A Hybrid Bio-Inspired Metaheuristic Applied to PID Controller Parameter Tuning

Leonardo Ramos Rodrigues
Electronics Division, Institute of Aeronautics and Space
Electronics Engineering Department, Aeronautics Institute of Technology
São José dos Campos - SP, Brazil
leonardolrr@ita.br

João Paulo Pordeus Gomes
Department of Computer Science, Federal University of Ceará
Fortaleza - CE, Brazil
jpaulo@lia.ufc.br

Abstract

In this paper, a hybrid population-based, bio-inspired metaheuristic is applied to solve the PID controller parameter tuning problem. The applied algorithm, denoted by CSASOS, is a multi-population metaheuristic that combines good characteristics of the Crow Search Algorithm (CSA) and the Symbiotic Organisms Search (SOS). In CSASOS, one subpopulation follows the search rules of the CSA algorithm and is responsible for finding promising regions in the search space. The second subpopulation follows the search rules of the SOS algorithm and is responsible for performing a refined search in the promising regions to find the best solution for the optimization problem under consideration. Numerical experiments are carried out in order to compare the performance of CSASOS with the performance of the stand-alone versions of both CSA and SOS in the solution of a PDI controller parameter tuning problem, which is a common problem that appears in many industrial applications. The results show that CSASOS presented a better performance in comparison with the competing algorithms.

Keywords
Metaheuristic, multi-population, hybrid algorithms, symbiotic organisms search, crow search algorithm, PID controller.

1. Introduction
Bio-inspired metaheuristics have become popular in the last decades due to their ability to find close-to-optimal solutions for complex optimization problems in a reasonable amount of time. These algorithms have been successfully applied to solve a wide range of optimization problems that commonly appear in different industry sectors (Slowik and Kwasnicka 2018). Several bio-inspired algorithms have been proposed such as Particle Swarm Optimization (PSO) (Kennedy and Eberhart 995), Ant Colony Optimization (Dorigo et al. 1996), Genetic Algorithms (GA) (Holland 1992), Differential Evolution (DE) (Storn and Price 1997), and Grey Wolf Optimizer (GWO) (Mirjalili et al. 2014), among many others. However, despite the large number of bio-inspired algorithms available in the literature, the development of new algorithms and improved versions of the existing ones is still a relevant research topic. Another relevant research topic is the development of hybrid metaheuristics that use characteristics of different existing algorithms to build new metaheuristics with better performance. It has been reported that hybrid algorithms developed from proper combinations of basic metaheuristics may result in more efficient and accurate algorithms (Blum and Roli 2008).

Different hybrid algorithms that combine two or more existing metaheuristics are available in the literature. For instance, Singh and Singh (2017) proposed a hybrid algorithm that uses the intensification capability of PSO and the diversification capability of GWO. An inertia constant is applied to control the equilibrium between diversification and intensification. They conducted experiments using several benchmark functions and observed that their algorithm presented a better performance in terms of both convergence speed and solution quality in comparison with the stand-alone versions of PSO and GWO.

Recently, Al-Tashi et al. (2019) proposed a binary version of the hybrid algorithm presented in Singh and Singh (2017). In this binary version, a new position update rule was proposed. This new rule uses a sigmoid function to adapt the original algorithm to solve binary problems. The algorithm outperformed state-of-the-art methods in the solution of feature selection problems.

Hybrid algorithms that combine characteristics of different metaheuristics may use a single population or a multi-population strategy. In a single population strategy, the search agents are considered as one unique population and a selection mechanism is used to select one of the basic metaheuristics to guide the search process during each phase of the algorithm. One example of a single population hybrid metaheuristic was proposed by Yue et al. (2020). In a multi-population strategy, however, the search agents are divided into a set of subpopulations. Each subpopulation evolves following the rules of one basic metaheuristic. Examples of multi-population strategies for hybrid metaheuristics can be found in Ma et al. (2019), Ma et al. (2020), and Rodrigues (2022).

In this context, the objective of this paper is to apply and evaluate the performance of a hybrid metaheuristic in the solution of the adaptive PID controller parameter tuning, which is a complex optimization problem that appears in several industry sectors. The hybrid metaheuristic considered in this paper, which is denoted by CSASOS and was originally proposed by Rodrigues (2022), is a multi-population, bio-inspired metaheuristic that combines the diversification capability of the Crow Search Algorithm (CSA) and the intensification capability of the Symbiotic Organisms Search (SOS). The hybridization between CSA and SOS is a promising combination due to the exploration and exploitation characteristics of the stand-alone versions of the basic algorithms. The CSASOS was successfully applied in the solution of a load-sharing optimization problem in a gas compressor station, which is also a complex optimization problem with real applications. Numerical experiments will be conducted with the stand-alone version of both CSA and SOS, in order to evaluate the benefits of combining the good characteristics of existing metaheuristics in the development of a hybrid algorithm.

The remaining sections of this paper are organized as follows. Section 2 introduces the PID controller parameter tuning problem. Section 3 presents a brief overview of CSA, SOS, and the hybrid CSASOS algorithm. Section 4 shows the results obtained in a numerical case study carried out to compare the performance of the CSASOS with the performance of the stand-alone versions of CSA and SOS. Concluding remarks and opportunities for future research are discussed in section 5.

2. PID Controller Parameter Tuning Problem

Proportional-Integral-Derivative (PID) controllers have been extensively used in a wide range of industrial applications due to their simplicity (Roeva and Slavov 2012). In order to enhance the performance of PID controllers in comparison with the use of fixed gains, nonlinear techniques such as sliding mode control (Kuo and Huang 2005) and adaptive control (Puchta et al. 2020) can be applied. The control signal of a standard PID controller, denoted by $u(e)$, is computed according to Equation (1).

$$u(e) = K_P e + K_I \int_0^t e(t)dt + K_D \frac{de}{dt} \tag{1}$$

where K_P is the proportional gain, K_I is the integral gain, K_D is the derivative gain, and e is the error between the reference value and the system output.

In order to improve the performance of standard PID controllers, adaptive gains based on Gaussian functions can be adopted to replace the fixed gains K_P, K_I, and K_D. Gaussian functions are smooth and have smooth derivatives, which are characteristics that avoid instability issues caused by abrupt changes in the control signal. The use of Gaussian functions for adaptive PID controllers has been previously discussed in Puchta et al. (2020) and Lucas et al. (2015).

The gains of the adaptive PID controller change according to the error based on a Gaussian function. A basic configuration of an adaptive PID controller is presented in Figure 1. The control signal of the controller, $u(e)$, is computed according to Equation (2).

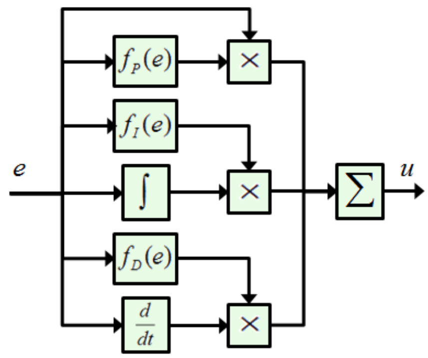

Figure 1. Configuration of an adaptive PID controller (adapted from Puchta et al. (2020))

$$u(e) = f_P e + f_I \int_0^t e(t)dt + f_D \frac{de}{dt} \tag{2}$$

where f_P, f_I, and f_D are the functions that replace the fixed gains and define the adaptive gains, computed according to Equations (3), (4), and (5), respectively.

$$f_P(e) = LB_P + (UB_P - LB_P) \cdot \exp\left(-q_P \cdot e^2\right) \tag{3}$$
$$f_I(e) = LB_I + (UB_I - LB_I) \cdot \exp\left(-q_I \cdot e^2\right) \tag{4}$$
$$f_D(e) = LB_D + (UB_D - LB_D) \cdot \exp\left(-q_D \cdot e^2\right) \tag{5}$$

where UB_P and LB_P are the upper and the lower bounds for the proportional gain function f_P; UB_I and LB_I are the upper and the lower bounds for the integral gain function f_I; UB_D and LB_D are the upper and the lower bounds for the derivative gain function f_D; and q_P, q_I, and q_D are the parameters that define the concavity openness of the curves for f_P, f_I, and f_D, respectively.

The design of the adaptive PID controller consists in defining the values of UB, LB, and q for the gain functions f_P, f_I, and f_D, resulting in a total of nine parameters. In order to reduce the number of parameters to be defined from 9 to 6, the simplified model defined by Equations (6) to (8) can be adopted.

$$LB_P = K_P / x_P; \quad UB_P = K_P \cdot x_P \tag{6}$$
$$LB_I = K_I / x_I; \quad UB_I = K_I \cdot x_I \tag{7}$$
$$LB_D = K_D / x_D; \quad UB_D = K_D \cdot x_D \tag{8}$$

where K_P, K_I, and K_D are the fixed gains obtained from a standard PID controller; and x_P, x_I, and x_D are parameters of the adaptive PID controller that must be defined.

In order to find the best set of parameters for the adaptive PID controller, a performance indicator must be chosen to quantify the quality of each candidate solution. In this paper, the ITAE (Integral Time Absolute Error) is adopted as the performance indicator since it has been recommended as the most suitable performance indicator for the adaptive PID controller design problem (Jagatheesan and Anand, 2012). The ITAE can be computed according to Equation (9).

$$ITAE = \int_0^{\infty} t \cdot |e(t)| dt \tag{9}$$

3. Theoretical Background

This section presents a brief description of the Crow Search Algorithm (CSA), the Symbiotic Organism Search (SOS) algorithm, and the hybrid CSASOS algorithm. These algorithms are applied in this paper to solve the PID controller parameter tuning problem.

3.1 Crow Search Algorithm

The Crow Search Algorithm (CSA) is a population-based, bio-inspired metaheuristic that emulates the behavior of crows. It is inspired by the ability of crows to hide and remember the location of the extra food. In addition, it simulates the ability of crows to protect their food from being stolen by other crows (Askarzadeh 2016).

The CSA algorithm considers a flock of n crows. In CSA, the *i*-th crow is represented by a *d*-dimensional vector $\mathbf{x}_i$, and has an associated *d*-dimensional vector $\mathbf{m}_i$ that represents the best location visited by the crow during the execution of the algorithm. The awareness probability, denoted by *AP*, is used to define whether the *j*-th crow notices that it is being followed by another crow. The position of the *i*-th crow in each iteration of the algorithm is updated according to Equation (10).

$$\mathbf{x}_i^* = \begin{cases} \mathbf{x}_i + r \cdot (\mathbf{m}_j - \mathbf{x}_i); & r \geq AP \\ rand^d(LB, UB); & otherwise \end{cases} \tag{10}$$

where $\mathbf{x}_i$ represents the original position of the crow, $\mathbf{x}_i^*$ is the updated position, r is a random variable chosen from a uniform distribution between 0 and 1, $\mathbf{m}_j$ is the best position visited by the crow, $rand^d$ is a random *d*-dimensional vector, LB is a vector containing the lower bound for each element of $rand^d$, and UB is a vector containing the upper bound for each element of $rand^d$.

3.2 Symbiotic Organisms Search

The Symbiotic Organism Search (SOS) is a population-based, bio-inspired algorithm that simulates symbiotic interactions adopted by organisms to survive in an ecosystem (Cheng and Prayogo, 2014). Each iteration of the SOS algorithm has three main phases: mutualism, commensalism, and parasitism.

In the mutualism phase, for each organism i, a different organism j is randomly selected. The solutions associated with organisms i and j, denoted respectively by $\mathbf{x}_i$ and $\mathbf{x}_j$, are used to generate new candidate solutions for both organisms according to Equations (11) and (12).

$$\mathbf{x}_i^* = \mathbf{x}_i + rand(0,1) \times (\mathbf{x}_{BEST} - (M \times BF_i)) \tag{11}$$

$$\mathbf{x}_j^* = \mathbf{x}_j + rand(0,1) \times (\mathbf{x}_{BEST} - (M \times BF_j)) \tag{12}$$

where BF_i and BF_j are, respectively, the benefit factors for organisms i and j computed according to Equations (13) and (14), respectively; $\mathbf{x}_{BEST}$ is the current best solution among all the organisms; and M is the Mutual Vector between organisms i and j, computed according to Equation (15). The original candidate solutions $\mathbf{x}_i$ and $\mathbf{x}_j$ are replaced with the new candidate solutions $\mathbf{x}_i^*$ and $\mathbf{x}_j^*$ whenever the new candidate solutions produce better solutions for the optimization problem under consideration.

$$BF_i = round(1 + rand(0,1)) \tag{13}$$

$$BF_j = round(1 + rand(0,1)) \tag{14}$$

$$M = (\mathbf{x}_i + \mathbf{x}_j)/2 \tag{15}$$

In the commensalism phase, for each organism i, a different organism j is randomly selected. A new candidate solution is generated for organism i, according to Equation (16). If the new solution $\mathbf{x}_i^*$ is better than $\mathbf{x}_i$, then $\mathbf{x}_i$ is replaced with $\mathbf{x}_i^*$.

$$\mathbf{x}_i^* = \mathbf{x}_i + rand(-1,1) \times (\mathbf{x}_{BEST} - \mathbf{x}_j) \tag{16}$$

In the parasitism phase, for each organism i, a parasite vector is generated by randomly modifying some dimensions of organism i. An organism j is randomly chosen. If the parasite vector has a better solution than organism j, it kills organism j and occupies its position in the ecosystem.

3.3 Hybrid CSASOS

This section briefly describes the CSASOS algorithm, which combines the good diversification capability of CSA with the good intensification capability of SOS. A more detailed description of the CSASOS algorithm can be found in Rodrigues (2022). Figure 2 shows a flowchart of the hybrid CSASOS algorithm.

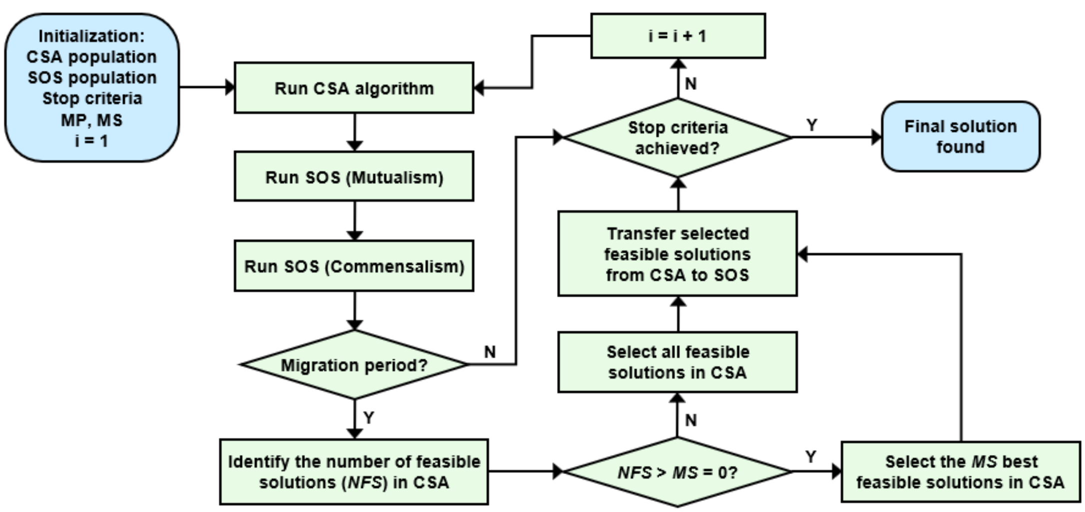

Figure 2. Flowchart of the hybrid CSASOS algorithm

The CSASOS divides the search agents into two subpopulations. One subpopulation follows the search rules of the CSA algorithm, while the other subpopulation follows the search rules of the SOS algorithm. In CSASOS, CSA is responsible for finding promising regions of the search space. After a predefined number of iterations, denoted by migration period (*MP*), the best solutions in the CSA subpopulation are transferred to the SOS subpopulation, which performs a refined search in the promising regions to find the best solution for the optimization problem under consideration. In CSASOS, only two phases of SOS are used: the mutualism and the commensalism phases. These two phases are responsible for the intensification capability of SOS. The parasitism phase, which is responsible for the diversification in SOS, is eliminated in CSASOS because the diversification capability of CSASOS is provided by the CSA.

The population initialization strategy used by a metaheuristic affects its performance. In CSASOS, the opposition-based learning strategy is adopted to improve the diversification of the initial CSA subpopulation. The motivation for using the opposition-based learning strategy is that a pair composed by a randomly generated solution and its opposite has a higher probability to be closer to the optimal solution when compared with a pair composed of two randomly generated solutions (Tizhoosh, 2005). In CSASOS, half of the initial solutions of the CSA subpopulation are randomly generated, while the remaining initial solutions are the opposite ones. All the initial solutions of the SOS subpopulation are randomly generated.

In this paper, the PID controller parameter tuning is considered as a constrained optimization problem, i.e. each element of a candidate solution must be in the interval defined by the corresponding lower and upper bounds. However, during the execution of the algorithm, candidate solutions that violate these bounds can be generated. In order to handle these situations, a bound constraint handling method must be applied to move the values of each element of candidate solutions to the corresponding valid interval. Several bound constraint handling methods can be found in the literature such as midpoint base, rand base, and resampling, among others (Biedrzycki et al., 2019). In CSASOS, the rand base method is used due to its random behavior that increases the diversification capability of the algorithm.

Another crucial aspect in multi-population metaheuristics is the mechanism used to guide the communication between the different subpopulations. This mechanism is responsible for the information-sharing capability of the algorithm. In CSASOS, the two subpopulations communicate with each other after a predefined number of iterations, denoted by migration period (*MP*). At the end of each migration period, the best feasible solutions in the CSA subpopulation migrate to the SOS subpopulation. The maximum number of solutions that migrate at the end of each migration period is denoted by migration size (*MS*). Both *MP* and *MS* are hyperparameters that must be defined.

In order to improve the search process, the first stage of the search strategy should focus on the exploration capability, generating candidate solutions along all regions of the search space. As the execution progress, the exploitation capability is intensified to refine the search in the neighborhood for the best solutions (Morales-Castañeda et al., 2020). During the earlier iterations of CSASOS, the CSA subpopulation is larger than the SOS subpopulation to provide a better diversification capability. During the execution of the algorithm, the migration of solutions decreases the CSA subpopulation and increases the SOS subpopulation to enhance the intensification capability of CSASOS.

4. Case Study

This section presents a case study that illustrates the application of CSASOS in the solution of an adaptive PID controller parameter tuning. The performance of CSASOS is compared with the performance of the stand-alone versions of CSA and SOS to evaluate the improvements obtained by combining the best characteristics of each algorithm. In the PID controller parameter tuning optimization problem considered in this paper, feasible solutions are the ones that provide dynamic responses for the controlled system that meet pre-defined response requirements.

4.1 System Description

The case study consists in designing a PID controller for a speed control loop of a DC motor. Figure 3 presents a schematic of the DC motor. The dynamic behavior of the DC motor can be described by the model presented in Equations (17) to (20) (Emhemed and Mamat 2012).

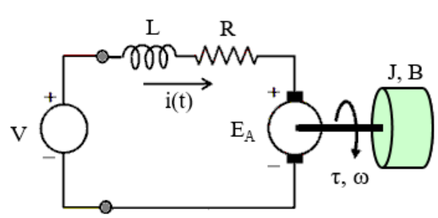

Figure 3. DC motor schematic

$$V = Ri + L\frac{di}{dt} + E_A \tag{17}$$

$$E_A = K_m \omega \tag{18}$$

$$T_m = J \frac{d\omega}{dt} + B\omega \tag{19}$$

$$T_e = K_t i \tag{20}$$

where V is the DC motor input voltage, R is the armature resistance, i is the armature current, L is the armature inductance, E_A is the back electromotive force, K_m is a motor constant, ω is the angular speed, T_m is the mechanical torque, J is the load moment of inertia, B is the viscous friction coefficient, T_e is the electrical torque, and K_t is the motor electrical torque constant.

Figures 4 and 5 show, respectively, the implementations of a standard PID controller with fixed gains and an adaptive PID controller for speed control of the DC motor with a derivative filtering in a Matlab Simulink® environment.

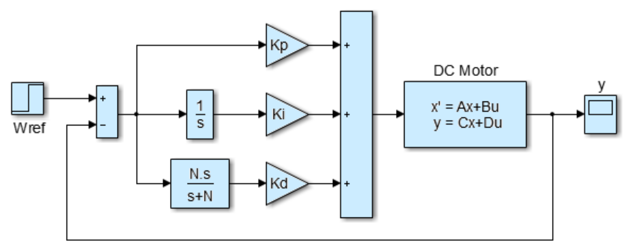

Figure 4. Implementation of a standard PID controller in a Matlab/Simulink® environment

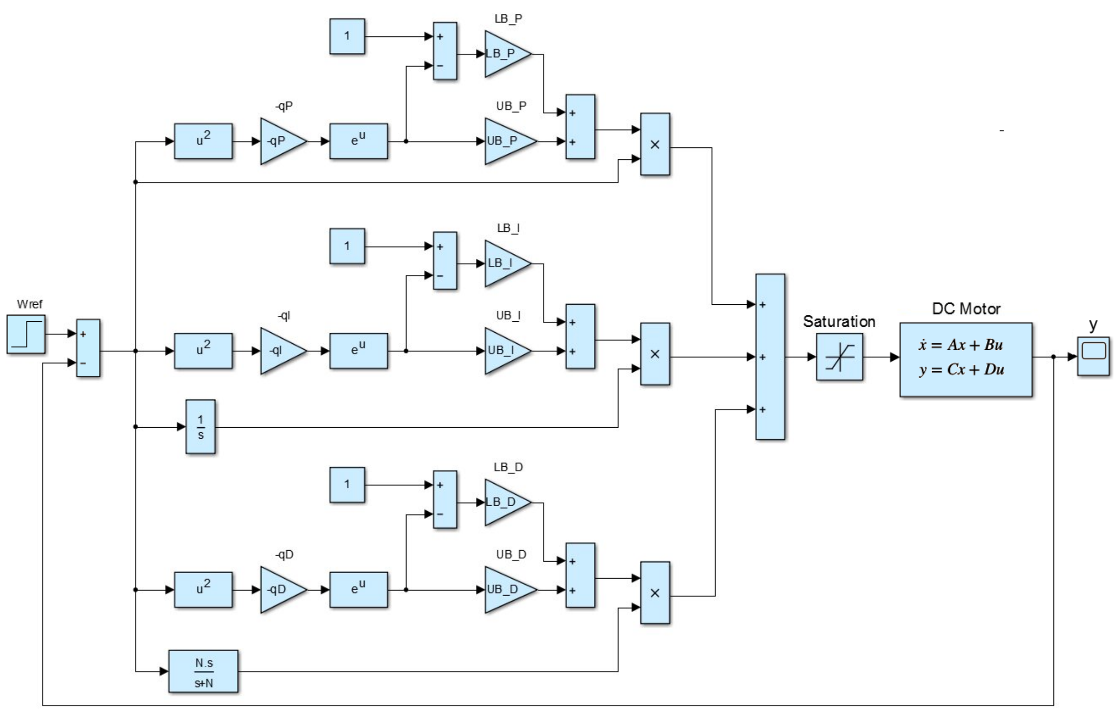

Figure 5. Implementation of an adaptive PID controller in a Matlab/Simulink® environment

4.2 Simulation Data

Table 1 shows the parameter values used in the simulation of the DC motor. The dynamic behavior of the DC motor can be represented by the state-space model presented in Equations (21) and (22).

Table 1. Parameter values used in the simulation of the DC motor

Parameter	Value	Unit
R	0.50	Ω
L	0.10	H
K_m	0.25	$V \cdot s / rad$
J	0.40	$kg \cdot m^2$
B	0.80	$N \cdot m \cdot s / rad$
K_t	0.12	$N \cdot m / A$

$$\begin{bmatrix} di/dt \\ d\omega/dt \end{bmatrix} = \begin{bmatrix} -5 & -2.5 \\ 0.3 & -2 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \end{bmatrix} \cdot V \quad (21)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} \quad (22)$$

The ITAE value for each candidate solution obtained during the execution of each algorithm was computed by simulating the step response of the DC motor using the model presented in Figure 5. In order to compute the ITAE indicator, the step input was applied for t = 0.5s. The ITAE indicator was computed considering the interval between zero and 3s. The saturation block represents the range of input voltages that can be applied to the DC motor and has a minimum value of -24V, and a maximum value of +24V. For the derivative filtering, a value of 100 was used for parameter N.

As mentioned earlier, besides the control parameters of CSA and SOS, the hybrid CSASOS has two additional control parameters: the migration period *MP*, and the maximum migration size *MS*. The additional control parameters define how the size of the subpopulations will change during the execution of the algorithm.

A stop criterion of 10,000 objective function evaluations was used for all metaheuristics. A total population of 55 was used for all the algorithms. For CSA and CSASOS, an awareness probability *AP* of 0.1 and a flight length *FL* of 2 were used. These values were defined based on experimental tests. For the hybrid CSASOS, the initial size of the CSA and the SOS subpopulations were 50 and 5, respectively. A migration period *MP* of 20 and a maximum migration size *MS* of 5 were used. These values were also defined based on experimental tests.

4.3 Simulation Results

First, a standard PID controller was designed to act as a reference for the simplified model of the adaptive PID controller. The design was based on the pole placement method, and the requirements considered were a maximum overshoot of 15% and a maximum peak time of 0.5s. The fixed gains obtained are presented in Table 2.

Table 2. Parameter values used in the simulation of the DC motor

Parameter	Value
K_P	21.9
K_I	83.1
K_D	2.8

Figure 6 shows a comparison between the step responses obtained with the standard PID controller and with the adaptive PID controller. It can be noticed that, although the step response obtained with the standard PID controller meets the design requirements, the use of an adaptive PID controller provides a much better response in terms of overshoot and peak time.

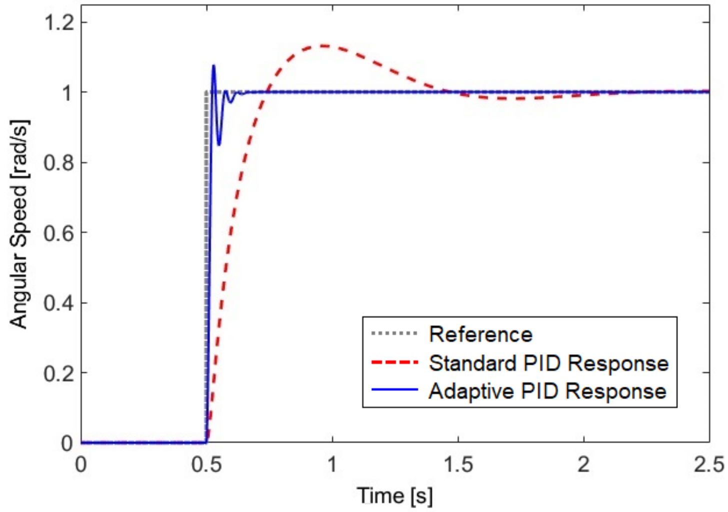

Figure 6. Comparison between the step responses obtained with a standard and an adaptive PID controller

Then, we designed an adaptive PID controller using the proposed CSASOS algorithm. For comparison purposes, we also designed adaptive PID controllers using the stand-alone versions of CSA and SOS, which are the basic metaheuristics combined in CSASOS. For each algorithm, a Monte Carlo approach with 50 repetitions was conducted. Table 3 presents the parameters obtained in the best solution found by each algorithm. In addition, Table 3 shows the statistics for the ITAE indicator obtained with each algorithm over all the 50 repetitions. Based on these results, it is possible to observe that the hybrid CSASOS outperformed the stand-alone versions of both CSA and SOS in terms of the performance indicator ITAE. The CSA algorithm presented the highest average value and the highest standard deviation of ITAE. It indicates that CSA has a low exploitation capability in comparison with SOS. CSASOS presented the smallest average and the smallest standard deviation for the ITAE indicator, showing its better capacity to solve the optimization problem under consideration.

Table 3. Simulation results

Parameter	CSA	SOS	CSASOS
x_p	97.23	99.55	99.59
x_i	3.00	5.40	5.41
x_d	17.62	25.38	26.00
q_p	2.50	0.50	0.51
q_i	8.97	0.50	1.08
q_d	0.52	0.51	0.50
$ITAE \quad (\times 10^{-4})$	4.57 ± 0.41	3.97 ± 0.22	**3.71 ± 0.14**

5. Conclusions

In this paper, we investigated the benefits of combining the good characteristics of two metaheuristics in the development of a hybrid algorithm. In CSASOS, the good diversification capability of CSA and the good intensification capability of SOS are used. The CSA subpopulation finds promising regions of the search space while the SOS subpopulation intensifies the search in the promising regions to find the final solution for the optimization problem under consideration.

A case study was presented in which the proposed CSASOS is used to define the parameters of an adaptive PID controller, which is a problem that appears in many industrial applications. The results showed that the hybrid CSASOS outperformed the stand-alone versions of both CSA and SOS in terms of solution quality. The ITAE (Integral Time Absolute Error) was used as the indicator to assess the performance of the algorithms. CSASOS presented the best average value for the ITAE indicator. It also presented the smallest standard deviation, indicating that it consistently finds good solutions.

Future research may extend the scope of this paper by investigating the performance of the hybrid CSASOS in other classes of optimization problems such as multi-objective problems and binary problems. Another possible extension of the work presented in this paper is to investigate new strategies to guide the communication between the two subpopulations.

Acknowledgements

The authors acknowledge the support of the Brazilian National Council for Scientific and Technological Development - CNPq (grant 423023/2018-7).

References

Al-Tashi, Q., Abdul Kadir, S. J., Rais, H. M., Mirjalili, S. and Al-hussian, H., Binary optimization using hybrid grey wolf optimization for feature selection, *IEEE Access*, vol. 7, pp. 39496–39508, 2019.

Askarzadeh, A., A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, *Computers & Structures*, vol. 169, pp. 1–12, 2016.

Biedrzycki, R., Arabas, J. and Jagodzinski, D., Bound constraints handling in differential evolution: An experimental study, *Swarm and Evolutionary Computation*, vol. 50, p. 100453, 2019.

Blum, C. and Roli, A., Hybrid metaheuristics: An introduction, in *Hybrid Metaheuristics: An Emerging Approach to Optimization*, Blum, C., Roli, A. and Sampels, M., Eds. Springer-Verlag Berlin Heidelberg, 2008, pp. 1–30.

Cheng, M. Y. and Prayogo, D., Symbiotic organisms search: A new metaheuristic optimization algorithm, *Computers & Structures*, vol. 139, pp. 98–112, 2014.

Dorigo, M., Maniezzo, V. and Colorni, A. The ant system: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 26, no. 1, pp. 29-41, 1996.

Emhemed, A. A. A. and Mamat, R. B., Modelling and simulation for industrial dc motor using intelligent control, *Procedia Engineering*, vol. 41, pp. 420–425, 2012.

Holland, J. H., *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence*. Cambridge, MA, USA: MIT Press, 1992.

Jagatheesan, K. and Anand, B., Dynamic performance of multi-area hydro thermal power systems with integral controller considering various performance indices methods, *Proceedings of the 2012 International Conference on Emerging Trends in Science, Engineering and Technology*, pp. 474–478, Tiruchirappalli, India, 2012.

Kennedy, J. and Eberhart, R. C., Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, 1995.

Kuo, T. C. and Huang, Y. J., A sliding mode PID-controller design for robot manipulators, *Proceedings of the 2005 International Symposium on Computational Intelligence in Robotics and Automation*, pp. 625–629, Espoo, Finland, 2005.

Lucas, R., Oliveira, R. M., Nascimento, C. B. and Kaster, M. S., Performance analysis of an adaptive Gaussian nonlinear PID control applied to a step-down CC-CC converter, *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 1022–1026, Buzios, Brazil, 2015.

Ma, H., Fei, M., Jiang, Z., Li, L., Zhou, H. and Crookes, D., A multipopulation-based multiobjective evolutionary algorithm, *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 689–702, 2020.

Ma, H., Shen, S., Yu, M., Yang, Z., Fei, M. and Zhou, H., Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey, *Swarm and Evolutionary Computation*, vol. 44, pp. 365–387, 2019.

Mirjalili, S., Mirjalili, S. M. and Lewis, A., Grey wolf optimizer, *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

Morales-Castañeda, B., Zaldívar, D., Cuevas, E., Fausto, F. and Rodríguez, A., A better balance in metaheuristic algorithms: Does it exist?, *Swarm and Evolutionary Computation*, vol. 54, p. 100671, 2020.

Puchta, E. D. P., Siqueira, H. V. and Kaster, M. D. S., Optimization tools based on metaheuristics for performance enhancement in a Gaussian adaptive PID controller, *IEEE Transactions on Cybernetics*, vol. 50, no. 3, pp. 1185–1194, 2020.

Rodrigues, L. R., A hybrid multi-population metaheuristic applied to load-sharing optimization of gas compressor stations, *Computers & Electrical Engineering*, vol. 97, pp. 107632, 2022.

Roeva, O. and Slavov, T., PID controller tuning based on metaheuristic algorithms for bioprocess control, *Biotechnology & Biotechnological Equipment*, vol. 26, no. 5, pp. 3267–3277, 2012.

Singh, N. and Singh, S. B., Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance, *Journal of Applied Mathematics*, vol. 2017, p. 2030489, 2017.

Slowik, A. and Kwasnicka, H., Nature inspired methods and their industry applications – swarm intelligence algorithms, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1004-1015, 2018.

Storn, R. and Price, K., Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

Tizhoosh, H. R., Opposition-based learning: A new scheme for machine intelligence, *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol. 1, pp. 695–701, Vienna, Austria, 2005.

Yue, Z., Zhang, S. and Xiao, W., A novel hybrid algorithm based on grey wolf optimizer and fireworks algorithm, *Sensors*, vol. 20, no. 7, p. 2147, 2020.

Biographies

Leonardo Ramos Rodrigues is currently a technologist at the Institute of Aeronautics and Space (IAE), Brazil, where he performs research in the field of aerospace technologies. He received his Master's Degree (2008) and D.Sc. Degree (2013) in Aeronautical Engineering from the Instituto Tecnológico de Aeronáutica (ITA), Brazil. From 2006 to 2015 he worked at EMBRAER S.A., in Brazil, where he was part of the PHM (Prognostics and Health Monitoring) research group performing research in the fields of PHM techniques and Data Analytics solutions for the aviation industry.

João Paulo Pordeus Gomes is currently an Assistant Professor at the Federal University of Ceará, in Brazil. He received his Master's Degree (2006) in Aeronautical Engineering, and his Doctorate's Degree (2011) in Electronic Engineering from the Instituto Tecnológico de Aeronáutica (ITA), Brazil. He worked at EMBRAER S.A. between 2006 and 2013 as a Technology Development Engineer focusing on fault monitoring applications for aeronautical systems.