UDC 631.316:519.852
DOI: 10.15587/1729-4061.2025.342750

DEVELOPMENT HYBRID OA-RG WITH MULTI-ROW TIME-AGGREGATED COVER CUTS FOR SOLVING MINLP IN COFFEE PLANTATION MAINTENANCE

Eko Hariyanto
Doctoral Candidate*
Poltak Sihombing
Corresponding author
Doctor of Computer Science, Professor*
E-mail: poltak@usu.ac.id
Erna Budhiarti Nababan
Doctor of Computer Science, Associate Professor*
Sawaluddin
Doctor of Computer Science, Associate Professor
Department of Mathematics and Natural Sciences**
*Department of Computer Science and Information Technology**
**Universitas Sumatera Utara
Dr. T. Mansur str., 9, Medan, Indonesia, 20155

The object of this research is the NP-hard combinatorial optimization problem in the allocation of limited resources for the maintenance of smallholder coffee plantations. In this study, a hybrid method of outer Approximation (OA) and reduced gradient (RG), enhanced by multi-row time-aggregated cover cuts (MTACC) is proposed to address the computational time efficiency problem in mixed-integer nonlinear programming (MINLP)-based combinatorial optimization problems. The testing was conducted using plantation land data from the Rahmat Kinara Coffee Farmers Association, which includes 538 land blocks with a total area of 825.5 hectares. Based on the numerical results obtained, it shows a reduction in the number of iterations by up to 38.83% and an increase in the speed of convergence time by up to 12.84%. The n_w feature in MTACC specifically controls the length of the time window to form multi-row covering slices that are suitable for the characteristics of the constraints, which affects the master and RG subproblems in overcoming the computational load. The evaluation results for testing parameters $n_w = 7$ and $n_w = 14$ show an increased contribution to convergence time of up to 10.1% by reducing the average master MILP time by 6.16%. Evaluation of the area under curve (AUC) metric confirms that MTACC is more stable in controlling optimality gaps across global iterations based on AUC (abs) assessment, which decreased by 21.6%; AUC per iteration decreased by 19.9%, and normalized AUC also decreased by 18.6%.

The results obtained can be effectively applied in small to large-scale coffee plantations, especially in decision support systems on low-power computing devices for production sustainability

Keywords: outer approximation, reduced gradient, MTACC, MINLP, plantation maintenance scheduling, limited resources optimization, smallholder coffee plantations, combinatorial optimization, time window constraints, decision support systems

Received 06.10.2025
Received in revised form 10.11.2025
Accepted 11.12.2025
Published 17.12.2025

How to Cite: *Hariyanto, E., Sihombing, P., Nababan, E. B., Sawaluddin, S. (2025). Development hybrid OA-RG with multi-row time-aggregated cover cuts for solving MINLP in coffee plantation maintenance. Eastern-European Journal of Enterprise Technologies, 6 (4 (138)), 36–48. https://doi.org/10.15587/1729-4061.2025.342750*

1. Introduction

Coffee is a high-value agricultural product with a strong global market and significant annual growth potential [1]. In Indonesia, coffee ranks among the largest plantation commodities and is vital to the rural economy [2], with 96% of plantations owned by smallholders [3]. However, maintenance remains traditional, shaped by natural conditions such as steep terrain and local socio-cultural factors, particularly reliance on manual labor [4]. These factors drive up operational expenses, often causing farmer losses and threatening sustainability. Therefore, a strategic and adaptive approach to plantation maintenance is essential as a decision-support system for managers to ensure sustainability.

The mixed-integer nonlinear programming (MINLP) model can be modified for use as an optimization model in intricate agricultural operations [5]. In this context, 'discrete decisions' refer to selecting from a finite set of alternatives, such as choosing specific times for planting or harvesting. "Nonlinearity of constraint functions" means that the relationships between variables aren't always proportional. For instance, changing the amount of water used for irrigation might not directly change the amount of crops produced. "Interactions between indicators" means that multiple factors, like water, energy, and food supplies, influence each other within the supply chain [6]. Nonlinear constraints often arise in agricultural or plantation operations. Examples include the relationship between harvest time, irrigation, and crop quality [7] or the interaction between water, energy, and food in supply chain design [8]. For coffee plants, 'optimal time window rules' refer to timeframes when maintenance tasks yield the best results, and 'proportional spacing' ensures that plants are distributed at intervals that maximize growth and yield. These factors form 'optimal time window inequalities,' meaning certain time and spacing requirements must be met for proper maintenance [9]. This makes them a nonlinear factor.

Unfortunately, the NP-hard nature of mixed-integer nonlinear programming (MINLP) meaning that it is a problem that is computationally very difficult to solve derives from mixed-integer linear programming (MILP), which is also NP-hard.

Because of this, MINLP leads to a vast and non-convex (not bowl-shaped) solution space, which is difficult to explore thoroughly within a reasonable computational time [10]. Non-convexity makes it more challenging to utilize mathematical relaxations (approximations) and hinders the straightforward certification of global optimality. Therefore, exploration-exploitation strategies (methods for balancing the search for new solutions versus reusing known beneficial solutions) need to be carefully adjusted [11]. This makes it impossible for plantation managers to operate the MINLP optimization model on low-power computing devices. As a result, developing methods to efficiently and lightly explore the MINLP solution space becomes crucial. This makes this research relevant for improving computational performance to generate optimal plantation maintenance schedules.

Applying effective decomposition techniques to solve MINLP models with complex structures improves computational performance and maintains solution quality [12]. Outer Approximation (OA) guarantees convergence to a global solution within reasonable computing time by dividing complex models into a mixed-integer linear programming (MILP) master problem and continuous nonlinear or nonlinear programming (NLP) subproblems [13]. Reduced gradient (RG), based on the generalized reduced gradient, generates feasible primal solutions and multiplier estimates, making it suitable for use as an NLP subproblem solver within a decomposition framework, particularly OA [14]. OA and RG hybridization provides a framework that balances the exploration and exploitation of the solution space, offering a strong alternative to address structural and computational challenges in realistic non-convex MINLPs. However, this hybrid has been little explored, meaning that it requires further study [15].

On the other hand, cutting planes help accelerate the process of finding optimal solutions [16]. Research on cutting strategies in OA continues to develop based on specific needs [17]. This paper confirms the critical role of cutting planes in improving the efficiency of exploring the MINLP solution space. This research contributes a theoretical understanding of the influence of hybrid OA-RG with time-aggregation-based cutting strategies for daily capacity constraints and proportional spacing on MINLP performance. Practically, this research produces a planned maintenance schedule that adheres to agronomic rules. It serves as a decision-making tool for plantation managers to allocate labor to maintenance tasks in a specific order, aiming to reduce operational expenses.

In modern conditions, smallholder coffee plantations face simultaneous pressure to remain economically viable, comply with agronomic requirements, and operate under tightening resource constraints. Managers are expected to plan labor-intensive maintenance activities over extended horizons while dealing with heterogeneous topography and limited budgets, yet they often rely on low-power computing devices or basic information systems. At the same time, advances in MINLP modeling and decomposition have not yet fully translated into tools that can be used routinely in such operational settings because the underlying NP-hard problems typically demand substantial computational effort and specialized solvers. These gaps make it necessary to conduct scientific research that designs algorithmic frameworks explicitly tailored to the structure of plantation maintenance problems so that complex time-window and capacity constraints can be handled efficiently within realistic computational limits.

The results of these studies can directly support practice by providing a scheduling framework that is both implementable and interpretable for plantation managers. By integrating the hybrid OA-RG approach with multi-row time-aggregated cover cuts, the proposed method is able to generate maintenance schedules that respect daily capacity limits, proportional spacing rules, and optimal time windows, while reducing the computational burden compared with classical MINLP approaches. Such an improvement creates concrete opportunities to embed the model into decision support systems running on low-specification devices, allowing managers to allocate labors, sequence maintenance tasks across land blocks, and assess cost scenarios in a more systematic manner. Beyond coffee, the methodological insights obtained here can be adapted to other resource-constrained agricultural systems that require long-term, rule-based maintenance planning, reinforcing the role of optimization as a practical tool for sustainable production.

Therefore, research on developing a lightweight and accurate optimization model for scheduling maintenance in coffee plantations is relevant. This model must handle daily capacity constraints and proportional spacing in optimal time window inequalities. Such research is essential for developing costly computational solutions and helps ensure the sustainability of production in traditional agriculture.

2. Literature review and problem statement

Several studies on OA hybridization in MINLP cases have aimed to make the global search more focused and achieve faster convergence. Research [18] uses logic-based OA hybridization. This method replaces the algebraic reformulation of MINLP with the direct solution of logic and (non)convex subproblems. This process generates feasible cuts and incumbents, improves exploitation of disjunctive structure, leads to earlier infeasibility detection, and prunes the global search space to relevant regions. However, the per-iteration cost for subproblems increases. This expense is because it involves solving more complex structured logic or computation and depends on specialized solvers.

The hybrid OA variant with parallelism and decomposition distributes the master/subproblem work or slice generation across multiple threads. This mitigates scalability bottlenecks by dividing the computational load. The approach enhances the ability to handle larger instances through parallel processing. This process speeds up the OA cycle without altering the global structure of the solution. However, this approach is mitigative, not eliminative, of the scale issue. Careful load balancing and orchestration are still needed [19].

Strengthening the formulation through cutting planes serves as another important accelerator in the solution process. Research [20] demonstrates this by utilizing multiple-generation cuts and partial surrogate cuts to improve the quality of information at each subproblem call. These strategies impact the global solution space. The master receives more informative cuts, which leads to improved integer candidates and lower bounds, accelerated convergence, and often reduced runtime. However, using many diverse cuts increases the master's complexity and lifts the computational burden with each iteration.

The disjunctive cutting tradition develops valid inequalities that approach the convex hull of the feasible set. This improves the quality of relaxation bounds and the effectiveness of cutting procedures [21]. In 0–1 integer programs, perspective cuts provide significant strengthening for on/off variables. This structure often appears in scheduling decisions for time slots [22]. In convex MINLP, the extended cutting plane (ECP) systematically aggregates inequalities and remains compatible with MILP-based OA solvers [23]. Intersection cuts and split cuts for structured convex sets also enrich the set of inequalities. These cuts improve branching policies and increase relaxation accuracy [24]. Despite this diversity, the cutting literature has not explicitly examined the construction of multi-row, time-aggregated inequality that combines daily capacity and agronomic spacing rules for plantation maintenance scheduling. This leaves room for contribution.

This research views the existing OA hybrid architecture as still lacking operational criteria at the subproblem level to determine when and on which candidate's RG should be called so that the resulting cuts remain consistent with global relaxation, while also reducing dependence on global NLP in non-convex cases. Hybrid OA literature indicates that strengthening the formulation can accelerate convergence but tends to enlarge the master and raise numerical issues [19], while logic-based OA variants enrich structural information but increase the complexity of subproblems per iteration and still require a global strategy when non-convexity occurs [25]. This research integrates RG into the OA subproblem based on the principle of proximity (through limited neighborhood/local branching) to balance relaxation strength and computational burden without enlarging the master.

Additionally, this research introduces multi-row time-aggregated cover cuts (MTACC) as a cutting plane that incorporates daily capacity and task execution, spreading constraints proportionally into tighter time window inequalities, rather than summing daily limits [26, 27]. To our knowledge, there has been no research that explicitly integrates OA-RG with multi-row time-aggregated inequality for an agricultural context that combines productivity heterogeneity due to topography, transportation constraints, and daily labor number decisions at the smallholder plantation scale. MINLP and RCPSP-based scheduling literature demonstrates the relevance of time windows and capacity but has not yet presented cutting planes that capture inter-round and inter-day correlations as needed in coffee plantations [28]. Surrogate cuts, perspective cuts, disjunctive cuts, the extended cutting plane, and intersection cuts have been shown to tighten the formulation on the cutting plane side, but they have not been specifically directed toward multi-line time aggregation with spacing rules [20–24].

All this allows to assert that it is expedient to conduct a study on develop more efficient and practical optimization models that can not only solve problems faster but also be applied to devices with limited resources. This is urgent to create more efficient solutions for managing coffee plantation maintenance, which will ultimately support the optimal sustainability of coffee production.

3. The aim and objectives of the study

The aim of this study is to develop a method for solving combinatorial optimization problems in MINLP that integrates hybrid outer approximation (OA) decomposition techniques and reduced gradient (RG) with aggregation-based cutting-plane modifications according to constraint characteristics in the context of limited resource allocation optimization in coffee plantation maintenance. In practical terms, this method is expected to provide a lightweight and accurate decision support system solution for low-power computing devices.

To achieve this aim, the following objectives are accomplished:

– to develop hybrid system with MTACC architecture;

– to evaluate the performance of the hybrid method with MTACC on the MINLP model;

– to visualize optimization models for limited resource allocation problems on land blocks and maintenance stages corresponding to optimal time windows.

4. Materials and methods

4. 1. The objects and hypotheses of the study

The object of this study is the NP-hard combinatorial optimization problem in the allocation of limited resources for the maintenance of smallholder coffee plantations. This research focuses on optimizing labor allocation with constraints on daily capacity, minimum frequency, proportional spacing, and optimal time windows in maintaining smallholder coffee plantations using the MINLP model. This problem falls under combinatorial optimization problems. The main hypothesis used states that integrating reduced gradient (RG) and multi-row time-aggregated cover cuts (MTACC) into outer approximation (OA) decomposition has a significant effect on computational time efficiency and optimality gap. This study assumes that the labor requirements allocated to multiple plots for maintenance tasks are limited.

4. 2. Mathematical formulation

Table 1 contains the decision variables, which values are determined by the solver to achieve an optimal solution. Equations (2)–(8) represent the constraints that limit the values of the decision variables to remain within the feasible solution set. Tables 2, 3 present sets and parameters that influence these three components.

Table 1

Decision variables

Variable	Description
Binary $y_{i,k,t}$	1 if stage k is performed in area i on day t, 0 otherwise
Continue $h_{i,k,t}$	The worker's requirement for stage k in area i on day t
Integer w_t	The total number of workers employed on day t
Integer b_t	The number of transportation trips required on day t

Table 2

Set

Notation	Description	Value
i	Maintained land blocks	{1, 2, 3, …,5 38}
k	Maintenance stage	{pr (pruning), fr (fertilization), wd (weed control), ps (pest control)}
t	Time horizon (days) for maintenance	{1, 2, 3, …,3 60}
$T_k \subseteq T$	Time window (days) for stage k	
	T_{pr}	{1, …, 90}
	T_{fr}	{151, …, 360}
	T_{wd}	{1, …, 360}
	T_{ps}	{1, …, 180}

Table 3

Parameter

Parameter	Description
A_i	Area for each plantation block i (in hectares)
$p_{i,k,t}$	Worker productivity (hectares per worker per day) depending on land slope
f_k	Number of mandatory executions $f_{pr} \geq 3$; $f_{fr} \geq 2$; $f_{wd} \geq 6$; $f_{ps} \geq 2$
W	Wage cost per worker per day
C	Cost per trip for transportation
$L_{\max}$	Maximum number of available workers

Based on the model description, the objective function as an MINLP model can be formulated as follows

$$\min Z = \sum_i \sum_k \sum_t \left(W \cdot w_t + C \cdot b_t\right). \tag{1}$$

Subject to:

$$\sum_{t \in T_{pru}} y_{i,pu,t} \geq 3, \ \sum_{t \in T_{fer}} y_{i,pu,t} \geq 2, \ \sum_{t \in T_{weed}} y_{i,gi,t} \geq 6,$$

$$\sum_{t \in T_{pest}} y_{i,ha,t} \geq 2, \ \forall i \in I, \tag{2}$$

$$\sum_{t \in T_k} z_{ikmt} = 1, \ z_{ikmt} \leq y_{ikt}, \ \forall i,k,m,t, \tag{3}$$

$$\tau_{ikm} = \sum_{t \in T_k} t z_{ikmt}, \ \forall i,k,m, \tag{4}$$

$$h_{ikt} \geq \frac{A_i}{p_{si}} y_{ikt}, \ \forall i,k,t, \tag{5}$$

$$\sum_{i \in I} \sum_{k \in K} h_{ikt} \leq w_t, \ \forall t, \tag{6}$$

$$0 \leq w_t \leq L_{\max}, \ \forall t, \tag{7}$$

$$2b_t \geq w_t. \ \forall t. \tag{8}$$

Equation (1) is the objective function to minimize the operational costs of all maintenance stages that must be carried out on all blocks within the time horizon, while equations (2)–(8) are the constraints that must be adhered to. Constraint (2) determines the frequency of implementation for each stage. Constraints (3), (4) mandate the implementation of each stage with a proportional spacing in different areas. Constraint (5) requires the minimum worker requirements for scheduling stage k. Constraints (6), (7) determines the limit on the total number of workers that can be allocated so as not to exceed available capacity. Constraint (8) relates the number of workers allocated to the number of motorcycles used.

4. 3. Numerical experiment

Numerical experiments were conducted using Windows-based computer hardware with the following specifications: CPU Intel Core i7-10750 2.6 GHz, RAM 16 GB DDR4, and GPU NVidia GeForce GTX 1650 Ti. The simulation was conducted in several trials using dataset instances from Table 4, to be solved and evaluated. The tolerance value ε_{gap} is 1% and the time limit per instance is 3 hours.

Table 4

Instance dataset

Instance ID	Numbers of land blocks coverage	Total area of sloping land (ha)	Total area of steep land (ha)	Total area (Ha)
Ins1	50	62.9	10	72.9
Ins2	100	129.4	22.1	151.5
Ins3	150	205.3	29.8	235.1
Ins4	200	271.7	40.5	312.2
Ins5	250	335.9	52.5	388.4
Ins6	300	379	84.3	463.3
Ins7	350	398.1	139	537.1
Ins8	400	423.7	182.9	606.6
Ins9	450	461.8	225.1	687.9
Ins10	500	503.1	270.6	773.7
Ins11	538	525.9	299.6	825.5

Table 4 contains 11 instances representing the number of land blocks covered in a tiered manner, as well as the total area of each land block. This data is used to obtain the results of the scalability analysis during the testing process.

5. Results of the hybrid method utilizing multi-row time-aggregated cover cuts

5. 1. Hybrid architecture

The system architecture is presented in Fig. 1. The system constructs an initial relaxation in the form of an OA-based master MILP. The master solves the relaxation model and calculates the lower bound, then passes the information to the two reinforcement paths. The first path is the RG Subproblem Block, which manages the calling of subproblems through the candidate selection gate, runs the NLP subproblem with RG, extracts gradient or subgradient information, and applies a consistency filter to the global relaxation before adding the OA and incumbent cuts to the master. The second path is the MTACC Block, which triggers time window preselection, aggregates requests and capacity per window, determines multi-row covers, performs spacing-based rule lifting, evaluates violations and cut depth, and applies a consistency filter to the global relaxation before adding the MTACC cuts to the master. The iterative process continues by updating the master and repeating both paths until the solution gap does not exceed the tolerance ε.

The following is an explanation of the proposed hybrid architecture.

1. Integration of reduced gradient.

Let's integrate reduced gradient (RG) into the continuous subproblems of OA decomposition. The following is the process flow of RG.

Input from OA Master and the global lower bound. RG receives the master OA output, which includes a relaxation solution ($\bar{x}$) and a global lower bound (LB). The LB value serves as an evaluation anchor for continuous subproblems, while $\bar{x}$ serves as a reference for the active cut structure. This information guides the RG process to focus on candidates relevant to the global optimum, rather than just local improvements.

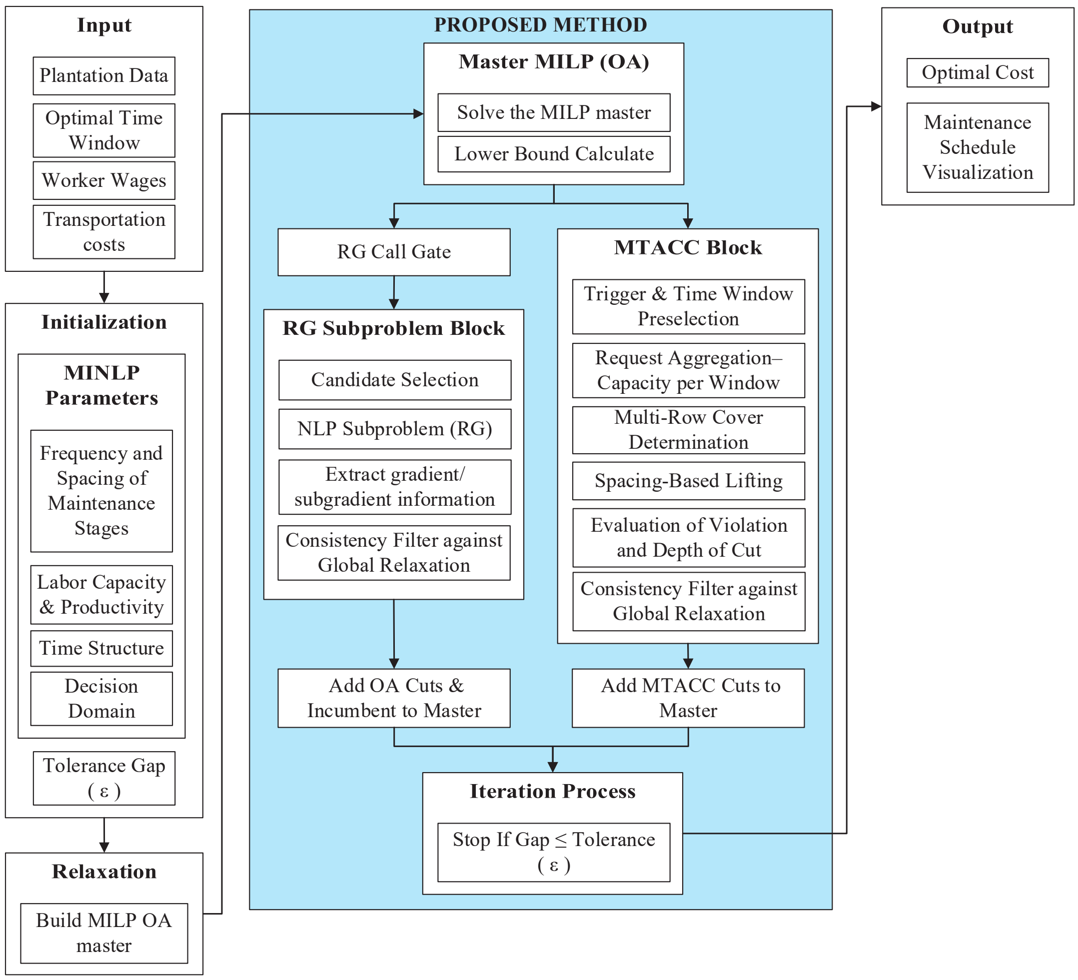

Fig. 1. Block diagram of the overall system

Candidate selection (RG call gate) is based on proximity and quality of the boundary. The system generates integer candidates $\hat{x}$ around a reference solution (e.g., the incumbent or the last master solution) using structured rounding or local branching within the environment $\|\hat{x}-x^{ref}\|_1 \leq \Delta$, and the system then filters the candidates using bound quality against LB with the criterion: $Z_{relax}(\hat{x}) - LB \leq \tau$ where $Z_{relax}(\hat{x})$ is the objective function value of the relaxation. This screening ensures that only promising candidates are processed. The system limits the number of RG calls per iteration to control computational costs.

Continuous initialization and stable trust-region. For each selected $\hat{x}$ the system forms a continuous initial guess y^0 (taken from the relaxation or incumbent) and sets the trust region so that the RG step remains close to a valid relaxation solution. This strategy stabilizes local iterations when nonlinear constraints are sharp or sensitive to changes.

Solving the continuous subproblem with RG to enforce nonlinear feasibility through reduced space and generate a continuous feasible solution y^*. After that, the system will obtain the candidate of $UB(\hat{x})$. If $UB(\hat{x})$ outperforms the current best UB, the system updates the incumbent and related operational plan.

The system extracts derivative information (gradient/subgradient) at the point $(\hat{x}, y^*)$ and forms the OA segment (tangent/subgradient), which tightens the relaxation in the master. This snippet filters out regions (x, y) that are no longer supported by valid local information.

Filter consistency against global relaxation and coefficient weakening. Before the cut is inserted into the master, the system checks the global validity of the cut against the active relaxation. If the cut is too aggressive (risking cutting off the global solution), the system weakens the coefficient based on the subgradient of that relaxation until the cut is valid again. This step is important for non-convex cases to maintain the integrity of the LB.

Master update. The pieces that passed the filter were added to the master along with the incumbent's fixes, and then the OA master was recompiled. In the same/next cycle, the MTACC block adds time-aggregated closing pieces based on R_t and a window of duration n_w, causing the discrete space to narrow and nonlinear relaxation to increase sharply synergistically.

2. Development of multi-row time-aggregated cover cuts.

There is a conflict between the constraints in the above MINLP model, where capacity constraints apply on a daily basis, while the spreading rules are cumulative across time. This condition can cause looseness in the master LP relaxation. Therefore, let's develop multi-row time-aggregated cover cuts (MTACC), a cutting technique that forms daily demand coefficients based on productivity and performs lifting based on spacing rules. The goal is to strengthen the constraints by considering the interdependencies of several consecutive periods, particularly the proportional distance between tasks. Fig. 2 uses pseudocode to computationally illustrate the concept of MTACC.

Algorithm 1: MTACC Block for OA–GRG (Multi-row Time-Aggregated Cover Cuts)

Input: Resource capacity per period $\{R_t\}$; base demand parameters $\{\alpha_{i,t}\}$; spacing parameter θ;
candidate window set S_{n_w} (all contiguous subranges of length n_w);
current relaxed master solution $\bar{x}$; violation threshold $\varepsilon > 0$; cut budget $M \in \mathbb{N}$.
Output: A set of valid lifted multi-row time-aggregated cover cuts to be added to the OA master.

Step 1: Trigger & Preselection of Windows
$S_{\text{cand}} \leftarrow$ PreselectWindows$(S_{n_w}, \bar{x}, \{\alpha_{i,t}\}, \{R_t\}, \theta)$

Step 2: Aggregate Demand & Capacity per Window
foreach $S \in S_{\text{cand}}$ **do**
  $B(S) \leftarrow \sum_{t\in S} R_t$ // Aggregated capacity
  foreach *activity-time* (i,t) **do**
    $a_{i,t}(S) \leftarrow \begin{cases} \alpha_{i,t}, & t \in S \\ 0, & \text{otherwise} \end{cases}$

Step 3: Minimal Multi-row Cover Identification
foreach $S \in S_{\text{cand}}$ **do**
  $J(S) \leftarrow$ FindMinimalCover$(\{a_{i,t}(S)\}_{t\in S}, B(S))$ // $\sum_{(i,t)\in J(S)} a_{i,t}(S) > B(S)$ and minimal

Step 4: Spacing-based Lifting of Coefficients
foreach $S \in S_{\text{cand}}$ **do**
  foreach $(i,t) \in J(S)$ **do**
    $\tilde{a}_{i,t}(S) \leftarrow a_{i,t}(S) +$ LiftSpacing$((i,t), S, \theta, \bar{x})$

Step 5: Violation and Depth Evaluation
Initialize candidate cut list $\mathcal{C} \leftarrow \emptyset$
foreach $S \in S_{\text{cand}}$ **do**
  $v(S) \leftarrow \sum_{(i,t)\in J(S)} \tilde{a}_{i,t}(S)\, \bar{x}_{i,t} - B(S)$
  if $v(S) > \varepsilon$ **then**
    depth$(S) \leftarrow$ ComputeDepth$(v(S), \{\tilde{a}_{i,t}(S)\})$ // e.g., $v(S)/\|\tilde{\mathbf{a}}(S)\|_1$
    Append $(S, J(S), \{\tilde{a}_{i,t}(S)\}, B(S), \text{depth}(S))$ to $\mathcal{C}$

Step 6: Consistency Filter vs Global Relaxation
$\mathcal{C}_{\text{sel}} \leftarrow$ TopM$(\mathcal{C}, M)$ // select top-M by depth
foreach *candidate* $(S, J(S), \{\tilde{a}_{i,t}(S)\}, B(S)) \in \mathcal{C}_{\text{sel}}$ **do**
  if CheckValidity$(\{\tilde{a}_{i,t}(S)\}, B(S))$ *is* ***false*** **then**
    $\{\tilde{a}_{i,t}(S)\} \leftarrow$ WeakenCoeffs$(\{\tilde{a}_{i,t}(S)\})$ // e.g., subgradient from convex relaxation
  if CheckValidity $(\{\tilde{a}_{i,t}(S)\}, B(S))$ *is* ***true*** **then**
    Add cut $\underset{(i,t)\in J(S)}{\quad}\tilde{a}_{i,t}(S)\, x_{i,t} \le B(S)$ to the master

return *All admitted MTACC cuts*

Fig. 2. Pseudocode for the proposed cutting technique

The MTACC concept is as follows:

1. Time aggregation. This technique aggregates time-related constraints across multiple periods to create tighter bounds on the solution space. By considering the order of tasks for example, pruning, fertilizing, and pest control this cutting ensures scheduling adheres to the temporal dependencies between tasks.

2. Coverage inequality. This inequality restricts the feasible region by eliminating solutions that do not meet the inter-task spacing requirements. The cutting operation tightens the feasible region, thereby improving the bound on the objective function and accelerating convergence in the optimization process.

Here's an explanation of the MTACC algorithmic block within the OA-RG hybrid:

1) Input.

The system receives input on resource capacity per period (R_t), basic requirement parameters per activity-period ($\alpha_{i,t}$), and the minimum distance parameter (θ). The system also accepts the window length (n_w), and the set of candidate windows (S_{nw}), which contains all continuous subranges of duration n_w. The current state of the solution is represented by the master relaxation solution ($\bar{x}$) or the best incumbent if available.

2) Output.

The process involves producing the MTACC cut family, which has been lifted and validated against the current global relaxation. These cuts tighten the MILP master along the time dimension without changing the meaning of the cost or wage terms, resulting in a stronger lower bound for the master, a narrower fractional solution space, and more directed subsequent OA iterations.

3) Procedure:

a) trigger & window preselection. Calculate the capacity utilization density indicator at $\bar{x}$ for each window $S \in \mathcal{S}$

$$load(S) = \frac{\sum_{t\in S}\sum_i \alpha_{i,t}\bar{x}_{i,t}}{\sum_{t\in S} R_t}.$$

Choose $S_{cand} \subseteq S$ containing the window of $load(S)$ with the highest or most potentially conflicting spacing patterns (task spacing < θ).

Demand-capacity aggregation per window: for each $S \in S_{cand}$, the aggregated "knapsack-cover" inequality takes the form Aggregate capacity

$$B(S)=\sum_{t\in S} R_t.$$

Basic demand coefficient per activity-day

$ai,t(S) = \alpha_{i,t}$ (only if $t \in S$, otherwise zero).

b) multi-line cover determination (minimum cover set). Find the minimal set of indices $\mathcal{J}(S) \subseteq \{(i, t) : t \in S\}$, such that

$$\sum_{(i,t)\in \mathcal{J}(S)} a_{i,t}(S) > B(S);$$

c) spacing-based lifting. For each $(i, t) \in \mathcal{J}(S)$, calculate the lifted coefficient: $\tilde{a}_{i,t}(S)=a_{i,t}(S)+lift(i,t)|S,\theta,\bar{x}$, where $lift(.)$ adds a proportional penalty to activities adjacent ($< \theta$) in S to sharpen the contribution to cross-day capacity consumption;

d) evaluate violations & cutting depth. Calculate violations against the aggregate limit

$$v(S)=\sum_{(i,t)\in \mathcal{J}(S)} \tilde{a}_{i,t}(S)\tilde{x}_{i,t}-B(S).$$

Calculate the depth as the ratio of violations to the coefficient norm

$$depth(S)=\frac{v(S)}{\tilde{a}(S)_1},$$

select $S_{sel} \subseteq S_{cand}$ containing the window with the largest $depth(S)$ and $v(S) > \varepsilon$;

e) consistency filter for global relaxation. For each $S \in S_{sel}$, test the validity of the slice against the relaxation supporting the master. If invalid, weaken the lifting coefficient $\tilde{a}_{i,t}(S)=\max\{a_{i,t}(S),\, \tilde{a}_{i,t}(S)-\delta\}$, or discard the candidate;

f) add MTACC cuts to the master. Add multi-row lifted cover inequalities

$$\sum_{(i,t)\in \mathcal{J}(S)} \tilde{a}_{i,t}(S)x_{i,t} \le B(S),$$

for all $S_{sel} \subseteq S_{cand}$ that pass the test, with a limit on the number of cuts per iteration (budget) to control the size of the master.

5. 2. Evaluating the performance of multi-row time-aggregated cover cuts

Let's conduct testing on the parameter n_w used in MTACC to optimize the scheduling of coffee plantation maintenance. The parameter n_w refers to the length of the time window used in the aggregation process to form multi-row cover cuts. The purpose of this analysis is to evaluate the effect of changes in the n_w value on computational performance and optimal solutions, as well as to assess the trade-off between time efficiency and the quality of the resulting solutions. Tables 5, 6 show the results of the MTACC test on the OA-RG hybrid with $n_w = 7$ and $n_w = 14$.

The test results show that the optimality gap is usually smaller at $n_w = 14$ than at $n_w = 7$. The result means that a longer time window gives solutions that are closer to optimality. For example, at Ins1, the optimality gap for $n_w = 7$ is 0.86, while at $n_w = 14$, it becomes 0.32. This difference is reflected in most instances, indicating that using $n_w = 14$ yields better solutions in terms of quality, although there are some exceptions for large instances, such as Ins10 and Ins11.

From the perspective of computational time to solve the optimization problem (convergence), the comparison between the two parameters shows little difference. Fig. 3 presents a comparison of convergence times between the two n_w MTACC parameters, which involve the master MILP time, MTACC separation time, and RG subproblem time.

The convergence time for $n_w = 7$ yields longer results compared to $n_w = 14$ on a small to medium scale (Ins1 to Ins8), whereas on a large scale (Ins9 to Ins11), $n_w = 7$ is superior to $n_w = 14$. Let's also observe significant differences in the master MILP time and MTACC separation. The master MILP time for $n_w = 7$ tends to be greater than for $n_w = 14$, but the MTACC separation time for $n_w = 7$ is smaller than for $n_w = 14$ across all problem scales. This data indicates that the parameter $n_w = 7$ yields more MTACC cuts than $n_w = 14$. When combined with the master MILP, the master problem becomes larger and denser from iteration to iteration, necessitating more time to process the increasingly complex model.

Table 5

MTACC Test results with parameter $n_w = 7$

No.	Instance	$n_w = 7$					
		Convergence time (s)	Optimality gap	MILP master time (s)	MTACC separation time (s)	Total time of RG sub problems (s)	Number of OA iterations
1	Ins1	69.9	0.86	42.3	13.4	14.2	14
2	Ins2	120.9	0.77	61.8	29.6	29.5	29
3	Ins3	165.1	0.63	82.1	41.8	41.2	41
4	Ins4	218.7	0.45	104.7	57.3	56.7	56
5	Ins5	275	0.5	130.6	72.1	72.3	71
6	Ins6	334.8	0.52	159.9	87.5	87.4	86
7	Ins7	401.1	0.66	194.4	103.8	102.9	101
8	Ins8	470.5	0.74	235	118.3	117.2	116
9	Ins9	544.8	0.85	282.2	131.8	130.8	129
10	Ins10	623.3	0.88	337.6	138.2	147.5	146
11	Ins11	710	0.92	392.3	159.6	158.1	157

Table 6

MTACC Test results with parameter $n_w = 14$

No.	Instance	$n_w = 14$					
		Convergence time (s)	Optimality gap	MILP master time (s)	MTACC separation time (s)	Total time of RG sub problems (s)	Number of OA iterations
1	Ins1	65.7	0.32	30.6	23.3	11.8	11
2	Ins2	111.7	0.38	46.2	38.9	26.6	26
3	Ins3	155.9	0.42	62.9	55.6	37.4	37
4	Ins4	208.7	0.48	81.6	73.3	53.8	52
5	Ins5	263.8	0.55	103.2	92.9	67.7	66
6	Ins6	325	0.58	129	113.7	82.3	81
7	Ins7	391	0.63	159.6	135.3	96.1	95
8	Ins8	466.7	0.75	196.2	158.9	111.6	110
9	Ins9	546.9	0.85	239.9	183.6	123.4	122
10	Ins10	641.3	0.95	291.7	209.4	140.2	139
11	Ins11	727.8	0.82	342.5	235.2	150.1	149

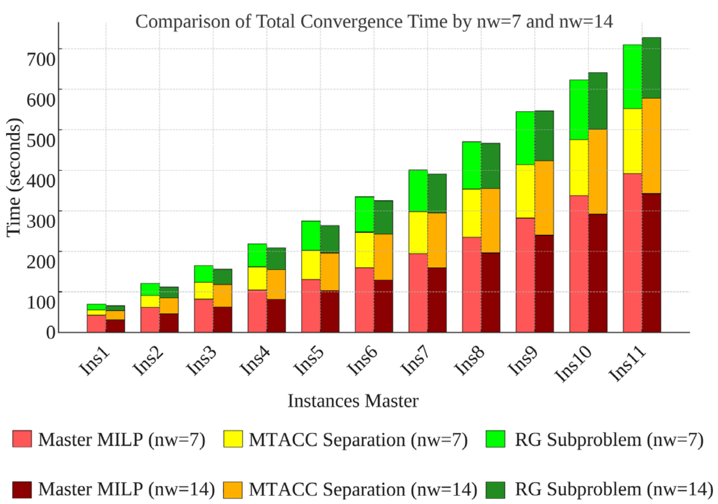

Fig. 3. Comparison of convergence time between parameter n_w

The longer separation time of MTACC at $n_w = 14$ is due to the longer time window processed (14 days or two weeks) being combined into a single piece (cover) for problems with longer scheduling periods, thus increasing computational complexity. This MTACC separation also impacts the global number of iterations.

From the perspective of the total number of global iterations (OA), the parameter $n_w = 14$ successfully reduced the number of OA iterations because it performed fewer MTACC cuts, which affected the OA iterations globally (Fig. 4). In larger instances like Ins10 and Ins11, changes in the n_w value have a significant impact on the number of iterations and computation time. For example, in Ins11, the number of iterations for $n_w = 7$ is 157, while for $n_w = 14$, it decreases to 149. This difference in the number of iterations shows that $n_w = 14$ is more efficient in reducing the number of iterations, even though it requires more time per iteration. The difference in the number of OA iterations for both parameters increases as the number of problems (instances) used in the testing increases, even though it comes at the cost of MTACC separation time.

From an optimality gap perspective, let's use detailed parameter data from 3 instances representing each class: Ins3 for the small class, Ins7 for the medium class, and Ins10 for the large class.

Fig. 5 shows a comparison of the optimality gap for the parameter n_w in Ins3. The rate of gap reduction at $n_w = 14$ is steeper in the first half of the iterations, reaching 84.34% in 18 iterations (with an average of 4.69% per iteration), compared to $n_w = 7$, which reached 76.33% in 20 iterations (with an average of 3.82% per iteration).

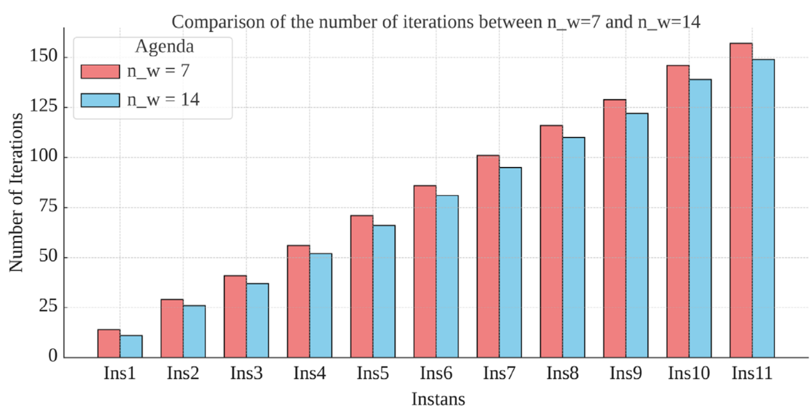

Fig. 4. Comparison of the number of iterations for the parameter n_w

In the last half of the iterations, the gap reduction for both n_w parameters shows a more stable rate. At $n_w = 14$, the reduction rate is 14.61% in 19 iterations, or an average of 0.77% per iteration, while at $n_w = 7$, the reduction rate is 22.16% in 21 iterations, or an average of 1.06% per iteration. Fig. 6 shows a comparison of the optimality gap for the parameter n_w in the medium class (Ins7). The difference in the rate of gap reduction between the two parameters, n_w, appears significant. The gap reduction for $n_w = 14$ is sharper in the first half of the iterations than it is for $n_w = 7$. At $n_w = 14$, the gap reduction reached 80.05% in just 47 iterations, or an average of 1.70% per iteration, while the gap reduction at $n_w = 7$ reached 64.22% in 50 iterations, or an average of 1.28% per iteration. However, for the last half of the iterations, the gap reduction speed for $n_w = 14$ was lower than for $n_w = 7$, at 18.96% in 48 iterations (an average of 0.39% per iteration) compared to 34.77% in 51 iterations (an average of 0.61% per iteration).

Fig. 7 shows a comparison of the optimality gap for the parameter n_w in the large class (Ins10). In the large class, the decrease in the optimality gap for both n_w parameters shows relatively the same speed. There is only a noticeable difference in the first 40 iterations, where the decrease in the gap for $n_w = 14$ is 66.59% (average 1.66% per iteration) and for $n_w = 7$ is 65.81% (average 1.65% per iteration). The remaining final iterations show a smooth decrease in speed, where $n_w = 14$ reaches 32.24% in 99 iterations (average 0.33% per iteration) and $n_w = 7$ reaches 33.14% in 106 iterations (average 0.31% per iteration).

Fig. 8 shows a comparison of convergence time versus the number of global iterations between the two OA-RG hybrids.

Hybrid OA-RG testing using MTACC ($n_w = 7$) and hybrid OA-RG without MTACC was also conducted to evaluate this method. This test aims to assess the differences in convergence time and the number of OA iterations, two parameters that are crucial in evaluating the efficiency and solution quality of the applied optimization method. Without additional cutting strategies, the OA-RG hybrid without MTACC relies solely on the optimality cut and feasibility cut available in OA. The hybrid with MTACC significantly reduces convergence time and the number of global iterations compared to the hybrid without MTACC. The MTACC evaluation was also conducted on other cutting plane methods, particularly regarding the total number of global iterations generated during the convergence process. Table 7 presents a comparison of the number of iterations derived from the computational test results.

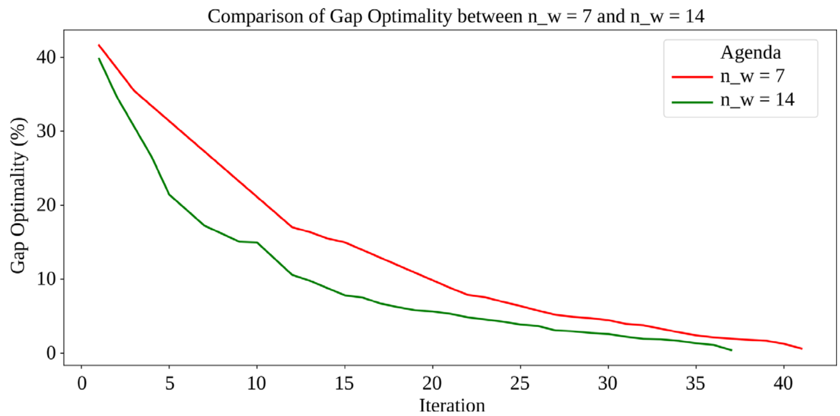

Fig. 5. Comparison of the optimality gap for parameter n_w in Ins3

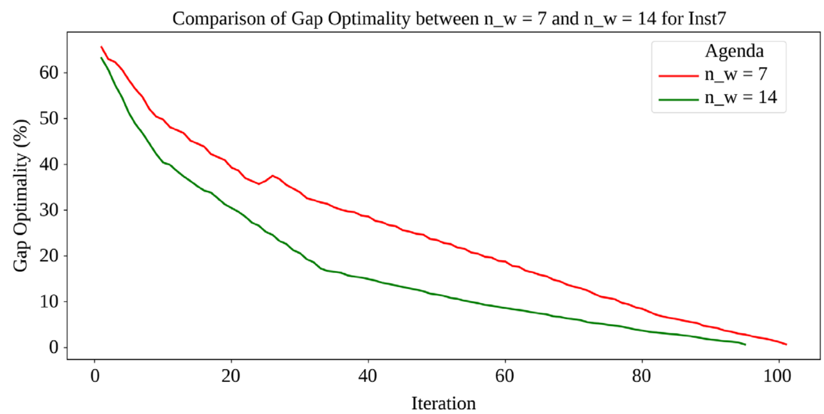

Fig. 6. Comparison of the optimality gap for parameter n_w in Ins7

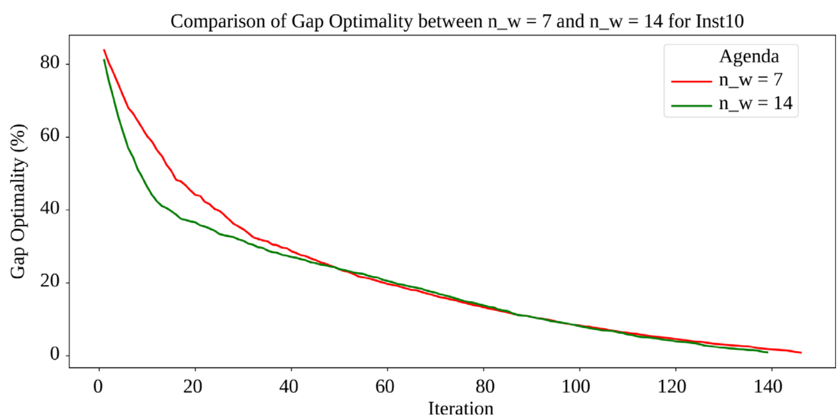

Fig. 7. Comparison of the optimality gap for parameter n_w in Ins10

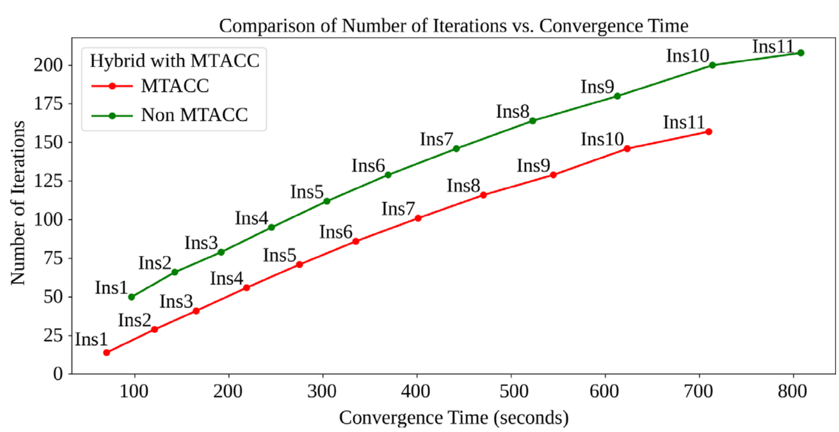

Fig. 8. Comparison of time vs. iterations

Table 7

Computational results: number of iterations

No.	Instance	MTACC ($n_w = 14$)	IRS cuts	Extended cutting plane	Disjunctive cuts	Perspective cuts	Aggregation cuts
1	Ins1	11	12	16	14	14	14
2	Ins2	26	27	35	30	29	30
3	Ins3	37	39	45	42	41	41
4	Ins4	52	54	62	58	58	57
5	Ins5	66	69	80	72	72	71
6	Ins6	81	84	95	89	89	88
7	Ins7	95	98	110	101	102	100
8	Ins8	110	115	130	122	122	119
9	Ins9	122	128	145	139	138	134
10	Ins10	139	145	165	155	155	151
11	Ins11	149	156	180	169	169	163

The number of iterations from these test results demonstrates MTACC's ability to handle the optimization of limited key resource allocation, namely labor, with daily capacity constraints and time-related constraints such as minimum frequency rules, proportional distance, and varying optimal time windows for each task to be performed.

5. 3. Visualization of an optimization model for an optimal plantation maintenance schedule

In this study use the area under the curve (AUC) to measure the performance of the n_w parameter by capturing the dynamics of the gap reduction over iterations. AUC is used as an aggregate measure of "total cumulative suboptimality" during the convergence process to summarize how large (and for how long) the optimality gap persists throughout the iterations.

Fig. 9 is a visualization of the maintenance schedule prototype, generated as the output of the MINLP model, in Gantt chart format. This Gantt chart displays the plantation area ID and the time horizon in weeks. Inside, maintenance tasks are arranged and color-coded, with the number of workers allocated to each task. The number of maintenance tasks displayed indicates the frequency of task execution.

Table 8 presents the results of the AUC analysis for each n_w parameter, with the average value calculated from the aggregation of the three instances (Ins3, Ins7, Ins10).

Analysis of the n_w value shows that increasing the time window from $n_w = 7$ to $n_w = 14$ has a significant impact on the pattern of decreasing optimality gap. The average AUC (abs) value decreased by approximately 21.6%, while the AUC per iteration decreased by approximately 19.9%.

This decline indicates that throughout the iteration process, the optimal gap trajectory was almost one-fifth lower at $n_w = 14$ compared to $n_w = 7$. This impact indicates an improvement in the efficiency of the convergence mechanism, where the optimization process not only rapidly approaches the optimal value but also maintains a smaller gap more stably throughout the iterations. The normalized AUC value also decreased by 18.6%, indicating that the superiority of $n_w = 14$ remained consistent even after accounting for differences in initial scale and iteration length. This indicates that the efficiency of $n_w = 14$ is not merely a result of data size or process length but truly stems from better algorithmic performance in controlling the optimality gap.

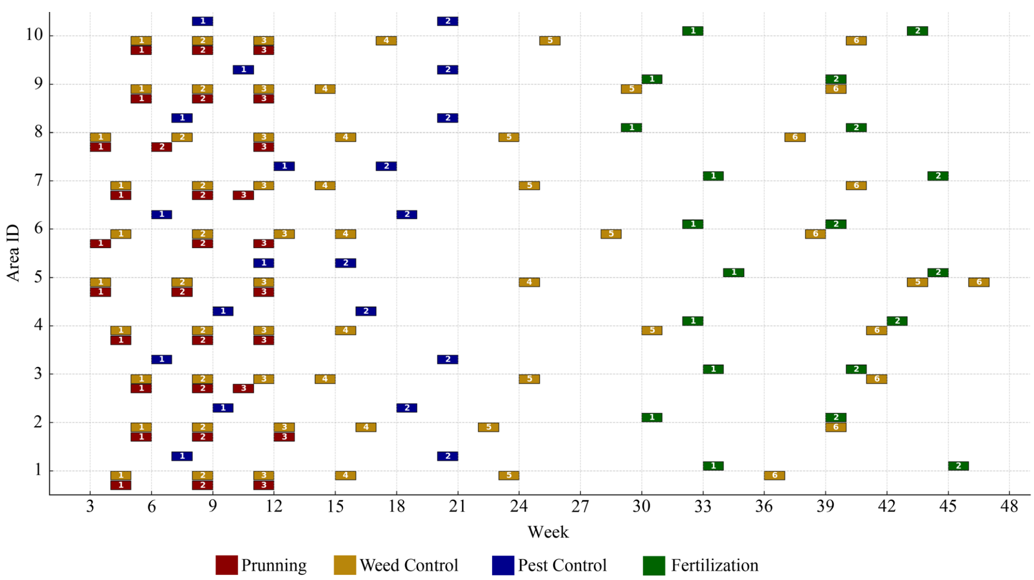

Fig. 9. Visualization of the maintenance schedule prototype

Table 8

AUC analysis for each n_w value

Metric	Average $n_w = 7$	Average $n_w = 14$	Δ% (14 vs 7)
AUC (abs)	2019.19	1583.91	−21.6
AUC per iteration	19.59	15.69	−19.9
AUC normalized	31.27	25.44	−18.6

6. Discussion of results of the hybrid method utilizing multi-row time-aggregated cover cuts

This study resulted in a maintenance schedule for smallholder coffee plantations with optimal resource allocation for the limited resources optimization problem within an acceptable computational time, using a hybrid approach of outer approximation (OA) and reduced gradient (RG) enhanced by multi-row time-aggregated cover cuts (MTACC).

The comparison results of the parameter values n_w in Tables 5, 6 show that the MTACC separation time significantly contributes to the total time, especially for larger problem scales. When n_w changes from 7 to 14, the average time for the MILP master decreases by approximately 6.16%. This analysis contrasts with the study in [19], which employs large-scale problem-solving methods such as aggregation and disjunctions, resulting in a very large and complex master problem and causing numerical issues. MTACC introduces a separation time influenced by the parameter nw. Comparing the parameter n_w across each problem scale (Ins1 to Ins11) reveals the impact of separation time on the overall total time (convergence time), particularly the time to solve the MILP master and subproblems, as shown in Fig. 3. This indicates a trade-off between the efficiency of the MILP formulation and the complexity of MTACC. The larger the value of n_w, the more temporal information MTACC must handle, which improves the model but slows down constraint separation. The average separation time increased by 36.65%, while the contribution of separation time to the Master MILP increased by 35%, and the contribution of separation time to convergence time increased by 10.10%. This data confirms that MTACC becomes an increasingly dominant component as the n_w parameter increases (from 7 to 14), with more "computational cost" spent on the separation process (cutting plane separation). The change also impacts the number of global iterations (representing the number of MTACC pieces), which decreases as shown in Fig. 4. This analysis differs from the study [25], which increases the complexity of subproblems per iteration and still requires a global strategy when non-convexity occurs.

The comparison of the n_w parameter in MTACC has been proven to improve the quality of the optimality gap, with a sharper decrease in the gap in the first 30%–40% of iterations and a tighter decrease in the remaining iterations until convergence is reached, as shown in Fig. 5–7. Unlike previous research such as [16, 18, 21, 22, 27], MTACC has aggregation control parameters, which provide better reinforcement than simply processing constraints one by one; the result includes optimal gap reduction, which requires many cutting iterations to achieve convergence. Table 7 shows that the average number of iterations for other cutting methods compared to MTACC increased by 4.38% to 19.71%, with a standard deviation ranging from 104.41% to 117.18%. The increase indicates the instability of these methods' performance across instances.

Although hybrids without MTACC and classical OA, like the study in [12], did not produce separation time, their convergence time was greater with a larger number of iterations compared to using MTACC, as shown in Fig. 8. Quantitatively, MTACC is able to accelerate the average convergence time by 12.84% compared to MTACC without it. This decrease indicates that the cutting plane mechanism generated by MTACC successfully narrowed the solution space from the initial iteration, thus eliminating the need for the solver to explore irrelevant areas to reach the optimal point. The number of OA iterations also went down sharply, to 38.83%, which shows that the model with MTACC reached stability faster and needed fewer master relaxation updates.

The results of this study indicate that the MTACC-based optimization model can be effectively applied to optimize coffee plantation maintenance scheduling with limited resources, particularly in terms of labor allocation and daily capacity constraints. Based on the testing (Tables 5 and 6) with small- to medium-scale coffee plantation data (such as Ins1 to Ins5), this model shows a significant improvement in convergence time efficiency and a reduction in the number of iterations. On a larger scale, as reflected in Ins9 to Ins11, this model is also able to reduce labor allocation errors that can occur due to inaccuracies in manual scheduling.

The results of this research can be applied to real-world coffee plantation environments with hilly topography and land areas of up to hundreds of hectares. This model is suitable for use by small farmers or plantation managers who have low-specification computing devices such as smartphones, allowing it to be used as a decision support system based on simple computing devices. The application of this model can reduce operational costs by optimizing labor allocation according to daily capacity and more efficient task execution frequency rules, improve time efficiency in maintenance scheduling, which can speed up task completion time without sacrificing work quality, and improve the quality of harvest results by maintaining proportional distance rules between different tasks (e.g., pruning, fertilizing, and pest control) and ensuring tasks are performed at optimal times for the best results.

This study is limited to only testing performance in deterministic cases without considering labor productivity uncertainty or weather variations that could affect the validity of the optimal time window. This allows for further studies by integrating stochastic programming methods to accommodate uncertainty.

The disadvantages of this study stem from the MTACC formulation, which depends on the static parameter of the time window length (n_w), resulting in the truncation's effectiveness varying across institutions and failing to adapt dynamically to the iterations' dynamics. Therefore, further development could be directed toward the dynamic adaptation of MTACC parameters or considering adaptive mechanisms for selecting n_w (based on load density/violation) to address these disadvantages.

7. Conclusion

1. The development of this system's architecture demonstrates that the hybridization of outer approximation (OA) and reduced gradient (RG), enhanced by multi-row time-aggregated cover cuts (MTACC), results in an effective approach for addressing combinatorial optimization problems in complex coffee plantation maintenance. This hybrid architecture is designed to optimize the allocation of limited resources more efficiently through more structured problem decomposition and the use of time-aggregated-based cutting techniques, which improves accuracy in maintenance scheduling.

2. Performance evaluation of the feature value n_w MTACC shows an average decrease in master MILP time of up

to 6.16% and an increase in average contribution to convergence time of up to 10.10%. Comparative evaluation of MTACC with other cutting methods shows a significant reduction in the number of iterations of up to 38.83% and an increase in convergence speed of 12.84%. This indicates that MTACC is more efficient in optimizing computation time and producing faster and more stable solutions.

3. Model data visualization produces a maintenance schedule that displays the results of optimizing labor allocation on plantation land to perform maintenance tasks/stages without violating constraints accurately based on the test results of AUC (abs) ∇21.6%, AUC per iteration ∇19.9%, and normalized AUC value ∇18.6%.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study, whether financial, personal, authorship or otherwise, that could affect the study and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Manuscript has no associated data.

Use of artificial intelligence

The authors declare the use of generative AI in the research and preparation of the manuscript. Tasks delegated to generative AI tools under full human supervision: generation of methodological approaches or identification of viable models for the initial proposal for further testing by the authors during the study; visualizing the original author's data in the form of figures.

Declaration submitted by: *Eko Hariyanto.*

Authors' contributions

Eko Hariyanto: Conceptualization, Methodology, Software, Formal Analysis, Investigation, Resources, Data Curation, Writing Original Draft Preparation, Visualization; **Poltak Sihombing**: Methodology, Formal Analysis, Investigation, Writing Review and Editing; **Erna Budhiarti Nababan**: Methodology, Formal Analysis, Writing Review and Editing; **Sawaluddin Sawaluddin**: Methodology, Formal Analysis, Writing Review and Editing.

References

1. Tharatipyakul, A., Pongnumkul, S., Riansumrit, N., Kingchan, S., Pongnumkul, S. (2022). Blockchain-Based Traceability System From the Users' Perspective: A Case Study of Thai Coffee Supply Chain. IEEE Access, 10, 98783–98802. https://doi.org/10.1109/access.2022.3206860
2. Wulandari, S., Djufry, F., Villano, R. (2022). Coping Strategies of Smallholder Coffee Farmers under the COVID-19 Impact in Indonesia. Agriculture, 12 (5), 690. https://doi.org/10.3390/agriculture12050690
3. Parmawati, R., Risvita, W., Hakim, L., Rahmawati, N. O., Gunawan, F. K., Ashari, F. M., Haqqi, S. S. (2023). Sustainability Index of Robusta Coffee Plantation (Case Study: Wagir District Smallholder Coffee Plantation in Malang, Indonesia). International Journal of Design & Nature and Ecodynamics, 18 (2), 279–288. https://doi.org/10.18280/ijdne.180205
4. Hartoyo, A. P. P., Hutagalung, L., Kulsum, F., Sunkar, A., Herliyana, E. N., Hidayati, S. (2023). Species composition, structure, and management in gayo coffee-based agroforestry system. Jurnal Pengelolaan Sumberdaya Alam Dan Lingkungan (Journal of Natural Resources and Environmental Management), 13 (1), 37–48. https://doi.org/10.29244/jpsl.13.1.37-48
5. Jiang, Y., Chen, L., Fang, Y. (2018). Integrated Harvest and Distribution Scheduling with Time Windows of Perishable Agri-Products in One-Belt and One-Road Context. Sustainability, 10 (5), 1570. https://doi.org/10.3390/su10051570
6. Hajirad, I., Ahmadaali, K., Liaghat, A. (2025). Crop yield and water productivity modeling using nonlinear growth functions. Scientific Reports, 15 (1). https://doi.org/10.1038/s41598-025-16096-0
7. Abd El Baki, H. M., Fujimaki, H., Tokumoto, I., Saito, T. (2024). Optimization of irrigation scheduling using crop–water simulation, water pricing, and quantitative weather forecasts. Frontiers in Agronomy, 6. https://doi.org/10.3389/fagro.2024.1376231
8. Baghizadeh, K., Cheikhrouhou, N., Govindan, K., Ziyarati, M. (2021). Sustainable agriculture supply chain network design considering water-energy-food nexus using queuing system: A hybrid robust possibilistic programming. Natural Resource Modeling, 35 (1). https://doi.org/10.1111/nrm.12337
9. Baihaqi, A., Sofiana, U., Usman, M., Bagio, B. (2022). Risk analysis of arabica coffee supply chain in Aceh Tengah regency, Aceh Province, Indonesia. Coffee Science, 16, 1–9. https://doi.org/10.25186/.v16i.1984
10. Kronqvist, J., Bernal Neira, D. E., Grossmann, I. E. (2025). 50 years of mixed-integer nonlinear and disjunctive programming. European Journal of Operational Research. https://doi.org/10.1016/j.ejor.2025.07.016
11. Patty, S., Das, R., Mandal, D., Roy, P. K. (2025). Self-adaptive multi-population quadratic approximation guided jaya optimization applied to economic load dispatch problems with or without valve-point effects. Results in Control and Optimization, 19, 100543. https://doi.org/10.1016/j.rico.2025.100543
12. Liñán, D. A., Ricardez-Sandoval, L. A. (2025). Trends and perspectives in deterministic MINLP optimization for integrated planning, scheduling, control, and design of chemical processes. Reviews in Chemical Engineering, 41 (5), 451–472. https://doi.org/10.1515/revce-2024-0064
13. De Mauri, M., Gillis, J., Swevers, J., Pipeleers, G. (2020). A proximal-point outer approximation algorithm. Computational Optimization and Applications, 77 (3), 755–777. https://doi.org/10.1007/s10589-020-00216-9

14. Liu, Z. E., Long, W., Chen, Z., Littlefield, J., Jing, L., Ren, B. et al. (2024). A novel optimization framework for natural gas transportation pipeline networks based on deep reinforcement learning. Energy and AI, 18, 100434. https://doi.org/10.1016/j.egyai.2024.100434
15. Wu, O., Muts, P., Nowak, I., Hendrix, E. M. T. (2024). On the use of overlapping convex hull relaxations to solve nonconvex MINLPs. Journal of Global Optimization, 91 (2), 415–436. https://doi.org/10.1007/s10898-024-01376-2
16. Kohar, A., Jakhar, S. K., Agarwal, Y. K. (2023). Strong cutting planes for the capacitated multi-pickup and delivery problem with time windows. Transportation Research Part B: Methodological, 176, 102806. https://doi.org/10.1016/j.trb.2023.102806
17. Melo, W., Fampa, M., Raupp, F. (2020). Two linear approximation algorithms for convex mixed integer nonlinear programming. Annals of Operations Research, 316 (2), 1471–1491. https://doi.org/10.1007/s10479-020-03722-5
18. Javaloyes-Antón, J., Kronqvist, J., Caballero, J. A. (2022). Simulation-based optimization of distillation processes using an extended cutting plane algorithm. Computers & Chemical Engineering, 159, 107655. https://doi.org/10.1016/j.compchemeng.2021.107655
19. Muts, P., Nowak, I., Hendrix, E. M. T. (2020). The decomposition-based outer approximation algorithm for convex mixed-integer nonlinear programming. Journal of Global Optimization, 77 (1), 75–96. https://doi.org/10.1007/s10898-020-00888-x
20. Deza, A., Khalil, E. B. (2023). Machine Learning for Cutting Planes in Integer Programming: A Survey. Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 6592–6600. https://doi.org/10.24963/ijcai.2023/739
21. Kronqvist, J., Misener, R. (2020). A disjunctive cut strengthening technique for convex MINLP. Optimization and Engineering, 22 (3), 1315–1345. https://doi.org/10.1007/s11081-020-09551-6
22. Bestuzheva, K., Gleixner, A., Vigerske, S. (2023). A computational study of perspective cuts. Mathematical Programming Computation, 15 (4), 703–731. https://doi.org/10.1007/s12532-023-00246-4
23. Lundell, A., Kronqvist, J., Westerlund, T. (2022). The supporting hyperplane optimization toolkit for convex MINLP. Journal of Global Optimization, 84 (1), 1–41. https://doi.org/10.1007/s10898-022-01128-0
24. Vo, T., Baiou, M., Nguyen, V. H., Weng, P. (2025). Learning to Cut Generation in Branch-and-Cut Algorithms for Combinatorial Optimization. ACM Transactions on Evolutionary Learning and Optimization, 5 (3), 1–27. https://doi.org/10.1145/3728371
25. Peng, Z., Cao, K., Furman, K. C., Li, C., Grossmann, I. E., Neira, D. E. B. (2024). A Convexication-based Outer-approximation Method for Convex and Nonconvex MINLP. 34th European Symposium on Computer Aided Process Engineering / 15th International Symposium on Process Systems Engineering, 3211–3216. https://doi.org/10.1016/b978-0-443-28824-1.50536-6
26. Lundell, A., Kronqvist, J. (2021). Polyhedral approximation strategies for nonconvex mixed-integer nonlinear programming in SHOT. Journal of Global Optimization, 82 (4), 863–896. https://doi.org/10.1007/s10898-021-01006-1
27. Bodur, M., Del Pia, A., Dey, S. S., Molinaro, M., Pokutta, S. (2017). Aggregation-based cutting-planes for packing and covering integer programs. Mathematical Programming, 171 (1-2), 331–359. https://doi.org/10.1007/s10107-017-1192-x
28. Zhang, Y., Chang, R., Omrany, H., Zuo, J., Burry, J., Gu, N. (2025). Policy-gradient scheduling optimisation under multi-skill constraints: A comparative study on computational algorithms. Journal of Building Design and Environment. https://doi.org/10.70401/jbde.2025.0017