Power Transmission Network Design by Greedy Randomized Adaptive Path Relinking

Haroldo Faria, Jr., *Student Member, IEEE*, Silvio Binato, Mauricio G. C. Resende, and Djalma M. Falcão, *Fellow, IEEE*

***Abstract*—This paper presents results obtained by a new metaheuristic approach called Greedy Randomized Adaptive Path Relinking (GRAPR), applied to solve static power transmission network design problems. This new approach uses generalized GRASP concepts to explore different trajectories between two "high-quality" solutions previously found. The results presented were obtained from two real-world case studies of Brazilian systems.**

***Index Terms*—GRASP, metaheuristics, path relinking, power transmission network design problems.**

I. Introduction

THIS PAPER presents results obtained by a new metaheuristic approach (Greedy Randomized Adaptive Path Relinking (GRAPR)—see [2]) applied to solve static power transmission network design problems. GRAPR consists of a generalization of GRASP's mechanisms to manage the trajectory exploration of the Path Relinking approach [12]. GRASP was formalized by Feo and Resende in [9] and its mechanisms are a greedy randomized construction phase, where a feasible solution is iteratively built through a greedy randomized procedure; and a local search phase, which explores the neighborhood of the construction phase solution.

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

The electric power system expansion planning task is a large-scale optimization problem that must devise a system meeting load and reliability criterion, minimizing, at the same time, the investment and operational costs. This is an extremely complex problem which cannot be solved without some simplifications. One of these simplifications consists in separating this problem into its major agents, namely generation, transmission and distribution. In this work, we are interested in the transmission expansion planning problem which assumes as known the generation plan.

Manuscript received January 9. 2004. Paper no. TPWRS-00200-2003.

H. Faria, Jr., is with the Fundação Centro Tecnológico de Juiz de Fora, Brazil (e-mail: hfjr@ieee.org).

S. Binato is with the PSR Consultoria, Rio de Janeiro, Brazil (e-mail: silvio@binato.com.br).

M. G. C. Resende is with AT&T Labs, Florham Park, NJ 07932 USA (e-mail: mgcr@research.att.com).

D. M. Falcão is with COPPE/UFRJ, Rio de Janeiro, Brazil (e-mail: falcao@coep.ufrj.br).

Digital Object Identifier 10.1109/TPWRS.2004.835627

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 has a dynamic nature, i.e., it requires the consideration of multiple time periods, determining a sequence (stage-by-stage) of transmission expansion plans. A subproblem of the dynamic version is the static problem, which aims to determine, for just one stage, *where* new transmission facilities should be installed, i.e., the *timing* consideration is relaxed. A suboptimal solution of the dynamic problem can be obtained by the solution of a sequence of static problems.

This paper addresses the static version of the transmission network design problem. Because of its combinatorial nature, finding an optimal solution is a very hard task. Both, combinatorial and heuristic techniques can be used, but the use of combinatorial techniques is restricted to small instances due to the complexity of these problems. On the other hand, heuristic approaches can provide "high-quality" solutions in an acceptable computational time, even for large-scale problems. In this way, several metaheuristic methods have already been proposed to deal with these problems, e.g., Simulated Annealing [11], [13], GRASP [4], [5], Reactive GRASP [4], Tabu Search [10], [16], Hybrid Tabu Search [10], [16], Genetic Algorithms [7], [10].

The paper aims to show the effectiveness of GRAPR in solving power transmission network design problems. We include results from two real-world, medium-scale, case studies of Brazilian systems already analyzed in the literature [4].

This paper is organized as follows: Section II presents the formulation of the transmission network design problems. Section III introduces the concepts of GRAPR and discusses its similarities and differences with GRASP and Path Relinking. Section IV illustrates the results obtained by GRAPR and, finally, in Section V we discuss some conclusions.

II. Static Power Transmission Network Design Problems

Denoting the set of all nodes by $\mathcal{N}$ (size($\mathcal{N}$) is written as $|\mathcal{N}|$), the set of all existing branches by $\mathcal{E}$, and the set of all candidate branches by $\mathcal{C}$, the static long-term power transmission network design problem, which is a simplified version of the complete model (see [3]) can be formulated as

$$\text{minimize } z = \sum_{kl \in \mathcal{C}} c_{kl} x_{kl} \tag{1a}$$

$$\text{subject to :} \tag{1b}$$

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

$$f_{kl}^0 - \gamma_{kl}^0(\theta_k - \theta_l) = 0, \quad kl \in \mathcal{E} \tag{1d}$$
$$f_{kl}^1 - x_{kl}\gamma_{kl}^1(\theta_k - \theta_l) = 0, \quad kl \in \mathcal{C} \tag{1e}$$
$$\left|f_{kl}^0\right| \le \bar{f}_{kl}^0, \quad kl \in \mathcal{E} \tag{1f}$$
$$\left|f_{kl}^1\right| \le \bar{f}_{kl}^1 x_{kl}, \quad kl \in \mathcal{C} \tag{1g}$$
$$0 \le g_k \le \bar{g}_k, \quad k \in \mathcal{N}, \tag{1h}$$
$$x_{kl} \in \{0,1\}, \quad kl \in \mathcal{C} \tag{1i}$$

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

The objective function corresponds to the investment cost of new transmission facilities. In this formulation the operation costs are neglected, otherwise a term $\sum_{k\in\mathcal{N}} c_k^O g_k$, where c_k^O represents the unit cost of generation at bus k, must be introduced. However, considering operational costs does not affect substantially the formulation and performance of the proposed approach. The same observation applies to the inclusion of a capital recovery factor in the objective function. The consideration of other criteria like reliability, dynamic security, etc., leading to a multicriteria formulation is possible but beyond the scope of this paper.

Constraints (1c) are the power flow balance equations for all nodes of the network, and constraints (1d) and (1e) are the linearized power flow equations for the existing and candidate network, respectively. The remaining constraints are operational limits and integrality conditions. Constraints (1i) represent the integrality conditions over the decision variables x_{kl}. 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 (1g). Also, the second Kirchoff law (1e) should not be enforced for this branch. On the other hand, when $x_{kl} = 1$, i.e., the klth candidate branch is built, the second Kirchoff law is made valid, the branch flow is limited by $\bar{f}_{kl}$ and constraint (1d) must be enforced.

Problem (1a)–(1i), throughout referred as problem (1), is a mixed nonlinear $(0-1)$ 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 the problem (1), we will get the following LP problem,

$$\text{minimize } \hat{z} = \sum_{k\in\mathcal{N}} r_k, \quad \text{subject to:} \tag{2a}$$
$$\sum_{l\in\mathcal{E}_k} f_{kl}^0 + \sum_{l\in\mathcal{C}_k} f_{kl}^1 + g_k = d_k, \quad k \in \mathcal{N} \tag{2b}$$
$$f_{kl}^0 - \gamma_{kl}^0(\theta_k - \theta_l) = 0, \quad kl \in \mathcal{E} \tag{2c}$$
$$f_{kl}^1 - \hat{x}_{kl}\gamma_{kl}^1(\theta_k - \theta_l) = 0, \quad kl \in \mathcal{C} \tag{2d}$$
$$\left|f_{kl}^0\right| \le \bar{f}_{kl}^0, \quad kl \in \mathcal{E} \tag{2e}$$
$$\left|f_{kl}^1\right| \le \bar{f}_{kl}^1 \hat{x}_{kl}, \quad kl \in \mathcal{C} \tag{2f}$$
$$0 \le g_k \le \bar{g}_k, \quad k \in \mathcal{N} \tag{2g}$$
$$0 \le r_k \le d_k \quad k \in \mathcal{N} \tag{2h}$$

where r_k is the unsupplied load in the kth bus and $\hat{z}$ is the amount of unsupplied load in the network. This operation model is referred as problem (2) throughout of this paper.

Transmission losses were not modeled explicitly in problem (1), but they can be included in the operation problem as additional loads, which are evaluated as a quadratic function of power flow, into terminal nodes of each network branch. This approach to take into account transmission losses requires the iterative solution of problem (2) until the convergence of branch losses. The production costs for the Brazilian systems studied in this work are neglected because of their essentially hydroelectric nature.

Finally, note that $\hat{z}$ can be used as a measure of network infeasibility for the trial transmission expansion plan $(\hat{x})$. In the case that $\hat{z} = 0$ the trial solution $\hat{x}$ is a feasible solution of problem (1), i.e., $\hat{x}$ is a feasible transmission expansion plan.

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

Fig. 1. General description of GRASP.

III. Greedy Randomized Adaptive Path Relinking

GRAPR consists of a generalization of GRASP concepts applied to the Path Relinking framework to better explore paths linking two guiding solutions in the search space. For a better understanding of the new approach, we will first describe the basic mechanisms of GRASP and the Path Relinking approach.

The GRASP metaheuristic [9] is a multistart iterative approach, in which each GRASP iteration is composed of two phases: construction and local search. The best solution found over all iterations is reported as the final result. A generic pseudocode of GRASP is illustrated in Fig. 1, where `MaxIter` is the number of GRASP iterations to be performed and `Seed` is the initial seed for the pseudorandom number generator.

The basic mechanisms of GRASP are the construction and local search phases. In the construction phase, a feasible solution is iteratively 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 a local search 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 by Glover [12] as an intensification strategy to explore trajectories linking two elite solutions. The idea behind Path Relinking is to mix attributes of two 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 its neighborhood, and an iteration of Path Relinking, which consists in making a movement and checking the feasibility and optimality of the new solution obtained repeating this process, until the guiding solution is reached.

In the original Path Relinking approach, movements are selected based on a given greedy function, i.e., all possible movements from a solution should be analyzed and the best one, according to a greedy function, is selected to be done. Thus, the 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 propose the application of GRASP construction phase concepts to randomize the selection of which movement should be selected at each iteration of a path construction, 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.

A. Implementation

First, we discuss how we have implemented Path Relinking to solve power transmission network design problems. We use, as guiding solutions, the ith iterate solution found by a GRASP approach and an elite solution selected at random from an elite set $\mathcal{E}(|\mathcal{E}| = \texttt{EliteSize})$, and insert a new phase—`PathRelinking`—in the main loop of GRASP, as illustrated in Fig. 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.

To become an elite solution, 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 a 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.

A function that must be devised 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 Fig. 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 line 6, a candidate circuit belonging to the set of circuits that

procedure `GRASP+PR(MaxIter,Seed,EliteSize)`

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

Fig. 2. GRASP with Path Relinking pseudocode.

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

Fig. 3. Path Relinking pseudocode.

must be removed from the initial solution $\hat{x}^S, \Delta^r$, 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 done in lines 6 and 7. Following the candidate circuit removal, the infeasibility of the resulting network must be checked (line 8). If the network remains feasible, i.e., $\hat{z} = 0$, the working solution, or a trial transmission expansion plan, $\hat{x}$, is candidate to be the result of the `PathRelinking` procedure in lines 13 and 14.

If $\hat{z} > 0$, indicating an infeasible network, candidate circuits must be added. In order to select which addition movement should be done in line 10, we use an index based on the sensitivity of the operation problem (2) with respect to the branch susceptance, i.e., $(\partial\hat{z}/\partial\gamma)$. It was shown in [8] that one can estimate this sensitivity index by $\pi^{\gamma}_{kl} = (\pi^d_l - \pi^d_k)(\theta_k - \theta_l), \forall kl \in \mathcal{C}$, where π^d_k 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} as the feasibility sensitivity π^{γ}_{kl} divided by the cost of each candidate branch, i.e.,

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

The greedy addition movement in the `PathRelinking` procedure is the selection and addition of the candidate branch in Δ^a with the highest h_{kl} value, as indicated in lines 10 and 11 of Fig. 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 the `PathRelinking`, lines 13 and 14.

Instead of always making greedy movements, which could jeopardize the power of Path Relinking, GRAPR selects movements 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 ($\mathtt{RCL}^r$)

$$\mathtt{RCL}^r = \{kl \in \Delta^r \mid \bar{c} - \alpha^r(\bar{c} - \underline{c}) \leq c_{kl} \leq \bar{c}\} \tag{4}$$

where $\alpha^r \in [0, 1]$ is a parameter, $\bar{c} = \max_{ij \in \Delta^r}\{c_{ij}\}$ and $\underline{c} = \min_{ij \in \Delta^r}\{c_{ij}\}$. We set $\alpha^r = 1$.

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

$$\mathtt{RCL}^a = \{kl \in \Delta^a \mid \bar{h} - \alpha^a(\bar{h} - \underline{h}) \leq h_{kl} \leq \bar{h}\} \tag{5}$$

where α^a is also set to 1.

Using standard GRASP concepts, the movement selection is always made at random. However, it was shown in [4] that the use of a linear bias function, instead of a random function, produces better results for a GRASP for this type of problem, since it bias the search toward the best candidate branches. In our GRAPR algorithm, we also implemented a linear bias function defined as $\text{bias}(k) = (1/k), k \in \mathtt{RCL}$, where $|\mathtt{RCL}|$ is the size of the `RCL`. Let $\text{rank}(kl)$ and $\text{bias}(\text{rank}(kl))$ 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 the `RCL` is

$$P_{kl} = \frac{\text{bias}(\text{rank}(kl))}{\sum_{(ij) \in \mathtt{RCL}} \text{bias}(\text{rank}(ij))}. \tag{6}$$

Introducing these modifications in the Path Relinking pseudocode, illustrated in Fig. 3, we obtain the GRAPR procedure, illustrated in Fig. 4.

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

Fig. 4. GRAPR pseudocode.

IV. 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, 500 MHz with 192 Mbytes 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 [4], [6], [15]. The second case study refers to the reduced southeastern Brazilian system, which has been studied in [4].

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 from an elite set. To assess the effect of Path Relinking and GRAPR within a GRASP approach, four case studies 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, the 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 well as the reactive approach used to self-adjust α can be obtained in [4].

A. 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 $(3 \times (62 + 17))$. Fig. 5 gives an idea of the topology of this power system and illustrates existing circuits (solid lines). Main load buses are indicated by circles in the figure, while the main generators are located in buses 14, 16, 17, and 19; and new generation units must be connected to the main network in buses 28 and 31. 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 [14], but proved to be the optimal solution in [6]. Its investment cost is of US$154.26 mil-

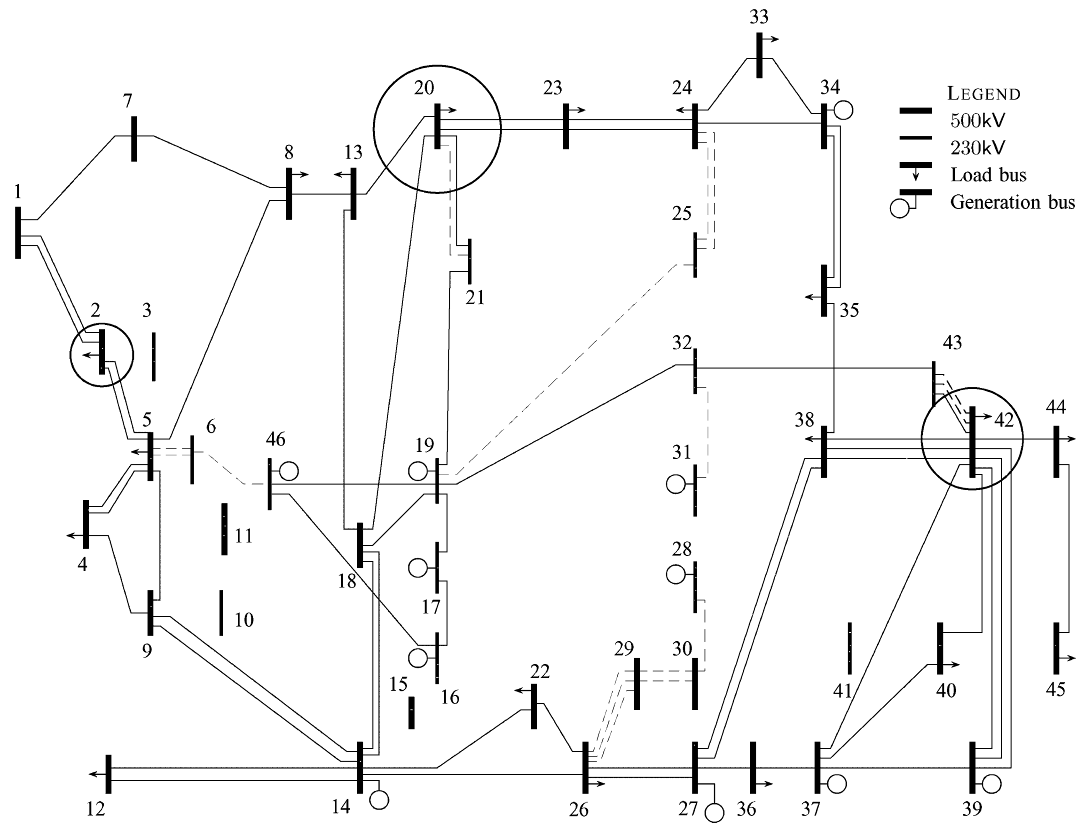

Fig. 5. The reduced southern Brazilian system.

TABLE I
BEST KNOWN SOLUTION FOR THE REDUCED SOUTHERN BRAZILIAN SYSTEM

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

TABLE II
SOLUTION VALUES FOR ALL GRASP CASES FOR THE REDUCED SOUTHERN BRAZILIAN SYSTEM

	linear bias			random bias		
	avg.	best	variance	avg.	best	variance
GRASP	154	154	0.0	158	154	11.12
GRASP+PR	154	154	0.0	156	154	8.97
GRASP+GRAPR-10	154	154	0.0	157	154	13.9
GRASP+GRAPR-50	154	154	0.0	156	154	11.29

lion, which corresponds to the addition of 16 candidate circuits. Table I and Fig. 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 II.

Remark that in all cases the optimal solution was obtained. Analyzing the average values we can see that the usage of a linear bias function produces better results than those obtained using random bias function. Mixing GRASP with either Path Relinking or GRAPR also produced improvements in the average values. The proposed method constructing either 10 or 50 paths found the optimal solution earlier in the search than the other methods analyzed when using linear bias function.

The average CPU time required to process all cases of GRASP was about 8.5 min when linear bias function was applied, and 11.5 min when using a random bias function. GRASP + PR causes a little increase of CPU time. In the first case it was about 10.7 min and in the second around 14.0 min. GRASP + GRAPR, considering 10 path-constructions at each GRASP iteration, consumes around 16.7 min and 20.0 min using linear and random bias function respectively. Finally, GRAPR building 50 paths each GRASP iteration requires much more time than in the prior case study, 20.0 min were required in the case of linear bias function and 23.0 min in the case of random bias function. The differences of CPU time observed regarding linear and random bias functions are due to the more efficient construction phase used in the former approach.

B. The Reduced Southeastern Brazilian Network

The reduced Southeastern Brazilian network has 79 nodes and 155 existing branches. Fig. 6 shows the network, illustrating existing circuits by solid lines and main consuming buses by circles. The main generation units are located in buses 21, 27, 203, 211, and 251; and new generation units must be integrated to the main network at buses 244 and 253. 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

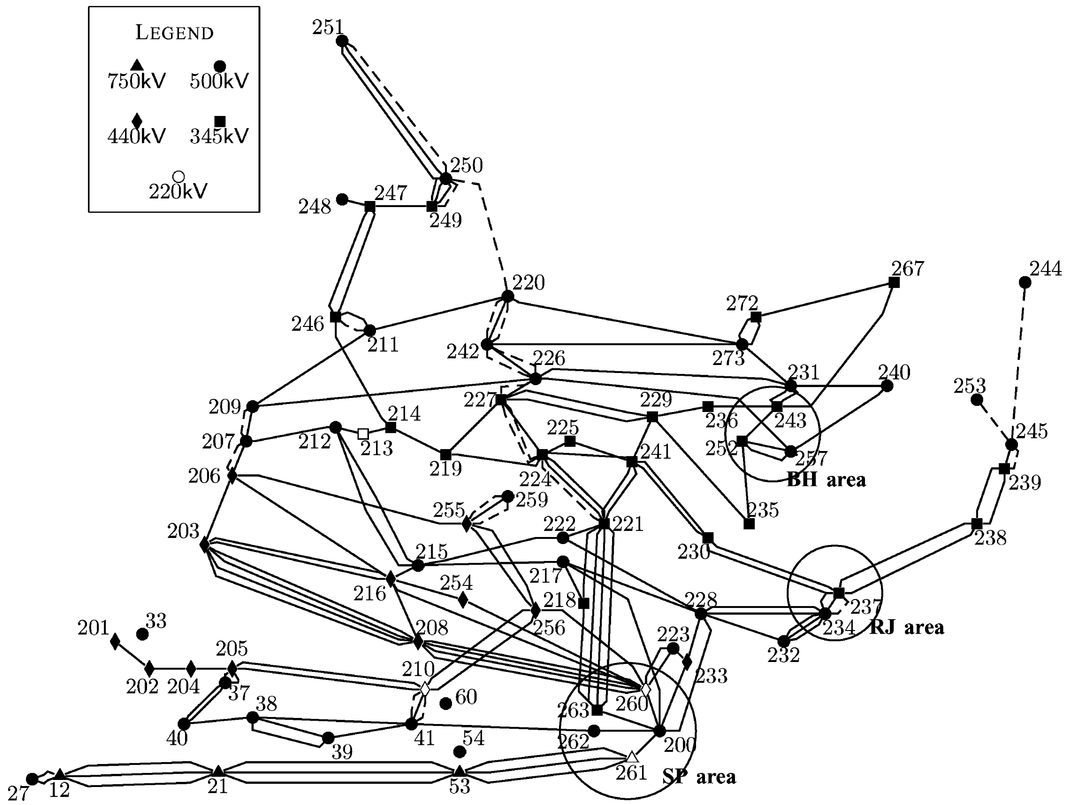

Fig. 6. The reduced southeastern Brazilian power network.

TABLE III
BEST-KNOWN SOLUTION FOR THE REDUCED SOUTHEASTERN BRAZILIAN SYSTEM

Fr	To	#Ad	Fr	To	#Ad	Fr	To	#Ad
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 IV
SOLUTION VALUES FOR ALL GRASP CASES FOR THE REDUCED SOUTHEASTERN BRAZILIAN SYSTEM

	linear bias			random bias		
	avg.	best	variance	avg.	best	variance
GRASP	431.8	424	124.2	454.0	443	65.5
GRASP+PR	429.0	424	68	446.4	430	144.8
GRASP+GRAPR-10	427.6	424	10.8	447.6	443	67.7
GRASP+GRAPR-50	423.6	422	0.8	445.8	443	9.2

also because it is necessary to select candidate circuits among five different voltage levels (750, 500, 440, 330, and 230 kV).

The optimal solution for this case study has an investment cost of US$422 million obtained with the construction of 24 candidate circuits. This solution was first published (as an upper bound) in [4] but proved optimal in [1]. Table III and Fig. 6 present, 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 IV.

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 solution values, we can see that linking GRASP and elite solutions building just one path (Path Relinking) or several paths (GRAPR) improved the results.

Concerning CPU time, about 25 min were required on average to process all 5 case studies of GRASP with linear bias function and 39 min to process the random bias function case studies. When GRASP and Path Relinking were used, the average CPU time increases to around 26 min and 40 min, respectively. Building 10 paths with GRAPR at each iteration of GRASP requires about 34 min in the first case and 53 min in the second. Finally, building 50 paths in the GRASP + GRAPR approach consumes, on average, 37 min of CPU time with a linear bias function and 56 min with a random bias function.

With a modest increase of 12 min of CPU time, on average, to process all five case studies, in comparison to a pure GRASP algorithm with linear bias function, the GRASP + GRAPR algorithm with linear bias function was able to obtain the optimum solution and cause a reduction of the total investment cost of about US$2 million.

Although statistics on the variance of results in successive runs is shown, we can not assess the robustness of the algorithm due to the small sample used in the studies.

V. CONCLUSION

GRAPR is a new metaheuristic approach to solve combinatorial optimization problems. It consists of a generalization of

GRASP concepts in order to improve the exploration characteristics of the Path Relinking approach. Instead of building just one path between two guiding solutions (Path Relinking approach), GRAPR randomizes the selection of movements, generating several paths. Hence, the search space is better explored.

In this work, we have presented 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 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 GRASP or GRASP with Path Relinking.

Based on the results shown, we can conclude that GRAPR can be applied to solve real-world instances of static power transmission network design problems. Future work will be done with the objective of reducing the CPU time required by the GRAPR approach. Besides that, 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 *et al.*, *IEEE Trans. Power Syst.*, vol. 16, pp. 560–565, Aug. 2001.

[2] S. Binato *et al.*, "Greedy randomized adaptive path relinking," in *Proc. 4th Metaheuristic Int. Conf.*, Porto, Portugal, July 2001, pp. 393–397.

[3] S. Binato and G. C. Oliveira, "A heuristic approach to cope with multiyear transmission expansion planning," in *Proc. Stockholm Power Tech. Conf.*, Stockholm, Sweden, 1995.

[4] ——, "A reactive GRASP for transmission network expansion planning," in *Essays and Surveys on Metaheuristics*, C. C. Ribeiro and P. Hansen, Eds. Norwell, MA: Kluwer, 2001, pp. 81–100.

[5] S. Binato, G. C. Oliveira, and J. L. Araujo, "A greedy randomized adaptive search procedure for transmission network expansion planning," *IEEE Trans. Power Syst.*, vol. 16, pp. 247–253, May 2001.

[6] S. Binato *et al.*, "A new Benders decomposition approach to solve power transmission network design problems," *IEEE Trans. Power Syst.*, vol. 16, pp. 235–240, May 2001.

[7] S. Binato and S. P. Roméro, "Power transmission network expansion planning by a hybrid genetic algorithm," in *Proc. IX Latin-IberoAmerican Congress on Operations Research (IX CLAIO)*, Buenos Aires, Argentina, July/Aug. 1998.

[8] EPRI, "Mathematical Decomposition Techniques for Power System Expansion Planning—Analysis of the Linearized Power Flow Model Using the Benders Decomposition Technique," Tech. Rep. RP 2473-6, 1988.

[9] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *J. Global Optimiz.*, vol. 6, pp. 109–133, 1995.

[10] R. A. Galego *et al.*, "Comparative studies on nonconvex optimization methods for transmission network expansion planning," *IEEE Trans. Power Syst.*, vol. 13, pp. 822–828, Aug. 1998.

[11] R. A. Gallego *et al.*, "Parallel simulated annealing applied to long term transmission expansion planning," *IEEE Trans. Power Syst.*, vol. 12, pp. 181–187, Feb. 1997.

[12] F. Glover, "Tabu search and adaptive memory programing—Advances, applications, and challenges," in *Interfaces in Computer Science and Operations Research*, R. S. Barr, R. V. Helgason, and J. L. Kennington, Eds. Norwell, MA: Kluwer, 1996, pp. 1–75.

[13] R. Romero *et al.*, "Transmission system expansion planning by simmulated annealing," *IEEE Trans. Power Syst.*, vol. 11, pp. 364–369, 1996.

[14] R. Romero and A. Monticelli, "A hierarchical decomposition approach for transmission expansion planning," *IEEE Trans. Power Syst.*, vol. 9, pp. 373–380, Feb. 1994.

[15] ——, "A zero-one implicit enumeration method for optimizing investments in transmission expansion planning," *IEEE Trans. Power Syst.*, vol. 9, pp. 1385–1391, Aug. 1994.

[16] E. L. Silva *et al.*, "Transmission network expansion planning under a Tabu search approach," *IEEE Trans. Power App. Syst.*, vol. 16, pp. 62–68, Feb. 2001.

Haroldo Faria, Jr. (S'95) received the B.Sc. degree in electrical engineering from the Federal University of Juiz de Fora, Juiz de Fora, Brazil, in 1996 and the M.Sc. degree in electrical engineering from Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, in 1998, where he is currenlty pursuing the D.Sc. degree.

He is presently working on the development of power systems planning and operation software.

Silvio Binato received the B.Sc. degree in electrical engineering (power systems) from the Federal University of Santa Maria, Santa Maria, Brazil, in 1988, the M.Sc. degree in electrical engineering from the Catholic University of Rio, Rio, Brazil, in 1992, and the D.Sc. degree in system and computing engineering from Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, in 2000.

He joined PSR Consultoria, Rio de Janeiro, in 2003, where he is involved in the development of computational tools using stochastic optimization and integer programming for expansion planning of hydroelectric system. Previously, he worked at CEPEL with power transmission network expansion planning, metaheuristics to solve combinatorial problems, and stochastic optimization.

Mauricio G. C. Resende received the B.Sc. degree in electrical engineering from the Catholic University of Rio de Janeiro (PUC-Rio), Brazil, in 1978, the M.Sc. degree in operations research from Georgia Tech, Atlanta, in 1979, and the Ph.D. degree in operations research from the University of California at Berkeley in 1987.

Since then, he has been with AT&T Bell Labs and AT&T Labs, Florham Park, NJ, in the Algorithms and Optimization Research Department.

Djalma M. Falcão (M'75–SM'96–F'04) received the B.Sc. degree in 1971 from the from Federal University of Paraná, Paraná, Brazil, the M.Sc. degree in 1974 from the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, and the Ph.D. degree in 1981 from the University of Manchester Institute of Science and Technology, Manchester, U.K.

He is currently a Professor of Electrical Engineering at the Federal University of Rio de Janeiro. From 1991 to 1993, he was on sabbatical leave at the University of California at Berkeley. His general research interest is in the area of computer methods for power system simulation, optimization, and control.