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

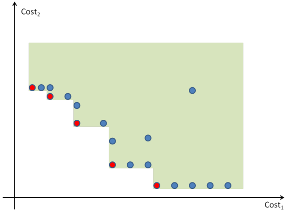
- Parameters: variables with constant value, only partially constrained.
- *Parameterized systems* are pervasive
- Choice of appropriate parameters valuation: widely spread engineering problem, a form of design space exploration where the parameters can represent different design or deployment decisions.
- 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]

Which parameter valuations?

- 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 γ strictly dominates a valuation γ', written $\gamma \prec \gamma'$, if each value of γ is not strictly greater than the corresponding value of γ', 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 Γ that are not strictly dominated by any other point in Γ.
The Pareto front $PF(\textsc{Cost}, \varphi) \subseteq \Gamma$ is the set of parameter assignments that are valid for φ and that are Pareto-optimal with respect to $\textsc{Cost}$.

Overview

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 $\preceq$ over the parameter valuations Γ.

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, ...
- $\textsc{Reachable}_S(U, X)$ is the set of reachable states in S under a given valuation
- from $\textsc{Reachable}_S(U, X) \wedge \gamma$ to $\textsc{Reachable}_{S_\gamma}(X)$
  the reachable state space of a parameterized system S can be seen as an association between a parameter valuation γ and the set of reachable states in the corresponding (non-parameterized) transition system S_γ.

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: $\{\gamma \mid S_\gamma \not\models \phi$, i.e. there exists $\sigma \in \mathcal{L}(S_\gamma), \sigma \not\models \phi\}$
    - universal: $\{\gamma \mid S_\gamma \models \phi$, 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

- MaxBMC [RSSB14]: circuit initialization.

  Pareto front: length of initialization sequence vs initialized flops.

  Existential: a trace gives a valid parameter valuation.
- 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

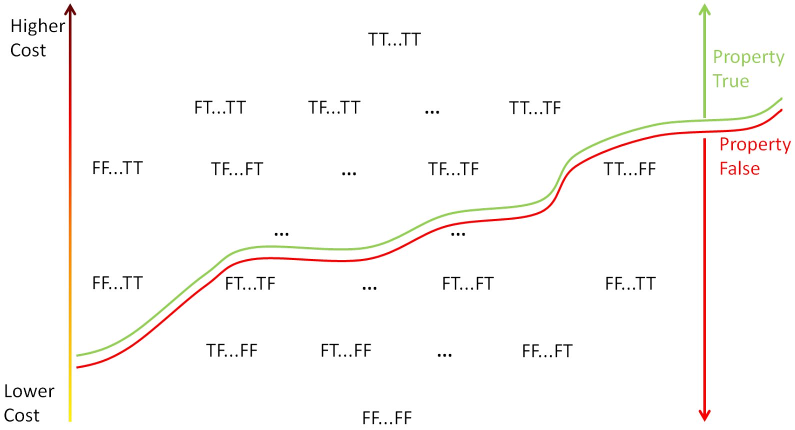
- monotonicity of the "property holds" relation
  We say that $S \models \varphi$ is monotonic w.r.t. Γ iff

  $$\forall \gamma,\ \text{If } S_\gamma \not\models \varphi \text{ then } \forall \gamma'.\ \gamma' \preceq \gamma \Rightarrow S_{\gamma'} \not\models \varphi$$

  If the property holds under a given valuation, then it also holds for all the successors.
- monotonicity of the cost function
  We say that $\textsc{Cost}$ is monotonic w.r.t. Γ iff

  $$\forall \gamma, \gamma'.\ \text{If } \gamma \preceq \gamma' \text{ then } \textsc{Cost}(\gamma) \preceq \textsc{Cost}(\gamma')$$

Property-Monotonicity and Cost-Monotonicity

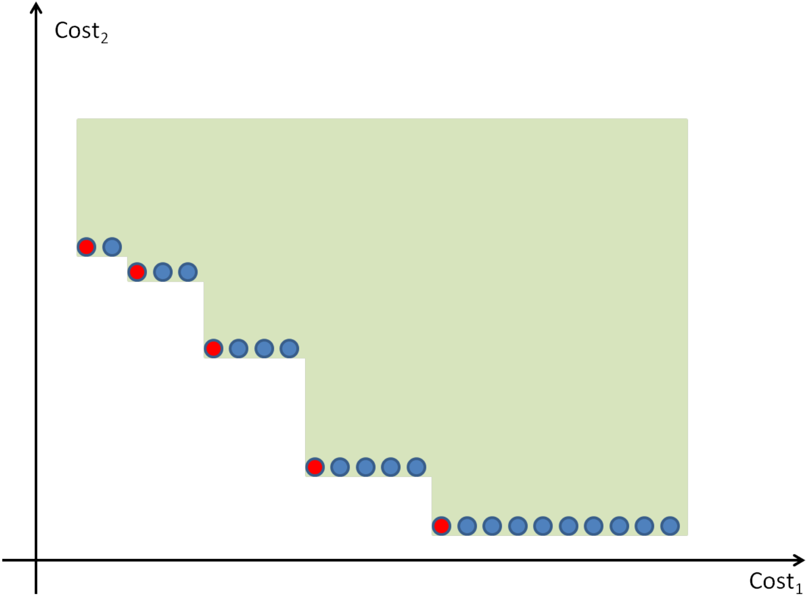


Algorithms: overview

Three approaches:

- *Valuations-first:* compute whole set of good valuations VALIDPARS up-front; then compute the Pareto front.
- *One-cost slicing:* we "slice" the space 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 VALIDPARS directly, but navigate through the valuations lattice driven by the cost functions and test on-the-fly membership of points to VALIDPARS.

Valuations-first Approach



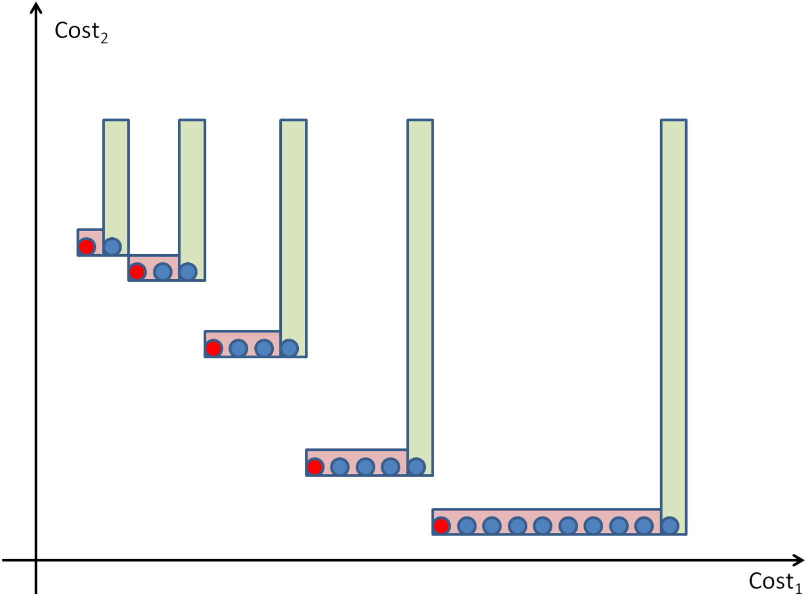
Valuations-first Approach

function $\textsc{ValuationsFirst}(S, \textsc{Cost}, \varphi)$
  $VP := \textsc{ValidPars}(S, \varphi)$
  return $\textsc{ParetoFront}(\textsc{Cost}, VP)$
end function

function $\textsc{ValidPars}(S, \varphi)$
  $Bad := \bot$
  $S = (U, X, I, T)$
  while $S \not\models \varphi$ **do**
    $\gamma' :=$ project counter-example on U
    $Bad := Bad \vee \gamma'$
    $I := I \wedge \neg Bad$
  end while
  return $\neg Bad$
end function

$$\textsc{ParetoFront}(U) = VP(U) \wedge \nexists U'.((U' \prec_{\textsc{Cost}} U) \wedge VP(U'))$$

One-cost slicing Approach



One-cost slicing Approach

function $\textsc{Slicing}(S, \textsc{Cost}, \varphi)$

  PF $:= \emptyset$; $\gamma = \top$;

  $c_1 := \textsc{Cost}_1(\gamma)$

  $S' := \textsc{FixCost}(S, \textsc{Cost}_1 = c_1)$

  $VP_{\textsc{Cost}_1} := \textsc{ValidPars}(S', \varphi)$

  while $VP_{\textsc{Cost}_1} \neq \emptyset$ **do**

    $(\gamma, c_2) = \textsc{Minimize}\ (\textsc{Cost}_2,\ VP_{\textsc{Cost}_1})$

    $(\gamma, c_1) := \textsc{Reduce}_{\textsc{Cost}_1}(S,\ \gamma,\ \varphi,\ c_2)$

    PF.add$(\gamma,\ c_1,\ c_2)$

    $c_1 := c_1 - 1$

    $S' := \textsc{FixCost}(S, \textsc{Cost}_1 = c_1)$

    $VP_{\textsc{Cost}_1} := \textsc{ValidPars}(S', \varphi)$

  end while

  return PF

end function

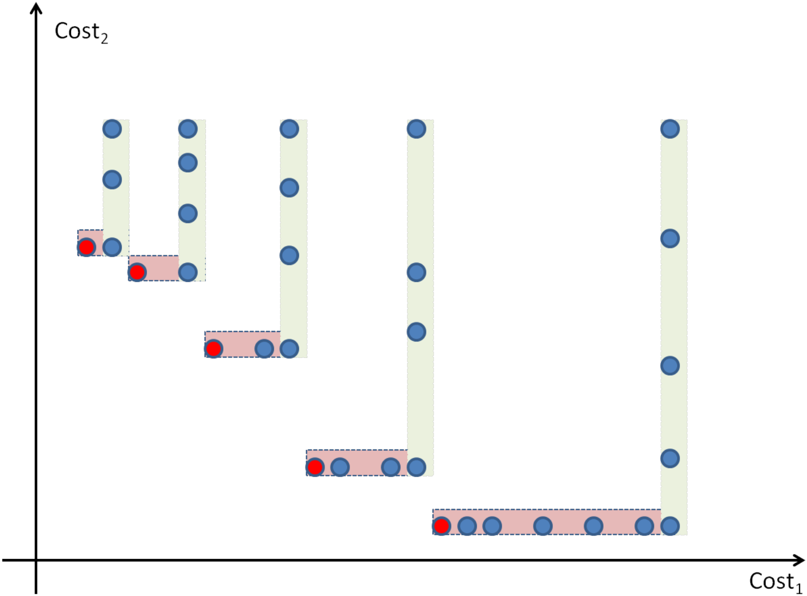
function $\textsc{FixCost}(\mathrm{S},\ \mathit{CostExpr})$

  $S = (U, X, I, T)$

  $S' := (U, X, I \wedge \mathit{CostExpr}, T)$ **return** S'

end function

Cost-first Approach



Cost-first Approach

function $\textsc{CostsFirst}(S, \textsc{Cost}, \varphi)$
    PF $:= \emptyset$
    $\gamma := \top$;
    $c_1 = \textsc{Cost}_1(\gamma)$; $\overline{c_2} = \textsc{Cost}_2(\gamma)$
    repeat
        $c_2 = \overline{c_2}$
        for $\gamma_i \in \textsc{MaxSmallerCandidate}_{\textsc{Cost}_2}(c_1, c_2)$ **do**
            if $S_{\gamma_i} \models \varphi$ **then**
                $(\gamma, c_2) := \textsc{Reduce}_{\textsc{Cost}_2}(S,\ \gamma,\ \varphi,\ c_1)$
            end if
        end for
        $(\gamma, c_1) := \textsc{Reduce}_{\textsc{Cost}_1}(S,\ \gamma,\ \varphi,\ c_2)$
        PF.add$(\gamma,\ c_1,\ c_2)$
        $c_1 := c_1 - 1$
    until No solution exists for $\textsc{FixCost}(S, \textsc{Cost}_1 = c_1)$
    return PF
end function

Cost-first Approach: IC3-based implementation

function $\textsc{CostsFirstIC3}(S, \textsc{Cost}, \varphi)$
  PF $:= \emptyset$
  $\gamma := \top$;
  $c_1 = \textsc{Cost}_1(\gamma)$; $\overline{c_2} = \textsc{Cost}_2(\gamma)$
  repeat
    $c_2 := \overline{c_2}$
    for $\gamma_i \in \textsc{MaxSmallerCandidate}_{\textsc{Cost}_2}(c_1, c_2)$ **do**
      $(res, \psi) :=$ IC3$(S, \gamma_i \rightarrow \varphi)$ // $S_{\gamma_i} \models \varphi$ *iff* $S \models \gamma_i \rightarrow \varphi$
      if $res ==$ Safe **then**
        // ψ *is an inductive invariant s.t.* $\psi \models \gamma_i \rightarrow \varphi$
        $(\gamma_i, c_1, c_2) := \textsc{Reduce}_{\textsc{Cost}_2}(\psi, \gamma_i, \varphi)$
      end if
    end for
    $(\gamma_i, c_1, c_2) := \textsc{Reduce}_{\textsc{Cost}_1}(\psi, \gamma_i, \varphi)$
    PF.add$(\gamma,\ c_1,\ c_2)$
    $c_1 := c_1 - 1$
  until No solution exists for $FixCost(S, \textsc{Cost}_1 = c_1)$
  return PF
end function

Motivating domain

Sensor Placement:

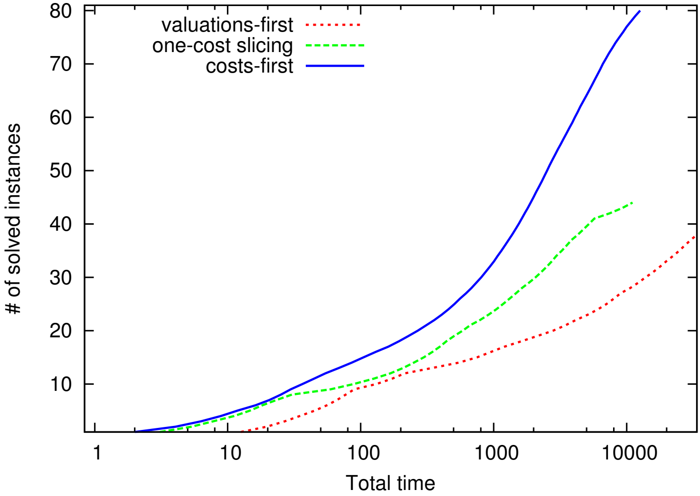
- Are the sensors enough to guarantee diagnosability?
- More sensors imply better diagnosability.
- Sensors have costs, weights, ...
- Find corresponding Pareto front to explore trade-off

Benchmarks from sensor placement and product lines.

Experiments: solved instances

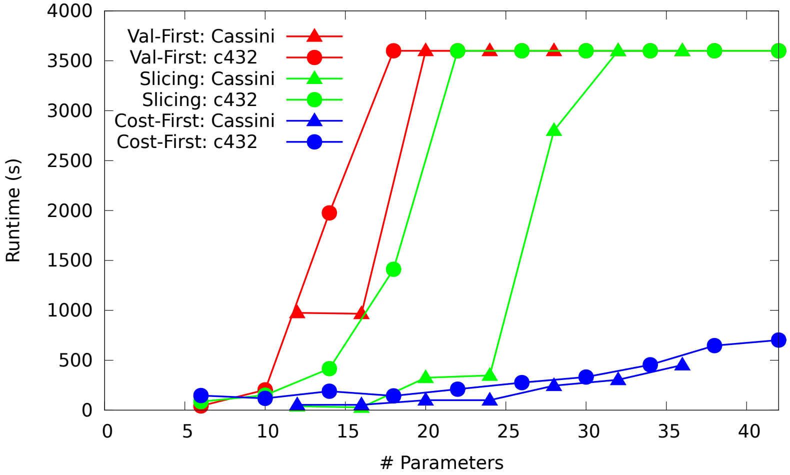
Family	#Instances	valuations-first	one-cost slicing	costs-first
c432	32	11	13	**32**
cassini	21	6	12	**21**
elevator	4	4	4	4
orbiter	4	4	4	4
roversmall	4	4	4	4
roverbig	4	4	4	4
x34	4	4	4	4
product lines	8	6	4	**8**
TOTAL	81	43	49	**81**

Experiments: performance



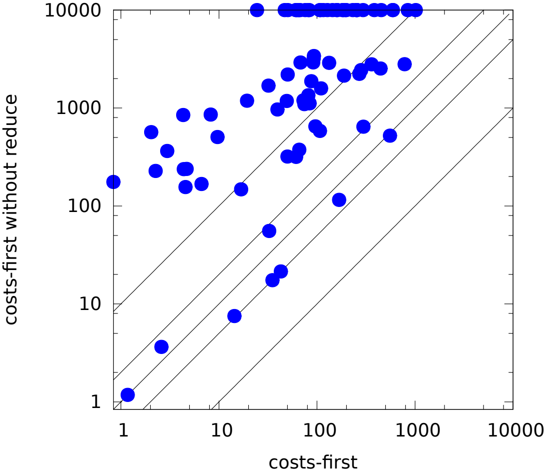
Accumulated-time plot showing the number of solved instances (x-axis) in a given total time (y-axis) for the various algorithms.

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?

References I

É. André, L. Fribourg, U. Kühne, and R. Soulat.
IMITATOR 2.5: A tool for analyzing robustness in scheduling problems.
In *FM*, pages 33–36, 2012.

É. André and U. Kühne.
Parametric analysis of hybrid systems using HyMITATOR.
In *iFM*, pages 16–19, 2012.

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.
In *ATVA*, pages 162–176. Springer, 2007.

References II

A. Cimatti, A. Griggio, S. Mover, and S. Tonetta.
Parameter synthesis with ic3.
In *FMCAD*, pages 165–168. IEEE, 2013.

A. Classen, P. Heymans, P.-Y. Schobbens, and A. Legay.
Symbolic model checking of software product lines.
In *ICSE*, pages 321–330, 2011.

A. Cimatti, L. Palopoli, and Y. Ramadian.
Symbolic computation of schedulability regions using parametric timed automata.
In *RTSS*. IEEE Computer Society, 2008.

G.Frehse, S.K. Jha, and B.H. Krogh.
A counterexample-guided approach to parameter synthesis for linear hybrid automata.
In *HSCC*, pages 187–200, 2008.

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.

In *Hybrid Systems*, pages 265–293, 1994.

C. Hang, P. Manolios, and V. Papavasileiou.

Synthesizing cyber-physical architectural models with real-time constraints.

In *CAV*, pages 441–456, 2011.

J. Legriel, C. Le Guernic, S. Cotton, and O. Maler.

Approximating the pareto front of multi-criteria optimization problems.

In *TACAS*, pages 69–83, 2010.

References IV

S. Mamagkakis, D. Atienza, C. Poucet, F. Catthoor, D. Soudris, and J.M. Mendias.

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

In *DATE*, pages 874–875, 2006.

P. Manolios, D. Vroon, and G. Subramanian.

Automating component-based system assembly.

In *ISSTA*, pages 61–72, 2007.

V. Pareto.

Manuale di economia politica.

Collezione saggi & documenti. Edizioni Studio Tesi, 1994.

S. Reimer, M. Sauer, T. Schubert, and B. Becker.

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.

S. Samii, P. Eles, Z. Peng, and A. Cervin.
Design optimization and synthesis of flexray parameters for embedded control applications.
In *DELTA*, pages 66–71, 2011.

R. Schneider, D. Goswami, S. Zafar, M. Lukasiewycz, and S. Chakraborty.
Constraint-driven synthesis and tool-support for flexray-based automotive control systems.
In *CODES+ISSS*, pages 139–148, 2011.

References VI

F. Wang.

Symbolic parametric safety analysis of linear hybrid systems with bdd-like data-structures.

IEEE Trans. Soft. Eng., 31(1):38–51, 2005.