Online versus Offline Optimization Methods for Raman Amplifier Optimization

Online versus Offline Optimization Methods for Raman Amplifier Optimization

U. C. de Moura[1], T. Pinto[1], A. M. Rosa Brusin[2], A. Carena[2], A. Napoli[3], D. Zibar[1], F. Da Ros[1]
1. DTU Fotonik, Technical University of Denmark, DK-2800, Kgs. Lyngby, Denmark
2. DET, Politecnico di Torino, Corso Duca degli Abruzzi, 24 - 10129, Torino, Italy
3. Infinera, Munich, Germany
fdro@fotonik.dtu.dk

Abstract: *In this work, we evaluate machine learning (offline) and evolutionary strategy (online) techniques for the Raman pump power optimization. Experimental results show that, although reusable and accurate, online tools may be time-consuming for reconfigurable amplifiers.*
Keywords: *Raman amplifier, differential evolution algorithm, neural network, artificial intelligence*

I. Introduction

The ability to provide broadband gain at any frequency has recently motivated researchers to consider Raman amplifiers (RA) for the amplification of multi-band signals [1]-[5]. Some of the aforementioned works have explored an exclusive RA feature: the shaping of gain spectrum by adjusting the powers of the pump lasers. This is a non-trivial optimization problem – as it involves differential equations, where given a target gain profile, the solution is the pump power configuration providing a gain as close as possible to the target.

Recently, machine learning (ML) has been applied to learn the complex pump-signal relations in RAs [6]. It has been shown to reach highly accurate gain profile optimization for C-band [7], S+C and S+C+L-bands [2]-[4] transmissions. Based on neural networks (NN) that learns from data, such a data-driven approach performs ultra-fast offline optimization, and normally dedicated NN models are required for each specific experimental scenario [8].

An alternative approach for Raman amplifier pump power optimization uses bio-inspired evolutionary strategies (ES) such as genetic algorithm [9]-[11], differential evolution (DE) algorithm [12][13], and most recently co-variance matrix adaptation evolutionary strategy [14]. Starting from a large set of random and non-trivial solutions, these population-driven approaches navigate through the solution space looking for the target performance. So far, these methods have been applied to the RA offline optimization only, using a numerical model to emulate the RA. This helps to speed up the optimization by allowing parallel evaluation of the solutions within the population. However, for the solution to be useful experimentally, a thorough physical layer characterization is also required [14]. This makes the offline ES approach also dependent on the experimental scenario. However, ES can also be applied online, directly interacting with the experimental setup to evaluate the solutions. Some examples are the optimization of frequency combs [15] and optical transmitters [16]. Nevertheless, online ES approaches have not yet been applied to the RA case. This would eliminate the need for a physical layer characterization, making the approach independent of the experimental scenario.

In this work, experimental Raman gain profiles are optimized online using a DE algorithm. During the procedure, the real response of a distributed 50-km standard single-mode fiber (SSMF) RA is monitored. The obtained results are compared to state-of-the-art offline optimizers based on inverse system design using ML [6]. DE solutions have superior accuracy, achieving maximum errors of 0.15 and 0.48 dB when optimizing flat gains of 3 and 5 dB, respectively. The main drawback of such an online iterative approach is the optimization time per target gain, which can reach more than 9 hours, against the milliseconds when using the offline approaches with trained NN models.

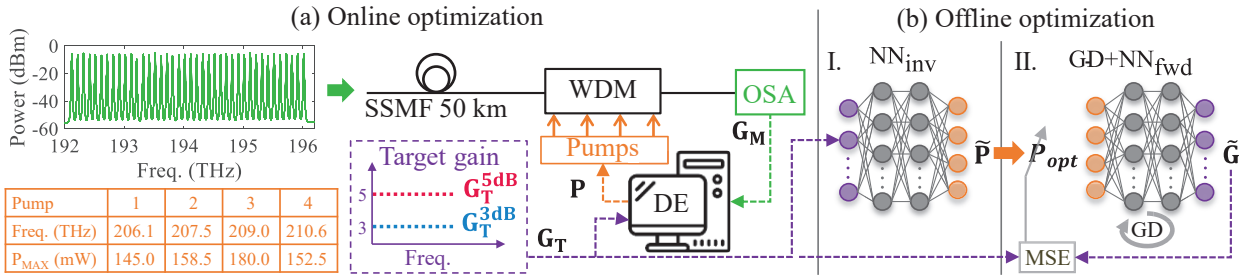
II. Experimental Setup

The experimental RA setup used for the online optimization is depicted in Fig.1(a). This setup was also used to generate the data set to build the ML-based inverse system design and to evaluate the final solutions. The input signal is generated by 40 continuous-wave lasers with a total power of +16.15 dBm and covering the entire C-band (192-196 THz). This signal goes through a 50-km SSMF, where it is distributedly amplified by 4 back-propagating Raman pump lasers. Pump frequencies are fixed and maximum powers P_{MAX} are also shown in Fig.1(a). Pump powers are remotely controlled and their powers are defined as $\mathbf{P} = [P_1,P_2,P_3,P_4]$. Pumps and signals are combined by a wavelength division multiplexer (WDM). At the RA output, an optical spectrum analyzer (OSA) measures the signal with a resolution of 0.1 nm. The Raman on-off gain profile $\mathbf{G} = [G_1,G_2,\ldots,G_{40}]$ is calculated as the difference between output spectra with the pump lasers turned on and off for each signal channel [8]. In Fig.1(a), $\mathbf{G_T}$ stands for target gain and $\mathbf{G_M}$ for measured gain.

III. Online optimization with differential evolution algorithm

The DE algorithm is a derivative-free, population-driven, and bio-inspired optimization method [15][17]. It relies on iteratively going through three stages: mutation, crossover, and fitness evaluation. The process starts at iteration k = 0 by

This project has received funding from the European Research Council through the ERC-CoG FRECOM project (grant agreement no. 771878), the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754462, the Villum Foundations (VYI OPTIC-AI grant no. 29344), and Ministero dell'Istruzione, dell'Università e della Ricerca (PRIN 2017, project FIRST).



Pump	1	2	3	4
Freq. (THz)	206.1	207.5	209.0	210.6
P_{MAX} (mW)	145.0	158.5	180.0	152.5

Fig. 1. (a) Experimental setup for the online optimization using differential evolution (DE) algorithm. (b) Offline optimization based on neural network models. MSE: mean squared error, GD: gradient descent, $\mathbf{G_T}$: target gain, $\mathbf{G_M}$: measured gain.

generating the population $\mathbf{X}_{k=0}=\{\mathbf{x}_{0,1},\mathbf{x}_{0,2},\ldots,\mathbf{x}_{0,N_c}\}$, where N_c is the number of solution candidates. For the RA gain profile optimization, each initial solution $\mathbf{x}_{0,i}$ is described by a pump power configuration $\mathbf{P}$ with values uniformly distributed in the interval $[0,P_{MAX}]$. For iteration k = 0, just the fitness evaluation stage is performed. In this case, all solutions within $\mathbf{X}_{k=0}$ are evaluated according to the fitness function $f(\mathbf{x}_{0,i}) = \max(|\mathbf{G_T}-\mathbf{G_M}(\mathbf{x}_{0,i})|)$. $f(\mathbf{x}_{0,i})$ corresponds to the maximum absolute error along the channels between target $\mathbf{G_T}$ and measured $\mathbf{G_M}(\mathbf{x}_{0,i})$ gains, with $\mathbf{G_M}(\mathbf{x}_{0,i})$ being the measured gain when applying the solution $\mathbf{x}_{0,i}$ to the pump powers in the experimental setup.

For each iteration $k > 0$, in the mutation stage, a donor vector $\mathbf{v}_{k,i}$ is created by mixing random solutions in the previous population $\mathbf{x}_{k-1,1:N_c}$. Each $\mathbf{v}_{k,i}$ is calculated as $\mathbf{v}_{k,i} = \mathbf{x}_{k-1,r_1} + F(\mathbf{x}_{k-1,r_2}-\mathbf{x}_{k-1,r_3})$, where r_1, r_2 and r_3 are population indexes randomly generated within $[1,N_c]$, and F is a scaling factor $\in [0,1]$. In the crossover stage, a trial vector $\mathbf{u}_{k,i}$ is generated by mixing $\mathbf{v}_{k,i}$ and $\mathbf{x}_{k-1,i}$ in the following way: for each element $u_{k,i,j} \in \mathbf{u}_{k,i}$, $u_{k,i,j} = v_{k,i,j}$ with a crossover probability p_c, and $u_{k,i,j} = x_{k-1,i,j}$ otherwise. Recall that j is the pump index ($j = [1..4]$). The boundaries of $\mathbf{u}_{k,i}$ are checked to avoid values outside the pump power limits. Finally, in the evaluation stage, $\mathbf{u}_{k,i}$ is evaluated using $f(\cdot)$. The new solution candidate $\mathbf{x}_{k,i}$ is updated according to $\mathbf{x}_{k,i} = \mathbf{u}_{k,i}$ if $f(\mathbf{u}_{k,i}) < f(\mathbf{x}_{k-1,i})$, or $\mathbf{x}_{k,i} = \mathbf{x}_{k-1,i}$ otherwise. The overall best solution candidate $\mathbf{x}_{best}$ is updated according to $\mathbf{x}_{best} = \mathbf{x}_{k,i}$ if $f(\mathbf{x}_{k,i}) < f(\mathbf{x}_{best})$. The final solution is given by $\mathbf{P_{DE}} = \mathbf{x}_{best}$.

To apply the DE, no prior knowledge of the physical layer is needed. Therefore, the setup time needed to prepare the optimizer for different experimental setups is negligible. In this work, we consider $N_c = 30$, F = 0.6, and $p_c = 0.8$. The maximum number of iterations is K = 100, with early-stop for no improvements after 25 consecutive iterations.

IV. Offline Optimization with Machine Learning

The offline optimization method applies the data-driven ML framework for the RA inverse design detailed in [6] and illustrated in Fig. 1(b). NN_{inv} (Fig. 1(b)-I) is the core of the framework and models the *inverse* mapping of the RA, i.e., the gain profile $\mathbf{G}$ to the pump configuration $\mathbf{P}$ ($\mathbf{G} \rightarrow \mathbf{P}$). Therefore, NN_{inv} provides a direct retrieval of the proper pump power configuration $\widetilde{\mathbf{P}}$ for a target gain profile $\mathbf{G_T}$. In case NN_{inv} does not provide sufficient accuracy, an additional fine optimization routine is applied. It consists of using NN_{fwd} in a trajectory-driven gradient descent (GD) loop to refine the pump configuration (Fig. 1(b)-II). NN_{fwd} models the *forward* RA mapping $\mathbf{P} \rightarrow \mathbf{G}$. It is equivalent to solving the Raman coupled equations, with the difference of being trained on measured data and providing nearly-instantaneous gain profile predictions. The optimization starts with the NN_{inv} outcome as the initial solution. During the optimization, the mean squared error (MSE) between $\mathbf{G_T}$ and NN_{fwd} prediction $\widetilde{\mathbf{G}}$ is minimized. NN_{fwd} provides a fast and differentiable model to calculate the gradient and speed up the fine optimization. The final solutions are $\mathbf{P_{NN_{inv}}} = \widetilde{\mathbf{P}}$, and $\mathbf{P_{GD+NN_{fwd}}} = \mathbf{P_{opt}}$.

There are two important steps needed before having the framework ready to be used. (1) *Experimental data acquisition*, where uniformly randomly pump configurations are applied to the experimental RA (Fig. 1(a)) and the corresponding Raman gains are measured. The final dataset is given by $\mathcal{D} = \{\mathbf{P_n},\mathbf{G_n} \mid n = 1,\ldots,N\}$, where N is the total number of cases. (2) *Supervised training*, consisting in training NN_{inv} and NN_{fwd} using $\mathcal{D}$. Details about how the NNs architecture, training and performance in a wide range of gain profiles (flat, tilted, and arbitrary) can be found in [7]. The first step is the most time-consuming. In our experimental setup, it took ~20 seconds to capture each case in $\mathcal{D}$. Since we consider N = 3000, the total setup time is 16.5 hours. The second step takes only few minutes and has, therefore, negligible contribution to the total setup time. However, after training, the optimization time is almost instantaneous, as shown in the following.

V. Results and Discussion

Here we apply the aforementioned DE and ML-based methods to optimize two flat target gain profile cases, $\mathbf{G_T^{3dB}}$ and $\mathbf{G_T^{5dB}}$ (Fig. 1(a)). As a reference, we look for measured gain profiles $\mathbf{G_M}$ inside the dataset $\mathcal{D}$ that are close to $\mathbf{G_T^{3dB}}$ and $\mathbf{G_T^{3dB}}$. This pseudo brute-force approach provides absolute maximum errors $E_{MAX} = \max(|\mathbf{G_T} - \mathbf{G_M}|)$ of 0.22 and 0.48 dB, respectively. These values can be used as a reference since they are related to gain profiles achievable by the experimental setup. However, better solutions might be still possible since it is not a comprehensive brute-force search.

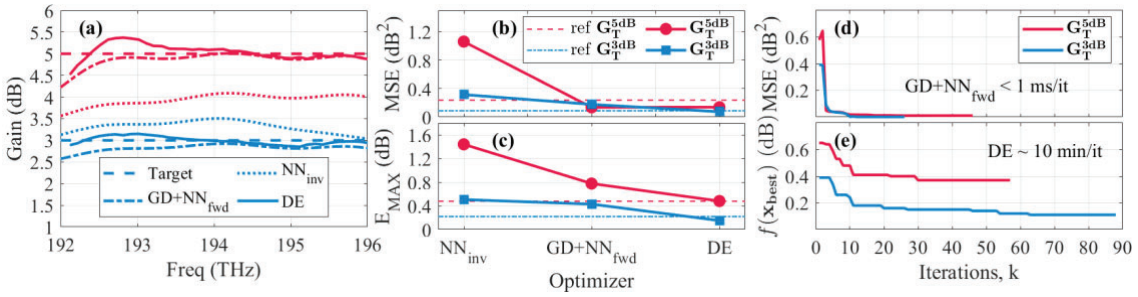


Fig. 2. (a) Target and measured gain profiles. Optimization performance in terms of (b) mean squared error (MSE) and (c) maximum absolute error (E_{MAX}) for the different optimizers and the pseudo brute-force search (ref). Convergence analysis for (d) GD + NN_{fwd} and (e) DE.

The computed solutions $\mathbf{P_{DE}}$, $\mathbf{P_{NN_{inv}}}$, and $\mathbf{P_{GD+NN_{fwd}}}$ are applied to the experimental setup in Fig. 1(a) and their corresponding gains are measured. Target and measured gain profiles are shown in Fig. 2(a). Fig. 2 also shows optimization and pseudo brute search (ref) performance in terms of MSE (Fig. 2(b)) and E_{MAX} (Fig. 2(c)).

The worst performance for NN_{inv} in optimizing $\mathbf{G_T^{5dB}}$ is because there is just one case in the training dataset with a gain profile close to $\mathbf{G_T^{5dB}}$, while there are ten cases close to $\mathbf{G_T^{3dB}}$ (where *close* means $E_{MAX} < 0.5$ dB). In fact, a uniform distribution on the pump powers (NN_{fwd} input) does not guarantee a uniform distribution on the gain profiles (NN_{inv} input). Since NN_{inv} was not trained with sufficient points around $\mathbf{G_T^{5dB}}$, it will not be able to provide good pump predictions (generalize) for $\mathbf{G_T^{5dB}}$ as it does for $\mathbf{G_T^{3dB}}$ [6]. NN_{fwd}, on the other hand, was trained over a uniformly distributed input (pump powers). Therefore, NN_{fwd} can be accurate enough to help GD + NN_{fwd} in overcome the NN_{inv} limitations, as shown by MSE and E_{MAX} improvements.

When applying DE, there are no significant MSE improvements w.r.t. GD + NN_{fwd} as observed for E_{MAX}. This is because GD is based on MSE while DE on E_{MAX}. By minimizing E_{MAX}, DE may also minimize MSE because a low E_{MAX} can lead to a low MSE. However, the opposite is not true. This means that a low MSE can still have high errors that are averaged along the signal channels. Therefore, DE and GD + NN_{fwd} reach accurate MSE results, while only DE has a better E_{MAX}. Moreover, DE directly monitors the pump power solutions in the experimental setup during the optimization routine, with no approximations as for the GD + NN_{fwd}.

Fig. 2(d-e) shows how the cost functions MSE and $f(\mathbf{x}_{best})$ evolve along the iterations for the GD + NN_{fwd} and DE methods, respectively. Although both methods optimize $\mathbf{G_T^{5dB}}$ after a similar number of iterations, the total optimization time is very different. At each iteration, DE experimentally evaluates $N_c = 30$ candidate solutions, which takes around 10 minutes, while GD + NN_{fwd} evaluates a single candidate in less than 1 millisecond using NN_{fwd}.

Table 1 summarizes the optimization performance, including the optimization time to achieve each target gain and the setup time, considering a standard personal computer (Intel Core i5 @3.4 GHz, 8 GB RAM). The main drawback of NN-based optimizers is the setup time due to the experimental data acquisition. These optimizers are also dependent on the considered experimental setup used during the preparation steps. However, once they are ready, ultra-fast RA gain optimization can be achieved, relying on a simple and fast matrix multiplication if just NN_{inv} is applied, or taking just a few milliseconds with GD + NN_{fwd}. In contrast, the proposed DE can be reusable for any RA experimental setup with minimum time and effort, not requiring physical layer characterization as in an offline approach [14]. Nevertheless, DE's flexibility and highly accurate solutions come with the cost of having a very long optimization time that scales up with the number of target gains. To optimize both $\mathbf{G_T^{3dB}}$ and $\mathbf{G_T^{5dB}}$, DE takes a total time (setup + optimization) of 24 hours, while NN-based approaches have a total time close to the setup time.

VI. Conclusions

Optimization tools are very important to leverage the ability to shape the gain profile of the Raman amplifiers. Many offline tools have been reported in the literature with this purpose. However, they are highly dependent on the experimental setup used to build them. In this work, an online optimization tool, highly accurate and reusable for different experimental setups, was compared to some state-of-the-art offline tools. The obtained results show that the only drawback is the high optimization time. This imposes critical limitations in future flexible optical networks, where rapid actions are desirable. Therefore, in scenarios requiring reconfigurability, NN-based optimizers may be preferable.

TABLE I
COMPARING THE OPTIMIZERS IN TERMS OF ERRORS (MSE, AND E_{MAX}, LOWEST VALUES IN BOLD) AND TIME (OPTIMIZATION, SETUP AND TOTAL).

Optimizer	$\mathbf{G_T^{3dB}}$			$\mathbf{G_T^{5dB}}$			Setup time	Total time
	MSE (dB)	E_{MAX} (dB)	Optimization time	MSE (dB)	E_{MAX} (dB)	Optimization time		
NN_{inv}	0.32	0.51	~0	1.06	1.44	~0	16.5 h	16.5 h
GD + NN_{fwd}	0.18	0.43	17 ms	**0.14**	0.78	23 ms	16.5 h	16.5 h
DE	**0.07**	**0.15**	14.5 h	**0.14**	**0.48**	9.5 h	~0	24 h
Pseudo brute search (ref)	0.09	0.22	~0	0.24	**0.48**	~0	16.5 h	16.5 h

REFERENCES

[1] M. A. Iqbal, L. Krzczanowicz, I. D. Philips, P. Harper, and W. Forysiak, "Noise Performance Improvement of Broadband Discrete Raman Amplifiers Using Dual Stage Distributed Pumping Architecture," J. Lightw. Technol., vol. 37, no. 14, pp. 3665–3671, 2019.

[2] M. Ionescu, "Machine Learning for Ultrawide Bandwidth Amplifier Configuration," in Proc. 21st Int. Conf. Transparent Opt. Netw., 2019, paper We.B7.3.

[3] X. Ye, A. Arnould, A. Ghazisaeidi, D. L. Gac, and J. Renaudier, "Experimental Prediction and Design of Ultra-Wideband Raman Amplifiers Using Neural Networks," in Proc. Opt. Fiber Commun. Conf., 2020, paper W1K.3.

[4] U. C. de Moura, M. A. Iqbal, M. Kamalian, L. Krzczanowicz, F. Da Ros, A. M. R. Brusin, A. Carena, W. Forysiak, S. Turitsyn, and D. Zibar, "Multi-Band Programmable Gain Raman Amplifier," J. Lightw. Technol., vol. 39, no. 2, pp. 429–438, 2021.

[5] Ferrari, Alessio, et al. "Assessment on the achievable throughput of multi-band ITU-T G. 652. D fiber transmission systems. " J. Lightw. Technol., vol. 38, no. 16, pp. 4279–4291, 2020.

[6] D. Zibar et al., "Inverse System Design Using Machine Learning: The Raman Amplifier Case," J. Lightw. Technol., vol. 38, no. 4, pp. 736–753, 2020.

[7] U. C. de Moura, F. Da Ros, A. M. R. Brusin, A. Carena, and D. Zibar, "Experimental Characterization of Raman Amplifier Optimization Through Inverse System Design," J. Lightw. Technol., vol. 39, no. 4, pp. 1162–1170, 2021.

[8] U. C. de Moura, D. Zibar, A. M. Rosa Brusin, A. Carena, and F. Da Ros, "Generalization Properties of Machine Learning-based Raman Models," in Proc. Opt. Fiber Commun. Conf., 2021, paper Th1A.28.

[9] V. E. Perlin and H. G. Winful, "Optimal Design of Flat-Gain Wide-band Fiber Raman Amplifiers," J. Lightw. Technol., vol. 20, no. 2, p. 250, Feb 2002.

[10] B. Neto, A. L. J. Teixeira, N. Wada, and P. S. Andre, "Efficient use of hybrid genetic algorithms in the gain optimization of distributed Raman amplifiers," Opt. Express, vol. 15, no. 26, pp. 17 520–17 528, Dec 2007.

[11] X. Zhou, C. Lu, P. Shum, and T. Cheng, "A simplified model and optimal design of a multiwavelength backward-pumped fiber Raman amplifier," IEEE Phot. Tech. Lett., vol. 13, no. 9, pp. 945–947, 2001.

[12] J. Chen and H. Jiang, "Optimal Design of Gain-Flattened Raman Fiber Amplifiers Using a Hybrid Approach Combining Randomized Neural Networks and Differential Evolution Algorithm," IEEE Photonics Journal, vol. 10, no. 2, pp. 1–15, 2018.

[13] M. Li, L. Zhang, B. Liu, Q. Zhang, Y. Wang, Q. Tian, X. Yin, X. Xin, K. Wang, and J. Zou, "Design of multipumped Raman fiber amplifier by differential evolution optimization," in Proc. 13th Int. Conf. Transparent Opt. Netw., 2014, pp. 1–4.

[14] G. Borraccini, S. Staullu, S. Piciaccia, A. Tanzi, G. Galimberti, and V. Curri, "Cognitive Raman Amplifier Control Using an Evolutionary Optimization Strategy," IEEE Phot. Tech. Lett. vol. 34, no. 4, pp. 223-226, 2022.

[15] T. Pinto, U. C. de Moura, F. D. Ros, M. Krstic, J. V. Crnjanski, A. Napoli, D. M. Gvozdic, and D. Zibar, "Optimization of frequency combs spectral-flatness using evolutionary algorithm," Opt. Express, vol. 29, no. 15, pp. 23 447–23 460, Jul 2021.

[16] J. C. M. Diniz, F. Da Ros, E. P. da Silva, R. T. Jones, and D. Zibar, "Optimization of DP-M-QAM Transmitter Using Cooperative Coevolutionary Genetic Algorithm," J. Lightw. Technol., vol. 36, no. 12, pp. 2450–2462, 2018.

[17] S. Koziel and X.-S. Yang, Computational optimization, methods and algorithms. Springer, 2011, vol. 356.