Optimizing survivability of vulnerable series–parallel multi-state systems

Gregory Levitin*, Anatoly Lisnianski
Department of Reliability and Equipment, The Israel Electric Corporation Ltd, Bait Amir, P.O. Box 10, Haifa 31000, Israel
Received 7 September 2002; accepted 14 October 2002

Abstract

In this paper we consider vulnerable systems which can have different states corresponding to different combinations of available elements composing the system. Each state can be characterized by a system performance rate, which is the quantitative measure of a system’s ability to perform its task. Both the impact of external factors (attack) and internal causes (failures) affect system survivability which is determined as probability of meeting a given demand.

We formulate the problem of finding structure of series–parallel multi-state system (including choice of system elements, their separation and protection) in order to achieve a desired level of system survivability by the minimal cost.

An algorithm based on the universal generating function method is suggested for determination of the vulnerable series–parallel multi-state system survivability. A genetic algorithm is used as optimization tool in order to solve the structure optimization problem.

Keywords: System survivability; Multi-state systems; Separation of elements; Universal generating function; Genetic algorithm

1. Introduction

Survivability, the ability of a system to tolerate intentional attacks or accidental failures or errors, is becoming especially important when a system operates in battle conditions or is affected by a corrosive medium or other hostile environment.

A survivable system is one that is able to 'complete its mission in a timely manner, even if significant portions are incapacitated by attack or accident' [1]. This definition presumes two important things:

First, both the impact of external factors (attack) and internal causes (failures) affect system survivability. Therefore, as was stated in Ref. [2], it is important to take into account the influence of reliability (availability) of system elements on the entire system survivability.

Second, a system can have different states corresponding to different combinations of failed or damaged elements composing the system. Each state can be characterized by a system performance rate, which is the quantitative measure of a system’s ability to perform its task [2–5]. For example, in Refs. [2,5] each system state is characterized by an available ship propulsion power or by an available electric power respectively. Therefore a system should be considered a multi-state one when its survivability is analyzed.

When applied to multi-state systems, mission success depends on a system’s ability to meet demand (required performance level). In this case, the outage effect will be essentially different for units with different nominal capacities and will also depend on demand. Therefore, the performance rates (productivity or capacity) of system elements should be taken into account as well as the level of demand when the entire system’s survivability is estimated.

In some studies [6,7] that focus on the survivability of communication networks, the survivability is treated as a function of the network connectivity. Probabilistic techniques usually associate the connectivity level of the network with the reliability (availability) of communication paths between specific network nodes. The failures of nodes and links and the impact of external factors are not distinguished in these works.
Rates may be considered as a problem of system structure optimization. The redundancy optimization reliability constraints, is well known as the redundancy problem of total investment cost minimization, subject to engineers try to achieve this level with minimal cost. The (availability), redundant elements are included. Usually performance rates.

formulation) and do not take into account element k common cause failure approach [8–17] . All these studies of external factors on the system survivability based on numerous studies were devoted to estimating the impact on the system survivability based on common cause failure approach [8–17]. All these studies consider systems with identical elements (k-out-of-n formulation) and do not take into account element performance rates.

To provide a required level of system reliability (availability), redundant elements are included. Usually engineers try to achieve this level with minimal cost. The problem of total investment cost minimization, subject to reliability constraints, is well known as the redundancy optimization problem. The redundancy optimization problem for a system with different element performance rates may be considered as a problem of system structure optimization. This problem was introduced in Ref. [18], where the general MSS optimization approach was formulated.

One of the ways to enhance system survivability is to separate elements with the same functionality (parallel elements). Adding more parallel elements will improve a MSS availability but will not be effective from a vulnerability standpoint without sufficient separation between elements [2]. The separation can be performed by spatial dispersion, by encapsulating different elements into different protective casings, etc.

Parallel elements not separated from one another are considered to belong to the same protection group. All elements belonging to the same protection group can be
destroyed by the same impact while at least all the elements belonging to \( N - 1 \) different protection groups out of \( N \) will survive a single impact. Obviously, separation has its price. Allocating all the parallel elements together (within a single protection group) is usually cheaper than separating them. The separation usually requires additional areas, constructions, communications, etc. Moreover, each separated group can be intentionally protected against the external impact, which requires additional investments. There can be different levels of protection achieved that are characterized by different efficiency (vulnerability of the group) and cost.

Since system elements with the same functionality can have different performance rates and different availability, the choice of elements to be included into the system strongly affects system survivability. Other factors that strongly affect system survivability are the partitioning of elements into protection groups and the choice of protection levels for each separated group.

In this paper we formulate the problem of finding structure of series–parallel MSS (including choice of system elements, their separation and protection) in order to achieve a desired level of system survivability by the minimal cost. To solve the problem we extend an algorithm for MSS structure optimization suggested in Refs. [19,20]. This algorithm finds the optimal system structure by choosing the appropriate product (version of a system element) from the list of available products for each type of equipment. Each product is characterized by its capacity, reliability and price. The objective is to minimize the total cost of the system subject to the requirement of meeting the demand with the desired level of reliability. This approach allows the reliability engineer to solve practical problems in which a variety of products exist and in which analytical dependencies are unavailable for the cost of system components.

In accordance with Refs. [2,21] we define MSS mission success as its ability to meet a demand \( W \) and measure the system survivability as

\[
S_{\text{MSS}}(W) = \Pr\{G_{\text{MSS}} \geq W\},
\]

where \( G_{\text{MSS}} \) is output performance of the MSS.

For MSS which have a finite number of states there can be \( K \) different levels of output performance: \( G_{\text{MSS}} \in G = \{G_k, 1 \leq k \leq K\} \) and system output performance distribution (OPD) can be defined by two finite vectors \( G \) and \( q = \{q_k\} \), where \( q_k = \Pr\{G = G_k\} \), \( 1 \leq k \leq K \). Therefore following Ref. [19] we can define MSS survivability as the probability that a system remains in those states in which \( G_k \geq W \):

\[
S_{\text{MSS}}(W) = \sum_{G_k \geq W} q_k.
\]

A method for evaluating the reliability (availability) of series–parallel MSS consisting of elements with different performance rates was suggested in Ref. [22]. This method based on universal generating functions (UGF) proved to be convenient for numeric implementation and effective at solving problems of MSS redundancy [19,20,23–25] and maintenance [26–28] optimization, as well as importance analysis [29]. Unlike fault-tree analysis, the UGF method provides for the possibility of treating systems with similar topologies but with different nature of elements interaction in a similar way.

In this work we use the UGF method to evaluate the survivability of MSS with a given structure, separation and protection of the elements. To find the optimal solution a genetic algorithm (GA) is used which is based on principles of evolution. The solution encoding technique of GA is adapted to represent MSS element separation.

Section 2 of the paper consists of a general description of the series–parallel MSS model used and a formulation of the survivability optimization problem. Section 3 describes the technique used for evaluating the entire MSS survivability index (1) while the performance rate and reliability of its elements are given, as well as the partition of the elements between the protection groups and vulnerability of these groups. Section 4 describes the optimization approach used and its adaptation to the problem formulated. Section 5 contains examples in which the best-found solutions for a series–parallel system are demonstrated.

2. Model description and problem formulation

Consider a system consisting of \( M \) components connected in series. Each component contains elements connected in parallel (Fig. 5). Different versions and number of elements may be chosen for any given system component. Elements are characterized by their availability, nominal performance rate and cost, according to their version. All MSS elements are elements with total failures, i.e. they have two states: functioning with nominal performance and failure. The failures of MSS elements are mutually statistically independent. In order to survive, the MSS should meet a demand \( W \).

For each component \( m \), there exist a list of \( H_m \) different versions of available elements. A vector of parameters \( g_m(h), A_m(h), e_m(h) \) can be specified for each version \( h \) of element of type \( m \). The structure of system component \( m \) is defined, therefore, by a vector containing numbers of versions of elements chosen for the component \( h_m = \{h_{m1}, \ldots, h_{mM_m}\} \), where \( h_{m} \in \{0, 1, \ldots, H_m\} \). Note that including a dummy version 0 corresponding to the absence of elements allows one to represent a different number of elements included in component \( m \) by vectors \( h_m \) of the same length \( E_m \). The total cost of elements chosen for the \( m \)th component is

\[
C_m = \sum_{i=1}^{E_m} e_m(h_{mi}).
\]

The elements belonging to component \( m \) can be separated into \( x \) independent protection groups, where \( x \) can vary from
l (all the elements are gathered together) to $E_m$ (all the elements are separated one from another). For each protection group $j$ within component $m$ different levels of protection can be chosen $\gamma_{mj} \in \{1, \ldots, \Gamma_m\}$. It is assumed that all the elements of the $m$th component belonging to the same protection group $j$ can be destroyed by the impact of an enemy or a corrosive medium with probability $v_m(\gamma_{mj})$, which depends on the chosen protection level $\gamma_{mj}$ and characterizes the group vulnerability. The failures of MSS elements and external impacts are considered to be independent events.

The cost of each protective group depends on the type of elements protected (number of component $m$), on the number of elements belonging to the group and on the chosen protection level $\gamma_{mj}$.

The elements separation problem for each component $m$ can be considered as a problem of partitioning a set $E_m$ into a collection of $E_m$ mutually disjoint subsets $\Phi_{mi}$, i.e. such that

$$
\bigcup_{i=1}^{E_m} \Phi_{mi} = \Phi_m,
$$

(4)

$$
\Phi_{mi} \cap \Phi_{mj} = \emptyset, \quad i \neq j.
$$

(5)

Each set can contain from 0 to $E_m$ elements. The partition of the set $\Phi_m$ can be represented by the vector $r_{mj} = \{r_{mj}, 1 \leq j \leq E_m\}$, where $r_{mj}$ is the number of the subset to which element $j$ belongs. One can easily obtain the cardinality of each subset $\Phi_{mi}$ as

$$
n(m, i) = |\Phi_{mi}| = \sum_{j=1}^{E_m} f(r_{mj}, i),
$$

(6)

where

$$
f(a, b) = \begin{cases} 1, & a = b \\
0, & a \neq b \end{cases}
$$

The cost of element separation and protection for the component $m$ determined by the pair of vectors $r_m$, $\gamma_m$ can be calculated as

$$
C_{\text{sep}}^m = \sum_{i=1}^{E_m} c_m(n(m, i), \gamma_m),
$$

(7)

where $c_m(0, \gamma_m) = 0$ by definition.

Concatenation of vectors $h_m$, $r_m$ and $\gamma_m$ for $1 \leq m \leq M$ determines the structure of the entire system. The total MSS cost can be determined as

$$
C_{\text{MSS}}(h, r, \gamma) = \sum_{m=1}^{M} [C_{\text{sep}}^m + C_{\text{pro}}^m]
$$

$$
= \sum_{m=1}^{M} \sum_{i=1}^{E_m} [c_m(n(m, i), \gamma_m) + e_m(h_{mi})].
$$

(8)

where $h = \{h_1, \ldots, h_M\}$, $r = \{r_1, \ldots, r_M\}$ and $\gamma = \{\gamma_1, \ldots, \gamma_M\}$.

Now we can formulate the optimal separation problem as follows. Find vectors $h$, $r$ and $\gamma$ that provide the desired MSS survivability $S_{\text{MSS}}$ with the minimal cost:

$$
(h, r, \gamma) = \arg\{C_{\text{MSS}}(h, r, \gamma) \rightarrow \min |S_{\text{MSS}}(h, r, \gamma) \geq S^*\}. \quad (9)
$$

3. MSS reliability estimation based on a universal generating function

To estimate the survivability of MSS with given structure, separation and protection of elements $S_{\text{MSS}}(h, r, \gamma)$ one has to apply a procedure which calculates the performance distribution of a given series–parallel structure consisting of elements with given performance rate and availability.

The procedure used in this paper for MSS OPD and survivability evaluation is based on the universal $z$-transform (also called $u$-function or universal generating function) technique [22]. A detailed description of UGF applied to MSS reliability estimation is presented in Ref. [19]. A brief introduction to the technique is given here.

The UGF ($u$-transform) of a discrete random variable $X$ is defined as a polynomial

$$
U(z) = \sum_{k=1}^{K} q_k z^{x_k},
$$

(10)

where the variable $X$ has $K$ possible values and $q_k$ is the probability that $X$ is equal to $x_k$.

To evaluate the probability that the random variable $X$ exceeds the value $w$ the coefficients of polynomial $U(z)$ should be summed for every term with $x_k \geq w$:

$$
\Pr(X \geq w) = \sum_{x_k \geq w} q_k.
$$

This can be done using the following $\delta$ operator over $U(z)$:

$$
\delta(U(z), w) = \delta \left( \sum_{k=1}^{K} q_k z^{x_k}, w \right) = \sum_{k=1}^{K} \delta(q_k z^{x_k}, w),
$$

(11)

where for individual term $q_k z^{x_k}$:

$$
\delta(q_k z^{x_k}, w) = \begin{cases} q_k, & x_k \geq w \\
0, & x_k < w \end{cases}
$$

In our case, the polynomial $U(z)$ can define performance distributions, i.e. it represents all the possible states of the system (or element) by relating the probabilities of each state $q_k$ to the performance $G_k$ of the system in that state. Note that the system OPD defined by the vectors $\{G_k, 1 \leq k \leq K\}$ and $\{q_k, 1 \leq k \leq K\}$ can now be represented as

$$
U(z) = \sum_{k=1}^{K} q_k z^{G_k}.
$$

(12)
Consider single element with total failures, which is characterized by nominal performance rate \( g \) and availability \( A \),
\[
\Pr(X = g) = A,
\]
\[
\Pr(X = 0) = 1 - A.
\]
The \( u \)-function of such an element has only two terms and can be defined as
\[
U(z) = (1 - A)z^0 + Az^1,
\]
(13)
which corresponds to the following vectors of element performance distribution: \( \{0, g\} \) and \( \{(1 - A), A\} \).

To obtain the \( u \)-function of a subsystem (component) containing a number of elements, composition operators are introduced. These operators determine the polynomial \( U(z) \) for a group of elements connected in parallel and in series, respectively, using simple algebraic operations on the individual \( u \)-functions of elements; in some cases composition operators can be developed for structures with more complex topologies, such as bridges [23,24]. All the composition operators take the form
\[
\Omega_u(U_1(z), U_2(z)) = \Omega_u \left[ \sum_{i=1}^{l} p_i z^{\phi_i} \right] \sum_{j=1}^{J} \phi_j z^{\delta_j}
\]
\[
= \sum_{i=1}^{l} \sum_{j=1}^{J} p_i \phi_j z^{\mu(i,j)}
\]
(14)
and satisfy the following conditions:
\[
\Omega_u\{ U_1(z), \ldots, U_k(z), U_{k+1}(z), \ldots, U_n(z) \}
\]
\[
= \Omega_u\{ U_1(z), \ldots, U_k(z), U_{k+1}(z), \ldots, U_n(z) \}
\]
(15)
\[
\Omega_u\{ U_1(z), \ldots, U_k(z), U_{k+1}(z), \ldots, U_n(z) \}
\]
\[
= \Omega_u\{ \Omega_u\{ U_1(z), \ldots, U_k(z) \}, \Omega_u\{ U_{k+1}(z), \ldots, U_n(z) \} \}
\]
for arbitrary \( k \).

The function \( \omega(\cdot) \) in composition operators expresses the entire performance rate of a subsystem consisting of two elements connected in parallel or in series in terms of the individual performance rates of the elements. The definition of the function \( \omega(\cdot) \) strictly depends on the physical nature of system performance measure and on the nature of the interaction of system elements. In Ref. [19] two types of MSS are considered. For the sake of simplicity we consider here only those MSS in which performance measure is defined as productivity or capacity (continuous materials or energy transmission systems, manufacturing systems, power supply systems). To apply the suggested method to other types of MSS one has only to choose the corresponding functions \( \omega(\cdot) \) [19,26,29].

In MSS of considered type the total capacity of elements connected in parallel is equal to the sum of the capacities of the elements. Therefore, for a pair of elements connected in parallel
\[
\omega(e,f) = \pi(e,f) = e + f.
\]
(16)
The \( u \)-function \( \bar{U} \) of a group of \( n \) parallel elements with their individual \( u \)-functions \( U_j(z) \) defined in Eq. (13) can be obtained as a product of polynomials:
\[
\bar{U}(z) = \Omega_u(U_1(z), \ldots, U_n(z)) = \prod_{j=1}^{n} U_j(z)
\]
\[
= \prod_{j=1}^{n} \left[ 1 - A_j + A_j z^0 \right].
\]
(17)
When the components (elements) are connected in series, the element with the lowest performance rate becomes the bottleneck of the system. Therefore for a pair of elements connected in series
\[
\omega(e,f) = \sigma(e,f) = \min(e,f).
\]
(18)
For a system consisting of \( N \) components (elements) with \( u \)-functions \( U_j(z) \) connected in series, the \( u \)-function of the entire system is defined as
\[
\Omega_u(\bar{U}_1, \ldots, \bar{U}_N).
\]

When some elements of component \( m \) are combined within a single protection group with vulnerability \( v_m \), this means that all these elements can be destroyed with the same probability \( v_m \). Probabilities of all the states in which the group has nonzero performance rates should be multiplied by \( (1 - v_m) \). Indeed, one can consider a combination of parallel elements within a protection group as the addition of an element with performance distribution
\[
Pr(g = \infty) = 1 - v_m, \quad Pr(g = 0) = v_m
\]
(19)
connected in series to the group. Such an addition reflects the fact that the output performance rate of the group will be zeroed with probability \( v_m \) and will not be changed with probability \( 1 - v_m \). If a group of parallel elements with \( u \)-function \( \bar{U} \) is combined within a single protection group, the \( u \)-function of the group can be determined using the \( \Omega_u \) operator over \( \bar{U} \) and \( u \)-function representing the performance distribution (19). The following \( \varphi \) operator obtains the \( u \)-function of the protection group
\[
\varphi(\bar{U}(z)) = \Omega_u(\bar{U}(z), (1 - v_m)z^0 + v_m z^0)
\]
\[
= \Omega_u \left( \sum_{i=1}^{I} \alpha_i z^\beta_i (1 - v_m)z^\alpha_i + v_m z^0 \right)
\]
\[
= (1 - v_m) \sum_{i=1}^{I} \alpha_i z^\min(\beta_i, \alpha_i) + v_m \sum_{i=0}^{I} \alpha_i z^\min(\beta_i, 0)
\]
\[
= (1 - v_m)\bar{U}(z) + v_m z^0.
\]
(20)

Each \( j \)-th protection group belonging to MSS component \( m \), which contains elements \( j_1, \ldots, j_{n(m,j)} \), can be rep-
Applying composition operators $E_m$ protection groups is

$$U_m(z) = \Omega_{\pi}[\varphi(U_m(z)), \ldots, \varphi(U_{mE_m}(z))].$$

(22)

Applying composition operators $\Omega_\sigma$ one can obtain the $u$-function of the entire MSS in the form (Eq. (10)):

$$U_{\text{MSS}}(z) = \Omega_{\sigma}[\hat{U}_1(z), \ldots, \hat{U}_K(z)].$$

(23)

The sequence of using the operators $\Omega_\pi, \Omega_\sigma$ and $\varphi$ for obtaining the $u$-function of series–parallel system is presented in Fig. 1.

After obtaining this $u$-function one actually has two vectors $\mathbf{G}$ and $\mathbf{q}$ of OPD; the simple numerical example of using the universal $z$-transform for MSS OPD evaluation can be found in Ref. [29]). Using the operator $\delta$ (Eq. (11)) for the given demand $W$ one can evaluate MSS survivability (Eq. (2)).

4. Optimization technique

Eq. (9) formulates a complicated combinatorial optimization problem. An exhaustive examination of all possible solutions is not realistic, considering reasonable time limitations. As in most combinatorial optimization problems, the quality of a given solution is the only information available during the search for the optimal solution. Therefore, a heuristic search algorithm is needed which uses only estimates of solution quality and which does not require derivative information to determine the next direction of the search.

The recently developed family of GAs is based on the simple principle of evolutionary search in solution space. GAs have been proven to be effective optimization tools for a large number of applications. Successful applications of GAs in reliability engineering are reported in Refs. [19,20, 23–28,30–44].

It is recognized that GAs have the theoretical property of global convergence [45]. Despite the fact that their convergence reliability and convergence velocity are contradictory, for most practical, moderately sized combinatorial problems, the proper choice of GA parameters allows solutions close enough to the optimal one to be obtained in a short time.

4.1. Genetic algorithm

Basic notions of GAs are originally inspired by biological genetics. GAs operate with ‘chromosomal’ representation of solutions, where crossover, mutation and selection procedures are applied. Unlike various constructive optimization algorithms that use sophisticated methods to obtain a good singular solution, the GA deals with a set of solutions (population) and tends to manipulate each solution in the simplest manner. Chromosomal representation requires the solution to be coded as a finite length string.

A brief introduction to GAs is presented in Ref. [46]. More detailed information on GAs can be found in Goldberg’s comprehensive book [47], and recent developments in GA theory and practice can be found in books [37, 38]. The basic structure of the version of GA referred to as GENITOR [48] is as follows.

An initial population of $N_s$ solutions (strings) is generated at random. Within this population, new solutions are obtained during the genetic cycle by using the crossover operator. This operator produces an offspring from a randomly selected pair of parent solutions. The newly obtained offspring undergoes mutation with the probability $P_m$. In our GA, the mutation procedure adds or subtracts (with equal probability) 1 from the value of randomly chosen element of the solution encoding string.

Each new solution is decoded and its objective function (fitness) values are estimated. These values, which are a measure of quality, are used to compare different solutions. The comparison is accomplished by a selection procedure that determines which solution is better: the newly obtained solution or the worst solution in the population. The better solution joins the population, while the other is discarded. If the population contains equivalent solutions following selection, redundancies are eliminated and the population size decreases as a result.

After new solutions are produced $N_{\text{rep}}$ times, new randomly constructed solutions are generated to replenish the shrunken population, and a new genetic cycle begins.

The GA is terminated after $N_c$ genetic cycles. The final population contains the best solution achieved. It also contains different near-optimal solutions which may be of interest in the decision-making process.

To apply the GA to a specific problem, a solution representation and decoding procedure must be defined as well as basic GA procedures and parameters.
4.2. Solution representation

As described in Section 2, three things determine the structure of MSS:

- the list of version numbers of elements chosen for each component;
- separation of elements between protection groups within each component;
- levels of protection for each protection group.

Consider substring \( Q_m = \{u_{m1}, \ldots, u_{mF}\} \), corresponding to \( m \)th system component. Let all the elements of the substring belong to the range \((0, H_m + G_m)\). We use the following rules to encode the structure of \( m \)th MSS component using the substring (see Fig. 2).

- If \( \theta_{mj} \leq H_m \), \( \theta_{mj} \) determines the version of element included in the component: \( h = \theta_{mj} (\theta_{mj} = 0 \text{ means that no element is included}) \).
- If \( \theta_{mj} > H_m \), \( \theta_{mj} \) constitutes a separator and determines the protection level of the group of elements: \( \gamma = \theta_{mj} - H_m \).
- The adjacent string elements located between the beginning of the string and separator or between two separators are considered to belong to the same protection group with the protection level determined by the right separator.
- The last element of each substring \( \theta_{mF} \) is always considered to be a separator. The corresponding protection level is determined as \( \gamma = \text{mod}_{H_m + G_m} (\theta_{mF}) + 1 \).

One can see that in order to determine an arbitrary structure of component \( m \) with up to \( E_m \) elements, one has to use substring \( \Theta_m \) of length \( F = 2E_m \) (for the case when all the elements are separated). The string of this length, corresponding to exactly \( E_m \) elements gathered within single protection group, contains one separator (\( \theta_{mF} \)) and \( E_m - 1 \) zeros.

Concatenating substrings \( \Theta_m \) for \( m = 1, \ldots, M \), one obtains the string \( \Theta \) which determines the structure of the entire MSS. In order to allow all the string elements distributed within the same range to represent feasible solutions, we determine this range as \((0, \max_{1 \leq m \leq M} H_m + \max_{1 \leq m \leq M} G_m)\).

When the string is decoded, we transform each string element \( \theta_{mj} \), corresponding to \( m \)th component in the following way:

\[
\theta_{mj} = \text{mod}_{H_m + G_m + 1} (\theta_{mj})
\]

and apply rules 1–4 to the obtained value of \( \theta_{mj} \). Transform (24) just ensures that each \( \theta_{mj} \) belongs to the range \((0, H_m + G_m)\). The unification of the distribution range of all the string elements simplifies the string generation procedure, as well as mutation and crossover operators.

Consider, for example, an MSS with \( M = 2, H_1 = 3, G_1 = 2, H_2 = 4, G_2 = 4 \). Let \( F = 10 \). The string of integers generated in the range \((0, 8)\) :

\[
4\ 8\ 1\ 0\ 5\ 6\ 3\ 0\ 1\ 0\ 3\ 0\ 8\ 7\ 1\ 5\ 4\ 2\ 1\ 6
\]

after transformation takes the form:

\[
4\ 2\ 1\ 0\ 5\ 0\ 3\ 0\ 1\ 0\ 3\ 0\ 8\ 7\ 1\ 5\ 4\ 2\ 1\ 6
\]

After decoding the string using rules 1–4, we obtain (separators are underlined and their values are replaced with corresponding values of protection levels):

\[
1\ 2\ 1\ 0\ 1\ 2\ 0\ 3\ 0\ 1\ 1\ 3\ 0\ 4\ 3\ 1\ 1\ 4\ 2\ 1\ 3\ 3
\]

which corresponds to the structure presented in Table 1. One can see that the string determines two empty groups for

<table>
<thead>
<tr>
<th>No. of component</th>
<th>No. of group</th>
<th>Versions of elements belonging to the group</th>
<th>Protection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>–</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2, 1</td>
<td>2, 1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3, 1</td>
<td>3, 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1, 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4, 2, 1</td>
<td>4, 2, 1</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1
Example of MSS structure
which protection level values have no meaning and should not be considered.

4.3. Solution decoding procedure

The following procedure determines the fitness value for an arbitrary solution defined by integer string \( \Theta \).

1. Assign 1 to the number of component \( m \). Assign 0 to the total cost \( C_{\text{MSS}} \).
2. Decode substring \( \Theta_m \), containing elements \( \theta_{(m-1)F+1} \) to \( \theta_{mF} \) of string \( \Theta \) as described in Section 4.2, to obtain versions of elements belonging to different groups and corresponding protection levels.
3. For each nonempty group, determine parameters \( g(h), A(h) \) and \( e(h) \) of elements it contains in accordance with their versions \( h \) and define \( u(z) \) functions of these elements using Eq. (13). Determine \( \bar{U}(z) \) for the entire group using \( \Omega_{\pi} \) operator (Eq. (17)). Determine group vulnerability \( v_m \) as function of group size and protection level and apply the corresponding operator \( \varphi \) (Eq. (20)) to \( \bar{U}(z) \).
4. Obtain the \( u \)-function of the \( m \)th component using \( \Omega_{\pi} \) operator (Eq. (22)).
5. Determine the cost of \( m \)th component in accordance with Eqs. (3) and (7) and add this value to \( C_{\text{MSS}} \).
6. Increment \( m \) and if \( m \leq M \) return to step 2.
7. Applying \( \Omega_{\pi} \) operator (Eq. (23)), obtain the \( u \)-function of the entire MSS and evaluate the system survivability \( S_{\text{MSS}} \) using operator \( \delta(U_{\text{MSS}}(z), W) \) (Eq. (11)) for the given demand \( W \).
8. So that the GA will search for the solution with minimal total cost and with survivability not less than the required value \( S^* \), evaluate the solution quality (fitness) \( \Lambda \) as follows:

\[
\Lambda = A^* - C_{\text{MSS}} - \lambda \min(S^* - S_{\text{MSS}}, 0), \tag{25}
\]

where \( \lambda \) is a sufficiently large penalty and \( A^* \) is a sufficiently large constant value used to keep \( \Lambda \) positive. Note that for solutions with \( S_{\text{MSS}} \geq S^* \), the fitness of the solution depends only on the system cost.

4.4. Crossover procedures

The crossover operator is aimed at producing a new solution (string) which inherits some properties of both parent solutions by combining parts of their strings.

Two different traditionally used crossover procedures [49] were tested for the algorithm:

Two-point (fragment) (F) crossover- in which elements belonging to the fragment defined by two randomly chosen crossover sites are copied from the first parent and elements located out of the fragment, from the second parent.

Uniform (U) crossover in which each element is copied either from the first parent or from the second one with equal probability. The crossover sites have no sense in this operator, but it produces on the average \( L/2 \) crossings on a string of length \( L \).

The following example presents two different offspring strings, obtained by these procedures:

first parent: 305100210110002
second parent: 001100210340130
U-crossover offspring:

Table 2
Parameters of available elements

<table>
<thead>
<tr>
<th>No. of component</th>
<th>No. of version</th>
<th>g</th>
<th>A</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.2</td>
<td>0.97</td>
<td>3.1</td>
</tr>
<tr>
<td>2</td>
<td>1.6</td>
<td>4.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.8</td>
<td>4.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.0</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5.0</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5.0</td>
<td>14.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

v

Table 3
Characteristics of available protection levels

<table>
<thead>
<tr>
<th>No. of component</th>
<th>Type of component</th>
<th>Protection level</th>
<th>Protection description</th>
<th>Vulnerability, v</th>
<th>Cost, ε</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transformers</td>
<td>1</td>
<td>Outdoor location</td>
<td>0.35</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Indoor location</td>
<td>0.15</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Underground</td>
<td>0.05</td>
<td>15.7</td>
</tr>
<tr>
<td>2</td>
<td>Capacitors</td>
<td>1</td>
<td>Outdoor location</td>
<td>0.01</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>Lines</td>
<td>1</td>
<td>Overhead</td>
<td>0.60</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Overhead insulated</td>
<td>0.35</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Underground</td>
<td>0.15</td>
<td>17.0</td>
</tr>
<tr>
<td>4</td>
<td>Commutation blocks</td>
<td>1</td>
<td>Outdoor location</td>
<td>0.10</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Indoor location</td>
<td>0.03</td>
<td>4.2</td>
</tr>
</tbody>
</table>

1. The average relative distance from the best solution

\[
I_x = 0.1 \frac{1}{10} \sum_{i=1}^{10} \frac{A_B - A_m}{A_B}, \quad I_t = 0.1 \frac{1}{10} \sum_{i=1}^{10} \frac{A_B - A_f}{A_B},
\]

where \(A_m\) and \(A_f\) are the fitness of the solution obtained by GA with uniform and fragment crossovers, respectively for \(i\)th randomly generated problem, and \(A_B\) is the fitness of the best solution obtained for the given test problem. The \(I_x\) and \(I_t\) indices characterize the quality of solutions, obtained by the two versions of GA. The greater the index the greater the average deviation of fitness of the solutions obtained by the corresponding GA from the fitness of best solutions obtained for the same problems by both versions of GA together.

2. The average time \(T_u, T_f\) it takes to obtain the best-in-population solution (time taken for the last modification of the best solution obtained) by GA with uniform and fragment crossovers, respectively.

Fig. 3 presents variation of \(I\) and \(T\) indices for the both crossovers for different \(N_c\). The test shows that GA with uniform crossover has much lower index \(I\) than one with fragment crossover. The time \(T\) for uniform crossover has a minimal dependency on the \(N_c\) while for fragment crossover it increases with increase of initial population size. The best solutions are obtained for GA with uniform crossover and \(N_c\) varying in the range 100–200. In this experiment the mutation rate \(m\) is chosen to be 1, which appears to be the best value.

Fig. 4 presents the average index \(I_x\) obtained for the same set of problems as a function of mutation rate \(P_m\) (\(P_m \geq 1\) implies that each newly obtained solution undergoes mutation \(P_m\) times; \(P_m < 1\) implies that it undergoes mutation with probability \(P_m\)). The index \(I_x\) for each \(P_m\) was obtained for \(N_c \in \{50, 100, 150, 200, 250\}\), where for each \(N_c\), 10 problems were solved (total of 50 GA runs for each \(P_m\)). One can see that GAs with \(m = 1\) outperform both GAs with \(P_m > 1\) and \(P_m < 1\). The same result was obtained for GA with F-crossover.
5. Illustrative example

5.1. Minimal cost MSS structures

Consider the series–parallel multi-state system (power substation), which consists of four components:

1. power transformers;
2. capacitor banks;
3. output medium voltage line sections;
4. blocks of commutation equipment.

Each component should be built from elements of certain functionality. The cost, performance rate and availability of elements that can be included in each component are presented in Table 2. Within each component, the elements can be separated in an arbitrary way, but each new protection group requires additional investment. For the sake of simplicity in the example presented, the cost of separation and/or protection for each protection group does not depend on the number of elements it contains, but does depend strictly on the level of protection provided for this group. The descriptions and costs of different protection levels available for each component and the vulnerabilities corresponding to these protection levels are presented in Table 3. The system should meet demand $W = 5$.

The structure optimization problem was solved for four different values of desired system survivability $S^* \in \{0.85, 0.90, 0.95, 0.99\}$. One can see the obtained solutions in Figs. 5–8, where each system element is marked with its
Fig. 9. MSS survivability functions $S_{MSS}(W)$ for four obtained MSS structures.

Fig. 10. Coefficient of variation of ‘best in population solution’ fitness (obtained by 10 search processes as function of number of crossovers) for two separation problems.
version number and each protection group is encased into a rectangle (dashed for protection level 1, regular for protection level 2 and double lined for protection level 3). In Fig. 9, the system survivabilities \( S_{MSS} \) (cumulative probabilities \( Pr(G_{MSS} \geq W) \)) for the obtained four structures are presented as functions of demand \( W \).

5.2. Computational effort and algorithm consistency

The average running time for the C language realization of the algorithm with chosen parameters on a Pentium II PC did not exceed 75 min for a single optimization problem.

To demonstrate the consistency of the suggested algorithm, GA was repeated 10 times with different starting solutions (initial population) for each of four problems with different values of \( S' \). The coefficient of variation was calculated for fitness values of best-in-population solutions obtained during the genetic search by different GA search processes. The variation of this index during the GA proceeding is presented in Fig. 10. One can see that the standard deviation of the final solution fitness does not exceed 1% of its average value.

References


