Resolving the Optimal Selection of a Natural Reserve using the Particle Swarm Optimisation by Applying Transfer Functions

Boris Almonacid

Unaffiliated
boris.almonacid.g@mail.pucv.cl
orcid:0000-0002-6367-9802

Abstract. The optimal selection of a natural reserve (OSRN) is an optimisation problem with a binary domain. To solve this problem the metaheuristic algorithm Particle Swarm Optimization (PSO) has been chosen. The PSO algorithm has been designed to solve problems in real domains. Therefore, a transfer method has been applied that converts the equations with real domains of the PSO algorithm into binary results that are compatible with the OSRN problem. Four transfer functions have been tested in four case studies to solve the OSRN problem. According to the tests carried out, it is concluded that two of the four transfer functions are apt to solve the problem of optimal selection of a natural reserve.

Keywords: Biodiversity conservation, Nature-inspired algorithms, Particle Swarm Optimisation, Metaheuristic.

1 Introduction

The problem of an optimal selection of natural reserve (OSNR) [13] is within the framework of one of the objectives of sustainable development, specifically in objective number 15 called "life on land" [1]. The OSNR problem is that employing a set of parameters, variables and restrictions, an excellent choice can be determined to choose which or which will be the natural reserves that will be a priority for their protection. The mathematical model OSNR covers the objective of maximising the number of types of species to be protected. Also, the mathematical model considers that organisations or governments have a limited budget.

In order to solve the OSNR problem, the Particle Swarm Optimisation (PSO) metaheuristic algorithm [7,8] has been used. Several problems have been solved using the PSO algorithm as energy management strategy [4], electromagnetic [14], economic [17,6,12], task assignment [15], manufacture [5], and underwater wireless sensor networks [18] problems. Due to the applicability of a large number of problems, the PSO algorithm has been chosen. However, the challenge is to be able to integrate this algorithm that is designed to solve problems in real domains in the OSNR problem that is modelled in binary domain. To do this, a method using transfer functions and discretisation from a real domain to a discrete domain is used [9,10,16]

The main benefits expected from this work are as follows: The resolution of the OSNR problem using the PSO metaheuristic algorithm. A comparative study of the results of the algorithms taking aspects such as the robustness and best values found. Report the new optimal values found using four transfer functions. To our knowledge, these contributions have not been reported.

2 Materials and Methods

2.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a stochastic optimization technique inspired by the social behaviour of bird quartering or aquaculture. PSO has been designed to solve optimization problems with real domains. Two equations are fundamental in this algorithm. Equation 1 consists of a velocity vector of the particle. Equation 2 consists of the sum of the particle with the velocity, product of this sum will give us a new location of the particle.

$$v_i^{t+1} = v_i^t + c_1 * rand * (pbest_i - p_i^t) + c_2 * rand * (gbest - p_i^t) \quad (1)$$

$$p_i^{t+1} = p_i^t + v_i^{t+1} \quad (2)$$

Where v_i^t corresponds to the velocity of particle p_i in iteration t, c_1 and C_2 is an acceleration coefficient, $rand$ corresponds to a value that belongs to the interval

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26941v2 | CC BY 4.0 Open Access | rec: 29 May 2018, publ: 29 May 2018

$[0, 1] \in \mathbb{R}$, p_i^t is the current position of particle p_i at iteration t, *pbest* is the best feasible solution that particle p_i has been able to obtain and and *gbest* indicates the best solution obtained from the swarm of particles.

The steps of the PSO algorithm for maximization problems are described in Algorithm 1.

Algorithm 1: Particle Swarm Optimisation

Initialize a population of particles with random values positions and velocities.
while *termination condition not reached* **do**
    for *each particle* p_i **do**
        Adapt velocity of the particle using equation 1.
        Update the position of the particle using equation 2.
        Evaluate the fitness $f(p_i)$.
        if $f(p_i) > f(pbest_i)$ **then**
            $pbest_i \leftarrow p_i$.
        end
        if $f(p_i) > f(gbest)$ **then**
            $gbest \leftarrow p_i$.
        end
    end
end
Get the best solution *pbest* from the population.

2.2 The mathematical model for Optimal Selection of a Natural Reserve

The problem of the optimal selection of one or some natural reserves (OSNR), corresponds that employing a series of preselected natural reserves and a series of species that are distributed in those natural reserves, it is possible to determine which are the adequate reserves to protect.

Additionally, certain restrictions must be met: the first restriction consists of covering the most significant number of types of species living within the space of the natural reserve; The second restriction is that a limited budget should try to comply with the first restriction.

The OSNR problem has been modelled by [13]. Table 1 shows the parameters, variables and iterators that make up the mathematical model.

Table 1. Symbology, description, range and domain for parameters, variables and iterators

Symbol	Description	Type	Range	Domain
A	The number of areas	Parameter	$\mathbb{N}^+$	-
S	The number of species	Parameter	$\mathbb{N}^+$	-
i	The index of areas	Iterator	$i \in \{1, \ldots, A\}$	-
j	The index of species	Iterator	$j \in \{1, \ldots, S\}$	-
M_{ij}	The areas-species incidence matrix	Parameter	Binary matrix	-
B	The total budget	Parameter	$\mathbb{N}^+$	-
K_i	The cost K_i of each area i	Parameter	$\mathbb{N}^+$	-
x_i	A vector of areas with a size A	Variable	-	Binary
y_j	A vector of species with a size S	Variable	-	Binary

The incidence matrix M_{ij} indicate what kind of species live in the pre-selected nature reserve (see table 2). The areas-species incidence matrix has a binary domain and size $A \times S$, the meaning of each value for M_{ij} is given by equation 3.

$$M_{ij} = \begin{cases} 1 \text{ if area } i \text{ conserves species } j. \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

Table 2. Incidence matrix

	Species 1	Species 2	Species 3	Species 4	Species 5	Species 6	Species 7	Species 8
Area 1	0	0	1	1	1	1	1	0
Area 2	1	1	1	1	0	0	0	0
Area 3	0	0	0	0	1	1	1	1
Area 4	1	1	1	0	0	0	0	0
Area 5	0	0	0	1	0	1	1	1

The vectors of variables have the following characteristics:

- Variables x_i: A vector of areas with a size A in that $x_i \in \{0, 1\}$, the meaning of each value for x_i is given by equation 4.
- Variables y_j: A vector of species with a size S in that $y_j \in \{0, 1\}$, the meaning of each value for y_j is given by equation 5.

$$x_i = \begin{cases} 1 \text{ if site } i \text{ is included in the final selection of nature reserves.} \\ 0 \text{ otherwise.} \end{cases} \tag{4}$$

$$y_j = \begin{cases} 1 \text{ if species } j \text{ is covered.} \\ 0 \text{ otherwise.} \end{cases} \tag{5}$$

The objective function it's determined by equation 6

$$\text{maximize} \quad \sum_{j=1}^{S} y_j \tag{6}$$

Subject to constraints 7 and 8.

$$\sum_{i=1}^{A} x_i M_{i,j} \geq y_j \qquad \forall_j, \tag{7}$$

$$\sum_{i=1}^{A} x_i K_i \leq B \tag{8}$$

2.3 Method that transfers functions from the Real domain to the Binary domain

Because the OSNR problem is a binary optimization problem and PSO is an algorithm that has been designed for problems in real numbers. A series of steps must be taken to adapt the PSO algorithm to support binary problems [9,10,16] (see figure 1).

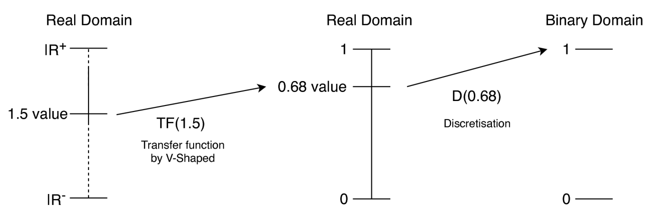


Fig. 1. Example of the method that transfers functions from the Real domain to the Binary domain

The first step is to take a particle p_i from the population and restrict the value in each cell of the variable matrix x_i to a range $[0, 1] \in \mathbb{R}$. To perform this step, it is necessary to apply a transfer function (see equations 9, 10, 11, and 12). The transfer functions 9, 10, 11, and 12 can be visualized in figure 2.

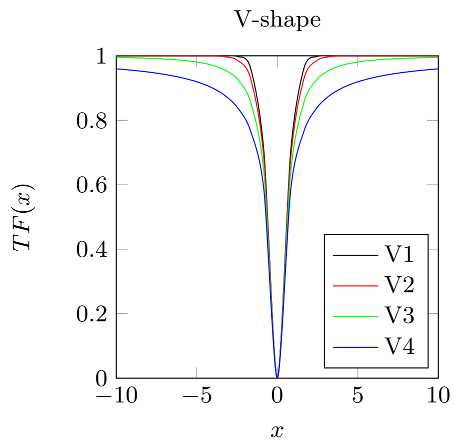
$$\text{PSO-V1 TF1}(x) = \left| erf\left(\frac{\sqrt{\pi}}{2}x\right) \right| = \left| \frac{\sqrt{2}}{\pi} \int_{0}^{\frac{\sqrt{\pi}}{2}x} e^{-t^2} dt \right| \tag{9}$$

$$\text{PSO-V2 TF2}(x) = |tanh(x)| \tag{10}$$

$$\text{PSO-V3 TF3}(x) = \left|\frac{x}{\sqrt{1+x^2}}\right| \tag{11}$$

$$\text{PSO-V4 TF4}(x) = \left|\frac{2}{\pi} arctan\left(\frac{\pi}{2}x\right)\right| \tag{12}$$

Fig. 2. Visualization of V-shaped families of transfer functions



The second step is to convert the results obtained through the transfer functions to binary numbers, for which equation 13 is used.

$$D(x_{k[i]}) = \begin{cases} 1 & \text{if } rand \leq F(x_{k[i]})) \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

3 Results and Discussion

The data set to reproduce the experiment is available in [2], and the results in raw data are available in [3].

3.1 Test configuration

The parameters used by PSO in all configurations (PSO-V1, PSO-V2, PSO-V3 and PSO-V4) are:

- The number of particles is 25.
- The values for c_1 and c_2 is 1.0.
- The value for $vmin$ is 1.
- The values for $vmax$ is 10.

Table 3 describes 4 cases of studies of the OSNR problem that have been used to compare the results of the different configurations of the PSO algorithm.

Table 3. Description of the problems for an optimal choice of a nature reserve. The meanings of the columns are: instance number, number of areas (A), number of species (S), density number and percentage of non-zeroes in incidence matrix (M_{ij}) and the budget total (B)

Instance number	Areas	Species	Density	Density-%	Budget total
1	20	100	1008	50.4 %	£200
2	20	100	575	28.75 %	£300
3	20	200	1962	49.05 %	£250
4	20	200	1199	29.98 %	£250

All versions of PSO have been run independently 31 times, with a maximum execution time of 1 seconds.

The PSO was coded in Java version 1.8.0_92 using IntelliJ IDEA 15 (15.0.6). For running the PSO has been used Java(TM) SE Runtime Environment (build 1.8.0_92). Finally the hardware features that have been run instances are a MacBook Pro computer (Retina, 13-inch, Late 2013) with an Intel Core i5 (Processor Speed = 2,4 GHz; Number of processors = 1; Total number of cores = 2; L2 cache per core = 256KB; L3 cache = 3MB), 4 GB RAM 1600 MHz DDR3 and a Video Card Intel Iris 1536 MB running OS X El Capitan version 10.11.2 (15C50).

3.2 Results

The results for the PSO-V1, PSO-V2, PSO-V3 and PSO-V4 algorithms are described below. Table 4, 5, 6 and 7 are given the following attributes: Column 1 (ID) corresponds to the identifier assigned to each test problem. Column 2 (Best) and column 3 (Worst) indicate the best and worst optimum value found. Column 4 (Mean), column 5 (St. Dev), column 6 (Med) and column 7 (IQR) describe the mean, standard deviation, median and interquartile range respectively.

Figures 3, 4, 5 and 6 describe the quality of the solutions obtained in the execution of the OSNF problem using the four transfer functions applied in PSO.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26941v2 | CC BY 4.0 Open Access | rec: 29 May 2018, publ: 29 May 2018

Table 4. Results for PSO-V1

ID	Best	Worst	Mean	St. dev	Med	IQR
01	100	95	98.12903	1.2039362	98	1.5
02	87	76	82.09677	2.4406944	82	2.5
03	200	197	199.03226	0.8749808	199	2.0
04	185	173	181.09677	2.7246387	182	3.0

Table 5. Results for PSO-V2

ID	Best	Worst	Mean	St. dev	Med	IQR
01	100	95	98.00000	1.2110601	98	2.0
02	89	76	82.80645	3.3706513	82	3.0
03	200	197	198.87097	0.8058923	199	1.0
04	189	173	181.58065	3.9730273	181	4.5

Table 6. Results for PSO-V3

ID	Best	Worst	Mean	St. dev	Med	IQR
01	100	92	97.51613	1.8050109	98	1.5
02	86	78	82.03226	2.3449786	82	3.5
03	200	197	198.80645	0.7924374	199	1.0
04	187	172	180.93548	3.1930705	181	4.0

Table 7. Results for PSO-V4

ID	Best	Worst	Mean	St. dev	Med	IQR
01	100	96	97.6129	1.2021486	97	1.5
02	87	78	82.6129	2.6163259	83	5.0
03	200	196	198.6452	0.9146361	199	1.0
04	187	177	181.9032	2.4406944	182	4.0

Because the OSRN problem corresponds to a maximization problem, the boxes that are in a higher position in the cash flow diagram versus the other boxes in the chart indicate that they have a better quality in their solutions. In addition, the smaller the length of the box, it means that the version of the algorithm has a more robust behaviour [11]. Then, it is considered that an algorithm is robust when in the execution of the algorithm it always obtains the same value, or very close values between the outputs of the algorithm. In other words, it has a standard deviation of zero or very close to zero in the results obtained.

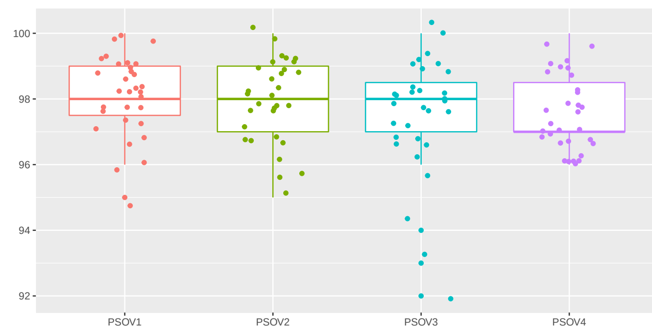


Fig. 3. Boxplot in which the distribution of the result set for problem 1 is displayed.

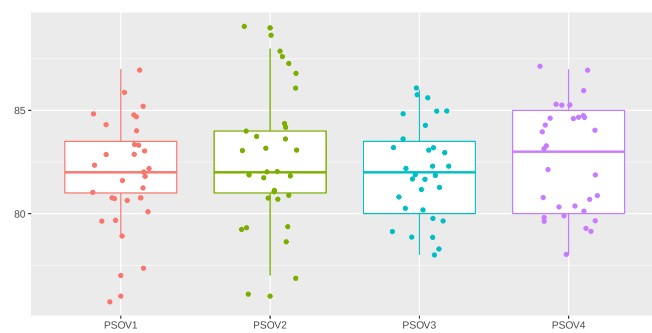


Fig. 4. Boxplot in which the distribution of the result set for problem 2 is displayed.

3.3 Discussion

Table 8 shows the best results obtained through the different versions of PSO. In problem 01 using the configurations PSO-V1, PSO-V2, PSO-V3, and PSO-V4 they obtained solutions that managed to cover 100 of 100 protected species

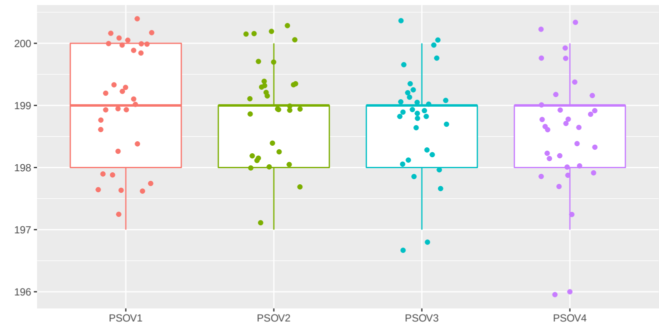


Fig. 5. Boxplot in which the distribution of the result set for problem 3 is displayed.

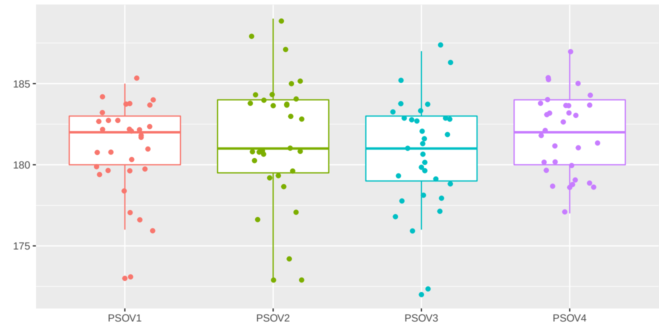


Fig. 6. Boxplot in which the distribution of the result set for problem 4 is displayed.

in natural reserves. In problem 02 the best solution obtained is the PSO-V2 configuration, which reached 89 out of 100 species. In problem 03, the PSO-V1, PSO-V2, PSO-V3 and PSO-V4 configurations achieve solutions that cover 200 of 200 protected species in natural reserves. In problem 04 the best solution obtained is the PSO-V2 configuration, which reached 189 of 200 species.

Table 9 shows a summary focused on the standard deviation of the results obtained. In relation, with the best standard deviation obtained for each problem using the four configurations, it can be seen that the PSO-V3 and PSO-V4 configurations have 2 better standard deviations compared to the other PSO-V1 and PSO-V2 configurations (see numbers in bold). The row "St. dev. Sum" shows the sum of each problem of the standard deviations of the four PSO configurations, in which, the PSO-V4 configuration obtained the sum of the standard deviations lower compared to the rest configurations. The PSO-V3 configuration reached the smallest standard deviation for problem 03, and the PSO-V2 configuration obtained the most significant standard deviation for problem 04.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26941v2 | CC BY 4.0 Open Access | rec: 29 May 2018, publ: 29 May 2018

Table 8. Optimum values of the tests performed by the PSO-V1, PSO-V2, PSO-V3, and PSO-V4 configurations

Problem	PSO-V1	PSO-V2	PSO-V3	PSO-V4
01	**100**	**100**	**100**	**100**
02	87	**89**	86	87
03	**200**	**200**	**200**	**200**
04	185	**189**	187	187

Table 9. Standard deviation of the tests performed by the PSO-V1 PSO-V2 PSO-V3 PSO-V4 configurations.

Problem	PSO-V1	PSO-V2	PSO-V3	PSO-V4
01	1.2039	1.2111	1.8050	**1.2021**
02	2.4407	3.3707	**2.3450**	2.6163
03	0.8750	0.8059	**0.7924**	0.9146
04	2.7246	3.9730	3.1931	**2.4407**
St. dev. Sum	7.2443	9.3606	8.1355	**7.1738**
St. dev. Min	0.8750	0.8059	**0.7924**	0.9146
St. dev. Max	2.7246	**3.9730**	3.1931	2.6163

4 Conclusions

In this research the binary problem of an optimal selection of a natural reserve has been considered, also considering restrictions of a limited budget. To solve this problem, the algorithm Particle Swarm Optimization has been used. Since PSO is an algorithm designed to solve real problems, a method has been used that converts the variables from a real domain to a binary domain. Finally, it can be concluded that:

- According to table 8, the best solutions have been found by the PSO-V2 configuration.
- According to table 9, the most robust results have been achieved by the PSO-V4 configuration.
- According to the results obtained, it is recommended to use the PSO-V3 and PSO-V4 configurations to solve the OSNR problem.

As future work, the functions of transfers in other metaheuristic algorithms can be applied in the resolution of the OSNR problem.

5 Conflicts of Interest

The authors declare that they have no conflicts of interest.

6 Acknowledge

Boris Almonacid is supported by Complexity Science Research Group; by Animal Behaviour Society, USA (Developing Nations Research Awards 2016) and by Ph.D (h.c) Sonia Alvarez, Chile.

References

1. sustainable development goals: 17 goals to transform our world 2016 (2016), http://www.un.org/sustainabledevelopment/
2. Almonacid, B.: Dataset (May 2018). https://doi.org/10.6084/m9.figshare.6279149, https://figshare.com/articles/_/6279149/0
3. Almonacid, B.: Results (May 2018). https://doi.org/10.6084/m9.figshare.6279137, https://figshare.com/articles/_/6279137/0
4. Chen, Z., Xiong, R., Wang, K., Jiao, B.: Optimal energy management strategy of a plug-in hybrid electric vehicle based on a particle swarm optimization algorithm. Energies **8**(5), 3661–3678 (2015)
5. Durán, O., Rodriguez, N., Consalter, L.A.: Collaborative particle swarm optimization with a data mining technique for manufacturing cell design. Expert Systems with Applications **37**(2), 1563–1567 (2010)
6. Gaing, Z.L.: Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE transactions on power systems **18**(3), 1187–1195 (2003)
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Neural Networks, 1995. Proceedings., IEEE International Conference on. vol. 4, pp. 1942–1948 vol.4 (Nov 1995)
8. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of machine learning, pp. 760–766. Springer (2011)
9. Mirjalili, S., Hashim, S., Taherzadeh, G., Mirjalili, S., Salehi, S.: A study of different transfer functions for binary version of particle swarm optimization. GEM'11 (2011)
10. Mirjalili, S., Lewis, A.: S-shaped versus v-shaped transfer functions for binary particle swarm optimization. Swarm and Evolutionary Computation **9**, 1–14 (2013)
11. Osaba, E., Diaz, F., Carballedo, R., Onieva, E., Perallos, A.: Focusing on the golden ball metaheuristic: an extended study on a wider set of problems. The Scientific World Journal **2014** (2014)
12. Park, J.B., Lee, K.S., Shin, J.R., Lee, K.Y.: A particle swarm optimization for economic dispatch with nonsmooth cost functions. IEEE Transactions on Power systems **20**(1), 34–42 (2005)
13. Polasky, S., Camm, J.D., Garber-Yonts, B.: Selecting biological reserves cost-effectively: an application to terrestrial vertebrate conservation in oregon. Land Economics **77**(1), 68–78 (2001)
14. Robinson, J., Rahmat-Samii, Y.: Particle swarm optimization in electromagnetics. IEEE transactions on antennas and propagation **52**(2), 397–407 (2004)
15. Salman, A., Ahmad, I., Al-Madani, S.: Particle swarm optimization for task assignment problem. Microprocessors and Microsystems **26**(8), 363–371 (2002)
16. Saremi, S., Mirjalili, S., Lewis, A.: How important is a transfer function in discrete heuristic algorithms. Neural Computing and Applications **26**(3), 625–640 (2015)

17. Selvakumar, A.I., Thanushkodi, K.: A new particle swarm optimization solution to nonconvex economic dispatch problems. IEEE transactions on power systems **22**(1), 42–51 (2007)
18. Zhang, Y., Liang, J., Jiang, S., Chen, W.: A localization method for underwater wireless sensor networks based on mobility prediction and particle swarm optimization algorithms. Sensors **16**(2), 212 (2016)