Introduction to Intelligent Vehicles
[4. System Design]

Chung-Wei Lin

cwlin@csie.ntu.edu.tw

CSIE Department

National Taiwan University

Fall 2019

We would like to thank Prof. Yao-Wen Chang (NTU) for contributing the lecture material

Announcement

- **Lecture plan**
  - No class on October 14
  - We will consider a make-up class in December (after the midterm)
- **Homework**
  - Homework 1
    - Graded
    - Reminder: definition of B_i, ceiling function, connection between Q1 and Q2
    - Regrade request due on October 9 (Wednesday) 11:59pm
  - Homework 2
    - Posted
    - Due on October 31 (Monday) noon
- **If you have any question, please feel free to**
  - Post on NTU COOL or send me an email

Revisit CAN Timing Analysis

- $$\underbrace{Q_i}_{(1)} = \underbrace{B_i}_{(2)} + \underbrace{\underbrace{\sum_{(\text{for all } j,\ P_j<P_i)}}_{(3)} \underbrace{\left\lceil \frac{Q_i + \tau}{T_j} \right\rceil}_{(4)} \underbrace{C_j}_{(5)}}_{(6)}$$
- $$R_i = Q_i + \underbrace{C_i}_{(7)}$$

(1): (2)+(6)

(2): blocking time of the longest lower **or same priority** message

(3): index set of all higher priority messages

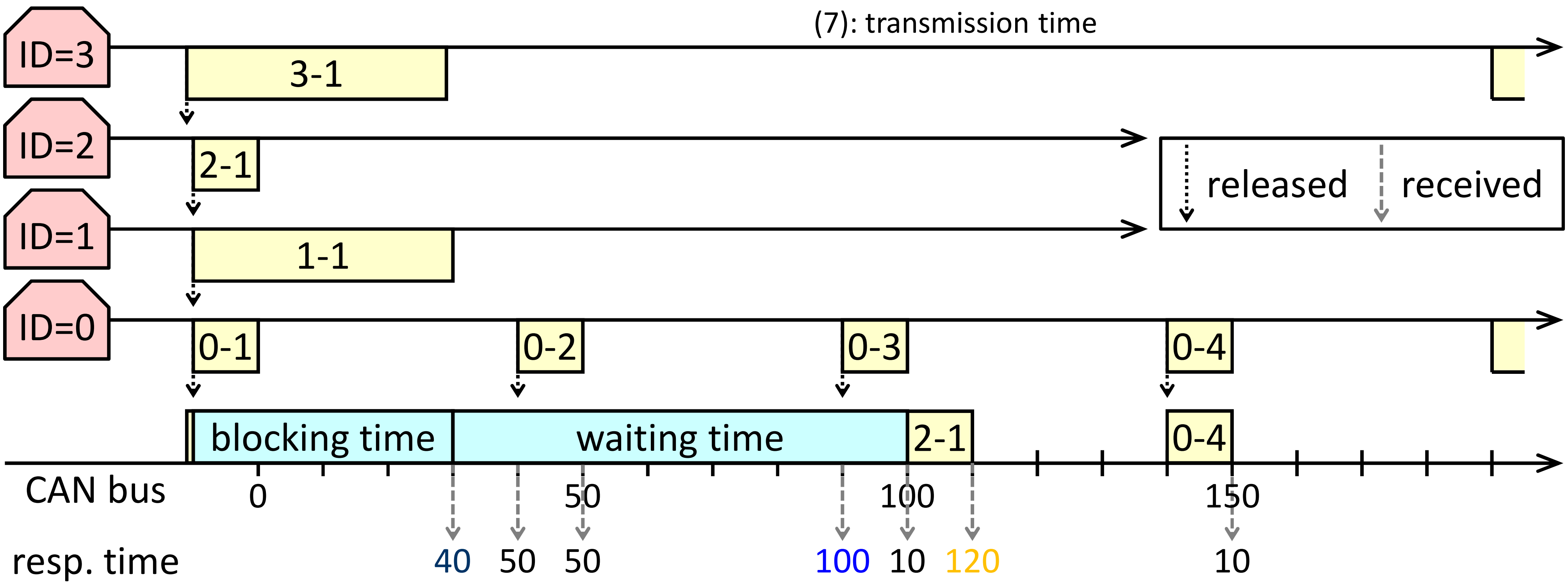
(4): max number of queued instances of message j within (1)

τ: transmission of one bit (Slide P31 for the reason)

(5): transmission time of an instance of message j

(6): waiting time

(7): transmission time



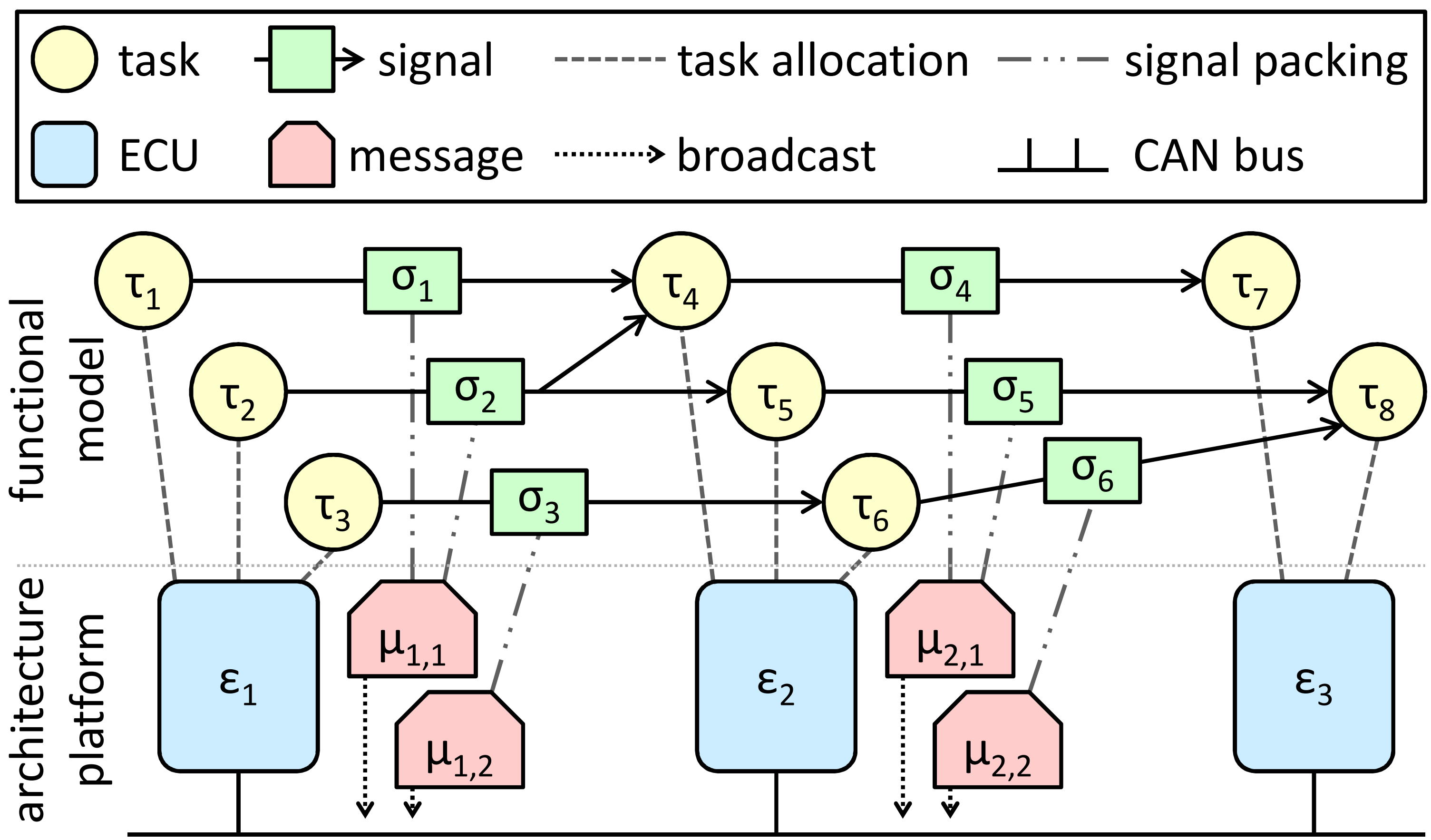
$R_0 = 50$ $R_1 = 100$ $R_2 = 120$ $R_3 = 40?$

System Design

- What is system design?
- Why is system design needed?
- When is system design done?
- Where is system design done?
- Who performs system design?
- How to perform system design?

Example (Piece) of System Design

- What is system design? / Why is system design needed?
- When / Where is system design done?
- Who / How to perform(s) system design?



Cores of Model-Based Design

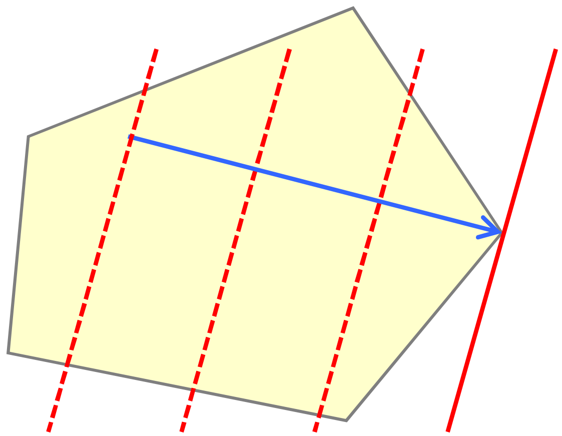
- **Modeling**
  - Work (design and analyze) upon models rather than real systems
- **Design**
  - Optimize some objectives (performance, robustness, security, etc.)
  - Satisfy some constraints
- **Analysis**
  - Check if there is any flaw or how good the designs are?
- **(Implementation)**
  - Make it real

Outline

- **Mixed Integer Linear Programming (MILP)**
- Simulated Annealing
- Mapping Problem
  - It is just a piece of system design

Linear Programming

- Linear Programming (LP)
  - Maximize
    - $2x_1 - x_2$
  - Subject to
    - $x_1 - x_2 \le 1$
    - $2x_2 \le 3$
    - $x_1, x_2 \ge 0$
    - x_1, x_2 are real numbers



- Canonical form

  min $\mathbf{c}^T\mathbf{x}$

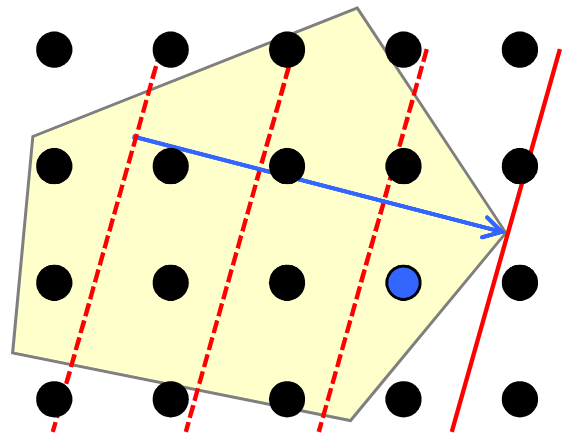
  subject to $\boldsymbol{A}\mathbf{x} \le \mathbf{b}$, $\mathbf{x} \ge \mathbf{0}$, x in R

- Simplex algorithm
  - https://en.wikipedia.org/wiki/Simplex_algorithm

Integer (Linear) Programming (ILP)

Integer (Linear) Programming (ILP)

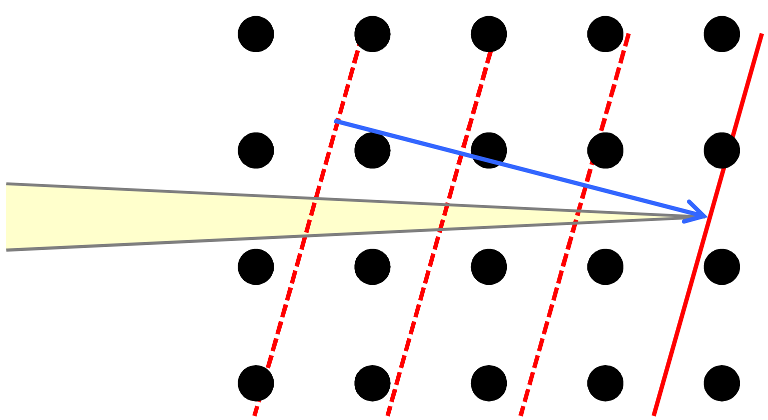
- Maximize
  - $2x_1 - x_2$
- Subject to
  - $x_1 - x_2 \leq 1$
  - $2x_2 \leq 3$
  - $x_1, x_2 \geq 0$
  - x_1, x_2 are **integers**



Canonical form

max $\quad \mathbf{c}^T\mathbf{x}$

subject to $\quad \boldsymbol{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$, x in Z

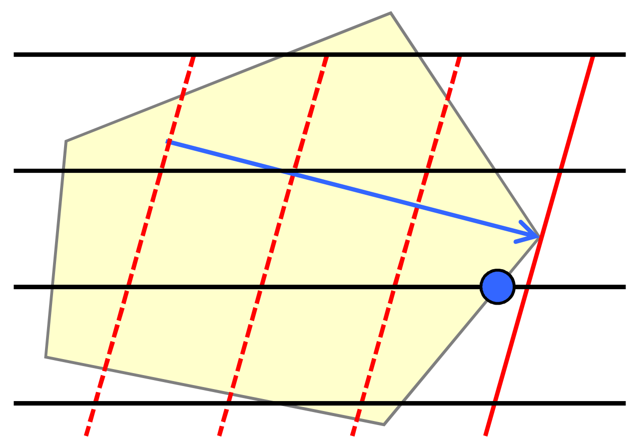


NP-complete

- Why is it hard?

Mixed Integer Linear Programming (MILP)

- Mixed Integer Linear Programming (MILP)
  - Maximize
    - $2x_1 - x_2$
  - Subject to
    - $x_1 - x_2 \leq 1$
    - $2x_2 \leq 3$
    - $x_1, x_2 \geq 0$
    - x_1 is **real number**
    - x_2 is **integer**



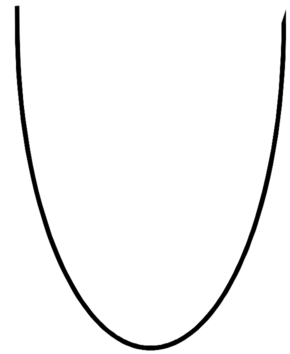
- Canonical form

max $\mathbf{c}^T\mathbf{x}$

subject to $\boldsymbol{A}\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, some x are integers

- Many tools are available for solving LP, ILP, and MILP

Quadratic Programming (QP)



❑ Canonical form

min $\quad 1/2\ \mathbf{x}^T \boldsymbol{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}$

subject to $\quad \boldsymbol{A}\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$

- Q is symmetric

➢ A special case of convex optimization

❑ Convex optimization

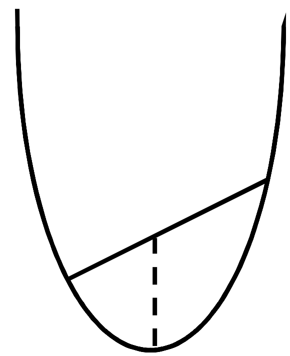
max $\quad f(\mathbf{x})$

subject to $\quad g_i(\mathbf{x}) \leq 0$ for all i

- f, g_i are convex

➢ A function f is convex if

- For all x_1 and x_2 in the domain of f, t in [0,1]
  $f(\ t \cdot x_1 + (1 - t) \cdot x_2\ ) \leq t \cdot f(x_1) + (1 - t) \cdot f(x_2)$



Practice

- Bin packing problem [Wikipedia]
  - Given
    - A set of bins S_1, S_2, … with the same size V
    - A set of n items with sizes $a_1, a_2, \ldots, a_n$
  - "Pack" all items into bins and minimize the number of used bins
- LP? ILP? MILP? QP?

max $\quad \mathbf{c}^T\mathbf{x}$

subject to $\quad \boldsymbol{A}\mathbf{x} \le \mathbf{b}$, $\mathbf{x} \ge \mathbf{0}$, x in Z

min $\quad y_1 + y_2 + \ldots + y_n$

subject to $\quad x_{i1} + x_{i2} + \ldots + x_{in} = 1 \qquad$ for all i

$a_1x_{1j} + a_2x_{2j} + \ldots + a_nx_{nj} \le Vy_j \qquad$ for all j

x_{ij} = 0 or 1 → 1: if and only if item i it packed in to bin j

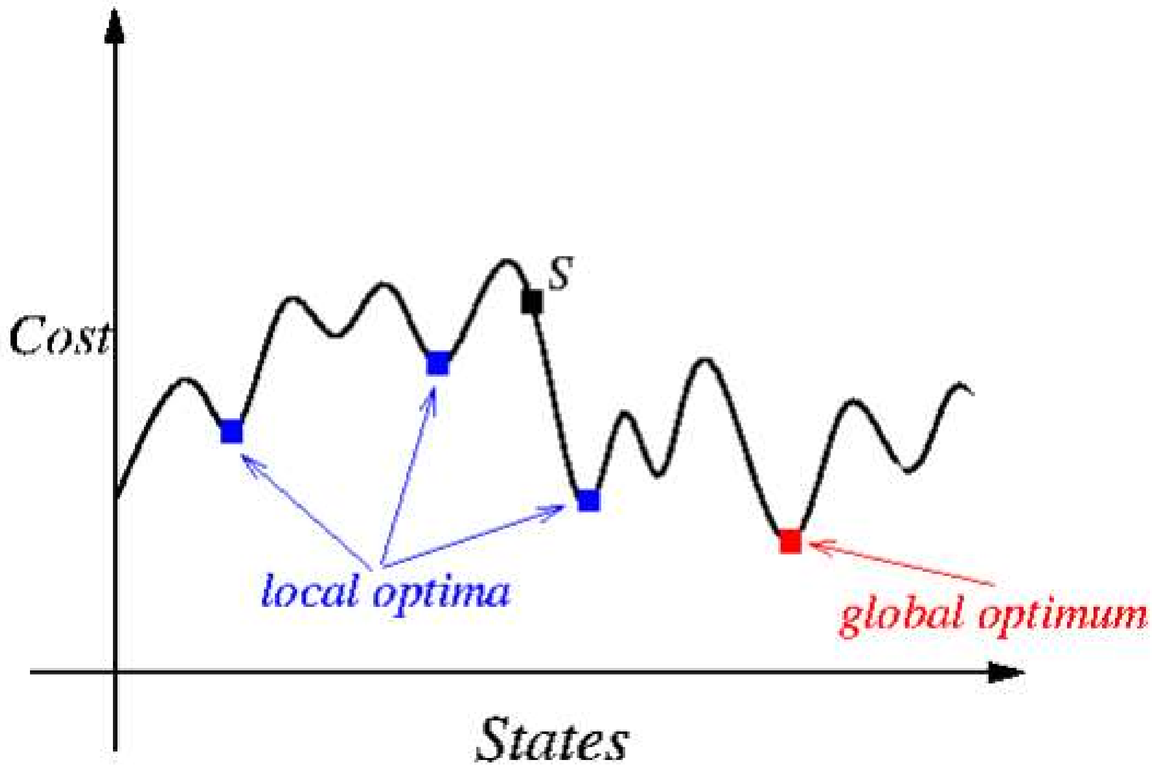
y_j = 0 or 1 → 1: if and only if bin j is used

Outline

- Mixed Integer Linear Programming (MILP)
- **Simulated Annealing**
- Mapping Problem
  - It is just a piece of system design

Simulated Annealing: Background

- Kirkpatrick, Gelatt, and Vecchi, "Optimization by simulated annealing," Science, May 1983.



Simulated Annealing: Basics

- Non-zero probability for "up-hill" moves
- Probability depends on
  - Magnitude of the "up-hill" movement
  - Total search time
- $Prob(S \rightarrow S') = \min(1, e^{-\Delta C/T})$
  - Equivalence
    - $Prob(S \rightarrow S') = 1$ if $\Delta C \leq 0$ ("down-hill" move)
    - $Prob(S \rightarrow S') = e^{-\Delta C/T}$ if $\Delta C > 0$ ("up-hill" move)
  - $\Delta C = cost(S') - cost(S)$
    - "Smaller" is better
  - T = temperature
    - T will cool down gradually

Simulated Annealing: Algorithm

- Get an initial solution S
- Get an initial temperature T > 0
- S* = S
- While "not yet frozen" (i.e., T is large enough)
  - (Iterate many times)
    - Pick a random neighbor S' of S
    - ΔC = cost(S') – cost(S)
    - If cost(S') < cost (S*), then S* = S'
    - If $\Delta C \leq 0$, then S = S'
    - If $\Delta C > 0$, then S = S' with probability $e^{-\Delta C/T}$
  - T = rT (where r < 1)
- Return S*

Simulated Annealing

- Basic ingredients
  - Solution space
    - What are the feasible solutions?
  - Neighborhood structure
    - How to find a neighboring solution from the current one?
  - Cost function
    - How to evaluate the quality of a solution?
  - Annealing schedule
    - How to conduct the search process to find a desired solution?
- Philosophy

Practice

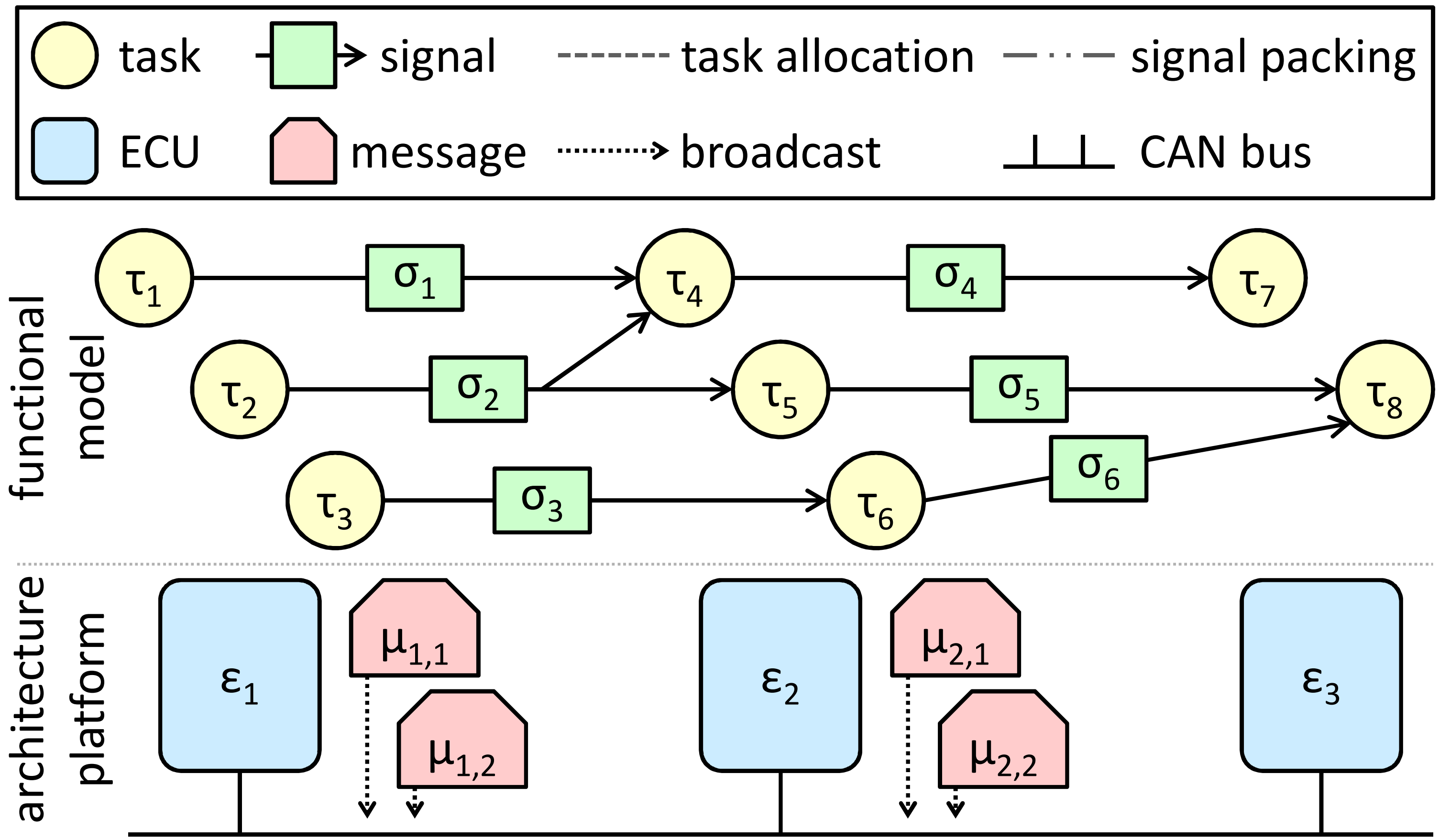
- Bin packing problem [Wikipedia]
  - Given
    - A set of bins S_1, S_2, … with the same size V
    - A set of n items with sizes a_1, a_2, …, a_n
  - "Pack" all items into bins and minimize the number of used bins
- Basic ingredients
  - Solution space
    - What are the feasible solutions?
  - Neighborhood structure
    - How to find a neighboring solution from the current one?
  - Cost function
    - How to evaluate the quality of a solution?
  - Annealing schedule
    - How to conduct the search process to find a desired solution?

Outline

- Mixed Integer Linear Programming (MILP)
- Simulated Annealing
- **Mapping Problem**
  - It is just a piece of system design

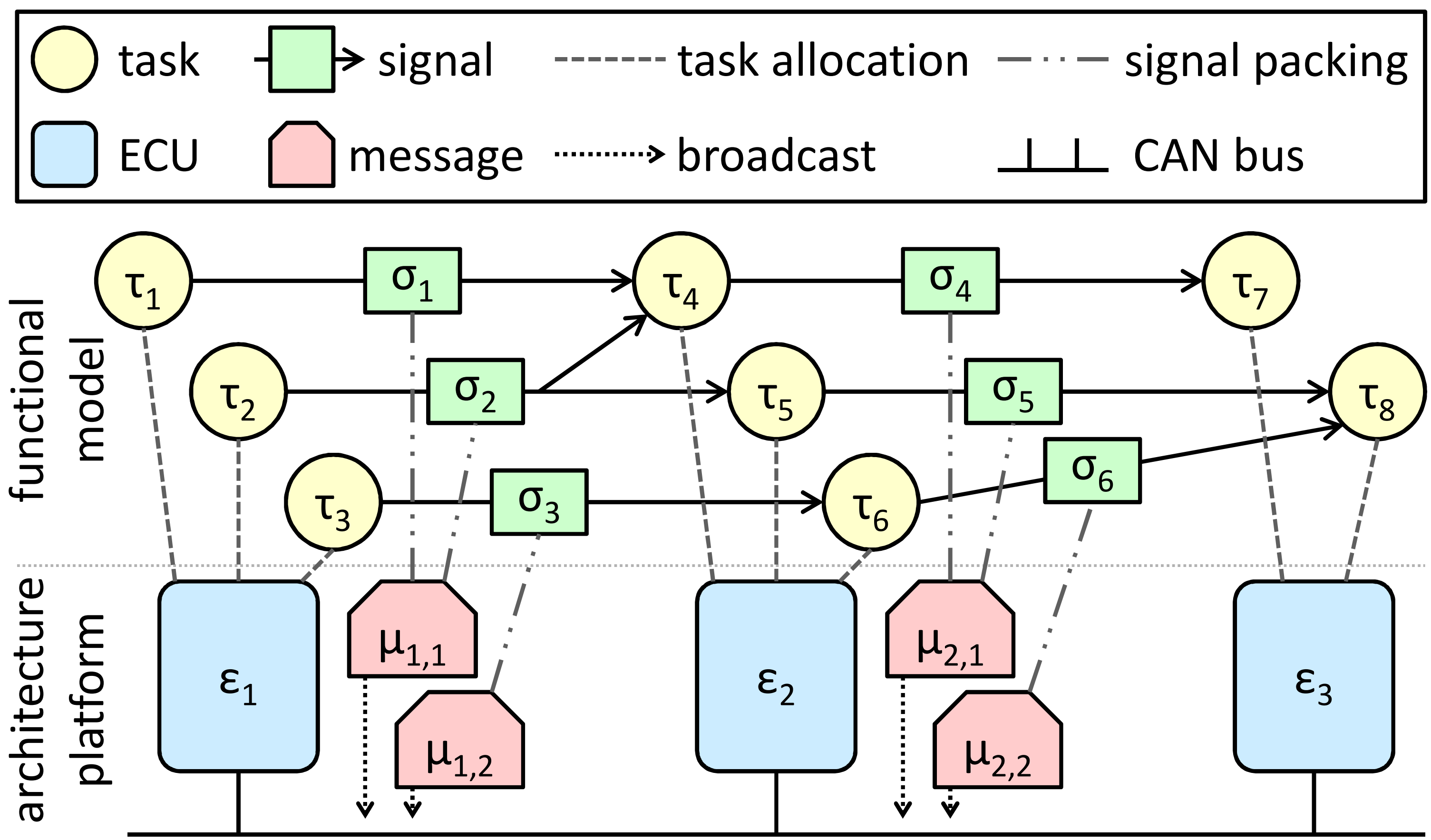
Mapping from Software to Hardware

- Software (functional model): task graph
- Hardware (architectural platform): distributed Electronic Control Units (ECUs) connected by a network



Problem Formulation

- Decide task allocation, signal packing, and priority assignments (tasks on ECUs and messages on CAN bus)
- Satisfy timing constraints for tasks, signals, and paths



Task Allocation

Indices and variables

- i, j: index of a task
- k: index of an ECU
- $a_{i,k}$: [0,1] task i is allocated onto ECU k
- $s_{i,j}$: [0,1] tasks i and j are allocated onto the same ECU

Constraints

- $\sum_k a_{i,k} = 1$ for all i
- $a_{i,k} + a_{j,k} + s_{i,j} \neq 2$ for all i, j, k

Questions

- What are the meanings of the constraints?
- Why do we need $s_{i,j}$?
- Is it possible that all tasks are allocated onto one ECU?

Task Priority Assignment

Indices and variables

- i, j, j': index of a task
- $p_{i,j}$: [0,1] task i has a higher priority than another task j

Constraints

- $p_{i,j} + p_{j,i} = 1$ for all $i, j, i \neq j$
- $p_{i,j} + p_{j,j'} - 1 \leq p_{i,j'}$ for all $i, j, j', i \neq j, i \neq j', j \neq j'$

Questions

- What are the meanings of the constraints?
- Why do we use binary variables $p_{i,j}$, not integer $p_i = 1, 2, 3, \ldots$?
  - $p_i \neq p_j$
  - $p_i < p_j$ or $p_j < p_i$
  - $p_{i,j} + p_{j,i} = 1$ and $(p_i - p_j) < (1 - p_{i,j})\, M$ and $(p_j - p_i) < p_{i,j}\, M$
    - M: a large constant

Signal Packing

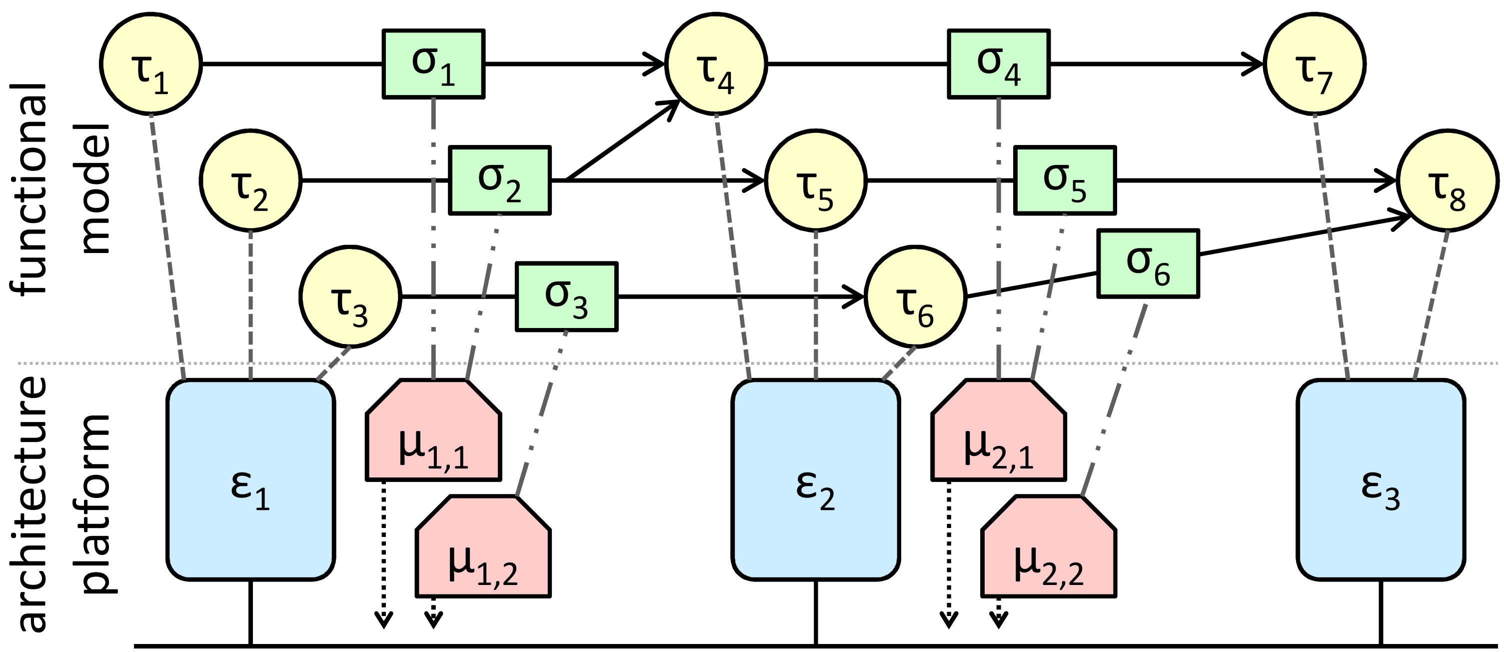
- Indices, constant parameters, and variables
  - i, j: index of a task;   k: index of an ECU;   l: index of a message
  - $T_{i,j}$: the period of the signal from task i to task j
  - $T_{k,l}$: the period of ECU k's message l
  - $a_{i,k}$: [0,1] task i is allocated onto ECU k
  - $t_{i,j,k,l}$: [0,1] the signal from task i to task j is packed into ECU k's message l
  - $v_{k,l}$: [0,1] ECU k's message l is used
- Constraints
  - $\sum_l t_{i,j,k,l} = a_{i,k}(1 - a_{j,k})$ for all i, j, k
  - $t_{i,j,k,l} \leq v_{k,l}$ for all i, j, k, l
  - $t_{i,j,k,l}\, T_{k,l} \leq T_{i,j}$; $t_{i,j,k,l}\, T_{i,j} \leq T_{k,l}$ for all i, j, k, l
- Questions
  - What are the meanings of the constraints?
  - Why do we need $v_{k,l}$?

Message Priority Assignment

- Similar to task priority assignment

Timing Constraints

- Task: response time ≤ period (given)
  - Response time? $r_i = c_i + \sum_{p_j<p_i} \lceil r_i / T_j \rceil C_j$
- Message and signal: response time ≤ period (given)
  - Response time? $q_i = b_i + \sum_{p_j<p_i} \left\lceil \frac{q_i + \tau}{T_j} \right\rceil c_j$ and $r_i = q_i + c_i$
- Path: latency ≤ deadline (given)
  - Latency?



Remaining Questions

- Variables
  - $s_{i,j}$: [0,1] tasks i and j are allocated onto the same ECU
  - $v_{k,l}$: [0,1] ECU k's message l is used
- Questions
  - Task allocation
    - Why do we need $s_{i,j}$?
      - $s_{i,j} \cdot p_{i,j}$ in response time computation
    - Is it possible that all tasks are allocated onto one ECU?
      - We have timing constraints
  - Signal packing
    - Why do we need $v_{k,l}$?
      - $v_{k,l} \cdot$ (message size) in response time computation

Solved by MILP: Linearization

- Inequality of three binary variables: $\alpha + \beta + \gamma \neq 2$
  - $\alpha + \beta + \gamma \neq 2 \;\leftrightarrow\; \alpha + \beta - \gamma \leq 1;\ \alpha - \beta + \gamma \leq 1;\ -\alpha + \beta + \gamma \leq 1$
- Ceiling function: ceil(f)
  - Replace ceil(f) by an integer x
  - $\mathrm{ceil}(f) = x \;\leftrightarrow\; 0 \leq x - f < 1$
- Multiplication of two binary variables: $\alpha \cdot \beta$
  - Replace $\alpha \cdot \beta$ by a binary variable γ
  - $\alpha \cdot \beta = \gamma \;\leftrightarrow\; \alpha + \beta - 1 \leq \gamma;\ \gamma \leq \alpha;\ \gamma \leq \beta$
- Multiplication of a binary variable α and a real variable x: $\alpha \cdot x$
  - Replace $\alpha \cdot x$ by a real variable y
  - $\alpha \cdot x = y \;\leftrightarrow\; 0 \leq y \leq x;\ x - M(1 - \alpha) \leq y \leq M\alpha$
    - M: a large constant

Solved by MILP: Scalability Issue

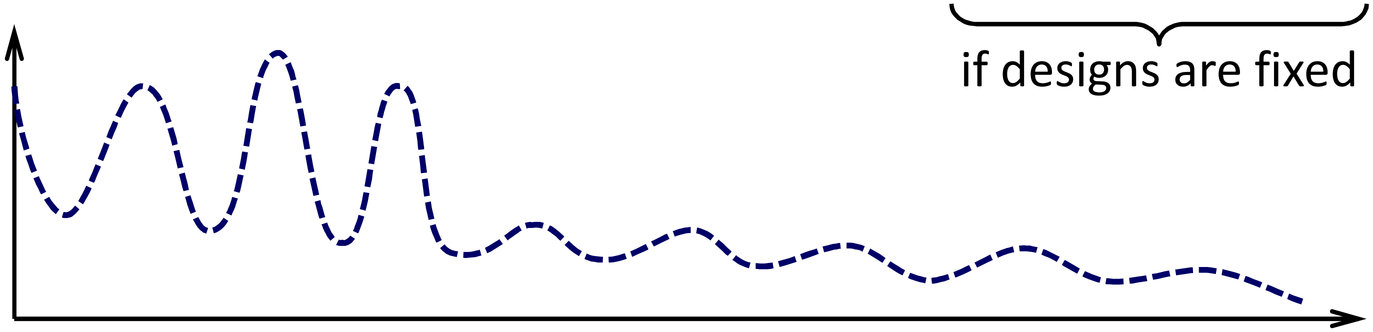
- Let us ignore "group assignment" first

design specifications and constrains → Step 1 → Step 2 → Step 3 → solution

	Step 1	Step 2	Step 3	
Given by Previous Steps		task allocation task priority	task allocation task priority signal packing message priority	solution
Defined by Assumptions	signal packing message priority group assignment	group assignment		
Decision Variables	task allocation task priority	signal packing message priority	group assignment	

$\underbrace{\text{Step 3 column}}_{\text{if designs are fixed}}$

objective = summation of the end-to-end latencies of selected paths



Solved by Simulated Annealing

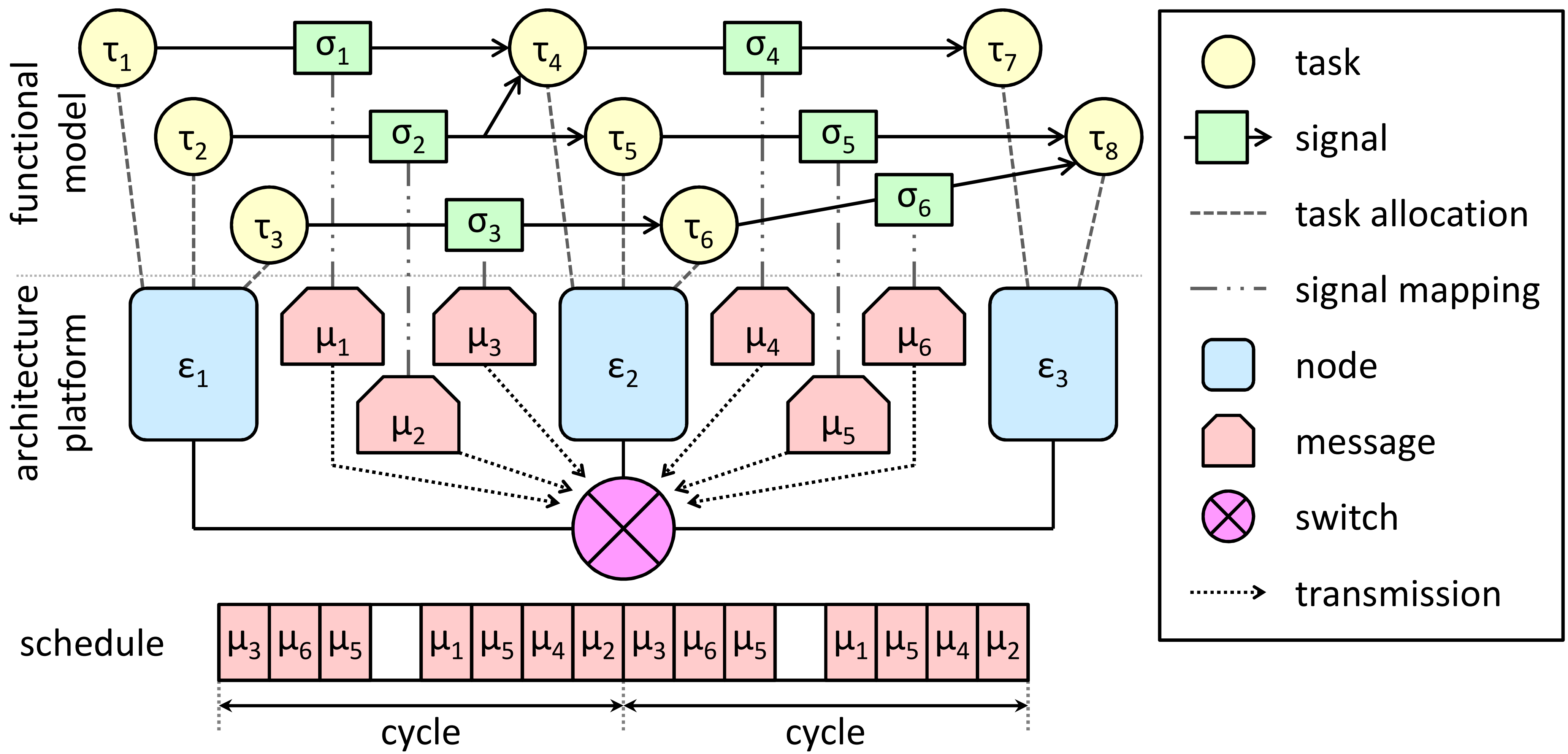
- Basic ingredients
  - Solution space
    - What are the feasible solutions?
  - Neighborhood structure
    - How to find a neighboring solution from the current one?
  - Cost function
    - How to evaluate the quality of a solution?
  - Annealing schedule
    - How to conduct the search process to find a desired solution?

Implications

- Allocate tasks onto the same ECU
  - A signal does not need to be transmitted if its source and target tasks are allocated onto the same ECU
  - Two signals can be packed into the same message if their source tasks are allocated onto the same ECU
- However, not allocate too many tasks onto the same ECU
  - Long task response time
- In most cases, pack signals into the same message
  - Save headers!

Another Problem Formulation

- Task allocation, task priority assignment, scheduling



Q&A