A comparison of tabu search and GRASP for the switch allocation problem

Alexander Benavides[1], Mariane S. Machado[1], Alysson M. Costa[2],
Marcus Ritt[1], Luciana S. Buriol[1], Vinícius J. Garcia[3] e Paulo M. França[4]

[1] Instituto de Informática - Universidade Federal do Rio Grande do Sul - UFRGS
[2] Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo - USP
[3] Universidade Federal do Pampa - UNIPAMPA
[4] Faculdade de Ciências e Tecnologia - Universidade Estadual Paulista Júlio de Mesquita Filho - UNESP
{ajbenavides, mmachado, mrpritt, buriol}@inf.ufrgs.br,
alysson@icmc.usp.br, viniciusgarcia@unipampa.edu.br, paulo.morelato@fct.unesp.br

Abstract

We study the problem of allocating automatic switches in electrical networks in order to improve their reliability. Our approach combines the solution of the switch allocation problem with the related subproblem of optimal network reconfiguration. This paper presents a GRASP for solving this joint problem, as well as a faster method for the evaluation of the electrical constraints. We compare this method to a tabu search applied on two sets of instances, a set of known instances from the literature and a set of synthetically generated instances. Our results show that in general GRASP results are slightly better compared to the tabu search.

Keywords: Reliability, switch allocation, electrical networks, GRASP, tabu search.
Paper topics: Applications to Energy (AE), Metaheuristics (MH).

1 Introduction

In order to avoid large scale blackouts, methods for improving the reliability of electrical power systems have been studied over the years. According to Teng and Liu (2003), most of the faults take place in the distribution network of electrical power systems.

Electrical power systems are built as interconnected networks. They are arranged to be radial in normal operating conditions. They are divided in three subsystems: generation, transmission, and distribution. An example of distribution network is presented in Figure 1. It is composed by distribution substations (black nodes), consumers (white nodes), and feeders. The dotted lines are switched tie-lines and are disconnected under normal operation conditions. The radial topology may change by opening and closing some feeders in order to isolate failures and serve affected consumers by alternate feeders. This is called network reconfiguration.

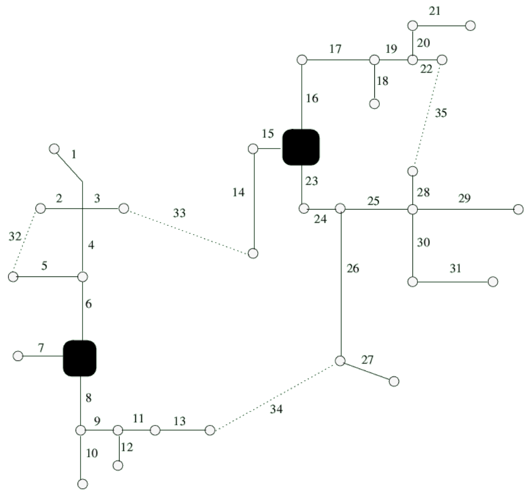

Figure 1. Distribution Network Example

Some characteristics of the network are as follows. Each consumer has a power demand and each substation has a power capacity. Each feeder has a resistance, reactance, and a current flow. Each one of those feeders may have one switch. The initial topology has open/closed switches generating a radial configuration. For new topology reconfigurations, electrical limitations must be respected.

There are different approaches to improve the reliability of electrical power systems. The most common way is to add redundant connections with switches and thus easily alter the network topology in case of failures. The costs of implementing automatic switches all over the network are impracticable due to high costs. Because of that, the places where switches should be installed, must be carefully chosen. This problem is called the switch allocation problem.

In this work, we present a GRASP and an extension to the tabu search algorithm proposed by Costa et al. (2007) to solve the switch allocation problem. The remainder of the paper is organized as follows. In Section 2 we explain reconfiguration and allocation problems. In Section 3 we present the network reliability evaluation methods. In Section 4 we describe a GRASP for the switch allocation problem. In Section 5 we describe a tabu search algorithm for the switch allocation problem. In Section 6 we propose an electrical distribution system instance generator used to test our algorithms. In Section 7 we show some computational results. Concluding remarks are given in Section 8.

2 The Switch Reconfiguration and Allocation Problems in Power Distribution Systems

This section presents the two main optimization problems related to switches and the maximization of the power distribution system reliability.

2.1 The switch reconfiguration problem

In case of a power failure, some switches are opened to isolate the failure. Afterwards, other switches can be closed to reconnect the areas which do not have neither failure nor power supply. We can easily understand how switch reconfiguration reduces the unattended area considering the example given in Figure 1. Consider a failure in feeder 17. If there are no switches in the subtree under feeder 16, the whole branch would be unattended. Now, assume there are automatic switches in feeders 16, 19, and 35. We can isolate the failure by opening the switches 16 and 19. Finally, we can restore the service of some consumers by closing the switch 35 which is open in normal operating conditions.

In this case, the optimal solution is easy to be found, but in real problems we must consider other issues, for example there are not installed switches in every feeder and electrical constraints must be respected (such as feeder and substation capacities and acceptable voltage drop).

The reconfiguration problem aims to serve as many consumers as possible, considering network restrictions. It is a complex non-linear combinatorial problem (Thakur and Jaswanti, 2006). This problem could present different forms considering any of the following objectives: maximize the reliability of the network, losses reduction to reduce overall system power loss, load balancing to avoid overloads, independence between the initial and final switches configuration, minimize maintenance operations, reduce reconfiguration costs, on line reconfiguration subject to variable demand from commercial, residential and/or industrial consumers, etc.

This problem has been studied extensively in the literature. Among the metaheuristics proposed to solve it are simulated annealing (Jeon et al., 2002; Santander et al., 2005), tabu search (Zhang et al., 2005; Zhang, Fu and Zhang, 2007), genetic algorithms (Delbem et al., 2005; Carreno et al., 2007), ant colony optimization (Su et al., 2005; Khoa and Binh, 2006), particle swarm optimization (Zhang, Zhang and Gu, 2007; Wu et al., 2007), and plant growth simulation algorithm (Wang and Cheng, 2008; Wang et al., 2008).

The present paper considers this problem as a subproblem of the switch allocation problem. The objective function is to maximize the attended demand after a failure.

2.2 The switch allocation problem

According to Billinton and Jonnavithula (1996), switches play a key role in the reliability of a power distribution system. The service restoration capability is directly related to the number and position of the switches in the network. The installation of automatic switches in the distribution system allows a better and faster reconfiguration, and hence increases reliability. Since automatic switches have a considerable cost, installing one at every feeder is not possible. Therefore, the adequate selection of their locations is very important in system planning. The problem of selecting locations to install switches in a distribution network is called the switch allocation problem.

The problem consists in finding a set of feeders to install i new switches such that the network reliability is maximized. In this paper we measure the reliability as the average percentage of attended demands over all single feeder failures for given failure probabilities. Note that, differently to the reconfiguration problem, we must consider every possible fault, reconfigure and evaluate the respective load that is still possible to attend, and finally calculate the average of the attended demand.

Hence, the optimization objective can be defined as to minimize the non supplied power areas in the case of network power failures, subject to the available number of switches for allocation and the electrical constraints of the embedded reconfiguration problem.

This problem has been studied by several authors with different approaches. Billinton and Jonnavithula (1996) propose a simulated annealing approach. They consider the investment, maintenance and outage costs in a single global cost function to determine the best number and location of switches. Similar costs evaluations for the objective function are found in other works with genetic algorithm and a tabu search by da Silva et al. (2004, 2008) and a three state particle swarm optimization by Moradi and Fotuhi-Firuzabad (2008). These works assume that, given a fault, it is easy to determine the operating network after the reconfiguration process and they approximate the overall cost of the fault.

Carvalho et al. (2005) presented a divide-and-conquer approach. They use an exhaustive evaluation of the possible failure reconfigurations to compute the reliability. They reduce the problem complexity by using a polynomial-time partitioning algorithm to divide the set of possible location places into several independent subsets or subproblems to be solved by a greedy algorithm.

3 Network Reliability Evaluation

In the reconfiguration problem it must be decided which switches to open or close in order to isolate the fault and recover as much non-served load as possible. Solutions for the switch allocation problem based on the reconfiguration problem are interesting because both problems are intrinsically correlated. If treated separately, an apparently good solution of the switch allocation problem might be negatively affected when checking the reconfiguration problem for that distribution. Due to the complexity of both switch problems and the importance of considering them together, Costa et al. (2007, 2008) proposed a solution to the switch allocation problem based on the reconfiguration problem. They proposed two approaches to evaluate the network reliability in the case of failures for a given distribution of switches in the network. These two approaches differ in the failure recovery algorithm. They consider the network connectivity (referred as reliability upper bound) and an electrical restrictions evaluation (referred as reliability lower bound). In this section we explain briefly both approaches and a modification to accelerate the electrical restriction evaluation.

The common part of the network reliability evaluation is described in Algorithm 1. The algorithm calculates a weighted percentage of served demands evaluating every possible single feeder failure. For each failure it executes three steps. First, it expands the failure (lines $3-6$) marking the edges and nodes with failure. Second, it recovers the served area with one of the recovery algorithms explained later in this section. This recovery algorithm marks the served consumers as attended. Third, it calculates the percentage of attended consumers (lines $8-14$). Finally, it calculates the average served demand for all closed feeders in normal operating conditions.

Algorithm 1 Network Reliability Evaluation

```
Input: Distribution Network, Installed switch positions
 1: for all feeders closed in normal operation do
 2:     simulate a failure in the feeder $f$
 3:     repeat
 4:         Expand the failure to nearby feeders without switches
 5:         Mark the involved nodes with failure
 6:     until the failure area is isolated by switches
 7:     Restore non served areas with a recovery algorithm marking consumer nodes as $attended$
 8:     for all consumer nodes do
 9:         if consumer node is attended then
10:             $Served \leftarrow Served + ConsumerDemand$
11:         end if
12:     end for
13:     $ServedPercentage \leftarrow Served/TotalNetworkDemand$
14:     $Total \leftarrow Total + ServedPercentage$
15: end for
16: return $Total/$ Number of feeders closed in normal operation
```

3.1 Reliability upper bound

An upper bound for the network reliability can be obtained by dropping the electrical constraints and evaluating the demand that can possibly be served based only on the network connectivity (Costa et al., 2007). To evaluate the upper bound, we expand the connectivity from the substations, to the neighbours without failure that had not been attended, through the feeders closed in normal operation or feeders without switches. This can be achieved by a breadth-first search after the failure isolation.

3.2 Reliability lower bound

We can find a lower bound for the network reliability by considering the electrical feasibility test proposed by Costa et al. (2008) and calculating an underestimate of the restored area. Algorithm 2 presents a modification (lines $2-4$) to the lower bound recovery algorithm proposed by them. We expand optimistically the test sector considering as frontier only normally open switches, i.e. switches closed in normal operation are treated as connected in the initial test sector. If that test sector is not feasible, the recovery algorithm restarts as proposed by them, with the smallest test sector considering any switch as frontier (lines $5-6$). If the test sector is still feasible, the involved nodes are marked as attended and the frontier feeders in L are processed. The test sector is expanded by closing a frontier feeder, and the feasibility is reevaluated. Again, if feasible, the involved nodes are marked as attended and L is updated. This improvement saves up to 50% of the runtime as we can see in Section 7.

Algorithm 2 Lower Bound Recovery Algorithm

```
Input: Distribution Network with expanded failure marked area, Installed switch positions
 1: for all substations in the network do
 2:     Create interconnected $TestSector$ starting from substation limited only by normally open switches.
 3:     $L \leftarrow$ frontier feeders (adjacent open switches in normal operation)
 4:     if $TestSector$ is not feasible then
 5:         Create interconnected $TestSector$ starting from substation limited by any kind of switches.
 6:         $L \leftarrow$ frontier feeders (all adjacent switches)
 7:     end if
 8:     if $TestSector$ is feasible then
 9:         mark consumer nodes in the $TestSector$ as $attended$
10:         while $L \neq \emptyset$ do
11:             $feeder \leftarrow L$.pop()
12:             Close the switch in $feeder$
13:             Expand the $TestSector$ limited by switches
14:             if $TestSector$ is feasible then
15:                 mark consumer nodes in the $TestSector$ as $attended$
16:                 Add new frontier feeders in list $L$
17:             end if
18:         end while
19:     end if
20: end for
21: return Distribution Network with recovered consumer nodes marked as $attended$
```

4 A GRASP for the switch allocation problem

According to Resende and Ribeiro (2003), greedy randomized adaptive search procedure (GRASP) is an iterative process, where each GRASP iteration consists of a semi-greedy construction phase and a local search phase. The construction phase builds a feasible solution, whose neighbourhood is explored in the local search phase. The best solution over all GRASP iterations is returned as the result. Input for GRASP include the stop criterion, which might be a fixed time or a maximum number of iterations.

4.1 Construction phase

The construction phase builds a feasible solution one element at a time, as illustrated in Algorithm 3. A greedy algorithm selects the best element each time, whereas a semi-greedy algorithm selects one element at a time from a restricted candidate list. This restricted candidate list keeps a set of the best elements and one of them is picked randomly. In this case, the candidate list is built by ordering all possible switch locations according to the reliability improvement of installing each switch. Then a portion of α switches with the highest reliability are kept. A value of $\alpha = 0$ is equivalent to a greedy algorithm and selects always the best element, and $\alpha = 1$ is equivalent to a random construction. Finally the selected switch is added to the solution.

Algorithm 3 Semi-greedy Constructive Algorithm

```
Input: $SwitchNumber$, $\alpha$ randomness
$Solution \leftarrow \emptyset$
while $SwitchNumber$ is not attained do
    $CandidateList \leftarrow$ feasible switch locations
    $RestrictedCandidateList \leftarrow$ best $\alpha$ switch locations
    $s \leftarrow$ select a switch from $RestrictedCandidateList$
    $Solution \leftarrow Solution \cup s$
end while
return $Solution$
```

4.2 Local search phase

The solutions generated by a GRASP construction phase are not guaranteed to be locally optimal. Hence, GRASP improves each built solution with a local search. The local search explores the neighbourhood proposed by Costa et al. (2007). This neighbourhood is defined by the reallocation of one switch position in the current solution to a new feeder.

The local search was implemented in two ways: best improvement and first improvement. Algorithm 4 depicts a pseudocode of the first improvement local search from our implementation. Those algorithms receive as parameters the initial solution created by the semi-greedy constructive algorithm and a stop criterion. If it finds a better solution, it becomes the current solution. It stops when there are no better solutions in the neighbourhood. The best improvement searches through all the neighbourhood to select the best new solution, while the first improvement stops when it finds any better solution (line 10). Finally the best found solution is returned.

Algorithm 4 First Improvement Local Search Algorithm

```
Input: $StopCriteria$, $InitialSolution$
 1: Evaluate $InitialSolution$
 2: $BestSolution \leftarrow CurrentSolution \leftarrow InitialSolution$
 3: while $StopCriteria$ is not satisfied do
 4:     for all feeders $f_a$ without switch do
 5:         for all feeders $f_b$ with switch do
 6:             if can reallocate a switch from $f_b$ to $f_a$ then
 7:                 $NewSolution \leftarrow$ Move the switch in $CurrentSolution$
 8:                 Evaluate the $NewSolution$
 9:                 if $NewSolution > BestSolution$ then
10:                     $BestSolution \leftarrow NewSolution$
11:                     exit for
12:                 end if
13:                 Restore $CurrentSolution$
14:             end if
15:         end for
16:     end for
17:     $CurrentSolution \leftarrow BestSolution$
18: end while
19: return $BestSolution$
```

5 A tabu search algorithm for the switch allocation problem

tabu search is a metaheuristic proposed by Glover (1989, 1990). In Costa et al. (2007), they tested a tabu search and a greedy construction algorithm. The tabu search used a best improvement neighbourhood search. We extended their implementation with a first improvement neighbourhood search and the semi-greedy constructive algorithm we used for GRASP.

Algorithm 5 shows our implementation of a first improvement tabu search for this problem. The algorithm starts with the semi-greedy constructive phase, and then the solution is improved by a local search. The stop criterion might be a fixed number of iterations or a number of iterations without improvement. The neighbourhood search is the same of GRASP, moving one switch between two feeders. The algorithm keeps track of the best neighbourhood solution found in the current iteration to use it as start in the next one (line 10). The neighbourhood search stops if the improvement is better than the current solution (lines $11 - 13$). If no better solution is found among the neighbours that are not in the tabu list (restricted neighbourhood), the best solution among the restricted neighbours becomes the new current solution. The algorithm continues to search for better solutions in the restricted neighbourhood of the new current solution. After the neighbourhood search, the best and the current solutions are updated. Everytime the method moves to another solution, we mark the moved switch as tabu for a given number of iterations. Finally, the best overall solution is returned by the tabu search.

Algorithm 5 First Improvement Tabu Search Algorithm

```
Input: $StopCriteria$
 1: Create and evaluate $InitialSolution$
 2: $BestSolution \leftarrow CurrentSolution \leftarrow InitialSolution$
 3: while $StopCriteria$ is not satisfied do
 4:     Clear $BestNeighbourSolution$ for neighbours search
 5:     for all feeders $f_a$ without switch do
 6:         for all feeders $f_b$ with switch do
 7:             if they are not in tabu list and can reallocate a switch from $f_b$ to $f_a$ then
 8:                 Move the switch in $CurrentSolution$ to form a New Solution
 9:                 Evaluate the $NewSolution$
10:                 if $NewSolution > BestNeighbourSolution$ then
11:                     $BestNeighbourSolution \leftarrow NewSolution$
12:                     if $BestNeighbourSolution > CurrentSolution$ then
13:                         exit for
14:                     end if
15:                 end if
16:                 Restore $CurrentSolution$
17:             end if
18:         end for
19:     end for
20:     if $BestNeighbourSolution > BestSolution$ then
21:         $BestSolution \leftarrow BestNeighbourSolution$
22:     end if
23:     $CurrentSolution \leftarrow BestNeighbourSolution$
24:     Put the old switch location in the tabu list.
25: end while
26: return $BestSolution$
```

6 Generator of synthetic electrical distribution networks

A problem faced by researchers who work with electrical systems is the small amount of test cases available. In most cases researchers have to create an artificial electrical network and show it to a specialist that validates the instance. Despite this approach is reliable, since an experienced specialist can design real world instances, it has also drawbacks. First, it needs the work of a specialist, who may not be always available. Second, it is laborious, since the specialist has to build each electrical network manually. Because of this, it is hard to create many instances, which is undesirable since it is important to test reliability algorithms with a considerable amount of instances. These reasons lead us to design a simple electrical distribution systems generator.

Our aim is to design output instances similar to real world electrical distribution systems. In other words, we want a generator that creates good abstraction models of real electrical distribution networks.

The designed generator expects as input a network topology representing a primary distribution system. Analogous to Figure 1, each vertex in this graph represents a secondary distribution network (consumer white nodes) or a substation (black nodes) and each edge corresponds to a feeder. Each instance has only one substation. The generator will choose the same vertex described in the graph as the substation. The substation is the node that feeds the electrical network. Since electrical distribution systems are radial (with some redundant open feeders in order to increase reliability), the generator will also open and close the edges (feeders) using a breath-first search, which leads to a tree rooted at the substation.

6.1 Vertex and edges properties

Tension of operation is an important issue when dealing with electrical primary distribution systems. According to Table 1.1 of Pransini (2005), primary distribution systems have voltages between 2400 V and 34500 V. Because of that, we randomly select an integer number between 2400 and 34500 to set the tension of operation of the generated electrical network.

We also have to choose the loads of each secondary distribution system. The load is the amount of power a consumer vertex requires from the substation vertex. Since information about typical brazilian consumer loads were not available, we used information from India. According to Pabla (2004), a domestic consumer in India has a load of 0.85 KW and a commercial consumer has a load of 1.34 KW. Small industries have a load of 11.16 KW. Secondary distribution systems consist of a small group of consumers, e.g. Souza et al. (2006) use a real example of domestic secondary distribution system with 62 consumers. Because of that, in our generator, we attribute loads between 50 and 250 KW to each secondary distribution system and consequently to each vertex in our generated primary distribution system.

It is important to notice that substations have a capacity instead of a load. In order to guarantee that enough energy will be available, in our generator of electrical networks, the capacity assigned to substation is the sum of all loads multiplied by a constant value bigger than 1. We used a constant equal to 1.3 in the generated datasets. Thus the substation is capable of support all the consumer loads and it will also have 30% of extra energy.

Another property of our vertices is the power factor. Power factor is the ratio between the true power and the apparent power. It is a number between 0 and 1. Inductance is the element in the circuit which is pulling the power factor below 1. Most electrical equipments contain inductance and capacitance in some degree, thus it is common to have small values below 1. More explanation about power factor can be found in Pransini (2005). In Pabla (2004) are listed some common power factors for various kinds of electrical equipments. The values range from 0.3 to 0.9. Since we suppose that industries will use capacitors, which help to increase low power factors, we attribute values between 0.6 and 1.0 in the nodes of the generated electrical network.

In order to obtain a network similar to a real one, we have to choose some properties for the feeders too. For example, each cable has a capacity. Our generator use always the same type of cable, with capacity of 4000 ampere. This value was chosen because it is enough for most of the cases.

6.2 Generated Instances

To evaluate the usefulness of the generator, we used twelve graphs as input. We obtained the electrical distribution systems summarized in Table 1. These topologies are originally proposed by Fortz and Thorup (2004) and are from telecommunication problems. They are divided in three classes. The first four networks named "hier" are 2-level hierarchical communication networks generated using the generator discussed in Zegura (2005). The four networks named "rand" are random networks. In random networks the probability of having an arc between two nodes is given by a parameter that controls the density of the network. Finally, the least four networks are Waxman graphs. In this class of graphs, the probability of having and edge between two nodes is proportional to their euclidean distance. Hence, nodes close to each other have more probability of having an edge connecting them than nodes far away from each other (Waxman, 1988). We added a prefix ”e_” to remark that these are electrical instances. These instances are available under request.

Table 1. Synthetic Instances

	Nodes	Edges	Capacity	Load (KW)			Fpot.			Tension
			(MVA)	Min.	Max.	Avrg.	Min.	Max.	Avrg.	(V)
e_hier100	100	141	23.91	50	248	148.14	0.61	0.987	0.81	6964
e_hier100a	100	181	25.03	157	249	154.14	0.60	0.999	0.81	22802
e_hier50	50	75	11.31	50	248	142.22	0.60	0.999	0.81	15062
e_hier50a	50	107	13.19	52	249	161.86	0.61	0.998	0.82	27595
e_rand100	100	394	25.19	50	249	150.40	0.61	0.998	0.79	31039
e_rand100b	100	485	25.47	52	249	156.17	0.60	0.996	0.81	30335
e_rand50	50	219	11.87	57	247	148.12	0.61	0.999	0.81	6045
e_rand50a	50	235	12.26	55	244	148.94	0.60	0.994	0.78	20133
e_wax100	100	381	23.52	54	249	141.63	0.60	0.986	0.79	5966
e_wax100a	100	463	25.78	55	247	155.34	0.60	0.999	0.79	3646
e_wax50	50	163	11.95	60	247	146.71	0.60	0.999	0.81	24805
e_wax50a	50	221	11.18	50	247	137.78	0.61	0.988	0.79	21505

7 Experimental Results

For our tests we used two groups of problem instances. The first group is composed of four instances used by Costa et al. (2007, 2008). Table 2 gives details of those instances, such as the amount of nodes, lines, and the demand type. The demand type can be *uniform* if every consumer has the same demand or *random* if each consumer has different demand. The second group is a set of 12 instances, made with the generator proposed in Section 6.

Table 2. First Group Problem Instances

Instance	Number of nodes		Number of lines		Demand Type
	substations	consumers	radial	redundant	
1U	1	33	33	7	uniform
1R	1	33	33	7	random
2U	5	88	92	17	uniform
2R	5	88	92	17	random

First we tested the optimistic modification of lower bound algorithm proposed by Costa et al. (2008). As explained in Section 3.2 the optimistic approach expands a feasible area that we assume is working in a regular operation state. This saves many unnecessary electrical feasibility tests. We obtained the same results, but it saved about $40\% - 50\%$ of the runtime as it can be observed in Table 3. We used the optimistic evaluation for the remaining tests.

Table 3. Tabu search execution time comparison (in seconds)

Instance	Switches	Costa et al. (2008)	With Optimistic Recovery	Speedup Factor
1R	10	16.52	8.43	51.0
1U	10	18.05	8.82	48.8
2R	10	563.70	229.07	40.6
2U	10	576.61	231.29	40.1

We run GRASP and tabu search algorithms using best and first improvement local search. For these four combinations, we tested five randomness α values: 0.0 (totally greedy), 0.25, 0.50, 0.75 and 1.0 (totally random). For our tests we used the following parameters: The GRASP stop criterion is 10 iterations. The GRASP local search runs until no better solution is found in the neighbourhood. The tabu tenure corresponds to 10 iterations. The tabu stop criteria is a fixed number of 100 iterations. In both algorithms the objective function is our lower bound reliability evaluation (with electrical restrictions). We use the upper bound reliability (only connectivity) found with a best improvement tabu search to compare the quality of solutions.

Table 4. Experiment Results for instances from Costa et al. (2007)

Problem	Switches	α value	Conn.	GRASP				Tabu search			
			Upper	Best-Improvement		First-Improvement		Best-Improvement		First-Improvement	
			Bound	Best	Time(s)	Best	Time(s)	Best	Time(s)	Best	Time(s)
1R	5	0.50	65.851	**54.725**	1.95	**54.725**	1.81	**54.725**	2.32	**54.725**	1.79
		0.25	65.851	**54.725**	1.86	**54.725**	1.83	**54.725**	2.42	**54.725**	2.00
	10	0.50	78.425	70.892	8.53	70.639	6.26	**71.043**	8.60	70.672	5.62
		0.25	78.425	70.790	6.93	**71.043**	6.33	**71.043**	8.46	**71.043**	5.39
	15	0.50	82.226	**77.728**	19.76	**77.728**	12.79	**77.728**	15.21	76.891	12.54
		0.25	82.226	**77.728**	16.99	**77.728**	9.26	**77.728**	15.44	**77.728**	10.53
	20	0.50	83.849	**79.420**	29.11	**79.420**	17.61	**79.420**	19.97	**79.420**	14.22
		0.25	83.849	**79.420**	23.19	**79.420**	14.72	**79.420**	19.64	**79.420**	13.29
1U	5	0.50	64.205	**51.042**	1.63	49.148	1.04	**51.042**	2.31	**51.042**	1.77
		0.25	64.205	**51.042**	1.47	49.148	1.21	**51.042**	2.39	**51.042**	1.65
	10	0.50	77.746	**68.277**	7.95	**68.277**	5.75	66.572	8.17	68.182	5.60
		0.25	77.746	67.992	6.86	66.572	4.23	66.572	8.36	**68.277**	6.47
	15	0.50	81.629	**74.716**	16.06	**74.716**	12.88	**74.716**	18.19	**74.716**	12.22
		0.25	81.629	**74.716**	16.17	**74.716**	12.85	74.432	18.99	**74.716**	12.65
	20	0.50	83.144	**77.746**	30.84	77.652	22.95	**77.746**	21.50	77.652	15.67
		0.25	83.144	**77.746**	27.20	77.652	18.40	**77.746**	22.77	**77.746**	13.68
2R	5	0.50	84.979	**80.934**	62.64	80.396	82.32	80.396	109.88	80.396	79.12
		0.25	84.979	**80.934**	62.30	80.396	75.27	80.396	113.99	80.396	88.00
	10	0.50	87.429	81.562	211.84	81.562	190.77	**83.596**	278.87	81.562	141.67
		0.25	87.429	80.943	208.42	80.943	163.23	80.943	224.13	80.939	142.89
	15	0.50	88.971	81.769	491.82	**84.779**	329.33	81.656	371.54	81.769	196.33
		0.25	88.971	84.186	407.99	84.186	234.00	81.656	373.85	84.186	302.93
	20	0.50	89.838	**85.099**	829.50	84.473	519.22	84.473	701.39	81.733	335.36
		0.25	89.838	84.473	785.71	84.473	371.01	84.473	701.15	82.402	322.84
2U	5	0.50	84.835	80.948	64.74	**81.868**	65.07	80.179	110.21	80.179	85.36
		0.25	84.835	**81.868**	57.85	**81.868**	57.18	80.179	115.45	80.151	83.62
	10	0.50	87.390	82.376	232.64	80.852	162.58	80.591	235.23	**83.462**	180.89
		0.25	87.390	**83.462**	218.16	**83.462**	142.49	80.591	235.14	**83.462**	178.74
	15	0.50	88.915	83.214	461.79	83.929	270.41	81.278	369.81	**85.014**	248.02
		0.25	88.915	83.929	418.46	83.929	259.70	81.278	373.51	84.299	319.05
	20	0.50	89.753	**85.508**	841.35	84.670	371.49	84.684	741.98	81.484	374.97
		0.25	89.753	84.670	660.67	**85.508**	413.42	82.349	583.99	84.684	374.82

Table 4 show results with $\alpha = 0.25$ and $\alpha = 0.5$ for the first group of instances. Table 5 shows results for some synthetic instances. The best found solutions have good quality as shown by the difference with an upper bound of less than 5% for 2R and 2U and less than 12% for 1R and 1U problems. The larger difference in problems 1R and 1U can be explained by a more concentrated demand in less consumers. The best results of both heuristics are always found for an α value of 0.25 or 0.5, even when the results difference to other α values is not greater than 3%. This indicates that the best α is between 0.25-0.5. The runtime of GRASP with $\alpha = 0.25$ is in average about 10% less than $\alpha = 0.5$. The runtimes of GRASP with an $\alpha = 1$ are the highest because the start solution is random and the local search takes more time. In our experiments, the tabu search with a semi-greedy initial solution finds slightly better final solutions on average, when comparing with completely greedy or random initial solutions. Even while the initial solution improves with higher greedyness, a totally greedy initial solution does not lead to a better solution compared with semi-greedy. This indicates that semi-greedy multistart heuristics have more chances of finding better solutions. The first improvement and the best improvement variants of both metaheuristics find similar results, the larger difference with the best found solution is 6% and the average is 1.1%, but first improvement uses less runtime since the explored neighbourhood is smaller. GRASP generally finds equal or slightly better results than tabu search, i.e. with $\alpha = 0.25$ and $\alpha = 0.5$, GRASP has an average difference of 0.55% with the best found solution and tabu search has 0.96%.

Table 5. Experiment Results for the Synthetic Instances

Problem	Switches	α value	Conn.	GRASP				Tabu search			
			Upper	Best-Improvement		First-Improvement		Best-Improvement		First-Improvement	
			Bound	Best	Time(s)	Best	Time(s)	Best	Time(s)	Best	Time(s)
e_hier50a	5	0.25	82.335	**82.335**	15.50	**82.335**	9.94	**82.335**	25.70	**82.335**	21.05
	10	0.25	87.713	**87.713**	51.13	**87.713**	37.15	**87.713**	68.72	87.642	54.68
	15	0.25	90.414	**90.414**	97.80	**90.414**	67.04	**90.414**	118.20	90.405	87.54
	20	0.25	91.615	**91.615**	170.76	**91.615**	73.20	91.548	169.69	91.548	106.38
e_rand50a	5	0.25	88.357	88.157	42.44	88.157	24.45	**88.357**	75.93	**88.357**	70.18
	10	0.25	90.294	90.069	139.20	89.637	66.49	**90.294**	184.27	90.241	144.44
	15	0.25	91.718	91.097	263.37	90.626	121.62	**91.718**	318.97	91.451	222.91
	20	0.25	92.726	91.251	384.03	91.335	187.57	**92.726**	505.96	92.353	348.61
e_wax50a	5	0.25	87.415	87.354	40.59	**87.415**	25.98	**87.415**	68.00	**87.415**	62.12
	10	0.25	90.746	90.200	103.33	**90.541**	58.16	90.392	169.84	90.163	139.74
	15	0.25	92.373	**91.962**	250.94	91.888	131.39	91.887	258.38	91.826	214.71
	20	0.25	93.172	92.902	412.72	92.699	249.98	92.859	406.98	**93.084**	288.42

8 Concluding Remarks

In this paper we studied the switch allocation problem, with the switch reconfiguration problem as a subproblem. The objective is to improve network reliability by decreasing the unattended demand in case of feeder failures. We presented a GRASP metaheuristic for the switch allocation problem and compared it with a tabu search. We also presented an improved evaluation of the electrical constraints with an optimistic heuristic. We further introduced a new set of synthetic instances of problems. A comparison of both metaheuristics on these synthetic instances and other instances from the literature Costa et al. (2007, 2008) shows that the GRASP finds slightly better results than tabu search. In both metaheuristics, the first improvement strategy is able to find results of good quality in less time.

Acknowledgements

This work is supported by the project "Metodologia Multiobjetivo para Análise e Melhoria da Confiabilidade de Sistemas de Distribuição de Energia Elétrica" (Auxílio Integrado CNPq, CTEnerg processo 554900/2006-8).

References

Billinton, R. and Jonnavithula, S. (1996). Optimal switching device placement in radial distribution systems, *Power Delivery, IEEE Transactions on* **11**(3): 1646–1651.

Carreno, E. M., Moreira, N. and Romero, R. (2007). Distribution network reconfiguration using an efficient evolutionary algorithm, *Power Engineering Society General Meeting, 2007. IEEE* pp. 1–6.

Carvalho, P., Ferreira, L. and da Silva, A. (2005). A decomposition approach to optimal remote controlled switch allocation in distribution systems, *Power Delivery, IEEE Transactions on* **20**(2): 1031–1036.

Costa, A. M., Ritt, M., Buriol, L. S. and França, P. M. (2007). Alocação de chaves para melhoria da confiabilidade da rede de distribuição de energia elétrica, *XXXIX Simpósio Brasileiro de Pesquisa Operacional*, Fortaleza – Brazil.

Costa, A. M., Ritt, M., Buriol, L. S., Garcia, V. J. and França, P. M. (2008). Melhoria da confiabilidade da rede de distribuição de energia elétrica, *XL Simpósio Brasileiro de Pesquisa Operacional*, João Pessoa – Brazil.

da Silva, L. G. W., Pereira, R. A. F., Abbad, J. R. and Mantovani, J. R. S. (2008). Optimised placement of control and protective devices in electric distribution systems through reactive tabu search algorithm, *Electric Power Systems Research* **78**(3): 372 – 381.

da Silva, L. G. W., Pereira, R. A. F. and Mantovani, J. R. S. (2004). Optimized allocation of sectionalizing switches and control and protection devices for reliability indices improvement in distribution systems, *Transmission and Distribution Conference and Exposition: Latin America, 2004 IEEE/PES* pp. 51–56.

Delbem, A., de Carvalho, A. and Bretas, N. (2005). Main chain representation for evolutionary algorithms applied to distribution system reconfiguration, *Power Systems, IEEE Transactions on* **20**(1): 425–436.

Fortz, B. and Thorup, M. (2004). Increasing internet capacity using local search, *Computational Optimization and Applications* **29**: 13–48.

Glover, F. (1989). Tabu Search–Part I, *INFORMS JOURNAL ON COMPUTING* **1**(3): 190–206.

Glover, F. (1990). Tabu Search–Part II, *INFORMS JOURNAL ON COMPUTING* **2**(1): 4–32.

Jeon, Y.-J., Kim, J.-C., Kim, J.-O., Shin, J.-R. and Lee, K. (2002). An efficient simulated annealing algorithm for network reconfiguration in large-scale distribution systems, *Power Delivery, IEEE Transactions on* **17**(4): 1070–1078.

Khoa, T. and Binh, P. (2006). A hybrid ant colony search based reconfiguration of distribution network for loss reduction, pp. 1–7.

Moradi, A. and Fotuhi-Firuzabad, M. (2008). Optimal switch placement in distribution systems using trinary particle swarm optimization algorithm, *Power Delivery, IEEE Transactions on* **23**(1): 271–279.

Pabla, A. S. (2004). *Electric Power Distribution*, McGraw Hill Professional.

Pransini, A. J. (2005). *Guide to Electrical Power Distribution Systems*, The Fairmont Press.

Resende, M. G. C. and Ribeiro, C. C. (2003). Greedy randomized adaptive search procedures, *in* F. Glover and G. Kochenberger (eds), *Handbook of Metaheuristics*, Kluwer Academic Publishers, pp. 219–249.

Santander, L. G., Chacra, F. A., Opazo, H. and Lopez, E. (2005). Minimal loss reconfiguration based on simulated annealing meta-heuristic, *CONIELECOMP '05: Proceedings of the 15th International Conference on Electronics, Communications and Computers*, IEEE Computer Society, Washington, DC, USA, pp. 95–99.

Souza, A. et al. (2006). Sistema de avaliação de rede secundária de distribuição utilizando algoritmos genéticos, *Espaço Energia* (5): 34–41.

Su, C.-T., Chang, C.-F. and Chiou, J.-P. (2005). Distribution network reconfiguration for loss reduction by ant colony search algorithm, *Electric Power Systems Research Volume 75,*, pp. 190–199.

Teng, J.-H. and Liu, Y.-H. (2003). A novel ACS-based optimum switch relocation method, *Power Systems, IEEE Transactions on* **18**(1): 113–120.

Thakur, T. and Jaswanti (2006). Study and characterization of power distribution network reconfiguration, *Transmission & Distribution Conference and Exposition: Latin America, 2006. TDC '06. IEEE/PES* pp. 1–6.

Wang, C. and Cheng, H. Z. (2008). Optimization of network configuration in large distribution systems using plant growth simulation algorithm, *Power Systems, IEEE Transactions on*, pp. 119–126.

Wang, C., Cheng, H. Z. and Yao, L. Z. (2008). Reactive power optimization by plant growth simulation algorithm, *Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008. Third International Conference on*, pp. 771–774.

Waxman, B. M. (1988). Routing of multipoint connections, *Selected Areas in Communications, IEEE Journal on* **6**(9): 1617–1622.

Wu, W.-C., Tsai, M.-S. and Hsu, F.-Y. (2007). A new binary coding particle swarm optimization for feeder reconfiguration, *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007.*, pp. 1–6.

Zegura, E. (2005). *GT-ITM: Georgia Tech Internetwork Topology Models (software)* .

Zhang, C., Zhang, J. and Gu, X. (2007). The application of hybrid genetic particle swarm optimization algorithm in the distribution network reconfigurations multi-objective optimization, *Natural Computation, 2007. ICNC 2007. Third International Conference on*, pp. 455–459.

Zhang, D., Fu, Z. and Zhang, L. (2007). An improved ts algorithm for loss-minimum reconfiguration in large-scale distribution systems, *Electric Power Systems Research* **77**(5-6): 685 – 694.

Zhang, D., Fu, Z., Zhang, L. and Song, Z. (2005). Network reconfiguration in distribution systems using a modified ts algorithm, *MMACTE'05: Proceedings of the 7th WSEAS International Conference on Mathematical Methods and Computational Techniques In Electrical Engineering*, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, pp. 310–314.