POWER TRANSMISSION NETWORK DESIGN BY A GREEDY RANDOMIZED ADAPTIVE PATH RELINKING APPROACH

Haroldo Faria Jr.
COPPE/UFRJ
Rio de Janeiro, RJ, Brazil
faria@vishnu.coep.ufrj.br

Silvio Binato
CEPEL
Rio de Janeiro, RJ, Brazil
silvio@cepel.br

Mauricio G. C. Resende
AT&T Research
Florham Park, NJ, USA
mgcr@research.att.com

Abstract - This paper illustrates results obtained by a new metaheuristic approach, Greedy Randomized Adaptive Search Path Relinking, applied to solve static power transmission network design problems. This new approach consists of a generalization of GRASPs concepts to explore different trajectories between two "high-quality" solutions. The results presented were obtained from two real-world case studies from Brazilian systems.

Keywords - Power transmission network design problems, metaheuristics, GRASP, Path Relinking

1 Introduction

THIS PAPER will present results obtained by a new metaheuristics approach – Greedy Randomized Adaptive Path Relinking, see [3], applied to solve long term power transmission network design problems. Greedy Randomized Adaptive Path Relinking – or just GRAPR – consists of a generalization of GRASP's basic mechanisms to manage the trajectory exploration of Path Relinking approach [14]. GRASP was formalized by Feo and Resende in [11] and its basic mechanisms are a greedy randomized construction phase, where a feasible solution is iteratively built through a greedy randomized procedure; and local search phase, where the neighborhood of the construction phase solution is explored.

The idea behind GRAPR is to improve the exploration characteristics of the Path Relinking approach, which consists in the generation of just one path to connect two "high-quality" solutions. However, a very large number of different paths exist and, hence, several different solutions could be reached exploring these different trajectories.

Static power transmission network design problems consists in choosing, from a pre-defined set of candidate circuits, those that should be built in order to minimize the investment and operational costs, and to supply the forecasted demand along a planning horizon. This problem is dynamic because it aims to determine the stage-by-stage transmission expansion plan. A subproblem of the dynamic version is the static problem, which aims to determine, for a given year in the future, *where* and *what* type of new transmission facilities should be installed, i.e. the *timing* consideration is relaxed.

This paper addresses the static version of the transmission design problem, which is also a combinatorial problem. Because of its combinatorial nature, finding an optimal solution is a very hard task. Both, combinatorial techniques and heuristic approaches could be used, but the use of combinatorial techniques is restricted to small or medium-scale instances due to its complexity. On the other hand, heuristic approaches could provide high-quality, but not necessarily optimal, solutions in an acceptable computational time, even for large-scale. However, working with good feasible solutions is not bad, because the problem being solved is just a small part of the whole, much more complex, power transmission network design problem. Several metaheuristics methods have already been proposed to deal with this problem, Simulating Annealing [17, 13], GRASP [5, 4], Tabu Search [12, 20], Genetic Algorithms [12, 7].

The paper aims to show the effectiveness of GRAPR in solving power transmission network design problems and is organized as follows: Section 2 presents the formulation of the static power transmission network design problems. Section 3 introduces the concepts of Greedy Randomized Adaptive Path Relinking and discusses its similarities and differences with its antecessors, GRASP and Path Relinking. Section 4 illustrates the results obtained by GRAPR in solving two real world case studies of power transmission network expansion problems. Finally, section 5 draws some conclusions and presents some directions for future development and investigation.

2 Static power transmission network design problems

Denoting by $\mathcal{N}$ the set of all nodes (cardinality of $\mathcal{N}$ is written as $|\mathcal{N}|$), by $\mathcal{E}$ the set of existing branches, and by $\mathcal{C}$ the set of all candidate branches that may be added to the initial network, the static long-term power transmission network design problem can be formulated as

$$\text{minimize } z = \sum_{kl \in \mathcal{C}} c_{kl} x_{kl}, \text{subject to:} \quad (1a)$$

$$\sum_{l \in \mathcal{E}_k} f^0_{kl} + \sum_{l \in \mathcal{C}_k} f^1_{kl} + g_k = d_k, \;\; k \in \mathcal{N}, \quad (1b)$$

$$f^0_{kl} - \gamma^0_{kl}\left(\theta_k - \theta_l\right) = 0, \;\; kl \in \mathcal{E}, \quad (1c)$$

$$f^1_{kl} - x_{kl}\gamma^1_{kl}\left(\theta_k - \theta_l\right) = 0, \;\; kl \in \mathcal{C}, \quad (1d)$$

$$\left|f^0_{kl}\right| \leq \overline{f}^0_{kl}, \;\; kl \in \mathcal{E}, \quad (1e)$$

$$\left|f^1_{kl}\right| \leq \overline{f}^1_{kl} x_{kl}, \;\; kl \in \mathcal{C}, \quad (1f)$$

$$0 \leq g_k \leq \overline{g}_k, \;\; k \in \mathcal{N}, \quad (1g)$$

$$x_{kl} \in \{0, 1\}, \;\; kl \in \mathcal{C}, \quad (1h)$$

where c_{kl} is the investment cost to build candidate branch kl. $\mathcal{E}_k$ and $\mathcal{C}_k$ represent the set of all existing and candidate branches directly connected with bus k. Superscripts indices ^{0} (^{1}) are references for existing (candidate) network variables, respectively. Using this notation, f_{kl} is the power flow in the branch kl, g_k is the active generation at node k, γ_{kl} states the branch susceptance for branch kl, θ_k is the k-node voltage angle, $\overline{f}_{kl}$ and $\overline{g}_k$ state, respectively, kl-branch capacity and k-node generation limit.

The objective function corresponds to the investment costs of new transmission facilities. In this formulation the operation costs are neglected, otherwise a term $\sum_{k\in\mathcal{N}} c^O_k g_k$, where c^O_k represents the unit cost of generation at bus k, must be introduced in the objective function. Constraints (1b) are the power flow balance equations for all nodes of the network, and constraints (1c) and (1d) are the linearized power flow equations for the existing and candidate network. The remaining constraints are operational limits and integrality conditions. Constraints (1h) represent the integrality conditions over the decision variables x. Note that if the kl candidate branch is not built, i.e $x_{kl} = 0$, the corresponding branch flow over this candidate branch is required to be zero because of constraint (1f). Also, the second Kirchoff law (1d) should not be enforced for this branch. On the other hand, when $x_{kl} = 1$, i.e. the kl-th candidate branch is built, the second Kirchoff law is made valid, the branch flow is limited by $\overline{f}_{kl}$ and constraint (1c) must be enforced.

The problem (1) is a mixed non-linear $(0-1)$ programming problem. Solving it by classical combinatorial optimization approaches (e.g. branch-and-bound) is very difficult. One alternative is to employ heuristic approaches, which can provide good feasible solutions, but not necessarily the optimal. Examples of heuristic approaches are greedy methods that select one candidate circuit to be built at a time, i.e. the vector x is iteratively constructed. Let the vector $\hat{x}$ represent this partial solution. If we substitute $x = \hat{x}$ in (1) we will get the following LP problem,

$$\text{minimize } \hat{z} = \sum_{k\in\mathcal{N}} r_k, \text{subject to:} \tag{2a}$$

$$\sum_{l\in\mathcal{E}_k} f^0_{kl} + \sum_{l\in\mathcal{C}_k} f^1_{kl} + g_k = d_k, \;\; k \in \mathcal{N}, \tag{2b}$$

$$f^0_{kl} - \gamma^0_{kl}\left(\theta_k - \theta_l\right) = 0, \;\; kl \in \mathcal{E}, \tag{2c}$$

$$f^1_{kl} - \hat{x}_{kl}\gamma^1_{kl}\left(\theta_k - \theta_l\right) = 0, \;\; kl \in \mathcal{C}, \tag{2d}$$

$$\left|f^0_{kl}\right| \leq \overline{f}^0_{kl}, \;\; kl \in \mathcal{E}, \tag{2e}$$

$$\left|f^1_{kl}\right| \leq \overline{f}^1_{kl}\hat{x}_{kl}, \;\; kl \in \mathcal{C}, \tag{2f}$$

$$0 \leq g_k \leq \overline{g}_k, \;\; k \in \mathcal{N}, \tag{2g}$$

$$0 \leq r_k \leq d_k \;\; k \in \mathcal{N}, \tag{2h}$$

where r_k is the unsupplied load in the k-th bus and $\hat{z}$ is the amount of unsupplied load in the network. Note than, $\hat{z}$ can be used as a measure of network infeasibility for the trial transmission expansion plan ($\hat{x}$). In case of $\hat{z} = 0$ the trial solution $\hat{x}$ is a feasible solution of problem (1), i.e. $\hat{x}$ is a feasible transmission network expansion plan.

3 Greedy randomized adaptive Path Relinking

Greedy randomized adaptive Path Relinking consists of a generalization of GRASP concepts in order to better explore paths linking two guiding solutions, or the paths endpoints. Thus, initially we will describe the GRASPs concepts used and the Path Relinking approach.

The GRASP metaheuristic [11] is a multi-start and iterative approach, in which each GRASP iteration is composed of two phases: construction and local search. The best, overall solution, is reported as the final solution. A generic pseudo code of GRASP is illustrated in Figure 1, where `MaxIter` is the number of GRASP iterations and `Seed` is the initial seed for the pseudo-random number generator.

```
procedure GRASP(MaxIter, Seed)
    ReadInput();
    for k = 1,..., MaxIter do
      x̂ ← RandomConstruction(Seed);
      x̂ ← LocalSearch(x̂);
      x̄ ← UpdateSolution(x̂);
    enddo;
    return x̄
end GRASP;
```

Figure 1: General description of GRASP.

The basic mechanisms of GRASP are construction and local search phases. In the construction phase, a feasible solution must be built by a randomized adaptive greedy algorithm. Thus, the implementation of the GRASP construction phase requires the selection of a greedy function for the problem being solved. The local search phase starts from the solution provided by the construction phase. Using an enumeration procedure the neighborhood of this solution is explored. Improvements found by this phase in the current solution should cause a restart of the local search phase.

Path Relinking was originally proposed as an intensification strategy to explore trajectories linking two elite solutions. The idea behind Path Relinking is to mix attributes of both guiding solutions, exploring the search space between them with the objective of discovering new, and better, solutions. The process of introducing arguments in a solution characterizes a movement in the neighborhood, and an iteration of Path Relinking, i.e. make a movement, check feasibility and optimality, make a movement and so forth, until the guiding solution is reached.

In the original Path Relinking approach movements are selected based on a given greedy function, i.e. all movements from a solution should be analyzed and the best one, according to a greedy function, is selected to be done. Thus, Path Relinking approach can be viewed as a greedy procedure according to the objective of exploring trajectories between two given solutions.

To improve the exploration characteristic of Path Relinking, we are proposing applying GRASP construction phase concepts to randomize the selection of what movement should be selected at each iteration of a path construction, thus introducing a degree of diversification in the search. Remark that now many trajectories can be explored linking the two guiding solutions, but computational time will also be higher. As this new approach inherits its characteristics from both GRASP and Path Relinking, we denominate it Greedy Randomized Adaptive Path Relinking, or just GRAPR.

3.1 Implementation

First we will present how we have implemented Path Relinking to solve power transmission network design problems. We use, as guiding solutions, the $i-$th iterate solution found by a GRASP approach and an elite solution selected at random from an elite set $\mathcal{E}$ ($size(\mathcal{E})$ = `EliteSize`), and insert a new phase – `PathRelinking` – in the main loop of GRASP, as illustrated in Figure 2. In line 7 the current GRASP solution is relinked with an elite solution, and vice-versa in line 8. Further, `GRASP+PR` approach needs an additional parameter, the size of elite set, and two new procedures, one to insert new solutions into the elite set, line 5, and another to select an elite solution from the elite set, line 6.

procedure `GRASP+PR(MaxIter,Seed,EliteSize)`
1 `ReadInput()`;
2 **for** $k = 1, \dots,$ `MaxIter` **do**
3 $\hat{x} \leftarrow$ `RandomConstruction(Seed)`;
4 $\hat{x} \leftarrow$ `LocalSearch(`$\hat{x}$`)`;
5 $\mathcal{E} \leftarrow$ `UpdateEliteSet(`$\hat{x}$`)`;
6 $\mathcal{E}_i \leftarrow$ `SelectSolution(`$\mathcal{E}$`)`;
7 $\hat{x}^R \leftarrow$ `PathRelinking(`$\hat{x}, \mathcal{E}_i, \hat{x}$`)`;
8 $\hat{x} \leftarrow$ `PathRelinking(`$\mathcal{E}_i, \hat{x}, \hat{x}^R$`)`;
9 $\bar{x} \leftarrow$ `UpdateSolution(`$\hat{x}$`)`;
10 **enddo**;
11 **return** $\bar{x}$
end `GRASP+PR`;

Figure 2: GRASP with Path Relinking pseudo-code.

To become an elite solution, the solution $\hat{x}$ must be either better than the best member of $\mathcal{E}$, or better than the worst member of $\mathcal{E}$ and sufficiently different from all other elite solutions (how different it must be is an user parameter). Initially this set is empty and the cost of the worst elite member is arbitrarily set to infinity, and is kept set to infinity until the first `EliteSize` elite transmission expansion plans are included in the elite set.

The first function that must be done when implementing a Path Relinking procedure is one that builds a structure of all movements that, when applied to the initial solution, will lead to the guiding solution. This function (`Diff`), line 3 of Figure 3, compares solutions $\hat{x}^S$ and $\hat{x}^T$ returning two vectors Δ^a (and Δ^r) containing the index of all candidate circuits that should be added to (or removed from) $\hat{x}^S$ to reach the solution $\hat{x}^T$.

The second key point of any Path Relinking implementation is the movement selection, which, is done in lines 6 and 10. In the first case, a candidate circuit belonging to the set of circuits that must be removed from initial $\hat{x}^S$, Δ^r, is selected to be removed. The index used to rank these movements is the investment cost, such that the removal of the most expensive candidate circuit in Δ^r is the greedy movement, made in lines 6 and 7. Following, the candidate circuit removal, the infeasibility of the resulting network must be recomputed (line 8) and, if the network remains feasible, i.e. $\hat{z} = 0$, the working solution $\hat{x}$ is candidate to be the result of `PathRelinking` procedure, lines 14 and 15.

procedure `PathRelinking(`$\hat{x}^S, \hat{x}^T, \hat{x}^R$`)`
1 $z_{min} = \min\left(\texttt{cost}(\hat{x}^S), \texttt{cost}(\hat{x}^T), \texttt{cost}(\hat{x}^R)\right)$; $\hat{x}^{min} = \hat{x}^R$;
2 $\hat{x} = \hat{x}^S$;
3 $(\Delta^a, \Delta^r) \leftarrow \texttt{Diff}(\hat{x}^S, \hat{x}^T)$;
4 **while** $\mid \Delta^a \cup \Delta^r \mid \geq 2$ **do**
5 **if** $\Delta^r = \emptyset$ **return** $\hat{x}^{min}$;
6 $kl = \arg\max_{ij \in \Delta^r} \{c_{ij}\}$;
7 $\hat{x}_{kl} = 0$, $\Delta^r = \Delta_r \backslash kl$;
8 `Solve program` (2);
9 **while** $\hat{z} > 0$ **and** $\Delta^a \neq \emptyset$ **do**
10 $kl = \arg\max_{ij \in \Delta^a} \{h_{ij}\}$;
11 $\hat{x}_{kl} = 1$, $\Delta^a = \Delta^a \backslash kl$;
12 `Solve program` (2);
13 **enddo**;
14 **if** $\hat{z} = 0$ **and** $\texttt{cost}(\hat{x}) < z_{min}$
15 $z_{min} = \texttt{cost}(\hat{x})$; $x_{min} = \hat{x}$;
16 **enddo**;
17 **return** $\hat{x}^{min}$;
end `PathRelinking`;

Figure 3: Path Relinking pseudo-code.

Once the network is infeasible, i.e. $\hat{z} > 0$, candidate circuits must be added. In order to select which addition movement should be done, line 10, we use an index based on the sensitivity of the operation problem (2) with respect to the branch susceptance, i.e. $\frac{\partial \hat{z}}{\partial \gamma}$. It was shown in [9] that one can estimate this sensitivity index by $\pi_{kl}^{\gamma} = (\pi_l^d - \pi_k^d)(\theta_k - \theta_l), \forall kl \in \mathcal{C}$, where π_k^d is the Lagrange multiplier (dual variables) of constraint (2b) in problem (2) Usually this index is negative indicating the marginal benefit of adding a new branch to the network. To take into account the investment costs, we define the greedy function h_{kl} in function of the feasibility sensitivity π_{kl}^{γ} divided by the cost of each candidate branch, i.e.

$$h_{kl} = -\frac{\pi_{kl}^{\gamma}}{c_{kl}}, \forall kl \in \mathcal{C}. \tag{3}$$

Then, the greedy choice for the addition movement in `PathRelinking` procedure is to select and add the candidate branch in Δ^a with the highest h_{kl} value, as indicated in lines 10 and 11 of Figure 3. Following the addition movement, the infeasibility of the resulting network must be recomputed (line 8) and, if the network remains infeasible a new addition movement should be done. Otherwise, if $\hat{z} = 0$, the new working solution $\hat{x}$ is again candidate to be the solution of `PathRelinking` procedure, lines 14 and 15.

Instead of taking always greedy movements, which could jeopardize the power of Path Relinking, in GRAPR the movement selection is chosen at random from the list of movements Δ^r or Δ^a. In the first case (removal movement), the candidate circuit to be removed is randomly selected from the restricted candidate list,

$$\texttt{RCL}^r = \left\{ kl \in \Delta^r \mid \overline{c} - \alpha^R \left(\overline{c} - \underline{c} \right) \leq c_{kl} \leq \overline{c} \right\}, \quad (4)$$

where α^R is a parameter, $\overline{c} = \max_{ij \in \Delta^r} \{c_{ij}\}$ and $\underline{c} = \min_{ij \in \Delta^r} \{c_{ij}\}$. we use $\alpha^R = 1$.

In the second case (addition movement), the movement selection is made based on the greedy function $h_{kl}, \forall kl \in \Delta^a$. Defining $\overline{h} = \max_{kl \in \Delta^a} (h_{kl})$ and $\underline{h} = \max_{kl \in \Delta^r} (h_{kl})$, the restricted candidate list of addition movements ($\texttt{RCL}^a$) can be computed by

$$\texttt{RCL}^a = \left\{ kl \in \Delta^a \mid \overline{h} - \alpha^R \left(\overline{h} - \underline{h} \right) \leq h_{kl} \leq \overline{h} \right\}. \quad (5)$$

Using standard GRASP concepts, the movement selection is always made at random. However, it was showed in [4] that the use of linear bias functions, instead of a random function, produces better results for a GRASP for this problem. Then, in GRAPR we also implemented a linear bias function defined as $bias(k) = \frac{1}{k}, k \in \texttt{RCL}$, where $|\texttt{RCL}|$ is the size of the $\texttt{RCL}$. Let $rank(kl)$ and $bias(rank(kl))$ denote, respectively, the rank and the value of the linear bias function for the candidate branch (kl). The probability of selecting this candidate branch from RCL is,

$$P_{kl} = \frac{bias(rank(kl))}{\sum_{(ij) \in \texttt{RCL}} bias(rank(ij))}. \quad (6)$$

Introducing these modification in the Path Relinking pseudo-code, illustrated in Figure 3, we obtain the GRAPR procedure, illustrated in Figure 4.

procedure GRAPR($bias, \hat{x}^S, \hat{x}^T, \hat{x}^R$)
$z_{min} = \min \left(\texttt{cost}(\hat{x}^S), \texttt{cost}(\hat{x}^T), \texttt{cost}(\hat{x}^R) \right)$; $\hat{x}^{min} = \hat{x}^R$;
$\hat{x} = \hat{x}^S$;
$(\Delta^a, \Delta^r) \leftarrow \texttt{Diff}(\hat{x}^S, \hat{x}^T)$;
while $| \Delta^a \cup \Delta^r | \geq 2$ **do**
 if $\Delta^r = \emptyset$ **return** $\hat{x}^{min}$;
 Build $\texttt{RCL}^r$ according to (4);
 $kl = RandSelection(bias, \texttt{RCL}^r)$;
 $\hat{x}_{kl} = 0$, $\Delta^r = \Delta_r \backslash kl$;
 Solve program (2);
 while $\hat{z} > 0$ **and** $\Delta^a \neq \emptyset$ **do**
 Build $\texttt{RCL}^a$ according to (5);
 $kl = RandSelection(bias, \texttt{RCL}^a)$;
  $\hat{x}_{kl} = 1$, $\Delta^a = \Delta^a \backslash kl$;
  Solve program (2);
 enddo;
 if $\hat{z} = 0$ **and** $\texttt{cost}(\hat{x}) < z_{min}$
  $z_{min} = \texttt{cost}(\hat{x})$; $x_{min} = \hat{x}$;
enddo;
return $\hat{x}^{min}$;
end GRAPR;

Figure 4: Greedy randomized adaptive Path Relinking pseudo-code.

4 Computational results

The GRASP for transmission network expansion planning was implemented using C and Fortran programming languages, and the results reported were obtained on a PC-Pentium III, 500MHz with 192Mbytes of memory.

Two power transmission expansion case studies will be presented to illustrate our approach. The first case study corresponds to a two-high voltage level network of the reduced southern Brazilian system. It has been discussed in many references, including [6, 4, 16, 10, 9, 18, 19]. The second case study refers to the reduced southeastern Brazilian system, which has been studied in [4, 15, 8]. Data for these case studies can be obtained by email or in reference [2].

Both, Path Relinking and GRAPR were implemented besides a GRASP approach with an elite set of solutions. Guiding solutions were the GRASP iterate solution and an elite solution, which is selected at random. To assess the effect of Path Relinking and GRAPR within a GRASP approach four cases were formulated: traditional GRASP, GRASP with Path Relinking and GRASP with GRAPR. In this last case, two instances were used, 10 and 50 iterations of GRAPR. Each case was processed 10 times, with linear and random bias function and five different initial random seeds. The number of GRASP iterations was 500, the size of the elite set was 20, GRASP α parameter was adjusted by a reactive approach using $\delta = 1$, cardinality of set $\mathcal{A} = 10$ and $k-$block value 50. Neighborhood structure in the GRASP local search was $1-$exchange. Additional details of GRASP parameters, as the reactive approach used to self-adjust α can be obtained in reference [4].

4.1 The Reduced Southern Brazilian Network

The reduced southern Brazilian network has 46 nodes (2 of them are new generation units and must be connected to the network, nodes 28 and 31), 62 existing branches and 17 new rights of way (corridors). The number of candidate circuits is 237 $(3 \times (62 + 17))$. Figure 1 gives an idea of the topology of this power system and illustrates existing circuits (solid lines). Circles and arrows are used to indicate, respectively, the main load and generation buses. If we formulated this problem as problem (1), it would have 237 binary variables, 437 linear variables, and 345 constraints, excluding bounds on variables.

The optimal solution for this case study was first published, as a best known upper bound, in [18] but it was proved to be the optimal solution in reference [6]. Its investment cost is US$\$154.26$ millions, which corresponds to the addition of 16 candidate circuits. Table 1 and Figure 5 present, respectively, the list of candidate circuits added and the resulting network where all 16 additions are represented by dashed lines. Investment costs obtained in all runs of this case study are summarized in Table 2.

Remark that in all cases the optimal solution was obtained. Analyzing the average values we can see that using linear bias function produces better results that using random bias function. Further, mixing GRASP with either Path Relinking or GRAPR also produced improvements in the average values.

Table 1: Best known solution for the reduced southern Brazilian system.

From	To	# add.	From	To	# add.
26	29	3	19	25	1
42	43	2	46	6	1
24	25	2	31	32	1
29	30	2	28	30	1
5	6	2	20	21	1

Table 2: Solution values for all GRASP cases for the reduced southern Brazilian system

	linear bias		random bias	
	avg.	best	avg.	best
GRASP	154.26	154.26	154.26	158.34
GRASP+PR	154.26	154.26	154.26	156.20
GRASP+GRAPR-10	154.26	154.26	154.26	156.98
GRASP+GRAPR-50	154.26	154.26	154.26	156.20

The average CPU time required to process all cases of GRASP was about 8.5min when linear bias function was applied, and 11.5min when using a random bias function. Using GRASP+PR causes a little increase of CPU time, in the first case it was about $10,7\text{min}$ and in the second around $14,0\text{min}$. GRASP+GRAPR, considering 10 path-constructions at each GRASP iteration, consumes around 17.2min and 21min in the second. Finally, GRAPR building 50 paths each GRASP iteration requires much more time than in the prior case study, 42.2min were required in the case of linear bias function and 43min in the case of random bias function. The differences observed in the elapsed CPU time regarding linear and random bias functions is because the construction phase using the former is much more objective than the latter.

4.2 The Reduced Southeastern Brazilian Network

The reduced Southeastern Brazilian network has 79 nodes and 155 existing branches. Figure 6 shows the network system, illustrating existing circuits (solid lines), main consuming (circles) and main generation (arrows) regions. Formulating this problem as problem (1), it would have 429 (3×143) binary variables, 821 linear variables and 663 constraints, excluding bounds on variables. This problem instance is much more difficult to solve, not only because the number of candidate circuits is higher but also because it is necessary to select candidate circuits among five different voltage levels (750kV, 500kV, 440kV, 330kV and 230kV).

The optimal solution for this case study has an investment cost of US$$422$ millions obtained with the construction of 24 candidate circuits. This solution was first published (as an upper bound) in ref. [4] but proved optimal in the work [1]. Table 3 and Figure 2 presents, respectively, the list of candidate circuits added and the resulting network where dashed lines represent the candidate circuit additions needed. Investment costs obtained in all runs of this case study are summarized in Table 4.

Table 3: Best known solution for the reduced southeastern Brazilian system.

From	To	# add.	From	To	# add.
224	227	2	210	41	2
255	259	2	220	242	2
226	242	2	220	250	1
234	237	1	221	224	1
245	253	1	245	239	1
244	245	1	226	259	1
211	246	1	226	227	1
250	251	1	207	206	1
207	209	1	249	250	1
216	215	1			

Table 4: Solution values for all GRASP cases for the reduced southeastern Brazilian system

	linear bias		random bias	
	avg.	best	avg.	best
GRASP	431.8	424	454.0	443
GRASP+PR	429.0	424	446.4	430
GRASP+GRAPR-10	427.6	424	447.6	443
GRASP+GRAPR-50	423.6	422	445.8	443

First, it can be seen that the optimal solution was found only one time, using a GRASP+GRAPR approach with 50 path generations and linear bias function. Regarding to the bias functions, the remarks are the same of the prior case study, i.e. construction phase using linear bias function produces better results than using a random bias function. Analyzing average solutions values, we can see that linking GRASP and elite solutions building just one path (case of Path Relinking) or several paths (case of GRAPR) improved the results.

Concerning to the CPU time, it was required about 25min in average to process all 5 case studies of GRASP with linear bias function and 38min to process the random bias function case studies. When GRASP and Path Relinking is used, the average CPU time increases to around 22min and 41min. Building 10 paths with GRAPR each iteration of GRASP requires about 32min in the first case and 53min in the second. Finally, building 50 paths in the GRASP+GRAPR approach consumes, in average, 42min with a linear bias function and 79min with a random bias function of CPU time.

5 Conclusions

Greedy randomized adaptive Path Relinking – GRAPR – is a new metaheuristic approach to solve combinatorial problems. It consists of a generalization of GRASP concepts in order to improve the exploration characteristics of Path Relinking approach. Instead of building just one path between two guiding solutions as in the Path Relinking approach, GRAPR randomizes the selection of movement, generating several paths. Hence, the search space is better explored.

In this work we have illustrated the results obtained by GRAPR in solving two different case studies of real world static power transmission network design problems with Brazilian network systems. In both cases, the application of GRAPR was a success. For the reduced southern Brazilian southern system, improvements made were not significant but GRAPR achieved the optimal solution. For the reduced southeastern Brazilian system, GRAPR improved the solution provided either by a GRASP or by a GRASP with Path Relinking.

Based on the results showed, we can conclude that GRAPR can be applied to solve real world instances of static power transmission network design problems. Future work will be made with the objective of reducing the CPU time required by GRAPR approach. Also, we will check if GRAPR can replace the GRASP local search phase, which is the most time consuming phase in a GRASP procedure.

References

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

[2] S. Binato. *Optimal power transmission expansion planning by Benders decomposition and cutting planes tecniques*. D.Sc. thesis, COPPE – Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brasil, Apr 2000. (in portuguese).

[3] S. Binato, H. Faria Jr., and M. G. C. Resende. Greedy randomized adaptive path relinking. In *Proceedings of the Fourth Metaheuristic International Conference*, pages 393–397, Porto, Portugal, Jul 2001.

[4] S. Binato and G. C. Oliveira. A reactive GRASP for transmission network expansion planning. In C.C. Ribeiro and P. Hansen, editors, *Essays and surveys on metaheuristics*, pages 81–100. Kluwer Academic Publishers, 2001.

[5] S. Binato, G. C. Oliveira, and J. L. Araujo. A greedy randomized adaptive search procedure for transmission network expansion planning. *Power Systems, IEEE Transactions on*, 16(No.2):247–253, May 2001.

[6] S. Binato, M. V. F. Pereira, and S. Granville. A new Benders decomposition approach to solve power transmission network design problems. *Power Systems, IEEE Transactions on*, 16(No.2):235–240, May 2001.

[7] S. Binato and S. P. Roméro. Power transmission network expansion planning by a hybrid Genetic Algorithm. In *Proceedings of the IX Latin-IberoAmerican Congress on Operations Research (IX CLAIO)*, Buenos Aires, Arg., Jul/Aug 1998.

[8] S. Binato and S. P. Roméro. Genetic Algorithms with Genetic Engineering. In *Proceedings of the Third Metaheuristic International Conference*, pages 63–68, Jul 1999.

[9] EPRI. Mathematical decomposition techniques for power system expansion planning – Analysis of the linearized power flow model using the Benders decomposition technique. Technical Report RP 2473-6, EPRI, 1988.

[10] EPRI. Mathematical decomposition techniques for power system expansion planning – Decomposition methods and uses. Technical Report RP 2473-6, EPRI, 1988.

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

[12] R. A. Galego, A. Monticelli, and R. Romero. Comparative studies on non-convex optimization methods for transmission network expansion planning. *Power Systems, IEEE Transactions on*, 13(No.3):822–828, Aug 1998.

[13] R. A. Gallego, A. B. Alves, A. Monticelli, and R. Romero. Parallel Simulated Annealing applied to long term transmission expansion planning. *Power Systems, IEEE Transactions on*, 12(No.1):181–187, Feb 1997.

[14] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39(3):653–684, 2000.

[15] G. C. Oliveira, A. P. Costa, and S. Binato. Large scale transmission network planning using optimization and heuristic techniques. *Power Systems, IEEE Transactions on*, 10(No.4):1828–1834, Nov 1995.

[16] M. V. F. Pereira, L. M. V. G. Pinto, S. H. F. Cunha, and G. C. Oliveira. A decomposition approach to automated generation-transmission expansion planning. *Power Apparatus and Systems, IEEE Transactions on*, PAS-104(No.11), Nov 1985.

[17] R. Romero, R. A. Gallego, and A. Monticelli. Transmission system expansion planning by Simmulated Annealing. In *Proceedings of the conference on Power Industry Computer Applications*, Salt Lake City, US, May 1995.

[18] R. Romero and A. Monticelli. A hierarchical decomposition approach for transmission expansion planning. *Power Systems, IEEE Transactions on*, 9(No.1):373–380, Feb 1994.

[19] R. Romero and A. Monticelli. A zero-one implicit enumeration method for optimizing investments in transmission expansion planning. *Power Systems, IEEE Transactions on*, 9(No.3), Aug 1994.

[20] E. L. Silva, J. M. Areiza, G. C. Oliveira, and S. Binato. Transmission network expansion planning under a Tabu Search approach. *Power Apparatus and Systems, IEEE Transactions on*, 16(1):62–68, Feb 2001.

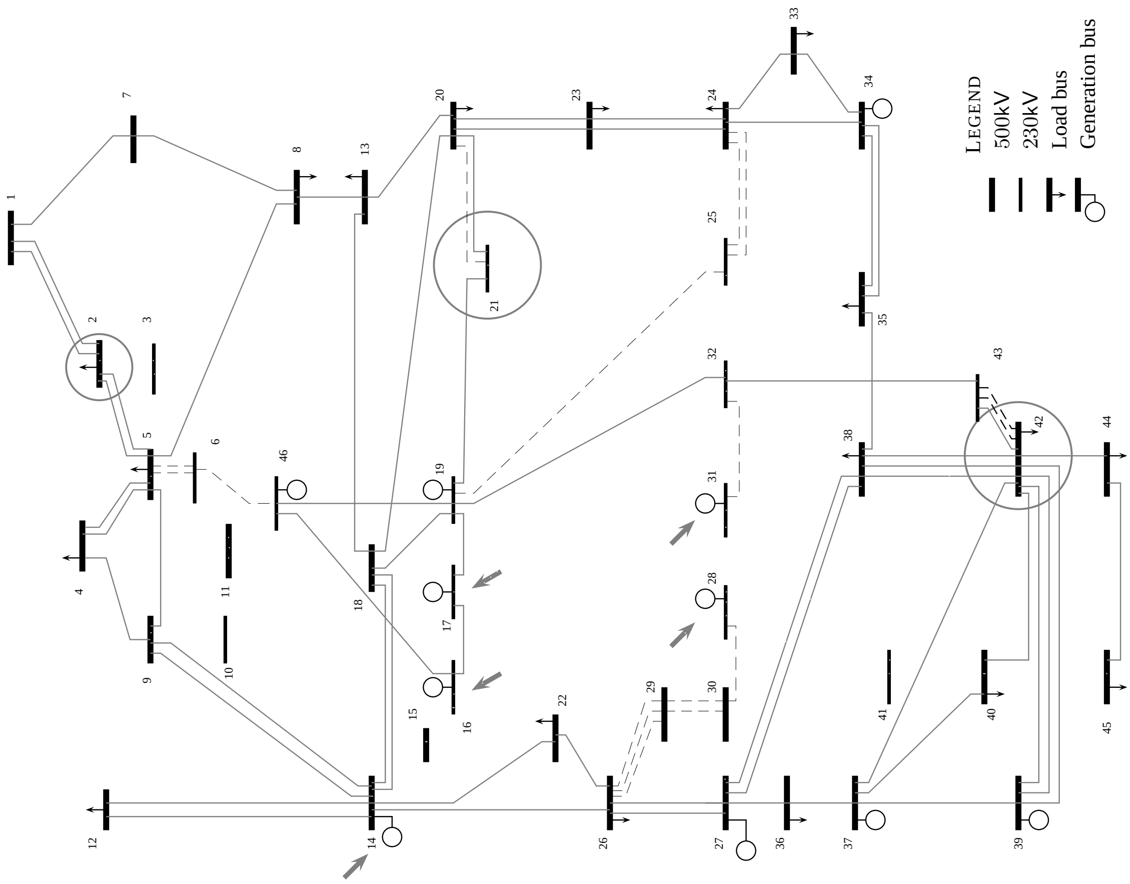

Figure 5: The reduced southern Brazilian system.

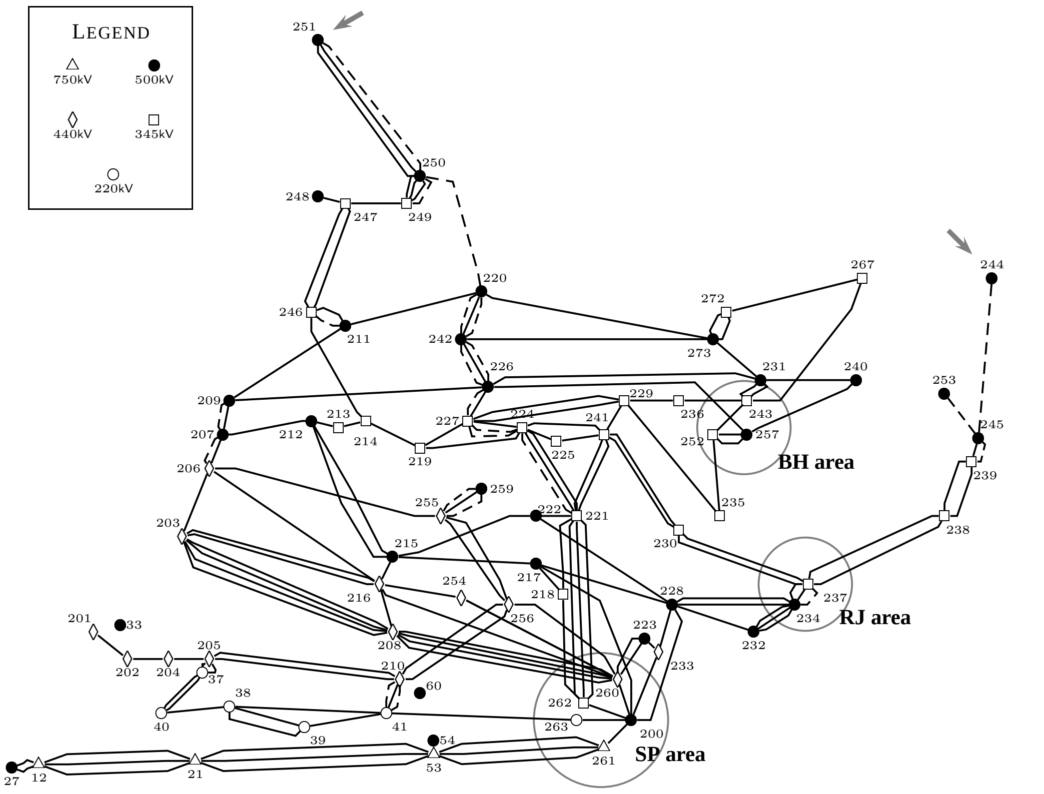

Figure 6: The reduced southeastern Brazilian power network.