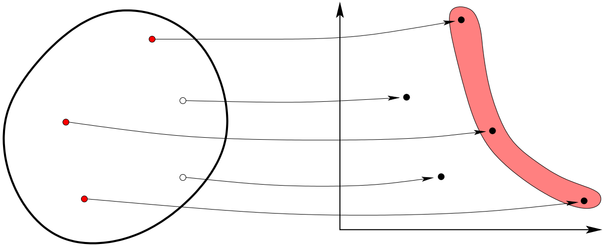
On The Effect of Populations in Evolutionary Multi-objective Optimization

Oliver Giel[1]   Per Kristian Lehre[2]

[1]Fachbereich Informatik, Lehrstuhl 2
Universität Dortmund

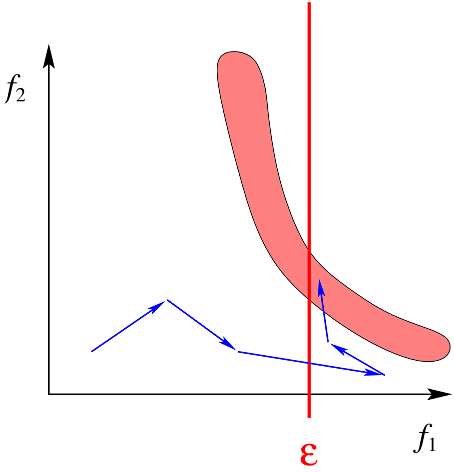
[2]Dept. of Computer and Information Science
Norwegian University of Science and Technology (NTNU)

Are populations needed in MOEAs?



- Populations in **multi-objective** EAs are often motivated by the need to find a **set of Pareto-optimal solutions**.
- Why not use individual-based algorithm with restarts?

ε-constrained method



ε-constrained method

Laumanns, Thiele, Zitzler 2004

Algorithm	LOTZ	COCZ
ε-constrained (restarts)	$\Omega(n^3)$	$\Omega(n^2 \log n)$
SEMO	$O(n^3)$	$O(n^2 \log n)$
FEMO	$O(n^2 \log n)$	$O(n^2 \log n)$
GEMO	$O(n^2 \log n)$	$O(n^2)$

$\Longrightarrow$ Population-based approaches can slightly outperform a single-individual approach.

But: All considered methods are efficient and FEMO/GEMO seem tailored to LOTZ/COCZ.

Outline

1. Bi-objective function
2. Single-individual based algorithms
3. A population-based algorithm
4. Conclusion

Bi-objective function

Definition

For bitstrings x of length $n = k \cdot m$

$$f(x) \ := \ (\ 2^{j \cdot m} \cdot (v+1),\ 2^{(k-j-1) \cdot m} \cdot (v+1)\ )$$

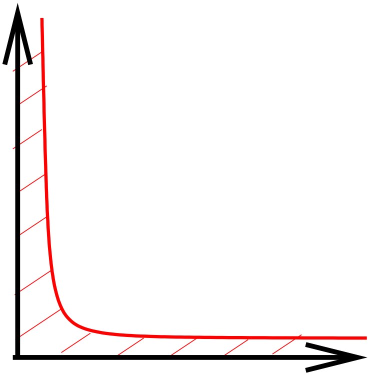
- **active block index**

  $j :=$ leftmost block with with least number of 1-bits.
- **active block value**

  $v :=$ number of 1-bits in the active block.

x	$\lvert x \rvert_0$	$\lvert x \rvert_1$	$\lvert x \rvert_2$	$\lvert x \rvert_3$	j	$f(x)$
000 000 000 000	0	0	0	0	0	(1, 512)
111 011 010 001	3	2	1	1	2	(128, 16)
010 111 011 001	1	3	2	1	0	(2, 1024)
111 111 111 111	3	3	3	3	0	(4, 2048)

Bi-objective function



- Non-convex Pareto front.

Properties of objective function (1)

Proposition

For search points x and y with active block indices i and j

1. x *and* y *are comparable if and only if* $i = j$.
2. *Moreover, if* $i = j$, *then* $x \preceq y$ *is equivalent to* $|x|_i \leq |y|_j$.

Example

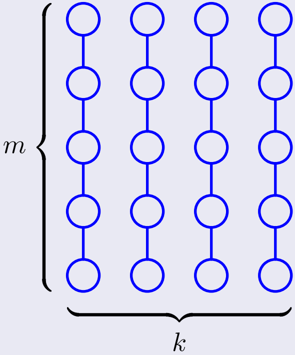
$x =$ 0000 0000 0011 0011 $\qquad f(x) = (1, 4096)$

$y =$ 0001 0000 0011 0011 $\qquad f(y) = (16, 16)$

$z =$ 0011 0001 0011 0011 $\qquad f(z) = (32, 32)$

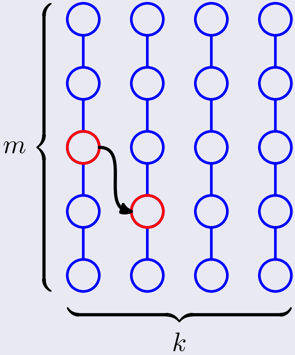
$$x \parallel y \qquad x \parallel z \qquad y \prec z$$

Properties of objective function (2)



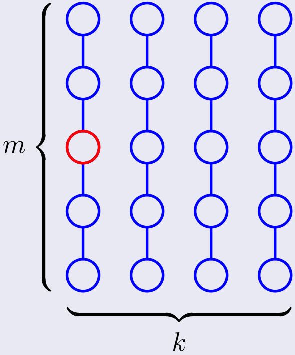
- k independent paths.

Properties of objective function (2)



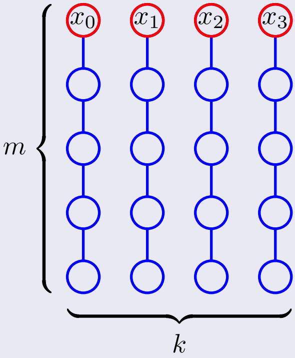
- k independent paths.
- Short distance between paths.

Properties of objective function (2)



- k independent paths.
- Short distance between paths.
- Each level in path a plateau.

Properties of objective function (2)



- k independent paths.
- Short distance between paths.
- Each level in path a plateau.
- Pareto set at end of paths.

$x_0 = $ 11111 11111 11111 11111

$x_1 = $ 11111 01111 01111 01111

$x_2 = $ 11111 11111 01111 01111

$x_3 = $ 11111 11111 11111 01111

Single-individual based MOEAs

Choose x uniformly from $\{0,1\}^n$.
Repeat
Apply mutation to x to obtain x'.
If selection favors x' over x
then $x := x'$.

Mutation	local: flip one uniformly chosen bit.
	global: flip each bit independently with prob $1/n$.
Selection	strong: x' dominates x.
	weak: x' weakly dominates x.
	weakest: x' weakly dominates x or incomparable.
	ε-constraint: according to ε-constraint method.

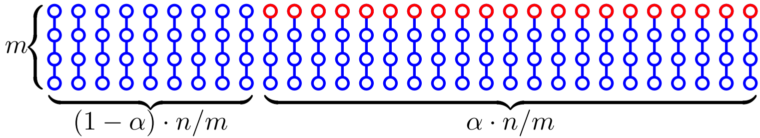
Exponential runtime with strong or weak selection

Theorem

Let the block length m be an arbitrary constant. For every constant fraction $0 < \alpha < 1$, there is a subset F of the Pareto front F^* such that $|F| \geq \alpha|F^*|$ and $e^{c \cdot n}$ runs find any solution from F with a probability of only $e^{-\Omega(n)}$.

Proof Idea.

- Active block index will not change during a run.
- W.h.p., initial active block index less than $(1-\alpha) \cdot n/m$.
- Exponentially many restarts needed to reach active blocks higher than $(1-\alpha) \cdot n/m$.



Exponential runtime with weakest selection

Theorem

For the number of blocks $k \geq 4$, the probability that $e^{c' \cdot n}$ runs of upto $e^{c \cdot n}$ steps find any Pareto optimal solution is only $e^{-\Omega(n)}$.

Proof Idea.

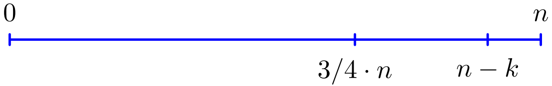
Pareto optimal points have at least $n - k$ 1-bits.

If number of 1-bits is higher than $3/4 \cdot n$ then

- less than $n/4$ 0-bits can be flipped to 1.
- more than $3/4 \cdot n - m \geq n/2$ 1-bits can be flipped to 0, and

Increasing 1-bits from $3/4 \cdot n$ to $n - k$ is hard.

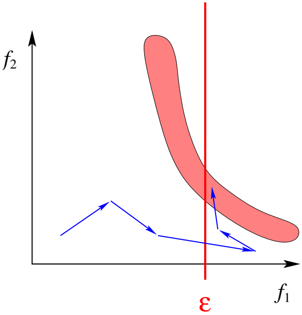
- Gambler's Ruin / Drift Theorem.



Exponential runtime with ε-constrained method

Theorem

Let the block length $m \geq 2$ be a constant. For every constant fraction $0 < \alpha < 1$, there is a subset F of the Pareto Front F^* such that $|F| \geq \alpha |F^*|$ and $e^{c \cdot n}$ runs of upto $e^{c' \cdot n}$ steps find any solution from F with a probability of only $e^{-\Omega(n)}$.



SEMO - A population-based algorithm

$P := \{x\}$, where x is uniformly chosen from $\{0, 1\}^n$.
Repeat
Choose x uniformly from P.
Apply mutation to x to obtain x'.
If x' is not dominated by any individual in P
then add x' to P, and remove all individuals weakly dominated by x' from P.

Polynomial runtime with SEMO

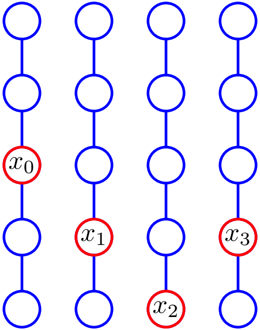
Theorem

The expected time until the population covers the Pareto front is $O(nk^2 \log m) = O(n^3 \log n)$.

Proof Idea.

- The expanding population quickly covers all paths.
- The paths are optimized in parallell.

Polynomial runtime with SEMO



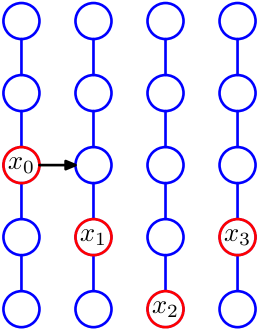
$$x_0 = \textcolor{red}{11100}\ 01101\ 01011\ 01100$$

$$x_1 = 11010\ \textcolor{red}{10010}\ 10100\ 10001$$

$$x_2 = 10001\ 01010\ \textcolor{red}{00000}\ 00000$$

$$x_3 = 11001\ 11010\ 00011\ \textcolor{red}{00001}$$

Polynomial runtime with SEMO



$x_0 =$ 11100 01101 01011 01100

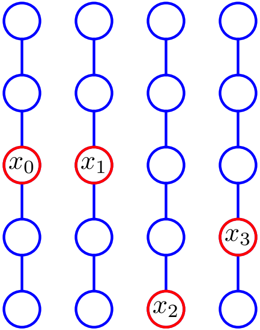
$\rightarrow$ 11110 01101 01011 01100

$x_1 =$ 11010 10010 10100 10001

$x_2 =$ 10001 01010 00000 00000

$x_3 =$ 11001 11010 00011 00001

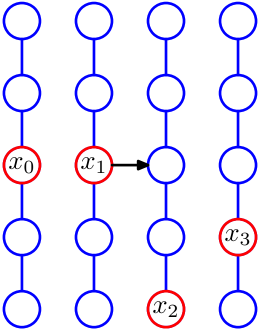
Polynomial runtime with SEMO



$x_0 =$11100 01101 01011 01100
$x_1 =$11110 01101 01011 01100

$x_2 =$10001 01010 00000 00000
$x_3 =$11001 11010 00011 00001

Polynomial runtime with SEMO



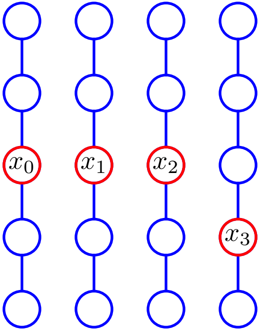
$x_0 =$ 11100 01101 01011 01100

$x_1 =$ 11110 01101 01011 01100

$\rightarrow$ 11110 01111 01011 01100

$x_2 =$ 10001 01010 00000 00000

$x_3 =$ 11001 11010 00011 00001



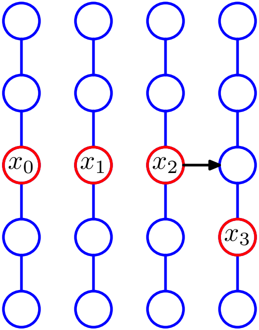
$x_0 =$ 11100 01101 01011 01100

$x_1 =$ 11110 01101 01011 01100

$x_2 =$ 11110 01111 01011 01100

$x_3 =$ 11001 11010 00011 00001

Polynomial runtime with SEMO



x_0 = 11100 01101 01011 01100

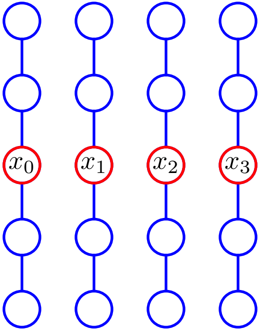
x_1 = 11110 01101 01011 01100

x_2 = 11110 01111 01011 01100

$\rightarrow$ 11110 01111 01111 01100

x_3 = 11001 11010 00011 00001

Polynomial runtime with SEMO



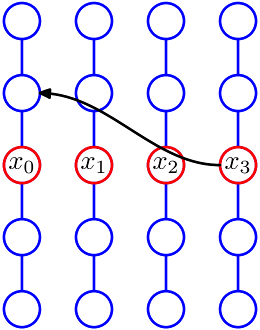
$x_0 =$ 11100 01101 01011 01100

$x_1 =$ 11110 01101 01011 01100

$x_2 =$ 11110 01111 01011 01100

$x_3 =$ 11110 01111 01111 01100

Polynomial runtime with SEMO



x_0 = 11100 01101 01011 01100

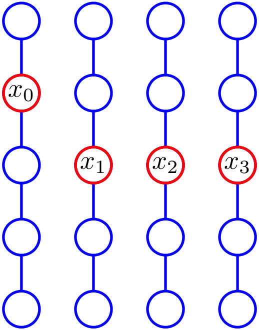
x_1 = 11110 01101 01011 01100

x_2 = 11110 01111 01011 01100

x_3 = 11110 01111 01111 01100

$\rightarrow$ 11110 01111 01111 11100

Polynomial runtime with SEMO



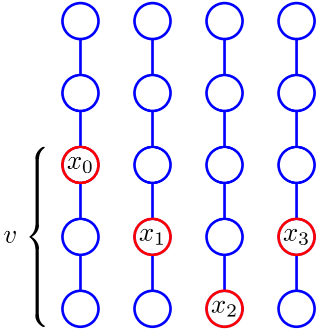
$x_0 =$ 11110 01111 01111 11100

$x_1 =$ 11110 01101 01011 01100

$x_2 =$ 11110 01111 01011 01100

$x_3 =$ 11110 01111 01111 01100

Polynomial runtime with SEMO



x_0 =11100 01101 01011 01100

x_1 =11010 10010 10100 10001

x_2 =10001 01010 00000 00000

x_3 =11001 11010 00011 00001

- Active path value never decreases.
- Success probability at least $(1/k) \cdot (m - v)/n$.

$\longrightarrow$ Leading active path value increases within $k \cdot \frac{kn}{(m-v)}$ steps.

- $\sum_{i=1}^{k-1} \frac{kn}{m-v} = k^2 n \sum_{v=1}^{m} 1/m = O(k^2 n \log m)$

Conclusion

We presented a problem where

- population-based SEMO is **efficient**,
- the following single individual approaches are **not successful:**
  - linear aggregation functions
    (not applicable, non-convex Pareto front),
  - multiple runs with strong, weak, and weakest selection
    (exponentially small success probability),
  - multiple runs with ε-constrained method
    (exponentially small success probability, independent of ε).

$\rightarrow$ Strong (?) evidence that a population of incomparable solutions can be essential in MOEAs.