Generator Start-up Sequences Optimization for Network Restoration Using Genetic Algorithm and Simulated Annealing

Paul Kaufmann
Department of CS
University of Paderborn, Germany
paul.kaufmann@gmail.com

Cong Shen
Department of EE and CS
University of Kassel, Germany
cong.shen82@yahoo.de

ABSTRACT

In the domain of power grid systems, scheduling tasks are widespread. Typically, linear programming (LP) techniques are used to solve these tasks. For cases with high complexity, linear system modeling is often cumbersome. There, other modeling approaches allow for a more compact representation being typically also more accurate as non-linear dependencies can be captured natively.

In this work, we focus on the optimization of a power plant start-up sequence, which is part of the network restoration process of a power system after a blackout. Most large power plants cannot start on their own without cranking energy from the outside grid. These are the non-black start (NBS) units. As after a blackout we assume all power plants being shut down, self-contained power plants (black start (BS) units), such as the hydroelectric power plants, start first and boot the NBS units one after each other. Once a NBS unit is restored, it supports the restoration process and because an average NBS unit is much larger than a BS unit, NBS unit's impact on the restoration process is typically dominant. The overall restoration process can take, depending on the size of the blackout region and the damaged components, some hours to weeks. And as the blackout time corresponds directly to economic and life losses, its reduction, even by some minutes, is worthwhile.

In this work we compare two popular metaheuristics, the genetic (GA) and simulated annealing (SA) algorithms on start-up sequence optimization and conclude that an efficient restoration plan can be evolved reliably and, depending on the implementation, in a very short period of time allowing for an integration into a real-time transmission system operation tool.

1. INTRODUCTION

With better and cheaper communication and monitoring capabilities the modern electric power systems became more and more flexible and robust. However, the risk of a total power system blackout is still present and with the increasing share of renewable energy sources even on the rise. A power system blackout can cause dramatic consequences. Recent power system blackouts (for instance, the Northeast America blackout in 2003 [6], the Japan power system collapse caused by an earthquake in 2011 [1] and the Northern, Eastern and Northeast India power system blackout in July 2012 [8]) have demonstrated that an efficient power system restoration plan is of utmost importance.

For an optimized generator start-up sequence, multiple approaches have been investigated. In [11], the generator start-up sequence is formalized as a mixed integer linear programming problem. However, while optimizing the start-up sequence the paper considers only the temporal constraints. Important performance indexes such as the power increasing rate, capacity, reliability and node importance degree of an NBS power plant are neglected. As performance indexes are often rated differently regarding their relevances by human experts, using them as a part of the goal function requires harmonization. This can be done by the the analytic hierarchy process and vague sets, as presented in [12] or by the fuzzy Choquet integral operator and group decision making, as done in [5] and [4].

A very similar problem to the generator start-up sequence optimization is the optimization of restoration paths, where a rebooted power plant starts powering up neighboring loads. This is the subsequent step after restoring power plants and has been investigated in [2] by reformalizing the process as an combinatorial problem and making it input to a quantum-inspired evolutionary algorithm. In [9] the restoration path selection has been solved by using a multi-objective evolutionary algorithm giving the system operator different solutions that are maximizing the load shedding and minimizing the switching operations. The similarity of restoration path and generator start-up sequence optimization comes not only from the fact that the two tasks are closely interleaved during the restoration process. Their encodings and the algorithmic approaches for solving them are very similar. Combining generator start-up sequence and restoration path optimization into a single task would allow for better solutions. However, the search space grows exponentially. To achieve appropriate computing times decomposition of the algorithmic components can be required.

The entire restoration process is dynamic, composing of many subproblems and is inherently multi-objective. Its formalization as a single and monolithic optimization task would allow for high-quality solutions but would also be very

GECCO '15, July 11 - 16, 2015, Madrid, Spain
© 2015 ACM. ISBN 978-1-4503-3472-3/15/07...$15.00
DOI: http://dx.doi.org/10.1145/2739480.2754647

likely not fast enough for making decisions within minutes. Decomposing the restoration process into tasks handled by metaheuristics allows for computing potential decisions immediately, continuously improving their quality. New status information can easily make its way into the running optimization process as well as help avoiding a full optimization restart, which can be necessary when using linear programming techniques. The contribution of this paper is therefore a study of a metaheuristic-friendly encoding for capturing the generator start-up sequence, analysis of appropriate operators such as perturbation/mutation and recombination, first work towards efficient multi-objective evolutionary optimization of generator start-up sequences, and the comparison of two popular methods for combinatorial challenges, the Simulated Annealing and Genetic Algorithm.

The paper is organized as follows: Section 2 describes the generator start-up sequence formulation with its constraints and the objective function. Section 3 sketches the employed optimization algorithms, their parameters and operators. Section 4 presents the methodological approach of the work, shows the data set and the performance metrics, investigates the inner mechanisms of the proposed operators and finally compares Simulated Annealing and the Genetic Algorithms on generator startup sequence optimization. Finally, Section 5 concludes the analysis, summarizing the restrictions and outlining current and future efforts.

2. FORMULATION OF GENERATOR START-UP PROCEDURE

In this section, we introduce a simplified boot sequence model of a power plant and present constraints as well as objective functions for the overall optimization problem.

2.1 Temporal BS Unit Boot Sequence Model

A generator start-up sequence begins with booting all NB units. NB units can boot independently. As soon as there is energy in the grid, the first NBS unit starts booting which draws energy from the grid for some time. When there is enough free energy available again, the next NBS unit is started, and so on. Fig. 1 illustrates an example for a booting procedure for a grid with one BS and one NBS unit. Fig. 1 (a) shows the output power of a BS unit during the restoration process. Starting at $t^{\text{BS}}_{\text{start}}$, denoted as t_0 in the total available grid power diagram in Fig. 1 (c), the BS unit initiates its internal components for $t^{\text{BS}}_{\text{prep}}$ units of time. Then, at the time point $t_1 = t^{\text{BS}}_{\text{start}} + t^{\text{BS}}_{\text{prep}}$, the BS unit starts injecting energy into the grid and increases its output power linearly for $t^{\text{BS}}_{\text{inc}}$ time units until reaching 90% of the maximal output power $P^{\text{BS}}_{\max}$ at $t_7 = t^{\text{BS}}_{\text{start}} + t^{\text{BS}}_{\text{prep}} + t^{\text{BS}}_{\text{inc}}$. As a general rule, the maximal output of BS and NBS units is assumed to be 90% of their rated power due to reserve power for frequency and voltage control. The capacity of a unit is defined as the sum of the rated and cranking powers. The cranking power of a BS unit is set to be zero. A closed form of a BS unit output function during restoration is:

$$P^{\text{BS}}_{\text{out}}(t) = \begin{cases} 0 & \text{if } t < t^{\text{BS}}_{\text{start}} + t^{\text{BS}}_{\text{prep}} =: t_1 \\ \frac{0.9 \cdot P^{\text{BS}}_{\max}}{t^{\text{BS}}_{\text{inc}}}(t - t_1) & \text{if } t_1 \leq t < t_1 + t^{\text{BS}}_{\text{inc}} =: t_7 \\ 0.9 \cdot P^{\text{BS}}_{\max} & \text{else.} \end{cases} \quad (1)$$

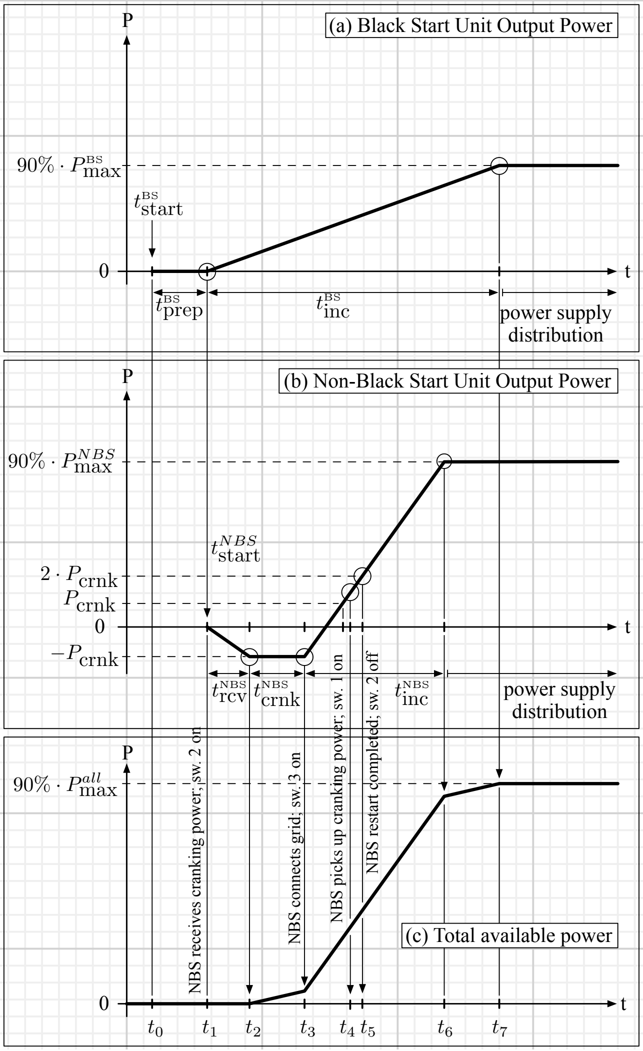
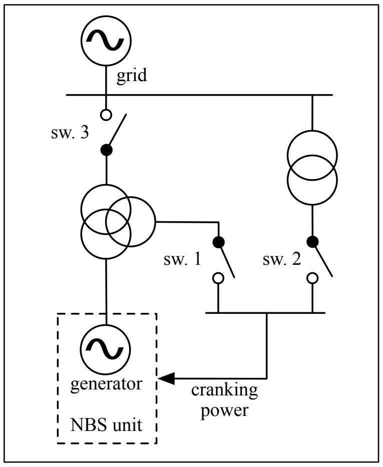


Figure 1: Temporal model of the boot process of a grid with one BS and one NBS unit.

2.2 Temporal NBS Unit Boot Sequence Model

Fig. 2 illustrates the simplified interconnect between a generator of an NBS unit and the outer grid. Three power flows can be identified in this figure. In the regular operation mode the energy produced by the generator, presented by a circle with a sine wave inside, is flowing through a transformer, presented by three overlapping circles, and a closed switch no. 3 to the outside grid. Additionally, part of the produced energy flows through the same transformer and the closed switch no. 1 to NBS unit's own ancillary devices. Switch no. 2 is open. This way, the NBS unit can produce and consume it's own cranking power, which is the regular case in normal operation conditions.

In case an NBS unit is currently booting and producing not enough power to supply its ancillary devices, it needs support form the grid. This can be realized by enabling a third power flow from the grid over the second transformer, presented by two overlapping circles, and the closed switch no. 2. The correct switching sequence during a start up is presented at the bottom of Fig. 2 and will be explained



	sw. 1	sw. 2	sw. 3	description
	○	○	○	after blackout
t_1	○	●	○	receiving cranking power
t_3	○	●	●	connecting grid
t_4	●	●	●	picking up cranking pwr
t_5	●	○	●	restart completed

Figure 2: Temporal model of the boot process of a NBS unit.

later on in this section. Last thing to mention is that any switch operation can cause electrical fluctuations. Therefore, switch operations need a synchronization phase and some time period after the switching point to ensure that frequency and voltage fluctuations are coming back to normal values. This, and other electrical properties such as over-voltage problems caused by self-excitation and energizing unload transmission lines, frequency control during the load restoration, and cold load pick up inrush have to be either validated before starting or checked during the optimization procedure. In our previous work we have already presented handling these requirements in an optimization algorithm [10, 9]. Therefore, we will skip this, as the focus of the current work is to find good performing encodings, operators and optimization algorithms for generator start-up sequence optimization. In our future work, we will incorporate electric checks into the optimization challenge.

The boot procedure of the NBS unit is shown in Fig. 1 (b) and is modeled as follows:

1. After a blackout, we assume all NBS units being disconnected from the grid. All three switches of an NBS unit are open in Fig. 2. If some NBS unit is selected for restoration, it closes first switch no. 2 and starts consuming cranking energy from the grid. In Fig. 1 this happens at $t_1 = t^{\text{NBS}}_{\text{start}} = t^{\text{BS}}_{\text{start}} + t^{\text{BS}}_{\text{prep}}$. For simplicity, we assume that the NBS unit consumes the entire free energy of the grid but at most P_{crnk}.

2. As the BS unit increases its output power linearly from 0, its output reaches P_{crnk} after $t^{\text{NBS}}_{\text{rcv}}$ time units at $t_2 = t^{\text{NBS}}_{\text{start}} + t^{\text{NBS}}_{\text{rcv}}$. From now on, the NBS units consumes P_{crnk} for $t^{\text{NBS}}_{\text{crnk}}$ time units until $t_3 = t^{\text{NBS}}_{\text{start}} + t^{\text{NBS}}_{\text{rcv}} + t^{\text{NBS}}_{\text{crnk}}$. In our optimization set up we assume that consuming cranking power will not be interrupted. In a more realistic model this event should also be considered adding some penalty time to $t^{\text{NBS}}_{\text{crnk}}$.

3. After consuming cranking power for $t^{\text{NBS}}_{\text{crnk}}$ time units and energizing its ancillary devices, the NBS unit synchronizes to the grid, closes the switch no. 3 and starts producing it's own power as well as injecting it into the grid. Switch no. 2 stays closed until the NBS unit produces enough energy to power its own ancillary devices. For the same reason switch no. 1 stays open.

4. After the NBS unit is able to produce more energy than its own ancillary devices consume, it closes the switch no. 1 at t_4 preparing to power its ancillary devices by itself.

5. If the output power of the NBS unit reaches some security margin, in our case we set this to twice the amount of the cranking power, the NBS unit opens the switch no. 2. From this moment on, the NBS unit produces all the energy it requires to operate by itself and does not rely anymore on the energy from the outside grid. In Fig. 1 (b) this happens at t_5. The switch positions reach also their regular configuration.

6. Finally, after injecting energy for $t^{\text{NBS}}_{\text{inc}}$ time units the NBS unit reaches at $t_6 = t_3 + t^{\text{NBS}}_{\text{inc}}$ 90% of it's rated power and enters the normal operation conditions. The power produced by the NBS unit is available as cranking power to other NBS units.

A closed form formulation of the output power function $P^{\text{NBS}}_{\text{out}}$ of a NBS unit is presented in the equation below. In contrast to $P^{\text{BS}}_{\text{out}}$, which depends only on the input parameter time, $P^{\text{NBS}}_{\text{out}}$ depends also on the available power of the system, which in our case is $P^{\text{BS}}_{\text{out}}$, and the time point where this system power gets available: $t^{\text{NBS}}_{\text{start}} = t^{\text{BS}}_{\text{start}} + t^{\text{BS}}_{\text{prep}} = t_1$. With this, $P^{\text{NBS}}_{\text{out}}(t, P_{\text{in}}, t^{\text{NBS}}_{\text{start}})$ amounts to

$$P^{\text{NBS}}_{\text{out}} = \begin{cases} 0 & \text{if } t < t^{\text{NBS}}_{\text{start}} =: t_1 \\ -P_{\text{in}} & \text{if } t_1 \leq t < t_1 + t^{\text{NBS}}_{\text{rcv}} =: t_2 \\ -P_{\text{crnk}} & \text{if } t_2 \leq t < t_2 + t^{\text{NBS}}_{\text{crnk}} =: t_3 \\ \frac{0.9(P^{\text{NBS}}_{\text{max}} + P_{\text{crnk}})(t-t_3)}{t^{\text{NBS}}_{\text{inc}}} - P_{\text{crnk}} & \text{if } t_3 \leq t < t_3 + t^{\text{NBS}}_{\text{inc}} =: t_6 \\ 0.9 \cdot P^{\text{NBS}}_{\text{max}} & \text{else.} \end{cases} \tag{2}$$

2.3 Combined Start-up Model

The total power in the system during the booting process is a sum of $P^{\text{BS}}_{\text{out}}$ and $P^{\text{NBS}}_{\text{out}}$ and is illustrated in Fig. 1 (c). Since after t_3 the NBS unit can send its power to the network, the ramp rate between t_3 and t_6 equals the sum of the output powers of the BS and NBS units. The NBS unit reaches 90% of the nominal output at t_6 and the ramp rate reduces to the output power increasing rate of the BS unit. After t_7, the BS unit also reaches 90% of its nominal value completing the booting process of this example.

In a larger network model with several BS and NBS units, the NBS unit start sequence determines the temporal booting procedures of the power plants. We have developed a corresponding simulator that takes as input a given power network with generators and their characteristics as well as an NBS start-up sequence and computes the development of the overall power and the overall restoration time of the power grid. The operation time of switches and switch insulators is neglected as this time is short compared to the total restoration time.

2.4 Constraints

Following constraints have to be respected during the restoration process:

1. Each NBS unit should be able to start. This criteria is relaxed in this work to: All BS units have to have enough accumulated power P^B to restart any of the NBS units with the cranking power of P_{crnk}: $P^B \geq P_{\text{crnk}}$.

2. An NBS unit should receive its cranking power $P_{\text{crnk},i}$ uninterrupted for at least T_i^{min} time to be able to start. We assume this constraint to be valid in the simulation. For real situations one has to check, whether there is always enough cranking power in the grid between t_2 and t_3, adding otherwise penalty time of m hours to the restoration process of an NBS unit.

3. Since cranking power is consumed mainly by induction motors, it is important to ensure that node voltages and the network frequency lie in acceptable ranges when BS units send cranking power to NBS units.

The benchmarks used in this work are constructed such that the constraints hold. However, we have implemented and used run-time constraint checks detecting invalid solutions and rendering their fitness insufficient in our previous work [9, 10].

2.5 Objective Function

The two most important goals of the restoration process are to maximize the reliability of the generator start-up sequence, which essentially means minimizing the possibility that the power system collapses again during the boot process, and to minimize the booting time for the generator start-up sequence.

Given N as the number of NBS units, a generator start-up sequence is defined as $s = (s_1, s_2, \ldots, s_N), 1 \leq s_i \leq N, s_i \neq s_j$ if $i \neq j$, where the indices s_i refer to individual NBS units. Based on the reliability index $r[s_i]$ of a single generator, we can express the reliability $R(s)$ of a generator start-up sequence s as:

$$R(s) = \left[\sum_{i=1}^{N}\left[1-(i-1)\frac{1}{N}\right]\right]^{-1}\left[\sum_{i=1}^{N}\left[r[s_i]\left(1-(i-1)\frac{1}{N}\right)\right]\right]$$
$$= \frac{2}{N+1}\sum_{i=1}^{N}\left[r[s_i]\left(1-\frac{i-1}{N}\right)\right]. \tag{3}$$

The first part of Eqn. 3 is for normalization and the second presents the non-normalized reliability.

Consider as an example a grid with $N = 4$ NBS units with reliability indices of $r[\text{NBS}_1] = 0.9$, $r[\text{NBS}_2] = 0.95$, $r[\text{NBS}_3] = 0.8$ and $r[\text{NBS}_4] = 0.7$. For the starting sequence $s = (\text{NBS}_1, \text{NBS}_2, \text{NBS}_3, \text{NBS}_4)$ the non-normalized reliability amounts to $0.9 \cdot 1 + 0.95 \cdot (1 - 0.25) + 0.8 \cdot (1 - 2 \cdot 0.25) + 0.7 \cdot (1 - 3 \cdot 0.25) = 2.1875$ and the normalized reliability $R(s)$ is given by $\frac{2.1875}{2.5} = 0.875$. The generator start up sequence $s' = (\text{NBS}_2, \text{NBS}_1, \text{NBS}_3, \text{NBS}_4)$ results in an overall reliability of $R(s') = \frac{2.2}{2.5} = 0.88$. Thus, the second sequence should be preferred over the first as the chance for a brake-down during the booting process is lower.

Along with the reliability, the booting time for a generator start-up sequence is the second criteria. The optimization goal is to arrange the generator start-up sequences such that the reliability R is maximized and the restoration time T is minimized. We define a single-objective goal function by linearly weighting and aggregating the reliability $R(s)$ and the restoration time $T(s)$ as:

$$F_{R,T}(s) = \frac{1}{2}\left(R(s) + \left(1 - \frac{T(s)}{T_{\max}}\right)\right), \tag{4}$$

where $T_{\max}$ is set slightly larger than the worst case time effort for all power plants.

Table 1: Temperature control strategies. T_0, T_N, and t are the start, terminal, and current temperatures. N is the number of SA iterations.

$T_t^{(1)}$	$\leftarrow T_0 - \frac{T_0 - T_N}{N}$
$T_t^{(2)}$	$\leftarrow T_0 \left(\frac{T_n}{T_0}\right)^{\frac{t}{N}}$
A B $T_t^{(3)}$	$\leftarrow \frac{(T_{\text{start}} - T_{\text{end}})(N+1)}{N}$ $\leftarrow T_{\text{start}} - A$ $\leftarrow \frac{A}{t+1} + B$
$T_t^{(4)}$	$\leftarrow 0.5(T_0 - T_N)(1 + \cos(\frac{\pi t}{N})) + T_N$
$T_t^{(5)}$	$\leftarrow 0.5(T_0 - T_N)(1 - \tanh(\frac{10t}{N} - 5)) + T_N$
$T_t^{(6)}$	$\leftarrow \frac{T_0 - T_N}{\cosh(\frac{10t}{N})} + T_N$
$T_t^{(7)}$	$\leftarrow T_0 \exp(-\frac{1}{N} \ln(\frac{T_0}{T_N})t)$
$T_t^{(8)}$	$\leftarrow T_0 \exp(-\frac{1}{N^2} \ln(\frac{T_0}{T_N})t^2)$

The selection of an aggregated and not a Pareto-based fitness function has the following reasons: For a fair comparison we would like to employ the same goal function definition for all candidate optimization algorithms. A linear combination of the objective functions is the easiest way to achieve this and is used in this work. A common Pareto-based goal function is ongoing work.

3. ALGORITHMS, ENCODINGS AND OPERATORS

3.1 Simulated Annealing (SA)

Simulated Annealing (SA) is a trace-based algorithm iterating from one solution to the other by means of a perturbation operator. Unlike Hill Climbing (HC), SA may also accept worse solutions at a rate, reciprocal to the functional qualities of the current and the derived solutions. Also unlike the Metropolis Algorithms, this rate reduces over time letting the SA account for the state of the optimization run. The rational behind this is that at the beginning SA globally searches for regions with a good potential for local and global optima while towards the end the perturbation horizon is tightened for locally exploiting a region in a HC manner. The implementation of a contracting neighborhood definition is realized by a temperature parameter T_t, which is reduced during an SA run gradually. There are, however applications benefitting from episodic "reheating" of T, such as floorplanning in chip design. In our experiments we have exhaustively evaluated eight temperature control strategies. They are presented in Tab. 1.

In our implementation we have realized the regular SA algorithm. The only variations are that the size of a temperature level, the period between updating T_t is set to one and the rejection factor, which is the number of iterations without a change of the current solution, to ∞. These parameters have had marignal impact during our experiments.

3.2 Generator Start-up Encoding

The encoding of a power plant boot sequence is done as a string of unique numbers. The numbers are identifying the power plants. The order of numbers in the string indicates, which power plant to boot first, second, and so on. With this encoding, SA's perturbation and GA's mutation operators can be implemented canonically by swapping two numbers in the sequence, as presented in Fig. 3. The encoding remains correct after such a modification.

While this kind of encoding allows for a simple implementation of SA and EA operators, it has also the drawback of a potentially high epistasis. (High) Epistasis denotes an effect where changing some bits in the encoding of a solution (dramatically) impacts on the way some other bits in the encoding are evaluated by the goal function. A high epistatic encoding shows often no regularities, letting the perturbation/mutation and recombination operators having almost random impact on the goal function. Unfortunately, many combinatorial optimization problems have high-epistatic encodings. Despite the potential for high epistasis, Sec. 4 shows that the presented perturbation/mutation and the uniform order-based recombination work well with this encoding.

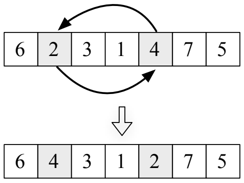


Figure 3: Perturbation / Mutation operator and power plant boot sequence definition.

3.3 Genetic Algorithm (GA)

In this work we use the standard Genetic Algorithm scheme where new population of individuals are derived from the old population in a loop where each time two parent individuals are selected by a 2-tournament selection, recombined and the off-spring individuals mutated. The recombination and mutation probabilities define, how frequently the individuals are modified instead of just cloning them. For instance, a mutation probability of 0.5 let the mutation operator return the original solution in 50% of the cases, returning a mutated individual otherwise. The recombination and mutation rates, on the opposite, specify the percentage of the gene material that is going to be modified. Before starting the GA loop, best 5% but at least one individual is copied to the new population.

3.4 The Recombination Operator

SA and GA share the same problem encoding and the perturbation/mutation operator. To realize global search behavior, GA uses additionally a recombination operator. Because ordered sequences are a widely used encoding model for many real world applications including the traveling salesman problem, some previous work has been done on meaningful recombination operators for this kind of representations. We have selected the uniform order-based crossover, which decides according to a uniform probability distribution for each gene (power plant number), whether to recombine or not. Then, these genes are not transferred to a second chromosome (start-up sequence encoding) but reordered in the original chromosome according to the sorting of these genes in the second chromosome. This way, ordered structures are transferred between chromosomes without producing incorrect encodings. An example is shown in Fig. 4. Recombined genes are marked by a "-", other genes by a "+". Genes selected for recombination in chromosome A are 3, 4, and 7. Their sequence in chromosome B is 3, 7, and 4. Thus, these three genes are reordered in chromosome A to 3, 7, 4. The chromosome A' shows the resulting solution. The same procedure is applied also to chromosome B. There, 7, 4, and 2 have to be reshuffled according to their order in chromosome A: 2, 4, and 7. The resulting chromosome B' is shown at the bottom of Fig. 4.

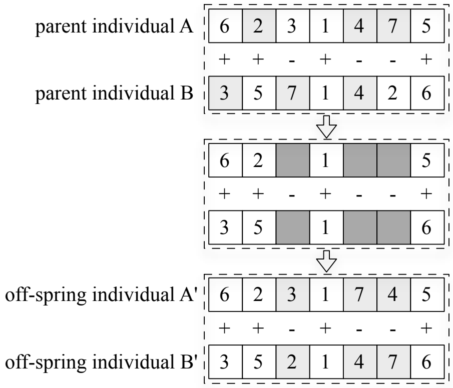


Figure 4: Uniform Order-Based Crossover: transplanting ordered subsets. A "-" denotes a gene selected for recombination. A set of genes selected for recombination in chromosome A: "3", "4", and "7" are reshuffled according to the order of these genes in chromosome B: "3", "7", and "4".

4. ANALYZING GENERATOR START-UP OPTIMIZATION

This section presents the evaluation methodology and metrics, analyzes good operator configurations, investigates optimization algorithm parameterizations, and finally compares Simulated Annealing and Genetic Algorithms on the task of generator start-up sequence optimization.

4.1 Evaluation Methodology and Metrics

GA and SA are randomized algorithms. Therefore, for each algorithm parameterization GA and SA have been executed 30 times with varying random seeds. The algorithm runs have been terminated after 100000 fitness evaluations and evaluated regarding the mean, standard deviation, first, second (median), and third quartiles as well as peak values. To compute these numbers, the best functional quality from each of the 30 runs has been extracted. In case of SA, the best solution is stored in a separate variable and printed out on exit. Elitism-based GA, on contrary, never forgets the best solution and propagates it into the final population, where it also gets printed out.

Once best performing SA and GA configurations have been found, the best functional qualities of the 30 runs for each of the algorithms are checked for statistical similarities using the two-tailed Mann-Whitney (MW) U and the two-tailed Kolmogorov-Smirnov (KS) tests at the significance level of $\alpha = 0.05$. The MW U test checks the hypothesis whether two independent samples come from distributions with equal medians. The KS test distinguishes between H0="Two independent sequences A and B follow the same distribution" and HA="Two independent sequences A and B follow different distributions."

Finally, we compare SA and GA regarding their computation times. For this task, the Computational Effort (CE) metric is a popular approach computing a bound on fitness evaluations to reach some optimization goal f_g at a specific probability z [3]. CE also computes the number of fitness evaluations after which an optimization run has to be restarted, avoiding stuck in local optima. In our investigation, we set z to 99% and f_g to 0.67. f_g is selected such that most of SA and GA configurations reach this functional quality, allowing for an precise comparison using the CE metric.

4.2 Data Test Case Setup

For first investigations we have used the New England 39 test case network defined in [7]. It is a greatly simplified model of a real network case consisting of 39 busses with 10 synchronous generators. We have extended this benchmark by additional 47 generators, which are parameterized similar to the 10 original generators. This network sizes are not uncommon to transmission and medium voltage distribution grids. However, much larger networks are also relevant and susceptible to a total blackout.

All restoration parameters of the test benchmark are presented in Tab. 2. The table columns show the power plant index, its type, the power increasing rate, the cranking power, the restart time, the rated power, and reliability. The cranking power is set uniformly to 20% of the rated power. Older power plants may need more and modern power plants less cranking power. The restart time denotes the preparation time ($t_{\text{prep}}^{\text{BS}}$) for black and cranking power receiving time ($t_{\text{crnk}}^{\text{NBS}}$) for non black start units.

4.3 Evaluating Operators

In first experiments we would like to get the intuition on how successful the perturbation/mutation and recombination operators are throughout the optimization process and how many genetic material the operators are modifying. We have configured a GA scheme setting the perturbation/mutation and recombination probabilities to 1.0 and the perturbation/mutation and recombination rates to randomly values between 0.0 and 1.0 sampled anew each time an operator is executed. Population sizes have been set to 4, 8, 16, 32, 64, and 128. The results are presented in Fig. 5 an Fig. 6 and are common to all GA parameterizations. Fig. 5 (a) presents the success rates and the amount of modified genetic material for the mutation operator. As expected, mutating even large amount of genes in the initial search phase often improves the functional quality. The distribution of modified genetic material, which corresponds to the perturbation/mutation rate, follows roughly $\exp(-x)$.

Table 2: Column 1: generator index, 2: generator type (1=BS, 2=NBS), 3: power increasing rate [MW/min], 4: cranking power (P_{crnk}) [MW], 5: restart time (BS: t_{prep}, NBS: t_{crnk}) [min], 6 - rated power ($P_{\max}$) [MW] (NBS: 5 $\cdot$ cranking power), 7 - reliability.

1	2	3	4	5	6	7
1	1	6.4	NA	10	90	0.89
2	2	6.4	70	10	70/0.2	0.93
3	2	5.6	110	12	110/0.2	0.73
4	2	6.8	90	20	90/0.2	0.94
5	2	5.8	70	20	70/0.2	0.96
6	2	6.2	70	25	70/0.2	0.78
7	2	4.9	120	20	120/0.2	0.76
8	2	4.2	50	15	50/0.2	0.69
9	2	6.6	95	10	95/0.2	0.76
10	2	4.6	89	25	89/0.2	0.86
11	2	5.8	95	18	95/0.2	0.83
12	2	3.2	50	40	40/0.2	0.89
13	2	5.6	60	50	50/0.2	0.98
14	2	4.2	40	46	40/0.2	0.93
15	2	2.2	30	30	30/0.2	0.92
16	2	6.1	60	50	60/0.2	0.83
17	2	5.2	60	40	60/0.2	0.78
18	2	3.4	40	40	40/0.2	0.96
19	2	1.6	18	20	18/0.2	0.73
20	2	6.2	60	50	60/0.2	0.91
21	2	3.3	40	40	40/0.2	0.92
22	2	3.1	35	36	35/0.2	0.86
23	2	2.3	25	28	25/0.2	0.84
24	2	3.2	50	40	50/0.2	0.88
25	2	6.8	66	57	66/0.2	0.98
26	2	6.2	60	50	60/0.2	0.94
27	2	6.3	66	57	66/0.2	0.98
28	2	2.8	36	38	36/0.2	0.95
29	2	5.8	70	48	70/0.2	0.85
30	2	4.6	18	65	18/0.2	0.78
31	2	7.2	60	67	60/0.2	0.77
32	2	2.3	40	89	40/0.2	0.87
33	2	5.1	35	87	35/0.2	0.68
34	2	9.3	25	43	25/0.2	0.78
35	2	3.2	50	34	50/0.2	0.92
36	2	5.8	66	56	56/0.2	0.97
37	2	7.2	60	23	60/0.2	0.63
38	2	2.3	66	45	66/0.2	0.87
39	2	5.8	36	56	36/0.2	0.86
40	2	7.8	70	23	70/0.2	0.83
41	2	5.3	66	56	66/0.2	0.97
42	2	7.8	36	67	36/0.2	0.87
43	2	6.8	70	67	70/0.2	0.92
44	2	3.6	18	56	18/0.2	0.95
45	2	4.2	60	53	60/0.2	0.98
46	2	5.3	40	67	40/0.2	0.91
47	2	6.1	35	36	35/0.2	0.93
48	2	2.3	25	45	25/0.2	0.97
49	2	8.2	50	46	50/0.2	0.87
50	2	3.8	66	89	66/0.2	0.86
51	2	5.2	60	56	60/0.2	0.83
52	2	6.3	66	56	66/0.2	0.87
53	2	7.8	36	89	36/0.2	0.92
54	2	5.8	70	56	70/0.2	0.91
55	2	4.3	66	45	66/0.2	0.73
56	2	3.8	36	76	36/0.2	0.86
57	2	6.8	70	56	70/0.2	0.88

Table 3: Perturbation / Mutation mechanism: The number of perturbed / mutated gene pairs is uniformly sampled from the presented table. These gene pairs are selected randomly and the genes are swapped.

1	1	1	1	1	1	1	2	2	2	3	5

After about 20000 fitness evaluations the distribution of successfully modified gene material stabilizes, allowing us to define the perturbation/recombination operator that samples its perturbation/mutation rate uniformly from the table presented in Tab. 3.

When analyzing the success behavior of the recombination operator, apart from the expected gradual reduction of the success probability over the time, no obvious behavior in the amount of the modified genes can be observed in Fig. 5 (b). However, when looking more precisely at the distribution of the recombined genes, as presented in Fig. 6 for the GA experiment with 16 individuals in a population and for the optimization interval between 10000 to 20000 fitness evaluations, the distribution becomes clear. The recombination operator is more successful when configured to recombine small and large amount of genes. The effect of the recombination operator is not symmetric. While recombining a small amount of genes, the effect of the recombination operator is very similar to the mutation operator. But recombining a lot of genes copies large and consolidated parts of a chromosome to the offspring individual at a slightly different position. This behavior comes close to shifting large parts of a chromosome around. Later experiments show that the recombination operator helps GA to excel when configured at the recombination rate of around 90%.

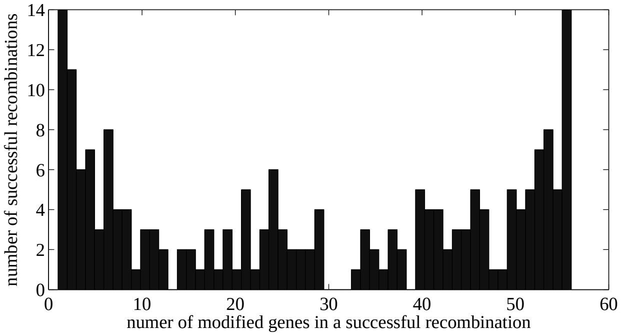


Figure 6: Distribution of modified genetic material for successful recombination operations between fitness evaluations 10000 and 20000.

4.4 Parametrizing GA and SA

After have identified good perturbation/mutation and recombination parameters, we have exhaustively tested SA and GA. SA was evaluated for all possible temperature range combinations between $T_{\text{start}} = 1000, 100, 10, 1, 0.1, 0.01, 0.001$ and $T_{\text{stop}} = 10, 1, 0.1, 0.001, 0.0001$ with $T_{\text{start}} > T_{\text{stop}}$ and regarding all eight cooling schemes. In total, 176 SA experiments with 30 SA runs each have been executed. GA have been first executed for all mutation and recombination probability combinations between 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 0.99 and population sizes of 4, 10, 20, 50, and 100. Higher mutation and recombination rates in combination with 20 individuals in a population showed best performances. Therefore, GA parameterization were refined in final experiments to all mutation and recombination probability combinations between 0.90, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99 and 0.999 and a population size of 20. In total, 705 GA experiments with 30 GA runs each have been executed.

Table 4: Results

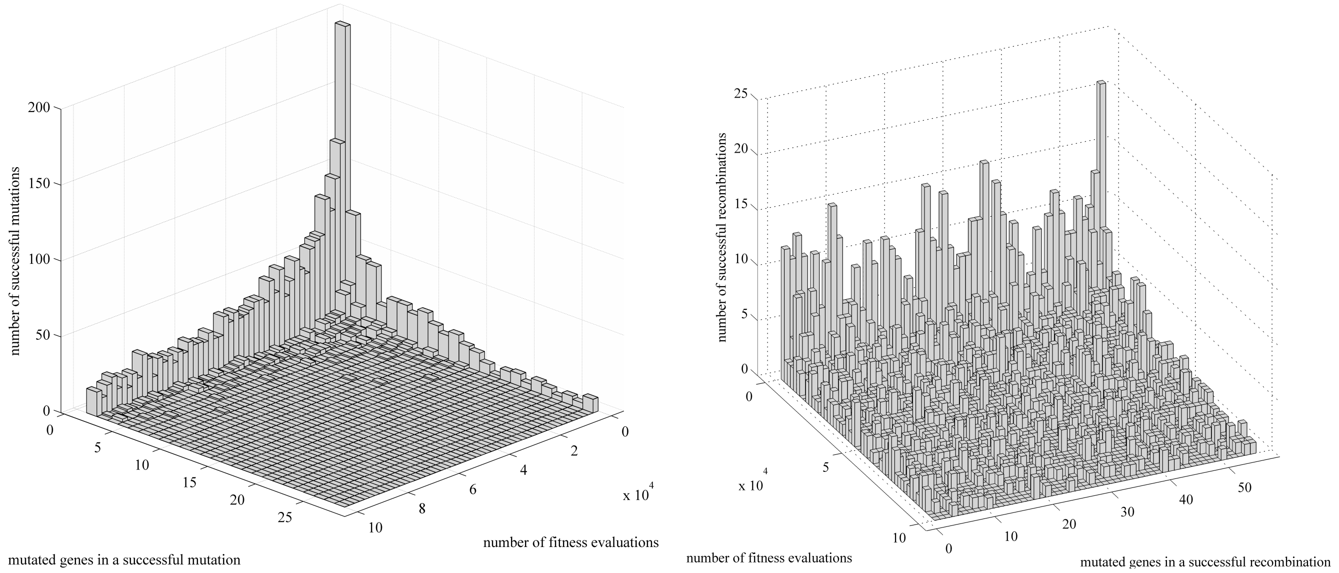
	SA	GA
Peak value	0.68342	0.68311
Mean±SD	$0.68199 \pm 1.2e^{-3}$	$0.67841 \pm 1.5e^{-3}$
1st Quartile	0.68139	0.67762
Median	0.68234	0.67835
3rd Quartile	0.68300	0.67963
MW U test, p-value	$1.6947e^{-09}$	
KS test, p-value	$1.1088e^{-08}$	
CE at $f_g = 0.67$	6201	6237
CE restart after	6201	6237

4.5 Comparing GA and SA

Tab. 4 presents the evaluation of best SA and GA configurations. SA achieves the best peak performance of 0.683423 when started at $T_{\text{start}} = 10.0$, $T_{\text{stop}} = 0.0001$ and the cooling scheme 3. GA achieves a peak performance of 0.683112 with the recombination probability/rate of 0.9 and the mutation probability of 0.9. While SA is slightly ahead of GA, similar peak values indicate that the exhaustive parameter tuning explored the most promising GA and SA parameter combinations.

The second group of lines in Tab. 4 summarizes the figures for the SA and GA configurations with best average finesses. For this, SA has to be configured with $T_{\text{start}} = 0.001$, $T_{\text{stop}} = 0.0001$ and the cooling scheme 6 and GA with the recombination probability/rate of 0.9 and the mutation probability of 0.999. The table figures show that while the standard deviations for both algorithms are compact and similar, the distance between the means amounts for 0.004. The distances for the medians and the quartiles are differing also for 0.003 to 0.004, underlying the disparity of performances. While the differences seem to be marginal, the MW U and the KS tests reject the h_0 hypothesis with very low p-values. Interestingly, the CE for SA is more than twice as large as for GA. This could indicate that SA is able to steadily improve the best solution even in the final search phase while the GA is fast in the beginning. In our Matlab implementation a regular Intel i7 notebook processor requires a second to compute 500 fitness values. To evolve solution with a functional quality of 0.67 and above with a probability of 99%, on average 12 seconds are therefore needed. However, implementing the optimization algorithms using C or C++ and parallelizing the GA can reduce the computation time significantly.

The best CE value achieved by SA amounts for 6201 fitness evaluations at $T_{\text{start}} = 1.0$, $T_{\text{stop}} = 0.0001$ and the cooling scheme 3. This is similar to the best GA figures



(a) Mutation operator

(b) Recombination operator

Figure 5: Number of successful executions of the perturbation/mutation and recombination operators and the distribution of the genetic material modified during this executions.

indicating again, that given similar or identical operators and problem encodings, the exhaustive parameter tuning was able to find for each of the optimization algorithms a parameterization allowing to reach similar peak values.

5. CONCLUSION

This paper investigates on the task of power plant boot scheduling, encoding models, operators and nature-inspired metaheuristics. The insights we have gained in our work are that SA and GA are highly effective for generator start-up sequence optimization with suitable computation times for real-time control systems for network restoration. Additionally, while epistasis is very likely a problem of the presented encoding, the uniform order-based crossover seems to work very well.

In ongoing and future work, we will implement a Pareto-based MOEA using the operators presented in this paper, try to mathematically formalize the restoration procedure more precisely, improve the execute time, and compare the algorithms to linear programming approaches.

6. REFERENCES

[1] M. Bruch, V. Münch, M. Aichinger, M. Kuhn, M. Weymann, and G. Schmid. Power Blackout Risks. *Allianz*, 2011.

[2] J. Hou, Z. Xu, Z. Y. Dong, and K. P. Wong. Permutation-based Power System Restoration in Smart Grid Considering Load Prioritization. *Electric Power Components and Systems*, 42(3-4):361–371, 2014.

[3] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[4] W. Liu and Z. Lin. Analysis and Optimization of the Preferences of Decision-Makers in Black-Start Group Decision-Making. *IET Generation, Transmission & Distribution*, 7:pp.14–23, 2012.

[5] W. Liu, Z. Lin, and F. Wen. Intuitionistic Fuzzy Choquet Integral Operator-based Approach for Black-start Decision-making. *IET Generation, Transmission & Distribution*, 6:378–386, 2012.

[6] New York Independent System Operator. Interim Report on the August 14, 2003 Blackout. *Harvard Kennedy School*, 2004.

[7] North American Rockwell Corporation and Edison Electric Institute. *On-line Stability Analysis Study: RP90-1.* North American Rockwell Information Systems Company, October 12, 1970.

[8] Power Systems Engineering Research Center. Selected Information About The July 31 Blackout in India Affecting the Northern and Eastern Regions. *Arizona State University*, 2012.

[9] C. Shen, P. Kaufmann, and M. Braun. A New Distribution Network Reconfiguration and Restoration Path Selection Algorithm. In *Power Systems Computation Conference (PSCC)*. IEEE, 2014.

[10] C. Shen, P. Kaufmann, and M. Braun. Optimizing the Generator Start-up Sequence After a Power System Blackout. In *IEEE Power and Energy Society General Meeting (IEEE GM)*, 2014.

[11] S. Wei and L. Chen-Ching. Optimal Generator Start-Up Strategy for Bulk Power System Restoration. *IEEE Transactions on Power Systems*, 26(3):1357–1366, 2011.

[12] S. Zeng, Z. Lin, and F. Wen. A New Approach for Power System Black-Start Decision-Making With Vague Set Theory. *International Journal of Electrical Power & Energy Systems*, 34:114–120, 2012.