Particle swarm optimization-assisted approach for the extraction of VCSEL model parameters

A Particle Swarm Optimization Assisted Approach for the Extraction of VCSEL Model Parameters

ANDREA MARCHISIO[1,*], ENRICO GHILLINO[2], VITTORIO CURRI[1], ANDREA CARENA[1], AND PAOLO BARDELLA[1]

[1] *Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italia*
[2] *Synopsys Inc., 400 Executive Blvd, Ossining, NY 10562, United States*
[*] *andrea_marchisio@polito.it*

Compiled December 22, 2023

We propose a direct Particle Swarm Optimization (PSO) method for extracting the parameters of a physical model describing the behavior of Vertical-Cavity Surface-Emitting Lasers (VCSELs), starting from the light-current (L-I) characteristics and the small signal modulation (S21) responses, at different currents and temperatures. With an optimal choice of the hyperparameters of the algorithm, the method is able to predict parameters that accurately reproduce the behavior of the device. Its prediction capabilities are compared to those of two commonly used nonlinear optimizers (Interior Point and Levenberg-Marquardt), to benchmark its performances.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

Vertical-Cavity Surface-Emitting Lasers (VCSELs) are experiencing an ever-growing success due to their simple manufacturing, low production costs, and adaptability to a wide range of applications, particularly for short-reach communications in data centers, optical sensing, and other industrial applications [1–3].

However, their behavior is not simple to model, due to their inherent nonlinearities, strong thermal effects, and other phenomena, such as spatial hole burning, thus requiring the introduction of a large number of equations and parameters to properly reproduce the experimental findings. Therefore, it is important to possess reliable methods for an easy and rapid physical characterization of such devices: parameter values needed for numerical models must be inferred from experimental measurements. The problem of parameter extraction and, similarly, the problem of inverse design are traditionally tackled with brute-force techniques that are extremely inefficient, both for their computational cost and for the time required to find a solution. In recent years, other approaches have been gaining more and more momentum, in particular Machine Learning (ML) techniques [4, 5] and genetic or evolutionary algorithms, such as Particle Swarm Optimization (PSO) [6].

Since the generation of large data sets for the training of Artificial Neural Networks (ANNs) is generally time consuming and computationally expensive, and, to the best of our knowledge, evolutionary algorithms have been used mainly as companion optimizers to ANNs during inverse design of VCSELs, in this letter we explore the use of a PSO algorithm for the direct extraction of VCSEL physical model parameters. In particular, our work considers the VCSEL model included in Synopsys OptSim™ [7], which allows simpler subsequent simulations at the system level, realistically reproducing the behavior of actual devices. We analyze the impact of the hyperparameters of the PSO algorithm on the outcome of the optimization and present its capability of extracting a set of parameters that accurately reproduce the behavior of the device, with reasonable computation time for a non real-time application. To prove the generality of the method, the optimization is repeated for 100 different VCSELs, thus obtaining a reliable statistical distribution of the prediction errors. The optimizations are performed on simulated curves for better reproducibility, however the method could be extended to experimental measurements of devices to be characterized. Finally, the PSO algorithm is compared against two nonlinear optimizers from the MATLAB Optimization Toolbox to benchmark its performances when addressing a complicated problem as the physical model parameter extraction of a VCSEL.

2. VCSEL MODEL AND PSO ALGORITHM

For this analysis, the VCSEL behavior is described by means of a well-established physical model based on carrier and photon rate equations, taking into account temperature effects [8]. It is important to note that the proposed approach can be straightforwardly extended to more complex models for VCSELs or even to other active devices.

First of all, for simplicity's sake, the explicit spatial dependency for the carrier and photon numbers is neglected by assuming a cylindrical geometry with no azimuthal variations and by describing the distribution of carriers in the radial direction r through its expansion in a two-term Bessel series [8]:

$$N(r,t) = N_0(t) - N_1(t)J_0(\sigma_1 r/R) \qquad \textbf{(1)}$$

with σ_1 first nonzero root of J_1, J_0 and J_1 Bessel functions of the first kind, and R effective radius of the active layer. The temporal evolution of the expansion coefficients $N_0(t)$ and $N_1(t)$ is given by the following spatially independent rate equations:

Table 1. Investigated VCSEL parameters, with variation ranges and values of the target device.

Parameters	Range	Target	Parameters	Range	Target
Injection efficiency η_i	0.70 to 1.00	0.72	Transp. number N'_{tr_0}	2.00×10^6 to 1.00×10^8	2.03×10^7
Power coeff. k_f (nW)	10.00 to 60.00	30.00	Transp. number coeff. C'_{n_1} (kK^{-1})	−100.00 to −1.00	-5.45
Carrier lifetime τ_n (ns)	0.50 to 5.00	3.98	Transp. number coeff. C'_{n_2} (kK^{-2})	0.00 to 100.00	12.94
Photon lifetime τ_p (ps)	1.50 to 3.50	2.73	Leakage current factor I_{l_0} (A)	1.00 to 2.00	1.70
Gain coeff. G'_0 (ms^{-1})	−360.0 to −11.1	−210.0	Leakage current coeff. a_0 (K)	2.00×10^3 to 1.00×10^4	7.38×10^3
Gain coeff. a'_{g_1} (kK^{-1})	−5.00 to −0.50	-1.36	Leakage current coeff. a_1 (K)	0.00 to 3.00×10^{-4}	9.81×10^{-5}
Gain coeff. a'_{g_2} (kK^{-2})	−50.00 to −2.00	-12.88	Leakage current coeff. a_2	1.00×10^{-9} to 4.00×10^{-8}	2.60×10^{-8}
Gain coeff. b'_{g_1} (kK^{-1})	−100 to 0	-51.35	Diffusion parameter h_diff	1.00 to 20.00	14.53
Gain coeff. b'_{g_2} (kK^{-2})	5.56 to 900.0	201.8	Thermal impedance R_th (K/W)	5.00×10^2 to 8.00×10^3	3.89×10^3
Gain saturation factor ϵ	1×10^{-6} to 3×10^{-6}	2.95×10^{-6}			

$$\frac{\mathrm{d}N_0}{\mathrm{d}t} = +\frac{\eta_\mathrm{i} I}{q} - \frac{N_0}{\tau_\mathrm{n}} - \frac{I_\mathrm{l}(N_0, T)}{q} - \frac{G(T)\left[\gamma_{00}(N_0 - N_\mathrm{tr}(T)) - \gamma_{01}N_1\right]}{1+\epsilon S} S \tag{2}$$

$$\frac{\mathrm{d}N_1}{\mathrm{d}t} = -\frac{N_1}{\tau_\mathrm{n}}(1 + h_\mathrm{diff}) + \frac{G(T)\left[\phi_{100}(N_0 - N_\mathrm{tr}(T)) - \phi_{101}N_1\right]}{1+\epsilon S} S \tag{3}$$

with η_i injection efficiency, I injected current, q electron charge, τ_n carrier lifetime, T temperature, $G(T)$ gain coefficient, $N_\mathrm{tr}(T)$ transparency carrier number, $I_\mathrm{l}(N_0, T)$ leakage current, ϵ gain compression factor, h_diff diffusion coefficient. The coefficients $\gamma_{00}, \gamma_{01}, \phi_{100}, \phi_{101}$ quantify the overlap between the fundamental transverse mode, assumed to have a Gauss-Laguerre profile, and the active region [9].

The set of rate equations is completed by the differential equation describing the total number of photons S:

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\frac{S}{\tau_\mathrm{p}} + \frac{\beta_\mathrm{sp}N_0}{\tau_\mathrm{n}} + \frac{G(T)\left[\gamma_{00}(N_0 - N_\mathrm{tr}(T)) - \gamma_{01}N_1\right]}{1+\epsilon S} \tag{4}$$

with τ_p photon lifetime and β_sp spontaneous emission coefficient. The output power P_out is proportional to S through a proper coupling coefficient k_f.

The following dynamic expression is employed to estimate the device temperature and to take into account the thermal dependencies [10]:

$$T = T_\mathrm{amb} + (IV - P_\mathrm{out})R_\mathrm{th} - \tau_\mathrm{th}\frac{\mathrm{d}T}{\mathrm{d}t} \tag{5}$$

with V applied voltage, T_amb ambient temperature, R_th thermal impedance, and $\tau_\mathrm{th} = 1\,\mu\mathrm{s}^{-1}$ thermal time constant. Finally, three empirical laws describe the temperature-dependent gain coefficient $G(T)$, the transparency carrier number $N_\mathrm{tr}(T)$ and the leakage current $I_\mathrm{l}(N_0, T)$ [8]:

$$G(T) = G_0\frac{a_{\mathrm{g}_0} + a_{\mathrm{g}_1}T + a_{\mathrm{g}_2}T^2}{b_{\mathrm{g}_0} + b_{\mathrm{g}_1}T + b_{\mathrm{g}_2}T^2} = G'_0\frac{1 + a'_{\mathrm{g}_1}T + a'_{\mathrm{g}_2}T^2}{1 + b'_{\mathrm{g}_1}T + b'_{\mathrm{g}_2}T^2} \tag{6}$$

$$N_\mathrm{tr}(T) = N_{\mathrm{tr}_0}\left(C_{\mathrm{n}_0} + C_{\mathrm{n}_1}T + C_{\mathrm{n}_2}T^2\right) = N'_{\mathrm{tr}_0}\left(1 + C'_{\mathrm{n}_1}T + C'_{\mathrm{n}_2}T^2\right) \tag{7}$$

$$I_\mathrm{l}(N_0, T) = I_{\mathrm{l}_0}\exp\left(\frac{-a_0 + a_1N_0 + a_2N_0T - a_3/N_0}{T}\right) \tag{8}$$

In Eq. (6) and Eq. (7), the rightmost expressions are introduced as substitutes for those from [8] to uniquely define the values of the parameters and avoid ambiguities.

The PSO algorithm, first implemented to mimic the behavior of flocks of birds, was discovered to accurately predict global minima in optimization problems [11]. Optimization is performed by a swarm of N_p particles that move in a N-dimensional solution space with a specific velocity: each dimension corresponds to one of the parameters to optimize and each position in the space represents a solution. Each particle is moving within the solution space according to specific rules that are related to the distance of the particles with respect to the global and personal best positions. These positions are associated with a *fitness* parameter, which is a measure of the distance from the target and thus of the accuracy of the optimization itself.

Starting from an initial configuration in which the position of the particles is randomized in the given parameter ranges, at each k-th step of the algorithm, the velocity $\mathbf{v}_j$ and position $\mathbf{x}_j$ of the j-th particle are updated with the following rules [11]:

$$\mathbf{v}_j^{k+1} = c_\mathrm{i}\mathbf{v}_j^k + c_\mathrm{c}r_1(\mathbf{p}_j^k - \mathbf{x}_j^k) + c_\mathrm{s}r_2(\mathbf{p}_\mathrm{gl}^k - \mathbf{x}_j^k) \tag{9}$$

$$\mathbf{x}_j^{k+1} = \mathbf{x}_j^k + \mathbf{v}_j^{k+1} \tag{10}$$

with c_i inertia coefficient, c_c cognitive acceleration coefficient, c_s social acceleration coefficient, r_1 and r_2 random scaling factors, $\mathbf{p}_j^k$ personal best position for the j-th particle, and $\mathbf{p}_\mathrm{gl}^k$ global best position. Then, after the update of the positions, the new fitness of each particle is computed in order to check whether new personal or global bests have been found. For this particular application, the solution space has N=19 dimensions, since that is the number of physical parameters that we are extracting. The 19 parameters and their associated ranges of values are reported in Tab. 1. Note that the bounds for the modified fitting parameters for Eq. (6) and Eq. (7) are obtained from the ranges of the original ones. These ranges create a hard boundary for the solution space and, if a particle reaches one of them, it is "bounced off" back inside the solution space.

The optimization target is a set of L-I curves and S21 responses of a simulated device, whose parameters are reported in Tab. 1. For the purpose of this research, the L-I characteristics are computed by numerically solving the rate equations under stationary conditions, while the responses S21 are evaluated as the response of the output power to a current impulse [12].

To allow the extraction of the coefficients describing the thermal effects, M_LI=3 L-I curves are calculated at 25 °C, 40 °C, and 55 °C. For each temperature, the values are saved for 16 equally spaced currents ranging from 0 mA to 15 mA. Furthermore, to have a complete picture of the frequency response, M_S21=6 small signal modulation responses are evaluated at 2 mA, 4 mA, and 6 mA for T = 25 °C, at 2 mA and 4 mA for T = 40 °C, and at 5 mA for T = 55 °C: for each case, 16 data samples are stored, logarithmically spaced between 100 MHz and 15 GHz. As a result, we deal with 48 samples extracted from the L-I curves and 96 samples from the S21 curves, for a total of 144 samples.

In order to compute the fitness, the average Mean Squared Errors (MSEs) of the L-I and S21 curves in linear scale are calculated at each step of the optimization with respect to the target

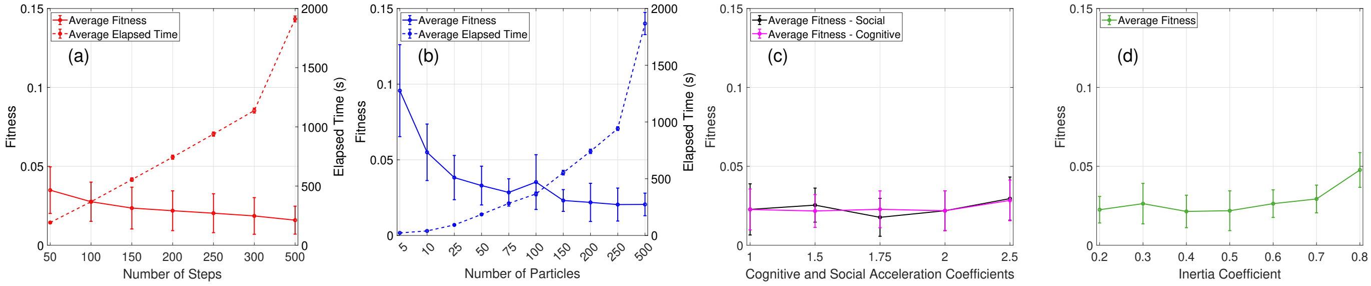

Fig. 1. Impact of PSO hyperparameters. **(a)** Fitness and computation time vs. maximum number of steps N_s. **(b)** Fitness and computation time vs. number of particles N_p. **(c)** Fitness vs. social c_s and cognitive c_c acceleration coefficients. **(d)** Fitness vs. inertia coefficient c_i.

device. For the j-th particle, at the k-th step,

$$\mathrm{MSE}^k_{\mathrm{LI},j} = \frac{1}{M_{\mathrm{LI}}} \sum_{m=1}^{M_{\mathrm{LI}}} \frac{||P^m_{\mathrm{target}} - P^{m,k}_{\mathrm{PSO},j}||}{||P^m_{\mathrm{target}}||} \quad \textbf{(11)}$$

$$\mathrm{MSE}^k_{\mathrm{S21},j} = \frac{1}{M_{\mathrm{S21}}} \sum_{m=1}^{M_{\mathrm{S21}}} \frac{||S21^m_{\mathrm{target}} - S21^{m,k}_{\mathrm{PSO},j}||}{||S21^m_{\mathrm{target}}||} \quad \textbf{(12)}$$

where $||\cdot||$ indicates the norm-2 of the vectors containing the power P and small signal response $S21$ data samples. Finally, the fitness f^k_j is defined as the average of $\mathrm{MSE}^k_{\mathrm{LI},j}$ and $\mathrm{MSE}^k_{\mathrm{S21},j}$; during the optimization, the quantity $f^k = \min_j f^k_j$ is minimized.

3. HYPERPARAMETER ANALYSIS

We started by analyzing the impact of hyperparameters on the performance of the PSO optimizer itself, implemented with a custom MATLAB code for maximum flexibility. The results of this analysis can easily be generalized for problems with a comparable number of parameters. The optimization was run multiple times, changing one hyperparameter at a time. In particular, we focused on the number of steps N_s, number of particles N_p (the convergence depends on how many agents we consider and on their initial position), inertia c_i (how fast the particle moves in the solution space), social acceleration c_s (how fast the particle tends to the current global best position), and cognitive acceleration coefficient c_c (how much the particle explores new solutions, around the best personal position of the particle). For the first two variables, we also studied the impact of the hyperparameters on the simulation time, since we are targeting a fast optimizer. The performance of the optimizer depends on the random initialization of the particles and the random coefficients present in Eq. Eq. (9): for this reason, the simulations have been repeated for 10 different random seeds to generate a statistical distribution of the results. The reported results have been obtained on a Windows 10 workstation with Intel Core i7-8700 and 48 GB of RAM.

In Fig. 1a, we report the fitness trends (solid line) and the computation time (dashed line) for different values of the number of simulation steps, averaged over 10 random seeds. The other hyperparameters are set to $N_p = 200$, $c_i = 0.5$, $c_s = 2$, and $c_c = 2$. The error bars represent the standard deviation from the average value. As expected, the fitness decreases with a larger number of steps, since the optimizer has time to converge to even more accurate solutions. Note that lower numbers of steps have a larger variation with respect to the random seed choice. On the other hand, the elapsed computation time increases sharply with the number of steps; therefore, we have to carefully balance the number of steps in order to have the best trade-off. A good balance, taking into account the dispersion of random seeds, is represented by a number of simulation steps ranging from 150 to 200.

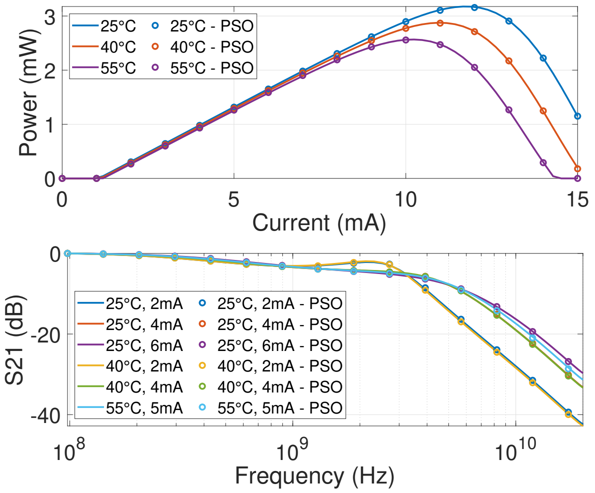

Fig. 2. PSO predictions (circles) of L-I and S21 curves compared with the target curves (solid lines).

Similar conclusions can be drawn from the trends obtained by changing the number of particles, reported in Fig. 1b. In this case, the other hyperparameters are fixed at $N_s = 200$, $c_i = 0.5$, $c_s = 2$, and $c_c = 2$. A reasonable fitness-time trade-off can be obtained for values N_p between 100 and 200.

Regarding velocity coefficients, the simulations are performed with $N_s = 200$ and $N_p = 200$ and the trends are represented in Fig. 1c and Fig. 1d. In the case of social and cognitive coefficients, the fitness does not change drastically, and even the standard deviations (the error bars) remain near constant for all the selected values. Similarly, the inertia coefficient does not appear to substantially influence the fitness, but slightly worse results are obtained for higher values of c_i: a better convergence is generally obtained with slower particles, since they are less prone to overshoot the target position.

4. RESULTS

According to our previous findings, we performed an analysis using the optimal values of the hyperparameters obtained so far ($N_s = 200$, $N_p = 200$, $c_i = 0.4$, $c_s = 2$, $c_c = 2$) repeated for 10 different random seeds, each requiring approximately 16 minutes to complete. In Fig. 2, the comparison of the best predictions and the target L-I and S21 curves is shown. The graphical agreement is excellent and the fitness of this optimization is f=7.04 × 10^{-3} ,

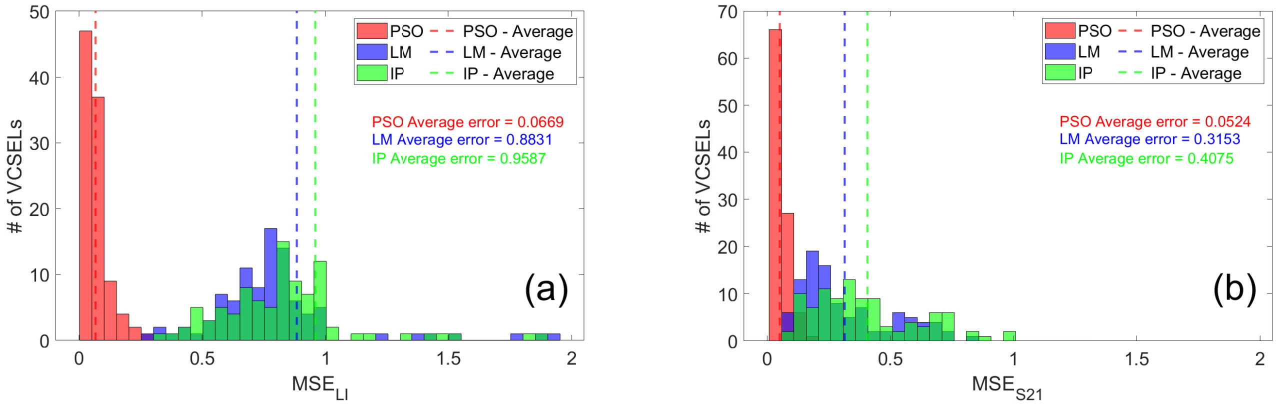

Fig. 3. Distributions of the average error for the L-I characteristics **(a)** and for the S21 responses **(b)** for 100 simulated devices, for the custom PSO algorithm and the Levenberg-Marquardt and Interior Point optimizers. The dashed vertical lines represent the average values of the distributions.

while the average value of 10 seeds is 2.14×10^{-2}, showing that PSO still performs well even for less effective seeds.

As previously discussed, the performances of the algorithm depend heavily on the initialization of the position of the particles and convergence might depend on the target curves. Therefore, to show that the method is able to adapt easily to different VCSELs, the optimization is repeated on the L-I and S21 curves coming from 100 different simulated target devices. In Fig. 3, we reported the MSEs distribution for the L-I curves (a) and the S21 responses (b), using the PSO with optimal hyperparameters (in red). In both cases, the errors are small, showing not only that the method is reliable, but also that the optimization of the hyperparameters previously discussed is general, since the performances align with those of the example of Fig. 2. Indeed, using the same error metric, the optimization of the device of Fig. 2 produces mean errors of 5.53×10^{-3} and 8.55×10^{-3} for the L-I curves and the S21 responses, respectively.

Finally, we compared the PSO performances for the parameter extraction of the 100 VCSELs with those of two built-in algorithms from the MATLAB Optimization Toolbox [13] and we reported their error histograms on Fig. 3a and Fig. 3b. In particular, we tested the Levenberg-Marquardt (LM, in blue) and the Interior Point (IP, in green) algorithms that can be specified in the *lsqcurvefit* MATLAB function. In Fig. 3, we also report the average errors for each optimizer (the dashed vertical lines). The LM and IP optimizers are launched with a high maximum number of steps (1×10^4) and a small target tolerance (1×10^{-15}) to ensure convergence. From the graphs, it can be appreciated that, predictions-wise, PSO is outperforming the other optimizers both for the L-I characteristics and the S21 responses, as it provides error distributions that are narrower and closer to zero, as can be observed from the smaller average values. The other optimizers seem to struggle more with the L-I curves than the S21 responses, whereas for PSO this effect is marginal. With our current implementation, PSO is six to eight times slower than the other optimizers; we expect further optimizations and parallelizations of the code to reduce its computational time.

5. CONCLUSIONS AND OUTLOOK

We analyzed the performance of a PSO algorithm applied to the extraction of the parameters of the VCSEL physical model from the device responses. The PSO is capable of extracting the set of 19 parameters that reproduce well the L-I and S21 curves of a target VCSEL and it can perform well on any other device that can be described by the presented model. With its more consistent predictions, PSO is also shown to outperform the LM and IP optimizers, which are not able to deal with such a complex nonlinear problem effectively. The strength of this algorithm lies in its adaptability: the complexity of the model can be increased, the number of parameters can be scaled up, and the target curves can be modified, but the optimizer will still suit the new problem and converge to a solution, with limited computation time. This makes the PSO particularly well suited for characterizing any unknown device starting from a set of measurements, while a machine learning engine would require *ad hoc* data sets. Real VCSEL measurements could deviate from the ideal curves considered in this work. However, the proposed methodology is not affected, as long as the measurements are properly processed before optimization (removing artifacts and properly filtering the measurement noise); more complex physical models could be required to describe additional effects not included in [8].

Funding. A. Marchisio's PhD scholarship is co-funded by the Italian Ministry of University and Research (MUR) D.M. n. 352/2022 project funds (in the framework of the Italian *Piano Nazionale di Ripresa e Resilienza* and the European *NextGenerationEU*) and by Synopsys Inc.

Disclosures. The authors declare no conflicts of interest.

Data Availability. Data underlying the results are not publicly available but may be obtained from the authors upon reasonable request.

REFERENCES

1. M. Grabherr, in *Proceedings of SPIE,* (2015), p. 938102.
2. A. Liu, P. Wolf, J. A. Lott, and D. Bimberg, Photonics Res. **7**, 121 (2019).
3. J. Tatum, *VCSEL Fundamentals* (John Wiley & Sons, Ltd, 2021), chap. 2, p. 29.
4. A. Marchisio, I. Khan, L. Tunesi, *et al.*, in *European Conference on Integrated Optics,* (2023), p. 264.
5. I. Khan, L. Tunesi, M. U. Masood, *et al.*, in *International Conference on Software, Telecommunications and Computer Networks,* (2022), p. 1.
6. Z. Ma and Y. Li, Opt. Express **28**, 21971 (2020).
7. Synopsys OptSim, https://www.synopsys.com/photonic-solutions/pic-design-suite.html.
8. P. Mena, J. Morikuni, S.-M. Kang, *et al.*, J. Light. Technol. **17**, 2612 (1999).
9. P. Mena, J. Morikuni, and K. Wyatt, in *IEEE Lasers and Electro-Optics Society Annual Meeting,* (2000), p. 234.
10. N. Bewtra, D. Suda, G. Tan, *et al.*, IEEE J. Sel. Top. Quantum Electron. **1**, 331 (1995).
11. J. Kennedy and R. Eberhart, in *International Conference on Neural Networks,* , vol. 4 (1995), p. 1942.
12. P. Bardella, W. W. Chow, and I. Montrosset, Photonics **3** (2016).
13. J. Nocedal and S. Wright, *Numerical Optimization* (Springer, 2006), 2nd ed.