Towards Pareto-Optimal Parameter Synthesis for Monotonic Cost Functions

FMCAD 2014, Lausanne

B. Bittner, M. Bozzano, A. Cimatti, M. Gario, A. Griggio

October 23, 2014
Motivations

- Parameters: variables with constant value, only partially constrained.
- *Parameterized systems* are pervasive
- Choice of appropriate parameters valuation: widely spread engineering problem, a form of design space exploration where the parameters can represent different design or deployment decisions.
- Examples:
  - function allocation [MVS07, HMP11]
  - automated configuration of communication media: time-triggered ethernet protocols [SD11], flexray [SEPC11, SGZ+11]
  - product lines [CHSL11]
  - dynamic memory allocation [MAP+06]
  - schedulability analysis [CPR08]
  - sensor placement [Gra09, BBCO12]
Finding one valuation is rarely sufficient. Finding *the most appropriate* valuation with respect to some cost: weight, latency, memory footprint, flexibility, reliability. Our work: several of the above dimensions must be taken into account at the same time. Trade off multiple cost functions: Pareto optimality. Constructing the so-called Pareto front [Par94] the set of parameter valuations that cannot be improved along one dimension without increasing the cost along the others.
Multiple cost functions: Pareto optimality

One valuation $\gamma$ strictly dominates a valuation $\gamma'$, written $\gamma \prec \gamma'$, if each value of $\gamma$ is not strictly greater than the corresponding value of $\gamma'$, and at least one value is strictly less.

$\gamma_i \leq \gamma'_i$ for each $i$, and $\gamma_i < \gamma'_i$ for some $i$.

The Pareto front is the set of points from $\Gamma$ that are not strictly dominated by any other point in $\Gamma$.

The Pareto front $PF(\text{Cost}, \varphi) \subseteq \Gamma$ is the set of parameter assignments that are valid for $\varphi$ and that are Pareto-optimal with respect to $\text{Cost}$.
Overview

Problem Definition

Problem Solution

Experiments

Conclusions and Future Work
Problem Definition

Parameterized transition system: \( S = (U, X, I, T) \)
- \(U\) is the set of parameters
- \(X\) is the set of state variables
- \(I(U, X)\) is the initial condition
- \(T(U, X, X')\) is the transition relation

Boolean parameters, valuations in \(\Gamma = \mathbb{B}^{|U|}\).

The order relation \(<\) over \(\mathbb{B}\) induces a partial order \(\leq\) over the parameter valuations \(\Gamma\).

A valuation \(\gamma \in \Gamma\) yields a non-parameterized transition system \(S_\gamma = (X, I(\gamma, X), T(\gamma, X, X'))\)
Symbolic representation

The “usual” symbolic representation

- $X$, $U$, $I(X, U)$, $T(U, X, X')$, boolean connectives, existential quantification, ...

- $\text{Reachable}_S(U, X)$ is the set of reachable states in $S$ under a given valuation

- from $\text{Reachable}_S(U, X) \land \gamma$ to $\text{Reachable}_{S\gamma}(X)$

the reachable state space of a parameterized system $S$ can be seen as an association between a parameter valuation $\gamma$ and the set of reachable states in the corresponding (non-parameterized) transition system $S\gamma$. 
Finite- vs Infinite-state

The techniques apply to finite- and infinite-state systems.

In the case of finite-state systems, termination is guaranteed.

In the infinite case, convergence depends on the termination of the calls to the underlying model checking engine.
Parameter synthesis and optimization

Relevant dimensions:

- combinational (e.g., SMT) problems versus sequential (e.g., reachability) problems
- discrete parameters versus real-valued parameters
- number and quality of parameter valuations found
  - one valuation vs all valuations
  - one vs optimal vs Pareto-optimal
- universal vs existential with respect to the traces of the transition system being analyzed
  - existential: \( \{ \gamma \mid S_\gamma \not\models \phi \}, \text{i.e. there exists } \sigma \in \mathcal{L}(S_\gamma), \sigma \not\models \phi \}
  - universal: \( \{ \gamma \mid S_\gamma \models \phi \}, \text{i.e. for all } \sigma \in \mathcal{L}(S_\gamma), \sigma \models \phi \}

Our setting: sequential, discrete parameters, all Pareto-optimal valuations, universal
Related work


- **Combinational Pareto front [LGCM10, MAP⁺ 06]:** Dynamic memory allocation and generalization. Combinational problem (SAT/SMT).

- **Real-valued parameter synthesis:** Schedulability [CPR08], IC3-based generalization [CGMT13]. Real-time/hybrid systems [HH94, Wan05, GJK08, AFKS12, AK12]. Universal, all valuations, no cost functions considered.

- **Automatic Synthesis of Fault Trees [BCT07]:** minimal fault configurations Synthesis of all valuations for discrete parameter; monotonicity hypothesis. Existential parameters. No costs taken into account.

- **Synthesis of Observability Requirements [Gra09, BBCO12]:** Sensor configurations for diagnosability. Single cost function (no Pareto front); monotonicity.
Monotonicity Assumptions

- **monotonicity of the “property holds” relation**
  We say that $S \models \phi$ is monotonic w.r.t. $\Gamma$ iff

  \[ \forall \gamma, \text{ If } S_{\gamma} \not\models \phi \text{ then } \forall \gamma'. \gamma' \preceq \gamma \Rightarrow S_{\gamma'} \not\models \phi \]

  If the property holds under a given valuation, then it also holds for all the successors.

- **monotonicity of the cost function**
  We say that $\text{Cost}$ is monotonic w.r.t. $\Gamma$ iff

  \[ \forall \gamma, \gamma'. \text{ If } \gamma \preceq \gamma' \text{ then } \text{Cost}(\gamma) \preceq \text{Cost}(\gamma') \]
Property-Monotonicity and Cost-Monotonicity
Three approaches:

- **Valuations-first**: compute whole set of good valuations \( \text{VALIDPARS} \) up-front; then compute the Pareto front.

- **One-cost slicing**: we “slice” the space \( \text{VALIDPARS} \) by one dimension: compute one of the slices at the time; once a slice has been computed, we minimize w.r.t. to the other costs.

- **Cost-first**: we do not compute \( \text{VALIDPARS} \) directly, but navigate through the valuations lattice driven by the cost functions and test on-the-fly membership of points to \( \text{VALIDPARS} \).
Valuations-first Approach
Valuations-first Approach

\textbf{function} \textsc{ValuationsFirst}(S, \textsc{Cost}, \varphi)
\begin{align*}
\text{VP} & := \textsc{ValidPars}(S, \varphi) \\
\text{return} & \textsc{ParetoFront}(\textsc{Cost}, \text{VP})
\end{align*}
\textbf{end function}

\textbf{function} \textsc{ValidPars}(S, \varphi)
\begin{align*}
\text{Bad} & := \bot \\
S & := (U, X, I, T) \\
\text{while} \ S \not\models \varphi \text{ do} \\
\gamma' & := \text{project counter-example on } U \\
\text{Bad} & := \text{Bad} \lor \gamma' \\
I & := I \land \neg \text{Bad} \\
\text{end while} \\
\text{return} & \neg \text{Bad}
\end{align*}
\textbf{end function}

\textsc{ParetoFront}(U) = \textsc{VP}(U) \land \not\exists U'.((U' \prec_{\textsc{Cost}} U) \land \textsc{VP}(U'))
One-cost slicing Approach
One-cost slicing Approach

function \textsc{Slicing}(S, \text{Cost}, \varphi) 
    PF := \emptyset; \gamma = T; 
    c_1 := \text{Cost}_1(\gamma) 
    S' := \text{FixCost}(S, \text{Cost}_1 = c_1) 
    VP_{\text{Cost}_1} := \text{ValidPars}(S', \varphi) 
    \textbf{while} \ VP_{\text{Cost}_1} \neq \emptyset \ \textbf{do} 
        (\gamma, c_2) = \text{Minimize} (\text{Cost}_2, \ VP_{\text{Cost}_1}) 
        (\gamma, c_1) := \text{Reduce}_{\text{Cost}_1}(S, \gamma, \varphi, c_2) 
        PF.add(\gamma, c_1, c_2) 
        c_1 := c_1 - 1 
        S' := \text{FixCost}(S, \text{Cost}_1 = c_1) 
        VP_{\text{Cost}_1} := \text{ValidPars}(S', \varphi) 
    \textbf{end while} 
    \textbf{return} PF 
end function

function \textsc{FixCost}(S, \text{CostExpr}) 
    S = (U, X, I, T) 
    S' := (U, X, I \land \text{CostExpr}, T) \textbf{return} S' 
end function
Cost-first Approach
function CostsFirst(S, Cost, \( \varphi \))
    PF := \emptyset
    \( \gamma \) := \top;
    \( c_1 = \text{Cost}_1(\gamma); \overline{c_2} = \text{Cost}_2(\gamma) \)
    repeat
        \( c_2 = \overline{c_2} \)
        for \( \gamma_i \in \text{MaxSmallerCandidate}_{\text{Cost}_2}(c_1, c_2) \) do
            if \( S_{\gamma_i} \models \varphi \) then
                (\( \gamma, c_2 \)) := Reduce_{\text{Cost}_2}(S, \gamma, \varphi, c_1)
            end if
        end for
        (\( \gamma, c_1 \)) := Reduce_{\text{Cost}_1}(S, \gamma, \varphi, c_2)
        PF.add(\( \gamma, c_1, c_2 \))
        \( c_1 := c_1 - 1 \)
    until No solution exists for FixCost(S, Cost_1 = c_1)
    return PF
end function
function CostsFirstIC3(S, Cost, ϕ)

    PF := ∅
    γ := T;
    $c_1 = \text{Cost}_1(γ)$; $\overline{c_2} = \text{Cost}_2(γ)$

    repeat
        $c_2 := \overline{c_2}$
        for $γ_i \in \text{MaxSmallerCandidate}_{\text{Cost}_2}(c_1, c_2)$ do
            $(res, ψ) := \text{IC3}(S, γ_i \rightarrow ϕ)$ // $S_{γ_i} \models ϕ$ iff $S \models γ_i \rightarrow ϕ$
            if $res == \text{Safe}$ then
                // $ψ$ is an inductive invariant s.t. $ψ \models γ_i \rightarrow ϕ$
                $(γ_i, c_1, c_2) := \text{Reduce}_{\text{Cost}_2}(ψ, γ_i, ϕ)$
            end if
        end for
        $(γ_i, c_1, c_2) := \text{Reduce}_{\text{Cost}_1}(ψ, γ_i, ϕ)$
        PF.add($γ, c_1, c_2$)
        $c_1 := c_1 - 1$
    until No solution exists for FixCost($S, \text{Cost}_1 = c_1$)

    return PF

end function
Motivating domain

Sensor Placement:
- Are the sensors enough to guarantee diagnosability?
- More sensors imply better diagnosability.
- Sensors have costs, weights, ...
- Find corresponding Pareto front to explore trade-off

Benchmarks from sensor placement and product lines.
Experiments: solved instances

<table>
<thead>
<tr>
<th>Family</th>
<th>#Instances</th>
<th>valuations-first</th>
<th>one-cost slicing</th>
<th>costs-first</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>32</td>
<td>11</td>
<td>13</td>
<td>32</td>
</tr>
<tr>
<td>cassini</td>
<td>21</td>
<td>6</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>elevator</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>orbiter</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>roversmall</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>roverbig</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>x34</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>product lines</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>TOTAL</td>
<td>81</td>
<td>43</td>
<td>49</td>
<td>81</td>
</tr>
</tbody>
</table>
Experiments: performance

Accumulated-time plot showing the number of solved instances (x-axis) in a given total time (y-axis) for the various algorithms.
Experiments: scalability wrt parameters

Runtime for different number of parameters
Experiments: Impact of REDUCE in costs-first
Conclusions and Future Work

Conclusions:

▶ from $S \models \phi$ to $\{\gamma \mid S\gamma \models \phi\}$
▶ from one valuation/best valuation, to Pareto front construction
▶ various algorithms, tight integration within IC3
▶ experiments are encouraging: significant scalability improvements

Future work:

▶ scalability for multiple cost functions
▶ when does the monotonicity hypothesis hold?
▶ real-valued parameters?
Questions?
É. André, L. Fribourg, U. Kühne, and R. Soulat.
IMITATOR 2.5: A tool for analyzing robustness in scheduling problems.

É. André and U. Kühne.
Parametric analysis of hybrid systems using HyMITATOR.

B. Bittner, M. Bozzano, A. Cimatti, and X. Olive.
Symbolic Synthesis of Observability Requirements for Diagnosability.
In *AAAI*, 2012.

M. Bozzano, A. Cimatti, and F. Tapparo.
Symbolic fault tree analysis for reactive systems.
References II

A. Cimatti, A. Griggio, S. Mover, and S. Tonetta.
Parameter synthesis with ic3.

A. Classen, P. Heymans, P.-Y. Schobbens, and A. Legay.
Symbolic model checking of software product lines.

A. Cimatti, L. Palopoli, and Y. Ramadian.
Symbolic computation of schedulability regions using parametric timed automata.

A counterexample-guided approach to parameter synthesis for linear hybrid automata.
References III

A. Grastien.
Symbolic testing of diagnosability.
In Twentieth International Workshop on Principles of Diagnosis (DX-09), 2009.

Thomas A. Henzinger and Pei-Hsin Ho.
Hytech: The cornell hybrid technology tool.

C. Hang, P. Manolios, and V. Papavasileiou.
Synthesizing cyber-physical architectural models with real-time constraints.
In CAV, pages 441–456, 2011.

Approximating the pareto front of multi-criteria optimization problems.
References IV

Automated exploration of pareto-optimal configurations in parameterized dynamic memory allocation for embedded systems.

Automating component-based system assembly.

V. Pareto.
*Manuale di economia politica.*

Using maxbmc for pareto-optimal circuit initialization.
In *DATE*, pages 1–6, 2014.
References V

W. Steiner and B. Dutertre.
Layered diagnosis and clock-rate correction for the ttethernet clock synchronization protocol.
In *PRDC*, pages 244–253, 2011.

Design optimization and synthesis of flexray parameters for embedded control applications.

Constraint-driven synthesis and tool-support for flexray-based automotive control systems.
F. Wang.
Symbolic parametric safety analysis of linear hybrid systems with bdd-like data-structures.