Toward Efficient Programming of Reconfigurable Radio Frequency (RF) Receivers

Jun Tao, Ying-Chih Wang, Minhee Jun, Xin Li, Rohit Negi, Tamal Mukherjee and Lawrence T. Pileggi
Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213
{juntao, yingchiw, mjun, xinli, negi, tamal, pileggi}@andrew.cmu.edu

Abstract – Reconfigurable radio frequency (RF) system is an emerging component to mitigate the growing engineering cost for wireless chip design. In this paper, we propose a new methodology for efficient programming of reconfigurable RF receiver. The proposed method is facilitated by two novel techniques: two-phase relaxation search and Pareto-based search space reduction. Our numerical experiments demonstrate that the proposed methodology is more robust (i.e., close to global optimum) and/or efficient (i.e., with low computational cost) than other traditional algorithms based on either local relaxation or simulated annealing.

I. Introduction

For most commercial wireless communication applications, rapid introduction of new wireless standards, increased autonomous applications and aggressive miniaturization result in a significant growth of engineering cost for developing various dedicated wireless chips. The advent of reprogrammable RF electronic systems provides a promising avenue for cost reduction by allowing circuit reuse via post-manufacturing reprogrammability [1].

Generally speaking, a reconfigurable RF system consists of a number of circuit blocks (e.g., low noise amplifier (LNA), mixer, filter, etc.) that can be adaptively reconfigured in terms of their block-level performance metrics (e.g., gain, bandwidth, nonlinearity, noise figure (NF), etc.) in order to meet a set of given performance specifications at system level (e.g., signal-to-noise ratio (SNR), bit error rate, etc.). Recently, several reconfigurable RF systems have been developed, including reconfigurable vector signal analyzer and software-defined receiver [2], multi-standard RF front-end [3] and ultra-low-power RF transmitter [4]. For a reconfigurable RF system, all circuit blocks must be carefully configured to ensure their desired functionality and performance. The process of finding the optimal configurations for all circuit blocks to meet the given system-level specifications is referred to as RF system *programming* in this paper. Our objective is to develop efficient algorithms to accomplish this programming task.

It is important to mention that most traditional algorithms developed for general-purpose analog optimization are not directly applicable to RF receiver programming. Traditionally, analog optimization often takes a fixed circuit topology and optimizes the device-level parameters, such as the widths and lengths of transistors, to meet a set of given performance specifications [5]. These traditional methods fall into two broad categories [6]: equation-based [7] and simulation-based [8]-[9]. Equation-based methods, such as geometric programming [7], rely on design equations that may require a lot of efforts to derive. Furthermore, these design equations may not be sufficiently accurate since various approximations are often made [5]. On the other hand, simulation-based methods, such as simulated annealing [8] and genetic programming [9], often take excessive computational time to extensively search the design space in hopes of reaching a solution close to global optimum [5]. For these reasons, there is a strong need to develop a new optimization technique that can solve our proposed problem of RF receiver programming both robustly (i.e., close to global optimum) and efficiently (i.e., with low computational cost).

Towards this goal, two novel techniques are proposed in this paper by exploiting the unique characteristics of RF receiver programming.

- *Two-phase relaxation search*: As will be discussed in Section II, RF receiver programming can be cast into an optimization problem that minimizes a cost function subject to a single constraint. Given this unique problem formulation, we propose to adopt a two-phase relaxation search algorithm. During the first phase, the constraint function is maximized without taking into account the cost function. Next, the cost function is minimized subject to the given constraint during the second phase. Such a two-phase approach was initially developed to explore the performance trade-offs between power and delay for gate sizing of digital circuits [10]. It has been demonstrated to be extremely robust (i.e., avoiding a large number of local optima) in the literature.
- *Pareto-driven search space reduction*: While most traditional algorithms were developed to optimize analog circuit blocks by tuning the device-level parameters, RF receiver programming should be performed at system level based on reconfigurable circuit blocks. In this case, we can take advantage of block-level Pareto optimal fronts (POFs) [11] to substantially reduce the search space. The POF of a circuit block captures the optimal trade-offs of the block-level performance metrics, such as the trade-off between nonlinearity and power of an LNA. For RF receiver programming, the block-level configurations on POFs are most likely to yield the optimal system-level performance. Therefore, it is desirable to search the optimal configurations among these candidates on POFs only for RF receiver programming.

The remainder of this paper is organized as follows. In Section II, we first describe the problem formulation of RF receiver programming and then develop the proposed programming algorithm in Section III. The efficiency of our proposed technique is demonstrated by a reconfigurable RF receiver example in Section IV. Finally, we draw conclusions in Section V.

II. Problem Formulation

The general goal of reconfigurable RF system design is to enable a common hardware architecture that reuses a set of circuit blocks for disparate wireless applications and standards through programming. As a result, the design cost can be reduced dramatically compared to the traditional non-configurable system. A reconfigurable RF system should be able to operate in the severely crowded and rapidly changing modern commercial environment, while maintaining the optimal performance. To arrive at an optimal reconfigurable RF system, it is crucial to minimize silicon area and power consumption for multiple wireless applications and standards by maximizing hardware sharing [3]. In addition, programming a reconfigurable RF system must be efficiently done by appropriately setting a number of programmable "knobs".

For a reconfigurable RF receiver, its programming can be performed through two possible avenues. First, the receiver can be programmed at system level by changing its architecture (e.g., transforming an RF receiver from superheterodyne to homodyne by using switch boxes). Second, the receiver can be programmed at block level by changing its block-level performances (e.g., varying the gain of an LNA by tuning its bias current) [2]. To successfully program a reconfigurable RF receiver, both system-level and block-level knobs must be appropriately set to achieve the desired functionality and performance. In this paper, we will first derive efficient algorithms for block-level programming, and then extend our proposed techniques to system-level programming for multiple receiver architectures.

As an example for system-level and block-level programming, a reconfigurable RF receiver is shown in Figure 1. In this system, the RF receiver can be programmed to two different architectures: superheterodyne and homodyne, by using system-level knobs, i.e., switch boxes. All circuit blocks, including LNAs, band-pass filter (BPF), low-pass filter (LPF), oscillators (LO1, LO2 and LO3), mixers, and variable gain amplifier (VGA), can be reconfigured by block-level programmable knobs, such as block-level switch boxes and/or tunable bias currents.

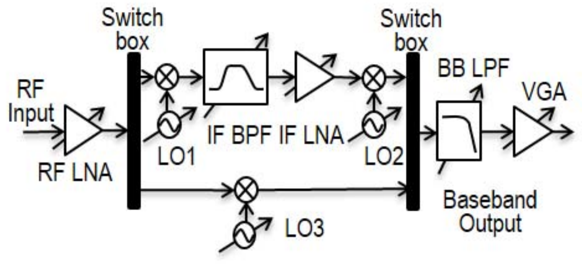

Figure 1. Simplified schematic is shown for a reconfigurable RF receiver which can be programmed to two different architectures: superheterodyne (the upper signal path) and homodyne (the lower signal path).

Mathematically, let **x** denote all block-level programmable knobs (e.g., the values of tunable bias currents used to control the performance metrics of circuit blocks) and **y** denote all system-level programmable knobs (e.g., the configurations of switch boxes used to set the receiver architecture). In general, the possible values of these programmable knobs are discrete and they are controlled by a finite number of digital bits (e.g., a finite set of possible bias current values controlled by a digital-to-analog converter). The problem of reconfigurable RF receiver programming can be formulated as an optimization with multiple constraints:

$$\begin{aligned}&\min_{\mathbf{x},\mathbf{y}} \quad F(\mathbf{x},\mathbf{y})\\ &\text{s.t.} \quad g_j(\mathbf{x},\mathbf{y}) \geq G_j \quad (j=1,2,\cdots,J)\end{aligned}, \tag{1}$$

where $F(\mathbf{x},\mathbf{y})$ (i.e., the cost function) and $g_j(\mathbf{x},\mathbf{y})$ (i.e., the constraint function) denote different system-level performances (e.g., power, area, SNR, etc.), and G_j is the given specification. In order to simplify the RF receiver programming problem, Eq. (1) can be rewritten as an optimization formulation that minimizes the cost function subject to a single constraint:

$$\begin{aligned}&\min_{\mathbf{x},\mathbf{y}} \quad F(\mathbf{x},\mathbf{y})\\ &\text{s.t.} \quad S(\mathbf{x},\mathbf{y}) = \min\left(g_1(\mathbf{x},\mathbf{y})-G_1,\cdots,g_J(\mathbf{x},\mathbf{y})-G_J\right) \geq 0\end{aligned}, \tag{2}$$

where $S(\mathbf{x},\mathbf{y})$ is a new constraint function constructed by all the original constraints in (1). In the remainder of this paper, we will take the formulation in (2) to derive the algorithm for efficient receiver programming.

The optimization problem in (2) is nonlinear, non-convex and discrete. There are two important technical challenges when developing efficient numerical algorithms to solve (2) for RF receiver programming.

- *Robustness*: The proposed optimization algorithm must be sufficiently robust to find the optimum of (2) that is at least close to global optimum. In other words, the optimization algorithm should not easily get stuck at local optima. Note that finding a robust solution for (2) is not trivial, since the problem formulation in (2) is non-convex and, hence, there may exist a large number of local optima in the search space.
- *Complexity*: The optimization algorithm for solving (2) must be computationally inexpensive so that programming an RF receiver can be done quickly. In particular, since evaluating the system-level performance metrics (i.e., $F(\mathbf{x},\mathbf{y})$ and $g_j(\mathbf{x},\mathbf{y})$ in (2)) requires expensive simulation (e.g., more than 10 minutes per simulation for our receiver example shown in Section IV), the proposed optimization algorithm must "smartly" explore the search space with few simulations. For this reason, even though many traditional analog optimization algorithms (e.g., simulated annealing [8]) can find a robust solution that is close to global optimum, they are not suitable for RF receiver programming due to their high computational complexity.

In what follows, we will propose a novel optimization framework that exploits the unique characteristics of RF receiver programming. It, in turn, facilitates us to find a robust solution of (2) with low computational cost.

III. Proposed Approach

As previously mentioned, there are two distinct types of variables in the optimization problem defined by (2): (i) the block-level knobs **x** to program the circuit implementation of each block, and (ii) the system-level knobs **y** to select the appropriate receiver architecture. In this paper, we propose to optimize **x** and **y** by applying a hierarchical approach. Namely, for each possible value of **y** that defines a particular receiver architecture, we optimize **x** to find out the optimal configurations of all circuit blocks. Next, the system-level performance metrics are compared among different architectures and the optimal architecture is selected to satisfy the given constraints with minimal cost function.

The aforementioned programming strategy is adopted because the optimal block-level implementations and their performance metrics are often substantially different for different receiver architectures. For instance, the performance requirements on noise and distortion are significantly different between superheterodyne and homodyne receivers [12]. Hence, it is not trivial to design an algorithm to optimize both **x** and **y** simultaneously.

In what follows, we first describe two novel ideas, two-phase relaxation search and Pareto-driven search space reduction, that facilitate us to efficiently find the optimal **x** (i.e., block-level configurations) for a given **y** (i.e., receiver architecture). Next, we further discuss the overall programming flow with optimal architecture selection.

A. Two-phase Relaxation Search

To find the optimal block-level implementations for a given receiver architecture, we re-write the optimization problem in (2) as:

$$\begin{aligned}&\min_{\mathbf{x},\mathbf{y}} F\left(\mathbf{x},\mathbf{y}^{(q)}\right)\\&\text{s.t. } S\left(\mathbf{x},\mathbf{y}^{(q)}\right)=\min\left(g_1\left(\mathbf{x},\mathbf{y}^{(q)}\right)-G_1,\cdots,g_J\left(\mathbf{x},\mathbf{y}^{(q)}\right)-G_J\right)\geq 0\end{aligned},(3)$$

where $\mathbf{x}$ is the optimization variable and $\mathbf{y}=\mathbf{y}^{(q)}$ corresponds to a particular receiver architecture. Without loss of generality, we assume that there are M programmable blocks and the configuration of the m-th block is defined by a row vector $\mathbf{x}_m$, where $m=1,2,\dots,M$. Concatenating all these vectors $\{\mathbf{x}_m|m=1,2,\dots,M\}$ forms the optimization variable $\mathbf{x}$, i.e., $\mathbf{x}=[\mathbf{x}_1\quad \mathbf{x}_2\quad \cdots\quad \mathbf{x}_M]$. We further denote the set of all possible values of $\mathbf{x}_m$ as $\left\{\mathbf{x}_m^{(i)}\middle|i=1,2,\dots,M_m\right\}$, where each $\mathbf{x}_m^{(i)}$ corresponds to one possible configuration and M_m is the total number of configurations for the m-th block.

When solving the optimization problem in (3), we must repeatedly evaluate the receiver performances for different values of $\mathbf{x}$. It requires a large number of expensive numerical simulations. To make the computational cost affordable, a local relaxation approach may be adopted to choose a single circuit block for optimization at one time.

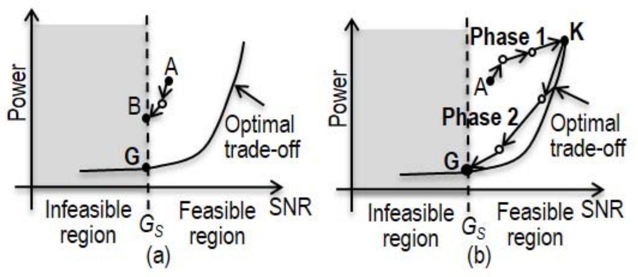

Figure 2. Two different optimization algorithms are compared with the objective to minimize power subject to a given SNR constraint. (a) The traditional algorithm of local relaxation starts from the initial point ***A*** and is stuck at a local optimum ***B***. (b) The proposed algorithm of two-phase relaxation search starts from the initial point ***A*** and reaches the global optimum ***G***.

Such a simple approach, however, often gets stuck at a local optimum, especially when the initial starting point is not appropriately set. To understand the reason, we consider a simple example shown in Figure 2(a) where the objective is to find the global optimum ***G*** by minimizing power subject to an SNR constraint. Suppose that the SNR specification is plotted as a vertical line G_S separating the feasible and infeasible regions in Figure 2. When local relaxation is applied, one circuit block is selected for optimization at each iteration step. The optimal configuration of the selected circuit block (say, the *m*-th block) should be found to minimize power while simultaneously satisfying the SNR constraint. All configurations of the *m*-th block that violate the SNR constraint are considered as "infeasible". Due to this reason, if the initial starting point is close to the constraint boundary G_S (e.g., the point ***A*** shown in Figure 2(a)), a large number of possible configurations of the *m*-th block may be considered to be infeasible because they violate the SNR constraint. In other words, starting from the initial point ***A*** in Figure 2(a) would prevent us from exploring many possible configurations of the *m*-th block that are infeasible in the current iteration step, but should be considered to be feasible if the configurations of other circuit blocks can be changed. The optimization, therefore, fails to converge to global optimum. It is one of the major limitations for the local relaxation algorithm.

To address this issue, we adopt a heuristic technique to solve the optimization problem in (3) via two phases. During the first phase, local relaxation is applied to maximize the constraint function without considering the cost function:

$$\max_{\mathbf{x}} S\left(\mathbf{x},\mathbf{y}^{(q)}\right)=\min\left(g_1\left(\mathbf{x},\mathbf{y}^{(q)}\right)-G_1,\cdots,g_J\left(\mathbf{x},\mathbf{y}^{(q)}\right)-G_J\right). \quad (4)$$

Our goal is to find the configuration that is "farthest" away from the constraint boundaries $G_1,G_2,\dots,G_J$, such as the point ***K*** in Figure 2(b), by maximizing the minimal distance between the constraint function $g_j\left(\mathbf{x},\mathbf{y}^{(q)}\right)$ and the given specification G_j. During the second phase, we further take the optimization result from the first phase as the initial starting point and apply local relaxation to solve (3), i.e., minimize the cost function subject to the given constraints. In this case, since the initial starting point is not close to the constraint boundaries, a large number of configurations should be "feasible" when optimizing a particular circuit block by local relaxation. In other words, the proposed two-phase approach is able to avoid a lot of local optima as compared to the simple local relaxation algorithm shown in Figure 2(a).

Algorithm 1: Constraint Function Maximization (Phase 1)

```
1. Start from the optimization formulation in (3) with a given $\mathbf{y}_q$ corresponding to a particular receiver architecture.
2. Let $\mathbf{x}^{(I)}$ denote the initial value of block-level knobs.
3. Simulate the receiver with $\mathbf{x}^{(I)}$ to obtain the value of the constraint function $S^{(I)}$.
4. Set $\mathbf{x}^{(OLD)}=\mathbf{x}^{(NEW)}=\mathbf{x}^{(I)}$ and $S^{(OPT)}=S^{(I)}$.
5. For $m=1,2,\dots,M$
6.     Set $\mathbf{x}_m$ to different values $\mathbf{x}_m=\mathbf{x}_m^{(i)}$, where $i=1,2,\dots,M_m$, and simulate the receiver at each of these cases to obtain the values of the constraint function $\{S_m^{(i)}\ |i=1,2,\dots,M_m\}$. The configurations of all other blocks are defined by $\left\{\mathbf{x}_i^{(NEW)}\middle|i\neq m\right\}$ and they should be unchanged for these simulations.
7.     Find the optimal value of $\mathbf{x}_m$ (say, $\mathbf{x}_m^{(OPT)}$) corresponding to the largest value of the constraint function (say, $S_m^{(OPT)}$) in the set $\left\{S_m^{(i)}\middle|i=1,2,\dots,M_m\right\}$.
8.     If $S^{(OPT)}<S_m^{(OPT)}$, set $S^{(OPT)}=S_m^{(OPT)}$ and $\mathbf{x}^{(NEW)}=[\mathbf{x}_1^{(NEW)}\ \mathbf{x}_2^{(NEW)}\cdots\mathbf{x}_m^{(OPT)}\cdots\mathbf{x}_M^{(NEW)}]$.
9. End For
10. If $\mathbf{x}^{(NEW)}\neq\mathbf{x}^{(OLD)}$, set $\mathbf{x}^{(OLD)}=\mathbf{x}^{(NEW)}$ and go back to step 5. Otherwise, stop iteration and $\mathbf{x}^{(NEW)}$ is the optimal configuration with the largest value of the constraint function for the given receiver architecture.
```

Algorithm 2: Cost Function Minimization (Phase 2)

```
1. Start from the optimization formulation (2) and the optimal value $\mathbf{x}^{(NEW)}$ solved by Algorithm 1.
2. Calculate the cost function $F^{(OPT)}$ with the configuration $\mathbf{x}^{(NEW)}$.
3. Set $\mathbf{x}^{(OLD)}=\mathbf{x}^{(NEW)}$.
4. For $m=1,2,\dots,M$
5.     Set $\mathbf{x}_m$ to different values $\mathbf{x}_m=\mathbf{x}_m^{(i)}$, where $i=1,2,\dots,M_m$, and simulate the receiver at each of these cases to obtain the values of the cost function $\{F_m^{(i)}\ |i=1,2,\dots,M_m\}$ and the values of the constraint function $\{S_m^{(i)}\ |i=1,2,\dots,M_m\}$. The configurations of all other blocks are defined by $\left\{\mathbf{x}_i^{(NEW)}\middle|i\neq m\right\}$ and they should be unchanged for these simulations.
```

6. Calculate the following evaluation function values $\{E_m^{(i)} | i = 1,2,\ldots,M_m\}$:

$$E_m^{(i)} = \begin{cases} F_m^{(i)} & if \quad S_m^{(i)} \geq 0 \\ INF & if \quad S_m^{(i)} < 0 \end{cases}. \tag{5}$$

7. Find the optimal value of $\mathbf{x}_m$ (say, $\mathbf{x}_m^{(OPT)}$) corresponding to the smallest evaluation function value (say, $E_m^{(OPT)}$) in the set $\{E_m^{(i)} | i = 1,2,\ldots,M_m\}$.
8. If $E_m^{(OPT)} < F^{(OPT)}$, set $F^{(OPT)} = E_m^{(OPT)}$ and $\mathbf{x}^{(NEW)} = [\mathbf{x}_1^{(NEW)}\ \mathbf{x}_2^{(NEW)} \cdots \mathbf{x}_m^{(OPT)} \cdots \mathbf{x}_M^{(NEW)}]$.
9. End For
10. If $\mathbf{x}^{(NEW)} \neq \mathbf{x}^{(OLD)}$, set $\mathbf{x}^{(OLD)} = \mathbf{x}^{(NEW)}$ and go back to step 4. Otherwise, stop iteration and $\mathbf{x}^{(NEW)}$ is the optimal configuration for the given receiver architecture.

Algorithm 1 and Algorithm 2 describe the detailed implementations of the aforementioned two phases. Several important clarifications should be made for these two algorithms. First, the aforementioned technique of two-phase relaxation search was initially developed for gate sizing of digital circuits where the total power should be minimized subject to a given delay constraint [10]. As demonstrated in the literature, it is able to avoid a large number of local optima for gate sizing. In this paper, we adopt the two-phase approach for RF receiver programming, since both gate sizing and RF receiver programming are discrete and they share a similar optimization formulation (i.e., minimizing a cost function subject to a given constraint).

Second, it is important to note that the computational cost of Algorithm 1 and Algorithm 2 is often dominated by the simulation time for performance evaluation, because accurately estimating system-level performances requires us to simulate the RF receiver over a long time period. As will be demonstrated by our numerical examples in Section IV, evaluating the system performances for a receiver with a given configuration takes more than 10 minutes, even if behavioral modeling is applied to speed up the simulation. Since the aforementioned performance evaluation must be repeated within the optimization loop of Algorithm 1 and Algorithm 2, it is extremely important to minimize the total number of required performance evaluations so that our proposed algorithm of two-phase relaxation search is computationally efficient for practical applications. For this reason, we will further develop a novel approach for search space reduction in order to minimize the computational cost of RF receiver programming.

B. Pareto-driven Search Space Reduction

Our key idea of search space reduction is to remove the configurations that cannot be the optimum of (3) or (4), before running Algorithm 1 or Algorithm 2. It, in turn, reduces the search space and, hence, the computational cost of receiver programming. Such a goal is facilitated by the concept of Pareto optimal front (POF) that was previously developed for analog system-level optimization in the literature [11].

Let $\mathbf{p}_m = [p_{m,1}, p_{m,2}, \ldots, p_{m,D_m}]$ denote the performance metrics of interest for the m-th circuit block (e.g., gain, NF, IIP3 and power for an LNA). POF captures the "best" trade-off among these performance metrics. If a point $\mathbf{p}_m^{(i)}$ is on POF (i.e., referred to as a Pareto point), it cannot be dominated by any other point (say, $\mathbf{p}_m^{(j)}$) in the performance space:

$$\mathbf{p}_m^{(i)} \neq \mathbf{p}_m^{(j)} \text{ and } p_{m,k}^{(i)} \leq p_{m,k}^{(j)} \quad (k = 1,2,\cdots,D_m). \tag{6}$$

In (6), we assume that the value of a performance metric (say, $p_{m,k}^{(j)}$) dominates another value (say, $p_{m,k}^{(i)}$), if $p_{m,k}^{(j)}$ is greater than $p_{m,k}^{(i)}$. In general, a linear transformation can be appropriately applied to any given performance metric to make this assumption valid. For instance, a small NF is often preferred for LNA design. In this case, we can define $-1 \cdot \text{NF}$, instead of NF, as the performance of interest. Figure 3 shows a simple example of POF with three performance metrics: $p_{m,1}$, $p_{m,2}$ and $p_{m,3}$.

In our application of RF receiver programming, each possible configuration, $\mathbf{x}_m^{(i)}$ where $i = 1,2,\ldots,M_m$, is associated with a particular performance value $\mathbf{p}_m^{(i)}$ for the m-th block. To achieve the optimal performance for an RF receiver, we only need to consider the configurations on POF as "valid" programming options. Taking the RF LNA of the superheterodyne receiver shown in Figure 1 as an example, if the constraint function is set to SNR and we want to maximize it during the first phase, the RF LNA should be configured to have large gain, low NF and large IIP3. If a configuration option of the RF LNA is not on the POF with respect to gain, NF and IIP3, it is unlikely to reach the maximal SNR and can be simply ignored during the programming process.

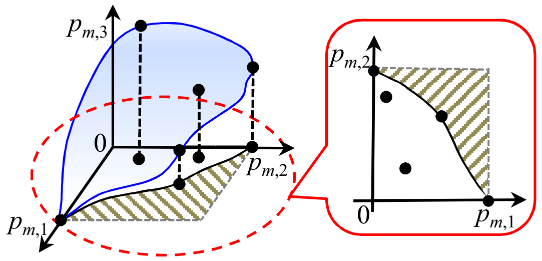

Figure 3. A 3-D POF is plotted for three performance metrics $p_{m,1}$, $p_{m,2}$ and $p_{m,3}$ and is projected to the 2-D plane defined by $p_{m,1}$ and $p_{m,2}$.

The aforementioned idea of Pareto-driven search space reduction is particularly useful when Algorithm 1 is applied to maximize the constraint function without considering the cost function during the first phase. Since the cost function $F(\mathbf{x}, \mathbf{y}^{(q)})$ is ignored in Algorithm 1, the block-level performance metrics which only affect $F(\mathbf{x}, \mathbf{y}^{(q)})$ can also be ignored. Therefore, the total number of block-level performance metrics and, hence, the dimensionality of the block-level performance space could be reduced for each circuit block. Considering the simple example shown in Figure 2(b) where the cost function is set to power and the constraint function is set to SNR, the block-level power consumption could be ignored when maximizing SNR during the first phase. Namely, the dimensionality of the block-level performance space is reduced by one. In this case, when we project the block-level performance points to a lower-dimensional space, a large number of configuration options may not be on POF and, therefore, can be simply ignored. To intuitively illustrate the reason, Figure 3 shows a simple 3-D POF example. If one of the performance metrics (i.e., $p_{m,3}$) is ignored and the 3-D performance points are projected onto the 2-D space defined by $p_{m,1}$ and $p_{m,2}$, many of these performance points are not on POF. Hence, the corresponding configuration options can be ignored.

Algorithm 3 summarizes the simplified flow of our proposed Pareto-driven search space reduction. It starts from a set of performance values $\Theta_m = \{\mathbf{p}_m^{(i)} | i = 1,2,\ldots,M_m\}$ corresponding to different configuration options of the m-th circuit block. Based on the definition of POF, a Pareto point $\mathbf{p}_m^{(i)}$ can be identified by choosing the performance point that has the greatest value for one of the performance metrics (say, $p_{m,1}$). Then all points in Θ_m that are dominated by $\mathbf{p}_m^{(i)}$ are removed. Next, the point $\mathbf{p}_m^{(j)}$ which has the second greatest value for the performance metric $p_{m,1}$ in Θ_m is found and it is taken as the second Pareto point because it is not dominated by $\mathbf{p}_m^{(i)}$ or any other point remaining in Θ_m. The aforementioned selection process is repeated until all Pareto points are found. Finally, the configuration options of the m-th circuit block corresponding to all selected Pareto points are identified and these configurations will be considered as valid options for RF receiver programming.

Algorithm 3: Pareto-driven Search Space Reduction

```
1. Start from a set $\Theta_m = \{\mathbf{p}_m^{(i)} | i = 1,2,\dots,M_m\}$ containing all
   performance values corresponding to $M_m$ configuration options
   of the $m$-th circuit block.
2. Initialize the set $\Xi = \{\}$.
3. Find the Pareto point $\mathbf{p}_m$ from $\Theta_m$ that has the greatest value
   for the first performance metric $p_{m,1}$.
4. Remove all points in $\Theta_m$ that are dominated by $\mathbf{p}_m$.
5. Remove $\mathbf{p}_m$ from the set $\Theta_m$, and add $\mathbf{p}_m$ to the set $\Xi$.
6. If the set $\Theta_m$ is not empty, go to Step 3. Otherwise, stop
   iteration and the set $\Xi$ contains all Pareto points.
7. Determine the configuration options corresponding to the
   performance values belonging to the set $\Xi$.
```

C. *Multi-Architecture Receiver Programming*

The previous sub-sections describe several efficient algorithms to find the optimal value of the block-level programmable knobs $\mathbf{x}$ for a given receiver architecture defined by the system-level knobs $\mathbf{y}$. Suppose that the receiver system can be possibly implemented with Q different architectures: $\{\mathbf{y}^{(q)} | q = 1,2,\dots,Q\}$. For each architecture, the optimization problem in (3) is solved to determine the optimal block-level configurations. Next, all architectures are compared according to their system-level performance values, and the optimal architecture is selected to achieve the minimal cost while simultaneously satisfying the given constraints.

Algorithm 4: Reconfigurable RF Receiver Programming

```
1. Start from the optimization formulation (2) where $\mathbf{y} \in \{\mathbf{y}^{(q)} | q = 1,2,\dots,Q\}$.
2. For each $\mathbf{y}^{(q)}$ where $q = 1,2,\dots,Q$
3.   Apply Algorithm 1~Algorithm 3 to find the optimal block-
     level configurations for the given receiver architecture
     defined by $\mathbf{y}^{(q)}$. Calculate the corresponding cost and
     constraint functions.
4. End For
5. Determine the optimal receiver architecture that can achieve the
   minimal cost subject to the given constraints.
```

Algorithm 4 summarizes the overall flow for RF receiver programming. By taking advantage of the efficient techniques described in this section, the proposed receiver programming only requires very few (e.g., less than 50) simulations to reach convergence, as will be demonstrated by our numerical examples in Section IV.

IV. Numerical Experiments

In this section, a reconfigurable RF receiver (including baseband signal processing) shown in Figure 1 is used as an example to demonstrate the efficiency of the proposed programming algorithm. This RF receiver is designed for the IEEE 802.11a WLAN standard where the channel bandwidth is 20MHz. In this example, the goal of RF receiver programming is to minimize power subject to a given SNR specification. At system level, the receiver can be programmed to two different architectures: superheterodyne (SH) and homodyne (HD). At block level, three circuit blocks are designed to be reconfigurable: RF LNA, IF LNA and IF BPF. Both the RF LNA and the IF LNA can be programmed to 11 configurations respectively, where the LNA performance metrics (i.e., gain, NF, IIP3 and power) are different for different configurations. The IF BPF can be programmed to 9 configurations with different bandwidth and unloaded quality factor. In total, there are 1100 different configurations for the aforementioned RF receiver. Each circuit block of the receiver is represented as a macromodel to facilitate efficient system-level simulation of the RF receiver in MATLAB SIMULINK.

A. *Programming Results*

Table 1. Programming results for the reconfigurable RF receiver with different SNR specifications

SNR Spec (dB)	Algorithm	SNR (dB)	Power (mW)	Selected Arch
6	Exhaustive	7.36	1.32	HD
	Relaxation	7.36	1.32	HD
	Annealing	7.36	1.32	HD
	Proposed	7.36	1.32	HD
8	Exhaustive	8.61	1.44	SH
	Relaxation	8.61	1.44	SH
	Annealing	8.61	1.44	SH
	Proposed	8.61	1.44	SH
9	Exhaustive	9.53	2.04	SH
	Relaxation	9.66	2.60	HD
	Annealing	9.53	2.04	SH
	Proposed	9.53	2.04	SH
10	Exhaustive	10.51	3.32	SH
	Relaxation	Infeasible	Infeasible	Infeasible
	Annealing	10.51	3.32	SH
	Proposed	10.51	3.32	SH

For testing and comparison purposes, four different programming algorithms are implemented: (i) exhaustive search (Exhaustive), (ii) local relaxation (Relaxation), (iii) simulated annealing (Annealing), and (iv) two-phase relaxation (Proposed). The exhaustive search method guarantees to find the globally optimal configuration by exploring all possible options. Hence, it is used to generate the “golden” results to evaluate the “robustness” of other programming methods.

Table 1 summarizes the programming results for different algorithms with different SNR specifications. Studying Table 1 reveals an important observation that the traditional algorithm of local relaxation fails to find the globally optimal configuration, when the SNR specification is 9 dB or 10 dB. On the other hand, the proposed algorithm of two-phase relaxation converges to the global optima in all test cases, thereby demonstrating its superior performance over the simple relaxation method. The simulated annealing algorithm is also able to find the global optima in this example; however, its computational cost is substantially more expensive than our proposed approach, as will be demonstrated in the next sub-section.

B. Computational Cost

During the programming process, system-level performances, such as SNR, must be repeatedly evaluated for different receiver configurations. For the RF receiver shown in Figure 1, a single SIMULINK simulation takes more than 10 minutes to evaluate its SNR for a given receiver configuration. Hence, the computational cost of receiver programming is completely dominated by the simulation time, and we can use the number of required simulations as a metric to compare the complexity of different programming algorithms, as shown in Table 2.

Table 2. The number of required simulations for different programming algorithms with different SNR specifications

SNR Spec (dB)	Exhaustive	Relaxation	Annealing	Proposed
6	1100	32	170	35
8	1100	32	179	34
9	1100	36	331	34
10	1100	36	259	40

Note that both exhaustive search and simulated annealing require a large number of simulations before reaching convergence. Our proposed algorithm of two-phase relaxation with Pareto-based search space reduction and the traditional algorithm of local relaxation can greatly reduce the computational cost by more than 5×. On the other hand, the proposed algorithm is more “robust” than local relaxation, as shown in Table 1. These observations demonstrate that our proposed method is superior over all other traditional approaches tested in this example.

In our experiments, we further observe that if the proposed heuristic method of Pareto-based search space reduction is not applied, more than 80 simulations are required to find the optimal configuration by using two-phase relaxation search. In other words, the proposed search space reduction successfully reduces the computational cost by more than 2× in this example.

V. Conclusions

In this paper, we develop an efficient methodology for programming reconfigurable RF receivers to achieve minimal cost function value subject to a set of given constraints. It is mathematically formulated as a nonlinear, discrete optimization problem. Two novel techniques, two-phase relaxation search and Pareto-based search space reduction, are proposed to program the RF receiver both robustly (i.e., close to global optimum) and efficiently (i.e., with low computational cost). Our numerical experiments demonstrate that the proposed approach is superior over other traditional algorithms based on either local relaxation (in terms of robustness) or simulated annealing (in terms of complexity).

VI. Acknowledgements

This work is sponsored by the DARPA RF-FPGA (Radio Frequency-Field Programmable Gate Arrays) program under Grant HR0011-12-1-0005. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

[1] H. Darabi, A. Mirzaei and M. Mikhemar, "Highly integrated and tunable RF front ends for reconfigurable multiband transceivers: A tutorial," *IEEE Trans. on. CAS-I,* vol. 58, no. 9, pp. 2038-2050, 2011.

[2] A. Goel, B. Analui and H. Hashemi, "A 130-nm CMOS 100-Hz–6-GHz reconfigurable vector signal analyzer and software-defined receiver," *IEEE Trans. on MTT,* vol. 60, no. 5, pp. 1375-1389, 2012.

[3] F. Agnelli, G. Albasini, I. Bietti, etc., "Wireless multi-standard terminals: System analysis and design of a reconfigurable RF front-end," *IEEE Circuits and Systems Magazine,* vol. 6, no. 1, pp. 38-59, 2006.

[4] F. Carrara and G. Palmisano, "High-efficiency reconfigurable RF transmitter for wireless sensor network applications," *IEEE RFIC*, pp. 179-182, 2010.

[5] H. Liu, A. Singhee, R. A. Rutenbar, etc., "Remembrance of circuits past: Macromodeling by data mining in large analog design spaces," *IEEE DAC*, pp. 437-472, 2002.

[6] G.G. Gielen and R.A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proceedings of the IEEE,* vol. 88, no. 12, pp. 1825-1854 , 2000.

[7] M. Hershenson, S. Boyd and T. Lee, "GPCAD: A tool for CMOS Op-Amp synthesis," *IEEE ICCAD*, pp. 296-303, 1998.

[8] G. Gielen, H. Walscharts and W. Sansen, "Analog circuit design optimization based on symblic simulation and simulated annealing," *IEEE JSSC,* vol. 25, no. 3, pp. 707-713, 1990.

[9] M. Krasnicki, R. Phelps, R.A. Rutenbar, etc., "MAELSTROM: Efficient simulation-based synthesis for analog cells," *IEEE DAC*, pp. 945-950, 1999.

[10] O. Coudert, "Gate sizing for constrained delay/power/area optimization," *IEEE Trans. on VLSI,* vol. 5, no. 4, pp. 465-472, 1997.

[11] T. Eeckelaert, T. McConaghy and G. Gielen, "Efficient multiobjective synthesis of analog circuits using hierarchical Pareto-optimal performance hypersurface," *IEEE DATE*, pp.1070-1075, 2005.

[12] B. Razavi, *RF Microelectronics*, Prentice Hall, 2011.