Reliability Optimization — how many redundant copies of each block to install

Menu: System ▸ Reliability Optimization...

Part 1 — What the function does

What it calculates. You have drawn a system on the RBD canvas (chapter 4.1). You can make it more reliable by adding identical copies of some blocks in parallel - a second pump beside the first, a third fan beside two - and each copy costs money. Reliability Optimization tells you how many copies of each block to install, for one of two objectives:

Objective	What ReliaEn finds
Maximize reliability within budget	The design with the highest system reliability at the mission time whose added cost stays within your budget
Minimize cost to reach target reliability	The cheapest design whose system reliability reaches your target

ReliaEn does not search by trial. It evaluates every design in the space you define - every combination of copy numbers from 1 to the maximum you allow for each block - exactly, through the same reliability algebra the RBD evaluation uses, so no design in that space beats the optimum it reports. The report gives the optimal design, its system reliability R_sys(T) and its added cost, the rule that decided it, the Pareto frontier (the designs for which no other design is both cheaper and more reliable), the steps of the classic greedy allocator on the same space for comparison, and exact checks of the result.

From which data. The diagram on the RBD canvas, with each block's life distribution. The form adds, for each plain block, the most copies you would install (Max copies n) and the price of one added copy (Cost per copy), then the objective, the mission time, and the budget or the target. No worksheet.

What a cost means here. Costs price added copies only — the base system is sunk. The drawn system costs 0; a design's added cost is (copies − 1) × Cost per copy, summed over the blocks.

When an engineer uses it. When the choice is how many identical copies of each block to add in parallel, and every copy has a price.

When not to use it.

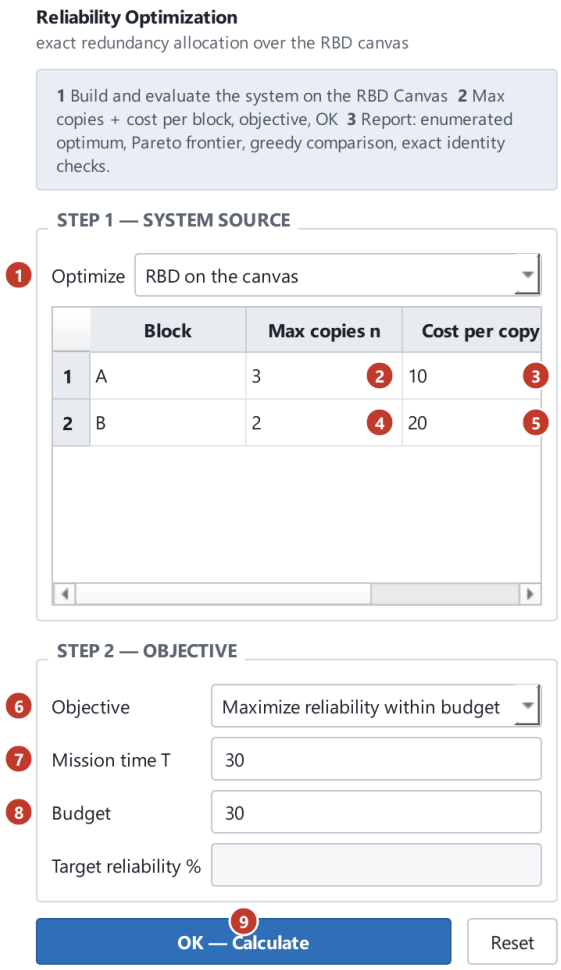
Situation	Use instead
You want only the reliability of the diagram as drawn	Evaluate it on the canvas: System ▸ RBD Canvas (chapter 4.1)
Repairs, repair crews or preventive maintenance matter	System ▸ RAM Simulation... (chapter 4.5)
You want to size a stock of spare parts	System ▸ Spare Parts Optimization... (chapter 5.3)
You want to change the number of units inside a k-of-n or standby group	Not covered in version 1.0.0: such a group shows group — fixed in v1 in the block table

Part 2 — The example's input data

Setting. A plant has two stages in series. Stage A has a constant failure rate of 0.02 per hour; up to 3 copies of it can be installed, at 10 baht per added copy. Stage B has a constant failure rate of 0.01 per hour; up to 2 copies, at 20 baht per added copy. A mission lasts 30 hours, and the plant has 30 baht to spend. Times are in hours.

Where	Entry
The RBD canvas	+ Add Block A, Exponential (λ) 0.02; + Add Block B, Exponential (λ) 0.01; the lines START → A → B → END
Row A of the block table	Max copies n 3, Cost per copy 10
Row B of the block table	Max copies n 2, Cost per copy 20
Objective	Maximize reliability within budget
Mission time T	30
Budget	30
File in the worked-example zip	none - the plant is drawn on the canvas as Part 3 shows

The form set for this example, as Part 3's steps 3 to 10 set it:

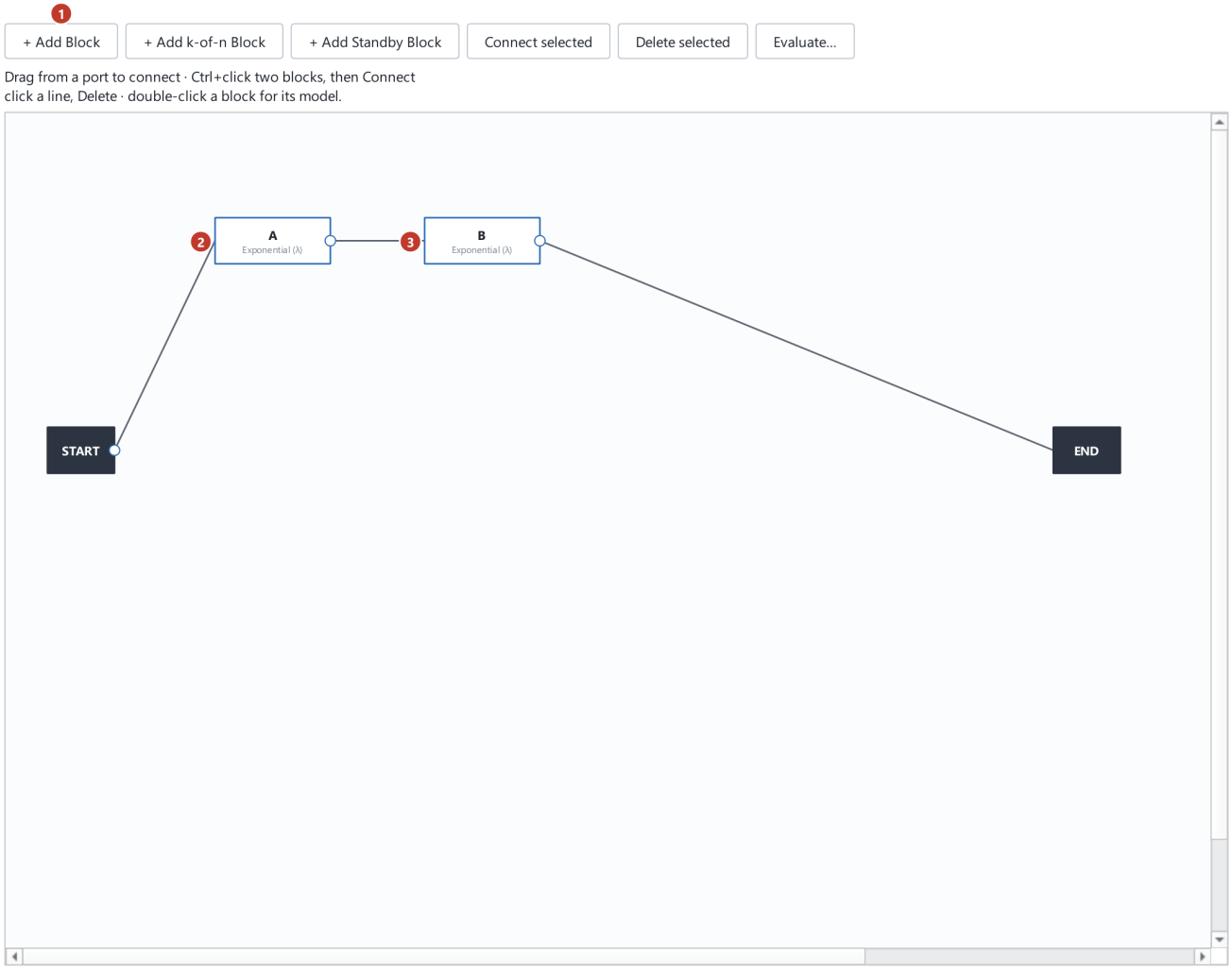


The example's values in the form

- ① Optimize RBD on the canvas.
- ② to ⑤ The block table: row A Max copies n 3, Cost per copy 10; row B 2 and 20.
- ⑥ to ⑧ Objective Maximize reliability within budget, Mission time T 30, Budget 30.
- ⑨ OK — Calculate.

Part 3 — How to use it, step by step

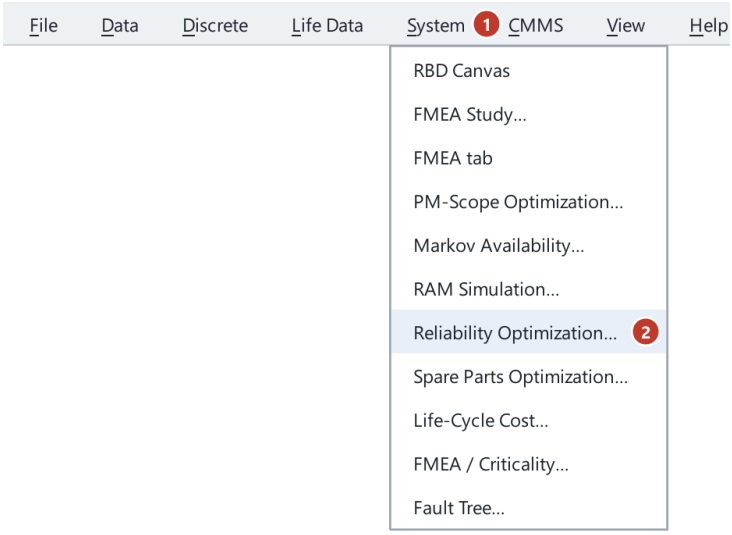
1. **Draw the plant on the RBD canvas** as chapter 4.1 shows: on the main tab RBD Canvas, + Add Block A with Exponential (λ) 0.02 in its Block properties, + Add Block B with Exponential (λ) 0.01, and the lines START → A → B → END.



Step 1 — the plant on the canvas

- ① + Add Block: adds one block, after asking for its name.
- ② A, the first stage, with its model Exponential (λ) under the name.
- ③ B, the second stage.

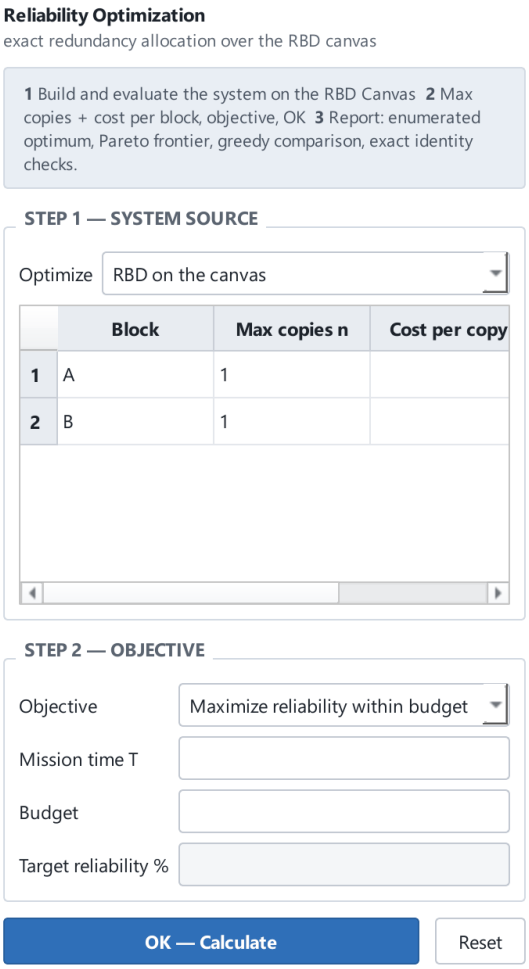
2. **Choose System ▸ Reliability Optimization...** in the menu bar. The form Reliability Optimization opens in the left panel, its block table holding one row per block of the canvas.



Step 2 — the menu path

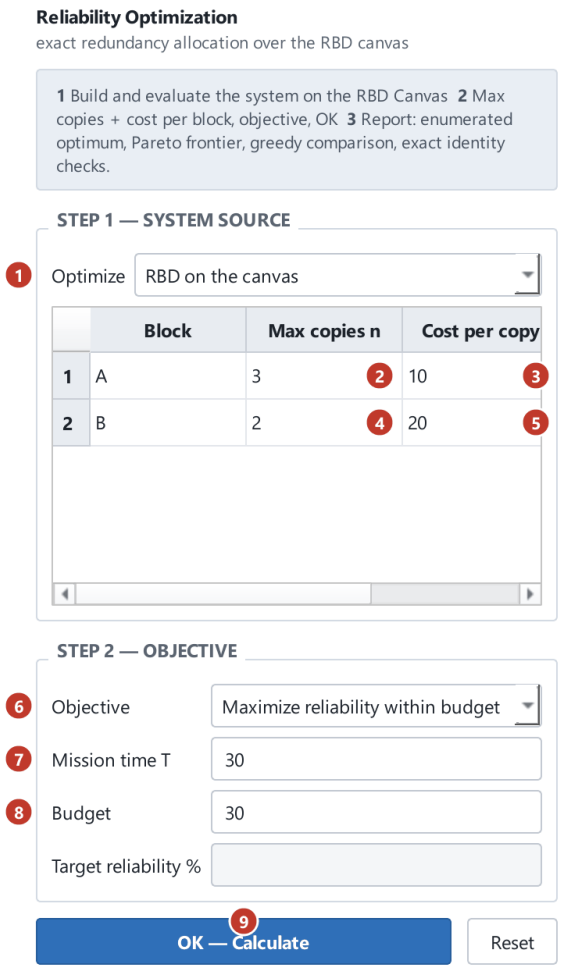
- ① System: the menu.
- ② Reliability Optimization...: the entry to click.

The form as it opens, rows A and B at Max copies n 1 - the plant as drawn:



Step 2 — the form as it opens

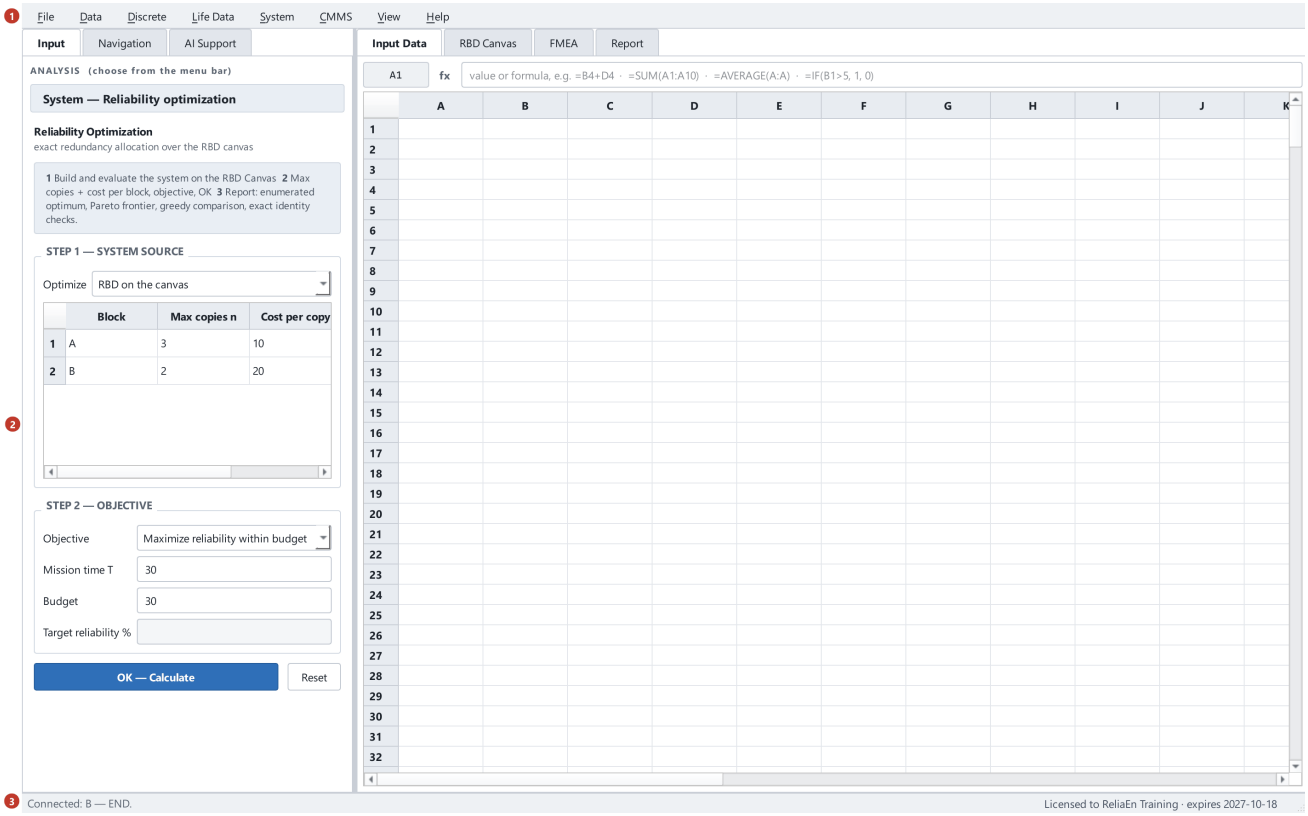
Steps 3 to 10 set the form, marked ① to ⑧; ⑨ is the button of step 11:



Steps 3 to 10 — the form set for this example

3. ① **Check that Optimize reads RBD on the canvas**, the diagram drawn on the canvas now - the only source in version 1.0.0.
4. ② **In row A, type 3 under Max copies n.** A whole number, 1 or more.
5. ③ **In row A, type 10 under Cost per copy.** The price of one added copy, greater than 0.
6. ④ **In row B, type 2 under Max copies n.**
7. ⑤ **In row B, type 20 under Cost per copy.**
8. ⑥ **Check that Objective reads Maximize reliability within budget.**
9. ⑦ **In Mission time T, type 30.**
10. ⑧ **In Budget, type 30.** The most to spend on added copies.

11. **Click OK — Calculate** at the foot of the form (② in the picture below). ReliaEn evaluates all 6 designs and shows the report.



Step 11 — the window just before OK

- ① The menu bar: File, Data, Discrete, Life Data, System, CMMS, View, Help.
- ② The form, set as in steps 3 to 10; OK — Calculate at its foot.
- ③ The status bar, with Licensed to ReliaEn Training · expires 2027-10-18 on the right.

Part 4 — The report, card by card

INPUT DATA

1	System	2 searched block(s)
2	Objective	Maximize reliability within budget
3	Mission time T	30
4	Budget	30
5	Designs enumerated	6

Input data

- ① System 2 searched block(s). How many blocks ReliaEn searched and how many groups it held fixed.
- ② Objective Maximize reliability within budget. The objective of this run.
- ③ Mission time T 30. The mission time at which R_sys(T) is evaluated.
- ④ Budget 30. The most the run may spend on added copies.
- ⑤ Designs enumerated 6. How many designs ReliaEn evaluated, every one of them.

RESULTS

1	Optimal design	`A ×2 / B ×2`
2	R_sys(T)	`0.74292878`
3	Band	`none of the winner's blocks carry fitted uncertainty (manual parameters).`
4	Added cost	`30`
5	Binding	`budget (added cost 30 of 30)`

Results

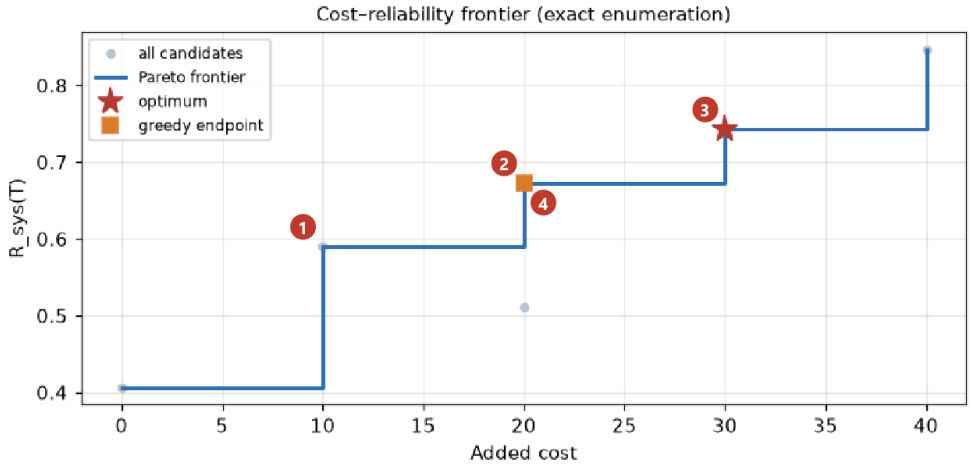
- ① Optimal design A ×2 / B ×2. The design to install: the number of copies of each searched block.
- ② R_sys(T) 0.74292878. The system reliability of that design at the mission time.
- ③ Band none of the winner's blocks carry fitted uncertainty (manual parameters).. Whether R_sys(T) carries an uncertainty bracket, and why it does not here.
- ④ Added cost 30. What the added copies cost.
- ⑤ Binding budget (added cost 30 of 30). The rule that decided the design, with its figures.

INTERPRETATION

1 Every candidate in the design space was evaluated exactly through the verified reliability algebra — the optimum is enumerated, not searched heuristically, so it provably cannot lose to any design in the space. The greedy card below shows what the classic allocator would have chosen on the same space.

Interpretation

- ① Every candidate in the design space was evaluated exactly through the verified reliability algebra — the optimum is enumerated, not searched heuristically, so it provably cannot lose to any design in the space. The greedy card below shows what the classic allocator would have chosen on the same space.



Interactive Plot

Plot

- ① The points: all candidates.
- ② The line: Pareto frontier.
- ③ The points: optimum.
- ④ The points: greedy endpoint.

OPTIMAL DESIGN — CONVENTIONS

1 Costs price added copies only — the base system is sunk. Install and fixed costs are a named v2.
Added copies fail independently but share the block's fitted model — parameter uncertainty is common to all copies and does not average out; common-cause failures are not modeled.

Optimal design — conventions

- ① `Costs price added copies only — the base system is sunk. Install and fixed costs are a named v2.

PARETO FRONTIER (5 UNDOMINATED DESIGNS OF 6)

1. The frontier plot's points ARE these rows — same numbers, stated.
Frontier rows are point estimates; the winner's band above carries the fitted blocks' uncertainty.

2 Design	3 Added cost	4 R
5 A ×1 / B ×1	0	0.40656966
6 A ×2 / B ×1	10	0.59000916
7 A ×3 / B ×1	20	0.67277493
8 A ×2 / B ×2	30	0.74292878
9 A ×3 / B ×2	40	0.84714593

Pareto frontier (5 undominated designs of 6)

- ① `The frontier plot's points ARE these rows — same numbers, stated.
- ② Design
- ③ Added cost
- ④ R
- ⑤ A ×1 / B ×1 · 0 · 0.40656966
- ⑥ A ×2 / B ×1 · 10 · 0.59000916
- ⑦ A ×3 / B ×1 · 20 · 0.67277493
- ⑧ A ×2 / B ×2 · 30 · 0.74292878
- ⑨ A ×3 / B ×2 · 40 · 0.84714593

GREEDY COMPARISON — MARGINAL-GAIN ALLOCATOR (STOPPED: NO AFFORDABLE ADDITION)

1. the classic allocator left 0.07015386 on the table
Greedy can never beat the enumerated optimum (identity J5).

2 Step	3 Added	4 Design	5 Added cost	6 R	7 ΔR
8 1	A	A ×2 / B ×1	10	0.59000916	0.18343950
9 2	A	A ×3 / B ×1	20	0.67277493	0.08276577

Greedy comparison — marginal-gain allocator (stopped: no affordable addition)

- ① `the classic allocator left 0.07015386 on the table
- ② Step
- ③ Added
- ④ Design
- ⑤ Added cost
- ⑥ R
- ⑦ ΔR
- ⑧ 1 · A · A ×2 / B ×1 · 10 · 0.59000916 · 0.18343950
- ⑨ 2 · A · A ×3 / B ×1 · 20 · 0.67277493 · 0.08276577

EXACT IDENTITY CHECKS — EVERY CHECK IS ==, NEVER A TOLERANCE

① J2a (route agreement, reliability(T) == evaluate()) is pytest-resident on the key plant.

② Id	③ Check	④ Verdict
⑤ J1	optimum unbeaten over 6 candidates	holds
⑥ J2	all-ones design == the base tree exactly	holds
⑦ J4	monotone increments; frontier strictly increasing	holds
⑧ J5	greedy endpoint cannot beat the optimum	holds

Exact identity checks — every check is ==, never a tolerance

- ① J2a (route agreement, reliability(T) == evaluate()) is pytest-resident on the key plant.
- ② Id
- ③ Check
- ④ Verdict
- ⑤ J1 · optimum unbeaten over 6 candidates · holds
- ⑥ J2 · all-ones design == the base tree exactly · holds
- ⑦ J4 · monotone increments; frontier strictly increasing · holds
- ⑧ J5 · greedy endpoint cannot beat the optimum · holds

Part 5 — How to interpret the results

What to install. A ×2 / B ×2 - one extra copy of each stage. It gives R_sys(30) 0.74292878 for an added cost of 30 baht, the whole budget (Binding budget (added cost 30 of 30)). The plant as drawn, A ×1 / B ×1, survives the 30 hours with probability 0.40656966 only (e^(−0.02 × 30) × e^(−0.01 × 30)).

Why this design and not another. ReliaEn evaluated all 6 designs (3 copy numbers for A × 2 for B) exactly, and the check J1 reads optimum unbeaten over 6 candidates. Each copy of a stage multiplies that stage's failure probability: two copies of A fail together with probability $(1 - e^{(-0.6)})^2 = 0.2036$, so the stage survives with 0.7964, and two copies of B survive with 0.9328; in series 0.7964 × 0.9328 = 0.7429.

What one step further costs. The next frontier design, A ×3 / B ×2, reaches 0.84714593 for an added cost of 40 baht - above the budget. Raise the budget by 10 baht only if that gain is worth it to the plant.

What a simple method would have done. The greedy allocator adds the copy with the largest gain per baht first: a copy of A (+0.18343950), then another copy of A (+0.08276577), reaching A ×3 / B ×1 at 0.67277493 for 20 baht - and then cannot afford a copy of B. It leaves 0.07015386 of reliability on the table; the enumerated optimum does not.

What the result does not include. Installing costs and fixed costs are not priced; copies of a block fail independently, and common-cause failures - one cause stopping a block and all its copies at once - are not modelled. If such a cause exists, the real gain of a copy is smaller.