Prog Artif Intell (2016) 5:85–89
DOI 10.1007/s13748-016-0082-4

REGULAR PAPER

Evolutionary algorithms for large-scale global optimisation: a snapshot, trends and challenges

Daniel Molina Cabrera[1]

Received: 31 December 2015 / Accepted: 7 January 2016 / Published online: 3 February 2016

Abstract In the last years, several real-world problems that require to optimise an increasing number of variables have appeared. This type of optimisation, called large-scale global optimisation, is hard due to the huge increase of the domain search due to the dimensionality. Large-scale global optimisation is a research area getting more attention in the last years, thus many algorithms, mainly evolutionary algorithms, have been specially designed to tackle it. In this paper, we give a brief introduction of several of them and their techniques, remarking techniques based on grouping of variables and memetic algorithms, because they are two promising approaches. Also, we have reviewed the winners of the different competitions in the area, to give a snapshot of the algorithms that have obtained the best results in this area. Finally, several interesting trends in the research area have been pointed out, and some future trends and challenges have been suggested.

Keywords Large-scale global optimisation · Large scale · High-dimensional problems · Real-coding optimisation · Evolutionary algorithms

1 Introduction

Continuous optimisation problems appear very frequently in many important design and control problems in several areas: engineering, telecommunication, etc. Evolutionary algorithms, EAs [2], are very efficient algorithms that can obtain accurate solutions in complex problems without specific information about them, something very important in real-world problems.

In recent years, with improvement in processing capabilities, and the huge increase in data to analyse, new challenges have appeared; one of them is to tackle optimisation problems with an increasing number of variables to optimise. When the number of variables to optimise reaches a high value, it is called large-scale global optimisation, LSGO [4,22]. Therefore, LSGO is the optimisation characterised by a high number of variables to optimise, and it presents the great drawback that the domain search is increased exponentially with the dimension size. Thus, algorithms that tackle these problems have to be even more efficient than the ones designed for problems with a lower dimensionality.

In real-world optimisation problems, and especially when the number of variable is high, there are several dependencies among the variables. The contribution of the variables to the final result could be independent of the other ones, but that is unusual. In the majority of problems there are groups of variables (i.e. related to the same real-world components, or inter-dependencies components) with a strong relationship between them. The first group is called separable functions and they are easier but less common in real problems, and the second one is called non-separable functions and they are more usual and unfortunately more difficult to solve. In non-separable functions, if some variables are related with more than a group of variables, they are called overlapping functions and they are the most difficult case. In order to be able to identify efficient EAs for LSGO and capable to optimise effectively non-separable functions, in the last years some competitions have been held using proposed benchmarks specially designed for LSGO [20], with interesting results.

In this paper, we are going to give a quick snapshot of the algorithms specially designed for LSGO, remarking the

✉ Daniel Molina Cabrera
daniel.molina@uca.es

[1] Department of Computer Science, University of Cádiz, Cádiz, Spain

issues they have in common and what make them different from the other algorithms. Also, we are going to indicate what are, in our opinion, the present trends in the area, and we are going to suggest several open challenges in this growing research area.

2 Snapshot of algorithms for LSGO

There is an increasing interest in LSGO, as can be observed by the special sessions and competitions organised [9,24], and the several special issues and reviews published [7,10,13]. Thus, here we are going to give a quick snapshot of the techniques used and several relevant algorithms. First, we are going to describe briefly co-evolutionary algorithms, that are very used in LSGO optimisation. Then, we are going to describe the algorithm winners of the different LSGO competitions.

2.1 Co-evolutionary algorithms and grouping variables techniques

Co-evolutionary algorithms [26] are algorithms that have been usually applied in large-scale optimisation [30]. These algorithms use a divide-and-conquer approach to decompose a large-scale problem into a set of lower dimensional problems which are easier to optimise. The main challenge is to find an optimal decomposition, because it strongly depends on the problem.

Co-evolutionary algorithms like multilevel cooperative evolution for large-scale optimisation, MLCC [31], or DE-CC-G [30] are algorithms that have obtained good results on the different large-scale competitions. Following that tendency, there have been proposed co-evolutionary versions of algorithms like Particle Swarm Optimisation, PSO, [23], Differential Evolution, DE, [30,32] or Artificial Bee Colony, ABC [21].

Co-evolutionary algorithm are characterised not only by the different algorithms used to solve each subproblem, but also by the decomposition technique used. The more simple is random grouping technique [30] that divides randomly the variables in several non-overlapping groups. Random grouping can increase the probability of several interacting variables being grouped in the same sub-component, without any prior knowledge of the non-separability of a problem. Another more advanced technique is differential grouping [18], that studies the dependencies among variables comparing solutions with small changes in them. Then, it uses that information for grouping together the variables with dependencies among them. This technique implies a cost in evaluations, but it has proven not only give better results over CC using the previous technique [19], but also it allows algorithms, like CMAES [5], with bad behaviour in high-dimensional problems [11], to be applied with very good results [12].

2.2 Winners of LSGO competitions

Since 2008 several LSGO special sessions and special sessions and competitions have been organised at the IEEE flagship conference Congress on Evolutionary Computation (CEC) [9,24].

These competitions on LSGO are very interesting, because we can see from the different years, the current tendencies and which algorithms have obtained the best results. That feedback is very useful to improve the algorithms.

As a general conclusion, one important group of algorithms that have presented very good results in LSGO are memetic algorithms, MAs [16,17], that combine several algorithms with different exploitation factor, using in many cases a local search, LS, method to increase the efficiently of the algorithm. In general, MAs are very efficient algorithms, which is an important feature for LSGO problems. In the CEC'2008 and CEC'2010 competitions, in which more algorithms participated, MAs were the majority of proposals. Also, in following competitions, MAs were the algorithms with best results.

In the following paragraphs, we are going to give a quick summary of the different winners of the different LSGO competitions.

In CEC'2008 the best algorithm was MTS [25], an MA which introduced several local search methods, LS, specially designed for large-scale problems. One of them, called MTS-LS1, has been used by other winners in following years. Other algorithms with good results were an EDA [28], one MA using PSOs [33], and a DE with a dynamic population size, increasing when a certain area is considered interesting enough [3].

In CEC'2010 competition, the winner was MA-SW-Chains [15], a MA combining a genetic algorithm with an LS chaining; the idea is to apply several times the LS to the same solution, a promising solution, and it uses a memory of LS parameters to make it equivalent to apply it once with a greater LS intensity. Other algorithms with good results were a 2-stage ensemble algorithm, that first uses an EA to detect the more promising region and then uses LS to explore that one [27], and other MAs, one using PSO with the powerful harmony search [34], and another one using an ant colony algorithm [6].

In 2011 there was a competition related to a special issue journal [13]. All the proposals were improved by SADEMMTS [35], a self-adaptive DE using the LS proposed originally in MTS, MTS-LS1.

In CEC'2013 the winner was MOS [7,8] an algorithm that combines several very different algorithms: a genetic algorithm, several DEs, etc., in combination with MTS-LS1 and with an adaptive application criterion that selects in each step an algorithm using a certain probability; and these probabilities are adapted increasing the probability of the algorithm that obtains the best results in each case. Other good algorithms were the previously commented DECC-G [30] and CC-CMA-ES [12].

In CEC'2015 the best algorithm in the competition was IHDELS [14], that applies iteratively a self-adaptive DE an LS method, in each step the LS method is chosen using a adaptive probability (similar to MOS) between two: the MTS-LS1 and the quasi-Newton L-BFGS. Although IHDELS could not improve the results obtained by MOS, IHDELS obtained best results than MOS in many non-separable and overlapping functions.

In a recent paper, several modern LSGO algorithms are compared, and the results obtained suggest that MOS could be still considered the current state-of-the-art in LSGO for continuous problems [7].

3 Present trends and challenges

In this section, we are going to describe the present trends and guess about the future trends in this area research. This section is a personal point of view based on my experience both as designer of algorithms for scale global optimisation (I am one of the authors of MA-SW-Chains [15] and IHDELS [14]) and as a organiser of special issue and special sessions on LSGO.

First, a promising line of research is the technique of automatic grouping of variables for LSGO. The random grouping have proven to be rather effective, and different grouping techniques clearly improve them. However, in different grouping techniques, the effort still required for detecting dependencies among variables usually is too high to be very useful in many problems. In the future, more work is expected in this area to reduce the related cost to make it more useful. It is a difficult task, but if it could be applied in real-world problems, when the inter-dependencies are very common, it could be a great advance. Also, I would like to remark that, while LSGO problems are useful to improve these techniques, they can be equally applied for medium-size optimisation problems.

Considering the experience in the different competitions, it has been observed that the good behaviour of the algorithms depends a lot on the landscape of the functions. This is partially an expected behaviour [29], but it is even more pronounced for LSGO optimisation, due to the fact that the huge domain search requires a very effective search, and the efficiently of each algorithm strongly depends on the landscape to optimise. Several algorithms, like MOS, are composed by very different algorithms expecting that, for each function, the algorithm with the best behaviour could lead the search. The main drawback of this approach is that the resulting algorithm is more complex than it could be desired. Because MOS can be considered the state-of-the-art in LSGO, it is a open challenge to design an algorithm that improves it, maintaining at the same time a lower complexity.

In the different competitions the results for different number of evaluations are measured for each function, to study the performance. Considering these results, usually the algorithms with final best results were for many functions the slower in comparisons against others. Thus, another interesting challenge is to obtain algorithms capable of obtaining competitive results during all the running, avoiding algorithms that achieve good results mainly in the final stages of the algorithm. This is very important for a real-problem, because the industry requires algorithms that are efficient under a wide range of processing time.

If we observe in detail the MAs with best results, MTS, MOS or IHDELS, we can observe that they use an LS method specially designed for large-scale problems, MTS-LS1 [25], and it is one of the reasons of their good results. Although there are several LS methods for continuous and real-coding, many of them are not scalable for high-dimensional problems. One possible alternative is using the grouping variable technique before the LS method [1] transparently. However, it could be interesting to have more LS methods designed for high-dimensional problem, like MTS-LS1, and the design of more scalable LS methods is a promising challenge.

About competitions, the benchmarks proposed until now have been focused on comparing algorithms for continuous LSGO. However, the design of a benchmark with functions with different degree of separability could also be applied for combinatorial problems. It could be interesting to consider benchmarks for combinatorial large-scale optimisation, for comparing among them algorithms adequate to that type of problems.

Finally, there is another important issue that has not been indicated until now in this paper, but that it is always very important. Optimising a high-dimensional problem usually takes a lot of time, so, in real applications, algorithms designed for this type of problems should be designed to be implemented easily to run in parallel to reduce their processing time. A greater effort in this feature should be in future proposals, because this could be a crucial feature to decide whether an algorithm is useful in real applications.

4 Conclusions

Large-scale global optimisation, LSGO, is an interesting growing research area that tackles optimisation with a high number of variables. LSGO is a hard problem because the domain search increases exponentially with the dimension size, thus it requires algorithms specially efficient for it. In the last years, several benchmarks for LSGO have been suggested and many EAs specially designed for LSGO have been proposed. Because these benchmarks contain functions with different degree of inter-dependencies among variables, these functions are also a good way to study techniques and algorithms able to obtain good results in non-separable and overlapping functions, usual in real-world problems.

An interesting approach to deal with LSGO is to follow a conquer-and-divide technique, grouping the variables in groups and optimising each one individually. Several different techniques have been proposed for doing these partitions, and the co-evolutionary algorithms that follow this approach obtain good results. Memetic algorithms also yield very good results, specially those that use an LS method adequate for large-scale problems.

In the last years, different competitions have been developed; we have given a brief description of the winners of the different competitions on the *IEEE Congress on Evolutionary Competition* special sessions. From that quick review, we have observed that the algorithms that obtain the best results are hybrid algorithms which use an adequate combination of algorithms. Considering the results of these competitions under the proposed benchmarks, MOS could be considered the current state-of-the-art in LSGO for continuous problem [7]. However, there is margin to improve it and new algorithms obtaining better results could be presented in the near future.

Finally, we have indicated the current trends and suggested several interesting challenges: development of efficient decomposition techniques for grouping variables with inter-dependencies; new LS methods for large-scale problems; new algorithms that could improve MOS; algorithms with better features: simplicity, efficient during all runs, and easier to run in parallel. Also, it is pointed out that it could be interesting to have similar benchmarks for combinatorial problems with different degrees of separability among variables. Thus, it is an active research area with important open challenges.

Acknowledgments This work was supported by the Spanish Ministry of Education and Science under Grants TIN2012-37930-C02-01, TIN2014-57251-P and Research Regional Projects P10-TIC-6858, P12-TIC-2958.

References

1. Ali, A., Hassanien, A., Snášel, V.: The nelder-mead simplex method with variables partitioning for solving large scale optimization problems. In: Abraham, A., Krömer, P., Snášel, V. (eds.) Innovations in Bio-inspired Computing and Applications. Advances in Intelligent Systems and Computing, vol. 237, pp. 271–284. Springer International Publishing (2014)
2. Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. IOP Publishing Ltd., Bristol (1997)
3. Brest, J., Zamuda, A., Fister, I., Maučec, M.: Large scale global optimization using self-adaptive differential evolution algorithm. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2010)
4. Cao, Y., Sun, D.: A parallel computing framework for large-scale air traffic flow optimization. IEEE Trans. Intell. Transp. Syst. **13**(4), 1855–1864 (2012)
5. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. **1**(11), 1–18 (2003)
6. Korosec, P., Tashkova, K., Silc, J.: The differential ant-stigmergy algorithm for large-scale global optimization. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2010)
7. LaTorre, A., Muelas, S., Peña, J.M.: A comprehensive comparison of large scale global optimizers. Inf. Sci. **316**, 517–549 (2015)
8. LaTorre, A., Muelas, S., Pena, J.M.: Large scale global optimization: Experimental results with mos-based hybrid algorithms. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 2742–2749 (2013)
9. Li, X., Tang, K., Omidvar, M., Yang, Z., Qin, K., Tang, K.: Benchmark functions for the CEC'2013 special session and competition on large scale global optimization. Tech. rep., Evolutionary Computation and Machine Learning Group, RMIT University, Australia (2013)
10. Li, X., Tang, K., Suganthan, P., Yang, Z.: Editorial for the special issue of Information Sciences Journal (ISJ) on nature-inspired algorithms for large scale global optimization. Inf. Sci. **316**, 437–439 (2015)
11. Liao, T., Molina, D., Stützle, T.: Performance evaluation of automatically tuned continuous optimizers on different benchmark sets. Appl. Soft Comput. **27**, 490–503 (2015)
12. Liu, J., Tang, K.: Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. In: Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., Yao, X. (eds.) Intelligent Data Engineering and Automated Learning IDEAL 2013. Lecture Notes in Computer Science, vol. 8206, pp. 350–357. Springer Berlin Heidelberg (2013)
13. Lozano, M., Molina, D., Herrera, F.: Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. Soft Comput. **15**(11), 2085–2087 (2011)
14. Molina, D., Herrera, F.: Iterative hybridization of de with local search for the cec'2015 special session on large scale global optimization. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1974–1978 (2015)
15. Molina, D., Lozano, M., Herrera, F.: MA-SW-Chains: memetic algorithm based on local search chains for large scale continuous global optimization. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2010)
16. Moscato, P.: On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetic Algorithms. Tech. rep., Caltech Concurrent Computation Program. California Institute of Technology, Pasaden (1989)
17. Neri, F., Cotta, C.: Memetic algorithms and memetic computing optimization: a literature review. Swarm Evol. Comput. **2**, 1–14 (2012)
18. Omidvar, M., Li, X., Mei, Y., Yao, X.: Cooperative co-evolution with differential grouping for large scale optimization. IEEE Trans. Evol. Comput. **18**(3), 378–393 (2014)
19. Omidvar, M., Mei, Y., Li, X.: Effective decomposition of large-scale separable continuous functions for cooperative co-evolutionary algorithms. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1305–1312 (2014)
20. Omidvar, M.N., Li, X., Tang, K.: Designing benchmark problems for large-scale continuous optimization. Inf. Sci. **316**, 419–436 (2015)

21. Ren, Y., Wu, Y.: An efficient algorithm for high-dimensional function optimization. Soft Comput. **17**(6), 995–1004 (2013)
22. Shi, Y., Zhang, J., O'Donoghue, B., Letaief, K.: Large-scale convex optimization for dense wireless cooperative networks. IEEE Trans. Signal Process. **63**(18), 4729–4743 (2015)
23. Sun, L., Yoshida, S., Cheng, X., Liang, Y.: A cooperative particle swarm optimizer with statistical variable interdependence learning. Inf. Sci. **186**(1), 20–39 (2012)
24. Tang, K., Li, X., Suganthan, P.N., Yang, Z., Weise, T.: Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization. Tech. rep., Nature Inspired Computation and Applications Laboratory (2009)
25. Tseng, L.Y., Chen, C.: Multiple trajectory search for large scale global optimization. In: IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence), pp. 3052–3059 (2008)
26. van den Bergh, F., Engelbrecht, A.: A cooperative approach to particle swarm optimization. IEEE Trans. Evolut. Comput. **8**(3), 225–239 (2004)
27. Wang, Y., Li, B.: Two-stage based ensemble optimization for large-scale global optimization. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2010)
28. Wang, Y., Member, S., Li, B.: A restart univariate estimation of distribution algorithm: sampling under mixed gaussian and lévy probability distribution. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC2008), Hongkong, pp. 3218–3925 (2008)
29. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)
30. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Inf. Sci. **178**(15), 2985–2999 (2008)
31. Yang, Z., Tang, K., Yao, X.: Multilevel cooperative coevolution for large scale optimization. In: IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence), pp. 1663–1670 (2008)
32. Yang, Z., Zhang, J., Tang, K., Yao, X., Sanderson, A.: An adaptive coevolutionary differential evolution algorithm for large-scale optimization. In: IEEE Congress on Evolutionary Computation, 2009. CEC '09, pp. 102–109 (2009)
33. Zhao, S., Liang, J., Suganthan, P., Tasgetiren, M.: Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In: IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence), pp. 3845–3852 (2008)
34. Zhao, S.Z., Suganthan, P., Das, S.: Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2010)
35. Zhao, S.Z., Suganthan, P., Das, S.: Self-adaptive differential evolution with multi-trajectory search for large-scale optimization. Soft Comput. **15**(11), 2175–2185 (2011)