A Comparison of Crossover Operators in Genetic Algorithms for Switch Allocation Problem in Power Distribution Systems

Ronaldd Pinho and Filipe Saraiva
Faculty of Computer Science
Institute of Exacts and Natural Sciences
Federal University of Pará
Belém, PA, Brazil
Email: ronaldppinho@gmail.com, saraiva@ufpa.br

***Abstract*—Combinatorial optimization problems are commonly found in different stages of electric distribution systems deploying. Such problems demand the use of metaheuristics to find good solutions with acceptable computational effort. Among these metaheuristics, genetic algorithms prove to be an interesting method for this kind of problem due to the good solutions found by them in several applications. From this context, the present paper proposes an analysis of the effect of different crossover operators on the quality of obtained solutions in a genetic algorithm applied to the switch allocation problem in power distribution systems. The comparisons were conducted based on a hypothetical system from the literature with 135 buses and 1 feeder. The experiments showed which the restriction degree imposed on the search space influences the differences between crossover operators. The results suggest which exists an ideal number of cut points for the multi-points crossover operator which found better results than the one-point, uniform, and other crossovers.**

***Index Terms*—Power Distribution Systems, Switch Allocation Problem, Power System Reliability, Combinatorial Optimization, Genetic Algorithms, Crossover Operators**

I. Introduction

In the planning stage of an electrical power distribution network, there is a critical concern related to the capacity to supply electricity continuously. It is expected to the power systems supply the consumers loads with high reliability and service quality [1].

Many factors contribute to the occurrence of interruptions on the energy supply service of a distribution system, such as atmospheric discharges, tree branches, interferences from external agents, etc [2].

There are several types of devices that can be placed in the network such as sectionalizing switches, circuit breakers, capacitor banks, and fuse switches. These devices can manage the system operation in several ways in order to decrease the impact of unexpected interruptions and minimize the failure propagation, keeping a high continuity of service, hence improving its reliability [3].

Formally, the protective and controlling devices allocation problem consists of finding the best number, locations, and types of devices to improve the reliability and minimize the costs of the system [3]. This problem is classified as a combinatorial optimization problem and there is not a polynomial approach to solve it [3]. Thus, it demands the use of metaheuristics approaches to find good solutions in an acceptable computational time and effort.

Several metaheuristics have been studied and applied to switch allocation problems such as reactive tabu search [2], particle swarm optimization [4], ant colony optimization [5], [6], genetic algorithms [7]–[11], and others. All cited methodologies have obtained success in finding good solutions to the problem, despite the difference between the solutions found.

In genetic algorithms (GA), the main operator accountable for the search strategy of the method is the crossover operator [12]. Crossovers create new solutions from the recombination of selected ones made in the selection step of the GA. There are several ways to recombine solutions, and it is common to compare these approaches in order to find the best methods in terms of results obtained, speed of the convergence, and more. In optimization literature, there are some studies aimed to compare crossover operators in many contexts such as in multi-objective optimization [13], video-game controller automation [14], and also for specific problems, like the job shop scheduling problem [15], travelling salesman problem [16], the vehicle routing problem [17], and the university course timetabling problem [18].

This paper presents an experimental analysis of the uniform, one-point and multi-points crossovers recombination effect in results obtained by a GA applied to the switching devices allocation problem in power distribution systems, a constrained combinatorial optimization problem. For this reason, a more basic version of the genetic algorithm was chosen to keep the analyzes focused on the operators studied. The results obtained show the multi-points crossover in a specific quantity of points can reach better results than other operators, mainly when compared to the one-point crossover, the most common operator utilized in GA for the most applications.

The paper is organized as follows. Section II formalizes the

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

definition and formulation of the switch allocation problem. Section III illustrates the definition of the GA implemented. Section IV presents the crossover operators studied in this paper. Section V shows the used resources in tests and the experimental results. Section VI presents the conclusions drawn from the results in the previous section and points about future works.

II. Problem formulation

This paper considers just the allocation of sectionalizing switches for reliability improvement of networks. The problem was formulated as a constrained optimization problem (COP) [19] where the objective is to minimize the costs around the configuration of allocated switches on the system according to some reliability limit.

A. Graph representation of the distribution systems

The methodology used to model the distribution network was based on modeling used by De Assis et al. [3]. Using graph theory concepts, the network is represented as a directed tree where the vertices represent the load buses except the source vertex (root) that represents the energy substation, and the edges represent the lines between the buses.

Given a distribution network represented by a graph $G = (V, E)$, where V is the set of vertices and E the set of edges, sectionalizing switches can be placed in edges, delimiting disjoint subsets S_i so-called sections, where $S_i \subset V$. All vertices have a number of final customers and a power demand value associated with itself.

B. Reliability indices

A distribution system is said reliable when supply power with a high continuity degree. In order to estimate this degree there are several metrics to measures, among them were chosen the two most used: the System Average Interruption Duration Index (SAIDI) and Energy Not Supplied (ENS) [20].

Both are used to probabilistically measure reliability. SAIDI estimates an average time of interruption persistence per customer. The duration of interruptions is computed using an estimated average time to repair the failures. ENS measures the total energy not supplied during an interruption time and it is also used to measure the financial losses caused by outages. The metrics are defined in (1) and (2).

$$SAIDI = \frac{\sum\limits_{k \in S} T_k N_k}{\sum\limits_{k \in S} N_k}, \tag{1}$$

$$ENS = \sum_{k \in S} T_k L_k \tag{2}$$

where N_k is the number of customers associated to the buses in a section k, L_k is the total power demand of all buses into a section k and T_k is the expected average time of interruptions in section k, given by:

$$T_k = \sum_{l \in S} \left(\sum_{v \in V_l} \lambda_v \right) \cdot t_{kl} \tag{3}$$

In (3), λ_v is a failure rate of each vertex $v \in V$, based on a failure probability on the system by line length (Km) upstream of the vertex v. t_{kl} represents the expected time of an interruption in section k caused by a failure in a section l and it is determined through the average time needed to repair a failure.

C. Objective function

A configuration of sectionalizing switches is represented by a set of decision variables such as they represent the decision to allocate or not a switch on the line immediately upstream to each load bus. It is a minimization problem in which function to be optimized should consider the financial costs to a given configuration restricted to a reliability value. To do this the approach used the acquisition costs of these devices and the estimation of energy not supplied costs calculated through (2).

The configuration reliability is considered as a constraint that uses a pre-defined reference value.

Finally, the mathematical formulation uses constant values for costs such as the electricity cost C_e which is the kWh current cost and the acquisition costs of a sectionalizing switch C_{ss}. The objective function is formulated as:

Min

$$Costs(X) = C_e \cdot ENS(X) + \sum_{x_i \in X} x_i C_{ss}. \tag{4}$$

Subject to:

$$SAIDI(X) \leq SAIDI_{limit} \tag{5a}$$

$$\forall v \in V - V_{src}, \exists k \in S \mid v \in k \tag{5b}$$

$$x_i \in \{0, 1\},\ para\ i = 1, 2, 3, ..., n. \tag{5c}$$

In constraint (5a), $SAIDI_{limit}$ is the maximum reliability value to the solution represented by the set of decision variables X be feasible. The constraint (5b) defines that all vertices from set V except V_{src} (source vertex that represents the substation) should belong to a section from the set of sections S – it defines that should not exist load buses without switch protection. Constraint (5c) just defines that the decision variables were composed of boolean values.

III. The Genetic Algorithm

Genetic Algorithms (GA) are adaptative optimization algorithms that get inspiration from the natural selection process using the concepts of "*survival of the fittest*" [21]. Over the years, it became popular as a method for solving combinatorial optimization problems.

In general, GAs encode solutions for complex problems in data structures called chromosomes, that are treated as individuals from a population in constant evolution. The Darwinian theory of evolution concepts inspires internal algorithms named genetic operators [21] such as the selection, crossover and mutation, generating better solutions.

The GA utilized in this paper is the classic version proposed by Goldberg [21] called "simple genetic algorithm" (SGA).

This version consists of basic genetic operators and was chosen because the focus of the research is investigate the effects of the different crossover operators, therefore, SGA is a good option in relation to more complex algorithm such as more advanced recent versions of GA or hybrid algorithms. A flowchart defining the steps of the genetic algorithm can be seen in the Fig.1.

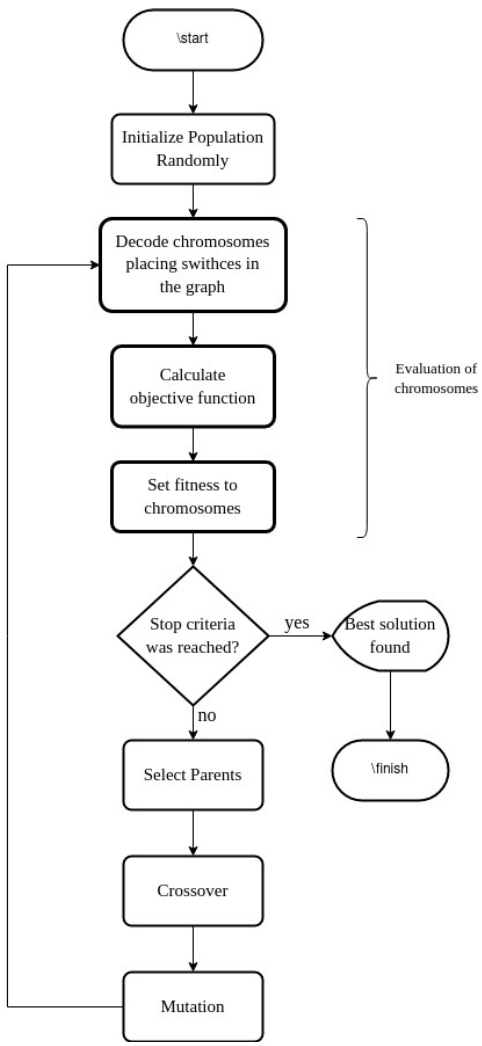


Fig. 1. Flowchart of the Simple Genetic Algorithm

A. Chromosome encoding

The chromosome structure is an important part of a GA. In this paper, chromosomes are encoded using binary encoding, where a chromosome represents an instance of decisions variable shown in (4) and is composed of boolean values where each value is mapped to a vertex on the graph of the distribution system representing the decision to place a sectionalizing switch on the upstream edge of that respective vertex, in this case, 0 (zero) sign that vertex doesn't have a switch placed and 1 (one) sign that vertex has a switch placed in line upstream of it, an illustration is shown in the Fig.2.

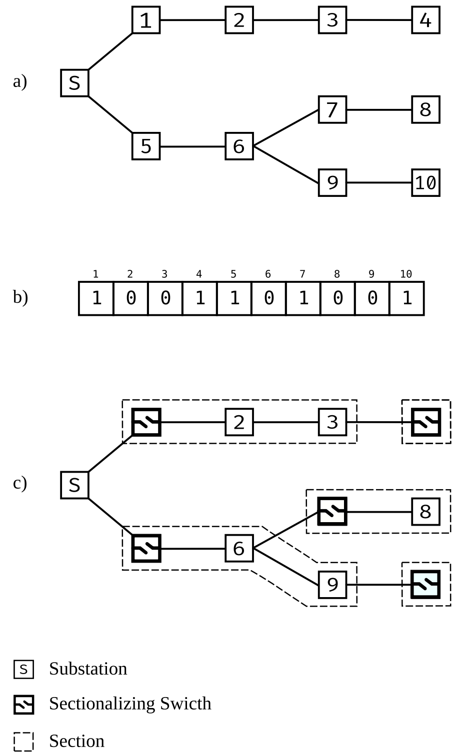


Fig. 2. (a) An example of a small distribution system topology. (b) Example of chromosome applied on topology. (c) The respective solution coded by chromosome on the system.

The population is randomly initialized and evaluated using a fitness function. As the problem consists in minimize the costs, the fitness function is the inverse of the objective function defined in (4).

$$fitness(X) = \frac{1}{Costs(X)} \tag{6}$$

The fitness function (6) sets a value for the chromosome which represents the quality factor of the chromosome as a solution to the problem. The fitness value is also used to choose probabilistically the chromosomes that will be used to generate new solutions for the next generation of the population. The selection method chosen was the deterministic tournament [16], wich consist of select the best chromosome of a random sampling of chromosomes in the population.

The main genetic operators are recombination also called crossover, and mutation, both aim to produce variation in the population over generations. The crossover operator generates new chromosomes through the combination between two parents chromosomes, and mutation operator cause random changes in chromosome with a certain probability rate [16].

One of the most important operators guiding the search process in GA is the crossover operator. How this operator is the main focus of this paper, next subsection will describe the crossover operator in general and the crossovers utilized for the study.

IV. Crossover operators

A feature that sets GAs apart from most metaheuristics techniques is the idea of recombining solutions to generate better solutions. This is one of the key processes that make GAs efficient at solving optimization problems. Many recombination methods have been studied for many different problems in the field of operational research [15]–[18].

The purpose of the crossover operators is generating a pair or more of offspring individuals from the combination of two or more good individuals. In each generation of a GA, the crossover operator acts after the select operation on pairs of individuals with a predefined probability called crossover rate. The crossover operators covered in this paper are the best known and traditional in the literature, the one point crossover, the multi cut points crossover, and the uniform crossover.

A. One-point crossover

It is the most simple crossover operator and the most used in GAs. When this operator is applied, a point is randomly generated and both parent chromosomes are split at this point and the different parts are swapped to generating the offspring chromosomes. For each pair of chromosomes is generated a different point. Fig.3 shows an example of the operation of this crossover.

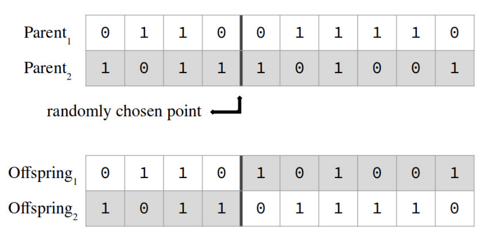


Fig. 3. Example of 1-point crossover operation

Studies indicate that this is the most efficient operator concerning the computational effort [15].

B. Multi-points crossover

Similarly to the one-point crossover operator, the multi-point crossover split the parent chromosomes, however the number of crossover points is increased.

Two points crossover is an instance commonly used in GAs, but other quantities are also used. For example, Fig.4 show an operation with four crossover points.

Theoretical and empirical studies implicate the number of crossover points influences the performance of GA [15]. However, the study performed in this paper concern the convergence of the search process in the algorithm.

C. Uniform crossover

This operator combines two chromosomes selecting randomly the genes that will be swapped between them. A mask can be used to represent the genes that will be swapped in the crossover process like shown in Fig.5. The mask is a sequence of boolean values generated using a uniform probability distribution, values 1 in the mask indicates the genes of the parent chromosomes that will be exchanged in offspring.

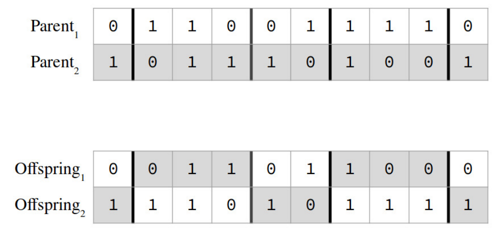


Fig. 4. Example of multi-point crossover operation using 4 crossover points

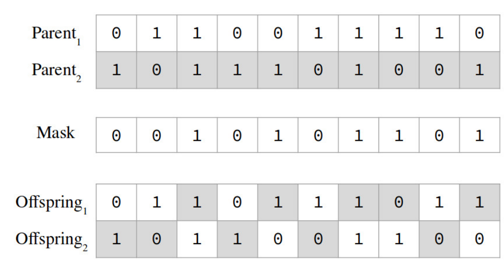


Fig. 5. Uniform crossover example

V. Experiments Design and Results

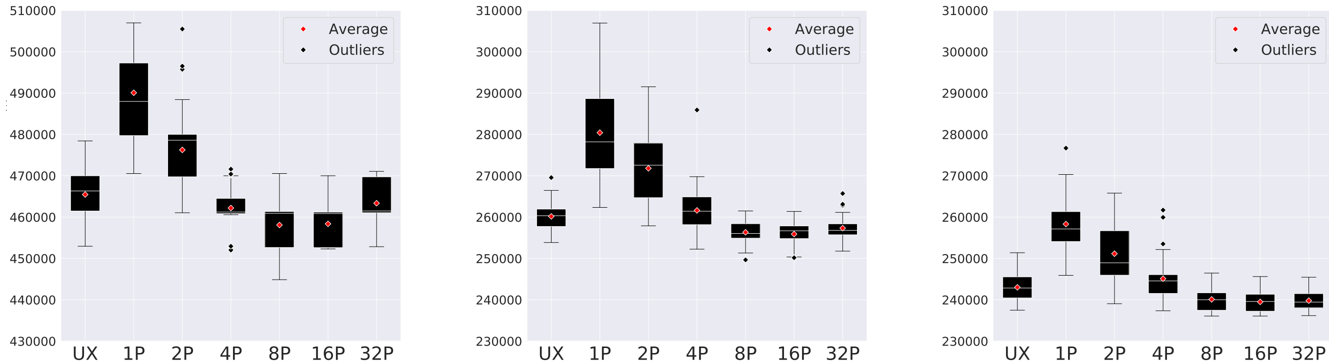
The distribution system used for the experiments is a hypothetical radial system with 135 buses and only one feeder. It was adapted from a test system made available by the Laboratory of Electrical Power System Planning from UNESP Ilha Solteira [22].

The implementations were made using C++ programming language with the Boost's graph library [23] for modeling the graph structures and the ParadisEO framework to evolutionary computation [24], implementing the GA.

The experimental tests were performed in a computer running Arch Linux with version 5.4.2-1 of the Linux kernel, 4GB of RAM and a quad-core Intel® Core™ i3-4005U 64-bit CPU with 1.70 GHz of processing speed. The source code was compiled with version 9.2.0 of the GCC compiler.

A. Definition of test cases

The experiments consider different cases for reliability constraint defined through $SAIDI_{limit}$ value in (5a). Each value for this constraint has a different effect in search space, the lower the $SAIDI_{limit}$ value defined, the fewer feasible solutions in the search space.



$SAIDI_{limit} = 1$ $SAIDI_{limit} = 5$ $SAIDI_{limit} = 10$

Fig. 6. Comparisons of average costs for different restriction degrees situations.

For the choice of $SAIDI_{limit}$ values for the test cases, the GA was executed free of reliability restrictions and the obtained solutions had a reliability index around 10. For this reason, three situations for reliability constraint was chosen classified by restriction degree. $SAIDI_{limit} = 10$ was defined as a value with a low restriction degree, $SAIDI_{limit} = 1$ was chosen as a situation with a high restriction degree, and $SAIDI_{limit} = 5$ was chosen as an intermediary value between previous two.

B. Parameters for experiments

The selection step chosen is the tournament method. A fixed number of generations as stop criteria. The GA had some of its input parameters with fixed values, keeping variation restricted to parameters related to the recombination process. Table I shows the chosen parameters which were fixed and Table II presents the parameters that were isolated in the experimental phase. In Table III is presented the parameters for real-world variables chosen for the simulation. All financial costs values are based in current costs from Belém, a city in the North region of Brazil.

TABLE I
FIXED PARAMETERS IN THE GENETIC ALGORITHM

Parameter	Value
Size of population	400
Number of generations	100
Tournament selection's ring	4
Mutation rate by chromosome	10%
Mutation rate by gene	5%

TABLE II
SETS OF VARIABLE PARAMETERS IN EXPERIMENTS

Parameter	Set of values
$SAIDI_{limit}$	{1, 5, 10}
Crossing rate	{60%, 80%, 100%}

The experiments set consists of a same number of executions for each combination of the parameters defined in Table II. Were performed a total of 40 executions. The parameter variation aims consider situations that influences the recombination process of the GAs, hence, the exploration and convergence process. Data generated through these executions are used to compare the operators through average and variation of obtained results with each given crossover rate.

The crossovers operator used were the uniform, one-point, and multi-points considering an increase in the number of points N according to a geometric progression with $a_1 = 2$ and ratio $r = 2$ in order to have better contrast between the instances of this operator.

C. Experimental results

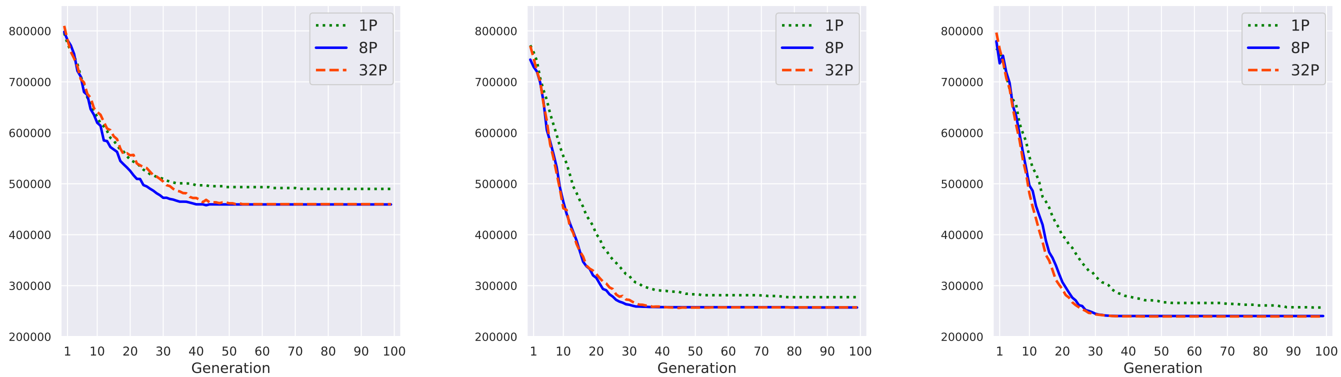
Fig.6 shows the results of the comparisons for each situation considering a crossover rate of 100%. The investigation consists to compare the averages and dispersion of obtained results by GA for each crossover operator in different versions of the search space, defined by restriction degree.

As the different restriction degrees naturally guide to different ranges of financial costs, the three boxplot groups in Fig.6 were kept to the same scale referent to points in the y-axis (financial costs found) to maintain a good comparison quality between the different degrees of restriction considered in the experiments.

In all cases, uniform crossover (UX) had a good average of results. For the one-point (1P) and multi-points (2P, 4P, 8P, 16P, and 32P) operators, as the number of crossover points

TABLE III
REAL-WORLD BASED VARIABLES

Description	Variable	Value
Cost of electricity	C_e	0.599 R$/kWh
Cost of switches	C_{SS}	8.860,50 R$
Failure rate	λ_v	0.18
Expected repair time	t_{kl}	2.2 hours



$SAIDI_{limit} = 1$ $SAIDI_{limit} = 5$ $SAIDI_{limit} = 10$

Fig. 7. Comparison of convergence when using one-point, 8-points and 32-points operator.

increases, the average of the obtained results drops until a particular number of crossover points when the average is increased again. These results suggest a specific number of crossover points where GA has more efficient convergence. For this instance of the problem, the 8-points and the 16-points operators had better results. The numeric results for all the test case may be seen in Table IV.

Moreover, for each one of the restriction degrees covered, a greater contrast was observed on the effect of increasing the number of crossover points. For the case where the restriction degree is higher (i. e. the narrowest search space), the improvements in results are better than in other cases. This suggests that in this constrained optimization problem, the convergence of the algorithm can be influenced by both the number of points and how restricted is the reliability index to be reached.

Despite the changes in the search space motivated by the restriction degree, it is interesting to observe the behaviour of the crossover operators compared are quite similar, and the best configurations were the same of all the three cases analysed.

Observing and aligning the averages of results obtained through the use of each of the cut-points crossovers, it is clear that the increase in the number of points forms a kind of parable. By isolating the "crossover rate" parameter, as shown in Fig.8, it is noted that the restriction condition of the search space also influences the effect that the crossover rate has on the search process. In a less restricted solution space, the variation in crossover rate doesn't cause considerable advantage for a specific crossover operator. However, in a more limited space of solutions, there is a steeper curve between the efficiencies of each operator, showing that in these circumstances there is a great advantage in using a specific number of cut-points, even when compared to uniform crossover.

By observing the convergence of the objective function with multi-points crossover operators as shown in Fig.7, it is possible to note the behavior of the GA when using the best

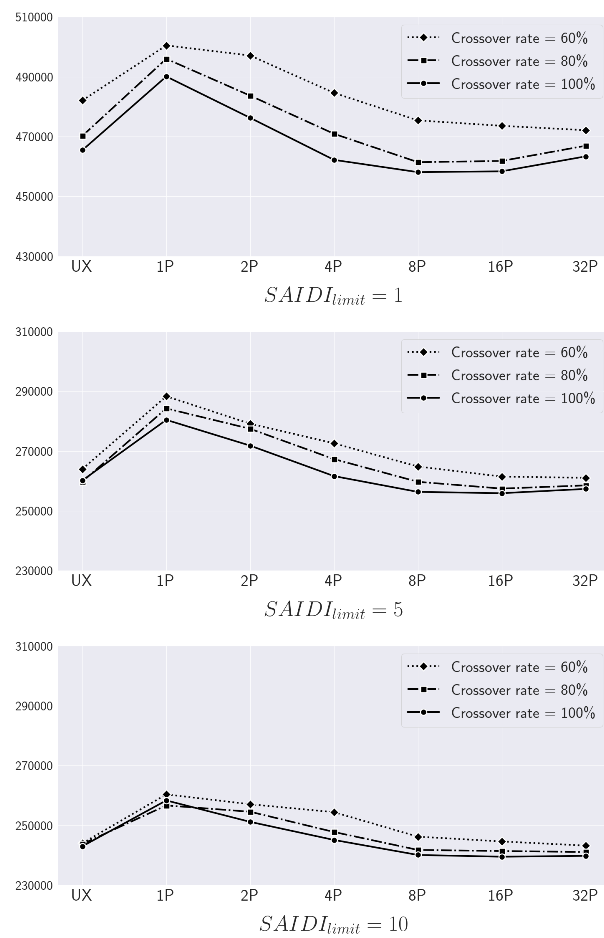


Fig. 8. Effect of variation of crossover rate on the obtained results

operator pointed by the results (8P) is similar to the maximum number of points considered (32P) depending on restriction on the search space. In summary, the narrower the search space, the more premature the 8P convergence is regarding

TABLE IV
NUMERIC RESULTS

		$SAIDI_{limit} = 1$			$SAIDI_{limit} = 5$			$SAIDI_{limit} = 10$		
Crossover Rate	Crossover Operator	Best Result	Average	Std. Deviation	Best Result	Average	Std. Deviation	Best Result	Average	Std. Deviation
60%	UX	463347.7	482084.93	9050.61	253656.2	264004.09	5989.90	237774.0	244035.10	**3728.89**
	1P	470687.4	500463.25	13481.79	269570.8	288302.92	9357.61	245083.2	260352.20	8068.25
	2P	479333.5	497086.81	9896.33	260464.1	279139.03	9854.60	243980.5	257051.57	6981.52
	4P	462391.5	484596.21	11532.79	256992.9	272578.96	8422.23	240444.4	254399.34	8688.31
	8P	461037.9	475399.25	11403.25	250008.4	264827.42	7548.22	236631.8	246214.16	5173.76
	16P	461099.9	473583.97	**7718.97**	251775.0	261450.80	5999.12	236643.1	244618.44	5557.15
	32P	**460856.3**	**472112.82**	9353.89	**249997.6**	**261035.70**	**4304.03**	**236564.1**	**243222.39**	4046.74
80%	UX	453072.8	470239.75	6548.73	249586.7	259818.73	3991.81	236242.2	243748.49	3814.77
	1P	480300.1	495987.04	8885.79	267020.3	284246.11	10685.39	244942.1	256656.94	5863.45
	2P	469746.7	483674.23	9499.13	258268.9	277423.51	8516.63	240873.5	254548.15	6919.02
	4P	460695.3	470912.46	7177.27	253502.9	267300.17	8015.77	237704.2	247771.79	5284.97
	8P	**452127.4**	**461430.50**	6493.64	250929.8	259709.12	4651.36	236631.8	241778.04	3028.33
	16P	452617.5	461835.82	**6024.01**	**249476.6**	**257443.83**	**3304.27**	236178.3	241410.17	3734.99
	32P	452409.7	466914.14	6263.85	250286.6	258504.45	3766.27	**236068.3**	**241072.88**	**2933.71**
100%	UX	452956.8	465463.48	5613.02	253858.0	260178.11	3263.45	237492.3	243015.44	3552.84
	1P	470540.2	490076.29	10598.99	262339.0	280421.68	10931.3	245913.8	258325.02	5982.42
	2P	461060.2	476256.88	11184.21	257896.2	271802.11	8321.42	239054.2	251158.62	6918.84
	4P	**452034.9**	462199.14	5558.81	252250.6	261619.60	5825.91	237348.3	245110.95	5604.80
	8P	444871.1	**458096.82**	5690.53	**249663.3**	256334.77	2715.38	**236068.3**	240098.09	2838.19
	16P	452303.5	458374.42	**5458.30**	250060.9	**255899.37**	2898.87	**236068.3**	**239503.02**	2504.03
	32P	452846.3	463361.64	5461.70	251774.6	257332.99	**2699.57**	236178.3	239798.84	**2062.38**

the increase in the number of points.

Charts shown in Fig.7 were created by running the GA five times, getting the best fitness of each one of 100 generations and, using the mean of fitnesses for each generation. It also shows that the increase in the number of generations not would affect significantly the convergence of the SGA. The operators used were one-point (1P) which has the lower number of points, the 32-points crossover which is the higher number of points considered in this study and, the 8-points crossover operator which has the best results in the comparison shown in Fig.6.

VI. FINAL REMARKS

This paper was presented a comparative study of crossover operators applied to the switches allocation problem in power distribution systems. The results show that multi-points crossover has an ideal number of cut points to solving the current instance of the problem more efficiently, using binary chromosome encoding.

In this study, only one instance of the problem was used for analysis. However, one should keep in mind that the effect of multi-point crossovers is related to the size of the chromosomes to which they are applied. In the present work, chromosomes were coded with the number of power demand bars (135 genes). It is interesting to observe which the values chosen for restriction degrees in the experiments concerning to the chromosomes with 135 genes i. e. there are 2^{135} possible solutions in the search space. In other instances of the switch allocation problem, these same values would can have different effects. For a more accurate analysis, it would be interesting to apply this study to instances of this problem with different sizes.

About the performance of the multi-points crossovers was verified which there is not a significant decrease in the time of processing when the number of points is increased. Concerning the convergence of the searching process, this increase produces an improvement in the optimization depending on the restriction of search space. The presented results around the restriction degree of the problem demand more studies applying to other constrained optimization problems.

As an expansion of this paper, the authors intend to perform similar studies in other constrained optimization problems, preferentially, known problems in the computer science and operational research field. Another proposal is a mathematical formulation for the dynamic definition of the number of points to reach a balance between the convergence of the search and performance.

ACKNOWLEDGEMENT

The authors would like to express their thanks to *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) and *Fundação Amazônia de Amparo a Estudos e Pesquisas* (FAPESPA), two Brazilian research incentive institutions that granted funds to this study.

References

[1] R. E. Brown, "Impact of Smart Grid on Distribution System design," *IEEE Power and Energy Society 2008 General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century, PES*, 2008.

[2] L. G. W. da Silva, R. A. F. Pereira, and J. R. S. Mantovani, "Alocação otimizada de dispositivos de controle e proteção em redes de distribuição," *Sba: Controle & Automação Sociedade Brasileira de Automatica*, vol. 21, no. 3, pp. 294–307, 2010.

[3] L. S. De Assis, J. F. V. González, F. L. Usberti, C. Lyra, C. Cavellucci, and F. J. Von Zuben, "Switch allocation problems in power distribution systems," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 246–253, 2015.

[4] W. Prommee, N. Pongprapunt, and W. Ongsakul, "Improved reliability model and optimal protective device placement in micro grid by improved binary particle swarm optimization," in *8th International Conference on Power Electronics - ECCE Asia*, May 2011, pp. 1514–1519.

[5] H. Falaghi, M. Haghifam, and C. Singh, "Ant colony optimization-based method for placement of sectionalizing switches in distribution networks using a fuzzy multiobjective approach," *IEEE Transactions on Power Delivery*, vol. 24, no. 1, pp. 268–276, Jan 2009.

[6] B. Zhang and P. Crossley, "Reliability improvement using ant colony optimization applied to placement of sectionalizing switches," in *Energy Procedia*, vol. 142. Elsevier Ltd, 2017, pp. 2604–2610.

[7] H. Hashemi, H. Abyaneh, K. Mazlumi, M. Yaghubinia, and M. Zanganeh, "Protective and switching devices allocation according to total cost minimization by genetic algorithm in distribution systems," *ELECO 2011 - 7th International Conference on Electrical and Electronics Engineering*, vol. 7, 01 2011.

[8] H. H. Dezaki, H. A. Abyaneh, K. Mazlumi, M. R. Yaghubinia, and M. Zanganeh, "Protective and switching devices allocation according to total cost minimization by genetic algorithm in distribution systems," *ELECO 2011 - 7th International Conference on Electrical and Electronics Engineering*, no. 424, pp. I–376–I–380, 2011.

[9] H. do Nascimento Alves and R. S. de Sousa, "A Multi-Population Genetic Algorithm to solve multi-objective remote switches allocation problem in distribution networks," in *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*. IEEE, dec 2014, pp. 155–162.

[10] A. P. Caetano, K. Pereira, E. A. Amorim, and E. M. Carreno, "Switch and tie lines allocation in distribution systems using NSGA II," *Proceedings of the 2014 IEEE Central America and Panama Convention, CONCAPAN 2014*, no. Concapan Xxxiv, 2014.

[11] L. Zemite, M. Gorobecs, A. Smats, A. Jasevics, and A. Levchenkovs, "Genetic algorithm for optimization of power switch allocation in distribution network," in *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, 2017, pp. 1–6.

[12] C. R. Reeves, "Genetic algorithms," in *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds. Springer, 2010, pp. 109–139.

[13] K. Sekine and T. Tatsukawa, "A parametric study of crossover operators in multi-objective evolutionary algorithm," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 1196–1203.

[14] G. K. Soon, T. T. Guan, C. K. On, R. Alfred, and P. Anthony, "A comparison on the performance of crossover techniques in video game," in *2013 IEEE International Conference on Control System, Computing and Engineering*, 2013, pp. 493–498.

[15] J. Magalhães-Mendes, "A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem," *WSEAS Transactions on Computers*, vol. 12, no. 4, pp. 164–173, 2013.

[16] A. Bala and A. K. Sharma, "A comparative study of modified crossover operators," *Proceedings of 2015 3rd International Conference on Image Information Processing, ICIIP 2015*, pp. 281–284, 2016.

[17] T. M. Stehling and S. R. de Souza, "A comparison of crossover operators applied to the vehicle routing problem with time window," in *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, Oct 2017, pp. 300–305.

[18] W. Chinnasri, S. Krootjohn, and N. Sureerattanan, "Performance comparison of genetic algorithm's crossover operators on university course timetabling problem," in *2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*, vol. 2, April 2012, pp. 781–786.

[19] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed. Springer Publishing Company, Incorporated, 2015.

[20] R. Billinton and R. N. Allan, *Realibility Evaluation of Power Systems*, 2nd ed. Springer Science, 1996.

[21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[22] "Laboratory of electrical power system planning," https://www.feis.unesp.br/#!/departamentos/engenharia-eletrica/pesquisas-e-projetos/lapsee/english/downloads/testing-systems/, accessed: 2019-06-30.

[23] J. Siek, L.-Q. Lee, and A. Lumsdaine, *The Boost graph library: A Guide and Reference Manual*. Addison-Wesley, 2002.

[24] S. Cahon, N. Melab, and E. G. Talbi, "ParadisEO: A framework for the reusable design of parallel and distributed metaheuristics," *Journal of Heuristics*, vol. 10, no. 3, pp. 357–380, 2004.