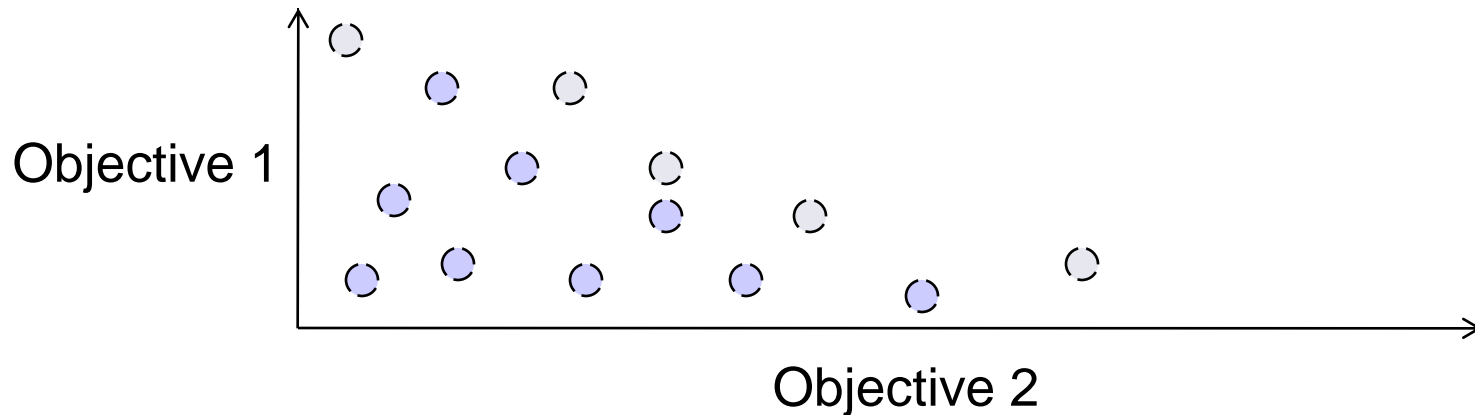


Optimisation with EAST-ADL

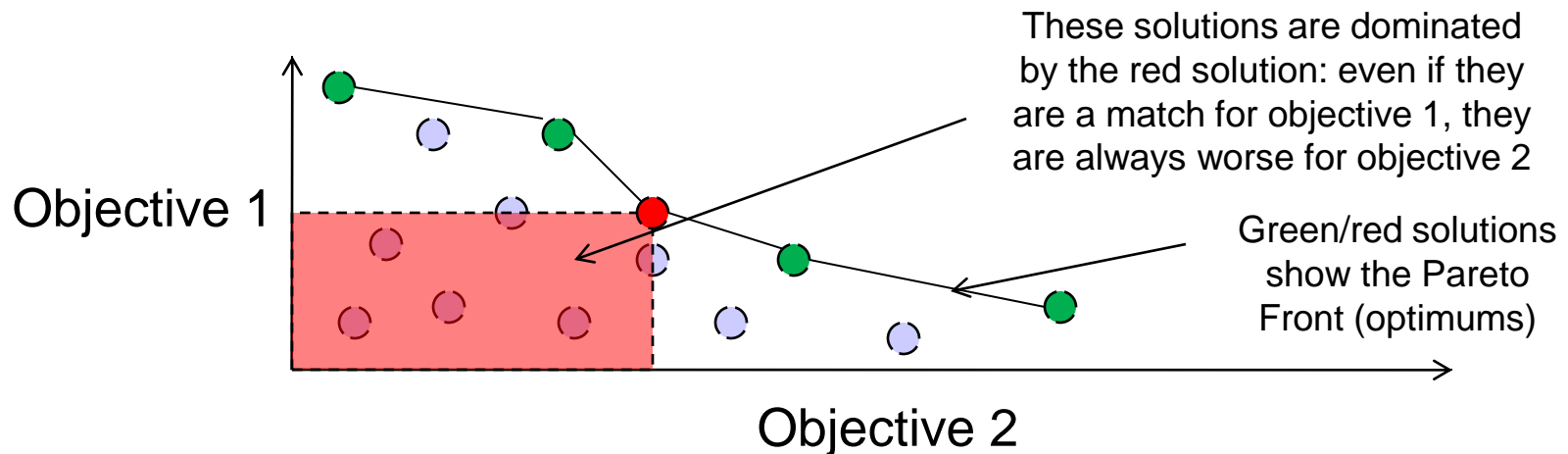
Architecture Optimization

- Optimization is the process of finding the best solution(s) from all feasible solutions
 - A solution is evaluated against one or more objectives
 - It can also be subject to certain constraints
- A multi-objective optimization searches for the optimum solutions which have the best trade-offs for each objective



Pareto Optimality

- Multiple objectives means there is no single “best” solution
- Instead, the best solutions are optimal trade-offs
 - A solution may be better in one objective but worse in another
 - However, for one given objective value, it is better than all others with that same objective value
- The result is the ‘Pareto’ set:



Analysis for Architecture Optimization

- Functional requirements can be met by many designs
 - Each has different extra-functional characteristics
- Analysis offers results for one architectural scenario
 - But the space of possible architecture alternatives may be huge!
- This is often a multi-objective problem
 - Designs need analysing for each objective (e.g. cost and reliability)
- Leverage analysis for architecture optimization
 - Formalize the design space for architecture exploration
 - Reduce the number of possible solutions (design constraints)
 - Avoid dead ends paths that do not lead to a solution
 - Use knowledge of experienced designers (methodological support)

Design Space Description

- The design space (or search space) is the set of all possible feasible solutions to be explored
 - Must be defined before optimization can occur
- Can be achieved by providing different alternative parameters for model elements or attributes (e.g. cost)
 - Some may conflict, e.g. cost increases as safety increases
- Can be achieved in EAST-ADL using variability
- Important to maintain **substitutability**
 - For an optimization to be viable, it must always be possible to substitute one parameter option/variant for another

Optimization Objectives

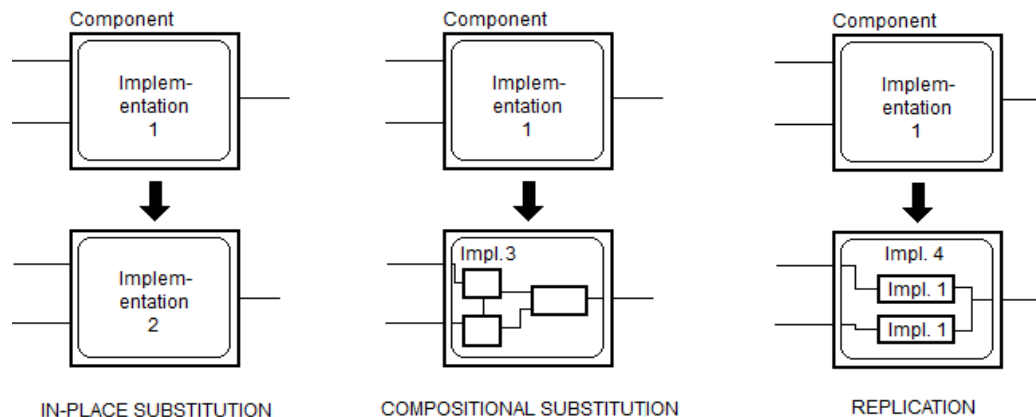
- For safety and timing analysis, objective can be to maximize margins on constraints
 - i.e. to exceed safety or timing goals by as much as possible
 - The constraint becomes a minimum that the optimization tries to surpass
- For timing analysis, objectives can include:
 - End-to-end response times vs deadlines
 - Resource reservation vs resource utilization
 - Maximum jitters vs jitter constraints
- For safety analysis, objectives can include:
 - Cost vs unavailability (probability that system will be inoperable)
 - Meeting safety constraints with a minimum cost

Design/Search Space

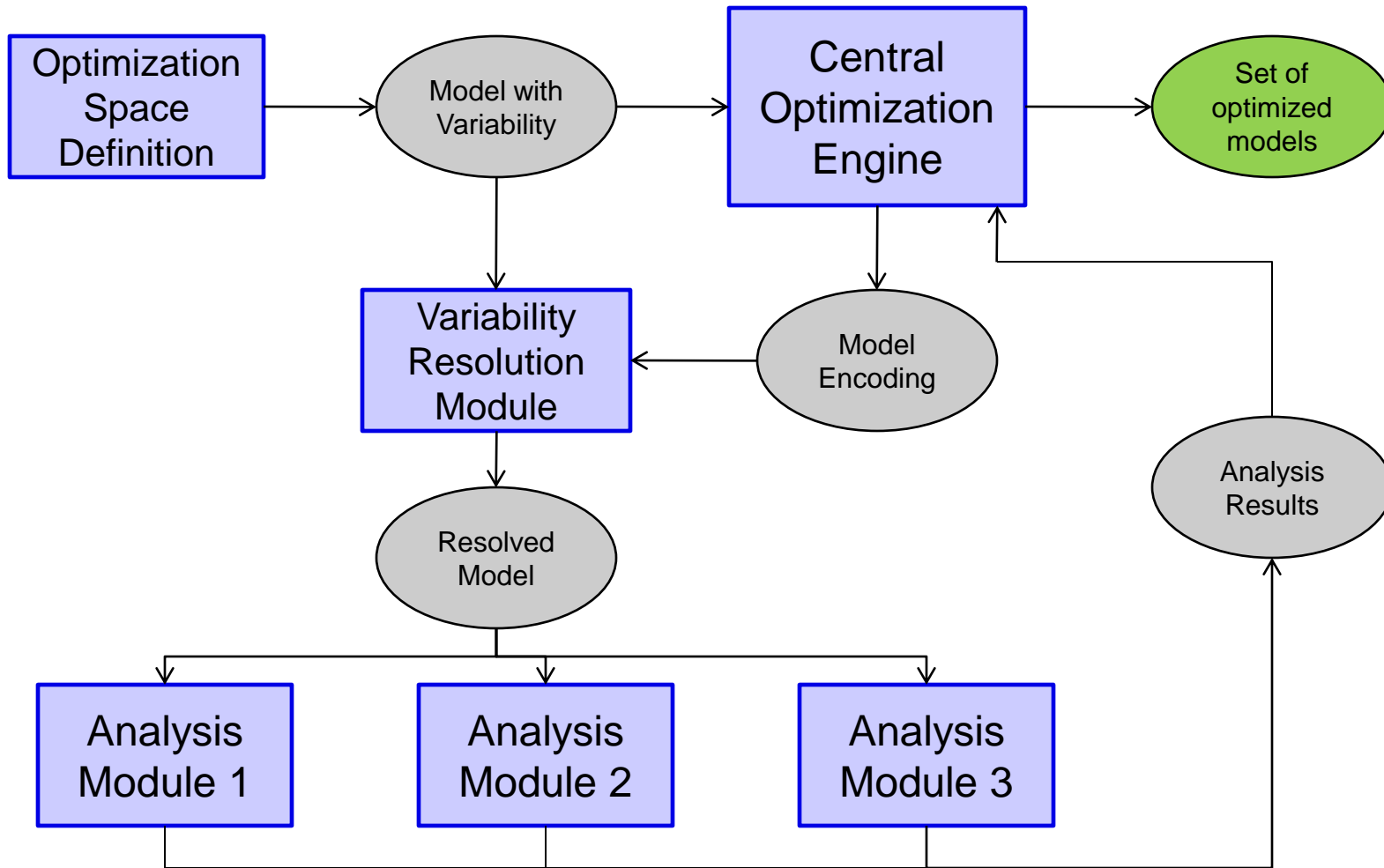
- Expressing the design space is dependent on abstraction level of the architecture
- Allocation decisions of functions on hardware can have a large effect e.g. on both timing and safety
 - Thus different allocation strategies can be important when optimizing at the EAST-ADL design level
 - At higher abstraction levels, this may not be relevant
- For safety, different redundancy strategies are also important
 - This can take place at higher abstraction levels too

Notion of substitution

- Three main forms of substitution
 - Substitution with a different implementation, e.g. components with same function but from different manufacturers
 - Replacement with a new architecture (e.g. voter, standby)
 - Simple replication of parallel components
- Substitute needs a compatible interface and same function

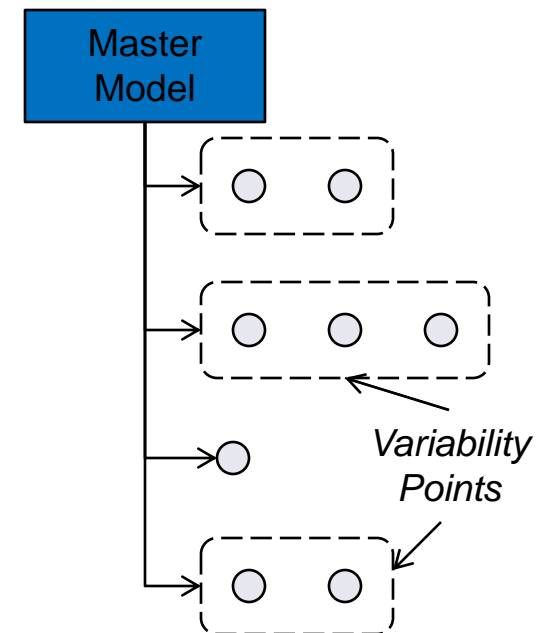


Optimisation Process Framework



Optimization Space Definition Module

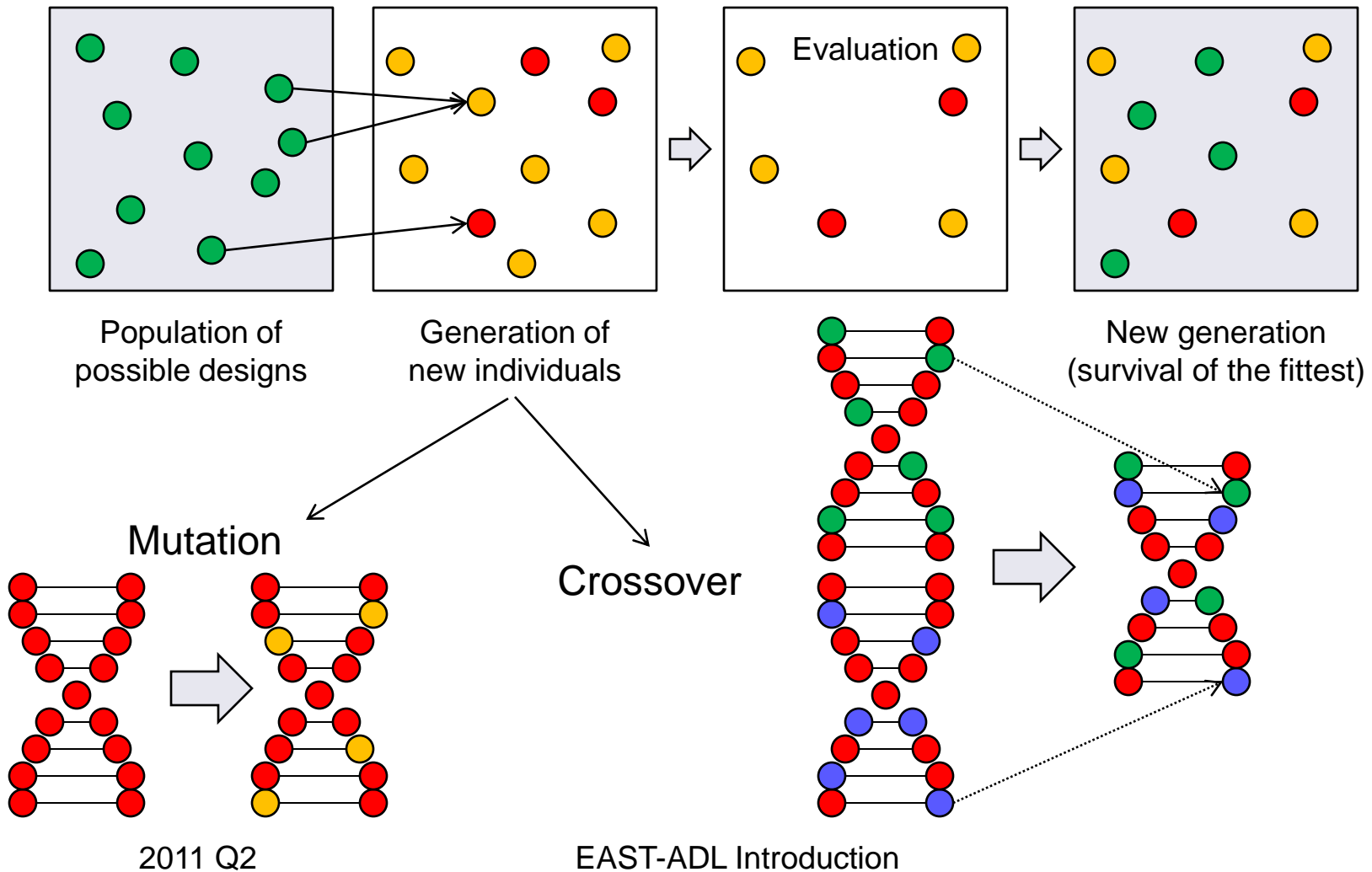
- OSDM examines initial model and takes all points of variability
- From this it creates a tree-structure that defines the optimization search space
- By selecting which tree nodes to include/exclude, the optimization can configure the model to test different strategies



Optimization Engine

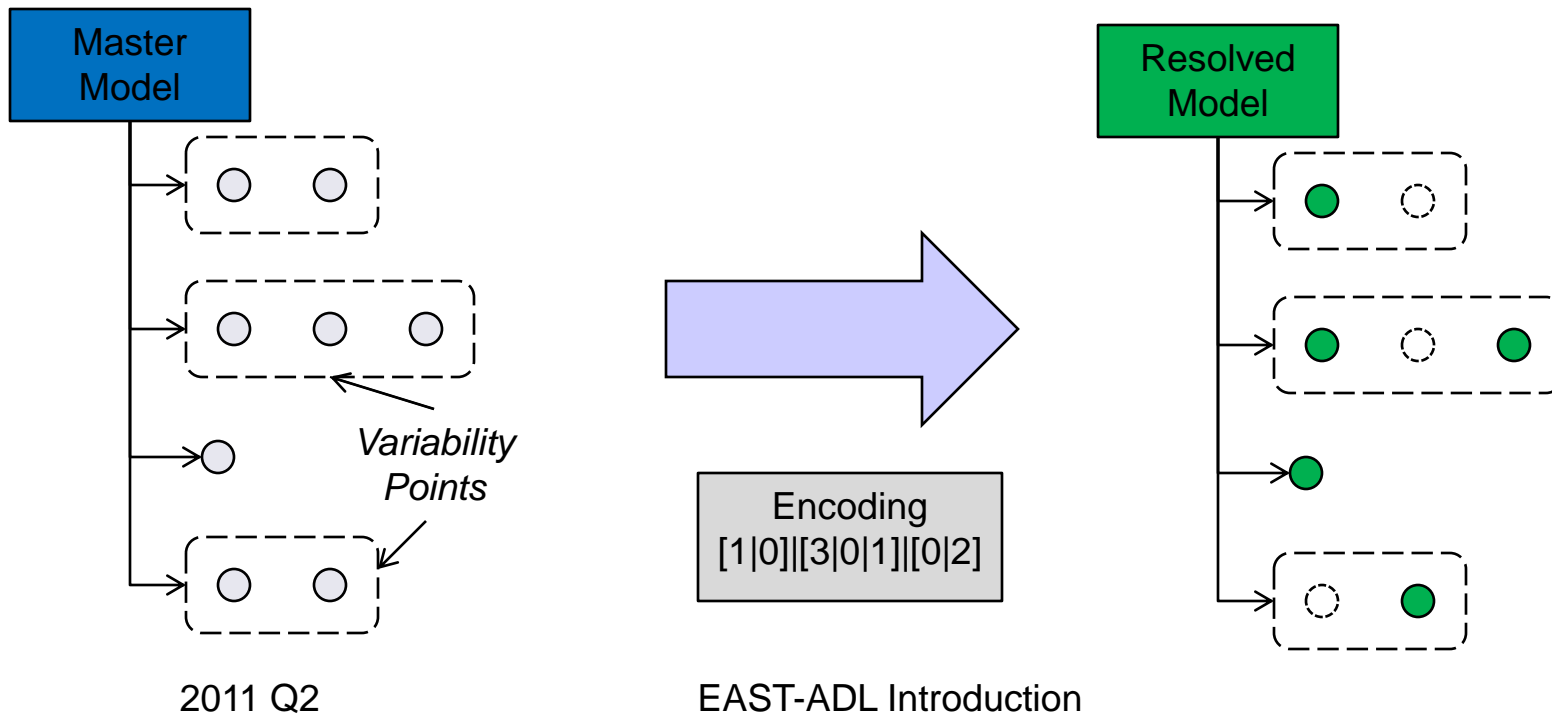
- Central Optimization Engine (COE) is based on HiP-HOPS technology
- Optimization implemented with genetic algorithms that emulate the process of natural evolution
- Main input is a structure that defines the design space
 - E.g. a tree-based structure containing all the variability points
- The genetic algorithms manipulate this structure to test out different ‘encodings’ - like DNA – for different models
- Analysis evaluates the model and on that basis the COE either discards the model or keeps it for the next iteration

Optimization Engine Process



Variability Resolution Mechanism

- The VRM takes an encoding from the optimization engine and uses it to resolve the variability in the original model
- The resulting ‘resolved’ model can then be analysed



Analysis

- Each fully resolved model is analysed according to each objective
 - Safety analysis to determine unavailability
 - Cost analysis to calculate total cost
 - Timing analysis to determine schedulability
 - Etc
- The results are then used by the COE to evaluate designs against each other
- Pareto solutions are kept, others can be discarded
 - Sometimes poor solutions are also kept to increase diversity
 - Helps to avoid getting stuck in an evolutionary dead-end

Optimisation engine results

