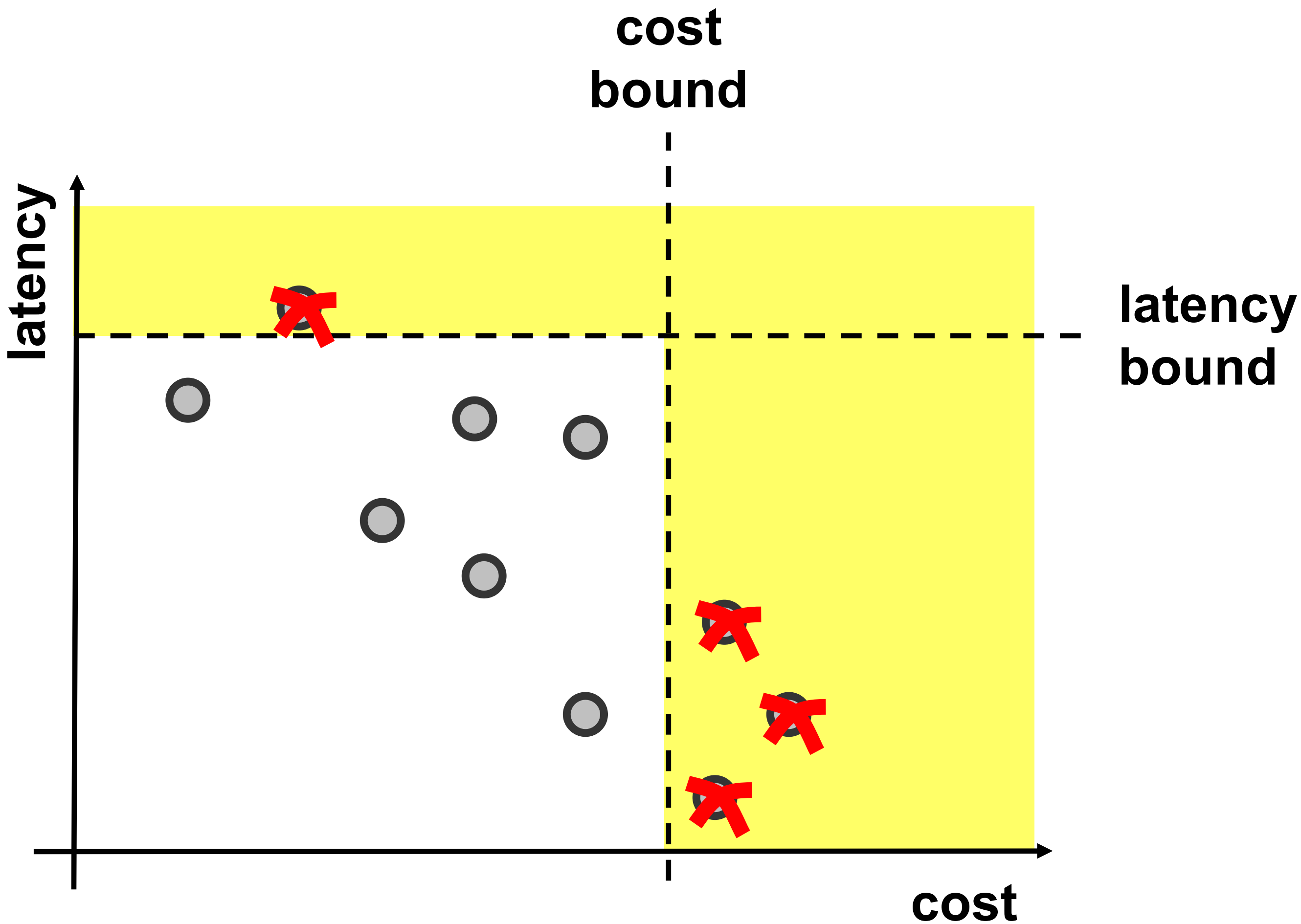
Accelerated Design Space Exploration for Heterogeneous MPSoCs Using Symbolic Techniques

3rd Workshop on Mapping of Applications to MPSoCs 2010, June 29th

Christian Haubelt (haubelt@cs.fau.de)
University of Erlangen-Nuremberg, Germany

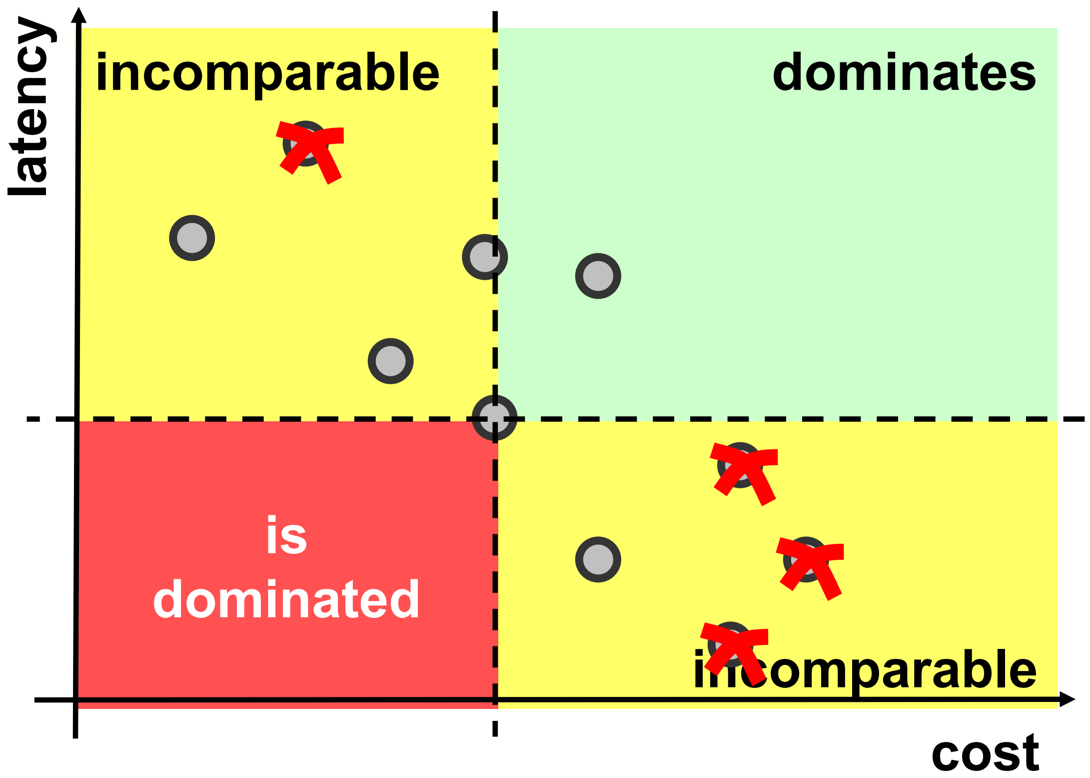
With contributions from:
M. Glaß, M. Lukasiewycz, F. Reimann, T. Schlichter, J. Teich

Design Space Exploration



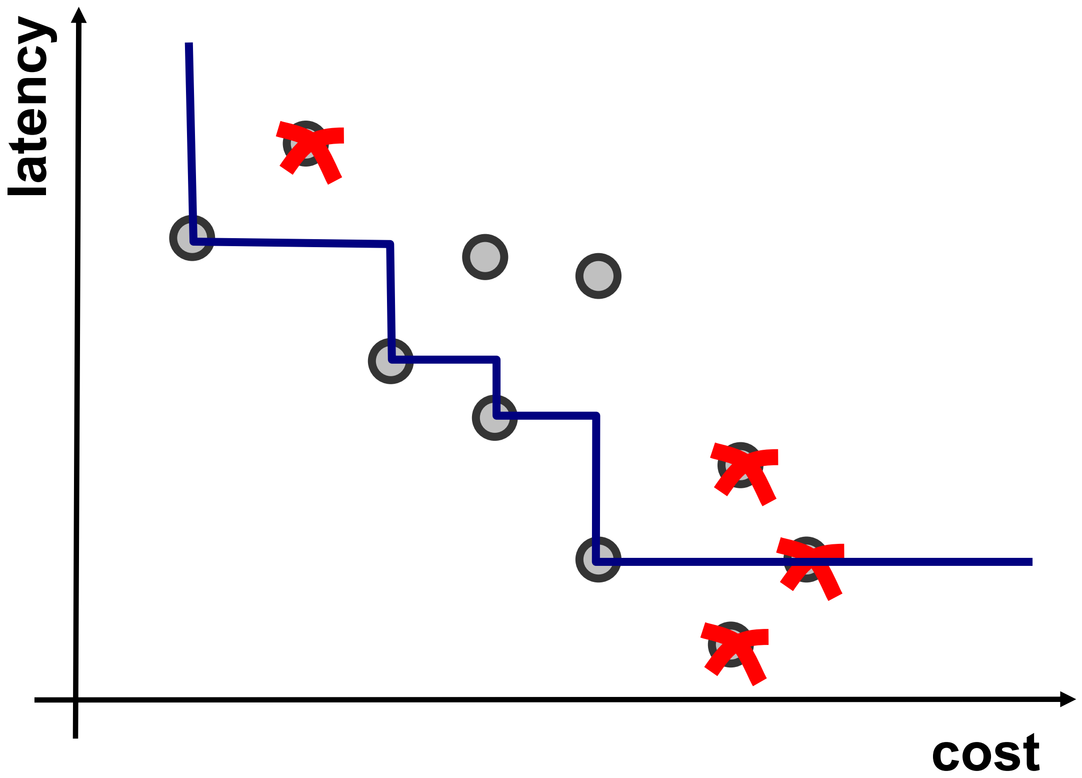
Pareto Dominance

- Without loss of generality, only minimization problems are assumed in the following



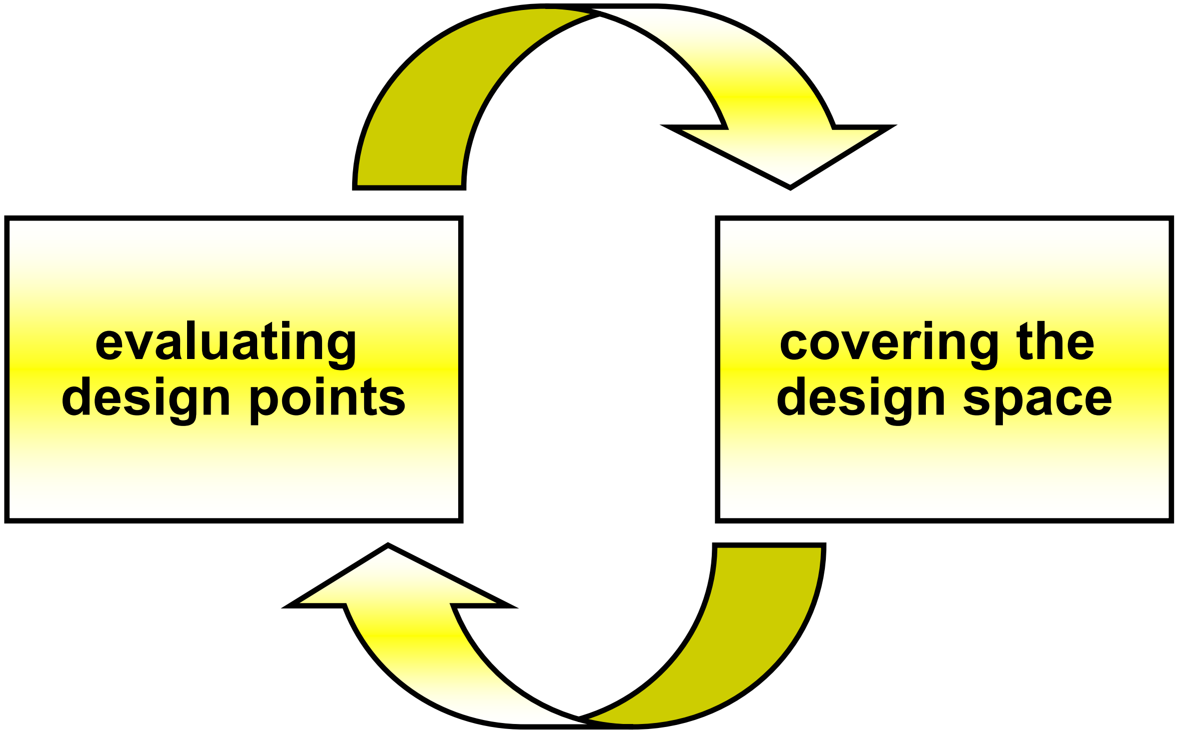
Optimization Goals

- find Pareto-optimal solutions
- or a good approximation (convergence, diversity)
- with a minimal number of function calls



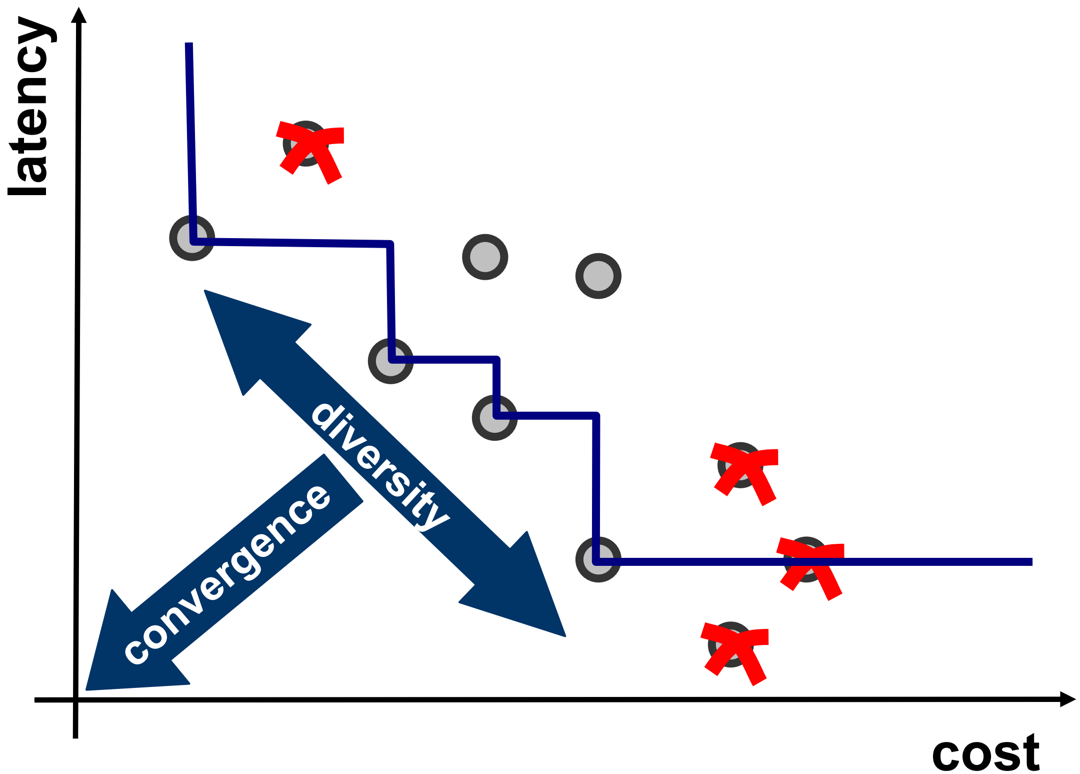
Design Space Exploration

- **Design Space Exploration is a twofold task:**
  - **How can a single design point be evaluated?**
  - **How can the design space be covered during the exploration process**

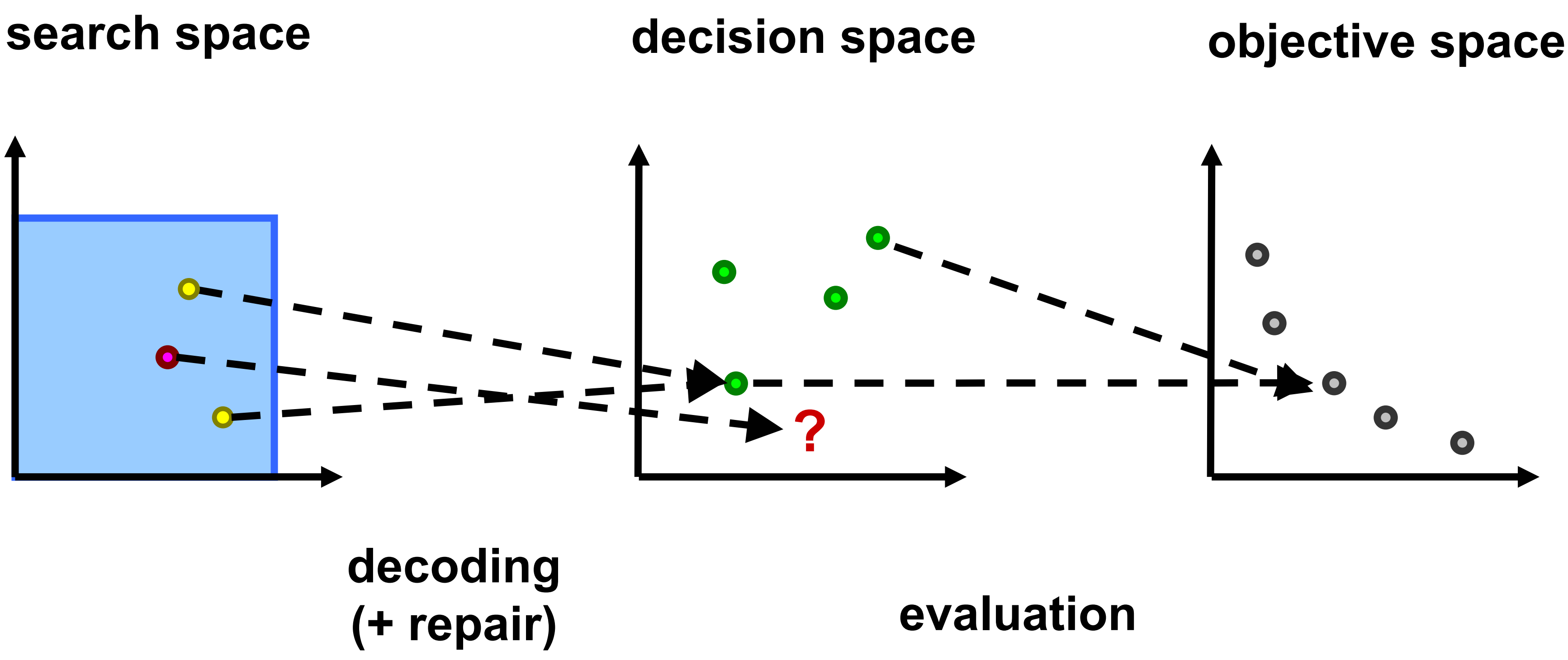


Multi-Objective Evolutionary Algorithms

- Many design points are explored in parallel
- Recombination (Mutation, Crossover) tries to improve already good solutions
- Requires appropriate problem encoding



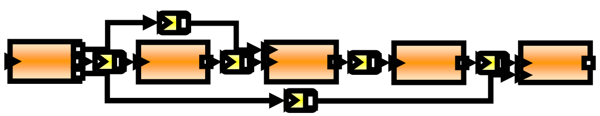
Problem Statement



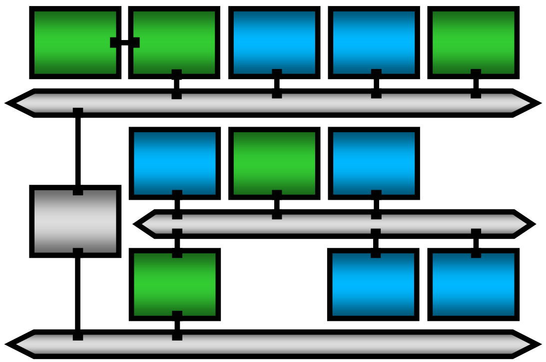
Outline

- Problem Formulation
- The Decision Problem
- Symbolic Representation
- SAT Decoding
- SMT Decoding
- Summary

Platform-Based System Synthesis

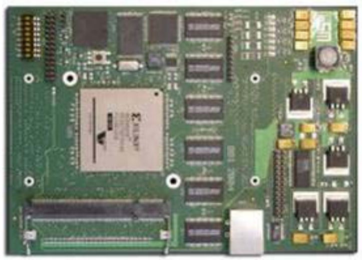


Application



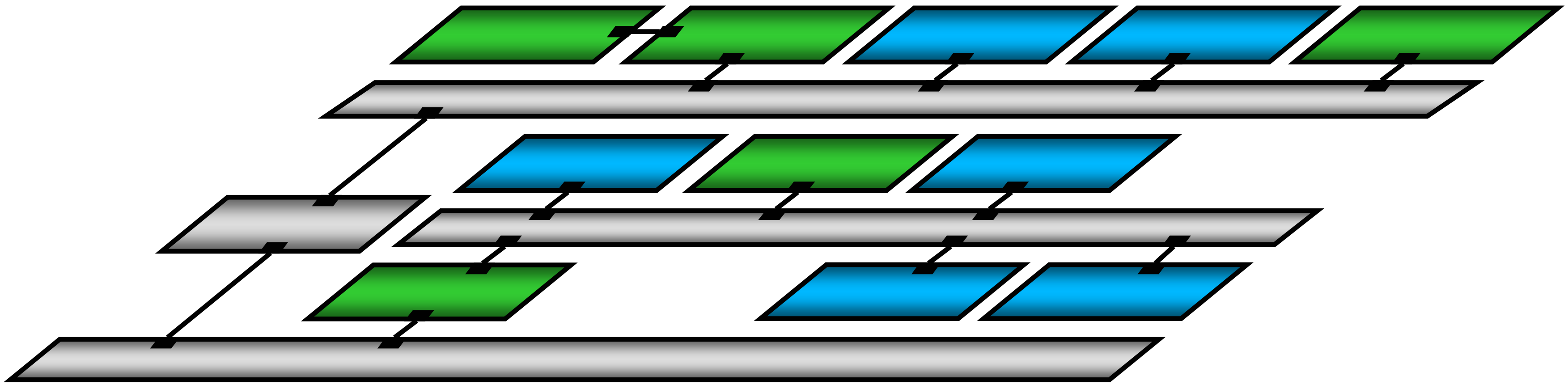
Platform

System Synthesis



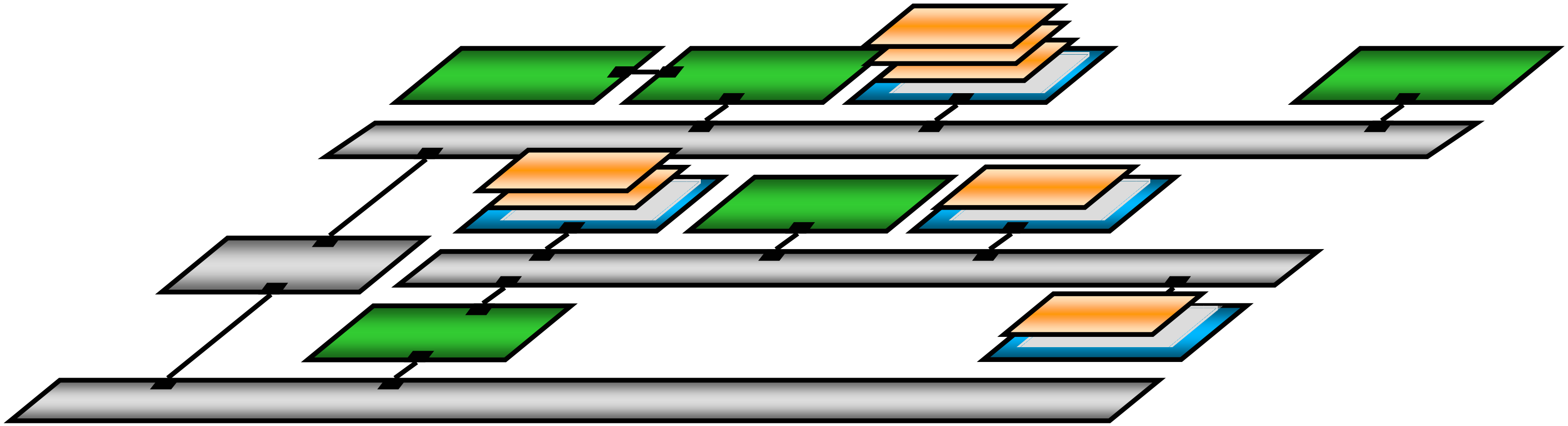
Resource Allocation

- Resource allocation, i.e., select resources from a platform for implementing the application



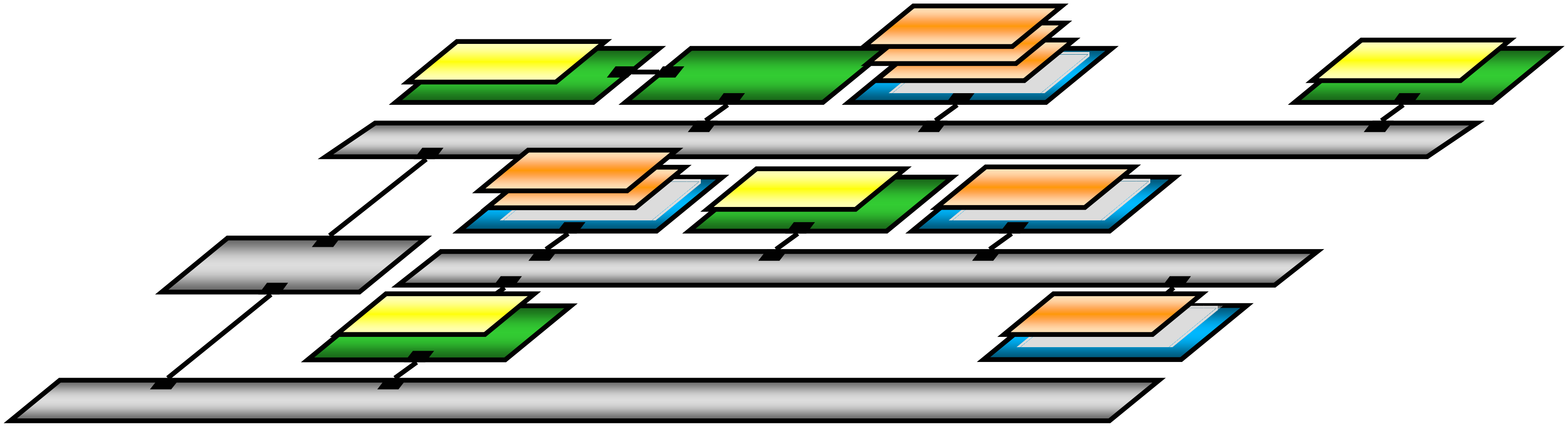
Process Binding

- Process binding, i.e., bind processes onto allocated computational resources
- Each process has to be bound exactly once



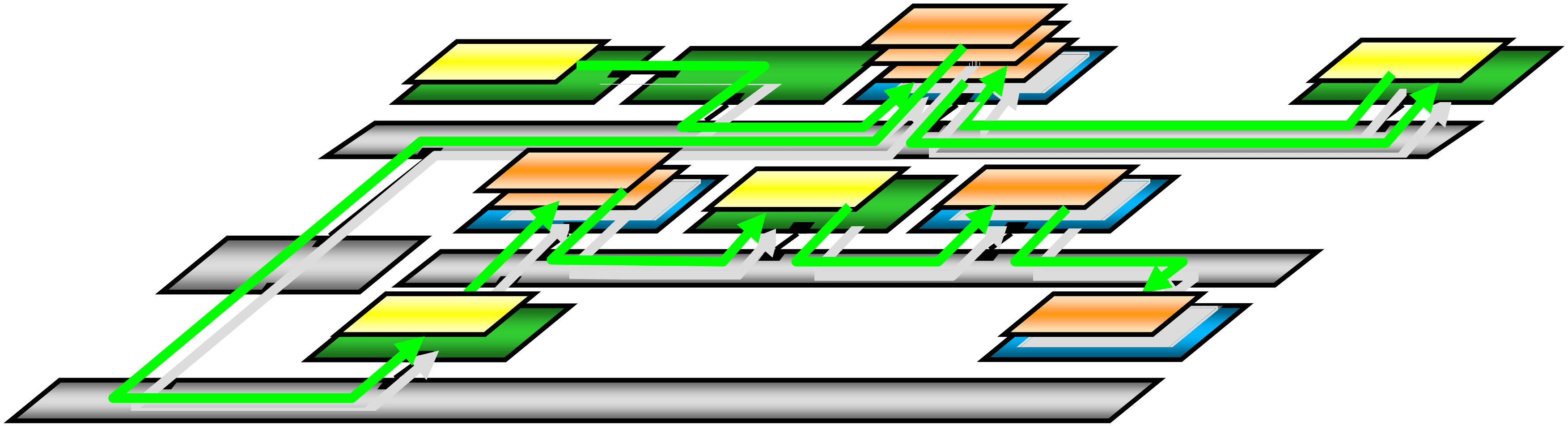
Channel Mapping

- Channel mapping, i.e., assign channels to address spaces
- Each channel has to be mapped exactly once

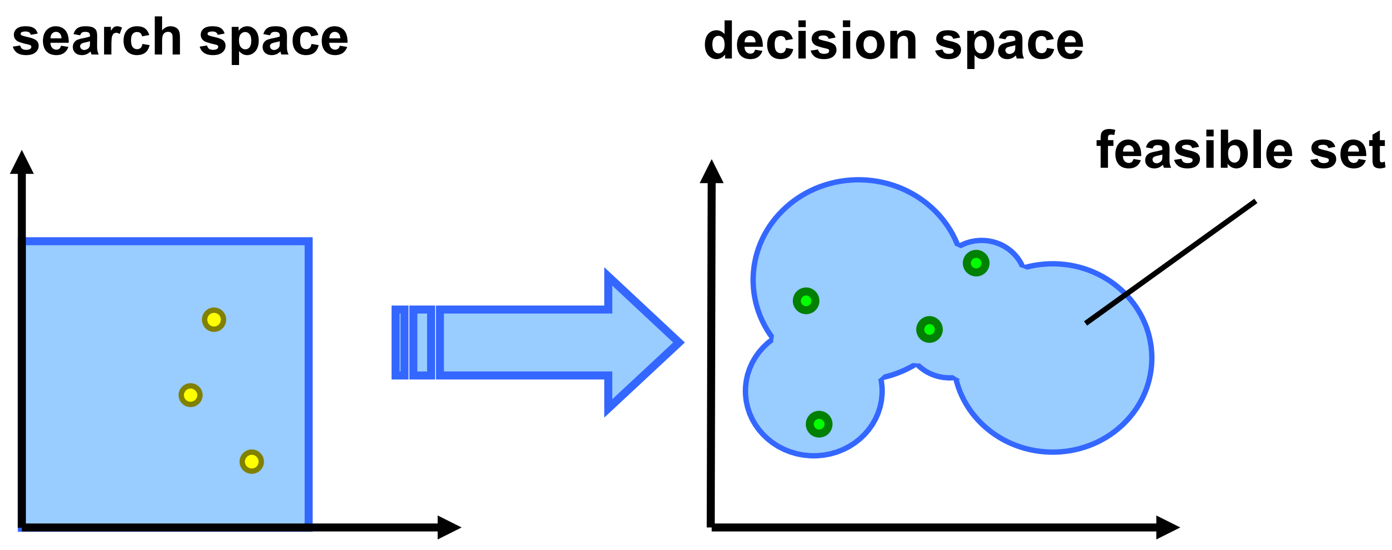


Transaction Routing

- Transaction routing, i.e., compute paths over allocated resources for all memory accesses
- Transactions, which cannot be routed, lead to infeasible solutions



Problem Statement



feasibility preserving decoding?

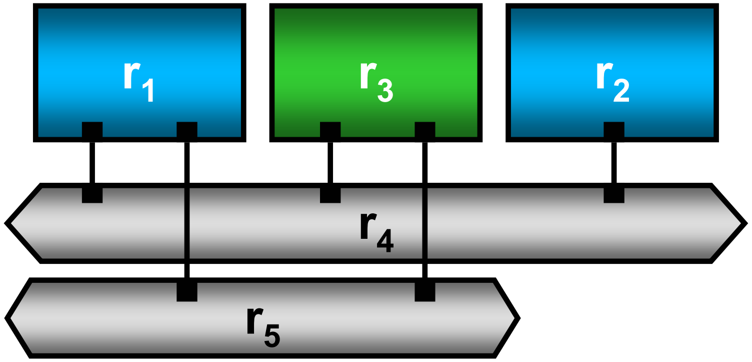
How to represent feasible set?

Outline

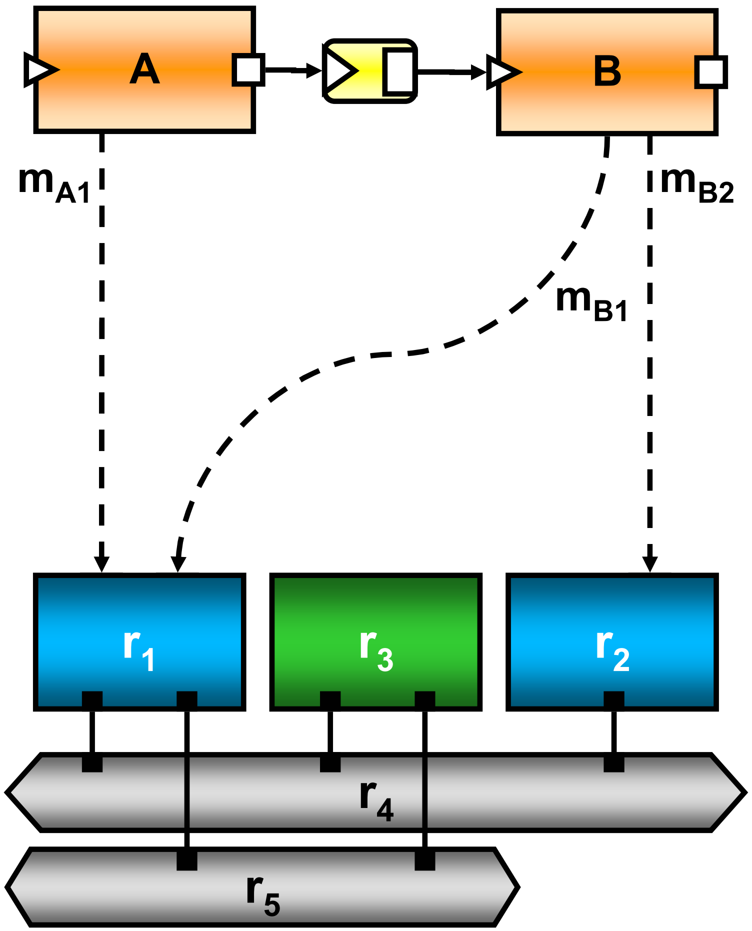
- Problem Formulation
- The Decision Problem
- Symbolic Representation
- SAT Decoding
- SMT Decoding
- Summary

Formalizing the Constraints: Allocation

$$\varphi = (r_1 \vee r_2 \vee r_3 \vee r_4 \vee r_5)$$



Formalizing the Constraints: Binding

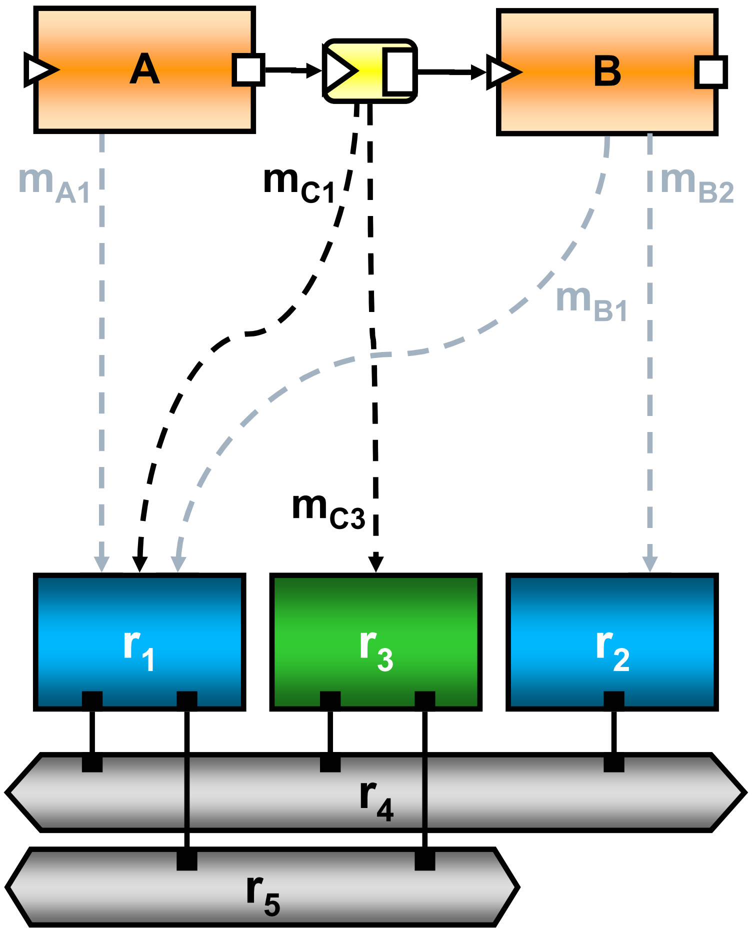


$$\varphi = (r_1 \vee r_2 \vee r_3 \vee r_4 \vee r_5)$$

$$\wedge\ (m_{A1} \wedge (m_{A1} \rightarrow r_1))$$

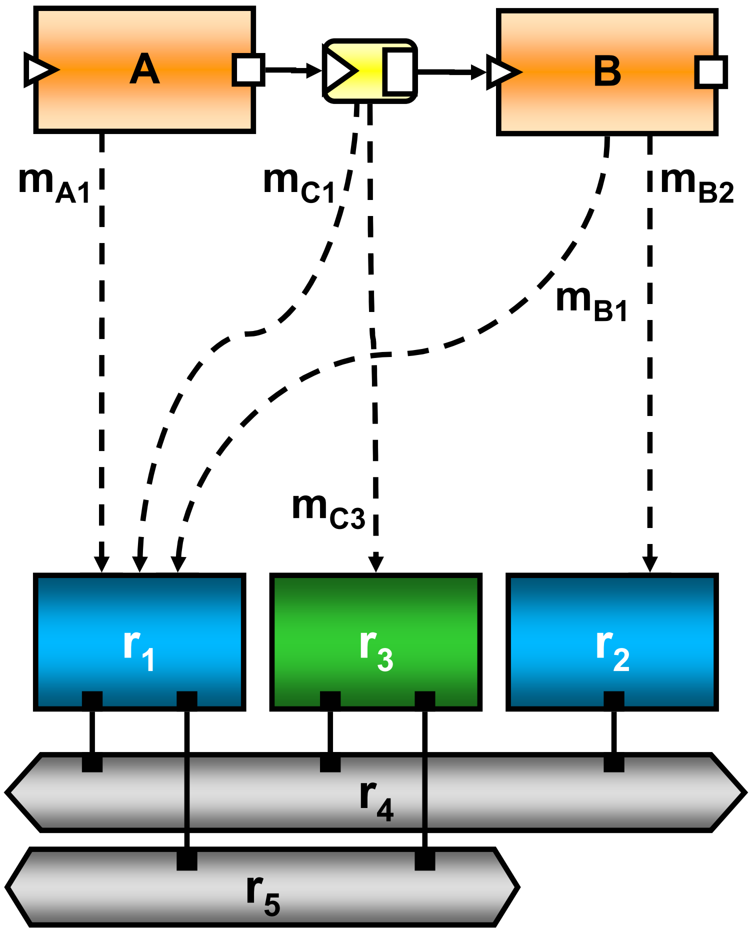
$$\wedge\ ((m_{B1} \vee m_{B2}) \wedge \overline{(m_{B1} \wedge m_{B2})}$$
$$\wedge\ (m_{B1} \rightarrow r_1) \wedge (m_{B2} \rightarrow r_2))$$

Formalizing the Constraints: Mapping



$$\varphi = (r_1 \vee r_2 \vee r_3 \vee r_4 \vee r_5)$$
$$\wedge \, (m_{A1} \wedge (m_{A1} \rightarrow r_1) \,)$$
$$\wedge \, (\, (m_{B1} \vee m_{B2}) \wedge \overline{(m_{B1} \wedge m_{B2})}$$
$$\wedge \, (m_{B1} \rightarrow r_1) \wedge (m_{B2} \rightarrow r_2) \,)$$
$$\wedge \, (\, (m_{C1} \vee m_{C3}) \wedge \overline{(m_{C1} \wedge m_{C3})}$$
$$\wedge \, (m_{C1} \rightarrow r_1) \wedge (m_{C3} \rightarrow r_3) \,)$$

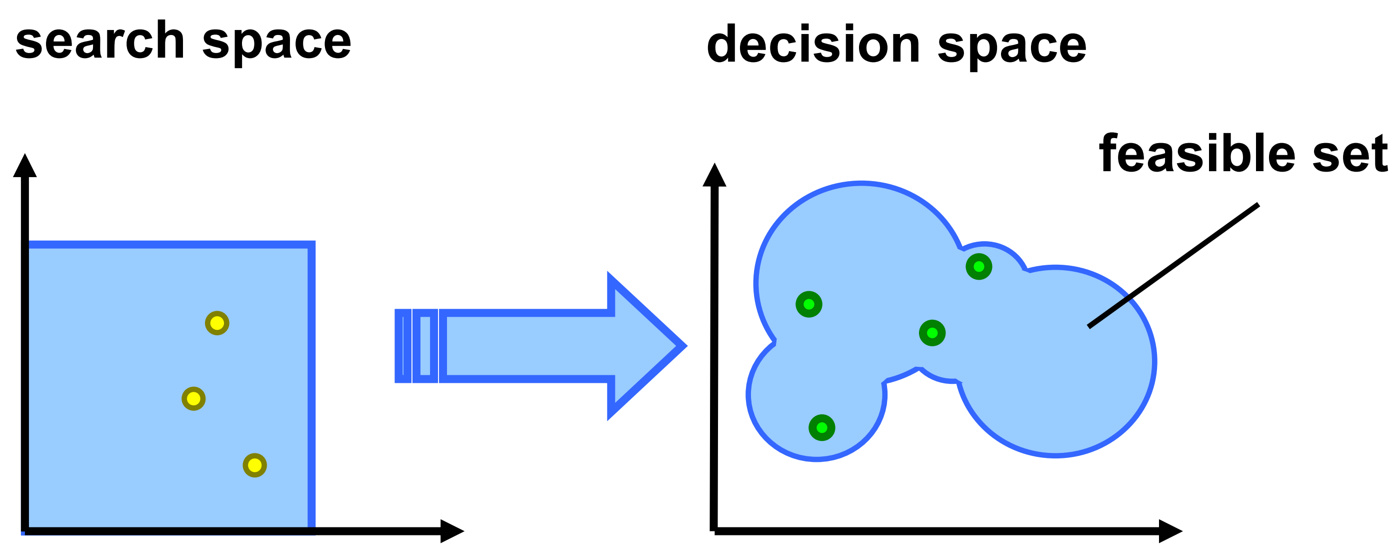
Formalizing the Constraints: Routing



$$
\begin{aligned}
\varphi = & (r_1 \vee r_2 \vee r_3 \vee r_4 \vee r_5) \\
& \wedge (m_{A1} \wedge (m_{A1} \rightarrow r_1)) \\
& \wedge ((m_{B1} \vee m_{B2}) \wedge \overline{(m_{B1} \wedge m_{B2})} \\
& \wedge (m_{B1} \rightarrow r_1) \wedge (m_{B2} \rightarrow r_2)) \\
& \wedge ((m_{C1} \vee m_{C3}) \wedge \overline{(m_{C1} \wedge m_{C3})} \\
& \wedge (m_{C1} \rightarrow r_1) \wedge (m_{C3} \rightarrow r_3)) \\
& \wedge ((m_{A1} \wedge m_{C1}) \rightarrow t_{AC1,1}) \\
& \wedge ((m_{A1} \wedge m_{C3}) \rightarrow \\
& \quad (t_{AC1,1} \wedge (t_{AC4,2} \vee t_{AC5,2}) \wedge t_{AC3,3}) \\
& \quad \wedge (t_{AC4,2} \rightarrow r_4) \wedge (t_{AC5,2} \rightarrow r_5)) \\
& \wedge ((m_{C1} \wedge m_{B1}) \rightarrow t_{CB1,1}) \\
& \wedge ((m_{C3} \wedge m_{B1}) \rightarrow \\
& \quad (t_{CB3,1} \wedge (t_{CB4,2} \vee t_{CB5,2}) \wedge t_{CB1,3}) \\
& \quad \wedge (t_{CB4,2} \rightarrow r_4) \wedge (t_{CB5,2} \rightarrow r_5)) \\
& \wedge ((m_{C3} \wedge m_{B2}) \rightarrow \\
& \quad (t_{CB3,1} \wedge (t_{CB4,2} \vee t_{CB5,2}) \wedge t_{CB2,3}) \\
& \quad \wedge (t_{CB4,2} \rightarrow r_4) \wedge (t_{CB5,2} \rightarrow r_5))
\end{aligned}
$$

Symbolic System Synthesis

- Each satisfiable variable assignment for φ represents a feasible implementation, i.e., SAT(φ)



feasibility preserving decoding?

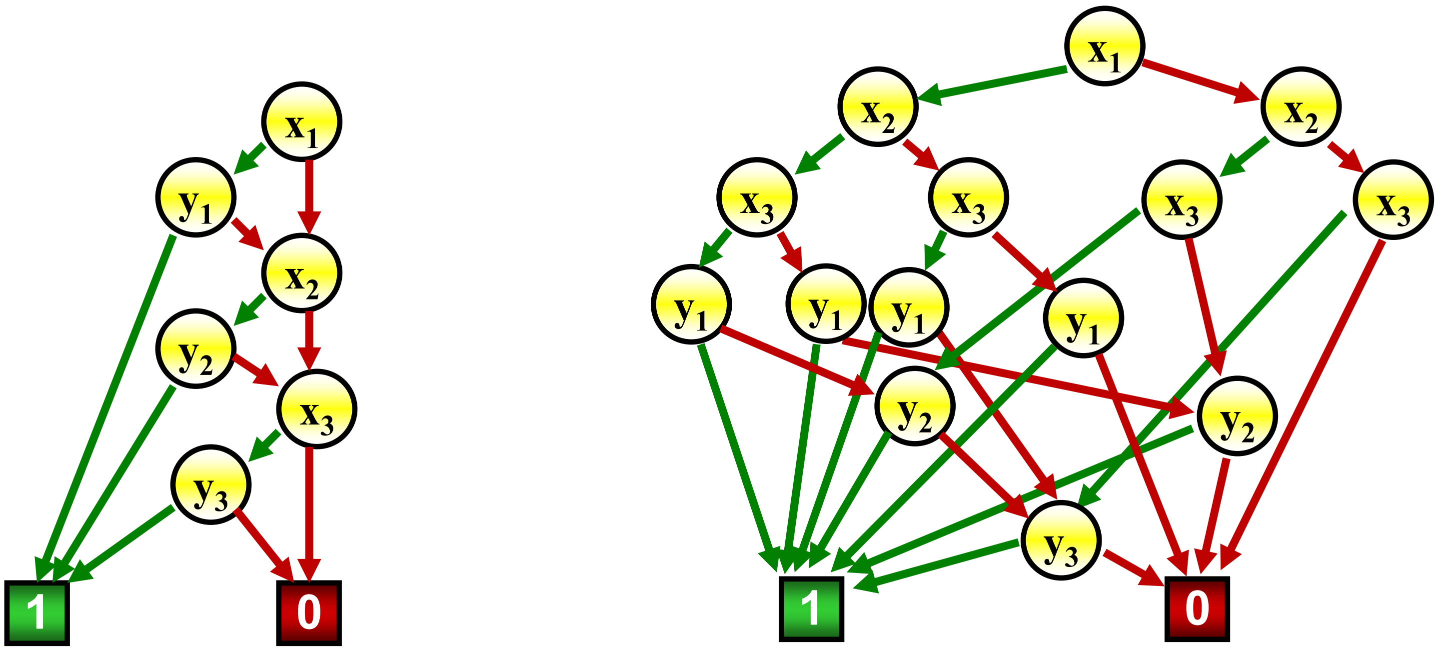
how to represent feasible set?

Outline

- Problem Formulation
- The Decision Problem
- Symbolic Representation
- **SAT Decoding**
- **SMT Decoding**
- **Summary**

Binary decision diagram (BDD)

- **Properties:**
  - **Test for satisfiability can be done in $O(1)$**
  - **The variable order can influence the size of a BDD**
  - **Worst case: exponential complexity!**
- **Example:**
  - $f(x_1, x_2, x_3, y_1, y_2, y_3) = (x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee (x_3 \wedge y_3)$



SAT (Boolean Satisfiability)

- **Does there exists at least one $x \in X$ such that $\varphi(x) = 1$?**
  - **$\varphi : x \in X \rightarrow \{0,1\}$ with $X = \{0,1\}^n$**
  - ***Boolean Satisfiability* is *NP*-complete**
  - **The Function φ is given in conjunctive normal form (CNF)**

$$\underbrace{(x_1 \lor \overline{x_2} \lor x_3)}_{\text{„clause“}} \land (\overline{x_1} \lor \underbrace{\overline{x_6}}_{\text{„literal“}}) \land \ldots$$

- ***SAT solvers:* Programs designed for efficiently solving the Satisfiability Problem**

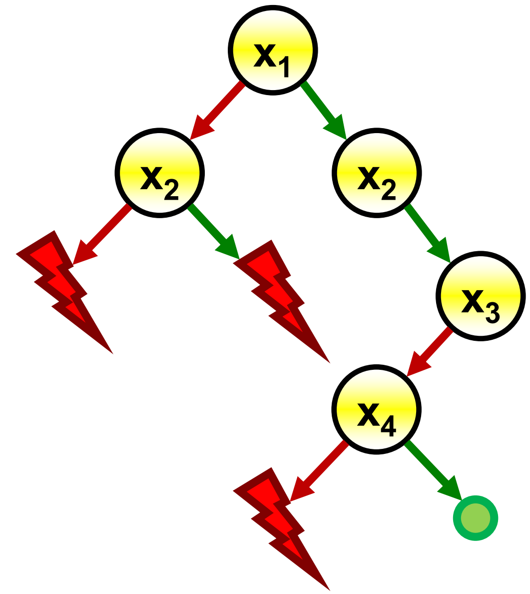
SAT-based System Synthesis

$$\varphi(x_1, x_2, x_3, x_4) = (x_1 \lor x_2) \land (x_1 \lor \overline{x_2}) \land (\overline{x_1} \lor x_2) \land (\overline{x_1} \lor \overline{x_2} \lor \overline{x_3} \lor \overline{x_4}) \land (\overline{x_1} \lor \overline{x_2} \lor x_3 \lor x_4)$$

while ***true*** **do**
  branch(ρ,σ)
  if ***CONFLICT(***  ***)*** **then**
    BACKTRACK()
  else if ***SATISFIED(***  ***)*** **then**
    return ***x***
  end if
end while

$\rho = (x_1, x_2, x_3, x_4)$

$\sigma = (0, 1, 0, 0)$



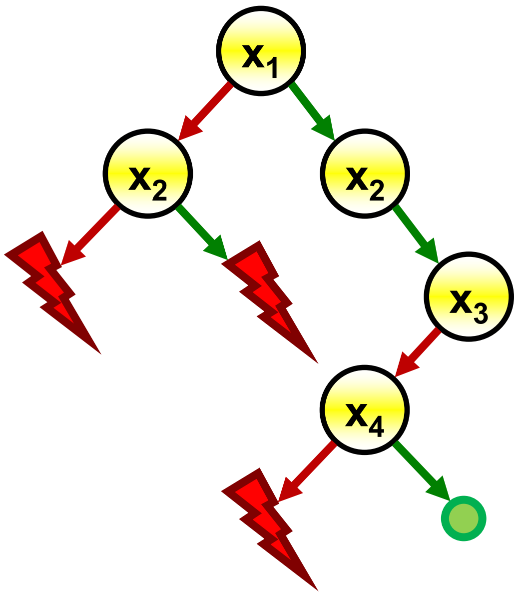
$X = (1, 1, 0, 1)$

SAT-Solver (Branching)

- Different decision strategies lead to different solutions

$\rho = (x_1, x_2, x_3, x_4)$

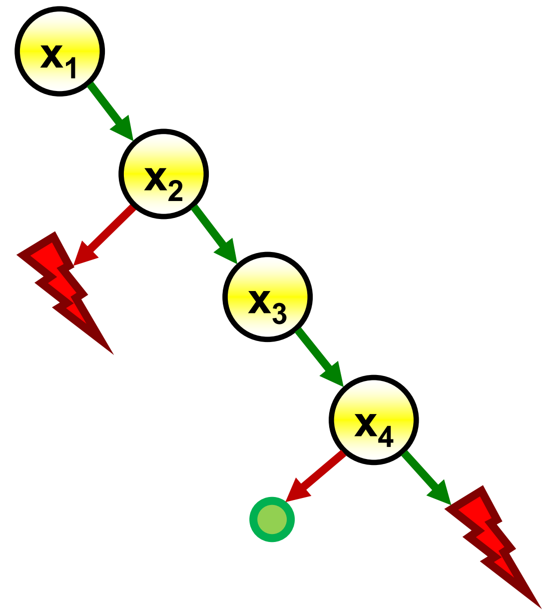
$\sigma = (0, 1, 0, 0)$



X = (1, 1, 0, 1)

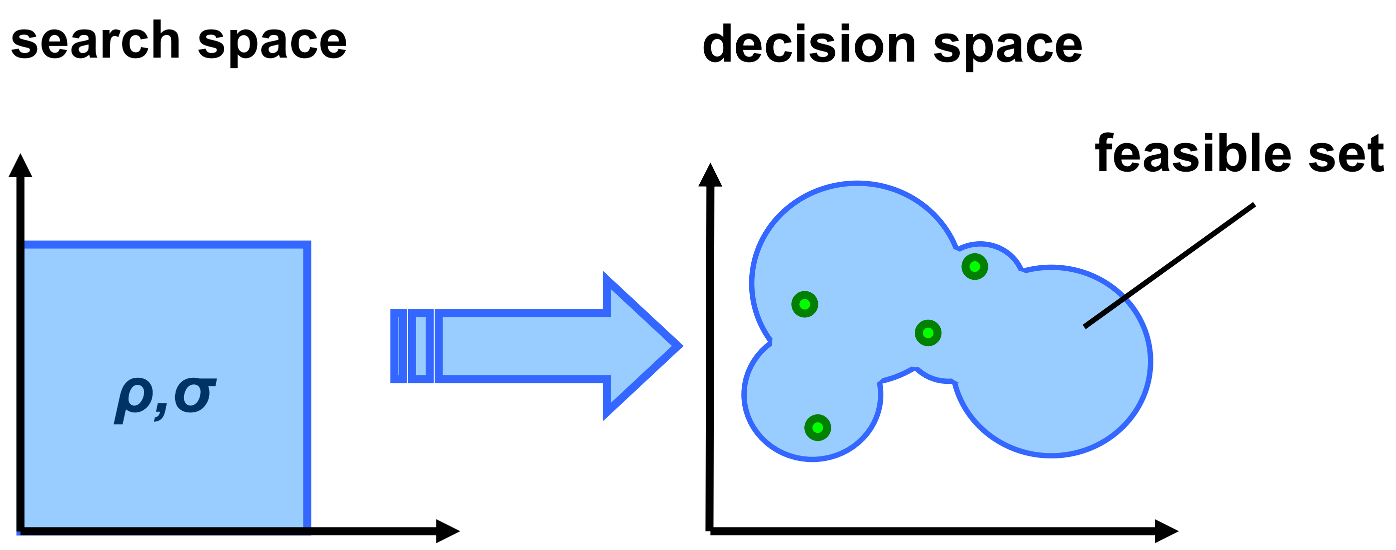
$\rho = (x_1, x_2, x_3, x_4)$

$\sigma = (1, 0, 1, 1)$



X = (1, 1, 1, 0)

Symbolic System Synthesis

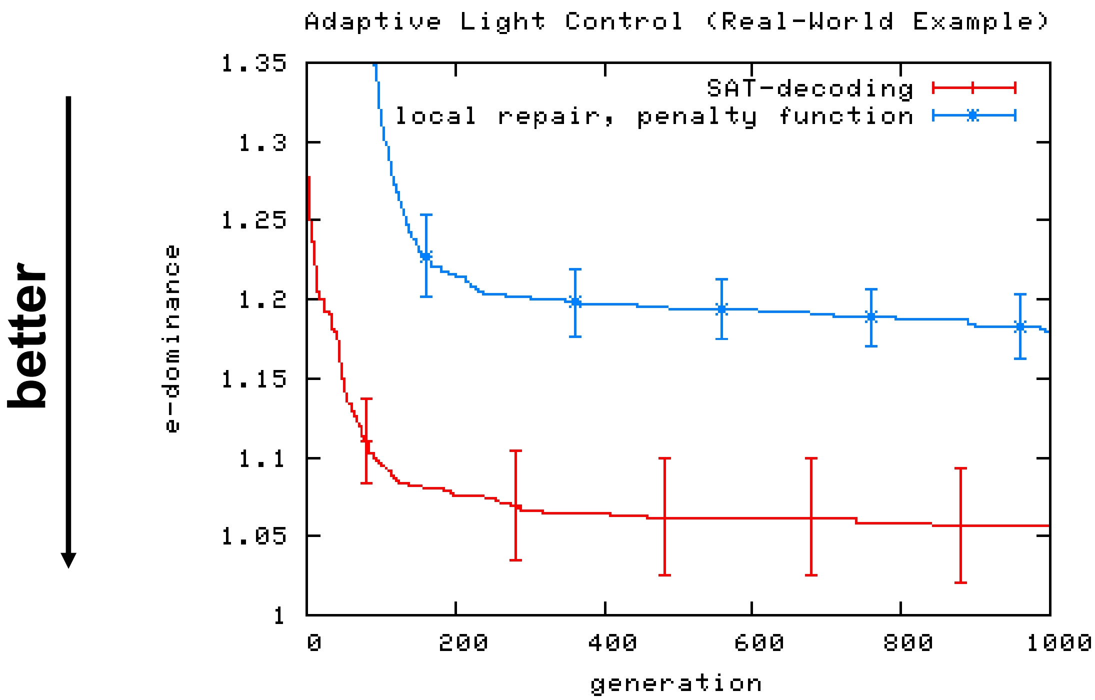


feasibility preserving decoding?

how to represent feasible set?

Results SAT Decoding

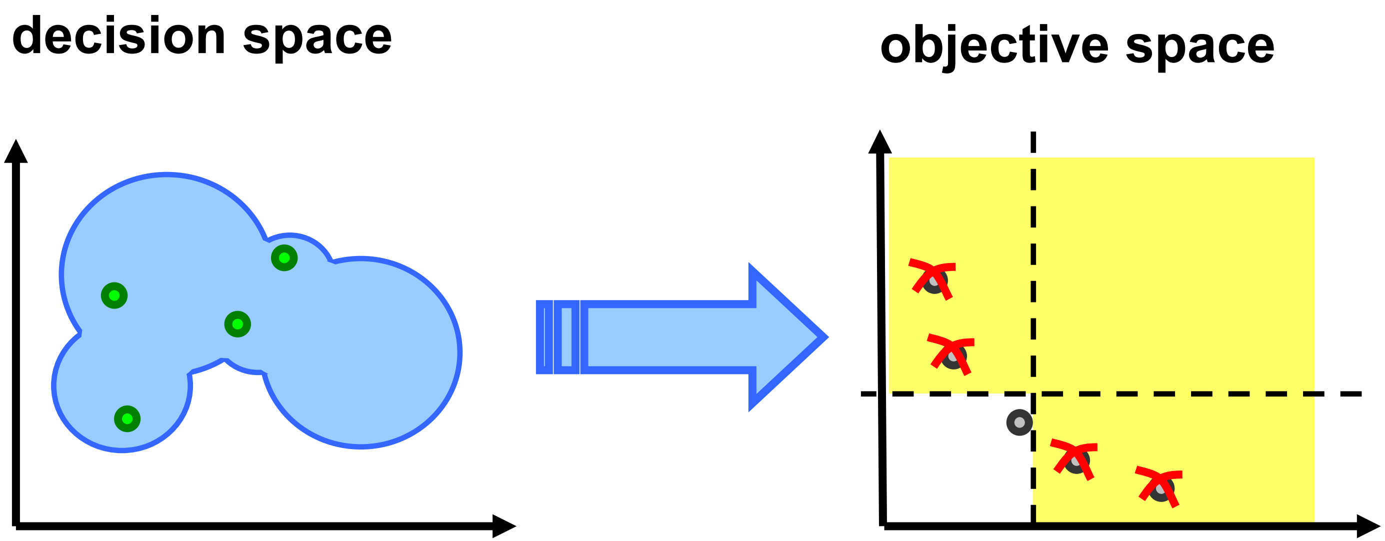
- Real-world problem: Automotive Application
  - To find a single feasible solution is NP-complete
  - Runtime over 1000 Generations nearly the same for both methods!
  - SAT Decoding is superior in quality of the results!



Outline

- Problem Formulation
- The Decision Problem
- Symbolic Representation
- SAT Decoding
- **SMT Decoding**
- **Summary**

Symbolic System Synthesis



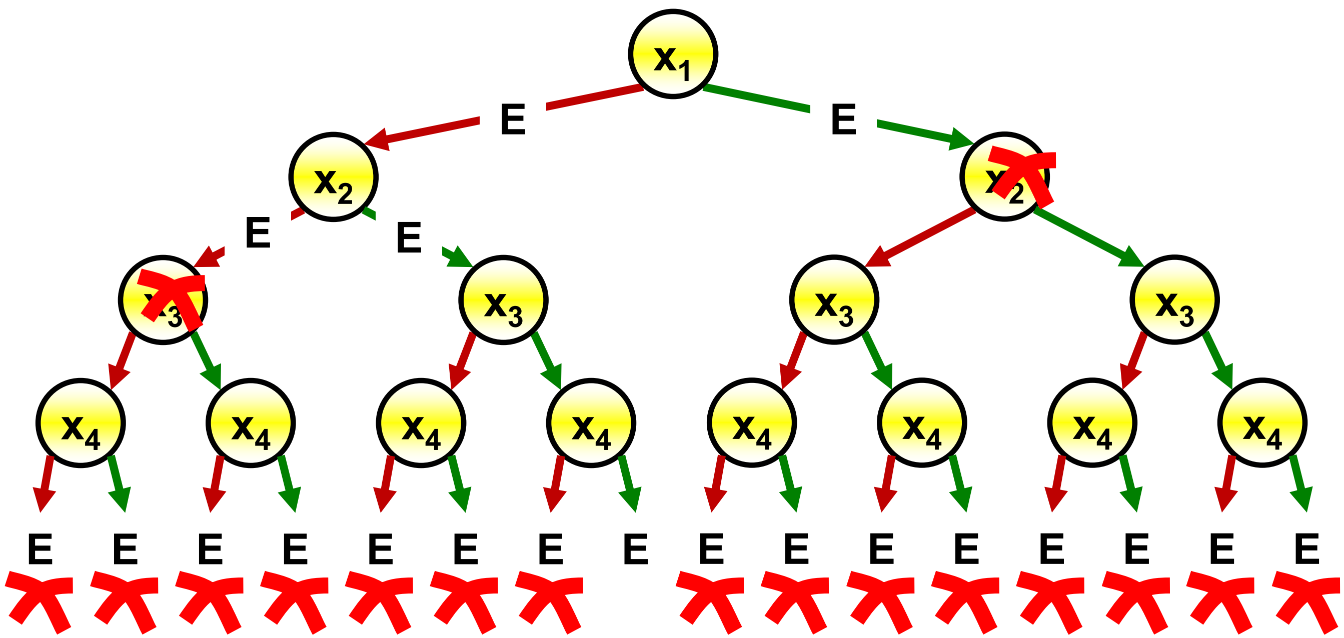
feasibility preserving decoding?

how to represent feasible set?

how to handle stringent constraints efficiently?

Idea

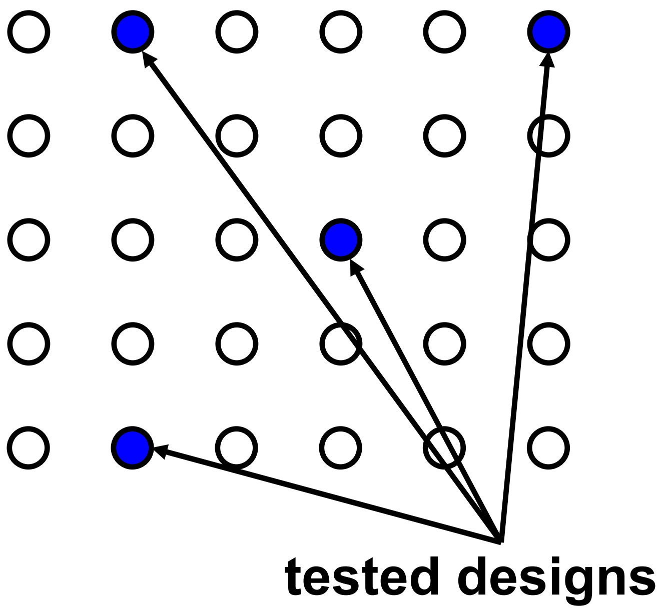
- Early Constraint Checking



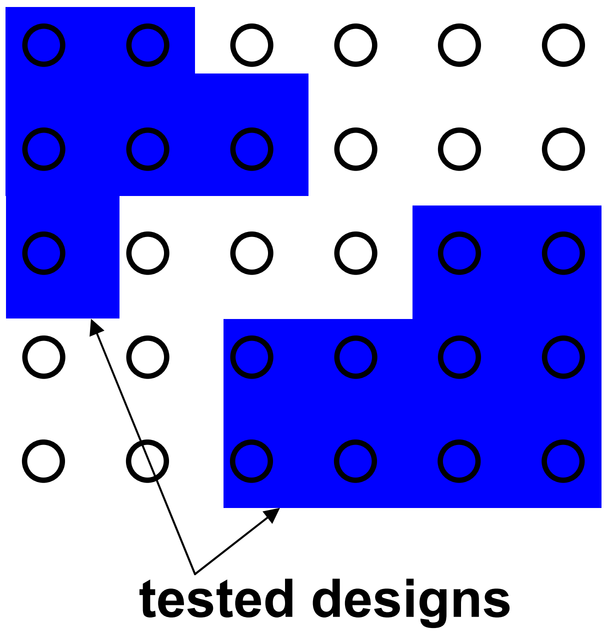
- Early Constraint Checking + Learning in Boolean Formula = Satisfiability Modulo Theories (SMT) Solving
- In other words: SAT problem is solved with respect to given background theories

SMT Decoding

SAT Decoding with subsequent constraint checking



SMT Decoding



- Requires monotonous constraint checking functions
- Even methods for checking non-linear constraints can be incorporated as background theory
- First results will be presented at ESWEEK 2010

Outline

- Problem Formulation
- The Decision Problem
- Symbolic Representation
- SAT Decoding
- SMT Decoding
- **Summary**

Summary

- Design space exploration requires for repeatedly solving the system synthesis problem
- Feasibility preserving decoding is particularly useful in design spaces with small and complex feasible region
- Encoding the feasible region by Boolean formulas permits use of SAT solvers
- For design spaces with stringent constraints, early constraint checking together with learning in the Boolean formula significantly speeds up DSE