

GRADUATE COURSE • OPTIMIZATION & INTELLIGENT SYSTEMS

Multi-Objective Optimization (MOO)

Principles • Pareto optimality • Methods • Evaluation • Applications

Arna Fariza



Learning outcomes

By the end of this lecture, students should be able to...

Formulate

Represent decision variables, constraints, and multiple objective functions mathematically.

Explain

Distinguish dominance, non-dominated solutions, Pareto optimality, and trade-offs.

Compare

Select suitable scalarization, evolutionary, and constraint-based methods.

Evaluate

Interpret Pareto fronts and use quality indicators critically.

Apply

Design a small MOO experiment and justify a final decision using explicit preferences.

From optimization to multi-objective optimization

Optimization

Find a feasible decision that performs well against one or more criteria.

Decision vector

$x = (x_1, x_2, \dots, x_n)$ represents design or decision variables.

Feasible region

$\Omega = \{x : g_i(x) \leq 0, h_j(x) = 0, x \in X\}$. Constraints may be physical, legal, budgetary, or operational.

Single objective

minimize $f(x)$, subject to $x \in \Omega$.

Multi-objective

optimize a vector $F(x) = (f_1(x), f_2(x), \dots, f_n(x))$ rather than one scalar value.

$$\text{minimize } F(x) = [f_1(x), f_2(x), \dots, f_n(x)]^T$$

Objective direction and formulation

Minimization convention

Many algorithms are expressed as minimization; a maximization objective can be transformed using $-f(x)$.

Example

Maximize accuracy and minimize inference time $\rightarrow$ minimize ($-$ accuracy, latency).

Constraints are not objectives

A minimum safety standard can be a constraint, while maximizing safety beyond the threshold may be an objective.

Modeling caution

Transforming direction does not solve scale differences; normalization may still be required.

Why do we need more than one objective?

Drone design

- Maximize flight time
- Maximize image resolution
- Minimize total mass

Healthcare AI

- Maximize sensitivity
- Minimize false alarms
- Minimize inference latency

Energy planning

- Minimize cost
- Minimize emissions
- Maximize reliability

There is often no single design that is best on every criterion. MOO makes the trade-offs explicit and supports a transparent decision.

Conflicting objectives and trade-offs

Conflict

Improving one objective may worsen another because of limited resources, physical constraints, or competing stakeholder needs.

Non-conflicting objectives

Sometimes objectives can improve together; MOO is still valid, but the trade-off may be weak or absent.

Trade-off rate

A local change in one objective may require a measurable sacrifice in another.

Research implication

Do not assume objectives conflict: test the relationship using data, models, or sensitivity analysis.

Pareto dominance: the central comparison rule

For minimization: $x < y$ if $f_i(x) \leq f_i(y)$ for all i , and $<$ for at least one i

Solution A

Cost = 8, error = 4

Solution B

Cost = 9, error = 5

Solution C

Cost = 6, error = 6

A dominates B (better in both objectives). A and C are non-dominated relative to each other: each wins on one criterion.

Pareto-optimal solutions, set, and front

Pareto-optimal solution

A feasible solution that no other feasible solution dominates.

Pareto set

The set of Pareto-optimal decision vectors in decision space.

Pareto front

The corresponding objective vectors in objective space.

Important distinction

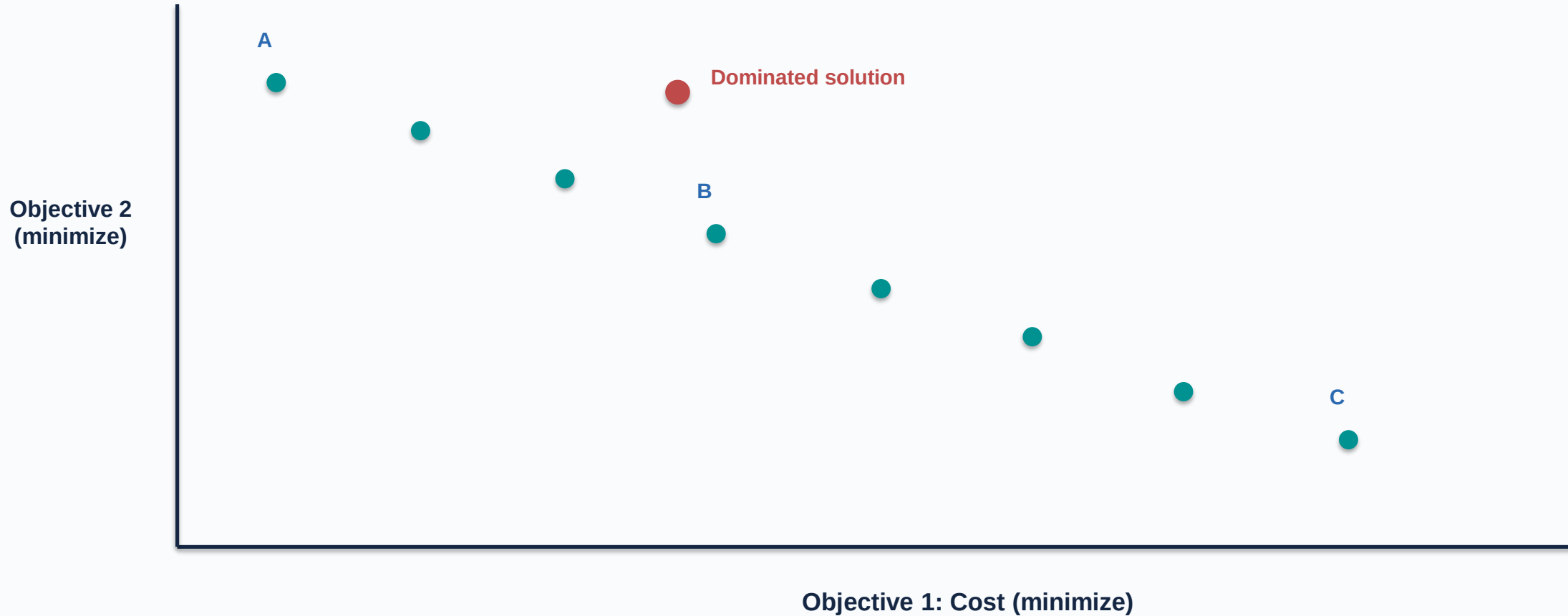
A front is not one universally best answer. It is a collection of efficient alternatives.

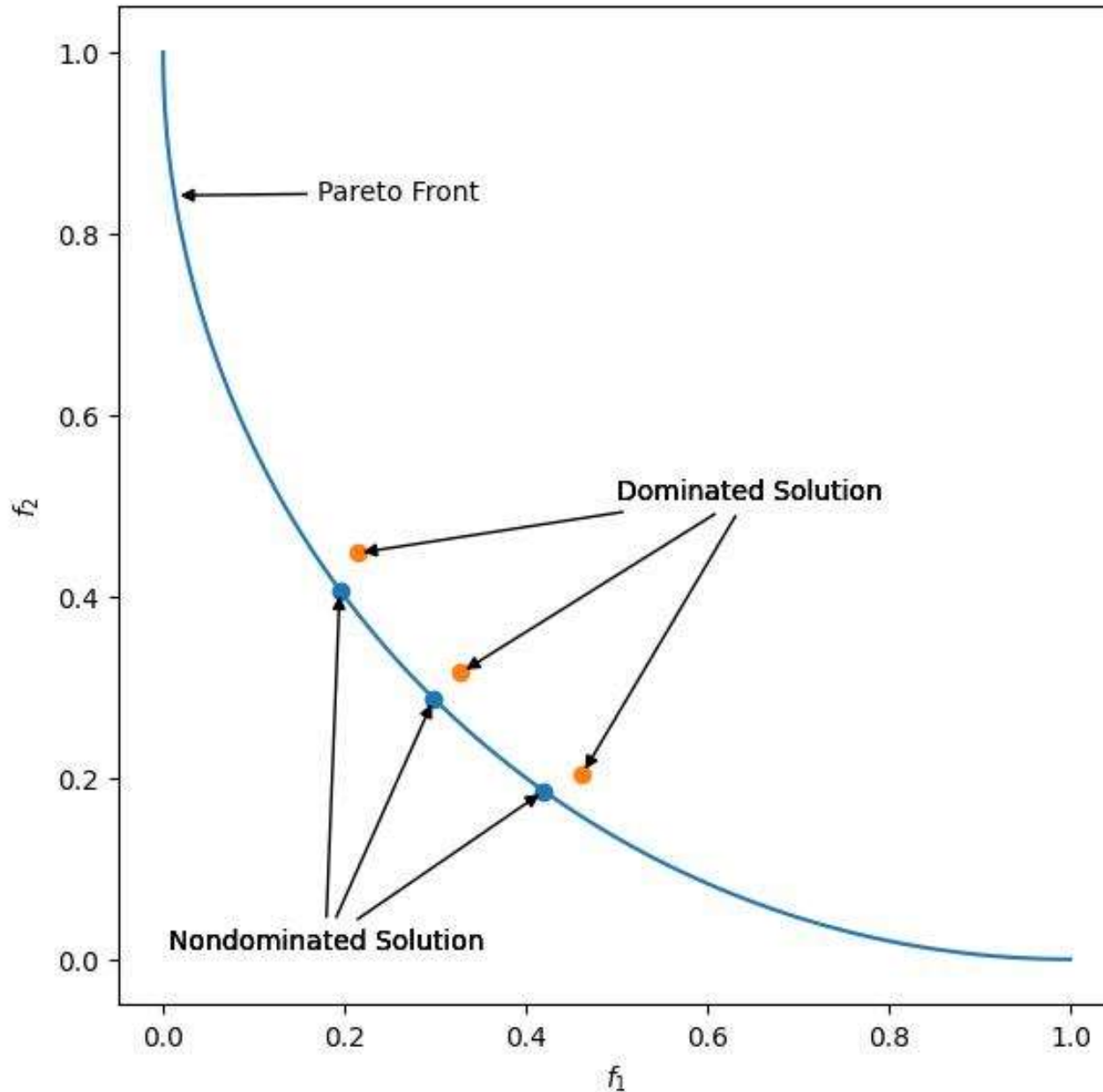
Decision-making after optimization

A stakeholder chooses among Pareto solutions using preferences, constraints, or additional decision analysis.

Reading a two-objective Pareto front

Each point on the front represents a different trade-off; moving along the front improves one objective while worsening another.





Pareto dominance digunakan untuk menyaring solusi yang tidak efisien.

- Dominated solutions: solusi yang dikalahkan oleh setidaknya satu solusi lain.
- Non-dominated solutions: solusi yang tidak didominasi oleh solusi lain dalam himpunan solusi yang sedang diperiksa.
- Pareto front: representasi solusi-solusi non-dominated dalam ruang objektif.

A practical MOO workflow



Iterate: problem formulation, algorithm parameters, and stakeholder preferences may need revision after inspecting the first Pareto approximation.

Main families of MOO methods

Scalarization

Combine objectives into one score.
Examples: weighted sum, Tchebycheff.

ϵ -constraint

Optimize one objective while bounding the others.

Pareto-based

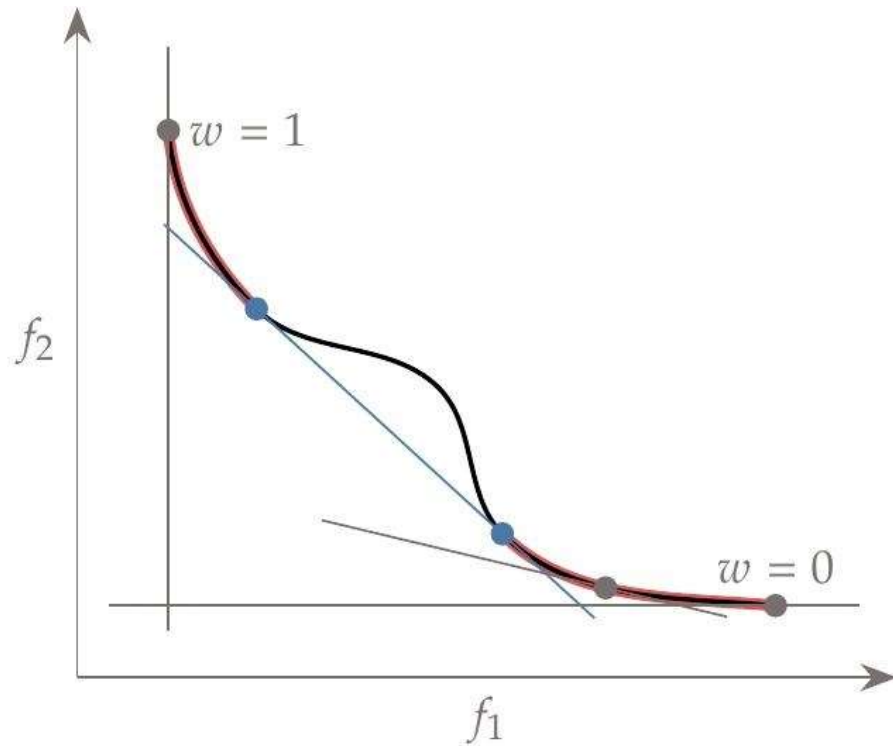
Search directly for non-dominated alternatives.

Interactive

Update search using decision-maker feedback.

Method choice depends on objective count, variable types, convexity, computational budget, preference information, and whether one solution or a set is needed.

Scalarization methods



Metode ini menggabungkan berbagai objektif menggunakan bobot sehingga masalah MOO dapat diselesaikan sebagai masalah optimasi satu objektif (single-objective optimization).

Contoh: sebuah perusahaan ingin meminimalkan biaya produksi dan memaksimalkan kualitas produk.

Kita definisikan:

- $f_1(x)$: biaya produksi yang ingin diminimalkan.
- $f_2(x)$: kualitas produk yang ingin dimaksimalkan.

Agar keduanya menjadi masalah minimisasi, kualitas dapat ditulis sebagai $-f_2(x)$. Selanjutnya, gunakan weighted sum:

$$\min F(x) = w_1 f_1(x) - w_2 f_2(x)$$

dengan $w_1 + w_2 = 1$ dan $w_i \geq 0$.

Jika bobot biaya lebih besar, optimasi akan lebih memprioritaskan biaya berdasarkan fungsi dan skala yang digunakan.

Scalarization methods

Metode yang termasuk keluarga ini:

- Weighted Sum Method
- Weighted Tchebycheff Method

Kelebihan: sederhana, mudah diimplementasikan, dan dapat menggunakan algoritma optimasi yang sudah tersedia.

Keterbatasan: hasil sangat dipengaruhi oleh bobot dan skala objektif. Weighted sum standar juga tidak dapat menemukan semua solusi Pareto-optimal pada non-convex Pareto front.

Weighted-sum scalarization

$$\text{minimize } F(\mathbf{x}) = \sum_i w_i \cdot \tilde{f}_i(\mathbf{x}), \quad w_i \geq 0, \quad \sum_i w_i = 1$$

$\tilde{f}_i(\mathbf{x})$ denotes a suitably scaled objective; normalization is important when units differ.

Weights express relative importance in the scalar score, but are not automatically equivalent to intuitive percentages of value.

Repeat optimization for multiple weight vectors to sample different trade-offs.

Strength: simple, interpretable, and compatible with many solvers.

Limitation: for non-convex Pareto fronts, some Pareto-optimal points cannot be found by any positive weighted sum.

Why normalization matters

Normalized objective: $\tilde{f}_i(\mathbf{x}) = (f_i(\mathbf{x}) - z_i^{\min}) / (z_i^{\max} - z_i^{\min})$

Without normalization

Cost may range from 1,000–10,000 while error ranges from 0–1. Cost can dominate the weighted sum simply due to units.

With normalization

Objectives are placed on comparable ranges before applying weights. Estimated bounds should be chosen carefully.

Use reliable ideal/nadir estimates when available; min–max scaling can be unstable if the estimated range is poor.

Worked example: cost versus quality

Let $x \in [0, 4]$ represent a product-design setting.

Cost to minimize: $f_1(x) = 10x^2 + 5$.

Quality to maximize: $q(x) = 20 - 2x$.

Convert quality to minimization: $f_2(x) = -q(x) = -20 + 2x$.

Weighted sum: $F(x) = w_1(10x^2 + 5) + w_2(-20 + 2x)$, with $w_1 + w_2 = 1$.

Worked example: derive the minimizer

$$F(x) = 10w_1x^2 + 2w_2x + 5w_1 - 20w_2$$

$$dF/dx = 20w_1x + 2w_2 = 0$$

$$x^* = -w_2 / (10w_1)$$

If x^* lies outside $[0,4]$, the constrained optimum is at the nearest feasible boundary.

For $w_1 = 0.7$ and $w_2 = 0.3$, $x^* = -0.0429$, so the constrained solution is $x = 0$.

The example exposes a modeling issue: the chosen quality function decreases with x , so it may not express the intended cost–quality conflict.

ϵ -constraint method

Core idea

Optimize one objective and convert the remaining objectives into constraints.

Formulation

minimize $f_1(x)$, subject to $f_2(x) \leq \epsilon_2, \dots, f_m(x) \leq \epsilon_m$, and $x \in \Omega$.

How it explores the front

Vary ϵ thresholds to obtain alternative trade-offs.

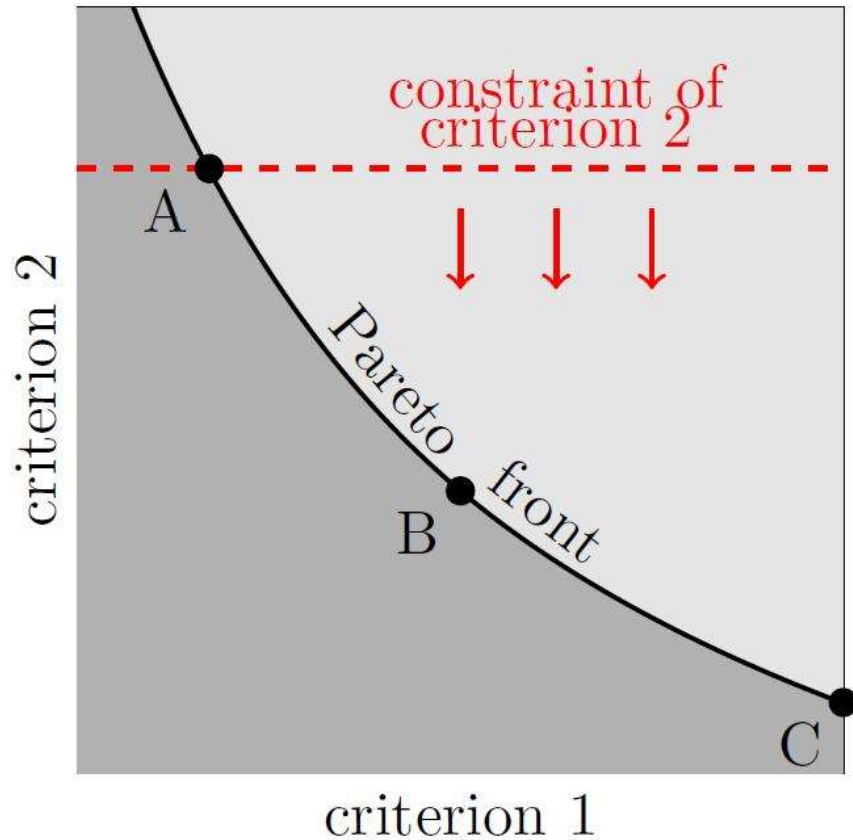
Advantages

Can recover non-convex Pareto points and gives explicit performance thresholds.

Limitations

Requires sensible ϵ values; infeasible thresholds can return no solution.

ϵ -constraint method



satu objektif dioptimalkan, sedangkan objektif lainnya dijadikan batasan.

Misalnya, kita ingin meminimalkan biaya produksi, tetapi kualitas produk harus minimal 90.

Formulasinya:

$$\begin{array}{ll} \min & f_1(x) \\ \text{subject to} & f_2(x) \geq 90 \\ & x \in \Omega \end{array}$$

Biaya merupakan objektif utama, sedangkan kualitas menjadi batasan.

Dengan mengubah nilai ambang ϵ , kita dapat mengeksplorasi berbagai alternatif solusi.

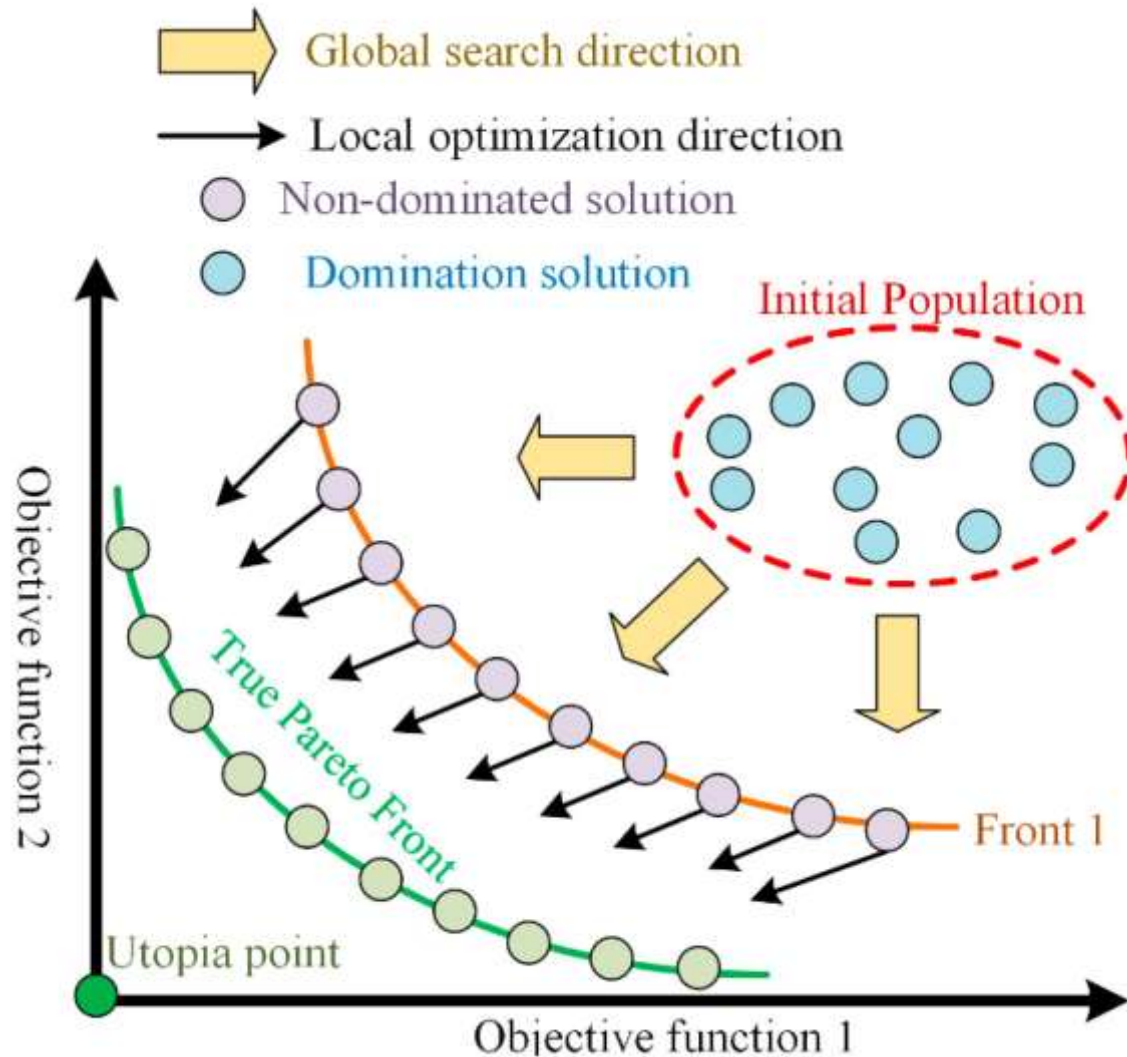
Untuk objektif yang diminimalkan, batasnya umumnya ditulis $f_2(x) \leq \epsilon$ untuk objektif yang dimaksimalkan, bentuk pertidaksamaannya menyesuaikan arah optimasi.

ϵ -constraint method

Kelebihan: batasan dapat mewakili persyaratan nyata, seperti kualitas minimum, anggaran maksimum, atau batas emisi. Metode ini juga dapat menemukan bagian Pareto front non-konveks yang tidak dapat ditemukan oleh weighted sum standar.

Keterbatasan: penentuan ambang yang tepat membutuhkan pemahaman terhadap masalah. Ambang yang terlalu ketat dapat menghasilkan masalah tidak feasible.

Pareto-based methods



Metode ini tidak harus mengubah semua objektif menjadi satu nilai. Algoritma mengevaluasi berbagai kandidat solusi dan mempertahankan alternatif yang memberikan trade-off menarik.

Salah satu metode yang populer adalah NSGA-II (Non-dominated Sorting Genetic Algorithm II), yang menggunakan prinsip algoritma evolusioner.

Kelebihan: sesuai untuk masalah kompleks, nonlinier, non-diferensiabel, dan memiliki banyak alternatif solusi.

Keterbatasan: membutuhkan banyak evaluasi objektif dan tidak menjamin bahwa Pareto front sebenarnya telah ditemukan secara sempurna.

Why evolutionary algorithms are popular for MOO

Population-based search

Maintains multiple candidate solutions in one run, rather than returning only one optimum.

Black-box compatibility

Can handle non-differentiable, discontinuous, simulation-based, or mixed-variable objectives.

Diversity

Selection mechanisms can preserve solutions across different parts of the Pareto front.

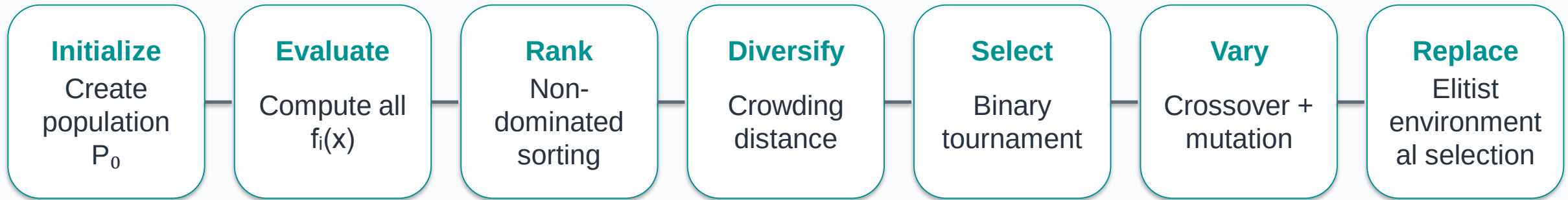
Trade-offs

They are often computationally expensive and do not guarantee the true Pareto front in finite time.

Examples

NSGA-II, SPEA2, MOEA/D, and multi-objective variants of differential evolution and particle swarm optimization.

NSGA-II: high-level procedure



At each generation, parents and offspring are combined; the next population is chosen by Pareto rank first and crowding distance second.

NSGA-II: non-dominated sorting

Front F_1

Solutions not dominated by any other solution in the current population.

Front F_2

Solutions dominated only by members of F_1 , but not by other members of F_2 .

Subsequent fronts

Repeat the process to assign each solution a Pareto rank.

Selection rule

Lower rank is preferred. Rank alone cannot distinguish solutions within the same front.

Caution

Ranks are relative to the current population, not proof of global Pareto optimality.

NSGA-II: crowding distance and diversity

Purpose

Estimate how isolated a solution is among neighboring solutions in objective space.

Mechanism

For each objective, sort solutions; boundary solutions receive a large/infinite distance, while interior distances are normalized by objective range.

Selection

Within the same non-dominated rank, larger crowding distance is preferred.

Why it matters

It reduces premature concentration in one region and helps maintain a spread of trade-offs.

Limitation

Crowding distance is a local diversity estimate and may be less informative with many objectives.

Case study: vehicle design

Decision variables

Vehicle mass, engine setting,
tire design, safety-system
configuration

Objectives

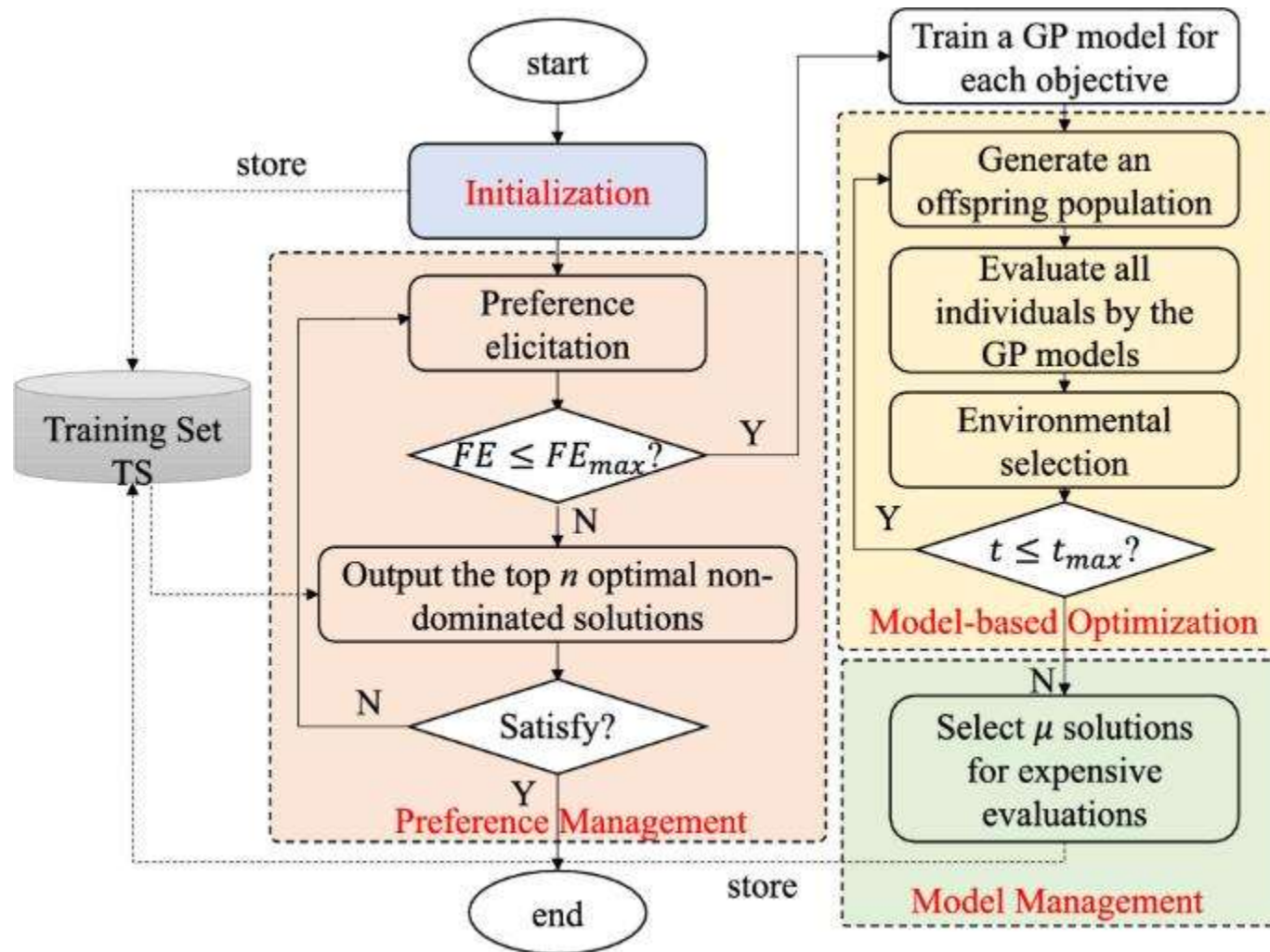
Minimize fuel consumption
Minimize lifecycle cost
Maximize crash-safety score

Constraints

Regulatory safety threshold,
emissions limits, package size,
budget

NSGA-II can return a portfolio of feasible designs. Engineers then inspect the Pareto front and select a design consistent with regulation, budget, and risk tolerance.

Interactive methods



Metode ini menggabungkan pencarian solusi dengan masukan preferensi dari pengguna atau pakar, bukan hanya menentukan preferensi sebelum optimasi dimulai.

Kelebihan: cocok ketika preferensi pengambil keputusan belum diketahui sepenuhnya sejak awal.

Keterbatasan: memerlukan waktu dan keterlibatan pakar. Hasil juga dapat dipengaruhi oleh perubahan atau inkonsistensi preferensi.

Interactive methods

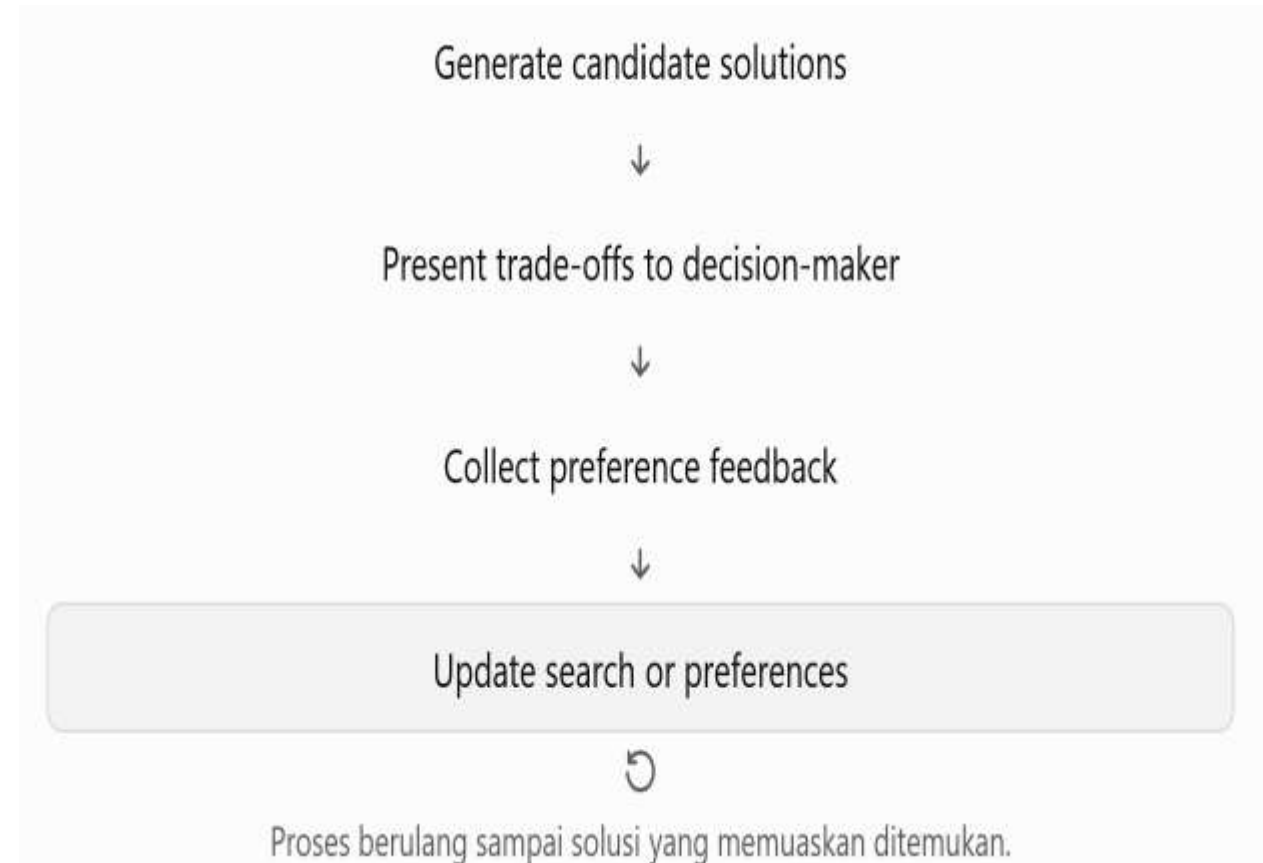
Contoh: sebuah rumah sakit ingin memilih model AI untuk mendeteksi penyakit.

Awalnya, algoritma menghasilkan beberapa kandidat model dengan perbedaan sensitivitas, waktu inferensi, dan kebutuhan memori.

Dokter kemudian memberikan masukan bahwa sensitivitas lebih penting daripada kecepatan.

Berdasarkan masukan tersebut, sistem mempersempit atau mengarahkan pencarian ke solusi yang lebih sesuai.

Pengambil keputusan dapat memberikan masukan kembali sampai diperoleh alternatif yang dapat diterima.



Bagaimana memilih metode yang tepat?

- Scalarization: Jika bobot kepentingan setiap objektif sudah diketahui dan kita hanya memerlukan satu solusi untuk setiap kombinasi bobot.
- ϵ -constraint: Jika terdapat persyaratan yang eksplisit, misalnya biaya maksimal Rp100 juta atau sensitivitas minimal 95%.
- Pareto-based methods: Jika masalah kompleks dan kita ingin menemukan banyak alternatif solusi untuk dianalisis, misalnya menggunakan NSGA-II.
- Interactive methods: Jika keputusan membutuhkan masukan pakar dan preferensinya dapat berkembang setelah melihat hasil optimasi.

Method comparison: when should you use what?

Method	Useful when	Key limitation
Weighted sum	Weights are meaningful; fast scalar solver exists	Misses unsupported points on non-convex fronts
ϵ-constraint	Thresholds have real-world meaning	Choosing ϵ values can be difficult
NSGA-II	Black-box problem; need a diverse solution set	Many objective evaluations may be required
MOEA/D	Many-objective problems with decomposable subproblems	Performance depends on decomposition and neighborhoods
Interactive MOO	Decision-maker can provide iterative preferences	Requires stakeholder time and consistent feedback

Evaluating Pareto-front quality

Convergence

How close is the approximation to the true or reference Pareto front?

Diversity

How evenly and widely are solutions distributed across the front?

Hypervolume (HV)

Measures the objective-space volume dominated by the approximation relative to a reference point; larger is generally better for minimization under consistent settings.

Inverted Generational Distance (IGD)

Measures distance from reference-front points to the approximation; lower is better, but results depend on reference-front quality.

Spacing / spread

Summarizes distribution uniformity or extent; no single metric captures every aspect.

Fair comparison

Use identical evaluation budgets, reference points, normalization, repeated random seeds, and statistical summaries.

The many-objective challenge

Dominance loses discrimination

As the number of objectives grows, a large fraction of solutions may become mutually non-dominated.

Implications

Selection pressure weakens, visualization becomes difficult, and stakeholder interpretation becomes harder.

Possible responses

Objective reduction, preference articulation, reference-point methods, decomposition, indicator-based selection, or dimensionality-reduction visualization.

Important question

Are all objectives genuinely necessary, or are some redundant proxies for the same underlying concern?

Uncertainty, robustness, and sensitivity

Uncertain evaluations

Simulation noise, incomplete data, changing environments, and model error can alter objective values.

Robust MOO

Optimize expected performance, worst-case outcomes, variance, or a risk measure depending on the application.

Sensitivity analysis

Vary weights, constraints, model assumptions, and algorithm parameters; check whether preferred solutions remain stable.

Reproducibility

Report random seeds, parameter settings, stopping criteria, objective definitions, and computational budgets.

Research-level insight

A visually attractive Pareto front is not enough if its ranking changes drastically under plausible uncertainty.

Applications across disciplines

Healthcare AI

Sensitivity vs. specificity;
accuracy vs. latency; fairness vs.
predictive utility.

Energy & climate

Cost vs. emissions vs. reliability.

Finance

Expected return vs. risk vs.
liquidity.

Engineering

Mass vs. strength vs. durability.

Supply chain

Cost vs. delivery time vs. service
level.

Machine learning

Model size vs. performance vs.
energy consumption.

Mini case: selecting a clinical AI model

Decision

Choose a model configuration for an assistive diagnostic system.

Objectives

Maximize sensitivity; minimize false-positive rate; minimize inference latency; minimize memory footprint.

Constraints

Sensitivity must exceed a clinically agreed threshold; inference must run on available hardware; validation data must represent intended use.

Pareto interpretation

A larger model may improve sensitivity but increase latency and memory use; a compact model may be deployable but miss more cases.

Governance

The final choice requires clinical validation, subgroup analysis, human oversight, and explicit acceptance criteria—not optimization metrics alone.

Common modeling and reporting pitfalls

Combining objectives with different units without normalization.

Treating weights as objective truth rather than preference assumptions.

Reporting one algorithm run without repeated seeds or evaluation budget.

Calling every non-dominated solution globally Pareto-optimal without qualification.

Showing a Pareto plot without explaining objective directions and units.

Selecting the final solution without documenting stakeholder preferences or constraints.

Ignoring uncertainty, feasibility, computational cost, and deployment requirements.

In-class exercise (small groups)

Problem

Design a lightweight AI model for mobile deployment.

Objectives:

- maximize validation F1-score
- minimize latency (ms)
- minimize model size (MB)

Constraints:

- F1-score ≥ 0.85
- model size ≤ 100 MB

Tasks

1. Define variables and objective directions.
2. Explain how you would normalize objectives.
3. Identify two dominated and two non-dominated candidate models from a hypothetical table.
4. Choose weighted sum, ϵ -constraint, or NSGA-II and justify.
5. State how you would evaluate front quality and choose a final model.

Deliverable: a one-page problem formulation and a short defense of your method choice.

Key takeaways

MOO is a decision-support framework

It reveals trade-offs rather than forcing every criterion into one apparently objective score.

Pareto dominance is fundamental

It identifies inefficient alternatives without requiring a complete preference ranking.

Methods have different strengths

Weighted sum is simple; ϵ -constraint supports explicit thresholds; NSGA-II searches for diverse Pareto approximations.

Evaluation must be multi-dimensional

Convergence, diversity, feasibility, uncertainty, and computational budget all matter.

Optimization is not the final decision

Stakeholder preferences, safety, ethics, and deployment constraints determine which Pareto solution is appropriate.

Thank you