

A Fast Multi-objective Evolutionary Approach for Designing Large-Scale Optical Mode Sorter

Panichella, A.; Di Domenico, Giuseppe

DOI

[10.1145/3583131.3590479](https://doi.org/10.1145/3583131.3590479)

Publication date

2023

Document Version

Final published version

Published in

GECCO 2023 - Proceedings of the 2023 Genetic and Evolutionary Computation Conference

Citation (APA)

Panichella, A., & Di Domenico, G. (2023). A Fast Multi-objective Evolutionary Approach for Designing Large-Scale Optical Mode Sorter. In L. Paquete (Ed.), *GECCO 2023 - Proceedings of the 2023 Genetic and Evolutionary Computation Conference* (pp. 1445–1453). (GECCO 2023 - Proceedings of the 2023 Genetic and Evolutionary Computation Conference). ACM/IEEE. <https://doi.org/10.1145/3583131.3590479>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



A Fast Multi-objective Evolutionary Approach for Designing Large-Scale Optical Mode Sorters

Annibale Panichella
Delft University of Technology
Delft, The Netherlands
a.panichella@tudelft.nl

Giuseppe Di Domenico
attocube Systems AG
Munich, Germany
giuseppedidomenico8@gmail.com

ABSTRACT

Spatial mode division de-multiplexing of optical signals has many real-world applications, such as quantum computing and both classical and quantum optical communication. In this context, it is crucial to develop devices able to efficiently sort optical signals according to the optical mode they belong to and route them on different paths. Depending on the mode selected, this problem can be very hard to tackle. Recently, researchers have proposed using multi-objective evolutionary algorithms (MOEAs)—and NSGA-II in particular—combined with Linkage Learning (LL) to automate the process of design mode sorter. However, given the very large-search scale of the problem, the existing evolutionary-based solutions have a very slow convergence rate. In this paper, we proposed a novel approach for mode sorter design that combines (1) stochastic linkage learning, (2) the adaptive geometry estimation-based MOEA (AGE-MOEA-II), and (3) an adaptive mutation operator. Our experiments with two- and three-objectives (beams) show that our approach is faster (better convergence rate) and produces better mode sorters (closer to the ideal solutions) than the state-of-the-art approach. A direct comparison with the vanilla NSGA-II and AGE-MOEA-II also further confirms the importance of adopting LL in this domain.

CCS CONCEPTS

• Theory of computation → Evolutionary algorithms; • Hardware → Emerging optical and photonic technologies.

KEYWORDS

Many-objective optimization, Mode Sorter, Linkage Learning

ACM Reference Format:

Annibale Panichella and Giuseppe Di Domenico. 2023. A Fast Multi-objective Evolutionary Approach for Designing Large-Scale Optical Mode Sorters. In *Genetic and Evolutionary Computation Conference (GECCO '23)*, July 15–19, 2023, Lisbon, Portugal. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3583131.3590479>

1 INTRODUCTION

Our modern world heavily relies on optical communications. Simply put, it involves transmitting information from one place to another using light [1]. This process can be divided into three main phases:



This work is licensed under a Creative Commons Attribution International 4.0 License. *GECCO '23, July 15–19, 2023, Lisbon, Portugal*
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0119-1/23/07.
<https://doi.org/10.1145/3583131.3590479>

modulation, transport, and detection. Modulation involves encoding bits of information (or quantum bits) into the properties of an optical beam. The light must then be transported from one place to another, either through free propagation or guided by an optical fiber. Finally, the beam is detected to recover the information.

In theory, any property of an optical beam could work to encode information. Besides amplitude, changes in phase, polarization, or wavelength of the carrier light can be used, but it requires a device able to discriminate optical beams based on that property. This requirement has restricted the use of spatial degrees of freedom in communication systems, which means encoding information in the shape of an optical beam. Distinguishing beams based on their shapes is a difficult problem [35, 48] especially when it needs to be done on a miniaturized device [42]. Additionally, the beam shape must also remain unchanged during propagation, meaning it needs to be a mode. Encoding information into modes offers several advantages, including inherent resistance to noise [5], the capability to encode quantum information [17, 33], and, in theory, allows for an infinite number of modes to serve as the alphabet [12].

A device that distinguishes beams based on their mode is a device that routes every mode to different outputs. This allows the encoded information to be recovered by placing a light detector at each output. A mode sorter can be realized by designing a refractive index distribution that causes each mode to propagate differently, ensuring that the energy of each mode can be efficiently coupled to a specific detector. Designing such devices is an inverse problem where the optimal refractive index distribution must be determined to perform different tasks on different inputs simultaneously.

To tackle this problem, researchers have used evolutionary algorithms [9] to evolve devices (or mode sorter) that can effectively demultiplex different modes (or light signals). In particular, Di Domenico et al. [9] used L2-NSGA [23], which combines the multi-objective NSGA-II [8] with *linkage-learning* (LL) [41]. The search objectives to optimize are the distances between (1) the theoretical locations where the different modes should be redirected (target outputs) and (2) the actual locations where the device redirects the modes. L2-NSGA uses Pareto ranking to select and evolve better devices. Instead, linkage learning is applied to identifying and preserving linkage structures, i.e., features of the devices that must be replicated altogether into the offspring.

While prior studies show that L2-NSGA is effective at designing Pareto-efficient two-mode and three-mode devices, it requires many fitness evaluations to converge. Di Domenico et al. [9] reported that L2-NSGA requires a large population of 1000 devices and thousands of generations to converge. This is due to the very large-scale nature of the problem since a device for demultiplexing two beams has more than 12K decision variables.

In this paper, we aim to tackle this slow converge rate by proposing a novel evolutionary approach, namely L2-AGE-MOEA, that extends the *Improved* (AGE-MOEA-II) [26]. Our approach combines AGE-MOEA-II with linkage learning methods to preserve the link structures of the best devices generated through the generations. L2-AGE-MOEA maintains the environmental selection methods of AGE-MOEA-II but uses linkage-based recombination to generate new candidate devices in each iteration. Furthermore, our approach uses an adaptive mutation operator that linearly decreases the mutation rate through the generations. It increases exploration at the beginning of the search (higher mutation rate) while promoting exploitation in the later generations (lower mutation rate).

We conducted an empirical study with four different types of devices with two and three Hermite-Gauss beams to demultiplex. These devices differ in the number of beams to demultiplex and the device's size. Our experimental results show that (1) L2-AGE-MOEA outperforms the state-of-the-art L2-NSGA used in prior studies, and (2) L2-AGE-MOEA outperforms the AGE-MOEA-II (i.e., without linkage learning and the adaptive mutation operator). Furthermore, a direct comparison between the NSGA-II and AGE-MOEA-II with the linkage learning-based variants (i.e., L2-AGE-MOEA and L2-NSGA) further confirms how linkage learning is a fundamental ingredient for designing Pareto efficient mode sorters.

The remainder of the paper is organized as follows. Section 2 summarizes the preliminaries on mode sorters, multi-objective optimization, and AGE-MOEA-II. Section 3 introduces our approach, highlighting its novel aspects over AGE-MOEA-II. Section 4.1 describes the empirical study and discusses the achieved results. Section 5 concludes the paper and discusses potential future directions.

2 BACKGROUND

Multi-objective optimization has a wide range of applications engineering problems [21]. A multi-objective problem requires to optimize multiple objectives simultaneously:

$$\begin{aligned} \min F(x) &= [f_1(x), f_2(x), \dots, f_M(x)]^T \\ \text{subject to } x &\in \Omega \subseteq \mathbb{R}^D \end{aligned} \quad (1)$$

where $x = (x_1, \dots, x_D)^T$ is the decision vector with D decision variables from the feasible region $\Omega \subset \mathbb{R}^D$; $f : \Omega \rightarrow \mathbb{R}^M$ is an objective vector with M conflicting objectives to optimize.

A solution x *dominates* another solution y ($x < y$), if $f_i(x) \leq f_i(y) \forall f_i \in F$ and there exists at least one objective $f_j \in F$ such that $f_j(x) < f_j(y)$. A *Pareto optimal* solution $x^* \in \Omega$ is a solution that is non-dominated by any other solution in Ω , i.e., $\nexists y \in \Omega$ such that $y < x^*$. The set of all Pareto optimal solutions is called *Pareto optimal solution set* (PS), while the corresponding objective vectors form the so-called *Pareto front* (PF).

2.1 Mode Sorter

Mode sorters (or demultiplexers) are devices that distinguish beams based on their mode and route them to different outputs. The most common mode of the electromagnetic wave equation is the Gaussian beam due to both theoretical and practical considerations. Theoretically, Gaussian beams have a simple mathematical formulation characterized by a radially symmetrical intensity profile that follows a Gaussian function with respect to the distance from the

beam axis [3]. In practice, most lasers produce Gaussian beams due to their design and construction of laser resonators [13, 36].

We focus on the family of Hermite-Gauss (HG) beams [12], which intensity distribution is represented by Hermite polynomials of various orders, with Gaussian beams being the first order¹. HG beams are ideal in communication systems due to their ability to resist noise [5] and their orthogonality which allows them to be perfectly distinguished [32] limiting the crosstalk between channels. Besides, their unlimited number does not restrict the size of the alphabet that can be used [12].

Inverse design of optical devices. Researchers have proposed various techniques to automatically design and produce optical devices with various functionalities, such as wavelength demultiplexers [28, 34], polarization splitters [30], and mode converters [16]. These methods differ in the underline optimization techniques (e.g., line search [34] or steepest descent method [28]), which typically start with a randomly initialized device with continuous refractive index distribution that is iteratively updated until converging to a binary device structure [20]. Furthermore, these approaches can successfully design small-scale devices, with size in the order of a few μm^2 and target a pre-defined fixed number of beams and channels (e.g., three channels in [34]).

Inverse design of large-scale mode sorters. In this work, we focus on a specific type of optical devices, namely large-scale mode sorters. In other words, we want to design large-scale (i.e., $> 100nm^2$ in size) devices that can distinguish and route the input beams based on their mode (mode sorter) and with different numbers of beams. To the best of our knowledge, the only prior work that focused on large-scale mode sorters is the work by Di Domenico et al. [9]. The authors have formulated the problem of designing mode sorters for HG beams on a surface plasmon polarization platform as a multi-objective optimization problem, which can be tackled using evolutionary algorithms. In particular, a device X is encoded as an $n \times k$ binary matrix, where each entry $x_{i,j}$ corresponds to a $400nm \times 400nm$ region of the device. $x_{i,j}=1$ indicates that the region in row i and column j contains a polymethyl methacrylate (PMMA) layer, which increases the refractive index locally; while $x_{i,j}=0$ indicate the absence of such layers. The refractive index distribution that makes up the devices (i.e., a grid of PMMA layers) affects the propagation of various beams differently and will determine where each beam will be routed.

Given a device X , the goal is to route M HG beams (modes) towards M different output locations. Therefore, the objectives to optimize are the distances between the target locations and the actual locations where the HG beams are routed by the device X . In other words, let HG_0, \dots, HG_M input beams, the objectives to optimize are:

$$\min \begin{cases} f_1 = \|t(HG_0) - g(X, HG_0)\|_2 \\ \dots \\ f_M = \|t(HG_{M-1}) - g(X, HG_{M-1})\|_2 \end{cases} \quad (2)$$

where $t(HG_i)$ indicates the theoretical (target) output signal for the beam HG_i while $g(X, HG_{M-1})$ denotes the actual output produced by the device X when given as input the beam HG_i . Note that

¹In this context, Gaussian beams are called "HG00", which stands for Hermite-Gauss beam of zero order, while a general "HGmn" is a Hermite-Gauss beam of order m in the x direction and n in the y direction, given z as the axis of propagation. One index is dropped, as in "HGn", for the case of two-dimensional beams.

Algorithm 1: Pseudo-code of AGE-MOEA-II [26]

```

Input:  $M$ : Number of objectives and  $N$ : Population size
Result: Final population  $P$ 
1 begin
2    $P \leftarrow \text{RANDOM-POPULATION}(N)$ 
3   while not (stop_condition) do
4      $Q \leftarrow \text{GENERATE-OFFSPRING}(P)$ 
5      $\mathbb{F} \leftarrow \text{FAST-NONDOMINATED-SORT}(P \cup Q)$ 
6      $\mathbb{F} \leftarrow \text{NORMALIZE}(\mathbb{F})$ 
7      $p \leftarrow \text{NEWTON-RAPHSON}(\mathbb{F}_1)$  /* Eq. 3 */
8      $d \leftarrow 1$  /* First non-dominated rank */
9     while  $|P| + |\mathbb{F}_d| \leq N$  do
10       $\mathbb{F}_\perp \leftarrow \text{MANIFOLD-PROJECTION}(\mathbb{F}_d, p)$ 
11       $D \leftarrow \text{GEODESIC-DIV}(\mathbb{F}_\perp, p)$ 
12       $\text{SURVIVAL-SCORE}(D, \mathbb{F}, d, p)$ 
13       $P \leftarrow P \cup \mathbb{F}_d$ 
14       $d \leftarrow d + 1$ 
15       $\text{SORT}(\mathbb{F}_d)$  /* by survival scores */
16       $P \leftarrow P \cup \mathbb{F}_d[1 : (N - |P|)]$ 
17 return  $P$ 

```

the output is simulated using the beam propagation method [9] (BPM), considering the material's characteristics (e.g., refractive index) that will be used to implement the device. The problem is inherently multi-objective since the goal is to design a device that routes multiple beams. Devices that find the optimal structure to route one individual beam may introduce obstacles for the other beams to route through the same device.

2.2 The Improved Adaptive Geometry Estimation based MOEA (AGE-MOEA-II)

AGE-MOEA-II is a many-objective evolutionary algorithm (MOEA) that adapts the diversity and proximity metrics based on the shape (or geometry) of the non-dominated front. Algorithm 1 [26] reports the pseudo-code AGE-MOEA-II. There are three main steps in AGE-MOEA-II [26]: (1) non-dominated sorting, (2) fronts normalization, (3) computing the curvature p of the first-non dominated front, and (4) applying the environmental selection based on *proximity* and *diversity*, both computed using the estimated value for p . In the following, we briefly describe these main steps.

2.2.1 Non-dominated sorting and front normalization. First, the parent and offspring populations are combined, and their solutions are ranked in subsequent fronts using the *fast non-dominated sorting algorithm* from NSGA-II [8] (line 5 in Algorithm 1). Then, all fronts are normalized such that the objectives are scaled into the interval $[0; 1]^M$, where M is the number of objectives (line 6 in Algorithm 1). The normalization is done by using the hyperplane method of NSGA-III [2]. In particular, the front \mathbb{F}_i is scaled to the origin of the axes using the ideal point \mathbb{F}_i^{\min} and divided by the hyperplane intercepting the extreme points of the front [26].

2.2.2 Front modeling. After normalization, AGE-MOEA-II uses the Newton-Raphson method to determine its geometry, or curvature p (line 7 in Algorithm 1) of the first non-dominated front. The Newton-Raphson method is an iterative procedure used in AGE-MOEA-II [26], which provides a better (more accurate) estimation of the curvature [26] compared to the heuristic used in AGE-MOEA [25]. More specifically, the iterative Newton-Raphson

formula for computing p is the following:

$$p_{n+1} = p_n + \frac{\log \left[\sum_{i=1}^M (a_i)^{p_n} \right]}{\left[\sum_{i=1}^M (a_i)^{p_n} \cdot \log(a_i) \right]} \quad (3)$$

with p_n and p_{n+1} are the approximated roots obtained at the iterations n and $n + 1$, respectively. To initialize the procedure, the initial value p_0 is set 1, and the formula is applied for four iterations. As reported by Panichella [26], four iterations are enough to achieve an accurate approximation of p with an error lower than 0.0001.

2.2.3 Environmental selection. Given the curvature p computed using Equation 3, the non-dominated solutions are projected onto the estimated front manifold with curvature p . Finally, each solution is assigned a *survival score* that combines the *convergence* and *diversity* scores. The convergence score measures how distance each solution is from the utopia/ideal points. Instead, the diversity score measures the distance between the non-dominated points on the manifold with curvature p . The diversity of each non-dominated solution A corresponds to the *geodesic distance*, which generalizes the Euclidean distance for manifolds with curvature $p \neq 1$. In particular, the geodesic distance between two points A and B is as follows:

$$gd(A, B) = \|A - C_\perp\|_2 + \|B - C_\perp\|_2 \quad (4)$$

where C_\perp is the central point of the arch (curved line) connecting A and B ; $\|A - C_\perp\|_2$ is the Euclidean distance between A and C_\perp ; finally, and $\|B - C_\perp\|_2$ is the Euclidean distance between B and middle point C_\perp .

3 APPROACH

This paper introduces a variant of AGE-MOEA-II, hereafter referred to as L2-AGE-MOEA, which combines the adaptive geometry-based Pareto front modeling with linkage learning.

Motivation. We have decided to use AGE-MOEA-II as starting point for our approach since it was designed for large-scale problems [26] and outperformed other state-of-the-art MOEAs, including LMEA [46], NSGA-III [7], and MOEA/D [24]. Besides, in our preliminary experiments, we have also observed that the Pareto fronts for our problem (designing mode sorters) have a hyperbolic geometry (curvature). Prior studies [22, 25, 26, 31] showed that AGE-MOEA-II performs better than other state-of-the-art algorithms for problems with hyperbolic shapes.

Furthermore, not all elements in a device (i.e., entries in the device matrix as explained in Section 2) do contribute to routing the beams towards the right/correct locations. The core idea for using linkage learning methods is that (through the generations) these methods are able to infer which entries in the matrix structure contribute to the objectives (i.e., routing the beams successfully) and, therefore, should be inherited by the children for the next generation.

Device encoding and evaluation. We use the same encoding scheme used by Di Domenico et al. [9] and discussed in Section 2. Therefore, a device X is encoded as an $n \times k$ binary matrix, where each entry $x_{i,j}$ corresponds to a $400nm \times 400nm$ region of the device. Each entry $x_{i,j}$ takes values in 0, 1 and denotes the absence or presence of a PMMA layer in that location of the device.

Algorithm 2: L2-AGE-MOEA

```

Input:  $M$ : Number of objectives and  $N$ : Population size
Result: Final population  $P$ 
1 begin
2    $P \leftarrow \text{RANDOM-POPULATION}(N)$ 
3    $\mathbb{F} \leftarrow \text{FAST-NONDOMINATED-SORT}(P \cup Q)$ 
4    $\text{FOS} \leftarrow \text{INFER-MODEL}(\mathbb{F}_1, \Lambda)$ 
5   while not (stop_condition) do
6      $Q \leftarrow \emptyset$ 
7     forall  $i$  in  $1..|P|$  do
8        $\text{Parent} \leftarrow P_i$ 
9        $\text{Donor} \leftarrow \text{TOURNAMENT-SELECTION}(P)$ 
10       $O \leftarrow \text{REPRODUCTION}(\text{Parent}, \text{Donor}, \text{FOS})$ 
11       $O \leftarrow \text{ADAPTIVE-MUTATION}(O)$ 
12       $Q \leftarrow Q \cup \{O\}$ 
13     $\mathbb{F} \leftarrow \text{FAST-NONDOMINATED-SORT}(P \cup Q)$ 
14     $\mathbb{F} \leftarrow \text{NORMALIZE}(\mathbb{F})$ 
15     $p \leftarrow \text{NEWTON-RAPHSON}(\mathbb{F}_1)$  /* Eq. 3 */
16     $d \leftarrow 1$  /* First non-dominated rank */
17    while  $|P| + |\mathbb{F}_d| \leq N$  do
18       $\mathbb{F}_\perp \leftarrow \text{MANIFOLD-PROJECTION}(\mathbb{F}_d, p)$ 
19       $D \leftarrow \text{GEODESIC-DIV}(\mathbb{F}_\perp, p)$ 
20       $\text{SURVIVAL-SCORE}(D, \mathbb{F}, d, p)$ 
21       $P \leftarrow P \cup \mathbb{F}_d$ 
22       $d \leftarrow d + 1$ 
23     $\text{SORT}(\mathbb{F}_d)$  /* by survival scores */
24     $P \leftarrow P \cup \mathbb{F}_d[1 : (N - |P|)]$ 
25     $\text{FOS} \leftarrow \text{INFER-MODEL}(\mathbb{F}_1, \Lambda)$ 
26  return  $P$ 

```

The solutions/devices are evaluated using the beam propagation method [9] and considering the following device and beam characteristics. We consider a device with a plasmonic refractive index for air silver interface $n_0=1.008856$. As for the beams, we consider a plasmonic field with wavelength $\lambda = 1.064 \cdot 10^{-6}/n_0$ and vacuum wavevector $2\pi/\lambda$. The difference of index of refraction is $1.049321 - n_0$ for the case of 40nm of PMMA. Finally, for the target fields, we consider the width $w_{0C} = 1.6 \cdot 10^{-6}$ and field displacement equal to $12 \cdot 10^6$.

Pseudo-code. The Pseud-code of L2-AGE-MOEA is outlined in Algorithm 2. The Algorithm starts in line 2 with a randomly generated population of N individuals. In our case, an individual is a randomly-generated binary matrix whose size corresponds to the dimensions (number of rows and columns) of the device to design. Then, the solutions in the initial population are ranked using the fast non-dominated sorting Algorithm (line 3 of Algorithm 2). The first front \mathbb{F}_1 is used to derive the linkage structures and extract the *family of subsets* (FOS). We detail this procedure in Section 3.1.

The linkage model is then used in lines 8-12 to generate the new offspring population Q . For each solution $P_i \in P$, a new offspring solution O is created by recombining P_i with a *donor* solution selected from P using the *binary tournament* selection. The recombination procedure is detailed in Section 3.2. It uses the linkage model to determine which genes (linkage structures) of the *donor* solutions will be replicated altogether in the offspring solution.

The *binary tournament selection* operator promotes solutions with better (lower) dominance ranking or better (larger) *survival score* (see Section 2.2.3) at the same level of dominance rank. The offspring O is created using the REPRODUCTION routine in line 10 Algorithm 2, which uses the FOS to decide which gene will be copied from the parent or the donor solution. This procedure is detailed in Section 3.2. Furthermore, the new solution O is further mutated using the *adaptive mutation* operator detailed in Section 3.3.

The remainder of Algorithm 2 is identical to the main loop of the original AGE-MOEA-II. In a nutshell, parent P and offspring Q populations are combined and sorted using the fast non-dominated sorting algorithm (line 13) and normalized (line 14). The first front \mathbb{F}_1 is then used to compute the curvature p using the Newton-Raphson method (Equation 3). Finally, in lines 17-22, the survival score (line 20) is computed iteratively for all fronts using the geodesic distance (line 19).

The loop between lines 17 and 24 inserts as many individuals to the new population P as possible, based on their ranks and until reaching the population size N . L2-AGE-MOEA first selects the non-dominated solutions from \mathbb{F}_1 ; if $|\mathbb{F}_1| < N$, the loop selects the non-dominated solutions from \mathbb{F}_i , etc. The loop ends when inserting all the solutions from the front \mathbb{F}_d exceeds the maximum population size, as shown in line 24. In this case, the algorithm selects the remaining solutions from the front \mathbb{F}_d according to the descending order of *survival score* in lines 24-25.

Finally, the linkage model is retrained in line 25 based on the first non-dominated front \mathbb{F}_1 previously computed on line 13, i.e., on the latest population.

3.1 Linkage learning with Clustering Methods

Linkage learning [41] is a category of techniques applied to infer linkage structures from a population, i.e., groups or clusters of promising decision variable values that contribute to achieving better (more optimal) results. The linkage structures are critical to identifying and preserving good partial solutions.

Various techniques have been proposed in the literature to derive linkage structures, such as Bayesian Network [27], Dependency Structure Matrix [44], and hierarchical clustering [23, 37]. In this work, we focus on hierarchical clustering since they have been used in the past for binary problems [23, 37] and by Di Domenico et al. [9] for the same problem we are addressing in this paper, i.e., mode sorter design.

We selected UPGMA [14] (unweighted pair group method with arithmetic mean) as the hierarchical clustering method and used the *hamming distance* to measure the dissimilarities between genes (or decision variables). UPGMA is a bottom-up approach that iteratively clusters decision variables into larger clusters. The output of the clustering is a tree structure (or *dendrogram*), as in the example of Figure 1. In the dendrogram, the leaf nodes are the decision variables, which are, in turn, clustered together based on their distance. In the example of Figure 1, the variables x_3 and x_4 are first clustered together, and the resulting cluster is then augmented with another variable closest to this cluster. This process is repeated until all decision variables are clustered in the root of the dendrogram. The internal nodes of the dendrogram are the *linkage structures*, also referred to as *Family Of Subsets* (FOS) in the literature [9, 23, 37].

The FOS has D leaf nodes and $D-1$ internal nodes, where D is the number of decision variables. The FOS is a set $\{\Omega^0, \Omega^1, \dots, \Omega^{D-1}\}$ in which an entry Ω_i is the subset of decision variables clustered together at the i -th level (or node) in the dendrogram.

The distance between two decision variables x_i and x_j measures how frequently they have the same values in the best individuals of a population P . For example, if x_i and x_j always have the same values (either zero or one) in all individuals, their distance will

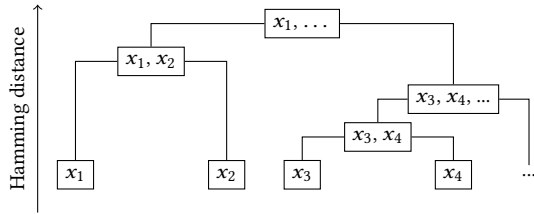


Figure 1: Example of dendrogram or Family Of Subset (FOS)

be zero and clustered together in the lower internal nodes of the dendrogram. Instead, decision variables that never have the same values across the different individuals are clustered together but in the top internal nodes of the dendrogram. The distance between pairs of decision variables is computed using the *hamming distance*, a well-known distance function for binary vectors. This is also the distance recommended in the literature due to its low (linear) computational complexity [9, 23].

3.2 Linkage-based Reproduction

The FOS derived used the method described in Section 3.1 is then used during reproduction. In line 10 of Algorithm 2, a new solution O is obtained by recombining a solution A (parent) with a *donor* solution selected via binary tournament selection. Recombination is done by copying all decision variable values from the parent to the offspring solution O . Then, k randomly selected decision variable values are copied by the donor (mask) into the offspring solution based on the clusters in the FOS.

To this aim, the dendrogram is cut at the distance level so that the number of clusters is equal to $D/4$, where D is the number of decision nodes. Then, 50% of these clusters are randomly selected and used as recombination masks. Let \mathbb{C} be the set of clusters obtained from the dendrogram with cut point $D/4$; let $\mathbb{C}^* \subset \mathbb{C}$ be the 50% or randomly selected clusters from \mathbb{C} ; the new offspring solution O is created as follows:

$$O_i = \begin{cases} \text{Donor}_i & \exists \text{ cluster} \in \mathbb{C}^* \text{ such that } i \in \text{cluster} \\ A_i & \nexists \text{ cluster} \in \mathbb{C}^* \text{ such that } i \in \text{cluster} \end{cases} \quad (5)$$

where O_i , A_i , and Donor_i denote the i -th decision variable of O , A , and Donor respectively.

3.3 Adaptive Mutation

Adaptive mutation operators have been widely investigated in the related literature and shown to be both theoretically [10, 11] and empirically [43] useful in improving the performance of evolutionary algorithms. In this paper, we propose and investigate a simple linear adaptive mutation in L2-AGE-MOEA. We aim to consider more advanced adaptive schema as part of our future agenda.

Given that we handle a binary problem, L2-AGE-MOEA uses the *bit-flip* mutation, which randomly flips some decision variable values from one to zero (or vice versa) based on the mutation rate R . Instead of using a constant mutation rate, we use a linearly decreasing mutation rate, such as it has a large mutation rate in the initial generation (for better exploration) and a lower mutation rate in the last generation (for better exploitation). More precisely, the mutation rate R at the generation i is computed as follows:

$$R_i = R_{end} + \frac{FES_{max} - FES}{FES_{max}} \times (R_{start} - R_{end}) \quad (6)$$

where R_{start} is the initial mutation rate, while R_{end} is the final mutation rate (with $R_{start} > R_{end}$); FES denotes the number of solution evaluations performed at the generation i and FES_{max} is the maximum number of solution evaluations (total search budget).

4 EMPIRICAL STUDY

To assess the performance of L2-AGE-MOEA in generating mode sorters, we performed a computational experiment with different types of sorters. We describe the setup of our study in Section 4.1 while the results are discussed in Section 4.2.

4.1 Empirical Setup

In this paper, we consider four formulations of the mode-sorter design problem. First, we consider two-mode and three-mode sorters that aim at demultiplex two and three Hermite-Gauss beams, respectively. In the following, we refer to these sorters as Two-Beams and Three-Beams. The two-beams sorter is encoded as a 95×130 binary matrix, corresponding to 12350 decision variables in total. The two-beams sorter is a 90×180 binary matrix with 16200 decision variables in total.

In addition, we consider another additional objective that aims at simplifying the generated devices. As described in Section 2, the main objectives measure how well the device route the input beams towards the target output locations. On top of these objectives, we aim to minimize the number of entries in the device (binary matrix) equal to one, which leads to removing the PMMA layers that are not useful for sorting the beams, simplifying the device structure.

This results in two additional problem formulations, hereafter referred to as TwoBeams+Structure and ThreeBeams+Structure. Given a device X , the objectives for the TwoBeams+Structure problem are as follows:

$$\min \begin{cases} f_1 = \|t(HG_0) - g(X, HG_0)\|_2 \\ f_2 = \|t(HG_1) - g(X, HG_1)\|_2 \\ f_3 = \frac{1}{n \times k} \sum_{i=1}^n \sum_{j=1}^k X(i, j) \end{cases} \quad (7)$$

with $X(i, j)$ being the value of the device/matrix in row i and column j . The TwoBeams+Structure problem has four objectives: three for the beam output locations, and one for the device's structure.

4.1.1 Baselines selection. As baseline, we selected L2-NSGA [9], AGE-MOEA-II [26], and NSGA-II [8]. We selected L2-NSGA since it is the algorithm used by Di Domenico et al. [9] to generate mode sorters. Hence, our goal is to determine whether our approach outperforms the state-of-the-art algorithm used in the literature for the problem addressed in this paper. Our second baseline is AGE-MOEA-II [26] since it corresponds to the starting point we use to develop/design our approach. Finally, we considered NSGA-II as the third baseline because it does not use any linkage learning or front modeling techniques. Furthermore, this baseline allows us to directly compare NSGA-II with L2-NSGA. Note that Di Domenico et al. [9] only reported the results of L2-NSGA without a direct comparison with vanilla NSGA-II, i.e., without linkage learning.

4.1.2 Parameter Values. For all MOEAs, we used the same parameter settings for the shared genetic operators [45]. For all MOEAs, we set the population size $N = 100$ devices and the number of generations $G = 1000$, corresponding to 100K solutions evaluations.

For AGE-MOEA-II and NSGA-II we use the *uniform* crossover [45]. With this operator, each gene in the offspring solution O has a 50% probability of being selected from the first parent A and 50% probability from the second parent B . The choice is made using a number α randomly sampled from the uniformly distributed interval $[0, 1]$. For all MOEAs, we used the crossover probability $p_c=1.00$ [39]. As the mutation operator, we use the bit-flip mutation with mutation rate $R = 1/D$, where D is the number of decision variables [39, 45].

L2-AGE-MOEA required to set some additional parameters. It uses the linkage-learning-based recombination described in Section 3.2. This operator requires setting the cut point for the dendrogram. In our preliminary experiments, we experimented with different values for the cut point used to extract clusters from the dendrogram. We obtained good results by selecting the cut point to $D/4$. Furthermore, L2-AGE-MOEA uses the adaptive mutation rate described in Section 3.3. We set the final mutation rate $R_{end} = 1/D$ and the initial mutation rate $R_{start} = 2\%$. Note that this is a large mutation rate considering that $D = 12350$ and $D = 16200$ for the devices with two and three HG beams to demultiplex, respectively. For both L2-NSGA and L2-AGE-MOEA, we recompute the linkage model (FOS) every $\Lambda=20$ generation (the constant Λ in lines 4 and 25 of Algorithm 2) as suggested by Di Domenico et al. [9].

4.1.3 Implementation. We have implemented L2-AGE-MOEA, the state-of-the-art L2-NSGA, and the four formulations of the device design problem in Matlab within the PlatEMO framework [39]. We have selected this framework because it contains the original implementation of AGE-MOEA-II. It is also publicly available on GitHub², and it is easy to extend by adding more optimization problems and MOEAs. We reimplemented L2-NSGA in PlatEMO, re-using the original code by Di Domenico et al. [9] already written in Matlab and available on Zenodo³. We re-adapted the code to be compatible with PlatEMO by applying as little change as possible to the original code of L2-NSGA. Instead, for AGE-MOEA-II and NSGA-II, we used their implementation already available on PlatEMO.

The source code of L2-AGE-MOEA is also available on Zenodo at: <https://doi.org/10.5281/zenodo.7851998>

We conducted all experiments on the same machine with the following characteristics: 6-Core, Intel Core i7 processor running at 3.2GHz with 16GB RAM.

4.1.4 Performance metrics. Given the stochastic nature of the experimented MOEAs, we ran them 20 times for each problem formulation. In each run, we stored the non-dominated front produced at the end of the search by each MOEA. Then, we assess the overall performance of the different MOEAs by using the *inverted generational distance* (IGD) [49] as well as the *hypervolume* (HV) [29]. Both metrics take values in $[0, 1]$, but they measure different properties of the generated fronts. The IGD measures the distance of the front P_A generated by an algorithm A ; therefore, smaller IGD values are preferable. Instead, HV measures the (hyper)volume (or portion) of the feasible region that is dominated by the generated front P_A . Hence, a large HV value indicates better results.

To compute IGD and HV scores, we need to compare the non-dominated fronts produced by the MOEAs against the optimal

(true) Pareto front. Since we do not know the true front for the mode sorter design problem, we build the so-called *reference front*, which combines the fronts produced by the evaluated MOEAs and selects the overall non-dominated solutions in this union set. More precisely, let $\{F_1, \dots, F_k\}$ be all non-dominated fronts produced the different MOEAs; the reference front \mathcal{R} is built as follows:

$$\mathcal{R} \subseteq \bigcup_{i=1}^k F_i, \forall p_1 \in \mathcal{R}, p_2 \in \mathcal{R} : p_2 < p_1 \quad (8)$$

To calculate the HV score we have to specify a reference point [15] (or reference vector). In this study, we consider the reference point to be $2 \cdot Z^{max}$, where Z^{max} denotes the nadir point of the reference front \mathcal{R} computed using Equation 8.

To assess the significance of the differences among the different MOEAs, we use the Wilcoxon rank-sum test [4] with a significance level $\alpha = 0.05$. A significant p -value indicates that an algorithm A achieves significantly lower IGD values (or significantly higher HV values) than another algorithm B across the 20 runs.

4.2 Results

Table 1 reports the median and the Interquartile Range (IQR) values for the IGD indicator (top half) and HV (bottom half) scores achieved by the different MOEAs over 20 independent repetitions. In the Table, results highlighted with \blacktriangledown indicate that the IGD and HV values achieved by L2-AGE-MOEA are statistically better than the baselines, as indicated by the Wilcoxon rank-sum test [4].

From the tables, we can observe that L2-AGE-MOEA achieves significantly better (lower) IGD values and better (higher) HV values than all baselines, according to the Wilcoxon rank-sum test, independently of the problem formulation. In the following, we discuss the comparison among the different MOEAs separately.

4.2.1 L2-AGE-MOEA vs. L2-NSGA. From a direct comparison between the approach used by Di Domenico [9] and our approach, we can observe that the latter achieves both better HV and IGD values. We can also observe that for Two-Beams, Two-Beams+Structure, and Three-Beams, there is one order of magnitude difference between the IGD values yielded by L2-AGE-MOEA over the baseline. Our approach's IQR values are also smaller, indicating that it produces more stable (less variable) results over the different runs.

Furthermore, L2-NSGA always achieves smaller HV values than L2-AGE-MOEA, with differences of one order of magnitude for the Three-Beams and Three-Beams+Structure devices.

4.2.2 L2-AGE-MOEA vs. AGE-MOEA-II. Our approach also outperforms its predecessor AGE-MOEA-II in terms of IGD values. Note that the latter MOEA does not use linkage learning or the adaptive mutation operator. The differences are statistically significant for to the Wilcoxon test but also remarkably large. For example, on Two-Beams, L2-AGE-MOEA achieves (on average) less than 1/3 of the IGD values produced by the baseline. Similar results can be observed regarding HV values where our approach achieves better scores, e.g., 0.258 vs. 3.35 HV values for Three-Beams+Structure.

4.2.3 AGE-MOEA-II vs. L2-NSGA. This comparison is particularly noteworthy. The latter is the algorithm used by Di Domenico et al. [9] to generate mode sorters and also uses linkage learning

²<https://github.com/BIMK/PlatEMO>

³<https://zenodo.org/record/5115118>

Table 1: Median and Interquartile Range (IQR) values achieved by the different evolutionary algorithms. M denotes the number of objectives while D is the number of decision variables. The best values are highlighted in grey color.

Results for IGD						
Problem	M	D	NSGA-II	AGE-MOEA-II	L2-NSGA	L2-AGE-MOEA
Two-Beams	2	12350	1.4232e+0 (4.85e-1) ▼	8.6429e-1 (3.13e-1) ▼	1.0086e+0 (4.65e-1) ▼	2.6545e-1 (2.19e-1)
Two-Beams + Structure	3	12350	1.2744e+0 (2.40e-1) ▼	7.3520e-1 (3.44e-1) ▼	6.2873e-1 (1.86e-1) ▼	4.1567e-1 (2.02e-1)
Three-Beams	3	16200	1.1844e+1 (1.73e+0) ▼	2.5116e+0 (9.56e-1) ▼	6.4420e+0 (3.62e+0) ▼	3.5408e-1 (2.10e-1)
Three-Beams + Structure	4	16200	6.7468e-1 (2.92e-2) ▼	3.701e-1 (9.61e-2) ▼	2.462e-1 (2.48e-2) ▼	1.9041e-1 (3.86e-3)
Results for HV						
Problem	M	D	NSGA-II	AGE-MOEA-II	L2-NSGA	L2-AGE-MOEA
Two-Beams	2	12350	2.0254e-1 (4.65e-2) ▼	2.6338e-1 (3.51e-2) ▼	2.7187e-1 (4.40e-2) ▼	3.4791e-1 (4.13e-2)
Two-Beam + Structure	3	12350	7.7192e-2 (1.63e-2) ▼	2.1024e-1 (1.76e-2) ▼	1.3591e-1 (7.30e-2) ▼	3.3073e-1 (2.47e-2)
Three-Beams	3	16200	5.2256e-5 (2.17e-3) ▼	2.2892e-1 (3.84e-2) ▼	7.7827e-2 (1.10e-1) ▼	3.7184e-1 (2.34e-2)
Three Beams + Structure	4	16200	2.5452e-2 (1.18e-2) ▼	2.5838e-1 (2.78e-2) ▼	1.4414e-2 (4.82e-3) ▼	3.3585e-1 (1.76e-2)

methods. Instead, AGE-MOEA-II does not use any linkage learning methods but uses front modeling methods. As we can observe, AGE-MOEA-II outperforms L2-NSGA in producing more optimal devices according to the IGD indicator. The differences are also confirmed as statistically significant by the Wilcoxon rank-sum test (p -values < 0.01). Instead, AGE-MOEA-II outperforms L2-NSGA in terms of HV values only for three out of four problems. Indeed, for Two-Beams, both MOEAs achieve a statistically equivalent hypervolume, on average. The statistical difference holds for Two-Beams, Three-Beams, and Three-Beams+Structure (all p -values are < 0.01).

4.2.4 L2-NSGA vs. NSGA-II. A direct comparison between L2-NSGA vs. NSGA-II shows that the former clearly outperforms the latter in terms of IGD values. The only difference between these two MOEAs is that L2-NSGA uses linkage learning methods while NSGA-II does not. The differences in terms of IGD values between these two MOEAs are statistically significant for all problem formulations (all p -values are < 0.01). This also empirically supports the choice made by Di Domenico [9] in using L2-NSGA over NSGA-II.

4.2.5 The impact of using linkage learning. Our results confirm the importance of using linkage learning methods for large-scale binary problems like designing mode sorters. Indeed, it is worth highlighting that L2-AGE-MOEA outperforms its predecessor AGE-MOEA-II, while L2-NSGA outperforms its predecessor NSGA-II according to the median IGD values reported in Table 1.

4.2.6 Graphical comparison. To graphically compare the different MOEAs experimented in this study, Figure 2 depicts the non-dominated fronts achieved for Two-Beams as well as the reference Pareto front \mathcal{R} . For the sake of this analysis, we selected the front with the median IGD values across the 20 independent runs.

As we can observe, L2-AGE-MOEA produced a set of non-dominated solutions (marked as red points) that are very close to the referent front \mathcal{R} . The corresponding median IGD value is 0.2654. Unlike L2-AGE-MOEA, the baseline MOEAs generate fronts that are farthest from the reference front. AGE-MOEA-II produces the second-best non-dominated front, followed by L2-NSGA, while NSGA-II produces the least optimal front.

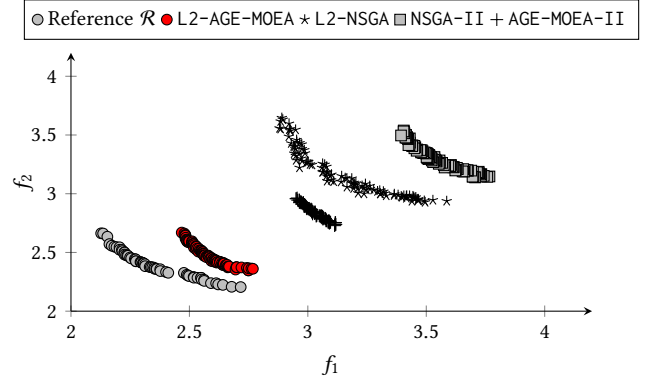


Figure 2: Fronts produced by the different MOEAs for Two-Beams with $M=2$ and population size $N=100$. The reference front \mathcal{R} includes all best (non-dominated) solutions produced among all runs and algorithms as described in Equation 4.1.4 (Equation 8).

We can also observe that the reference Pareto front \mathcal{R} has a hyperbolic geometry (convex) with $p < 1$ curvature. Similarly, all fronts generated by the MOEAs follow a hyperbolic geometry. This observation explains why L2-AGE-MOEA and AGE-MOEA-II are the best-performing MOEAs since they use front modeling methods that adapt the diversity and proximity measures based on this geometry. As reported by prior studies [25, 26], geometry-based MOEAs work better than NSGA-II, NSGA-III and other traditional MOEAs (e.g. MOEA/D) for problems with $p < 1$ Pareto front curvatures.

4.2.7 Generated devices and simulation results. In this section, we aim to analyze the characteristics of the device by selecting the knee points among the non-dominated front obtained by our approach. First, we selected the front with the median IGD value among the 20 independent runs. Then, from such a median front, we selected the knee point [47], which is the solution with the maximum marginal rates of return. We opted for the knee point since it is the point on the Pareto front that achieves the highest HV value [6]. The resulting device and its beam propagation simulation results are

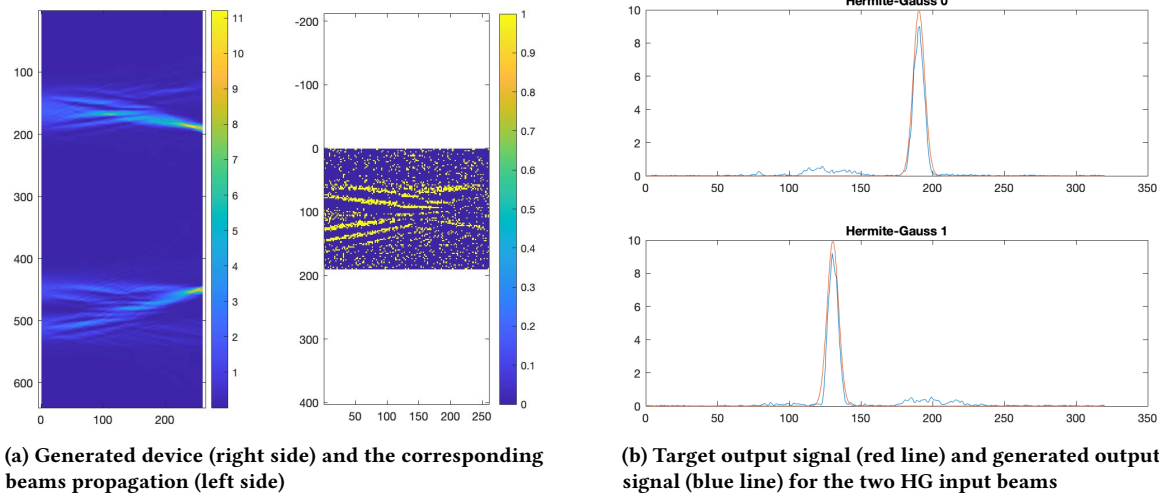


Figure 3: Example of device generated by L2-AGE-MOEA for the Two-Beams+Structure problem.

depicted in Figure 3. Figure 3a plots the device (right-hand side); in the figure, the yellow dots are the device grid entries set to one. Figure 3a also depicts the beam propagation results (left-hand side), showing how the two beams are routed into the desired locations. The plot in Figure 3b highlights the high quality of the beam routing. In the figure, the red line indicates the target output signals, while the blue lines indicate the output obtained by propagating the two Hermite-Gauss beams through the device. As we can observe, the two lines are comparable.

The device in Figure 3a (right side) displays a distinct pattern in the central area, while the four corners do not contribute to beam routing. This suggests two potential improvements for future work: First, the current mutation operator targets all entries in the device/matrix structure, which can be mutated with equal probability. Therefore, the first direction for future improvements is related to using non-uniform probabilities for the entries in the different areas of the devices (higher probability for the central area). Second, different methods could be considered to initialize the search with initial devices (individuals) such that more focus (variety) is given to the central area of the devices.

4.2.8 Additional Analysis. As explained in Section 3.1, we used the *hamming distance* (HD) for the UPGMA method because of its linear complexity. Previous work on linkage learning methods suggested using mutual information (MI) instead [19, 38]. However, this function has a much larger computational complexity compared to HD. To assess this, we performed some preliminary experiments using MI over HD. Inferring the linkage structures and extracting the FOS (routine INFER-MODEL in Algorithm 1) takes 2.7s on average with HD compared to 7.42 minutes with MI for the Two-Beam devices.

5 CONCLUSION AND FUTURE WORK

Optical devices that demultiplex optical signals with different modes have a wide range of real-world applications. However, the design of such devices is particularly complex and can not be performed

manually. Multi-objective evolutionary algorithms (MOEAs) have been recently applied in the literature [9] to generate such devices in an automated fashion. The state-of-the-art approach used L2-NSGA, a variant of the NSGA-II algorithm that relies on linkage learning to produce high-accurate demultiplexers. However, this algorithm is very slow to converge due to the very large scale of the problem (with more than 12K decision variables for a two-beams device).

In this paper, we proposed a novel multi-objective approach called L2-AGE-MOEA. The new MOEA incorporates linkage learning methods into AGE-MOEA-II, a many-objective algorithm that dynamically adapts its environmental selection based on the geometry of the Pareto front. We extend AGE-MOEA-II by introducing a recombination operator based on linkage learning and hierarchical clustering (UPMGA method) in particular [14]. Second, we introduce an adaptive mutation operator that linearly decreases the mutation rate such that exploration and exploitation are promoted in different stages of the search process.

We conducted an empirical study to assess the performance of L2-AGE-MOEA in designing different demultiplexers for Hermite-Gauss beams. A comparison between L2-NSGA and AGE-MOEA-II showed that L2-AGE-MOEA achieved better performance w.r.t. the IGD and HV indicators. Besides, our results also indicate that linkage learning plays a critical role in achieving better results.

In our future work, we aim to assess the performance of our approach on different types of demultiplexers and beams, such as non-linear complex devices and devices capable of sorting beams from different families (e.g., Hermite-Gauss from Airy beams). We plan to investigate different adaptive schemes for the mutation operators (e.g., [10, 11, 43]) as well as different distance functions for the hierarchical clustering algorithm. We also aim to apply L2-AGE-MOEA to other large-scale engineering problems, such as feature selection [40] and sparse signal reconstruction [18].

REFERENCES

- [1] Govind P Agrawal. 2012. *Fiber-Optic Communication Systems*. John Wiley & Sons.

- [2] Julian Blank, Kalyanmoy Deb, and Proteek Chandan Roy. 2019. Investigating the normalization procedure of NSGA-III. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 229–240.
- [3] Robert W Boyd. 2020. *Nonlinear optics*. Academic press.
- [4] W. J. Conover. 1998. *Practical Nonparametric Statistics* (3rd edition ed.). Wiley.
- [5] Mitchell A Cox, Luthando Maqondo, Ravin Kara, Giovanni Milione, Ling Cheng, and Andrew Forbes. 2019. The resilience of Hermite–and Laguerre–Gaussian modes in turbulence. *Journal of Lightwave Technology* 37, 16 (2019), 3911–3917.
- [6] Kalyanmoy Deb and Shivam Gupta. 2011. Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering optimization* 43, 11 (2011), 1175–1204.
- [7] K. Deb and H. Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (Aug 2014), 577–601. <https://doi.org/10.1109/TEVC.2013.2281535>
- [8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.
- [9] Giuseppe Di Domenico, Dror Weisman, Annibale Panichella, Dolev Roitman, and Ady Arie. 2022. Large-Scale Inverse Design of a Planar On-Chip Mode Sorter. *ACS Photonics* 9, 2 (2022), 378–382.
- [10] Benjamin Doerr, Carola Doerr, and Franziska Ebel. 2015. From black-box complexity to designing new genetic algorithms. *Theoretical Computer Science* 567 (2015), 87–104.
- [11] Benjamin Doerr, Carola Doerr, and Johannes Lengler. 2019. Self-adjusting mutation rates with provably optimal success rules. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1479–1487.
- [12] Jörg Enderlein and Francesco Pampaloni. 2004. Unified operator approach for deriving Hermite–Gaussian and Laguerre–Gaussian laser modes. *JOSA A* 21, 8 (2004), 1553–1558.
- [13] Grant R Fowles. 1989. *Introduction to modern optics*. Courier Corporation.
- [14] Ilan Gronau and Shlomo Moran. 2007. Optimal implementations of UPGMA and other common clustering algorithms. *Inform. Process. Lett.* 104, 6 (2007), 205–210.
- [15] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. 2018. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary computation* 26, 3 (2018), 411–440.
- [16] Hao Jia, Ting Zhou, Xin Fu, Jianfeng Ding, and Lin Yang. 2018. Inverse-design and demonstration of ultracompact silicon meta-structure mode exchange device. *ACS Photonics* 5, 5 (2018), 1833–1838.
- [17] Mario Krenn, Johannes Handsteiner, Matthias Fink, Robert Fickler, and Anton Zeilinger. 2015. Twisted photon entanglement through turbulent air across Vienna. *Proceedings of the National Academy of Sciences* 112, 46 (2015), 14197–14201.
- [18] J Liang, M Gong, H Li, C Yue, and BY Qu. 2018. Problem definitions and evaluation criteria for the cec special session on evolutionary algorithms for sparse optimization. *Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, Report# 2018001* (2018).
- [19] Ngoc Hoang Luong, Han La Poutre, and Peter AN Bosman. 2018. Multi-objective gene-pool optimal mixing evolutionary algorithm with the interleaved multi-start scheme. *Swarm and Evolutionary Computation* 40 (2018), 238–254.
- [20] Wei Ma, Maojing Hou, Ruiqi Luo, Bo Xiong, Nan Liu, Guandong Liu, and Tao Chu. 2023. Topologically-optimized on-chip metamaterials for ultra-short-range light focusing and mode-size conversion. *Nanophotonics* 12, 6 (2023), 1189–1197.
- [21] R Timothy Marler and Jasbir S Arora. 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization* 26, 6 (2004), 369–395.
- [22] Alan T Murray and Jiwon Baik. [n. d.]. Opensource spatial optimization in GIScience for strategic positioning. *Transactions in GIS* ([n. d.]).
- [23] Mitchell Olsthoorn and Annibale Panichella. 2021. Multi-objective test case selection through linkage learning-based crossover. In *Search-Based Software Engineering: 13th International Symposium, SSBSE 2021, Bari, Italy, October 11–12, 2021, Proceedings 13*. Springer, 87–102.
- [24] Suat Özdemir, A Attea Bara'a, and Önder A Khalil. 2013. Multi-objective evolutionary algorithm based on decomposition for energy efficient coverage in wireless sensor networks. *Wireless personal communications* 71, 1 (2013), 195–215.
- [25] Annibale Panichella. 2019. An adaptive evolutionary algorithm based on non-euclidean geometry for many-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 595–603.
- [26] Annibale Panichella. 2022. An improved Pareto front modeling algorithm for large-scale many-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 565–573.
- [27] Martin Pelikan, David E Goldberg, Erick Cantú-Paz, et al. 1999. BOA: The Bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99*, Vol. 1. Citeseer, 525–532.
- [28] Alexander Y Piggott, Jesse Lu, Konstantinos G Lagoudakis, Jan Petykiewicz, Thomas M Babinec, and Jelena Vučković. 2015. Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer. *Nature Photonics* 9, 6 (2015), 374–377.
- [29] Ke Shang, Hisao Ishibuchi, Linjun He, and Lie Meng Pang. 2020. A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 25, 1 (2020), 1–20.
- [30] Bing Shen, Peng Wang, Randy Polson, and Rajesh Menon. 2015. An integrated-nanophotonics polarization beamsplitter with 2.4x 2.4 μm² footprint. *Nature Photonics* 9, 6 (2015), 378–382.
- [31] Yuxuan Shen and Yue Pan. 2022. A Multi-objective Optimization Approach of Green Building Performance Based on LGBM and AGE-MOEA. In *Proceedings of the 2022 International Conference on Green Building, Civil Engineering and Smart City*. Springer, 202–210.
- [32] William T. Silfvast. 1997. *Laser Fundamentals*. Cambridge University Press.
- [33] Alicia Sit, Frédéric Bouchard, Robert Fickler, Jérémie Gagnon-Bischoff, Hugo Larocque, Khabat Heshami, Dominique Elser, Christian Peuntinger, Kevin Günther, Bettina Heim, et al. 2017. High-dimensional intracavity quantum cryptography with structured photons. *Optica* 4, 9 (2017), 1006–1010.
- [34] Logan Su, Alexander Y Piggott, Neil V Sapra, Jan Petykiewicz, and Jelena Vučković. 2018. Inverse design and demonstration of a compact on-chip narrowband three-channel wavelength demultiplexer. *ACS Photonics* 5, 2 (2018), 301–305.
- [35] Yikai Su, Yu He, Haoshuo Chen, Xiaoying Li, and Guifang Li. 2021. Perspective on mode-division multiplexing. *Applied Physics Letters* 118, 20 (2021), 200502.
- [36] Orazio Svelto, David C Hanna, et al. 2010. *Principles of lasers*. Vol. 1. Springer.
- [37] Dirk Thierens. 2010. The linkage tree genetic algorithm. In *International Conference on Parallel Problem Solving from Nature*. Springer, 264–273.
- [38] Dirk Thierens and Peter AN Bosman. 2011. Optimal mixing evolutionary algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. 617–624.
- [39] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. 2017. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *IEEE Computational Intelligence Magazine* 12 (2017), 73–87. Issue 4.
- [40] Ye Tian, Chang Lu, Xingyi Zhang, Kay Chen Tan, and Yaochu Jin. 2020. Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks. *IEEE transactions on cybernetics* 51, 6 (2020), 3115–3128.
- [41] Richard A Watson, Gregory S Hornby, and Jordan B Pollack. 1998. Modeling building-block interdependency. In *Parallel Problem Solving from Nature—PPSN V: 5th International Conference Amsterdam, The Netherlands September 27–30, 1998 Proceedings 5*. Springer, 97–106.
- [42] Jingfu Ye, Yan Li, and Shiliang Qu. 2021. On-Chip Orbital Angular Momentum Sorting With a Surface Plasmon Polariton Lens. *Journal of Lightwave Technology* 39, 5 (2021), 1423–1428.
- [43] Jiao-Hong Yi, Suash Deb, Junyu Dong, Amir H Alavi, and Gai-Ge Wang. 2018. An improved NSGA-III algorithm with adaptive mutation operator for Big Data optimization problems. *Future Generation Computer Systems* 88 (2018), 571–585.
- [44] Tian-Li Yu, David E Goldberg, Kumara Sastry, Claudio F Lima, and Martin Pelikan. 2009. Dependency structure matrix, genetic algorithms, and effective recombination. *Evolutionary computation* 17, 4 (2009), 595–626.
- [45] Xinjie Yu and Mitsuo Gen. 2010. *Introduction to evolutionary algorithms*. Springer Science & Business Media.
- [46] Xingyi Zhang, Ye Tian, Ran Cheng, and Yaochu Jin. 2016. A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Transactions on Evolutionary Computation* 22, 1 (2016), 97–112.
- [47] Xingyi Zhang, Ye Tian, and Yaochu Jin. 2014. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 19, 6 (2014), 761–776.
- [48] Yiyi Zhou, Jiapeng Zhao, Zhimin Shi, Seyed Mohammad Hashemi Rafsanjani, Mohammad Mirhosseini, Ziyi Zhu, Alan E Willner, and Robert W Boyd. 2018. Hermite–Gaussian mode sorter. *Optics letters* 43, 21 (2018), 5263–5266.
- [49] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation* 7, 2 (2003), 117–132.