# Algorithm design for mixed-integer black-box optimization problems with uncertainty

Thomaser, A.M.

# Chapter 5

# Handling Discretization

Optimization problems can be characterized by whether the input variables are continuous or discrete (Section 2.1). However, in practice, the distinction between continuous and discrete variables is not as binary as it may first appear. For instance, the optimization of industrial designs must contend with the physical limits of precision, effectively introducing a level of discretization to variables that are theoretically continuous. Additionally, the computational representation of continuous variables in the form of floating-point numbers possesses finite precision.

This chapter examines the impact of variable discretization on the performance of CMA-ES. The presented results build upon [124]. A method for discretizing continuous functions is introduced (Section 5.1) and a comprehensive study is conducted to assess the effects (Section 5.3). The performance of various CMA-ES variants are compared alongside with an EA (Section 5.2) designed for discrete optimization problems. The results are presented in Section 5.4 followed by a short conclusion (Section 5.5).

## 5.1 Function Discretization

To investigate the impact of discretization on optimization algorithm performance a test function is required. Any continuous optimization problem can be transformed into a discrete optimization problem by limiting the set of feasible values and rounding continuous values to this finite set. However, such a transformation can significantly change the search landscape. The location of the global optimum does not remain constant in the general case.

The proposed method only considers problems for which the location of the global optimum $\mathbf{x}^*$ is known. This ensures that the location of the optimum can be adjusted after the discretization if necessary. The discretization process begins by creating a grid with a specified number of levels $n_{\text{levels}}$ between a lower bound $l$ and an upper bound $u$ for each dimension $d_i$, resulting in evenly spaced numbers over the interval, with a distance $\Delta$ between two discrete values:

$$G_{d_i} = l + j \cdot \underbrace{\frac{u-l}{n_{\text{levels}}-1}}_{\Delta}, \quad \text{for } j = 0, 1, \ldots, n_{\text{levels}} - 1. \tag{5.1}$$

Continuous values are then rounded to the nearest feasible value on the grid. This procedure creates a modified landscape in which each discrete value lies in the center of a plateau of identical fitness. To ensure that the global optimum $\mathbf{x}^*$ is included in the discretized space, a subsequent translation is applied. Since the bounds remain fixed, this translation can cause values adjacent to the bounds to shift outside the feasible domain. To address this, all out-of-bounds values are clipped to the nearest bound. Additionally, the plateaus near the bounds may be smaller because values on one side are rounded to the nearest grid point within the bounds, while values on the other side fall outside the bounds. Figure 5.1 illustrates the discretization of an ellipsoid function in one dimension for different numbers of levels $n_{\text{levels}}$.

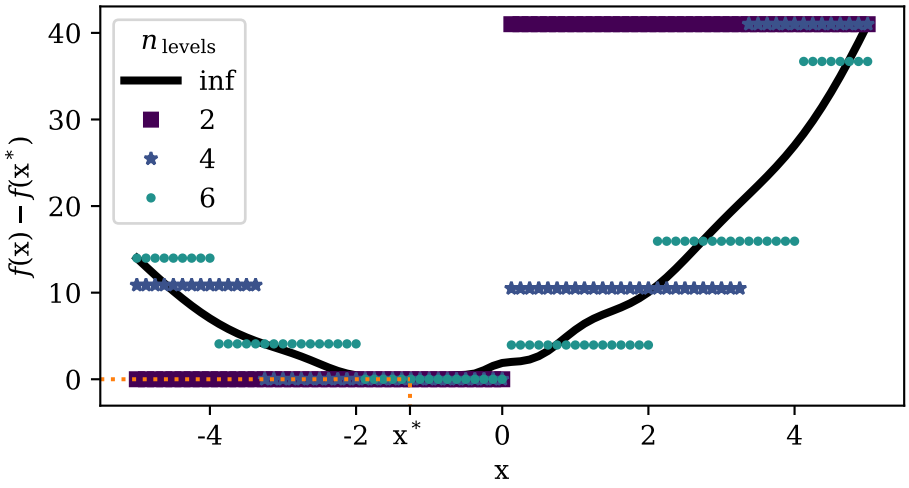


**Figure 5.1:** Discretized one-dimensional ellipsoid function $f(\mathbf{x})$, with $\mathbf{x} \in [-5, 5]$ for the different numbers of levels $n_{\text{levels}} \in \{\text{inf}, 2, 4, 6\}$ with the global optimum at $\mathbf{x}^*$.

Discretization alters the original landscape of an optimization problem and can thereby change its key characteristics. Figure 5.2 illustrates the effect of discretizing a two-dimensional Rosenbrock function (BBOB function F8 (Table 2.1)) for different numbers of levels. The primary challenge for optimization algorithms operating on the continuous Rosenbrock function is the need to constantly adjust search directions to navigate along a narrow, curved ridge toward the optimum.

The application of the discretization method introduces additional complexity by creating multiple local optima. The original unimodal problem is transformed into a multimodal one. These artificially induced local optima can mislead an optimization algorithm and complicate the search process. The emergence of multimodality in an originally unimodal problem due to discretization has also been reported by Tušar et al. [130]. However, this effect does not occur if the original problem is axis-parallel (such as the BBOB sphere function $f_1$ or ellipsoid function $f_2$), where one of the neighboring discrete values always improves the objective function.

Reducing the number of levels $n_{\text{levels}}$ to a small number, such as five levels, results again in a unimodal landscape. However, this simplification comes at the cost of losing the characteristic ridge feature of the Rosenbrock function. Thus, maintaining the intricate features of a complex landscape without introducing multimodality presents a significant challenge. Conversely, if there are too few levels, these defining features are inevitably lost.

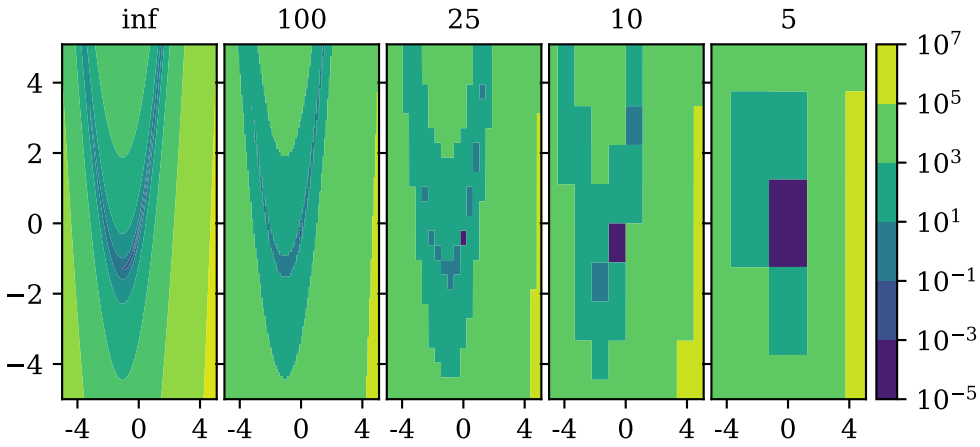


**Figure 5.2:** Discretized two-dimensional Rosenbrock function $f(\mathbf{x})$, with $\mathbf{x} \in [-5, 5]^2$ for the different numbers of levels $n_{\text{levels}} \in \{\text{inf}, 100, 25, 10, 5\}$. Several local optima can be created due to the discretization, clearly observable for $n_{\text{levels}} = 25$.

## 5.2 EA for Integer Programming

A benchmark algorithm for solving discrete optimization problems is needed to compare and assess the effects and the extent of the impact of variable discretization on the performance of CMA-ES. The Evolutionary Algorithm for Integer Programming (int-EA), devised by Rudolph in 1994 [110], represents such a specialized approach for solving optimization problems within integer domains and is described in the following.

The int-EA is characterized by employing a mutation distribution that is designed for integer search spaces, deviating from traditional continuous space evolutionary strategies. Theoretical analysis has revealed the maximum entropy mutation distribution for unbounded integer domains to be [110]:

$$p_k = \frac{p}{2-p}(1-p)^{|k|}, \tag{5.2}$$

where $p_k$ denotes the probability of the integer $k$ being selected. This distribution is symmetric around zero. Therefore, smaller mutations are favored, while larger variations are still possible. To generate random numbers that conform to this specialized distribution, a technique can be employed that involves the difference of two independent random variables. These variables, $G_1$ and $G_2$, are parameterized by:

$$p = 1 - m/((1+m)^{1/2} + 1), \tag{5.3}$$

where $m$ is the deviation parameter that directly controls the mutation variance of the int-EA [110]. The deviation parameter $m$ within int-EA is analogous to the mutation strength $\sigma$ in traditional ES algorithms and is encoded as a strategy parameter, similar to $\sigma$, allowing the int-EA to self-adapt its mutation rate over time. Structurally, the int-EA closely follows the general framework of the $(\mu, \lambda)$-ES, with nearly identical core mechanics.

To enable the int-EA to operate on the same discretized function, the discrete set of values is encoded as an integer space $\mathbf{z} \in [0, n_{\text{levels}}]^d \subseteq \mathbb{Z}^d$. These values are then transformed back to the original space using the formula $\mathbf{x} = l + \mathbf{z} \cdot \frac{u-l}{n_{\text{levels}}}$, where $l$ and $u$ denote the lower and upper bounds of the original search space, respectively. This encoding and transformation process allows both integer-based and continuous solvers to be assessed on the same discretized version of the original objective function $f(\mathbf{x})$. This facilitates a direct comparison between, e.g., the int-EA and CMA-ES.

## 5.3 Experimental Setup

The experimental setup is designed to study the effects of discretization on the performance of different CMA-ES variants and to compare them with the int-EA using the proposed discretization approach (Section 5.1). The BBOB sphere function $f_1$ and the ellipsoid function $f_2$ serve as the original continuous optimization problems. These problems are chosen because they are unimodal and also separable, which ensures that the discretization does not introduce multimodality. Thus, there are no local optima where the optimization algorithm can get stuck in both continuous and discretized domains.

The performance metric chosen for this study is the success rate, which measures the proportion of runs that solve the problem within a given budget. The BBOB framework defines a problem as "solved" if the distance to the optimum in the objective space, defined as $\delta_{f^*} = f(\mathbf{x}) - f(\mathbf{x}^*)$, is less than $10^{-8}$.

For the experimental analysis, the first ten instances of the BBOB problems $f_1$ and $f_2$ are utilized, with 20 independent optimization runs conducted for each instance, resulting in a total of 200 runs per setting. The experiments cover a range of different numbers of levels, with $n_{\text{levels}} \in \{2, 3, 5, 10, 10^2, 10^3, 10^4, \text{inf}\}$, across dimensions $d \in \{2, 5, 10, 20, 40\}$. Each run is allocated a budget of 2 000 function evaluations per dimension, amounting to 40 settings per BBOB problem and algorithm.

The used configuration of the int-EA is for the population parameters $\mu = 30$ and $\lambda = 100$, which showed good performance. The initial values for the deviation parameter $m_i$ mirror the initialization of $\sigma_i$ in traditional ES, providing a comparable baseline for the evolutionary process. This configuration ensures that the int-EA is well-equipped to navigate the integer search space.

## 5.4 Results

To investigate the impact of discretization on the performance of the standard $(\mu, \lambda)$-CMA-ES without integer handling, the algorithm is applied to a discretized five-dimensional sphere function with a spectrum of different numbers of levels $n_{\text{levels}}$. Figure 5.3 shows the evolution of the distance to the optimal function value $\delta_{f^*}$ across successive generations for 100 independent runs for each of the five considered number of levels $n_{\text{levels}} \in \{2, 10, 10^3, \text{inf}\}$.

In the absence of discretization ($n_{\text{levels}} = \text{inf}$), CMA-ES consistently solved the sphere function across all 100 runs. The distance to the optimal function value $\delta_{f^*}$ consistently declines as the algorithm progresses through generations. Upon introducing discretization (Equation 5.1), two notable patterns emerged (Figure 5.3).

First, with a finite number of levels, the runs diverged. A fraction of the runs rapidly achieves a $\delta_{f^*}$ value below $10^{-8}$, indicating successful optimization, while others experienced stagnation. This divergence is attributed to the size of the region surrounding the optimum, which is of the size $\Delta^d$. When the algorithm reaches this optimal region, the distance to the optimal function value $\delta_{f^*}$ directly falls to zero.

Second, if the optimal plateau is not reached quickly, the CMA-ES can become trapped on a suboptimal plateau, progressively reducing the step size. This lowers the probability of reaching another, potentially superior plateau. The likelihood of convergence to the optimum decreases. As the number of levels increases, these effects diminish. Nevertheless, for 1 000 levels some runs of CMA-ES still exhibits stagnation on the discretized five-dimensional sphere function.

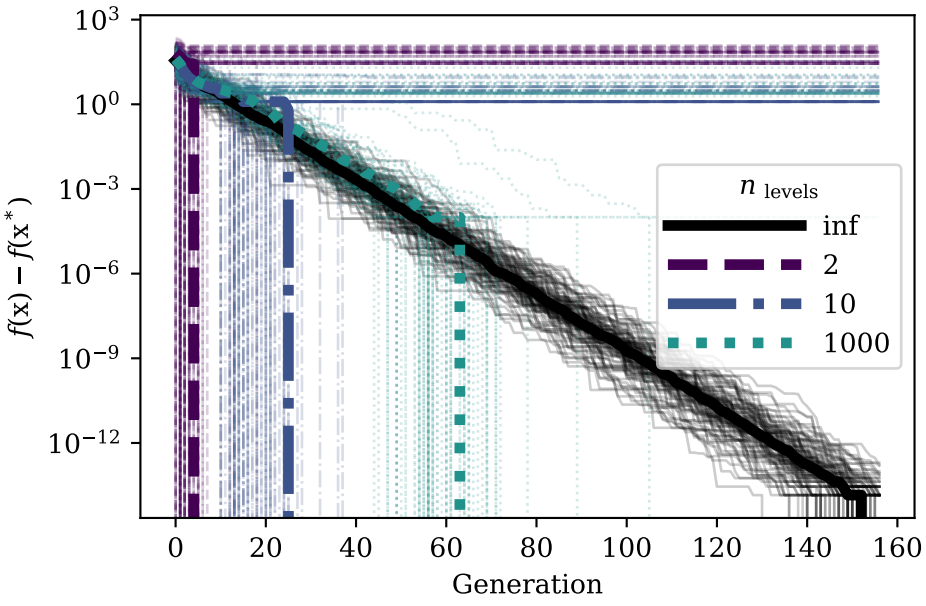


**Figure 5.3:** Distance to the optimal function value $\delta_{f^*}$ over generations of single CMA-ES runs on a discretized five-dimensional sphere function with a spectrum of different numbers of levels $n_{\text{levels}} \in \{2, 10, 10^3, \text{inf}\}$ for a total of 100 runs and the median (bold line) for each number of levels.

### 5.4.1 Success Rates

The discretization not only slows convergence but causes stagnation, stopping the convergence progress of the standard $(\mu, \lambda)$-CMA-ES entirely. Therefore, the rate of successful runs emerges as a good metric for evaluating the impact of discretization on the performance of CMA-ES. The analysis commences with a comparison between the standard CMA-ES, the CMA-ESwM and the int-EA. Figure 5.4 shows the results for different numbers of levels and dimensions (Section 5.3). For comparison, in the continuous case, where $n_{\text{levels}} = \text{inf}$, all runs of CMA-ES on the two considered BBOB problems $f_1$ and $f_2$ successfully converge within the prescribed budget across all dimensions, achieving a success rate of 1.0.

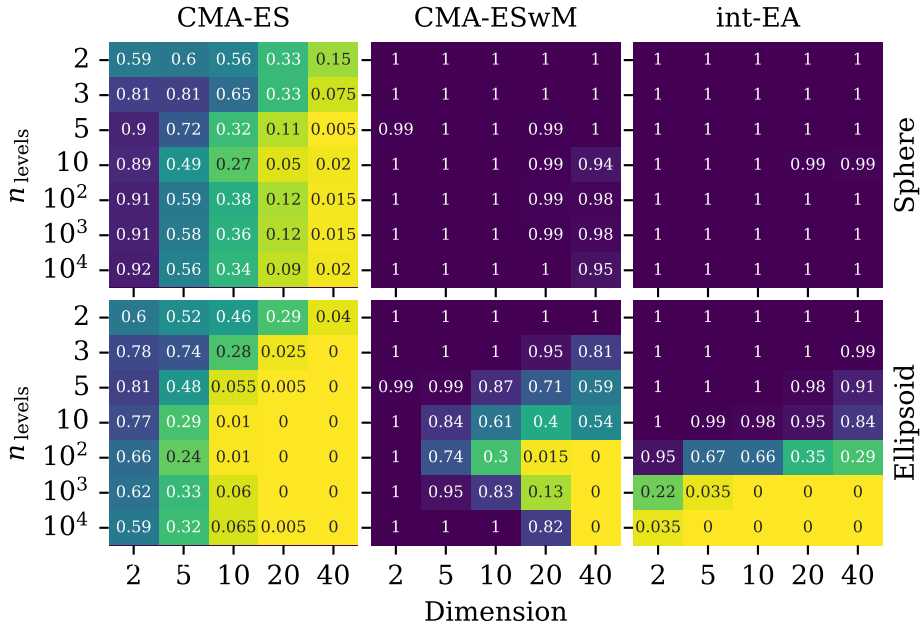


**Figure 5.4:** Success rates for CMA-ES, CMA-ESwM and int-EA on the sphere function (top row) and ellipsoid function (bottom row), across various dimensions (x-axis) and discretization levels $n_{\text{levels}}$ (y-axis). Each algorithm is given a budget of 2 000 evaluations per dimension, with 200 runs conducted for each setting. The target function value is set to $10^{-8}$.

For all the considered algorithms, the discretized ellipsoid function is more difficult to solve than the discretized sphere function. The sphere function is isotropic, which facilitates a more straightforward optimization process even after discretization. In contrast, the ellipsoid function is anisotropic. The variable scale varies along different axes.

The step size adaptation mechanism of CMA-ES is sensitive to the scale of the search space. The various scales along different dimensions of the ellipsoid function mean that the step size needs to be carefully balanced in order to progress. This balancing becomes increasingly complex within a discretized search landscape. Here, the step size needs to be sufficiently large to escape the plateaus in some dimensions while simultaneously being adequately restrained to permit precise, incremental advancements in others. The interplay between these requirements can significantly hinder the optimization process.

Adding a certain amount of discretization considerably increases the difficulties faced by the standard $(\mu, \lambda)$-CMA-ES. The higher the dimension, the more the discretization has a negative impact on the performance of CMA-ES (Figure 5.4 left column). However, already in the two-dimensional setting, the discretization leads to a failure rate of at least around 10% for the sphere function and around at least 20% for the ellipsoid function, indicating that CMA-ES stagnates. In higher dimensions, such as 20 and particularly 40, the success rate decreases further, with almost no runs finding the solution. The extension with the margin improves the performance significantly and outperforms the standard CMA-ES on all problem settings (Figure 5.4 center column). For the sphere function, the CMA-ESwM nearly always solves the problem, regardless of the setting. However, for the ellipsoid function, CMA-ESwM struggles particularly with a moderate number of levels. The complexity of the problem amplifies with increasing dimensionality. A small number of levels simplifies the optimization task due to the enlarged region containing the optimal solution. Conversely, a large number of levels diminishes the size of the plateaus with identical function values, reducing the risk of stagnation. These effects, however, are mitigated as the dimensionality rises. For example, in the 20-dimensional space with 100 levels, the success rate decreases to a mere 0.13 on the ellipsoid function.

When applied to the sphere function, the int-EA performs comparably to the CMA-ESwM, successfully resolving the problem nearly every time, regardless of the discretization (Figure 5.4 right column). However, in the case of the ellipsoid function, particularly at moderate numbers of levels and in higher dimensions, the int-EA achieves a higher success rate than the CMA-ESwM. In settings with discretization levels above 1 000, the ability of the int-EA to find the solution diminishes significantly. Consequently, while there are scenarios where the CMA-ESwM performs worse than the int-EA, for cases involving a large number of levels, the CMA-ESwM emerges as the preferred algorithm. For a small number of levels, both int-EA and CMA-ESwM exhibit comparable performance.

To enhance the capability of the standard CMA-ES in addressing the challenges posed by discretization, three modifications are considered. A BBOB problem is defined within the search space $\mathbf{x} \in [-5, 5]^d$. Especially in higher dimensions, the probability of the initial population being generated entirely outside of this space increases. Therefore, the repair method projection is employed for box constraint handling (Section 2.4.2). This significantly improves the performance of CMA-ES (Figure 5.5 left column) by preventing stagnation beyond the given lower and upper bounds. With the box-constraint handling, on the sphere function for 5 and higher numbers of levels, the success rate approaches 1.0 across the considered dimensions, with the exception of the higher dimensions 20 and 40, where the success rate drops to approximately 0.9 and 0.7, respectively. The performance is still suboptimal for a very small number of levels, such as two and three. For the ellipsoid function, CMA-ES with box constraint handling continues to face challenges, particularly in solving problems with a moderate number of levels.

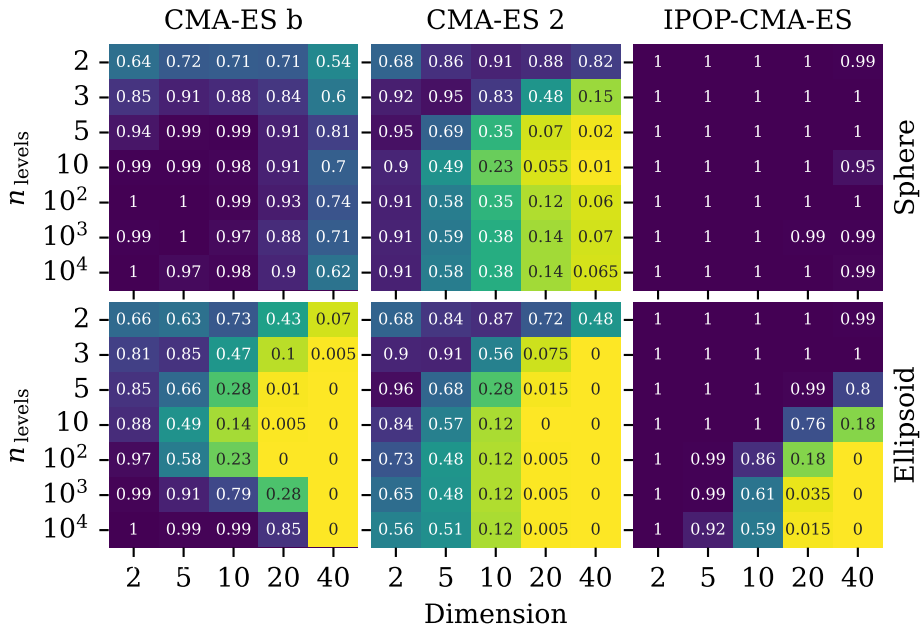


**Figure 5.5:** Success rates for CMA-ES with box-constraint handling (CMA-ES b), CMA-ES with a doubled default population size (CMA-ES 2) and CMA-ES with an IPOP restart strategy (IPOP-CMA-ES) on the sphere function (top row) and ellipsoid function (bottom row), across various dimensions (x-axis) and discretization levels $n_{\text{levels}}$ (y-axis). Each algorithm is given a budget of 2 000 evaluations per dimension, with 200 runs conducted for each setting. The target function value is set to $10^{-8}$.

Doubling the default population size of CMA-ES generally improves performance across all considered settings (Figure 5.5 center column). The primary advantage of a larger population is the increased probability of sampling individuals outside of the plateaus created by discretization. This enhancement enables CMA-ES to escape plateaus, thus mitigating the risk of stagnation more effectively. However, despite these improvements, stagnation of the optimization progress does occur in some runs across all settings, with a higher incidence in scenarios involving higher dimensions and larger numbers of levels.

The adoption of a restart strategy with an increasing population size affords CMA-ES the opportunity for additional runs in the event of early stagnation, provided that the evaluation budget has not been fully expended. This approach achieves a success rate approaching 1.0 in the majority of settings considered (Figure 5.5 right column). However, while a restart is potentially beneficial, a restart does not inherently preclude the possibility of subsequent stagnation. Notably, in the case of the 40-dimensional ellipsoid function and 100 or more levels, the majority of runs still fail to find a solution and continue to stagnate. Therefore, employing a restart strategy should be considered a measure of last resort.

To address the low success rates of the CMA-ESwM on the ellipsoid function with a moderate number of levels, both an increased population size and a restart strategy are applied to the CMA-ESwM (Figure 5.6). At first, the combination of a larger population size and the margin extension (CMA-ESwM) synergistically enhances the capacity of CMA-ES to escape the plateaus. As a result, doubling the default population size of the CMA-ESwM leads to a significant increase in success rates, particularly for the more challenging settings in 20 dimensions with 5 or more levels. However, particularly for 10 levels, the int-EA still slightly outperforms CMA-ESwM (Figure 5.6 left column).

The addition of a restart strategy significantly improves the performance of CMA-ESwM, which is now able to solve all problem settings except in cases with 100 or more levels in 20 and 40 dimensions, where the success rate does not reach 1.0 (Figure 5.6 center column). When the restart strategy is combined with the increased population size, CMA-ESwM consistently finds the solution across all considered settings, with the sole exception of the three settings with 100 or more levels in 40 dimensions on the ellipsoid function (Figure 5.6 right column). It is important to note that in these particular settings, the int-EA does not exhibit superior performance either (Figure 5.4 right column).

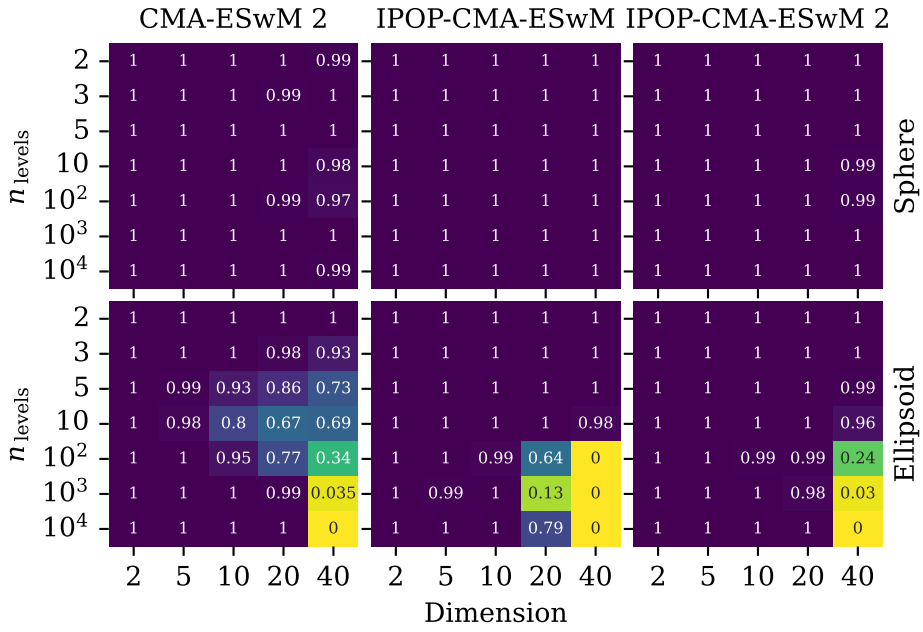


**Figure 5.6:** Success rates for CMA-ESwM with a doubled default population size (CMA-ESwM 2), CMA-ESwM with an IPOP restart strategy (IPOP-CMA-ESwM) and CMA-ESwM with an IPOP restart strategy and a doubled default population size (IPOP-CMA-ESwM 2) on the sphere function (top row) and ellipsoid function (bottom row), across various dimensions (x-axis) and discretization levels $n_{\text{levels}}$ (y-axis). Each algorithm is given a budget of 2 000 evaluations per dimension, with 200 runs conducted for each setting. The target function value is set to $10^{-8}$.

### 5.4.2 ECDF

So far, the success rate has only been considered after a predefined budget of evaluations. To assess the convergence performance over time, success rates can be illustrated with the ECDF over the complete evaluation process. The performance of IPOP-CMA-ESwM with a doubled population size on the discretized 10-dimensional ellipsoid function is compared with the standard CMA-ES on the continuous 10-dimensional ellipsoid function (Figure 5.7).

The ECDF for the IPOP-CMA-ESwM on the discretized ellipsoid function closely resembles that of the CMA-ES on the continuous ellipsoid function in shape. All runs achieve the target threshold of $10^{-8}$ within the allocated budget of 20 000 evaluations, and practically no runs stagnated. However, the ECDF curves reveal differences in the number of evaluations required to attain specific success rates.

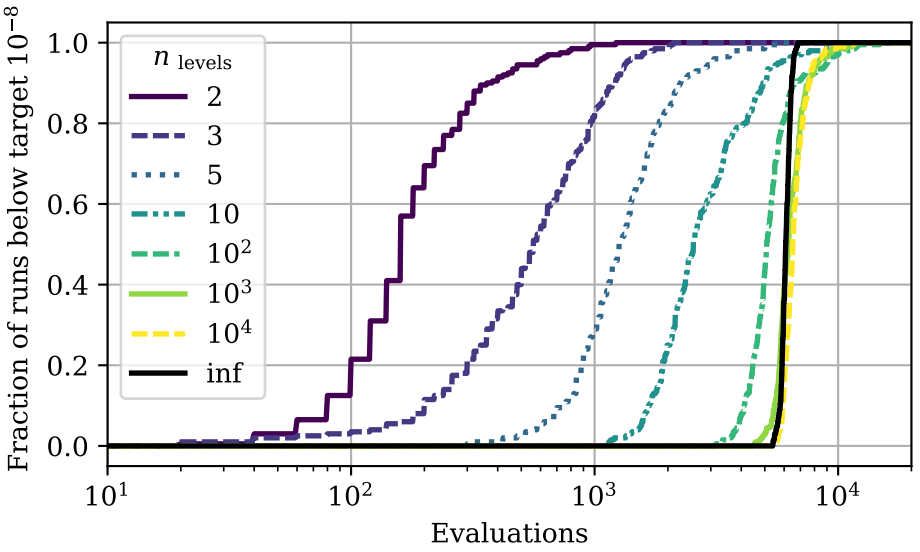


**Figure 5.7:** Empirical Cumulative Distribution Function for the IPOP-CMA-ESwM with a doubled default population size on the 10 dimensional ellipsoid function for $n_{\text{levels}} \in \{2, 3, 5, 10, 10^2, 10^3, 10^4\}$ and the standard CMA-ES for $n_{\text{levels}} \in \{\text{inf}\}$. The target function value is set to $10^{-8}$ and 200 runs are conducted for each setting.

For settings with 10 or fewer levels, the number of evaluations required to achieve a given success rate decreases with fewer levels and is consistently lower than that required for the standard CMA-ES on the continuous function. Therefore, CMA-ESwM seems to benefit from discretization, transforming the negative effects of discretization on standard CMA-ES performance into a positive outcome when the margin extension is incorporated into CMA-ES.

In the setting with 100 levels, only for the last 10 percent, as the success rate approaches 1.0, the number of evaluations required by the IPOP-CMA-ESwM exceeds that of the standard CMA-ES on the continuous function. This increase in required evaluations can be attributed to the need for a restart to achieve a success rate of 1.0. Without such a restart, CMA-ESwM achieves only a success rate of 0.95 (Figure 5.6 left column).

For 1 000 and 10 000 levels, the ECDF curves closely align with the continuous case. However, particularly towards the latter stages of optimization, CMA-ESwM requires a slightly higher number of evaluations to achieve a success rate of 1.0 across all runs compared to the CMA-ES on the continuous function. This observation means that discretization can slow down convergence somewhat but does not prevent the

algorithm from being successful. Remarkably, for both the 1 000 and 10 000 levels, a success rate of 1.0 is achieved with a reduced number of evaluations relative to the setting with 100 levels, which implies that a higher number of levels aids in facilitating convergence. Additionally, the absence of stagnation in individual runs indicates that a restart strategy is not required for these settings to attain a success rate of 1.0 (Figure 5.6 left column).

For the 20-dimensional discretized ellipsoid function, the ECDF curves show a pattern similar to those observed in the 10-dimensional settings (Figure 5.8). Notably, for 10 levels, the ECDF curve displays a slight inflection at a success rate of 0.4, suggesting that restarts have been conducted. In the absence of restarts, the success rate for CMA-ESwM is 0.67 (Figure 5.6 left column). For 100 levels, the success rate without restarts increases to 0.77, the inflection in the ECDF curve occurs later, at a higher success rate. This suggests a slower convergence speed compared CMA-ES on the continuous ellipsoid function. Similarly, at 1 000 and 10 000 levels, the convergence speed is slightly slower than that observed in the 10-dimensional case when benchmarked against CMA-ES on the continuous ellipsoid function.

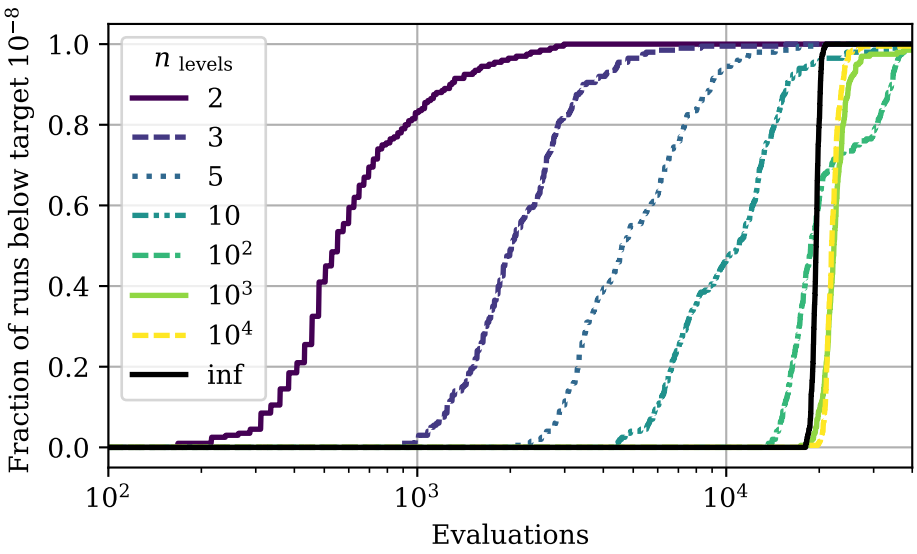


**Figure 5.8:** Empirical Cumulative Distribution Function for the IPOP-CMA-ESwM with a doubled default population size on the 20 dimensional ellipsoid function for $n_{\text{levels}} \in \{2, 3, 5, 10, 10^2, 10^3, 10^4\}$ and the standard CMA-ES for $n_{\text{levels}} \in \{\text{inf}\}$. The target function value is set to $10^{-8}$ and 200 runs are conducted for each setting.

## 5.5 Conclusion

This chapter introduced a method for discretizing continuous optimization problems by creating a grid with a specified number of discrete levels per dimension. The discretized problems are then used to assess the impact of different levels of discretization on the performance of an optimization algorithm. The analysis showed that the discretized ellipsoid function is more difficult to solve than the discretized sphere function.

The standard CMA-ES struggles with discretized spaces. This issue is addressed by employing CMA-ESwM for integer handling. This variant improves the performance on the sphere function and on the ellipsoid function across all levels.

The int-EA, here used as a benchmark algorithm, outperforms CMA-ESwM on the ellipsoid function for moderate numbers of discrete levels and particularly in higher dimensions. However, for higher numbers of levels, the int-EA cannot compete with the CMA-ESwM on the ellipsoid function across all considered dimensions. The combination of CMA-ESwM with the IPOP restart strategy and an increased initial population size outperforms the int-EA in all considered settings.

The convergence speed of the modified CMA-ESwM compared to the standard CMA-ES on the continuous ellipsoid function is not significantly impeded across the majority of considered settings. In fact, the discretization with small numbers of levels simplifies the difficulty of the problem, and convergence of the CMA-ESwM is accelerated. Nonetheless, for 100 levels, the convergence speed of CMA-ESwM is, on average, slightly reduced towards the end of the optimization process because of the need for restarts, particularly in higher dimensions such as 20.

In the 40-dimensional setting with a higher number of levels, neither the IPOP-CMA-ESwM with an increased population size nor int-EA managed to solve these settings effectively. Therefore, to solve these settings, another strategy or optimization algorithm is required. However, up to 20 dimensions, CMA-ESwM proved to be a robust optimization algorithm for wide numbers of discrete levels, from binary optimization problems with just two levels to settings with over 10 000 levels.

In conclusion, the results from this chapter reveal the capability of CMA-ESwM to address discretized optimization problems across diverse dimensions and numbers of discrete levels. The use of a restart strategy and an increased population size is highly recommended.