2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

RiverOpt: A Multiobjective Optimization Framework based on Modified River Formation Dynamics Heuristic

Satyabrata Dash[†], Sukanta Dey[‡], Anish Augustine[§], Rudra Sankar Dhar[†], Jan Pidanič[††], Zdenek Nemec[††], Gaurav Trivedi[§]
[†]*Department of Electronics and Communication Engineering, National Institute of Technology Mizoram, Aizawl*
[§] *Department of Electronics and Electrical Engineering &* [‡]*Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam*
[††]*Fakulta Elektrotechniky a Informatiky, Univerzita Pardubice, Pardubice, Czechia*
Email: satyabrata.ece@nitmz.ac.in

***Abstract*—In river formation dynamics (RFD) method, water drops pursue a probable path to flow from high altitudes to flat surface. This geographical metaphor adopts a decreasing gradient principle supported by sedimentation and erosion mechanisms to reach for a feasible solution. In this paper, a new multi-objective optimization framework, RiverOpt is presented based on a modified RFD method. In this method, the probability of selecting the next path in RFD method is modified to exploit both transverse and longitudinal slopes. Further, the sedimentation parameter in RFD method is improved by introducing a sediment coefficient. Later, an external archive is integrated with RiverOpt framework to keep track of nondominated solutions in each generation. For benchmarking the performance of the proposed framework, a set of standard multiobjective test problems is employed. The results are compared with peer multiobjective optimization algorithms using two performance indicators (i.e., generational distance and hypervolume). Experimental results show that the proposed RiverOpt framework demonstrates competitive results in terms of convergence and diversity of Pareto optimal solutions. Finally, a case study of low noise amplifier circuit is analyzed to showcase effectiveness of the proposed framework.**

***Keywords*-Multiobjecitve optimization, Pareto optimal solution, River formation dynamics.**

I. Introduction

In order to find satisfactory solutions to hard optimization problems, several nature-inspired methods have been presented in literature over the years [1]. As hard optimization problems cannot be solved to optimality, or to any guaranteed bound, it is difficult to solve using deterministic methods within a reasonable amount of time. Therefore, nature-inspired methods (specifically metaheuristics) are applied to such problems for which there are limited satisfactory algorithms to solve them. Although metaheuristics are very productive in providing solutions to NP-complete problems, solving diverse hard optimization problems is quiet a challenging task [1]. Further, many real-world problems consist of several competing objectives which require simultaneous optimization (multiobjective optimization) and finding single optimal solutions for these problems do not provide accountability. In such cases, a set of alternate solutions that are superior to other solutions in the search space are obtained by considering all objectives. These nondominated solutions are known as Pareto optimal solutions.

Nature-inspired metaheuristics are suggested to be best suited for such multiobjective search and for obtaining Pareto optimal solutions than other blind search strategies [2]. Since the past three decades, several nature-inspired multiobjective optimization methods have been developed, capable of searching for Pareto optimal solutions in a single run [3]. These methods make use of Pareto dominance relation [4] or scalar function [4] or hypervolume assessment [3] for fitness evaluation during multiobjective optimization. Pareto dominance-based algorithms, e.g., NSGA-II [5], usually demonstrate superior performance while solving multiobjective problems having two or three objectives than scalar function-based and hypervolume-based algorithms. This is due to the generation of nondominated solutions in the early generations. As it is desirable to have nondominated solutions that approximate the entire Pareto front of a multiobjective problem, the algorithms which provide a wide variety of nondominated solutions are assumed to have better performance. However, these algorithms often lack the appropriate selection pressure that drives the early generated nondominated solutions towards the true Pareto front. Therefore, various approaches have been presented in literature to improve the convergence property of Pareto dominance-based algorithms [3]. However, the improvement in convergence often affects the extent of diversity in obtained Pareto optimal solutions. Various diversity preservation schemes [3] are introduced inside optimization frameworks to maintain the diversity in each generation. Such modifications to existing optimization frameworks along with various preservation schemes drive the research towards proposal of new optimization algorithms.

According to no-free-lunch theorem [3], no metaheuristic can outperform other metaheuristics. Therefore, in spite of various proposals, it is often difficult to determine, whether to develop a better general optimization framework based on an appropriate algorithm to effectively solve a wide spectrum of problems, or to design an application specific algorithm that fully exploits the mathematical properties of the problem. As application specific algorithms lack quantitative comparative studies while solving hard optimization problems, it is often a good practice to develop a generic optimization algorithm which can be applicable to a diverse set of problems.

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00059

In this paper, a generic optimization framework, i.e., RiverOpt, is developed to solve a set of diverse hard optimization problems. RiverOpt is based on modified river formation dynamics (RFD) method followed by a diversity preservation scheme. An overview of the RFD method and proposed modifications are presented in section II. Further, an external archive is incorporated inside the framework to store nondominated solutions obtained during the search, followed by a crowding distance scheme [5] to preserve diversity. The implementation details of proposed RiverOpt framework are discussed in section III. Moreover, to assess the performance of RiverOpt, several experiments are performed on a set of diverse multiobjective optimization problems and various performance metrics are employed in section IV to compare the results with other peer multiobjective algorithms. A low noise amplifier circuit is analyzed in section V to study the tradeoff among gain and noise figure using RiverOpt framework subject to a set of design constraints. Finally, section VI is devoted to concluding remarks and future perspectives.

II. River Formation Dynamics (RFD) and Proposed Modifications

River formation dynamics method is based on the analogy that water drops traverse from source to destination (sea) by following a decreasing gradient principle [6]. Each water drop is allowed to follow same or different paths probabilistically in the search space for finding a solution. Movement of water drops across the landscape can be viewed as a change in the values of design variables (e.g., x_i, where $i \in N$) in ascending or descending direction to achieve convergence within the solution space. Because of random probabilistic behavior of RFD method, transition function can be realized in the form of a highly dependent probability distribution function $P_n(x)$ [6].

$$P_n(x) = \begin{cases} \frac{\nabla Dr(x, x_{ngb})}{\sum_{ngb=1}^{\deg(x)} \nabla Dr(x, x_{ngb})} & \text{if } x_{ngb} \in N(x), \\ 0 & \text{if } x_{ngb} \notin N(x), \end{cases} \quad (1)$$

where $P_n(x)$ is constructed from a sequence of design variables $\mathbf{x}$ in the search space and $\nabla Dr(x, x_{ngb})$ represents the gradient between variable, x and its neighbor variable, x_{ngb} with altitudes $Y_{alt}(x)$ and $Y_{alt}(x_{ngb})$, respectively. $\nabla Dr(x, x_{ngb})$ can be expressed as [6],

$$\nabla Dr(x, x_{ngb}) = \frac{Y_{alt}(x) - Y_{alt}(x_{ngb})}{\Phi(x, x_{ngb})}, \quad (2)$$

where $\Phi(x, x_{ngb})$ denotes the weight of edge or distance $e(x, x_{ngb})$. The overall optimization procedure is described in Algorithm 1. Both probability, P_n and decreasing gradient, ∇D_r among variables (individuals) and their neighbors are evaluated using (1) and (2), respectively. The path (edge) having maximum probability is selected, and the variable associated with the path is chosen as the next individual

Algorithm 1: Procedure for finding minimum cost function using RFD method.

Input : Design parameter, x
Output : Minimum cost function, $\mathtt{F}_{\min}$

1: **for** $j \leftarrow 1$ to NUM_EXP **do**
2: Generate random population of size N for x
3: $total\_erosion \leftarrow \phi$
4: $\mathtt{F} \leftarrow f(x)$ $\triangleright$ Evaluate population
5: **while** $stopping\ criteria$ **do**
6: $r \leftarrow \mathtt{random(0,1)}$
7: Find number of $x_{neighbor}$
8: **for** $k \leftarrow 1$ to N_x **do**
9: Select $x_{neighbor}$ at random
$\triangleright$ N_x denotes number of neighbors of x
10: **if** $r < Pr(x, x_{neighbor})$ **then**
11: $break$ $\triangleright$ $Pr(x, x_{neighbor}) \equiv P_n(x)$
12: **end if**
13: **end for**
14: $x_{next} \leftarrow x^k$ $\triangleright$ find the next variable
15: $total\_erosion \leftarrow total\_erosion + Er(x, x_{next})$
$\triangleright$ update erosion using (3)
16: **if** $x_{next} < x^L_{next} \mid\mid x_{next} > x^U_{next}$ **then**
17: $x_{next} \leftarrow x^L_{next} + rand(0,1) \times (x^U_{next} - x^L_{next})$
18: **end if**
19: **if** $x_{next} == \phi$ **then**
20: $x_{next} \leftarrow x_{next} + \epsilon$
21: **end if**
22: $x_{next} \leftarrow x_{next} - total\_erosion$
23: $x \leftarrow x_{next}$
24: $\mathtt{F}' \leftarrow f(x)$
25: **end while**
26: **if** $\mathtt{F}' \nleq \mathtt{F}\ \forall\ x$ **then**
27: $\mathtt{F}_{\min} \leftarrow \mathtt{F}$
28: **end if**
29: **end for**

(x_{next}). Additionally, the variable value (fitness value) is changed by reducing an amount of erosion (Er) between the two variables (individuals). The amount of erosion (Er) between two variables, x and x_{next} is evaluated as [6],

$$Er(x, x_{next}) = \frac{\alpha}{(N_S - 1)d} \times \nabla D_r(x, x_{next}), \quad (3)$$

where $Er(x, x_{next})$ represents the amount of erosion when a water drop travels from individual x to the next individual x_{next}; $\alpha = \mathtt{rand}(0,1)$ represents the erosion parameter; N_S denotes the number of individuals (variables) in search space S and $d = 25$ represents the number of drops employed in each generation. The number of drops, d is considered as per the analysis reported in [7]. The variable value is updated in each iteration as described in step 22 of Algorithm 1. The process is continued until all variables in the search space are analyzed (or a stopping criteria is met). Finally, the values of all variables are evaluated and the best fitness value ($\mathrm{F}_{\min}$) is selected through Pareto dominance [5].

Although the current RFD method is capable of evaluating an optimal solution, the ability to fine tune the solution according to changes in search space dimension or to any uncertainty is comparatively weak as compared to other standard nature-inspired methods. One of the reasons for such performance is due to lack of efficient utilization of certain fundamental factors, such as probabilistic decision making and efficient sedimentation during flow of water drops. Therefore, we propose to improve the search capability of RFD method by incorporating a probabilistic decision making capability based on both longitudinal and transverse slopes followed by a modified sedimentation process.

A. Modified Probability

In RFD method, water drops follow a sequence of paths having maximum probability. This random probabilistic decision is made by evaluating only longitudinal slope (i.e., ∇Dr) between two altitudes as described in (2). However, in the presence of elevated river channel beds, the modified probability ($\hat{P}_n$) can be evaluated by considering both longitudinal and transverse slopes as given below.

$$\hat{P}_n(x) = \begin{cases} \frac{S_r(x,x_{ngb})}{\sum_{ngb=1}^{\deg(x)} S_r(x,x_{ngb})} & \text{if } x_{ngb} \in N(x), \\ 0 & \text{if } x_{ngb} \notin N(x), \end{cases} \quad (4)$$

where, S_r denotes the resultant slope, $|S_r|=\sqrt{\nabla Dr^2 + S_n^2}$; S_n represents the transverse slope, and it can be expressed as follows,

$$S_n = \frac{Y_{alt}(x) - Y_{alt}(x_{tngb})}{\Phi(x, x_{tngb})} \quad (5)$$

where $Y_{alt}(x_{tngb})$ denotes the altitude of the neighbor variable x_{tngb} present at the transverse side of x.

B. Sedimentation Parameter

In case of excess erosion, the variable values (altitudes) may become close to zero or may become equal. In such cases, the movement of water drop comes to an end, which results in premature convergence. The current RFD method makes use of a general sedimentation parameter, ϵ $\left(= \frac{(total\_erosion)}{(N_S-1)}\right)$ as reported in [6] to increase the values of variables in each iteration and to avoid any premature convergence (step 20 in Algorithm 1). As the value of ϵ depends only on the amount of erosion, it becomes supplement to the variable values if excess erosion occurs multiple times. As a consequence, there is often inadequate increase in variable values and it reduces the rate of convergence. Therefore, it is proposed to introduce a sediment coefficient, ξ while evaluating ϵ as follows,

$$\epsilon = \xi \times \frac{(total\_erosion)}{(N_S - 1)}, \quad (6)$$

where, $\xi = \texttt{logsig}\left(\frac{\Theta-\theta}{2\omega}\right) \times rand()$, ξ is a coefficient that weight the contribution of sedimentation parameter ϵ; `logsig()` represents the logarithmic sigmoid transfer function; ω is a constant used to change the slope of `logsig()`; Θ and θ represent the maximum and the current iteration numbers, and $rand()$ function evaluates a uniform random value between 0 and 1.

III. Proposed Multiobjective Framework

In this section, a multiobjective framework based on modified RFD method, i.e., RiverOpt is presented. Similar to the RFD method described in Algorithm 1, RiverOpt is also a population based multiobjective optimization framework, where each population individual corresponds to water drops and the altitude values denote the fitness values of variables in a search space.

A. General Framework

RiverOpt is a nature-inspired optimization framework that makes use of Pareto dominance principle to approximate the Pareto front of multiobjective optimization problems. The working of the framework is described as follows.

1) Input : A multiobjective optimization problem (MOP), A stopping criteria, Number of population individuals (N_S), Constant value (K) for changing `logsig()`, Archive size (a_r)
2) Output : The nondominated solutions, N_S available in external archive A, objectives $\mathbf{F}_{\min}$
3) Step 1 : Initialization
   - Generate an initial population, x_i, $i \in N_S$.
   - Set the generation number, $g = 1$.
   - Evaluate constraints, i.e., both boundary constraints using random constraint evaluation method [8] and constraint functions using constraint dominance principle [5].
   - Evaluate initial population, $\mathbf{F} = f(x_i)$, $i \in N_S$.
4) Step 2 : Update external archive, A
   - After evaluation of initial population, store the initial fitness values $\mathbf{F}$ in external archive, A.
   - Generate new population individuals, x_j from x_i, where $i, j \in N_S$ using modified RFD method as described in section II, i.e., employ modified probability and new sedimentation parameter during analysis.
   - Evaluate the new population, $\mathbf{F} = f(x_j)$ and apply Pareto dominance to generate nondominated solutions, $\mathbf{F}_{\min}$.
   - If $|A| < a_r$, add nondominated solutions to A.
   - If $|A| \geq a_r$, sort nondominated solutions and rank them using crowding distance scheme [5], i.e., A is maintained by keeping the individuals having better nondominated solutions using crowding distance scheme followed by niche count [5].
   - Employ polynomial mutation operation [5] to generate new individuals from best individuals (i.e., individuals having best fitness value) in A and replace the best individual if the mutated individual has better fitness value.
5) Step 3 : Stopping criteria
   - If stopping criteria is satisfied, then terminate the procedure and the nondominated solutions

available in A are regarded as output. Otherwise, proceed to step 2.

6) Step 4 : $g = g + 1$

The RiverOpt framework begins the optimization process by initializing the altitudes corresponding to each drop in the population with an equal non-zero random value. In the first step, a definite population (N_S) of decision variables (x_i) are generated at random. Each individual in the population is evaluated to obtain a solution to the fitness function. All initial solutions are treated as nondominated solutions and are stored in an external archive (A) as described in step 2. The decision variables (x_i) are updated in each iteration using modified RFD method and the corresponding solutions are evaluated as described in step 2 of RiverOpt framework. The process is continued for a number of iterations, which is referred as generations. As mentioned above, RiverOpt makes use of an external archive, A to store nondominated solutions. The criteria based on which the archive is maintained, plays a pivotal role in ensuring both convergence to true Pareto optimal front as well as diversity of solutions. At first, initial nondominated solutions are stored in A. As new nondominated solutions are created in each generation, they are appended to A. When A is full, i.e., $A \geq a_r$, excess solutions in the archive are removed by employing crowding distance scheme in order to accommodate the new nondominated solutions in each generation. However, individuals having identical fitness values can be awarded similar crowded distance values and increase in number of these individuals affects the diversity among nondominated solutions. Therefore, niche count [5] is evaluated for each individual in A to preserve diversity.

B. Preserving Diversity

To exhibit diversity among the nondominated solutions during subsequent generations, a polynomial mutation operation [5] is performed on the best fitness individual (i.e., individual having the best fitness value) in each generation of RiverOpt. The mutated individual and best fitness individual are compared, and the individual having the best fitness value is selected and stored in the archive A. The procedure for applying polynomial mutation operator [5] to a variable x in RiverOpt is expressed as follows,

1) Generate a random number, β between 0 and 1.
2) Generate a parameter $\hat{\lambda}$, such that

$$\hat{\lambda} = \begin{cases} [2\beta + (1-2\beta)(1-\lambda)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1 & \text{if } \beta \leq 0.5, \\ 1 - [2(1-\beta) + 2(\beta - 0.5)(1-\lambda)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} & \text{otherwise,} \end{cases} \quad (7)$$

where $\lambda = \min|(x - x^L), (x^U - x)]/(x^U - x^L)|$, x^U and x^L denote the upper and lower limits of the variable x, respectively; η_m represents distribution index of polynomial mutation and $\eta_m \in$ [20,100] [9].

3) The mutated value of the variable x is evaluated using, $x_m = x + \hat{\lambda}(x^U - x^L)$.

IV. Results and Discussion

The RiverOpt framework is implemented on Linux environment using C/C++ language. The simulation experiments are performed on a machine with Intel Xeon E5-2620 processor having 64GB of RAM. During optimization, a total population size of 200 is considered to evolve over a maximum generation of 1000. The population size and the number of generations (i.e., 2×10^5 number of function evaluations) are set according to existing studies [1]. The size of external archive is fixed at 100. In addition, 25 independent runs are performed for each experiment to qualitatively assess the performance of RiverOpt. Further, several standard multiobjective optimization test problems (eight unconstrained problems and four constrained problems) are considered for evaluation. Different performance indicators [4], i.e., generational distance (GD) and hypervolume (HV) are employed to measure the convergence and diversity of nondominated solutions along the Pareto front. GD metric measures the minimum Euclidean distance between obtained solutions and sample reference points (true Pareto front). GD measure the convergence of a solution towards true Pareto front and lower GD value indicates better convergence towards true Pareto front. Contrarily, HV metric evaluates the maximum area covered by the nondominated solutions with respect to origin. If the HV value is close to one, it indicates better convergence and diversity. Further, for the purpose of comparison, authors implementation of five different state-of-art multiobjective optimization techniques, i.e., NSGA-II [5], SPEA-II [10], dMOPSO [3], SMPSO [11] and MOEA/D [4] are considered. Various parameters of these algorithms are set according to standard implementation procedure available in literature.

Table I represents the mean GD and HV values for RiverOpt along with other standard optimization algorithms based on 25 independent runs. It can be observed from Table I that RiverOpt has demonstrated superior performance (lower GD values) for seven benchmarks (ZDT1, ZDT2, ZDT3, ZDT6, KUR, CEX and OZY [5], [12]) and competitive performance for three benchmarks (SCH, FON and TNK). However, the performance of RiverOpt is inferior in case of two benchmarks, i.e., ZDT4 and BIN. This is due to the unavailability of additional parameter tuning for handling ZDT4 and BIN benchmarks. Further, it can be observed from Table I that RiverOpt obtains higher HV values in nine benchmarks (ZDT1-ZDT4, ZDT6, FON, KUR, BIN and TNK) and competitive HV values in other three benchmarks (SCH, CEX and OZY). The usage of RiverOpt to solve various optimization problems appear to be fruitful as the obtained Pareto fronts are observed to be closer to true

Table I
MEAN VALUES OF GD AND HV METRICS OF DIFFERENT ALGORITHMS ON STANDARD TEST FUNCTIONS

Methods	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	SCH	FON	KUR	BIN	CEX	TNK	OZY
GD metric:												
NSGA-II	1.1517e-01	2.0507e-01	9.6721e-02	4.0117e+01	0.2416e+01	1.1460e-02	2.4156e-03	7.3262e-02	1.8161e+01	6.0349e-03	1.1905e-02	0.2725e+01
SPEA-II	3.4137e-03	7.7292e-03	1.3506e-03	1.9055e-01	1.1563e-01	4.3401e-01	2.6145e-04	2.4346e-04	3.2894e-02	2.5679e-04	1.7636e-03	5.9922e-04
dMOPSO	1.0490e-04	4.9559e-05	9.8536e-05	8.9505e-05	4.1801e-05	2.2700e-04	**1.2825e-04**	4.1136e-04	**9.6670e-03**	1.8696e-02	5.470e+01	0.9520e+01
SMPSO	1.0256e-04	4.8441e-05	1.1824e-04	**7.4796e-05**	1.6137e-03	**2.2366e-04**	1.2841e-04	2.4206e-04	2.7278e-02	2.0802e-04	**1.6630e-03**	9.0512e-04
MOEA/D	3.2241e-03	2.7419e-01	1.7304e-01	1.7458e-03	2.8336e-01	8.5561e-01	2.7198e-01	1.7886e+01	-	-	-	-
RiverOpt	**1.0141e-04**	**2.1050e-05**	**4.3431e-05**	1.0410e+01	**3.8891e-05**	1.7810e-03	1.2862e-03	**2.4030e-04**	2.1102e+01	**1.1081e-04**	1.4894e-02	**2.7812e-04**
HV metric:												
NSGA-II	0.6575	0.3241	0.5131	0.6518	0.3773	0.6620	0.3079	0.3994	0.7249	0.7629	0.3343	0.3621
SPEA-II	0.6164	0.0528	0.4868	0.6577	0.0447	0.5504	0.3102	0.3996	0.7242	0.7752	0.7045	0.3047
dMOPSO	0.6615	0.3283	0.5136	0.6609	0.4013	**0.8284**	0.3118	0.3962	0.7283	**0.8305**	0.7567	**0.7194**
SMPSO	0.6617	0.3286	0.5155	0.6615	0.4012	0.8208	0.3124	0.4001	0.7275	0.7756	0.6981	0.3047
MOEA/D	0.6644	0.3311	0.5161	0.6646	0.4047	0.7936	0.3155	0.4034	-	-	-	-
RiverOpt	**0.6892**	**0.6301**	**0.7160**	**0.7101**	**0.4055**	0.7863	**0.4205**	**0.6985**	**0.7780**	0.7901	**0.7967**	0.4784

Pareto fronts and the solutions are uniformly distributed along the Pareto front, i.e., lower GD and higher HV values.

The reason for better performance of RiverOpt is attributed to the incorporation of new search strategy (probabilistic decision and changes in sedimentation parameter evaluation) and efficient selection of individuals having unique fitness values in the external archive. As mentioned in section III, RiverOpt makes use of crowding distance scheme followed by niche count to maintain diversity in its archive population. However, it is often difficult to generate a uniform distribution of Pareto optimal solutions in each generation. Usually, the diversity among the nondominated solutions in the current generation is associated with the diversity among the solutions generated in previous generation. Therefore, employing a mutation operation in each generation, such as polynomial mutation operator, results in better exploration of search space and the nondominated solutions are properly chosen during selection. This increases the generation of evenly distributed nondominated solutions along the Pareto front. In view of this, RiverOpt has the ability to produce a uniform distribution of Pareto solutions while optimizing various multiobjective optimization problems as listed Table I. To showcase the applicability of RiverOpt, a design example of low noise amplifier is analyzed in the following section.

V. DESIGN EXAMPLE: LOW NOISE AMPLIFIER

In this section, a design example of low noise amplifier (LNA) circuit is considered as shown in Figure 1 to study the tradeoff among two different specifications, i.e., transducer power gain (G_T) and noise figure (NF). The LNA is designed using 0.18μm CMOS technology to have a 2.4GHz center frequency. The power supply voltage is kept at 1.8V while the NF is kept at minimum value ($i.e.$, $\leq$ 2.5 dB) for a drain current of $1m$A at an expense of $8m$W power consumption. The quality factor is maintained between ($Q \in [3, 5]$) for the design. While designing a LNA, it is crucial to have a minimum NF at the output of the amplifier with subsequent amount of G_T to drive the signal to subsequent blocks of the communication system. In this regard, a tradeoff among minimum NF and maximum G_T (= $20log|S21|$) is considered during the design of LNA subject to a set of design constraints, and the design problem is analyzed as a constrained multiobjective optimization problem. The closed form expressions of G_T, NF and other design constraints of LNA are based on the design as reported in [13].

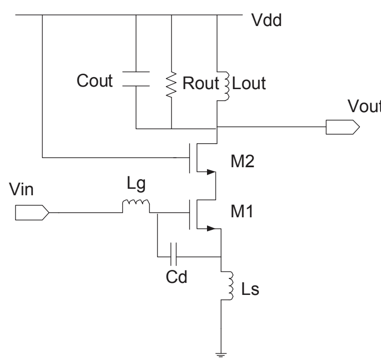
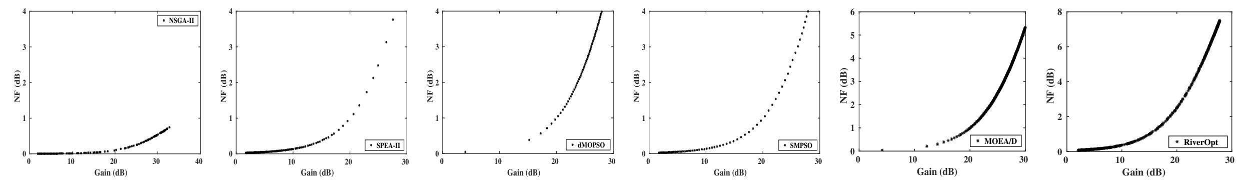


Figure 1. Low noise amplifier circuit [13].

Table II
LNA CONSTRAINTS AND SPECIFICATIONS.

Specifications	Description	Constraints
$S11$[dB]	Input reflection coefficient @ 2.4GHz	≤ -20
$S12$[dB]	Reverse isolation @ 2.4GHz	≤ -50
$S21$[dB]	Forward power gain @ 2.4GHz	≥ 10
$S22$[dB]	Output reflection coefficient @ 2.4GHz	≤ -0.9
NF_{min}[dB]	Min. noise figure @ 2.4GHz	≤ 2.5
P_{max}[mW]	Max. power consumption @ 2.4GHz	≤ 8
f_{cf}[GHz]	Center frequency	$[2.1, 2.7]$
Q	Quality factor	$[3, 5]$
$Z_{out}^{min}[\Omega]$	Min. output impedance (real part)	≥ 50

During optimization of LNA, three different design variables, $W_1 \in [1\mu\text{m}, 100\mu m]$, $W_2 \in [1\mu\text{m}, 100\mu m]$ and $I_d \in [0.1m\text{A}, 4.5m\text{A}]$ are considered. Table II summarizes various design constraints and their specifications that are employed during optimization of LNA. As RiverOpt framework is developed for minimization of multiple objective functions simultaneously, during analysis of LNA, the sign of G_T is inverted to convert the maximization problem into a minimization problem. The final nondominated solutions



(a) (b) (c) (d) (e) (f)

Figure 2. Different plots of nondominated fronts having maximum G_T and minimum NF of LNA obtained by using (a) NSGA-II, (b) SPEA-II, (c) dMOPSO (d) SMOPSO, (e) MOEA/D, and (f) RiverOpt.

obtained using RiverOpt and other standard optimization algorithms are shown in Fig. 2. It can be observed from Fig. 2 that the distribution of solutions along the Pareto front is continuous for RiverOpt, while in case of other optimization algorithms, the solutions are observed to be crowded in one of the extreme corners of the Pareto front. Further, it can be observed from Fig. 2(f) that NF increases slowly with increase in G_T below 15 dB, and consequently NF continues to increase at a faster rate for high gain values. The reason for this occurrence of distributed nondominated solutions along the Pareto front of RiverOpt is due to better local search capability of RFD scheme and preservation of diversity while storing excess nondominated solutions in the external archive.

VI. Conclusion

In this paper, a simple and generic nature-inspired multiobjective optimization framework based on modified RFD method, called RiverOpt is presented. Both probability and sedimentation parameter associated with RFD method are modified to improve the local search capability. An external archive of nondominated solutions is maintained in the framework to obtain a set of uniformly distributed solutions along the Pareto front. To maintain the diversity across generations, polynomial mutation operation is performed on the best fitness values. To demonstrate the effectiveness of the proposed RiverOpt framework, various standard multiobjective test problems are solved and the results are compared with several peer multiobjective algorithms. It is observed that RiverOpt has outperformed other peer algorithms on most test instances. The RiverOpt framework has many aspects for future studies, among which the following three are worth mentioning. First, the effect of sedimentation coefficient, ξ with increase in the size of population on the convergence of RiverOpt is yet to be investigated. Second, it is interesting to explore the variation in decision variables with respect to variation in the number of function evaluations [1] while solving standard multiobjective test functions. Finally, the performance of RiverOpt with increase in the dimension of optimization problems needs to be investigated to showcase scalability. Further, a hybrid of mathematical programming methods and modified RFD method can be developed to improve the performance in optimizing complicated multiobjective and various many-objective optimization problems.

References

[1] K. Sörensen, M. Sevaux, and F. Glover, "A history of metaheuristics," *Handbook of heuristics*, pp. 1–18, 2018.

[2] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. II. application example," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 1, pp. 38–47, Jan 1998.

[3] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2017.

[4] Y. Marca, H. Aguirre, S. Zapotecas, A. Liefooghe, B. Derbel, S. Verel, and K. Tanaka, "Pareto dominance-based MOEAs on problems with difficult pareto set topologies," in *GECCO*. ACM, 2018, pp. 189–190.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[6] P. Rabanal, I. Rodríguez, and F. Rubio, "Applications of river formation dynamics," *Journal of Computational Science*, vol. 22, pp. 26–35, 2017.

[7] S. Dash, D. Joshi, and G. Trivedi, "CMOS analog circuit optimization via river formation dynamics," in *26th International Conference Radioelektronika*. IEEE, 2016, pp. 51–55.

[8] J. Lampinen, "A constraint handling approach for the differential evolution algorithm," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 2. IEEE, 2002, pp. 1468–1473.

[9] K. Deb and S. Tiwari, "Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1062–1087, 2008.

[10] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Computer Engineering and Networks Laboratory (TIK), ETH, Zurich, Switzerland, Tech. Rep. 103, 2001.

[11] A. Nebro, J. Durillo, J. García-Nieto, C. Coello Coello, F. Luna, and E. Alba, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *IEEE MCDM*, 2009, pp. 66–73.

[12] C. A. Coello, G. B. Lamont, D. A. Van Veldhuizen *et al.*, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[13] P. Andreani and H. Sjoland, "Noise optimization of an inductively degenerated CMOS low noise amplifier," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 9, pp. 835–841, 2001.