Optimization of the Sizing of a Solar Thermal Electricity Plant: Mathematical Programming Versus Genetic Algorithms

José M. Cabello, José M. Cejudo, Mariano Luque, Francisco Ruiz
University of Málaga
Campus El Ejido s/n, 29071 Málaga, Spain
jmcabello@uma.es, jmcejudo@uma.es, mluque@uma.es, rua@uma.es

Kalyanmoy Deb and Rahul Tewari
Department of Mechanical Engineering
Indian Institute of Technology Kanpur, PIN 208016, India
deb@iitk.ac.in, rtewari@iitk.ac.in

KanGAL Report Number 2008008

Abstract— **Genetic algorithms (GAs) have been argued to constitute a flexible search thereby enabling to solve difficult problems which classical optimization methodologies may find hard to solve. This paper is intended towards this direction and show a systematic application of a GA and its modification to solve a real-world optimization problem of sizing a solar thermal electricity plant. Despite the existence of only three variables, this problem exhibits a number of other common difficulties — black-box nature of solution evaluation, massive multi-modality, wide and non-uniform range of variable values, and terribly rugged function landscape – which prohibits a classical optimization method to find even a single acceptable solution. Both GA implementations perform well and a local analysis is performed to demonstrate the optimality of obtained solutions. This study considers both classical and genetic optimization on a fairly complex yet typical real-world optimization problems and demonstrates the usefulness and future of GAs in applied optimization activities in practice.**
Keywords: Solar thermal electricity plant, optimization, genetic algorithms, classical optimization, multi-modality, noisy objective function.

I. Introduction

Energy is directly related to sustainable human development. Energy consumption affects social aspects (2 billion people have not access to modern energy supplies), damages human health and alters the atmosphere causing the global warming. All the energy sources came from the sun, directly or indirectly. Nowadays, there exist many technologies that use this enormous source of energy. Among them, solar thermal electricity is a very promising one that will contribute significantly to increase the electricity generation by renewable sources. In [7], a review of the solar thermal electricity technology can be found.

Obviously the main problem of the extension of thermal solar plants is the cost. They require very high inversions and the electricity production cost is lower in conventional fossil fuel plants (if no internationalization of the external costs is performed). This paper analyses the optimal sizing of a DSG solar thermal electricity plant that is promoted by the private firms Endesa (`http://www.endesa.es/Portal/en`) and Solar Millenium (`http://www.solarmillenium.com`) in the framework of a collaborative project between German and Spanish enterprises and public research centers. The project is called GDV-500 Plus.

From the mathematical point of view, we want to determine the optimal size of the main components, in order to maximize the expected annual profits of the plant. To this end, an optimization model has been built. Other examples of mathematical programming models for this kind of problems can be found in [3], where an integer optimization problem is built to determine the equipment operating configuration of a central energy plant, and in [10], where an optimization model is presented that defines a multi-auction capacity allocation strategy which is optimal with the explicit representation of uncertainty. The main problem that we face in our particular model is the fact that, due to the complex nature of the system and to legal regulations, the profits cannot be expressed as an explicit function of the decision variables. Rather than that, the profit function takes the form of a black box, which has been modeled as an evaluation subroutine. This subroutine takes into account all the technical and legal requirements, in order to determine the working strategy of the plant and, as a result, the annual profits. The problem is that the function implicitly defined by the subroutine, due to the nature of the process modeled, is not even continuous and it has many local optima. This has made it impossible to solve the problem using traditional optimization solvers (even able of handling non-convex global optimization problems), and this is the reason why we have chosen to use a genetic approach.

The reminder of this paper is organized as follows. In section II, the problem is described and modeled. In section

Kalyanmoy Deb is a Finland Distinguished Professor at the Department of Business Technology, Helsinki School of Economics, FIN 00101, Helsinki, Finland (Kalyanmoy.Deb@hse.fi).

III, we report the attempts to solve the model using two well known global optimization solvers, and we state possible reasons for their failure. In section IV, solutions obtained through a real-coded genetic algorithm is described, followed by that through a modified approach. Some final remarks are given in section V, and the paper ends with some conclusions thereafter.

II. Description of the model

A. The DSG solar plant

Figure 1 shows the elements of the solar plant. In the solar field, solar radiation is converted into heat. The condensate that comes from the block of power (BOP) increases its temperature and pressure and it is again suitable to produce work. When the radiation level is insufficient to produce the required mass flow of steam, a thermal storage and auxiliary power system are disposed in parallel to produce the supplement energy to the BOP. Thermal storage is designed to collect energy during daylight and dispatch when necessary. This system increases the number of hours of operation of the plant. The auxiliary system is a gas boiler that is designed to maintain a minimum temperature in the plant in order to reduce start up periods, and to contribute to electricity generation.

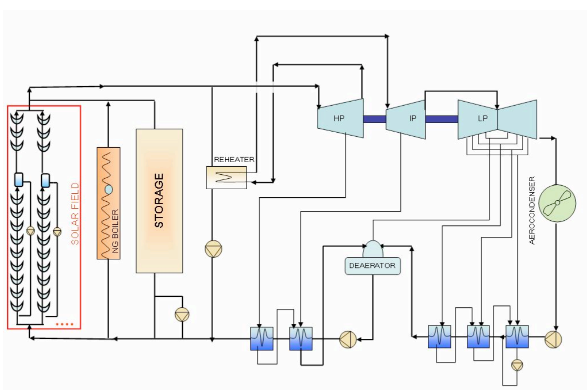

Fig. 1. DSG solar plant modeled.

Therefore, there are three main quantities to be dimensioned in the optimization process: the solar collector area, the storage capacity and the power of the auxiliary boiler.

B. Main assumptions

Due to the complex technical limitations of the plant, and in agreement with the organizations participating in the study, the following assumptions have been made on the component systems of the plant and on the operation strategy.

With respect to the solar collector field, it uses direct solar radiation. The steam mass generated has been considered to depend only on the direct solar radiation received. Therefore, based on a file of expected hourly solar radiation for the whole year, the steam mass flow produced per square meter at the solar field has been determined, and these data are used to feed the evaluation subroutine. Due to technical reasons, the maximum size of the solar field has been set to $750000\,\mathrm{m}^2$.

TABLE I

Decision variables of the model.

Variable	Description	Unit
A_C	Solar collector field size	m^2
E	Storage capacity	kJ
P_{AUX}	Power of the auxiliary boiler	kW

The capacity of the storage is measured in terms of the number of hours that the tanks can provide the energy necessary to drive the block of power. But a tank cannot be arbitrarily large. Therefore, whenever a tank reaches a maximum possible capacity (equivalent to 8 hours of storage), a new tank has to be built. This causes discontinuities in the costs function, given that every 8 hours of storage, the cost is incremented in 15 million € (the fixed cost of building a new tank). On the other hand, in order to account for ambient losses, the energy flow coming from the storage is multiplied by 0.9 if one tank is used, by 0.85 if two tanks are used, by 0.8 if three tanks are used, and so on.

On the other hand, the operation strategy affects the optimal size of the components of the solar plant. In this paper, the operation strategy has been defined in order to reproduce the complexity of the problem. The strategy defined is based on experience of operation of this kind of plants. The operation for each hour can be summarized as follows

1) Evaluate direct solar radiation and calculate the mass steam production with the collector field model.
2) If the mass flow is enough to activate the plant to at least a 75% of the power (load fraction), the plant is producing electricity just with solar energy. If the mass flow exceeds the necessary amount for a 100% charge, the remaining energy is stored.
3) In the case that the steam mass production does not reach the minimum value fixed before, the storage complements the required energy. This is only possible if there is enough energy already stored.
4) When the steam mass cannot be obtained with the solar collector field and the storage, the auxiliary boiler supplements the rest. Due to legal regulations, the overall yearly operation of the auxiliary boiler is limited to 15% of the net electricity production of the plant.
5) The collector field charges the storage system during daylight if 75% of the gross power of the plant cannot be obtained with the previously described scheme.

Taking these assumptions into account, the model has been built as follows.

C. The Optimization model

As previously mentioned, the decision variables of the model are the sizes of the three main components of the central, as displayed in Table I.

Making use of these variables, the (apparently simple) optimization problem to be solved is:

TABLE II

OPERATION STRATEGY RELATED VARIABLES (HERE, $i = 1, \ldots, 8760$).

Variable	Description	Unit
E_i	Energy stored after hour i	kJ
$FUNC_i$	Load fraction of hour i	%
$EAUX_i$	Energy generated by the auxiliary system in hour i	kJ
$PERC_i$	Accumulated percentage of energy generated by the auxiliary system until hour i	%

$$\begin{aligned} \text{maximize} \quad & P(A_C, E, P_{AUX}), \\ \text{subject to} \quad & 0 \leq A_C \leq 750000, \\ & 0 \leq E, \\ & 0 \leq P_{AUX}, \end{aligned} \tag{1}$$

where P is the profit function. Broadly speaking, $P = I - C$, where I are the expected incomes obtained by selling the electricity, and the costs C include installation, maintenance, fuel, insurance and contingency costs. As previously mentioned, the problem is that P does not have an explicit mathematical form as a function of the decision variables. In order to evaluate P for each value of the decision variables, the following subroutine (which contains all the assumptions described in section II-B) must be run.

D. Evaluation subroutine

In this section, we will outline the main steps of the evaluation subroutine, which has been implemented in C++ language, in order to compile it together with the solver. This way, the subroutine is called every time the solver needs a function evaluation. In summary, once the values of the decision values are set, the subroutine determines the operation strategy of the plant for each of the 8760 hours of the year, and the profits (incomes and costs) are obtained accordingly. Therefore, apart from the value of the profit function P, the subroutine creates a series of variables that define the operation strategy, as displayed in table II. Variable $FUNC_i$ indicates the load fraction at hour i, and thus it can be equal to 0 if the system does not work, or any value between 75 and 100.

Let us now describe the evaluation subroutine step by step. Let us assume that certain values of the decision variables, A_C, E and P_{AUX} are given. Then, we proceed in the following way.

1) **Initial calculations.** Given the value of E,
   a) Calculate the number of tanks to be installed, by dividing E by the maximum capacity of a tank.
   b) Determine the performance of the tanks, which depends on the number of tanks installed, as described in section II-B.
   c) The number of tanks also influences the amount of soil that has to be used for the plant. Namely, for any new tank starting from the third one, a supplementary amount of soil has to be considered.

2) **Operation loop**. The operation strategy has to be determined now, according to points 1–5 of section II-B. Namely, for each hour of the year, we determine the load fraction of the plant, in the following way.
   a) The direct solar radiation of hour i is read from the data file, and the steam mass per square meter is calculated accordingly. This value is multiplied by A_C to obtain the total steam mass of the hour.
   b) If the steam produced at the solar field is enough for a 100% charge, $FUNC_i$ is given the value 1 (100%), and the remaining energy is added to the previously stored amount, and accounted for in variable E_i. This value can never exceed the total storage capacity given by the decision variable E. The auxiliary system is not used.
   c) If the steam mass provides a charge between 75% and 100%, the plant works at the highest possible charge percentage (this is the value given to $FUNC_i$), with no aid from the storage or from the auxiliary system.
   d) If the steam mass generated at the solar field does not suffice for a 75% charge, then several situations can occur:
      i) If there is enough energy stored to reach the 75% charge, then the necessary amount is taken from the tanks, E_i is actualized accordingly, $FUNC_i$ is set to 0.75, and the auxiliary system is not used.
      ii) If there is not enough energy stored, we need to complement the rest from the auxiliary system. In order to do this, the two following conditions must hold:
         - The installed capacity of the auxiliary system (given by decision variable P_{AUX}) must be enough to produce the required energy.
         - The accumulated (up to hour i) percentage of energy supplied by the auxiliary system cannot exceed the limit (15%).

         If any of these two conditions fail, then the system does not work at hour i. Therefore, the energy produced at the solar field is stored, E_i is actualized accordingly, and $FUNC_i$ is set to 0.

         If the two conditions hold, then the storage is emptied ($E_i = 0$), the value of $EAUX_i$ is the energy supplied by the auxiliary system at this hour, and $FUNC_i$ is set to 0.75.
   e) The accumulated hybridization percentage $PERC_i$ is actualized, depending on the values of $EAUX_i$ and $FUNC_i$.
   f) The incomes corresponding to the hour i are calculated according to the value of $FUNC_i$ and to the selling price p_i.

      Once these calculations are completed, the subroutine goes back to point a) for the next hour.

3) **Final calculations**. Once the hourly loop is completed, the profit function is calculated in the following way:
   a) The global yearly incomes are calculated as the sum of the 8760 hourly incomes.
   b) The installation costs comprise the costs of the solar panels, the tanks, the auxiliary system (these three depend on the values of the decision variables), the block of power and the soil. All these costs are annualized for given life cycle and discount rate.
   c) Annual maintenance costs are assumed to be a fixed percentage of the total installation costs.
   d) The fuel cost (for the auxiliary system) has a fixed monthly component and a variable component which depends on the corresponding values of $EAUX_i$.
   e) Insurance and contingency costs are also assumed to be fixed percentages of the total installation costs.

The global scheme of the subroutine can be seen in the flowchart displayed in Figure 2.

III. Difficulties of the problem

At a first stage, we made an attempt to solve the problem with traditional global optimization solvers. To this end, both the evaluation subroutine and the main module have been implemented on a personal computer, in C++ language, using the Microsoft Visual C++ compiler. Both the entry data and the solution output are provided by simple text files, which can be automatically converted into Microsoft Excel files. The module was prepared to call an optimization solver. In this attempt, two global solvers have been used.

The LGO software (Library of Global Optimization), developed by Pintér Consulting Services [8], has been designed to solve general nonlinear optimization problems, including problems with multiple optima, and it has been successfully tested with many real non-convex problems. It combines local and global optimization strategies, with or without information about derivatives, with deterministic or stochastic parameters. Namely, the library incorporates the following solvers: A global branch and bound solver, a global adaptive random search solver, a multi-start global random search solver and a constrained local search solver based on the reduced gradient. For further details, see [8], [9].

On the other hand, the LINDO API system (`http://lindo.com/`) also includes global and local optimization solvers. Namely, there is a multi-start global optimization solver, a general non-linear solver, simplex based solvers for linear programming problems, an interior point solver for linear and quadratic programming problems and several solvers for mixed integer linear and non-linear problems. The global solver combines a series of range bounding and range reduction techniques within a branch and bound framework. The multi-start solver generates a sequence of candidate starting points in the solution space, while a traditional non-linear programming technique is used to find a local optimum for each starting point.

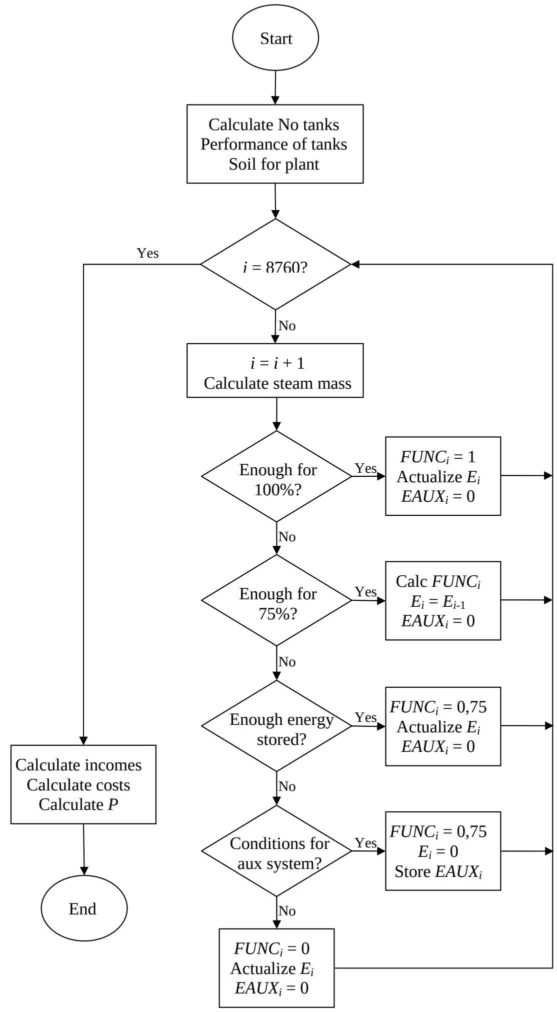

Fig. 2. Flowchart of the evaluation subroutine.

None of the two libraries succeeded in finding a consistently good solution for the problem. The different solvers used either got stuck in a local optimum (in the best case) or even produced solutions that were clearly outperformed by others. Moreover, certain applications from specific initial points have produced good intermediate solutions. When both solvers are restarted from these solutions, they did not result in acceptable solutions.

The possible reasons for the erratic behavior of the solvers are the following:

- The objective function is a black box, from which the solvers do not seem to be able to obtain useful information other than the value of the function for each point.
- The problem seems to have plenty of local optima.
- The function has lots of discontinuities, which cause significant "jumps" in the values of the profit function, thereby causing internal derivative computation (if required) problematic.
- The structure of the evaluation subroutine, basically

consisting on nested if-then commands results in a noisy behavior that misleads the solvers.

- Variables take widely different ranges of values, thereby making it difficult for the solvers to provide adequate emphasis to correct variable combinations. For the simplex search, this may come from the generation of a skewed (with a large aspect ratio) simplex.

In order to get an idea of the difficulty of the function landscape, we have created five sets of 10,000 random points and evaluated them. Table III presents the best solution and its function value among each one of these five sets. The best solutions are quite different from each other, thereby providing no clue about the possible good search regions in this problem.

TABLE III
COMPARISON OF GA-OPTIMIZED SOLUTION WITH FIVE SETS OF RANDOMLY CREATED SOLUTIONS.

Set	$P(\mathbf{x})$ (€)	$\mathbf{x}$		
		A_C (m^2)	E (kJ)	P_{aux} (kW)
1	27049676	693450	5405000000	174300
2	23514158	594975	2492000000	268100
3	22306797	655575	4981000000	535300
4	22996207	688200	2478000000	312300
5	28176740	712125	6490000000	98010

To get a more specific idea of the nature of the objective function, we compute the objective function for several values of the variable (A_C) in the range [700,000, 750,000] m^2 at a step of 10 m^2 and keep $E = 6,346,926,197.10$ kJ and $P_{AUX} = 92,768.3$ kW (which, as we will see later, are their optimal values). This produces 5,001 solutions in total. Figure 3 shows the variation of objective values with A_C in the above range. The inset figure clearly shows that the function has many local optimum and is also too noisy to compute gradients properly by using any computational method.

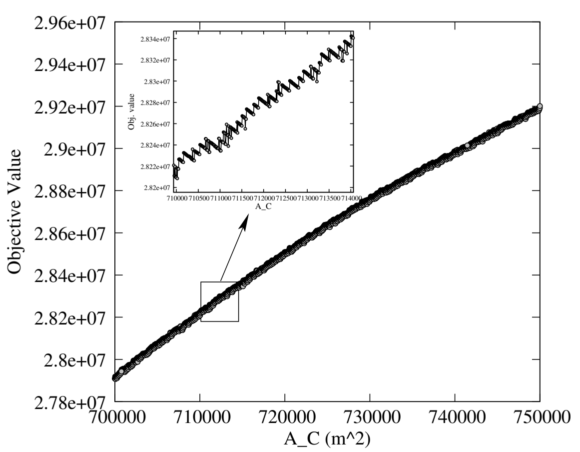

Fig. 3. Profit function variation with A_C reveals multi-modality, noise and jumps in the objective function ($E = 6,346,926,197, P_{AUX} = 92,768$).

IV. Genetic Algorithm as an Optimization Tool

Genetic algorithms (GAs) are population based optimization algorithms which do not use any gradient information [4], [6]. While dealing with practical problems having different complexities, such as noise, multimodality, numerical scaling of variables and others, many of which are prevalent to this problem, GAs have demonstrated their usefulness in the past. First, we apply a standard GA to the solar thermal electricity plant optimization problem and then discuss a modified approach.

A. A Standard GA

All variables of this problem are real-valued, thus we use a real-coded GA (RGA) for this problem. A C-code implementing RGA is available from website `http://www.iitk.ac.in/kangal/soft.htm` and is used here. The solution evaluation code supplied by the organization is compiled and linked with the compiled RGA code in a linux operating system. For evaluating a solution $\mathbf{x}$, RGA sends the variable vector to the evaluation code which then returns the function value, $P(\mathbf{x})$, of the supplied solution vector. Figure 4 shows a schematic diagram of the linking procedure of RGA with the solution evaluator. Starting with a set of population members, RGA works in iteration by creating new solutions which get evaluated by the solution evaluator. The optimized solution is then printed.

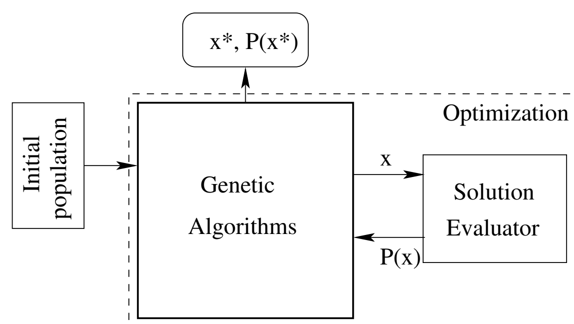

Fig. 4. The linking of an existing GA code with a solution evaluator.

RGA uses binary tournament selection, simulated binary crossover (SBX) [2], and a polynomial mutation operator [1]. A population of size 50, a crossover probability of 0.9 with SBX index of 2, a mutation probability of 1/3 with index 10 are chosen. The GA is run for 150 generations. These are standard values suggested in previous studies. To initialize the GA population, we use the following artificial upper bound for variables E and P_{AUX}: $E \le 10^{20}$, $P_{AUX} \le 10^{20}$. We obtain the following solution ($\mathbf{x} = (A_C, E, P_{AUX})^T$):

$$A_C = 749,980.86 \text{ m}^2, \quad E = 6,191,823,943.05 \text{ kJ},$$
$$P_{AUX} = 92,898.24 \text{ kW}, \quad P(\mathbf{x}) = 29,189,994.89 \text{ €}.$$

First of all, we observe that our chosen artificial upper bounds on E and P_{AUX} did not influence the obtained solution. Secondly, the optimized value of A_c is very close to the supplied upper bound of $750,000$ m^2. Thus, if this upper bound can be increased, a better objective value is expected. Figure 5 shows how the population best and average objective values improve with generation number. The initial population best

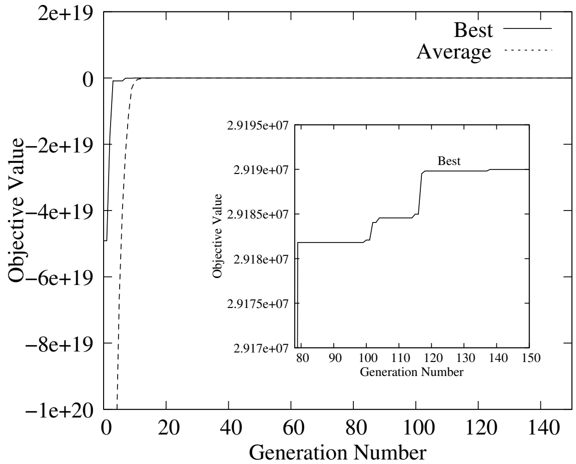

Fig. 5. Population best and average objective function values with generation number.

solution has an objective value $-4.9106e19$ (a negative profit due to income (I) being smaller than cost (C)). At generation 56, the population-best solution becomes positive for the first time, and then keeps on improving with generation before stabilizing to its converged value $2.9189e07$. Thus, the GA is able to make a significant progress from a very large negative value to a very high positive value in a span of only about 120 generations. The inset plot in Figure 5 shows the detailed progress of the algorithm after the 80-th generation. GA seems to progress steadily with generation finally reaching the optimized value at generation 138. Note that the GA took a total of 50×150 or $7,500$ solution evaluations. It is also interesting to note that the solution obtained by GA with 7,500 solution evaluations is better than the best solution found by a random selection of 10,000 solutions, as reported in Table III.

From the GA-optimized solution, we make another interesting observation. Each variable takes quite a different order of magnitude. The supplied and chosen variable bounds were quite large, thereby making any optimization algorithm difficult to focus near the true optimum in this problem. To make the search more focused, we propose a modified GA with a continuously updated variable bound scheme.

B. A Continuously Updated Genetic Algorithm

In the modified GA, we run the above GA for 150 generations and note the best solution (say $\mathbf{x}^*$) found thus far. For each variable x_i, the population standard deviation σ_i is computed. Thereafter, for the next 50 generations (we call an epoch) we update the variable bounds as follows:

$$x_i^{(L)} = x_i^* - \sigma_i, \quad (2)$$
$$x_i^{(U)} = x_i^* + \sigma_i. \quad (3)$$

All the existing population members which are within the above variable bounds are accepted in the new population. The remaining population slots are filled by creating random solutions within the above variable bounds. This procedure is continued for every 50 generations until there is no difference in the best solutions of two consecutive epochs. This continuously updated variable bound procedure will allow a focused search and will allow the modified GA to converge to a solution with generations.

We use identical GA parameter values as before. The proposed GA runs for 650 generations before converging to the following solution:

$$A_C = 749,999.99 \text{ m}^2, \quad E = 6,346,926,947.98 \text{ kJ},$$
$$P_{AUX} = 92,768.27 \text{ kW}, \quad P(\mathbf{x}) = 29,201,019.61 \text{ €}.$$

This solution is slightly better (about 0.04%) than that obtained by our previous GA. The variable A_C reaches very close to its specified upper bound and other two variable values are also close to that found by previous GA procedure. Figure 6 shows the population best and average function value with generation. Since the range of objective values from initial generation to final generation is quite significant, the progress of the algorithm is not comprehensible from the overall plot. The inset figure shows the function values after generation 100 and the steady progress of the algorithm is clear from the figure.

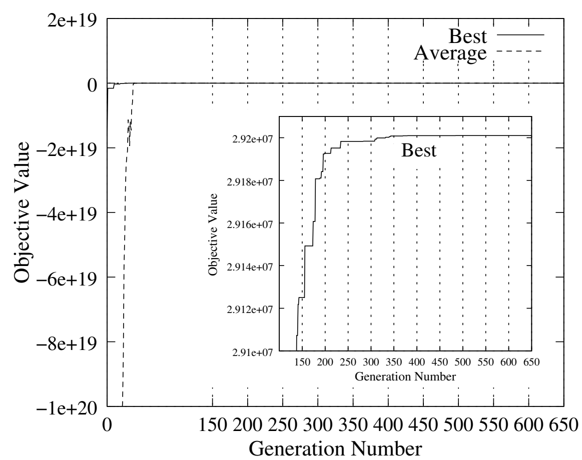

Fig. 6. Population best and average objective function values with generation number for the modified GA.

To investigate the accuracy of this optimized solution and to support its probable optimality, we compute the solutions in the vicinity of the optimized solution. For an analysis for the variable E, we fix $A_C = 749,999.99$ m^2 and $P_{AUX} = 92,768$ kW in their optimized values and vary E in [6.335e9, 6.350e9] kJ with an increment of 1000 kJ. This range is chosen around the optimized value of E. This resulted in 10,000 solutions and we plot the corresponding objective values in Figure 7. There are two distinct facts to be observed from this figure. First, the objective function seems to be quite sensitive to E and as

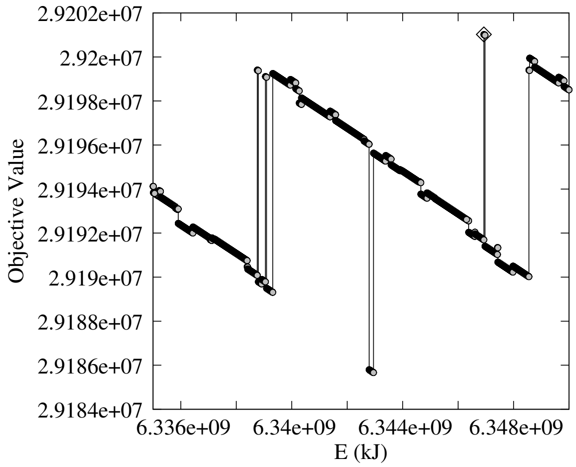

Fig. 7. An exhaustive search for variable E in the vicinity of GA solution confirms the accuracy of GA solution by the modified approach ($A_C = 749,999.99, P_{AUX} = 92,768$).

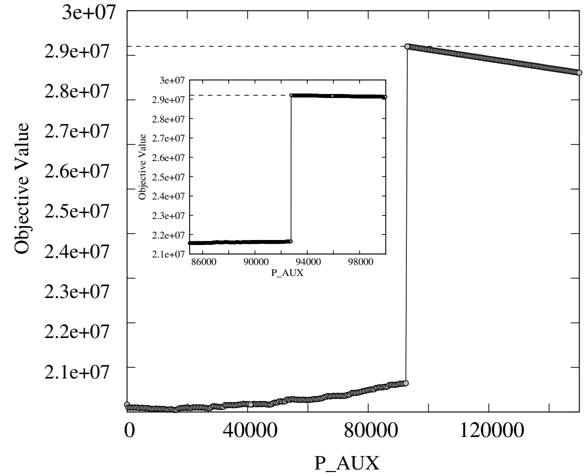

Fig. 8. An exhaustive search for variable P_{AUX} in the vicinity of GA solution confirms the accuracy of GA solution by the modified approach ($A_C = 749,999.99, E = 6,346,926,947.98$).

argued in section III the underlying objective function seems to be noisy in nature. It is not surprising that derivative based optimization methodologies faced difficulties in solving this problem. Secondly, our optimized solution seems to lie right on the best solution found in the above range of E. Interestingly, there are only a few good solutions in the vicinity of our optimized solution and most neighboring solutions are drastically worse than the optimized solution. Such difficulties are termed as 'isolation' elsewhere [5] and were studied using hand-crafted test problems. In this paper, we come across a real-world problem which clearly exhibits the presence of isolation near an optimal solution. In solving this problem, we faced difficulties with a couple of state-of-the-art classical optimization algorithms and simultaneously demonstrate the ease of solving the problem using a standard GA and a modification of it for a more reliable application.

We also compute and plot a wide range of values of P_{AUX} with fixed values of $A_C = 749,999.99$ m^2 and $E = 6,346,926,947.98$ kJ in Figure 8. The objective function seems to vary rather smoothly on this variable, except a sudden jump at $P_{AUX} = 92,768.27$ kW. The objective value seems to be largest at this P_{AUX} value, an increase of which reduces the objective value. Such a discontinuity right at the optimal value will cause many derivative and trust-region based optimization methods to fail. The inset figure shows that even in this case, the proposed GA is able to find the right optimal value. The GA-optimized function value, shown in a dashed line, is found to match with the best objective value of the plot, thereby indicating that the GA-optimized solution is optimal.

A similar observation is also found for the variable A_C in which the optimal value of A_C is at the upper boundary. Figure 9 indicates that the objective value has a noisy increasing trend with higher value of A_C and the inset figure shows that optimum lies at $A_C = 750,000$ m^2, which is found by the proposed GA. Many optimization algorithms will have difficulties in handling such noisy objective values and avoid all local optima and converge to the true global optimum.

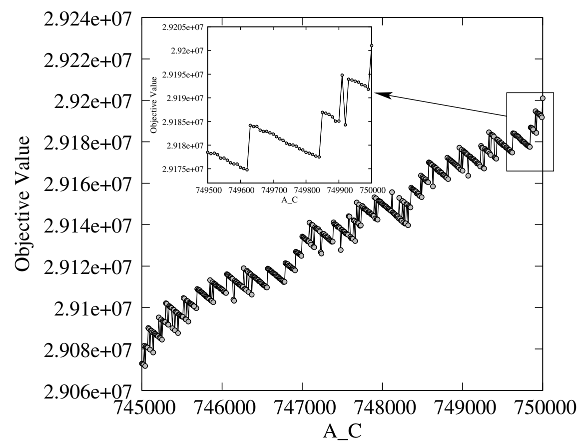

Fig. 9. An exhaustive search for variable A_C n the vicinity of GA solution confirms the accuracy of GA solution by the modified approach ($E = 6,346,926,947.98, P_{AUX} = 92,768$).

To investigate the robustness of the modified approach, we perform 10 different simulations starting from different initial populations. Table IV shows the best solution obtained in each simulation terminating at 650-th generation. The variation in obtained solutions are not much (0.0017%) difference between best and worst obtained solutions. Moreover, the corresponding variable values are also close to each other. These results indicate that the proposed GA procedure is robust for the present application.

V. Final Remarks

From the point of view of the plant, the optimal solution obtained can be interpreted as follows. First, the value $A_C =$

TABLE IV
BEST SOLUTION OBTAINED IN 10 RUNS OF MODIFIED GA APPROACH.

$P(\mathbf{x}^*)$ (€)	$\mathbf{x}^*$		
	A_C (m^2)	E (kJ)	P_{AUX} (kW)
29201019.61	749999.99	6346926947.98	92768.27
29201018.75	749999.94	6346929408.75	92768.26
29201018.62	749999.94	6346929669.93	92768.26
29201018.62	749999.94	6346929478.79	92768.27
29200967.67	749997.61	6347055515.19	92768.46
29200967.42	749997.62	6347055515.19	92768.46
29200957.23	749997.01	6347087739.59	92768.27
29200953.07	749997.00	6347092303.42	92768.31
29200922.51	749999.96	6346929711.66	92777.49
29200522.97	749992.62	6318673893.16	92768.70

$750,000\,\mathrm{m}^2$ is the bound imposed by the firm. In fact, in preliminary studies where we did not establish this limit, the ideal area was around $950,000\,\mathrm{m}^2$. Second, the value $P_{AUX} = 92,768.27\,\mathrm{kW}$ reflects the necessary power that can enable the plant to reach 75% of its full production, using exclusively the auxiliary system. Finally, the total storage capacity would be one full tank and around 90% of the second one. There are also two other remarkable data in the optimal solution. The sum over the whole year of the variables $FUNC_i$ equals 5,413.70 hours. This means that the plant is working 61.80% out of the 8,760 hours of the year. The final value $PERC_{8760}$ is 15%, that is, the legal hybridization limit is reached at the end of the year.

Having solved the problem using GAs and then providing justification for the optimality of the obtained solution, we have now understood various challenges provided by the three-variable optimization problem of sizing the solar thermal electricity plant. We outline them in the following:

- The objective function is noisy.
- The objective function has massive multimodality.
- The objective function has discontinuities.
- The optimal solution lies on a discontinuous point in the search space.
- The optimal solution lies on a variable boundary.
- The optimal solution is isolated and is surrounded by not-so-good solutions, resembling a local needle-in-haystack problem.
- The optimal decision variable values are of different orders of magnitude with a maximum difference of five orders of magnitude.

Any of the above challenges is difficult for most derivative and classical optimization methods. The combination of these challenges is even worse. The flexibility and global search perspective of GAs make them suitable for solving such problems. Finally, this problem indicates that a small sized problem (with only three variables in this problem) need not always be termed as an easy problem for an optimization algorithm. The function landscape provides a true picture of the challenges offered by a problem.

VI. CONCLUSIONS

Many optimization studies demonstrated in the literature usually involve smooth objective functions and well-scaled variables. However, the practice is far from being so ideal. In this paper, we come across a three-variable maximization problem which exhibits common complexities which many real-world optimization problems possess. Some of these difficulties are (i) black-box optimization, (ii) noisy objective function, (iii) massive multi-modality, (iv) non-uniform range of variable values, and (v) extremely wide range of search region. When attempted to solve using a couple of classical gradient based optimization techniques, the effort resulted in no useful solution due to the inflexibilities involved with the classical approaches in dealing with above difficulties.

On the contrary, with genetic algorithms, we have experienced a completely different outcome. A standard real-coded GA, starting from not-so-good initial random solutions, has been able to progress close to reasonably good solutions quickly by negotiating all the above difficulties. To make the performance better, we have also suggested a modified GA with a continuously updated search procedure. An analysis of solutions around the vicinity of the obtained solution has supported the optimality of the obtained solution.

This study clearly demonstrates the usefulness of genetic algorithms in practical optimization and flexibility of handling different vagaries of problem difficulties which often arise in applied optimization problems.

ACKNOWLEDGMENTS

This work has been done in the frame of the joint project "GDV-500 Plus" with the firm Endesa Generación S.L.

J.M. Cabello, M. Luque and F. Ruiz acknowledge the support from the Spanish Ministry of Education and Science (Research Project MTM2006-01921), and from the Andalusian Regional Ministry of Innovation, Science and Enterprise (PAI group SEJ-445).

K. Deb and R. Tewari acknowledge the support from Academy of Finland and the Foundation of Helsinki School of Economics.

REFERENCES

[1] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.

[2] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[3] R. D. Doering and B. W. Lin. Optimum operation of a total energy plant. *Computers and Operations Research*, 6(1):33–38, 1979.

[4] D. E. Goldberg. *Genetic Algorithms for Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[5] D. E. Goldberg. *The design of innovation: Lessons from and for Competent genetic algorithms*. Kluwer Academic Publishers, 2002.

[6] J. H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: MIT Press, 1975.

[7] D. Mills. Advances in solar thermal electricity technology. *Solar Energy*, 76:19–31, 2004.

[8] J. D. Pinter. *Computational Global Optimization in Nonlinear Systems. An Interactive Tutorial.* Lionheart Publishing, Inc., Atlanta, GA, 2001.

[9] J. D. Pinter. *Nonlinear Optimization with MPL/LGO: Introduction and User's Guide.* Maximal Software, 2006.

[10] C. Triki, P. Beraldi, and G. Gross. Optimal capacity allocation in multi-auction electricity markets under uncertainty. *Computers and Operations Research*, 32(2):201–217, 2005.