



**HAL**  
open science

# Adding Exploration to Tree-Based MIMO Detectors Using Insights from Bio-Inspired Firefly Algorithm

Bastien Trotobas, Youness Akourim, Amor Nafkha, Yves Louët

## ► To cite this version:

Bastien Trotobas, Youness Akourim, Amor Nafkha, Yves Louët. Adding Exploration to Tree-Based MIMO Detectors Using Insights from Bio-Inspired Firefly Algorithm. IEEE Vehicular Technology Conference, Apr 2021, Helsinki, Finland. hal-03081802

**HAL Id: hal-03081802**

**<https://hal.science/hal-03081802v1>**

Submitted on 18 Dec 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Adding Exploration to Tree-Based MIMO Detectors Using Insights from Bio-Inspired Firefly Algorithm

Bastien Trotobas\*, Youness Akourim<sup>†</sup>, Amor Nafkha\* and Yves Louët\*

\*SCEE/IETR UMR CNRS 6164, CentraleSupélec Avenue de la boulaie, 35576 Cesson Sévigné, France

Email: {bastien.trotobas, amor.nafkha, yves.louet}@centralesupelec.fr

<sup>†</sup>École normale supérieure de Rennes, Campus de Ker Lann, Avenue Robert Schuman, 35170 Bruz, France

Email: youness.akourim@ens-rennes.fr

**Abstract**—The standard multiple-input multiple-output (MIMO) detectors exploit the available information to resolve the detection problem. Alternative algorithms, such as bio-inspired or geometrical detectors, mix exploitation with exploration to bypass local minima and enhance the results. This paper examines the benefits of adding exploration to the traditional tree-based detectors. For this purpose, a new interpretation of the bio-inspired detector based on the firefly algorithm (FA) is proposed. It is studied in a tree search paradigm and extended to soft-outputs. The findings suggest that the addition of a stochastic exploration to tree-based detectors significantly improves performance with a small computational overhead.

**Index Terms**—MIMO detection, firefly algorithm, soft-output, exploration vs. exploitation, tree-search, Pareto analysis

## I. INTRODUCTION

Multiple-input multiple-output systems have made their way into most of the standards such as WiFi (IEEE 802.11n/ac), WiMAX, long-term evolution (LTE), and 5G. This technology exploits the spatial components of the wireless channel to increase link robustness and provide significant capacity gain via transmit diversity and space-division multiplexing (SDM). The latter technology improves throughput and link quality without requiring new frequency bands. However, the substantial increase in data rates comes at the expense of a more complex receiver design. Indeed, all data streams add up in the same time-frequency slot. New detection algorithms are therefore required to separate the data streams and retrieve signals transmitted from different antennas. The MIMO receivers typically rely on pilot signals to get information about the channel state information (CSI).

In spatial multiplexing MIMO systems, the optimum maximum likelihood decoding (MLD) problem leads to an integer least-squares (ILS) problem, which is equivalent to finding the closest lattice point to a given point. Due to the discrete search space of the ILS problem, it is NP-hard [1]. Therefore, it is impossible to find an optimal detection algorithm in reasonable polynomial calculation time on a Turing machine. Such an algorithm could exist if  $P=NP$ , but this possibility is considered unlikely, and no algorithm has been found so far despite intensive researches. Thus, several optimal algorithms, as well as polynomial heuristics, have been developed to tackle the MLD detection problem. Both types of algorithms provide hard and soft outputs versions. Hence, they can both

be coupled with channel coding techniques to improve the MIMO link quality.

The MLD detection problem is an instance of combinatorial optimization problem, which seeks to find the best solution to a problem out of a large discrete set (the feasible set). In general, combinatorial optimization can be solved using classical algorithms that rely on a tree representation. Branch and bound approaches, known as sphere decoding (SD) or depth-first tree-search, provides an optimal solution in a non-polynomial time [2]–[6]. These algorithms can detect in an acceptable time when the number of antennas is limited [7]. The application of Dijkstra’s algorithm, called best-first (BF) tree-search, is another method providing an optimal result in a reasonably short time [8]–[10]. However, this type of detector requires a lot of storage space and is harmed by input dependent run-time. Breadth-first detectors have been introduced as approximations of the previous algorithms to produce results in a polynomial time. Thus, the M-algorithm [11] and its successor, the K-best detector [12]–[14], benefit from predictable the time and storage utilization, still providing near-optimal performance.

The above detectors are based just on tree exploitation step given the available information. Other techniques are available in literature based on mixing exploitation with exploration in order to escape from local minima and produce more reliable results. In particular, bio-inspired metaheuristics such as ant colony optimization (ACO) [15] or FA [16] have been proposed. These algorithms are much more complex than tree-based techniques. However, they are also more flexible. Indeed, they add an exploration component instead of the pure exploitation used in tree-based detectors. Another method used a geometrical approach to perform exploration and exploitation steps has been recently proposed in [17].

This paper investigates the use of an exploratory element in common tree-based algorithms. To achieve this goal, we reformulate the FA detector described in [16] as a tree-path algorithm associated with an exploratory factor. We propose a new tree-based detector with parameters that can be adjusted to allow an exploitation-exploration trade-off. The proposed detector can provide soft outputs for higher-order modulation schemes. Moreover, this detector is compared with well-known tree-based detectors without exploration step and the geometrical exploration-exploitation detector.

The paper is organized as follows. The MIMO system model under consideration, the MLD detection problems, and the QR decomposition are presented in Section II. In Section III, we introduce the hard-output FA detector, and we extend it to produce the soft-output in Section IV. In Section VI, we present the simulation results of the proposed algorithm. Finally, Section VII concludes the paper.

## II. SYSTEM MODEL AND MATHEMATICAL DEFINITIONS

The MIMO model is introduced in Section II-A. Section II-B defines the hard-output, and soft-output detection problems and Section II-C rewrites these problems introducing the QR decomposition.

### A. MIMO Model

We model the  $n \times n$  MIMO system for a single time/frequency slot. This model is suitable for any link such that the interferences between frequency slots and between time slots are negligible. Let  $\mathbf{H} \in \mathbb{C}^{n \times n}$  be the channel matrix such that  $H_{ij}$  is the complex gain corresponding to the path from antenna  $j$  to antenna  $i$ . Let  $\mathcal{Q}$  be the set of all the constellation symbols available. Let  $\mathbf{y} \in \mathbb{C}^n$  be the signal received on each antenna that corresponds to the symbols transmitted  $\mathbf{x} \in \mathcal{Q}^n$  after propagation through the channel added to the circularly-symmetric Gaussian noise  $\mathbf{w} \sim \mathcal{CN}(0, \sigma^2)$ . The link model is then expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}. \quad (1)$$

### B. Definition of the Detection Problems

On the one hand, the hard-output detection problem refers searching for the most reliable transmitted symbols given the channel state and the received vector. This process is equivalent to solve the combinatorial optimization problem

$$\arg \min_{\hat{\mathbf{x}} \in \mathcal{Q}^n} \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|^2. \quad (2)$$

We further denote by the objective function the expression

$$E(\hat{\mathbf{x}}) = \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|^2. \quad (3)$$

On the other hand, the soft-output detection problem denotes the computation of the log-likelihood ratios (LLRs) of each transmitted bit  $b_{ij}$  with  $b_{ij}$  the  $i^{\text{th}}$  bit encoded in the  $j^{\text{th}}$  symbol emitted. The LLRs are commonly approximated using the max-log approximation

$$L_{ij} \approx \frac{1}{2\sigma^2} \left( \min_{\hat{\mathbf{x}} \in \mathcal{X}_{ij}^0} \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|^2 - \min_{\hat{\mathbf{x}} \in \mathcal{X}_{ij}^1} \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|^2 \right), \quad (4)$$

where  $\mathcal{X}_{ij}^k = \{\hat{\mathbf{x}} \in \mathcal{Q}^n : b_{ij} = k\}$  is the set of all symbols with  $b_{ij}$  equals to  $k$  [5], [18], [19]. Most detectors approximate (4) using a list  $\mathcal{L} \subset \mathcal{Q}^n$  rather than computing the  $2^n$  objective functions. Therefore, the LLRs expression becomes

$$L_{ij} \approx \frac{1}{2\sigma^2} \left( \min_{\hat{\mathbf{x}} \in \mathcal{L} \cap \mathcal{X}_{ij}^0} \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|^2 - \min_{\hat{\mathbf{x}} \in \mathcal{L} \cap \mathcal{X}_{ij}^1} \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|^2 \right). \quad (5)$$

If  $\mathcal{L} \cap \mathcal{X}_{ij}^k = \emptyset$ , it is assumed that  $b_{ij} = k$  and  $L_{ij}$  is set to its maximum or minimum value to express the reliability on this bit.

### C. Rewriting using QR Decomposition

Let  $\mathbf{H} = \mathbf{Q}\mathbf{R}$  be the QR decomposition of the channel matrix with  $\mathbf{Q}$  a unitary matrix and  $\mathbf{R}$  an upper triangular one. Since  $\mathbf{Q}$  is an isometry, we can rewrite the objective function as

$$E(\hat{\mathbf{x}}) = \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|^2 = \|\mathbf{Q}^*\mathbf{y} - \mathbf{Q}^*\mathbf{Q}\mathbf{R}\hat{\mathbf{x}}\|^2 = \|\tilde{\mathbf{y}} - \mathbf{R}\hat{\mathbf{x}}\|^2 \quad (6)$$

with  $\tilde{\mathbf{y}} \triangleq \mathbf{Q}^*\mathbf{y}$  the rotated and reflected received vector. We can then introduce the distance  $d_i$  such that

$$E(\hat{\mathbf{x}}) = \|\tilde{\mathbf{y}} - \mathbf{R}\hat{\mathbf{x}}\|^2 = \sum_{i=1}^n d_i \quad (7)$$

where

$$d_i \triangleq \left| \tilde{y}_i - \sum_{j=i}^n R_{i,j} \hat{x}_j \right|^2 \quad (8)$$

This new expression allows for the computation of  $d_i$  using only the  $i$  last components of  $\mathbf{x}$ . Therefore, the detector can decide on the last component (i.e., the symbol of the last antenna) without any assumption on the other symbols. Then, each symbol can be detected from the last to the first without the need for cross-assumptions.

## III. FA-BASED AS A RANDOMIZED TREE-BASED DETECTOR

The firefly algorithm is a bio-inspired metaheuristic that can handle the majority of optimization problems. A particular variant of it has been proposed in [16] to tackle the combinatorial optimization in the hard-output detection problem (2). Section III-A reminds the main steps of this detector in the common swarm-based framework. Section III-B revisits the FA-based detector as a randomized tree-search to highlight that the swarm-based framework and the tree-based one overlap.

### A. FA as a Swarm-Based Detector

The FA detector, as described in [16], is a swarm-based bio-inspired meta-heuristic detector. As an illustration, Fig. 1 shows the firefly algorithm implemented for a MIMO system with  $2 \times 2$  antennas using a BPSK modulation scheme. From a bio-inspired perspective, a transmit antenna can be referred to as a nest, and each constellation symbol is depicted by a stationary firefly (i.e. the stars). The FA detector simulates moving bugs (i.e. the disks) that have to choose a mate in each nest, based on their attractiveness. Once a moving firefly reaches the last nest, its chosen mates correspond to a decoded symbol vector. For instance, a firefly selecting each time the biggest star on the figure will detect  $\hat{\mathbf{x}} = (+1, -1)$ . The swarm is composed of  $F$  bugs that travel at once, and the final result is the best path selected among all the achieved ones (i.e. the one with the lower objective function value).

The attractiveness of the stationary firefly  $i$  to the moving firefly  $m$  is computed as

$$\beta_{i,m} = e^{-\gamma d_{i,m}^k}, \quad (9)$$

where  $\gamma \geq 0$ ,  $k \geq 1$  are two parameters, and  $d_{i,m}$  the distance computed using (8) assuming the previous choices of the firefly  $m$ . As discussed in Section II-C, the QR decomposition allows to compute the attractiveness of a firefly based on the mates chosen on the previously visited nests but no assumptions on the next choices are required. The parameters  $\gamma$  and  $k$  control the exploration-exploitation trade-off. The bigger these parameters are, the more the difference in the objective function are amplified, and therefore, the more the detector focus on good children. Conversely, when these parameters are small, the algorithm is more likely to explore paths rather than focusing on a few promising ones.

A moving firefly selects a mate in each nest based on its attractiveness using a non-uniform random choice. For each nest, the moving firefly  $m$  will choose the mate  $i$  with probability

$$P_{i,m} = \frac{\beta_{i,m}}{\sum_j \beta_{j,m}} \quad (10)$$

with  $\sum_j \beta_{j,m}$  the sum of the attractiveness of all the fireflies in this nest. Therefore, the FA swarm explores several possibilities thanks to the stochastic term rather than using only an exploitation process as in most MIMO detectors.

#### B. FA as a Randomized Tree-Based Detector

The FA detector may be interpreted as a stochastic tree-path search. In this case, each stationary firefly in a nest represents a node in a tree. Besides, a moving firefly run corresponds to a path from the root to a tree leaf. Fig. 2 shows the same example as Fig. 1, highlighting the tree structure. Consequently, decode a message corresponds to build as many tree paths as the number of moving fireflies. The partial path is extended following a stochastic process defined by the probability of (10). The random process can be considered an exploration since it allows extending the path to nodes that are not the best, hoping to be compensated for the over-cost and obtained a better final result.

This paradigm shift provides a more straightforward comparison with many other typical detectors such as K-best or BF. Indeed, the complexity of these algorithms is often measured through the number of visited nodes. This metric is now easily accessible for FA. To illustrate, K-Best visits roughly  $Kn$  nodes with  $K$  the algorithm parameter whereas FA accesses  $Fn$  nodes. Therefore, this paradigm shift highlights that  $K$  and  $F$  have exactly the same role in evaluating the complexity of these two detectors. Section VI-A will introduce a re-evaluation of the FA complexity by exploiting the tree approach.

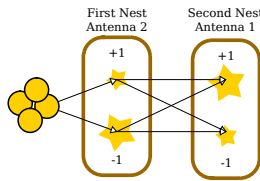


Fig. 1. Representation of FA as a firefly swarm with  $n = 2$  and a BPSK.

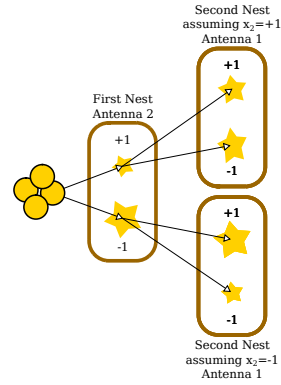


Fig. 2. Representation of FA as tree-search algorithm ( $n = 2$  and BPSK).

### IV. PROPOSED SOFT-OUTPUT FA-BASED DETECTOR

In this section, we propose a new detector based on the FA. Section IV-A introduces the new algorithm that tackles the soft-output detection problem defined in II-B. Then, Section IV-B provides a simplification for the attractiveness computation.

#### A. Description of the Soft-Output FA-Based Detector

Most soft-output MIMO detectors rely on the max-log approximation computed on a restricted list of candidates as described in Section II-B. The FA-based detector can easily be adapted to generate a list of rather than a symbol vector  $\mathbf{x}$ . Indeed, its detection process is based on the parallel construction of  $F$  paths. Therefore, it is possible to build the list  $\mathcal{L}$  from the aggregation of the  $F$  paths and derive the LLRs from equation (5).

Fig. 3 sums up all the steps of the proposed soft-output detector. This pseudo-code is similar to the hard-output detector except for the aggregation of all the built paths in the list  $\mathcal{L}$ . This algorithm can be used as such for any modulation scheme, including higher-order modulations.

#### B. Padé Approximant for the Attractiveness Computation

The attractiveness computation requires the evaluation of an exponential function, which may represent a high cost. Nevertheless, any approximation of (9) has to keep its major characteristics, i.e.,  $\beta_{i,m}$  is strictly positive and strictly decreasing with  $d_{i,m}$ . Hence, we propose to estimate (9) by the simplest Padé approximant satisfying these two properties. The Padé approximant of order (0,1) of the exponential function is given by

$$e^x \approx \frac{1}{1-x}, \quad (11)$$

the attractiveness expression becomes

$$\beta_{i,m} \approx \frac{1}{1 + \gamma d_{i,m}^k} \quad (12)$$

This new formulation replaced the exponential with addition and an inversion, which is easier to compute. In the following, the proposed detector is tested using this attractiveness expression.

---

**Inputs:**

- Received vector:  $\mathbf{y}$
- QR decomposition of the channel matrix:  $\mathbf{Q}, \mathbf{R}$
- Noise variance:  $\sigma^2$

Compute  $\tilde{\mathbf{y}} = \mathbf{Q}^* \mathbf{y}$ .

Initialize an empty candidate list  $\mathcal{L} = \emptyset$ .

**for all** firefly/path  $m \in \{0 \dots F\}$  **do**

Start at the root node with objective function  $E_m = 0$ .

**for** coordinate  $c = n$  **to** 1 **do**

Evaluate the attractiveness  $\beta_{i,m}$  of each child  $i$ .

Compute the probability to select each child using (10).

Select a child randomly according to the probabilities.

Increment the objective function  $E_m$  by  $d_c$  from (8).

**end for**

Add the completed path to the list  $\mathcal{L}$ .

**end for**

Approximate LLRs based on  $\sigma^2$ ,  $\mathcal{L}$  and (5).

**return** the approximated LLRs.

---

Fig. 3. Pseudo-code of the proposed soft-output FA-based detector. The number of fireflies/paths  $F \in \mathbb{N}_*$  is set for all the detections.

## V. EVALUATION OF THE PROPOSED DETECTOR

This section introduces the evaluation criteria and the simulation setup in Section V-A and the reference detectors in Section V-B.

### A. Simulation Settings and Evaluation Criteria

The addition of exploration in the tree search and the proposed algorithm are evaluated according to two opposing criteria: computational complexity and bit error rate (BER) performance. The complexity is assessed through the number of product operations required per detected vector. The Padé approximant presented in Section IV-B simplifies the computation of the exponential into a product and division. The complexity is investigated as a statistic since the detector behavior is stochastic. The statistics of number of the product operations and the BER performance are based on Monte-Carlo simulations. For each SNR, the simulation runs until 200 binary errors occur or  $5 \cdot 10^5$  bits are transmitted. The data message is encoded by the irregular, systematic LDPC code of ratio 1/2 from the WiMAX standard IEEE 802.16e. Each block contains 720 bits of information, and decoding is performed by 15 iterations of the belief-passing min-sum algorithm. For simulation codes, see [20]. The balance between these two competing criteria is explored through a Pareto analysis. A detector is considered Pareto efficient if no algorithm can enhance one criterion without degrading the others. The set of Pareto efficient detectors constitutes the Pareto front, and selecting one or the other depends on the complexity-performance trade-off.

### B. Reference Detection Algorithms

The new detector will be compared to several well-known detectors to evaluate its suitability and determine the relevance of the exploration in the tree-based algorithm. Three detectors

are considered for comparison: two tree-based algorithms without exploration and a geometrical detector mixing exploration and exploitation. These detectors are selected for their differences regarding the exploration step and their well-suited structure for hardware implementation. Those algorithms create a list and use the max-log approximation described in (5) to produce the soft-output values. The remainder of this section briefly discusses the characteristics of each algorithm. The number of product operations of the tree-based detector and the geometrical algorithm was already given in [17].

1) *Breadth-first K-best* [12]: K-best is a breadth-first tree-based detector with sub-optimal performance. This algorithm searches the tree from root to leaf, retaining only a given number of paths at each tree level. This approximation allows it to achieve near-optimal performance with a polynomial-time and storage space. Furthermore, a single parameter adjusts the complexity-performance trade-off being the number of paths retained.

2) *Best-first tree search* [10]: The approximation-free best-first detectors achieve optimal performance but require a significant memory space. Reference [10] introduces an approximated best-first tree search reducing memory usage and maintaining near-optimal performance. This is achieved by constraining the size of the node heaps.

3) *Geometrical-based detector* [17]: The geometrical detector explores based on the channel matrix singular value decomposition (SVD) and then exploits the intermediate result through local search. This combination of exploration and exploitation makes it more efficient than tree-based detectors in some scenarios. Nevertheless, the lack of hindsight on this technique, especially on the mathematical aspect, complicates the refinement of this class of algorithms.

## VI. RESULTS ON COMPLEXITY AND PERFORMANCE

Section VI-A evaluates the impact of the exploration on the complexity compared to the usual tree-search pattern. Then, Section VI-B explores the performance-complexity trade-off to assess the proposed detector's relevance in comparison with state-of-the-art algorithms.

### A. Complexity Cost of Exploration vs. Usual Tree-Search

The conventional tree-based detectors build paths starting from the root and extending some partial paths. By contrast, the proposed FA-based detector independently builds  $F$  paths from the root to a leaf. Thus, the same node can be visited several times when constructing the  $F$  paths. It is then interesting to avoid recomputing the metric and to reuse the values already calculated. Fig. 4 shows the ratio of the number of unique nodes to the total number of visited nodes for a 4x4 MIMO system and a 16-QAM. The three boxplots show the minimum, maximum and quartiles for parameters settings. For the three tests,  $k = 3$  and only  $F$  and  $\gamma$  change. The first two lines only differ in the number of fireflies/paths. They show that, as  $F$  decreases, the ratio of the number of unique nodes increases since the total number of nodes decreases. The last two lines display the evolution of the ratio as a function of

$\gamma$ . As reported in Section III-A, the more  $\gamma$  increases, the more the detector explores more different nodes. The three boxplots in Fig. 4 are typical cases of parameter settings. The boxplots point out that there are always less than 50% unique nodes. Furthermore, the threshold of 20% unique nodes is rarely crossed with two third quartiles below this level and one with a ratio of 24%. This descriptive statistic confirms that it is interesting to share the results rather than to recompute the metric each time the node is encountered. We further assume that each node is computed once for all the paths that visited it. This assumption change neither the pseudo-code from Fig. 3 nor the results of the detector.

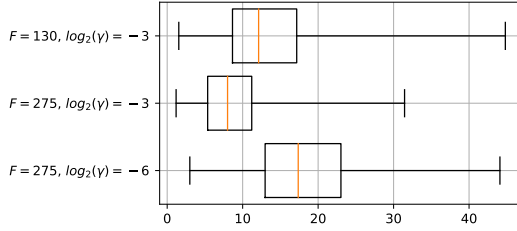


Fig. 4. Boxplot of the ratio of the number of unique to total amount of visited nodes. The ratio is displayed as percents, whiskers represent the minimum and maximum,  $k = 3$  and SNR is set to 17dB.

### B. Performance-Complexity Trade-off Analysis

Fig. 5 displays the Pareto analysis of the performance-complexity trade-off. An exhaustive search sweeps all the tuning parameters of each algorithm to explore the potential of each detector. The algorithms are compared based on the number of products per detected vector and according to the SNR required for a BER of  $1.10^{-4}$ . The Pareto analysis demonstrates that the proposed detector is efficient across a wide operating range from the worst SNRs to the average regimes where it outperforms K-Best and the geometrical detector. Moreover, the proposed detector shows better flexibility than Best-First detection. The previous references are only efficient when complexity is the critical aspect. Indeed, the proposed detector needs a fair number of fireflies/paths to achieve reasonable performance and is therefore not adapted when the complexity must be extremely low.

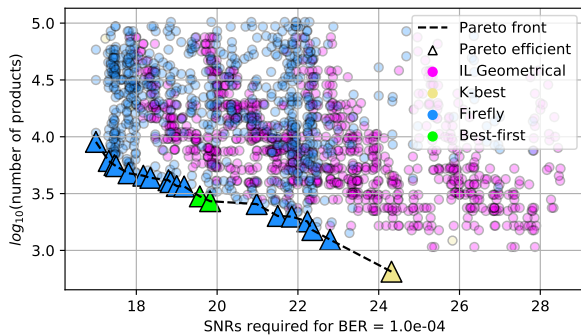


Fig. 5. Pareto analysis of Performance versus Complexity for a 16-QAM.

## VII. CONCLUSION

In this paper, the bio-inspired FA-based detector has been redesigned in a tree path paradigm and extended to soft-output. A Pareto analysis demonstrated its interest compared to several references. Besides, this result indicates that exploration an advantage for tree-based detectors. Furthermore, the exploration-exploitation approach presented is more efficient than the one used in geometrical detectors.

## REFERENCES

- [1] D. Micciancio, "The hardness of the closest vector problem with preprocessing," *IEEE T. Inform. Theory*, vol. 47, no. 3, pp. 1212–1215, Mar. 2001.
- [2] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. of Computation*, vol. 44, no. 170, pp. 463–471, 1985.
- [3] C. P. Schnorr and M. Euchner, "Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems." *Math. prog.*, vol. 66, no. 1-3, pp. 181–199, 1994.
- [4] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE T. Inform. Theory*, no. 8, pp. 2201–2214, Aug. 2002.
- [5] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE T. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.
- [6] C. Studer and H. Bolcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE T. Inform. Theory*, vol. 56, no. 10, pp. 4827–4842, Oct. 2010.
- [7] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE T. Signal Proces.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.
- [8] C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, "A 74.8 mw soft-output detector ic for 8 x 8 spatial-multiplexing mimo communications," *IEEE J. Solid-ST. Circ.*, vol. 45, no. 2, pp. 411–421, Feb. 2010.
- [9] R. Y. Chang and W.-H. Chung, "Best-first tree search with probabilistic node ordering for mimo detection," *IEEE Trans. Wirel. Commun.*, vol. 11, no. 2, pp. 780–789, Feb. 2012.
- [10] G. He, X. Zhang, and Z. Liang, "Algorithm and architecture of an efficient mimo detector with cross-level parallel tree-search," *IEEE T. VLSI Syst.*, vol. 28, no. 2, pp. 467–479, Feb. 2020.
- [11] K. K. Y. Wong and P. J. McLane, "A low-complexity iterative mimo detection scheme using the soft-output m-algorithm," *IST Mob. Sum.*, p. 5, Jun. 2005.
- [12] Z. Guo and P. Nilsson, "Algorithm and implementation of the k-best sphere decoding for mimo detection," *IEEE J. Sel. Area Comm.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.
- [13] Z. Yan, G. He, Y. Ren, W. He, J. Jiang, and Z. Mao, "Design and implementation of flexible dual-mode soft-output mimo detector with channel preprocessing," *IEEE T. Circuits-I*, vol. 62, no. 11, pp. 2706–2717, Nov. 2015.
- [14] S.-J. Choi, S.-J. Shim, Y.-H. You, J. Cha, and H.-K. Song, "Novel mimo detection with improved complexity for near-ml detection in mimo-ofdm systems," *IEEE Access*, vol. 7, pp. 60 389–60 398, 2019.
- [15] J. C. Marinello and T. Abrão, "Lattice reduction aided detector for mimo communication via ant colony optimisation," *Wireless Pers. Commun.*, vol. 77, no. 1, pp. 63–85, Jul. 2014.
- [16] A. Datta and V. Bhatia, "A near maximum likelihood performance modified firefly algorithm for large mimo detection," *Swa. and Evol. Computation*, vol. 44, pp. 828–839, Feb. 2019.
- [17] B. Trotobas, A. Llave, A. Nafkha, and Y. Louet, "When should we use geometrical-based mimo detection instead of tree-based techniques? a pareto analysis," *To be published in IEEE Access*, 2020.
- [18] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference," in *IEEE Global Telecommu. Conf. and Exhib.*, Dec. 1990, pp. 1679–1684.
- [19] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain," in *IEEE Int. Conf. on Commu.*, vol. 2, Jun. 1995, pp. 1009–1013.
- [20] B. Trotobas and Y. Akourim, "Simulation codes," 2020. [Online]. Available: [bitbucket.org/scee\\_ietr/exploration\\_tree\\_search](http://bitbucket.org/scee_ietr/exploration_tree_search)