

Supplementary material: On the performance of the hybridisation between migrating birds optimisation variants and differential evolution for large scale continuous problems

Eduardo Segredo^{a,b,*}, Eduardo Lalla-Ruiz^c, Emma Hart^a, Stefan Voß^{c,d}

^a*School of Computing, Edinburgh Napier University, 10 Colinton Road, Edinburgh, EH10 5DT, Scotland, United Kingdom*

^b*Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, San Cristóbal de La Laguna, Spain*

^c*Institute of Information Systems, University of Hamburg, Hamburg, Germany*

^d*Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile*

1. Description of the parameter tuning procedure

This supplementary material provides the study regarding the parameterisation applied together with each algorithm considered in the paper. The parameter tuning procedure was performed for the first, second and third experiments reported in Section 4 of the paper, later required to compare the best-performing variants in subsequent experiments.

Particularly, parameterisations were defined by using a *Taguchi's Orthogonal Array* factorial design of four factors and three levels, with the aim of reducing the number of different configurations of a full-factorial design with the same number of factors and levels. This results in 9 different parameterisations of E-MBO and MBO, each of them using parameter values shown in Table 1.

Table 1: Parameter values for the different configurations of E-MBO and MBO

Configuration	0	1	2	3	4	5	6	7	8
Population size (n)	150	150	150	250	250	250	350	350	350
Neighbourhood size (k)	7	9	11	7	9	11	7	9	11
Number of iterations (m)	10	5	15	5	15	10	15	10	5
Number of neighbours to be shared (x)	1	2	3	3	1	2	2	3	1

To assess the performance of the different configurations of E-MBO and MBO, a ranking procedure was used. It was calculated as follows. For a particular approach, the number of times it was able to statistically outperform other configurations (\uparrow), the number of times it was statistically outperformed by the remaining configurations (\downarrow), and the number of times it did not show statistically significant differences with other configurations (\leftrightarrow), considering all functions, were calculated by applying the statistical procedure explained at the beginning of Section 4. Configuration A statistically outperforms configuration B if there exists statistically significant differences between them, and if at the same time, A provides a lower mean and median of the error with respect to the objective value than B , since we are dealing with minimisation problems. For each function, every configuration was statistically compared to the remaining ones, thus performing 36 pairwise statistical comparisons. Therefore, a total number of 540 pairwise statistical comparisons were carried out for each of both schemes E-MBO and MBO, considering all functions. Once statistical comparisons were performed, the score assigned to a given configuration was calculated as the difference between the number of schemes it was able to beat and the number of schemes that were able to beat it. Then, a ranking was established by sorting strategies in descending order taking into account the scores assigned. Finally, the configurations of E-MBO and MBO that obtained the highest scores are shown in **boldface**.

*Corresponding author

Email addresses: e.segredo@napier.ac.uk (Eduardo Segredo), eduardo.lalla-ruiz@uni-hamburg.de (Eduardo Lalla-Ruiz), e.hart@napier.ac.uk (Emma Hart), stefan.voss@uni-hamburg.de (Stefan Voß)

Table 2: Ranking of E-MBO and MBO applying the neighbourhood operator based on DE/rand/1/bin with fixed parameters

Configuration	↑	↓	↔	Score	Ranking	Configuration	↑	↓	↔	Score	Ranking
E-MBO-0	63	17	40	46	2	MBO-0	61	6	53	55	1
E-MBO-1	64	16	40	48	1	MBO-1	61	11	48	50	3
E-MBO-2	60	17	43	43	4	MBO-2	61	9	50	52	2
E-MBO-3	68	24	28	44	3	MBO-3	48	25	47	23	4
E-MBO-4	50	34	36	16	5	MBO-4	44	29	47	15	6
E-MBO-5	46	33	41	13	6	MBO-5	47	28	45	19	5
E-MBO-6	7	75	38	-68	7	MBO-6	5	72	43	-67	8
E-MBO-7	7	75	38	-68	7	MBO-7	6	72	42	-66	7
E-MBO-8	6	80	34	-74	8	MBO-8	1	82	37	-81	9

1.1. First experiment

Table 2 shows the ranking resulting of studying MBO variants making use of the explorative neighbourhood operator based on DE/rand/1/bin with fixed parameters. It can be observed that configurations of E-MBO and MBO with a smaller population size performed better, i.e., obtained higher scores, than parameterisations making use of a larger population size. In fact, E-MBO-1 and MBO-0 obtained the highest scores, both of them using a population of $n = 150$ individuals. Since the neighbour generation operator based on DE/rand/1/bin promotes exploration, using a very large population may be counter-productive as it can lead to a reduction in the convergence speed. Hence, smaller populations are preferred.

In order to analyse the robustness of E-MBO with respect to the remaining parameters, we only considered the results obtained in all functions by its configurations using $n = 150$ individuals (E-MBO-0, E-MBO-1, and E-MBO-2), and repeated the ranking calculation. Significant differences among configurations with different values for parameters k , m , and x did not appear in 34 out of 45 statistical tests, which represents more than 75% of the cases. The same fact was observed with the configurations of MBO using $n = 150$ individuals (MBO-0, MBO-1, and MBO-2), but in this particular case, statistically significant differences did not arise in 95.5% of the comparisons, which is even more noticeable. Taking into account the configurations of E-MBO and MBO using $n = 250$ and $n = 350$ individuals, results were similar, where statistically significant differences did not arise in 80% of the cases, with the exception of the configurations of E-MBO with $n = 250$ individuals, where statistical differences did not appear in more than 57% of the comparisons.

As a result, we can conclude that both E-MBO and MBO are very robust from the point of view of their parameters when they are applied together with an explorative neighbourhood operator based on DE/rand/1/bin. By modifying the values of parameters k , m , and x , the performance of both approaches does not change in a significant manner. A suitable selection of the population size, however, must be carried out. By applying this particular neighbourhood operator, E-MBO and MBO provide better results with smaller populations.

1.2. Second experiment

Table 3 shows the ranking of the different configurations of E-MBO and MBO applying the explorative neighbourhood operator based on the adaptive (JADE) DE/rand/1/bin. Similar conclusions to those stated in previous section can be drawn regarding the analysis of the population size n . It can be observed that configurations E-MBO-0 and MBO-0 obtained the highest score: both contain $n = 150$ individuals, i.e., the lowest population size and get the first position in the ranking. Furthermore, in general terms, configurations with larger population sizes performed worse than approaches with lower populations, with the exception of schemes E-MBO-4 and E-MBO-5, as well as MBO-4 and MBO-5. The above configurations, which were applied with $n = 250$ individuals, achieved similar scores than those attained by E-MBO-0 and MBO-0 respectively, which used $n = 150$ individuals. In fact, E-MBO-5 obtained the same score than E-MBO-0. Since the neighbour generation operator applied in this experiment is based on an adaptive version of DE, the population size might be a more robust parameter in this case. Using adaptation sustains the superior performance of E-MBO and MBO in some cases, even though the population size is increased. However, we note that the adaptive neighbour generation operator is based on the the explorative variant DE/rand/1/bin. As a result, the usage of a very large population, such as $n = 350$, which promotes even more the exploration ability of the optimisation scheme, might significantly decrease the performance of E-MBO and MBO, as it also happened in the first experiment.

Table 3: Ranking of E-MBO and MBO applying the neighbourhood operator based on the adaptive DE/rand/1/bin (JADE)

Configuration	↑	↓	↔	Score	Ranking	Configuration	↑	↓	↔	Score	Ranking
E-MBO-0	43	21	56	22	1	MBO-0	48	24	48	24	1
E-MBO-1	27	33	60	-6	5	MBO-1	39	30	51	9	4
E-MBO-2	31	31	58	0	3	MBO-2	37	33	50	4	5
E-MBO-3	28	35	57	-7	6	MBO-3	33	36	51	-3	6
E-MBO-4	44	26	50	18	2	MBO-4	44	28	48	16	3
E-MBO-5	46	24	50	22	1	MBO-5	42	24	54	18	2
E-MBO-6	32	40	48	-8	7	MBO-6	25	47	48	-22	8
E-MBO-7	31	36	53	-5	4	MBO-7	27	41	52	-14	7
E-MBO-8	30	66	24	-36	8	MBO-8	31	63	26	-32	9

Another explanation to the above fact might be the following. Observing Algorithm 1 and how the method by which the control mechanism provided by JADE is integrated into E-MBO and MBO (explained at the end of Section 3.2), we note that the mechanism responsible for updating the mean μ_{CR} is executed m times at each generation of both algorithms, with m being the number of iterations. The larger the number of iterations m , the larger the number of updates of μ_{CR} per generation, and therefore leading to more accurate control of CR depending on the current moment of the search process. At the same time, Equation 1 shows the number of neighbours S generated at each of those m iterations of E-MBO and MBO depending on their parameters, which corresponds to the number of times that the distribution responsible for providing values for CR is sampled at each iteration m .

$$S = k + (n - 1) \cdot (k - x) \quad (1)$$

With a larger number of samples S , the variability of CR values, and consequently, the variability of potential successful CR values is also larger. It is important to recall, from Section 3.2, that successful values of CR are used by the mechanism provided by JADE to update μ_{CR} . Therefore, with larger values of S , the control of the crossover rate CR might be much more accurate. However, by increasing S , the number of generations performed by E-MBO and MBO decreases, as also observed when increasing the number of iterations (or μ_{CR} updates) m . This is depicted in Equation 2, which shows the number of generations g of a given run depending on the values of its parameters and its total number of evaluations $numEvals$.

$$g = \left\lceil \frac{numEvals}{m \cdot S} \right\rceil \quad (2)$$

Hence, the performance of E-MBO and MBO combined with this particular adaptive neighbour generation operator may be significantly altered depending on the values assigned to their parameters n , m , k , and x , which are related to the mechanism provided by JADE to control the crossover rate CR used by the neighbourhood operator¹. This would explain, on the one hand, the fact that configurations with different values for their parameters are able to perform in a similar manner, such as E-MBO-0 and E-MBO-5, and on the other hand, the fact that configurations sharing some values for their parameters perform significantly different, like MBO-3 with respect to MBO-4 and MBO-5, all of them applied with $n = 250$ individuals. For instance, in the first case, both E-MBO-0 and E-MBO-5 were applied with $m = 10$ iterations, and therefore, both of them updated $m = 10$ times per generation the value of μ_{CR} . Nevertheless, the value of S considered by E-MBO-0 was lower ($S = 901$) than the value used by E-MBO-5 ($S = 2252$), while the number of generations of the former ($g = 333$) was higher than the number of generations of the latter ($g = 134$). As a result, E-MBO-0 was able to provide similar performance than E-MBO-5 by carrying out a higher number of generations, in spite of adapting CR in a more inaccurate manner.

In summary, the population size n of E-MBO and MBO, making use of a neighbour generation operator based on an adaptive version of DE/rand/1/bin, becomes a more robust parameter, since altering its value does not significantly change the performance of both approaches. The usage of a very large population, however, might be counter-productive. In addition, it is important to properly set the remaining parameters

¹All the observations made in the current section in regard to the control mechanism provided by JADE for adapting the crossover rate CR are also extensible to the control mechanism responsible for adapting the mutation scale factor F .

Table 4: Ranking of E-MBO and MBO applying the neighbourhood operator based on the adaptive DE/current-to-p-best/1/bin (JADE)

Configuration	↑	↓	↔	Score	Ranking	Configuration	↑	↓	↔	Score	Ranking
E-MBO-0	9	74	37	-65	7	MBO-0	13	68	39	-55	7
E-MBO-1	3	84	33	-81	9	MBO-1	1	78	41	-77	8
E-MBO-2	1	81	38	-80	8	MBO-2	1	81	38	-80	9
E-MBO-3	40	41	39	-1	6	MBO-3	39	38	43	1	6
E-MBO-4	59	15	46	44	3	MBO-4	63	13	44	50	3
E-MBO-5	44	30	46	14	5	MBO-5	41	31	48	10	5
E-MBO-6	72	8	40	64	2	MBO-6	66	8	46	58	2
E-MBO-7	57	18	45	39	4	MBO-7	50	17	53	33	4
E-MBO-8	81	15	24	66	1	MBO-8	75	15	30	60	1

of E-MBO and MBO. A suitable balance between the number of generations g , the number of times m that the updating mechanisms provided by JADE are executed at each of those generations, and the number of neighbours S produced at each iteration, should be established in order to properly control the parameters of the neighbour generation operator, and consequently, to improve the performance of the approaches.

1.3. Third experiment

In Table 4, the resulting ranking of the different configurations for E-MBO and MBO using the exploitative neighbourhood operator based on the adaptive (JADE) DE/current-to-p-best/1/bin is reported. As it can be observed, in opposition to previous experiments, the configurations of E-MBO and MBO with a larger population size provided better overall performance than schemes using a smaller population. In fact, approaches E-MBO-8 and MBO-8, both of them applied with $n = 350$ individuals, obtained the highest scores in the ranking. Since the neighbourhood operator is based on the more exploitative version DE/current-to-p-best/1/bin, the usage of very small populations may produce a negative effect, due to a significant inclination towards exploitation, which drives the algorithm to prematurely converge to local optima. In contrast, using larger populations allows the exploration and exploitation capabilities of the algorithms to be balanced properly.

As an exception to the above however, we note that configurations E-MBO-4 and MBO-4, both of them applied with $n = 250$ individuals obtained better scores than E-MBO-7 and MBO-7, respectively, which were executed considering $n = 350$ individuals. As in the case of the second experiment, parameters of the neighbourhood operator were also updated during the execution, with the only difference being that not only CR , but also F , were adapted by the control mechanisms provided by JADE. As a result, the population size may be a more robust parameter, as previously stated in the second experiment. Controlling both parameters of this particular neighbourhood operator through JADE allows the performance of E-MBO and MBO to be retained, or even improved, for some cases, although the population size is decreased. However, the application of a very small population, like $n = 150$, which further promotes the exploitation ability of E-MBO and MBO, could significantly reduce their performance. With respect to the analysis carried out in the second experiment regarding the remaining parameters of E-MBO and MBO, the same conclusions can be drawn here, as both parameters F and CR are adapted by the control mechanisms provided by JADE. Thus, it is also important to properly set those parameters when both approaches are combined with the neighbourhood operator based on the adaptive DE/current-to-p-best/1/bin.