A Study of VEPSO Approaches for Multiobjective Real World Applications

Omar Andres Carmona Cortes*, Andrew Rau-Chaplin†, Duane Wilson† and Jürgen Gaiser-Porter‡
*Informatics Academic Department
Instituto Federal de Educação, Ciência e Tecnologia do Maranhão
São Luis, MA, Brazil
omar@ifma.edu.br
†Risk Analytics Lab
Dalhousie University
Halifax, NS, Canada
arc@cs.dal.ca, dwilson@gmail.com
‡Global Analytics
Willis Group
London, UK
gaiserporterj@willis.com

***Abstract*—The purpose of this paper is to evaluate the performance of two approaches based on Vector Evaluated Particle Swarm Optimization (VEPSO) algorithm in two real world applications, which are the environmental economic dispatch problem and the optimization of a reinsurance contract portfolio. The two tested algorithms are the canonical VEPSO and a new version called VEPSO-N, where in the last one the global updating on each swarm is done based on the archive. The performance is evaluated using the following metrics: hypervolume, number of solutions and coverage, showing that both approaches can present good outcomes.**

Keywords–Real World Applications; Reinsurance Contract Optimization; Environmental Economic Dispatch; Vector Evaluated PSO; Multiobjective.

I. INTRODUCTION

Real world applications involve solving problems whose objectives are normally in conflict. For example, in an oil-based power plant, the lower the cost of generating energy, the bigger the emission of pollutants. Thus, companies have to figure out the best trade-off due to mainly environmental regulations. In a financial investment, the bigger the risk, the bigger the return. Therefore, investors are interested in smaller risks and to obtain bigger returns.

Those kind of problems are called Multiobjective Optimization Problems (MOPs) and their solution lay in the concept of Pareto Optimality, where solutions are characterized as a set of trade-off points. Gradient-based optimization techniques [1] [2] can be used to detect Pareto optimal solutions [3]; however, to do so the objectives have to be aggregated in a single objective function, and only one solution can be found per run [4], adding heavy computational cost to the whole process. In the same sense, traditional mono-objective evolutionary algorithms present the same kind of problem, *i.e.*, only one solution can be computed at the time.

Taking the computational cost into account, swarm/evolutionary multiobjective algorithms (MOEA) represent a viable alternative to solve MOPs. Indeed, MOEAs have been recognized to be well-suited for this kind of application because of their abilities to explore multiple solutions in parallel and to find a widespread set of non-dominated solutions in a single run [5]. Among the available MOEAs based on PSO is the VEPSO [6], which was based on Vector Evaluated Genetic Algorithm (VEGA) [7]. The idea behind VEPSO is to evolve two separated swarms, one per evaluation function, where considering two swarms, the direction of a swarm is guided by the best solution (g_{best}) found in the other one.

VEPSO has been successfully used in application ranging from the optimization of radiometry array antenna [8], passing through the optimization of a steady-state performance of a power system [9], to the optimization of the energy communications for heterogeneous network through cognitive sensing [10]. Its popularity can be mainly attributed to two reasons: it is easy to implement and the parallelization can be done in a straight-full way [4]. However, VEPSO might stagnate in bad approximation of the Pareto front as we can see in Matthysen et al. [11] and Lim et al. [12], where the authors show an analysis of VEPSO and VEGA, and explain why the canonical VEPSO tends to get trapped in sub-optimal Pareto frontiers.

In this context, a modification has been proposed by Lim et. al [13] in order to overcome problems with stagnation, where the best solution of a particular swarm is updated using a point stored in the archive, which contains only non-dominated solutions, rather than the g_{best} from the second swarm as in the canonical VEPSO algorithm. The main advantage of this approach is to keep the same number of callings to the evaluation function, therefore the comparison can be done in terms of iterations.

Two applications are considered in this paper. The first one is the Environmental Economic Dispatch problem (EED), where we want to determine the lower cost of generate energy using six generators and producing a smaller amount

 ISBN: 978-1-61208-358-2

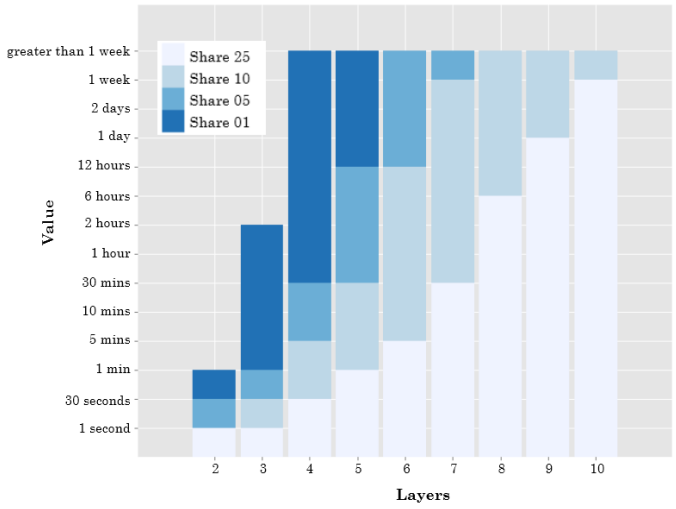


Figure 1. Estimated time for solving the RCO problem in R with different levels of discretization

of pollutants at the same time. The second application is the optimization of a Reinsurance Contract Portfolio (RCP) where, from the insurance company perspective, we want to hedge more risk and receive back more money in case of massive claims.

In terms of EED application, Qu's work [14] proposes a multiobjective fast-evolutionary programming for dealing with the problem; however, the Pareto frontiers are compared only visually, *i.e.*, he did not use proper metrics for comparison. Abedinia's work [15] deals with six and fourteen generators but the evaluation is done based only on the best results also without a proper metric. Farhat [16] did the same thing as previous works considering 3 different emission functions.

Regarding RCP, the first two works to address this kind of problem were [17] and [18]. Even though those two papers rely on mono-objective functions, they proved that it is worth to use evolutionary computation in this kind of application because it saves a considerable amount of time if compared with the enumeration method. Figure 1 shows the time required to solve the problem using the enumeration method, for instance, considering a 5% of discretization the enumeration method demands much more than a week to execute.

The remainder of this paper is organized as follows: Section II outlines the fundamentals of multiobjective problems; Section III presents the PSO-based algorithms, including VEPSO and the improved version called VEPSO-N; Section IV-A shows how the performance evaluation is done, the real world applications considered in this work and the simulation results; finally, Section V presents the conclusions and future works.

II. Multiobjective Fundamentals

A multiobjective optimization problem has to deal with two or more conflicting objective function [5] at the same time. These functions must be in conflict in order to build a Pareto frontier, where there are no solutions better than others; otherwise, the answer to the problem would be only one point in the search space.

Thus, assuming that a solution to a MOP is a vector in a search space X with m elements. A function $f : X \rightarrow Y$ evaluates the quality of a solutions mapping it into an objective space. Therefore, a multi-objective problem is defined as presented in (1), where f is a vector of objective functions, m is the dimension of the problem and n the number of objective functions.

$$Max\ y = f(x) = (f_1(x_1, ..., x_m), ..., f_n(x_1, ..., x_m)) \quad (1)$$

In order to determine whether a solution belongs to the Pareto frontier or not, we need the concept of optimality, which state that given two vectors $x, x* \in \Re$ and $x \neq x*$, x dominates $x*$ (denoted by $x \succeq x*$) if $f_i(x)$ is not worse than $f_i(x*), \forall\ i$ and $\exists$ at least one i where $f_i(x) > f_i(x*)$ in maximization cases and $f_i(x) < f_i(x*)$ otherwise. Hence, a solution x is said Pareto optimal if there is no solution that dominates x, in such case, x is called non-dominated solution. Mathematically, assuming a set of non-dominated solutions $\wp$, a Pareto frontier(pf) is represented as $pf = \{f_i(x) \in \Re | x \in \wp\}$

III. PSO-Based Algorithms

The particle swarm optimization was proposed by Kennedy and Eberhart [19] in 1995. The algorithm consists of particles which are placed into a search space, and move itself combining its own history position and the global optimal solution found so far. A particle position is represented in the search space as $X_i^D = (x_i^1, x_i^2, ..., x_i^D)$ and it is updated based on its velocity $V_i^D = (v_i^1, v_i^2, ..., v_i^D)$, where D represents the problem dimension. The new position is determined by means of (2) and (3), where w represents the inertia weight, c_l and c_2 are acceleration constants, r_l and r_2 are random number in the range $[0, 1]$, p_i^d is the best position reached by the particle P, and g^d is a vector which stores the global optima of the swarm.

$$v_i^d = w \times v_i^d + c_1 r_1 \times (p_i^d - x_i^d) + c_2 r_2 \times (g^d - x_i^d) \quad (2)$$

$$x_i^d = x_i^d + v_i^d \quad (3)$$

The Algorithm 1 outlines how PSO works. Initially, the swarm is created at random, where each particle has to be within the domain $[a_i^d, b_i^d]$. Then particles are evaluated in order to initialize the P matrix and the g^d vector, which are the best experience of each particle and the best solution that has been found so far, respectively. Thereafter, the velocity and the position of a particle are updated within a loop that obeys some stop criteria. In the pseudo code presented in the Algorithm 1, the stop criteria is a certain number of iterations.

A. VEPSO

The VEPSO algorithm is a multiobjective heuristic based on Vector Evaluated Genetic Algorithms (VEGA) [7]. The main idea behind this algorithm is to "evolve" two independent swarms and exchange information between them, *i.e.*, assuming two swarms S_1 and S_2, and two functions to be

Generate a swarm of particles **X** of size *s* from $[a_i^d, b_i^d]$;
for *i = 1* to *swarm_size* **do**
  Evaluate swarm
  Update the best position g
  Update p of the particles
  for *j = 1 to D* **do**
    Update velocity V using (2)
    Update position X using (3)
  end
end
Verify if the current g is better than the best of the current swarm

Algorithm 1: Particle Swarm Optimization (PSO)

optimized f_1 and f_2, being solved by S_1 and S_2, respectively. The swarm S_1 updates its velocity using the best particle of S_2 (g_2). On the other hand, S_2 updates its velocity based on the best particle of S_1 (g_1). Then, an archive is held after each iteration joining both swarms in order to obtain only the non-dominated solutions.

B. VEPSO-N

The VEPSO-N uses the idea behind Lim's work [13] where an archive with non-dominated solutions is maintained, then each swarm updates its own global best (g_{best}) using the best result in its respective function. Mathematically, considering two swarms S_1 and S_2, and $A = \{a_1, a_2, ..., a_n\}$ as being the set of non-dominated solutions, if $f(a_i)$ presents the best solutions regarded to S_1 and $f(a_j)$ shows the best solutions regarded to S_2, then a_i and a_j replace $g_{best}^{S_1}$ and $g_{best}^{S_2}$, respectively. In other words, we use the best solutions in the archive for updating the global best on each swarm.

IV. Experiments

A. Performance Evaluation and Parameters

In this section, we discuss the experimental evaluation of EED and RCP optimization problems as follows. Firstly, the number of non-dominated points found in the Pareto frontier after 31 trials were determined. Secondly, the hypervolume, which is the volume of the dominated portion of the objective space, as presented in (4), was measured, where for each solution $i \in Q$ a hypercube v_i is constructed. Having computed each v_i, we can calculate the final hypervolume by the union of all of them.

$$hv = volume(\bigcup_{i=1}^{|Q|} v_i) \tag{4}$$

Thirdly, the dominance relationship between Pareto frontiers (coverage) obtained with different algorithms was calculated as depicted in (5). Roughly speaking, the coverage is the ratio between the number of solutions dominated by A divided by the number of elements from set B [27]. If $C(A, B) = 1$ then all solutions in A dominate B. Therefore, $C(A, B) = 0$ means the opposite.

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \preceq b\}|}{|B|} \tag{5}$$

The parameters used in all experiments for PSO algorithms were: $c_1 = c_2 = 0.5 + log(2)$; numbers of particles = 50; w_0 = 0.9; w_f = 0.1, where w has a linear updating based on (6) as proposed by Nikabadi and Ebadzadeh [22], where w_0 is the initial weight, w_f is the final one, N_G is the number of iterations and i depicts the current generations. Moreover, all study cases are executed using 500, 1000 and 2000 iterations; the initialization was done as recommended by Clerc [23]; and, we are not narrowing the archive size.

$$w = (w_0 - w_f) \times \frac{N_G}{i} \tag{6}$$

All the parameters were chosen empirically and all tests have been conducted using R version 2.15.0 [24] and RStudio [25] on a Windows 7 64-bit Operating System running on an Intel i7 3.4 Ghz processor, with 16 GB of RAM.

B. Real World Applications

1) Environmental Economic Dispatch: The environmental economic dispatch involves the optimization of both fuel cost and pollution emission simultaneously as presented in (7) and (8), where P_i is the power used on the i^{th} generator, a_i, b_i, c_i, α_i, β_i and γ_i are coefficients presented in Tables I and II.

$$min\ Fc = \sum_{i=1}^{n} (a_i P_i^2 + b_i P_i + c_i) \tag{7}$$

$$min\ E = \sum_{i=1}^{n} (\alpha_i P_i^2 + \beta_i P_i + \gamma_i) \tag{8}$$

subject to

$$\sum_{min}^{max} P_i \geq P_d \tag{9}$$

$$P_{min} \leq P_i \leq P_{max} \tag{10}$$

The constraint presented in (9) represents the required demand, *i.e.*, the sum of all powers has to be equal or greater than a specific demand, and the constraint shown in (10) depicts the operation boundaries of each generator which are also presented in Tables I and II. The coefficients and boundaries where obtained from Singh and Kumar [26], and, at this stage, we are not considering the power loss. Moreover, we are taking into account demands equals to 500 MW and 700 MW.

TABLE I. Generators and cost coefficients

P_{min}	P_{max}	a	b	c
5	50	0.01	2	10
5	60	0.012	1.5	10
5	100	0.004	1.8	20
5	120	0.006	1	10
5	100	0.004	1.8	20
5	60	0.01	1.5	10

Table III shows the average results in terms of number of solutions and hypervolume for a EED problem comprises of 6

TABLE II. GENERATORS AND EMISSION COEFFICIENTS

P_{min}	P_{max}	α	β	γ
5	50	0.00419	0.32767	13.85932
5	60	0.00419	0.32767	13.85932
5	100	0.00683	-0.54551 4	0.26690
5	120	0.00683	-0.54551	40.26690
5	100	0.00461	-0.51116	42.89553
5	60	0.00461	-0.51116	42.89553

generators and a demand of 500 MW, following the legend I for VEPSO and II for VEPSO-N. As expected, increasing the number of iterations the number of non-dominated solutions also increases in both algorithms. We can also observe the number of solutions in VEPSO-N is smaller than in VEPSO. On the other hand, the average hypervolume is better in VEPSO-N, showing that the Pareto frontier might be better on the respective algorithm.

TABLE III. NUMBER OF SOLUTIONS AND HYPERVOLUME FOR EED PROBLEM CONSIDERING A DEMAND OF 500 MW

	500 it		1000 it		2000 it	
	I	II	I	II	I	II
#NS	67.64	57.61	177.13	62.93	399.42	74.71
HV	1.8e6	1.96e6	1.91e6	2.46e6	1.74e6	2.09e6

Table IV shows the coverage metrics in the final Pareto frontier, which indicate that VEPSO presents better solutions, particularly with 2000 iterations, where VEPSO dominates 74% of the solutions. Figure 2 shows the final Pareto frontier after 31 trials, where we can see that VEPSO tends to find out a better Pareto frontier. When 2000 iterations are used, VEPSO-N presents an extension of the Pareto frontier; even though, those points are not interesting because they are dominated points. Also, we have to point out that VEPSO-N tends to concentrate its solutions in the beginning of the Pareto frontier when 500 iterations are used, which is not good in terms of diversity.

TABLE IV. COVERAGE FOR EED PROBLEM WITH DEMAND OF 500 MW

	500 it		1000 it		2000 it	
	I	II	I	II	I	II
I	-	0.25	-	0.28	-	0.74
II	0.07	-	0.23	-	0.25	-

Table V presents the average results in terms of number of solutions and hypervolume for a EED problem comprises of 6 generators and a demand of 700 MW, following the legend I for VEPSO and II for VEPSO-N. The problem is harder to be solved because the demand constraint is stronger. Nonetheless, the number of solutions increase as we increase the number of iterations. However, this increment is clearly bigger in VEPSO than in VEPSO-N. On the other hand, the hypervolume in VEPSO-N is better indicating that its solutions might be slightly better. Figure 3 shows the final Pareto frontier after 31 trials. Visually, it seems that the final frontier is better for VEPSO which is confirmed in Table VI, excepting for 500 iterations where VEPSO-N dominates 36% of the solutions, whereas VEPSO dominates 29%.

Summarizing, VEPSO seems to cover better the final Pareto frontier than VEPSO-N for demands of 500 and 700 MW. On the other hand, when the demand constraint is harder (700 MW), VEPSO-N presented best results with lower number of iterations.

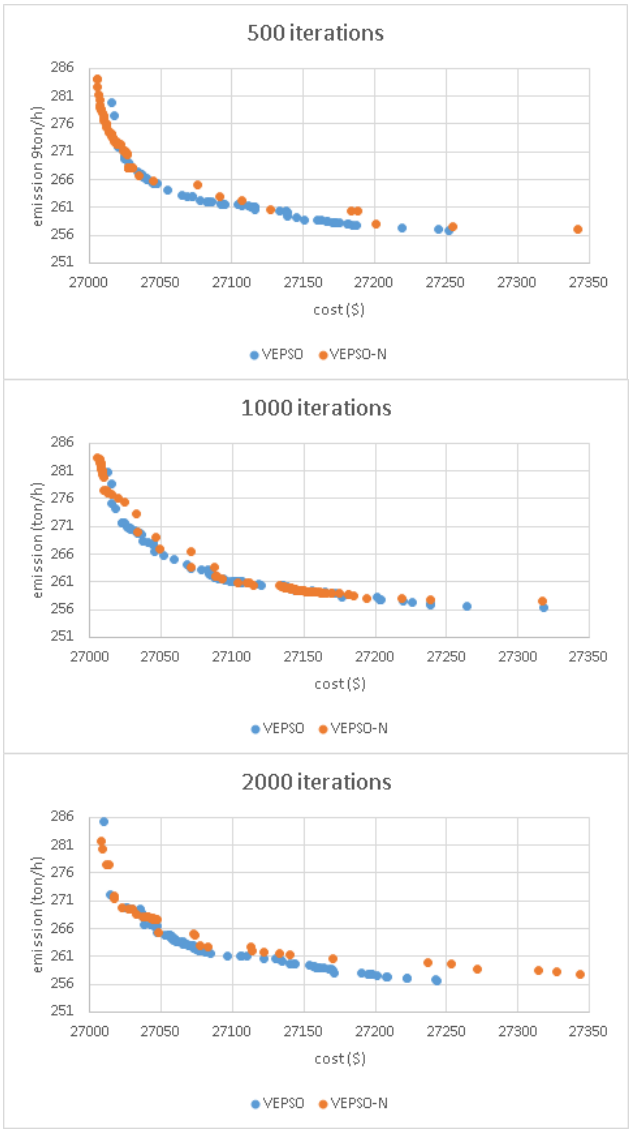


Figure 2. Pareto frontier after 31 executions for EED and demand of 500

TABLE V. NUMBER OF SOLUTIONS AND HYPERVOLUME FOR EED PROBLEM CONSIDERING A DEMAND OF 700 MW

	500 it		1000 it		2000 it	
	I	II	I	II	I	II
#NS	141.9	93.6	323.81	103.94	546.58	137.42
HV	4.8e6	4.9e6	4.8e6	5.3e6	5.3e6	5.4e6

TABLE VI. COVERAGE FOR EED PROBLEM WITH DEMAND OF 700 MW

	500 it		1000 it		2000 it	
	I	II	I	II	I	II
I	-	0.29	-	0.35	-	0.37
II	0.36	-	0.30	-	0.13	-

2) Reinsurance Contract Optimization: The reinsurance process consists of hedging risk from the insurance company to a bigger one, called reinsurance company. The main purpose of doing so is to survive in case of massive claims mainly caused by natural catastrophes. The reinsurance contract optimization problem consists of given a treaty structure to figure out the best combination of placements or shares in order to transfer

 ISBN: 978-1-61208-358-2

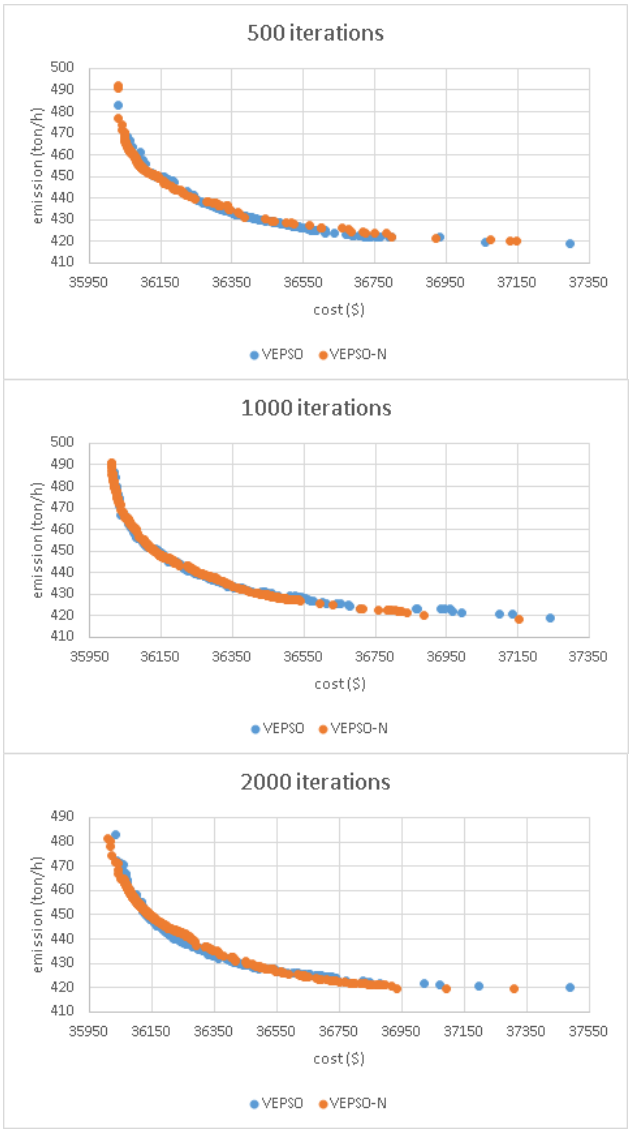


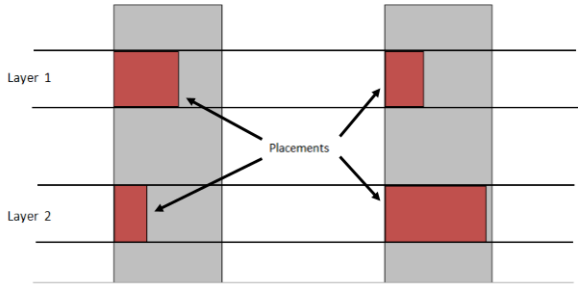
Figure 3. Pareto frontier after 31 executions for EED and demand of 700

the maximum of risk, and at the same time, to receive the maximum return when facing massive claims. Therefore, the main purpose of a RCO problem is to find out the best combination of shares or placements which maximize both the transfered risk and the expected return. Figure 4 is an example of a structure with two different solutions in terms of shares.

The Equation 11 represents the RCO in terms of an optimization problem, where VaR is a risk metric, $\mathbf{R}$ is a function based on a combination of shares pi, and E is the expected value. For further details about the problem refer to [17] and [20].

$$\begin{array}{ll} maximize & f_1(x) = VaR_\alpha(\mathbf{R}(\pi)) \\ maximize & f_2(x) = E[\mathbf{R}(\pi)] \end{array} \quad (11)$$

Table VII shows the average number of solutions and the average hypervolume (all hypervolume values are multiplied by 1×10^{15}) for the RCO problem consisting of 7 layers of real anonymized data and a discretization of 5% obtained by



Solution	Layer	Limit	Deductible	Placement	Max Recovery	Premium
1	Layer 1	200	300	60%	120	$40
	Layer 2	200	100	20%	40	$40
2	Layer 1	200	300	30%	60	$25
	Layer 2	200	100	95%	190	$100

Figure 4. Structure and two solutions with different placements

rounding solutions. The following legends are I for VEPSO and II for VEPSO-N. Again, the number of solutions increase as we increase the number of iterations for both approaches; however, in this particular experiment, VEPSO-N presented a greater number of solutions for 500 and 2000 iterations, nonetheless the average hypervolumes were smaller, indicating that VEPSO could present better solutions.

TABLE VII. NUMBER OF SOLUTIONS AND HYPERVOLUME FOR RCO PROBLEM WITH 7 LAYERS

	500 it		1000 it		2000 it	
	I	II	I	II	I	II
#NS	1394	2152.52	4677.13	4360.06	11650	16360.65
HV	2.26	2.21	2.26	2.2	2.26	2.26

Figure 5 depicts the final Pareto frontier for RCO problem using 7 layers for 500, 1000 and 2000 iterations, respectively. Visually, solutions seem to be very similar, perhaps with a little advantage to VEPSO, specially when 2000 iterations are taking into account. Thus, Table VIII presents the coverage metrics on the final Pareto frontier, where VEPSO dominates 4% more solutions with 500 iterations, and gets worse with 1000 iterations dominating 6% less solutions. This scenario is confirmed with 2000 iterations where VEPSO-N could dominate even more points, *i.e.*, 30%.

TABLE VIII. COVERAGE FOR RCO PROBLEM WITH 7 LAYERS

	500 it		1000 it		2000 it	
	I	II	I	II	I	II
I	-	0.31	-	0.21	-	0.17
II	0.27	-	0.27	-	0.30	-

In order to increase the difficulty of solving this optimization problem, 8 layers were synthetically added to the previous 7 layers structure. Table IX shows the results in terms of the average number of solutions and average hypervolume (all hypervolume values are multiplied by 1×10^{15}) when 15 layers are considered. Again, the number of solutions increase as the number of iterations; however, in this particular case, VEPSO and VEPSO-N presented similar results, excepting for 2000 iterations where VEPSO reached more non-dominated points. On the other hand, VEPSO presented better average hypervolume, indicating better solutions.

Figure 6 represents the final Pareto frontier for RCO problem using 15 layers for 500, 1000 and 2000 iterations,

 ISBN: 978-1-61208-358-2

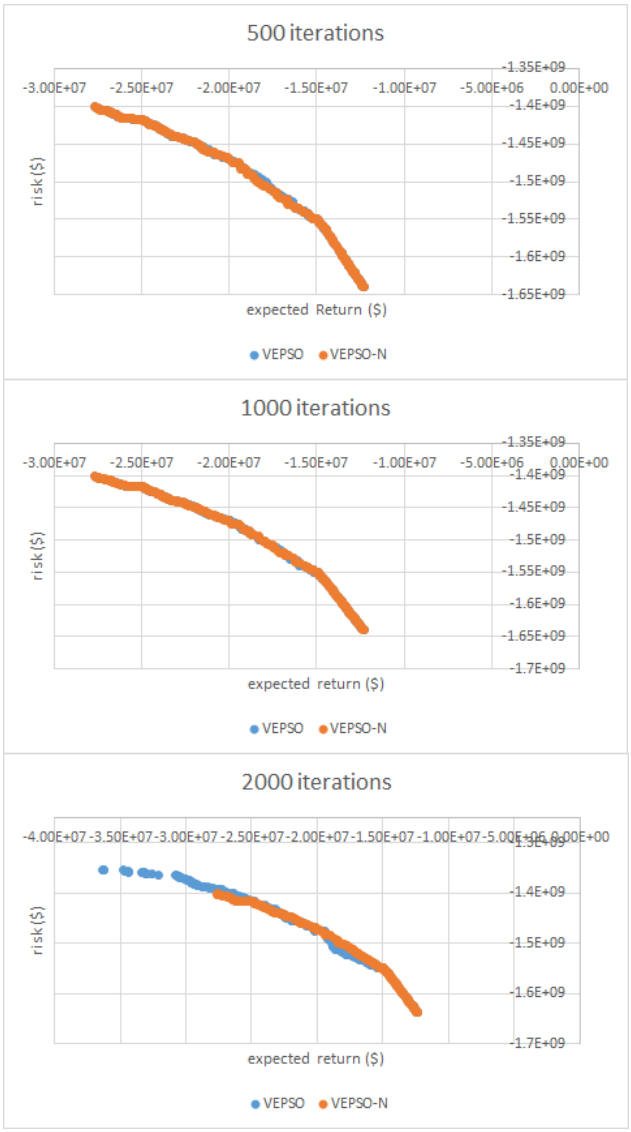


Figure 5. Reinsurance contract optimization Pareto frontier after 31 executions using 7 layers

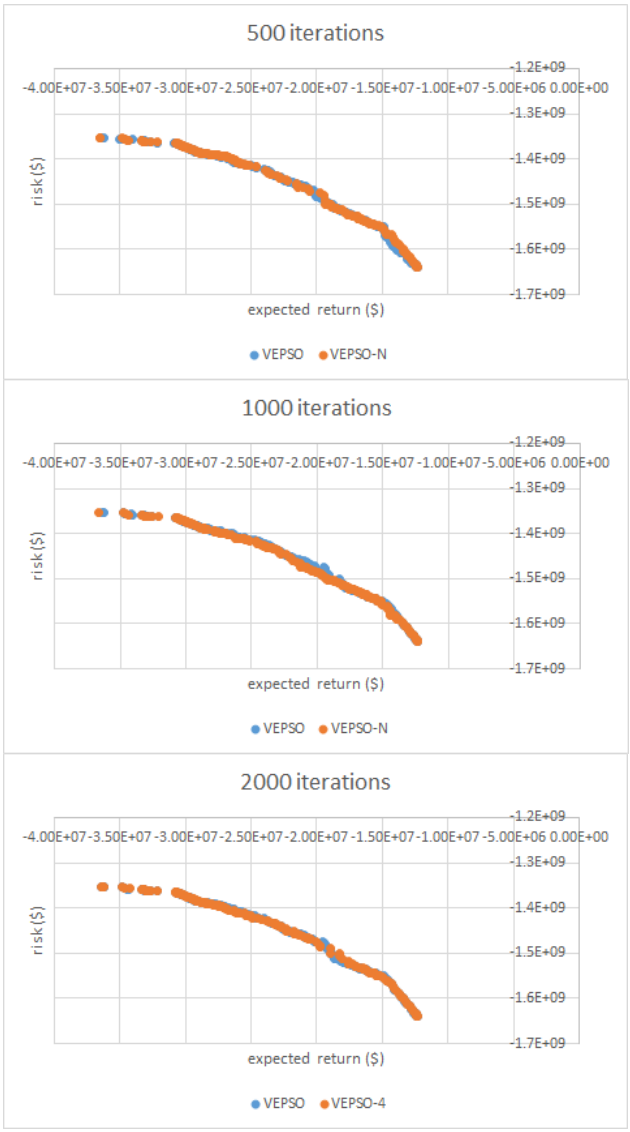


Figure 6. Reinsurance contract optimization Pareto frontier after 31 executions using 15 layers

TABLE IX. NUMBER OF SOLUTIONS AND HYPERVOLUME FOR RCO PROBLEM WITH 15 LAYERS

	500 it		1000 it		2000 it	
	I	II	I	II	I	II
#NS	186.68	453.9	1001.97	2214.1	3916.8	3299.7
HV	4.57	4.1	4.6	4.1	4.55	4.2

TABLE X. COVERAGE FOR RCO PROBLEM WITH 15 LAYERS

	500 it		1000 it		2000 it	
	I	II	I	II	I	II
I	-	0.23	-	0.56	-	0.44
II	0.55	-	0.22	-	0.33	-

respectively. Solving the problem with more layers is clearly more difficult. Nonetheless, both algorithms presents visually similar results. Thus, Table X shows the coverage for the algorithms where we can see that differences in this particular case are evident in the following cases: (i) VEPSO-N with 500 iterations dominating 55% of VEPSO solutions; (ii) VEPSO with 1000 iterations dominating 56% of VEPSO-N points; and, VEPSO with 2000 iterations dominating 44% of VEPSO against 33% in VEPSO-N.

V. CONCLUSION

This paper presented a study about the performance of the canonical VEPSO and the algorithm called VEPSO-N in solving multiobjective world real problems. The comparison shows that both approaches are suitable for finding out non-dominated points; however, the traditional VEPSO has an advantage in the EED problems, since present more solutions in the search space and tends to dominate more points as well, for example, using 2000 iterations VEPSO domains 70% of the solutions from VEPSON-N with a demand of 500 MW against 25% in the way around. Further, when a demand of 700 MW is considered, VEPSO domains 37% of the solutions from VEPSO-N using 2000 iterations.

 ISBN: 978-1-61208-358-2

In RCO, VEPSO-N tends to present more solutions than VEPSO with less layers, and dominates up to 30% of the solutions from VEPSO. This behavior is similar when solving the problem with 15 layers using 500 and 1000 iterations; nevertheless, VEPSO-N is overcame when 2000 iterations are used, since VEPSO starts to domain 44% of points against %33 of VPSON-N.

Future work includes a comparison against other modern approaches, such as Vector Evaluated Differential Evolution (VEDE) [28], Strength Pareto Evolutionary Algorithm (SPEA2) [29] and Multiobjective Evolutionary Algorithm/Distributed (MOEA/D) [27], hybridization of VEPSO with other metaheuristics and a parallel version.

ACKNOWLEDGMENT

This research was financially supported by Flagstone Re, Halifax, Canada and by the Science without Border program of CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brazil), and Instituto Federal de Educação, Ciência e Tecnologia do Maranhão. We also would like to thank the Willis group for providing the anonymized real world data for the RCO problem.

REFERENCES

[1] Y. Bengio, "Gradient-Based Optimization of Hyperparameters", Neural Computation , vol. 12, no. 8, 2000, pp. 1889–1900.

[2] R. Haupt, "Comparison Between Genetic and Gradient-Based Optimization Algorithms for Solving Electromagnetics Problems", IEEE Transaction on Magnetics, vol. 31, no. 03, 1995, pp. 1932–1935.

[3] D. A. Iancu and N. Trichakis, "Pareto Efficiency in Robust Optimization", Articles in Advance Management Science, 2013, pp. 1–18.

[4] K. E. Parsopoulos, D. K. Tasoulis, M. N. Vrahatis, and K. Words, "Multiobjective Optimization Using Parallel Vector Evaluated Particle Swarm Optimization", In Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, pp. 823–828, ACTA Press, 2004.

[5] K. Deb, "Multi-objective Optimization using Evolutionary Algorithms", John Wiley and Sons LTDA, 2001.

[6] K. E. Parsopoulos and M. N. Vrahatis, "Particle Swarm Optimization Method in Multiobjective Problems", In Proceedings of the 2002 ACM Symposium on Applied Computing, pp. 603–607, ACM Press, 2002.

[7] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms", in Proceedings of the 1st International Conference on Genetic Algorithms, pp. 93–100, L. Erlbaum Associates Inc. Hillsdale, NJ, USA, 1985.

[8] D. Glies and Y. Rahmat-Samii, "Vector evaluated particle swarm optimization (VEPSO): Optimization of a radiometer array antenna", in Proc. of the IEEE International Symposium on Antennas and Propagation,vol. 3, 2004, pp. 2297-2300.

[9] J. G. Vlachogiannis and K. Y. Lee, "Multi-objective based on parallel vector evaluated particle swarm optimization for optimal steady-state performance of power systems", Expert Systems with Applications, vol. 36, no. 8, 2009, pp. 802-808.

[10] S. Hou, X. Zhang, H. Zheng, L. Zhao, and W. Fang, "An effective interference management framework to achieve energy-efficient communications for heterogeneous network through cognitive sensing", International ICST Conference on Communications and Networking in China (CHINACOM), pp. 536,541, 2012.

[11] W. Matthysen, A. P. Engelbrecht, and K. M. Malan, "Analysis of stagnation behavior of vector evaluated particle swarm optimization", IEEE Symposium on Swarm Intelligence (SIS), pp. 155,163, 2013

[12] K. S. Lim et al., "Convergence and diversity evaluation for vector evaluated Particle Swarm Optimization", Proceedings of International Conference on Modelling, Identification & Control (ICMIC), pp. 280-285, 2013.

[13] K. S. Lim et al., "Improving Vector Evaluated Particle Swarm Optimisation by Incorporating Nondominated Solutions", The Scientific World Journal, vol. 2013, Article ID 510763, 2013.

[14] B.-Y. Qu, P. N. Suganthan, V. R. Pandi, and K. B. Panigrahi, "Multi objective evolutionary programming to solve environmental economic dispatch problem", International Conference on Control Automation Robotics & Vision (ICARCV), pp. 1673-1679, 2010.

[15] O. Abedinia, N. Amjady, and M. S. Naderi, "Multi-objective Environmental/Economic Dispatch using firefly technique", International Conference on Environment and Electrical Engineering (EEEIC), pp. 461-466, 2012.

[16] I. A. Farhat and M. E. El-Hawary, "Multi-objective economic-emission optimal load dispatch using bacterial foraging algorithm", IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), pp. 1-5, 2012.

[17] O. A. C. Cortes, A. Rau-Chaplin, D. Wilson, and J. Gaiser-Porterz, "Efficient Optimization of Reinsurance Contracts using Discretized PBIL", In Proceedings of Data Analytics, pp. 18-24, Porto, 2013.

[18] O. A. C. Cortes, A. Rau-Chaplin, D. Wilson, and J. Gaiser-Porter, "On PBIL, DE and PSO for Optimization ofReinsurance Contracts", EvoStar, EvoFin, Barcelona, 2014.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization", Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948, 1995.

[20] J. Cai, K. S. Tan, C. Weng, and Y. Zhang, "Optimal reinsurance under VaR and CTE risk measures", Insurance: Mathematics and Economics, no. 43, 2008, pp. 185-196.

[21] R. Giusti and G. E. A. P. A. Batista, "Discovering Knowledge Rules with Multi-Objective Evolutionary Computing", 2010 Ninth International Conference on Machine Learning and Applications (ICMLA), pp. 119–124, 2010.

[22] A. Nikabadi and M. Ebadzadeh, "Particle swarm optimization algorithms with adaptive Inertia Weight : A survey of the state of the art and a Novel method", IEEE journal of evolutionary computation, 2008.

[23] M. Clerc, "Standard Particle Swarm Optimisation", hal-00764996, version 1, Retrieved [June,2013], available in: http://hal.archives-ouvertes.fr/hal-00764996, 2012.

[24] _, "The Comprehensive R Archive Network", Retrieved [August, 2014], available in: http://cran.r-project.org/, 2014.

[25] _, "RStudio", Retrieved [August, 2014], available in: http://www.rstudio.com/, 2014.

[26] N. Singh and Y. Kumar, "Economic load dispatch with environmental emission using MRPSO", IEEE 3rd International Advance Computing Conference (IACC), pp. 995-999, 2013.

[27] Q. Zhang and H.i Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition", IEEE Transactions on Evolutionary Computation, vol.11, no.6, 2007, pp. 712–731.

[28] K. E. Parsopoulos, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Vector Evaluated Differential Evolution for Multiobjective Optimization", Congress on Evolutionary Computation, vol. 1, pp. 204–211, 2004.

[29] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", Technical Report, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 2001.

 ISBN: 978-1-61208-358-2