This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation

An Adaptive Convergence-Trajectory Controlled Ant Colony Optimization Algorithm with Application to Water Distribution System Design Problems

Feifei Zheng, Aaron C. Zecchin, Jeffery P. Newman, Holger R. Maier, and Graeme C. Dandy

***Abstract*— Evolutionary algorithms and other metaheuristics have been employed widely to solve optimization problems in many different fields over the past few decades. Their performance in finding optimal solutions often depends heavily on the parameterization of the algorithm's search operators, which affect an algorithm's balance between search diversification and intensification. While many parameter-adaptive algorithms have been developed to improve the searching ability of meta-heuristics, their performance is often unsatisfactory when applied to real-world problems. This is, at least in part, because available computational budgets are often constrained in such settings due to the long simulation times associated with objective function and/or constraint evaluation, thereby preventing convergence of existing parameter-adaptive algorithms. To this end, this study proposes an innovative parameter-adaptive strategy for ant colony optimization (ACO) algorithms based on controlling the convergence trajectory in decision space to follow any pre-specified path, aimed at finding the best possible solution within a given, and limited, computational budget. The utility of the proposed convergence- trajectory controlled ACO (ACO_{CTC}) algorithm is demonstrated using six water distribution system design problems (WDSDPs, a difficult type of combinatorial problem in water resources) with varying complexity. The results show that the proposed ACO_{CTC} successfully enables the specified convergence trajectories to be followed by automatically adjusting the algorithm's parameter values. Different convergence trajectories significantly affect the algorithm's final performance (solution quality). The trajectory with a slight bias towards diversification in the first half and more emphasis on intensification during the second half of the search exhibits substantially improved performance compared to the best available ACO variant with the best parameterization (no convergence control) for all WDSDPs and computational scenarios considered. For the two large-scale WDSDPs, new best-known solutions are found by the proposed ACO_{CTC}.**

***Index Terms*— Convergence trajectory, Parameter adaptive, Ant Colony Optimization (ACO), Water distribution system design problems (WDSDPs)**

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456".

F. Zheng is with College of Civil Engineering and Architecture, Zhejiang University, Anzhong Building, Zijingang Campus, 866 Yuhangtang Rd, Hangzhou 310058, China (e-mail: feifeizheng@zju.edu.cn).

H. R. Maier is with School of Civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, SA, Australia, and College of Civil Engineering and Architecture, Zhejiang University (holger.maier@adelaide.edu.au). A. C. Zecchin, J. P. Newman, and G. C. Dandy are with The University of Adelaide, Adelaide, SA, Australia (e-mail: aaron.zecchin@adelaide.edu.au, jeffrey.newman@adelaide.edu.au, graeme.dandy@adelaide.edu.au).

I. Introduction

EVOLUTIONARY algorithms and other meta-heuristics have often been used to solve complex optimization problems in many different application areas over the past few decades, including engineering, science and business [1-3]. Their wide up-take in practical applications is mainly due to the fact that they provide a global search strategy that does not need gradient or hessian information. This makes them ideally suited for application to real-world problems, which typically have high complexity, non-linear objectives and/or constraints, discontinuities, interdependencies among variables and large solution spaces [1], [4], [5].

The performance of meta-heuristics is often measured in terms of the quality of the solutions identified and the efficiency with which these solutions are reached. Typically, an algorithm's searching performance is a function of problem characteristics (fitness function), as well as algorithm searching behavior [5]. The latter is often characterized by diversification and intensification, where diversification (exploration) provides an indication of how widely the decision-space is being searched, while intensification (exploitation) represents the exploitative strength in local regions in the decision space [6]. Consequently, for a given problem, the performance of meta-heuristics, in terms of solution quality and computational efficiency, is controlled by the trade-off between search diversification and intensification. Such trade-offs are typically governed by the algorithm operator parameterization [7] (i.e., operator parameter settings govern the balance of the search behavior).

Much research has been carried out to determine the parameters for meta-heuristics that can achieve an appropriate balance between diversification and intensification (i.e. optimal trade-offs). Traditionally, parameter tuning through trial-and-error has often been used [8]. However, despite its simplicity, parameter tuning typically requires significant computational overheads to conduct calibration studies, hindering the algorithm's broad application in practice. Subsequently, a number of parameter-adaption strategies have been developed in order to enable the most appropriate parameter values to be identified in a more computationally efficient manner [7].

Many previous adaption strategies are based on recognising well performing parameter values [9-11] or operators [12], thereby assigning pre-defined parameter trajectories to improve the algorithm's performance [13-15]. However, these adaption strategies are often challenged when dealing with real-world problems, due to the fact that, in most instances, the available computational budgets for such problems are insufficient to ensure algorithm convergence, which accordingly results in unsatisfactory algorithm performance in terms of being able to identify optimal solutions [16-19]. This is because for such real-world problems, the evaluation of solution quality can have significant computational overheads, as this requires the running of computationally expensive simulation models of the system to be optimized [5]. In such situations, the aim of the optimization is not necessarily the identification of the globally-optimal solution, but to identify the best possible solutions within the available computational budget [20]. Consequently, there is a need to develop a parameter adaptation strategy that not only enables pre-specified parameter trajectories to be followed, but that also ensures that convergence to increasingly higher fitness sub-regions in decision space occurs for a given computational budget.

In order to achieve this, a fundamentally different approach to parameter adaption is introduced in this paper. The basis of the proposed strategy is to directly control the algorithm's *convergence trajectories in decision space*, thereby ensuring the algorithm's convergence in decision space at the completion of the given computational budgets. The proposed strategy is able to automatically adjust algorithm parameters to follow a pre-specified convergence trajectory, thereby appropriately balancing the trade-off between search diversification and intensification within the given computational constraints. That is, in the early stages of the search, the algorithm parameters are controlled to allow the generation of a broad distribution of solutions throughout the solution space. As the search continues, and the algorithm learns more about the objective space, the parameters are adaptively changed to ensure controlled convergence of the search to the promising regions of the of the search space. This means that best use is made of available computational budgets, making the approach particularly attractive for application to real-world problems where solution evaluation is computationally expensive, and the focus is on identifying the best possible solutions within acceptable computational budgets, rather than necessarily identifying the globally optimal solution.

In order to enable the desired adaptive convergence control strategy to be implemented, ant colony optimization (ACO) is used as the metaheuristic. A detailed reason for this, including a theoretical proof, is presented in Section III. To demonstrate how the proposed parametric convergence-trajectory-controlled ACO (ACO_{CTC}) can be used in a real-world setting, it is applied to the water distribution system design problem (WDSDP), which is a benchmark problem in civil engineering

The remainder of the paper is organized as follows. Section II outlines the background and motivation for the convergence based adaption strategy. Section III outlines the proposed ACO_{CTC}, firstly with a conceptual overview followed by a detailed formulation of the algorithm. Section IV outlines the experimental methodology that is used to empirically demonstrate the effectiveness of the proposed ACO_{CTC}, as well as the description of the WDSDP, and Section V presents the results and discussion. Finally, the conclusions are given in Section VI.

II. Background and Motivation

Most commonly, the use of parameter-adaption schemes aims to identify globally optimal solutions with the least amount of computational effort, but without placing deliberate constraints on the available computational budget. Adaptive strategies with this focus generally adjust parameters based on feedback from the fitness function during the search. For instance, Sakanashi *et al.* [9] proposed a parameter-adaption approach that ensures that genetic algorithms explore the search space sufficiently based on the average and best performance at the current generation compared with those at a number of generations in the past. Subsequently, Qin *et al*. [10] and Zheng *et al.* [11] proposed self-adaptive differential evolution algorithms applied to global numeric optimization and water distribution network design problems, respectively. In their work, algorithm parameters were adjusted according to the solution quality of the offspring within the optimization. In a similar way, Hadka and Reed [12] developed the hybrid Borg algorithm, where the operators that produce better solutions than others are given a higher likelihood of producing offspring.

In the domain of ant colony optimization (ACO), various parameter-adaption methods have been developed to improve the algorithm's performance. For example, Li and Wu [21] proposed an objective-function-based heuristic assignment approach to automatically update the pheromone within the optimization process, and Chiang *et al.* [22] developed a self-adaptive ACO to guide the artificial ants to effectively construct feasible solutions for multi-modal resource-constrained project scheduling problems. Subsequently, Afshar [23] incorporated a Gaussian probability density function into the ACO to represent the pheromone concentration over the allowable range of each decision variable, rather than pre-setting the pheromone updating parameters, and Favuzza *et al.* [24] adapted the ACO's control parameters according to the number of the generations without improvement in solution quality. Similar to the majority of other parameter-adaption based evolutionary algorithms, these ACOs were designed to efficiently find globally optimal solutions without considering computational constraints.

In recent years, there has been an increased focus on the identification of parameters that enable the best possible solutions to be identified for restricted computational budgets [5]. This is motivated by the application of metaheuristics to large real-world problems, for which simulation models with reasonably long run-times (e.g. in the order of many seconds or even minutes) are used for evaluating objective function values and/or constraints [20], [25]. This requires a degree of control over the trade-offs between diversification and intensification within the available computational budgets that can generally not be achieved using the most commonly used adaptive methods outlined above [20].

In order to attain the desired level of control, parameter-adaptive schemes based on user-defined trajectories can be used. For example, Krink *et al*. [14] presented deterministic control mechanisms for the mutation and selection operators of meta-heuristics based on concepts of self-organized criticality, and Herrera and Lozano [15] used fuzzy logic controllers to adapt the choice of the crossover operator and the selective pressure of meta-heuristics. These parameter-adaptive strategies typically enable greater emphasis to be placed on diversification during the early part of the search and more emphasis to be placed on intensification during the latter parts of the search, thereby facilitating convergence in good regions of the solution space. However, a shortcoming of these approaches is that the selected adaptive trajectory in parameter space is not necessarily translated to the desired convergence trajectory in decision space, thereby not necessarily achieving the desired trade-offs between search diversification and intensification. This is because convergence behavior in the decision space has to be inferred by knowledge of the relative impact of different parameters on diversification and intensification.

As an alternative approach to address this issue, Gibbs *et al*. [16] identified the population size of genetic algorithms (GAs) required to ensure convergence in decision space for a given computational budget. However, the approach only considers convergence due to genetic drift, ignoring the effect of selection pressure, which is a function of other user-selected parameters (e.g. probability of cross-over and mutation). Consequently, there is no explicit control over the convergence trajectory in solution space, making it unlikely that computational resources are divided optimally between diversification and intensification.

In order to overcome this shortcoming of existing adaptive approaches, an ACO-based parameter-adaptation strategy is introduced in this paper that (i) enables target convergence trajectories to be pre-defined in decision space, rather than in the parameter space; and (ii) adapts its parameters during the search to ensure that this target convergence trajectory is followed (in contrast to existing methods that infer a trajectory in the decision space from a pre-specified trajectory in the parameter space). By adopting a parameter adaption strategy centered on controlling convergence, the balance between diversification and intensification is explicitly controlled, and a rational increasingly focused allocation of computational resources throughout the search can be ensured. While this might not necessarily enable globally optimal solutions to be identified, it ensures convergence after a given computational budget has been used up and enables users to control the relative allocation of available computational resources to diversification and intensification throughout the optimization run. This makes the approach especially suitable to application to real-world problems where computationally expensive simulation models are used for solution evaluation.

It should be noted that the use of ACO is central to the successful implementation of the proposed approach, as it enables the proposed adaptation strategy to be implemented. This is due to the fundamental nature in which optimization problems are structured and solution spaces are explored when ACO is used, as detailed in Section III.

III. The Convergence-trajectory Controlled adaptive ACO

The proposed convergence-trajectory-controlled adaptive ACO (ACO_{CTC}) is developed within this section. ACO is a population-based meta-heuristic analogically derived from the foraging behavior of a colony of ants [26], where a colony is able to determine the shortest path from its nest to a food source through indirect communication by the deposition of pheromone over time. In the following sections, a conceptual overview of the main components of the ACO_{CTC} is firstly presented in Section III.A, followed by a detailed development and formulation of the ACO_{CTC} algorithm in Section III.B.

A. Overview of the Proposed ACO_{CTC}

The fundamental concept behind the proposed algorithm is the development of a parameter adaption strategy based on controlling the algorithm's convergence behavior *in decision space*. This enables users to define a target convergence trajectory that provides an appropriate balance between diversification and intensification at various stages of the search, as well as ensuring algorithm convergence to improved, or higher fitness, sub-regions within the decision space at the conclusion of the optimization run. As algorithm parameters adapt automatically to ensure the pre-specified target convergence trajectory is being followed, there is no need for sensitivity analyses to determine appropriate values for the algorithm parameters or secondary parameters that control the way in which the primary parameters adapt throughout the search. Consequently, the proposed approach is well suited to identifying the best possible solutions for a given computational budget, as is often the case for real-world applications.

A schematic overview of the proposed framework is given in Fig. 1. The two primary *new concepts* within the framework are: (i) the selection of a *target convergence trajectory in decision space* as an input to the algorithm (top dashed box); and (ii) the *adaptive control* of the algorithm's parameters to ensure that the actual convergence behavior matches the target convergence trajectory (top box within iteration loop). These are discussed further in the following sections.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation

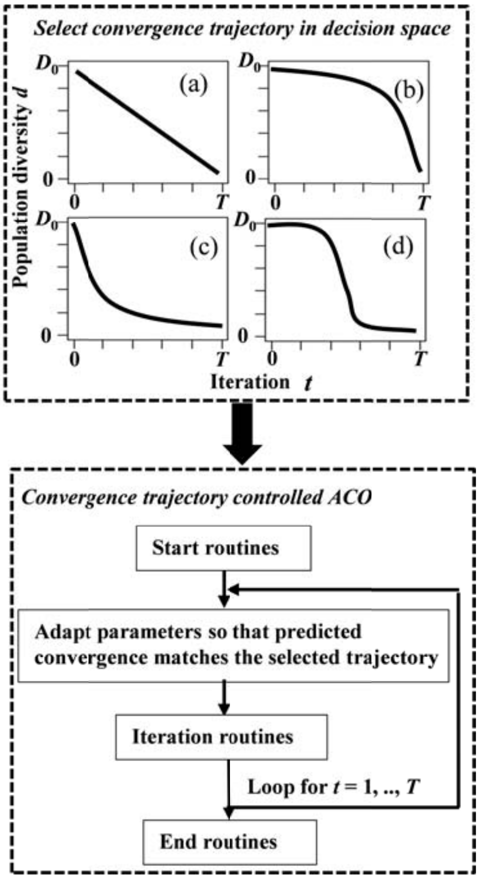

Fig. 1. Schematic outline of proposed ACO_{CTC}. The first stage (top box) involves the selection of a predefined convergence trajectory in decision space, where the vertical axis is population diversity measure d (D_0 is the initial population diversity), and the horizontal axis is the iteration number t (T is the maximum number of iterations allowed). The four depicted convergence trajectories represent: (a) a linear (balanced) convergence; (b) an exploration focused search; (c) an exploitation focused search; and (d) a logistic-type convergence. The second stage (bottom box) corresponds to the proposed optimization process, where the primary difference with standard ACO algorithms is the first box within the iteration loop, where the ACO parameters are adapted at the start of each iteration to ensure that the predicted diversity matches the predefined convergence trajectory.

1) Target convergence trajectories in decision space

Maintaining an appropriate convergence behavior in decision space over the searching time is important to ensure that algorithms perform well. In previous studies ([17-19]), the poor performance of meta-heuristics (including ACOs) has been observed to be typically associated with either (i) premature-convergent behavior, or (ii) non-convergent behavior. In the case of (i), algorithms do not use the available computational budget to effectively explore the search space, as convergence is reached before the computational budget has been exhausted – a result of too much emphasis on intensification and insufficient diversification within the searching process. In the case of (ii), algorithms have not converged by the time the available computational budget is exhausted, and the algorithm is still exploring the decision space widely at the end of the search – this case owing to too much focus on diversification and not enough intensification within the available constraints of the computational budget.

Properly allocating computational resources between diversification and intensification processes is a challenging problem for parameter adaptive strategies. The objective of the proposed methodology is to control the balance between diversification and intensification by directly controlling the algorithm's convergence behavior to follow a predefined convergence trajectory in decision space. These trajectories can be viewed as an iteration-wise allocation of computational search effort to increasingly smaller sub-regions of the search space.

A range of candidate target convergence trajectories are depicted in the top dashed box of Fig 1. The vertical axis indicates the diversity of solutions (d) throughout the decision space, and the horizontal axis is the iteration number (t). Candidate (a) represents a linear (balanced) convergence, candidate (b) shows a highly diversified trajectory during the early stages of the search, with rapid convergence towards the end of the search, and candidate (c) indicates a rapidly convergent and intensive search at the beginning. For candidate (d), the convergence follows a logistic-type pattern of an initial period of exploration followed by convergent and intensive search. Previous work on studying convergence behavior (e.g. [17-19]) has shown that trajectories (c) and (d) occur as natural convergence trajectories for ACO variants without parameter adaption. The algorithm presented in this paper has allowed for the exploration of a range of alternative convergence trajectories.

2) Parameter adaption

The central notion of the proposed parameter adaption strategy is the modification of the values of the parameters that control an optimization algorithm's relative degree of diversification and intensification in an iteration by iteration manner, such that the algorithm's predicted convergence behavior in decision space matches the target convergence trajectory. Within the proposed ACO_{CTC}, the parameter adaption manifests itself as the first operation within ACO's iteration loop (top box within iteration loop in Fig. 1), where the parameters are changed at the start of each iteration. The motivation for using ACO within the proposed strategy is that its problem representation allows for the development of an analytic predictor for convergence behavior, which is not possible with other types of algorithms.

ACO generates solutions based on a graph of selection probabilities. The learning process within ACO involves the manipulation of the selection probabilities to increase the likelihood of generating improved solutions from iteration to iteration. The manipulation of the selection probabilities is directly controlled by the ACO parameters - explorative settings lead to more even selection probabilities, and exploitative settings lead to greater differences in the probabilities.

Given knowledge of these selection probabilities at the start of an iteration, it is possible to analytically determine the predicted diversity amongst the population of solutions generated [see Proposition 1 in Section III.B(3)]. The relationship between the selection probabilities and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation

predicted population diversity, in combination with ACO's direct parametric control over the selection probabilities, has enabled the development of a systematic approach to adjusting parameters in order to achieve pre-defined target population diversity values within each iteration of the search.

B. Development and Formulation of ACO_{CTC}

As outlined in Fig. 1, the proposed convergence-trajectory-controlled adaptive ACO (ACO_{CTC}) works by, firstly, defining a convergence trajectory in decision space that provides the desired trade-off between diversification and intensification at different stages of the search, and secondly, by dynamically adjusting algorithm parameters throughout the search to enable the defined convergence trajectory to be followed. To attain this objective, we need to: (i) define a measure of convergence, representing a trade-off between diversification and intensification; and (ii) demonstrate that ACO convergence can be controlled at iteration level through run-time parameter adjustment, and hence enable the target trajectory to be followed

To enable a clear interpretation of ACO_{CTC} in this section, the following structure is used. Firstly, ACO is briefly introduced, followed by the definition of ACO convergence in decision space, for which a metric for algorithm convergence is proposed. Next, a derivation of the convergence predictor for ACO is provided, which is based on a proposition that analytically proves that ACO run-time convergence is a direct function of the decision space probability values for multi-graph problem structures (i.e. a graph problem type that contains multiple links between nodes, where each node is only connected to its immediately downstream or upstream nodes [28]). As a corollary, it is noted that the proposition demonstrates that ACO convergence can be controlled through dynamically adjusting parameters (it should be noted that this section does not aim to prove the utility of the proposed ACO_{CTC}, which is demonstrated empirically using case studies in Section V). Finally, the algorithm framework for ACO_{CTC} is outlined.

1) Overview of ACO

ACO deals with combinatorial problems represented as a directed graph $G(E,V)$, where each ant generates a solution by probabilistically constructing a permissible path through $G(E,V)$. At each decision point in an ant's path through $G(E,V)$, edges are probabilistically selected based on two factors [26]. The first factor is an edge's pheromone value τ, which is representative of the algorithm's learned information. At the completion of an iteration, good solution components are reinforced with pheromone, causing a relative increase in their τ values. The second factor is the visibility value η, representing an edge's localized influence on the objective function. Within iteration t, at decision point i, the j-th edge from this decision point, denoted as edge (i,j), is selected with probability

$$p_{ij}(t) = \frac{\tau_{ij}(t)^{\alpha}\eta_{ij}^{\beta}}{\sum_{k=1}^{N_i}\tau_{ik}(t)^{\alpha}\eta_{ik}^{\beta}} \tag{1}$$

where τ_{ij} is the pheromone value on edge (i,j), η_{ij} is the visibility value for edge (i,j), N_i is the number of decision options for decision point i, and α and β are the weighting exponents for the pheromone and visibility values. The α and β parameters control the algorithm's emphasis on pheromone and visibility in constructing solutions.

The updating of the pheromone values from iteration t to t+1 is the central component of ACO learning, as knowledge of good solution components (edges) is embodied by the relative magnitude of these values. The pheromone update rule contains two main processes, namely pheromone decay and pheromone reinforcement, and is given by

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{2}$$

where ρ is the pheromone persistence factor ($0<\rho<1$), and $\Delta\tau_{ij}(t)$ is the pheromone addition for edge (i,j). The incorporation of pheromone decay allows for a greater emphasis to be placed on more recent information, as edges that are not regularly updated will experience continual decay. Pheromone addition is used to proportionally reinforce edges that are found to be components of good solutions [26]. That is, for any given solutions $\boldsymbol{\theta}$ and $\boldsymbol{\theta}$' where $f(\boldsymbol{\theta}) < f(\boldsymbol{\theta}')$ (for a minimization problem), then the edges in $\boldsymbol{\theta}$ will receive a greater pheromone addition than the edges in the inferior solution $\boldsymbol{\theta}$'.

Pheromone addition typically occurs at the end of an iteration, and is a function of the entire set of the colony's solutions. Many different strategies exist for computing $\Delta\tau_{ij}(t)$, as this is the main aspect that defines different ACO variants. Common examples of ACO update strategies are reviewed in Zecchin *et al.* [17], including: ant system, where each ant adds pheromone to its own path, where the pheromone value is inversely proportional to the solution's objective function value; elitist ant system , which is similar to the ant system, but where additional pheromone is only added to the current global best solution; the elitist-rank ant system, where only the top ranking ants add pheromone to their paths, in addition to the elite global best solution; and the rank based ant system, where the top ranked ants add pheromone proportionally according to their ranks (i.e., better solutions contribute more pheromone). The form of the elitist-rank pheromone addition used within this work is defined in (16) and (17) in Section IV below.

As with most meta-heuristic applications to constrained problems, the application of ACO requires embedding the constraints into the objective f through the use of a penalty function [27]. That is, ACO solves the modified problem

$$\min_{\boldsymbol{\theta}} \ f(\boldsymbol{\theta}) + p(\boldsymbol{\theta}) \tag{3}$$

where the penalty function p is zero if all constraints are satisfied for solution $\boldsymbol{\theta}$, and nonzero if any constraints are violated.

2) *Definition of convergence*

Within this paper, the degree of convergence of a population of solutions is defined as the average pairwise distance within the decision space between all solutions, termed the *mean population distance*. Defining $\Theta = \{\boldsymbol{\theta}^{[1]},..., \boldsymbol{\theta}^{[m]}\}$ as the set of colony solutions, where $\boldsymbol{\theta}^{[k]}$ is the solution found by the *k*-th ant, the mean population distance is given as

$$\bar{d}(\Theta) = \frac{2}{m(m-1)} \sum_{k=1}^{m} \sum_{l=k+1}^{m} d\left(\boldsymbol{\theta}^{[k]}, \boldsymbol{\theta}^{[l]}\right) \tag{4}$$

where *m* is the number of ants in the colony, and $d\left(\boldsymbol{\theta}^{[k]}, \boldsymbol{\theta}^{[l]}\right)$ is the distance metric between solutions $\boldsymbol{\theta}^{[k]}$ and $\boldsymbol{\theta}^{[l]}$ in decision space. The adopted distance metric defines the topology of the search space, and its form typically depends on the characteristics of the problem being solved. The measure $\bar{d}(\Theta)$ can be considered as the averaged diameter of the decision sub-space hypersphere within which the algorithm is searching. For combinatorial problems with *N* decision variables, for which there is not necessarily a meaningful ordering of the decision options, an appropriate metric is the Hamming distance [17], which is defined as

$$d(\boldsymbol{\theta}, \boldsymbol{\theta}') = N - \sum_{i=1}^{N} I(\theta_i, \theta_i') \tag{5}$$

where $I(a,b)$ is the indicator function, which is equal to 1 if $a = b$ and zero otherwise. $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are random variables that are distributed identically to the solutions in the solution set $\boldsymbol{\Theta}$. Defining distance in this sense means that the distance between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ is the number of decision variables that take different values between these two solutions. Given these definitions, within this paper convergence is defined as $\bar{d}(\Theta(t)) \rightarrow 0$ for $t \rightarrow T$, where *T* is the maximum number of iterations.

At a given iteration *t*, where $t = 1, \ldots, T$, the algorithm search can be considered to be concentrating its search efforts within the decision sub-space defined by the hypersphere

$$\Lambda(t) = \left\{ \boldsymbol{\theta} : d(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}(t)) < \frac{\bar{d}(\Theta(t))}{2} \right\} \tag{6}$$

where $\bar{\boldsymbol{\theta}}(t)$ is the center point of $\Theta(t)$ (i.e. $\bar{\boldsymbol{\theta}}(t)$ is the point in the decision space that posesses a minimum averaged distance to all solutions in $\Theta(t)$). Within the early stages of the search, $\Lambda(t)$ should cover the majority of the decision space, as limited information on promising decision space regions has been determined. As the search continues, $\Lambda(t)$ should gradually reduce, as the algorithm should progressively focus its search on exploring increasingly smaller promising regions of the decision space. Convergence that is too rapid represents an under-allocation of resources to exploring the search space. Convergence that is too slow represents an under-allocation of search effort to intensively exploit promising sub-regions. Within the final stages of the search, $\Lambda(t)$ should be focused within close neighborhoods of the best solutions found.

3) *Analytic predictor for ACO* convergence

The proposition presented in this section provides the theoretical foundation for the convergence controlled parameter adaption strategy in ACO_{CTC}. The basis of this strategy is the development of an analytic predictor of the expected value of the mean population distance (the outcome of Proposition 1), which can only be done for ACO, and is the reason why ACO is central to the proposed approach. As outlined in the next subsection, this analytic predictor enables determination of the required parameter adaptations to achieve the target convergence trajectory.

Within each iteration, the mean population distance $\bar{d}(\Theta(t))$ (i.e. degree of convergence) is a random variable that is dependent on the decision point probability values, as given by (1). The proposition below derives the direct analytic relationship between the expected value of $\bar{d}(\Theta(t))$ and the probability values at each decision point. As the probability values themselves are a function of the algorithm parameters, the following derived relationship provides an avenue to directly control the breadth of an ACO algorithm's search from iteration to iteration.

The work proposed within this paper is focused on optimization problems that involve a sequence of *N* independent decisions, where each decision is drawn from a finite set of options. That is, the decision space for problems considered in this work is defined as the set of solutions $\{\boldsymbol{\theta} = [\theta_1 \Lambda \theta_N] : \theta_i \in \{d_{i1},...,d_{iN_i}\}, i = 1,...,N\}$ where *N* is the number of decision variables, d_{ij} is the *j*-th option for decision *i* and N_i is the total number of options for decision *i*. Within the ACO framework, such problems are defined by a multi-graph structure $G(E,V)$ with edge set $E = \{l_{1,2}^{[1]},...,l_{1,2}^{[N_1]},...,l_{N-1,N}^{[1]},...,l_{N,N+1}^{[N_N]}\}$, where $l_{i,i+1}^{[j]}$ is the *j*-th link that connects node *i* to node *i*+1, and node set *V* = {1,…,*N*+1}. For this problem structure, note: node *i* is only connected to node *i* + 1 via the N_i links in the set $\{l_{i,i+1}^{[1]},...,l_{i,i+1}^{[N_i]}\}$; decision point *i* corresponds to node *i*, for *i* = 1, .., *N*; and link $l_{i,i+1}^{[j]}$ corresponds to option d_{ij} and pheromone value τ_{ij}.

Proposition 1: Consider a problem with multi-graph structure $G(E,V)$ with edge set $E = \{l_{1,2}^{[1]},...,l_{1,2}^{[N_1]},...,l_{N-1,N}^{[1]},...,l_{N,N+1}^{[N_N]}\}$ and node set *V* = {1,…,*N*+1}; a given set of pheromone values $\boldsymbol{\tau} = \{\tau_{11},...,\tau_{1N_1},...,\tau_{N1},...,\tau_{NN_N}\}$ where τ_{ij} is the pheromone on edge $l_{i,i+1}^{[j]}$; a set of *m* solutions $\Theta = \{\boldsymbol{\theta}^{[1]},..., \boldsymbol{\theta}^{[m]}\}$ where $\boldsymbol{\theta}^{[k]}$ corresponds to the solution generated by the k-th ant's tour through the graph $G(E,V)$ according to the probabilities in (1); and the mean population distance as defined in (4) based on the Hamming distance metric as given in (5). The expected value of the mean population distance for Θ conditioned on the pheromone value set $\boldsymbol{\tau}$ is

$$\mathrm{E}\left[\bar{d}(\Theta)\middle|\boldsymbol{\tau}\right] = N - \sum_{i=1}^{N}\sum_{j=1}^{N_i} p_{ij}^2 \quad (7)$$

Note that the expectation in (7) is over the probability space defined by the decision point probability values as given in (1), and is taken as conditional on $\boldsymbol{\tau}$ to remove its dependency on the stochasticity of the pheromone values (which are dependent on the history of the pheromone updates from the ACO search).
Proof. Taking the conditional expectation of (4) yields

$$\mathrm{E}\left[\bar{d}(\Theta)\middle|\boldsymbol{\tau}\right] = \frac{2}{m(m-1)}\sum_{k=1}^{m}\sum_{l=k+1}^{m}\mathrm{E}\left[d\left(\boldsymbol{\theta}^{[k]},\boldsymbol{\theta}^{[l]}\right)\middle|\boldsymbol{\tau}\right] \quad (8)$$

To evaluate the expectation in (8), two important observations are highlighted. Firstly, given the multi-graph problem structure, the ants' solution construction process consists of N independent decisions, where there is no path dependency of the decision point probability values (i.e. selection probabilities for decision point i are not dependent on the selected option for decision point i-1). Secondly, as shown in (1), the decision point probability values are a function of the pheromone set $\boldsymbol{\tau}$. For the proposed formulation, within an iteration, the pheromone set remains constant, and is only updated after all m ants have generated their solutions. This means that the decision point probability values are constant for each ant within an iteration. These two observations lead to the fact that the set of solutions in Θ are m independently and identically distributed samples arising from each ant's sojourn through the graph $G(E,V)$. This means that the expectation on the right side of (8) is identical for all pairs in Θ. Given that there are $m(m$-1)/2 pairs, (8) simplifies to

$$\mathrm{E}\left[\bar{d}(\Theta)\middle|\boldsymbol{\tau}\right] = \mathrm{E}\left[d\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\middle|\boldsymbol{\tau}\right] \quad (9)$$

Given (5), the right side of (9) is expressed as

$$\mathrm{E}\left[d\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\middle|\boldsymbol{\tau}\right] = N - \sum_{i=1}^{N}\mathrm{E}\left[I\left(\theta_i,\theta_i'\right)\middle|\boldsymbol{\tau}\right] \quad (10)$$

Now, the expectation in (10) is over the entire conditional joint probability space for the solution vectors $\boldsymbol{\theta}$ and $\boldsymbol{\theta}$'. Given the problem multi-graph structure $G(E,V)$, the elements of these vectors are independent, meaning that the joint probability of the solution vectors are just the product of the marginal probabilities (i.e., the probability of a variable without reference to the other variables) of the solution vector elements θ_i and θ_i', which are given by the decision probabilities in (1). Given this analytic expression for the marginal probability distributions of θ_i and θ_i', and the independence of these variables, the expectation of the indicator function in the summand of (10) is simply the summation of the product of the probabilities of θ_i and θ_i' over all events for which $\theta_i = \theta_i'$. That is

$$\mathrm{E}\left[I\left(\theta_i,\theta_i'\right)\middle|\boldsymbol{\tau}\right] = \sum_{j=1}^{N_i} p_{ij}^2 \quad (11)$$

where substitution into (10) yields the required form (7). Consequently, the proposed method can be applied to all ACO variants (and possibly other EAs that use the same probability-based decision policies).

The significance of Proposition 1 [namely, Eq. (7)] is that it demonstrates that, given the decision point selection probability values, the breadth of the algorithm's search through the search space can be analytically predicted. In other words, Proposition 1 shows that ACO parameters can be dynamically adjusted to change or control the algorithm's convergence behaviour in decision space. An approach for dynamically adjusting ACO parameters is developed in the next section.

4) Algorithmic framework of the ACO$_{CTC}$

From an algorithm control perspective, a corollary to Proposition 1 is that through changing the algorithm parameters, the probability values can be manipulated to achieve a target expected mean population distance. That is, given a desired convergence trajectory (defined as a pre-specified sequence of target mean population distances $\bar{d}_0(1),\ldots,\bar{d}_0(T)$), Proposition 1 allows the user to determine ACO parameter settings that will enable this behavior to be achieved. Given the form of the update equation in (2) and the probability function in (1), there are many different parameter adaption strategies to control an ACO algorithm's convergence that could be developed using Proposition 1. From (1) and (7), the expected mean population distance (i.e. the convergence predictor) can be expressed directly as the following function of parameter set $\Psi = \{\alpha,\beta,\rho\}$

$$\mathrm{E}\left[\bar{d}(\Theta)\middle|\boldsymbol{\tau}\right](\Psi) = N - \sum_{i=1}^{N}\frac{\sum_{j=1}^{N_i}\tau_{ij}^{2\alpha}\eta_{ij}^{2\beta}}{\left(\sum_{k=1}^{N_i}\tau_{ik}^{\alpha}\eta_{ik}^{\beta}\right)^2} \quad (12)$$

where the ρ dependency occurs through the pheromone update equation (2), and the notation $\mathrm{E}[\cdot|\cdot](\Psi)$ is used to indicate the parametric dependency of the expectation operator on Ψ. The proposed algorithm for implementing the ACO$_{CTC}$ is summarized in Fig. 2. The algorithm requires the standard inputs (problem characteristics and ACO parameters) with the addition of a target trajectory [meaning a pre-specified target mean population distance at each iteration point $\bar{d}_0(1),\ldots,\bar{d}_0(T)$]. The target mean population distance sequence defines the algorithm's convergence trajectory, and should be a gradually decreasing function from the maximum distance $\bar{d}_0(1)$ at t = 1, down to the target converged value $\bar{d}_0(T) = 0$ (or near 0). This sequence is a user-specified sequence, where different options for the sequence of target mean population distance values are outlined in the experimental methodology of Section IV. Standard ACO initialization is undertaken in Step 2 (e.g. initializing the link pheromone values to τ_0), followed by the iterative search (Steps 3 to 9). Within Step 4, at the beginning of every iteration, the new values of $\Psi(t)$ are computed by minimizing an error norm (e.g. the absolute value of the difference) between the expected mean population distance $\mathrm{E}\left[\bar{d}(\Theta)\middle|\boldsymbol{\tau}(t)\right](\Psi)$ and the target mean population distance $\bar{d}_0(t)$. Selecting this setting of the parameter set $\Psi(t)$ ensures that the expected mean population distance (which is parametrically dependent on $\Psi(t)$) will be as close as possible to the target mean population distance, which implies that the colony solution set $\Theta(t)$ constructed in Step 6 adheres to the desired criteria, such that $\bar{d}(\Theta(t)) \approx \bar{d}_0(t)$. Steps 5 to 8 represent the standard ACO steps, but where the probability values are calculated with the updated $\Psi(t)$.

1. **Input:** Problem characteristics, algorithm parameters, and the pre-specified target mean population distance sequence $\bar{d}_0(1),\ldots,\ \bar{d}_0(T)$
2. Initialize pheromone set $\boldsymbol{\tau}(t{=}0)$
3. **For** $t = 1, \ldots, T$
4. Compute $\Psi(t)$ from
$$\Psi(t) = \arg\min_{\Psi}\left\|\bar{d}_0(t) - \mathrm{E}\left[\bar{d}(\Theta)\middle|\boldsymbol{\tau}(t)\right](\Psi)\right\| \quad (13)$$
where $\mathrm{E}\left[\bar{d}(\Theta)\middle|\boldsymbol{\tau}(t)\right](\Psi)$ is computed from (12)
5. Compute probability values from (1)
6. Construct colony solution set $\Theta(t)$
7. Evaluate objective values $f(\boldsymbol{\theta}) + p(\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta(t)$
8. Update pheromone set $\boldsymbol{\tau}(t{+}1)$ from (2)
9. **End**

Fig. 2. Proposed convergence-trajectory-controlled adaptive ACO (ACO_{CTC})

A successful application of the proposed ACO_{CTC} has to pre-specify a suitable convergence trajectory, with which the algorithm's run-time searching is followed. However, the determination of the best trajectory can be challenging in practice, as it is associated with the fitness landscape properties of the problem being solved, and hence can be problem dependent. Therefore, one disadvantage of the proposed convergence-trajectory controlled framework is that it cannot automatically offer an appropriate targeted convergence trajectory according to fitness landscape characteristics of the given problem. Another limitation of the proposed ACO_{CTC} is the use of mean population distance to represent the algorithm's convergence. This is because the mean population distance can only indicate the overall distribution of the solutions in the decision space, while it cannot fully represent the composition or clusters of denser solution regions within the algorithm search.

IV. Experimental Methodology

A. Overview

The purpose of the experimental methodology is (i) to demonstrate the ability of the proposed ACO_{CTC} algorithm to follow the pre-defined convergence trajectories in decision space; (ii) to assess the impact of different pre-defined convergence trajectories on final solution quality for different instances of a real-world optimization problem type, (iii) to assess the impact of the proposed adaptive strategy on final solution quality for different optimization scenarios by comparing its performance with the equivalent fixed-parameter ACO algorithm, and (iv) to assess the absolute quality of the solutions obtained by the proposed algorithm relative to the previously best-found solutions in literature. It should be noted that the purpose of this benchmarking exercise is not to determine whether the proposed ACO_{CTC} algorithm is able to find better solutions than other reported algorithms, but to assess the quality of the solutions that can be obtained with limited computational budgets for different convergence trajectories, as mentioned above.

The real-world problem type considered is the optimal design of water distribution systems, as (i) this corresponds to a real-world problem for which trade-offs between solution quality and available computational budgets are a significant issue [20], (ii) it is a problem type to which a wide variety of metaheuristics have been applied previously [27-34], thereby providing a benchmark against which the quality of the solution obtained using the proposed ACO_{CTC} can be assessed, (iii) guidelines are available for determining optimal parameters for the non-adaptive version of the algorithm [35], enabling the impact of the proposed adaptive strategy to be isolated, and (iv) it adheres to a multi-graph problem structure, as required by Proposition 1.

For the numerical studies presented in this paper, the ACO_{CTC} algorithm was based on the elitist-rank ant system (AS_{rank}) [17], and is denoted by $AS_{rank+CTC}$ (where the "+CTC" indicates that the algorithm uses convergence-trajectory control to adapt its parameters). The reason for the selection of AS_{rank} is that it has been found to be one of the most successful versions of ACO applied to the WDSDP [17], [35]. A total of eight different convergence trajectories and three computational budgets were applied to six WDSDPs, with the number of the decision variables ranging from $N = 21$ to 1274.

As shown in Fig. 3, for the four relatively small case studies (NYTP, HP, FOS and PES), 30 runs, with different starting random number seeds, were performed to enable a statistically significant characterization of algorithm performance. For the large BN and KL problems with 454 and 1274 decision variables respectively, 10 runs were undertaken due to the long simulation times associated with these two case studies. Given that 1000 simulations of the KL problem required approximately 4 seconds using a Pentium PC (Inter R) Core(TM) i7-5500U CPU at 2.4GHz, and the total number of hydraulic simulations was 5.4×10^8 (calculated from parameter values shown in Fig.3, Tables I and II), the total time used by the KL case study was approximately 25 days. The designed convergence trajectories, case studies, and implementation of the proposed ACO_{CTC} method are discussed below.

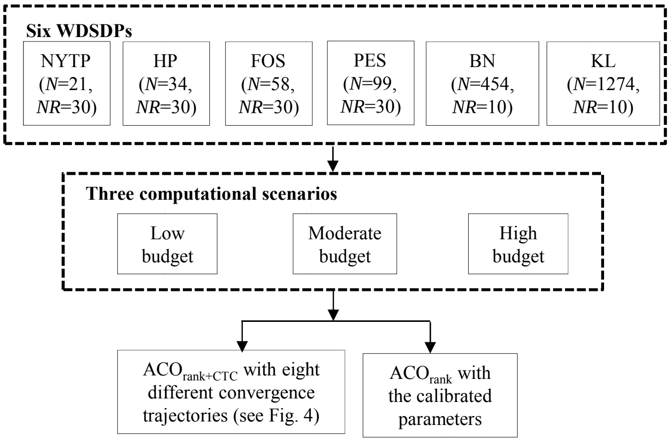

Fig. 3. Flowchart of the experiment study for assessing the performance of the proposed $\mathrm{ACO}_{\mathrm{rank+CTC}}$. The N below the case study indicates the number of the decision variables, and NR indicates the number of replicates with different random number seeds.

B. Convergence Trajectories

A crucial input to the proposed $\mathrm{ACO}_{\mathrm{CTC}}$ algorithm are the convergence trajectories as characterized by the sequence of target mean population distance values $\bar{d}_0(1),..., \bar{d}_0(T)$. Clearly $\bar{d}_0$ should be large early in the search, and small in the final stages of the search, but the trajectory between these bounds is open for specification. To enable a broad characterization of convergence paths, a family of trajectory cases has been developed using the following functional form

$$\bar{d}_0(t) = D_0\left(1-\frac{t}{T}\right)^a \tag{14}$$

where D_0 is the maximum distance, and a is an exponent parameter. For this family of curves parameterized by a, five cases were considered (as depicted in Fig. 4), namely: a = 1/5, which is a highly explorative case (diversification) that maintains a broad search for most of the run-time, and only converges in the final stages; a = 1, which provides a linear (balanced) convergence throughout the run-time; a = 5, which is a highly exploitative case (intensification) that converges quickly and spends the majority of the run-time within a small region in the vicinity of the best found solution; and the intermediate explorative (a = 2/3) and exploitative (a = 3/2) cases.

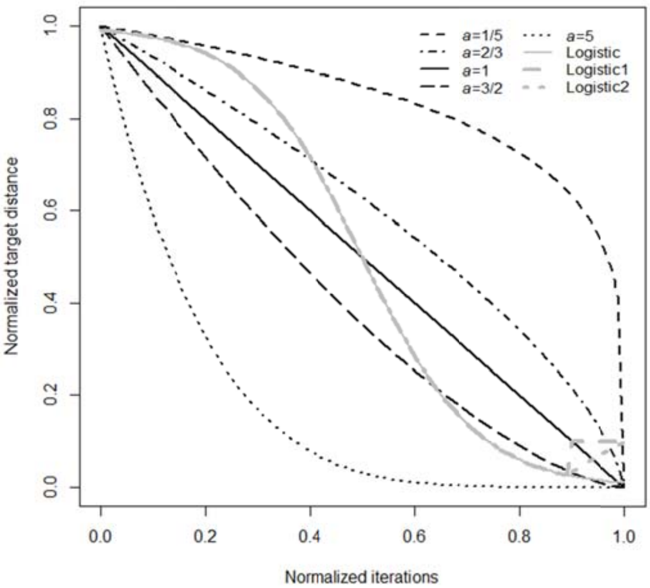

Fig. 4. Convergence trajectories as characterized by the normalized target mean population distance values. Curves are: strongly explorative (a = 1/5, upper dashed line); explorative (a = 2/3, dash-dot line); balanced (a = 1, solid line); exploitative (a = 3/2, lower dashed line); and strongly exploitative (a = 5, dotted line); logistic (grey solid line); logistic with a sudden increase at the end iterations (grey dashed line, logsitic1); logistic with a gradual increase at the end iterations (grey dotted line, logistic2).

In addition to the cases outlined above, three logistic convergence trajectories were considered: one with an overall monotonic decease (grey solid line); the second with a sudden increase in population diversity at the end stage of search (grey dashed line, denoted as logistic1); and the third involving a gradual increase in population diversity during the final searching phase (grey dotted line, denoted as logistic2). The first logistic trajectory reflects the typical behavior observed for ACO applied to WDSDPs, which maintains a broad initial search, followed by a rapid convergence, and a final intensive search [17]. The latter two trajectories represent perturbations to the solutions during the final searching stage, aiming to examine whether an increase in mean population distance (and consequent increased emphasis on exploration) at the end iterations can enhance the solution quality, or not. The starting point of the mean population distance increase was selected to be at 90% of the allowed total iterations, with the final mean population distance value at a level of 10% of the initial mean population distance. The initiation time of the increase was selected as ACO was observed to stagnate in solution quality after appropriately 90% of the given computational budgets.

C. The Water Distribution System Design Problem

Water distribution systems consist of networks of electro-mechanical or hydraulic elements (e.g. pipes, pumps, valves, tanks) that deliver potable water from the treatment plant to consumers. The classical WDSDP is an idealized embodiment of a distribution system design problem. It typically involves the selection of pipe sizes for each link in the network to minimize overall cost (i.e. smaller pipes are preferred) in a way that satisfies the design constraint of the adequate provision of water demands and pressure heads to all consumers (smaller pipes tend to cause infeasible designs). The WDSDP is therefore a constrained combinatorial problem, where significant problem difficulty is attributed to the highly complex structure of the feasible design region, as governed by the nonlinearity of the design constraints. Sizes typically considered in the literature range from tens to hundreds of links [25].

A WDSDP has an associated pipe-network graph $G(L,M)$ consisting of a set of links $\{l_1,...,l_N\} \subseteq L$ for which design decisions are required, a set of source nodes $M_o \subset M$ (e.g. reservoirs or water treatment plants), and a set of demand nodes $M_d \subset M$ (e.g. consumers). For each of these pipe links i = 1, …,N each decision consists of the selection of a design option $\theta_i \in \{d_{i1},..., d_{iN_i}\}$, where the options d_{ij} corresponds to either no action, the selection of a pipe of a given diameter and material, or the remediation of a pipe (this is specified by the particular case study). The design problem can be stated as

$$\min_{\boldsymbol{\theta}} \; f(\boldsymbol{\theta}) = \sum_{i=1}^{N} C(\theta_i) L_i \quad (15)$$
$$\text{subject to} \quad h_j(\boldsymbol{\theta}) \geq \overline{h}_j, \; j \in M_d$$

where $\boldsymbol{\theta}$ is the set of decisions, $C(\theta_i)$ is the cost per unit length of implementing decision θ_i for link i, and L_i is the length of link i, $h_j(\boldsymbol{\theta})$ is the pressure head at node j for design decision set $\boldsymbol{\theta}$ and $\overline{h}_j$ is the minimum allowable pressure head for node j.

The inequality constraints in (15) represent the design requirements that the pressure head at each consumer node be greater than the minimum required head. Computing the vector of nodal pressure heads for a given design $\boldsymbol{\theta}$, involves solving the fundamental physical equations governing network steady-state hydraulics. This set of equations (many of which are non-linear) can be very large for real-world WDSP applications (e.g. [20]) and is typically solved using Newton-based methods (e.g. [36]), which can require significant computational effort. An external engine is often used to compute the nodal pressure heads (e.g. EPANET2.0 [35]).

D. Case Studies

Details of the six WDSDP case studies investigated are shown in Table I. These case studies were selected as they (i) represent different types of WDSDPs (e.g., the NYTP considers the rehabilitation of the existing network via pipe duplication, the HP has a very small feasible search space, the FOS and PES are typical design problems that have large feasible regions; the Balerma network (BN) is an adaptation of an irrigation network, the KL represents a real-world problem, with an extremely large and difficult search space), (ii) have been used to test the performance of a range of metaheuristics in a large number of studies [27-34], and (iii) cover a range of search space sizes, ranging from 1.93×10^{25} to 10^{1274} as shown in Table II, thereby enabling the effectiveness of the proposed approach for the three selected computational budgets to be assessed more clearly. The current best-known solutions for the NYTP, HP, FOS, PES BN and KL problems are \$38.64 million [27], \$6.081 million [37], \$29,577 [38], \$1.81 million [38], \$1.923 million [39] and \$8.307 million [25], respectively.

TABLE I
CASE STUDY PROPERTIES

Case Study	Network Properties				Maximum Iteration (generation) Cases, T
	No. of decisions N	No. of options	Search Space Size	No. of nodes	T_{low}, T_{med}, T_{high}
NYTP	21	16	1.93×10^{25}	20	200, 500, 1000
HP	34	6	2.86×10^{26}	32	400, 1000, 2000
FOS	58	22	7.25×10^{77}	37	400, 1000, 2000
PES	99	13	1.91×10^{110}	71	400, 1000, 2000
BN	454	10	10^{454}	447	1000, 2000, 3000
KL	1274	10	10^{1274}	936	1000, 2000, 3000

In order to assess the ability of the proposed ACO_{CTC} algorithm to control convergence trajectories in decision space, a range of computational budgets, as specified by the maximum iteration number T, was considered for each case study. The selected cases for T were based on the convergence time observed for the standard ACO using the recommended parameter settings [35], denoted as T_o, and are outlined as follows: $T_{low} \ll T_o$ (standard ACO would not converge); $T_{med} \approx T_o$ (standard ACO would approximately converge) and; $T_{high} \gg T_o$ (standard ACO would converge prematurely within the allowed run-time). As outlined in Table I, the high maximum allowable iterations (T_{high}) are 1,000 for the NYTP, 2,000 for the HP, FOS and PES problems, and 3,000 for the BN and KL case studies. For the first four problems, the low (T_{low}) and medium (T_{med}) computational budgets are 20% and 50% of T_{high}, respectively. For the large BN and KL problems, T_{low}=1,000 and T_{med}=2,000 were used. It should be noted that one of the advantages of the proposed approach is that convergence at the end of the search is guaranteed, irrespective of computational budget used.

E. Algorithm Implementation and Comparison

To ensure that the controlled convergence tracks the best information, elitism is a necessary component of the proposed ACO_{CTC} method (as for many EAs). As such, the elitist-rank ACO algorithm (AS_{rank}, see [17] for details) was adopted for the pheromone update strategy. That is, the pheromone update is given as

$$\Delta\tau_{ij}(t) = \sigma \Delta\tau\left(l_{ij}, \boldsymbol{\theta}^{gb}(t)\right) + \sum_{k=1}^{\sigma-1} (\sigma - k) \Delta\tau\left(l_{ij}, \boldsymbol{\theta}^{(k)}(t)\right) \quad (16)$$

where σ is the number of elitist ants, $\boldsymbol{\theta}^{(k)}(t)$ is the k-th best solution found in iteration t, and the pheromone update equation is

$$\Delta\tau(l,\boldsymbol{\theta}) = \begin{cases} \dfrac{Q}{f(\boldsymbol{\theta}) + p(\boldsymbol{\theta})} & \text{if } l \in \boldsymbol{\theta} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where Q is the pheromone reward factor. The advantage of the AS_{rank} update scheme is that it biases the search towards promising regions of the search space (by updating only good solution components in a proportional manner), but encourages a focused exploration by including the top σ-1 unique solutions in the update scheme.

The proposed ACO_{CTC} method outlined in Fig. 2 is generic and is applicable to all parameters in (12). In other words, multiple parameters can be modified to ensure that ACO_{CTC} correctly tracks the specified convergence trajectory. However, the present study only considers the parameter α of AS_{rank} due to its greater impact on the performance of ACO algorithms applied to the WDSDP ([17], [35]), as well as the more direct control it provides in using the learned information (pheromone) within the searching process compared with other parameters. For example, as α tends to zero, the probability functions become more uniform, and exploration is encouraged. However, as α is increased, greater emphasis is placed on learned information with historically good solutions having a higher probability of selection, and as a result, intensification is encouraged. Furthermore, while similar outcomes could have been achieved by varying multiple parameters simultaneously, modifying a single parameter is more desirable in this instance. This is because the use of a single parameter, instead of simultaneously considering multiple parameters, significantly reduces the numeric complexity of the parameter adaption step. This is explained by Fig. 2, where it is shown that the adaption step involves solving a real-valued minimization problem of the same dimensions as the number of controlling parameters. In other words, as the primary purpose of the adaptive parameter control is to achieve a pre-defined convergence trajectory, the fact that this can be achieved by varying a single parameter is an advantage and indicates that there is no need to adapt any of the other parameters. The minimization problem governing the parameter adaption, as outlined in Step 4 of Fig. 2, was solved using Newton's method in this study, although other methods could have been used. It was observed in the simulation study that solving Equation (13) in Fig. 2 involved only a negligible amount of time in comparison to the time required for the computation of the objective function, which required a full hydraulic simulation.

To determine the effectiveness of the proposed $AS_{rank+CTC}$, its performance was compared with that of the standard AS_{rank} with the best parameterization, but no convergence trajectory control, across different computational budgets. As mentioned previously, this enables the impact of the use of the proposed convergence trajectory control to be assessed. The AS_{rank} parameter settings were determined based on the guideline presented in Zecchin *et al*. [17] (see Table II), with the exception of β = 0.25 and σ = 5 being selected based on the results of a preliminary analysis in order to maximize algorithm performance and thereby enable a fair comparison. The initial parameter values of the proposed $AS_{rank+CTC}$ were identical to those in AS_{rank} (see Table II), with the only difference being that the value of α was dynamically adjusted throughout the optimization run to enable the target convergence trajectory to be followed (as discussed above). For WDSDP applications, an active penalty function p (see Equation 3) was used to deal with infeasible solutions [i.e. solutions that do not satisfy the inequality constraints in (15)]. The penalty function was taken as proportional to the maximum pressure head violation [34], meaning that it is proportional to the maximum value of $\bar{h}_j - h_j(\boldsymbol{\theta})$, for any $j \in M_d$ for which $h_j(\boldsymbol{\theta}) < \bar{h}_j$. The maximum allowable number of simulations for each computational scenario applied to each case study was outlined in Table II.

TABLE II
THE PARAMETERIZATION OF AS_{RANK} (THE INITIAL PARAMETER VALUES OF THE PROPOSED $AS_{RANK+CTC}$, $\alpha = 1, \beta = 0.25, \rho = 0.98, \sigma = 5$ WERE USED FOR ALL PROBLEMS)

Case Study	Algorithm parameters			
	m	Q	τ_0	Maximum allowable number of simulations for three computational scenarios
NYTP	90	3.0×10^8	139.5	18,000, 45,000, 90,000
HP	100	1.1×10^7	25.7	40,000, 100,000, 200,000
FOS	270	1.5×10^6	2047.2	108,000, 270,000, 540,000
PES	350	1.9×10^7	378.7	140,000, 350,000, 700,000
BN	500	2.2×10^7	746.4	500,000, 1,000,000, 1,500,000
KL	1000	1.1×10^8	1381.4	1,000,000, 2,000,000, 3,000,000

A statistical analysis using the Wilcoxon test was conducted to check whether the performance differences between the proposed $AS_{rank+CTC}$ and the standard AS_{rank} were statistically significant. It is noted that the EPANET2.0 input files and decision variable information for all case studies have been uploaded as supplemental documents, in which the network layout, decision variable options, Hazen-Williams coefficients, and configuration (pipe sizes) of the best found solutions were included. This detailed information allows hydraulic verification (e.g. feasibility) of the best solutions identified in the present study.

V. RESULTS AND DISCUSSION

Within this section, firstly the ability of the proposed algorithm to follow pre-defined convergence trajectories in decision space is demonstrated for one of the case studies (the NYTP); secondly, the impact of different convergence trajectories, computational budgets and the proposed parameter adaptation strategy on solution quality are assessed; and finally, a summary and implication of the results are outlined.

A. Example of the Ability to Follow Pre-defined Convergence Trajectories

For this example application (i.e. the NYTP), the convergence behavior of $AS_{rank+CTC}$ with a linear convergence trajectory is explored and compared with that of the standard AS_{rank} for the range of maximum iteration cases in Table I. Fig.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation

5 illustrates the run-time search behavior of $AS_{rank+CTC}$, where Fig. 5 (a) shows the convergence trajectory of $AS_{rank+CTC}$ (controlled) for three different allowable computational budgets compared with that of the standard AS_{rank} (i.e. uncontrolled convergence and constant α throughout the run), and Fig. 5 (b) shows the variation in the adapted α parameter for $AS_{rank+CTC}$ versus the number of iterations for the three linear trajectories considered.

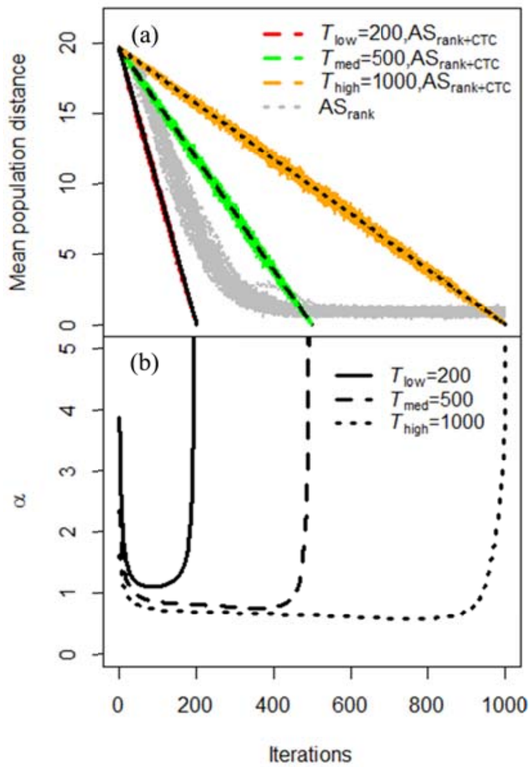

Fig. 5 Run-time search properties of the proposed $AS_{rank+CTC}$ applied to the NYTP case study with the designed linear convergence trajectory under different computational scenarios. Plot (a) shows: the target mean population distance $\bar{d}_0(t)$ (black lines) and the observed mean population distance $\bar{d}(\Theta(t))$ for the proposed algorithm (red, green and orange lines); and the observed mean population distance for the standard algorithm (grey dots showing a non-linear response). Plot (b) shows the variation of $\alpha(t)$ for each scenario of T for the proposed algorithm. Results are based on 30 runs.

Within Fig. 5 (a), the black lines indicate the target linear convergence trajectory for $AS_{rank+CTC}$, and the red, green and orange lines indicate the observed convergence trajectories for 30 runs of the $AS_{rank+CTC}$ and the standard AS_{rank} (grey dotted lines deviating from the black lines). Consistent with [17], AS_{rank} is seen to converge rapidly in the initial stages of the search, where near convergence is reached around iteration t = 380, after which AS_{rank}'s convergence plateaus, and for the remainder of the run-time its search is limited to a very small sub-region in the decision space (as indicated by the small mean population distance values). This natural behavior (no convergence trajectory control) shows that the majority of the computational time is allocated to searching within only small regions of the decision-space.

In contrast to AS_{rank}, the convergence trajectories for $AS_{rank+CTC}$ are observed to follow the target linear convergence trajectories (the observed $AS_{rank+CTC}$ trajectories are the grey lines that fluctuate closely about the black lines, respectively) very closely. For all computational budgets considered (T = 200, 500 and 1000), convergence was achieved by the final stage of t = T. This means that $AS_{rank+CTC}$ successfully avoided non- and pre-convergence, unlike AS_{rank}, which did not converge for the shorter computational budget (T = 200), and converged prematurely for the longer computational budget (T = 1000). Comparing the trajectories of $AS_{rank+CTC}$ and AS_{rank}, combined with the fact that a consistently good match between the target and observed values for $AS_{rank+CTC}$ was achieved, demonstrates that the proposed parameter adaption scheme can indeed control the convergence trajectory in decision space, and drive $AS_{rank+CTC}$ to converge within a pre-specified computational budget along a pre-specified trajectory.

The averaged values of $\alpha(t)$ for $AS_{rank+CTC}$ over the 30 runs for the observed convergence trajectories in Fig. 3(a) are shown in Fig. 5(b) for each of the runs with different computational budgets. A similar pattern observed from this figure is that the α parameter is high in the initial stages of the search, followed by a quick reduction to a lower plateau (around α =1), and finally by a sharp increase to a high α value once again. High values of α indicate a greater emphasis on the learned information to guide the search [the pheromone values in the probability values, as in (1)], and as such are indicative of an exploitative behavior. Interpreting Fig. 5(b) from this perspective means that, for $AS_{rank+CTC}$: an initially high value of α was needed to first drive convergence; followed by a reduced emphasis on intensification, as the natural learning operations within $AS_{rank+CTC}$ continued to drive convergence; and in the final stages of the search, a much greater emphasis on intensification was required again to intensify the search within the small regions about the best found solutions.

To demonstrate the uniqueness of the parameter adaption of α, as in Step 4 of the proposed algorithm framework, Fig. 6(a) shows the profile of the expected mean population distance error, $\bar{d}_0(t)$- $\mathrm{E}\left[\bar{d}(\Theta)\middle|\tau(t)\right](\alpha)$, across different α values, for a typical run of the proposed method, at iteration t=100 (black line) and 180 (grey line). It is seen that the profile is smooth with a unique root, suggesting low numeric complexity in obtaining $\alpha(t)$ within each iteration. To further explore the influence of the proposed method on the evolution of the decision probability functions, Fig. 6(b) presents the probability distribution of the decision options for decision point i=7 with α = 1.1 and α =1.9. As shown in Fig. 6(b), for a constant pheromone value, the probability distribution varies as a function of α, indicating that the method of using α to alter the probability distribution (i.e. population diversity) of the offspring solutions is effective and that there is no need to control other parameters in order to achieve the desired convergence trajectory.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation

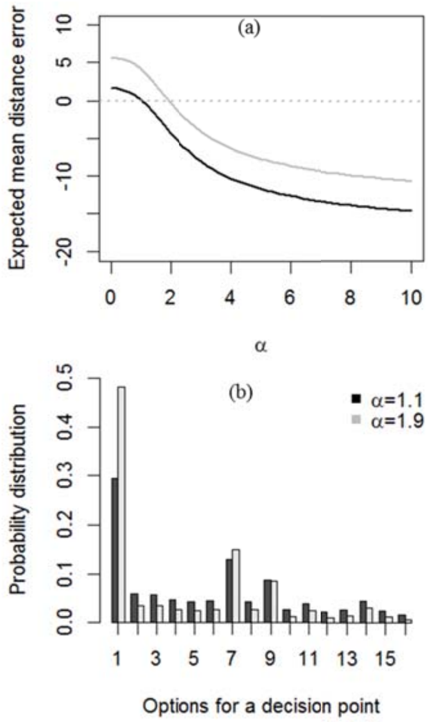

Fig. 6 Search properties of the proposed $AS_{rank+CTC}$ at iterations t = 100 (black line) and 180 (grey line) for the NYTP case study with the designed linear convergence line for a computational budget of T=200. Plot (a) shows the variation of the mean population distance error $\bar{d}_0(t) - \mathrm{E}\left[\bar{d}(\Theta(t))\middle|\tau(t)\right](\alpha)$ with α. Plot (b) shows the probability values at decision point i = 7 determined for two values of α.

The convergence properties of the proposed method applied to other case studies, and target convergence trajectories, were similar to those shown in Figs. 5 and 6. This demonstrates that the proposed parameter-adaptive strategy is able to enable the ACO convergence to follow the designed trajectories at the iteration level.

B. Performance for Different Convergence Trajectories and Computational Budgets

Boxplots of the performance results (as measured by the best cost found) of the proposed $AS_{rank+CTC}$,with the eight targeted convergence trajectories, and the standard AS_{rank} applied to the six WDSPs, are given in Figs. 7-12. The current best-known solutions previously reported for each case study are shown by grey dotted lines. A common pattern from all case study results is that convergence trajectories significantly affected the searching quality across the different computational budgets (as given in Table I). It is also noted that the performance of the proposed $AS_{rank+CTC}$ with three logistic convergence trajectories slightly varied for the six problems, with cases of mean population distance increase during the end searching stage showing improved performance compared to that with a monotonic reduction in mean population distance throughout the optimization process. This is especially the case for relatively low computational budgets, as a stochastic-searching algorithm with smaller maximum iterations T is more likely to converge to local optimal regions, and hence a slight increase in searching distance during the final stage is observed to find slightly better solutions.

As stated previously (Fig. 4), for the proposed $AS_{rank+CTC}$, convergence trajectories with a < 1 indicate that diversification dominates convergence behavior, trajectories with a >1 suggest that a large proportion of the search focuses on intensification, and a =1 implies a balanced effort between diversification and intensification throughout the search. It was observed that, for each run of the $AS_{rank+CTC}$ with different pre-defined convergence trajectories (not for logsitic1 and logsitic2), the majority of the population members (ants) tended to converge at an identical optimal solution at the end of the optimization run. For example, for the four smallest problems (NYTP, HP, FOS and PES), an average of 97% of the algorithm's searching members have converged to the same final solution (not necessarily the global optimal solution) in their runs. This percent value was slightly lower (94%) for the two large problems (BN and KL) due to the significant increase in the number of the decision variables. This is expected, as the proposed algorithm forced convergence during the later searching stage by adapting values of the parameter α (i.e., the increase in the value of α as illustrated in Fig. 5(b)), theoretically leading to convergence to a single point in decision space (few solutions might be generated in a close proximity to this point due to the stochastic nature of the search).

For the cases of $T = T_{min}$ (very low computational budgets), as shown in sub-plots (a) of Figs 7-12, the proposed $AS_{rank+CTC}$ with the designed convergence trajectories of a = 2/3 and 1 (see Fig. 4) exhibited overall better performance than the standard AS_{rank} in terms of average cost solutions for all case studies. For the NYTP, $AS_{rank+CTC}$ with a = 2/3 was able to find its current best-known solution ($38.64 million) once over 30 runs, which is 1.11% lower than the best solution located by the standard AS_{rank} ($39.07 million). For the HP problem (a difficult problem due to its small feasible region [37]), the standard AS_{rank} was unable to find any feasible solutions when such a limited computational budget was used. In contrast, $AS_{rank+CTC}$ was capable of finding the feasible region for all of the eight specified convergence trajectories. As displayed in Fig. 7(a), $AS_{rank+CTC}$ with a = 2/3 found a solution that deviated from the current best-known solution ($6.081 million) by only 1.4%, despite using such limited computational resources. For the FOS problem, the proposed method with a = 2/3, 1, and 3/2 outperformed the standard AS_{rank} in terms of both the mean and best cost solutions over the 30 runs and for the PES problem, the best and average cost solutions provided by the proposed method were $1.83 and $1.87 million, respectively, which are approximately $0.03 million lower than those from the standard AS_{rank}. For the largest BN and KL problems with 454 and 1274 decision variables respectively, $AS_{rank+CTC}$ with a = 2/3 exhibited significantly improved performance relative to the standard AS_{rank} in terms of both the mean and the best cost solutions over the 10 runs.

Another common pattern across the results for very low computational budgets is that $AS_{rank+CTC}$, with a highly explorative convergence trajectory (a = 5), did not perform as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation

well as $AS_{rank+CTC}$ with the other convergence trajectories. This finding demonstrates the importance of exploitation in producing good solutions for scenarios with computationally limited budgets. That is, in computationally limited scenarios, too much emphasis on exploration inhibits an algorithm's ability to quickly learn about the objective space properties, consequently diminishing the algorithm's performance.

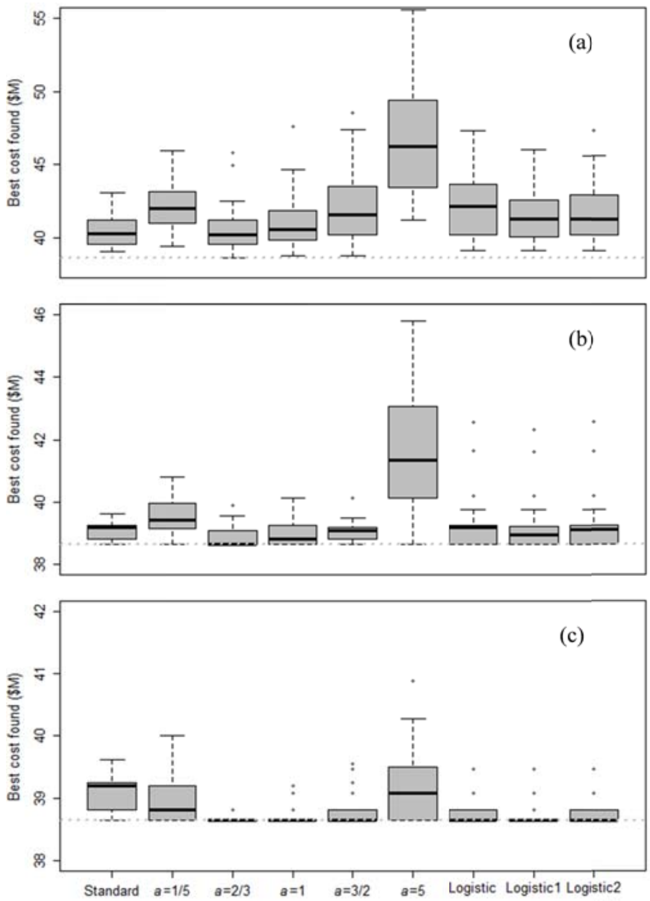

Fig 7. Comparison of algorithm performance applied to the NYTP for a range of maximum number of iterations: (a) T = 200; (b) T = 500; and (c) T = 1000. Results are based on 30 runs. Current known best solution is given by the horizontal dashed line.

When moderate computational resources were allowed ($T = T_{med}$), the proposed $AS_{rank+CTC}$ with convergence trajectories possessing relatively balanced diversification and intensification (including a = 2/3, 1, 3/2 and logistic) consistently outperformed the standard AS_{rank} for the six WDSDPs considered. The advantage of $AS_{rank+CTC}$ is more prominent for the difficult (HP, Fig.7(b)) and large (PES, BN and KL, Fig. 10(b) -12(b)) problems, as demonstrated by, for example, $AS_{rank+CTC}$'s ability to locate better solutions than the previously reported best solution for the PES and KL problems. The performance of the proposed $AS_{rank+CTC}$ with the extreme exploitative and explorative trajectories (a = 1/5 and 5) was unsatisfactory for the NYTP, PES, BN and KL problems, but was reasonable for the other two case studies. The fact that the proposed $AS_{rank+CTC}$ with a = 2/3, 1, 3/2 and logistic convergence consistently outperformed the standard AS_{rank} [even though the computational budget for this scenario was determined based on the natural convergence behavior of the standard AS_{rank} (i.e. the AS_{rank} algorithm was approximately converged at the completion of the moderate computational budgets)], highlights the important role that the convergence trajectory plays in determining solution quality, and not just the fact that the algorithm has converged.

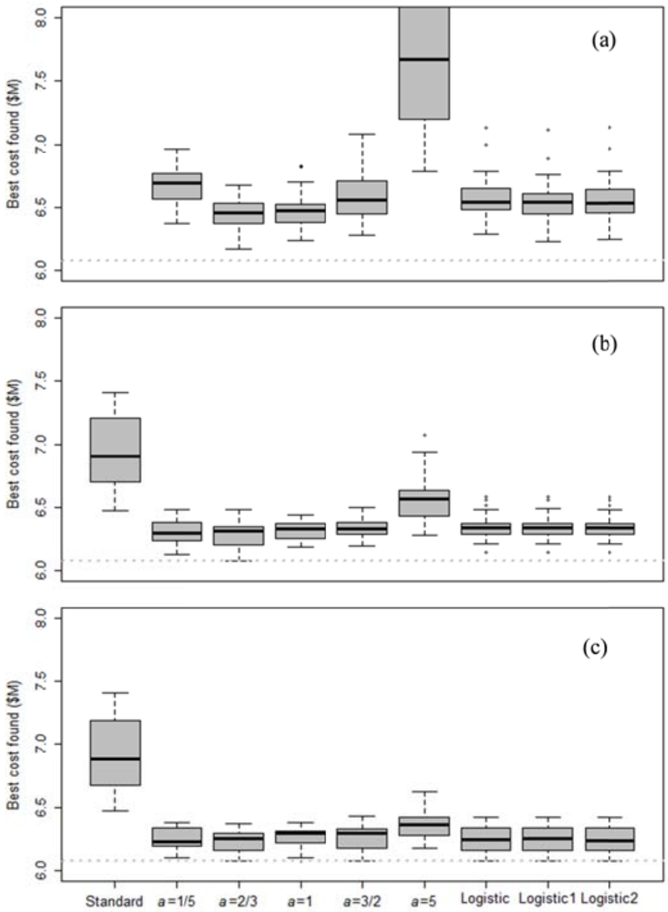

Fig 8. Comparison of algorithm performance applied to the HP for (a) T = 400; (b) T = 1000; and (c) T = 2000. Results are based on 30 runs. Current known optimum solution is given by the dashed line. No feasible solutions were found for the standard AS_{rank} when T=400.

When large computational resources were permitted, the proposed method exhibited superior performance to the standard AS_{rank} for all six case studies, irrespective of the value of a, apart from a =5 for the PES and BN, as illustrated in Fig. 7(c) -12(c) (T=T_{max}). This can be explained by the observed premature convergence associated with the standard AS_{rank} using the recommended parameter settings [35], which is indicative of the standard algorithm not allocating sufficient resources to exploring the search space. In contrast, the proposed $AS_{rank+CTC}$ method was able to dynamically adjust the searching effort between diversification and intensification according to the available computational budgets, thereby fully utilizing these resources to identify the best possible solutions. The performance of $AS_{rank+CTC}$ was particularly good for the PES and KL problems, where lowest costs of \$1.79 and \$8.02

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation

million were identified, which are 0.02 and 0.29 million lower respectively than the previous best solutions reported by Wang *et al*. [38] and Bi *et al*. [25]. For the BN problem, the cost of the best solution found by the proposed method was \$2.02 million, which was only 5% higher than the previous best solution identified by a hybrid optimization technique, where a differential evolution algorithm was combined with nonlinear programming [39]. Despite this, the solution found by the proposed method was lower than the optimal solutions found by the majority of non-hybrid meta-heuristics using similar computational overheads (e.g. 12.2% lower than the best solution found using the genetic algorithms [37]). In addition, the average cost of the optimal solutions obtained in this paper was \$2.14 million, which was obtained using 1.5 million evaluations, whereas the best solution obtained by the genetic algorithms with a cost of \$2.31 million used a computational budget of 10 million evaluations [39]. The Wilcoxon test results showed that convergence trajectories generated using a = 2/3 and 1 (Fig. 4) consistently exhibited statistically significantly better performance (at the 10% level) for all case studies across all computational budgets as shown in Figs. 7-12, except the NYTP problem with a low computational budget (Fig. 7(a)).

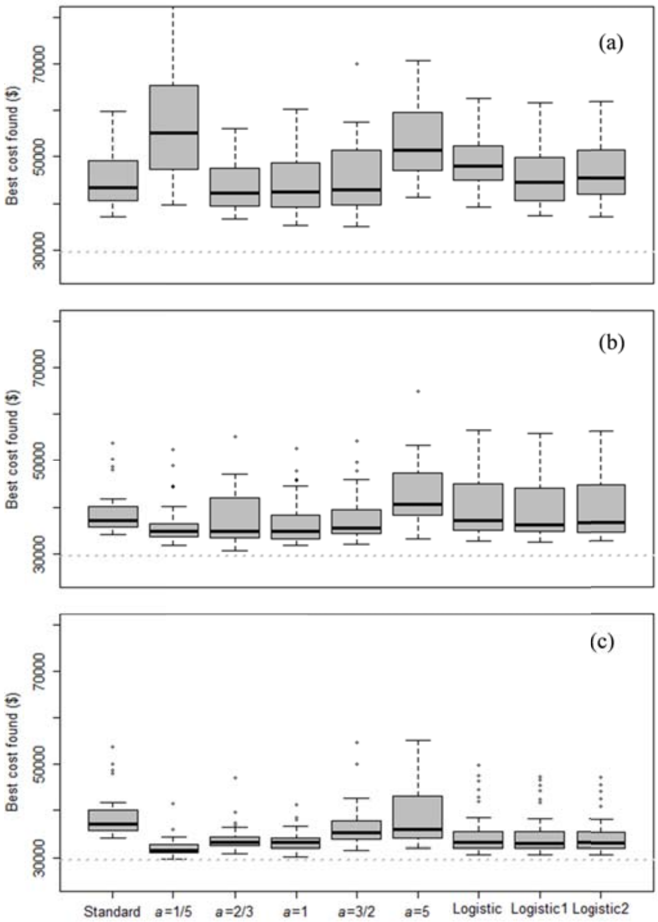

Fig 9. Comparison of algorithm performance applied to the FOS for: (a) T = 400; (b) T = 1000; and (c) T = 2000. Results are based on 30 runs. Current known optimum solution is given by the dashed line.

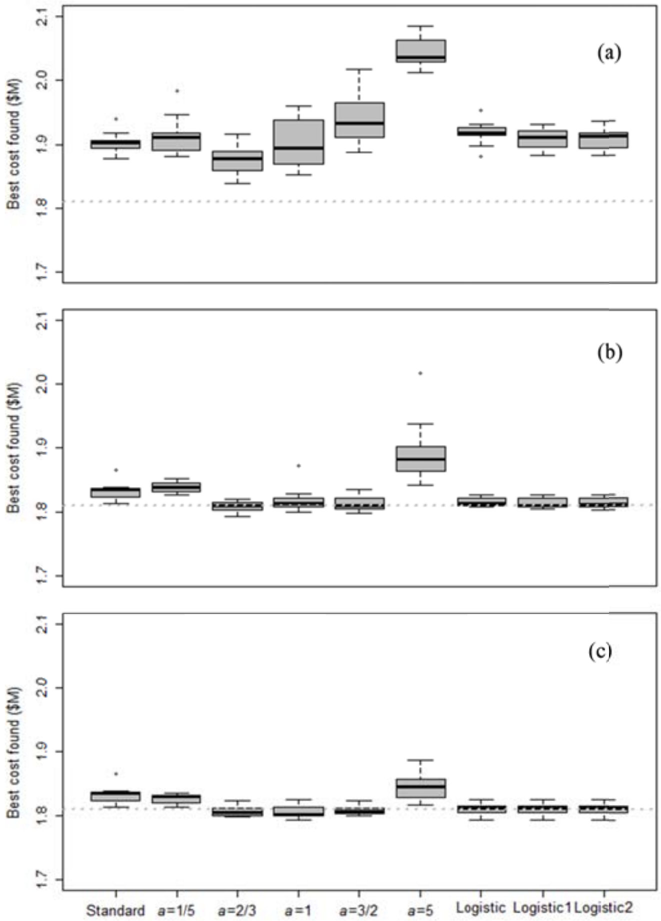

Fig 10. Comparison of algorithm performance applied to the PES for: (a) T = 400; (b) T = 1000; and (c) T = 2000. Results are based on 30 runs. Current known optimum solution is given by the dashed line.

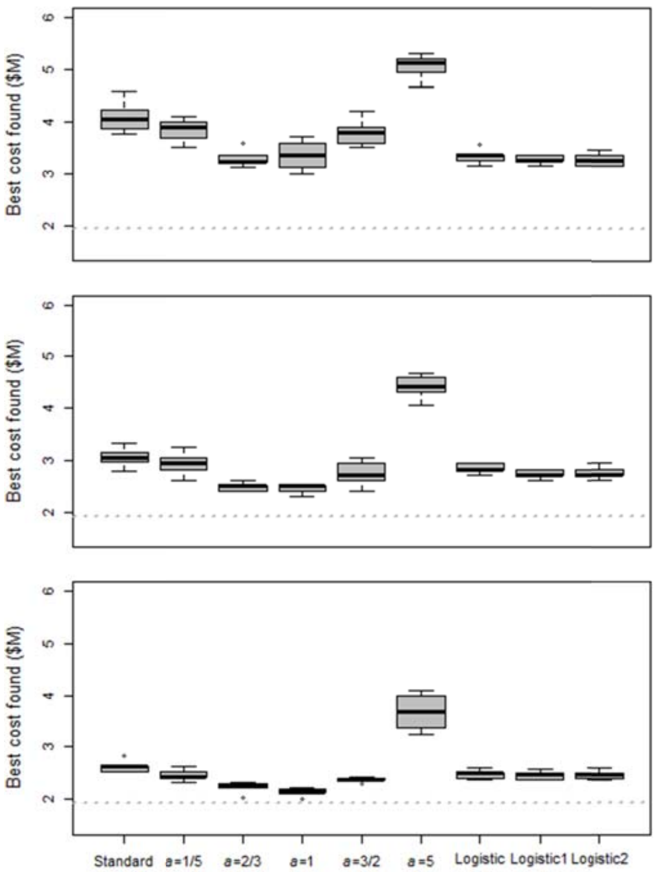

Fig 11. Comparison of algorithm performance applied to the BN for: (a) T = 1000; (b) T = 2000; and (c) T = 3000. Results are based on 10 runs. Current known optimum solution is given by the dashed line.

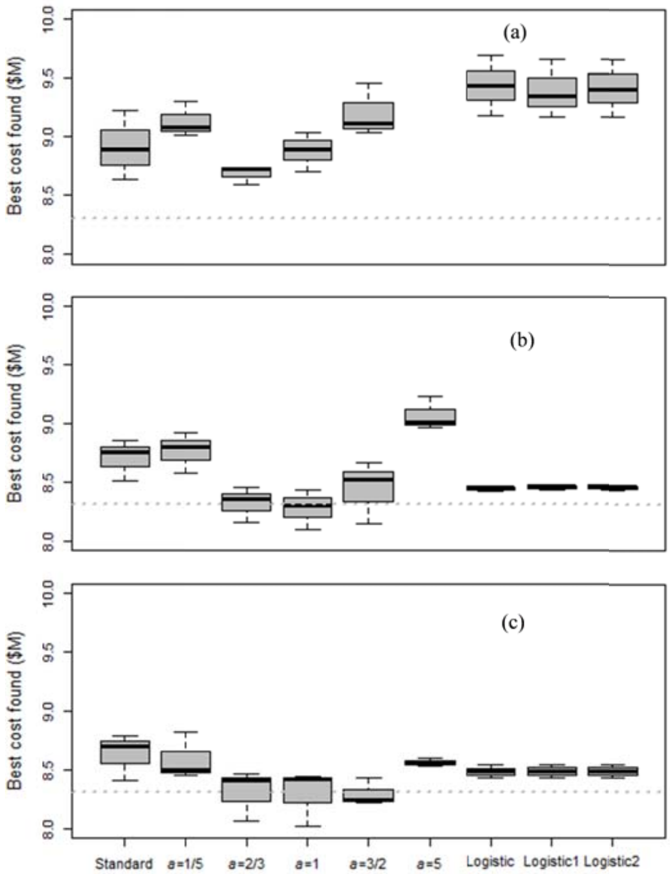

Fig 12. Comparison of algorithm performance applied to the KL for: (a) T = 1000; (b) T = 2000; and (c) T = 3000. Results are based on 10 runs. Current known optimum solution is given by the dashed line. No feasible solutions were found for the $AS_{rank+CTC}$ with a=5 when T=1000.

To further understand the performance of the proposed algorithm, Fig. 13 presents the run-time solution quality of the proposed $AS_{rank+CTC}$ and the standard ACO algorithm applied to the NYTP (smallest) and KL (largest) problems. These results were obtained by averaging the cost of solutions at the same iteration over all different runs to enable a statistically meaningful discussion. Interestingly, the patterns of improvement in solution quality over the searching time were quite similar for these two different problems. Additionally, the trajectories of solution improvement (in objective space) showed similar patterns with the convergence trajectories in decision space as given in Fig. 4. This implies that the solution quality in objective space and the convergence in decision space are strongly correlated, which agrees well with the observations in previous studies [17-19, 40]. The standard ACO algorithm and the $AS_{rank+CTC}$ with a=5 were able to find better solutions at the early searching stages compared to other algorithms due to their greater convergence rate, but at the expense of premature convergence as they did not make full use of the available computational resources. In contrast, $AS_{rank+CTC}$ with the logistic and a=1/5 trajectories exhibited slow improvement in solution quality throughout the optimization process as they focused too much on diversification during the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation

early to middle searching stages. $AS_{rank+CTC}$ with an appropriate balance between diversification and intensification (a=2/3, 1, and 3/2) showed an overall better performance in finding the optimal solutions than their counterparts, agreeing well with the boxplot results in Figs. 7 and 12. Similar observations were made for other computational scenarios and case studies.

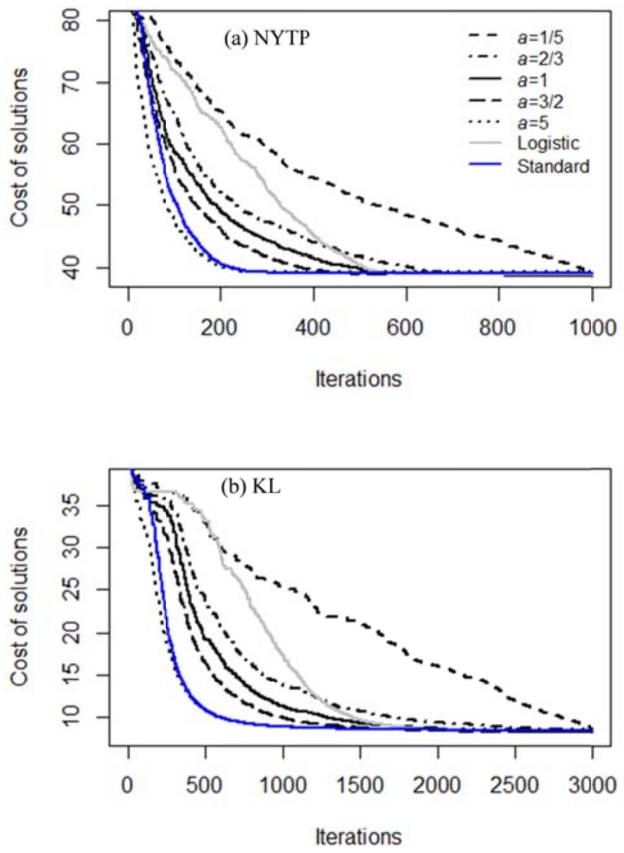

Fig 13. Run-time algorithm performance of the algorithms applied to the NYTP (a) and KL (b). Results are averages of multiple runs for each problem.

C. Summary and Discussion of Results

The implications of the results presented in Figs. 7-13 can be summarized as follows.

1) The proposed $AS_{rank+CTC}$ method enables the search to follow a specified convergence trajectory in decision space. This not only ensures convergence within a given computational budget, but also enables the management of the diversification and intensification throughout the search through the explicit specification of how broad and intensive the search is at each iteration.
2) A strong relationship exists between the convergence trajectory in decision space and the searching quality in objective space. The successful convergence trajectories (e.g., a = 2/3 in Fig. 4, which represents a balanced trade-off between diversification and intensification) consistently led to significantly improved performance compared to the standard ACO without convergence control, while poor convergence paths (e.g., a = 1/5 in Fig. 4, which strongly focused on intensification at the very early stages of the search) were observed to result in inferior searching results. In practice, the best convergence strategy could be determined based on a preliminary analysis of a range of potential paths, as in the current study. This finding has potentially significant implications for the development of future meta-heuristics, as it provides an effective alternative to improving an algorithm's performance (solution quality in objective space) through controlling its convergence trajectory in the decision space.
3) Among the eight convergence trajectories considered (Fig. 4), the a = 2/3 trajectories were found to significantly outperform the standard AS_{rank} for the six WDSDPs, irrespective of the computational budget. This demonstrates that the searching strategy with a slight bias on diversification during the first half of the search and greater emphasis on intensification within the second half of the search is effective for the WDSDPs considered. It is also noted that the evenly distributed computational efforts between diversification and intensification throughout the run (i.e. a = 1, constant convergence rates) exhibited satisfactory performance (overall statistically significant better than the standard AS_{rank}) in finding near-optimal solutions across different computation scenarios.
4) As demonstrated by the results from the three logistic trajectories with varying convergence properties in the later stages, an increase in the mean population distance during the final stages of the searching was observed to only slightly enhance the algorithm's performance compared to the case with a continuous reduction in the population diversity, especially when a relatively low computational budget was used.
5) The performance of the proposed $AS_{rank+CTC}$ method with a = 2/3 or a = 1 was very good compared with the best-found results in literature, especially for the largest computational budget used (within 5% if the best-known solution). This is despite the fact that the results in this study were (i) obtained with very limited computational budgets relative to those used in the majority of previous studies (e.g., 10 million evaluations were used for the BN problem in most previous studies [37, 39], whereas only 1.5 million evaluations were allowed in this study), and (ii) without any sensitivity analysis of the parameters, whereas the best-found solutions in literature were obtained generally with the aid of extensive parameter sensitivity analysis [20].

VI. Conclusion

Meta-heuristics have been used widely to solve various optimization problems in many different domains, particularly in the engineering design field. Their performance, often measured in terms of solution quality and computational efficiency, is heavily affected by algorithm searching behavior in the decision space of a given problem. Such behavior is typically represented by the balance between diversification and intensification, which is ultimately controlled by algorithm parameterization.

Many previous studies have been undertaken to develop parameterization strategies that enable meta-heuristics to achieve optimal trade-offs between diversification and intensification, attempting to find globally optimal solutions as quickly as possible. While these approaches have been reported to exhibit good performance for relatively simple test problems, their performance is often unsatisfactory when applied to real-world problems that generally require the use of computationally expensive simulation models for objective function and/or constraint evaluation. This is because the allowable computational budgets for real-world problems are typically insufficient to achieve final convergence when previously developed parameterization strategies are used, resulting in the identification of inferior solutions.

In recognition of the fact that computational budgets are often constrained when meta-heuristics are applied in practical applications, attempts have been made previously to develop parameterization schemes that allow meta-heuristics to identify the best possible solutions within a given computational budget. This is achieved by either pre-specifying run-time trajectories in parameter space or determining parameters to ensure an algorithm's final convergence in decision space. While both of these methods achieve certain levels of convergence control, they are unable to control an algorithm's convergence trajectory in decision space directly.

This paper has introduced a novel parameter-adaptive ant colony optimization (ACO_{CTC}) method, aiming to control ACO convergence trajectories in decision space. The fundamental premise underpinning this approach is that through controlling convergence trajectories in decision space, search quality can be improved through the balanced allocation of computational resources to searching in increasingly smaller regions in the solution space that are recognized as promising from the algorithm's learning.

The proposed ACO_{CTC} framework was used to develop a convergence controlled version of the elitist-rank ant system (termed $AS_{rank+CTC}$) based on adapting the pheromone weight α parameter. The effectiveness of the proposed $AS_{rank+CTC}$ was verified using six cases of the water distribution system problem (WDSDP), which is a hard combinatorial problem type common in water resources.

Eight different targeted convergence trajectories were considered (Fig. 4), representing cases ranging from emphasis on high diversification to high intensification. For each case study, three different computation scenarios were considered (in terms of maximum number of iterations), corresponding to the cases of non-convergence, approximate convergence and premature convergence with respect to the standard AS_{rank}.

Results from the six WDSDPs show that the proposed $AS_{rank+CTC}$ successfully enables the algorithm's search to follow the specified target convergence trajectories in decision space. It was found that the convergence trajectories can significantly affect the algorithm's final performance in terms of solution quality. The experimental results also show that, among the eight specified convergence trajectories, the trajectory with a slight emphasis on intensification (a=2/3 from Fig. 4) performed best overall and exhibited significantly improved performance compared with the standard AS_{rank}. Such superiority was observed for each case study across the three computation scenarios considered, representing robustness in performance. For the PES and KL problems with 99 and 1274 decision variables respectively, new best-known solutions were found by the proposed method, which are 1.1% and 3.4% lower (respectively) in cost than the previously reported best solutions [37], [25].

It should be noted that while the proposed approach has been proven theoretically and illustrated for six examples of a particular real-world problem type, further work is required to test the utility of the proposed approach. For example, the approach should be tested on different real-world problem types and there is a need to ascertain whether the findings in this study in relation to the relative performance of the different convergence trajectories are generally applicable or specific to different problem types (e.g., WDS design problem with multi-demand loading cases or operational optimization problems). In addition, there would be value in comparing the absolute performance of the proposed approach in terms of final solution quality with that of other adaptive algorithms.

In closing, the main conceptual innovation of this work is the development of a parameter adaption strategy to ensure that an algorithm follows a target convergence trajectory in decision space. The proposed ACO_{CTC} framework not only removes the requirement for extensive parameter calibration, but also exhibits appreciably improved performance in identifying near-optimal solutions within a limited computational budget when an appropriate target trajectory is provided. Such advantages are anticipated to enable wider up-take of the proposed method for practical applications.

REFERENCES

[1] A. Gogna, and A. Tayal, "Metaheuristics: review and application," *Journal of Experimental and Theoretical Artificial Intelligence,* vol. 25, no. 4, pp. 503-526, Jan. 2013.

[2] T. Zhenguo, and L. Yong, "A robust stochastic genetic algorithm (StGA) for global numerical optimization," *IEEE Trans. Evol. Comput.,* vol. 8, no. 5, pp. 456-470, 2004.

[3] S. Das, and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Trans. Evol. Comput.,* vol. 15, no. 1, pp. 4-31, 2011.

[4] J. Nicklow, P. Reed, D. Savic *et al.*, "State of the art for genetic algorithms and beyond in water resources planning and management," *J. Water Resources Planning Manage.*, vol. 136, no. 4, pp. 412-432, 2010.

[5] H. R. Maier, Z. Kapelan, J. Kasprzyk *et al.*, "Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions," *Environ. Model. Softw.,* vol. 62, pp. 271-299, Dec. 2014.

[6] C. Blum, and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.,* vol. 35, no. 3, pp. 268-308, 2003.

[7] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, "Parameter Control in Evolutionary Algorithms: Trends and Challenges," *IEEE Trans. Evol. Comput.,* vol. 19, no. 2, pp. 167-187, 2015.

[8] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 19–31, 2011.

[9] H. Sakanashi, K. Suzuki, and Y. Kakazui, "Controlling dynamics of GA through filtered evaluation function," in *Proc. PPSN*, 1994, pp. 239–248.

[10] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.,* vol. 13, no. 2, pp. 398-417, 2009.

[11] F. Zheng, A. Zecchin, and A. Simpson, "Self-Adaptive Differential Evolution Algorithm Applied to Water Distribution System Optimization," *J. Comput. Civil Eng.,* vol. 27, no. 2, pp. 148-158, 2013.

[12] D. Hadka, and P. Reed, "Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework," *Evolutionary Computation*, pp. 1-30, 2012.

[13] S. Favuzza, G. Graditi, and E. R. Sanseverino, "Adaptive and Dynamic Ant Colony Search Algorithm for Optimal Distribution Systems Reinforcement Strategy," *Applied Intelligence,* vol. 24, no. 1, pp. 31-42, 2006.

[14] T. Krink, P. Rickers, and R. Thomsen, "Applying self-organised criticality to evolutionary algorithms," in *Proceedings of the 6th Conference on Parallel Problem Solving From Nature* (Lecture Notes in Computer Science) vol. 1917, M. Schoenauer *et al.*, Eds. Berlin, Germany: Springer, 2000, pp. 375–384.

[15] F. Herrera and M. Lozano, "Adaptation of genetic algorithm parameters based on fuzzy logic controllers," in *Genetic Algorithms and Soft Computing*. Physica-Verlag, Heidelberg, Germany, 1996, pp. 95–125.

[16] M. S. Gibbs, G. C. Dandy, and H. R. Maier, "A genetic algorithm calibration method based on convergence due to genetic drift," *Inf. Sci.,* vol. 178, no. 14, pp. 2857-2869, 2008.

[17] A. C. Zecchin, A. R. Simpson, H. R. Maier *et al.*, "Improved understanding of the searching behavior of ant colony optimization algorithms applied to the water distribution design problem," *Water Resour. Res.,* vol. 48, no. 9, pp. W09505, 2012.

[18] F. Zheng, "Comparing the Real-Time Searching Behavior of Four Differential-Evolution Variants Applied to Water-Distribution-Network Design Optimization," *J. Water Resources Planning Manage.*, pp. 04015016, 2015, to be published.

[19] F. Zheng, A. C. Zecchin, and A. R. Simpson, "Investigating the run-time searching behavior of the differential evolution algorithm applied to water distribution system optimization," *Environ. Model. Softw.,* vol. 69, no. 0, pp. 292-307, Jul. 2015.

[20] A. Marchi, E. Salomons, A. Ostfeld *et al.*, "Battle of the Water Networks II," *Journal of Water Resources Planning and Management,* vol. 140, no. 7, pp. 04014009, 2014.

[21] Yanjun Li, and Tiejun Wu, "An adaptive ant colony system algorithm for continuous-space optimization problems," *Journal of Zhejiang University,* vol. 4, no. 1, pp. 40-46, 2004.

[22] C. W. Chiang, Y. Q. Huang, and W. Y. Wang, "Ant colony optimization with parameter adaptation for multi-mode resource-constrained project scheduling," *Journal of Intelligent & Fuzzy Systems,* vol. 19, pp. 345-358, 2008.

[23] M. H. Afshar, "A parameter free Continuous Ant Colony Optimization Algorithm for the optimal design of storm sewer networks: Constrained and unconstrained approach," *Advances in Engineering Software,* vol. 41, no. 2, pp. 188-195, 2010.

[24] S. Favuzza, G. Graditi, and E. R. Sanseverino, "Adaptive and Dynamic Ant Colony Search Algorithm for Optimal Distribution Systems Reinforcement Strategy," *Applied Intelligence,* vol. 24, no. 1, pp. 31-42, 2006.

[25] W. Bi, G. C. Dandy, and H. R. Maier, "Improved genetic algorithm optimization of water distribution system design by incorporating domain knowledge," *Environ. Model. Softw.,* vol. 69, no. 0, pp. 370-381, Jul. 2015.

[26] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[27] H. R. Maier, A. R. Simpson, A. C. Zecchin *et al.*, "Ant Colony Optimization for Design of Water Distribution Systems," *J. Water Resources Planning Manage.,* vol. 129, no. 3, pp. 200-209, 2003.

[28] A. R. Simpson, G. C. Dandy, and L. J. Murphy, "Genetic algorithms compared to other techniques for pipe optimization," *Journal of Water Resources Planning and Management,* vol. 120, no. 4, pp. 423-443, 1994.

[29] M. d. C. Cunha, and J. Sousa, "Hydraulic Infrastructures Design Using Simulated Annealing," *Journal of Infrastructure Systems,* vol. 7, no. 1, pp. 32-39, 2001

[30] M. M. Eusuff, and K. E. Lansey, "Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm," *Journal of Water Resources Planning and Management,* vol. 129, no. 3, pp. 210-225, 2003.

[31] D. A. Savic, and G. A. Walters, "Genetic algorithms for least-cost design of water distribution networks," *J. Water Resources Planning Manage.,* vol. 123, no. 2, pp. 67-77, 1997.

[32] Z. W. Geem, "Particle-swarm harmony search for water network design," *Engineering Optimization,* vol. 41, no. 4, pp. 297 - 311, 2009.

[33] K. V. K. Varma, S. Narasimhan, and S. M. Bhallamudi, "Optimal Design of Water Distribution Systems Using an NLP Method," *J. Environ. Eng.,* vol. 123, no. 4, pp. 381-388, 1997.

[34] F. Zheng, A. R. Simpson, and A. C. Zecchin, "A decomposition and multistage optimization approach applied to the optimization of water distribution systems with multiple supply sources," *Water Resour. Res.,* vol. 49, no. 1, pp. 380-399, 2013..

[35] A. C. Zecchin, A. R. Simpson, H. R. Maier *et al.*, "Parametric study for an ant algorithm applied to water distribution system optimization," *IEEE Trans. Evol. Comput.,* vol. 9, no. 2, pp. 175-191, 2005.

[36] E. Todini, and S. Pilati, "A Gradient Algorithm for the Analysis of Pipe Networks, Computer Applications in Water Supply," Coulbeck, B. and Orr, C. -H. (eds), Research Studies Press, Letchworth, Hertfordshire, UK, pp. 1-20, 1998

[37] J. Reca, and J. Martínez, "Genetic algorithms for the design of looped irrigation water distribution networks," *Water Resour. Res.,* vol. 42, no. 5, pp. W05416, 2006.

[38] Q. Wang, M. Guidolin, D. Savic *et al.*, "Two-Objective Design of Benchmark Problems of a Water Distribution System via MOEAs: Towards the Best-Known Approximation of the True Pareto Front," *J. Water Resources Planning Manage.,* vol. 141, no. 3, pp. 04014060, 2015.

[39] F. Zheng, A. R. Simpson, and A. C. Zecchin, "A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems," *Water Resour. Res.,* vol. 47, no. 8, pp. W08531, 2011.

[40] W. Bi, H. R. Maier, and G. C. Dandy, "Impact of Starting Position and Searching Mechanism on the Evolutionary Algorithm Convergence Rate," *Journal of Water Resources Planning & Management,* vol. 142, no. 9, 2016.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation

Feifei Zheng was born in Leiyang, China, in 1982. He received the B.E. degree in Environmental Engineering from Sichuan Univeristy in China, in 2005, the M.S. degree in Municipal Engineering from Tongji University in China, in 2009, and the Ph.D. degree from The University of Adelaide, Adelaide, Australia, in 2013.

He has worked as a Senior Research Associate in the School of Civil, Environmental and Mining Engineering, The University of Adelaide, Australia, from 2012 to 2015. His current position is a Professor in the College of Civil Engineering and Architecture, Zhejiang University, China. His research interests include applications of meta-heuristics to water resources problems, hydrological modelling and urban flooding risk analysis.

Aaron C. Zecchin (born in Adelaide, Australia, 1979) has received a B.Sc, B.Eng. (hons), M.M.Sc, and Ph.D from the University of Adelaide. Since 2010 he has held the positions of lecturer and senior lecturer in the School of Civil, Environmental and Mining Engineering at the University of Adelaide. His research interests include: the development, analysis and application of evolutionary algorithms to engineering design and parameter estimation problems; the development and analysis of steady- and transient-state solvers for hydraulic pipeline distribution networks; and hydraulic signal processing methods applied to transient-state pipeline systems.

Jeffrey Newman was born in South Australia in 1984. He received a Bachelor of Science with first class honors in 2008, and a Bachelor of Engineering in 2010, both at the University of Adelaide, Australia. He is currently a PhD Candidate and Research Associate at the University of Adelaide in the School of Civil, Environmental and Mining Engineering. His research interests include the application of simulation models and optimization methods for environmental and natural resources management, particularly water resource problems.

Holger R. Maier was born in Hanau/Main, Germany, in 1968. He received a Bachelor of Engineering (Civil) with first class honors from the University of Adelaide, Australia, in 1990 and a PhD from the same institution in 1996.

He worked as a Design Engineer with Kinhill Engineers, as an Environmental Engineer with the Australian Water Quality Centre, as Senior Civil Engineer and Engineering Adviser with the Samoa Water Authority and as a Postdoctoral Research Fellow at the University of British Columbia. He is currently Professor of Civil and Environmental Engineering at the University of Adelaide, Australia, as well as an Adjunct Professor in the College of Civil Engineering and Architecture, Zhejiang University, China. His research interests include the application of artificial neural techniques to modeling environmental and water resources variables, the application of evolutionary optimization methods to water resources and supply problems, and the formulation and application of risk-based performance criteria to water resources and supply problems.

Graeme Dandy was born in Geelong, Australia in 1948. He received a Bachelor of Engineering (Civil) with first class honors in 1971 and a Master of Engineering Science in 1973 from the University of Melbourne and a PhD from the Massachusetts Institute of Technology in 1976. He is currently an Emeritus Professor of Civil and Environmental Engineering at the University of Adelaide and a Director of the software company Optimatics Pty Ltd. He is a Fellow of the Australian Academy of Technological Sciences and Engineering and a Fellow of Engineers Australia. His research interests including the application of evolutionary optimization techniques to water resources problems and the application of artificial neural networks for forecasting water resources and environmental variables.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2682899, IEEE Transactions on Evolutionary Computation