# EVOLUTIONARY ALGORITHMS IN MULTIOBJECTIVE PROBLEMS

A.M.Syomkin, A.I.Zmitrovich

**Abstract**. Many real-world problems involve two types of difficulties: 1) multiple, conflicting objectives and 2) a highly complex search space. Efficient evolutionary strategies have been developed to deal with both types of difficulties. Evolutionary algorithms possess several characteristics such as parallelism and robustness that make them preferable to classical optimization methods. In this work I conducted comparative studies among the well-known evolutionary algorithms based on NP-hard 0-1 multiobjective knapsack problem.

## Overview of evolutionary algorithms

Six of the most salient multiobjective evolutionary algorithms have been taken for comparison studies.

1. VEGA (Schaffer's Vector Evaluated Genetic Algorithm)
2. NPGA (Horn, Nafpliotis, and Goldberg's Niched Pareto Genetic Algorithm)
3. FFGA (Fonseca and Fleming's Multiobjective Genetic Algorithm)
4. NSGA (Srinivas and Deb's Nondominated Sorting Genetic Algorithm)
5. SPEA (Strength Pareto Evolutionary Algorithm)

As additional points of reference, the following modification of SPEA has been added to comparison studies: SPEA-R (Strength Pareto Evolutionary Algorithm with restricted matting).

Single-objective evolutionary algorithms using the weighting method have been chosen as well. In contrast to other algorithms 100 independent runs were performed per test problem, each run optimizes toward another randomly chosen linear combination of objectives. Two versions of single –objective evolutionary algorithms were used: one with 100 generation per linear combination (SO-1), another with 500 generations (SO-5).

## Performance measures

To assess each particular solution and evaluate the overall performance of each algorithm the following measures are usually used: $s$ - measure (measure of dominated space), $C$ – measure (coverage of two sets).

The function $s$ is measure of how much the objective space is weakly dominated by a given nondominated set $A$.

Let $A=(x_1,x_2,...,x_l)\subseteq X$ be the set of $l$ decision vectors. The function $s(A)$ gives the volume enclosed by the union of the polytopes $p_1,p_2,...,p_l$, where each $p_i$ is formed by the intersections of the following hyperplanes arising out of $x_i$, along with the axes: for each axis in the objective space, there exists a hyperplane perpendicular to the axis and passing through the point $(f_1(x_i),f_2(x_i),...,f_k(x_i))$.

Note, in two-dimensional case, each $p_i$ represents a rectangle defined by the points $(0,0)$ and $(f_1(x_i),f_2(x_i))$.

Let $A,B\subseteq X$ be two sets of decision vectors. The function $C$ maps the ordered pair $(A,B)$ to the interval [0,1] as follows:

$$C(A,B)=\frac{|\{b\in B \mid \exists a\in A: a \succeq b\}|}{|B|} \quad (1)$$

The value $C(A,B)=1$ means that all decision vectors in $B$ are weakly dominated by $A$. The opposite value $C(A,B)=0$ means that none of decision vectors in $B$ are weakly dominated by $A$.

## Multiobjective knapsack problem

### Problem statement

Generally, 0-1 knapsack problem consists of items, weight and profit associated with each item and capacity of the knapsack. The task is to find a subset of items to reach the maximum total profit, yet all selected items fit into the knapsack.

This problem can be extended to multiobjective optimization problem by allowing several numbers of knapsacks. Formally, given a set of $n$ items and $k$ 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 $i$, $i=\overline{1,n}$, $j=\overline{1,k}$,

find the vector $x=(x_1,x_2,...,x_n)\in\{0,1\}^n$ such to satisfy the following:

$$\begin{cases} y=f(x)=(f_1(x),f_2(x),...,f_k(x))\to\max \\ f_i(x)=\sum_{j=1}^{n} p_{i,j}x_j,(1\le i\le k) \\ e_i(x)=\sum_{j=1}^{n} w_{i,j}x_j\le c_i,(1\le i\le k) \\ x=(x_1,x_2,...,x_n)\in\{0,1\}^n \end{cases} \quad (2)$$

### Test data

In order to reach reliable results, nine different test cases were investigated where number of items and knapsacks were varied. Two, three, four objectives were taken in combination with 250, 500, 750 items. Uncorrelated profit and weighs were used, where $p_{i,j}, w_{i,j}$ were random integers in the interval $[0,100]$. The knapsack capacities were defined as follows:

$$c_i=\frac{1}{2}\sum_{i=1}^{n} w_{i,j} \quad (3)$$

### Constrains handling

A particular individual $i\in\{0,1\}^n$ encodes a solution $x\in\{0,1\}^n$. The mapping function $x=m(i)$ realizes a simple method that decodes an individual $i$ according to the following schema: First, set $x=i$, then remove step by step items from $x$ as long as any capacity constrains is violated. The order in which items are removed is determined by the maximum profit/weight ratio $q_j$ per item,

$$q_j=\max_{i=1}^{n}\left\{\frac{p_{i,j}}{w_{i,j}}\right\}. \quad (4)$$

**Parameters settings** Independent to particular algorithm the following parameters were