LII Simpósio Brasileiro de Pesquisa Operacional
João Pessoa - PB, 3 a 5 de novembro de 2020

AN IMPROVED SYMBIOTIC ORGANISMS SEARCH APPLIED TO ADAPTIVE PID CONTROLLER DESIGN

Leonardo Ramos Rodrigues
Instituto de Aeronáutica e Espaço - IAE
Praça Marechal Eduardo Gomes, 50, São José dos Campos, SP, Brasil, 12228-904
leonardolrr2@fab.mil.br

Daniel Borba Paulino Coelho
EMBRAER S.A.
Rua Sete, 498, Belo Horizonte, MG, Brazil, 31035-199
daniel.coelho@embraer.com.br

ABSTRACT

Nature-inspired algorithms have become popular due to their efficiency in finding close-to-optimal solutions for complex optimization problems. The development of new metaheuristic algorithms and variant versions of existing ones is a hot research topic. The Symbiotic Organisms Search (SOS) is a nature-inspired, population-based optimization metaheuristic that simulates symbiotic relationships observed in nature. It has been successfully applied to solve different optimization problems. In this paper, we propose a modified version of SOS, named MSOS, that adds an allocation step to the original SOS to select organisms for each phase of the algorithm in such a way that the probability of generating improvements in the current solutions is increased. A numerical example considering the design of an adaptive PID controller is presented to illustrate the application of the proposed algorithm. The experiments show that the proposed MSOS outperformed the original SOS for the optimization problem under consideration.

KEYWORDS. Metaheuristic. Symbiotic Organisms Search. Adaptive Control. Optimization.

Paper topic: MH - Metaheuristics.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

1. Introduction

Nature-inspired metaheuristic optimization algorithms have become very popular due to their capability of finding good solutions for complex optimization problems in an acceptable computational time. The literature contains a wide variety of metaheuristic algorithms such as the Ant Colony Optimization (ACO) [Dorigo, 1992] and the Genetic Algorithm (GA) [Holland, 1992], among others. Despite the high number of algorithms available, the development of new metaheuristic algorithms and variant versions of existing ones is still a hot research topic.

The Symbiotic Organisms Search (SOS) algorithm was originally proposed by Cheng and Prayogo [2014]. The SOS algorithm is inspired by the symbiotic interactions between organisms in an ecosystem. It has three main phases: mutualism, commensalism, and parasitism. Recent works have reported the good performance obtained with SOS in the solution of different problems such as power flow optimization [Duman, 2017], and scheduling optimization [Abdullahi et al., 2016].

Modified versions of SOS have also been developed. For instance, Tejani et al. [2016] proposed a variant of SOS that uses adaptive parameters and a better compromise between exploration and exploitation was observed. Nama et al. [2016] included a Predation Phase in the SOS algorithm, which simulates a fourth symbiotic relationship. The literature also contains hybrid algorithms between SOS and other algorithms [Sulaiman et al., 2018], [Choe et al., 2018]. Modified versions of SOS to solve multi-objective optimization problems have also been proposed [Tran et al., 2016].

In this paper, we propose a new modified version of SOS, denoted by MSOS. In the proposed algorithm, we add an allocation step to SOS to select the most suitable organism for each phase of the algorithm. The organisms are assigned to each phase to maximize the probability of generating better organisms, i.e. solutions with better fitness values. The performance of MSOS is evaluated in the solution of an adaptive PID controller design problem.

The remaining sections of this paper are organized as follows. Section 2 introduces the original SOS algorithm and describes the adaptive PID controller design problem. Section 3 presents the proposed MSOS algorithm. Section 4 presents a numerical case study conducted to compare the performance of MSOS with the performance of the original SOS to solve the adaptive PID controller design optimization problem. Concluding remarks are presented in section 5.

2. Theoretical Background

2.1. Symbiotic Organisms Search

The Symbiotic Organism Search (SOS) algorithm is a population-based, nature-inspired metaheuristic optimization algorithm. It simulates symbiotic interactions used by organisms to survive in an ecosystem. Symbiosis is a relationship between two distinct species. The most common symbiotic relationships observed in nature are mutualism, commensalism, and parasitism [Cheng and Prayogo, 2014].

Mutualism is a symbiotic relationship between two different species in which both benefit from each other. Mutualistic relationships can be either obligate or facultative. In an obligate mutualism relationship, one organism cannot survive without the other. In a facultative mutualism, each organism can survive by itself. However, the relationship provides benefits for both organisms if they live together. Commensalism is a symbiotic relationship between two different species in which one species benefits from the relationship while the other is unaffected. Parasitism is a symbiotic relationship between two different species in which one benefits and the other is harmed.

The SOS algorithm simulates the different symbiotic relationships to find close-to-optimal solutions for the optimization problem under consideration. Each iteration of the SOS algorithm has three main phases: the mutualism phase, the commensalism phase, and the parasitism phase.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

During each phase of the algorithm, one symbiotic relationship is simulated using a mathematical operator. The generation of new candidate solutions occurs according to the characteristics of the corresponding relationship [Talatahari, 2016]. During the mutualism phase, new candidate solutions are generated for both the organisms involved because both organisms benefit from this relationship. During the commensalism phase, a new candidate solution is generated for one organism only, while the other organism remains unchanged. In the parasitism phase, one of the organisms can be replaced with a new random solution. This process is repeated until the stop criteria are met. The mathematical operators used in each phase of SOS are described in the next section.

2.1.1. SOS Operators

During the mutualism phase (the first phase in each iteration of SOS), for each organism i in the ecosystem, an organism j, with $j \neq i$, is randomly selected to interact with organism i. The solutions associated with organisms i and j are denoted by X_i and X_j, respectively. New candidate solutions for both organisms i and j are generated according to Equations (1) and (2), respectively.

$$X_i^* = X_i + \texttt{rand}(0,1) \times (X_{best} - (M \times BF_i)) \tag{1}$$
$$X_j^* = X_j + \texttt{rand}(0,1) \times (X_{best} - (M \times BF_j)) \tag{2}$$

where BF_i and BF_j are the benefit factors for organisms i and j computed according to Equations (3) and (4), respectively; X_{best} is the solution associated with the organism with the best solution in the ecosystem; and M is the Mutual Vector that represents the relationship between organisms i and j, computed according to Equation (5). If the new candidate solution X_i^* is better than X_i, then X_i is replaced with X_i^* for the next iteration. Similarly, if X_j^* is better than X_j, it replaces X_j for the next iteration.

$$BF_i = \texttt{round}\,(1 + \texttt{rand}(0,1)) \tag{3}$$
$$BF_j = \texttt{round}\,(1 + \texttt{rand}(0,1)) \tag{4}$$
$$M = \frac{X_i + X_j}{2} \tag{5}$$

During the commensalism phase (the second phase in each iteration of SOS), for each organism i in the ecosystem, an organism j, with $j \neq i$, is randomly selected to interact with organism i. However, in a commensalism interaction, only organism i benefits from the interaction, while organism j is not affected. Thus, a new candidate solution is generated for organism i only, and is computed according to Equation (6). If the new candidate solution X_i^* is better than X_i, then X_i is replaced with X_i^* for the next iteration of the algorithm.

$$X_i^* = X_i + \texttt{rand}(-1,1) \times (X_{best} - X_j) \tag{6}$$

During the parasitism phase (the last phase in each iteration of SOS), for each organism i in the ecosystem, a parasite vector is generated. This parasite vector is created by duplicating the current solution of organism i and then modifying some dimensions that are randomly selected. Then, for each organism i in the ecosystem, an organism j, with $j \neq i$, is randomly chosen from the population to act as the host of this parasite vector. If the parasite vector has a better solution than

organism j, it kills organism j and occupies its position in the ecosystem. Otherwise, organism j is considered immune to the parasite vector, and the parasite vector is discarded.

2.2. Adaptive PID Controller

Proportional-Integral-Derivative (PID) controllers are widely used in industrial applications due to their simplicity and well-known behavior [Roeva and Slavov, 2012]. Nonlinear techniques such as sliding mode [Kuo and Huang, 2005], optimal control [Das et al., 2013], and adaptive control [Puchta et al., 2020] can be used to improve the performance of PID controllers. The output u of a standard PID controller is computed according to Equation (7).

$$u(e) = K_p \cdot e + K_i \int_0^t e + K_d \frac{de}{dt} \tag{7}$$

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, the use of adaptive PID controllers based on Gaussian functions has been proposed [Puchta et al., 2020]. In this approach, the gains of the PID controller vary as a function of the error based on a Gaussian function. Due to the smooth characteristic of Gaussian functions, the adaptive PID controller does not present abrupt changes in the gains. Figure 1 shows the structure of an adaptive PID controller. In this paper, the output u of the adaptive PID controller is computed according to Equation (8).

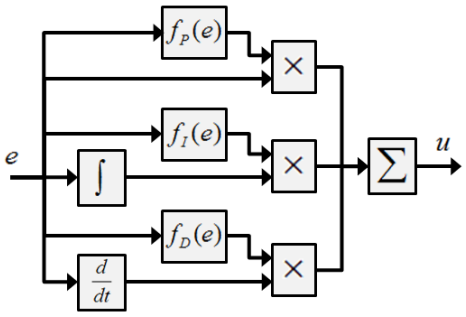


Figure 1: Structure of an adaptive PID controller (adapted from Puchta et al. [2020])

$$u(e) = f_P \cdot e + f_I \int_0^t e + f_D \frac{de}{dt} \tag{8}$$

where f_P, f_I, and f_D are computed according to Equations (9), (10), and (11), respectively.

$$f_P(e) = lb_p + (ub_p - lb_p) \cdot \exp(-q_p \cdot e^2) \tag{9}$$
$$f_I(e) = lb_i + (ub_i - lb_i) \cdot \exp(-q_i \cdot e^2) \tag{10}$$
$$f_D(e) = lb_d + (ub_d - lb_d) \cdot \exp(-q_d \cdot e^2) \tag{11}$$

where ub_p and lb_p are the upper and the lower bounds for f_P; ub_i and lb_i are the upper and the lower bounds for f_I; ub_d and lb_d are the upper and the lower bounds for 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 each function f_P, f_I, and f_D, which is a total of nine parameters. In order to reduce the number of parameters to be defined, a simplified model defined by (12) can be used.

$$\begin{cases} lb_p = K_p/x_p; & ub_p = K_p \cdot x_p \\ lb_i = K_i/x_i; & ub_i = K_i \cdot x_i \\ lb_d = K_d/x_d; & ub_d = K_d \cdot x_d \end{cases} \tag{12}$$

where K_p, K_i, and K_d are the gains of the standard PID controller; and x_p, x_i, and x_d are parameters of the adaptive PID controller.

Using this simplified model, we first design a standard PID controller. Then, there will be six parameters to be defined during the design of the adaptive PID controller: $[x_p, x_i, x_d, q_p, q_i, q_d]$.

2.2.1. Performance Indicator

In order to define the best set of parameters for the adaptive PID controller, it is necessary to define an indicator that quantifies the performance of each candidate solution. Jagatheesan and Anand [2012] investigated the use of different performance indicators such as the Integral Square Error (ISE), the Integral Absolute Error (IAE), and the Integral Time Square Error (ITSE), among others. They concluded that the indicator with the best performance for the adaptive PID controller design problems is the Integral Time Absolute Error (ITAE), which is computed according to Equation (13). In this paper, we use the ITAE to compute the fitness value for each candidate solution.

$$ITAE = \int_0^\infty t \cdot |e(t)| dt \tag{13}$$

3. Proposed MSOS Algorithm

This section presents the proposed variant of SOS, denoted by MSOS. The motivation for the proposed MSOS is to select organisms that are more suitable for each symbiotic relationship.

In SOS, for each organism i in the ecosystem, an organism j, with $j \neq i$ is randomly selected to interact with organism i in a predefined symbiotic relationship. However, when this procedure is used, an organism j can be selected to a relationship for which it is not suitable. For example, an organism j with a good solution could be randomly chosen for the parasitism phase. It is easy to see that this organism is unlikely to be eliminated by the parasite vector. Also, an organism j with a solution that is similar to the best organism in the ecosystem (or even the best organism in the ecosystem itself) could be randomly chosen for the commensalism phase. In this case, the new candidate solution X_i^* computed according to Equation (6) would be very close to the current solution X_i, and the probability of generating a better solution would be low.

In the proposed MSOS, for each organism i in the ecosystem, three organisms j_1, j_2 and j_3, with $j_k \neq i$ for $k \in \{1,2,3\}$, are randomly chosen from the ecosystem. The solution associated with each organism j_k is denoted by X_{jk}, for $k \in \{1,2,3\}$. At this time, the symbiotic relationships between organism i and organisms j_k are unknown. Then, an allocation step is performed in order to assign a symbiotic relationship to each organism j_k. The main goal of the allocation step is to define a relationship for each organism j_k such that the probability of generating better solutions is increased.

For the allocation step, three weights are defined for each organism j_k. These weights, denoted by α_{mk}, α_{ck}, and α_{pk}, represent the suitability of each organism j_k for a mutualism, commensalism and parasitism relationship, respectively. Weights α_{mk}, α_{ck}, and α_{pk} are obtained according to Equations (14), (15), and (16), respectively.

$$\alpha_{mk} = \|X_{best} - M_k\| \tag{14}$$
$$\alpha_{ck} = \|X_{best} - X_{jk}\| \tag{15}$$
$$\alpha_{pk} = \|X_{jk} - X_P\| \tag{16}$$

where $\|\cdot\|$ is the Euclidean norm, M_k is the mutual vector between organisms i and j_k computed according to Equation (5), and X_P is the solution associated with the parasite vector.

Each weight quantifies the potential that a relationship between organisms i and j will generate new candidate organisms that are different from the original organisms. The goal is to increase the diversification capability of the algorithm, which reduces the probability of premature convergence of the algorithm [Črepinšek et al., 2013].

After computing weights α_{mk}, α_{ck}, and α_{pk}, an allocation optimization problem is solved according to Equations (17) to (24).

$$\max J = \sum_{k=1}^{3} (\alpha_{mk}\gamma_{mk} + \alpha_{ck}\gamma_{ck} + \alpha_{pk}\gamma_{pk}) \tag{17}$$
$$\text{s.t.} \quad \gamma_{m1} + \gamma_{m2} + \gamma_{m3} = 1 \tag{18}$$
$$\gamma_{c1} + \gamma_{c2} + \gamma_{c3} = 1 \tag{19}$$
$$\gamma_{p1} + \gamma_{p2} + \gamma_{p3} = 1 \tag{20}$$
$$\gamma_{m1} + \gamma_{c1} + \gamma_{p1} = 1 \tag{21}$$
$$\gamma_{m2} + \gamma_{c2} + \gamma_{p2} = 1 \tag{22}$$
$$\gamma_{m3} + \gamma_{c3} + \gamma_{p3} = 1 \tag{23}$$
$$\gamma_{mk}, \gamma_{ck}, \gamma_{pk} \in \{0,1\} \tag{24}$$

where γ_{mk} is a binary decision variable that assumes value 1 if organism j_k is assigned to the mutualism phase and zero otherwise, γ_{ck} is a binary decision variable that assumes value 1 if organism j_k is assigned to the commensalism phase and zero otherwise, and γ_{pk} is a binary decision variable that assumes value 1 if organism j_k is assigned to the parasitism phase and zero otherwise.

Equations (18) to (20) are the constraints that ensure that each relationship is allocated to one organism only. Equations (21) to (23) are the constraints ensure that each organism is allocated to one relationship only. Equation (24) is the constraint that limits the range of values that the decision variables γ_{mk}, γ_{ck}, and γ_{pk} may assume.

It is worth pointing out that the optimization problem in the proposed allocation step has a fixed size of 3×3, which does not depend on the size of the main optimization problem under consideration. Since the allocation problem has a very small size, the computational cost is very small. Different algorithms can be used to solve the allocation problem. One example is the Hungarian method [Kuhn, 1955].

The procedure to define the relationships between organism i and organisms j_k in the proposed MSOS is summarized in Algorithm 1.

Algorithm 1 Relationship definition in MSOS

for each organism i **do**
  randomly select 3 distinct organisms $j_k \neq i$
  for each organism j_k **do**
    compute $\alpha_m^{(k)}$
    compute $\alpha_c^{(k)}$
    compute $\alpha_p^{(k)}$
  end for
  solve the allocation problem to define j_m, j_c and j_p
  allocate organism j_m to the mutualism phase
  allocate organism j_c to the commensalism phase
  allocate organism j_p to the parasitism phase
end for

4. Numerical Experiments

4.1. System Description

As mentioned earlier, in this paper we consider the adaptive PID controller design problem. The PID controller is used in a speed control loop of a DC motor. Figure 2 presents the DC motor schematic. The DC motor model is described by Equations (25) to (28) [Emhemed and Mamat, 2012].

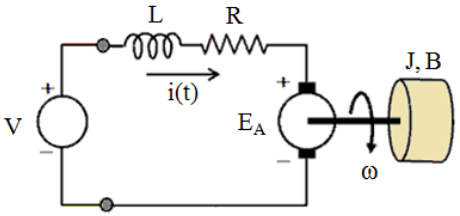


Figure 2: DC motor schematic

$$V = Ri + L\frac{di}{dt} + E_A \quad (25)$$

$$E_A = K_m\omega \quad (26)$$

$$T_m = J\frac{d\omega}{dt} + B\omega \quad (27)$$

$$T_e = K_t i \quad (28)$$

where V is the 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 electrical torque, and K_t is the electrical torque constant.

Figure 3 shows a Matlab Simulink® implementation of a standard PID controller with fixed gains for speed control of the DC motor with a derivative filtering. As mentioned earlier, in this case study we consider an adaptive PID controller and use the ITAE as a performance indicator.

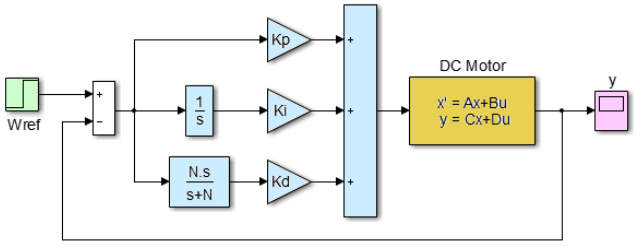


Figure 3: Matlab Simulink® implementation of a standard PID controller with fixed gains

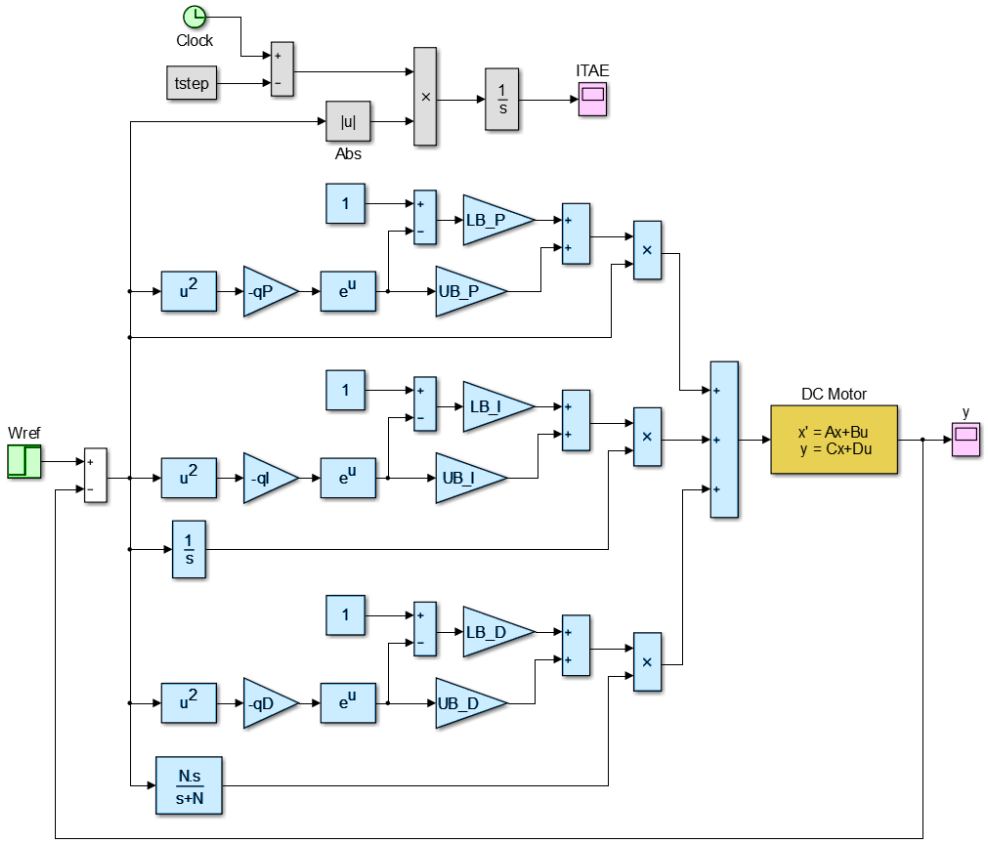


Figure 4: Matlab Simulink® implementation of an adaptive PID controller

Figure 4 shows the Matlab Simulink® implementation of the adaptive PID controller used in the experiments.

4.2. Simulation Data

Table 1 shows the values for the DC motor parameters used in the simulations.

Based on the DC motor model presented in Equations (25) to (28) and the parameter values presented in Table 1, the dynamic behavior of the model can be represented by the state-space model presented in Equations (29) and (30).

Table 1: DC motor parameters

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

$$\begin{bmatrix} di/dt \\ d\omega/dt \end{bmatrix} = \begin{bmatrix} -5.0 & -2.5 \\ 0.3 & -2.0 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \end{bmatrix} \cdot V \tag{29}$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} \tag{30}$$

The fitness value associated with each candidate solution was computed by simulating the step response of the system using the model presented in Figure 4. In order to compute the fitness value, the step input was applied for $t = 0.5$s, and the ITAE indicator was computed for the interval [0,3]s. For both SOS and MSOS, we used a population size of 20 and a maximum iteration number of 200. These values were defined based on experimental tests.

4.3. Simulation Results

Firstly, we designed a standard PID controller to act as a reference for the adaptive PID controller. This design was based on the pole placement method, and the requirements used were a maximum overshoot of 15% and a maximum peak time of 500ms. For the derivative filtering, a value of 100 was used for parameter N. The gains obtained are presented in Table 2. Secondly, we designed an adaptive PID controller as described in Equations (9) to (11) using the original SOS algorithm.

Table 2: Parameters of the standard PID controller

Parameter	Value
K_p	21.9
K_i	83.1
K_d	2.8

Figure 5(a) shows a comparison between the step responses obtained with the standard PID controller and with the adaptive PID controller designed using the SOS algorithm. 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 provided a much better response in terms of overshoot and peak time.

Finally, we designed an adaptive PID controller using the proposed MSOS algorithm. Figure 5(b) shows a comparison between the step responses obtained with the adaptive PID controllers designed with both SOS and MSOS. Table 3 shows the parameters obtained with both algorithms. It also shows the statistics for the ITAE indicator and the average simulation time. For each algorithm, we conducted a Monte Carlo approach with 30 repetitions. The proposed MSOS outperformed the

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

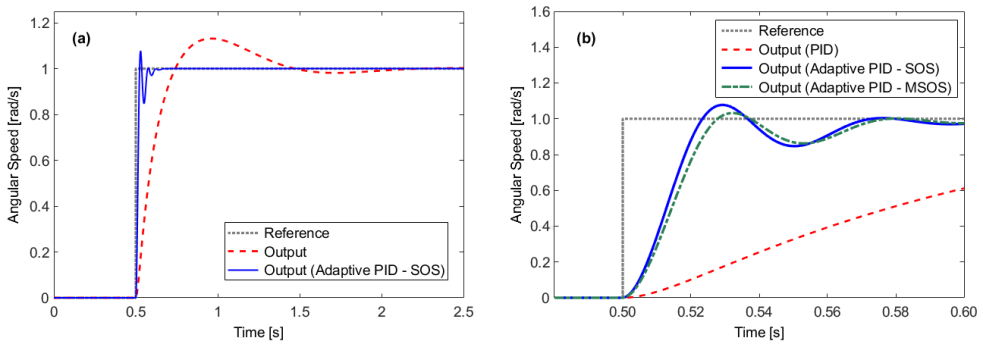


Figure 5: Step response comparisons. (a) standard PID controller and adaptive PID using SOS. (b) adaptive PID controllers using SOS and MSOS

Table 3: Adaptive PID controller design for different algorithms

Parameter	SOS	MSOS	Parameter	SOS	MSOS
x_p	96.97	99.73	q_p	4.03	4.57
x_i	3.50	2.87	q_i	6.32	4.73
x_d	18.45	16.98	q_d	0.50	0.60
t_{sim}[min]	25.57	26.13	ITAE	$(1.736 \pm 0.062) \times 10^{-3}$	$(1.234 \pm 0.023) \times 10^{-3}$

original SOS in terms of the ITAE indicator. As expected, MSOS presented a slightly higher average simulation time (2.19%) due to the allocation step.

As mentioned earlier, the idea behind the proposed MSOS is to select organisms for each phase such that the probability of generating better solutions is increased. Figure 6 shows the number of solution improvements generated during the phases of the algorithms. It can be seen that, when compared with SOS, MSOS presented a higher number of solution improvements. The proposed MSOS presented a higher number of solution improvements in the commensalism and in the parasitism phases, while the original SOS presented a higher number of solution improvements in the mutualism phase. The proposed MSOS presented a better distribution of the solution improvements among the different phases. According to Migliorini et al. [2015], the mutualism and the commensalism phases of SOS are related to the intensification capability, while the parasitism phase is related to the diversification capability. As a result, it can be concluded that SOS is unbalanced. The better distribution in the solution improvements obtained with MSOS contributes to improve the balance between the diversification and the intensification capabilities of SOS.

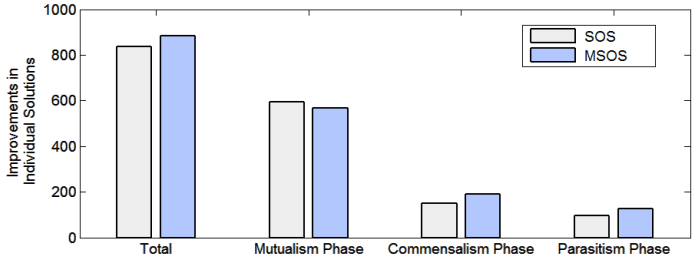


Figure 6: Number of new solutions generated in each phase of the algorithms

5. Conclusions

A modified version of the SOS algorithm was proposed. In the original SOS, organisms are randomly selected for a specific symbiotic relationship defined *a priori*. In the proposed MSOS, three organisms are randomly selected and an allocation step is performed to define which organism will be assigned to each phase of the algorithm. The allocation step consists in solving an optimization subproblem.

A case study was presented to illustrate the application of the proposed algorithm to solve an adaptive PID controller design optimization problem. The results show that the use of SOS to design an adaptive PID controller considerably increased the quality of the step response of a DC motor with speed control. Also, the proposed MSOS outperformed the original SOS in terms of the ITAE indicator.

A higher number of solution improvements was observed in the commensalism and parasitism phases of the proposed MSOS, when compared with the original SOS. Although MSOS presented a reduction in the number of solution improvements during the mutualism phase, the total number of solution improvements in MSOS was higher than the number observed in SOS. Also, the MSOS presented a better distribution of the number of improvements obtained in each phase. These results contribute to a better balance between the diversification and the intensification capabilities of the algorithm.

Future research may extend the scope of this paper by investigating different forms to compute weights $\alpha_m k$, $\alpha_c k$, and $\alpha_p k$. Another opportunity for future research is to investigate the application of the proposed MSOS algorithm in other classes of optimization problems such as multi-objective and combinatorial problems.

Acknowledgments

The authors would like to thank the Brazilian National Council for Scientific and Technological Development - CNPq (grant 423023/2018-7).

References

Abdullahi, M., Ngadi, M. A., and Abdulhamid, S. M. (2016). Symbiotic organism search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems*, 56:640–650.

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

Choe, S., Li, B., Ri, I., Paek, C., Rim, J., and Yun, S. (2018). Improved hybrid symbiotic organism search task-scheduling algorithm for cloud computing. *KSII Transactions on Internet and Information Systems*, 12(8):3516–3541.

Das, S., Pan, I., Halder, K., Das, S., and Gupta, A. (2013). LQR based improved discrete PID controller design via optimum selection of weighting matrices using fractional order integral performance index. *Applied Mathematical Modelling*, 37(6):4253–4268.

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy.

Duman, S. (2017). Symbiotic organisms search algorithm for optimal power flow problem based on valve-point effect and prohibited zones. *Neural Computing and Applications*, 28(11):3571–3585.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

Emhemed, A. A. A. and Mamat, R. B. (2012). Modelling and simulation for industrial DC motor using intelligent control. *Procedia Engineering*, 41:420–425.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA. ISBN 0262082136.

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

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Kuo, T. C. and Huang, Y. J. (2005). A sliding mode PID-controller design for robot manipulators. In *2005 International Symposium on Computational Intelligence in Robotics and Automation*, p. 625–629, Espoo.

Migliorini, W., André, L., and Parpinelli, R. S. (2015). Análise das rotinas de intensificação e diversificação no algoritmo inspirado em organismos simbióticos. In *Anais do 12º Congresso Brasileiro de Inteligência Computacional*, p. 1–6, Curitiba, Brazil (**in Portuguese**). Abricom.

Nama, S., Saha, A. K., and Ghosh, S. (2016). Improved symbiotic organisms search algorithm for solving unconstrained function optimization. *Decision Science Letters*, 5(3):361–380.

Puchta, E. D. P., Siqueira, H. V., and d. S. Kaster, M. (2020). Optimization tools based on meta-heuristics for performance enhancement in a gaussian adaptive PID controller. *IEEE Transactions on Cybernetics*, 50(3):1185–1194.

Roeva, O. and Slavov, T. (2012). PID controller tuning based on metaheuristic algorithms for bioprocess control. *Biotechnology & Biotechnological Equipment*, 26(5):3267–3277.

Sulaiman, M., Ahmad, A., Khan, A., and Muhammad, S. (2018). Hybridized symbiotic organism search algorithm for the optimal operation of directional overcurrent relays. *Complexity*, 2018: 4605769.

Talatahari, S. (2016). Symbiotic organisms search for design of frame and grillage systems. *Asian Journal of Civil Engineering*, 17(3):299–313.

Tejani, G. G., Savsani, V. J., and Patel, V. K. (2016). Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. *Journal of Computational Design and Engineering*, 3(3):226–249.

Tran, D.-H., Cheng, M.-Y., and Prayogo, D. (2016). A novel multiple objective symbiotic organisms search (MOSOS) for time-cost-labor utilization tradeoff problem. *Knowledge-Based Systems*, 94: 132–145.

Črepinšek, M., Liu, S.-H., and Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys*, 45(3):35:1–35:33.