CLARA: Factual–Counterfactual Duality and Reoptimization for Linear Programming

Yoonsik Jung[a,*]

[a] *Department of Industrial and Management Engineering, Korea University, Seoul, Republic of Korea*

Abstract

Linear programming (LP) solvers return optimal solutions, but practitioners need more than a single optimum: they need to know how stable that solution is, what minimal change would alter it, and whether a perturbed instance requires re-solving. We address these questions through two central contributions. First, we formalize the *factual–counterfactual duality* in LP: the basis robustness d_0, computable in $O(mn)$ time from the simplex basis inverse B^{-1} alone, is a lower bound on the cost of any basis-changing counterfactual explanation, connecting classical sensitivity analysis with the emerging field of counterfactual explanations for optimization. Second, we present a *reoptimization pipeline* that integrates change detection, Oguz-bound impact analysis, automatic warm-start method selection (primal simplex, dual simplex, or parametric LP), and structured diff reports within a single decision framework. Built on the same geometric foundation, CLARA also provides a publicly available implementation of simultaneous sensitivity regions via the Chebyshev center of the basis-preserving polyhedron, together with objective-change attribution via first-order and Shapley decomposition that annotates diff reports. Computational experiments on 127 instances and approximately 3,500 perturbation pairs confirm solver correctness (optimality gap below 2e-12), a 28-fold mean pivot reduction under warm-start (4.6-fold wall-clock speedup), and that one-at-a-time (OAT) sensitivity anal-

*Corresponding author

Email address: ys_jung@korea.ac.kr (Yoonsik Jung)

ysis overestimates the safe perturbation region by approximately tenfold on average. CLARA is available as open-source software under the MIT license.

Keywords: continuous optimization, linear programming, sensitivity analysis, explainable optimization, counterfactual explanation, reoptimization

1. Introduction

Modern linear programming (LP) solvers such as CPLEX, Gurobi, and HiGHS solve large-scale instances rapidly (Bertsimas and Tsitsiklis, 1997), yet practitioners typically gain little insight into the stability of the returned solution or what minimal change would alter it.

Commercial solvers report shadow prices and reduced costs—the classical outputs of sensitivity analysis (Dantzig, 1963)—but these are one-at-a-time (OAT) quantities: each measures the effect of changing a single parameter while holding all others fixed. When multiple parameters are uncertain simultaneously, OAT ranges can be misleading (Koltai and Terlaky, 2000). The tolerance approach (Wendell, 1985) and its extensions provide the theory for simultaneous sensitivity analysis, but this theory remains absent from commercial LP packages (Jansen et al., 1997; Koltai and Terlaky, 2000). Recent work on counterfactual explanations (CEs) for optimization (Korikov et al., 2021; Kurtz et al., 2025) addresses the complementary question of what minimal change would lead to a different outcome, but the mathematical relationship between the sensitivity region (factual) and the CE distance (counterfactual) has not been formalized. Existing XAI approaches for optimization use ML models or inverse formulations as intermediaries (Goerigk and Hartisch, 2023; Busch et al., 2025; Korikov et al., 2021; Kurtz et al., 2025), introducing approximation where exact analysis is available. None directly leverages the simplex basis inverse B^{-1}, which already contains shadow prices, reduced costs, and the information needed for simultaneous sensitivity and attribution.

A separate practical question concerns reoptimization. In model predictive control, column generation, and branch-and-bound, LP instances are

solved repeatedly with slight parameter modifications (Rawlings et al., 2017; Lübbecke and Desrosiers, 2005; Bolusani et al., 2024). Practitioners either always re-solve (potentially wasting effort on negligible changes) or skip re-solving based on ad-hoc thresholds. The only available bound on the cost of neglecting reoptimization is due to Oguz (2000), but no tool integrates this bound into an automatic decision framework.

In this paper, we present CLARA (Classical LP Analysis for Reoptimization and Attribution), a framework that unifies factual and counterfactual perspectives on LP explainability through two central contributions. First, we formalize a *factual–counterfactual duality*: a formal bound showing that the basis robustness d_0 (computable in $O(mn)$ from B^{-1}) lower-bounds the cost of any basis-changing counterfactual explanation, connecting classical sensitivity analysis with counterfactual explanations for optimization (Section 4.2). Second, we present a *reoptimization pipeline* that integrates change detection, Oguz-bound impact analysis, automatic warm-start method selection (primal simplex, dual simplex, or parametric LP), and structured diff reports within a single decision framework (Section 5). Built on the same geometric foundation, we also provide a publicly available implementation of *simultaneous sensitivity regions* via the Chebyshev center of the basis-preserving polyhedron (Section 4.1), and an *objective-change attribution* module based on first-order and Shapley decomposition that annotates diff reports (Section 5.4). A computational study on 127 LP instances and ∼3,500 perturbation pairs validates solver correctness (optimality gap $< 2\times10^{-12}$), warm-start pivot reductions of 28× when applicable, and an order-of-magnitude overestimation (∼10×) by OAT analysis of the safe perturbation region.

2. Related Work

2.1. Sensitivity analysis in linear programming

Sensitivity analysis examines how the optimal solution changes when problem parameters change. The standard approach, implemented in every commercial LP solver, is OAT analysis: for each right-hand-side coefficient

b_i or objective coefficient c_j, the solver reports the range within which that single parameter can vary—holding all others fixed—without changing the optimal basis. The resulting shadow prices and reduced costs, originating from Dantzig (1963), remain the most widely used sensitivity information in practice.

OAT analysis, however, can be misleading when multiple parameters change simultaneously: individual ranges may each appear safe, yet combinations within those ranges can cause a basis change. The *tolerance approach* of Wendell (1985) addresses this by computing the maximum percentage τ^* within which all selected coefficients can simultaneously vary while preserving optimality; Ravi and Wendell (1989) extend the approach to entries of the constraint matrix A, and Ward and Wendell (1990) survey OAT, parametric, and tolerance methods along informativeness, ease of use, and tractability.

Filippi (2005) then developed an iterative geometric algorithm yielding tolerance regions strictly larger than Wendell's in degenerate cases, and Borgonovo et al. (2018) merged the approach with Wagner's global sensitivity analysis for joint variations in b and c.

Despite extensive theoretical development, the practical adoption of simultaneous sensitivity analysis has been limited. Koltai and Terlaky (2000) highlight the gap between the managerial and mathematical interpretations of sensitivity results: practitioners often misinterpret OAT ranges as valid for simultaneous changes, leading to flawed decisions. Koltai and Tatay (2011) propose a practical approach to sensitivity analysis under degeneracy—where shadow prices are not unique—and reinforce the long-standing observation of Jansen et al. (1997) that the simultaneous sensitivity theory, though well-developed, remains absent from commercial LP packages. This remains true today: CPLEX, Gurobi, and HiGHS all provide OAT sensitivity reports, but none offers simultaneous sensitivity regions, tolerance percentages, or objective change attribution.

Parametric programming (Gal, 1979) traces the optimal value as a piecewise-linear function of a scalar parameter and identifies all basis-change breakpoints; despite its theoretical completeness, it remains absent from commercial solver APIs.

2.2. Reoptimization

Reoptimization—solving a modified optimization problem by leveraging information from a previously solved instance—is a fundamental operation in both LP theory and practice. For the simplex method, warm-starting from an existing basis after right-hand-side or objective changes is a textbook technique (Bertsimas and Tsitsiklis, 1997): if b changes, the old basis may become primal infeasible but remains dual feasible, so dual simplex recovers optimality in few pivots; if c changes, the old basis remains primal feasible, and primal simplex restores dual feasibility.

For interior-point methods (IPMs), warm-starting is considerably harder because the previous iterate may lie far from the central path of the perturbed problem. Worst-case iteration bounds were given by Yıldırım and Wright (2002), and practical recovery strategies by Gondzio and Grothey (2003, 2008) and Colombo et al. (2011). As MOSEK's documentation summarizes, restarting capabilities for IPMs remain less reliable than for the simplex method.

The theoretical foundation for quantifying the cost of *not* reoptimizing was laid by Oguz (2000), who showed that the opportunity cost of neglecting reoptimization is bounded by $2\delta/(1+\delta)$, where δ measures the relative data change. This bound applies to LP but has not been extended to MIP. Albici et al. (2010) provide a taxonomy for classifying parameter changes (Type R, C, V, X, RC, etc.) and matching each type to the appropriate warm-start method.

For mixed-integer programs, Gamrath et al. (2015) pioneer systematic reoptimization in SCIP by reusing the branch-and-bound frontier. The MIP Workshop 2023 dedicated its computational competition to reoptimization (Bolusani et al., 2024); the winning approach of Patel (2024) combined primal solution reuse, pseudocost transfer, and online parameter tuning. On the constraint-matrix side, Miftari et al. (2024) study sensitivity for linear perturbations $A + \lambda D$, going beyond the classical b/c setting addressed here.

Reoptimization arises routinely in practice—model predictive control (Rawlings et al., 2017), column generation (Lübbecke and Desrosiers, 2005; Barnhart et al., 1998), rolling horizon planning, and every node LP in branch-

and-bound. In each setting, practitioners either always reoptimize or skip it by ad-hoc thresholds, without theoretical guarantees on the resulting sub-optimality.

A common thread in this literature is that all work focuses on *how* to reoptimize efficiently (faster warm-start, better branching reuse), while the question of *whether* to reoptimize and *how much is lost* by not reoptimizing remains largely unaddressed. The only theoretical tool is the Oguz bound, which is LP-specific and provides no integrated decision framework. CLARA addresses this gap by combining change detection, impact analysis (via the Oguz bound and sensitivity ranges), automatic method selection, and warm-start execution in a single pipeline.

2.3. Explainable optimization

XAI methods such as SHAP (Lundberg and Lee, 2017) and LIME (Ribeiro et al., 2016) provide post-hoc explanations for ML models, but explainability for optimization solvers has received far less attention (De Bock et al., 2024; Busch et al., 2025). Goerigk and Hartisch (2023) propose a decision tree that maps each future problem instance to one of a precomputed set of solutions, trading off solution quality for interpretability and explaining which *class* of solution an instance belongs to rather than a specific optimum. Busch et al. (2025) encode LPs in neural networks and apply Integrated Gradients/Saliency to the encodings, while Aigner et al. (2024, 2025) add data-driven explanation terms and LIME-style local surrogates constrained by coherence with the problem structure. These approaches leverage the ML-XAI toolkit but introduce approximation: the explanation pertains to the surrogate, not to the solver's actual reasoning.

A separate line of work adapts counterfactual explanations—the question of what minimal change in the input would lead to a desired output—from ML to optimization. Korikov et al. (2021) introduce CEs in the GDPR context, showing that for weighted-sum-of-binary objectives, inverse optimization can be avoided; Korikov and Beck (2023) extend this to general linear discrete problems via inverse constraint programming. Kurtz et al. (2025) translate CEs to linear optimization, proposing strong, weak, and relative

CE types and showing that relative CEs exploit hidden convex structure for efficient computation on NETLIB instances. Engelhardt et al. (2025) extend the framework to integer programs, proving CE computation is Σ_2^p-complete even for binary programs with one mutable constraint. CEs are mathematically rigorous and actionable, but they address the counterfactual half of the explainability question; the mathematical connection between the factual sensitivity region (the set of perturbations preserving the current basis) and the CE distance (the minimal perturbation changing the basis) has not been formalized, despite both being rooted in the same basis-preserving polyhedron.

None of these approaches directly leverages the simplex basis inverse B^{-1}, which already contains shadow prices, reduced costs, and sensitivity ranges as byproducts of the simplex method, with no additional approximation. Recent implementations such as linrax (Gould et al., 2025), a JAX-compatible simplex solver, demonstrate that retaining tableau and basis information is a viable design choice, but linrax provides no sensitivity analysis, attribution, or explanation modules atop these artifacts. We adopt the solver-intrinsic approach and establish the formal connection between factual and counterfactual perspectives within a single geometric framework.

Table 1 summarizes how CLARA relates to the approaches reviewed above. CLARA combines exactness (no ML approximation), solver-intrinsic extraction (directly from B^{-1}), and a formal duality linking the factual sensitivity region to counterfactual perturbations.

3. The CLARA Framework

The central design principle of CLARA is *basis inverse preservation*: after solving an LP with the revised simplex, the basis inverse B^{-1} is retained rather than discarded. From B^{-1} alone one can compute shadow prices ($y = c_B^\top B^{-1}$), basic variable values ($x_B = B^{-1}b$), sensitivity ranges, and the polyhedron of simultaneous perturbations that preserve the basis—so retaining B^{-1} enables all downstream explanation and reoptimization modules without additional optimization.

Table 1: Comparison of explainability approaches for optimization. "What-if?" analyzes parameter changes; "Re-solve?" decides whether reoptimization is needed.

Approach	Method	Exact	Solver-intrinsic	What-if?	Re-solve?	Open-source
Commercial solvers	OAT sensitivity	Yes	Partial	Partial	No	No
Goerigk and Hartisch (2023)	Decision tree	Approx	No	No	No	No
Busch et al. (2025)	Neural encoding	Approx	No	No	No	Yes
Aigner et al. (2025)	LIME surrogate	Approx	No	Partial	No	Yes
Korikov et al. (2021)	Inverse opt. (CE)	Yes	No	Yes	No	No
Kurtz et al. (2025)	CE for LP	Yes	No	Yes	No	Yes
linrax (Gould et al., 2025)	JAX simplex	Yes	Yes	No	No	Yes
CLARA (ours)	B^{-1} **intrinsic**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

Figure 1 illustrates the framework. The solver engine produces a *SolveState* encapsulating the optimal solution x^*, value z^*, basis inverse B^{-1}, basis indices, and per-parameter sensitivity ranges. Three downstream modules consume this state independently: the *Region Analyzer* computes the simultaneous sensitivity region and the basis robustness d_0 underlying the duality bound (Section 4.1); the *Reoptimization Pipeline* handles change detection, impact analysis, warm-start method selection, and attribution-annotated diff reports (Section 5); and the *SolveState* itself reports per-constraint shadow prices, reduced costs, and the condition number $\kappa(B)$ as diagnostics. All modules share one SolveState, guaranteeing mutually consistent results.

CLARA ships two solver backends. The internal revised simplex maintains B^{-1} explicitly at every pivot, prioritizing transparency over speed. The HiGHS backend (Huangfu and Hall, 2018) is a production-grade reference solver; after HiGHS solves the LP, CLARA reconstructs B^{-1} from the reported basis, computes $\kappa(B)$, and flags numerically unstable instances. Section 6 reports optimality gaps below 2×10^{-12} across all 127 instances. The framework is not tied to either backend: any simplex-based solver exposing the optimal basis can serve as a backend.

Scope.. The Region Analyzer and Reoptimization Pipeline are fully implemented for standard-form LPs with b and c perturbations. The Region Analyzer computes the Chebyshev center for Δb; the joint $(\Delta b, \Delta c)$ polyhedron is formulated in Section 4.1 and requires only dual feasibility constraints

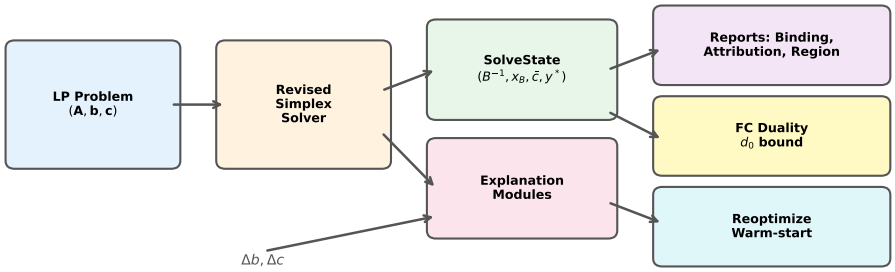

Figure 1: The CLARA framework. The revised simplex solver produces a SolveState (basis inverse B^{-1}, basic values x_B, reduced costs $\bar{c}$, dual variables y^*); downstream modules consume it for the sensitivity region and factual–counterfactual bound d_0, and for warm-start reoptimization with attribution-annotated diff reports. Perturbation inputs $(\Delta b, \Delta c)$ feed the change-analysis modules.

added to the same Chebyshev LP. Constraint-matrix perturbations (ΔA) are out of scope. The internal solver is practical for $n \leq 100$; larger problems should use HiGHS, with B^{-1} reconstructed for downstream analysis. Diagnostics for degeneracy (number of degenerate basic variables, $\kappa(B)$, per-constraint degeneracy flags) are computed from the SolveState and reported as part of the Region Analyzer output; details are provided in the Online Supplementary.

4. Sensitivity Region and Factual–Counterfactual Duality

4.1. Simultaneous sensitivity region

OAT sensitivity, reported by commercial solvers, provides ranges for each parameter individually; simultaneous changes within these ranges may still violate basis optimality. The tolerance approach (Wendell, 1985; Filippi, 2005; Borgonovo et al., 2018) provides the theoretical foundation for simultaneous sensitivity, but no publicly available solver implements it. CLARA addresses this gap by computing the Chebyshev center of the basis-preserving polyhedron.

Definition 1 (Basis-preserving polyhedron)**.** *For perturbation $\delta \in \mathbb{R}^m$ (RHS changes), the basis B remains optimal iff $B^{-1}(b+\delta) \geq 0$, equivalently $-B^{-1}\delta \leq x_B$. The set $\mathcal{S}_{\text{raw}} = \{\delta \in \mathbb{R}^m : -B^{-1}\delta \leq x_B\}$ is unbounded (its recession cone is full-dimensional); intersecting with the OAT bounding box yields the bounded polyhedron*

$$\mathcal{S} = \mathcal{S}_{\text{raw}} \cap \left\{\delta : -\alpha_k^- \leq \delta_k \leq \alpha_k^+ \ \forall k\right\} = \{H\delta \leq h\}, \tag{1}$$

with

$$H = \begin{pmatrix} -B^{-1} \\ I_m \\ -I_m \end{pmatrix} \in \mathbb{R}^{3m \times m}, \quad h = \begin{pmatrix} x_B \\ \alpha^+ \\ \alpha^- \end{pmatrix} \in \mathbb{R}^{3m},$$

where α^+, α^- stack the per-parameter OAT allowable increases and decreases. The OAT box rows guarantee boundedness and may be redundant when $\mathcal{S}_{\text{raw}}$ already satisfies the OAT bounds.

Definition 2 (Chebyshev center)**.** *The Chebyshev center of $\mathcal{S}$ is the center δ^* of the largest Euclidean ball inscribed in $\mathcal{S}$, with radius r^*:*

$$r^* = \max_{\delta, r} \ r \quad s.t. \quad H_i\delta + \|H_i\|_2 \cdot r \leq h_i \ \ \forall\, i, \quad r \geq 0, \tag{2}$$

where H and h are defined in (1)*. Since $\mathcal{S}$ is bounded by construction, r^* is finite. This is a linear program with $m+1$ variables and $3m$ constraints.*

Definition 3 (Simultaneity ratio)**.**

$$\rho = \frac{r^*}{\alpha_{\min}}, \qquad \alpha_{\min} = \min_{k:\, \alpha_k^+ > 0} \min(\alpha_k^+, \alpha_k^-), \tag{3}$$

excluding degenerate parameters with zero tolerance. When $\rho < 1$, the inscribed ball radius is smaller than the smallest individual OAT tolerance; ρ is interpretable as the factor by which OAT analysis overestimates the safe perturbation radius.

The polyhedron above considers only RHS perturbations. When both b and c change, the basis is preserved iff both primal and dual feasibility hold.

Primal feasibility requires $B^{-1}(b + \delta_b) \geq 0$ as before. For each nonbasic $j \in \mathcal{N}$, dual feasibility requires

$$\bar{c}_j + \delta_{c_j} - (B^{-1}a_j)^\top \delta_{c_B} \geq 0, \tag{4}$$

which we write as $-g_j^\top \delta_c \leq \bar{c}_j$ with $g_j \in \mathbb{R}^n$ having -1 in position j, $(B^{-1}a_j)_l$ in position $\mathcal{B}_l$ for $l = 1, \ldots, m$, and zero elsewhere. Stacking the $g_j^\top$ into $G \in \mathbb{R}^{|\mathcal{N}| \times n}$, the joint basis-preserving polyhedron becomes

$$\mathcal{S}^+ = \{(\delta_b, \delta_c) \in \mathbb{R}^{m+n} : -B^{-1}\delta_b \leq x_B,\ -G\delta_c \leq \bar{c}\}. \tag{5}$$

As with $\mathcal{S}$, the raw $\mathcal{S}^+$ may be unbounded and should be intersected with OAT bounding boxes. The decoupled structure implies $r_*^+ \leq \min(r_b^*, r_c^*)$, with r_b^*, r_c^* the Chebyshev radii of the primal and dual sub-polyhedra. For visualization, CLARA projects $\mathcal{S}$ (or $\mathcal{S}^+$) onto selected parameter pairs via directional LP scans (see Section 6).

The Chebyshev center inscribes the largest ℓ_2-ball in $\mathcal{S}$, which differs in geometry from the tolerance-based approaches of Wendell (1985) and Borgonovo et al. (2018). The Wendell tolerance inscribes the largest scaled ℓ_∞-ball, yielding a per-parameter percentage guarantee ($t = \pm$ x%); the Borgonovo global tolerance similarly yields a uniform-percentage ℓ_∞-ball but extends the analysis to joint variations in b and c. The Chebyshev radius r^* instead provides an absolute ℓ_2-distance guarantee ($\|\delta\|_2 \leq r^*$), which is the natural quantity for the factual–counterfactual duality of Section 4.2: it is directly comparable with the Euclidean cost of basis-changing CEs. All three approaches share the same input (B^{-1}, x_B) and reduce to a single LP; neither tolerance variant has been previously implemented in an available tool.

4.2. Factual–counterfactual duality

While Section 4.1 characterizes the factual sensitivity region $\mathcal{S}$, counterfactual explanations (Kurtz et al., 2025) ask the complementary question: what is the smallest parameter change driving the solution toward a target

property $\mathcal{D}$? We now formalize the relationship between these two perspectives.

Let $\mathcal{S}_{\text{raw}} = \{\delta \in \mathbb{R}^m : B^{-1}(b + \delta) \geq 0\}$ denote the (unbounded) primal-feasibility region from Definition 1, and write $H = -B^{-1}$, $h = x_B = B^{-1}b$ for its defining constraints. Any CE that requires a basis change must satisfy $\delta^{CE} \notin \text{int}(\mathcal{S}_{\text{raw}})$: the perturbation must violate the primal feasibility of at least one basic variable, regardless of whether it stays within the OAT bounding box.

Definition 4 (Basis robustness)**.** *The* basis robustness *is the minimum Euclidean distance from the origin to the boundary of* $\mathcal{S}_{\text{raw}}$*:*

$$d_0 = \min_{i=1,\ldots,m} \frac{(x_B)_i}{\|(B^{-1})_{i,:}\|_2}. \tag{6}$$

This is computable in $O(mn)$ *time directly from* B^{-1} *and* x_B*, without solving any LP.*

Theorem 1 (Factual–counterfactual bound)**.** *Let* δ^{CE} *be any RHS perturbation that induces a basis change, i.e.,* $\delta^{CE} \notin \text{int}(\mathcal{S}_{\text{raw}})$*. Then*

$$d_0 \leq \|\delta^{CE}\|_2. \tag{7}$$

In particular, no counterfactual explanation with $\|\delta\|_2 < d_0$ *can require a basis change.*

Proof sketch. Primal-infeasibility of δ^{CE} at some row i combined with Cauchy–Schwarz gives $\|\delta^{CE}\|_2 \geq (x_B)_i / \|(B^{-1})_{i,:}\|_2 \geq d_0$. Full proof in the Online Supplementary. □

The bound also identifies the most vulnerable direction for basis change.

Proposition 2 (Cheapest basis flip)**.** *The index* $i^* = \arg\min_i (x_B)_i / \|(B^{-1})_{i,:}\|_2$ *identifies the basic variable that is easiest to drive to zero. The perturbation*

$$\delta^* = -d_0 \cdot \frac{(B^{-1})_{i^*,:}}{\|(B^{-1})_{i^*,:}\|_2} \tag{8}$$

Table 2: Unified factual–counterfactual framework. All quantities are derived from the same basis inverse B^{-1}.

Quantity	Question answered	Computation	Type
$\mathcal{S}$	How far can b change safely?	Polyhedron $\{H\delta \leq h\}$	Factual
d_0	How robust is the basis (primal)?	$O(mn)$, no LP	Factual
d_0^{dual}	How robust is the basis (dual)?	$O(mn)$, no LP	Factual
r^*	What is the largest safe ball?	Chebyshev LP	Factual
δ^* (Prop. 2)	Which direction flips the basis cheapest?	$O(mn)$, no LP	Bridge
d^{CE}	How much change for a desired outcome?	CE formulation	Counterfactual

reaches the boundary of $\mathcal{S}_{\text{raw}}$ *with minimum Euclidean norm, and* $\|\delta^*\|_2 = d_0$.

Proof. The i^*-th primal-feasibility constraint is tight at δ^*, and the remaining constraints are slack by definition of d_0. Full proof in the Online Supplementary. □

Finally, basis robustness, the Chebyshev radius r^*, and the CE cost d^{CE} are linked by a chain of inequalities, $\min(d_0, \alpha_{\min}) \leq r^* \leq d^{CE} + \|\delta^c\|_2$, which simplifies to $d_0 \leq r^*$ when the OAT bounds are non-binding for the inscribed ball (the case in all our experiments). The full statement and proof are given in the Online Supplementary.

The preceding results are stated for RHS perturbations under the Euclidean norm. Analogous bounds for objective perturbations (via a *dual basis robustness* d_0^{dual}) and for alternative ℓ_p norms hold by symmetry; full statements are given in the Online Supplementary.

Remark 1 (Unified interpretation)**.** *Table 2 summarizes how* d_0*,* r^**, and* d^{CE} *unify factual and counterfactual perspectives: a large* d_0 *implies an expensive-to-flip basis, while a small* d_0 *signals a fragile basis where counterfactual explanations are cheaply achievable.*

5. Reoptimization Pipeline

Section 4 addressed the explanation of a *single* LP solution. In practice, LP models are rarely solved once: parameters are updated, scenarios

Table 3: Change type classification and warm-start method selection. Feasibility refers to the old basis applied to the new problem data.

Type	Change	Primal feas.	Dual feas.	Method
R	$\Delta b \neq 0$, $\Delta c = 0$	No	Yes	dual simplex
C	$\Delta b = 0$, $\Delta c \neq 0$	Yes	No	primal simplex
RC	$\Delta b \neq 0$, $\Delta c \neq 0$	No	No	parametric LP
V	new variables added	—	—	scratch

change, and the model is re-solved repeatedly. The reoptimization pipeline in CLARA automates a four-stage process: given an old problem P with SolveState S and a new problem P', (1) DetectChange classifies the perturbation $(\Delta b, \Delta c)$ into a change type τ; (2) AnalyzeImpact decides whether to reoptimize and selects a method m; (3) Reoptimize runs the chosen warm-start method to produce S'; and (4) DiffReport compares S with S' to produce a structured report D.

5.1. Change detection

Given an old problem instance $P = (c, A, b)$ and a new instance $P' = (c', A', b')$, the change detector computes the perturbation vectors $\Delta b = b' - b$ and $\Delta c = c' - c$, along with the relative magnitudes $\delta_b = \|\Delta b\|/\|b\|$ and $\delta_c = \|\Delta c\|/\|c\|$. Following the taxonomy of Albici et al. (2010), the change is classified into one of the types listed in Table 3. This classification determines which warm-start method is applicable, as each type preserves different feasibility properties of the old basis.

5.2. Impact analysis and decision rule

Not every parameter change requires re-solving the LP. The impact analyzer applies a three-stage screening procedure. *First*, for each changed parameter it checks whether the new value falls within the OAT sensitivity range stored in the SolveState; if all changes lie within their ranges and the change is Type R or Type C (not compound), the old basis is guaranteed optimal and no re-solving is needed. (For Type RC, individual OAT ranges do not guarantee simultaneous preservation—see Section 4.1.) *Second*, when this check fails, the analyzer computes an upper bound on the opportunity

cost of not reoptimizing. Let z'^* be the optimal value of the perturbed problem and $z'(x^*) = c'^\top x^*$ the old solution's value under the new objective. Following Oguz (2000),

$$\frac{\text{OC}}{|z'^*|} = \frac{z'(x^*) - z'^*}{|z'^*|} \leq \frac{2\delta}{1+\delta}, \tag{9}$$

where $\delta = \max(\|\Delta b\|/\|b\|, \|\Delta c\|/\|c\|)$. If the bound falls below a threshold ε (default 1%), the change is classified as negligible and re-solving is skipped. The bound assumes x^* remains feasible for the perturbed problem; when $\Delta b \neq 0$, feasibility is checked separately. *Third*, if neither screen permits skipping, the analyzer recommends a warm-start method by change type (Table 3): dual simplex for Type R (primal feasibility violated, dual preserved), primal simplex for Type C (dual violated, primal preserved), and parametric LP for Type RC (both violated). The output is a decision record with the recommended method, Oguz bound value, and a textual reason.

5.3. Warm-start methods

CLARA supports three warm-start methods, one per change type. For Type C changes, *primal simplex warm-start* applies: since c changes alone, $x_B = B^{-1}b \geq 0$ remains primal feasible but reduced costs $\bar{c}'_j = c'_j - (c'_B)^\top B^{-1}a_j$ may turn negative. CLARA restarts primal simplex from the old basis, using the most negative $\bar{c}'_j$ as entering variable and the minimum-ratio test for the leaving variable. Since the start is already primal feasible and typically close to the new optimum, few pivots suffice (our experiments report a mean 28× reduction; see Section 6.5).

For Type R changes, *dual simplex warm-start* is used: reduced costs remain non-negative but $x'_B = B^{-1}(b + \Delta b)$ may have negative entries. The dual simplex picks the most negative x'_{B_i} as leaving variable and performs the dual ratio test

$$j^* = \arg\min_{j:\,(B^{-1}a_j)_i < 0} \frac{-\bar{c}_j}{(B^{-1}a_j)_i}, \tag{10}$$

maintaining dual feasibility while restoring primal feasibility. Under non-

degeneracy, at most m pivots are needed (Bertsimas and Tsitsiklis, 1997); Bland's rule guards against cycling under degeneracy.

For Type RC changes, neither feasibility is preserved, and CLARA applies a *parametric LP* method that traces a path $b(\theta) = b + \theta\Delta b$, $c(\theta) = c + \theta\Delta c$ for $\theta \in [0, 1]$. At each step, the next breakpoint is $\bar{\theta} = \min(\bar{\theta}_P, \bar{\theta}_D, 1)$, where the primal breakpoint is

$$\bar{\theta}_P = \min_{i:\,(B^{-1}\Delta b)_i < 0} \frac{(x_B)_i}{-(B^{-1}\Delta b)_i}, \tag{11}$$

and $\bar{\theta}_D$ is obtained by an analogous ratio test on the reduced costs. If $\bar{\theta} < 1$, a primal or dual pivot is performed and the process continues until $\theta = 1$. The method is exact and handles arbitrary compound changes; moderate perturbations typically need only a few breakpoints (our Albici golden test needs 2).

5.4. Attribution-annotated diff report

The final stage compares old and new SolveStates and produces a diff report; for compound (Type RC) changes the report annotates $\Delta z = z'^* - z^*$ with a per-effect attribution.

Theorem 3 (First-order attribution)**.** *If the optimal basis is preserved under the parameter change, then*

$$\Delta z = \underbrace{y^\top \Delta b}_{\textit{RHS effect}} + \underbrace{\Delta c^\top x^*}_{\textit{objective effect}} + \underbrace{\Delta c_B^\top B^{-1} \Delta b}_{\textit{interaction}}, \tag{12}$$

where $y = c_B^\top B^{-1}$ is the dual vector and Δc_B denotes the components of Δc on basic variables.

Proof. Expand $z' = (c_B + \Delta c_B)^\top B^{-1}(b + \Delta b)$ and collect terms using $y = c_B^\top B^{-1}$, $x_B = B^{-1}b$. Full details in the Online Supplementary. □

The per-parameter contributions are $\mathrm{rhs}_i = y_i \cdot \Delta b_i$ and $\mathrm{obj}_j = \Delta c_j \cdot x_j^*$. When the basis is preserved, (12) is exact; otherwise we measure the nonlinearity by $\eta = |r|/|\Delta z|$, where r is the residual, and complement the decomposition with a Shapley split.

Definition 5 (Shapley attribution)**.** *Let z_b denote the optimal value when only b changes, and z_c when only c changes. The Shapley values for the two players (RHS change, objective change) are $\phi_b = \frac{1}{2}[(z_b - z^*) + (z'^* - z_c)]$ and $\phi_c = \frac{1}{2}[(z_c - z^*) + (z'^* - z_b)]$. By construction, $\phi_b + \phi_c = \Delta z$.*

The Shapley decomposition requires two additional LP solves (one with only Δb, one with only Δc) but guarantees $\phi_b + \phi_c = \Delta z$ regardless of basis changes. A per-parameter Shapley decomposition among all $\Delta b_i, \Delta c_j$ would require 2^{m+n} coalition evaluations and is computationally intractable; CLARA therefore uses first-order attribution for individual parameters and reserves the Shapley mechanism for the two-player RHS-vs-objective split.

Diff report contents.. The diff report combines this attribution with per-variable value/status changes (including basis entries/exits), per-constraint slack and binding-status changes, and a *binding-status shift* flag when the set of active constraints changes.

6. Computational Study

We evaluate CLARA along four dimensions: solver correctness and scalability (Section 6.2), attribution quality (Section 6.3), simultaneous sensitivity analysis (Section 6.4), and reoptimization effectiveness (Section 6.5). All experiments are fully reproducible via a single script included in the repository.

6.1. Experimental setup

Instances.. We use three instance sets: (i) 120 random LPs with $n = m \in \{5, 10, 15, 20, 30, 50, 80, 100\}$, density $d \in \{0.3, 0.5, 0.8\}$, and 5 seeds per configuration; matrix entries and objective coefficients drawn from Uniform$(-10, 10)$, RHS from Uniform$(10, 100)$ to promote feasibility. None of the 120 random instances is degenerate; median $\kappa(B) = 3.2 \times 10^2$ (max 7.8×10^3) and median $d_0 = 0.064$ (min 0.0003). The absence of degeneracy is a limitation of the random set: all 7 Netlib instances (set (ii): afiro, adlittle, sc50a, sc50b, sc105, kb2, share2b) exhibit degeneracy with $d_0 \approx 0$, and the maximum $\kappa(B)$ across the full 127-instance set reaches 4.8×10^5. Set (iii) is the Albici

Table 4: Cross-validation: CLARA internal simplex vs. HiGHS.

Instance group	Instances	Solved	Max gap	Mean gap
Random LP	120	120	3.86×10^{-14}	2.28×10^{-15}
Netlib	7	7	1.51×10^{-12}	2.28×10^{-13}
Total	**127**	**127**	1.51×10^{-12}	1.47×10^{-14}

et al. golden test (Albici et al., 2010) (base LP + 5 reoptimization scenarios of known optimal values).

Perturbations.. For each base instance, we generate 13 perturbation types across 3 seeds, yielding 39 perturbations per instance. Types span RHS-only, objective-only, and compound scenarios at small/medium/large magnitudes (5–50% of original values), plus asymmetric compound variants. This produces approximately 3,500 (base, perturbation) pairs.

Environment.. All experiments were conducted in Python 3.10 with NumPy and HiGHS (via `highspy`) on an Apple M-series processor. Timing results reflect single-threaded execution.

6.2. Solver correctness and scalability

To verify numerical correctness, we solve all 127 instances with both the internal revised simplex solver and HiGHS and compare the optimal values. Table 4 reports the results: the maximum optimality gap across all instances is 1.51×10^{-12} and the mean gap is 1.46×10^{-14}, well within the range of floating-point arithmetic. The internal solver and HiGHS thus produce numerically identical solutions, validating the correctness of CLARA's simplex implementation and B^{-1} computation. Of the 8 Netlib instances attempted, 7 match exactly; the remaining instance (blend, 83 variables, 74 constraints) hits the internal solver's iteration limit due to extensive degeneracy cycling, a known limitation of textbook revised simplex without anti-cycling rules.

At scale, the internal solver handles $n = 50$ in under 1 second and $n = 100$ in about 6 seconds; at $n = 200$, only 2 of 5 seeds finish within the iteration limit, confirming a practical range of $n \leq 100$. Figure 2 shows the log-log scaling, consistent with the $O(m^2 n)$ per-iteration cost of dense revised

simplex. The internal solver is not intended to compete with production solvers on speed; its purpose is to expose B^{-1} for downstream explanation modules.

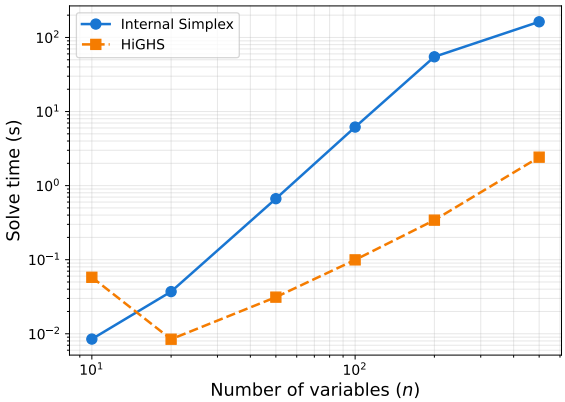

Figure 2: Solve time vs. problem size (log-log, 5 seeds per size). Mean times: 8 ms (n=10), 37 ms (n=20), 0.5 s (n=50), 6.1 s (n=100), 55 s (n=200). CLARA's internal simplex (blue, solid) is substantially slower than HiGHS (orange, dashed) and is practical for $n \leq 100$.

6.3. Attribution

We apply the first-order attribution (Theorem 3) and Shapley decomposition (Definition 5) to all 1,330 compound (RC) perturbation pairs that solve successfully (20 pairs fail due to infeasibility). Table 5 reports the results: mean RHS contribution 148.1%, mean objective contribution 122.8%. Figure 3 shows the breakdown across perturbation types; asymmetric variants (RC_asym_bc, RC_asym_cb) show extreme imbalance with one effect dominating, and large perturbations produce interaction effects exceeding 300% of $|\Delta z|$—confirming that formal decomposition is essential for interpreting compound changes. The basis is preserved in only 1 of 1,330 cases (0.08%), so compound changes almost always trigger basis changes. The nonlinearity measure has mean $\eta = 1.60$ and median 0.30 (max 148.2), so the first-order approximation is reasonable in typical cases but can fail in extremes; the Shapley decomposition, which guarantees $\phi_b + \phi_c = \Delta z$, is therefore essential for reliable attribution.

Table 5: Objective change attribution for RC perturbations. Contributions as percentages of $|\Delta z|$ (may exceed 100% due to opposing effects).

	Pairs	RHS (%)	Obj (%)	Basis preserved
RC_small	270	138.6	131.9	0
RC_medium	270	140.0	135.2	1
RC_large	260	194.4	173.9	0
RC_asym_bc	270	44.4	138.2	0
RC_asym_cb	260	227.8	33.2	0
All RC	**1330**	**148.1**	**122.8**	**1**

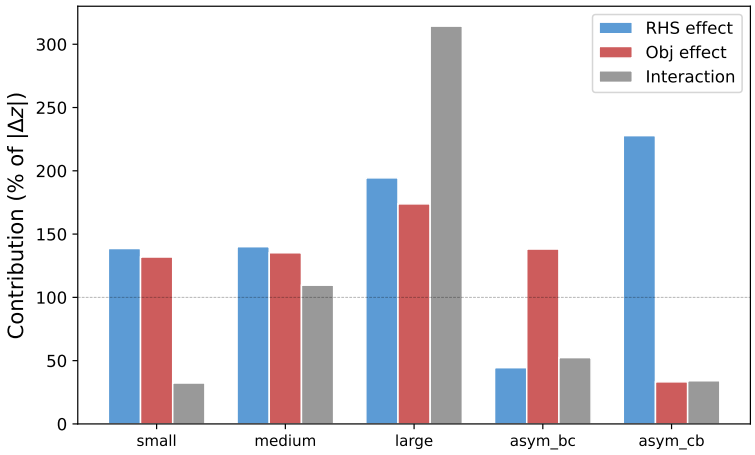

Figure 3: Mean attribution decomposition by RC perturbation type. RHS effect (blue), objective effect (red), and interaction (gray) as percentages of $|\Delta z|$; values above 100% (dashed line) indicate opposing effects that partially cancel. Large and asymmetric scenarios exhibit the strongest interaction effects.

6.4. Simultaneous sensitivity

We compute the Chebyshev center and simultaneity ratio (Definitions 2 and 3) for the 90 random LP instances with $n \leq 50$ (larger instances exceed the internal solver's range for the auxiliary Chebyshev LP). The OAT baseline takes $\alpha_{\min} = \min_k \alpha_k$ over the individual OAT tolerances (excluding degenerate parameters), and $\rho = r^*/\alpha_{\min}$ indicates OAT overestimation when $\rho > 1$.

Table 6 reports results by problem size. The mean ratio is $\rho = 9.9$ with substantial variance (std 23.9), reflecting wide instance-to-instance variation.

Table 6: Simultaneous sensitivity by problem size. $\rho = r^*/\alpha_{\min}$; $\rho > 1$ indicates OAT overestimation.

$n = m$	Instances	Mean r^*	Mean ρ	Std ρ
5–10	30	2.15	12.7	17.5
11–20	30	0.76	7.4	16.3
21–50	30	0.21	9.5	34.2
All	**90**	—	**9.9**	**23.9**

In practical terms, a practitioner trusting individual OAT ranges for simultaneous changes would believe the safe region is roughly 10× larger than the actual basis-preserving polyhedron on average.

Figure 4 (left) shows a 2D projection of the basis-preserving polyhedron for the Albici instance onto constraints 0 and 1. The polygon represents the actual simultaneous safe region, while the dashed rectangle represents the OAT ranges. The rectangle substantially exceeds the polygon, visually confirming the overestimation. Figure 4 (right) shows the distribution of simultaneity ratios across all 90 instances.

The mean ρ does not exhibit a clear monotonic trend with problem size (Table 6), and the correlation between basis robustness d_0 and ρ is weak ($r = 0.10$), as is the correlation between $\kappa(B)$ and ρ ($r = -0.11$), suggesting that the overestimation factor depends more on the specific constraint geometry than on problem dimension alone. The large standard deviations confirm that ρ is highly instance-dependent, and the mean of $\sim$10× should be interpreted as an order-of-magnitude characterization rather than a precise constant. The absolute values of r^* and ρ depend on constraint scaling; normalization strategies for heterogeneous-unit models are discussed in the Online Supplementary.

Beyond the RHS-only region, we evaluate the *joint* $(\Delta b, \Delta c)$ *region* by computing $\mathcal{S}^+$ (equation (5)) on 23 instances with $n \leq 20$. The joint Chebyshev radius r_*^+ is substantially smaller than the RHS-only radius r_b^*: the mean ratio $r_*^+/r_b^* = 0.11$, indicating that dual feasibility constraints (objective coefficient perturbations) reduce the safe region to approximately 11% of the RHS-only region. For the Albici instance, $r_b^* = 185.2$ while

$r_*^+ = 1.17$ (ratio 0.006), reflecting the tight reduced-cost margins of the non-basic variables. Practitioners who assess RHS sensitivity alone may therefore dramatically overestimate the robustness of their solution to simultaneous parameter changes. A *comparison with Wendell tolerance* on the same 90 instances gives a mean tolerance $t = 1.4\%$ (median 0%, max 24.2%); the zero medians reflect degenerate cases where $x_{B_i} = 0$ forces $t = 0$. Wendell's t correlates strongly with the Chebyshev radius ($r = 0.72$), confirming both quantify the same polyhedron; the practical distinction is interpretability, with tolerance giving a percentage guarantee ($\pm 1.4\%$) and the Chebyshev center giving an absolute ℓ_2-distance guarantee ($\|\delta\|_2 \leq r^*$) linked directly to the CE duality (Section 4.2).

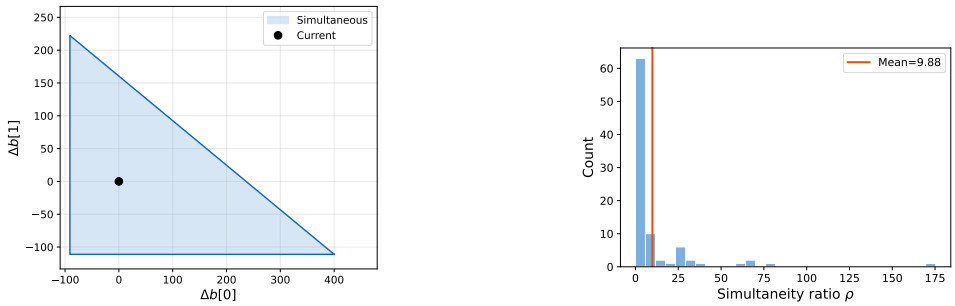

Figure 4: Left: 2D projection of $\mathcal{S}$ for the Albici instance (Δb_0 vs. Δb_1); the shaded polygon is the simultaneous safe region. Right: distribution of simultaneity ratios $\rho = r^*/\alpha_{\min}$ across 90 random LPs; mean $\rho = 9.88$ (orange).

6.5. Reoptimization effectiveness

The impact analyzer (Section 5.2) is evaluated on 3,480 (base, perturbation) pairs. For each pair, the Oguz bound determines whether reoptimization is recommended; the ground truth is obtained by solving the perturbed problem from scratch and checking whether the old optimal value differs from the new one by more than a relative tolerance of 10^{-4}. Table 7 reports the results, with overall accuracy of 64.6%. The false positive rate (29%, 1,008 pairs) dominates the errors: the Oguz bound is conservative by design, and a false positive results only in unnecessary computation, not in suboptimality. The false negative rate is 6.4% (223 pairs), concentrated in Type R changes

Table 7: Reoptimization decision quality. TP/TN: correctly recommends/skips reoptimization; FP: unnecessary (conservative); FN: missed.

Change type	Pairs	TP	TN	FP	FN	Accuracy
Type R	1,340	660	291	243	146	71.0%
Type C	810	88	337	383	2	52.5%
Type RC	1,330	736	137	382	75	65.6%
Total	**3,480**	**1,484**	**765**	**1,008**	**223**	**64.6%**

(146 FN, 10.9% FN rate), where the Oguz bound underestimates the impact of RHS perturbations that are large enough to change the basis but small in relative norm. Type C changes have a negligible FN rate (0.2%, 2 pairs), indicating that the bound is most reliable for objective-only perturbations. The 52.5% accuracy for Type C reflects high FP rates (383 false positives) rather than missed changes.

The dominance of false positives over false negatives across all change types (Table 7) confirms that the Oguz bound errs on the side of caution—recommending unnecessary reoptimization rather than missing beneficial changes.

Warm-start effectiveness.. Of 2,150 perturbation pairs (Table 8), 488 (23%) are routed to warm-start (primal simplex for Type C, dual simplex for Type R); the rest fall back to scratch solving, primarily due to dimension incompatibility from upper-bound augmentation (affecting all instances with $n \geq 50$). Among warm-started pairs, 362 (74%) produce optimal values matching the scratch solution within 10^{-6}. The mean pivot reduction is 28× (scratch 32 vs. warm 0.5 pivots) and the mean wall-clock speedup 4.6× (median 3.7×, max 18.7×); in 69% of cases, the old basis is already optimal for the new problem (0 pivots). Dual simplex (Type R) achieves 4.2× speedup (1.5 vs. 38.6 pivots), primal simplex (Type C) 4.7× (0.2 vs. 30.1 pivots). The gap between pivot (28×) and wall-clock (4.6×) ratios reflects per-pivot overhead of CLARA's educational solver; for production solvers the wall-clock speedup would be closer to the pivot ratio. The full speedup distribution (median 22.0× pivot reduction, 3.7× wall-clock) is shown in the Online Supplementary.

Table 8: Warm-start effectiveness by method. Pivot counts and wall-clock speedups for matched pairs (warm-start value within 10^{-6} of scratch); "Pivot ratio" is the mean of per-instance ratios.

Method	Pairs	WS pivots	Scratch pivots	Pivot ratio	Wall-clock
Primal simplex	279	0.2	30.1	28×	4.7×
Dual simplex	83	1.5	38.6	28×	4.2×
Scratch (fallback)	1,662	—	—	1×	1.0×
Warm-start cases	**362**	**0.5**	**32.0**	**28×**	**4.6×**
All pairs	**2,150**	—	—	—	**1.7×**

When compared to HiGHS's advanced-basis feature, CLARA's internal solver is slower in absolute terms but achieves a 44.5× pivot reduction when warm-started; details are reported in the Online Supplementary. End-to-end validation against the five Albici reoptimization scenarios—covering Type R, C, V, and RC changes—confirms that all optimal values match the ground truth, with the compound scenario using 2 parametric breakpoints versus 5 pivots from scratch (full results in the Online Supplementary).

7. Discussion

Limitations.. The internal solver, a dense revised simplex with $O(m^2 n)$ per-iteration cost, is practical only for $n \leq 100$; this is a deliberate design choice to expose B^{-1} transparently rather than to compete on speed, and any simplex backend can replace it. Under degeneracy, shadow prices are not unique (Koltai and Terlaky, 2000): CLARA reports those of the current basis and flags degenerate constraints but does not enumerate alternative dual solutions. Degeneracy affects 6 of 7 Netlib instances (but not the random set), making the diagnostics of condition number $\kappa(B)$ and degeneracy flags (Online Supplementary) practically important.

The warm-start pipeline requires compatible dimensions between base and perturbed problems; when upper-bound augmentation changes the augmented size, CLARA falls back to scratch. This fallback affects 83% of our perturbation pairs (all instances with $n \geq 50$); standardizing the augmented form is a priority for future work. The joint $(\Delta b, \Delta c)$ region currently

builds G explicitly, $O((n-m)\cdot n)$; on-demand row computation via $B^\top$ back-substitution could reduce memory for large instances. Finally, the FC bound (Theorem 1) lower-bounds the cost of any basis-changing CE, but the gap to actual CE cost may be large when the desired property $\mathcal{D}$ restricts the perturbation to a small parameter subset; tightening this bound for structured CEs is an open question.

Relationship to commercial solvers.. CLARA complements rather than replaces commercial solvers: CPLEX, Gurobi, and HiGHS provide OAT sensitivity but none offers simultaneous regions, attribution, reoptimization decisions, or structured diff reports. The intended use is small-to-medium instances where explanation is the primary goal (a single MPC step, a classroom exercise), or as an explanation layer atop a production solver that exposes basis information. We envision B^{-1} exposure becoming standard in simplex APIs, letting CLARA operate on any backend.

8. Conclusion

We presented CLARA, which contributes two central results to LP explainability: a *factual–counterfactual duality* showing that the basis robustness d_0 (computable in $O(mn)$ from B^{-1}) lower-bounds the cost of any basis-changing counterfactual explanation, and a *reoptimization pipeline* integrating change detection, Oguz-bound impact analysis, automatic warm-start method selection, and attribution-annotated diff reports. Built on the Chebyshev center of the basis-preserving polyhedron, the framework also provides a publicly available implementation of simultaneous sensitivity regions and an exact objective-change attribution module. Experiments on 127 instances and ∼3,500 perturbation pairs validate solver correctness (gap $< 2\times10^{-12}$), 4.6× wall-clock warm-start speedup (28× pivot reduction when applicable), and a tenfold overestimation by OAT of the safe perturbation region. CLARA is available as open-source software under the MIT license.

Future work includes integrating counterfactual explanations (Kurtz et al., 2025) using d_0 and the cheapest flip direction (Proposition 2) as warm-start for CE computation, validating the joint $(\Delta b, \Delta c)$ region on diverse instance

classes, extending the Oguz bound to mixed-integer programs, and using CLARA's structured reports as input to LLMs for natural-language LP explanations.

CRediT authorship contribution statement

Yoonsik Jung: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – Original Draft, Visualization.

Declaration of competing interest

The author declares that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The CLARA software and all experimental scripts are publicly available at `https://github.com/yoonsik-jung-opt/clara-opt` under the MIT license. The package can be installed via `pip install clara-opt`.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author used Claude (Anthropic) in order to improve language and readability. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References

Aigner, K.M., Clarner, J.P., Goerigk, M., Hartisch, M., 2024. Data-driven explanations for optimization models. Preprint.

Aigner, K.M., Clarner, J.P., Kurtz, J., 2025. Coherent local explanations for mathematical optimization (CLEMO). arXiv:2502.04840.

Albici, R., Teselios, D., Tenovici, C., 2010. Reoptimization of linear programming problems. Working paper; golden test instance and ground-truth solutions used in this work are available in the CLARA software repository.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H., 1998. Branch-and-price: Column generation for solving huge integer programs. Operations Research 46, 316–329.

Bertsimas, D., Tsitsiklis, J.N., 1997. Introduction to Linear Optimization. Athena Scientific, Belmont, MA.

Bolusani, S., et al., 2024. The MIP workshop 2023 computational competition on reoptimization. Mathematical Programming Computation 16, 255–266.

Borgonovo, E., Buzzard, G.T., Wendell, R.E., 2018. A global tolerance approach to sensitivity analysis in linear programming. European Journal of Operational Research 267, 321–337.

Busch, F., Zečević, D., Kersting, K., Dhami, D.S., 2025. Elucidating linear programs by neural encodings. Frontiers in Artificial Intelligence 8, 1549085.

Colombo, M., Gondzio, J., Grothey, A., 2011. A warm-start approach for large-scale stochastic linear programs. Mathematical Programming 127, 371–397.

Dantzig, G.B., 1963. Linear Programming and Extensions. Princeton University Press, Princeton, NJ.

De Bock, K.W., Coussement, K., De Caigny, A., 2024. Explainable AI for operational research: A survey. European Journal of Operational Research In press.

Engelhardt, F., Kurtz, J., Birbil, Ş.İ., Ralphs, T.K., 2025. Counterfactual explanations for integer optimization problems. arXiv:2510.17624.

Filippi, C., 2005. A fresh view on the tolerance approach to sensitivity analysis in linear programming. European Journal of Operational Research 167, 1–19.

Gal, T., 1979. Postoptimal Analyses, Parametric Programming, and Related Topics. McGraw-Hill, New York.

Gamrath, G., Hiller, B., Witzig, J., 2015. Reoptimization techniques for MIP solvers, in: Experimental Algorithms (SEA 2015), Springer. pp. 181–192.

Goerigk, M., Hartisch, M., 2023. A framework for inherently interpretable optimization models. European Journal of Operational Research 310, 1128–1150.

Gondzio, J., Grothey, A., 2003. Reoptimization with the primal-dual interior point method. SIAM Journal on Optimization 13, 842–864.

Gondzio, J., Grothey, A., 2008. A new unblocking technique to warmstart interior point methods based on sensitivity analysis. SIAM Journal on Optimization 19, 1184–1210.

Gould, B., Harapanahalli, A., Coogan, S., 2025. linrax: A JAX-compatible, simplex method linear program solver. arXiv:2509.19484.

Huangfu, Q., Hall, J.A.J., 2018. Parallelizing the dual revised simplex method. Mathematical Programming Computation 10, 119–142.

Jansen, B., de Jong, J.J., Roos, C., Terlaky, T., 1997. Sensitivity analysis in linear programming: just be careful! European Journal of Operational Research 101, 15–28.

Koltai, T., Tatay, V., 2011. A practical approach to sensitivity analysis in linear programming under degeneracy for management decision making. International Journal of Production Economics 131, 392–398.

Koltai, T., Terlaky, T., 2000. The difference between the managerial and mathematical interpretation of sensitivity analysis results in linear programming. International Journal of Production Economics 65, 257–274.

Korikov, A., Beck, J.C., 2023. Objective-based counterfactual explanations for linear discrete optimization, in: Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR 2023), Springer. pp. 18–34.

Korikov, A., Shleyfman, A., Beck, J.C., 2021. Counterfactual explanations for optimization-based decisions in the context of the GDPR, in: Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021), pp. 4097–4103.

Kurtz, J., den Hertog, D., Birbil, Ş.İ., 2025. Counterfactual explanations for linear optimization. European Journal of Operational Research In press.

Lübbecke, M.E., Desrosiers, J., 2005. Selected topics in column generation. Operations Research 53, 1007–1023.

Lundberg, S.M., Lee, S.I., 2017. A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems 30 (NeurIPS 2017).

Miftari, B., Derval, G., Ernst, D., Louveaux, Q., 2024. Sensitivity analysis for linear changes of the constraint matrix of a linear program. arXiv:2410.14443.

Oguz, O., 2000. Bounds on the opportunity cost of neglecting reoptimization in mathematical programming. Management Science 46, 1009–1012.

Patel, K.K., 2024. Progressively strengthening and tuning MIP solvers for reoptimization. Mathematical Programming Computation 16, 267–295.

Ravi, N., Wendell, R.E., 1989. The tolerance approach to sensitivity analysis of matrix coefficients in linear programming. Management Science 35, 1106–1119.

Rawlings, J.B., Mayne, D.Q., Diehl, M., 2017. Model Predictive Control: Theory, Computation, and Design. 2nd ed., Nob Hill Publishing.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. "Why should I trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016), pp. 1135–1144.

Ward, J.E., Wendell, R.E., 1990. Approaches to sensitivity analysis in linear programming. Annals of Operations Research 27, 3–38.

Wendell, R.E., 1985. The tolerance approach to sensitivity analysis in linear programming. Management Science 31, 564–578.

Yıldırım, E.A., Wright, S.J., 2002. Warm-start strategies in interior-point methods for linear programming. SIAM Journal on Optimization 12, 782–810.