J. C. MOREIRA, E. MIGUEZ, C. VILACHÁ, Antonio F. OTERO

Department of Electrical Engineering, University of Vigo

A Parallel Branch and Bound Approach to Optimal Power Flow with Discrete Variables

***Abstract**. An optimal power flow (OPF) with discrete variables is a non-convex, nonlinear combinatorial problem. Usually the discrete variables present in an OPF are treated as continuous variables. The solution obtained using this method is clearly infeasible, but it is considered to be close to the discrete real solution and can be attained easily without producing an excessive degradation in optimality. These hypotheses can easily be refuted by demonstrating the need for a more robust general mechanism for treating the discrete variables in the OPF. Finding the exact solution is intractable due to the high computing cost it requires--this fact causes the heuristic techniques to be seen as a natural way to obtain good solutions quickly. This article presents an algorithm based on a branch and bound technique that, with the help of the parallel computing power a personal computer (PC) provides, allows pseudo-optimal solutions to be attained with good calculating times. The numerical results obtained by applying the technique proposed in IEEE networks of 118 and 300 nodes and a real size network derived from the Spanish transport network, demonstrate that the algorithm proposed has good execution times, provides solutions close to the optimum, and naturally manages the infeasibilities that are produced during the process.*

***Streszczenie.** Obliczenie optymalnego rozpływu mocy dla zmiennych dyskretnych jest problemem nieliniowym kombinacyjnym. Zwykle zakłada się, że zmienne dyskretne są traktowane jak zmienne ciągłe. Ta metoda uniemożliwia uzyskanie dokładnego wyniku w akceptowalnym czasie. Stąd powstało wiele technik heurystycznych, umożliwiających szybkie uzyskanie wyniku o dobrej dokładności. W artykule zaprezentowano algorytm na podstawie techniki podziału i ograniczeń, która, przy zastosowaniu obliczeń równoległych w nowoczesnych komputerach PC, pozwala na szybkie uzyskanie wyniku sub-optymalnego. Algorytm zastosowano do obliczeń sieci IEEE o 118 I 300 węzłach oraz rzeczywistej sieci, uzyskując krótkie czasy obliczeń i wyniki bliskie optymalnym. (**Algorytm równoległy podziału i ograniczeń ze zmiennymi dyskretnymi do obliczeń optymalnego rozpływu mocy**)*

Keywords: Optimal power flow, discrete control, interior point method, branch and bound, parallel computing.
Słowa kluczowe: Optymalny rozpływ mocy, sterowanie dyskretne, metoda punktu wewnętrznego, metoda podziału i ograniczeń, przetwarzanie równoległe

Introduction

Despite the major developments experienced in solving OPF problems since its inception [1] it still presents some challenges [2], one of which is treating naturally discrete variables associated with the modeling of the elements that make up the network, such as the positions of the taps in the transformers and the connection/disconnection of reactances.

The existence of discrete control variables converts an OPF into a nonlinear mixed integer combinatorial optimization problem. In this type of problem the search for the optimal solution is usually discarded in favor of looking for solutions close to the optimum, except in very small cases with little practical use, far from the sizes of the problems that are posed in real networks.

The algorithms most used to solve this problem type are heuristic in nature, and the fastest, and possibly the most widespread method for trying to find a solution to a problem with discrete variables consists of solving the OPF assuming all variables are continuous (COPF), and rounding them off to their nearest discrete values. This is the simplest and fastest method, which is why it should surely be the first test done on any other algorithm designed to solve discrete OPFs (DOPF).

Nevertheless, rounding off the variables can fail for two reasons: it can significantly degrade the result's optimality or be infeasible due to the combination of values assigned after rounding. For this reason, after its application, the values obtained must be checked to ensure they provide a feasible solution and sufficient optimality.

A slight loss in optimality can be accepted due to the impossibility of finding the global optimum when the number of discrete variables is high, or if the objective function is too flat-shaped near the optimum. To the contrary, infeasibility is a serious problem and it is necessary to provide a robust mechanism to the process of assigning values to the discrete variables, which allows a quasi-optimal feasible solution to be found. This need has fostered the publication of various algorithms which try to cover this aspect.

The method that is proposed in [3] is a rounding off step application. First, an OPF is solved with all the continuous variables. Next, a set of variables close to a discrete value is chosen, and they are rounded off to that position. With that set of variables already fixed, an OPF is executed again and the process is repeated again until there are no continuous variables left to be assigned.

The authors of [4] use a method for assigning the discrete variables based on a probability function definition of the interval defined by the discrete values immediately less than and greater than the value obtained as a solution of a COPF. First, the variables closest to the ends are assigned and then in subsequent iterations, the ones furthest from the ends are assigned. Only the shunts are considered as discrete variables.

In [5] they propose an algorithm where terms are added in the cost function that penalize the variables for their divergence from the discrete values. In other words, if x_c is the value of an x variable obtained as a result of a COPF, and x_i is one of the discrete values, an $M(x_c - x_i)^2$ term is added to the cost function, where M is a penalty factor.

In [6] and [7] algorithms based on ordinal optimization are presented. In the former, comparative computational effort values can be found among different heuristic techniques. In the latter, the algorithm divides the network into areas and then the ordinal optimization is applied to each one of those areas. Other techniques used are based on the sensitivity analysis [8] or on the cutting plane method applied to interior point algorithms [9].

In all the cases there is an obvious concern that exists over the capacity of the various methods to solve problems with an elevated number of binary variables, since the computing effort needed increases exponentially.

This article presents an algorithm to solve an optimal power flow problem with discrete variables. This algorithm is based on an enumeration process which uses a binary tree whose nodes are associated with subproblems constrained to continuous variables (COPF). This process is similar to a branch and bound (B&B) technique used to solve mixed-integer linear problems.

The most relevant aspects of this algorithm are: a strategy which allows managing the infeasible subproblems in a robust manner and a parallel implementation that allows accelerating the search when executing various COPF simultaneously in multiple core processors.

Optimal Power Flow Problem

An OPF with discrete variables can be written in the following manner:

$$\min_{c,d} f(c,d)$$
$$g(c,d)=0$$
$$\underline{h} \le h(c,d) \le \overline{h}$$

where

- *f(c,d)* is the objective function to be minimized that can be the minimization of losses, generation costs, etc.
- *g(c,d)* are the nodal equations of the network.
- *h(c,d)* are functions that represent constraints on the network variables.
- $\underline{h}$ and $\overline{h}$ are the minimum and maximum limits of the constraints.
- *c* and *d* are the continuous and discrete variable vectors, respectively.

Parallel Branch and Bound Algorithm

To obtain OPF optimal solutions with discrete variables, it is proposed that a classic branch and bound (B&B) algorithm be used. In this method the search process is organized structurally in the form of a binary tree, where every one of its nodes represents an OPF in which all the discrete variables are relaxed and treated as continuous variables (COPF). The nodes are generated from the root node, adding constraints on the discrete variables. Each COPF is solved using the multiple centralities interior point algorithm by Gondzio [10,11].

Let *x* be any component of the discrete variable vector *c*, constrained to the discrete values $\{x_i / i = 1,...n(x)\}$ where $n(x)$ is the number of discrete values allowed by x, $x_i < x_{i+1}$, and x_c its value corresponding to the optimal solution obtained in the COPF associated with a node in the B&B tree. If $x_i < x_c < x_{i+1}$, is satisfied, the solution obtained is not feasible. In this case, two optimization problems derived from the executed COPF are built, adding the constraint $x <= x_i$ to one, and $x >= x_{i+1}$ to the other. These optimization problems are associated with two child nodes from the previous node.

Next, the COPFs corresponding to the two new nodes generated, are executed. Using a heuristic rule, one of the two is selected, and the algorithm searches again for another discrete variable x in its solution so that x_c is infeasible. Once it has been obtained, the process of bounding and generating new nodes is repeated.

A new solution to the problem is obtained when in one node, all the discrete variables take a feasible value. Once this has been achieved, its cost serves to prune the branches of the tree in the event one wants to continue searching for new solutions.

The B&B algorithm is a method that suffers from the so-called curse of dimensionality. For this reason, the objective of the algorithm proposed is not to obtain the optimal solution to the problem, but rather to obtain a solution which can satisfy a certain optimality criterion. To overcome this difficulty, we propose resorting to two strategies:

- Using heuristic techniques to guide the variable selection process in the B&B tree toward a pseudo-optimal solution as quickly as possible.
- Using parallel computing to simultaneously evaluate multiple nodes of the tree.

Various methods exist for selecting the next node to continue going deeper in the tree and for selecting the variables to bound [12]. In the algorithm proposed, the node chosen as the next one to go deeper in the tree, is the node with the lowest cost among the nodes that have just been executed. And to choose the infeasible variables to bound and generate new nodes, two methods are used:

- Simultaneous selection of n_f infeasible variables.
- Selection of n_r infeasible variables whose COPF (x_c) result is close (*Tol*) to some discrete value.

$$(x_i - x_c)^2 \le Tol^2$$

Selection of n_f Simultaneous Variables

The largest possible number of n_f variables is chosen so that 2^{n_f} is less than the number of computing cores. Ideally, 2^{n_f} would be equal to the number of (n_c) cores so as not to underutilize the computing resources, but this is not always possible if the number of cores is not a power of *2*. Nor is it desirable to generate more tasks than cores since a multitasking environment distributes the processing time in a CPU among various tasks, producing context switch, which decreases performance.

Each nonfeasible discrete variable generates two new derived subproblems, obtained as a result of dividing the original feasible interval of the variable $[\underline{x}, \overline{x}]$ into two subintervals: $[\underline{x}, x_i]$ and $[x_{i+1}, \overline{x}]$. In each subproblem the variable is constrained to each one of the two previous intervals, thus eliminating the infeasible solution x_c and all the others included in (x_i, x_{i+1}). This process generates two new child nodes in the B&B tree, associated with the new constrained subproblems.

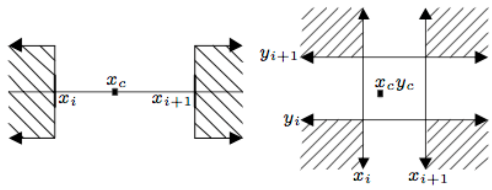

Fig. 1. Divison of the space after the bounding of one and two variables

Figure 1 show how the division of the feasible region of an *x* variable generates two unconnected regions and the division of the two variables (x, y) generates four. In general, the simultaneous selection of n_f variables generates 2^{n_f} unconnected regions with their corresponding nodes in the binary tree (Fig. 2).

If all the nodes can be executed parallely, a better exploration of the solutions space can be attained in practically the same execution time as a COPF.

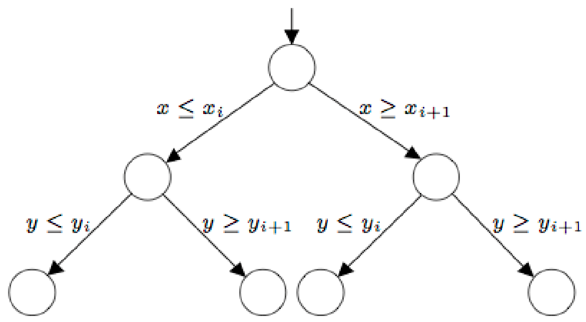

Fig. 2. Nodes generated by the simultaneous selection of two variables

The selection of the n_f infeasible discrete control variables is realized taking into account the following order of preference:

- First, those control variables close to their working limits are identified, for which some threshold value of proximity is defined, and then those infeasible discrete control variables whose variation has a greater effect on the control variables, are chosen. For example, if the voltage in a node is close to its limits, the infeasible discrete variables associated with the transformer taps and the connection of the shunts directly connected to that node, will be selected. The aim is to truncate the infeasible subtrees of the B&B tree as soon as possible.
- If with the previous step the n_f variables being sought are not obtained, more variables are selected depending on the distance of the continuous solution (x_c) to the center of the interval where that value is located $(x_i \le x_c \le x_{i+1})$. First, those variables whose continuous value's distance from the center of the interval is less than a certain percentage (p%) of the interval's length, are chosen. If the number of chosen variables is not sufficient, the value (p\%) is slowly increased until the desired number of variables is attained.

Selection of n_r Variables by Rounding Off

Let n be a node of the B&B tree. The solution of the COPF associated with said node can contain values of infeasible discrete variables. In other words, the values of these variables do not coincide with any of the allowed discrete values. Once a method of proximity (Tol) is established, the NS set comprised of the infeasible discrete variables "close" to an allowed discrete value is defined, in the following manner:

$$NS = \left\{ z \in X \quad / \quad (z_i - z_c)^2 \le Tol \right\}$$

The simplest and quickest alternative of trying to attain a feasible solution consists of establishing the value of the variables of NS to its closest permissible discrete variable and executing the COPF again. The objective function probably will not vary significantly from the initial COPF. This procedure is one of the most usual, and conceptually it is the one that follows, for example, in the reference [3].

Nevertheless, if it were not possible to find a feasible solution after rounding off, there is not another mechanism available capable of eliminating the cause of infeasibility.

This article proposes a methodology similar to the one above, but with the advantage of providing a robust mechanism for managing the infeasible subproblems.

From an n node of the B&B tree, a new subtree is created like the one described in the following steps:

1. Initially the n node is marked as the *selected node*.
2. A z variable is chosen from the NS set.
3. Two child nodes, associated with the two continuous subproblems are added to the *selected node*. The $z \le z_i$ constraint is added to the first child node and the $z \ge z_{i+1}$ constraint is added to the second child node.
4. The first of these is defined as new selected node if $(z - z_i) \le (z_{i+1} - z)$, otherwise the *selected node* will be the second child.
5. If unselected variables remain from the NS set, the process repeats from step 2.

The last *selected node* is associated with a COPF subproblem which differs from one which would be obtained using a simple rounding off, in that the added constraints are unequal, limiting each variable instead of assigning it a fixed value.

This approach presents several advantages:

- It provides a mechanism for managing infeasible subproblems. If a solution does not exist for a COPF, it can continue exploring the solutions space in a methodical manner, traversing the B&B tree.
- The variables, except at the ends of the interval, do not set a discrete value; the variable interval is only bound. This allows for a certain degree of freedom to obtain better solutions and avoid infeasibilities

As the variables are selected and bound, the subproblem is modified more with respect to the original COPF (associated with an n node), which is why it is advisable to set a maximum number of variables to be selected in this step.

With this selection method the intent is to set the value of the n_r variables indirectly. For this reason, when the number of selected variables is less than the maximum number of variables that can be selected simultaneously, this method provides no advantage over the simultaneous selection, and in this case, is discarded.

Construction of Subtree $n_f + n_r$

Fig. 3 shows an example of two variables selected by rounding off (x, y) and two others using the multiple selection method (s, z).

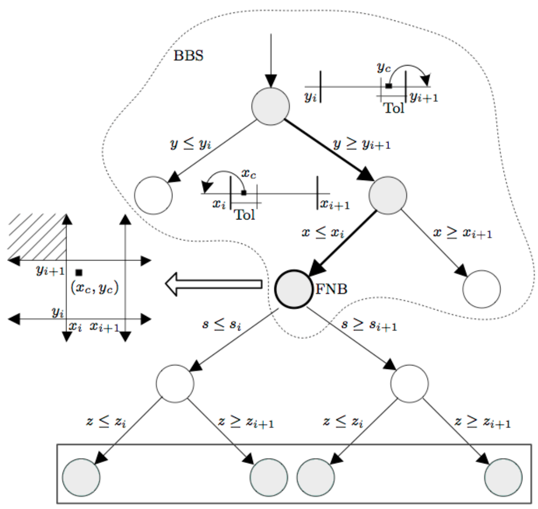

Fig. 3 Subtree created with the selection of the variables

The continuous values (x_c, y_c) of variables (x, y) are close to the discrete values x_i and y_{i+1}. The selection of these two variables as rounded off variables generates the BBS subtree which includes the node FNB. The constraints that bound the region of space closest to the point (x_c, y_c) are contained in that node and if the associated COPF is executed it is likely that the new values of the variables (x, y) will be located on its limits (x_i, y_{i+1}).

Once the construction of the subtree with all the variables of the NS set is complete, simultaneous n_f variables are selected which are different from the previous ones (as is explained in previous sections) and a new subtree is built from node FSB.

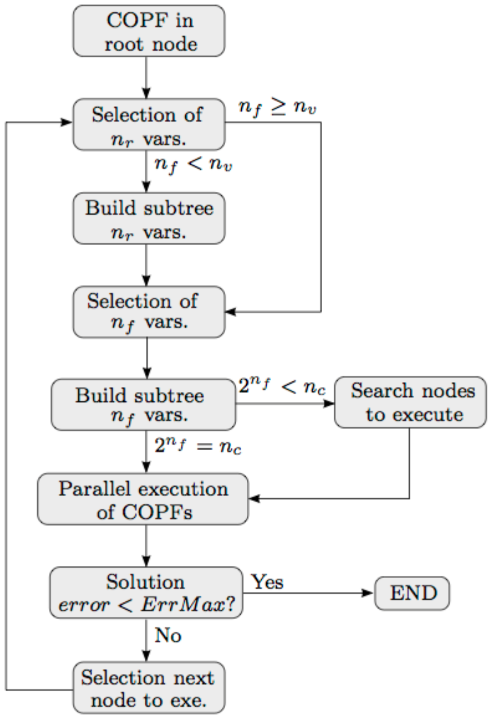

Fig. 4 Flow chart of the main algorithm steps for creating search tree nodes

Implementation

Figure 4 shows the flow chart from the algorithm's implementation. The algorithm begins executing a COPF with all the continuous variables and assigning this problem to the tree root. Next the main loop of the algorithm is executed, which is comprised of the following parts:

1. If the number of infeasible discrete variables (n_v) is greater than n_f, $min\{n_r, n_v - n_f\}$ rounded off variables are selected. Otherwise, no rounded off variables are selected.
2. Starting from the last *selected node* generated in the previous step, or from the initial node if no rounded off variables were able to be selected, the subtree nodes associated with simultaneous variables $min\{n_v, n_f\}$ are built. If $n_v < n_f$ and some cores remain idle, new subproblems from the nodes with lower cost infeasible variables are selected from the whole tree. In this case only two subproblems will be generated for each node, constraining only one variable.
3. The COPFs of all of the additional nodes generated are executed parallely.
4. If among the nodes executed, there is a feasible discrete solution that satisfies the optimality requisites, the algorithm ends. Otherwise, a new node is selected among those already executed, and the algorithm returns to step 1.

As feasible solutions are obtained from the DOPF problem, the upper bound of the global optimum is obtained. Given that the COPF problems are not convex, there is no guarantee that the interior point algorithm used will obtain a global optimum. Accordingly, it cannot be guaranteed that the cost of the solution obtained from the COPF associated with a B&B tree node constitutes a lower bound of the constrained problems located downstream. Nevertheless, it is assumed to be a reasonable lower bound. Therefore, it is possible to narrow the gap between the optimal solution and the last feasible solution found.

Numerical Case Studies

The OPF used in this article is programmed by the authors in C++ using the compiler Microsoft Visual Studio 2008 with Intel's MKL libraries. The maximum permissible error in the OPF is 1E-6 for all of the cases and the maximum number of iterations allowed is 30. The hardware used is a machine with two Intel Xeon E5440 processors, each with a quad core and 16GB of main memory. The operating system is Windows Server 2008 SP2.

Test Systems and OPF Problems Descriptions

To check the efficiency of the methodology proposed, various numerical experiences will be carried out on the IEEE networks of 118 and 300 nodes and on a network of 1415 nodes, based on the Spanish transport network. The minimizing functions are the losses (IEEE networks and the network based on the Spanish transport network) and the minimization of the generation costs (IEEE networks). All the cases are run on a different number of cores to check the scalability of the algorithm before the increase in the parallel computing power.

In the networks of 118 and 300 nodes, all of the transformation relations of the voltage regulating transformers are considered as variables limited to the interval [0.9, 1.1]. In the network of 300 nodes, the total number of module transformers now becomes 106 versus the original 400 and in the network of 118 nodes there is a total of 9 regulating transformers. In the two cases the number of taps considered is 21, which provides a step of 0.01 p.u.

In the network of 118 nodes, 50 shunts have been added, whose location and associated data have been extracted from reference [4]. In the network of 300 nodes 14 shunts have been added, whose values and positions have been extracted from [8]. In the two networks the voltages of the nodes are limited between the values [0.9, 1.1].

The data from the generators, costs and values of the maximum and minimum powers, for the IEEE-300 network are extracted from reference [8]. The data of all the networks has been made public in [13]. To the REEB-1415 network, the additional constraint that all the generators present in a same node must meet the condition of generating the same reactive power in values per unit has been added.

A summary of each network's main parameters is shown in table 1. In this table n_b, l, t, g, s, c_v and d_v are the number of buses, lines, transformers, generators, shunts, number of continuous control variables and number of discrete control variables, respectively.

Table 1. Test Systems Summary

System	n_b	l	t	g	s	c_v	d_v
IEEE 118	118	177	9	54	50	72	59
IEEE 300	300	375	106	69	14	133	120
REEB 1415	1415	1549	411	304	81	608	492

In the result tables the number of variables n_f (Vars) simultaneously selected, the number of cores used in the execution, the lower bound obtained from the solution of the COPFs (Min), the cost of the best solution found (Opt), the error committed $(Gap = (Opt - Min) / Min*100)$ and the time consumed, are shown. In every case the maximum number of variables selectable by rounding off (n_r) is 10 and the percentage value of tolerance for its selection (%p) is 10%.

Table 2. Results of IEEE-118 Minimun Cost Test Case

Vars	Cores	Costs ($)		Gap %	Time(ms)
		Min.	Opt.		
1	2	13.4527	13.4547	0.0144	548
2	4	13.4529	13.4547	0.0133	309
3	8	13.4529	13.4546	0.0124	290

Table 3. Results of IEEE-118 Minimum Losses Test Case

Vars	Cores	Losses(MW)		Gap %	Time(ms)
		Min.	Opt.		
1	2	93.567	93.896	0.3512	652
2	4	93.206	93.887	0.3045	315
3	8	93.655	93.882	0.2421	284

Tables 2 and 3 show the execution times and results of minimizing the functions of losses and costs using different numbers of cores for solving them. In every case the totality of the cores available are assigned to the search for solutions. As one can see, the times obtained are lower as we increase the number of cores assigned in the execution.

Table 4. Results of IEEE-300 Minimum Losses Test Case

Vars	Cores	Losses(MW)		Gap %	Time(s)
		Min.	Opt.		
1	2	198.7269	199.0452	0.1647	4.8
2	4	198.7466	199.0555	0.1554	2.9
3	8	198.7271	199.1630	0.2193	2.2

Table 5. Results IEEE-300 Minimun Cost Test Case

Vars	Cores	Costs ($)		Gap %	Time(s)
		Min.	Opt.		
1	2	795.0745	798.2378	0.3978	4.9
2	4	795.0745	798.2118	0.3945	3.1
3	8	795.0745	798.3300	0.1579	2.5

Table 6. Minimum Power Losses with Discrete Trafos and Shunts

Vars	Cores	Losses(MW)		Gap %	Time(s)
		Min.	Opt.		
1	2	710.346	903.866	0.679	177
2	4	743.286	898.946	0.545	84
3	8	715.976	837.436	0.426	64

Table 7. Minimum Power Losses With Only Discrete Shunts

Vars	Cores	Losses(MW)		Gap %	Time(s)
		Min.	Opt.		
1	2	710.346	751.826	0.145	12.1
2	4	743.286	752.066	0.030	7.1
3	8	712.556	750.696	0.133	6.9

Tables 4 and 5 show the results obtained by the algorithm in the IEEE 300 network, with a total of 120 discrete variables.

The REEB-1415 network has a total de 492 discrete variables between the transformers (411) and the shunts (81). The minimizing function has been limited to the losses in the network. Table 6 shows the results of the system, which considers only the shunts as discrete. Table 7 shows the times and the results, taking into account the transformers and the shunts.

Systems with a Number of Cores Different from 2^{n_f}

In cases where it is not possible to select a number of n_f variables which satisfies the relationship $cores = 2^{n_f}$, the remaining idle cores can be used to limit the minimal cost of the problem. The procedure is the following:

1. From the B&B tree, the leaf node (without children) associated with the minimum cost COPF, but infeasible since it does not coincide with any variable with its admissible discrete values, is selected.
2. Two child nodes are generated that are associated with the subproblems which result from constraining any infeasible variable.
3. While cores remain available the next minimum cost leaf node is selected and a new pair of child nodes is generated.
4. The child nodes are executed simultaneously in the idle cores, along with the other subproblems.
5. The minimum cost of all the tree leaf nodes constitutes a lower bound of the objective function, obviating any exception derived from the non convexity as has been previously explained.

If the minimizing costs problem is executed on the IEEE-118 network, on a machine with 6 cores selecting two simultaneous variables (4 cores), and a bound solution is required with a relative gap of less than 0.2%, the execution time obtained is 8.4 s. if 2 idle cores are used to bound the solution, and 9.1s. if they are not used.

Conclusions

This article presents an algorithm for assigning discrete variables parallelizing a branch and bound algorithm. Its use is recommended when the easiest and quickest methods like rounding off to the nearest discrete value fails for some reason (infeasibility or loss of optimality).

Its main advantages over other solutions are:

- It naturally ma nages the infeasibilities that could be incurred during the course of the adjustments of the discrete variables.
- The algorithm allows for specifying, and therefore, limiting the maximum error allowed in the solution.
- With the selection and the parallel execution of various nodes the computation time is significantly reduced, a greater number of combinations between the variables is explored and better optimums can be obtained.
- The algorithm aims to take advantage of the current trend to include an increasing number of cores in the processors.

REFERENCES

[1] H. Dommel andW. Tinney, "Optimal power flow solutions," IEEE Trans. Power App. Syst., vol. PAS-87, no. 10, pp. 1866–1876, Oct. 1968.

[2] W. Tinney, J. Bright, K. Demaree, and B. Hughes, "Some deficiencies in optimal power flow," IEEE Trans. Power Syst., vol. 3,no. 2, pp. 676–683, May 1988.

[3] K. Karoui, L. Platbrood, H. Crisciu, and R. A. Waltz, "New largescale security constrained optimal power flow program using a new interior point algorithm," in 2008 5th International

Conference on the European Electricity Market. IEEE, May 2008, pp.1–6.
[4] P. J. Macfie, G. A. Taylor, M. R. Irving, P. Hurlock, and H.-B. Wan, "Proposed shunt rounding technique for large-scale security constrained loss minimization," IEEE Trans. Power Syst., vol. 25, no. 3, pp. 1478–1485, Aug. 2010.
[5] Mingbo Liu, S. Tso, and Ying Cheng, "An extended nonlinear primal-dual interior-point algorithm for reactive-power optimization of large-scale power systems with discrete control variables," IEEE Trans. Power Syst., vol. 17, no. 4, pp. 982–991, Nov.2002.
[6] S.-S. Lin, C.-H. Lin, and S.-C. Horng, "An efficient algorithm for qcdp and implementation," IEEE Trans. Power Syst., vol. 25, no. 1, pp. 234–242, Feb. 2010.
[7] C.-H. Lin and S.-Y. Lin, "Distributed optimal power flow with discrete control variables of large distributed power systems," IEEE Trans. Power Syst., vol. 23, no. 3, pp. 1383–1392, Aug. 2008.
[8] F. Capitanescu and L. Wehenkel, "Sensitivity-based approaches for handling discrete variables in optimal power flow computations," IEEE Trans. Power Syst., vol. 25, no. 4, pp. 1780–1789, Nov. 2010.
[9] L. Liu, X. Wang, X. Ding, and H. Chen, "A robust approach to optimal power flow with discrete variables," IEEE Trans. Power Syst., vol. 24, no. 3, pp. 1182–1190, Aug. 2009.
[10] J. Gondzio, "Multiple centrality corrections in a primal-dual method for linear programming," Computational Optimization and Applications, vol. 6, no. 2, pp. 137–156–156, Sep. 1996.
[11] G. Torres and V. Quintana, "On a nonlinear multiple-centralitycorrections interior-point method for optimal power flow," IEEE Trans. Power Syst., vol. 16, no. 2, pp. 222–228, May 2001.
[12] C. A. Floudas, Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications (Topics in Chemical Engineering). Oxford University Press, USA, Oct. 1995.
[13] "Discrete optimal power flow." [Online]. Available:
http://dee.uvigo.es/member/jcmeira/page/DOPF

***Authors**: Ph.D. J. C. Moreira, Ph.D. E. Míguez, M.Sc. C. Vilachá, Ph.D. Antonio F. Otero. Department of Electrical Engineering, Universidade de Vigo, Lagoas-Marcosende 9, 36310 Vigo, Spain, email: jcmeira@uvigo.es.* *C. Vilachá thesis is possible thanks to the Spanish Ministerio de Educación by the 2010 training program.*