Efficient and Effective Multi-Objective Optimization for Real-Time Multi-Task Systems

Shashank Jadhav Heiko Falk

shashank.jadhav / heiko.falk@tuhh.de

Institute of Embedded Systems, TUHH

21st International Workshop on Worst-Case Execution Time Analysis (WCET), Vienna, Austria

July 11, 2023

Hamburg University of Technology

Overview

Introduction

- **Multi-task real-time systems**
  Important properties:
  -> Worst-Case Execution Time (WCET)
  -> Schedulability
  -> Energy Consumption

- **Multi-task real-time systems**
  Important properties:
  -> Worst-Case Execution Time (WCET)
  -> Schedulability
  -> Energy Consumption
- **Compiler-level Static SPM Allocation**
  -> Statically allocate basic blocks from Flash to SPM

Introduction

- **Multi-task real-time systems**
  Important properties:
  - -> Worst-Case Execution Time (WCET)
  - -> Schedulability
  - -> Energy Consumption
- **Compiler-level Static SPM Allocation**
  - -> Statically allocate basic blocks from Flash to SPM
- **Multi-objective optimization**
  - -> Minimize objectives and find set of Pareto-optimal Solutions

Introduction

- **Multi-task real-time systems**
  Important properties:
  - -> Worst-Case Execution Time (WCET)
  - -> Schedulability
  - -> Energy Consumption
- **Compiler-level Static SPM Allocation**
  - -> Statically allocate basic blocks from Flash to SPM
- **Multi-objective optimization**
  - -> Minimize objectives and find set of Pareto-optimal Solutions
- **Metaheuristic algorithms**
  - -> Flower Pollination Algorithm (FPA)
  - -> Strength Pareto Evolutionary Algorithm (SPEA)

Introduction

- **Multi-task real-time systems**
  Important properties:—> Time-consuming analyses
  - -> Worst-Case Execution Time (WCET)
  - -> Schedulability
  - -> Energy Consumption
- **Compiler-level Static SPM Allocation**
  - -> Statically allocate basic blocks from Flash to SPM
- **Multi-objective optimization**
  - -> Minimize objectives and find set of Pareto-optimal Solutions
- **Metaheuristic algorithms**
  - -> Flower Pollination Algorithm (FPA)
  - -> Strength Pareto Evolutionary Algorithm (SPEA)

Introduction

- **Multi-task real-time systems**
  Important properties:—> Time-consuming analyses
  -> Worst-Case Execution Time (WCET)
  -> Schedulability
  -> Energy Consumption
- **Compiler-level Static SPM Allocation**—> Search space dimension
  -> Statically allocate basic blocks from Flash to SPM
- **Multi-objective optimization**
  -> Minimize objectives and find set of Pareto-optimal Solutions
- **Metaheuristic algorithms**
  -> Flower Pollination Algorithm (FPA)
  -> Strength Pareto Evolutionary Algorithm (SPEA)

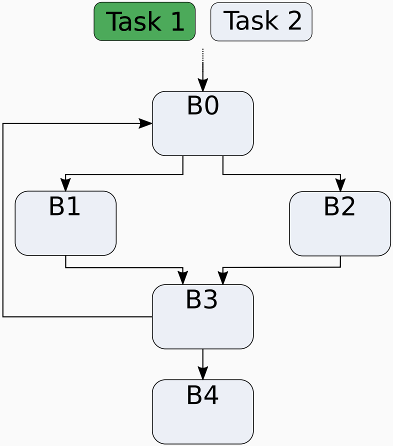
Introduction

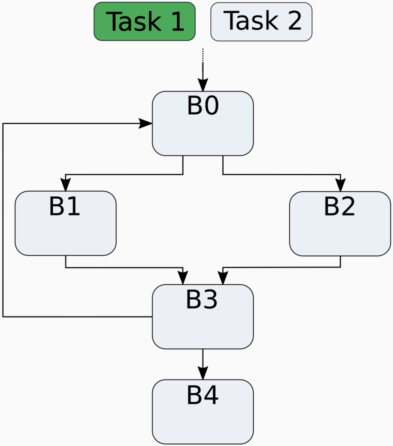
- **Multi-task real-time systems**
  Important properties:—> Time-consuming analyses
  - -> Worst-Case Execution Time (WCET)
  - -> Schedulability
  - -> Energy Consumption
- **Compiler-level Static SPM Allocation**—> Search space dimension
  - -> Statically allocate basic blocks from Flash to SPM
- **Multi-objective optimization**
  - -> Minimize objectives and find set of Pareto-optimal Solutions
- **Metaheuristic algorithms**—> Iterative algorithms
  - -> Flower Pollination Algorithm (FPA)
  - -> Strength Pareto Evolutionary Algorithm (SPEA)

- **Multi-task real-time systems**
  Important properties:—> Time-consuming analyses
  - -> Worst-Case Execution Time (WCET)
  - -> Schedulability
  - -> Energy Consumption
- **Compiler-level Static SPM Allocation**—> Search space dimension
  - -> Statically allocate basic blocks from Flash to SPM
- **Multi-objective optimization**
  - -> Minimize objectives and find set of Pareto-optimal Solutions
- **Metaheuristic algorithms**—> Iterative algorithms
  - -> Flower Pollination Algorithm (FPA)
  - -> Strength Pareto Evolutionary Algorithm (SPEA)

  —> Reduce search space dimension and number of iterations needed by Metaheuristic algorithms

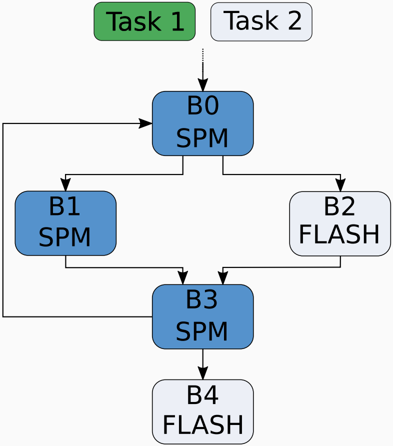
Overview


Task 1
Task 2
B0
B1
B2
B3
B4



- **Search Space**

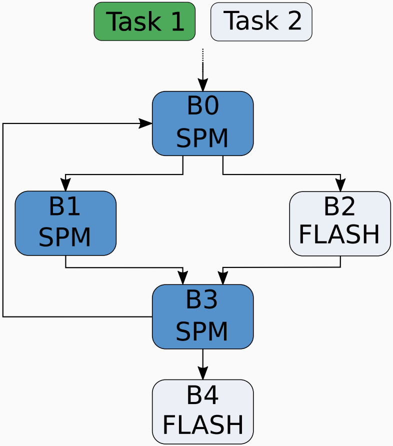
$$X = \left\{ \begin{array}{c} x = (x^1, .., x^T) \forall t = \overline{1, T} \\ x^{tv} \in \{0, 1\} \forall v = \overline{1, p^t} \\ x \in \{0, 1\}^d, d = \sum_{t=1}^{T} \sum_{v=1}^{p^t} v \end{array} \right\}$$



- **Search Space**

$$X = \left\{ \begin{array}{c} x = (x^1, .., x^T)\forall t = \overline{1, T} \\ x^{tv} \in \{0, 1\}\forall v = \overline{1, p^t} \\ x \in \{0, 1\}^d, d = \sum_{t=1}^{T} \sum_{v=1}^{p^t} v \end{array} \right\}$$

Multi-Objective Memory Allocation



- **Search Space**

$$X = \left\{ \begin{array}{c} x = (x^1, .., x^T) \forall t = \overline{1, T} \\ x^{tv} \in \{0, 1\} \forall v = \overline{1, p^t} \\ x \in \{0, 1\}^d, d = \sum_{t=1}^{T} \sum_{v=1}^{p^t} v \end{array} \right\}$$

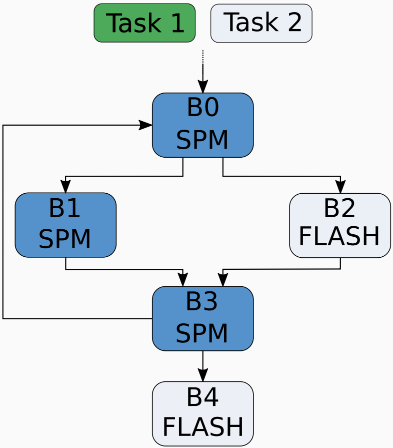
- **Objective Space**

$$\Theta = \{ \ F(x) = (F_1(x), F(x)_2, F_3(x)) \mid x \in X \ \}$$

where, $F_1(x) =$ **WCET**,

$F_2(x) =$ **Energy Consumption**, and

$F_3(x) =$ **Schedulability**



- **Search Space**

$$X = \left\{ \begin{array}{c} x = (x^1, .., x^T) \forall t = \overline{1, T} \\ x^{tv} \in \{0,1\} \forall v = \overline{1, p^t} \\ x \in \{0,1\}^d, d = \sum_{t=1}^{T} \sum_{v=1}^{p^t} v \end{array} \right\}$$

- **Objective Space**

$$\Theta = \{ \ F(x) = (F_1(x), F(x)_2, F_3(x)) \mid x \in X \ \}$$

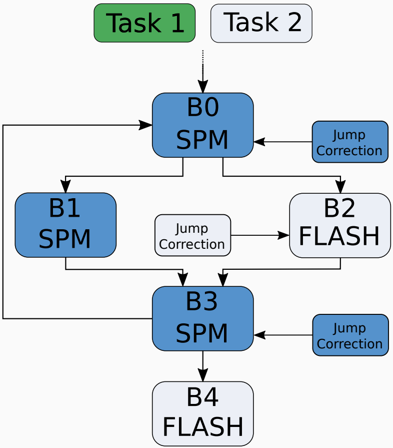
where, $F_1(x) =$ **WCET**,

$F_2(x) =$ **Energy Consumption**, and

$F_3(x) =$ **Schedulability**

- **Multi-Objective Problem**

$$\min_{x \in X} F(x)$$



- **Search Space**

$$X = \left\{ \begin{array}{c} x = (x^1, .., x^T) \forall t = \overline{1, T} \\ x^{t_v} \in \{0, 1\} \forall v = \overline{1, p^t} \\ x \in \{0, 1\}^d, d = \sum_{t=1}^{T} \sum_{v=1}^{p^t} v \end{array} \right\}$$

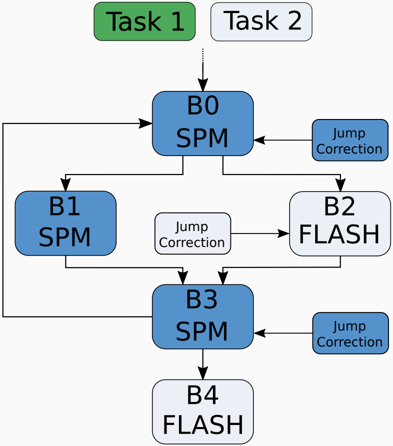
- **Objective Space**

$$\Theta = \{ \quad F(x) = (F_1(x), F(x)_2, F_3(x)) \mid x \in X \quad \}$$

where, $F_1(x) =$ **WCET**,
$F_2(x) =$ **Energy Consumption**, and
$F_3(x) =$ **Schedulability**

- **Multi-Objective Problem**

$$\min_{x \in X} F(x)$$



- **Search Space**

$$X = \left\{ \begin{array}{c} x = (x^1, .., x^T) \forall t = \overline{1, T} \\ x^{tv} \in \{0, 1\} \forall v = \overline{1, p^t} \\ x \in \{0, 1\}^d, d = \sum_{t=1}^{T} \sum_{v=1}^{p^t} v \end{array} \right\}$$

- **Objective Space**

$$\Theta = \{ \quad F(x) = (F_1(x), F(x)_2, F_3(x)) \mid x \in X \quad \}$$

where, $F_1(x) =$ **WCET**,
$F_2(x) =$ **Energy Consumption**, and
$F_3(x) =$ **Schedulability**

- **Multi-Objective Problem**

$$\min_{x \in X} F(x)$$

- **Constraint**

$$g(x) = \sum_{t=1}^{T} \sum_{v=1}^{p^t} B^{tv} x^{tv} - S_{SPM}$$

$$- \sum_{t=1}^{T} \sum_{v=1}^{p^t - 1} s^{tv} |x^{tv} - x^{tv+1}| \leq 0$$

Overview

Metaheuristic algorithm

Multi-objective optimization solved using: FPA and SPEA

Metaheuristic algorithm

Multi-objective optimization solved using: FPA and SPEA

Algorithm SPM allocation-based multi-objective optimization

Initialization: Initialize the initial population, perform jump corrections, and evaluate them.
Input: Initialized initial population, stopping criterion;
Output: approximated Pareto front.
while stopping criteria is not fulfilled **do** ▷ Iterate over all generations
    for $j = 1 : N$ **do** ▷ Iterate over all individuals
        Update the individual using update operators
        Evaluate the individual
    Using the selection operator, update to next generation

Metaheuristic algorithm

Multi-objective optimization solved using: FPA and SPEA

Algorithm SPM allocation-based multi-objective optimization

Initialization: Initialize the initial population, perform jump corrections, and evaluate them.
Input: Initialized initial population, stopping criterion;
Output: approximated Pareto front.
while stopping criteria is not fulfilled **do** ▷ Iterate over all generations
  for $j = 1 : N$ **do** ▷ Iterate over all individuals
    Update the individual using update operators
    Evaluate the individual
  Using the selection operator, update to next generation

Stopping criteria:

- Maximum number of generations
- Maximum number of generations for which the population remains the same

Multi-objective optimization solved using: FPA and SPEA

Algorithm SPM allocation-based multi-objective optimization

1: **Initialization:** Initialize the initial population, perform jump corrections, and evaluate them.
2: **Input:** Initialized initial population, stopping criterion;
3: **Output:** approximated Pareto front.
4: **while** stopping criteria is not fulfilled **do** ▷ Iterate over all generations
5:     **for** $j = 1 : N$ **do** ▷ Iterate over all individuals
6:         Update the individual using update operators
7:         Evaluate the individual
8:     Using the selection operator, update to next generation

Stopping criteria:

- Maximum number of generations
- Maximum number of generations for which the population remains the same
  —> It is affected by the dimension of search space

Overview

Path-based Constraint Approach (PCA)

Execution properties used by PCA:

- Worst-Case Execution Path (WCEP)
- Average-Case Execution Path (ACEP)

Path-based Constraint Approach (PCA)

Execution properties used by PCA:

- Worst-Case Execution Path (WCEP)
- Average-Case Execution Path (ACEP)

To constraint the individual:

- Collect basic blocks on WCEP
- Collect basic blocks on ACEP
- Basic blocks not on either WCEP and ACEP are constrained

Path-based Constraint Approach (PCA)

Algorithm SPM allocation-based multi-objective optimization with PCA

Initialization: Initialize the initial population, perform jump corrections, and evaluate them.
Input: Initialized initial population, stopping criterion;
Output: approximated Pareto front.
Call PCA Algorithm
Collect constraints for initial population
while stopping criteria is not fulfilled **do** ▷ Iterate over all generations
    for $j = 1 : N$ **do** ▷ Iterate over all individuals
        Update the individual using update operators using PCA constraints
        Evaluate the individual
        Call PCA Algorithm
        Collect constraints for next generation
    Using the selection operator, update to next generation

Overview

Impact-based Constraint Approach (ICA)

- **Execution properties used by ICA:**
  - Worst-Case Execution Count (WCEC)
  - Average-Case Execution Count (ACEC)

Impact-based Constraint Approach (ICA)

- **Execution properties used by ICA:**
  - Worst-Case Execution Count (WCEC)
  - Average-Case Execution Count (ACEC)
- **For each basic block calculate:**
  - Impact metric: Impact metric calculates impact of each basic block to overall WCET and energy objective

Impact-based Constraint Approach (ICA)

- **Execution properties used by ICA:**
  - Worst-Case Execution Count (WCEC)
  - Average-Case Execution Count (ACEC)
- **For each basic block calculate:**
  - Impact metric: Impact metric calculates impact of each basic block to overall WCET and energy objective
- **Size Constraint:** $\sum_{b=1}^{\eta} B_b <= \alpha * S_{SPM}$

Impact-based Constraint Approach (ICA)

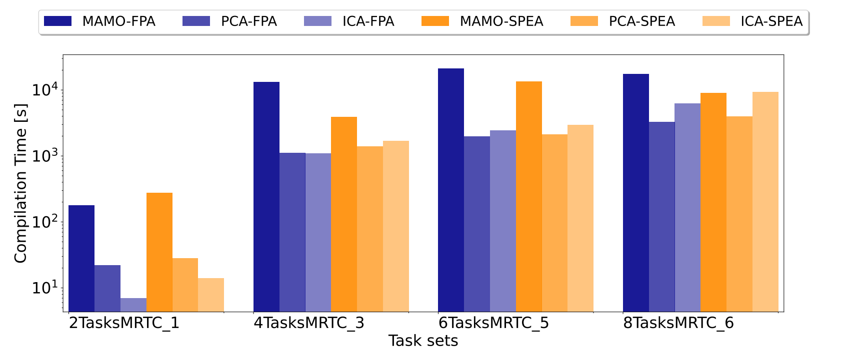
- **Execution properties used by ICA:**
  - Worst-Case Execution Count (WCEC)
  - Average-Case Execution Count (ACEC)
- **For each basic block calculate:**
  - Impact metric: Impact metric calculates impact of each basic block to overall WCET and energy objective
- **Size Constraint:** $\sum_{b=1}^{\eta} B_b <= \alpha * S_{SPM}$
- **To constraint the individual:**
  - Basic blocks with smaller impact metric that do not fulfill the size constraint are constrained

Impact-based Constraint Approach (ICA)

Algorithm SPM allocation-based multi-objective optimization with ICA

Initialization: Initialize the initial population, perform jump corrections, and evaluate them.
Input: Initialized initial population, stopping criterion;
Output: approximated Pareto front.
Initialize and Call ICA Algorithm
Collect constraints for initial population
while stopping criteria is not fulfilled **do** ▷ Iterate over all generations
    for $j = 1 : N$ **do** ▷ Iterate over all individuals
        Update the individual using update operators using ICA constraints
        Evaluate the individual
        Call ICA Algorithm
        Collect constraints for next generation
    Using the selection operator, update to next generation

Compilation time



- PCA required 85% and ICA required 77% less compilation time than MAMO
- PCA solved with FPA algorithm achieved most reduction in compilation time

Overview

Search Space Reduction

Calculate Search Space Reduction:

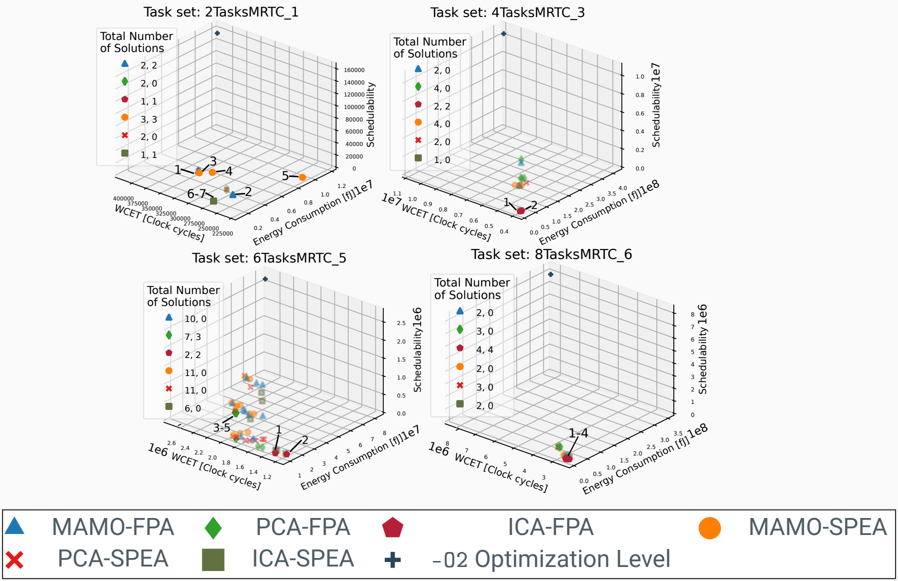
- For PCA: d – (total number of BBs not on WCEP and ACEP)
- For ICA: d – (total number of BB constrained by the Impact metric)

Total reduction in search space:

- PCA achieved, on average, 60% reduction in search space
- ICA achieved, on average, 87% reduction in search space

Overview

Pareto Fronts



Overview

Quality Metrics

- Coverage: $\mathcal{C} = 1 - \frac{|\{a \in A : \exists p \in \mathcal{P}, a \preceq p\}|}{|A|}$
- Non-Dominance Ratio: $NDR = \frac{|\mathcal{P} \cap A|}{|\mathcal{P}|}$
- Non-Dominated Solutions: $NDS = \frac{|a \in A : a \in \mathcal{P}|}{|A|}$

- Coverage: $\mathcal{C} = 1 - \frac{|\{a \in A : \exists p \in \mathcal{P}, a \preceq p\}|}{|A|}$
- Non-Dominance Ratio: $NDR = \frac{|\mathcal{P} \cap A|}{|\mathcal{P}|}$
- Non-Dominated Solutions: $NDS = \frac{|a \in A : a \in \mathcal{P}|}{|A|}$

From Overall Evaluations:

–> Comparison between FPA and SPEA

- SPEA and FPA algorithm provided relatively same quality of solutions for MAMO and PCA
- FPA performed better for ICA

–> Comparison between MAMO, PCA, and ICA

- ICA using FPA provided best quality solutions compared to other approaches

Overview

Conclusion

- Formulated and solved 3-dimensional SPM allocation-based multi-objective optimization problem for multi-task systems
- Proposed ICA and PCA for search space reduction
- Achieved drastic reduction in search space and compilation time
- Achieved best quality solutions with ICA approach using FPA algorithm

Future Work

- Better strategies to initialize Metaheuristic algorithms
- Pessimistic WCET and energy estimations

Thank You