Sample Algorithms in Multi-start Searches for the Switch Allocation Problem

Alexander J. Benavides, Luciana S. Buriol, and Marcus Ritt

Instituto de Informática, Universidade Federal do Rio Grande do Sul - UFRGS
{ajbenavides, buriol, mrpritt}@inf.ufrgs.br

Abstract. We study the problem of allocating switches in electrical distribution networks to improve their reliability. We present a sample construction algorithm and a sample local search for this problem. We compare these approaches with other consruction and local search strategies (and combinations of them). We present and comment experimental results, showing that sample approaches are inexpensive and find good quality solutions.

Keywords: local search, sample algorithms, switch allocation.

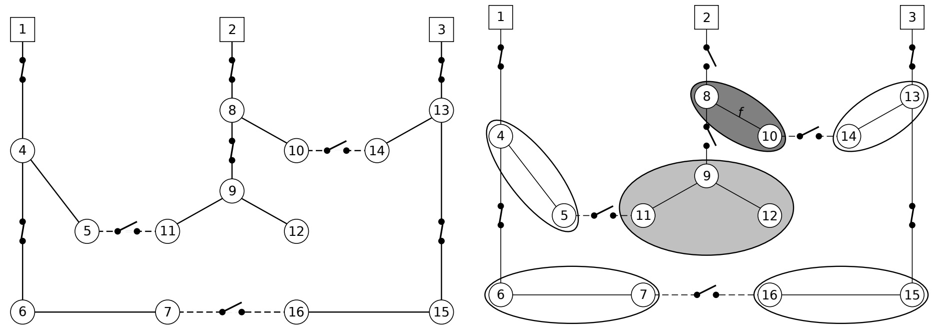
1 Introduction

According to Teng and Liu [19], most of the faults of an electrical power system take place in the distribution network. The most common method to improve the reliability of a distribution network is to add redundant lines with switches. Thus, in case of failures, the network topology is easily alterated and the affected areas are reduced. The installation of automatic switches all over the network is impracticable due to high costs. Because of that, companies must choose carefully the places where switches shall be installed. This combinatorial optimization problem is called the switch allocation problem.

The remainder of the paper is organized as follows. Section 2 explains the service restoration and the switch allocation problems. It also describes distribution networks using a graph model, and presents a method for network reliability estimation. Section 3 describes the construction algorithms (random, sample, greedy and semi-greedy) and the local search strategies (sample search, first and best improvement). Section 4 shows and discusses computational results. Concluding remarks are given in Section 5.

2 Description of the problems

Fig. 1 shows an example of an electric power distribution network taken from Civanlar et al. [6]. Fig. 1a shows the network under normal operation. Due to electrical constraints, the basic circuit of an operational distribution network has no cycles. The basic circuit is composed by substations (square nodes), consumers (round nodes), and feeder lines (black lines). Redundant feeder lines



a. Normal operating conditions. b. Sectors affected by a failure.

Fig. 1. Distribution Network Example

with switches (dotted lines) exist to reduce the time of blackouts in areas affected by failures. In normal conditions, switches of redundant lines are disconnected, opening the circuit, while switches on the basic circuit are connected, closing the circuit. Because of this, redundant lines are called normally open and basic circuit lines are called normally closed.

2.1 Graph model of distribution networks

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 $V(G) = N$ for the node set and $E(G) = A$ for the edge set of a graph or subgraph G. The presence of a switch on an edge $a \in A$ is indicated by a boolean value $B_a \in \{0, 1\}$. We represent a solution for the switch allocation problem with the set $A_B \subset A$ of lines themselves that are selected to install new switches ($A_B = \{a\}$, $B_a = 1$).

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. We define the *set of sectors* $\mathcal{SS} = \{\mathcal{S}(n)\}$ that contains all the disjoint sectors of nodes $n \in N$.

2.2 The service restoration problem

After a power failure is detected, the network topology must be modified to isolate the failure and to restore the energy supply by alternate feeder lines. The network reconfiguration is the process of opening and closing some switches in the feeder lines to change the topology. Fig. 1b shows an example of this process. Consider a failure in line $\{8, 10\}$. Without switches, the whole tree under substation 2 would be unattended. When the automatic switches on lines $\{2, 8\}$ and $\{8, 9\}$ are opened, the failure is isolated in sector $\mathcal{S}(8)$ (in dark gray). Then, sector $\mathcal{S}(9)$ (in light gray) becames without failure but is still unattended. When the automatic switch on line $\{5, 11\}$ is closed, the service is restored in sector $\mathcal{S}(9)$. The **service restoration problem** consists in choosing which switches must be opened or closed to minimize the unattended area after the isolation of a failure.

This problem has been studied extensively in the literature. Among the metaheuristics proposed to solve it are simulated annealing [11, 17], tabu search [24, 25], genetic algorithms [3, 9], ant colony optimization [13, 18], particle swarm optimization [22, 23], and plant growth simulation algorithm [20, 21].

This paper considers this problem as a subproblem of the switch allocation problem.

2.3 The switch allocation problem

According to Billinton and Jonnavithula [2], switches play a key role in the reliability of a power distribution system. 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 [14]. Automatic sectionalizing switches are able to diagnose a fault and eventually to automatically reschedule the respective configuration [5]. The installation of automatic switches in distribution systems allows a better and faster reconfiguration in case of failures, 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. Thus, **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, and it is subject to the number of available switches for allocation and to the electrical constraints.

This problem has been studied with different approaches, e.g. simulated annealing approach [2], divide-and-conquer approach [4], genetic algorithm [7], tabu search [8], three state particle swarm optimization [15], Ant Colony Optimization [10].

Many of the mentioned 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 reliability with statistical data or 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.

2.4 Network Reliability Estimation

We use expected energy non supplied (EENS) [10] to measure the network reliability. The EENS is calculated as

$$\text{EENS} = \sum_{f \in A_{nc}} \lambda_f r_f \sum_{n \in N_f} P_n \qquad \text{(MWh/year)}, \tag{1}$$

where A_{nc} is the set of feeder lines that can fail, N_f is the set of affected nodes by a failure f, r_f is the average outage time (in hours), λ_f is the average failure rate, and P_n is the energy normally consumed by node n.

Our approach takes into account the service restoration problem as a subproblem of the switch allocation problem. To estimate the reliability of a set of switch locations that represent a solution of the switch allocation problem, we must consider every possible failure, isolate it, maximize the restored area, and calculate the partial EENS.

We use the algorithm in Fig. 2 to estimate the reliability. This algorithm processes all the possible failures in lines of a sector $\mathcal{S}(n)$ together (lines 2-9), saving computing time. 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 its frontier is known for isolation. 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).

Note that frontier feeder lines (normally closed with switches) must still be processed separately (lines 10-17), because they are not within any sector. The algorithm in Fig. 2 follows a similar process for each line with a failure f. It determines and isolates the black sector $\mathcal{S}(f)$ easily with help of the defined sectors and frontiers (lines 12 and 13). Finally, the algorithm returns the total EENS.

We use an algorithm proposed by Benavides et al. [1] to simulate the service restoration after a failure and to calculate the affected area. This algorithm expands iteratively the supplied area and checks the feasibility of electrical constraints. The considered electrical constraints are lines and substation capacities and acceptable voltage drop. The electrical simulation is computationally very expensive, but electrical constraints are important to reflect a real approximation of the attended area.

3 Construction and local search algorithms

In this section we explain the construction and local search algorithms proposed to solve the switch allocation problem. Semi-greedy construction, and first and best improvement local searches were originally proposed by Benavides et al. [1].

Reliability Evaluation Algorithm

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

Output: Estimated reliability EENS.

1: EENS $\leftarrow 0$
2: **for** $\forall \mathcal{S}_i \in \mathcal{SS}$ **do** // Sectors
3: Simulate a failure f in $\mathcal{S}_i$
4: Assume the black area $\mathcal{S}(f) = \mathcal{S}_i$
5: Isolate the black area by opening the frontier switches $\mathcal{F}(\mathcal{S}(f)) = \mathcal{F}(\mathcal{S}_i)$
6: Determine affected nodes N_f with a service restoration algorithm
7: $\text{EENS}_f \leftarrow \sum_{a \in E(\mathcal{S}(f))} \lambda_a r_a \cdot \sum_{n \in N_f} P_n$
8: EENS $\leftarrow$ EENS + EENS_f
9: **end for**
10: **for** $\forall a = \{u, v\} \in A_{nc}, B_a = 1$ **do** // Frontier lines
11: Simulate a failure f in a
12: Assume the black area $\mathcal{S}(f) = \mathcal{S}(a) = \mathcal{S}(u) \cup \mathcal{S}(v) \cup (\{u, v\}, a)$
13: Isolate the black area by opening the frontier switches $\mathcal{F}(\mathcal{S}(f)) = (\mathcal{F}(\mathcal{S}(u)) \cup \mathcal{F}(\mathcal{S}(v))) \setminus \{a\}$
14: Determine affected nodes N_f with a service restoration algorithm
15: $\text{EENS}_f \leftarrow \lambda_f r_f \sum_{n \in N_f} P_n$
16: EENS $\leftarrow$ EENS + EENS_f
17: **end for**
18: **return** EENS

Fig. 2. Network reliability evaluation by sectors

Semi-greedy Construction Algorithm

Input: Distribution network $G = (N, A)$, number of switches k, α randomness.

Output: Set of lines with installed switches A_B.

1: $A_B \leftarrow \emptyset$
2: **while** $|A_B| < k$ **do**
3: *Candidate List* $\leftarrow A \setminus A_B$
4: Estimate reliability gain of all elements in *Candidate List*
5: *Restricted Candidate List* $\leftarrow \alpha$ portion of best elements in *Candidate List*
6: $a \leftarrow$ select randomly a switch location from *Restricted Candidate List*
7: $A_B \leftarrow A_B \cup \{a\}$
8: **end while**
9: **return** A_B

a. Semi-greedy.

Sample Construction Algorithm

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

Output: Set of lines with installed switches A_B.

1: $A_B \leftarrow \emptyset$
2: **while** $|A_B| < k$ **do**
3: *Candidate List* $\leftarrow A \setminus A_B$
4: *Sample Candidate List* $\leftarrow$ sample randomly β percent from *Candidate List*
5: Estimate reliability gain of all elements in *Sample Candidate List*
6: $a \leftarrow$ select the best switch location from *Sample Candidate List*
7: $A_B \leftarrow A_B \cup \{a\}$
8: **end while**
9: **return** A_B

b. Sample.

Fig. 3. Costruction algorithms.

First Improvement Local Search Algorithm

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: $A_B \leftarrow A_{Bbest}$
5: **for** $\forall a \in A_B$ **do** // With switch
6: **for** $\forall b \in A \setminus A_B$ **do** // Without switch
7: $A_{Bnew} \leftarrow (A_B \setminus \{a\}) \cup \{b\}$ // Move
8: Estimate reliability of A_{Bnew}
9: **if** $A_{Bnew} < A_{Bbest}$ **then**
10: $A_{Bbest} \leftarrow A_{Bnew}$
11: // Missing line in best improvement
exit for to line 3
12: **end if**
13: **end for**
14: **end for**
15: **end while**
16: **return** A_{Bbest}

a. First improvement.

Sample Local Search Algorithm

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

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: $A_B \leftarrow A_{Bbest}$
5: $A_{S1} \leftarrow$ sample randomly β lines from A_B
6: $A_{S2} \leftarrow$ sample randomly β lines from $A \setminus A_B$
7: **for** $\forall a \in A_{S1}$ **do** // With switch
8: **for** $\forall b \in A_{S2}$ **do** // Without switch
9: $A_{Bnew} \leftarrow (A_B \setminus \{a\}) \cup \{b\}$ // Move
10: Estimate reliability of A_{Bnew}
11: **if** $A_{Bnew} < A_{Bbest}$ **then**
12: $A_{Bbest} \leftarrow A_{Bnew}$
13: **end if**
14: **end for**
15: **end for**
16: **end while**
17: **return** A_{Bbest}

b. Sample.

Fig. 4. Local search algorithms.

3.1 Construction algorithms

We use four construction algorithms: random, sample, greedy and semi-greedy. **Random construction** selects n switches randomly and evaluates the resulting solution. **Greedy construction** builds a feasible solution element by element, evaluating all the elements to select the best each time. Semi-greedy and sample constructions (depicted in Fig. 3) also build a feasible solution one element at a time. Both use a reduced list of candidate elements to select one and add it to the solution. The difference lies in the way they create that small list. **Semi-greedy construction** (in Fig 3a) first evaluates every possible element. Then, a portion of α switches with the highest reliability is kept. And finally, one element is randomly picked from the restricted candidate list. ($\alpha = 0$ selects always the best element, and $\alpha = 1$ selects randomly between all the elements). **Sample construction** (in Fig. 3b) first selects randomly a portion of β switches. Then, it evaluates the sample candidate list to choose the best. ($\beta = 0\%$ corresponds to a random construction, and $\beta = 100\%$ corresponds to a greedy construction).

3.2 Local search algorithms

A local search algorithm iteratively replaces the current solution with a better neighbour. It starts from an initial solution created by a construction algorithm. And in this case, it searches in a neighbourhood defined by the relocation of one switch. We used three local search strategies: by sample, first improvement and best improvement.

First improvement local search is depicted in Fig 4a. It searches in the neighbourhood for an improvement of the current solution. When a better solution is found, it becomes the current solution for the next iteration. The search stops when there are no better solutions in the neighbourhood. Finally, the last found solution is returned. **Best improvement** searches through all the neighbourhood to select the best neighbour for the next iteration, while first improvement breaks the search out to the next iteration when it finds any better solution without evaluating the all neighbourhood. This difference lies in the **exit for** after the improvement test (line 11).

Sample local search is depicted in Fig 4b. It does not explore the whole neighbourhood, but a sample of β percent of lines with switches (line 5) and a β percent of places to move a switch (line 6). If the algorithm finds a better solution in the sample, it is taken for the next iteration. Finally, it returns the last solution. 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 another local search strategy after the sample local search, or intersperse an exhaustive neighbourhood search after a number of iterations.

4 Experiments

For our tests we used two instances. The small instance is known as RBTS Bus 4, introduced by Billinton and Jonnavithula [2]. The large instance is the sixth from the REpository of Distribution Systems (REDS) maintained by Kavasseri and Ababei [12]. Table 1 shows details for these instances.

To complete the necessary information, we followed the adaptation of part of the RBTS bus 6 by Falaghi et al. [10]. We assume an outage time $r = 2\ h$, a resistance $r = 0.257\ \Omega/km$, a reactance $x = 0.087\ \Omega/km$, a failure rate $\lambda = 0.065\ f/yr/km$, and a capacity $I_{MAX} = 500\ A$ for every line. The failure rate for REDS is calculated as $\lambda = 0.0696 * r$.

Table 1. Instances for the experiments.

	RBTS Bus 4	REDS 6^{th}
Network instances	B4	R6
Substations	3	3
Consumers	38	201
feeder lines	67	201
loop lines	5	15
Operation voltage (V)	11000	33600
Total power demand (kW)	24580	32437
Consumer power factor *	0.9	0.85
Consumer demand * (kW)	$[415, 1500]$	$[0, 1211]$
Line resistance (Ω)	$[0.1542, 0.2056]$	$[0.000, 0.187]$
Line reactance (Ω)	$[0.0522, 0.0696]$	$[0.000, 0.254]$
Line failure rates	$[0.039, 0.052]$	$[0.000, 0.013]$

* per load point.

We combined construction and local search methods as shown in Table 2. Sample construction and sample local search use $\beta = 10\%$ and semi-greedy construction has $\alpha = 0.5$. Stop criterion for sample local search is ten iterations without improvement. The SplBI combinations execute a best improvement local search after the sample local search, to guarantee a local minimum. We run tests to allocate 15 and 20 switches. We repeat each experiment 1000 times for B4, and 100 times for R6, except Gr-BI which is one time.

Table 2. Combinations of construction and local search algorithms for tests.

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

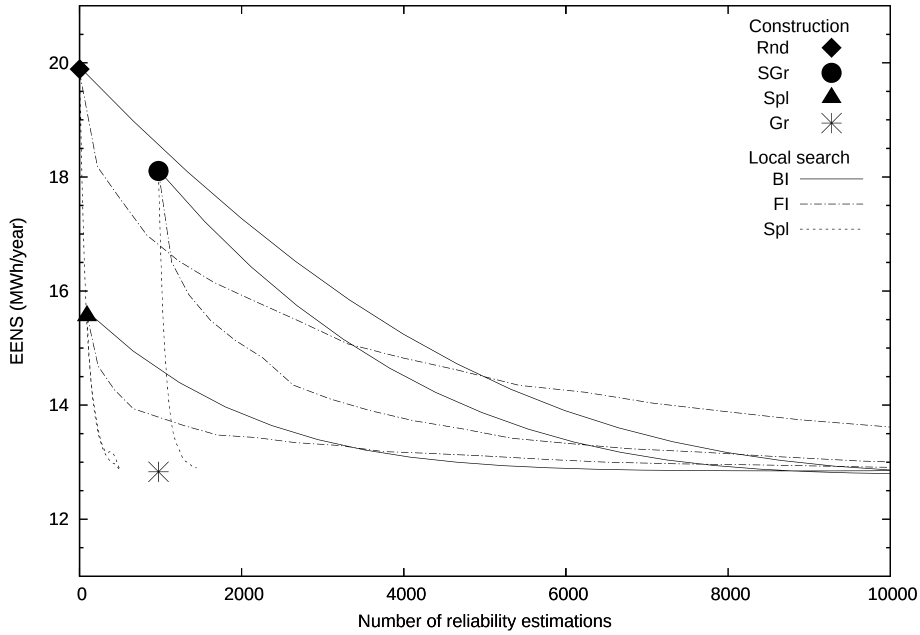
We present the results for instance B4 in Table 3 and Figure 5, and for the instance R6 in Table 4 and Figure 6. The tables show the average EENS and the number of reliability estimations used to generate the initial solutions with the

Table 3. Comparison of construction and local search algorithms, instance B4.

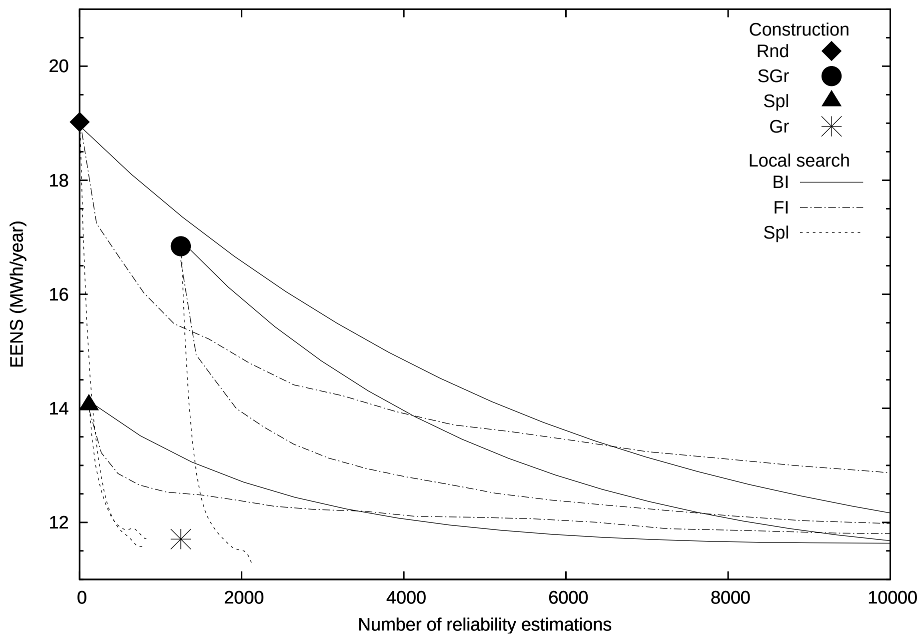
	Algorithm combination	Construction EENS	Construction N.Est.	Local search final solution EENS	Min.	N.Est.	Time	=Gr	<Gr
15 switches	Gr-BI	12830	975	12830		1830	1.7	1	0
	SGr-FI	18151±1027	975	12782±107	**12565**	14542±3638	18.9± 4.9	247	432
	SGr-BI	18124±1032	975	12789± 91	**12565**	10523±1290	12.5± 1.7	631	269
	SGr-Spl	18042±1058	975	13452±681	12599	1208± 77	0.6± 0.1	1	14
	SGr-SplBI	18159±1056	975	12811± 74	**12565**	6256±1353	7.5± 1.8	700	165
	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
	Rnd-Spl	19867±1017	1	13482±689	12618	257± 75	0.3± 0.1	0	13
	Rnd-SplBI	19923±1011	1	12819± 73	**12565**	5186±1290	7.0± 1.8	644	139
	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
	Spl-Spl	15585±1176	91	13418±637	12624	262± 71	0.3± 0.1	1	11
	Spl-SplBI	15534±1177	91	12835± 52	**12565**	5201±1300	7.0± 1.8	703	54
20 switches	Gr-BI	11707	1250	11707		2290	3.5	1	0
	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
	SGr-Spl	16822±1226	1250	11923±446	**11262**	1710± 133	1.6± 0.4	7	268
	SGr-SplBI	16796±1235	1250	11551±183	**11262**	7112±1477	14.5± 3.5	556	444
	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
	Rnd-Spl	19060±1108	1	11947±419	**11262**	526± 128	1.0± 0.3	9	228
	Rnd-SplBI	19042±1109	1	11568±177	**11262**	5902±1432	13.9± 3.4	598	402
	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
	Spl-Spl	14056±1129	116	12031±441	11308	479± 133	0.8± 0.3	6	96
	Spl-SplBI	13991±1154	116	11651±131	**11262**	5936±1398	13.4± 3.2	834	166

Table 4. Comparison of construction and local search algorithms, instance R6.

	Algorithm	Construction EENS	Construction N.Est.	Local search final solution EENS	Min.	N.Est.	Time	<Gr
15 switches	Gr-BI	2508	3135	2489		15195	38.9	1
	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
	SGr-Spl	5329±585	3135	2717±217	2354	4164± 338	6.5± 1.4	19
	SGr-SplBI	5371±620	3135	2335± 99	**2236**	31948± 7034	97.1± 24.3	93
	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
	Rnd-Spl	6394±568	1	2677±213	2327	1157± 343	3.8± 1.3	25
	Rnd-SplBI	6448±495	1	2328± 78	**2236**	28087± 7176	91.0± 26.2	98
	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
	Spl-Spl	3177±289	307	2672±198	2306	981± 292	2.5± 1.0	21
	Spl-SplBI	3198±296	307	2343± 86	**2236**	27273± 6359	81.8± 22.7	96
20 switches	Gr-BI	1925	4130	1794		31570	163.5	1
	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
	SGr-Spl	4668±593	4130	2011±143	1822	6580± 671	20.6± 4.9	32
	SGr-SplBI	4641±563	4130	1840± 68	**1793**	40287±10479	213.8± 60.8	82
	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
	Rnd-Spl	5975±607	1	1997±143	1814	2704± 755	15.6± 4.9	38
	Rnd-SplBI	5882±575	1	1853± 82	**1793**	38744±10708	224.4± 61.6	77
	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
	Spl-Spl	2541±215	404	1998±150	1800	2173± 721	10.9± 4.2	38
	Spl-SplBI	2533±206	404	1834± 68	**1793**	39151±11822	220.9± 66.7	87

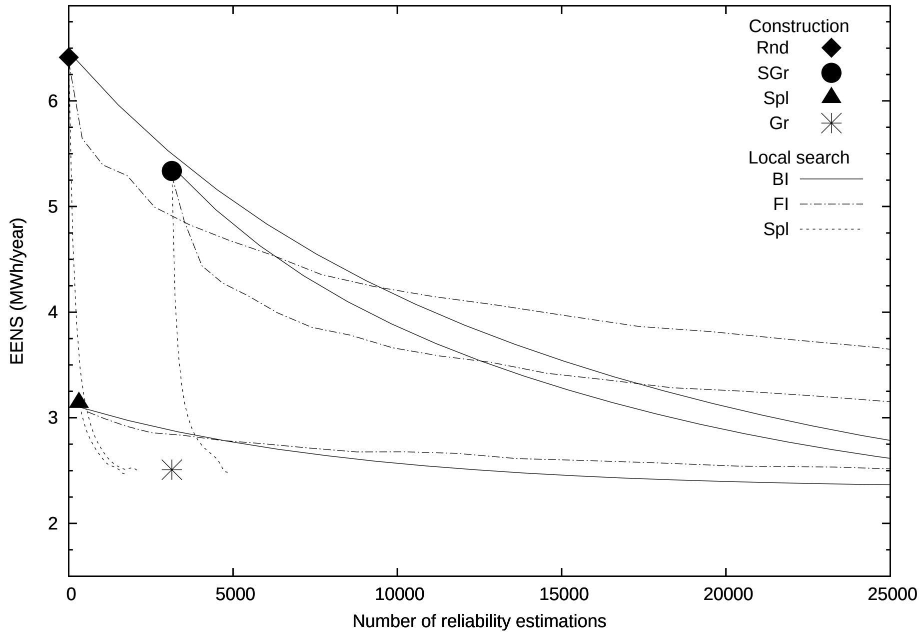


a. 15 switches.

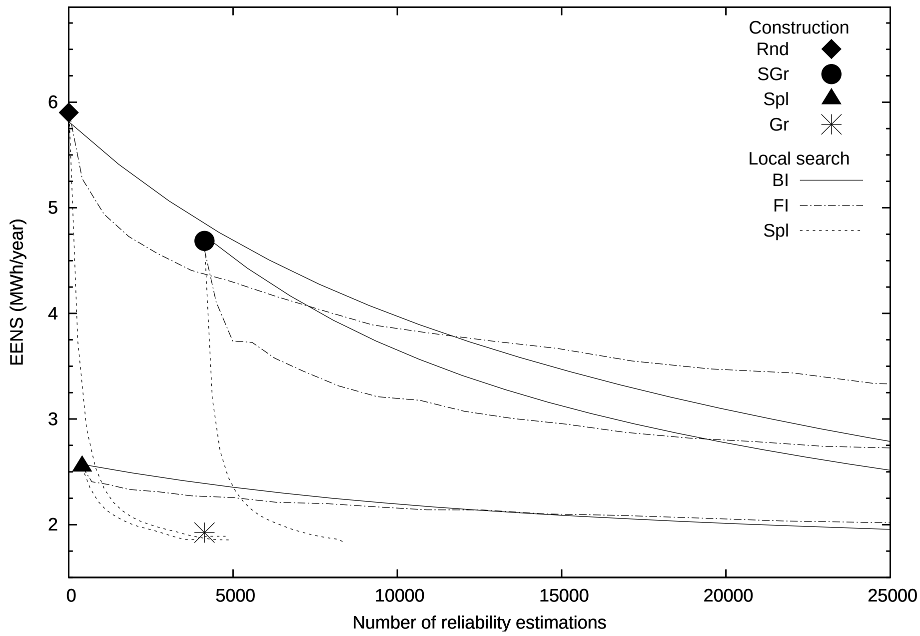


b. 20 switches.

Fig. 5. Average performance for instance B4.



a. 15 switches.



b. 20 switches.

Fig. 6. Average performance for instance R6.

construction algorithms. For the final solutions obtained after local searches, the tables present the average EENS, the average number of reliability estimations, the average running time and the best solution found by each combination within all the repetitions (Min. column). The last columns compare the number of final solutions that reach (=GR column) or overcome (<GR column) the corresponding greedy solution. The figures 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 (FI), best improvement (BI) and sample local searches. The three local searches show the same behavior for all the test cases, independently of the constructive algorithms.

First, we analyze construction algorithms. Solutions created by a semi-greedy algorithm are better than random solutions in average by 2000 KWh/year (for B4, 1100 for R6), but the required number of reliability estimations 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, 3000 for R6). Greedy construction generates always the best initial solution at the same cost than semi-greedy, but this solution is 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 (for B4, 3300 for R6), and they require less than 120 estimations (for B4, 410 for R6). Thus, sample construction 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 its high number of reliability estimations.

Now, we analyze the local search algorithms. The average final solutions of FI and BI are very close, and they yield the best result with all construction algorithms. The biggest difference between FI and BI is 26 MWh/year (semi-greedy for R6 with 20 switches), and it is half of the smallest standard deviation. The difference between FI and BI is in their performance over time. The figures show that FI progresses quickly in the beginning, but BI becomes better after some iterations. BI has an stable number of reliability estimations in each iteration along the whole search. FI takes any solution better than current and the number of estimations varies with the iterations. This is an advantage in early iterations because FI finds easily better solutions, but becomes a disvantage in the late iterations because FI restarts the local search with any small improvement when the number of reliability estimations is almost the same than BI. Thus, FI spends more time than BI in average.

The average final solutions of sample local search are worse than FI and BI. The difference with FI and BI is less than 700 KWh/year (for B4, 400 for B6). Moreover, sample local search was able to find the best solution for instance B4 with 20 switches. The time that it spent is very small, about half the time of the greedy or semi-greedy construction alone. The number of reliability estimations of sample local search is constant in each iteration like BI, but is 100 times smaller because the neighbourhood is restricted randomly to ten percent of switches and ten percent of free lines.

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 in small time. When a BI is applied after sample local search, it reaches the average results than BI or FI alone, but saving at least a quarter of the running time. For instance B4, about half of the final solutions stuck in the greedy solution after FI or BI local search, in particular after sample construction.

Finally, we analyze the combinations of construction and local search. If we consider each row of Tables 3 and 4 as one multi-start iterated local search, with 1000 iterations (for B4, 100 for R6), and each row with semi-greedy construction as a greedy randomized adaptive search procedure (GRASP) [16], we observe that iterated search processes with FI and BI 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 times the shown average (for B4, 100 for R6).

A GRASP is as effective as an iterated local search with random initial solutions, but needs less time. Rnd-BI is the combination that finds the biggest number of solutions that overcome the greedy solution. The most expensive combination is Rnd-FI.

The cheapest method for an iterated local search would be the Spl-Spl combination, its execution time is at least two times faster than a greedy or semi-greedy construction algorithm alone. The best method for an iterated local search would be the Spl-SplBI combination, because it is the cheapest combination in terms of execution time that is able to find the best solution. This verifies that a restricted neighbourhood speeds up the construction and the search processes.

5 Concluding Remarks

In this paper, we presented construction and local search methods for the switch allocation problem, with the service restoration problem as a subproblem. The objective is to improve network reliability by decreasing the unattended demand in case of line failures. We presented and compared the combination of four construction algorithms and three local searches strategies. Experimental results show that sample construction and sample local search are very inexpensive and create good and diverse solutions. They also show that semi-greedy construction is expensive and does not generate significative improvements in start solutions.

The present work indicates that a more directed local search combined with sample construction might give better results. In future work, we intend to propose an iterated search that uses a path relinking between solutions created by sample construction.

References

[1] A. J. Benavides, M. S. Machado, A. M. Costa, M. Ritt, L. S. Buriol, V. J. Garcia, and P. M. Franca. A comparison of tabu search and GRASP for the switch allocation problem. In *XLI Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, Porto Seguro, Brazil, Jul 2009. Sociedade Brasileira de Pesquisa Operacional (SOBRAPO).

[2] R. Billinton and S. Jonnavithula. Optimal switching device placement in radial distribution systems. *Power Delivery, IEEE Transactions on*, 11(3):1646–1651, Jul 1996. ISSN 0885-8977. doi: 10.1109/61.517529.

[3] E. M. Carreno, N. Moreira, and R. Romero. Distribution network reconfiguration using an efficient evolutionary algorithm. *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–6, June 2007. ISSN 1932-5517. doi: 10.1109/PES.2007.385648.

[4] P. M. S. Carvalho, L. A. F. M. Ferreira, and A. J. C. da Silva. A decomposition approach to optimal remote controlled switch allocation in distribution systems. *Power Delivery, IEEE Transactions on*, 20(2):1031–1036, April 2005. ISSN 0885-8977. doi: 10.1109/TPWRD.2004.838470.

[5] G. Celli and F. Pilo. Optimal sectionalizing switches allocation in distribution networks. *Power Delivery, IEEE Transactions on*, 14(3):1167–1172, Jul 1999.

[6] S. Civanlar, J. J. Grainger, H. Yin, and S. S. H. Lee. Distribution feeder reconfiguration for loss reduction. *Power Delivery, IEEE Transactions on*, 3(3): 1217–1223, July 1988.

[7] L. G. W. da Silva, R. A. F. Pereira, and J. R. S. Mantovani. 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*, pages 51–56, Nov. 2004.

[8] L. G. W. da Silva, R. A. F. Pereira, J. R. Abbad, and J. R. S. Mantovani. Optimised placement of control and protective devices in electric distribution systems through reactive tabu search algorithm. *Electric Power Systems Research*, 78(3):372 – 381, 2008. ISSN 0378-7796. doi: DOI: 10.1016/j.epsr.2007.03.005.

[9] A. C. B. Delbem, A. C. P. L. F. de Carvalho, and N. G. Bretas. Main chain representation for evolutionary algorithms applied to distribution system reconfiguration. *Power Systems, IEEE Transactions on*, 20(1):425–436, Feb. 2005. ISSN 1558-0679. doi: 10.1109/TPWRS.2004.840442.

[10] H. Falaghi, M.-R. Haghifam, and C. Singh. Ant colony optimization-based method for placement of sectionalizing switches in distribution networks using a fuzzy multiobjective approach. *Power Delivery, IEEE Transactions on*, 24(1):268–276, Jan. 2009. ISSN 0885-8977. doi: 10.1109/TPWRD.2008.2005656.

[11] Y.-J. Jeon, J.-C. Kim, J.-O. Kim, J.-R. Shin, and K. Y. Lee. An efficient simulated annealing algorithm for network reconfiguration in large-scale distribution systems. *Power Delivery, IEEE Transactions on*, 17(4):1070–1078, Oct 2002. ISSN 0885-8977. doi: 10.1109/TPWRD.2002.803823.

[12] R. Kavasseri and C. Ababei. Reds: Repository of distribution systems, 2009. URL http://venus.ece.ndsu.nodak.edu/ kavasseri/reds.html.

[13] T. Q. D. Khoa and P. T. T. Binh. A hybrid ant colony search based reconfiguration of distribution network for loss reduction. pages 1–7, Aug. 2006. doi: 10.1109/TDCLA.2006.311584.

[14] G. Levitin, S. Mazal-Tov, and D. Elmakis. Genetic algorithm for optimal sectionalizing in radial distribution systems with alternate supply. *Electric Power Systems Research*, 35:149–155, 1995.

[15] A. Moradi and M. Fotuhi-Firuzabad. Optimal switch placement in distribution systems using trinary particle swarm optimization algorithm. *Power Delivery, IEEE Transactions on*, 23(1):271–279, Jan. 2008. ISSN 0885-8977. doi: 10.1109/TPWRD.2007.905428.

[16] M. G. C. Resende and C. C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–249. Kluwer Academic Publishers, 2003.

[17] L. G. Santander, F. A. Chacra, H. Opazo, and E. Lopez. Minimal loss reconfiguration based on simulated annealing meta-heuristic. In *CONIELECOMP '05: Proceedings of the 15th International Conference on Electronics, Communications and Computers*, pages 95–99, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2283-1. doi: http://dx.doi.org/10.1109/CONIEL.2005.46.

[18] C.-T. Su, C.-F. Chang, and J.-P. Chiou. Distribution network reconfiguration for loss reduction by ant colony search algorithm. *Electric Power Systems Research*, 75(2-3):190 – 199, 2005. ISSN 0378-7796. doi: DOI: 10.1016/j.epsr.2005.03.002.

[19] J.-H. Teng and Y.-H. Liu. A novel ACS-based optimum switch relocation method. *Power Systems, IEEE Transactions on*, 18(1):113–120, Feb 2003. ISSN 0885-8950. doi: 10.1109/TPWRS.2002.807038.

[20] C. Wang and H. Z. Cheng. Optimization of network configuration in large distribution systems using plant growth simulation algorithm. In *Power Systems, IEEE Transactions on*, volume 23, pages 119–126, Feb. 2008. doi: 10.1109/TPWRS.2007.913293.

[21] C. Wang, H. Z. Cheng, and L. Z. Yao. Reactive power optimization by plant growth simulation algorithm. In *Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008. Third International Conference on*, pages 771–774, April 2008. doi: 10.1109/DRPT.2008.4523509.

[22] W.-C. Wu, M.-S. Tsai, and F.-Y. Hsu. A new binary coding particle swarm optimization for feeder reconfiguration. In *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, pages 1–6, Nov. 2007. doi: 10.1109/ISAP.2007.4441672.

[23] C. Zhang, J. Zhang, and X. Gu. 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*, pages 455–459, 2007. doi: 10.1109/ICNC.2007.708.

[24] D. Zhang, Z. Fu, L. Zhang, and Z. Song. 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*, pages 310–314, Stevens Point, Wisconsin, USA, 2005. World Scientific and Engineering Academy and Society (WSEAS). ISBN 999-2222-11-X.

[25] D. Zhang, Z. Fu, and L. Zhang. An improved ts algorithm for loss-minimum reconfiguration in large-scale distribution systems. *Electric Power Systems Research*, 77(5-6):685 – 694, 2007. ISSN 0378-7796. doi: DOI: 10.1016/j.epsr.2006.06.005.