Limits of Scalability of Multi-objective Estimation of Distribution Algorithms

Kumara Sastry[1], Martin Pelikan[2], David E. Goldberg[1]

[1]Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of General Engineering
University of Illinois at Urbana-Champaign

[2]Department of Math and Computer Science
University of Missouri St. Louis

ksastry@uiuc.edu, mpelikan@cs.umsl.edu, deg@uiuc.edu

Supported by **AFOSR** F49620-03-1-0129, **NSF/DMR at MCC** DMR-99-76550, NSF/ITR at CPSD DMR-0121695, DOE at FS-MRL DEFG02-91ER45439.

Foreword: Limits of MOEDA Scalability

- Limited scalability analysis of multiobjective evolutionary algorithms and estimation of distribution algorithms
- Scalable single-objective problems are extremely hard
  - Nicher gets overwhelmed due to exponential increase in Pareto-optimal solutions
- Ensuring scalability
  - Problems with only some competing sub-solutions between multiple objectives
  - Multiple objectives should mostly share same sub-solutions

Outline

- Background & motivation
- Usual scalability analysis
- Multiobjective EDA (MOEDA)
- Overwhelming the niching method
- Limits of scalability of MOEDAs
- Summary and conclusions

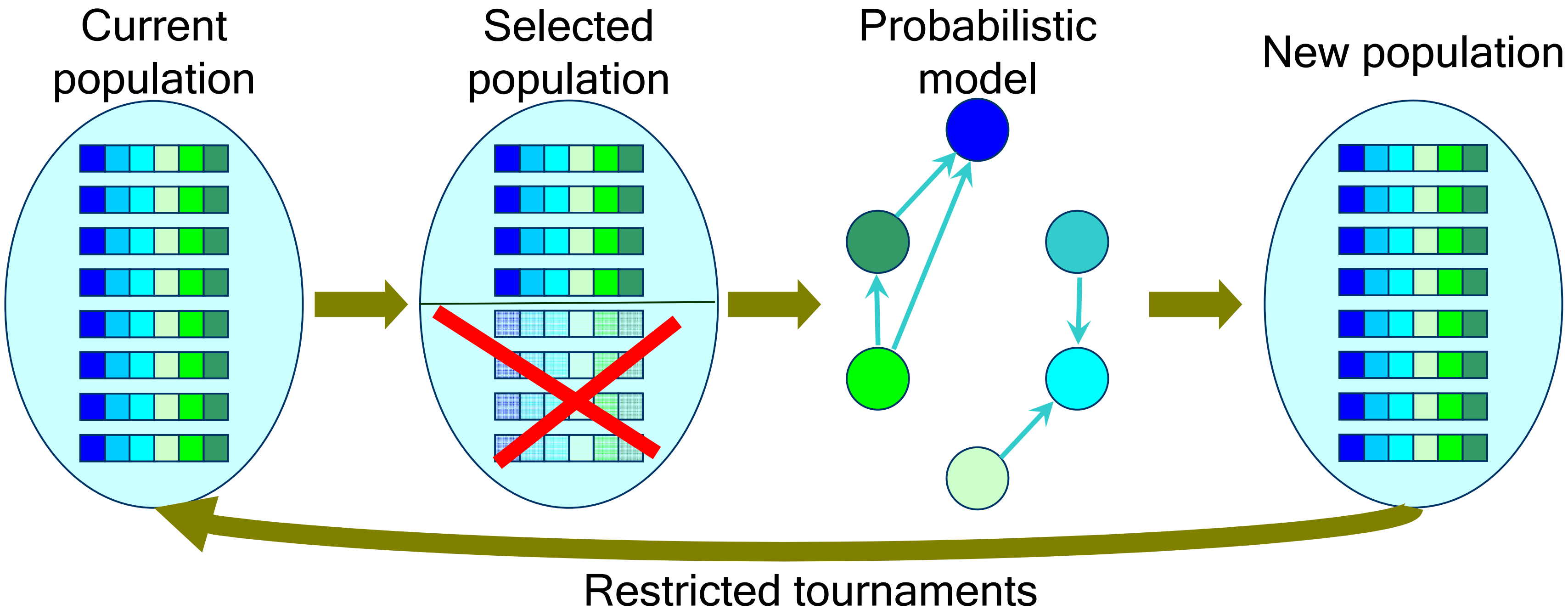
Initial Purpose: Scalability of MOEDAs

- Multiobjective EDAs [Bosman & Thierens, 2002, Khan *et al*, 2002, Ocenasek, 2002, Ahn, 2005]
    - Model-building and model-sampling of EDAs
    - Selection and replacement of multiobjective GAs
- Outperform MOEAs on boundedly-difficult additively and hierarchically decomposable problems
- Limited scalability analysis
    - Demonstrated scalability is a strong point of EDAs

How do MOEDAs scale with problem size on additively separable boundedly-difficult problems?

Multiobjective Estimation of Distribution Algorithm

- EDA: Extended compact GA [Harik, 1999]
- Multiobjective selection: NSGA-II [Deb *et al*, 2002]
- Replacement method:
  - Elitist crowding (NSGA-II)
  - Restricted tournament replacement (RTR) [Harik, 1995]



Sub-Structure Identification: Models built by eCGA

- Model-building procedure of extended compact GA
  - Partition genes into (mutually) independent groups
  - Start with the lowest complexity model
  - Search for a least-complex, most-accurate model

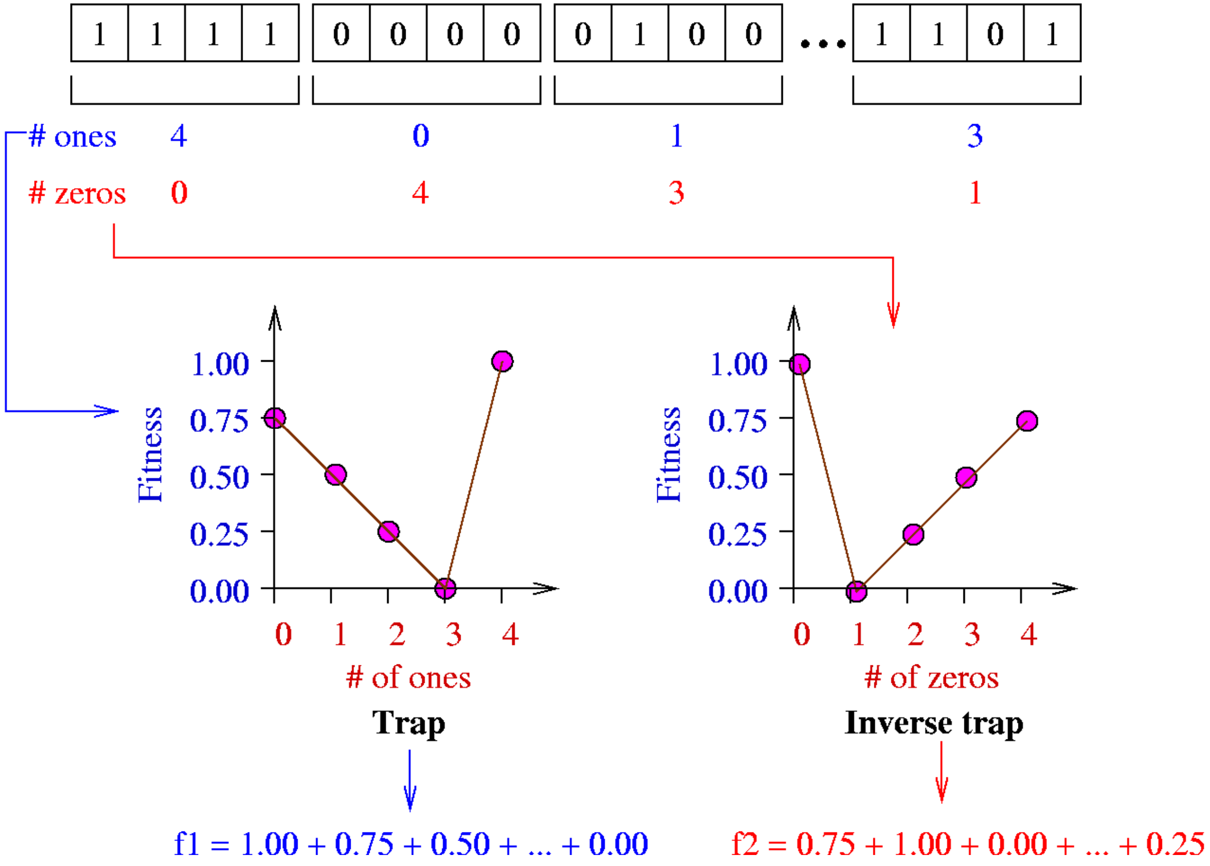
Model Structure	**Metric**
$[X_0]\ [X_1]\ [X_2]\ [X_3]\ [X_4]\ [X_5]\ [X_6]\ [X_7]\ [X_8]\ [X_9]\ [X_{10}]\ [X_{11}]$	1.0000
$[X_0]\ [X_1]\ [X_2]\ [X_3]\ [X_4X_5]\ [X_6]\ [X_7]\ [X_8]\ [X_9]\ [X_{10}]\ [X_{11}]$	0.9933
$[X_0]\ [X_1]\ [X_2]\ [X_3]\ [X_4X_5X_7]\ [X_6]\ [X_8]\ [X_9]\ [X_{10}]\ [X_{11}]$	0.9819
$[X_0]\ [X_1]\ [X_2]\ [X_3]\ [X_4X_5X_6X_7]\ [X_8]\ [X_9]\ [X_{10}]\ [X_{11}]$	0.9644
⋮	⋮
$[X_0]\ [X_1]\ [X_2]\ [X_3]\ [X_4X_5X_6X_7]\ [X_8X_9X_{10}X_{11}]$	0.9273
⋮	⋮
$[X_0X_1X_2X_3]\ [X_4X_5X_6X_7]\ [X_8X_9X_{10}X_{11}]$	0.8895

NSGA-II

- Main difference from standard GAs
  - Selection, comparison, and replacement of candidate solutions
- Two measures
  - Non-dominated ranking
    - Extract subsets of non-dominated solutions (until empty)
    - Assign ranks: Solutions extracted earlier are ranked higher
  - Crowding
    - Assign each solution a crowding distance
    - Crowding distance = front density in the neighborhood
- Comparing solutions
  - Ranks first, crowding distance second

Usual Scalability Game

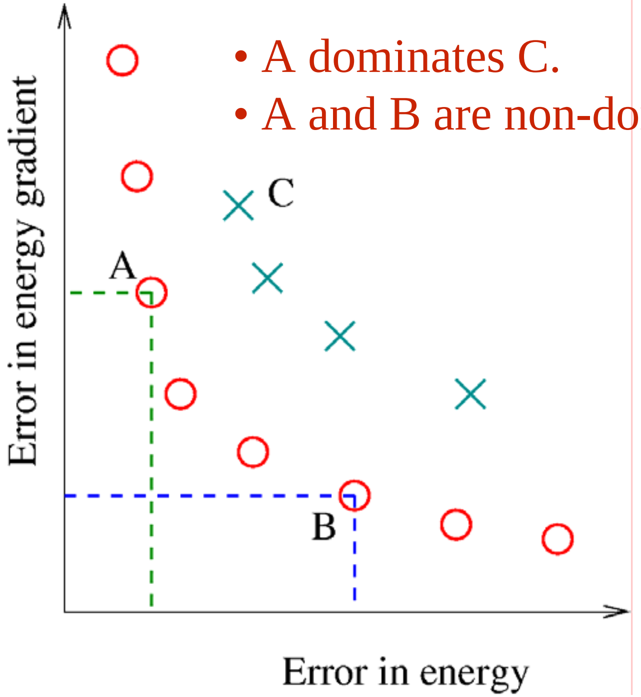
- Boundedly-difficult problem where linkage learning is critical
- Scalability as a function of # building blocks, *m*
- Use similar approach for MOEDAs



Non Domination

- Unlike single-objective problems, multi-objective problems involve a set of Pareto-optimal solutions.

- Notion of Non-Dominating Solutions

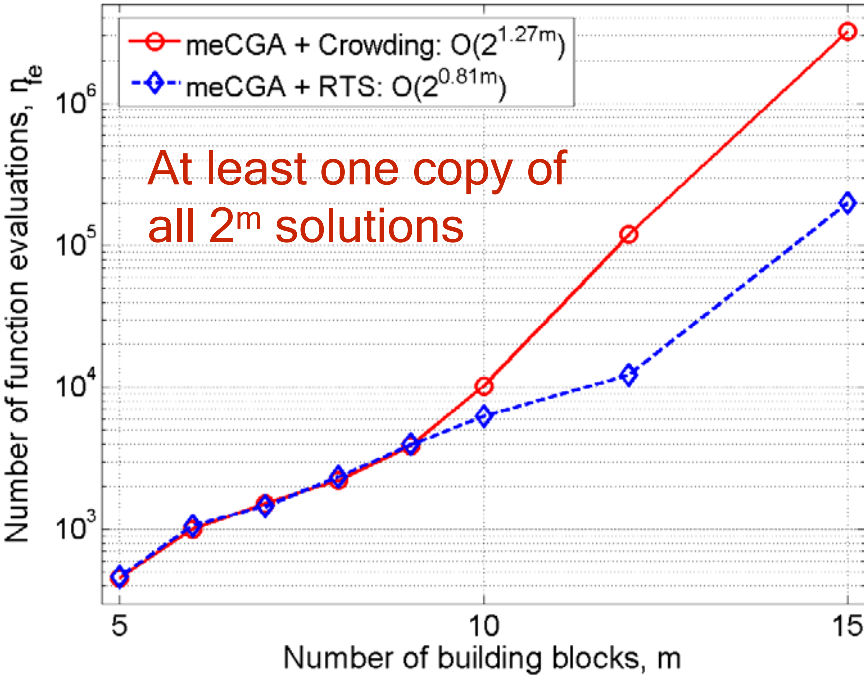


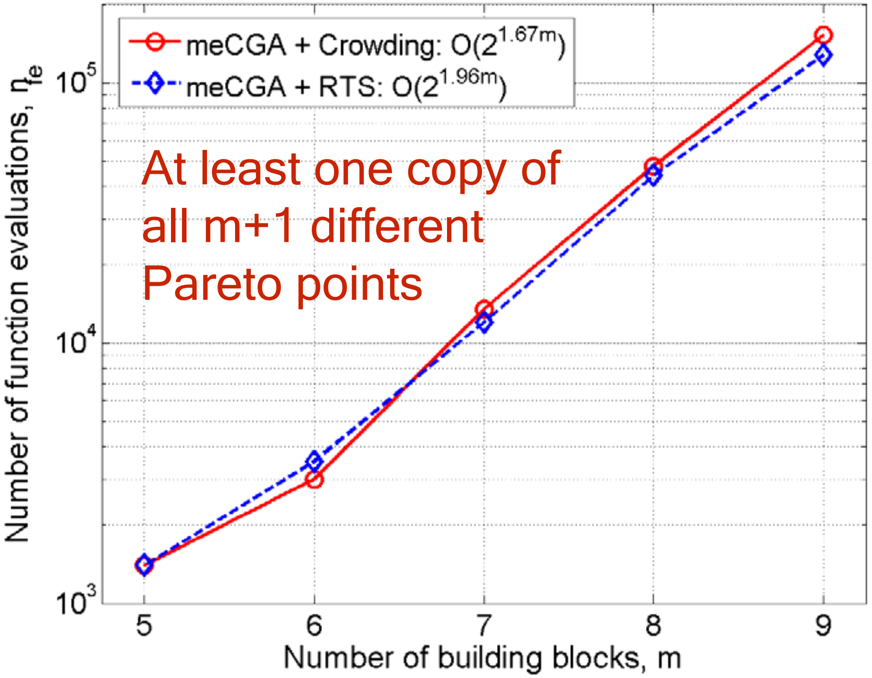
- A dominates C.
- A and B are non-dominant.

- Solution X dominates Y if:
  - X is no worse than Y in all objectives
  - X is strictly better than Y in at least one objective

Usual Scalability Game Fails!

- Building blocks are accurately identified and sampled
- meCGA scales exponentially
  - Optimal solutions composed by 0000 and 1111 blocks
  - 2^m solutions in the Pareto-optimal set
  - m+1 distinct points in the Pareto-optimal set



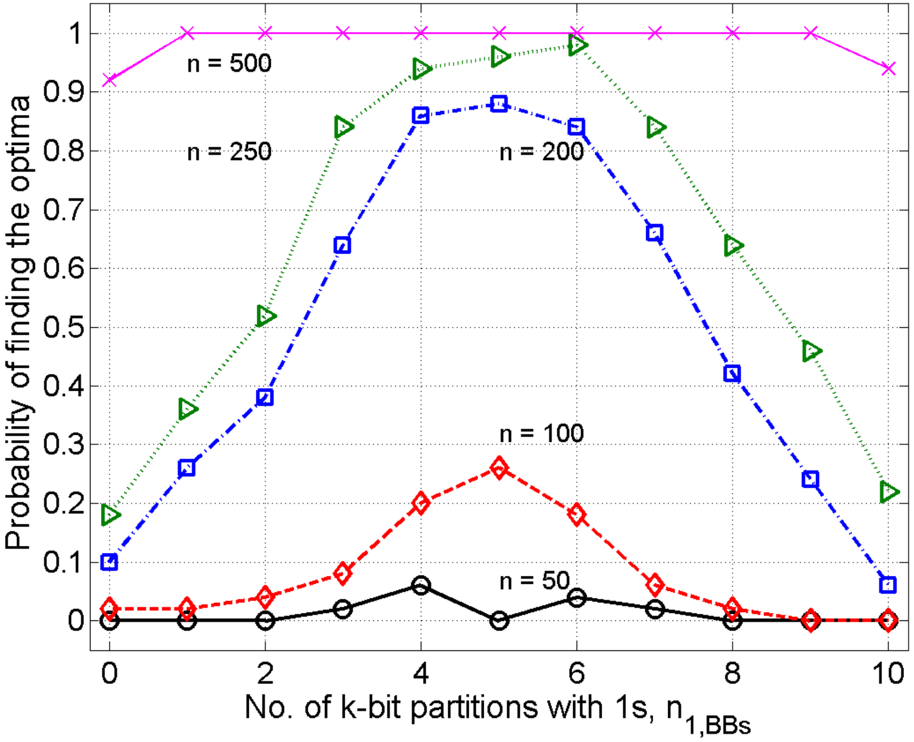


Niching Method Gets Overwhelmed

- Non-uniform distribution and exponential growth of solutions

$n_{1,BBs}$	0	1	$\cdots$	i	$\cdots$	m
$n_{0,BBs}$	m	$m-1$	$\cdots$	$m-i$	$\cdots$	0
# solutions	1	m	$\cdots$	$\binom{m}{i}$	$\cdots$	1

- One solution in the extremes
- Exponential in the middle
- Need exponentially sized population to maintain at least on copy of 2^m solutions
- Solutions in the middle affected by drift
- Solutions in the extreme are hard to obtain

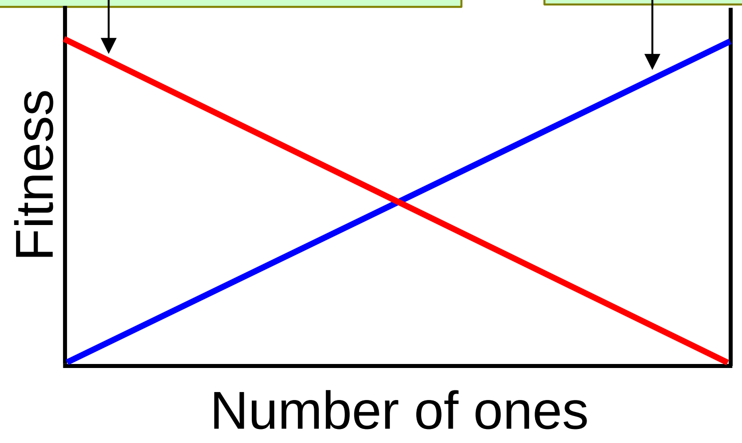


GA-Easy Problem: OneMax-ZeroMax

- Objective #1: Maximize # ones in a bitstring
- Objective #2: Maximize # zeros in a bitstring

$$\text{zeromax}\left(X_1, X_2, \ldots, X_n\right) = \sum_{i=1}^{n}\left(1 - X_i\right)$$

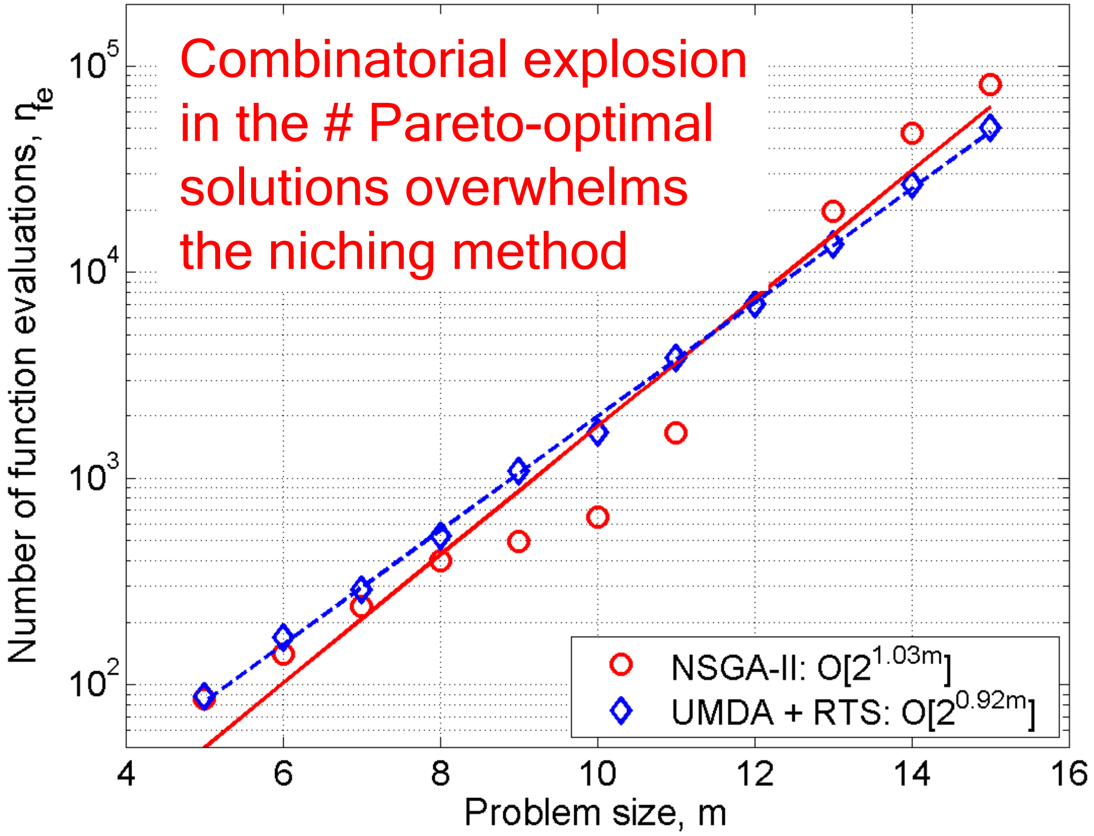
$$\text{onemax}\left(X_1, X_2, \ldots, X_n\right) = \sum_{i=1}^{n} X_i$$



- Single objective optimization: sub-quadratic scalability
- Entire search space is Pareto-optimal!
- Linkage learning not required

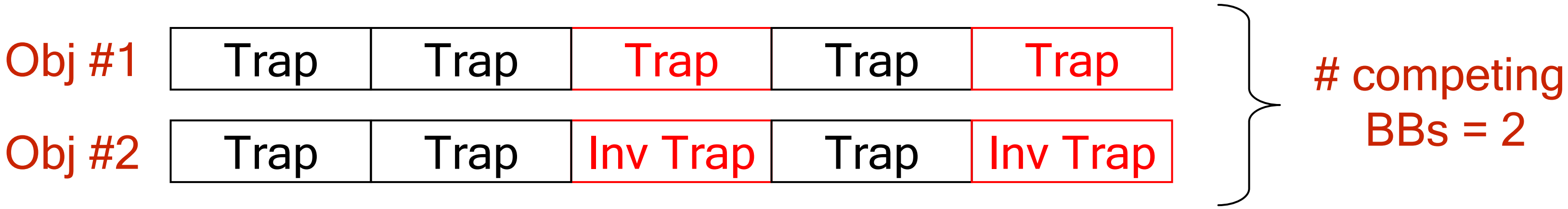
GA-Easy Problem: OneMax-ZeroMax

- Pareto-optimal points are (*i*, *n-i*)
- Number of representatives has binomial distribution
- Strong bias to the middle of the front!
- Extremes of the Pareto-optimal front have 1 representative


Combinatorial explosion in the # Pareto-optimal solutions overwhelms the niching method

Achieving Tractability in MOEAs: 3 Scenarios

- Practical population sizes can't yield all optimal solutions
  - EDAs outperform simple MOEAs [Pelikan, et al 2005]
- Size population adequately to handle exponential growth in the # Pareto-optimal solutions
  - Linearly with the # Pareto-optimal solutions [Mahfoud, 1994]
- Understand the fundamental limit of growth in # Pareto-optimal solutions that can yield polynomial scale-up
  - Control the growth in # competing building blocks



Obj #1	Trap	Trap	Trap	Trap	Trap
Obj #2	Trap	Trap	Inv Trap	Trap	Inv Trap

competing BBs = 2

Amount of Competition

- Two extremes
  - No partitions compete (like single objective)
  - All partitions compete (tough)
- Other problems in between
  - But overlap can lead to interesting stuff, too.
- Basic scenario
  - Same decomposition for all objectives
  - Objectives agree in some partitions
  - Objectives compete in other partitions

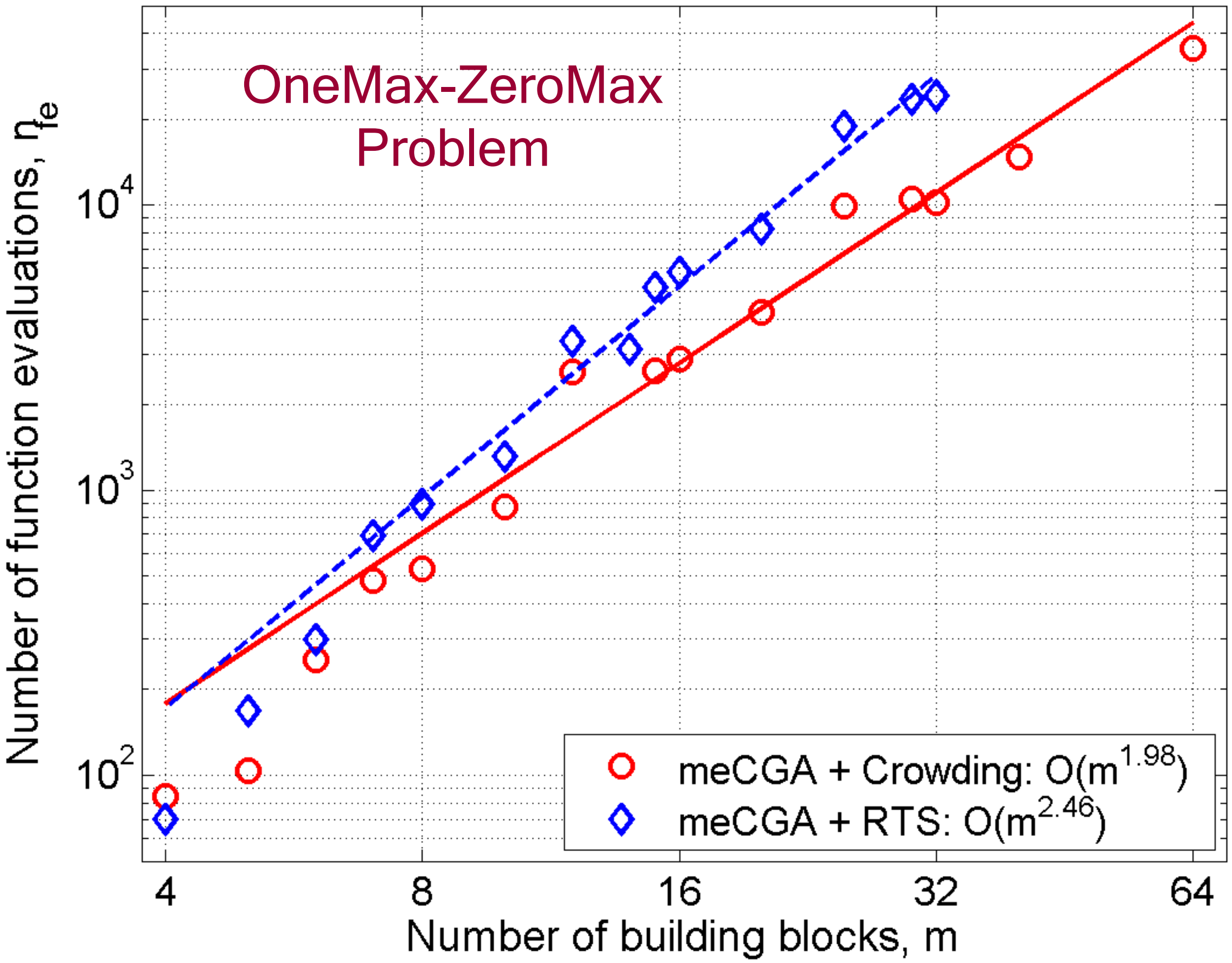
Limits on Growth Rate of Competing BBs

- eCGA population sizing: $\boxed{n = c_1 \cdot 2^k \cdot m \log(m)}$
- Niching population sizing: $\boxed{n \approx c_2 \cdot 2^{m_d}}$ → # competing BBs
- Ensuring polynomial scalability

Niching Pop. sizing ≤ eCGA pop. sizing

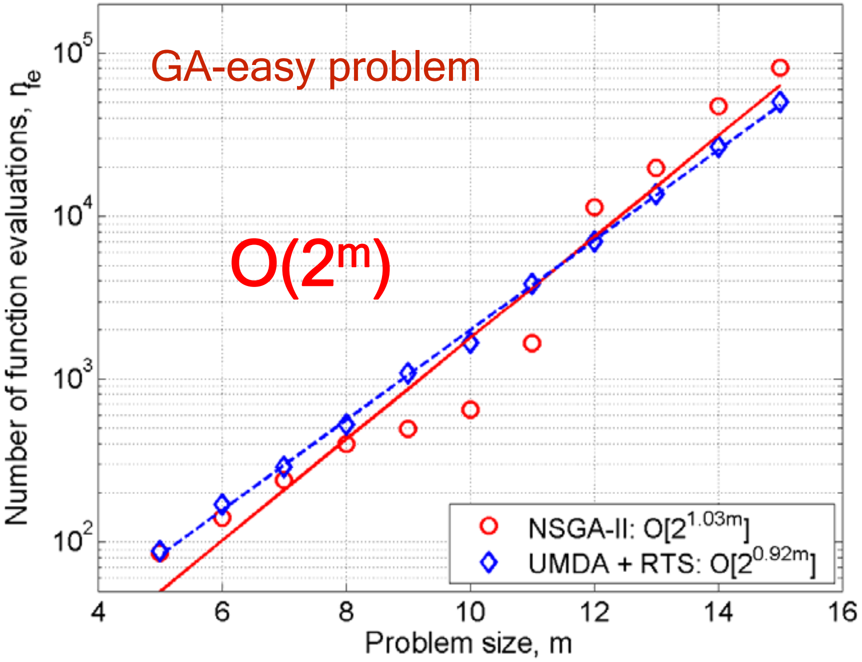
$$\boxed{m_d \approx k + \log_2(m)}$$

Logarithmic Growth of Competing BBs Ensures Polynomial Scalability



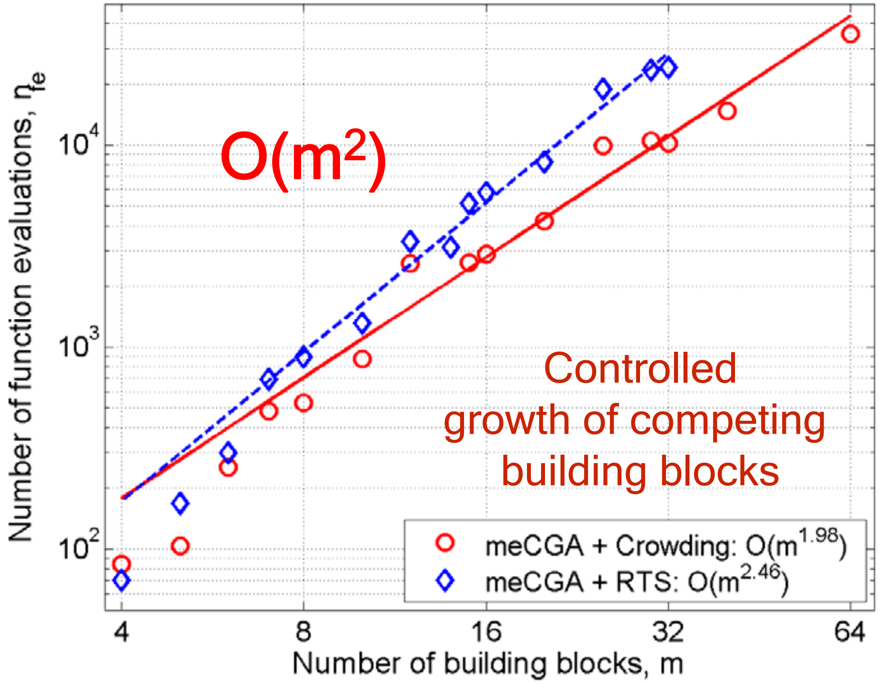
Summary: Scalability of Multi-objective EDAs

- Exponential scalability!
  - Even with effective linkage learning
  - Even on simplest problems
- Niching method gets overwhelmed quickly
  - Competing building blocks have to be limited



[Sastry, Pelikan, & Goldberg, 2005]

$m_{\text{competing}} \propto \log_2(m)$



Some Open Questions?

- Multiple objectives can make problems easier
- What if different objectives need different BB sizes?