

High-Dimensional Constrained Discrete Expensive Black-Box Optimization Using a Two-Phase Surrogate Approach

Rommel G. Regis(✉)

Saint Joseph's University, Philadelphia, PA 19131, USA
rregis@sju.edu

Abstract. This paper develops an extension of a surrogate-based algorithm for high-dimensional discrete constrained black-box optimization called CONDOR that can be used when no feasible initial point is provided. The proposed extension uses a two-phase approach where the first phase searches for a feasible point whose objective value is then improved in the second phase. Each iteration of Phase I identifies the infeasible points that are nondominated according to three criteria: number of constraint violations, maximum constraint violation, and sum of squares of constraint violations. Then, multiple trial points are generated in some neighborhood of a randomly chosen nondominated point. The function evaluation point is then chosen from the trial points according to the predicted values of the above criteria. Moreover, each iteration of Phase II generates multiple trial points in some neighborhood of the current best feasible point. Then, among the trial points that are predicted to be feasible, the function evaluation point is chosen according to the predicted objective value and distance from previous sample points. In the numerical experiments, radial basis function (RBF) surrogates are used and the proposed algorithm is applied to test cases and to a benchmark based on a car structure design problem that has 222 discrete ordinal variables and 54 black-box constraints. The proposed method outperforms a genetic algorithm and a direct search method in terms of the number of function evaluations to obtain a feasible point and in the best feasible objective value obtained within a relatively limited computational budget.

Keywords: Constrained black-box optimization · Discrete ordinal variables · High-dimensional optimization · Surrogates · Radial basis functions

1 Introduction

Many surrogate-based approaches have been developed for constrained black-box optimization where surrogates are used to approximate expensive black-box objective and constraint functions that could take a few minutes or many hours to evaluate for a given input. These black-box functions might be expensive because their values are outcomes of computational fluid dynamics (CFD) or finite element (FE) simulations. Some of these methods use Kriging or Gaussian Process (GP) models (e.g., [3,7,21]) while others use Radial Basis Function

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12953, pp. 366–381, 2021.
https://doi.org/10.1007/978-3-030-86976-2_25

(RBF) models (e.g., [1,18,25]). Most of these surrogate approaches for expensive black-box optimization have been developed for continuous optimization problems and relatively few of them can handle discrete variables [2]. Among the surrogate approaches for nonlinear integer or mixed-integer problems include the methods in [5,10,11,14,15,17,22,29]. However, most of these surrogate-based discrete black-box optimization methods have only been tested on problems with less than 25 variables and some do not handle black-box constraints (e.g., [15,29]). Moreover, surrogate methods that are suitable for high-dimensional problems with hundreds of discrete variables and many black-box constraints are still somewhat rare. One example is the CONDOR algorithm [26] and another is a constrained multi-objective evolutionary algorithm CMOEA/D assisted by an Extreme Learning Machine (ELM) surrogate [19]. The latter is designed for multi-objective constrained optimization, but it can be adapted for single-objective constrained optimization with discrete variables. CONDOR and ELM-assisted CMOEA/D have been tested on the 3-car Mazda benchmark problem [12,16] that involves 222 discrete variables and 54 black-box inequality constraints. CONDOR was applied to a single-objective version while ELM-assisted CMOEA/D was applied to the full multi-objective version of the Mazda benchmark. This paper proposes an extension of CONDOR that uses a two-phase surrogate approach to handle challenging high-dimensional constrained discrete expensive black-box optimization problems such as the Mazda benchmark.

Formally, this paper aims to find an approximate optimal solution to the following constrained optimization problem with ordinal discrete variables:

$$\begin{aligned} & \min_{x \in \mathbb{R}^d} \ f(x) \\ \text{s.t.} \quad & G(x) = (g_1(x), \ldots, g_m(x)) \leq 0 \\ & x^{(i)} \in D_i \subset \mathbb{R}, \ i = 1, \ldots, d \end{aligned} \tag{1}$$

where $x^{(i)}$ is the ith decision variable (the ith entry of $x \in \mathbb{R}^d$), and D_i is the finite set of the possible discrete ordinal settings of $x^{(i)}$. In practice, the elements in D_i do *not* have to be integers. They may be fractional values that the ith variable can take. For example, in the 3-car Mazda benchmark [12], $x^{(1)}$ takes on values from $D_1 = \{0.9, 0.95, 1.0, 1.2, 1.4, 1.5\}$. Moreover, the objective function $f(x)$ and the constraint functions $g_j(x)$, $j = 1, \ldots, m$ are black-box functions whose values are obtained from a computationally expensive simulation. In this paper, we assume that the simulations are deterministic and that there is no noise in the calculation of $f(x)$ and $G(x)$. The issue of noise in function evaluations will be addressed in future work.

In problem (1), the finite set $\prod_{i=1}^{d} D_i$ of (feasible and infeasible) points in $\mathbb{R}^d$ is the *search space* for the problem. The lower and upper bounds of the decision variables are given by $\ell_i := \min D_i$ and $u_i := \max D_i$ for $i = 1, \ldots, d$. For the proposed method, one *simulation* at a given point $x \in \prod_{i=1}^{d} D_i$ yields the values of the objective function $f(x)$ and all the components of the constraint function $G(x)$. However, some alternative optimization algorithms in the numerical comparisons require that separate calls are made to the objective function $f(x)$ and

the constraint function $G(x)$ and that more calls are made to the latter than the former in one iteration. Hence, in the numerical experiments, the methods are sometimes compared after some fixed number of objective and constraint function evaluations instead of a fixed number of simulations.

The CONDOR (CONstrained Discrete Optimization using Response surfaces) algorithm [26] is a surrogate approach that can handle computationally expensive and high-dimensional constrained black-box problems involving discrete ordinal variables and many black-box inequality constraints. CONDOR that uses RBF surrogates for the objective and the constraints significantly outperformed a genetic algorithm and NOMAD [13] on a high-dimensional benchmark problem based on a car structure design problem from Mazda [16] given a relatively small computational budget and when the algorithms are given an initial feasible point. The 3-car Mazda benchmark has 222 discrete ordinal variables and 54 black-box constraints [16] and is much larger than most application and test problems in the literature. Unlike the ELM-assisted CMOEA/D in [19] where the constraints are aggregated and reduced to only three constraints, CONDOR uses a surrogate for each for the 54 inequality constraints in the Mazda benchmark. However, CONDOR [26] requires at least one feasible initial point, which can be difficult to obtain in some engineering applications.

This paper proposes the *Two-Phase CONDOR* algorithm, which extends the capabilities of CONDOR to solve high-dimensional constrained discrete black-box optimization problems when no feasible initial point is given. The method is designed to quickly obtain a feasible point and then find other feasible points with better objective function values using only a relatively small number of simulations. In Phase I of the proposed algorithm, all previously evaluated points are infeasible and the algorithm searches for a feasible point using surrogates for the constraint and objective functions. In Phase II, the algorithm improves on the objective function of the feasible point found in Phase I. Two-Phase CONDOR is implemented using RBF surrogates and is applied to discrete versions of three artificial test problems. It is also tested on the 3-car Mazda benchmark with 222 discrete variables and 54 black-box constraints using a computational budget of only $10d = 2220$ simulations. The proposed method outperforms a genetic algorithm and the direct search method NOMAD [13] in terms of the number of function evaluations to obtain a feasible point and in the best feasible objective function value obtained within the given computational budgets.

2 The Two-Phase CONDOR Algorithm

2.1 Algorithm Description

This paper proposes the *Two-Phase CONDOR (CONstrained Discrete Optimization using Response surfaces)* algorithm for finding an approximate optimal solution to the computationally expensive constrained discrete black-box optimization problem in (1). The proposed method uses a two-phase approach to extend the surrogate-based CONDOR algorithm in [26] so that it can be used when no feasible point is available among the initial sample points. The original

CONDOR algorithm assumes that at least one feasible initial point is provided, but this might not be the case in some applications and finding such a feasible point can be challenging for high-dimensional problems with many constraints. The first phase of Two-Phase CONDOR finds a feasible point while the second phase searches for a solution that improves on the feasible point found in the first phase. In both phases, surrogates for the objective and constraint functions are used to select the points where the expensive simulations are carried out. The use of a two-phase approach for surrogate-based constrained optimization with continuous variables has been employed in the COBRA algorithm [25] and its extension SACOBRA [1]. However, Two-Phase CONDOR not only deals with ordinal discrete variables, its strategies for selecting the simulation points in both phases differ from those in the COBRA and SACOBRA algorithms.

As with any surrogate-based approach, Two-Phase CONDOR begins by selecting an initial set of points in the search space where the simulations will take place. These points may be generated by using a space-filling design that is suitable for discrete search spaces or by simply randomly selecting points uniformly from the discrete search space. Then simulations are performed to get the values of the objective and constraint functions at the initial points. These function values are then used to fit the initial surrogates. For constrained optimization problems, there is typically no guarantee that a feasible point is available among the initial points. Hence, Phase I of the proposed algorithm uses surrogates for the constraints and the objective to find a feasible point for the problem. Then, Phase II searches for a solution that improves on the objective function value of the first feasible point found using again the surrogates for the constraints and the objective. If there is a feasible point among the initial sample points, then the algorithm simply proceeds to Phase II and behaves in the same way as the original CONDOR algorithm in [26].

Phase I of the proposed method employs a multi-criteria approach in determining its best previously evaluated sample points, which are all infeasible in this phase. In particular, in every iteration of Phase I, the algorithm identifies the sample points that are nondominated according to three criteria: number of constraint violations, maximum constraint violation, and sum of squares of the constraint violations. These criteria are pairwise conflicting in general. For example, for $x = [x^{(1)}, x^{(2)}, x^{(3)}] \in \mathbb{R}^3$, consider the constraint function $G(x) = \left(\frac{5}{6}x^{(1)} + \frac{1}{2}x^{(2)} - x^{(3)},\ \ \frac{1}{3}x^{(1)} - \frac{1}{15}x^{(2)} + \frac{1}{15}x^{(3)},\ \ -\frac{2}{3}x^{(1)} - \frac{1}{15}x^{(2)} + \frac{16}{15}x^{(3)}\right)$ and the points $[1, 0, 1]$, $[0, 1, 1]$ and $[1, 1, 1]$. Then $G([1, 0, 1]) = (-1/6, 2/5, 2/5)$, $G([0, 1, 1]) = (-1/2, 0, 1)$ and $G([1, 1, 1]) = (1/3, 1/3, 1/3)$. Note that $[0, 1, 1]$ has the smallest number of constraint violations, $[1, 1, 1]$ has the smallest maximum constraint violation, and $[1, 0, 1]$ has the smallest sum of squares of the constraint violations.

After obtaining the nondominated points based on the constraint violation criteria, the algorithm fits or updates the surrogates for the objective and constraint functions. Next, the algorithm chooses a nondominated point uniformly at random from the current set of nondominated points and then generates many trial points in some neighborhood of this nondominated point. The trial points

are generated by perturbing some or all of the current variable settings of the chosen nondominated point as described below after the pseudocode. To select the simulation point, the algorithm uses the surrogates for the constraints to identify the *eligible* trial points, which are those that are predicted to be feasible. If none of the trial points are predicted to be feasible, the eligible trial points are those that have the smallest number of predicted constraint violations. The simulation point is then chosen from these trial points as the one with the best weighted combination of scaled values of the maximum predicted constraint violation (*MPCV criterion*) and scaled values of the sum of squares of the predicted constraint violations (*SPCV criterion*) as was done in the Two-Phase POSEIDON algorithm for constrained discrete multi-objective optimization [27]. On the other hand, if there are trial points that are predicted to be feasible, then the simulation point is chosen from these eligible trial points to be the one that minimizes a weighted combination of the scaled predicted objective value (*surrogate criterion*) and scaled distance from previous sample points (*distance criterion*) as was done in the ConstrLMSRBF algorithm [24]. Next, a simulation is performed to get the values of the objective and constraint functions at the chosen trial point. The algorithm goes through the iterations in Phase I until a feasible point is found, after which the algorithm proceeds to Phase II.

An alternative method for selecting the simulation point in Phase I is to use a multi-objective optimization algorithm to identify Pareto optimal points that provide a trade-off among the number of predicted constraint violations, the maximum predicted constraint violation, and the sum of squares of the predicted constraint violations subject to surrogates of the constraints. From these Pareto optimal points, the simulation point can be selected. This approach requires a multi-objective optimization algorithm that can handle discrete variables and constraints and will be explored in future work.

In every iteration of Phase II, the algorithm again fits the surrogates of the objective and the constraints and identifies the feasible sample point with the best objective function value. Next, the algorithm perturbs the variable settings of the current best feasible point multiple times to generate many trial points in some neighborhood of this best point. Then, as in Phase I, the surrogates for the constraints are used to determine the eligible trial points, which are those that are predicted to be feasible or at least those that have the smallest number of predicted constraint violations. Among these eligible trial points, the simulation point is selected to be the best trial point according to a weighted ranking of scaled values of two criteria as was done in [24]: predicted value of the objective function (*surrogate criterion*), and minimum distance between the trial point and the previously evaluated sample points (*distance criterion*). Once the best trial point is selected, a simulation is performed to get the values of the objective and constraint functions at that point, and then the algorithm iterates until the maximum number of simulations has been reached.

Below is a pseudo-code that outlines the main steps of the Two-Phase CONDOR algorithm for approximately solving the constrained discrete black-box optimization problem of the form (1).

Two-Phase CONDOR Algorithm

(1) *(Initial Simulations)* Run simulations to get the values of $f(x)$ and $G(x)$ at an initial set of points $\mathcal{I}_0 = \{x_1, \ldots, x_{n_0}\}$ in the search space $\prod_{i=1}^{d} D_i$. Set the simulation counter $n \leftarrow |\mathcal{I}_0| = n_0$ and the sample points $\mathcal{P}_{n_0} = \mathcal{I}_0$. If one of the points in $\mathcal{I}_0$ is feasible, go to Step 3 (Phase II).

(2) *(Phase I Iterations)* While a feasible sample point has not been found or while $n < n_{\max}$ do:

(a) *(Determine Nondominated Set)* Identify the sample points $\mathcal{N}_n^* \subseteq \mathcal{P}_n$ (all infeasible) that are nondominated according to three criteria: number of constraint violations, maximum constraint violation, and sum of squares of the constraint violations.

(b) *(Build Surrogates)* Build or update the surrogate for the objective $s_n^f(x)$ and the surrogates for the constraints $s_n^{g_i}(x),\ i = 1, \ldots, m$ using info from all previous sample points $\mathcal{P}_n$.

(c) *(Generate Trial Points)* Select a point x_n^* uniformly at random from $\mathcal{N}_n^*$ and generate many trial points $\mathcal{T}_n$ in some neighborhood of x_n^*. (Described below.)

(d) *(Identify Eligible Trial Points)* Evaluate the surrogates for the constraints $s_n^{g_i}(x)$ at the trial points $\mathcal{T}_n$ and identify the trial points $\mathcal{T}_n^*$ that are predicted to be feasible or at least those with the smallest number of predicted constraint violations.

(e) *(Choose Simulation Point)* If the trial points in $\mathcal{T}_n^*$ are predicted to be feasible, then do (i) below, else do (ii).

(i) Evaluate the surrogate for the objective $s_n^f(x)$ at the trial points in $\mathcal{T}_n^*$ and choose the simulation point x_{n+1} to be the trial point in $\mathcal{T}_n^*$ with the best weighted combination of scaled values of the surrogate and distance criteria.

(ii) From the eligible trial points $\mathcal{T}_n^*$, choose the simulation point x_{n+1} to be the trial point with the best weighted combination of scaled values of the MPCV and SPCV criteria.

(f) *(Simulate)* Perform one simulation at x_{n+1} to get the values of $f(x_{n+1})$ and $G(x_{n+1}) = (g_1(x_{n+1}), \ldots, g_m(x_{n+1}))$. Set $\mathcal{P}_{n+1} = \mathcal{P}_n \cup \{x_{n+1}\}$ and reset $n \leftarrow n + 1$.

end.

(3) *(Phase II Iterations)* While $n < n_{\max}$ do:

(a) *(Identify Best Feasible Point)* Determine the feasible sample point with the best objective function value x_n^*.

(b) *(Build Surrogates)* Build or update the surrogate for the objective $s_n^f(x)$ and the surrogates for the constraints $s_n^{g_i}(x),\ i = 1, \ldots, m$ using info from all previous sample points $\mathcal{P}_n$.

(c) *(Generate Trial Points)* Generate many trial points $\mathcal{T}_n$ in some neighborhood of x_n^*. (Described below.)

(d) *(Determine Eligible Trial Points)* Evaluate the surrogates for the constraints $s_n^{g_i}(x)$ at the trial points in $\mathcal{T}_n$ and determine the trial points $\mathcal{T}_n^*$ that are predicted to be feasible or at least those with the smallest number of predicted constraint violations.

(e) *(Choose Simulation Point)* Evaluate the surrogate for the objective $s_n^f(x)$ at the trial points in $\mathcal{T}_n^*$ and choose the simulation point x_{n+1} to be the trial point in $\mathcal{T}_n^*$ that with best weighted combination of scaled values of the surrogate and distance criteria.

(f) *(Simulate)* Perform one simulation at x_{n+1} to get the values of $f(x_{n+1})$ and $G(x_{n+1}) = (g_1(x_{n+1}), \ldots, g_m(x_{n+1}))$. Set $\mathcal{P}_{n+1} = \mathcal{P}_n \cup \{x_{n+1}\}$ and reset $n \leftarrow n + 1$.

end.

(4) *(Return Best Feasible Point)* Return the best feasible solution x_n^* and the values $f(x_n^*)$ and $G(x_n^*) = (g_1(x_n^*), \ldots, g_m(x_n^*))$.

In every iteration of the proposed algorithm, many trial points are generated in some neighborhood of a center point, which is either the chosen nondominated point in Phase I or the current best feasible point in Phase II. A trial point is generated by changing some of the variable settings of the center point. As with the original CONDOR algorithm [26], each component is modified with a certain probability, which we denote by p_{pert}. A perturbation consists of changing the current value of a discrete variable by either increasing or decreasing its value by a few discrete steps. The neighborhood depth parameter, which we denote by $depth_{\text{nbhd}}$, is the percentage of the number of settings that the variable is allowed to increase or decrease. For example, in the 3-car Mazda benchmark [16], there are 9 possible settings of $x^{(17)}$ given by $D_{17} = \{0.9, 0.95, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.1\}$ and suppose the current setting of variable $x^{(17)}$ at the current center point is 0.95. If $depth_{\text{nbhd}} = 20\%$, then $x^{(17)}$ is allowed to take 20% of the possible discrete values for that variable above or below the current value. Hence, $x^{(17)}$ may be increased or decreased from 0.95 up to $\lceil 0.2(9) \rceil = 2$ discrete steps, and so, $x^{(17)}$ is allowed to take on the possible settings $\{0.9, 1.0, 1.2\}$. Note that the current setting of 0.95 is excluded from the possible values in order to force the value of $x^{(17)}$ to change.

2.2 Radial Basis Function Interpolation Model

The Two-Phase CONDOR algorithm can be implemented using any type of surrogate such as Kriging or Gaussian process modeling [3,8], radial basis functions (RBF) (e.g., [1,23,25]) and neural networks (e.g., [6]). The numerical experiments below use the RBF interpolation model in Powell [23], which is described below. Fitting this model involves solving a linear system that possesses desirable mathematical properties such as a guarantee that the interpolation matrix is nonsingular under mild technical conditions. This RBF model is suitable for the Two-Phase CONDOR on the Mazda benchmark because it has been successfully applied to high-dimensional problems with over a hundred decision variables and many black-box inequality constraints (e.g., see [1,24,25]). In contrast, Kriging models sometimes suffer from numerical issues and can be computationally expensive to build on high-dimensional problems with hundreds of variables.

Next, we describe how to build this RBF model. Let $u(x)$ be the objective or one of the constraint functions. Suppose the function values at the $x_1, x_2, \ldots,$

$x_n \in \mathbb{R}^d$ are known, so that we have the data points $(x_1, u(x_1))$, $(x_2, u(x_2))$, $\ldots, (x_n, u(x_n))$. The RBF model from Powell [23] has the form:

$$s_n(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|) + p(x), \ x \in \mathbb{R}^d,$$

where $\|\cdot\|$ is the 2-norm, $\lambda_i \in \mathbb{R}$ for $i = 1, 2, \ldots, n$, $p(x)$ is a linear function in d variables, and ϕ can take many forms such as the *cubic* form ($\phi(r) = r^3$), the thin plate spline ($\phi(r) = r^2 \log r$), or the Gaussian form ($\phi(r) = \exp(-\gamma r^2)$, where γ is a hyperparameter). A cubic RBF model is used in the numerical experiments because it does not require a hyperparameter that needs to be tuned and because of its success in previous RBF methods (e.g., [24,25]). More details about how to fit this model are found in Powell [23].

3 Numerical Experiments

3.1 Description of the Test Problems and the Mazda Benchmark

The proposed Two-Phase CONDOR method is tested on discrete modifications of the well-known benchmark problems G7 ($d = 10$ variables, $m = 8$ constraints), G9 ($d = 7$, $m = 4$) and G19 ($d = 15$, $m = 5$) (e.g., see [25]). The modified problems are referred to as G7-Discrete, G9-Discrete and G19-Discrete. The number of discrete settings for each variable in the modified problems ranged from 6 to 18 and the number of points in the search spaces are 1.99×10^{10} for G7-Discrete, 1.30×10^{7} for G9-Discrete, and 1.12×10^{17} for G19-Discrete.

Two-Phase CONDOR is also tested on a single-objective version of the 3-car Mazda Benchmark [9,12,16,20] that is based on a car structure design problem. This benchmark has 222 discrete ordinal decision variables (thicknesses of structural parts) and 54 black-box inequality constraints (including collision safety performances), and it is among the largest application-based benchmark problems in expensive black-box optimization. The objective is to minimize the total weight of three types of Mazda cars (Mazda CX-5 (SUV), Mazda 6 (LV), and Mazda 3 (SV)) subject to the constraints. Each decision variable can only take values from a finite number of discrete settings ranging from 4 to 18. Moreover, the total number of possible combinations of values of the discrete variables is 4.4427×10^{198}, which is very large even by astronomical standards.

Evaluating the constraint functions in the design optimization of car structures requires running expensive simulations, but the evaluation of the Mazda benchmark is fast because the collision safety constraints are modeled by response surface approximations [12]. Moreover, calculating the objective function is relatively cheap since it is simply the total weight of three types of cars. However, in this paper, the objective function is also assumed to be black-box and expensive, which is the case in other applications. Future work will consider variants of the proposed method that take advantage of inexpensive objective functions. The Mazda benchmark is challenging for surrogate optimization because it has hundreds of decision variables and many black-box constraints, and it

is larger than the vast majority of test problems used in surrogate-based optimization. In addition, it is not easy to obtain feasible points for the problem by uniform random sampling because the size of the discrete feasible region is very small relative to the astronomically large discrete search space. Non-surrogate methods such as evolutionary algorithms and direct search methods are not expected to produce feasible points quickly on the Mazda benchmark and this is confirmed in the numerical experiments. More details about the Mazda benchmark can be found at https://ladse.eng.isas.jaxa.jp/benchmark/index.html.

3.2 Experimental Setup

Two-Phase CONDOR is implemented using RBF surrogates with the cubic form ($\phi(r) = r^3$) and three variants are considered: *CONDOR (RBF-local)*, *CONDOR (RBF-global)*, and *CONDOR (RBF-mixed)*. The first performs more local search for Phase I and Phase II, the second allows more global search for both phases, and the third is a mixture of more global search for Phase I combined with more local search for Phase II. The second phase of the first two variants coincide with the original CONDOR methods tested in [26], which only work when a feasible initial point is given. In each of the two phases, the three variants generate $20d$ trial points from the current center point (either a nondominated point in Phase I or the current best feasible point in Phase II) in each iteration. Moreover, the values of the perturbation probability p_{pert} and the neighborhood depth parameter $depth_{\text{nbhd}}$ vary in a cyclical manner in all variants. In particular, the algorithm parameters $(p_{\text{pert}}, depth_{\text{nbhd}})$ vary according to the following cycles: CONDOR (RBF-local) uses $\langle(0.5, 30\%), (0.1, 10\%), (0.05, 10\%), (0.01, 10\%)\rangle$ as in [26], CONDOR (RBF-global) uses $\langle(0.5, 50\%), (0.3, 50\%), (0.2, 10\%), (0.1, 10\%), (0.05, 10\%)\rangle$ as in [26], and CONDOR (RBF-mixed) uses the same cycle as CONDOR (RBF-global) for Phase I and the same cycle as CONDOR (RBF-local) in Phase II.

When the perturbation probability p_{pert} is fixed, the number of variables modified in the chosen center point is a binomial random variable whose mean is $d \cdot p_{\text{pert}}$. This means that when CONDOR uses $p_{\text{pert}} = 0.01$ in the Mazda benchmark, $d \cdot p_{\text{pert}} = 222(0.01) = 2.22$ variables are modified on average. Combining this with a depth parameter of 10%, CONDOR generates trial points that are highly local and close to the current center point. On the other hand, when CONDOR uses $p_{\text{pert}} = 0.5$, the number of variables modified on average is $222(0.5) = 111$, and combining this with a depth parameter of 50%, the algorithm yields trial points that are far from the center point and that have potential explore the search space globally.

The three variants of Two-Phase CONDOR are compared with NOMAD [13] and Matlab's Genetic Algorithm (GA) solver [28] with various crossover fractions. Before GA and NOMAD are applied, the possible discrete ordinal values of each variable are mapped to the integers 0, 1, 2, up to the maximum number of possible discrete values minus one. NOMAD is run on Matlab through the OPTI toolbox [4] and using its default settings. GA mostly used default settings, but with different crossover fractions set to 100%, 75%, 50% and 25%. Moreover,

GA is run with a population size of $5d$ for the artificial problems and 300 for the Mazda benchmark as in [12]. It would be best if CONDOR is compared with another surrogate-based method. Unfortunately, there does not seem to be any publicly available code for a surrogate-based algorithm for constrained discrete black-box optimization. As mentioned earlier, only a relatively small number of surrogate-based methods in the literature can be used for discrete black-box optimization [2], and even less can handle high-dimensional constrained black-box optimization problems.

Each algorithm is run for 30 trials on the artificial benchmark problems up to a maximum of $40d$ simulations (equivalent to $80d$ function evaluations when counting objective and constraint function calls separately). For the single-objective Mazda benchmark, each algorithm is run for 10 trials up to a maximum of $10d = 2220$ simulations (equivalent to $20d = 4440$ objective and constraint function evaluations). For a fair comparison, the CONDOR and GA algorithms used the same initial population in each trial. NOMAD also used the same initial points used by GA and CONDOR except that one of these initial points is chosen as the starting point of the algorithm. This starting point is chosen from among the initial points with the smallest number of constraint violations to be the best according to a weighted combination of scaled values of two criteria: maximum constraint violation and sum of squares of constraint violations. Moreover, the first trial on the Mazda benchmark used the same initial population as the one used in earlier experiments [16]. For the other trials on the Mazda benchmark and for all trials on the artificial problems, an initial population was obtained by randomly selecting points using a uniform distribution on the search space. In almost all trials, except for two trials on G9-Discrete, none of the points in the initial population are feasible, so the CONDOR algorithms started with Phase I.

All numerical experiments are carried out in Matlab 9.4 using an Intel(R) Core(TM) i7-7700T CPU @ 2.90 GHz, 2904 Mhz, 4 Core(s), 8 Logical Processor(s) Windows-based machine. The Mazda benchmark is run on Matlab through a system call that runs the executable provided in [16].

3.3 Results and Discussion

Table 1 shows the mean and standard error over 30 trials of the number of objective and constraint function evaluations (including those for the initial population) it took the different algorithms to obtain a feasible sample point on the artificial benchmark problems. The results for CONDOR (RBF-mixed) are not included since its Phase I is identical to that of CONDOR (RBF-global). Moreover, the results for GA with a crossover fraction of 25% are not included since they are comparable to the other GA results. Note that the CONDOR algorithms consistently required much less objective and constraint function evaluations to obtain a feasible point than NOMAD and GA on the artificial problems. Also, the results for the local and global variants are comparable with the global variant performing slightly better on the G9-Discrete and G19-Discrete problems.

Next, Table 2 shows the mean and standard error over 30 trials of the best feasible objective function value obtained by the CONDOR and NOMAD

Table 1. Mean (over 30 trials) of the number of objective and constraint function evaluations to obtain a feasible point on the test problems. The number inside the parenthesis is the standard error of the mean.

Test problem	CONDOR (RBF-local)	CONDOR (RBF-global)	NOMAD	GA (CR = 100%)	GA (CR = 75%)	GA (CR = 50%)
G7-Discrete	**168.87 (8.20)**	176.73 (12.71)	260.80 (11.38)	656.23 (32.69)	709.63 (34.98)	688.27 (42.05)
G9-Discrete	78.00 (3.24)	**75.93 (2.58)**	122.87 (8.41)	98.43 (8.04)	109.37 (8.94)	> 166.47
G19-Discrete	156.13 (0.88)	**154.93 (0.46)**	288.80 (11.43)	828.00 (74.26)	706.40 (50.26)	588.13 (45.19)

Table 2. Mean (over 30 trials) of the best feasible objective function value obtained by the algorithms in $80d$ objective and constraint function evaluations on the test problems. The number inside the parenthesis is the standard error of the mean.

Test problem	CONDOR (RBF-local)	CONDOR (RBF-global)	CONDOR (RBF-mixed)	NOMAD
G7-Discrete	**51.17 (5.66)**	65.97 (7.50)	60.57 (9.29)	69.00 (7.59)
G9-Discrete	728.00 (5.97)	**704.53 (3.15)**	725.07 (7.57)	746.23 (8.38)
G19-Discrete	341.52 (21.02)	**294.34 (16.70)**	343.70 (16.83)	1329.59 (389.62)

algorithms in $80d$ objective and constraint function evaluations on the artificial problems. The results for GA are not included since some of its trials did not even find a feasible sample point within the given computational budget. Note that the CONDOR algorithms consistently obtained better feasible objective function values than NOMAD (and also GA) especially on G19-Discrete.

The Two-Phase CONDOR algorithms are also compared with GA and NOMAD in how quickly they are able to obtain a feasible sample point on the Mazda benchmark. Figure 1 shows the mean of the minimum number of constraint violations (left plot) and the mean of the best sum of squares of constraint violations (right plot) over 10 trials obtained by the different algorithms at computational budgets up to about 2400 objective and constraint function evaluations on the Mazda benchmark. Both plots include error bars that correspond to 95% t confidence intervals for the mean. The t distribution is used in the calculation of confidence intervals since the number of trials for each algorithm is small. GA results are obtained for various crossover fractions of 25%, 50%, 75% and 100%. However, to simplify the presentation of the results, only the results for crossover fractions of 50% and 100% are shown since the other results are similar. Again, CONDOR (RBF-mixed) is not included in these comparisons since its Phase I is identical to that of CONDOR (RBF-global).

Figure 1 shows that the Two-Phase CONDOR algorithms are able to quickly reduce both the number of constraint violations and the sum of squares of the constraint violations of its sample points to zero and consistently yield feasible sample points within a relatively small number of objective and constraint function evaluations on the Mazda benchmark. Moreover, the results for Two-Phase CONDOR are much better than those for NOMAD and GA. In fact, none of

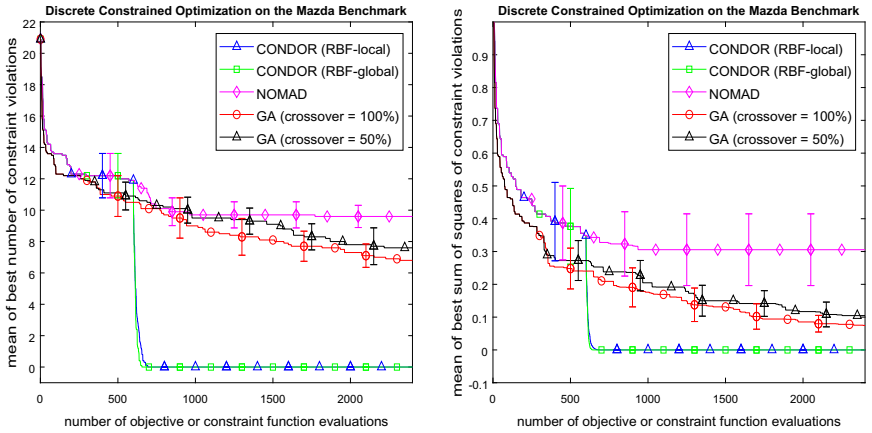

Fig. 1. Mean over 10 trials and 95% t confidence intervals for the minimum number of constraint violations (left) and the best sum of squares of the constraint violations (right) obtained by the Two-Phase CONDOR algorithms and alternative methods at different computational budgets on the Mazda benchmark.

Table 3. Number of simulations to feasibility and first feasible objective value obtained for 10 trials of Two-Phase CONDOR on the Mazda benchmark.

Algorithm	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10	Mean
CONDOR	340	331	320	331	325	318	324	312	310	347	325.8
(RBF-local)	3.0779	3.0369	3.1085	3.0464	3.0992	3.0998	3.0895	3.0751	3.1020	3.1461	
CONDOR	319	319	313	318	313	328	323	324	315	313	318.5
(RBF-global)	3.1005	3.0967	3.1048	3.0868	3.0980	3.0645	3.1281	3.0067	3.0468	3.1160	

the trials of NOMAD and GA yielded a feasible sample point within the budget of $20d = 4440$ objective and constraint function evaluations (equivalent to $10d$ simulations). These results show the difficulty in obtaining a feasible sample point for the high-dimensional and highly constrained Mazda benchmark.

Next, Table 3 shows the number of simulations (including those for the initial population) it took the CONDOR algorithms to obtain a feasible sample point and the corresponding feasible objective function value for 10 trials on the Mazda benchmark. The last column of this table shows the mean of number of simulations needed to achieve feasibility for the algorithm. GA and NOMAD are not included since they did not produce any feasible sample point in any of the trials within the computational budget that is equivalent to $10d$ simulations.

Table 3 shows how quickly and consistently the CONDOR algorithms are able to obtain a feasible sample point within a small number of iterations on the Mazda benchmark while neither NOMAD nor GA found a feasible point within the equivalent of $10d = 2220$ simulations. In fact, removing the initial population size of 300 from the mean values in Table 3 shows that it only takes an average of 25.8 and 18.5 simulations for CONDOR (RBF-local) and CONDOR

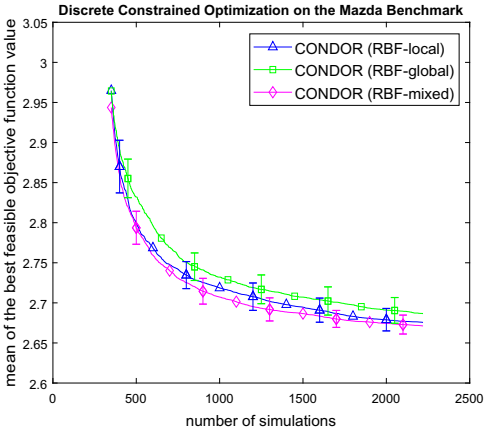

Fig. 2. Mean over 10 trials and 95% t confidence intervals for the best feasible objective value found by Two-Phase CONDOR algorithms at different computational budgets.

(RBF-global), respectively, to find a feasible point. Also, the global variant of Two-Phase CONDOR found a feasible point in less simulations than the local variant on the Mazda benchmark. Recall that the global variant found a feasible point in slightly less function evaluations than the local variant on G9-Discrete and G19-Discrete. This indicates that a more global search strategy for Phase I of the Two-Phase CONDOR might be more effective in achieving feasibility.

Next, Fig. 2 shows the mean of the best feasible objective function value over 10 trials obtained by the Two-Phase CONDOR algorithms at different computational budgets on the Mazda benchmark. As before, the error bars correspond to 95% t confidence intervals for the mean. The plots start at a budget of 350 simulations since all trials of the CONDOR algorithms have found a feasible sample point at this budget (see Table 3). Moreover, the mean of the best feasible objective function values (and standard error in parenthesis) obtained by the three variants after $10d = 2220$ simulations on the Mazda benchmark are 2.6757 (0.0060) for the Local variant, 2.6866 (0.0067) for the Global variant, and 2.6712 (0.0052) for the Mixed variant.

Figure 2 shows that the three variants of Two-Phase CONDOR have comparable performances with some slight advantage for CONDOR (RBF-mixed) in the best feasible objective function value obtained during Phase II. However, the earlier result in [26] comparing one trial each of the local and global variants of the original CONDOR when the feasible initial point from [16] is provided shows better performance for the local variant. These results, which are averaged over 10 trials, suggest that there is still some slight advantage for the local variant over the global variant on the Mazda benchmark. Recall that the global variant found a feasible point faster than the local variant. By using the search strategy of the global variant in Phase I and the search strategy of the local variant in Phase II, the mixed variant seems to perform slightly better than both the local

and global variants on the Mazda benchmark. This might be because the more global search strategy in Phase I provides more diverse sample points for building the surrogates that are used to approximate the objective and constraints in both feasible and infeasible regions of the search space. More accurate surrogates are helpful in making progress in finding a feasible point in Phase I and in improving the best feasible objective function value in Phase II.

4 Summary and Conclusions

This paper introduced the surrogate-based Two-Phase CONDOR algorithm, which extends the CONDOR method [26] for high-dimensional and computationally expensive constrained black-box optimization problems with discrete ordinal decision variables and many black-box inequality constraints. The original CONDOR algorithm requires a feasible initial point while Two-Phase CONDOR can be used even when such a feasible point is not available. In Phase I, the algorithm searches for a feasible point by using the surrogates of the constraints and objectives. In Phase II, the algorithm finds a solution that improves on the objective function value of the feasible point found in Phase I.

Three variants of Two-Phase CONDOR that employ cubic RBF surrogates are tested on discrete versions of three well-known artificial benchmark problems and also on the Mazda benchmark, which is based on a car structure design problem that has 222 discrete ordinal variables and 54 black-box constraints. The CONDOR (RBF-local) variant uses a more local search strategy for Phase I and Phase II, CONDOR (RBF-global) uses a more global search strategy for both phases, and CONDOR (RBF-mixed) uses a mixture of more global search in Phase I with more local search for Phase II. Numerical results show that the Two-Phase CONDOR algorithms required much less objective and constraint function evaluations to find feasible sample points compared to NOMAD and GA on the artificial test problems. Moreover, they generally obtained better feasible objective function values than NOMAD and GA on the artificial problems. In fact, some of the trials of GA did not even yield a feasible point within $80d$ objective and constraint function evaluations on these problems. On the Mazda benchmark, the results also show that the Two-Phase CONDOR algorithms consistently found a feasible sample point within a relatively small number of Phase I iterations while the GA and NOMAD algorithms did not find any feasible sample points in all of the 10 trials within $20d = 4440$ objective and constraint function evaluations (equivalent to $10d = 2220$ simulations).

In addition, the global and mixed variants of Two-Phase CONDOR required slightly less function evaluations than the local variant to find a feasible point on two of the three artificial problems and on the Mazda benchmark. This indicates that a global search strategy might be more effective for Phase I of the algorithm. However, the performances of the three variants are comparable in terms of the best feasible objective value obtained in Phase II. Overall, these results suggest that the Two-Phase CONDOR algorithm is promising for discrete expensive black-box optimization involving hundreds of variables and many black-box constraints when an initial feasible point is not provided.

Acknowledgements. Many thanks to Dr. Takehisa Kohira for explaining some of the details of the Mazda Benchmark.

References

1. Bagheri, S., Konen, W., Emmerich, M., Bäck, T.: Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. Appl. Soft Comput. **61**, 377–393 (2017)
2. Bartz-Beielstein, T., Zaefferer, M.: Model-based methods for continuous and discrete global optimization. Appl. Soft Comput. **55**, 154–167 (2017)
3. Bouhlel, M.A., Bartoli, N., Regis, R.G., Otsmane, A., Morlier, J.: Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. Eng. Optim. **50**(12), 2038–2053 (2018)
4. Currie, J., Wilson, D. I.: OPTI: lowering the barrier between open source optimizers and the industrial MATLAB user. In: Foundations of Computer-Aided Process Operations, Georgia, USA (2012)
5. Dong, H., Wang, P., Song, B., Zhang, Y., An, X.: Kriging-assisted Discrete Global Optimization (KDGO) for black-box problems with costly objective and constraints. Appl. Soft Comput. **94**, 106429 (2020)
6. Dushatskiy, A., Mendrik, A.M., Alderliesten, T., Bosman, P.A.N.: Convolutional neural network surrogate-assisted GOMEA. In: GECCO 2019: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 753–761 (2019)
7. Feliot, P., Bect, J., Vazquez, E.: A Bayesian approach to constrained single- and multi-objective optimization. J. Global Optim. **67**, 97–133 (2017)
8. Forrester, A.I.J., Sobester, A., Keane, A.J.: Engineering Design via Surrogate Modelling: A practical guide. Wiley, New York (2008)
9. Fukumoto, H., Oyama, A.: Benchmarking multiobjective evolutionary algorithms and constraint handling techniques on a real-world car structure design optimization benchmark problem. In: GECCO 2018: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 177–178 (2018)
10. Garrido-Merchán, E.C., Hernández-Lobato, D.: Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. Neurocomputing **380**, 20–35 (2020)
11. Kim, S.H., Boukouvala, F.: Surrogate-based optimization for mixed-integer nonlinear problems. Comput. Chem. Eng. **140**, 106847 (2020)
12. Kohira, T., Kemmotsu, H., Oyama, A., Tatsukawa, T.: Proposal of benchmark problem based on real-world car structure design optimization. In: GECCO 2018: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 183–184 (2018)
13. Le Digabel, S.: Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm. ACM Trans. Math. Softw. **37**(4), 44:1–44:15 (2011)
14. Liu, B., Sun, N., Zhang, Q., Grout, V., Gielen, G.: A surrogate model assisted evolutionary algorithm for computationally expensive design optimization problems with discrete variables. In: 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, pp. 1650–1657 (2016)
15. Luong, P., Gupta, S., Nguyen, D., Rana, S., Venkatesh, S.: Bayesian optimization with discrete variables. In: Liu, J., Bailey, J. (eds.) AI 2019. LNCS (LNAI), vol. 11919, pp. 473–484. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35288-2_38

16. Mazda Benchmark Problem. https://ladse.eng.isas.jaxa.jp/benchmark/index.html. Accessed 30 Apr 2020
17. Mueller, J., Shoemaker, C.A., Piche, R.: SO-I: a surrogate model algorithm for expensive nonlinear integer programming problems including global optimization applications. J. Global Optim. **59**, 865–889 (2014)
18. Nuñez, L., Regis, R.G., Varela, K.: Accelerated random search for constrained global optimization assisted by radial basis function surrogates. J. Comput. Appl. Math. **340**, 276–295 (2018)
19. Ohtsuka, H., Kaidan, M., Harada, T., Thawonmas, R.: Evolutionary algorithm using surrogate assisted model for simultaneous design optimization benchmark problem of multiple car structures. In: GECCO 2018: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 55–56 (2018)
20. Oyama, A., Kohira, T., Kemmotsu, H., Tatsukawa, T., Watanabe, T.: Simultaneous structure design optimization of multiple car models using the K computer. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, pp. 1–4 (2017)
21. Parr, J.M., Keane, A.J., Forrester, A.I., Holden, C.M.: Infill sampling criteria for surrogate-based optimization with constraint handling. Eng. Optim. **44**(10), 1147–1166 (2012)
22. Pelamatti, J., Brevault, L., Balesdent, M., Talbi, E.G., Guerin, Y.: Efficient global optimization of constrained mixed variable problems. J. Global Optim. **73**, 583–613 (2019)
23. Powell, M.J.D.: The theory of radial basis function approximation in 1990. In: Light, W. (ed.) Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions, pp. 105–210. Oxford University Press, Oxford (1992)
24. Regis, R.G.: Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. Comput. Oper. Res. **38**(5), 837–853 (2011)
25. Regis, R.G.: Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Eng. Optim. **46**(2), 218–243 (2014)
26. Regis, R.G.: Large-scale discrete constrained black-box optimization using radial basis functions. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 2924–2931 (2020). https://doi.org/10.1109/SSCI47803.2020.9308581
27. Regis, R.G.: A two-phase surrogate approach for high-dimensional constrained discrete multi-objective optimization. In: GECCO 2021: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 1870–1878 (2021)
28. The Mathworks: Matlab Global Optimization Toolbox (2018)
29. Zaefferer, M.: Surrogate Models for Discrete Optimization Problems. Ph.D. thesis, Technische Universität Dortmund (2018)