Covariance Matrix Adaptation Evolutionary Strategy with Worst-Case Ranking Approximation for Min–Max Optimization and its Application to Berthing Control Tasks

ANONYMOUS AUTHOR(S)

In this study, we consider a continuous min–max optimization problem $\min_{x\in\mathbb{X}}\max_{y\in\mathbb{Y}} f(x, y)$ whose objective function is a black-box. We propose a novel approach to minimize the worst-case objective function $F(x) = \max_y f(x, y)$ directly using a covariance matrix adaptation evolution strategy (CMA-ES) in which the rankings of solution candidates are approximated by our proposed worst-case ranking approximation (WRA) mechanism. We develop two variants of WRA combined with CMA-ES and approximate gradient ascent as numerical solvers for the inner maximization problem. Numerical experiments show that our proposed approach outperforms several existing approaches when the objective function is a smooth strongly convex–concave function and the interaction between x and y is strong. We investigate the advantages of the proposed approach for problems where the objective function is not limited to smooth strongly convex–concave functions. The effectiveness of the proposed approach is demonstrated in the robust berthing control problem with uncertainty.

CCS Concepts: • **Mathematics of computing → Continuous optimization**.

Additional Key Words and Phrases: Black-Box Min–Max Continuous Optimization, Covariance Matrix Adaptation Evolution Strategy, Nonconvex Nonconcave function, Robust Berthing Control Problem, Worst-Case Ranking Approximation

ACM Reference Format:

Anonymous Author(s). 2018. Covariance Matrix Adaptation Evolutionary Strategy with Worst-Case Ranking Approximation for Min–Max Optimization and its Application to Berthing Control Tasks. 1, 1 (September 2018), 27 pages. https://doi.org/XXXXXXX.XXXXXXX

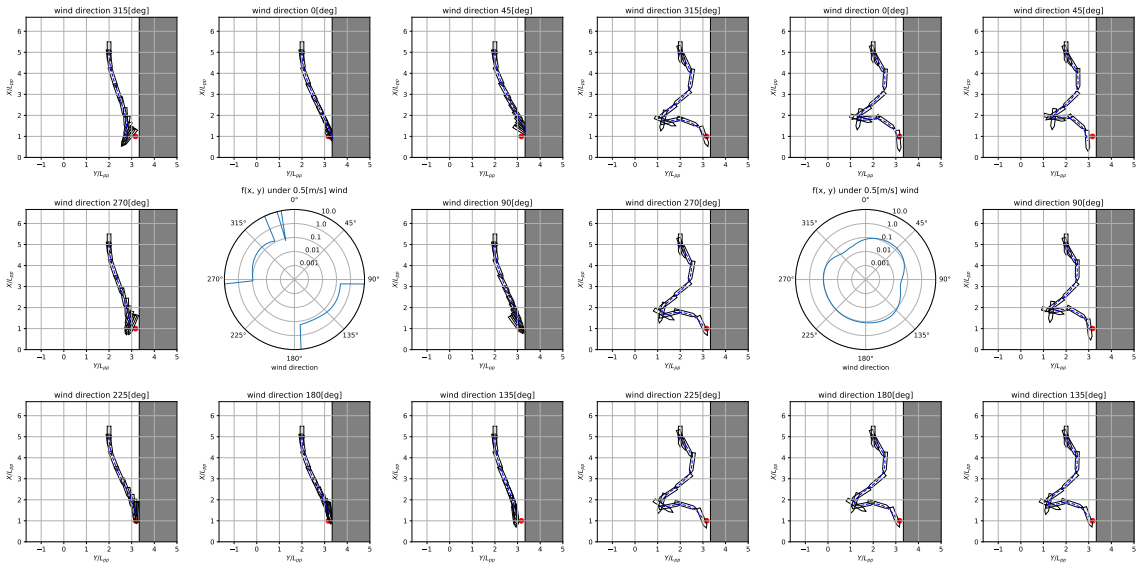
1 INTRODUCTION

Background. Simulation-based optimization is an attractive technique in various industrial fields. Given a design vector $x \in \mathbb{X} \subseteq R^m$, the objective function $h_{\text{sim}} : \mathbb{X} \to R$ is evaluated via numerical simulation. Simulation-based optimization has been used in several real-world applications, such as berthing control [Maki et al. 2020; Miyauchi et al. 2022], well placement [Bouzarkouna et al. 2012; Miyagi et al. 2018; Onwunalu and Durlofsky 2010], and topology design [Fujii et al. 2018; Marsden et al. 2004]. To perform simulation-based optimization, it is necessary to determine simulation conditions in advance so that the objective function values in the real-world, $h_{\text{real}}(x)$, are accurately computed. In other words, a simulator such that $h_{\text{sim}}(x) \approx h_{\text{real}}(x)$ must be developed. However, owing to some real-world uncertainties, the predetermined conditions often contain errors and hence $h_{\text{sim}}(x)$ does not approximate $h_{\text{real}}(x)$ well. In such situations, there is a risk that the optimal solution obtained in simulation-based optimization, $x_{\text{sim}} = \text{argmin}_{x\in\mathbb{X}} h_{\text{sim}}(x)$, does not perform well in the real-world and results in $h_{\text{real}}(x_{\text{sim}}) \gg h_{\text{sim}}(x_{\text{sim}})$.

One approach to find a robust solution is to formulate the problem as a min–max optimization problem

$$\min_{x\in\mathbb{X}} \max_{y\in\mathbb{Y}} f(x, y), \tag{1}$$

Manuscript submitted to ACM



(a) Controller optimized for no wind condition (b) Controller optimized for the worst wind condition

Fig. 1. Visualization of trajectories obtained by controllers for (a) no wind condition and (b) the worst wind condition with a 0.5-[m/s] maximum wind velocity. Center: the objective function values $f(x, y)$ under y representing a wind velocity of 0.5 [m/s] and varying wind direction presented by the polar axis. An objective function value smaller than 10 implies that the ship is controlled without a collision with the berth. The others: trajectories observed under wind for $0, 45, \ldots, 315$ [deg] with a velocity of 0.5 [m/s]. The red points are the target positions. The controllers were obtained in [Akimoto et al. 2022].

where $f(x, y)$ denotes the objective function and $y \in \mathbb{Y} \subseteq R^n$ is a parameter vector for the simulation conditions, called a scenario vector in this study, and is uncertain at the optimization stage. This approach aims to find the global min–max solution $x^* = \mathrm{argmin}_{x \in \mathbb{X}} F(x)$ that minimizes the worst-case objective function $F(x) := \max_{y \in \mathbb{Y}} f(x, y)$. In this formulation, the simulator designed by an expert engineer, h_{sim}, corresponds to $f(\cdot, y_{\mathrm{sim}})$ with a scenario vector $y_{\mathrm{sim}} \in \mathbb{Y}$, and the real-world objective, h_{real}, corresponds to $f(\cdot, y_{\mathrm{real}})$ with a scenario vector $y_{\mathrm{real}} \in \mathbb{Y}$, which is unknown and may change over time. Minimizing the worst-case objective function F minimizes the upper bound of the objective function values in the real-world, i.e., $F(x) \geqslant f(x, y_{\mathrm{real}})$ provided $y_{\mathrm{real}} \in \mathbb{Y}$.

In this study, we propose a novel approach for simulation-based optimization to obtain a robust solution against uncertainties involved in numerical simulations. We assume that the gradient of the objective function f with respect to x and y is unavailable, and that the objective function f and worst-case objective function F are nonexplicit (black-box functions). We refer to such a problem as a black-box min–max optimization. We focus on the following two types of problems, for which existing approaches [Akimoto et al. 2022; Liu et al. 2020] for the black-box min–max optimization fail to converge or exhibit slow convergence. (A) f is smooth and strongly convex–concave around x^*, but a strong interaction between x and y exists. (B) f is not smooth or strongly convex–concave around x^*. The objective function with respect to y in real-world applications is often nonconcave [Bertsimas et al. 2010b; Razaviyayn et al. 2020], i.e., falls into the latter situation.

Robust berthing control problem. As an example of real-world applications, we consider the automatic berthing control problem [Maki et al. 2020; Miyauchi et al. 2022]. The objective is to obtain a controller that realizes a fine control

of a ship toward a target state with the least collision risk and minimum elapsed time. Given a controller parameterized by x, a trajectory of the ship motion is simulated by numerically solving a ship maneuvering model. The objective function values are evaluated based on the computed trajectory. However, this simulation contains some uncertainties.

In a previous study [Akimoto et al. 2022], the problem of finding a robust berthing controller was formulated as a min–max optimization problem. Figure 1 shows the importance of considering the uncertainties. Figure 1a and Figure 1b, respectively, show the trajectories and objective function values under different wind conditions when a controller optimized by the (1+1)-covariance matrix adaptation evolution strategy (CMA-ES) [Arnold and Hansen 2010] under no wind disturbance and a controller optimized by Adversarial CMA-ES (`ADV-CMA-ES` [Akimoto et al. 2022]), where the uncertainty of wind direction ($[0, 360]$ [deg]) and wind speed ($[0, 0.5]$ [m/s]) are considered. When the controller optimized under the no wind assumption is used, we often observe the collision of the ship and berth under wind disturbance of a velocity of 0.5 [m/s]. Meanwhile, the robust controller obtained by `ADV-CMA-ES` successfully avoids collision for all wind directions.

The robust berthing control problem is expected to fall into Type (B). Its vestiges can be seen in Figure 1. The central figure shows that the objective function $f(x, y)$ is multimodal with respect to the wind direction. Therefore, $f(x, y)$ is nonconcave in terms of y. Moreover, we observed in our preliminary experiments that the worst-case scenario switches between offshore-to-berth wind y_{sea} and berth-to-offshore wind y_{land}. This is explained as follows. In offshore-to-berth wind y_{sea}, the optimum control avoids getting too close to the berth to avoid a collision. Under such control, berth-to-offshore wind y_{land} becomes the worst-case scenario because the ship stops at a position far from the target position near the berth, resulting in a high objective function value. Conversely, if the optimum control for y_{land} is operated, the worst-case scenario is y_{sea}, which causes the ship to collide with the berth. Therefore, the control that minimizes the objective function at the worst-case scenario is expected to exist on the boundary of the regions where the worst-case scenario changes between y_{land} and y_{sea}, and it is not the optimal solution under either scenario.

Related works. Recently, gradient-based min–max optimization has been actively studied. However, most existing studies investigate the min–max problems of functions that are concave in y, although several real-world problems are not necessarily concave in y [Razaviyayn et al. 2020]. In addition, a general nonconvex nonconcave min–max problem is theoretically intractable [Daskalakis et al. 2021]. Some studies have been conducted to identify the structures of nonconcave min–max problems that make it efficiently solvable [Diakonikolas et al. 2021; Liu et al. 2021; Nouiehed et al. 2019; Vlatakis-Gkaragkounis et al. 2021] or to exploit a small domain for scenario vectors [Ostrovskii et al. 2021]. These studies do not cover Type (B). Gradient-based approaches for general nonconcave min–max problems, where both implementation error and parameter uncertainty are considered, have been developed in previous studies [Bertsimas et al. 2010a,b]. However, this approach is designed to exploit the existence of implementation error, and it is not trivial to extend it to derivative-free situations via gradient approximation.

Derivative-free approaches for min–max optimization include coevolutionary approaches [Al-Dujaili et al. 2019; Barbosa 1999; Herrmann 1999; Qiu et al. 2018], simultaneous descent–ascent approaches [Akimoto et al. 2022; Liu et al. 2020], and surrogate-model-based approaches [Bogunovic et al. 2018]. Particularly, `ADV-CMA-ES` [Akimoto et al. 2022] and `ZO-Min-Max` [Liang and Stokes 2019] are theoretically guaranteed to converge to the optimal solution and its neighborhood, respectively, in smooth strongly convex concave min–max problems. Nevertheless, they fail to converge in Type (B) and exhibit slow convergence in Type (A) [Akimoto et al. 2022]. Although some coevolutionary approaches, such as minimax differential evolution [Qiu et al. 2018], are intended to address the difficulty in Type (B), they fail to converge not only on such problems but also on strongly convex–concave problems [Akimoto et al. 2022]. STABLEOPT

[Bogunovic et al. 2018], a Bayesian optimization approach, is expected to address the difficulty in Type (B). However, because of the high computational time of the Gaussian process, it is impractical for problems where numerous f-calls are required to obtain satisfactory solutions, according to [Liu et al. 2020].

Contribution. The contributions of this study are as follows.

Approach (Section 5). Aiming at addressing the limitations of existing approaches observed in Types (A) and (B), a novel approach that directly searches for the global min–max solution is proposed. The proposed approach minimizes the worst-case objective function using CMA-ES [Akimoto and Hansen 2020; Hansen and Auger 2014; Hansen and Ostermeier 2001] wherein the rankings of solution candidates are approximated by our proposed worst-case ranking approximation (WRA) mechanism. The WRA mechanism approximately solves the maximization problem $\max_{y\in\mathbb{Y}} f(x, y)$ for each solution candidate x. To save f-calls required to solve each maximization problem, we design a warm starting strategy and an early stopping strategy. We propose two variants of WRA implementations using CMA-ES and approximate gradient ascent (AGA) as inner solvers. To consider nonconvex real-world applications, we incorporate a restart strategy and a local search strategy.

Evaluation (Section 6). We designed 11 test problems with different characteristics, including both Types (A) and (B). Numerical experiments on the 11 test problems reveal the limitations of existing approaches and show that the proposed approach can handle both Types (A) and (B). We experimentally show that the scaling of the runtime on a smooth strongly convex–concave with respect to the interaction term (denoted by b in Section 6) is significantly improved over existing approaches. To understand when the proposed approach is effective in Type (B), we investigate the effect of each component of the WRA mechanism on each of the following situations: (S) the global min–max solution x^* is a strict min–max saddle point, (W) x^* is a weak min–max saddle point, and (N) x^* is not a min–max saddle point.

Application (Section 7). The proposed approach and existing approaches are applied to the robust berthing control problem with three types of scenario vectors. In the cases where the wind direction is included in a scenario vector, we confirm that the proposed approach often obtains controllers that can avoid collision with the berth in the worst-case scenario, whereas the controllers obtained by the existing approaches tend to fail to avoid collision with the berth in the worst-case scenario. In the case where an existing approach can often obtain collision-free controllers, we confirm that the existing approach achieves better worst-case performance than the proposed approach. We also demonstrate the effect of a hybrid of the existing and proposed approaches.

Implementation. Our implementations are publicly available[1].

2 PRELIMINARIES

The objective of this study is to find the global minimum solution to the worst-case objective function F defined as follows:

$$F(x) = \max_{y\in\mathbb{Y}} f(x, y), \tag{2}$$

where $f : \mathbb{X}\times\mathbb{Y} \to R$ denotes the objective function, $\mathbb{X} \subseteq R^m$ denotes the search domain for *design vector* x, and $\mathbb{Y} \subseteq R^n$ denotes the search domain for *scenario vector* y. We assume that f and F are black boxes and their gradient information and higher order information are unknown. For each $x \in \mathbb{X}$, let $\hat{Y}(x) = \{y \mid F(x) = f(x, y)\} = \operatorname{argmax}_{y\in\mathbb{Y}} f(x, y)$ be the *worst-case scenario set*. If $\hat{Y}(x)$ is a singleton, i.e., $|\hat{Y}(x)| = 1$, then the unique element is called the *worst-case scenario*

[1] Hidden-for-blind-review

and is denoted by $\hat{y}(x)$. The global minimum solution of F is called the *global min–max solution* of f and is denoted by $x^* = \mathrm{argmin}_{x \in \mathbb{X}} F(x)$.

The *min–max saddle point* (Definition 2.1) is an essential concept that characterizes the difficulties in obtaining x^*. In what follows, a neighborhood of design vector x is a set $\mathcal{E}_x$ such that there exists an open ball $\mathbb{B}(x, r) = \{\tilde{x} \in R^m \mid \|x - \tilde{x}\| < r\}$ included in $\mathcal{E}_x$ as a subset. We analogously define a neighborhood $\mathcal{E}_y$ of scenario vector y. A critical point $(x, y) \in \mathbb{X} \times \mathbb{Y}$ of the objective function f is such that $\nabla f(x, y) = (\nabla_x f(x, y), \nabla_y f(x, y)) = 0$.

Definition 2.1 (min–max saddle point)**.** A point $(\tilde{x}, \tilde{y}) \in \mathbb{X} \times \mathbb{Y}$ is a local min–max saddle point of a function $f : \mathbb{X} \times \mathbb{Y} \to R$ if there exists a neighborhood $\mathcal{E}_x \times \mathcal{E}_y \subseteq \mathbb{X} \times \mathbb{Y}$ including $(\tilde{x}, \tilde{y})$ such that for any $(x, y) \in \mathcal{E}_x \times \mathcal{E}_y$, the condition $f(\tilde{x}, y) \leqslant f(\tilde{x}, \tilde{y}) \leqslant f(x, \tilde{y})$ holds. If $\mathcal{E}_x = \mathbb{X}$ and $\mathcal{E}_y = \mathbb{Y}$, the point $(\tilde{x}, \tilde{y})$ is called the global min–max saddle point. If the equality holds only if $(x, y) = (\tilde{x}, \tilde{y})$, it is called a strict min–max saddle point. A saddle point that is not a strict min–max saddle point is called a weak min–max saddle point.

We focus on whether x^* is a strict min–max saddle point. If so, the problem of locating x^* turns into the problem of locating the global min–max saddle point. In such a situation, locating multiple local min–max saddle points, $\{(x_i, y_i)\}_{i=1}^K$, and selecting the best, $\mathrm{argmin}_{1 \leqslant i \leqslant K} \max_{1 \leqslant j \leqslant K} f(x_i, y_j)$, can offer the optimal solution x^* provided the global min–max saddle point is included in $\{(x_i, y_i)\}_{i=1}^K$. Existing approaches [Akimoto et al. 2022; Liu et al. 2020] for locating local min–max saddle points may be used for this purpose. However, if x^* is not a strict min–max saddle point, the above approach may not provide a reasonable solution; a different approach is required.

3 TEST PROBLEMS

Table 1 lists the test problems used in our experiments. Although it is difficult to formally frame our target problems, this list provides examples of problems in the scope of this study. We describe the characteristics of these problems in the following.

The functions f_3, f_5, f_6, f_7, f_8, and f_{11} are strictly convex–concave. On such problems, the worst-case scenario set $\hat{Y}(x)$ is a singleton for each $x \in \mathbb{X}$. The global min–max solution x^* is the strict global min–max saddle point $(x^*, \hat{y}(x^*))$. Particularly, f_5, f_6, and f_{11} are strongly convex–concave, and f_5, f_7, and f_{11} are smooth. f_5 and f_{11} are both smooth and strongly convex–concave, where the convergence of the existing approaches is investigated. Different from f_5 and the other functions, f_{11} is designed to be highly ill-conditioned in y to demonstrate the impact of ill-conditioning. Although x^* is a strict global min–max saddle point, based on our experiments, the existing approaches fail to converge if the objective function is nonsmooth (f_6 and f_8) or exhibit slow convergence if the objective function is not strongly convex–concave (f_7).

The functions f_1, f_2, and f_3 are convex–linear. On such problems, the worst-case scenario is typically located at the boundary of the scenario domain $\mathbb{Y}$. On f_1 and f_2, where the former is bilinear and the latter is strongly convex in x, the global min–max solution x^* forms a weak min–max saddle point (x^*, y) for any $y \in \mathbb{Y}$. Hence, the worst-case scenario set at x^* is $\hat{Y}(x^*) = \mathbb{Y}$, but the worst-case scenario $\hat{y}(x)$ in a neighborhood of x^* is one of the 2^n vertices $\bar{Y}$ of $\mathbb{Y}$. However, x^* is not a global minimum point of $f(x, y)$ for any $y \in \bar{Y}$. For f_3, $|\hat{Y}(x^*)| = 1$ and $(x^*, \hat{y}(x^*))$ is a strict global min–max saddle point.

The functions f_4, f_9, and f_{10} are not convex–concave. On these problems, the global min–max solution x^* does not form a min–max saddle point. Similar to f_1 and f_2, the worst-case scenarios of f_4 are located at the vertices $\bar{Y}$ of $\mathbb{Y}$. However, different from f_1 and f_2, $\hat{Y}(x^*) = \bar{Y}$ and x^* does not form a min–max saddle point in f_4. For f_9, the worst-case scenarios are not at the vertices of $\mathbb{Y}$ but at some specific points inside $\mathbb{Y}$, and $|\hat{Y}(x^*)| > 1$. These two

Table 1. Test problem definitions and their worst-case scenarios. The search domains for x and y are $\mathbb{X} = [\ell_x, u_x]^m$ and $\mathbb{Y} = [-b_y, b_y]^n$, respectively. The interaction between x and y is controlled by $m \times n$ matrix B. For f_3, we assume that B is of full column rank, and let $B^\dagger = [b_1^\dagger, \ldots, b_m^\dagger]$ be the Moore–Penrose inverse of B, $\alpha = -\frac{\min(|u_x|, |\ell_x|)}{(30/7)\max_{i=1,\ldots,m}\|b_i^\dagger\|_1}$, and $\gamma > 0$. For f_9, we set $n_* = \min\{n, 3\}$. For f_{10}, we assume $m = n$ and $B = \mathrm{diag}(1, \ldots, 1)$. The optimal solutions for the worst-case objective functions are $x^* = \mathbf{0}$, except for f_1, f_3, and f_9. The optimal solution is $[Bx^*]_i = 0$ for f_1, $[Bx^*]_i = \alpha$ for f_3, and $[B^\mathrm{T}x^*]_i = -\sinh(1)$ for $i \leqslant n_*$ and $[B^\mathrm{T}x^*]_i = 0$ for $i > n_*$ for f_9. Here, $[x]_i$ denotes the ith coordinate of a vector x.

Definition	$[\hat{y}(x)]_i$ (here $z = B^\mathrm{T}x$ for short)
$f_1 = x^\mathrm{T}By$	$\begin{cases} b_y \operatorname{sign}([z]_i) & [z]_i \neq 0 \\ \text{arbitrary} & [z]_i = 0 \end{cases}$
$f_2 = \frac{1}{2}\|x\|_2^2 + x^\mathrm{T}By$	$\begin{cases} b_y \operatorname{sign}([z]_i) & [z]_i \neq 0 \\ \text{arbitrary} & [z]_i = 0 \end{cases}$
$f_3 = \frac{1}{2}\|B^\mathrm{T}x - (\alpha - \gamma b_y)\mathbf{1}_n\|_2^2 + \gamma x^\mathrm{T}By$	$\begin{cases} b_y \operatorname{sign}([z]_i) & [z]_i \neq 0 \\ \text{arbitrary} & [z]_i = 0 \end{cases}$
$f_4 = \frac{1}{2}\|x\|_2^2 + x^\mathrm{T}By + \frac{1}{2}\|y\|_2^2$	$\begin{cases} b_y \operatorname{sign}([z]_i) & [z]_i \neq 0 \\ \pm b_y & [z]_i = 0 \end{cases}$
$f_5 = \frac{1}{2}\|x\|_2^2 + x^\mathrm{T}By - \frac{1}{2}\|y\|_2^2$	$\begin{cases} [z]_i & \lvert[z]_i\rvert \leqslant b_y \\ b_y \operatorname{sign}([z]_i) & \lvert[z]_i\rvert > b_y \end{cases}$
$f_6 = \frac{1}{2}\|x\|_2^2 + \|x\|_1 + x^\mathrm{T}By - \|y\|_1 - \frac{1}{2}\|y\|_2^2$	$\begin{cases} 0 & \lvert[z]_i\rvert \leqslant 1 \\ [z]_i - \operatorname{sign}([z]_i) & 1 < \lvert[z]_i\rvert \leqslant b_y + 1 \\ b_y \operatorname{sign}([z]_i) & b_y + 1 < \lvert[z]_i\rvert \end{cases}$
$f_7 = \frac{1}{4}\|x\|_2^4 + x^\mathrm{T}By - \frac{1}{4}\|y\|_2^4$	$\begin{cases} \frac{[z]_i}{\|z\|_2^{2/3}} & \frac{[z]_i}{\|z\|_2^{2/3}} \leqslant b_y \\ b_y \operatorname{sign}([z]_i) & \frac{[z]_i}{\|z\|_2^{2/3}} > b_y \end{cases}$
$f_8 = \|x\|_1 + x^\mathrm{T}By - \|y\|_1$	$\begin{cases} 0 & \lvert[z]_i\rvert \leqslant 1 \\ b_y \operatorname{sign}([z]_i) & \lvert[z]_i\rvert > 1 \end{cases}$
$f_9 = \sum_{i=1}^{n_*}\left([B^\mathrm{T}x]_i + \exp(\operatorname{sign}([y]_i)) \cdot \sin\left(\frac{\pi[y]_i}{b_y}\right)\right)^2 + \sum_{i=n_*+1}^{n}\left([B^\mathrm{T}x]_i^2 - [y]_i^2\right)$	$\begin{cases} (b_y/2) & [z]_i \geqslant -\sinh(1)\ \&\ i \leqslant n_* \\ -(b_y/2) & [z]_i \leqslant -\sinh(1)\ \&\ i \leqslant n_* \\ 0 & i > n_* \end{cases}$
$f_{10} = \|x^\mathrm{T}B\|_2^2 - 2\|y - x^\mathrm{T}B\|_2^2$	$\begin{cases} [z]_i & \lvert[z]_i\rvert \leqslant b_y \\ b_y \operatorname{sign}([z]_i) & \lvert[z]_i\rvert > b_y \end{cases}$
$f_{11} = \sum_{i=1}^{n}\left(\frac{1}{2}[x]_i^2 + 10^{-3i/n}[x^\mathrm{T}B]_i[y]_i - \frac{10^{-6i/n}}{2}[y]_i^2\right)$	$\begin{cases} 10^{3i/n}[z]_i & \lvert10^{3i/n}[z]_i\rvert \leqslant b_y \\ b_y \operatorname{sign}([z]_i) & \lvert10^{3i/n}[z]_i\rvert > b_y \end{cases}$

functions are multimodal in y for each $x \in \mathbb{X}$, and the global maximum (i.e., the worst-case scenario) changes depending on x. Different from f_4 and f_9, f_{10} is concave in both y and x. Because of the concavity in y, we have $|\hat{Y}(x)| = 1$ for all $x \in \mathbb{X}$. Moreover, $\hat{y}(x)$ is continuous. The worst-case objective function F is convex around x^*. However, x^* is not a min–max saddle point.

We focus on some characteristics related to the difficulty in approximating the local landscape of the worst-case objective function F. A characteristic common to f_1–f_4 and f_9 is that the worst-case scenario changes discontinuously. Particularly for f_1, f_2, f_4, and f_9, the worst-case scenarios spread over multiple distant points in a neighborhood of the global min–max solution x^*. The landscape of F cannot be approximated well around such a discontinuous point if we only have a single candidate $\tilde{y}$ of the corresponding worst-case scenario. We expect from Figure 1 that the robust berthing control problem discussed in Section 1 has the above difficulty. The landscape of F cannot be approximated well with a single candidate $\tilde{y}$ on f_{10} as well because of the concavity of f_{10} in x. The nonsmoothness of f_6 and f_8 in y

can also cause a difficulty in approximating $F(x)$ in a neighborhood of x^* by $f(x, \tilde{y})$ with a single candidate $\tilde{y}$. We expect that the landscape of F is easier to approximate for smooth convex–concave functions such as f_5, f_7, and f_{11}. However, if the worst-case scenario $\hat{y} : x \to \hat{y}(x)$ is continuous yet very sensitive, then approximating the landscape of F with a single candidate $\tilde{y}$ will be unreasonable. Such sensitivity is controlled by B in the test problem definition. The greater the greatest singular value of B is, the more sensitive the worst-case scenario is. In these situations, approximating the landscape of $F(x)$ locally around some point $\bar{x}$ by $f(x, \tilde{y})$ with a single candidate $\tilde{y} \approx \hat{y}(\bar{x})$ is inadequate.

4 LIMITATIONS OF EXISTING APPROACHES

As mentioned in Section 1, `ZO-Min-Max` [Liu et al. 2020] and `ADV-CMA-ES` [Akimoto et al. 2022] are promising approaches for the black-box min–max optimization. Both approaches are designed to converge to a strict local min–max saddle point $(\tilde{x}, \tilde{y})$. Let (x^t, y^t) be a pair of the solution candidate and the scenario candidate at iteration t. These approaches update it as

$$(x^{t+1}, y^{t+1}) = (x^t, y^t) + (\eta_x \cdot B_x, \eta_y \cdot B_y), \tag{3}$$

where η_x and η_y denote the learning rates, and B_x and B_y denote the update vectors for x and y, respectively. In `ZO-Min-Max`, (B_x, B_y) comprises approximate gradients of the objective function, $(-\widehat{\nabla_x f}(x^t, y^t), \widehat{\nabla_y f}(x^t, y^t))$. The learning rates need to be tuned for each problem. In `ADV-CMA-ES`, (B_x, B_y) comprises $(\bar{x}^t - x^t, \bar{y}^t - y^t)$, where $\bar{x}^t$ and $\bar{y}^t$ are approximations of $\operatorname{argmin}_{x \in \mathbb{X}} f(x, y^t)$ and $\operatorname{argmax}_{y \in \mathbb{Y}} f(x^t, y)$, respectively, obtained using (1+1)-CMA-ES [Arnold and Hansen 2010]. The learning rates are adapted during the optimization to alleviate tedious parameter tuning.

The above two existing approaches are theoretically guaranteed to converge to the global min–max saddle point [Akimoto et al. 2022] or its neighborhood [Liu et al. 2020] when the objective function is twice continuously differentiable and globally strongly convex–concave. Because the global min–max solution x^* is the global min–max saddle point of f in such problems, there is convergence to x^* or its neighborhood. In particular, the authors of [Akimoto et al. 2022] showed sufficient conditions for linear convergence. Although the global convergence is not theoretically guaranteed, updating x and y alternately as in (3) is expected to converge to a local min–max saddle point if the objective function is a locally smooth and strongly convex–concave around the local min–max saddle point.

In addition, the authors of [Akimoto et al. 2022] reported several limitations of the above two existing approaches. Among them, we focus on the following two limitations in this study.

Difficulty (I): slow convergence on smooth strongly convex–concave problems. First, we discuss the slow convergence issue on smooth strongly convex–concave problems highlighted in [Akimoto et al. 2022]. For instance, consider a convex–concave quadratic problem $f_{ex}(x, y) = (a/2)x^2 + bxy - (c/2)y^2$. The worst-case scenario is $\hat{y}(x) = (b/c)x$ for each x and the optimal solution is $\hat{x}(y) = -(b/a)y$ for each y. It is intuitive that both $\hat{y}(x)$ and $\hat{x}(y)$ should not be too sensitive to follow their change by (3). In fact, it has been theoretically derived that, for linear convergence, the learning rate must be set as $\eta_x, \eta_y \in O(ac/(ac + b^2))$ and the required number of iterations to find near-optimal solution is $\Omega(1 + b^2/(ac))$; refer to [Akimoto et al. 2022] for details. A similar limitation has been reported for the simultaneous gradient descent–ascent (SGDA) approach [Liang and Stokes 2019]. The same limitation is expected to exist in `ZO-Min-Max` because it is regarded as an approximation of the SGDA approach. The adaptation of the learning rates in `ADV-CMA-ES` can mitigate the difficulty in tuning learning rates. However, it cannot avoid the slow convergence problem.

The situation is worse if the objective function is convex–concave but not strongly convex–concave. For example, consider f_7 with $n = m = 1$ and $B = b$. This objective function is similar to f_{ex}, but the coefficients are regarded as $a = (1/2)x^2$ and $c = (1/2)y^2$, i.e., decreasing as the solution approaches the global min–max saddle point $(x^*, y^* = \hat{y}(x^*))$. In this problem, the learning rate must converge to zero as the solution approaches (x^*, y^*). This jeopardizes the advantage of the existing approaches, i.e., linear convergence to the min–max saddle point. In fact, the authors of [Akimoto et al. 2022] reported such an issue empirically.

Difficulty (II): nonconvergence to a min–max solution that is not a strict min–max saddle point. Next, we discuss the nonconvergence issue on problems where x^* is not a strict min–max saddle point. The existing approaches fail to converge to x^*. Such a situation occurs when the objective function is not strictly convex–concave. The situations can be categorized into two: (W) x^* is a weak min–max saddle point and (N) x^* is not a min–max saddle point. Among the test problems in Table 1, f_1 and f_2 fall into Category (W), and f_4, f_9, and f_{10} fall into Category (N). A numerical experiment in [Akimoto et al. 2022] has shown that `ADV-CMA-ES` fails to converge to x^* on such problems. A theoretical investigation in [Liang and Stokes 2019] has shown that SGDA fails to converge as well. Therefore, `ZO-Min-Max` is also expected to fail. The authors of [Liang and Stokes 2019] reported that with some modifications, SGDA can converge to the weak global min–max saddle point on bilinear functions. The existing approaches may tackle problems of Category (W) by incorporating such a modification. However, problems of Category (N) cannot be solved.

In our experiments, we also confirmed that there exists a situation where the existing approaches fail to converge even if x^* is a strict global min–max saddle point. Example functions are f_3, f_6, and f_8, which are strictly convex–concave but nonsmooth. The situation where x^* is a strict global min–max saddle point but f is nonsmooth is denoted as Category (S).

Direction to address Difficulties (I) and (II). One approach to avoid Difficulty (II) is to approximate the worst-case objective function F by solving $\max_{y \in \mathbb{Y}} f(x, y)$ numerically and optimize it directly. If F can be approximated well for each $x \in \mathbb{X}$, i.e., $\max_{y \in \mathbb{Y}} f(x, y)$ can be solved efficiently for each x, and F can be globally optimized efficiently by a numerical solver, it does not matter whether x^* is a min–max saddle point or not. Therefore, Difficulty (II) can be addressed naturally.

We also expect that there can be a solution to Difficulty (I). Because any smooth strongly convex–concave function can be approximated by a quadratic convex–concave function around the global min–max saddle point, we focus on f_{ex} for simplicity. Its worst-case objective function is $F(x) = \frac{1}{2}(a + b^2/c)x^2$. Because it is a convex quadratic function, a reasonable solver converges linearly to its global minimum point x^*. For $n > 1$ and $m > 1$, the worst-case objective function can be ill-conditioned. However, if we employ a solver that uses second-order information, such as CMA-ES [Akimoto and Hansen 2020; Hansen and Auger 2014; Hansen and Ostermeier 2001], we expect that it can be solved efficiently. Therefore, the number of f-calls spent by the approach that directly optimizes F is expected to be less sensitive to the interaction term. If the objective function is smooth and weakly convex–concave, this argument does not hold. However, considering the aforementioned example f_7, we have $F(x) = (1/4)x^4 + (3/4)(bx)^{4/3}$, which is nonsmooth at $x^* = 0$ but strongly convex. Therefore, we expect that a comparison-based approach, invariant to any increasing transformation of the objective function, can solve it efficiently.

5 PROPOSED APPROACH

We propose a novel approach to address Difficulties (I) and (II). The main idea is to directly minimize the worst-case objective function F. The bottleneck of directly minimizing F in the black-box min–max optimization setting is

the computational time for each $F(x)$ evaluation, which is required to solve maximization problem $\max_{y\in\mathbb{Y}} f(x,y)$ approximately. To tackle this bottleneck, we propose to employ CMA-ES to minimize F (Section 5.2), and propose the WRA mechanism that approximates the ranking of $\{F(x_i)\}_{i=1}^{\lambda_x}$ for the given solution candidates $\{x_i\}_{i=1}^{\lambda_x}$ (Section 5.1).

For the proposed approach to work effectively, we suppose (a) $|\hat{Y}(x)| = 1$ and $\hat{y}(x)$ is continuous almost everywhere in $\mathbb{X}$, (b) the solver for the inner maximization problem can globally maximize $f(x,y)$ with respect to y efficiently for each $x \in \mathbb{X}$, and (c) the solver for the outer minimization problem, CMA-ES in this study, can minimize F efficiently.

5.1 WRA

5.2 CMA-ES for outer minimization

The proposed approach tries to solve the outer minimization problem of (1) using CMA-ES. CMA-ES is a state-of-the-art derivative-free optimization approach for continuous black-box optimization problems [Hansen 2009; Hansen et al. 2010; Rios and Sahinidis 2013] and has been used in several real-world applications [Fujii et al. 2018; Maki et al. 2020; Miyagi et al. 2018; Tanabe et al. 2021; Urieli et al. 2011]. There are two essential characteristics of CMA-ES that attract attention. One is that it is a quasiparameter-free approach, i.e., one does not need any hyperparameter tuning except for a population size λ_x, which is desired to be increased if the problem is multimodal or noisy or if several CPU cores are available. Because the worst-case objective function F is a black-box and it is difficult to understand the characteristics of F in advance, the parameter-free nature is essential. The second is that it is parallel-implementation friendly. The objective function values (F in our case) of multiple solution candidates generated at an iteration can be evaluated in parallel. It is desired when the computational cost of the objective function evaluation is high. Because each evaluation of F is expensive as it requires solving maximization problem $\max_{y\in\mathbb{Y}} f(x,y)$ approximately, this is practically essential.

CMA-ES repeats the sampling, evaluation, and update steps until a termination condition is satisfied. Let $t \geqslant 0$ be the iteration counter. First, λ_x solution candidates $\{x_i\}_{i=1}^{\lambda_x}$ are generated independently from a Gaussian distribution $\mathcal{N}(m_x^t, \Sigma_x^t)$ with mean vector $m_x^t \in \mathbb{X}$ and covariance matrix $\Sigma_x^t \in R^{m\times m}$. Next, the worst-case objective function values of the λ_x solution candidates, $\{F(x_i)\}_{i=1}^{\lambda_x}$, are evaluated, and there rankings $\mathrm{Rank}_F(\{x_i\}_{i=1}^{\lambda_x})$ are computed, where the ith ranked solution candidate has the ith smallest F value. Finally, CMA-ES updates the distribution parameters, m_x^t and Σ_x^t, and other dynamic parameters using the solution candidates and their rankings. An important aspect of the update of CMA-ES is that it is comparison-based. That is, provided the rankings of the solution candidates, $\mathrm{Rank}_F(\{x_i\}_{i=1}^{\lambda_x})$, are computed, the worst-case objective function values, $\{F(x_i)\}_{i=1}^{\lambda_x}$, do not need to be accurately computed.

In this study, we implemented the version of CMA-ES proposed in [Akimoto and Hansen 2020], namely, dd-CMA-ES[2], as the default solver. The configuration of CMA-ES follows the default proposed procedure in [Akimoto and Hansen 2020]. If the search domain has a box constraint, we employ the mirroring technique along with upper-bounding the coordinate-wise standard deviation $\sqrt{[\Sigma_x^t]_{\ell,\ell}}$ for $\ell = 1,\dots,m$ [Yamaguchi and Akimoto 2018], where $[\Sigma_x^t]_{\ell,\ell}$ denotes the (ℓ,ℓ)-th element of Σ_x^t. The initial distribution parameters, m_x^0 and Σ_x^0, should be set problem-dependently. We terminate CMA-ES when $\max_\ell \sqrt{[\Sigma_x^t]_{\ell,\ell}} < V_{\min}^x$ is satisfied, where $V_{\min}^x$ is a problem-dependent threshold, or $\mathrm{Cond}(\Sigma_x) > \mathrm{Cond}_{\max}^x = 10^{14}$ is satisfied, where $\mathrm{Cond}(\Sigma_x)$ is the condition number, i.e., the ratio of the greatest to smallest eigenvalues, of Σ_x^t.

The proposed WRA mechanism approximates the rankings of solution candidates by roughly solving maximization problems $\max_{y\in\mathbb{Y}} f(x_i,y)$ for each solution candidate x_i. To save the inner f-calls to approximate the rankings $\mathrm{Rank}_F(\{x_i\}_{i=1}^{\lambda_x})$, we incorporate a warm starting strategy, where we try to start each maximization $\max_{y\in\mathbb{Y}} f(x_i,y)$

[2] https://gist.github.com/youheiakimoto/1180b67b5a0b1265c204cba991fa8518

with a good initial solution candidate and a good configuration of the inner solver (Section 5.2.1), and an early stopping strategy, where we try to stop each maximization $\max_{y\in\mathbb{Y}} f(x_i, y)$ once $\mathrm{Rank}_F(\{x_i\}_{i=1}^{\lambda_x})$, are considered well-approximated (Section 5.2.2). The overall framework is summarized in Algorithm 1.

Hereinafter, let $\mathcal{M}$ be a solver used to approximately solve $\max_{y\in\mathbb{Y}} f(x_i, y)$. Let ω represent the configurations of the solver $\mathcal{M}$ inherited over the WRA calls, and θ represent the other configurations that are not inherited.

5.2.1 Warm starting strategy. Two key ideas behind the design of our warm starting strategy are as follows.

First, we inherit the worst-case scenario candidates and the configurations from the last WRA call . The Gaussian distribution $\mathcal{N}(m^t, \Sigma^t)$ of CMA-ES for the outer minimization does not significantly change in one iteration. Then, the distribution of the worst-case scenarios for the solution candidates generated at iteration t is considered to be similar to that at iteration $t+1$. Therefore, we expect that using the solver configurations used at the last iteration will contribute to reduce the number of f-calls. This idea is expected to be effective for the problem where $|\hat{Y}(x)| = 1$ and $\hat{y}(x)$ is continuous almost everywhere in $\mathbb{X}$.

Second, we maintain N_ω ($\geqslant 1$) configurations. Consider situations (W) and (N) described in Section 3. The worst-case scenarios corresponding to solution candidates $\{x_i\}_{i=1}^{\lambda_x}$ generated in a single iteration may not be concentrated at one point but may be distributed around $|\hat{Y}(x^*)|$ distinct points even if $\{x_i\}_{i=1}^{\lambda_x}$ are concentrated around $x^* = \mathrm{argmin}_{x\in\mathbb{X}} F(x)$. If we only maintain one configuration, it may be a good initial configuration only for a small portion of $\{x_i\}_{i=1}^{\lambda_x}$. There is a high risk that F values are accurately estimated only for these candidates and they are underestimated for the others due to insufficient maximization. To address this difficulty, we maintain multiple configurations and try to keep them diverse.

Algorithm 1 WRA

Require: $x_1, \ldots, x_{\lambda_x}$
Require: $\{(y_k, \omega_k, p_k)\}_{k=1}^{N_\omega}$
Require: $\tau_{\text{threshold}}$, $p_{\text{threshold}}$, p_p, p_n
1: `// Warm starting`
2: **for** $i = 1, \ldots, \lambda_x$ **do**
3: evaluate $f(x_i, y_k)$ for all $k = 1, \ldots, N_\omega$
4: $k_i^{\text{worst}} = \mathrm{argmax}_{k\in\{1,\ldots,N_\omega\}} f(x_i, y_k)$
5: $\hat{y}_i = y_{k_i^{\text{worst}}}$, $\tilde{\omega}_i = \omega_{k_i^{\text{worst}}}$, and $F_i^0 = f(x_i, y_{k_i^{\text{worst}}})$
6: **end for**
7: `// Early stopping`
8: initialize $\tilde{\theta}_1, \ldots, \tilde{\theta}_{\lambda_x}$
9: **for** rd $= 1, 2, \ldots$ **do**
10: **for** $i = 1, \ldots, \lambda_x$ **do**
11: $F_i^{\text{rd}}, \hat{y}_i, \tilde{\omega}_i, \tilde{\theta}_i \leftarrow \mathcal{M}(F_i^{\text{rd}-1}, \hat{y}_i, \tilde{\omega}_i, \tilde{\theta}_i)$
12: **end for**
13: $\tau = \mathrm{Kendall}(\{F_i^{\text{rd}-1}\}_{i=1}^{\lambda_x}, \{F_i^{\text{rd}}\}_{i=1}^{\lambda_x})$
14: **break if** $\tau > \tau_{\text{threshold}}$
15: **end for**
16: `// Postprocessing`
17: $S^{\text{worst}} = \{k_i^{\text{worst}} \text{ for } i = 1, \ldots, \lambda_x\}$
18: **for** $\tilde{k} \in S^{\text{worst}}$ **do**
19: $\ell = \mathrm{argmin}_{i=1,\ldots,N_\omega} \{F_i^{\text{rd}} \mid k_i^{\text{worst}} = \tilde{k}\}$
20: $y_{\tilde{k}} = \hat{y}_\ell$, $\omega_{\tilde{k}} = \tilde{\omega}_\ell$
21: $p_{\tilde{k}} = \min(p_{\tilde{k}} + p_p, 1)$
22: **end for**
23: $p_k = p_k - p_n \mathbb{I}\{k \notin S^{\text{worst}}\}$ for all $k = 1, \ldots, N_\omega$
24: **for** $k = 1, \ldots, N_\omega$ **do**
25: refresh (y_k, ω_k, p_k) **if** $p_k < p_{\text{threshold}}$
26: **end for**
27: **return** $\{F_i^{\text{rd}}\}_{i=1}^{\lambda_x}$ and $\{(y_k, \omega_k, p_k)\}_{k=1}^{N_\omega}$ for the next call

These two ideas are implemented in our warm starting strategy. It comprises (1) selecting a good initial worst-case scenario candidate $\tilde{y}$ and configuration $\tilde{\omega}$ of solver $\mathcal{M}$ for each solution candidate x_i among N_ω pairs $\{(y_k, \omega_k)\}_{k=1}^{N_\omega}$ (Lines 1–6 in Algorithm 1) and (2) preparing N_ω pairs $\{(y_k, \omega_k)\}_{k=1}^{N_\omega}$ for the next WRA call (Lines 16–26 in Algorithm 1). For each x_i, we evaluate $f(x_i, y_k)$ for $k = 1, \ldots, N_\omega$ and select the worst-case scenario candidate. Let $k_i^{\text{worst}} = \mathrm{argmax}_k f(x_i, y_k)$ be the index of the worst-case scenario

candidate among $\{y_k\}_{k=1}^{N_\omega}$. Then, we select the configuration $\omega_{k_i^{\text{worst}}}$ of the solver that generated $y_{k_i^{\text{worst}}}$ as the initial configuration $\tilde{\omega}_i$ to search for the worst-case scenario for x_i. After approximating $\{F(x_i)\}_{i=1}^{\lambda_x}$, we update the set of configurations of the solver. Basically, we replace the selected configurations with the configurations obtained after the solver execution. If the same configuration is selected for different solution candidates, we replace the configuration with the one used for the solution candidate with the optimal approximated worst-case value.

Moreover, to avoid keeping used configurations, we refresh such configurations and try to have diverse configurations. For this purpose, we maintain a parameter $p_k \in (0, 1]$ for $k = 1, \ldots, N_\omega$ and initialize the parameter as 1. The parameter p_k is increased by p_p if kth configuration is selected, and decreased by p_n otherwise. Once we have $p_k \leqslant p_{\text{threshold}}$, the kth configuration and the corresponding worst-case scenario candidate are refreshed in the same manner as their initialization, and p_k is reset to 1.

5.2.2 Early stopping strategy. Our early stopping strategy is to save f-calls by terminating λ_x solvers once the rankings of the worst-case objective function values of the given solution candidates, $\text{Rank}_F(\{x_i\}_{i=1}^{\lambda_x})$, are regarded as well-approximated. The early stopping strategy is described at Lines 7–15 in Algorithm 1.

The main idea is as follows. As aforementioned, CMA-ES is a comparison-based approach. Therefore, the worst-case objective function values are not needed to be accurately estimated provided their rankings are computed. We further hypothesize that CMA-ES behaves similarly on the approximated rankings if the rankings of solution candidates are approximated with a high correlation to the true rankings, according to Kendall [Kendall and Gibbons 1990]. This hypothesis is often imposed in surrogate-assisted approaches and related approaches [Akimoto et al. 2020, 2019; Hansen 2019; Miyagi et al. 2021; Pitra et al. 2021] and is partly validated in theory [Akimoto 2022]. Because the true rankings of the worst-case objective function values are unknown, instead of trying to check the rank correlation between the true and approximate rankings, we keep track of changes in the rankings and stop if the change is regarded as sufficiently small.

To compute the rankings of the worst-case objective function values, λ_x solvers are run in parallel, and we periodically compute the rankings of the solution candidates using the approximated worst-case objective function values, $\{F_i^{\text{rd}} = f(x_i, \hat{y}_i)\}_{i=1}^{\lambda_x}$, where $\text{rd} \geqslant 0$ is the number of ranking computations so far and is called the round. After each round, we compute the Kendall's rank correlation between the current and last approximations of the rankings, $\tau(\{F_i^{\text{rd}-1}\}_{i=1}^{\lambda_x}, \{F_i^{\text{rd}}\}_{i=1}^{\lambda_x})$. If it is greater than the predefined threshold $\tau_{\text{threshold}} \geqslant 0$, we regard the rankings are well-approximated and terminate the solvers. A reasonable definition of a round of a solver call depends on the choice of the solver. We discuss the solver choice and round definition in the next section.

5.2.3 Hyperparameters. The hyperparameters for WRA are the threshold for Kendall's rank correlation $\tau_{\text{threshold}}$, number of configurations N_ω, threshold $p_{\text{threshold}}$, and parameters p_p and p_n for the refresh strategy. The initial configurations $\{\omega_k\}_{k=1}^{N_\omega}$ and initial worst-case scenario candidates $\{y_k\}_{k=1}^{N_\omega}$ must be set problem-dependently. We describe the impact of these hyperparameters.

Threshold $\tau_{\text{threshold}}$ should be set to a relatively high value to approximate $\text{Rank}_F(\{x_i\}_{i=1}^{\lambda_x})$ with high accuracy. However, setting a high value of $\tau_{\text{threshold}}$ (e.g., $\tau_{\text{threshold}} = 1$) has a risk of spending too many f-calls. Based on our parameter survey, we found that the performance of the proposed approach was not very sensitive to the change in $\tau_{\text{threshold}}$ on the test problems. Therefore, we set its default value as $\tau_{\text{threshold}} = 0.7$ and used this value throughout our experiments.

The number of configurations, N_ω, is desired to be set no smaller than the number $|\hat{Y}(x^*)|$ of worst-case scenarios around x^* to maintain good configurations and good initial solutions for each scenario in $\hat{Y}(x^*)$. In addition, because N_ω f-calls are required to select the initial configuration for each x, N_ω is desired to be as small as possible. However, $|\hat{Y}(x^*)|$ is unknown in advance and is problem-dependent. We suggest setting N_ω to be a few times greater than λ_x to allow λ_x solution candidates a chance to use λ_x distinct worst-case scenario candidates. The effect is discussed in Section 6.

The parameters $p_{\text{threshold}}$, p_p, and p_n affect the frequency of each configuration to be refreshed. If the configurations are frequently refreshed, our warm starting strategy may be less effective. Based on our preliminary investigation, we set $p_{\text{threshold}} = 0.1$, $p_p = 0.4$ and $p_n = 0.05$ as the default values. In this case, the configurations $\{\tilde{\omega}_i\}_{i=1}^{\lambda_x}$ are kept for at least $6 = (p_p - p_{\text{threshold}})/p_n$ outer loop iterations after the last use or $18 = (1 - p_{\text{threshold}})/p_n$ iterations after the initialization and last refresh.

5.3 Implementation of WRA

We implement two variants of WRA with CMA-ES (Section 5.3.1) and AGA (Section 5.3.2) as solvers $\mathcal{M}$.

5.3.1 WRA using CMA-ES. The first variant, summarized in Algorithm 2, uses dd-CMA-ES [Akimoto and Hansen 2020] as a solver $\mathcal{M}$. If the search domain has a box constraint, we employ the mirroring technique along with upper-bounding the coordinate-wise standard deviation [Yamaguchi and Akimoto 2018]. The configuration $\tilde{\omega}$ includes the mean vector $\tilde{m}$ and covariance matrix $\tilde{\Sigma}$, and $\tilde{\theta}$ includes other parameters such as evolution paths, iteration counter $t' \geqslant 0$ (initialized as $t' = 0$), and termination flag h (initialized as $h = \textsc{False}$).

Because the proposed approach is a double-loop approach, setting the termination conditions for the inner loop is crucial. Algorithm 2 runs CMA-ES until the worst-case scenario candidate is improved $c_{\max}$ times. If the worst-case scenario candidate is improved for $c_{\max}$ times, we regard that it is significantly improved. Similar to CMA-ES for outer minimization, we terminate the maximization process if all coordinate-wise standard deviations, $\sqrt{[\tilde{\Sigma}]_{\ell,\ell}}$, become smaller than $V_{\min}^y$. In this situation, we expect that the distribution is sufficiently concentrated and no more significant improvement will be obtained. We stop CMA-ES if the condition number, $\mathrm{Cond}(\tilde{\Sigma})$, becomes greater than $\mathrm{Cond}_{\max}^y$. If one of the latter two conditions is satisfied, we set $h = \textsc{True}$, and CMA-ES will not be executed in the following rounds in the current WRA call.

The distribution parameters are inherited over WRA calls. Once the condition $\max_\ell \sqrt{[\tilde{\Sigma}]_{\ell,\ell}} < V_{\min}^y$ is satisfied for some configurations, it is expected to be immediately satisfied in the next WRA call if these configurations are selected. However, because the objective function with respect to y, i.e., $f(x_i, y)$, differs as solution candidates x_i differ in each WRA call, there is a chance that the distribution will be enlarged due to the step-size adaptation mechanism of CMA-ES, and a significant improvement will be realized. Therefore, we force all coordinate-wise standard deviations to be no smaller than $V_{\min}^y$ once the greatest one becomes smaller than $V_{\min}^y$ (Lines 11–12) and CMA-ES to run at least $T_{\min}$ iterations for each WRA call.

The hyperparameters includes the initial configurations for inner CMA-ES ($\{m_k\}_{k=1}^{N_\omega}, \{\Sigma_k\}_{k=1}^{N_\omega}, \theta$), initial scenarios $\{y_k\}_{k=1}^{N_\omega}$, and termination conditions for Algorithm 2, $c_{\max}$, $V_{\min}^y$, and $T_{\min}$. The configuration and initialization of θ, including the initialization of evolution paths and population size λ_y, follow the values proposed in [Akimoto and Hansen 2020]. The parameter $c_{\max}$ impacts the approximation accuracy of the rankings on the worst-case objective function values $\mathrm{Rank}_F(\{x_i\}_{i=1}^{\lambda_x})$ and f-calls to approximate the rankings. If $c_{\max}$ is set to a greater value, WRA will require more f-calls. Meanwhile, setting $c_{\max}$ to a smaller value has a risk to terminate the scenario improvement before the ranking

on the worst-case objective function $\mathrm{Rank}_F(\{x_i\}_{i=1}^{\lambda_x})$ is estimated with sufficient accuracy. The parameter $T_{\max}$ can be set to a constant value, as CMA-ES can increase the standard deviation rapidly if it is desired. We set $T_{\max} = 10$ as the default value. The parameter $V_{\min}^y$ and initial distributions $\{(m_k, \Sigma_k)\}_{k=1}^{N_\omega}$ must be set problem-dependently. The initial scenarios $\{y_k\}_{k=1}^{N_\omega}$ are drawn from the initial distributions, i.e., $y_k \sim \mathcal{N}(m_k, \Sigma_k)$.

Algorithm 2 CMA-ES as $\mathcal{M}$

Require: $x, \hat{y}, F_y, \tilde{\omega} = (\tilde{m}, \tilde{\Sigma}), \tilde{\theta} = (h, t', \dots)$
Require: $V_{\min} > 0, c_{\max} \geqslant 1, \lambda_y = \lfloor 4 + 3\log(n) \rfloor$
$\tilde{\Sigma}_{\mathrm{init}} = \tilde{\Sigma}$, $c = 0$
while $c < c_{\max}$ **and** $h =$ False **do**
  Sample $\{\hat{y}'\}_{k=1}^{\lambda_y} \sim \mathcal{N}(\tilde{m}, \tilde{\Sigma})$
  Evaluate $f_k = f(x, \hat{y}_k')$ for all $k = 1, \dots, \lambda_y$
  Select the worst index $\tilde{k}^{\mathrm{worst}} = \mathrm{argmax}_{k=1,\dots,\lambda_y} f_k$
  if $f(x, \hat{y}'_{\tilde{k}^{\mathrm{worst}}}) > F_y$ **then**
    $F_y = \max_{k=1,\dots,\lambda_y} f_k$, $\hat{y} = \hat{y}'_{\tilde{k}^{\mathrm{worst}}}$, and $c = c + 1$
  end if
  Perform CMA-ES update using $\{\hat{y}_k', f_k\}_{k=1}^{\lambda_y}$
  if $\max_\ell \left\{\sqrt{[\tilde{\Sigma}]_{\ell,\ell}}\right\} < V_{\min}^y$ and $t' \geqslant T_{\min}$ **then**
    $D = \mathrm{diag}\left(\max\left(1, \frac{V_{\min}^y}{\sqrt{[\tilde{\Sigma}]_{1,1}}}\right), \dots, \max\left(1, \frac{V_{\min}^y}{\sqrt{[\tilde{\Sigma}]_{n,n}}}\right)\right)$
    $\tilde{\Sigma} = D\tilde{\Sigma}D$ and $h =$ True
  end if
  $h =$ True and set $\tilde{\Sigma} = \tilde{\Sigma}_{\mathrm{init}}$ **if** $\mathrm{Cond}(\tilde{\Sigma}) > \mathrm{Cond}_{\max}^y$
  $t' = t' + 1$
end while
return $\hat{y}, F_y, \tilde{\omega} = (\tilde{m}, \tilde{\Sigma}), \tilde{\theta} = (h, t', \dots)$

Algorithm 3 AGA as $\mathcal{M}$

Require: $x, \hat{y}, F_y, \tilde{\omega} = \tilde{\eta}, \tilde{\theta} = (h, \dots)$
Require: $U_{\min} > 0, c_{\max} \geqslant 1, \beta \in (0, 1)$
$c = 0$
while $c < c_{\max}$ **and** $h =$ False **do**
  Obtain approximated gradient $\bar{\nabla}_y f$ at $\hat{y}$
  $\hat{y}' = \hat{y} + \tilde{\eta}\bar{\nabla}_y f$
  if $f(x, \hat{y}') > F_y$ **then**
    $\tilde{\eta} = \tilde{\eta}/\beta$
  else
    while $f(x, \hat{y}') \leqslant F_y$ **do**
      $\tilde{\eta} = \tilde{\eta} \times \beta$
      $\hat{y}' = \hat{y} + \tilde{\eta}\bar{\nabla}_y f$
      $h =$ True **if** $\|\tilde{\eta}\bar{\nabla}_y f\|_\infty \leqslant U_{\min}$
    end while
  end if
  if $f(x, \hat{y}') > F_y$ **then**
    $F_y = f(x, \hat{y}')$, $\hat{y} = \hat{y}'$, and $c = c + 1$
  end if
end while
return $\hat{y}, F_y, \tilde{\omega} = \tilde{\eta}, \tilde{\theta} = (h, \dots)$

5.3.2 WRA using AGA. The second variant, summarized in Algorithm 3, uses AGA as a solver $\mathcal{M}$. The AGA solver uses the numerical gradient $\bar{\nabla}_y$ at the worst-case scenario candidate $\hat{y}$ obtained by `SLSQP` function in the SciPy module in Python. If the search domain for the scenario vector has a box constraint, the projected gradient idea is used to force the worst-case scenario candidate to be feasible. The configuration ω for Algorithm 3 includes the learning rate $\{\eta_k\}_{k=1}^{N_\omega}$, and the other parameters are included in θ.

We use a simple adaptation mechanism for the learning rate $\tilde{\eta}$ in Algorithm 3, similar to the backtracking line search. The learning rate $\tilde{\eta}$ is decreased by $\beta \in (0, 1)$ until the worst-case scenario candidate is improved. If the worst-case scenario candidate is improved for the first trial, the learning rate is increased by $1/\beta$. This is because a significant improvement of the worst-case scenario candidate is expected by a large learning rate in the next iteration.

The termination criteria of Algorithm 3 are described as follows. Algorithm 3 is terminated when the scenario is improved for $c_{\max}$ times. If the infinity norm of the update vector is smaller than $U_{\min}$, i.e., $\|\tilde{\eta}\bar{\nabla}_y f\|_\infty \leqslant U_{\min}$, we consider that a significant increase of the objective function value is not expected and terminate the solver. When Algorithm 3 is terminated by the latter condition, we set $h =$ True and $\mathcal{M}$ is not called with the current configuration in the current WRA call.

The hyperparameters include the initial learning rate $\{\eta_k\}_{k=1}^{N_\omega}$, parameter for updating the learning rate β, termination threshold $U_{\min}$, and maximum number of improvements, $c_{\max}$. They should be set problem-dependently.

5.4 Restart and Local Search Strategy

We implement two devices for practical use to enhance exploration (by restart) and exploitation (by local search).

A restart strategy is implemented to obtain good local optimal solutions when F is multimodal. When a termination condition is satisfied before an f-call budget or a wall clock time budget is exhausted, the λ_x solution candidates and N_ω worst-case scenario candidates at the last iteration are stored in $\mathcal{X}^*$ and $\mathcal{Y}^*$, respectively. We restart the search without inheriting any information from previous restarts. Once the budgets are exhausted, the last solution candidates and worst-case scenario candidates are stored as well. The final output of the algorithm, i.e., the candidate of the global min–max solution, is $\mathrm{argmin}_{x\in\mathcal{X}^*} \max_{y\in\mathcal{Y}^*} f(x,y)$. One can also include randomly sampled scenario vectors to $\mathcal{Y}^*$ when deciding the final output for a good estimate of $F(x)$. The resulting algorithms using CMA-ES and AGA with this restart strategy are denoted as `WRA-CMA` and `WRA-AGA`, respectively.

We implement an optional local search strategy using `ADV-CMA-ES`. If the problem is locally smooth and strongly convex–concave, `ADV-CMA-ES` exhibits significantly faster convergence than WRA. Therefore, by stopping each run of WRA early and performing `ADV-CMA-ES`, we expect that the solution candidate obtained by WRA will be more locally improved by `ADV-CMA-ES` than by spending the same f-calls by WRA. This is implemented as follows. When a termination condition is satisfied, let $\mathcal{Y} = \{y_k\}_{k=1}^{N_\omega}$ be the set of N_ω worst-case scenario candidates, $i_{\mathrm{Adv}} = \mathrm{argmin}_{i=1,\ldots,\lambda_x} \max_{y\in\mathcal{Y}} f(x_i,y)$ be the best-case solution candidate, and $k_{\mathrm{Adv}} = \mathrm{argmax}_{k=1,\ldots,N_\omega} f(x_{i_{\mathrm{Adv}}}, y_k)$ be the corresponding worst-case scenario index obtained at the last iteration. Then, `ADV-CMA-ES` is applied to optimize $f_{\mathcal{Y}}(x,y) = \max_{\tilde{y}\in\{y\}\cup\mathcal{Y}} f(x,\tilde{y})$, with distributions initialized around $(x_{i_{\mathrm{Adv}}}, y_{k_{\mathrm{Adv}}})$ to exhibit local search. The search distribution for x in `ADV-CMA-ES` is initialized by the distribution for x in WRA at the last iteration. The distribution parameters for search in y is initialized by those of $\omega_{k_{\mathrm{Adv}}}$ if Algorithm 2 is used, and the mean is initialized by $y_{k_{\mathrm{Adv}}}$ obtained in Algorithm 3, and a relatively small initial covariance matrix is set if Algorithm 3 is used. The other parameters of `ADV-CMA-ES` are set to the default values proposed in [Akimoto et al. 2022]. Once `ADV-CMA-ES` is terminated, we perform a restart as in `WRA-CMA` and `WRA-AGA`. The approaches using Algorithm 2 and Algorithm 3 with the local search and restart strategies are denoted as `WRA-CMA+ADV` and `WRA-AGA+ADV`, respectively.

6 NUMERICAL EXPERIMENTS ON TEST PROBLEMS

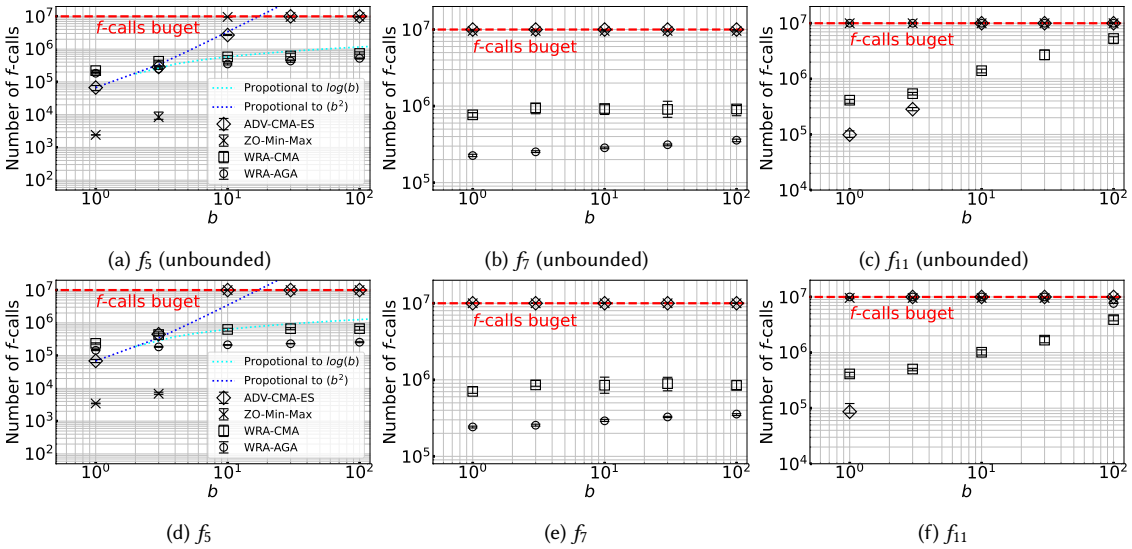
We performed numerical experiments to confirm that existing approaches `ADV-CMA-ES`[3] and `ZO-Min-Max`[4] face Difficulties (I) and (II), whereas the proposed approach can cope with them.

6.1 Common settings

We used test problems listed in Table 1. Unless otherwise specified, the dimensions were $m = n = 20$, and the search domains were $\mathbb{X} = [-3,3]^m$ and $\mathbb{Y} = [-3,3]^n$. The coefficient matrix B was set to $B = \mathrm{diag}(b,\ldots,b)$.

The proposed and existing approaches were configured as follows. The initial mean vector (`WRA-CMA` and `WRA-AGA`) and initial solution candidate (`ADV-CMA-ES` and `ZO-Min-Max`) for outer minimization were drawn from $\mathcal{U}(\mathbb{X})$. The initial covariance matrices for the outer minimization were set to $\left(\frac{u_x-\ell_x}{4}\right)^2 I_m$ in `WRA-CMA`, `WRA-AGA`, and `ADV-CMA-ES`. The initial mean vectors (`WRA-CMA`) and initial worst-case scenario candidates (`WRA-AGA`, `ADV-CMA-ES`, and `ZO-Min-Max`)

[3]The code for `ADV-CMA-ES` is downloaded from https://gist.github.com/youheiakimoto/ab51e88c73baf68effd95b750100aad0
[4]The code for `ZO-Min-Max` is downloaded from https://github.com/KaidiXu/ZO-minmax



(a) f_5 (unbounded) (b) f_7 (unbounded) (c) f_{11} (unbounded)

(d) f_5 (e) f_7 (f) f_{11}

Fig. 2. Average and standard deviation of the number of f-calls spent by WRA-CMA, WRA-AGA, ZO-Min-Max, and ADV-CMA-ES over successful trials on f_5, f_7, and f_{11} with $b \in \{1, 3, 10, 30, 100\}$. Left: unbounded search domains ($\mathbb{X} = R^m$ and $\mathbb{Y} = R^n$). Right: bounded search domains.

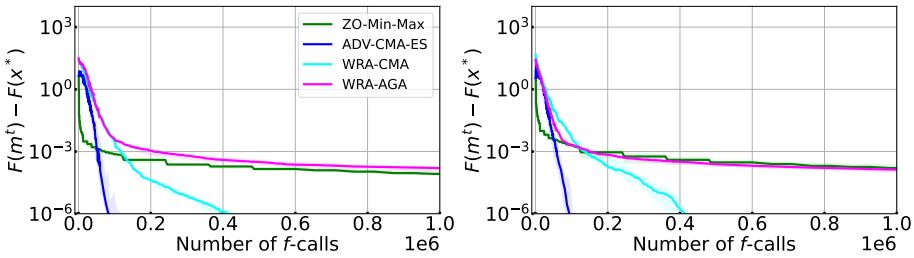
were drawn independently from $\mathcal{U}(\mathbb{Y})$. The initial covariance matrices for the inner maximization were set to $\left(\frac{b_y}{2}\right)^2 I_n$ in WRA-CMA and ADV-CMA-ES. For WRA-CMA and WRA-AGA, we set $V^x_{\min} = 10^{-12}$ (no restart occurred with this setting), $c_{\max} = 1$, and $N_\omega = 36 (= 3 \times \lambda_x)$. For WRA-CMA, we set $V^y_{\min} = 10^{-4}$ and $T_{\min} = 10$. For WRA-AGA, we set $U_{\min} = 10^{-5}$, $\beta = 0.5$, and the initial learning rate $\{\eta_k\}_{k=1}^{N_\omega} = 1$. For ZO-Min-Max, referencing [Liu et al. 2020], we set the learning rates as $\eta_x = 0.02$ and $\eta_y = 0.05$, the number of random direction vectors as $q = 5$, and the smoothing parameter for gradient estimation as $\mu = 10^{-3}$. For ADV-CMA-ES, referencing [Akimoto et al. 2022], we set the threshold parameter for restart as $G_{\text{tol}} = 10^{-6}$, the minimal learning rate as $\eta_{\min} = 10^{-4}$, and the minimal standard deviation as $\bar{\sigma}_{\min} = 10^{-8}$. For simplicity of the analysis, the restart strategy of WRA-CMA and WRA-AGA was not used in these experiments. ADV-CMA-ES performed restart because it is implemented by default. We also turned off the diagonal acceleration mechanism both in CMA-ES for the outer minimization and inner maximization in WRA-CMA.

The performance of each algorithm is evaluated by 20 independent trials. We regarded a trial as successful if $|F(z) - F(x^*)| \leqslant 10^{-6}$ was satisfied for $z = m^t_x$ in case of ADV-CMA-ES, WRA-CMA, and WRA-AGA and for $z = x^t$ in case of ZO-Min-Maxbefore 10^7 f-calls were spent. If the maximum f-calls were spent or some internal termination conditions were satisfied, we regard the trial as failed.

6.2 Experiment 1

To confirm that the proposed approach overcomes Difficulty (I), four approaches were applied to smooth convex–concave problems f_5, f_7 and f_{11} for varying b with and without bounds for the search domains.

6.2.1 Results. Figure 2 shows that WRA-CMA and WRA-AGA could successfully optimize f_5 and f_7 with all $b \in \{1, 3, 10, 30, 100\}$ in all trials, whereas ADV-CMA-ES and ZO-Min-Max failed to optimize them except for f_5 with $b \leqslant 3$. WRA-CMA was the only approach that successfully optimized f_{11} with all b values, whereas ADV-CMA-ES could optimize



(a) f_{11} (b) f_{11} (unbounded)

Fig. 3. Gap $|F(m^t) - F(x^*)|$ with the number of f-calls at $b = 1$ on f_{11}. Solid line: median (50 percentile) over 20 runs. Shaded area: interquartile range (25–75 percentile) over 20 runs.

f_{11} with $b = 1$ with and without boundary and $b = 3$ without boundary. From these results, we confirm that both ZO-Min-Max and ADV-CMA-ES fail in problems where the min–max solution is a global min–max saddle point but is not locally smooth and strongly convex–concave, and our approaches can solve such problems.

When the search domain is unbounded on f_5, both ZO-Min-Max and ADV-CMA-ES successfully locate near-optimal solutions for $b \leqslant 3$ with smaller f-calls than our approaches. However, for $b \geqslant 30$, they failed to converge, although f_5 is smooth and strongly convex–concave. For f_{11} with the unbounded search domain, ZO-Min-Max failed to converge at every trials, and ADV-CMA-ES could not obtain successful convergence for $b \geqslant 10$, although f_{11} is also smooth and strongly convex–concave. For ZO-Min-Max, an inadequate learning rate may be a possible reason. For convergence, it must be tuned problem-dependently. However, even if an appropriate value is set, the slow convergence issue discussed in Section 4 occurs. For ADV-CMA-ES, when $b \geqslant 30$ in f_5 and f_{11} with the unbounded search domain, slow convergence issue is the main reason, as the expected f-calls (blue dash line in Figure 2) in f_5 exceeded f-call budget. When the search domain is unbounded, we expect ADV-CMA-ES to obtain the successful convergence for f_{11} until $b = 10$ similarly as f_5. However, ADV-CMA-ES failed to converge in f_{11} with $b = 10$. We observed that ADV-CMA-ES suffered to approach x^* because the learning rate reached to lower bound $\eta_{\min}$. Therefore, to obtain successful convergence for f_{11} with $b = 10$, lower bound $\eta_{\min}$ for ADV-CMA-ES should be properly set. When there was a bound for search domain, ADV-CMA-ES failed to converge with $b = 10$ for f_5 and $b = 3$ for f_{11}.

The difference between WRA-CMA and WRA-AGA is in the speed of convergence for f_5 and f_7 as well as the performance for f_{11}. For f_5 and f_7, WRA-AGA converged faster than WRA-CMA. Meanwhile, WRA-AGA failed to optimize f_{11} within a given f-call budget. Figure 3 shows gap $F(m^t) - F(x^*)$ on f_{11} with $b = 1$. From Figure 3, we expect that WRA-AGA eventually converges, but the convergence speed is very slow. Preliminary, we confirmed that Algorithm 3 converges slowly on ill-conditioned function. Therefore, on f_{11} which is ill-conditioned in y, Algorithm 3 with small $c_{\max}$ cannot significantly improve the worst-case scenario candidate $\hat{y}$ and the early stopping strategy may terminate the inner maximization process before approximating the worst-case scenario in adequate accuracy. Because of the underestimation of the rankings on the worst-case objective functionRank$_F\{(x_i)_{i=1}^{\lambda_x})\}$ by WRA, outer minimization failed to converge at the global min–max solution, indicating the relevance of the choice of the inner solver for the WRA mechanism. Because CMA-ES is a variable metric approach and the covariance matrices are inherited over WRA calls, WRA-CMA could optimize f_{11} efficiently.

6.2.2 Discussion on the effect of the interaction term. We discuss the effect of the worst-case scenario sensitivity (coefficient matrix B of the interaction term $x^T By$) on f-calls spent by our approaches when the objective function is convex–concave. Figure 2 shows that the numbers of f-calls were in $O(\log(b))$ or even in $O(1)$ in terms of the coefficient of the interaction term, b. We provide a brief but not rigorous explanation of these results.

For simplicity, we focus on f_5 with $m = n$ and $B = \mathrm{diag}(b, \ldots, b)$. The worst-case objective function is $F(x) = \frac{(1+b^2)}{2}\|x\|_2^2$ and the worst-case scenario is $\hat{y}(x) = bx$ in this case. Moreover, we focus on WRA-CMA.

To proceed, we assume that CMA-ES converges linearly for such spherical functions. That is, a point in $\{x : \|x - x^*\| \leqslant \epsilon \cdot \|m_x^0 - x^*\|\}$ around the optimal solution x^* can be found in $O(\log(\|m_x^0 - x^*\|/\epsilon))$ f-calls. Although no rigorous runtime analysis has been performed for CMA-ES, we have ample empirical evidence. Moreover, (1+1)-ES, which is a simplified version of CMA-ES, converges linearly on Lipschitz smooth and strongly convex objective functions [Morinaga et al. 2021].

First, we consider how many iterations CMA-ES for the outer minimization spends to reach a point $m_x^{T_\epsilon}$ such that $\|m_x^{T_\epsilon} - x^*\| \leqslant \epsilon \|m_x^0 - x^*\|$. We call T_ϵ the runtime. WRA returns approximate rankings of given solution candidates, and they highly correlate with the true rankings. Then, CMA-ES is expected to behave similarly in these two rankings. Therefore, if CMA-ES converges linearly for F, we expect that CMA-ES converges linearly for the rankings given by WRA as well, which is partly supported by a theoretical investigation [Akimoto 2022]. Because the worst-case objective function $F(x)$ is spherical, CMA-ES is expected to converge linearly, i.e., the runtime T_ϵ is in $O(\log(\|m_x^0 - x^*\|/\epsilon))$.

Next, we consider how many f-calls WRA-CMA spends in each call. Let the current search distribution of CMA-ES for the outer minimization be $\mathcal{N}(m_x^t, \Sigma_x^t)$. Because F is spherical, Σ_x^t is expected to be proportional to the identity matrix I_m. Let us assume that $\Sigma_x^t \approx \sigma_t^2 I_m$. Then, the solution candidates $x_1, \ldots, x_{\lambda_x}$ given to WRA are independently $\mathcal{N}(m_x^t, \sigma_t^2 I_m)$-distributed. For the two solution candidates x_i and x_j, the expected difference in the worst-case objective function values is as follows:

$$\mathbb{E}[(F(x_i) - F(x_j))^2]^{1/2} = (1 + b^2)\,\mathrm{Tr}((\Sigma_x^t)^2)^{1/2} \approx m^{1/2}(1 + b^2)\sigma_t^2. \tag{4}$$

The early stopping strategy is expected to stop the maximization process once the rankings of the given candidate solutions are well-approximated in terms of Kendall's rank correlation. To have a high value of the Kendall's rank correlation, the orders of $F(x_i)$ and $F(x_j)$ and their approximate values, $f(x_i, \tilde{y}_i)$ and $f(x_j, \tilde{y}_j)$ must be concordant with high probability for each pair (x_i, x_j) among λ_x solution candidates $x_1, \ldots, x_{\lambda_x}$, where $\tilde{y}_i$ $(i = 1, \ldots, \lambda_x)$ is the approximate worst-case scenario for x_i obtained in WRA. It suffices to obtain $\tilde{y}_i$ and $\tilde{y}_j$ such that $|F(x_i) - f(x_i, \tilde{y}_i)|$ and $|F(x_j) - f(x_j, \tilde{y}_j)|$ are both less than $|F(x_i) - F(x_j)|$. With a simple derivation, we obtain

$$F(x) - f(x, \tilde{y}) = \frac{1}{2}\|\hat{y}(x) - \tilde{y}\|^2. \tag{5}$$

That is, if $\|\hat{y}(x_i) - \tilde{y}_i\| \leqslant c \cdot (1 + b^2)^{1/2}\sigma_t$ is satisfied for some $c > 0$, the true order of the two points among λ_x solution candidates will be correctly identified with high probability. Because the objective function of the inner maximization problem is spherical in y, to obtain such approximate worst-case scenarios, the required f-calls is $O\left(\log\left(\frac{\|y^{(0)} - \hat{y}(x_i)\|}{c \cdot (1+b^2)^{1/2}\sigma_t}\right)\right)$, where $\tilde{y}^{(0)}$ denotes the initial scenario for x_i.

Assuming that the Gaussian distribution of CMA-ES for the outer loop does not change significantly from the previous iteration, the worst-case scenario for the solution candidate in the current iteration, $\hat{y}(x)$, and that in the previous iteration are expected to follow the distribution $\mathcal{N}(bm_x^t, b^2\Sigma_x^t)$. Because the warm starting strategy selects the worst-case scenario among the set of scenarios including the ones obtained in the previous WRA call, the distance

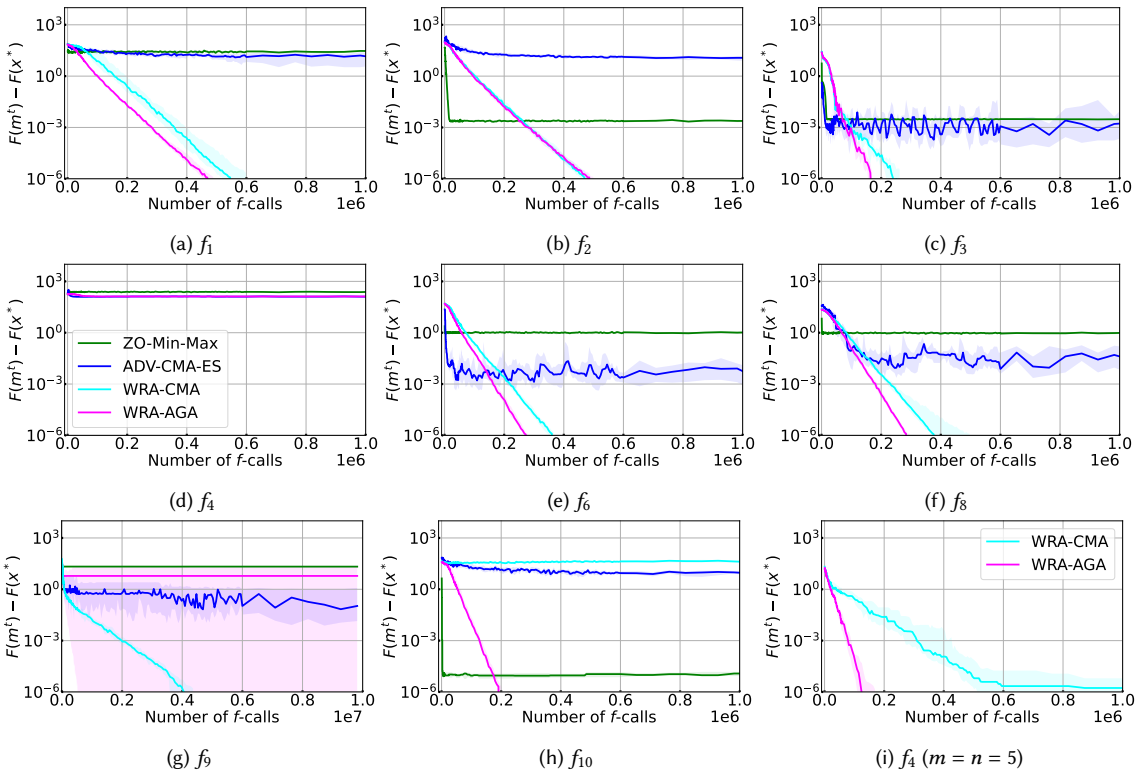


Fig. 4. Gap $|F(m^t) - F(x^*)|$ with the number of f-calls at $b = 1$ on f_1–f_4, f_6, and f_8–f_{10}. Solid line: median (50 percentile) over 20 runs. Shaded area: interquartile range (25–75 percentile) over 20 runs.

between $\hat{y}(x)$ and $\tilde{y}$ is expected to be no greater than $\mathbb{E}[\|\hat{y}(x) - \tilde{y}^{(0)}\|] = b^2 \operatorname{Tr}(\Sigma_x^t) = mb^2\sigma_t^2$. From this, we estimate $\|y^{(0)} - \hat{y}(x)\| \in O(b\sigma_t)$. As a result, f-calls required to approximate the worst-case objective function values for each solution candidate is $O\left(\log\left(\frac{b}{c\cdot(1+b^2)^{1/2}}\right)\right)$.

Altogether, the proposed approach is expected to locate a near-optimal solution with

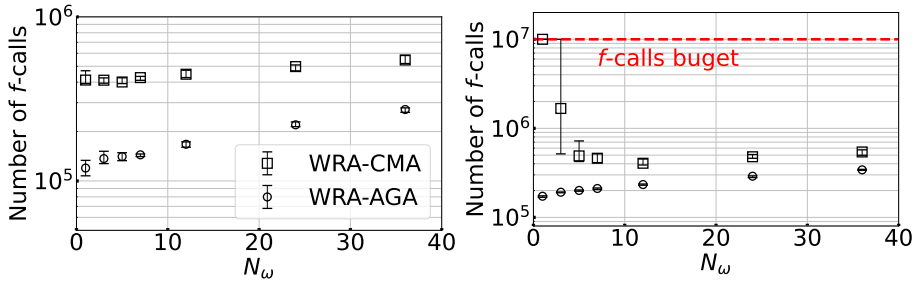
$$O\left(\lambda_y \log\left(\frac{b}{c\cdot(1+b^2)^{1/2}}\right)\cdot\log\left(\frac{\|m_x^{(0)} - x^*\|}{\epsilon}\right)\right) \tag{6}$$

f-calls. It scales as $\log(b)$ for $b \leqslant 1$ and is constant for $b \to \infty$, which well-estimates the behavior observed in Figure 2.

6.3 Experiment 2

We applied four approaches to the problems f_1–f_4, f_6, and f_8–f_{10} to investigate their performance on the functions that are nonsmooth and strongly convex–concave.

The results are shown in Figure 4. We confirm that near-optimal solutions were obtained by `WRA-CMA` for f_1–f_3, f_6, f_8, and f_9 and by `WRA-AGA` for f_1–f_3, f_6, f_8, and f_{10}. Moreover, the existing approaches failed to locate near-optimal solutions in all trials.



(a) f_8 (b) f_1

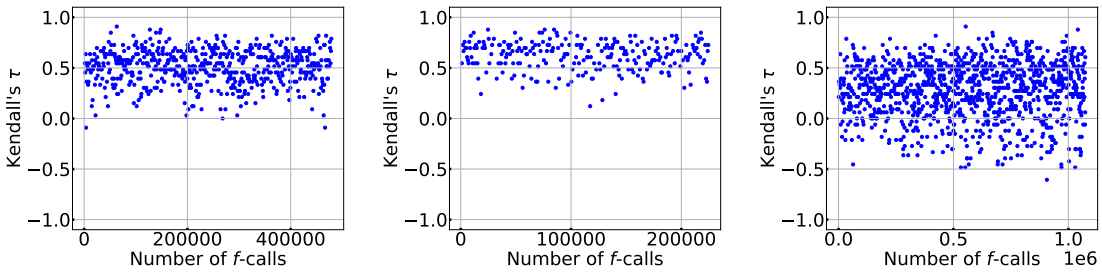
Fig. 5. Average and standard deviation of the number of f-calls spent by WRA-CMA and WRA-AGA with $N_\omega \in \{1, 3, 5, 7, 12, 24, 36\}$. Solid line: median (50 percentile) over 20 runs. Shaded area: interquartile range (25–75 percentile) over 20 runs.

6.3.1 Category (S) (f_3, f_6, and f_8). Our approaches can solve f_3, f_6, and f_8 even with $N_\omega = 1$. Figure 5a demonstrates the results of WRA-CMA and WRA-AGA with $N_\omega = 1$ for f_8. The worst-case scenario for f_3, f_6, and f_8 is a singleton $|\hat{Y}(x^*)| = 1$ and a constant around the global min–max solution x^*. Therefore, maintaining a single configuration ($N_\omega = 1$) was sufficient for the warm starting strategy in WRA to work efficiently on these problems. Figure 5a shows WRA-CMA and WRA-AGA with a smaller N_ω could converge to near global min–max solution with fewer f-calls. This may be because f-calls spent by the warm starting strategy are saved by setting smaller N_ω. The reduction of f-calls by a small N_ω was not significant; therefore, we do not consider N_ω should be daringly small.

6.3.2 Category W (f_1, f_2). Maintaining multiple configurations, i.e., $N_\omega > 1$, is crucial for the proposed approach to successfully converge to the near global min–max solution x^* for functions in Category (W) as we discussed in Section 5.2.1. Figure 5b shows the results of WRA-CMA and WRA-AGA for f_1 with $N_\omega \in \{1, 3, 5, 7, 12, 24, 36\}$. We confirmed that WRA-CMA with a small N_ω failed to converge to x^*. Meanwhile, WRA-AGA could converge to x^* with $N_\omega = 1$. This may be because AGA can rapidly maximize f_1 for y from any starting point in $\mathbb{Y}$ and the warm starting strategy is unnecessary for WRA-AGA in f_1.

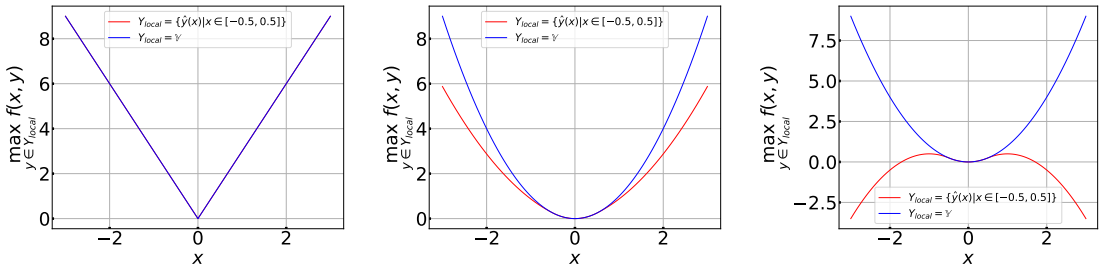
6.3.3 Category (N) (f_4, f_9, and f_{10}). Multimodality in y, particularly with a weak global structure, seems to make it difficult to obtain the global min–max solution. As we see in Figure 4 for f_4, WRA-CMA and WRA-AGA could not successfully converge. The objective function $f(x, \cdot)$ for f_4 has 2^n local solutions and is a multimodal function with a weak structure. Such an objective function is difficult to efficiently optimize with any of the currently proposed algorithms [Hansen 2009]. Therefore, we consider that the proposed approach failed to approximate the worst-case objective function values $\{F(x)\}_{i=1}^{\lambda_x}$ at many iterations; consequently, the outer CMA-ES could not converge to x^*.

Setting N_ω greater than the number of local maxima in $f(x, \cdot)$ is crucial to obtain successful convergence. As we see in Figure 4 for f_9, WRA-CMA could successfully converge. The objective function $f(x, \cdot)$ for f_9 has 8 local maxima. When $\{y_k\}_{k=1}^{N_\omega}$ can include every local solution because of $N_\omega = 36 > 8$, the solver explores the worst-case scenario using a good initial configuration in any case, i.e., the warm starting strategy works effectively. Meanwhile, WRA-AGA could not converge to x^* in most trials. AGA failed to even locally maximize f, possibly due to the ill-condition, more precisely, the Hessian matrix is not necessarily negative definite at some x. As a result of approximating the worst-case rankings $\mathrm{Rank}_F(\{x_i\}_{i=1}^{\lambda_x})$ in several iterations, the outer CMA-ES failed to converge to x^*. Further, for f_4, we confirmed the benefit of setting N_ω greater than the number of local maxima in $f(x, \cdot)$. For $n = 5$, $N_\omega = 36$ is greater than the



(a) f_1 (b) f_5 (c) f_{10}

Fig. 6. Kendall's τ between the rankings of the worst-case objective function values and rankings obtained using the WRA mechanism at each iteration in a typical run of WRA-CMA. WRA-CMA was terminated due to $\mathrm{Cond}(\Sigma_{\boldsymbol{x}}) > \mathrm{Cond}^{\boldsymbol{x}}_{\max}$ before f-calls reached 10^7.



(a) f_1 (b) f_5 (c) f_{10}

Fig. 7. Landscape of $\max_{y \in Y_{local}} f(x, y)$ with $Y_{local} = \mathbb{Y}$ and $Y_{local} = \{\hat{y}(x) | x \in [-0.5, 0.5]\}$ on f_1, f_5, and f_{10} with $m = n = 1$.

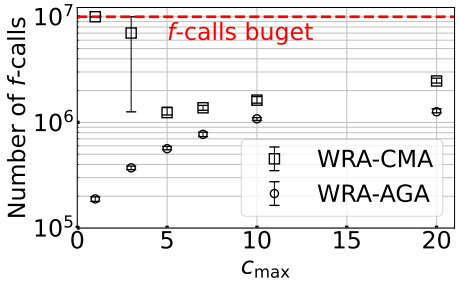


Fig. 8. Average and standard deviation of the number of f-calls spent by WRA-CMA and WRA-AGA for f_{10} with $c_{\max} \in \{1, 3, 5, 7, 10, 20\}$. Solid line: median (50 percentile) over 20 runs. Shaded area: interquartile range (25–75 percentile) over 20 runs.

number of local maxima, which is $2^5 = 32$. Figure 4i shows the experimental result from WRA-CMA and WRA-AGA for f_4 with $m = n = 5$. As shown in Figure 4i, WRA-AGA converged successfully and WRA-CMA converged to a near-optimal solution.

The relevance of $c_{\max}$ is pronounced in the results of WRA-CMA for f_{10}. Figure 4h shows that WRA-CMA with $c_{\max} = 1$ failed to converge for f_{10}. From Figure 6, the Kendall's τ for f_{10} is more frequently negative than those for f_1 and

f_5 in which WRA-CMA could converge successfully. The reason for the low τ value is explained using Figure 7, which visualizes the landscape of an approximated worst-case objective function $\max_{y \in Y_{local}} f(x, y)$ with $Y_{local} = \mathcal{Y}$ (i.e., the ground truth worst-case objective) and $Y_{local} = \{\hat{y}(x) | x \in [-0.5, 0.5]\}$. Figure 7 simulates the situation where the search distribution for x is concentrated around $[-0.5, 0.5]$ and hence N_ω worst-case scenario candidates in WRA are concentrated at the corresponding worst-case scenario region. Differently from f_1 and f_5, the worst-case objective function values of candidate solutions outside $[-0.5, 0.5]$, which are generated by chance, are significantly underestimated for f_{10}. In such a situation, the worst-case scenario search with a small $c_{\max}$ may be insufficient to correctly rank such solutions and they may be regarded as the best solutions. This will prevent convergence to the global min–max solution. From Figure 8, the performance of WRA-CMA is improved by setting a greater $c_{\max}$. However, a too large $c_{\max}$ value requires more f-calls. WRA-AGA could converge successfully for f_{10} even with $c_{\max} = 1$. This could be because the objective function for y in f_{10} was relatively easy for Algorithm 3; therefore, the worst-case scenario could be approximated with high accuracy even for small $c_{\max}$.

7 APPLICATION TO ROBUST BERTHING CONTROL

We confirm the effectiveness of WRA-CMA, WRA-AGA, WRA-CMA+ADV, and WRA-AGA+ADV on a robust berthing control problem presented in [Akimoto et al. 2022].

7.1 Problem description

We exactly follow the problem setup in [Akimoto et al. 2022]. We briefly describe the problem. The objective of this problem is to obtain a controller to control a ship to a target state located near a berth while avoiding collision with the berth. The ship's state is represented by $s \in R^6$, and the control signal is represented by $a \in U \subset R^4$. The state equation is the maneuvering modelling group model used in [Miyauchi et al. 2022]. The feedback controller $u_x : R^6 \to U$ is modeled by a neural network with $m = 99$ parameters. The domain of the network parameters is set to $\mathbb{X} = [-1, 1]^m$.

We consider three cases of uncertainty sets. Case A: The wind condition is the uncertainty vector. The wind condition is parameterized by the wind direction in $[-\pi, \pi]$ [rad] and wind velocity in $[0, 0.5]$ [m/s]. Case B: The coefficient in the state equation regarding the wind force is the uncertain vector, which comprises a 10-dimensional vector. Case C: Both uncertainties in Cases A and B. In all cases, the search domain is scaled to $\mathbb{Y} = [-1, 1]^n$.

The objective function $f(x, y)$ comprises two components. The first component measures the difference between the ship's final state and target state. The second component measures the penalty for a collision with the berth. If a ship collides with the berth during the control period comprising 200 [s], it receives a penalty greater than 10. Our objective is to minimize the worst-case objective function $\max_{y \in \mathbb{Y}} f(x, y)$, where the uncertainty set $\mathbb{Y}$ differs for Cases A, B, and C.

7.2 Experimental settings

The proposed approach and the existing approaches, ZO-Min-Max and ADV-CMA-ES, were applied to the robust berthing control problem. For each problem, we run 20 independent trials with different random seeds. The maximum number of f-calls was set to 2×10^6.

All approaches were configured as in Section 6, where $u_x = 1$, $\ell_x = -1$, and $b_y = 1$ were plugged, except that we turned on the restart strategy of the proposed approach and ADV-CMA-ES and the diagonal acceleration in CMA-ES for the outer minimization and in WRA-CMA and WRA-AGA. We set the number of configurations as $N_\omega = 34 (= 2 \times \lambda_x)$. The termination

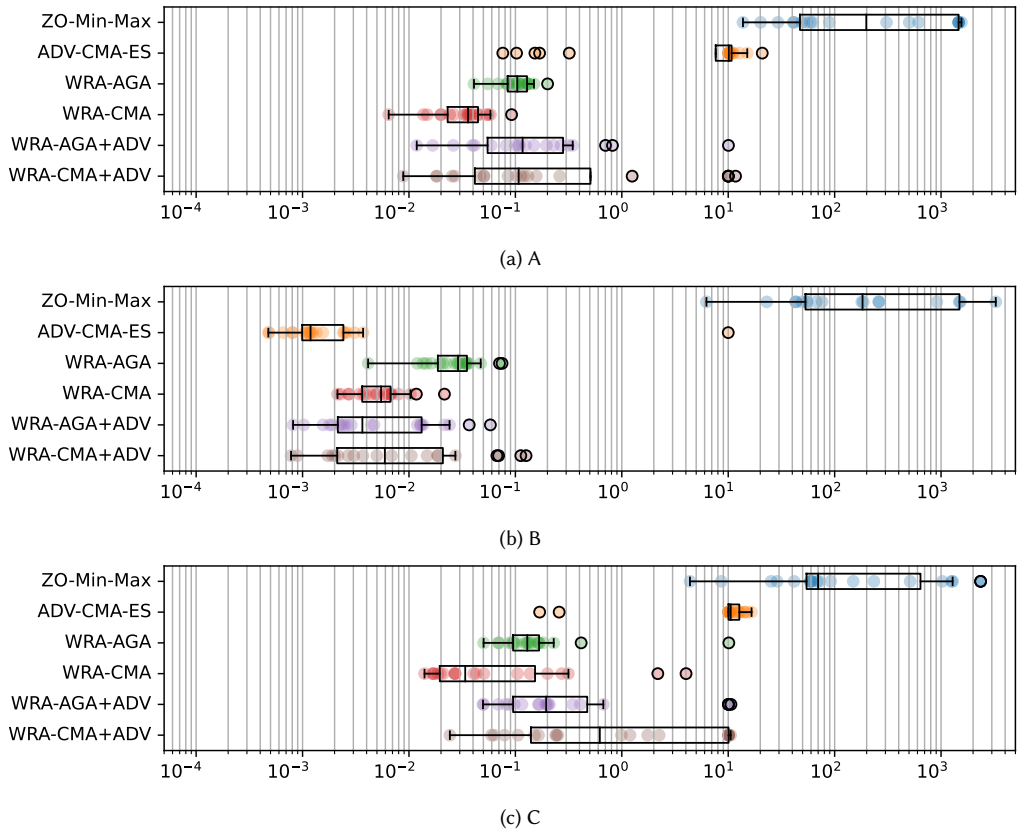
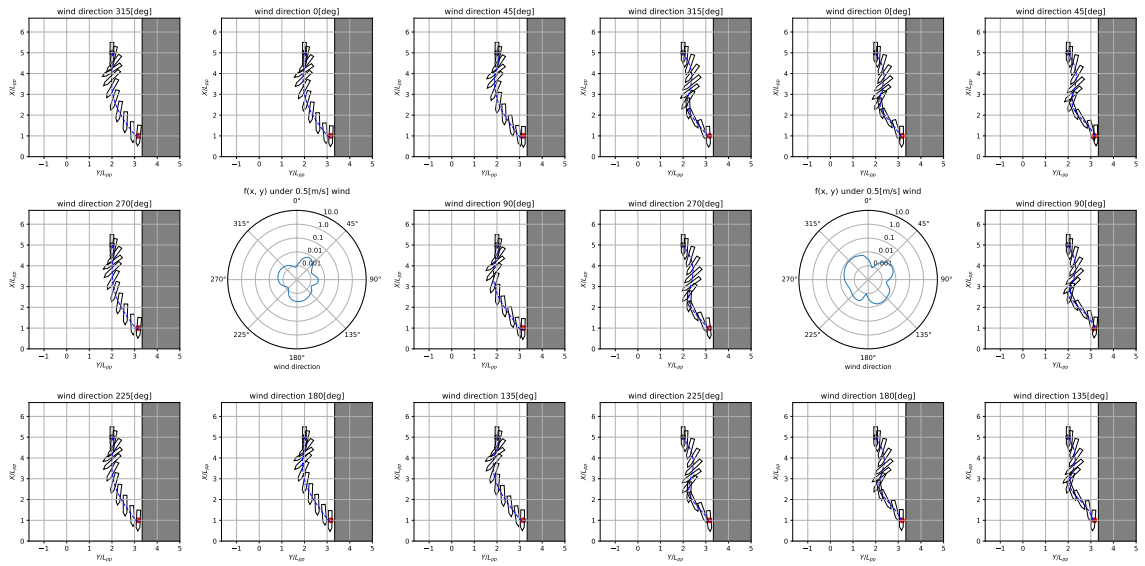


Fig. 9. Performance of the solution obtained in 20 independent trials of ZO-Min-Max, ADV-CMA-ES, WRA-AGA, WRA-CMA+ADV, and WRA-AGA+ADV for Cases A, B, and C. Side edge on each box indicates the lower quartile Q1 and upper quartile Q3, and middle line in each box indicates the median. The lower and upper whiskers are the lowest datum above Q1-1.5(Q3-Q1) and the highest datum below Q3+1.5(Q3+Q1).

thresholds were $V^x_{\min} = 10^{-6}$ and $\mathrm{Cond}^x_{\max} = 10^{14}$. In addition, we terminated the proposed approach if the best worst-case objective function value were not significantly improved. Precisely, we save $\tilde{F}^t_{\min} = \min_{i=1,\ldots,\lambda}(\tilde{F}(x_i))$, where $\tilde{F}(x_i)$ is the approximated worst-case objective function value of x_i computed in WRA, and terminate if $\max_{T=t-10,\ldots,t}(\tilde{F}^t_{\min}) - \min_{T=t-10,\ldots,t}(\tilde{F}^t_{\min}) < 0.1$ is satisfied[5].

For each trial, the obtained solution was evaluated on the worst-case objective function value as in the previous study [Akimoto et al. 2022]. To estimate the worst-case objective function value for each solution, we ran the (1+1)-CMA-ES to approximate $\max_{y\in\mathbb{Y}} f(x, y)$ with 100 different initial points. Then, by taking the maximum of the obtained worst-case scenario candidates, $y_1, \ldots, y_{100}$, the worst-case objective function value is evaluated. For the configuration of the (1+1)-CMA-ES, we exactly followed the previous study [Akimoto et al. 2022].

[5]We often observe that (1+1)-CMA-ES converges significantly faster than the standard CMA-ES (nonelitism CMA-ES) when optimizing a neural network. Probably because of this effect, ADV-CMA-ES could perform several restarts in this problem, whereas the proposed approach could not perform any restart without the termination condition. For the proposed approach to perform multiple restarts, we introduced the termination criterion at the risk of too early termination. As a result, we confirmed that the proposed approach performed the restart about 2, 4, and 2 times for Cases A, B, and C, respectively.



(a) WRA-CMA (b) WRA-CMA+ADV

Fig. 10. Visualization of the trajectories obtained by the controllers for the worst wind condition with the maximum wind velocity of 0.5 [m/s]. The best controller obtained by (a) WRA-CMA and (b) WRA-CMA+ADV are displayed. See the caption of Figure 1 for details of the figures.

7.3 Result and evaluation

The worst-case performances of the obtained solutions are summarized in Figure 9.

Results of ADV-CMA-ES *and* ZO-Min-Max. ADV-CMA-ES could find robust solutions in all but one trial in Case B. Meanwhile, the medians of the worst-case performances in Cases A and C were greater than 10, indicating collision with the berth. These results agree with those of a previous study [Akimoto et al. 2022]. ZO-Min-Max failed to obtain solutions that could avoid collision with the berth in the worst-case scenario in most trials in all cases.

Results of WRA-CMA *and* WRA-AGA. Except for a trial of WRA-AGA in Case C, WRA-CMA and WRA-AGA could find controllers that could avoid collision with the berth in the worst-case scenarios. As discussed in Section 1, we hypothesize that the problems in Cases A and C are such that the worst-case scenario around the optimal controller for F changes discontinuously, and they are difficult for ADV-CMA-ES and ZO-Min-Max. We consider this is one of the reasons for the superior worst-case performances of WRA-CMA and WRA-AGA in Cases A and C. Moreover, the worst-case performances of WRA-CMA and WRA-AGA were significantly worse than that of ADV-CMA-ES in Case B. One reason for this result is the termination criterion introduced in the experiment, which prevents performing an intensive local search. In addition, we confirmed that the worst-case performances of WRA-CMA and WRA-AGA were inferior to that of ADV-CMA-ES, even without this termination condition, attributable to the slower convergence of CMA-ES than (1+1)-CMA-ES for this problem.

Results of WRA-CMA+ADV *and* WRA-AGA+ADV. In Case B, WRA-CMA+ADV and WRA-AGA+ADV could obtain better worst-case performances in several trials. From these results, we confirm that the motivation of running ADV-CMA-ES after WRA-CMA and WRA-AGA, namely, improving the exploitation ability, was realized in Case B. The worst-case performances exhibited

more variance in Case A, and their median was significantly degraded in Case C. The negative effect of running ADV-CMA-ES after WRA-CMA and WRA-AGA may be explained as follows. The set $\bar{Y}$ of worst-case scenario candidates given to ADV-CMA-ES is expected to approximate the worst-case scenario set $\hat{Y}(\bar{x})$ of a given solution candidate $\bar{x}$ as a subset of $\bar{Y}$. During ADV-CMA-ES, $\bar{Y}$ is fixed and the solution candidate x is optimized under $\bar{Y}$ and a newly added scenario candidate y_{adv}. Because $\hat{Y}(x)$ may change with x, $\bar{Y}$ may not approximate $\hat{Y}(x)$ well after ADV-CMA-ES and a single scenario candidate y_{adv} may not be sufficient to recover $\hat{Y}(x)$. Thus, the solution obtained by ADV-CMA-ES may be overfitting to $\bar{Y}$ and there may be scenarios where the performance is worse.

Figure 10 shows the ship trajectory observed under the best controllers obtained by WRA-CMA and WRA-CMA+ADV. Under both controllers, collision is successfully avoided under wind from an arbitrary direction.

8 CONCLUSION

To address the limitation of existing approaches for black-box min–max optimization problems, ZO-Min-Max and ADV-CMA-ES, we propose a novel approach to minimize the worst-case objective function using CMA-ES while approximating the rankings of the worst-case objective function values of the solution candidates using a proposed WRA mechanism. To save f-calls inside the WRA mechanism, we implement a warm starting strategy and an early stopping strategy. We developed WRA-CMA and WRA-AGA by combining the WRA mechanism with CMA-ES and AGA, respectively. A restart strategy and a hybridization of the proposed approach and ADV-CMA-ES are implemented for practical use. The proposed approach was evaluated for 11 test problems and three cases of the robust berthing control problem.

The relevant findings from our numerical experiments are as follows. On smooth strivtly convex–concave problems, where ZO-Min-Max and ADV-CMA-ES have been analyzed for their convergence, the proposed approach exhibited slower convergence than existing approaches when the interaction between x and y is relatively weak. However, the f-calls were not increased significantly for the proposed approach when the interaction was stronger, whereas they were increased significantly for the existing approaches. On nonsmooth strictly convex–concave problems and problems where the global min–max solution is not a strict global min–max saddle point, ZO-Min-Max and ADV-CMA-ES failed to converge, whereas the proposed approach converged. For the former problems, the proposed approach could locate the global min–max solution even with $N_\omega = 1$; a sufficiently large N_ω was a key to the success of the proposed approach. When good initial configurations were not provided for some solution candidates, a greater $c_{\max}$ was helpful.

The effectiveness of WRA-CMA and WRA-AGA were demonstrated in three cases of the robust berthing control problem. For problems where wind direction was included in y, WRA-CMA and WRA-AGA could find controllers that avoid collision with the berth in the worst-case scenario, whereas controllers obtained by the existing approaches, ADV-CMA-ES and ZO-Min-Max, often collided with the berth in the worst-case scenario. For problems where the wind direction is included in y, the worst-case scenario is expected to change discontinuously around the optimal controller for F, and they are difficult for the existing approaches. Moreover, the proposed approach can address such a difficulty. Therefore, we consider that controllers obtained using the proposed approach were superior to those obtained using the existing approaches. For a problem where the existing approaches obtained controllers that avoid a collision, we confirmed that the existing approaches found a better solution than the solutions obtained using the proposed approach. In addition, for such a problem, several trials showed that better controllers were obtained by running ADV-CMA-ES after WRA-CMA and WRA-AGA.

Besides the above advantages of the proposed approach, one practical advantage of the proposed approach over ZO-Min-Max and ADV-CMA-ES is that it is parallel-implementation friendly. In WRA, λ_x solvers $\mathcal{M}(\omega_k)$ $(k = 1, \ldots, \lambda_x)$ can be run in parallel. The $\lambda_x N_\omega$ evaluations of $f(x_i, y_k)$ at the beginning of WRA can be performed in parallel.

Moreover, if WRA-CMA is used, λ_y f-calls at each iteration of $\mathcal{M}(\omega_k)$ can be performed in parallel. In total, roughly $\lambda_x\lambda_y$ times speedup in terms of the wall clock time can be achieved ideally. For example, in Case C of the robust berthing control problem, we have $m = 99$ and $n = 12$; hence, $\lambda_x = 17$ and $\lambda_y = 11$, resulting in a possible speedup of factor 187. Each f evaluation took about 0.1 s on average, amounting to about 2.3 days for each trial. If the ideal speedup is achieved, the wall clock time reduces to about 18 min. This compensates for the disadvantage of the proposed approach over ADV-CMA-ES: slower convergence.

The main limitation of this study is the lack of theoretical guarantees. For the WRA mechanism to work effectively, we assume that $\hat{y}(x)$ is continuous almost everywhere and $\max_{y\in\mathbb{Y}} f(x, y)$ can be solved efficiently for each x. However, questions as to how much $\hat{y}$ can be sensitive or how efficiently the inner solver $\mathcal{M}$ should solve $\max_{y\in\mathbb{Y}} f(x, y)$ are not answered formally in this study. Such a theoretical investigation provides not only a guarantee of the performance of the proposed approach but also a seed to improve it. Therefore, theoretical investigations of the WRA mechanism are important future research directions.

The black-box min–max optimization lacks the de facto standard benchmarking testbed, covering problems with different characteristics. In this study, we design 11 test problems from the perspective of the characteristics of the global min–max solution (whether it is a strict min–max saddle point, a weak min–max saddle point, or not a min–max saddle point), and the perspective of the smoothness and the strong convexity of f. Moreover, we limit our focus on problems where $f(x, y)$ has relatively simple characteristics with respect to x and y and difficulties in black-box optimization, such as ruggedness, non-separability, and ill-conditioning, are yet to be considered. For example, all test problems are convex in x except for f_{10} and the effect of the multimodality in x is not considered. The investigation of the effect of ill-conditioning of f is limited to the comparison between f_5 and f_{11}. Because approaches for black-box min–max optimization are designed and improved based on benchmarking and theoretical analyses on black-box min–max optimization are rather limited, developing a benchmarking test is highly desired. This is also an important future research direction.

REFERENCES

Y. Akimoto. 2022. Monotone Improvement of Information-Geometric Optimization Algorithms with a Surrogate Function. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 1354–1362. https://doi.org/10.1145/3512290.3528690

Y. Akimoto and N. Hansen. 2020. Diagonal Acceleration for Covariance Matrix Adaptation Evolution Strategies. *Evolutionary Computation* 28, 3 (09 2020), 405–435. https://doi.org/10.1162/evco_a_00260

Y. Akimoto, Y. Miyauchi, and A. Maki. 2022. Saddle Point Optimization with Approximate Minimization Oracle and Its Application to Robust Berthing Control. *ACM Trans. Evol. Learn. Optim.* 2, 1 (2022). https://doi.org/10.1145/3510425

Y. Akimoto, N. Sakamoto, and M. Ohtani. 2020. Multi-fidelity Optimization Approach Under Prior and Posterior Constraints and Its Application to Compliance Minimization. In *Parallel Problem Solving from Nature – PPSN XVI*. Springer International Publishing, Cham, 81–94. https://doi.org/10.1007/978-3-030-58112-1_6

Y. Akimoto, T. Shimizu, and T. Yamaguchi. 2019. Adaptive Objective Selection for Multi-Fidelity Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Prague, Czech Republic) *(GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 880–888. https://doi.org/10.1145/3321707.3321709

A. Al-Dujaili, S. Srikant, E. Hemberg, and U. O'Reilly. 2019. On the application of Danskin's theorem to derivative-free minimax problems. *AIP Conference Proceedings* 2070, 1 (2019), 020026. https://doi.org/10.1063/1.5089993

D. V. Arnold and N. Hansen. 2010. Active Covariance Matrix Adaptation for the (1+1)-CMA-ES. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation* (Portland, Oregon, USA) *(GECCO '10)*. Association for Computing Machinery, New York, NY, USA, 385–392. https://doi.org/10.1145/1830483.1830556

H. J. C. Barbosa. 1999. A coevolutionary genetic algorithm for constrained optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 3. 1605–1611 Vol. 3.

D. Bertsimas, O. Nohadani, and K. M. Teo. 2010a. Nonconvex Robust Optimization for Problems with Constraints. *INFORMS Journal on Computing* 22, 1 (2010), 44–58. https://doi.org/10.1287/ijoc.1090.0319 arXiv:https://doi.org/10.1287/ijoc.1090.0319

D. Bertsimas, O. Nohadani, and K. M. Teo. 2010b. Robust Optimization for Unconstrained Simulation-Based Problems. *Operations Research* 58, 1 (2010), 161–178. https://doi.org/10.1287/opre.1090.0715 arXiv:https://doi.org/10.1287/opre.1090.0715

I. Bogunovic, J. Scarlett, S. Jegelka, and V. Cevher. 2018. Adversarially Robust Optimization with Gaussian Processes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 5765–5775.

Z. Bouzarkouna, D. Y. Ding, and A. Auger. 2012. Well placement optimization with the covariance matrix adaptation evolution strategy and meta-models. *Computational Geosciences* 16 (2012), 75–92. https://doi.org/10.1007/s10596-011-9254-2

C. Daskalakis, S. Skoulakis, and M. Zampetakis. 2021. *The Complexity of Constrained Min-Max Optimization.* Association for Computing Machinery, New York, NY, USA, 1466–1478. https://doi.org/10.1145/3406325.3451125

J. Diakonikolas, C. Daskalakis, and M. Jordan. 2021. Efficient Methods for Structured Nonconvex-Nonconcave Min-Max Optimization. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 130)*. PMLR, 2746–2754. https://proceedings.mlr.press/v130/diakonikolas21a.html

G. Fujii, Y. Akimoto, and M. Takahashi. 2018. Exploring optimal topology of thermal cloaks by CMA-ES. *Applied Physics Letters* 112, 6 (2018), 061108. https://doi.org/10.1063/1.5016090

N. Hansen. 2009. Benchmarking a BI-Population CMA-ES on the BBOB-2009 Function Testbed. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers* (Montreal, Québec, Canada) *(GECCO '09)*. Association for Computing Machinery, New York, NY, USA, 2389–2396. https://doi.org/10.1145/1570256.1570333

N. Hansen. 2019. A Global Surrogate Assisted CMA-ES. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Prague, Czech Republic) *(GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 664–672. https://doi.org/10.1145/3321707.3321842

N. Hansen and A. Auger. 2014. *Principled Design of Continuous Stochastic Search: From Theory to Practice.* Springer Berlin Heidelberg, Berlin, Heidelberg, 145–180.

N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. 2010. Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation* (Portland, Oregon, USA) *(GECCO '10)*. Association for Computing Machinery, New York, NY, USA, 1689–1696. https://doi.org/10.1145/1830761.1830790

N. Hansen and A. Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evol. Comput.* 9, 2 (2001), 159–195. https://doi.org/10.1162/106365601750190398

J. W. Herrmann. 1999. A genetic algorithm for minimax optimization problems. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 2. 1099–1103 Vol. 2.

M. Kendall and J. D. Gibbons. 1990. *Rank Correlation Methods* (5th ed.). Oxford University Press.

T. Liang and J. Stokes. 2019. Interaction Matters: A Note on Non-asymptotic Local Convergence of Generative Adversarial Networks. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, 907–915. https://proceedings.mlr.press/v89/liang19b.html

M. Liu, H. Rafique, Q. Lin, and T. Yang. 2021. First-Order Convergence Theory for Weakly-Convex-Weakly-Concave Min-Max Problems. *J. Mach. Learn. Res.* 22, 1, Article 169 (jan 2021), 34 pages.

S. Liu, S. Lu, X. Chen, Y. Feng, K. Xu, A. Al-Dujaili, M. Hong, and U. O'Reilly. 2020. Min-Max Optimization without Gradients: Convergence and Applications to Black-Box Evasion and Poisoning Attacks. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 6282–6293. https://proceedings.mlr.press/v119/liu20j.html

A. Maki, N. Sakamoto, Y. Akimoto, H. Nishikawa, and N. Umeda. 2020. Application of optimal control theory based on the evolution strategy (CMA-ES) to automatic berthing. *Journal of Marine Science and Technology* 25, 1 (2020), 221–233.

A. L. Marsden, M. Wang, J. E. Dennis, and P. Moin. 2004. Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering* 5, 2 (2004), 235–262.

A. Miyagi, K. Fukuchi, J. Sakuma, and Y. Akimoto. 2021. Adaptive Scenario Subset Selection for Min–Max Black-Box Continuous Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '21)*. 697–705. https://doi.org/10.1145/3449639.3459291

A. Miyagi, H. Yamamoto, Y. Akimoto, and Z. Xue. 2018. Parallel Workflow to Optimize Well Placement in Heterogeneous Reservoir using Covariance Matrix Adaptation Evolution Strategy *(GHGT-14)*. 10 pages.

Y. Miyauchi, R. Sawada, Y. Akimoto, N. Umeda, and A. Maki. 2022. Optimization on planning of trajectory and control of autonomous berthing and unberthing for the realistic port geometry. *Ocean Engineering* 245 (2022), 110390. https://doi.org/10.1016/j.oceaneng.2021.110390

D. Morinaga, K. Fukuchi, J. Sakuma, and Y. Akimoto. 2021. Convergence Rate of the (1+1)-Evolution Strategy with Success-Based Step-Size Adaptation on Convex Quadratic Functions. (2021), 1169–1177. https://doi.org/10.1145/3449639.3459289

M. Nouiehed, M. Sanjabi, T. Huang, J. D. Lee, and M. Razaviyayn. 2019. Solving a Class of Non-Convex Min-Max Games Using Iterative First Order Methods. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/25048eb6a33209cb5a815bff0cf6887c-Paper.pdf

J. E. Onwunalu and L. J. Durlofsky. 2010. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences* 14, 1 (2010), 183–198. https://doi.org/10.1007/s10596-009-9142-1

D. M. Ostrovskii, B. Barazandeh, and M. Razaviyayn. 2021. Nonconvex-Nonconcave Min-Max Optimization with a Small Maximization Domain. https://doi.org/10.48550/ARXIV.2110.03950

Z. Pitra, M. Hanuš, J. Koza, J. Tumpach, and M. Holeňa. 2021. Interaction between Model and Its Evolution Control in Surrogate-Assisted CMA Evolution Strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 528–536. https://doi.org/10.1145/3449639.3459358

X. Qiu, J. Xu, Y. Xu, and K. C. Tan. 2018. A New Differential Evolution Algorithm for Minimax Optimization in Robust Design. *IEEE Transactions on Cybernetics* 48, 5 (2018), 1355–1368.

M. Razaviyayn, T. Huang, S. Lu, M. Nouiehed, M. Sanjabi, and M. Hong. 2020. Nonconvex Min-Max Optimization: Applications, Challenges, and Recent Theoretical Advances. *IEEE Signal Processing Magazine* 37, 5 (2020), 55–66. https://doi.org/10.1109/MSP.2020.3003851

L. M. Rios and N. V Sahinidis. 2013. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization* 56, 3 (2013), 1247–1293.

T. Tanabe, K. Fukuchi, J. Sakuma, and Y. Akimoto. 2021. Level Generation for Angry Birds with Sequential VAE and Latent Variable Evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 1052–1060. https://doi.org/10.1145/3449639.3459290

D. Urieli, P. MacAlpine, S. Kalyanakrishnan, Y. Bentor, and P. Stone. 2011. On Optimizing Interdependent Skills: A Case Study in Simulated 3D Humanoid Robot Soccer. In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'11)*. http://www.cs.utexas.edu/users/ai-lab?AAMAS11-urieli

E. V. Vlatakis-Gkaragkounis, L. Flokas, and G. Piliouras. 2021. Solving Min-Max Optimization with Hidden Structure via Gradient Descent Ascent. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 2373–2386. https://proceedings.neurips.cc/paper/2021/file/13bf4a96378f3854bcd9792d132eff9f-Paper.pdf

T. Yamaguchi and Y. Akimoto. 2018. A Note on the CMA-ES for Functions with Periodic Variables. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Kyoto, Japan) *(GECCO '18)*. Association for Computing Machinery, New York, NY, USA, 227–228. https://doi.org/10.1145/3205651.3205669