This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2713949, IEEE Transactions on Evolutionary Computation

Cooperative Co-evolution based Design Optimisation: A Concurrent Engineering Perspective

Xiaofen Lu, Stefan Menzel, Ke Tang, *Senior Member, IEEE,* and Xin Yao, *Fellow, IEEE*

***Abstract*—As a well-known engineering practice, concurrent engineering (CE) considers all elements involved in a product's life cycle from the early stages of product development, and emphasises executing all design tasks simultaneously. As a result, there exist various complex design problems in CE, which usually have many design parameters or require different disciplinary knowledge to solve them. To address these problems and enable concurrent design, different methods have been developed. The original problem is usually divided into small subproblems so that each subproblem can be solved individually and simultaneously. However, good decomposition, optimisation and communication strategies among subproblems are still needed in the field of CE. This paper attempts to study and analyse cooperative co-evolution (CC) based design optimisation in concurrent engineering by employing a parallel CC framework. Furthermore, it aims to develop new concurrent design methods based on parallel CC to solve different kinds of CE problems. To achieve this goal, a new novelty-driven CC is developed for design problems with complex structures and a novel concurrent design method is presented for quasi-separable multidisciplinary design optimisation problems. The efficacy of the new methods is studied on universal electric motor design problems and a general multidisciplinary design optimisation problem, and compared to that of some existing methods. Additionally, this paper studies how the communication frequency among subpopulations affects the performance of the proposed methods. The optimal communication frequencies under different communication costs are reported as experimental results for both proposed methods on the test problems. Based on this study, an effective self-adaptive method is proposed to be used in both optimisation schemes, which is able to adapt the communication frequency during the optimisation process.**

***Index Terms*—Concurrent engineering, cooperative co-evolution, communication frequency, self-adaptation.**

NOMENCLATURE

ATC	Analytical target cascading
ATC-MO	ATC with multi-objective formulation
CC	Cooperative co-evolution
CCDM	Co-evolutionary concurrent design method
CE	Concurrent engineering
CFS	Communication frequency self-adaptation
COSMOS	Collaborative optimisation strategy for multi-objective system
COSSOS	Collaborative optimisation strategy for single-objective system
MDO	Multidisciplinary design optimisation
MORDACE	Multidisciplinary optimisation and robust design approaches applied to concurrent engineering
NDCC	Novelty-driven CC
NDCC-MO	NDCC with multi-objective method
NDCC-MO-a	NDCC-MO with archive
NDCC-SS	NDCC with stochastic selection
NDCC-WS	NDCC with weighted sum method
NDCC-WS-a	NDCC-WS with archive

X. Lu, K. Tang (corresponding author) and X. Yao are with the OPAL Lab, Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), Shenzhen 518055, China. (E-mails: luxf@sustc.edu.cn, ketang@ustc.edu.cn, xiny@sustc.edu.cn).

S. Menzel is with Honda Research Institute Europe, 63073 Offenbach/Main, Germany. (E-mail: stefan.menzel@honda-ri.de).

X. Lu and X.Yao are also with CERCIA, School of Computer Science, University of Birmingham (UoB), Edgbaston, Birmingham B15 2TT, U.K.

I. INTRODUCTION

CONCURRENT engineering (CE) is a well-known engineering practice that can reduce product development time and cost, and improve product quality [1], [2]. It considers all elements involved in a product's life cycle (from functionality, producibility to maintenance issues, and finally disposal and recycling) from the early stages of product development, and advocates executing all design tasks in parallel or with some overlaps [3], [4]. This contrasts with traditional sequential engineering that conducts different tasks separately and at different times. Consequently, in CE, errors can be discovered earlier when the design is still flexible, and laborious re-designs can thus be avoided.

However, CE also increases the complexity of the design problem. Since it considers all parts or all aspects of the product design simultaneously, the design problem can have many design variables (if the product has a complex physical structure) or requires different disciplinary expertise to solve it. For problems with many design variables, optimising all the design variables as a whole might not be the best approach because of the curse of dimensionality. For multidisciplinary design optimisation (MDO) problems [5], the designer might not have access to the simulation models of each discipline, or the integration of different analysis codes might be too difficult [6], [7]. In these situations, decomposition-based optimisation strategies are preferred. In the literature, decomposition can be implemented in three ways, i.e., product decomposition, process decomposition and problem decomposition [8]. Product decomposition is based on the predefined product structure, while problem decomposition is carried out at the design parameter level, i.e., the optimisation task is decomposed based on the dependencies between design parameters, which might not be easily obtained beforehand. Process decomposition is more often applied in MDO problems where different disciplines share design variables and are usually coupled by coupling variables. To retain disciplinary autonomy, process decomposition strategies, such as collaborative optimisation [9], analytical target cascading (ATC) [10] and bi-level integrated system [11] break up the design problem according to engineering disciplines.

CE could benefit from evolutionary computation techniques in many aspects. For instance, CE is typically adopted in the design of complex systems, which often involve subproblems with non-convex or non-differentiable objective functions. Evolutionary algorithms (EAs) can be used as the subproblem solver in the framework of collaborative optimisation [12], [13] and ATC [14], [15]. More interestingly, the divide-and-conquer idea behind CE is essentially similar to that of cooperative co-evolution (CC). The only difference is that CC usually assumes that the problem is divided into several disjoint parts (i.e., the design parameters are divided into several disjoint groups), either prior to solving the problem based on domain knowledge [16] or automatically during the evolution [17]. On the other hand, such a decomposition does not necessarily hold for CE since the decomposition is mainly motivated by the real-world applications, and it is likely that different subproblems share the same design parameters. In the literature, although CC has been employed in CE [7], [18], [19], [20], the shared design parameters are handled as if they are normal parameters. That is, the problem is still decomposed into several fully disjoint parts, and the shared design parameters, although affecting more than one subproblem, are only evolved/updated in one subproblem. The other subproblems just take the updated variables as its input. Such a strategy on one hand makes existing CC methods directly applicable to CE, while on the other hand introduces additional questions like which shared variable should be evolved with which subproblem, and restricts the parallelism of the whole CE procedure.

Motivated by the potential of CC to CE, this work aims to investigate CC-based design optimisation in the context of CE. Specifically, new concurrent design methods based on parallel CC are proposed to solve two typical types of CE problems. The first problem is the design of a product that has several different parts, and the other is the quasi-separable MDO problem in which different disciplines share part of design variables. Quasi-separable MDO problems are frequently encountered in MDO [21] and a general MDO problem can be easily transformed into a quasi-separable MDO problem [5].

For the product design problem, the decomposition of the product is assumed to be known in prior and there is no shared design variables between different subproblems. In general, existing CC methods can be directly applied in this case. Nonetheless, to prevent CC from premature convergence to mediocre stable states, each individual in CC may need to be evaluated with a large number of collaborators [22], [23], which is inefficient and restricts the application of CC in CE. Although this situation could be alleviated by employing the novelty search approaches [24], [23] that score each individual based not only on its fitness but also on its novelty, the existing novelty calculation methods are still computationally inefficient. To address this difficulty, we propose a new novelty-driven CC which uses a new and computationally efficient novelty calculation method. In addition, it employs a stochastic selection process with an adaptive probability to adjust the trade-off between novelty and fitness. For the quasi-separable MDO problem, some design variables are shared among different disciplines, which is assumed to be known beforehand. Instead of evolving the shared variables in a singe subproblem, we propose a novel co-evolutionary concurrent design method in which the shared variables are handled through the use of duplicates and consistency constraints, where the stochastic ranking method [25] with an adaptive probability is employed to deal with the constraints. The performance of the two proposed methods is demonstrated through comparisons to the method that optimises all parts (disciplines) as a whole as well as other state-of-the-art methods on universal electric motor (UEM) design problems and a commonly used MDO test problem.

In a real-world concurrent design process, communications among different subtasks will introduce a cost and the cost might be different for different problems and environments. As a result, the exact time point to exchange information is not easy to determine. In the field of CE, how to set the communication frequency between different subtasks is a very important research topic and researchers have proposed a lot of approaches [26], [27], [1], [28], [29]. However, these approaches were developed for concurrent design of dependent tasks and the information exchange is unidirectional. There is only very limited research on communication frequency in cases that the subproblems are interdependent and the information exchange is bi-directional, which hold for the two CE problems considered in this work. Hence, extensive studies have been conducted to identify the best communication frequency among different subpopulations in the two proposed co-evolutionary concurrent methods. On this basis, a self-adaptive scheme is also proposed to automatically adjust the communication frequency among subpopulations during the optimisation process.

The rest of this paper is organised as follows. Section II will give the related work. Section III details the proposed co-evolutionary methods. In Section IV, the comparison results between the proposed methods and existing methods will be presented on the test problems and the best communication frequencies of them with/without communication costs will be investigated. Section V will describe the self-adaptive method for finding the best communication frequencies and report the comparison results. In Section VI, the conclusion and directions of future work will be given.

II. Background and Related Work

A. Problem Formulation

In this paper, two kinds of complex design problems, which are frequently encountered in CE, are considered. They are:

1) mono-disciplinary product design with multiple parts,
2) quasi-separable MDO problems.

For the first kind of problem, assume the product is composed of m parts and f denotes the performance metric of the design, the problem formulation can be given as follows:

$$\min_{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m} f(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m) \tag{1}$$

where $\boldsymbol{x}_i$ denotes the design parameter vector related to the i-th part of the product $(i = 1, 2, ..., m)$. Note that we assume throughout this paper that the decomposition of the product is known in prior and there is no shared design variables between different subproblems.

For the second kind of design problem, different disciplines aim to find a complete design that improves different physical aspects of a product. A quasi-separable MDO problem [21] with m disciplines can be formulated as:

$$\min_{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m, \boldsymbol{z}} \sum_{i=1}^{m} f_i(\boldsymbol{x}_i, \boldsymbol{z}) \tag{2}$$

where $(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m, \boldsymbol{z})$ denotes the design parameter vector of the entire design. $\boldsymbol{z}$ denotes the shared design parameter vector among disciplines. $\boldsymbol{x}_i$ denotes the local design parameter vector with respect to the i-th discipline and f_i denotes the objective of the i-th discipline $(i = 1, 2, ..., m)$. The final design from each discipline must be consistent on $\boldsymbol{z}$. Note that we assume throughout this paper that the shared design variables are known beforehand for this kind of problem.

B. Cooperative Co-evolution (CC)

A simple EA is given in Algorithm 1. The EA begins with a population of candidate solutions, $\boldsymbol{P}_G = \{\boldsymbol{x}_{i,G} | i = 1, 2, ..., NP\} (G = 0)$. Here, G denotes the generation number, NP is the population size, and each $\boldsymbol{x}_{i,G}$ denotes a candidate solution. After initialization, the EA iteratively uses selection, recombination, and mutation operators at each generation G to evolve the population until a stopping criterion is met. The current best $\boldsymbol{x}_{i,G}$ is the output.

Algorithm 1 The Framework of EA

```
1: Initialize a population P_G = {x_{i,G} | i = 1, 2, ..., NP}
2: Evaluate P_G
3: while the stopping criterion is not met do
4:     Select parents P_S from P_G based on each x_{i,G}'s fitness
5:     Recombine P_S to get P'_S
6:     Mutate P'_S to get P''_S
7:     Set P_{G+1} = P''_S and then G = G + 1
8: end while
```

In a CC framework, the decision variables are firstly decomposed into several subcomponents, and then each subcomponent is optimised separately by a subpopulation. The whole optimisation process of CC is decomposed into several cycles. In each cycle, each subpopulation is evolved with a specified EA for a fixed number of function evaluations. For parallel CC, all subpopulations are evolved simultaneously. At the end of each cycle, different subpopulations communicate with each other through collaborative individuals in their current populations, which are used to evaluate the individuals in the other subpopulations in the next cycle. In the literature, there exist different methods to select collaborative individuals [30] and the most commonly used method is to choose the best individual in the concurrent population. Fig. 1 shows a parallel CC optimisation process. In this figure, the decision variables are decomposed into two subcomponents ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$). f denotes the objective function. Each of t_1, t_2, t_3 denotes a communication time point between two subpopulations (i.e., Pop_1 and Pop_2).

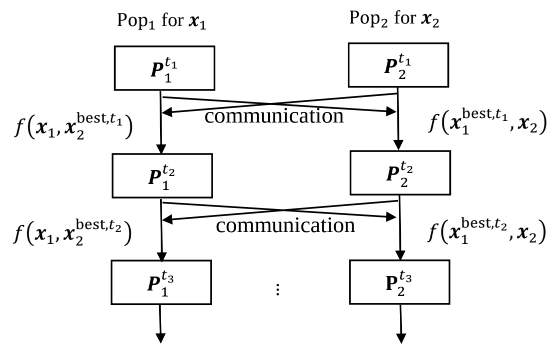

Fig. 1. Parallel Cooperative Co-evolutionary Optimisation Process.

C. Novelty-Driven Cooperative Co-evolution

Cooperative co-evolution encounters several issues as the fitness of individuals in each subpopulation depends highly on the collaborative individuals exchanged from the other subpopulations. It can easily be misled by the chosen collaborators and get trapped in the suboptimal equilibrium states in which changing each of the team members will result in lower performance [23]. Moreover, CC tends to identify local optima that have large basins of attraction [31]. However, these local optima may not correspond to global optima. This is known as *relative overgeneralisation* [32].

To prevent CC from converging to such mediocre stable states, researchers in [23], [24] have proposed considering not only the fitness but also the novelty of individuals when scoring them. Novelty search is a recently proposed evolutionary approach to solve deceptive problems [33]. In the original novelty search method, individuals are scored based on their behavioural novelty rather than fitness. This scheme makes the evolution continuously explore the regions of individuals with behavioural innovation rather than converging to one region. A novelty metric was proposed in [34] to measure how far an individual is from other individuals in the behaviour space, which calculates the novelty score of each individual as follows:

$$nov(\boldsymbol{x}) = \frac{1}{k} \sum_{i=1}^{k} dist(\boldsymbol{x}, \boldsymbol{\mu}_i) \tag{3}$$

where $\boldsymbol{\mu}_i$ is the i-th nearest neighbour of the individual $\boldsymbol{x}$ according to the distance metric $dist$. The neighbours include the other individuals in the current population and optionally archived past individuals. In previous work, the archive is composed of the most novel individuals [34], [35] or stochastically selected individuals from every generation [36]. This kind of novelty-based optimisation has shown better performance than the fitness-based optimisation methods in many different applications.

The use of novelty as the sole criterion does not always result in satisfying outcomes. It has been found that selecting individuals based on only novelty scores does not necessarily

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2713949, IEEE Transactions on Evolutionary Computation

lead to a high average fitness [37]. Researchers have combined novelty and fitness objectives together to score individuals and shown through experimental studies that this is an effective way to make use of novelty [23]. Different combination methods have been proposed in the literature [37], [38], [39]. In [37], the authors achieved the combination through a linearly weighted sum of them as shown in Eq. (4).

$$score(\boldsymbol{x}) = (1-\rho) * f_{\text{norm}}(\boldsymbol{x}) + \rho * nov_{\text{norm}}(\boldsymbol{x}) \quad (4)$$

where ρ is used to control the importance of the fitness and novelty and kept fixed during the optimisation process, and $f_{\text{norm}}(\boldsymbol{x})$ and $nov_{\text{norm}}(\boldsymbol{x})$ denote the normalised fitness and novelty, respectively. The normalisation process is proceeded according to:

$$f_{\text{norm}}(\boldsymbol{x}) = \frac{f(\boldsymbol{x}) - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}}, nov_{\text{norm}}(\boldsymbol{x}) = \frac{nov(\boldsymbol{x}) - nov_{\text{min}}}{nov_{\text{max}} - nov_{\text{min}}} \quad (5)$$

where f_{min} and f_{max} are the lowest and highest fitness in the current population, respectively; nov_{min} and nov_{max} are the corresponding lowest and highest novelty scores. In [38], a multi-objective evolutionary algorithm, non-dominated sorting genetic algorithm-II [40], was employed to balance between the novelty and fitness objectives. The weighted sum method and the multi-objective combination method were applied in CC in [24] and [23], respectively.

One difficulty in the use of the aforementioned novelty-driven CC methods is that the novelty calculation is computationally inefficient. The time complexity of calculating the novelty for a whole population according to Eq. (3) without considering archived individuals is $\mathcal{O}(NP^2 * \text{complexity}(dist) + NP^2 log(NP))$ (NP denotes population size). Moreover, using either a fixed weight or multi-objective method might not be the best trade-off approach as it might be better to change the emphasis to exploration or exploitation as the evolution proceeds.

D. Use of Evolutionary Algorithms in MDO

A general multidisciplinary design optimisation problem involves multiple disciplines that share some design variables and are coupled by coupling variables. Fig. 2 shows a two-disciplinary system. In this figure, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ denotes local variable vectors for disciplines 1 and 2, respectively; $\boldsymbol{x}_{\text{c}}$ stands for the common variable vector; f_1 and f_2 denote the corresponding objective functions of disciplines 1 and 2; g_1 and g_2 stand for the constraints of disciplines 1 and 2, respectively; f_{system} and g_{system} are the system objective and constraint functions, respectively. The f_{system} is a function of f_1 and f_2. Disciplines 1 and 2 are coupled by the coupling variables, $\boldsymbol{y}_{12}$ and $\boldsymbol{y}_{21}$. They are the output from the corresponding analysis of disciplines 1 and 2, and needed in calculating the objective and constraint functions of disciplines 2 and 1, respectively. In this paper, we focus on MDO problems that are coupled through only shared design variables, i.e., quasi-separable MDO problems, which are frequently encountered in MDO and the formulation for which is given in Eq. (2).

An MDO problem can be solved as a single optimisation problem. However, this method requires a high integration of different disciplines, which is impractical in some cases, e.g., when each disciplinary analysis is developed under a different specialised computer code and it is hard to combine these codes [21]. To solve MDO problems in such cases, researchers have developed distributed optimisation methods based on decomposition, which allow disciplinary autonomy. Since EAs can solve non-differential and multimodal optimisation problems, they are more and more widely employed as optimisers in MDO. Existing EA-based distributed optimisation methods for MDO include the MORDACE method (multidisciplinary optimisation and robust design approaches applied to concurrent engineering) [41], the COSMOS method (collaborative optimisation strategy for multi-objective systems) [42], and the ATC method (analytical target cascading) with the multi-objective formulation [14]. The following paragraphs will explain each of these methods in more detail.

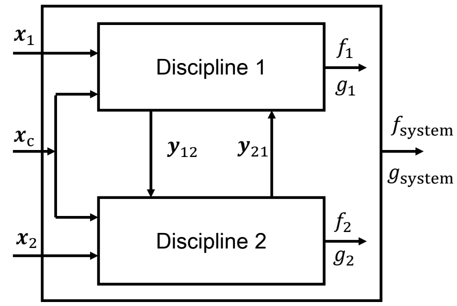

Fig. 2. A two-disciplinary system.

In the MORDACE method [41], each disciplinary optimisation is performed independently. When disciplinary optimisation is finished, the MORDACE method employs a compromise method on the common decision variables. As changes in common variable values due to compromise will make each disciplinary performance worse, a robust design approach is employed in each disciplinary optimisation to alleviate this. In the robust design approach, in addition to disciplinary objectives, disciplinary optimisation also aims to minimise the sensitivity of performance values to variations of common variables.

The COSMOS method [42] tries to solve multi-objective problems in a multidisciplinary context. It uses a nested decomposition and includes two-level optimisation: supervisor-level and disciplinary-level optimisation. The supervisor level optimises the common variables and provides the values of common variables to the disciplinary level. The disciplinary level optimises the disciplinary variables based on the given common variable values. It then returns the best function value to the supervisor level, which will then be used as the performance value of the given common variables. It can be seen that the optimisation in the disciplinary level aims to calculate the fitness of the given common variable values, and thus is a nested optimisation. In the COSMOS method, the supervisor level employs a MOGA to search the best common variable values based on the performance values returned from the disciplinary-level optimisation.

The ATC method [14] is a hierarchical multi-level methodology. It propagates system targets through a hierarchical structure and minimises the unattainability when the targets are unattainable [5]. When using ATC to solve the MDO problem in Fig. 2, copies of common design variables and coupling variables and consistency constraints are created to make the common variables and coupling variables consistent in the final design. ATC uses a penalty function to deal with the consistency constraints. Fig. 3 illustrates the decomposition and coordination of ATC in solving the problem in Fig. 2. In this figure, T denotes the design targets. The $\boldsymbol{x}_\mathrm{c}^\mathrm{u}$ are the duplicate common variables for system optimisation; $\boldsymbol{x}_\mathrm{c}^\mathrm{l1}$ and $\boldsymbol{x}_\mathrm{c}^\mathrm{l2}$ are for disciplinary optimisation ($\boldsymbol{x}_\mathrm{c}^\mathrm{u} = \boldsymbol{x}_\mathrm{c}^\mathrm{l1} = \boldsymbol{x}_\mathrm{c}^\mathrm{l2}$ at optimality). The $\boldsymbol{y}_{12}^\mathrm{u}$ and $\boldsymbol{y}_{21}^\mathrm{u}$, and $\boldsymbol{y}_{12}^\mathrm{l}$ and $\boldsymbol{y}_{21}^\mathrm{l}$ are duplicate coupling variables for system and disciplinary optimisation, respectively ($\boldsymbol{y}_{12}^\mathrm{u} = \boldsymbol{y}_{12}^\mathrm{l}$ and $\boldsymbol{y}_{21}^{u} = \boldsymbol{y}_{21}^\mathrm{l}$ at optimality). The system-level and discipline-level optimisations are carried out in turn in a loop. When the system-level optimisation is completed, the optimal f_1^u, f_2^u, $\boldsymbol{x}_\mathrm{c}^\mathrm{u}$, $\boldsymbol{y}_{12}^\mathrm{u}$ and $\boldsymbol{y}_{21}^\mathrm{u}$ are passed to the discipline-level optimisation, which will pass the optimal f_1^l, f_2^l, $\boldsymbol{x}_\mathrm{c}^\mathrm{l1}$, $\boldsymbol{x}_\mathrm{c}^\mathrm{l2}$, $\boldsymbol{y}_{12}^\mathrm{l}$ and $\boldsymbol{y}_{21}^\mathrm{l}$ to the system-level optimisation. Then, another loop begins. In the study in [14], the penalty function in the system optimisation in ATC was transformed into a multi-objective formulation as shown in Eq. (6), and the non-dominated sorting genetic algorithm-II [40] was applied to solve the multi-objective problem.

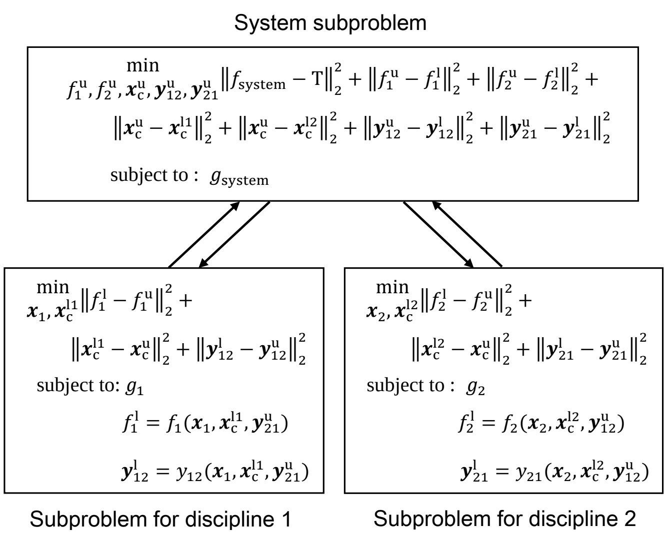

Fig. 3. ATC decomposition and coordination.

$$
\begin{aligned}
&\min \|f_\mathrm{system} - \mathrm{T}\|_2^2 \\
&\min \|f_1^\mathrm{u} - f_1^\mathrm{l}\|_2^2 + \|f_2^\mathrm{u} - f_2^\mathrm{l}\|_2^2 \\
&\min : \\
&\|\boldsymbol{x}_\mathrm{c}^\mathrm{u} - \boldsymbol{x}_\mathrm{c}^\mathrm{l1}\|_2^2 + \|\boldsymbol{x}_\mathrm{c}^\mathrm{u} - \boldsymbol{x}_\mathrm{c}^\mathrm{l2}\|_2^2 + \|\boldsymbol{y}_{12}^\mathrm{u} - \boldsymbol{y}_{12}^\mathrm{l}\|_2^2 + \|\boldsymbol{y}_{21}^\mathrm{u} - \boldsymbol{y}_{21}^\mathrm{l}\|_2^2 \\
&\text{suject to} : g_\mathrm{system}
\end{aligned}
\tag{6}
$$

As mentioned earlier, cooperative co-evolutionary algorithm was employed to solve MDO problems [7], [18], [19], [20]. In these studies, the common variables of an MDO problem are decomposed into disjoint sets and assigned to different disciplines. The coupling variables are handled with duplicate variables or using the implicit iteration strategy [7]. These methods need to make decisions on how the common variables should be decomposed, but such methods have not been developed yet. In [43], the authors suggested decomposing common variables based on their main effects on the disciplinary objectives via orthogonal array-based experimental design before starting optimisation.

III. The Proposed Methods

In this section, two CC-based methods for two kinds of CE problems are introduced, respectively. To begin with, we introduce a new novelty-driven CC method with stochastic selection (NDCC-SS in brief) for the mono-disciplinary design problems, and then a novel co-evolutionary concurrent design method (CCDM in brief) for the quasi-separable MDO problems.

A. A New Novelty-Driven Cooperative Co-evolution with Stochastic Selection (NDCC-SS)

In NDCC-SS, we hope to prevent CC from converging to suboptimal states by maintaining the diversity of the selected individuals in the selection process because the search direction of CC depends on the selected individuals. Motivated by this, we use a new novelty metric in NDCC-SS that measures individuals' novelty score based only on the individuals that are already selected for generating offsprings. Therefore, novelty calculation and EA selection are conducted together in NDCC-SS. For the sake of computational efficiency, the novelty score of one individual, $\boldsymbol{x}$, is evaluated as follows:

$$
nov(\boldsymbol{x}) = dist(\boldsymbol{x}, \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{x}_\mathrm{s}^i) \tag{7}
$$

where $\boldsymbol{x}_\mathrm{s}^i$ denotes the i-th already selected individual and k denotes the number of the individuals already selected. Moreover, the binary tournament selection is employed and Algorithm 2 describes the selection process in our study. Note the cost of the novelty calculation process in Algorithm 2 increases linearly with population size, which has a time complexity of $\mathcal{O}(NP * \mathrm{complexity}(dist))$ (NP denotes population size).

Algorithm 2 Novelty Calculation and Binary Tournament Selection

Select the best individual in the current population as $\boldsymbol{x}_s^1$
Set $\boldsymbol{c} = \boldsymbol{x}_\mathrm{s}^1$ and $k = 2$
while $k \leq population\ size$ **do**
    Randomly pick two individuals from the current population, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$
    Calculate the fitness of them and set the novelty of $\boldsymbol{x}_i (i = 1, 2)$ as $dist(\boldsymbol{x}_i, \boldsymbol{c})$
    Select the winner of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ as $\boldsymbol{x}_\mathrm{s}^k$
    Set $\boldsymbol{c} = (\boldsymbol{c} * (k-1) + \boldsymbol{x}_\mathrm{s}^k)/k$ and $k = k + 1$
end while
Let all $\boldsymbol{x}_\mathrm{s}^i (i = 1, 2, ...k-1)$ enter next generation

In addition, we use a stochastic selection process with an adaptive probability to make a trade-off between novelty and fitness objectives. Algorithm 3 shows the comparison process between two individuals, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The novelty and fitness score of $\boldsymbol{x}_i (i = 1, 2)$ is $nov(\boldsymbol{x}_i)$ and $f(\boldsymbol{x}_i)$, respectively. The parameter p_n denotes the probability of selecting individuals based on their novelty scores, and $\mathrm{rand}(0, 1)$ is a random number generated from a uniform distribution between 0 and 1. The value of p_n in the new method is linearly decreased from p_0 to p_f within a specified number of generations and is kept constant as p_f after that. In detail, p_n^G is calculated as follows:

$$\begin{cases} p_n^G = p_0 - (p_0 - p_f) * \frac{G}{r * MaxGen}, & if\ G \leq r * MaxGen \\ 0, & otherwise \end{cases} \tag{8}$$

where G is the current number of generations, r is a real number ($0 \leq r \leq 1$), and $MaxGen$ denotes the maximum number of generations.

Algorithm 3 Comparison($\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $f(\boldsymbol{x}_1)$, $nov(\boldsymbol{x}_1)$, $f(\boldsymbol{x}_2)$, $nov(\boldsymbol{x}_2)$, p_n)

```
if rand(0,1) < p_n then
    if nov(x_1) > nov(x_2) then
        x_1 wins
    else
        x_2 wins
    end if
else
    if f(x_1) > f(x_2) then
        x_1 wins
    else
        x_2 wins
    end if
end if
```

Through the use of stochastic selection with an adaptive probability, NDCC-SS has two advantages in balancing the novelty and fitness objectives compared with the weighted sum method and multi-objective method. One is that the stochastic selection shows a direct and explicit scheme to determine how many comparisons are dominated by the novelty or fitness objectives, while the importance of novelty or fitness is unclear in the weighted sum method. The other is that our adaptive scheme changes the emphasis on objectives as the evolution proceeds. Through the linear decrease of p_n, it is expected that the new method will focus more on exploration in the early stage of search and on exploitation in the later stage.

For mono-disciplinary design problems, the parallel CC framework in Fig. 1 with the new novelty-driven method is used in our work. Assume a product has two parts ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$). In the beginning, an initial subpopulation for each of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is randomly generated. At each communication step (e.g., t_1, t_2, t_3), two subpopulations communicate with each other the best individual in their current populations. Then, $\boldsymbol{x}_1$ is optimised using an EA for a fixed number of generations (until the next communication step) using the new $\boldsymbol{x}_2$ in fitness evaluation, meanwhile $\boldsymbol{x}_2$ is optimised using an EA with the new $\boldsymbol{x}_1$ in fitness evaluation. In each EA optimisation, individuals in each subpopulation are selected based on either novelty (calculated using Eq. (7)) or fitness scores according to Algorithm 3.

B. A Novel Co-evolutionary Concurrent Design Method (CCDM) for Quasi-Separable MDO

For quasi-separable MDO problems, the original problem is decomposed along disciplines and duplicate variables are introduced to deal with the common design variables $\boldsymbol{z}$ in CCDM. The design is assumed to involve two disciplines and each has local design parameters $\boldsymbol{x}_i (i = 1, 2)$, respectively. Through the use of duplicate variables, $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$, the original problem in Eq. (2) can be decomposed as follows:

1) subproblem 1

$$\begin{aligned} &\min_{\boldsymbol{x}_1, \boldsymbol{z}^1} \ f_1(\boldsymbol{x}_1, \boldsymbol{z}^1) \\ &\text{subject to} : \boldsymbol{z}^1 = \boldsymbol{z}^2 \end{aligned} \tag{9}$$

2) subproblem 2

$$\begin{aligned} &\min_{\boldsymbol{x}_2, \boldsymbol{z}^2} \ f_2(\boldsymbol{x}_2, \boldsymbol{z}^2) \\ &\text{subject to} : \boldsymbol{z}^2 = \boldsymbol{z}^1 \end{aligned} \tag{10}$$

After the decomposition, the CCDM optimises each subproblem in a subpopulation with a specified EA simultaneously, as shown in Fig. 1. At each communication step, they communicate with each other their best design information (i.e., $\boldsymbol{z}^i$ ($i = 1, 2$)) in the current population in order to evaluate their separate designs.

To deal with the equality constraints in Eqs. (9) and (10), CCDM transforms them into an inequality constraint as follows:

$$\frac{\sum_{k=1}^{nc} |z_k^1 - z_k^2|}{nc} \leq \delta \tag{11}$$

where nc is the number of common variables; z_k^1 and z_k^2 are the k-th element of $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, respectively; δ is the tolerance parameter, and an adaptive setting [44] is applied here. In this adaptive setting, δ is adapted by using the following expression:

$$\delta(t+1) = \frac{\delta(t)}{\hat{\delta}} \tag{12}$$

where $\delta(0)$ is set as the median of equality constraint violations (i.e., $\frac{\sum_{k=1}^{nc} |z_k^1 - z_k^2|}{nc}$) over all individuals in the initial population; $\hat{\delta}$ is the rate of decay, which is set in such a way that the δ value will decrease to a final value δ_f after a specified number of generations and will be kept as δ_f after that.

To deal with the generated inequality constraints and other existing inequality constraints, a state-of-the-art constraint handling method, the stochastic ranking method [25] is used. In the stochastic ranking method, when making comparisons among feasible individuals, the individual with better objective value is selected. Otherwise, the comparison is made based on the objective values of individuals with the probability P_f and based on the constraint violation values with the probability

$1 - P_f$. The constraint violation for the inequality constraint in Eq. (11) is defined as:

$$\phi(\boldsymbol{z}) = \begin{cases} 0 & \text{if } \frac{\sum_{k=1}^{nc} |z_k^1 - z_k^2|}{nc} \leq \delta \\ \delta - \frac{\sum_{k=1}^{nc} |z_k^1 - z_k^2|}{nc} & \text{otherwise} \end{cases} \quad (13)$$

In our work, the probability parameter, P_f, is decreased linearly from $P_f = 0.475$ in the initial generation to $P_f = 0.25$ in the final generation. After the optimisation, the mean values of $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$ in the best solutions are used as the final values of $\boldsymbol{z}$ in CCDM.

In both NDCC-SS and CCDM, the communication interval denotes the difference between the generation numbers of two sequential communications in the parallel CC as shown in Fig. 1. There will be more than two subpopulations if more than two parts (disciplines) exist in the design, but the whole framework remains unchanged.

IV. Experimental Studies

A. Case Studies

To evaluate the efficacy of the two proposed methods, universal electric motor (UEM) design problems and a geometric programming problem are used as case studies. The UEM design problem is a widely used benchmark problem in the field of product family design [45], [46], [47]. In this study, two scenarios of the UEM design problems are considered.

1) The Baseline UEM Design Problem: One UEM has 8 design parameters, $\{N_c, N_s, A_{wf}, A_{wa}, I, r_o, t, L\}$ [48]. The description of each parameter can be found in [46]. The design of an electric motor can be formulated as a two-objective constrained optimisation problem, given as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} \{Mass(\boldsymbol{x}), -\eta(\boldsymbol{x})\} \\ &\text{where :} \\ &\boldsymbol{x} = \{N_c, N_s, A_{wf}, A_{wa}, I, r_o, t, L\} \\ &\text{subject to :} \begin{cases} H(\boldsymbol{x}) \leq 5000[\text{Amp} \cdot \text{turns/m}] \\ \frac{r_o}{t} \geq 1 \\ Mass(\boldsymbol{x}) \leq 2[\text{kg}] \\ \eta(\boldsymbol{x}) \geq 15\% \\ P(\boldsymbol{x}) = 300[\text{W}] \\ T(\boldsymbol{x}) = a[\text{Nm}] \end{cases} \end{aligned} \quad (14)$$

where $Mass(\boldsymbol{x})$ and $\eta(\boldsymbol{x})$ denote the weight and efficiency calculation function, respectively. In Eq. (14), a denotes the users' requirement of the torque output ($T(\boldsymbol{x})$), which is a real number and specified a priori. In this paper, for each design $\boldsymbol{x}$, the penalty with respect to the constraints in Eq. (14) is calculated and added to $\{Mass(\boldsymbol{x}), -\eta(\boldsymbol{x})\}$. The resulting $\{Mass(\boldsymbol{x}), -\eta(\boldsymbol{x})\}$ are used as the final objective values of the design. Details about the penalty calculation and the formulations of $H(\boldsymbol{x})$, $Mass(\boldsymbol{x})$, $\eta(\boldsymbol{x})$, $P(\boldsymbol{x})$ and $T(\boldsymbol{x})$ can be found in the Appendix A.

For the UEM design problem, if the preference between $Mass(\boldsymbol{x})$ and $-\eta(\boldsymbol{x})$ is known, a weighted sum of these two objective functions can be defined as the following expression:

$$f_{\text{weighted}}(\boldsymbol{x}) = w_1 * (1 - \eta(\boldsymbol{x})) + w_2 * Mass_{\text{normalised}}(\boldsymbol{x}) \quad (15)$$

where $Mass_{\text{normalised}}$ represents the normalised weight derived by dividing the original weight by the maximum maximum allowable weight: $Mass_{\text{max}} = 2$ [kg]). w_1 and w_2 are weight coefficients that are assumed to be equal (i.e., $w_1 = w_2 = 0.5$) in [49]. Through the weighted sum, the UEM design problem is transformed into a single-objective optimisation problem.

2) Scenario 1: Single Motor Design: This scenario aims to design a motor with respect to one torque requirement a (Eq. (14)). We assume for this scenario that 8 design parameters are decomposed into two groups $\{N_c, A_{wf}, I, t\}$ and $\{N_s, A_{wa}, r_o, L\}$ in the beginning. This scenario is used as a case study for the mono-disciplinary design problems in this paper. For this scenario, the single-objective case is considered. In this case, we use the weighted sum function in Eq. (15) as the goal function, and set both w_1 and w_2 to 0.5. The range for each design parameter of each UEM motor is set the same as in [46], which is given in Table I. Two design problems are considered by using different a values in this paper. One is $a = 0.3$, and the other is $a = 0.5$. When applying the proposed NDCC-SS to this scenario, all groups of the design variables of a motor are optimised simultaneously by different subpopulations.

TABLE I
Range setting for motor parameters

Parameter	Range
N_c	[100, 1500]
N_s	[1, 500]
A_{wf}	[0.01, 1.00] ([mm^2])
A_{wa}	[0.01, 1.00] ([mm^2])
I	[0.1, 6.0] (Amp)
r_o	[1, 10] (cm)
t	[0.5, 100] (mm)
L	[0.1, 20] (cm)

3) Scenario 2: Overlapping Motors Design: In this scenario, two or more overlapping motors need to be designed, each with a different torque requirement (i.e. a in Eq. (14)). Here, overlapping motors refers to different motors sharing some design parameters. This scenario captures some of the important characteristics of designing a product family. In the literature, the authors in [50] considered designing multiple aircrafts, each with a different mission but sharing a common wing. In this scenario, the overall goal is:

$$\begin{aligned} &\min_{\boldsymbol{x},\boldsymbol{z}} \sum_{i=1}^{m} f_{\text{weighted},i}(\boldsymbol{x}_i, \boldsymbol{z}) \\ &\text{i.e.,} \min_{\boldsymbol{x},\boldsymbol{z}} \sum_{i=1}^{m} (w_1 * (1 - \eta(\boldsymbol{x}_i, \boldsymbol{z})) + w_2 * Mass_{\text{normalised}}(\boldsymbol{x}_i, \boldsymbol{z})) \end{aligned} \quad (16)$$

where m is the number of motors, $\boldsymbol{x}_i$ denotes the local design parameters of the i-th motor, and $\boldsymbol{z}$ denotes the common design parameters. In this scenario, we consider designing 2 motors with different a values in Eq. (14) and assume they have 2 parameters (t, L) in common. Three different test problems are studied by using different a values. For the first problem, the a values of the two motors are set to 0.1 and 0.125, respectively. For the second problem, they are set to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2713949, IEEE Transactions on Evolutionary Computation

0.1 and 0.3, respectively. For the third problem, they are 0.05 and 0.5, respectively. This scenario is used as a case study for the quasi-separable MDO problems and the proposed CCDM.

4) A Geometric Programming Problem: The original geometric programming problem [51] is formulated as follows:

$$\min_{\boldsymbol{z}\geq 0} \; z_1^2 + z_2^2$$
$$\text{subject to}: \begin{cases} z_3^{-2} + z_4^2 - z_5^2 \leq 0 \\ z_5^2 + z_6^{-2} - z_7^2 \leq 0 \\ z_8^2 + z_9^2 - z_{11}^2 \leq 0 \\ z_8^{-2} + z_{10}^2 - z_{11}^2 \leq 0 \\ z_{11}^2 + z_{12}^{-2} - z_{13}^2 \leq 0 \\ z_{11}^2 + z_{12}^2 - z_{14}^2 \leq 0 \\ z_1^2 - z_3^2 - z_4^{-2} - z_5^2 = 0 \\ z_2^2 - z_5^2 - z_6^2 - z_7^2 = 0 \\ z_3^2 - z_8^2 - z_9^{-2} - z_{10}^{-2} - z_{11}^2 = 0 \\ z_6^2 - z_{11}^2 - z_{12}^2 - z_{13}^2 - z_{14}^2 = 0 \end{cases} \tag{17}$$

By using the equality constraints, this problem can be transformed into the following formulation:

$$\min_{\boldsymbol{x}_i,\boldsymbol{z}\geq 0} \sum_{i=1}^{2} f_i(\boldsymbol{x}_i, \boldsymbol{z})$$
$$\text{where}:$$
$$\boldsymbol{x}_1 = \{z_4, z_8, z_9, z_{10}\},\; \boldsymbol{x}_2 = \{z_7, z_{12}, z_{13}, z_{14}\},\; \boldsymbol{z} = \{z_5, z_{11}\}$$
$$f_1(\boldsymbol{x}_1, \boldsymbol{z}) = z_4^{-2} + z_5^2 + z_8^2 + z_9^{-2} + z_{10}^{-2} + z_{11}^2$$
$$f_2(\boldsymbol{x}_2, \boldsymbol{z}) = z_5^2 + z_7^2 + z_{11}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2$$
$$\text{subject to}: \begin{cases} (z_8^2 + z_9^{-2} + z_{10}^{-2} + z_{11}^2)^{-1} + z_4^2 - z_5^2 \leq 0 \\ z_5^2 + (z_{11}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2)^{-1} - z_7^2 \leq 0 \\ z_8^2 + z_9^2 - z_{11}^2 \leq 0 \\ z_8^{-2} + z_{10}^2 - z_{11}^2 \leq 0 \\ z_{11}^2 + z_{12}^{-2} - z_{13}^2 \leq 0 \\ z_{11}^2 + z_{12}^2 - z_{14}^2 \leq 0 \end{cases} \tag{18}$$

It can be seen from Eq. (18) that the original problem is transformed into a constrained quasi-separable MDO problem with common variables (z_5, z_{11}). This problem is used as another case study for the quasi-separable MDO problems and the proposed CCDM. It is decomposed into two subproblems with duplicate variables ($(z_{5,1}, z_{5,2})$ and $(z_{11,1}, z_{11,2})$) as follows:

1) subproblem 1

$$\min_{\boldsymbol{x}_1,\boldsymbol{z}\geq 0} \; z_4^{-2} + z_{5,1}^2 + z_8^2 + z_9^{-2} + z_{10}^{-2} + z_{11,1}^2$$
$$\text{subject to}: \begin{cases} (z_8^2 + z_9^{-2} + z_{10}^{-2} + z_{11,1}^2)^{-1} \\ \quad + z_4^2 - z_{5,1}^2 \leq 0 \\ z_8^2 + z_9^2 - z_{11,1}^2 \leq 0 \\ z_8^{-2} + z_{10}^2 - z_{11,1}^2 \leq 0 \\ z_{5,1} - z_{5,2} = 0 \\ z_{11,1} - z_{11,2} = 0 \end{cases} \tag{19}$$

2) subproblem 2

$$\min_{\boldsymbol{x}_2,\boldsymbol{z}\geq 0} \; z_{5,2}^2 + z_7^2 + z_{11,2}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2$$
$$\text{subject to}: \begin{cases} z_{5,2}^2 + (z_{11,2}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2)^{-1} - z_7^2 \leq 0 \\ z_{11,2}^2 + z_{12}^{-2} - z_{13}^2 \leq 0 \\ z_{11,2}^2 + z_{12}^2 - z_{14}^2 \leq 0 \\ z_{5,1} - z_{5,2} = 0 \\ z_{11,1} - z_{11,2} = 0 \end{cases} \tag{20}$$

B. Compared Algorithms

In the experiments, our proposed methods, NDCC-SS and CCDM, are evaluated on the aforementioned case studies, and then comparisons are made between these two methods and some existing methods in terms of the quality of the best solution obtained with a fixed number of function evaluations.

Existing methods used for comparison on Scenario 1 of the UEM design problem include:

- CC: the original CC without considering novelty,
- Novelty-driven CC with the weighted sum method (NDCC-WS) [37],
- Novelty-driven CC with the multi-objective trade-off method (NDCC-MO) [23].

For Scenario 2, the proposed CCDM is compared with the baseline method that optimises all decision variables as a whole without any decomposition (i.e., addressing the optimisation problem in Eq. (16) directly). This is to study whether the decomposition and coordination of the common variables that are used in CCDM are successfully working. We also implemented the COSMOS method (collaborative optimisation strategy for multi-objective systems) [42] that uses nested decomposition on Scenario 2. The resulting method is named collaborative optimisation strategy for single-objective system (COSSOS) in this paper. COSSOS and CCDM are also compared. For the geometric programming problem, the solution quality obtained by CCDM is compared with the result achieved by two existing methods:

- AAO (All-At-Once): it optimises all decision variables at once [52],
- ATC-MO: the ATC method with the MO formulation [23].

For all the experiments, except for AAO and ATC-MO for which the original results in [52] and [23] are used for comparison, we use a real-coded genetic algorithm with binary tournament selection and standard operators for simulated binary crossover and polynomial mutation [53] as the underlying EA. The crossover probability is set to 0.9 and the mutation probability is set to $\frac{1}{n}$ (where n is the number of decision variables). The distribution indices [53] for the crossover and mutation operators are set to 15 and 20, respectively. Note that the same EA is used to allow for a fair comparison.

For NDCC-WS, three different settings of the weight ρ in Eq. (4) are considered: $\rho = 0.25$, $\rho = 0.5$, and $\rho = 0.75$. Two different settings of the archive used in novelty calculation are applied. One is without an archive, the other uses an archive in calculating the novelty and in this setting λ randomly selected individuals in the population are added to the archive in every generation. In the experiments, novelty computation is done with $k = 15$ and $\lambda = 6$ according to [37], [39]. In total, there are 6 different settings of NDCC-WS used for comparison, i.e., NDCC-WS(0.25), NDCC-WS(0.50), NDCC-WS(0.75), NDCC-WS-a(0.25), NDCC-WS-a(0.50), NDCC-WS-a(0.75) ('a' means with an archive).

For NDCC-MO, both versions, with and without an archive, are considered. They are noted as NDCC-MO and NDCC-MO-a, respectively. When implementing NDCC-MO and NDCC-MO-a in the experiments, the binary tournament selection is based on the domination between two individuals. In the selection process, the one that dominates the other is selected. If neither is dominated by the other, one of the two individuals will be chosen randomly.

Table II summarises the compared methods on each test scenario considered in the experimental study. For all methods that use the parallel CC framework, different subpopulations communicate with each other at every generation in the implementation.

TABLE II
A SUMMARY OF COMPARED METHODS ON THE TEST PROBLEMS

Problem	Method
Scenario 1	CC, NDCC-WS(0.25), NDCC-WS(0.50), NDCC-WS(0.75), NDCC-WS-a(0.25), NDCC-WS-a(0.50), NDCC-WS-a(0.75), NDCC-MO, NDCC-MO-a, NDCC-SS
Scenario 2	CCDM, Baseline Method, COSSOS
Geometric programming	CCDM, AAO, ATC-MO

C. Comparison Results

1) Scenario 1: Single Motor Design: For this scenario, the proposed NDCC-SS is compared to CC, NDCC-WS(0.25), NDCC-WS(0.50), NDCC-WS(0.75), NDCC-WS-a(0.25), NDCC-WS-a(0.50), NDCC-WS-a(0.75), NDCC-MO and NDCC-MO-a on the two single motor design problems. For all these 10 methods, the population size for each subpopulation was set to 50, and the maximum number of function evaluations was set to 80000. The parameters of NDCC-SS, p_0, p_f and r, were set to 0.45, 0 and 0.4, respectively. Each method was run for 100 times independently. Then, we used Wilcoxon rank-sum test at a significance level of 0.05 to make comparisons between NDCC-SS and each other method.

Table III gives the means and standard deviations of the best function values over 100 runs obtained by each method as well as the statistical comparison results between NDCC-SS and each of the other methods on the two test problems. It can be seen from Table III that the proposed NDCC-SS performed the best among all algorithms on both test problems. The evolutionary curves of CC, NDCC-WS-a(0.25), NDCC-MO, and NDCC-SS on the single motor design problems with $a = 0.3$ and $a = 0.5$ are given in Figs. 4 and 5, respectively.

TABLE III
COMPARISON BETWEEN CC, NDCC-WS, NDCC-MO AND NDCC-SS FOR SINGLE MOTOR DESIGN PROBLEMS. THE MINUS SIGN ($-$) DENOTES THE METHOD IN THAT ROW IS STATISTICALLY WORSE THAN NDCC-SS ON THE TEST PROBLEM IN THAT COLUMN.

Method	$a = 0.3$	$a = 0.5$
CC	4.90e-01±1.41e-01 $-$	1.87e+01±6.00e+01 $-$
NDCC-WS(0.25)	8.73e-01±2.72e+00 $-$	4.56e+00±1.22e+01 $-$
NDCC-WS(0.50)	6.67e-01±9.93e-01 $-$	4.26e+00±1.37e+01 $-$
NDCC-WS(0.75)	2.93e+00±1.97e+01 $-$	6.35e+00±2.14e+01 $-$
NDCC-WS-a(0.25)	5.08e-01±5.45e-01 $-$	1.00e+00±1.86e+00 $-$
NDCC-WS-a(0.50)	1.14e+00±1.40e+00 $-$	1.56e+00±3.94e+01 $-$
NDCC-WS-a(0.75)	1.20e+01±2.36e+01 $-$	5.97e+01±9.01e+01 $-$
NDCC-MO	5.01e-01±4.96e-02 $-$	1.94e+00±3.55e+00 $-$
NDCC-MO-a	5.35e-01±6.93e-02 $-$	1.92e+00±4.38e+00 $-$
NDCC-SS	4.43e-01±3.59e-02	5.66e-01±3.72e-02

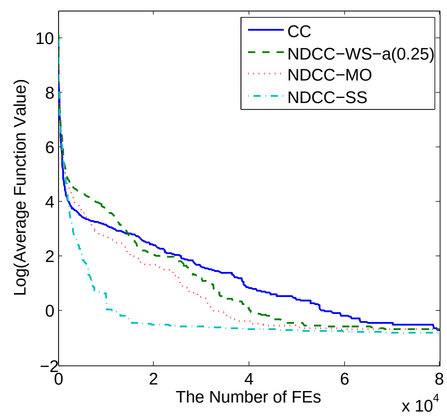

Fig. 4. Evolution Curves of CC, NDCC-WS-a(0.25), NDCC-MO, and NDCC-SS on the single objective case ($a = 0.3$) in Scenario 1.

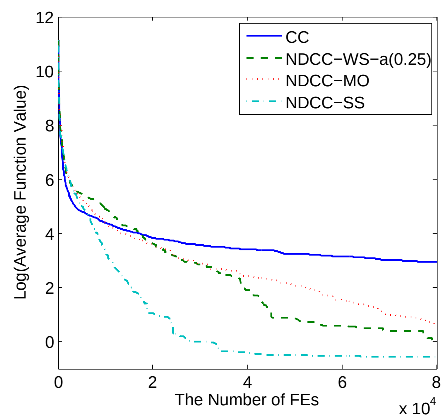

Fig. 5. Evolution Curves of CC, NDCC-WS-a(0.25), NDCC-MO, and NDCC-SS on the single objective case ($a = 0.5$) in Scenario 1.

To investigate whether both the new novelty calculation and the adaptive stochastic selection process contribute to the good performance of NDCC-SS, we furthermore conducted comparative experiments. In the first experiment, we fixed the probability parameter p_n in the stochastic selection in NDCC-SS and conducted experiments on the single motor design problems. Three values for p_n were used, which are 0.25, 0.50, and 0.75. We also replaced the adaptive selection process in NDCC-SS with the multi-objective trade-off method and tested the performance of the resulting method on the single motor design problems. Table IV summarises the statistical comparison results between NDCC-SS with these methods. The minus sign ($-$) in Table IV denotes that the method in that row is statistically worse than NDCC-SS on the test problem in that column. From Table IV, we can observe that NDCC-SS performed better than any other method. This demonstrates the advantage of the use of the adaptive strategy in NDCC-SS.

TABLE IV
COMPARISON BETWEEN NDCC-SS WITH NDCC-SS WITH $p_n = 0.25, 0.50, 0.75$ AND MULTI-OBJECTIVE TRADE-OFF FOR SINGLE MOTOR DESIGN PROBLEMS.

Method	$a = 0.3$	$a = 0.5$
$p_n = 0.25$	5.13e-01±4.45e-02 $-$	6.14e-01±4.64e-02 $-$
$p_n = 0.50$	5.41e-01±4.68e-02 $-$	6.48e-01±5.29e-02 $-$
$p_n = 0.75$	1.54e+00±2.05e+00 $-$	7.73e+00±1.54e+01 $-$
Multi-objective	5.29e-01±4.14e-02 $-$	6.63e-01±1.41e-01 $-$
NDCC-SS	4.43e-01±3.59e-02	5.66e-01±3.72e-02

In the second experiment, we replaced the proposed novelty calculation in NDCC-SS by the novelty calculation proposed in [34] and tested the performance of the resulting method on single motor design problems. In the experiments, we considered two settings of archive in the novelty calculation, with and without an archive, and used four settings of p_0, p_f and r values in the adaptive stochastic selection. Table V summarises the statistical comparison results between NDCC-SS with these methods. The minus sign ($-$) in Table V denotes the method in that row is statistically worse than NDCC-SS on the test problem in that column. It can be seen from Table V that NDCC-SS outperformed the methods that used the existing novelty calculation. This demonstrates that the new novelty calculation strategy has better performance in addition to computational efficiency. This might be because that the new novelty calculation strategy can help maintain better diversity of the selected individuals than the existing novelty calculation strategy since the new novelty calculation is based on only the individuals that are already selected in the selection process of CC while the existing one calculates individuals novelty based on the whole population before selection. As the selected individuals decide the search direction of CC, the new novelty strategy may help preserve more diversity in the population and this is good for preventing CC from converging to suboptimal states.

TABLE V
COMPARISON BETWEEN NDCC-SS WITH THE METHODS THAT USED THE NOVELTY CALCULATION IN [34]

Method Archive/No(p_0,p_f,r)	$a = 0.3$	$a = 0.5$
No(0.5,0.25,0.4)	1.32e+00±4.58e+00 $-$	3.96e+00±8.72e+00 $-$
No(0.5,0.25,0.8)	5.42e-01±3.41e-01 $-$	4.98e+00±1.16e+01 $-$
No(0.25,0,0.4)	5.45e-01±7.14e-01 $-$	1.57e+01±3.18e+01 $-$
No(0.25,0,0.8)	1.36e+00±7.09e+00 $-$	1.34e+01±3.53e+01 $-$
Archive(0.5,0.25,0.4)	4.98e-01±2.22e-01 $-$	1.15e+00±3.01e+00 $-$
Archive(0.5,0.25,0.8)	4.93e-01±1.07e-01 $-$	1.16e+00±2.33e+00 $-$
Archive(0.25,0,0.4)	5.77e-01±6.02e-01 $-$	1.66e+01±3.06e+01 $-$
Archive(0.25,0,0.8)	4.86e-01±1.06e-01 $-$	7.58e+00±1.95e+01 $-$
NDCC-SS	4.43e-01±3.59e-02	5.66e-01±3.72e-02

In short, the additional comparative experimental results above showed the efficacy of both the new novelty calculation and the adaptive stochastic selection in NDCC-SS. This is the reason why NDCC-SS performed well on single motor design problems, as shown in Table III. Note that the new novelty calculation is not the optimum novelty calculation method as it considers computational efficiency. More work is still worth doing along the direction of developing better novelty-driven CC algorithms.

2) Scenario 2: Overlapping Motors Design: For this scenario, the proposed CCDM is compared to the baseline method that puts all design variables in a single optimisation process on the three overlapping motor design problems. For the baseline method, the population size was set to 100 in the experiments. For CCDM, the population size was set to 50. The maximum number of function evaluations was set to 32000 for the baseline method. While the baseline method needs to evaluate both f_1 and f_2 in a function evaluation on the overlapping motor design problems with two motors, CCDM only needs to evaluate one $f_i (i = 1, 2)$. Therefore, the maximum number of function evaluations was set to 32000*2 for CCDM. The parameter δ_f in CCDM was set to 0.005, and the ratio of the specified number of generations to the maximum number of generations was set to 0.8.

We also compare CCDM with the COSSOS method. For the COSSOS method, the population size for the supervisor-level optimisation was set to 20, and for each disciplinary-level optimisation, it was set to 50. As the COSSOS method also needs to evaluate only one $f_i (i = 1, 2)$ in a function evaluation in one disciplinary-level optimisation, the maximum number of function evaluations was set to 32000*2 for COSSOS. Besides, the maximum number of function evaluations for each disciplinary-level optimisation in COSSOS was set to 800 in our experiments. We independently ran each experiment for 100 times, and then used Wilcoxon rank-sum test at a significance level of 0.05 to conduct comparisons.

Table VI presents the means and standard deviations of the best function values over 100 runs obtained by the three methods. The statistical comparison results between the two compared methods and CCDM are also given in this table. It can be seen from this table that both decomposition-based methods (i.e. CCDM and COSSOS) outperformed the baseline method. Between the two decomposition methods, the proposed CCDM achieved better solutions on all test problems.

3) The Geometric Programming Problem: For the geometric programming problem, the best solutions obtained by the all-at-once method (AAO) in [52] and the ATC with multi-objective formulation (ATC-MO) in [14] are: (z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}, z_{11}, z_{12}, z_{13}, z_{14})=(2.84, 3.09, 2.36, 0.76, 0.87, 2.81, 0.94, 0.97, 0.87, 0.8, 1.3, 0.84, 1.76, 1.55) and (2.77, 3.14, 2.28, 0.76, 0.88, 2.86, 0.94, 0.96, 0.95, 0.85, 1.35, 0.84, 1.79, 1.58), respectively. Table VII shows the corresponding f values and the constraint violation values of g and h constraints for these two solutions. It can be seen that both AAO and ATC-MO are unable to satisfy all constraints and the maximum violation tolerance for each constraint is 0.055 (see $h3$ in the column of AAO). In our experiments,

TABLE VI
COMPARISON BETWEEN THE BASELINE METHOD, COSSOS AND CCDM FOR OVERLAPPING MOTORS DESIGN. THE MINUS SIGN ($-$) AND THE APPROXIMATION SIGN ($\approx$) DENOTES THE COMPARED METHOD IS STATISTICALLY WORSE THAN, AND SIMILAR TO CCDM, RESPECTIVELY.

a values	Baseline Method	COSSOS method	CCDM
$a_1 = 0.10, a_2 = 0.125$	3.69e+00±1.61e+01 $-$	2.13e+00±6.63e+00 $-$	6.87e-01±3.42e-01
$a_1 = 0.10, a_2 = 0.30$	4.30e+00±1.66e+01 $-$	3.68e+00±6.81e+00 $-$	2.47e+00±1.48e+01
$a_1 = 0.05, a_2 = 0.50$	1.57e+01±3.46e+01 $\approx$	1.51e+01±2.67e+01 $-$	1.51e+01±1.02e+02

CCDM was implemented with the same setting (0.055) for the maximum constraint violation tolerance. The population size for CCDM was set to 100, and the maximum number of function evaluations was set to 10000. The parameter δ_f was set to 0.01 and the ratio of the specified number of generations to the maximum number of generations was set to 0.3 for CCDM.

The best solution achieved by CCDM over 100 runs on this test problem was recorded. It is: (z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}, z_{11}, z_{12}, z_{13}, z_{14})=(2.88, 3.00, 2.40, 0.73, 0.81, 2.77, 0.86, 0.99, 0.81, 0.76, 1.26, 0.97, 1.62, 1.59). Its f value and the constraint violation values of g and h constraints are given in Table VII. It can be seen that CCDM found a solution with better f value compared with AAO and ATC-MO.

TABLE VII
COMPARISON BETWEEN CCDM, AAO AND ATC WITH THE MO FORMULATION ON THE GEOMETRIC PROGRAMMING PROBLEM

	AAO [52]	ATC-MO [14]	CCDM
f	17.6137	17.5325	17.3364
g1	0.0002	0	0.0505
g2	0	0.0131	0.0473
g3	0.0078	0.0016	0.0393
g4	0.0128	0	0.0462
g5	0.0096	0.0356	0.0318
g6	0	0.0317	0.0193
h1	0.0078	0.0312	0
h2	0.0115	0.0220	0
h3	0.0550	0.0378	0
h4	0.0004	0.0490	0

D. The Effect of Communication Frequency

In this section, the best communication frequency among subpopulations in the proposed methods, with and without communication costs, is studied computationally in the two scenarios of motor design problems and the geometric programming problem.

1) Best Communication Frequency without Communication Cost: In the first part, we study in both NDCC-SS and CCDM the best communication frequencies that obtain solutions of highest quality within a fixed number of function evaluations without considering communication costs. We ran NDCC-SS and CCDM on the test problems with each communication interval independently for 100 times and recorded the function values of the best solutions obtained using each communication interval over 100 runs.

As we wish to know which communication frequencies lead to significantly good performance, the Kruskal-Wallis test and a post-hoc test proposed by Conover and Iman [54], [55] were applied with significance level at 0.05. The Kruskal-Wallis test is a nonparametric multiple comparison test that compares more than two populations based on random samples by using rank sums. In our study, the Kruskal-Wallis was applied on each test problem. The function value of the best solution achieved in one run was considered as a sample, and the 100 samples obtained using one communication frequency over 100 runs were considered as a population. The null hypothesis (H0) we used in the Kruskal-Wallis test states that all communication frequencies achieve the equivalent performance. If the null hypothesis is rejected, the communication frequency with the minimum rank sum will be selected as the control frequency, and the post-hoc test will be used to perform a comparison between the control frequency and every other communication frequency. To control the family-wise error, we further applied Finner's adjustment of p-values [56] in this multiple hypothesis testing. Any communication frequency whose performance is not significantly different from the control frequency will be recorded as the best communication frequency along with the control frequency.

The parameter settings of NDCC-SS and CCDM are the same as those in Sections IV.C.1-IV.C.3 except for the setting of communication frequencies. In this study, we considered 12 different communication intervals for the two scenarios of motor design problems and 12 different communication intervals for the geometric programming problem we used. They are $\{1, 2, 4, 5, 8, 10, 16, 20, 32, 40, 80, 160\}$ and $\{1, 2, 4, 5, 8, 10, 20, 25, 40, 50, 80, 100\}$, respectively. Here, each number denotes the difference between the generation numbers of two sequential communication in the co-evolutionary process.

Table VIII gives the best communication intervals for NDCC-SS on the single motor design problems. It shows that it is best to exchange information among subpopulations every 1 or 2 generations when using NDCC-SS to solve the single motor design problems.

TABLE VIII
BEST COMMUNICATION INTERVALS OF NDCC-SS ON SCENARIO 1 WITHOUT COMMUNICATION COST

a value	Best communication intervals
$a = 0.3$	1,2
$a = 0.5$	1,2

Table IX presents the best communication intervals for CCDM on the overlapping motor design problems and the geometric programming problem. As it can be seen in Table IX, it is best to make subpopulations communicate with each other every generation on the first design problem, every 1 or 2 generations on the second and third problems. For the geometric programming problem, all communication intervals performed comparatively well.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2713949, IEEE Transactions on Evolutionary Computation

TABLE IX
BEST COMMUNICATION INTERVALS OF CCDM ON SCENARIO 2 AND THE GEOMETRIC PROGRAMMING PROBLEM WITHOUT COMMUNICATION COST

Problem	Best communication intervals
$a_1 = 0.10, a_2 = 0.125$	1
$a_1 = 0.10, a_2 = 0.3$	1,2
$a_1 = 0.05, a_2 = 0.50$	1,2
Geometric programming	1,2,4,5,8,10,20,25,40,50,80,100

Our general conclusion based on Tables VIII and IX is that subpopulations should communicate with each other as frequently as possible when applying the proposed methods to solve the test design problems without considering the communication cost.

2) Best Communication Frequency with Communication Cost: In the second part, we investigate the best communication frequency considering different communication costs. The communication cost is considered as a fixed number of function evaluations in our study. We varied the communication cost from $0 * NP$ to $40 * NP$ (NP denotes population size) function evaluations for all test problems, and carried out experiments using each communication interval from 1 to 20 generations for every communication cost. The other experimental settings are the same as in Section IV.D.1. We recorded the function values of the best solutions achieved by every communication interval over 100 runs. Then, the Kruskal-Wallis test with Conover's and Iman's post-hoc test and Finner's adjustment of p-values was used to give the best-performing communication frequency.

Tables X and XI summarise the best communication intervals for NDCC-SS on the single motor design problems, and CCDM on the overlapping motor design problems and the geometric programming problem, respectively. A general observation from Tables X and XI is that the best communication frequency varies as the design problem and the communication cost vary.

When comparing Tables X-XI with Tables VIII-IX, we can see that the advantage of frequent communication was affected by the existence of communication cost. That is, one-generation communication is no longer always the best option when the communication cost is considered. This is because that more frequent communication will cause more time to be spent on communications rather than on optimisation. However, the situation is different for each problem. It can be seen from Table X that one generation is always among the best communication intervals for the design problem with $a = 0.5$ even when the communication cost is increased to $40 * NP$ function evaluations. The reason might be that the correlation between the decomposed subcomponents makes the optimisation pay to communicate more frequently.

Comparing each column in Tables X-XI, we can observe that larger communication intervals are more likely to be better when the communication cost becomes larger. Note that sometimes there is the trend that higher communication frequencies become better again as communication cost becomes larger (e.g. one-generation communication is no longer the best at the cost of $16*NP$ but becomes best again at the cost of $18*NP$ in Table X where $a = 0.3$). As we use a fixed maximum number

TABLE X
BEST COMMUNICATION INTERVALS OF NDCC-SS ON SCENARIO 1 WITH DIFFERENT COMMUNICATION COSTS

Communication cost	$a = 0.3$	$a = 0.5$
$2 * NP$	2-8	1-4
$4 * NP$	1-20	1-8,10-11
$6 * NP$	1-20	1-20
$8 * NP$	1-20	1-20
$10 * NP$	3-20	1-20
$12 * NP$	1-20	1-20
$14 * NP$	2,5-20	1-20
$16 * NP$	4-20	1-20
$18 * NP$	1-20	1-20
$20 * NP$	3-20	1-20
$40 * NP$	1-20	1-20

TABLE XI
BEST COMMUNICATION INTERVALS OF CCDM ON SCENARIO 2 AND THE GEOMETRIC PROGRAMMING PROBLEM WITH DIFFERENT COMMUNICATION COSTS

Cost	$a_1 = 0.10$ $a_2 = 0.125$	$a_1 = 0.10$ $a_2 = 0.30$	$a_1 = 0.05$ $a_2 = 0.50$	Geometric programming
$2 * NP$	2-5	2-5	1-5	1-20
$4 * NP$	2-7	2-7	4-5	1-20
$6 * NP$	3-7	3-7	3-5	1-20
$8 * NP$	4-8	2-8	2,4-9	1-20
$10 * NP$	2-9,11	2-8	3-7,9	1-20
$12 * NP$	3-11	3-6	4-8,10	1-20
$14 * NP$	4-6, 9-10	2-7, 9-10	3-8, 12-13,20	1-20
$16 * NP$	3-9,18,20	5,7,8,19,20	3-11,16,18-20	1-20
$18 * NP$	8,17,20	3,5,20	3-5,7-10, 12,14-20	1-20
$20 * NP$	5-9,12,14, 16-17,20	3-8,10,12, 13, 15-20	4-7,12, 14,16-20	1-20
$40 * NP$	18-20	17-20	16-20	3-6, 8-20

of function evaluations for different communication costs, the available numbers of function evaluations to do optimisation is different for different communication costs. Furthermore, the difference of the available numbers of function evaluations between different communication intervals becomes smaller as the communication cost increases. That means, the effect of communication cost on frequent communication becomes smaller as the communication cost increases. Consequently, frequent communication can become better again as communication cost increases.

V. COMMUNICATION FREQUENCY SELF-ADAPTATION

As the best communication frequencies for both proposed methods vary with problem and communication cost, it is not a trivial task to find a good communication frequency for a new problem. Motivated by this, we propose a communication frequency self-adaptation method (CFS) to automatically adjust a good communication frequency during the concurrent design process of NDCC-SS and CCDM.

A. The Proposed Self-Adaptive Method

The CFS method is inspired by the adaptation method proposed in [57], which is used to adapt the population size of an EA. In the beginning, the CFS method randomly initializes a communication interval value p_0. Then, two neighbors

$\{p_1, p_2\}$ in different directions are generated for p_0. That is, $p_1 < p_0 < p_2$. The whole design process is divided into cycles in CFS. In each cycle, each of $\{p_0,p_1,p_2\}$ is used as the communication interval for a specified number of function evaluations. In this process, when p_i $(i = 0, 1, 2)$ is in turn, subpopulations in both NDCC-SS and CCDM communicate with each other every p_i generations, and then the obtained improvement is calculated and recorded. After this process, the value of p_0 is set to the p_i $(i = 0, 1, 2)$ that obtained the largest improvement. Then, two new neighbors $\{p_1, p_2\}$ in different directions are generated for p_0 and the next cycle begins (see Appendix B). The update of p_0 and the re-generation of two neighbors $\{p_1, p_2\}$ for the updated p_0 are based on the following greedy rules (for clarity, the new value of p_i $(i = 0, 1, 2)$ is denoted by $p_i^{'}$ $(i = 0, 1, 2)$):

- If p_1 achieves the largest improvement, then let $p_0^{'} = p_1$, $p_1^{'} = p_1/2$, and $p_2^{'} = (p_1 + p_0)/2$.
- If p_2 achieves the largest improvement, then let $p_0^{'} = p_2$, $p_1^{'} = (p_0 + p_2)/2$, and $p_2^{'} = p_2 * 2$.
- If p_0 has the largest improvement, then let $p_0^{'} = p_0$, $p_1^{'} = (p_0 + p_1)/2$, and $p_2^{'} = (p_0 + p_2)/2$.
- If all p_i (i=1,2,3) achieve the same improvement, then let $p_0^{'} = p_0$, $p_1^{'} = p_0 - 2 * (p_0 - p_1) = 2 * p_1 - p_0$, and $p_2^{'} = p_0 + 2 * (p_2 - p_0) = 2 * p_2 - p_0$. This is to enlarge the search region.
- If no improvement is achieved by any p_i (i=1,2,3), then let $p_0^{'} = p_1/2$, $p_1^{'} = p0^{'}/2$, and $p_2^{'} = (p_0^{'}+p_1)/2$ to make subpopulations communicate more frequently to obtain improvement.

Note that each of $p_1^{'}$, $p_0^{'}$, and $p_2^{'}$ needs to be kept as an integer and the relationship of $p_1^{'} < p_0^{'} < p_2^{'}$ needs to be maintained during this process.

The degree of improvement of the performance is measured by the ratio of the improvement over either the best objective function value or constraint violation value. Furthermore, the improvement is normalised according to the efforts (the number of function evaluations) spent on it. For NDCC-SS on the single motor design problem, the improvement for each p_i $(i = 0, 1, 2)$ is defined as follows:

$$I = (f_{\text{best}}^{t} - f_{\text{best}}^{t+1})/(f_{\text{best}}^{t} * (Evals_{t+1} - Evals_t)) \tag{21}$$

where$f_{\text{best}}^{j} (j = t, t + 1)$ is the best objective function value at time j, and $Evals_j$ is the number of function evaluations spent until time j. For CCDM on the overlapping motor design problems, if the best solution satisfies the constraints, the improvement calculation is the same as in Eq. (21). Otherwise, the improvement for each p_i $(i = 0, 1, 2)$ is calculated as the ratio of improvement on the constraint violation value normalised by the used number of function evaluations.

Note that it is not easy to determine how many function evaluations to run for each p_i in one cycle. First, we cannot have a fixed value for all p_i values as it needs to be a multiple of all $(p_i * pop_{\text{num}} * NP + cost)$ where pop_{num} is the number of subpopulations, NP is the size of subpopulations, and cost is the communication cost. Otherwise, a fixed p_i value would be too large and thus, the number of cycles would be too small to do an effective self-adaptation. Second, if we cannot have a fixed number of function evaluations for all p_i, this method would have some biases. For example, if p_0 and p_1 always get less function evaluations than p_2 does in every cycle, the method might perform badly on problems that prefer small communication intervals. To minimise the bias, we run p_2 for $(p_2 * pop_{\text{num}} * NP + cost)$ function evaluations, and run each p_i $(i = 0, 1)$ for a randomly selected number of function evaluations from $\lfloor (p_2 * pop_{\text{num}} * NP + cost)/(p_i * pop_{\text{num}} * NP + cost) \rfloor * (p_i * pop_{\text{num}} * NP + cost)$ and $\lceil (p_2 * pop_{\text{num}} * NP + cost)/(p_i * pop_{\text{num}} * NP + cost) \rceil * (p_i * pop_{\text{num}} * NP + cost)$ in each cycle.

B. Experimental Results

In this section, we assume a fixed communication cost and study the performance of the proposed CFS method by comparing it with a random method and a method of constant communication frequency in both NDCC-SS and CCDM. In the random method, at each communication step, it randomly decides the next communication time from the specified communication interval range. In fixed communication method, a fixed communication interval of 1 generation is used.

In the experiments, we set the communication cost to $pop_{num} * NP$ and the communication interval range to [1,50]. The maximum number of function evaluations ($MaxFEs$) was set to 2e5, and the specified number of generations in both NDCC-SS and CCDM was set to $\lfloor MaxFEs/(pop_{\text{num}} * NP + cost) \rfloor$. Other settings are the same as in Sections IV.C.1 and IV.C.2.

Tables XII and XIII summarise the means and standard deviations of the function values of the best solutions obtained over 100 runs by using each of the random method, the fixed method and the adaptive method in NDCC-SS and CCDM. We used the Wilcoxon rank-sum test at a significance level of 0.05 to make comparisons between the CFS method and the two compared methods over each test problem, and the statistical test results are also given in Tables XII and XIII. The signs '$\approx$' and '$-$' in Tables XII and XIII denote the compared method is similar to and significantly worse than the proposed CFS method, respectively.

According to the statistical test results, the CFS method performed similarly to the random method on the geometric programming problem in CCDM, and performed better than it on the single motor design problems in NDCC-SS and on the overlapping motors design problems in CCDM. When compared to the fixed method, the CFS method performed better on one test problem and similarly on the other test problem in NDCC-SS. When applied in CCDM, the CFS method and the fixed method obtained similar results on the geometric programming problem. However, the CFS method performed better than the fixed method on the other test problems. According to these comparison results, we consider the CFS method is clearly an effective method for both NDCC-SS and CCDM when information about how to set the communication frequency is unavailable beforehand. The results also show that all of the random method, the fixed method, and the CFS method performed similarly on the geometric programming problem. This is because the use of

TABLE XII
COMPARISON BETWEEN THE CFS METHOD AND THE COMPARED METHODS FOR NDCC-SS ON SCENARIO 1

Problem	Random method	Fixed method	CFS method
$a = 0.3$	1.34e+00±5.71e+00 −	4.87e-01±4.13e-02 −	4.54e-01±4.77e-02
$a = 0.5$	1.65e+01±3.62e+01 −	5.80e-01±3.94e-02 ≈	2.97e+00±1.33e+01

TABLE XIII
COMPARISON BETWEEN THE CFS METHOD AND THE COMPARED METHODS FOR CCDM ON SCENARIO 2 AND THE GEOMETRIC PROGRAMMING PROBLEM

Problem	Random method	Fixed method	CFS method
$a_1 = 0.10, a_2 = 0.125$	1.47e+00±3.83e+00 −	1.02e+00±2.04e+00 −	1.01e+00±1.89e+00
$a_1 = 0.10, a_2 = 0.3$	4.09e+00±1.42e+01 −	1.77e+01±1.17e+02 −	9.99e-01±2.38e+00
$a_1 = 0.05, a_2 = 0.5$	3.10e+01±2.59e+02 −	1.26e+01±6.42e+01 −	2.36e+00±1.33e+01
Geometric programming	4.20e+01±2.43e+01 ≈	4.97e+01±2.25e+01 ≈	4.78e+01±2.54e+01

different communication intervals led to similar results on this problem. The CFS method cannot bring performance improvement on such problems.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed two co-evolutionary concurrent design methods, NDCC-SS and CCDM, based on the parallel CC framework to solve two different kinds of design problems in CE. The NDCC-SS method was developed for design problems with a complex structure. In NDCC-SS, a computationally efficient novelty calculation was used and a stochastic selection process with an adaptive probability was used to adjust a good trade-off between novelty and fitness. In addition, we evaluated the efficacy of NDCC-SS on UEM design problems. The experimental results have shown that the proposed NDCC-SS obtained better designs than normal CC and existing novelty-driven CC methods. To solve quasi-separable MDO problems, duplicate variables and the stochastic ranking method were used in CCDM to deal with the common variables. The superior performance of CCDM has been demonstrated through the experimental comparison with other MDO methods on the UEM design problems and a general MDO problem.

One of the key issues in CE and CC is the frequency of communications among sub-components or subpopulations. Infrequent communications help to increase parallelism and thus computational efficiency, but can lead to optimisation in a subpopulation based on out-of-date information from other subpopulations. Frequent communications help to ensure that the latest information from other subpopulations is used in optimisation, but slow down the parallel CC and thus the computational efficiency. The best communication frequency in each proposed method was studied in this paper. By using different problems and communication costs, the experimental results showed that the best frequency varies as the problem and communication cost change. Motivated by this, a self-adaptive method was developed to adjust a good communication frequency automatically during the concurrent design process. The experimental studies showed that the proposed strategy performed equally good as or better than the random method and the method with a fixed communication interval of one generation.

In future work, the proposed methods will be tested on more benchmark and real-world problems. So far, this paper considered only single objective design optimisation problems. In reality, however, many CE problems are multi-objective design optimisation problems. When developing methods for multi-objective design problems, a different novelty calculation method and common variable handling technique might be required. This is another direction of our future work.

ACKNOWLEDGEMENTS

This work was supported by the Honda Research Institute Europe (HRI-EU), EPSRC (Grant No. EP/K001523/1), and NSFC (Grant No. 61329302). Xin Yao was also supported by a Royal Society Wolfson Research Merit Award. The authors would like to thank Timo Friedrich for supporting the set-up of the universal electric motor design problem. The authors thank Ms. Shuyi Zhang and Dr. Sebastian Schmitt who assisted in the proof-reading of the manuscript. The authors also wish to thank the reviewers for the helpful comments and suggestions.

REFERENCES

[1] A. A. Yassine, R. S. Sreenivas, and J. Zhu, "Managing the exchange of information in product development," *European Journal of Operational Research*, vol. 184, no. 1, pp. 311–326, 2008.

[2] R. Addo-Tenkorang, "Concurrent engineering: A review literature report," in *Proceedings of the World Congress on Engineering and Computer Science*. IEEE, 2011, pp. 1074–1080.

[3] A. Kusiak, *Concurrent engineering: automation, tools, and techniques*. John Wiley & Sons, 1993.

[4] A. Yassine and D. Braha, "Complex concurrent engineering and the design structure matrix method," *Concurrent Engineering*, vol. 11, no. 3, pp. 165–176, 2003.

[5] J. R. Martins and A. B. Lambe, "Multidisciplinary design optimization: a survey of architectures," *American Institute of Aeronautics and Astronautics (AIAA) Journal*, vol. 51, no. 9, pp. 2049–2075, 2013.

[6] N. Kang, M. Kokkolaras, and P. Y. Papalambros, "Solving multiobjective optimization problems using quasi-separable mdo formulations and analytical target cascading," *Structural and Multidisciplinary Optimization*, vol. 50, no. 5, pp. 849–859, 2014.

[7] P. B. Nair and A. J. Keane, "Coevolutionary genetic adaptation–a new paradigm for distributed multidisciplinary design optimization," in *The 40th AIAA/ASME/ASCE/AHS/ASC Structures Structural Dynamics and Materials Conf., St. Louis, MO, AIAA Paper*, no. 1999-1428. Citeseer, 1999.

[8] A. Kusiak and N. Larson, "Decomposition and representation methods in mechanical design," *Journal of Vibration and Acoustics*, vol. 117, no. B, pp. 17–24, 1995.

[9] R. D. Braun, "Collaborative optimization: an architecture for large-scale distributed design," Ph.D. dissertation, Stanford University, 1996.

[10] S. Tosserams, L. Etman, and J. Rooda, "Augmented lagrangian coordination for distributed optimal design in mdo," *International Journal for Numerical Methods in Engineering*, vol. 73, no. 13, pp. 1885–1910, 2008.

[11] J. Sobieszczanski-Sobieski, J. S. Agte, and R. R. Sandusky, "Bilevel integrated system synthesis," *American Institute of Aeronautics and Astronautics (AIAA) Journal*, vol. 38, no. 1, pp. 164–172, 2000.

[12] P. M. Zadeh, V. V. Toropov, and A. S. Wood, "Metamodel-based collaborative optimization framework," *Structural and Multidisciplinary Optimization*, vol. 38, no. 2, pp. 103–115, 2009.

[13] P. Dépincé, B. Guédas, and J. Picard, "Multidisciplinary and multi-objective optimization: Comparison of several methods," in *7th World Congress on Structural and Multidisciplinary Optimization*, 2007.

[14] F. Moussouni, S. Kreuawan, S. Brisset, F. Gillon, P. Brochet, and L. Nicod, "Analytical target cascading for optimal design of railway traction system," in *International Conference on Engineering Optimization*, 2008, pp. 1–10.

[15] X. Zhang, H.-Z. Huang, Z. Wang, Y. Liu, and Y. Li, "A pareto set coordination method for analytical target cascading," *Concurrent Engineering*, p. 1063293X13499358, 2013.

[16] M. A. Potter and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evolutionary computation*, vol. 8, no. 1, pp. 1–29, 2000.

[17] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.

[18] Q. Chen and J. Dai, "Distributed coevolutionary multidisciplinary design optimization: A flexible approach," in *The 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 4-6 September 2002, Altanta, Georgia*, 2002.

[19] ——, "A distributed multi-objective evolutionary algorithm and application in missile design," in *The 2002 International Conference on Control and Automation (ICCA 2002)*. IEEE, 2002, pp. 197–197.

[20] Y. Xing and S. Tang, "A distributed coevolutionary multidisciplinary design optimization algorithm," in *The 2010 International Joint Conference on Computational Science and Optimization (CSO)*, vol. 2. IEEE, 2010, pp. 77–80.

[21] S. Tosserams, L. P. Etman, and J. Rooda, "An augmented lagrangian decomposition method for quasi-separable problems in mdo," *Structural and Multidisciplinary Optimization*, vol. 34, no. 3, pp. 211–227, 2007.

[22] L. Panait, "Theoretical convergence guarantees for cooperative coevolutionary algorithms," *Evolutionary Computation*, vol. 18, no. 4, pp. 581–615, 2010.

[23] J. Gomes, P. Mariano, and A. L. Christensen, "Novelty-driven cooperative coevolution," *Evolutionary Computation*, 2015.

[24] ——, "Avoiding convergence in cooperative coevolution with novelty search," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 1149–1156.

[25] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on evolutionary computation*, vol. 4, no. 3, pp. 284–294, 2000.

[26] V. Krishnan, S. D. Eppinger, and D. E. Whitney, "A model-based framework to overlap product development activities," *Management Science*, vol. 43, no. 4, pp. 437–451, 1997.

[27] T. A. Roemer, R. Ahmadi, and R. H. Wang, "Time-cost trade-offs in overlapped product development," *Operations Research*, vol. 48, no. 6, pp. 858–865, 2000.

[28] C. H. Loch and C. Terwiesch, "Communication and uncertainty in concurrent engineering," *Management Science*, vol. 44, no. 8, pp. 1032–1048, 1998.

[29] J. Lin, Y. Qian, W. Cui, and Z. Miao, "Overlapping and communication policies in product development," *European Journal of Operational Research*, vol. 201, no. 3, pp. 737–750, 2010.

[30] C. Stoen, "Various collaborator selection pressures for cooperative coevolution for classification," in *International Conference of Artificial Intelligence and Digital Communications (AIDC)*, 2006.

[31] A. Bucci and J. B. Pollack, "On identifying global optima in cooperative coevolution," in *Proceedings of the 7th Annual Conference on Genetic and evolutionary computation (GECCO)*. ACM, 2005, pp. 539–544.

[32] L. Panait, R. P. Wiegand, and S. Luke, "A visual demonstration of convergence properties of cooperative coevolution," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 892–901.

[33] J. Lehman and K. O. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary Computation*, vol. 19, no. 2, pp. 189–223, 2011.

[34] ——, "Exploiting open-endedness to solve problems through the search for novelty," in *ALIFE*, 2008, pp. 329–336.

[35] A. Liapis, G. N. Yannakakis, and J. Togelius, "Constrained novelty search: A study on game content generation," *Evolutionary Computation*, vol. 23, no. 1, pp. 101–129, 2015.

[36] J. Lehman and K. O. Stanley, "Efficiently evolving programs through the search for novelty," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2010, pp. 837–844.

[37] G. Cuccu and F. Gomez, "When novelty is not enough," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2011, pp. 234–243.

[38] J.-B. Mouret, "Novelty-based multiobjectivization," in *New Horizons in Evolutionary Robotics*. Springer, 2011, pp. 139–154.

[39] J. Gomes, P. Mariano, and A. L. Christensen, "Devising effective novelty search algorithms: a comprehensive empirical study," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM, 2015, pp. 943–950.

[40] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[41] A. Giassi, F. Bennis, and J.-J. Maisonneuve, "Multidisciplinary design optimisation and robust design approaches applied to concurrent design," *Structural and Multidisciplinary Optimization*, vol. 28, no. 5, pp. 356–371, 2004.

[42] S. Rabeau, P. Dépincé, and F. Bennis, "Collaborative optimization of complex systems: a multidisciplinary approach," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 1, no. 4, pp. 209–218, 2007.

[43] P. B. Nair and A. J. Keane, "Coevolutionary architecture for distributed optimization of complex coupled systems," *American Institute of Aeronautics and Astronautics (AIAA) Journal*, vol. 40, no. 7, pp. 1434–1443, 2002.

[44] R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561–579, 2010.

[45] T. W. Simpson, "Product platform design and customization: status and promise," *AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 18, no. 01, pp. 3–20, 2004.

[46] T. W. Simpson, J. R. Maier, and F. Mistree, "Product platform design: method and application," *Research in Engineering Design*, vol. 13, no. 1, pp. 2–22, 2001.

[47] A. Khajavirad, J. J. Michalek, and T. W. Simpson, "An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality," *Structural and Multidisciplinary Optimization*, vol. 39, no. 2, pp. 187–201, 2009.

[48] T. Friedrich and S. Menzel, "A cascaded evolutionary multi-objective optimization for solving the unbiased universal electric motor family problem," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC'14)*. IEEE, 2014, pp. 3184–3191.

[49] S. V. Akundi, T. W. Simpson, and P. M. Reed, "Multi-objective design optimization for product platform and product family design using genetic algorithms," in *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2005, pp. 999–1008.

[50] K. Willcox and S. Wakayama, "Simultaneous optimization of a multiple-aircraft family," *Journal of Aircraft*, vol. 40, no. 4, pp. 616–622, 2003.

[51] C.-S. Beightler, D.-T. Phillips, R.-S. Dembo, G.-V. Reklaitis, and R.-E.-D. Woolsey, *Applied geometric programming*. John Wiley & Sons, Inc., Distributed by Instrument Society of America, 1975.

[52] H. M. Kim, N. F. Michelena, P. Y. Papalambros, and T. Jiang, "Target cascading in optimal system design," *Journal of Mechanical Design*, vol. 125, no. 3, pp. 474–480, 2003.

[53] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 3, pp. 1–15, 1994.

[54] W. J. Conover and R. L. Iman, "On multiple-comparisons procedures. informal report," Los Alamos Scientific Laboratory., Tech. Rep., 1979.

[55] T. Pohlert, "The pairwise multiple comparison of mean ranks package (pmcmr)," *R package*, pp. 2004–2006, 2014.

[56] H. Finner, "On a monotonicity problem in step-down multiple test procedures," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 920–923, 1993.

[57] R. Hinterding, Z. Michalewicz, and T. C. Peachey, "Self-adaptive genetic algorithm for numeric functions," in *Proceedings of the 4th Conference on Parallel Problem Solving from Nature–PPSN IV*. Springer, 1996, pp. 420–429.