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.

Ant Colony Optimization for Mixed-Variable Optimization Problems

Tianjun Liao, Krzysztof Socha, Marco A. Montes de Oca, Thomas Stützle *Senior Member, IEEE*, and Marco Dorigo, *Fellow, IEEE*

***Abstract*—In this paper, we introduce ACO$_{\mathrm{MV}}$, an ant colony optimization (ACO) algorithm that extends the ACO$_{\mathbb{R}}$ algorithm for continuous optimization to tackle mixed-variable optimization problems. In ACO$_{\mathrm{MV}}$, the decision variables of an optimization problem can be explicitly declared as continuous, ordinal, or categorical, which allows the algorithm to treat them adequately. ACO$_{\mathrm{MV}}$ includes three solution generation mechanisms: a continuous optimization mechanism (ACO$_{\mathbb{R}}$), a continuous relaxation mechanism (ACO$_{\mathrm{MV}}$-o) for ordinal variables, and a categorical optimization mechanism (ACO$_{\mathrm{MV}}$-c) for categorical variables. Together, these mechanisms allow ACO$_{\mathrm{MV}}$ to tackle mixed-variable optimization problems.**

We also define a novel procedure to generate artificial, mixed-variable benchmark functions and we use it to automatically tune ACO$_{\mathrm{MV}}$'s parameters. The tuned ACO$_{\mathrm{MV}}$ is tested on various real-world continuous and mixed-variable engineering optimization problems. Comparisons with results from the literature demonstrate the effectiveness and robustness of ACO$_{\mathrm{MV}}$ on mixed-variable optimization problems.

***Index Terms*—Ant colony optimization, mixed-variable optimization problems, artificial mixed-variable benchmark functions, automatic parameter tuning, engineering optimization**

I. Introduction

Many real-world optimization problems can be modeled using combinations of continuous and discrete variables. Due to the practical relevance of these *mixed-variable problems*, a number of optimization algorithms for tackling them have been proposed. These algorithms are mainly based on Genetic Algorithms [1], Differential Evolution [2], Particle Swarm Optimization [3] and Pattern Search [4]. The discrete variables in these problems can be ordinal or categorical. Ordinal variables exhibit a natural ordering relation (e.g., integers) and are usually handled using a *continuous relaxation approach* [5], [6], [7], [8], [9], [10], [11], [12]. Categorical variables take their values from a finite set of categories [13], which often identify non-numeric elements of an unordered set (e.g., colors, shapes or types of material). Categorical variables do not have a natural ordering relation and therefore require the use of a *categorical optimization approach* [14], [15], [16], [17], [18], [19], [13] that does not assume any ordering relation. To the best of our knowledge, the approaches to mixed-variable problems available in the literature are targeted to either handle mixtures of continuous and ordinal variables or mixtures of continuous and categorical variables. In other words, they do not consider the possibility that the formulation of a problem may involve at the same time the three types of variables. Hence, there is a need for algorithms that allow the explicit declaration of each variable as either continuous, ordinal or categorical.

In this paper, we extend an ant colony optimization algorithm for continuous optimization (called ACO$_{\mathbb{R}}$) [20] to tackle mixed-variable optimization problems. Ant colony optimization (ACO) was originally introduced to solve discrete optimization problems [21], [22], [23] and its adaptation to solve continuous or integer optimization problems enjoys an increasing attention [20], [24], [25], [26], [27], [28], [29]. Our ACO algorithm, called ACO$_{\mathrm{MV}}$, allows the user to explicitly declare each variable of a mixed-variable optimization problem as continuous, ordinal or categorical. Continuous variables are handled with a continuous optimization approach (ACO$_{\mathbb{R}}$), ordinal variables are handled with a continuous relaxation approach (ACO$_{\mathrm{MV}}$-o), and categorical variables are handled with a categorical optimization approach (ACO$_{\mathrm{MV}}$-c).

We also introduce a new set of artificial, mixed-variable benchmark functions and describe the method to construct them. These benchmark functions provide a flexible environment for investigating the performance of mixed-variable optimization algorithms and the effect of different parameter settings on their performance. They are also useful as a training set for deriving high-performance parameter settings through the usage of automatic configuration methods. Here, we use Iterated F-Race [30], [31] to automatically tune the parameters of ACO$_{\mathrm{MV}}$ on a set of artificial, mixed-variable benchmark functions.

As a final step, we compare the performance of ACO$_{\mathrm{MV}}$ with results from the literature on eight mixed-variable engineering optimization problems. Our results show that ACO$_{\mathrm{MV}}$ reaches a very high performance: it improves over the best known solutions for two of the eight engineering problems, and in the remaining six it finds the best-known solutions using fewer objective function evaluations than most algorithms from the literature.

The paper is organized as follows. Section II introduces mixed-variable optimization problems and Section III describes ACO$_{\mathrm{MV}}$. Section IV presents the proposed artificial mixed-variable benchmark functions and the tuning of the parameters of ACO$_{\mathrm{MV}}$ on these benchmark functions. In Section V, we compare the results obtained with ACO$_{\mathrm{MV}}$ on

T. Liao, T. Stützle and M. Dorigo are with Institut de Recherches Interdisciplinaires et de Développements en Intelligence Atificielle (IRIDIA), Université libre de Bruxelles, 1050 Brussels, Belgium. (e-mail: tliao@ulb.ac.be; stuetzle@ulb.ac.be; mdorigo@ulb.ac.be).

M.A. Montes de Oca is with Department of Mathematical Sciences, University of Delaware, Newark, DE 19716 USA. (e-mail: mmontes@math.udel.edu).

real-world problems to those obtained by other algorithms. In Section VI we conclude and give directions for future work. The Appendix contains further experimental results and a mathematical formulation of the engineering benchmark problems we tackle.

II. Mixed-variable optimization problems

A model for a mixed-variable optimization problem (MVOP) may be formally defined as follows:

Definition *A model $R = (\mathbf{S}, \mathbf{\Omega}, f)$ of a MVOP consists of*

- *a search space $\mathbf{S}$ defined over a finite set of both discrete and continuous decision variables and a set $\mathbf{\Omega}$ of constraints among the variables;*
- *an objective function $f : \mathbf{S} \rightarrow \mathbb{R}_0^+$ to be minimized.*

The search space $\mathbf{S}$ is defined by a set of $n = d + r$ variables $x_i, i = 1, \ldots, n$, of which d are discrete and r are continuous. The discrete variables include o ordinal variables and c categorical ones, $d = o + c$. A solution $S \in \mathbf{S}$ is a complete value assignment, that is, each decision variable is assigned a value. A feasible solution is a solution that satisfies all constraints in the set $\mathbf{\Omega}$. A global optimum $S^* \in \mathbf{S}$ is a feasible solution that satisfies $f(S^*) \leq f(S)\ \forall S \in \mathbf{S}$. The set of all globally optimal solutions is denoted by $\mathbf{S}^*, \mathbf{S}^* \subseteq \mathbf{S}$. Solving an MVOP requires finding at least one $S^* \in \mathbf{S}^*$.

The methods proposed in the literature to tackle MVOPs may be divided into three groups:

- The first group is based on a *two-partition approach*, in which the variables are partitioned into continuous variables and discrete variables. Variables of one partition are optimized separately for fixed values of the variables of the other partition [32], [33]. This approach often leads to a large number of objective function evaluations [34]. Additionally, since the dependency between variables belonging to different partitions is not explicitly handled, algorithms using this approach are prone to finding sub-optimal solutions.
- The second group takes a *continuous relaxation approach*. In this group, all variables are handled as continuous variables. Ordinal variables are relaxed to continuous variables, and are repaired when evaluating the objective function. The repair mechanism is used to return a discrete value in each iteration. The simplest repair mechanisms are truncation and rounding [5], [8]. It is also possible to treat categorical variables using continuous relaxations [35]. However, in this case the performance of continuous relaxation may decline when the number of categories increases, as we also show in Section A of the Appendix to this paper. In general, the performance of algorithms based on the continuous relaxation approach depends on the continuous solvers and on the repair mechanism.
- The third group uses a *categorical optimization approach* to directly handle discrete variables without a continuous relaxation. Thus, any possible ordering relations that may exist between discrete variables are ignored and, thus, all discrete variables, ordinal and categorical, are treated as categorical ones.[1] In this group, continuous variables are handled by a continuous optimization method. Genetic adaptive search [14], pattern search [15], and mixed Bayesian optimization [17] are among the approaches that have been proposed.

Researchers often take one specific group of approaches to develop mixed-variable optimization algorithms and to test them on MVOPs with either categorical or ordinal variables. In our study, we combine a continuous relaxation and a categorical optimization approach.

III. ACO$_{\mathbf{MV}}$ for Mixed-Variable Optimization Problems

We start by describing the structure of ACO$_{\mathbf{MV}}$. Then, we describe the probabilistic solution construction for continuous variables, ordinal variables and categorical variables, respectively.

A. *ACO$_{\mathbf{MV}}$ structure*

ACO algorithms for combinatorial optimization problems make use of a so-called pheromone model in order to probabilistically construct solutions. A pheromone model consists of a set of numerical values, called pheromones, that are a function of the search experience of the algorithm. The pheromone model is used to bias the solution construction towards regions of the search space containing high quality solutions. As such, ACO algorithms follow a model-based search paradigm [36] as, for example, also estimation of distribution algorithms [37] do; the similarities and differences between ACO algorithms and estimation of distribution algorithms have been discussed by Zlochin et al. [36]. In ACO for combinatorial optimization problems, the pheromone values are associated with a finite set of discrete components. This is not possible if continuous variables are involved. Therefore, ACO$_{\mathbf{MV}}$ uses a *solution archive, SA,* as a form of pheromone model for the derivation of a probability distribution over the search space, following in this way the principle of population-based ACO [38]. The solution archive contains k complete solutions of the problem. While a pheromone model in combinatorial optimization can be seen as an implicit memory of the search history, a solution archive is an explicit memory.

Given an n-dimensional MVOP and k solutions, ACO$_{\mathbf{MV}}$ stores the value of the n variables and the objective function value of each solution in the solution archive. Fig. 1 shows the structure of the solution archive. It is divided into three groups of columns, one for continuous variables, one for ordinal variables and one for categorical variables.

The basic flow of the ACO$_{\mathbf{MV}}$ algorithm is as follows. The solution archive is initialized with k randomly generated solutions. Then, these k solutions are sorted according to their quality (from best to worst). A weight ω_j is associated with solution S_j. This weight is calculated using a Gaussian function defined by:

[1]Note that the special case of MVOPs, where the variables can be either continuous or categorical, is also called mixed-variable programming problem [15], [18].

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.

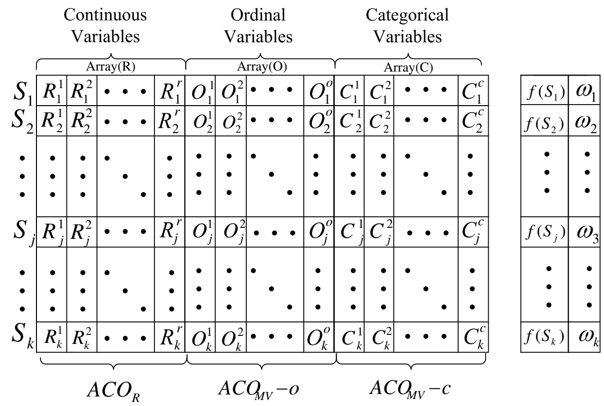


Fig. 1. The structure of the solution archive used by $\text{ACO}_{\mathbf{MV}}$. The solutions in the archive are sorted according to their quality (i.e., the value of the objective function $f(S_j)$); hence, the position of a solution in the archive always corresponds to its rank.

$$\omega_j = \frac{1}{qk\sqrt{2\pi}} e^{\frac{-(rank(j)-1)^2}{2q^2k^2}}, \tag{1}$$

where $rank(j)$ is a function that returns the rank of solution S_j, and q is a parameter of the algorithm. By computing $rank(j) - 1$, which corresponds to setting the mean of the Gaussian function to 1, the best solution receives the highest weight, while the weight of the other solutions decreases exponentially with their rank. At each iteration of the algorithm, m new solutions are probabilistically constructed by m ants, where an ant is a probabilistic solution construction procedure. The weight of a solution determines the level of attractiveness of that solution during the solution construction process. A higher weight means a higher probability of sampling around that solution. Once the m solutions have been generated, they are added into the solution archive. The $k+m$ solutions in the archive are then sorted and the m worst ones are removed. The remaining k solutions constitute the new solution archive. In this way, the search process is biased towards the best solutions found during the search. During the probabilistic solution construction process, an ant applies the construction mechanisms of $\text{ACO}_{\mathbb{R}}$, $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c. $\text{ACO}_{\mathbb{R}}$ handles continuous variables, while $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c handle ordinal variables and categorical variables, respectively. Their detailed description is given in the following subsection. An outline of the $\text{ACO}_{\mathbf{MV}}$ algorithm is given in Algorithm 1. The functions Best and Sort in Algorithm 1 implement the sorting of the archive and the selection of the k best solutions.

B. Probabilistic Solution Construction for Continuous Variables

Continuous variables are handled by $\text{ACO}_{\mathbb{R}}$ [20]. In $\text{ACO}_{\mathbb{R}}$, the construction of new solutions by the ants is accomplished in an incremental manner, variable by variable. First, an ant chooses probabilistically one of the solutions in the archive. The probability of choosing solution j is given by:

$$p_j = \frac{\omega_j}{\sum_{l=1}^{k} \omega_l}, \tag{2}$$

Algorithm 1 Outline of $\text{ACO}_{\mathbf{MV}}$

```
Initialize decision variables
Initialize and evaluate k solutions
{Sort solutions and store them in the archive SA}
SA ← Sort(S_1 ··· S_k)
while termination criterion is not satisfied do
  {ConstructAntSolution}
  for 1 to m do
    Probabilistic Solution Construction for ACO_R
    Probabilistic Solution Construction for ACO_MV-o
    Probabilistic Solution Construction for ACO_MV-c
    Store and evaluate newly generated solutions
  end for
  {Sort solutions and select the best k solutions}
  SA ← Best(Sort(S_1 ··· S_{k+m}), k)
end while
```

where ω_j is calculated according to Equation (1).

An ant then constructs a new continuous variable solution around the chosen solution j. It assigns values to variables in a fixed variable order, that is, at the i-th construction step, $1 \le i \le r$, an ant assigns a value to continuous variable i. To assign a value to variable i, the ant samples the neighborhood around the value R_j^i of the chosen j-th solution. The sampling is done using a normal probability density function with mean μ and standard deviation σ:

$$g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{3}$$

When considering continuous variable i of solution j, we set $\mu = R_j^i$. Furthermore, we set

$$\sigma = \xi \sum_{l=1}^{k} \frac{|R_l^i - R_j^i|}{k-1}, \tag{4}$$

which is the average distance between the values of the i-th continuous variable of the solution j and the values of the i-th continuous variables of the other solutions in the archive, multiplied by a parameter ξ. This parameter has an effect similar to that of the pheromone persistence in ACO. The higher the value of ξ, the lower the convergence speed of the algorithm. This process is repeated for each dimension by each of the m ants.

Thanks to the pheromone representation used in $\text{ACO}_{\mathbb{R}}$ (that is, the solution archive), it is possible to take into account the correlation between the decision variables. A non-deterministic adaptive method for doing so is presented in [20]. It is effective on the rotated benchmark functions proposed in Table I and it is also used to handle the variable dependencies of MVOP engineering problems in Section V.

C. Probabilistic Solution Construction for Ordinal Variables

If the considered optimization problem includes ordinal variables, the continuous relaxation approach, $\text{ACO}_{\mathbf{MV}}$-o, is used. $\text{ACO}_{\mathbf{MV}}$-o does not operate on the actual values of the ordinal variables but on their indices in an array. The values of the indices for the new solutions are generated as real numbers, as it is the case for the continuous variables. However, before the objective function is evaluated, the continuous values are rounded to the nearest valid index, and the value at that index is then used for the objective function evaluation. The reason for this choice is that ordinal variables do not necessarily have numerical values; for example, an ordered variable may take as possible values $\{\text{small}, \text{medium}, \text{large}\}$. $\text{ACO}_{\mathbf{MV}}$-o otherwise works exactly as $\text{ACO}_{\mathbb{R}}$.

D. Probabilistic Solution Construction for Categorical Variables

While ordinal variables are relaxed and treated by the original $\text{ACO}_{\mathbb{R}}$, categorical variables are treated differently by $\text{ACO}_{\mathbf{MV}}$-c as this type of variables has no predefined ordering. At each step of $\text{ACO}_{\mathbf{MV}}$-c, an ant assigns a value to one variable at a time. For each categorical variable i, $1 \leq i \leq c$, an ant chooses probabilistically one of the t_i available values $v_l^i \in \{v_1^i, \ldots, v_{t_i}^i\}$. The probability of choosing the l-th value is given by

$$p_l^i = \frac{w_l}{\sum_{j=1}^{t_i} w_j}, \tag{5}$$

where w_l is the weight associated to the l-th available value. The weight w_l is calculated as

$$w_l = \begin{cases} \dfrac{\omega_{j_l}}{u_l^i} + \dfrac{q}{\eta}, & \textit{if}(\eta > 0, u_l^i > 0), \\ \dfrac{\omega_{j_l}}{u_l^i}, & \textit{if}(\eta = 0, u_l^i > 0), \\ \dfrac{q}{\eta}, & \textit{if}(\eta > 0, u_l^i = 0), \end{cases} \tag{6}$$

where ω_{j_l} is calculated according to Equation (1) with j_l being the index of the highest quality solution that uses value v_l^i for the categorical variable i. u_l^i is the number of solutions that use value v_l^i for the categorical variable i in the archive (hence, the more common the value v_l^i is, the lower is its final weight); thus, u_l^i is a variable whose value is adapted at run-time and that controls the weight of choosing the l-th available value. $u_l^i = 0$ corresponds to the case in which the l-th available value is not used by the solutions in the archive; in this case the weight of the l-th value is equal to $\frac{q}{\eta}$. η is the number of values from the t_i available ones that are not used by the solutions in the archive; $\eta = 0$ (that is, all values are used) corresponds to the case in which $\frac{q}{\eta}$ is discarded. Again, η is a variable that is adapted at run-time and, if $\eta = 0$, it is natural to discard the second component in Equation (6). Note that u_l^i and η are nonnegative numbers, and their values are never equal to zero at the same time. q is the same parameter of the algorithm that was used in Equation (1).

The weight w_l is therefore a sum of two components. The first component biases the choice towards values that are chosen in the best solutions but do not occur very frequently among all solutions in the archive. The second component plays the role of exploring values of the categorical decision variable i that are currently not used by any solution in the archive; in fact, the weight of such values according to the first component would be zero and, thus, this mechanism helps to avoid premature convergence (in other words, to increase diversification).

In Appendix D, we experimentally explore different options for the shape of Equation (6); the details of the experimental setup used in Appendix D is explained in Section IV, which should therefore be consulted before reading the appendix.

E. Restart strategy

$\text{ACO}_{\mathbf{MV}}$ uses a simple restart strategy for fighting stagnation. This strategy consists in restarting the algorithm without forgetting the best-so-far solution in the archive. A restart is triggered if the number of consecutive iterations with a relative solution improvement lower than a certain threshold ε is larger than *MaxStagIter*. Since this is a component that can be used with any algorithm and not only with $\text{ACO}_{\mathbf{MV}}$, we compare the performance of $\text{ACO}_{\mathbf{MV}}$ with and without this restart mechanism to that of other algorithm.

IV. Artificial mixed-variable benchmark functions and parameter tuning of $\text{ACO}_{\mathbf{MV}}$

A. Artificial mixed-variable benchmark functions

The real world mixed-variable benchmark problems found in the literature often originate from the mechanical engineering field. Unfortunately, these problems cannot be easily parametrized and flexibly manipulated for investigating the performance of mixed-variable optimization algorithms in a systematic way. In this section, we propose a set of new, artificial mixed-variable benchmark functions that allow the definition of a controlled environment for the investigation of algorithm performance and automatic tuning of algorithm parameters [31], [39]. Our proposed artificial mixed-variable benchmark functions are defined in Table I. These functions originate from some typical continuous functions of the CEC'05 benchmark set [40]. The decision variables consist of continuous and discrete variables; n is the total number of variables and $\mathbf{M}$ is a random, normalized, $n \times n$ rotation matrix. The problems' global optima $\vec{S^*}$ are shifted in order not to give an advantage to population-based methods that may have a bias towards the origin of the search space [41]. The proposed benchmarks allow three settings for discrete variables. The first setting consists of only ordinal variables; the second setting consists of only categorical variables; the third setting consists of both ordinal and categorical variables. *MinRange* and *MaxRange* denote the lower and upper bound of variable domains, respectively.

We use the two-dimensional, not shifted, randomly rotated Ellipsoid mixed-variable function as an example of how to construct artificial mixed-variable benchmark functions. We start with a two-dimensional, continuous, not shifted, randomly rotated Ellipsoid function:

$$f_{EL}(\vec{x}) = \sum_{i=1}^{2} (\beta^{\frac{i-1}{2-1}} z_i)^2, \quad \begin{cases} x_1, x_2 \in [-3, 7], \\ \vec{z} = \mathbf{M}\vec{x}, \\ \beta = 5. \end{cases} \tag{7}$$

In order to transform this continuous function into a mixed-variable one, we discretize the continuous domain of variable $x_1 \in [-3, 7]$ into a set of discrete values, $\mathbf{T} = \{\theta_1, \theta_2, ..., \theta_t\}$:

TABLE I
ARTIFICIAL MIXED-VARIABLE BENCHMARK FUNCTIONS. IN THE UPPER PART THE OBJECTIVE FUNCTIONS ARE DEFINED; THE VARIABLES ARE DEFINED IN THE LOWER PART OF THE TABLE.

Objective functions	
$f_{Ellipsoid_{MV}}(\vec{x}) = \sum_{i=1}^{n}(\beta^{\frac{i-1}{n-1}} z_i)^2,$	
$f_{Ackley_{MV}}(\vec{x}) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i^2)}} - e^{\frac{1}{n}\sum_{i=1}^{n}(\cos(2\pi z_i))} + 20 + e,$	
$f_{Rastrigin_{MV}}(\vec{x}) = 10n + \sum_{i=1}^{n}(z_i^2 - 10\cos(2\pi z_i^2)),$	
$f_{Rosenbrock_{MV}}(\vec{x}) = \sum_{i=1}^{n-1}[100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2],$	
$f_{Sphere_{MV}}(\vec{x}) = \sum_{i=1}^{n} z_i^2,$	
$f_{Griewank_{MV}}(\vec{x}) = \frac{1}{4\,000}\sum_{i=1}^{n} z_i^2 - \prod_{i=1}^{n}\cos(\frac{z_i}{\sqrt{i}}) + 1,$	
Definition of mixed variables	
1st setting:	$\begin{cases} \vec{z} = \mathbf{M}(\vec{x} - \vec{S^*}) : \vec{S^*} = (R_*^1\, R_*^2 \dots R_*^r\, O_*^1\, O_*^2 \dots O_*^o)^t, & \\ \mathit{if}(f_{Rosenbrock_{MV}}), \vec{z} = \vec{z} + 1, & \\ \vec{S^*} \text{ is a shift vector, } n = o + r, & \\ \vec{x} = (R^1\, R^2 \dots R^r\, O^1\, O^2 \dots O^o)^t, & \\ R^i \in (\mathit{MinRange}^i, \mathit{MaxRange}^i), & i = 1, \dots, r \\ O^i \in \mathbf{T}, \mathbf{T} = \{\theta_1, \theta_2, ..., \theta_{t_i}\} : \forall l\ \theta_{t_l} \in (\mathit{MinRange}^i, \mathit{MaxRange}^i) & i = 1, \dots, o \end{cases}$
2nd setting:	$\begin{cases} \vec{z} = \mathbf{M}(\vec{x} - \vec{S^*}) : \vec{S^*} = (R_*^1\, R_*^2 \dots R_*^r\, C_*^1\, C_*^2 \dots C_*^c,)^t, & \\ \mathit{if}(f_{Rosenbrock_{MV}}), \vec{z} = \vec{z} + 1, & \\ \vec{S^*} \text{ is a shift vector, } n = c + r, & \\ \vec{x} = (R^1\, R^2 \dots R^r\, C^1\, C^2 \dots C^c)^t, & \\ R^i \in (\mathit{MinRange}^i, \mathit{MaxRange}^i), & i = 1, \dots, r \\ C^i \in \mathbf{T}, \mathbf{T} = \{\theta_1, \theta_2, ..., \theta_{t_i}\} : \forall l\ \theta_{t_l} \in (\mathit{MinRange}^i, \mathit{MaxRange}^i) & i = 1, \dots, c \end{cases}$
3rd setting:	$\begin{cases} \vec{z} = \mathbf{M}(\vec{x} - \vec{S^*}) : \vec{S^*} = (R_*^1\, R_*^2 \dots R_*^r\, O_*^1\, O_*^2 \dots O_*^o\, C_*^1\, C_*^2 \dots C_*^c)^t, & \\ \mathit{if}(f_{Rosenbrock_{MV}}), \vec{z} = \vec{z} + 1, & \\ \vec{S^*} \text{ is a shift vector, } n = o + c + r, & \\ \vec{x} = (R^1\, R^2 \dots R^r\, O^1\, O^2 \dots O^o\, C^1\, C^2 \dots C^c)^t, & \\ R^i \in (\mathit{MinRange}^i, \mathit{MaxRange}^i), & i = 1, \dots, r \\ O^i \in \mathbf{T}, \mathbf{T} = \{\theta_1, \theta_2, ..., \theta_{t_i}\} : \forall l\ \theta_{t_l} \in (\mathit{MinRange}^i, \mathit{MaxRange}^i) & i = 1, \dots, o \\ C^i \in \mathbf{T}, \mathbf{T} = \{\theta_1, \theta_2, ..., \theta_{t_i}\} : \forall l\ \theta_{t_l} \in (\mathit{MinRange}^i, \mathit{MaxRange}^i) & i = 1, \dots, c \end{cases}$

$\theta_i \in [-3, 7]$. This results in the following mixed-variable test function:

$$f_{EL_{MV}}(x_1, x_2) = z_1^2 + \beta \cdot z_2^2, \quad \begin{cases} x_1 \in \mathbf{T}, \\ x_2 \in [-3, 7], \\ \vec{z} = \mathbf{M}\vec{x}, \\ \beta = 5. \end{cases} \tag{8}$$

The set $\mathbf{T}$ is created by choosing t uniformly spaced values from the original domain $[-3, 7]$ so that $\exists_{i=1,\dots,t}\ \theta_i = 0$. In this way, it is always possible to find the optimum value $f_{EL_{MV}}(0, 0)^t = 0$, regardless of the chosen t discrete values.

Problems that involve ordinal variables are easy to simulate with the aforementioned procedure because the discrete points in the discretization for variable x_1 are naturally ordered. The left plot in Fig. 2 shows how the algorithm "sees" such a naturally ordered rotated ellipsoid function, with variable x_1 being the discrete variable. The test function is presented as a set of points representing different solutions. To simulate problems involving categorical variables only, the discrete points are ordered randomly. In this setting, a different ordering is generated for each run of the algorithm. This setting allows us to investigate how the algorithm performs when the ordering of the discrete points is not well defined or unknown. The right plot of Fig. 2 shows how the algorithm "sees" such a modified problem for a given single random ordering.

The artificial mixed-variable benchmark functions have characteristics such as non-separability, ill-conditioning and multi-modality. Non-separable functions often exhibit complex dependencies between decision variables. Ill-conditioned functions often lead to premature convergence. Multi-modal functions have multiple local optima and require an efficient global search. Therefore, these characteristics are expected to be a challenge for different mixed-variable optimization algorithms. The flexibility in defining functions with different numbers of discrete points and the possible mixing of ordered and categorical variables enables systematic experimental studies addressing the impact of function features on algorithm performance. In fact, using these benchmark functions we verified that $\mathrm{ACO_{MV}}$-o is more effective than $\mathrm{ACO_{MV}}$-c on problems that have ordinal variables while the opposite is true on problems with categorical variables. A detailed experimental analysis that corroborates this statement is given in Section A

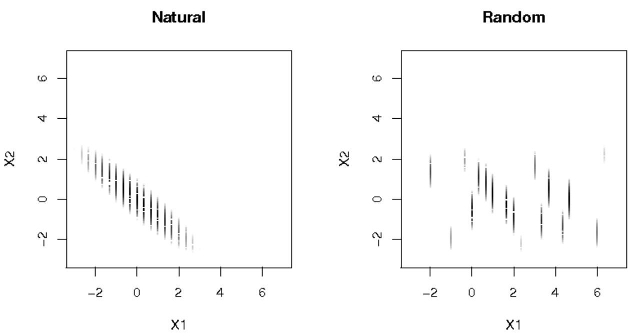


Fig. 2. Randomly rotated ellipsoid function ($\beta = 5$) with discrete variable $x_1 \in \mathbf{T}$. The left plot presents the case in which the natural ordering of the intervals is used, while the right one presents the case in which a random ordering is used. The darker the point, the higher the quality of the solution.

of the Appendix. This result also validates our design choice for combining these two approaches in $\text{ACO}_{\mathbf{MV}}$.

B. *Parameter tuning of ACO*$_{\mathbf{MV}}$

Besides serving for experimenal studies, the new benchmark functions can be used to generate a training set of problems for the automatic parameter tuning of mixed-variable optimization algorithms. The tuning of an algorithm on a training set that is different from the test set is important to allow for an unbiased assessment of the algorithm's performance on (by the algorithm unseen) test problems [42]. We therefore generate a training set of benchmark functions across all six mixed-variable benchmark functions, across various dimensions [43] (taken from the set $n \in$ {2, 4, 6, 8, 10, 12, 14}), and across various ratios of ordinal and categorical variables. As tuning method we use Iterated F-Race [30], [31]. In Iterated F-Race, the training benchmark functions are sampled in a random order. The performance measure for tuning is the objective function value of each instance after 10 000 function evaluations. The maximum tuning budget for Iterated F-Race is set to 5 000 runs of $\text{ACO}_{\mathbf{MV}}$. We use the default settings of Iterated F-Race [31].

The obtained parameter settings after tuning are given in Table II. We first use these parameter settings (i) to analyze the effectiveness of $\text{ACO}_{\mathbf{MV}}$'s restart mechanism and (ii) to obtain numerical results of $\text{ACO}_{\mathbf{MV}}$ on artificial mixed-variable benchmark problems, which can serve as a benchmark for future developments of algorithms for mixed-variable optimization problems. The corresponding results are given in Sections B and C of the Appendix, respectively. Finally, as mentioned before, we also analyzed the influence alternative choices for Equation (6) would have on the performance of $\text{ACO}_{\mathbf{MV}}$. In particular, we study three alternative choices and we report the results in Section D of the Appendix. These experimental results confirm the advantage of our original choice of Equation (6).

Next, we use these parameter settings for a final validation of $\text{ACO}_{\mathbf{MV}}$'s performance, namely for solving real world engineering optimization problems; these results are reported in the next section.

TABLE II
PARAMETER SETTINGS FOR $\text{ACO}_{\mathbf{MV}}$ TUNED BY ITERATED F-RACE.

Parameter	Symbol	Value
Number of ants	m	5
Influence of best quality solutions	q	0.05099
Width of the search	ξ	0.6795
Archive size	k	90
Stagnating iterations before restart	*MaxStagIter*	650
Relative improvement threshold	ε	10^{-5}

V. APPLICATION TO ENGINEERING OPTIMIZATION PROBLEMS

Here, we conduct experiments on mixed-variable engineering benchmark problems and compare the results of $\text{ACO}_{\mathbf{MV}}$ with those found in the literature. Since the algorithms presented in the literature do not use restarts, we additionally present computational results of a variant $\text{ACO}_{\mathbf{MV}}^{noR}$, where we switched off the restart in $\text{ACO}_{\mathbf{MV}}$. This was done to examine whether possible advantages of $\text{ACO}_{\mathbf{MV}}$ over other algorithms may be due to this particular algorithm feature. For reducing the variability of the results, we used the method of common random numbers as a variance reduction technique, so that if a problem is actually solved without restart, the reported results for $\text{ACO}_{\mathbf{MV}}^{noR}$ and $\text{ACO}_{\mathbf{MV}}$ are identical. In fact, our experimental results show that only on three of the eight problems tested the algorithm restarts actually contribute to improved performance; we will highlight these cases in the text.

Note that our experiments comprise a larger set of benchmark problems than in the papers found in the literature, since these latter are often limited to a specific type of discrete variables (either ordinal or categorical). First, we classify the available engineering optimization problems in the literature into four groups according to the types of the decision variables used (see Table III).

TABLE III
THE CLASSIFICATION OF ENGINEERING OPTIMIZATION PROBLEMS.

Groups	The type of decision variables
Group I	Continuous variables†
Group II	Continuous and ordinal variables
Group III	Continuous and categorical variables
Group IV	Continuous, ordinal and categorical variables

† Problems with only continuous variables are considered as a particular class of mixed variables with an empty set of discrete variables, since $\text{ACO}_{\mathbf{MV}}$ is also capable to solve pure continuous optimization problems.

Group I includes the welded beam design problem case A [44]; Group II the pressure vessel design problem [45] and the coil spring design problem [45]; Group III the thermal insulation systems design problem [16]; and Group IV the welded beam design problem case B [46]. The mathematical formulations of the problems are given in Section E of the Appendix. In this section, we compare the results obtained by $\text{ACO}_{\mathbf{MV}}$ to those reported in the literature for these problems. We also show the run-time behavior of $\text{ACO}_{\mathbf{MV}}$ by using run-length distributions (RLDs, for short) [47]. An (empirical) RLD provides a graphical view of the development of the empirical frequency of finding a solution of a certain quality as a function of the number of objective function evaluations. It is important to note that NM-PSO [48] and PSOLVER [49] report infeasible solutions that violate the problems' constraints; Črepinšek et al. [50] pointed out that the authors of TLBO [51] used an incorrect formula for computing the number of objective function evaluations. Therefore, we did not include these three algorithms in our comparison. For our experiments, the tuned parameter configuration from Table II was used. For simplifying the algorithm and giving prominence to the role of the $\text{ACO}_{\mathbf{MV}}$ heuristic itself, the most fundamental constraint handling technique was used, which consists in rejecting all infeasible solutions in the optimization process (also called "death penalty"). 100 independent runs were performed for each engineering problem. In the comparisons, f_{Best}, f_{Mean} and f_{Worst} are the abbreviations used to indicate the best, average and worst objective function values obtained, respectively. SR_B denotes the success rate of reaching the best known solution value. Sd gives the standard deviation of the mean objective function value; a value of Sd lower than 1.00E−10 is reported as 0. FEs gives the maximum number of objective function evaluations in each algorithm run. Note that the value of FEs may vary from algorithm to algorithm. To define the value of FEs for $\text{ACO}_{\mathbf{MV}}$, we first checked which is the smallest value of FEs across all competing algorithms; let this value be denoted by $\text{FEs}_{\min}$. Then the value of FEs for $\text{ACO}_{\mathbf{MV}}$ is set to $\text{FEs}_{\min}$. Often, however, $\text{ACO}_{\mathbf{MV}}$ reached the best known solution values for the particular problem under concern in all runs (that is, with a 100% success rate) much faster than its competitors. In such cases, for $\text{ACO}_{\mathbf{MV}}$ we give, instead of the value $\text{FEs}_{\min}$, in parenthesis the maximum number of objective function evaluations we observed across the 100 independent runs. The best solutions obtained by $\text{ACO}_{\mathbf{MV}}$ for each engineering problem are available in the supplementary information page http://iridia.ulb.ac.be/supp/IridiaSupp2011-022; there we also report details on the run time of $\text{ACO}_{\mathbf{MV}}$ on the engineering problems, which generally lies in the range of few seconds and, thus, shows that $\text{ACO}_{\mathbf{MV}}$ is a feasible alternative to other algorithms in practice.

A. *Group I : Welded beam design problem case A*

Recently, many methods have been applied to the welded beam design problem case A. Table IV shows basic summary statistics of the results obtained by nine other algorithms and $\text{ACO}_{\mathbf{MV}}$. Most other algorithms do not reach a success rate of 100% within a maximum number of objective function evaluations ranging from 30 000 (for $(\mu + \lambda)$ES [52]) to 200 000 (for CPSO [53]), while $\text{ACO}_{\mathbf{MV}}$ finds the best-known solution value in every run using at most 2 303 objective function evaluations (measured across 100 independent trials). The only other algorithm that reaches the best-known solution value in every run is DELC [54]; it does so using in every run at most 20 000 objective function evaluations (measured across 30 independent trials). Hence, $\text{ACO}_{\mathbf{MV}}$ is a very efficient and robust algorithm for this problem. The run-time behavior of $\text{ACO}_{\mathbf{MV}}$ on this problems is illustrated also in Fig. 3, where the RLD for this problem is given. The average and minimum number of objective function evaluations for $\text{ACO}_{\mathbf{MV}}$ are 2 122 and 1 888, respectively.

TABLE IV
BASIC SUMMARY STATISTICS FOR THE WELDED BEAM DESIGN PROBLEM CASE A. THE BEST-KNOWN SOLUTION VALUE IS 1.724852. f_{Best}, f_{Mean} AND f_{Worst} DENOTE THE BEST, MEAN AND WORST OBJECTIVE FUNCTION VALUES, RESPECTIVELY. SD DENOTES THE STANDARD DEVIATION OF THE MEAN OBJECTIVE FUNCTION VALUE. FES DENOTES THE MAXIMUM NUMBER OF OBJECTIVE FUNCTION EVALUATIONS IN EACH ALGORITHM RUN. FOR $\text{ACO}_{\mathbf{MV}}$ WE REPORT IN PARENTHESIS THE LARGEST NUMBER OF OBJECTIVE FUNCTION EVALUATIONS IT REQUIRED IN ANY OF THE 100 INDEPENDENT RUNS ($\text{ACO}_{\mathbf{MV}}$ REACHED IN EACH RUN OF AT MOST 20 000 EVALUATIONS THE BEST KNOWN SOLUTION VALUE). "-" MEANS THAT THE INFORMATION IS NOT AVAILABLE.

Methods	f_{Best}	f_{Mean}	f_{Worst}	Sd	FEs
GA1 [44]	1.748309	1.771973	1.785835	1.12E−02	-
GA2 [55]	1.728226	1.792654	1.993408	7.47E−02	80 000
EP [56]	1.724852	1.971809	3.179709	4.43E−01	-
$(\mu + \lambda)$ES [52]	1.724852	1.777692	-	8.80E−02	30 000
CPSO [53]	1.728024	1.748831	1.782143	1.29E−02	200 000
HPSO [57]	1.724852	1.749040	1.814295	4.01E−02	81 000
CLPSO [11]	1.724852	1.728180	-	5.32E−03	60 000
DELC [54]	1.724852	1.724852	1.724852	0	20 000
ABC [58]	1.724852	1.741913	-	3.10E−02	30 000
$\text{ACO}_{\mathbf{MV}}^{noR}$	1.724852	1.724852	1.724852	0	(2 303)
$\text{ACO}_{\mathbf{MV}}$	1.724852	1.724852	1.724852	0	(2 303)

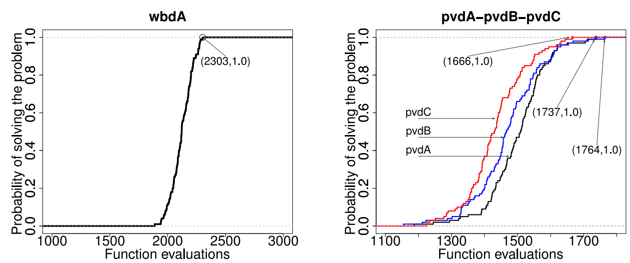


Fig. 3. The RLDs of $\text{ACO}_{\mathbf{MV}}$ for the welded beam design problem case A and the pressure vessel design problem case A, B and C (wbdA, pvdA, pvdB and pvdC are the abbreviations of those problems, respectively).

B. *Group II: Pressure vessel design problem case A, B, C and D*

There are four distinct cases (A, B, C and D) of the pressure vessel design problem defined in the literature. These cases differ by the constraints posed on the thickness of the steel used for the heads and the main cylinder. In case A, B and C (see Table V), $\text{ACO}_{\mathbf{MV}}$ reaches the best-known solution value with a 100% success rate in a maximum of 1 737, 1 764 and 1 666 objective function evaluations, respectively, while other algorithms do not reach a success rate of 100% with respect to the best-known solution value even after many more objective function evaluations. The run-time behavior of $\text{ACO}_{\mathbf{MV}}$ is illustrated in Fig. 3, where the RLDs for these problems are given.

Case D is more difficult to solve due to the larger range of side constraints for decision variables. Therefore, Case D was analyzed in more detail in recent literature. We limit $\text{ACO}_{\mathbf{MV}}$ to use a maximum number of 30 000 objective function evaluations, the same as done for several other approaches from the literature. Table VI shows clearly the second best performing algorithm for what concerns the average and the worst

TABLE V

RESULTS FOR CASE A, B AND C OF THE PRESSURE VESSEL DESIGN PROBLEM. f_{Best} DENOTES THE BEST OBJECTIVE FUNCTION VALUE. SR$_B$ DENOTES THE SUCCESS RATE OF REACHING THE BEST KNOWN SOLUTION VALUE. FEs DENOTES THE MAXIMUM NUMBER OF OBJECTIVE FUNCTION EVALUATIONS IN EACH ALGORITHM RUN. FOR ACO$_{\mathbf{MV}}$ WE REPORT IN PARENTHESIS THE LARGEST NUMBER OF OBJECTIVE FUNCTION EVALUATIONS IT REQUIRED IN ANY OF THE 100 INDEPENDENT RUNS (ACO$_{\mathbf{MV}}$ REACHED IN EACH RUN THE BEST KNOWN SOLUTION VALUE). GIVEN IS ALSO THE AVERAGE NUMBER OF OBJECTIVE FUNCTION EVALUATIONS OF THE SUCCESSFUL RUNS. "-" MEANS THAT THE INFORMATION IS NOT AVAILABLE.

Case A	NLIDP [45]	MIDCP [59]	DE [60]	ACO$^{noR}_{\mathbf{MV}}$	ACO$_{\mathbf{MV}}$		
f_{Best}	7 867.0	7 790.588	7 019.031	7 019.031	7 019.031		
SR$_B$	-	-	89.2%	100%	100%		
FEs	-	-	10 000	(1 737) (1 500.0)	(1 737) (1 500.0)		
Case B	**NLIDP [45]**	**SLA [61]**	**GA [62]**	**DE [60]**	**HSIA [8]**	**ACO$^{noR}_{\mathbf{MV}}$**	**ACO$_{\mathbf{MV}}$**
f_{Best}	7 982.5	7 197.734	7 207.497	7 197.729	7 197.9	7 197.729	7 197.729
SR$_B$	-	-	-	90.2%	-	100%	100%
FEs	-	-	-	10 000	-	(1 764) (1 470.48)	(1 764) (1 470.48)
Case C	**NLMDP [63]**	**EP [64]**	**ES [65]**	**DE [60]**	**CHOPA [66]**	**ACO$^{noR}_{\mathbf{MV}}$**	**ACO$_{\mathbf{MV}}$**
f_{Best}	7 127.3	7 108.616	7 006.9	7 006.358	7 006.51	7 006.358	7 006.358
SR$_B$	-	-	-	98.3%	-	100%	100%
FEs	-	-	4 800	10 000	10 000	(1 666) (1 433.42)	(1 666) (1 433.42)

TABLE VI

BASIC SUMMARY STATISTICS FOR THE PRESSURE VESSEL DESIGN PROBLEM CASE D. THE BEST-KNOWN OBJECTIVE FUNCTION VALUE IS 6059.7143. f_{Best}, f_{Mean} AND f_{Worst} DENOTES THE BEST, MEAN AND WORST OBJECTIVE FUNCTION VALUES, RESPECTIVELY. SD DENOTES THE STANDARD DEVIATION OF THE MEAN OBJECTIVE FUNCTION VALUE. FEs DENOTES THE MAXIMUM NUMBER OF OBJECTIVE FUNCTION EVALUATIONS IN EACH ALGORITHM RUN."-" MEANS THAT THE INFORMATION IS NOT AVAILABLE.

Methods	f_{Best}	f_{Mean}	f_{Worst}	Sd	FEs
GA1 [44]	6 288.7445	6 293.8432	6 308.1497	7.413E+00	-
GA2 [55]	6 059.9463	6 177.2533	6 469.3220	1.309E+02	80 000
$(\mu+\lambda)$ES [52]	6 059.7143	6 379.9380	-	2.10E+02	30 000
CPSO [53]	6 061.0777	6 147.1332	6 363.8041	8.645E+01	200 000
HPSO [57]	6 059.7143	6 099.9323	6 288.6770	8.620E+01	81 000
RSPSO [67]	6 059.7143	6 066.2032	6 100.3196	1.33E+01	30 000
CLPSO [11]	6 059.7143	6 066.0311	-	1.23E+01	60 000
DELC [54]	6 059.7143	6 059.7143	6 059.7143	0	30 000
ABC [58]	6 059.7143	6 245.3081	-	2.05E+02	30 000
ACO$^{noR}_{\mathbf{MV}}$	6 059.7143	6 065.7923	6 089.9893	1.22E+01	30 000
ACO$_{\mathbf{MV}}$	6 059.7143	6 059.7164	6 059.9143	1.94E−02	30 000

objective function values. In fact, ACO$_{\mathbf{MV}}$ reaches a 100% success rate (measured over 100 independent runs) at 30 717 objective function evaluations, while at 30 000 evaluations it reached a success rate of 98%, which is slightly lower than the success rate of 100% reported by DELC [54]. In fact, on this problem, ACO$_{\mathbf{MV}}$ actually profits from the possible restarts of the algorithm, as the slightly worse results of ACO$^{noR}_{\mathbf{MV}}$ show. The run-time behavior of ACO$_{\mathbf{MV}}$ is illustrated in Fig. 4, where the RLD for this problem is given. The average and minimum number of objective function evaluations is 9 448 and 1 726, respectively.

It is noteworthy that DELC [54] reaches the aforementioned performance using parameter settings that are specific for each test problem, while we use a same parameter setting for all test problems. Using instance specific parameter settings potentially biases the results in favor of the DELC algorithm. In a practical setting, one would not know a priori which parameter

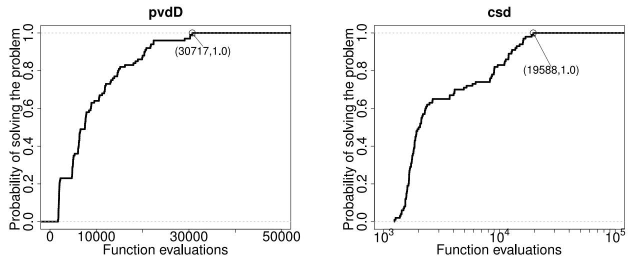


Fig. 4. The RLDs of ACO$_{\mathbf{MV}}$ for the pressure vessel design problem case D and the coil spring design problem (pvdD and csd are the abbreviations of those problems, respectively).

setting to apply before actually solving the problem. Thus, there are methodological problems in the results presented for DELC [54].

C. Group II: Coil spring design problem

Most of the research reported in the literature considering the coil spring design problem focused on reaching the best-known solution or improving the best-known one. Only recent work [60], [68] gave some attention to the number of objective functions evaluations necessary to reach the best-known solution. A comparison of the obtained results is presented in Table VII. Only a differential evolution algorithm [60] and ACO$_{\mathbf{MV}}$ obtained the best-known objective function value, 2.65856. At 8 000 evaluations ACO$_{\mathbf{MV}}$ reached a success rate of 74%, which is lower than the success rate of 95% reported by the DE algorithm of [60]; However, ACO$_{\mathbf{MV}}$ reaches a 100% success rate with 19 588 objective function evaluations because it can profit from the possibility of algorithm restarts, which generally occur after the stopping criterion of 8 000 algorithm evaluations. The run-time behavior of ACO$_{\mathbf{MV}}$ is illustrated in Fig. 4, where the RLD for this problem is given. The average and minimum number of objective function evaluations of ACO$_{\mathbf{MV}}$ are 9 948 and 1 726, respectively. It is important to note that the DE algorithm of [60] was not designed to handle categorical variables. Another DE algorithm proposed in [68] did not report a success rate, but the corresponding objective function values were reported to be in the range of $[2.658565, 2.658790]$ and the number of objective function evaluations varies in the range $[539\,960, 3\,711\,560]$, thus, showing a clearly worse performance than ACO$_{\mathbf{MV}}$.

D. Group III: Thermal insulation systems design problem

The thermal insulation systems design problem is one of the engineering problems used in the literature that deals with categorical variables. In previous studies, the categorical variables describing the type of insulators used in different layers were not considered as optimization variables, but rather as parameters. Only the more recent work of Kokkolaras et al. [16] and Abramson et al. [19], which are able to handle such categorical variables properly, consider these variables for optimization. Research focuses on improving the best-known solution value for this difficult engineering problem. ACO$_{\mathbf{MV}}$ reaches a better solution than MVP [16] and FMGPS [19];

TABLE VII

RESULTS FOR THE COIL SPRING DESIGN PROBLEM. f_{Best} DENOTES THE BEST OBJECTIVE FUNCTION VALUE. SR_B DENOTES THE SUCCESS RATE OF REACHING THE BEST KNOWN SOLUTION VALUE. FEs DENOTES THE MAXIMUM NUMBER OF OBJECTIVE FUNCTION EVALUATIONS IN EACH ALGORITHM RUN. FOR ACO_{MV} WE REPORT IN PARENTHESIS THE LARGEST NUMBER OF OBJECTIVE FUNCTION EVALUATIONS IT REQUIRED IN ANY OF THE 100 INDEPENDENT RUNS (ACO_{MV} REACHED IN EACH RUN THE BEST KNOWN SOLUTION VALUE). "-" MEANS THAT THE INFORMATION IS NOT AVAILABLE.

Algs	NLIDP [45]	GA [69]	GA [62]	DE [60]	HSIA [8]	DE [68]	ACO_{MV}^{noR}	ACO_{MV}
N	10	9	9	9	9	9	9	9
D [inch]	1.180701	1.2287	1.227411	1.223041	1.223	1.223044	1.223041	1.223041
d [inch]	0.283	0.283	0.283	0.283	0.283	0.283	0.283	0.283
f_{Best}	2.7995	2.6709	2.6681	2.65856	2.659	2.658565	2.65856	2.65856
SR_B	-	-	-	95.0%	-	-	74%	74% (100%)
FEs	-	-	-	8 000	-	-	8 000	8 000 (19 588)

TABLE VIII

COMPARISON OF THE BEST FITNESS VALUE FOR THE THERMAL INSULATION SYSTEMS DESIGN PROBLEM.

Objective function	MVP [16]	FMGPS [19]	ACO_{MV}^{noR}	ACO_{MV}
Power($\frac{PL}{A}(\frac{W}{cm})$)	25.294	25.58	24.148	24.148

Table VIII presents a new best-known objective function value, 24.148, obtained by ACO_{MV}. The corresponding solution, which has 22 continuous variable values and 11 categorical variable values, is given in the supplementary information page mentioned above. The evolution of the best solution as a function of number of objective function evaluations of ACO_{MV} is shown in Fig.5. In fact, as the number of objective function evaluations increases, the solution quality continues to improve. At 50 000 objective function evaluations, ACO_{MV} reaches the new best-known solution value 24.148.

E. Group IV: Welded beam design problem case B

The welded beam design problem case B is taken from Deb and Goyal [46] and Dimopoulos [10]. It is a variation of case A and it includes ordinal and categorical variables. Table IX shows that ACO_{MV} reaches a new best-known solution value with a 100% success rate. Additionally, the average number of objective function evaluations required by ACO_{MV} is also fewer than that of PSOA [10]. If restarts are not used, as done in version ACO_{MV}^{noR}, then slightly worse average results are obtained, which, however, are still much better than those of the other algorithms. The run-time behavior of ACO_{MV} is illustrated in Fig. 6.

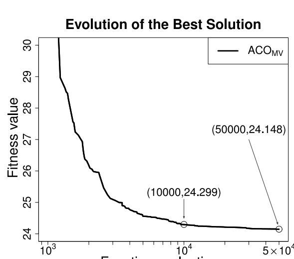


Fig. 5. The development of the best solution quality over the number of function evaluations for ACO_{MV} on the thermal insulation systems design problem.

TABLE IX

BASIC SUMMARY STATISTICS FOR WELDED BEAM DESIGN PROBLEM CASE B. f_{Best} AND f_{Mean} DENOTES THE BEST AND MEAN OBJECTIVE FUNCTION VALUES, RESPECTIVELY. SD DENOTES THE STANDARD DEVIATION OF THE MEAN OBJECTIVE FUNCTION VALUE. MEAN-FES-SUCCESS DENOTES THE AVERAGE NUMBER OF EVALUATIONS OF THE SUCCESSFUL RUNS. "-" MEANS THAT THE INFORMATION IS NOT AVAILABLE.

Methods	f_{Best}	f_{Mean}	Sd	Mean-FEs-Success
GeneAS [46]	1.9422	-	-	-
RSPSO [67]	1.9421	-	-	-
PSOA[10]	1.7631	1.7631	0	6 570
CLPSO [11]	1.5809	1.7405	2.11E−01	-
ACO_{MV}^{noR}	1.5029	1.52	4.69E−02	985
ACO_{MV}	1.5029	1.5029	0	1 436

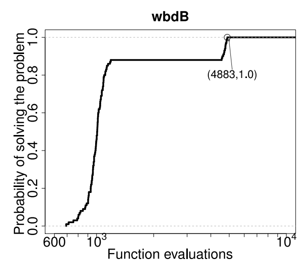


Fig. 6. The RLDs of ACO_{MV} for the welded beam design problem case B (wbdB is its abbreviation).

VI. CONCLUSIONS

In this paper, we have introduced ACO_{MV}, an ant colony optimization algorithm for tackling mixed-variable optimization problems. ACO_{MV} integrates a continuous optimization solver ($ACO_{\mathbb{R}}$), a continuous relaxation approach (ACO_{MV}-o) and a categorical optimization approach (ACO_{MV}-c) to solve continuous and mixed-variable optimization problems.

We also proposed artificial mixed-variable benchmark functions. These provide a sufficiently controlled environment for the investigation of the performance of mixed-variable optimization algorithms, and a training environment for automatic parameter tuning. Based on the benchmark functions, a rigorous comparison between ACO_{MV}-o and ACO_{MV}-c was conducted, which confirmed our expectation that ACO_{MV}-o is better than ACO_{MV}-c for ordinal variables while ACO_{MV}-c is better than ACO_{MV}-o for categorical variables.

The experimental results for real-world engineering problems illustrate that ACO_{MV} not only can tackle various classes of decision variables robustly, but also that it is efficient in finding high-quality solutions. In the welded beam design case A, ACO_{MV} is the one of the two available algorithms that reach the best-known solution with a 100% success rate; in the pressure vessel design problem case A, B and C, ACO_{MV} is the only available algorithm that reaches the best-known solution with a 100% success rate. In these four problems, ACO_{MV} does so using fewer objective function evaluations than those used by the competing algorithms. In the pressure vessel design problem case D, ACO_{MV} is one of the two available algorithms that reach the best-known solution with a 100% success rate, and it does so using only slightly more objective function evaluations than the other algorithm, which uses problem specific parameter tuning to boost algorithm performance. In the coil spring design problem, $\text{ACO}_{\mathbf{MV}}$ is the only available algorithm that reaches the best-known solution with a 100% success rate. In the thermal insulation systems design problem, $\text{ACO}_{\mathbf{MV}}$ obtains a new best solution, and in the welded beam design problem case B, $\text{ACO}_{\mathbf{MV}}$ obtains a new best solution with a 100% success rate in fewer evaluations than those used by the other algorithms.

The $\text{ACO}_{\mathbf{MV}}$ solution archive provides a flexible framework for resizing the population size and hybridization with a local search procedure to improve solutions in the archive. Thus, it would be interesting to use mechanisms such as an incremental population size and local search to further boost performance [70], [27]. We also intend to integrate or develop an effective constraint-handling technique for $\text{ACO}_{\mathbf{MV}}$ in order to tackle constrained mixed-variable optimization problems [71], [72]. A promising application for $\text{ACO}_{\mathbf{MV}}$ are algorithm configuration problems [31], in which typically not only the setting of numerical parameters but also that of categorical parameters needs to be determined. To do so, we will integrate $\text{ACO}_{\mathbf{MV}}$ into the `irace` framework [73].

VII. Acknowledgments

The research leading to the results presented in this paper has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement nº 246 939, and from the Meta-X project from the Scientific Research Directorate of the French Community of Belgium. Thomas Stützle and Marco Dorigo acknowledge support from the Belgian F.R.S.-FNRS, of which they are a Research Associate and a Research Director, respectively. Tianjun Liao acknowledges a fellowship from the China Scholarship Council.

Appendix

A. Analysis of $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c

In this section, we verify the relevance of the design choice we have taken in $\text{ACO}_{\mathbf{MV}}$, namely combining a continuous relaxation approach, $\text{ACO}_{\mathbf{MV}}$-o, and a native categorical optimization approach, $\text{ACO}_{\mathbf{MV}}$-c, in one single algorithm. We analyze the performance of $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c on two sets of the mixed-variable benchmark functions that were proposed in Section IV. The first set of benchmark functions involves continuous and ordinal variables. The second set of benchmark functions involves continuous and categorical variables.

1) Experimental Setup: For the two settings described in Section IV, we evaluate the performance of $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c on six benchmark functions with different numbers t of discrete points in the discretization, $t \in \{2, 5, 10, 20, 30, ..., 90, 100, 200, 300, ..., 900, 1\,000\}$, and dimensions 2, 6 and 10; this results in 18 groups of experiments (six benchmark functions and three dimensions) for the first and the second set of benchmark functions. In this study, half of the dimensions are continuous variables and the other half are discrete variables. The continuous variables in these benchmark functions are handled by $\text{ACO}_{\mathbb{R}}$, while the discrete variables are handled by $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c, respectively.

To ensure a fair comparison in every group of experiments, we tuned the parameters of $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c using Iterated F-Race [30], [31] with the same tuning budget on a training set of benchmark functions. The training set involves ordinal and categorical variables with a random number of t discrete points, $t \in \{2, 5, 10, 20, 30, ..., 90, 100, 200, 300, ..., 900, 1\,000\}$. In a test phase, we conducted experiments with benchmark functions different from those used in the training phase. The comparisons for each possible number t of discrete points were performed independently in each experiment group (defined by benchmark function and dimension). In total, we conducted $378 = 21 \times 6 \times 3$ comparisons for ordinal and categorical variables, respectively. In each experiment, we compare $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c without restart mechanism by measuring the solution quality obtained by 50 independent runs. A uniform random search (URS) method [74] is included as a baseline for comparison. It consists in sampling search points uniformly at random in the search domain and keeping the best solution found.

2) Comparison results: Table X summarizes the results of the comparison between $\text{ACO}_{\mathbf{MV}}$-o, $\text{ACO}_{\mathbf{MV}}$-c and URS for ordinal and categorical variables. The Wilcoxon rank-sum test at the 0.05 α-level is used to test the statistical significance of the differences in each of the 378 comparisons. In the case of ordinal variables, the statistical analysis revealed that in 63% of the 378 comparisons $\text{ACO}_{\mathbf{MV}}$-o reaches statistically significantly better solutions than $\text{ACO}_{\mathbf{MV}}$-c, in 2% of the experiments $\text{ACO}_{\mathbf{MV}}$-c is statistically significantly better than $\text{ACO}_{\mathbf{MV}}$-o, and in the remaining 35% of the cases there was no statistically significant difference. As expected, both $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c outperform URS: they perform significantly better in 98% and 93% of the cases, respectively, and they never obtain statistically significantly worse results than URS. In the case of categorical variables, the statistical analysis revealed that in 93% of the 378 comparisons $\text{ACO}_{\mathbf{MV}}$-c reaches statistically significantly better solutions than $\text{ACO}_{\mathbf{MV}}$-o and in 7% of the experiments $\text{ACO}_{\mathbf{MV}}$-o is statistically significantly better than $\text{ACO}_{\mathbf{MV}}$-c. Again, both $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c outperform URS. They perform better in 96% and 78% of the cases, respectively, and $\text{ACO}_{\mathbf{MV}}$-c never obtains statistically significantly worse results than URS.

These experiments confirm our expectation that $\text{ACO}_{\mathbf{MV}}$-o is more effective than $\text{ACO}_{\mathbf{MV}}$-c on problems with ordinal variables, while $\text{ACO}_{\mathbf{MV}}$-c is more effective than $\text{ACO}_{\mathbf{MV}}$-o on problems with categorical variables. In Fig. 7, the comparisons on $f_{Rastrigin_{MV}}$ are shown. As seen in the figure, the categorical optimization approach, $\text{ACO}_{\mathbf{MV}}$-c, reaches approximately the same objective function values no matter whether the discrete variables are ordinal or categorical. The continuous relaxation approach $\text{ACO}_{\mathbf{MV}}$-o performs better than $\text{ACO}_{\mathbf{MV}}$-c in the case of ordinal variables, but its performance is not as good when applied to the categorical case.

TABLE X
COMPARISON BETWEEN $\text{ACO}_{\mathbf{MV}}$-O, $\text{ACO}_{\mathbf{MV}}$-C AND UNIFORM RANDOM SEARCH (URS) FOR TWO SETUPS OF DISCRETE VARIABLES. FOR EACH COMPARISON, WE GIVE THE FREQUENCY WITH WHICH THE FIRST MENTIONED ALGORITHM IS STATISTICALLY SIGNIFICANTLY BETTER, INDISTINGUISHABLE, OR WORSE THAN THE SECOND ONE.

	1st setup Ordinal variables	2nd setup Categorical variables
$\text{ACO}_{\mathbf{MV}}$-o vs. $\text{ACO}_{\mathbf{MV}}$-c	0.63, 0.35, 0.02	0.07, 0.00, 0.93
$\text{ACO}_{\mathbf{MV}}$-o vs. URS	0.98, 0.02, 0.00	0.78, 0.12, 0.10
$\text{ACO}_{\mathbf{MV}}$-c vs. URS	0.93, 0.07, 0.00	0.96, 0.04, 0.00

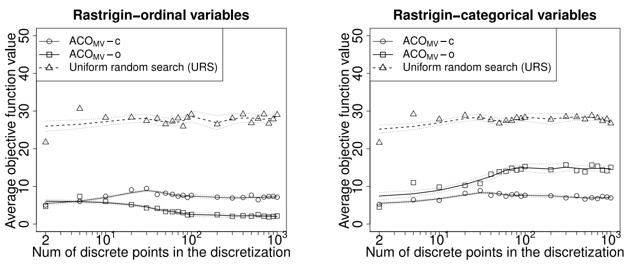


Fig. 7. The plot shows the average objective function values obtained by $\text{ACO}_{\mathbf{MV}}$-o and $\text{ACO}_{\mathbf{MV}}$-c on the 6 dimensional function $f_{Rastrigin_{MV}}$ after 10 000 evaluations, with the number t of discrete points in the discretization $t \in \{2, 5, 10, 20, 30, ..., 90, 100, 200, 300, ..., 900, 1\,000\}$.

B. Effectiveness of the restart mechanism

Here we show that $\text{ACO}_{\mathbf{MV}}$'s restart mechanism really helps in improving its performance. We conducted 50 independent runs using a maximum of 1 000 000 evaluations in each run. In Fig.8, we show $\text{ACO}_{\mathbf{MV}}$'s run-length distributions (RLDs, for short) on two multi-modal functions $f_{Ackley_{MV}}$ and $f_{Griewank_{MV}}$ with continuous and categorical variables with $t = 100$ discrete points. An empirical RLD gives the estimated cumulative probability distribution for finding a solution of a certain quality as a function of the number of objective function evaluations. (For more information about RLDs, we refer the reader to [47].) As expected, $\text{ACO}_{\mathbf{MV}}$'s performance is strongly improved by the restart mechanism. For example, in the case of $f_{Ackley_{MV}}$ in two, six and ten dimensions $\text{ACO}_{\mathbf{MV}}$ reaches a solution whose objective function value is equal to or less than 1.00E−10 with probability 1 or 100% success rate, and in the case of $f_{Griewank_{MV}}$ in two, six and ten dimensions $\text{ACO}_{\mathbf{MV}}$ reaches a solution whose objective function value is equal to or less than 1.00E−10 with probability 1, 0.82 and 0.85 respectively. Without restart, $\text{ACO}_{\mathbf{MV}}$ stagnates at much lower success rates.

C. Performance on benchmark functions

We evaluate $\text{ACO}_{\mathbf{MV}}$ on the two setups of artificial mixed-variable benchmark functions with dimensions two, six and ten. Half of the dimensions are discrete variables and the other half are continuous variables. Table XI gives the numerical results of $\text{ACO}_{\mathbf{MV}}$. The results are again measured across 50 independent runs of 1 000 000 objective function evaluations for instances with $t = 100$ discrete points. $\text{ACO}_{\mathbf{MV}}$ found a solution whose objective function value is equal to or less than 1.00E−10 with 100% success rate in all the two

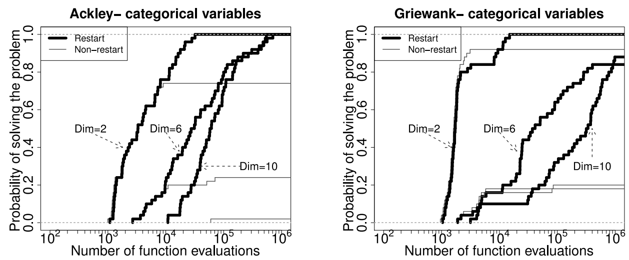


Fig. 8. The RLDs obtained by $\text{ACO}_{\mathbf{MV}}$ with and without restarts. The solution quality threshold is 1.00E−10. Dim indicates the dimensionality of the benchmark problem. Half of the dimensions are categorical variables and the other half are continuous variables.

dimensional benchmark functions. $\text{ACO}_{\mathbf{MV}}$ found solutions of the same quality (function value equal to 1.00E−10) for each of the six dimensional benchmark function at least once. On the ten dimensional benchmark functions with ordinal variables, $\text{ACO}_{\mathbf{MV}}$ found the optimal solution of $f_{Ackley_{MV}}$, $f_{Rosenbrock_{MV}}$, $f_{Sphere_{MV}}$ and $f_{Griewank_{MV}}$. On the ten dimensional benchmark functions with categorical variables, $\text{ACO}_{\mathbf{MV}}$ found the optimal solution of $f_{Ackley_{MV}}$, $f_{Sphere_{MV}}$ and $f_{Griewank_{MV}}$. Over dimension two, six and ten, $\text{ACO}_{\mathbf{MV}}$ obtained 100% success rate when applied to solve $f_{Ackley_{MV}}$ and $f_{Sphere_{MV}}$ with both setups, and obtained more than 80% success rate when applied to $f_{Griewank_{MV}}$ with both setups.

TABLE XI
EXPERIMENTAL RESULTS OF $\text{ACO}_{\mathbf{MV}}$ WITH DIMENSIONS D = 2, 6, 10. $F1 - F6$ REPRESENT $f_{Ellipsoid_{MV}}$, $f_{Ackley_{MV}}$, $f_{Rastrigin_{MV}}$, $f_{Rosenbrock_{MV}}$, $f_{Sphere_{MV}}$ AND $f_{Griewank_{MV}}$, RESPECTIVELY. THE VALUES BELOW 1.00E−10 ARE APPROXIMATED TO 0.00E+00, AND ARE HIGHLIGHTED IN **BOLDFACE**.

D	Functions	Two setups of discrete variables							
		Ordinal variables				Categorical variables			
		Avg.	Median	Max.	Min.	Avg.	Median	Max.	Min.
2	$F1$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
	$F2$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
	$F3$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
	$F4$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
	$F5$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
	$F6$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
6	$F1$	8.47e−03	**0.00e+00**	1.65e−01	**0.00e+00**	1.31e+00	4.13e−01	1.26e+01	**0.00e+00**
	$F2$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
	$F3$	1.91+00	1.78e+00	4.38e+00	**0.00e+00**	2.10e+00	2.29e+00	4.38e+00	**0.00e+00**
	$F4$	7.82e−01	**0.00e+00**	1.04e+01	**0.00e+00**	1.00e+01	6.90e+00	5.95e+01	**0.00e+00**
	$F5$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
	$F6$	2.43e−07	**0.00e+00**	1.22e−05	**0.00e+00**	8.41e−04	**0.00e+00**	1.26e−02	**0.00e+00**
10	$F1$	1.99e+00	1.40e+00	1.10e+01	1.17e−01	1.20e+01	7.32e+00	5.48e+01	5.84e−01
	$F2$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
	$F3$	1.37e+01	1.48e+01	2.46e+01	2.93e+00	1.03e+01	9.65e+00	2.03e+01	3.77e+00
	$F4$	1.23e+01	1.32e+01	3.74e+01	**0.00e+00**	4.37e+01	1.91e+01	1.80e+02	1.03e+01
	$F5$	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
	$F6$	2.54e−03	**0.00e+00**	4.67e−02	**0.00e+00**	4.52e−03	**0.00e+00**	4.67e−02	**0.00e+00**

D. Analysis of Equation (6)

To illustrate the influence of alternative choices for Equation (6) and its parameter settings, we perform three experiments on two multi-modal functions $f_{Ackley_{MV}}$ and $f_{Griewank_{MV}}$ with continuous and categorical variables with $t = 100$ discrete points. The three experiments are based on the following alternative choices for Equation (6) and its parameter settings.

(1) We modify Equation (6) to

$$w_l = \begin{cases} \omega_{j_l}, & if(u_l^i > 0), \\ 0, & if(u_l^i = 0). \end{cases} \tag{9}$$

That is, we omit the terms u_l^i and $\frac{q}{\eta}$ in Equation (6).

(2) We modify Equation (6) to

$$w_l = \begin{cases} \frac{\omega_{j_l}}{u_l^i}, & if(u_l^i > 0), \\ 0, & if(u_l^i = 0). \end{cases} \quad (10)$$

That is, we omit the term $\frac{q}{\eta}$ in Equation (6).

(3) We use five different values of parameter q in Equation (6); in particular, we choose $q \in \{0.01,\ 0.1,\ 1,\ 10,\ 100\} \times 0.05099$, where 0.05099 is the setting obtained in the parameter tuning (see Table II).

We tuned the parameters of two versions of $\text{ACO}_{\mathbf{MV}}$ that use the two alternative Equations (9) and (10), respectively, by the same automatic tuning procedure used for tuning the original $\text{ACO}_{\mathbf{MV}}$ with Equation (6) to ensure a fair comparison; for the experiments with alternative settings of parameter q, the other parameters were kept to the values shown in Table II. The detailed experimental results of the comparisons of the resulting comparisons are given in the supplementary information page http://iridia.ulb.ac.be/supp/IridiaSupp2011-022. Summary information based on RLDs are given in Fig. 9 and 10. The results of experiment (1) show that the RLDs obtained by using Equation (6) clearly dominate those obtained by using Equation (9). In fact, the success rates obtained by Equation (9) in dimensions six and ten are zero and therefore not shown in Figure 9. The same conclusions hold for experiment (2): the RLDs obtained by using Equation (6) dominate those obtained by using Equation (10) in all cases, illustrating in this way the benefit of Equation (6). Similar results are obtained in experiment (3), that is, the setting $q = 0.05099$ outperforms the other settings. The only exception is for the problems in dimension two, where a setting of $q = 10 \times 0.05099$ is competitive to $q = 0.05099$. Detailed results are available at http://iridia.ulb.ac.be/supp/IridiaSupp2011-022.

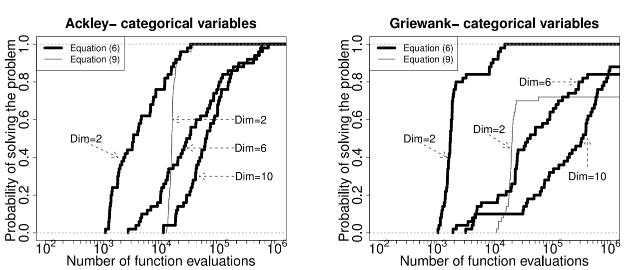


Fig. 9. The RLDs obtained by the two $\text{ACO}_{\mathbf{MV}}$ variants with Equation (6) and (9) in 50 independent runs. The solution quality threshold is 1.00E−10. Dim indicates the dimensionality of the benchmark problem. Half of the dimensions are categorical variables and the other half are continuous variables.

E. Mathematical formulation of engineering benchmark problems

1) Welded beam design problem case A: The mathematical formulation of the welded beam design problem is given in Table XII. The schematic view of this problem is shown in Fig. 11

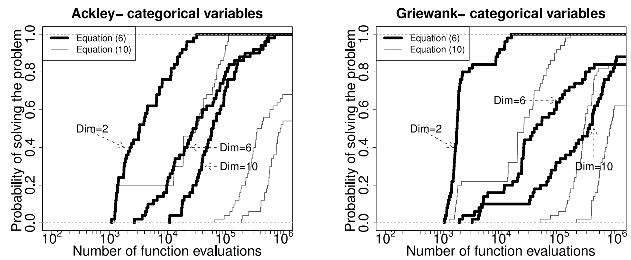


Fig. 10. The RLDs obtained by the two $\text{ACO}_{\mathbf{MV}}$ variants with Equation (6) and (10) in 50 independent runs. The solution quality threshold is 1.00E−10. Dim indicates the dimensionality of the benchmark problem. Half of the dimensions are categorical variables and the other half are continuous variables. The RLDs obtained by $\text{ACO}_{\mathbf{MV}}$ with Equation (10) in dimensions two, six and ten are correspond to the left-most, the middle and the right-most RLDs for label "Equation (10)".

TABLE XII
THE MATHEMATICAL FORMULATION OF WELDED BEAM DESIGN PROBLEM CASE A.

	$\min f(\vec{x}) = 1.10471\, x_1^2 x_2 + 0.04811\, x_3 x_4\, (14 + x_2)$
g_1	$\tau(\vec{x}) - \tau_{max} \le 0$
g_2	$\sigma(\vec{x}) - \sigma_{max} \le 0$
g_3	$x_1 - x_4 \le 0$
g_4	$0.10471\, x_1^2 + 0.04811\, x_3 x_4\, (14 + x_2) - 5 \le 0$
g_5	$0.125 - x_1 \le 0$
g_6	$\delta(\vec{x}) - \delta_{max} \le 0$
g_7	$P - P_c(\vec{x}) \le 0$
g_8	$0.1 \le x_1, x_4 \le 2.0$
g_9	$0.1 \le x_2, x_3 \le 10.0$
where	$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$ $\tau' = \frac{P}{\sqrt{2}x_1 x_2}, \tau'' = \frac{MR}{J}, M = P(L + \frac{x_2}{2})$ $R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}$ $J = 2\left\{\sqrt{2}x_1 x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$ $\sigma(\vec{x}) = \frac{6PL}{x_4 x_3^2}, \delta(\vec{x}) = \frac{4PL^3}{E x_3^3 x_4}$ $P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$ $P = 6\,000lb, L = 14in., E = 30 \times 10^6 psi, G = 12 \times 10^6 psi$ $\tau_{max} = 1\,360psi, \sigma_{max} = 30\,000psi, \delta_{max} = 0.25in.$

2) Welded beam design problem case B: The welded beam design problem case B is a variation of case A. It is extended to include two types of welded joint configuration and four possible beam materials. The changed places are shown in Equation 11 and Table XIII.

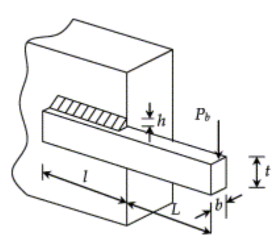


Fig. 11. Schematic view of welded beam design problem case A [49].

$$
\begin{aligned}
&\min f(\vec{x}) = (1+c_1)\, x_1^2 x_2 + c_2\, x_3 x_4\, (14+x_2) \\
&\sigma(\vec{x}) - S \leq 0 \\
&J = 2\left\{\sqrt{2}x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\},\ \textit{if } x_6 : \textit{twoside} \\
&J = 2\left\{\sqrt{2}x_1 \left[\frac{(x_1+x_2+x_3)^3}{12}\right]\right\},\ \textit{if } x_6 : \textit{fourside} \\
&\tau_{max} = 0.577 \cdot S
\end{aligned}
\tag{11}
$$

TABLE XIII

MATERIAL PROPERTIES FOR THE WELDED BEAM DESIGN PROBLEM CASE B

Methods x_5	$S(10^3 psi)$	$E(10^6 psi)$	$G(10^6 psi)$	c_1	c_2
Steel	30	30	12	0.1047	0.0481
Cast iron	8	14	6	0.0489	0.0224
Aluminum	5	10	4	0.5235	0.2405
Brass	8	16	6	0.5584	0.2566

3) Pressure vessel design problem: The pressure vessel design problem requires designing a pressure vessel consisting of a cylindrical body and two hemispherical heads such that the manufacturing cost is minimized subject to certain constraints. The schematic picture of the vessel is presented in Fig. 12. There are four variables for which values must be chosen: the thickness of the main cylinder T_s, the thickness of the heads T_h, the inner radius of the main cylinder R, and the length of the main cylinder L. While variables R and L are continuous, the thickness for variables T_s and T_h may be chosen only from a set of allowed values, these being the integer multiples of 0.0625 inch. The mathematical formulation of the four cases A, B, C and D is given in Table XIV.

4) Coil spring design problem: The problem consists in designing a helical compression spring that holds an axial and constant load. The objective is to minimize the volume of the spring wire used to manufacture the spring. A schematic of the coil spring to be designed is shown in Fig. 13. The decision variables are the number of spring coils N, the outside diameter of the spring D, and the spring wire diameter d. The number of coils N is an integer variable, the outside diameter of the spring D is a continuous one, and finally, the spring wire diameter d is a discrete variable, whose possible values are given in Table XV. The mathematical formulation is in Table XVI.

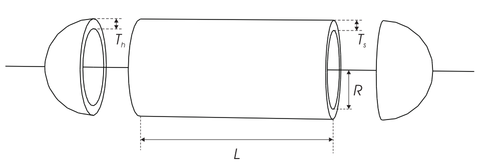


Fig. 12. Schematic view of the pressure vessel to be designed.

TABLE XIV

THE MATHEMATICAL FORMULATION THE CASES (A, B, C AND D) OF THE PRESSURE VESSEL DESIGN PROBLEM.

No	Case A	Case B	Case C	Case D
$\min f = 0.6224\, T_s R L + 1.7781\, T_h R^2 + 3.1611\, T_s^2 L + 19.84\, T_s^2 R$				
g_1	$-T_s + 0.0193\, R \leq 0$			
g_2	$-T_h + 0.00954\, R \leq 0$			
g_3	$-\pi\, R^2 L - \frac{4}{3}\pi\, R^3 + 750 \cdot 1728 \leq 0$			
g_4	$L - 240 \leq 0$			
g_5	$1.1 \leq T_s \leq 12.5$	$1.125 \leq T_s \leq 12.5$	$1 \leq T_s \leq 12.5$	$0 \leq T_s \leq 100$
g_6	$0.6 \leq T_h \leq 12.5$	$0.625 \leq T_h \leq 12.5$		$0 \leq T_h \leq 100$
g_7	$0.0 \leq R \leq 240$			$10 \leq R \leq 200$
g_8	$0.0 \leq L \leq 240$			$10 \leq L \leq 200$

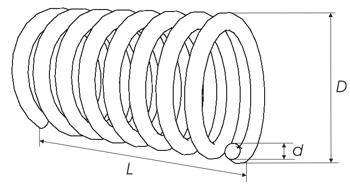


Fig. 13. Schematic view of the coil spring to be designed.

TABLE XV

STANDARD WIRE DIAMETERS AVAILABLE FOR THE SPRING COIL.

Allowed wire diameters [inch]					
0.0090	0.0095	0.0104	0.0118	0.0128	0.0132
0.0140	0.0150	0.0162	0.0173	0.0180	0.0200
0.0230	0.0250	0.0280	0.0320	0.0350	0.0410
0.0470	0.0540	0.0630	0.0720	0.0800	0.0920
0.1050	0.1200	0.1350	0.1480	0.1620	0.1770
0.1920	0.2070	0.2250	0.2440	0.2630	0.2830
0.3070	0.3310	0.3620	0.3940	0.4375	0.5000

TABLE XVI

THE MATHEMATICAL FORMULATION FOR THE COIL SPRING DESIGN PROBLEM.

$\min f_c(N,D,d) = \frac{\pi^2\, D d^2 (N+2)}{4}$	
	Constraint
g_1	$\frac{8\, C_f F_{\max} D}{\pi\, d^3} - S \leq 0$
g_2	$l_f - l_{\max} \leq 0$
g_3	$d_{\min} - d \leq 0$
g_4	$D - D_{\max} \leq 0$
g_5	$3.0 - \frac{D}{d} \leq 0$
g_6	$\sigma_p - \sigma_{pm} \leq 0$
g_7	$\sigma_p + \frac{F_{\max} - F_p}{K} + 1.05(N+2)d - l_f \leq 0$
g_8	$\sigma_w - \frac{F_{\max} - F_p}{K} \leq 0$
where	$C_f = \frac{4\frac{D}{d} - 1}{4\frac{D}{d} - 4} + \frac{0.615\, d}{D}$ $K = \frac{G d^4}{8\, N D^3}$ $\sigma_p = \frac{F_p}{K}$ $l_f = \frac{F_{\max}}{K} + 1.05(N+2)d$

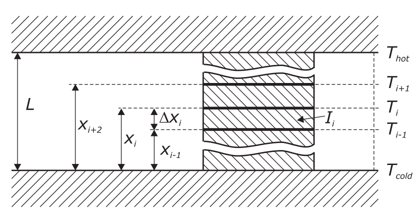


Fig. 14. Schematic view of the thermal insulation system.

5) Thermal insulation systems design problem: The schema of a thermal insulation system is shown in Fig. 14. Such a thermal insulation system is characterized by the number of intercepts, the locations and temperatures of the intercepts, and the types of insulators allocated between each pair of neighboring intercepts. In the thermal insulation system, heat intercepts are used to minimize the heat flow from a hot to a cold surface. The heat is intercepted by imposing a cooling temperature T_i at locations x_i, $i = 1, 2, ..., n$.

The basic mathematical formulation of the classic model of thermal insulation systems is defined in Table XVII. The effective thermal conductivity k of all these insulators varies with the temperature and does so differently for different materials. Considering that the number of intercepts n is defined in advance, and based on the model presented (n=10), we may define the following problem variables:

- $I_i \in \mathbf{M},\ i = 1, ..., n+1$ — the material used for the insulation between the $(i-1)$-th and the i-th intercepts (from a set of $\mathbf{M}$ materials).
- $\Delta x_i \in \mathbb{R}_+,\ i = 1, ..., n+1$ — the thickness of the insulation between the $(i-1)$-th and the i-th intercepts.
- $\Delta T_i \in \mathbb{R}_+,\ i = 1, ..., n+1$ — the temperature difference of the insulation between the $(i-1)$-th and the i-th intercepts.

This way, there are $n+1$ categorical variables chosen from a set of $\mathbf{M}$ of available materials. The remaining $2n+2$ variables are continuous.

TABLE XVII
THE MATHEMATICAL FORMULATION FOR THE THERMAL INSULATION SYSTEMS DESIGN PROBLEM.

$$f(\mathbf{x}, \mathbf{T}) = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} AC_i \left(\frac{T_{\text{hot}}}{T_i} - 1\right)\left(\frac{\int_{T_i}^{T_{i+1}} k dT}{\Delta x_i} - \frac{\int_{T_{i-1}}^{T^i} k dT}{\Delta x_{i-1}}\right)$$

	Constraint
g_1	$\Delta x_i \geq 0,\ i = 1, ..., n+1$
g_2	$T_{\text{cold}} \leq T_1 \leq T_2 \leq ... \leq T_{n-1} \leq T_n \leq T_{\text{hot}}$
g_3	$\sum_{i=1}^{n+1} \Delta x_i = L$
where	$C = 2.5$ if $T \geq 71\,\text{K}$ $C = 4$ if $71\,\text{K} > T > 4.2\,\text{K}$ $C = 5$ if $T \leq 4.2\,\text{K}$

REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley, 1989.
[2] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.
[3] J. Kennedy and R. C. Eberhart, *Swarm intelligence*. San Francisco, CA, USA: Morgan Kaufmann, 2001.
[4] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on Optimization*, vol. 7, pp. 1–25, 1997.
[5] J. Lampinen and I. Zelinka, "Mixed integer-discrete-continuous optimization by differential evolution - part 1: the optimization method," in *Proceedings of 5th International Mendel Conference of Soft Computing*, P. Ošmera, Ed. Brno University of Technology, Brno, Czech Republic, 1999, pp. 71–76.
[6] ——, "Mixed integer-discrete-continuous optimization by differential evolution. Part 2: a practical example," in *Proceedigns of 5th International Mendel Conference of Soft Computing*, P. Ošmera, Ed. Brno University of Technology, Brno, Czech Republic, 1999, pp. 77–81.
[7] N. Turkkan, "Discrete optimization of structures using a floating point genetic algorithm," in *Proceedings of the Annual Conference of the Canadian Society for Civil Engineering*. Moncton, N.B., Canada, 2003, pp. 4–7.
[8] C. Guo, J. Hu, B. Ye, and Y. Cao, "Swarm intelligence for mixed-variable design optimization," *Journal of Zhejiang University Science*, vol. 5, no. 7, pp. 851–860, 2004.
[9] S. S. Rao and Y. Xiong, "A hybrid genetic algorithm for mixed-discrete design optimization," *Journal of Mechanical Design*, vol. 127, no. 6, pp. 1100–1112, 2005.
[10] G. G. Dimopoulos, "Mixed-variable engineering optimization based on evolutionary and social metaphors," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 4-6, pp. 803 – 817, 2007.
[11] L. Gao and A. Hailu, "Comprehensive learning particle swarm optimizer for constrained mixed-variable optimization problems," *International Journal of Computational Intelligence Systems*, vol. 3, no. 6, pp. 832–842, 2010.
[12] M. H. Mashinchi, M. A. Orgun, and W. Pedrycz, "Hybrid optimization with improved tabu search," *Applied Soft Computing*, vol. 11, no. 2, pp. 1993–2006, 2011.
[13] M. A. Abramson, C. Audet, J. W. Chrissis, and J. G. Walston, "Mesh adaptive direct search algorithms for mixed variable optimization," *Optimization Letters*, vol. 3, no. 1, pp. 35–47, 2009.
[14] K. Deb and M. Goyal, "A flexible optimization procedure for mechanical component design based on genetic adaptive search," *Journal of Mechanical Design*, vol. 120, no. 2, pp. 162–164, 1998.
[15] C. Audet and J. E. Dennis, Jr., "Pattern search algorithms for mixed variable programming," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 573–594, 2001.
[16] M. Kokkolaras, C. Audet, and J. Dennis Jr., "Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system," *Optimization and Engineering*, vol. 2, no. 1, pp. 5–29, 2001.
[17] J. Ocenasek and J. Schwarz, "Estimation distribution algorithm for mixed continuous-discrete optimization problems," in *Proceedings of the 2nd Euro-International Symposium on Computational Intelligence*. IOS Press, Amsterdam, The Netherlands, 2002, pp. 227–232.
[18] M. A. Abramson, "Pattern search algorithms for mixed variable general constrained optimization problems," Ph.D. dissertation, École Polytechnique de Montréal, Canada, 2002.
[19] ——, "Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm," *Optimization and Engineering*, vol. 5, no. 2, pp. 157–177, 2004.
[20] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173, 2008.
[21] M. Dorigo, "Optimization, learning and natural algorithms *(in italian)*," Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
[22] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, vol. 26, no. 1, pp. 29–41, 1996.
[23] T. Stützle and H. H. Hoos, "$\mathcal{MAX}$–$\mathcal{MIN}$ Ant System," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.
[24] K. Socha, "ACO for continuous and mixed-variable optimization," in *Proceedings of ANTS 2004, the 4th International Conference on Swarm Intelligence*, ser. LNCS, M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, Eds., vol. 3172. Springer-Verlag, Berlin, Germany, 2004, pp. 25–36.
[25] ——, "Ant colony optimization for continuous and mixed-variable domains," Ph.D. dissertation, Université libre de Bruxelles (ULB), IRIDIA, Brussels, Belgium, 2008.
[26] X.-M. Hu, J. Zhang, H. S.-H. Chung, Y. Li, and O. Liu, "SamACO: Variable sampling ant colony optimization algorithm for continuous optimization," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 40, pp. 1555–1566, 2010.
[27] T. Liao, M. A. Montes de Oca, D. Aydin, T. Stützle, and M. Dorigo, "An incremental ant colony algorithm with local search for continuous optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'11*. New York, NY, USA: ACM, 2011, pp. 125–132.
[28] M. Schlüter, J. A. Egea, and J. R. Banga, "Extended ant colony optimization for non-convex mixed integer nonlinear programming," *Computers & Operations Research*, vol. 36, no. 7, pp. 2217–2229, 2009.
[29] A. Fetanat and G. Shafipour, "Generation maintenance scheduling in power systems using ant colony optimization for continuous domains based 0-1 integer programming," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9729–9735, 2011.
[30] P. Balaprakash, M. Birattari, and T. Stützle, "Improvement strategies for the F-Race algorithm: sampling design and iterative refinement," in *Proceedings of the 4th International Conference on Hybrid Metaheuristics*, ser. LNCS, Bartz-Beielstein *et al.*, Eds. Springer, Berlin, Germany, 2007, vol. 4771, pp. 108–122.
[31] M. Birattari, Z. Yuan, P. Balaprakash, and T. Stützle, "F-Race and iterated F-Race: An overview," in *Experimental Methods for the Analysis of Optimization Algorithms*, Bartz-Beielstein *et al.*, Eds. Springer, Berlin, Germany, 2010, pp. 311–336.
[32] S. Praharaj and S. Azarm, "Two-level non-linear mixed discrete-continuous optimization-based design: An application to printed circuit

board assemblies," *Advances in Design Automation*, vol. 1, no. 44, pp. 307–321, 1992.

[33] S. Lucidi, V. Piccialli, and M. Sciandrone, "An algorithm model for mixed variable programming," *SIAM Journal on Optimization*, vol. 15, no. 4, pp. 1057–1084, 2005.

[34] M. Stelmack and S. Batill, "Concurrent subspace optimization of mixed continuous/discrete systems," in *Proceedings of AIAA/ASME/ASCE/AHS/ASC 38th Structures, Structural Dynamic and Materials Conference*. AIAA, Kissimmee, Florida, 1997.

[35] K. Abhishek, S. Leyffer, and J. T. Linderoth, "Modeling without categorical variables: a mixed-integer nonlinear program for the optimization of thermal insulation systems," *Optimization and Engineering*, vol. 11, no. 2, pp. 185–212, 2010.

[36] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo, "Model-based search for combinatorial optimization: A critical survey," *Annals of Operations Research*, vol. 131, no. 1–4, pp. 373–395, 2004.

[37] P. Larrañaga and J. A. Lozano, Eds., *Estimation of distribution algorithms: A new tool for evolutionary computation*. Kluwer Academic Publishers, Boston, MA, 2002.

[38] M. Guntsch and M. Middendorf, "A population based approach for ACO," in *Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2002*, ser. LNCS, S. Cagnoni *et al.*, Eds., vol. 2279. Springer, Berlin, Germany, 2002, pp. 71–80.

[39] F. Hutter, H. Hoos, K. Leyton-Brown, and T. Stützle, "ParamILS: an automatic algorithm configuration framework," *Journal of Artificial Intelligence Research*, vol. 36, no. 1, pp. 267–306, 2009.

[40] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Nanyang Technological University, Tech. Rep. 2005005, 2005.

[41] A. Eiben and T. Bäck, "Empirical investigation of multiparent recombination operators in evolution strategies," *Evolutionary Computation*, vol. 5, no. 3, pp. 347–365, 1997.

[42] M. Birattari, *Tuning Metaheuristics: A Machine Learning Perspective*. Springer, Berlin, Germany, 2009.

[43] T. Liao, M. A. Montes de Oca, and T. Stützle, "Tuning Parameters across Mixed Dimensional Instances: A Performance Scalability Study of Sep-G-CMA-ES," in *Proceedings of the Workshop on Scaling Behaviours of Landscapes, Parameters and Algorithms of the Genetic and Evolutionary Computation Conference, GECCO'11*. New York: ACM, 2011, pp. 703–706.

[44] C. A. Coello Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[45] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *Journal of Mechanical Design*, vol. 112, pp. 223–229, 1990.

[46] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[47] H. Hoos and T. Stützle, *Stochastic Local Search: Foundations & Applications*. San Francisco, CA, USA: Morgan Kaufmann, 2004.

[48] E. Zahara and Y. Kao, "Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3880–3886, 2009.

[49] A. Kayhan, H. Ceylan, M. Ayvaz, and G. Gurarslan, "PSOLVER: A new hybrid particle swarm optimization algorithm for solving continuous optimization problems," *Expert Systems with Applications*, vol. 37, no. 10, pp. 6798–6808, 2010.

[50] M. Črepinšek, S.-H. Liu, and L. Mernik, "A note on teaching-learning-based optimization algorithm," *Information Sciences*, vol. 212, no. 0, pp. 79–93, 2012.

[51] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[52] E. Mezura Montes and C. A. Coello Coello, "Useful infeasible solutions in engineering optimization with evolutionary algorithms," in *MICAI 2005: Advances in Artificial Intelligence*, ser. LNCS, A. Gelbukh, . de Albornoz, and H. Terashima-Marn, Eds. Springer, Berlin, Germany, 2005, vol. 3789, pp. 652–662.

[53] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 89–99, 2007.

[54] L. Wang and L.-p. Li, "An effective differential evolution with level comparison for constrained engineering design," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 947–963, 2010.

[55] C. A. Coello Coello and E. Mezura Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," *Advanced Engineering Informatics*, vol. 16, no. 3, pp. 193–203, 2002.

[56] C. A. Coello Coello and R. L. Becerra, "Efficient evolutionary optimization through the use of a cultural algorithm," *Engineering Optimization*, vol. 36, no. 2, pp. 219–236, 2004.

[57] Q. He and L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 2, pp. 1407–1422, 2007.

[58] B. Akay and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *Journal of Intelligent Manufacturing*, In press.

[59] J.-F. Fu, R. Fenton, and W. Cleghorn, "A mixed integer-discrete-continuous programming method and its application to engineering design optimization," *Engineering Optimization*, vol. 17, no. 4, pp. 263–280, 1991.

[60] J. Lampinen and I. Zelinka, "Mechanical engineering design optimization by differential evolution," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. McGraw-Hill, London, UK, 1999, pp. 127–146.

[61] H. Loh and P. Papalambros, "Computation implementation and test of a sequential linearization approach for solving mixed-discrete nonlinear design optimization," *Journal of Mechanical Design*, vol. 113, no. 3, pp. 335–345, 1991.

[62] S.-J. Wu and P.-T. Chow, "Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization," *Engineering Optimization*, vol. 24, no. 2, pp. 137–159, 1995.

[63] H.-L. Li and C.-T. Chou, "A global approach for nonlinear mixed discrete programing in design optimization," *Engineering Optimization*, vol. 22, pp. 109–122, 1994.

[64] Y. Cao and Q. Wu, "Mechanical design optimization by mixed-variable evolutionary programming," in *Proceedings of the IEEE Conference on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 1997, pp. 443–446.

[65] G. Thierauf and J. Cai, "Evolution strategies—parallelization and application in engineering optimization," in *Parallel and distributed processing for computational mechanics: systems and tools*, B. Topping, Ed. Saxe-Coburg Publications, Edinburgh, UK, 2000, pp. 329–349.

[66] H. Schmidt and G. Thierauf, "A combined heuristic optimization technique," *Advances in Engineering Software*, vol. 36, pp. 11–19, 2005.

[67] J. Wang and Z. Yin, "A ranking selection-based particle swarm optimizer for engineering design optimization problems," *Structural and Multidisciplinary Optimization*, vol. 37, pp. 131–147, 2008.

[68] D. Datta and J. Figueira, "A real-integer-discrete-coded differential evolution algorithm: A preliminary study," in *Evolutionary Computation in Combinatorial Optimization*, ser. LNCS, P. Cowling and P. Merz, Eds. Springer, Berlin, Germany, 2010, vol. 6022, pp. 35–46.

[69] J. Chen and Y. Tsao, "Optimal design of machine elements using genetic algorithms." *Journal of the Chinese Society of Mechanical Engineers*, vol. 14, no. 2, pp. 193–199, 1993.

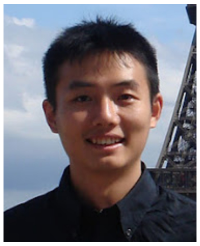
[70] M. A. Montes de Oca, T. Stützle, K. Van den Enden, and M. Dorigo, "Incremental social learning in particle swarms," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 41, no. 2, pp. 368–384, 2011.

[71] R. M. Bernardo Jr and P. C. Naval Jr, "Implementation of an ant colony optimization algorithm with constraint handling for continuous and mixed variable domains," in *Proceedings of the 10th Philippine Computing Science Congress, PCSC'10*. Computing Society of the Philippines, 2010, pp. 95–101.

[72] M. Schlüter and M. Gerdts, "The oracle penalty method," *Journal of Global Optimization*, vol. 47, no. 2, pp. 293–325, 2010.

[73] M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, and M. Birattari, "The `irace` package, iterated race for automatic algorithm configuration," IRIDIA, Université Libre de Bruxelles, Belgium, Tech. Rep. TR/IRIDIA/2011-004, 2011.

[74] S. Brooks, "A discussion of random methods for seeking maxima," *Operations Research*, vol. 6, no. 2, pp. 244–251, 1958.



Tianjun Liao received the Bachelor degree in System Engineering at the National University of Defense Technology, Changsha, China, in July 2007. He received the Ph.D. degree in Engineering Sciences at the Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA), Université Libre de Bruxelles, Brussels, Belgium, in June 2013.

Tianjun Liao is currently a postdoctoral researcher at IRIDIA, Université Libre de Bruxelles. His current research interests include heuristic optimization algorithms for continuous and mixed discrete-continuous optimization problems, and automated algorithm configuration.

Krzysztof Socha received a Master degree in computer science at the AGH University of Science and Technology, Krakow, Poland in 2000. He received the Ph.D. degree in Engineering Sciences at the Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA), Université Libre de Bruxelles, Brussels, Belgium, in 2008.

His research interests include swarm intelligence algorithms for continuous and mixed discrete-continuous optimization problems. He has also been working as a consultant in IT security and currently working as an IT security officer at the European Commission.

Marco A. Montes de Oca is a postdoctoral researcher at the Department of Mathematical Sciences at the University of Delaware since 2011. He earned his Ph.D. in Engineering Sciences at the Université Libre de Bruxelles, Brussels, Belgium. He also holds a B.S. in Computer Systems Engineering from the Instituto Politecnico Nacional, Mexico, and a M.S. in Intelligent Systems from the Instituto Tecnologico y de Estudios Superiores de Monterrey, Monterrey, Mexico.

Marco Montes de Oca is interested in the theory and practice of swarm intelligence, complex systems, and optimization. He has published in journals and conferences that deal with the three main areas of application of swarm intelligence, namely, data mining, optimization, and robotics. He is a member of the editorial board of the journal Swarm Intelligence.

Thomas Stützle received the Diplom, M.S., degree, in business engineering from the Universität Karlsruhe (TH), Karlsruhe, Germany in 1994, and the Ph.D. degree and the "Habilitation" in computer science both from the Computer Science Department of Technische Universität Darmstadt, Darmstadt, Germany in 1998 and 2004, respectively.

He is a Research Associate of the Belgian F.R.S.-FNRS working in the Institut de Recherches Interdisciplinaires et de Développements en Intelligence Atificielle (IRIDIA), Université libre de Bruxelles, Brussels, Belgium. He is author of the two books: *Stochastic Local Search: Foundations and Applications* (Morgan Kaufmann) and *Ant Colony Optimization* (MIT Press). He has published extensively in the wider area of metaheuristics (more than 160 peer-reviewed articles in journals, conference proceedings, or edited books). His research interests range from stochastic local search (SLS) algorithms, large scale experimental studies, automated design of algorithms, to SLS algorithms engineering.

Marco Dorigo (S'92–M'93–SM'96–F'06) received the Laurea, Master of Technology, degree in industrial technologies engineering in 1986, and the Ph.D. degree in electronic engineering in 1992 from the Politecnico di Milano, Milan, Italy, and the title of Agrégé de l'Enseignement Supérieur, from ULB, in 1995.

From 1992 to 1993, he was a Research Fellow at the International Computer Science Institute, Berkeley, CA. In 1993, he was a NATO-CNR Fellow, and from 1994 to 1996, a Marie Curie Fellow. Since 1996, he has been a tenured Researcher of the FNRS, the Belgian National Funds for Scientific Research, and co-director of IRIDIA, the artificial intelligence laboratory of the ULB. He is the inventor of the ant colony optimization metaheuristic. His current research interests include swarm intelligence, swarm robotics, and metaheuristics for discrete optimization. He is the Editor-in-Chief of Swarm Intelligence, and an Associate Editor or member of the Editorial Boards of many journals on computational intelligence and adaptive systems.

Dr. Dorigo is a Fellow of the ECCAI. He was awarded the Italian Prize for Artificial Intelligence in 1996, the Marie Curie Excellence Award in 2003, the Dr. A. De Leeuw-Damry-Bourlart award in applied sciences in 2005, the Cajastur International Prize for Soft Computing in 2007, and an ERC Advanced Grant in 2010.