2023 Design, Automation & Test in Europe Conference (DATE 2023)

Maximizing Computing Accuracy on Resource-Constrained Architectures

Van-Phu Ha
Univ Rennes, Inria, IRISA
Rennes, France
van-phu.ha@inria.fr

Olivier Sentieys
Univ Rennes, Inria, IRISA
Rennes, France
olivier.sentieys@inria.fr

***Abstract*—With the growing complexity of applications, designers need to fit more and more computing kernels into a limited energy or area budget. Therefore, improving the quality of results of applications in electronic devices with a constraint on its cost is becoming a critical problem. Word Length Optimization (WLO) is the process of determining bit-width for variables or operations represented using fixed-point arithmetic to trade-off between quality and cost. State-of-the-art approaches mainly solve WLO given a quality (accuracy) constraint. In this paper, we first show that existing WLO procedures are not adapted to solve the problem of optimizing accuracy given a cost constraint. It is then interesting and challenging to propose new methods to solve this problem. Then, we propose a Bayesian optimization based algorithm to maximize the quality of computations under a cost constraint (i.e., energy in this paper). Experimental results indicate that our approach outperforms conventional WLO approaches by improving the quality of the solutions by more than 170%.**

I. Introduction

The rapid development in scientific and technological innovations during the last decade has driven the demand for more powerful chips with higher performance to handle complex tasks in various fields, such as artificial intelligence (AI), big data, and the Internet of Things (IoT). The requirement for integrating more applications and services, while ensuring at the same time high performance and quality of service (QoS) in a limited cost budget (energy and/or area), poses new challenges for energy-efficient computing. As an example, Deep Neural Networks (DNN) are used in multiple applications such as object detection, speech recognition, navigation, etc. Many of these applications operate on battery-powered intelligent embedded systems such as autonomous drones, self-driving cars, smart phones, and wearable devices. Thus, increasing the performance and/or QoS under a limited energy budget is a key competitive advantage among electronic device manufacturers.

Likewise, implementing applications on ASIC or FPGA devices also faces challenges of optimizing the performance and/or QoS within a constrained cost budget. In theses cases, besides energy, relevant cost constraints include either limited area inside a System on Chip (SoC) or limited resources like the number of LUTs, DSP blocks, memory size in the context of FPGA architectures. As an example, in the development life cycle of some SoC products, upgrading only some modules while keeping the remaining ones untouched helps shortening the time to market. Replaced modules usually have new features that enhance the performance of the system. This requires optimization in terms of maintaining high implementation performance and/or QoS within the available silicon area. This process takes more effort to optimize because the new modules are usually more sophisticated and complex than older ones, requiring more time and effort to create, test, and debug them to fit into the limited, unchanged area (or a given energy budget). Therefore, a design technique that optimizes for the performance and/or QoS within a constrained cost budget of the system would be of great concern, but is not really covered in the state of the art.

Approximate Computing (AC) emerged as an effective solution to adjust the trade-off between QoS and cost constraints, while satisfying design requirements. The use of fixed-point arithmetic in embedded systems, combined with a precision reduction, is one of the AC techniques for reducing cost and energy. This technique requires to determine the optimal fixed-point word-lengths (i.e., bit-widths) for representing all variables of the application that still fulfil some quality (i.e., accuracy of computations) requirement. This procedure is called Word-Length Optimization (WLO). Many methods are proposed to solve the WLO problem with QoS constraints quickly and efficiently [1]–[3]. However, these methods traditionally explore the design space to minimize a cost function (e.g., area or energy) under a given accuracy constraint at the output of the system.

However, as mentioned previously, it might be of great interest to limit the area or energy cost budget of a computing kernel below a specified bound. In this case, the stringent constraint becomes the cost, and it is important to maximize accuracy (or the minimize accuracy degradation) as a quality metric under this cost constraint. In this paper, we propose new methods to solve this problem of maximizing QoS within a given cost budget. We first highlight the limitations of traditional methods to solve WLO under cost constraints. Then, we propose a method relying on Bayesian optimization to address the resource-constrained WLO problem. Experimental results show that our proposed algorithm outperforms the latest conventional approaches, improving solution quality by up to 300% compared to Uniform Word-Length Optimization (UWLO) and by up to 170% compared to a state-of-the-art classical WLO approach.

The remaining sections are organized as follows. In Section II, we provide background information and discuss related work, followed by an explanation of our motivations and problem statement in Section III. In Section IV, we describe

978-3-9819263-7-8/DATE23/© 2023 EDAA

our proposed Bayesian optimization-based method. In Section V, we compare the performance of our approach with traditional approaches on various benchmarks, before to conclude in Section VI.

II. Related Work

In this section, the well-studied accuracy-constrained Word-Length Optimization (WLO) problem is presented, followed by the related state of the art. Then, we motivate the need for a new method to solve a cost-constrained WLO problem.

A number expressed using fixed-point arithmetic has both integer and fractional word-lengths (WLs), which are correspondingly represented by I and F bits, respectively. The dynamic range of the number represented with this format is covered by the integer WL, while its accuracy is controlled by the fractional WL. In this paper, we concentrate on the WLO for the fractional WL since it is the most time-consuming exploration in the design process. The dynamic range is usually evaluated through static analysis or simulation to determine the integer WL to guarantee no or low-probability overflow during computations.

Let the vector $\boldsymbol{W} = [W_0, W_1, \ldots, W_{N-1}]$ denote a word-length configuration with N effective variables from the considered application to be explored for fixed-point conversion. The main objective of WLO is to determine a good-enough word-length configuration that minimizes a cost function under a quality (accuracy) constraint:

$$\min \boldsymbol{C}(\boldsymbol{W}) \quad \text{Subject to} \quad \boldsymbol{\lambda}(\boldsymbol{W}) \geq \boldsymbol{\lambda_{obj}}, \tag{1}$$

where $\boldsymbol{C}$ and $\boldsymbol{\lambda}$ are the cost and accuracy functions depending on $\boldsymbol{W}$, respectively. $\boldsymbol{\lambda_{obj}}$ is the maximal degradation of the accuracy at a given application output that is acceptable for the required quality of service. Analytical models and simulation-based methodologies are both used to construct $\boldsymbol{C}$ and $\boldsymbol{\lambda}$.

There are two approaches to solve this well-studied accuracy-constrained WLO problem: simulation-based and analytical approaches. The aim of analytical approaches is to approximate quality and cost functions of the WLO problem to be convex functions which can be solved quickly by some convex optimization algorithms, e.g., CVX [4]. Existing techniques are limited to modeling noise power metrics of Linear and Time-Invariant (LTI) systems (with some extensions) [1], [5], [6]. Simulation-based approaches [7]–[11] use simulations and iterative search to address WLO. Uniform WordLength Optimization (UWLO) can quickly evaluates several solutions which are constructed by the same wordlength for each variable to obtain the best one. However, the quality of the solution obtained is much worse than the ones obtained by the non-uniform wordlength optimization methods [12]. Some more advanced methods leverage noise budgeting techniques [2], [13] to reduce the exploration time in solving WLO for large applications.

Bayesian Optimization (BO) is a machine-learning-based optimization technique [14] designed to optimize functions that often lack mathematical expressions and/or derivatives. BO typically consists of two essential components: i) a probabilistic surrogate model for modeling the unknown objective function based on previously observed samples and ii) an acquisition function that optimizes over the surrogate model to recommend further samples. The procedure of choosing surrogate models is a key distinction between BO approaches and those of others. While Gaussian Processes (GP) are often employed for moderate-size continuous-domain issues, tree-based models such as Random Forest and Tree-structured Parzen Estimator (TPE) are advantageous for large-size discrete-domain problems [15]. TPE identifies points that might have been drawn based on the assessment of a loss function at other points. BO was first used to speed up WLO in a hybrid method [3].

III. Resource-Constrained WLO

All the previously mentioned techniques solve the Word-Length Optimization problem without any constraint on the cost of the solution. The objective of the optimizer is to find a solution with minimum cost that guarantees the accuracy being higher than a given constraint. However, as motivated in the introduction, with the growing complexity of applications, it might be of great interest to limit the area or energy cost budget of a computing kernel under a restricted bound. In this case, the stringent constraint becomes the cost and it is important to find the maximal accuracy (or the minimum accuracy degradation) under this cost constraint.

The resource-constrained WLO problem can be stated as

$$\max \boldsymbol{\lambda}(\boldsymbol{W}) \quad \text{Subject to} \quad \boldsymbol{C}(\boldsymbol{W}) \leq \boldsymbol{C}_{budget}, \tag{2}$$

where $\boldsymbol{C}$ and $\boldsymbol{\lambda}$ represent for the cost and accuracy functions of a WL configuration vector $\boldsymbol{W}$. The objective of this problem is to find a WL configuration $\boldsymbol{W}_o$ that maximizes the accuracy at the application output given $\boldsymbol{C}_{budget}$ as a cost constraint. The WL components W_i of $\boldsymbol{W}$ can be bounded as

$$m_i \leq W_i \leq n_i \qquad m_i, n_i \text{ and } W_i \in \mathbb{Z}^+, \tag{3}$$

to limit the exploration space.

A. Room for Accuracy Improvement Given a Cost Budget

We use a subset of the solution space of a 5-stage 33-tap Finite Impulse Response (FIR) filter as an example to motivate our work. The filter has 17 different variables represented in fixed-point arithmetic. Fig 1 shows 10^5 solutions obtained from a random search. Each solution evaluates the FIR filter output accuracy (PSNR) and the associated cost (energy in nJ) for a random WL configuration. The integer WL is set to 12 bits for all variables to avoid overflow and the fractional WL of each variable is varied in the range of 3 to 20 bits, which can be considered large enough to reduce the chance of missing good solutions. The energy is evaluated based on a library of arithmetic operators characterized for various WL in a 28nm FDSOI technology (see Sec.V-A for more details). Given an energy level, there is a large set of solutions consuming the same amount of energy but resulting in a significantly different quality at the application output. For instance, with 0.0075 nJ considered as a maximum energy budget, the quality of satisfying solutions can vary from -20dB to 60dB. Hence, to maximize the performance or quality of the system given a cost budget, it is worth to design an optimization method that can obtain the highest quality solution satisfying the cost constraint from possible solutions.

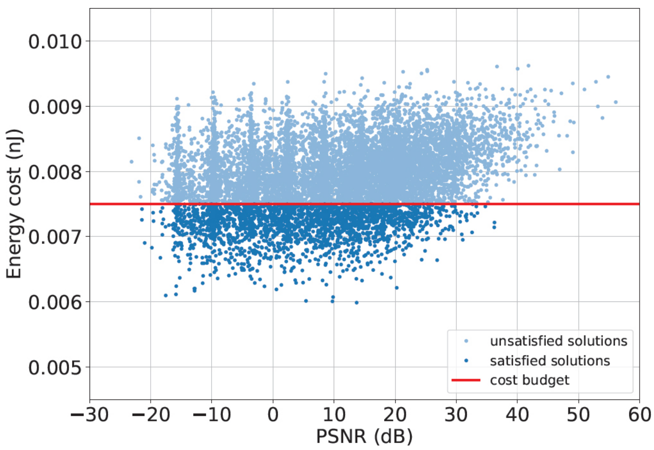

Fig. 1: 10^5 random solutions as a representative subset of all possible solutions, for the FIR filter benchmark. The red line represents an energy budget of 0.0075 nJ

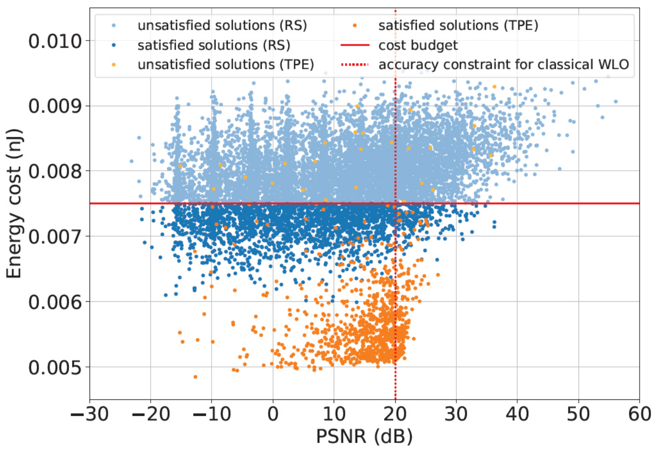

Fig. 2: The classical WLO problem in the context of the cost-constrained WLO problem. The data points includes the 10000 randomized solutions and 1000 solutions found by TPE for the classical WLO problem with an accuracy constraint as 25 dB of PSNR.

B. Resource-Constrained WLO as a New Problem

The classical accuracy-constrained WLO focuses on minimizing cost as much as possible while satisfying QoS at the system output. Meanwhile, the objective of the resource-constrained WLO is to exploit the available resource (energy and/or area) of the system to enhance its overall performance. The objective of these two problems are totally different, leading to the different exploration targets. For the design purpose that enhances quality on a system with specific resources as much as possible, the search objective of the original problem is no longer relevant. Hence, it is important to design new methods to solve the resource-constrained WLO problem. We illustrate the objective target of the classical WLO for FIR filter in the context of a system with an energy budget. We use TPE as a search method for the classical problem with a PSNR quality target of 25 dB. The use of TPE follows the setup in [3].

In Fig. 2, the explored solutions discovered by TPE for

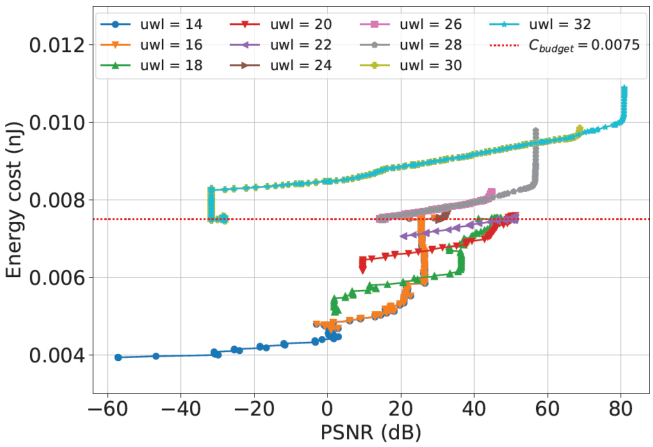

Fig. 3: Different exploration trajectories of Tabu search method corresponding to different starting points for the resource-constrained WLO problem on the FIR application. The energy budget is 0.0075 nJ. Each color corresponds to a search procedure with a given starting point. Fixed-point variables of each procedure are chosen with uniform word-lengths (UWLs) at the starting point. The number in the legend indicates the word-length given for the UWL. For each procedure, the starting point is further from the cost budget line than the ending point. Some procedures overlapped during the exploration process.

the classical problem are compared to those discovered by Random Search (RS) in the context of a system with an energy budget as 0.0075 nJ. Clearly, TPE focuses on exploring low-cost solutions around the accuracy constraint, i.e. around the vertical line 20 dB, missing possible solutions with higher accuracy and still satisfying the energy budget, i.e., solutions found by RS range in 30 to 40 dB of PSNR. Similar to TPE, recent simulation-based techniques mentioned in [3], [11] such as Min+1 and Tabu Search also rely on a search model to restrict the exploration space, followed by a fine tuning around the quality constraint to find a better solution.

C. Limitations of Classical WLO Methods

Numerous methods to address the WLO problem have been proposed in the literature. In [3], [16], the performance and execution time of several state-of-the-art techniques, including Min+1, Max-1, Heuristic approach, Tabu Search, and GRASP, have been compared. The comparison shows that variants of bi-directional searches that combine steepest-descent and mildest-ascent procedures, such as heuristic approach [17] and Tabu search [11], outperform the mono-directional searches. However, due to the greedy nature of those algorithms, the final solution is not guaranteed to be a global minima and is dependent on the starting point. Additionally, most of these searches are based on local iterative searches that travel across small distances in the discrete domain. As a result, if the starting point is distant from a local minimum, convergence will be slow [3].

Fig. 3 illustrates the limitations of the classical methods. We choose Tabu search as a typical method to address the resource-constrained WLO. The FIR filter with a maximum energy supply as 0.0075 nJ is still used for the illustration. The search procedure is initialized at different points. Each initial point is constituted from fixed-point variables with UWL. The integer word-length of variables is fixed as 12 bits. The fractional word-length is incremented every two bits from 2 bits to 20 bits, thus the UWL varies from 14 bits to 32 bits for each starting point. The quality of obtained solutions with Tabu search and different starting points significantly differs and strongly depends on the initialization. At starting points with UWL equal to 30 or 32, the obtained solutions have a bad PSNR of around -30 dB. At the remaining initial points, Tabu search produced better solutions than RS from 10 to 50 dB of PSNR. However, the solution quality is varying dramatically. Besides, procedures that start farther away from the cost constraint tend to converge more slowly than those that start closer to it. The trends are consistent with other benchmarks evaluated in Section V.

IV. Solving Resource-Constrained WLO

In this section, we introduce a BO-based method to address the resource-constraint WLO. The loss function is constructed first, followed by the description of our BO-based approach.

A. Loss Function

The resource-constrained WLO problem statement of Eq. 2 is changed as follows using the Lagrange multipliers approach to turn a constrained problem into an unconstrained one:

$$f(\boldsymbol{W}) = -\boldsymbol{\lambda}(\boldsymbol{W}) + \alpha(\boldsymbol{C}(\boldsymbol{W}) - \boldsymbol{C}_{budget}), \tag{4}$$

with a positive and big enough α. By minimizing the loss function, TPE tends to sample more frequently solutions that are of high quality and satisfy the cost constraint. The factor α is important for exploration. If α is very small, TPE will ignore the restricted cost condition and concentrate on exploring solutions with quality as high as possible. If α is very large, the solution of TPE would be highly dependent on the restricted cost condition. That is, TPE focuses on finding solutions that cost less than $\boldsymbol{C}_{budget}$ while neglecting solutions of high quality.

Based on many WLO experiments, we found that the cost and quality functions tend to be proportional to $\boldsymbol{W}$. A high quality solution usually comes at a high cost. Thus, the best solutions to satisfy the cost constraint with high quality are likely to be those around the cost budget $\boldsymbol{C}_{budget}$. Therefore, for faster convergence, we force the loss function to cover only a narrow range $[c_l, c_h]$ around $\boldsymbol{C}_{budget}$. Solutions with costs outside this range of interest are penalized with the positive infinite loss value. The loss function is then defined as

$$f(\boldsymbol{W}) = \begin{cases} -\boldsymbol{\lambda}(\boldsymbol{W}) + \alpha(\boldsymbol{C}(\boldsymbol{W}) - \boldsymbol{C}_{budget}) & \text{if } \boldsymbol{C}(\boldsymbol{W}) \in [c_l, c_h], \\ +\infty & \text{otherwise.} \end{cases} \tag{5}$$

B. TPE Algorithm

For the resource-constrained WLO problem of Eq. 2, a WL configuration formed by the choice of each variable, i.e., $W_i \in [m_i, n_i]$ represents a solution. All possible combination of different values of W_i creates a solution space of this WLO problem. With the leverage of Bayesian Optimization relying on TPE [18] to solve the WLO problem, each W_i is considered as a hyper-parameter to be tuned. Each W_i is initially mapped to a prior of quantized uniform distribution in which the value is sampled uniformly in $[m_i, n_i]$ and then rounded to the nearest integer value.

Algorithm 1 describes the TPE algorithm for the resource-constrained WLO problem. The algorithm is first initialized by uniform word-length (UWL) configurations and corresponding loss values evaluated by the loss function. These configurations are then served as first samples in an observation history $\boldsymbol{\mathcal{H}}$. TPE works as an iterative approach. At each iteration, a WL configuration $\boldsymbol{W}_i$ is evaluated by the loss function $f(\boldsymbol{W}_i)$. Then, obtained samples $\{\boldsymbol{W}_i, f(\boldsymbol{W}_i)\}$ stored in the observation history $\boldsymbol{\mathcal{H}}$ are divided into two groups. The first group contains *good* samples where the loss f_i is less than a threshold γ^*, whereas the second group consists of the remaining, considered as *bad* samples. TPE uses these two sample groups to model two likelihood probability density functions $l(\boldsymbol{W}) = p(\boldsymbol{W}|f(\boldsymbol{W}) < \gamma^*)$ and $g(\boldsymbol{W}) = p(\boldsymbol{W}|f(\boldsymbol{W}) \geq \gamma^*)$, respectively. Then, the TPE algorithm decides which hyper-parameter to try in the next iteration by maximizing the ratio

$$\frac{l(\boldsymbol{W})}{g(\boldsymbol{W})} = \frac{p(\boldsymbol{W}|f(\boldsymbol{W}) < \gamma^*)}{p(\boldsymbol{W}|f(\boldsymbol{W}) \geq \gamma^*)}.$$

Algorithm 1 TPE algorithm

$\boldsymbol{\mathcal{H}} \leftarrow$ Uniform Word-Length Initialization
for $i \in [1, \ldots, T]$ **do**
  $\boldsymbol{W}^* = argmax \frac{l(\boldsymbol{W})}{g(\boldsymbol{W})}$
  Evaluate $f(\boldsymbol{W}^*)$
  $\boldsymbol{\mathcal{H}} \leftarrow \boldsymbol{\mathcal{H}} \cup (\boldsymbol{W}^*, f(\boldsymbol{W}^*))$
  Update $l(\boldsymbol{W})$ and $g(\boldsymbol{W})$ given $\boldsymbol{\mathcal{H}}$
end for

Our initialization procedure constructed from UWL configurations samples the solution space at some design points ranging from low-accuracy-and-low-cost solutions to high-accuracy-and-high-cost solutions. These design points are then used to construct density functions at the very first iteration of the algorithm. Under the evaluation of the loss function, TPE can start exploring from the points that yield low loss values. This initialization provides better results and faster convergence than an initialization constructed from random configurations.

V. Evaluation

A. Experiment setup

We choose Uniform Word-Length Optimization (UWLO) as our baseline. Our goal is to evaluate how much our approach can perform better than UWLO, a fast and simple method considering all variables having the same WL. In our approach, we employed Adaptive TPE (ATPE), an improved extension of TPE that was implemented on Hyperopt [18] to update the hyper-parameters of TPE in real time. The cost and quality functions, $\boldsymbol{\lambda}(\boldsymbol{W})$ and $\boldsymbol{C}(\boldsymbol{W})$, in the loss function are normalized within a range. First, UWLO is performed to obtain the quality and the cost of uniform-wordlength solutions. $\boldsymbol{\lambda}(\boldsymbol{W})$ is normalized between upper and lower bounds. The lower bound is the solution which satisfies the cost constraint and has the highest quality. The upper bound is the highest quality value obtained from those solutions without the cost constraint. The normalization range of $\boldsymbol{C}(\boldsymbol{W})$ is $[c_l, c_h]$, where c_l and c_h are separated from the central value, $\boldsymbol{C}_{budget}$, by a distance of 10% of the difference between the minimum and the maximum cost values obtained from the UWL solutions. We conduct experiments on our benchmarks with five values of α, i.e., $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5\}$, to evaluate its impact to the approach. The number of iterations for each experiment is chosen as 5000 which can cover mostly the possibility of improving the solution quality as much as possible. Note that more complex applications would need more iterations to increase the possibility of obtaining better solutions. To further demonstrate the limitations of classical methods, we conduct several experiments with Tabu search [11] on our benchmarks. In each benchmark, Tabu Search is initialized at different uniform wordlengths.

All experiments were conducted on a Linux-powered Intel Xeon E5640 processor at 2.67GHz with 4GB of RAM. Energy is the basis for our cost model. The energy model counts the operations carried out by each operator and determines the overall energy cost using the empirically determined energy consumption of an operation from several synthesis for various WLs. An operator is characterized by the WLs of the operands, the WL of the result, and the arithmetic operation performed. A 28nm FDSOI technology is used for characterization along with Synopsys Design Compiler and Prime Time.

Three applications are used as the benchmarks to evaluate our approach: FIR, IIR and NLM. The FIR filter, also used in previous sections, is implemented with a 5-stage cascaded structure of 33 taps each. The IIR filter has a 5-stage cascaded structure of 2^{nd}-order filters. The number of effective variables to optimize is 17 for FIR and 33 for IIR. Non-local means (NLM) [19] is a denoising approach in image processing. To denoise the target pixel, the algorithm calculates the mean of all pixels in the picture, weighted by their similarity to the target pixel. The implementation of NLM uses 19 effective variables. We explore only the fractional word-lengths. The integer word-length of each signal in the applications is determined beforehand so that the dynamic range of each signal is covered. The integer word-length of FIR, IIR and NLM are set to 12, 12 and 6, respectively. The word-lengths of the variables used in FIR and IIR range from 13 to 32 bits, whereas NLM is explored from 8 to 32 bits.

We use Peak Signal to Noise Ration (PSNR) for filters and Structural Similarity Index Measure (SSIM) for NLM as quality metrics. We choose three cost budget objectives for every application as shown in Table I, thus resulting in nine different benchmarks. The targets are selected proportionally to the operation counts of each application, i.e., the number of additions and multiplications. The cost budgets of NLM are higher than the remaining applications due to its higher computation complexity.

B. Performance Evaluation

Figure 4 presents a comparison between our approach and UWLO serving as a baseline. In overall, our approach outperforms UWLO by obtaining solutions with better quality for a given cost budget. The results show that solutions obtained by our approach with different values of α are always better than those obtained by UWLO. Table II summarizes the quality improvement of our solutions compared to those obtained by UWLO. In several cases, including FIR (Target 1, 2 and 3), IIR (Target 1 and 2), and NLM (Target 1), our approach significantly improves the quality of the solutions. This indicates that a solution can be obtained quickly by UWLO, but which is far from the optimal; this also confirms the conclusion in [12].

TABLE I: The benchmarks for the evaluation

Application	Target 1 (nJ)	Target 2 (nJ)	Target 3 (nJ)
FIR	0.0065	0.0070	0.0075
IIR	0.0010	0.0012	0.0014
NLM	250	300	400

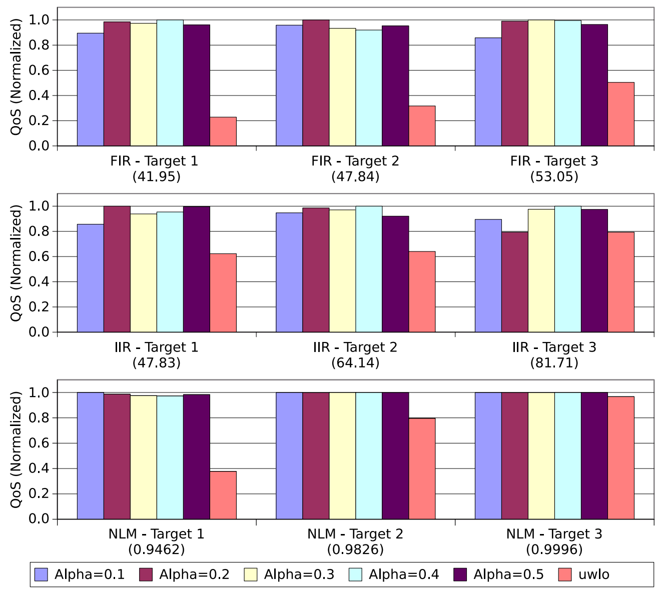

Fig. 4: Performance comparison between our approach and UWLO in terms of quality of results for different benchmarks. The results are normalized with the highest quality value indicated in parentheses for each benchmark.

TABLE II: Quality improvement of our solutions in percentage compared to UWLO solutions. Given value for each benchmark is the average of those obtained by different values of α.

Application	Target 1	Target 2	Target 3
FIR	321.19%	200.48%	90.68%
IIR	52.39%	50.67%	16.91%
NLM	160.50%	25.63%	3.32%

For some benchmarks, the quality of solutions obtained by explorations with small α, such as $\alpha = \{0.1, 0.2\}$, is worse than for higher α values. As explained in Section IV-A, the cost constraint will be not respected with small α. This means that the exploration strategy will focus more on improving the solution quality without being strictly constrained by the cost budget, which causes an excessive search in the infeasible

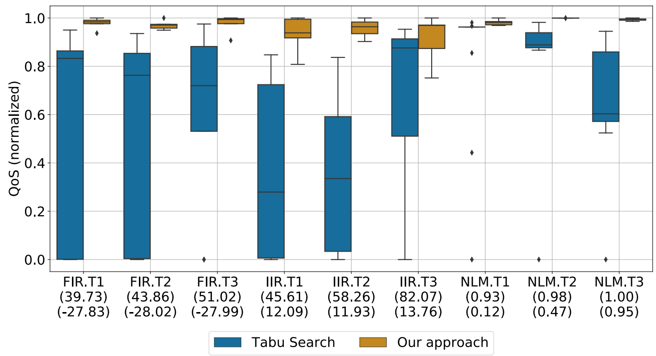

Fig. 5: Comparison of the normalized quality of solutions obtained by Tabu Search and our approach for different benchmarks. The solutions obtained by Tabu search are the best found solutions under the target cost constraint, but for different starting points. Our solutions are obtained with different α. The normalization range are mentioned in parentheses, where the numbers above and below represent for upper and lower bounds, respectively. Letter "T" is the abbreviation of "Target".

TABLE III: Average quality improvement of the solutions provided by our approach over those obtained by Tabu Search (TS). The results are normalized with the range in Figure 5.

Benchmark - Target	TS	Our approach	Avg. Impr.
FIR - Target 1	0.5258	0.9776	85.93%
FIR - Target 2	0.5017	0.9704	93.40%
FIR - Target 3	0.6116	0.9749	59.40%
IIR - Target 1	0.3599	0.9318	158.93%
IIR - Target 2	0.3499	0.9569	173.46%
IIR - Target 3	0.6717	0.9130	35.94%
NLM - Target 1	0.8058	0.9814	21.79%
NLM - Target 2	0.8288	0.9996	20.61%
NLM - Target 3	0.6388	0.9935	55.52%

region. As a result, this will reduce the possibility of improving the quality of solutions in the feasible region.

Figure 5 illustrates a comparison of the quality of solutions found by our proposed method and Tabu Search. The quality of results found using Tabu Search varies a lots among benchmarks, the solution strongly depending on its starting point. In the meanwhile, the quality of solutions obtained via our method is much more stable. Figure 5 also illustrates that the average quality obtained by our method is always superior than that of Tabu search. Table III reports the average improvements of the quality of solutions provided by our method over Tabu search. The table shows that the solutions can be improved from 20% to 173% using our optimized BO-based algorithm to solve the resource-constrained WLO problem. Especially, for IIR with Target 1 and Target 2, our method outperforms Tabu by up to 158% and 173%, respectively.

VI. Conclusion

In this paper, we present our Bayesian-Optimization-based approach to maximize the quality of digital applications implemented under a resource-constrained budget. We first show the importance of the resource-constrained word-length optimization (WLO) problem in systems which have limited silicon area and energy supply. Then, we highlight the limitations of classical approaches for WLO, such as Tabu search, which are not well adapted to solve the resource-constrained WLO problem. We thus propose a method based on BO and the TPE algorithm with an adaptive loss function. The results from our experiments show that our approach significantly outperforms UWLO and Tabu search, a state-of-the-art method. Our method is the first efficient approach addressing the resource-constrained WLO problem and also the starting point for further research that will address other aspects on the scalability and optimality of this problem, as well as other cost constraints, such as limited area or maximum number of resources.

References

[1] K. N. Parashar, D. Menard, and O. Sentieys, "A polynomial time algorithm for solving the word-length optimization problem," in *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, pp. 638–645, 2013.

[2] D. Novo, I. Tzimi, U. Ahmad, P. Ienne, and F. Catthoor, "Cracking the complexity of fixed-point refinement in complex wireless systems," in *IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 18–23, 2013.

[3] V.-P. Ha and O. Sentieys, "Leveraging bayesian optimization to speed up automatic precision tuning," in *24th IEEE/ACM Design, Automation and Test in Europe (DATE)*, 2021.

[4] M. Grant, S. Boyd, and Y. Ye, "cvx users' guide," *online: http://www. stanford. edu/˜ boyd/software. html*, 2009.

[5] S.-C. Chan and K. M. Tsui, "Wordlength optimization of linear time-invariant systems with multiple outputs using geometric programming," *IEEE Trans. on Circ. and Syst.*, vol. 54, no. 4, pp. 845–854, 2007.

[6] P. D. Fiore, "Efficient approximate wordlength optimization," *IEEE Trans. on Computers*, vol. 57, no. 11, pp. 1561–1570, 2008.

[7] D.-U. Lee, A. A. Gaffar, R. C. Cheung, O. Mencer, W. Luk, and G. A. Constantinides, "Accuracy-guaranteed bit-width optimization," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 25, no. 10, pp. 1990–2000, 2006.

[8] M.-A. Cantin, Y. Savaria, D. Prodanos, and P. Lavoie, "An automatic word length determination method," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, vol. 5, pp. 53–56, 2001.

[9] K. Han, I. Eo, K. Kim, and H. Cho, "Numerical word-length optimization for cdma demodulator," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, vol. 4, pp. 290–293, 2001.

[10] G. A. Constantinides, P. Y. Cheung, and W. Luk, "Wordlength optimization for linear digital signal processing," *IEEE Trans. on CAD of Int. Circ. and Syst.*, vol. 22, no. 10, pp. 1432–1442, 2003.

[11] H.-N. Nguyen, D. Ménard, and O. Sentieys, "Novel algorithms for word-length optimization," in *19th European Signal Processing Conf.*, pp. 1944–1948, IEEE, 2011.

[12] H. Choi and W. Burleson, "Search-based wordlength optimization for vlsi/dsp synthesis," in *IEEE Work. on VLSI Signal Processing VII*, pp. 198–207, 1994.

[13] V.-P. Ha, T. Yuki, and O. Sentieys, "Towards generic and scalable word-length optimization," in *DATE 2020-23rd IEEE/ACM Design, Automation and Test in Europe*, pp. 1–6, IEEE, 2020.

[14] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[15] K. Eggensperger, M. Feurer, F. Hutter, J. Bergstra, J. Snoek, H. Hoos, and K. Leyton-Brown, "Towards an empirical foundation for assessing bayesian optimization of hyperparameters," in *NIPS workshop on Bayesian Optimization in Theory and Practice*, vol. 10, p. 3, 2013.

[16] M.-A. Cantin, Y. Savaria, and P. Lavoie, "A comparison of automatic word length optimization procedures," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, vol. 2, pp. II–II, 2002.

[17] W. Sung and K.-I. Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems," *IEEE transactions on Signal Processing*, vol. 43, no. 12, pp. 3087–3090, 1995.

[18] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, pp. 2546–2554, 2011.

[19] A. Buades, B. Coll, and J.-M. Morel, "Non-local means denoising," *Image Processing On Line*, vol. 1, pp. 208–212, 2011.