WARWICK BUSINESS SCHOOL

The use of surrogates in Genetic Programming

Juergen Branke, Torsten Hildebrandt

Hildebrandt, T.; Branke, J.: “On using surrogates with genetic programming”. Evolutionary Computation Journal 23(3), 2015, pp. 343-367

THE UNIVERSITY OF WARWICK

Outline

- Motivation
- Challenges
- Generation of dispatching rules
- Phenotypic distance measures
- Empirical results
- Conclusion

Motivation

- Evaluating a single solution can be computationally very expensive
- Evaluating a solution can be costly
- Evaluating a solution can be dangerous
- Evaluating a solution may require user interaction

→ Number of fitness evaluations is limited

Solution

- Learn surrogate fitness model
- Use surrogate models to estimate fitness of solutions
- Discard some solutions without evaluating their fitness

Surrogate assisted evolutionary algorithms

1. Initialize population
2. Evaluate population
3. Train surrogate model(s)
4. Create offspring
5. Estimate fitness of offspring based on surrogate
6. Decide which solutions to evaluate
7. Update surrogate model(s)
8. Merge offspring and parent population
9. Go to 4.

Challenges

- Which solutions to evaluate
  - Promising solutions
  - Solutions where surrogate model is uncertain
  - Solutions that improve accuracy of surrogate model
- What model(s) to use
  - Gaussian Processes
  - Artificial Neural Networks
  - Regression
  - All models require a distance metric

Challenges in combination with GP

- GP typically uses a tree representation
- Not clear how to define distance between trees

Genotypic distance

$$\mathrm{SHD}(T_1, T_2) = \begin{cases} 1 & \text{if } arity(p) \neq arity(q) \\ hd(p,q) & \text{if } arity(p) = arity(q) = 0 \\ \frac{1}{m+1}\left(hd(p,q) + \sum_{i=1}^{m} \mathrm{SHD}(s_i, t_i)\right) & \\ & \text{if } arity(p) = arity(q) = m \end{cases}$$

- p, q: root nodes
- Si, ti: i-th subtree of p, q
- HD: Hamming distance, 0 if same terminal/non-terminal

[Moraglio and Poli 2005]

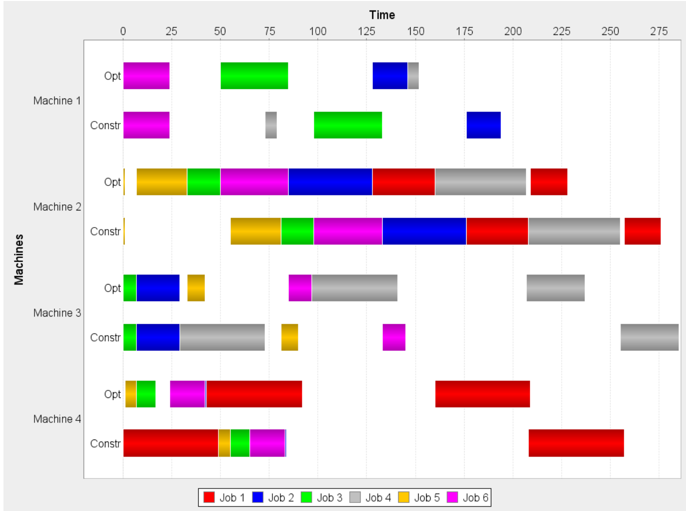
Challenges in combination with GP

- GP typically uses a tree representation
- Not clear how to define distance between trees
- Different trees can encode the same solution
  - Permutations
  - Equal meaning
  - Bloat

Idea: Phenotypic distance [Hildebrandt & Branke 2015]

- Distance not between genotypes (trees) but between behaviour
- Problem specific

Scheduling



- What job to produce
  when on which machine

- Omnipresent in manufacturing
- Large impact on cost
- Very complex (NP hard)

➡ A lot of research has gone into scheduling

Real world challenges

- Most environments are dynamic
  - New jobs arriving over time
- Most environments are stochastic
  - Stochastic processing times
  - Machine failures
  - Stochastic rework

➡ Repeated re-scheduling

➡ Dispatching rules

Job shop scheduling

- Jobs consist of an ordered sequence of operations
- Each operation takes a certain time processing on a certain machine
- Order of machines can be different for each job
- A machine can process only one operation at a time
- Operations can not be interrupted
- Objectives: Minimize tardiness or mean flow time

Dispatching rules / Self-organization

- No global schedule generated
- Decision rule to determine next action whenever a machine becomes idle
- Popular examples: FIFO, SPT, EDD

Advantages:

- Always take latest information into account
- Easy to implement and to compute

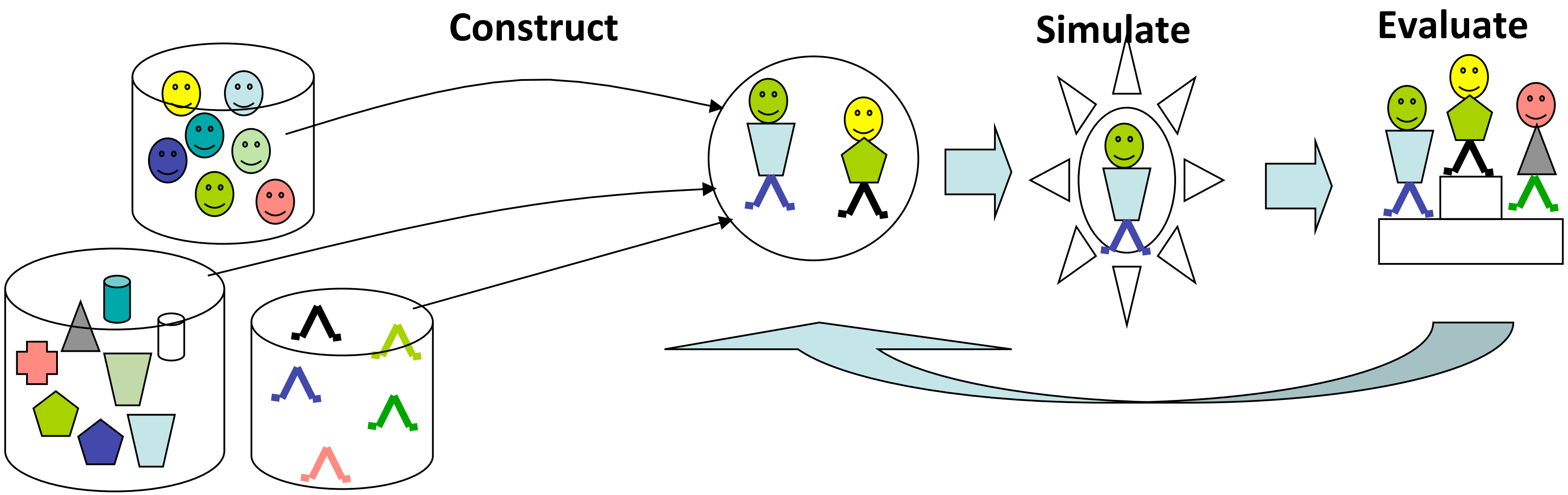
Design challenge

- Dispatching rules are based on local information
- Performance is measured globally

➡ How to design local dispatching rules to achieve best possible global performance?
  - Which attributes?
  - How combined?

Simulation-based design

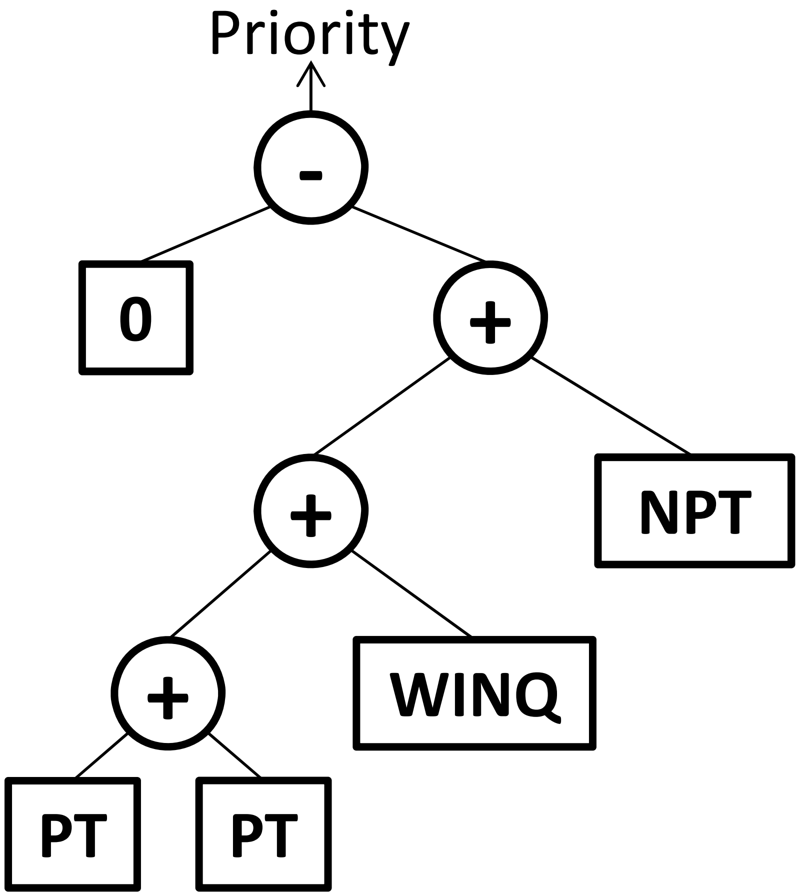
- Construction of several alternatives
- Simulation to evaluate the alternatives
- Attempt to find a better solution



Automatic generation of dispatching rules [Branke et al. 2010]

- Genetic Programming can generate Lisp expressions
- Evaluation of a dispatching rule via stochastic simulation

NPT+WINQ+2*PT



Challenges

- Simulation computationally expensive
  - Parallel execution on machine with 8 processors
  - Runtime: ca. 7 hours
- Stochastic simulation
  - Typical approaches of averaging over space or averaging over multiple runs doesn‘t work
  - Equal seed within a generation
  - Store best solutions of each iteration
  - Clean-up after optimisation with OCBA
- Trade-off: Quality and complexity of rule
  - Multicriteria approach

Application

Benchmark from

semiconductor manufacturing (MASM)

- 31 machine groups, some with parallel machines
- Batch machines
- Some machines with setup times
- 2 product categories, 92 and 19 operations
- Minimise weighted tardiness

Terminals

- Processing time
- Processing time on next machine
- Number of operations remaining
- Remaining processing time
- Work in next queue
- Time in queue
- Time in system
- Slack
- Time until deadline
- Weight
- Setup time
- Number of compatible jobs for batching

Results

- Rule of length 9: w/max(L,P)-s+b
- Rule of length 98:

$$\text{ifte}(\max(1,r)-\max(1,r,L),w,b)*b*\max(r/L+\max(-\text{ifte}(b-L,w,b)+s+b,S+b*\text{ifte}(\max(1,r)-\max(L,d),w,b)-s-\max(1,r,L)+\max(1,r)+1)*\text{ifte}(b-L,w,b)-s,S+b*\text{ifte}(\max(1,r)-L,w,b)*(2*r/L-s)+r/L-s+1)$$

Results (2)

Comparison with best rules from literature

Util 93.8%; Product mix 30/70

Rule	WeightedTardiness
ATCS/MBS(5)	2336
GP9	1669
GP98	782
GP199	696

Util 85%; Product mix 30/70

Rule	WeightedTardiness
ATCS/MBS(4)	451
GP9	451
GP98	47
GP199	95

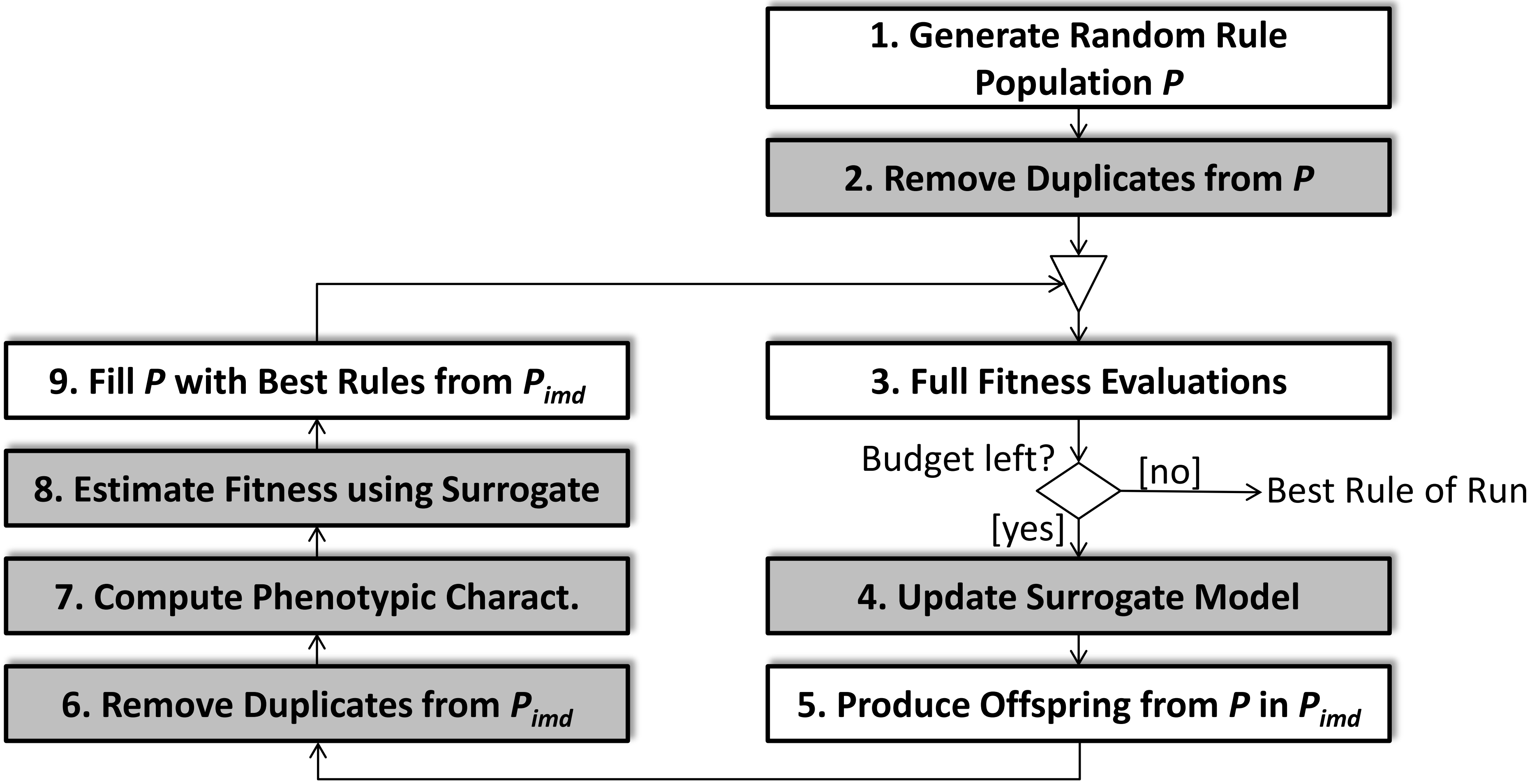
Util 85%; Product mix 70/30

Rule	WeightedTardiness
WMOD/MBS(1)	216
GP9	644
GP98	51
GP199	98

Util 93.8%; Product mix 70/30

Rule	WeightedTardiness
WMOD/MBS(3)	1245
GP9	868
GP98	206
GP199	279

Our EA



1. Generate Random Rule Population P
2. Remove Duplicates from P
3. Full Fitness Evaluations
Budget left?
[no]
Best Rule of Run
[yes]
4. Update Surrogate Model
5. Produce Offspring from P in P_imd
6. Remove Duplicates from P_imd
7. Compute Phenotypic Charact.
8. Estimate Fitness using Surrogate
9. Fill P with Best Rules from P_imd

Phenotypic characterization

decision situation	attribute set **s** s_1	s_2	s_3	ranking by reference rule	ranking by other rule	decision vector **d**
1	3	4	8	1	2	
1	7	6	15	**2**	1	**2**
2	23	17	1	2	2	
2	8	9	3	**3**	1	**3**
2	6	4	6	1	3	
⋮		⋮		⋮	⋮	⋮
k	7	3	9	2	2	
k	4	8	6	**1**	1	**1**

Database and distance function

	d_1	d_2	…	d_k	fitness
$rule_1$:	2	3	…	1	1456
$rule_2$:	1	2	…	2	1123
⋮		⋮			⋮
$rule_m$:	1	3	…	1	1293

$$D(d^A, d^B) = \sqrt{\sum_{i=1}^{k} \left(d_i^A - d_i^B\right)^2}$$

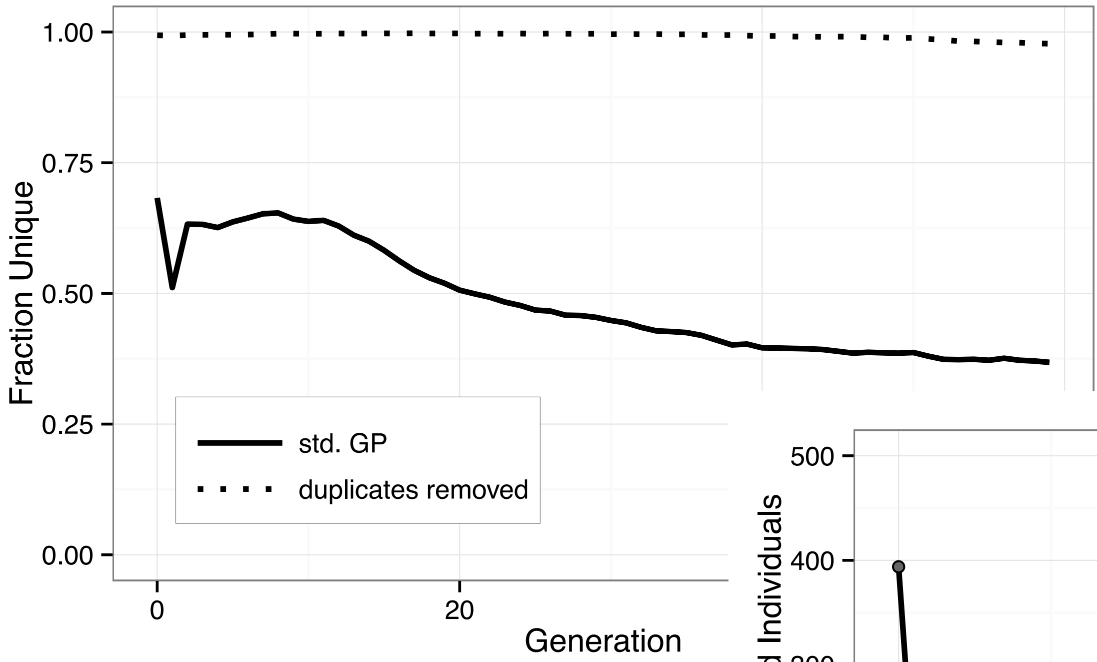
Phenotypic characterization

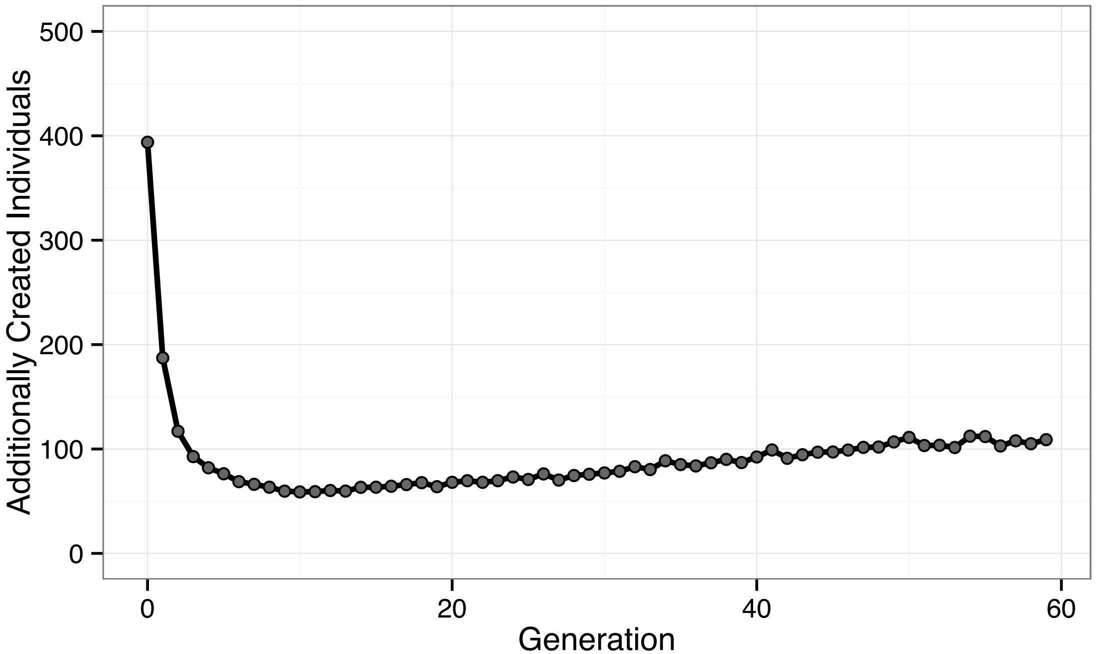
Algorithm 1 Compute the phenotypic characterization

Input: r_{new}: the dispatching rule to characterize
Input: r_{ref}: the reference rule
Input: S: set of $|S|$ decision situations
Output: d: decision vector with $|S|$ elements

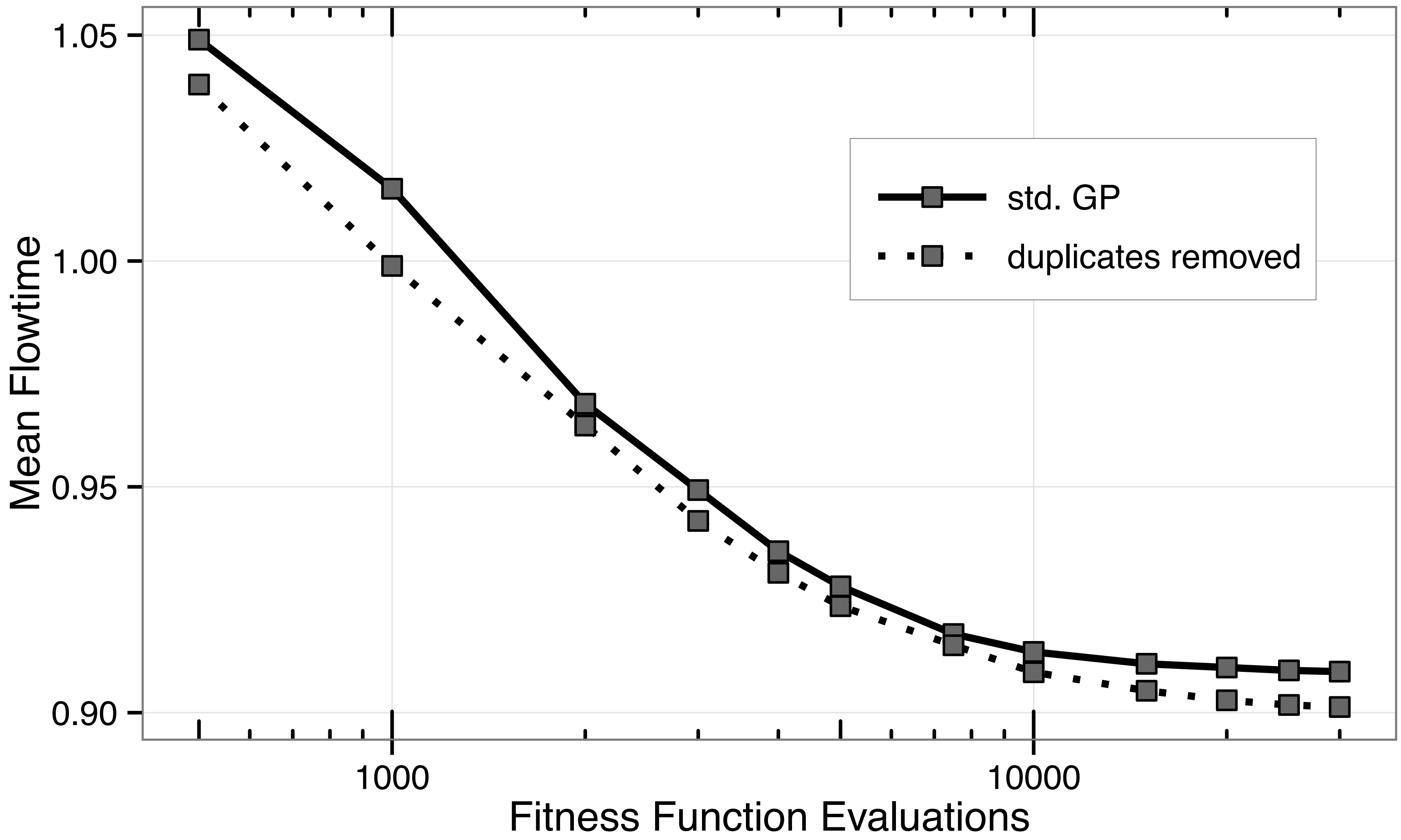
$d \leftarrow$ new integer vector with $|S|$ elements
for $i \leftarrow 1, |S|$ **do**
  $s \leftarrow S[i]$ ▷ for each decision situation $s \in S$
  $p_{\text{ref}} \leftarrow \text{apply}(r_{\text{ref}}, s)$ ▷ compute $|s|$ priorities applying r_{ref} to s
  $k_{\text{ref}} \leftarrow \text{ranks}(p_{\text{ref}})$ ▷ find ranks, highest priority gets rank 1
  $p_{\text{new}} \leftarrow \text{apply}(r_{\text{new}}, s)$
  $k_{\text{new}} \leftarrow \text{ranks}(p_{\text{new}})$
  $j \leftarrow \arg\min(k_{\text{new}})$ ▷ find index with rank 1
  $d[i] \leftarrow k_{\text{ref}}[j]$
end for
return d

Duplicate removal





Benefit of duplicate removal



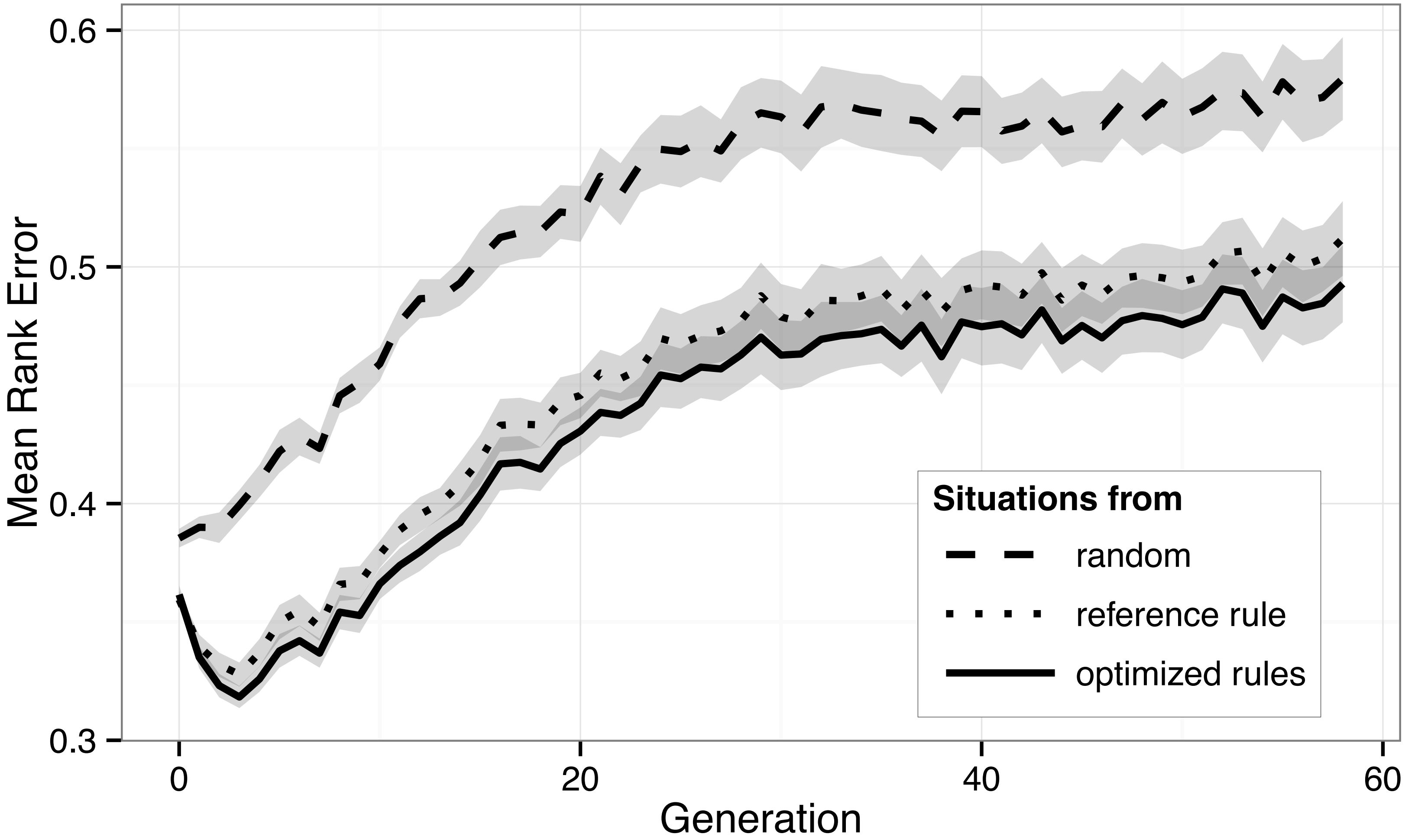
Surrogate model used

- Nearest neighbor
- Pre-selection
  - Number of offspring n times larger
  - Select top 1/n using surrogate model

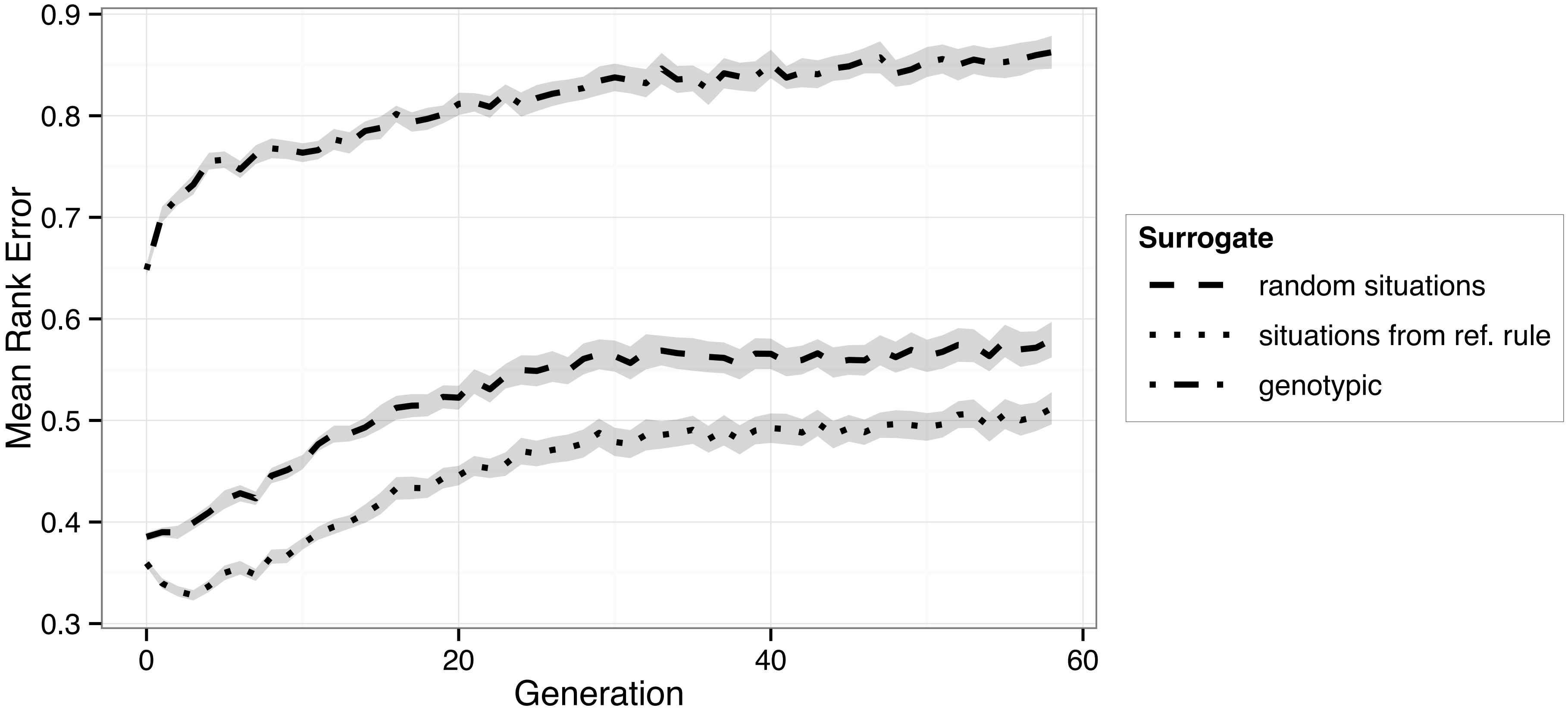
How to select “decision situations”

- *Random* based on typical value ranges, attributes independent
- *Reference rule:* From a simulation with a pre-selected simple rule (Holthaus)
- *Optimized:* From a simulation using the best found rules

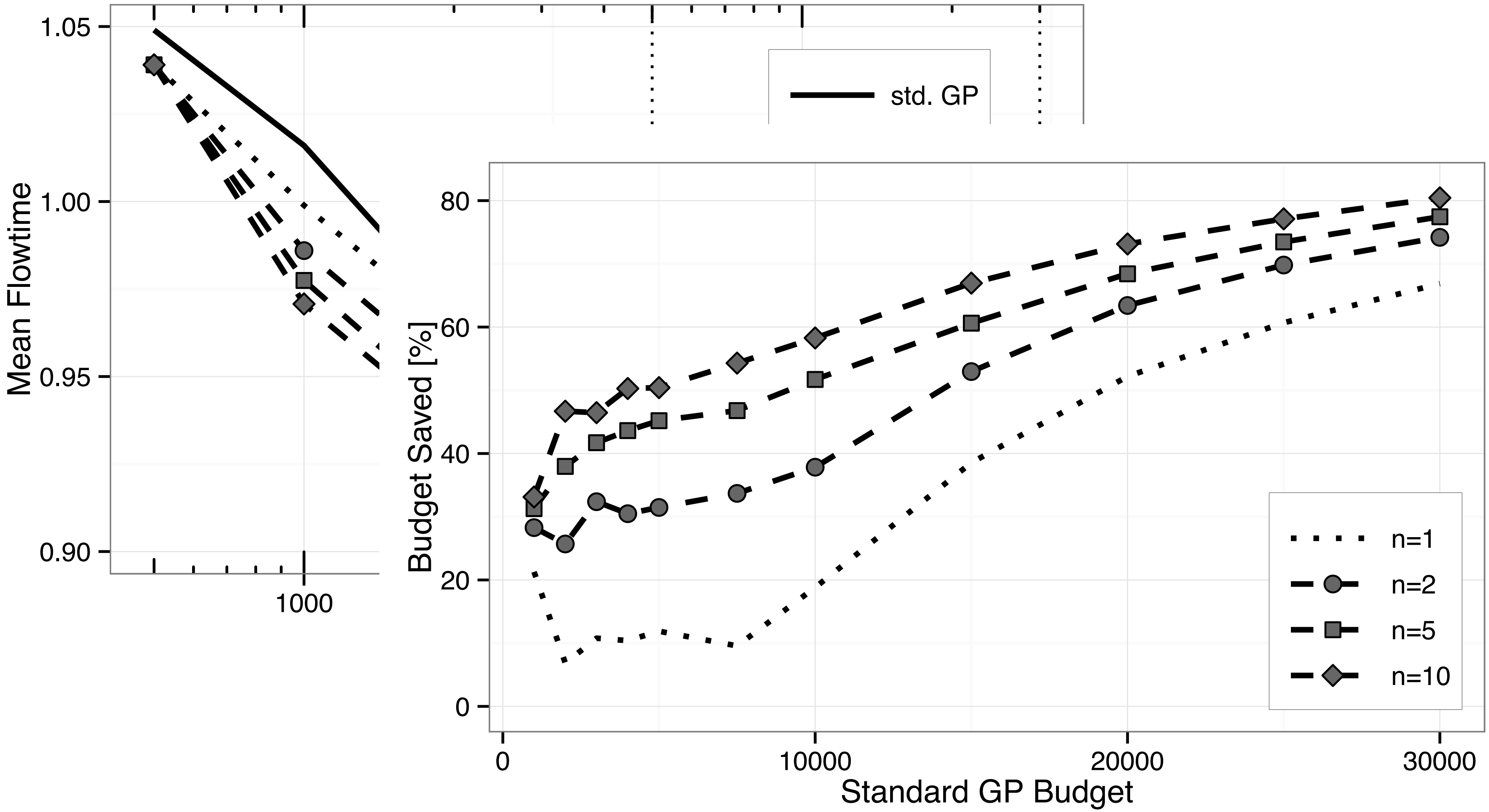
Mean rank error during optimization



Phenotypic vs. genotypic distance



Empirical performance



Relative performance difference

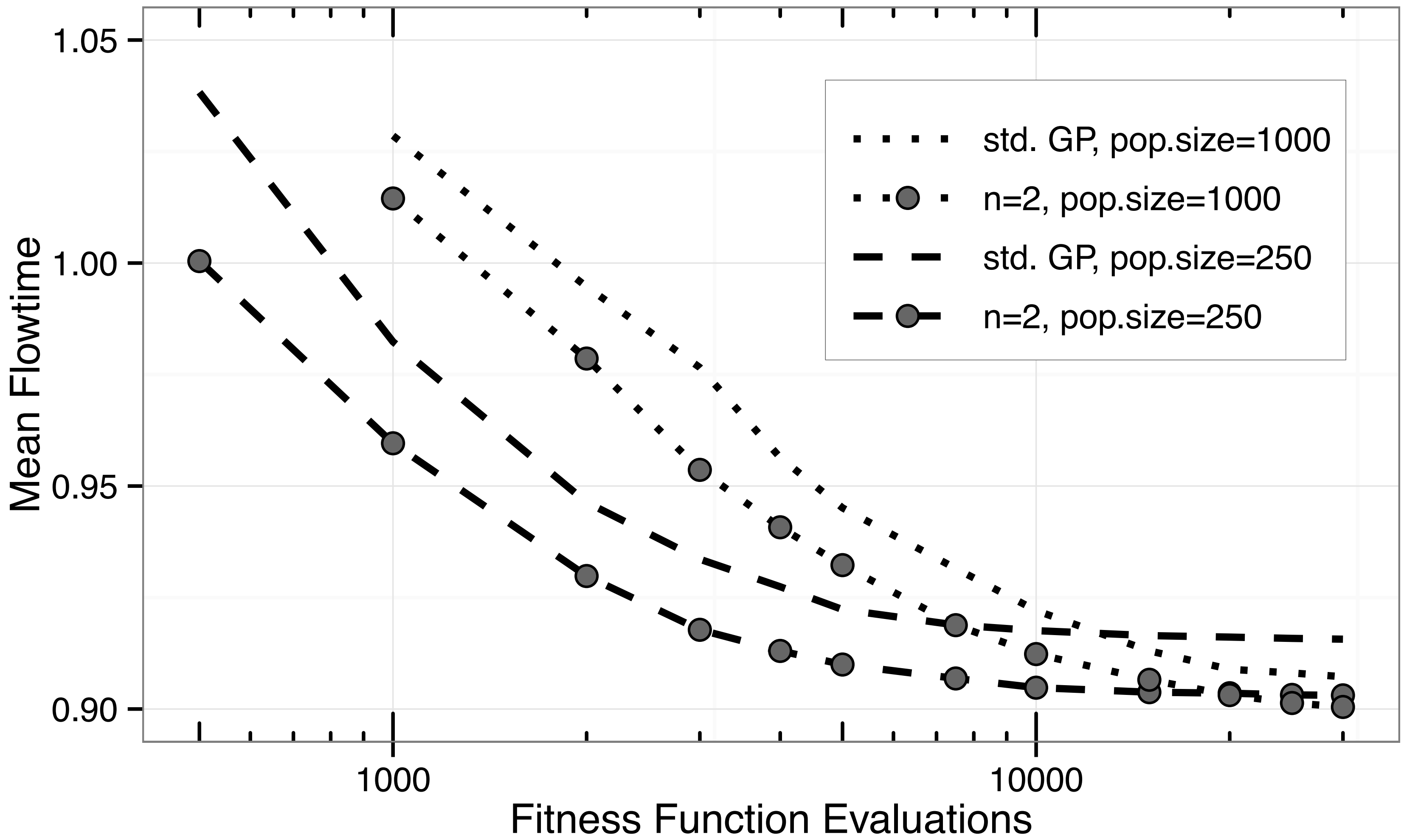
After 5,000 evaluations

	n=1	n=2	n=5	n=10
standard	**5.7 (+)**	**14.0 (++)**	**20.0 (++)**	**22.6 (++)**
n=1		**8.3 (++)**	**14.3 (++)**	**16.9 (++)**
n=2			**6.0 (++)**	**8.6 (++)**
n=5				2.6 (o)

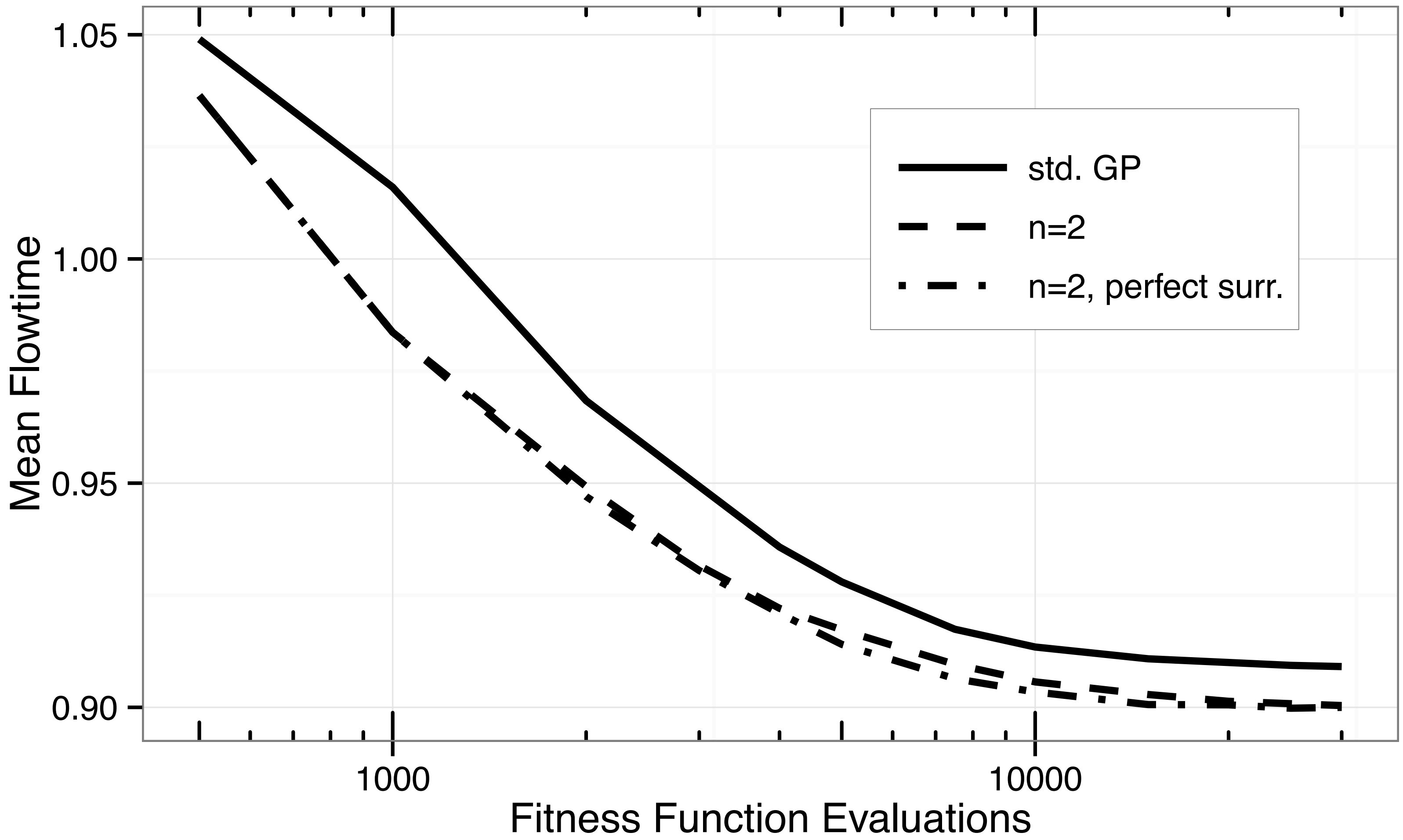
After 30,000 evaluations

	n=1	n=2	n=5	n=10
standard	**10.2 (++)**	**10.7 (++)**	**8.5 (++)**	**7.1 (++)**
n=1		0.5 (o)	-1.7 (o)	**-3.1 (+)**
n=2			-2.2 (o)	**-3.6 (++)**
n=5				-1.4 (o)

Effect of population size



Perfect surrogate



Recent alternatives [Nguyen et al., Trans. on Cybern., 2016]

- Use a simplified simulation model
  - Shorter warm-up period
  - Shorter simulation
  - Reduce complexity by reducing the number of machines and number of operations per job


QUESTIONS