DEGENERACY NO MORE: A SURPRISINGLY SIMPLE METHOD

A PREPRINT

Zhe Liang
School of Economics and Management
Tongji University
Shanghai, 200092
liangzhe@tongji.edu.cn

Siqi Guo
School of Economics and Management
Tongji University
Shanghai, 200092
guosiqi@tongji.edu.cn

March 5, 2026

ABSTRACT

Degeneracy remains a fundamental challenge in linear programming (LP), severely impacting both the convergence of algorithms like the simplex method and the performance of widely-used decomposition algorithms, such as column generation, Dantzig-Wolfe decomposition, and Benders' decomposition. All these algorithms, which have been documented in tens of thousands of research papers and applied across thousands of applications, critically rely on stable dual values, a requirement that degeneracy directly undermines. Mathematically, a primal degenerate solution could correspond to a great number of dual solutions, leading to issues like cycling, dual oscillations, slow convergence, and misleading shadow prices. Although existing approaches attempt to mitigate degeneracy and stabilize the dual values by all available means, they have not fully resolved this pervasive challenge. In this paper, we propose the Low-Granularity Model (LGM), which completely eliminates LP degeneracy and is remarkably simple to implement, requiring only a double for-loop of fewer than ten lines of computer code. The key innovation of the LGM lies in the introduction of *probe variables* that precisely regulate dual values throughout the optimization process. Despite its simplicity, the LGM offers rich theoretical insights. We rigorously define the *dual optimal central distance* (DOCD) as a quantitative metric for dual solution quality, and prove that the LGM deterministically identifies the dual solution with the minimal DOCD among all dual optima for a degenerate primal optimal solution. Computational experiments on the airline crew scheduling and cutting stock problems demonstrate that the LGM and its extension substantially accelerate convergence while preserving optimality. By delivering stable, high-quality, and unique dual solutions, our approach has the potential to benefit the vast body of research, encompassing tens of thousands of studies, that relies on dual-guided decomposition methods.

Keywords Degeneracy · linear programming · probe variables · dual optimal central distance

1 Introduction

On the long road that led linear programming to its monumental success, a constellation of great names and their pioneering algorithms — such as the simplex method and revised simplex method proposed by Dantzig and Wolfe [Dantzig et al., 1955], the column generation algorithm proposed by Kelley, Gilmore, and Gomory [Kelley, 1960, Gilmore and Gomory, 1961], the ellipsoid method proposed by Khachiyan [Aspvall and Stone, 1980], the interior-point algorithm proposed by Karmarkar [Karmarkar, 1984] — have all pushed the boundaries of effectiveness and efficiency in solving linear programs. Echoing Lord Kelvin's turn-of-the-century observation [Kelvin, 1901], the clear sky of linear programming theory—a field whose profound beauty lies in the intricate dance between primal and dual formulations—is now dimmed by a few lingering clouds. Among these, degeneracy stands as a persistent and troubling one [Hoffman, 2003]. Once overlooked as a minor concern [Beale, 1955], degeneracy has, as linear programming (LP) spread its dominion across academia and industry, struck unexpectedly and profoundly. Astonishingly, the vast majority of real-world problems are degenerate, and the challenges of slow convergence and reduced efficiency introduced by degeneracy remain unconquered to this day.

Theoretically, degeneracy not only slows down the linear programming convergence itself, but also has a broader impact on a number of important decomposition algorithms, especially when tackling the large-scale combinatorial optimization problems. For algorithms such as column generation, Dantzig–Wolfe decomposition, branch-and-price, and Benders' decomposition, the original problem is often decomposed into a master problem (MP) and a set of subproblems, while dual values play a critical role in guiding optimal direction for master problem and subproblems [Bazaraa et al., 2011]. Consequently, the convergence performance of such approaches primarily depends on the quality of the dual solutions. For example, the effectiveness of generated columns [Lübbecke and Desrosiers, 2005] or the strength of cuts [Rahmaniani et al., 2017]. However, when degeneracy occurs, one primal solution might correspond to a large number of dual solutions with identical objective values, which may lead to random and/or less effective columns and cuts, and thus slow down the convergence of the algorithms.

Practically, when applying the aforementioned decomposition algorithms to solve degenerate problems, practitioners often rely on predefined stopping criteria to terminate the algorithms. For example, a time limit [Parmentier et al., 2023], a maximum number of iterations with negligible improvement in the objective [Saddoune et al., 2012], or a small tolerance between upper and lower bounds [Schälicke and Nachtigall, 2025]. These heuristic criteria can generally ensure solutions of acceptable quality, yet they offer no theoretical guarantee of optimality. In many real-world applications, such approximations may suffice; however, the absence of a provably optimal solution continues to motivate the development of more principled and theoretically sound approaches.

Existing methods for addressing degeneracy can be broadly classified into primal-based and dual-based approaches. Primal-side techniques typically introduce perturbations or modified pivoting rules to prevent cycling, whereas dual-side strategies aim to stabilize or regularize the dual variables. While these methods have achieved notable progress in improving convergence in certain cases, none have yet achieved both efficiency and general applicability, and several theoretical and practical issues remain unresolved. The first key issue is about the uniqueness of the dual solution. Since solving the primal model often leads to an optimal basis selected in a largely arbitrary manner, existing methods cannot consistently map all dual solutions associated with a degenerate primal point to a unique representation. In this sense, these methods tend to mitigate degeneracy rather than fundamentally eliminate it, as multiple dual optima may still coexist.

Second, the long-standing question of what constitutes a "good" dual solution has not been satisfactorily answered. In much of the literature, the quality of a dual solution is assessed heuristically. For example, being located near the center of multiple dual solutions or within the interior of the dual space [Subramanian and Sherali, 2008]. Yet, no rigorous quantitative criterion exists to evaluate and compare different dual solutions.

Third, many methods are empirically driven and sensitive to parameter choices, such as the box size in the "box" method [Marsten et al., 1975], the criteria for partition updates in the dynamic constraint aggregation (DCA) [Elhallaoui et al., 2005], the step size and centering parameter in the interior-point method, the weight parameter setting and adjusting scheme in dual smoothing technique [Pessoa et al., 2018], the perturbation magnitude, or even the variable ordering in simplex-type approaches. These parameter dependencies not only influence the convergence efficiency but may also affect the final dual solution obtained. As highlighted in Lübbecke and Desrosiers [2005], degeneracy thus remains an unsolved challenge in practice.

Building upon the above considerations, we propose a Low-Granularity Model (LGM) for a general LP problem, in which auxiliary *probe variables* are introduced to automatically detect the unique dual solution. We prove that, for any degenerate LP problems, the dual solution obtained from the LGM is unique while the primal optimality remains unchanged. By deterministically regulating dual values through a structured mechanism, the LGM completely eliminates degeneracy and guarantees a well-defined dual optimum, which provides a significant advance in dual stabilization for degenerate optimization problems. In contrast, conventional approaches such as the simplex and interior-point methods offer no mechanism to directly control dual values under degeneracy, leaving the dual solution largely unpredictable.

Moreover, we define the *dual optimal central distance* (DOCD) as a rigorous and quantitative criterion for evaluating dual solution quality, measured as the squared Euclidean distance between the given dual solution and the centroid of the dual optimal polytope. We prove that, for a center-symmetric dual optimal polytope with equal weights across all dimensions, the LGM attains the dual solution with the minimal DOCD among all dual optima corresponding to a degenerate primal solution. To further generalize this property, we develop a weighted variant of the LGM (WLGM) that accommodates heterogeneous dual optimal values. When the relative ratios among dual optimal values are known, the WLGM deterministically guides the search toward a unique dual optimum with the minimal DOCD. Remarkably, this yields a more realistic dual representation in practice, one that better reflects the true shadow price of each constraint.

We further validate the effectiveness of the proposed frameworks through two representative applications. Specifically, the LGM is applied to the airline crew scheduling problem, modeled as a set covering problem and known to be

highly degenerate. In this problem, all flight legs are assumed to be equally important, so the dual optimal polytope is center-symmetric with equal weights across all dimensions. The WLGM is applied to the cutting stock problem, where the optimal dual values corresponding to item coverage constraints are roughly proportional to the widths of the items. Computational results on both problems confirm that the proposed models significantly enhance solution efficiency compared with classical formulations.

Finally, the LGM approach is extremely easy to implement. Fewer than ten lines of additional code, typically a double for-loop, are sufficient to extend any original model. Furthermore, the introduction of probe variables is problem-independent, making the method broadly applicable across diverse LP formulations. Consequently, Our proposed method could have a very broad impact in the literature: because tens of thousands of research papers on column generation, Dantzig-Wolfe decomposition, branch and price, and Benders' decomposition suffer the slow convergence due to degeneracy, we believe our method will benefit all these research by providing stabilized, high-quality, and unique dual values and therefore significantly alleviate or completely remove the negative impact of degeneracy.

The remainder of this paper is organized as follows. Section 2 briefly reviews existing research on degeneracy in linear programming. Section 3 introduces the Low-Granularity Model (LGM) using the set covering problem as a benchmark. Section 4 presents theoretical analyses demonstrating the ability of the LGM to deterministically identify the unique dual solution. Section 5 defines the dual optimal central distance as a rigorous quantitative criterion for evaluating dual solution quality, and develops a weighted variant of the LGM (WLGM) to accommodate heterogeneous dual optimal values. Section 6 proposes a no-cycling pivot rule based on the LGM and analyzes the additional computational effort incurred by introducing probe variables. Section 7 reports numerical experiments on the airline crew scheduling and cutting stock problems to demonstrate the practical effectiveness of the proposed approach. Finally, Section 8 presents the main conclusions of this study.

2 Literature review

In this section, we briefly review the studies that aim to alleviate degeneracy in linear programming. The discussion is organized from two complementary perspectives: the primal and the dual. The remainder of this section outlines representative approaches under each perspective.

From the primal perspective, early research primarily aimed to prevent infinite cycling caused by degeneracy by enforcing deterministic pivot selection rules [Chvátal, 1983]. A representative example is the perturbation method proposed by Charnes [1952], which introduces perturbations to the right-hand sides of the constraints to slightly shift the intersection points of hyperplanes. Although not explicitly a pivot rule, this approach effectively imposes an implicit lexicographic order among degenerate bases, thereby ensuring a unique pivot sequence and eliminating cycling. Wolfe [1963] and Ryan and Osborne [1988] both validated the effectiveness of this strategy. Later developments introduced a series of anti-cycling pivoting rules, such as Bland's rule and the lexicographic pivot rule [Bland, 1977, Bazaraa et al., 2011], to ensure finite termination under degeneracy.

From the dual perspective, existing methods aim to obtain more stable or improved dual solutions, thereby mitigating degeneracy and improving convergence. Early approaches include the trust-region-based "box" method [Marsten et al., 1975] that imposes bounds on the dual variables, and related penalization techniques that limit the deviation from known optimal dual values [Wentges, 1997, Du Merle et al., 1999, Ben Amor et al., 2009]. An interior-point stabilization approach [Rousseau et al., 2007] selects dual solutions within the dual space rather than at extreme points, while smoothing and parameter self-adjusting schemes [Pessoa et al., 2018] further improve stability by averaging or adapting dual values.

Another key approach reduces the dimensionality of the dual space by retaining a single representative constraint among a set of constraints, thereby improving dual stability and computational efficiency. The dynamic constraint aggregation (DCA) algorithm [Elhallaoui et al., 2005] reduces the number of primal constraints (and thus dual variables), resulting in dual solutions that can be interpreted as projections of the original ones. Subsequent variants, including the multi-phase DCA [Elhallaoui et al., 2010] and bi-DCA [Elhallaoui et al., 2008], improved performance by incorporating pricing strategies or reducing the size of both the master problem and pricing subproblems. Building on this line of research, the improved primal simplex (IPS) method [Elhallaoui et al., 2011] decomposes the original problem into a reduced problem with fewer constraints and a complementary problem for computing the dual values of discarded constraints. Later, the IPS was also shown to be successfully applied in column generation contexts [Desrosiers et al., 2014, Bouarab et al., 2017].

A further widely applied strategy involves introducing valid inequalities in the dual space, which correspond to additional variables in the primal formulation. For example, Valério de Carvalho [2005] proposed dual cuts for the one-dimensional cutting stock problem, which were later extended by Clautiaux et al. [2011] through introducing cuts

to eliminate solutions considered as combinations of two other solutions, and by Alves and de Carvalho [2008] within branch-and-price-and-cut frameworks. Ben Amor et al. [2006] formalized two classes of dual inequalities: dual-optimal inequalities (DOIs), which preserve all dual-optimal solutions, and deep dual-optimal inequalities (DDOIs), which retain only a subset of dual-optimal solutions. These approaches all operate by expanding the primal solution space, and consequently require additional procedures to recover feasibility and optimality in the original problem. Subsequent generalizations and applications include vector packing, vertex coloring, and bin packing with conflicts [Gschwind and Irnich, 2016]. More recent variants, such as Flexible-DOIs (F-DOIs) [Lokhande et al., 2020], Smooth-Flexible DOIs (SF-DOIs) [Haghani et al., 2022], and dynamic separation of aggregated rows (dyn-SAR) [Costa et al., 2022], further refine dual-space cutting by selectively excluding regions of the dual feasible space, thereby controlling dual values and improving stability in column generation. Recently, Guo et al. [2025] proposed a Lift-DDOI approach, which lifts the dual space without enlarging the original primal solution space, introducing both variables and constraints solely to cut off parts of the dual space.

In summary, existing approaches to mitigating degeneracy exhibit inherent limitations. Although the primal-based methods successfully avoid cycling, this comes at a substantial cost: the search process is forced to traverse around a set of the perturbed vertices that corresponds to a single origin degenerate vertex. Dual-based methods, by contrast, only alleviate degeneracy and do not guarantee a unique dual solution at a degenerate point. As a result, these approaches mitigate the symptoms of degeneracy but do not fundamentally resolve it.

3 Low-granularity model for set covering problem

For a rigorous analysis of degeneracy, we employ a classic and highly degenerate problem: the Set Covering Problem (SCP). It serves as an ideal example for demonstrating the proposed model, theorem, and solution methods, in addition to being one of the most widely adopted models in practice. Formally, let J denote the set of jobs that must be served at least once, indexed by j, and let P represent the set of paths to cover these jobs, indexed by p. For instance, in applications such as the crew pairing problem, vehicle routing problem, or task assignment problem, J may correspond to the set of flights, visits, or tasks, while P may represent flight pairings, vehicle routes, or task sequences, respectively. Each path $p \in P$ is associated with a cost c_p. A binary parameter a_{jp} equals 1 if path p covers job $j \in J$, and 0 otherwise. The primary decision variable $x_p = 1$ if path p is selected, and 0 otherwise. To ensure model feasibility, we introduce an artificial variable u_j for each job $j \in J$, where $u_j = 1$ indicates that job j is not covered by any selected path, and 0 otherwise. Each u_j incurs a sufficiently large penalty θ, such that $\theta \gg c_p$ for all $p \in P$. With these notations, the LP relaxation of the Set Covering Model (SCM) can be formulated as:

$$\text{(SCM)} \quad \min \sum_{p \in P} c_p x_p + \sum_{j \in J} \theta u_j \tag{1}$$

$$s.t. \sum_{p \in P} a_{jp} x_p + u_j \geq 1, \qquad \forall j \in J, \tag{2}$$

$$x_p \geq 0, \qquad \forall p \in P, \tag{3}$$

$$u_j \geq 0, \qquad \forall j \in J. \tag{4}$$

The objective function (1) aims to minimize the total cost, which includes the cost for selected paths and the penalty for uncovered jobs. The set covering constraints (2) ensure that each job is either contained in at least a selected path or is uncovered. The non-negative decision variable constraints are in (3)-(4).

Degeneracy is a common and often unavoidable phenomenon in solving SCMs, since each path variable x_p typically represents a combination of multiple jobs. For instance, a pairing in the crew pairing problem may contain up to 20 flight legs [Saddoune et al., 2012], while a route in the vehicle routing problem can include as many as 50 customers [Fisher, 1995]. As a result, the number of selected path variables is usually much smaller than the number of constraints/jobs. To construct a feasible basis, a large number of zero-valued variables must be included in the basis, which naturally leads to a highly degenerate situation.

In a degenerate primal solution, the number of such zero-valued basic variables is referred to as the degree of degeneracy [Bazaraa et al., 2011]. By selecting different sets of zero-valued variables, the same primal extreme point may correspond to multiple feasible bases. Moreover, Guo et al. [2025] provided a comprehensive analysis of the relationship between the degree of degeneracy and the number of degenerate bases. In other words, a degenerate primal solution can give rise to a large set of dual solutions with identical dual objective values. Such multiplicity renders the dual variables unstable, which in turn may significantly slow down or even obstruct the convergence of algorithms.

To overcome the degeneracy issue, we modify the SCM by introducing $K+1$ *probe variables* $d_{j0}, d_{j1}, \ldots, d_{jk}, \ldots, d_{jK}$ for each constraint, where K is a predefined constant. Each probe variable d_{jk} is assigned a cost $c_{jk} = \frac{\theta k}{K}$, with bounds

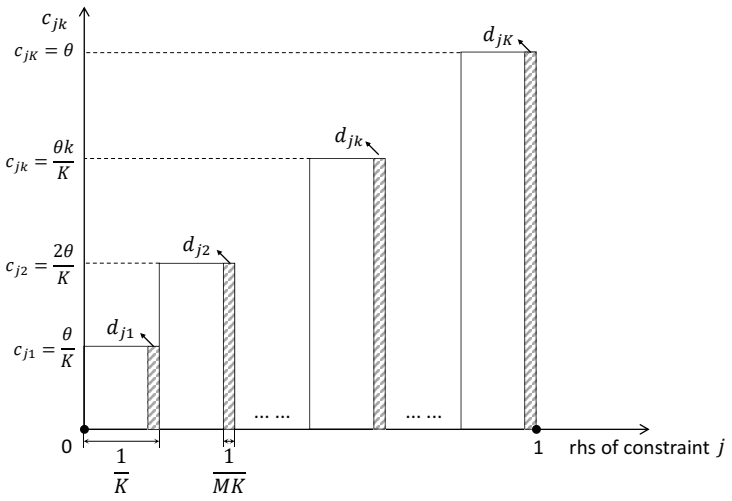


Figure 1: Illustration of the range and cost of probe variables

$0 \leq d_{jk} \leq \epsilon = \frac{1}{MK}$, where M is a sufficiently large constant. By construction, the costs $c_{j0}, c_{j1}, \ldots, c_{jk}, \ldots, c_{jK}$ are in strictly increasing order, so the probe variables are naturally arranged by ascending cost, with $c_{j0} = 0$ and $c_{jK} = \theta$. Intuitively, this setup discretizes the job penalty θ into K equal intervals. For each interval, we create a probe variable d_{jk} with a tiny slice space $\epsilon = \frac{1}{MK}$, as illustrated in Figure 1. In this way, the probe variables are characterized by two critical parameters, K and M. Specifically, K determines the number of probe variables, while M controls the slice size of each probe variable. By increasing K and lowering the granularity of each interval, we will show that the resulting formulation provides surprisingly favorable theoretical and computational properties. Therefore, we refer to the modified model as the Low-Granularity Model (LGM).

$$\text{(LGM)} \quad \min \sum_{p \in P} c_p x_p + \sum_{j \in J} \theta u_j + \sum_{j \in J} \sum_{k=0}^{K} c_{jk} d_{jk} \tag{5}$$

$$s.t. \sum_{p \in P} a_{jp} x_p + u_j + \sum_{k=0}^{K} d_{jk} \geq 1, \qquad \forall j \in J, \tag{6}$$

$$x_p \geq 0, \qquad \forall p \in P, \tag{7}$$

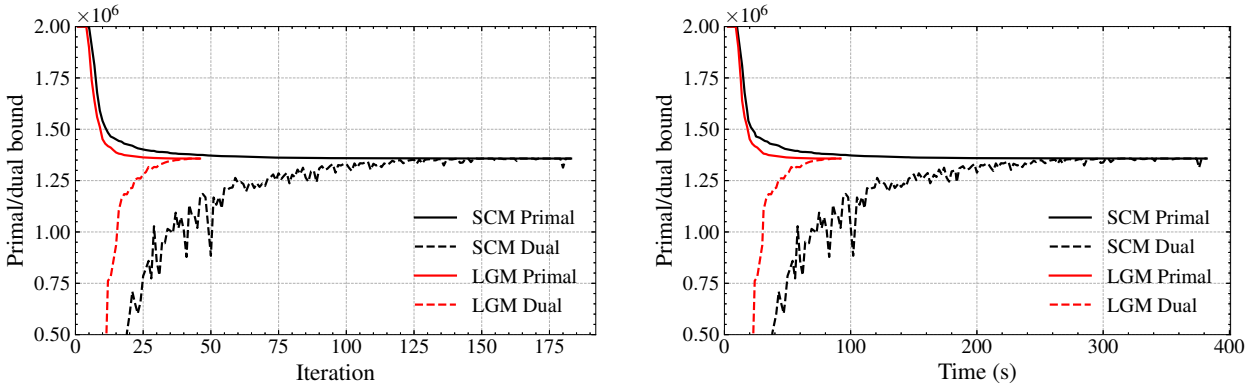
$$u_j \geq 0, \qquad \forall j \in J, \tag{8}$$

$$0 \leq d_{jk} \leq \epsilon, \qquad \forall j \in J, k \in \{0, \cdots, K\}. \tag{9}$$

The model above requires only minor modifications to the SCM and is straightforward to implement. In fact, integrating the probe variables into the original SCM requires fewer than ten lines of code within a simple double for-loop. Moreover, the introduction of probe variables is independent of the problem structure. Despite its simplicity, the approach proves highly effective in resolving degeneracy. For instance, in a small crew rostering case with 11 aircraft, 435 flight legs, and 68 crew members, the LGM demonstrated a striking improvement. As shown in Figure 2, the primal and dual bounds converge rapidly under the LGM. Here, the primal bound corresponds to the current LP objective value, while the dual bound is obtained by adding, for each crew member, the most negative reduced cost among their potential variables (if it exists) to the objective value. With parameters $K = 10$, $M = 1,000$, and $\theta = 15,000$, the number of iterations decreased from 183 (using the SCM) to only 46, yielding a 76% reduction in computational time and nearly eliminating the long-tail effect. As further demonstrated in Section 7, only tens of probe variables per constraint are sufficient to achieve substantial improvement, which is relatively small for large-scale combinatorial optimization problems. Based on our experience, this is likely one of the simplest and most effective methods available in the literature to overcome degeneracy.

4 Theoretical advantages of the LGM

To elucidate the theoretical advantages of the LGM, this section proceeds as follows. First, we establish the relationship between the SCM and the LGM. We then demonstrate that the introduction of probe variables enables precise



(a) Convergence over iterations

(b) Convergence over time

Figure 2: Primal and dual bound convergence in a crew rostering LP example

identification of the dual optimal values for each constraint. Furthermore, we prove that the dual solution obtained via the LGM is unique for any degenerate primal solution, thereby establishing a rigorous theoretical basis for its anti-degeneracy properties.

4.1 The relationship between the SCM and the LGM

In this section, we demonstrate that the introduction of probe variables in the LGM does not affect the optimality of the original SCM, as long as the parameter M is chosen sufficiently large. Intuitively, although the probe variables enlarge the feasible region, their impact can be made arbitrarily small, thereby ensuring that the LGM is asymptotically equivalent to the SCM. Proposition 1 formally establishes the equivalence between the two models from the primal perspective; the proof is provided in Appendix 9.1.

Proposition 1. *Let $(\boldsymbol{x}^M, \boldsymbol{u}^M, \boldsymbol{d}^M)$ denote an optimal solution of the LGM with parameter M. Then, as $M \to \infty$, the components $(\boldsymbol{x}^M, \boldsymbol{u}^M)$ converge to an optimal solution of the original SCM. Conversely, given any optimal solution $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ to the SCM, there exists a corresponding optimal solution $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{d}^*)$ of the LGM such that $\boldsymbol{d}^* \to 0$ as $M \to \infty$.*

Having established the primal equivalence, we now turn to the dual perspective to understand how the dual solutions of the LGM relate to those of the SCM. Specifically, let $\hat{\pi}_j$ denote the non-negative dual variables associated with constraints (2), the corresponding dual problem of the SCM is given by

$$\text{(DSCM)} \quad \max \sum_{j \in J} \hat{\pi}_j \tag{10}$$

$$s.t. \sum_{j \in J} a_{jp} \hat{\pi}_j \leq c_p, \qquad \forall p \in P, \tag{11}$$

$$0 \leq \hat{\pi}_j \leq \theta, \qquad \forall j \in J. \tag{12}$$

Similarly, let π_j denote the non-negative dual variables associated with constraints (6) of the LGM. Consider the probe variable d_{jk} as a free variable with two bounded constraints $d_{jk} \geq 0$ and $d_{jk} \leq \epsilon$. Let σ^-_{jk} denote the non-positive dual variables corresponding to $d_{jk} \leq \epsilon,\ \forall j \in J, k \in \{0, \cdots, K\}$, and let σ^+_{jk} denote the non-negative dual variables corresponding to $d_{jk} \geq 0,\ \forall j \in J, k \in \{0, \cdots, K\}$, then the dual problem of the LGM is given by

$$\text{(DLGM)} \quad \max \sum_{j \in J} \pi_j + \sum_{j \in J} \sum_{k=0}^{K} \epsilon \sigma^-_{jk} \tag{13}$$

$$s.t. \sum_{j \in J} a_{jp} \pi_j \leq c_p, \qquad \forall p \in P, \tag{14}$$

$$\pi_j + \sigma^-_{jk} + \sigma^+_{jk} = c_{jk}, \qquad \forall j \in J, \forall k = \{0, \cdots, K\}, \tag{15}$$

$$0 \leq \pi_j \leq \theta, \qquad \forall j \in J, \tag{16}$$

$$\sigma^-_{jk} \leq 0, \sigma^+_{jk} \geq 0, \qquad \forall j \in J, \forall k = \{0, \cdots, K\}. \tag{17}$$

Then, the following proposition shows that, as $M \to \infty$, the dual optimal solution of the LGM coincides with a dual optimal solution of the SCM, and the proof is provided in Appendix 9.1.

Proposition 2. *Denote by Δ_{DLGM} and Δ_{DSCM} the dual feasible regions of the DLGM and the DSCM, respectively. Let $proj_{\mathbb{R}^{|J|}}(\Delta_{DLGM})$ be the orthogonal projection of Δ_{DLGM} onto the subspace $\mathbb{R}^{|J|}$, we have $proj_{\mathbb{R}^{|J|}}(\Delta_{DLGM}) \subseteq \Delta_{DSCM}$. Let $(\boldsymbol{\pi}^M, \boldsymbol{\sigma}^M)$ be a dual optimal solution of the LGM with parameter M. Then, as $M \to \infty$, the component $(\boldsymbol{\pi}^M)$ converges to a dual optimal solution of the SCM.*

Proposition 1 and 2 establish that, as $M \to \infty$, both the primal and dual solutions of the LGM converge to those of the original SCM. In other words, while the LGM introduces probe variables, these modifications do not alter the primal optimal solutions of the original SCM. These results confirm that the LGM is a faithful reformulation of the SCM at the primal level, but with a smaller dual space. Building on this, we arrive at the following observation.

Observation 1. *The LGM can be interpreted as a lexicographic optimization problem. Specifically, as $M \to \infty$, the objective separates into two components with strictly ordered priorities:*

1. ***Primary objective:** minimize the original set covering cost $\sum_{p \in P} c_p x_p + \sum_{j \in J} \theta u_j$;*
2. ***Secondary objective:** subject to the primary optimum, minimize the total probe variable cost $\sum_{j \in J} \sum_{k=0}^{K} c_{jk} d_{jk}$.*

Under this lexicographic interpretation, the LGM preserves the primal optimal solutions of the SCM, while the additional probe costs serve only to refine the dual space. In the subsequent sections, we will demonstrate that, although the probe costs are asymptotically negligible, the introduction of probe variables provides valuable properties and insights into the SCM.

4.2 Dual values and the costs of probe variables

In the previous section, we introduced probe variables and showed that their inclusion does not affect the optimality of the original problem. In this section, we demonstrate that the probe variables could help to gain more control over the dual optimal solutions. Specifically, we formalize the connection between the dual optimal solution and the costs of probe variables in the following theorem, then show how these costs provide explicit bounds on the dual optimal solution of the LGM (proof in Appendix 9.1).

Theorem 1. *Given optimal primal/dual solutions to the LGM, for any job j, let k^* be the largest index such that $d_{jk^*} > 0$ and therefore $d_{jk^*+1} = 0$. Then the dual optimal value π_j for constraint j must satisfy $c_{jk^*} \leq \pi_j \leq c_{jk^*+1}$.*

From Theorem 1, we can see that the probe variables automatically detect the range of dual value of each constraint. In particular, a probe variable is pushed to its upper bound whenever its associated cost is less than the dual optimal value π_j. As the granularity of the cost intervals of probe variables becomes sufficiently small, the costs of the probe variables can closely approximate the corresponding dual optimal values, providing a practical mechanism to detect and bound the dual optimal solution of the LGM.

4.3 The uniqueness of the dual solution provided by the LGM

Next, we present the following theorem to demonstrate that the dual optimal solution produced by the LGM is unique. Specifically, when the parameters K and M are sufficiently large, this unique dual minimizes $\sum_{j \in J} (\pi_j)^2$. Thus, among all originally degenerate dual optimal solutions, the LGM consistently selects a single and identical solution. A complete proof is provided in Appendix 9.1.

Theorem 2. *As $K \to \infty$ and $M \to \infty$, we have $\lim_{K \to \infty, M \to \infty} \sum_{j \in J} \sum_{k=0}^{K} c_{jk} d_{jk} = \frac{1}{2M\theta} \sum_{j \in J} (\pi_j)^2$, and the corresponding dual optimal solution $\boldsymbol{\pi}^*$ obtained from the LGM is unique.*

To interpret Theorem 2, when the LGM minimizes the objective associated with the probe variables, i.e., $\sum_{j \in J} \sum_{k=0}^{K} c_{jk} d_{jk}$, it simultaneously yields a unique dual solution that minimizes $\sum_{j \in J} (\pi_j)^2$. Unlike traditional approaches that address degeneracy from the dual side, our method introduces a general lifting procedure for the set covering model (and, more broadly, for linear programs) without affecting primal feasibility and optimality. A related attempt is the DOI/DDOI framework of Ben Amor et al. [2006], which also lifts the primal problem. However, in contrast to our method, DOI/DDOI is highly problem-dependent and enlarges the feasible region of the primal problem. Furthermore, our idea also fundamentally differs from the dynamic constraint aggregation approach of Elhallaoui et al. [2005], which projects the primal problem into a lower-dimensional subspace to eliminate zero-valued variables in the projected basis. In contrast, our lifting increases the dimensionality while preserving the original feasible region.

5 Dual optimal central distance

In the previous section, we showed that the LGM yields a unique dual optimal solution that minimizes $\sum_{j\in J}\sum_{k=0}^{K} c_{jk}d_{jk}$, and, when M and K are sufficiently large, it approaches minimizing $\sum_{j\in J}(\pi_j)^2$. Yet, these results alone do not justify why this particular solution should be regarded as superior to other dual optima. Our aim is not only to obtain uniqueness, but to identify a solution that can be deemed the best under a principled criterion. To bridge this gap, we introduce the concept of *dual optimal central distance* as a quantitative measure of dual quality. Building upon this idea, we extend the LGM to general linear programs and propose a weighted version to handle heterogeneous dual structures. Finally, we demonstrate that the directional guidance embedded in the weighted LGM framework provides the foundation for designing a polynomial-time algorithm to approximate the dual optimal center.

5.1 Dual optimal polytope, dual optimal center, and dual optimal central distance

In this section, in order to define a meaningful "center" of dual optimal solutions, we first formalize the set in which this center will be located, i.e., *dual optimal polytope*.

Definition 1 (Dual Optimal Polytope)**.** *Let z^* denote the optimal value of the SCM. Define the feasible region of the DSCM as $\mathcal{D} := \{\boldsymbol{\pi} \in \mathbb{R}^{|J|} \mid \boldsymbol{A}^\top \boldsymbol{\pi} \leq \boldsymbol{c}\}$, and the hyperplane corresponding to the dual optimal value as $\mathcal{H} := \{\boldsymbol{\pi} \in \mathbb{R}^{|J|} \mid \mathbf{1}^T\boldsymbol{\pi} = z^*\}$. The set of all dual optimal solutions, called the* dual optimal polytope*, is then defined as $\mathcal{D}^* := \mathcal{D} \cap \mathcal{H}$, and the set of extreme points of $\mathcal{D}^*$ is denoted by $\Delta_{\mathcal{D}^*}$.*

Depending on the problem instance, $\mathcal{D}^*$ may take various forms: it could be a single point (corresponding to a unique dual optimal solution), a line segment, or, in degenerate cases, a higher-dimensional polyhedron containing multiple optimal solutions. To capture a representative "center" of this set, we define the *dual optimal center* as the centroid of $\mathcal{D}^*$.

Definition 2 (Dual Optimal Center)**.** *Let $\boldsymbol{\pi}^*$ denote the* dual optimal center*, defined as the centroid of the dual optimal polytope $\mathcal{D}^*$: $\boldsymbol{\pi}^* = \frac{1}{|\Delta_{\mathcal{D}^*}|}\sum_{\boldsymbol{\pi}\in\Delta_{\mathcal{D}^*}} \boldsymbol{\pi}$.*

As noted in Goffin et al. [1993] and Martinson and Tind [1999], the center of the dual polyhedron typically provides stronger and more stable dual cuts, resulting in significantly improved computational performance. Motivated by this, we use the *dual optimal center* as a reference for evaluating the quality of any dual optimal solution. Specifically, we measure the squared Euclidean distance between a given dual optimal solution and the dual optimal center, which we term the *dual optimal central distance*, defined as follows:

Definition 3 (Dual Optimal Central Distance (DOCD))**.** *Given a dual optimal solution $\boldsymbol{\pi} \in \mathcal{D}^*$, the* dual optimal central distance *between $\boldsymbol{\pi}$ and the dual optimal center $\boldsymbol{\pi}^*$ is defined by $v(\boldsymbol{\pi}) = \|\boldsymbol{\pi} - \boldsymbol{\pi}^*\|_2^2$.*

Intuitively, a dual optimal solution that lies far from the dual optimal center will have a large DOCD, whereas a solution closer to the center will have a smaller DOCD. In the special case where the dual optimal solution coincides with the dual optimal center, the DOCD is zero.

In practice, however, accurately characterizing the dual optimal polytope and identifying its center are often challenging, which makes the direct computation of the DOCD intractable. To overcome this difficulty, we introduce the concept of the *dual optimal direction*, which captures the relative contribution of each dual component at the dual optimal center. Once this direction is known, the dual optimal center that minimizes the DOCD can be recovered through the following Proposition 3 (proof in Appendix 9.1).

Definition 4 (Dual Optimal Direction)**.** *Define the normalized vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{|J|})$ as* dual optimal direction*, where $\eta_j = \frac{\pi_j^*}{\sum_{j'\in J}\pi_{j'}^*}$ and $\sum_{j\in J}\eta_j = 1$. Here, η_j represents the relative ratio of the j-th dimension in the dual optimal center $\boldsymbol{\pi}^*$.*

Proposition 3. *Given the dual optimal direction $\boldsymbol{\eta}$, defined with the dual optimal center $\boldsymbol{\pi}^*$ for the SCM, the dual optimal solution to $\min_{\boldsymbol{\pi}\in\mathcal{D}^*}\sum_{j\in J}\frac{(\pi_j)^2}{\eta_j}$ coincides with the solution to $\min_{\boldsymbol{\pi}\in\mathcal{D}^*} v(\boldsymbol{\pi})$, which minimizes the DOCD, and is precisely the dual optimal center $\boldsymbol{\pi}^*$.*

The implication of the above proposition is that it is not necessary to know the absolute value of each dual optimal variable $\boldsymbol{\pi}^*$ or the exact shape of the dual optimal polytope $\mathcal{D}^*$; rather, we only need to consider the relative ratios between different dual variables, i.e., the direction of dual optimal center. This requirement is often much easier to satisfy in real-world applications. For example, in many set covering problems, where tasks typically have comparable importance, the dual optimal polytope usually exhibits a nearly center-symmetric structure. Consequently, the relative

ratios among the dual optimal values of different tasks tend to be close to one, reflecting minor variations in their shadow prices. Besides, in the case of a degenerate solution, it is reasonable to expect that all tasks along a selected path share similar shadow prices. Conversely, a scenario in which a single task absorbs the entire dual value while others receive none is generally unacceptable.

Recalling that the LGM minimizes the squared L2 norm of the dual values across all dual optima, we now establish a more formal connection. When all components of the dual optimal direction are equal, i.e., $\eta_j = 1/|J|, \forall j \in J$, this minimization is equivalent to minimizing the DOCD. The following theorem formalizes this relationship and shows that the dual optimal solution of the LGM indeed attains the minimum DOCD under this condition (proof in Appendix 9.1).

Theorem 3. *When the dual optimal direction $\boldsymbol{\eta} = \frac{1}{|J|}\mathbf{1}$, as $K \to \infty$ and $M \to \infty$, the LGM produces a unique dual optimal solution, precisely the dual optimal center among all dual optimal solutions of the SCM.*

5.2 Weighted LGM for general linear programming

In the previous section, we introduced the concept of the DOCD, which is well defined regardless of the problem structure and provides a consistent basis for comparing multiple dual solutions. Furthermore, we show that in the set covering problem, if the dual optimal polytope is relatively center-symmetric, that is, the components of the dual optimal direction are similar across different jobs, the dual optimal solution with the minimal DOCD can be obtained by the LGM as $K \to \infty$ and $M \to \infty$. However, such symmetry does not necessarily hold in more general settings. For example, in the cutting stock problem, the demand associated with longer items typically yields larger dual values than that of shorter items [Valério de Carvalho, 2005, Ben Amor et al., 2006]. Furthermore, the demand for different items might no longer be the same. In this case, the components of the dual optimal direction are no longer uniform, and the LGM in its original form may fail to produce the dual solution with the minimal DOCD. This observation motivates an extension of the LGM to accommodate heterogeneous dual components, allowing it to compute the dual optimal solution with the minimal DOCD even when the dual values vary across constraints.

Consider the linear program P below. For notational simplicity, relative to the SCM presented earlier, we modify the right-hand side of constraint j from 1 to $b_j \in \mathbb{R}^+$, and allow $a_{jp} \in \mathbb{R}^+$ to be a general positive constant rather than restricting it to $\{0, 1\}$. Additionally, the penalty θ_j for u_j is now allowed to vary across different constraints, in contrast to the previous setting where all u_j shared the same cost parameter θ.

$$\text{(P)} \quad \min \sum_{p \in P} c_p x_p + \sum_{j \in J} \theta_j u_j \tag{18}$$

$$s.t. \sum_{p \in P} a_{jp} x_p + u_j \geq b_j, \qquad \forall j \in J, \tag{19}$$

$$x_p \geq 0, \qquad \forall p \in P, \tag{20}$$

$$u_j \geq 0, \qquad \forall j \in J. \tag{21}$$

To analyze the dual optimal direction of the LP problem P, we seek to transform the problem into a form that closely resembles the SCM, allowing us to apply its conclusions. Specifically, we scale each constraint j by its right-hand side b_j, and introduce a variable substitution by defining $\hat{u}_j = \frac{1}{b_j} u_j$. This modification preserves the problem's structure while normalizing the right-hand side to 1. The resulting scaled form of the LP problem P is given by:

$$\text{(ScaledP)} \quad \min \sum_{p \in P} c_p x_p + \sum_{j \in J} \theta_j b_j \hat{u}_j \tag{22}$$

$$s.t. \sum_{p \in P} \frac{a_{jp}}{b_j} x_p + \hat{u}_j \geq 1, \qquad \forall j \in J, \tag{23}$$

$$x_p \geq 0, \qquad \forall p \in P, \tag{24}$$

$$\hat{u}_j \geq 0, \qquad \forall j \in J. \tag{25}$$

Define the dual variables of the constraints in (23) as μ_j for all $j \in J$. It is straightforward to observe that after scaling, $\mu_j = b_j \pi_j$. According to Definition 4, the dual optimal direction of the scaled problem $ScaledP$ is given by $\eta_j = \frac{\mu_j^*}{\sum_{j' \in J} \mu_{j'}^*}, \forall j \in J$, where $\boldsymbol{\mu}^*$ represents the dual optimal center of $ScaledP$. Since $\mu_j = b_j \pi_j$, the dual optimal direction of the scaled problem can be expressed in terms of the original dual optimal center $\boldsymbol{\pi}^*$ of the problem P, leading to the following definition of the *weighted dual optimal direction*:

Definition 5 (Weighted Dual Optimal Direction)**.** *Let $\boldsymbol{\pi}^*$ denote the dual optimal center of the LP problem P and define the normalized vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{|J|})$ as the* weighted dual optimal direction*, where $\eta_j = \frac{b_j \pi_j^*}{\sum_{j' \in J}(b_{j'} \pi_{j'}^*)}$ and $\sum_{j \in J} \eta_j = 1$. Here, each component η_j measures the relative weighted importance of the j-th dimension in $\boldsymbol{\pi}^*$.*

Geometrically, as illustrated in Figure 3, the weighted dual optimal direction points from the origin toward the dual optimal center $\boldsymbol{\mu}^*$ of the scaled problem $ScaledP$. By reversing the scaling transformation applied to $\boldsymbol{\mu}^*$, we can recover the exact dual optimal center $\boldsymbol{\pi}^*$ of the original LP problem P. Similarly, once the weighted dual optimal direction is determined, minimizing the DOCD leads to an optimal solution identical to that of minimizing the weighted squared L2 norm of the dual values across all dual optimal solutions, as stated in Proposition 4 (proof in Appendix 9.1).

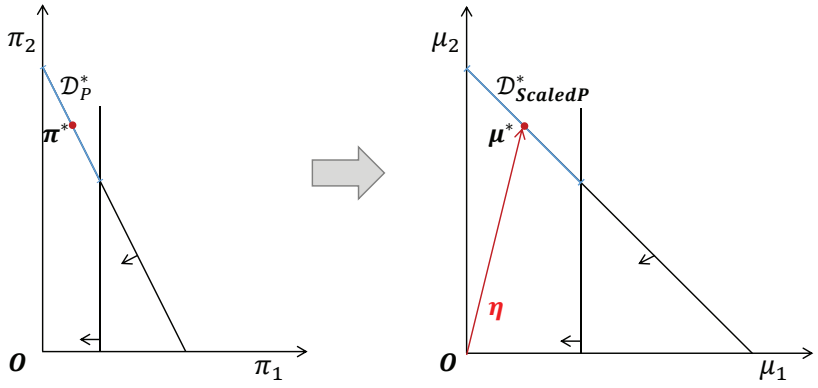


Figure 3: Illustration of weighted dual optimal direction

Proposition 4. *Given the weighted dual optimal direction $\boldsymbol{\eta}$, defined based on the dual optimal center $\boldsymbol{\pi}^*$ for the LP problem P, the optimal solution to $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} \sum_{j \in J} \frac{(b_j \pi_j)^2}{\eta_j}$ coincides with the solution to $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} v(\boldsymbol{\pi})$, which minimizes the DOCD and is precisely $\boldsymbol{\pi}^*$.*

To obtain the unique dual optimal solution that minimizes the DOCD, a variant of the LGM, based on the weighted dual optimal direction $\boldsymbol{\eta}$, is proposed and referred to as the WLGM:

$$\text{(WLGM)} \quad \min \sum_{p \in P} c_p x_p + \sum_{j \in J} \theta_j u_j + \sum_{j \in J} \sum_{k=0}^{K} c_{jk} d_{jk} \tag{26}$$

$$s.t. \sum_{p \in P} a_{jp} x_p + u_j + \sum_{k=0}^{K} b_j d_{jk} \geq b_j, \qquad \forall j \in J, \tag{27}$$

$$x_p \geq 0, \qquad \forall p \in P, \tag{28}$$

$$u_j \geq 0, \qquad \forall j \in J, \tag{29}$$

$$0 \leq d_{jk} \leq \epsilon, \qquad \forall j \in J, k \in \{0, \cdots, K\}. \tag{30}$$

Here, $\theta_j = \theta \eta_j$, and thus $c_{jk} = \frac{\theta_j k}{K}$. That is, all θ_j values are adjusted according to the weighted dual optimal direction, reflecting the weighted importance of each constraint. As a result, the costs of the probe variables for each constraint are changed accordingly. Similar to Propositions 1 and 2, it can be shown that as $M \to \infty$, the WLGM becomes asymptotically equivalent to the problem P, and the dual optimal solution of the WLGM coincides with a dual optimal solution of the problem P. Next, in the following theorem, we demonstrate that the WLGM precisely provides the dual optimal solution with the minimal DOCD, even when the dual optimal polytope is asymmetric (proof in Appendix 9.1).

Theorem 4. *Let $\mathcal{D}^*$ denote the dual optimal polytope of the LP problem P, and let $\boldsymbol{\pi}^*$ represent its dual optimal center. Given the weighted dual optimal direction $\boldsymbol{\eta}$ defined with $\boldsymbol{\pi}^*$, as $K \to \infty$ and $M \to \infty$, the unique dual optimal solution proposed by the WLGM is exactly $\boldsymbol{\pi}^*$.*

When degeneracy arises, traditional approaches fail to control the search direction of the dual variables, often resulting in uncertain or non-unique dual solutions. By introducing probe variables whose costs are calibrated according to the weighted dual optimal direction, Theorem 4 shows that the WLGM explicitly steers the dual search along a prescribed direction. Consequently, if a dual optimal solution exists in that direction, it will be attained. This mechanism enhances both the stability and interpretability of dual solutions under highly degenerate instances, and offers a systematic means of incorporating prior knowledge about the expected dual structure to improve computational reliability.

5.3 Directional guidance of the WLGM

In theory, if the exact weighted dual optimal direction $\boldsymbol{\eta}$ is known, it would guide the WLGM to recover the dual optimal center, as established in Theorem 4. Specifically, by intersecting the ray $\{r\boldsymbol{\eta} : r \geq 0\}$ with the dual optimal polytope of the corresponding $ScaledP$, the intersection point can be scaled back to recover the dual optimal center of the original LP problem P. In practice, however, the exact direction is rarely available. A natural approach is to begin with an estimated direction derived from prior knowledge or experience, though such an estimate may deviate from the exact value. This raises a fundamental question: what happens if the prescribed direction does not match any solution in $\mathcal{D}^*$?

Proposition 5 addresses this scenario. It shows that, depending on whether the specified direction intersects with the scaled dual optimal polytope, the scaled dual optimal solution obtained from the WLGM either aligns with this direction or, if the direction lies outside the scaled polytope, resides on its boundary (proof in Appendix 9.1).

Proposition 5. *Let $\mathcal{D}^*$ denote the dual optimal polytope of problem P, and $\mathcal{D}^*_{ScaledP}$ the dual optimal polytope of its scaled counterpart. Given an estimated weighted dual optimal direction $\hat{\boldsymbol{\eta}}$, the dual optimal solution $\boldsymbol{\pi}^{\hat{\eta}}$ produced by the WLGM satisfies the following properties:*

1. *If the ray $\{r\hat{\boldsymbol{\eta}} : r \geq 0\}$ has a nonempty intersection with $\mathcal{D}^*_{ScaledP}$, then $\pi_j^{\hat{\eta}} = \frac{r\hat{\boldsymbol{\eta}}_j}{b_j}, \exists r \geq 0$.*
2. *Otherwise, $\boldsymbol{\pi}^{\hat{\eta}}$ lies on the boundary of $\mathcal{D}^*$.*

In summary, Proposition 5 implies that if the estimated dual optimal direction $\hat{\boldsymbol{\eta}}$ is reasonable, it can guide the WLGM to directly attain the target dual optimal solution. Such interior solutions are often sufficient in practice to alleviate degeneracy, providing a stable and well-centered dual representation without requiring further refinement. Conversely, the WLGM would adjust the estimated weighted dual optimal direction $\hat{\boldsymbol{\eta}}$ to a solution on the boundary of $\mathcal{D}^*$. The difference $(z^*\hat{\boldsymbol{\eta}} - \boldsymbol{\pi}^{\hat{\eta}})$ can then serve as a meaningful direction for updating the estimated weighted dual optimal direction.

Leveraging this property, the WLGM can be used to probe the boundary of the dual optimal polytope. By purposely setting the initial direction $\boldsymbol{\eta}$ and varying the costs associated with the probe variables, one can collect a diverse set of boundary solutions, which in turn provides an approximation of the dual optimal polytope. This observation motivates the following section, which presents a tractable boundary-sampling algorithm for estimating the dual optimal center based on representative points generated by the WLGM.

5.4 Estimated dual optimal center

Building on the directional selection property of the WLGM, this section develops an algorithmic framework for estimating the dual optimal center. Computing the exact physical center of the dual optimal polytope $\mathcal{D}^*$ is known to be NP-hard for general linear programs, but the boundary-sampling capability of the WLGM provides a tractable alternative: by generating representative boundary points, we can construct a reliable approximation in polynomial time.

The intuition is as follows. Referring to Proposition 5, by deliberately varying the weighted dual optimal direction $\boldsymbol{\eta}$, we can guide the WLGM to produce a sufficiently rich set of dual solutions distributed across the boundary of $\mathcal{D}^*$. Collectively, the rays emanating from the origin and passing through these boundary points form a cone that encloses $\mathcal{D}^*$, while the boundary points themselves offer a discrete yet informative approximation of its geometry. In practice, however, such exhaustive sampling is unnecessary. Instead, we propose a lightweight approximation scheme that requires only a limited set of rays to capture the essential geometry of $\mathcal{D}^*$. Specifically, we adopt directions of the form

$$\boldsymbol{\eta} \;=\; \xi\mathbf{1} \;+\; \big(1 - \xi|J|\big)\boldsymbol{e}_j, \tag{31}$$

where $\boldsymbol{e}_j$ is the j-th unit vector and ξ is a sufficiently small positive constant. Intuitively, this construction enforces one dual component to be dominant while the others remain small. Repeating this process for each dimension $j \in J$ yields a collection of boundary solutions that approximate the geometry of the dual optimal polytope. By averaging these solutions, we obtain a practical approximation of the dual optimal center. The procedure is summarized in Algorithm 1, and a simple, concrete example is provided in Example B.1 (Appendix 9.2) to illustrate both the construction and sampling process.

Algorithm 1 might be valuable in applications such as column generation. In large-scale settings, it is infeasible to enumerate all primal variables (equivalently, all dual constraints), and therefore the exact shape of $\mathcal{D}^*$ is unattainable. Nevertheless, by generating representative boundary points through the WLGM, we can form meaningful approximations of the dual optimal center. These approximations, in turn, provide effective guidance for stabilizing and accelerating the column generation process. Although our primary focus is on approximating the centroid of the dual

Algorithm 1: Approximation of the dual optimal center

Input: Parameter ξ
Output: Approximate dual optimal center $\hat{\boldsymbol{\pi}}$
Initialize $\hat{\boldsymbol{\Pi}} \leftarrow \emptyset$;
for *each constraint* $j \in J$ **do**
  Construct weighted dual optimal direction $\boldsymbol{\eta} = \xi \mathbf{1} + (1 - \xi|J|)\boldsymbol{e}_j$;
  Solve the WLGM under direction $\boldsymbol{\eta}$ to obtain the dual optimal solution $\boldsymbol{\pi}$;
  Update the set $\hat{\Pi} \leftarrow \hat{\Pi} \cup \{\boldsymbol{\pi}\}$;
Compute $\hat{\boldsymbol{\pi}} = \frac{1}{|J|} \sum_{\boldsymbol{\pi} \in \hat{\boldsymbol{\Pi}}} \boldsymbol{\pi}$;
return $\hat{\boldsymbol{\pi}}$;

optimal polytope, the same approach can be extended to other notions of center—such as the analytic center, the Chebyshev center, or the barycentric centroid. Since the WLGM inherently samples boundary points of $\mathcal{D}^*$, once these samples are obtained, different computational schemes can be readily applied to approximate the desired center.

From a practical perspective, however, computing the exact center—even with Algorithm 1—can still be time-consuming and is not strictly necessary. While the true center may offer slightly better numerical performance, it is often sufficient to select any direction corresponding to an interior point of $\mathcal{D}^*$. This choice ensures effective control of the dual solution, allowing the WLGM to consistently deliver a unique outcome that, after scaling, aligns with the prescribed direction.

6 Implementation and computational issues

This section aims to address two important aspects of the proposed LGM. In Section 6.1, we introduce a pivoting rule for the LGM that eliminates cycling, and in Section 6.2, we demonstrate that the introduction of probe variables leads to only a marginal increase in computational complexity. While most commercial solvers can already handle cycling issues, and this is typically not a concern for practitioners, we provide a rigorous proof that the proposed algorithm can effectively prevent cycling, ensuring theoretical completeness. Similarly, although the number of probe variables is usually much smaller than the general variables in a highly degenerate problem, and can be easily managed by commercial solvers, we show that their introduction results in only minimal computational overhead.

6.1 Elimination of cycling under degeneracy

In the previous sections, we demonstrated that the LGM can select a unique dual optimal solution from the set of all dual optimal solutions of the SCM, which correspond to a degenerate primal optimal solution. Building on this property, we now propose a new pivoting rule to eliminate cycling caused by degeneracy.

The pivoting procedure follows a clear hierarchy in which probe variables are given absolute priority over non-probe variables. A probe variable is always selected for update whenever its reduced cost is favorable, and only after all probe variables have been processed and no further updates are possible do non-probe variables become eligible to enter the basis. Notably, this priority is enforced even when a non-probe variable has a more attractive reduced cost than a probe variable. The step-by-step implementation of this pivoting rule is summarized in the following algorithm.

Algorithm 2. Probe Cycling-Free Pivoting Rule

Initialize Given the current partition of the coefficient matrix $[\boldsymbol{B}, \boldsymbol{N}_1, \boldsymbol{N}_2]$, all decision variables (both $\boldsymbol{x}$ and $\boldsymbol{d}$) partitioned accordingly as $(\boldsymbol{x}_{\boldsymbol{B}}, \boldsymbol{x}_{\boldsymbol{N_1}}, \boldsymbol{x}_{\boldsymbol{N_2}})$, where $\boldsymbol{x}_{\boldsymbol{B}}$ are the basic variables, and $\boldsymbol{x}_{\boldsymbol{N_1}}$ and $\boldsymbol{x}_{\boldsymbol{N_2}}$ are the nonbasic variables at their lower and upper limits, respectively.

Step 0 Compute the basic feasible solution $\bar{\boldsymbol{x}}_{\boldsymbol{B}}$, the dual solution $\boldsymbol{\pi}$, and the objective value z.

Step 1 Compute the reduced costs $\bar{c}_{jk} = c_{jk} - \pi_j$ of all probe variables $d_{jk}, \forall j \in J, k \in \{0, \cdots, K\}$.

Step 2 Consider all probe variables at their lower bounds with $\bar{c}_{jk} < 0$ and at their upper bounds with $\bar{c}_{jk} > 0$. From this subset of probe variables, select the variable $d_{j^*k^*}$ with the largest absolute reduced cost $|\bar{c}_{jk}|$. If $d_{j^*k^*} \in \boldsymbol{x}_{\boldsymbol{N_1}}$, proceed to Step 2.1; if $d_{j^*k^*} \in \boldsymbol{x}_{\boldsymbol{N_2}}$, proceed to Step 2.2; if no such probe variable exists, proceed to Step 3.

2.1 Compute the direction vector $\boldsymbol{y} = \boldsymbol{B}^{-1}\boldsymbol{e}_{j^*}$. The step size $\tau = \min\left\{\min_{y_i>0} \frac{\bar{x}_{B,i}}{y_i}, \min_{y_i<0} \frac{u_{B,i}-\bar{x}_{B,i}}{-y_i},\ \epsilon\right\}$, here $u_{B,i}$ is the upper bound for the i-th basic variable. The conventions are: $\min_{y_i>0} \frac{\bar{x}_{B,i}}{y_i} = +\infty$ if no $y_i > 0$ exists, and $\min_{y_i<0} \frac{u_{B,i}-\bar{x}_{B,i}}{-y_i} = +\infty$ if no $y_i < 0$ exists. Then:

- If $\tau = \epsilon$, move $d_{j^*k^*}$ from $\boldsymbol{x}_{\boldsymbol{N_1}}$ to $\boldsymbol{x}_{\boldsymbol{N_2}}$, set its value to τ, and return to Step 0.
- If $\tau < \epsilon$, perform a pivot with step size τ: $d_{j^*k^*}$ enters the basis with value τ and the leaving basic variable is determined by the minimum ratio test. Update the variable partition and return to Step 0.

2.2 Compute the direction vector $\boldsymbol{y} = \boldsymbol{B}^{-1}\boldsymbol{e}_{j^*}$. The step size $\tau = \min\left\{\min_{y_i>0} \frac{u_{B,i}-\bar{x}_{B,i}}{y_i}, \min_{y_i<0} \frac{\bar{x}_{B,i}}{-y_i},\ \epsilon\right\}$, here $u_{B,i}$ is the upper bound for the i-th basic variable. The conventions are: $\min_{y_i>0} \frac{u_{B,i}-\bar{x}_{B,i}}{y_i} = +\infty$ if no $y_i > 0$ exists, and $\min_{y_i<0} \frac{\bar{x}_{B,i}}{-y_i} = +\infty$ if no $y_i < 0$ exists. Then:

- If $\tau = \epsilon$, move $d_{j^*k^*}$ from $\boldsymbol{x}_{\boldsymbol{N_2}}$ to $\boldsymbol{x}_{\boldsymbol{N_1}}$, set its value to 0, and return to Step 0.
- If $\tau < \epsilon$, perform a pivot with step size τ: $d_{j^*k^*}$ enters the basis with value $\epsilon - \tau$ and the leaving basic variable is determined by the minimum ratio test. Update the variable partition and return to Step 0.

Step 3 Compute the reduced costs of all non-probe non-basic variables.

Step 4 If any non-probe variable has a negative reduced cost, select the one with the most negative reduced cost to enter the basis and determine the leaving variable using the classical minimum ratio test. If no such variable exists, the current solution is optimal.

The following theorem rigorously establishes that this modified pivoting rule completely eliminates cycling in degenerate situations (proof in Appendix 9.1).

Theorem 5. *Consider a (possibly degenerate) basis $\boldsymbol{B}$ of the LGM. Under the* Probe Cycling-Free Pivoting Rule*, by first updating all probe variables before selecting the entering non-probe variable, cycling due to degeneracy is completely eliminated.*

Although the above algorithm and theorem focus on the set covering model, the proposed pivot rule is applicable to general linear programming problems. To demonstrate this generality, we revisit a classical example (Example B.2 in Appendix 9.2) from Beale [1955], where the standard simplex method cycles and eventually returns to the initial basic solution after six iterations.

6.2 Computational efficiency analysis of the LGM

In this section, we analyze the computational efficiency of the LGM from two perspectives. The first concerns the per-iteration complexity when a probe variable enters the basis. Since the most computationally demanding step in such an iteration is updating the inverse of the feasible basis matrix, we characterize the complexity of computing the inverse of the new basis after a probe variable enters. The second concerns the number of additional iterations required to reach the LGM optimal basis starting from a degenerate optimal basis of the SCM. Together, these two aspects capture the extra computational effort incurred by the LGM compared with the SCM.

First, the following proposition formalizes the computational cost of updating the basis inverse when a probe variable enters the basis (proof in Appendix 9.1).

Proposition 6. *Let $\boldsymbol{B} \in \mathbb{R}^{|J|\times|J|} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_j, \ldots, \boldsymbol{b}_{|J|}\}$ be a feasible basis of the LGM, and let $\boldsymbol{B}^{-1}$ denote its inverse. Suppose a probe variable $d_{j^*k^*}$ enters the basis and the i^*-th basic variable leaves. The new basis can be expressed as $\boldsymbol{B}' = \boldsymbol{B} + (\boldsymbol{e}_{j^*} - \boldsymbol{b}_{i^*})\boldsymbol{e}_{i^*}^T$, where $\boldsymbol{e}_{j^*}$ and $\boldsymbol{e}_{i^*}$ are unit vectors with 1 in the j^*-th and i^*-th components, respectively. The inverse of the new basis $\boldsymbol{B}'$ can be computed in $O(|J|^2)$ time.*

This result shows that although the basis inverse needs to be updated in each iteration, the cost remains quadratic in the basis dimension, which is significantly cheaper than updating the inverse when a general variable enters the basis, i.e., $O(|J|^3)$.

We now analyze the number of additional iterations required for the LGM to reach its optimal basis from a degenerate optimal basis of the SCM. According to Theorem 1, the optimal dual value of constraint j in the LGM lies within the interval bounded by the probe variable with the largest cost among those with positive values and the probe variable with the smallest cost among those with zero values. Intuitively, for constraints whose optimal dual values have already been attained, the reduced costs of their associated probe variables no longer satisfy the conditions for entering the basis, and hence no further pivoting is needed. In contrast, for those constraints whose dual values have not yet converged to the optimal ones, the corresponding probe variables will continue to be adjusted through successive iterations. Therefore,

the upper bound on the number of additional iterations required by the LGM is intrinsically determined by the optimal dual values, which directly leads to the following proposition (proof provided in Appendix 9.1).

Proposition 7. *Given a degenerate optimal basis of the SCM, the number of iterations required to obtain the optimal basis of the corresponding LGM is bounded above by* $\sum_{j\in J}\left\lceil \frac{\pi_j^* K}{\theta}\right\rceil$.

This bound characterizes the worst-case number of probe-variable entering steps needed to eliminate degeneracy and reach the LGM optimal basis, providing a theoretical guarantee on the global convergence efficiency of the LGM. Taken together, Propositions 6 and 7 quantify the marginal computational overhead introduced by the LGM relative to the SCM. The first shows that the additional per-iteration cost remains polynomially manageable, while the second provides a bound on the number of extra iterations required to reach the LGM optimal basis. These findings demonstrate that the improved dual solution quality delivered by the LGM can be achieved with limited and well-characterized computational effort.

7 Case study and computational results

In this section, we demonstrate the performance of the proposed LGM/WLGM with probe variables through two representative applications: the airline crew scheduling problem and the cutting stock problem. Our focus is on the convergence of LP relaxations rather than solving the integer programs (IP). Furthermore, since a high-quality LP solution provides a solid foundation for the IP, integer programming approaches, such as branch-and-price, are also expected to benefit from the LGM/WLGM framework.

7.1 Airline crew scheduling problem

In the airline crew scheduling problem, sequences of flights, referred to as *rosters*, are assigned to crews such that each flight is operated by at least one qualified crew. Let R denote the set of feasible rosters, indexed by r, where each roster r incurs a cost c_r reflecting practical operational considerations. In line with common practice, the roster cost is decomposed into four components: duty time cost, deadhead cost, layover cost, and duty-day cost. Formally, for a roster $r \in R$, c_r is calculated as

$$c_r = c^{\text{hour}} \sum_{l\in L_r} \left(t_l^{\text{arr}} - t_l^{\text{dep}}\right) + c^{\text{ddh}}\, n_r^{\text{ddh}} + c^{\text{lay}}\, n_r^{\text{lay}} + c^{\text{day}}\, n_r^{\text{day}}, \tag{32}$$

where L_r is the set of duties contained in roster r, and each duty $l \in L_r$ may include one or more flights or deadhead segments. The departure time of the first segment and the arrival time of the last flight within duty l are denoted by t_l^{dep} and t_l^{arr}, respectively, so that the first term represents the total duty time multiplied by the per-hour cost c^{hour}. The second term accounts for the total number of deadhead segments n_r^{ddh}, with c^{ddh} as the associated cost per deadhead. The third term captures layover expenses, computed as the number of overnight stays n_r^{lay} multiplied by the per-night cost c^{lay}. Finally, the duty-day cost reflects fixed payments for each calendar day on which the crew is assigned duties, calculated as c^{day} times n_r^{day}. Together, these four components provide a comprehensive measure of the operational costs associated with a crew roster.

Let F be the set of planned flights indexed by f, and M the set of crew members indexed by m. For each crew member m, denote R_m as the subset of feasible rosters available for crew m. The binary coefficient a_{fr} equals 1 if flight f is included in roster r, and 0 otherwise. The decision variable x_r indicates whether roster r is selected, while the artificial variable u_f represents whether flight f is canceled. In the LP relaxation, both x_r and u_f are treated as non-negative continuous variables. With these definitions, the airline crew scheduling problem can be formulated as a set covering model with side constraints, and its LP relaxation is given by:

$$\text{(ACSM)} \quad \min \sum_{r\in R} c_r x_r + \sum_{f\in F} \theta u_f \tag{33}$$

$$\text{s.t.} \quad \sum_{r\in R} a_{fr} x_r + u_f \geq 1, \qquad \forall f \in F, \tag{34}$$

$$\sum_{r\in R_m} x_r \leq 1, \qquad \forall m \in M, \tag{35}$$

$$x_r \geq 0, \qquad \forall r \in R, \tag{36}$$

$$u_f \geq 0, \qquad \forall f \in F. \tag{37}$$

The objective function (33) minimizes the total cost of the resulting crew schedule. The flight covering constraints (34) ensure that each flight is either covered by at least one selected roster or canceled, while the crew assignment constraints (35) ensure that each crew member is assigned to at most one roster. Constraints (36)–(37) define the feasible range of the decision variables. The ACSM is solved using a column generation framework, with the detailed algorithmic procedure provided in Appendix 9.3.

7.1.1 Test cases

To evaluate the computational efficiency of the proposed LGM, a set of test instances was generated based on real-world operational data from a major Chinese airline. All instances share a planning horizon of one week. Specifically, Cases 1, 3, 5, 7, and 9 are derived from the same week that contains public holidays, Cases 2, 4, 6, 8, and 10 are derived from a regular operational week. The detailed characteristics of these test cases are summarized in Table 1, including the numbers of aircraft, flights, deadheads, layover stations, crew members, and pre-assigned ground activities (e.g., training or vacation) for specific crew members. It is also noteworthy that all flights are eligible to serve as deadheads for crew repositioning.

Table 1: Characteristics of test cases

Case	# Aircraft	# Flight	# Deadhead	# Layover Station	# Crew	# Ground Activity
Case1	24	767	1,188	49	142	62
Case2	24	722	1,188	49	142	86
Case3	31	962	1,188	49	176	78
Case4	31	932	1,188	49	176	102
Case5	42	1,212	1,188	49	228	218
Case6	42	1,185	1,188	49	228	234
Case7	55	1,606	1,188	49	305	396
Case8	55	1,521	1,188	49	305	370
Case9	64	1,841	1,188	49	356	456
Case10	64	1,799	1,188	49	356	388

7.1.2 Computational and parameters settings

All test problems were solved on a computer with a 3.60 GHz Intel Xeon Gold 6544Y CPU, 128 GB RAM, and 64 threads running Windows Server 2019. The models and algorithms were implemented in C++ using Visual Studio 2022, and all LP models were solved using the commercial solver CPLEX 22.1.0 with default settings [IBM Corporation, 2022]. All subproblems were solved in parallel using 64 threads. The maximum number of variables added to the master problem at each iteration is set to $3|M|$, i.e., at most three new beneficial rosters per crew. The total runtime for all cases is limited to 10 hours, and the algorithm terminates either when the optimal solution is obtained or upon reaching the predefined time limit. The relevant cost parameters are summarized in Table 2.

Table 2: Parameter settings for roster cost calculation

Parameter	**Value**	**Description**
c^{hour}	300	Unit cost per duty hour
c^{ddh}	500	Unit cost per deadhead
c^{lay}	1,000	Unit cost per layover
c^{day}	2,000	Fixed cost per calendar duty day

7.1.3 Effectiveness of the LGM

In this section, we evaluate the computational efficiency of the proposed LGM. The baseline is the ACSM, while the proposed LG-ACSM extends each flight covering constraint (34) by adding $K = 10$ probe variables (with the first probe variable, which has a cost of 0, omitted for simplicity). The comparison between the ACSM and the LG-ACSM is presented in Table 3. For each case, the table reports the number of iterations (# Iter), the number of constraints (# Cons), the numbers of roster and probe variables (# Roster Var and # Probe Var), the LP optimal objective value (LP Obj), the total computation time of the master problems (MP Time), the total computation time of the pricing subproblems (SP Time), and the overall time required to obtain the LP optimal solution (Time). In particular, we compute the optimal total roster cost (Opt LP Roster Cost) by using all the roster variables in the final LG-ACSM and

setting the upper bounds of the probe variables to 0, which serves as the LP optimal objective value of the ACSM. All computational times are measured in seconds. The "Saving # Iter" and "Saving Time" metrics are calculated as (ACSM - LG-ACSM)/ (ACSM) $\times 100\%$ based on the number of iterations and total computation time, respectively.

Table 3: Comparison between the ACSM and the LG-ACSM

	ACSM							LG-ACSM								Saving	
Case	# Iter	# Cons	# Roster Var	LP Obj	MP Time	SP Time	Time	# Iter	# Cons	# Roster Var	# Probe Var	LP Obj	Opt LP Roster Cost	MP Time	SP Time	Time	# Iter (%)	Time (%)
Case1	4,149	971	1,253,209	1,565,250	2,099	31,511	34,363	151	971	47,679	8,290	1,564,870	1,565,250	67	459	554	96.36	98.39
Case2	3,813	950	1,470,402	1,603,750	3,101	32,239	36,000	157	950	58,510	8,080	1,602,660	1,603,040	91	599	711	>95.88	>98.03
Case3	2,345	1,216	1,049,499	1,948,380	2,369	33,197	36,000	197	1,216	84,513	10,400	1,939,610	1,940,050	153	1,628	1,814	>91.60	>94.96
Case4	2,317	1,210	989,875	1,986,120	2,546	33,047	36,000	190	1,210	73,070	10,340	1,981,050	1,981,510	147	1,458	1,633	>91.80	>95.46
Case5	1,005	1,658	692,534	2,695,770	3,820	32,014	36,000	210	1,658	133,565	14,300	2,680,510	2,681,200	497	3,159	3,690	>79.10	>89.75
Case6	910	1,647	629,913	2,632,310	4,033	31,773	36,000	204	1,647	128,080	14,190	2,616,820	2,617,500	488	3,148	3,671	>77.58	>89.80
Case7	1,135	2,307	1,039,611	3,619,420	5,613	30,150	36,000	280	2,307	239,537	20,020	3,587,520	3,588,450	1,055	4,005	5,109	>75.33	>85.81
Case8	724	2,196	622,735	3,562,490	5,807	30,071	36,000	237	2,196	192,055	18,910	3,534,190	3,535,090	1,250	5,483	6,769	>67.27	>81.20
Case9	588	2,653	654,941	4,060,480	5,875	30,044	36,000	314	2,653	335,065	22,970	3,975,490	3,976,420	2,181	8,083	10,333	>46.60	>71.30
Case10	566	2,543	612,006	4,256,750	6,645	29,247	36,000	270	2,543	288,185	21,870	4,159,660	4,160,840	2,209	8,050	10,328	>52.30	>71.31
Avg.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	>77.38	>87.60

As shown in Table 3, only the first case achieves an LP optimal solution within the ten-hour time limit using the ACSM, highlighting significant degeneracy in the remaining instances. Due to the time constraint, larger cases even exhibit fewer iterations but are farther from the optimal solution. In contrast, the LG-ACSM significantly accelerates the solution process without sacrificing optimality, reducing the number of iterations and total computation time by 77.38% and 87.60% on average, respectively. Notably, we expect the actual savings to be much greater, given that the ACSM does not provide an optimal solution within the time limit, which compromises the fairness of the comparison. These results demonstrate that probe variables effectively select a unique dual solution from the multiple optima caused by degeneracy, thereby stabilizing the dual values and facilitating the convergence of the column generation procedure.

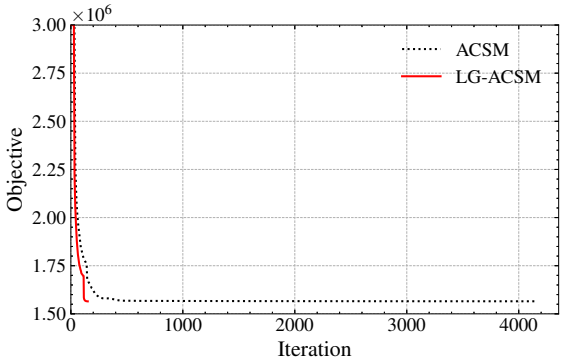
To illustrate the convergence behavior of the ACSM and the LG-ACSM, we plotted the objective values over the iterations, as shown in Figure 4. It can be seen that the long-tail effect is significantly less pronounced for the LG-ACSM than for the ACSM, indicating that the inclusion of probe variables substantially mitigates this effect. Furthermore, across all cases, a sharp drop can be observed at a certain iteration. This phenomenon arises from a heuristic strategy adopted in the column generation process: during the early stages, a simplified dominance rule was applied in the multi-label shortest path algorithm to accelerate computation. When the objective value began to decrease slowly, the full dominance rule was reinstated to ensure the optimality of the final solution.

7.1.4 Comparison with interior point method

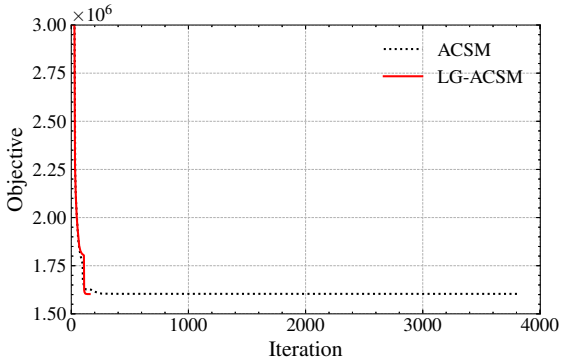
In this section, we compare the proposed LGM framework with a classical interior point method. Specifically, the ACSM is solved using CPLEX's built-in Barrier method without crossover, which directly produces an interior point dual solution without converting it into a basic feasible solution, providing a natural baseline for comparison with the LG-ACSM. In practice, this is achieved by setting `IloCplex::RootAlg = IloCplex::Algorithm::Barrier` and `IloCplex::Param::SolutionType = 2` [IBM Corporation, 2022]. The results are summarized in Table 4. For each instance, the table reports the number of iterations (# Iter), the numbers of constraints (# Cons), the number of roster and probe variables (# Roster Var and # Probe Var), the LP optimal objective value (LP Obj), the optimal roster cost (Opt LP Roster Cost), the total computation time of the master problems (MP Time), the total computation time of the pricing subproblems (SP Time), and the overall time to reach the LP optimal solution (Time), all measured in seconds. The metrics "Saving # Iter" and "Saving Time" denote the relative reductions in iterations and total computation time, respectively, computed as $(\text{ACSM} - \text{LG-ACSM})/\text{ACSM} \times 100\%$.

Table 4: Comparison with interior point method

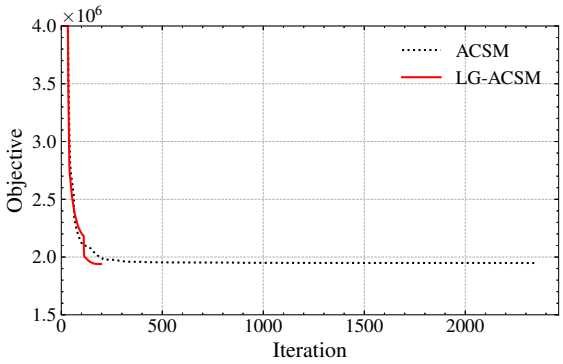
	ACSM + barrier							LG-ACSM								Saving	
Case	# Iter	# Cons	# Roster Var	LP Obj	MP Time	SP Time	Time	# Iter	# Cons	# Roster Var	# Probe Var	LP Obj	Opt LP Roster Cost	MP Time	SP Time	Time	# Iter (%)	Time (%)
Case1	267	971	87,043	1,565,250	67	1,337	1,459	151	971	47,679	8,290	1,564,870	1,565,250	67	459	554	43.45	62.03
Case2	226	950	84,718	1,603,040	78	1,173	1,286	157	950	58,510	8,080	1,602,660	1,603,040	91	599	711	30.53	44.71
Case3	704	1,216	299,886	1,940,050	357	9,132	9,623	197	1,216	84,513	10,400	1,939,610	1,940,050	153	1,628	1,814	72.02	81.15
Case4	251	1,210	96,716	1,981,510	113	2,378	2,529	190	1,210	73,070	10,340	1,981,050	1,981,510	147	1,458	1,633	24.30	35.43
Case5	975	1,658	649,255	2,688,690	1,356	34,465	36,000	210	1,658	133,565	14,300	2,680,510	2,681,200	497	3,159	3,690	>78.46	>89.75
Case6	983	1,647	657,174	2,624,710	1,367	34,432	36,000	204	1,647	128,080	14,190	2,616,820	2,617,500	488	3,148	3,671	>79.25	>89.80
Case7	1,154	2,307	978,737	3,603,860	1,959	33,821	36,000	280	2,307	239,537	20,020	3,587,520	3,588,450	1,055	4,005	5,109	>75.74	>85.81
Case8	765	2,196	613,436	3,549,880	1,768	34,072	36,000	237	2,196	192,055	18,910	3,534,190	3,535,090	1,250	5,483	6,769	>69.02	>81.20
Case9	599	2,653	624,404	4,060,480	1,804	34,037	36,000	314	2,653	335,065	22,970	3,975,490	3,976,420	2,181	8,083	10,333	>47.58	>71.30
Case10	480	2,543	486,095	4,256,750	1,743	34,156	36,000	270	2,543	288,185	21,870	4,159,660	4,160,840	2,209	8,050	10,328	>43.75	>71.31
Avg.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	>56.41	>71.25



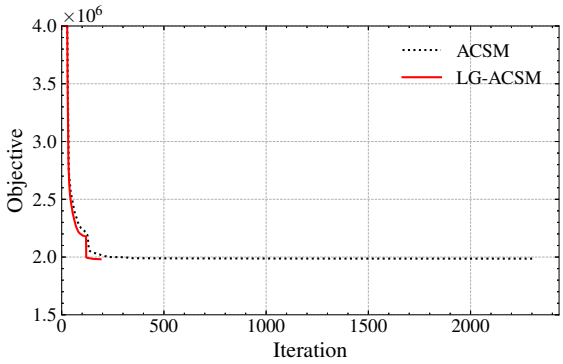
(a) Case 1



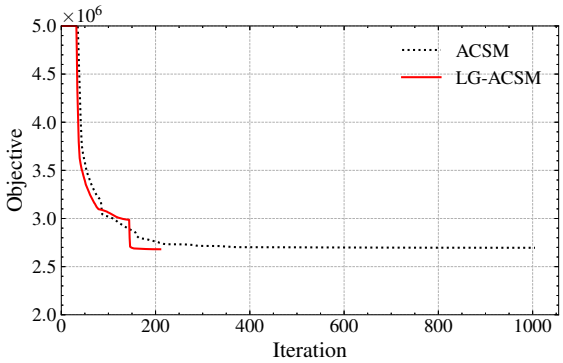
(b) Case 2



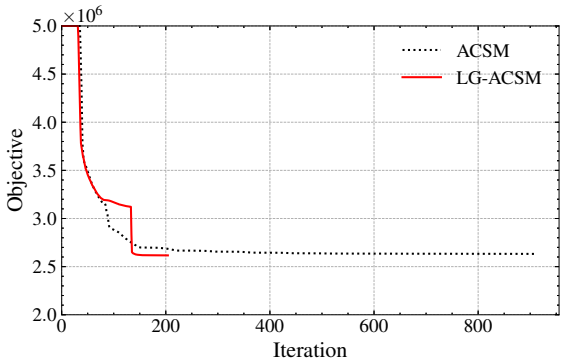
(c) Case 3



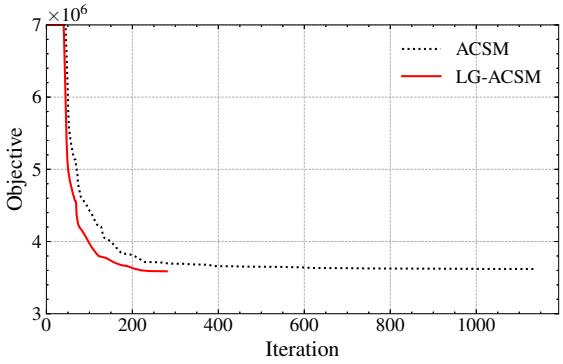
(d) Case 4



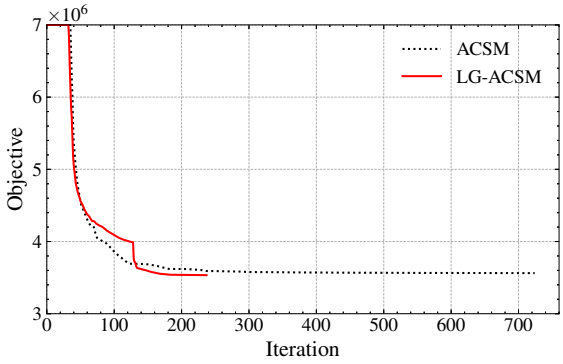
(e) Case 5



(f) Case 6

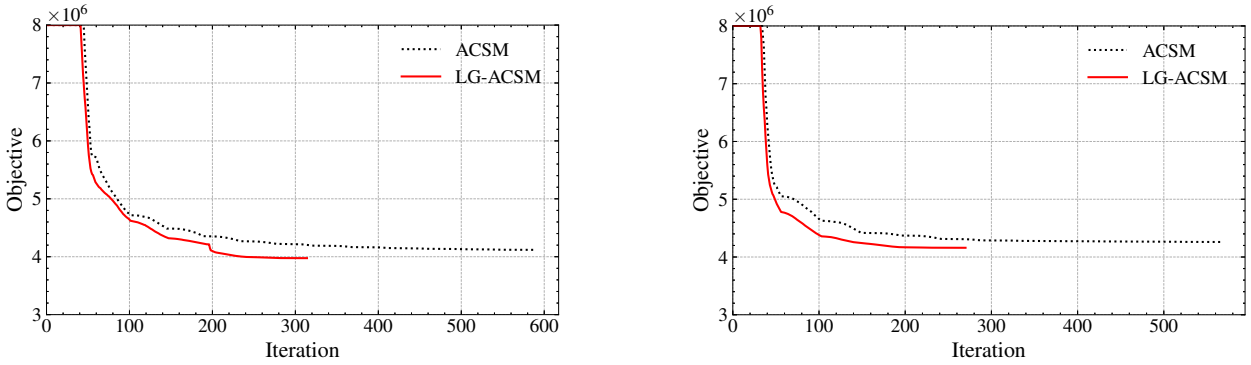


(g) Case 7



(h) Case 8

Figure 4: Convergence of objectives for the ACSM and the LG-ACSM



(i) Case 9 (j) Case 10

Figure 4: Convergence of objectives for the ACSM and the LG-ACSM

As shown in Table 4, the LG-ACSM consistently outperforms the classical interior point method (Barrier without crossover) across all test instances. On average, the LG-ACSM reduces the number of iterations by 56.41% and the total computation time by 71.25%. Notably, for Cases 5 to 10, the solutions obtained by the ACSM with the interior point method were not optimal within the given time limit, suggesting that the actual performance improvements could be even greater than those reported. While the interior point method mitigates degeneracy by avoiding traversal of the extreme points in dual space, its lack of control over the dual solutions can still lead to inefficiencies. In contrast, the LG-ACSM ensures a unique dual solution, stabilizing the process and accelerating convergence. These results confirm the practical advantage of the proposed LGM framework over standard interior point approaches.

7.1.5 Sensitivity analysis of the number of probe variables

In this section, we investigate the sensitivity of the LG-ACSM with respect to the parameter K. As defined in Section 3, the cost interval between two consecutive probe variables is given by θ/K. Intuitively, the narrower the interval, the smaller the range of dual oscillations. In this test, we conduct experiments with four different values of K, corresponding to probe cost intervals of 100, 250, 500, and 1000. For each setting, we evaluate the computational performance across all test cases. The detailed results are presented in Table 5, which reports the number of iterations (# Iter), the number of constraints (# Cons), the numbers of roster and probe variables (# Roster Var and # Probe Var), the LP optimal objective value (LP Obj), the total computation time of the master problems (MP Time), the total computation time of the pricing subproblems (SP Time), and the overall time required to obtain the LP optimal solution (Time).

As shown in Table 5, the sensitivity analysis confirms that increasing the number of probe variables per constraint and using a smaller cost interval, which narrows the range of allowable dual oscillations, generally accelerates convergence and reduces the number of iterations needed to reach the LP optimal solution. Remarkably, even adding only five probe variables per constraint yields a clear improvement in computational efficiency compared with the baseline ACSM. This demonstrates that even a relatively small number of probe variables can substantially stabilize dual values, improve the column generation process, and accelerate convergence.

7.2 Cutting stock problem

In the cutting stock problem, given a kind of rolls with width W, the objective is to determine a set of cutting patterns that satisfy the demands of different item types. Let I denote the set of items indexed by i, each with demand d_i and width w_i. Let S be the set of feasible cutting patterns. Each pattern $s \in S$ is represented by a vector $(a_{1s}, \ldots, a_{|I|s})$, where a_{is} denotes the number of items of type i included in pattern s. A pattern is feasible if it satisfies the width constraint $\sum_{i\in I} a_{is}w_i \leq W, a_{is} \in \mathbb{N}, \forall i \in I$. The cost of pattern s is defined as $c_s = c_0 + c_w \left(W - \sum_{i\in I} a_{is}w_i\right)$, where c_0 is the fixed cost associated with using a pattern, and c_w denotes the cost per unit of unused width (waste). Let x_s be the decision variable indicating whether pattern s is selected. In the LP relaxation, x_s is a non-negative continuous variable, while u_i represents the unsatisfied demand for item i. With these definitions, the LP relaxation is

Table 5: Sensitivity analysis of the number of probe variables

Case	K	θ/K	# Iter	# Cons	# Roster Var	# Probe Var	LP obj	MP Time	SP Time	Time
Case1	5	1,000	155	971	49,038	4,145	1,565,100	62	515	604
	10	500	151	971	47,679	8,290	1,564,870	67	459	554
	20	250	162	971	49,666	16,580	1,564,420	72	524	622
	50	100	160	971	57,189	41,450	1,563,060	82	516	629
Case2	5	1,000	167	950	59,329	4,040	1,602,880	75	703	804
	10	500	157	950	58,510	8,080	1,602,660	91	599	711
	20	250	134	950	47,594	16,160	1,602,190	68	480	567
	50	100	139	950	48,234	40,400	1,600,810	77	569	665
Case3	5	1,000	206	1,216	86,717	5,200	1,939,880	154	1,701	1,879
	10	500	197	1,216	84,513	10,400	1,939,610	153	1,628	1,814
	20	250	201	1,216	84,991	20,800	1,939,080	146	1,178	1,359
	50	100	190	1,216	79,211	52,000	1,937,470	185	1,524	1,739
Case4	5	1,000	193	1,210	73,188	5,170	1,981,330	133	1,533	1,691
	10	500	190	1,210	73,070	10,340	1,981,050	147	1,458	1,633
	20	250	179	1,210	70,125	20,680	1,980,500	145	1,280	1,444
	50	100	182	1,210	68,985	51,700	1,978,840	190	1,364	1,574
Case5	5	1,000	212	1,658	134,435	7,150	2,680,910	461	3,307	3,799
	10	500	210	1,658	133,565	14,300	2,680,510	497	3,159	3,690
	20	250	203	1,658	129,714	28,600	2,679,690	564	2,996	3,595
	50	100	192	1,658	122,769	71,500	2,677,230	532	2,881	3,452
Case6	5	1,000	208	1,647	132,420	7,095	2,617,210	429	3,230	3,695
	10	500	204	1,647	128,080	14,190	2,616,820	488	3,148	3,671
	20	250	207	1,647	131,858	28,380	2,616,020	577	3,005	3,618
	50	100	196	1,647	126,104	70,950	2,613,630	540	2,714	3,291
Case7	5	1,000	308	2,307	280,594	10,010	3,588,070	1,258	4,979	6,311
	10	500	280	2,307	239,537	20,020	3,587,520	1,055	4,005	5,109
	20	250	270	2,307	236,026	40,040	3,586,420	1,230	3,882	5,175
	50	100	269	2,307	234,057	100,100	3,583,100	1,269	4,028	5,349
Case8	5	1,000	235	2,196	190,797	9,455	3,534,720	1,144	5,648	6,839
	10	500	237	2,196	192,055	18,910	3,534,190	1,250	5,483	6,769
	20	250	240	2,196	189,046	37,820	3,533,120	1,569	5,804	7,411
	50	100	218	2,196	179,941	94,550	3,529,900	1,400	4,878	6,322
Case9	5	1,000	351	2,653	391,263	11,485	3,976,040	2,482	10,589	13,159
	10	500	314	2,653	335,065	22,970	3,975,490	2,181	8,083	10,333
	20	250	316	2,653	338,904	45,940	3,974,360	2,507	8,435	11,019
	50	100	304	2,653	330,391	114,850	3,970,990	2,052	7,819	9,949
Case10	5	1,000	327	2,543	342,071	10,935	4,160,340	2,748	11,645	14,460
	10	500	270	2,543	288,185	21,870	4,159,660	2,209	8,050	10,328
	20	250	264	2,543	250,489	43,740	4,158,290	2,330	7,817	10,211
	50	100	264	2,543	250,350	109,350	4,154,160	2,181	8,028	10,265

formulated as follows:

$$\text{(CSM)} \quad \min \sum_{s \in S} c_s x_s + \sum_{i \in I} \theta_i u_i \tag{38}$$

$$\text{s.t.} \sum_{s \in S} a_{is} x_s + u_i \geq d_i, \qquad \forall i \in I, \tag{39}$$

$$x_s \geq 0, \qquad \forall s \in S, \tag{40}$$

$$u_i \geq 0, \qquad \forall i \in I. \tag{41}$$

The objective function (38) minimizes the total cost of the selected cutting patterns along with the penalties for unmet demands, while the demand covering constraints (39) ensure that the demand for each item is satisfied. Constraints (40)–(41) specify the feasible range of the decision variables. The CSM is solved using a column generation approach, with the detailed algorithmic procedure provided in Appendix 9.4.

7.2.1 Test cases

To evaluate the performance of the proposed WLGM framework, we conduct computational experiments on the classical cutting stock problem. The instance generation procedure follows that described in Valério de Carvalho

[2005]. Specifically, three roll widths are considered: 100, 120, and 150. For each roll width, two instance sizes are generated, containing 200 and 500 item types, respectively. The item widths are generated using two distributions: the first is a uniform distribution over the interval $[10, \text{roll width} - 10)$, and the second is a multi-peak distribution, where 5–6 random peak values are chosen within $[10, 90)$, each corresponding to a randomly determined quantity of items, resulting in clusters of item widths around these peaks. For each configuration (distribution, roll width, item size), 20 random instances are generated. In our tests, each cutting pattern has a fixed cost $c_0 = 50$, a unit waste cost $c_w = 0.5$, and a demand $d_i = 1$ for each item i. The computational settings follow those described in Section 7.1.2, with only the single most beneficial variable (if it exists) added in each iteration.

7.2.2 Effectiveness of the WLG-CSM

We compare the baseline CSM, given in Eq.(38)-(41), to the CSM with width-weighted probe variables (WLG-CSM). We consider that the cost of covering an item is roughly proportional to its width. Accordingly, we define $\eta_i = w_i$ to estimate the relative importance of each item's dual value. Note that, in contrast to the theoretical definition of the weighted dual optimal direction, we omit the normalization denominator $\sum_{i \in I} w_i$ here. This simplification allows us to directly control the absolute penalty for uncovering each item, setting $\theta_i = 100 \times \eta_i = 100 \times w_i$. In the WLG-CSM, the parameter K is set to $200 \times w_i$ for each item i, such that the cost of the k-th probe variable in constraint i is given by $k \times \frac{\theta_i}{K} = 0.5 \times k$. To improve computational efficiency, for each item i, we compute the target dual $\tilde{\pi} i = \frac{w_i}{\text{roll width}} \times c_0$. Probe variables are then added to the WLG-CSM only when their corresponding cost c_{ik} lies within the interval $[\tilde{\pi}_i - 3, \tilde{\pi}_i + 3]$.

Table 6 summarizes the computational results for the CSM and the WLG-CSM. Specifically, the results presented are averages over 20 random instances for each configuration, showing the number of iterations (# Iter), the number of degenerate iterations (# Dege-iter), and the total running time required to reach the LP optimal solutions, measured in seconds (Time). The definition of degenerate iteration follows Valério de Carvalho [2005]: a degenerate iteration occurs when the objective value does not improve after inserting a new column into the restricted master problem. Additionally, the "Saving # Iter" and "Saving # Dege-iter" metrics are calculated as $(\text{CSM} - \text{WLG-CSM})/(\text{CSM}) \times 100\%$, based on the reductions in the number of iterations and degenerate iterations, respectively.

Table 6: Comparison between the CSM and the WLG-CSM

Distribution	Roll Width	# Item	Model	# Iter	# Dege-iter	Time	Saving # Iter (%)	Saving # Dege-iter (%)
Normal	100	200	CSM	347.80	136.35	0.09	-	-
			WLG-CSM	240.55	1.50	0.25	30.84	98.90
	100	500	CSM	906.00	376.25	0.42	-	-
			WLG-CSM	601.30	2.75	2.36	33.63	99.27
	120	200	CSM	346.10	133.30	0.10	-	-
			WLG-CSM	236.55	1.50	0.23	31.65	98.87
	120	500	CSM	916.35	337.40	0.49	-	-
			WLG-CSM	624.30	2.75	2.49	31.87	99.18
	150	200	CSM	356.35	117.00	0.12	-	-
			WLG-CSM	257.70	2.75	0.32	27.68	97.65
	150	500	CSM	898.60	325.15	0.48	-	-
			WLG-CSM	619.65	3.00	2.49	31.04	99.08
Multi-peak	100	200	CSM	1,153.50	954.70	0.32	-	-
			WLG-CSM	253.80	13.50	0.30	78.00	98.59
	100	500	CSM	5,019.80	4,535.90	10.86	-	-
			WLG-CSM	1,169.90	35.20	4.80	76.69	99.22
	120	200	CSM	950.80	741.70	0.25	-	-
			WLG-CSM	246.10	15.00	0.30	74.12	97.98
	120	500	CSM	5,140.50	4,643.45	10.63	-	-
			WLG-CSM	953.95	42.15	3.91	81.44	99.09
	150	200	CSM	1,258.00	1,068.45	0.36	-	-
			WLG-CSM	264.65	20.05	0.33	78.96	98.12
	150	500	CSM	4,537.70	3,848.80	11.30	-	-
			WLG-CSM	920.55	45.30	4.09	79.71	98.82
Avg.			CSM	1,819.29	1,434.87	2.94	-	-
			WLG-CSM	532.42	15.45	1.82	70.73	98.92

As shown in Table 6, the WLG-CSM significantly reduces the total number of iterations compared to the classical CSM. On average, the WLG-CSM achieves a substantial 70.73% reduction in iterations. This improvement is due to the proportional control of each item's contribution, which helps generate dual solutions closer to the ideal center,

thereby accelerating convergence more effectively. The impact on degenerate iterations is even more pronounced: the WLG-CSM reduces them by 98.92%, demonstrating that the introduction of probe variables effectively mitigates degeneracy. Notably, degeneracy is more severe in multi-peak instances, where the classical CSM experiences hundreds or even thousands of degenerate iterations. In these challenging cases, the WLG-CSM significantly reduces degenerate iterations, highlighting its robustness across both uniform and multi-peak item distributions.

8 Conclusion

In this work, we introduce the Low-Granularity Model (LGM), a surprisingly simple yet highly effective method that incorporates probe variables to achieve precise control over dual values during the solution process. We rigorously show that the dual value of each constraint is bounded within an interval defined by the probe variable with the largest cost among those with positive values and the probe variable with the smallest cost among those with zero values. Furthermore, we define the *dual optimal central distance* (DOCD) as a quantitative criterion for evaluating the quality of a given dual solution. The DOCD is measured as the squared Euclidean distance between the dual solution and the centroid of the dual optimal polytope. We prove that, for a center-symmetric dual optimal polytope with equal weights across all dimensions, the LGM attains the dual solution with the minimal DOCD among all dual optima corresponding to a degenerate primal solution. We then extend the LGM to general linear programs through the Weighted LGM (WLGM) framework. When the relative ratios among dual optimal values are known, the WLGM can deterministically guide the search toward a unique dual optimum with the minimal DOCD. This property provides a theoretically grounded approach to mitigating the adverse effects of degeneracy and, in practice, yields a more realistic dual representation that better reflects the true shadow price of each constraint. Computational experiments on airline crew scheduling and cutting stock problems confirm that our methods substantially accelerate convergence while maintaining optimality.

Building on these foundational results, our proposed approach has the potential for broad impact across the optimization literature. Tens of thousands of studies on column generation, Dantzig-Wolfe decomposition, branch-and-price, and Benders decomposition are affected by slow convergence due to degeneracy. By providing stabilized and high-quality unique dual solutions, our method can significantly mitigate, or even entirely eliminate, the negative effects of degeneracy, benefiting a wide range of optimization research involving dual-guided decomposition methods.

References

George B Dantzig, Alex Orden, and Philip Wolfe. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific Journal of Mathematics*, 5(2):183–195, 1955.

James E Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

Paul C Gilmore and Ralph E Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859, 1961.

Bengt Aspvall and Richard E Stone. Khachiyan's linear programming algorithm. *Journal of Algorithms*, 1(1):1–13, 1980.

Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pages 302–311, 1984.

Lord Kelvin. Nineteenth century clouds over the dynamical theory of heat and light. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(7):1–40, 1901.

AJ Hoffman. Cycling in the simplex algorithm. In Charles A Micchelli, editor, *Selected Papers of Alan J Hoffman with Commentary*. World Scientific Publishing Company, 2003.

E. M. L. Beale. Cycling in the dual simplex algorithm. *Naval Research Logistics Quarterly*, 2(4):269–275, 1955.

Mokhtar S Bazaraa, John J Jarvis, and Hanif D Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, Hoboken, New Jersey, 2011.

Marco E Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6): 1007–1023, 2005.

Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

Axel Parmentier, Rafael Martinelli, and Thibaut Vidal. Electric vehicle fleets: Scalable route and recharge scheduling through column generation. *Transportation Science*, 57(3):631–646, 2023.

Mohammed Saddoune, Guy Desaulniers, Issmail Elhallaoui, and François Soumis. Integrated airline crew pairing and crew assignment by dynamic constraint aggregation. *Transportation Science*, 46(1):39–55, 2012.

Maik Schälicke and Karl Nachtigall. Solving the real-time train dispatching problem by column generation. *Transportation Science*, 59(3):587–602, 2025.

Shivaram Subramanian and Hanif D Sherali. An effective deflected subgradient optimization scheme for implementing column generation for large-scale airline crew scheduling problems. *INFORMS Journal on Computing*, 20(4): 565–578, 2008.

Roy E Marsten, William W Hogan, and Jacob Watson Blankenship. The boxstep method for large-scale optimization. *Operations Research*, 23(3):389–405, 1975.

Issmail Elhallaoui, Daniel Villeneuve, François Soumis, and Guy Desaulniers. Dynamic aggregation of set-partitioning constraints in column generation. *Operations Research*, 53(4):632–645, 2005.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30(2):339–360, 2018.

Vašek Chvátal. *Linear programming*. Macmillan, 1983.

Abraham Charnes. Optimality and degeneracy in linear programming. *Econometrica*, 20(2):160–170, 1952.

Philip Wolfe. A technique for resolving degeneracy in linear programming. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):205–211, 1963.

David Murray Ryan and Michael Robert Osborne. On the solution of highly degenerate linear programmes. *Mathematical Programming*, 41(1):385–392, 1988.

Robert G Bland. New finite pivoting rules for the simplex method. *Mathematics of Operations Research*, 2(2):103–107, 1977.

Paul Wentges. Weighted Dantzig-Wolfe decomposition for linear mixed-integer programming. *International Transactions in Operational Research*, 4(2):151–162, 1997.

Olivier Du Merle, Daniel Villeneuve, Jacques Desrosiers, and Pierre Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1-3):229–237, 1999.

Hatem Ben Amor, Jacques Desrosiers, and Antonio Frangioni. On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167–1184, 2009.

Louis-Martin Rousseau, Michel Gendreau, and Dominique Feillet. Interior point stabilization for column generation. *Operations Research Letters*, 35(5):660–668, 2007.

Issmail Elhallaoui, Abdelmoutalib Metrane, François Soumis, and Guy Desaulniers. Multi-phase dynamic constraint aggregation for set partitioning type problems. *Mathematical Programming*, 123:345–370, 2010.

Issmail Elhallaoui, Guy Desaulniers, Abdelmoutalib Metrane, and François Soumis. Bi-dynamic constraint aggregation and subproblem reduction. *Computers & Operations Research*, 35(5):1713–1724, 2008.

Issmail Elhallaoui, Abdelmoutalib Metrane, Guy Desaulniers, and François Soumis. An improved primal simplex algorithm for degenerate linear programs. *INFORMS Journal on Computing*, 23(4):569–577, 2011.

Jacques Desrosiers, Jean Bertrand Gauthier, and Marco E Lübbecke. Row-reduced column generation for degenerate master problems. *European Journal of Operational Research*, 236(2):453–460, 2014.

Hocine Bouarab, Issmail El Hallaoui, Abdelmoutalib Metrane, and François Soumis. Dynamic constraint and variable aggregation in column generation. *European Journal of Operational Research*, 262(3):835–850, 2017.

José M Valério de Carvalho. Using extra dual cuts to accelerate column generation. *INFORMS Journal on Computing*, 17(2):175–182, 2005.

François Clautiaux, Cláudio Alves, José Valério de Carvalho, and Jürgen Rietz. New stabilization procedures for the cutting stock problem. *INFORMS Journal on Computing*, 23(4):530–545, 2011.

Cláudio Alves and JM Valério de Carvalho. A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. *Computers & Operations Research*, 35(4):1315–1328, 2008.

Hatem Ben Amor, Jacques Desrosiers, and José Manuel Valério de Carvalho. Dual-optimal inequalities for stabilized column generation. *Operations Research*, 54(3):454–463, 2006.

Timo Gschwind and Stefan Irnich. Dual inequalities for stabilized column generation revisited. *INFORMS Journal on Computing*, 28(1):175–194, 2016.

Vishnu Suresh Lokhande, Shaofei Wang, Maneesh Singh, and Julian Yarkony. Accelerating column generation via flexible dual optimal inequalities with application to entity resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1593–1602, 2020.

Naveed Haghani, Claudio Contardo, and Julian Yarkony. Smooth and flexible dual optimal inequalities. *INFORMS Journal on Optimization*, 4(1):29–44, 2022.

Luciano Costa, Claudio Contardo, Guy Desaulniers, and Julian Yarkony. Stabilized column generation via the dynamic separation of aggregated rows. *INFORMS Journal on Computing*, 34(2):1141–1156, 2022.

Siqi Guo, Fan Xiao, and Zhe Liang. Nested set-covering/packing problem: Degeneracy alleviation and dual stabilization. *Operations Research*, online first:1–28, 2025.

Marshall Fisher. Vehicle routing. *Handbooks in Operations Research and Management Science*, 8:1–33, 1995.

J-L Goffin, Alain Haurie, J-Ph Vial, and Dao Li Zhu. Using central prices in the decomposition of linear programs. *European Journal of Operational Research*, 64(3):393–409, 1993.

Ruben Kirkeby Martinson and Jørgen Tind. An interior point method in Dantzig–Wolfe decomposition. *Computers & Operations Research*, 26(12):1195–1216, 1999.

IBM Corporation. *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual*, 2022. URL `https://www.ibm.com/docs/en/icos/22.1.0?topic=optimization-cplex-users-manual`. Version 22.1.0.

9 Proofs of Statements and Additional Material

9.1 Proof of statements

Proposition 1: Let $(\boldsymbol{x}^M, \boldsymbol{u}^M, \boldsymbol{d}^M)$ denote an optimal solution of the LGM with parameter M. Then, as $M \to \infty$, the components $(\boldsymbol{x}^M, \boldsymbol{u}^M)$ converge to an optimal solution of the original SCM. Conversely, given any optimal solution $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ to the SCM, there exists a corresponding optimal solution $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{d}^*)$ of the LGM such that $\boldsymbol{d}^* \to 0$ as $M \to \infty$.

Proof. Proof Denote by Δ_{LGM} and Δ_{SCM} the feasible regions of the LGM and the SCM, respectively. Let $proj_{\mathbb{R}^{|P|+|J|}}(\Delta_{LGM})$ be the orthogonal projection of Δ_{LGM} onto the subspace $\mathbb{R}^{|P|+|J|}$.

First, given any feasible solution $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}) \in \Delta_{SCM}$, choosing $\tilde{\boldsymbol{d}} = \mathbf{0}$ yields a feasible solution $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{d}}) \in \Delta_{LGM}$. Thus,

$$\Delta_{SCM} \subseteq proj_{\mathbb{R}^{|P|+|J|}}(\Delta_{LGM}). \tag{A.1}$$

Conversely, given a feasible solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{d}}) \in \Delta_{LGM}$, the additional contribution of the probe variables to the set covering constraint for each j is bounded by

$$\sum_{k=0}^{K} \hat{d}_{jk} \leq (K+1)\epsilon = \frac{K+1}{MK}, \tag{A.2}$$

where K is a fixed and given constant representing the number of probe variables. Hence, for an arbitrarily small δ, we can always find a sufficiently large $M \geq \frac{K+1}{K\delta}$ so $\sum_{k=0}^{K} \hat{d}_{jk} \leq \delta$, which implies $\lim_{M\to\infty} \sum_{p\in P} a_{jp}\hat{x}_p + \hat{u}_j + \sum_{k=0}^{K} \hat{d}_{jk} = 1, \forall j \in J$. Therefore, we have

$$\lim_{M\to\infty} (proj_{\mathbb{R}^{|P|+|J|}}(\Delta_{LGM})) \subseteq \Delta_{SCM}. \tag{A.3}$$

From Eq.(A.1) and Eq.(A.3), we can obtain

$$\lim_{M\to\infty} (proj_{\mathbb{R}^{|P|+|J|}}(\Delta_{LGM})) = \Delta_{SCM}. \tag{A.4}$$

Next, consider the objective function. In the LGM, the additional contributions from all probe variables are bounded by

$$\sum_{j\in J}\sum_{k=0}^{K} c_{jk} d_{jk}^M \leq \sum_{j\in J}\sum_{k=0}^{K} \frac{k\theta}{K} \cdot \frac{1}{MK} = \sum_{j\in J} \frac{\theta}{MK^2} \sum_{k=0}^{K} k = \sum_{j\in J} \frac{\theta(K+1)}{2MK} = \frac{\theta(K+1)|J|}{2MK}. \tag{A.5}$$

Thus, for an arbitrarily small δ, we can always find a sufficiently large $M \geq \frac{\theta(K+1)|J|}{2K\delta}$ so $\sum_{j\in J}\sum_{k=0}^{K} c_{jk}d_{jk}^{M} \leq \frac{\theta(K+1)|J|}{2MK} \leq \delta$, which implies the two objectives of the SCM and the LGM coincide:

$$\lim_{M\to\infty}\left(\sum_{p\in P} c_p x_p + \sum_{j\in J}\theta u_j + \sum_{j\in J}\sum_{k=0}^{K} c_{jk}d_{jk}\right) = \sum_{p\in P} c_p x_p + \sum_{j\in J}\theta u_j. \tag{A.6}$$

Finally, we establish the correspondence of optimal solutions:

- Let $(\boldsymbol{x}^M, \boldsymbol{u}^M, \boldsymbol{d}^M)$ be an optimal solution of the LGM. Its projection $(\boldsymbol{x}^M, \boldsymbol{u}^M)$ lies in the feasible region of the SCM, and the objective value converges to the SCM optimal value as $M \to \infty$. Hence, any limit point of $(\boldsymbol{x}^M, \boldsymbol{u}^M)$ is an optimal solution of the SCM.
- Given an optimal solution $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ of the SCM, take $\boldsymbol{d}^* = \boldsymbol{0}$. Then $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{d}^*)$ is feasible for the LGM and achieves the same objective value as in the SCM. Therefore, it is also an optimal solution of the LGM.

□

Proposition 2: Denote by Δ_{DLGM} and Δ_{DSCM} the dual feasible regions of the DLGM and the DSCM, respectively. Let $proj_{\mathbb{R}^{|J|}}(\Delta_{DLGM})$ be the orthogonal projection of Δ_{DLGM} onto the subspace $\mathbb{R}^{|J|}$, we have $proj_{\mathbb{R}^{|J|}}(\Delta_{DLGM}) \subseteq \Delta_{DSCM}$. Let $(\boldsymbol{\pi}^M, \boldsymbol{\sigma}^M)$ be a dual optimal solution of the LGM with parameter M. Then, as $M \to \infty$, the component $(\boldsymbol{\pi}^M)$ converges to a dual optimal solution of the SCM.

Proof. Proof Let $(\tilde{\boldsymbol{\pi}}, \tilde{\boldsymbol{\sigma}}) \in \Delta_{DLGM}$ be a feasible dual solution of the LGM. By definition, the components $(\tilde{\boldsymbol{\pi}})$ satisfies the dual constraints (14) and (16) in the DLGM, which coincide with the constraints defining the dual feasible region Δ_{DSCM} of the SCM. Hence, we have $\tilde{\boldsymbol{\pi}} \in \Delta_{DSCM}$, and consequently, $proj_{\mathbb{R}^{|J|}}(\Delta_{DLGM}) \subseteq \Delta_{DSCM}$.

Then, given a pair of optimal primal and dual solutions of the LGM, denoted as $(\boldsymbol{x}^M, \boldsymbol{u}^M, \boldsymbol{d}^M)$ and $(\boldsymbol{\pi}^M, \boldsymbol{\sigma}^M)$. From Proposition 1, as $M \to \infty$, $(\boldsymbol{x}^M, \boldsymbol{u}^M)$ converges to an optimal primal solution to the SCM, and we have $\lim_{M\to\infty}(\sum_{p\in P} c_p x_p^M + \sum_{j\in J}\theta u_j^M + \sum_{j\in J}\sum_{k=0}^{K} c_{jk}d_{jk}^M) = \sum_{p\in P} c_p x_p^M + \sum_{j\in J}\theta u_j^M$. Next, we are going to prove that $(\boldsymbol{\pi}^M)$ could converge to a dual optimal solution of the SCM.

From constraints (15) of the DLGM, we have $\sigma_{jk}^{-M} = c_{jk} - \pi_j^M - \sigma_{jk}^{+M}$. Since $0 \leq \pi_j^M \leq \theta$, and $\sigma_{jk}^{+M} > 0$ implies that $d_{jk} = 0$ (hence $\sigma_{jk}^{-M} = 0$), it follows that $c_{jk} - \theta \leq \sigma_{jk}^{-M} \leq 0$. Therefore, the contribution of σ_{jk}^{-M} to the dual objective (13) is bounded below by

$$\sum_{j\in J}\sum_{k=0}^{K}\epsilon\sigma_{jk}^{-M} \geq \sum_{j\in J}\sum_{k=0}^{K}\epsilon(c_{jk}-\theta) \geq \sum_{j\in J}\sum_{k=0}^{K} k\frac{\theta}{K}\frac{1}{MK} - \sum_{j\in J}\sum_{k=0}^{K}\frac{1}{MK}\theta = -\frac{\theta(K+1)|J|}{2MK}. \tag{A.7}$$

As $M \to \infty$, we have $\sum_{j\in J}\sum_{k=0}^{K}\epsilon\sigma_{jk}^{-M} \geq -\frac{\theta(K+1)|J|}{2MK} \to 0$. Additionally, since $\sum_{j\in J}\sum_{k=0}^{K}\epsilon\sigma_{jk}^{-M} \leq 0$, it follows that $\lim_{M\to\infty}\left(\sum_{j\in J}\sum_{k=0}^{K}\epsilon\sigma_{jk}^{-M}\right) = 0$. Thus, the objective of the DLGM reduces to

$$\lim_{M\to\infty}(\sum_{j\in J}\pi_j^M + \sum_{j\in J}\sum_{k=0}^{K}\epsilon\sigma_{jk}^{-M}) = \sum_{j\in J}\pi_j^M. \tag{A.8}$$

Furthermore, referring to strong duality theorem, we have

$$\sum_{p\in P} c_p x_p^M + \sum_{j\in J}\theta u_j^M + \sum_{j\in J}\sum_{k=0}^{K} c_{jk}d_{jk}^M = \sum_{j\in J}\pi_j^M + \sum_{j\in J}\sum_{k=0}^{K}\epsilon\sigma_{jk}^{-M}. \tag{A.9}$$

Thus, as $M \to \infty$, we can obtain

$$\sum_{p\in P} c_p x_p^M + \sum_{j\in J}\theta u_j^M = \sum_{j\in J}\pi_j^M, \tag{A.10}$$

which shows that $\boldsymbol{\pi}^M$ achieves the dual optimal objective of the SCM. Finally, since $proj_{\mathbb{R}^{|J|}}(\Delta_{DLGM}) \subseteq \Delta_{DSCM}$, $\boldsymbol{\pi}^M$ is a dual feasible solution of the SCM, Hence, $\boldsymbol{\pi}^M$ is a dual optimal solution of the SCM. □

Theorem 1: Given optimal primal/dual solutions to the LGM, for any job j, let k^* be the largest index such that $d_{jk^*} > 0$ and therefore $d_{jk^*+1} = 0$. Then the dual optimal value π_j for constraint j must satisfy $c_{jk^*} \leq \pi_j \leq c_{jk^*+1}$.

Proof. Proof Given an optimal solution to the LGM, we have the following cases for each of the probe variable,

- If $d_{jk} = \epsilon$, by complementary slackness we have $\sigma_{jk}^- \leq 0$ and $\sigma_{jk}^+ = 0$, then from Constraint (15), it implies $c_{jk} - \pi_j \leq 0$;
- If $0 < d_{jk} < \epsilon$, by complementary slackness we have $\sigma_{jk}^- = 0$ and $\sigma_{jk}^+ = 0$, then from Constraint (15), it implies $c_{jk} - \pi_j = 0$;
- If $d_{jk} = 0$, by complementary slackness we have $\sigma_{jk}^- = 0$ and $\sigma_{jk}^+ \geq 0$, then from Constraint (15), it implies $c_{jk} - \pi_j \geq 0$.

Therefore, we have $c_{jk^*} \leq \pi_j$ and $\pi_j \leq c_{jk^*+1}$, and hence complete the proof. □

Theorem 2: As $K \to \infty$ and $M \to \infty$, we have $\lim_{K\to\infty, M\to\infty} \sum_{j\in J} \sum_{k=0}^{K} c_{jk} d_{jk} = \frac{1}{2M\theta} \sum_{j\in J} (\pi_j)^2$, and the corresponding dual optimal solution $\boldsymbol{\pi}^*$ obtained from the LGM is unique.

Proof. Proof

According to Theorem 1, for constraint j, the largest index k_j^* such that $d_{jk} = \epsilon$ is determined as $k_j^* = \left\lfloor \frac{\pi_j K}{\theta} \right\rfloor$. Thus, $d_{jk} = \epsilon$ if $k \leq k_j^*$, and $d_{jk} = 0$ if $k > k_j^*$. Then, the contribution of the probe variables corresponding to job j to the objective function (5) can be computed as:

$$\sum_{k=0}^{K} c_{jk} d_{jk} = \sum_{k=0}^{k_j^*} c_{jk} d_{jk} + \sum_{k=k_j^*+1}^{K} c_{jk} d_{jk} = \sum_{k=0}^{k_j^*} c_{jk} \epsilon. \tag{A.11}$$

Substituting $c_{jk} = k\frac{\theta}{K}$ and $\epsilon = \frac{1}{MK}$ into Eq.(A.11), we have

$$\begin{aligned} \sum_{k=0}^{K} c_{jk} d_{jk} = \sum_{k=0}^{k_j^*} c_{jk} \epsilon = \sum_{k=0}^{k_j^*} (k\frac{\theta}{K}) \frac{1}{MK} = \frac{\theta}{MK^2} \sum_{k=0}^{k_j^*} k \\ = \frac{\theta}{MK^2} \times \frac{1}{2} \left\lfloor \frac{\pi_j K}{\theta} \right\rfloor (\left\lfloor \frac{\pi_j K}{\theta} \right\rfloor + 1) = \frac{\theta}{2MK^2} \left\lfloor \frac{\pi_j K}{\theta} \right\rfloor \left\lceil \frac{\pi_j K}{\theta} \right\rceil. \end{aligned} \tag{A.12}$$

Take the limit $K \to \infty$, the sum in Eq.(A.12) converges to

$$\lim_{K\to\infty} \sum_{k=0}^{K} c_{jk} d_{jk} = \lim_{K\to\infty} \frac{\theta}{2MK^2} \left\lfloor \frac{\pi_j K}{\theta} \right\rfloor \left\lceil \frac{\pi_j K}{\theta} \right\rceil = \frac{\theta}{2MK^2} \frac{\pi_j K}{\theta} \frac{\pi_j K}{\theta} = \frac{(\pi_j)^2}{2M\theta}. \tag{A.13}$$

Because θ is constant, Eq.(A.13) shows that, in the limit $K \to \infty$, minimizing $\sum_{j\in J} \sum_{k=0}^{K} c_{jk} d_{jk}$ is equivalent to minimizing $\sum_{j\in J} (\pi_j)^2$.

Furthermore, it is easy to see that the set of all dual optimal solutions of the SCM is a nonempty closed convex set. The function $f(\boldsymbol{\pi}) = \sum_{j\in J} (\pi_j)^2$ is strictly convex, and thus admits a unique minimizer over this convex set. Since the LGM selects precisely this minimizer in the limit $K \to \infty$ and $M \to \infty$, the $\boldsymbol{\pi}^*$-component of the dual optimal solution provided by the LGM is unique. □

Proposition 3: Given the dual optimal direction $\boldsymbol{\eta}$, defined with the dual optimal center $\boldsymbol{\pi}^*$ for the SCM, the dual optimal solution to $\min_{\boldsymbol{\pi}\in\mathcal{D}^*} \sum_{j\in J} \frac{(\pi_j)^2}{\eta_j}$ coincides with the solution to $\min_{\boldsymbol{\pi}\in\mathcal{D}^*} v(\boldsymbol{\pi})$, which minimizes the DOCD, and is precisely the dual optimal center $\boldsymbol{\pi}^*$.

Proof. Proof For any $\boldsymbol{\pi} \in \mathcal{D}^*$, we have

$$v(\boldsymbol{\pi}) = \sum_{j \in J} (\pi_j - \pi_j^*)^2 = \sum_{j \in J} \pi_j^2 - 2 \sum_{j \in J} \pi_j \pi_j^* + \sum_{j \in J} (\pi_j^*)^2. \tag{A.14}$$

Since the last term is constant with respect to $\boldsymbol{\pi}$, minimizing $v(\boldsymbol{\pi})$ is equivalent to minimizing $\sum_j \pi_j^2 - 2 \sum_j \pi_j \pi_j^*$. By the definition of η_j, we have $\pi_j^* = \eta_j \sum_{j' \in J} \pi_{j'}^* \equiv \eta_j z^*$, where z^* denotes the optimal objective value of the SCM and $z^* = \sum_{j \in J} \pi_j^*$. By substituting π_j^*, the objective becomes $\sum_j \pi_j^2 - 2z^* \sum_j \eta_j \pi_j$. This is a strictly convex quadratic function in $\boldsymbol{\pi}$ and achieves its unique minimum when $\pi_j = z^* \eta_j, \forall j \in J$.

We now show that $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} \sum_{j \in J} \frac{(\pi_j)^2}{\eta_j}$ attains the same minimizer $\pi_j = z^* \eta_j, \forall j \in J$. To do so, we decompose the problem into a constrained optimization problem as follows:

$$\min_{\boldsymbol{\pi} \in \mathcal{D}} \sum_{j \in J} \frac{\pi_j^2}{\eta_j} \quad \text{s.t.} \quad \sum_{j \in J} \pi_j = z^*. \tag{A.15}$$

Introducing a Lagrange multiplier λ, the Lagrangian is given by $\mathcal{L}(\boldsymbol{\pi}, \lambda) = \sum_{j \in J} \frac{\pi_j^2}{\eta_j} - \lambda \Big(\sum_{j \in J} \pi_j - z^* \Big)$. Taking the first-order condition with respect to each π_j yields $\partial \mathcal{L} / \partial \pi_j = 2\pi_j / \eta_j - \lambda = 0, \forall j \in J$, implying $\pi_j = \lambda \eta_j / 2, \forall j \in J$. Enforcing the normalization $\sum_j \pi_j = z^*$ gives $\sum_{j \in J} (\lambda \eta_j / 2) = z^*$, and since $\sum_j \eta_j = 1$, we obtain $\lambda = 2z^*$. Substituting back, we finally have $\pi_j = z^* \eta_j, \forall j \in J$.

Consequently, the unique minimizer of $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} \sum_{j \in J} \frac{(\pi_j)^2}{\eta_j}$ coincides with the minimizer found for $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} v(\boldsymbol{\pi})$. □

Theorem 3: When the dual optimal direction $\boldsymbol{\eta} = \frac{1}{|J|} \mathbf{1}$, as $K \to \infty$ and $M \to \infty$, the LGM produces a unique dual optimal solution, precisely the dual optimal center among all dual optimal solutions of the SCM.

Proof. Proof From Proposition 3, we know that when $\boldsymbol{\eta} = \frac{1}{|J|} \mathbf{1}$, the dual optimal solution to $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} \sum_{j \in J} (\pi_j)^2$ coincides with the solution to $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} v(\boldsymbol{\pi})$, which is precisely the dual optimal center.

From Theorem 2, as $K \to \infty$ and $M \to \infty$, we have $\lim_{K \to \infty, M \to \infty} \sum_{j \in J} \sum_{k=0}^{K} c_{jk} d_{jk} = \frac{1}{2M\theta} \sum_{j \in J} (\pi_j)^2$, which shows that the unique dual optimal solution produced by the LGM corresponds to the dual optimal solution to $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} \sum_{j \in J} (\pi_j)^2$.

In conclusion, we have demonstrated that when $\boldsymbol{\eta} = \frac{1}{|J|} \mathbf{1}$, as $K \to \infty$ and $M \to \infty$, the LGM produces a unique dual optimal solution, which is precisely the dual optimal center among all dual optimal solutions of the SCM.

□

Proposition 4: Given the weighted dual optimal direction $\boldsymbol{\eta}$, defined based on the dual optimal center $\boldsymbol{\pi}^*$ for the LP problem P, the optimal solution to $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} \sum_{j \in J} \frac{(b_j \pi_j)^2}{\eta_j}$ coincides with the solution to $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} v(\boldsymbol{\pi})$, which minimizes the DOCD and is precisely $\boldsymbol{\pi}^*$.

Proof. Proof The key idea is that the minimizer of $v(\boldsymbol{\pi})$, which defines the dual optimal central distance (DOCD), is exactly the dual optimal center $\boldsymbol{\pi}^*$. Therefore, to prove this proposition, it suffices to show that the solution to $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} \sum_{j \in J} \frac{(b_j \pi_j)^2}{\eta_j}$ coincides with $\boldsymbol{\pi}^*$.

To demonstrate this, let z^* denote the optimal objective value of the LP problem P, we decompose $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} \sum_{j \in J} \frac{(b_j \pi_j)^2}{\eta_j}$ into a constrained optimization problem as follows:

$$\min_{\boldsymbol{\pi} \in \mathcal{D}} \sum_{j \in J} \frac{(b_j \pi_j)^2}{\eta_j} \quad \text{s.t.} \quad \sum_{j \in J} b_j \pi_j = z^*. \tag{A.16}$$

By introducing a Lagrange multiplier λ, its Lagrangian relaxation is given as follows:

$$\mathcal{L}(\boldsymbol{\pi}, \lambda) = \sum_{j \in J} \frac{(b_j \pi_j)^2}{\eta_j} - \lambda \Big(\sum_{j \in J} b_j \pi_j - z^* \Big). \tag{A.17}$$

The first-order optimality condition with respect to each π_j gives

$$\frac{\partial \mathcal{L}}{\partial \pi_j} = \frac{2(b_j)^2 \pi_j}{\eta_j} - \lambda b_j = 0 \quad \Longrightarrow \quad \pi_j = \frac{\lambda}{2b_j} \eta_j. \tag{A.18}$$

Then, substituting $\pi_j = \frac{\lambda}{2b_j}\eta_j$ into the constraint $\sum_{j \in J} b_j \pi_j = z^*$, and noting that $\sum_{j \in J} \eta_j = 1$, we obtain

$$\sum_{j \in J} b_j \pi_j = \frac{\lambda}{2} \sum_{j \in J} \eta_j = z^* \quad \Longrightarrow \quad \lambda = 2z^*. \tag{A.19}$$

By combining Eq.(A.18) and Eq.(A.19), we obtain the closed-form optimal solution:

$$\hat{\pi}_j = \frac{\eta_j}{b_j} z^*, \ \forall j \in J. \tag{A.20}$$

Next, from the definition of the weighted dual optimal direction (Definition 5), we have

$$\eta_j = \frac{(b_j \pi_j^*)}{\sum_{j' \in J} (b_{j'} \pi_{j'}^*)}, \ \forall j \in J. \tag{A.21}$$

Substitute this expression into Eq.(A.20), and since $\sum_{j' \in J} (b_{j'} \pi_{j'}^*) = z^*$, we have

$$\hat{\pi}_j = \frac{(b_j \pi_j^*) / \sum_{j' \in J} (b_{j'} \pi_{j'}^*)}{b_j} z^* = \frac{(b_j \pi_j^*)/z^*}{b_j} z^* = \pi_j^*, \ \forall j \in J. \tag{A.22}$$

Therefore, the optimal solution to $\min_{\boldsymbol{\pi} \in \mathcal{D}^*} \sum_{j \in J} \frac{(b_j \pi_j)^2}{\eta_j}$ coincides with the dual optimal center $\boldsymbol{\pi}^*$, which completes the proof. □

Theorem 4: Let $\mathcal{D}^*$ denote the dual optimal polytope of the LP problem P, and let $\boldsymbol{\pi}^*$ represent its dual optimal center. Given the weighted dual optimal direction $\boldsymbol{\eta}$ defined with $\boldsymbol{\pi}^*$, as $K \to \infty$ and $M \to \infty$, the unique dual optimal solution proposed by the WLGM is exactly $\boldsymbol{\pi}^*$.

Proof. Proof Consider the dual constraint corresponding to the probe variable d_{jk} in the WLGM, given by $b_j \pi_j + \sigma_{jk}^- + \sigma_{jk}^+ = c_{jk}$. Recall from the proof of Theorem 1, we analyze the following cases for each probe variable in the context of an optimal solution to the WLGM:

- If $d_{jk} = \epsilon$, complementary slackness implies $\sigma_{jk}^- \leq 0$ and $\sigma_{jk}^+ = 0$, which gives $k\frac{\theta_j}{K} - b_j \pi_j \leq 0$;
- If $0 < d_{jk} < \epsilon$, complementary slackness implies $\sigma_{jk}^- = 0$ and $\sigma_{jk}^+ = 0$, leading to $k\frac{\theta_j}{K} - b_j \pi_j = 0$;
- If $d_{jk} = 0$, complementary slackness implies $\sigma_{jk}^- = 0$ and $\sigma_{jk}^+ \geq 0$, hence $k\frac{\theta_j}{K} - b_j \pi_j \geq 0$.

Therefore, for constraint j, the largest index k_j^* such that $d_{jk} = \epsilon$ is determined as $k_j^* = \left\lfloor \frac{b_j \pi_j K}{\theta_j} \right\rfloor$. So the probe variables satisfy: $d_{jk} = \begin{cases} \epsilon, & \text{if } k \leq k_j^* \\ 0, & \text{if } k > k_j^* \end{cases}$.

Next, the contribution of the probe variables corresponding to constraint j to the objective function (26) can be computed as:

$$\sum_{k=0}^{K} c_{jk}d_{jk} = \sum_{k=0}^{k_j^*} c_{jk}\epsilon = \sum_{k=0}^{k_j^*}(k\frac{\theta_j}{K})\frac{1}{MK} = \frac{\theta_j}{MK^2}\sum_{k=0}^{k_j^*} k$$
$$= \frac{\theta_j}{MK^2} \times \frac{1}{2}\Big\lfloor \frac{b_j\pi_j K}{\theta_j}\Big\rfloor(\Big\lfloor \frac{b_j\pi_j K}{\theta_j}\Big\rfloor + 1) = \frac{\theta_j}{2MK^2}\Big\lfloor \frac{b_j\pi_j K}{\theta_j}\Big\rfloor\Big\lceil \frac{b_j\pi_j K}{\theta_j}\Big\rceil. \tag{A.23}$$

Take the limit $K \to \infty$, the expression in Eq.(A.23) converges to

$$\lim_{K\to\infty}\sum_{k=0}^{K} c_{jk}d_{jk} = \lim_{K\to\infty}\frac{\theta_j}{2MK^2}\Big\lfloor \frac{b_j\pi_j K}{\theta_j}\Big\rfloor\Big\lceil \frac{b_j\pi_j K}{\theta_j}\Big\rceil = \frac{\theta_j}{2MK^2}\frac{b_j\pi_j K}{\theta_j}\frac{b_j\pi_j K}{\theta_j} = \frac{(b_j\pi_j)^2}{2M\theta_j} = \frac{(b_j\pi_j)^2}{2M\theta\eta_j}. \tag{A.24}$$

In summary, as $M \to \infty$ and $K \to \infty$, the unique dual optimal solution of the WLGM minimizes $\sum_{j\in J}\frac{(b_j\pi_j)^2}{\eta_j}$. By Proposition 4, the minimizer of this weighted squared L2 norm coincides with the dual optimal center $\boldsymbol{\pi}^*$. Hence, the WLGM yields the unique dual optimal center $\boldsymbol{\pi}^*$, as claimed. □

Proposition 5: Let $\mathcal{D}^*$ denote the dual optimal polytope of problem P, and $\mathcal{D}^*_{ScaledP}$ the dual optimal polytope of its scaled counterpart. Given an estimated weighted dual optimal direction $\hat{\boldsymbol{\eta}}$, the dual optimal solution $\boldsymbol{\pi}^{\hat{\eta}}$ produced by the WLGM satisfies the following properties:

1. If the ray $\{r\hat{\boldsymbol{\eta}} : r \geq 0\}$ has a nonempty intersection with $\mathcal{D}^*_{ScaledP}$, then $\pi_j^{\hat{\eta}} = \frac{r\hat{\boldsymbol{\eta}}_j}{b_j}, \exists r \geq 0$.
2. Otherwise, $\boldsymbol{\pi}^{\hat{\eta}}$ lies on the boundary of $\mathcal{D}^*$.

Proof. Proof In the case where the ray $\{r\hat{\boldsymbol{\eta}} : r \geq 0\}$ has a nonempty intersection with $\mathcal{D}^*_{ScaledP}$, the WLGM produces a unique dual optimal solution given by $\pi_j^{\hat{\eta}} = r\hat{\eta}_j/b_j$ for some scalar $r \geq 0$, as established in the proof of Theorem 4. This solution satisfies the condition that the scaled dual variables $b_j\pi_j^{\hat{\eta}}$ maintain constant ratios defined by $\hat{\boldsymbol{\eta}}$ across all constraints $j \in J$.

When the ray $\{r\hat{\boldsymbol{\eta}} : r \geq 0\}$ does not intersect $\mathcal{D}^*_{ScaledP}$, the objective function $\sum_{j\in J}\frac{(b_j\pi_j)^2}{\eta_j}$ (from Proposition 4) can be interpreted as the squared weighted Euclidean distance from the origin along the direction $\hat{\boldsymbol{\eta}}$ in the scaled dual optimal polytope. Thus, minimizing this function over $\mathcal{D}^*_{ScaledP}$ is equivalent to finding the point in $\mathcal{D}^*_{ScaledP}$ that is closest to the ray. Since the ray does not intersect the feasible set, the minimizer $\boldsymbol{\pi}^{\hat{\eta}}$ must lie on the boundary of $\mathcal{D}^*_{ScaledP}$. Through the inverse scaling transformation, this boundary point corresponds to a point on the boundary of the original dual optimal polytope $\mathcal{D}^*$, where one or more constraints are tight. Geometrically, $\boldsymbol{\pi}^{\hat{\eta}}$ is the orthogonal projection (in the weighted norm) of the ray onto $\mathcal{D}^*_{ScaledP}$, and this projection lies on the boundary of both the scaled and the original dual optimal polytopes. □

Theorem 5: Consider a (possibly degenerate) basis $\boldsymbol{B}$ of the LGM. Under the *Probe Cycling-Free Pivoting Rule*, by first updating all probe variables before selecting the entering non-probe variable, cycling due to degeneracy is completely eliminated.

Proof. Proof To prove that no cycling occurs under the Probe Cycling-Free Pivoting Rule, we first demonstrate that if an entering variable x_p satisfies $\bar{c}_p < 0$ and allows a positive pivot direction, then after updating all probe variables, the pivot results in a strictly improved objective value. This ensures that no cycling occurs among the x_p variables. Next, we show that cycling cannot arise during the update process of the probe variables themselves.

Part 1: No Cycling Among the x_p Variables.

Consider a non-probe variable x_p entering the basis. If the pivot is non-degenerate, i.e., the step size is strictly positive, then x_p increases to a positive value, and the objective decreases strictly because $\bar{c}_p < 0$ and the step size is positive. If the pivot is degenerate, i.e., the step size is zero and x_p remains at zero after the pivot. In this case, let $\boldsymbol{B}^0$ denote the basis after pivoting x_p, with the corresponding non-basic variables at their upper bounds denoted by $\boldsymbol{N}_2(\boldsymbol{B}^0)$.

The objective values corresponding to $\boldsymbol{B}$ and $\boldsymbol{B}^0$ are identical. However, the dual solution associated with $\boldsymbol{B}^0$ differs from that of $\boldsymbol{B}$. Consequently, by Theorem 1, there exists at least one probe variable d_{jk} whose reduced cost violates the optimality conditions, and need to be updated. After pivoting all such beneficial probe variables, we denote the resulting basis as $\boldsymbol{B}_d^0$, with non-basic variables at upper bounds given by $\boldsymbol{N}_2(\boldsymbol{B}_d^0)$. In the following, we show that the objective value corresponding to $\boldsymbol{B}_d^0$ strictly differs from that of $\boldsymbol{B}^0$; that is, updating all probe variables leads to a strictly improved objective value.

Th objective value of the LGM corresponds to basis $\boldsymbol{B}$ is computed by

$$\begin{aligned} z_{LGM}(\boldsymbol{B}) &= \boldsymbol{c_B}\boldsymbol{B}^{-1}\Big(\mathbf{1} - \epsilon \sum_{d_{jk}\in \boldsymbol{N}_2(\boldsymbol{B})} \boldsymbol{e}_j\Big) + \epsilon \sum_{d_{jk}\in \boldsymbol{N}_2(\boldsymbol{B})} c_{jk} = \boldsymbol{c_B}\boldsymbol{B}^{-1}\mathbf{1} + \epsilon\Big(\sum_{d_{jk}\in \boldsymbol{N}_2(\boldsymbol{B})} \Big(c_{jk} - \boldsymbol{c_B}\boldsymbol{B}^{-1}\boldsymbol{e}_j\Big)\Big) \\ &= \boldsymbol{c_B}\boldsymbol{B}^{-1}\mathbf{1} + \epsilon\Big(\sum_{d_{jk}\in \boldsymbol{N}_2(\boldsymbol{B})} \Big(c_{jk} - \pi_j\Big)\Big) \end{aligned} \tag{A.25}$$

Due to the different dual solutions, the set of probe variables attaining their upper bounds must change. This change can manifest in two ways. First, the set of non-basic probe variables at their upper bounds changes, i.e., $\boldsymbol{N}_2(\boldsymbol{B}_d^0) \neq \boldsymbol{N}_2(\boldsymbol{B}^0)$. According to Eq. (A.25), the first term remains unchanged because $x_p = 0$, thus this change immediately implies $z_{LGM}(\boldsymbol{B}^0) \neq z_{LGM}(\boldsymbol{B}_d^0)$. Second, the values of probe variables within the basis that reach their upper bounds may change. Specifically, either a probe variable originally at its upper bound leaves the basis and attains its lower bound, or a probe variable originally at its lower bound enters the basis and attains its upper bound. In both cases, the corresponding step size in that iteration is strictly positive, and thus the objective value must change, i.e., $z_{LGM}(\boldsymbol{B}^0) \neq z_{LGM}(\boldsymbol{B}_d^0)$.

Part 2: No Cycling During the Probe Variable Update Process.

Given a primal basic solution and its corresponding dual solution, the probe variables enter the basis in a well-defined order. At first, the algorithm selects a probe variable d_{j0} with minimal cost c_{j0} whose corresponding dual value π_j is maximal, ensuring the reduced cost is maximized. Once d_{j0} enters the basis, the dual value is updated to $\pi'_j = c_{j0}$, which implies that for all other probes $d_{jk}, k \in \{1, \ldots, K\}$, associated with the same constraint, $c_{jk} - \pi'_j = c_{jk} - c_{j0} > 0$, so no other probe from the same constraint can immediately enter the basis. The algorithm then proceeds to select the next probe variable among those with the same cost but associated with different constraints. In this way, all probes sharing the same cost but belonging to different constraints are updated as a group.

We next aim to show that pivoting among probe variables within a same-cost group cannot cycle. To do so, consider a set of feasible bases of the LGM that share $|J| - 1$ vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{|J|-1}$, with each basis including a distinct standard unit vector $\boldsymbol{e}_j$ for $j \in J$. Let $\boldsymbol{c} = (c_1, \ldots, c_{|J|-1}, c_d)$ denote the cost vector, where c_d corresponds to the probe variable included in the basis, and let the dual vector corresponding to any feasible basis $\boldsymbol{B}$ be $\boldsymbol{\pi} = \boldsymbol{c}\boldsymbol{B}^{-1}$. Consider any sequence of pivot bases

$$\boldsymbol{B}^1 \to \boldsymbol{B}^2 \to \cdots \to \boldsymbol{B}^o, \tag{A.26}$$

where each pivot replaces one basic vector with a nonbasic one whose reduced cost is negative, and let the corresponding dual vectors be $\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^o$. Denote $\boldsymbol{A} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{|J|-1}]$, which has rank $|J| - 1$. Let $\boldsymbol{v}$ be a nonzero row vector such that $\boldsymbol{v}\boldsymbol{A} = 0$. By construction,

$$\boldsymbol{\pi}^1\boldsymbol{A} = \cdots = \boldsymbol{\pi}^o\boldsymbol{A} = (c_1, \ldots, c_{|J|-1}), \tag{A.27}$$

so for any two consecutive bases $\boldsymbol{B}^i$ and $\boldsymbol{B}^{i+1}$, there exists a scalar $\lambda_{i,i+1} \neq 0$ such that

$$\boldsymbol{\pi}^i - \boldsymbol{\pi}^{i+1} = \lambda_{i,i+1}\boldsymbol{v}. \tag{A.28}$$

Consider the $(i+1)$-th component. Since $\pi_{i+1}^{i+1} = c_d$, we have

$$\pi_{i+1}^i - c_d = \lambda_{i,i+1}v_{i+1}. \tag{A.29}$$

As the $(i+1)$-th variable is selected to enter the basis at $\boldsymbol{B}^i$, it follows that $\pi_{i+1}^i > c_d$, and consequently $\lambda_{i,i+1}v_{i+1} > 0$.

Next, examine the pivot feasibility. Let $\boldsymbol{y} = (\boldsymbol{B}^i)^{-1}\boldsymbol{e}_{i+1}$, so that $\boldsymbol{B}^i\boldsymbol{y} = \boldsymbol{e}_{i+1}$, which can be written as

$$\boldsymbol{A}\boldsymbol{y}_A + y_{|J|}\boldsymbol{e}_i = \boldsymbol{e}_{i+1}, \tag{A.30}$$

where $\boldsymbol{y}_A$ contains the first $|J| - 1$ components of $\boldsymbol{y}$. Left-multiplying Eq.(A.30) by $\boldsymbol{v}$ yields

$$\boldsymbol{v}\boldsymbol{A}\boldsymbol{y}_A + y_{|J|}v_i = v_{i+1}. \tag{A.31}$$

Since $\boldsymbol{v}\boldsymbol{A} = 0$, we obtain $y_{|J|} = \frac{v_{i+1}}{v_i}$. Pivot feasibility requires $y_{|J|} > 0$, implying that v_i and v_{i+1} have the same sign. Repeating this argument shows that all $v_i, \forall i \in \{1, 2, \cdots, o\}$ share the same sign. Without loss of generality, assume $v_i > 0$ for all i (the case $v_i < 0$ for all i is symmetric). Then, from $\lambda_{i,i+1} v_{i+1} > 0$, it follows that $\lambda_{i,i+1} > 0$. Now suppose the sequence of pivots returns to the initial basis $\boldsymbol{B}^1$. The total change in dual vectors would be

$$\boldsymbol{\pi}^o - \boldsymbol{\pi}^1 = \sum_{i=1}^{o-1} (\boldsymbol{\pi}^{i+1} - \boldsymbol{\pi}^i) = -\Big(\sum_{i=1}^{o-1} \lambda_{i,i+1} \Big) \boldsymbol{v}. \tag{A.32}$$

Considering the first component, we obtain

$$\pi_1^o - \pi_1^1 = -\Big(\sum_{i=1}^{o-1} \lambda_{i,i+1} \Big) v_1. \tag{A.33}$$

Since $v_1 > 0$ and $\lambda_{i,i+1} > 0$ for all i, we conclude that $\pi_1^o < \pi_1^1 = c_d$. This contradicts the requirement for pivoting back to $\boldsymbol{B}_1$, which would need $\pi_1^o > c_d$. Therefore, the sequence cannot revisit any previously visited basis, and pivoting among probe variables within a same-cost group cannot cycle.

Thus, within such a group, either at least one probe variable eventually reaches its upper bound, causing a strict improvement in the objective value, or all probe variables are completed, with no further adjustments possible. Once the updates for this group are finished, the algorithm either proceeds to the next set of probe variables with a different cost, or terminates the probe update procedure if no further probe variables can improve the objective. Hence, it follows that the entire probe update process will not cycle.

By combining the two parts above, the Probe Cycling-Free Pivoting Rule effectively eliminates any cycling induced by degeneracy. □ □

Proposition 6: Let $\boldsymbol{B} \in \mathbb{R}^{|J| \times |J|} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_j, \ldots, \boldsymbol{b}_{|J|}\}$ be a feasible basis of the LGM, and let $\boldsymbol{B}^{-1}$ denote its inverse. Suppose a probe variable $d_{j^* k^*}$ enters the basis and the i^*-th basic variable leaves. The new basis can be expressed as $\boldsymbol{B}' = \boldsymbol{B} + (\boldsymbol{e}_{j^*} - \boldsymbol{b}_{i^*}) \boldsymbol{e}_{i^*}^T$, where $\boldsymbol{e}_{j^*}$ and $\boldsymbol{e}_{i^*}$ are unit vectors with 1 in the j^*-th and i^*-th components, respectively. The inverse of the new basis $\boldsymbol{B}'$ can be computed in $O(|J|^2)$ time.

Proof. Proof According to Sherman–Morrison rank-one update, when $1 + \boldsymbol{e}_{i^*} \boldsymbol{B}^{-1} (\boldsymbol{e}_{j^*} - \boldsymbol{b}_{i^*}) \neq 0$, the inverse of the new basis $\boldsymbol{B}'$ is computed as

$$\boldsymbol{B}'^{-1} = \boldsymbol{B}^{-1} - \frac{\boldsymbol{B}^{-1} (\boldsymbol{e}_{j^*} - \boldsymbol{b}_{i^*}) (\boldsymbol{e}_{i^*}^T \boldsymbol{B}^{-1})}{1 + \boldsymbol{e}_{i^*}^T \boldsymbol{B}^{-1} (\boldsymbol{e}_{j^*} - \boldsymbol{b}_{i^*})}. \tag{A.34}$$

From Eq.(A.34), with the inverse $\boldsymbol{B}^{-1}$ already available, the updated inverse $\boldsymbol{B}'^{-1}$ can be computed using a single matrix–vector multiplication $\boldsymbol{B}^{-1} (\boldsymbol{e}_{j^*} - \boldsymbol{b}_{i^*})$ and a single vector–matrix multiplication $\boldsymbol{e}_{i^*}^T \boldsymbol{B}^{-1}$, after which an outer product and a scalar division yield the final result. Thus, the time complexity of each update of $\boldsymbol{B}'^{-1}$ is $O(|J|^2)$. □

Proposition 7: Given a degenerate optimal basis of the SCM, the number of iterations required to obtain the optimal basis of the corresponding LGM is bounded above by $\sum_{j \in J} \left\lceil \frac{\pi_j^* K}{\theta} \right\rceil$.

Proof. Proof

According to the probe cycling-free pivoting rule described in Algorithm 2, probe variables corresponding to different constraints are processed sequentially without repetition, ensuring that no cycling occurs during the update process. Next, consider a specific constraint j. The consecutive probe variables for this constraint have costs that differ by $\frac{\theta}{K}$. Therefore, each pivot involving a probe variable for constraint j changes its dual variable by at least $\frac{\theta}{K}$. In the worst case, the dual component π_j requires $\left\lceil \frac{\pi_j^* K}{\theta} \right\rceil$ updates to reach its optimal value π_j^*. Summing over all constraints $j \in J$, we obtain a conservative upper bound on the total number of probe-processing iterations required for the LGM to reach the optimal basis from the degenerate optimal basis of the SCM: $\sum_{j \in J} \left\lceil \frac{\pi_j^* K}{\theta} \right\rceil$. □

9.2 Supplementary examples

Example B.1. *Consider the following linear program:*

$$\begin{aligned} \min \quad & 1000x_1 + 400x_2 + 1000u_1 + 1000u_2 + 1000u_3 && \text{(B.1)}\\ \textit{s.t.} \quad & x_1 + x_2 + u_1 = 1, && \text{(B.2)}\\ & x_1 + x_2 + u_2 = 1, && \text{(B.3)}\\ & x_1 + u_2 = 1. && \text{(B.4)} \end{aligned}$$

The corresponding dual optimal polytope is

$$\mathcal{D}^* = \{\pi_1 + \pi_2 + \pi_3 = 1000,\ \pi_1 + \pi_2 \le 400,\ \pi_1 \le 1000,\ \pi_2 \le 1000,\ \pi_3 \le 1000\}.$$

Its extreme points are $(\pi_1, \pi_2, \pi_3) = (0, 400, 600)$*,* $(0, 0, 1000)$*, and* $(400, 0, 600)$*. The exact dual optimal center is* $(133.3, 133.3, 733.3)$*.*

We next apply the WLGM formulation:

$$\begin{aligned} \min \quad & 1000x_1 + 400x_2 + 1000\eta_1 u_1 + 1000\eta_2 u_2 + 1000\eta_3 u_3 \\ & + \sum_{k=0}^{5}(200\eta_1 k)d_{1k} + \sum_{k=0}^{5}(200\eta_2 k)d_{2k} + \sum_{k=0}^{5}(200\eta_3 k)d_{3k} && \text{(B.5)}\\ \textit{s.t.} \quad & x_1 + x_2 + u_1 + \sum_{k=0}^{5} d_{1k} = 1, && \text{(B.6)}\\ & x_1 + x_2 + u_2 + \sum_{k=0}^{5} d_{2k} = 1, && \text{(B.7)}\\ & x_1 + u_2 + \sum_{k=0}^{5} d_{3k} = 1, && \text{(B.8)}\\ & 0 \le d_{jk} \le 0.001, \quad j \in \{1,2,3\},\ k \in \{0,1,2,3,4,5\}. && \text{(B.9)} \end{aligned}$$

According to Algorithm 1, we first apply the weighted dual optimal direction $\boldsymbol{\eta} = (0.998, 0.001, 0.001)$*, which yields the dual solution* $(\pi_1, \pi_2, \pi_3) = (399.6, 0.4, 600)$*. Next, with* $\boldsymbol{\eta} = (0.001, 0.998, 0.001)$*, the solution is* $(\pi_1, \pi_2, \pi_3) = (0.6, 399.4, 600)$*. Finally, with* $\boldsymbol{\eta} = (0.001, 0.001, 0.998)$*, we obtain* $(\pi_1, \pi_2, \pi_3) = (1, 1, 998)$*.*

By averaging these representative boundary points, the estimated dual optimal center is $(133.7, 133.6, 732.7)$*, which is very close to the exact center.*

Example B.2. *Consider the following linear program P with four variables and three constraints. We first normalize it into standard form. Then, to demonstrate the proposed approach, we augment each constraint with four probe variables, then apply the simplex method with the probe cycling-free pivoting rule to solve the resulting LGM-P:*

(P)

$$\begin{aligned} \min\ & -\tfrac{3}{4}x_4 + 20x_5 - \tfrac{1}{2}x_6 + 6x_7 \\ \text{s.t.}\ & -\tfrac{1}{4}x_4 + 8x_5 + x_6 - 9x_7 \ge 0 \\ & -\tfrac{1}{2}x_4 + 12x_5 + \tfrac{1}{2}x_6 - 3x_7 \ge 0 \\ & -x_6 \ge -1 \\ & x_4, x_5, x_6, x_7 \ge 0 \end{aligned}$$

(LGM-P)

$$\begin{aligned} \min\ & -\tfrac{3}{4}x_4 + 20x_5 - \tfrac{1}{2}x_6 + 6x_7 + \textstyle\sum_{j=1}^{3}\sum_{k=0}^{3} k d_{jk} \\ \text{s.t.}\ & -x_1 - \tfrac{1}{4}x_4 + 8x_5 + x_6 - 9x_7 + \textstyle\sum_{k=0}^{3} d_{1k} = 0 \\ & -x_2 - \tfrac{1}{2}x_4 + 12x_5 + \tfrac{1}{2}x_6 - 3x_7 + \textstyle\sum_{k=0}^{3} d_{2k} = 0 \\ & -x_3 - x_6 + \textstyle\sum_{k=0}^{3} d_{3k} = -1 \\ & x_1, x_2, x_3, x_4, x_5, x_6, x_7 \ge 0 \\ & 0 \le d_{jk} \le 0.001,\ j = \{1,2,3\},\ k = \{0,1,2,3\} \end{aligned}$$

The simplex iterations to solve the LGM-P under the proposed pivot rule proceed as follows:

- ***Iteration 1: Basic variables*** $(x_1, x_2, x_3) = (0, 0, 1)$***, dual solution*** $(0, 0, 0)$***, objective value*** 0***.***

  At the first stage of the pivoting process, no probe variable at lower bound has a negative reduced cost, and thus no update is required.

  Next, among the non-probe variables, x_4*, which has the most negative reduced cost, enters the basis while* x_2 *leaves.*

- ***Iteration 2: Basic variables*** $(x_1, x_3, x_4) = (0, 1, 0)$***, dual solution*** $(0, 1.5, 0)$***, objective value*** 0***.***

  At the first stage of the pivoting process, the probe variables are updated as follows:

  - d_{20} *enters the basis while* x_1 *leaves. The resulting basic and dual solutions are* $(d_{20}, x_3, x_4) = (0, 1, 0)$ *and* $(3, 0, 0)$*, respectively, with objective value* 0*.*
  - d_{10} *enters the basis while* d_{20} *leaves at its upper bound. The updated solutions become* $(d_{10}, x_3, x_4) = (0.0005, 1, 0.002)$ *and* $(0, 1.5, 0)$*, with objective value* -0.0015*.*
  - d_{21} *is directly updated to its upper bound, yielding* $(d_{10}, x_3, x_4) = (0.001, 1, 0.004)$ *and dual* $(0, 1.5, 0)$*, with objective value* -0.002*.*

  Next, among the non-probe variables, x_6*, which has the most negative reduced cost, enters the basis while* d_{10} *leaves.*

- ***Iteration 3: Basic variables*** $(x_3, x_4, x_6) = (1, 0.004, 0)$***, dual solution*** $(-5/3, 7/3, 0)$***, objective value*** -0.002***.***

  At the first stage of the pivoting process, the probe variables are updated as follows:

  - d_{22} *is directly updated to its upper bound, resulting in* $(x_3, x_4, x_6) = (0.9993, 0.0067, 0.0007)$ *and dual* $(-5/3, 7/3, 0)$*, with objective value* -0.002375*.*

  Next, among the non-probe variables, x_1*, which has the most negative reduced cost, enters the basis while* x_3 *leaves.*

- ***Iteration 4: Basic variables*** $(x_1, x_4, x_6) = (0.7495, 1.006, 1)$***, dual solution*** $(0, 1.5, 1.25)$***, objective value*** -1.2515***.***

  At the first stage of the pivoting process, the probe variables are updated as follows:

  - d_{30} *is directly updated to its upper bound, resulting in* $(x_1, x_4, x_6) = (0.75025, 1.007, 1.001)$ *and dual* $(0, 1.5, 1.25)$*, with objective value* -1.25275*.*
  - d_{22} *is directly updated to its lower bound, giving* $(x_1, x_4, x_6) = (0.75075, 1.005, 1.001)$ *and dual* $(0, 1.5, 1.25)$*, with objective value* -1.25325*.*
  - d_{31} *is directly updated to its upper bound, yielding* $(x_1, x_4, x_6) = (0.7515, 1.006, 1.002)$ *and dual* $(0, 1.5, 1.25)$*, with objective value* -1.2535*.*

  Since all nonbasic variables now have nonnegative reduced costs, no variable enters the basis, and the simplex iterations terminate.

This example illustrates that by adjusting the probe variables in the first stage of each pivot, the objective value decreases strictly even under degeneracy, thereby ensuring finite termination and preventing cycling.

9.3 The column generation algorithm for the ACSM

In this section, we adopt a column generation framework to solve the ACSM efficiently. The framework iteratively solves a restricted master problem (RMP) containing a subset of roster variables, and generates beneficial new rosters through pricing subproblems (PSP).

Let $\alpha_f \geq 0$ denote the dual variable associated with the flight covering constraint (34) and $\omega_m \leq 0$ denote the dual variable associated with the crew assignment constraint (35). According to the roster cost defined in Eq.(32), the reduced cost of a roster variable x_r for crew member m is computed as follows:

$$\begin{aligned}\bar{c}_r &= c_r - \sum_{f \in F} a_{fr}\alpha_f - \omega_m \\ &= c^{\text{hour}} \sum_{l \in L_r} \left(t_l^{\text{arr}} - t_l^{\text{dep}}\right) + c^{\text{ddh}}\, n_r^{\text{ddh}} + c^{\text{lay}}\, n_r^{\text{lay}} + c^{\text{day}}\, n_r^{\text{day}} - \sum_{f \in F} a_{fr}\alpha_f - \omega_m. \end{aligned} \tag{C.1}$$

The PSP aims to identify new roster variables with negative reduced cost, i.e., to minimize $\bar{c}_r$ over all feasible rosters $r \in R_m$ for each crew m. This problem can be formulated as a shortest path problem in a flight connection network $G^m(N^m, E^m)$, as illustrated in Figure C.1. The node set N^m consists of a source node and task nodes representing flights, deadheads, or ground activities, while the arc set E^m consists of:

1. *Start arcs*, linking the source node to tasks departing from the source station.
2. *Same-day connection arcs*, connecting two tasks on the same day if the time interval exceeds the minimum turn time and stations match.
3. *Cross-day connection arcs*, linking two tasks on different days if the time interval exceeds the minimum overnight rest time, and both the stations match and allow crew members to layover.

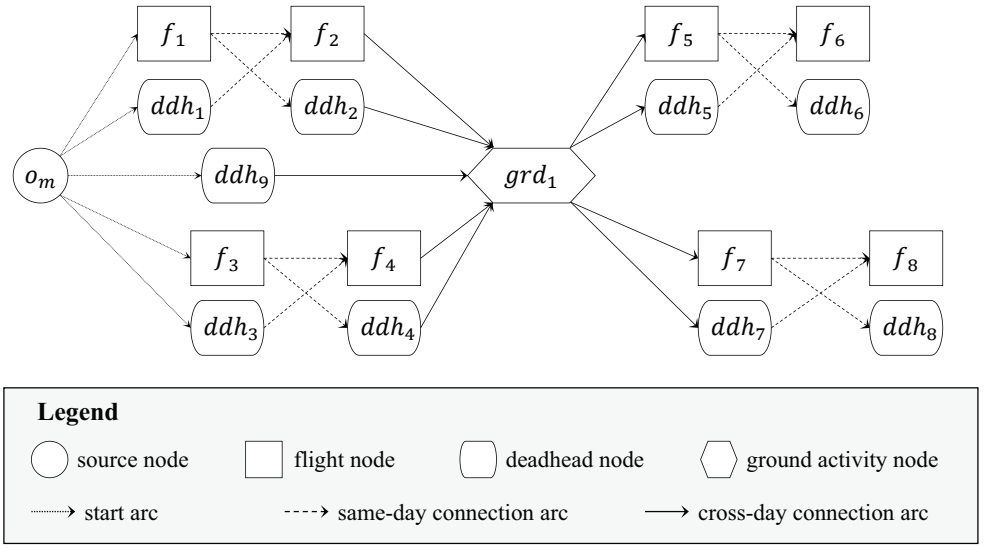


Figure C.1: Flight connection network for crew member m.

Due to the acyclic nature of the network, a multi-label algorithm is employed to solve the PSP efficiently. Each label maintains a reduced cost component and multiple resource components, such as duty flying time, duty duration, the number of flights per duty, the number of deadheads per duty, consecutive duty days, and cumulative periodic working time. To compute the reduced cost $\bar{c}_r$ during the shortest path search process, the reduced cost Eq.(C.1) is decomposed, with its components assigned to the corresponding nodes and arcs in the network. Specifically, each deadhead node incurs a cost of c^{ddh}, while each cross-day connection arc incurs a layover cost of $c^{\text{lay}} n_e^{\text{lay}}$, where n_e^{lay} represents the number of layover days associated with arc e. Additionally, the dual value $-\alpha_f$ is assigned to each flight node, and $-\omega_m$ is associated with the source node. Furthermore, the duty time cost and fixed duty day cost are determined during the search process, whenever the duty component is identified.

The overall procedure of the multi-label setting algorithm is as follows. At first, we sort all nodes in topological order according to their departure times. A single initial label is then created for the source node to record the historical state information of crew m. Subsequently, the label sets of the remaining nodes are updated sequentially following the topological order. When a partial path represented by a label at node n is extended along an arc $e = (n, n')$, the reduced cost and resource components are updated through predefined extension functions, generating a new label for node n'. If this new label satisfies all roster feasibility constraints, it is inserted into the label set of node n', and any dominated labels are removed. The dominance rule is defined as follows: label 1 dominates label 2 if and only if all components of label 1 are less than or equal to those of label 2, and at least one component of label 1 is strictly less than the corresponding component of label 2.

After processing all nodes, non-dominated labels corresponding to feasible rosters are extracted from task nodes that permit overnight stays for the crew. These labels are then sorted by ascending reduced cost, and a predefined number of the most beneficial rosters (if any exist) are added to the RMP. The RMP is then resolved, updating the dual variables for the next iteration. This process is repeated iteratively until no roster with negative reduced cost can be found, at which point the ACSM reaches its optimal solution.

9.4 The column generation algorithm for the CSM

The CSM is solved using a column generation framework, which decomposes the problem into a restricted master problem (RMP) and a pricing subproblem (PSP). At each iteration, the RMP is solved over a limited subset of patterns, and the corresponding dual information is used to identify new patterns that can potentially improve the objective value.

Let β_i denote the non-negative dual variable associated with the demand constraint (39), the reduced cost of a variable x_s is computed as

$$\bar{c}_s = c_s - \sum_{i \in I} a_{is}\beta_i = c_0 + c_w \left(W - \sum_{i \in I} a_{is} w_i \right) - \sum_{i \in I} a_{is}\beta_i \tag{C.1}$$

A pattern s with $\bar{c}_s < 0$ indicates a potential improvement to the current solution and should be added to the RMP. Hence, the PSP aims to find the pattern with the most negative reduced cost, which can be formulated as

$$\min \quad c_0 + c_w \left(W - \sum_{i \in I} a_{is} w_i \right) - \sum_{i \in I} a_{is}\beta_i \tag{C.2}$$

$$\text{s.t.} \quad \sum_{i \in I} a_{is} w_i \leq W, \tag{C.3}$$

$$a_{is} \in \mathbb{N}, \ \ \forall i \in I. \tag{C.4}$$

When the demand for each item equals 1, we have $a_{is} \in \{0, 1\}$. In this case, the subproblem reduces to a *0–1 knapsack problem*, where each item i is assigned a "value" of $c_w w_i + \beta_i$ and a "weight" of w_i, and the knapsack capacity corresponds to the roll width W. The goal is to identify the combination of items that maximizes the total value without exceeding W. Formally, the PSP can be written as

$$\max \quad \sum_{i \in I} (c_w w_i + \beta_i) a_{is} \tag{C.5}$$

$$\text{s.t.} \quad \sum_{i \in I} w_i a_{is} \leq W, \tag{C.6}$$

$$a_{is} \in \{0, 1\}, \ \ \forall i \in I. \tag{C.7}$$

To solve this problem, a standard dynamic programming approach is employed. The detailed steps of the algorithm are presented in the pseudocode provided in Algorithm C.1

By iteratively solving the RMP and PSP, at most one new pattern with a negative reduced cost is added to the RMP in each iteration (if such a pattern exists). This process continues until no further patterns with negative reduced cost can be identified, at which point the CSM attains its optimal solution.

Algorithm C.1: Dynamic Programming for the PSP of the CSM

Input: $|I|$ items with values $(c_w w_1 + \beta_1), \ldots, (c_w w_{|I|} + \beta_{|I|})$ and weights $w_1, \ldots, w_{|I|}$, roll width W
Output: Set of selected items that yield the maximum total value without exceeding capacity
Initialize $DP[0 \ldots |I|, 0 \ldots W] \leftarrow 0$, $Flag[0 \ldots |I|] \leftarrow$ False;
Initialize $I^* \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** $|I|$ **do**
  for $w \leftarrow 0$ **to** W **do**
    if $w_i \leq w$ **then**
      $DP[i, w] \leftarrow \max(DP[i-1, w], DP[i-1, w-w_i] + c_w w_i + \beta_i)$;
      if $DP[i, w] = DP[i-1, w-w_i] + c_w w_i + \beta_i$ **then**
        $Flag[i] \leftarrow$ True;
    else
      $DP[i, w] \leftarrow DP[i-1, w]$;
Initialize $w \leftarrow W$;
for $i \leftarrow |I|$ **to** *1* **do**
  if $Flag[i] =$ *True* **then**
    $I^* \leftarrow I^* \cup \{i\}$;
    $w \leftarrow w - w_i$;
return I^*;