UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ALEXANDER JAVIER BENAVIDES ROJAS

Service Restoration and Switch Allocation in Power Distribution Networks: Bounds and Algorithms

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dr. Luciana S. Buriol
Advisor

Prof. Dr. rer. nat. Marcus Ritt
Coadvisor

Porto Alegre, May 2010

CIP – CATALOGING-IN-PUBLICATION

Benavides Rojas, Alexander Javier

Service Restoration and Switch Allocation in Power Distribution Networks: Bounds and Algorithms / Alexander Javier Benavides Rojas. – Porto Alegre: PPGC da UFRGS, 2010.

111 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2010. Advisor: Luciana S. Buriol; Coadvisor: Marcus Ritt.

1. Heuristics. 2. Switch allocation. 3. Electric power distribution networks. I. Buriol, Luciana S.. II. Ritt, Marcus. III. Service Restoration and Switch Allocation in Power Distribution Networks: Bounds and Algorithms.

DEDICATION

To my parents, Carmen and Oscar, and to my great-grandfather, Manuel, who taught me the meaning of perseverance and hard work.

ACKNOWLEDGMENTS

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).

I want to use this lines to thank and acknowledge all the people that made this dissertation possible.

First and foremost, I would like to thank my advisors Luciana Buriol and Marcus Ritt for their patience, guidance and support over the past two years. I am greatly thankful for the opportunity they gave me when they took me as a student, the encouragement they gave me to feel the work I was doing was worthwhile, and their strong and active interest during the research. I would not have made it this far without them.

I also thank to the research team members. Specially to Prof. Paulo Morelato França, who provided financial support through the previously mentioned CNPq project; and to Mariane Machado, who worked side by side with me.

I also owe much to innumerable friends, professors and students at UFRGS and at the Salesians Congregation in Porto Alegre. I specially thank to Fr. Marcio Lacoski, Daniel Siwik, Daniel Welfer, Diego Pereira, and Mario Machado, for their unselfish friendship and support.

Finally, I wish to express my love and gratitude to my parents, Oscar and Carmen, my brother, Eduardo, my sister, Boni, and my girlfriend, Ana Livia, for their love, support, and encouragement.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS

ABSTRACT

The improvement of reliability in electrical power distribution networks is an important issue for electricity supply industries, due to strict regulations in many countries.

After a failure in the network, some switches are used to isolate the failure, while others restore the energy to some consumers. The optimal selection of the switches to open or close to restore energy is called the service restoration problem. The installation of switches in strategic places may reduce the outage time in case of blackouts, and thus improve the reliability of the network. The optimal selection of places to install switches is called the switch allocation problem. These two problems are closely related.

This dissertation studies the switch allocation problem, considering the service restoration problem as a sub-problem. Two methods are proposed to estimate the reliability of a distribution network with a given set of installed switches. The main focus is in heuristics to solve the joint problem. It proposes methods like tabu search, greedy randomized adaptive search procedure, and iterated sample construction with path relinking. It also studies the benefit of greedy, semi-greedy, random, and sample construction methods, and studies the performance of sample, first improvement and best improvement local search strategies. The different methods are compared and analyzed. The results show that sample approaches are inexpensive and lead to solutions of good quality. Iterated sample construction with path relinking is the best method to solve the joint problem that is proposed in this dissertation.

Keywords: Heuristics, switch allocation, electric power distribution networks.

RESUMO

A melhora da confiabilidade em redes de distribuição de energia elétrica é um tema importante para as indústrias de fornecimento de eletricidade, devido aos regulamentos estritos em muitos países.

Depois de uma falha na rede, algumas chaves são usadas para isolar a falha, enquanto outras restauram a energia a alguns consumidores. A ótima seleção das chaves que serão abertas ou fechadas para restaurar a energia é conhecido como o problema de restauração de serviço. A instalação de chaves em posições estratégicas pode reduzir o tempo de parada, e assim melhorar a confiabilidade da rede. A seleção ótima de posições para instalar chaves é conhecido como o problema de alocação de chaves. Estes dois problemas estão relacionados estreitamente.

Esta dissertação estuda o problema de alocação de chaves, considerando o problema de restauração de serviço como um subproblema. Dois métodos são propostos para estimar a confiabilidade de uma rede de distribuição com um conjunto dado de chaves instaladas. O foco principal está nas heurísticas para resolver o problema composto. Propõe-se aqui métodos como busca tabu, procedimento de busca gulosa adaptativa aleatória (sigla em inglês: GRASP), e procedimento iterativo de construção por amostras com reconexão de caminhos. Também estuda-se o benefício dos métodos de construção gulosa, semigulosa, aleatória e por amostras, e estuda-se o desempenho das estratégias de busca local por amostras, primeira melhoria e melhor melhoria. Os diferentes métodos são comparados e analisados. Os resultados mostram que os métodos por amostras são baratos e levam a soluções de boa qualidade. O procedimento iterativo de construção por amostras com reconexão de caminhos é o melhor método proposto para resolver o problema composto que é proposto nesta dissertação.

Palavras-chave: Heurísticas, alocação de chaves, redes de distribuição de energia elétrica.

1 INTRODUCTION

The purpose of electrical power distribution networks is to deliver the energy to the consumers with an adequate quality. The capacity to deliver energy adequately is called reliability of the network. Depending on the country, the government regulations establish limits and goals to improve periodically the reliability of the distribution networks. Besides, consumer demand increases, requiring the expansion of networks without jeopardizing its reliability. The improvement of the reliability is an important issue for electric distribution enterprises, because they can be penalized if the reliability falls out of reference limits.

Many measures are used to estimate the reliability of a distribution network. In this research, the focus of reliability measures is the impact of failures in elements of the distribution system over the consumers. When a failure occurs, there is a blackout in a part of the distribution network. Sectionalizing devices such as switches and fuses are used to isolate failures and reconnect areas without energy. In this way, the time and number of power blackouts is reduced and the reliability is increased.

The consumers that are affected will depend on the number and location of the sectionalizing devices in the network. During the planning process for improving or expanding an actual distribution network, a combinatorial optimization problem emerges. This problem is the selection of the optimal locations for installing sectionalizing devices in the network.

1.1 Research objective and contribution

1.1.1 Objective of this research

This research focuses on heuristic methods to approximate the optimal locations for installing automatic switches in an electrical power distribution network and thus improve its reliability.

1.1.2 Contribution of this research

This dissertation presents three major contributions. The first contribution is a new method to estimate upper and lower bounds for the reliability of an electrical power distribution network. The second contribution is a library of test instances for the switch allocation problem. This library standardizes instances found in literature and adapts new instances from telecommunication networks to be used with different reliability measures. The third contribution is the proposal and comparison of heuristics to solve the switch allocation problem. The proposed algorithms are greedy randomized adaptive search procedure (GRASP), sample construction, sample local search, and iterated sample construction with path relinking.

1.2 Research context

This dissertation outlines research efforts concerning switch allocation in distribution networks to improve their reliability. Partial results for this research were published in two papers (BENAVIDES et al., 2009b, 2009a), but the results presented in the papers and in the dissertation are different. This section explains the timeline progress of the research, describing its contributions in more detail, and pointing out the differences of results presented in this dissertation and in previous publications.

This study continues previous research of Costa et al. (2007, 2008). Costa et al. (2007) proposed a method to approximate the network reliability, ignoring the electrical constraints. They use this method with two algorithms, a greedy construction algorithm and a tabu search. The algorithms are compared using four instances (described in Section 5.1). They concluded that solutions generated tabu search overcame greedily generated solutions. Costa et al. (2008) proposed another method to approximate the network reliability, but now considering the electrical constraints and the service restoration. They proposed the use of these two methods to estimate upper and lower bounds for the reliability. They only used the greedy algorithm over the reliability estimation with electrical constraints and calculated the upper and lower bounds for the generated solutions, justifying that the calculations of electrical constraints are too costly for the tabu search. They tested the same four instances to validate the approach.

This is the starting point for the research presented in this dissertation. The first contributions of our research were published by Benavides et al. (2009b). They include the optimistic improvement over the service restoration algorithm (Section 3.4), the adaptation of telecommunication networks to generate test instances for the switch allocation problem (Section 5.3), two local search strategies (first and best improvement in Section 4.2.2) that were tested within two metaheuristics, GRASP (Section 4.2) and tabu search (Section 4.3). This paper compared the performance of the two metaheuristics with the instances used by Costa et al. (2007, 2008) and the new synthetic instances. It concludes that a multi-start strategy gives better results than a strategy to escape local minima.

The above research was mainly based on the four instances of Costa et al. (2007) and a nonstandard reliability measure used by Costa et al. (2007, 2008) and Benavides et al. (2009b). We searched for known instances and a standard measure to compare our methods with other authors. The first attempt was to use expected energy non supplied (EENS) with the RBTS bus 4 instance.

Other contributions of our research were published by Benavides et al. (2009a). They include the use of EENS as a standard reliability measure, the adaptation of the RBTS bus 4 instance for service restoration problem to the switch allocation problem (Section 5.2), sample construction and local search algorithms (Section 4.4), and their comparison with other construction and local search strategies. The paper used an outage time of $r_f = 3h$ to calculate EENS. In this dissertation, the outage time is $r_f = 2h$ for all test cases. This created a difference in the results that reduces the EENS values of the paper in a factor of $1/3$. The paper concludes that a sample construction obtains better results than semi-greedy and random constructions, and that a neighbourhood restricted by a random sample speeds up the local search.

Novel unpublished contributions are the comparison of other reliability measures (Section 3.1), the improvement of reliability estimation by sectors (Section 3.2), the adaptation of other instances from a repository of distribution systems (Section 5.4), and an iterated sample construction with path relinking.

The use of other reliability measures and the new reliability estimation by sectors motivated the reproduction of the experiments from previous publications for the writing of this dissertation. Concurrently, we extend the experiments to the new set of test cases with large instances.

1.3 Scope of the dissertation

There are several issues that could be considered for the switch allocation problem and/or the service restoration problem. We had to limit some of them for different reasons. Here, we list some limitations and their implications.

Many works in literature use costs functions that minimize the investment in devices and the detriment of power losses. But the costs they use in their approaches is not always clear. The determination of power losses costs is affected by the type of consumer, e.g., residential, industrial, government, health care, they have different costs for energy losses. Moreover, costs of sectionalizing devices vary with time and depend on importation costs and taxes from different countries. These reasons difficult the comparison and use of approaches based in cost functions. Our approach supposes that an investor defines a budget and a number of switches to be installed. For these reasons, our research focuses in reliability optimization using estimation measures that are not based in costs, and the proposed optimization methods receive the number of switches as a configurable parameter.

There exist many different sectionalizing devices such as switches, fuses, or reclosers. They differ in costs and time of response after a failure. The model required to simulate the response of devices from different kinds is more complicated and requires specific knowledge of the devices. Therefore, our research focuses on one type of device, automatic switches, and the time of response is considered immediate.

1.4 Overview of the dissertation

The remainder of this dissertation is organized as follows. Chapter 2 introduces the electric power distribution networks and presents a graph-based model for the problem. Chapter 2 also describes the service restoration problem and the switch allocation problem, and explains the motivation for considering them together. Chapter 3 reviews the reliability measures and proposes reliability algorithms to estimate upper and lower bounds. Chapter 4 depicts the proposed heuristics and meta-heuristics for solving the switch allocation problem. In Chapter 5, problem instances are described and adapted to create a set of test cases for our algorithms. Chapter 6 provides the experimental tests and results of this research. Finally, Chapter 7 presents the conclusions of this research and ideas for future study.

2 SWITCH RELATED RELIABILITY PROBLEMS OF ELECTRIC POWER DISTRIBUTION NETWORKS

This chapter describes the two optimization problems related to switches and to reliability maximization in distribution networks, and presents a literature review. With this aim, the main characteristics of electric power distribution networks are introduced and a graph model is presented in Section 2.1. Section 2.2 explains the service restoration problem. Finally, Section 2.3 describes the switch allocation problem, and the motivation for considering the service restoration problem as a subproblem.

2.1 Electric power distribution networks

The purpose of an electric power system is to satisfy the system load requirements, providing an adequate supply of electric energy. Electric energy cannot be stored in large scale, but must be distributed to the consumers in real time. Electric power systems are divided in three subsystems:

Generation System. The plants of this system generate the electric energy. There are different generation methods: hydroelectric, eolic, solar thermal, geothermal, nuclear, etc.

Transmission System. The transmission lines carry energy for long distances. Energy flows from generation plants to distribution substations. They operate with power tensions above $60\ kV$. The lines that work on shorter distances with less power tension are called subtransmission lines.

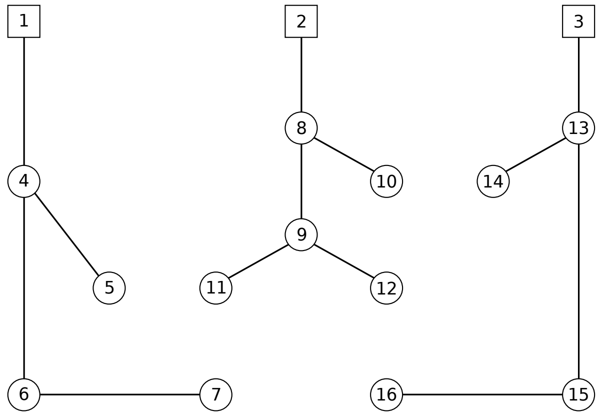
Distribution System. These systems link the substations and the final consumers. They work with power tension between $2.4\ kV$ and $34.5\ kV$.

These three subsystems are interconnected with substations. Substations are facilities that change alternating current voltage between different voltage levels. **Step-up substations** receive power from generation systems and increase the voltage for transmission to distant locations. **Step-down substations** receive high voltage and decrease it for local distribution. **Distribution substations** transform the voltages to consumer voltages. They are usually located in the street posts or underground, close to the consumers.

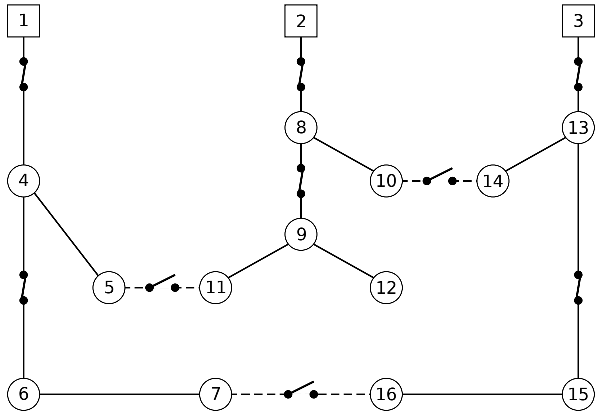
2.1.1 Distribution networks

Fig. 2.1 shows an example of an electric power distribution network taken from Civanlar et al. (1988). The basic circuit of a distribution system (Fig. 2.1a) is a set of non-cyclic circuits that supplies energy to the consumers. It contains **substations** (square nodes) that serve as sources of energy for their local areas of distribution, **consumers** (round nodes) that receive the energy, and **feeder lines** (black lines) that carry the energy between the substation and consumers. In the case of a failure, relays at the substation will open a circuit breaker de-energizing the entire circuit. This leaves consumers without power until repairs are made.

To reduce the outage time (Fig. 2.1b), the circuit may have redundant **loop feeder lines** (dotted lines), and a number of switches. The switches in loop lines are called *normally open* because they are disconnected under normal operating conditions to preserve the radiality of the network. The switches in feeder lines of the basic circuit are called *normally closed* because they close the circuit and bring energy in normal operating conditions. The function of the switches is to modify the network topology in the case of a failure, isolating the failure and restoring



a. Basic circuit.



b. Circuit with loop lines.

Figure 2.1: Example of a distribution network.

the energy supply in parts of the network. This topology modification is called network reconfiguration. An example of a reconfiguration is given in Section 2.2.

2.1.2 Graph-theoretic model

We model an electric distribution network as an undirected graph $G = (N, A)$, where the set of nodes $N = N_S \cup N_C$ represents the set of substations (N_S) and consumer load points (N_C), and the edge set $A = A_{nc} \cup A_{no}$ represents *normally closed* (A_{nc}) and *normally open* (A_{no}) feeder lines.

We write $G' \subseteq G$ to express that G' is a subgraph of G. We also write $V(G') = N$ and $E(G') = A$ for the vertex and edge set of a graph or subgraph $G' \subseteq G$. We define the union of two graphs as the union of their respective node and edge sets, i.e., $G_1 \cup G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$.

The presence of a switch on an edge $a \in A$ is indicated by a boolean value $B_a \in \{0, 1\}$. Observe that a loop line without a switch ($a \in A_{no}$, $B_a = 0$) indicates that the loop line is not operative. Thus, two solution representations can be used for the switch allocation problem. A sequence of boolean values $B = \{B_a\}, a \in A$ that indicates which lines do or do not have a switch installed, or the set of lines themselves with switches $A_B = \{a\}, a \in A, B_a = 1$.

The *sector* $\mathcal{S}(n)$ corresponding to a node $n \in N$ is defined as the largest connected subgraph of G which contains n and is connected only with basic circuit feeder lines that have no switch installed ($a \in A_{nc}, B_a = 0$). For any edge $a = \{u, v\}$ we define the corresponding sector $\mathcal{S}(a) = \mathcal{S}(u) \cup \mathcal{S}(v) \cup (\{u, v\}, \{a\})$ as the union of the sectors of the nodes that it connects.

The *frontier* of a sector $\mathcal{F}(\mathcal{S}(n))$ is the set of edges $a \in A$ which are incident to exactly one node in the sector. Note that frontier edges $a \in \mathcal{F}(\mathcal{S}(n))$ for a node $n \in N$ are redundant lines ($a \in A_{no}$) or lines with switches ($a \in A, B_a = 1$).

For a fault in feeder line $f \in A$, we call the sector $\mathcal{S}(f)$ as *black*. Sectors are called *gray* when they are not connected to any substation in the basic circuit without the black sector. We abbreviate N_f for the set of affected nodes in black and gray sectors where energy can not be restored.

Figure 2.2 shows the example network divided by switches into sectors. The set of nodes $N = \{1, \ldots, 16\}$ has three substations $N_S = \{1, 2, 3\}$. The set of loop lines is $A_{no} = \{\{5, 11\}, \{7, 16\}, \{10, 14\}\}$. Assuming a fault in feeder line $f = \{8, 10\}$, we have the black sector $\mathcal{S}(f)$ with node set $V(\mathcal{S}(f)) = \{8, 10\}$ and edge set $E(\mathcal{S}(f)) = \{\{8, 10\}\}$, limited by the frontier $\mathcal{F}(\mathcal{S}(f)) = \{\{2, 8\}, \{8, 9\}, \{10, 14\}\}$. Note that switches in the frontier must be opened to isolate the failure. The fault also leads to a gray sector $\mathcal{S}(9) = (\{9, 11, 12\}, \{\{9, 11\}, \{9, 12\}\})$. The other sectors shown in white are still energized.

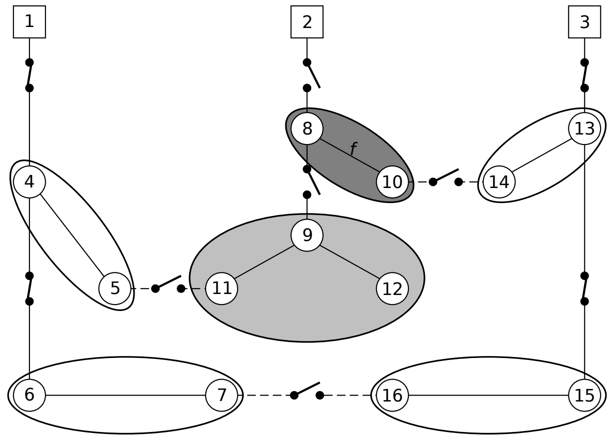


Figure 2.2: Sectored distribution network with a failure.

2.1.3 Electrical characteristics and constraints

Distribution networks present electrical characteristics and constraints that must be respected in normal operation conditions. Next, we list the electrical characteristics of the network components.

- Each line $a \in A$ has a maximum capacity c_a (in Ampere), a resistance R_a and a reactance X_a (in Ohms).
- Each substation $n \in N_S$ has a capacity c_n (in MW).
- Each consumer load point $n \in N_C$ has a power demand D_n, a real load P_n (in W), a reactive load Q_n (in VAR), and a number of consumers M_n. The real and reactive power loads can be calculated from the power demand with a power factor pf and the equations $P_n = pf * D_n$ and ${Q_n}^2 = {D_n}^2 - {P_n}^2$.
- The whole network has a nominal voltage level V (in V) and a maximum voltage drop (as a percentage).

A configuration of switch states is considered electrically feasible if it satisfies electrical operating constraints. The constraints that we consider are:

Radiality of the network. The graph formed by the feeder lines without switches and feeder lines with closed switches has to be a collection of trees, with each tree having one substation as its root.

Capacity constraints. The capacity of the substations and the feeder lines must be respected.

Voltage constraints. The voltage drop at each load point must not exceed the predefined limit.

The method used to test the electrical constraints is explained in Section 3.4.

2.2 The service restoration problem

The process of reconfiguration in electric power distribution systems with a given set of installed switches, consists in opening and closing some switches to obtain a new network topology. The **network reconfiguration problem** consists finding a new feasible topology which optimizes some objective functions. Possible objectives are to reduce the overall system power loss, to balance the load and relieve network overloads or manage load variations of different consumer types, to maximize reliability, to minimize the number of switching operations and thus reduce the reconfiguration costs.

Network reconfiguration has been studied extensively in the literature, and the most common objective function is power loss minimization. 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; ZHANG, 2007), genetic algorithms (DELBEM; CARVALHO; BRETAS, 2005; CARRENO; ROMERO; PADILHA-FELTRIN, 2008), ant colony optimization (SU; CHANG; CHIOU, 2005; KHOA; BINH, 2006), particle swarm optimization (ZHANG; ZHANG; GU, 2007; WU; TSAI; HSU, 2007), and plant growth simulation algorithm (WANG; CHENG, 2008; WANG; CHENG; YAO, 2008). Thakur & Jaswanti (2006) present a detailed survey on power distribution network reconfiguration.

The **service restoration problem** is a special case of the network reconfiguration problem. Given a set of installed switches and a failure, the problem consists in choosing which switches must be opened or closed to maximize the attended area, and thus maximize the network reliability.

In the example of Figure 2.2, when a failure condition is detected, the switches on the fault frontier ($\mathcal{F}(\mathcal{S}(f)) = \{\{2, 8\},\ \{8, 9\},\ \{10, 14\}\}$) must be opened to isolate the fault, and other switches can be closed to restore the energy supply in other sectors (loop line $\{5, 11\}$). The resulting topology is shown in Figure 2.3 where the sector $\mathcal{S}(9) = (\{9, 11, 12\}, \{\{9, 11\}, \{9, 12\}\})$ is energized again.

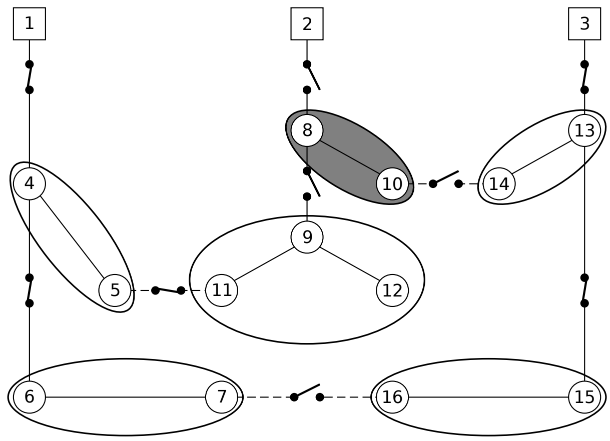


Figure 2.3: Reconfigured distribution network.

The service restoration problem is a subproblem of the switch allocation problem, discussed next.

2.3 The switch allocation problem

Switches play a key role for the reliability of electric power distribution networks (BILLINTON; JONNAVITHULA, 1996a). The number of unattended consumers and the amount of non-supplied energy depend directly on the number and position of the switches in the network (LEVITIN; MAZAL-TOV; ELMAKIS, 1995). Automatic sectionalizing switches are able to diagnose a fault and eventually to automatically reschedule the respective configuration (CELLI; PILO, 1999). The installation of automatic switches in distribution systems allows a better and faster reconfiguration, and hence increases reliability. Electric power distribution networks are large, and installing automatic switches at every line feeder is not possible due to high costs. Therefore, the adequate selection of switch installation locations is important in system planning.

The **switch allocation problem** consists in selecting a set of feeder lines to install i new automatic switches in a distribution network. The objective is to maximize the reliability, i.e., to maximize the attended area in the case of failures, and it is subject to the number of available switches for allocation and to the electrical constraints.

This problem has been studied by several authors with different approaches. Next, we describe some of them.

Levitin, Mazal-Tov & Elmakis (1995) presented a genetic algorithm to determine the location of a user-specified number of switches, and modified the crossover operator to adapt it to the problem. They used a binary representation, and as reliability measure they approximate SAIDI from the annual non-supplied energy. They tested the proposed methods in a network with 96 lines and 52 load points.

Billinton & Jonnavithula (1996a) proposed a simulated annealing approach. They considered the investment, maintenance and outage costs in a single global cost function to determine the best number and location of switches. They used ECOST for the outage costs. For tests, they used two previously proposed test systems known as RBTS bus 4 and RBTS bus 6 (ALLAN et al., 1991; BILLINTON; JONNAVITHULA, 1996b).

Carvalho, Ferreira & Silva (2005) presented a decomposition approach based on divide and conquer. They represented the solution as a binary array of the possible location places for remote controlled switches. They reduced the problem complexity by using a polynomial-time partitioning algorithm to decompose the problem into a set of convex independent subproblems to be solved independently. The objective function was the sum of installation costs and the EENS multiplied by a fixed cost. The algorithms were tested in a real distribution network with eleven possible places for remote switches, showed that reduces the search space from 2^{11} evaluations to 60 evaluations, and found the best solution.

Chen et al. (2006) and Lin et al. (2006) proposed an immune algorithm and compared the results with a genetic algorithm. The objective function was the sum of the consumer interruption cost and the investment cost of installation switches. A sample test system with 19 load points and a real distribution system with 90 load points from Taiwan Power Company were used for tests.

Silva, Pereira & Mantovani (2004a, 2004b) proposed genetic algorithms to allocate switches, reclosers and fuses. The device allocation problem was modeled with non-linear integer programming models. The objective function for the first paper was SAIFI and for the second was a cost estimation for installation and non supplied power. They tested their approach in real circuits of 134 and 20 lines respectively.

Silva et al. (2008) proposed a reactive tabu search to the device allocation problem. The objective function was a sum of estimated costs. Two sets of neighbours had been defined for a given solution. The first set of neighbours contains all the possible solutions where a single device is moved from its current location to any other allowed empty location within the feeder. The second set of neighbours is defined by increasing in one unit, any type of device independently. The tabu list kept the evaluated solutions of a pre-established number of iterations, considering also a maximum number of devices that can change their positions simultaneously. The reactive mechanism increased the size of the tabu list when solutions are repeating, and decreases it when there were no repeated solutions for a number of iterations. Again, presented results were from tests with the 134 lines real circuit.

Moradi & Fotuhi-Firuzabad (2008) treated the problem with a three state Particle Swarm Optimization (PSO). Their ternary representation considered the installation of sectionalizing and circuit breaker switches. The optimization function is ECOST and the test instances were the RBTS bus 4 (ALLAN et al., 1991) and the IEEE 123 node test feeder (KERSTING, 2001).

Villasanti, Baran & Gardel (2008) proposed an evolutionary multiobjective algorithm. Three objective functions were selected, minimization of the number of new installed switches, minimization of unavailability index, and maximization of the number of successful restoration simulations. A restoration is successful if it isolates every load point with failure and restores the energy supply to all other load points. They used a test system from Paraguay called "tres bocas" with 80 load points and 117 lines.

Falaghi, Haghifam & Singh (2009) presented an ant colony optimization algorithm. They used a multiobjective approach with fuzzy membership sets. The objectives were defined as minimization of the cost of sectionalizing switches and reliability improvement measured with EENS. They used two instances for tests, they adapted part of the RBTS bus 6 (BILLINTON; JONNAVITHULA, 1996b) and used a private network of Iran.

For a survey of other solution methods, we refer the reader to (ELMAKIAS, 2008). Many of the described approaches use a simplification to calculate the unattended areas assuming that, for a given set of switches and a failure, the affected nodes are known or easy to compute, estimating the reliability measures with statistical data and assuming that gray sectors can be restored if there exists a loop line. This disregards the underlying service restoration problem with electrical constraints. For example, if there exist a loop line that can restore the energy supply to a gray sector, there still exist the possibility that the substation can not support it or that the voltage drops out of allowed limits.

The described approaches use different objective functions, allocate different types of switches and evaluate on different test instances. Frequently, the instances are not completely specified and their optimum value is not known. This makes impossible to compare directly their results.

Our approach takes into account the service restoration problem as a subproblem. To evaluate a solution for the switch allocation problem, we must solve the embedded service restoration problem for every possible fault, taking into account electrical constraints. In this way, we obtain a better approximation for the network reliability of the switch allocation solution, and we handle larger and more complex network instances, that do not satisfy the assumptions of simple restoration heuristics. The method we propose to estimate the reliability is explained in Chapter 3.

3 RELIABILITY ESTIMATION OF ELECTRIC POWER DISTRIBUTION NETWORKS

Electric power distribution systems deliver energy with a reasonable assurance of continuity and quality. The ability of the system to provide an adequate supply of electric energy is usually designated by the term of reliability. There exist many reliability measures, and the most common are described in Section 3.1. They are used to assess the reliability of real networks and to estimate the reliability during the network planning process.

In this chapter, two methods are proposed to estimate the upper and lower bounds of reliability of an electric power distribution network. The common part of the reliability estimation methods is explained in Section 3.2. The different service restoration methods to calculate the reliability lower and upper bounds are explained in Sections 3.3 and 3.4, respectively. We explain the upper and lower bounds using EENS reliability measure as example. Costa et al. (2007, 2008) and Benavides et al. (2009b) used APSE reliability measure with inverted bounds, because APSE and EENS are inversely correlated (see Section 6.1). Other measures can be used, taking into account the bounds order.

3.1 Distribution network reliability measures

Generally, a device or system is said to perform satisfactorily if it does not fail during service time. Power distribution systems are expected to go under failures, be repaired and then return to service. In this case, more appropriate measures of reliability are indices of energy unavailability.

There are many measures to assess the reliability of a power distribution network. The most common measures in the literature are expected outage cost (ECOST), expected energy non supplied (EENS), system average interruption frequency index (SAIFI) and system average interruption duration index (SAIDI).

Another measure used by Costa et al. (2007, 2008) and Benavides et al. (2009b) is average percentage of supplied energy (APSE). This reliability measure is nonstandard, but it was used in the early phases of our research.

The measures are defined as:

$$\begin{aligned}
\text{ECOST} &= \sum_{f\in A_{nc}} \lambda_f \cdot \sum_{n\in N_f} P_n C_n(r_f) && \text{(US\$)},\\
\text{EENS} &= \sum_{f\in A_{nc}} \lambda_f r_f \sum_{n\in N_f} P_n && \text{(MWh/period)},\\
\text{SAIFI} &= \frac{1}{M} \sum_{f\in A_{nc}} \lambda_f \cdot \sum_{n\in N_f} M_n && \text{(Interruptions/period)},\\
\text{SAIDI} &= \frac{1}{M} \sum_{f\in A_{nc}} \lambda_f \cdot r_f \cdot \sum_{n\in N_f} M_n && \text{(h/period)},\\
\text{APSE} &= \frac{1}{|A_{nc}|} \cdot \frac{1}{P} \sum_{f\in A_{nc}} \sum_{n\in N_C\setminus N_f} P_n && \text{(Percentage)}
\end{aligned}$$

where A_{nc} is the set of feeder lines that can fail, N_f is the set of affected nodes by a failure f and $N \setminus N_f$ is the set of unaffected nodes. r_f is the average outage time (in hours) and λ_f is the average failure rate. P_n is the energy normally consumed by node n and P is the total demand of the network. M_n is the number of affected consumers on node n and M is the total number of consumers. $C_n(r_f)$ is an outage cost function that depends on outage time and consumer type.

Different measures depend on different parameters, and serve for different purposes. For example, SAIFI and SAIDI depend on the number of consumers and are used to measure the consumer satisfaction, ECOST and EENS depend on the non-supplied energy and are used to estimate the lost energy and its respective cost. All these measures consider the failure rate for every line (except APSE) and the related outage time (except SAIFI and APSE). The most simple measure is APSE because it only depends on the consumers demand.

We use four of these measures (EENS, SAIFI, SAIDI and APSE) and compare them in Section 6.1. A detailed explanation of these and other common measures can be found in IEEE Std 1366 (2003) and Goel & Billinton (1991).

3.2 Network reliability: assessment and bounds

This research proposes methods to solve the switch allocation problem taking into account the service restoration problem. The reliability estimation for a given solution to this joint problem is not easy. This means, for a given set of lines with new automatic switches installed, we have to determine the reliability after restoration for each possible failure in the network. Costa et al. (2007, 2008) proposed two approaches to estimate this reliability. They differ in the service restoration algorithm. The function of the restoration algorithm is to determine if a load point n does or does not belong to the set of nodes N_f affected by a failure f. The first restoration algorithm considers only the network connectivity and ignores the electrical constraints mentioned in Section 2.1.3, and therefore obtains a lower bound for the EENS. The second algorithm includes these constraints, obtaining an electrically feasible solution for the service restoration problem, and therefore obtains an upper bound for the EENS. These two service restoration algorithms are explained in the next sections.

The common part of these two approaches is described in Algorithm 3.1. For every failure f it performs three steps. First, it expands and isolates the failure, obtaining the black area and its frontier. Second, it determines the served and non-served load points with one of the service restoration algorithms. Third, it calculates the partial EENS$_f$ of the consumers $n \in N_f$ affected by the failure f. Finally, it sums the total EENS.

An algorithmic improvement proposed in this work is an evaluation by sectors. The sectors defined by a given set of switch positions and open lines do not change during an evaluation. The most expensive process is the failure restoration algorithm. So, we define a set of sectors $\mathcal{SS} = \{\mathcal{S}(n) | n \in N\}$ that contains all the disjoint sectors of nodes $n \in N$. Note that we define the set with sectors of nodes and not of edges, because a sector $\mathcal{S}(a) = \mathcal{S}(u) \cup \mathcal{S}(v) \cup (\{u, v\}, a)$ can be formed by two non-overlapping sectors of nodes if $a = \{u, v\}$ is a frontier edge.

Algorithm 3.2 shows the improved estimation. This time we process sector by sector (lines 2-9), saving computing time. Note that frontier feeder lines (normally closed with switches) must still be processed separately (lines 10-17), because they are not within any sector. First, it simulates a failure in each sector from the sector set $\mathcal{SS}$. The black area is the current sector, so the failure does not need to be expanded and the frontier is known to be isolated. Second, it determines the non-served load points with a service restoration algorithm. Third, it calculates the partial EENS$_f$ of the consumers $n \in N_f$ affected by the failure f, evaluating it for every feeder line $a \in E(\mathcal{S}(f))$ in the black sector at once (line 7).

We follow a similar process than the original method to evaluate the frontier lines, but we determine and isolate the black sector $\mathcal{S}(f)$ easily with help of the defined sectors (lines 12 and 13). Finally, the algorithm returns the total EENS.

We can also describe the estimation by sectors of the EENS as:

$$\text{EENS} = \sum_{\mathcal{S}_f \in \mathcal{SS}} \left(\Big(\sum_{a \in E(\mathcal{S}_f)} \lambda_a r_a \Big) \Big(\sum_{n \in N_f} P_n \Big) \right) + \sum_{f \in A_{nc} \cap A_B} \lambda_f r_f \sum_{n \in N_f} P_n$$

where the first term is the summation of the partial EENS of every sector and the

Algorithm 3.1: Network reliability estimation

Input: Distribution network $G = (N, A)$, installed switch positions S.
Output: Estimated reliability EENS.

EENS $\leftarrow 0$
for $\forall a \in A_{nc}$ **do**
    Simulate a failure f in the feeder a
    Expand the failure to define the black area $\mathcal{S}(f)$
    Isolate the black area by opening the frontier switches $\mathcal{F}(\mathcal{S}(f))$
    Determine affected nodes N_f with a service restoration algorithm
    $\text{EENS}_f \leftarrow \lambda_f r_f \sum_{n \in N_f} P_n$
    EENS $\leftarrow$ EENS + EENS_f
end for
return EENS

Algorithm 3.2: Network reliability estimation by sectors

Input: Distribution Network $G = (N, A)$, installed switch positions S.
Output: Estimated reliability EENS.

EENS $\leftarrow 0$
for $\forall \mathcal{S}_i \in \mathcal{SS}$ **do**
    Simulate a failure f in $\mathcal{S}_i$
    Assume the black area $\mathcal{S}(f) = \mathcal{S}_i$
    Isolate the black area by opening the frontier switches $\mathcal{F}(\mathcal{S}(f)) = \mathcal{F}(\mathcal{S}_i)$
    Determine affected nodes N_f with a service restoration algorithm
    $\text{EENS}_f \leftarrow \sum_{a \in E(\mathcal{S}(f))} \lambda_a r_a \cdot \sum_{n \in N_f} P_n$
    EENS $\leftarrow$ EENS + EENS_f
end for
for $\forall a = \{u, v\} \in A_{nc}, B_a = 1$ **do** // Frontier feeder lines
    Simulate a failure f in a
    Assume the black area $\mathcal{S}(f) = \mathcal{S}(a) = \mathcal{S}(u) \cup \mathcal{S}(v) \cup (\{u, v\}, a)$
    Isolate the black area by opening the frontier switches
    $\mathcal{F}(\mathcal{S}(f)) = (\mathcal{F}(\mathcal{S}(u)) \cup \mathcal{F}(\mathcal{S}(v))) \setminus \{a\}$
    Determine affected nodes N_f with a service restoration algorithm
    $\text{EENS}_f \leftarrow \lambda_f r_f \sum_{n \in N_f} P_n$
    EENS $\leftarrow$ EENS + EENS_f
end for
return EENS

second is the summation of the partial EENS of every frontier line. This calculates the same EENS, but reduces the number of failure simulations.

3.3 Reliability lower bound

A lower bound for the EENS reliability measure can be obtained by dropping the electrical constraints and evaluating the demand that can possibly be served based only on the network connectivity. Algorithm 3.3 represents a breadth-first implementation that determines the affected nodes. This method was originally proposed by Costa et al. (2007). First, it assumes that no consumer has energy. Then, it expands the energy across the network, starting from the substations, to neighbour nodes without failure, through normally closed feeder lines or loop lines with switches. Finally, it returns the set of remaining non-served load points.

Algorithm 3.3: Lower bound service restoration

Input: Distribution network $G = (N, A)$, installed switch positions S, black sector $\mathcal{S}(f)$.
Output: Set of affected nodes N_f.

$N_f \leftarrow N_C$
$Q \leftarrow N_S$
while $Q \neq \emptyset$ **do**
    pick a node $u \in Q$
    $Q \leftarrow Q \setminus \{u\}$
    for $\forall a = \{u, v\}, v \in N, \{u, v\} \in A$ **do**
        if $v \in N_f$ **and** $v \notin \mathcal{S}(f)$ **then** // Has no failure and no energy
            if $a \in A_{nc}$ **or** $B_a = 1$ **then** // Is normally closed or has a switch
                $N_f \leftarrow N_f \setminus \{v\}$
                $Q \leftarrow Q \cup \{v\}$
            end if
        end if
    end for
end while
return N_f

3.4 Reliability upper bound

We can find an upper bound for the EENS reliability measure by considering the electrical feasibility test and calculating an overestimation of the restored area. Algorithm 3.4 presents an improved version of the upper bound service restoration algorithm proposed originally by Costa et al. (2008). The improvement consists in evaluating first the basic circuit, reducing the number of electrical feasibility evaluations in an optimistic fashion.

The algorithm starts assuming that no consumer has energy, and expands the first test sector to its basic circuit avoiding the black sector (lines 4-12). To do this, the algorithm considers normally closed lines as connected and normally open lines as frontier. If the basic circuit is not electrically feasible, the algorithm restarts from the substation sector, as proposed by Costa et al. (2008). Then, the test sector is expanded sector by sector, closing one frontier switch at a time and reevaluating the electrical feasibility. If the expansion is feasible, the new frontier list is updated, otherwise, the expansion is reverted. When there are no more frontiers, the set of affected nodes excludes the nodes attended in the last feasible test sector (line 31). Finally, the algorithm returns the set of remaining non-served load points.

The electrical feasibility evaluation applied in the lines 13, 17 and 23 verifies the electrical constraints described in Section 2.1.3. To do this, the network power flow is calculated by a backward-forward sweep proposed by Baran & Wu (1989a, 1989b) and implemented by Costa et al. (2008). It represents the most time consuming part of the reliability upper bound estimation.

Observe that Algorithm 3.3 determines the connected components of the graph and has a unique solution, i.e., its result does not depend on the order of switches selection in line 4. In contrast, Algorithm 3.4 depends on the order of selections in lines 6 and 20. Furthermore, there exists an order that corresponds to the optimal solution. Algorithm 3.4 is a greedy approximation to the best value.

Algorithm 3.4: Upper bound service restoration

Input: Distribution network $G = (N, A)$, installed switch positions S, black sector $\mathcal{S}(f)$.
Output: Set of affected nodes N_f.

$N_f \leftarrow N_C$
for $\forall s \in N_S$ **do**
  $\mathcal{S}_{test} \leftarrow \mathcal{S}(s)$
  $L \leftarrow \mathcal{F}(\mathcal{S}_{test}) \cap A_{nc}$
  while $L \neq \emptyset$ **do** // Expand the basic circuit
    pick $a = \{u, v\} \in L$, $u \in \mathcal{S}_{test}$, $v \notin \mathcal{S}_{test}$
    if $\mathcal{S}(v) \neq \mathcal{S}(f)$ **then**
      $\mathcal{S}_{test} \leftarrow \mathcal{S}_{test} \cup \mathcal{S}(v) \cup (\{u, v\}, \{a\})$
      $L \leftarrow L \cup (\mathcal{F}(\mathcal{S}(v)) \cap A_{nc})$
    end if
    $L \leftarrow L \setminus \{a\}$
  end while
  if $\mathcal{S}_{test}$ is not electrically feasible **then**
    $\mathcal{S}_{test} \leftarrow \mathcal{S}(s)$ // Restart from substation sector
  end if
  $L \leftarrow \mathcal{F}(\mathcal{S}_{test})$
  if $\mathcal{S}_{test}$ is electrically feasible **then**
    while $L \neq \emptyset$ **do**
      $\mathcal{S}_{bk} \leftarrow \mathcal{S}_{test}$
      pick $a = \{u, v\} \in L$, $u \in \mathcal{S}_{test}$, $v \notin \mathcal{S}_{test}$
      if $\mathcal{S}(v) \neq \mathcal{S}(f)$ **then**
        $\mathcal{S}_{test} \leftarrow \mathcal{S}_{test} \cup \mathcal{S}(v) \cup (\{u, v\}, \{a\})$
        if $\mathcal{S}_{test}$ is electrically feasible **then**
          $L \leftarrow L \cup (\mathcal{F}(\mathcal{S}(v)) \cap \mathcal{F}(\mathcal{S}_{test}))$ // Update frontier list
        else
          $\mathcal{S}_{test} \leftarrow \mathcal{S}_{bk}$ // Restore last feasible $\mathcal{S}_{test}$
        end if
      end if
      $L \leftarrow L \setminus \{a\}$
    end while
    $N_f \leftarrow N_f \setminus V(\mathcal{S}_{test})$
  end if
end for
return N_f

4 ALGORITHMS FOR SWITCH ALLOCATION

This chapter explains the heuristics and metaheuristics developed for solving the switch allocation problem. The order they are presented is not chronological, but was chosen facilitate their explanation and understanding.

The first approach is a greedy constructive algorithm and it is presented in Section 4.1. Section 4.2 presents a greedy randomized adaptive search procedure (GRASP) for the switch allocation problem. Section 4.2.2 explains the neighbourhood that is explored by local search strategies. Section 4.3 shows an implementation of tabu search for the switch allocation problem. A sample construction algorithm and a sample local search algorithm are explained in Section 4.4. And finally, Section 4.5 presents an iterated sample construction with path relinking for the switch allocation problem.

4.1 Greedy algorithm

The most intuitive heuristic to solve the switch allocation problem is probably a greedy algorithm. Algorithm 4.1 depicts a greedy algorithm. It builds a solution by installing one switch at a time, selecting the switch location with the largest reliability improvement in each step. This algorithm was originally proposed by Costa et al. (2007, 2008).

Algorithm 4.1: Greedy construction.

Input: Distribution network $G = (N, A)$, number of switches k.
Output: Set of lines with installed switches A_B.

$A_B \leftarrow \emptyset$
while $|A_B| < k$ **do**
  $Candidate\ List \leftarrow A \setminus A_B$
  Estimate reliability gain of all elements in $Candidate\ List$
  $a \leftarrow$ the best switch location from $Candidate\ List$
  $A_B \leftarrow A_B \cup \{a\}$
end while
return A_B

4.2 Greedy randomized adaptive search procedure (GRASP)

Algorithm 4.2 illustrates a generic GRASP in pseudo-code. According to Resende & Ribeiro (2003), GRASP is an iterative process, where each 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 result of a GRASP is best solution found over all the iterations. The stop criterion might be a fixed time or a maximum number of iterations.

Algorithm 4.2: Generic GRASP.

while stop criterion is not satisfied **do**
  $Constructed\ Solution \leftarrow$ Semi-greedy construction
  $Solution \leftarrow$ Local Search ($Constructed\ Solution$)
  if $Solution$ is better than $Best\ Solution$ **then**
    $Best\ Solution \leftarrow Solution$
  end if
end while
return $Best\ Solution$

Next, we explain the semi-greedy construction phase, the neighbourhood for local exploration and the local search phase.

4.2.1 Semi-greedy construction

The semi-greedy construction phase builds a feasible solution one element at a time. As illustrated in Algorithm 4.3, a semi-greedy construction selects randomly

Algorithm 4.3: Semi-greedy construction.

Input: Distribution network $G = (N, A)$, number of switches k, α randomness.
Output: Set of lines with installed switches A_B.

```
1: A_B ← ∅
2: while |A_B| < k do
3:     Candidate List ← A \ A_B
4:     Estimate reliability gain of all elements in Candidate List
5:     Restricted Candidate List ← α fraction of best elements in Candidate List
6:     a ← select randomly a switch location from Restricted Candidate List
7:     A_B ← A_B ∪ {a}
8: end while
9: return A_B
```

one element from a restricted candidate list with the best elements, instead of the best element like greedy algorithm.

For the switch allocation problem, the candidate list is built by ordering all possible switch install locations according to the reliability improvement of installing each switch. Then, a fraction of α switches with the highest reliability improvements is kept in the restricted candidate list. Specifically, it contains the best $max(\alpha * |A \setminus A_B|, 1)$ fraction of the $|A \setminus A_B|$ lines without switches. Thus, a value of $\alpha = 0$ is equivalent to a greedy algorithm and selects always the best element, and with $\alpha = 1$ selects randomly one line after evaluating all of them. Finally, the selected switch is added to the solution.

According to Feo, Resende & Smith (1994), the adaptive component of GRASP emerges from the update of the benefits of previous selected elements when GRASP evaluates the benefits associated with the addition of a new element in each iteration. The greedy component of GRASP is the restriction of the best elements in the candidate list (depending on the α parameter). The probabilistic component of GRASP is reflected in the random selection of one element from the restricted candidate list, but not always the best one. This method for selection of elements allows the creation of a good set of different solutions at each GRASP iteration without affecting the adaptive greedy component. It is necessary to generate a good pool of initial solutions for the subsequent local search.

4.2.2 Local search

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. A local search algorithm repetitively replaces the current solution with a better neighbour.

Given a solution for the switch allocation problem represented by a set of lines with switches A_B, the neighbourhood $\mathcal{N}(A_B) = \{A'_B\}$ is defined by every solution $A'_B = (A_B \setminus \{a\}) \cup \{b\}$ that results from removing a switch from a line $a \in A_B$ and installing it at another line $b \in A \setminus A_B$. This neighbourhood is used by every local search in this dissertation.

The local search was implemented in two ways: best improvement and first improvement. Best improvement searches through all the neighbourhood to select the best neighbour for the next iteration, while first improvement accepts the first better solution found and breaks the search out to the next iteration without exploring all the neighbourhood.

Algorithm 4.4: First improvement local search.

Input: Distribution network $G = (N, A)$, initial solution A_{B0}.
Output: Best found solution A_{Bbest}.

1: Estimate reliability of A_{B0}
2: $A_{Bbest} \leftarrow A_{B0}$
3: **while** stop criterion is not satisfied **do**
4: $\quad A_B \leftarrow A_{Bbest}$
5: $\quad$ **for** $\forall a \in A_B$ **do** // With switches
6: $\quad\quad$ **for** $\forall b \in A \setminus A_B$ **do** // Without switches
7: $\quad\quad\quad A_{Bnew} \leftarrow (A_B \setminus \{a\}) \cup \{b\}$ // Moves the switch
8: $\quad\quad\quad$ Estimate reliability of A_{Bnew}
9: $\quad\quad\quad$ **if** A_{Bnew} has better reliability than A_{Bbest} **then**
10: $\quad\quad\quad\quad A_{Bbest} \leftarrow A_{Bnew}$
11: $\quad\quad\quad\quad$ **exit for** to line 3 // Missing line in best improvement
12: $\quad\quad\quad$ **end if**
13: $\quad\quad$ **end for**
14: $\quad$ **end for**
15: **end while**
16: **return** A_{Bbest}

Algorithm 4.4 depicts the first improvement local search from our implementation. It starts from the initial solution created by the semi-greedy constructive algorithm, and explores the neighbourhood previously explained. If the algorithm finds a better solution, it becomes the current solution. The search stops when there are no better solutions in the neighbourhood. Finally, the last solution is returned. We obtain a best improvement local search by removing the line 11 from Algorithm 4.4.

4.3 Tabu Search

Tabu search is a metaheuristic proposed by Glover (1989, 1990) as a method that allows local search heuristics to overcome local minima. When a local minimum is found, the local search continues its execution by allowing non-improving movements. Cycling is avoided by forbidding the return to recent elements in search trajectory. This is accomplished by maintaining a tabu list with the recent visited solutions (or certain elements of recent solution) by a determinated tabu tenure time (or number of iterations).

The implemented tabu search explores the same neighbourhood described in Section 4.2.2. Algorithm 4.5 shows our implementation of a tabu search for the switch allocation problem. It starts with a given initial solution that can be generated by a random construction. It searches for a better solution in the neighbourhood of the current solution. If the algorithm finds a better solution, it updates the current solution and continues the search (line 13). If there is no better solution in the neighbourhood, the search does not stop. It replaces the current solution with the best solution in the neighbourhood, even if it is worse than the current one (line 15). And, in order to avoid cycling, the method forbids some old movements in a tabu list by a tenure time or number of iterations. In this case, the tabu list contains the last lines that had a switch (line 19). The stop criterion might be a fixed time, a maximum number of iterations or a number of iterations without improvement. Finally, the best overall solution is returned.

Algorithm 4.5: Tabu search.

Input: Distribution network $G = (N, A)$, initial solution A_{B0}.
Output: Best found solution A_{Bbest}.

1: Estimate reliability of A_{B0}
2: $A_{Bnext} \leftarrow A_{Bbest} \leftarrow A_{B0}$
3: $Tabu\ List \leftarrow \emptyset$
4: **while** stop criterion is not satisfied **do**
5:     $A_B \leftarrow A_{Bnext}$
6:     $A_{Bnext} \leftarrow \emptyset$
7:     **for** $\forall a \in A_B$ **do** // With switches
8:         **for** $\forall b \in A \setminus A_B$ **do** // Without switches
9:             $A_{Bnew} \leftarrow (A_B \setminus \{a\}) \cup \{b\}$ // Moves the switch
10:             Estimate reliability of A_{Bnew}
11:             **if** A_{Bnew} has better reliability than A_{Bbest} **then**
12:                 $A_{Bbest} \leftarrow A_{Bnext} \leftarrow A_{Bnew}$
13:                 **exit for** to line 19
14:             **else if** A_{Bnew} has better reliability than A_{Bnext} **and** $b \notin Tabu\ List$ **then**
15:                 $A_{Bnext} \leftarrow A_{Bnew}$
16:             **end if**
17:         **end for**
18:     **end for**
19:     $Tabu\ List \leftarrow Tabu\ List \cup (A_B \setminus A_{Bnext})$ // Forbids old switch location $\{a\}$
20:     Update tenures in $Tabu\ List$
21: **end while**
22: **return** A_{Bbest}

4.4 Sample construction and local search

This section describes a sample construction and a sample local search. Our motivation for proposing these sample algorithms is to reduce the number of reliability estimations within construction and local search algorithms. The reliability estimation is the most expensive operation in the algorithms proposed for solving the switch allocation problem, mainly when it verifies the electrical constraints.

4.4.1 Sample construction

The sample construction algorithm builds a feasible solution element by element. The basic idea is to take a sample of the possible elements a priori, evaluate them, and choose the best among them to include it into the solution.

Algorithm 4.6: Sample construction.

Input: Distribution network $G = (N, A)$, number of switches k, β sample percentage.

Output: Set of lines with installed switches A_B.

$A_B \leftarrow \emptyset$

while $|A_B| < k$ **do**

  $Candidate\ List \leftarrow A \setminus A_B$

  $Sample\ Candidate\ List \leftarrow$ sample randomly β percent from $Candidate\ List$

  Estimate reliability gain of all elements in $Sample\ Candidate\ List$

  $a \leftarrow$ select the best switch location from $Sample\ Candidate\ List$

  $A_B \leftarrow A_B \cup \{a\}$

end while

return A_B

Algorithm 4.6 depicts the sample construction algorithm for the switch allocation problem. First, it selects randomly a sample of β percent of the candidate feeder lines. Then, it evaluates the installation of a switch in each line of the sample candidate list. And finally, it includes the best switch candidate line of the sample into the solution. Specifically, the size of the sample is $max(\beta * |A \setminus A_B|/100, 1)$ of the $|A \setminus A_B|$ lines without switches. Thus, a sample of $\beta = 100\%$ is equivalent to a greedy algorithm, because it evaluates all the possible feeder lines to install one switch. And a sample of $\beta = 0\%$ chooses randomly one feeder line and installs the switch.

Comparing the sample and the semi-greedy construction algorithms, we note that both create a small list of candidates and select one element to be added to the current solution. The difference lies in the way they create that small list and in the evaluation of the elements. The semi-greedy construction first evaluates every possible element, and then restricts the candidate list to finally select one element randomly. The sample construction first reduces the candidate list to a sample, and then evaluates the reduced list to select the best one.

Sample construction can be an alternative construction for GRASP, because the adaptive, greedy and probabilistic components are maintainded in a different fashion. The random selection of the sample manifests the probabilistic component. The adaptive component is reflected by the influence of the previously selected elements over the reliability gain of the elements in the sample candidate list. And the greedy component defines the selection of the new element. Resende & Werneck (2004) describe this and other alternatives to GRASP construction.

4.4.2 Sample local search

A similar idea is applied to create a sample local search, i.e., to take a sample of the neighbours a priori, evaluate them, and select the best among them.

Algorithm 4.7 depicts a sample local search for the switch allocation problem. The initial solution can be generated by any construction algorithm, e.g., random, sample, or semi-greedy. It explores the same neighbourhood described in Section 4.2.2, but not completely. It samples β percent of the lines with switches ($a \in A_B$) and β percent of lines without switches ($b \in A \setminus A_B$) to relocate one switch. Thus, it reduces the size of the explored neighbourhood and the number of reliability estimations. If the algorithm finds a better solution in the sample, it is taken for the next iteration. Finally, it returns the last solution.

Algorithm 4.7: Sample local search.

Input: Distribution network $G = (N, A)$, initial solution A_{B0},
β sample percentage.
Output: Best found solution A_{Bbest}.

Estimate reliability of A_{B0}
$A_{Bbest} \leftarrow A_{B0}$
while stop criterion is not satisfied **do**
  $A_B \leftarrow A_{Bbest}$
  $A_{Sample1} \leftarrow$ sample randomly β line feeders from A_B
  $A_{Sample2} \leftarrow$ sample randomly β line feeders from $A \setminus A_B$
  for $\forall a \in A_{Sample1}$ **do** // With switches
    for $\forall b \in A_{Sample2}$ **do** // Without switches
      $A_{Bnew} \leftarrow (A_B \setminus \{a\}) \cup \{b\}$ // Moves the switch
      Estimate reliability of A_{Bnew}
      if A_{Bnew} has better reliability than A_{Bbest} **then**
        $A_{Bbest} \leftarrow A_{Bnew}$
      end if
    end for
  end for
end while
return A_{Bbest}

Obviously, this neighbourhood exploration is not exhaustive and does not guarantee to find the local minimum. Thus, the stop criterion may be a maximum number of iterations or a number of iterations without improvement. To guarantee that the local minimum is reached, we can execute a first or best improvement local search after the sample local search, or intersperse an exhaustive neighbourhood search after a fixed number of iterations or after a number of iterations without improvement.

4.5 Iterated sample construction with path relinking

Sample local search explores quickly the neighbourhood and improves quickly the solution at the beginning of the search, as shown in the results of Chapter 6. But, it looses effectiveness after a certain number of iterations because it is not an exhaustive local search. This motivated us to search for a more directed local search that can take advantage of the already available sample algorithms.

Iterated sample construction with path relinking is inspired by an iterated greedy algorithm applied to the permutation flowshop scheduling problem developed by Ruiz & Stützle (2007). This algorithm is closely related to iterated local search (STÜTZLE, 1998). It can be considered as an iterated local search with a greedy perturbation scheme.

A general iterated greedy algorithm is presented in Algorithm 4.8. It starts generating a greedy solution and applying a local search to it. Later, it iterates over three phases: destruction, construction and local improvement. During the destruction phase some random elements are removed from the previous complete candidate solution. During the construction phase, the solution is reconstructed greedily. Once a new solution is complete, it is improved by a local search. Finally, the best found solution is returned.

Algorithm 4.8: Iterated greedy algorithm.

Input: number of elements d for destruction and reconstruction
Output: $Best\ Solution$

```
Constructed Solution ← Greedy construction
Solution ← Local Search (Constructed Solution)
Best Solution ← Solution
while stop criterion is not satisfied do
    Solution′ ← remove d elements randomly from Solution
    Solution″ ← reconstruct greedily d elements from Solution′
    Solution ← Local Search (Solution″)
    if Solution is better than Best Solution then
        Best Solution ← Solution
    end if
end while
return Best Solution
```

Algorithm 4.9: Iterated sample construction with path relinking.

Input: number of elements d for destruction and reconstruction.
Output: $Best\ Solution$

```
Best Solution ← Solution ← Sample construction
while stop criterion is not satisfied do
    Solution′ ← remove d elements randomly from Solution
    Solution″ ← Sample reconstruct d elements from Solution′
    Solution ← Path relinking from Solution″ to Best Solution
    if Solution is better than Best Solution then
        Best Solution ← Solution
    end if
end while
return Best Solution
```

In this section, we propose an iterated sample construction with path relinking. Its pseudocode is depicted in Algorithm 4.9. It initializes by generating a solution with a sample construction algorithm. After initialization, it iterates over three steps. First, some elements are removed from the previous solution. Then, the solution is reconstructed by a sample construction algorithm. And later, a path relinking uses the best solution to guide a directed local search starting from the current solution. Finally, the best solution is returned.

Iterated sample construction with path relinking works similar to iterated greedy algorithm, but uses sample construction instead of greedy construction, and uses path relinking as a guided local search. In this way, it reduces the number of reliability estimations during construction and local search phases.

Next, we explain with more detail the sample reconstruction and the path relinking for the switch allocation problem.

4.5.1 Destruction and sample construction

The destruction and sample reconstruction algorithm for the switch allocation problem is presented in Algorithm 4.10. This algorithm takes a previous solution A_{B0} and modifies it to generate a new solution A_B. The destruction phase randomly removes k elements from the previous solution. The construction phase replaces the k elements, one by one, using the sample construction algorithm. To replace one element, the algorithm first selects randomly a sample of β percent of the feeder lines that are not part of neither the previous nor the new solutions. And then, it evaluates the elements in the sample candidate list and includes the best into the solution. Finally, the algorithm returns the new solution.

Algorithm 4.10: Destruction and sample reconstruction.

Input: Distribution network $G = (N, A)$, number of switches k,
β sample percentage, previous set of lines with switches A_{B0}.
Output: Set of lines with installed switches A_B.

```
A_B ← A_B0
while |A_B| > |A_B0| − k do                                   // Destruction phase
    a ← select randomly an element from A_B
    A_B ← A_B \ {a}
end while
while |A_B| < |A_B0| do                                       // Construction phase
    Candidate List ← A \ (A_B ∪ A_B0)
    Sample Candidate List ← sample randomly β percent from Candidate List
    Estimate reliability gain of all elements in Sample Candidate List
    a ← select the best switch location from Sample Candidate List
    A_B ← A_B ∪ {a}
end while
return A_B
```

4.5.2 Path relinking

Originally, path relinking was suggested as an improvement to scatter search by Glover (1994). Algorithm 4.11 presents the proposed path relinking for the switch allocation problem. It receives an initial solution A_{B0} and a guiding solution

A_{Bg} as parameters. Path relinking iteratively shortens the distance between the two solutions by relocating one switch of the initial solution to a position defined by the guiding solution. To determine the switch relocation, path relinking explores the same neighbourhood described in Section 4.2.2, but limited to the differences between the initial and the guiding solutions, i.e., the neighbourhood of the initial solution $\mathcal{N}(A_{B0})$ is limited to remove a switch from a line in the initial solution that is not part of the guiding solutions ($a \in A_{B0} \setminus A_{Bg}$) and to install it at a line that is part of the guiding solution but not of the initial solution ($b \in A_{Bg} \setminus A_{B0}$). The best element in the restricted neighbourhood always replaces the initial solution in the next iteration. Before finishing each iteration, the best overall solution is updated. The process is repeated until the difference between the initial and the guiding solutions is one line with switch. Finally, path relinking returns the best overall solution.

Algorithm 4.11: Path relinking.

Input: Distribution network $G = (N, A)$, initial set of switches A_{B0}, guiding set of switches A_{Bg}.
Output: Best found solution A_{Bbest}.

Estimate reliability of A_{B0} and A_{Bg}
$A_{Bnext} \leftarrow A_{Bbest} \leftarrow A_{B0}$
while $|A_{B0} \setminus A_{Bg}| > 1$ **do**
  $A_{B0} \leftarrow A_{Bnext}$
  $A_{Bnext} \leftarrow \emptyset$
  for $\forall a \in A_{B0} \setminus A_{Bg}$ **do** // With switches
    for $\forall b \in A_{Bg} \setminus A_{B0}$ **do** // Without switches
      $A_{Bnew} \leftarrow (A_{B0} \setminus \{a\}) \cup \{b\}$ // Moves the switch
      Estimate reliability of A_{Bnew}
      if A_{Bnew} has better reliability than A_{Bnext} **then**
        $A_{Bnext} \leftarrow A_{Bnew}$
      end if
    end for
  end for
  if A_{Bnext} has better reliability than A_{Bbest} **then**
    $A_{Bbest} \leftarrow A_{Bnext}$ // Keeps track of the best overall solution
  end if
end while
return A_{Bbest}

5 INSTANCES OF POWER DISTRIBUTION NETWORKS

A problem for our research was the lack of available distribution network instances to test our algorithms. Instances in the literature are often incomplete, with missing network reliability or electrical constraints parameters. For example, the instance of Levitin, Mazal-Tov & Elmakis (1995) does not specify the electrical characteristics of the lines, the voltage level and the number of consumers. Instances for the network reconfiguration problem with loss reduction such as Civanlar et al. (1988) and Su, Chang & Chiou (2005) miss information to estimate reliability, like failure rates and outage time. Other networks are simply without complete information because they are private and cannot be disclosed. For example, instances used by Carvalho, Ferreira & Silva (2005), Chen et al. (2006), Lin et al. (2006), Villasanti, Baran & Gardel (2008) are real private networks.

In order to use the reliability measures described in Section 3.1, information about failure rates, outage time, demand and number of consumers must be completed. And, to take into account electrical constraints, information about substation capacities, line capacities and resistance, and voltage drop limits must be introduced. The instances presented in this chapter were preparated in collaboration with Mariane Machado, during her work with a DTI-CNPq grant. Next, we explain the modifications and data completion applied to the networks used in our tests. Section 5.1 presents the first instances used in our research, Section 5.2 presents an instance found with almost complete information, Section 5.3 describes some instances generated from telecommunication networks, and Section 5.4 shows instances from a repository for service restoration with loss reduction.

5.1 Instances of Baran and Augugliaro

The first two instances used in our research were originally adapted for the service restoration problem by Garcia (2005) from Baran & Wu (1989a) and Augugliaro, Dusonchet & Sanseverino (2000). Later they were used for the switch allocation problem by Costa et al. (2007, 2008) and Benavides et al. (2009b), but only with APSE reliability measure. In order to use other measures described in Section 3.1, electrical data and failure statistics had to be completed.

Four instances were created, BR and BU from Baran instance and AR and AU from Augugliaro instance. Figures 5.1 and 5.2 show the topology of Baran and Augugliaro instances, respectively, and Table 5.1 shows details of these instances. The first letter stands for the authors, and the second letter represents the distribution used to assign the load for consumers: Consumer demand was completed for BR and AR with random values, and for BU and AU with a uniform distribution of a constant value. The latter value is the result of dividing the

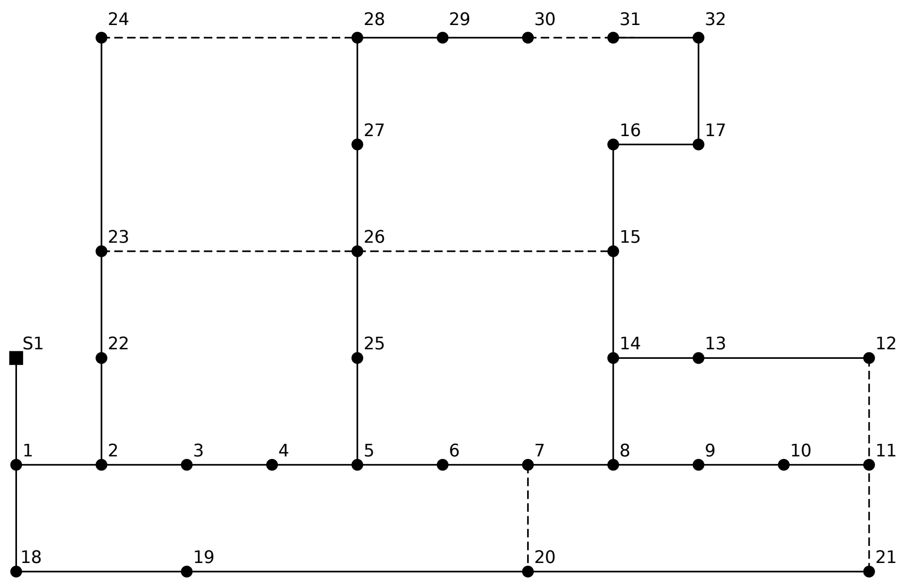


Figure 5.1: Baran-Wu instance.

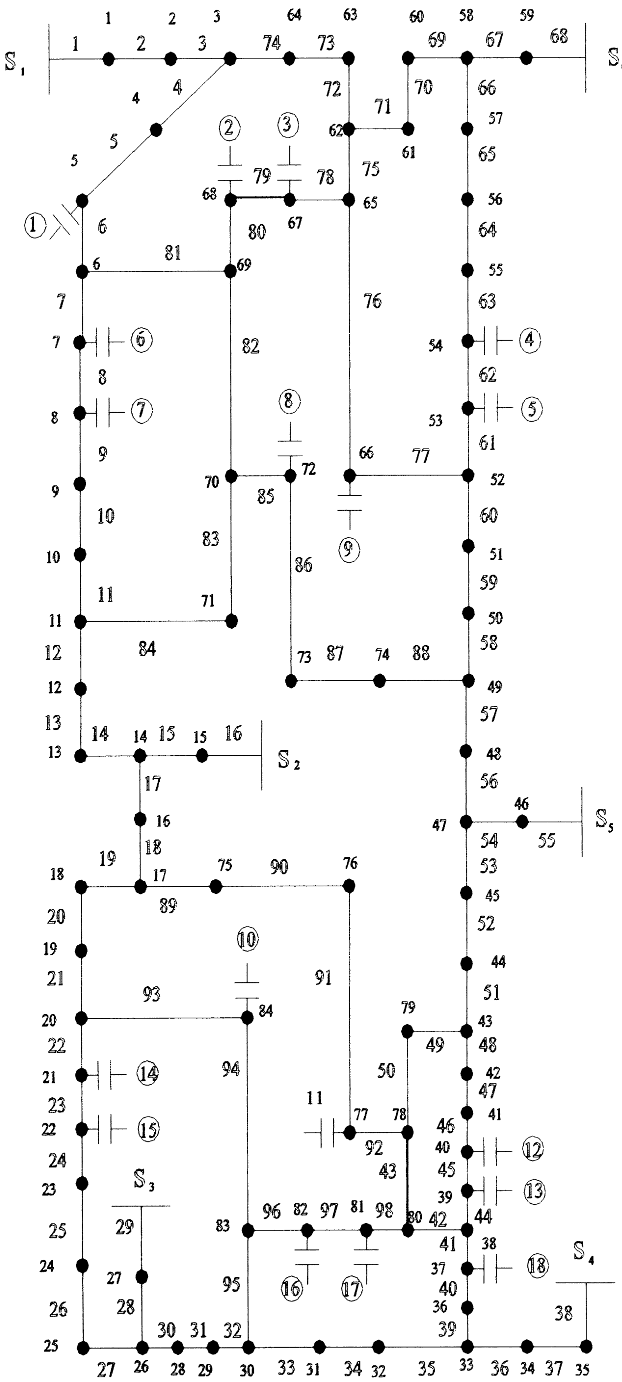


Figure 5.2: Augugliaro-Dusonchet-Sanseverino instance.

Table 5.1: First group of problem instances.

Network instances	Baran		Augugliaro	
	BR	BU	AR	AU
Substations	1	1	5	5
Consumers	32	32	80	80
feeder lines	32	32	80	80
loop lines	7	7	29	29
Operation voltage (kV)	12.66	12.66	20.0	20.0
Total power demand (MW)	3.7	3.7	28.6	28.6
Consumer power factor *	0.8	0.8	$[0.39, 0.99]$	$[0.39, 0.99]$
Consumer demand * (kW)	$[45, 420]$	116	$[0, 585]$	357
Number of consumers *	$[24, 168]$	46	$[0, 234]$	142
Line resistance (Ω)	$[0.050, 1.093]$	$[0.050, 1.093]$	0.9341	0.9341
Line reactance (Ω)	$[0.026, 0.574]$	$[0.026, 0.574]$	0.4905	0.4905
Line capacity (A)	4000	4000	4000	4000
Line failure rates	$[0.003, 0.075]$	$[0.003, 0.075]$	0.065	0.065

* per load point.

nominal total demand by the number of load points. The number of consumers was calculated assuming that each one consumes $2.5\ kW$. The capacity of every feeder line was set to $I_{MAX} = 4000\ A$, assuming the same cable size for every feeder line in the network instance. Resistance and reactance of Baran instances were reduced by a factor of 0.55 from the original values to simulate a cable with a larger diameter, resulting in a resistance $r = 0.9341\ \Omega/km$ and a reactance $x = 0.4905\ \Omega/km$. Resistance and reactance of Augugliaro instances were set to constant values, assuming that every feeder line is $1\ km$ long. The adapted failure rate $\lambda = 0.065\ f/yr/km$ is a function of the distance, like the resistance. Thus, we decided to multiply the resistance by a factor to create the final failure rate $\lambda = 0.0696(f/yr/\Omega) * r$, given in f/yr.

5.2 Billinton instance

Another instance used in our tests is the RBTS bus 4 proposed by Roy Billinton in (ALLAN et al., 1991). RBTS stands for Roy Billinton Test System. Table 5.2 shows details for RBTS bus 4 and Figure 5.3 shows its topology. This is the most complete instance we found. It has 38 load points and 72 lines. 30 joint points with no power demand were added. RBTS bus 4 only has lines of three different distances ($0.6, 0.75$ and 0.8 km). To complete the necessary information, we followed the adaptation example of a part of the RBTS bus 6 by Falaghi, Haghifam & Singh (2009). We assume a resistance $r = 0.257$ Ω/km, a reactance $x = 0.087$ Ω/km, a failure rate $\lambda = 0.065$ $f/yr/km$, and a capacity $I_{MAX} = 500$ A for every line.

Table 5.2: RBTS bus 4 instance details.

Network instances	B4
Substations	3
Consumers	$38 + 30$
feeder lines	67
loop lines	5
Operation voltage (kV)	11.0
Total power demand (MW)	24.58
Consumer power factor *	0.9
Consumer demand * (kW)	$[415, 1500]$
Number of consumers *	$[1, 220]$
Line resistance (Ω)	$[0.1542, 0.2056]$
Line reactance (Ω)	$[0.0522, 0.0696]$
Line capacity (A)	500
Line failure rates	$[0.039, 0.052]$

* per load point.

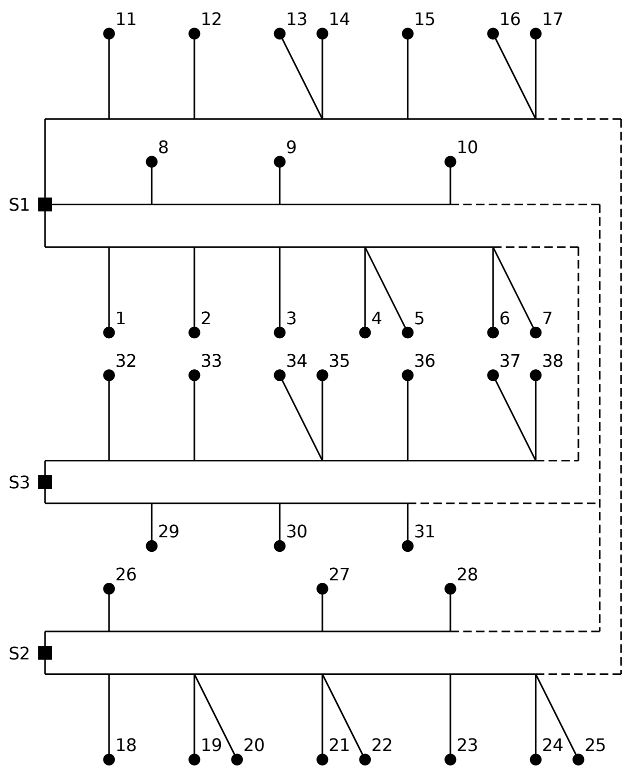


Figure 5.3: RBTS bus 4 instance.

5.3 Synthetic instances

Due to the small number of available electric distribution network instances, we designed a simple electrical distribution network generator. It receives a network topology or graph as input, and generates an electrical distribution network instance by adding values of electrical characteristics, based on information found in Pabla (2004) and Pransini (2005).

The generated instance has one substation selected by the user. The basic circuit of the network is defined by the forward arcs of a breadth-first search algorithm. The rest of the links are defined to be loop lines. The operation voltage is chosen randomly between 2400 and $34500\ V$. Power demand for load points are generated randomly between 50 and $250\ kW$. The number of consumers for each load point is the demand divided by $2.5\ kW$ per consumer. Reactive power load is calculated with a random power factor between 0.4 and 1.0. Substation capacity is 30% higher than the nominal total power load demand. The generator uses the same cable type with resistance $r = 0.9341\ \Omega/km$, reactance $x = 0.4905\ \Omega/km$, failure rate $\lambda = 0.065\ f/yr/km$, and capacity $I_{MAX} = 4000\ A$ for every line.

Twelve instances were generated. Their topologies were originally proposed for telecommunication problems by Fortz & Thorup (2004). They are divided in three classes, each one with four instances. The first class of networks named "hier" are 2-level hierarchical communication networks generated using a generator proposed by Zegura (2005). The networks named "rand" are random networks, which have a parameter that controls the density of the network and the probability of creating arcs. Finally, the last four networks are Waxman graphs, i.e., planar graphs where the probability of creating an arc between two nodes is inversely proportional to their euclidean distance. Thus, closer nodes have a higher probability of being connected by an arc than distant nodes (WAXMAN, 1988). We added a prefix "e_" to emphasize that these are electrical instances. Tables 5.3, 5.4 and 5.5 show details for synthetic instances.

Table 5.3: Hierarchical synthetic instances.

Network instances	e_hier50a	e_hier50b	e_hier100a	e_hier100b
Substations	1	1	1	1
Consumers	49	49	99	99
feeder lines	49	49	99	99
loop lines	25	57	41	81
Operation voltage (V)	15062	27595	6964	22802
Total power demand (kW)	6969	7931	14666	15260
Line distance (km)	$[0.002, 2.167]$	$[0.001, 1.590]$	$[0.001, 2.202]$	$[0.001, 1.746]$
Line resistance (Ω)	$[0.002, 2.024]$	$[0.001, 1.485]$	$[0.001, 2.058]$	$[0.001, 1.631]$
Line reactance (Ω)	$[0.001, 1.063]$	$[0.001, 0.780]$	$[0.001, 1.080]$	$[0.001, 0.857]$
Line failure rates	$[0.001, 0.141]$	$[0.001, 0.104]$	$[0.000, 0.143]$	$[0.000, 0.113]$

Table 5.4: Random synthetic instances.

Network instances	e_rand50a	e_rand50b	e_rand100a	e_rand100b
Substations	1	1	1	1
Consumers	49	49	99	99
feeder lines	49	49	99	99
loop lines	169	185	294	385
Operation voltage (V)	6045	20133	31039	30335
Total power demand (kW)	7258	7298	14890	15461
Line distance (km)	$[0.020, 1.179]$	$[0.027, 1.153]$	$[0.053, 1.137]$	$[0.025, 1.051]$
Line resistance (Ω)	$[0.018, 1.102]$	$[0.025, 1.077]$	$[0.050, 1.062]$	$[0.023, 0.982]$
Line reactance (Ω)	$[0.010, 0.579]$	$[0.013, 0.565]$	$[0.026, 0.558]$	$[0.012, 0.516]$
Line failure rates	$[0.001, 0.076]$	$[0.002, 0.075]$	$[0.004, 0.074]$	$[0.002, 0.068]$

Table 5.5: Waxman graph synthetic instances.

Network instances	e_wax50a	e_wax50b	e_wax100a	e_wax100b
Substations	1	1	1	1
Consumers	49	49	99	99
feeder lines	49	49	99	99
loop lines	113	171	281	363
Operation voltage (V)	24805	21505	5966	3646
Total power demand (kW)	7189	6751	14021	15379
Line distance (km)	$[0.029, 1.137]$	$[0.028, 1.052]$	$[0.010, 1.143]$	$[0.010, 1.204]$
Line resistance (Ω)	$[0.027, 1.062]$	$[0.026, 0.983]$	$[0.009, 1.069]$	$[0.009, 1.125]$
Line reactance (Ω)	$[0.014, 0.558]$	$[0.014, 0.516]$	$[0.005, 0.561]$	$[0.005, 0.591]$
Line failure rates	$[0.002, 0.074]$	$[0.002, 0.068]$	$[0.001, 0.074]$	$[0.001, 0.078]$

5.4 REDS: REpository of Distribution Systems

The last group of instances is the REpository of Distribution Systems (REDS). This is an effort of Kavasseri & Ababei (2009) for collecting and maintaining instances for research on power flow, network reconfiguration, etc. They propose a new format for easy parsing of the instances. This format inspired the actual template for our test files.

REDS is formed by eight instances. Instance R1 is the tree feeder instance of Civanlar et al. (1988) that was used in the examples of Chapter 2. Instance R2, also known as IEEE 30 bus, is presented by Eminoglu & Hocaoglu (2005). Instance R3 is the same instance of Baran & Wu (1989a), but it keeps the original resistance and reactance. Instance R4 was originally proposed by Su, Chang & Chiou (2005). Instances R5 and R6 were adapted and proposed by Guimarães & Castro (2005). Instances R7 and R8 are hypothetical testcases that REDS research group created using data from R5 and R6.

The instances had electrical information such as resistance, reactance, active and reactive power load. The rest of the information had to be completed. Demand and power factor are calculated with the equations ${D_n}^2 = {P_n}^2 + {Q_n}^2$ and $pf = P_n/D_n$. Power factor pf was set to a constant value 0.8, 0.85, or 0.9 when there is no demand in the load point. The number of consumers is the demand divided by 2.5 kW per consumer. No demand means no consumers, but the number is rounded up to 1 consumer if the demand is between 0.01 and 2.50 kW. Substation capacities are set to the double of the total demand in the underlying subtree, allowing substations to attend more load points in case of failures. Line capacities are 1000 A for every line. The failure rate is calculated as $\lambda = 0.0696(f/yr/\Omega) * r$ and given in f/yr. R2 is given in the per-unit values system, then the power load was multiplied by 1000. The operation voltage is set to 12660 V. R6, R7 and R8 presented some electrical problems that were solved with a higher operation voltage. R6 uses 33600 V, and R7 and R8 use 126600 V.

Note that voltage operation for R7 and R8 is not common in distribution systems, but the hypothetical networks cannot energize the whole network without this high voltage. Table 5.6 presents the details of the instances.

Table 5.6: REDS instances.

Network instances	R1	R2	R3	R4
Substations	3	1	1	11
Consumers	13	29	32	83
feeder lines	13	29	32	83
loop lines	3	1	5	13
Operation voltage (V)	12660	12660	12660	12660
Total power demand (kW)	33.8	102.3	4548.4	35200.0
Consumer power factor *	$[0.74, 0.99]$	$[0.84, 0.89]$	$[0.32, 0.99]$	$[0.71, 0.98]$
Consumer demand * (kW)	$[0.61, 5.83]$	$[0.0, 5.2]$	$[54.1, 632.5]$	$[0, 25000]$
Number of consumers *	$[1, 2]$	$[0, 2]$	$[22, 253]$	$[0, 1000]$
Line resistance (Ω)	$[0.04, 0.12]$	$[0.043, 0.260]$	$[0.092, 2.000]$	$[0.024, 0.537]$
Line reactance (Ω)	$[0.04, 0.18]$	$[0.007, 0.045]$	$[0.047, 2.000]$	$[0.052, 1.104]$
Line failure rates	$[0.003, 0.056]$	$[0.003, 0.018]$	$[0.006, 0.139]$	$[0.002, 0.037]$
Network instances	R5	R6	R7	R8
Substations	8	3	7	84
Consumers	135	201	873	10476
feeder lines	135	201	873	10476
loop lines	21	15	27	260
Operation voltage (V)	12660	33600	126600	126600
Total power demand (kW)	19963	32437	148990	1778644
Consumer power factor *	$[0.90, 0.93]$	0.85	$[0.27, 0.93]$	$[0.27, 0.93]$
Consumer demand * (kW)	$[0, 1636]$	$[0, 1211]$	$[1, 815]$	$[1, 815]$
Number of consumers *	$[24, 655]$	$[0, 485]$	$[1, 326]$	$[1, 326]$
Line resistance (Ω)	$[0.002, 2.962]$	$[0.000, 0.187]$	$[0.011, 2.963]$	$[0.011, 2.963]$
Line reactance (Ω)	$[0.004, 2.684]$	$[0.000, 0.254]$	$[0.017, 1.685]$	$[0.017, 1.685]$
Line failure rates	$[0.001, 0.206]$	$[0.003, 0.013]$	$[0.001, 0.206]$	$[0.001, 0.206]$

* per load point.

6 EXPERIMENTAL TESTS AND RESULTS

The first three sections present issues that emerged during the research and that affect the final results. Section 6.1 compares the reliability measures. Section 6.2 studies the influence of electrical constraints for different voltage drop limits. And Section 6.3 discusses two improvements for the reliability estimation algorithms described in Chapter 3.

The last four sections present the results of the algorithms proposed in Chapter 4 to solve the switch allocation problem. Section 6.4 compares the results obtained with a tabu search and a greedy construction algorithm. Section 6.5 shows the results of GRASP. Section 6.6 studies the performance of construction and local search algorithms, including sample algorithms. Finally, Section 6.7 presents the results achieved for the iterated sample construction with path relinking.

To compare the results of the algorithms, they were executed with the same parameters. Tests include both reliability estimation improvements studied in Section 6.3. We use EENS as standard measure, a constant outage time $r_f = 2h$, and a voltage drop limit for upper bound reliability estimation of five percent.

We must remark that the lower bounds presented in this chapter represent a method to estimate the network reliability, and not a lower bound for the optimal solution of the problem, because our algorithms approximate its best value. We study it, because it reduces the reliability estimation time with a simplification of the problem, and gives another approximation of the reliability to compare with.

To standardize the presented results, time values are in seconds and EENS values are in KWh/year. The scale in the graphs is Interruptions/year for SAIDI, MWh/year for EENS, and a percentage for APSE. When an average is presented, the corresponding standard deviation together (average $\pm$ standard deviation).

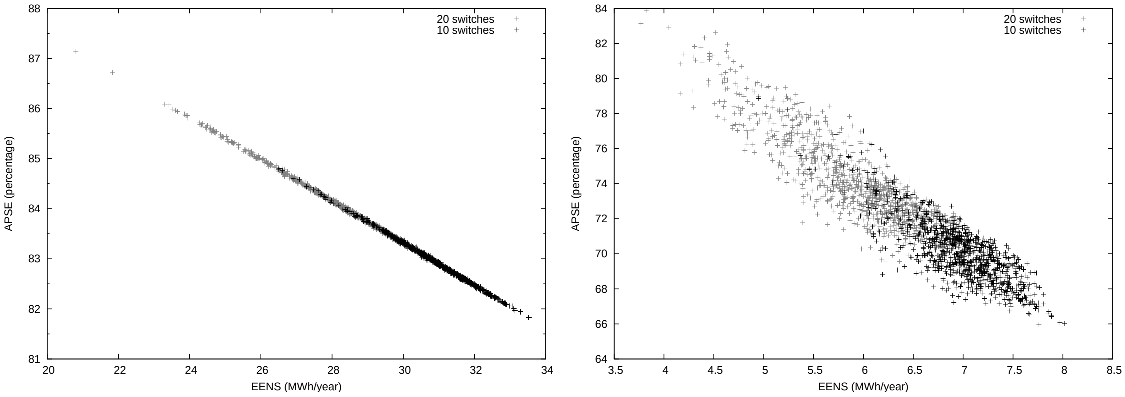
All the tests have been executed on an Intel Core 2 processor with a 2.33 GHz clock and 4 GB of main memory, and have been compiled with GNU C++ with the command "`g++ *.cpp -Wall -O3 -static`".

6.1 Comparison of reliability measures

An important question that emerges with the reliability estimation of distribution networks is which measure we should use in the tests. We found that many measures were used in the literature to express the reliability. Section 3.1 described the most commonly used measures. Here, we compare three reliability measures (APSE, EENS and SAIFI) to define one for the rest of our tests. Since we defined the average outage time r_f as a constant for our tests, we have an exact relation SAIDI $= r_f *$ SAIFI, and therefore implicitily also evaluate SAIDI. We do not compare with ECOST since cost data is volatile and difficult to obtain.

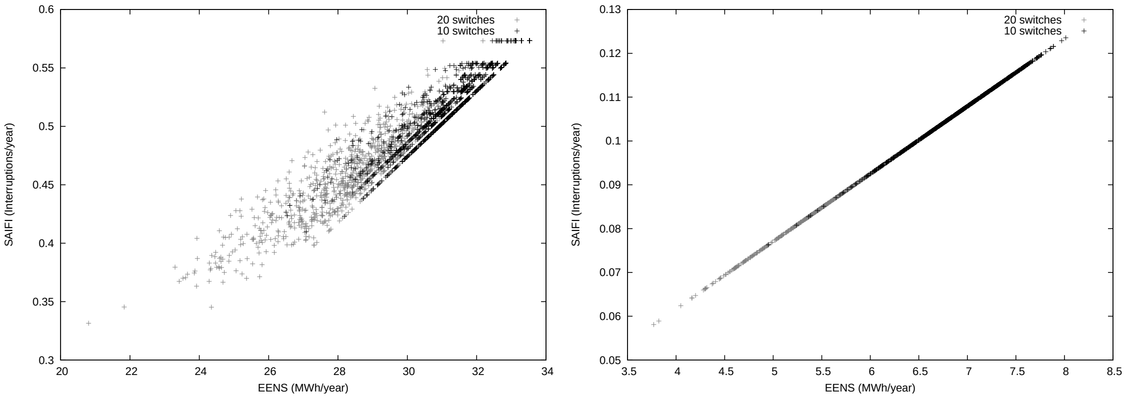
We present scatter plots comparing thereliability estimation of different measures on the same set of solutions. We estimate the upper bound reliability of 2000 randomly generated solutions, i.e., 1000 for allocating 10 switches and 1000 for 20 switches. Figure 6.1 compares APSE and EENS, Figure 6.2 compares SAIFI and EENS, and Figure 6.3 compares SAIFI and APSE. Figures 6.1a, 6.2a and 6.3a show values for the instance B4, while Figures 6.1b, 6.2b and 6.3b for the instance R6. The correlation coefficient is given in the caption of each figure.

As we can see in the figures, a strong negative correlation is found between APSE and EENS and between SAIFI and APSE, while SAIFI and EENS have a strong positive correlation. APSE is based on served load points while SAIFI and EENS are



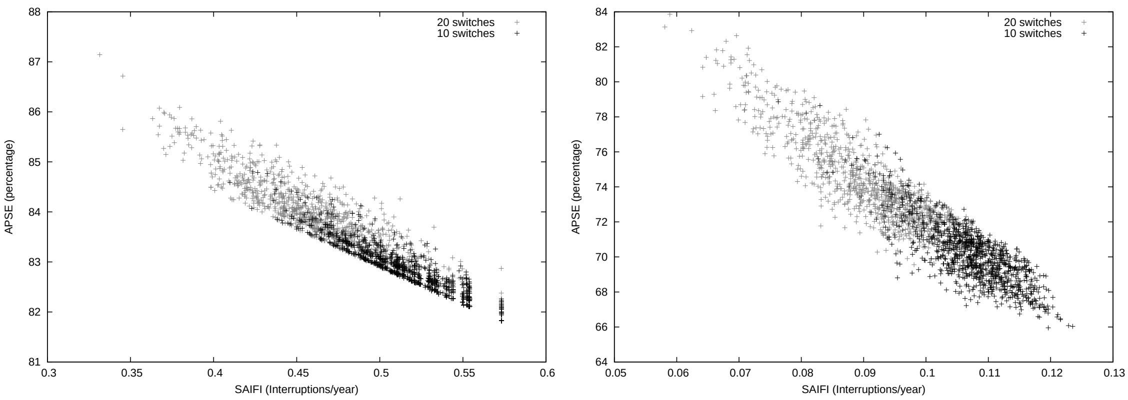
a. Instance B4, correlation –1.00. b. Instance R6, correlation –0.92.

Figure 6.1: Comparison of APSE and EENS.



a. Instance B4, correlation 0.94. b. Instance R6, correlation 1.00.

Figure 6.2: Comparison of SAIFI and EENS.



a. Instance B4, correlation –0.95. b. Instance R6, correlation –0.92.

Figure 6.3: Comparison of SAIFI and APSE

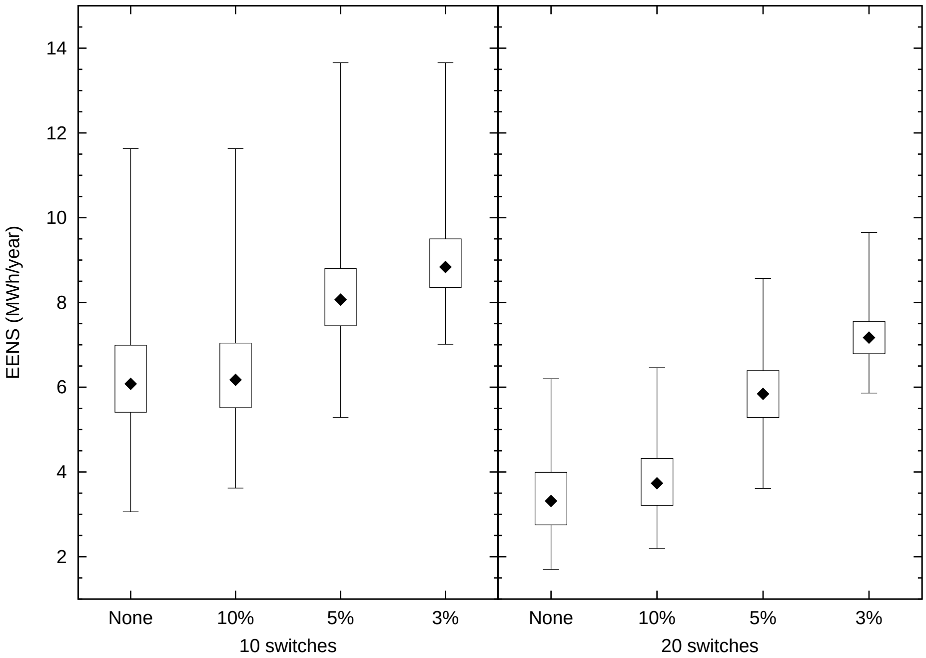
based on the quantity of affected (non served) load points. Thus, APSE varies in an approximately inverse proportion than SAIFI and EENS, while SAIFI and EENS vary almost proportionally. Observe that an optimal solution for one measure is not necessarily the optimal solution for others, and that the best of two different solutions depends of the measure selected for the comparison.

Since all measures all strongly correlated, we can expect a similar solution quality using any of them. We selected EENS for the rest of our tests because it is the most common in literature, because it presents a good correlation with the other measures, and because it depends only on the non-supplied load, failure rate and outage time for its estimation.

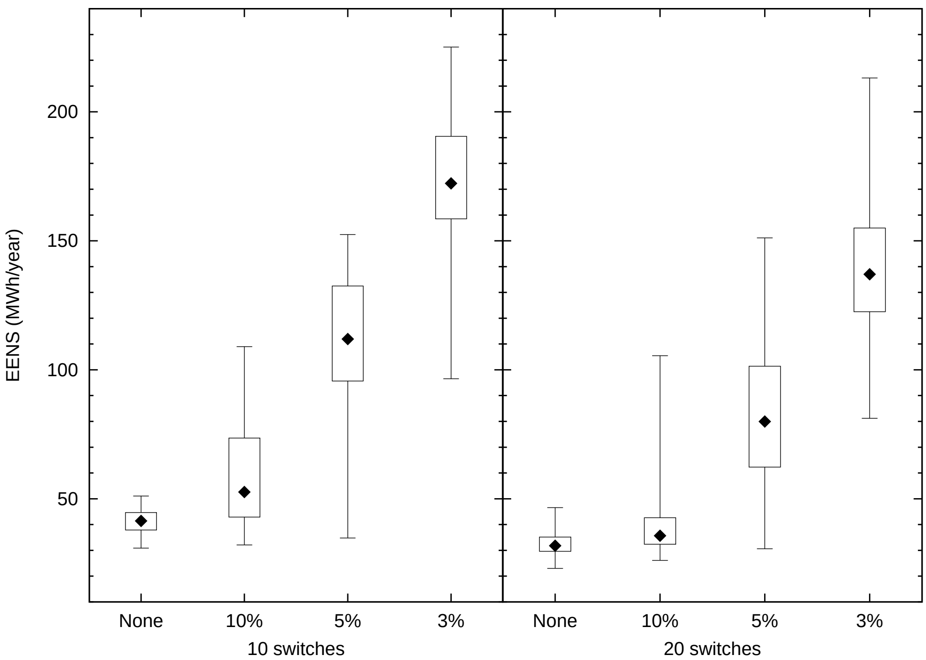
6.2 Influence of electrical constraints

Another question is how the electrical constraints influence the reliability estimation. Variations on capacities of lines or substations produce similar results than variations on voltage drop limits, but they are usually constants in real networks. Voltage drop limits vary for different countries depending on regulations. Therefore, we study the influence of different voltage drop limit.

Figures 6.4a and 6.4b compare the influence of the voltage drop limit in EENS estimations for instances R3 and AR. 1000 solutions were generated randomly and evaluated with four different voltage drop limits (3%, 5%, 10% and unlimited). The graphs present a box and whisker diagram, representing the minimum, the lower quartile, the median, the upper quartile, and the maximum value of the resulting EENS. We observe that EENS increases when the voltage drop limit becomes tighter. For instance AR, the EENS of almost all random solutions is below 50 MWh/year when evaluated without voltage drop limit, but three quartiles goes over 60 MWh/year with a voltage drop limit of five percent and over 120 MWh/year with three percent.



a. Instance R3.



b. Instance AR.

Figure 6.4: EENS quartiles for 1000 random solutions.

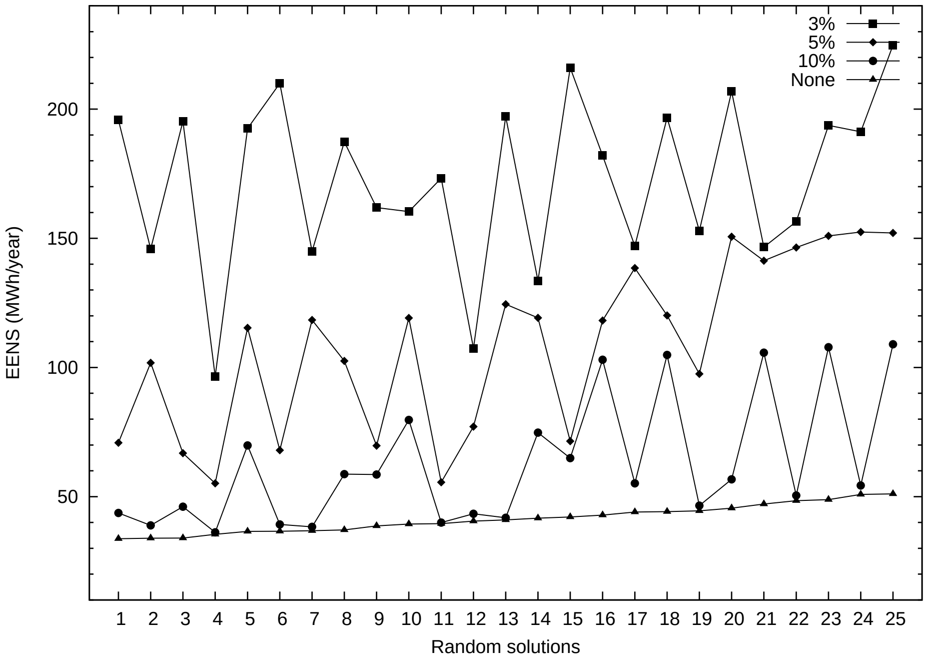
Figure 6.5 presents 25 randomly generated solutions for instance AR. It shows how EENS changes with different voltage drop limits. Note that an apparently good solution when considering one voltage drop limit can be a bad solution with a different voltage drop limit. Thus, the result of comparing the EENS of two solutions depends on the used voltage drop limit. This shows that a reliability estimation for the switch allocation problem without considering the service restoration problem and electrical constraints can mislead us into solutions theoretically good, but worse when considering electrical constraints in reality.

6.3 Performance improvements in reliability estimation

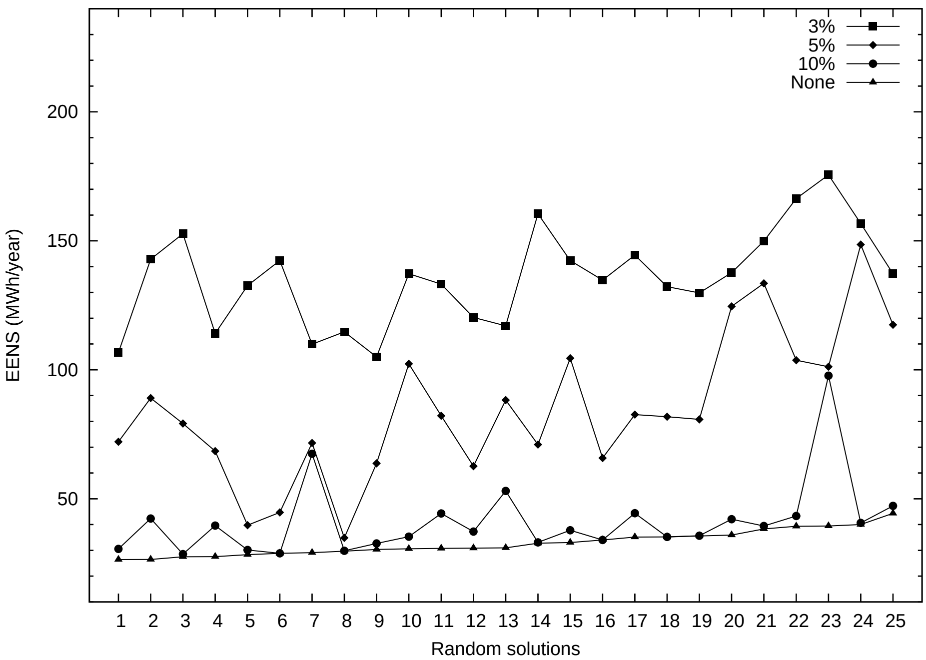
In this work, two improvements are proposed for the original reliability estimation of Costa et al. (2007, 2008). The first one is the reliability estimation by sectors described in Section 3.2, and the second is the optimistic service restoration described in Section 3.4. The reliability upper and lower bounds can be estimated by sectors, but the optimistic service restoration can only be used within the upper bound estimation.

Table 6.1 shows the total execution time for the reliability estimation of 1000 solutions generated randomly for various instances with 10 and 20 switches. It presents times for executions with and without the improvements. We observe that the improvements significantly reduce the processing time. For the upper bound, reliability estimation by sectors saves more time than the optimistic restoration improvement, but they save even more time when used together. The last column (Sp) shows the speedup of the combined two improvements. Depending of the instance, they can reduce runtime up to a factor of 100.

In order to explain the speedups of Table 6.1, we study the number of failure simulations and the number of electrical feasibility evaluations in Table 6.2. These quantities are presented in thousands. We observe a reduction in the number of failure simulations when we compare the reliability estimations by lines and



a. 10 switches.



b. 20 switches.

Figure 6.5: EENS for 25 random solutions of instance AR.

Table 6.1: Running time in reliability estimation.

Inst.	Sw.	Lower Bound		Upper Bound				
		No Impr	Impr 1	No Impr	Impr 1	Impr 2	Impr 1&2	Sp%.
BU	10	0.04	0.03	0.64	0.36	0.16	0.08	8.0
BR	10	0.04	0.02	0.67	0.36	0.16	0.09	7.4
R3	10	0.04	0.02	0.54	0.28	0.37	0.2	2.7
B4	10	0.16	0.04	4.64	1.36	1.24	0.38	12.2
AU	10	0.33	0.08	5.06	1.22	2.23	0.54	9.4
AR	10	0.32	0.08	5.2	1.26	2.26	0.56	9.3
R4	10	0.34	0.09	8.18	2.51	2.79	0.87	9.4
R5	10	0.81	0.14	17.81	2.97	6.92	1.17	15.2
R6	10	1.78	0.17	35.54	3.53	10.94	1.1	32.3
R7	10	28.85	0.81	815.32	22.13	291.32	8.11	100.5
BU	20	0.05	0.04	1.76	1.39	0.42	0.31	5.7
BR	20	0.05	0.03	1.78	1.41	0.46	0.33	5.4
R3	20	0.04	0.03	1.34	1.1	0.92	0.76	1.8
B4	20	0.19	0.08	7.93	3.62	1.82	0.84	9.4
AU	20	0.34	0.13	8.93	3.35	3.47	1.31	6.8
AR	20	0.36	0.12	9.14	3.41	3.5	1.32	6.9
R4	20	0.32	0.14	12.47	5.52	3.73	1.66	7.5
R5	20	0.85	0.22	29.14	7.63	9.55	2.55	11.4
R6	20	1.77	0.32	74.17	13.04	23.57	4.16	17.8
R7	20	28.72	1.36	1283.74	58.8	343.58	15.9	80.7

Impr 1: Improvement of reliability estimation by sectors instead of lines.
Impr 2: Improvement of optimistic service restoration.

Table 6.2: Thousands of failure simulations and feasibility evaluations.

Inst.	Sw.	Failure simulations		Electrical feasibility evaluations				
		No Impr	Impr 1	No Impr	Impr 1	Impr 2	Impr 1&2	Sp%
BU	10	32.0	14.3	159.8	79.5	43.2	18.6	8.6
BR	10	32.0	14.3	162.2	80.5	44.4	19.2	8.5
R3	10	32.0	15.1	155.7	78.4	148.6	77.9	2.0
B4	10	67.0	19.4	798.9	223.3	264.7	75.6	10.6
AU	10	80.0	18.8	1107.8	255.6	920.7	216.8	5.1
AR	10	80.0	18.8	1104.8	255.7	924.4	217.6	5.1
R4	10	83.0	25.0	1770.0	522.6	1257.6	375.7	4.7
R5	10	135.0	22.3	2324.3	374.4	1578.2	258.7	9.0
R6	10	201.0	19.2	2256.6	212.0	872.3	83.0	27.2
R7	10	873.0	23.4	13906.4	365.4	6659.0	177.8	78.2
BU	20	32.0	24.2	433.3	336.5	103.1	74.5	5.8
BR	20	32.0	24.3	440.1	341.4	106.5	77.4	5.7
R3	20	32.0	25.4	394.9	319.1	362.4	295.4	1.3
B4	20	67.0	31.0	1484.6	674.2	409.8	185.9	8.0
AU	20	80.0	30.1	1979.2	737.0	1454.9	546.2	3.6
AR	20	80.0	29.9	1995.4	738.5	1465.1	546.7	3.6
R4	20	83.0	37.3	2767.2	1225.4	1669.3	742.9	3.7
R5	20	135.0	35.8	3859.8	1007.4	2219.4	584.1	6.6
R6	20	201.0	34.8	4692.9	816.7	1777.2	309.5	15.2
R7	20	873.0	39.6	22449.4	1012.8	7762.3	351.3	63.9

Impr 1: Improvement of reliability estimation by sectors instead of lines.
Impr 2: Improvement of optimistic service restoration.

by sectors. This is because the number of sectors is associated to the number of installed switches, which is always smaller than the number of lines. We also compare the number of electrical feasibility tests. In average, the optimistic service restoration halves the number of feasibility tests, and combined with the estimation by sectors, it is able to reduce it up to 78 times in the largest instance.

6.4 Greedy construction and tabu search results

The first results of our research are a comparison of tabu search and a greedy construction. For the tabu search, we use a tabu tenure of 10 iterations and the stop criterion is 100 iterations. We run tests for installing 5, 10, 15 or 20 switches in every instance. Each test case is executed ten times.

In order to study the performance of tabu search, we separately inform time and EENS for the first local minimum found by the tabu search. Before this point, the behavior of tabu search is the same of a first improvement local search.

Table 6.3 and Table 6.4 show results for lower and upper bound reliability. We show the EENS and the execution time for the greedy construction. For the tabu search, we report further average values for the initial random solution (column 1S), the first local minimum solution (column LS), and the final solution (column TS). We also present the best solution found within the ten repetitions (column Min.). Best values found are in boldface.

Upper and lower bound present similar results. The first seven instances (except R5, R6 and R7) present a difference in EENS between LS and TS, i.e., tabu search escapes the first local minimum and finds better solutions. In average, the time difference between LS and TS shows that first local minimum solution is found early during the tabu search, within the first 15% of the running time. The first local minimum improves a random generated solution by half. The difference between GR and LS shows that local search finds better solutions than a greedy

Table 6.3: Lower bounds by greedy construction and tabu search.

Inst.	Sw.	Greedy Constr.		Tabu Search					
		EENS	Time	EENS				Time	
		GR	GR	1S	LS	TS	Min.	LS	TS
BU	5	2220	0.0	4305±622	2167±85	**2022**±0	**2022**	0.0±0.0	0.2±0.0
	10	1214	0.0	3182±477	1141±95	**1082**±0	**1082**	0.0±0.0	0.5±0.0
	15	903	0.0	2094±467	827±30	**810**±0	**810**	0.0±0.0	0.9±0.1
	20	793	0.0	1660±446	719±30	**700**±0	**700**	0.1±0.0	1.1±0.1
BR	5	1998	0.0	4002±664	1888±74	**1809**±0	**1809**	0.0±0.0	0.2±0.0
	10	1207	0.0	3119±736	1060±75	**1024**±0	**1024**	0.0±0.0	0.5±0.0
	15	1023	0.0	2122±588	806±22	786±8	**784**	0.0±0.0	0.9±0.1
	20	780	0.0	1494±391	682±15	669±1	**667**	0.1±0.0	1.1±0.1
R3	5	4405	0.0	7757±800	4307±158	**4078**±0	**4078**	0.0±0.0	0.2±0.0
	10	2390	0.0	6473±1425	2180±4	**2176**±0	**2176**	0.0±0.0	0.4±0.0
	15	1824	0.0	4155±873	1776±18	**1759**±0	**1759**	0.1±0.0	0.8±0.1
	20	1560	0.0	3522±979	1527±6	**1522**±0	**1522**	0.1±0.0	1.0±0.1
B4	5	16720	0.0	21393±561	**16625**±0	**16625**±0	**16625**	0.0±0.0	0.9±0.1
	10	12384	0.0	20767±903	**12329**±0	**12329**±0	**12329**	0.1±0.0	2.5±0.2
	15	10776	0.1	20150±1044	10687±116	10664±145	**10496**	0.2±0.1	4.2±0.4
	20	9576	0.1	18817±1793	8992±403	8992±403	**8742**	0.3±0.1	6.4±0.7
AU	5	34993	0.0	49065±3112	33452±654	**32673**±0	**32673**	0.1±0.1	2.5±0.1
	10	26547	0.1	43964±2658	24746±397	**24040**±0	**24040**	0.5±0.1	5.4±0.5
	15	21024	0.1	37393±2972	19492±436	18940±67	**18889**	1.1±0.2	8.0±1.6
	20	17497	0.2	33007±5609	17014±415	16160±150	**16058**	1.9±0.5	10.9±4.0
AR	5	34502	0.0	46932±2770	32774±688	**32253**±0	**32253**	0.2±0.0	2.3±0.2
	10	26598	0.1	40725±4340	24933±686	24039±153	**23962**	0.5±0.1	5.2±0.9
	15	20590	0.1	36764±6024	19653±385	19118±185	**18740**	1.3±0.3	8.1±2.2
	20	17594	0.2	33681±4580	16486±462	15939±115	**15803**	2.5±0.5	8.3±2.5
R4	5	3664	0.0	5042±259	3601±89	**3573**±0	**3573**	0.1±0.0	2.0±0.1
	10	**2861**	0.1	4751±296	2902±43	2866±15	**2861**	0.4±0.2	4.2±0.5
	15	2380	0.1	4284±297	2306±25	2306±25	**2234**	0.9±0.3	4.6±1.1
	20	2059	0.2	3902±427	1959±75	1959±74	**1853**	1.3±0.3	6.3±2.8
R5	5	**9333**	0.1	14636±632	**9333**±0	**9333**±0	**9333**	0.6±0.0	4.7±0.7
	10	7356	0.2	13647±795	7481±139	7218±105	**7115**	1.5±0.2	6.7±3.2
	15	5722	0.3	13348±702	8243±2001	8243±2001	**5367**	2.0±1.5	2.2±1.8
	20	4879	0.4	11047±820	6537±1629	6537±1630	**4743**	2.9±1.5	2.9±1.7
R6	5	5123	0.1	7370±476	4546±172	4430±240	**3747**	1.4±0.4	6.7±1.4
	10	2300	0.3	6850±782	3280±820	3268±840	**1926**	3.3±1.2	3.7±1.8
	15	1477	0.5	5998±1229	3578±1262	3578±1262	**1362**	4.0±1.9	5.3±6.2
	20	**1109**	0.8	4782±1135	2523±1256	2523±1256	1134	3.9±1.7	3.9±1.7
R7	5	**772798**	1.9	1135195±61331	923310±80379	923310±80379	781443	35.3±9.4	35.3±9.4
	10	**571365**	4.9	1109368±31758	932279±68753	932279±68753	801416	13.6±7.7	13.6±7.7
	15	**430193**	9.1	1072579±83982	1004829±116733	1004829±116733	776778	7.7±4.4	7.7±4.4
	20	**337110**	14.9	993414±95883	959461±110712	959461±110712	735060	7.9±3.2	7.9±3.2

Table 6.4: Upper bounds by greedy construction and tabu search.

Inst.	Sw.	Greedy Constr.		Tabu Search					
		EENS	Time	EENS				Time	
		GR	GR	1S	LS	TS	Min.	LS	TS
BU	5	**2220**	0.0	4305± 732	2225± 14	2220± 1	**2220**	0.0± 0.0	0.5± 0.0
	10	1316	0.0	3080± 579	1351± 67	1292± 15	**1285**	0.1± 0.0	3.0± 0.3
	15	1029	0.1	2436± 386	984± 47	**926**± 0	**926**	0.5± 0.1	7.3± 0.8
	20	940	0.1	2003± 470	799± 28	**782**± 0	**782**	0.7± 0.2	12.3± 1.1
BR	5	1998	0.0	4033± 680	1960± 8	**1957**± 0	**1957**	0.0± 0.0	0.6± 0.0
	10	1215	0.0	2963± 361	1184± 6	1175± 2	**1174**	0.1± 0.0	3.3± 0.3
	15	985	0.0	2424± 640	883± 4	**881**± 0	**881**	0.5± 0.1	8.9± 0.8
	20	795	0.1	1600± 316	765± 8	760± 1	**759**	0.7± 0.3	13.7± 1.1
R3	5	8918	0.0	10254± 1838	7056± 769	**6469**± 0	**6469**	0.0± 0.0	0.9± 0.1
	10	8820	0.0	8687± 1233	4196± 357	**4026**± 0	**4026**	0.4± 0.1	4.6± 0.7
	15	4738	0.0	6888± 889	**3399**± 0	**3399**± 0	**3399**	0.9± 0.1	13.6± 1.7
	20	3761	0.1	5888± 681	**3278**± 0	**3278**± 0	**3278**	1.5± 0.3	21.5± 1.8
B4	5	17893	0.1	21349± 698	17546± 171	**17058**± 0	**17058**	0.2± 0.0	8.7± 0.5
	10	**14235**	0.3	20513± 799	14319± 160	**14235**± 0	**14235**	1.6± 0.4	34.0± 2.7
	15	12830	0.6	19527± 1192	12807± 113	12777± 111	**12565**	4.7± 1.0	83.1± 5.7
	20	11707	1.2	19078± 1228	11515± 173	11469± 213	**11262**	8.2± 1.6	172.3±25.1
AU	5	39124	0.2	137791± 32555	39212± 381	**38288**± 0	**38288**	1.4± 0.2	9.9± 2.6
	10	30863	0.6	123246± 31508	33369± 7377	33141± 7476	**30074**	6.7± 1.6	24.2±19.8
	15	26639	1.3	90829± 16858	26709± 517	26430± 644	**25897**	17.0± 3.2	53.4±32.9
	20	23948	2.4	83654± 32956	24235± 1850	23910± 1936	**22834**	38.4± 9.4	126.0±75.6
AR	5	38451	0.2	112802± 24044	37332± 201	**37269**± 0	**37269**	1.4± 0.3	12.3± 4.0
	10	30999	0.6	118963± 31830	34264± 6846	34033± 6992	**29811**	6.9± 2.1	14.7±14.7
	15	26471	1.4	91521± 23902	29091± 8084	28692± 8229	**25266**	18.2± 4.3	52.2±35.6
	20	24063	2.4	78907± 21342	25619± 4483	25458± 4602	**21833**	35.4±10.8	54.5±59.0
R4	5	**4354**	0.2	9279± 1608	**4354**± 0	**4354**± 0	**4354**	0.4± 0.1	20.3± 1.0
	10	3848	0.7	8654± 2396	3801± 25	**3770**± 0	**3770**	8.7± 1.1	62.6± 3.8
	15	3483	1.4	7728± 2033	3450± 23	3436± 2	**3435**	14.1± 2.7	124.3±23.9
	20	3307	2.4	6748± 2352	3216± 95	3099± 58	**3063**	21.5±10.7	176.1±37.3
R5	5	**11389**	0.5	26433± 4270	**11389**± 0	**11389**± 0	**11389**	2.7± 0.4	35.5± 9.1
	10	9760	1.6	26247± 4534	10274± 1610	10274± 1610	**9688**	17.3± 6.8	33.9±22.7
	15	**8553**	3.6	26396± 3136	11883± 5279	11883± 5279	8619	24.8±18.4	28.2±27.4
	20	**7885**	6.9	21582± 5230	11310± 2489	11310± 2489	8394	28.1± 5.7	28.1± 5.7
R6	5	5123	0.4	7418± 404	5027± 5	**4564**± 0	**4564**	5.2± 1.5	34.6± 7.4
	10	3540	1.7	6881± 568	3924± 466	3924± 466	**3251**	27.7± 7.8	27.7± 7.8
	15	**2508**	4.5	6354± 426	4048± 526	4048± 526	3147	29.0±13.8	29.0±13.8
	20	**1925**	9.4	6016± 590	3815± 669	3815± 669	2322	40.3±18.6	40.3±18.6
R7	5	**803930**	18.9	1154688± 40662	962477± 28307	962477± 28307	900464	286.5±70.6	286.5±70.6
	10	**583187**	66.8	1103935± 44685	924418± 78245	924418± 78245	832114	115.4±57.8	115.4±57.8
	15	**478566**	159.1	1039929±112788	953747±130213	953747±130213	673430	107.9±83.1	107.9±83.1
	20	**384885**	316.0	966043± 68177	901696± 68442	901696± 68442	803063	122.3±59.1	122.3±59.1

construction. The difference between LS and TS shows that tabu search finds better solutions than greedy construction and local search. But the time increases 7 times between LS and TS. The small standard deviations in column TS indicate that the best known solution is found almost always (except for AU and AR).

In the large instances (R5, R6 and R7), the results are not as good as in the smaller ones. In average, there are no big differences in EENS between LS and TS, mainly with more than 5 switches. This indicates that tabu search does not overcome the first local minimum. The lack of time difference between LS and TS indicates that a first local minimum is not found by the tabu search within the 100 iterations, i.e., 100 iterations are not enough for the large instances. A large number of iterations would be too expensive for these large instances. The worst results have been obtained for the largest instance R7, that did not find a local minimum even for 5 switches.

In general, tabu search is effective with small instances, but not with large instances. Tabu search finds better results than local search, but with a higher cost. Averages of tabu search and local search overcome the greedy solution for small instances. Tabu search, either does not reach, or does not overcome the first local minimum in 100 iterations for the large instances.

The small difference in EENS and big difference in time between LS and TS for the small instances indicated that a multi-start method like GRASP could obtain better results than trying to escape local minima.

6.5 GRASP Results

This section presents the results obtained with GRASP for ten instances and 5, 10, 15 and 20 switches. Each experiment was executed ten times for every test case, except R6 which was executed five times and R7 which was executed just once. The tested GRASP is formed by a semi-greedy construction and a first improvement local search. The stop criterion is ten GRASP iterations and the local search runs until there is no improvement in the neighbourhood. We tested four α values (0.25, 0.5, 0.75 and 1.0) with four instances (BR, BU, AR and AU). The different α values produced similar results with different execution times. The best times were obtained with $\alpha = 0.25$ and $\alpha = 0.5$. This indicates that the best α value is between 0.25 and 0.5. These two α values were used with the rest of instances (R3, B4, R4, R5, R6 and R7). Tables 6.5 and 6.6 present the average EENS, the average execution time and the best solution found with GRASP (column Min.). The new bounds found by GRASP are marked with "✓" and the bounds of GRASP are marked with "$\times$" when they did not reach the best bound of tabu search.

The difference in average EENS of results with $\alpha = 0.25$ and $\alpha = 0.5$ is less than four percent and the difference in execution time is less than ten percent. Thus, the results of GRASP with different α values are statistically about the same.

The average results of GRASP and tabu search are the same for the instances BR, BU, and B3. The results show some differences for the instances B4, AR, AU, and R4, where GRASP finds two better, twelve equal and three worse results than tabu search when using the lower bound, and five better, seven equal and four worse when using the upper bound. The execution time of GRASP for the small instances (BR, BU, R3, B4 and R4) is about twice than tabu search, except instances AR and AU that show a big difference in running time, up to 20 times slower than tabu search.

GRASP is able to find good results in the large instances (R5, R6, and R7), where tabu search was not. The greedy construction time is the same than semi-greedy construction time. For this reasons, GRASP execution time of the instances R5, R6 and R7 can not be compared.

Table 6.5: Lower bounds by GRASP.

Inst.	Sw.	$\alpha = 0.25$			$\alpha = 0.5$		
		EENS	Min.	Time	EENS	Min.	Time
BU	5	2161± 95	**2022**	0.1± 0.0	2154± 85	2056	0.1± 0.0
	10	**1082**± 0	**1082**	0.4± 0.0	**1082**± 0	**1082**	0.4± 0.0
	15	811± 1	**810**	0.7± 0.0	811± 1	**810**	0.8± 0.1
	20	703± 7	**700**	1.0± 0.1	703± 7	**700**	0.9± 0.0
BR	5	1928± 63	**1809**	0.1± 0.0	1858± 69	**1809**	0.1± 0.0
	10	**1024**± 0	**1024**	0.5± 0.0	**1024**± 0	**1024**	0.5± 0.0
	15	785± 3	**784**	0.8± 0.0	**784**± 0	**784**	0.8± 0.1
	20	668± 1	**667**	1.0± 0.0	668± 1	**667**	1.0± 0.0
R3	5	4209±169	**4078**	0.1± 0.0	4209±169	**4078**	0.1± 0.0
	10	2177± 2	**2176**	0.4± 0.0	**2176**± 0	**2176**	0.4± 0.0
	15	1760± 2	**1759**	0.8± 0.1	**1759**± 0	**1759**	0.8± 0.1
	20	**1522**± 0	**1522**	0.9± 0.0	**1522**± 0	**1522**	0.8± 0.0
B4	5	**16625**± 0	**16625**	0.7± 0.0	**16625**± 0	**16625**	0.6± 0.1
	10	**12329**± 0	**12329**	2.1± 0.1	**12329**± 0	**12329**	2.1± 0.1
	15	**10531**± 0	**10531**×	3.9± 0.1	10539± 16	**10531**×	3.9± 0.2
	20	**8742**± 0	**8742**	5.3± 0.4	**8742**± 0	**8742**	5.5± 0.4
AU	5	32700± 88	**32673**	2.0± 0.2	**32673**± 0	**32673**	2.0± 0.2
	10	24305±288	**24040**	7.6± 0.6	24068± 73	**24040**	7.3± 0.4
	15	18940± 60	**18889**	15.1± 1.1	18940± 91	**18889**	13.6± 0.9
	20	16411±144	16244	25.2± 1.5	16216± 79	**16058**	22.5± 1.4
AR	5	32315± 45	**32253**	2.0± 0.1	32285± 44	**32253**	2.1± 0.2
	10	24266±352	24057	7.2± 0.4	24096±147	**23992**×	6.9± 0.6
	15	18843±105	**18740**	15.6± 0.9	18848±100	**18740**	14.6± 0.7
	20	16151±134	15985	27.5± 1.8	15991± 95	**15889**×	25.3± 1.7
R4	5	**3573**± 0	**3573**	1.8± 0.1	**3573**± 0	**3573**	1.9± 0.1
	10	2860± 2	**2858**✓	7.2± 0.4	2859± 2	**2858**✓	6.9± 0.4
	15	2305± 25	**2234**	13.0± 0.7	2266± 40	**2234**	12.7± 0.6
	20	1882± 61	**1853**	20.0± 1.0	**1853**± 0	**1853**	19.5± 0.9
R5	5	**9333**± 0	**9333**	8.0± 0.3	**9333**± 0	**9333**	7.7± 0.3
	10	7178± 48	**7115**	22.7± 1.5	7178± 48	**7115**	22.3± 1.3
	15	5396± 63	**5367**	52.4± 4.4	5387± 63	**5367**	50.7± 3.8
	20	4368± 8	**4364**✓	99.6± 5.3	4370± 10	**4364**✓	96.2± 7.1
R6	5	3876±170	**3747**	18.5± 1.2	4133±278	3875	17.8± 1.5
	10	1945± 42	**1926**	85.2± 5.9	**1926**± 0	**1926**	77.2± 7.1
	15	1363± 20	**1353**✓	187.8± 2.5	1356± 4	**1353**✓	160.9± 6.6
	20	1056± 1	**1054**✓	256.6±44.3	1057± 5	**1054**✓	225.0±17.4
R7	5		726830	1309.5		**722688**✓	1276.2
	10		**529067**✓	2551.2		**529067**✓	2297.0
	15		**382953**✓	4551.0		382959	4666.0
	20		**310626**✓	9648.0		**310626**✓	10504.0

✓: new best bound, ×: previous best bound not reached.

Table 6.6: Upper bounds by GRASP.

Inst.	Sw.	$\alpha = 0.25$ EENS	$\alpha = 0.25$ Min.	$\alpha = 0.25$ Time	$\alpha = 0.5$ EENS	$\alpha = 0.5$ Min.	$\alpha = 0.5$ Time
BU	5	**2220**± 0	**2220**	0.6± 0.0	**2220**± 0	**2220**	0.5± 0.0
	10	1287± 5	**1285**	3.8± 0.4	**1285**± 0	**1285**	4.0± 0.4
	15	**926**± 0	**926**	12.7± 0.8	**926**± 0	**926**	11.0± 0.7
	20	783± 3	**782**	22.4± 1.1	784± 3	**782**	21.1± 1.8
BR	5	**1957**± 0	**1957**	0.5± 0.0	**1957**± 0	**1957**	0.5± 0.0
	10	1179± 2	**1174**	4.2± 0.4	1180± 2	1175	4.2± 0.3
	15	**881**± 0	**881**	12.7± 0.8	**881**± 0	**881**	13.2± 1.3
	20	**759**± 0	**759**	22.7± 1.4	759± 1	**759**	21.3± 0.9
R3	5	6481± 24	**6469**	0.8± 0.1	6487± 41	**6469**	0.8± 0.1
	10	**4026**± 0	**4026**	10.3± 0.5	**4026**± 0	**4026**	9.9± 1.0
	15	**3399**± 0	**3399**	24.7± 1.1	**3399**± 0	**3399**	24.4± 1.6
	20	**3278**± 0	**3278**	43.3± 1.7	**3278**± 0	**3278**	41.4± 2.2
B4	5	17221±262	**17058**	8.0± 0.4	**17058**± 0	**17058**	8.0± 0.4
	10	**14244**± 0	**14244**×	41.0± 2.0	**14244**± 0	**14244**×	41.1± 1.4
	15	12685± 80	**12599**×	114.3± 4.4	12656± 97	**12599**×	124.5± 6.9
	20	11283± 44	**11262**	219.1± 10.8	**11262**± 0	**11262**	222.4± 5.7
AU	5	**38660**± 0	**38660**×	35.0± 1.6	**38660**± 0	**38660**×	33.0± 1.7
	10	30306±165	**30074**	177.5± 8.8	30296±159	**30074**	173.7± 6.8
	15	25651±249	**25108**✓	455.1± 24.0	25711±180	25572	421.6± 24.0
	20	23005±260	**22694**✓	909.5± 50.1	22945±203	22741	870.3± 45.2
AR	5	**37269**± 0	**37269**	41.6± 1.1	**37269**± 0	**37269**	39.4± 1.9
	10	30208±298	**29811**	212.8± 12.7	30133±286	**29811**	204.6± 12.1
	15	26008±171	25761	509.7± 23.9	25926±253	**25554**×	470.6± 32.8
	20	22400±485	**21317**✓	1108.7± 57.3	22373±345	21506	1037.3± 79.4
R4	5	**4354**± 0	**4354**	20.7± 0.5	**4354**± 0	**4354**	21.4± 0.4
	10	3763± 14	3736	166.7± 5.4	3752± 19	**3726**✓	170.7± 7.3
	15	3370± 70	**3281**✓	315.2± 23.0	3343± 66	**3281**✓	314.0± 21.2
	20	3063± 1	**3063**	390.1± 18.6	3063± 0	**3063**	390.5± 22.5
R5	5	**11389**± 0	**11389**	85.5± 1.8	**11389**± 0	**11389**	81.4± 2.6
	10	9673± 10	**9667**✓	478.0± 21.8	**9667**± 0	**9667**✓	460.1± 30.9
	15	8467± 22	**8406**✓	1506.4± 61.4	8474± 0	8474	1547.1± 56.9
	20	7706± 78	**7575**✓	2270.5±193.3	7612± 78	**7575**✓	2256.1±195.5
R6	5	4865±223	**4564**	93.3± 2.7	5025± 0	5025	94.9± 4.6
	10	3147± 12	**3139**✓	776.8± 26.4	3149± 15	**3139**✓	756.1± 42.7
	15	2277± 46	**2245**✓	1997.4± 58.7	**2245**± 0	**2245**✓	2041.9± 58.0
	20	1794± 0	1794	6675.2±721.6	1794± 1	**1793**✓	7134.6±505.6
R7	5		**769558**✓	15895.0		**769558**✓	16326.0
	10		572979	43459.0		**569354**✓	44575.0
	15		**435809**✓	126237.0		**435809**✓	119309.0
	20		361420	291757.0		**356233**✓	313825.0

✓: new best bound, ×: previous best bound not reached.

6.6 Sample algorithms vs. conventional algorithms

Various questions emerged with the previous tests. The first question is if the greedy and semi-greedy constructions produce a worthy initial solution in exchange of the processing time required to build it, or if random initial solutions are better for the subsequent local search processes. Another doubt is which strategy is the best for local search: first best improvement or best improvement. Another question is if another strategy with a restricted neighbourhood can improve the previous construction and local search algorithms.

We tried to answer these questions with the comparison of some construction and local search algorithms, including the sample construction and the sample local search described in Section 4.4.

We compare the sample construction algorithm with a random construction and a semi-greedy construction, and compare the sample local search with the first improvement and the best improvement strategies. We combined constructive and local search methods in nine tests, as shown in Table 6.7.

Table 6.7: Combinations of sample and construction algorithms for tests.

		Construction algorithm		
		Semi-greedy	Random	Sample
Local search	Sample	SGr-Spl	Rnd-Spl	Spl-Spl
	First improvement	SGr-FI	Rnd-FI	Spl-FI
	Best improvement	SGr-BI	Rnd-BI	Spl-BI

Preliminary tests of sample algorithms with $\beta = 5\%, 10\%, 20\%$ on instance B4 presented slightly better results for $\beta = 10\%$. In these tests, sample construction and sample local search use $\beta = 10\%$ and semi-greedy construction uses $\alpha = 0.5$. As the sample local search is not an exhaustive search, its stop criterion is ten iterations without improvement.

We selected two instances for this test, one small and one large. B4 was selected because its greedy solutions are also local minimum (for 15 and 20 switches), and R6 because is large enough to confirm results without causing excessively long time process. We run tests allocating 15 and 20 switches. We repeat each experiment 1000 times for instance B4, and 100 times for instance R6. We present the results for instance B4 in Table 6.8 and Figure 6.6, and for the instance R6 in Table 6.9 and Figure 6.7.

Tables 6.8 and 6.9 show the average EENS and the number of reliability estimations used to generate the initial solutions with the construction algorithms. We use the number of reliability estimations instead of the average time in the comparisons, because it is a more precise measure than the uncomparable and very small times in the construction of the initial solutions. For the final solutions obtained with the local search methods, the tables present the average EENS, the average number of reliability estimations, and the average running time. The tables also present the best solution found by each combination within all the repetitions (column Min.). The last columns compare the number of final solutions that reach (column =GR) or overcome (column <GR) the corresponding greedy solution.

Figures 6.6 and 6.7 compare the average EENS achieved with the required number of reliability estimations. Four points show the average result of the construction algorithms (random, semi-greedy, sample and greedy). Three lines start from each point (except greedy), they outline the average performance of first improvement, best improvement and sample local search strategies.

First, we analyze the initial solutions of the construction algorithms. Solutions created by the semi-greedy algorithm are better than random solutions in average by 2000 KWh/year and 1100 KWh/year for instances B4 and R6, but the execution time and the number of reliability estimations to build the initial solution increases significantly. A random solution requires only one reliability estimation, while the semi-greedy and the greedy algorithms require more than 900 estimations for B4 and more than 3000 estimations for R6. Greedy construction generates always the best initial solution at the same cost than semi-greedy, but this solution is

Table 6.8: Comparison of construction and local search algorithms, instance B4.

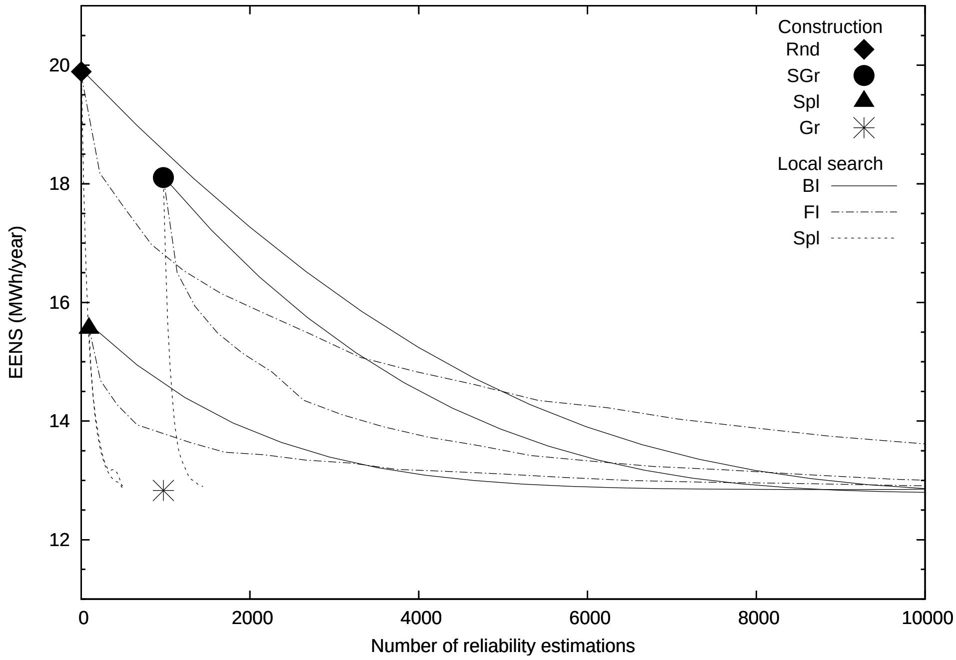
	Algorithm	Initial solution		Final solution					
		EENS	N.Est.	EENS	Min.	N.Est.	Time	=GR	<GR
15 switches	SGr-Spl	18042±1058	975	13452±681	12599	1208± 77	0.6± 0.1	1	14
	SGr-FI	18151±1027	975	12782±107	**12565**	14542±3638	18.9± 4.9	247	432
	SGr-BI	18124±1032	975	12789± 91	**12565**	10523±1290	12.5± 1.7	631	269
	Rnd-Spl	19867±1017	1	13482±689	12618	257± 75	0.3± 0.1	0	13
	Rnd-FI	19899±1005	1	12770±117	**12565**	21053±4340	27.2± 5.1	123	518
	Rnd-BI	19908± 981	1	12793± 97	**12565**	11331±1286	13.5± 1.5	466	305
	Spl-Spl	15585±1176	91	13418±637	12624	262± 71	0.3± 0.1	1	11
	Spl-FI	15537±1166	91	12840± 70	**12565**	9843±3167	13.4± 4.2	408	111
	Spl-BI	15556±1164	91	12841± 50	**12565**	7360±1255	9.7± 1.7	638	43
20 switches	SGr-Spl	16822±1226	1250	11923±446	**11262**	1710± 133	1.6± 0.4	7	268
	SGr-FI	16835±1186	1250	11509±175	**11262**	19264±5073	42.5±11.4	401	599
	SGr-BI	16872±1211	1250	11505±189	**11262**	14075±1726	28.5± 4.1	442	558
	Rnd-Spl	19060±1108	1	11947±419	**11262**	526± 128	1.0± 0.3	9	228
	Rnd-FI	19009±1126	1	11524±158	**11262**	28804±6030	63.3±13.2	373	627
	Rnd-BI	19000±1176	1	11535±179	**11262**	16444±1948	33.1± 3.7	488	512
	Spl-Spl	14056±1129	116	12031±441	11308	479± 133	0.8± 0.3	6	96
	Spl-FI	14080±1180	116	11642±134	**11262**	12354±4064	28.4± 8.8	797	203
	Spl-BI	14027±1137	116	11641±141	**11262**	9431±1617	20.9± 3.4	811	189

Greedy solution (GR) for instance B4 with 15 and 20 switches is 12830 and 11707 KWh/year respectively.

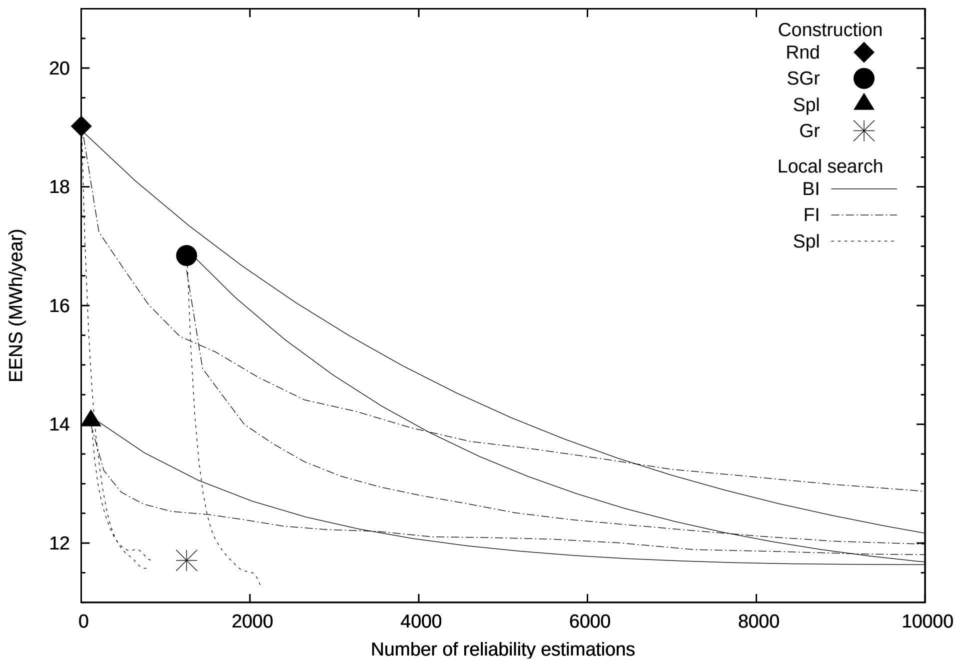
Table 6.9: Comparison of construction and local search algorithms, instance R6.

	Algorithm	Initial solution		Final solution				
		EENS	N.Est.	EENS	Min.	N.Est.	Time	<GR
15 switches	SGr-Spl	5329±585	3135	2717±217	2354	4164± 338	6.5± 1.4	19
	SGr-FI	5293±586	3135	2320± 86	**2236**	119571±30943	377.0± 97.5	96
	SGr-BI	5380±621	3135	2315± 78	**2236**	51891± 6817	157.8± 25.6	97
	Rnd-Spl	6394±568	1	2677±213	2327	1157± 343	3.8± 1.3	25
	Rnd-FI	6367±550	1	2322± 84	**2236**	174435±46355	568.3±159.3	95
	Rnd-BI	6466±538	1	2346± 94	**2236**	51319± 6150	157.9± 25.2	95
	Spl-Spl	3177±289	307	2672±198	2306	981± 292	2.5± 1.0	21
	Spl-FI	3157±281	307	2369±100	**2236**	44586±19044	137.0± 62.6	89
	Spl-BI	3102±271	307	2345± 69	**2236**	34319± 6415	103.6± 24.4	99
20 switches	SGr-Spl	4668±593	4130	2011±143	1822	6580± 671	20.6± 4.9	32
	SGr-FI	4602±741	4130	1827± 55	**1793**	202289±50048	1159.1±313.8	90
	SGr-BI	4735±635	4130	1853± 86	**1793**	81435±10229	431.8± 71.8	79
	Rnd-Spl	5975±607	1	1997±143	1814	2704± 755	15.6± 4.9	38
	Rnd-FI	5921±578	1	1848± 82	**1793**	306406±72093	1810.3±479.1	84
	Rnd-BI	5814±585	1	1868± 88	**1793**	86244±11295	460.5± 75.6	73
	Spl-Spl	2541±215	404	1998±150	1800	2173± 721	10.9± 4.2	38
	Spl-FI	2550±214	404	1836± 69	**1793**	88137±33966	512.8±201.8	84
	Spl-BI	2572±234	404	1843± 68	**1793**	56855±10334	320.8± 61.1	85

Greedy solution (GR) for instance R6 with 15 and 20 switches is 2508 and 1925 KWh/year respectively.

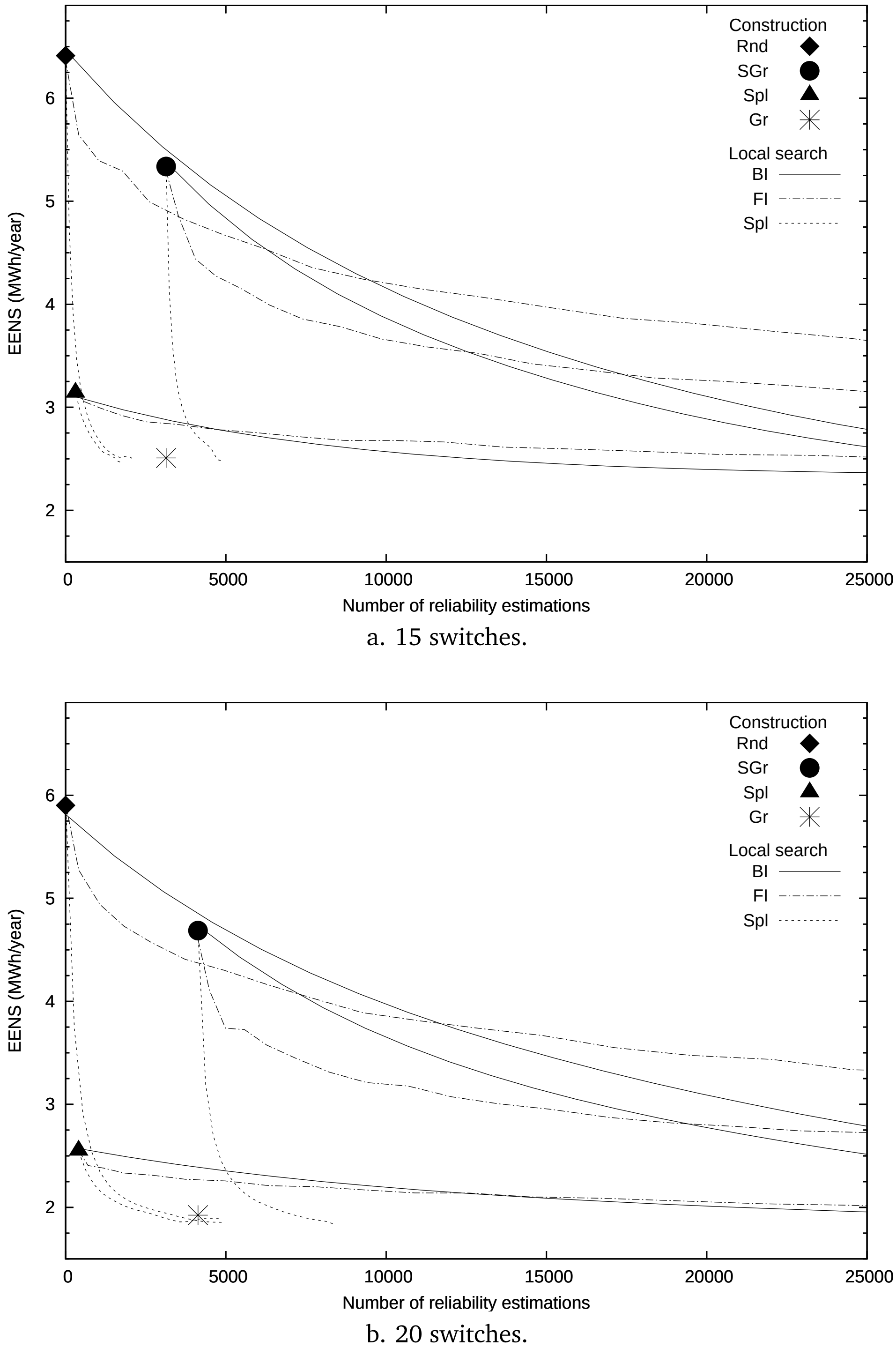


a. 15 switches.



b. 20 switches.

Figure 6.6: Average performance for instance B4.



a. 15 switches.

b. 20 switches.

Figure 6.7: Average performance for instance R6.

usually close to (or is itself) a local minimum, that is undesirable for a multi-start procedure. Solutions created by the sample algorithm are better than random solutions in average by 4600 KWh/year and 3300 KWh/year for instances B4 and R6, and they require less than 120 estimations for B4 and less than 410 estimations for R6. Thus, sample algorithm creates better solutions than semi-greedy algorithm, and in less than ten percent of the corresponding time. The good cost/benefit of the sample construction algorithm can be seen in the graphs by its proximity to the origin, i.e., low EENS and low number of reliability estimations. Contrarily, semi-greedy construction generates the worst solutions considering the high number of reliability estimations.

Now, we analyze the local search algorithms. The average final solutions of first and best improvement are very close, and they yield the best result for all construction algorithms. The biggest difference between first and best improvement is 26 MWh/year. It is found with semi-greedy construction for R6 with 20 switches, and it is half of the smallest standard deviation. The difference between first and best improvement is in their execution time and their performance over time. First improvement spends more time than best improvement. The average final solutions of sample local search are worse than first and best improvement. The difference of sample local search with other search strategies is less than 700 KWh/year and 400 KWh/year for instance R4 and B6, respectively. Moreover, the sample local search was able to find the best solution for instance B4 with 20 switches. The time that sample local search spent is very small, about half the time of the greedy or semi-greedy construction alone.

In Figures 6.6 and 6.7, the graphs of the three local search strategies show the same behavior for all the test cases, independently of the constructive algorithms. Figure 6.8 shows isolatedly the performance of the three local search strategies for the instance B4 with 15 switches and the sample constructive algorithm. We observe that first improvement progresses quickly in the beginning, but best improvement becomes better after some iterations. Best improvement has an stable number of reliability estimations in each iteration along the whole search. First improvement takes any solution better than current and the number of estimations varies with the iterations. This is an advantage in early iterations because first improvement finds easily better solutions, but becomes a disvantage in the late iterations because first improvement restarts the local search with any small improvement when the number of reliability estimations is almost the same than best improvement. Thus, first improvement finally spends more time than best improvement in average.

Another observation is that sample local search has a very fast progress compared with the other strategies. The number of reliability estimations of sample local search is constant in each iteration like best improvement, but is 100 times smaller because the neighbourhood is restricted randomly to ten percent of switches and ten percent of free lines. The sample local search is not an exhaustive search in the neighbourhood, i.e., it does not guarantee to find the local minimum, but it finds good results fast.

Another important observation emerges from the comparison with the greedy

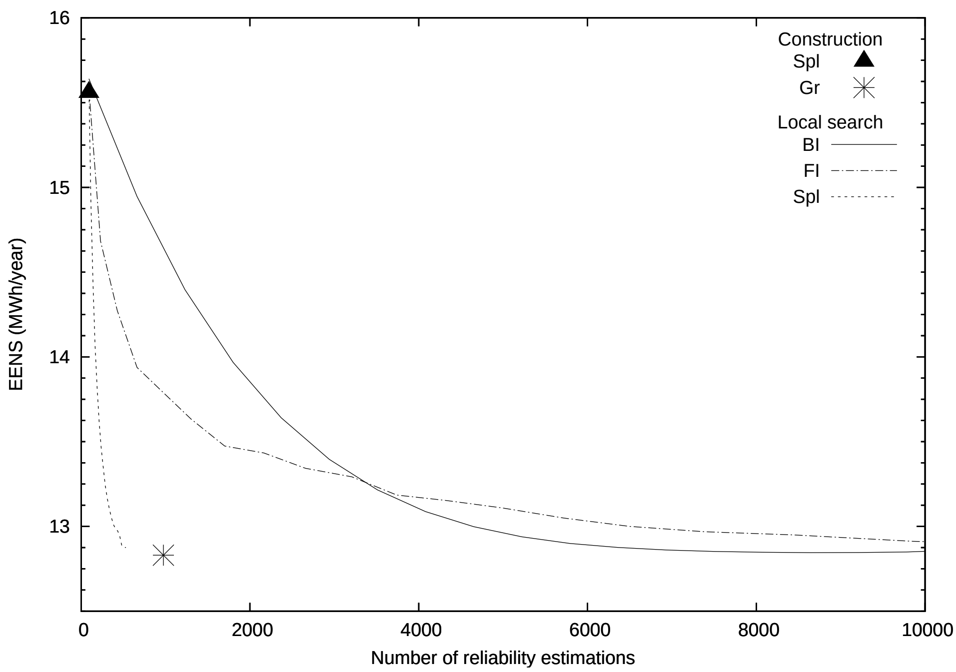


Figure 6.8: Comparison of local search methods on instance B4 with 15 switches.

solution. For instance B4, almost half of the final solutions of first and best improvement strategies are stuck in the greedy solution, in particular with best improvement or when the initial solution is built by the sample algorithm. Random construction with the first improvement local search is the combination that finds the biggest number of solutions that overcome the greedy solution. For instance R6, the local search algorithms do not get stuck in the greedy solution, and the first and best improvement overcome the greedy solution in 89 percent of the cases.

Finally, we analyze the combinations of construction and local search. If we consider each row of Tables 6.8 and 6.9 as one iterative local search, with 1000 and 100 iterations for instance B4 and R6, and each row with semi-greedy construction as one GRASP, we observe that iterative search processes with first and best improvement are effective to reach the best known upper bound for the test cases. But the number of iterations to obtain this results is very high, and the accumulated running time is 1000 and 100 times the shown average for instance B4 and R6, respectively.

A GRASP is as effective as an iterative local search with random initial solutions, but needs less time.

The best method for an iterative local search would be the combination of sample construction and best improvement local search, because it is the cheapest combination in terms of execution time that is able to find the best solution.

The most expensive combination is random construction with first improvement local search. This is the reason for tabu search to spend many early iterations without finding the first local minimum in large instances.

The cheapest method for an iterative local search would be the combination of sample construction and sample local search, its execution time is at least two times faster than a greedy or semi-greedy construction algorithm alone. This verifies that a restricted neighbourhood speeds the search up.

6.7 Iterated sample construction with path relinking

A question that emerged from the good performance of sample local search is if there exists a better method to restrict the neighbourhood than a random sample. We also wanted to take advantage of the fast sample construction algorithm. This lead us to study a process of path relinking between two solutions generated by sample construction.

This section presents results for the iterated sample construction with path relinking algorithm described in Section 4.5. The stop criterion is 50 iterations. The reconstruction algorithm rebuilds a completely new solution, forbidding the switch positions of the previous one, and using a sample of $\beta = 10\%$. The path relinking searches the whole path from the new solution to the best found solution. Test cases are seven instances with the installation of 5, 10, 15 and 20 switches. Each test case is executed ten times.

Table 6.10 shows lower and upper bound results. It presents the average EENS, the average time, and the best solution found in all ten repetitions (column Min.).

Iterated sample construction with path relinking is able to find three new lower bounds and nine new upper bounds (marked with "✓"). Two lower bounds and three upper bounds (marked with "$\times$") do not reach the best known bounds given by GRASP. The difference in the solutions that do not overcome GRASP is less than 0.8% of the best known bound. The average EENS of the iterated algorithm overcome the corresponding greedy solution almost always, except for the instance R5 with 5 switches, where the greedy solution seems to be the global minimum.

The iterated algorithm is faster than GRASP for small instances (B4, AU, AR, R4), except B4 and R4 with 20 switches. In the large instances (R5, R6, R7) iterated sample construction with path relinking is many times faster than GRASP. Taking into account that our tests run 10 iterations for GRASP and 50 iterations

Table 6.10: Bounds by iterated sample construction with path relinking.

		Lower bound			Upper bound		
Inst.	Sw.	EENS	Min.	Time	EENS	Min.	Time
B4	5	**16625**± 0	**16625**	0.2±0.0	**17058**± 0	**17058**	1.9± 0.1
	10	**12329**± 0	**12329**	1.2±0.0	**14235**± 0	**14235**	19.8± 0.5
	15	10503± 18	**10496**	4.5±0.1	12600± 40	**12565**	98.9± 3.7
	20	**8742**± 0	**8742**	13.6±0.3	11301± 51	**11262**	314.5± 12.0
AU	5	**32673**± 0	**32673**	0.3±0.0	38539± 294	**38288**	3.7± 0.1
	10	**24040**± 0	**24040**	1.8±0.1	30083± 29	**30074**	38.7± 1.1
	15	18907± 39	**18889**	6.8±0.2	25744± 339	**24969**✓	201.0± 8.6
	20	16118± 49	**16058**	20.7±0.7	22119± 365	**21627**✓	760.8± 25.8
AR	5	32289± 48	**32253**	0.3±0.0	37769± 370	**37269**	3.8± 0.1
	10	**23962**± 0	**23962**	1.9±0.1	**29811**± 0	**29811**	40.7± 2.0
	15	18769± 21	**18740**	6.9±0.3	25661± 279	**25101**✓	202.3± 10.5
	20	15864± 41	**15803**	21.9±0.6	21923± 433	**21317**	764.7± 32.3
R4	5	3581± 24	**3573**	0.3±0.0	**4354**± 0	**4354**	4.2± 0.1
	10	2860± 18	**2830**✓	1.9±0.1	3766± 39	**3697**✓	37.0± 0.5
	15	2243± 14	**2234**	7.1±0.2	3306± 53	**3281**	176.6± 2.6
	20	**1853**± 0	**1853**	20.9±0.3	3062± 3	**3056**✓	586.1± 20.7
R5	5	9428± 223	**9333**	0.5±0.0	11439± 159	**11389**	9.1± 0.2
	10	7147± 34	**7115**	3.0±0.1	9673± 19	**9667**	75.7± 2.4
	15	**5367**± 0	**5367**	11.4±0.3	8518± 42	**8474**×	388.2± 7.7
	20	4365± 1	**4364**	33.3±0.6	7695± 79	**7553**✓	1395.2± 41.3
R6	5	3671± 29	**3648**✓	0.7±0.0	4599± 71	**4564**	5.0± 0.1
	10	1927± 1	**1926**	4.5±0.1	3189± 11	**3159**×	53.0± 0.8
	15	1361± 9	**1353**	17.8±0.3	2249± 12	**2236**✓	328.6± 9.4
	20	1058± 2	**1054**	53.4±0.9	1797± 6	**1793**	1322.0± 30.4
R7	5	724898±2049	**722688**	10.1±0.1	770910±4256	**769558**	168.3± 1.7
	10	510966±2615	**507035**✓	37.0±0.2	570341±1707	**568837**✓	877.1± 14.2
	15	385240±1765	**383654**×	104.5±1.0	445403±6553	**436823**×	3717.3± 73.6
	20	311209± 572	**310717**×	257.7±1.3	360487±2628	**355951**✓	11897.7±320.4

✓: new best bound, ×: previous best bound not reached.

for iterated sample construction with path relinking, we conclude that iterated sample construction with path relinking is always faster than GRASP.

We observe that the running time of iterated sample construction with path relinking strongly depends on the number of switches. Indeed, even while the sample and the semi-greedy constructions have the same time complexity $O(|A|^2)$ for $|A|$ lines, sample construction executes in β percent of the time of semi-greedy construction. Furthermore, for solutions with s switches, iterated path relinking has time complexity $O(s^3)$, while GRASP has time complexity $O(|A| \cdot s^2)$. On the one hand, path relinking searches in a neighbourhood restricted by the lines with switches in the other solution, on the other hand, GRASP searches in the whole neighbourhood of free lines. This is the reason for the reduction of the running time with the new iterated sample construction with path relinking.

With the small differences in the results with GRASP and the big reduction of running time, we conclude that iterated sample construction with path relinking is the best approach proposed in this dissertation for solving the switch allocation problem.

7 CONCLUDING REMARKS AND FUTURE RESEARCH

In this dissertation, we explained the switch allocation problem, the service reconfiguration problem, and the reasons for considering the second as a subproblem of the first one. We described the network reliability, its measures and methods to estimate its boundaries. We adapted and proposed instances for our tests. We proposed and implemented algorithms to estimate the reliability and to solve the joint problem. We showed the results obtained by various experiments executed during our research.

Our experiments studied different issues such as the comparison of different reliability measures, the influence of voltage drop limit as electrical constraint, the improvements proposed for reliability estimation, the comparison of greedy, semi-greedy, sample and random contruction algorithms, the comparison of sample, first and best improvement local search strategies, the comparison of tabu search, GRASP and iterated greedy construction with path relinking.

The major concluding remarks of our research are presented next. The sample construction generates the best solutions with a low cost for a future local search, and the worst performance comes from the semi-greedy construction. A restriction over the neighbourhood speeds up the local search, even when it is restricted by random samples. The tabu search method is not effective on large instances, while multi-start search like GRASP is effective in small and large instances. The best strategy proposed to solve the switch allocation problem is an iterative sample construction with path relinking. It finds better solutions and in less time than GRASP and tabu search, and it is able to solve large instances fast.

7.1 Ideas for future research

Next, we present some ideas of tests and improvements for future research.

The combination of sample construction and sample local search (Spl-Spl) has a good performance, but it is not an exhaustive local search. An idea for future work within an iterative search is to execute a best improvement local search after the Spl-Spl combination to ensure that a local minimum is reached in each iteration.

The results found with tabu search in previous works (COSTA et al., 2008; BENAVIDES et al., 2009b) suggested that a greedy or a semi-greedy initial solution helps the tabu search to find better solutions quickly, but other results show that Spl-Spl combination is more efficient than greedy and semi-greedy construction algorithms (BENAVIDES et al., 2009a). Another idea is the use of Spl-Spl combination to seed the first initial solution for a tabu search, saving time to find the first local minimum.

There are some variations for the proposed iterated sample construction with path relinking that can be tested in the switch allocation problem and in other combinatorial optimization problems. For example, to forbid elements used in more than one previous solution, to create new solutions far from the last solutions. Or, to intersperse a complete neighbourhood exploration at some iterations, to ensure the local minimum.

We did many efforts to compare our approaches with other authors. We tested many standard measures, and we adapted various instances to be used with the measures. But unfortunately, we found no puublished works with a similar reliability estimation and complete instance data to compare with. With the inclusion of costs based measures and different kinds of devices, or changing to a multi-objective optimization function, the comparison with other authors might be possible.

An unexplored research area is use of exact algorithms such as branch-and-bound to find optimal solutions. This may be impossible for large instances with the consideration of all electrical constraints. The consideration of some electrical constraints such as capacities can result in a better aproximation of lower bound.

REFERENCES

ALLAN, R. N. et al. A reliability test system for educational purposes - basic distribution system data and results. *Power Systems, IEEE Transactions on*, v. 6, n. 2, p. 813–820, May 1991. ISSN 0885-8950.

AUGUGLIARO, A.; DUSONCHET, L.; SANSEVERINO, E. R. Multiobjective service restoration in distribution networks using an evolutionary approach and fuzzy sets. *International Journal of Electrical Power & Energy Systems*, v. 22, n. 2, p. 103 – 110, 2000. ISSN 0142-0615.

BARAN, M. E.; WU, F. F. Network reconfiguration in distribution systems for loss reduction and load balancing. *Power Delivery, IEEE Transactions on*, v. 4, n. 2, p. 1401–1407, Apr 1989. ISSN 0885-8977.

BARAN, M. E.; WU, F. F. Optimal sizing of capacitors placed on a radial distribution system. *Power Delivery, IEEE Transactions on*, v. 4, n. 1, p. 735–743, Jan 1989.

BENAVIDES, A. J. et al. Sample construction and local search algorithms for the switch allocation problem. In: *III Encontro Regional de Pesquisa Operacional da Região Sul (ERPOSul)*. Porto Alegre, Brazil: Sociedade Brasileira de Pesquisa Operacional (SOBRAPO), 2009.

BENAVIDES, A. J. et al. A comparison of tabu search and GRASP for the switch allocation problem. In: *XLI Simpósio Brasileiro de Pesquisa Operacional (SBPO)*.

Porto Seguro, Brazil: Sociedade Brasileira de Pesquisa Operacional (SOBRAPO), 2009.

BILLINTON, R.; JONNAVITHULA, S. Optimal switching device placement in radial distribution systems. *Power Delivery, IEEE Transactions on*, v. 11, n. 3, p. 1646–1651, Jul 1996. ISSN 0885-8977.

BILLINTON, R.; JONNAVITHULA, S. A test system for teaching overall power system reliability assessment. *Power Systems, IEEE Transactions on*, v. 11, n. 4, p. 1670–1676, Nov 1996.

CARRENO, E. M.; ROMERO, R.; PADILHA-FELTRIN, A. An efficient codification to solve distribution network reconfiguration for loss reduction problem. *Power Systems, IEEE Transactions on*, v. 23, n. 4, p. 1542–1551, Nov. 2008. ISSN 0885-8950.

CARVALHO, P. M. S.; FERREIRA, L. A. F. M.; SILVA, A. J. C. da. A decomposition approach to optimal remote controlled switch allocation in distribution systems. *Power Delivery, IEEE Transactions on*, v. 20, n. 2, p. 1031–1036, April 2005. ISSN 0885-8977.

CELLI, G.; PILO, F. Optimal sectionalizing switches allocation in distribution networks. *Power Delivery, IEEE Transactions on*, v. 14, n. 3, p. 1167–1172, Jul 1999.

CHEN, C.-S. et al. Optimal placement of line switches for distribution automation systems using immune algorithm. *Power Systems, IEEE Transactions on*, v. 21, n. 3, p. 1209–1217, Aug. 2006. ISSN 0885-8950.

CIVANLAR, S. et al. Distribution feeder reconfiguration for loss reduction. *Power Delivery, IEEE Transactions on*, v. 3, n. 3, p. 1217–1223, July 1988.

COSTA, A. M. et al. Alocação de chaves para melhoria da confiabilidade da rede de distribuição de energia elétrica. In: *XXXIX Simpósio Brasileiro de Pesquisa Operacional*. Fortaleza – Brazil: [s.n.], 2007.

COSTA, A. M. et al. Melhoria da confiabilidade da rede de distribuição de energia elétrica. In: *XL Simpósio Brasileiro de Pesquisa Operacional*. João Pessoa – Brazil: [s.n.], 2008.

DELBEM, A. C. B.; CARVALHO, A. C. P. L. F. de; BRETAS, N. G. Main chain representation for evolutionary algorithms applied to distribution system reconfiguration. *Power Systems, IEEE Transactions on*, v. 20, n. 1, p. 425–436, Feb. 2005. ISSN 1558-0679.

ELMAKIAS, D. *New Computational Methods in Power System Reliability*. [S.l.]: Springer Publishing Company, Incorporated, 2008. (Studies in Computational Intelligence, v. 111). ISBN 3540778101, 9783540778103.

EMINOGLU, U.; HOCAOGLU, M. H. A new power flow method for radial distribution systems including voltage dependent load models. *Electric Power Systems Research*, v. 76, n. 1-3, p. 106 – 114, 2005. ISSN 0378-7796.

FALAGHI, H.; HAGHIFAM, M.-R.; SINGH, C. Ant colony optimization-based method for placement of sectionalizing switches in distribution networks using a fuzzy multiobjective approach. *Power Delivery, IEEE Transactions on*, v. 24, n. 1, p. 268–276, Jan. 2009. ISSN 0885-8977.

FEO, T.; RESENDE, M.; SMITH, S. A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, v. 42, p. 860–878, 1994.

FORTZ, B.; THORUP, M. Increasing internet capacity using local search. *Computational Optimization and Applications*, v. 29, p. 13–48, 2004.

GARCIA, V. J. *Metaheurísticas multiobjetivo para o problema de restauração do serviço em redes de distribuição de energia elétrica*. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação - UNICAMP, Campinas, Brazil, 2005.

GLOVER, F. Tabu Search–Part I. *INFORMS JOURNAL ON COMPUTING*, v. 1, n. 3, p. 190–206, 1989.

GLOVER, F. Tabu Search–Part II. *INFORMS JOURNAL ON COMPUTING*, v. 2, n. 1, p. 4–32, 1990.

GLOVER, F. Genetic algorithms and scatter search: unsuspected potentials. *Statistics and Computing*, v. 4, n. 2, p. 131 – 140, 1994. ISSN 0960-3174.

GOEL, L.; BILLINTON, R. Evaluation of interrupted energy assessment rates in distribution systems. *Power Delivery, IEEE Transactions on*, v. 6, n. 4, p. 1876–1882, Oct 1991. ISSN 0885-8977.

GUIMARÃES, M.; CASTRO, C. Reconfiguration of distribution systems for loss reduction using tabu search. In: *15th PSCC - Power Systems Computation Conférence*. [S.l.: s.n.], 2005. p. 10:2:1–10.

IEEE Std 1366. *IEEE Guide for Electric Power Distribution Reliability Indices*. 2003.

JEON, Y.-J. et al. An efficient simulated annealing algorithm for network reconfiguration in large-scale distribution systems. *Power Delivery, IEEE Transactions on*, v. 17, n. 4, p. 1070–1078, Oct 2002. ISSN 0885-8977.

KAVASSERI, R.; ABABEI, C. *REDS: REpository of Distribution Systems*. 2009. Disponível em: <http://venus.ece.ndsu.nodak.edu/ kavasseri/reds.html>.

KERSTING, W. H. Radial distribution test feeders. In: *Power Engineering Society Winter Meeting, 2001. IEEE*. [S.l.: s.n.], 2001. v. 2, p. 908–912.

KHOA, T. Q. D.; BINH, P. T. T. A hybrid ant colony search based reconfiguration of distribution network for loss reduction. In: . [S.l.: s.n.], 2006. p. 1–7.

LEVITIN, G.; MAZAL-TOV, S.; ELMAKIS, D. Genetic algorithm for optimal sectionalizing in radial distribution systems with alternate supply. *Electric Power Systems Research*, v. 35, p. 149–155, 1995.

LIN, C.-H. et al. Optimal switching placement for customer interruption cost minimization. In: *Power Engineering Society General Meeting, 2006. IEEE*. [S.l.: s.n.], 2006. p. 6 pp.–.

MORADI, A.; FOTUHI-FIRUZABAD, M. Optimal switch placement in distribution systems using trinary particle swarm optimization algorithm. *Power Delivery, IEEE Transactions on*, v. 23, n. 1, p. 271–279, Jan. 2008. ISSN 0885-8977.

PABLA, A. J. *Electric Power Distribution*. [S.l.]: McGraw Hill Professional, 2004.

PRANSINI, A. J. *Guide to Electrical Power Distribution Systems*. [S.l.]: The Fairmont Press, 2005.

RESENDE, M. G. C.; RIBEIRO, C. C. Greedy randomized adaptive search procedures. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). *Handbook of Metaheuristics*. Netherlands: Kluwer Academic Publishers, 2003. p. 219–249.

RESENDE, M. G. C.; WERNECK, R. F. A hybrid heuristic for the p-median problem. *Journal of Heuristics*, Kluwer Academic Publishers, Hingham, MA, USA, v. 10, n. 1, p. 59–88, 2004. ISSN 1381-1231.

RUIZ, R.; STÜTZLE, T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, v. 177, n. 3, p. 2033 – 2049, 2007. ISSN 0377-2217.

SANTANDER, L. G. et al. Minimal loss reconfiguration based on simulated annealing meta-heuristic. In: *CONIELECOMP '05: Proceedings of the 15th International Conference on Electronics, Communications and Computers*. Washington, DC, USA: IEEE Computer Society, 2005. p. 95–99. ISBN 0-7695-2283-1.

SILVA, L. G. W. da et al. Optimised placement of control and protective devices in electric distribution systems through reactive tabu search algorithm. *Electric Power Systems Research*, v. 78, n. 3, p. 372 – 381, 2008. ISSN 0378-7796.

SILVA, L. G. W. da; PEREIRA, R. A. F.; MANTOVANI, J. R. S. Allocation of protective devices in distribution circuits using nonlinear programming models and genetic algorithms. *Electric Power Systems Research*, 2004.

SILVA, L. G. W. da; PEREIRA, R. A. F.; MANTOVANI, J. R. S. Optimized allocation of sectionalizing switches and control and protection devices for reliability indices improvement in distribution systems. In: *Transmission and Distribution Conference and Exposition: Latin America, 2004 IEEE/PES*. [S.l.: s.n.], 2004. p. 51–56.

STÜTZLE, T. *Applying Iterated Local Search to the Permutation Flow Shop Problem*. Darmstad, Germany, 1998.

SU, C.-T.; CHANG, C.-F.; CHIOU, J.-P. Distribution network reconfiguration for loss reduction by ant colony search algorithm. *Electric Power Systems Research*, v. 75, n. 2-3, p. 190 – 199, 2005. ISSN 0378-7796.

THAKUR, T.; JASWANTI. Study and characterization of power distribution network reconfiguration. *Transmission & Distribution Conference and Exposition: Latin America, 2006. TDC '06. IEEE/PES*, p. 1–6, Aug. 2006.

VILLASANTI, A.; BARAN, B.; GARDEL, P. Multiobjective allocation of remotely controlled switches in an electric distribution power system. In: *Transmission and Distribution Conference and Exposition: Latin America, 2008 IEEE/PES*. [S.l.: s.n.], 2008. p. 1–7.

WANG, C.; CHENG, H. Z. Optimization of network configuration in large distribution systems using plant growth simulation algorithm. In: *Power Systems, IEEE Transactions on*. [S.l.: s.n.], 2008. v. 23, n. 1, p. 119–126. ISSN 0885-8950.

WANG, C.; CHENG, H. Z.; YAO, L. Z. Reactive power optimization by plant growth simulation algorithm. In: *Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008. Third International Conference on*. [S.l.: s.n.], 2008. p. 771–774.

WAXMAN, B. M. Routing of multipoint connections. *Selected Areas in Communications, IEEE Journal on*, v. 6, n. 9, p. 1617–1622, 1988.

WU, W.-C.; TSAI, M.-S.; HSU, F.-Y. A new binary coding particle swarm optimization for feeder reconfiguration. In: *Intelligent Systems Applications to*

Power Systems, 2007. ISAP 2007. International Conference on. [S.l.: s.n.], 2007. p. 1–6.

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

ZHANG, C.; ZHANG, J.; GU, X. The application of hybrid genetic particle swarm optimization algorithm in the distribution network reconfigurations multi-objective optimization. In: *Natural Computation, 2007. ICNC 2007. Third International Conference on*. [S.l.: s.n.], 2007. p. 455–459.

ZHANG, D.; FU, Z.; ZHANG, L. An improved ts algorithm for loss-minimum reconfiguration in large-scale distribution systems. *Electric Power Systems Research*, v. 77, n. 5-6, p. 685 – 694, 2007. ISSN 0378-7796.

ZHANG, D. et al. Network reconfiguration in distribution systems using a modified ts algorithm. In: *MMACTE'05: Proceedings of the 7th WSEAS International Conference on Mathematical Methods and Computational Techniques In Electrical Engineering*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2005. p. 310–314. ISBN 999-2222-11-X.

APPENDIX A RESTAURAÇÃO DE SERVIÇO E ALOCAÇÃO DE CHAVES EM REDES DE DISTRIBUIÇÃO: LIMITES E ALGORITMOS

A melhora da confiabilidade em redes de distribuição de energia elétrica é um tema importante para as indústrias de fornecimento de eletricidade, devido aos regulamentos estritos em muitos países.

Depois de detectar a ocorrência de uma falha na rede, algumas chaves são usadas para isolar a falha, enquanto outras restauram a energia a alguns consumidores. A ótima seleção das chaves que serão abertas ou fechadas para restaurar a energia é conhecido como o problema de restauração de serviço. A instalação de chaves em posições estratégicas pode reduzir o tempo de parada, e assim melhorar a confiabilidade da rede. A seleção ótima de posições para instalar chaves é conhecido como o problema de alocação de chaves.

Os problemas de restauração de serviço e de alocação de chaves estão relacionados estreitamente. O Capítulo 2 desta dissertação descreve com mais detalhe estes dois problemas. Para isso, também apresenta as principais características das redes de distribuição de energia elétrica e um modelo da rede baseado em grafos.

Os sistemas de distribuição de energia elétrica devem entregar a energia, garantindo um nível de continuidade e qualidade. A habilidade do sistema para cumprir com este objetivo é chamado de confiabilidade. Existem muitas medidas para estimar a confiabilidade de uma rede. O Capítulo 3 descreve as medidas mais comuns na literatura. Também descreve dois métodos para estimar a confiabilidade de uma rede de distribuição, achando um limite superior e um limite inferior. A estimação do limite superior considera as restrições elétricas, enquanto a estimação do limite inferior as ignora. Os algoritmos descritos para a estimação da confiabilidade propõem duas melhoras, uma estimação por setores ao invés de linhas (Algoritmo 3.2) e uma heurística otimista para a estimação do limite superior (Seção 3.4).

O foco principal da pesquisa é criar métodos para resolver o problema de alocação de chaves, considerando o problema de restauração de serviço como um subproblema. O Capítulo 4 descreve os métodos propostos que são: busca tabu, procedimento de busca gulosa adaptativa aleatória (sigla em inglês: GRASP), e procedimento iterativo de construção por amostras com reconexão de caminhos. Também descreve diferentes métodos de construção (gulosa, semigulosa, aleatória e por amostras), e de busca local (por amostras, primeira melhoria e melhor melhoria).

Um problema para nossa pesquisa foi a falta de instâncias de redes elétricas para experimentar nossos algoritmos. As instâncias encontradas na literatura não estão descritas completamente, ou são redes privadas. O Capítulo 5 apresenta as instâncias usadas, e descreve os dados que foram completados em cada uma delas.

Os resultados são apresentados no Capítulo 6. A três seções iniciais deste capítulo apresentam estudos referentes à estimação de confiabilidade. Primeiro comparamos as medidas de confiabilidade, depois estudamos a influência das restrições elétricas com diferentes limites de queda de voltagem, e finalmente estudamos as duas melhoras para a estimação de confiabilidade que são descritas no Capítulo 3. Os resultados mostram que as medidas estão estreitamente correlacionadas, que a consideração das restrições elétricas nos leva a uma melhor estimação da confiabilidade, e que as duas melhoras nos métodos de estimação de confiabilidade podem acelerar a estimação em até 100 vezes.

As últimas quatro seções do Capítulo 6 apresentam os resultados dos algoritmos propostos para resolver o problema de alocação de chaves. A primeira comparação é feita entre a busca tabu e um algoritmo de construção gulosa, depois apresenta os resultados do GRASP. A Seção 6.6 estuda o benefício dos métodos de construção gulosa, semigulosa, aleatória e por amostras, e estuda o desempenho das estratégias de busca local por amostras, primeira melhoria e melhor melhoria. Finalmente os resultados que apresenta o procedimento iterativo de construção por amostras com reconexão de caminhos são os melhores.

O Capítulo 7 apresenta as conclusões da pesquisa e algumas idéias para trabalhos futuros. As principais conclusões da nossa pesquisa são apresentadas a seguir. O algoritmo de construção por amostras gera as melhores soluções com um custo baixo para uma futura busca local, enquanto a construção semigulosa tem o pior rendimento. A restrição na vizinhança acelera a busca local, ainda se a restrição é feita com amostras aleatórias. A busca tabu não é efetiva nas instâncias maiores, enquanto uma busca multi- inicio como GRASP é efetiva nas instâncias menores e maiores. A melhor estratégia proposta para resolver o problema de alocação de chaves é o procedimento iterativo de construção por amostras com reconexão de caminhos, pois consegue encontrar melhores soluções em menos tempo do que GRASP ou busca tabu, e encontra rapidamente soluções para as instâncias maiores.