

Reliable Run-time Adaptation in Resource-constrained Embedded Systems

Twan Basten

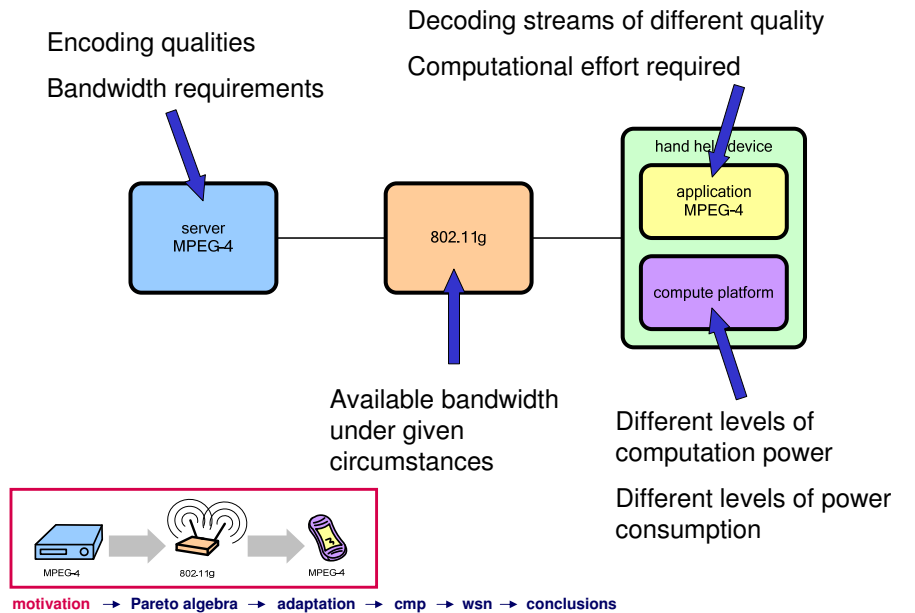
Joint work with

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

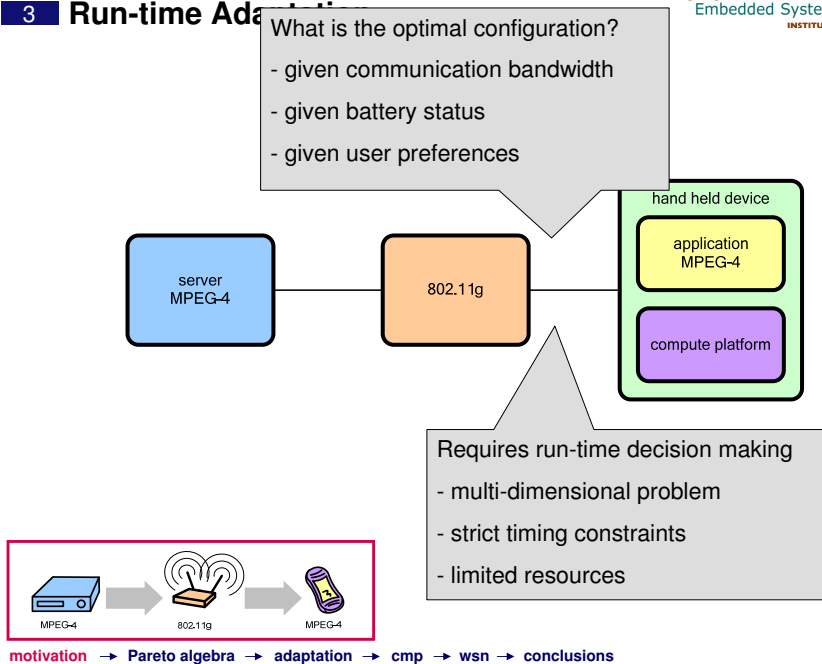
Funding:
EC FP6 Betsy & WASP
EC FP7 MNEMEE

'Knowing is not understanding.'
Charles Kettering

2 Run-time Adaptation



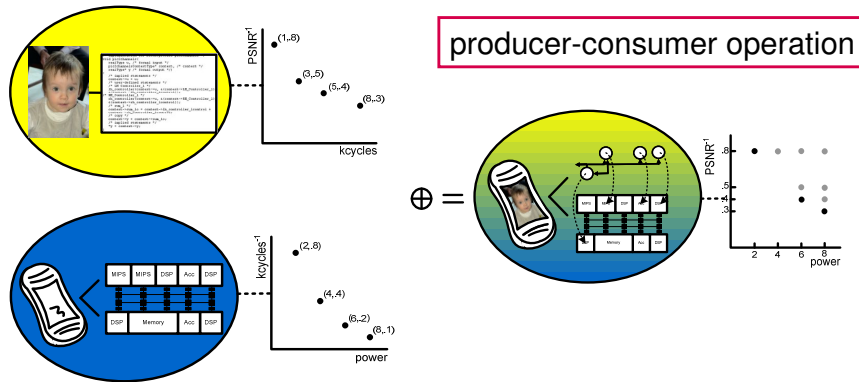
3 Run-time Adaptation



Pareto Algebra

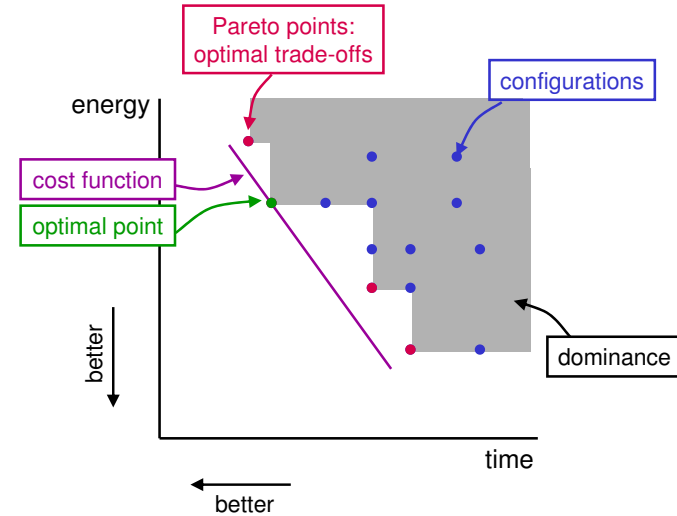
5 Approach: Pareto Algebra

Goal: Compositional computation of trade-offs



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

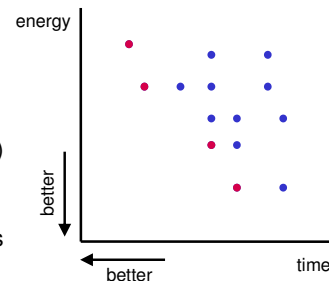
6 A Pareto Space



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

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

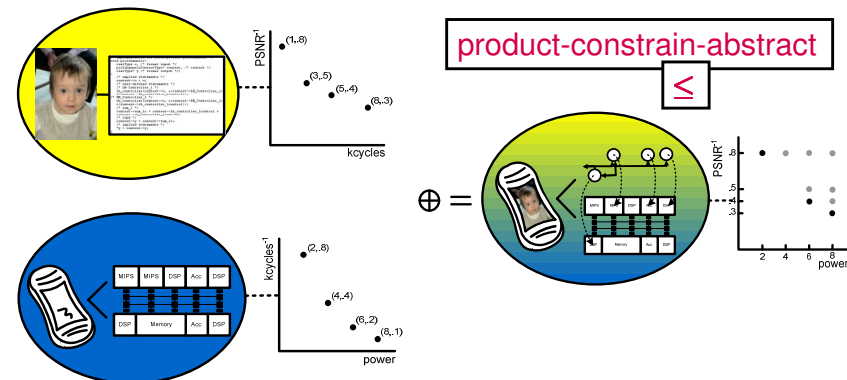


prerequisite
monotonicity

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

8 Compound Operations

- producer-consumer operation



- join operation (relational databases)

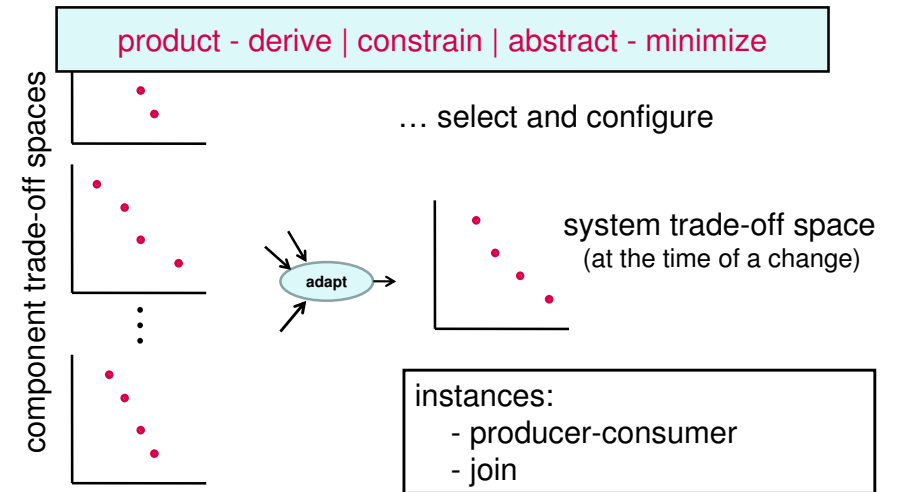
product-constrain-abstraction
=

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

Run-time Adaptation

10 Pareto-Algebraic Characterization

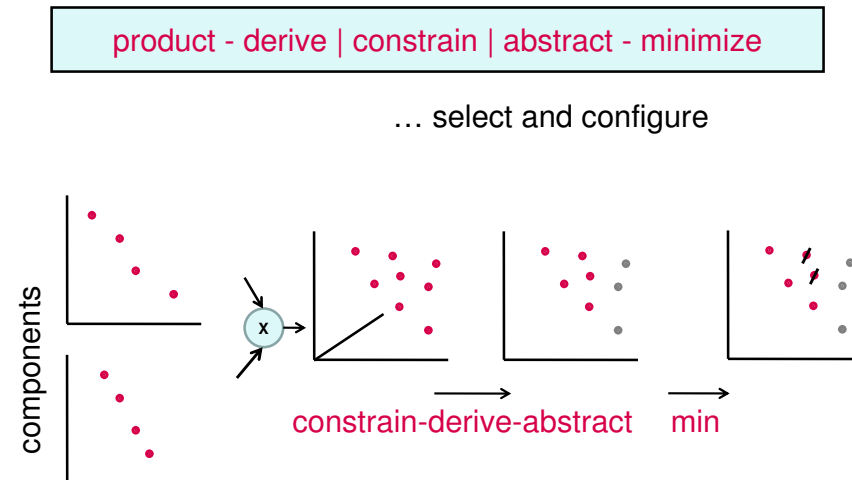
... of run-time adaptation



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

11 Pareto-Algebraic Characterization

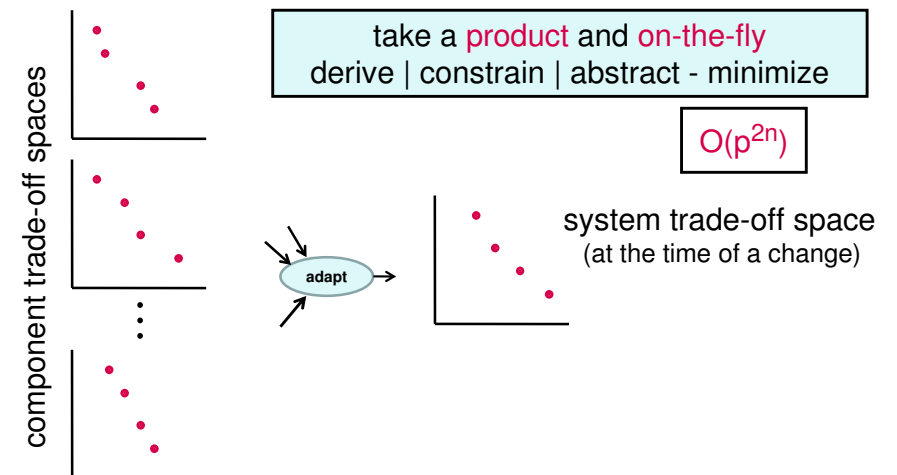
... of run-time adaptation



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

12 Complexity

n components with p Pareto points each



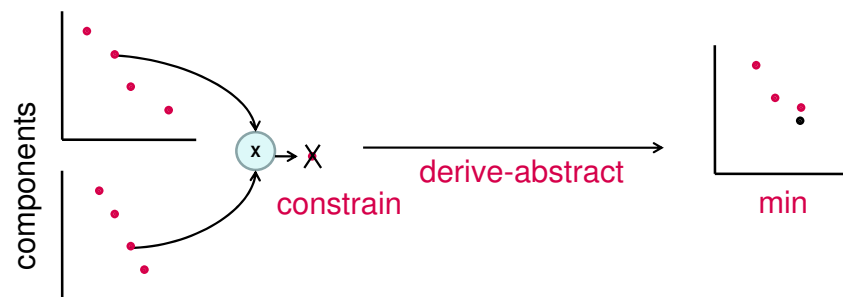
motivation → Pareto algebra → adaptation → cmp → wsn → conclusions

13 Complexity

n components with p Pareto points each

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

$$O(p^{2n})$$

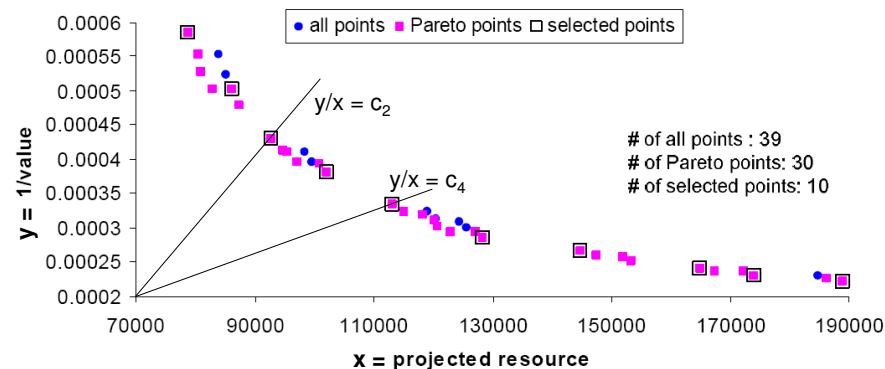


motivation → Pareto algebra → **adaptation** → **cmp** → **wsn** → conclusions

14 Complexity Control

keep n and p small

- Compositional reasoning, among others
- Approximate Pareto sets

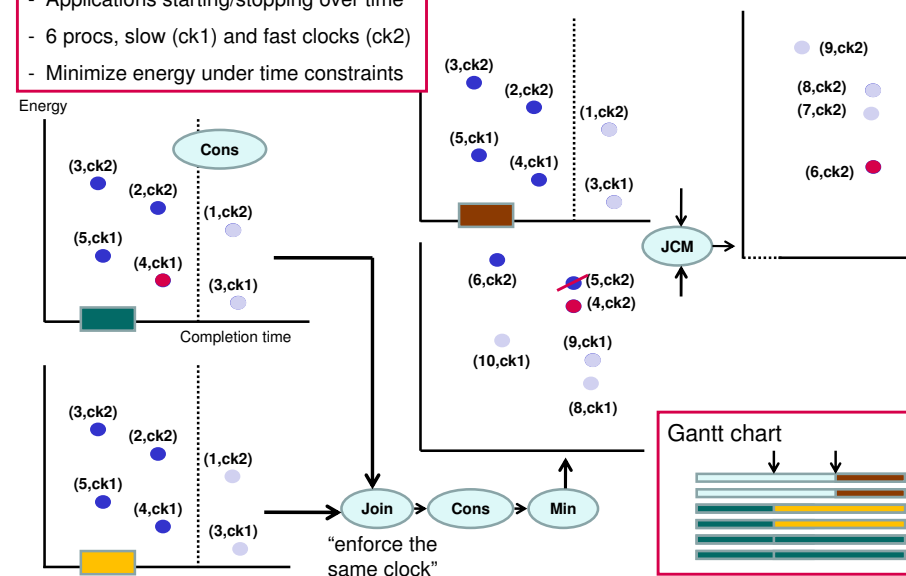


motivation → Pareto algebra → **adaptation** → **cmp** → **wsn** → conclusions

Chip Multi-Processors

16 Run-time Application Management

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



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

17 Multi-dimensional Multiple-choice Knapsack Problem

MMKP

- 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

NP hard

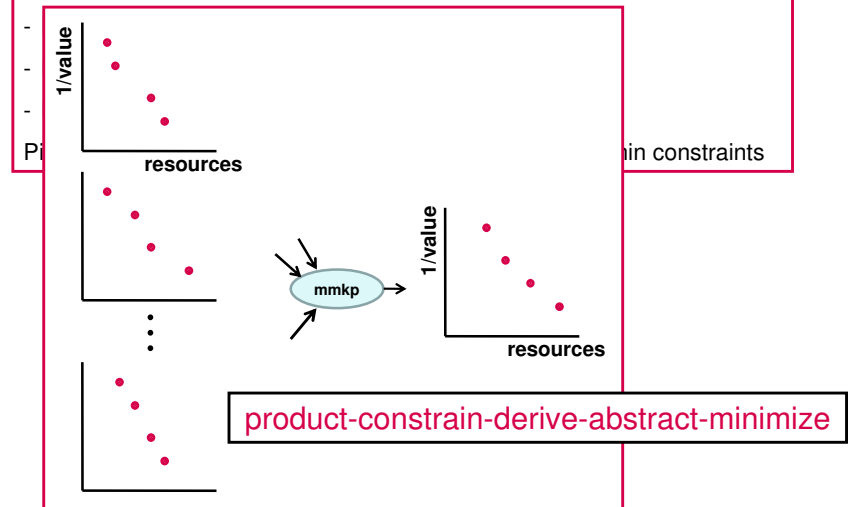
motivation → Pareto algebra → adaptation → cmp → wsn → conclusions

18 Multi-dimensional Multiple-choice Knapsack Problem

MMKP

- One optimization objective (value)

NP hard



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

19 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

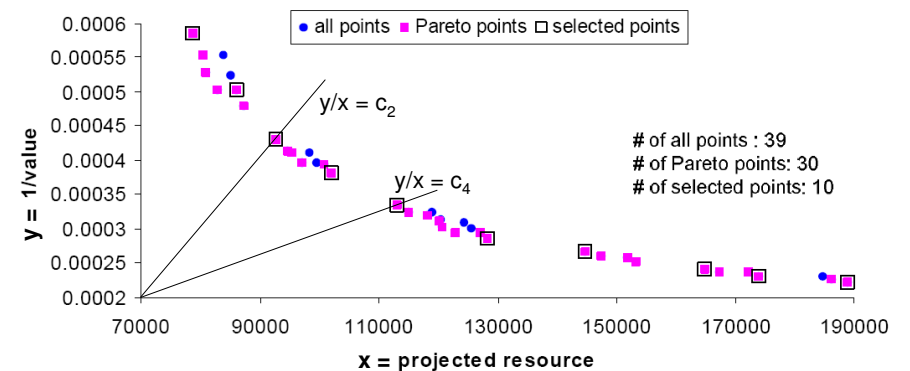
- Applying resource constraints
- Computing values, projecting all resource dimensions into one (taking simply the sum)
- Minimizing the configuration set in 2-dimensional space

product-constrain-derive-abstract-minimize

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

20 A Parameterized Compositional Heuristic

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



product-constrain-derive-abstract-minimize

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

21 A Parameterized Compositional Heuristic Embedded Systems INSTITUTE TU/e

- 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

$$\text{analysis time} \leq c \cdot n \cdot p^2$$

(*n* number of applications; *c* platform constant)

product-constrain-derive-abstract-minimize

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

22 MMKP Benchmark Embedded Systems INSTITUTE TU/e

(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

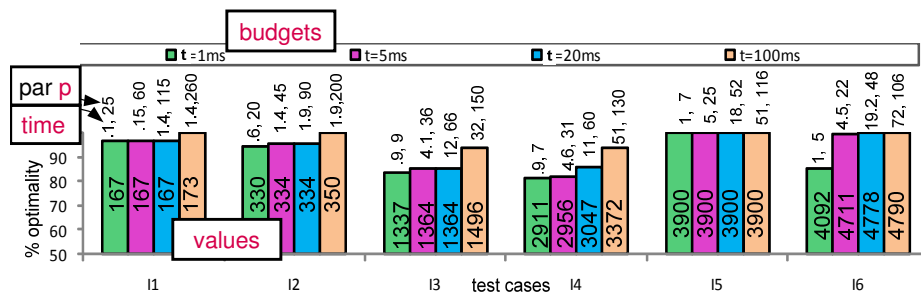
CMP cases

Tested on the
SimIt-ARM simulator
206 Mhz StrongARM

large cases

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

23 Controlling Time Budgets Embedded Systems INSTITUTE TU/e



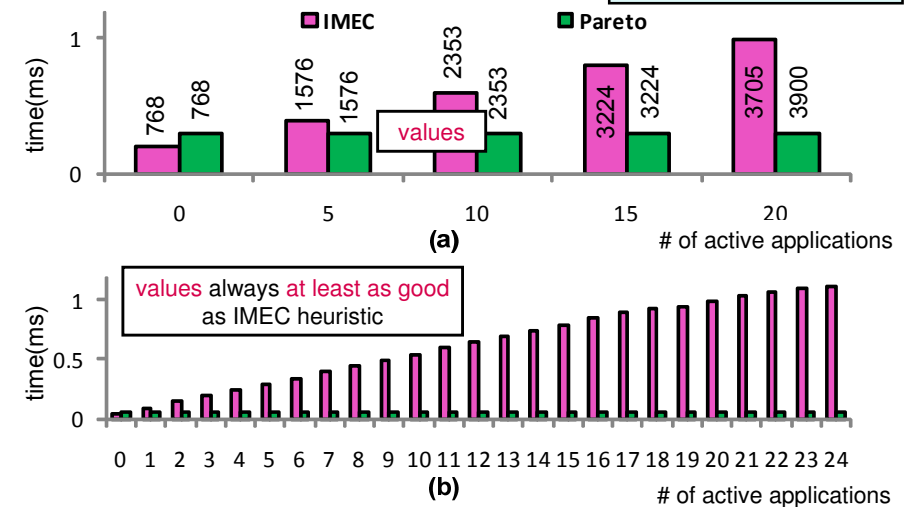
- Always within budget
- Good values
- Similar results to the IMEC, SOC 2006, non-compositional state-of-the-art heuristic

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

24 Compositionality Embedded Systems INSTITUTE TU/e

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

what about applications stopping ?



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

Wireless Sensor Networks

EWSN 2009

26 Configuring a Static WSN

Wireless sensor network

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

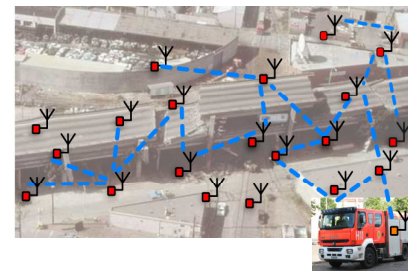
Problem

How to configure the network ?

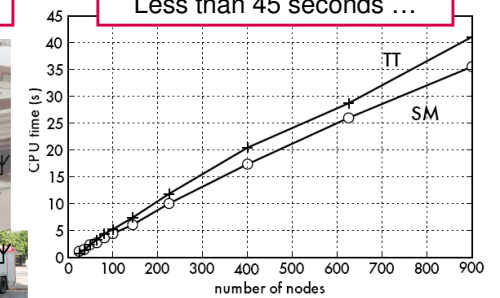
Approach

Compositional computation following the routing-tree structure

27^{900} configurations in 4D space



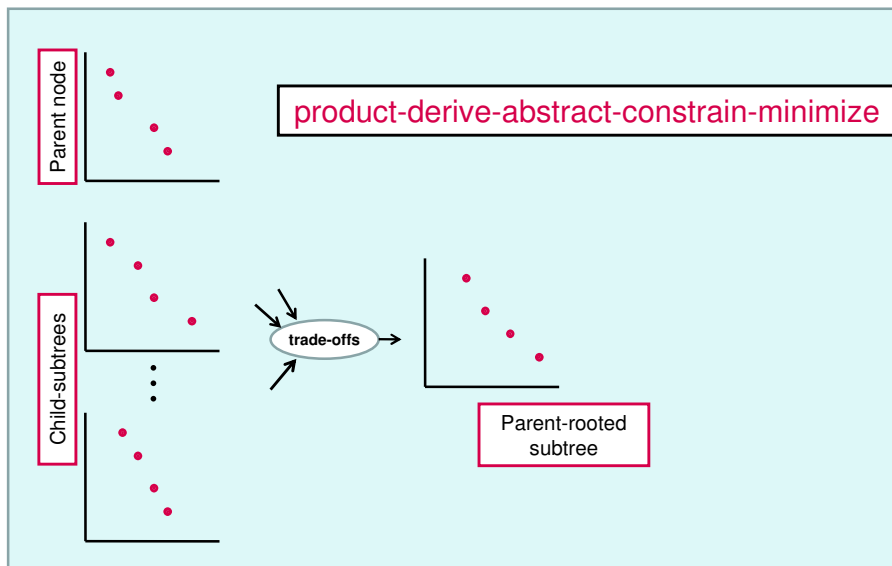
Less than 45 seconds ...



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

MSWiM 2007 best paper

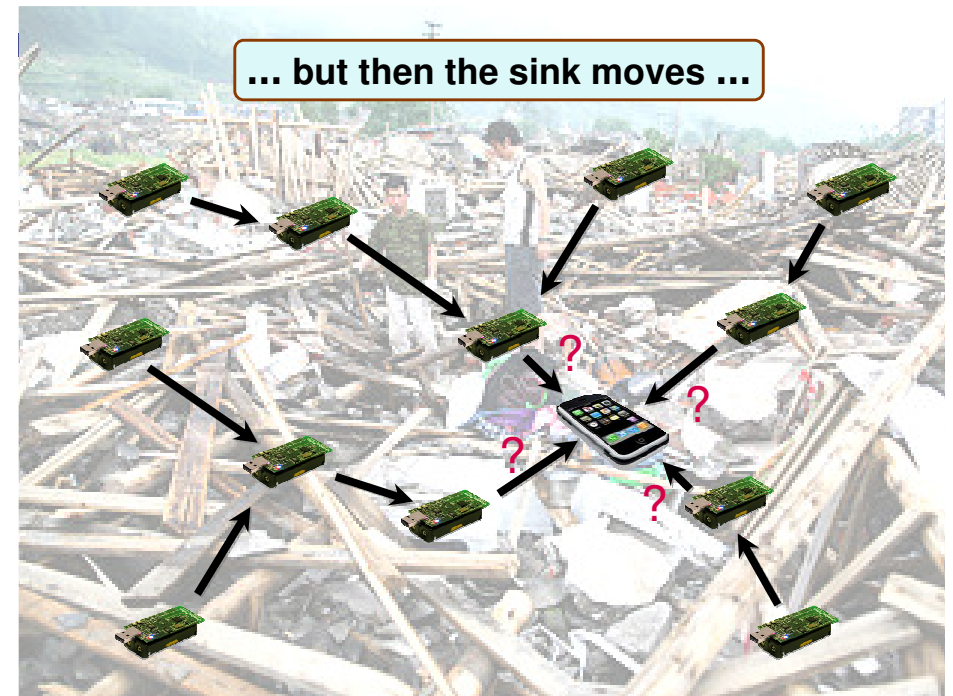
27 Configuring a Static WSN



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

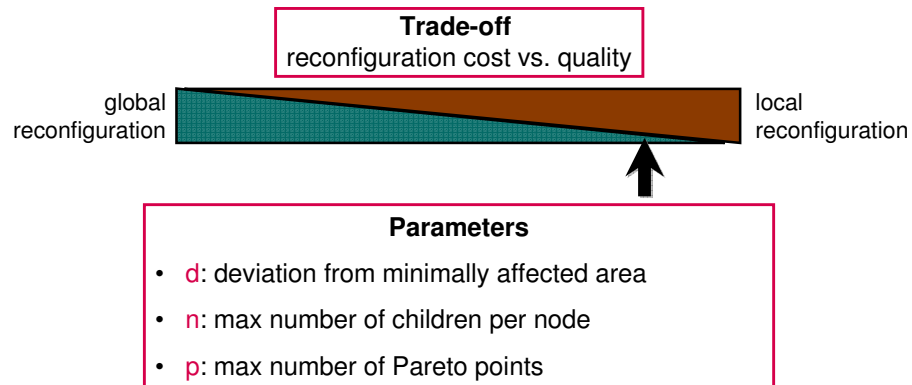
MSWiM 2007 best paper

... but then the sink moves ...



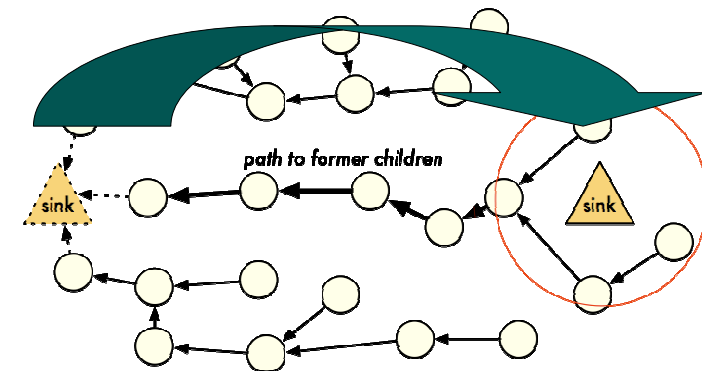
29 Run-time Adaptation

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



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

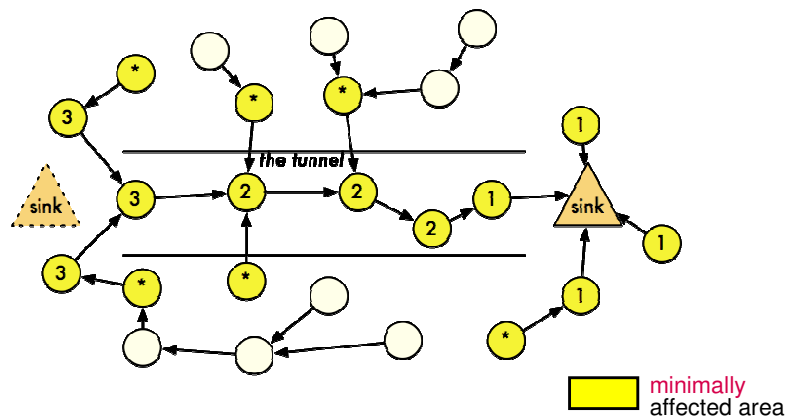
30 Repairing the Tree: QuickFix



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

31 Repairing the Tree: QuickFix

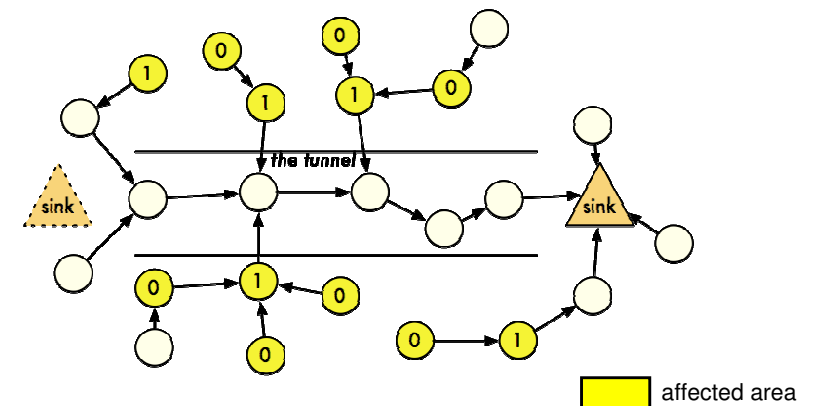
Reverse the path from the former children of the sink to the new children



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

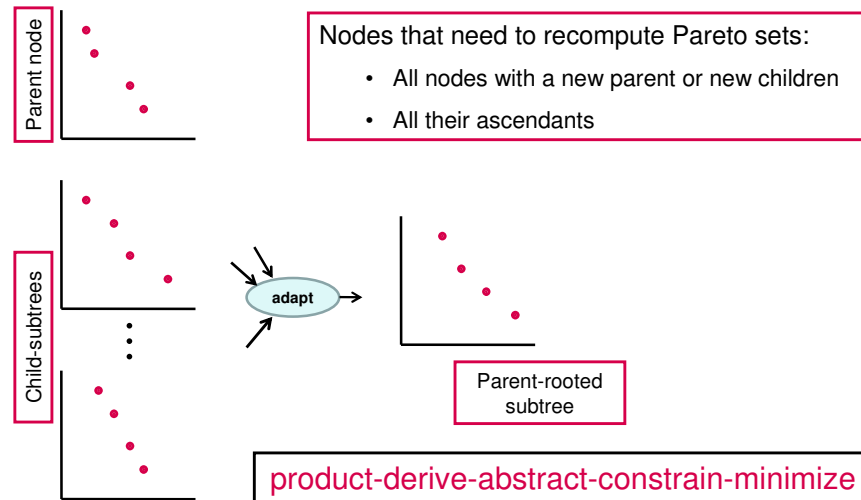
32 Quality Improvement: Controlled Flooding

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



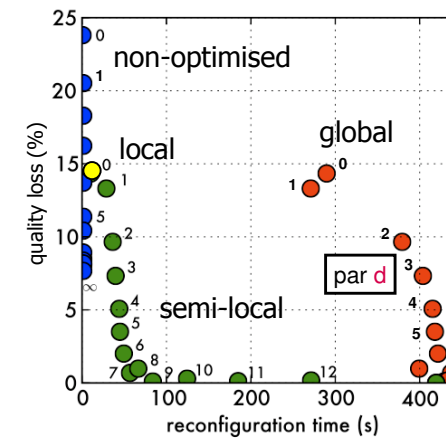
motivation → Pareto algebra → adaptation → cmp → **wsn** → conclusions

33 Computation at a Given Node



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

34 Reconfiguration Time vs. Quality



900 node network
Simulated in the OMNET++ simulator
Calibrated for TelosB sensor nodes
8 Mhz processor
250 kbps transceiver bit rate

Good quality reconfiguration within
one minute

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

35 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
 - Open question: components leaving the composition ?
- Feasible for resource-constrained embedded systems
- Wide variety of applications

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

36 Thank you !

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