Multi-Objective Optimization: Linear Combinations Do Not Cover Pareto Set, So What Does?

Cesar F. Gomez Guillen, Miguel A. Sepulveda, and Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso
cfgomezguil@miners.utep.edu, masepulveda@miners.utep.edu, vladik@utep.edu

Formulation of the problem. In some practical situation, we have a clear objective: to optimize a known objective function $f(x)$ – e.g., profit for companies. Many effective algorithms are known for solving such well-defined optimization problems.

Often, however, the problems are not so well-defined: there are several different objective functions $f_1(x), \ldots, f_n(x)$. Usually, we calibrate them so that the status quo state s (when we do not make any decision) corresponds to $f_i(s) = 0$. This way, we are only looking for alternatives x for which $f_i(x) \geq 0$ for all i. In such situations, we do not want alternatives x which are *dominated* by others, i.e., for which, for some y, we have $f_i(x) \leq f_i(y)$ for all i and $f_j(x) < f_j(y)$ for some j. We thus want to generate the set P of all non-dominated alternatives, so that a human decision maker can make a final decision. This problem is known as *multi-objective optimization*, and the set P is known as the *Pareto set*.

Since there are many effective algorithms for solving traditional optimization problem, a natural idea is to reduce multi-objective optimization to several optimization ones. Namely, we select a continuous function $F(v_1, \ldots, v_n)$ which (1) is (non-strictly) increasing relative to each of its inputs and (2) which does not change under permutations. Then, we optimize functions $F(c_1 \cdot f_1(x), \ldots, c_n \cdot f_n(x))$ corresponding to different $c_i \geq 0$.

It is also reasonable to require that (3) the function F is *homogeneous*, i.e., $F(c \cdot v_1, \ldots, c \cdot v_n) = c \cdot F(v_1, \ldots, v_n)$ for all c and v_i, and (4) that if both the Pareto set and a point $p \in P$ are invariant under a permutation, the corresponding values c_i should be invariant too.

Practitioners often use $F(v_1, \ldots) = \sum\limits_i v_i$, which means considering linear combinations of objective functions. The problem is that if the set S of possible values of $(f_1(x), \ldots, f_n(x))$ is not convex, optimizing linear combinations does not cover the whole Pareto set: an example is given below. So what shall we do?

A reasonably simple solution: use min **instead the sum.** One can easily show that if we use $F(v_1, \ldots) = \min(v_1, \ldots, v_n)$, then we do cover the whole Pareto set: each point $x_0 \in P$ is covered when we take $c_i = 1/f_i(x_0)$. In this case, for $c_i = 1/f_i(x_0)$, for the function $g(x) \stackrel{\text{def}}{=} F(c_1 \cdot f_1(x), \ldots)$, we have $g(x_0) = 1$, but for every other y, we will have $g(y) \leq 1$ – otherwise y would dominate x_0.

Of course, for any $c > 0$, the function $c \cdot \min(v_1, \ldots)$ has the same property.

A natural mathematical question: are $c \cdot \min$ **the only functions with this property?** Our answer is "yes". Here is a proof for $n = 2$. Let us first prove that $F(1, a) = F(1, 1)$ for all $a > 1$. Indeed, let us consider the Pareto set consisting of line segments $(0, a) - (1, a) - p = (1 + \varepsilon, 1 + \varepsilon) - (a, 1) - (a, 0)$. Both set P and the point p do not change when we swap v_1 and v_2, so we should have $c_1 = c_2$ for the values that lead to the maximum at p. Since we have a maximum at p, we get $F(c_1 \cdot (1 + \varepsilon), c_1 \cdot (1 + \varepsilon)) \geq F(c_1 \cdot 1, c_1 \cdot a)$. Hence due to homogeneity $F(1 + \varepsilon, 1 + \varepsilon) \geq F(1, a)$. In the limit $\varepsilon \to 0$, we get $F(1, 1) \geq F(1, a)$. However, due to monotonicity, $F(1, 1) \leq F(1, a)$, so $F(1, a) = F(1, 1)$ for all $a > 1$.

For any v, due to homogeneity, we have $F(v, v) = v \cdot F(1, 1) = c \cdot v$. Similarly for any $v_1 < v_2$, we have $F(v_1, v_2) = v_1 \cdot F(1, v_2/v_1)$. Since $v_2 > v_1$, we have $v_2/v_1 > 1$, so $F(1, v_2/v_1) = F(1, 1) = c$ and thus, $F(v_1, v_2) = c \cdot v_1$. Due to symmetry, for $v_1 > v_2$, we have $F(v_1, v_2) = F(v_2, v_1)$ and thus, $F(v_1, v_2) = c \cdot v_2$. In both cases, $F(v_1, v_2) = c \cdot \min(v_1, v_2)$.