Proceedings of the 16th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2016
4–8 July, 2016.

On applying a parallel Teaching-Learning-Based optimization procedure for automatic heliostat aiming

N.C. Cruz[1], J.L. Redondo[1], J.D. Álvarez[1], M. Berenguel[1] and P.M. Ortigosa[1]

[1] *Dpt. of Informatics, University of Almería, CeiA3, Spain*

emails: ncalvocruz@ual.es, jlredondo@ual.es, jhervas@ual.es, beren@ual.es, ortigosa@ual.es

Abstract

The operative configuration of the heliostat field of solar central receiver plants is a vital part of their controlling tasks. The subset of active heliostats must be carefully configured to set the operational state as desired while also avoiding dangerous flux distributions and radiation peaks over the receiver surface. In this context, a general and automatic aiming methodology is being developed by the authors of this work. However, the mathematical formulation of this problem leads to a complex large-scale optimization problem in which every active heliostat requires a certain two-dimensional aiming point over the receiver. In this work, the possibility of applying TLBO, a population based large-scale optimizer, is studied. Considering the potential computational costs of this task, a preliminary parallel version of TLBO has been developed. The application of this method to perform a large exploration of the search-space, in a high-performance computing environment, is described. The parallelization of the algorithm turns out to be quite useful to accelerate the procedure for the problem at hand. Therefore, the possibility of including additional steps to the method remains feasible.

Key words: Parallel computing, large-scale optimization, TLBO, Heliostat aiming

1 Introduction

Solar Central Receiver Systems, SCRS in what follows, are power generation facilities based on the exploitation of solar energy by concentrating the incident radiation. In general terms, considering the scope of this work, they are formed by a large group of high-reflectance orientable mirrors and a radiation receiver on the top of a tower. The mirrors, which are

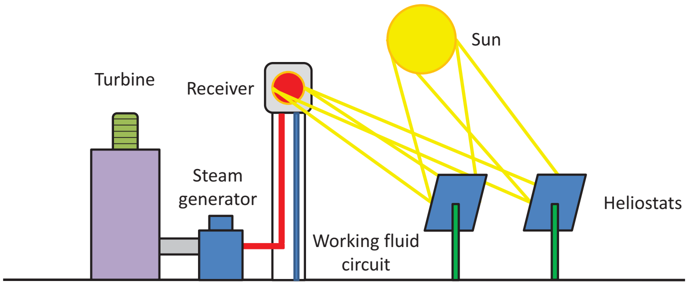

Figure 1: Scheme of a solar tower power plant.

called ‘heliostats’, track the apparent movement of the Sun through the day to concentrate the incident solar radiation over the receiver. Consequently, there is a very high radiation density over its surface. This power is then transferred to a working fluid which is in circulation inside the receiver. After increasing its temperature, this fluid can be finally used in a classic thermodynamic cycle for electric power generation. In Fig. 1, an illustrative schema of this kind of facilities is shown. Stability of production and operative efficiency due to the maturity of the underlying technologies are key aspects of this kind of power facilities. The interested reader is referred to [2, 9] for further information about them.

Controlling the flux distribution formed by the heliostat field over the receiver surface is of major importance to avoid dangerous temperature gradients, thermal stress and premature aging of its components [1, 3, 7, 11]. This is a key factor for increasing the operative life of the receiver, what has a direct influence on the production costs of STTP as highlighted in [7]. Considering that the heliostat field is usually formed by hundreds (occasionally thousands) of heliostats, the definition of the active subset of them, as well as their specific aiming point over the receiver according to the desired flux distribution, leads to face a very complex multi-staggered problem. In [3, 7], this problem is addressed by a pre-defined set of heliostats and possible aiming points, looking for an homogeneous flux distribution, with good results. However, in the context of this work, a generalization of their approach is being developed by trying to automatically configure the whole field for a given instant of time (i.e., solar position) and a desired flux distribution to achieve. This methodology would even include the possibility of disabling unnecessary heliostats to replicate the given reference (as heliostat fields are commonly oversized to face unfavorable operating conditions such as cloudy days). In any case, that step is out of the scope of the present paper and it will be assumed that the selected heliostats are already known. At this point, good overall results are obtained by applying gradient-based local optimizers to define their corresponding aiming point. Unfortunately, these approaches have a local scope and the objective function is known to have multiple local optima. It is intended to add a computationally efficient global optimizer to the process to get a wide perspective of the problem. An existing large-scale-oriented population-based global optimizer, the Teaching-Learning-Based algorithm (TLBO) [5], will be considered. Consequently, the deployment of this method in a high-performance computing environment is the main aim of this work.

In Section 2, the problem at hand is formally described. Then, in Section 3, the TLBO algorithm is exposed. In Section 4, the selected parallelization strategy for the algorithm is commented. Finally, experimentation and results are shown and conclusions are drawn in Sections 5 and 6 respectively.

2 Problem definition

In order to model the present problem, it is necessary to define the target flux distribution to achieve, F. It is a matrix of size $Y \times X$, where X and Y are referred to the number of rows and columns of the matrix respectively. All the elements of F are known in both position and magnitude (flux density) as it is part of the information of the problem. This matrix can be seen as a 'picture' of the flux distribution to replicate over the receiver, which is considered as a flat rectangle, with the active heliostats. Its dimension Y is linked to the vertical of the receiver plane, the direction from the plane towards the zenith in a three-dimensional Cartesian system. Similarly, its dimension X is linked to the horizontal direction of the receiver plane along the West-East direction. In this context, the discretization axes are also known as vectors $Y' = y_0, \cdots, y_Y$ and $X' = x_0, \cdots, x_X$ whose length is Y and X respectively. Consequently, every element of the matrix is referred to a particular zone of the receiver, whose surface is intrinsically discretized by the step used when defining F.

In relation to the active heliostats, it is an ordered set $H = \{h_1, h_2, \cdots, h_T\}$ with cardinality T. Every heliostat h_i projects a certain flux distribution f_{h_i} over the receiver when it is operative, which is a known bi-dimensional continuous function of the radiation density. It is defined as a bi-dimensional Gaussian density function as shown in Eq. 1

$$f_{h_i}(x, y) = \frac{P}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{\left(-\frac{1}{2(1-\rho^2)}\left(\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\right)\right)} \tag{1}$$

where x and y are the coordinates on the plane defined by the receiver rectangular aperture in its X and Y dimensions respectively, P is the power contribution of the heliostat h_i over the receiver, ρ is the correlation between x and y, σ_x and σ_y are the standard deviation along x and y respectively and μ_x and μ_y, which are the mean in the Gaussian probability function, define the central point of the flux distribution, i.e. the aiming point of the heliostat h_i. This approach is similar to the one selected by [3, 7], where a specific circular Gaussian density function is applied according to the HFLCAL model [8]. All the parameters that define the shape of the flux distribution of every heliostat, P, ρ, σ_x and σ_y, are known (from detailed simulation and curve-fitting procedures) while its central point (μ_x, μ_y) should be

 ISBN: 978-84-608-6082-2

determined to replicate the reference. Once every heliostat has a certain aiming point, the configuration vector of the field c can then be defined by concatenating the pair of aiming coordinates of every heliostat. Finally, the obtained flux distribution, F_c^*, is formed then by the convolution of the particular flux distribution of each one, over the receiver plane, along the discretization axises.

Taking into consideration the previous definitions, a $2T$-dimensional minimization problem can be formulated as the accumulation of the square difference, at every discretization point, between the matrices F and F_c^* as shown in Eq. 2.

$$min\, O = min \sum_{x=x_0}^{x_X} \sum_{y=y_0}^{y_Y} \left(F(x,y) - F_c^*(x,y)\right)^2 \tag{2}$$

3 Teaching-Learning-Based Optimization

Teaching-Learning-Based Optimization, TLBO, is an stochastic population-based global optimizer presented in [5]. The algorithm models the behavior of a class of students, who form the population of candidate solutions. They are progressively improved by simulating both the teaching process of the teacher of the class and the interaction between the students. This algorithm is mainly characterized by its performance and its virtual lack of specific parameters, as only the population size and the number of cycles need to be specified by the user. Although the original form of the algorithm for continuous non-constrained problems has been selected for this work, the interested reader is referred to [6] for further information about it, its versions and applications. The linked works of [4] and [10] should be also examined by the reader interested in TLBO.

Students, i.e. candidate solutions, are defined as N-dimensional vectors, where N is the number of dimensions of the studied problem. Every dimension is considered as a 'subject' in the context of TLBO, and the natural representation of the class is a $P \times N$ matrix in which P is the population size. The quality of the class is assumed to follow a normal distribution whose average value must be increased by academic interaction. Consequently, the value of students at every subject is altered along the cycles in order to improve the overall average as desired. To achieve it, the plain TLBO algorithm relies on two stages per cycle, the Teacher and the Learners phases [5], which are summarized next for a certain cycle k from a minimization perspective.

3.1 Teacher Phase (TS)

At this step, the average value per subject, i.e. per column, is calculated from the current population. An N-dimensional vector, M, is obtained. Then, the best student, whose value according to the objective function should be the minimum, is promoted to become the teacher of the cycle. A random integer in $[1, 2]$ called 'Teaching Factor', T_F, is generated

 ISBN: 978-84-608-6082-2

as an overall weighting factor of its teaching capabilities. After that, an N-dimensional random vector r of real values in $[0, 1]$ is generated to model the skills of the teacher at teaching every subject and the aptitudes of the students while learning them. With this information, an N-dimensional global shifting vector DM is computed according to Eq. 3, where T is the content of the teacher as a candidate solution. It must be noted that Eq. 3 is referred to dimensional element-by-element operations. Finally, DM is added as a vector to every existing individual. However, only improved students, after the evaluation of the objective function, are kept, while the change is discarded otherwise.

$$DM = r(T - T_F M) \tag{3}$$

3.2 Learner Phase (LS)

At this step, an N-dimensional random vector r of real values in $[0, 1]$ is generated to model the advancing possibilities at every subject in a similar way as done for the TS. Then, every final individual i from the previous step is randomly paired with another one j different from itself. After that, the individual i is shifted from its current position depending on whether i has a better value of the objective function than j or not as noted in Eq. 4, where the dimensional element-by-element operation scheme is maintained. Finally, the individual i is only updated when its value has been improved after the change.

$$Student_i = \begin{cases} Student_i + r(Student_j - Student_i), & \text{if } j \text{ better than } i \\ Student_i + r(Student_i - Student_j), & \text{if } i \text{ better than } j \end{cases} \tag{4}$$

At this point, the population of the next cycle would have been defined. However, an additional step not commented in [5] may should have been included to remove duplicate solutions, what is highlighted in [4] and addressed in [10]. This procedure is expected to look for equal solutions and to randomly re-initialize a dimension of one of them, what requires its re-evaluation as candidate solution.

4 Parallelization strategy

Considering the described structure of TLBO, and assuming a computationally demanding objective function, the underlying iterative structures linked to both TS and LS (updating and studying every student) could be assigned to different execution units. By proceeding this way, the whole procedure would be the same but, at every stage, the management of the available individuals would be distributed between execution units. Consequently, the required evaluations of the objective function would be directly shared between the available execution units. This is the selected approach for the present work in a thread-based environment. Its efficiency is linked to the number of individuals and the intrinsic cost of the evaluation of the objective function. However, it does not depend on the number of cycles.

Finally, it is important to mention that for extremely large populations and/or hard objective functions, the previous strategy could be easily generalized. Particularly, the population could be divided in different subsets that would be internally altered and evaluated also in parallel. It would be suitable for a hybrid process-thread-based environment when being able to amortize the communication costs.

5 Experimentation and results

A sample instance of the defined problem aims to form an homogeneous flat form over the receiver. Consequently, matrix F is formed by a single and replicated value: 80 kW/m^2 over a 6×6 meters receiver. The subset of heliostats to activate is adequately selected by our current procedure, which selects 110 heliostats to deploy. In this context, the developed parallel TLBO implementation will be launched with different configurations to check its computational performance. It has been implemented in C with OpenMP directives for threading purposes.

The execution platform is a cluster node featuring an Intel Xeon E5 2650v2 with 16 cores and 128 GB RAM. The number of cycles of TLBO has been fixed to 150 after preliminary adjustment. In relation to the number of individuals, what sets the direct computational load of every cycle, it has been configured to be 50, 100, 200 and 400. Considering the hardware, the number of threads has been fixed to 2, 4, 8 and 16 for all cases.

In Fig. 2, the speedup achieved with the parallel version of TLBO is shown for the different population sizes. Additionally, a black dotted line represents the theoretical linear speedup. These results have been averaged after five executions. As can be seen, the speedup is almost linear for all the instances. In fact, with 2 and 4 threads, it could be considered linear for all the population sizes. The peak of performance is achieved with the largest population and 16 active threads, where the speedup is 14.10. However, as expected, it is slightly worse when the population size is reduced and the number of active threads is too high. In other words, the speedup is progressively separated from linearity when the ratio between active threads and individuals is reduced. Additionally, considering the monotonic ascending tendency in all cases, the scalability of the process can be also highlighted.

Finally, in Fig. 3, the corresponding flux distribution of the best solution found by TLBO is shown. It has been obtained with a population of 400 individuals along 150 cycles of search. Its top is relatively flat and near 80 kW/m^2 as intended. Consequently, the algorithm is not simply compatible with an efficient parallel execution, but its results seem to be promising to be used as the initial point of further local gradient-based optimizers.

 ISBN: 978-84-608-6082-2

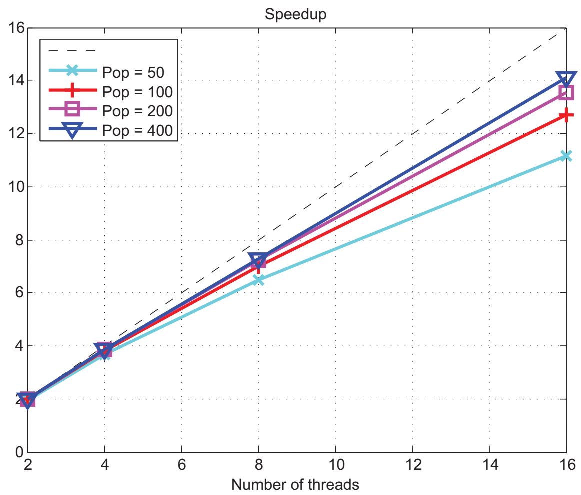

Figure 2: Achieved speedup with the thread-based parallel version of TLBO.

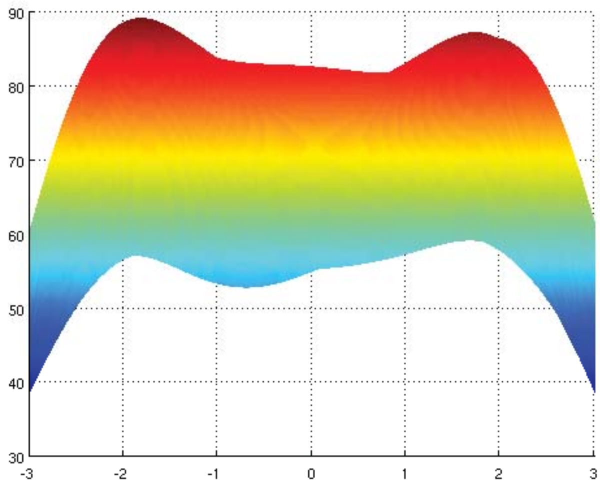

Figure 3: X-Z plane of the obtained solution by TLBO when reproducing a flat distribution.

 ISBN: 978-84-608-6082-2

6 Conclusions and future work

In this work, the problem of defining the aiming points of a set of heliostats has been presented. Then, it has been formalized from a mathematical perspective as an unconstrained large-scale minimization problem. It is known to have numerous local optima and a complexity which is increased with the number of active heliostats. Therefore, an existing population-based global optimizer, TLBO, has been selected to be studied for efficient vast explorations of the search-space. Considering the computational cost of the objective function and the necessity of handling large populations, an OpenMP-based TLBO implementation has been developed. It shows an almost-linear speedup and an scalable behavior with acceptable results. Consequently, it seems to be adequate to be used as the global guide of further local optimization stages.

As future work, taking into account the positive results, the inclusion of the parallel TLBO optimizer in our automatic heliostat aiming procedure will be considered in depth. Additionally, the possibility of deploying a hybrid parallel version of TLBO that can be executed in different nodes of a cluster will be addressed.

Acknowledgements

This work has been funded by grants from the Spanish Ministry of Economy and Competitiveness (TIN2015-66680-C2-1-R and ENERPRO DPI 2014-56364-C2-1-R), Junta de Andalucía (P11-TIC7176 and P12-TIC301). Nicolás Calvo Cruz is supported by a FPU Fellowship from the Spanish Ministry of Education. Juana López Redondo and José Domingo Álvarez Hervás are fellows of the Spanish 'Ramón y Cajal' contract program, co-financed by the European Social Fund.

References

[1] M. Carasso and M. Becker. Solar Thermal Central Receiver Systems: Performance evaluation standards for solar central receivers (Vol. 3). *Springer Verlag*, 1991.

[2] O. Behar, A. Khellaf and K. Mohammedi. A review of studies on central receiver solar thermal power plants. *Renewable and Sustainable Energy Reviews*, 23:12–39, 2013.

[3] S.M. Besarati, D.Y. Goswami and E.K. Stefanakos. Optimal heliostat aiming strategy for uniform distribution of heat flux on the receiver of a solar power tower plant. *Energy Conversion and Management*, 84:234–243, 2014.

[4] M. Crepinsek, S.H. Liu and L. Mernik. A note on teachinglearning-based optimization algorithm. *Information Sciences*, 212:79-93, 2012.

[5] R.V. Rao, V.J. Savsani and D.P. Vakharia. Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information Sciences*, 183(1):1-15, 2012.

[6] R.V. Rao. *Teaching Learning Based Optimization Algorithm: And Its Engineering Applications. Springer*. 2015.

[7] A. Salomé, F. Chhel, G. Flamant, A. Ferrière and F. Thiery. Control of the flux distribution on a solar tower receiver using an optimized aiming point strategy: Application to THEMIS. *Solar Energy*, 94:352–366, 2013.

[8] P. Schwarzbözl, M. Schmitz and R. Pitz-Paal. Visual HFLCAL - A Software Tool for Layout and Optimisation of Heliostat Fields. In SolarPACES'09, 2009.

[9] W. Stine and M. Geyer. Power from the Sun, 2001. Public website, available from http://powerfromthesun.net/book.html (Last access: May 10, 2016)

[10] G. Waghmare. Comments on 'A note on teaching-learning-based optimization algorithm'. *Information Sciences*, 229:159-169, 2013.

[11] C.J. Winter, R.L. Sizmann and L.L. Vant-Hull. Solar power plants: fundamentals, technology, systems, economics. *Springer-Verlag New York*, 1991.

  ISBN: 978-84-608-6082-2