Foundations of Evolutionary Multi-Objective Optimization

Tobias Friedrich
Algorithms and Complexity
Max Planck Institute for Informatics

Frank Neumann
School of Computer Science
University of Adelaide

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

max planck institut informatik

Introduction

- Evolutionary algorithms are in particular successful for multi-objective optimization problems
- Why?
- Multi-objective problems deal with several (conflicting) objective functions.
- Compute different trade offs with respect to the given objective functions (Pareto front, Pareto optimal set).
- Population of an EA may be used to compute/approximate the Pareto front.

This tutorial: Theoretical understanding of EAs for multi-objective optimization

Analyze basic features of such algorithms and point out differences

Multi-Objective Optimization

Multi-Objective Optimization

$f\colon \mathbb{B}^n \to \mathbb{R}^m$

Dominance in the objective space

u *weakly dominates* v $(u \succeq v)$ iff $u_i \geq v_i$ for all $i \in \{1, \ldots, m\}$

u *dominates* v $(u \succ v)$ iff $u \succeq v$ and $u \neq v$.

Concept may be translated to search points

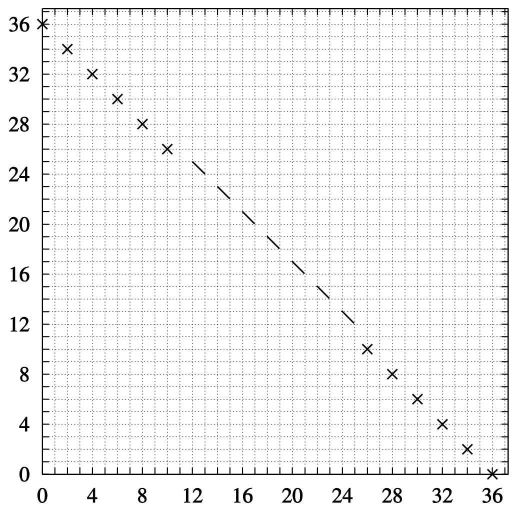
$x \succeq y$ iff $f(x) \succeq f(y)$
$x \succ y$ iff $f(x) \succ f(y)$

Non-dominated objective vectors constitute the Pareto front

Classical goal:
Compute for each Pareto optimal objective vector a corresponding solution

Large Pareto Front

Problem: Pareto front may be large



Think about approximations!!!

Approximations

Large Fronts can not be computed in polynomial time

Goal: Compute a good approximation

Two measures of approximation

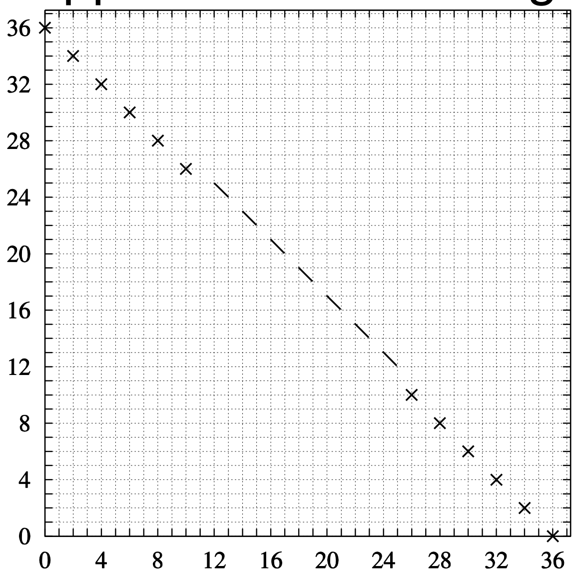
- Multiplicative epsilon dominance

  u ϵ-dominates v $(u \succeq_\epsilon v)$ iff $(1+\epsilon) \cdot u_i \geq v_i$ for all $i \in \{1, \ldots, m\}$.

- Additive epsilon dominance

  u ϵ-dominates v $(u \succeq_\epsilon v)$ iff $u_i + \epsilon \geq v_i$ for all $i \in \{1, \ldots, m\}$.

Approximations of large fronts



Simple Evolutionary Multi-objective Optimizer (SEMO)

- ★ choose an initial population P with $|P| = 1$ uniformly at random
- ★ Repeat
  - ▶ choose a parent $x \in P$ uniformly at random
  - ▶ create an offspring y by flipping each bit of x with probability $1/n$
- ▶ If $(\nexists z \in P\colon z \succ y)$, set $P \leftarrow (P \setminus \{z \in P \mid y \succeq z\}) \cup \{y\}$

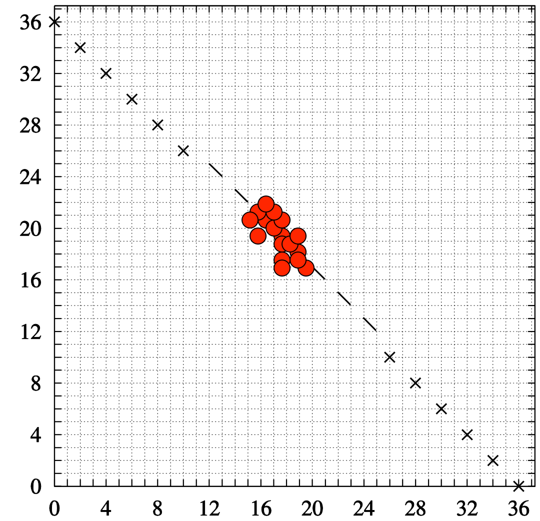
- SEMO keeps for each non-dominated objective vector found so far, one single individual.

Theory

Point of interest in the following:

- Runtime to compute the compute/approximate the Pareto front
- Number of fitness evaluations
- Expected polynomial time
- Exponential time with probability exponentially close to 1

SEMO on LF



Exponential time

Diversity Mechanisms

Diversity Evolutionary Multi-Objective Optimizer (DEMO)

Ensure diversity with respect to objective vectors

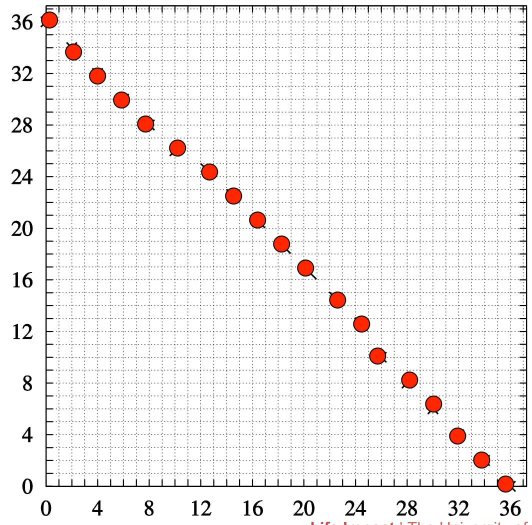
$$b(x) = (b_1(x), \ldots, b_m(x)) \text{ with } b_i(x) := \lfloor f_i(x)/\delta \rfloor$$

Laumanns, Thiele, Zitzler (2003)

- choose an initial population P with $|P| = 1$ uniformly at random
- Repeat
  - choose a parent $x \in P$ uniformly at random
  - create an offspring y by flipping each bit of x with probability $1/n$
  - If $(\nexists z \in P\colon z \succ y \vee b(z) \succ b(y))$, set $P \leftarrow (P \setminus \{z \in P \mid b(y) \succeq b(z)\}) \cup \{y\}$

DEMO keeps an additive delta-approximation of the search points examined so far.

DEMO on LF



$\delta = \epsilon$

Runtime

$O(n^2 \log n)$

Frank Neumann

Rank- And Distance-based Evolutionary Multi-objective Optimizer (RADEMO)

- choose an initial population P with $|P| = \mu$ uniformly at random
- Repeat
  - choose a parent $x \in P$ uniformly at random
  - create an offspring y by flipping each bit of x with probability $1/n$
  - choose an individual $z \in P \cup \{y\}$ for removal.
  - set $P \leftarrow (P \cup \{y\}) \setminus \{z\}$

Input: set of search points Q

- set $Q' \leftarrow \arg\max_{x \in Q} rank_Q(x)$
- set $Q'' \leftarrow \arg\min_{x \in Q'} distance_Q(x)$
- $z \in Q''$ chosen uniformly at random

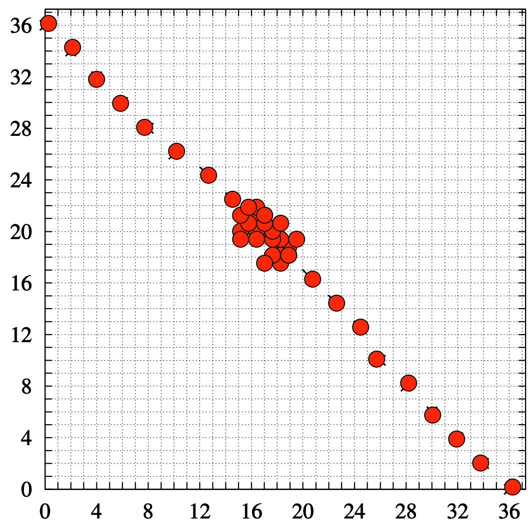
$rank_Q(x) := |\{y \in Q \mid y \succ x\}|$

$distance_Q(x) := (distance_Q^0(x), \ldots, distance_Q^{|Q|-1}(x))$

$distance_Q^k(x)$: distance $d(f(x), f(y))$ from $x \in Q$ to its k-th nearest neighbor

maximum metric: $d(u, v) := \max_{i \in \{1,\ldots,m\}} |u_i - v_i|$

RADEMO on LF



$\mu \geq n/2 + 1$

Runtime

$O(\mu n \log n)$

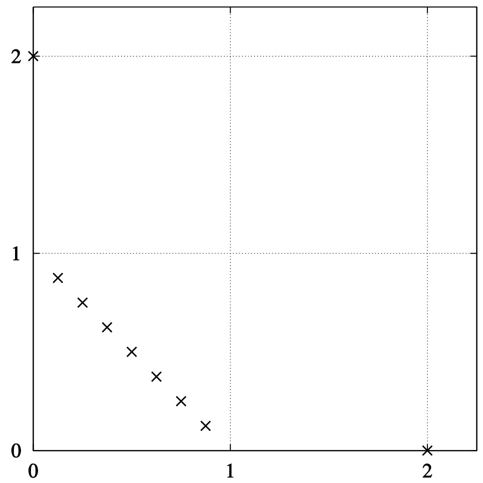
Frank Neumann

- Delta-Dominance and density estimator help to approximate a large Pareto front

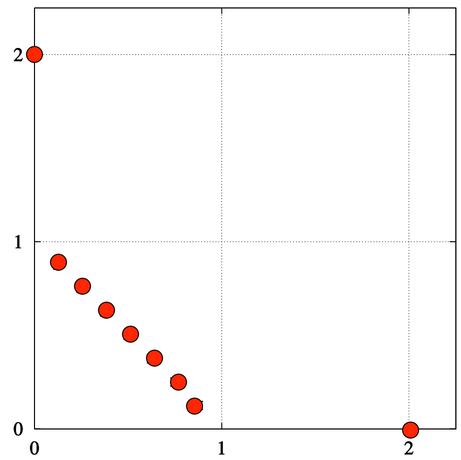
Now:

- Point out the differences of the two approaches
- Show where they even fail on small Pareto fronts

Small Front SF



SEMO on SF

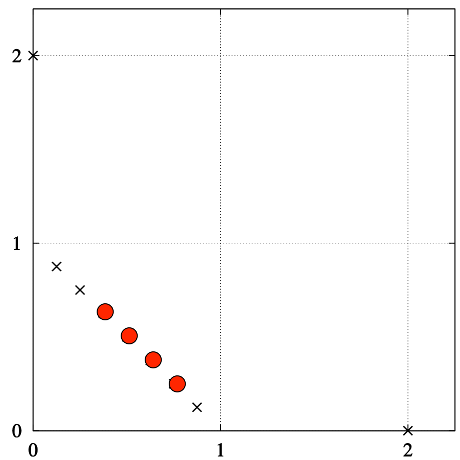


Runtime

$O(n^2 \log n)$

Frank Neumann

DEMO on SF

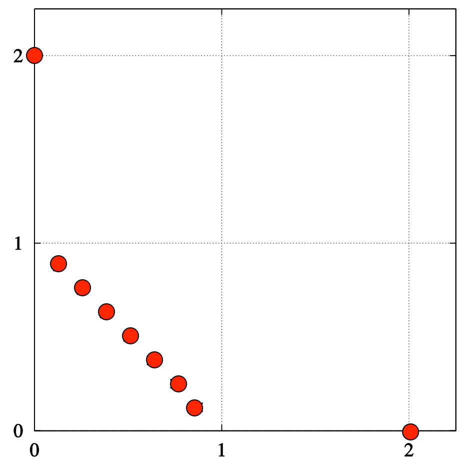


$\delta \geq \epsilon$

Exponential time

Frank Neumann

RADEMO on SF



$\mu \geq 2$

Runtime

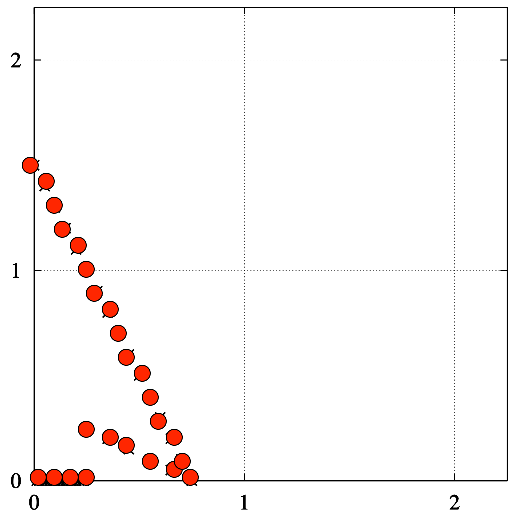
$O(\mu n \log n)$

Frank Neumann

- Delta-Dominance approach does random search if the size of the boxes is too large.
- Even simple problems can not be approximated well

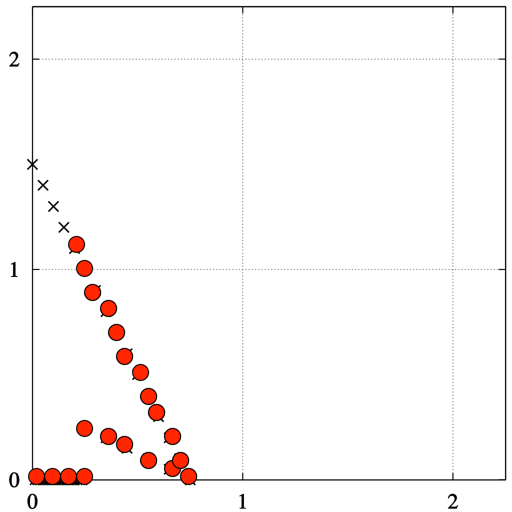
- Consider the drawback of the density estimator
- Which structures are difficult when using this approach?

SEMO on TF



Runtime
$O(n^3)$

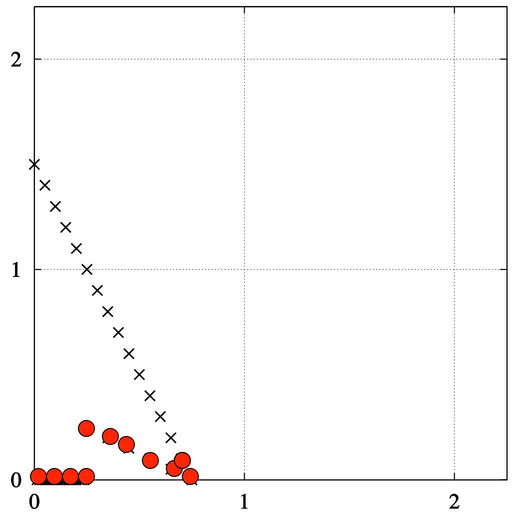
DEMO on TF



Runtime
$O(n^3)$

RADEMO on TF

RADEMO on TF



$2 \leq \mu = O(n^{1/2-\epsilon})$

Exponential time

Summary

	SEMO	DEMO	RADEMO
LF	exp	poly	poly
SF	poly	exp	poly
TF	poly	poly	exp

Summary on Diversity

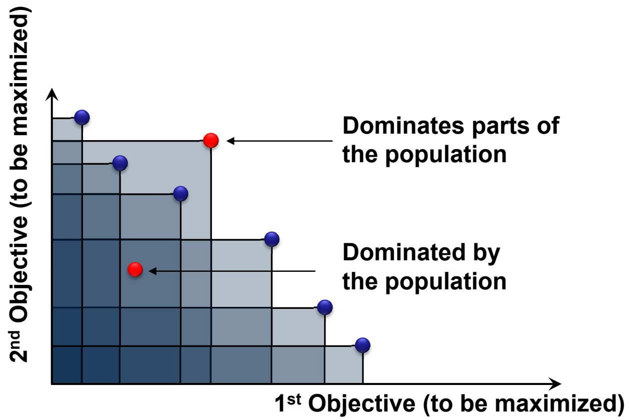
- Many multi-objective problems have large Pareto fronts
- Diversity mechanisms are necessary to achieve a good approximation (see SPEA2, NSGA-II)
- Rigorous results for the use of such mechanisms
- Delta-dominance and density estimator help to spread over a large front
- Simple situations where such mechanisms fail
- Might even fail to approximate small Pareto front that is easily computable by SEMO

The Hypervolume Indicator

Hypervolume Indicator

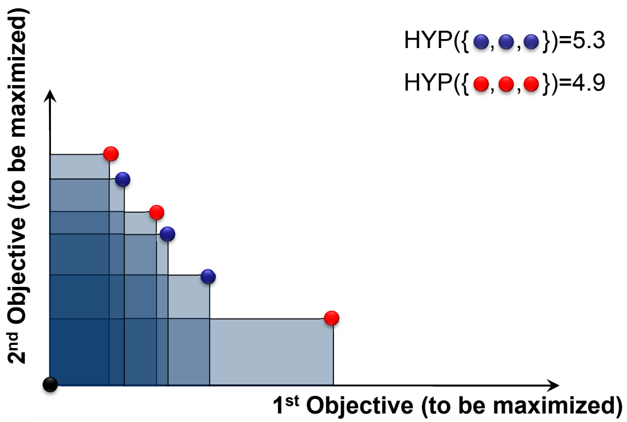
max planck institut informatik

- A Multi-objective fitness function:



Hypervolume Indicator

- Which population is better?



Tobias Friedrich

Hypervolume Indicator

- Given: n axis-parallel boxes in d-dimensional space (boxes all have (0,...,0) as bottom corner)
- Task: Measure (volume) of their union
- Property of "strict Pareto compliance":
  - Consider two Pareto sets A and B:
  - Hypervolume indicator values A higher than B if the Pareto set A dominates the Pareto set B

Tobias Friedrich

Hypervolume Indicator

- Given: n axis-parallel boxes in d-dimensional space (boxes all have (0,...,0) as bottom corner)
- Task: Measure (volume) of their union
- Popular Algorithms:
  - HSO: $\mathcal{O}(n^d)$ [Zitzler'01, Knowles'02]
  - BR: $\mathcal{O}(n^{d/2} \log n)$ [Beume Rudolph'06]
  - Many (heuristical) improvements and specialized algorithms for small dimensions
  - Only Lower Bound: $\Omega(n \log n)$ [Beume et al.'07]

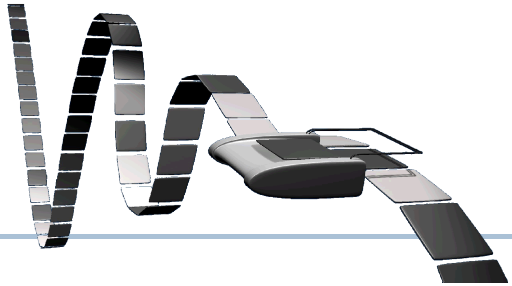
Tobias Friedrich

Computational Complexity of the Hypervolume Indicator

Tobias Friedrich

#P-hardness of HYP

- P = deterministic polynomial time
- NP = non-deterministic polynomial time
  (Is there an accepting path?)
- #P = counting in polynomial time ("sharp-P")
  (How many accepting paths?)



#P-hardness of HYP

- Consider #MON-CNF:
- Given: monotone Boolean formula in CNF

$$f = \bigwedge_{k=1}^{n} \bigvee_{i \in C_k} x_i$$

with clauses $C_k \subseteq \{1, \dots, d\}$

- Task: Compute number of satisfying assignment
- Known: **#P**-hard
- Plan: reduce #MON-CNF to HYP

#P-hardness of HYP

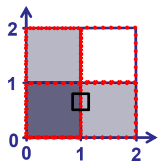
- Take a MON-CNF $f = \bigwedge_{k=1}^{n} \bigvee_{i \in C_k} x_i$
- Consider its negated formula $\bar{f} = \bigvee_{k=1}^{n} \bigwedge_{i \in C_k} \neg x_i$
- For each clause $\bigwedge_{i \in C_k} \neg x_i$

  construct a box $[0, a_1^k] \times \cdots \times [0, a_d^k]$

  with $a_i^k = \begin{cases} 1, & \text{if} \quad i \in C_k \\ 2, & \text{otherwise} \end{cases}$

- Example:

$$\underbrace{\neg x_1}_{C_1 = \{1\}} \vee \underbrace{(\neg x_1 \wedge \neg x_2)}_{C_2 = \{1,2\}} \vee \underbrace{\neg x_2}_{C_3 = \{2\}}$$



#P-hardness of HYP

- This proves that the hypervolume is #P-hard in the number of objectives, i.e., it cannot be solved in time polynomial in the number of objectives (unless P=NP)
- Note that the hypervolume is not hard in the number of boxes, i.e., it can be solved in polytime for constant d

[Bringmann and F., ISAAC 2008, CGTA 2010]

Approximation of the Hypervolume Indicator

Approximation of HYP

- Given: boxes $\{B_1, \ldots, B_n\}$ in d-dimensional space and an error rate ϵ
- Task: Compute $\tilde{V}$ such that

$$\Pr\left[(1-\epsilon)\,V \leq \tilde{V} \leq (1+\epsilon)\,V\right] \geq \frac{3}{4}$$

with $V := \text{VOL}\left(\bigcup_{i=1}^{n} B_i\right)$

- Time: polynomial in n, d and $1/\epsilon$

→ Gives fully polynomial-time randomized approximation scheme (FPRAS)

Approximation of HYP

- Given: boxes $\{B_1, \ldots, B_n\}$ in d-dimensional space and an error rate ϵ
- Algorithm:
  - $V_i := \text{VOL}(B_i)$
  - $S := \sum_{i=1}^{n} V_i$
  - $c(x) :=$ number of boxes B_i with $x \in B_i$
  - loop $\Omega(n^2/\epsilon^2)$ often
    - pick random $i \in \{1, \ldots, n\}$ with prob. $\frac{V_i}{S}$
    - pick random $x \in B_i$ uniformly
    - set $Z_k := \frac{S}{c(x)}$
  - return $\tilde{V}$:=average Z_k

[Bringmann and F., ISAAC 2008, CGTA 2010]

Approximation of HYP

- Given: boxes $\{B_1, \ldots, B_n\}$ in d-dimensional space and an error rate ϵ
- Easy to see that
  - Resulting $\tilde{V}$ has correct expectation
  - It's sufficiently concentrated to be an FPRAS
- Gives runtime $\mathcal{O}(n^2 d/\epsilon^2)$
- Can be improved to $\mathcal{O}(nd/\epsilon^2)$ with self-adjusting algorithm [Karp Luby J.Complexity '85]

[Bringmann and F., ISAAC 2008, CGTA 2010]

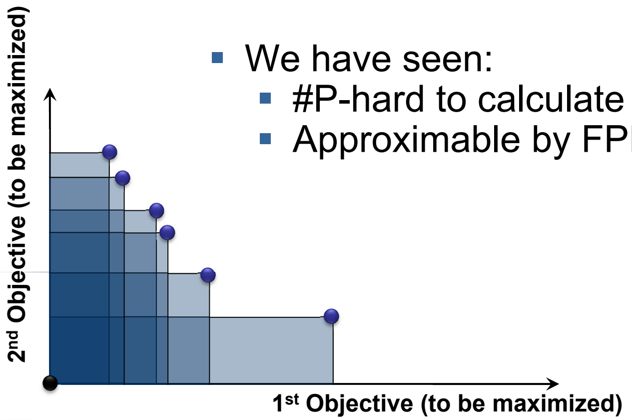
Approximation of HYP

- This shows that the Hypervolume can be approximated efficiently, i.e., in time
  - polynomial in the number of objectives
  - polynomial in the number of solutions
  - polynomial in the approximation quality

Computational Complexity of Hypervolume Contributions

Hypervolume

- Recall Hypervolume $\mathrm{HYP}(M)$

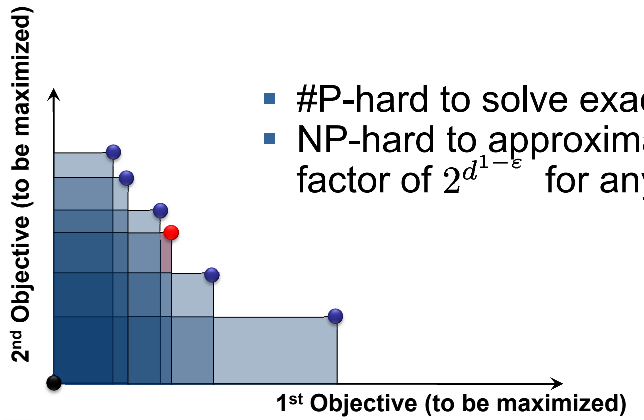


- We have seen:
  - #P-hard to calculate exactly
  - Approximable by FPRAS

Hypervolume Contribution

- Recall the Hypervolume Contribution

$$\mathrm{CON}(M, x) := \mathrm{HYP}(M) - \mathrm{HYP}(M \setminus x)$$

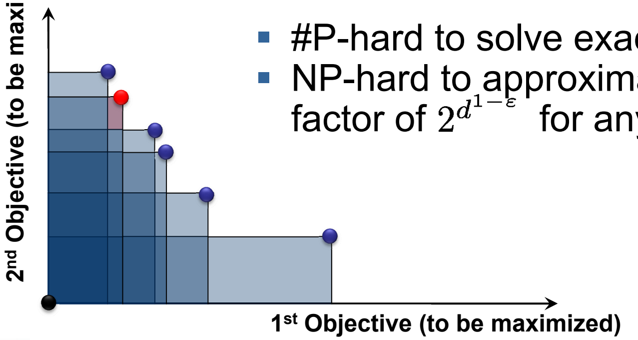


- #P-hard to solve exactly
- NP-hard to approximate by a factor of $2^{d^{1-\varepsilon}}$ for any $\varepsilon > 0$

Hypervolume Contribution

- We are actually interested in the box with the *minimal hypervolume contribution*, i.e.,

$$\mathrm{MINCON}(M) := \min_{x \in M} \mathrm{CON}(M, x)$$

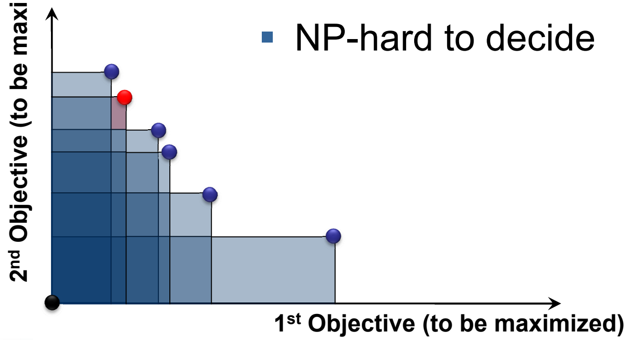


- #P-hard to solve exactly
- NP-hard to approximate by a factor of $2^{d^{1-\varepsilon}}$ for any $\varepsilon > 0$

Least Contributor

- We actually only want to calculate *which* box has the least contribution, i.e.,

$$\mathrm{LC}(M) := \mathrm{argmin}_{x \in M} \mathrm{CON}(M, x)$$

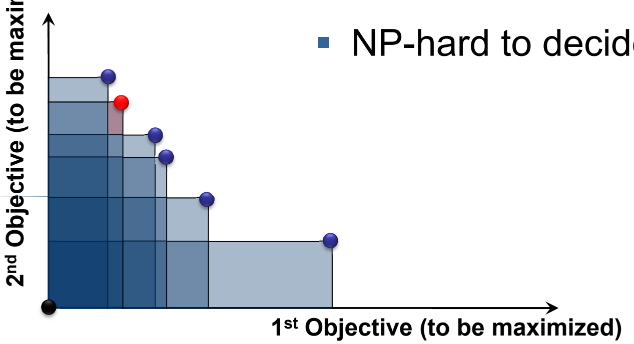


- NP-hard to decide

Approximate Least Contributor

- It usually suffices to find a box with contribution at most (1+ϵ) times the minimal contribution of any box in M, i.e.,

$$\mathrm{CON}(M, \varepsilon\mathrm{LC}(M)) \leq (1+\varepsilon)\,\mathrm{MINCON}(M)$$



- NP-hard to decide

Approximation of Hypervolume Contributions

Approximate Least Contributor

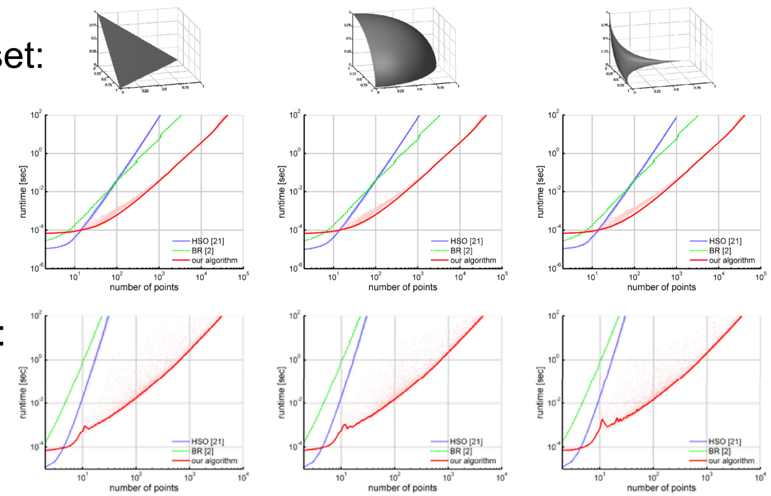
- Unless NP=BPP, there is no worst-case polynomial time algorithm for approximately determining a solution with a small contributor
- But there are several approximation algorithms:
  - [Bringmann and F., EMO 2009]
  - [Bader, Deb, and Zitzler, MCDM 2008] [Bader and Zitzler, ECJ 2010]
  - [Ishibuchi, GECCO 2010]

Approximate Least Contributor

- There is an algorithm for determining a small contributor, i.e., given a set *M*, ϵ>0 and δ>0, with probability 1-δ it finds a box with contribution at most $(1+\epsilon)\,\mathrm{MINCON}(M)$
- Algorithm Idea:
  - Determine for each box the minimal bounding box of the space that is uniquely overlapped by the box
  - Sample randomly in the bounding boxes and count how many random points are uniquely dominated and how many are not
  - Estimate contributions and deviations until least contributor found with good probability

Experimental evaluation

- dataset:
- d=3:
- d=10:



(a) Spherical dataset. (b) Linear dataset. (c) Concave dataset.

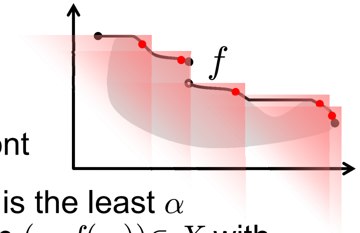
Approximate Least Contributor

- The higher the dimension, the higher the speed-up of the approximation algorithm:
  - For d=100 within 100 seconds, the approximation algorithm solved all problems with n≤6000 while HSO and BR could not solve any problem for n≥6
  - seven solutions on the 100-dimensional linear front take 7 hours with BR, 13 minutes with HSO and 0.5 milliseconds with the approximation algorithm

Finally: Is the Hypervolume the right measure, at all?

What is approximation?

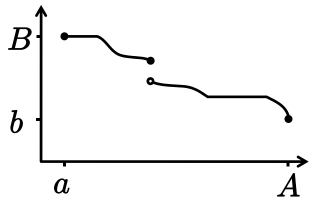
- We restrict our attention to the bi-objective case
- Let f: $D \rightarrow \mathbb{R}$ be a monotonically decreasing function describing the Pareto front
- We look for a (small) set of points $X = \{(x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$ which "nicely" approximates the front



- The *approximation ratio* of a set X is the least α such that for each $x \in D$ there is an $(x_i, f(x_i)) \in X$ with $x \le \alpha\, x_i$ and $f(x) \le \alpha\, f(x_i)$
- **Aim**: Find a set of points with a small approximation ratio.
- **Question**: Is this what we get from maximizing HYP?

"Optimal" approximation

- Let us restrict to Pareto fronts $f \in \mathcal{F}$ where f: $[a,A] \rightarrow [b,B]$ is a monotonically decreasing, upper semi-continuous function with $f(a)=B$ and $f(A)=b$
- Let $\mathcal{X}$ be the set of all populations of a fixed size n
- Let $\alpha(f,X)$ be the approximation ratio achieved by the set X with respect to the front f



- Then the optimal approximation ratio achievable by sets from $\mathcal{X}$ with respect to fronts from the function class $\mathcal{F}$ is

$$\alpha_{\text{OPT}} = \sup_{f \in \mathcal{F}} \inf_{X \in \mathcal{X}} \alpha(f, X)$$

HYP's approximation

- The overall aim of all hypervolume-based algorithms is to find for a front f a population which maximizes HYP:

$$\mathcal{X}^f_{\text{HYP}} = \left\{ X \in \mathcal{X} \;\middle|\; \text{HYP}(X) = \max_{Y \in \mathcal{X}} \text{HYP}(Y) \right\}$$

- This gives a worst-case approximation factor of

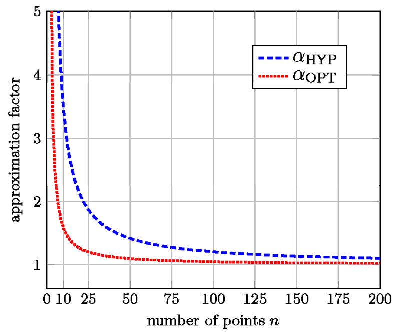
$$\alpha_{\text{HYP}} = \sup_{f \in \mathcal{F}} \sup_{X \in \mathcal{X}^f_{\text{HYP}}} \alpha(f, X)$$

- **Question**: How large is α_{HYP} compared to

$$\alpha_{\text{OPT}} = \sup_{f \in \mathcal{F}} \inf_{X \in \mathcal{X}} \alpha(f, X) \ ?$$

HYP's approximation

- One can prove $\alpha_{\text{OPT}} = 1 + \Theta(1/n)$ and $\alpha_{\text{HYP}} = 1 + \Theta(1/n)$
- Hence maximizing HYP is "asymptotically optimal"
- Plot of bounds for functions f: $[a,A] \to [b,B]$ with $a=b=1$ and $A=B=100$:



HYP's approximation

- One can prove $\alpha_{\text{OPT}} = 1 + \Theta(1/n)$ and $\alpha_{\text{HYP}} = 1 + \Theta(1/n)$
- Hence maximizing HYP is "asymptotically optimal"
- But how large are the constants hidden in the Θ ?
- Let us now for an easier presentation assume that the front is symmetric, that is, $A/a = B/b$
- Then one can prove that

$$\alpha_{\text{OPT}} \approx 1 + \frac{\log(A/a)}{n}$$

and

$$\alpha_{\text{HYP}} \approx 1 + \frac{\sqrt{A/a}}{n}$$

Hence maximizing HYP does ***not*** yield the optimal mult. approximation ratio

Additive approximation

- So what about additive approximation instead?
- Recall: The *multiplicative approximation ratio* of a set X is the least α such that for each $x \in D$ there is an $(x_i, f(x_i)) \in X$ with $x \leq \alpha\, x_i$ and $f(x) \leq \alpha\, f(x_i)$
- Analogously: The *additive approximation ratio* of a set X is the least α such that for each $x \in D$ there is an $(x_i, f(x_i)) \in X$ with $x \leq \alpha + x_i$ and $f(x) \leq \alpha + f(x_i)$
- Then for the additive approximation ratio we can prove that

$$\alpha^{+}_{\text{OPT}} = \frac{A-a}{n}$$

$$\alpha^{+}_{\text{HYP}} \leq \frac{A-a}{n-2}$$

Hence maximizing HYP yields a close-to-optimal additive approximation ratio

Logarithmic Hypervolume

- How to achieve a good multiplicative approximation?
- **Answer**: Logarithm all axes before computing HYP!
- This defines a new indicator whose multiplicative approximation factor is much better:

$$\alpha_{\text{OPT}} \approx 1 + \frac{\log(A/a)}{n}$$

$$\alpha_{\text{HYP}} \approx 1 + \frac{\sqrt{A/a}}{n}$$

$$\alpha_{\text{LOGHYP}} \approx 1 + \frac{\log(A/a)}{n-2}$$

Hence maximizing logHYP yields a close-to-optimal mult. approximation ratio

Executive Summary

- If you want a good additive approximation ratio, you should maximize HYP
- If you want a good multiplicative approximation ratio, you should maximize logHYP

Multi-Objective Models for Single-Objective Problems

Single-Objective vs. Multi-Objective Optimization

General assumption:

- Multi-objective optimization is more (as least as) difficult as single-objective optimization.
- True, if criteria to be optimized are independent.

Examples:

- Minimum Spanning Tree Problem (MST) (in P).
- MST with at least 2 weight functions (NP-hard).
- Shortest paths (SP) (in P).
- SP with at least 2 weight functions (NP-hard).

- Assume that the criteria to be optimized are not independent.
- Question: Can a multi-objective model give better hints for the optimization of single-objective problems by evolutionary algorithms ?
- Yes!!!

Examples:

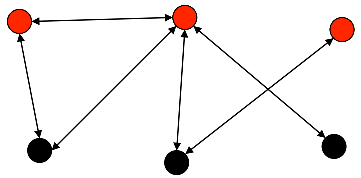
- Minimum Spanning Trees (N., Wegener (2006)).
- (Multi)-Cut Problems (N., Reichel (2008)).
- Helper Objectives (Brockhoff, Friedrich, Hebbinghaus, Klein, N., Zitzler (2007)).

Interest here:

- Theoretical investigations for the Vertex Cover Problem.

The Problem

The Vertex Cover Problem:
Given an undirected graph G=(V,E).



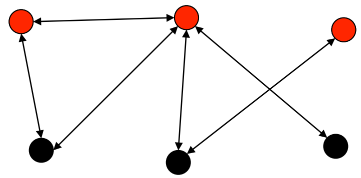
Find a minimum subset of vertices such that each edge is covered at least once.

NP-hard, several 2-approximation algorithms.

Simple single-objective evolutionary algorithms fail!!!

The Problem

The Vertex Cover Problem:
Given an undirected graph G=(V,E).



Integer Linear Program (ILP)

$$\min \sum_{i=1}^{n} x_i$$
$$\text{s.t. } x_i + x_j \geq 1 \quad \forall \{i,j\} \in E$$
$$x_i \in \{0,1\}$$

Linear Program (LP)

$$\min \sum_{i=1}^{n} x_i$$
$$\text{s.t. } x_i + x_j \geq 1 \quad \forall \{i,j\} \in E$$
$$x_i \in [0,1]$$

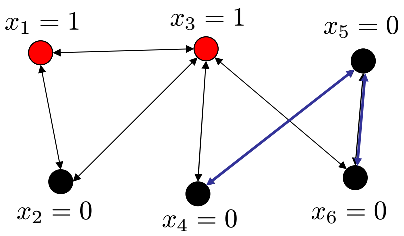
Decision problem:
Is there a set of vertices of size at most k covering all edges?

Fixed parameter algorithm runs in time $O(1{:}2738^k + kn)$ (Chen et al 2006)

Our parameter: Value of an optimal solution (OPT)

Evolutionary Algorithm

Representation: Bitstrings of length n



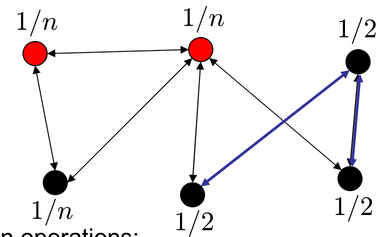
Minimize fitness function:

$$f_1(x) = (|x|_1, |U(x)|)$$
$$f_1(x) = (2, 2)$$

$$f_2(x) = (|x|_1, LP(x))$$
$$f_2(x) = (2, 1)$$

$U(x)$: Edges not covered by x
$G(x) = G(V, U(x))$
$LP(x)$: value of LP applied to $G(x)$

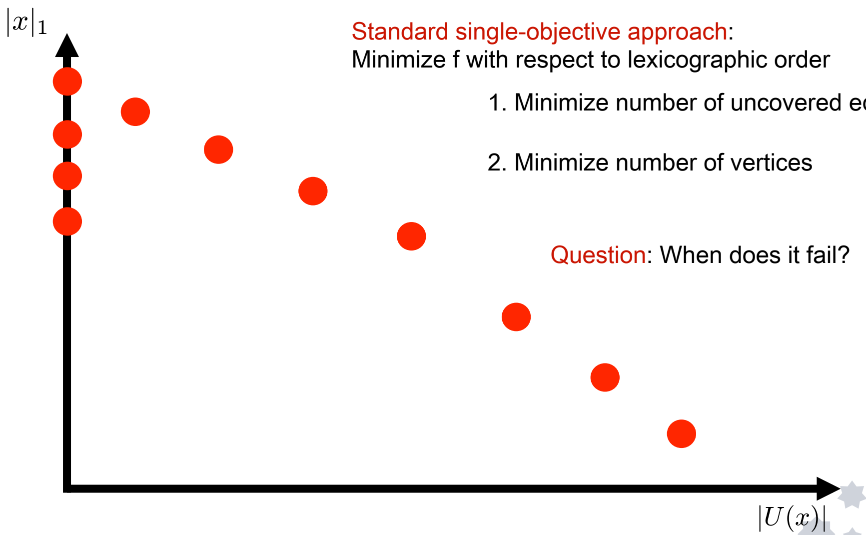
Evolutionary Algorithm



Two mutation operations:

1. Standard bit mutation with probability $1/n$
2. Mutation probability $1/2$ for vertices adjacent to edges of $U(x)$. Otherwise mutation probability $1/n$.

Decide uniformly at random which operator to use in next iteration



Standard single-objective approach:
Minimize f with respect to lexicographic order

1. Minimize number of uncovered edges
2. Minimize number of vertices

Question: When does it fail?

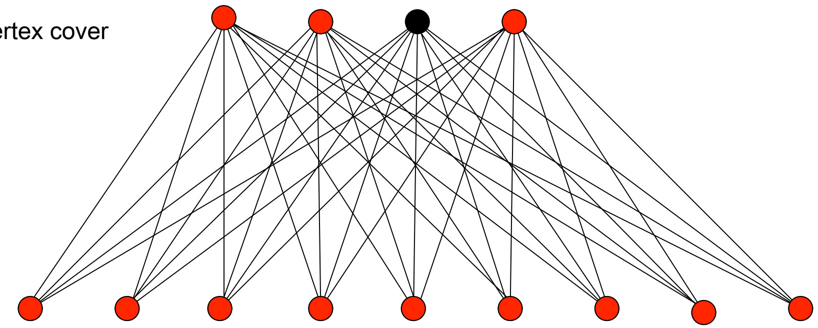
(1+1) EA and Vertex Cover Problem

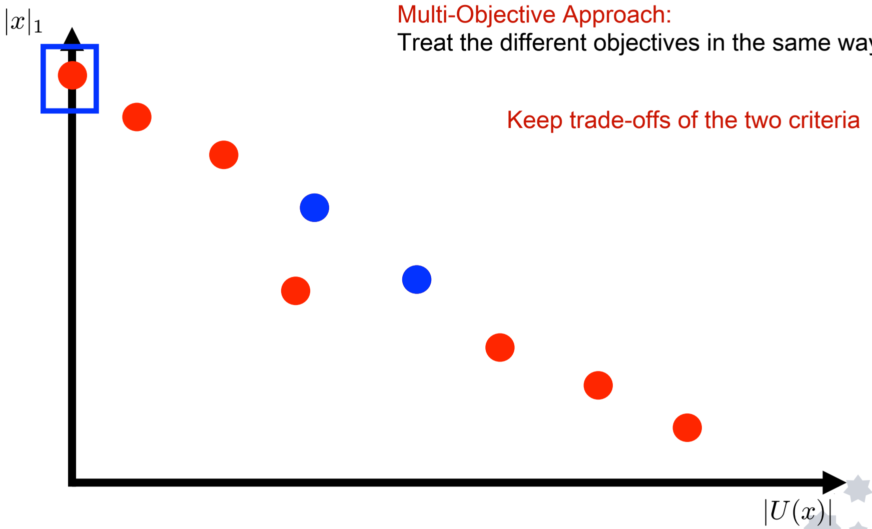
Friedrich, He, Hebbinghaus, N., Witt (2007)

Exponential expected optimization time

Approximation may be arbitrary bad

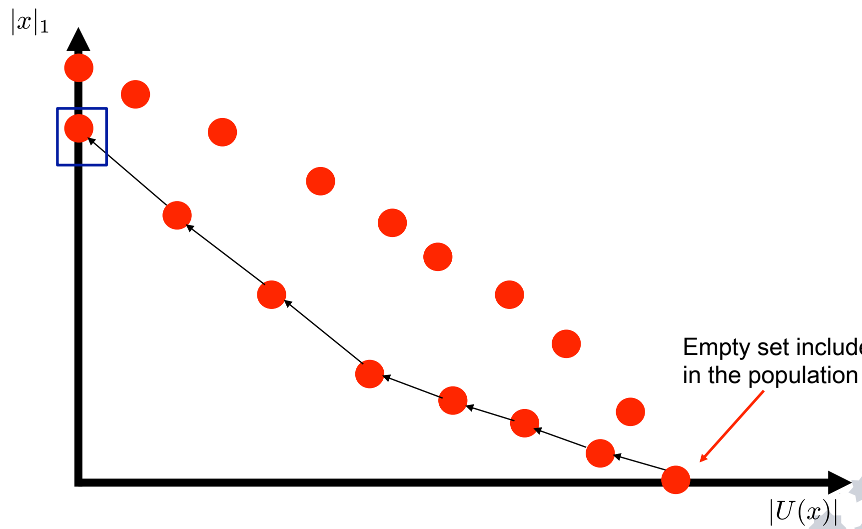
vertex cover

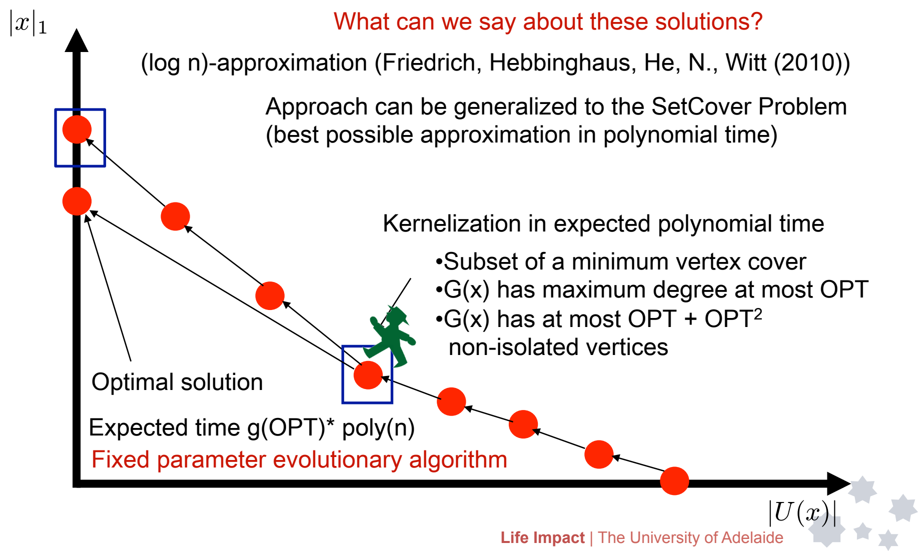


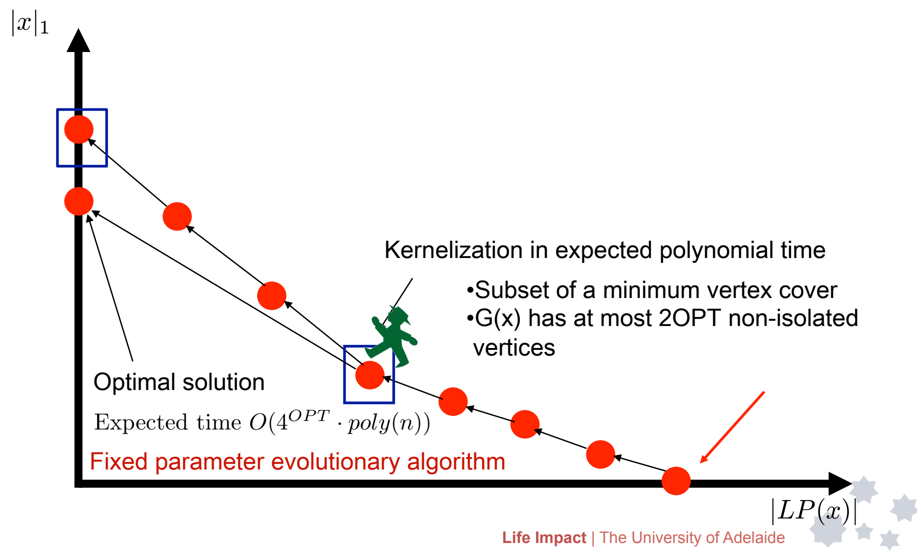


Multi-Objective Approach:
Treat the different objectives in the same way

Keep trade-offs of the two criteria







Linear Programming

Combination with Linear Programming

- LP-relaxation is half integral, i.e.

$$x_i \in \{0, 1/2, 1\}, 1 \leq i \leq n$$

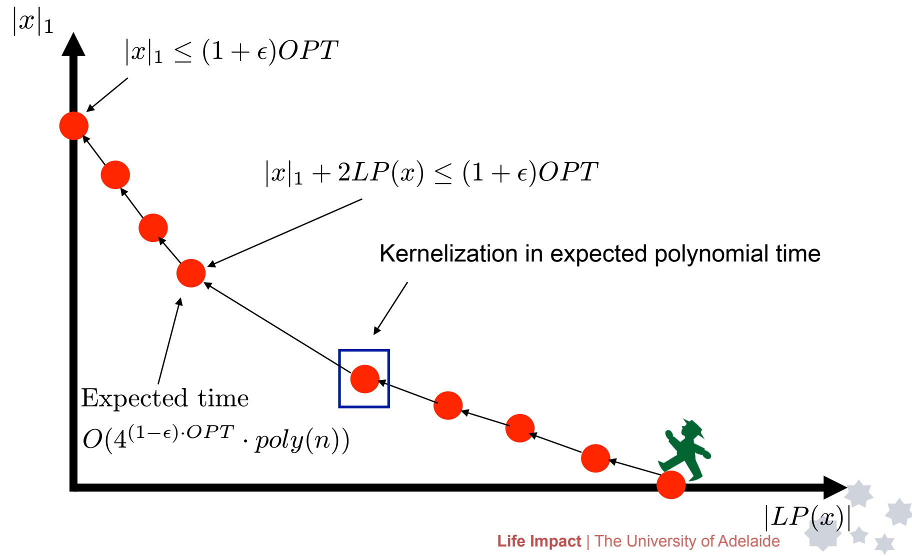
Theorem (Nemhauser, Trotter (1975)):
Let x^* be an optimal solution of the LP. Then there is a minimum vertex cover that contains all vertices v_i where $x_i^* = 1$.

Lemma:
All search points x with $LP(x) = LP(0^n) - |x|_1$ are Pareto optimal. They can be extended to minimum vertex cover by selecting additional vertices.

Can we also say something about approximations?

Life Impact | The University of Adelaide

Approximations



Summary Multi-objective Models

- Multi-Objective models can be helpful for solving single-objective optimization problems.
- Give additional hints for the search process.
- Example study for the NP-hard vertex cover problem.
- Single-objective approach fails.
- Good approximations for multi-objective EAs.
- Fixed-parameter evolutionary algorithms.

Thank you!