Computational Project

2105623 Optimization of Chemical Processes

Department of Chemical Engineering, Chulalongkorn University
Semester 1/2026 | Instructor: Assoc. Prof. Dr. Soorathep Kheawhom
Weight: 15 percent of the course grade | Primary outcome: CLO 4

---

1 Purpose and outcome mapping

The computational project is the capstone assessment of this course. A group takes an open engineering situation, writes a structured specification for it in the Week 6 format, translates that specification into a mathematical optimization model, implements the model in Pyomo, verifies that the implementation means what the specification says, solves it, and interprets what the numbers mean for a decision maker. Unlike the homework sets, the project supplies neither a data table nor a fixed model: deciding what to include and what to leave out is part of the assessed work. Drafting model code with an AI assistant is permitted and expected; what is graded is the specification behind the model and the evidence that the model is correct. The rubric of Section 7 carries the outcomes as follows.

- **CLO 1, rows R1 and R3, 27 points.** Write the specification and the algebraic model (decision vector x, objective $f(x)$, constraints $g(x) \leq 0$ and $h(x) = 0$), then produce the evidence that the implemented model agrees with it. Row R3 also evidences CLO 4.
- **CLO 2, row R4, 13 points.** Select and justify an algorithm and solver for the model class, and demonstrate optimality rather than assume it.
- **CLO 3, row R5, 18 points.** Sensitivity analysis, interpretation and trade-offs.
- **CLO 4 (primary vehicle), rows R2 and R8, 24 points.** Implement and solve the model computationally: data input, model modification, solution interpretation, reproducibility.
- **Communication, rows R6 and R7, 18 points.** Report and presentation, all outcomes.

The five entries partition the rubric total exactly: $27 + 13 + 18 + 24 + 18 = 100$ points.

2 Group formation

- Groups have **two or three members**. Groups of one are not permitted; a group of four requires written approval and a correspondingly larger scope.
- Groups are self-formed and registered in Week 4; students without a group at the end of Week 4 are assigned to one. One member is registered as correspondent, submits every deliverable, and is the contact for scheduling the presentation slot.
- All submitted material and the presentation are in English. Mixed Thai and international groups are encouraged.

- Two groups may not take the same topic number unless their systems, data and model classes differ substantially; topics are allocated on a first-approved basis at the Week 6 deadline.
- Every member must be able to explain the entire model. Division of labor is expected, division of understanding is not, and question time is used to check it.

3 Menu of project topics

Twelve topics are offered. Each spans material from at least two weeks of the course, and the week numbers in each model-class line refer to the 2026 topic map: modeling in Weeks 2 to 5, AI-assisted modeling and verification in Week 6, LP theory and duality in Weeks 7 and 8, nonlinear programming in Weeks 9 and 10, integer programming in Week 11, uncertainty in Week 12, and data-driven and AI-based methods in Week 13. Week 6 is not listed against individual topics because its verification discipline applies to all twelve. The stated model class is the expected outcome of a competent formulation; a group reaching a different but defensible class must justify the difference. Where a public source is named, the group must check that the data are current and cite them. Otherwise data may be synthesized, provided the generating assumptions, the units and the plausibility of the magnitudes are stated in the report.

T1. Multiperiod production planning for a multiproduct specialty plant A specialty chemicals site makes three to five grades on shared equipment and must commit a production, inventory and sales plan over a twelve-month horizon. Demand is seasonal, raw material prices drift, storage is limited, and changeovers consume capacity. The decision is how much of each grade to make, store and sell in each period, plus feedstock purchases. The natural formulation is a multiperiod **LP** in which inventory balances couple adjacent periods; minimum campaign lengths, setup costs or a discrete operating mode convert it to an **MILP**. Demand and price series may be adapted from published chemical market indices or from plant data the group can lawfully use, or synthesized on a stated basis. Extension: treat demand as uncertain, solve a two-stage stochastic program with recourse in sales, and report the value of the stochastic solution. **Model class:** LP, extending to MILP or two-stage stochastic LP. Weeks 2, 5, 11, 12.

T2. Supply chain and facility location for battery materials A regional producer of cathode active material must decide where to site precursor and calcination plants and how to route material from mines and refineries to cell assembly plants. Candidate sites differ in fixed capital cost, capacity and unit conversion cost; transport cost depends on lane distance and mode. The decisions are the binary build decision for each site and capacity level, and the continuous flow on every lane. The model class is **MILP**, a multi-echelon capacitated facility location problem, optionally with single-sourcing constraints. Lane distances can come from public map services, capital cost intensities from published techno-economic assessments, and demand from announced cell plant capacities; missing quantities may be synthesized on a stated basis. Extension: impose a carbon price or a per-lane emissions budget and trace the cost versus emissions frontier by the epsilon-

constraint method. **Model class:** MILP. Weeks 3, 5, 11.

T3. Refinery blending and pooling A refinery blends naphtha, reformate and cracked streams into two or three gasoline grades meeting octane, sulfur, Reid vapor pressure and aromatics specifications. When streams mix in shared tanks before final blending, each pool quality becomes a variable and the component balances contain bilinear terms. The decision is the flow of every component into every pool and every final blend, and hence the pool qualities. The model class is a nonconvex **NLP**, the classical pooling problem; a variant with discrete recipes or the option to bypass a pool is an **MINLP**. Property data may be adapted from public assay tables or from worked examples in Edgar, Himmelblau and Lasdon; synthesized data are acceptable if property ranges are plausible and blending indices are used where linear blending fails. Extension: compare the p-formulation and the q-formulation from many starting points and bound the global optimum with a convex relaxation. **Model class:** Nonconvex NLP, extending to MINLP. Weeks 4, 10, 13.

T4. Heat exchanger network targeting and utility minimization A process with four to eight hot and cold streams needs a minimum utility target and a network reaching it. Given supply and target temperatures, heat capacity flow rates and a minimum approach temperature, the minimum utility duties follow from the transshipment model, an **LP** over the heat cascaded between temperature intervals. The targeting decision is the interval heat flow; the synthesis decision is which stream matches exist and how much area each exchanger carries, an **MILP** for match selection and an **MINLP** once capital cost enters the objective. Stream data may be taken from the pinch analysis literature or synthesized; the group must state the approach temperature, the film coefficients and the utility prices. Extension: sweep the approach temperature, build the energy versus capital trade-off curve, and locate the economic optimum and its sensitivity to fuel price. **Model class:** LP for targeting, MILP for match selection, MINLP with capital cost. Weeks 3, 5, 11.

T5. Reactor design and optimal operating conditions A reaction system with a desired product and a competing side reaction is to be operated for maximum profit or selectivity. Typical settings are a jacketed CSTR or a plug flow reactor carrying an exothermic reaction in which temperature accelerates both the wanted and the unwanted path. The decisions are reactor volume, feed composition and flow, coolant temperature and, for a plug flow reactor, the axial temperature profile. Arrhenius rates and nonlinear mole balances make the model an **NLP**; discretizing the profile enlarges it, and a choice of reactor type gives an **MINLP**. Kinetic parameters may come from reaction engineering texts or public kinetics databases, or be synthesized with stated activation energies. Extension: derive the KKT conditions by hand for a reduced two-variable version, check the constraint qualification, and confirm the solver multipliers. **Model class:** NLP, extending to MINLP. Weeks 9, 10, 11.

T6. Parameter estimation for reaction kinetics or a battery equivalent-circuit model A kinetic mechanism or a battery equivalent-circuit model is fitted to measured data. In the kinetic case the group estimates pre-exponential factors, activation energies

and reaction orders from concentration versus time data at several temperatures. In the battery case it fits the series resistance and the RC time constants of a first or second order Thevenin model to a pulse test or drive-cycle voltage response. The decision variables are the parameters, the objective is weighted least squares or maximum likelihood, and the constraints are the discretized dynamics. The class is a constrained **NLP**, frequently nonconvex with several local minima. Public battery cycling datasets and published kinetic studies are suitable sources; synthetic data from a known parameter set corrupted with noise are encouraged, because the true answer is then known. Extension: report confidence intervals from the covariance of the estimate and discuss identifiability. **Model class:** Nonconvex NLP, nonlinear least squares. Weeks 9, 10, 13.

T7. Short-term scheduling of a multiproduct batch plant A batch plant with several stages and shared units must sequence customer orders to minimize makespan, total tardiness or cost. Each order visits the stages in a fixed route, unit capacities and processing times differ by product, and the plant may run a zero-wait or finite intermediate storage policy with sequence-dependent cleaning times. The decisions are the assignment of each batch to a unit, the sequence within each unit and the start times. The class is **MILP** on a continuous time axis with precedence binaries or on a discrete time grid; a state task network formulation is an acceptable alternative. Processing time and changeover data may be adapted from published benchmark instances or synthesized. The instance must be small enough to close the gap, or the gap must be reported honestly. Extension: compare the continuous and discrete time formulations on one instance in time, node count and final gap. **Model class:** MILP. Weeks 5, 11.

T8. Dispatch of a battery energy storage system under a time-of-use tariff A commercial or industrial site operates a battery energy storage system alongside a metered load and, optionally, a rooftop photovoltaic array. Under a time-of-use tariff with peak, partial-peak and off-peak energy prices plus a monthly demand charge, the operator chooses the charge and discharge power in every interval of a representative horizon, subject to state-of-charge limits, power ratings and round-trip efficiency. With constant efficiency and no bar on simultaneous charging and discharging the model is an **LP**; mutual exclusivity, a minimum dispatch block or a demand ratchet requires binaries and gives an **MILP**. Tariff schedules are published by the Metropolitan and Provincial Electricity Authorities in Thailand and by most utilities elsewhere; load and generation profiles may be synthesized from typical daily shapes. Extension: add degradation as a throughput or depth-of-discharge penalty, or treat load and generation as uncertain in a two-stage stochastic dispatch. **Model class:** LP, extending to MILP or two-stage stochastic program. Weeks 2, 5, 11, 12.

T9. Hydrogen production and distribution network design A regional plan for low-carbon hydrogen must decide which production technologies to build, at what scale and location, and how to move the product to industrial and mobility demand by pipeline or tube trailer. Candidate technologies include alkaline and PEM electrolysis on a variable renewable supply, and steam methane reforming with carbon capture. The decisions are

the binary build decisions, the installed capacity of each unit, and the seasonal or hourly production, storage and transport flows. Capital cost scaling with capacity to a power below one, and load-dependent electrolyzer efficiency, make the model an **MINLP**; piecewise linearization of those curves reduces it to a tractable **MILP**. Cost and efficiency data are available from public agency and academic techno-economic assessments; synthesized values on a stated basis are acceptable. Extension: co-optimize against an electricity price signal and report the levelized cost of hydrogen against renewable capacity factor. **Model class:** MINLP, reducible to MILP by piecewise linearization. Weeks 3, 5, 10, 11.

T10. Design of experiments and surrogate-based optimization of an expensive simulation An expensive simulation, for example a CFD run, a rigorous flowsheet or a physics-based battery model, is to be optimized when one evaluation costs minutes to hours and derivatives are unavailable. The decisions are the variables passed to the simulation and, at each iteration, where to sample next. The workflow is a design of experiments (Latin hypercube or D-optimal), the fit of a surrogate such as a Gaussian process or a quadratic response surface, and optimization on the surrogate under an infill criterion such as expected improvement. The class is **derivative-free, surrogate-based global optimization** with an inner NLP or MINLP subproblem per iteration. The simulation is deterministic here, which distinguishes this topic from T11. A cheap analytic test function standing in for the simulation is preferred for verification, since the true optimum is then known. Extension: compare the surrogate-based search against multistart, simulated annealing or a genetic algorithm under an equal evaluation budget. **Model class:** Derivative-free, surrogate-based global optimization, inner NLP or MINLP. Weeks 9, 13.

T11. Bayesian optimization of a laboratory experimental campaign A research group must find operating conditions for a physical experiment in which every run costs a day of laboratory time and a batch of material: the calendering schedule of a battery electrode, the additive package of a vanadium electrolyte, or a catalytic microreactor. The decision is twofold, which experiment to run next and which recipe to recommend once the budget is spent. The workflow is an initial design of experiments (Latin hypercube or D-optimal), a Gaussian process fitted to noisy measurements, and maximization of an infill criterion such as expected improvement, constrained when a specification must also hold. The class is **Bayesian optimization**, a sequential surrogate-based search whose inner problem is a small multistart NLP. Unlike T10 the observations are noisy and the budget is fixed at 15 to 30 runs, so the replication policy is itself a decision. Data may come from a published experimental dataset, or from a simulator with added noise standing in for the rig, which keeps the true optimum known. Extension: compare single-point against batch selection of four experiments under one budget. **Model class:** Bayesian optimization with a Gaussian process surrogate, inner multistart NLP. Weeks 9, 13.

T12. A machine learning model embedded inside an optimization problem A unit whose behavior is known only from data must be placed inside a decision model. Two variants are offered, and a group takes one. In **variant A** a data-driven surrogate of a unit operation, for example electrolyzer efficiency against load and stack temperature,

or crystallizer yield against supersaturation, is trained offline and then enters a plant-level model as a constraint. The decision is the operating plan; a trained ReLU network enters exactly as an **MILP** through big-M or indicator encoding, while a smooth surrogate (quadratic, radial basis, Gaussian process mean) enters as an **NLP**. In **variant B** one system is solved twice: an exact dynamic program as in Week 12, and a learned policy trained by reinforcement learning, compared on identical instances by expected cost, worst case and computing time. Training data may be generated from a mechanistic simulator or taken from public process and battery cycling datasets, with the split and the validation error reported. Extension: quantify how far the surrogate moves the optimum by re-evaluating the mechanistic model there. **Model class:** MILP with an embedded ReLU network, or NLP with a smooth surrogate; variant B compares exact dynamic programming against a learned policy. Weeks 5, 11, 12, 13.

4 Proposing your own topic

A group may propose a topic that is not on the menu, including one drawn from its own thesis or an industrial internship. Self-proposed topics are submitted on the Week 6 form and require written approval before further work counts toward the grade. Approval is granted when all six criteria are met.

C1. Decision content. A genuine decision under constraints, not a simulation, a literature survey or a data-analysis exercise with an optimization label attached.

C2. Course relevance. Exercises material from at least two weeks of the 15-week map, in a chemical, process, energy or energy-storage setting.

C3. Model class. The expected class (LP, MILP, NLP, MINLP or stochastic) is identified in advance, with a stated reason.

C4. Tractability. The instance is solvable to optimality, or to a reported gap, with freely available solvers on a laptop in a few minutes.

C5. Data availability. A source is named, or a synthesis plan with stated assumptions is given. Confidential industrial data must be scaled or anonymized, with permission confirmed.

C6. Non-duplication. The work is new for this course. Work submitted elsewhere may be background, not the deliverable, and any overlap must be declared on the form.

A proposal failing one criterion is returned once with comments and may be resubmitted within seven days; a group whose topic is still not approved takes a menu topic and keeps the Week 9 date.

5 Required deliverables

Five items, submitted by the correspondent through the course management system. Late submission costs 10 percent of that item's rubric rows per calendar day, to five days maximum, after which it scores zero.

D1. Technical report, four pages

Four pages maximum in `Project_Report_Template.tex`, 12 pt, single column, 1 in margins. References, nomenclature, the AI and verification disclosure of Section 9 and up to two pages of appendix fall outside the limit; the appendix is read at the grader's discretion and may not carry results the main text needs. Structure and target lengths: (1) problem statement and scope, half a page, naming the system, the decision maker, the decision, and what is deliberately excluded; (2) mathematical formulation, one page, with sets, parameters and units, variables, $f(x)$, $g(x) \leq 0$, $h(x) = 0$, and the model class with its reason; (3) data and assumptions, half a page, every value with its source or synthesis rule and its units; (4) solution approach, a quarter page, with solver, algorithm, model size, termination criteria and run time; (5) results and model verification, one page, including the summary table of the test suite required by D3; (6) sensitivity and interpretation, three quarters of a page; (7) conclusions and limitations, a quarter page; (8) disclosure, references and nomenclature, outside the limit.

D2. Model code

A runnable repository or notebook holding the model, the data and the scripts that produce every number and figure in the report.

- Pyomo is the default. GAMS, JuMP, AMPL or a documented direct solver interface is acceptable if the formulation stays readable.
- Named `Set`, `Param`, `Var`, `Constraint` and `Objective` components, data separated from model logic in a CSV, JSON or clearly marked data block, and no hard-coded numbers inside constraint rules.
- An explicit check of the solver termination condition after every solve, as in the course notebooks, and figures produced by the code using the Teal-Amber Lab Palette v1.0.
- Total run time under five minutes on a standard laptop, or a documented reason why not.

D3. Model test suite

A script or notebook, `test_model.py` or `tests.ipynb`, that runs unattended, prints a pass or fail line per test, and exits nonzero on any failure. It is built in the style of Week 6, Section 5, and the minimum content is fixed.

- **At least three degenerate-instance tests with closed-form expected answers.** Each test constructs an instance whose optimum is derivable by hand (zero demand, a single period, a single unlimited resource, a symmetric instance, all costs equal), states the closed form in a comment, and asserts agreement to a stated tolerance. The expected value must be computed by arithmetic that does not build the model under test: a test that reuses the code it is testing proves nothing.
- **A monotonicity check.** Sweep one capacity, budget or demand parameter over a stated range and assert the direction that theory requires: for a minimization, relaxing a constraint never raises the optimum and tightening it never lowers it.

Report the sweep as one figure. A sweep in which the optimum does not move at all is reported and explained, since a flat response is the signature of a parameter wired to nothing.

- **A units audit.** Every parameter, variable and objective term carries a declared unit, and the audit asserts that both sides of every constraint and every term of the objective reduce to the same unit vector. The Week 6 `check_dimensions` helper may be reused with attribution.
- Tests are written from the specification, not from the code, and the suite runs in under one minute. Groups may add the shadow-price plausibility and constraint-activity checks of Week 6 for credit under row R3.

D4. Presentation

Twelve minutes plus three minutes of questions, in Week 14 or Week 15. Every member speaks. Suggested allocation: two minutes on the problem and why it matters, three on the formulation, two on implementation and verification, four on results and sensitivity, one on conclusions. The chair stops the talk at twelve minutes; material not reached is not graded. Slides are submitted as PDF before the session.

D5. Reproducibility statement

One page submitted with the code, stating: operating system and Python (or equivalent) version; solver name and version; the exact command that reproduces every reported result and the exact command that runs the test suite of D3; expected run time; the random seed for any stochastic component; the AI and verification disclosure required by Section 9; and a table mapping each figure and table to the script or notebook cell that generates it. A grader following the statement on a clean machine must obtain the reported objective value to within the stated tolerance and must see the test suite pass.

6 Milestones

The weeks are unchanged from previous years, but what is expected at each one has moved forward with the teaching order. All four modeling weeks and the Week 6 workshop precede the proposal, so the proposal is a specification, not a sketch. The verification toolkit is taught in Week 6, so the test suite is expected to exist by Week 9 and to be complete by Week 12, not assembled in the final week.

Week	Milestone	Submission and treatment
Week 4	Group registration	Names, student numbers, and two or three candidate topic numbers with one line of reason each. Weeks 2 to 4 have covered linear, network and blending models, so the candidates should already name a decision and a likely model class. Not graded.
Week 6	**Proposal due**	`Project_Proposal_Form.tex`, two pages, carrying a **structured specification** in the Week 6 template format: scope and decision, sets and indices, parameters with units and sources, variables with units, domains and bounds, objective with units and sense, constraints numbered in words, and assumptions stated explicitly. Also the planned verification tests and the intended use of AI assistance. Returned as approved, approved with comments, or resubmit within seven days. Feeds rubric row R1.
Week 9	**Formulation checkpoint**	The specification turned into algebra: complete model, one to two pages, with model size and intended solver, plus a running units audit and at least one passing degenerate-instance test. Discussed in a fifteen-minute meeting. Feedback only, but an absent or unusable submission caps R1 at the adequate band.
Week 12	**Preliminary results**	Two pages: a running model, one solved instance, objective value, solver status, model size and one figure, together with the **complete test suite of D3** and its output (three degenerate tests, the monotonicity sweep, the units audit). Confirms the scope is feasible and that the model is doing what the specification says.
Weeks 14 to 15	**Presentations**	Twelve minutes plus three of questions, one slide of which reports the verification evidence. Slot order drawn at the end of Week 12; slides submitted as PDF before the session.
End of Week 15	**Final submission**	Report including the disclosure section, code, test suite, reproducibility statement, and one peer-evaluation form per student, submitted individually and treated confidentially.

Missing the Week 6 or the Week 12 milestone without prior arrangement costs 5 points from the rubric total per milestone.

7 Grading rubric

The rubric totals 100 points: 15+18+12+13+18+8+10+6 = 100. A grader assigns the band first, then a value inside the band; intermediate values are used only when the work is genuinely between two descriptors. Instructor and teaching assistant apply the same rubric, and any row on which they differ by more than one band is resolved by discussion before grades are released. Row R3 is new in 2026 and carries the verification discipline of Week 6; the points for it were taken from report quality, presentation, reproducibility and, in smaller measure, from rows R2, R4 and R5.

Criterion	Pts	Excellent	Adequate	Insufficient
R1. Specification and problem formulation (CLO 1)	15	**13 to 15.** Scope stated with the decision maker named. Sets, parameters and variables defined with units, domains and bounds, as in the Week 6 specification template. Objective and every constraint written in words and then algebraically, each justified in one sentence. Assumptions and exclusions deliberate and stated. Degrees of freedom consistent with the system described.	**9 to 12.** Model complete and readable, but at least one of: units missing or inconsistent, one or two constraints asserted without justification, assumptions left implicit, or a scope statement vague about what is excluded.	**0 to 8.** Model incomplete, contradictory, or copied without adaptation. Variables or constraints in the code absent from the report. Objective does not match the stated decision.
R2. Model correctness and implementation (CLO 4)	18	**15 to 18.** Code reproduces the algebraic model line for line, with named `Set`, `Param`, `Var`, `Constraint`, `Objective`. Data separated from logic. Termination condition asserted. Model size reported. Runs clean in a fresh environment.	**10 to 14.** Code solves the intended problem but shows hard-coded constants inside rules, weak naming, no separation of data and model, or a missing status check. Report and code differ only in ways that do not change the answer.	**0 to 9.** Code does not run, does not match the reported model, solves a different problem than described, or cannot produce the reported result. An AI-drafted model submitted without the test suite of D3 scores zero on this row whatever the numbers look like.
R3. Model verification (CLO 1, with CLO 4)	12	**10 to 12.** Test suite runs unattended and passes. Three or more degenerate instances with closed-form answers, each derived independently of the model code. Monotonicity sweep reported with its figure and the expected direction argued. Units audit covers every constraint and the objective. The disclosure names what was AI-drafted and which test validated it, and any test that failed is reported with the correction it forced.	**6 to 9.** Suite present but thin: fewer than three degenerate instances, an expected answer computed with the model under test, a monotonicity check on an inactive parameter, a units audit by inspection rather than by code, or a disclosure that lists tools without linking them to tests.	**0 to 5.** No test suite, tests that assert nothing, tests that do not run, or a disclosure that is absent or contradicted by the code history. A generated model presented as verified when it is not is handled under Section 9.
R4. Solution quality and optimality (CLO 2)	13	**11 to 13.** Optimality demonstrated, not assumed: gap reported for MILP, several starting points for nonconvex NLP, and at least one independent check beyond the D3 suite (hand calculation on a reduced instance, a second solver, or a bound). Infeasible or unbounded outcomes diagnosed.	**7 to 10.** Solution obtained and solver status reported, but the independent check is weak, or the gap is not reported, or a nonconvex problem is solved from one starting point without comment.	**0 to 6.** Solver output accepted at face value, no status check, or physically impossible numbers (negative flows, violated balances) pass unnoticed.

Criterion	Pts	Excellent	Adequate	Insufficient
R5. Analysis and sensitivity (CLO 3)	18	**15 to 18.** Shadow prices, reduced costs or parametric sweeps computed and read correctly in engineering units. Active constraints identified. At least one trade-off curve or scenario comparison. Conclusions state what a decision maker should change, by how much, and over what range the statement holds.	**11 to 14.** Sensitivity performed but reported mechanically: correct numbers with thin interpretation, no distinction between binding and nonbinding constraints, or a sweep whose range is not justified.	**0 to 10.** No sensitivity analysis, or shadow prices quoted with the wrong units or the wrong sign, or conclusions that do not follow from the reported numbers.
R6. Report quality	8	**7 to 8.** Within the page limit, follows the required structure, notation consistent with the course, every figure and table captioned and referenced, sources cited, nomenclature complete, US English, clean typography.	**5 to 6.** Readable and complete, but with structural drift, an unreferenced figure, inconsistent notation, incomplete nomenclature, or an overrun up to half a page.	**0 to 4.** Sections missing, overrun beyond half a page, figures unreadable or unlabeled, sources absent, or an argument the reader cannot follow.
R7. Presentation	10	**9 to 10.** Finishes within twelve minutes. All members speak. Formulation and results legible from the back of the room. Questions answered directly and correctly by more than one member, including at least one question about a modeling choice.	**6 to 8.** Delivered within or slightly over time; content complete but unbalanced (for example most of the time on background), slides dense, or questions answered by one member only or only in part.	**0 to 5.** Stopped before the results, a member does not speak, slides unreadable, or answers reveal that the speaker does not understand the submitted model.
R8. Reproducibility	6	**5 to 6.** A grader following the statement on a clean machine reproduces every reported number within the stated tolerance at the first attempt and sees the D3 suite pass. Versions, seeds, disclosure and the figure-to-script map are complete.	**4.** Reproducible after minor unstated fixes (installing an undeclared package, correcting a path, changing solver). Statement present but incomplete.	**0 to 3.** Not reproducible, no statement, missing data files, or results depending on an environment that is never described.
Total	**100**	**Converted directly to the 15 percent course weight.**		

Adjustments after the rubric total. Missed Week 6 or Week 12 milestone, minus 5 points each. Late deliverable, minus 10 percent of that item's rows per calendar day, five days maximum. Peer-evaluation multiplier between 0.7 and 1.1 as in Section 8. Undisclosed use of an AI tool or of external code is handled under Section 9, not by rubric deduction.

Row values at a glance. R1 15, R2 18, R3 12, R4 13, R5 18, R6 8, R7 10, R8 6, total 100. By outcome: CLO 1 rows R1 and R3, 27; CLO 2 row R4, 13; CLO 3 row R5, 18; CLO 4 rows R2 and R8, 24; communication rows R6 and R7, 18.

8 Peer evaluation of intra-group contribution

Each student submits one form individually and confidentially at the end of Week 15, covering every member including the student. Forms are not shared with the group, and the instructor takes the mean of the scores awarded to each student by the others. A student within the normal range receives the group score unchanged; corroborated persistent under-contribution gives a multiplier as low as 0.7, and corroborated exceptional contribution gives up to 1.1. A single outlying evaluation never changes a grade on its

own: the instructor interviews the group before applying any multiplier below 1.0. Failing to submit the form forfeits any upward adjustment.

Form. Score every member, yourself included, from 1 (did not contribute) to 5 (contributed fully and reliably), then allocate a contribution percentage summing to 100.

Category (1 to 5)	**Self**	**Mem. 2**	**Mem. 3**	**Mem. 4**
P1. Problem definition and formulation				
P2. Model implementation, testing and debugging				
P3. Data collection and the model test suite				
P4. Analysis, sensitivity and figures				
P5. Report writing and editing				
P6. Reliability: meetings, deadlines, responsiveness				
Contribution share, percent (total 100)				

Narrative (required, two to four sentences). Name one specific contribution by each other member, and state any difficulty the group met and how it was resolved. "Everyone worked hard" is not evidence and is disregarded.

Signature Date

9 Academic integrity and the use of AI tools

This is a graduate engineering course, and the standard applied is that of professional practice: the work you sign is the work you can defend, and every contribution that is not yours is attributed. Collaboration inside the group is unlimited; between groups it covers concepts, solver installation and debugging only, and sharing model files, data or report text between groups is plagiarism, as is reusing a previous year's submission. External code, published formulations, datasets and figures may be used where they save effort, provided each is cited at the point of use with a specific reference rather than a general acknowledgment.

Generative AI: permitted, expected, and graded on the verification

Week 6 teaches AI assistance as a modeling instrument, and this project applies that teaching. Using a large language model or a code assistant to draft the specification, to translate a specification into Pyomo, to propose constraints you may have forgotten, to review your formulation adversarially, or to write plotting and test code is **permitted and expected**. It is a normal professional practice and it is not misconduct. What is assessed is not whether a human typed the model but whether the group can show that

the model is right, because a generated model is a hypothesis and nothing more until it has been checked against dimensional consistency, degenerate instances with closed-form answers, monotonicity in the parameters the specification says are active, and a constraint-by-constraint reading against the numbered list in the specification.

Three rules follow, and they are enforced.

A1. Disclose. The report carries a dedicated disclosure section, outside the four-page limit, stating for each tool its name and version, which artifacts it drafted or edited (specification, algebraic model, Pyomo components by name, data synthesis, test code, figures, prose), and which test in the D3 suite validates each drafted artifact. Prose written or edited by a tool is disclosed as well. A general acknowledgment such as "AI was used for coding help" does not meet this requirement.

A2. Verify. Every generated artifact must be tied to a check that could have failed. An **unverified generated model scores zero on row R2**, the model-correctness row, no matter how plausible the objective value, how clean the code reads, or how confidently the report asserts the result. Numbers that look right are not evidence; a passing test written from the specification is.

A3. Own. The group is responsible for every line it submits. Fabricated data, fabricated citations, a disclosure contradicted by the file history, or results the group cannot reproduce and explain on request are academic misconduct and are referred to the Faculty under university regulations. Any member may be asked in an oral check to explain any line of the submitted model, to justify a constraint against the physics, or to state what a given test would have caught, and inability to do so is evidence in such a review.

10 Common pitfalls: self-audit before submission

Work through this list as a group in the week before the deadline. Every item is a deduction that graders apply routinely. Bring formulation questions to the Week 9 checkpoint or to office hours, and raise solver installation problems early: after Week 12 they excuse neither late nor incomplete work.

Formulation

☐ Every parameter carries a unit, and units balance in every constraint and the objective.

☐ The objective matches the decision stated in Section 1 of the report: minimizing cost while discussing selectivity is a mismatch. No constraint is implied by another, and no missing balance leaves the model unbounded.

☐ Binary variables are actually needed, and Big-M values are as tight as the physics allows, with the derivation stated: arbitrary constants give weak relaxations and wrong answers.

Implementation

☐ The termination condition is checked in code, not read off the screen, and the model class matches the solver: no MILP sent to a continuous NLP solver, no nonconvex NLP called globally optimal because a local solver converged.

☐ Index names in the code match those in the report, data live outside the constraint rules where a reader can check them, and the code runs from a clean directory with no state left

over from an interactive session.

Verification

☐ The test suite runs from a clean directory, prints one line per test and fails loudly: at least three degenerate instances with closed-form answers, a monotonicity sweep, and a units audit.

☐ Every expected answer is computed by arithmetic that does not build the model under test, and every test was written from the specification rather than from the code.

☐ The disclosure section names each AI tool, what it drafted, and the test that validates each drafted artifact. Nothing generated is submitted with no check attached to it.

Results and analysis

☐ The objective value is stated with units and a sensible number of significant figures: six decimals on a cost in baht is not precision.

☐ The optimality gap is reported for every MILP, several starting points are tried for every nonconvex NLP, and one independent check beyond the test suite is described precisely enough to repeat.

☐ Shadow prices carry the right sign and units and are quoted only over their valid range, and a degenerate or alternative optimum is not reported as unique.

Report, code and presentation

☐ Four pages or fewer of main text in the supplied template, nomenclature complete, and every figure and table referenced from the text.

☐ Every figure uses the Teal-Amber Lab Palette, carries axis labels with units, and is readable in grayscale.

☐ The reproducibility statement and the AI and verification disclosure are complete and tested on another machine, and the talk is rehearsed to finish inside twelve minutes.