DOI: 10.48011/sbse.v1i1.2248

A Novel Parallel SPEA2 for Solving the Environmental-Economic Dispatch Problem: Competitive vs Cooperative Approach

João Pedro Augusto Costa * **Omar Andres Carmona Cortes** ** **Osvaldo Ronald Saavedra** ***

* *Programa de Pós-Graduação em Engenharia da Computação e Sistemas, Universidade Estadual do Maranhão, MA, (e-mail: jpac1207@gmail.com)*
** *Departamento de Computação, Instituto Federal do Maranhão, MA, (e-mail: omar@ifma.edu.br)*
*** *Instituto de Energia Elétrica, Universidade Federal do Maranhão, MA, (e-mail: o.saavedra@ieee.org)*

Abstract
This paper aims to compare two different parallel approaches (cooperative and competitive) of the SPEA2 for solving the environmental-economic dispatch problem. The idea is to solve the problem by executing the SPEA2 algorithm along with three different meta-heuristics (Genetic Algorithms, Particle Swarm Optimization, and Differential Evolution) to perform changes in the population. The different meta-heuristics work in parallel using two different approaches. The first one is the competitive approach, in which meta-heuristics compete for producing the best set of candidate solutions for solving the problem. Whereas, the cooperative approach selects the new population merging all individuals from all meta-heuristics, then selecting the solution set for the Pareto frontier. The proposal was implemented in C++ using MPI in a master-slave parallel model. Two study cases were used: the first one with six generators and the second one with forty generators. Results showed that the cooperative approach presented the best Pareto frontier for the case of 40 generators.

Keywords: Evironmental-Economic Dispatch; SPEA2; Parallel; Competitive, Cooperative.

1. INTRODUCTION

The environmental-economic dispatch (EED) problem is a significant optimization problem in the power system operation due to the growing concern about environmental pollution caused by thermal power plants (Bora et al., 2019). The primary purpose of the EED for electric power generation is to program the dedicated generating unit outputs for matching the load demand at a minimum operating cost and, at the same time, to reach the minimum pollutant emission. Therefore, we can realize that the EED problem is a multi-objective optimization problem with conflicting objectives, *i.e.*, the lesser the operation cost, the bigger the pollutant emission and vice-versa.

Literature has reported different techniques for dealing with the EED problem. In Granelli et al. (1992), the problem is reduced to a single objective one by transforming the emission into a constraint, in which the user can define an allowable limit. This approach, however, has a critical difficulty in obtaining the trade-off relations between cost and emission (Osman et al., 2009), since the user must be aware of the emission limits. Another approach is to convert the EED into a single objective problem by using a linear combination of the different objectives, *i.e.*, combining both objectives using a weighted sum, as we can see in Aydin et al. (2014) and Bhattacharjee et al. (2014).

On the one hand, the weighted sum approach makes the algorithm implementation easier. On the other hand, the main drawback of this approach is the difficulty in obtaining the Pareto frontier because each execution can reach only one point at a time. Therefore, we have to execute the program at least as many times as we want solutions, which is time-consuming and imprecise. This scenario demands algorithms that can build Pareto frontiers and presents the possible trade-offs between cost and emission. In this context, multiobjective algorithms such as SPEA2 (Strong Pareto Evolutionary Algorithm 2)(Zitzler et al., 2001) come to solve this kind of problem. In fact, the SPEA2 has been successfully used in works that deal with multiobjective problems, such as Lanza-Gutierrez et al. (2013); Xiao et al. (2014); Wang et al. (2015); Golchin and Liew (2016); Qasim and Ismail (2017).

Although SPEA2 can found Pareto frontiers efficiently, the market usually demands quick decision-make. In this context, we can leverage hardware advances and perform complex optimizations quickly and scalable in competitive domains (Brown et al., 2014). One of these hardware advances is the popularization of multi-core processors, which improves the overall throughput by executing multiple threads in parallel on different cores (Kumar et al., 2017). Such technology became available in the middle of 2000s.

Since then, they became widespread even in mobile devices such as tablets and cell phones.

In this context, the main advanced in this paper is to explore two different parallel approaches (cooperative and competitive) on updating populations in the SPEA2 for solving the EED problem. The idea starts from Costa et al. (2017), in which the population on SPEA2 can be updated adaptively using three different meta-heuristics: Genetic Algorithms (GA) (Michalewicz, 1999), Particle Swarm Optimization (PSO) (Eberhart et al., 1995), and Differential Evolution (DE) (Storn and Price, 1995). In our proposal, we use a parallel approach. Thus, the adaptivity is no longer necessary because the different meta-heuristics are executed in parallel. Thus, we aim to answer the following question: which approach produces better results, the cooperative or the competitive model for the EED problem. To answer this question, we implemented the new approach using Message Passing Interface (MPI) MPI (2017) in C++ Language.

This paper is divided as follows: Section 2 introduces the basic concept of multiobjective optimization using the Pareto approach; Section 3 presents how the SPEA2 works and how the algorithm has been parallelized; Section 4 shows the results of this paper; finally, Section 5 presents the conclusions and future work.

2. MULTIOBJECTIVE FUNDAMENTALS

A multiobjective optimization problem (MOP) has to deal with two or more conflicting objective function Deb and Kalyanmoy (2001) 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 $f : X \to Y$ evaluates the quality of solutions mapping it into an objective space. Therefore, a multiobjective problem is defined as $Max\ or\ Min\ y = f(x) = (f_1(x_1, ..., x_m), ..., f_n(x_1, ..., x_m))$, where f is a vector of objective functions, m is the dimension of the problem and n the number of objective functions.

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 $pf = \{f_i(x) \in \mathbf{R} \mid x \in \wp\}$.

2.1 Environmental-Economic Dispatch

The environmental economic dispatch is a bi-objective problem in which both fuel cost and pollution emission have to be minimized. The cost is computed using Equation 1 while the emission is calculated by Equation 2, where P_i is the power in the i^{th} generator, a_i, b_i, c_i, α_i, β_i, and γ_i are operation coefficients.

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

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

subject to

$$\sum_{min}^{max} P_i \geq P_d \tag{3}$$

$$P_{min} \leq P_i \leq P_{max} \tag{4}$$

The constraint presented in Equation 3 is the required demand, *i.e.*, the sum of all powers has to be equal or greater than a specific demand. The constraint shown in Equation 4 depicts the operation boundaries of each generator.

3. SPEA2

The SPEA2 was proposed by Zitzler et al. (2001). The main difference between SPEA2 and the previous version is the narrowing of the archive to save some memory space. Either, the archive is always filled up whether there are enough non-dominated points or not. Thus, the selection process was modified to order the solutions according to their importance. The Algorithm 1 outlines how the SPEA2 works.

$archive\_size \leftarrow input\_size(n)$
$archive \leftarrow \emptyset$
$pop \leftarrow init\_population(fun_k, pop\_size, dim)$
$archive < -non\_dominated\_sol(pop)$
for (i=1 to max_it) **do**
  $Rt \leftarrow mix(pop, archive)$
  $s \leftarrow compute\_s(Rt)$
  $raw \leftarrow compute\_raw(s)$
  $d \leftarrow compute\_density(Rt)$
  $fitness \leftarrow raw + d$
  $indexes \leftarrow (fitness < 1)$
  $archive\_tmp \leftarrow Rt[indexes,]$
  if $(\#non\_dom == archive\_size)$ **then**
    $archive \leftarrow archive\_tmp$
  else if $(\#non\_dom < archive\_size)$ **then**
    $archive \leftarrow archive\_tmp$
    $archive \leftarrow fill()$
  else
    $archive \leftarrow clustering()$
  end if
  $pop \leftarrow gen\_operators()$
end for

Algorithm 1. SPEA2 Pseudo Code

The SPEA2 starts with an empty archive of size n and a random population for k objective functions ($k \geq 2$). Then, the first set of non-dominated solutions is determined and assigned to the *archive*. Then, a temporary population Rt is made by combining the current population and the archive. Afterwards, the algorithm needs to compute the s vector according to Equation 5, which means the cardinality of a point i in terms of dominance, *i.e.*, how many solutions i dominates in Rt.

$$s(i) = |\{j | j \in Rt \wedge i \succ j\}| \tag{5}$$

The next step is to calculate the *raw* vector, which represents the strength of each non-dominated solution as presented in Equation 6. In other words, if a solution i is dominated by solutions j_1, j_2 and j_3, then its *raw* is $s[j_1] + s[j_2] + s[j_3]$. Therefore, all non-dominated solutions present $raw = 0$. On the other hand, a high $raw[i]$ value means to be dominated by many individuals.

$$raw[i] = \sum_{j \in Rt, j \prec i} s(j) \tag{6}$$

Although the raw fitness provides a sort of niching mechanism based on the concept of Pareto dominance, it may fail when there are too much dominated solutions. Thus, additional density information is incorporated to discriminate between individuals having identical raw fitness values as shown in Equation 7, where σ is the distance to the k^{th} individual. Usually, k is calculated according to $k = |\sqrt{pop\_size}|$.

$$d(i) = \frac{1}{\sigma_i^k + 2} \tag{7}$$

Thus, the final Rt fitness is equivalent to $fitness = raw + d$. Being $d < 1$ and $raw = 0$ in all non-dominated solutions; consequently, all non-dominated solutions will present $fitness < 1$. If the number of non-dominated points is equal to the archive size, then the temporary archive replaces the old one. If the length of the temporary archive is less than the archive size, then the new archive has to be filled up with non-dominated solutions chosen from the smallest to the highest fitnesses. Otherwise, the non-dominated solutions have to be clustered, in which the closer the solutions, the greater the probability of being eliminated, maintaining the diversity in the archive. Finally, the population undergoes genetic operators to move along the search space. Typically, these operators are crossover and mutation, and therefore, the same ones originated from genetic algorithms can be applied.

3.1 The Parallel Version: Competitive and Cooperative

The competitive approach works similarly to a Master-Slave model (Dubreuil et al., 2006). In this model, the SPEA2 algorithm is executed by a master process. While the master process executes methods such as raw and density computation, for example, three slaves are waiting for receiving the main population from the master process. The entire population is sent to slaves. Each slave executes a different meta-heuristic. For instance, process 1 runs a GA, process 2 executes a PSO, and process 3 executes a DE algorithm. After that, the slaves sent their populations to the master process, which selects the best one based on their hypervolume. In other words, the master process will keep the population of the meta-heuristic that presents the best hypervolume. Finally, the chosen population undergo the regular SPEA2 algorithm.

In the cooperative approach, the communication architecture is similar to the competitive model; however, when the master process receives all populations, the algorithm selects the population performing a clustering algorithm, which selects non-dominated solutions from the populations of all three slaves. Regarding the parallelism, it is essential to perform the clustering algorithm that all populations be available to the master process. Doing so demands the use of a barrier to guarantee the availability of data.

4. COMPUTATIONAL EXPERIMENTS

All tests were executed 31 times in a computer with a core i5 processor of 4 cores, 2 real and 2 logic (with hyperthreading), 2.70 GHz frequency, and 6GB of RAM. The number of executions was chosen based on the central limit theorem, which stands that any sample with more than thirty trials tends to be normal. Two metrics have been used: hypervolume and spread. The first one represents the area of dominated solutions, i.e., the better the hypervolume, the better the Pareto frontier (in theory). The second metric is the spread, in which higher values indicate a better distribution of the solutions along the Pareto frontier. All experiments were executed 50 times with 300 iterations. Table 1 shows the parameters of each meta-heuristic.

Table 1. Meta-Heuristic's parameters

GA	$P_m = 0.1$ and $P_c = 0.8$
PSO	$w = 0.9$ and $c_1 = c_2 = 0.5$
DE	$CR = 0.5$ and $F = 1.0$

4.1 Case Study 1: 6 Generators

Table 2 presents the coefficients and boundaries for the case of 6 generators. Data were obtained from Singh and Kumar (2016) and Cortes et al. (2014) work.

Regarding hypervolume and the spread metrics, Table 3 shows the results obtained by both cooperative and competitive approaches for the demands of 500MW and 700MW. Even though the competitive approach presents a better hypervolume, the cooperative algorithm presents a better spread. Visually, the cooperative approach seems to present more non-dominated solutions for 500MW, as shown in Figure 1. On the other hand, the competitive algorithm shows more solutions when the cost of production is lower.

Figure 2 shows the Pareto frontiers for 700MW. In this case, the competitive mode dominates the cooperative when the lowest generation cost is considered as we can see in the solutions presented in Table 4. That is the reason why the competitive approach presents a better hypervolume.

4.2 Case Study 2: 40 Generators

Table 5 presents the data for the 40 generators case study, and Table 6 shows de metric results attending a demand for 10500MW in which the competitive approach reached the best results. However, a detail in the Figure 3 shows that, even though the competitve approach presents the

Table 2. Generators and cost coefficients

i	P_{min}	P_{max}	a	b	c	α	β	γ
1	10	125	0.01	2	10	0.00419	0.32767	13.85932
2	10	150	0.012	1.5	10	0.00419	0.32767	13.8593
3	35	225	0.004	1.8	20	0.00683	-0.54551	40.26690
4	35	210	0.006	1	10	0.00683	-0.54551	40.26690
5	130	325	0.004	1.8	20	0.00461	-0.51116	42.89553
6	125	315	0.01	1.5	10	0.00461	-0.51116	42.89553

Table 3. Metrics for the EED: Hypervolume and Spread

Hypervolume		
Demand	Cooperative	Competitive
500MW	62.28	**64.71**
700MW	3369.37	**3828.38**
Spread		
Demand	Cooperative	Competitive
500MW	**601.29**	588.981
700MW	**783.869**	756.262

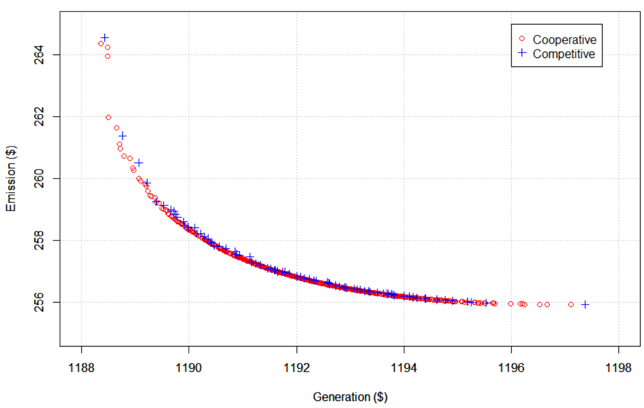

Figure 1. Cooperative vs Competitive Pareto frontiers with demand of 500MW

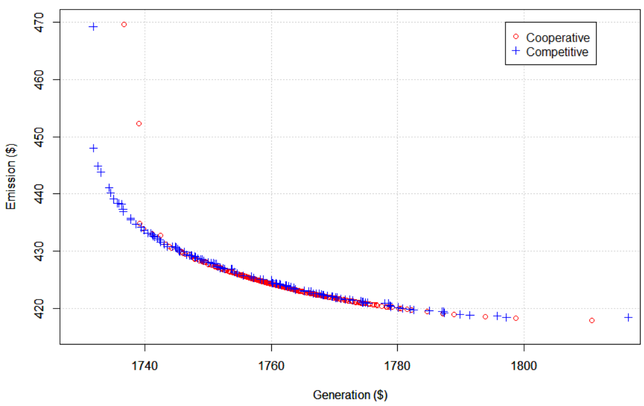

Figure 2. Cooperative vs Competitive Pareto frontiers with demand of 700MW

best outcomes, the cooperative one dominates part of the competitive frontier reaching a better trade-off within the interval $[134000, 142000]$ of the Generation axis as presented in Figure 4.

Table 4. Lowest solutions - 6 generators: production cost vs emission cost

500MW		
	Cooperative	Competitive
Generation	**($1188.36, $264.363)**	($1188.43, $264.569)
Emission	**($1197.11, $255.93)**	($1197.37, $255.947)
700MW		
	Cooperative	Competitive
Generation	($1736.76, $469.647)	**($1731.82, $469.213)**
Emission	($1810.72, $417.901)	($1816.46, $418.417)

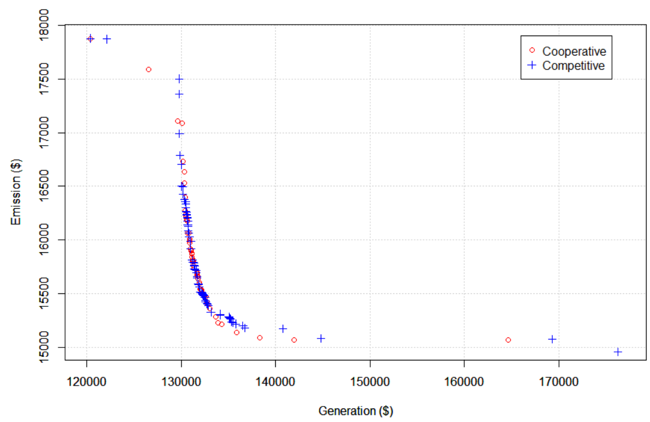

Figure 3. Cooperative vs Competitive Pareto frontiers with demand of 10500W

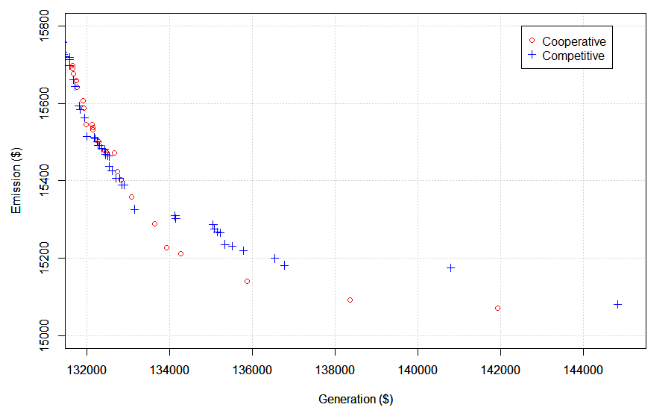

Figure 4. Cooperative vs Competitive Pareto frontiers with demand of 10500W - Zoom

Concerning solutions of the case study 2, the best solutions in terms of generation and emission costs are presented in Table 7. As we can see, the best solutions are non-dominated ones in both cooperative and competitive. However, when we analyze a trade-off we can see how the

Table 5. Generators and cost coefficients for 40 generators

i	P_{min}	P_{max}	a	b	c	α	β	γ
1	36	114	0.0069	6.73	94.705	0.0057	0.033	7.248
2	36	114	0.0069	6.73	94.705	0.0046	0.0458	19.834
3	60	120	0.02028	7.07	309.54	0.0025	0.0469	18.317
4	60	190	0.00942	8.18	369.03	0.0028	-0.0446	19.22
5	47	97	0.0114	5.35	148.89	0.0058	0.0008	10.18
6	68	140	0.01142	8.05	222.33	0.0053	0.0481	14.774
7	110	300	0.00357	6.99	278.71	0.0052	0.0167	6.007
8	135	300	0.00492	6.6	391.98	0.0056	0.0478	17.934
9	135	300	0.00573	6.6	455.76	0.0057	0.0499	14.468
10	130	300	0.00605	12.9	722.82	0.0052	0.0411	17.984
11	94	375	0.00515	12.9	635.2	0.0033	-0.0553	11.002
12	94	375	0.00569	12.8	654.69	0.0059	0.0281	21.727
13	125	500	0.00421	12.5	913.4	0.0047	0.01	16.742
14	125	500	0.00752	8.84	1760.4	0.0047	-0.0319	5.492
15	125	500	0.00708	9.15	1728.3	0.004	0.0498	17.754
16	125	500	0.00708	9.15	1728.3	0.0056	0.046	19.684
17	220	500	0.00313	7.97	647.85	0.0059	-0.0208	13.608
18	220	500	0.00313	7.97	649.69	0.0043	-0.0417	6.374
19	242	550	0.00313	7.97	647.83	0.0051	-0.0034	17.277
20	242	550	0.00313	7.97	647.81	0.0049	0.0463	6.81
21	254	550	0.00298	6.63	785.96	0.0024	0.0092	20.634
22	254	550	0.00298	6.63	785.96	0.004	0.0387	11.574
23	254	550	0.00284	6.66	794.53	0.005	0.0479	9.36
24	254	550	0.00284	6.66	794.53	0.0036	0.0462	19.848
25	254	550	0.00277	7.1	801.32	0.0027	0.0497	12.101
26	254	550	0.00277	7.1	801.32	0.0038	0.0356	18.162
27	10	150	0.52124	3.33	1055.1	0.0056	0.0054	21.305
28	10	150	0.52124	3.33	1055.1	0.006	0.0088	18.734
29	10	150	0.52124	3.33	1055.1	0.0025	0.0472	19.399
30	47	97	0.0114	5.35	148.89	0.0024	-0.0435	14.765
31	60	190	0.0016	6.43	222.92	0.0029	0.0491	5.914
32	60	190	0.0016	6.43	222.92	0.0049	-0.0328	7.28
33	60	190	0.0016	6.43	222.92	0.0051	0.0311	7.546
34	90	200	0.0001	8.95	107.87	0.0042	-0.0313	20.767
35	90	200	0.0001	8.62	116.58	0.005	0.0069	22
36	90	200	0.0001	8.62	116.58	0.006	-0.0009	9.143
37	25	110	0.0161	5.88	307.45	0.0058	0.03	7.102
38	25	110	0.0161	5.88	307.45	0.0022	0.0423	11.21
39	25	110	0.0161	5.88	307.45	0.0056	0.0327	11.206
40	242	550	0.00313	7.97	647.83	0.0026	-0.0408	6.195

Table 6. Metrics for the EED: Hypervolume and Spread. The case of 40 generators.

10500MW		
Demand	Cooperative	Competitive
Hypervolume	95157152	**125900356**
Spread	166496	**178593**

cooperative solution dominates the competitive one, which is seen in Figure 4 as well.

Table 7. Lowest solutions - 40 generators: production cost vs emission cost

10500MW		
	Cooperative	Competitive
Generation	(\$120368, \$17877)	(\$120368, \$17877)
Emission	(\$164647, \$15071.5)	(\$176281, \$14959.9)
Trade-off	**(\$135871, \$15139.9)**	(\$136779, \$15181.9)-

4.3 Sppedup

Table 8 shows the speedup for 6 generators. The best speedup was obtained by the competitive approach reaching a speedup of 2.45 that leads to an efficiency of 82%. The cooperative approach presents a worse speedup because the master process has to process all populations to obtain the Pareto frontier, and update the archive to the next iteration.

Table 8. Speedup - 6 Generators

	Serial Coop.	Parallel Coop.	Serial Comp.	Parallel Comp.
Mean	177.818	146.253	139.871	57.201
Std. Dev.	3.192	1.478	6.911	11.765
Speedup	-	1,22	-	**2,45**
Efficiency	-	0,41	-	**0,82**

Table 9. Speedup - 40 Generators

	Serial Coop.	Parallel Coop.	Serial Comp.	Parallel Comp.
Mean	13569.13	3540.186	14496.97	3385.48
Std. Dev.	892.003	53.952	2751.189	102.68
Speedup	-	3.8328	-	4.282
Efficiency	-	0.9582	-	1.0705

5. CONCLUSIONS

This work presented two parallel approaches to the SPEA2 algorithm for solving the EED problem. The competitive approach presented better hypervolumes in all case studies. Especially for the demand of 700MW in the six generators case and 100500MW in 40 generators case, which are problems more challenging to solve. However, the cooperative approach present a better Pareto frontier in the 40 generators case. Regarding the speedup, the six generators case reached a speedup of 2.45 in the competitive mode and 4.28 in the case of 40 generators. Because the second case study is computationally intense, both approaches reaches almost the ideal speedup.

Future work includes increasing the scalability of the algorithm allowing to execute as many meta-heuristic as the number of cores. Then, devise an algorithm to select which meta-heuristic goes to which core. Also, a fuzzy-based meta-heuristic selection can be implemented.

REFERENCES

Aydin, D., Özyön, S., Yaşar, C., and Liao, T. (2014). Artificial bee colony algorithm with dynamic population size to combined economic and emission dispatch problem. *International Journal of Electrical Power & Energy Systems*, 54, 144–153.

Bhattacharjee, K., Bhattacharya, A., and nee Dey, S.H. (2014). Solution of economic emission load dispatch problems of power systems by real coded chemical reaction algorithm. *International Journal of Electrical Power & Energy Systems*, 59, 176–187.

Bora, T.C., Mariani, V.C., and Coelho, L.S. (2019). Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm. *Applied Thermal Engineering*, 146, 688 – 700.

Brown, L., Beria, A.A., Cortes, O.A.C., Rau-Chaplin, A., Wilson, D., Burke, N., and Gaiser-Porter, J. (2014). Parallel mo-pbil: Computing pareto optimal frontiers efficiently with applications in reinsurance analytics. In *2014 International Conference on High Performance Computing Simulation (HPCS)*, 766–775.

Cortes, O.A.C., Rau-Chaplin, A., Wilson, D., Cook, I., and Gaiser-Porter, J. (2014). A study of VEPSO approaches for multiobjective real world applications. In *Proceedings of The Third International Conference on Data Analytics*, 42–48.

Costa, J.P.A., Cortes, O.A.C., and Júnior, E.C. (2017). An adaptive algorithm for updating populations on SPEA2. In *Simpósio Brasileiro de Automação Inteligente (SBAI)*.

Deb, K. and Kalyanmoy, D. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Inc., New York, NY, USA.

Dubreuil, M., Gagné, C., and Parizeau, M. (2006). Analysis of a master-slave architecture for distributed evolutionary computations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(1), 229–235.

Eberhart, R.C., Kennedy, J., et al. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, volume 1, 39–43. New York, NY.

Golchin, M. and Liew, A.W.C. (2016). Bicluster detection using strength pareto front evolutionary algorithm. In *Proceedings of the Australasian Computer Science Week Multiconference*, 30, 1–6. ACM, New York, NY, USA.

Granelli, G.P., Montagna, M., Pasini, G., and Marannino, P. (1992). Emission constrained dynamic dispatch. *Electric Power Systems*, 24, 56–64.

Kumar, R., Cano, J., Brankovicy, A., Pavlouz, D., Stavrouz, K., Gibertx, E., Martínez, A., and González, A. (2017). Hw/sw co-designed processors: Challenges, design choices and a simulation infrastructure for evaluation. In *2017 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 185–194.

Lanza-Gutierrez, J.M., Gomez-Pulido, J.A., Vega-Rodriguez, M.A., and Sanchez-Perez, J.M. (2013). A parallel evolutionary approach to solve the relay node placement problem in wireless sensor networks. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, 1157–1164. ACM, New York, NY, USA.

Michalewicz, Z. (1999). *Genetic Algorithms + Data Structures = Evolution Programs (3rd Ed.)*. Springer-Verlag, Berlin, Heidelberg, 3rd edition.

MPI, O. (2017). Open MPI: Open source high performance computing, https://www.open-mpi.org, last visit: Sep-01st-2017.

Osman, M., Abo-Sinna, M., and Mousa, A.a. (2009). An e-dominance-based multiobjective genetic algorithm for economic emission load dispatch optimization problem. *Electric Power Systems Research*, 79, 1561–1567.

Qasim, S.Z. and Ismail, M.A. (2017). Research problems in search-based software engineering for many-objective optimization. In *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, 1–6.

Singh, N. and Kumar, Y. (2016). Economic load dispatch with environmental emission using mrpso. In *IEEE 3rd International Advance Computing Conference (IACC)*, 995–999.

Storn, R. and Price, K. (1995). *Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces*, volume 3. ICSI Berkeley.

Wang, H., Cortes, O.A.C., and Rau-Chaplin, A. (2015). Dynamic optimization of multi-layered reinsurance treaties. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 125–132. ACM, New York, NY, USA.

Xiao, J., Wu, Z., and Tang, J.C. (2014). Hybridization of electromagnetism with multi-objective evolutionary algorithms for rcpsp. In *Proceedings of Conference on Genetic and Evolutionary Computation (GECCO)*. ACM, New York, NY, USA.

Zitzler, E., Laumanns, M., Thiele, L., Zitzler, E., Zitzler, E., Thiele, L., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm.