

AJC

Academic Journal on Computing, Engineering and Applied Mathematics

EAM

2020

Volume 1 Issue 1

Academic Journal on Computing, Engineering and Applied Mathematics



ISSN: 2675-3588

UF
UNIVERSIDADE FEDERAL
DO TOCANTINS

Universidade Federal do Tocantins

Reitor

Prof. Dr. Luís Eduardo Bovolato

Vice-Reitor

Prof. Dr. Marcelo Leineker Costa

Pró-Reitoria de Graduação

Prof. Dr. Eduardo José Cezari

Pró-Reitoria de Pesquisa e Pós-Graduação

Prof. Dr. Raphael Sanzio Pimenta

Pró-Reitoria de Extensão e Cultura

Profa. Dra. Maria Santana Ferreira dos Santos

Pró-Reitoria de Administração e Finanças

Me. Carlos Alberto Moreira de Araújo Júnior

Pró-Reitoria de Assuntos Estudantis e Comunitários

Prof. Dr. Kherlley Caxias Batista Barbosa

Pró-Reitoria de Avaliação e Planejamento

Prof. Dr. Eduardo Andrea Lemus Erasmo

Pró-reitoria de Gestão e Desenvolvimento de Pessoas

Profa. Dra. Vânia Maria de Araújo Passos

Pró-Reitoria de Tecnologia da Informação e Comunicação

Prof. Dr. Ary Henrique Morais Oliveira

Direção do Campus de Palmas

Prof. Dr. Moisés de Souza Arantes Neto

Coordenação do Curso de Ciência da Computação

Prof. Dr. Alexandre Tadeu Rossini da Silva

Dados Internacionais de Catalogação na Publicação (CIP)

Academic Journal on Computing, Engineering and Applied Mathematics (AJCEAM) [recurso eletrônico] / Universidade Federal do Tocantins, Curso de Ciência da Computação. – vol. 01, n. 01 ([março/maio], 2020) – Palmas - TO, UFT, 2020. ISSN nº 2675-3588.

Quadrimestral no primeiro ano de publicação 2020

Semestral.

Disponível em:

<https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/index>

1. Ciência da Computação - periódico. 2. Matemática Aplicada. 3. Computação Aplicada. 4. Engenharias. 5. Ciências Exatas. I. Universidade Federal do Tocantins.

CDD 22.ed. 004

Ficha Catalográfica elaborada por Edson de Sousa Oliveira – CRB/2 – 1069.

Expediente

Editor-Chefe

Me. Tiago da Silva Almeida (UFT), Brasil

Editores

Dr. Edeilson Milhomem Silva (UFT), Brasil

Dr. Rafael Lima de Carvalho (UFT), Brasil

Me. Tiago da Silva Almeida (UFT), Brasil

Dr. Warley Gramacho da Silva (UFT), Brasil

Realização

Fundação Universidade Federal do Tocantins (UFT)

Quadra 109 Norte, Avenida NS-15, ALCNO-14 | Bloco III | sala 214 |Plano Diretor Norte | 77001-090 |
Palmas / TO | Brasil

Periodicidade

Este periódico possui periodicidade semestral e utiliza a Licença Creative Commons 4.0 - CC BY-NC 4.0. Contudo, a publicação dos artigos em modalidade avançada ou ahead of print, ou seja, tão logo os manuscritos aprovados sejam editados para publicação, é possível. O AJCEAM não possui taxas de publicação, tanto pouco de submissão de manuscritos, sendo totalmente gratuita para autores e leitores.

Indexadores

Google Acadêmico, desde 9 de maio de 2020

International Standard Serial Number – ISSN, desde 28 de maio de 2020

Crossref, desde 7 de junho de 2020

Revistas de Livre Acesso – LivRe, desde 24 de junho de 2020

Sumário

1 Editorial (Português): Academic Journal on Computing, Engineering and Applied Mathematics	vi
ALMEIDA	
2 Hybrid particle swarm optimization with spiral-shaped mechanism for solving high-dimension problems	1
DUARTE AND CARVALHO	
3 Technical report of Raspberry Pi prototype for lectures in public universities	7
ALMEIDA	
4 Steganography Genetic Algorithm Hyperparameter Tuning through Response Surface Methodology	13
DA SILVA, CARVALHO E MARTINS	
5 Inferring trust in web-based social networks: an analysis from TidalTrust and T-SWEETS algorithms	18
MILHOMEM SILVA ET AL.	

Editorial (Português): Academic Journal on Computing, Engineering and Applied Mathematics

Tiago da Silva Almeida¹

¹ *Universidade Federal do Tocantins, Palmas / TO, Brasil*

O AJCEAM surgiu de um sonho e uma necessidade. Do sonho de quatro professores de computação de criar um canal de qualidade de comunicação científica. E da necessidade de um canal focado em computação e áreas afim no estado do Tocantins. Alguns meses de muito trabalho e a AJCEAM toma forma e lançamos nossa primeira edição.

A primeira edição da AJCEAM representa o empenho do corpo editorial, caracterizado em quatro artigos. Apesar de endógenos, esses artigos representam pesquisas com temáticas recentes em computação.

O primeiro trabalho, intitulado “Hybrid particle swarm Optimization with spiral-shaped mechanism for solving high-dimension problems”, escrito por Duarte e Carvalho [1], representa uma avaliação importante na área de meta-heurísticas. O problema avaliado é uma formulação híbrida de um PSO (Particle Swarm Optimization) em vinte e quatro benchmarks. Evidentemente, meta-heurísticas são amplamente empregadas na resolução de problemas complexos em diversos campos de atuação. Trabalhos como esse são norteadores para outros pesquisadores.

O segundo trabalho, intitulado “Technical report of Raspberry Pi prototype for lectures in public universities”, escrito por Almeida [2], descreve, de maneira simples, como podem ser utilizados dispositivos programados para redução de custos, assim, como também na automação de instituições de ensino. É importante que diversos tipos de usuários tirem proveitos da cultura “Maker” e do barateamento dos dispositivos microcontrolados e microprocessados.

O terceiro trabalho, intitulado “Steganography Genetic Algorithm Hyperparameter Tuning through Response Surface Methodology”, escrito por da Silva, Carvalho e Martins [3], avalia quais parâmetros de um Algoritmo Genético podem influenciar na otimização de ocultação de mensagens em imagens (esteganografia). Com a análise proposta, os autores ainda sugerem a Metodologia da Superfície de Resposta, como forma de melhor ajuste de parâmetros. Todo algoritmo metaheurístico possui parâmetros de difícil ajuste para o problema em questão. E técnicas de esteganografia possuem amplas aplicações em diversas áreas distintas.

Por fim, o quarto trabalho, intitulado “Inferring trust in web-based social networks: an analysis from TidalTrust and T-SWEETS algorithms”, escrito por Silva et. al. [4], apresenta uma análise do algoritmo TSWEETS em relação ao algoritmo clássico TidalTrust. Esses algoritmos realizam a inferência de relacionamentos ocultos de confiança em ambientes virtuais sociais. Essa avaliação possui clara aplicação em marketing digital para as empresas e até no entendimento de relações sociais, e interações entre usuários em redes sociais.

Mesmo sendo poucos trabalhos em nossa primeira edição, são trabalhos sérios com grande potencial. A partir de agora, caberá aos leitores avaliá-los e se julgá-los coerente, citá-los. Esperamos fortemente, que esta primeira edição seja a primeira de muitas, e que nosso periódico possa contribuir para ciência da computação e áreas afins. Pois é nisso que nosso trabalho está focado. Boa leitura.

REFERÊNCIAS

- [1] H. M. Mendonça Duarte and R. Lima de Carvalho, “Hybrid particle swarm optimization with spiral-shaped mechanism for solving high-dimension problems,” *Academic Journal on Computing, Engineering and Applied Mathematics*, vol. 1, no. 1, p. 1–6, mar. 2020. [Online]. Available: <https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/article/view/8437>
- [2] T. d. S. Almeida, “Technical report of raspberry pi prototype for lectures in public universities,” *Academic Journal on Computing, Engineering and Applied Mathematics*, vol. 1, no. 1, p. 7–12, mar. 2020. [Online]. Available: <https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/article/view/8438>
- [3] W. Gramacho da Silva, R. Lima de Carvalho, and G. A. de Souza Martins, “Steganography genetic algorithm hyperparameter tuning through response surface methodology,” *Academic Journal on Computing, Engineering and Applied Mathematics*, vol. 1, no. 1, p. 13–17, mar. 2020. [Online]. Available: <https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/article/view/8529>
- [4] E. Milhomem, D. Oliveira Rodrigues, A. Carolina Salgado, and S. Romero Lemos Meira, “Inferring trust in web-based social networks: an analysis from tidaltrust and t-sweets algorithms,” *Academic Journal on Computing, Engineering and Applied Mathematics*, vol. 1, no. 1, p. 18–22, mar. 2020. [Online]. Available: <https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/article/view/8534>

Hybrid particle swarm optimization with spiral-shaped mechanism for solving high-dimension problems

Humberto M. M. Duarte¹ and Rafael Lima de Carvalho¹

¹ Universidade Federal do Tocantins, Curso de Ciência da Computação, Tocantins, Brazil

Reception date of the manuscript: 11/02/2020

Acceptance date of the manuscript: 28/02/2020

Publication date: 09/03/2020

Abstract— Particle swarm optimization (PSO) is a well-known metaheuristic, whose performance for solving global optimization problems has been thoroughly explored. It has been established that without proper manipulation of the inertia weight parameter, the search for a global optima may fail. In order to handle this problem, we investigate the experimental performance of a PSO-based metaheuristic known as HPSO-SSM, which uses a logistic map sequence to control the inertia weight to enhance the diversity in the search process, a spiral-shaped mechanism as a local search operator, as well as two dynamic correction factors to the position formula. Thus, we present an application of this variant for solving high-dimensional optimization problems, and evaluate its effectiveness against 24 benchmark functions. A comparison between both methods showed that the proposed variant can escape from local optima, and demonstrates faster convergence for almost every evaluated function.

Keywords—Particle swarm optimization, High-dimensional global optimization, Optimization

I. INTRODUCTION

Global optimization is the process of selecting a specific answer, out of a space of possibilities, that better solves a given problem. For minimization problems, this is often represented in the following way [1]:

Given: $f : A \rightarrow \mathbb{R}$

Find: $X^* \in A$ such that $f(X^*) \leq f(X)$ for all $X \in A$

Where f is defined as the objective function, or fitness function, $A \subseteq \mathbb{R}^d$ denotes a decision space of d dimensions, and each vector $X^* = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$ represents a candidate solution. A solution that minimizes, or maximizes, the objective function is called an optimal solution.

For a very broad range of applications, there are many theoretical and real-world problems that can be modeled in this framework [2]. Therefore, algorithms that can efficiently find optimal solutions for these problems become essential for the science and engineering fields. This is specially true for problems involving a high number of dimensions, which exponentially expands the size of the search space, and increases the computational cost for most optimization methods.

This paper presents a modification of the hybrid particle swarm optimization with a spiral-shaped mechanism (HPSO-

SSM), as presented in [3], for solving high-dimensional global optimization problems. The effectiveness of the proposed algorithm is evaluated against a set of well-known benchmark test functions [4], with up to 5000 dimensions.

II. HPSO-SSM METAHEURISTIC

The particle swarm algorithm (PSO), proposed by [5], is a metaheuristic based in the social behaviors of organisms, similar to a bird flock or school of fish. A population of particles is created, each representing a candidate solution to the chosen problem. Each of those particles has a position and velocity, whose values are iteratively adjusted based on their own best known previous position, as well as the current best position of the swarm. This way, the search-space is explored until a stop criteria is met.

Hybrid particle swarm optimization with spiral-shaped mechanism (HPSO-SSM) [3] is a modification of the traditional PSO, originally designed for feature selection in machine learning tasks. It utilizes the logistic map sequence to improve diversity, and a logarithmic spiral as a local search operator.

In the following, we describe the general procedure for this algorithm, as roughly described in Fig. 1, along with the changes made in order to adapt it for global optimization problems.

Firstly, the population of particles is created, with n individuals. During the whole procedure, the population size is constant. For a given iteration t , the current position and

Contact data: Humberto M. M. Duarte, 109 Norte, Av. NS15, Câmpus Universitário de Palmas, hbertoduarte@gmail.com

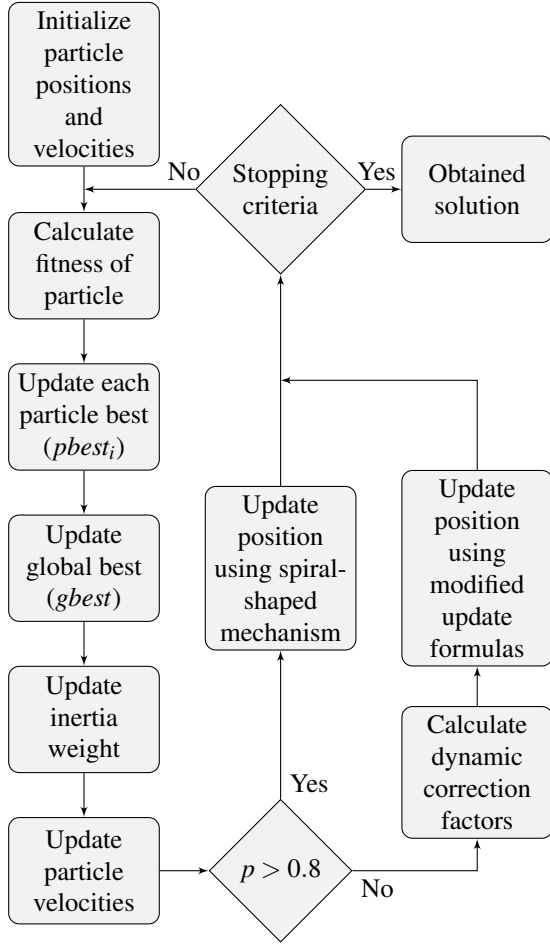


Fig. 1: HPSO-SSM algorithm.

velocity of each particle i are given by, respectively:

$$\begin{aligned} x_i^t &= (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t) \\ v_i^t &= (v_{i1}^t, v_{i2}^t, \dots, v_{id}^t) \end{aligned}$$

The position and velocity vectors are initialized with random values, uniformly distributed within the range of the search-space.

At every iteration, the fitness of each particle is calculated, using the position vector x_i directly as the argument of the objective function f . According the fitness value, the current best known position for each particle, and the best know position for the entire swarm, are updated, here denoted as:

$$\begin{aligned} \text{Fitness}_{\min} &= f(x_i) \\ pbest_i &= (pbest_{i1}, pbest_{i2}, \dots, pbest_{id}) \\ gbest &= (gbest_1, gbest_2, \dots, gbest_d) \end{aligned}$$

In order to guide the convergence of the result towards the global optima, the HPSO-SSM algorithm utilizes the logistic map to update the inertia weight w , as follows:

$$w^t = \mu \times w^{t-1} \times (1 - w^{t-1})$$

where μ is a positive constant ($\mu = 4$). The characteristics of randomness and ergodicity of the sequence enhance the ability of the algorithm to escape locally optimal solutions.

Next, the flying velocities of each particle are updated, utilizing the values of $pbest$ and $gbest$ to steer the swarm to-

wards the solution, according to the following formula:

$$\begin{aligned} v_{id}^{t+1} &= w^t + v_{id}^t \\ &+ c_1 \times r_{1i} \times (pbest_{id} - x_{id}^t) \\ &+ c_2 \times r_{2i} \times (gbest_d - x_{id}^t) \end{aligned}$$

where c_1 and c_2 are position constants, usually set as $c_1 = c_2 = 2$, and the variables r_{1i} and r_{2i} are random values, uniformly distributed, between 0 and 1.

In the HPSO-SSM method, there are two different ways in which the particle positions are updated. The first makes use of the spiral-shaped mechanism, and the second utilizes a modification of the standard PSO position update formula. The method to be used is determined by the variable p , a random value between 0 and 1, chosen at every operation. Should $p > 0.8$, the spiral-shaped mechanism is used, otherwise, the modified update formula is used.

Per the spiral-shaped mechanism, the next position of a particle is calculated with the following:

$$x_{id}^{t+1} = |gbest_d - x_{id}^t| \times \exp(b \times l) \times \cos(2\pi l) + gbest_d$$

where b is a positive constant ($b = 2$), and l is a random value, between -1 and 1. Otherwise, the position is updated using the modified formula below:

$$x_{id}^{t+1} = \mathfrak{R}_1 \times x_{id}^t + \mathfrak{R}_2 \times v_{id}^{t+1}$$

where \mathfrak{R}_1 and \mathfrak{R}_2 are dynamic correction factors, introduced in the HPSO-SSM algorithm, in order to control the trade-off between exploration and exploitation within the search.

Since that, in the HPSO-SSM algorithm, the position values are limited to the range between 0 and 1, the formulas for the dynamic correction factors were slightly modified, in order to preserve their functionality in the domain of the chosen objective function. Therefore:

$$\begin{aligned} \mathfrak{R}_1^t &= \frac{1}{(1 + \exp(a \times (-\frac{\min(SP)}{\max(SP)})))^t} \\ \mathfrak{R}_2^t &= 1 - \mathfrak{R}_1^t \\ SP_i^t &= \frac{x_i^{t-1}}{M} \times \left[\frac{x_i^{t-1}}{M} \right]^T \\ M &= \max(|A_{\min}|, |A_{\max}|) \end{aligned}$$

where a is a constant ($a = 2$), SP is the modified position magnitude formula, and M is the normalization factor, in which A_{\min} and A_{\max} denote the boundaries of the search-space.

The search loop is executed either until a set fitness value is reached, or for a fixed number of iterations. At the end, the particle with the best known position yields the obtained solution. However, the HPSO-SSM algorithm, as with the traditional PSO, does not guarantee that an optimal solution will be found.

III. EXPERIMENTAL RESULTS

In order to evaluate the effectiveness of the HPSO-SSM algorithm over global optimization problems with high dimensions, the following experimentation was conducted. A set

TABLE 1: TEXT FUNCTIONS UTILIZED IN THE EXPERIMENTATION.

Function Name	Function Formula	Search Range
Sphere	$f_1(x) = \sum_{i=1}^d x_i^2$	[-100,100]
Sum-Square	$f_2(x) = \sum_{i=1}^d ix_i^2$	[-10,10]
Schwefel's 2.22	$f_3(x) = \sum_{i=1}^d x_i + \prod_{i=1}^d x_i $	[-10,10]
Rotated hyper-ellipsoid	$f_4(x) = \sum_{i=1}^d (\sum_{j=1}^d x_j)^2$	[-100,100]
Schwefel's 2.21	$f_5(x) = \max\{ x_i , 1 \leq x_i \leq d\}$	[-100,100]
Rosenbrock	$f_6(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]$	[-30,30]
Step	$f_7(x) = \sum_{i=1}^d ([x_i + 0.5])^2$	[-100,100]
Quartic	$f_8(x) = \sum_{i=1}^d ix_i^4$	[-1.28,1.28]
Noise	$f_9(x) = \sum_{i=1}^d ix_i^4 + \text{random}[0, 1)$	[-1.28,1.28]
Sum-Power	$f_{10}(x) = \sum_{i=1}^d x_i ^{i+1}$	[-1,1]
Rastrigin	$f_{11}(x) = \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12,5.12]
Ackley	$f_{12}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}) - \exp(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)) + 20 + e$	[-32,32]
Griewank	$f_{13}(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]
Levy	$f_{14}(x) = \sum_{i=1}^{d-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + x_d - 1 [1 + \sin^2(3\pi x_n)]$	[-10,10]
Alpine	$f_{15}(x) = \sum_{i=1}^d x_i \sin(x_i) + 0.1x_i $	[-10,10]
Inverted Cosine mixture	$f_{16}(x) = 0.1d - (0.1 \sum_{i=1}^d \cos(5\pi x_i) - \sum_{i=1}^d x_i^2)$	[-1,1]
Zakharov	$f_{17}(x) = \sum_{i=1}^d x_i^2 + (\sum_{i=1}^d 0.5ix_i)^2 + (\sum_{i=1}^d 0.5ix_i)^4$	[-5,10]
Pathological	$f_{18}(x) = \sum_{i=2}^d 0.5 + \frac{\sin^2(\sqrt{100x_{i-1}^2 + x_i^2}) - 0.5}{1 + 0.001(x_{i-1}^2 - 2x_{i-1}x_i + x_i^2)^2}$	[-100,100]
Levy and Montalvo	$f_{19}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{d-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) + (x_d - 1)^2 (1 + \sin^2(2\pi x_d)))$	[-5,5]
Elliptic	$f_{20}(x) = \sum_{i=1}^d (10^6)^{\frac{i-1}{d-1}} x_i^2$	[-100,100]
Easom	$f_{21}(x) = (-1)^{d+1} \prod_{i=1}^d \cos(x_i) \cdot \exp(-\sum_{i=1}^d (x_i - \pi)^2)$	[-100,100]
Salomon	$f_{22}(x) = 1 - \cos(2\pi \sqrt{\sum_{i=1}^d x_i^2}) + 0.1 \sqrt{\sum_{i=1}^d x_i^2}$	[-100,100]
Schaffer	$f_{23}(x) = 0.5 + \frac{\sin^2(\sqrt{\sum_{i=1}^d x_i^2}) - 0.5}{(1 + 0.001(\sum_{i=1}^d x_i^2))^2}$	[-100,100]
Stretched V-sine	$f_{24}(x) = \sum_{i=1}^{d-1} (x_i^2 + 2x_{i+1}^2)^{0.25} \cdot ((\sin 50(x_i^2 + x_{i+1}^2)^{0.1})^2 + 1)$	[-10,10]

of 24 well-known benchmark functions was assembled, from existing literature involving optimization algorithms [4], that are delineated in Table 1. For all functions, the theoretical optima is $f_{min} = 0$.

For all tests, each function was executed 30 times, for dimension sizes of $d = \{30, 100, 500, 1000, 5000\}$. The population size n was fixed to 30, and the maximum number of iterations t_{max} was 500, for all simulations, resulting in a maximum of 15000 fitness function evaluations per test. In addition, the same control parameters were used for both PSO and HPSO-SSM algorithms, as follows: the position constants c_1 and c_2 were both set as 2, and the initial value of the inertia weight w was set as 0.5. The HPSO-SSM constants μ , a , and b were set to 4, 2, and 2, respectively.

Both algorithms, as well as the benchmark functions, were coded in Python 3.6, and all experiments were performed on the cloud-based environment Google Colaboratory [6].

a. Comparison with traditional PSO

First, a comparison is made between the traditional PSO and the proposed HPSO-SSM method, for a low number of dimensions ($d = 30$). The mean of the fitness values reached for each function, and their standard deviations, are arranged in Table 2. It can be seen in Table 2 that for almost every benchmark function, the HPSO-SSM method reaches the global optima, with exception of six functions (i.e., f_6 , f_9 , f_{14} , f_{19} , f_{22} , and f_{23}).

The PSO method, however, demonstrated results orders of magnitude worse, only reaching global optima on two

TABLE 2: COMPARISON RESULTS BETWEEN PSO AND HPSO-SSM, WITH 30 DIMENSIONS.

F	PSO (Mean \pm Std Dev)	HPSO-SSM (Mean \pm Std Dev)
f_1	$1.67 \times 10^3 \pm 3.73 \times 10^3$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_2	$4.47 \times 10^2 \pm 4.45 \times 10^2$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_3	$3.04 \times 10^1 \pm 1.49 \times 10^1$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_4	$2.04 \times 10^4 \pm 1.01 \times 10^4$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_5	$2.92 \times 10^1 \pm 8.04 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_6	$1.87 \times 10^4 \pm 3.57 \times 10^4$	$2.90 \times 10^1 \pm 1.53 \times 10^{-2}$
f_7	$1.67 \times 10^3 \pm 3.73 \times 10^3$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_8	$2.24 \times 10^0 \pm 5.14 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_9	$1.49 \times 10^1 \pm 5.26 \times 10^0$	$1.24 \times 10^1 \pm 9.40 \times 10^{-1}$
f_{10}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{11}	$2.67 \times 10^1 \pm 2.16 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{12}	$1.00 \times 10^0 \pm 1.52 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{13}	$1.21 \times 10^1 \pm 3.07 \times 10^1$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{14}	$3.66 \times 10^1 \pm 5.19 \times 10^1$	$5.77 \times 10^1 \pm 1.99 \times 10^0$
f_{15}	$3.02 \times 10^0 \pm 3.27 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{16}	$2.91 \times 10^{-1} \pm 2.14 \times 10^{-1}$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{17}	$3.43 \times 10^2 \pm 1.06 \times 10^2$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{18}	$2.99 \times 10^0 \pm 8.96 \times 10^{-1}$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{19}	$1.59 \times 10^0 \pm 1.51 \times 10^0$	$2.99 \times 10^0 \pm 2.03 \times 10^{-2}$
f_{20}	$9.69 \times 10^7 \pm 1.05 \times 10^8$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{21}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{22}	$1.48 \times 10^0 \pm 1.63 \times 10^0$	$9.30 \times 10^{-2} \pm 2.84 \times 10^{-1}$
f_{23}	$3.63 \times 10^{-1} \pm 7.88 \times 10^{-2}$	$2.09 \times 10^{-1} \pm 1.97 \times 10^{-1}$
f_{24}	$3.70 \times 10^1 \pm 9.96 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$

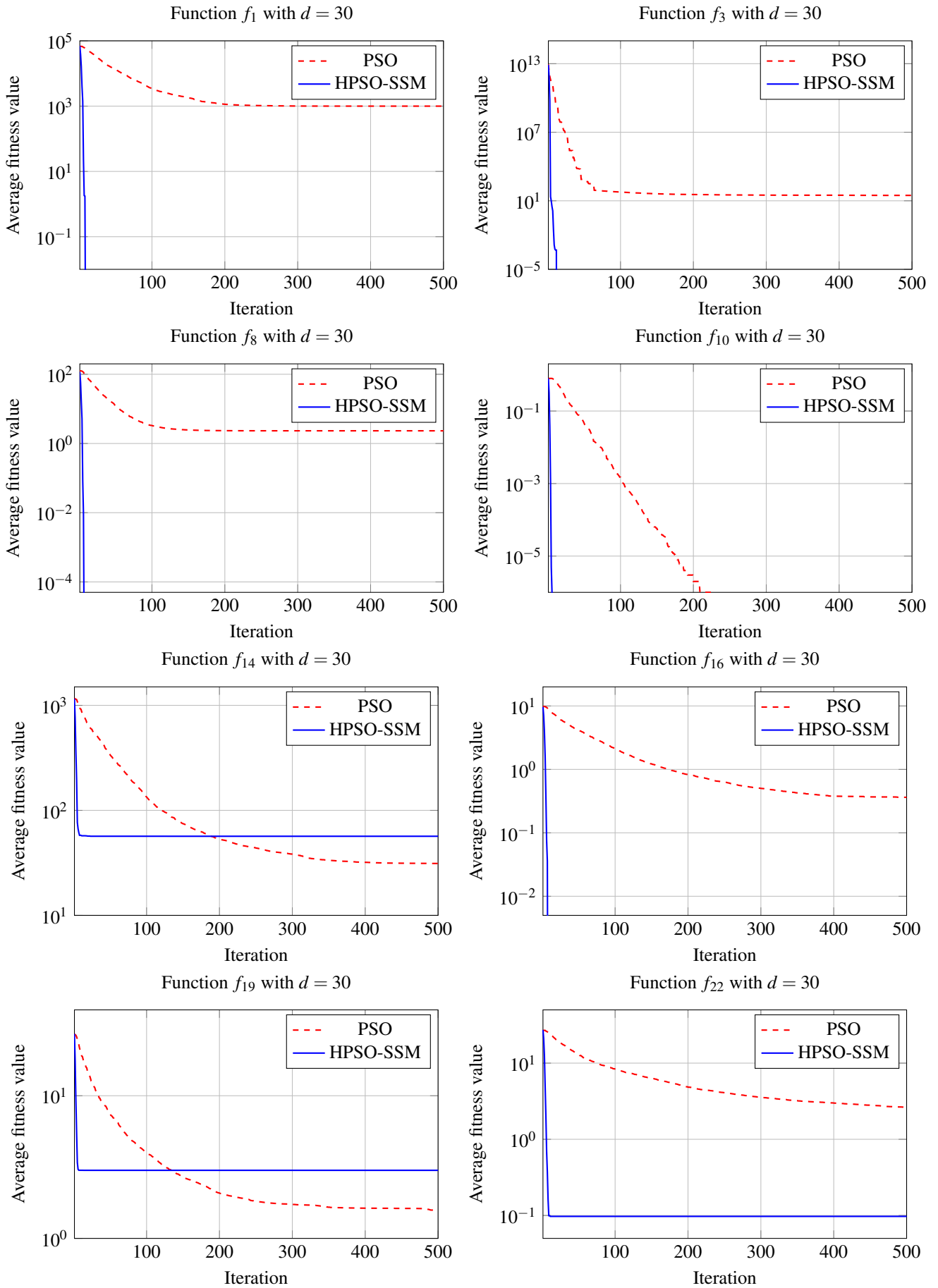


Fig. 2: Convergence curves of PSO and HPSO-SSM on eight benchmark functions, with 30 dimensions.

TABLE 3: RESULTS OBTAINED OVER 30 INDEPENDENT RUNS ON 100, 500, 1000, AND 5000 DIMENSIONS.

Function	Mean \pm Std Dev ($d = 100$)	Mean \pm Std Dev ($d = 500$)	Mean \pm Std Dev ($d = 1000$)	Mean \pm Std Dev ($d = 5000$)
f_1	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_2	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_3	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_4	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_5	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$6.65 \times 10^0 \pm 2.49 \times 10^1$	$1.33 \times 10^1 \pm 3.40 \times 10^1$
f_6	$9.90 \times 10^1 \pm 2.56 \times 10^{-2}$	$4.99 \times 10^2 \pm 2.99 \times 10^{-2}$	$9.99 \times 10^2 \pm 3.07 \times 10^{-2}$	$5.00 \times 10^3 \pm 1.50 \times 10^{-1}$
f_7	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_8	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_9	$4.60 \times 10^1 \pm 2.16 \times 10^0$	$2.40 \times 10^2 \pm 3.97 \times 10^0$	$4.90 \times 10^2 \pm 7.18 \times 10^0$	$2.47 \times 10^3 \pm 1.32 \times 10^1$
f_{10}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$3.44 \times 10^{-1} \pm 8.77 \times 10^{-1}$	$8.90 \times 10^{-1} \pm 1.78 \times 10^0$
f_{11}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{12}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{13}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{14}	$1.99 \times 10^2 \pm 3.53 \times 10^0$	$9.98 \times 10^2 \pm 1.91 \times 10^{-1}$	$1.99 \times 10^3 \pm 3.08 \times 10^1$	$1.04 \times 10^4 \pm 1.44 \times 10^3$
f_{15}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{16}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{17}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{18}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{19}	$9.99 \times 10^0 \pm 1.41 \times 10^{-2}$	$5.00 \times 10^1 \pm 2.80 \times 10^{-3}$	$1.00 \times 10^2 \pm 5.22 \times 10^{-3}$	$5.00 \times 10^2 \pm 8.51 \times 10^{-3}$
f_{20}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{21}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{22}	$7.74 \times 10^{-2} \pm 2.92 \times 10^{-1}$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{23}	$1.36 \times 10^{-1} \pm 2.10 \times 10^{-1}$	$1.86 \times 10^{-1} \pm 2.30 \times 10^{-1}$	$3.14 \times 10^{-1} \pm 2.24 \times 10^{-1}$	$3.30 \times 10^{-1} \pm 2.33 \times 10^{-1}$
f_{24}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$

occasions (i.e., f_{10} , and f_{21}). Nevertheless, it did display marginally better results than the HPSO-SSM method for only two functions, for which neither the PSO nor the HPSO-SSM were able to reach the theoretical optima (i.e., f_{14} and f_{19}).

This results can be further illustrated by the convergence curves of both algorithms, for eight selected benchmark functions, displayed in Fig. 2. It can be seen that, for the HPSO-SSM method, the convergence is much faster, and there is no apparent tendency towards local optima for most functions. For this benchmark, these partial results display the advantages of proper control over the inertia weight parameter, and that the modified position update strategies can yield a faster search.

b. Scalability over high dimensions

In order to verify the scalability of the HPSO-SSM method, we ran tests over a higher number of dimensions ($d = \{100, 500, 1000, 5000\}$). The mean and standard deviation of such experiment is shown in Table 3. It could be observed that, for most of the benchmark functions, the HPSO-SSM method was able to reach the global optima for any number of evaluated dimensions. Although for five of the benchmark functions, the method was not able to reach global optima (i.e., f_6 , f_9 , f_{14} , f_{19} , and f_{23}). Besides that, for two of the benchmark functions (i.e. f_5 and f_{10}), the algorithm was only able to reach the theoretical optima for dimension 100 and 500, with decreasing effectiveness on higher dimensions.

It should also be noted that, only in the case of function f_{22} , the method was only unable to reach the global optima for dimension 100, but performed better on the ensuing numbers of dimensions. To better illustrate the changes in convergence speed and robustness over increasing dimensions, the

convergence values of four selected benchmark functions, for the first 50 iterations, are displayed in Fig. 3.

We can observe that, in Function f_{16} , for which the algorithm was always able to reach global optima, the convergence speed is about the same for all dimensions. As for the functions, for which the algorithm was unable to ever reach global optima (i.e. f_{14} and f_{19}), the convergence speed is significant at first, however, once an apparent local optima is reached, the search is unable to escape from it, in the remaining iterations. On a function for which the algorithm was only able to reach the theoretical optima for lower dimensional sizes (i.e. f_{10}), the convergence becomes more sensitive to local optima as the dimension increases.

We also noted that, for most functions, the convergence behavior of the algorithm remains the same, given a set of control parameters, regardless of the number of dimensions. Nonetheless, some changes in convergence behavior was only observed for two functions (i.e. f_5 and f_{10}). This suggests that, once the control parameters have been fine-tuned for a specific function, the algorithm could be able to reach a similar convergence speed for any number of dimensions.

The observed changes in the convergence behavior over higher dimensional functions suggests different approaches for control parameter tuning. For example, one could perform partial executions of the algorithm, with gradual increases in dimension, in order to find the tuned parameters. A procedure like this could probably minimize the number of iterations in order to find optimized control parameters.

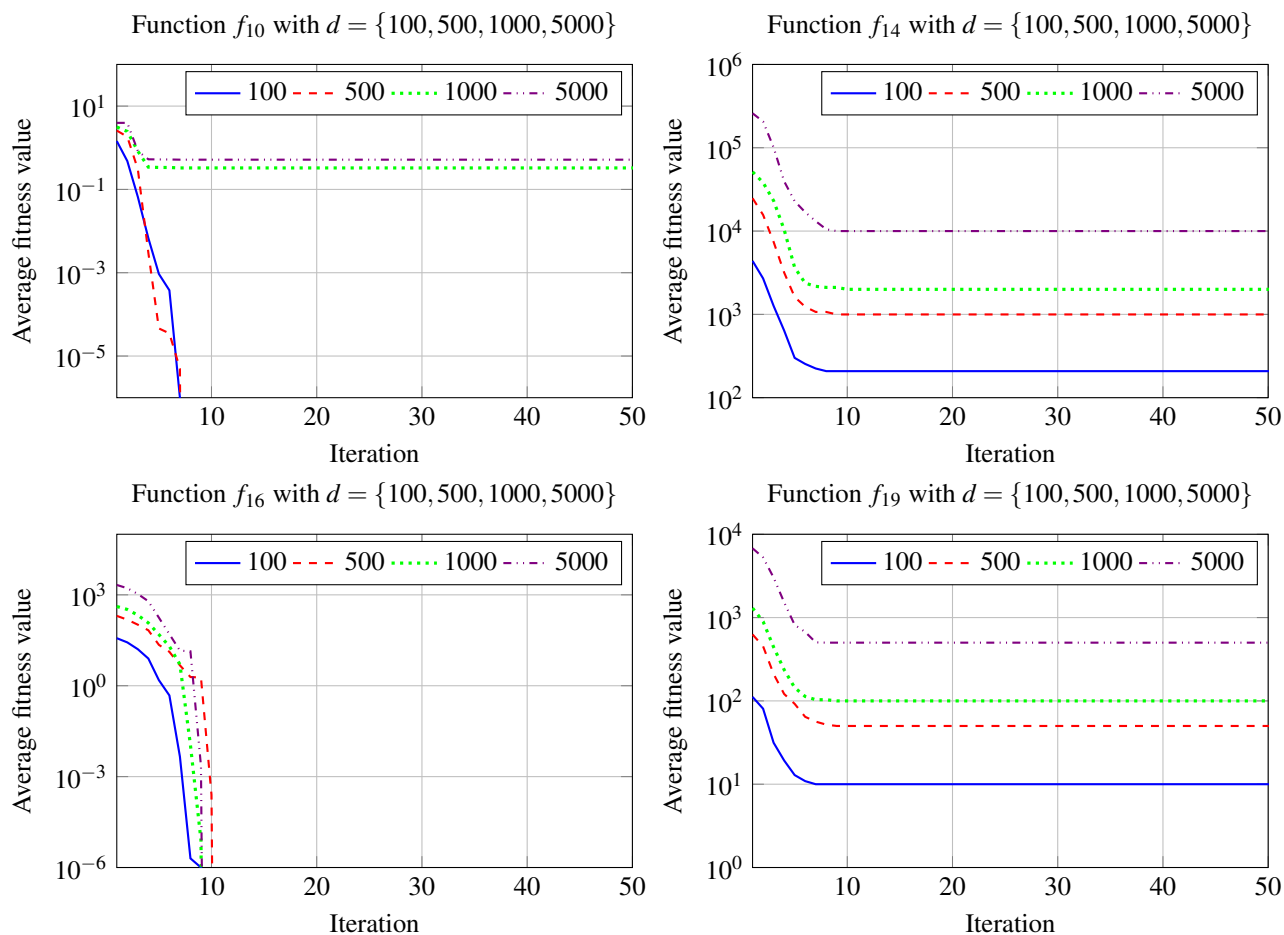


Fig. 3: Convergence curves of HPSO-SSM on four benchmark functions for the first 50 iterations, with 100, 500, 1000, and 5000 dimensions.

IV. FINAL CONSIDERATIONS

The hybrid particle swarm optimization with a spiral-shaped mechanism (HPSO-SSM), a variant of the particle swarm optimization (PSO), has been introduced, and its use on solving global optimization problems with a high number of dimensions was demonstrated. The efficiency of the considered method was tested against a set of 24 benchmark minimization functions, and the results were compared against those of the traditional PSO algorithm.

We have determined that the effectiveness of the proposed method surpassed that of the traditional algorithm by orders of magnitude, for most of the analyzed functions, both in the achieved fitness values and in the speed of convergence. The experiment results demonstrated that, for a considerable number of functions, the HPSO-SSM algorithm displays similar convergence behavior and speed, even after considerable increments in the number of dimensions. As future works, a meta-optimization process of control parameters could be investigated in order to get the best performance of HPSO-SSM.

REFERENCES

- [1] P. M. Pardalos and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications*, ser. Lecture Notes in Computer Science. Berlin, etc.: Springer-Verlag, 1987, vol. 268.
- [2] R. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on*

Evolutionary Computation (IEEE Cat. No.01TH8546), vol. 1, 2001, pp. 81–86.

- [3] K. Chen, F.-Y. Zhou, and X.-F. Yuan, "Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection," *Expert Systems with Applications*, vol. 128, pp. 140–156, 2019.
- [4] W. Long, T. Wu, X. Liang, and S. Xu, "Solving high-dimensional global optimization problems using an improved sine cosine algorithm," *Expert Systems with Applications*, vol. 123, pp. 108–126, 2019.
- [5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
- [6] Google. (2019) Colaboratory: Frequently asked questions. Retrieved from <https://research.google.com/colaboratory/faq.html> (12/02/2019).

Technical report of Raspberry Pi prototype for lectures in public universities

Tiago da Silva Almeida¹

¹ *Universidade Federal do Tocantins, Computer Science, Tocantins, Brazil*

Reception date of the manuscript: 11/02/2020

Acceptance date of the manuscript: 27/02/2020

Publication date: 09/03/2020

Abstract— This paper presents a low-cost prototype for lectures in public universities based on Raspberry Pi. It is described how to connect and configure the prototype using an infrared remote control. Technologies applied in education are widely explored in recent researches and can be held to lower the cost. The proposed prototype is 86% cheaper on average (compared to ordinary computer) and can be used for automation of the classrooms besides lectures. For example, the prototype can be additionally used to access control, environment monitoring, and management of the environment utilization by the users.

Keywords—Raspberry Pi, Prototype, Lectures

I. INTRODUCTION

With the advent of the Internet of Things (IoT), System-on-Chip (SoC) devices are widely used in many applications for different areas. SoCs can be used from military, medical, commerce, logistics, entertainment and educational applications.

In many educational institutions in Brazil, the allocation of resources for the development of activities is a bottleneck. This directly impacts the quality of the applied activities. Thus, low-cost resources are important to the maintenance of the activities.

For the simple activity of lectures, the most used equipment is a multimedia projector and a computer. At public universities, costs on average R\$ (Reais) 5,518.12¹ and computer costs on average R\$ 4,369.00². This is a high cost for public universities in periods of limited spending.

Many research papers have been done using Raspberry Pi due to its facility to work and implement solutions. For example, [1] describes a data processing system based on Raspberry Pi to inertial sensors. The paper is just an introduction and does not bring any results. [2] developed a touchscreen platform for rodent testing to allow cognitive testing based on

Raspberry Pi and Arduino. This has advantages compared to the standard maze apparatuses typically employed in rodent behavioral testing, according to the authors.

[3] proposed the extension of the AP (Access Point) configuration algorithm to deal with this dynamic nature (users often repeat joining and leaving the network) using Raspberry Pi for the AP. [4] proposed a renewable energy monitoring system for data acquisition and transmission applied to real-time cloud monitoring of a decentralized photovoltaic plant also based on Raspberry Pi embedded in each plant.

In the same field, [5] used a Raspberry Pi as a web server for home automation, with all transducers (sensors and actuators) connect by the internet to the Raspberry Pi.

On the other side, a lot of papers focus on “how to use new technologies or devices to improve education and solve its issues”. For example: [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] and [16].

In this context, this paper proposes a prototype for uses in lectures based on Raspberry Pi. The prototype was built based on the work of [17]. Section II is explained how the prototype was connected and configured. Section III concludes the paper and makes some discussion about the prototype.

II. METHODS AND MATERIALS

The materials and devices used for the prototype were:

- 1 -A Raspberry Pi 3 B with an 32 GB SD card (R\$ 360,00);
- 2 - Li-ion Battery 2.200 mAh (R\$ 50,00 each);
- 1 - TP4056 Battery Charger Module (R\$ 6,00);

Contact data: Tiago da Silva Almeida, Quadra 109 Norte, Avenida NS-15, ALCNO-14, Plano Diretor Norte, Palmas/TO, Brazil, tiagoalmeida@uft.edu.br

¹According to data from the Brazilian federal government portal, available for consultation at: <http://paineldeprecos.planejamento.gov.br/analise-materiais>.

²Also according to data from the Brazilian federal government portal, available for consultation at: <http://paineldeprecos.planejamento.gov.br/analise-materiais>.

- 1 - 1838B IR receiver (R\$ 7,00);
- 1 - Universal Remote for datashow (R\$ 130,00).

The IR (InfraRed) sensor has just three pins, which was connect with three pins on the GPIO (General-Purpose Input Output) connector in Raspberry Pi (VCC, GND, and SIGNAL). The connection is: a) VCC pin of IR receiver to 3.3V pin of Raspberry Pi; b) GND pin of IR receiver to any Ground pin of Raspberry Pi; c) SIGNAL pin to GPIO 18 of Raspberry Pi.

Each button of an IR remote control has a string of specific encoding. When a button is pressed, the IR transmitter in the remote control was send out the corresponding IR encoding signals. On the other side, when the IR receiver receives certain encoding signals, it will decode them to identify which button is pressed.

To this prototype, it was used the LIRC³ library to read infrared signals returned by buttons of the remote control and translate them to button values. Then, it was used pylirc (Python) to simplify the process for reading values from the remote control.

It's important to highlight that LIRC library runs without problems with Raspbian Jessie operational system until 2017. For newer versions of Raspbian is required other configuration of the libraries.

To configure the Raspberry Pi for remote control is required these steps:

1. Download the LIRC library using the following command from terminal:

```
1 sudo apt-get install lirc
```

2. Add the two lines below to `/etc/modules`. This will start the modules on boot of Raspbian. Pin 18 will be used to take the output from the IR sensor. Use the following command:

```
1 sudo nano /etc/modules
```

And add this two lines in the file opened:

```
1 lirc\_dev
2 lirc\_rpi gpio\_in\_pin=18
```

Also, edit the `/boot/config.txt` file. Open the file using the command:

```
1 sudo nano /boot/config.txt
```

And add the following line to the file:

³LIRC is a package that allows you to decode infrared signals of many (but not all) commonly used remote controls. LIRC runs as a daemon that will decode IR signals received by the device drivers and provide the information on a socket. We will then write a program in the user space to monitor this socket for input events using the LIRC client library.

```
1 dtoverlay=lirc-rpi,gpio\_in\_pin=18
```

3. Edit `/etc/lirc/hardware.conf` using the command:

```
1 sudo nano /etc/lirc/hardware.conf
```

And type the following code exactly as shown below.

```
1 # /etc/lirc/hardware.conf
2 #
3 # Arguments which will be used when launching
  lircd
4 LIRCD_ARGS="--uinput"
5 # Don't start lircmd even if there seems to be
  a good config file
6 # START_LIRCMD=false
7 # Don't start irexec, even if a good config
  file seems to exist.
8 # START_IEXEC=false
9 # Try to load appropriate kernel modules
10 LOAD_MODULES=true
11 # Run "lircd --driver=help" for a list of
  supported drivers.
12 DRIVER="default"
13 # usually /dev/lirc0 is the correct setting
  for systems using udev
14 DEVICE="/dev/lirc0"
15 MODULES="lirc_rpi"
16 # Default configuration files for your
  hardware if any
17 LIRCD_CONF=""
18 LIRCMD_CONF=""
```

Type the command to reboot.

4. To perform a quick test to check if LIRC is working, it is required to stop the LIRC daemon and start mode2. mode2 shows the pulse/space length of infrared signals.

To start the LIRC in mode2 type the following command:

```
1 sudo /etc/init.d/lirc stop
2 mode2 -d /dev/lirc0
```

5. The `irrecord` command helps to discover the IR codes used by the remote control and assist with creating a `lircd.conf` file which will be used by LIRC.

To record the IR code type the following command from the terminal:

```
1 irrecord -d /dev/lirc0 ~/lircd.conf
```

Once started, `irrecord` is going to show detailed instructions on how to configure the remote control. Each button should be a pre-defined name. Running `irrecord-list-namespace` is going to display a list of available names. It can be set any name for any button from the `irrecord-list-namespace` list. After successfully recording the configuration file, it looks like the following listing:

```

1 # Please make this file available to others
2 # by sending it to <lirc@bartelmus.de>
3 #
4 # this config file was automatically generated
5 # using lirc -0.9.0-pre1(default) on Mon Dec 16
6 #       20:26:03 2019
7 #
8 # contributed by
9 #
10 # brand: /home/pi/lircd.conf
11 # model no. of remote control:
12 # devices being controlled by this remote:
13 #
14 begin remote
15
16 name /home/pi/lircd.conf
17 bits 16
18 flags SPACE_ENC|CONST_LENGTH
19 eps 30
20 aeps 100
21
22 header 9093 4490
23 one 645 1625
24 zero 645 494
25 ptrail 647
26 pre_data_bits 16
27 pre_data 0x8C73
28 gap 108500
29 toggle_bit_mask 0x0
30
31 begin codes
32 KEY_LEFT 0x13EC
33 KEY_UP 0x639C
34 KEY_RIGHT 0x936C
35 KEY_DOWN 0xE31C
36 KEY_ENTER 0x53AC
37 KEY_TAB 0xE11E
38 KEY_ALTRELEASE 0x31CE
39 KEY_ALTPRESS 0xF10E
40 KEY_CLOSE 0x916E
41 KEY_PRESENTATION 0xD32C
42 KEY_ESC 0x33CC
43 KEY_POWER 0x817E
44 end codes
45
46 end remote
    
```

Now replace the existing configuration file (which is most likely empty) with the created new one, by the following command:

```
1 sudo cp lircd.conf /etc/lirc/lircd.conf
```

6. Restart LIRC by the following command:

```
1 sudo /etc/init.d/lirc restart
```

irw can be used to test the new configuration file. irw sends data from Unix domain socket to stdout.

7. At this stage was created a virtual keystroke from the remote control. It was used the Python (pylirc) code to do this. pylirc is LIRC Python wrapper and it is required to access LIRC from Python programs. To install pylirc should be typed following command in the terminal:

```
1 sudo apt-get install python-pylirc
```

Create a lircrc.conf file. The lircrc file is used to map the key symbols defined in lircd.conf to application-specific strings. Thus, this file cannot be configured until lircd has been configured to provide proper key symbols as displayed by irw. A lircd was already configured in the previous step.

The lircrc file should be placed in the home directory as ~/.config/lircrc. Optionally you can create a system-wide configuration file located in /etc/lirc/lircrc which will be used when no lircrc file can be found in the /home directory.

The syntax of the lircrc file consists of one or more of the following constructions:

```

1 begin
2 prog = ...
3 remote = ...
4 button = ...
5 [button = ...] (optional, for key sequences)
6 repeat = ...
7 delay = ...
8 ignore_first_events = ...
9 config = ...
10 [config = ...] (optional, for toggle button
11     behaviour)
12 mode = ...
13 flags = ...
14 end
    
```

The complete lircrc file that was prepared for the prototype is as follows:

```

1 begin
2 button = KEY_ESC
3 prog = irexec
4 repeat = 0
5 config = esc
6 end
7 begin
8 button = KEY_TAB
9 prog = irexec
10 repeat = 0
11 config = tab
12 end
13 begin
14 button = KEY_LEFT
15 prog = irexec
16 repeat = 0
17 config = left
18 end
19 begin
20 button = KEY_RIGHT
21 prog = irexec
22 repeat = 0
23 config = right
24 end
25 begin
26 button = KEY_DOWN
27 prog = irexec
28 repeat = 0
29 config = down
30 end
31
32 begin
33 button = KEY_UP
    
```



```

34 prog = irexec
35 repeat = 0
36 config = up
37 end
38 begin
39 button = KEY_ALTRELEASE
40 prog = irexec
41 repeat = 0
42 config = altrelease
43 end
44 begin
45 button = KEY_ALTPRESS
46 prog = irexec
47 repeat = 0
48 config = altpress
49 end
50 begin
51 button = KEY_CLOSE
52 prog = irexec
53 repeat = 0
54 config = close
55 end
56 begin
57 button = KEY_PRESENTATION
58 prog = irexec
59 repeat = 0
60 config = presentation
61 end
62 begin
63 button = KEY_POWER
64 prog = irexec
65 repeat = 0
66 config = power
67 end

```

8. PyUserInput is a cross-platform Python module to take control of the mouse and keyboard in Python. PyUserInput is registered on PyPI (Python Package Index) and updated periodically, so tools such as pip can be used to install. Type the following command from terminal:

```
1 pip install PyUserInput
```

Here is the complete python program to emulate keyboard from remote key-press:

```

1 #!/usr/bin/env python
2
3 import pylirc, time
4 from pykeyboard import PyKeyboard
5 from pymouse import PyMouse
6 import os
7
8 k = PyKeyboard()
9 m = PyMouse()
10
11 blocking = 0;
12
13 x = 0
14 y = 0
15
16 def setup():
17     pylirc.init("irexec")
18
19 def key_press(config):
20     if config == 'esc':
21         k.tap_key(k.escape_key)
22         print 'ESC key'
23     if config == 'tab':

```

```

24         k.tap_key(k.tab_key)
25         print 'TAB key'
26     if config == 'presentation':
27         k.press_key(k.control_key)
28         k.press_key(k.shift_key)
29         k.tap_key('p')
30         k.release_key(k.shift_key)
31         k.release_key(k.control_key)
32         print 'Presentation mode in Okular'
33     if config == 'power':
34         os.system('sudo shutdown now')
35         print 'Power Down'
36     if config == 'close':
37         k.press_key(k.control_key)
38         k.tap_key('q')
39         k.release_key(k.control_key)
40         print 'Closing Okular'
41     if config == 'altrelease':
42         k.release_key(k.alt_key)
43         print 'Releasing ALT'
44     if config == 'altpress':
45         k.press_key(k.alt_key)
46         print 'Pressing ALT'
47
48 def loop():
49     while True:
50         s = pylirc.nextcode(1)
51         while(s):
52             for (code) in s:
53                 print 'Command: ', code["config"]
54                 key_press(code["config"])
55                 if(not blocking):
56                     s = pylirc.nextcode(1)
57             else:
58                 s = []
59
60 def destroy():
61     pylirc.exit()
62
63 if __name__ == '__main__':
64     try:
65         setup()
66         loop()
67     except KeyboardInterrupt:
68         destroy()

```

9. Save the file as remote.py to the directory /home/pi in the Raspberry Pi. Make the file executable by the following command:

```
1 sudo chmod +x remote.py
```

10. Open the autostart file by the command:

```
1 sudo nano /.config/lxsession/LXDE-pi/autostart
```

Add the following two lines at the end of the file:

```

1 @pcmanfm /home/pi/media --wallpaper-mode=tile
2 @python /home/pi/remote.py

```

So far, the flash drive with the PDF presentation is going to load when the Raspbian startup. To do so, it is automatic open up the /home/pi/media directory to choice the flash drive. The presentation file is open with Okular PDF reader and it is pre-installed in Raspberry Pi. At the same time, it is also started remote.py file.

TABLE 1: POWER SPECIFICATION OF ALL RASPBERRY PI MODELS.

Product	Recommended PSU current capacity	Maximum total USB peripheral current draw	Typical bare-board active current consumption
Raspberry Pi Model A	700mA	500mA	200mA
Raspberry Pi Model B	1.2A	500mA	500mA
Raspberry Pi Model A+	700mA	500mA	180mA
Raspberry Pi Model B+	1.8A	600mA/1.2A (switchable)	330mA
Raspberry Pi 2 Model B	1.8A	600mA/1.2A (switchable)	350mA
Raspberry Pi 3 Model B	2.5A	1.2A	400mA
Raspberry Pi 3 Model A+	2.5A	Limited by PSU, board, and connector ratings only.	350mA
Raspberry Pi 3 Model B+	2.5A	1.2A	500mA
Raspberry Pi 4 Model B	3.0A 1.2A	600mA	
Raspberry Pi Zero W/WH	1.2A	Limited by PSU, board, and connector ratings only.	150mA
Raspberry Pi Zero	1.2A	Limited by PSU, board, and connector ratings only	100mA

To add the battery inside of a case of the Raspberry Pi, the default micro USB (Universal Serial Bus) port is not a good option. A good option is providing power using GPIO. For this, it was used the GPIO of Raspberry Pi of VCC (pin 4) and GND (pin 6). The power specification of all Raspberry Pi models are shown in Table 1⁴.

For tests were connected to two batteries of 3.7 V in series, adding voltage to 7.4 V and keep the current. How the batteries have 2,200 mAh, the prototype was able to supply the system for 2.5 hours. It is a low time supply for four hour-lecture (50 minutes). But, adding two more batteries of the same features is enough for almost 5 hours of running.

III. CONCLUSION

This paper is a technical report about the construction of a low-cost prototype to use in lectures by universities based on Raspberry Pi. Due to the simplicity, the prototype runs well and it is easy to implement.

The Raspberry Pi was chosen because of its capability of processing, besides the low cost. The prototype can be installed in classrooms and uses to control the environment, like temperature monitoring and access control of the classrooms. It can be used as a web server, or such as a node in a local network.

At the end, the total costs were R\$ 603.00 on average, against R\$ 4,369.00 of an ordinary computer in public universities, also on average. This represents 86.19% of cost reduction just for regular lectures, besides other possibilities of implementation using the same equipment.

REFERENCES

[1] A. Alvarellos, A. Vázquez, and J. Rabuñal, "Raspberry pim: Raspberry pi based inertial sensor data processing system," *Proceedings*, vol. 2, no. 18, 2018. [Online]. Available: <https://www.mdpi.com/2504-3900/2/18/1159>

[2] J. D. O'Leary, O. F. O'Leary, J. F. Cryan, and Y. M. Nolan, "A low-cost touchscreen operant chamber using a raspberry pi™," *Behavior Research Methods*, vol. 50, no. 6, pp. 2523–2530, Dec 2018. [Online]. Available: <https://doi.org/10.3758/s13428-018-1030-y>

[3] M. M. Islam, N. Funabiki, M. Kuribayashi, S. K. Debnath, K. I. Munene, K. S. Lwin, R. W. Sudibyo, and M. S. A. Mamun, "Dynamic

access-point configuration approach for elastic wireless local-area network system and its implementation using raspberry pi," *International Journal of Networking and Computing*, vol. 8, no. 2, pp. 254–281, 2018.

[4] R. I. Pereira, I. M. Dupont, P. C. Carvalho, and S. C. Jucá, "Iot embedded linux system based on raspberry pi applied to real-time cloud monitoring of a decentralized photovoltaic plant," *Measurement*, vol. 114, pp. 286–297, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S026322411730605X>

[5] V. Vujović and M. Maksimović, "Raspberry pi as a sensor web node for home automation," *Computers & Electrical Engineering*, vol. 44, pp. 153–171, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045790615000257>

[6] N. Miglani and P. Burch, "Educational technology in india: The field and teacher's sensemaking," *Contemporary Education Dialogue*, vol. 16, no. 1, pp. 26–53, 2019. [Online]. Available: <https://doi.org/10.1177/0973184918803184>

[7] G. Friedland, W. Hürst, and L. Knipping, "The future of multimedia education and educational multimedia," in *Proceedings of the International Workshop on Educational Multimedia and Multimedia Education*, ser. Emme '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 125–126. [Online]. Available: <https://doi.org/10.1145/1290144.1290167>

[8] S. Xu, T. Hu, and Q. Zhou, "A case study in jiuwuji minority middle school on rural teachers' educational technology training in terms of a switch of educational product," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, May 2011, pp. 318–320.

[9] M. Qian, B. Zhao, and Y. Gao, "Exploring the training path of design thinking of students in educational technology," in *2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI)*, Aug 2019, pp. 315–319.

[10] J. Hongyan and R. Jianfeng, "A brief analysis on the seminar of professional construction for modern preschool educational technology," in *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, April 2012, pp. 2262–2264.

[11] A. I. Taganov, A. N. Kolesenkov, D. A. Perepelkin, and D. S. Zhuravlev, "Management of educational risk on the basis of data mining in gis," in *2017 International Conference "Quality Management, Transport and Information Security, Information Technologies" (IT QM IS)*, Sep. 2017, pp. 577–580.

[12] C. Wei and L. Yuan, "Reflection on college informatized teaching model under the background of educational informatization," in *2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI)*, Aug 2019, pp. 81–83.

[13] G. R. Iksanova, "Improvement of university education and educational service consumer interconnection in russia," in *2013 International Conference on Interactive Collaborative Learning (ICL)*, Sep. 2013, pp. 666–668.

⁴<https://www.raspberrypi.org/documentation/faqs/>.

- [14] S. Tatiana Alexandrovna, "The role of distance educational technologies in the development of educational programmes in a network form: Experience and relevant problems," in *2018 IV International Conference on Information Technologies in Engineering Education (Inforino)*, Oct 2018, pp. 1–4.
- [15] H. M. Fardoun, A. S. Mashat, and L. Gonzalez, "New subject to improve the educational system: Through a communication channel between educational institution-company," in *2013 Federated Conference on Computer Science and Information Systems*, Sep. 2013, pp. 709–712.
- [16] T. A. Tabishev, M. V. Alikaeva, and A. L. Betuganova, "Electronic informational and educational environment and organization of the educational process of a modern university (on the materials of the kabardino-balkar state university)," in *2019 International Conference "Quality Management, Transport and Information Security, Information Technologies" (IT QM IS)*, Sep. 2019, pp. 569–572.
- [17] Taifur. (2011) Raspberry pi presentation machine with remote control. Tomado de <https://www.instructables.com/id/Raspberry-Pi-Presentation-Machine-With-Remote-Cont/>.

Steganography Genetic Algorithm Hyperparameter Tuning through Response Surface Methodology

Warley Gramacho da Silva¹, Rafael Lima de Carvalho¹ and Glêndara Aparecida de Souza Martins²

¹ Federal University of Tocantins - UFT, Computer Science Department, Palmas-TO, Brazil

² Federal University of Tocantins - UFT, Food Engineering Department, Palmas-TO, Brazil

Reception date of the manuscript: 27/02/2020

Acceptance date of the manuscript: 28/02/2020

Publication date: 09/03/2020

Abstract— Steganography consists of hiding bits of an information source into a host source. In image processing, a common way of doing the hiding process is to break each byte from the message information and embed into the message bytes in a way that the differences among the original host and the embedded one are minimized. A Genetic Algorithm (GA) can be used to find the proper combination of bits in order to minimize such differences, but some hyperparameters need to be optimized in order to get an optimized performance. This work investigates the application of Response Surface Methodology (RSM) to find the best hyperparameters of a genetic algorithm applied to image steganography. As a result, RSM was able to point out fine-tuning hyperparameters for the GA.

Keywords—RSM, Steganography, Genetic Algorithms, Hyperparameter Optimization

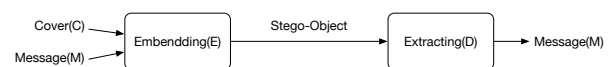


Fig. 1: Fluxograma de esteganografia

I. INTRODUÇÃO

A esteganografia é a arte de ocultar informações digitais, a fim de impedir a detecção de mensagens ocultas, ou seja, o objetivo da esteganografia é evitar a suspeita na transmissão de uma mensagem secreta. Inclui-se técnicas para ocultar uma imagem, um arquivo de texto, um arquivo de áudio e até mesmo um programa executável dentro de uma imagem de cobertura sem distorção visual da imagem [1, 2]. Como apresentado em [3], esteganografia pode ser formalmente definida como o processo de incorporação que descreve um mapeamento de $E : C \times M \rightarrow C$, em que C é o conjunto de coberturas possíveis e M é o conjunto de possíveis mensagens. O processo de extração consiste em um mapeamento de $D : C \rightarrow M$, extraíndo a mensagem secreta de uma imagem de cobertura.

Definição I.1 O quádruplo $\Psi = \langle C, M, D, E \rangle$, onde C é o conjunto de coberturas possíveis, M o conjunto de mensagens secretas com $|C| \geq |M|$, $E : C \times M \rightarrow C$ a função de mapeamento e $D : C \rightarrow M$, a função de extração, com a propriedade que $D(E(c, m)) = m$ para todos os $m \in M$ e $c \in C$ [3].

O processo de mapeamento é definido de forma que a imagem de cobertura e o objeto “stego” correspondente sejam perceptivamente semelhantes. O esquema de esteganografia pode ser visto na Figura 1.

De acordo com [2], existem seis categorias de esteganografia, a saber: técnicas de sistema de substituição, técnicas de transformação de domínio, técnicas de espectro de espalhamento, técnicas de método estatístico, técnicas de distorção e técnicas de geração de cobertura. Na técnica do sistema de substituição, os *bits* redundantes ou desnecessários são substituídos de uma imagem de cobertura pelos *bits* da mensagem secreta usando, por exemplo, o método *Bit Menos Significativo* (LSB, do inglês, *Least Significant Bit*) para codificar a mensagem secreta. No entanto, a esteganografia do LSB é suscetível as técnicas de análise de esteganização [4]. Para evitar a instalação de detecção de esteganografia através do uso do método LSB, em [5], foi proposta uma nova estratégia de substituição dos *bits*. Com o objetivo de otimizar o desempenho desse esquema proposto, foi utilizado um Algoritmo Genético (GA).

Os GAs possuem alguns parâmetros de entrada que são, geralmente, definidos manualmente após várias tentativas de ajustes. Uma técnica usada para definir um ajuste para os parâmetros dos AGs é a técnica da Metodologia da Superfície de Resposta (RSM). O RSM é uma combinação de técnicas matemáticas e estatísticas adequadas para modelagem e análise de problemas nos quais a variável resposta é afetada por várias variáveis de entrada, que visam otimizar as respostas [6].

	Decimal representation	Binary representation
Original Pixel	6 1	0 0 1 1 1 1 0 1
Pixel after substitution	4 9	0 0 1 1 0 0 0 1
Optimized Pixel	6 4	0 1 0 0 0 0 0 0

Binary chunk of the message to be hidden

Fig. 2: Modelo de otimização adotado. Na imagem, o pixel original (valor 61 em decimal) recebe o par de pixels (00). Após a substituição, o valor do pixel otimizado se torna 64.

Neste artigo, propõe-se o uso da MRS para melhor ajuste dos parâmetros do AG para otimizar a esteganografia de imagens usando uma estratégia de substituição dos *bits* conforme proposto por [5].

II. ALGORITMO GENÉTICO PARA O PROBLEMA DE ESTEGANOGRAFIA

Algoritmos Genético é uma metahaurística de otimização bio-inspirada, proposta por John Holland nos anos 1960 [7]. O GA é um método de busca e otimização que simula os processos naturais da evolução que consistem basicamente nos processos de cruzamento, mutação, mapeamento de fenótipo e cálculo de aptidão (*fitness*) [8].

A idéia básica do GA é mostrada no Algoritmo 1 e funciona da seguinte maneira. Uma população inicial é gerada. Depois disso, o *loop* do algoritmo genético começa com a etapa de cruzamento, onde dois indivíduos são selecionados para serem os pais de um novo indivíduo, essa etapa é repetida até que não haja mais pais. Logo após a mutação ocorrer, é necessária uma baixa taxa de mutação nesta etapa para não comprometer a geração. E, finalmente, o processo de seleção natural. Para concluir o *loop*, um critério de parada é avaliado (número máximo de iteração).

Algoritmo 1 Basic Genetic Algorithm [8]

- 1: inicializa a população
 - 2: **repeat**
 - 3: **repeat**
 - 4: Aplica crossover
 - 5: Aplica mutação
 - 6: Calculo da função *fitness*
 - 7: **until** população completa
 - 8: **until** número de geração atingido
-

O princípio de otimização usado neste trabalho segue o proposto em [5]. Nesta abordagem, cada *byte* de destino deve passar por uma fase de verificação. Para ilustrar como isso funciona, considere um *byte*, como ilustrado na Figura 2. Na figura, o primeiro *byte* (61) é o valor original do pixel encontrado na imagem de destino. Neste exemplo, a mensagem será colocada na terceira e quarta posição do pixel de destino. Ao fazer a substituição, o valor do pixel se torna 49, o que fornece uma diferença de luminância de 12. Depois de otimizar o valor do pixel, pode ser visto que o valor 64 está mais próximo do valor original, enquanto o bloco de mensagens a ser oculto permanece incorporado.



Fig. 3: Imagem-alvo utilizada nos experimentos.

A representação binária do pixel favorece uma escolha natural para abordagem de otimização tais como Algoritmos Genéticos. Neste artigo, a função de aptidão (*fitness*) é uma função quadrática que contabiliza a diferença entre os *pixels* originais e os otimizados. Esta função pode ser conforme.

$$\sum_i^N (pixel_i - optimizedPixel_i)^2 \quad (1)$$

A solução ótima para o problema é a mais próxima do conjunto original de *pixels*, a população do Algoritmo Genético é gerada tendo pontos aleatórios ao redor da imagem original. Depois disso, a população inicial é atualizada para manter os *bits* da mensagem oculta. Em sequência, o *fitness* de cada indivíduo é calculado para prosseguir com o restante do algoritmo AG.

III. EXPERIMENTOS

Para execução do AG, a Figura 3 foi utilizada para realização dos testes juntamente com uma mensagem de texto aleatória a ser oculta na imagem de destino.

Para ilustrar a abordagem de ajuste dos parâmetros do AG, selecionou-se um conjunto de parâmetros para observação das respectivas influências no desempenho do AG, independentemente do problema estudado. Neste caso, os seguintes parâmetros foram analisados: probabilidade de mutação (*pMut*), probabilidade de cruzamento (*pCros*), tamanho da população (*sPop*) e número de gerações (*nGen*).

Empregou-se nesse projeto o planejamento fatorial completo 2^k , com $k = 4$ parâmetros do AG (fatores principais) e três níveis. Os níveis são mostrados como -1 , quando o fator está em um nível mínimo, 0 , quando o fator está no nível central, e $+1$, quando o fator pretendido está em um nível máximo [9]. Os parâmetros e seus níveis correspondentes podem ser vistos na Tabela 1.

Os experimentos realizados têm por objetivo buscar o melhor ajuste dos parâmetros de entrada do Algoritmo Genético de tal modo que sejam encontradas as melhores respostas médias para a função *fitness*. Desta forma, foram executadas

TABLE 1: VALORES PARA OS PARÂMETROS DO AG.

Parâmetros do AG	Codificado	pCros	pMut	sPop	nGen
Mínimo	-1	0.01	0.01	10	10
Central	0	0.50	0.50	105	105
Máximo	1	0.99	0.99	200	200

3 repetições e para cada repetição o algoritmo genético foi executado 10 vezes, retornando a resposta média. Os valores encontrados podem ser observados na Tabela 2.

TABLE 2: VALOR DE *fitness* PARA DIFERENTES AJUSTES DE PARÂMETROS DO AG.

ponto	pCros	pMut	sPop	nGen	fitness
1	0.01	0.01	10	10	14530.9
2	0.01	0.01	10	105	14514.9
3	0.01	0.01	10	200	14491.5
4	0.01	0.01	105	10	14045.6
5	0.01	0.01	105	105	14046.5
6	0.01	0.01	105	200	14039.6
7	0.01	0.01	200	10	13874.72
8	0.01	0.01	200	105	13884.2
9	0.01	0.01	200	200	13897.6
10	0.01	0.5	10	10	14581.6
11	0.01	0.5	10	105	14387.1
12	0.01	0.5	10	200	14496.5
13	0.01	0.5	105	10	14089.0
14	0.01	0.5	105	105	14023.7
15	0.01	0.5	105	200	14051.5
16	0.01	0.5	200	10	13937.5
17	0.01	0.5	200	105	13957.8
18	0.01	0.5	200	200	13925.5
19	0.01	0.99	10	10	14538.6
20	0.01	0.99	10	105	14447.3
21	0.01	0.99	10	200	14548.1
22	0.01	0.99	105	10	14097.0
23	0.01	0.99	105	105	14040.8
24	0.01	0.99	105	200	14050.5
25	0.01	0.99	200	10	13918.7
26	0.01	0.99	200	105	13950.9
27	0.01	0.99	200	200	13941.3
28	0.5	0.01	10	10	14375.9
29	0.5	0.01	10	105	14134.8
30	0.5	0.01	10	200	14235.5
31	0.5	0.01	105	10	13641.1
32	0.5	0.01	105	105	13644.1
33	0.5	0.01	105	200	13570.2
34	0.5	0.01	200	10	13609.0
35	0.5	0.01	200	105	13417.7
36	0.5	0.01	200	200	13426.0
37	0.5	0.5	10	10	14423.0
38	0.5	0.5	10	105	14554.0
39	0.5	0.5	10	200	14503.7
40	0.5	0.5	105	10	13885.1
41	0.5	0.5	105	105	13851.6

42	0.5	0.5	105	200	13886.9
43	0.5	0.5	200	10	13730.6
44	0.5	0.5	200	105	13763.2
45	0.5	0.5	200	200	13820.1
46	0.5	0.99	10	10	14456.4
47	0.5	0.99	10	105	14460.0
48	0.5	0.99	10	200	14486.2
49	0.5	0.99	105	10	13787.9
50	0.5	0.99	105	105	13890.6
51	0.5	0.99	105	200	13926.5
52	0.5	0.99	200	10	13801.2
53	0.5	0.99	200	105	13844.6
54	0.5	0.99	200	200	13787.7
55	0.99	0.01	10	10	14135.3
56	0.99	0.01	10	105	14309.5
57	0.99	0.01	10	200	14275.4
58	0.99	0.01	105	10	13675.3
59	0.99	0.01	105	105	13469.5
60	0.99	0.01	105	200	13527.2
61	0.99	0.01	200	10	13532.2
62	0.99	0.01	200	105	13332.5
63	0.99	0.01	200	200	13322.0
64	0.99	0.5	10	10	14351.8
65	0.99	0.5	10	105	14458.8
66	0.99	0.5	10	200	14415.6
67	0.99	0.5	105	10	13872.5
68	0.99	0.5	105	105	13790.7
69	0.99	0.5	105	200	13756.7
70	0.99	0.5	200	10	13663.3
71	0.99	0.5	200	105	13681.0
72	0.99	0.5	200	200	13581.2
73	0.99	0.99	10	10	14336.7
74	0.99	0.99	10	105	14346.1
75	0.99	0.99	10	200	14389.2
76	0.99	0.99	105	10	13843.0
77	0.99	0.99	105	105	13820.1
78	0.99	0.99	105	200	13817.9
79	0.99	0.99	200	10	13632.4
80	0.99	0.99	200	105	13695.3
81	0.99	0.99	200	200	13670.6

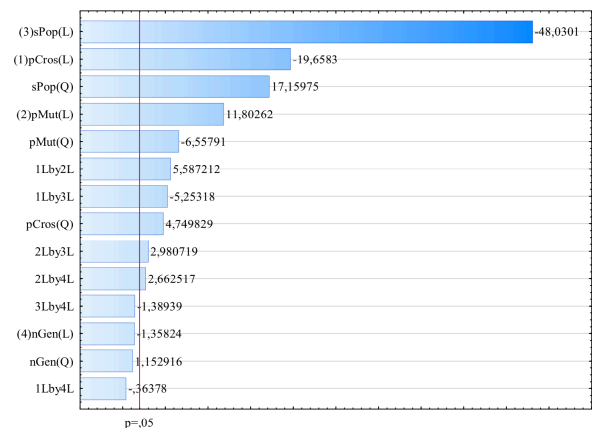


Fig. 4: Diagrama de Pareto.

IV. RESULTADOS E DISCUSSÕES

O diagrama de Pareto apresentado na Figura 4 mostra que os parâmetros que mais influenciaram na minimização do *fitness* são o tamanho da população (*sPop*) e o percentual de crossover (*pCros*) quando avaliados separadamente. No entanto, quando combinados os parâmetros, o *pCros* x *pMut* associado ao *pCros* x *sPop* exerceram maior influência sob o *fitness*.

A relação entre os valores preditos e os valores ajustados (Figura 5) mostram alta correlação entre as variáveis estudadas. Não obstante, destaca-se, ainda, que o valor do coeficiente de determinação encontrado para os modelos estudados foi de 93,48%, fato esse que, conforme descreve [10], indica elevada confiabilidade do modelo, uma vez que R^2 é o valor que mede o efeito da variável independente na variação dependente e, quanto mais próximo de 1, maior a correlação.

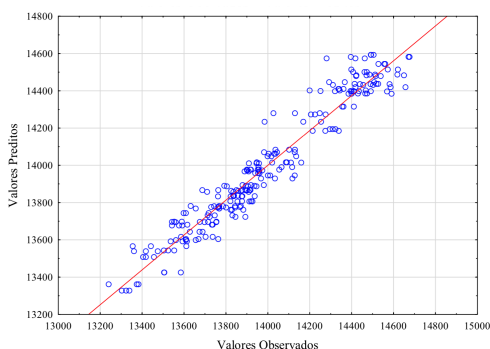


Fig. 5: Valores observados x Valores preditos.

Quando associadas, as variáveis *nGen* e *sPop* (Figura 6), observa-se que, independente dos valores de *nGen*, quanto maior forem os valores de *sPop* melhor será a resposta, corroborando com o descrito no diagrama de Pareto (Figura 4) que apresenta uma correlação direta entre a resposta e a variável *sPop*, ou seja, a medida que uma aumenta a outra possivelmente seguirá um comportamento semelhante.

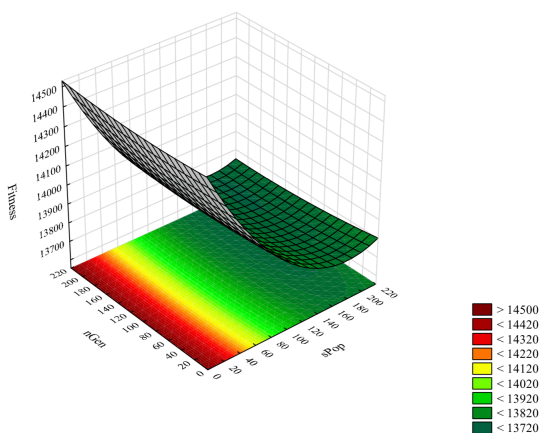


Fig. 6: Superfície de Resposta para a Interação dos parâmetros *nGen* x *sPop*

O valor de *fitness* observado durante a interação entre *sPop* x *pMut* (Figura 7) indica que as melhores condições de ação dessas variáveis são em faixas elevadas de *sPop* e reduzidas de *pMut*. Nesse sentido o diagrama de Pareto (Figura 4) também destaca a relação inversa entre a variável *pMut* e

a resposta *fitness*, uma vez que o índice negativo indica que a redução de uma implicará no aumento da outra.

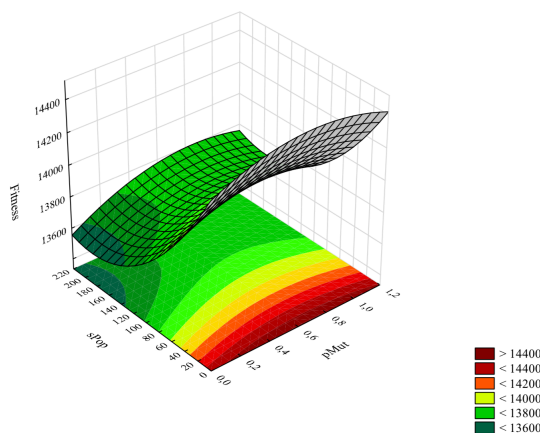


Fig. 7: Superfície de Resposta para a Interação dos parâmetros *nMut* x *sPop*.

Ao correlacionar as variáveis *nGen* e *pMut* (Figura 8) destaca-se que o *nGen* não exerce qualquer influência sobre a resposta, corroborando com o apresentado pelo Diagrama de Pareto (Figura 8). Por outro lado, a *pMut* apresentou o mesmo comportamento observado em outras interações das quais participou, ou seja, sua faixa desejável de atuação está em valores próximos ao parâmetro mínimo estudado (−1).

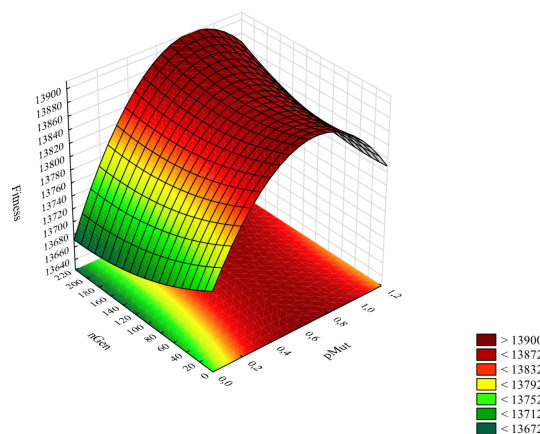


Fig. 8: Superfície de Resposta para a Interação dos parâmetros *nGen* x *pMut*.

No âmbito da correlação entre a *sPop* e a *pCros* (Figura 9) observa-se que a melhor faixa da variável resposta encontra-se em valores próximos ao parâmetro máximo (1) para ambas as variáveis independentes, confirmando, novamente, o comportamento descrito pelo Diagrama de Pareto (Figura 4).

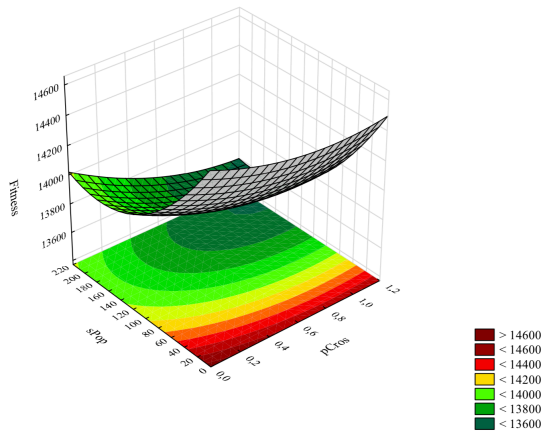


Fig. 9: Superfície de Resposta para a Interação dos parâmetros $nCros$ x $sPop$.

V. CONCLUSÕES

Este trabalho apresentou a utilização de Algoritmos Genéticos (AG) para a área de Esteganografia, mostrando a aplicação da metodologia de Superfície de Resposta (MRS) como método de otimização de seus hiper-parâmetros. Um experimento foi conduzido de maneira a observar quatro hiperparâmetros do AG em questão. O experimento levou a concluir que a variável independente $nGen$ não exerce influência significativa na variável resposta e que as melhores faixas de $fitness$ encontram-se próximas ao parâmetro mínimo de $pMut$ e máximo de $sPop$ e $pCros$.

Neste sentido, conclui-se que após o uso da MSR, evitou-se, tanto quanto possível, os riscos e erros associados ao método de tentativa e erro na definição de parâmetros do AG avaliado. Como observado nos gráficos na seção IV, a MSR indicou a região de valores para melhor configuração de parâmetros que levaram a minimização do valor da função $fitness$ em questão.

REFERENCES

- [1] F. Y. Shih, *Digital Watermarking and Steganography: Fundamentals and Techniques*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2007.
- [2] G. Kipper, *Investigator's Guide to Steganography*. CRC Press, 2003.
- [3] S. Katzenbeisser and F. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, ser. Artech House computer security series. Artech House, 2000.
- [4] S. Mungmode, R. Sedamkar, and N. Kulkarni, "A modified high frequency adaptive security approach using steganography for region selection based on threshold value," *Procedia Computer Science*, vol. 79, pp. 912 – 921, 2016.
- [5] Gangeshwar and J. Attri, "Optimizing image steganography using genetic algorithm," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 24, no. 1, pp. 32–38, 2015.
- [6] D. Montgomery, *Design and Analysis of Experiments, 6th Edition Set*. John Wiley & Sons, Limited, 2007.
- [7] M. Mitchell, *An Introduction to Genetic Algorithms*, 1st ed., ser. Complex Adaptive Systems. The MIT Press, 1996.
- [8] O. Kramer, *Genetic Algorithm Essentials*, ser. Studies in Computational Intelligence. Springer, 2017, vol. 679.
- [9] T.-Y. Wang and K.-B. Wu, "A parameter set design procedure for the simulated annealing algorithm under the computational time constraint," *Computers & Operations Research*, vol. 26, no. 7, p. 665–678, Jun. 1999.

Infering trust in web-based social networks: an analysis from TidalTrust and T-SWEETS algorithms

Edeilson Milhomem Silva¹, Diego Oliveira Rodrigues², Ana Carolina Salgado³ e Silvio Romero Lemos Meira³

¹ Federal University of Tocantins - UFT, Department of Computer Science, Palmas-TO, Brazil

² University of Campinas - UNICAMP, Institute of Computing, Campinas-SP, Brazil

³ Federal University of Pernambuco - UFPE, Informatics Center, PE, Brazil

Data de recebimento do manuscrito: 28/02/2020

Data de aceitação do manuscrito: 03/03/2020

Data de publicação: 09/03/2020

Abstract— This work presents an analysis between two approaches for inferring trust in social networks: TidalTrust and T-SWEETS. The first part of this research is composed by an experiment with the Epinions, Slashdot and Wikipedia datasets in order to define the most suitable type of trust value (integer or fractioned, positive or negative, and so on). Furthermore, the second part of this research reports a comparative analysis between both algorithms. The results indicate that T-SWEETS outperforms TidalTrust in terms of accuracy and maintains the transitivity principle, which is the basic principle of trust.

Keywords—trust inference, social network, recommender systems

I. INTRODUÇÃO

Confiança oferece subsídios para apoiar a sociedade em tomadas de decisões, que vão desde as mais básicas até as mais complexas – é um dos elementos que mantém a sociedade [1]. O conceito de confiança é objeto de pesquisas em diferentes áreas de conhecimento, por exemplo, Psicologia [2, 3], Sociologia [4], Filosofia [5] e Computação [6, 7]. Um trabalho pioneiro sobre confiança em ambientes computacionais foi o desenvolvido por Marsh [4], que concebeu um modelo para aplicação em sistemas de Inteligência Artificial.

As plataformas virtuais sociais possuem uma estrutura que habilita pesquisas avançadas para a identificação de relacionamentos de confiança entre indivíduos. Há três elementos que estão relacionados à confiança que são fortemente discutidos na literatura, de acordo com Reis [8], são eles, subjetividade, contexto e dinamismo.

- I. **subjetividade**: confiança é subjetiva e, portanto, assimétrica, isto é, a confiança de uma pessoa p , em relação a uma pessoa p_1 não é a mesma confiança da pessoa p_1 em relação a p ;
- II. **contexto**: confiança depende do contexto, ou seja, pode ser medida de uma pessoa em relação à outra, em um

grupo ou até mesmo diante de uma determinada situação;

- III. **dinamismo**: confiança é dinâmica, isto é, muda com o tempo - pode aumentar se as experiências forem positivas, e pode diminuir se as experiências forem negativas.

Assim, no âmbito computacional, para a aplicação em redes sociais, Golbeck [7] define algumas propriedades de “confiança”: transitividade, agregabilidade e assimetria. A transitividade representa a possibilidade de transferir a confiança de pessoa a pessoa. A agregabilidade representa a condição que define se uma pessoa procurará informações adicionais para decidir se confia ou não em outra pessoa. Já a assimetria, indica que o nível de confiança de uma pessoa em relação à outra é unilateral.

Nesta perspectiva, diversos algoritmos foram propostos para a inferência de relacionamentos ocultos de confiança em ambientes virtuais sociais, tais como, *TidalTrust* [9]; o RN-Trust, [10]; SUNNY [11, 12]; FlowTrust [8]; e de Mamami [9] e o T-SWEETS [13]. Neste sentido, o presente trabalho tem como objetivo apresentar uma análise do algoritmo T-SWEETS em relação ao algoritmo clássico TidalTrust. A análise é baseada nos resultados apresentados no trabalho de Golbeck [9] e fontes de dados de relacionamentos de confiança oriundos do *Epinions*¹, *Slashdot*² e *Wikipedia*³.

¹Epinions.com: Product Reviews and Consumer Reports: <http://www.epinions.com/>

²Slashdot: News for nerds, stuff that matters: <http://slashdot.org/>

³Wikipédia, a enciclopédia livre: <https://pt.wikipedia.org/>

II. TIDALTRUST E T-SWEETS

Para que o algoritmo *TidalTrust* possa realizar a inferência dos relacionamentos de confiança entre os indivíduos, inicialmente, faz uma seleção dos nós que serão considerados na análise. Nesta seleção são estabelecidos o tamanho máximo dos caminhos e os valores mínimos de confianças entre os nós que poderão ser explorados durante a análise de confiança. Em seguida, com estes valores é possível excluir alguns nós, os considerados caminhos menos irrelevantes, que segundo Golbeck [9], reduziriam a precisão da análise de confiança.

T-SWEETS, um algoritmo desenvolvido por nós [13] para inferência de confiança em ambientes virtuais sociais [13], baseia-se em 4 (quatro) elementos para inferência destes relacionamentos de confiança, são eles: **similaridade entre perfis**: considera as pessoas que possuem perfis similares; **relacionamento de confiança**: considera as relações de confiança de um usuário a em relação a um usuário b atribuída explicitamente por a; ou a probabilidade de a confiar em b dado que existe um caminho de relacionamento de confiança (e. g.) entre eles; **nível de maturidade**: representa o quanto as produções dos usuários são bem avaliadas por outros usuários na rede social; e **reputação**: é a visão que um grupo de pessoas g tem sobre uma pessoa p , determinada pelo grau de confiança que g tem em relação a p .

Destes elementos, o **Relacionamento de Confiança**, será objeto de estudo neste trabalho, já que é o elemento que baseia-se em características similares ao algoritmos clássico da literatura, *TidalTrust* [9]. Este elemento baseia-se nas propriedades de confiança definidas por Golbeck [7], que são: Transitividade, Assimetria e Agregabilidade.

III. EXPERIMENTO

a. Objetivo

Para a análise da eficácia do algoritmo T-SWEETS, inicialmente, foi realizado um experimento considerando os resultados obtidos nos relacionamentos de confiança oriundos do trabalho desenvolvido por Leskovec [14]; e depois foi realizada uma análise em relação ao algoritmo *TidalTrust* [9].

b. Cenário

Foram utilizadas como fonte de relacionamentos de confiança, dados oriundos do Epinions, Slashdot e Wikipedia. Epinions é um ambiente em que os usuários podem realizar avaliações a determinados produtos (e.g. notebook e smart-phone) ou avaliar pessoas (os próprios avaliadores). A partir das avaliações realizadas entre estes avaliadores forma-se uma rede social de relacionamentos de confiança. Slashdot, um site voltado para publicações de notícias de tecnologia. Neste ambiente, os usuários podem associar tags a outros usuários classificando-os como amigos ou inimigos. Wikipedia, uma enciclopédia online em que as pessoas podem colaborar livremente para o seu desenvolvimento. Os usuários podem passar para a categoria de administradores da enciclopédia, a partir das avaliações que podem receber de outros usuários, que são positivas, negativas e neutras. Estas fontes de dados são oriundas do trabalho de Leskovec [14], e estão disponíveis no projeto SNAP4, desenvolvido

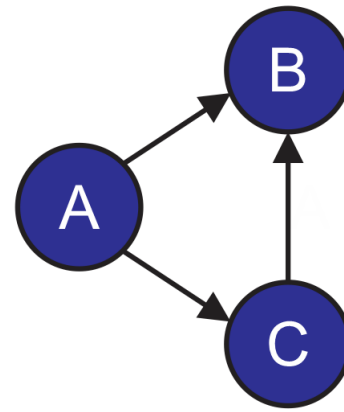


Figura 1: Estrutura de uma tríade

por pesquisadores de Stanford. O trabalho de Leskovec [14] analisa estes relacionamentos a partir de associações ternárias (tríades), que são relacionamentos formados pelo conjunto de três nós, conforme apresentado na Figura 1.

Cada relacionamento forma uma relação de confiança de valores $\{-1, +1\}$, em que o valor positivo indica que há a confiança entre os nós, enquanto o valor negativo indica a ausência de confiança. De acordo com este cenário, a Tabela 1 apresenta a proporção dos dados originados de cada uma das fontes supracitadas.

TABELA 1: CLASSIFICAÇÕES E QUANTIDADES DE TRÍADES NAS FONTES DE DADOS DO EPINIONS, SLASHDOT E WIKIPÉDIA [14]. TRÍADE.

Tríade T_i	$ T_i $	$p(T_i)$
T_3 + + +	11.640.257	87%
T_2 + - -	947.855	7,1%
T_1 + + -	698.023	5,2%
T_0 - - -	89.272	0,7%

As tríades podem variar conforme o valor atribuído aos relacionamentos na rede social, conforme apresentado na Tabela 1 Para executar o algoritmo T-SWEETS nas bases supracitadas, o tamanho do caminho entre dois nós também foi limitado a 3 – tamanho esse limitado pela tríade.

c. Resultados da Primeira Fase do Experimento

De acordo com as associações ternárias e os valores positivos e negativos, alguns cenários para a inferência de relacionamentos de confiança podem ser representados. A Figura 2 apresenta estes cenários.

Leskovec [14] apresentou um estudo de algumas possibilidades para a construção de tríades, de acordo com os valores $\{-1, +1\}$. Neste estudo, consta que dado dois relacionamentos com valores de confiança quaisquer (+1 ou -1), qual a probabilidade do terceiro ser positivo ou negativo. O relacionamento com a seta vermelha na Figura 2 representa este terceiro relacionamento. Este quadro de probabilidades pode ser visualizado na Tabela 2.

As probabilidades apresentadas na Tabela 2 foram utilizadas para realizar uma análise dos algoritmos objeto de estudo deste trabalho. Assim, para cada uma das configurações apresentadas (Figura 2 e Tabela 2), a Tabela 3 apresenta os resultados obtidos.

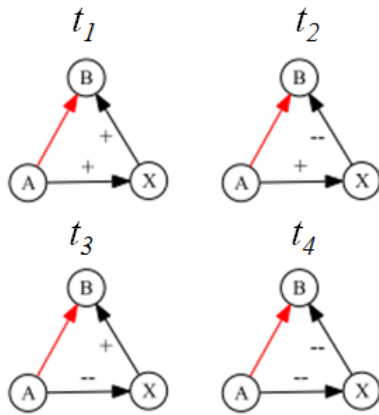


Figura 2: Possibilidades de tríades que podem ser usadas nas análises

TABELA 2: PROBABILIDADES DE INFERÊNCIA DE RELACIONAMENTOS POSITIVOS E NEGATIVOS PARA CADA TRÍADE. ([14] - ADAPTADA)

t_i	$P(+)$	$P(-)$
1	0,97	0,03
2	0,54	0,46
3	0,30	0,70
4	0,23	0,77

TABELA 3: PORCENTAGEM DA QUANTIDADE DE RELACIONAMENTOS POSITIVOS EM CADA TRÍADE

t_i	$P(+)$		
	Leskovec [14]	<i>TidalTrust</i>	T-SWEETS
4	97%	100%	100%
3	54%	0%	0%
2	30%	0%	0%
1	23%	100%	100%

Conforme resultados apresentados na Tabela 3, tanto os resultados obtidos do T-SWEETS, quanto do *TidalTrust*, são valores muito determinísticos. Isto é, ou o grau de confiança foi nulo ou foi integral (100%). Contudo, esta precisão pode ser justificável, uma vez que foram utilizados valores inteiros positivos e negativos. Para evitar esta limitação, o ideal é utilizar valores fracionados em uma escala de $[-1 \dots +1]$.

1. Limitação identificadas

Leskovec [14] faz uso dessas tríades para analisar os seguintes princípios: “o amigo do meu amigo é meu amigo”; “o amigo do meu inimigo é meu inimigo”; “o inimigo do meu amigo é meu inimigo”; e “o inimigo do meu inimigo é meu amigo”. Além disso, considera apenas valores inteiros de confiança, $\{-1, +1\}$. Essas características inviabilizam a utilização de um caminho no grafo social maior que 3, já que tornaria ainda mais ineficaz a avaliação de confiança entre dois indivíduos. Essa ineficácia deve-se, principalmente, por tornar totalmente transitiva a relação de confiança entre dois nós *A* e *B*, independente do tamanho do caminho entre eles. Esta ineficácia também ocorre em um caminho 3, porém, nestes casos, há uma representação dos princípios supraci-

tados (por exemplo, o amigo do meu inimigo é meu inimigo) – o propósito de Leskovec. Tanto o T-SWEETS quanto o *TidalTrust*, realizam inferência com maior eficácia com entradas positivas fracionárias no intervalo $[0, 1]$. Assim, é possível a obtenção de valores de confiança não discretos entre indivíduos da rede. Ou seja, a inferência de confiança não é totalmente transitiva, um dos princípios da análise de confiança.

d. Resultados da Segunda Fase do Experimento

Para realizar uma análise sobre valores fracionados e positivos, foi usado o conjunto de dados apresentados no trabalho de [9], que são usados para analisar o algoritmo *Tidaltrust*.

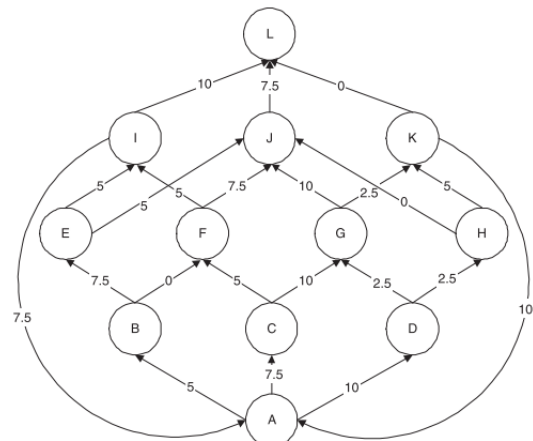


Figura 3: Grafo de relacionamentos utilizado para analisar os algoritmos ([15])

Após a aplicação de ambos algoritmos, *TidalTrust* e T-SWEETS, foram inferidos valores de confiança entre os nós apresentados no grafo da Figura 3. Com a aplicação dos algoritmos no cenário apresentado (Figura 3), o valor médio de confiança e o desvio padrão de ambos os algoritmos são apresentados na Tabela 4.

TABELA 4: MÉDIA E DESVIO PADRÃO DOS VALORES DOS QUADROS 5 E 6.

	Média	Desvio Padrão
T-SWEETS	4,73	3,59
<i>TidalTrust</i>	6,27	3,13

De acordo com o que pode ser observado na Tabela 4, a média dos valores de confiança entre os relacionamentos é maior no algoritmo *TidalTrust*. O valor menor obtido no T-SWEETS é em decorrência de que este se baseia no princípio que quanto maior os caminhos de confiança que unem dois nós, menor é a relação de confiança entre eles. Assim, T-SWEETS considera que pessoas que estejam mais próximas, têm maior tendência que haja maior relação de confiança entre elas, já que esta relação de confiança é baseada em informações (e.g. trocas de experiências) que as pessoas possuem umas sobre as outras.

Além da dispersão (existem pontos sobrepostos) dos valores obtidos com a execução dos algoritmos *TidalTrust* e T-SWEETS, são aprestadas também as tendências dessas dispersão, considerando o aumento na distância do caminho so-

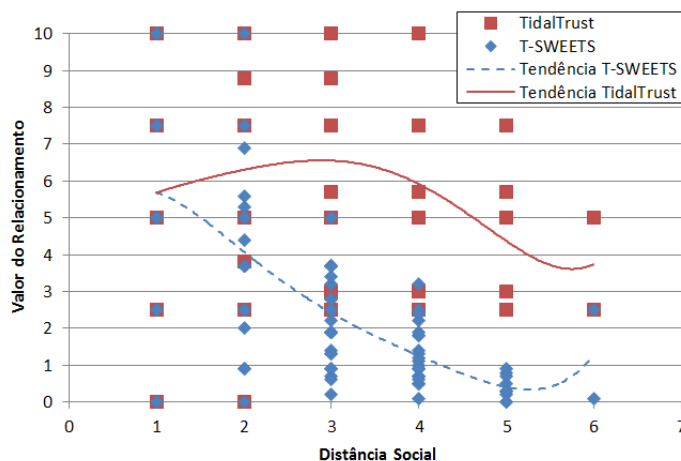


Figura 4: Dispersão dos valores obtidos para o grafo da Figura 3 com os algoritmos *TidalTrust* e T-SWEETS.

cial que liga dois nós no grafo. Essas linhas de tendências foram calculadas a partir da interpolação dos pontos na dispersão em um polinômio de grau 5. De acordo com o gráfico apresentado na Figura 4 (linha pontilhada) é possível observar que no T-SWEETS a tendência é decrescente, na maior parte de sua extensão, à medida que a distância social aumenta. Ao mesmo tempo, o algoritmo *TidalTrust* apresenta um comportamento sem padrão em relação à distância social. Diante da premissa que quanto maior a distância social entre dois nós há uma tendência de menor relação de confiança entre eles, o T-SWEETS apresenta um comportamento mais adequado.

IV. CONCLUSÕES

A partir das investigações realizadas no presente trabalho, constatou-se que a abordagem T-SWEETS [13] considera com maior eficácia a relação entre o grau de confiança e a distância social em que duas pessoas estão na rede social, uma vez que tem uma tendência natural a diminuir o nível de confiança entre dois indivíduos se estes estão a uma distância social maior um do outro. Ao mesmo tempo, a abordagem *TidalTrust* [9] não possui um comportamento padrão considerando este afastamento entre as pessoas (a distância social entre elas).

Além disso, foi constatado também com esta análise que a utilização da escala que determina o grau de confiança entre dois indivíduos na rede social pode influenciar nos resultados das inferências dos relacionamentos de confiança que estão implícitos no ambiente. Assim, é sugerido que, em uma eventual implantação das abordagens em uma plataforma social, o ideal é que seja utilizada uma escala fracionada $[-1 \dots +1]$ ao invés de valores inteiros positivos e negativos.

Como trabalho futuro, espera-se implantar os algoritmos de confiança apresentado no presente trabalho em momentos diferentes e verificar a eficácia de ambos em um ambiente real e, após, confrontar os resultados com os resultados obtidos no presente estudo. Além disso, espera-se analisar as demais características presentes no T-SWEETS e que também estejam presentes em outras abordagens disponíveis na literatura para identificar possíveis pontos fortes ou fragilidades existentes. Nesta perspectiva, espera-se obter análises que possam direcionar-nos a um formalismo ideal de um algoritmo para inferência de relacionamentos ocultos de confi-

ança em ambientes virtuais sociais.

REFERÊNCIAS

- [1] Y. Yamamoto, "A morality based on trust: Some reflections on japanese morality," 1990.
- [2] T. Yamagishi and M. Yamagishi, "Trust and commitment in the united states and japan," *Motivation and Emotion*, vol. 18, pp. 129–166, 1994.
- [3] T. Yamagishi, K. S. Cook, and M. Watabe, "Uncertainty, trust, and commitment formation in the united states and japan," 1998.
- [4] H. Murayama, A. Taguchi, S. Ryu, S. Nagata, and S. Murashima, "Institutional trust in the national social security and municipal health-care systems for the elderly in Japan," *Health Promotion International*, vol. 27, no. 3, pp. 394–404, 09 2011.
- [5] F. J. McDonald, "Worlds of Truth: A Philosophy of Knowledge By Israel Scheffler," *Analysis*, vol. 70, no. 2, pp. 383–385, 02 2010.
- [6] S. Marsh, *Formalising Trust as a Computational Concept*, ser. Technical report (University of Stirling, Department of Computing Science and Mathematics). University of Stirling, 1994.
- [7] S. P. Marsh, "Formalising trust as a computational concept," Tech. Rep., 1994.
- [8] S. Ries, J. Kangasharju, and M. Mühlhäuser, "A classification of trust systems," in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, R. Meersman, Z. Tari, and P. Herrero, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 894–903.
- [9] J. Golbeck, "Computing and applying trust in web-based social networks," Ph.D. dissertation, 2005.
- [10] M. Taherian, M. Amini, and R. Jalili, "Trust inference in web-based social networks using resistive networks," in *2008 Third International Conference on Internet and Web Applications and Services*, June 2008, pp. 233–238.
- [11] U. Kuter, "Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models," in *In Proceedings of the National Conference on Artificial Intelligence (AAAI, 2007)*.
- [12] U. Kuter and J. Golbeck, "Using probabilistic confidence models for trust inference in web-based social networks," *ACM Trans. Internet Technol.*, vol. 10, no. 2, Jun. 2010. [Online]. Available: <https://doi.org/10.1145/1754393.1754397>
- [13] E. M. Silva, D. O. Rodrigues, J. G. Souza, P. F. Brito, A. C. Salgado, S. R. Meira, and J. A. Costa, "Inferring hidden trust relationships in social networks for encouraging collaboration and cooperation among individuals," in *Proceedings of the 19th International Conference on Collaboration and Technology - Volume 8224*. Berlin, Heidelberg: Springer-Verlag, 2013, p. 42–60.
- [14] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1361–1370.

- [15] M. Taherian, M. Amini, and R. Jalili, "Trust inference in web-based social networks using resistive networks." *2008 Third International Conference on Internet and Web Applications and Services*, pp. 233–238, 2008.