Technische Universiteit Eindhoven University of Technology

Embedded Systems INSTITUTE

Reliable Run-time Adaptation in Resource-constrained Embedded Systems

Twan Basten

Joint work with

Marc Geilen, AmirHossein Ghamarian, Rob Hoes, Hamid Shojaei, Sander Stuijk, and others

Funding:
EC FP6 Betsy & WASP
EC FP7 MNEMEE

'Knowing is not understanding.'
Charles Kettering

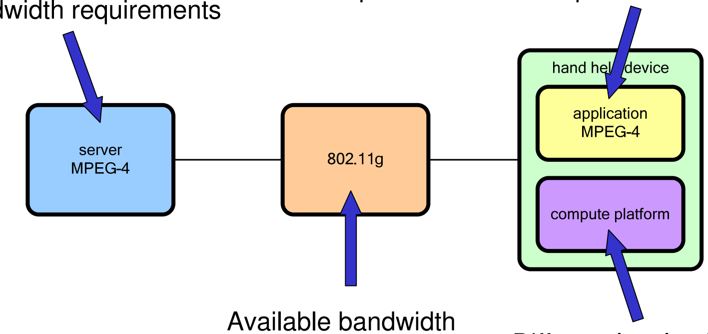
2 Run-time Adaptation

Encoding qualities

Bandwidth requirements

Decoding streams of different quality

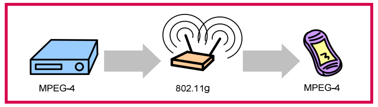
Computational effort required



Available bandwidth under given circumstances

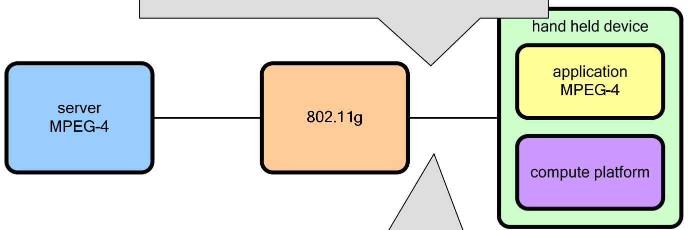
Different levels of computation power

Different levels of power consumption

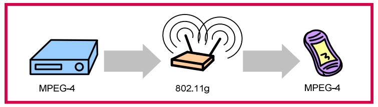


3 Run-time Adaptation

What is the optimal configuration?
- given communication bandwidth
- given battery status
- given user preferences



Requires run-time decision making
- multi-dimensional problem
- strict timing constraints
- limited resources

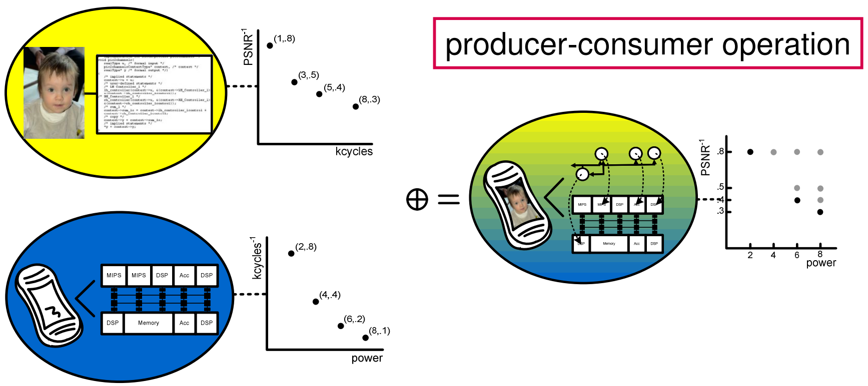


Pareto Algebra

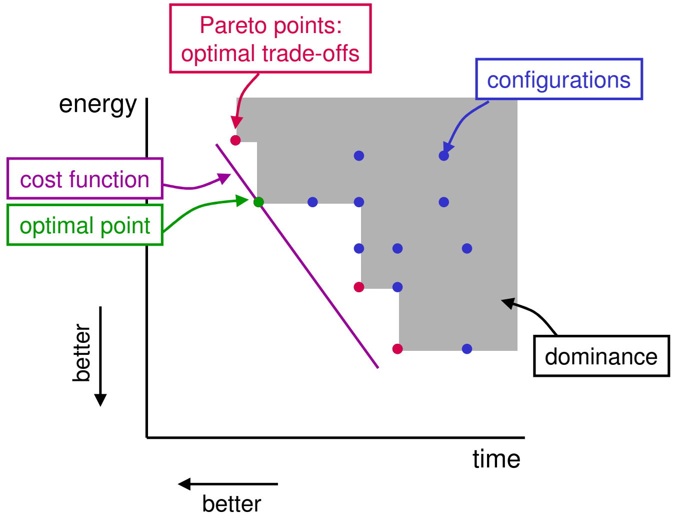
Fundamenta Informaticae, 2007

5 Approach: Pareto Algebra

Goal: Compositional computation of trade-offs

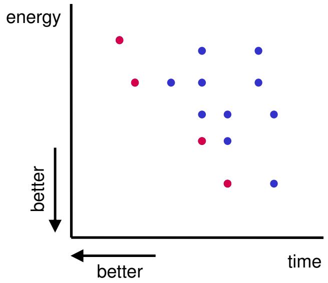


6 A Pareto Space



7 Pareto Algebra

- The elements: sets of configurations
- (Relevant) operators:
  - **Minimization**
    Gives the Pareto points (optimal trade-offs)
  - **Product**
    Cartesian product of configurations
    e.g. application and platform configurations
  - **Constraint**
    Selects solutions according to constraints
    e.g. all application configurations with some minimal quality
  - **Abstraction**
    Discards information about solutions
    e.g. bandwidth usage in bandwidth × energy × quality configurations
  - **Derived metric**
    Derive a new metric from other metrics
    e.g. total power from power of components

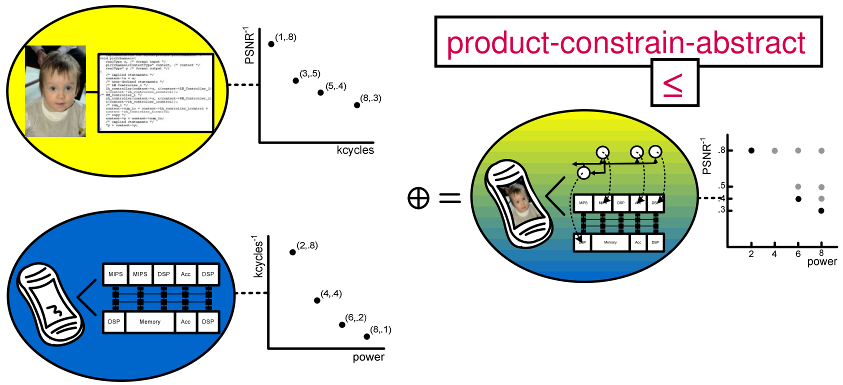


prerequisite
monotonicity

8 Compound Operations

- producer-consumer operation

product-constrain-abstract
≤



- join operation (relational databases)

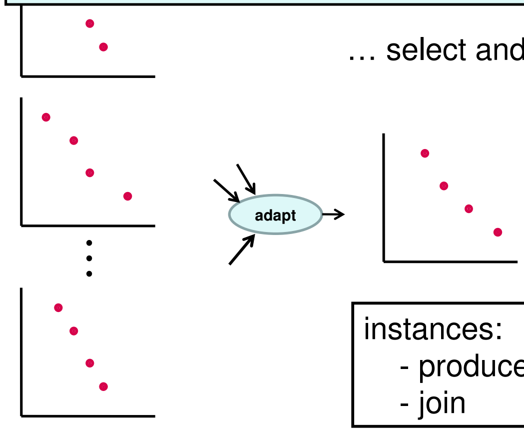
product-constrain-abstract
=

Run-time Adaptation

10 Pareto-Algebraic Characterization

… of run-time adaptation

product - derive | constrain | abstract - minimize

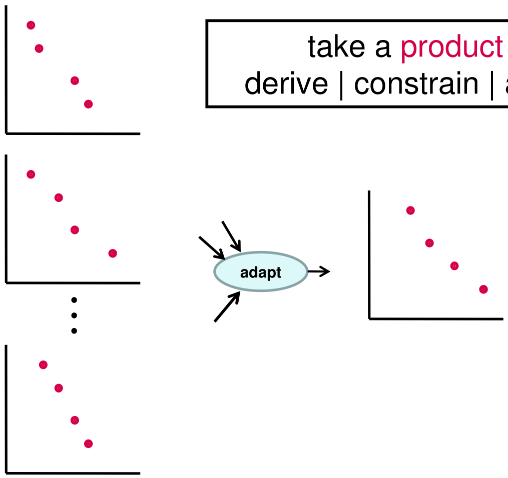


… select and configure

instances:
- producer-consumer
- join

11 Complexity

n components with p Pareto points each



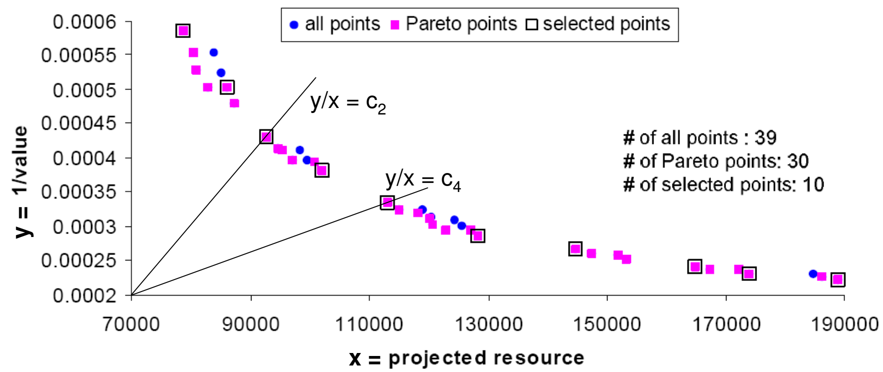
take a product and on-the-fly
derive | constrain | abstract - minimize

$O(p^{2n})$

12 Complexity Control

keep n and p small

- Compositional reasoning, among others
- Approximate Pareto sets



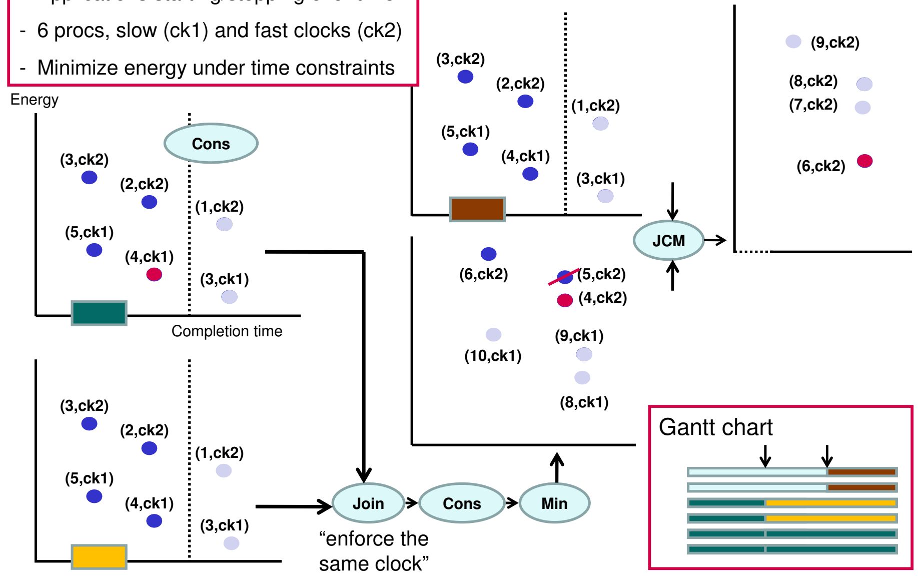
Chip Multi-Processors

DAC 2009

14 Run-time Application Management

Embedded Systems INSTITUTE TU/e

- Applications starting/stopping over time
- 6 procs, slow (ck1) and fast clocks (ck2)
- Minimize energy under time constraints



15 Multi-dimensional Multiple-choice Knapsack Problem

MMKP

NP hard

- One optimization objective (value)
- Multiple resource dimensions with capacity constraints
- Multiple independent applications
- Multiple independent configurations per application

Pick one configuration per application optimizing value within constraints

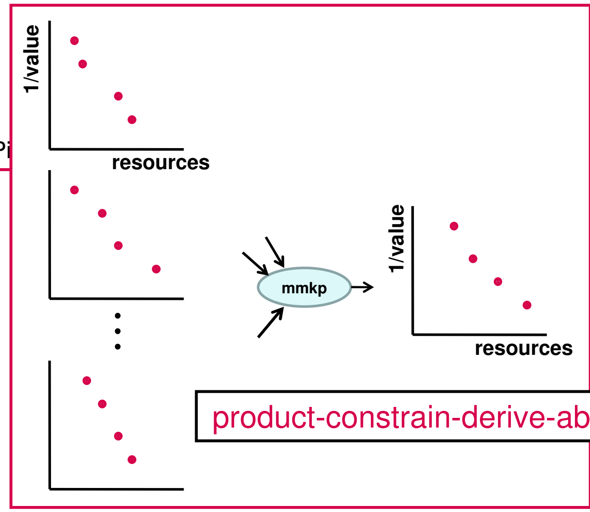
16 Multi-dimensional Multiple-choice Knapsack Problem

MMKP

NP hard

- One optimization objective (value)
-
-
-

Pi ... in constraints



product-constrain-derive-abstract-minimize

17 A Parameterized Compositional Heuristic

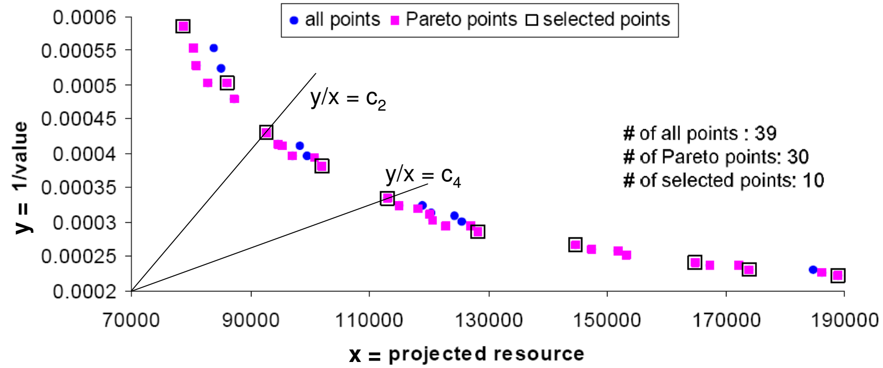
- Project all resource dimensions into one dimension
- Approximate Pareto set in 2-dimensional space

product constrain derive abstract minimize	Compute product while on-the-fly • Applying resource constraints • Computing values, projecting all resource dimensions into one (taking simply the sum) • Minimizing the configuration set in 2-dimensional space

product-constrain-derive-abstract-minimize

18 A Parameterized Compositional Heuristic

- Project all resource dimensions into one dimension
- Approximate Pareto set in 2-dimensional space



product-constrain-derive-abstract-minimize

19 A Parameterized Compositional Heuristic

- Project all resource dimensions into one dimension
- Approximate Pareto set in 2-dimensional space

parameter p: maintain (at most) p Pareto points

Allows to budget analysis time

analysis time $\leq c \cdot n \cdot p^2$

(n number of applications; c platform constant)

product-constrain-derive-abstract-minimize

20 MMKP Benchmark

(www.laria.u-picardie.fr/hifi/OR-Benchmark/MMKP)

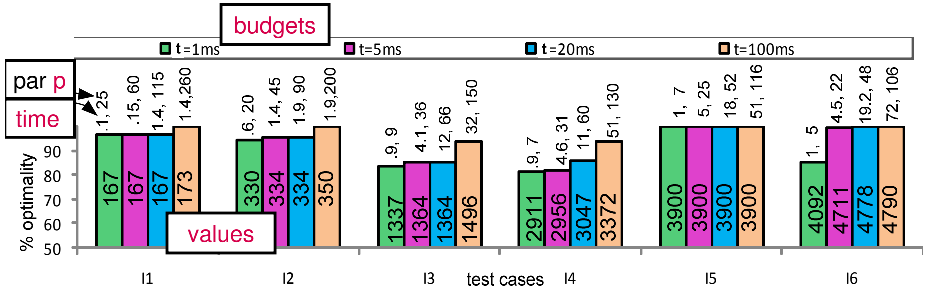
Test case	# of applications	# of configurations	# of resources
I1	**5**	**5**	**5**
I2	**10**	**10**	**5**
I3	**15**	**10**	**10**
I4	**20**	**10**	**10**
I5	**25**	**10**	**10**
I6	**30**	**10**	**10**
I7	100	10	10
I8	150	10	10
I9	200	10	10
I10	250	10	10
I11	300	10	10
I12	350	10	10
I13	400	10	10

CMP cases

Tested on the SimIt-ARM simulator 206 Mhz StrongARM

large cases

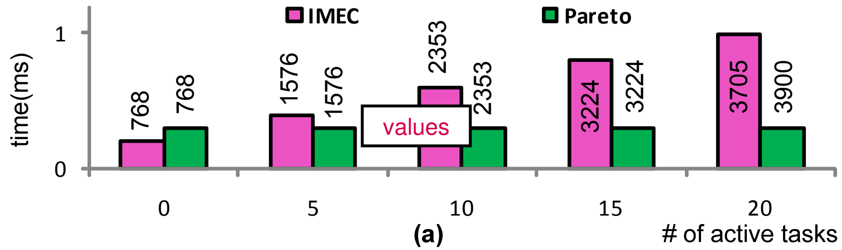
21 Controlling Time Budgets

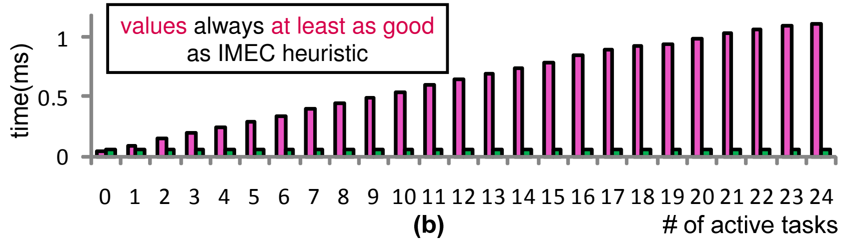


- Always within budget
- Good values
- Similar results to the IMEC, SOC 2006, non-compositional state-of-the-art heuristic

22 Compositionality

- Applications start at various times
- 5 at a time (a), 1 at a time (b)





Wireless Sensor Networks

24 Configuring a Static WSN

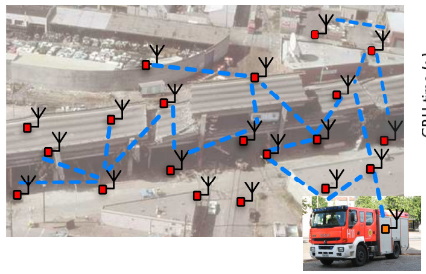
Wireless sensor network

- 900 nodes
- 27 configurations per node
- 4 quality metrics
- Routing tree to data sink

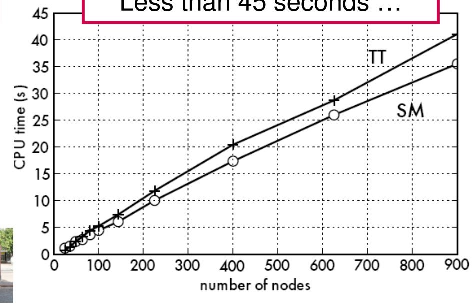
Problem
How to configure the network ?

Approach
Compositional computation following the routing-tree structure

27^{900} configurations in 4D space

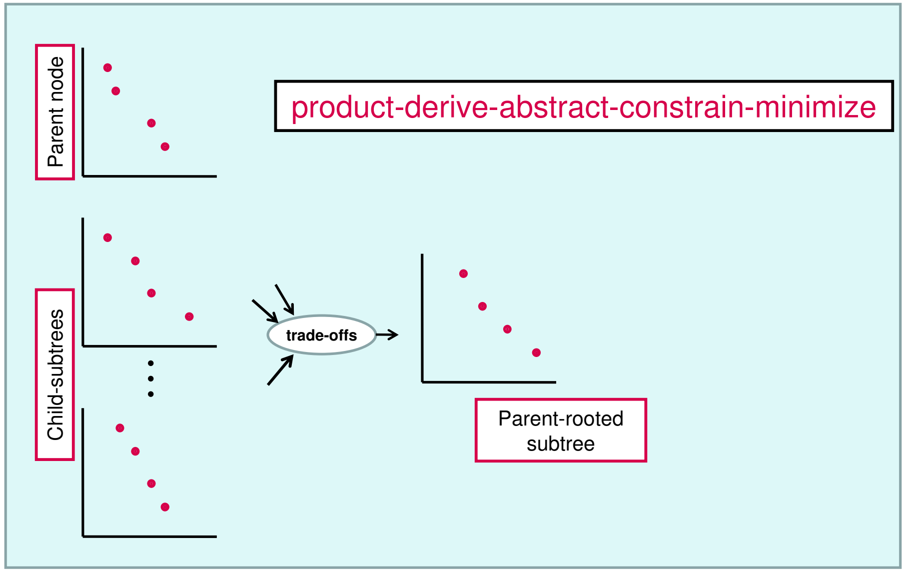


Less than 45 seconds …

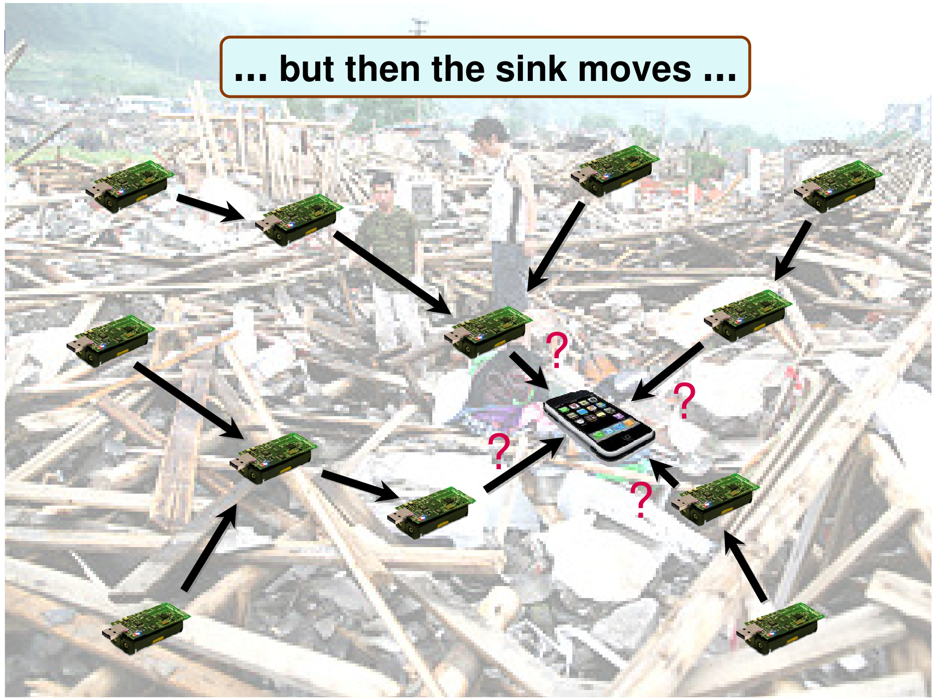


MSWiM 2007 best paper

25 Configuring a Static WSN



MSWiM 2007 best paper

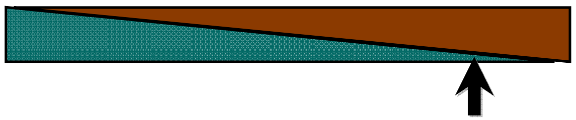


27 Run-time Adaptation

- Repair the routing tree
- Compute QoS trade-offs
- Select and load new configuration

Trade-off
reconfiguration cost vs. quality

global reconfiguration

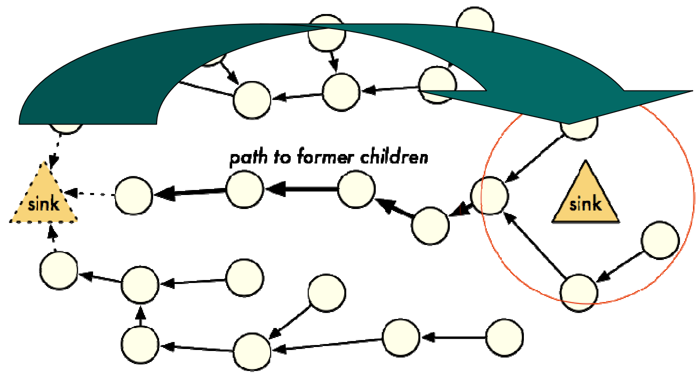


local reconfiguration

Parameters

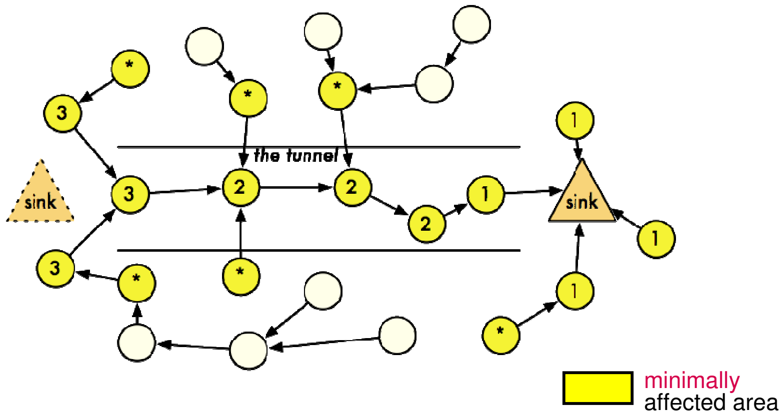
- d: deviation from minimally affected area
- n: max number of children per node
- p: max number of Pareto points

28 Repairing the Tree: QuickFix



Repairing the Tree: QuickFix

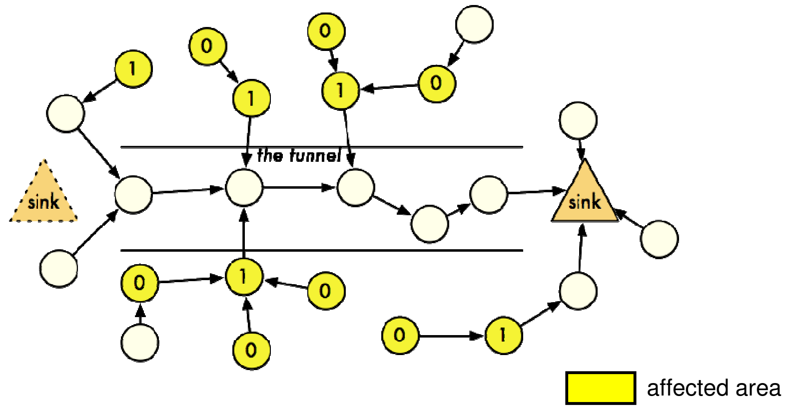
Reverse the path from the former children of the sink to the new children



Quality Improvement: Controlled Flooding

Embedded Systems INSTITUTE TU/e

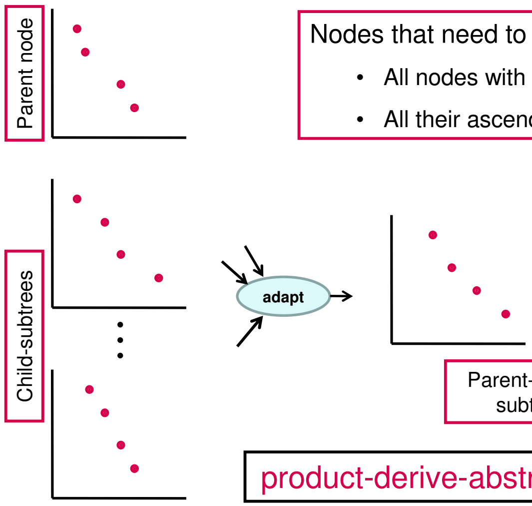
Reconfigure up to some given deviation depth d from the affected area



motivation → Pareto algebra → adaptation → cmp → wsn → conclusions

Computation at a Given Node

Embedded Systems INSTITUTE TU/e



Nodes that need to recompute Pareto sets:

- All nodes with a new parent or new children
- All their ascendants

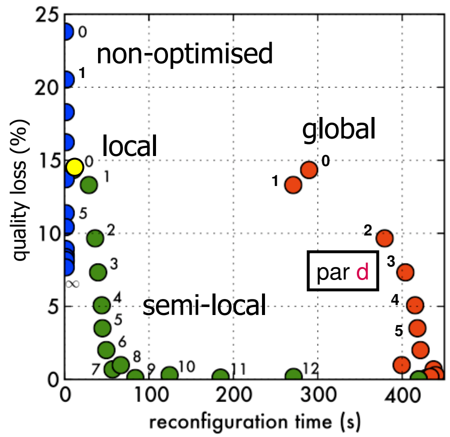
Parent-rooted subtree

product-derive-abstract-constrain-minimize

motivation → Pareto algebra → adaptation → cmp → wsn → conclusions

Reconfiguration Time vs. Quality

Embedded Systems INSTITUTE TU/e



900 node network

Simulated in the OMNET++ simulator
Calibrated for TelosB sensor nodes
8 Mhz processor
250 kbps tranceiver bit rate

Good quality reconfiguration within one minute

motivation → Pareto algebra → adaptation → cmp → wsn → conclusions

Conclusions

- Run-time adaptation:
  product - derive | constrain | abstract - minimize
  … select and configure
- Parameterized compositional method
  - Allows to trade off quality with analysis time
  - Allows to bound analysis time
- Feasible for resource-constrained embedded systems
- Wide variety of applications

Thank you !

Embedded Systems INSTITUTE TU/e

Questions ?

More info: www.es.ele.tue.nl/~tbasten/
www.es.ele.tue.nl/pareto/

'An understanding of the natural world and what's in it is a source of not only a great curiosity but great fulfillment.'
David Attenborough