ALIGNFOLIO RESEARCH

Contribution Allocation in Multi-Asset Portfolios:

A Comparison of Two Internally Consistent Methodologies

Alex Odas, PhD
Licensed Financial Advisor, British Columbia, Canada
Alignfolio
July 2026

Working paper. This document reports an internal methodology comparison and is made available for transparency. It is not peer-reviewed and does not constitute investment advice.

Conflict of Interest Disclosure
The author is the founder of Alignfolio, where Algorithm A (Section 3.1) is currently implemented in production software. This disclosure is provided so readers can evaluate the assumptions, methodology, and conclusions with appropriate context.

Abstract

Portfolio management guidance commonly recommends directing new cash contributions toward underweight asset classes ahead of selling. Rarely specified is how that contribution should be divided when more than one asset class is simultaneously underweight. This paper documents and compares two internally consistent methodologies for that decision: a proportional dollar-shortfall approach ("Algorithm A", currently implemented in Alignfolio) and a relative-drift equalization approach ("Algorithm B"). We prove algebraically, and confirm numerically, that target dollar values must be computed against the post-contribution portfolio total for either method to correctly restore a portfolio to target, and that both methodologies necessarily converge to an identical result once the contribution reaches what we term the exact-restoration regime: the point at which every asset class is at or below its post-contribution target before allocation. Short of that regime - the partial-funding case examined in this study - the two methodologies diverge substantially, and we characterize that divergence across 20,000 simulated portfolios (ten independent random seeds) and a five-dimensional sensitivity analysis. This paper does not establish that either objective function is universally preferable. It establishes that the choice between them is a consequential, previously under-specified implementation decision that should be made and documented explicitly rather than left implicit.

Why This Paper Exists

Portfolio literature widely agrees that new contributions should be directed toward underweight asset classes ahead of selling existing holdings. To the author's knowledge, widely available practitioner guidance and commercial software documentation rarely specify the precise rule for dividing a contribution among multiple simultaneously underweight asset classes. Different advisors, spreadsheets, and software platforms may answer this question differently, using methodologies that are each internally reasonable but that were never designed to agree with one another.

The absence of a standardized contribution-allocation rule becomes apparent whenever different practitioners or software implementations produce different recommendations for identical portfolios. This paper formalizes two such methodologies, verifies their properties directly against production code, and characterizes where and how much they disagree.

The purpose of this paper is not to recommend a single methodology for industry adoption, but to demonstrate that contribution-allocation methodology is itself an explicit implementation decision whose consequences can be quantified.

1. Objectives

This paper compares two specific, internally consistent contribution-allocation methodologies. It addresses:

- Given a fixed new cash contribution and more than one underweight asset class, what does each methodology recommend?
- Do the two methodologies converge, diverge, or both, depending on conditions?
- Where they diverge, how large is the practical difference, and how sensitive is that difference to portfolio structure and simulation assumptions?

This paper does not attempt to establish that one objective function is universally correct, and does not survey the full space of possible allocation algorithms. It also does not address a related but distinct question - when a portfolio has drifted far enough to warrant attention - which is a monitoring decision addressed elsewhere.

2. Problem Definition

Consider a portfolio with n asset classes, current dollar values $v_1 \ldots v_n$, and target allocation percentages $t_1 \ldots t_n$ (summing to 100%). An investor contributes new cash $C \geq 0$. The question addressed here is: how much of C should be directed to each asset class?

A first design decision, often left implicit, is the basis against which each asset class's target dollar value is computed:

- Current-total basis: $\text{target}_i = t_i \times$ (current portfolio total)
- Post-contribution basis: $\text{target}_i = t_i \times$ (current portfolio total + C)

Section 5.1 demonstrates, as a proven mathematical result rather than a simulation finding, that the current-total basis fails a specific, testable correctness case: it does not restore a portfolio to its exact target allocation even when the contribution is precisely sufficient to do so. For this reason, both algorithms compared in this paper use the post-contribution basis exclusively.

The methodologies compared here should be understood as two explicit answers to this operational question, rather than as competing interpretations of an established standard.

3. Methodology

3.1 Algorithm A: Proportional Dollar-Shortfall Allocation

Algorithm A is a proportional dollar-shortfall methodology currently implemented in Alignfolio. For each asset class, it computes the post-contribution target value and resulting dollar shortfall:

$$\text{target}_i = (\text{current\_total} + C) \times t_i$$

$$shortfall_i = \max(0, target_i - v_i)$$

The contribution is distributed in proportion to shortfall:

$$amount_i = C \times (shortfall_i \div \Sigma\, shortfall_j)$$

This is a proportional heuristic and is not presented as the exact optimizer of the squared-dollar-shortfall measure or of any general dollar-based objective. It is evaluated in this paper as a specific, reproducible allocation rule currently implemented in Alignfolio, not as a solution to a stated optimization problem. The true minimizer of total squared remaining dollar shortfall is a water-filling solution that equalizes remaining shortfalls additively rather than preserving their proportions; a numerical example illustrating the difference is given in Appendix B.

3.2 Algorithm B: Relative-Drift Equalization

Algorithm B is an alternative methodology that allocates contributions by equalizing remaining relative drift. For each asset class, it computes relative drift against the post-contribution target:

$$relDrift_i = (v_i \div target_i) - 1$$

The contribution is allocated to bring every underweight asset class's relative drift up to a common level ρ, solved so total allocated dollars equal exactly C:

$$amount_i = \max(0, \rho - relDrift_i) \times target_i, \quad \text{where } \Sigma\, amount_i = C$$

ρ is found by binary search. Algorithm B exactly equalizes remaining relative drift among funded categories, subject to non-negative contribution amounts - categories whose initial relative drift already exceeds the solved common level ρ receive zero and are excluded from the equalization, rather than a proportional approximation being applied to any category.

The two algorithms are not symmetric in their mathematical construction. Algorithm A applies a proportional dollar-shortfall heuristic, while Algorithm B exactly equalizes remaining relative drift among funded categories. This asymmetry reflects the two methodologies as actually specified and implemented - Algorithm A as deployed in production, Algorithm B as proposed - rather than a deliberate attempt to match algorithmic sophistication. A systematic comparison against the exact minimizer of squared dollar shortfall was not performed and is noted as a direction for future work in Section 7.

3.3 Evaluation Measures

Two measures are proven mathematically (Section 5.1) and require no simulation:

- Exact restoration: whether a sufficient contribution brings every asset class exactly to target.
- Convergence: whether the two algorithms necessarily agree once funding is sufficient.

Three further measures require simulation, because they depend on the distribution of portfolios and contributions tested, not on either algorithm's formal properties alone:

- Squared-dollar-shortfall evaluation measure: Σ (remaining dollar shortfall$_i$)^2 across asset classes, lower is better under this measure.
- Squared-relative-shortfall evaluation measure: Σ (remaining relative shortfall$_i$)^2 across asset classes, lower is better under this measure.
- Contribution Concentration Rate: the proportion of simulated allocations in which a single asset class with a small target (below a stated threshold of the post-contribution total) receives more than a stated share (default: 50%) of the total contribution.

The Contribution Concentration Rate is an operational, not a correctness, measure. It does not indicate that either algorithm produced a mathematically invalid result - both remain internally consistent by construction - only that the resulting recommendation is concentrated in a single, small position. Whether such concentration is desirable depends on the intended application: it may be undesirable in a general-purpose consumer tool and entirely appropriate for an investor deliberately catching up an underfunded position.

3.4 Simulator Documentation

For reproducibility, the simulation is documented in full.

Random number generation: simulations used the seeded mulberry32 pseudorandom number generator to permit exact replication across runs. Seeds 1 through 10 were used for the primary comparison (Section 5.2) and seed 1 for each sensitivity condition (Section 5.3). The algorithm implementations, simulation code, seeds, and the raw seed-level results underlying Table 3 are provided as a reproducibility package accompanying this paper.

Target-weight generation proceeds in four steps. (1) With a stated probability (default 40%), a satellite category is included with target s drawn uniformly from 1% to 5%; otherwise s = 0. (2) The number of core categories is drawn uniformly as an integer from 3 to 6. (3) Core weights are drawn via the exact-Dirichlet construction described below and normalized to sum to 1. (4) Every core weight is multiplied by (100 - s), and the satellite (if present) is appended with weight s, so the full vector - core weights plus satellite - sums to exactly 100%.

Core weights in step 3 are generated by drawing, for each category, the sum of k independent unit-exponential values (k held fixed within a given portfolio, drawn per-portfolio as an integer between 2 and 4), then normalizing across categories by their sum. Since a sum of k i.i.d. unit-exponential draws is Gamma(k, 1) distributed, and normalized i.i.d. Gamma(k, 1) draws are exactly Dirichlet(k, ..., k) distributed, this construction produces an exact symmetric Dirichlet draw for each portfolio - not merely an approximation - though the concentration parameter is restricted to integer values rather than varying continuously. This was confirmed

empirically: 200,000 draws at k = 3 matched the theoretical Dirichlet(3,3,3,3) mean and variance to within simulation error.

Portfolio size: log-uniformly distributed between $50,000 and $1,000,000.

Current-allocation drift: for each portfolio, a noise scale σ was drawn uniformly from 1.5 to 4.5 percentage points and shared across every category in that portfolio. Each category then received an independent perturbation drawn uniformly from [−σ, σ] and added to its target percentage. Resulting values were floored at 0.01 (not zero, to avoid a degenerate zero-value category) and the full vector was renormalized to sum to exactly 100%. No portfolios were rejected at this step; every generated portfolio was retained regardless of how much its current allocation had drifted from target.

Contribution size: drawn as a random fraction (default 20-80%) of the total dollar shortfall computed against the current (pre-contribution) total - a deliberately simple sizing rule, not intended to represent the true distribution of contribution sizes made by real investors. This sizing rule does not by construction guarantee that the resulting contribution stays below the true, post-contribution-basis exact-restoration threshold C* defined in Section 5.3; the two quantities are computed differently. This was therefore checked directly rather than assumed: across all 20,000 simulated portfolios underlying Table 3, the drawn contribution was below C* in every case (0 of 20,000 fell at or beyond the exact-restoration regime). No portfolios were rejected or redrawn; every generated case was retained and happened to fall within partial funding under this sizing rule.

Seed handling: the primary result (Section 5.2) reports the mean and standard deviation across ten independent seeds (1 through 10), 2,000 portfolios each, 20,000 simulated portfolios in total. The sensitivity analysis (Section 5.3) uses a single seed per condition to isolate the effect of the varied parameter; absolute values there should be read as indicative rather than as independently replicated estimates.

4. Implementation and Verification

Both algorithms were implemented directly in TypeScript and executed programmatically rather than computed by hand. All figures in this paper are direct outputs of that code.

During implementation, a materially incorrect production implementation of Algorithm A was identified: an earlier version computed target values against the current, pre-contribution total. This produced a demonstrable defect, shown in Section 5.1, and has since been replaced with the post-contribution-basis implementation described in Section 3.1 across all production surfaces, covered by an automated regression test suite that includes the exact-restoration case below.

5. Results

5.1 Proven Results

The following results follow from the algorithm definitions in Section 3 and are demonstrated by direct construction, not by simulation. They do not depend on any distributional assumption.

Worked example. Portfolio: US Equity $72,000 (60% target), International $20,000 (30% target), Bonds $8,000 (10% target); current total $100,000. Contribution: $10,000.

Asset Class	Algorithm A	Final Value	Final %	Final Drift
US Equity	$0	$72,000	65.45%	+5.45 pts
International	$8,125	$28,125	25.57%	−4.43 pts
Bonds	$1,875	$9,875	8.98%	−1.02 pts

Table 1. Algorithm A output for the worked example, verified against the production implementation.

Exact-restoration test. The same portfolio requires exactly $20,000 to reach 60/30/10 precisely.

Basis Tested	US Equity	International	Bonds	Result
Post-contribution total (Algorithm A, current)	$0	$16,000	$4,000	60.00% / 30.00% / 10.00%
Current total (superseded implementation)	$4,800	$12,400	$2,800	64.00% / 27.00% / 9.00%

Table 2. The current-total basis fails exact restoration despite a mathematically sufficient contribution; this motivated its replacement. This is a proven correctness result, not a simulation finding.

Algorithm B, evaluated independently on the same portfolio and contribution, produces the identical $16,000 / $4,000 split. This agreement is not a coincidence - it is proven next.

Convergence proof. We define the exact-restoration regime as the condition where every asset class is at or below its post-contribution target before allocation - equivalently, $v_i \le t_i(P + C)$ for every i, where P is the current portfolio total. This condition holds if and only if C is at least as large as the threshold:

$$C^* = \max(0, \max_i[v_i \div t_i - P])$$

The exact-restoration regime begins at C = C*. Within this regime, total shortfall equals exactly:

$$\Sigma\ \text{shortfall}_i = (\text{current\_total} + C) \times \Sigma\ t_i - \Sigma\ v_i = (\text{current\_total} + C) - \text{current\_total} = C$$

since target percentages sum to 100% and current values sum to the current total by definition. The equality Σ shortfall$_i$ = C is a property of the target structure rather than of either algorithm. Because Algorithms A and B both allocate exactly

the defined shortfalls in this regime, both land on the unique target portfolio; this does not imply that every conceivable allocation rule would do the same, only that the two algorithms studied here do.

This was confirmed by direct construction: across 500 randomly generated portfolios, given a contribution of 20× total portfolio value, Algorithms A and B produced identical output to the cent in 500 of 500 cases.

5.2 Simulation-Dependent Findings: Primary Comparison

The following results depend on the simulator described in Section 3.4 and should be interpreted as conditional on the tested portfolio distributions, not as population estimates for real investor portfolios. Ten independent random seeds were run (2,000 portfolios each, 20,000 total), restricted to the partial-funding regime.

Measure	Algorithm A (mean)	Algorithm B (mean)	Paired difference A-B, 95% t-CI	A lower / B lower
Squared dollar shortfall	61,714,450	62,986,524	−1,272,075 [−1,563,404, −980,745]	10 / 0
Squared relative shortfall	0.0534	0.0216	+0.0318 [0.0303, 0.0333]	0 / 10
Contribution Concentration Rate	5.20%	17.19%	−12.01 pts [−12.69, −11.33]	10 / 0

Table 3. Paired comparison across ten independent seeds (20,000 simulated portfolios total). Both algorithms were evaluated on identical simulated portfolios within each seed, making the paired design - rather than separate per-algorithm confidence intervals - the statistically appropriate comparison.

For each seed s, the paired difference $d_s = \text{Metric}_{A,s} - \text{Metric}^{B},_{s}$ was computed, and the 95% confidence interval reported is a two-sided Student's t interval on the ten paired differences (t = 2.262, 9 degrees of freedom).

Algorithm A produced a lower squared-dollar-shortfall score in all ten independent runs; the paired-difference confidence interval excludes zero, so this difference is statistically distinguishable from zero under the tested conditions, even though the average margin was modest relative to the scale of the underlying values. Algorithm B produced a lower squared-relative-shortfall score in all ten runs, with a substantially larger paired difference. Algorithm A also produced a lower Contribution Concentration Rate in all ten runs, with a paired difference of roughly 12 percentage points. All three paired-difference confidence intervals exclude zero.

Divergence between the two algorithms' recommended dollar amounts was computed as:

$$\text{divergence} = 100 \times \max_i |a_i^A - a_i^B| \div C$$

the largest single-category dollar difference between the two recommendations, as a percentage of the total contribution C. This is a maximum, not an average, across

categories within each portfolio; each portfolio contributes exactly one divergence value to the pooled distribution below, regardless of how many asset classes it contains. Computed across all 20,000 simulated portfolios (ten seeds pooled): mean 23.8%, median 22.4%, interquartile range [11.7%, 34.1%], 95th percentile 52.6%, maximum 77.2%. Mean divergence was higher for portfolios containing a small satellite position (27.7%, n = 8,138) than for those without one (21.1%, n = 11,862), and was stable across the ten individual seed means (standard deviation 0.38 percentage points).

As a complementary measure, we also computed the reassignment fraction - the share of the total contribution that would need to move between categories to convert one algorithm's recommendation into the other's:

$$\text{reassignment} = 100 \times (1 \div 2C) \times \Sigma_i |a_i^A - a_i^B|$$

Across the same pooled sample: mean 24.3%, median 23.3%, interquartile range [12.1%, 34.9%], 95th percentile 52.8%, maximum 77.2%. The two measures are close in this sample because most simulated portfolios have only two categories receiving a nonzero allocation in the partial-funding regime, a case in which the two statistics coincide exactly.

5.3 Sensitivity Analysis

The Contribution Concentration Rate finding (Algorithm A lower than Algorithm B) was tested across five dimensions, each varied independently from the primary design while holding other parameters fixed. Single-seed runs of 1,000 portfolios were used per condition.

Dimension	Condition	Algorithm A	Algorithm B	Ratio (B/A)
Contribution size	5-30% of shortfall funded	6.4%	21.5%	3.4×
Contribution size	30-60% of shortfall funded	5.1%	16.3%	3.2×
Contribution size	60-95% of shortfall funded	2.8%	11.1%	4.0×
Drift/noise level	0.5-1.5 pts	5.6%	16.8%	3.0×
Drift/noise level	1.5-4.5 pts (primary)	4.7%	16.0%	3.4×
Drift/noise level	4.5-9 pts	2.9%	12.4%	4.3×
Concentration share threshold	25% share	8.6%	17.0%	2.0×
Concentration share threshold	50% share (primary)	4.7%	16.0%	3.4×
Concentration share threshold	75% share	1.4%	9.8%	7.0×
Small-category threshold	3% of total	1.7%	7.2%	4.2×
Small-category threshold	5% of total (primary)	4.7%	16.0%	3.4×
Small-category threshold	10% of total	10.4%	27.7%	2.7×
Satellite probability	0% (no small positions)	1.4%	5.1%	3.6×

Satellite probability	40% (primary)	4.7%	16.0%	3.4×
Satellite probability	80%	6.3%	25.8%	4.1×

Table 4. Sensitivity of the Contribution Concentration Rate finding across five independently varied dimensions. The direction of the effect (Algorithm A lower than Algorithm B) holds in all fifteen tested conditions; the magnitude of the ratio varies considerably, most notably with the concentration share threshold itself.

Two results merit specific note. First, the finding holds even with zero probability of a small satellite position (1.4% vs. 5.1%), indicating the effect is not confined to portfolios containing an explicit small position and arises more generally whenever target percentages vary across underweight categories. Second, the ratio is sensitive to the concentration share threshold itself (2.0× at a 25% threshold versus 7.0× at a 75% threshold), which should temper any single headline ratio reported from this analysis; the direction of the effect is more consistent than any specific magnitude.

6. Discussion

The central question is not which algorithm wins universally, but which allocation principle each method expresses and what practical consequences follow. Algorithm A distributes contributions proportionally to post-contribution dollar shortfalls. Algorithm B allocates contributions to equalize remaining relative drift among funded categories. In the simulations, Algorithm A produced modestly lower squared-dollar-shortfall scores, while Algorithm B produced substantially lower squared-relative-shortfall scores. Algorithm A also produced fewer contributions concentrated primarily in a small-target category, with the direction of that result consistent across all fifteen single-seed sensitivity conditions tested. These findings describe different characteristics of the methods; they do not establish a universally preferable methodology.

These are different allocation principles serving different practical purposes. A practitioner or platform that places greater weight on reducing absolute dollar shortfalls may reasonably prefer Algorithm A. A practitioner primarily concerned with proportional discipline across differently-sized positions - particularly for portfolios with several meaningfully different target weights - may reasonably prefer Algorithm B, and should weigh that preference against the higher rate of concentrated allocations documented in Section 5.3.

7. Limitations

- All simulation-dependent findings (Sections 5.2-5.3) are conditional on the synthetic portfolio distributions described in Section 3.4, not on real client data. Specific figures should be read as evidence of a real, non-trivial, directionally consistent effect, not as estimates of real-world frequency. The sensitivity analysis (Section 5.3) used a single seed per condition; only the primary comparison (Section 5.2) was replicated across multiple independent seeds.

- The target-weight generator draws an exact symmetric Dirichlet distribution per portfolio, but restricts the concentration parameter to integer values (2, 3, or 4) rather than allowing it to vary continuously; a continuous-concentration sampler was not implemented.
- Only two allocation methodologies were compared, as actually specified and implemented (Algorithm A) or proposed (Algorithm B). A systematic comparison against the exact mathematical minimizer of squared dollar shortfall (see Appendix B), or against other candidate methodologies (e.g. hybrid or threshold-based rules), was not performed and is a natural direction for future work.
- The Contribution Concentration Rate is one operational measure of recommendation concentration and should not be read as a general measure of recommendation quality, suitability, or mathematical correctness. Alternative operational criteria may produce different comparative assessments.
- This paper addresses contribution allocation only, not withdrawal allocation, tax-lot selection, or the separate decision of when a portfolio's drift warrants active rebalancing.
- The literature review in Section 2 is intentionally narrow in scope; a comprehensive survey of contribution-based rebalancing, tolerance-band design, and related resource-allocation literature was not undertaken.

8. Conclusion

The two methods examined here are mathematically coherent but encode different allocation principles. They agree exactly in the exact-restoration regime and can differ substantially under partial funding. Under the tested conditions, Algorithm A produced modestly lower squared-dollar-shortfall values and fewer highly concentrated contributions to small-target categories; Algorithm B produced substantially lower squared-relative-shortfall values. These results do not establish universal superiority for either method. They demonstrate that contribution-allocation methodology is a consequential design decision that should be explicitly documented rather than left implicit in software or spreadsheet logic.

Acknowledgements

The relative-drift equalization approach evaluated as Algorithm B in this paper was proposed by a third party in the course of public discussion of Algorithm A's methodology. The author thanks that participant for a well-constructed and independently useful proposal.

References

Daryanani, G. (2008). Opportunistic Rebalancing: A New Paradigm for Wealth Managers. Journal of Financial Planning. Addresses rebalancing frequency and

tolerance-band design; does not address division of a specific contribution across multiple underweight asset classes.

Vanguard. Rebalancing your portfolio: How to rebalance. Describes the general practice of directing new contributions toward underweight asset classes ahead of selling; does not specify a division rule for multiple simultaneously underweight categories.

Both references are cited in support of the broader, well-established principle that new cash should be directed toward underweight assets ahead of selling. Neither describes or endorses either specific algorithm evaluated in this paper. A fuller review of tolerance-band rebalancing, contribution-based rebalancing, and general resource-allocation literature was outside the scope of this working paper; see Section 7.

Appendix A: Proof of the No-Surplus Property

Claim: for Algorithm A, within the exact-restoration regime (every asset class at or below its post-contribution target), the proportional-split formula never requires a separate rule for handling contribution in excess of total shortfall, because such an excess cannot occur.

Let P be the set of asset classes with positive shortfall at contribution C, and let:

$$W = \Sigma_{i \in P}\, t_i \quad \text{(total target weight of the underweight set)}$$

$$V = \Sigma_{i \in P}\, v_i \quad \text{(total current value of the underweight set)}$$

Then total shortfall at contribution C is:

$$\Sigma\ \text{shortfall}(C) = W \times (\text{current\_total} + C) - V$$

When P includes every asset class ($W = 1$, $V = \text{current\_total}$), this simplifies to $\Sigma\ \text{shortfall}(C) = C$ exactly, for any C - not merely at one boundary value. Numerically verified across contributions of $20,000 through $1,000,000 on the Section 5.1 portfolio (Table 5) and across 500 randomly generated portfolios at 20× portfolio value (Section 5.1).

Contribution	Total Future Shortfall	Shortfall − Contribution
$20,000	$20,000.00	$0.00
$60,000	$60,000.00	$0.00
$100,000	$100,000.00	$0.00
$1,000,000	$1,000,000.00	$0.00

Table 5. Total future shortfall exactly equals the contribution amount for any sufficiently large contribution, confirming the algebraic result above.

Appendix B: Proportional Allocation Does Not Minimize Squared Dollar Shortfall

This appendix demonstrates, by direct counterexample, that Algorithm A's proportional allocation rule does not generally coincide with the minimizer of total squared remaining dollar shortfall - including in cases with only two asset classes and no equal-target-percentage structure.

Consider two asset classes with dollar shortfalls of $80 and $20, and a contribution of $50.

Allocation Rule	Category 1	Category 2	Remaining Shortfall	Sum of Squared Residuals
Proportional (Algorithm A)	$40.00	$10.00	$40.00 / $10.00	1,700
True squared-shortfall minimizer	$50.00	$0.00	$30.00 / $20.00	1,300

Table 6. Proportional allocation produces a strictly higher sum of squared remaining dollar shortfall than the true constrained minimizer, confirmed by direct computation.

The true minimizer follows from the Lagrangian condition for minimizing $\Sigma(\text{shortfall}_i - x_i)^2$ subject to $\Sigma x_i = C$ and $x_i \geq 0$: absent the non-negativity constraint, the optimum reduces every shortfall by an identical dollar amount (an additive water-filling solution), not a proportionally equal fraction. When the unconstrained solution would require a negative allocation - as it does here, where the unconstrained optimum calls for $55 to Category 1 and −$5 to Category 2 - the constrained optimum instead sets the infeasible allocation to zero and gives the remainder to the other category, as shown in Table 6.