# Applications of Dynamic Parameter Control in Evolutionary Computation

Gregor Papa

Jožef Stefan Institute
Ljubljana, Slovenia
gregor.papa@ijs.si

https://gecco-2021.sigevo.org/


2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07...$15.00.
doi:10.1145/3449726.3461435

## Instructors

**Gregor Papa** is a Senior researcher and a Head of Computer Systems Department at the Jožef Stefan Institute, Ljubljana, Slovenia, and an Associate Professor at the Jožef Stefan International Postgraduate School, Ljubljana, Slovenia. He received the PhD degree in Electrical engineering at the University of Ljubljana, Slovenia, in 2002. His research interests include meta-heuristic optimisation methods and hardware implementations of high-complexity algorithms, with a focus on dynamic setting of algorithms' control parameters. He regularly organizes conferences and workshops in the field of nature-inspired algorithms from the year 2004 till nowadays. He led and participated in several national and European projects.



- **Previous tutorials:** A series of tutorials by Carola Doerr, followed by our common tutorials
  - Theoretical, shortened first part, is based on previous slides of Carola [80] (some texts and figures are taken from there)
- **Version Management:** Slides are regularly updated. You can find the latest version on:
  - http://cs.ijs.si/papa/files/GECCO2021tutorial.pdf
- **Reference of this tutorial:**

```
@inproceedings{Papa2021,
  author = {Gregor Papa},
  title = {Applications of Dynamic Parameter Control in Evolutionary Computation},
  booktitle = {2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21
Companion)},
  date = {July 10–14},
  year = {2021},
  place = {Lille, France},
  publisher = {ACM},
  url = {https://doi.org/10.1145/3449726.3461435},
  isbn = {978-1-4503-8351-6/21/07},
  doi = {10.1145/3449726.3461435}

}
```



## Survey Articles

Summaries of the state-of-the-art techniques and a good starting point for your research can be found in the following surveys
(extensive reference list is at the end)

- **Empirical works**
  - Karafotias, Hoogendoorn, Eiben, 2015 [KHE15]
    (detailed survey of empirical works)
  - Aleti, Moser, 2016 [AM16]
    (systematic literature survey with additional pointers)
  - Eiben, Hinterding, Michalewicz, 1999 [EHM99]
    (classic seminal paper, introduced a widely accepted classification scheme)
  - Lobo, Lima, Michalewicz, 2007 [LLM07]
    (book on parameter selection, includes chapters on tuning and control)
- **Theoretical works**
  - Doerr, Doerr, 2020 [DD20]
    (surveys theoretical works which prove performance bounds with mathematical rigor; introduces the revised classification scheme)

## Outline of the Tutorial

- **Part 1:** Introduction
  *What are the goals of parameter control?*
- **Part 2:** Motivation
  *What are the basics?*
- **Part 3:** Taxonomy of Parameter Control Mechanisms
  *Which parameter control techniques exist?*
- **Part 4:** Real-world optimization
  *What are the characteristics of the real world problems?*
- **Part 5:** Applications of Parameter Control
  *Where is parameter control used in practice?*
- **Part 6:** Wrap Up
  *What's next?*

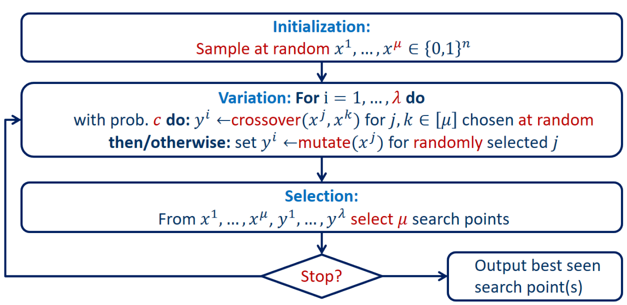


**Introduction**



## Parameters in iterative / evolutionary optimization

A "typical" evolutionary algorithm, a $(\mu+\lambda)$ GA with crossover




$\rightarrow$ There are several parameters that need to be decided: population size, crossover rate, mutation rate, selective pressure, etc.



## The importance of control parameter values

- The very early days of EC:
  - "EAs are robust problem solvers" and need no tuning of parameters
- It was soon realized that this is not true
  - i.e., the "no free lunch" theorem [104]

$\rightarrow$ It is widely acknowledged today that the control parameter values have a decisive influence on the performance of an EA.

## The importance of control parameter values

**How to find good parameter values?**

- Finding optimal parameter values is far from being trivial
- Small changes in one parameter can cause huge performance gaps
- The optimal parameter values for one problem, might be much different for similarly-looking problems
- Optimal parameter values can change during the optimization process



## Goals of Parameter Control

- To identify good parameter values "on the fly"
  - When prior training or tuning is not possible
  - $\rightarrow$ Integrate the tuning procedure into the optimization process
- To track good parameter values when they change during the optimization process
  - They are not only constant factors
  - Significant performance gains are possible



## Controlling Multiple Parameters

- Most EAs have several parameters
- There is no reason to not control more than one or even all of them
- Some works on controlling more than one parameter exist, [55]
- The problem how to best control several parameters at the time is widely open



**Motivation**

## The LeadingOnes Problem

- Classic benchmark problem often studied in the theory of evolutionary computation [85]
  - One of the simplest examples of a non-separable function
  - to test the performance of evolutionary algorithm
- Function

  $\text{LO}:\{0,1\}^n \to \mathbb{R}, x \mapsto \text{LO}(x) = \max\{i \in [n] \mid \forall j \leq i : x_i = 1\}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |

  - LO $= 2$ (two initial ones)



## The (1+1) EA example

> Initialization:

Choose $x \in 0, 1^n$ uniformly at random

> Optimization: in iteration $t = 1, 2, \ldots$ do

1. Mutation:
   - create $y$ from $x$ by standard bit mutation (flip each bit with probability $p$, independently of other bits)
2. Selection:
   - if $f(y) \geq f(x)$ $\rightarrow$ replace $x$ by $y$

**Critical parameter**

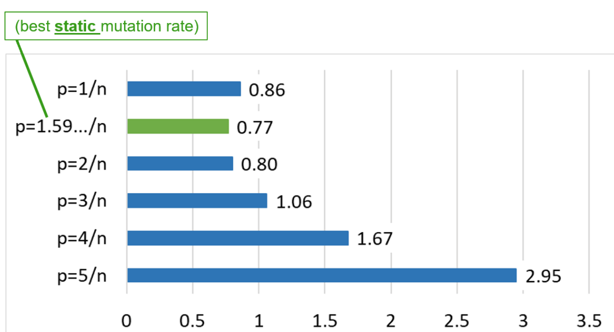
- The mutation rate $p$

  $\rightarrow$ often recommended as $p = 1/n$



## Theoretical Optimization Times

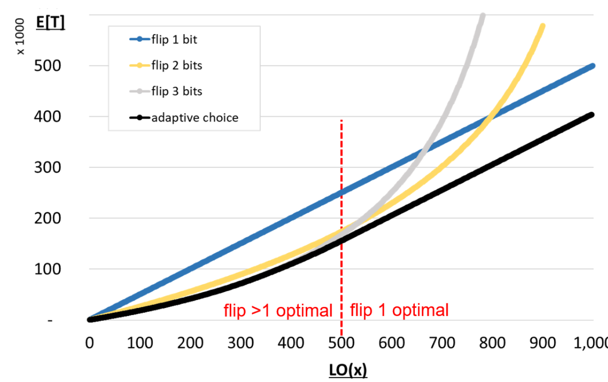
- The expected optimization time of the (1+1) EA on LeadingOnes [12]
  - $\frac{1}{2p^2}\left(\frac{1}{(1-p)^{n-1}} - 1 + p\right)$





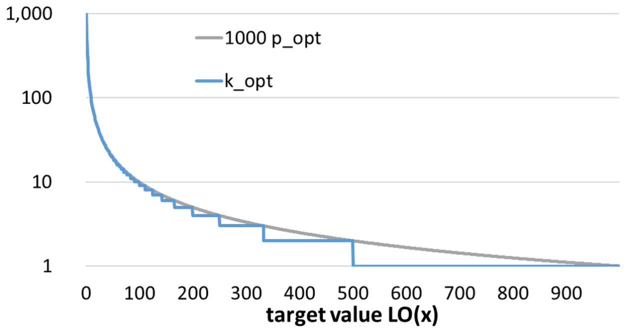

## Fixed-Target Running Times

- Expected fixed-target running times for dimension n=1000



## Mutation Rates Optimality

- Expected fixed-target running times for dimension n=1000

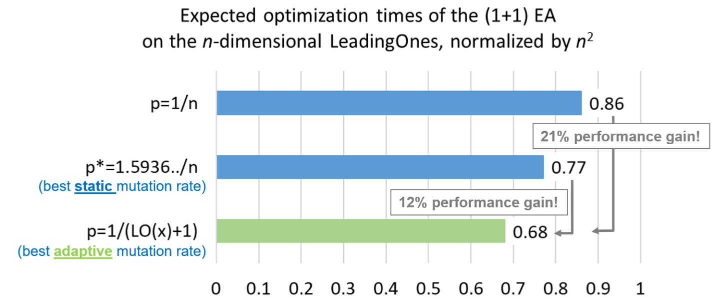



$p_{opt} = \frac{1}{\mathrm{LO}(x)+1}$ [12, 92]

$k_{opt} = \lfloor \frac{1}{\mathrm{LO}(x)+1} \rfloor$ [37, 35]



## Gain of Dynamic Mutation Rates




Measured results from [12]



## (1+1) EA with Adaptive Mutation Rates

> Initialization:
  Choose $x \in 0, 1^n$ uniformly at random

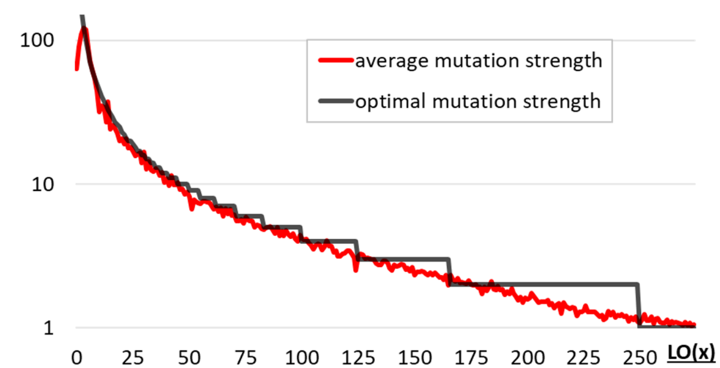
> Optimization: in iteration $t = 1, 2, \ldots$ do
  1. Mutation:
     - create $y$ from $x$ by standard bit mutation (flip each bit with probability $p$, independently of other bits)
  2. Selection + Update:
     - if $f(y) \geq f(x)$
       $\rightarrow$ replace $x$ by $y$
       $\rightarrow$ replace $p$ by $Ap$
     - if $f(y) < f(x)$
       $\rightarrow$ replace $p$ by $bp$

- Based on a variant from [37]



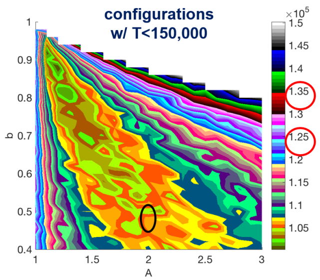
## Tested mutation rates

- Results for LeadingOnes $n$=500, based on [37]
  - Update strengths: $A = 2$, $b = 1/2$
  - Plot compares average number of bits flipped (red) vs. optimal number (black)
  - Logarithmic scale with zoom into $\mathrm{LO}(x) \leq 250$



## Fitness Landscape of Tuning Problem

- The performance gain is not very sensitive with respect to the choice of the hyper-parameters $A$ and $b$ [37]
- Heatmap on average optimization time for combinations of $A$ and $b$ for the adaptive (1+1)EA on $n = 500$ LeadingOnes
  - $A = 2$, $b = 1/2$ gives an average runtime of ≈104,000 function evaluations
  - the static (1+1) $EA_{>0}$ needs ≈135,000 function evaluations
  - RLS needs ≈125,000 function evaluations






## The 1/5-Success Rule

- One of the most famous success-based parameter adaptation rule
- Rechenberg observed that for the sphere function and a corridor landscape the optimal success rate of the (1+1) ES is around 1/5 [82]
- Approach:
  - If (observed success rate $> 1/5$) → increase mutation rate
    Informal interpretation: we seem to be in an easy part of the optimization problem → increasing mutation rates might result in larger progress per step
  - If (observed success rate $< 1/5$) → decrease mutation rate
    Informal interpretation: we could be approaching an optimum and should focus our search → decrease mutation rate for a more conservative search

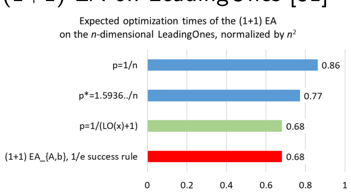
> Similar rules have been proposed by [89] and [29]

> The same idea can also be used to control other parameters, such as the population size, crossover probabilities, etc.

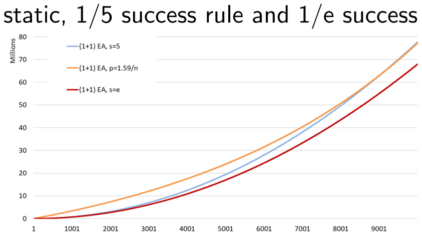


## 1/5 Success Rule in Discrete Optimization

- The 1/e success rule yields optimal mutation rates for the (1+1) EA on LeadingOnes [31]




- Average Fixed-Target Running Times for the (1+1) EA with static, 1/5 success rule and 1/e success rule mutation rates






**Taxonomy**

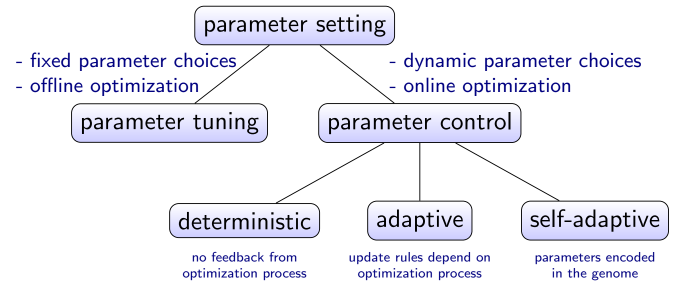
## Main Questions in Parameter Control

1. Which parameter is adapted? (and what is affected: individual / population)
   - Population size
   - Mutation rate, Crossover probability
   - Selection pressure
   - Fitness function (e.g., penalty terms for constraints)
   - ...
2. What is the basis/evidence for the update?
   - Time elapsed: number of evaluations, generation count, CPU time
   - Progress (e.g., in terms of absolute or relative fitness gain)
   - Diversity measures
   - ...
3. How do we update the parameter(s):
   - Multiplicative updates
   - Learning-inspired parameter selection
   - Endogenous/self-adaptive parameter selection
   - Hyper-heuristics
   - ...



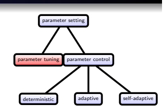
## Initial Classification Scheme

The most popular classification scheme is the one of Eiben, Hinterding and Michalewicz [39]





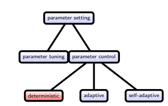

## Parameter Tuning



- Typical tuning approach
  - Run some initial tests and observe how the performance depends on the chosen parameter values
  - Choose the parameter values that seem the most promising
  - → Requires a (large) budget for the training
  - Recent survey on automatic parameter tuning methods for metaheuristics [50]
- Several parameter tuning tools are available
  - irace [68], SPOT [8], GGA [3], ParamILS [52], SMAC [51], HyperBand [62], BOHB [40], ...
  - Advantage: automated identification of reasonable parameter values → supports human and reduces bias
  - Disadvantage: recommended parameter values are static!
  - Note: even when focusing on dynamic parameter choices, parameter tuning can be very essential to select good hyper-parameters [9]

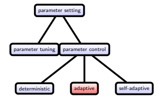


## Deterministic Parameter Control



- Optimal parameters often follow a similar pattern
  - E.g. first allow for exploration, then for exploitation
  - Time-dependent parameter settings can be used
    - time = number of generations, fitness evaluations, wall-clock time, etc.
- Examples:
  - Cooling schedule of the selective pressure ("temperature") in Simulated Annealing
  - Start with some (large) mutation rate $p(0)$, decrease $p$ after every 10,000 fitness evaluations
  - After each 1,000 iterations, draw a random mutation probability
- More suitable terms would be "time-dependent" or "scheduled" update scheme
- Note: finding the optimal deterministic update rules requires tuning

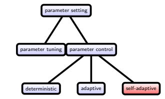
## Adaptive Parameter Control



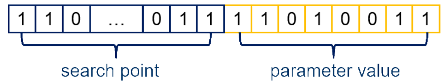
- Global estimate for parameter quality, not individual-based
  - Feedback from the optimization process
  - Change the parameters according to some pre-described rule
- Relevant feedback includes
  - Function values of the search points in the population
  - Diversity of the search points
  - Absolute or relative progress obtained within the last $\tau$ iterations
- Examples
  - 1/5-success rule
  - CMA-ES update of covariance, step size, population size
  - ...



## Self-Adaptive Parameter Control



- Use an EA to determine parameter values
- Many different ways to do this
  - Create a new population of parameter values, choose from this parameter values, possibly apply variation to them, and employ them in EA, select based on progress made
  - Append to the solution candidates a string which encodes the parameter value, first mutate the parameter value part, then use this parameter to change the search point, selection as usual

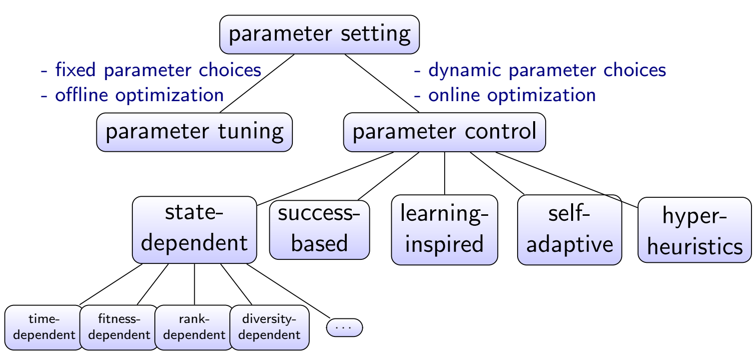



  - Improve parameter selection and fitness at the same time
- Some theoretical works on a self-adaptive choice of the mutation strength, [24] and [34]



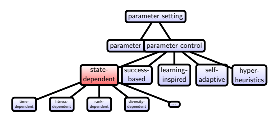
## Revised Classification Scheme

Revised classification from Doerr and Doerr [30] to better fit current research directions and allow more consistent use of terms

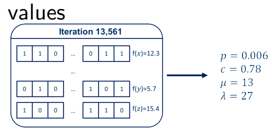





## State-Dependent Parameter Selection

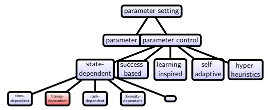


- Selection mechanisms do not depend on the history of the optimization process, but on the current state
  - E.g., a snapshot of the current population is maped to parameter values




- Most commonly used indicators
  - Time elapsed (# fitness evaluations, iteration counter, CPU time, ...) → corresponds to "deterministic" parameter setting in the classification of [39]
  - Function values (absolute values, diversity, ranks, etc.)
  - Genotypic properties (e.g., diversity of the population)

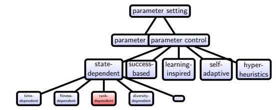
## Fitness-Dependent Parameter Selection




- Requires a good understanding of how the parameters should depend on the function values
- Empirically
  - [5], [6], [41] for OneMax
- Theoretically
  - [20], [32], [7] for OneMax and [35], [36], [66] for LeadingOnes



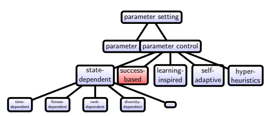
## Rank-Dependent Parameter Selection




- Bad search points should undergo large variation ($\rightarrow$ large mutation rates)
- Good individuals should be modified only moderately ($\rightarrow$ small mutation rates)
  - Example, from [23]
    - Rank search points in the current population
    - Each search point is assigned a mutation rate that depends on its rank:
      rank 1: mutation rate $p_{min}$ // best individual of population
      ...(linear interpolation)
      rank s: mutation rate $p_{max}$ // worst individual of population
    - The rank-based GA first selects an individual from the population and then modifies it with the mutation rate given by this ranking



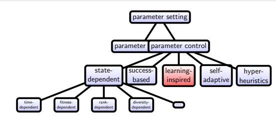
## Success-Based Parameter Selection




- After each (or after every $\tau$) iteration(s) adjust the current parameter value depending on whether or not the last ($\tau$) iteration(s) have been successful
  - Examples
    - Finding a strictly better search point
      this is probably the most common measure
    - Finding a search point that is integrated into population
      used by the adaptive (1+1) $\text{EA}_{>0}$
    - Finding a fitness-increase of at least x%
    - Finding point(s) that increase the diversity of the population
    - ...
  - Success-based parameter selection is classified as "adaptive parameter control" in the taxonomy of [39]



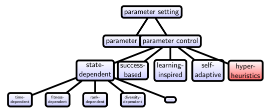
## Learning-Inspired Parameter Control




- To have a set of possible parameter values according to some rule
- Test one (or some) of these values based on the feedback from the optimization process
- Update the likelihood to employ the tested value
  - Example through multi-armed bandits (MAB)
    - $\kappa$ experts in each round
    - You have to chose one of them and you follow their advice
    - You update your confidence in each expert depending on the quality of their forecast
  - Key questions:
    - How to UPDATE the confidences?
    - How to SELECT based on the confidences (greedy, random in proportion to confidence, etc.)

## Hyper-Heuristics

- Hyper-heuristics covers much more than controlling parameters
  → the main idea is to control the whole algorithm, in the sense of dynamically choosing which heuristic is best at a given state
- Surveys: [19], [93]
- Recent theoretical works: [66], [33], [67]






**Real-world optimization**



## Real-world optimization

- Real-world optimization problems occur in many applications
  - Engineering design,
  - Scientific modelling,
  - Image processing,
  - Production,
  - Transportation,
  - Bioinformatics,
  - Finances, etc.
- Real-world systems are, in general, large and very complex. They need to process a large amount of data, to perform complex optimization and make decisions fast [56].



## Real-world optimization

- Real-world problems in general
  - Contain non-linear objective functions of mixed design variables (i.e. continuous and discrete)
  - Contain linear as well as non-linear constraints
  - Might have several local optima
- For a wide range of real-world optimization problems, a near-optimal or a better-than-known solution is considered a satisfactory result of an optimization problem.

## Real-world optimization

- There are several characteristics that increase the complexity of the optimum solution search and for which parameter control could be advantageous
  - **Number and type of variables**: a large number of decision variables, including mixed-integer problems, where different types of variables are optimised
  - **Dynamic problems**: problems that are changing over time
  - **Problems under uncertainty**: the variables of the problem have some uncertainty
  - **Number of objectives**: problems that require optimizing more than one objective function simultaneously and need to be solved by a multi/many-objective approach
  - **Nested problems**: multi/bi-level optimization, where one optimization problem has another optimization problem as a constraint
- Some problems have combination of these characteristics



## Real-world optimization

- Properly defined control parameters play a crucial role in effectively handling the mentioned characteristics and solving such problems.
  - For example: with increasing dimensionality of the problem its landscape complexity grows and the search space increases exponentially.

**But**

An optimization algorithm must still be able to explore the entire search space efficiently



**Large scale global optimization**



## Large scale global optimization

- LSGO [75] - the problem dimension $D$ (the number of variables to be optimised) has an order of magnitude of up to several thousand (for real values) or even billions (for integer or binary values)
- An active research field due to the growing number of large-scale optimization problems in engineering, manufacturing and economy applications (such as bio-computing, data or web mining, scheduling, vehicle routing, etc.) [21], [64]
  - Advances in machine learning and the wide use of deep artificial neural networks result in optimization problems with over a billion variables [49]

## Large scale global optimization

**A major challenge of large-scale optimization**

Most engineering problems have an exponential increase in the number of required decision variables [76], [98]

- The challenges motivated the design of many kinds of efficient, effective, and robust kinds of metaheuristic algorithms to solve LSGO problems with high-quality solutions and high convergence performance as well as with low computational cost [72]



## Large scale global optimization

- To achieve acceptable results even for the same problem, different parameter settings along with different reproduction schemes at different stages of optimization process are needed
- Several techniques (e.g., [110], [26]) have been designed to adjust control parameters in an adaptive or self-adaptive manner (instead of a trial-and-error procedure)



## Large scale global optimization

- Some examples of LSGO
  - Data analytic and learning problems [116]
  - Shape design optimization for aircraft wings and turbine blades [106]
  - Satellite layout design [95]
  - Parameter calibration of water distribution system [103]
  - Seismic waveform inversion [100]



**Dynamic optimization**

## Dynamic optimization

- Real-world optimization problems are usually subject to changing conditions over time
- The effects of these changes could influence several aspects of the problem, such as the objective function, the problem instance, its constraints, etc.
- The optimal solution of the problem might change over time.

**Dynamic optimization**

Changing problems, when solved by an adaptive optimization algorithm on-the-fly, are called dynamic optimization problems (DOPs) [74]



## Dynamic optimization

- The algorithm is expected to be able to track the current optimal solution as well as the changing optimal solution over time
- The optimization procedure has to be able to detect these changes and react quick enough
  - This also requires dynamic change of the ratio for exploration and exploitation parts of the search
  - Both adaptive [101] and self-adaptive [17] parameter control can be used



## Dynamic optimization

- Based on a comprehensive survey [13], four different strategies can be used to help evolutionary/population-based algorithms to adapt in dynamical environments:
  - **Increasing diversity** of the population after a change is detected, (e.g., by increasing mutation rate every $N$ generations)
  - **Maintaining diversity** throughout the run, to avoid convergence of the population on one point
  - **Memory based approaches**, taking into consideration older solutions and sometimes making predictions based on historical data
  - **Multi-population approaches**, where many small populations track their own peaks as the environment changes



## Dynamic optimization

- Some examples of dynamic optimization
  - Production scheduling [109]
  - Energy demand optimization [44]
  - Transportation [108], [65]
  - Financial optimization [48]

**Optimization under uncertainty**



## Optimization under uncertainty

- The presence of (a range of) uncertainties has to be taken into account for solving many real-world applications with evolutionary algorithms
- [54] categorize the uncertainties that influence EA performance into four types
  - When there is some **noise in the fitness function**
  - When there are changes of **design and environmental parameters after the optimization**
  - When **fitness function is an approximation**
  - When the **optimum changes over time** (as in dynamic optimization).



## Optimization under uncertainty

- Methodologies for addressing noisy fitness function
  - *Explicit averaging* by calculating the average of the fitness values over a number of randomly sampled disturbances [47], [71]
  - *Implicit averaging* sample size as an inverse function of the population size [42]
  - *Fitness inheritance* where the offspring inherits also the mean and standard deviation of the objective value [18]
  - *Selection modification* [94]
- These methods assume that the search space follows a homogeneous noise distribution, such as a uniform or a normal distribution [97]



## Optimization under uncertainty

- Some examples of optimization under uncertainty
  - Financial optimization [48]
  - Transportation in unknown environments [111]
  - Space applications [99], [10]

**Multi-objective optimization**



## Multi-objective optimization

- Multi-objective (and also many-objective) optimization approaches are used for optimization problems where several criteria need to be optimised, but they are equally treated and not merged (e.g., by weights) into one single objective
- The output of multi-objective optimization is a set of solutions that approximates the Pareto front
- There is no unique measure that would indicate how good a current approximation of the Pareto front is

**In multi-objective cases**

Adaptive parameter control is a bit more complicated to design and additional considerations are needed to design phenotype feedback collection part



## Multi-objective optimization

- Possible assessment of the optimization process stage
  - Monitor the proportion of non-dominated solutions in the population [105]
  - Convergence detection [73]
- The most common indicators that are also used as input to parameter control are the crowding distance and the contributing hyper-volume [53], [11]



## Multi-objective optimization

- Other metrics can also be applied
  - $\varepsilon$-dominance
  - Generational distance
  - Delta indicator
  - Two set coverage, and so on [83].
- Compared to adaptive control, self-adaptive control is easier to design and implement because less modifications are needed to upgrade an existing multi-objective optimization algorithm [102] [22].

## Multi-objective optimization

- Some examples of multi-objective optimization
  - Engineering design [46]
  - Transportation [88], [61]
  - Production [45], [4]



**Multilevel optimization**



## Multilevel optimization

- In many real-world processes there is a hierarchy of decision-makers and decisions are taken at different levels [70]
  - The constraint domain associated with a multilevel problem is implicitly determined by a series of optimization problems which must be solved in a predetermined sequence
- The simplest form of a multilevel problem has two levels (i.e., bi-level optimization problem)
  - The optimization of such problem aims to achieve the optimum solution of the upper level, while the optimum of the lower optimization level is also taken into account
  - Since the lower level landscape changes for every upper level vector, parameter control seems to be useful approach



## Multilevel optimization

- An interesting application of bilevel optimization is connected to parameter tuning of EAs as bilevel optimization problem
  - [91] propose the parameter tuning problem as an inherently bilevel programming problem involving algorithmic performance as the objective(s)
  - [1] created a bilevel framework for parallel tuning of optimization control parameters, and compared it to irace proving that it can be competitive
  - Bilevel control parameter tuning can be used to design a parameter control mechanism [2]

**Examples**



## Image processing: Feature selection

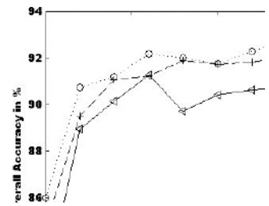
LS

- Feature selection for reducing the dimensionality in classification of hyperspectral images [27]

**Self-adaptive differential evolution SADE is used**

- SADE is used in combination with Fuzzy kNN classifier
- Compared to GA-based and ACO-based approaches [87]

- Significant improvement for overall classification accuracy and Kappa coefficient






## Image processing: Animated Tree Reconstruction

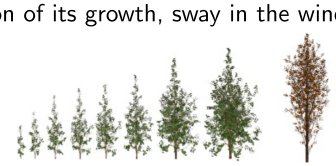
LS UN

- Feature extraction to reconstruct three dimensional procedural models of trees; to lower problem dimensionality needed for encoding local parameters [112]
- The reconstruction is iteratively optimized using DE, which samples procedural tree model parameters to obtain a parameterized procedural model for instantiating a geometrical model

**jDE is used**

- DE with self-adaptive control parameter settings [14]

- Examples of reconstructed model animation are shown, such as simulation of its growth, sway in the wind, or adding leaves





## Intravascular Ultrasound Image Analysis: Medicine

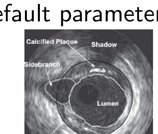
DY UN

- To optimize for the parameters of feature detectors in the multi agent image analysis system to detect lumen, vessel, shadows, sidebranches, and calcified plaques in IVUS images [63]

**MIES (mixed integer evolution strategies) algorithm is used**

- implements self-adaptation of the width of the mutation distribution (step size) [63]
- step size control is used for real, integer, and categorical variables

- The results show that MIES solutions performed better than or equal to the default expert solution
- MIES drawn contours are more smooth than contours detected with the default parameter settings



## Production: Scheduling

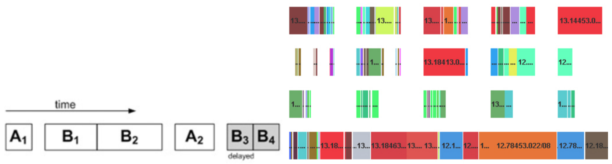
DY

- Optimal operations scheduling in the production of different components considering various constraints [81]

**PLES algorithm is used**

- PLES is based on general GA with modified implementation of functions that allow varying population size, mutation and crossover [79]
- Compared to standard non-adapting GA and GA with customized local search [58]

- Faster convergence of PLES and comparable results to GA for various problem instances






## Transport: Scheduling

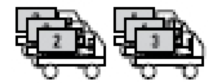
DY

- Solving a constrained transportation scheduling problem, for transporting goods in emergency situations [57]

**PLES algorithm is used**

- PLES is based on general GA with modified implementation of functions that allow varying population size, mutation and crossover [79]
- Compared to non-adaptive Ant-stigmergy algorithm [60]

- The satisfying performance in finding solutions and escaping from local optima, for different transportation modes





## Power systems: Scheduling

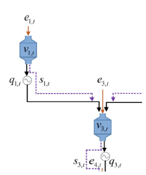
LS
DY

- Addressing total fuel costs and emissions minimization by appropriate hydro and thermal generation schedules [44]

**NPdyn$\epsilon$jDE and PSADEs algorithms are used**

- NPdyn$\epsilon$jDE is based on jDE self-adaptation [15], population size reduction [16], and $\epsilon$ level adjustment [113]
- Surrogate parallel self-adaptive DE (PSADEs) is based on self-adaptation [43], with pre-computed surrogate model

- The satisfied 24-h system demand is obtained by using a new DE architecture





## Optical network wavelength allocation

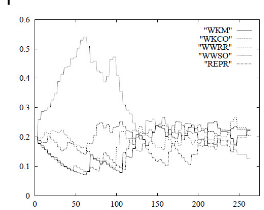
LS
DY

- Determining the route and wavelength to be used by each of the individual traffic requirements of multi-wavelength all-optical transport networks [90]

**GA algorithm is used**

- every few generations probabilities were recalculated, according to the success of the predecessors of the improved solution

- The results compare different sizes of adaptation



## Water pipeline system: Parameter calibration

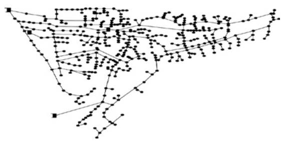
LS

- Parameter calibration strengthens the model accuracy of the water distribution system. Many factors influence the reliability of WDS simulation [103]

**ensemble optimization evolutionary algorithm (EOEA) is used**

- Combining global shrinking stage (to shrink the searching scope to the promising area) [16] and local exploration [107] stage with self-adaptive group sizing
- Different problems were constructed/tested: 100D, 200D, 300D and 454D

- Results show good scalability of EOEA on this real-world application





## Seismic waveform inversion: Configuration

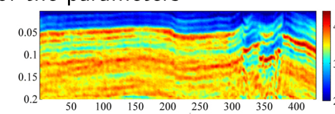
LS UN

- Waveform inversion for whole Earth geophysics and exploration geophysics, to develop an accurate Earth model and for understanding of subsurface structures [100]

**cooperative coevolutionary DE (CCDE) algorithm is used**

- All subcomponents are cooperatively evolved to solve high-dimensional optimization problems through decomposition
- The next generations are selected according to the global fitness values
- The parameter adaptation scheme of jDE [14] is used

- The CCDE results are very effective and have significant advantages over some other methods. CCDE is not sensitive to the size of the parameters






## Underwater glider: Path planning

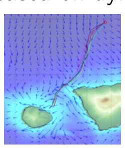
DY UN

- Optimization of a short-term sea trajectory, with opportunistic sampling of dynamic mesoscale ocean structures (eddies), which offer short-term opportunities for underwater glider path optimization [114] [111]

**jDE is used**

- Slightly modified jDE [14], combining DE and underwater glider path planning (UGPP)

- Gliders operational capabilities benefit from improved path planning, especially when dealing with opportunistic short-term missions focused on dynamic structures





## Underwater glider: Path planning 2

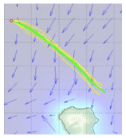
DY UN

- Optimization of a sea trajectory for underwater glider path optimization [115]

**Success-History Based Adaptive Differential Evolution Algorithm (SHADE) including Linear population size reduction (L-SHADE) is used**

- $L - SHADE_5$ was used including different population sizes and population sizing strategies

- Increased opportunity for mission scenario re-tests or in very hard scenarios



## Electrical motor: Geometry selection

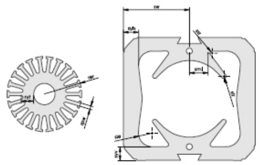
UN

- Optimization of geometrical parameters of the electrical motor rotor and stator geometry [78]

**PLES algorithm is used**

- PLES is based on general GA with modified implementation of functions that allow varying population size, mutation and crossover [79]
- Compared to generational evolutionary algorithm (GEA) [96] and multilevel ant stigmergy algorithm (MASA) [59]

- The results show fast convergence of the PLES but is not always able to find global optimum





## Target shape design: optimization

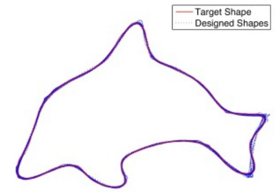
LS MO

- To tackle the target shape design optimization problems (TSDOPs) with B-spline as the geometry representation [106]

**CMA-ES-CC algorithm is used**

- CMA-ES with Cooperative Coevolution was implemented
- Compared with CMA-ES, iES [77], RCGA [28]

- The performance of CMA-ES-CC was stable, and the results of CMA-ES-CC were significantly better than with other EAs for TSDOPs






## Structure topology optimization

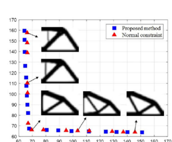
MO

- Topology optimization was performed with two different compliances (structure in different load cases are used as two different objective functions to be minimized) were considered as conflicting objective functions [86]

**Adaptive weight multi-objective algorithm is used**

- Conflicting objective functions are converted to a single objective function by applying weights, and these weights are adaptively updated to find evenly distributed solutions on the Pareto front.

- The results confirm that optimized solutions, obtained by using the proposed method, are evenly distributed on the Pareto front





## Robotic hand: configuration

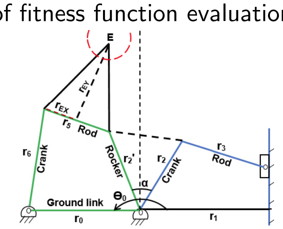
DY

- Design of a three-finger end effector of the robot [69]

**Differential Evolution with Combined Variants (DECV-S)**

- Dynamic control mechanism was added for the *F* parameter of DE
- Compared against the fine-tuned original version of the algorithm (DECV) and against the fine-tuned version of a DE-based approach

- DECV-S was able to provide similar results to those of the fine-tuned compared approaches, but with a considerable lower number of fitness function evaluations



## Robotic hand: configuration 2

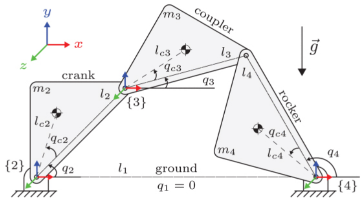
- The speed regulation of four-bar mechanisms in industrial processes [84]

DY MO

**multiobjective metaheuristic optimizers**

- Comparison of NSGA-II, multiobjective evolutionary algorithm based on decomposition and DE (MOEA/D-DE), S-metric selection evolutionary multiobjective algorithm (SMS-EMOA), nondominated sorting genetic algorithm III (NSGA-III), and HV-MODE

- The proposed adaptive controller tuning strategy shows to be effective for speed control of the FBM






## Pendulum: Reinforcement learning

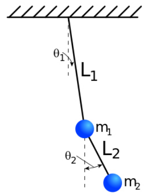
- Considers modeling of inverted pendulum and double-pendulum swing-up [25]

DY

**CMA-ES algorithm is used**

- CMA-ES uses reproducing kernel Hilbert space (RKHS)
- Compared to standard CMA-ES and adaptive CMA-ES direct policy search CMA-ES-A

- The results show that CMA-ES-RKHS is able to avoid local optima and clearly outperformes other methods





## Urban mass rapid transit: Transport

- Multi-objective simulation-based headway optimization for complex urban mass rapid transit systems in Vienna [88]

DY MO

**CMA-ES algorithm is used**

- multi-objective version of CMA-ES (MO-CMA-ES) is compared to single objective version (SO-CMA-ES) and NSGA-II

- The results show similar performance of all tested algorithms




## Aerobic fermentation process: Production

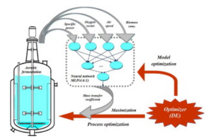
- The optimization of the oxygen mass transfer coefficient in stirred bioreactors, where the oxygen transfer in the fermentation broths has a significant influence on the growth of cultivated microorganism [38]

DY MO

**SADE-NN-1 algorithm is used**

- An improved, simple, and flexible self-adaptive variant of DE, in combination (hybridized) with neural networks

- The improvements (hybridization) of the algorithm resulted in higher efficiency of the whole methodology



**Summary**



## Summary

The distribution of presented cases according to problem characteristics

| large-scale | dynamic | uncertain | multi-objective |
|---|---|---|---|
| 7 | 6 | 12 | 5 |

The distribution of topologies (as presented earlier) used in presented cases:

- some state-dependent
- mostly success-based
- a few learning-inspired



## Summary 2

The need for dynamic parameter control.

- High
  - Traffic
  - Logistics / Transportation
  - Energy demand
- Moderate
  - Production
  - Finances
  - Data analytics
- Low
  - Parameter calibration



## Wrap Up

- You should be (now) convinced that
  - Dynamic parameter choices can help to significantly improve the performance of your EA
  - Already quite simple mechanisms can be surprisingly efficient
  - Research and work on parameter control can be fun ☺
  - Non-static control parameter values should be the new standard in the field ☺
- A lot needs to be done to make this change happen
  - Do not get frightened by the fact that (quite) some work has already been done.
    There is still much room for creativity and we are just starting to understand how good mechanisms look like!

→ If you get to work on parameter control, we would be very much interested in your results, positive and negative!

gregor.papa@ijs.si

## Acknowledgements


- This work was supported by the Slovenian Research Agency (research core funding No. P2-0098).
- This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 692286. (SYNERGY)




## References I


[1] Martin Andersson.
*A Bilevel Approach to Parameter Tuning of Optimization Algorithms Using Evolutionary Computing.*
PhD thesis, University of Skövde, 2018.

[2] Martin Andersson, Sunith Bandaru, Amos Ng, and Anna Syberfeldt.
Parameter tuning of MOEAs using a bilevel optimization approach.
In António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, editors, *Evolutionary Multi-Criterion Optimization*, pages 233–247, Cham, 2015. Springer, Springer International Publishing.

[3] Carlos Ansótegui, Yuri Malitsky, Horst Samulowitz, Meinolf Sellmann, and Kevin Tierney.
Model-based genetic algorithms for algorithm configuration.
In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 733–739. AAAI Press, 2015.

[4] Elnaz Asadollahi-Yazdi, Julien Gardan, and Pascal Lafon.
Multi-objective optimization of additive manufacturing process.
*IFAC-PapersOnLine*, 51(11):152 – 157, 2018.
16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[5] Thomas Bäck.
The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm.
In *Proc. of Parallel Problem Solving from Nature (PPSN'92)*, pages 87–96. Elsevier, 1992.

[6] Thomas Bäck.
*Evolutionary algorithms in theory and practice - evolution strategies, evolutionary programming, genetic algorithms.*
Oxford University Press, 1996.

[7] Golnaz Badkobeh, Per Kristian Lehre, and Dirk Sudholt.
Unbiased black-box complexity of parallel search.
In *Proc. of Parallel Problem Solving from Nature (PPSN'14)*, volume 8672 of *Lecture Notes in Computer Science*, pages 892–901. Springer, 2014.

[8] Thomas Bartz-Beielstein, Oliver Flasch, Patrick Koch, and Wolfgang Konen.
SPOT: A toolbox for interactive and automatic tuning in the r environment.
In Frank Hoffmann and Eyke Hüllermeier, editors, *Proceedings 20. Workshop Computational Intelligence*, pages 264–273. Universitätsverlag Karlsruhe, 2010.

[9] Nacim Belkhir, Johann Dréo, Pierre Savéant, and Marc Schoenauer.
Per instance algorithm configuration of CMA-ES with limited budget.
In *Proc. of Genetic and Evolutionary Computation (GECCO'17)*, pages 681–688. ACM, 2017.

[10] Christopher R. Bessette and David B. Spencer.
Optimal space trajectory design: A heuristic-based approach.
In *Spaceflight Mechanics 2006 - Proceedings of the AAS/AIAA Space Flight Mechnaics Meeting*, Advances in the Astronautical Sciences, pages 1611–1628, 2006.
Spaceflight Mechanics 2006 - AAS/AIAA Space Flight Mechnaics Meeting ; Conference date: 22-01-2006 Through 26-01-2006.

[11] Nicola Beume, Boris Naujoks, and Michael Emmerich.
SMS-EMOA: Multiobjective selection based on dominated hypervolume.
*European Journal of Operational Research*, 181(3):1653 – 1669, 2007.




## References II


[12] Süntje Böttcher, Benjamin Doerr, and Frank Neumann.
Optimal fixed and adaptive mutation rates for the LeadingOnes problem.
In *Proc. of Parallel Problem Solving from Nature (PPSN'10)*, volume 6238 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2010.

[13] Jürgen Branke.
Evolutionary approaches to dynamic optimization problems - updated survey.
In *GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, pages 27–30, 2001.

[14] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer.
Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems.
*IEEE Transactions on Evolutionary Computation*, 10(6):646–657, December 2006.

[15] J. Brest, V. Zumer, and M. S. Maucec.
Self-adaptive differential evolution algorithm in constrained real-parameter optimization.
In *2006 IEEE International Conference on Evolutionary Computation*, pages 215–222, 2006.

[16] Janez Brest and Mirjam Sepesy Maučec.
Population size reduction for the differential evolution algorithm.
*Appl. Intell.*, 29(3):228–247, 2008.

[17] Janez Brest, Aleš Zamuda, Borko Bošković, Mirjam Sepesy Maučec, and Viljem Žumer.
Dynamic optimization using self-adaptive differential evolution.
In *IEEE Congress on Evolutionary Computation, 2009. CEC'09*, pages 415–422. IEEE, 2009.

[18] Lam T Bui, Hussein A Abbass, and Daryl Essam.
Fitness inheritance for noisy evolutionary multi-objective optimization.
In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 779–785. ACM, 2005.

[19] Edmund K. Burke, Michel Gendreau, Matthew R. Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, and Rong Qu.
Hyper-heuristics: a survey of the state of the art.
*Journal of the Operational Research Society*, 64:1695–1724, 2013.

[20] Nathan Buskulic and Carola Doerr.
Maximizing drift is not optimal for solving onemax.
In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'19, Companion)*. ACM, 2019.
Full version available online at http://arxiv.org/abs/1904.07818. See also https://github.com/NathanBuskulic/OneMaxOptimal for more project data.

[21] Daniel Molina Cabrera.
Evolutionary algorithms for large-scale global optimisation: a snapshot, trends and challenges.
*Progress in Artificial Intelligence*, 5(2):85–89, May 2016.

[22] Ruifen Cao, Guoli Li, and Yican Wu.
A self-adaptive evolutionary algorithm for multi-objective optimization.
In De-Shuang Huang, Laurent Heutte, and Marco Loog, editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 553–564, Berlin, Heidelberg, 2007. Springer.




## References III


[23] Jorge Cervantes and Christopher R. Stephens.
Limitations of existing mutation rate heuristics and how a rank GA overcomes them.
*IEEE Transactions on Evolutionary Computation*, 13:369–397, 2009.

[24] Duc-Cuong Dang and Per Kristian Lehre.
Self-adaptation of mutation rates in non-elitist populations.
In *Proc. of Parallel Problem Solving from Nature (PPSN'16)*, volume 9921 of *LNCS*, pages 803–813. Springer, 2016.

[25] Viet Hung Dang, Ngo Anh Vien, and Tae Choong Chung.
A covariance matrix adaptation evolution strategy in reproducing kernel hilbert space.
*Genetic Programming and Evolvable Machines*, pages 1–23, 6 2019.

[26] Swagatam Das, Sankha Subhra Mullick, and P.N. Suganthan.
Recent advances in differential evolution – an updated survey.
*Swarm and Evolutionary Computation*, 27:1 – 30, 2016.

[27] A. Datta, S. Ghosh, and A. Ghosh.
Wrapper based feature selection in hyperspectral image data using self-adaptive differential evolution.
In *2011 International Conference on Image Information Processing*, pages 1–6, 2011.

[28] K. Deb, A. Anand, and D. Joshi.
A computationally efficient evolutionary algorithm for real-parameter optimization.
*Evolutionary Computation*, 10(4):371–395, 2002.

[29] Luc Devroye.
*The compound random search.*
Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 1972.

[30] Benjamin Doerr and Carola Doerr.
Theory of parameter control for discrete black-box optimization: Provable performance gains through dynamic parameter choices.
In *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, pages 271–321. Springer, 2020.

[31] Benjamin Doerr, Carola Doerr, and Johannes Lengler.
Self-adjusting mutation rates with provably optimal success rules.
In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'19)*, pages 1479–1487. ACM, 2019.
Full version available online at http://arxiv.org/abs/1902.02588.

[32] Benjamin Doerr, Carola Doerr, and Jing Yang.
Optimal parameter choices via precise black-box analysis.
In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'16)*, pages 1123–1130. ACM, 2016.

[33] Benjamin Doerr, Andrei Lissovoi, Pietro S. Oliveto, and John Alasdair Warwicker.
On the runtime analysis of selection hyper-heuristics with adaptive learning periods.
In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'18)*, pages 1015–1022. ACM, 2018.

## References IV


[34] Benjamin Doerr, Carsten Witt, and Jing Yang.
Runtime analysis for self-adaptive mutation rates.
In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, pages 1475–1482, New York, NY, USA, 2018. ACM.

[35] Carola Doerr.
Dynamic parameter choices in evolutionary computation.
In Manuel López-Ibáñez, Anne Auger, and Thomas Stützle, editors, *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, pages 890–922. ACM, 2019.

[36] Carola Doerr and Markus Wagner.
On the effectiveness of simple success-based parameter selection mechanisms for two classical discrete black-box optimization benchmark problems.
In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'18)*, pages 943–950. ACM, 2018.

[37] Carola Doerr and Markus Wagner.
Simple on-the-fly parameter selection mechanisms for two classical discrete black-box optimization benchmark problems.
In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018, Kyoto, Japan, July 15-19, 2018*, pages 943–950, 2018.

[38] Elena-Niculina Dragoi, Silvia Curteanu, Anca-Irina Galaction, and Dan Cascaval.
Optimization methodology based on neural networks and self-adaptive differential evolution algorithm applied to an aerobic fermentation process.
*Applied Soft Computing*, 13(1):222 – 238, 2013.

[39] Agoston Endre Eiben, Robert Hinterding, and Zbigniew Michalewicz.
Parameter control in evolutionary algorithms.
*IEEE Transactions on Evolutionary Computation*, 3:124–141, 1999.

[40] Stefan Falkner, Aaron Klein, and Frank Hutter.
BOHB: Robust and efficient hyperparameter optimization at scale.
In *ICML*, pages 1436–1445, 2018.

[41] Álvaro Fialho, Luís Da Costa, Marc Schoenauer, and Michèle Sebag.
Extreme value based adaptive operator selection.
In *Proc. of Parallel Problem Solving from Nature (PPSN'08)*, volume 5199 of *Lecture Notes in Computer Science*, pages 175–184. Springer, 2008.

[42] J Michael Fitzpatrick and John J Grefenstette.
Genetic algorithms in noisy environments.
*Machine learning*, 3(2-3):101–120, 1988.

[43] A. Glotić, A. Glotić, P. Kitak, J. Pihler, and I. Tičar.
Parallel self-adaptive differential evolution algorithm for solving short-term hydro scheduling problem.
*IEEE Transactions on Power Systems*, 29(5):2347–2358, 2014.

[44] Arnel Glotić and Aleš Zamuda.
Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution.
*Applied Energy*, 141:42 – 56, 2015.




## References V


[45] G. Golkarnarenji, M. Naebe, K. Badii, A. S. Milani, A. Jamali, A. Bab-Hadiashar, R. N. Jazar, and H. Khayyam.
Multi-objective optimization of manufacturing process in carbon fiber industry using artificial intelligence techniques.
*IEEE Access*, 7:67576–67588, 2019.

[46] Abhijith M. Gopakumar, Prasanna V. Balachandran, Dezhen Xue, James E. Gubernatis, and Turab Lookman.
Multi-objective optimization for materials discovery via adaptive design.
*Scientific Reports*, 8:3738, 2018.

[47] Horst Greiner.
Robust optical coating design with evolutionary strategies.
*Applied Optics*, 35(28):5477–5483, Oct 1996.

[48] Nizar Hachicha, Bassem Jarboui, and Patrick Siarry.
A fuzzy logic control using a differential evolution algorithm aimed at modelling the financial market dynamics.
*Information Sciences*, 181(1):79–91, January 2011.

[49] G. E. Hinton and R. R. Salakhutdinov.
Reducing the dimensionality of data with neural networks.
*Science*, 313(5786):504–507, 2006.

[50] Changwu Huang, Yuanxiang Li, and Xin Yao.
A survey of automatic parameter tuning methods for metaheuristics.
*IEEE Transactions on Evolutionary Computation*, 24(2):201–216, 2020.

[51] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown.
Sequential model-based optimization for general algorithm configuration.
In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, LION'05, pages 507–523, Berlin, Heidelberg, 2011. Springer-Verlag.

[52] Frank Hutter, Holger H. Hoos, Kevin Leyton-Brown, and Thomas Stützle.
Paramils: An automatic algorithm configuration framework.
*Journal of Artificial Intelligence Research*, 36(1):267–306, September 2009.

[53] Christian Igel, Nikolaus Hansen, and Stefan Roth.
Covariance matrix adaptation for multi-objective optimization.
*Evolutionary computation*, 15(1):1–28, 2007.

[54] Yaochu Jin and Jürgen Branke.
Evolutionary optimization in uncertain environments-a survey.
*IEEE Transactions on evolutionary computation*, 9(3):303–317, 2005.

[55] Giorgos Karafotias, Mark Hoogendoorn, and A.E. Eiben.
Parameter control in evolutionary algorithms: Trends and challenges.
*IEEE Transactions on Evolutionary Computation*, 19:167–187, 2015.




## References VI


[56] Peter Korošec, Uroš Bole, and Gregor Papa.
A multi-objective approach to the application of real-world production scheduling.
*Expert Systems with Applications*, 40(15):5839–5853, 2013.

[57] Peter Korošec and Gregor Papa.
Metaheuristic approach to transportation scheduling in emergency situations.
*Transport*, 28(1):46–59, 2013.

[58] Peter Korošec, Gregor Papa, and Vida Vukašinović.
Application of memetic algorithm in production planning.
In *Proc. Bioinspired Optimization Methods and their Applications, BIOMA 2010*, pages 163–175, May 2010.

[59] Peter Korošec and Jurij Šilc.
The multilevel ant stigmergy algorithm: An industrial case study.
In *Proceedings of the 8th Joint Conference on Information Sciences*, pages 475–478, Salt Lake City, UT, July 21-25 2005.

[60] Peter Korošec, Jurij Šilc, and Bogdan Filipič.
The differential ant-stigmergy algorithm.
*Information Sciences*, 192(1):82–97, 2012.

[61] Laurent Lemarchand, Damien Massé, Pascal Rebreyend, , and Johan Håkansson.
Multiobjective optimization for multimode transportation problems.
*Advances in Operations Research*, page 13, 2018.

[62] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar.
Hyperband: A novel bandit-based approach to hyperparameter optimization.
*J. Mach. Learn. Res.*, 18(1):6765–6816, 2017.

[63] Rui Li, Michael T.M. Emmerich, Jeroen Eggermont, Thomas Bäck, M. Schütz, J. Dijkstra, and J.H.C. Reiber.
Mixed integer evolution strategies for parameter optimization.
*Evolutionary Computation*, 21(1):29–64, 2013.
PMID: 22122384.

[64] X. Li, K Tang, P. N. Suganthan, and Z. Yang.
Editorial for the special issue of information sciences journal (ISJ) on 'nature-inspired algorithms for large scale global optimization.
*Information Science*, 316:437–439, 2015.

[65] T.W. Liao, P.J. Egbelu, and P.C. Chang.
Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations.
*Applied Soft Computing*, 12(11):3683 – 3697, 2012.

[66] Andrei Lissovoi, Pietro S. Oliveto, and John Alasdair Warwicker.
On the runtime analysis of generalised selection hyper-heuristics for pseudo-Boolean optimisation.
In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'17)*, pages 849–856. ACM, 2017.
Extended version available at https://arxiv.org/abs/1801.07546.




## References VII


[67] Andrei Lissovoi, Pietro S. Oliveto, and John Alasdair Warwicker.
Hyper-heuristics can achieve optimal performance for pseudo-boolean optimisation.
*Arxiv e-prints*, 1801.07546, 2018.

[68] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Thomas Stützle, and Mauro Birattari.
The irace package: Iterated racing for automatic algorithm configuration.
*Operations Research Perspectives*, 3:43–58, 2016.

[69] Efrén Mezura-Montes, Edgar-Alfredo Portilla-Flores, and Edith Capistran-Gumersindo.
Dynamic parameter control in differential evolution with combined variants to optimize a three-finger end effector.
In *2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pages 1–6, 2015.

[70] A. Migdalas, Panos M. Pardalos, and Peter Vrbrand.
*Multilevel Optimization: Algorithms and Applications.*
Springer Publishing Company, Incorporated, 1st edition, 2012.

[71] Brad L. Miller.
*Noise, sampling, and efficient genetic algorithms.*
PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[72] Ali Wagdy Mohamed and Abdulaziz S. Almazyad.
Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems.
*Applied Computational Intelligence and Soft Computing*, 2017:1–18, 2017.

[73] Boris Naujoks and Heike Trautmann.
Online convergence detection for multiobjective aerodynamic applications.
In *2009 IEEE Congress on Evolutionary Computation*, pages 332–339, May 2009.

[74] Trung Thanh Nguyen, Shengxiang Yang, and Juergen Branke.
Evolutionary dynamic optimization: A survey of the state of the art.
*Swarm and Evolutionary Computation*, 6:1–24, 2012.

[75] Mohammad Nabi Omidvar, Xiaodong Li, Daniel Molina, and Antonio LaTorre.
Evolutionary large-scale global optimization, 2019.
CEC 2019 Tutorial.

[76] Mohammad Nabi Omidvar, Ming Yang, Yi Mei, Xiaodong Li, and Xin Yao.
DG2: A faster and more accurate differential grouping for large-scale black-box optimization.
*IEEE Transactions on Evolutionary Computation*, 21(6):929–942, 2017.

[77] Pan Zhang, Xin Yao, Lei Jia, B. Sendhoff, and T. Schnier.
Target shape design optimization by evolving splines.
In *2007 IEEE Congress on Evolutionary Computation*, pages 2009–2016, 2007.

## References VIII


[78] Gregor Papa.
Parameter-less evolutionary search.
In *Proc. Genetic and Evolutionary Computation Conference (GECCO 2008)*, pages 1133–1134, 2008.

[79] Gregor Papa.
Parameter-less algorithm for evolutionary-based optimization.
*Computational Optimization and Applications*, 56(1):209–229, 2013.

[80] Gregor Papa and Carola Doerr.
Dynamic control parameter choices in evolutionary computation: GECCO 2020 tutorial.
In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, GECCO '20, page 927–956, New York, NY, USA, 2020. Association for Computing Machinery.

[81] Gregor Papa, Vida Vukašinović, and Peter Korošec.
Guided restarting local search for production planning.
*Engineering Applications of Artificial Intelligence*, 25(2):242–253, 2012.

[82] Ingo Rechenberg.
*Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*.
Frommann-Holzboog, 1973.

[83] Nery Riquelme, Christian Von Lücken, and Benjamin Baran.
Performance metrics in multi-objective optimization.
In *Computing Conference (CLEI), 2015 Latin American*, pages 1–11. IEEE, 2015.

[84] Alejandro Rodríguez-Molina, Miguel G. Villarreal-Cervantes, Efrén Mezura-Montes, and Mario Aldape-Pérez.
Adaptive controller tuning method based on online multiobjective optimization: A case study of the four-bar mechanism.
*IEEE Transactions on Cybernetics*, 51(3):1272–1285, 2021.

[85] Günter Rudolph.
*Convergence properties of evolutionary algorithms*.
Kovac, 1997.

[86] Namhee Ryu and Seungjae Min.
Multi-objective optimization with an adaptive weight determination scheme using the concept of hyperplane: Multi-objective optimization with an adaptive weight.
*International Journal for Numerical Methods in Engineering*, 118, 12 2018.

[87] F. Samadzadegan and T. Partovi.
Feature selection based on ant colony algorithm for hyperspectral remote sensing images.
In *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, 2010.

[88] David Schmaranzer, Roland Braune, and Karl Franz Dörner.
Multi-objective simulation optimization for complex urban mass rapid transit systems.
*Annals of Operations Research*, 2019.




## References IX


[89] Michael A. Schumer and Kenneth Steiglitz.
Adaptive step size random search.
*IEEE Transactions on Automatic Control*, 13:270–276, 1968.

[90] M. C. Sinclair.
Operator-probability adaptation in a genetic-algorithm/heuristic hybrid for optical network wavelength allocation.
In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 840–845, 1998.

[91] Ankur Sinha, Pekka Malo, Peng Xu, and Kalyanmoy Deb.
A bilevel optimization approach to automated parameter tuning.
In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 847–854, New York, NY, USA, 2014. ACM.

[92] Dirk Sudholt.
A new method for lower bounds on the running time of evolutionary algorithms.
*IEEE Transactions on Evolutionary Computation*, 17:418–435, 2013.

[93] Daniel R. Tauritz and John Woodward.
Hyper-heuristics tutorial.
In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, GECCO '20, page 652–681, New York, NY, USA, 2020. Association for Computing Machinery.

[94] Jürgen Teich.
Pareto-front exploration with uncertain objectives.
In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 314–328. Springer, 2001.

[95] Hong-Fei Teng, Yu Chen, Wei Zeng, Yan-Jun Shi, and Qing-Hua Hu.
A dual-system variable-grain cooperative coevolutionary algorithm: Satellite-module layout design.
*IEEE Transactions on Evolutionary Computation*, 14(3):438–455, June 2010.

[96] Tea Tušar, Peter Korošec, Gregor Papa, Bogdan Filipič, and Jurij Šilc.
A comparative study of stochastic optimization methods in electric motor design.
*Applied Intelligence*, 27(2):101–111, 2007.

[97] Marta Vallejo and David W. Corne.
Evolutionary algorithms under noise and uncertainty: A location-allocation case study.
*2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–10, 2016.

[98] G.N. Vanderplaats.
*Very large scale optimization*.
Nat. Aeronautics Space Admin., Washington, DC, 2002.

[99] M. Vasile, E. Minisci, and M. Locatelli.
An inflationary differential evolution algorithm for space trajectory optimization.
*IEEE Transactions on Evolutionary Computation*, 15(2):267–281, 2011.




## References X


[100] C. Wang and J. Gao.
High-dimensional waveform inversion with cooperative coevolutionary differential evolution algorithm.
*IEEE Geoscience and Remote Sensing Letters*, 9(2):297–301, March 2012.

[101] Hongfeng Wang, Dingwei Wang, and Shengxiang Yang.
A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems.
*Soft Computing*, 13(8-9):763–780, 2009.

[102] Yao-Nan Wang, Liang-Hong Wu, and Xiao-Fang Yuan.
Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure.
*Soft Computing*, 14(3):193–209, 2010.

[103] Yu Wang, Jin Huang, Wei Shan Dong, Jun Chi Yan, Chun Hua Tian, Min Li, and Wen Ting Mo.
Two-stage based ensemble optimization framework for large-scale global optimization.
*European Journal of Operational Research*, 228(2):308 – 320, 2013.

[104] D. H. Wolpert and W. G. Macready.
No free lunch theorems for optimization.
*IEEE Transactions on Evolutionary Computation*, 1:67–82, 1997.

[105] Dongdong Yang, Licheng Jiao, and Maoguo Gong.
Adaptive multi-objective optimization based on nondominated solutions.
*Computational Intelligence*, 25(2):84–108, 2009.

[106] Zhenyu Yang, B. Sendhoff, Ke Tang, and Xin Yao.
Target shape design optimization by evolving b-splines with cooperative coevolution.
*Applied Soft Computing*, 48(C):672–682, November 2016.

[107] Zhenyu Yang, Ke Tang, and Xin Yao.
Large scale evolutionary optimization using cooperative coevolution.
*Information Sciences*, 178(15):2985 – 2999, 2008.
Nature Inspired Problem-Solving.

[108] Yan Ye, Jingfeng Li, Kaibin Li, and Hui Fu.
Cross-docking truck scheduling with product unloading/loading constraints based on an improved particle swarm optimisation algorithm.
*International Journal of Production Research*, 56(16):5365–5385, 2018.

[109] Ali R. Yildiz.
A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations.
*Applied Soft Computing*, 13(3):1561 – 1566, 2013.
Hybrid evolutionary systems for manufacturing processes.

[110] A. Zamuda, J. Brest, B. Bošković, and V. Žumer.
Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution.
In *IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3718–3725, June 2008.




## References XI


[111] A. Zamuda, J. Daniel Hernández Sosa, and L. Adler.
Improving constrained glider trajectories for ocean eddy border sampling within extended mission planning time.
In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1727–1734, 2016.

[112] Ales Zamuda and Janez Brest.
Vectorized procedural models for animated trees reconstruction using differential evolution.
*Information Sciences*, 278:1–21, 2014.

[113] Ales Zamuda, Janez Brest, Borko Boskovic, and Viljem Zumer.
Differential evolution with self-adaptation and local search for constrained multiobjective optimization.
In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2009, Trondheim, Norway, 18-21 May, 2009*, pages 195–202. IEEE, 2009.

[114] Aleš Zamuda and José Daniel Hernández Sosa.
Differential evolution and underwater glider path planning applied to the short-term opportunistic sampling of dynamic mesoscale ocean structures.
*Applied Soft Computing*, 24:95 – 108, 2014.

[115] Aleš Zamuda and José Daniel Hernández Sosa.
Success history applied to expert system for underwater glider path planning using differential evolution.
*Expert Systems with Applications*, 119:155 – 170, 2019.

[116] Z. H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams.
Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum].
*IEEE Computational Intelligence Magazine*, 9(4):62–74, Nov 2014.