Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Goal Programming, ReLU Networks and Endogenous Priorities: An α-Fair Unification of Scalarization

Manish Bajpai

manishbajpai7@gmail.com

University of Delaware

Research Article

Keywords: Goal programming, Multiple objective programming, Neural networks, Scalarization, Fairness, Conservation planning

Posted Date: November 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11526508/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Goal Programming, ReLU Networks and Endogenous Priorities: An α-Fair Unification of Scalarization

Manish Bajpai*

Department of Applied Economics and Statistics, University of Delaware, Newark, DE 19716, USA

manishbajpai7@gmail.com ORCID: 0009-0007-2383-754X

*Corresponding author

Abstract

Weighted goal programming and the feedforward neural network both rank alternatives by a weighted aggregate of criteria plus an offset. We show that the resemblance is an identity: a linear achievement function is an affine neuron whose bias is minus the weighted aspiration level, weighted goal programming with one-sided deviations is a one-hidden-layer network of rectified linear units with nonnegative readout weights, and Chebyshev goal programming is the same layer followed by max-pooling. We then let the priorities depend on the alternative being evaluated through a power rule with a single exponent, which in network terms is softmax attention. Every fixed point of the induced weighted-sum iteration is a supported Pareto optimum and, on convex problems, a stationary point of the well-known alpha-fair aggregate, so the exponent spans extreme-point seeking, the equal-weight sum, Nash bargaining and max–min compromise; below one, the fixed point is unique on convex problems and is computed by a Frank–Wolfe scheme. Maximizing the self-weighted score directly instead maximizes a Lehmer mean, which by a classical property respects Pareto dominance exactly when the exponent lies between zero and one, and selects a different compromise. Fixed points never reach unsupported Pareto points. Experiments on test problems and on the 2010 project portfolio of a U.S. Department of Defense conservation programme show that endogenous priorities recover the goal-programming portfolio without any weights being chosen, while the max–min portfolio is unsupported and, as the theory predicts, out of their reach.

Keywords: Goal programming; Multiple objective programming; Neural networks; Scalarization; Fairness; Conservation planning

1 Introduction

Two traditions in quantitative decision making rank alternatives in the same way. Goal programming (Charnes et al., 1955; Charnes and Cooper, 1961; Tamiz et al., 1998; Jones and Tamiz, 2010) scores an alternative by weighted deviations of its criteria from aspiration levels. The artificial neuron scores an input by a weighted sum plus a bias, passed through a nonlinearity such as the rectified linear unit (Nair and Hinton, 2010). Practitioners moving between operations research and machine learning routinely notice the resemblance. The purpose of this paper is to make it exact, and to use the exact form to analyse a family of scalarizations

in which the priority weights are not supplied by the decision maker but generated from the alternative itself.

The question arose from an application. Messer et al. (2016) used goal programming, alongside binary linear programming, to select conservation projects around U.S. military installations for the Department of Defense's Readiness and Environmental Protection Integration (REPI) programme, and showed that optimization yields substantially more readiness and environmental protection for the same budget than ranking projects by benefit scores. If a goal programme scores projects by weighted deviations from aspirations, and a neural network scores inputs by weighted sums and biases, how far does the resemblance go, and what follows when the priorities themselves respond to the portfolio being scored? Section 7.4 returns to the REPI data.

The paper has two parts. The first is a correspondence. We prove that the linear achievement function is an affine neuron with bias equal to minus the weighted aspiration level; that the weighted goal programme with one-sided deviation variables is a one-hidden-layer ReLU network with nonnegative readout; and that Chebyshev (min–max) goal programming (Flavell, 1976; Romero, 1991) is a ReLU layer followed by max-pooling (Section 3). The deviation variable $d^- = \max(0, g - x)$ of goal programming is a ReLU unit, and the correspondence holds in both directions on the cone of nonnegative readout weights.

The coincidence has a history. The function $\max(0, \cdot)$ entered mathematical models of neural activity with Householder (1941) and multilayer networks with Fukushima (1969); Hahnloser et al. (2000) made its biological and computational case, Nair and Hinton (2010) named it the rectified linear unit, and it became the default activation of deep learning after Glorot et al. (2011) and Krizhevsky et al. (2012). The deviation variables of Charnes et al. (1955), from which goal programming developed, already took the form $\max(0, g - x)$. Operations research thus used the rectifier as a deviation variable more than a decade before it appeared in multilayer networks and half a century before it was named, although Householder's 1941 model is earlier still. Neural networks have also been used to *solve* goal programmes: Van Hulle (1991) designed a Hopfield-type network whose dynamics converge to the solution of a goal programme. That work uses a network as a solver; the correspondence established here instead identifies the goal-programming achievement function itself with a feedforward ReLU network.

The second part studies endogenous priorities. Given criteria values $x \in \mathbb{R}^p_{++}$ we set

$$\lambda_\alpha(x)_i = \frac{x_i^{\alpha-1}}{\sum_{j=1}^p x_j^{\alpha-1}} = \operatorname{softmax}\big((\alpha-1)\log x\big)_i,$$

so that each criterion's priority depends on its own attained level. In the aggregation literature, weighted averages whose weights are generated by the inputs in this way are called mixture functions (Špirková et al., 2019). In network language this is a single attention head (Bahdanau et al., 2015; Vaswani et al., 2017) with keys $\log x$ and inverse temperature $\alpha - 1$, which is negative when lagging criteria are favoured. Because λ_α depends on x, maximizing $\langle \lambda_\alpha(x), x \rangle$ is not a weighted-sum problem, and two natural readings exist: treat λ_α as fixed during each maximization and iterate to a fixed point, or differentiate through it. We analyse both.

Contributions.

1. An exact dictionary between goal programming and feedforward ReLU networks (Propositions 1–3, Table 1).

2. For the fixed-point reading, a characterization of fixed points by a variational inequality, which on convex problems is stationarity of the aggregate F_α (Theorem 8); Pareto optimality and supportedness of every fixed point (Theorem 7); monotone ascent for $\alpha > 1$ with extreme-point iterates (Theorem 9); uniqueness and an $O(1/k)$ algorithm for $\alpha < 1$ (Theorem 11); and the limits $\alpha = 1, 0, -\infty$ (Proposition 12).

3. For the direct reading, the observation that the self-weighted score is a Lehmer mean, so that the classical monotonicity range of Lehmer means (Farnsworth and Orr, 1986) is exactly the range $\alpha \in [0, 1]$ in which it respects Pareto dominance (Proposition 14), and that the two readings select different compromises.

4. A negative result: in the fixed-point reading no choice of α reaches unsupported Pareto points (Theorem 13), and a self-normalized rule $\lambda = w/\mathbf{1}^\top w$ adds no expressive power to the neuron (Proposition 5).

5. Numerical confirmation on bi-objective fronts and a three-objective production problem, including a comparison of network, self-normalized, goal-programming and endogenous-priority scores, and an application to the 44-project REPI conservation dataset of Messer et al. (2015) (Section 7).

Scope of the contribution. The paper is a bridge between two literatures rather than a new solution method, and it should be read as such. Its ingredients—the α-fair family, input-dependent weights, the monotonicity of Lehmer means and the Frank–Wolfe method—are established, and the prior work is credited below. What the paper adds is the exact correspondence itself, the reading of goal priorities as attention that it makes possible, precise statements of what each reading of endogenous priorities guarantees and does not guarantee, and a real-data illustration. The correspondence also works in the interpretive direction: the readout weights of a ReLU layer with nonnegative readout can be read as goal priorities and its biases as aspiration levels.

Relation to prior work. The α scale on which our results are stated is not new, and we do not claim it. For $\alpha \leq 1$ the aggregate F_α is the α-fair utility of Mo and Walrand (2000), introduced for congestion control; their fairness parameter equals $1 - \alpha$ in our notation, so proportional fairness (Kelly et al., 1998) is our $\alpha = 0$ and max–min fairness our $\alpha \to -\infty$. That a single parameter interpolates between the utilitarian sum, Nash bargaining (Nash, 1950) and max–min fairness is standard in fair optimization (Ogryczak et al., 2014; Chen and Hooker, 2023), and fairness–efficiency trade-offs along this family are studied by Bertsimas et al. (2011) and Lan et al. (2010). The same interpolation has recently been applied across objectives rather than agents in multi-objective reinforcement learning, through generalized p-means of the objective values (Fan et al., 2023; Kim et al., 2025). The range $\alpha > 1$, with convex aggregates, lies outside

the α-fair family. Nash bargaining has been used to weight task gradients in multi-task learning (Navon et al., 2022), and Ban and Ji (2024) apply α-fair utilities to the per-task rates of loss decrease; multi-task learning as multiobjective optimization goes back to Désidéri (2012) and Sener and Koltun (2018).

Weights generated by the values being aggregated are likewise known. In aggregation theory the score $\langle \lambda(x), x \rangle$ with $\lambda_i(x) \propto \varphi(x_i)$ is a mixture function, and its monotonicity has been studied in detail (Špirková et al., 2019; Wilkin and Beliakov, 2015); with $\varphi(t) = t^{\alpha-1}$ it is the Lehmer mean, whose monotone range is classical (Farnsworth and Orr, 1986). In multiobjective optimization, priority schemes that depend on the alternative include ordered weighted averaging (Yager, 1988; Ogryczak and Śliwiński, 2003), whose weights depend on the rank order of criteria, the adaptive weighted-sum method of Kim and de Weck (2005), which adapts weights to the geometry of the front, and sequential reference-point methods that update the search direction from the objective values already attained (Jiménez et al., 2021). Finally, the fixed-point iteration we analyse for $\alpha < 1$ is a Frank–Wolfe method (Frank and Wolfe, 1956; Jaggi, 2013) on F_α, the standard algorithmic view of α-fair network utility maximization.

Our contribution relative to these works is therefore not the aggregate, the mixture form of the weights or the algorithm, but (i) the exact goal-programming–network identity in which they sit, including max-pooling for Chebyshev goal programming; (ii) the reading of endogenous goal priorities as softmax attention and the analysis of the resulting fixed points as a scalarization method in the sense of Miettinen (1999) and Ehrgott (2005), including the supported-point limitation; (iii) the distinction between the fixed-point and direct readings, with the classical Lehmer monotonicity range recast as an exact Pareto-compliance condition for the self-weighted score; and (iv) the application to a real conservation portfolio. Mixed-integer encodings of trained ReLU networks (Fischetti and Jo, 2018; Anderson et al., 2020) use the same $\max(0, \cdot)$ structure in the opposite direction, embedding a network in an optimization model rather than reading an optimization model as a network.

The remainder of the paper is organized as follows. Section 2 fixes notation. Section 3 establishes the goal-programming–network correspondence. Section 4 introduces endogenous priorities. Section 5 develops the fixed-point theory and Section 6 the direct reading. Section 7 reports computations and Section 8 discusses limitations and extensions.

2 Preliminaries

We consider $p \geq 2$ criteria, all to be maximized, and work in criterion space. For a feasible decision set Y and criterion map $f : Y \to \mathbb{R}^p$, let $\mathcal{X} = f(Y)$ denote the attainable set.

Assumption 1. $\mathcal{X} \subset \mathbb{R}^p_{++}$ is nonempty and compact. Criteria have been normalized to a common scale and shifted to be strictly positive, for example $x_i = (f_i - z_i^{\mathrm{nad}})/(z_i^{\mathrm{id}} - z_i^{\mathrm{nad}}) + \varepsilon$ with ideal point z^{id}, nadir point z^{nad} and $\varepsilon > 0$.

For $x, y \in \mathbb{R}^p$ we write $y \succ x$ (y dominates x) if $y \geq x$ and $y \neq x$. A point $x^* \in \mathcal{X}$ is *Pareto optimal* if no $y \in \mathcal{X}$ dominates it, and *weakly Pareto optimal* if no $y \in \mathcal{X}$ satisfies $y > x^*$ componentwise. A Pareto optimal point is *supported* if it maximizes $\langle \lambda, x \rangle$ over $\mathcal{X}$ for some $\lambda \in \mathbb{R}^p_{++}$ (Geoffrion, 1968; Ehrgott, 2005). Weighted sums reach only supported points

(Das and Dennis, 1997), whereas every Pareto point is obtainable by some weighted Chebyshev scalarization (Bowman, 1976), which may also return weakly Pareto optimal points (Steuer and Choo, 1983).

We write $\Delta^{p-1} = \{\lambda \in \mathbb{R}^p_+ : \mathbf{1}^\top \lambda = 1\}$, $\mathrm{ReLU}(t) = \max(0, t)$ componentwise, $\odot$ and $\oslash$ for componentwise product and division, and $N_{\mathcal{X}}(x)$ for the normal cone of a convex set $\mathcal{X}$ at x.

Goal programming. Given aspiration levels $g \in \mathbb{R}^p$, priorities $\lambda \in \Delta^{p-1}$, overachievement weights $\mu \geq 0$ and normalization constants $k_i > 0$, written as scales $w_i = 1/k_i$, the *weighted goal programming achievement function* is

$$A_{\mathrm{W}}(x) = \min_{d^-, d^+ \geq 0} \Big\{ \sum_{i=1}^{p} w_i(\lambda_i d_i^- + \mu_i d_i^+) \ : \ x_i + d_i^- - d_i^+ = g_i, \ i = 1, \ldots, p \Big\}, \tag{1}$$

to be minimized over $\mathcal{X}$, and the *Chebyshev achievement function* is

$$A_{\mathrm{C}}(x) = \min_{d^-, d^+ \geq 0} \Big\{ \max_i \lambda_i w_i d_i^- \ : \ x_i + d_i^- - d_i^+ = g_i, \ i = 1, \ldots, p \Big\}. \tag{2}$$

These are the standard normalized weighted and min–max variants (Romero, 1991; Tamiz et al., 1998; Jones and Tamiz, 2010).

3 Goal programming as a network layer

Three objects are kept separate throughout: priorities λ, scales w and aspirations g. Keeping them separate is what makes the following identities exact.

We first record the linear case. The *linear achievement function* $-\sum_i \lambda_i w_i(g_i - x_i) = -\sum_i \lambda_i w_i(d_i^- - d_i^+)$ penalizes underachievement and rewards overachievement at the same rate; it corresponds to (1) with $\mu = -\lambda$, which lies outside the goal-programming model since $\mu \geq 0$ there, but it is the natural linear relative of that model.

Proposition 1 (Linear achievement function = affine neuron)**.** *Let $\theta = \lambda \odot w$ and $b = -\langle \theta, g \rangle$. Then*

$$-\sum_{i=1}^{p} \lambda_i w_i(g_i - x_i) = \langle \theta, x \rangle + b.$$

Conversely, for given scales $w \in \mathbb{R}^p_{++}$, every affine neuron (θ, b) with $\theta \geq 0$, $\theta \neq 0$ equals c times such an achievement function, where $\lambda = (\theta \oslash w)/\|\theta \oslash w\|_1$, $c = \|\theta \oslash w\|_1 > 0$ and g is any solution of $\langle \theta, g \rangle = -b$.

Proof. The first statement is algebra. For the converse, $\theta = c(\lambda \odot w)$ by construction, and $\{g : \langle \theta, g \rangle = -b\}$ is a nonempty hyperplane because $\theta \neq 0$. The factor c does not change any maximizer. □

A neuron identifies only the scalar $\langle \theta, g \rangle$, not the individual aspirations. The one-sided penalties that distinguish goal programming from a weighted sum require a hidden layer.

Table 1: Dictionary between goal programming and feedforward networks.

Goal programming	Feedforward network
Linear achievement function	Affine neuron with $b = -\langle \lambda \odot w, g \rangle$
Scaled underachievement $w_i d_i^-$	ReLU unit $\mathrm{ReLU}(w_i g_i - w_i x_i)$
Weighted goal programme	ReLU layer with nonnegative linear readout
Chebyshev goal programme	ReLU layer with max-pooling
Normalization constants k_i	Input scales $w_i = 1/k_i$
Endogenous priorities $\lambda_\alpha(x)$	Softmax attention on $\log x$, inverse temperature $\alpha - 1$
Fixed-point priorities (Section 5)	Attention with stop-gradient
Directly optimized priorities (Section 6)	Attention without stop-gradient

Proposition 2 (Weighted goal programme = one-hidden-layer ReLU network)**.** *Let* $w \in \mathbb{R}^p_{++}$, $\lambda \geq 0$ *and* $\mu \geq 0$. *Then*

$$A_{\mathrm{W}}(x) = \begin{bmatrix} \lambda \\ \mu \end{bmatrix}^\top \mathrm{ReLU}\left(\begin{bmatrix} -\operatorname{diag}(w) \\ \operatorname{diag}(w) \end{bmatrix} x + \begin{bmatrix} w \odot g \\ -w \odot g \end{bmatrix} \right),$$

a network with $2p$ *hidden ReLU units, a diagonal first layer and nonnegative readout. If* $\mu = 0$, *the overachievement units carry zero readout weight and may be dropped. Conversely, every network* $N(x) = \langle \nu, \mathrm{ReLU}(Ax + c) \rangle$ *with* $\nu \geq 0$, $\nu \neq 0$ *and no output bias equals* $\|\nu\|_1$ *times the weighted achievement function of the goals* $y_j \geq c_j$ *on the derived linear criteria* $y_j = -a_j^\top x$, *with unit scales, priorities* $\nu/\|\nu\|_1$ *and* $\mu = 0$.

Proof. Problem (1) separates over i. The feasible pairs satisfy $d_i^- - d_i^+ = g_i - x_i$ with $d_i^\pm \geq 0$, so $d_i^- \geq \mathrm{ReLU}(g_i - x_i)$ and $d_i^+ = d_i^- - (g_i - x_i)$. Along this ray the objective $w_i(\lambda_i d_i^- + \mu_i d_i^+)$ is nondecreasing in d_i^-, so its minimum value is attained at $d_i^- = \mathrm{ReLU}(g_i - x_i)$, $d_i^+ = \mathrm{ReLU}(x_i - g_i)$ (uniquely if $\lambda_i + \mu_i > 0$; otherwise the value is zero for every feasible pair, which the formula also gives). Since $w_i > 0$, $w_i \,\mathrm{ReLU}(g_i - x_i) = \mathrm{ReLU}(w_i g_i - w_i x_i)$ and likewise for overachievement, which gives the stated network. For the converse, $\mathrm{ReLU}(a_j^\top x + c_j) = \mathrm{ReLU}(c_j - y_j)$ is the underachievement of the goal $y_j \geq c_j$, and $N = \|\nu\|_1 \sum_j (\nu_j/\|\nu\|_1) \,\mathrm{ReLU}(c_j - y_j)$. □

Remark 1. Negative readout weights reward a deviation rather than penalizing it, which has no goal-programming meaning; the correspondence therefore holds exactly on the cone $\nu \geq 0$. Dense first layers correspond to goals on linear combinations of criteria, which is common in practice, for example ratio or balance goals.

Proposition 3 (Chebyshev goal programme = max-pooling)**.** *If* $w \in \mathbb{R}^p_{++}$ *and* $\lambda \geq 0$, *then* $A_{\mathrm{C}}(x) = \max_i \mathrm{ReLU}\big(\lambda_i w_i (g_i - x_i)\big)$*: a ReLU layer followed by max-pooling.*

Proof. The objective of (2) is nondecreasing in each d_i^-, so minimal underachievements $d_i^- = \mathrm{ReLU}(g_i - x_i)$ are optimal as in Proposition 2; positive scaling commutes with ReLU. □

Table 1 collects the correspondence, including the endogenous-priority entries developed next.

4 Endogenous priorities

4.1 Definition

Definition 1 (Endogenous priority map)**.** For $\alpha \in \mathbb{R}$, let $\lambda_\alpha : \mathbb{R}^p_{++} \to \Delta^{p-1}$,

$$\lambda_\alpha(x)_i = \frac{x_i^{\alpha-1}}{\sum_{j=1}^p x_j^{\alpha-1}}, \qquad i = 1, \dots, p.$$

For $\alpha > 1$ the priority of a criterion grows with its attained level; for $\alpha < 1$ it shrinks, so lagging criteria receive more weight; $\alpha = 1$ gives equal priorities. The case $\alpha = 2$, $\lambda_2(x) = x/\mathbf{1}^\top x$, which we call the *proportional rule*, is the alternative-dependent analogue of the self-normalized rule $\lambda = w/\mathbf{1}^\top w$ of Section 4.3 that motivated this study. Define the potential

$$F_\alpha(x) = \begin{cases} \dfrac{1}{\alpha} \sum_{i=1}^p x_i^\alpha, & \alpha \neq 0, \\ \sum_{i=1}^p \log x_i, & \alpha = 0, \end{cases} \qquad \nabla F_\alpha(x) = \big(x_1^{\alpha-1}, \dots, x_p^{\alpha-1}\big).$$

The identity $\lambda_\alpha(x) = \nabla F_\alpha(x)/\|\nabla F_\alpha(x)\|_1$ drives all results below. As a function of α, F_α is discontinuous at $\alpha = 0$; the shifted form $\sum_i (x_i^\alpha - 1)/\alpha$, which converges to $\sum_i \log x_i$, has the same gradient and maximizers and may be used where continuity in α matters. For $\alpha \leq 1$ and up to the reparametrization $\alpha \mapsto 1 - \alpha$, F_α is the α-fair utility of Mo and Walrand (2000); for $\alpha > 1$ it is convex and lies outside that family. Its role as a one-parameter bridge between utilitarian and max–min objectives is standard (Ogryczak et al., 2014; Chen and Hooker, 2023); what is specific to this paper is that F_α arises as the potential of goal priorities generated by the alternative.

Proposition 4 (Endogenous priorities = attention)**.** $\lambda_\alpha(x) = \mathrm{softmax}\big((\alpha - 1)\log x\big)$. *Hence* $\langle \lambda_\alpha(x), x \rangle$ *is a single attention head with keys* $\log x$*, values* x *and inverse temperature* $\alpha - 1$*, which may be negative. As* $\alpha \to +\infty$ *the weights concentrate on the indices attaining* $\max_i x_i$*, and as* $\alpha \to -\infty$ *on those attaining* $\min_i x_i$*, split equally among ties.*

Proof. $\exp((\alpha-1)\log x_i) = x_i^{\alpha-1}$; normalize. For the limits, divide numerator and denominator by $(\max_j x_j)^{\alpha-1}$, respectively $(\min_j x_j)^{\alpha-1}$: the terms of non-extremal indices vanish and each extremal index keeps weight one. □

4.2 Two readings

Because the priorities depend on x, "maximize the endogenously weighted sum" admits two readings.

Definition 2 (Fixed-point and direct readings)**.** The *fixed-point reading* holds priorities fixed while maximizing and iterates:

$$x^{k+1} \in \arg\max_{x \in \mathcal{X}} \langle \lambda_\alpha(x^k), x \rangle. \tag{3}$$

A point x^* is an *endogenous-priority fixed point* if $x^* \in \arg\max_{x \in \mathcal{X}} \langle \lambda_\alpha(x^*), x \rangle$. The *direct reading* maximizes $s_\alpha(x) = \langle \lambda_\alpha(x), x \rangle$ over $\mathcal{X}$, differentiating through the priorities.

Algorithm 1 Endogenous-priority scalarization (fixed-point reading)

Require: $\mathcal{X}$ (convex if $\alpha \leq 1$), exponent α, start $x^0 \in \mathcal{X}$, tolerance τ

for $k = 0, 1, 2, \ldots$ **do**
  $\lambda^k \leftarrow \lambda_\alpha(x^k)$
  $s^k \in \arg\max_{x \in \mathcal{X}} \langle \lambda^k, x \rangle$ ▷ weighted-sum oracle
  if $\langle \lambda^k, s^k - x^k \rangle \leq \tau$ **then** ▷ stationarity gap
    return x^k
  end if
  if $\alpha > 1$ **then**
    $x^{k+1} \leftarrow s^k$ ▷ full step, Theorem 9
  else
    $x^{k+1} \leftarrow x^k + \frac{2}{k+2}(s^k - x^k)$ ▷ damped step, Theorem 11
  end if
end for

Algorithm 1 states the fixed-point reading with an optional damped step. Its linear subproblem is a weighted-sum problem over the original decision set, solvable by any single-objective method; in the production example of Section 7 it is a linear programme. The damped step forms convex combinations of points of $\mathcal{X}$ and is therefore used only when $\mathcal{X}$ is convex; on nonconvex sets only the full step (3) is meaningful.

4.3 The self-normalized priority rule

A natural way to attach priorities to a neuron is to set $\lambda = w/\mathbf{1}^\top w$ for its own weights $w \in \mathbb{R}^p_{++}$, giving the score $\langle \lambda \odot w, x \rangle + b$. The following shows that this is a reparametrization rather than a new model.

Proposition 5 (Self-normalized priorities)**.** *Let $\theta(w) = w \odot w/\mathbf{1}^\top w$ for $w \in \mathbb{R}^p_{++}$. Then:*

1. *θ is a bijection of $\mathbb{R}^p_{++}$ onto itself, with inverse $w_i = \sqrt{\theta_i} \sum_j \sqrt{\theta_j}$. The scores $\{\langle \lambda \odot w, x \rangle + b\}$ therefore form exactly the class of affine neurons with positive weights.*
2. *Relative priorities are squared: $\theta_i/\theta_j = (w_i/w_j)^2$.*
3. *Over a polytope $\mathcal{X}$, the neuron with weights w and the self-normalized score have a common maximizing vertex if and only if w and $w \odot w$ lie in the normal cone of a common vertex.*

Proof. (1) Given $\theta \in \mathbb{R}^p_{++}$ and $S = \mathbf{1}^\top w$, the equations $w_i^2 = \theta_i S$ give $w_i = \sqrt{\theta_i S}$, and summing gives $\sqrt{S} = \sum_j \sqrt{\theta_j}$; the map and its inverse are continuous and positive. (2) is immediate. (3) A positive linear functional is maximized at a vertex v of a polytope exactly when it lies in $N_{\mathcal{X}}(v)$; positive scaling does not change normal-cone membership. □

Self-normalization therefore neither removes the need to choose or learn weights nor adds expressive power. It does move decisions: by part (3), the two scores choose different vertices whenever squaring carries the weight vector across a normal-cone boundary. Section 7 gives a concrete instance. This motivates replacing the self-referential rule by priorities that depend on the alternative, which is the subject of the rest of the paper.

5 Fixed-point theory

Lemma 6 (Normalization invariance)**.** *For any $c > 0$, $\arg\max_{x\in\mathcal{X}}\langle c\lambda, x\rangle = \arg\max_{x\in\mathcal{X}}\langle\lambda, x\rangle$. Hence* (3) *and its fixed points are unchanged if $\lambda_\alpha(x^k)$ is replaced by $\nabla F_\alpha(x^k)$; the normalization affects only reported score values.*

Theorem 7 (Pareto optimality)**.** *Under Assumption 1, every endogenous-priority fixed point is a supported Pareto optimum, for every $\alpha \in \mathbb{R}$.*

Proof. Since $x^* \in \mathbb{R}^p_{++}$, $\lambda^* = \lambda_\alpha(x^*) \in \mathbb{R}^p_{++}$. If some $y \in \mathcal{X}$ dominated x^*, then $\langle\lambda^*, y - x^*\rangle > 0$, contradicting the maximality of x^*. Supportedness holds with $\lambda = \lambda^*$. □

Remark 2. If Assumption 1 is weakened to $\mathcal{X} \subset \mathbb{R}^p_+$, a fixed point with $x^*_i = 0$ and $\alpha > 1$ has $\lambda^*_i = 0$, and only weak Pareto optimality follows (compare $(1, 0)$ with $(1, 0.1)$); for $\alpha < 1$ the priority map is not even defined at such a point. The positivity shift ε is therefore part of the method, not a technicality.

Theorem 8 (Variational characterization)**.** *$x^* \in \mathcal{X}$ is an endogenous-priority fixed point if and only if*

$$\langle\nabla F_\alpha(x^*), x - x^*\rangle \le 0 \qquad \text{for all } x \in \mathcal{X}. \tag{4}$$

If $\mathcal{X}$ is convex, (4) *states $\nabla F_\alpha(x^*) \in N_{\mathcal{X}}(x^*)$, the first-order condition for $\max_{x\in\mathcal{X}} F_\alpha(x)$. In particular, for convex $\mathcal{X}$ every maximizer of F_α is a fixed point, so fixed points exist for every α.*

Proof. $x^* \in \arg\max_{\mathcal{X}}\langle\lambda_\alpha(x^*), x\rangle$ if and only if $\langle\lambda_\alpha(x^*), x - x^*\rangle \le 0$ on $\mathcal{X}$; multiply by $\|\nabla F_\alpha(x^*)\|_1 > 0$. For convex $\mathcal{X}$ the condition is the normal-cone inclusion; if x^* maximizes the differentiable F_α over convex $\mathcal{X}$, then for every $x \in \mathcal{X}$ the function $t \mapsto F_\alpha(x^* + t(x - x^*))$ on $[0, 1]$ is maximized at $t = 0$, so its right derivative there, $\langle\nabla F_\alpha(x^*), x - x^*\rangle$, is nonpositive. Compactness gives a maximizer. □

Theorem 8 is the central structural fact: *on convex problems, endogenous priorities equal to the normalized gradient of a potential make the weighted-sum fixed point coincide with a stationary point of that potential.* Choosing α is choosing the aggregate. Two cautions apply. For $\alpha > 1$, F_α is convex and its stationary points on convex $\mathcal{X}$ include its minimizers as well as its maximizers. For nonconvex $\mathcal{X}$, condition (4) is a global variational inequality rather than stationarity in the usual sense.

Theorem 9 (Case $\alpha > 1$: monotone ascent)**.** *Let $\alpha > 1$. Then F_α is strictly convex on $\mathbb{R}^p_{++}$, the full-step iteration* (3) *satisfies*

$$F_\alpha(x^{k+1}) - F_\alpha(x^k) \;\ge\; \|\nabla F_\alpha(x^k)\|_1\,\gamma_k \;\ge\; 0, \qquad \gamma_k = \max_{x\in\mathcal{X}}\langle\lambda_\alpha(x^k), x - x^k\rangle, \tag{5}$$

$\{F_\alpha(x^k)\}$ converges, and the stationarity gap $\gamma_k \to 0$, so Algorithm 1 terminates for every $\tau > 0$. If $\mathcal{X}$ is finite, or a polytope and the oracle returns vertices, the iteration reaches a fixed point in finitely many steps. Moreover the maximum of F_α over a compact convex $\mathcal{X}$ is attained at an extreme point.

Proof. Each $t \mapsto t^\alpha/\alpha$ is strictly convex for $\alpha > 1$. Convexity gives $F_\alpha(x^{k+1}) \geq F_\alpha(x^k) + \langle \nabla F_\alpha(x^k), x^{k+1} - x^k \rangle$, and the inner product equals $\|\nabla F_\alpha(x^k)\|_1 \gamma_k \geq 0$ because x^{k+1} maximizes $\langle \lambda_\alpha(x^k), \cdot \rangle$ over a set containing x^k. A bounded monotone sequence converges. Since $\mathcal{X} \subset \mathbb{R}^p_{++}$ is compact, $\|\nabla F_\alpha\|_1$ is bounded below by a positive constant on $\mathcal{X}$, so summability of the increments forces $\gamma_k \to 0$. If the candidate set is finite (all of $\mathcal{X}$, or the vertices of a polytope for $k \geq 1$), strict convexity implies that any step with $x^{k+1} \neq x^k$ strictly increases F_α, so no candidate repeats and the iteration stops at a fixed point. A convex function on a compact convex set attains its maximum at an extreme point (Rockafellar, 1970, Section 32). □

Corollary 10. *Let $\alpha > 1$ and let the oracle return extreme points. Then every iterate x^k, $k \geq 1$, is an extreme point of $\mathcal{X}$, and F_α increases along the iterates toward points far from the origin of the normalized space (in ℓ_2 when $\alpha = 2$). Fixed points, however, need not be extreme: on the closed segment joining $(0.1, 1.1)$ and $(1.1, 0.1)$ with $\alpha = 2$, the midpoint $(0.6, 0.6)$ has equal priorities, zero gap and is a fixed point, although it minimizes F_2 on the segment. The point reached therefore depends on x^0.*

Theorem 11 (Case $\alpha < 1$: unique compromise)**.** *Let $\alpha < 1$ and $\mathcal{X}$ be convex. Then F_α is strictly concave on $\mathbb{R}^p_{++}$, the endogenous-priority fixed point is unique and equals $\arg\max_{\mathcal{X}} F_\alpha$, and the damped iteration of Algorithm 1 satisfies $F_\alpha(x^*) - F_\alpha(x^k) \leq 2LD^2/(k+2)$ for all $k \geq 1$, where D is the diameter of $\mathcal{X}$ and L a Lipschitz constant of ∇F_α on $\mathcal{X}$.*

Proof. For $\alpha < 1$, $\alpha \neq 0$, the second derivative of t^α/α is $(\alpha - 1)t^{\alpha-2} < 0$; for $\alpha = 0$, $\log$ is strictly concave. A separable sum of strictly concave functions is strictly concave, so the maximizer over compact convex $\mathcal{X}$ is unique, and by Theorem 8 stationary points of a concave programme are its maximizers. By Lemma 6 the damped iteration is the Frank–Wolfe method (Frank and Wolfe, 1956) applied to minimizing the convex function $-F_\alpha$ over $\mathcal{X}$. Because $\mathcal{X} \subset \mathbb{R}^p_{++}$ is compact, ∇F_α is Lipschitz on $\mathcal{X}$, the curvature constant is at most LD^2, and the standard rate for step size $2/(k+2)$ applies for $k \geq 1$ (Jaggi, 2013). □

Remark 3. Damping is required for $\alpha < 1$, even on convex sets. On the segment joining $(0.1, 1.1)$ and $(1.1, 0.1)$ with $\alpha = 0$, the full step from either endpoint moves to the other endpoint, because the priorities favour the lagging coordinate, and the iteration alternates indefinitely; the damped iteration converges to the unique fixed point $(0.6, 0.6)$.

Proposition 12 (Limits in α)**.** *1. $\alpha = 1$: $\lambda_1 \equiv \mathbf{1}/p$, the equal-weight sum.*

2. *$\alpha = 0$: $F_0 = \sum_i \log x_i$; for convex $\mathcal{X}$ the fixed point is the Nash bargaining solution with disagreement point 0 (Nash, 1950), i.e. proportional fairness (Kelly et al., 1998). For nonconvex $\mathcal{X}$ the Nash point may be unsupported and then is not a fixed point (Theorem 13).*

3. *$\alpha \to -\infty$: let $M_\alpha(x) = \left(\frac{1}{p}\sum_i x_i^\alpha\right)^{1/\alpha}$. For $\alpha < 0$, $\arg\max_{\mathcal{X}} F_\alpha = \arg\max_{\mathcal{X}} M_\alpha$, and every cluster point of maximizers x^*_α of F_α as $\alpha \to -\infty$ maximizes $\min_i x_i$ over $\mathcal{X}$, the equal-weight Chebyshev solution, which may be only weakly Pareto optimal. For convex $\mathcal{X}$ these maximizers are the fixed points (Theorem 11).*

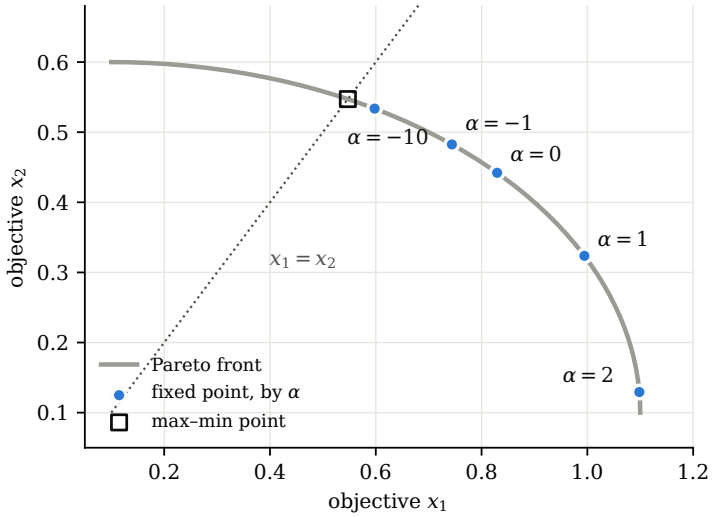


Figure 1: Endogenous-priority fixed points on the convex front $x = (0.1 + \cos t,\ 0.1 + 0.5 \sin t)$, $t \in [0, \pi/2]$. As α decreases from 2 to -10 the selected point moves from an extreme point toward the max–min point (square), as predicted by Theorem 9 and Proposition 12. Because the front is not symmetric, the $\alpha = -10$ point is close to, but distinct from, the max–min point.

Proof. (1) and (2) are immediate. (3) For $\alpha < 0$, F_α is maximized iff $\sum_i x_i^\alpha$ is minimized, iff M_α is maximized, since $u \mapsto (u/p)^{1/\alpha}$ is decreasing. Because x_i^α is decreasing in x_i, $(\min_i x_i)^\alpha \le \sum_i x_i^\alpha \le p(\min_i x_i)^\alpha$, whence $\min_i x_i \le M_\alpha(x) \le p^{-1/\alpha} \min_i x_i$. Thus $M_\alpha \to \min_i x_i$ uniformly on compact $\mathcal{X}$, and cluster points of maximizers of uniformly convergent continuous functions maximize the limit. □

Theorem 13 (Unsupported points are not fixed points)**.** *For every $\alpha \in \mathbb{R}$, the endogenous-priority fixed points are supported Pareto points. If the Pareto front of a nonconvex $\mathcal{X}$ contains unsupported points, none of them is a fixed point, even when it maximizes F_α over $\mathcal{X}$; and fixed points need not exist.*

Proof. The first two claims follow from Theorem 7. For non-existence, take the two-point set $\mathcal{X} = \{(1.1, 0.1), (0.1, 1.1)\}$ and $\alpha = 0$: at each point the priority vector favours the other point's large coordinate, so the iteration alternates and neither point is fixed. □

In the fixed-point reading, endogenous priorities therefore change *which* supported point is selected, not *which points are reachable*. The direct reading of Section 6 is not subject to this limitation: on the nonconvex front of Section 7, the maximizers of L_0 and $L_{1/2}$ are the unsupported Pareto points $(0.455, 0.517)$ and $(0.415, 0.569)$; the latter is the Nash point, as Remark 6 explains. Figure 1 shows the selection on a convex front: as α decreases the fixed point moves from an extreme point toward the max–min point.

Remark 4 (Reference-point dependence). Fixed points are not invariant to affine shifts of criterion space, since $\nabla F_\alpha(x + c\mathbf{1}) \neq \nabla F_\alpha(x)$ in general. On a quarter circle centred at the origin every front point has the same norm and is a fixed point for $\alpha = 2$; after a shift by $0.1\,\mathbf{1}$ the iteration selects the midpoint (Table 2). The normalization of Assumption 1, including the ref-

erence point, must therefore be stated as part of the method, as with reference-point methods generally (Wierzbicki, 1980).

6 The direct reading

A decision maker who accepts endogenous priorities may also ask directly for the alternative with the largest self-weighted score, rather than for a fixed point. This is the direct reading of Definition 2, and it is also what a network computes when the attention weights are not held fixed. Its score is

$$s_\alpha(x) = \langle \lambda_\alpha(x), x \rangle = \frac{\sum_i x_i^\alpha}{\sum_i x_i^{\alpha-1}} = L_\alpha(x),$$

the Lehmer mean of order α (Lehmer, 1971; Bullen, 2003), which is the mixture function (Špirková et al., 2019) with weighting function $t^{\alpha-1}$. In particular L_0 is the harmonic mean, L_1 the arithmetic mean and L_2 the contraharmonic mean.

The monotonicity of Lehmer means is classical. In the parametrization $\sum_i x_i^{q+1} / \sum_i x_i^q$ of the Gini-mean literature, which is our $\alpha = q + 1$, the Lehmer mean is monotone for $q \in [-1, 0]$ (Farnsworth and Orr, 1986), and outside this range it is at best weakly monotone (Wilkin and Beliakov, 2015). The proposition below restates this fact in our notation, with a short proof for completeness; its content for this paper is the consequence for Pareto optimality.

Proposition 14 (Pareto compliance of the direct score; cf. Farnsworth and Orr, 1986)**.** *On $\mathbb{R}^p_{++}$ with $p \geq 2$, L_α is strictly increasing in every coordinate if and only if $\alpha \in [0, 1]$. Consequently, for $\alpha \in [0, 1]$ every maximizer of s_α over any $\mathcal{X} \subset \mathbb{R}^p_{++}$ is Pareto optimal, while for $\alpha \notin [0, 1]$ there exist pairs $y \succ x$ in $\mathbb{R}^p_{++}$ with $s_\alpha(y) < s_\alpha(x)$, so that on $\mathcal{X} = \{x, y\}$ the maximizer of s_α is dominated.*

Proof. Write $N = \sum_j x_j^\alpha$ and $D = \sum_j x_j^{\alpha-1}$. Then

$$\frac{\partial L_\alpha}{\partial x_i} = \frac{x_i^{\alpha-2}}{D^2}\big(\alpha x_i D - (\alpha - 1)N\big).$$

For $\alpha \in (0, 1)$ both $\alpha x_i D$ and $-(\alpha - 1)N$ are positive; for $\alpha = 0$ the bracket is $N > 0$; for $\alpha = 1$ it is $x_i D > 0$. A strictly increasing function cannot be maximized at a dominated point. For $\alpha > 1$, fix x_j, $j \neq i$, and let $x_i \to 0^+$: then $x_i D \to 0$ while $(\alpha - 1)N$ stays bounded away from zero, so the derivative is eventually negative. For $\alpha < 0$, let $x_i \to \infty$: D tends to a positive constant, so $\alpha x_i D \to -\infty$ while $(\alpha - 1)N$ stays bounded, and again the derivative is eventually negative. Increasing x_i on such an interval produces a dominating point with a strictly lower score. □

Remark 5 (Bounded criterion ranges). The "only if" part of Proposition 14 uses unbounded ratios between criteria. When criteria are confined to a box, as after the normalization of Assumption 1, the compliant range is wider: numerically, L_α is monotone on $[0.1, 1.1]^p$ for $\alpha \in [-0.167, 1.167]$ when $p = 2$ and $\alpha \in [-0.136, 1.136]$ when $p = 3$. The interval $[0, 1]$ is the range that is safe for every box.

Remark 6 (The two readings select different compromises). The direct reading maximizes L_α, the fixed-point reading maximizes F_α (on convex $\mathcal{X}$), and these generally differ. For example, $L_{1/2}(x) = \sqrt{x_1 x_2}$ when $p = 2$, so the Nash point is the direct-reading solution at $\alpha = \frac{1}{2}$, whereas the direct reading at $\alpha = 0$ maximizes the harmonic mean L_0, which is neither the Nash point nor invariant to rescaling individual criteria.

Proposition 14 concerns rankings. A non-monotone score can still have Pareto-optimal maximizers on a given $\mathcal{X}$; Section 7 shows that this happens in the production example, while rankings of non-optimal alternatives are frequently reversed. The risk is therefore greatest where scores are used to rank or select among many alternatives, as in population-based search or in screening many candidate portfolios, and smaller where only the optimum is reported.

7 Numerical experiments

Computational setup. All computations use Python 3 with NumPy and SciPy, whose interface to the HiGHS solver is used for linear and mixed-integer programmes. The scripts that produce every number, table and figure accompany the submission.

Use of a large language model. In accordance with the journal's policy, the use of a large language model in this research is documented here. The author used Claude (Opus 5.5, Anthropic) as a research assistant. The author conceived the research question and proposed the self-normalized priority rule that motivated the study. The model was used to analyse that rule and develop its generalization through the exponent α; to draft the statements and proofs of the propositions and theorems; to write and run the Python code for all experiments, verification checks and figures; to draft and revise the text; and to search for and check the literature. A separate instance of the model, without access to the drafting process, re-derived the proofs and reproduced the reported numbers as an independent check, and the manuscript was revised in response to its reports. The author has reviewed all derivations, code and text and takes full responsibility for the content of the paper.

Pareto certificates. Pareto optimality of a computed point $\hat{x}$ is certified by the linear programme $\max\{\mathbf{1}^\top s : x \in \mathcal{X},\ x \geq \hat{x} + s,\ s \geq 0\}$, whose optimal value is zero exactly when $\hat{x}$ is Pareto optimal.

7.1 Verification of the correspondence

Propositions 2 and 3 were checked on 500 random instances with $p \in \{2, \ldots, 6\}$, random criteria, aspirations, priorities and scales. Each goal programme was solved as a linear programme and compared with the network output; the largest absolute difference was 1.3×10^{-15} for the weighted and 0 for the Chebyshev case. Proposition 4 holds to machine precision. A finite-difference search over 2×10^4 random points per exponent found no negative partial derivative of L_α for $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and found negative derivatives for $\alpha \in \{-10, -2, -1, -0.5, 1.5, 2, 5\}$, as Proposition 14 predicts.

Table 2: Points reached on shifted bi-objective fronts (five starting points each). Convex fronts: Algorithm 1 (full step for $\alpha = 2$, damped step for $\alpha \leq 0$). Nonconvex front: full-step iteration (3), since the damped step leaves a nonconvex $\mathcal{X}$.

Front	α	Point reached	$\arg\max_{\mathcal{X}} F_\alpha$
Linear	2	$(0.1, 1.1)$ or $(1.1, 0.1)$, start-dependent	extreme points
	0	$(0.600, 0.600)$	$(0.6, 0.6)$
	-10	$(0.600, 0.600)$	$(0.6, 0.6)$
Circular	2	$\approx (0.807, 0.807)$	$(0.807, 0.807)$
	0	$(0.807, 0.807)$	$(0.807, 0.807)$
	-10	$(0.807, 0.807)$	$(0.807, 0.807)$
Nonconvex	2	$(0.1, 1.1)$ or $(1.1, 0.1)$	extreme points
	0	alternates between extremes	$(0.415, 0.569)$, unsupported
	-10	alternates between extremes	$(0.478, 0.487)$, unsupported

7.2 Bi-objective test fronts

Three discretized fronts with 4001 points each, shifted by $\varepsilon = 0.1$, were used: linear ($x_1 + x_2 = 1$), circular ($x_1^2 + x_2^2 = 1$) and nonconvex ($x_2 = (1 - x_1)^2$). Five starting points were taken along each front. Table 2 reports the points reached.

On the convex fronts the damped iteration reached $\arg\max F_\alpha$ from every start (Theorem 11); because these two fronts are symmetric, for $\alpha \leq 0$ this point also coincides with the max–min point, which is not true in general (Figure 1). For $\alpha = 2$ the outcome depends on the start on the linear front, while on the shifted circle it is the midpoint (Remark 4). On the nonconvex front the maximizers of F_0 and F_{-10} are unsupported and the full-step iteration alternates between the two extremes (Theorem 13); the max–min point $(0.482, 0.482)$ is reachable only by a nonlinear scalarization. (Applied there, the damped step would converge to the chord midpoint $(0.6, 0.6)$, which is not in $\mathcal{X}$.) All fixed points reached were certified Pareto optimal.

7.3 A three-objective production problem

A firm chooses quantities $(a, b, c) \geq 0$ of three products subject to machine time $a + 2b + c \leq 40$, material $3a + b + 2c \leq 60$ and demand $c \leq 15$. It maximizes profit $P = 5a + 4b + 3c$ and employment (labour hours) $H = 2a + 3b + 4c$, and minimizes emissions $E = 4a + 2b + c$. The payoff table gives ideal $(128, 101, 0)$ and nadir $(0, 0, -88)$ for $(P, H, -E)$; with $\varepsilon = 0.1$ the normalized criteria are $x_1 = P/128 + 0.1$, $x_2 = H/101 + 0.1$ and $x_3 = 1.1 - E/88$. Since the criteria are already normalized, scales are unit ($w = \mathbf{1}$); priorities are $\lambda = (0.5, 0.3, 0.2)$ and every aspiration is $g_i = 0.9$. The neuron therefore has weights $\theta = \lambda$, and the self-normalized score of Proposition 5 has weights $\lambda \odot \lambda / \mathbf{1}^\top \lambda$. The scores compared are

$$
\begin{aligned}
S_{\mathrm{NN}} &= 0.5x_1 + 0.3x_2 + 0.2x_3 - 0.9 && \text{(neuron, Prop. 1)},\\
S_{\mathrm{SN}} &= 0.25x_1 + 0.09x_2 + 0.04x_3 - 0.342 && \text{(self-normalized, Prop. 5)},\\
S_{\mathrm{GP}} &= -\textstyle\sum_i \lambda_i \,\mathrm{ReLU}(0.9 - x_i) && \text{(weighted goal programme, Prop. 2)},
\end{aligned}
$$

Table 3: Production plans selected by each scalarization. Emissions are to be minimized. All plans were certified Pareto optimal.

Method	Plan (a, b, c)	Profit	Labour	Emissions
Neuron, S_{NN}	$(7, 9, 15)$	116.0	101.0	61.0
Self-normalized, S_{SN}	$(7, 9, 15)$	116.0	101.0	61.0
Weighted goal programme	$(2.47, 11.27, 15)$	102.4	98.7	47.4
Chebyshev goal programme	$(0, 11.41, 15)$	90.6	94.2	37.8
Endogenous, $\alpha = 2$	$(7, 9, 15)$ or $(0, 0, 0)$	116.0 or 0	101.0 or 0	61.0 or 0
Endogenous, $\alpha = 0$	$(0, 12.5, 15)$	95.0	97.5	40.0
Endogenous, $\alpha = -10$	$(0, 9.31, 15)$	82.2	87.9	33.6
Max–min (Chebyshev, equal weights)	$(0, 8.86, 15)$	80.4	86.6	32.7

together with the Chebyshev goal programme with the same priorities, the endogenous-priority fixed points for $\alpha \in \{2, 0, -10\}$ and the max–min solution. Table 3 reports the plans.

Four observations follow.

Neuron and self-normalized score. Both are linear with positive coefficients and select the same vertex here: in plan coordinates $S_{\mathrm{NN}} = 0.0164a + 0.0200b + 0.0213c - 0.6$ and $S_{\mathrm{SN}} = 0.0097a + 0.0096b + 0.0090c - 0.264$. Their rankings are not equivalent, however. Over 2000 plans drawn uniformly from the feasible set (by rejection sampling from the box $[0, 20]^2 \times [0, 15]$), Kendall's rank correlation between them is 0.86, and they disagree on 7.0% of plan pairs. Shifting priority toward emissions, with $\lambda = (0.6(1 - e), 0.4(1 - e), e)$ the two scores select different plans on two intervals, whose end points (computed exactly from the vertex comparisons) are given to three decimals. For $e \in (0.320, 0.342)$ the neuron already selects $(0, 12.5, 15)$ while the self-normalized score keeps $(7, 9, 15)$. For $e \in (0.456, 0.725)$ the self-normalized score cuts production first to $(0, 0, 15)$ and, from $e = 0.533$, to nothing at all, while the neuron selects $(0, 12.5, 15)$ until $e = 0.574$ and $(0, 0, 15)$ thereafter. This is part (3) of Proposition 5.

Goal programming. The one-sided penalty stops rewarding profit once its aspiration is met, so the weighted goal programme trades 12% of profit for 22% lower emissions relative to the neuron.

Endogenous priorities. With $\alpha = 2$ the outcome depends on the start: three of four starts led to $(7, 9, 15)$ and one to producing nothing, the emissions-only extreme point (Theorem 9). With $\alpha = 0$ every start reached $(0, 12.5, 15)$, matching direct maximization of $\sum_i \log x_i$ to 2×10^{-7}; this plan cuts emissions by 34% relative to the neuron at a cost of 18% of profit, and requires neither priorities nor aspirations. With $\alpha = -10$ the plan approaches the max–min solution; the normalized criteria are $(0.742, 0.970, 0.718)$ against $(0.728, 0.957, 0.728)$, consistent with convergence only in the limit (Proposition 12).

Direct reading. Maximizing L_α by sequential quadratic programming from 40 random starts gave only Pareto-optimal local maxima for $\alpha \in \{2, 0.5, -10\}$. Rankings were another matter. Among 3000 plans drawn uniformly as above, 1284 (43%) were scored above some plan that dominates them when $\alpha = -10$, 49 (2%) when $\alpha = 2$, and none when $\alpha = 0.5$, in line with Proposition 14. For example, under $\alpha = 2$ the plan with normalized criteria $(0.260, 0.255, 0.964)$ scores $L_2 = 0.718$, above the dominating plan $(0.301, 0.363, 0.964)$ with $L_2 = 0.707$: L_2 decreases in x_i wherever x_i lies below half the score, which is the case for profit and employment here.

7.4 Real data: conservation projects for the REPI programme

The U.S. Department of Defense's Readiness and Environmental Protection Integration (REPI) programme funds land conservation around military installations. Messer et al. (2015) report the 44 projects considered in 2010, each with a military readiness score r_i, an environmental protection score e_i, a viability-of-agreement score v_i and a REPI funding request c_i, against a budget $W = \$54$ million (their Table 3). A portfolio is a set S of projects with $\sum_{i\in S} c_i \le W$, and $R(S) = \sum_{i\in S} r_i$, $E(S) = \sum_{i\in S} e_i$, $V(S) = \sum_{i\in S} v_i$. Because decisions are binary, the attainable set is finite and nonconvex: Theorems 7, 8 and 13 apply, whereas Theorem 11 and the damped step of Algorithm 1 do not, so the full step (3) is used.

Formulations. The single-objective maxima are $R^* = 1004.5$, $E^* = 368.25$ and $V^* = 645$, in agreement with the percentages in Table 6 of Messer et al. (2015). Criteria are normalized as achievement ratios $x(S) = \big(R(S)/R^*,\ E(S)/E^*\big) \in (0,1]^2$, which satisfy Assumption 1 without a shift. Five rules are compared, each solved exactly by mixed-integer programming:

(i) benefit targeting (BT), REPI's ranking rule, as reported by Messer et al. (2015);

(ii) binary linear programming (BLP): $\max\ R(S) + E(S) + V(S)$;

(iii) weighted goal programming, equation (9) of Messer et al. (2015) with $\lambda = \frac{1}{2}$,

$$\min_{S}\ \tfrac{1}{2}\,\mathrm{ReLU}\big(1 - R(S)/R^*\big) + \tfrac{1}{2}\,\mathrm{ReLU}\big(1 - E(S)/E^*\big),$$

which is the network of Proposition 2;

(iv) Chebyshev goal programming with the same weights,

$$\min_{S}\ \max\big\{\tfrac{1}{2}\,\mathrm{ReLU}(1 - R(S)/R^*),\ \tfrac{1}{2}\,\mathrm{ReLU}(1 - E(S)/E^*)\big\},$$

the max-pooling network of Proposition 3;

(v) endogenous priorities: $S^{k+1} \in \arg\max_S \langle \lambda_\alpha(x(S^k)), x(S)\rangle$, where each step is a 0–1 knapsack problem.

The data and the published selections reproduce the totals of Messer et al. (2015): 1612.5 for BT and 1951.5 for BLP. For goal programming, the selection in their Table 3 and their Table 5 at $\lambda = 0.5$ give $R = 985.5$ and $E = 366.5$ (their Table 4 lists 992 and 369).

The exact Pareto front. Solving $\max R(S)$ subject to $E(S) \ge \varepsilon$ for every attainable ε (scores are multiples of 0.25) yields 14 nondominated (R, E) portfolios. A linear-programming test of whether some strictly positive weight vector selects each point shows that 10 are supported (one of them only weakly, on an edge of the upper hull) and 4 are unsupported (Figure 2).

Table 4 and Figure 2 support four observations.

No weights are needed. Started from the readiness-maximizing, the environment-maximizing and the BLP portfolios, the endogenous-priority iteration reached a fixed point within two steps for each of $\alpha = 2$, 0 and -10, and every run ended at the same portfolio. That portfolio is the one weighted goal programming selects with $\lambda = \frac{1}{2}$, and it also maximizes the Nash product $x_1 x_2$ and F_{-10} over all 14 Pareto portfolios. Messer et al. (2015) had to choose λ and then examine 21 values of it in a sensitivity analysis; here the priorities are generated by the portfolio itself.

Table 4: REPI 2010 portfolios ($W = \$54\,\text{M}$). R/R^* and E/E^* are achievement ratios; "weaker" is $\min(R/R^*, E/E^*)$; Nash is $(R/R^*)(E/E^*)$. Bold marks the best value in each column.

Rule	$\lvert S\rvert$	R	E	V	$R{+}E{+}V$	R/R^*	E/E^*	weaker	Nash
(i) BT	19	784.5	299.0	529	1612.5	0.781	0.812	0.781	0.634
(ii) BLP, max $R{+}E{+}V$	25	970.5	356.0	**625**	**1951.5**	0.966	0.967	0.966	0.934
(iii) Weighted GP, $\lambda = \frac{1}{2}$	25	985.5	**366.5**	569	1921.0	0.981	**0.995**	0.981	**0.976**
(iv) Chebyshev GP, $\lambda = \frac{1}{2}$	25	**990.5**	363.5	558	1912.0	**0.986**	0.987	**0.986**	0.973
(v) Endogenous, $\alpha \in \{2, 0, -10\}$	25	985.5	**366.5**	569	1921.0	0.981	**0.995**	0.981	**0.976**

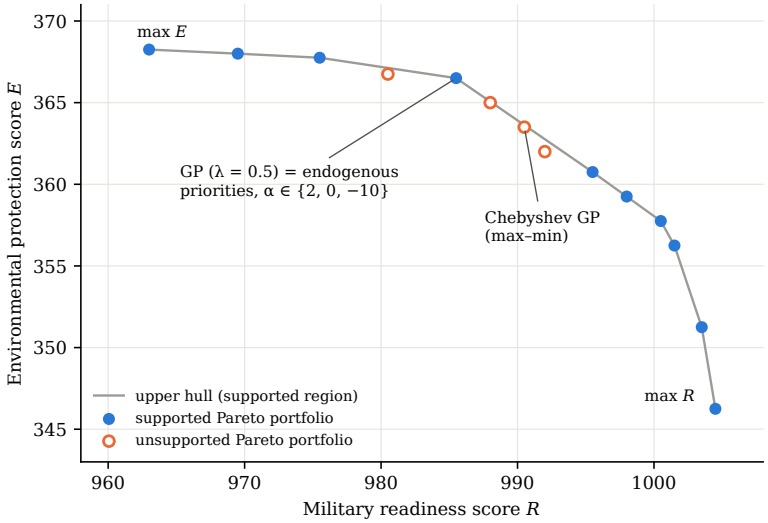


Figure 2: Exact readiness–environment Pareto front of the REPI 2010 projects under the \$54 million budget. Endogenous priorities and weighted goal programming select the same supported portfolio; the max–min portfolio chosen by Chebyshev goal programming is unsupported and cannot be a fixed point for any α (Theorem 13).

The unsupported max–min portfolio is out of reach. Chebyshev goal programming selects $(R, E) = (990.5, 363.5)$, which raises the weaker achievement ratio from 0.981 to 0.986. This portfolio is unsupported, so by Theorem 13 no endogenous-priority fixed point can equal it, and indeed the $\alpha = -10$ iteration stops at the supported portfolio instead. The real data thus exhibit the limitation that motivates the Chebyshev inner step proposed in Section 8.

The total score hides the trade-off. BLP maximizes $R + E + V$ but achieves lower readiness and lower environmental protection than both goal programmes, gaining instead on viability of agreement, a criterion that is not one of REPI's two statutory objectives.

Three criteria. Adding viability, with $x = (R/R^*, E/E^*, V/V^*)$, the fixed point for $\alpha = 0$ and $\alpha = -10$ is $(R, E, V) = (958, 363.25, 625)$, which coincides with equal-weight three-criterion goal programming and attains an average achievement of 96.97%, the best overall percentage reported in Table 6 of Messer et al. (2015). BLP and Chebyshev goal programming select $(970.5, 356, 625)$, with total 1951.5 and the higher minimum achievement (96.62% against 95.37%).

The readiness–environment front is flat: every Pareto portfolio attains at least 94% of each target. This is why all values of α select the same portfolio here; a dataset with a sharper

trade-off would separate the members of the α family more clearly.

8 Discussion

What the unification delivers. The correspondence of Section 3 lets results move in both directions. Goal-programming structure gives interpretable network parameters: readout weights are priorities, biases are aspirations, and input scales are normalization constants. Network structure gives goal programming a vocabulary for endogenous priorities, namely attention, and a question with a precise answer: whether the self-weighted score may be optimized directly without violating Pareto dominance (Proposition 14). The ingredients themselves—the α-fair family, mixture-type weights, the monotonicity of Lehmer means and the Frank–Wolfe method—are established; the unification lies in showing that goal programming, attention-weighted networks and this fairness scale are one structure, and in stating exactly what each reading of that structure guarantees.

Choosing α and the reading. If the purpose is a compromise solution, the fixed-point reading with $\alpha \leq 0$ is the relevant choice: on convex problems $\alpha = 0$ gives the Nash bargaining solution, which is invariant to positive rescaling of individual criteria, and large negative α approximates max–min. If the self-weighted score is to be optimized directly, the direct reading with $\alpha \in [0, 1]$ is Pareto-compliant, but it maximizes the Lehmer mean L_α rather than F_α, so its compromise differs (Remark 6); for $p = 2$ the Nash point is recovered at $\alpha = \frac{1}{2}$, not at $\alpha = 0$. Values $\alpha > 1$, including the proportional rule $\lambda_2(x) = x/\mathbf{1}^\top x$, ascend a convex aggregate, produce extreme-point iterates and depend on the starting point, and should be used only when an extreme solution is the intent.

Limitations. Three limitations are structural. First, in the fixed-point reading unsupported Pareto points are never selected, for any α (Theorem 13); this is the weighted-sum limitation inherited by the linear inner problem. The direct reading can reach such points, and does so with Pareto compliance for $\alpha \in [0, 1]$. Second, fixed points depend on the normalization and reference point (Remark 4). Third, the guarantees of Theorem 11 assume convexity; on nonconvex problems only the variational inequality (4) characterizes fixed points, and fixed points may fail to exist.

Extensions. The first limitation suggests replacing the inner weighted sum by a weighted Chebyshev problem, $x^{k+1} \in \arg\max_{x \in \mathcal{X}} \min_i x_i/\lambda_\alpha(x^k)_i$, whose solutions include unsupported points; its fixed-point theory is open. A second extension is to learn α, or a separate exponent per criterion, from observed choices, which by Proposition 2 is an inverse goal-programming problem posed as network training. A third is machine learning, where task weights computed from the tasks' own losses raise the analogous question of whether such weights should be held fixed or optimized through; the task-weighting schemes of Navon et al. (2022) and Ban and Ji (2024) apply bargaining and α-fair utilities to gradient inner products rather than to criterion values, and are not covered by the present analysis.

9 Conclusion

Weighted and Chebyshev goal programming are ReLU networks with nonnegative readout and max-pooling respectively, and the neuron is a linear achievement function whose bias is minus the weighted aspiration. Making the goal priorities endogenous, $\lambda_\alpha(x) \propto x^{\alpha-1}$, turns the weighted sum into softmax attention whose fixed points are supported Pareto optima and, on convex problems, stationary points of the α-fair aggregate F_α, so that one exponent places goal programming on the established fairness scale from the equal-weight sum through Nash bargaining to max–min compromise, extended beyond it to extreme-point ascent. Maximizing the self-weighted score directly is Pareto-compliant on the positive orthant exactly when $0 \leq \alpha \leq 1$, and selects a different compromise from the fixed point. A self-normalized priority rule adds no expressive power but squares relative priorities. The framework is therefore best read as a single scale on which weighted sums, goal programming, fairness criteria and attention can be compared, with exact statements of what each guarantees.

Acknowledgments

The author thanks Professor Kent D. Messer, whose work with colleagues on goal programming for the Department of Defense's REPI programme inspired this study and provided the data used in Section 7.4.

Statements and Declarations

Competing interests. The author has no competing interests to declare that are relevant to the content of this article.

Funding. The author did not receive support from any organization for the submitted work.

Data availability. The REPI project data analysed in Section 7.4 are published in Table 3 of Messer et al. (2015). All other data are generated by the scripts described below.

Code availability. The Python scripts that reproduce every number, table and figure are provided with the submission and are available from the author on request.

Author contributions. M. Bajpai is the sole author. He conceived the study, proposed the priority rule that motivated it, directed and verified the analysis, interpreted the results and approved the final manuscript. The use of a large language model is documented in Section 7.

Ethics approval and consent. Not applicable.

References

R. Anderson, J. Huchette, W. Ma, C. Tjandraatmadja, and J. P. Vielma. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, 183: 3–39, 2020. doi: 10.1007/s10107-020-01474-5.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

H. Ban and K. Ji. Fair resource allocation in multi-task learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 2715–2731, 2024.

D. Bertsimas, V. F. Farias, and N. Trichakis. The price of fairness. *Operations Research*, 59(1): 17–31, 2011. doi: 10.1287/opre.1100.0865.

V. J. Bowman, Jr. On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In H. Thiriez and S. Zionts, editors, *Multiple Criteria Decision Making*, volume 130 of *Lecture Notes in Economics and Mathematical Systems*, pages 76–86. Springer, 1976.

P. S. Bullen. *Handbook of Means and Their Inequalities*. Kluwer, 2003.

A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*. Wiley, 1961.

A. Charnes, W. W. Cooper, and R. O. Ferguson. Optimal estimation of executive compensation by linear programming. *Management Science*, 1(2):138–151, 1955. doi: 10.1287/mnsc.1.2.138.

V. X. Chen and J. N. Hooker. A guide to formulating fairness in an optimization model. *Annals of Operations Research*, 326(1):581–619, 2023. doi: 10.1007/s10479-023-05264-y.

I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14 (1):63–69, 1997. doi: 10.1007/BF01197559.

J.-A. Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathématique*, 350(5–6):313–318, 2012. doi: 10.1016/j.crma.2012.03.014.

M. Ehrgott. *Multicriteria Optimization*. Springer, 2nd edition, 2005.

Z. Fan, N. Peng, M. Tian, and B. Fain. Welfare and fairness in multi-objective reinforcement learning. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1991–1999. IFAAMAS, 2023.

D. L. Farnsworth and R. K. Orr. Gini means. *The American Mathematical Monthly*, 93(8): 603–607, 1986.

M. Fischetti and J. Jo. Deep neural networks and mixed integer linear optimization. *Constraints*, 23(3):296–309, 2018. doi: 10.1007/s10601-018-9285-6.

R. B. Flavell. A new goal programming formulation. *Omega*, 4(6):731–732, 1976.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1–2):95–110, 1956. doi: 10.1002/nav.3800030109.

K. Fukushima. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333, 1969. doi: 10.1109/TSSC.1969.300225.

A. M. Geoffrion. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22(3):618–630, 1968. doi: 10.1016/0022-247X(68)90201-1.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, 2011.

R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405 (6789):947–951, 2000. doi: 10.1038/35016072.

A. S. Householder. A theory of steady-state activity in nerve-fiber networks: I. Definitions and preliminary lemmas. *Bulletin of Mathematical Biophysics*, 3(2):63–69, 1941. doi: 10.1007/BF02478220.

M. Jaggi. Revisiting Frank–Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 427–435, 2013.

M. Jiménez, A. Bilbao-Terol, and M. Arenas-Parra. Incorporating preferential weights as a benchmark into a sequential reference point method. *European Journal of Operational Research*, 291(2):575–585, 2021. doi: 10.1016/j.ejor.2020.01.019.

D. Jones and M. Tamiz. *Practical Goal Programming*. Springer, 2010.

F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49 (3):237–252, 1998. doi: 10.1057/palgrave.jors.2600523.

C. W. Kim, J. Moondra, S. Verma, M. Pollack, L. Kong, M. Tambe, and S. Gupta. Navigating the social welfare frontier: Portfolios for multi-objective reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 30631–30653. PMLR, 2025.

I. Y. Kim and O. L. de Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, 29(2):149–158, 2005.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, 2012.

T. Lan, D. Kao, M. Chiang, and A. Sabharwal. An axiomatic theory of fairness in network resource allocation. In *Proceedings of IEEE INFOCOM 2010*, pages 1–9, 2010. doi: 10.1109/INFCOM.2010.5461911.

D. H. Lehmer. On the compounding of certain means. *Journal of Mathematical Analysis and Applications*, 36(1):183–200, 1971.

K. D. Messer, M. Kecinski, Z. Liu, M. A. Korch, and T. Bounds. Military readiness and environmental protection through cost-effective land conservation. APEC Research Report RR15-03, Department of Applied Economics and Statistics, University of Delaware, 2015.

K. D. Messer, M. Kecinski, Z. Liu, M. A. Korch, and T. Bounds. Military readiness and environmental protection through cost-effective land conservation. *Land Economics*, 92(3):433–449, 2016. doi: 10.3368/le.92.3.433.

K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, 1999.

J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, 2000. doi: 10.1109/90.879343.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, 2010.

J. F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

A. Navon, A. Shamsian, I. Achituve, H. Maron, K. Kawaguchi, G. Chechik, and E. Fetaya. Multi-task learning as a bargaining game. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16428–16446, 2022.

W. Ogryczak and T. Śliwiński. On solving linear programs with the ordered weighted averaging objective. *European Journal of Operational Research*, 148(1):80–91, 2003.

W. Ogryczak, H. Luss, M. Pióro, D. Nace, and A. Tomaszewski. Fair optimization and networks: A survey. *Journal of Applied Mathematics*, 2014:612018, 2014. doi: 10.1155/2014/612018.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

C. Romero. *Handbook of Critical Issues in Goal Programming*. Pergamon Press, 1991.

O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems 31*, pages 525–536, 2018.

J. Špirková, G. Beliakov, H. Bustince, and J. Fernandez. Mixture functions and their monotonicity. *Information Sciences*, 481:520–549, 2019.

R. E. Steuer and E.-U. Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344, 1983. doi: 10.1007/BF02591870.

M. Tamiz, D. Jones, and C. Romero. Goal programming for decision making: An overview of the current state-of-the-art. *European Journal of Operational Research*, 111(3):569–581, 1998.

M. M. Van Hulle. A goal programming network for linear programming. *Biological Cybernetics*, 65(4):243–252, 1991. doi: 10.1007/BF00206222.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, 2017.

A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Application*, volume 177 of *Lecture Notes in Economics and Mathematical Systems*, pages 468–486. Springer, 1980.

T. Wilkin and G. Beliakov. Weakly monotonic averaging functions. *International Journal of Intelligent Systems*, 30(2):144–169, 2015.

R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision-making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988. doi: 10.1109/21.87068.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- BajpaiORFsupplementarycode.zip