

Optimization Problem Modeling

Before you can learn a decision, you must be able to state one

Jinkyoo Park · KAIST

The atom every later method is built from

Where we are — the simplest cell of the map

PREVIOUS STEP	THIS CHAPTER'S QUESTION
Lecture 0 identified a decision and its evidence.	Now make the objective and constraints explicit.

We begin with a known one-step model. Convexity will tell us when a local condition certifies a global answer.

Solve a bounded one-variable problem and check its KKT conditions.

Learning route — formulate, solve, and certify

Bring: Derivatives, vectors and basic algebra.

FIRST PASS	WHAT TO DO
Follow the idea	Formulate → recognize convexity → state and check optimality → inspect a local approximation
Work without the solution	Solve a bounded one-variable problem and check its KKT conditions.
Return later	Optimization layers, the two design case studies and full sensitivity derivations follow the main route.

For the temperature thread: **predict** → **calculate** → **reveal and check** → **change one condition**. Complete the core calculation before reading the research extensions.

— ACT 1

The standard form

Q1. One shape that every optimisation problem can be poured into.

The standard form — one shape for all of them

Q1 State it?

Q2 Solve it?

Q3 Certify it?

Q4 Convexify it?

Every mathematical optimisation problem can be written:

$$\min_{x \in D} f(x) \quad \text{s.t.} \quad g_i(x) \leq 0, \quad i = 1, \dots, m, \quad h_j(x) = 0, \quad j = 1, \dots, p$$

- $x \in \mathbb{R}^n$ — the decision vector; D — its allowed domain; $f(x)$ — the cost;
- g_i — inequality constraints; h_j — equality constraints;
- the **optimal value** $p^* = \inf\{f(x) : x \text{ feasible}\}$ — the lowest achievable cost, or the lower limit if no point attains it.

Two degenerate cases worth naming: $p^* = +\infty$ if the problem is *infeasible* (no x satisfies the constraints), and $p^* = -\infty$ if it is *unbounded below*. Both are usually signs of a modelling error, not a solver failure.

Maximising f is minimising $-f$; everything is phrased as minimisation without loss of generality.

A production decision, written all the way down

an illustrative two-product model

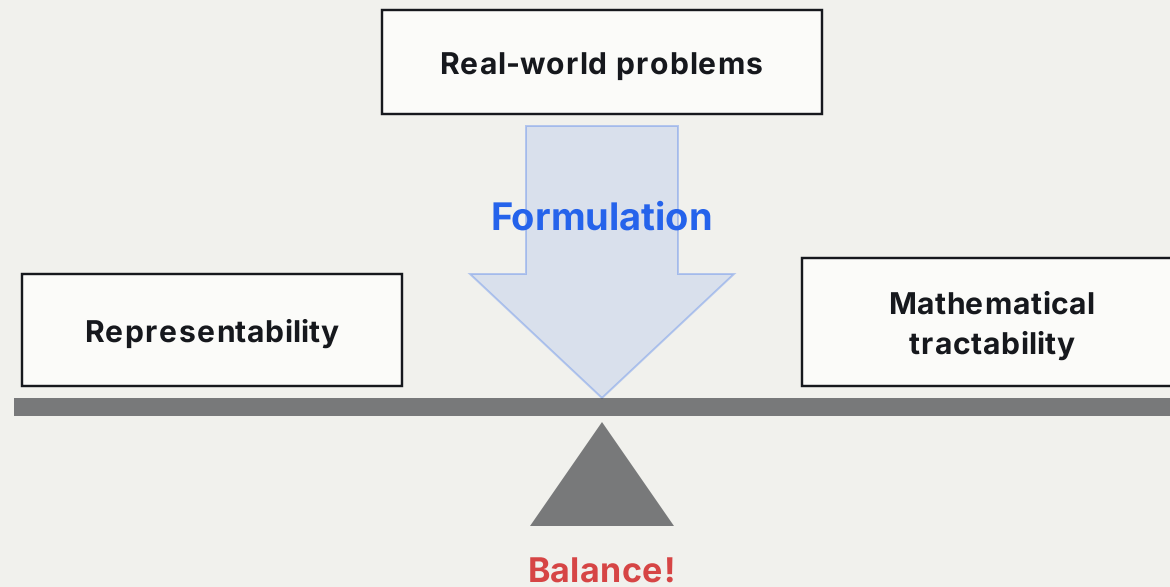
A workshop makes x_1 units of product A and x_2 of product B. Each unit earns 3 or 2, respectively. Machine time is limited to 4 units; product A uses two units of time and B uses one. Demand for A is at most one unit. Quantities are divisible.

$$\max_{x \geq 0} 3x_1 + 2x_2 \quad \text{s.t.} \quad 2x_1 + x_2 \leq 4, \quad x_1 \leq 1$$

In minimisation standard form, $f(x) = -3x_1 - 2x_2$, $g_1(x) = 2x_1 + x_2 - 4$ and $g_2(x) = x_1 - 1$. Non-negativity can be included in the domain $D = \mathbb{R}_+^2$ or written as $-x_i \leq 0$.

B earns 2 per unit of machine time; A earns only 1.5. Thus $x^* = (0, 4)$ earns 8. Indeed $3x_1 + 2x_2 = 2(2x_1 + x_2) - x_1 \leq 8$: **a feasible answer and a bound that proves it optimal.**

Formulation is a balance



A model that keeps every detail of the world cannot be solved; a model a solver loves may no longer be the problem you had. **Formulation is the act of balancing the two** — and it is where most of the real difficulty of applied decision making lives.

Equivalent problems — four rewrites worth knowing

The standard form is a shape, and most problems must be *put into* it. Four transformations do almost all of that work, and each leaves the optimal value unchanged.

REWRITE	FROM	TO
equality constraints	$f(\mathbf{A}_i x + \mathbf{b}_i)$ inside	$f(z_i)$ with $z_i = \mathbf{A}_i x + \mathbf{b}_i$
slack variables	$a_i^\top x \leq b_i$	$a_i^\top x + s_i = b_i, s \geq 0$
epigraph form	$\min_x f(x)$	$\min_{x,t} t$ s.t. $f(x) - t \leq 0$
minimise out a variable	$\min_{x_1, x_2} f(x_1, x_2)$	$\min_{x_1} \hat{f}(x_1), \hat{f} = \inf_{x_2} f$

The **epigraph form** is the one to remember: every convex problem can be written with a *linear* objective, because the difficulty pushes into the single constraint $f(x) \leq t$.

Check — putting it in the standard form

A plant requirement reads "*output must be at least 100 units.*" Written into the standard form $\min f$ s.t. $g(x) \leq 0$, the constraint becomes:

☒ A $100 - x \leq 0$

☐ B $x \geq 100$, left as it is — the form allows either direction

☐ C $x - 100 \leq 0$

☐ D It is an equality constraint, so it becomes $h(x) = x - 100 = 0$

The standard form admits **one** inequality direction, so every "at least" is flipped by negating both sides. It is bookkeeping, but it is the bookkeeping that lets a single solver read every problem in the course — and getting the sign backwards silently optimises the opposite plant.

— ACT 2

Convexity, the watershed

Q2. When does a local solution settle the global question?

Convexity — the line between easy and hard

Q1 State it?

Q2 Solve it?

Q3 Certify it?

Q4 Convexify it?

A problem is **convex** when f is a convex function and the feasible set is a convex set — g_i convex, h_j affine.

WHY CONVEXITY IS THE DIVIDING LINE

Local minimum: no better feasible point nearby. **Global minimum:** no better feasible point anywhere. For a convex problem, **every local minimum is global**.

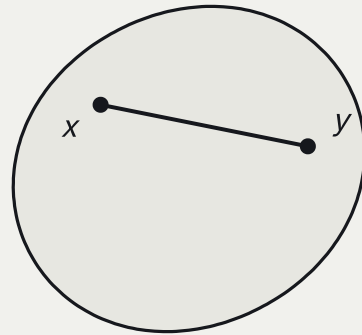
That fact gives local search a global meaning, **provided a solution exists and the algorithm reaches it:**

- converged local minima cannot be worse than a hidden global minimum;
- many standard convex classes admit reliable, efficient solvers;
- KKT conditions can certify a global optimum; their necessity needs regularity;
- duality supplies lower bounds and, under suitable conditions, exact certificates.

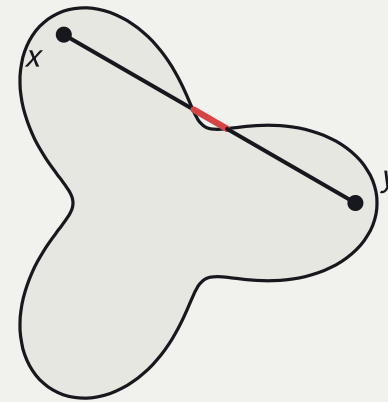
Convexity alone does not guarantee existence or convergence. The convex function e^x has infimum zero but no minimiser on $\mathbb{R}$. For $f(x) = x^2$, gradient descent gives $x_{k+1} = (1 - 2\alpha)x_k$ and diverges from nonzero starts when $\alpha > 1$. Step sizes, conditioning and stopping tolerances still matter.

Convex sets — the segment test

Non-convex problems enjoy none of those guarantees for free, so much of practical optimisation is the art of **getting a convex problem** — or a sequence of them. And both halves of one are the same test.



convex
every chord stays inside



not convex
this chord leaves the set

A set is convex when the segment joining any two of its points stays inside it. A function is convex when the segment joining any two points of its graph stays *above* it — equivalently, when its epigraph is a convex set. **Both halves of a convex problem are this one test.**

The same convexity, seen through the gradient

the inequality the rest of this lecture rests on

The segment test is the definition. Differentiate it and convexity takes the form that actually does the work — for differentiable f :

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) \quad \text{for all } x, y$$

Read the right-hand side. That is the first-order Taylor expansion of f at x — the tangent plane. For a general function it says something about a *neighbourhood* of x and nothing beyond it. For a convex function the inequality holds **everywhere**: the tangent never rises above the graph, so a linearisation built from purely local information is a **global underestimator**.

WHY A LOCAL MINIMUM IS A GLOBAL MINIMUM

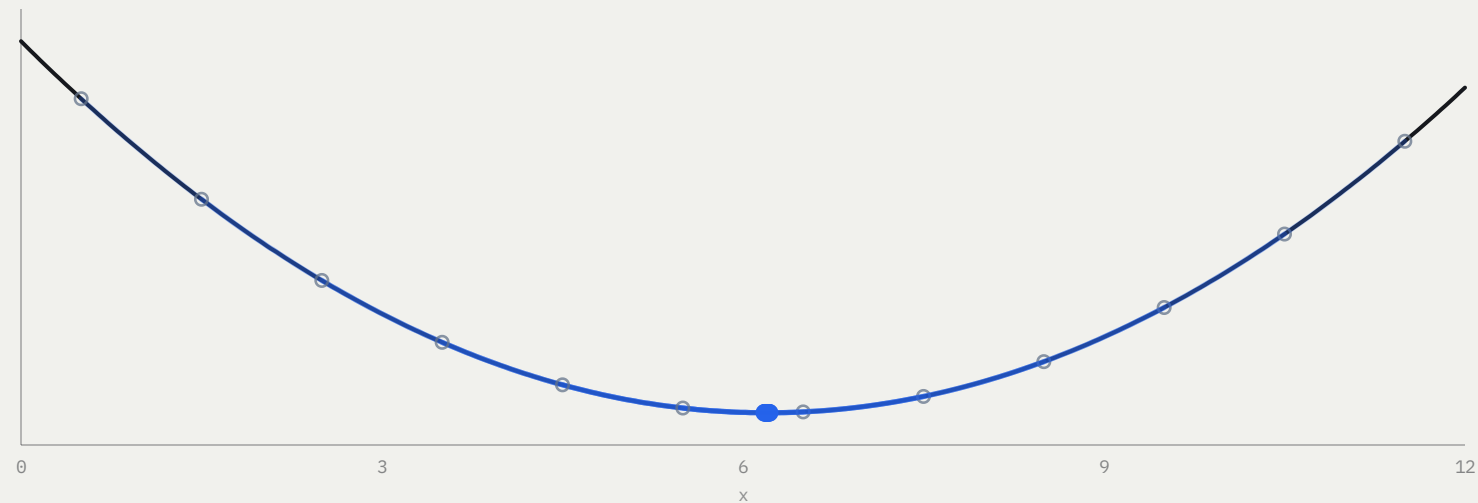
This is the promise Act 2 opened with, and it is now one line. If x^* is an unconstrained local minimum then $\nabla f(x^*) = 0$, so for **every** y :

$$f(y) \geq f(x^*) + 0 = f(x^*)$$

Local information about a convex function is global information. Nothing about y was assumed — it may be arbitrarily far away.

The watershed, run twice

LOCAL DESCENT, TWELVE STARTS, ONE OBJECTIVE



every start reached the same point – local minimum = global minimum

The same descent rule, the same twelve starting points, two objectives. For this strictly convex quadratic and this stable step size, the walkers approach the same minimiser. On the non-convex objective they can settle in different basins. The example isolates the effect of geometry; it does not make algorithm settings irrelevant.

The convex family — LP, QP, QCQP

Named convex forms, in order of generality.

CLASS	OBJECTIVE	EACH CONSTRAINT	THE SHAPE IT CARVES
Linear program (LP)	$c^\top x + d$	$a_i^\top x \leq b_i$	a half-space ; m of them cut a polyhedron — flat faces, sharp corners
Quadratic program (QP)	$\frac{1}{2}x^\top Px + q^\top x,$ $P \succeq 0$	$a_i^\top x \leq b_i$	the same polyhedron ; only the objective bent, plane into bowl
QCQP	convex quadratic	$\frac{1}{2}x^\top P_i x + q_i^\top x + r_i \leq 0$	an ellipsoid when $P_i \succ 0$; the walls are curved now, not flat

The progression matters because modelling is often a matter of *recognising* which named class your problem — or its convexified version — falls into; each has mature, reliable solvers. When a finite LP optimum exists and the feasible polyhedron has vertices, an optimal vertex exists; a whole face may also be optimal. The QP's optimum occurs where the lowest attainable objective level set meets the feasible region.

These same quadratic models reappear in Act 4 as the *local* approximation of hard problems — and, much later, as the trust-region subproblem inside TRPO (Lecture 10).

Diet — where linear programming started

Example 1.2

Buy quantities x_j of n foods as cheaply as possible, while meeting every nutritional requirement.

The data is a price list and a nutrition table: food j costs c_j , and one unit of it carries a_{ij} of nutrient i . The diet must supply at least b_i of each nutrient, and no quantity may be negative.

$$\min_{x \in \mathbb{R}_+^n} c^\top x \quad \text{s.t.} \quad \sum_j a_{ij} x_j \geq b_i \quad \forall i$$

Look at the shape rather than the story. The objective is a price vector dotted with a quantity vector; each constraint is one row of the nutrition table held against one requirement. In the standard form of Act 1, "at least b_i " becomes $b_i - a_i^\top x \leq 0$ — the same content with one sign flipped.

If a finite optimum exists, an optimal basic feasible diet can be chosen: it uses no more positive food quantities than there are linearly independent binding nutrient constraints. This is a property of **one sparse optimal solution**, not of every optimal mixture.

Least squares with bounds — nothing to check

a quadratic programme that is convex for every $\mathbf{A}$

Four problems follow, and each is put the same question: convex, and on what condition? This first one carries no condition at all.

$$\min_x \|\mathbf{A}x - b\|_2^2 \quad \text{s.t.} \quad l \leq x \leq u$$

Expanded, the objective is $x^\top \mathbf{A}^\top \mathbf{A}x - 2b^\top \mathbf{A}x + b^\top b$ — a quadratic programme outright, with $P = 2\mathbf{A}^\top \mathbf{A}$ and $q = -2\mathbf{A}^\top b$. Along any direction v , its Hessian $2\mathbf{A}^\top \mathbf{A}$ gives

$$v^\top \mathbf{A}^\top \mathbf{A} v = \|\mathbf{A}v\|_2^2 \geq 0.$$

A squared length cannot be negative, so $\mathbf{A}^\top \mathbf{A} \succeq 0$ for **every** $\mathbf{A}$ — whatever its shape, its rank, or how badly conditioned it is. The bounds are an intersection of $2n$ half-spaces, so the feasible set is a box. Both halves of the problem are convex by construction.

Convexity here is **structural, not conditional**. There is no assumption to verify and no data that could break it.

Check — why convexity is the watershed

What does convexity actually buy you?

- ☐ A A closed-form solution always exists
- ☒ B Every local minimum is automatically a global minimum
- ☐ C The problem can be solved in a fixed number of steps
- ☐ D The feasible set is guaranteed to be non-empty

A local minimum of a convex problem is global. This is a statement about a point that really is a local minimum; it does not guarantee that an arbitrary algorithm or stopping rule finds one. Next we derive conditions that let us check a candidate.

— ACT 3

Certifying optimality

Q3. A solver returns a point. How do you know it is *the* point?

When is a point optimal? — the first-order condition

Q1 State it? → Q2 Solve it? → **Q3 Certify it?** → Q4 Convexify it?

Let X be the **full feasible set**. Assume X is convex, f is convex and differentiable on an open set containing X , and $x^* \in X$.

$$x^* \text{ is globally optimal} \iff \nabla f(x^*)^\top (y - x^*) \geq 0, \quad \forall y \in X$$

Here ∇f collects the partial derivatives, $y - x^*$ is a direction, and their dot product is the initial rate of cost change in that direction. Call the right-hand statement $C(x^*)$, the **first-order condition**: no feasible direction initially decreases the cost.

NECESSARY • OPTIMAL $\Rightarrow$ CONDITION

Must every optimum pass this test? If a point fails, it cannot be optimal.

SUFFICIENT • CONDITION $\Rightarrow$ OPTIMAL

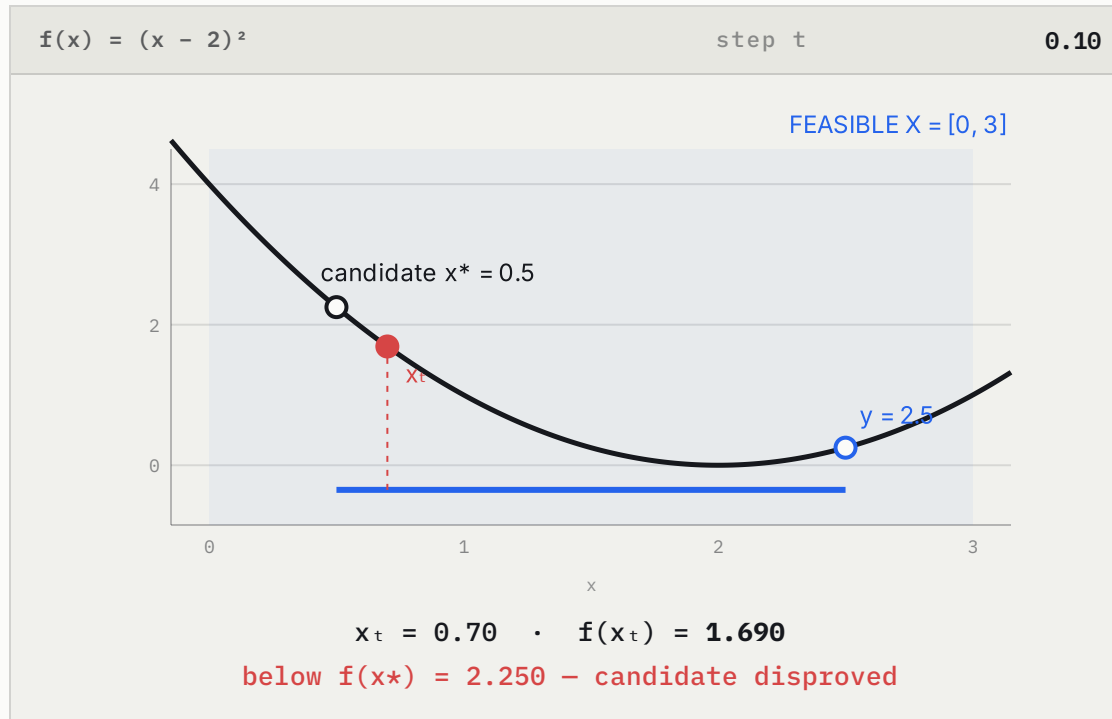
Does passing certify an optimum? If it passes, no feasible point can have lower cost.

Exact means both necessary and sufficient. We prove each arrow separately to show that the condition completely characterises optimality.

Necessary — an optimum cannot have a feasible downhill direction

Question: if x^* really is optimal, must it satisfy $C(x^*)$? We prove **optimal** $\Rightarrow$ **condition** by ruling out a violation.

SEE WHY A CANDIDATE FAILS



THE GENERAL ARGUMENT

Suppose some $y \in X$ violates the condition. Set $d = y - x^*$, so $\nabla f(x^*)^\top d < 0$.

The segment stays feasible because X is convex:

$$x_t = (1 - t)x^* + ty \in X, \quad 0 \leq t \leq 1.$$

The cost initially falls because f is differentiable:

$$\left. \frac{d}{dt} f(x_t) \right|_{t=0} = \nabla f(x^*)^\top d < 0.$$

Thus $f(x_t) < f(x^*)$ for all sufficiently small $t > 0$, contradicting optimality.

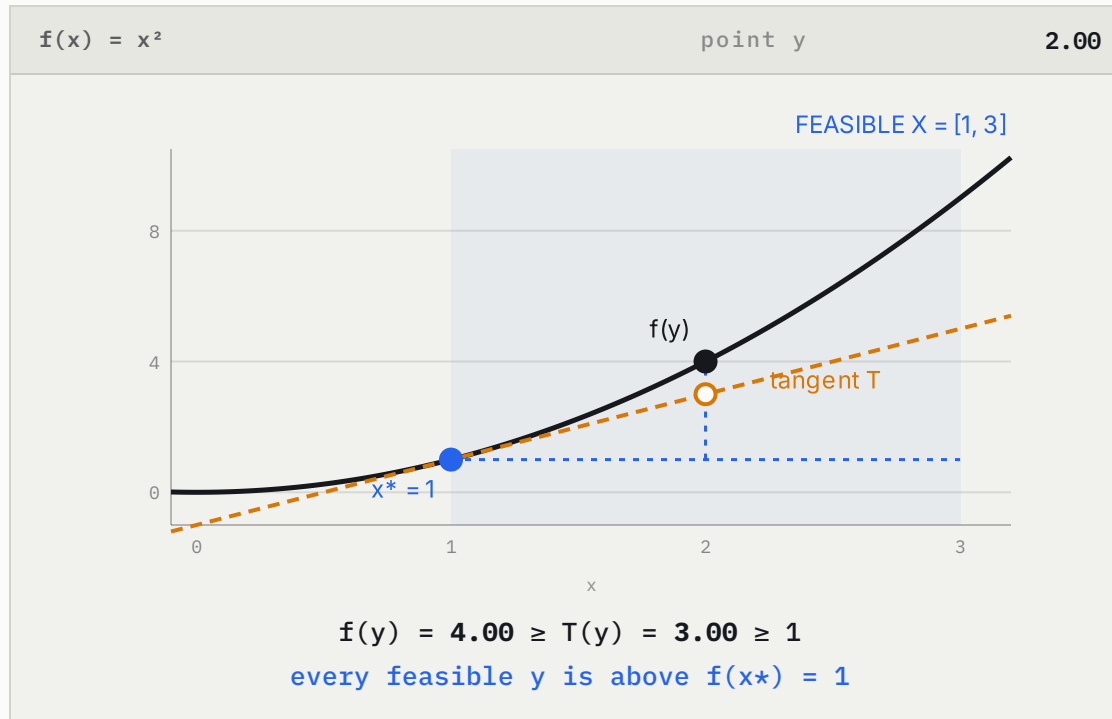
Here $f'(0.5)(2.5 - 0.5) = -6 < 0$. Move t to take a small feasible step and watch the cost fall.

An optimum must pass. **Necessity uses the convexity of X ; it does not require convexity of f .** The same argument applies to a local minimum.

Sufficient — a convex tangent bound certifies every feasible point

Question: if x^* satisfies $C(x^*)$, can we certify a global optimum? Now prove **condition** $\Rightarrow$ **optimal**.

WATCH THE TWO INEQUALITIES



THE GENERAL ARGUMENT

Convexity of f gives a global lower bound:

$$f(y) \geq \underbrace{f(x^*) + \nabla f(x^*)^\top (y - x^*)}_{T(y)}.$$

The assumed condition lifts that bound above the candidate cost: for every $y \in X$,

$$T(y) \geq f(x^*).$$

Combine them:

$$f(y) \geq T(y) \geq f(x^*), \quad \forall y \in X.$$

Every feasible comparison point has at least the candidate's cost.

This certifies a **global** minimum.

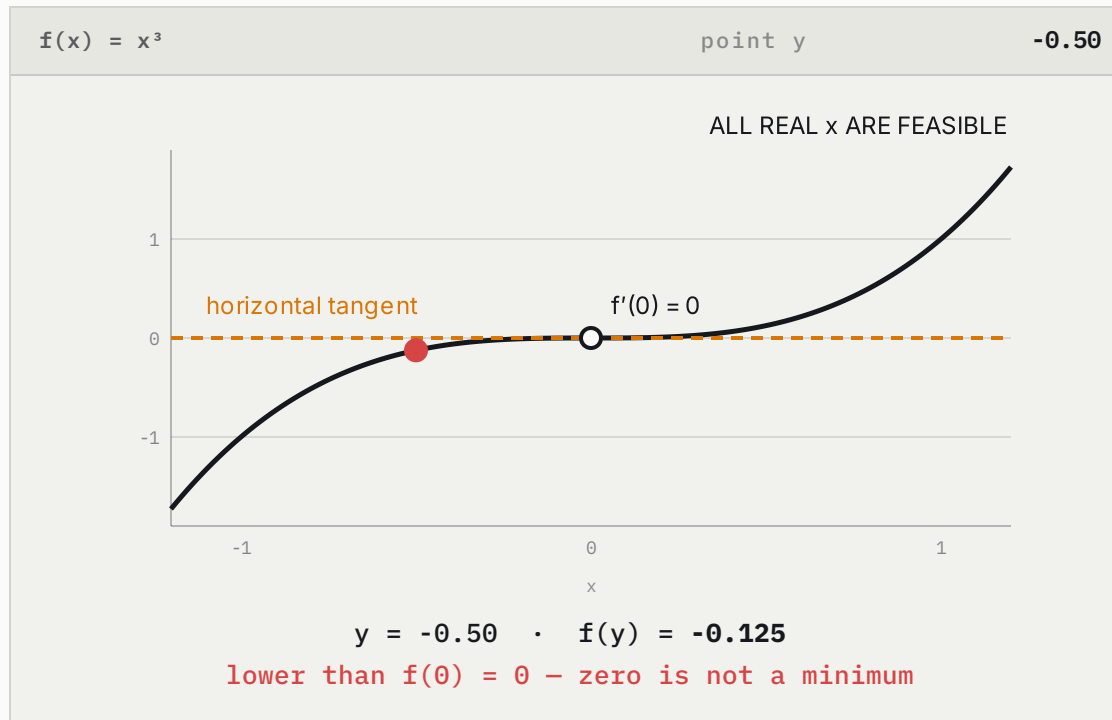
Illustration: minimise x^2 over $X = [1, 3]$. At $x^* = 1$, the orange tangent is $T(y) = 1 + 2(y - 1)$. Move the feasible comparison point y .

Convexity of f turns first-order information into a global certificate. A tangent to a general non-convex function need not be a lower bound.

Necessary alone does not certify a minimum

Remove convexity of the **objective**. Keep the convex feasible set $X = \mathbb{R}$ and take $f(x) = x^3$.

ZERO SLOPE, BUT LOWER POINTS ARBITRARILY CLOSE



The orange horizontal tangent passes through the candidate $x^* = 0$. Move y to the left: the curve falls below the tangent.

THE CONDITION PASSES — OPTIMALITY FAILS

At zero, $f'(0) = 0$. Therefore the full first-order condition holds:

$$f'(0)(y - 0) = 0 \geq 0, \quad \forall y \in \mathbb{R}.$$

Yet for every $\epsilon > 0$,

$$f(-\epsilon) = -\epsilon^3 < 0 = f(0).$$

There are better points arbitrarily close to zero. It is **not even a local minimum**.

The necessary implication still holds. The sufficient implication fails because this tangent is **not a global lower bound**.

Passing a necessary test only keeps a candidate in consideration. It does not, by itself, prove optimality.

Why the condition is exact — the two arrows do different work

For a differentiable f and feasible x^* , the two proofs use different geometric facts:

DIRECTION	CONVEXITY USED IN THE PROOF	WHAT IT GIVES
Necessary: optimal $\Rightarrow C(x^*)$	Convex set X	The segment toward any feasible y stays feasible; a negative initial slope would disprove optimality.
Sufficient: $C(x^*) \Rightarrow$ global optimal	Convex function f	The tangent is a global lower bound; the condition puts that bound above $f(x^*)$ on X .

IF ONLY NECESSITY HAS BEEN PROVED

Failing the condition rules out optimality. **Passing alone is inconclusive** — as x^3 demonstrates.

IF ONLY SUFFICIENCY HAS BEEN PROVED

Passing the condition certifies optimality. **Failing alone is inconclusive**: this arrow does not say that every optimum must pass.

With both assumptions in place, combine the arrows:

$$x^* \text{ globally optimal} \iff \nabla f(x^*)^\top (y - x^*) \geq 0, \quad \forall y \in X$$

Unconstrained optimality — combine two equivalences

Keep f **convex and differentiable**. The conclusion follows in two steps; the second step simplifies the condition from the first.

Step 1 — the general convex problem, already proved. For a convex feasible set X and $x^* \in X$,

$$x^* \text{ globally optimal} \iff \nabla f(x^*)^\top (y - x^*) \geq 0, \quad \forall y \in X.$$

Step 2 — no constraints means $X = \mathbb{R}^n$. Every direction is available, so the first-order inequality simplifies to

$$\nabla f(x^*)^\top (y - x^*) \geq 0, \quad \forall y \in \mathbb{R}^n \iff \nabla f(x^*) = 0.$$

We prove this second equivalence on the next slide. It is a statement about which directions are feasible.

Combine the two — convex + differentiable + unconstrained:

$$x^* \text{ globally optimal} \iff \nabla f(x^*) = 0.$$

Under these three assumptions, **zero gradient is both necessary and sufficient for global optimality.**

Why the first-order inequality becomes a zero gradient

Proving Step 2. Set $X = \mathbb{R}^n$ and write $C(x^*)$ for $\nabla f(x^*)^\top (y - x^*) \geq 0$ for every $y \in \mathbb{R}^n$.

ZERO GRADIENT $\Rightarrow$ FIRST-ORDER INEQUALITY

If $\nabla f(x^*) = 0$, then for every y ,

$$\nabla f(x^*)^\top (y - x^*) = 0 \geq 0.$$

So $C(x^*)$ holds immediately.

NECESSARY FOR AN UNCONSTRAINED OPTIMUM

A nonzero gradient gives a feasible descent direction $-\nabla f(x^*)$, so a small step would lower the cost and contradict optimality.

FIRST-ORDER INEQUALITY $\Rightarrow$ ZERO GRADIENT

Assume $C(x^*)$. Because every point is feasible, choose $y = x^* - t\nabla f(x^*)$ with any $t > 0$:

$$0 \leq \nabla f(x^*)^\top (y - x^*) = -t\|\nabla f(x^*)\|_2^2.$$

Because $t > 0$ and a squared norm is non-negative, the gradient must be zero.

SUFFICIENT WHEN F IS ALSO CONVEX

The global tangent bound becomes

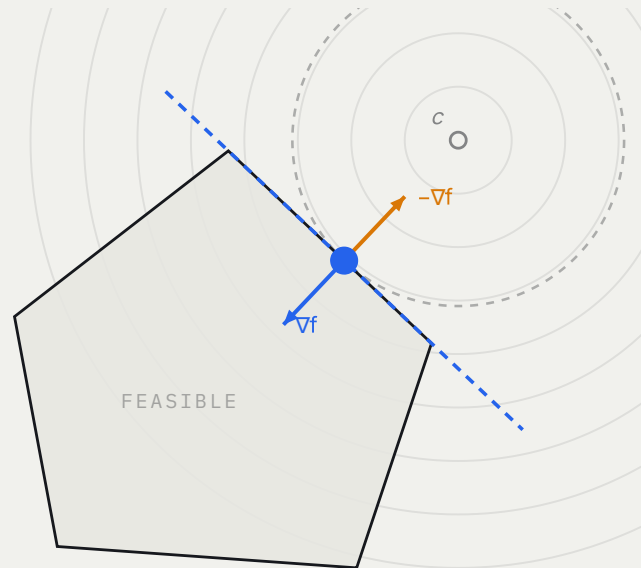
$$f(y) \geq f(x^*) + 0 = f(x^*), \quad \forall y.$$

Thus a zero gradient certifies a global minimum.

Step 2 needs no convexity of f ; **optimality $\iff \nabla f = 0$ also needs Step 1.** Without convexity, $f(x) = x^3$ still has $f'(0) = 0$ but no minimum at zero.

The condition, made draggable

DRAW THE POINT — IS ANY FEASIBLE DIRECTION DOWNHILL?



OPTIMAL

no feasible direction points downhill

$$x = (7.47, 6.65) \cdot f(x) = 9.63$$

$$\min_{y \text{ feasible}} \nabla f(x)^T (y - x) = \mathbf{0.00} \geq 0$$

The point c is the best choice without constraints, but it is outside the polygon. Click “snap to optimum”: the best allowed point lies on the boundary. There, $-\nabla f$ points outside and every feasible direction has a non-negative initial slope. The next example explains how a boundary can balance a nonzero gradient.

Minimise $\|x - c\|^2$ over the polygon. Drag the point: the red arrow shows a feasible decrease. At the optimum, $-\nabla f$ points outside the allowed set. The gradient need not be zero when constraints block descent.

The first-order test checks **every feasible** y . **KKT gives a finite set of conditions using the objective and constraint gradients.** We first see the idea with one bound.

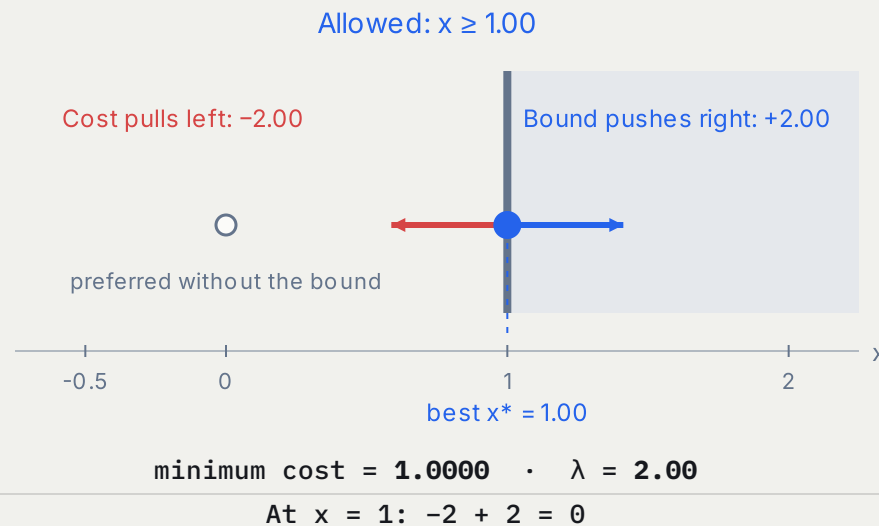
Why can the optimum stop at a wall?

one variable, one constraint — the idea behind KKT

Choose a setting x with cost x^2 . The setting must be at least 1:

$$\min_x x^2 \quad \text{subject to } x \geq 1 \quad (g(x) = 1 - x \leq 0)$$

ONE VARIABLE, ONE WALL



THREE SMALL STEPS

Without the wall: choose $x = 0$, where the cost is smallest.

With the wall: stop at $x^* = 1$. The cost still pulls left: $-f'(1) = -2$.

Balance the pull: $g'(x) = -1$, so the inward correction $-\lambda g'(x) = +\lambda$ points right. Balance requires $-2 + \lambda = 0$, giving $\lambda = 2$.

The nonzero objective gradient is balanced by a constraint. The multiplier λ is a mathematical weight; the “wall force” is a geometric interpretation.

KKT setup — the problem and its Lagrangian

introduce the variables before stating the conditions

Assume $f, g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ are continuously differentiable. Consider

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m, \\ & h_j(x) = 0, \quad j = 1, \dots, p. \end{aligned}$$

The **Lagrangian** combines their gradients through a weighted sum:

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{j=1}^p \nu_j h_j(x).$$

Hold the weights λ, ν fixed when differentiating with respect to x . We seek x^* and suitable weights λ^*, ν^* . **KKT specifies which combinations are valid.**

KKT conditions — formal statement

Karush–Kuhn–Tucker conditions

A point x^* satisfies KKT if **there exist** multipliers $\lambda^* \in \mathbb{R}^m$, $\nu^* \in \mathbb{R}^p$ satisfying all four conditions:

1. Primal feasibility: $g_i(x^*) \leq 0$, $h_j(x^*) = 0$.
2. Dual feasibility: $\lambda_i^* \geq 0$, ν_j^* unrestricted in sign.
3. Complementary slackness: $\lambda_i^* g_i(x^*) = 0$.
4. Stationarity: $\nabla_x L(x^*, \lambda^*, \nu^*) = 0$.

Every inequality condition holds for $i = 1, \dots, m$; every equality condition for $j = 1, \dots, p$. Expanded stationarity is

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^p \nu_j^* \nabla h_j(x^*) = 0.$$

Feasible point, valid multipliers, complementary slackness, and balanced gradients — all four together. We next interpret them, then state their optimality guarantee.

Read the four conditions in plain language

a constraint is active when its boundary is reached

For an inequality $g_i(x) \leq 0$, **active (tight)** means $g_i(x) = 0$; **slack** means $g_i(x) < 0$.

CONDITION	MEANING	OUR ONE-BOUND EXAMPLE
Primal feasibility	The decision obeys the constraints.	$x \geq 1$
Dual feasibility	Non-negative weights give inward constraint reactions.	$\lambda \geq 0$
Complementary slackness	A slack constraint has zero weight.	$x > 1 \Rightarrow \lambda = 0$
Stationarity	Objective and weighted constraint gradients sum to zero.	$2x - \lambda = 0$

Equality weights may have either sign. For $\min x^2$ with $x - 1 = 0$, stationarity gives $2x + \nu = 0$: at $x = 1$, $\nu = -2$.

Active does not mean positive weight. For $\min x^2$ subject to $x \geq 0$, the optimum $x = 0$ is active but $\lambda = 0$.

KKT optimality theorem — necessary and sufficient

conditions first; their guarantee second; the proof follows the examples

For the differentiable problem just stated, suppose f, g_i are **convex** and h_j are **affine**: $h_j(x) = a_j^\top x - b_j$.

SUFFICIENT: KKT CERTIFIES AN OPTIMUM

If (x^*, λ^*, ν^*) satisfies all four KKT conditions, then x^* is a **global minimizer**.

KKT $\implies$ global optimality.

This direction does **not** require Slater's condition.

NECESSARY: AN OPTIMUM HAS MULTIPLIERS

If **Slater's condition** holds, every global minimizer x^* admits multipliers satisfying KKT.

global optimality $\implies$ KKT.

Slater requires **some** point $\bar{x}$ with all $g_i(\bar{x}) < 0$ and all $h_j(\bar{x}) = 0$.

For $x \geq 1$, choose $\bar{x} = 2$. The optimum itself may lie on the boundary.

Under the stated convexity assumptions and Slater: x^* is globally optimal if and only if suitable KKT multipliers exist.

For a non-convex differentiable problem, a local minimum still has KKT multipliers under an appropriate constraint qualification, but a KKT point need not be a local minimum. Reference: Boyd & Vandenberghe, *Convex Optimization*, §5.5.3.

Solve the example — where do x and the multiplier come from?

complementary slackness gives two cases to check

For $\min x^2$ subject to $x \geq 1$, KKT reads

$$x \geq 1, \quad \lambda \geq 0, \quad \lambda(1 - x) = 0, \quad 2x - \lambda = 0.$$

CASE 1: THE BOUND IS SLACK

Suppose $x > 1$. Complementary slackness forces $\lambda = 0$.

Stationarity then gives $2x = 0$, so $x = 0$. This contradicts $x > 1$.

Discard this case.

CASE 2: THE BOUND IS ACTIVE

Set $x = 1$. Stationarity gives $2 - \lambda = 0$, so $\lambda = 2$.

The point is feasible, the multiplier is non-negative, and $\lambda(1 - x) = 0$.

All four conditions hold.

We obtained $x^* = 1$, $\lambda^* = 2$ from the equations. Convexity makes this KKT solution globally optimal; Slater tells us that every optimum must appear among the KKT candidates.

KKT example — check the four conditions

check the same example at $x^* = 1$ and $\lambda^* = 2$

Combine the cost and the constraint in one expression — the **Lagrangian**:

$$L(x, \lambda) = x^2 + \lambda(1 - x) \quad \frac{\partial L}{\partial x} = 2x - \lambda$$

QUESTION	KKT CONDITION	CHECK AT $(x, \lambda) = (1, 2)$
Is the setting allowed?	Primal feasibility: $1 - x \leq 0$	$1 - 1 = 0 \leq 0 \checkmark$
Does the wall push inward?	Dual feasibility: $\lambda \geq 0$	$2 \geq 0 \checkmark$
Is any unused wall powerless?	Complementary slackness: $\lambda(1 - x) = 0$	$2(1 - 1) = 0 \checkmark$
Do the two pushes balance?	Stationarity: $2x - \lambda = 0$	$2(1) - 2 = 0 \checkmark$

All four checks pass. **For this convex problem, that certifies the global optimum.**

Quick check: $x = 2, \lambda = 4$ passes stationarity. Is it optimal? No: $\lambda(1 - x) = -4 \neq 0$ fails complementary slackness. Stationarity alone is not enough.

Certifying the production decision with actual multipliers

Return to the workshop: minimise $-3x_1 - 2x_2$. Check the proposed decision $x^* = (0, 4)$.

1 · CHECK CONSTRAINTS AND IDENTIFY ZERO WEIGHTS

CONSTRAINT	VALUE AT THE CANDIDATE	IMPLICATION
$g_1 = 2x_1 + x_2 - 4$	0 — active	Solve for λ_1 .
$g_2 = x_1 - 1$	-1 — slack	$\lambda_2 = 0$
$g_3 = -x_1$	0 — active	Solve for λ_3 .
$g_4 = -x_2$	-4 — slack	$\lambda_4 = 0$

2 · SOLVE THE STATIONARITY EQUATIONS

Set each partial derivative of the Lagrangian to zero. With $\lambda_2 = \lambda_4 = 0$:

$$\begin{aligned} \text{In } x_2 : \quad & -2 + \lambda_1 = 0, \\ & \lambda_1 = 2. \end{aligned}$$

$$\begin{aligned} \text{In } x_1 : \quad & -3 + 2\lambda_1 - \lambda_3 = 0, \\ & \lambda_3 = 1. \end{aligned}$$

$\lambda^* = (2, 0, 1, 0) \geq 0$. The point is feasible, every $\lambda_i^* g_i(x^*) = 0$, and stationarity holds. **All four conditions pass: profit 8 is globally optimal.**

The machine-time multiplier is 2. An extra unit of machine capacity is worth 2 units of profit in this model.

Temperature thread — solve the one-step heater problem

shared teaching example · predict before revealing the calculation

Use $x = -2$, $x' = x + u$ and $c(u) = (-2 + u)^2 + u^2$, with $0 \leq u \leq 2$. The dynamics and cost are known in this chapter.

Predict: Will the optimum use zero, half, or all of the maximum command?

Calculate and check. $c'(u) = 4u - 4 = 0$ gives $u^* = 1$. It is feasible and $c''(u) = 4 > 0$, so it is the unique global minimizer. Both bound multipliers are zero; the resulting cost is **2**.

The command stops short of reaching 22°C because the objective also charges for effort.

Try it — the heater limit becomes 0.5

work independently · reveal only after writing an answer

Replace only the upper bound by $u \leq 0.5$. Find the optimal command and the multiplier for $u - 0.5 \leq 0$. Check feasibility, multiplier sign, stationarity and complementary slackness.

Check your answer. $u^* = 0.5$, $c = 2.5$. The lower-bound multiplier is zero. Stationarity gives $(4u - 4) + \lambda = 0$, so $\lambda = 2$. The upper constraint binds and $\lambda(u - 0.5) = 0$.

A constrained optimum can have a nonzero objective gradient. The constraint multiplier balances it.

Why KKT certifies a **global** optimum

reuse the zero-gradient result we already proved

Let f, g_i be convex and h_j affine, and suppose all four KKT conditions hold. Fix the multipliers and write $\ell(x) = L(x, \lambda^*, \nu^*)$.

1 • STATIONARITY GIVES A MINIMUM OF THE LAGRANGIAN

ℓ is convex: it sums convex f , non-negative multiples of convex g_i , and affine equality terms.

KKT gives $\nabla \ell(x^*) = 0$. By the earlier unconstrained result,

$$\ell(y) \geq \ell(x^*) \quad \text{for every } y.$$

2 • FEASIBILITY CONNECTS IT TO THE ACTUAL COST

For feasible y , $\lambda_i^* g_i(y) \leq 0$ and $h_j(y) = 0$, so $\ell(y) \leq f(y)$.

At x^* , complementary slackness makes every $\lambda_i^* g_i(x^*) = 0$; equality terms are also zero. Thus $\ell(x^*) = f(x^*)$.

Combine the two steps for any feasible y :

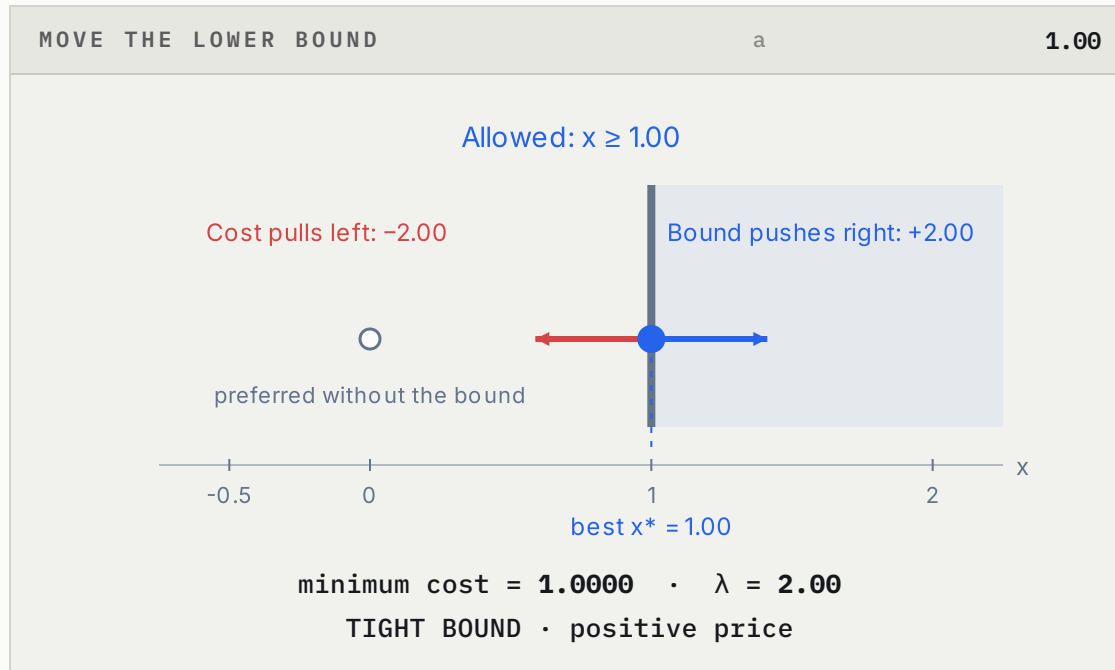
$$f(y) \geq \ell(y) \geq \ell(x^*) = f(x^*).$$

Every feasible point costs at least as much as x^* . This proves sufficiency. The expanded gradient calculation is kept in the appendix.

What is a little more freedom worth?

the same multiplier also measures the value of relaxing the wall

Now let the minimum allowed setting be a : minimise x^2 subject to $x \geq a$.



RELAX THE BOUND FROM 1 TO 0.9

Before: best $x = 1$, cost 1, multiplier $\lambda = 2$.

After: best $x = 0.9$, cost 0.81. The actual saving is **0.19**.

Predict the saving using the original price:

$$\lambda \times \text{relaxation} = 2 \times 0.1 = 0.20.$$

Close to 0.19: the price gives a **local approximation** for a small change.

Move a left to relax the bound. Once $a < 0$, the preferred point $x = 0$ is inside the allowed region: the bound is slack and its price is zero.

λ is the **shadow price**: how much the best cost improves per small unit of extra freedom. At $a = 0$, the bound is tight but its price is already zero.

Duality — a lower bound you get for free

The proof used a function below the feasible cost. Now fix **any** $\lambda \geq 0, \nu$ and minimise that function over all x , even infeasible ones:

$$d(\lambda, \nu) = \inf_x L(x, \lambda, \nu)$$

For **any** $\lambda \geq 0$ and any ν , and any feasible $\tilde{x}$:

$$d(\lambda, \nu) \leq L(\tilde{x}, \lambda, \nu) = f(\tilde{x}) + \underbrace{\sum_i \lambda_i g_i(\tilde{x})}_{\leq 0} + \underbrace{\sum_j \nu_j h_j(\tilde{x})}_{=0} \leq f(\tilde{x})$$

Thus $d(\lambda, \nu) \leq p^*$: **weak duality**. A feasible decision gives an upper bound; these multipliers give a lower bound. The dual problem chooses the highest lower bound, $d^* = \sup_{\lambda \geq 0, \nu} d(\lambda, \nu)$. The difference $p^* - d^* \geq 0$ is the **duality gap**.

For a convex problem with a constraint qualification the gap is zero, so the dual optimum *proves* the primal one. Even when you cannot solve the primal, any dual point brackets how far you might still be from optimal — which is the practical value of duality. With additional differentiability and regularity conditions, the KKT system also provides sensitivities for optimisation layers. The following examples separate that question from optimality.

Duality in numbers — the same example, one last time

an achievable cost and a lower bound meet at 1

For $\min x^2$ subject to $x \geq 1$, complete the square in the Lagrangian:

$$L(x, \lambda) = x^2 + \lambda(1 - x) = \left(x - \frac{\lambda}{2}\right)^2 + \lambda - \frac{\lambda^2}{4}.$$

Minimising over all real x sets $x = \lambda/2$, so $d(\lambda) = \lambda - \lambda^2/4$, with $\lambda \geq 0$.

A FEASIBLE DECISION GIVES AN UPPER BOUND

$x = 1$ is allowed and costs $f(1) = 1$.

The best possible cost cannot be higher:

$$p^* \leq 1.$$

THE KKT MULTIPLIER GIVES A LOWER BOUND

$\lambda = 2$ is allowed and gives $d(2) = 1$.

Weak duality says the cost cannot be lower:

$$p^* \geq 1.$$

$1 \leq p^* \leq 1$, so $p^* = 1$. The decision and the bound agree. This is the certificate from the KKT proof, written as a dual bound.

Check — the price of a slack constraint

At the optimum, an inequality constraint turns out to be *slack*— it is satisfied strictly, not at its boundary. What do the KKT conditions say about its multiplier λ ?

- ☐ A $\lambda > 0$, and it measures how far the constraint is from binding
- ☐ B λ is undetermined by the KKT conditions
- ☒ C $\lambda = 0$
- ☐ D $\lambda < 0$, since the constraint pushes the optimum outward

Since $g(x^*) < 0$, the equation $\lambda g(x^*) = 0$ forces $\lambda = 0$. The converse is false: zero multiplier does not prove that a constraint is slack. Under suitable sensitivity assumptions, zero multiplier means no first-order change in the optimal value; it does not generally guarantee an unchanged optimizer after a finite relaxation.

— ACT 4

When the problem is not convex

Q4. Real problems are not convex. Do we give up the guarantees, or manufacture them?

Successive convexification — solve a sequence of easy problems



Real engineering problems are usually non-convex — non-convex objective, non-convex constraints. One useful strategy is to **solve a sequence of local convex approximations**, then evaluate each proposed step in the original problem.

At the current iterate $x^{(k)}$, choose $B^{(k)} \succeq 0$ and build a local convex model for minimisation

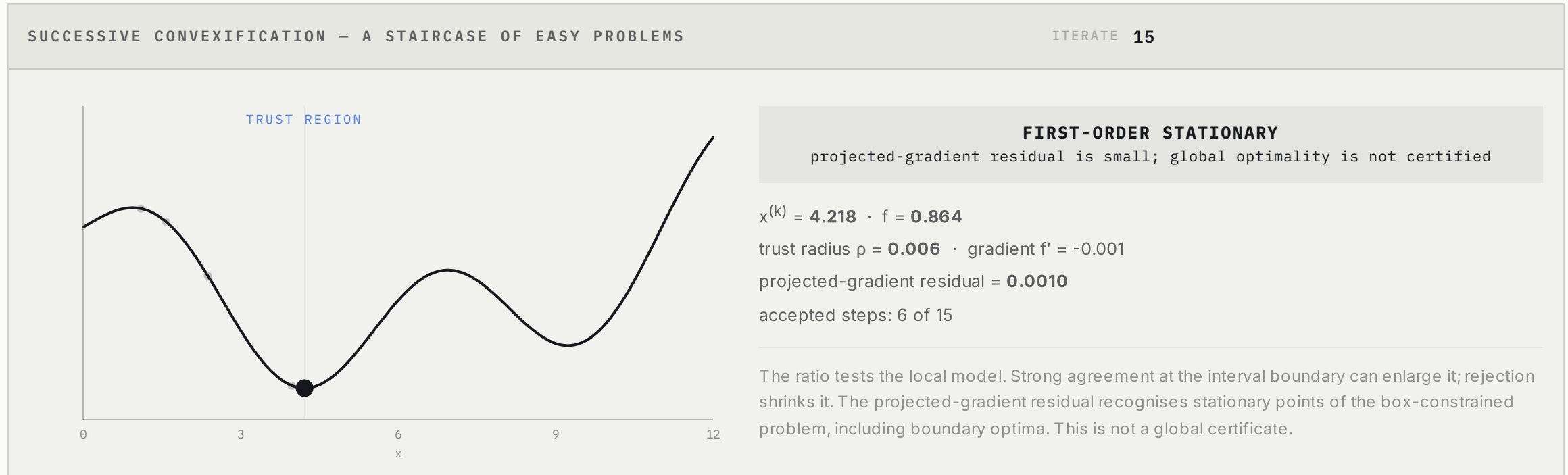
$$\tilde{f}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^\top (x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^\top B^{(k)}(x - x^{(k)})$$

construct suitable constraint approximations, and add a **trust region** $\|x - x^{(k)}\| \leq \rho^{(k)}$ so the model stays trustworthy.

The trust region is the key idea: **only trust the approximation nearby**.

Accept a feasible candidate only when actual improvement agrees sufficiently with predicted improvement. Shrink after rejection; expand after strong agreement at the boundary. General constraint linearisations need a feasibility or merit-function strategy. Convergence, when obtained under suitable assumptions, is usually to a stationary point rather than a global optimum.

The trust region, iterated



Watch the ratio of actual to predicted decrease. Poor agreement rejects a step; strong agreement at the boundary can enlarge the interval. The boxed problem uses a projected-gradient stopping test, so a boundary optimum is recognised. Lecture 10 applies the same idea of limiting an update through a KL constraint in TRPO, with a different step-selection procedure.

Wind-farm layout — the wakes make it a joint decision



Visible wakes behind offshore turbines. Photograph reproduced from original PowerPoint, slide 43, which credits WindAction.

THE DECISION IN THE PHOTOGRAPH

A turbine extracts energy from the wind and leaves slower, more turbulent air behind it. Downstream turbines therefore see a different inflow from the undisturbed wind.

Variables: the turbine coordinates $l = (l_1, \dots, l_N)$.

Objective: expected total farm power, accounting for wake interactions.

Constraints: installation boundaries and a prescribed minimum separation $d_{\min}$.

Moving one turbine changes several turbines' power. **Optimise the layout as a system.**

Read the engineering case using the same four questions

QUESTION	WIND-FARM DECISION
What do we choose?	Turbine locations, subject to site and separation constraints.
What do we value?	Predicted power over stated wind scenarios.
Why is it difficult?	Moving one turbine changes wakes and other turbines' outputs.
What does the algorithm check?	A local convex subproblem proposes a step; the original model evaluates whether the step deserves trust.

The full wake model, figures and optimization results are retained in the case-study appendix. First explain the **decision, objective, constraints and validation** without the network architecture.

Check — what a trust region is for

Why does a trust-region method bound the size of its step?

- ☐ A To keep the iterate inside the feasible set
- ☐ B To guarantee the objective decreases monotonically
- ☐ C To make each iteration cheaper to compute
- ☒ D Because the local model is only a good approximation nearby, so the step must stay where the model can be believed

The method builds a simple model — usually quadratic — of a complicated f around the current point. That model is **only trustworthy in a neighbourhood**, so the step is capped at a radius that is grown or shrunk according to how well the model just predicted reality. Lecture 10 uses the same principle to limit a policy update with a KL constraint; its step-selection procedure differs.

— CLOSING

Closing

We can state a decision and, when it is convex, prove we have solved it. Now the assumption we drop.

Where we are — and the assumption we now drop

MODEL - BASED		DATA - DRIVEN
Static, single	optimisation (<i>Lec 1 ✓</i>)	Bayesian / learned opt. (<i>Lec 2–6 →</i>)

We can now state a decision, certify a convex optimum, and use convex subproblems to improve a non-convex problem. The research cases already fitted models and used wind distributions, but each optimisation treated those inputs as fixed:

the chosen model and its coefficients defined **one optimisation problem**.

We hand off the template $\min f \text{ s.t. } g \leq 0$. Now ask what data tells us about its unknown parameters. A fixed surrogate or one fitted distribution hides uncertainty about that fit; Lecture 2 represents this uncertainty as a belief and updates it with observations.

Optimization is how a decision becomes mathematics — and, when the problem is convex, an answer we can not only find but *prove*.

An objective to minimise, levers to pull, limits to respect.

Questions?

Hold onto two pieces: the standard form — it is the skeleton of every objective to come — and the trust region, which returns in surrogate optimisation and again in trust-region RL. Everything else this term is this atom, perturbed by uncertainty.

— APPENDIX

Backup slides

Complete statements, kept out of the narrative.

Reading guide — certify an optimum, then differentiate a decision

one main idea to explain, one comparison, one application

ROLE	READ OR REVISIT	QUESTION TO ANSWER
Core	Boyd & Vandenberghe, <i>Convex Optimization</i> (2004), §§3.1, 4.1, 5.5	Which assumptions make first-order or KKT conditions a certificate?
Compare	Amos & Kolter, <i>OptNet: Differentiable Optimization as a Layer in Neural Networks</i> (ICML 2017)	How does changing a QP input change its optimal decision?
Apply	The production example, followed by the original heater and wind-farm cases	What remains convex, and what requires an approximation?

KKT conditions and a small worked example come before optimization layers. OptNet is the advanced application of sensitivity analysis; ICNN describes a different choice: how to parameterize a convex function.

Backup — the successive convex programming loop

Algorithm — trust-region SCP.

1. Start from a feasible $x^{(0)}$, radius $\rho_0 > 0$ and tolerance $\epsilon > 0$.
2. Check feasibility and a suitable first-order residual; stop only when both meet their tolerances. For a box, use $\|x - \Pi_D(x - \nabla f(x))\|$.
3. Build a model with $B_k \succeq 0$ and a feasible inner constraint approximation. Solve it inside the trust region to get $\hat{x}$.
4. Set $\text{pred} = m_k(x^{(k)}) - m_k(\hat{x})$. If it is not meaningfully positive, check stationarity or report stalling; do not divide by zero.
5. Set $r_k = [f(x^{(k)}) - f(\hat{x})]/\text{pred}$. Accept a feasible candidate if $r_k \geq \eta$; otherwise keep $x^{(k+1)} = x^{(k)}$ and shrink the radius.
6. Grow the radius only after strong agreement at the boundary. Return to step 2. A rejected step is never, by itself, evidence of convergence.

Why the ratio test. It measures whether the convex model can be trusted at the proposed step. Trust grows where the model predicts well and shrinks where it does not — the same modelling principle that motivates the KL-constrained update in TRPO, though TRPO uses a different step-selection procedure.

Backup — the general Lagrangian and KKT conditions

For $\min_x f(x)$ subject to $g_i(x) \leq 0$ and $h_j(x) = 0$, combine all constraints:

$$L(x, \lambda, \nu) = f(x) + \sum_i \lambda_i g_i(x) + \sum_j \nu_j h_j(x).$$

ALLOWED POINT AND INWARD REACTIONS

Primal feasibility:

$$g_i(x^*) \leq 0, \quad h_j(x^*) = 0.$$

Dual feasibility: $\lambda_i^* \geq 0$. Equality multipliers ν_j^* are free in sign: an equality has no allowed side.

NO UNUSED REACTION AND BALANCED FORCES

Complementary slackness:

$$\lambda_i^* g_i(x^*) = 0 \quad \forall i.$$

Stationarity:

$$\nabla f(x^*) + \sum_i \lambda_i^* \nabla g_i(x^*) + \sum_j \nu_j^* \nabla h_j(x^*) = 0.$$

For differentiable convex f , g_i and affine h_j , these four conditions are **sufficient** for global optimality. A constraint qualification such as Slater's makes them **necessary** as well. Without convexity, they are generally not sufficient. A slack inequality must have zero multiplier; a tight inequality can have either zero or a positive multiplier.

Backup — KKT sufficiency, expanded gradient proof

proof of the sufficient direction stated in the KKT theorem

Let f and every g_i be differentiable and convex, every h_j affine, and suppose (x^*, λ^*, ν^*) satisfies the stated KKT conditions. Take **any** feasible y :

$$\begin{aligned}
 f(y) &\geq f(x^*) + \nabla f(x^*)^\top (y - x^*) && \text{convexity of } f \\
 &= f(x^*) - \sum_i \lambda_i^* \nabla g_i(x^*)^\top (y - x^*) && \text{stationarity; the } h_j \text{ terms vanish} \\
 &\geq f(x^*) - \sum_i \lambda_i^* (g_i(y) - g_i(x^*)) && \text{convexity of } g_i, \lambda_i^* \geq 0 \\
 &= f(x^*) - \sum_i \lambda_i^* g_i(y) && \text{complementary slackness} \\
 &\geq f(x^*) && \lambda_i^* \geq 0, g_i(y) \leq 0
 \end{aligned}$$

The h_j terms vanish because an affine h_j has $\nabla h_j^\top (y - x^*) = h_j(y) - h_j(x^*) = 0$ for feasible y . So $f(y) \geq f(x^*)$ for every feasible y : **not a local claim but a global one**. This proves the sufficient direction: a KKT certificate gives global optimality for the stated convex problem. The converse requires the constraint qualification stated in the theorem.

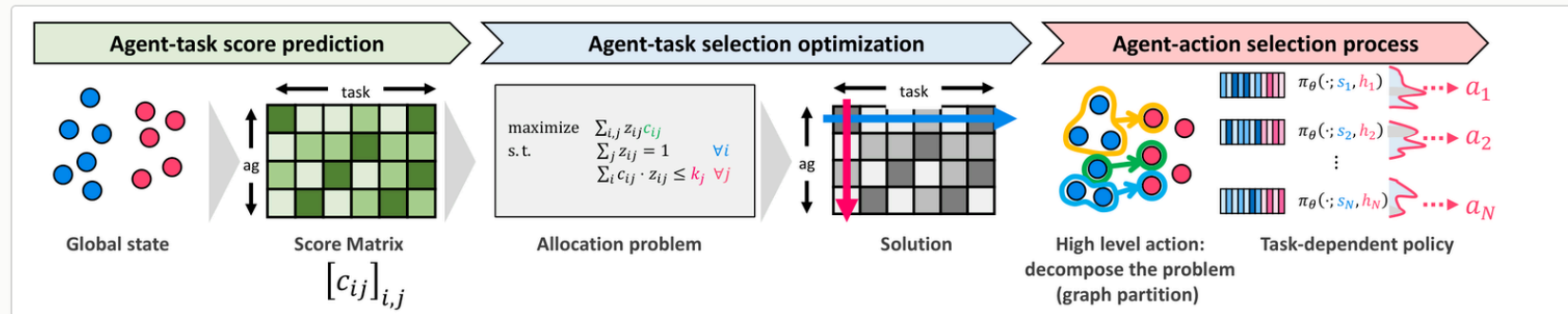
— EXTENSION

Research extension — optimization layers and joint design/control

Read after completing the main route.

An LP inside a policy — LPMARL

application preview — follow how an optimisation output becomes a decision



A score network feeds an allocation LP; each agent's policy is conditioned on the resulting task weights. Source: original PowerPoint, slide 36.

The core KKT argument is complete. In this application, learned scores c_{ij} enter an LP, and its allocation weights z_{ij} guide decisions:

$$\begin{aligned} &\max_{z \geq 0} && \sum_{i,j} c_{ij} z_{ij} \\ &\text{s.t.} && \sum_j z_{ij} = 1 \quad \forall i, \quad \sum_i c_{ij} z_{ij} \leq k_j \quad \forall j. \end{aligned}$$

Each row of z sums to one, but it can split across tasks. Here k_j bounds a **weighted total**, not the number of assigned agents. A hard assignment requires an additional discrete decision or a feasibility-preserving decoding rule.

An allocation LP need not return a matching

One agent, two tasks: let $c = (2, 1)$ and $k = (1, 1)$. The weighted-capacity model becomes

$$\max 2z_1 + z_2 \quad \text{s.t.} \quad z_1 + z_2 = 1, \quad 2z_1 \leq 1, \quad z_2 \leq 1, \quad z \geq 0.$$

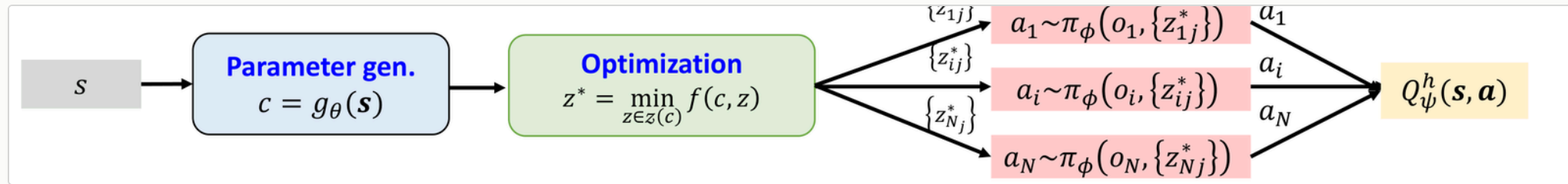
The optimum is $z = (\frac{1}{2}, \frac{1}{2})$, with value 1.5. The only feasible hard assignment is $(0, 1)$, with value 1. **The relaxation is fractional even at its best vertex.**

WHEN THE FAMILIAR INTEGRALITY THEOREM DOES APPLY

For a standard bipartite assignment model, use **count capacities** $\sum_i z_{ij} \leq k_j$, integer k_j , row sums of one and $z \geq 0$. If feasible, an integral optimal vertex exists. Ties can still admit fractional optimal mixtures; arbitrary weighted capacities lose this structural guarantee.

Training through an optimisation layer

optimality and differentiability are separate questions



The learned coefficients change the optimisation problem; its solution changes the actions seen by the critic. Source: original PowerPoint, slide 37. Each agent reads its row of allocation weights.

$$\frac{dJ}{d\theta} = \frac{\partial J}{\partial z^*} \frac{\partial z^*}{\partial c} \frac{\partial c}{\partial \theta}$$

This chain rule is useful only if the middle sensitivity exists. For an exact LP with a fixed feasible region, changing objective coefficients can leave the optimal vertex unchanged, then make it jump. A differentiable layer must specify its regularisation, smoothing or derivative convention; **an exact discrete argmax does not supply useful gradients automatically**. The weighted model above also changes its feasible region through c .

OptNet in one dimension — the decision is the layer output

Amos & Kolter · ICML 2017 · scalar illustration of a differentiable QP layer

A neural network predicts c . Instead of treating c as the final answer, solve a constrained decision:

$$z^*(c) = \arg \min_{z \geq 0} \frac{1}{2}(z - c)^2 = \max(c, 0).$$

PREDICTED INPUT	OPTIMAL DECISION	SENSITIVITY AWAY FROM THE BOUNDARY
$c = -1$	$z^* = 0$	$dz^*/dc = 0$; the constraint binds
$c = 2$	$z^* = 2$	$dz^*/dc = 1$; the constraint is inactive

If the desired decision is 3, use $L = \frac{1}{2}(z^* - 3)^2$. At $c = 2$, $dL/dc = (2 - 3)(1) = -1$. A gradient step of size 0.1 changes c to **2.1**, and the decision loss falls from **0.5 to 0.405**.

Training follows **network input** → **optimal decision** → **task loss**. At $c = 0$ this solution map has a kink; differentiability needs conditions. The next slide states those conditions. [OptNet](#)

What differentiating KKT actually requires

advanced derivation — optional on a first reading

Collect the primal variables and multipliers into $y = (z, \lambda, \nu)$. Locally, write a suitable KKT system as $F(y, c) = 0$. If it is differentiable and its Jacobian in y is nonsingular, the implicit function theorem gives

$$\frac{dy^*}{dc} = - \left(\frac{\partial F}{\partial y} \right)^{-1} \frac{\partial F}{\partial c}.$$

Convexity makes a feasible KKT solution globally optimal. It does **not** by itself give uniqueness, a nonsingular KKT system or smooth dependence on the data. Regularisation and active-set regularity are additional issues.

A stable local minimum of a non-convex problem can also have a correct local sensitivity; it need not describe the global optimum. LP layers commonly use quadratic or barrier regularisation to obtain useful sensitivities. [Mandi & Guns, 2020](#).

Convexity can be imposed on a learned model

Input Convex Neural Networks (ICNNs) constrain a network to be convex in selected inputs while remaining flexible in its parameters. **OptNet** and differentiable convex layers make an optimisation problem part of a trainable network. (*Amos, Xu & Kolter, 2017; Amos & Kolter, 2017; Agrawal et al., 2019*)

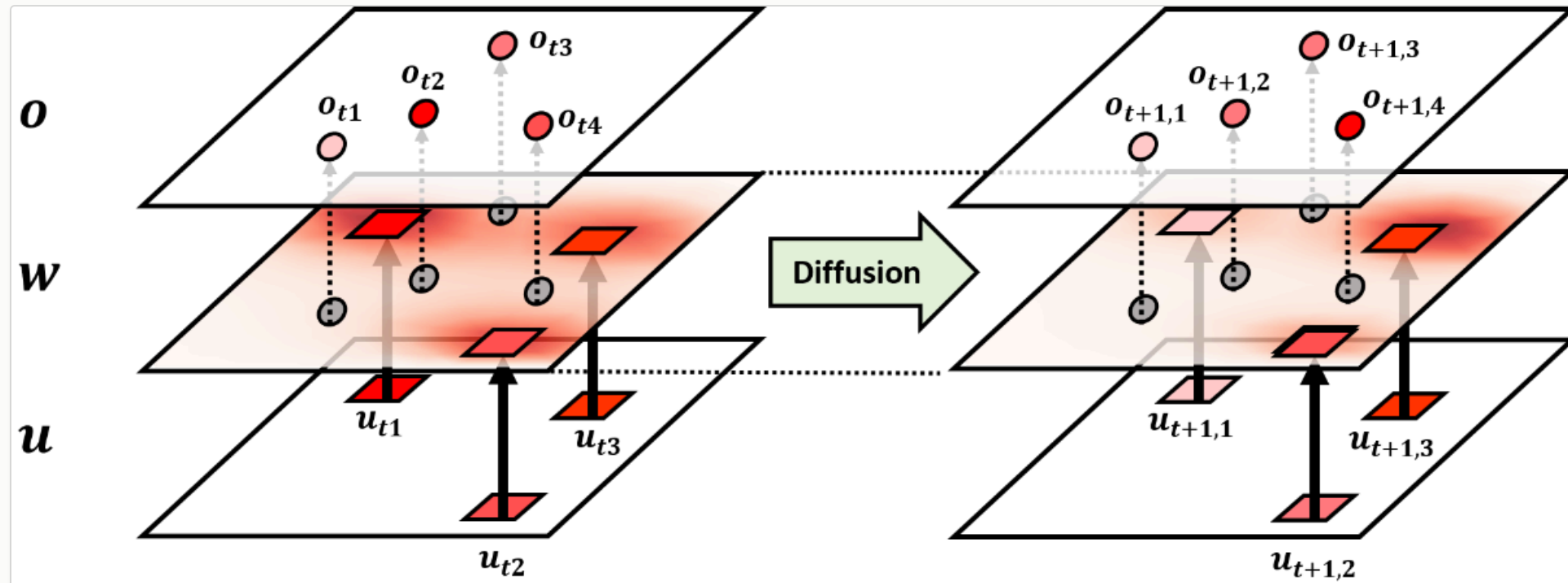
CHECK THE COMPLETE CONTROL PROBLEM

A convex prediction function is not enough. A nonlinear equality $x_{t+1} = f(x_t, u_t)$ generally defines a non-convex set, and composing a convex predictor with an arbitrary cost can destroy convexity. After eliminating the dynamics, check the **whole objective and every remaining constraint** in the control variables.

The modelling opportunity is to learn a useful function while preserving the structure that the **downstream decision problem** needs.

Heater placement changes the control problem

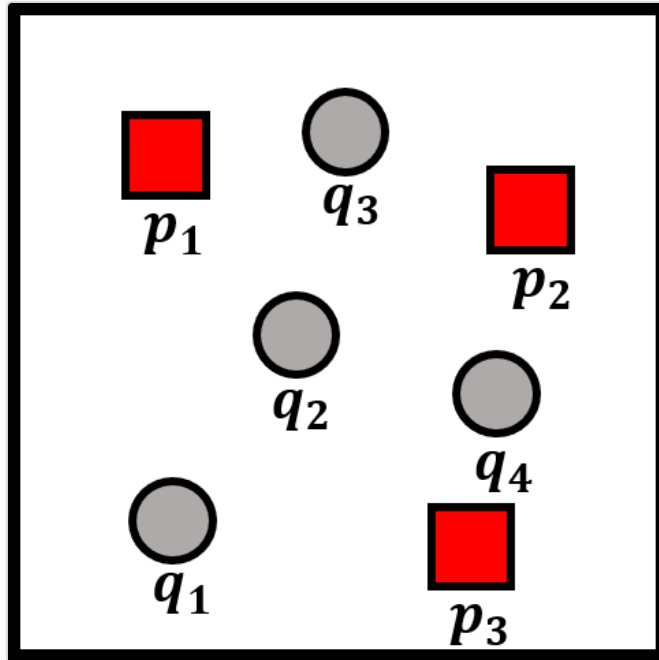
heat diffusion — a physical example of bilevel design



Heater inputs u_t inject heat into a field; diffusion changes the temperatures observed at sensors, $o_t \rightarrow o_{t+1}$. Source: original PowerPoint, slide 26.

We observe temperatures at a few sensor locations rather than the whole field. The model must predict how today's heating changes later observations. Moving a heater changes that response, so choosing a location also changes the controller's job.

Design is chosen once; heating is chosen repeatedly



Red squares are heater locations p_i ; grey circles are fixed sensor locations q_j . Source: original PowerPoint, slide 26.

TWO DECISIONS, TWO TIME SCALES

Design variables: heater coordinates p , constrained to an installation region.

Operation variables: heat inputs $u_0, \dots, u_{T-1}$, constrained by actuator bounds.

Objective: track target sensor temperatures while accounting for the cost of heating.

To compare two layouts fairly, optimise the heating schedule for **each** layout. A poor controller should not make a useful design look bad.

Bilevel design — evaluate a layout through its best operation

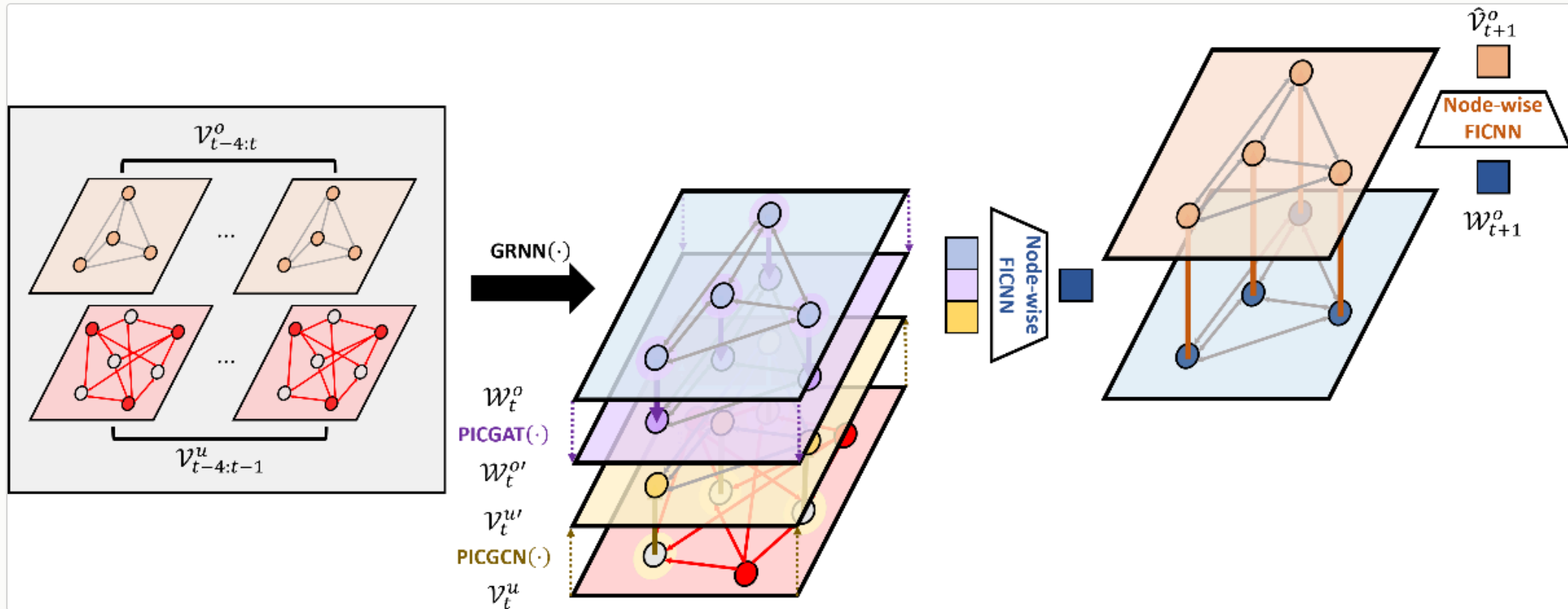
For a fixed initial condition and a fixed design p , let $J_p(u)$ be the cost of rolling out the dynamics under the input sequence u . For one operating scenario,

$$\begin{aligned} \min_{p \in \mathcal{P}} \quad & J_p(u^*(p)), \\ u^*(p) \in \arg \min_{u \in \mathcal{U}_p} \quad & J_p(u), \\ J_p(u) = \sum_{t=0}^{T-1} \mathcal{L}(x_{t+1}, u_t; p), \quad & x_{t+1} = f(x_t, u_t; p). \end{aligned}$$

The source averages this evaluation over several target scenarios. Each outer design update therefore requires several inner control solves. If those problems have stable solutions, their sensitivities tell us how a small heater movement changes the best achievable cost.

Convex inner problems make global operation optima certifiable. The outer design problem can remain non-convex, and a surrogate model can still be wrong about the physical plant. These are separate questions. Source: original PowerPoint, slides 24–27.

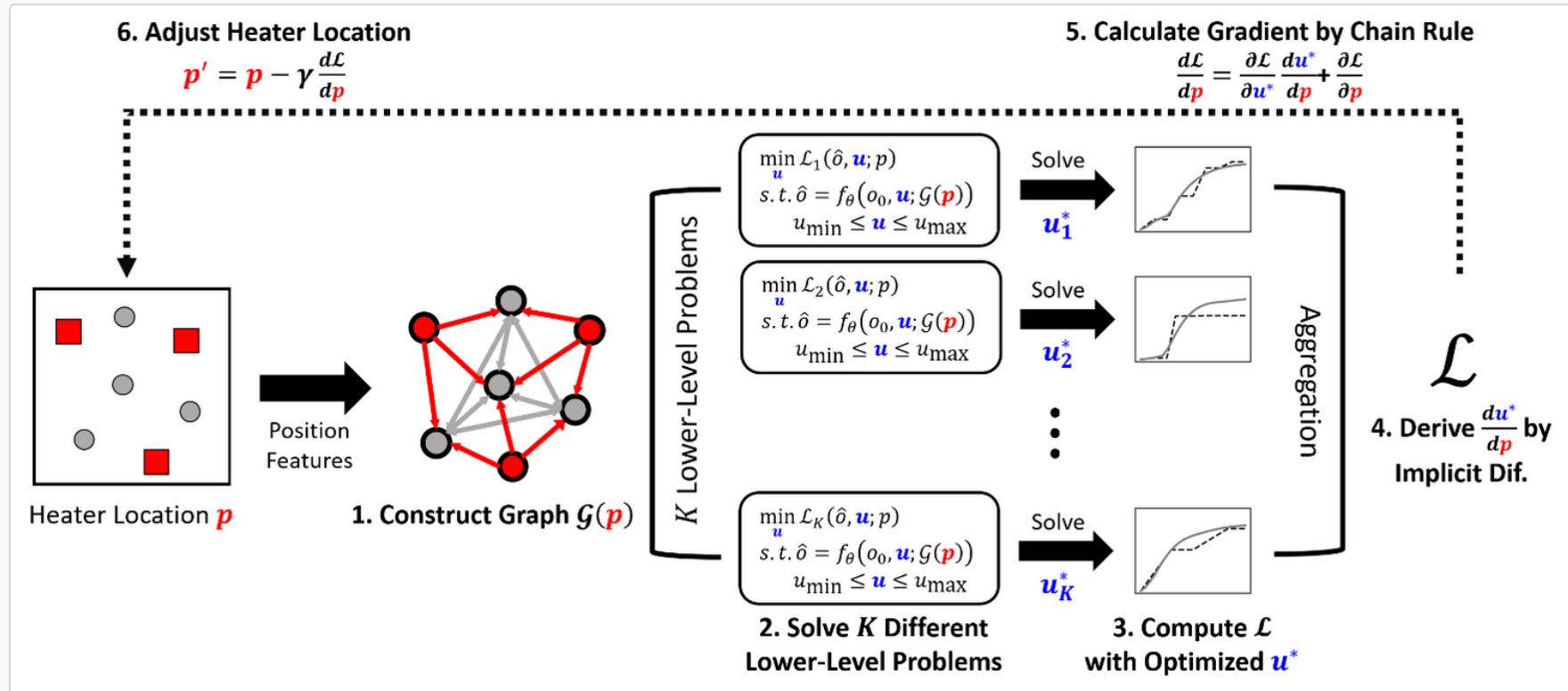
The graph separates operation inputs from geometry



Sensor and heater histories feed a graph model. Geometry enters through a separate path; selected operation inputs pass through convex subnetworks. Source: original PowerPoint, slide 27.

Holding the layout fixed defines the inner problem. For that problem, verify convexity of the reduced cost $J_p(u)$ and of $\mathcal{U}_p$, including the conditions needed when model outputs are composed over time. The network architecture provides part of this argument; the objective and constraints provide the rest.

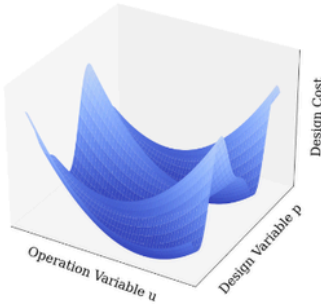
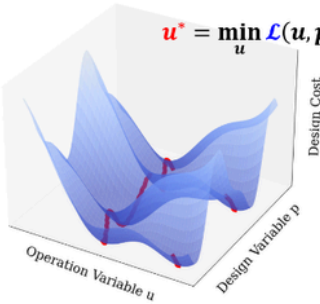
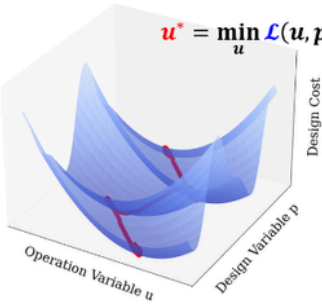
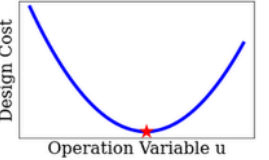
A design update combines several control solves



Build the layout graph, solve K operating scenarios and use their implicit sensitivities to update the heater positions. Source: original PowerPoint, slide 27.

This separates optimisation over operations from optimisation over geometry. A valid implicit derivative requires a sufficiently solved, regular inner problem. It is a sensitivity of the **surrogate problem**; testing the resulting layout on the physical dynamics remains essential.

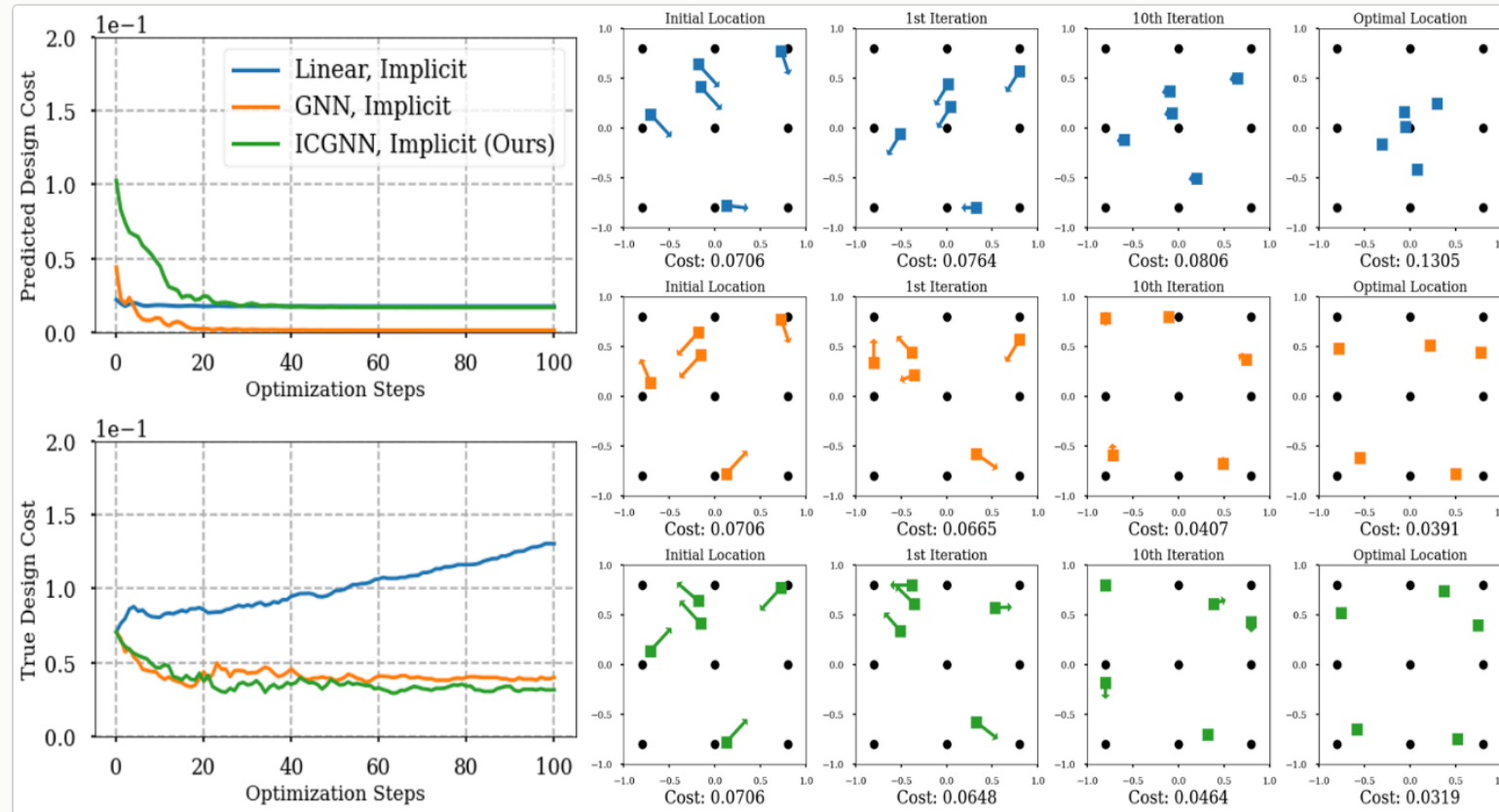
What convexity certifies in the inner problem

	ICGNN + Single-Layer	GNN + Implicit	ICGNN + Implicit (Our Method)
Design Cost Surface			
Optimization Search Space	Operation & Design Variable Space $\mathcal{U} \times \mathcal{P}$	Design Variable Space $\mathcal{P}$	Design Variable Space $\mathcal{P}$
Solve Lower-Level Optimization (Find Optimal u^* with fixed p)	-	 Non-Convex Optimization	 Convex Optimization
Satisfy the Sufficiency of KKT	-	No	Yes

The source compares a joint search, an implicit solve with a general GNN and an implicit solve with an input-convex model. Source: original PowerPoint, slide 25.

Read the last row narrowly: KKT is sufficient for global optimality of a differentiable convex inner problem. In a non-convex problem, a regular local minimum can still be differentiated correctly, but global optimality is not certified. Neither column alone establishes model accuracy or global optimality of the outer design.

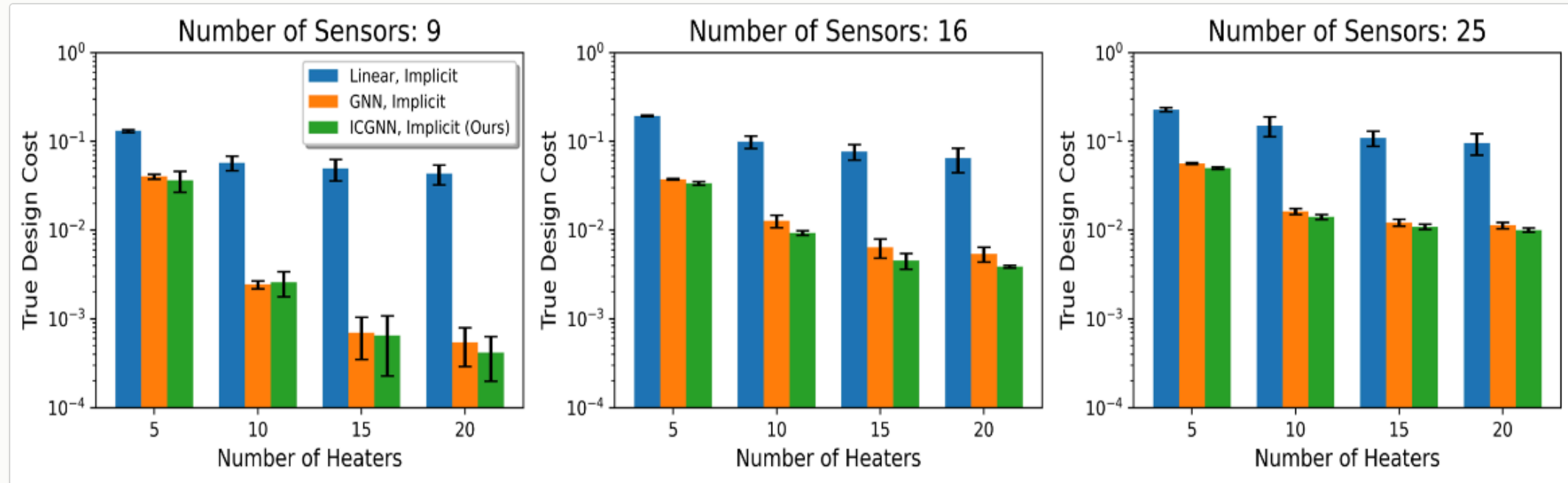
A lower predicted cost can hide a worse design



Predicted cost and true cost, with the heater layouts reached by each model. Source: original PowerPoint, slide 28.

The linear surrogate's predicted cost falls, but its reported true cost rises from **0.0706** to **0.1305**. The final reported costs are **0.0391** for GNN and **0.0319** for ICGNN. This experiment illustrates why a good optimisation result must also survive evaluation on the true dynamics; it is not a universal ranking of model classes.

Compare surrogates using the same optimisation method

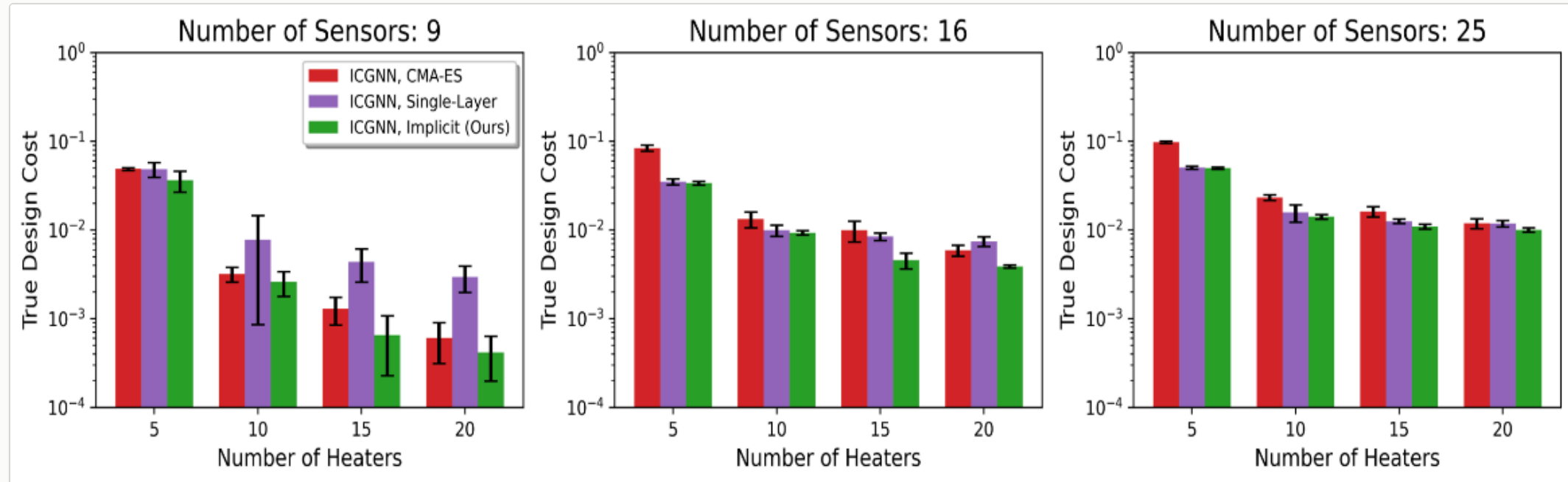


Compare Linear, GNN and ICGNN, all using implicit differentiation. Source: original PowerPoint, slide 28.

Read the logarithmic vertical axes as **true design cost**; lower is better. The panels vary sensor count, and the horizontal axes vary heater count. The linear model performs worse across the plotted settings; the gap between GNN and ICGNN is smaller and varies with the setting.

Holding the optimiser fixed helps isolate the contribution of the **learned surrogate**. The plotted comparisons are empirical results, not an ordering guaranteed by convexity.

Compare optimisers using the same surrogate



Compare CMA-ES, joint single-level search and implicit differentiation, all using ICGNN.

Source: original PowerPoint, slide 28.

Here the model class is fixed, so differences reflect the optimisation procedure and its configuration. The implicit approach is competitive across the plotted settings; read the error bars and the logarithmic scale before judging the size of a difference.

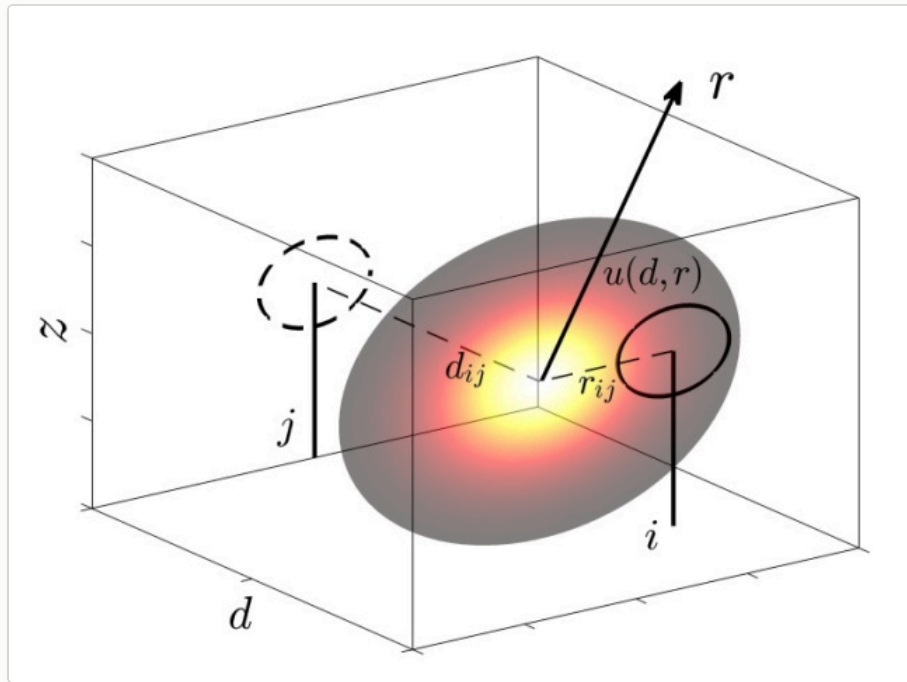
The optimiser can exploit a surrogate's errors. [Evaluate the decision on the system it is meant to improve.](#) Lecture 5 develops this failure in detail.

— EXTENSION

Case study — the full wind-farm optimization

Read after completing the main route.

A wake model turns geometry into a power prediction



Turbine j is upstream of i ; d_{ij} is downstream distance and r_{ij} is the radial offset. Source: original PowerPoint, slide 44.

READ THE MODEL TERM BY TERM

The source uses a smooth wake-deficit profile:

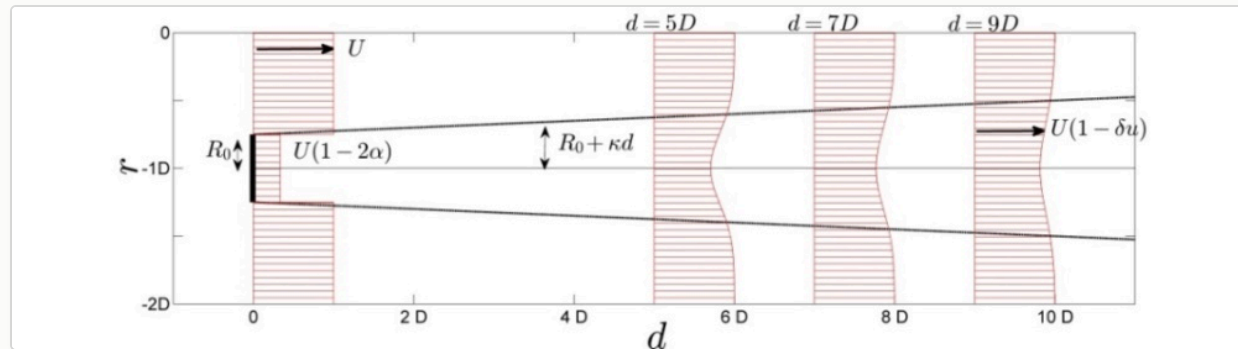
$$\delta u(d, r) = 2\alpha \left(\frac{R_0}{R_0 + \kappa d} \right)^2 e^{-\left(\frac{r}{R_0 + \kappa d} \right)^2}$$

Here R_0 is rotor radius, κ controls wake spreading and α controls extraction. For an upstream turbine, the local speed is modelled as $U(1 - \delta u)$ before combining multiple wakes.

Increasing d spreads and weakens this wake; increasing $|r|$ moves a turbine away from its centre. The optimiser uses these geometry-dependent changes in its power calculation.

Recovering from one wake does not remove the others

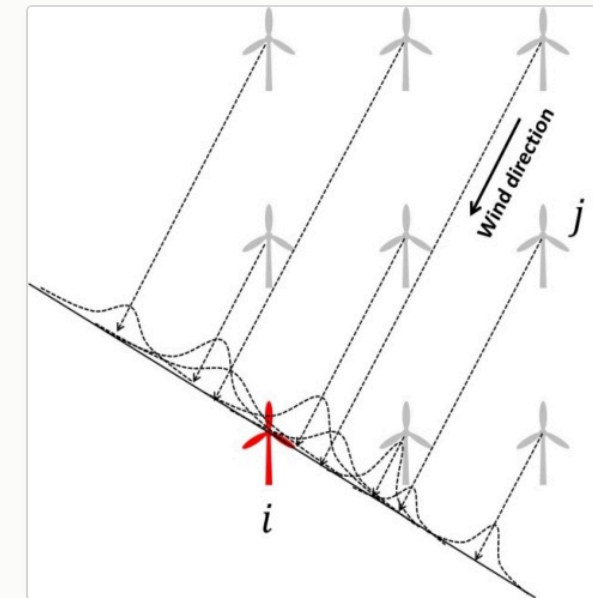
WAKE RECOVERY WITH DISTANCE



The wake widens and its central deficit decreases downstream. Source: original PowerPoint, slide 44.

The profiles at $5D$, $7D$ and $9D$ describe different downstream distances; $D = 2R_0$ is rotor diameter. More spacing helps this pair, but land and site boundaries limit how far turbines can move.

SEVERAL UPSTREAM NEIGHBOURS

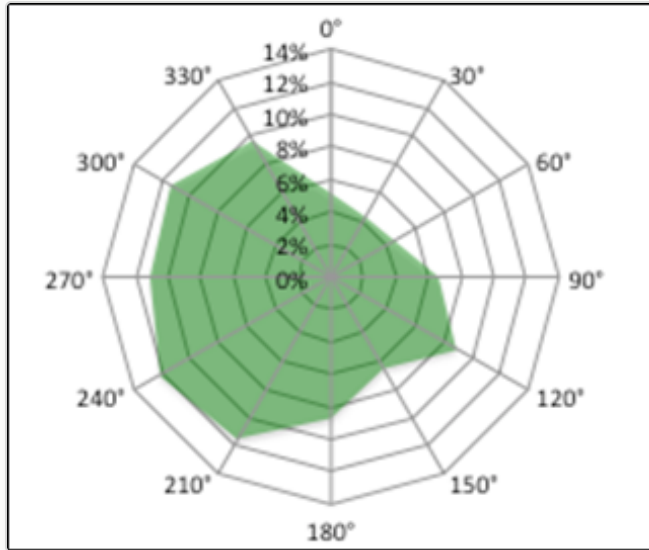


Turbine i can be affected by several upstream wakes. Source: original PowerPoint, slide 44.

The power function $P_i(l; U, \theta)$ depends on the **whole layout and the wind scenario**, not only the nearest neighbour.

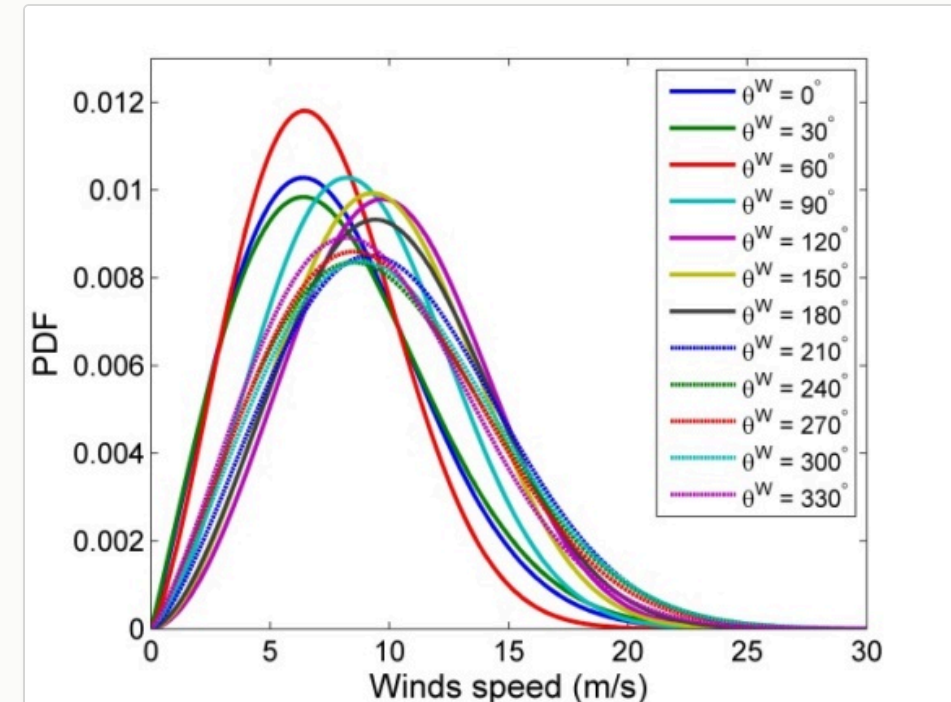
Wind direction and speed determine the scenario weights

HOW OFTEN DOES EACH DIRECTION OCCUR?



The wind rose supplies direction-bin frequencies.
Source: original PowerPoint, slide 45.

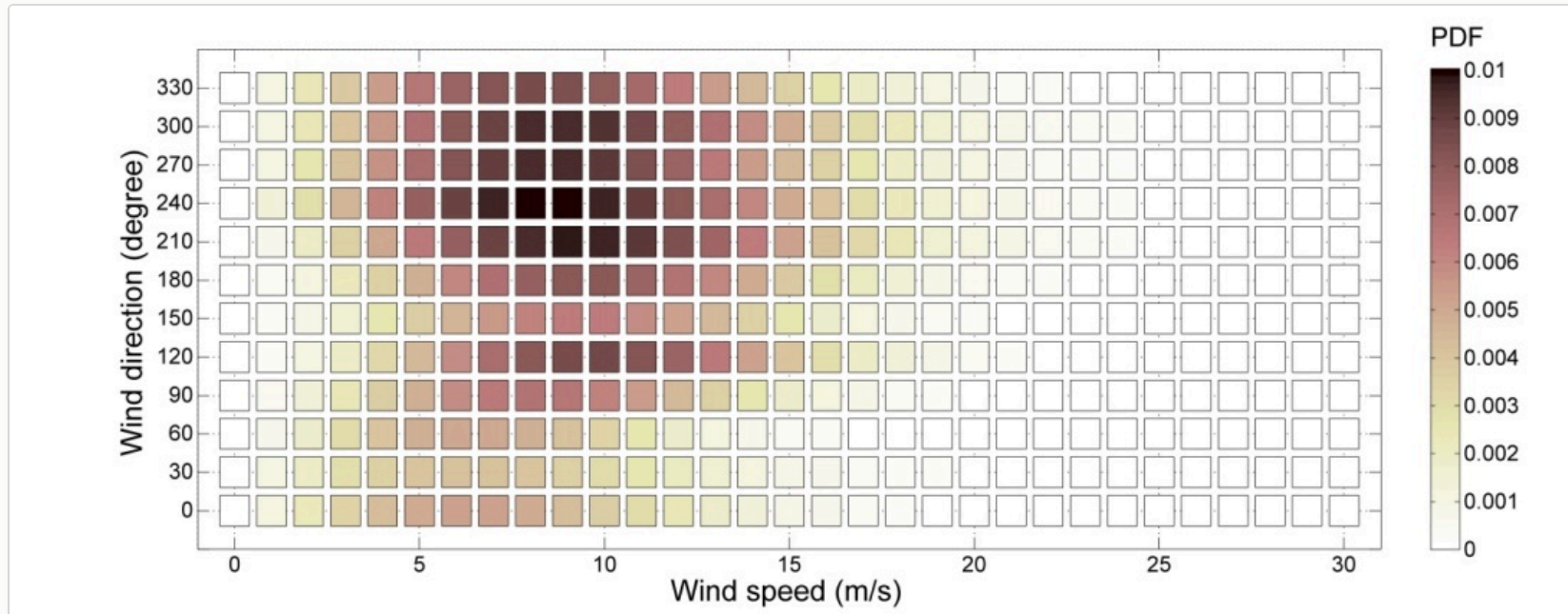
WHAT SPEEDS OCCUR WITHIN THAT DIRECTION?



Conditional Weibull speed densities, one for each direction bin. Source: original PowerPoint, slide 45.

The wind rose and the conditional speed distributions play different roles. A direction that occurs often should receive more weight, and its speed distribution determines how much power that direction can supply.

An expectation is a weighted collection of wind scenarios



Joint wind direction–speed distribution. Integrate the plotted density over each bin to obtain its probability mass. Source: original PowerPoint, slide 45.

$$\pi_{jk} = \Pr(\theta \text{ in bin } k) \Pr(U \text{ in bin } j \mid \theta \text{ in bin } k), \quad \sum_{j,k} \pi_{jk} = 1.$$

Powerful but rare winds need not dominate the expected objective. Use bin masses rather than raw density heights, especially when bins have different widths.

The layout problem, with one consistent separation rule

Let $l_i \in \mathbb{R}^2$ be turbine i 's position. Treat the fitted wind distribution and the wake model as fixed inputs to this optimisation:

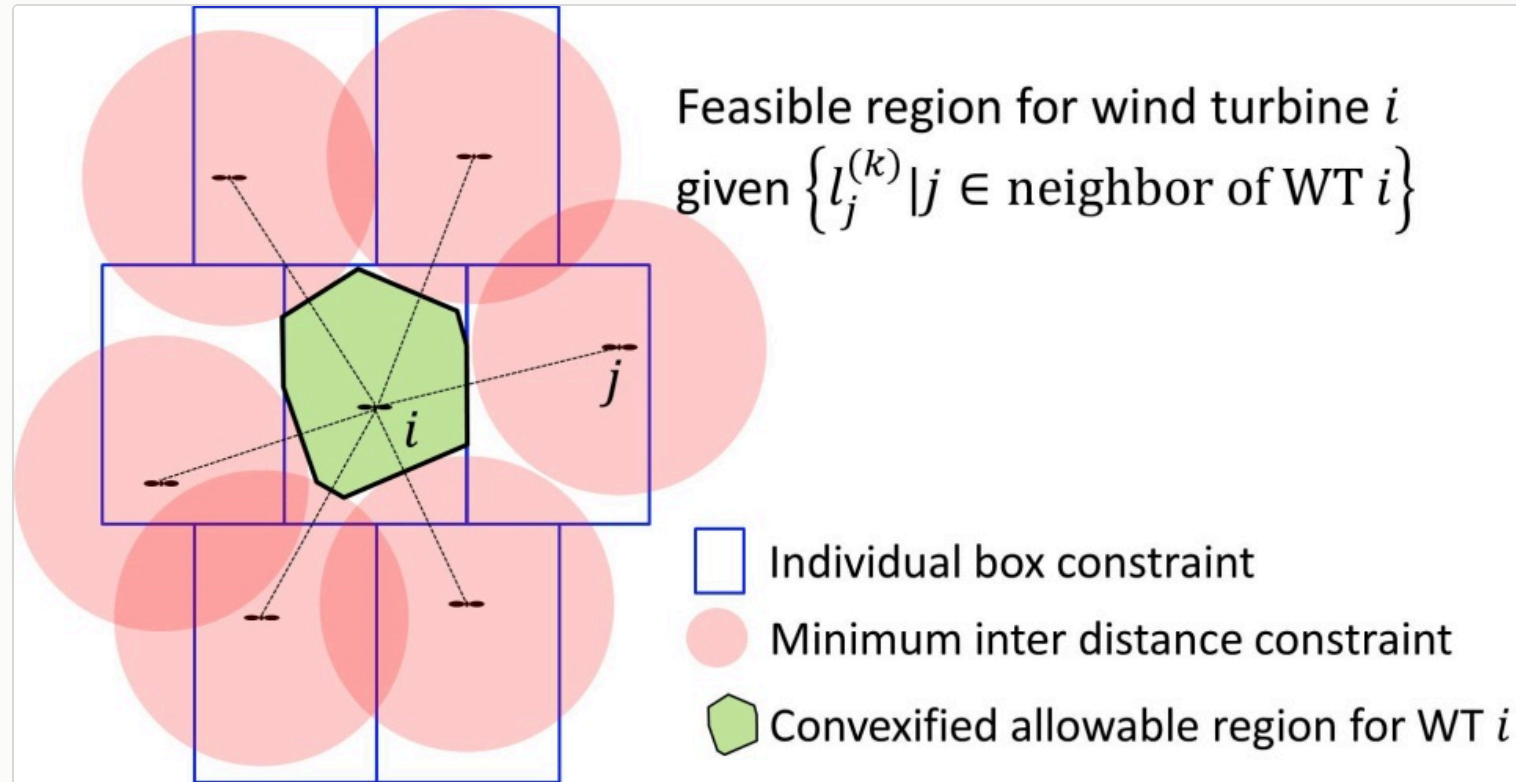
$$\max_l F(l), \quad F(l) = \sum_{j,k} \pi_{jk} \sum_{i=1}^N P_i(l; U_j, \theta_k).$$

$$\text{s.t.} \quad \|l_i - l_j\|_2 \geq d_{\min} \quad (i < j), \quad \underline{c} \leq Cl \leq \bar{c}.$$

The linear constraints encode the site. The minimum-distance constraint excludes a ball around each neighbour, which is non-convex. The wake-coupled objective is also generally non-concave. $d_{\min}$ denotes the **same prescribed distance throughout** the original problem and every subproblem.

A known expectation is still an ordinary objective function. Here the challenge is its **geometry**; uncertainty about the fitted model itself comes next in Lecture 2.

A convex inner region keeps turbines out of exclusion zones



Pink discs exclude positions too close to neighbours. The green polygon is a conservative convex region; blue boxes restrict movement. Source: original PowerPoint, slide 49.

This picture holds neighbouring positions fixed to explain one turbine's allowable motion. The joint algorithm uses relative displacements of both turbines. Its linear constraints should preserve the original separation, not quietly replace it with a smaller distance.

Deriving the safe linearised spacing constraint

Write $d_{ij} = l_i - l_j$ and $d_{ij}^{(k)} = l_i^{(k)} - l_j^{(k)}$. At a feasible iterate $d_{ij}^{(k)} \neq 0$, the convex norm has a global affine lower bound:

$$\|d_{ij}\|_2 \geq \frac{(d_{ij}^{(k)})^\top d_{ij}}{\|d_{ij}^{(k)}\|_2}.$$

Therefore enforce

$$(d_{ij}^{(k)})^\top d_{ij} \geq d_{\min} \|d_{ij}^{(k)}\|_2.$$

Its left side is linear in the new positions. The lower-bound inequality proves that satisfying this constraint also gives $\|d_{ij}\|_2 \geq d_{\min}$.

This is an **inner approximation**: it can exclude useful feasible layouts, but it does not admit layouts that violate the prescribed separation. The current feasible layout remains allowed.

A concave local power model gives a convex subproblem

For this **maximisation**, use a negative semidefinite curvature model $H_k \preceq 0$; an arbitrary approximate Hessian would not suffice.

With $s = l - l^{(k)}$,

$$m_k(l) = F(l^{(k)}) + \nabla F(l^{(k)})^\top s + \frac{1}{2} s^\top H_k s.$$

$$\begin{aligned} \max_l \quad & m_k(l) \\ \text{s.t.} \quad & (d_{ij}^{(k)})^\top (l_i - l_j) \geq d_{\min} \|d_{ij}^{(k)}\|_2, \\ & \underline{c} \leq Cl \leq \bar{c}, \quad \|l - l^{(k)}\|_\infty \leq \rho_k. \end{aligned}$$

The box trust region keeps this a QP: equivalently minimise $-m_k$, whose Hessian is positive semidefinite, over linear constraints. All inequalities are non-strict, so the boundary is available to the solver. The original F remains non-concave.

Accept a step only when the power model earns trust

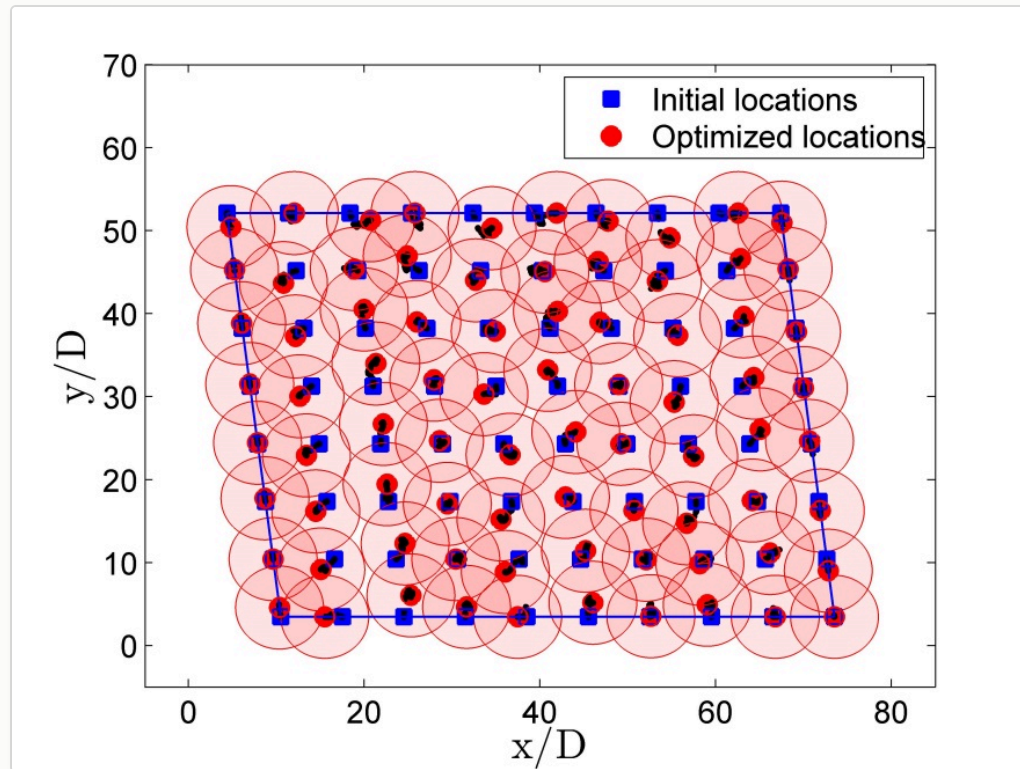
Let $\hat{l}$ solve the subproblem. Compare actual power gain with predicted gain:

$$\text{pred} = m_k(\hat{l}) - F(l^{(k)}), \quad \text{ared} = F(\hat{l}) - F(l^{(k)}), \quad r_k = \frac{\text{ared}}{\text{pred}}.$$

- Form the ratio only when the predicted gain is positive and numerically meaningful.
- Accept when $r_k \geq \eta$ for $0 < \eta < 1$; otherwise keep the current layout and shrink the radius.
- Increase the radius only after strong agreement, typically when a successful step also reaches the trust-region boundary.

A rejected step changes no coordinates. **Zero movement after rejection is not convergence.** Check first-order optimality and feasibility; a tiny radius with a large residual indicates stalling.

The optimised layout breaks repeated wake alignments



Blue squares: initial positions. Red circles: optimised positions. Coordinates are normalised by rotor diameter D . Source: original PowerPoint, slide 50.

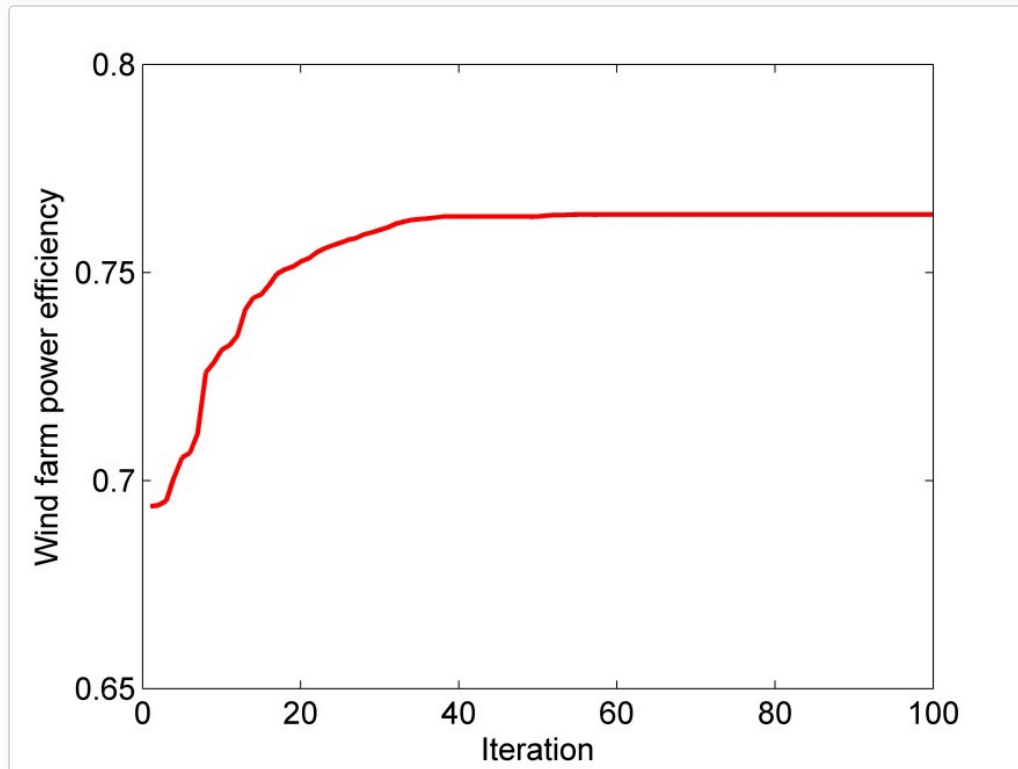
READ THE DISPLACEMENT PATTERN

The optimiser perturbs a regular initial grid while respecting the site's geometry. These movements change the chains of upstream and downstream turbines for common wind directions.

The layout is the **decision**, not the proof of success. It must be evaluated using the same wind scenarios and power model as the initial layout, with separation and boundary constraints checked separately.

These are the source study's reported layouts, not a new simulation. The general derivation uses $d_{\min}$ so that the separation requirement is carried consistently through each step.

The reported efficiency improves, then levels off



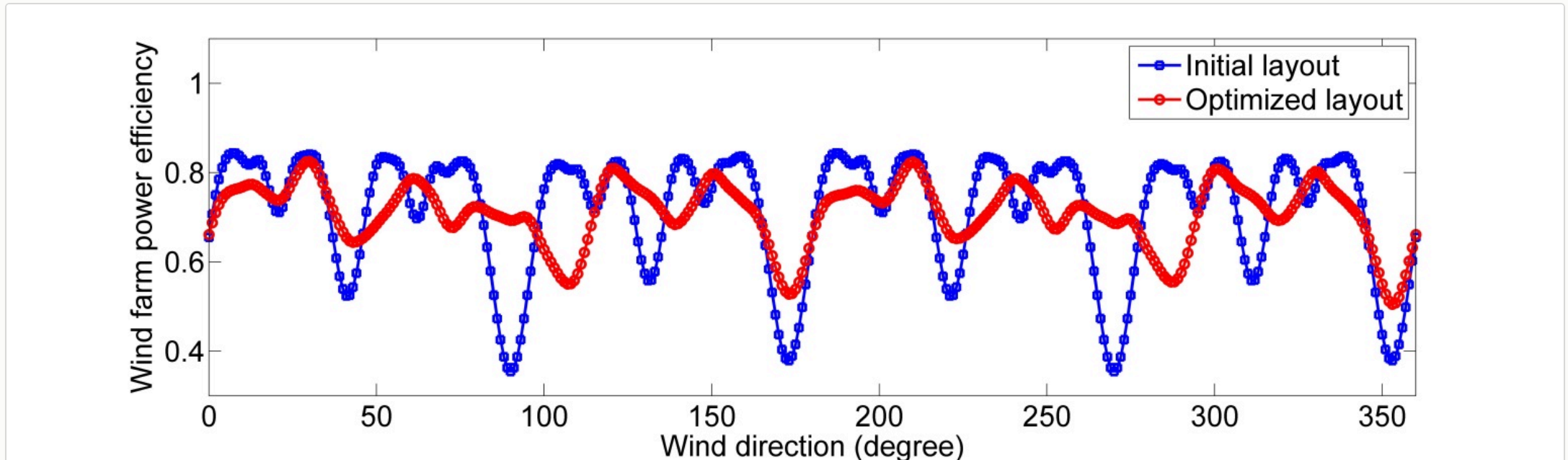
Reported wind-farm power efficiency versus optimisation iteration. Source: original PowerPoint, slide 50.

WHAT THE CURVE SUPPORTS

The plotted efficiency rises from approximately **0.69 to 0.77**, about eight percentage points. Most of the improvement occurs in the early iterations, followed by a plateau.

The curve demonstrates improvement in this modelled case. A plateau alone does not prove global optimality of the original non-convex layout problem, and modelled efficiency is not a measured increase in a real farm's annual energy production.

Better average power does not mean better in every direction



Efficiency by wind direction: initial layout in blue, optimised layout in red. Source: original PowerPoint, slide 50.

Several deep troughs become shallower, while some peaks decrease. The optimiser trades power between directions according to the scenario probabilities. This is compatible with an increase in the weighted average; it is not pointwise improvement for every wind direction.

The case closes the modelling loop: **state the decision, preserve feasibility, solve local models and evaluate the resulting layout.** Convex subproblems do not make the original problem globally convex.

— EXTENSION

Modelling extensions — alternative convex formulations

Read after completing the main route.

Piecewise-linear — the epigraph earns its keep

Example 1.3

A maximum of affine pieces is not linear. Its *epigraph* is, and that turns out to be enough.

THE PROBLEM

Minimise the largest of several affine pieces:

$$\min_x \max_i (a_i^\top x + b_i)$$

The objective is convex — a maximum of affine functions always is — but it is not itself a linear objective. Rewrite it before passing it to an LP solver.

THE REWRITE

Give the maximum a name, then push it down:

$$\min_{x,t} t \quad \text{s.t.} \quad a_i^\top x + b_i \leq t$$

One new variable, one constraint per piece, and it is an LP.

Why it is sound: nothing in the objective rewards a large t , so at the optimum t is squeezed down until it meets the largest piece. Minimising over the epigraph $\{(x, t) : t \geq f(x)\}$ therefore returns exactly $\min_x f(x)$.

The move is general — **any convex objective can be traded for a linear one** over its epigraph. It returns in Act 3, and again as the surrogate in Lecture 10.

Chebyshev centre — an infinity of constraints, collapsed

Example 1.4

Where inside a polyhedron can you put the largest ball? The unknowns are its centre x_c and its radius r .

Let $\mathcal{P} = \{x : a_i^\top x \leq b_i\}$. The ball of radius r about x_c lies inside $\mathcal{P}$ exactly when *every* point of it satisfies *every* face:

$$\sup_{\|u\|_2 \leq r} a_i^\top (x_c + u) \leq b_i \quad \forall i$$

As written that is one constraint for every u in a ball — infinitely many. But the supremum is available in closed form: $a_i^\top u$ is largest when u points along a_i , so it equals $r\|a_i\|_2$ by Cauchy–Schwarz. The infinity collapses to **one linear constraint per face**:

$$\max_{x_c, r \geq 0} r \quad \text{s.t.} \quad a_i^\top x_c + r\|a_i\|_2 \leq b_i \quad \forall i$$

$\|a_i\|_2$ is **data, not a variable** — so the constraint is linear in (x_c, r) , which is all an LP asks.

None of the three looked linear when it was stated

Diet was already an LP and only had to be written down. The piecewise-linear objective needed a new variable. The Chebyshev centre needed an infinite family of constraints reduced by an inequality. **Modelling is the act of finding the rewrite** — and it is where the expertise lives, because the solver is a commodity.

A linear programme with random cost

a non-negative variance penalty preserves convexity

The cost vector c is not known, only its mean $\bar{c}$ and covariance Σ . The objective $c^\top x$ is then itself a random number:

$$\mathbb{E}[c^\top x] = \bar{c}^\top x, \quad \text{Var}(c^\top x) = x^\top \Sigma x.$$

Minimising a random number is not yet a decision, so you must say what its spread is worth. Charge for it linearly:

$$\min_x \bar{c}^\top x + \gamma x^\top \Sigma x$$

γ is the first risk parameter of the course — the price, in units of expected cost, that you put on variance. A covariance matrix is positive semidefinite by construction, since $v^\top \Sigma v = \text{Var}(c^\top v) \geq 0$ for every v . Scaling a convex quadratic by a non-negative number keeps it convex and by a negative one flips it, so $\gamma \geq 0$ **guarantees convexity**. If $\gamma < 0$ and $\Sigma \neq 0$, the objective is not convex on the full space; a zero covariance or a restricted feasible subspace can be a degenerate exception.

$\gamma > 0$ is risk-averse and $\gamma = 0$ risk-neutral, returning the plain LP; risk-*seeking* $\gamma < 0$ produces a concave quadratic objective, generally making its minimisation non-convex. The condition is a modelling decision, not a technicality.

Robust LP — the worst case is convex for free

deterministic — the constraint must hold for every g_i in a set $\mathcal{E}_i$

Now the *constraint* is the uncertain thing rather than the objective. Lecture 0's split between deterministic and stochastic uncertainty gives two honest formulations of it, and the two do not fare alike.

Demand that the constraint survive every outcome in an uncertainty set $\mathcal{E}_i$:

$$g_i^\top x \leq h_i \quad \forall g_i \in \mathcal{E}_i$$

This is not one constraint but one per element of $\mathcal{E}_i$ — infinitely many of them when $\mathcal{E}_i$ is a continuum. Every one is a half-space in x , and an intersection of half-spaces is convex however many there are. The feasible set is therefore convex for **any $\mathcal{E}_i$ whatever**, a nonconvex or disconnected one included.

What the shape of $\mathcal{E}_i$ decides is *tractability*, not convexity. An ellipsoidal $\mathcal{E}_i$ collapses that infinite family into a single second-order cone constraint, $\bar{g}_i^\top x + \|P_i^\top x\|_2 \leq h_i$; a polyhedral one collapses by duality into finitely many linear constraints.

Robust LP — the chance constraint is not

stochastic — the constraint need hold only with probability η

Ask instead that the constraint hold merely often enough:

$$\mathbf{P}(g_i^\top x \leq h_i) \geq \eta$$

For a general distribution that set is **not necessarily convex**. Take g_i Gaussian, though, and it can be written out: $g_i^\top x$ is then a scalar Gaussian with mean $\bar{g}_i^\top x$ and standard deviation $\|\Sigma_i^{1/2} x\|_2$, so standardising the requirement gives

$$\bar{g}_i^\top x + \Phi^{-1}(\eta) \|\Sigma_i^{1/2} x\|_2 \leq h_i.$$

The norm is convex, so the whole is a second-order cone constraint — but only if the number multiplying it is non-negative, and $\Phi^{-1}(\eta) \geq 0$ exactly when $\eta \geq \frac{1}{2}$. Below even odds the sign flips, so this convexity guarantee is lost; special or degenerate instances can still have a convex feasible set.

Not "the usual choices", then, but three conditions: $\gamma \geq 0$, any $\mathcal{E}_i$ at all, and $\eta \geq \frac{1}{2}$. **These are structural convexity guarantees; particular instances may have additional structure.** Lecture 2 takes the stochastic reading much further; Lecture 5 meets the worst-case one again when a surrogate must be trusted only where the data supports it.