TRANSMISSION EXPANSION PLANNING WITH RE-DESIGN
A Greedy Randomized Adaptive Search Procedure

Rosa Figueiredo[1], Pedro Henrique González Silva[2] and Michael Poss[3]
[1]*CIDMA, Departamento de Matemática, Universidade de Aveiro, Aveiro, Portugal*
[2]*Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brasil*
[3]*GOM, Département d'Informatique, Université Libre de Bruxelles, Bruxelles, Belgique*

Keywords: Transmission expansion planning, Network design, Grasp.

Abstract: Transmission expansion planning with re-design has been recently proposed in the literature to improve on the classical transmission expansion planning by allowing to cut-off circuits while expanding the network. Although the reductions in the solution costs are significant, the resulting mixed-integer linear programming formulations are very difficult to solve exactly for large networks. In this work, we propose the first meta-heuristic for the transmission expansion planing problem with re-design: a simple yet efficient GRASP meta-heuristic. We show on realistic networks for which the optimal solutions are known that our method is able to provide in short amounts of time feasible solutions as cheap as the optimal ones. Moreover, we are able to compute a new feasible solution for benchmark instance *Brazil Southeast* that is cheaper than the best solution from the literature.

1 INTRODUCTION

With the growth of energy demand over the years, it becomes necessary for the managing entity to change the electrical power system, adding new transmission lines and power generators. Since transmission lines are expensive to build, one would like to build new generating units to tailor the supply of nearby consumers. However, it is usually not possible or not economical to build the new generating units close to consumption centers so that they must be constructed in distant places. Consider for instance the situation of Brazil. The country possesses large resources in hydropower. However, the later are usually located far from main cities and industries. Therefore, it is necessary to build new transmission circuits in order to integrate all power plants into the electrical network.

The decisions of the planning process yield an optimization problem that must decide of both the construction of the generating units and the transmission lines. This optimization problem is very hard to solve, even approximately, so that most practitioners split the problem into two smaller optimization problems. The first problem chooses the best generating units and their emplacements. The second problem designs the cheapest transmission network to connect these units and the consumption centers. In this paper, we focus on the problem of designing of the network, given that the new power generating units are already built.

Most works on transmission expansion planning consider only the possibility of adding new circuits to an existing transmission network, see the review of (Latorre et al., 2003). Namely, all of the existing circuits must be used in the expanded network. This problem is denoted by (TEP) in what follows. Recently, the authors of (Moulin et al., 2010) and (Khodaei et al., 2010) have independently shown that allowing for cutting-off existing circuits while expanding the network can lead to cheaper expansion plans. Nevertheless, the resulting optimization problems are extremely difficult to solve exactly for real-size networks. In (Moulin et al., 2010), the authors show that the classical transmission expansion planning problem can be solved exactly for large-scale networks in a couple of minutes using CPLEX 12 (IBM-ILOG, 2009). However, they show that the problem with re-design (denoted by $(\mathsf{TEP_R})$ in what follows) is significantly harder to solve exactly. In particular, they are unable to compute the optimal solution for their larger instance *Brazil Southeast*.

The extreme difficulty to solve $(\mathsf{TEP_R})$ exactly is the motivation for this work. We develop and test a

Figueiredo R., González Silva P. and Poss M..
TRANSMISSION EXPANSION PLANNING WITH RE-DESIGN - A Greedy Randomized Adaptive Search Procedure.
DOI: 10.5220/0003723403800385
In *Proceedings of the 1st International Conference on Operations Research and Enterprise Systems* (ICORES-2012), pages 380-385
ISBN: 978-989-8425-97-3

meta-heuristic procedure that intends to provide good solutions for $(\mathsf{TEP_R})$ in a limited amount of time. Our algorithm extends to $(\mathsf{TEP_R})$ the GRASP developed in (Binato et al., 2001a) for (TEP). We test the algorithm on the real instances used in (Moulin et al., 2010) and our algorithm finds the optimal solution for a set of instances for which the optimal solution is known. Moreover, it is able to provide in eleven minutes a new solution for *Brazil Southeast*, cheaper than the best solution found in the literature (Moulin et al., 2010) after several hours of branch-and-bound search.

This paper is structured as follows. The next section describes the notation used throughout the paper and a mathematical programming formulation for $(\mathsf{TEP_R})$. Section 3 then presents the key aspects of the GRASP developed herein. Finally, Section 4 presents a numerical comparison of our meta-heuristic and the integer programming formulation from (Moulin et al., 2010).

2 MODEL AND NOTATIONS

2.1 Nomenclature

2.1.1 Sets and Parameters

B Set of buses.

Ω Set of all circuits.

Ω^0 Set of existing circuits.

Ω^1 Set of candidate circuits.

γ_k Susceptance of circuit k.

c_k Investment cost of circuit k.

$\overline{f}_k$ Capacity of circuit k.

d_i Load at bus i.

$\overline{g}_i$ Maximal generation at bus i.

2.1.2 Variables

x_k Indicates if circuit k is present.

f_k Flow on circuit k.

g_i Generation at bus i.

θ_i Potential angle at bus i.

2.2 Mathematical Formulation

From the Combinatorial Optimization point of view, the electrical network is an undirected graph (B,Ω) where vertices $i \in B$ are called buses and edges $k \in \Omega$ are called circuits. The set of circuits is partitioned into a subset Ω^0, of existing circuits, and a subset of candidate circuits, denoted by Ω^1. For each circuit $k \in \Omega$, indices $i(k)$ and $j(k)$ denote, respectively, the head and the tail of the circuit (which are chosen arbitrarily), while $\gamma_k, \overline{f}_k$ and c_k are the circuit susceptance, capacity and cost, respectively. For each node $i \in B$, $\delta^-(i) := \{k \in \Omega : i(k) = i\}$ and $\delta^+(i) := \{k \in \Omega : j(k) = i\}$. The network can have parallel circuits, $k_1, k_2 \in \Omega$, denoted by $k_1 \parallel k_2$, linking the same terminal buses.

Different models can be found in the literature to describe power flow in a transmission network (Bienstock and Mattia, 2007). The most accurate one is called the *AC* flow model. In that model, the voltage at node i of the network is represented by a complex number, $U_i e^{\sqrt{-1}\theta_i}$, where U_i is the voltage amplitude and θ_i is the voltage angle at i. The power flowing from i to j along the (undirected) edge (ij) is a complicated function that involves quadratic terms in U_i and U_j multiplied by trigonometric functions of the difference $\theta_i - \theta_j$. Expressing the flow in this accurate way yields highly non-convex MINLPs, which are extremely difficult to solve exactly even for very small networks. For this reason, most approaches on transmission expansion planning problem use the linearized *DC* approach which defines the flow between two buses as

$$f_k = \gamma_k(\theta_{i(k)} - \theta_{j(k)}). \tag{1}$$

Using the *DC* model described in (1), $(\mathsf{TEP_R})$ can be written in the following form:

$$\min \quad \sum_{k\in\Omega^1} c_k x_k \tag{2}$$

$$\text{s.t.} \quad \sum_{k\in\delta^-(i)} f_k - \sum_{k\in\delta^+(i)} f_k = d_i - g_i \quad i \in B \tag{3}$$

$$f_k - x_k\gamma_k(\theta_{i(k)} - \theta_{j(k)}) = 0 \quad k \in \Omega \tag{4}$$

$$-\overline{f}_k \le f_k \le \overline{f}_k \quad k \in \Omega \tag{5}$$

$$0 \le g_i \le \overline{g}_i \quad i \in B \tag{6}$$

$$x_k \in \{0,1\} \quad k \in \Omega. \tag{7}$$

Objective function (2) contains the cost of circuits pertaining to Ω^1. Hence, the cost of cutting off existing circuits is assumed to be negligible. Constraints (3) ensure that the incoming flow at each bus is equal to the load requirement minus the generation at the bus. Each constraint in (4) is defined by a disjunction on the presence x_k of circuit k. If $x_k = 1$, the circuit is present so that the associated equation in (4) defines the flow for circuit k according to linearized Kirchoff law. If $x_k = 0$, circuit k is not present so that the associated equation in (4) yields a null flow for circuit k. Finally, constraints (5) and (6) are bounds on the

real variables and constraints (7) force x to be a binary vector. Notice that a negative flow means that electricity flows in the reverse direction on the circuit.

It is easy to see that (TEP) can be obtained from $(\mathsf{TEP_R})$ by adding the set of constraints

$$x_k = 1 \qquad k \in \Omega^0. \tag{8}$$

The practical interest of $(\mathsf{TEP_R})$ is that its optimal solution cost may be strictly smaller than the one of (TEP). We compare in Table 1 the optimal costs for several realistic networks; the "$\leq$" in the last line indicates that the optimal cost of $(\mathsf{TEP_R})$ is not known for that instance. More details on these instances are provided in the numerical experiments section.

Table 1: Optimal costs computed by (Moulin et al., 2010).

Name	(TEP)	$(\mathsf{TEP_R})$
Garver	110	110
IEEE24	152	152
South	154.4	146.2
South Red	72.87	63.2
Southeast	424.8	$\leq$ 405.9

Table 1 shows that for the three last instances, there is a clear benefit in allowing for re-design. However, problem $(\mathsf{TEP_R})$ is much harder to solve than (TEP) mainly because it contains more binary variables. In fact, while it is easy to solve (TEP) with modern solvers for realistic networks, $(\mathsf{TEP_R})$ is still very challenging, see (Moulin et al., 2010). The objective of this work is therefore to provide good feasible solutions for $(\mathsf{TEP_R})$ computable in a short amounts of time.

3 ALGORITHM DESCRIPTION

We present in this section our meta-heuristic GRASP. Our algorithm follows closely the line of the GRASP from (Binato et al., 2001a) which we adapted to handle the possibility of cutting-off some of the circuits. GRASP algorithms typically consists in two phases, see (Feo and Resende, 1995) for more details. In the construction phase, one intelligently constructs an initial solution via an adaptive randomized greedy function, see Algorithm 1. In the local search phase, one tries to improve the constructed solution by looking at its neighborhood, see Algorithm 2. These two phases are described for $(\mathsf{TEP_R})$ in Sections 3.1 and 3.2 below.

3.1 Construction Phase

The objective of this phase is to build a network that can attend the demand of each consumption site. Formally, we want to construct a set of circuits $\hat{\Omega} \subseteq \Omega$ with $\hat{x}_k = 1$ for $k \in \hat{\Omega}$ and 0 otherwise, such that there exists $(\theta, f, g) \in \mathbb{R}^{|B|^2 \times |\Omega|}$ that satisfies constraints (3)–(6) for $\hat{x}$. To check the feasibility of a given circuits set $\hat{\Omega}$, it will be useful to consider the following feasibility test:

$$\mathsf{FEAS}(\hat{\Omega}) := \quad \min \sum_{i \in B} r_i$$

s.t.

$$\sum_{k \in \hat{\delta}^-(i)} f_k - \sum_{k \in \hat{\delta}^+(i)} f_k = d_i - g_i - r_i \qquad i \in B \tag{9}$$

$$f_k - \gamma_k(\theta_{i(k)} - \theta_{j(k)}) = 0 \qquad k \in \hat{\Omega} \tag{10}$$

$$-\overline{f}_k \leq f_k \leq \overline{f}_k \qquad k \in \hat{\Omega}$$

$$0 \leq g_i \leq \overline{g}_i \qquad i \in B,$$

where r_i describes the shortage at bus i, $\hat{\delta}^-(i) := \{k \in \hat{\Omega} : i(k) = i\}$ and $\hat{\delta}^+(i) := \{k \in \hat{\Omega} : j(k) = i\}$ for each node $i \in B$. Constraints (9) are different from (3) in two aspects. First, for each $i \in B$ sets $\delta^+(i)$ and $\delta^-(i)$ are replaced by $\hat{\delta}^+(i)$ and $\hat{\delta}^-(i)$ to consider only the flows for circuits in $\hat{\Omega}$. Second, each constraint in (9) features a shortage variable r_i in its right-hand side to match the demand if the whole load of bus i cannot be supplied by the network defined by $\hat{\Omega}$. Constraints (10) are derived from constraints (4) by setting x_k to one for each $k \in \hat{\Omega}$ and relaxing constraints for $k \in \Omega \backslash \hat{\Omega}$. The last two sets of constraints are bounds on the flows and on the generations. We see that $\mathsf{FEAS}(\hat{\Omega}) = 0$ implies that $\hat{\Omega}$ is a feasible circuits set in the sense above.

To start the construction phase, we set $\hat{\Omega} := \emptyset$. Then, an iterative algorithm tries to construct a circuits set $\hat{\Omega}$ such that $\mathsf{FEAS}(\hat{\Omega}) = 0$ by adding one circuit at the time to $\hat{\Omega}$. In order to choose the circuit to add at each step, we compute a greedy function that approximates the benefit from adding each circuit individually. Then, using the greedy function we rank the circuits in order to construct the candidate list $CL \subseteq \Omega$, that is, the set of most promising circuits. Finally, we randomly choose an element from CL and add it to $\hat{\Omega}$. This procedure is repeated until either a feasible solution is found or all circuits have been added. In the latter situation, no feasible solution has been found. In a second step, we try to withdraw circuits one by one according to their costs. The whole procedure is resumed in Algorithm 1.

We are left to describe our greedy function and how to construct CL. Let $\pi \in \mathbb{R}^{|B|}$ be the set of dual variables associated to constraints (9). The authors of (DeChamps et al., 1979) have shown that the benefits of susceptance changes can be estimated by

Algorithm 1: Construction phase.

$\hat{\Omega}$ is *empty*;
repeat
  Evaluate the greedy function for each $k \in \Omega \backslash \hat{\Omega}$;
  Build *CL*;
  k is randomly selected from *CL*;
  $\hat{\Omega} := \hat{\Omega} \cup \{k\}$;
until $FEAS(\hat{\Omega}) = 0$ **or** *CL is empty*;
if $FEAS(\hat{\Omega}) > 0$ **then**
  No feasible solution found, **abort**;
else
  repeat
    Rank circuits in $\hat{\Omega}$ by decreasing cost c;
    k is the first element of the list;
    $\hat{\Omega} := \hat{\Omega} \backslash \{k\}$;
  until $FEAS(\hat{\Omega}) > 0$;
  $\hat{\Omega} := \hat{\Omega} \cup \{k\}$;
return $\hat{\Omega}$

$$\Pi_k = (\pi_{i(k)} - \pi_{j(k)})(\theta_{i(k)} - \theta_{j(k)}) \qquad (11)$$

for each $k \in \Omega \backslash \hat{\Omega}$. We construct *CL* by ordering the circuits in $\Omega \backslash \hat{\Omega}$ by decreasing value of Π and taking the 70% best circuits in the list. Algorithm 1 may not always find a feasible solution. This is due to the fact that (11) provides only an intuitive indication on the promising new circuits, it is not a rigorous measure. For this reason, the elements are selected *randomly* from *CL*.

The main difference between our algorithm and the one from (Binato et al., 2001a) is the following: their algorithm starts with $\hat{\Omega} = \Omega^0$ and always consider that potential new circuits belong to Ω^1. In opposition, our algorithm starts with an empty set $\hat{\Omega}$ and considers potential new circuits in the whole set Ω.

3.2 Local Search

Since the greedy function used in the construction phase does not involve the cost of the circuits added at each iteration, it is likely to end up with a costly circuits set $\hat{\Omega}$. In particular, there may exist a cheaper circuit set Ω^* that satisfies $\mathsf{FEAS}(\Omega^*) = 0$ as well and that has structure very similar to $\hat{\Omega}$. The concept of similar structure can be formalized by the introduction of neighborhoods. Generally speaking, a neighborhood is a mapping that associates a set of circuit set to each $\Omega^* \subseteq \Omega$. The local search phase tries therefore to improve the constructed solution $\hat{\Omega}$ by looking at its neighborhood.

Consider a circuit set $\Omega^* \subseteq \Omega$. In this work, we use the following neighborhoods: neighborhood $N^n(\Omega^*)$ of Ω^* is the set that contains all circuit sets with the same number of circuits as Ω^*, among which

Algorithm 2: Local search phase using N^n.

Let $\Omega^* \subseteq \Omega$ be the circuit set constructed with Algorithm 1;
repeat
  $\hat{\Omega} := \Omega^*$;
  foreach $\Omega' \in N^n(\Omega^*)$ **do**
    Compute $\mathsf{FEAS}(\Omega')$;
    if $FEAS(\Omega') = 0$ **and** $c(\Omega') < c(\hat{\Omega})$ **then**
    $\hat{\Omega} := \Omega'$;
  if $c(\hat{\Omega}) < c(\Omega^*)$ **then** $\Omega^* := \hat{\Omega}$;
  else Ω^* is a local solution;
until Ω^* *is a local solution*;

exactly n are different from the circuits in Ω^*. It is defined formally as

$$N^n(\Omega^*) := \{\Omega' \text{ s.t. } \Omega' \subseteq \Omega,\ |\Omega'| = |\Omega^*|,\ |\Omega^* \backslash \Omega'| = n\}.$$

We also define $c(\Omega^*) := \sum_{k \in \Omega^*} c_k$. We say that Ω^* is a *local solution* for neighborhood N^n if $c(\Omega^*) \leq c(\Omega')$ for each $\Omega' \in N^n(\Omega^*)$ that satisfies $\mathsf{FEAS}(\Omega') = 0$. The main ingredient of the local search phase is to look iteratively at the neighborhood of the current solution for a solution with lower cost until we are stuck at a local solution. The procedure is schematized in Algorithm 2.

The bottleneck of Algorithm 2 is located at the solution of the linear programs necessary to compute $\mathsf{FEAS}(\Omega')$ for each $\Omega' \in N^n(\Omega^*)$. If Ω does not contain parallel circuits, then the neighborhood size of a circuit set Ω^* is equal to

$$\binom{n}{|\Omega^*|}\binom{n}{|\Omega \backslash \Omega^*|} = \frac{|\Omega^*|!}{n!(|\Omega^*| - n)!}\frac{|\Omega \backslash \Omega^*|!}{n!(|\Omega \backslash \Omega^*| - n)!}, \qquad (12)$$

that is, the neighborhood size grows extremely fast with the value of n. In all our cases studies, the networks contain parallel circuits so that (12) must be replaced by a smaller number that depends on the number of parallel circuits allowed for each pair of buses. Although the resulting number is smaller than (12), it is still prohibitive when $n > 1$. For this reason, we limit our tests to $n = 1$ and $n = 2$ in our cases studies. Notice that additional tricks have been used in this work to decrease the number of elements from N^n that need to be considered, see (Binato et al., 2001a).

4 NUMERICAL EXPERIMENTS

In this section, we compare the solution times and the best feasible solutions found using the GRASP described in this paper and the integer programming formulation proposed in (Moulin et al., 2010).

4.1 Data

We compare the two solution methods on the realistic networks whose main characteristics are provided in Table 2. The first column indicates the name of the instance, the next two columns provide the number of buses and circuits, respectively, while the last column provides a reference where the complete instance details can be found. The difference between instances *South* and *South Red* concerns the maximum generations available: *South Red* allows for re-dispatch while *South* does not allow for re-dispatch, that is, the maximal generations for *South* have been scaled down to meet the total demand $\sum_{i\in B} d_i$.

Table 2: Networks Characteristics.

Name	$\|B\|$	$\|\Omega\|$	Reference
Garver	6	96	(Garver, 1970)
IEEE24	24	140	(Alguacil et al., 2003)
South	46	299	(Binato, 2000)
South Red	46	299	(Binato, 2000)
Southeast	79	405	(Binato, 2000)

4.2 Integer Programming

The non-linearity of $(\mathsf{TEP_R})$ arises from constraints (4) that turn the problem into a Non-Convex Mixed-Integer Linear Program. Although this class of programs has witnessed a tremendous attention in the past years (Belotti et al., 2008), they are still very challenging to solve to optimality, especially for problems as large as $(\mathsf{TEP_R})$. For this reason, most approaches for the transmission expansion planning problem use instead linearizations that introduce "big-M" coefficients, see (Bahiense et al., 2001; Binato et al., 2001b; Moulin et al., 2010; Villanasa, 1984), among others. One exception is (Rider et al., 2008) who tackle the non-convex MINLP by a branch-and-bound algorithm. Although their approach provides interesting results for their cases studies, they cannot guarantee the optimality of their solution because of the non-convexity of the problem.

In this work, we follow the traditional approach that linearizes (4) with the help of "big-M" coefficients. According to (Moulin et al., 2010), an efficient approach replaces constraints (4) with

$$-M_k(1-x_k) \leq f_k - \gamma_k(\theta_{i(k)} - \theta_{j(k)}) \leq M_k(1-x_k),$$

for each $k \in \Omega$, and constraints (5) with

$$-x_k\overline{f}_k \leq f_k \leq x_k\overline{f}_k \qquad \text{for each } k \in \Omega.$$

One of the difficulties of the resulting MILP arises from the symmetry existing among variables x_k that are associated with parallel circuits. Basically, parallel circuits yield feasible points that are indistinguishable by the objective function. In order to break this symmetry, we order parallel circuits by introducing the partial order $\prec$ among elements of Ω and introduce the precedence constraint $x_k \geq x_h$ for each pair $k,h \in \Omega$ such that $h \prec k$, see (Moulin et al., 2010) for the details.

4.3 GRASP

In practice, our GRASP starts by running Algorithm 1 until a feasible solution is found. Then, a local search is performed as described in Algorithm 2. Many iterations of Algorithm 1 were needed, ranging from 8 for *Garver* to 4078 for *Southeast* for the results presented in Table 3. Consequently, the whole algorithm spends significantly more time in the construction phase, with between 15% and 25% of the total solution time spent in the local search.

4.4 Results

Both approaches have been coded in Xpress Mosel 3.2.1 with the solver Xpress Optimizer 21.01.06 (FICO, 2009) on a computer using a processor Pentium Core 2 Quad Q6600 de 2.4 GHz and 4GB of RAM memory. In addition, we have also tested the integer programming approach with the Interactive Optimizer of CPLEX 12.2 (IBM-ILOG, 2009).

The best feasible solutions obtained with the GRASP are given in Table 3. The second column of the table shows the best feasible solution reported in the literature, see (Moulin et al., 2010), which are optimal for all networks but *Southeast*. The third column shows the best feasible solution obtained using the construction phase only, while the next two columns show the best solutions after performing a local search with neighborhood N^1 and N^2, respectively. For all instance but *Southeast*, the solutions found after any of the local search are the optimal ones. More interestingly, the local search with N^2 found a cheaper solution for *Southeast* than the best feasible solution reported in the literature.

Table 3: Results for GRASP after the construction phase, the local search with N^1, and the local search with N^2.

Name	Best	Cons	N^1	N^2
Garver	110	130	110	110
IEEE24	152	194	152	152
South	146.2	172.8	146.2	146.2
South Red	63.2	75.81	63.2	63.2
Southeast	405.9	406.8	405.9	392.8

In Table 4, we provide CPU times in seconds for the GRASP using N^2 and CPU times in seconds or duality gaps (when it is available) for the MILP formulations (a time limit of ten hours has been set) using Xpress and CPLEX, respectively. Of course, the GRASP only computes a feasible solution while the MILP formulations also provide a guarantee of optimality. Hence, we do not intend to compare directly these times but we rather want to check that GRASP is fast enough for the difficult instances. We see from Table 4 that GRASP finds good feasible solutions for the difficult instances *South* and *Southeast* in a couple of seconds and minutes, respectively, while both solvers can hardly solve these instances within 10 hours of computing time. The best solution found by CPLEX for *Southeast* has a cost of 527.70, while Xpress could find no feasible solution within the time limit.

Table 4: Running times in seconds (or duality gaps) for the MILP formulation and for the GRASP.

Name	GRASP	Xpress	CPLEX
Garver	3.27	0.062	0.27
IEEE24	0.92	6.47	5.94
South	7.41	**7.15%**	32319
South Red	7.68	48.8	31
Southeast	632.17	–	**71.41%**

ACKNOWLEDGEMENTS

Michael Poss is supported by an “Actions de Recherche Concertées” (ARC) projet of the “Communautée française de Belgique” and is a research fellow of the “Fonds pour la Formation à la Recherche dans l’Industrie et dans l’Agriculture” (FRIA). Rosa Figueiredo is supported by *FEDER* founds through *COMPETE*–Operational Programme Factors of Competitiveness and by Portuguese founds through the CIDMA (University of Aveiro) and FCT, within project PEst-C/MAT/UI4106/2011 with COMPETE number FCOMP-01-0124-FEDER-022690. Pedro Henrique González Silva thanks financial support of a master scholarship from brazilian funding agency CAPES.

REFERENCES

Alguacil, N., Motto, A. L., and Conejo, A. J. (2003). Transmission expansion planning: A mixed-integer lp approach. *IEEE Transactions on Power Systems*, 18(3):1070–1077.

Bahiense, L., Oliveira, G., Pereira, M., and Granville, S. (2001). A mixed integer disjunctive model for transmission network expansion. *IEEE Transactions on Power Systems*, 16(3):560–565.

Belotti, P., Lee, J., Liberti, L., Margot, F., and Waechter, A. (2008). Branching and bounds tightening techniques for non-convex minlp. Technical report, IBM Research Report RC24620.

Bienstock, D. and Mattia, S. (2007). Using mixed-integer programming to solve power grid blackout problems. *Discrete Optimization*, 4(1):115–141.

Binato, S. (2000). Optimal expansion of transmission networks by benders decomposition and cutting planes. Ph.D. dissertation (Portuguese), Federal University of Rio de Janeiro.

Binato, S., Oliveira, G. C., and L.Araújo, J. (2001a). A Greedy Randomized Adaptive Search Procedure for Transmission Expansion Planning. *IEEE Transactions on Power Systems*, 16(2):247–253.

Binato, S., Pereira, M. V. F., and Granville, S. (2001b). A new benders decomposition approach to solve power transmission network design problems. *IEEE Transactions on Power Systems*, 16:235–240.

DeChamps, C., Vankelecom, J., and Jamoulle, E. (1979). TRANEX–An interactive computer program for transmission expansion planning. In *IEEE IPES Summer Meeting,Vancouver, Canada, paper A79 476-3*.

Feo, T. A. and Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133.

FICO Xpress Optimization Suite (2009). Xpress-Optimizer Reference manual.

Garver, L. (1970). Transmission network estimation using linear programming. *IEEE Trans. Power Appar. Syst.*, 89(7):1688–1697.

IBM-ILOG (2009). IBM-ILOG Cplex.

Khodaei, A., Shahidehpour, M., and Kamalinia, S. (2010). Transmission switching in expansion planning. *IEEE Transactions on Power Systems*, 25(3):1722–1733.

Latorre, G., Cruz, R. D., Areiza, J. M., and Villegas, A. (2003). Classification of publications and models on transmission expansion planning. *IEEE Transactions on Power Systems*, 18(2):938–946.

Moulin, L. S., Poss, M., and Sagastizábal, C. (2010). Transmission expansion planning with re-design. *Energy Systems*, 1(2):113–139.

Rider, M., Garcia, A., and Romero, R. (2008). Transmission system expansion planning by a branch-and-bound algorithm. *IET Generation, Transmission & Distribution*, 2:90–99.

Villanasa, R. (1984). *Transmission network planning using linear and mixed linear integer programming*. PhD thesis, Ressenlaer Polythechnic Institute.