeography-based optimization is a generalization of genetic algorithms.

There are also important criticisms about whether the use of many of the proposed metaphors is useful at all in the development of optimization algorithms. A few analyses (Melvin et al. 2012; Piotrowski et al. 2014; Sörensen 2015) have found that often, although the inspiration source comes from a well-understood phenomenon described by mathematical models, in the corresponding proposed algorithm such models are modified, or even completely forgotten, so that the algorithm does not match the inspiring metaphor anymore. On the other hand, for many other inspiring metaphors there is no scientific model at all, and the algorithm uses unrealistic oversimplifications of the inspiring phenomenon. Thus, the pertinence of using a metaphor to guide the design of an algorithm becomes questionable if the inspirations provided by the metaphor are used in an inaccurate way or are wrongly translated—in some cases, even simply omitted—in the resulting algorithm.

As it has been discussed in great detail by Sörensen (2015), even though in the vast majority of cases metaphors are completely unnecessary to describe the metaphor-based algorithm, their role is frequently overstated by their authors. Unfortunately, very often not only the added value of using them is unclear, but their usage can also be misleading. In general, in the absence of a taxonomy that allows classifying the many different proposed metaphor-based algorithms, it remains quite challenging to understand whether these *“novel”* algorithms are indeed new or not—for the time being, formal or empirical analyses of these algorithms can only be done case by case.

In this paper, we study the intelligent water drops (IWD) algorithm and its relation with the well-known ant colony optimization metaheuristic (ACO, Dorigo and Stützle 2004). We start by describing the two approaches, the ACO metaheuristic in Sect. 2 and the IWD algorithm in Sect. 3, discussing their source of inspiration and highlighting their algorithmic components. Then, in Sect. 4, we perform a component-by-component comparison between the two approaches and we show that it is neither useful nor necessary to introduce the new metaphor and the new terminology used by IWD, as IWD is indeed a particular case of ACO. We also discuss the fact that the optimization process proposed does not bring any concepts that are related to optimization and is in fact based on unconvincing assumptions about river systems dynamic and erosion. In Sect. 5, we review published research on IWD. We provide compelling evidence that most of the ideas proposed to enhance the performance of IWD were already proposed in the previously published research on ACO.

In Sect. 6, we propose some criteria that should be verified to ensure that the use of a metaphor to present a new algorithm is meaningful and we discuss the fact that the proposed IWD algorithm does not meet any of these conditions. Finally, in Sect. 7 we draw some conclusions, the most important one being that adding the IWD algorithm to the optimization tool set is unnecessary and misleading.

2 Ant colony optimization

Ant colony optimization (ACO) is a metaheuristic proposed in the early 1990s by Dorigo (1992) and Dorigo et al. (1991a, b) inspired by Deneubourg et al. (1990)'s seminal work on the *Argentine* ant foraging behavior. Denebourg et al. showed that *Argentine* 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 their perceived pheromone intensity.

Based on Deneubourg's findings, Dorigo et al. showed that, in analogous way to real ants, artificial ants that

- move on a graph representation of a discrete optimization problem, where edges are solution components and where a path on the graph corresponds to a problem solution,
- deposit virtual pheromones on the graph edges (or equivalently on solution components), 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. The first algorithm proposed based on the metaphor of ants foraging behavior (Dorigo et al. 1991a, 1996) was a combination of many interactive agents, also called *artificial ants*, a reinforcement mechanism to give a *positive feedback* to selected solution component,[1] and a constructive greedy heuristic to build paths on a graph.

The publication of the seminal algorithm (Dorigo et al. 1991a, b; Dorigo 1992; Dorigo et al. 1996) was followed by many variants and improvements (Gambardella and Dorigo 1995; Dorigo et al. 1996; Dorigo and Gambardella 1997b; Stützle and Hoos 1997; Bullnheimer et al. 1999; Maniezzo 1999; Cordón et al. 2000; Guntsch and Middendorf 2002; Blum and

[1] As explained in Dorigo et al. (1991a), *positive feedback* allows ants to generate a process that reinforces itself, that is, the higher the number of ants following a trail, the more attractive that trail becomes for being followed.

Dorigo 2004; Blum 2005; Birattari et al. 2006; Alaya et al. 2007; Socha and Dorigo 2008); most of these works are summarized in the *Ant Colony Optimization* book (Dorigo and Stützle 2004). Throughout all these studies, ACO has been described as a constructive population-based metaheuristic composed of three main algorithmic components: (i) *stochastic solution construction*, involving the routines needed to construct solutions; (ii) *daemon actions*, containing optional routines to improve the solutions constructed by the ants; and (iii) *pheromone update*, involving the routines to modify the pheromone trails in order to ensure the exploration and exploitation of the search space. The main idea is that artificial ants are probabilistic procedures that construct solutions incrementally—that is, adding one solution component at a time to a partial solution—biased by virtual pheromones and heuristic information.

One iteration of ACO can be described as follows. Starting from an empty solution, an artificial ant implements the stochastic solution construction routines needed to add solution components until the solution is completed.[2] After the construction phase is over, daemon actions may take place. Daemon actions are routines that cannot be performed by a single ant. They may consist, for example, of a local search procedure that improves the solution constructed by an ant; or of a procedure that deposits an additional amount of pheromone on solution components that belong to solutions with some desirable characteristics. Finally, pheromone update consists in the modification of the pheromones with the goal to bias the construction process in the following iterations toward better solutions; pheromone update involves depositing pheromone on some selected solution components, generally those belonging to good-quality solutions, and evaporating pheromone in components producing solutions of lower quality.[3] Several iterations are executed until a termination condition is verified, and the algorithm stops. This process is shown in Algorithm 1.

Algorithm 1 ACO metaheuristic

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

As said above, virtual pheromones—pheromones for short in the following—and heuristic information are the main sources of information used by artificial ants to construct solutions stochastically. Pheromones, indicated by τ, are numerical values associated with solution components that are iteratively modified by ants in order to mark solution components that produce good solutions. The amount of pheromones in the solution components can be increased by *pheromone deposit*, or decreased through *pheromone evaporation*. While

[2] The rule for selecting solution components, called *transition rule*, implemented during the stochastic solution construction varies among ACO variants.

[3] In some ACO implementations, the *pheromone update* can be interleaved with the solution construction (e.g., see (Gambardella and Dorigo 1995; Dorigo and Gambardella 1997b)), an example being the *offline pheromone update* implemented in ACS (Dorigo and Gambardella 1997b).

pheromones represents the knowledge acquired during the algorithm's execution, the *heuristic information*, indicated by η and also associated with solution components, is a way to include problem-specific information to guide the search. The use of heuristic information greedily biases the selection of components that have a lower cost in the solution under construction. There are different strategies to weight the relative importance of parameters τ and η; we discuss some of them in Sect. 4.1.

In Table 1, we list 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 was proposed by Shah-Hosseini (2007) as a new problem solving algorithm for combinatorial optimization. According to the author, the algorithm is based on the observation of rivers in nature and is explained using a metaphor in which water streams are seen as groups of individual particles (water drops) removing soil from the ground over which the river flows, that is, the riverbed. In their journey from a source to a destination, in the IWD algorithm metaphor water drops prefer paths with less soil; also, on paths with less soil they move faster, and the faster they move the more soil they remove. Following this self-reinforced mechanism, the water drops are capable of finding shortest paths from a source to a destination.

In the words of the author:

> *In nature, we often see water drops moving in rivers, lakes and seas. As water drops move, they change their environment in which they are flowing . . . We also know that the water drops have no visible eyes to be able to find the destination (lake or river). If we put ourselves in place of a water drop of the river, we feel that some force pulls us toward itself (gravity).*
> (Shah-Hosseini 2007, pp. 3326)
> *In the water drops of a river, the gravitational force of the earth provides the tendency for flowing toward the destination. If there were no obstacles or barriers, the water drops would follow a straight path toward the destination, which is the shortest path from the source to the destination. However, due to different kinds of obstacles in their way to the destination, which constrain the path construction, the real path has to be different from the ideal path and lots of twists and turns in the river path is observed.*
> (Shah-Hosseini 2008, pp. 195)
> *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.*
> (Shah-Hosseini 2008, 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 collect more soil than others.*
> (Shah-Hosseini 2008, pp. 196)

Shah-Hosseini (2007, 2008, 2009) translated these ideas into an algorithm where water drops: (i) move in discrete steps on a graph representation of the considered optimization problem where edges are solution components and where each solution component j has an associated amount soil_j of soil; (ii) modify the amount of soil on the solution components (graph edges) as a function of their velocity and of problem-specific information called *heuristic undesirability*; and (iii) use the amount of soil associated with solution components to bias the construction of random paths.

The algorithm, as described in (Shah-Hosseini 2007, 2008, 2009), is a constructive population-based algorithm composed of three algorithmic components: (i) *stochastic solution construction*, (ii) *local soil update*, and (iii) *global soil update*. In the IWD algorithm, the water drops have two associated variables: *velocity* and total amount of *soil collected* and cooperatively construct solutions incrementally using a probabilistic rule, called *random selection rule*, biased by the amount of soil associated with solution components.

In one iteration of IWD, these three algorithmic components are applied as follows. First, during stochastic solution construction, each water drop starts from an empty solution and adds one solution component at a time until the solution is completed. Interleaved with stochastic solution construction, local soil update involves two actions after a solution component is added to a partial solution: (i) a decrease in the soil in the added solution component, and (ii) an increase in the soil collected in the water drop. In fact, every time a water drop adds a new solution component to the solution it is constructing, it updates its velocity and its total amount of soil collected. Finally, global soil update updates the soil in solution components of the iteration-best water drop (i.e., the water drop that built the best solution in the current iteration). The algorithm stops once a termination criterion is met. A high-level description of the algorithm is given in Algorithm 2.

Algorithm 2 Intelligent water drops algorithm

```
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 on amount of soil
        Apply local soil update
    until construction process is completed
    Apply global soil update
end while
Return best solution
```

It is clear that in the IWD algorithm, the soil associated with solution components plays the same role as pheromone in ACO: it biases the stochastic choice of solution components during the stochastic solution construction process. However, as we said and differently from artificial ants in ACO, each water drop k has two associated variables: the velocity vel^k, that represents the quality of the partial solution that the water drop has built and collected_soil^k that memorizes the soil collected by the water drop while building a solution. All water drops start one iteration of the algorithm with the same initial velocity. However, the velocity of water drops is updated as a function of the soil found in the components added to the partial solution under construction. Therefore, different water drops end up having different velocities. The velocity of a water drop is used to compute the amount of soil that the water drop collects when adding a new solution component.[4]

[4] Note that, even though this is highly counter-intuitive, the amount of soil that a water drop collects when adding a new solution component to the partial solution it is building is different from the amount of soil that

The variable collected_soil^k is used by the water drop to keep a record of the soil collected from the solution components added to the solution that it is constructing. The amount of soil added to collected_soil^k is also proportional to a value called *heuristic undesirability*[5] divided by the water drop velocity.

As mentioned above, the best water drop updates the solution components at the end of each iteration using the amount of soil collected in its associated variable collected_soil^{best}.

4 A comparison between ACO and IWD

In the previous two sections, we have described the ACO metaheuristic and the IWD algorithm starting with a description of their sources of inspiration and showing how these sources of inspiration were translated into algorithmic ideas to be used in optimization problems. In summary, the two techniques consist of the following three main algorithmic components:

- *stochastic solution construction*: to construct solutions biased by a quantity (pheromone/soil) associated with solution components,
- *local update*: to improve the search by interleaving the construction mechanism with an update of pheromone/soil on the last added solution component,
- *global update*: to provide a positive feedback via modifications of the pheromone/soil associated with specific solution components.

In this section, we present a detailed comparison of the two optimization techniques in order to clarify whether IWD is in fact a new algorithm, and deserves therefore to be called a novel approach, or should rather be considered a variant of ACO. To do so, in Table 1 we schematically present the algorithmic components proposed in some of the best-known ACO variants: Ant System (AS) (Dorigo et al. 1991a, b, 1996), Ant System with Q-learning (Ant-Q) (Gambardella and Dorigo 1995), $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) (Stützle and Hoos 2000), Ant Colony System (ACS) (Dorigo and Gambardella 1997b), Approximate Nondeterministic Tree-Search (ANTS) (Maniezzo 1999); and in IWD (Shah-Hosseini 2009).

Before moving to the component-by-component comparison of ACO and IWD, we briefly discuss the notions of soil and of water drop's velocity. This discussion will help the reader understand the analysis presented in this section.

In Sect. 3, we noted that the role played by the soil value associated with solution components in IWD is very similar to the one played by pheromones in ACO. However, on good solution components the value of soil tends to decrease over time while the value of pheromone tends to increase. Therefore, more attractive components are characterized by low soil values in IWD and by large pheromone values in ACO. The comparison of IWD and ACO is therefore prone to confusion. In practice, the difference is that the construction of solutions in ACO is biased toward higher pheromone values, while in IWD it is biased toward lower soil values. Additionally, in IWD soil can become negative in high-quality solution components, while in ACO it is always strictly positive.

The other concept that deserves some attention is the water drop's velocity, introduced in Sect. 3. This concept does not exist in ACO and, as mentioned in Sect. 3, it comes from the

is removed from the soil variable associated with the added component. For a detailed example, see the online supplementary material.

[5] The author calls *heuristic undesirability* to the inverse of the *heuristic information* used in ACO. For example, in the traveling salesman problem, ACO's *heuristic information* is commonly defined as $\eta_{ij} = 1/d_{ij}$, where d_{ij} indicates the distance between city i and city j. In IWD, the *heuristic undesirability* is, for the same problem, defined as $\text{HUD}_{ij} = d_{ij}$.

analogy with rivers, where water drops move with a certain velocity in the riverbed removing soil. Leaving aside the metaphor, in IWD the velocity vel^k of a water drop k is used to compute the amount of soil Δsoil_j that the water drop collects when adding a new solution component j and it is updated according to Eq. 1:

$$\text{vel}^k = \text{vel}^k + \frac{a_v}{b_v + c_v \times [\text{soil}_j]^2}, \tag{1}$$

where a_v, b_v, c_v are user-selected parameters.

Water drops that select "good" components (i.e., components that have a low soil value) tend to be faster, and therefore, a water drop's velocity somehow measures the quality of the partial solution under construction. A water drop's velocity determines the extent to which the soil will be decreased after a solution component is added to a water drop's partial solution: faster water drops remove more soil from the added solution components than slower water drops. As a consequence, since solution components with less soil have a higher probability of being selected by another water drop, velocity is also a way to control the exploration–exploitation capabilities of the algorithm. This can be done by selecting the initial value of the water drops velocity. If a low initial velocity is chosen, water drops will tend to have a more exploratory behavior, while if a high initial velocity is chosen they will tend to exploit more the soil information.

In the next three subsections, we will compare the *stochastic solution construction*, the *local update* and *global update* algorithmic components used in IWD with those used in some of the ACO algorithms proposed in the literature. In particular, we will show that the stochastic solution construction rule used by IWD—called *random selection rule*—is a simplification of the stochastic solution construction rule implemented in Ant System (Dorigo et al. 1991b; Dorigo 1992; Dorigo et al. 1996)—called *random proportional rule*; and that the *local update* and *global update* used in IWD—called *local soil update* and *global soil update*, respectively—are special cases of Ant-Q's *local reinforcement* (Gambardella and Dorigo 1995) and of Ant Colony System's *global pheromone trail updating rule* (Gambardella and Dorigo 1996; Dorigo and Gambardella 1996, 1997a).

4.1 Stochastic solution construction

Ants construct solutions by adding new components probabilistically chosen using a function of the pheromone values and of the heuristic information. We refer to this function as *transition rule* (see second column of Table 1). 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.

For example, in the transition rule of AS (Dorigo et al. 1996), the weighting strategy consists in using two parameters α and β that modulate the value of τ and η, respectively; in ANTS (Maniezzo and Carbonaro 2000), a parameter $\alpha \in [0, 1]$ allows changing the relative importance of τ and η in the transition rule (see Table 1).

Equations 2 and 3 are the transition rules used in Ant System and in IWD:

$$p_j^k = \frac{[\tau_j]^\alpha \cdot [\eta_j]^\beta}{\sum\limits_{h \in N^f} [\tau_h]^\alpha \cdot [\eta_h]^\beta}, \tag{2}$$

$$p_j^k = \frac{\frac{1}{\epsilon + g(\text{soil}_j)}}{\sum\limits_{h \in N^f} \left(\frac{1}{\epsilon + g(\text{soil}_h)}\right)}. \tag{3}$$

Table 1 Main algorithmic components used in Ant System (AS), Ant-Q, $\mathscr{MAX}$–$\mathscr{MIN}$ Ant System ($\mathscr{MM}$AS), Approximate Nondeterministic Tree-Search (ANTS), Ant Colony System (ACS) and intelligent water drops (IWD)

Algorithm	Transition rule	Local update	Global update
AS	*Random proportional rule*: $p_j^k = \frac{[\tau_j]^\alpha \cdot [\eta_j]^\beta}{\sum_{h \in N^f} [\tau_h]^\alpha \cdot [\eta_h]^\beta}$	*Ant density*: $\tau_j = \tau_j + Q_1$, *ant quantity*: $\tau_j = \tau_j + (Q_2 \cdot \eta_j)$, where Q_1 and Q_2 are constants	*Ant cycle*: $\tau_j = (1-\rho) \cdot \tau_j + \sum_{k=1}^{m} \Delta\tau_j^k$, where $\Delta\tau_j^k$ is defined as: $\begin{cases} F(k) & \text{if } j \in s^k \\ 0 & \text{Otherwise} \end{cases}$
Ant-Q	*Pseudo-random proportional rule*: $p_j^k = \begin{cases} \arg\max_{h \in N^f} \{\tau_h^\alpha \cdot \eta_h^\beta\} & \text{if } q \leq q_0 \\ \frac{[\tau_j]^\alpha \cdot [\eta_j]^\beta}{\sum_{h \in N^f} [\tau_h]^\alpha \cdot [\eta_h]^\beta} & \text{otherwise} \end{cases}$ where q is a uniformly distributed random value	*Local reinforcement*: $\tau_j = (1-\alpha) \cdot \tau_j + \alpha \cdot [\Delta_{\tau_j} + \gamma \cdot \max_{h \in N^f} \tau_h]$, where Δ_{τ_j} is defined as: $\frac{W}{cost}$, W is a constant and *cost* is the solution cost.	*Global reinforcement*: $\tau_j = (1-\alpha) \cdot \tau_j + \alpha \cdot (\Delta\tau_j^{\text{best}} + \gamma \cdot \max_{h \in N^f} \tau_h)$, where $\Delta\tau_j^{\text{best}}$ is defined as: $\begin{cases} F(s^{\text{best}}) & \text{if } j \in s^{\text{best}} \\ 0 & \text{otherwise} \end{cases}$
$\mathscr{MM}$AS	*Random proportional rule*: same as AS	—	*Pheromones update rule*: $\tau_j = \max\{\tau_{min}, \min\{\tau_{max}, (1-\rho) \cdot \tau_j + \Delta\tau_j^{\text{best}}\}\}$, where $\Delta\tau_j^{\text{best}}$ is defined as: $\begin{cases} F(s^{\text{best}}) & \text{if } j \in s^{\text{best}} \\ 0 & \text{otherwise} \end{cases}$
ANTS	*Additive random proportional rule*: $p_j^k = \frac{\alpha \cdot \tau_j + [1-\alpha] \cdot \eta_j}{\sum\limits_{h \in N^f} \alpha \cdot \tau_h + [1-\alpha] \cdot \eta_h}$	—	*Trail update*: $\tau_j = \tau_j + \Delta\tau_j^k$, where $\Delta\tau_j^k$ is defined as: $\tau_0 \cdot (1 - \frac{z_{curr} - LB}{\overline{z} - LB})$, $\overline{z}$ is the average cost of the solutions, z_{curr} is the cost of the current solution and LB is a lower bound for the problem solution cost

Table 1 continued

Algorithm	Transition rule	Local update	Global update
ACS	*Pseudo-random proportional rule*: $p_j^k = \begin{cases} \arg\max_{h \in N^f} \{\tau_h^\alpha \cdot \eta_h^\beta\} & \text{if } q \leq q_0 \\ \frac{[\tau_j]^\alpha \cdot [\eta_j]^\beta}{\sum_{h \in N^f} [\tau_h]^\alpha \cdot [\eta_h]^\beta} & \text{otherwise} \end{cases}$ where q is a uniformly distributed random value	*Local pheromone update rule*: $\tau_j = (1 - \varphi) \cdot \tau_j + \varphi \cdot \tau_0$, where τ_0 is the pheromone lower bound	*Global pheromone trail updating rule*: $\tau_j = \begin{cases} (1 - \rho) \cdot \tau_j + \rho \cdot \Delta\tau_j^{\text{best}} & \text{if } j \in s^{\text{best}} \\ \tau_j & \text{otherwise} \end{cases},$ where $\Delta\tau_j^{\text{best}} = F(s^{\text{best}})$
IWD	*Random selection rule*: $p_j^k = \frac{\frac{1}{\epsilon + g(\text{soil}_j)}}{\sum\limits_{h \in N^f} \left(\frac{1}{\epsilon + g(\text{soil}_h)}\right)}$	*Local soil update*: $\text{soil}_j = (1 - \varphi) \cdot \text{soil}_j - \varphi \cdot \Delta\text{soil}_j$, where Δsoil_j is defined in Eq. 7	*Global soil update*: $\text{soil}_j = \begin{cases} (1 + \rho) \cdot \text{soil}_j - \rho \cdot \Delta\text{soil}_j^{\text{best}} & \text{if } j \in s^{\text{best}} \\ \text{soil}_j & \text{otherwise} \end{cases},$ where $\Delta\text{soil}_j^{\text{best}} = F(s^{\text{best}})$

We do not show here the *daemon actions* component commonly integrated in ACO implementations. In the algorithmic components presented here under, s^k indicates the solution constructed by ant/water drop k, s^{best} indicates the best solution constructed either in the last iteration or from the beginning of the execution of the algorithm (using one or the other varies among ACO algorithm, some of them, as $\mathscr{M}\mathscr{M}$AS, include both options). The function $F(\cdot)$ is specific to the considered problem and is used to compute the amount of pheromone to be deposited on the solution components

In the equations, N^f is the set of feasible solution components that can still be added to the partially built solution, $j \in N^f$ is a solution component in the search space and k is one of the m ants/water drops building a solution. The parameter ϵ in Eq. 3 is a small positive constant added to avoid a possible division by zero.

It is easy to see that by setting $\alpha = -1$ and $\beta = 0$ in the transition rule of Ant System, IWD reduces to ACO. Note that IWD uses a transition rule that includes only the information given by the soil (i.e., heuristic information is not used).

Additionally, because the value of soil can become negative in the solution components, IWD applies a function $g(\cdot)$ to soil_j so that its value in Eq. 3 remains positive:

$$g(\text{soil}_j) = \begin{cases} \text{soil}_j & \text{if } \min\limits_{h \in N^f} \text{soil}_j \geq 0, \\ \text{soil}_j - \min\limits_{h \in N^f} \text{soil}_j & \text{otherwise.} \end{cases} \tag{4}$$

In both ACO and IWD, the initial value of pheromone/soil and other parameters, such as m, the number of ants/water drops, or the value of α, β, are user-selected parameters that have to be chosen according to the problem considered.[6]

4.2 Local update

The local pheromone update allows ants to update the pheromones not only after having built a complete solution, but also while constructing it.

An ACO variant implementing local pheromone update is Ant-Q (Gambardella and Dorigo 1995). In Ant-Q's local pheromone update, pheromones are updated immediately after a component is added to a partial solution using the formula shown in Eq. 5. Comparing Eq. 5 with IWD's local soil update given in Eq. 6, we can see that the two updates are very similar:

$$\tau_j = (1 - \alpha) \cdot \tau_j + \alpha \cdot \left[\Delta_{\tau_j} + \gamma \cdot \max_{h \in N^f} \tau_h\right] \tag{5}$$

$$\text{soil}_j = (1 - \varphi) \cdot \text{soil}_j - \varphi \cdot \Delta\text{soil}_j^k \tag{6}$$

In particular, if we set the value of $\gamma = 0$ in Eq. 5, the two equations become virtually identical.

However, while α, Δ_{τ_j}, γ, and φ are fixed parameters, the value of Δsoil_j^k in IWD has to be computed using Eq. 7, involving the velocity vel^k of the water drop and the *heuristic undesirability* (HUD_j) of the solution component j that is being added. Δsoil_j^k is computed for every water drop after a solution component is added to the partial solution the water drop is constructing. First, the water drop k updates its velocity vel^k according to Eq. 1, and then, Δsoil_j^k is computed using Eq. 7:

$$\Delta\text{soil}_j^k = \frac{a_s}{b_s + c_s \cdot [\text{HUD}_j / \text{vel}^k]^2}, \tag{7}$$

where a_s, b_s, c_s are user-selected parameters.

Therefore, the value of Δsoil_j^k tends to be larger for solution components with lower soil (because of the velocity update of Eq. 1) and for those with low HUD_j. In Eqs. 1 and 7, parameters b_v and b_s are used to avoid a possible division by zero.

[6] Finding values for the parameters of stochastic algorithms that guarantee a good algorithm performance is known to be a non-trivial task. See Stützle et al. (2012) for a comprehensive review of how this problem has been studied in the ACO literature.

As we have seen in Sect. 4, water drops' velocity can be seen as an indicator of the quality of the partial solution constructed so far, that is, faster water drops have added components with lower soil. However, computing the desirability of a solution component in terms of the velocity (quality of a partial solution) and of the heuristic undesirability, as is defined for $\Delta \mathrm{soil}_j^k$, is very similar to the abandoned idea of *ant quantity* (see AS local update procedure in Table 1).

As explained in Sect. 3, each water drop k memorizes the amount of soil collected from the solution components added to the solution that it is constructing in a variable called $\mathrm{collected\_soil}^k$. The new value of $\mathrm{collected\_soil}^k$ is computed by adding the value of $\Delta \mathrm{soil}_j^k$ to its current value (which contains the amount of soil collected from previous solution components), as it is shown in Eq. 8:

$$\mathrm{collected\_soil}^k = \mathrm{collected\_soil}^k + \Delta \mathrm{soil}_j^k, \tag{8}$$

Last, one might also ask whether the inspiring metaphor is a realistic model of the process of erosion in rivers. For example, if soil is removed, it is unclear why then the new amount of soil is computed by an equation such as Eq. 6 that uses a decay factor φ, and not simply by subtracting $\Delta \mathrm{soil}_j^k$. Additionally, the metaphor of water drops acting as individual particles removing the soil in the riverbeds is unconvincing, as water in a river should rather be seen as a moving fluid.

Therefore, if the goal of the author was to test the optimization capabilities of natural rivers (as it is mentioned repeatedly in (Shah-Hosseini 2007, 2008, 2009)), it would have been a better approach to start with some of the models available in the scientific literature describing this process (e.g., Merritt et al. 2003).

4.3 Global update

The global pheromone update in ACO is performed at the end of an iteration once all solutions have been completed. The main goal of this algorithmic component is to give a *positive feedback* by increasing the amount of pheromone associated with solution components that belong to good solutions; common choices in ACO algorithms are to update pheromones that belong to the components of the best solution s^{best} found in the current iteration (*iteration-best* update) or since the first iteration of the algorithm (*global-best* update), 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.

This algorithmic component was defined as follows in ACS[7] and IWD, [8] respectively:

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

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

[7] ACS is one of the oldest and best performing ACO algorithms (Dorigo and Gambardella 1997b); its global update rule is called *global pheromone trail updating rule*.

[8] There are two versions of this component in IWD. In the first one (Shah-Hosseini 2007), the ρ parameter was defined in [0, 1], making Eqs. 10 and 9 identical. However, for unknown reasons, in a later publication (Shah-Hosseini 2009) the interval of variability of parameter ρ was changed to $[-1, 0]$, leading to a somewhat different behavior of the global update procedure, as explained here.

where the parameter $\Delta\tau_j^{\text{best}}$ is commonly defined as the inverse of the total cost of the solution s^{best}, while $\Delta\text{soil}_j^{\text{best}}$ is proportional to the soil collected by the best water drop divided by the number of solution components:

$$\Delta\text{soil}_j^{\text{best}} = \text{collected\_soil}^{\text{best}}/N^{\text{best}} - 1 \tag{11}$$

The similarity between the two equations is clear. Equation 9 easily converts into Eq. 10 by multiplying ρ by -1. However, a more formal way to see this is via a redefinition of the interval over which the parameter ρ can vary in Eq. 9. That is, if we change this interval from its typical value of [0, 1] to [−1, 0], we also convert Eq. 9 into Eq. 10. Because of this, the *global soil update* component is a special case of the *global pheromone trail updating rule* proposed in Ant Colony System by Dorigo and Gambardella (1996).

The global soil update, as defined in Shah-Hosseini (2009), has different outcomes depending on the value of ρ and soil_j in the solution component. For simplicity, let us first consider the second summand in the first case of Eq. 10, that is, $-\rho \cdot \Delta\text{soil}_j^{\text{best}}$. Because $\Delta\text{soil}_j^{\text{best}}$ is defined as always positive (see Eq. 7) and as we have it multiplied by $-\rho$, the result of this second summand will always be negative and contribute with a *positive feedback* to the solution component, that is, a decrease in the value of *soil*.

On the other hand, the type of feedback given by the first summand in the first case of Eq. 10, $(1 + \rho) \cdot \text{soil}_j$, is more difficult to understand. It is easy to see that if $\text{soil}_j < 0$ the product $(1 + \rho) \cdot \text{soil}_j$ will be negative, and therefore, this summand contributes with a *positive feedback* to the solution component, which is the desired behavior (i.e., removing soil increases the probability that future water drops will select the component). However, if $\text{soil}_j > 0$, the resulting value of this summand will be positive, and therefore, it contributes with a *negative feedback* to the solution component and the function of the update in this case is just the opposite of what it should be.

5 Modifications of IWD

Very often, after a new algorithm is proposed, different modifications are proposed to enhance its performance and/or to overcome its drawbacks. In this section, we review the literature on IWD with a particular focus on the improvements that have been proposed since its initial publication in 2007 and we show that all these improvements were already present in the previous ACO algorithms. To select the relevant literature, we searched for the string "intelligent water drop" in the title or in the abstract of the articles indexed in Scopus (`www.scopus.com`) and Google Scholar (`http://scholar.google.com`). From this set of articles, we selected all those published in journals and those published in conferences that included at least one variant of the original IWD algorithmic components. The final set consisted of seven articles which are presented and discussed as follows:

1. Duan et al. (2008, 2009) were the first to propose a variant of IWD, where problem-specific information is added to IWD's random selection rule. This is a relatively minor modification that, in the case of ACO, was already present in its very first formulation (Dorigo et al. 1991b). Later, Booyavi et al. (2014) and Teymourian et al. (2016b) have proposed the use of a parameter λ to weight the importance of soil with respect to heuristic information. Also this weighting mechanism is a part of most ACO algorithms and in particular was already present in the very first ones (Dorigo and Stützle 2004).

2. Niu et al. (2012) proposed five modifications to enhance the original IWD; they are: *random soil and velocity initialization*, *conditional probability computation*, *bounded local soil update*, *elite global soil update*, and *combined local search*.

   - *Random soil and velocity initialization* aims to improve the diversity of the initial solutions of the algorithm and, according to the authors, helps avoiding premature convergence. Unfortunately, the authors did not test this hypothesis, for example, by comparing the random initialization with the scheme originally proposed for IWD.
   - *Conditional probability computation* consists of two changes in the stochastic solution construction procedure: (i) to include the heuristic information in the random selection rule along with a parameter to weight its relative importance; and (ii) to select the lower-cost component (i.e., greedy selection) with probability φ_0 and to use the random selection rule with the modification described in (i) otherwise. These two modifications were already proposed in the context of ACO. The use of a parameter to weight the relative importance of the heuristic information, modification (i), is part of the *random proportional rule* of Ant System (see AS transition rule in Table 1), the first ACO algorithm ever published (Dorigo et al. 1991b; Dorigo 1992; Dorigo et al. 1996); while modification (ii) was used in the *pseudo-random proportional rule* of Ant Colony System (see ACS transition rule in Table 1) and was first introduced in (Gambardella and Dorigo 1996).
   - *Bounded local soil update* uses two values, Δ_{max} and Δ_{min}, to set, respectively, the maximum and minimum changes in the amount of *soil* for a given solution component. This very same idea, in the form of upper and lower bounds to the value of pheromones, was introduced in the ACO variant called $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System, first proposed in a Technical Report in 1996 (Stützle and Hoos 1996), and later published in (Stützle and Hoos 2000).
   - *Elite global soil update* uses more than one water drop to update the soil in the *global soil update*. The idea of using more than one solution to update the pheromone trails was also explored in the context of ACO. This was done with Elitist Ant System, first proposed in Dorigo's PhD thesis (Dorigo 1992) and then published in (Dorigo et al. 1996).
   - *Combined local search* adds a local search phase to the IWD algorithm. As said when describing the ACO metaheuristic in Sect. 2, *daemon actions* often consist of a local search that improves the solutions constructed by the ants. The first publications to introduce local search in ACO algorithms are (Dorigo and Gambardella 1997b; Stützle and Hoos 1997); these were followed by many other (e.g., Maniezzo and Colorni 1999; Gambardella et al. 1999) and nowadays, the usage of a local search routine is pretty standard in the best performing ACO algorithms (Dorigo and Stützle 2004).

3. Msallam and Hamdan (2011) propose to reinitialize the soil and the velocities of all water drops after reaching a certain number of iterations without improving the global-best solution. The very same reinitialization scheme proposed in (Msallam and Hamdan 2011) was proposed for $\mathcal{MM}$AS (Stützle and Hoos 1997, 2000) and has been widely used in the ACO literature (Dorigo and Stützle 2004).
4. Alijla et al. (2014) have proposed to replace the original (i.e., Eq. 3) random selection rule of IWD with two ranking selection methods originally proposed for GAs[9], one

[9] The GA's community realized that proportionate selection is not best suited for optimization because it assumes only positive fitness values and cannot differentiate between small fitness differences. The linear

linear and one exponential. Both selection methods rank in descending order the feasible solution components according to their value of soil. In linear selection, the probability of selecting a solution component is computed using a linear function where a user-selected parameter, SP, controls the steepness of the gradient. In exponential selection, the feasible components are weighted exponentially according to their ranks. These two ranking selection methods aim to overcome three shortcomings of IWD's random selection rule, that is its inability (i) to accommodate negative soil values, (ii) to differentiate between solution components with small soil difference, and (iii) to control local stagnation. Unlike IWD, ACO algorithms do not present these shortcomings because pheromones have always positive values, which avoids shortcoming (i), and there are mechanisms to bound pheromones lower and upper limit, avoiding the shortcomings (ii) and (iii). Although the authors argue that these problems are caused by IWD's transition rule that implements fitness proportional selection, shortcomings (ii) and (iii) are rather the result of having negative soil values and of the way in which the global soil update is defined. That is, IWD has mechanisms to manage solution components with negative and positive values in the transition rule (see Eqs. 3 and 4); however, this is not the case for the global soil update (see Eq. 10) (see the discussion on the global soil update in Sect. 4.3).

In this review, we have identified the algorithmic ideas proposed to improve IWD since its initial publication in 2007. A few articles seek to overcome drawbacks and limitation of the algorithm (Msallam and Hamdan 2011; Alijla et al. 2014), while others propose modifications to enhance its exploration–exploitation capabilities (Duan et al. 2008, 2009; Niu et al. 2012; Booyavi et al. 2014; Teymourian et al. 2016a).

Although the research on IWD is not particularly rich, we found that everything proposed to modify this algorithm consists of ideas that were first proposed in the context of ACO. In fact, most of these modifications can be matched directly to well-known ACO algorithms that have been in the literature for many years, even before the first IWD algorithm was proposed.

6 Metaphors in stochastic optimization

Taking inspiration from the observation of natural systems has played an important role in the history of the field of metaheuristics. Such inspirations have provided new ideas that have resulted in a number of highly effective optimization frameworks that have changed and extended the optimization toolset (Corne et al. 1999; Gendreau and Potvin 2010; Kacprzyk and Pedrycz 2015), showing that ideas coming from other fields of knowledge could be useful and effective also in optimization. Notable examples are evolutionary computation (Holland 1975; Goldberg 1989), simulated annealing (Kirkpatrick et al. 1983), particle swarm optimization (Eberhart and Kennedy 1995), ant colony optimization (Dorigo 1992; Dorigo et al. 1996; Dorigo and Di Caro 1999), and so on.

In all these successful examples, there was always a direct match between the inspiring source (i.e., the natural metaphor) and the mathematical models and formal terminology used in the corresponding optimization algorithms. This has made the understanding of these algorithms particularly easy and, together with the fact that they provided high-quality results, favored their application to a vast set of different optimization problems. As a matter of fact, the key to the success of metaphor-based algorithms was effective ideas clearly translated into algorithm terms and insightful research supporting them. Unfortunately, in recent years many

(Baker 1987) and nonlinear (Goldberg and Deb 1991; Michalewicz 1992) ranking selection methods were introduced to alleviate these problems.

researchers started to propose optimization algorithms inspired by eye-catching metaphors which often bring very little, if any, to the understanding and performance of the proposed algorithms.

In fact, tens, if not hundreds, of optimization algorithms have been published, often in obscure journals and conferences, using all different types of metaphors (see Campelo 2017, for a long, although non-exhaustive list of metaphor-inspired optimization algorithms) that most of the time do not bring anything to the understanding of the optimization process. Unfortunately, it has become very frequent to come across articles proposing these types of metaphor-inspired algorithms, where it is very hard to understand the nature of the metaphor and the reason why the metaphor was being used to develop the algorithm.

Although one could get the impression that the usage of metaphors is the cause of this problem, we believe this not to be the case. On the contrary, when properly used, metaphors are a great tool to present, describe and understand new ideas, and their usage is very well established in science. In fact, when things are very tricky to explain in formal terms, the use of metaphors and analogies can be a powerful tool.[10] On the contrary, what is questionable is the way in which a large part of the research community has been using metaphors to introduce new algorithms.

We believe that the introduction of new metaphors should be limited to the cases in which they are indeed useful to express a new concept. This means that (i) the inspiring metaphor should bring some new concepts that are related to the optimization process proposed and that (ii) using the metaphor terminology should make it easier to convey these new concepts. Obviously, (iii) it should also be the case that the proposed algorithmic ideas are indeed new and not just a reformulation, using new terminology, of ideas already published in the literature. It is not acceptable to use a metaphor to promote an algorithm as interesting or novel just because the metaphor is interesting or novel for the author. In the context of metaheuristics, metaphors must have a clear and sound scientific motivation and conditions (i), (ii) and (iii) should always be verified.

In our future research, we intend to examine a number of the so-called *novel* metaphor-based algorithms to understand whether they meet the above-mentioned requirements and therefore deserve to be considered novel. In this article, we have contributed to this research program by providing an analysis of the IWD algorithm and by showing that this algorithm, once the proposed terminology based on water drops is removed, is just a particular instance of ACO. However, at the moment, it is very hard to say anything conclusive over the vast majority of recently proposed metaphor-based algorithms—the number of studies like the one presented in this article is still very low (Weyland 2010; Melvin et al. 2012; Sörensen 2015; Piotrowski et al. 2014).

7 Conclusions

In this article, we have analyzed the IWD algorithm and compared it with the well-known ACO metaheuristic. We found that the random selection rule of IWD is a simplified version of the random proportional rule of Ant System, the very first ACO algorithm. In particular, the transition rule of IWD simplifies to the one of ACO by setting the parameters $\alpha = -1$ and $\beta = 0$, which means that while in ACO high pheromone values indicate good-quality

[10] For example, consider the metaphoric explanation of gravity by Einstein in terms of space–time grid warping caused by an object's mass, or selfish genes in genetics to introduce the idea of organism differentiation within same species, or the terminology adopted in graph theory that includes terms such as tree, root, leaf, forest.

components, in IWD good-quality components have a low soil value and that in IWD the heuristic information is not used. Additionally, local soil update is a special case of the local reinforcement that was proposed in the Ant-Q algorithm. In fact, the only, small, difference between IWD and this earlier ACO algorithm is a parameter Δsoil_j computed using the water drop's velocity and the solution components' soil_j and heuristic undesirability. Unfortunately, the rationale behind the definition of Δsoil_j and the general idea over which the local soil update is based are based on unrealistic assumptions of how river systems erosion works. The last algorithmic component, global soil update, is a special case of the global pheromone trail updating rule proposed in Ant Colony System, in which the typical parameter interval of ρ in ACO is redefined to the interval $[-1, 0]$.

From our review of the literature of IWD, we conclude that research on IWD is just a repetition of research ideas that had already been explored in the context of ACO.

In fact, all the modifications reviewed in this article have a direct correspondence with a specific modification proposed before for an ACO algorithm.

Therefore, IWD and its variants are not novel and should rather be considered as particular instantiations of the ACO metaheuristic.

We discussed the use of metaphors in the field of metaheuristics, whose role in the development of a number of highly effective algorithms frameworks has been undeniable. However, many important problems have arisen due to the overuse of metaphors that do not bring anything to the understanding of new optimization algorithms. We believe that more analyses have to be done to algorithms published as *novel* metaphor-based approaches in the metaheuristics literature, so that we can clarify whether they are indeed novel or not and broaden our understanding of their contributions. However, it is clear that adopting the approach presented in this paper to analyze all the metaphor-inspired metaheuristic algorithms that have not been rigorously analyzed yet, which are in the order of hundreds (Campelo 2017), would require an unreasonable amount of work and time. Therefore, our future research will include searching for more efficient ways to address this issue.

Finally, if we reconsider the three criteria proposed in Sect. 6 of this article, namely the fact (i) that the inspiring metaphor should bring some new concepts that are related to the optimization process proposed, (ii) that using the metaphor terminology should make it easier to convey these new concepts, and (iii) that the proposed algorithmic ideas are indeed new and not just a reformulation, using new terminology, of ideas already published in the literature, we can conclude that the IWD algorithm fails on all criteria and is therefore unnecessary and misleading.

Acknowledgements Marco Dorigo and Thomas Stützle acknowledge support from the Belgian FRS-FNRS, of which they are Research Directors.

References

Alaya, I., Solnon, C., & Ghédira, K. (2007). Ant colony optimization for multi-objective optimization problems. In *Proceedings of the 19th IEEE international conference on tools with artificial intelligence* (ICTAI 2007) (Vol. 1, pp. 450–457) IEEE Computer Society Press, Los Alamitos, CA.

Alijla, B. O., Wong, L. P., Lim, C. P., Khader, A. T., & Al-Betar, M. A. (2014). A modified intelligent water drops algorithm and its application to optimization problems. *Expert Systems with Applications*, *41*(15), 6555–6569.

Askarzadeh, A. (2014). Bird mating optimizer: An optimization algorithm inspired by bird mating strategies. *Communications in Nonlinear Science and Numerical Simulation*, *19*(4), 1213–1228.

Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In *Proceedings of the 2007 congress on evolutionary computation (CEC 2007)* (pp. 3226–3231).

Baker, J.E. (1987). Reducing bias and inefficiency in the selection algorithm. In: Goldberg DE, Lingle R (Eds.) *Proceedings of the second international conference on genetic algorithms* (ICGA'87), Lawrence Erlbaum Associates (Vol. 206, pp. 14–21).

Birattari, M., Balaprakash, P., & Dorigo, M. (2006). The ACO/F-RACE algorithm for combinatorial optimization under uncertainty. In K. F. Doerner, M. Gendreau, P. Greistorfer, W. J. Gutjahr, R. F. Hartl, & M. Reimann (Eds.), *Metaheuristics: Progress in complex systems optimization, operations research/computer science interfaces series* (Vol. 39, pp. 189–203). New York, NY: Springer.

Birbil, Ş. I., & Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, *25*(3), 263–282.

Blum, C. (2005). Beam-ACO–hybridizing ant colony optimization with beam search: An application to open shop scheduling. *Computers & Operations Research*, *32*(6), 1565–1591.

Blum, C., & Dorigo, M. (2004). The hyper-cube framework for ant colony optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, *34*(2), 1161–1172.

Booyavi, Z., Teymourian, E., Komaki, G., & Sheikh, S. (2014). An improved optimization method based on the intelligent water drops algorithm for the vehicle routing problem. In *Proceedings of 2014 IEEE symposium on computational intelligence in production and logistics systems* (CIPLS) (pp. 59–66), IEEE Press

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, *89*, 319–328.

Campelo, F. (2017). Evolutionary computation bestiary. https://github.com/fcampelo/EC-Bestiary, version visited last on 22 January 2018

Colorni, A., Dorigo, M., & Maniezzo, V. (1992). Distributed optimization by ant colonies. In: F. J. Varela & P. Bourgine (Eds.) *Proceedings of the first european conference on artificial life* (pp. 134–142), MIT Press, Cambridge, MA

Cordón, O., de Viana, IF., Herrera, F., & Moreno, L. (2000). A new ACO model integrating evolutionary computation concepts: The best-worst ant system. In: M. Dorigo, et al. (Eds.) *Abstract proceedings of ANTS 2000: From ant colonies to artificial ants: Second international workshop on ant algorithms*, IRIDIA, Université Libre de Bruxelles, Belgium (pp. 22–29)

Corne, D., Dorigo, M., Glover, F., Dasgupta, D., Moscato, P., Poli, R., et al. (1999). *New ideas in optimization*. New York: McGraw Hill.

Cuevas, E., Miguel, C., Zaldívar, D., & Pérez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications*, *40*(16), 6374–6384.

Deneubourg, J. L., Aron, S., Goss, S., & Pasteels, J. M. (1990). The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, *3*(2), 159–168. https://doi.org/10.1007/BF01417909.

Dorigo, M. (1992). Optimization, learning and natural algorithms. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy (**in Italian**)

Dorigo, M., & Di Caro, G. A. (1999). The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New ideas in optimization* (pp. 11–32). London, UK: McGraw Hill.

Dorigo, M., & Gambardella, L. M. (1996). Ant Colony System. Tech. Rep. IRIDIA/96-05, IRIDIA, Université Libre de Bruxelles, Belgium

Dorigo, M., & Gambardella, L. M. (1997a). Ant colonies for the traveling salesman problem. *BioSystems*, *43*(2), 73–81. https://doi.org/10.1016/S0303-2647(97)01708-5.

Dorigo, M., & Gambardella, L. M. (1997b). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, *1*(1), 53–66.

Dorigo, M., Maniezzo, V., & Colorni, A. (1991a). The ant system: An autocatalytic optimizing process. Tech. Rep. 91-016 Revised, Dipartimento di Elettronica, Politecnico di Milano, Italy

Dorigo, M., Maniezzo, V., & Colorni, A. (1991b). Positive feedback as a search strategy. Tech. Rep. 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, *26*(1), 29–41.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Cambridge, MA: MIT Press.

Duan, H., Liu, S., & Wu, J. (2008). Air robot path planning based on intelligent water drops optimization. In *Proceedings of the international joint conference on neural networks* (IJCNN 2008 - Hong Kong) (pp. 1397–1401) IEEE.

Duan, H., Liu, S., & Wu, J. (2009). Novel intelligent water drops optimization approach to single UCAV smooth trajectory planning. *Aerospace Science and Technology*, *13*(8), 442–449.

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science* (pp. 39–43)

Erol, O. K., & Eksin, I. (2006). A new optimization method: Big bang-big crunch. *Advances in Engineering Software*, *37*(2), 106–111.

Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm-a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, *110*, 151–166.

Gambardella, L. M., & Dorigo, M. (1995). Ant-Q: A reinforcement learning approach to the traveling salesman problem. In: A. Prieditis & S. Russell (Eds.) *Proceedings of the twelfth international conference on machine learning (ML-95)* (pp. 252–260). Morgan Kaufmann Publishers, Palo Alto, CA.

Gambardella, L. M., & Dorigo, M. (1996). Solving symmetric and asymmetric TSPs by ant colonies. In T. Bäck, Fukuda & Z. Michalewicz (Eds.) *Proceedings of the 1996 IEEE international conference on evolutionary computation* (ICEC'96) (pp. 622–627). IEEE Press, Piscataway, NJ.

Gambardella, L. M., Taillard, É. D., & Dorigo, M. (1999). Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, *50*(2), 167–176.

Gandomi, A. H. (2014). Interior search algorithm (isa): A novel approach for global optimization. *ISA Transactions*, *53*(4), 1168–1183.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Journal of Simulation*, *76*(2), 60–68.

Gendreau, M., & Potvin, J. Y. (Eds.). (2010). *Handbook of Metaheuristics, International Series in Operations Research & Management Science* (2nd ed., Vol. 146). New York, NY: Springer.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Boston, MA: Addison-Wesley.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins (Ed.), *Foundations of genetic algorithms* (Vol. 1, pp. 69–93). Elsevier.

Guntsch, M., & Middendorf, M. (2002). A population based approach for ACO. In S. Cagnoni et al. (Eds.) *Applications of evolutionary computing, proceedings of evoworkshops 2002*, Springer, Heidelberg, Germany, Lecture Notes in Computer Science, vol 2279, pp 71–80

Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information Sciences*, *222*, 175–184.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Joslin, D. E., & Clements, D. P. (1999). Squeaky wheel optimization. *Journal of Artificial Intelligence Research*, *10*, 353–373.

Kacprzyk, J., & Pedrycz, W. (Eds.). (2015). *Springer handbook of computational intelligence*. Heidelberg: Springer.

Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: Charged system search. *Acta Mechanica*, *213*(3–4), 267–289.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.

Maniezzo, V. (1999). Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. *INFORMS Journal on Computing*, *11*(4), 358–369.

Maniezzo, V., & Carbonaro, A. (2000). An ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems*, *16*(8), 927–935.

Maniezzo, V., & Colorni, A. (1999). The Ant System applied to the quadratic assignment problem. *IEEE Transactions on Data and Knowledge Engineering*, *11*(5), 769–778.

Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological informatics*, *1*(4), 355–366.

Melvin, G., Dodd, T. J., & Groß, R. (2012). Why 'GSA: A gravitational search algorithm' is not genuinely based on the law of gravity. *Natural Computing*, *11*(4), 719–720.

Merritt, W. S., Letcher, R. A., & Jakeman, A. J. (2003). A review of erosion and sediment transport models. *Environmental Modelling & Software*, *18*(8–9), 761–799.

Michalewicz, Z. (1992). *Genetic algorithms + data structures = evolution program*. Berlin: Springer.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.

Msallam, M. M., & Hamdan, M. (2011). Improved intelligent water drops algorithm using adaptive schema. *International Journal of Bio-Inspired Computation*, *3*(2), 103–111.

Niu, S., Ong, S. K., & Nee, A. Y. (2012). An improved intelligent water drops algorithm for achieving optimal job-shop scheduling solutions. *International Journal of Production Research*, *50*(15), 4192–4205.

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, *22*(3), 52–67.

Piotrowski, A. P., Napiorkowski, J. J., & Rowinski, P. M. (2014). How novel is the "novel" black hole optimization approach? *Information Sciences*, *267*, 191–200.

Rao, R. V., Savsani, V. J., & Vakharia, D. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, *43*(3), 303–315.

Ray, T., & Liew, K. M. (2003). Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, *7*(4), 386–396.

Shah-Hosseini, H. (2007). Problem solving by intelligent water drops. In *Proceedings of the 2007 congress on evolutionary computation (CEC 2007)* (pp. 3226–3231). IEEE, IEEE Press, Piscataway, NJ

Shah-Hosseini, H. (2008). Intelligent water drops algorithm: A new optimization method for solving the multiple knapsack problem. *International Journal of Intelligent Computing and Cybernetics*, *1*(2), 193–212.

Shah-Hosseini, H. (2009). The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm. *International Journal of Bio-Inspired Computation*, *1*(1–2), 71–79.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713.

Simon, D., Rarick, R., Ergezer, M., & Du, D. (2011). Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. *Information Sciences*, *181*(7), 1224–1248.

Socha, K., & Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, *185*(3), 1155–1173. https://doi.org/10.1016/j.ejor.2006.06.046.

Sörensen, K. (2015). Metaheuristics–the metaphor exposed. *International Transactions in Operational Research*, *22*(1), 3–18. https://doi.org/10.1111/itor.12001.

Stützle, T., & Hoos, H. H. (1996). Improving the Ant System: A detailed report on the $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. Tech. Rep. AIDA–96–12, FG Intellektik, FB Informatik, TU Darmstadt, Germany

Stützle, T., & Hoos, H. H. (1997). The $\mathcal{MAX}$–$\mathcal{MIN}$ ant system and local search for the traveling salesman problem. In T. Bäck, Z. Michalewicz & X. Yao, (Eds.) Proceedings of the 1997 IEEE international conference on evolutionary computation (ICEC'97) (pp. 309–314). IEEE Press, Piscataway, NJ.

Stützle, T., & Hoos, H. H. (2000). $\mathcal{MAX}$–$\mathcal{MIN}$ ant system. *Future Generation Computer Systems*, *16*(8), 889–914.

Stützle, T., López-Ibáñez, M., Pellegrini, P., Maur, M., Montes de Oca, M. A., Birattari, M., et al. (2012). Parameter adaptation in ant colony optimization. In Y. Hamadi, E. Monfroy, & F. Saubion (Eds.), *Autonomous Search* (pp. 191–215). Berlin: Springer. https://doi.org/10.1007/978-3-642-21434-9_8.

Teymourian, E., Kayvanfar, V., Komaki, G. M., & Khodarahmi, M. (2016a). An enhanced intelligent water drops algorithm for scheduling of an agile manufacturing system. *International Journal of Information Technology & Decision Making*, *15*(02), 239–266.

Teymourian, E., Kayvanfar, V., Komaki, G. M., & Zandieh, M. (2016b). Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem. *Information Sciences*, *334*, 354–378.

Weyland, D. (2010). A rigorous analysis of the harmony search algorithm: How the research community can be misled by a "novel" methodology. *International Journal of Applied Metaheuristic Computing*, *12*(2), 50–60.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169–178). Springer

Yang, X. S. (2012). Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation* (pp. 240–249). Springer

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.