Journal of Polish Safety and Reliability Association
Summer Safety and Reliability Seminars, Volume 5, Number 1, 2014

Guze Sambor
Maritime University, Gdynia, Poland

Application of the knapsack problem to reliability multi-criteria optimization

Keywords

multi-criteria optimization, reliability, knapsack problem, genetic algorithms, SPEA

Abstract

The main aim of the paper is to translate reliability problems to the knapsack optimization problem. The review of the known methods of multi-criteria optimization is done. Particularly, the SPEA algorithm is presented. Furthermore, the 0-1 knapsack problem solution by SPEA algorithm is introduced and used to the reliability optimization of exemplary parallel-series system.

1. Introduction

The problem of finding optimal solutions is met in many areas of modern science, technology and economics. Civil engineer optimizes the structure of the building and construction material parameters in order to obtain the structure safe and cheap. The navigator must optimize the route of the ship due to safety, time of passage, fuel and cost [6], [9]. The researcher is looking for a mathematical function that best approximates the data collected during the experiment. Each of these problems can be (generally) formulated strictly as an optimization problem, if only we can specify three elements: a model of the phenomenon of distinguished decision variables, objective function - also known as a quality criterion - and limitations.

The same applies to reliability, safety and risk analysis. The current complexity of the technical systems [2] makes important more and more criteria for their safe and reliable operations [4]. This implies that more than one object is taken into account for solving the optimization problems ([6], [8]-[12]).

Some tools for solving the problems of complex technical systems operation, reliability, availability, safety and cost optimization are presented in [3]-[5].

The methods of the reliability prediction and optimization of complex technical systems related to their operation processes are introduced in [5].

In [6] is done the review of the known deterministic optimization methods for engineering and management. Some nondeterministic methods are introduced in [7]. The optimization methods for maritime transportation problems, i.e. weather routing or minimizing fuel consumption are introduced in [6], [9], respectively. The general methods of the optimization are shown in [10]-[12].

The paper shows possibility application of the knapsack problem [7], [12] to multi-criteria optimization of reliability problems.

2. Review of multi-criteria optimization methods

Generally, the single-objected optimization problem is defined as follows (minimizing or maximizing problem):

$$F(x_i) \to \min \text{ or } F(x_i) \to \max,$$
$$l_j(x_i) \le 0, l_j(x_i) \le 0, x_i \ge 0, i, j = 1,2,...,n \qquad (1)$$

where

x_i - decision variables, $i = 1,2,...,n$;

$F(x_i)$ - goal function;

$l_j(x_i)$ - limits function (low or high) for decision variables, $i, j = 1,2,...,n$.

In contrast, the multi-objective (multi-criteria) optimization model can be described as a vector function f that maps a tuple of m decision variables to a tuple of n objectives. The formal notation is as follows [12]:

$$y = f(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \rightarrow \text{min/max} \quad (2)$$

subject to $x = (x_1, x_2, \ldots, x_m) \in X$,
$y = (y_1, y_2, \ldots, y_n) \in Y$,

where
x - decision variable,
X - parameter (variable) space.
y - objective vector,
Y - objective space.

The set of multi-objective optimization problem solutions consists of all decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another. These vectors are known as Pareto optimal, what is related to the concept of domination vector by vector. It is simple to explain after introduce following definitions.

Definition 1. Let us take into account a maximization problem and consider two decision vectors $\boldsymbol{a}, \boldsymbol{b} \in X$, then $\boldsymbol{a}$ is said to dominate $\boldsymbol{b}$ if and only if

$$\forall i \in \{1,2,...,n\} : f_i(\boldsymbol{a}) \geq f_i(\boldsymbol{b}) \\ \wedge \\ \exists j \in \{1,2,...,n\} : f_j(\boldsymbol{a}) \geq f_j(\boldsymbol{b}). \quad (3)$$

Definition 2. All decision vectors which are not dominated by another decision vector are called non-dominated.

When the decision vectors are non-dominated within the entire search space, they are denoted as Pareto optimal (Pareto-optimal front).

These general formulations for single and multi-objective optimization problem are common for different types of engineering problems, which can be related to different optimization problems presented on *Figure 1*.

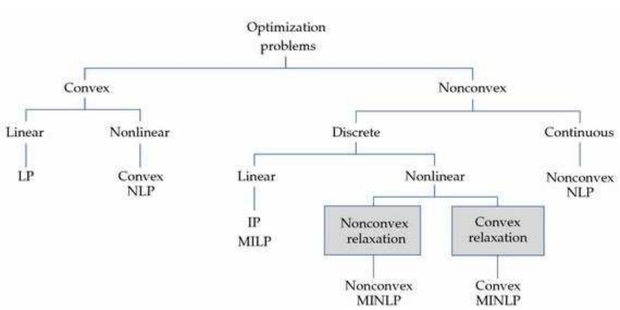

Figure 1. Different types of problem related to optimization problems [8].

The basic classification of the optimization methods consists in their division due to the number of criteria (one or multi-criteria). As it is shown on *Figure 2*, there is possibility to distinguish seven most frequently used methods of multi-criteria optimization and three for one-criterion.

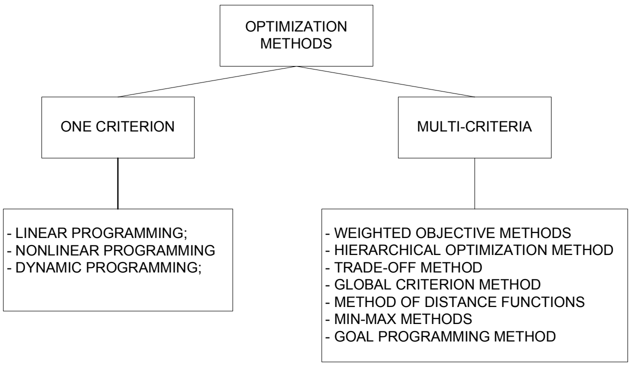

Figure 2. Chosen optimization methods.

These methods represent some general approaches for optimization, i.e.:
- deterministic,
- non-deterministic,
- heuristic,
- evolutionary/genetic.

The above approaches can provide general tools for solving optimization problems to obtain a global or an approximately global optimum. In second case the best way is using the evolutionary or genetic algorithms. The schema of basic genetic algorithm is presented on *Figure 3*.
General operation of genetic or evolutionary algorithms is based on the following steps (see *Figure 3*):
1. Initialization.
2. Calculate fitness.
3. Selection/Recombination/Mutations (parents and children).
4. Finished.

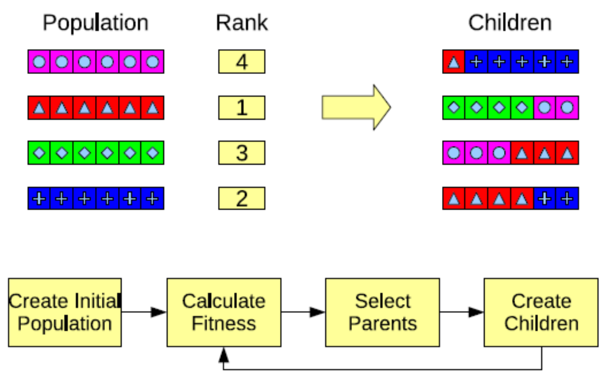

Figure 3. Basic genetic algorithm [11].

The each chromosome of the genetic or evolutionary algorithm is represented as a string of bits.
In the paper the Strength Pareto Evolutionary Algorithm (SPEA) is considered [12].
The basic notations for above algorithm are as follows:
t - number of generation,
P_t - population in generation t,
$\overline{P}_t$ - external set in generation t,
$\overline{P}'$ - temporary external set,
P' - temporary population.

Additionally, it is necessary to give following input parameters:
N - population size,
$\overline{N}$ - maximum size of external set,
T - maximum number of generations,
p_c - crossing probability,
p_m - mutation probability,
A - set of non-dominated solutions.

The Strength Pareto Evolutionary Algorithm is as follows [12]:
Step 1. Initialization:
The initial population P_0 is generated according to procedure:

a) To get item *i*.
b) To add item *i* to set P_0.

Next, the empty external set $\overline{P}_0$ is generated, where *t=0*.

Step 2. The complement of the external set is done.
Let $\overline{P}' = \overline{P}_t$

a) To copy non-dominated items from population P_t to population $\overline{P}'$.
b) To remove dominated items from set $\overline{P}'$.
c) To reduce the cardinality of the set $\overline{P}'$ to value N, using clustering and the solution give into $\overline{P}_{t+1}$.

Step 3. Determination fit function.
The value of the fit function F for items from sets P_t i $\overline{P}_t$ can be found according to following procedure:
The real value $S \in [0,1)$ is assigned for every item $i \in \overline{P}_t$ (called power). This value is proportional to number of items $j \in P_t$, which represents the solutions dominated by item *i*.
The adaptation of item *j* is calculated as sum of all items from external set, represents solution dominated by item *j*, increased by 1.

The aim of addition 1 is to ensure that items $i \in \overline{P}_t$ will have better value of fit function than items from set P_t, i.e.

$$S(i) = \frac{n}{N+1}, \qquad (4)$$

where:
$S(i)$ - power of item *i*,
n - number of items in population dominated by item *i*.

It is assumed that value of fit function for item *i* is equal to his power, i.e.

$$F(i) = S(i). \qquad (5)$$

Step 4. Selection
Let P' = Ø.
For i = 1,2,… k do

a) To choose randomly two items $i, j \in P_t \cup \overline{P}_t$.
b) If $F(i) < F(j)$ then $P' = P' \cup \{i\}$ else $P' = P' \cup \{j\}$, under assumption that value of fit is minimizing.

Step 4. Recombination.
Let P'' =Ø.
For i = 1,2,…N/2 do:

a) To choose two items $i, j \in P'$ and to remove it from $\overline{P}'$.
b) To create items: k, l by crossing the items i, j.
c) To add items k, l to set P'' with probability p_c, else add items i, j to set P''.

Step 5. Mutation
Let P''' =Ø.
For every item $i \in P''$ do:

a) To create item *j* by mutation the item *i* with probability p_m.
b) To add item *j* to set P'''.

Step 6. Finished
Let $P_{t+1} = P'''$ and $t = t+1$.
If $t \geq T$ then return A – non-dominated solution from population P_t and finish else back to *Step 2*.

3. The knapsack problem

This problem has been studied since 1897. It is combinatorial optimization problem. General description is based on given a set of items, each with a mass and a value. There is determined the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible (according to (1)). The knapsack problem is a subset of NP-hard problems. It means that there is non-polynomial algorithm to solve this problem. Therefore, the knapsack problem has been modified many times. i.e. to form of the 0-1 knapsack problem. This modification allows for formulation of knapsack problem as multi-objective optimization problem.

Generally, a 0-1 knapsack problem consists of a set of items, weight and profit associated with each item, and an upper bound for the capacity of the knapsack. The main goal is to find a subset of items which maximizes the profits and all selected items fit into the knapsack, i.e., the total weight does not exceed the given capacity [6].

This single-objective problem can be extended directly to the multi-objective case by allowing an arbitrary number of knapsacks. Formally, the multi-objective 0-1 knapsack problem can be defined in the following way [12]:

Given a set of m items and a set of n knapsacks, with

$p_{i,j} =$ profit of item j according to knapsack i,

$w_{i,j} =$ weight of item j according to knapsack i,

$c_i =$ capacity of knapsack i,

find a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \in \{0,1\}^m$, such that

$$\forall i \in \{1,2,...,n\} : \sum_{\mathbf{j=1}}^{\mathbf{m}} \mathbf{w_{i,j}} \cdot x_j \le c_i \qquad (6)$$

and for which $f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x}))$ is maximum, where

$$f_i(\mathbf{x}) = \sum_{\mathbf{j=1}}^{\mathbf{m}} \mathbf{p_{i,j}} \cdot x_j \qquad (7)$$

and $x_j = 1$ if and only if when item j is chosen.

The solutions of knapsack problem can be described in terms of a genetic or evolutionary methods.

4. Reliability of the two-state parallel-series system

In the case of two-state reliability analysis of parallel-series systems we assume that ([2]):

- n is the number of system components,
- E_{ij}, $i = 1,2,...,k_n$, $j = 1,2,...,l_i$, are components of a system,
- T_{ij} are independent random variables representing the lifetimes of components E_{ij}, $i = 1,2,...,k_n$, $j = 1,2,...,l_i$,
- $R_{ij}(t) = P(T_{ij} > t), t \in <0,\infty)$, is a reliability function of a component E_{ij}, $i = 1,2,...,k_n$, $j = 1,2,...,l_i$,
- $F_{ij}(t) = 1 - R_{ij}(t) = P(T_{ij} \le t), t \in <0,\infty)$, is the distribution function of the component E_{ij} lifetime T_{ij}, $i = 1,2,...,k_n$, $j = 1,2,...,l_i$, also called an unreliability function of a component E_{ij}, $i = 1,2,...,k_n$, $j = 1,2,...,l_i$.

Moreover, we assume that components E_{i1}, E_{i2}, …, E_{il_i}, $i = 1,2,...,k_n$, create a parallel subsystem S_i, $i = 1,2,...,k_n$, and that these subsystems create a series system.

Definition 3. A two-state system is called parallel-series if its lifetime T is given by

$$T = \min_{1 \le i \le k_n}\{\max_{1 \le j \le l_i} T_{ij}\}.$$

According to above definition, the reliability function of the two-state parallel-series system is given by

$$\overline{R}_{k_n, l_1, \ldots, l_{k_n}}(t) = \prod_{i=1}^{k_n}\left[1 - \prod_{j=1}^{l_i} F_{ij}(t)\right], t \in (-\infty, \infty). \qquad (8)$$

5. Multi-criteria methods for reliability optimization problems

We assume that the two-state parallel-series system with three main units S_i is given ($i = 1,2,3$). Every unit is the parallel subsystem consists of maximum three components which can be chosen to provide redundancy (see *Figure 4*).

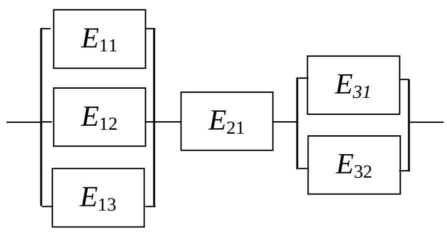

Figure 4. Exemplary scheme of a parallel-series system

Every component of the system can have two states, functioning with the nominal capacity or total failure, corresponding to capacity 0. The main

characteristics of these components are lifetime, nominal capacity, cost and weight. Without loss of generality, component capacities can be measured as a percentage of the maximum demand.
The main problem is how to design a system with long lifetime.

Under above conditions and according to the Section 3, we assume, that

- c_i - the time to failure of designed system;
- $p_{i,j}$ - the profit (lifetime) of using the particular component;
- $w_{i,j}$ - weight of the component in subsystem with cost of its installation.

The best way to represent the multi-criteria optimization problem is binary coding. Let us assume that a chromosome represents the reliability of whole system (see *Figure 5*). In this chromosome the gen equal to 1 means that given component is inserted to knapsack. On the other hand, the gen in this chromosome is equal to 0 means that this component is not inserted to knapsack. The length of a chromosome is the number of the components, which are under investigation (number of system components, see *Figure 5*).

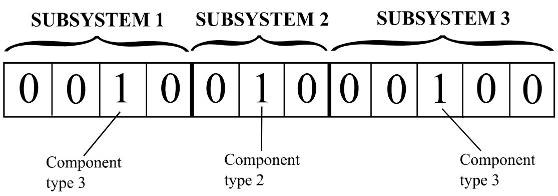

Figure 5. Exemplary chromosome coding of system components.

The characteristics of the system components available on market are given in *Table 1*.

The goal of the problem is maximize reliability of the system, while the cost and weight are minimal.
The execution of algorithm described in Section 2 step-by-step, taken into account the expressions (3) – (7) and information given in *Table 1*, allows us to find the optimal solution for considered reliability optimization problem (in sense Pareto-optimal). The one of the possible solutions can be given as the above chromosome (see *Figure 6*):

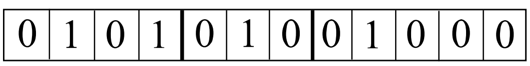

Figure 6. The resulting chromosome code

Table 1. Characteristics of the system components

Subsystem	Component type	Lifetime [h]	Feeding capacity [%]	Cost [USD]	Weight [g]
1	1	9950	120	99.0	34.4
	2	9840	100	59.0	24.2
	3	9450	85	37.0	28.6
	4	9200	33	22.5	22.5
2	1	9800	100	18.5	25.4
	2	9740	73	16.9	23.6
	3	9500	25	13.9	24.6
3	1	9960	128	189.0	29.7
	2	9980	97	91.0	33.6
	3	9760	74	83.3	31.4
	4	9820	55	79.6	30.5
	5	9710	36	76.7	32.7

This chromosome represents the system composed of four components, given on *Figure 7*.

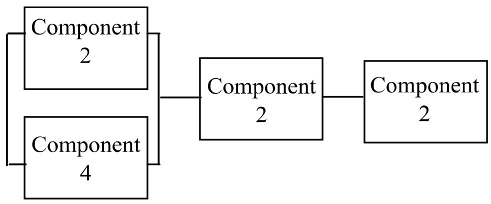

Figure 7. The system designed according to results of optimization

According to information from Section 4, the numerical characteristics of resulting system are as follows:

$$T = \min\{9840, 9740, 9980\} = 9740[h],$$

$$Cost = 178.9[USD],\ Weight = 103.9[g].$$

6. Conclusion

The review and classifications of the well-known multi-criteria optimization methods have been done. The SPEA algorithm has been described step-by-step and the knapsack problem with its binary modification has been presented. It has also been used the SPEA algorithm to solve the 0-1 knapsack problem. Finally, application of methods for multi-criteria optimization for reliability problem has been done.
The methods and algorithms presented in the paper can be applied to the safety and risk analysis.
The example, from Section 5, is only showing potential applications of methods and algorithms, which are described in the article.

References

[1] Guze, S. & Smolarek, L. (2011). Methods for risk minimizing in the process of decision-making under uncertainty. *Journal of Polish Safety and Reliability Association - JPSRA* Vol. 2, No. 1, 123 – 128.

[2] Kołowrocki, K. (2004). *Reliability of Large Systems*. Elsevier, Amsterdam - Boston - Heidelberg - London - New York - Oxford - Paris - San Diego - San Francisco - Singapore - Sydney - Tokyo.

[3] Kołowrocki, K. & Soszyńska, J. (2010). Optimization of complex technical systems operation processes. *Maintenance Problems* No 1, 31-40.

[4] Kołowrocki, K. & Soszyńska-Budny, J. (2011). *Reliability and Safety of Complex Technical Systems and Processes, Modeling – Identification – Prediction – Optimization.* Springer-Verlag.

[5] Kołowrocki, K. & Soszyńska-Budny, J. (2013). Reliability prediction and optimization of complex technical systems with application in port transport. *Journal of Polish Safety and Reliability Association, Summer Safety and Relibility Seminars – SSARS 2013* Vol. 3, No. 1-2, 263 – 279.

[6] Marie, S. & Courteille, E. (2009) Multi-Objective Optimization of Motor Vessel Route. *TransNav, The International Journal on Marine Navigation and Safety of Sea Transportation* Vol. 3, No. 2, 133-141.

[7] Martello, S. & Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. Chichester, U.K.: Wiley.

[8] Ming-Hua, L., Jung-Fa, T. & Chian-Son, Y. (2012). A Review of Deterministic Optimization Methods in Engineering and Management. *Mathematical Problems in Engineering*. Volume 2012.

[9] Sun, W. & Yuan, Y.-X. (2006). *Optimization theory and methods: nonlinear programming*. Springer-Verlag.

[10] Szłapczyńska, J. (2013). Multicriteria Evolutionary Weather Routing Algorithm in Practice. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* Vol. 7, No. 1, 61-65.

[11] Venter, G. (2010). *Review of Optimization Techniques, Encyclopedia of Aerospace Engineering*. John Willey and Sons, Ltd.

[12] Zitzler, E. & Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation* Vol. 3, No. 4, 257 – 271.