

Hindawi Publishing Corporation  
EURASIP Journal on Advances in Signal Processing  
Volume 2010, Article ID 513610, 11 pages  
doi:10.1155/2010/513610

## Research Article

# An Evolutionary Algorithm for Improved Diversity in DSL Spectrum Balancing Solutions

Johelden Bezerra,<sup>1,2</sup> Aldebaro Klautau,<sup>1</sup> Marcio Monteiro,<sup>1</sup> Evaldo Pelaes,<sup>1</sup> Eduardo Medeiros,<sup>1</sup> and Boris Dortschy<sup>3</sup>

<sup>1</sup>Signal Processing Laboratory (LaPS), Federal University of Para (UFPA), Belem, PA, Brazil

<sup>2</sup>Departamento de Ensino de Informação e Comunicação (DIC), Federal Institute of Para (IFPA), Belem, PA, Brazil

<sup>3</sup>Broadband Access Research Laboratory, Ericsson AB, Kista, stockholm, Sweden

Correspondence should be addressed to Johelden Bezerra, johelden@gmail.com

Received 4 December 2009; Revised 26 March 2010; Accepted 14 April 2010

Academic Editor: George Tombras

Copyright © 2010 Johelden Bezerra et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There are many spectrum balancing algorithms to combat the deleterious impact of crosstalk interference in digital subscriber lines (DSL) networks. These algorithms aim to find a unique operating point by optimizing the power spectral densities (PSDs) of the modems. Typically, the figure of merit of this optimization is the bit rate, power consumption or margin. This work poses and solves a different problem: instead of providing the solution for one specific operation point, it finds a set of operating points, each one corresponding to a distinct matrix with PSDs. This solution is useful for planning DSL deployment, for example, helping operators to conveniently evaluate their network capabilities and better plan their usage. The proposed method is based on a multiobjective formulation and implemented as an evolutionary genetic algorithm. Simulation results show that this algorithm achieves a better diversity among the operating points with lower computational cost when compared to an alternative approach.

## 1. Introduction

Digital subscriber line (DSL) is a technology that exploits existing telephony metallic loop plants for delivering triple-play services (voice, video, and high-speed Internet access). In DSL networks, *crosstalk* interference is a major impairment for improving data rate and reach of the lines. This interference results from the electromagnetic coupling among different copper pairs in the same binder [1]. Aiming to minimize the crosstalk effects, one efficient alternative is to use *spectrum balancing* (SB) algorithms [2–6], which optimize the power spectrum density (PSD) masks of the transmitters such that an appropriate operating point is used.

Spectrum balancing is intimately related to *dynamic spectrum management* (DSM). In contrast to static spectrum management (SSM) [7], which typically adopts the same PSD mask for all modems, in a DSM scenario the PSDs can be made adaptive and change depending on the line condition observed by a modem as example. There are many practical issues to impose a modem's PSD, and the term DSM

is sometimes applied to the overall procedure of changing PSDs, while SB denotes the stage of this procedure that finds the target PSDs. In some works the terms DSM and SB are used interchangeably.

When compared to the nonoptimized “worst-case” SSM solutions [8], SB can lead to substantial improvements in rate and reach. This improvement is prominent, for example, in “near-far” scenarios, some (near) users receive a strong DSL signal when compared to others (far users) because the former users have relatively high gains in their direct channels [8].

Instead of dealing with the conventional SB problem, this work solves a different problem: finding a set of operating points that are both Pareto optimal and *diverse*. Roughly speaking, a set is considered diverse if it properly describes the variety of possible solutions (formal definitions will be discussed later on). The requirement of diversity is important because, in an extreme case, all elements of the solution set could be clustered very close to each other, providing in practice few (different) options of operation points. The new

problem is called the *diverse set spectrum balancing* (DSSB). It is of practical interest, for example, from the perspective of a DSL system operator at the stage of planning the deployment of DSL service. This work proposes a solution to the DSSB problem, which is implemented as an evolutionary genetic algorithm (GA) [9] and based on a multiobjective optimization.

This work is organized as follows. Section 2 describes the multi-objective formulation and provides an overview of the diverse set spectrum balancing problem. Section 3 presents the proposed method to solve the DSSB problem. Section 4 concerns the simulation results obtained by the proposed method, and the conclusions are presented in Section 5.

## 2. Problem Description

This section describes the DSSB problem. First, it is useful to redefine the SB problem according to a multi-objective formulation. This formulation departs from the typically adopted single-objective optimization in the DSM literature [2–6]. It is convenient, for example, because it naturally incorporates the concept of *Pareto optimality* [9].

**2.1. Multi-Objective Spectrum Balancing.** It is assumed that the DSL line code is discrete multitone (DMT) [1]. The PSD masks of all users are organized in a matrix  $\mathbf{S} = \{s_n^k\}$  of dimension  $N \times K$ , where  $N$  is the number of users,  $K$  is the number of subchannels (tones), and the element  $s_n^k$  represents the PSD value of user  $n$  at tone  $k$ . The channel gains (direct and crosstalk) are represented by a three-dimensional array  $\overline{\mathbf{G}}$  of dimension  $N \times N \times K$ . For each tone  $k$ , the matrix  $\{g_{i,j}^k\}$  is the squared magnitude of the transfer function corresponding to the channel from user  $i$  to  $j$ . To determine  $\overline{\mathbf{G}}$  in practical scenarios, there are procedures to estimate the direct channel  $g_{i,i}^k, \forall k$  of the  $i$ th user [10, 11] and the far-end crosstalk (FEXT) channels  $g_{i,j}^k, \forall k, i \neq j$  [12]. It is assumed that in frequency-division duplex (FDD), the down- and upstream transmissions reside in nonoverlapping frequency bands. This way, the near-end crosstalk (NEXT) can be neglected [1]. Noise sources other than the crosstalk among the  $N$  users are described by a matrix  $\mathbf{N} = \{\sigma_n^k\}$  of dimension  $N \times K$ . The element  $\sigma_n^k$  is the value of the noise PSD at the receiver  $n$  at tone  $k$ , representing the combined effect of DSL impairments [13] like, for example, background noise, radio-frequency interference, and alien crosstalk.

Hence, the channel information (CI) is assumed to be described by  $(\mathbf{N}, \overline{\mathbf{G}})$ . An *operating point*  $(\mathbf{S}, \mathbf{N}, \overline{\mathbf{G}})$  is completely described by the PSDs of all users  $\mathbf{S}$  and the CI of the given network. Given  $(\mathbf{S}, \mathbf{N}, \overline{\mathbf{G}})$ , one can calculate [1], for example, the rate  $r_n = \mathcal{R}(\mathbf{S}, \mathbf{N}, \overline{\mathbf{G}})$  and total transmit power  $p_n = \mathcal{P}(\mathbf{S}, \mathbf{N}, \overline{\mathbf{G}})$  of the  $n$ th user, where  $\mathcal{R}$  and  $\mathcal{P}$  are the corresponding (many-to-one) mappings. For example, assuming standard arguments [1], the bit rate of the  $n$ th user is

$$r_n = F_s \sum_{k=1}^K \log_2 \left( 1 + \frac{g_{n,n}^k s_n^k}{\sigma_n^k + \sum_{m=1, m \neq n}^N g_{m,n}^k s_m^k} \right) \quad (1)$$

bits per second, where  $F_s$  is the symbol rate (typically  $F_s = 4$  kHz in DSL systems).

For convenience, let  $\overline{\mathbf{S}}_Z = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_Z\}$  denote  $Z$  matrices of dimension  $N \times K$  corresponding to PSDs, with  $\overline{\mathbf{S}}_\infty$  representing the case in which  $Z \rightarrow \infty$  or  $Z$  is unknown. The *multi-objective SB problem* is slightly different from the conventional single-objective SB problem that is discussed, for example, in [8]. The multi-objective version corresponds to searching the Pareto optimal PSDs  $\overline{\mathbf{S}}_\infty^\dagger$  according to

$$\begin{aligned} \overline{\mathbf{S}}_\infty^\dagger &= \arg \max_{\mathbf{S}} \mathcal{O}_i(\mathbf{S}, \mathbf{N}, \overline{\mathbf{G}}), \quad i = 1, \dots, L, \\ \text{such that } & q_j(\mathbf{S}) \leq 0, \quad j = 1, \dots, Q, \\ & t_m(\mathbf{S}) = 0, \quad m = 1, \dots, T, \\ & 0 \leq s_n^k \leq M_k, \quad k = 1, \dots, K, \end{aligned} \quad (2)$$

where  $\mathcal{O}_i$  (the arguments are omitted for simplicity) is the  $i$ th objective function. The functions  $q_j$  and  $t_m$  are, respectively, inequality and equality constraints imposed to the solution (to allow imposing the maximum power or minimum bit rate per user, e.g.). The values  $M_k$  correspond to a limiting PSD mask typically imposed by DSL standards and hardware limitations.

Each objective function  $\mathcal{O}_i$  could be eventually maximized or minimized (such as maximizing the bit rate or minimizing the total power consumption), but minimizing a function  $\mathcal{O}_i$  is equivalent to maximizing  $-\mathcal{O}_i$ , and having only maximizations is convenient. A value function can be also used. For example, if the user has prices defined for specific rates, a value function can be set to maximize profit.

Some extra definitions are useful to further discuss (2). A solution  $\mathbf{S}$  is feasible if the constraints in (2) are satisfied. The set of all feasible solutions is the *feasible region* of the *search space*  $S$ . The  $L$ -dimensional vector  $\mathbf{O} = (\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_L)$  with the values of all objective functions is located in a multidimensional space called *objectives space*. This is a fundamental distinction with respect to a single-objective optimization problem, which has a one-dimensional objectives space. Instead of seeking a unique solution to the problem, which is typically the case in conventional single-objective SB optimizations, the solution to the multi-objective SB problem is a (possibly infinite) set  $\overline{\mathbf{S}}_\infty^\dagger$  of *Pareto points*.

The solutions of (2) are called *Pareto optimal* because an improvement in one objective can only occur with the worsening of at least one other objective [9]. More formally,  $\mathbf{S}^*$  is Pareto optimal if there does not exist another  $\mathbf{S}$  that dominates  $\mathbf{S}^*$  (i.e.,  $\mathcal{O}_i(\mathbf{S}) \leq \mathcal{O}_i(\mathbf{S}^*) \forall i = 1, \dots, L$ ), and the solution  $\mathbf{S}^*$  is strictly better than  $\mathbf{S}$  in at least one objective (i.e.,  $\mathcal{O}_j(\mathbf{S}) < \mathcal{O}_j(\mathbf{S}^*)$  for at least one  $j = 1, \dots, L$ ). For a given system, the *Pareto frontier* is the set of feasible solutions that are all Pareto optimal.

Hereafter, without loss of generality, it is convenient to focus on the *rate-adaptive* (RA) version of the SB problem, where the goal is to maximize the bit rate [8]. The proposed solution is not restricted to the RA setup and is also valid, for example, when the goal is to minimize power [6].

A Pareto optimal operating point in the rate-adaptive version of (2) must belong to the Pareto frontier corresponding to  $L = N$  and  $\mathcal{O}_n = r_n$ . Besides, there are  $Q = N$  inequalities and  $q_n = p_n - P_n^{\max} \leq 0, \forall n$ , which limits the power per user  $p_n$  to a maximum value  $P_n^{\max}$ . There are no equality constraints ( $T = 0$ ) and no PSD masks limitation, that is,  $M_k = \infty$ . In summary, the solution of the RA version of the multi-objective SB is

$$\bar{\mathbf{S}}_\infty^\dagger = \arg \max_{\mathbf{S}} r_n, \quad n = 1, \dots, N, \quad (3)$$

$$\text{such that } p_n \leq P_n^{\max}, \quad n = 1, \dots, N.$$

The Pareto frontier of the optimization in (3) is called *rate region* because it characterizes all Pareto optimal data rate combinations among users.

**2.2. The Diverse Set Spectrum Balancing Problem.** Assuming that  $\mathcal{D}$  is the adopted definition of diversity and one is interested in obtaining  $Z$  operating points, the *DSSB problem* corresponds to searching for the best  $\bar{\mathbf{S}}_Z^\dagger$  according to

$$\bar{\mathbf{S}}_Z^\dagger = \arg \max_{\bar{\mathbf{S}}_Z} \mathcal{D}(\bar{\mathbf{S}}_Z), \quad (4)$$

where  $\bar{\mathbf{S}}_Z$  is restricted to contain only *Pareto optimal* solutions according to (2) or, alternatively, special cases such as (3). It remains to formally define  $\mathcal{D}$ .

There are several alternatives to define the diversity of a set of Pareto optimal solutions. One relevant example is *spacing* [9]. To simplify its description, let  $O_i^z = \mathcal{O}_i(\mathbf{S}_z, \mathbf{N}, \mathbf{G})$  denote the value of the  $i$ th objective function for the  $z$ th matrix  $\mathbf{S}_z$  with PSDs. The spacing diversity is given by

$$\mathcal{D}(\bar{\mathbf{S}}_Z) = \left( \frac{1}{Z} \sum_{z=1}^Z (d_z - \bar{d})^2 \right)^{-1/2}, \quad (5)$$

where

$$d_z = \min_{m=1, \dots, Z, m \neq z} \sum_{i=1}^L |O_i^z - O_i^m|, \quad (6)$$

and  $\bar{d} = 1/Z \sum_{z=1}^Z d_z$ . This metric corresponds to the inverse of the standard deviation of  $d_z$ , and  $d_z$  represents the smallest value of the sum of the absolute difference between two solution points in the objectives space. In other words,  $d_z$  is large when there are no solutions too close to the respective  $z$ th solution. The intuition is that a set  $\bar{\mathbf{S}}_Z$  is diverse when the solutions have similar values of  $d_z$ . As an extreme case, if all  $Z$  solutions had the same value  $d'_z$ , their corresponding standard deviation would be zero and  $\mathcal{D}(\bar{\mathbf{S}}_Z) = \infty$ .

An example helps discussing the spacing measure. Table 1 lists two different sets of solutions  $\bar{\mathbf{S}}_Z^a$  and  $\bar{\mathbf{S}}_Z^b$  that will be compared. Both assume a two-dimensional objectives space ( $L = 2$ ) and have  $Z = 4$  solutions. These solutions are represented in Figure 1. Assume the first set of solutions  $\bar{\mathbf{S}}_Z^a$  in Table 1 and the goal is to obtain  $d_1$ : the first solution candidate  $m = 2$  to minimize the accumulated differences

TABLE 1: Two different sets of solutions  $\bar{\mathbf{S}}_Z^a$  and  $\bar{\mathbf{S}}_Z^b$  for which  $L = 2$  and  $Z = 4$ . They are displayed in Figure 1.

z	$\bar{\mathbf{S}}_Z^a$		$\bar{\mathbf{S}}_Z^b$	
	$O_1$	$O_2$	$O_1$	$O_2$
1	0	1	0.57	0.82
2	0.51	0.85	0.67	0.75
3	0.88	0.47	0.78	0.63
4	1	0	0.82	0.57

in (6) is  $|0 - 0.51| + |1 - 0.85| = 0.66$ . For  $m = 3$ , the accumulated sum is  $0.88 + 0.53 = 1.41$  and for  $m = 4$  it is 2. Hence,  $d_1 = 0.66$  because it is the minimum value among 0.66, 1.41 and 2. Similarly,  $d_2 = 0.66$ ,  $d_3 = 0.59$  and  $d_4 = 0.59$ . The standard deviation of these  $d_z$  values is  $\sigma = 0.035$ , which leads to  $\mathcal{D}(\bar{\mathbf{S}}_Z^a) = 1/\sigma \approx 28.6$ . The second set of solutions leads to the following values of  $d_z = 0.17, 0.17, 0.1, 0.1$  which also correspond to  $\mathcal{D}(\bar{\mathbf{S}}_Z^b) = 1/0.035 \approx 28.6$ . From this comparison, it can be noted that the spacing metric does not take into account the spread of the solutions. There are other definitions of diversity that try to combat this limitation of spacing [9]. However, for the application proposed in this work, a new one is adopted.

The diversity definition adopted in this work is based on computational geometry [14] and can be explained with the following elements. Assume that the  $Z$  points are the vertices of an  $L$ -dimensional convex *polytope*, which has  $(L-1)$ -dimensional *facets*. These facets are themselves polytopes, whose facets are  $(L-2)$ -dimensional *ridges* (also called *subfacets*) of the original polytope. Ridges are once again polytopes whose facets give rise to  $(L-3)$ -dimensional boundaries of the original polytope, and so on. For example, when  $L = 2$  as in Figure 1, a facet is an edge, that is, a line segment (the reader may observe a case of  $L = 3$  jumping to Figure 8, where a facet is a polygon).

The proposed method computes the facet areas  $A_i, i = 1, \dots, U$  of the polytope using an algorithm such as Quickhull [15] and, after that, the standard deviation  $\sigma_A = \sqrt{(1/U) \sum_{i=1}^U (A_i - \mu_A)^2}$  of these areas, where  $\mu_A = (1/U) \sum_{i=1}^U A_i$ , is the empirical mean. The *polytope diversity* is then calculated as

$$\mathcal{D}(\bar{\mathbf{S}}_Z) = \frac{1}{\sigma_A}. \quad (7)$$

Other heuristics were evaluated for expressing diversity based on the facet areas, but (7) is simple and conducted to good results. It is similar to (5) in the fact that it is the inverse of an empirical standard deviation but uses the facet areas instead of  $d_z$ . Equation (5) is simpler to compute than (7) but, taking in account the role of  $d_z$  and  $A_i$  in the problem geometry, one can observe that the facet areas lead to a more adequate definition of diversity for the DSSB problem.

There are still two important details to make (7) work effectively for the DSSB problem. The first one is the inclusion of *shaping points*. For example, in the case of  $\bar{\mathbf{S}}_Z^b$

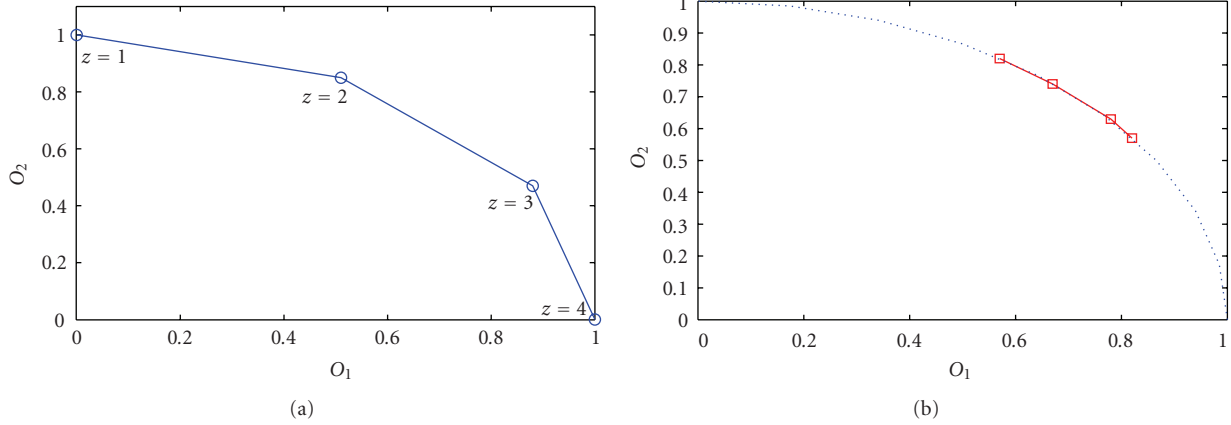


FIGURE 1: Example of two sets of solutions  $\bar{\mathbf{S}}_Z^a$  and  $\bar{\mathbf{S}}_Z^b$  (see Table 1) that have the same diversity of 28.6 according to the spacing metric of (5).

in Figure 1, a facet would connect points  $z = 1$  and 4 with an area of  $\sqrt{(0.57 - 0.82)^2 + (0.82 - 0.57)^2} = 0.3536$  (in this 2-dimensional case, the facet area corresponds to the length of the line segment). To avoid this kind of facet and promote the spread of the solutions over the objectives space,  $L + 1$  shaping points are always added to the  $Z$  points in  $\bar{\mathbf{S}}_Z$  before obtaining its convex hull. The first point is located at the origin  $(0, 0, \dots, 0)$ , and the others have only the  $i$ th element different from zero,  $i = 1, \dots, L$ . For example, when solving (3), the shaping point  $(r_1^{\max}, 0, 0, \dots, 0)$  is included, where  $r_1^{\max}$  is the rate obtained by user 1 when all other users are inactive. The rate  $r_2^{\max}$  of the second user imposes another shaping point  $(0, r_2^{\max}, 0, \dots, 0)$  and so on. After calculating the convex hull with the shaping points, all facet areas for which all vertices are shaping points are excluded from the calculation of  $\sigma_A$ .

An example using  $\bar{\mathbf{S}}_Z^b$  in Figure 1 is useful to understand the calculation of the polytope diversity of (7) and the role of the shaping points. Let us assume that  $r_1^{\max} = r_2^{\max} = 1$  such that the three shaping points are  $(0,0)$ ,  $(1,0)$ , and  $(0,1)$ . Note that all 7 points (the original four in  $\bar{\mathbf{S}}_Z^b$  and three shaping points) are in the convex hull of the new polytope. The inclusion of the shaping points increases the number of facets from 4 to 7, and their areas are 1, 1, 0.5977, 0.5977, 0.1221, 0.1628, and 0.0721. All vertices of the first two facets (with area equal to one) are shaping points and these facets are excluded. The standard deviation of the remaining five facets is  $\sigma_A = 0.2363$ , which leads to  $\mathcal{D}(\bar{\mathbf{S}}_Z^b) = 4.23$  using (7).

The second detail for making the diversity measure adequate for practical use in the DSSB problem is to exclude the solutions in  $\bar{\mathbf{S}}_Z$  that are not in the Pareto front, that is, in the convex hull of the polytope. This is important because, as it will be discussed later, there is a need for using suboptimal SB algorithms, and these algorithms may return a set  $\bar{\mathbf{S}}_Z$  that has only  $C < Z$  solutions in the Pareto front. Strictly, according to (2), if a solution is not in the Pareto front, it should not be part of the set  $\bar{\mathbf{S}}_Z$ . However, this is a consequence of adopting a suboptimal SB algorithm for decreasing the computational cost. Because of

that, when used in the DSSB problem, the (7) metric should be modified to penalize the existence of  $Z - C$  solutions that are not in the Pareto front. Other heuristics were tested, but a simple one obtained good results: first obtain  $\sigma'_A$ , the standard deviation of areas corresponding to the facets of the  $C$  vertices effectively in the convex hull (i.e., discard the solutions in  $\bar{\mathbf{S}}_Z$  that are not part of the convex hull), and divide  $\sigma'_A$  by the fraction  $C/Z$  of Pareto solutions. According to this procedure, the value of  $\sigma_A$  to be used in (7) is obtained by

$$\sigma_A = \left(\frac{Z}{C}\right)\sigma'_A. \quad (8)$$

The smaller the fraction  $C/Z$ , the more the diversity of (7) is penalized. For example, assume that a point  $(0.3, 0.3)$  not belonging to the convex hull is included in  $\bar{\mathbf{S}}_Z^b$  of Figure 1 to create a new set  $\bar{\mathbf{S}}_Z^c$  with  $Z = 5$ . The updated value for the diversity would be  $\mathcal{D}(\bar{\mathbf{S}}_Z^c) = (4/5)4.23 = 3.386$ . In the case of  $\bar{\mathbf{S}}_Z^a$ ,  $C = Z = 4$ , the areas are 0.5316, 0.4851, and 0.5304 and  $\sigma_A = \sigma'_A = 0.0216$ ; hence  $\mathcal{D}(\bar{\mathbf{S}}_Z^a) = 46.2$ , which is larger than  $\mathcal{D}(\bar{\mathbf{S}}_Z^b)$ , as desired. Now, counting with a more adequate definition of diversity, the next section focuses on the proposed algorithm. It should be noted that (7) will be used to evaluate the final solution set obtained by the proposed and baseline algorithms.

### 3. The Proposed Solution

The DSSB problem of (4) is rather complicated to solve and a brute force approach is unfeasible due to its high computational cost. In this work, a multi-objective evolutionary algorithm (MOEA) guides the search.

Most evolutionary algorithms, such as genetic algorithms, are very powerful especially because they are “model-free”, not requiring convexity or knowing derivatives, for example, [9]. However, they may have slow convergence when compared to algorithms that benefit from knowledge about the specific problem. To solve the DSSB problem (4), a search based only on pure evolutionary computing seems

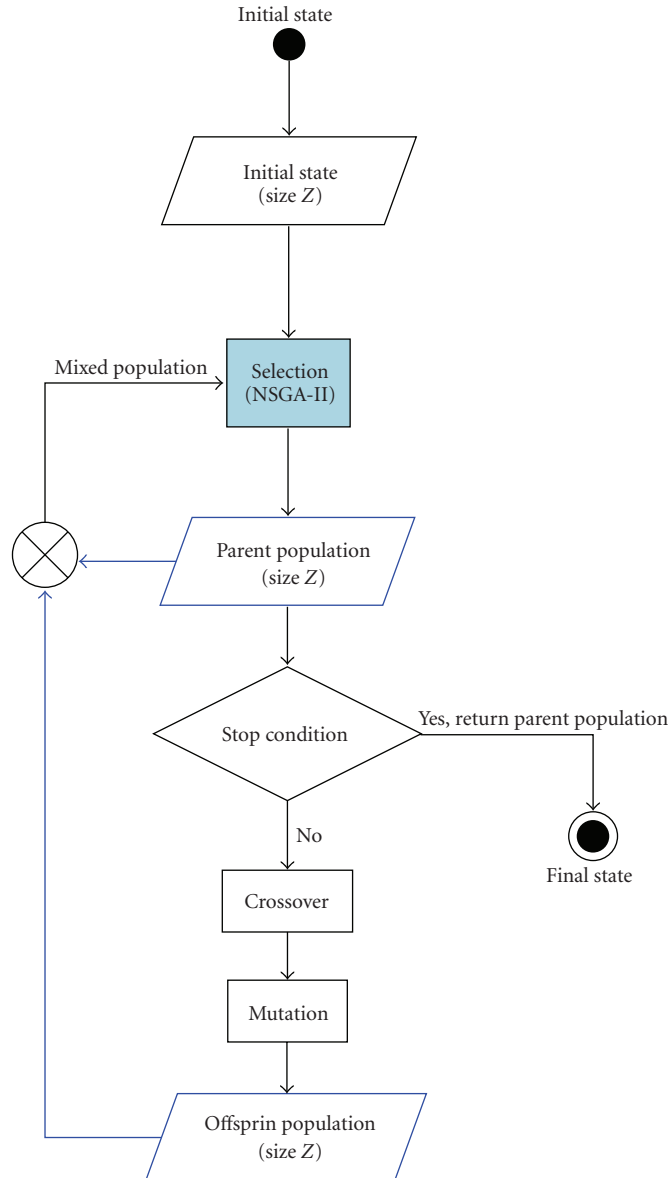


FIGURE 2: Proposed evolutionary optimization to solve the DSSB problem.

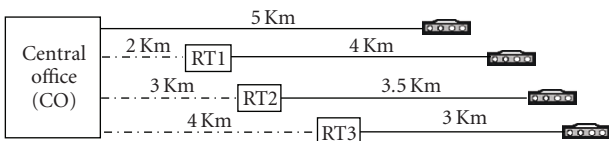


FIGURE 3: DSL “master” scenario for all simulations.

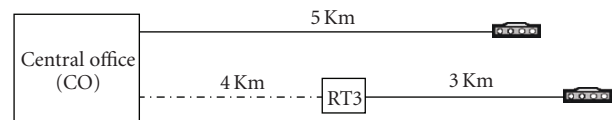


FIGURE 4: DSL scenario with 2-user ( $N = L = 2$ ).

again unfeasible. Therefore, this work advocates the adoption of a *hybrid approach*, where the evolutionary algorithm guides the search for a Pareto-optimal and diverse set of points, while a local search based on an SB algorithm provides the PSDs upon request of the MOEA.

This hybrid approach allows for any multi-objective evolutionary and SB algorithms, given that they properly

interface. Obviously, the characteristics of the overall procedure, such as its optimality and computational cost, depend on this combination of algorithms. This work explores the following combination, which is called the *diverseSB* algorithm: the MOEA is the nondominated sorting genetic algorithm (NSGA-II) [16] and the SB algorithm is assumed to adopt the weighted-sum formulation, such as iterative

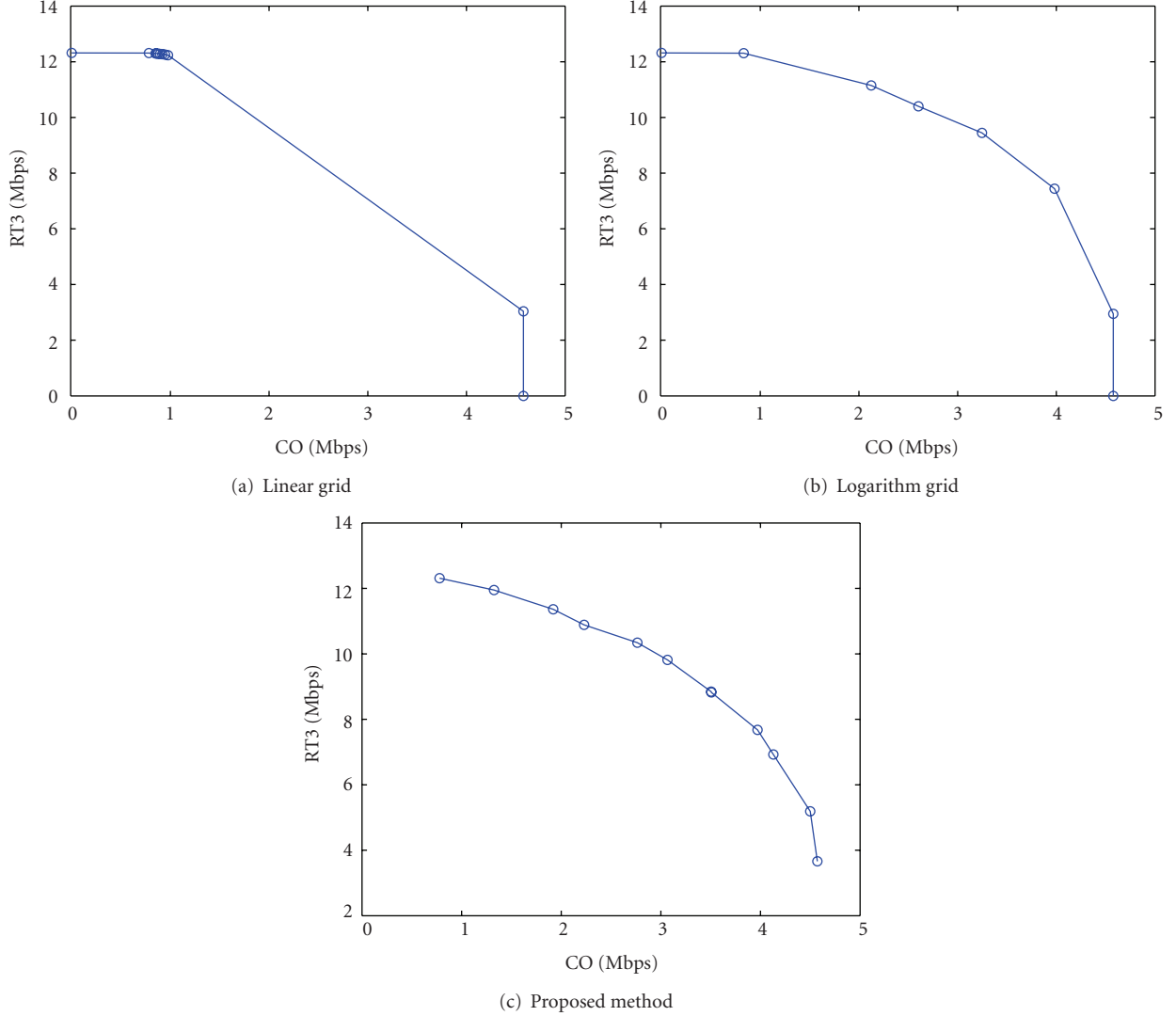


FIGURE 5: Operating points using (a) linear grid, (b) logarithm grid and (c) proposed *diverseSB* for the 2-user case. All three methods used 144 SB runs and provided  $Z = 12$  points (some points cannot be distinguished visually, such as one pair in (c)). Note that, in this case, the linear grid was clearly outperformed by the other two methods, while the proposed method is slightly better than the logarithm grid.

SB (ISB) [3] and Successive Convex Approximation for Low Complexity (SCALE) [4]. The following subsection describes the motivation for using weighted-sum SB in the context of rate-adaptive algorithms, and then the proposed algorithm is discussed.

**3.1. Weighted-Sum Rate-Adaptive SB Algorithms.** Instead of solving (2), conventional SB algorithms solve the single-objective version of the problem. The approach is based on the fact that a multi-objective optimization can be converted into a single-objective problem by constructing an *aggregate objective function* (AOF) [9]. The AOF combines all objective functions into a single functional form. A possible alternative is the “weighted linear sum” of the objectives or *weightedsum*, which is widely adopted in SB (see, e.g., [17]). In this case, scalar weights  $\omega_i$  can be specified for each objective  $\mathcal{O}_i$  to be optimized, and then the weighted sum

of objectives is combined into a function that can be solved by any single-objective optimizer. The obtained solution will depend on the values or, more precisely, the relative values, of the specified weights, which prioritize the objectives.

The conventional weighted-sum rate-adaptive (WSRA) SB algorithms [2, 4, 5, 17, 18] adopt

$$O = \omega_1 r_1 + \dots + \omega_N r_N \quad (9)$$

as the AOF and solve

$$\mathbf{S}^* = \arg \max_{\mathbf{S}} \sum_{n=1}^N \omega_n r_n \quad (10)$$

$$\text{such that } p_n \leq P_n^{\max}, \quad n = 1, \dots, N,$$

where  $\mathbf{S}^*$  is a single  $N \times K$  matrix with PSDs (corresponding to one operating point). It should be noted that  $\mathbf{S}^*$  may

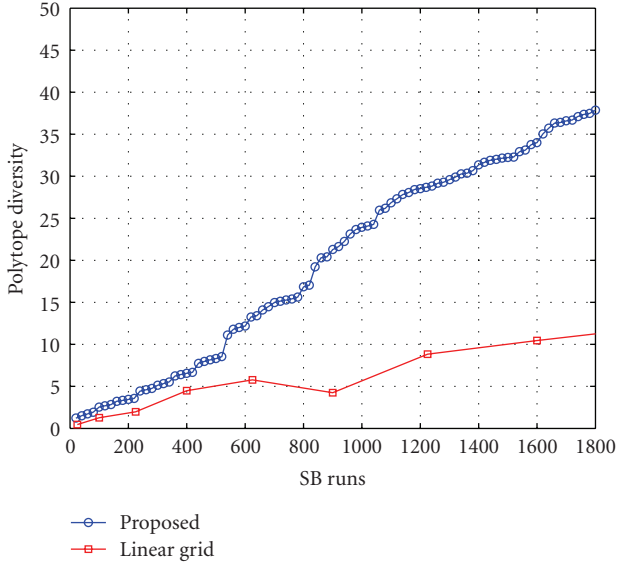


FIGURE 6: Diversity for different SB runs for the 2-user scenario.

not be in the Pareto front of (3), that is, maybe  $\mathbf{S}^* \notin \bar{\mathbf{S}}_\infty^\dagger$ . The reason is that most SB algorithms cannot guarantee optimality nor convergence, in spite of achieving good solutions in practice. The optimal SB (OSB) algorithm [19] is too slow for practical use.

The proposed *diverseSB* assumes the availability of a WSRA SB algorithm that attempts to solve (10), providing PSDs (eventually suboptimal) for one operating point, given the CI and  $N$  weights. In practice, one could be concerned with only  $N - 1$  weights because the sum of all weights can be restricted to one, that is,  $\sum_{n=1}^N \omega_n = 1$ , or another constant.

It should be noted that a relatively straightforward solution to find operating points according to (10) is to systematically vary the values of  $N$  weights, invoke the WSRA SB algorithm for each weight combination, and find distinct operating points by “navigating” through the rate region. However, the mapping between weights and objective functions is typically nonlinear and it is nontrivial to find a diverse set of optimal solutions by discretizing the space  $\mathbb{R}^N$  spanned by the weights (i.e., using a grid in the  $\mathbb{R}^N$  space). A linear or uniform *grid* consists of the Cartesian product of the  $N$  axes  $\{0, \Delta, 2\Delta, \dots, (G - 1)\Delta\}$ , where  $G$  is the number of points in each axis, and  $\Delta$  is the step size. Nonuniform grids can also be used, for example, organizing it according to a logarithmic scale. However, the computational cost for obtaining a diverse set using a grid is typically very high. “For example, as will be detailed later, a brute force implementation for  $N = 25$  users would require 27 days using a modern personal computer (this estimate assumes the SCALE algorithm and a grid with  $G = 20$  points per user, reducing the resolution to  $G = 10$  still corresponds to 40 hours.)” An alternative method is described in the sequel.

**3.2. The Proposed *diverseSB* Algorithm.** This subsection presents an efficient solution to the DSSB problem. The exponential cost of the grid search is alleviated by the hybrid

approach, in which the NSGA-II algorithm guides the search for a Pareto optimal and diverse set of points, while a local search based on a weighted-sum SB algorithm (e.g., SCALE or IWF) provides the PSDs for a given set of weights.

The NSGA-II algorithm keeps populations of individuals. Following the principles of natural evolution, the optimizer iteratively improves the population, generating new individuals according to mechanisms such as *mutation*, *crossover* and *elitism* [9].

In the evolutionary algorithm, each individual (also called *chromosome*)  $\mathbf{x}$  represents a set of  $N$  weights  $\omega_n$  in (10), with  $\omega_n \in [0; 1]$ . The population size is  $Z$ , the number of operating points specified by the user. The optimizer seeks solutions in the Pareto frontier. For that, it needs to calculate the vector  $\mathbf{O} = (r_1; r_2; \dots; r_N)$  with all objective functions in (10). This calculation requires invoking the adopted SB algorithm, which receives the weights represented by  $\mathbf{x}$  and the CI ( $\mathbf{N}, \bar{\mathbf{G}}$ ) as inputs and outputs the PSDs  $\mathbf{S}$ . This step corresponds to obtaining an operating point, which brings enough information to calculate all objective functions. The evolutionary algorithm can then improve the current population until convergence or reaching a pre-specified maximum number of iterations. The final result is a set  $\bar{\mathbf{S}}_Z$  of  $Z$  operating points. Figure 2 presents the flowchart of the *diverseSB* algorithm.

It should be noted that (7) is used to evaluate the final solution set. It is not used in the iterations of the *diverseSB* algorithm mainly to avoid the computational cost of computing the convex hull. Another reason is that a conventional MOEA requires only a metric of individual diversity (the crowding distance is used in this work) and would require modifications to take in account the overall set diversity.

The algorithm begins by creating a random population  $\mathbf{P}_1$  (where the subscript denotes the generation counter) with  $Z$  individuals, each one encoding a set of  $N$  weights  $\omega_n \in [0, 1]$ . These weights can be obtained, for example, from a uniform random distribution. Then, the first population  $\mathbf{P}_1$  is submitted to ordinary selection, crossover, and mutation operators to generate the first offspring  $\mathbf{P}_2$  (the second generation). After that, from the population  $\mathbf{P}_1 \cup \mathbf{P}_2$  (elitism),  $Z$  individuals are selected for the next generation  $\mathbf{P}_3$ , taking in account *dominance* and *individual diversity* metrics to rank the individuals and select the fittest (more details about the concepts of elitism, dominance and individual diversity in GAs can be found, e.g., in [9, 16]). This process is repeated until some stop criterion is met (e.g., maximum number  $I_{\max}$  of generations). An algorithmic description of the procedure is presented in Algorithm 1.

It should be noted that in this context, the individual diversity measure aims at helping to rank each solution point  $\mathbf{S}$  (PSD matrix), while the spacing and polytope diversity measures discussed in Section 2.2 are used to evaluate the whole set of points  $\bar{\mathbf{S}}$ . The individual diversity measure used in Algorithm 1 was the *crowding distance* [16, 20]. The crowding distance is calculated by the perimeter of the largest cuboid enclosing each point in the objectives space without including any other point. After normalizing the crowding distance for each nondominated solution, they are sorted in

**Input**

- (a)  $Z$ —desired number of operating points;
- (b)  $sba$ —a weighted-sum spectrum balancing algorithm;
- (c)  $moea$ —a multi-objective evolutionary algorithm;
- (d) Set of objective functions and constraints;
- (e)  $(N, \bar{G})$ —channel information (CI).

**Output:**  $\bar{S}_Z^\dagger$ —PSDs

- (1) Create an initial population  $P_1$  with  $Z$  individuals.
- (2)  $t \leftarrow 1$
- (3) **repeat**
- (4)     Execute  $sba$  for each individual  $x_i$  to obtain the respective  $N \times K$  matrix with PSDs  $S_i$ ;
- (5)     Calculate the objective functions for all individuals;
- (6)     Execute  $moea$  to improve the current  $P_t$ , creating a new population  $P_{t+1}$  of  $Z$  individuals, with improved objective functions and diversity.
- (7)      $t \leftarrow t + 1$
- (8) **until** convergence;
- (9) Convert the final population  $P^\dagger$  of weights into the corresponding set  $\bar{S}_Z^\dagger$  of PSD matrices
- (10) **return**  $\bar{S}_Z^\dagger$ ;

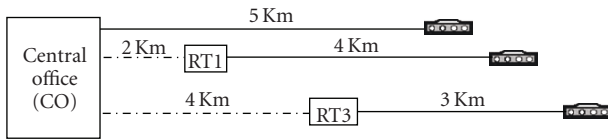
ALGORITHM 1: *diverseSB*.

FIGURE 7: DSL scenario with 3-user.

ascending order and used in step 6 of Algorithm 1. The next section presents results obtained by the proposed method and a comparison with baseline alternatives.

#### 4. Simulations Results

This section presents simulation results to assess the performance of the proposed *diverseSB* algorithm. These results are compared with two baselines, which correspond to the adoption of a uniform and a logarithm-spaced grids.

The simulations used DSL scenarios obtained from the “master” scenario in Figure 3. This scenario consists of a central office (CO) and three remote terminals (RTs) transmitting in downstream direction. The simulations were organized in three: 2-user simulations (involving the CO and RT3), 3-user simulations (CO, RT1, and RT3) and 4-user simulations (CO, RT1, RT2 and RT3).

All transceivers are assumed to be ADSL units. The cable model used was ANSI TP2 24 AWG for all lines. The *diverseSB* algorithm used a probability of crossover of 0.7 and probability of mutation of 0.3. The adopted spectrum balancing algorithm was SCALE [4]. The software Qhull [21] was adopted for calculating the facet areas when using (7).

The performance metric to compare the computational cost between the proposed method and grid searches will be the number of *SB runs*, that is, the number of times that the spectrum balancing algorithm was executed. This assumption is sensible because the cost of other routines,

such as choosing the weights, can be neglected when compared to the time required to execute an SB algorithm. For example, when using a grid, the weights vector is simply updated within the nested loops used to generate the grid. Excluding the time invested in SB runs, the computational overhead of the search using an MOEA is higher than using a grid but this overhead still takes less than 1% of the total execution time, which is dominated by the number  $B$  of SB runs and denoted as  $\mathcal{O}(B \times SB_{\text{complexity}})$ , where  $SB_{\text{complexity}}$  represents the complexity of the adopted SB algorithm.

In spite of both grid and MOEA algorithms having computational costs that depend linearly on the number  $B$  of SB runs, they differ in the parameters that  $B$  depends on. When using a  $N$ -dimensional grid and assuming the same number  $G$  of points in all Cartesian axes, the number of SB runs is  $B = G^N$ . For the MOEA,  $B$  depends on the population size  $Z$  and maximum number  $I_{\text{max}}$  of iterations and is given by  $B = I_{\text{max}}Z$ . To provide fair comparisons, the simulations in this work compared algorithms using the same number  $B$  of SB runs, such that their execution times were equivalent. The next subsection provides results for  $N = 2$  users.

**4.1. 2-User Simulations.** The scenario for the 2-user simulations, obtained from Figure 3, is depicted in Figure 4 for convenience.

The simulations conducted for this scenario used the same number of SB runs ( $S = 144$ ) and the same number of final solutions ( $Z = 12$ ) for all methods. For the linear grid search, a grid with  $G = 12$  points was used (i.e., a  $12 \times 12$  grid). In order to have a fair comparison, from all 144 operating points obtained by the grid search, a diverse set of 12 was selected using the crowding distance, the same measure of individual diversity used along the iterations of *diverseSB*. A similar procedure was adopted for a logarithm grid in the weights space. For the proposed *diverseSB* method, simulations were performed with a population size



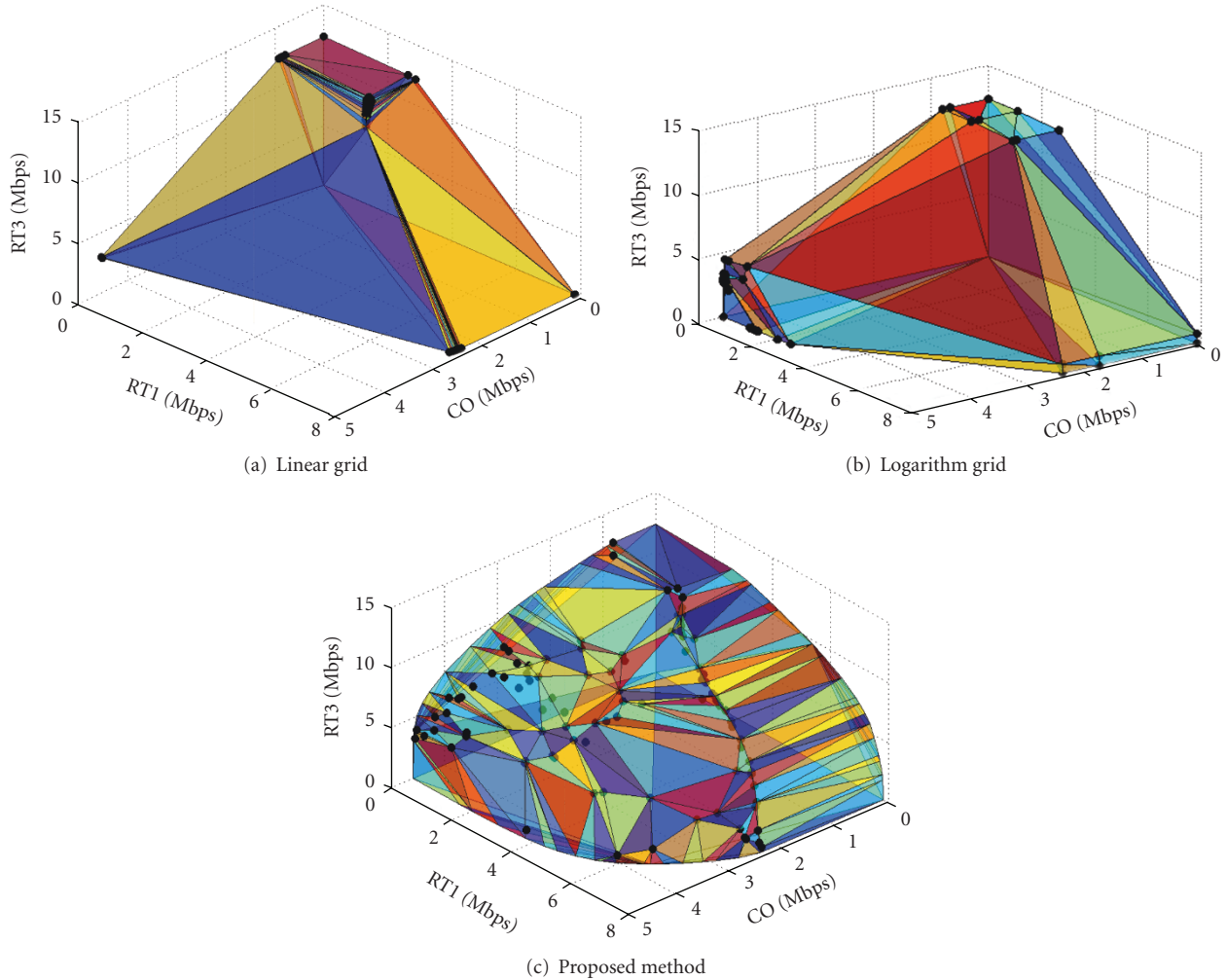


FIGURE 8: Results for the three-user case: (a) linear grid, (b) logarithm grid and (c) proposed *diverseSB*. All three methods used 1,000 SB runs and provided  $Z = 100$  points. The shaping points are used for better visualization. Note that, in the 3D case, the performance of the logarithm grid was relatively worse than its performance in the 2D case.

of  $Z = 12$  individuals and stopping criterion of  $I_{\max} = 12$  generations, also resulting in 144 SB runs. The results are depicted in Figure 5. By visual inspection, one can notice that the proposed method outperforms the other two with respect to diversity (recall that all methods use the same number of SB runs, which corresponds to approximately the same computational cost). The diversity values calculated with (7) lead to the same conclusion as the visual inspection: the proposed method, linear and logarithm grids have diversities of 0.82, 0.23, and 0.69, respectively.

Figure 5 indicates that the proposed method is able to provide a higher diversity for the same computational cost (i.e., SB runs). These results were obtained for a specific number of SB runs. It is also interesting to observe how the diversity behaves when the number  $B$  of SB runs is varied. Figure 6 was obtained varying  $B$  in the abscissa and using the same 2-user scenario (Figure 4). For the proposed method, we fixed a population size  $Z = 20$  and measured the diversity for every new generation (up to 20 generations) altogether

with the previous generations computed (no solution is discarded). For the baseline, a linear grid was obtained varying from  $G = 5$  to 50, with a step of 5. Figure 6 shows, for example, that for a diversity  $\mathcal{D} = 10.4$  the grid search needed nearly 1,600 SB runs, while the proposed method required 535 runs. The next subsection presents results for  $N = 3$ .

**4.2. 3-User Simulations.** The simulated 3-user scenario is depicted in Figure 7. For the grid simulations, a grid size of  $G = 10$  points was adopted, which corresponds to 1,000 SB runs. For the proposed *diverseSB* method, a population size of 100 individuals was used, with 10 generations (also leading to 1,000 SB runs). The results are shown in Figure 8. The diversity calculated with (7) for the proposed method, linear and logarithm grids are 0.47, 0.15, and 0.06, respectively.

**4.3. Simulations for More Than Three Users.** This subsection shows results for  $N = 4, 5$  and 6. The scenario simulated for 4

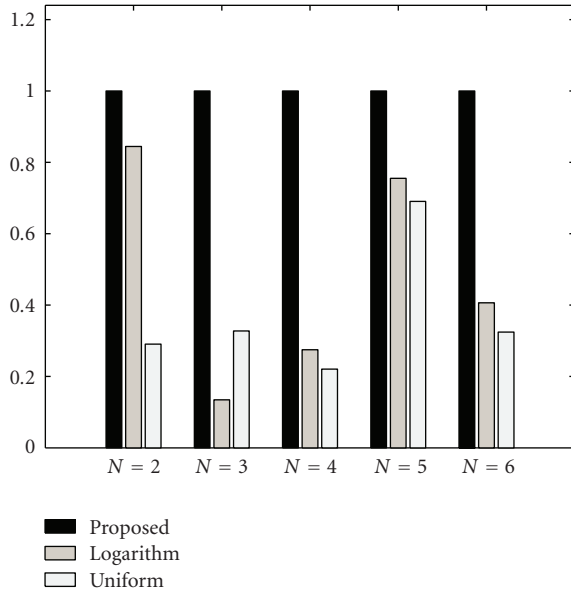


FIGURE 9: Summary of results with the polytope diversities normalized by the value corresponding to the proposed method.

users is depicted in Figure 3. The proposed *diverseSB* method used 10,000 SB runs (500 individuals and 20 generations). The grid simulations were adjusted to use the same number of SB runs, with  $G = 10$ . The diversities calculated with (7) for the proposed method, linear and logarithm grids are 0.34, 0.07, 0.09, respectively. Similarly, simulations for scenarios with 5 and 6 users were performed, generated by adding more lines to the CO. The grid simulations were adjusted to use  $G = 4$ , leading to 1,024 and 4,096 SB runs, respectively. In the same way, the proposed method was configured with  $Z = 16$  individuals and 64 generations (1,024 SB runs) for the 5-user scenario and  $Z = 32$  individuals and 128 generations (4,096 SB runs) for the 6-user scenario. The diversities calculated in the 5-user scenario with (7) are 0.0056, 0.0039, and 0.0042, for the proposed method, linear and logarithm grids, respectively. For the 6-user scenario, these results were 0.021, 0.0069 and 0.0021.

Figure 9 summarizes the comparison among the three methods for all five cases ( $N = 2, \dots, 6$ ). For convenience, the diversity values were normalized by the one corresponding to the proposed method in each case. The results indicate that, in these cases, the proposed method always presented a diversity larger than the one obtained by the best baseline.

## 5. Conclusions

This paper presented an efficient solution to the diverse set spectrum balancing problem. The computational cost is prohibitive if one adopts a straightforward (brute force) grid-search algorithm to solve the DSSB problem. This occurs because the runtime of these algorithms depends exponentially on  $N$ , the number of DSL users. The proposed solution is based on a multi-objective formulation of the

spectrum balancing problem and solved via an evolutionary genetic algorithm.

The simulation results showed that, for a given diversity, the computational cost of solving the DSSB problem using the proposed *diverseSB* algorithm can be much smaller than varying the weights according to a grid. When the *diverseSB* and the grid search were forced to have the same computational cost, the diversity of the former is much better.

## Acknowledgments

This work was partially supported by the Research and Development Center, Ericsson Telecommunications S.A., Brazil. B. Dortschy acknowledges financial support from the Swedish Agency for Innovation Systems, VINNOVA.

## References

- [1] T. Starr, J. M. Cioffi, and P. J. Silverman, *Understanding Digital Subscriber Line Technology*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1999.
- [2] W. Yu, G. Ginis, and J. M. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 1105–1115, 2002.
- [3] R. Cendrillon and M. Moonen, "Iterative spectrum balancing for digital subscriber lines," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, vol. 3, pp. 1937–1941, May 2005.
- [4] J. Papandriopoulos and J. S. Evans, "Low-complexity distributed algorithms for spectrum balancing in multi-user DSL networks," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, vol. 7, pp. 3270–3275, June 2006.
- [5] R. Moraes, B. Dortschy, A. Klautau, and J. R. Riu, "Semi-blind power allocation for digital subscriber lines," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 1420–1425, May 2008.
- [6] M. Monteiro, N. Lindqvist, and A. Klautau, "Spectrum balancing algorithms for power minimization in DSL networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, June 2009.
- [7] ANSI, "T1.417—Spectrum Management for Loop Transmission Systems, Issue 2," 2003.
- [8] T. Starr, M. Sorbara, J. M. Cioffi, and P. J. Silverman, *DSL Advances*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2003.
- [9] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York, NY, USA, 2001.
- [10] S. Galli and K. Kerpez, "Improved algorithms for single-ended loop make-up identification," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, vol. 1, pp. 67–72, 2006.
- [11] C. S. Sales, R. Menezes, F. Lindqvist, et al., "Line topology identification using multi-objective evolutionary computation," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, pp. 715–729, 2010.
- [12] F. Lindqvist, N. Lindqvist, B. Dortschy, et al., "Crosstalk channel estimation via standardized Two-Port measurements," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 916865, 14 pages, 2008.

- [13] P. Golden, H. Dedieu, and K. Jacobsen, *Fundamentals of DSL Technology*, Auerbach, Taylor & Francis, London, UK, 2006.
- [14] J. Goodman and J. O'Rourke, Eds., *The Handbook of Discrete and Computational Geometry*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2nd edition, 2004.
- [15] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.
- [16] 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.
- [17] R. Cendrillon, *Multi-user signal and spectra co-ordination for digital subscriber lines*, Ph.D. dissertation, Katholieke Universiteit Leuven, December 2004.
- [18] R. Lui and W. Yu, "Low-complexity near-optimal spectrum balancing for digital subscriber lines," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, vol. 3, pp. 1947–1951, May 2005.
- [19] R. Cendrillon, M. Moonen, J. Verliden, T. Bostoen, and W. Yu, "Optimal multi-user spectrum management for digital subscriber lines," in *Proceedings of the IEEE International Conference on Communications (ICC '04)*, vol. 1, pp. 1–5, June 2004.
- [20] C.-S. Tsou, S.-C. Chang, and P.-W. Lai, "Using crowding distance to improve multi-objective PSO with local search," in *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, chapter 5, Itech Education, Vienna, Austria, 2007.
- [21] "Qhull," <http://www.qhull.org/>.