A High-precision Decomposition Strategy for Large-scale Constrained Problems

Fabiola Paz[1], Guillermo Leguizamon[2], and Efren Mezura-Montes[3]

[1] Universidad Nacional de Jujuy, Argentina
fabypaz@fi.unju.edu.ar
[2] LIDIC, Universidad Nacional de San Luis, Argentina
legui@unsl.edu.ar
[3] Artificial Intelligence Research Center, University of Veracruz, Mexico
emezura@uv.mx

Abstract. Decomposition strategies, such as cooperative coevolutionary (CC) algorithms, have proven effective in solving various large-scale constrained optimization problems. This approach decomposes a large-scale constrained problem into simpler, smaller-scale subproblems without significantly affecting the performance of the optimization algorithm. The effectiveness of the approach depends greatly on the quality of this decomposition. An effective decomposition takes into account the interactions between the variables. However, recent strategies such as RDG and ERDG identify only additive interactions in the problem decomposition. As a result, these strategies tend to work well only for problems with this property. Nevertheless, in real-world problems, other types of separability besides additive separability may arise. This paper proposes a new decomposition strategy called Constrained Dual Differential Grouping (CDDG). The novelty of CDDG is that it can be used for both additive and multiplicative separable functions, which greatly increases the range of CC. To validate the effectiveness of CDDG, experiments and comparisons with state-of-the-art methods are conducted. The results demonstrate that CDDG can better adapt to the specific characteristics of each benchmark test problem.

Keywords: Cooperative Coevolution Algorithm, Large-Scale Constrained Problem, Differential Grouping

Una Estrategia de Descomposición de Alta Precisión para Problemas con Restricciones a Gran Escala

Abstract. Las estrategias de descomposición aplicadas a problemas de optimización, como los algoritmos Coevolutivos Cooperativos (CC), han demostrado ser eficaces para abordar diversos problemas de optimización con restricciones a gran escala. Este enfoque descompone un problema con restricciones de gran escala en subproblemas más simples y de menor escala, sin afectar negativamente el rendimiento del algoritmo de optimización. La eficacia del enfoque CC depende en gran medida de la calidad de dicha descomposición. Una buena descomposición tiene en cuenta la interacción entre las variables. Sin embargo, estrategias recientes como RDG y ERDG sólo identifican interacciones aditivas en la descomposición del problema. En consecuencia, estos métodos suelen ser eficaces únicamente para problemas que presentan esta propiedad. No obstante, en problemas reales pueden surgir otros tipos de separabilidad además de la aditiva. Este artículo propone una nueva estrategia de descomposición denominada Agrupación Diferencial Dual para problemas con restricciones (CDDG, por sus siglas en inglés). La novedad de CDDG radica en mejorar la descomposición en funciones separables, tanto aditivas como multiplicativas, lo que aumenta considerablemente el alcance de CC. Para validar la efectividad del CDDG, se llevan a cabo experimentos y comparaciones con métodos de vanguardia. Los resultados demuestran que CDDG puede adaptarse mejor a las características específicas de cada problema de prueba.

Keywords: Algoritmo Coevolutivo Cooperativo, Problemas con Restricciones a Gran Escala, Agrupación Diferencial, Separabilidad Aditiva y Multiplicativa

1 Introduction

Nowadays, many optimization problems include a substantial number of decision variables and are subject to a number of constraints that significantly complicate their solution. Such problems are called large-scale constrained optimization problems (LSCOP) [1]. A problem is considered large in this study if it presents scalability challenges to the most advanced algorithms. The scalability issue is known as the curse of dimensionality, attributed to: a) the exponential increase in the size of the search space and the number of local optima; b) numerous fitness evaluations required for good performance; c) high interaction between decision variables; and d) the complex search space generated by the possibility of partitioning it into multiple disconnected feasible regions [2]. Examples of this type of problem include power system [3], water distribution network [4], engineering design [5], and scheduling [6] and routing problems [7].

In recent years, the approach to deal with LSCOPs has been to divide the original problem into smaller and simpler subproblems to solve them separately. This approach, referred to as cooperative co-evolution (CC), has been utilized for various research objectives [8]. The CC approach is based on the divide and conquer strategy, which consists of three stages: (1) decomposition: an original large-scale problem, with or without constraints, is divided into simpler subproblems, commonly referred to as subcomponents; (2) optimization: each subproblem is solved using an evolutionary or swarm intelligence-based algorithm; and

(3) combination, in which the local solutions of the subproblems are integrated into a context vector to create a global solution. For the CC framework to have a favorable effect, it depends substantially on the quality of the decomposition process [9]. For the decomposition to be effective, it depends on how accurately the interactions between the variables are identified, so that they can be grouped into the same subcomponents.

Within the context of LSCOP, several decomposition strategies have been proposed, and we will briefly review them below. The decomposition of a problem often has two components: 1) a mechanism for identifying interactions between decision variables, and 2) a mechanism for generating a subproblem set based on the interaction information [2]. The mechanisms used to identify interactions in LSCOP decomposition are classified as:

- *Random Grouping*: is a basic technique that consists of randomly regrouping variables after each coevolutionary cycle to increase the probability of grouping interacting variables. There are two main drawbacks. First, the user must decide on the number and size of the components. Second, as the number of interacting variables increases, the probability of grouping multiple interacting variables simultaneously decreases significantly. The works of Vakhnin and Sopov [10] and Peng and Hui [11] provide examples of this group.
- *Fitness Difference Minimization*: These methods focus on adaptively reordering decision variables to minimize error functions while attempting to reduce the interaction between the resulting subcomponents. A drawback of these methods is that they require the user to specify the number or size of each subcomponent, such as the Variable Interaction Identification for Constrained Problems (VIIC) method proposed by Sayed et al. [12] and Aguilar-Justo et al. [13] developed Dynamic VIIC to address this problem by encouraging the generation of simpler subproblems without specifying the number of subgroups.
- *Finite Difference*: It is currently the most effective and accurate separability identification principle and is significantly better than the other two types of methods mentioned above. The finite difference principle is superior in terms of grouping accuracy and computational resource consumption, so it is adopted by state-of-the-art grouping methods. Differential Grouping (DG) and its extension DG2 are among the most widely used decomposition approaches, and have been applied to LSCOPs in the studies of Peng and Hui [14] and Aguilar-Justo and Mezura-Montes [15]. However, such methods require a high computational cost in terms of fitness evaluation (FE). Therefore, the Efficient RDG (ERDG) method [16] has emerged, which reduces the computational cost of RDG and improves the decomposition accuracy, thus offering significant potential for solving LSCOPs in the CC framework. ERDG has recently been implemented for LSCOPs in our previous work published in [17].

Although the DG, DG2, RDG, and ERDG methods are currently the most effective, they only detect additive interactions [18]. This limits their application to additively separable functions in LSCOPs. However, real-world problems could also exhibit other forms of separability. The Efficient Dual DG (EDDG) method recently proposed offers the ability to detect both additive and multiplicative interactions, achieving high efficiency and accuracy in solving large unconstrained problems [9]. In order to expand the application of CC in LSCOPs, this article proposes a decomposition strategy within the CC framework called CDDG. This decomposition strategy is designed to incorporate the advantages of EDDG to detect additive and multiplicative separability on LSCOP. To this end, it handles two sets of decompositions: one generated from the objective function and the other from the set of constraint functions.

The remainder of this paper is organized as follows: the next section briefly introduces the Basic Definitions. Section 3 provides a detailed introduction to our proposal, CDDG. Section 4 describes the numerical experiments on the benchmark function set. Section 5 discusses the experimental results. Finally, Section 6 concludes the paper.

2 Related Techniques

2.1 Basic Definitions

Problem Statement In a constrained optimization problem, the aim is to optimize $f(\mathbf{x})$ subject to specific constraints [19, 20]:

$$\begin{array}{rll} \text{Optimize} & f(\mathbf{x}) & \\ \text{Subject to:} & g_i(\mathbf{x}) \leq 0, & \text{for } i = 1, \ldots, q, \\ & h_j(\mathbf{x}) = 0, & \text{for } j = 1, \ldots, m. \end{array}$$

$g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ denote inequality and equality constraints, with q and m denoting their respective numbers. Equality constraints are usually transformed into inequalities as $|h_j(\mathbf{x})| - \epsilon \leq 0$, with a small tolerance of $\epsilon = 0.0001$. These constraints define the feasible search space for the algorithm to find solutions [21].

A solution is considered feasible if it belongs to the feasible region $\mathbb{F} \subseteq \mathbb{S}$, which is the set of solutions that satisfy all constraints. The feasibility of a solution is determined by the sum of the constraint violations. This sum must be 0 for the solution to be feasible, according to [19]. The sum of the constraint violations (φ) is represented as:

$$\varphi(\mathbf{x}) = \sum_{i=1}^{q} max(0, g_i(\mathbf{x})) + \sum_{j=1}^{m} max(0, |h_j(\mathbf{x})| - \epsilon) \tag{1}$$

The optimization process can be challenging when dealing with small or disjoint feasible regions using evolutionary or swarm intelligence algorithms.

Definition 1 (Separability). *A problem $f(\mathbf{x})$ is separable if:*

$$\arg \min_{(x_1,\ldots,x_D)} f(\mathbf{x}) = \left(\arg\min_{X_1} f(X_1), \ldots, \arg\min_{X_k} f(X_k) \right)$$

where $\mathbf{x}$ *is a vector of* D *decision variables* $(x_1, ..., x_D)$ *and* X_1,...,X_k *are disjoint subcomponents of* $\mathbf{x}$ *which are called non-separable variable groups.* k *is denoted as the number of subcomponents. When* $k = D$, $f(\mathbf{x})$ *is a fully separable problem. Otherwise,* $f(\mathbf{x})$ *is a partially separable function.*

Remark 1. A problem is considered separable when one or more variables can be optimized independently, without taking into account the influence of other variables. In such cases, the other variables are usually fixed by context vectors. The variables involved in a separable function are called "separable variables" in this work.

Definition 2 (Additive Separability [22]). *If the variables of a D-dimensional function* $f(\mathbf{x})$ *can be divided into* k $(k \geq 2)$ *variable groups* X_1,...,X_k*, satisfying:*

$$f(\mathbf{x}) = \sum_{i=1}^{k} f_i(X_i), \quad k = 2, \ldots, D \tag{2}$$

then the function is additively separable.

Definition 3 (Multiplicative Separability [22]). *If the variables of a D-dimensional function* $f(\mathbf{x})$ *can be divided into* k $(k \geq 2)$ *variable groups* X_1,...,X_k*, satisfying:*
1. $f(\mathbf{x}) = \prod_{i=1}^{k} f_i(X_i)$;
2. $f_i(X_i) \geq 0$ *or* $f_i(X_i) < 0, \quad i = 1, 2, \ldots, k$; *(are positive or negative).*
then the function is multiplicatively separable.

2.2 EDDG

The Efficient Dual DG (EDDG) method is a decomposition technique for solving large-scale expensive optimization problems [9]. EDDG combines the efficiency of ERDG with the ability of DDG to detect additive and multiplicative interactions, and its process is summarized below:

1. Divide the variable $\mathbf{x}$ into two groups: X_1 (the first variable) and X_2 (the remaining variables).
2. Identify the interactions of the variables. The differences in the fitness values ($f(\cdot)$) when perturbing variables in X_1 and X_2 are calculated. If the difference is not zero, there is an interaction.
   (a) Additive interactions: suppose that X_1 and X_2 are two mutually exclusive subsets of decision variables, which means that $X_1 \cap X_2 = \varnothing$. If there are two unit vectors $\mathbf{u}_1 \in U_{X_1}$ and $\mathbf{u}_2 \in U_{X_2}$, two arbitrary numbers $l_1, l_2 > 0$, and a candidate solution $\mathbf{x}^*$ in the search space, such that:

$$\Delta_1^{\text{add}} = f\left(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2\mathbf{u}_2\right) - f\left(\mathbf{x}^* + l_2\mathbf{u}_2\right) \tag{3}$$

$$\Delta_2^{\text{add}} = f\left(\mathbf{x}^* + l_1\mathbf{u}_1\right) - f\left(\mathbf{x}^*\right) \tag{4}$$

$$\left|\Delta_1^{\text{add}} - \Delta_2^{\text{add}}\right| > \varepsilon_{\text{add}} \tag{5}$$

(b) Multiplicative interaction: includes a logarithmic transformation to convert multiplicative interactions to additive interactions using the same approach as additive interactions.

$$\varDelta_1^{\text{multi}} = \ln\left(f\left(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2\mathbf{u}_2\right)\right) - \ln\left(f\left(\mathbf{x}^* + l_2\mathbf{u}_2\right)\right) \tag{6}$$

$$\varDelta_2^{\text{multi}} = \ln\left(f\left(\mathbf{x}^* + l_1\mathbf{u}_1\right)\right) - \ln\left(f\left(\mathbf{x}^*\right)\right) \tag{7}$$

$$\left|\varDelta_1^{\text{multi}} - \varDelta_2^{\text{multi}}\right| > \varepsilon_{\text{multi}} \tag{8}$$

$$\varepsilon_{\text{add}} = \gamma\sqrt{D+2} \cdot \left(|f(\mathbf{x})| + |f(\mathbf{x}')| + |f\mathbf{y})| + |f(\mathbf{y}')|\right) \tag{9}$$

$$\varepsilon_{\text{multi}} = \gamma\sqrt{D+2} \cdot \left(|\ln(f(\mathbf{x}))| + |\ln(f(\mathbf{x}'))| + |\ln(f(\mathbf{y}))| + |\ln(f(\mathbf{y}'))|\right) \tag{10}$$

ε_{add} and ε_{multi} are adaptive thresholds, $\gamma\sqrt{D+2}$ is associated with the IEEE 754 rounding error [23]. Here $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{y}$, and $\mathbf{y}'$ denote different evaluation points for identifying interactions according to RDG2 [24].

3. Group if there is an interaction. Integrate X_1 with the variables of X_2 responsible for the interaction. If there is no interaction, label X_1 as separable and proceed with the other variables.
4. Repeat the process recursively. Divide X_2 into smaller subgroups and repeat steps 2 and 3.
5. Treat large groups. If a resulting group is too large, divide it randomly into smaller groups, even if you lose theoretical accuracy.

EDDG simultaneously detects additive and multiplicative interactions without increasing FEs, reducing the computational cost to $O(D \log D)$ like ERDG.

3 Decomposition Strategy: CDDG method

Our proposed method, "Constrained Dual Differential Grouping" (CDDG), improves the accuracy of decomposition in constrained problems by combining the efficiency of EDDG in detecting additive and multiplicative interactions with that of CEDG [17] in identifying structure in constrained spaces. Both CEDG and EDDG benefit from the efficiency of ERDG. CDDG determines additive and multiplicative separability in both the objective function and the constraint variables. Algorithms 1 and 2 detail this procedure.

The Algorithm 1 starts by assigning x_1 to X_1 and the other variables to X_2. Iteratively, it evaluates the interactions between X_1 and X_2, reassigning the variables between the *sep* and *nonsep* groups according to the interaction analysis. Similar to [11, 25], CDDG selects the value of the objective function or constraint violations based on $flag$. The selected value is stored in F for later detection of interactions between variables. Unlike EDDG, CDDG uses an mx4-dimensional matrix F, where the m rows correspond to the number of constraints if $flag =$ `'cons'`; otherwise, m is 1 (single objective function). Moreover, CDDG maintains the sizes of the non-separable subcomponents ($NonSeps$) for enhanced decomposition accuracy and groups the separable variables into a single subcomponent (*Seps*). The CDDG returns the *seps* and *nonseps* groups, which create the initial subcomponents of the process.

Algorithm 1 CDDG process

Require: objective function: $f(\cdot)$, lower and upper bound: **lb**, **ub**, an indicator: *flag*
Ensure: separable groups: *sep*, nonseparable groups: *nonsep*

$sep, nonsep \leftarrow \emptyset$
$\mathbf{x}_{l,l} \leftarrow \mathbf{lb}$, $(y_{l,l}, vio_{l,l}) \leftarrow f(\mathbf{x}_{l,l})$
if $flag$ = 'cons' **then**
    $y_{l,l} \leftarrow vio_{l,l}$
end if
$X_1 \leftarrow \{x_1\}$, $X_2 \leftarrow \{x_2, ..., x_D\}$
while $X_2 \neq \emptyset$ **do**
    $\mathbf{x}_{u,l} \leftarrow \mathbf{x}_{l,l}$, $\mathbf{x}_{u,l}(X_1) \leftarrow \mathbf{ub}(X_1)$
    $(y_{u,l}, vio_{u,l}) \leftarrow f(\mathbf{x}_{u,l})$
    if $flag$ = 'cons' **then**
        $y_{u,l} \leftarrow vio_{u,l}$
    end if
    for $i = 1$ **to** m **do**
        $F_i \leftarrow \{y_{l,l}(i), y_{u,l}(i), \mathrm{nan}, \mathrm{nan}\}$
    end for
    $({X_1}^*, \hat{\beta}_{add}, \hat{\beta}_{multi}) \leftarrow$ **Interact**$(X_1, X_2, \mathbf{x}_{l,l}, \mathbf{x}_{u,l}, \mathbf{ub}, \mathbf{lb}, F, flag)$
    if $|{X_1}^*| = |X_1|$ **then**
        if $|{X_1}^*| > 1$ **then**
            $nonsep \leftarrow nonsep \cup \{{X_1}^*\}$
        else
            $sep \leftarrow sep \cup {X_1}^*$
        end if
        $X_1 \leftarrow \{x\}$, $X_2 \leftarrow X_2 - \{x\}$
    else
        $X_1 \leftarrow {X_1}^*$, $X_2 \leftarrow X_2 - X_1$
    end if
end while

Algorithm 2 calculates the additive (line 15) and multiplicative (line 19) differences for each function associated with a row of F, i.e., for each F_i. This enables simultaneous detection of variable interactions across all functions. If X_1 and X_2 have neither additive nor multiplicative separability, X_2 is split into two equal disjoint subsets, X_2' and X_2'' (line 29), and the interaction of X_1 with these subsets is evaluated recursively until all variables interacting with X_1 are identified. By splitting X_2, if X_1 does not interact with X_2', it follows that it interacts with X_2'' (line 34). Similarly, if X_1 interacts with X_2' and either $\beta_{add} = \hat{\beta}_{add}$ or $\beta_{multi} = \hat{\beta}_{multi}$ is met, the CDDG determines that X_1 does not interact with X_2'' (line 41). This structure, adapted from ERDG, enables CDDG to save FEs throughout the decomposition process.

4 Experimental Study

In this section, we perform numerical experiments to evaluate our proposal: CDDG. The "Experimental Setup" subsection details the experimental design, while the "Experimental Results" subsection provides the results of our proposal along with comparative methods.

Algorithm 2 Interact

Require: X_1, X_2, $\mathbf{x}_{l,l}$, $\mathbf{x}_{u,l}$, $\mathbf{ub}$, $\mathbf{lb}$, F, $flag$
Ensure: X_1^*, $\hat{\beta}_{add}$, $\hat{\beta}_{multi}$

initialize VecNS, NS $\leftarrow 1$
if $F[1,2] = \texttt{nan}$ **then** ▷ column 3 of F is NaN
  $\mathbf{x}_{m,l} \leftarrow \mathbf{x}_{l,l}$
  $\mathbf{x}_{m,l}(X_2) \leftarrow (\mathbf{lb}(X_2) + \mathbf{ub}(X_2))/2$
  $\mathbf{x}_{u,m} \leftarrow \mathbf{x}_{u,l}$
  $\mathbf{x}_{u,m}(X_2) \leftarrow (\mathbf{lb}(X_2) + \mathbf{ub}(X_2))/2$
  $(F[:,2], vio_{m,l}) \leftarrow f(\mathbf{x}_{m,l})$
  $(F[:,3], vio_{u,m}) \leftarrow f(\mathbf{x}_{u,m})$
  if $flag$ = **'cons'** **then**
    $F[:,2] \leftarrow vio_{m,l}$, $F[:,3] \leftarrow vio_{u,m}$
  end if
  for $i = 1$ **to** m **do**
    $\Delta_1^{\text{add}} \leftarrow (F[i,0] - F[i,1])$, $\Delta_2^{\text{add}} \leftarrow (F[i,2] - F[i,3])$
    $\beta_{\text{add}} \leftarrow |\Delta_1^{\text{add}} - \Delta_2^{\text{add}}|$
    if elements in F are all positive **then**
      $\Delta_1^{\log} \leftarrow \ln(F[i,0]) - \ln(F[i,1])$,
      $\Delta_2^{\log} \leftarrow \ln(F[i,2]) - \ln(F[i,3])$
      $\beta_{\text{multi}} \leftarrow |\Delta_1^{\log} - \Delta_2^{\log}|$
    end if
    if $\beta_{\text{add}} \leq \varepsilon_{\text{add}}$ **and** $\beta_{\text{multi}} \leq \varepsilon_{\text{multi}}$ **then**
      VecNS[i] $\leftarrow 0$
    end if
  end for
  NS $\leftarrow$ max(VecNS) ▷ NS=0 indicates separable variable
end if
if NS $= 1$ **then**
  if len$(X_2) > 1$ **then**
    Divide X_2 into X_2' y X_2''
    $F' = \{F[:,0], F[:,1], \texttt{nan}, \texttt{nan}\}$
    $(X_1', \hat{\beta}_{add}, \hat{\beta}_{multi}) = \textbf{Interact}(X_1, X_2', \mathbf{x}_{l,l}, \mathbf{x}_{u,l}, \mathbf{ub}, \mathbf{lb}, F', flag)$
    if $\beta_{add} \neq \hat{\beta}_{add}$ **and** $\beta_{multi} \neq \hat{\beta}_{multi}$ **then**
      if len(X_1') = len(X_1) **then**
        $(X_1'', \beta_{add}', \beta_{multi}') = \textbf{Interact}(X_1, X_2'', \mathbf{x}_{l,l}, \mathbf{x}_{u,l}, \mathbf{ub}, \mathbf{lb}, F, flag)$
      else
        $F' = \{F[:,0], F[:,1], \texttt{nan}, \texttt{nan}\}$
        $(X_1'', \beta_{add}', \beta_{multi}') = \textbf{Interact}(X_1, X_2'', \mathbf{x}_{l,l}, \mathbf{x}_{u,l}, \mathbf{ub}, \mathbf{lb}, F', flag)$
      end if
      $X_1 \leftarrow X_1' \cup X_1''$
    else
      $X_1 \leftarrow X_1'$
    end if
  else
    $X_1 \leftarrow X_1' \cup X_2$
  end if
end if

4.1 Experimental Setup

Our proposal is compared with the most recent decomposition techniques in the literature: CEDG and RDG [1] used for LSCOPs. The latter has been used in COCC-SF-DE [1] and COCCBSO20 [26].

Benchmark functions We adopt a benchmark function set consisting of 12 constrained functions proposed by [1], which are classified into 5 different categories. Each function is run 25 times with up to $3 \cdot 10^6$ FEs.

C1: f_1 is a fully-separable function

C2: f_2-f_3 are partially additive separable functions I
C3: f_4-f_8 are partially additive separable functions II
C4: f_9-f_{10} are overlapping functions
C5: f_{11}-f_{12} are fully non-separable functions

These 12 benchmark functions are the most recent; yet, they are not multiplicatively separable. To address this, we designed four Extended LSCOP (ExLSCOP) constructed and transformed from two multiplicatively separable basis functions. The Product of Squares Function (F_{prodsqu}) and the Product of Rastrigin Function (F_{prodras}), which were designed by [27], are included. We add a constraint of the additive separability type to each basis function to demonstrate that our CDDG method detects both types of separability simultaneously. The four ExLSCOP are shown below in Table 1.

Evaluation Metric We evaluate CDDG's performance using decomposition accuracy (DA). DA uses two indicators: (1) the number of FEs used to decompose the problem, and (2) the percentage of interacting decision variables grouped correctly. The best performance is defined by accuracy, but when methods achieve the same accuracy, the one requiring fewer FEs is superior [16]. Let *nonsep* and *sep* denote the groups of non-separable and separable variables, respectively. *nonsep*′ and *sep*′ denote the grouping result of a grouping method. The grouping accuracy of a method is defined as follows.

1. For separable variables
$$\frac{|sep' \cap sep|}{sep}$$
2. For non-separable variables
$$\frac{\sum_{\hat{g}_i \in nonsep'} |\hat{g}_i|}{\sum_{g_i \in nonsep} |g_i|}$$

4.2 Experimental Results

Table 2 summarizes the decomposition results. To enhance the readability of the table, rows associated with function results are interlaced with light gray shading. For additively and multiplicatively fully separable functions, such as f_1, f_{13}, and f_{14}, CDDG performs similarly to the other methods that were compared. For these functions, the methods succeed in determining 100% DA. Conversely, none of the decomposition methods can identify the separable variables in the objective function of f_3. Instead, they are all grouped into a single non-separable subcomponent. This difficulty may be due to the smooth interactions of the Ackley objective function, which directly affects the threshold ε_{add}. While the other methods fail for functions f_4, f_6, f_{15}, and f_{16}, CDDG achieves 100% accuracy

Table 1. Extended Benchmark Functions for LSCOP(ExLSCOP)

Func	Category	Objective Function and Constraint
f_{13}	Fully Separable Problem	$F_{13}(\mathbf{x}) = F_{\text{prodsqu}}(z) - 1 = \prod_{i=1}^{D}\left(\frac{z_i^2}{D} + 1\right) - 1$ $G(z) = 4 - \sum_{i=1}^{D} \lvert z_i \rvert \leq 0$
f_{14}		$F_{14}(\mathbf{x}) = F_{\text{prodras}}(z) - 1 = \prod_{i=1}^{D}\left(\frac{z_i^2 - 10\cos(2\pi z_i) + 10}{D} + 1\right) - 1$ $G(z) = 4 - \sum_{i=1}^{D} \lvert z_i \rvert \leq 0$
f_{15}	Partially separable problem with $\frac{D}{2m}$ groups	$F_{15}(\mathbf{x}) = \prod_{k=1}^{\frac{D}{2m}} F_{\text{rot\_prodsqu}}\left[z(P_{(k-1)*m+1} : P_{k*m})\right] \cdot F_{\text{prodsqu}}\left[z(P_{\frac{D}{2}+1} : P_D)\right] - 1$ $G(z) = \sum_{j=1}^{D}\left[z_j^2 - 5000\cos(0.1\pi z_j) - 4000\right] \leq 0$
f_{16}		$F_{16}(\mathbf{x}) = \prod_{k=1}^{\frac{D}{2m}} F_{\text{rot\_prodras}}\left[z(P_{(k-1)*m+1} : P_{k*m})\right] \cdot F_{\text{prodras}}\left[z(P_{\frac{D}{2}+1} : P_D)\right] - 1$ $G(z) = \sum_{j=1}^{D}\left[z_j^2 - 5000\cos(0.1\pi z_j) - 4000\right] \leq 0$

$\mathbf{z} = \mathbf{x} + 0$; $\mathbf{o}$ is the shifted global optimum.
$\mathbf{P}$ is a random permutation of all variables.
$F_{\text{rot}}(\mathbf{x}) = F(\mathbf{x} \cdot M)$ where M is a rotation matrix.

for partially separable and overlapping functions. CEDG's inaccuracy in decomposing f_4 and f_6 may be due to its use of an ε_{add} adopted from Sun et al. [28], whereas CDDG and RDG use an ε_{add} defined in their original papers Xu et al. [1]. CEDG and RDG only identify 10% of multiplicatively separable variables for f_{14} and f_{16} because they rely on the DG principle, which can only identify additive separability. Regarding FEs consumption, the table shows that CEDG is slightly better than CDDG and significantly better than RDG. The slight difference between CDDG and CEDG is primarily due to the size of their respective subcomponents. However, CDDG still saves more FEs than RDG.

5 Conclusion

This paper introduces CDDG as a method for addressing Large-Scale Constrained Optimization Problems (LSCOPs). We propose a decomposition technique, CDDG, capable of identifying both additively and multiplicatively separable variables,which can subsequently be integrated into a unified CC framework. In addition, an extension of LSCOP called ExLSCOP was developed. The instances in ExLSCOP are constructed based on multiplicative separability and consist of four benchmark problems, each with a single constraint and 1,000 dimensions. ExLSCOP addresses the gap in existing benchmarks, which previously lacked such problem types. We compared CDDG with leading decomposition methods on LSCOP and ExLSCOP. CDDG decomposes problems effectively and achieves higher overall accuracy, especially on additively and multiplicatively separable benchmark functions. Future work will integrate CDDG into a CC algorithm and analyze its performance using recent optimization algorithms to solve LSCOP problems with multiplicative and additive separability properties.

Table 2. The DAs and FEs used from CDDG, CEDG, and RDG in LSCOPs. The results of best performing algorithms are highlighted in boldface.

Function ID		FEs Used	CDDG Accuracy Sep	CDDG Accuracy Non-Seps	FEs Used	CEDG Accuracy Sep	CEDG Accuracy Non-Seps	FEs Used	RDG Accuracy Sep	RDG Accuracy Non-Seps
f_1	G_{obj}*	5996	100%	-	5996	100%	-	**5994**	**100%**	-
	G_ϕ*		100%	-		100%	-		**100%**	-
f_2	G_{obj}	11172	100%	100%	**11170**	**100%**	**100%**	19506	100%	100%
	G_ϕ		100%	100%		**100%**	**100%**		100%	100%
f_3	G_{obj}	**14773**	**0%**	**100%**	14856	0%	100%	29355	0%	100%
	G_ϕ		-	**100%**		-	100%		-	100%
f_4	G_{obj}	**20386**	-	**100%**	18175	-	55%	37962	-	100%
	G_ϕ		-	**100%**		-	100%		-	100%
f_5	G_{obj}	17168	-	100%	**17138**	-	**100%**	37638	-	100%
	G_ϕ		-	100%		-	**100%**		-	100%
f_6	G_{obj}	**30092**	-	**100%**	29609	-	73%	68376	-	100%
	G_ϕ		-	**100%**		-	100%		-	100%
f_7	G_{obj}	18204	-	100%	**18188**	-	**100%**	38334	-	100%
	G_ϕ		-	100%		-	**100%**		-	100%
f_8	G_{obj}	18810	-	100%	**18692**	-	**100%**	38622	-	100%
	G_ϕ		-	100%		-	**100%**		-	100%
f_9	G_{obj}	**28138**	-	**100%**	28101	-	5%	53862	-	100%
	G_ϕ		**100%**	-		100%	-		100%	-
f_{10}	G_{obj}	**16539**	-	**100%**	15416	-	35%	34662	-	100%
	G_ϕ		-	**100%**		-	84%		-	100%
f_{11}	G_{obj}	13852	-	100%	**13831**	-	**100%**	25494	-	100%
	G_ϕ		-	100%		-	**100%**		-	100%
f_{12}	G_{obj}	**7992**	-	**100%**	**7992**	-	**100%**	11982	-	100%
	G_ϕ		-	**100%**		-	**100%**		-	100%
f_{13}	G_{obj}	5996	100%	-	5996	100%	-	**5994**	**100%**	-
	G_ϕ		100%	-		100%	-		**100%**	-
f_{14}	G_{obj}	5996	100%	-	5996	100%	-	**5994**	**100%**	-
	G_ϕ		100%	-		100%	-		**100%**	-
f_{15}	G_{obj}	**13980**	**100%**	**100%**	11125	100%	10%	20580	100%	10%
	G_ϕ		**100%**	**100%**		100%	100%		100%	100%
f_{16}	G_{obj}	**14044**	**100%**	**100%**	11129	100%	10%	20766	100%	10%
	G_ϕ		**100%**	**100%**		100%	100%		100%	100%

G_{obj}*=$Group_{obj}$; G_ϕ*= $Group_\phi$

References

[1] P. Xu, W. Luo, X. Lin, J. Zhang, Y. Qiao, and X. Wang, "Constraint-objective cooperative coevolution for large-scale constrained optimization," *ACM Transactions on Evolutionary Learning and Optimization*, vol. 1, no. 3, pp. 1–26, 2021.

[2] M. Omidvar, X. Li, and X. Yao, "A review of population-based metaheuristics for large-scale black-box global optimization: Part b," *IEEE Transactions on Evolutionary Computation*, 2021.

[3] D. Liu, Z. Hu, Q. Su, and M. Liu, "A niching differential evolution algorithm for the large-scale combined heat and power economic dispatch problem," *Applied Soft Computing*, vol. 113, p. 108 017, 2021.

[4] Y.-H. Jia, Y. Mei, and M. Zhang, "A two-stage swarm optimizer with local search for water distribution network optimization," *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 1667–1681, 2021.

[5] A. Ghosh, K. Deb, E. Goodman, and R. Averill, "An interactive knowledge-based multiobjective evolutionary algorithm framework for practical op-

timization problems," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 223–237, 2023.

[6] J. Zhang, L. Xing, G. Peng, F. Yao, and C. Chen, "A large-scale multi-objective satellite data transmission scheduling algorithm based on svm y nsga-ii," *Swarm and Evolutionary Computation*, vol. 50, p. 100 560, 2019.

[7] J. Xiao, T. Zhang, J. Du, and X. Zhang, "An evolutionary multiobjective route grouping-based heuristic algorithm for large-scale capacitated vehicle routing problems," *IEEE transactions on cybernetics*, vol. 51, no. 8, pp. 4173–4186, 2019.

[8] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *International Conference on Parallel Problem Solving from Nature*, Springer, 1994, pp. 249–257.

[9] R. Zhong, E. Zhang, and M. Munetomo, "Cooperative coevolutionary surrogate ensemble-assisted differential evolution with efficient dual differential grouping for large-scale expensive optimization problems," *Complex and Intelligent Systems*, vol. 10, no. 2, pp. 2129–2149, 2024.

[10] A. Vakhnin and E. Sopov, "Improving de-based cooperative coevolution for constrained large-scale global optimization problems using an increasing grouping strategy," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 734, 2020, p. 012 099.

[11] C. Peng and Q. Hui, "Epsilon-constrained ccpso with different improvement detection techniques for large-scale constrained optimization," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, IEEE, 2016, pp. 1711–1718.

[12] E. Sayed, D. Essam, R. Sarker, and S. Elsayed, "Decomposition-based evolutionary algorithm for large scale constrained problems," *Information Sciences*, vol. 316, pp. 457–486, 2015.

[13] A. E. Aguilar-Justo, E. Mezura-Montes, S. M. Elsayed, and R. A. Sarker, "Decomposition of large-scale constrained problems using a genetic-based search," in *2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, IEEE, 2016, pp. 1–6.

[14] C. Peng and Q. Hui, "Comparison of differential grouping and random grouping methods on sccpso for large-scale constrained optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2016, pp. 2057–2063.

[15] A. E. Aguilar-Justo and E. Mezura-Montes, "A local cooperative approach to solve large-scale constrained optimization problems," *Swarm and Evolutionary Computation*, vol. 51, p. 100 577, 2019.

[16] M. Yang, A. Zhou, C. Li, and X. Yao, "An efficient recursive differential grouping for large-scale continuous problems," *IEEE Transactions on Evolutionary Computation*, 2020.

[17] F. Paz, G. Leguizamon, and E. Mezura-Montes, "Efficient differential grouping method for large-scale constrained optimization," in *30 Congreso Argentino de Ciencias de la Computacion (CACIC). Del 7 al 10 de octubre, 2024)*, 2024.

[18] J. Liu, R. Sarker, S. Elsayed, D. Essam, and N. Siswanto, "Large-scale evolutionary optimization: A review and comparative study," *Swarm and Evolutionary Computation*, p. 101 466, 2024.

[19] I. Rahimi, A. H. Gandomi, F. Chen, and E. Mezura-Montes, "A review on constraint handling techniques for population-based algorithms: From single-objective to multi-objective optimization," *Archives of Computational Methods in Engineering*, vol. 30, no. 3, pp. 2181–2209, 2023.

[20] T. Takahama and S. Sakai, "Efficient constrained optimization by the e-constrained adaptive differential evolution," in *IEEE congress on evolutionary computation*, IEEE, 2010, pp. 1–8.

[21] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, no. 2-4, pp. 311–338, 2000.

[22] J.-Y. Li, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Dual differential grouping: A more general decomposition method for large-scale optimization," *IEEE Transactions on Cybernetics*, 2022.

[23] IEEE, "Ieee standard for floating-point arithmetic," *IEEE Std 754-2019*, 2019.

[24] Y. Sun, M. N. Omidvar, M. Kirley, and X. Li, "Adaptive threshold parameter estimation with recursive differential grouping for problem decomposition," in *Proceedings of the genetic and evolutionary computation conference*, 2018, pp. 889–896.

[25] F. Paz, G. Leguizamon, and E. Mezura-Montes, "A novel cooperative coevolutionary algorithm for large-scale constrained optimization," in *Argentine Congress of Computer Science*, Springer, 2024, pp. 19–33.

[26] Y. Sun, P. Xu, Z. Zhang, T. Zhu, and W. Luo, "Brain storm optimization integrated with cooperative coevolution for large-scale constrained optimization," in *International Conference on Swarm Intelligence*, Springer, 2023, pp. 356–368.

[27] M. Chen, W. Du, Y. Tang, Y. Jin, and G. G. Yen, "A decomposition method for both additively and nonadditively separable problems," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1720–1734, 2022.

[28] Y. Sun, X. Li, A. Ernst, and M. N. Omidvar, "Decomposition for large-scale optimization problems with overlapping components," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2019, pp. 326–333.