Effects of the Random Forests Hyper-Parameters in Surrogate Models for Multi-Objective Combinatorial Optimization: A Case Study using MOEA/D-RFTS

Matheus Bernardelli de Moraes, *Student Member, IEEE* and Guilherme Palermo Coelho, *Senior Member, IEEE*

***Abstract*—Surrogate models are techniques to approximate the objective functions of expensive optimization problems. Recently, Random Forests have been studied as a surrogate model technique for combinatorial optimization problems. Nonetheless, Random Forests contain several hyper-parameters that are used to control the prediction process. Despite their importance, research on the effects of these hyper-parameters is scarce. Therefore, this paper performs a systematic investigation of the effects of different combinations of values for the Random Forest hyper-parameters on the approximation of well-known multi-objective combinatorial benchmark problems. The results show that the number of samples to consider when building each tree and the minimum number of samples to be at the leaf node are the two most important hyper-parameters in this context.**

***Index Terms*—Decomposition-Based Optimization, Evolutionary Algorithms, Expensive Objective Functions, Machine Learning, Online Learning.**

I. Introduction

Multi-Objective Optimization Problems (MOP) are problems containing multiple conflicting objective functions that must be optimized simultaneously [1]. Real-world MOP often imposes a challenge for researchers, for instance, where each evaluation requires expensive simulations that demands from a few minutes to several hours of computational time [1]. The expensive nature of these problems impacts the performance of multi-objective evolutionary algorithms (MOEA) as it drastically reduces the amount of possible objective function evaluations. To reduce the impacts of expensive evaluations, researchers have been proposing the application of Surrogate Models (SM), techniques to predict the objective function values using some sort of approximation function. Many SM have been used in the literature of MOP, for instance, Kriging Models [2], Radial Basis Function Networks [3], and Polynomial Regression [4], among others. The development of high-precision, low-cost SM is one of the main research topics for the future of MOP [1].

Most of these techniques have been proposed for continuous optimization problems, where the decision variables have real values. In many real-world problems, the decision variables have a discrete nature, the so-called Combinatorial Optimization Problems [5]. For this purpose, some researchers have been investigating the application of Random Forests (RF) [6] as a technique for surrogate modeling [5], [7]–[11]. The main benefits of using RF as an SM in combinatorial MOP include: (i) its natural ability to handle multiple objective functions, and (ii) its capacity to handle discrete decision variables, including categorical and nominal values.

RF contains a high number of hyper-parameters that are used to control the prediction process. The number of decision trees, the depth of the tree, the minimum number of samples required to split an internal node, the minimum number of samples required to be at the leaf node, the number of decision variables used to build each tree, and the number of samples to draw from the original data set are some examples of the RF hyper-parameters [6]. Despite their importance, a systematic research on the effects of these hyper-parameters on the performance of RF as a surrogate model for combinatorial MOP have not been studied. Upon this motivation, this paper applies a comprehensible methodology to test the performance of RF as a surrogate model for combinatorial MOP. The research is organized into two main sections: first (i) we perform a systematic investigation of the effects of different combinations of RF hyper-parameter values on the estimation of well-known multi-objective combinatorial benchmark problems; (ii) then, we test the best and worst hyper-parameter combinations in a recently proposed random forest-assisted MOEA called MOEA/D-RFTS [11] and compare its performance with other state-of-the-art MOEA.

This paper is organized as follows. Section II introduces the basic concepts used in this work. Section III discusses the related works and our contributions. Section IV presents the framework of MOEA/D-RFTS. Section V discusses the research methodology. Section VI presents the results and discussions. Finally, Section VII concludes this work and discusses future research.

II. Basic concepts

This section presents the basic concepts concerning multi-objective optimization problems (MOP), surrogate models (SM), and random forests (RF).

A. Multi-Objective Optimization (MOP)

Equation (1) defines a generic minimization MOP, without loss of generality:

$$\begin{aligned} \text{minimize} \quad & F(\vec{x}) = \langle f_1(\vec{x}), f_2(\vec{x}), \ldots, f_M(\vec{x}) \rangle \\ \text{subject to} \quad & \vec{x} \in \Omega \end{aligned} \tag{1}$$

Matheus Bernardelli de Moraes and Guilherme Palermo Coelho are with the School of Technology (FT), University of Campinas (UNICAMP), Brazil. E-mails: m121214@dac.unicamp.br and gpcoelho@unicamp.br

where $\vec{x} = [x_1, x_2, \ldots, x_N]$ is the decision vector composed by N decision variables, M is the number of objective functions, $F(\vec{x})$ is the objective function vector containing the results of the individual objective functions f_i, and Ω is the feasible region. The goal of a multi-objective optimization algorithm is to find the set of optimal solutions known as Pareto Set. These solutions are evaluated based on the concept of Pareto Optimality: given two solution vectors $\vec{x} \in \Omega$ and $\vec{y} \in \Omega$, solution $\vec{x}$ dominates $\vec{y}$ if $\vec{x}$ is no worse than $\vec{y}$ in all objectives and $\vec{x}$ is strictly better than $\vec{y}$ in at least one objective. A solution $\vec{x} \in \Omega$ is said to be Pareto Optimal with respect to Ω if and only if there is no $\vec{y} \in \Omega$ for which $\vec{y}$ dominates $\vec{x}$.

In expensive environments, access to the real objective function vector $F(\vec{x})$ is limited. In these cases, surrogate modeling can be employed, which main goal is to estimate the objective functions of a given decision vector through an approximation technique from a limited amount of data $D = \{(\vec{x}_1, F(\vec{x}_1)), \ldots, (\vec{x}_\eta, F(\vec{x}_\eta))\}$, where η is the number of real function evaluations that can be used to train the model [12]. The estimation of the objective functions of a given decision vector is formulated as follows, where $\hat{F}(\vec{x})$ is the predicted objective function and $\varepsilon(\vec{x})$ is the estimation error [12]:

$$F(\vec{x}) = \hat{F}(\vec{x}) + \varepsilon(\vec{x}) \tag{2}$$

B. Random Forests (RF)

RF is an ensemble method that constructs multiple Classification and Regression Trees (CART) and aggregates the individual results. A CART is a supervised learning method with a binary tree structure that predicts the outcome of a target decision vector by learning rules inferred from data [6]. To build a CART, a training data set composed of measured data is required, which can be described as: $\mathrm{D} = \{(\vec{x}_1, F(\vec{x}_1)), (\vec{x}_2, F(\vec{x}_2)), \ldots, (\vec{x}_\eta, F(\vec{x}_\eta))\}$, where D is the training data set, $(\vec{x}_i, F(\vec{x}_i))$ is a pair composed of a decision vector and its associated objective functions, and η is the size of the training data set. To build the binary tree structure, the input space is split into NR disjoint and non-overlapping regions $\{R_1, R_2, \ldots, R_{NR}\}$, and each region r provides a prediction $\hat{F}_r = \langle \hat{f}_1(\vec{x}), \hat{f}_2(\vec{x}), \ldots, \hat{f}_M(\vec{x}) \rangle$. Figure 1 illustrates the CART binary structure and its regions.

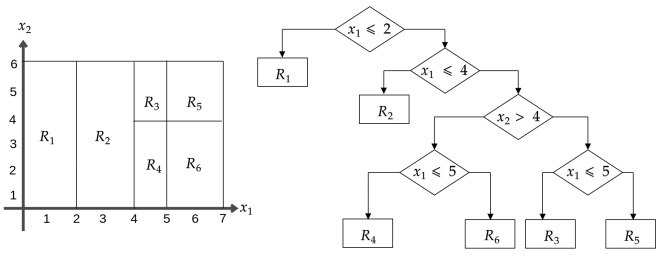

Fig. 1. Illustration of a CART binary tree structure on a hypothetical problem containing two decision variables, x_1 and x_2. Depending on their value, a given decision vector will be associated with its corresponding region. Each region r will produce a prediction $\hat{F}_r = \langle \hat{f}_1(\vec{x}), \hat{f}_2(\vec{x}), \ldots, \hat{f}_M(\vec{x}) \rangle$.

Therefore, in a RF, the final prediction is the average among the CART predictions. RF are built to reduce the prediction noise often found in individual decision trees [6]. RF involves two main randomization process: (i) train each CART with a randomly selected subset of samples with replacement from the entire data set D, and (ii) selection of a subset H of decision variables from the N available decision variables $(H < N)$. Step (i) is known as bootstrap sampling, where samples from the data set are randomly selected, with the possibility of being selected more than once [6]. By employing randomization of samples (i) and also of the decision variables (ii), RF guarantee that each CART is unique and the average result is immune to overfitting [6]. Therefore, given an unevaluated decision vector $\vec{y}$ that must have its objective functions predicted, a RF regression model with multiple trained CART can be described as:

$$\hat{F}_{RF}(\vec{y}) = \left[\frac{1}{C} \sum_{c=1}^{C} \hat{f}_1^c(\vec{y}), \ldots, \frac{1}{C} \sum_{c=1}^{C} \hat{f}_M^c(\vec{y}) \right] \tag{3}$$

where $\hat{f}_i^c(\vec{y})$ is the prediction of the objective function f_i by CART c, and C is the total number of CART. RF have several hyper-parameters that are used to tune its prediction. The main hyper-parameters this paper investigates are: (i) the number of CART built by the RF; (ii) the maximum depth of the tree; (iii) the minimum number of samples required to split an internal node; (iv) the number of features to build each tree; and (v) the number of samples to draw from the original data set *with replacement* (bootstrap sampling).

III. Related work

RF have been analyzed in problems with discrete decision variables, which have a completely different structure than continuous optimization, only in a few works. For instance, [5] developed a random forest-assisted MOEA, whereas the number of trees and the number of decision variables were considered as hyper-parameters. [7] proposed a data-driven optimization framework for combinatorial problems that combines RF and support vector machines. The authors considered only the number of trees as a RF hyper-parameter. In [8], [9], two random forest-assisted algorithms for combinatorial optimization were proposed, based on particle swarm optimization and an improved stochastic ranking strategy, respectively, and the authors used the number of trees and samples as RF hyper-parameters. In [10], the authors proposed a random forest-assisted evolutionary algorithm that was combined with a competitive neighborhood search mechanism, where the number of trees was the only hyper-parameter considered. These methods were tested on the constrained Multi-Objective Knapsack Problem [13].

Finally, [11] presented MOEA/D-RFTS, a random-forest assisted MOEA, which was tested on several combinatorial constrained and unconstrained benchmark problems. The defined hyper-parameters were the number of trees, number of samples, number of decision variables, the minimum number of samples to be at the leaf node, depth, and the minimum number of samples required to split a node. However, no

hyper-parameter investigation was performed, and the selected values were based on previous knowledge of the problems.

Each one of these works considered a different subset of the available RF hyper-parameters. In addition, their specific values are also divergent and were defined based on either previous knowledge or through a hyper-parameter tuning that considered only the investigated optimization functions. Therefore, it becomes fairly difficult to understand the impacts of each RF hyper-parameter on the prediction process of problems with discrete decision variables. Our work contributes to the literature by providing a systematic investigation of the effects of several RF hyper-parameter configurations on the prediction process of different constrained and unconstrained combinatorial benchmark problems. Therefore, this work provides a baseline for hyper-parameter configuration to guide future research that use RF as a SM technique for combinatorial optimization.

IV. Framework of MOEA/D-RFTS

MOEA/D-RFTS is a surrogate-assisted MOEA that uses RF to explore the search space of combinatorial MOP where access to the real objective functions is limited due to expensive evaluations [11]. The overall framework of MOEA/D-RFTS can be described as follows:

1) **Initialization**: the algorithms initialization is performed with the following steps:
   a) Randomly generate the initial population of size P;
   b) Create a set of P uniformly spread weight vectors using Simplex Lattice Design [14], and associate each weight vector to a corresponding individual i;
   c) Generate the initial reference point (in the objective space), which contains the best objective functions found so far;
   d) Define the neighborhood of each individual i, which contains the T closest individuals to i based on the Euclidean distance (in the weight vector space), where T is an user-input parameter;
   e) Create a tabu list, a collection of all decision vectors created so far with no duplicates;
   f) Create an external population archive, which stores all non-dominated solutions found so far;
   g) Create and train a RF with the initial population data.
2) **Main loop**: this step is performed until all offspring have been created. For each individual, select two parents from its current neighborhood. If the parents have different decision vectors, generate an offspring using crossover and mutation operators. If not, do the following:
   a) Calculate the center vector in the objective space of the current neighborhood.
   b) Generate P unique individuals, using local search, that are not in the Tabu List.
   c) Estimate objective function values of the P individuals using the Random Forest.
   d) Compute the Euclidean Distance between each objective function estimation and the Center Vector.
   e) Select as offspring the solution with the closest estimation to the Center Vector.
3) **Update population and model**: update the current population with the new offspring, through a decomposition function such as Tchebycheff's [14]. Train the Random Forest with the new offspring data. Update the external population if new non-dominated solutions have been found. Update the tabu list to include the new decision vectors. Also, update the reference point if better objective functions have been found;
4) **Termination**: if the stopping criteria have been achieved, end the execution and return the external population. If not, go back to the Main Loop (2).

As can be seen, RF prediction is an important functionality of MOEA/D-RFTS. A proper definition of the RF hyper-parameters can increase or decrease the algorithm's ability to identify solutions that will effectively improve its convergence. Therefore, it is crucial to understand the impacts of the RF hyper-parameters in the convergence process of algorithms such as MOEA/D-RFTS.

V. Methodology

This section presents information about the benchmark problems, performance metrics, and experimental setup. This paper employs a methodology to test the RF in an *online learning* process, since the decision vectors are not given beforehand, but become available one by one or in batches as the optimization progresses. Training and testing highly differ from the conventional offline learning process, where all data are assumed to be available. Figure 2 presents the flowchart of the employed methodology.

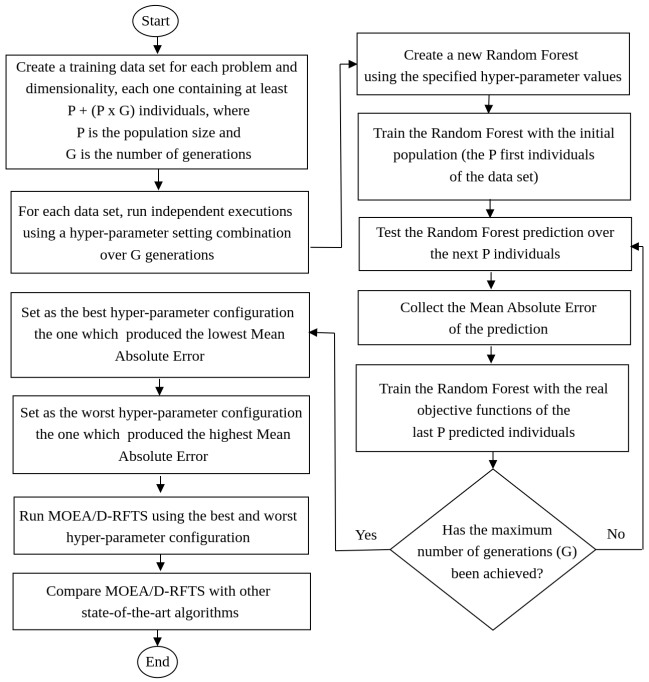

Fig. 2. Research methodology.

A. Benchmark Optimization Problems

The first benchmark is the *Binary Multi-objective Knapsack Problem* (BIN_MOKP), a NP-hard, multi-modal, constrained

combinatorial optimization problem [13]. Considering a set of N items and a set of K knapsacks, each item j has an associated profit $p_{k,j}$ and weight $w_{k,j}$, where $p_{k,j}$ is the profit of item j according to knapsack k and $w_{k,j}$ is the weight of item j according to knapsack k. Each knapsack k has a capacity c_k. The optimization goal is to find the set of solutions that maximizes Equation (4) [13]:

$$\begin{aligned} \text{maximize} \quad & F_k(\vec{x}) = \sum_{j=1}^{N} p_{k,j} \cdot x_j \quad \forall k \in \{1, \ldots, K\} \\ \text{subject to} \quad & \sum_{j=1}^{N} w_{k,j} \leq c_k : \forall j \in \{1, 2, \ldots, N\} \\ & c_k = 0.5 \sum_{j=1}^{N} w_{k,j} \\ & x_j \in \{0, 1\} \end{aligned} \tag{4}$$

where $x_j = 1$ if and only if j is selected. Infeasible individuals are repaired using a maximum profit/weight ratio mechanism, given in Equation 5, that interactively removes the items with the lowest profit per weight unit until all capacity constraints are fulfilled [13]:

$$q_j = \max_{i=1}^{K} \left\{ \frac{p_{k,j}}{w_{k,j}} \right\} \tag{5}$$

A test instance for each problem dimensionality was generated, considering N uncorrelated weights and profits, where w and p are random integers in the interval [10, 100].

The second test instance is an unconstrained variant of the BIN_MOKP called *Binary Multi-Objective Unconstrained Combinatorial Optimization Problem* (BIN_MUCOP). In the BIN_MUCOP, there is no capacity constraint, thus negative profits are allowed. Thus, an item can have a positive profit in one knapsack and a negative profit in the other. Equation 6 defines the BIN_MUCOP:

$$\begin{aligned} \text{maximize} \quad & F_k(\vec{x}) = \sum_{j=1}^{N} p_{k,j} \cdot x_j \quad \forall k \in \{1, \ldots, K\} \\ \text{subject to} \quad & x_j \in \{0, 1\} \end{aligned} \tag{6}$$

Similar to the BIN_MOKP, a test instance was generated for each dimensionality, with N uncorrelated profits, where p is a random integer in the interval [-50, 50].

Finally, the last benchmark problem is a modified version of the BIN_MUCOP called INT_MUCOP, which allows the picking of each item up to five times, which increases the search space and the problem complexity. Equation (7) defines the INT_MUCOP:

$$\begin{aligned} \text{maximize} \quad & F_k(\vec{x}) = \sum_{j=1}^{N} p_{k,j} \cdot x_j \quad \forall k \in \{1, \ldots, K\} \\ \text{subject to} \quad & x_j \in \{0, 1, 2, 3, 4, 5\} \end{aligned} \tag{7}$$

In these benchmark problems, the objective functions are conflicting, since each item has uncorrelated and divergent profits for each knapsack. Therefore, there is not a single solution that maximizes the profits of all knapsacks, but several non-dominated solutions that best consider the trade-offs between the objective functions.

B. Performance Metrics and Statistical Analysis

This work uses the Inverted Generational Distance (IGD) [15] and the Mean Absolute Error (MAE) [16] as the optimization and model evaluation metrics, respectively. The IGD computes the Euclidean distance between each objective vector of the approximated front and its closest objective vector in the reference set, averaged over the size of the reference set [15]. Equation 8 describes the general formula of the IGD:

$$IGD(A, PF) = \frac{1}{PF} \sum_{i=1}^{|PF|} \min_{j=1}^{|A|} \text{dist}(PF_i, a_j) \tag{8}$$

where A is the approximation front, PF is the true PF (or a reference set), a_j is the j-th element of A and PF_i is the i-th element of PF. In this work, finding the true PF of each test instance can only be achieved after all possible combinations of variables have been found, which cannot be achieved by a MOEA in polynomial time [17]. For this matter, we use Reference Sets. These sets contain the best solutions found by all evaluated algorithms, in all independent executions, for each test instance. The number of solutions in each reference set is variable, depending on the complexity and dimensionality of the test instance. Due to the characteristics of the test problems, the reference sets have a convex geometry. The Fast Non-Dominated Sorting [18] method was applied to obtain the non-dominated solutions found by each compared algorithm, at each stage of the optimization problems, and also to obtain the reference sets for each test instance.

MAE calculates the average absolute distance between each prediction and the respective objective function, according to the Equation 9:

$$\text{MAE} = \sum_{i=1}^{N_{pred}} \frac{|\hat{F}(\vec{x}_i) - F(\vec{x}_i)|}{N_{pred}} \tag{9}$$

where $\hat{F}(\vec{x}_i)$ are the objective function predictions, $F(\vec{x}_i)$ are the true objective functions, and N_{pred} is the total number of predictions. The non-parametric Wilcoxon test [19] was used with a significance level of 5% in all problems to identify whether or not the best hyper-parameter configuration of MOEA/D-RFTS achieved statistically significant superiority over the compared methods.

C. Experimental Setup

MOEA/D-RFTS was compared with two state-of-the-art multi-objective algorithms, including the conventional MOEA/D [14] and MOEA/D-NFTS [20]. All experiments were run 10 times independently and the initial population was randomly generated. Algorithm configurations and genetic operators were defined according to the studies in [20] and [11]. The number of individuals was set to 100 for the bi-objective problems and 120 for the three-objective problems. The stopping criterion was the number of real objective function evaluations, fixed at 1,000 for two-objective problems and 1,200 for three-objective problems. A small number of real objective function evaluations were defined to simulate an expensive environment where the use of SM would be

justifiable. Concerning the RF hyper-parameter values, the following possible values were defined, based on values that would be plausible for the selected benchmark problems:

- **trees**: 10, 30, 50, and 100;
- **samples_split**: 2, 4, 6, and 8;
- **depth**: 10, 30, 50, and None. None means the tree grows until all leaves are pure or contain a number of samples less than samples_split;
- **samples_leaf**: 2, 4, 6, and 8;
- **variables**: 40%, 60%, 80%, and 100% of the original number of variables;
- **samples**: 40%, 60%, 80%, and 100% of the original number of samples;

VI. Results and discussions

Table I presents the best and worst RF hyper-parameter configurations found for each problem. Starting with the number of *trees*, the results show that using higher numbers of trees tends to produce lower MAE for all test instances. Concerning the tree *depth*, increasing the tree depth produces smaller errors in the bi-objective problems. However, despite some exceptions, the majority of results show that the limitless growth of a tree will not necessarily promote a more accurate prediction. For almost all three-objective test instances, a medium-sized depth produces a more precise prediction.

Concerning the number of *samples*, it is clear that using a larger data sampling results in smaller errors, since the best combinations always contain the maximum allowed number of samples, in contrast to the worst combinations, which use the minimum allowed number of samples. This is an important aspect of the online learning process, since not considering all the data may lead to selecting a subset of samples that are distant from both the decision and the objective spaces of the newly created offspring solutions, which will eventually lead to poor approximations.

As the number of *decision variables*, for the binary bi-objective benchmark problems, using all variables leads to higher quality approximations. The RF require all data available from the decision variables to identify the differences between the decision vectors since they can be very similar due to their binary nature. This behavior was not observed in the test instances containing integer variables. For the INT_MUCOP problem, using the smallest allowed number of decision variables led to higher quality approximations. Since this problem contains a larger search space, it is better to reduce the number of decision variables to avoid including unnecessary information that would interfere with the learning process.

The minimum number of samples to split is another important source of information. There is a consensus that using the smallest possible number of samples is always the best choice for all problems, number of objectives, and dimensionalities.

Nonetheless, according to this study, the most important hyper-parameter is the minimum number of samples to be at the leaf node. To demonstrate this evidence, Figure 3 shows the MAE variations for each hyper-parameter value for the BIN_MUCOP problem with $M = 2$ objectives and $N = 100$ decision variables. There is a considerably high divergence between the prediction error considering the lowest value (1) and the highest value (8). Forcing the leaf node to retain more samples than necessary penalizes the prediction as more samples will affect the average result of that particular region. Thus, this hyper-parameter should be defined carefully.

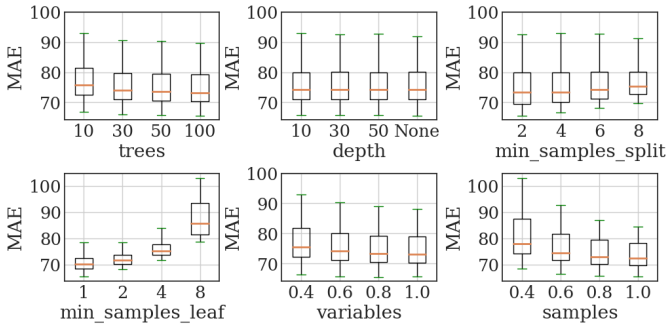

Fig. 3. Boxplot of MAE variations for all possible values for each hyper-parameter in the BIN_MUCOP problem with $M = 2$ objectives and $N = 100$ decision variables.

Following the investigation, an optimization procedure was performed to compare the effects of the best and worst combinations of the RF model hyper-parameters on the performance of MOEA/D-RFTS. Table II presents the performance comparison between MOEA/D-RFTS and its competitors, namely MOEA/D [14], a well-known and state-of-the-art MOEA, and MOEA/D-NFTS [20], which uses diversity preservation mechanisms and knowledge-assisted local search methods to enhance the exploration of the search space of the conventional MOEA/D in expensive problems.

It is notable the importance of a wise hyper-parameter definition. The best combination of RF hyper-parameter values allows MOEA/D-RFTS to achieve statistically superior optimization results than MOEA/D and MOEA/D-NFTS in, respectively, 14 and 11 test instances out of 18 total test instances.

Comparing the MOEA/D-RFTS optimization results with the best hyper-parameter combination with the optimization with the same algorithm, but with the worst evaluated combination, it achieved superior performance in 9 out of 18 total test instances. However, the evidence suggests that hyper-parameter tuning is most important in bi-objective problems. For three-objective problems, the best combination had equivalent performance to the worst combination in 6 out of 9 test instances, higher performance in only 2 and even a worst result in one of them. Increasing the number of objectives enhances the RF performance as the model is trained with more information, therefore it becomes less sensitive to the hyper-parameter definition. This evidence is supported by the average MAE difference between the best and the worst combinations (collected from Table I) among three-objective test instances, which is 30.62, in contrast to bi-objective problems, which is 51.07.

In summary, performing a hyper-parameter tuning of the RF hyper-parameters showed to be an important step to achieve better convergence in combinatorial MOP. We did not find a single combination that works for every scenario, but the

TABLE I

BEST AND WORST HYPER-PARAMETER CONFIGURATIONS FOR EACH BENCHMARK PROBLEM, NUMBER OF OBJECTIVES (M), AND NUMBER OF DECISION VARIABLES (N) OVER 10 INDEPENDENT RUNS. THE MAE (LOWER IS BETTER) COLUMN CORRESPONDS TO THE AVERAGE VALUE OVER 10 GENERATIONS. FOR THE COMPLETE RESULTS FOR EACH INDEPENDENT EXECUTION, PLEASE REFER TO THE SUPPLEMENTARY MATERIAL [21].

Problem	M	N	Rank	Tress	Depth	Samples_Split	Samples_Leaf	Features	Samples	MAE
BIN_MOKP	2	100	Best	100	50	2	1	100%	100%	69.40
			Worst	10	30	4	8	40%	40%	101.16
		300	Best	100	30	2	1	100%	100%	116.27
			Worst	10	10	2	8	40%	40%	185.37
		500	Best	100	30	2	1	100%	100%	151.30
			Worst	10	30	4	8	40%	40%	250.12
BIN_MOKP	3	100	Best	50	30	2	1	40%	100%	62.15
			Worst	10	10	6	8	40%	40%	82.57
		300	Best	100	50	2	1	60%	100%	113.43
			Worst	10	50	2	8	60%	40%	156.70
		500	Best	100	30	2	1	40%	100%	115.33
			Worst	10	10	2	8	40%	40%	159.62
BIN_MUCOP	2	100	Best	100	None	2	1	80%	100%	65.49
			Worst	10	50	8	8	40%	40%	103.04
		300	Best	100	10	2	1	100%	100%	101.38
			Worst	10	50	4	8	40%	40%	163.20
		500	Best	100	None	2	1	100%	100%	100.84
			Worst	10	10	6	8	40%	40%	179.01
BIN_MUCOP	3	100	Best	100	10	2	1	40%	100%	57.90
			Worst	10	50	6	8	40%	40%	89.83
		300	Best	100	None	2	1	40%	100%	81.83
			Worst	10	None	4	8	40%	40%	107.92
		500	Best	100	None	2	1	80%	100%	82.85
			Worst	10	30	4	8	40%	40%	113.83
INT_MUCOP	2	100	Best	100	30	2	1	40%	100%	35.89
			Worst	10	10	4	8	40%	40%	56.82
		300	Best	100	None	2	1	40%	100%	45.71
			Worst	10	30	8	8	40%	40%	70.37
		500	Best	50	50	2	1	100%	100%	67.40
			Worst	10	50	6	8	40%	40%	104.28
INT_MUCOP	3	100	Best	100	30	2	1	40%	100%	35.89
			Worst	10	10	4	8	40%	40%	56.82
		300	Best	100	None	2	1	40%	100%	45.71
			Worst	10	30	8	8	40%	40%	70.37
		500	Best	100	30	2	1	40%	100%	84.09
			Worst	10	30	8	8	60%	40%	117.52

TABLE II

MEAN $\pm$ STANDARD DEVIATION OF THE IGD (LOWER IS BETTER) METRIC ON THE BENCHMARK PROBLEMS OVER 10 INDEPENDENT RUNS, ACCORDING TO THE NUMBER OF OBJECTIVES (M) AND THE NUMBER OF DECISION VARIABLES (N). THE BEST RESULTS ARE HIGHLIGHTED IN BOLD. THE MARKERS "+", "=" AND "-" INDICATE THAT MOEA/D-RFTS (BEST) IS STATISTICALLY SIGNIFICANTLY SUPERIOR (+), EQUIVALENT (=), OR INFERIOR (-) TO THE COMPARED ALGORITHMS, RESPECTIVELY, ACCORDING TO THE WILCOXON TEST WITH A CONFIDENCE LEVEL OF 0.05. THE LAST LINE SHOWS THE NUMBER OF INSTANCES THAT A GIVEN METHOD ACHIEVED SUPERIOR ("+")/EQUIVALENT ("=")/INFERIOR ("-") IGD IN COMPARISON WITH MOEA/D-RFTS (BEST).

Problem	M	N	MOEA/D	MOEA/D-NFTS	MOEA/D-RFTS (Worst)	MOEA/D-RFTS (Best)
BIN_MOKP	2	100	519.5 $\pm$ 106.6 (+)	473.2 $\pm$ 81.0 (+)	350.3 $\pm$ 78.4 (+)	**250.8 $\pm$ 43.9**
		300	1450.0 $\pm$ 196.4 (+)	1447.7 $\pm$ 165.7 (+)	1321.0 $\pm$ 149.6 (+)	**1147.2 $\pm$ 186.8**
		500	2868.9 $\pm$ 193.8 (+)	2735.4 $\pm$ 402.1 (+)	2726.3 $\pm$ 308.8 (+)	**2451.0 $\pm$ 240.1**
BIN_MOKP	3	100	447.7 $\pm$ 55.2 (+)	361.8 $\pm$ 72.5 (+)	296.7 $\pm$ 69.3 (=)	**287.1 $\pm$ 64.7**
		300	1313.3 $\pm$ 145.1 (+)	**1211.2 $\pm$ 158.8 (-)**	1291.0 $\pm$ 148.1 (=)	1266.3 $\pm$ 162.8
		500	2677.3 $\pm$ 350.0 (+)	2589.0 $\pm$ 231.5 (+)	**2305.5 $\pm$ 397.3 (=)**	2427.0 $\pm$ 231.4
BIN_MUCOP	2	100	307.1 $\pm$ 46.3 (+)	293.5 $\pm$ 37.0 (+)	217.1 $\pm$ 48.7 (+)	**183.8 $\pm$ 43.6**
		300	1635.8 $\pm$ 103.9 (+)	1583.3 $\pm$ 128.8 (+)	1288.3 $\pm$ 123.7 (+)	**1196.4 $\pm$ 121.8**
		500	3025.2 $\pm$ 208.2 (+)	3024.2 $\pm$ 284.7 (+)	2679.1 $\pm$ 216.7 (+)	**2515.7 $\pm$ 191.5**
BIN_MUCOP	3	100	279.5 $\pm$ 26.7 (+)	252.8 $\pm$ 40.7 (+)	248.2 $\pm$ 30.2 (+)	**218.1 $\pm$ 30.5**
		300	1337.4 $\pm$ 122.8 (+)	1290.3 $\pm$ 121.3 (+)	1312.7 $\pm$ 168.3 (=)	**1219.2 $\pm$ 87.2**
		500	2509.9 $\pm$ 261.9 (+)	2462.6 $\pm$ 225.4 (+)	**2323.4 $\pm$ 214.0 (=)**	2364.1 $\pm$ 205.3
INT_MUCOP	2	100	528.6 $\pm$ 33.8 (+)	500.7 $\pm$ 57.6 (=)	489.4 $\pm$ 60.3 (+)	**478.7 $\pm$ 69.3**
		300	1501.7 $\pm$ 164.1 (=)	1493.7 $\pm$ 130.8 (=)	1518.5 $\pm$ 115.5 (=)	**1472.4 $\pm$ 70.3**
		500	2327.5 $\pm$ 221.5 (=)	**2314.1 $\pm$ 154.8 (=)**	2361.0 $\pm$ 202.9 (=)	2317.8 $\pm$ 175.8
INT_MUCOP	3	100	412.0 $\pm$ 47.1 (+)	**364.8 $\pm$ 27.4 (=)**	400.0 $\pm$ 67.6 (+)	375.8 $\pm$ 40.6
		300	1142.8 $\pm$ 181.8 (-)	**1126.0 $\pm$ 78.2 (-)**	1132.0 $\pm$ 104.7 (-)	1227.8 $\pm$ 104.5
		500	**2075.0 $\pm$ 167.1 (=)**	2093.2 $\pm$ 183.0 (=)	2100.0 $\pm$ 141.9 (=)	2081.6 $\pm$ 161.6
+ / = / -			1 / 3 / 14	2 / 5 / 11	1 / 8 / 9	

evidence shows that the number of samples to consider when building each CART and the minimum number of samples to be at the leaf node are less problem-dependant and can severely impact the RF prediction process.

VII. CONCLUSIONS AND FUTURE WORK

This work presented a research investigation concerning the impacts of the Random Forest (RF) hyper-parameters in the surrogate modeling of multi-objective combinatorial optimization problems. The results show that proper hyper-parameter

tuning is important to achieve statistically superior results, especially in bi-objective problems. The RF model tends to become less sensitive to the hyper-parameter definition when the number of objectives increases. The two most important hyper-parameters to be defined are the number of samples to use and the minimum number of samples to be at the leaf node, which must be selected carefully. For future research, we intend to evaluate the RF performance in more extreme conditions, considering even fewer data, and also in problems containing more than three objective functions.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of EPIC – Energy Production Innovation Center, hosted by the University of Campinas (UNICAMP) and sponsored by Equinor Brazil and FAPESP – São Paulo Research Foundation (2017/15736-3). We acknowledge the support of ANP (Brazil's National Oil, Natural Gas and Biofuels Agency) through the R&D levy regulation. Acknowledgements are extended to the Center for Energy and Petroleum Studies (CEPETRO), School of Mechanical Engineering (FEM) and the School of Technology (FT). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Financing Code 001.

REFERENCES

[1] C. A. Coello Coello, S. González Brambila, J. Figueroa Gamboa, M. G. Castillo Tapia, and R. Hernández Gómez, "Evolutionary multiobjective optimization: open research areas and some challenges lying ahead," *Complex & Intelligent Systems*, vol. 6, no. 2, pp. 221–236, 2020.

[2] Z. Tan and H. Wang, "A Kriging-Assisted Evolutionary Algorithm Using Feature Selection for Expensive Sparse Multi-Objective optimization," *2020 IEEE Congress on Evolutionary Computation, CEC 2020 - Conference Proceedings*, no. 61976165, 2020.

[3] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2014.

[4] Z. Zhou, Y. S. Ong, M. H. Nguyen, and D. Lim, "A study on polynomial regression and gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm," in *2005 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2832–2839 Vol. 3, 2005.

[5] H. Wang and Y. Jin, "A Random Forest-Assisted Evolutionary Algorithm for Data-Driven Constrained Multiobjective Combinatorial Optimization of Trauma Systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 536–549, 2018.

[6] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[7] Y. Zheng, X. Fu, and Y. Xuan, "Data-driven optimization based on random forest surrogate," in *2019 6th International Conference on Systems and Informatics (ICSAI)*, pp. 487–491, 2019.

[8] Q. Gu, Q. Wang, X. Li, and X. Li, "A surrogate-assisted multi-objective particle swarm optimization of expensive constrained combinatorial optimization problems," *Knowledge-Based Systems*, vol. 223, p. 107049, 2021.

[9] Q. Gu, Q. Wang, N. N. Xiong, S. Jiang, and L. Chen, "Surrogate-assisted evolutionary algorithm for expensive constrained multi-objective discrete optimization problems," *Complex & Intelligent Systems*, jan 2021.

[10] L. Han and H. Wang, "A random forest assisted evolutionary algorithm using competitive neighborhood search for expensive constrained combinatorial optimization," *Memetic Computing*, vol. 13, no. 1, pp. 19–30, 2021.

[11] M. B. De Moraes and G. P. Coelho, "A random forest-assisted decomposition-based evolutionary algorithm for multi-objective combinatorial optimization problems," in *2022 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, 2022.

[12] S. K. Dasari, A. Cheddad, and P. Andersson, "Random forest surrogate models to support design space exploration in aerospace use-case," in *Artificial Intelligence Applications and Innovations* (J. MacIntyre, I. Maglogiannis, L. Iliadis, and E. Pimenidis, eds.), (Cham), pp. 532–544, Springer International Publishing, 2019.

[13] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[14] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[15] C. A. Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 2972, pp. 688–697, 2004.

[16] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005.

[17] Y. Tian, C. He, R. Cheng, and X. Zhang, "A Multistage Evolutionary Algorithm for Better Diversity Preservation in Multiobjective Optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. November, pp. 1–15, 2019.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, apr 2002.

[19] F. Wilcoxon, "Individual Comparisons by Ranking Methods," in *Breakthroughs in Statistics*, pp. 196–202, 1992.

[20] M. B. de Moraes and G. P. Coelho, "A diversity preservation method for expensive multi-objective combinatorial optimization problems using Novel-First Tabu Search and MOEA/D," *Expert Systems with Applications*, vol. 202, p. 117251, sep 2022.

[21] M. B. de Moraes and G. P. Coelho, "Replication data for: Effects of the random forests hyper-parameters in surrogate models for multi-objective combinatorial optimization - A Case Study using MOEA/D-RFTS." Repositório de Dados de Pesquisa da Unicamp.

Matheus Bernardelli de Moraes is a Systems Analyst (University of Campinas - UNICAMP), with a M.Sc. degree (also from UNICAMP). He is currently pursuing the Ph.D. degree in Technology at the School of Technology (UNICAMP). His current research interests include multi-objective evolutionary optimization, bio-inspired algorithms, machine learning, petroleum production strategy optimization and their applications.

Guilherme Palermo Coelho is a Computer Engineer (University of Campinas - UNICAMP), with M.Sc. and Ph.D. degrees in Electrical Engineering (also from UNICAMP). He is an IEEE Senior Member and, currently, an Associate Professor at the School of Technology of the University of Campinas, Brazil. His research interests are Computational Intelligence in general, with recent work on metaheuristics for optimization (single and multi-objective), data mining, and machine learning.