International Journal of Computational Intelligence Systems, Vol. 9, No. 5 (2016) 957-970

Empirical Comparison of Differential Evolution Variants for Industrial Controller Design

Tony Wong [1] *, Pascal Bigras [1], Vincent Duchaîne [1], Jean-Philippe Roberge[1]

[1] *Department of Automated Manufacturing Engineering, École de technologie supérieure, 1100 Notre-Dame West, Montréal, Québec, Canada, H3C 1K3*

E-mail: {tony.wong,pascal.bigras,vincent.duchaine,jean-philippe.roberge}@etsmtl.ca

Received 21 March 2016

Accepted 21 June 2016

Abstract

To be cost-effective, most commercial off-the-shelf industrial controllers have low system order and a predefined internal structure. When operating in an industrial environment, the system output is often specified by a reference model, and the control system must closely match the model's response. In this context, a valid controller design solution must satisfy the application specifications, fit the controller's configuration and meet a model matching criterion. This paper proposes a method of solving the design problem using bilinear matrix inequality formulation, and the use of Differential Evolution (DE) algorithms to solve the resulting optimization problem. The performance of the proposed method is demonstrated by comparing a set of ten DE variants. Extensive statistical analysis shows that the variants best/1/{bin, exp} and rand-to-best/1/{bin, exp} are effective in terms of mean best objective function value, average number of function evaluations, and objective function value progression.

Keywords: Performance Measures, Differential Evolution Variants, Fixed-order Structured Controller, Model Matching, Optimization, Statistical Comparison.

1. Introduction

Traditional control design requires a full order controller: the order of the controller must always be greater than or equal to the dimension of the process model. This can be a severe restriction, especially when the process model dimension is high and the controller is an embedded device with limited memory and processing power. Moreover, traditional design techniques are unable to take into account the fixed structure of commercially available controllers, such as the popular PID (proportional, integral, derivative) feedback compensator. For the class of control problems that minimize a model matching criterion, a specific model is given and the design of the controller has to ensure that the compensated process meets the specific model's behavior. If the controller's structure is fixed, then the problem becomes one of finding its gains while minimizing the model matching error. Consequently, the design of low-order and fixed structure controllers with a model matching criterion poses a challenging problem with many practical consequences. Fortunately, the problem can be reformulated as a bilinear matrix inequality (BMI) problem, with linear matrix inequality (LMI) subproblems associated to the lower

* Corresponding author: Department of Automated Manufacturing Engineering, École de technologie supérieure, 1100 Notre-Dame West, Montréal, Québec, Canada, H3C 1K3.

and upper bound estimations. The resulting non-convex problem can be solved by the application of a branch and bound (BB) algorithm. Although BB is known to be optimal, it also has a very slow convergence speed.[1] Therefore, determining an effective alternative solution method is an ongoing research subject.

The differential evolution (DE) algorithm is a well-known population-based evolutionary optimizer designed for real-valued optimization purposes. Note that for simplicity's sake we will use "DE" to refer to "differential evolution algorithm" throughout this paper. A DE has four controlling parameters: the mutation factor (F), the crossover probability (CR), the population size (NP), and the maximum number of generations ($g_{\max}$). There are at least ten basic variants, and new variants and extensions are constantly being developed by researchers.[2] The aim here is to propose an evaluation procedure that can determine which DE variants are suitable as alternatives to the BB algorithm.

From a practical point of view, the outcome of an evaluation procedure should accurately indicate the set of DE optimizers that are effective for the particular engineering problem at hand. This indication is usually deduced from measurements made during and after algorithm executions. These measurements should cover both the efficacy and efficiency of the DEs. Here, "efficacy" is the ability to produce a desired result, and "efficiency" is the ability to produce a desired result with minimum wasted effort. Obviously, it is best to seek out DEs that are both efficacious and efficient. Since DEs are stochastic optimizers, the measurements are random variables and multiple trial runs are needed to estimate their underlying statistics. Thus, the relevant statistics are presented along with their confidence intervals, and conclusions are inferred by statistical tests.

The outline of this paper is as follows. Section 2 surveys recent experimental work on DE, with emphasis on the comparable methodological elements. Section 3 reviews different DE variants and their operating principles. Section 4 formalizes the non-convex optimization problem resulting from a BMI formulation of a structured controller with a matching model criterion. Section 5 details the performance measures and setup used during the experiments. Section 6 provides an extensive statistical analysis of the obtained results. Section 7 concludes the paper.

2. Related Works

Empirical comparison is a methodology of applied inductive research and development. Its goals are to discover knowledge through experimentation, to develop efficient solutions for a specific environment, and to validate performance claims.[3] Thus, an empirical comparison should have a stated goal, a reproducible experimental setting, a set of performance measures, and an explicit data analysis procedure. Each of these elements contributes to the inductive process that leads to the drawing of conclusions. This is the perspective adopted in the brief review of literature that follows.

In Ref. 4, Mezura-Montes *et al.* compared eight DE variants on 30 test functions with known optimal values. The DE parameters included a range of values for the mutation factor $F \in [0.3, 0.9]$, the crossover probability $CR \in \{0.0, 0.1, 0.2, \ldots, 1.0\}$, the population size $NP = 60$, the maximum number of generations $g_{\max} = 2000$, and the total number of trial runs $N_T = 100$. The reported performance measures are the mean best objective function values ($\overline{\text{BOFV}}$) and the average number of objective function evaluations ($\overline{\text{NOFE}}$) to reach the optimum, along with their respective confidence intervals (CI), which were computed by random sampling with replacement. In their work, unsuccessful trial runs are excluded from the results. It can be argued that the number of unsuccessful trial runs should not matter since the goal is to produce the best objective function value. This may be true in terms of solution accuracy, but from an efficiency point of view the success ratio is also an important performance measure.

To take into account unsuccessful trial runs, Auger *et al.*[5] divided the average number of evaluations for successful runs by the ratio of successful runs to the total number of trial runs. They considered two deterministic optimization algorithms: BFGS (Broyden–Fletcher–Goldfarb–Shanno quasi-

Newton method), and Powell's NEWUOA trusted region method. These algorithms were each compared with a DE variant (local-to-best/1/bin), CMA-ES (Covariance Matrix Adaptation Evolution Strategy), and PSO (Particle Swarm Optimization) on three test functions. Each of these optimization algorithms has its own set of controlling parameters. Auger *et al.* ran $N_T = 21$ independent trials with a maximum of 107 function evaluations for each of the test functions. Part of their results showed that NEWUOA outperformed all other algorithms on the separable Ellipsoid function up to a condition number of 106. The BFGS method performed better than PSO and CMA-ES, and the DE variant showed the worst performance. Although this multi-algorithm comparison is thoroughly comprehensive, it is difficult to make use of the findings because no relevant statistical measures were reported in their wok.

In part of Ref. 6, Jeyakumar and Shanmugavelayutham investigated the convergence nature of fourteen DE variants with fourteen test functions in 30D. They used $g_{\max} = 3000$ and $N_T = 100$, and the same values for NP, F and CR ranges as Ref. 4. They also adopted the same performance measures used in Ref. 5, with the addition of a population diversity measure defined as the Euclidean distance between the center of a population and its farthest individual.[7] The performance measures are given in terms of mean, standard deviation and variance. This is a methodological improvement compared to the preceding works; however, they based their observations on numerical inspection without statistical testing.

More recently, Sacco *et al.*[8] applied DE optimization to a nuclear core design problem. They compared the performance of two DE extensions: trigonometric mutation and opposition-based learning. Both algorithms were given the same NP, F and CR. They found that the opposition-based learning DE seemed to be more consistent and robust than the trigonometric mutation one. Here, the same set of parameter values should provide fairness and objectivity in the comparison procedure, but their conclusions were also based on simple numerical inspection.

Instead of relying on direct numerical comparison, Li *et al.*[9] used the Wilcoxon rank-sum test to compare two new DE extensions, based respectively on covariance matrix adaptation and cumulative learning evolution (JADEEP and rand/1/EP), against their original variants on twenty-eight test functions of $30D$, $50D$ and $100D$. The Wilcoxon rank-sum test is also a nonparametric test, but the null hypothesis is that paired samples come from the same population. Since the Wilcoxon rank-sum test is designed for two related samples, it suffices to compare each DE extension to its original DE variant. Similarly, Fan *et al.*[10] studied self-adaptive crossover and mutation strategies using 25 test functions and five DE variants. Their experimental setting included $N_T = 30$ independent runs, $NP = 50$, and $g_{\max} = 1000$. In their work, they applied the improved DE variants to estimate the kinetic parameters of a mercury oxidation model. Fan *et al.* highlighted the differences between the simulation and the actual values with graphical error bars. They also applied the Wilcoxon rank-sum test to compare the DE variants and the corresponding improved versions. Unfortunately, the Wilcoxon rank-sum test is only appropriate when testing two DEs, of which one may be an extension or an improved variant.

In order to compare more then two DEs, Apolloni *et al.*[11] made use of Friedman's rank test and post hoc Holm's correction to compare three variants of distributed DE on test functions of dimensions 100 and 500. Friedman's rank test is a nonparametric version of ANOVA, with data grouped in categorical factors or repeated measurements. In this case the factors are the test functions' dimensions. If the null hypothesis (no difference between the groups) is rejected, it is necessary to perform post hoc tests to find out which of the DE variants are different. Holm's correction is used in this context to control rejection errors. Olguin-Carbajal *et al.*[12] also applied the Friedman and post hoc tests to compare a micro DE with only five candidate solutions ($NP = 5$), an adjusted DE for high problem dimensionality, and a standard DE variant (rand/1/bin) using the same F for all algorithms. Both of these works performed extensive empirical comparisons. However, they both limited their experiments to the use of a single performance mea-

sure, namely the average difference between DE fitness values and the test function's optimal values.

In light of the preceding discussion, efficacy and efficiency performance measures and appropriate statistical analysis are needed to determine the effectiveness of the DE variants.

3. Differential Evolution and its Variants

Like other evolutionary algorithms, DE's main evolutionary operations are mutation, crossover and selection. The main difference between a DE and, for example, a standard genetic algorithm is that in the DE, the difference between two candidate solutions produces a difference vector to explore the search space. And whereas other advanced algorithms, such as interactive genetic algorithms, include a provision for human decision-making in the evolution process,[13,14] DEs apply the usual evolution operations to generate new candidate solutions. In DEs, mutation and crossover operations are responsible for search space exploration and the selection procedure provides the exploitation properties for the algorithm. Conceptually, the perturbation of a candidate solution, by mutation and crossover, is done by probabilistically replacing it with another solution. The latter is created by adding, to a randomly selected candidate solution, a change that is proportional to the difference between two other randomly selected solutions. Then the selection is done by a competition between the parent and its offspring. That is, the better of the two solutions moves on to the next generation.

Let D be the number of decision variables and NP the number of candidate solutions in the population. The candidate solution population at the g-th generation can be written as $\mathbf{k}_j^{(g)} = \left[k_{1,j}^{(g)} k_{2,j}^{(g)} \ldots k_{D,j}^{(g)}\right]^{\mathsf{T}}$, $j = 1,2,\ldots,NP$. Each candidate solution in the initial population is assumed to be randomly distributed on the intervals $[k_i^L, k_i^U]$, $i = 1,2,\ldots,D$, where k_i^L and k_i^U are the lower and upper bounds, respectively, of the i-th decision variable. Each of the NP candidate solutions undergoes mutation, crossover and selection operations. In a basic DE mutation operation, three candidate solutions, $\mathbf{k}_{r_1}^{(g)}$, $\mathbf{k}_{r_2}^{(g)}$ and $\mathbf{k}_{r_3}^{(g)}$, are randomly selected from the current population such that indices j, r_1, r_2 and r_3 are distinct. A new solution $\mathbf{v}_j^{(g+1)}$ is created by adding the weighted difference defined by

$$\mathbf{v}_j^{(g+1)} = \mathbf{k}_{r_1}^{(g)} + F \times (\mathbf{k}_{r_2}^{(g)} - \mathbf{k}_{r_3}^{(g)}), \qquad (1)$$

where F is the mutation factor and is a constant from $[0,2]$. Solution $\mathbf{v}_j^{(g+1)}$, which is called the donor vector in the literature, is used in the crossover operation. Of course, there are many possible variations based on Eq. 1.

Similarly, the crossover operation incorporates the current solution $\mathbf{k}_j^{(g)}$ and the donor vector to form another solution $\mathbf{u}_j^{(g+1)}$. The latter is called the trial vector and is developed from the components of the current target solution and the donor vector $\mathbf{v}_j^{(g+1)}$. Components of the donor vector enter the trial vector with probability CR. The result is the so-called binomial recombination scheme:

$$u_{i,j}^{(g+1)} = \begin{cases} v_{i,j}^{(g+1)} & \text{if } \mathsf{r} \leqslant CR \vee i = \mathsf{c} \\ k_{i,j}^{(g)} & \text{if } \mathsf{r} > CR \vee i \neq \mathsf{c} \end{cases}, \qquad (2)$$
$$i = 1,2,\ldots,D.$$

In Eq. 2, $\mathsf{r} \sim U(0,1)$ is a uniformly distributed random number, and c is a random variable following the discrete uniform distribution over the set $\{1,2,\ldots,D\}$. The use of c is to ensure that $\mathbf{u}_j^{(g+1)} \neq \mathbf{k}_j^{(g)}$ and because of this, Eq. 2 is an approximation of the binomial distribution.

Another commonly used DE crossover operation is to first choose an integer n with a discrete uniform distribution over the set $\{1,2,\ldots,D\}$. This integer acts as a starting point in the current target solution $\mathbf{k}_j^{(g)}$, from which the crossover or exchange of components with the donor vector begins. Then choose another integer l from the set $\{1,2,\ldots,D\}$ denoting the number of components the donor vector actually contributes to the target vector. After choosing n and

l the trial vector is obtained as

$$u_{i,j}^{(g+1)} = \begin{cases} v_{i,j}^{(g+1)} & \text{for } i = \langle n \rangle_D, \langle n+1 \rangle_D, \ldots, \\ & \langle n+l-1 \rangle_D \\ k_{i,j}^{(g)} & \text{for all other } i \in \{1, \ldots, D\}, \end{cases},$$
$$i = 1, 2, \ldots, D. \tag{3}$$

In Eq. 3, $\langle n \rangle_D$ is the modulo operation with modulus D.[2] To select $L \in 1, 2, \ldots, D$, start with $l = 1$ and increase its value incrementally until $U(0,1) > CR$ or $l > D$. Eq. 3 represents the exponential crossover operation. Finally, for the DE selection operation, the target solution $\mathbf{k}_j^{(g)}$ is compared to the trial vector $\mathbf{u}_j^{(g+1)}$, and the one with the lowest objective function value is admitted to the next generation:

$$\mathbf{k}_j^{(g+1)} = \begin{cases} \mathbf{u}_j^{(g+1)} & \text{if } f(\mathbf{u}_j^{(g+1)}) \leqslant f(\mathbf{h}_j^{(g)}) \\ \mathbf{k}_j^{(g)} & \text{otherwise} \end{cases}, \tag{4}$$
$$j = 1, 2, \ldots, NP.$$

From Eq. 4 it is clear that the selection procedure is a greedy one and may enable fast convergence.

3.1. Basic DE Variants

The identification scheme referencing the different DE variants is fairly consistent.[2,7] A basic DE descriptor is a triplet s/1,2/{bin, exp} where the first two elements describe the mutation type and the last element indicates the crossover type. In this descriptor, s is the source vector for perturbation from the current population. Therefore, s = best means using the best candidate solution, s = rand signifies a candidate solution is randomly chosen by the algorithm, and s = rand-to-best is a combination of s = rand and s = best. The second element {1, 2} denotes the number of weighted difference vectors used in the mutation operation. The last element {bin, exp} indicates whether the type of crossover is binomial (as in Eq. 2), or exponential (as in Eq. 3).

3.1.1. best/1/{bin, exp}

These variants use the best candidate solution in the current population, and one weighted difference vector for mutation. In this operation, two randomly selected candidate solutions are needed, such that

$$\mathbf{v}_j^{(g+1)} = \mathbf{k}_{\text{best}}^{(g)} + F \times (\mathbf{k}_{r_1}^{(g)} - \mathbf{k}_{r_2}^{(g)}). \tag{5}$$

3.1.2. rand/1/{bin, exp}

These variants use a random candidate solution in the current population, and one weighted difference vector for mutation. In this operation, three randomly selected candidate solutions are needed, such that

$$\mathbf{v}_j^{(g+1)} = \mathbf{k}_{r_1}^{(g)} + F \times (\mathbf{k}_{r_2}^{(g)} - \mathbf{k}_{r_3}^{(g)}). \tag{6}$$

3.1.3. rand-to-best/1/{bin, exp}

This mutation operation uses the best candidate solution as well as two randomly selected candidate solutions in the current population, such that

$$\mathbf{v}_j^{(g+1)} = \mathbf{k}_j^{(g)} + F \times (\mathbf{k}_{\text{best}}^{(g)} - \mathbf{k}_j^{(g)}) + F \times (\mathbf{k}_{r_1}^{(g)} - \mathbf{k}_{r_2}^{(g)}). \tag{7}$$

3.1.4. best/2/{bin, exp}

This mutation operation uses the best candidate solution in the current population and one weighted difference vector. There are four randomly selected candidate solutions involved, where

$$\mathbf{v}_j^{(g+1)} = \mathbf{k}_{\text{best}}^{(g)} + F \times (\mathbf{k}_{r_1}^{(g)} - \mathbf{k}_{r_2}^{(g)} + \mathbf{k}_{r_3}^{(g)} - \mathbf{k}_{r_4}^{(g)}). \tag{8}$$

3.1.5. rand/2/{bin, exp}

These variants use a random candidate solution in the current population and one weighted difference vector for mutation. A total of five randomly selected candidate solutions are needed, and

$$\mathbf{v}_j^{(g+1)} = \mathbf{k}_{r_5}^{(g)} + F \times (\mathbf{k}_{r_1}^{(g)} - \mathbf{k}_{r_2}^{(g)} + \mathbf{k}_{r_3}^{(g)} - \mathbf{k}_{r_4}^{(g)}). \tag{9}$$

In summary, the basic crossover operations are formulated in Eqs. 2–3. Eq. 4 describes the selection operation, which is the same for all DE variants. Finally, Eqs. 5–9 define the different mutation operations.

4. Controller Design Problem

Continuous linear time-invariant feedback control systems are generally characterized by a plant, sensors, and a feedback controller. In this work, it is assumed that the controller is structured (a constraint on the form of the controller) and has a fixed order. The gain adjustment that minimizes the difference between a reference model and the closed loop of the feedback control system results in a non-convex optimization problem. The block diagram corresponding to this system is illustrated by Fig. 1.

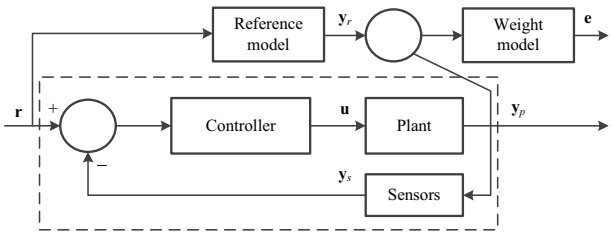

Fig. 1. Block diagram of the feedback control system including a reference model and a weight model.

Note that there exists a weight error model between the reference model and the closed loop system, which provides a more flexible control system. In Fig. 1 the reference model is described in state-space form by

$$\begin{aligned}\dot{\mathbf{x}}_r &= \mathbf{A}_r\mathbf{x}_r + \mathbf{B}_r\mathbf{r},\\ \mathbf{y}_r &= \mathbf{C}_r\mathbf{x}_r,\end{aligned} \tag{10}$$

where $\mathbf{x}_r \in \mathbb{R}^{n_r}$, $\mathbf{y}_r \in \mathbb{R}^{n_y}$, and $r \in \mathbb{R}^{n_y}$. The plant model is itself described by

$$\begin{aligned}\dot{\mathbf{x}}_p &= \mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\mathbf{u},\\ \mathbf{y}_p &= \mathbf{C}_p\mathbf{x}_p,\end{aligned} \tag{11}$$

where $\mathbf{x}_p \in \mathbb{R}^{n_p}$, $\mathbf{y}_p \in \mathbb{R}^{n_p}$, and $\mathbf{u} \in \mathbb{R}^{n_u}$. Similarly the sensor's model is described as

$$\begin{aligned}\dot{\mathbf{x}}_s &= \mathbf{A}_s\mathbf{x}_s + \mathbf{B}_s\mathbf{y}_p,\\ \mathbf{y}_s &= \mathbf{C}_s\mathbf{x}_s + \mathbf{D}_s\mathbf{y}_p,\end{aligned} \tag{12}$$

with $\mathbf{x}_s \in \mathbb{R}^{n_s}$ and $\mathbf{y}_s \in \mathbb{R}^{n_y}$. The weight error model is given by

$$\begin{aligned}\dot{\mathbf{x}}_w &= \mathbf{A}_w\mathbf{x}_w + \mathbf{B}_w\boldsymbol{\varepsilon},\\ \mathbf{e} &= \mathbf{C}_w\mathbf{x}_w,\end{aligned} \tag{13}$$

and $\boldsymbol{\varepsilon} = \mathbf{y}_r - \mathbf{y}_p = \mathbf{C}_r\mathbf{x}_p, \mathbf{x}_w \in \mathbb{R}^{n_w}$ and $\mathbf{e} \in \mathbb{R}^{n_y}$. Finally the structured controller model is described by

$$\begin{aligned}\dot{\mathbf{x}}_c &= \mathbf{A}_c(\mathbf{k})\mathbf{x}_c + \mathbf{B}_c(\mathbf{k})(\mathbf{r} - \mathbf{y}_s),\\ \mathbf{u} &= \mathbf{C}_c(\mathbf{k})\mathbf{x}_c + \mathbf{D}_c(\mathbf{k})(\mathbf{r} - \mathbf{y}_s),\end{aligned} \tag{14}$$

where $\mathbf{x}_c \in \mathbb{R}^{n_c}$. As shown in Ref. 1, the structured controller can be rearranged as

$$\begin{aligned}\begin{bmatrix}\mathbf{A}_c(\mathbf{k}) & \mathbf{B}_c(\mathbf{k})\\ \mathbf{C}_c(\mathbf{k}) & \mathbf{D}_c(\mathbf{k})\end{bmatrix} &= \begin{bmatrix}\mathbf{A}_{c0} & \mathbf{B}_{c0}\\ \mathbf{C}_{c0} & \mathbf{D}_{c0}\end{bmatrix} + \boldsymbol{\Theta}_L\mathbf{K}_s(\mathbf{k})\boldsymbol{\Theta}_R,\\ \boldsymbol{\Theta}_L\mathbf{K}_s(\mathbf{k})\boldsymbol{\Theta}_R &= \sum_{i=1}^{n_k}\boldsymbol{\Theta}_{L_i}k_i\boldsymbol{\Theta}_{R_i},\end{aligned} \tag{15}$$

where $\mathbf{k} \in \mathbb{R}^{n_k}$ is the gain vector, and $\boldsymbol{\Theta}_{L_i} \in \mathbb{R}^{n_u \times n_{k_i}}$ and $\boldsymbol{\Theta}_{R_i} \in \mathbb{R}^{n_{k_i} \times n_s}$ are full rank matrices. The system described by Eqs. 10, 11, 13, 14 and 15 can be rewritten in a single state-space model as

$$\begin{aligned}\dot{\mathbf{x}} &= (\mathscr{A} + \mathscr{B}_u\mathbf{K}_s(\mathbf{k})\mathscr{C}_u)\mathbf{x} + (\mathscr{B}_r + \mathscr{B}_u\mathbf{K}_s(\mathbf{k})\mathscr{D}_u)\mathbf{r},\\ \mathbf{e} &= \mathscr{C}\mathbf{x},\end{aligned} \tag{16}$$

with

$$\mathscr{A} = \begin{bmatrix}\mathbf{A}_r & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_{c0} & -\mathbf{B}_{c0}\mathbf{C}_s & \mathbf{0} & -\mathbf{B}_{c0}\mathbf{D}_s\mathbf{C}_p\\ \mathbf{0} & \mathbf{0} & \mathbf{A}_s & \mathbf{0} & \mathbf{B}_s\mathbf{C}_p\\ \mathbf{B}_w\mathbf{C}_r & \mathbf{0} & \mathbf{0} & \mathbf{A}_w & -\mathbf{B}_w\mathbf{C}_p\\ \mathbf{0} & \mathbf{B}_p\mathbf{C}_{c0} & -\mathbf{B}_p\mathbf{D}_{c0}\mathbf{C}_s & \mathbf{0} & \mathbf{A}_p - \mathbf{B}_p\mathbf{D}_{c0}\mathbf{D}_s\mathbf{C}_p\end{bmatrix},$$

and

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_r\\ \mathbf{x}_c\\ \mathbf{x}_m\\ \mathbf{x}_w\\ \mathbf{x}_p\end{bmatrix}, \mathscr{B}_u = \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{B}_p\end{bmatrix}, \mathscr{B}_r = \begin{bmatrix}\mathbf{B}_r\\ \mathbf{B}_{c0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{B}_p\mathbf{D}_{c0}\end{bmatrix},$$

$$\mathscr{D}_u = \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\end{bmatrix}, \mathscr{C}_u^{\mathsf{T}} = \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{I} & \mathbf{0}\\ \mathbf{0} & -\mathbf{C}_s^{\mathsf{T}}\mathbf{D}_c^{\mathsf{T}}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{0} & -\mathbf{C}_p^{\mathsf{T}}\mathbf{D}_s^{\mathsf{T}}\end{bmatrix}, \mathscr{C}^{\mathsf{T}} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{C}_w^{\mathsf{T}}\\ \mathbf{0}\end{bmatrix}.$$

4.1. Minimization Problem

For the closed loop system described by Eq. 16, the goal is to find the controller gain vector $\mathbf{k}$ that minimizes the error $\mathbf{e}$. This optimization problem can be formulated as

$$\mathbf{k} = \arg\min \|\mathbf{H}_{\varepsilon r}\|_2, \tag{17}$$

where

$$\mathbf{H}_{\varepsilon r}(s) = \mathscr{C}(s\mathbf{I} - (\mathscr{A} + \mathscr{B}_u\mathbf{K}_s(\mathbf{k})\mathscr{C}_u))^{-1}(\mathscr{B}_r + \mathscr{B}_u\mathbf{K}_s(\mathbf{k})\mathscr{D}_u),$$

and $\| \mathbf{H} \|_2$ is the 2-norm of system $\mathbf{H}$ that exists only if $\mathbf{H}$ is Hurwitz-stable. Accordingly, $\| \mathbf{H} \|_2$ is defined by

$$\|\mathbf{H}\|_2^2 = \frac{1}{2\pi}\text{trace}\int_{-\infty}^{\infty}\mathbf{H}(j\omega)\mathbf{H}(j\omega)^* d\omega.$$

In order to avoid the integral evaluation, the minimization problem given by Eq. 17 can be rewritten as

$$\mathbf{k} = \arg\min(\text{trace}(\mathscr{L}_1(\mathscr{B}_r + \mathscr{B}_u\mathbf{K}_s(\mathbf{k})\mathscr{D}_u, \mathbf{P}))), \tag{18}$$

where $\mathbf{P}$, a symmetric semi-positive definite matrix, is the solution of the following equation:

$$\mathscr{L}_2((\mathscr{A} + \mathscr{B}_u\mathbf{K}_s(\mathbf{k})\mathscr{C}_u), \mathbf{P}) + \mathbf{C}^\mathsf{T}\mathbf{C} = \mathbf{0}, \tag{19}$$

where

$$\mathscr{L}_1(\mathbf{B}, \mathbf{P}) = \mathbf{B}^\mathsf{T}\mathbf{P}\mathbf{B}, \tag{20}$$

and

$$\mathscr{L}_2(\mathbf{A}, \mathbf{P}) = \mathbf{P}\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{P}. \tag{21}$$

The optimization problem represented by Eqs. 18 – 21 is non-convex. The conventional solution technique for this problem involves the use of BB global search. However, the convergence speed is usually very slow because the BB's upper and lower bounds are difficult to obtain, and most estimations are conservative ones.

5. Performance measures and experiment settings

As presented in Section 2, comparison reports using various performance measures are typical in the literature. Depending on their intent and design, performance measures can be computed during the DE's trial run, after a trial run, or after all trial runs are completed. Several of these measures evaluate the DE's ability to find an optimal solution. In practice, finding an optimal solution requires appropriate computation resources. Thus, there exist performance measures that are used to gauge the required computation resources for the DE to achieve its goal. The following subsections define and classify the performance measures related to this work.

5.1. Efficacy Measures

Let f^* be the best known objective function value of a problem, O_i the best objective function value found by a DE variant during the i-th trial run, and $O_i^{(g)}$ the best objective function value obtained by a DE variant at generation g of the i-th trail run. Also, let n_s be the number of successful trials; that is, $n_s = \sum_{i=1}^{N_T}[|f^* - O_i| < \varepsilon]$, where ε is a small positive integer. The notation $[\cdot]$ is the Iversion bracket such that

$$[P] = \begin{cases} 1 & \text{if } P \text{ is true;} \\ 0 & \text{otherwise.} \end{cases}$$

It is thus possible to define the success ratio α of a particular DE as

$$\alpha = \frac{n_s}{N_T}. \tag{22}$$

The values of success ratio α lie between $[0,1]$ with 1 signifying 100% success. In practice, not all trial runs converge to the same objective function value, and the mean best objective function value must be computed to show a DE's average accuracy. The mean best objective function value ($\overline{\text{BOFV}}$) can be computed as

$$\overline{\text{BOFV}} = \frac{\sum_{i=1}^{N_T} O_i}{N_T}. \tag{23}$$

Since O_i is a random variable and usually of unknown distribution,[3] $\overline{\text{BOFV}}$ is estimated using the bootstrap method. To describe the relative likelihood that O_i will take on a given value, it is desirable to estimate its empirical probability density function. In other words, the probability of O_i to fall within a particular range of values is given by

the area under the density. A kernel smoothing density function estimator is used instead of the standard histogram approach, because the latter has a disproportionate effect caused by the choice of binning. A kernel estimator acts to smooth out the contribution of each observed data point over a local neighbourhood of that data point. The contribution of data point d_i to the estimate at some point d depends on the distance between d_i and d. The extent of this contribution is dependent upon the shape of the kernel function adopted and the width accorded to it. If we denote the kernel function as K and its width as h, the estimated density at any point d is $\widehat{f}(x) = \frac{1}{N_T}\sum_{i=1}^{N_T} K(\frac{d-d_i}{h})$, where $\int K(t)dt = 1$ to ensure that the estimate $f(x)$ integrates to 1. The kernel function K is usually chosen to be a smooth unimodal function with a peak at 0. In this work, the Gaussian kernel is used in the estimate of O_i.

It is often useful to measure a DE's optimization effort over time. This dynamic information can provide hints as to the convergence characteristics of an algorithm. A DE's optimization effort can be measured by its output, or by $\overline{\text{BOFV}}$ as a function of generation count. Thus, the generational mean objective function values can be computed as

$$\overline{O}^{(g)} = \frac{\sum_{i=1}^{N_T} O_i^{(g)}}{N_T}, \quad g = 1, 2, \ldots, g_{\max}\ , \tag{24}$$

where $g_{\max}$ is the maximum number of generations allowed. Again, $O_i^{(g)}$ is a random variable of unknown distribution, and its average $\overline{O}^{(g)}$ is obtained by bootstrap estimation. Since the mean objective function values are shown over many generations, it is more informative to represent $\overline{O}^{(g)}$ graphically and to include error bars representing the variable's standard error.

5.2. Efficiency Measures

For all successful trial runs, the average number of objective function evaluations, $\overline{\text{NOFE}}$, measures the average length of time for a DE to reach the best known objective function value. Define η as the set of successful trial indices, $\eta = \{i : |f^* - O_i| < \varepsilon\}, i = 1, 2, \ldots, N_T$. Then the average number of objective function evaluations is

$$\overline{\text{NOFE}} = \frac{\sum_{i\in\eta} c_i}{|\eta|}, \tag{25}$$

where c_i is the number of objective function evaluations for trial i. Once more, c_i is a random variable, so Eq. 25 and its 95% confidence interval are estimated by bootstrapping. Complementary to the $\overline{\text{NOFE}}$ is the mean best objective function value progression $\overline{\text{BOFVP}}$, defined by as[3]

$$\overline{\text{BOFVP}} = \frac{\sum_{i\in\eta} \log(\sqrt{O_i^{(0)} - O_i^{(\widehat{g})}})}{|\eta|}, \tag{26}$$

where $\widehat{g}$ is the generation in which the best solution is within ε of the best known objective function value f^*. This measure considers the difference between the initial objective function value at $g = 0$, and the best objective function value found at $g = \widehat{g}$. The measure $\overline{\text{BOFVP}}$ indicates how "far" a DE has to travel in order to reach its goal.

Lastly, it is possible to combine an efficiency measure with an efficacy measure to obtain a hybrid measurement. The Q-measure is designed to produce a scalar indicating the quality of a DE. As defined in Ref. 7, the Q-measure is the number of objective function evaluations divided by the DE's success ratio,

$$Q = \frac{\overline{\text{NOFE}}}{\alpha}. \tag{27}$$

5.3. Statistical tests

The performance data generated by a DE variant are essentially samples taken from random variables with unknown distributions. To verify this affirmation in practice, all samples are subjected to the Shapiro-Wilk's test for normality.[15] Its null hypothesis is that the performance data sample is from a population with normal distribution. If the null hypothesis is rejected, statistics of performance measures are then estimated by using the bootstrap resampling method. To provide further information on the estimation, its 95% confidence interval (CI) is estimated by the bootstrap bias-corrected and accelerated method (BCa). The BCa confidence interval is a percentile method adjusted for the skewness in the bootstrap sampling distribution.[16] When

not overlapping, the CIs can be used to determine the mean difference between two samples. However, overlapping CIs do not indicate anything that is statistically significant.[17] So, to compare DE performances, the Kruskal-Wallis rank test (or nonparametric one-way ANOVA test) with a 5% significance level is conducted in order to accept or reject the hypothesis that the mean ranks of the groups are the same, i.e., that the samples produced by the DE variants are the same for a given performance measure. It is important to select a statistical test that accounts for unequal sample sizes (when not all trial runs are successful) and to be able to test for several unmatched samples at once (since there are ten DE variants). If the null-hypothesis is rejected, then an all-possible pairwise comparison with Dunn–Sidàk significance level correction is applied to determine which DE variants are different. Dunn–Sidàk correction is known to be conservative under a dependence assumption, meaning that the actual corrected significance level is less than the prescribed 5% level.

5.4. Parameter settings

All trial runs are executed using MATLAB 8.4 on a 3 GHz PC computer. The best known controller gains ($\mathbf{k}^*$), and hence the best known objective function value f^*, are first obtained by a BB algorithm applied to the structured controller design problem instance that is specified in Fig. 2.

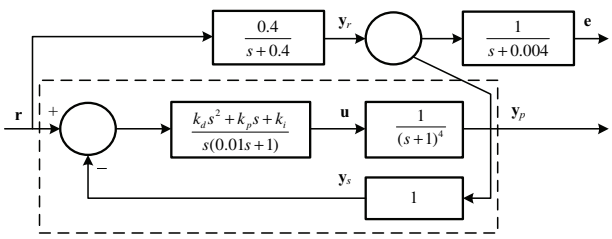

Fig. 2. An instance of the structured controller design problem.

In this problem instance, the controller has an imposed PID structure and the plant has a higher order than both the controller and the reference model. It took BB global search 54400 objective function evaluations to obtain $\mathbf{k}^* = [0.38940428\ 1.2219238\ 2.1276856]^\mathsf{T}$ and $f^* = 0.0489462$. The DE parameters and experiment variables are summarized in Table 1 and Table 2. In Table 1, D, $(\mathbf{x}^L, \mathbf{x}^U)$, N_T, NP and g_{max} are DE controlling parameters. The same values are used for all ten variants. The best known objective function value f^* is obtained by a BB algorithm, and ε determines the closeness tolerance of an objective function value to f^*. The population size NP, mutation factor F, and crossover rate CR are taken from numerical suggestions found in Ref. 2 and Ref. 18, and are manually tuned to one significant digit. The mutation factor and crossover rate are replaced by $1-F$ and $1-CR$ for DE variants that were unable to reach the vicinity of f^* during initial testing. Each DE variant has $N_T = 100$ trial runs with $g_{\max} = 100$. The aim is to determine, using these experiment settings, which of the ten DE variants is capable of solving the proposed problem instance efficaciously and efficiently.

Table 1. DE controlling parameters.

Parameter	Value
Problem dimension	$D = 3$
Decision var bounds	$\mathbf{x}^L = [0\,0\,0]^\mathsf{T}, \mathbf{x}^U = [10\,10\,10]^\mathsf{T}$
Nb of trials	$N_T = 100$
Population size	$NP = 20$
Max nb of generations	$g_{\max} = 100$
Best known obj value	$f^* = 0.0489462$
Obj function tolerance	$\varepsilon = 10^{-6}$

6. Results Analysis

In Table 3, α is the success ratio, $\overline{\text{BOFV}}$ is the mean best objective function value, $\overline{\text{NOFE}}$ is the average number of objective function evaluations, $\overline{\text{BOFVP}}$ is the average objective function value progression, and $\overline{Q}$ means the average Q-measure. Under the header of each column is the p-value computed from Kruskal-Wallis rank tests. A p-value smaller than 0.05 (5%) indicates strong evidence against the null hypothesis, which considers the corresponding results to be the same for all ten variants. A multiple comparison test is then applied to determine which of the DE variants are different. As a visual cue, the bolded numbers in Table 3 designate variants with no significant differences, and the numbers

Table 2. Mutation and crossover values for all ten DE variants.

Parameters	best/1/{bin, exp}	rand/1/{bin, exp}	rand-to-best/1/{bin, exp}	best/2/{bin, exp}	rand/2/{bin, exp}
F	0.8	0.8	0.8	0.2	0.2
CR	0.8	0.8	0.8	0.2	0.2

Table 3. Results produced by DE variants.

Variant	Efficacy		Efficiency		
	α	$\overline{\text{BOFV}}$ ($p = 4.5 \times 10^{-16}$)	$\overline{\text{NOFE}}$ ($p = 2.5 \times 10^{-156}$)	$\overline{\text{BOFVP}}$ ($p = 0.49$	$\overline{Q}$ ($p = 6.3 \times 10^{-158}$)
best/1/exp	**1**	**0.0489408**	**784.773**	**1.85763**	**784.773**
		[0.0489396, 0.0489417]	[768.137, 801.197]	[1.76135, 1.97109]	[84.394, 785.154]
best/1/bin	**1**	**0.0489406**	**791.886**	**1.88391**	**791.886**
		[0.0489394, 0.0489416]	[776.216, 808.144]	[1.79263, 1.98321]	[791.541, 792.291]
rand/1/exp	**0.98**	**0.0489515**	1426.54	**1.86809**	1455.66
		[0.0489429, 0.0489916]	[1387.2, 1466.08]	[1.75783, 1.99515]	[1454.8, 1456.51]
rand/1/bin	**0.99**	**0.0554089**	1401.83	**1.79189**	1415.99
		[0.0489406, 0.0879436]	[1369.14, 1433.4]	[1.68266, 1.91362]	[1415.21, 1416.73]
rand-to-best/1/exp	**1**	**0.0489405**	**797.472**	**1.87847**	**797.472**
		[0.0489395, 0.0489415]	[782.384, 813.515]	[1.76633, 1.99613]	[797.121, 797.82]
rand-to-best/1/bin	**1**	**0.0489407**	**802.695**	**1.73067**	**802.695**
		[0.0489395, 0.0489418]	[786.579, 820.561]	[1.63834, 1.83781]	[802.302, 803.091]
best/2/exp	**0.99**	**0.0489413**	1113.63	**1.77178**	1124.88
		[0.0489401, 0.0489423]	[1080.58, 1150.75]	[1.67938, 1.86596]	[1124.13, 1125.62]
best/2/bin	**1**	**0.0489409**	1098.76	**1.79183**	1098.76
		[0.0489398, 0.0489419]	[1063.55, 1137.12]	[1.70561, 1.87789]	[1097.94, 1099.6]
rand/2/exp	0.6	0.0580368	1954.74	**1.91234**	3257.91
		[0.0540585, 0.065564]	[1922.92, 1975.12]	[1.78742, 2.08857]	[3257.01, 3258.82]
rand/2/bin	0.52	0.0581408	1928.73	**1.78287**	3709.1
		[0.054442, 0.0642187]	[1890.51, 1958.08]	[1.65803, 1.94014]	[3707.8, 3710.53]

within brackets are the 95% confidence intervals that were computed by the BCa method. Results from multiple comparison tests for all four performance measures are shown in Fig. 3. It is used to determine which DE variants performed differently from others in terms of $\overline{\text{BOFV}}$, $\overline{\text{NOFE}}$, $\overline{\text{BOFVP}}$, and $\overline{Q}$. Fig. 4 gives the relative likelihood that the DE variants will reach their mean objective function values, and Fig. 5 presents the the mean objective function value convergence in terms of $\overline{\text{BOFV}}$ and the generation count.

As shown in Table 3, all DE variants are able to produce results close to or better than the best known objective function value f^*. Moreover, eight out of ten DE variants have a success ratio close to 1. Variants rand/2/{bin, exp} are the only ones with a lower success ratio, scoring 0.6 and 0.52 respectively. According to column $\overline{\text{BOFV}}$ in Fig. 3a and Table 3, in terms of objective function accuracy there are no statistically significant differences between best/{1, 2}/{bin, exp}, rand/1/{bin, exp}, and rand-to-best/1/{bin, exp}. Also, for these variants the empirical probability densities are located in the vicinity of f^* (Fig. 4a). For the remaining two DEs, their $\overline{\text{BOFV}}$ probability densities are scattered among several values (Fig. 4b).

In terms of efficiency, the variants best/1/{bin, exp} and rand-to-best/1/{bin, exp} have the lowest average number of objective function evaluations (Table 3, column $\overline{\text{NOFE}}$, and Fig. 12b). These four variants required on average 68 times less objective function evaluations than the BB algorithm. For a given computing system, the average number of evaluations $\overline{\text{NOFE}}$ shows the average length of time for a DE to reach f^*. This temporal efficiency measure is complemented by the average objective function value progression $\overline{\text{BOFVP}}$, which measures the "distance" traveled by the objective function value from its initial value at $g = 0$. In the case of this structured controller design problem instance, $\overline{\text{BOFVP}}$ has no significant difference among all DE variants (Fig. 12c). In other words,

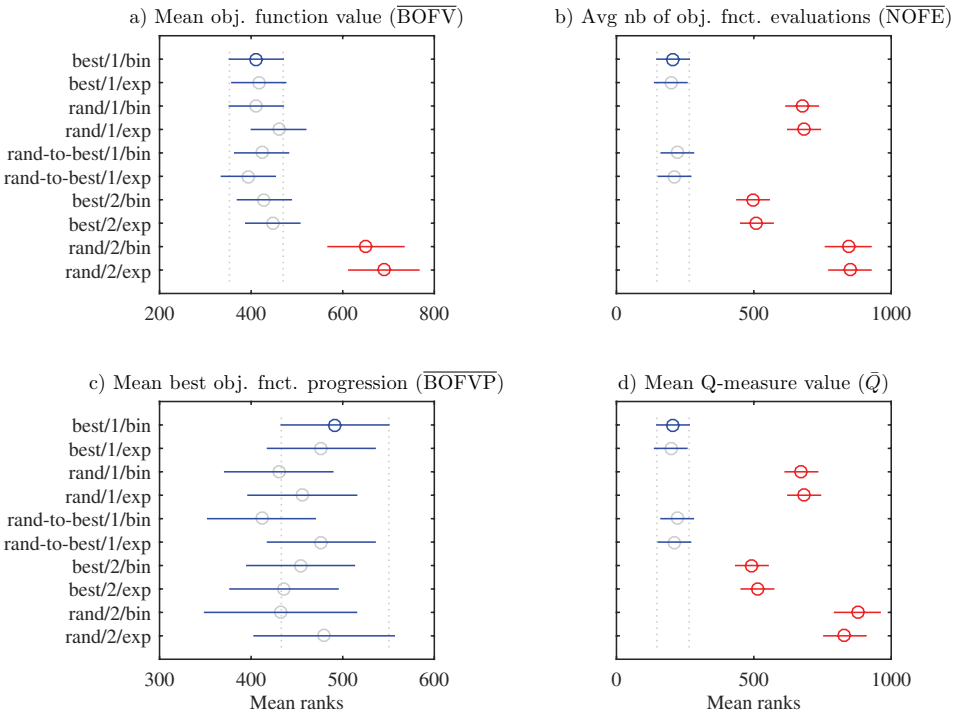

Fig. 3. Results from multiple comparison tests using Dunn–Sidàk significance level correction.

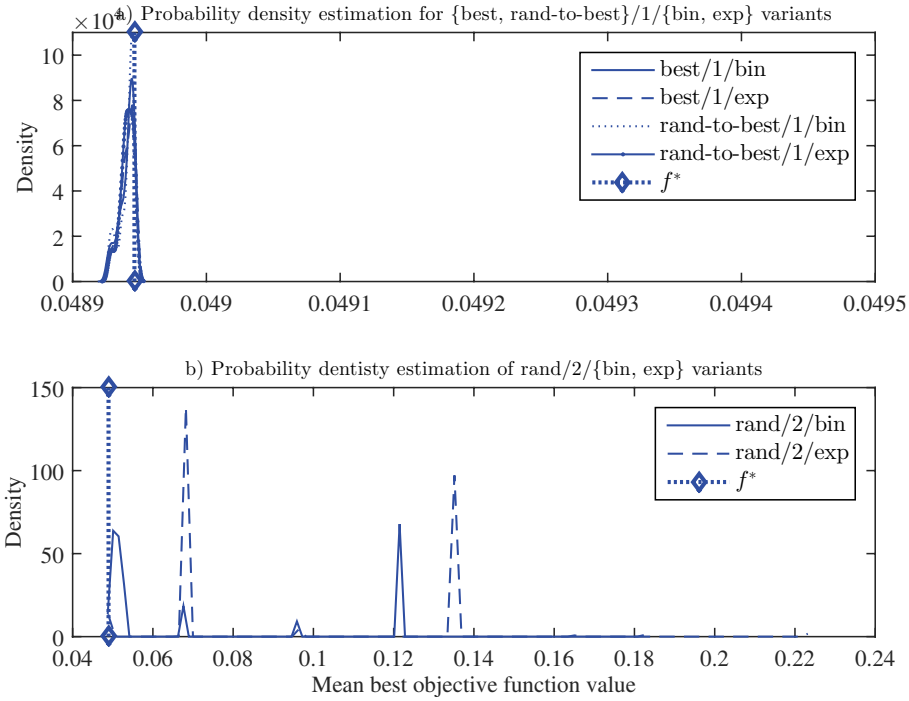

Fig. 4. Empirical probability density estimation of the mean objective function values.

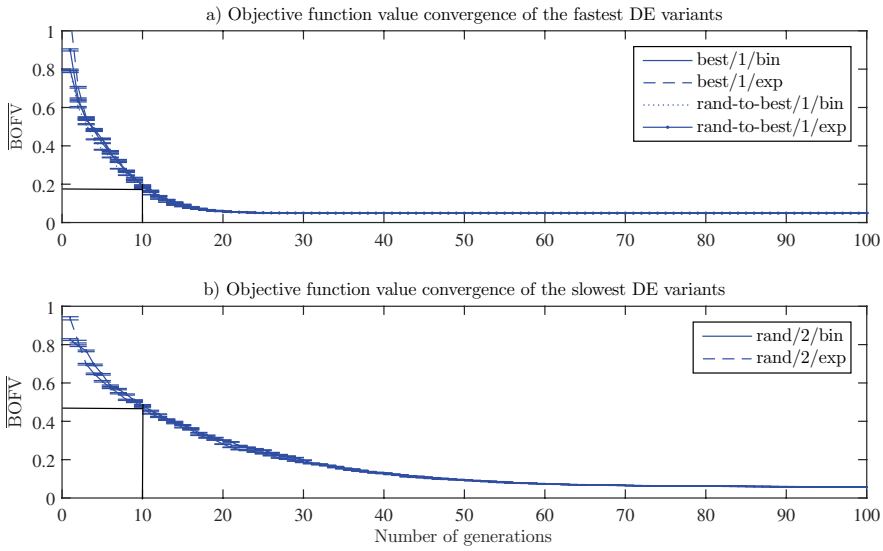

Fig. 5. Mean objective function value convergence.

all variants have more or less the same travel distances. Consequently, it is possible to discern fast and slow convergence within the variants. This is confirmed by inspecting Fig. 5. After ten generations, the $\overline{\text{BOFVP}}$ of variants best/1/{bin, exp} and rand-to-best/1/{bin, exp} is already about half of that found for the rand/2/{bin, exp} variants.

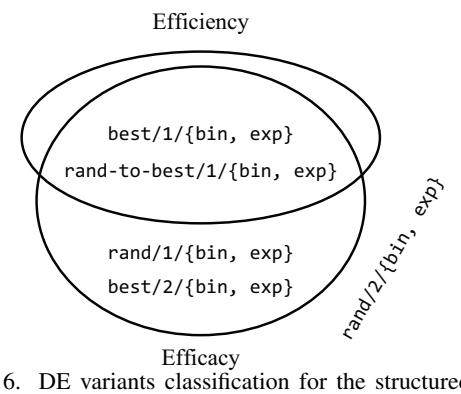

Fig. 6. DE variants classification for the structured controller design problem instance.

For this optimization instance, variants best/1/{bin, exp} and rand-to-best/1/{bin, exp} needed an average of 40 generations or 794 objective function evaluations to reach the best known objective value, whereas 95 generations were needed for rand/2/{bin, exp}. When considering both efficacy and efficiency measures, variants best/1/{bin, exp} and rand-to-best/1/{bin, exp} are equally applicable. Fig. 6 summarizes the classification of the DE variants in terms of efficacy and efficiency. For the structured controller design problem instance, variants best/1/{bin, exp} and rand-to-best/1/{bin, exp} are both efficacious and efficient. Variants rand/1/{bin, exp} and best/2/{bin, exp} are efficacious, and variants rand/2/{bin, exp} are neither efficacious nor efficient.

7. Conclusion

In this paper, ten DE variants were used to design a low-order and fixed structure controller. Their performances were compared in terms of efficacy and efficiency measures. Bootstrap estimation was used to estimate success ratio, mean best objective function values, empirical probability density, average number of objective function evaluations, and mean objective function value progression.

Inference using nonparametric multiple comparison tests indicated that variants best/1/{bin, exp} and rand-to-best/1/{bin, exp} have the best success ratios, mean objective function accuracies, and average number of objective function evaluations. The probability density estimations of their outputs also revealed their ability to reach the best

known objective function values with high probability. Variants rand/1/{bin, exp} and best/2/{bin, exp} produced the same mean objective function accuracies, but have a higher-than-average number of objective evaluations. Finally, variants rand/2/{bin, exp} showed lower objective function accuracies and higher requirements for the number of objective evaluations.

This work has shown that the proposed solution method is effective for low-order fixed structure controller design. Higher order controllers should become cost-effective in the near future, as new reconfigurable embedded hardware promises more data throughput and an increase in computing power. Thus, further research is needed to ascertain the impact on BMI formulation and DE optimization for high-order controller design. More work is also needed to overcome the difficulties of DE parameter tuning. As with other evolutionary algorithms, DEs are sensitive to control parameter changes and must be tuned. One way to alleviate the problem is to apply adaptive control mechanisms. The parameters to be adapted can be encoded directly in each individual and undergo mutation and crossover operations.[19] Good control parameter values will bring forth better individuals and propagate them to future generations. Another adaptive approach is to obtain feedback information from the search and update control parameters accordingly. Lastly, an interesting technique is to compute a population diversity measure and use it to adapt F and CR at each generation.[20]

References

1. F. Adegas and J. Stoustrup, Robust structured control design via LMI optimization, *Proceedings of the 18th IFAC world congress*, (2011) 7933–7938.
2. S. Das and P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Transactions on Evolutionary Computation*, **15** (1) (2011) 4–31.
3. T. Bartz-Beielstein, *Experimental Research in Evolutionary Computation The New Experimentalism*, (Springer-Verlag Berlin, 2006).
4. E. Mezura-Montes, J. Velazquez-Reyes and C. A. Coello Coello. A comparative study of differential evolution variants for global optimization, in *8th Annual Genetic and Evolutionary Computation Conference*, (2006) 485–492.
5. A. Auger, N. Hansen, J. P. Zerpa, R. Ros, and M. Schoenauer, Empirical comparisons of derivative free optimization algorithms, in *8th International Symposium, (SEA 2009)*, (2009) 3–15.
6. G. Jeyakumar, C. Shanmugavelayutham, Empirical measurements on the convergence nature of differential evolution variants, in *1st International Conference on Computer Science and Information Technology (CCSIT 2011)*, (2011) 472–480.
7. V. Feoktistov, *Differential evolution : in search of solutions*, (Springer New-York, 2006).
8. W. F. Sacco, A. A. De Moura Meneses, N. Henderson, Some studies on differential evolution variants for application to nuclear reactor core design, *Progress in Nuclear Energy*, **63** (2013) 49–56.
9. Y. L. Li, Z. H. Zhan, Y. J. Gong, W. N. Chen, J. Zhang and Y. Li, Differential evolution with an evolution path: a DEEP evolutionary algorithm, *IEEE Transactions on Cybernetics*, **45** (9) (2015) 1798–1810.
10. Q. Fan, X. Yan, Self-adaptive differential evolution algorithm with discrete mutation control parameters, *Expert Systems with Applications* **42** (3) (2015), 1551–1572.
11. J. Apolloni, J.Garcia-Nieto, E. Alba and G. Leguizamon, Empirical evaluation of distributed differential evolution on standard benchmarks, *Applied Mathematics and Computation*, **236** (2014) 351–366.
12. M. Olguin-Carbajal, J. C. Herrera-Lozada, J. Arellano-Verdejo, R. Barron-Fernandez and H. Taud, Micro differential evolution performance empirical study for high dimensional optimization problems, in *9th International Conference on Large-Scale Scientific Computations (LSSC 2013)*, (2014) 281–288.
13. R. Dou, C. Zong and G. Nan, Multi-stage interactive genetic algorithm for collaborative product customization, *Knowledge-Based Systems*, **92** (2016), 43–54.
14. R. Dou, C. Zong and M. Li, An interactive genetic algorithm with the interval arithmetic based on hesitation and its application to achieve customer collaborative product configuration design, *Applied Soft Computing*, **38** (2016) 384–394.
15. N. Razali, and Y. B. Wah, Power comparisons of Shapiro–Wilk, Kolmogorov–Smirnov, Lilliefors and Anderson–Darling tests, *Journal of Statistical Modeling and Analytics*, **2** (1) (2011) 21-–33.
16. M. R. Chernick and R. A. LaBudde *An Introduction to Bootstrap Methods with Applications to R*, (Wiley New Jersey, 2012).
17. M. J. Knol, W. R. Pestman and D. E. Grobbee. The mis(use) of overlap of confidence intervals to assess effect modification, *European Journal of Epidemiology* **26** (4) (2011) 253–254.

18. M. E. H. Pedersen, Good Parameters for Differential Evolution Optimization, in *Technical Report HL1001: Hvass Laboratories*, (2010) 1–10.
19. D. Jia, X. Duan and M. K. Khan, An Efficient Binary Differential Evolution with Parameter Adaptation, *International Journal of Computational Intelligence Systems*, **6** (2) (2013) 328-–336.
20. H. Lian, Y. Qin and J. Liu, An Adaptive Differential Evolution Algorithm Based on New Diversity, *International Journal of Computational Intelligence Systems*, **6** (6) (2013) 1094-–1107.