

VND-based Local Search Operator for Equality Constraint Problems in PSO Algorithm

Felipe O Mota^{a,1}, Elizabeth F Wanner^{b,2}, Eduardo J S Luz^{a,1}
and Gladston J P Moreira^{a,1}

^a *Department of Computing, Universidade Federal de Ouro Preto
Ouro Preto, Brazil 35400-000*

^b *Lab for Intelligent Collectives Engineering, Aston University
Birmingham, UK B4 7ET*

Abstract

This paper presents a hybrid PSO algorithm with a VND-based operator for handling equality constraint problems in continuous optimization. The VND operator can be defined both as a local search and a kind of elitism operator for equality constraint problems playing the role of “fixing” the best estimates of the feasible set. Experiments performed on benchmark problems suggest that the VND operator can enhance both the convergence speed and the accuracy of the final result.

Keywords: Equality Constraints Problem, Hybrid Algorithm, Non-Linear Problem.

1 Introduction

Although Evolutionary Algorithms (EA's) are a powerful tool in optimization, they have been originally developed to solve unconstrained problems [7]. When

¹ Email: flpccj@gmail.com, eduluz@iceb.ufop.br, gladston@iceb.ufop.br

² Email: e.wanner@aston.ac.uk

it comes to constrained problems, the randomness of the algorithm may lead into unfeasible areas, especially under equality constraints, resulting in zero-volume objects [4]. The traditional way to treat these equality constraints is using the penalty method, transforming the constrained problem into an approximate unconstrained one [4]. Although using the penalty could yield progress with the EA's, it is not enough to reach good solutions. An improvement one can do to EA's is the hybridizing the algorithm, using the precision of local searches.

Equality constraints optimization problems are ordinarily great, much harder than inequality-constrained ones, for the case of general nonlinear constraints. Besides, the usage of EA's technique, even the inequality constraint case is not fully solved yet, and there are only preliminary studies for dealing with equality constraints [4].

The main reason behind the hybridization of different algorithms is to exploit the complementary features of different optimization techniques [2]. The developing a successful hybrid approach is a hard task and the literature shows that it is nontrivial to generalize. A hybrid algorithm might work well for specific problems, but it might perform poorly for others. However, there are hybridization types that have shown to be successful for many applications. In general, hybrid algorithms are able to improve the quality of the final solutions with less running time.

In this context, we propose one hybridization to the *Particle Swarm Optimization* (PSO) algorithm to solve four benchmark equality constraint problems. After generating a population of solutions with the EA, we enhance it with the *Variable Neighborhood Descent* (VND) algorithm.

The rest of the paper is organized as follows. In [section 2](#) we explain the proposed VND algorithm. In [section 3](#) shows how to build the PSO and hybridize it with the local search. The [section 4](#) holds the results obtained through tests, and we conclude our findings in [section 5](#).

2 Variable Neighborhood Search

Variable neighborhood search (VNS) is a meta-heuristic for solving combinatorial and global optimization problems whose basic idea is a systematic change of neighborhood within a local search [6]. It focuses on exploring a small region of the space of solutions, searching for local minima.

The VNS technique is based on a concept which states that a global minimum is a local minimum regarding to all possible neighborhood structures and a local minimum to one neighborhood structure is not necessarily a local minimum to another neighborhood structure. By exploiting a series

of neighborhood structures, it is expected to check attraction basins for the best minimum around the starting point. In this work, we use a variable neighborhood descent approach. In this version, we search the neighborhoods in a deterministic order, and no local search is performed in the newly explored points. A basic implementation is shown in Algorithm 1.

Algorithm 1 VND algorithm

```

procedure ( $t_k$  = number of neighborhoods,  $s$  = neighborhood size,  $x$ )
   $k \leftarrow 0$ 
  while  $k < t_k$  do
    Given a neighborhood  $N^k(x)$  of  $x$  such that,  $N^{k-1}(x) \subset N^k(x)$ .
     $x' \leftarrow Best(N^k(x))$ 
    if  $f(x') < f(x)$  then  $x \leftarrow x'$ ;  $k \leftarrow 0$ ; else  $k \leftarrow k + 1$ 
    end if
  end while
  Return  $x$ 
end procedure

```

The neighborhoods are constructed based on the bound value of the variables. Let U_i and L_i be the upper and lower limits of the i -th variable, respectively. Then we build the base range $r = \max_i U_i - \min_i L_i$, and we establish the increasing neighborhoods being the hypersphere centered in the best solution, with the radius of the k -th neighborhood's hypersphere calculated as:

$$rad = r / (10 * (kmax - k))$$

Where $kmax = round(1.5 * numberofvariables)$.

3 PSO with VND

In this section, we describe the Particle Swarm Optimization (PSO) applied to equality constraints problems. The PSO, first introduced in 1995 by Eberhart & Kennedy [3], is a population-based cooperative algorithm, and shares similar characteristics with a general genetic algorithm. However, the manner in which the two algorithms traverse the search space to handle optimization problems is fundamentally different.

The PSO algorithm has two primary operators: velocity update and position update. Here, we use the global version of PSO, as presented in Algorithm 2:

The PSO algorithm update the velocity and position of the particle accord-

Algorithm 2 PSO algorithm

procedure (PSO parameters)

Initialize particles with random positions and velocities

while no stop criterion **do**

Evaluate particles

Store *pbest* and *gbest*

Update velocity and position

Reflection operator

end while

Return best solution

end procedure

ing to [Equation 1](#) and [Equation 2](#), respectively:

$$v_i^{k,t} = w^t * v_i^{k,t-1} + c_1 \gamma_{1i} (pbest_i^k - x_i^{k,t-1}) + c_2 \gamma_{2i} (gbest_i^k - x_i^{k,t-1}) \quad (1)$$

wherein $v_i^{k,t}$ is i th component of the velocity of the k th particle, $x_i^{k,t}$ is the i th component of the position of the k th particle, in t th step of the algorithm; w is the inertia factor, which ranges from 0.9 to 0.4; c_1 and c_2 are the social parameters, which can be varied to make the particles have more tendency to go in the direction of *pbest* and *gbest*, in this work both were used equal to 2.05; γ_{1i} and γ_{2i} are random values between 0 and 1.

From the velocity and the previous position of the particle, its new position is calculated as follows:

$$x_i^{k,t} = x_i^{k,t-1} + v_i^{k,t} \quad (2)$$

In the case of one individual being out of the admissible range, the reflection operator is executed to force the individual back inside the feasible region. The reflection by the lower limit (x_L) and upper limit (x_U) is given respectively by $x_r = x_L + |x - x_L|$ and $x_r = x_U - |x_U - x|$, wherein x is the individual outside of the feasible region and x_r represents the reflected individual.

The proposed hybrid algorithm, is a real-PSO coupled with VND algorithm and aims to deal with nonlinear equality constraint using the VND as a local search operator in the real-PSO algorithm. This kind of approach is used by many global optimization procedures with the goal of refining the solution of the problem and improving the speed of convergence of the algorithm. The main advantage obtained with this hybrid approach is that the search space of solutions is reduced to a “subspace” of local optima. The local search VND operator will be run every 5 generations. The current best solution of the PSO will be used in the VND operator. Finally, the output of the VND operator will deterministically replace the worst point of the current population.

4 Experiments and Results

The numerical experiments to compare the performance of PSO and PSO-VND algorithms is presented. Moreover, we compare with a Broyden-Fletcher-Goldfarb-Shanno (BFGS) operator [5], in which an estimate of the inverse of the Hessian matrix is constructed, iteratively, to define a search direction. The experiments were run using a 3.5 GHz Intel Core i7 CPU with 32 GB RAM and the implementation in MatLab. The following constraints problems [1] were used in analytical tests:

Problem 1:

$$\begin{aligned} \min f(x) &= x_1^2 + (x_2 - 1)^2 \\ \text{subject to: } &\begin{cases} x_2 - x_1^2 = 0 \\ -1 \leq x_i \leq 1, \quad i = 1, 2 \end{cases} \end{aligned} \quad (3)$$

The global optimum is at $x^* = (\pm 1/\sqrt{2}, 1/2)$ wherein $f(x^*) = 0.75$.

Problem 2:

$$\begin{aligned} \min f(x) &= e^{x_1 x_2 x_3 x_4 x_5} \\ \text{subject to: } &\begin{cases} x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ x_3 x_2 - 5 x_4 x_5 = 0 \\ x_1^3 + x_2^3 + 1 = 0 \\ -2.3 \leq x_i \leq 2.3, \quad i = 1, 2; -3.2 \leq x_i \leq 3.2, \quad i = 3, 4, 5 \end{cases} \end{aligned} \quad (4)$$

The global optimum is $x^* = (-1.7171, 1.5957, 1.8272, -0.76364, -0.7636)$ wherein $f(x^*) = 0.0539498$.

Problem 3:

$$\begin{aligned} \min f(x) &= 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1 x_2 - x_1 x_3 \\ \text{subject to: } &\begin{cases} x_1^2 + x_2^2 + x_3^2 - 25 = 0 \\ 8x_1 + 14x_2 + 7x_3 - 56 = 0 \\ 0 \leq x_i \leq 10, \quad i = 1, 2, 3 \end{cases} \end{aligned} \quad (5)$$

The global optimum is at $x^* = (3.512, 0.217, 3.552)$ wherein $f(x^*) = 961.715$.

Problem 4:

$$\begin{aligned} \max f(x) &= (\sqrt{n})^n \prod_{i=1}^n x_i \\ \text{subject to: } &\begin{cases} \sum_{i=1}^n x_i^2 - 1 = 0 \\ 0 \leq x_i \leq 1, \quad i = 1, 2, \dots, n \end{cases} \end{aligned} \quad (6)$$

wherein $n = 3$. The global maximum is at $x_i^* = 1/\sqrt{n}$, with $f(x^*) = 1$.

The PSO handles equality constraints by rewriting them as inequality constraints of the form $|h_j| \leq \epsilon$, where ϵ is called the tolerance. Thus, each equality constraint $h_j = 0$, were replaced by two inequality constraints: $h_{j_1}(x) = -h_j(x) - \epsilon \leq 0$ and $h_{j_2}(x) = h_j(x) - \epsilon \leq 0$, with $\epsilon = 1\text{E-}06$. If any inequality constraint h_{j_i} is violated then the objective function is penalized using $f = f + 10 \times h_{j_i}(x)$ if it is a minimization problem or $f = f - 10 \times h_{j_i}(x)$ if it is a maximization one.

Each algorithm was executed 30 times for each problem with the same parameters for both versions. The usual parameters were used following the ones presented in [section 3](#). The population size equals 100 and the maximum number of generations (only stop criterion) equals 100. For VND parameters, the number of neighborhoods was set to double the number of variables of the test instances, and neighborhood size was 50. At the end of 30 executions of algorithms, we obtained the mean value of the best individual throughout the 100 generations and a corresponding boxplot for all the runs. [Figure 1](#) show the mean value of the best individual throughout the generations. The x-axes represent the generation and the y-axes represent the mean objective function value of the best individual throughout the generations. It is worthwhile to notice that, in all problems, the PSO-VND hybrid shows a higher speed convergence tendency when compared with the PSO.

Observing [Figure 1](#) (c) and (d), it is possible to see that the PSO-VND presents a better final solution when compared with PSO and PSO-BFGS. Taking into consideration the boxplot of the final generation for all algorithms, there is no superposition of the boxes meaning that the final solution reached by the PSO-VND is also statistically different from the other solutions. Since the final boxplots for each algorithm present a superposition of the boxes, it is not possible to state that there is a difference amongst the final solution without a statistical test. However, it is possible to observe that around the 30th generation, the PSO-VND hybrid found a solution which is statistically better than the final solution found by the PSO.

[Table 1](#) shows the run time of PSO, PSO-BFGS and PSO-VND for each problem considering the maximum number of generation as the only stopping criterion. As expected, the run time for the hybrid algorithms, PSO-VND and PSO-BFGS, are higher when compared with the basic PSO. Moreover, comparing PSO-VND and PSO-BFGS, the run time of PSO-VND is smaller than PSO-BFGS. Despite the fact the run time of the PSO-VND being higher than the basic version, since PSO-VND requires less generation to find a good final solution, the run time using any other stopping criterion (such as stabilization of the best solution) would not impose a computational burden

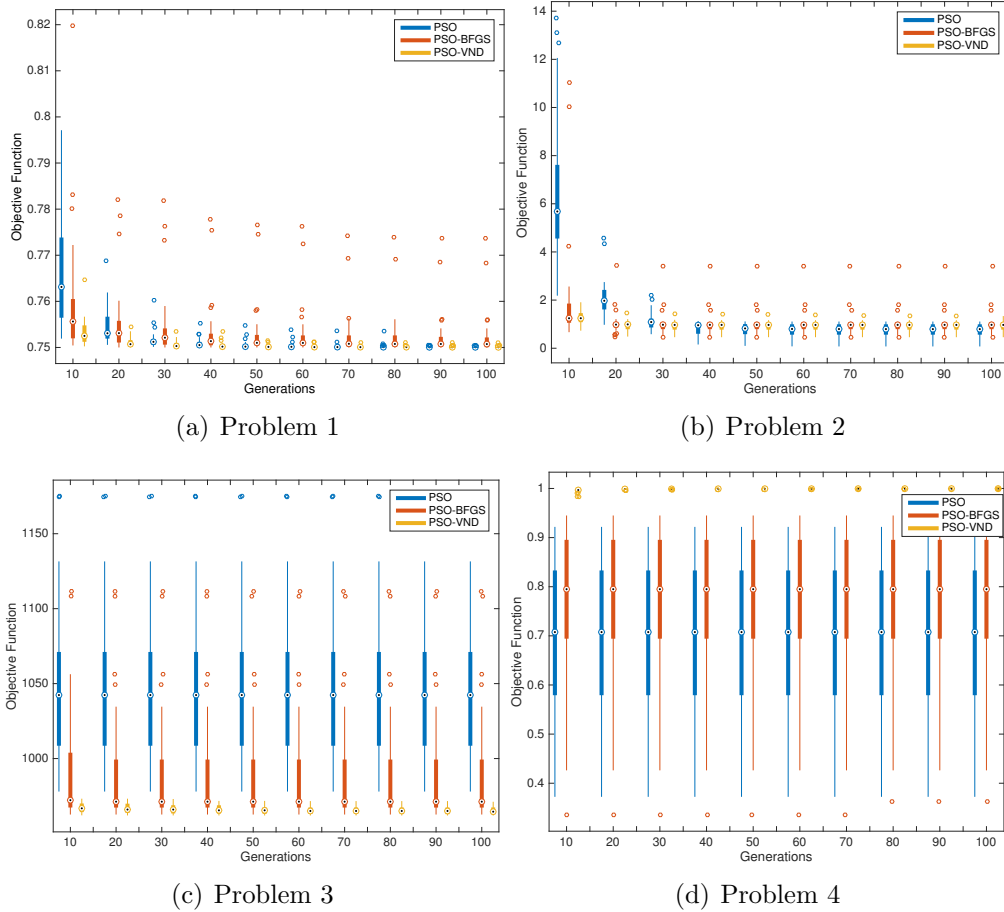


Figure 1. The mean value of the best individual throughout the 100 generations, in 30 runs. At every 10 generations, the boxplot of the best solutions found is shown.

for the PSN-VND.

Table 1
Run time of the algorithms for each problem test.

Run time (s)	PSO				PSO-BFGS				PSO-VND			
	P1	P2	P3	P4	P1	P2	P3	P4	P1	P2	P3	P4
mean	1.79	1.99	1.86	2.12	2.75	3.19	2.86	4.05	2.08	2.61	2.27	2.66
std	0.20	0.21	0.28	0.16	0.29	0.26	0.20	0.77	0.18	0.19	0.18	0.27
min	1.50	1.52	1.41	1.54	2.44	2.70	2.58	3.09	1.80	2.20	2.06	2.31
max	2.38	2.44	3.13	2.43	3.61	3.98	3.49	6.51	2.55	3.11	2.99	3.58

5 Conclusion

This work presents a VND-based local search operator to handle equality constraint problems in PSO. The hybrid algorithm was compared with a basic version of PSO and PSO coupled with a BFGS-based local search operator. Results show the usage of VND operator enhance the convergence properties of the PSO. Furthermore, PSO-VND hybrid algorithm converges to better solutions, requiring fewer generations runs, and as a consequence, compensating more function evaluations. The results on four benchmark problems provide evidence that VND operator handles equality constraints problems very well.

Acknowledgements

The authors thank UFOP and CNPq, Fapemig and CAPES.

References

- [1] Arturo Hernández Aguirre, Angel E. Muñoz Zavala, S. Botello Rionda, and E. Villa Diharce. Copso: Constrained optimization via pso algorithm. Technical report, CC/CIMAT, 2007.
- [2] Christian Blum, Jakob Puchinger, Günther R. Raidl, and Andrea Roli. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, 11(6):4135–4151, 2011.
- [3] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43, 1995.
- [4] C. H. Fonseca and E. F. Wanner. A quadratic approximation-based local search operator for handling two equality constraints in continuous optimization problems. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 4911–4917, 2016.
- [5] David G. Luenberger. *Linear and Nonlinear Programming*. Springer, 2003.
- [6] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.
- [7] Elizabeth F Wanner, Frederico G Guimaraes, Ricardo HC Takahashi, and Peter J Fleming. Local search with quadratic approximation in genetic algorithms for expensive optimization problems. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 677–683, 2007.