The Analysis of a Cooperative Hyper-Heuristic on a Constrained Real-world Many-objective Continuous Problem

Gian Fritsche
Computer Science Department
Federal University of Paraná
Curitiba, Paraná, Brazil
https://orcid.org/0000-0001-5629-9206

Aurora Pozo
Computer Science Department
Federal University of Paraná
Curitiba, Paraná, Brazil
aurora@inf.ufpr.br

***Abstract*—Despite the significant number of strategies proposed for many-objective optimization, most studies lack the evaluation of their algorithms in real-world applications. In this research, we present the analysis of the application of a recently proposed hyper-heuristic for many-objective optimization to the wind turbine design problem. This application is a representative problem for renewable energy, with 32 variables, 22 constraints, and five objectives. In this study, we solved this problem using the cooperative hyper-heuristic (HH-CO), which uses eight MOEAs on its pool, and switches among them during the search. These MOEAs are representative strategies to deal with multi and many-objective optimization; within HH-CO, every MOEA has its internal population. This behavior is particularly suitable for many-objective optimization, where each MOEA has its criteria to select the solutions that will be preserved for the next generation. Besides, the MOEAs exchange information after being applied, with this migration step, all MOEAs update their private information even if they were not employed. We also evaluate the set of eight state-of-the-art multi-objective evolutionary algorithms (MOEAs), adapted to handle constraints. The empirical analysis uses the hypervolume (HV) quality indicator to evaluate the results and, the HH-CO achieved results competitive to the best MOEAs.**

***Index Terms*—Many-objective optimization, Continuous optimization, Real-world applications, Constrained optimization, Selection Hyper-heuristics, Evolutionary Algorithms, Cooperative Hyper-heuristics**

I. Introduction

Many-objective optimization is a relevant topic in the evolutionary multi-objective field. Therefore, different benchmark problems have been proposed in the last two decades, for example, DTLZ [1], WFG [2], and MaF [3]. In response, several algorithms have appeared to tackle these problems. However, these MOEAs (Multi-objective Evolutionary Algorithms) are usually not evaluated on real-world applications. As a consequence, there are only a few studies on many-objective real-world problems [4]. A probable reason is the lack of well-established real-world problem formulations, mainly when dealing with continuous optimization.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 40001016034P5

Recently, the 3rd Evolutionary Computation Competition presented a real-world many-objective problem [5]. This problem is an instance of the wind-turbine design for wind power generation, a relevant problem for renewable energy. It provides the modeling for the problem and an evaluation module that computes the objectives and constraints given the variable values. In addition, it established a methodology for evaluating algorithms in this problem. It is important to highlight that this problem, and most real-world problems, have constraints that must be satisfied. Commonly, researches on many-objective evolutionary algorithms do not concern about dealing with constraints. On the other hand, there are examples of how to extend dominance-based and decomposition-based MOEAs to deal with these challenges [6].

In many-objective optimization, it is usually discussed that Pareto based evolutionary algorithms do not scale well when the number of objectives to be optimized increases. The primary characteristic observed is the lack of selection pressure, since, in a high dimensional space, most solutions become non-dominated to each other (not comparable). Therefore, different types of MOEAs were proposed, especially those based on decomposition, reference points, and indicators. However, depending on the problem characteristics, Pareto based MOEAs can achieve results similar to or better than state-of-the-art many-objective evolutionary algorithms [7]. Thus, choosing an appropriate algorithm to solve a problem becomes another difficult task to be solved.

In this scenario, the use of hyper-heuristic can contribute to solving many-objective optimization problems. Hyper-heuristics (HH) are high-level strategies for selecting (or generating) heuristics [8]. They assume that there is no single algorithm that can solve every problem in a reasonable time. In other words, if one algorithm is the best for one problem, another algorithm is better for another. This behavior can be observed on many-objective optimization since the performance of the algorithms depends on the problem characteristics [7]. Hence, HH may contribute to solving a many-objective problem. In the online case, the HH task is choosing the algorithms to be applied for a given problem during the search. Moreover, the best performing MOEA may be different for different stages of the search; therefore, the hyper-heuristic switches from MOEAs as the evolution progresses.

In this research, the recently proposed wind-turbine design problem is solved using a set of eight algorithms, representing different strategies for many-objective, and HH-CO, a recently proposed hyper-heuristic with favorable results compared to a state-of-art hyper-heuristic HH-LA [9]. Additionally, as it is a constrained problem, a constraint handling approach is incorporated into the algorithms. Thus, this study contributes to understanding the problem characteristics and the behavior of state-of-the-art MOEAs on a real-world application.

The remainder of this study is organized as follows: we present background about many-objective optimization, real-world applications, and multi and many-objective evolutionary algorithms, at Section II. Section III details the recently proposed Cooperative Hyper-heuristic and review other HHs for multi-objective optimization. Next, Section IV shows the empirical analysis, including the hypervolume during the search and the choices made by the hyper-heuristic. Finally, Section V presents the main findings and guidelines for future works.

II. Background

In this section, we present definitions and discuss some difficulties in solving many-objective optimization problems. We describe a classification for multi and many-objective evolutionary algorithms. Next, we present the strategies used in this study to deal with constraints. Finally, some real-world many-objective applications and their main characteristics are presented.

A. *Many-objective Optimization*

A multi-objective problem, represented by (1), searches for a set of values for n decision variables ($\mathbf{x}$), every one bounded by a lower (L) and upper (U) value, that minimizes M objective functions simultaneously (due to the duality principle, maximization problems can become minimization) [10]. Further, the solutions must satisfy K equality and J inequality constraints, such that $K, J \geq 0$.

$$\begin{array}{ll} \text{Minimize} & (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \\ \text{subject to} & g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \ldots, J, \\ & h_k(\mathbf{x}) = 0, \quad k = 1, 2, \ldots, K, \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \ldots, n. \end{array} \tag{1}$$

A many-objective problem is a problem where the number of objectives is higher than three. This subcategory has received the attention of the researchers due to the difficulties it presents [11], [12]. First, there is a lack of selection pressure since, in a high-dimensional space, most solutions become non-dominated to each other. Thus, the algorithm finds difficulties in distinguishing which solution is better than others, thus degrading the search ability. Second, the computational cost for both fitness evaluation and the algorithm operations increases considerably. Besides, the crossover operation becomes inefficient, as parents tend to be distant to one another. Moreover, it is hard to maintain the diversity of the solutions, since the required number of solutions to represent the Pareto front (which is a hyper-surface) grows exponentially as the number of objectives increases. Finally, when the dimensionality is higher than three, the visual representation of the solutions is hard, which challenges the decision-maker at choosing the final solution.

B. *Multi and Many-objective Evolutionary Algorithms*

Multi-objective problems may present Pareto front with different shapes due to non-linearities, multi-modalities, or others. Concavities in the objective front are quite common on many real problems, leading to discontinuities in the Pareto front. Therefore, derivative-based methods may have difficulties in optimizing them. Evolutionary Algorithms (EAs) are particularly suitable for multi-objective optimization with conflicting objectives due to their population-based meta-heuristic, gradient-free, black-box characteristics [4], [13], [14]: EAs can evaluate a set of solutions simultaneously and can output a set of solutions with different trade-offs between objectives in a single run. Further, EAs have a relatively low computational cost to get an acceptable approximation set and can solve a variety of optimization problems.

When applied to multi-objective problems, the EAs are called MOEAs. Usually, Pareto-based MOEAs, such as NSGA-II and SPEA2, are successful for problems with two and three objectives. However, increasing the number of objectives, new difficulties arise, as commented before. Due to these difficulties, different types of MOEAs were proposed [11]. For instance, some of these algorithms are quality indicators based, such as HypE and MOMBI2. Others are based on decomposition, like the MOEA/D, or on reference points, such as the NSGA-III. There are also strategies like the SPEA2SDE that uses modified diversity maintenance and the ThetaDEA, which uses a dominance relation other than Pareto. These eight algorithms are representative of different classes of MOEAs. Furthermore, their performance, compared to one another, depends on the problem at hand [9] — for more information about many-objective optimization, see [7], [11], [15].

1) Constraint handling: Only a few studies handle constraints in many-objective optimization [6]. In this research, we used the following strategy: for dominance-based algorithms (including Pareto-dominance such as NSGA-II, and other dominance relations, such as ThetaDEA and MOMBI2), the method computes the objective values and the violation of each constraint — i.e., how far the constraint value was from being feasible. Then, it computes the overall constraint violation of the solution, i.e., the sum of the violations for every constraint. Finally, every time that the algorithm needs to compare two solutions ($\mathbf{x}_1$, $\mathbf{x}_2$), this method defines when $\mathbf{x}_1$ dominates $\mathbf{x}_2$ by [6]:

1) $\mathbf{x}_1$ is feasible and $\mathbf{x}_2$ is infeasible,
2) $\mathbf{x}_1$ and $\mathbf{x}_2$ are infeasible and $\mathbf{x}_1$ has smaller constraint violation value, or

3) $\mathbf{x}_1$ and $\mathbf{x}_2$ are feasible and $\mathbf{x}_1$ dominates $\mathbf{x}_2$ (using some dominance relation).

Similarly, this method can be extended for decomposition-based MOEAs, replacing the dominance relation in the third condition by the aggregation function. In this study, we include this constraint handling method into the eight evaluated algorithms.

C. Real-world applications

There is in the literature examples of real-world many-objective applications. These examples include the engineering design [16], a generic formulation aiming at minimizing a set of constraints. Other examples are the air traffic control problem, this problem has hard (must be satisfied) and soft (must be minimized) constraints [17]; scheduling problems, such as nurse rostering [18], with more than 20 objectives; radar waveform design [19], with nine objectives and an integer decision space; hybrid car controller [20], with seven objectives; space trajectory design [21] that has up to six objectives; vehicle routing ($M = 6$) [4]. Other many-objective examples are the dispatch of produced electrical power and some problems from Search-Based Software Engineering (SBSE) [4]. A relevant example is the water resource planning [22], with five objectives and seven constraints. This problem uses recorded precipitation data for planning storm-drainage systems for West Lafayette city. Some works used this problem as validation, together to benchmark problems [6].

Most of these examples are discrete problems. The continuous problems usually fall into the engineering design. However, the descriptions provided are usually not enough for reproducibility; frequently, the objective functions, decision variable ranges, and constraints are not fully described. In other cases, the optimization requires a simulation module, commonly not easily accessible. In this study, we use the Wind Turbine Design problem, described next.

1) Wind Turbine Design Optimization Problem: It is a problem proposed for the 3rd Evolutionary Computation Competition organized by the Japanese Society of Evolutionary Computation [5]. All information about this problem is available in the competition website.[1] This problem has 32 real coded variables that represent measures and parameters of the turbine. In detail: four variables for blade chord length in different blade span directions; the blade maximum chord length position; four directions of blade mounting angle; five directions of spar-cap thickness; five for trailing edge panel thickness; three for blade precurve; the ratio between rotational speed and wind speed; maximum rotation speed; blade length; tower waist position; three height directions of tower outer diameter and three of tower thickness. The upper and lower bound variate for each variable, however, they are normalized encoded between 0.0 and 1.0 and later converted by the evaluation module.

[1]http://www.jpnsec.org/files/competition2019/EC-Symposium-2019-Competition-English.html

For the many-objective formulation, there are five objectives. All objectives are of minimization, using negative values for the first objective:

1) Annual power production: to increase profits.
2) Average annual cost: to reduce power generation costs.
3) Tower base load: to reduce construction cost.
4) Blade tip speed: to reduce noise.
5) Fatigue damage: to extend the service life.

Besides, there are 22 constraints, described in the form $g(\mathbf{x}) > 0$. When $g(\mathbf{x})$ is greater than zero, the constraint is satisfied. The constraints prevent damages, such as the collision of the blade with the tower, avoid resonance, and overload. Also, to ensure minimum life, guarantee manufacturability and weldability, and limits noise. Therefore, there are also constraints to prevent failures and to avoid non-physical solutions [5].

Moreover, the competition provides an evaluation module that computes the objectives and constraints given as input the decision variables. Besides, it provides a post-processing tool for calculating the hypervolume from the history of solutions found during the search. The maximum number of fitness evaluations (FE) is $10,000$. According to the competition, the module takes about three seconds to evaluate a single solution. Therefore it would take about eight hours only to evaluate all $10,000$ generated solutions [5]. This number of FE is relatively small, in comparison, a problem with the same number of decision variables, in the MaF benchmark, would run for $320,000$ FE [3].

III. Cooperative Hyper-Heuristic

In this research, we focus on the heuristic selection for controlling multiple metaheuristics [8]. The studies in this field can be split into three kinds [9]. In the first, the hyper-heuristic adaptively divides the population, i.e., the better the performance of a MOEA on the past generation, the higher is the number of individuals of the share population it will receive. Examples in this kind are the Multi-Indicator Genetic Algorithm (MIGA) [23], AMALGAM [24] and MOABHH [25] based on Copeland voting.

The second kind is competitive hyper-heuristics. In this category, at each decision point, the HH selects a MOEA and applies it for a fixed duration. Then, the executed MOEA delivers the final population to the subsequent MOEA. The next algorithm is selected based on its past performance. An example of this category is the Choice-Function based hyper-heuristic CF-HH [26], that online selects from three MOEAs — it outperformed the three MOEAs and AMALGAM. Another example is HH-LA [27], based on Learning Automata — HH-LA achieved better results than CF-HH.

Finally, there are cooperative hyper-heuristics. In this category, the HH iteratively selects one algorithm and applies it. Then, the employed algorithm shares solutions to other algorithms. At this step, every algorithm updates its internal population and parameters. In this study, we focus on this category, specifically on the Cooperative Hyper-heuristic (HH-CO) [9], a recently proposed HH. HH-CO was successfully applied to

Algorithm 1: Cooperative based Hyper-heuristic

```
   Data: pool of MOEAs
1  initialization;
2  while the stop criterion is not met do
3      foreach $moea \in MOEAs$ do
4          $old_i \leftarrow$ copy_population(moea);
5      end
6      selected $\leftarrow$ heuristic_selection(MOEAs);
7      offspring $\leftarrow$ execute(selected);
8      foreach $moea \in MOEAs$ do
9          migration(moea, offspring);
10         $new_i \leftarrow$ copy_population(moea);
11         reward $\leftarrow$ get_improvement($old_i$, $new_i$);
12         set_reward(moea, reward);
13     end
14 end
```

many-objective optimization, compared to the MOEAs that composed its pool and HH-LA. The next section details the HH-CO procedure and its parameters. For more information about multi-objective hyper-heuristics, see Section 6 of [8].

A. HH-CO

HH-CO is a recently proposed hyper-heuristic for many-objective optimization [9]. It achieved competitive results when compared to the MOEAs from its pool, and better results than a state-of-the-art HH. HH-CO uses the cooperation of multiple MOEAs to solve a problem instance. Its main characteristics are that every algorithm keeps its information (such as population, nadir approximation, and others) and communicates to others during all the search. The hyper-heuristic decides which algorithm is going to generate solutions in the next iteration. (Algorithm 1 details HH-CO [9]).

HH-CO receives as input a set of MOEAs and the problem instance to be solved. First, it initializes the MOEAs populations and their parameters; and the heuristic selection method. Next, it executes the main loop until the stop criterion is satisfied. In the main loop, it copies the population of all MOEAs. Later, a selection criterion chooses a MOEA from the pool. This MOEA is then executed and generates new solutions. At the cooperative phase, it sends the newly generated solutions to all other MOEAs. Then, each MOEA filters the incoming solutions using its environmental selection method — therefore, incorporating solutions according to their criteria. Finally, HH-CO rewards every MOEA based on the improvement of its population in this iteration. Based on this reward, the selection method is going to choose which algorithm will produce the next generation [9].

HH-CO uses the R2 improvement to reward the heuristics. The R2 measures the average of the best Tchebycheff aggregation value for each weight vector (we used the same vectors as NSGA-III [6]). Compared to the hypervolume, the R2 presents less computational cost (begin better suited to be used during the search); and it is assumed to favor uniformly distributed fronts [28]. After the reward, HH-CO applies a greedy selection strategy; i.e., it chooses the one with the best reward in the last iteration. Also, the communication topology is broadcast, as every MOEA shares information with all other MOEAs. However, the implementation supports any other topology strategy, as every MOEA keeps a list of neighbors [9].

IV. Empirical Analysis

In this section, we present an empirical analysis of eight state-of-the-art MOEAs and the HH-CO on a recently proposed many-objective problem. In this study, HH-CO uses eight MOEAs on its pool, namely: NSGA-II, SPEA2, ThetaDEA, NSGA-III, MOMBI2, SPEA2SDE, HypE, and MOEA/D. We selected those state-of-art algorithms due to their diversity of characteristics and good results presented in the literature. Including Pareto based MOEAs, since recent research has demonstrated that they can outperform many-objective EAs depending on the problem characteristics, even for a high number of objectives (e.g., 10) [7]. It is important to note that all the algorithms were modified to handle constraints.[2] Further, HH-CO reward, which uses the improvement of the R2 metric, was also adapted to consider the constraint violations (using the decomposition-based strategy). In these analyses, each MOEA uses its default parameter setting proposed in the literature [9]. Therefore, simulating an off-the-shelf use, rather than a fine-tuning to find the best parameter setting for each algorithm. The discussions are drawn over the different strategies used by each MOEA. Finally, the parameter setting of each MOEA is the same when applied individually or within the hyper-heuristic.

The problem instance is the Wind Turbine Design problem, with five objectives, 32 variables and 22 constraints, detailed at Section II-C1. A module, publicly available [5], evaluates the objective functions and constraints. The maximum number of fitness evaluations (FE) is $10,000$, and the population size is 210 for all algorithms [12]. That yields to 47 iterations ($9,870$ FE), including the random initialization of the populations. Finally, the hypervolume (HV) quality indicator evaluates the results. The HV measures the volume of the space dominated by an approximation front, bounded by a reference point [7] (given by the competition). Besides, the HV values presented are normalized and takes only the feasible solutions that dominate the reference point.

A. Wind Turbine Design Analysis Results

For the empirical analysis, we evaluated the hypervolume during the search of the eight MOEAs and the HH-CO. The hypervolume is computed over an unbounded repository of non-dominated solutions (i.e., every time one generates a new solution, its repository is updated, and the HV computed). In this analysis, hypervolume equals zero represents that no feasible solution dominates the reference point (probably, all solutions found so far are infeasible). Next, we present two

[2]The implementation of all MOEAs uses the jMetal framework: https://github.com/jMetal

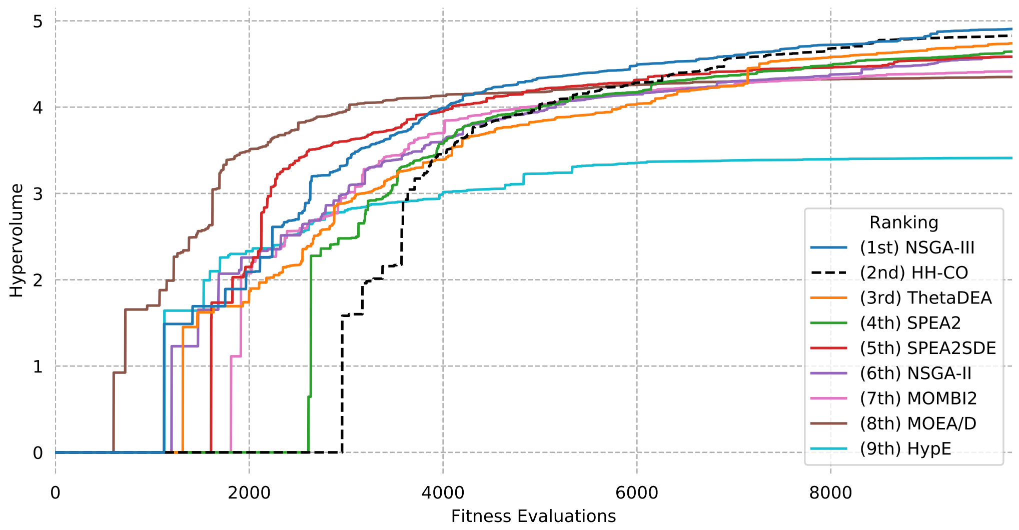

Fig. 1: The median hypervolume of an external repository of non-dominated solutions for the HH-CO (dashed line) and every MOEA (other lines) during the search.

analyses: first, the analysis of the median HV trial (considering the external repository at the end of the search) of each MOEA and HH-CO. Then, an analysis of the overall behavior for 21 runs.

1) Analysis for the median run: In Fig. 1, the dashed line shows the HV for HH-CO, and the other lines the HV of every MOEA. First, the best performing MOEA, considering the median HV, was NSGA-III, the second-best MOEA was ThetaDEA, note that it performed worse than most MOEAs during most of the search (from FE $2,000$ to FE $7,000$). Then, SPEA2 and SPEA2SDE achieved similar HV at the end. However, SPEA2 was the slowest MOEA to start converging, (about $2,500$ FE), on the other hand, SPEA2SDE was one of the best performing MOEAs on the beginning (from FE $2,000$ to FE $6,000$). Furthermore, NSGA-II and MOMBI2 were no better or worse than other MOEAs considering all the search. Finally, MOEA/D was the best MOEA up to FE $4,000$, but it stagnated and finished with the second worse result, followed only by HypE.

In this analysis, the HH-CO achieved the second-best hypervolume, worse than only NSGA-III. It is noticeable that HH-CO took longer than any MOEA to start converging since it generates a set of solutions with random values for each MOEA. Therefore, it took $8 \times 210 = 1,680$ FE only to initialize. After that, it converged and had results better than most MOEAs from FE $6,000$ onwards. The HV values of every MOEA during the search can help us to understand the choices of the HH-CO at different stages of the search, for example, which MOEAs it should select more often.

Fig. 2, presents the choices made by the HH-CO during the search. It makes its first choice after $1,680$ FE (i.e., after the initialization). We can observe that ThetaDEA was the most selected MOEA (8 times). ThetaDEA and SPEA2 were the most selected from the middle to the end of the search. That is, when they perform better, as we have seen on the hypervolume analysis. Likewise, SPEA2SDE and NSGA-II were the most selected from the beginning to the middle of the search. The HH-CO selected MOMBI2, NSGA-III, and HypE uniformly distributed over the search. Finally, MOEA/D was the less selected MOEA. The preference for SPEA2SDE at the beginning could be explained by its diversity preference (exploration); on the other hand, the application of ThetaDEA (with $\theta = 5$) at the end could provide more convergence (exploitation) [7].

It is noticeable that the MOEA with the best hypervolume when applied alone (NSGA-III), was not among the most selected MOEAs by HH-CO. A possible explanation is the greedy selection method applied. It seeks to find MOEAs improving now rather than one that is consistently improving. However, as the knowledge of every MOEA is kept internally safe, these wrong choices are not so critical for the search. Finally, we can notice that HH-CO selected all MOEAs at least once. In other words, every MOEA improved more than all others at a given point in the search. That points out to the importance of the diversity of a set of MOEAs that is used by HH-CO.

2) Overall analysis: For some analysis, we made use of boxplots. The boxplot allows visualizing the distribution of data through their quartiles. A box extends from the first to the third quartile (the interquartile range - IQR) and measures variability, with a line at the median. Circles represent outliers, i.e., data points distant from the box more than $1.5 \times IQR$ [29].

In Fig. 3, we present the boxplot analysis for hypervolume achieved by every MOEA and HH-CO. This analysis gives

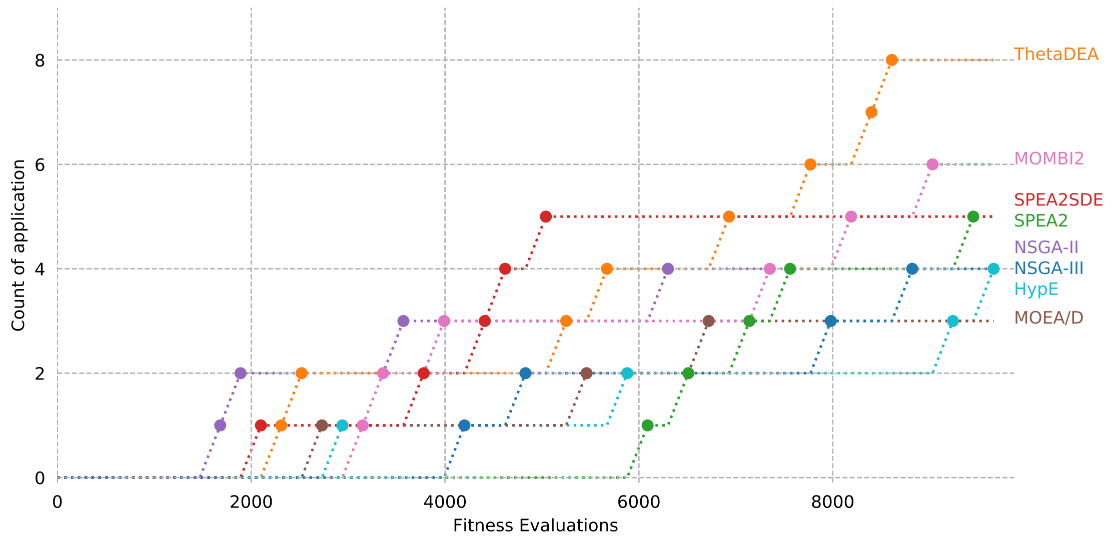

Fig. 2: The count of how many times HH-CO applied each MOEA. It was selected the median HV trial for this analysis.

support to the critical difference plot presented in Fig. 4. The critical difference plot connects the algorithms to their average ranking of 21 runs. A bold horizontal line connecting two (or more) algorithms represents that the algorithms dit not achieved a statistically significant difference to one another. This overall statistical evaluation is pairwise performed by the Nemenyi test, with 95% significance [30].

It is possible to observe that the HV of the MOEAs highly variates from the best to the worst. Also, we can observe that the two best MOEAs are from those specifically designed for many-objective (NSGA-III and ThetaDEA). In general, the best performing algorithms for this problem were NSGA-III, followed by the hyper-heuristic (HH-CO), and ThetaDEA. From those three, the statistical test did not find a significant difference. From the boxplot, we observe that the HH-CO was the one with the smallest variability in the results among different runs. Besides, SPEA2 and NSGA-II, Pareto-based MOEAs, achieved competitive results, without a significant difference to the state-of-the-art MOEAs like ThetaDEA and SPEA2SDE. Finally, the worse performing MOEAs for this problem where MOEA/D, MOMBI2, and HypE. Although the boxplot shows differences, the statistical test did not find a significant difference from HypE and the other four worse-performing MOEAs.

Fig. 5 presents the accumulated number of applications for every MOEA. In general, HH-CO applied each MOEA about five times. SPEA2 and SPEA2SDE stand out, applied around 7.5 times each. On the other hand, NSGA-II was the less applied MOEA. Since each MOEA performs differently during the search, HH-CO selects them at different search phases. However, they end up having a similar accumulated number of applications.

Finally, we present the boxplot with the average execution

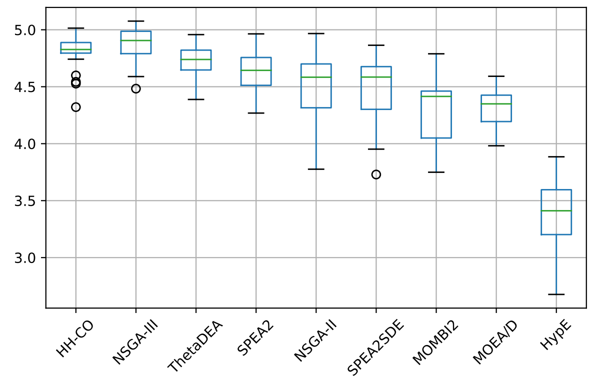

Fig. 3: Boxplot of hypervolume for HH-CO and each MOEA after 21 runs.

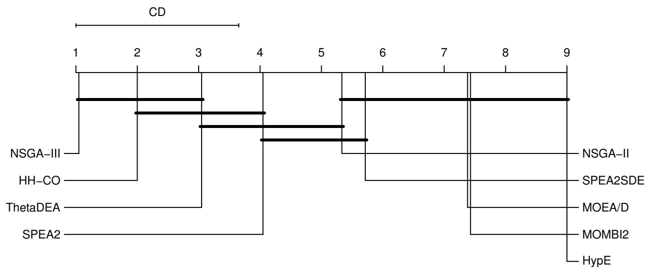

Fig. 4: Critical difference plot of Hypervolume. Each algorithm is connected to its average ranking from 21 trials, and a bold horizontal line connects algorithms without significant statistical difference

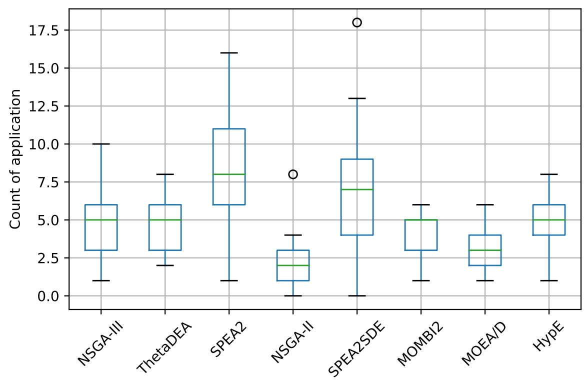

Fig. 5: Boxplot for the count of application of each MOEA for 21 runs.

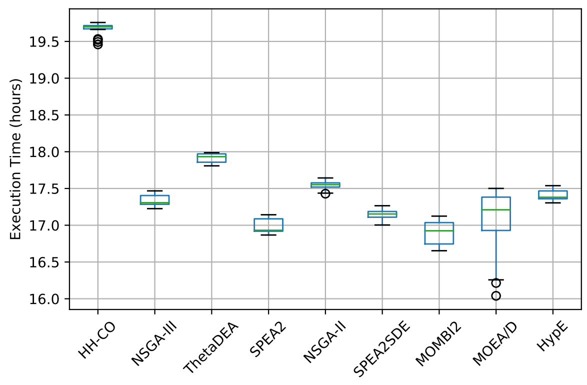

Fig. 6: Boxplot for the computational time spent by each MOEA and HH-CO

time, of 21 trials, for every MOEA and the HH-CO in Fig. 6. The experiments ran on an Intel Xeon CPU E5-2640 v3 of 2.60GHz with 32 CPUs and 94GB of RAM. The jobs were launched by a process queue every time the load average drops below 15. The MOEAs spent 16.5 to 18 hours (17.3 hours in average), being ThetaDEA the slowest. The HH-CO took about 19.5 to 20 hours (about 2.5 hours more than a MOEA in average). It means that, for this problem, running it once is about 14% slower than running an off-the-shelf MOEA. The HH-CO highest cost is the migration step. In this step, it executes the environmental selection method of each MOEA. Therefore, considering this method as the most costly, the HH-CO would take N times more time than a single MOEA (ignoring fitness evaluations), being N the number of MOEAs. In other words, as large the pool of MOEAs, higher the computational cost of HH-CO.

In summary, the HH-CO is slower than an off-the-shelf MOEA. However, it is usually unknown a priori the best MOEA for a problem. If we want to find the best MOEA for this problem, running every MOEA at once, it will take more than five days. Therefore, using HH-CO is faster than running all MOEAs to find the best one. Those conclusions agree with previous works about HH-CO [9]. It is worth noticing that the computational cost of HH-CO can be reduced by decreasing the number of MOEAs in the pool or using communication topologies other than broadcast. Moreover, as much expensive is the fitness evaluation, the smaller is the difference between HH-CO and the MOEAs. Finally, the implementation is the same for the MOEA applied inside the HH or alone; therefore, the difference in computation cost of HH-CO and MOEAs is not related to implementation details.

V. Conclusions

In this research, we have analyzed a recently proposed real-world many-objective problem. This problem presents interesting properties: 32 continuous variables, five objectives, and 22 constraints. It is simple to reproduce since the fitness evaluation module is publicly available. Moreover, it can serve as a baseline for comparison of MOEAs. Based on those characteristics, we solved this problem with a set of eight state-of-the-art MOEAs for many-objective optimization — adapted to incorporate constraint handling.

The MOEAs presented a significant difference in quality measured by the hypervolume indicator. Moreover, it was possible to identify that some MOEAs had a better HV value at the beginning of the search. Others stagnated after some iterations, while others took a while longer to start converging but keep converging during the rest of the search. Finally, the best-performing MOEAs for this problem were NSGA-III and ThetaDEA. Both specifically designed for many-objective optimization. Besides, SPEA2 presented good overall results. Those results motivated the study of a hyper-heuristic on this problem.

Next, we applied a state-of-the-art hyper-heuristic for many-objective optimization, the Cooperative Hyper-heuristic (HH-CO). Its main feature is the cooperation of the low-level heuristics, exchanging information at every generation. The achieved results were competitive to the best MOEA. Further, we evaluated the choices made by HH-CO. In general, the choices agree with the best performing MOEAs in different phases of the search. Finally, we evaluated the computational cost. The hyper-heuristic was slower than an average MOEA but with higher quality in terms of Hypervolume. Compared to the best algorithm, it achieved competitive results. Moreover, the search for the best option needs to run every MOEA at least once, and this will take more than five days.

Future works include the evaluation of the impact of different migration topologies for HH-CO, in both benchmark and real-world problems, considering the quality and the computing time. Another future work is the study of the interactions between MOEA pairs and how to create, online, a communication graph maximizing the quality of the solutions and minimizing the computational cost. Although the competition provided only one many-objective real-world problem, it would be interesting to evaluate HH-CO in other real-world problem instances. Finally, the HH-CO pool could be improved, including some of the latest MOEAs from the literature; also, strategies other than MOEAs, such as

MOPSOs and MOEDAs.

References

[1] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multiobjective Optimization*. London: Springer London, 2005, pp. 105–145. [Online]. Available: https://doi.org/10.1007/1-84628-137-7_6

[2] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Evolutionary Multi-Criterion Optimization*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 280–295.

[3] Cheng, Ran and Li, Miqing and Tian, Ye and Zhang, Xingyi and Yang, Shengxiang and Jin, Yaochu and Yao, Xin, "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, Mar 2017. [Online]. Available: https://doi.org/10.1007/s40747-017-0039-7

[4] Safi, Hayder H. and Ucan, Osman N. and Bayat, Oguz, "On the Real World Applications of Many-objective Evolutionary Algorithms," in *Proceedings of the First International Conference on Data Science, E-learning and Information Systems*, ser. DATA '18. New York, NY, USA: ACM, 2018, pp. 32:1–32:6. [Online]. Available: http://doi.acm.org/10.1145/3279996.3280028

[5] The Japanese Society of Evolutionary Computation, "The 3rd Evolutionary Computation Competition." [Online]. Available: http://www.jpnsec.org/files/competition2019/EC-Symposium-2019-Competition-English.html

[6] H. Jain and K. Deb, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, Aug 2014.

[7] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2017.

[8] John H. Drake and Ahmed Kheiri and Ender Özcan and Edmund K. Burke, "Recent Advances in Selection Hyper-heuristics," *European Journal of Operational Research*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221719306526

[9] Fritsche, Gian and Pozo, Aurora, "Cooperative Based Hyper-heuristic for Many-objective Optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '19. New York, NY, USA: ACM, 2019, pp. 550–558. [Online]. Available: http://doi.acm.org/10.1145/3321707.3321740

[10] Y. Sun and G. G. Yen and Z. Yi, "IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, April 2019.

[11] Hisao Ishibuchi and Noritaka Tsukamoto and Yusuke Nojima, "Evolutionary many-objective optimization: A short review," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, vol. , no. , June 2008, pp. 2419–2426.

[12] K. Deb and H. Jain, "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*, vol. 18, no. 4, pp. 577–601, Aug 2014.

[13] K. Wan and C. He and A. Camacho and K. Shang and R. Cheng and H. Ishibuchi, "A Hybrid Surrogate-Assisted Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, vol. , no. , June 2019, pp. 2018–2025.

[14] Y. Sun and B. Xue and M. Zhang and G. G. Yen, "A New Two-Stage Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 748–761, Oct 2019.

[15] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, Sep. 2015. [Online]. Available: https://doi.org/10.1145/2792984

[16] Fleming, Peter J. and Purshouse, Robin C. and Lygoe, Robert J., "Many-Objective Optimization: An Engineering Design Perspective," in *Evolutionary Multi-Criterion Optimization*, Coello Coello, Carlos A. and Hernández Aguirre, Arturo and Zitzler, Eckart, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 14–32.

[17] J. G. Herrero and A. Berlanga and J. M. M. Lopez, "Effective Evolutionary Algorithms for Many-Specifications Attainment: Application to Air Traffic Control Tracking Filters," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 151–168, Feb 2009.

[18] Sülflow, André and Drechsler, Nicole and Drechsler, Rolf, "Robust Multi-Objective Optimization in High Dimensional Spaces," in *Evolutionary Multi-Criterion Optimization*, Obayashi, Shigeru and Deb, Kalyanmoy and Poloni, Carlo and Hiroyasu, Tomoyuki and Murata, Tadahiko, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 715–726.

[19] Hughes, Evan J., "Radar Waveform Optimisation as a Many-Objective Application Benchmark," in *Evolutionary Multi-Criterion Optimization*, Obayashi, Shigeru and Deb, Kalyanmoy and Poloni, Carlo and Hiroyasu, Tomoyuki and Murata, Tadahiko, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 700–714.

[20] Rodemann, Tobias and Narukawa, Kaname and Fischer, Michael and Awada, Mohammed, "Many-Objective Optimization of a Hybrid Car Controller," in *Applications of Evolutionary Computation*, Mora, Antonio M. and Squillero, Giovanni, Ed. Cham: Springer International Publishing, 2015, pp. 593–603.

[21] A. L. Jaimes and A. Oyama and K. Fujii, "Space trajectory design: Analysis of a real-world many-objective optimization problem," in *2013 IEEE Congress on Evolutionary Computation*, vol. , no. , June 2013, pp. 2809–2816.

[22] Musselman, K. and Talavage, Joseph, "A Tradeoff Cut Approach to Multiple Objective Optimization," *Operations Research*, vol. 28, no. 6, pp. 1424–1435, 1980. [Online]. Available: https://doi.org/10.1287/opre.28.6.1424

[23] Vázquez-Rodríguez, J A and Petrovic, S., "A mixture experiments multi-objective hyper-heuristic," *Journal of the Operational Research Society*, vol. 64, no. 11, pp. 1664–1675, Nov 2013. [Online]. Available: https://doi.org/10.1057/jors.2012.125

[24] Vrugt, Jasper A. and Robinson, Bruce A., "Improved evolutionary optimization from genetically adaptive multimethod search," *Proceedings of the National Academy of Sciences*, vol. 104, no. 3, pp. 708–711, 2007. [Online]. Available: https://www.pnas.org/content/104/3/708

[25] de Carvalho, Vinicius Renan and Sichman, Jaime Simão, "Applying Copeland Voting to Design an Agent-Based Hyper-Heuristic," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '17. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 972–980. [Online]. Available: http://dl.acm.org/citation.cfm?id=3091125.3091263

[26] Mashael Maashi and Graham Kendall and Ender Özcan, "Choice function based hyper-heuristics for multi-objective optimization," *Applied Soft Computing*, vol. 28, pp. 312 – 326, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494614006449

[27] W. Li and E. Özcan and R. John, "A Learning Automata-Based Multiobjective Hyper-Heuristic," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 59–73, Feb 2019.

[28] D. Brockhoff, T. Wagner, and H. Trautmann, "On the properties of the r2 indicator," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 465–472. [Online]. Available: https://doi.org/10.1145/2330163.2330230

[29] H. Wickham and L. Stryjewski, "40 years of boxplots," had.co.nz, Tech. Rep., 2012.

[30] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, p. 1–30, Dec. 2006.