Fixed-Parameter Evolutionary Algorithms

Frank Neumann

School of Computer Science

University of Adelaide

Joint work with Stefan Kratsch (U Utrecht), Per Kristian Lehre (DTU Informatics), Pietro S. Oliveto (U Birmingham)

Computational Complexity of Evolutionary Algorithms

Theory of Evolutionary Algorithms

- Evolutionary algorithms are successful for many complex optimization problems.
- Rely on random decisions $\Rightarrow$ randomized algorithms
- Goal: Understand how and why they work
- Study the computational complexity of these algorithms on prominent examples

Runtime Analysis

Black Box Scenario:

- Measure the runtime T by the number of fitness evaluations.
- Studies consider time in dependence of the input to reach
  - An optimal solution.
  - A good approximation.

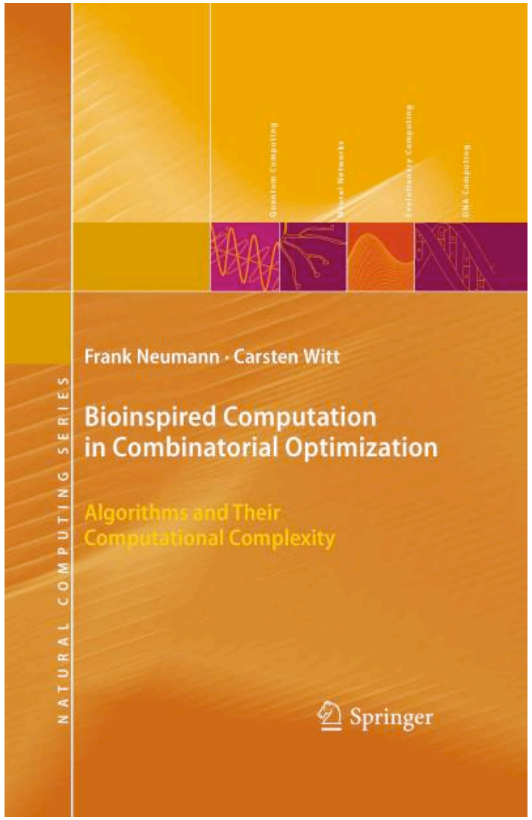
Interest:

- Expected number of fitness evaluations E[T].

Combinatorial Optimization

Analysis of runtime and approximation quality on combinatorial optimization problems, e. g.,

- sorting problems
- shortest path problems,
- subsequence problems,
- vertex cover,
- Eulerian cycles,
- minimum (multi)-cuts,
- minimum spanning trees,
- maximum matchings,
- partition problem,
- set cover problem,
- . . .



Book available at www.bioinspiredcomputation.com

Understand the behavior of bio-inspired computation on “natural” examples

Fixed Parameter Evolutionary Algorithms

- What makes a problem hard for an EA?

- Consider an additional parameter k to measure "hardness" of an instance
- Fixed parameter algorithm runs in time O(f(k) poly(n))

- Fixed parameter evolutionary algorithm runs in expected time O(f(k) poly(n))

- Consider maximum leaf spanning trees and minimum vertex covers as initial examples

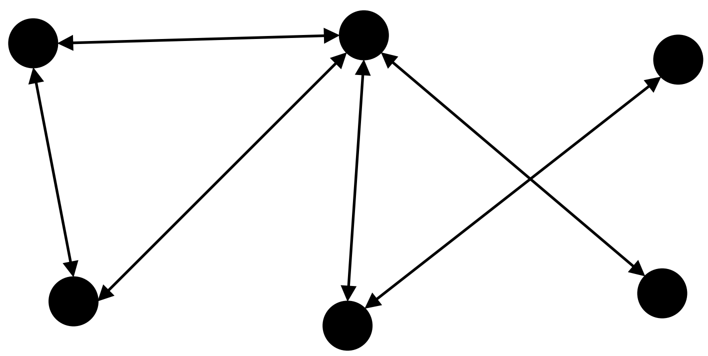
Life Impact | The University of Adelaide

Maximum Leaf Spanning Trees

The Problem

The Maximum Leaf Spanning Tree Problem:

Given an undirected connected graph G=(V,E).

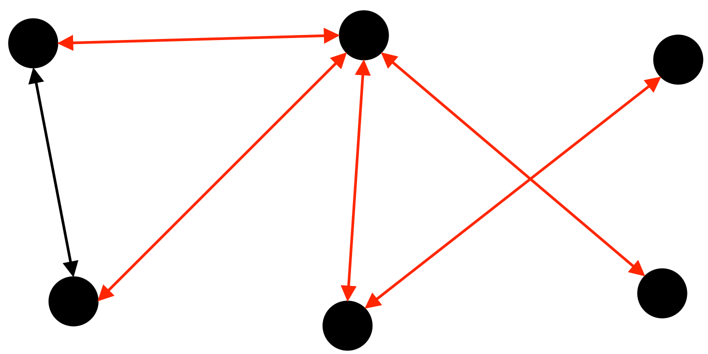


Find a spanning tree with a maximum number of leaves.

The Problem

The Maximum Leaf Spanning Tree Problem:

Given an undirected connected graph G=(V,E).



Find a spanning tree with a maximum number of leaves.

NP-hard, different classical FPT-studies

Two Evolutionary Algorithms

Algorithm 1 (Generic (1+1) EA)

1. *Choose a spanning tree of T uniformly at random.*
2. *Produce T' by swapping each edge of T independently with probability $1/m$.*
3. *If T' is a tree and $\ell(T') \geq \ell(T)$, set $T := T'$.*
4. *Go to 2.*

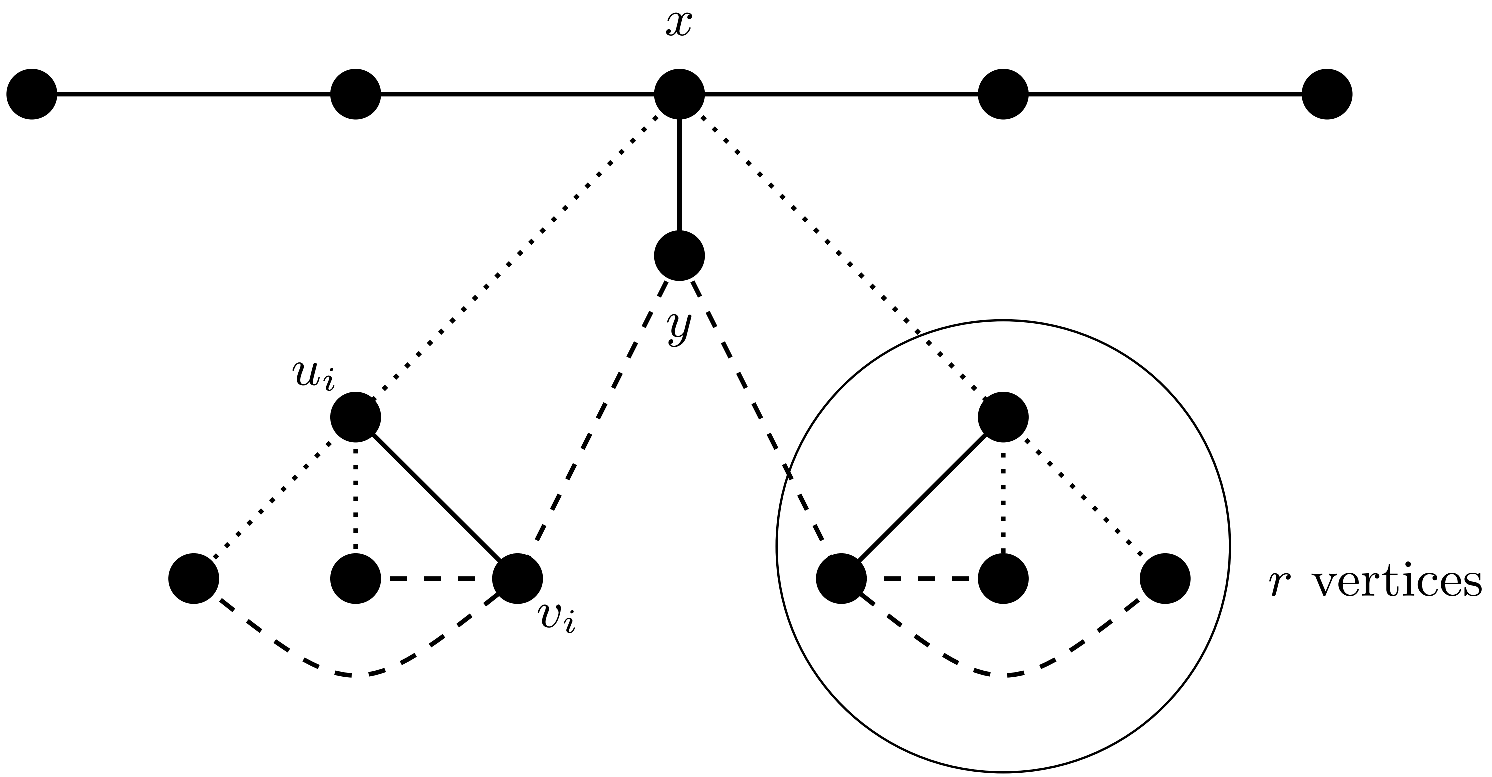
Algorithm 2 (Tree-Based (1+1) EA)

1. *Choose an arbitrary spanning tree T of G.*
2. *Choose S according to a Poisson distribution with parameter $\lambda = 1$ and perform sequentially S random edge-exchange operations to obtain a spanning tree T'. A random exchange operation applied to a spanning tree $\tilde{T}$ chooses an edge $e \in E \setminus \tilde{T}$ uniformly at random. The edge e is inserted and one randomly chosen edge of the cycle in $\tilde{T} \cup \{e\}$ is deleted.*
3. *If $\ell(T') \geq \ell(T)$, set $T := T'$.*
4. *Go to 2.*

Does the mutation operator make the difference between FPT and non-FPT runtime?

Frank Neumann

Local Optimum



Lower Bounds

Theorem 1. *The expected optimization time of Generic (1+1) EA on G_{loc} is lower bounded by $\left(\frac{m}{c}\right)^{2(r-2)}$ where c is an appropriate constant.*

Theorem 2. *The expected optimization time of Tree-Based (1+1) EA on G_{loc} is lower bounded by $(\frac{r-2}{c})^{r-2}$ where c is an appropriate constant.*

Idea for lower bounds:

Both algorithms may get stuck in local optimum.

For the Generic (1+1) EA it is less likely to escape local optimum as it often flips edges on the path.

Structural insights

Similar to Fellows, Lokshtanov, Misra, Mnich, Rosamond, Saurabh (2009)

Lemma 2. *Any connected graph G on n nodes and with a maximum number of k leaves in any spanning tree has at most $n+5k^2-7k$ edges and at most $10k-14$ nodes of degree at least three.*

Proof idea:

- Let T be a maximum leaf spanning tree with k leaves.
- Let P_0 be the set of all leaves and all nodes of degree at least three in T.
- Let P be the set of nodes that are of distance at most 2 (w. r. t. to T) to any node in P_0 and let Q be the set of remaining nodes.
- Show: all nodes of Q have degree 2 in G.
- Implies: Number of nodes in P is at most 10k-14
- No node has degree greater than k which implies bound on the number of edges.

Upper Bound

Theorem 3. *If the maximal number of leaf nodes in any spanning tree of G is k, then Algorithm 2 finds an optimal solution in expected time $O(2^{15k^2 \log k})$.*

Proof Idea:

- We call an edge distinguished if it is adjacent to at least one node of degree at least 3 in G.
- Number of distinguished edges on any cycle is at most 20k-28.
- Total number of edges in G: m <= n+5k^2–7k
- Probability to introduce a specific non-chosen distinguished edge is at least $1/(m-(n-1)) \geq 1/5k^2$
- Show: Length of created cycle is at most 20k.
- Probability to remove edge of the cycle that does not belong to optimal solution is at least $1/20k$

Proof Upper bound (continued)

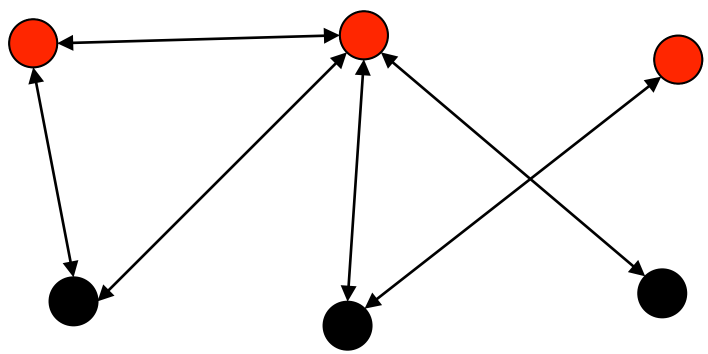
- Probability to obtain a specific spanning tree that can be obtained by an edge-swap is at least
  $$1/(20k \cdot 5k^2)$$
- Probability to produce optimal spanning tree which has distance $r \leq 5k^2$ is at least
  $$r! \cdot \frac{1}{er!} \cdot \left(\frac{1}{5k^2} \cdot \frac{1}{20k}\right)^r \geq \frac{1}{e}\left(\frac{1}{100k^3}\right)^{5k^2} \geq \frac{1}{e}\left(\frac{1}{100}\right)^{5k^2}\left(\frac{1}{k}\right)^{3 \cdot 5k^2},$$
- Implies that expected time to get maximum leaf spanning tree is at most $O(2^{15k^2 \log k})$ $\square$

The Minimum 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.

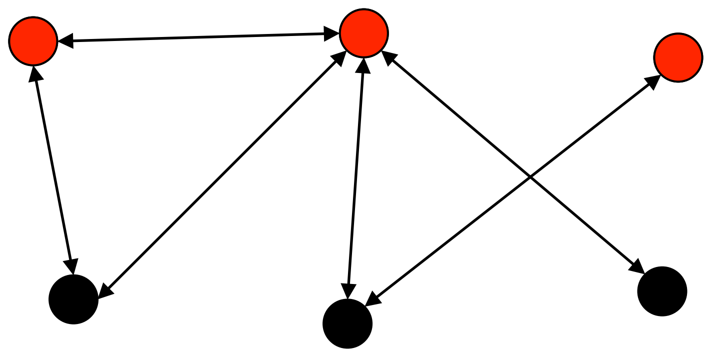
NP-hard, several 2-approximation algorithms.

Simple single-objective evolutionary algorithms fail!!!

The Problem

The Vertex Cover Problem:

Given an undirected graph G=(V,E).



Integer Linear Program (ILP)

$$\begin{aligned} &\min \textstyle\sum_{i=1}^{n} x_i \\ &\text{s.t. } x_i + x_j \geq 1 \qquad \forall \{i,j\} \in E \\ &x_i \in \{0,1\} \end{aligned}$$

Linear Program (LP)

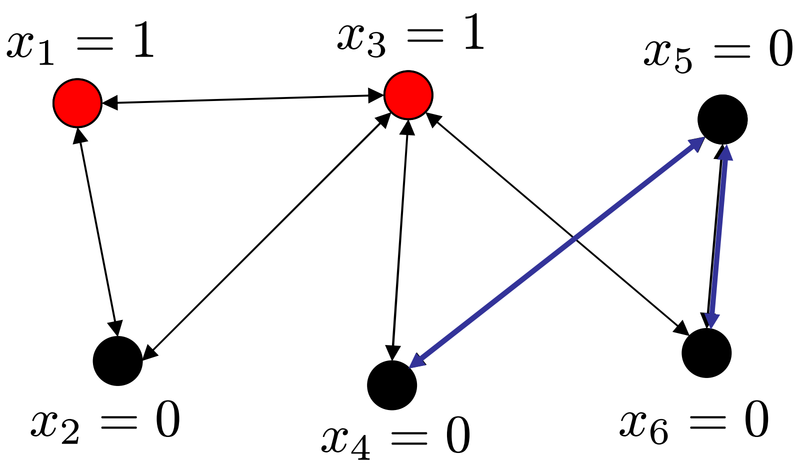
$$\begin{aligned} &\min \textstyle\sum_{i=1}^{n} x_i \\ &\text{s.t. } x_i + x_j \geq 1 \qquad \forall \{i,j\} \in E \\ &x_i \in [0,1] \end{aligned}$$

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

Our parameter: Value of an optimal solution (OPT)

Evolutionary Algorithm

Representation: Bitstrings of length n



Minimize fitness function:

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

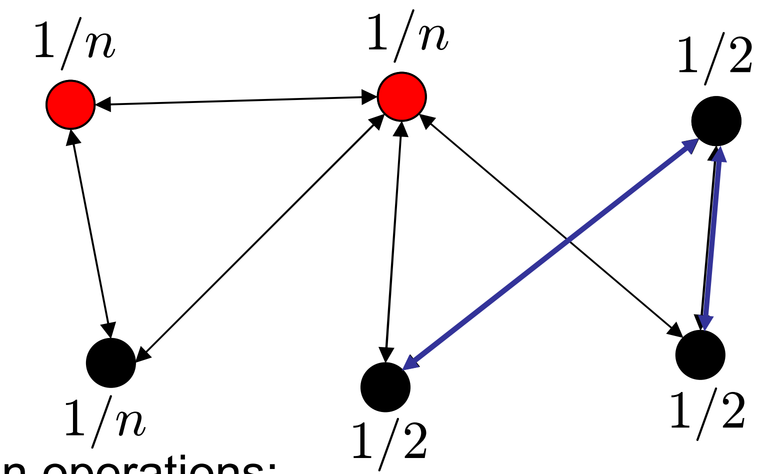
$$f_2(x) = (|x|_1, LP(x))$$
$$f_2(x) = (2, 1)$$

$U(x)$: 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:

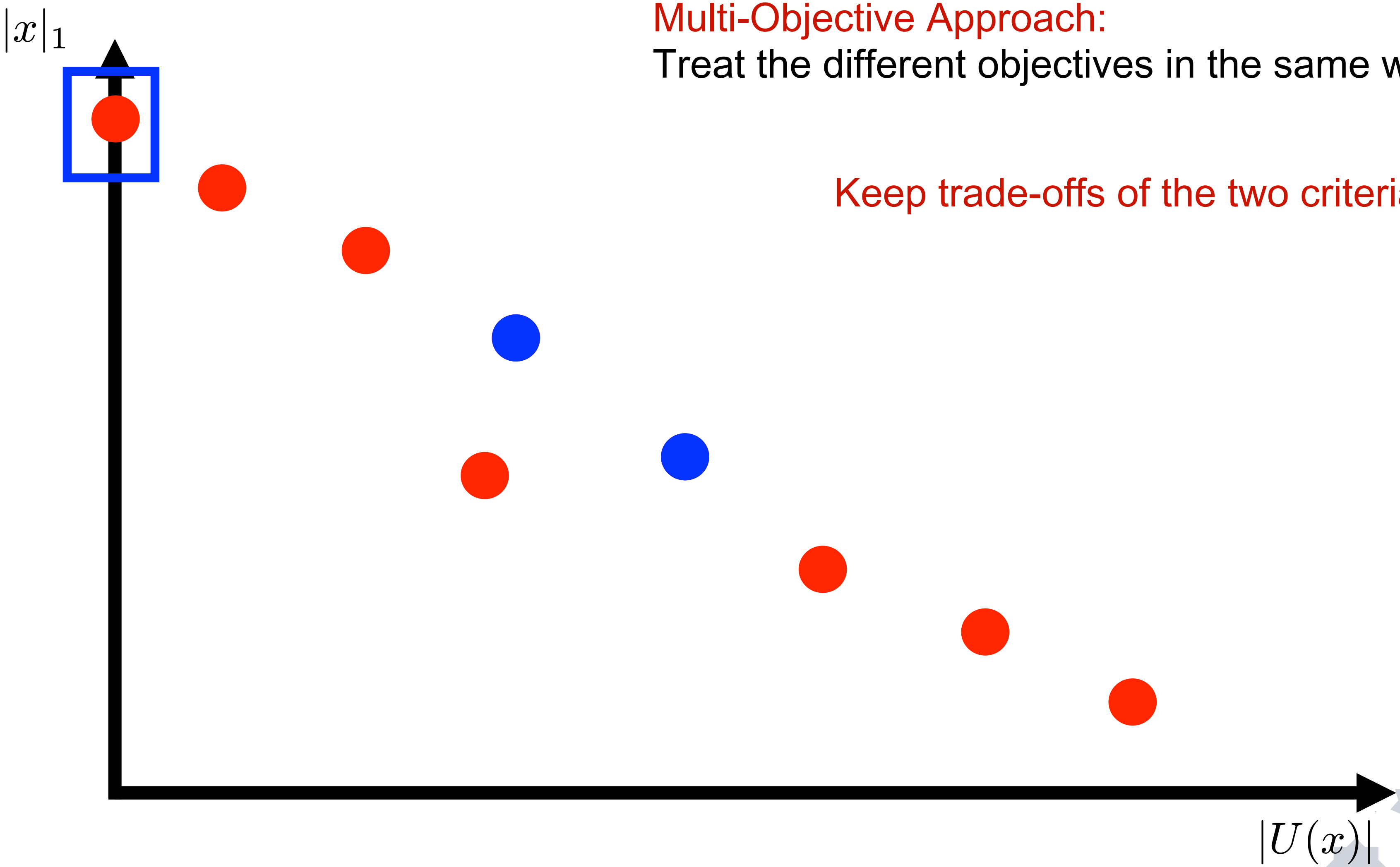
1. Standard bit mutation with probability $1/n$

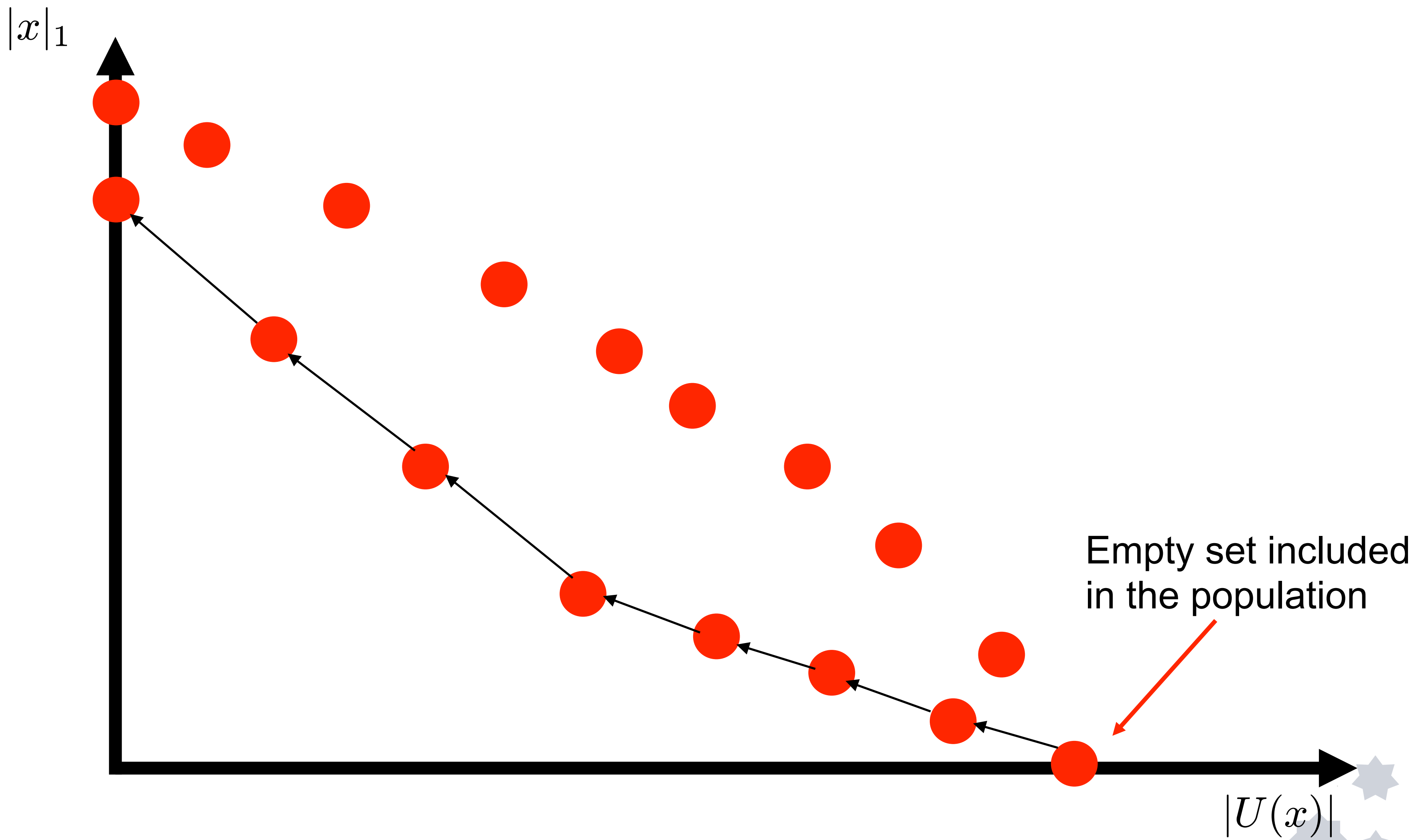
2. Mutation probability $1/2$ for vertices adjacent to edges of $U(x)$. Otherwise mutation probability $1/n$.

Decide uniformly at random which operator to use in next iteration

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

Keep trade-offs of the two criteria




$|x|_1$
Empty set included
in the population
$|U(x)|$

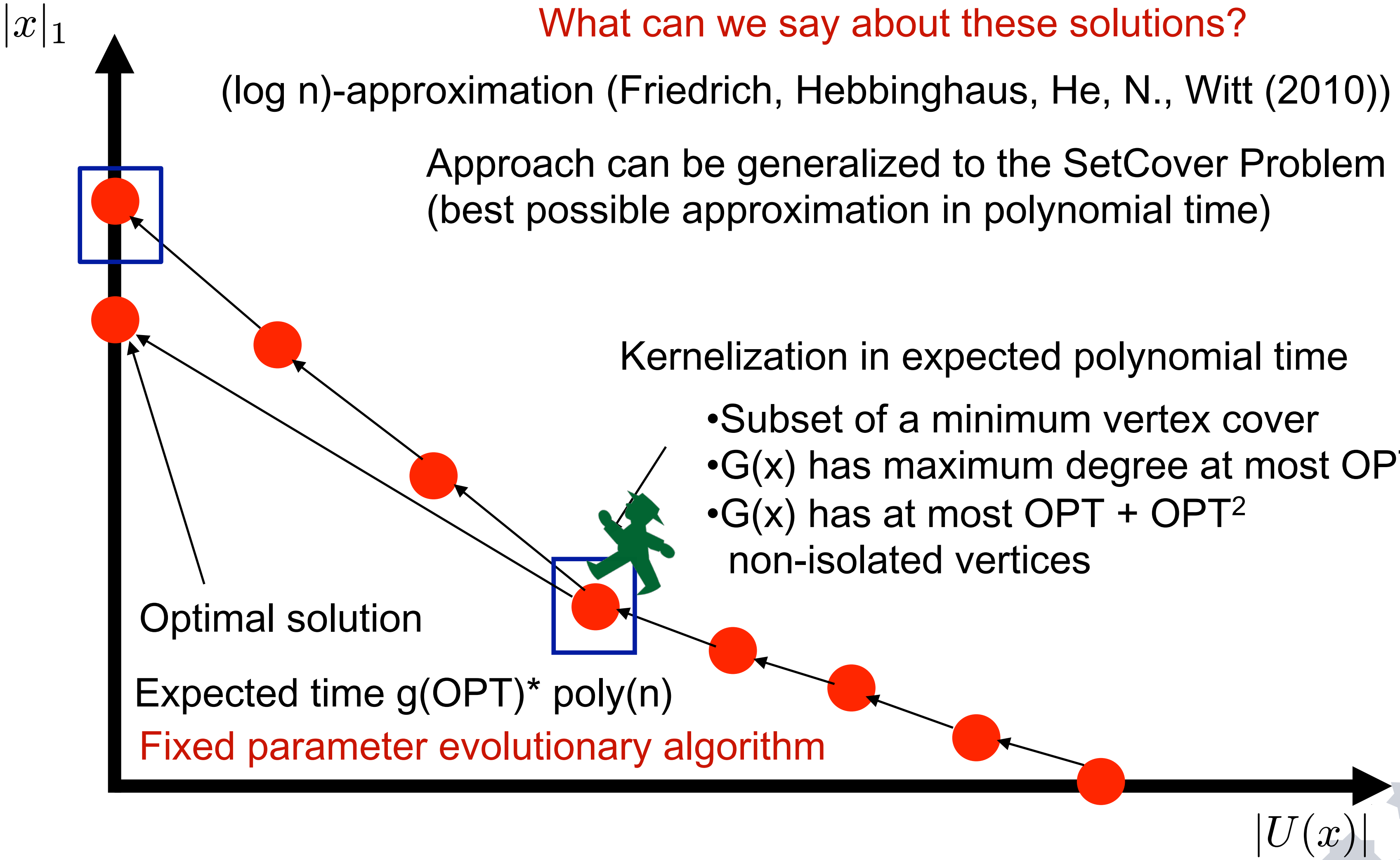
What can we say about these solutions?

(log n)-approximation (Friedrich, Hebbinghaus, He, N., Witt (2010))

Approach can be generalized to the SetCover Problem
(best possible approximation in polynomial time)

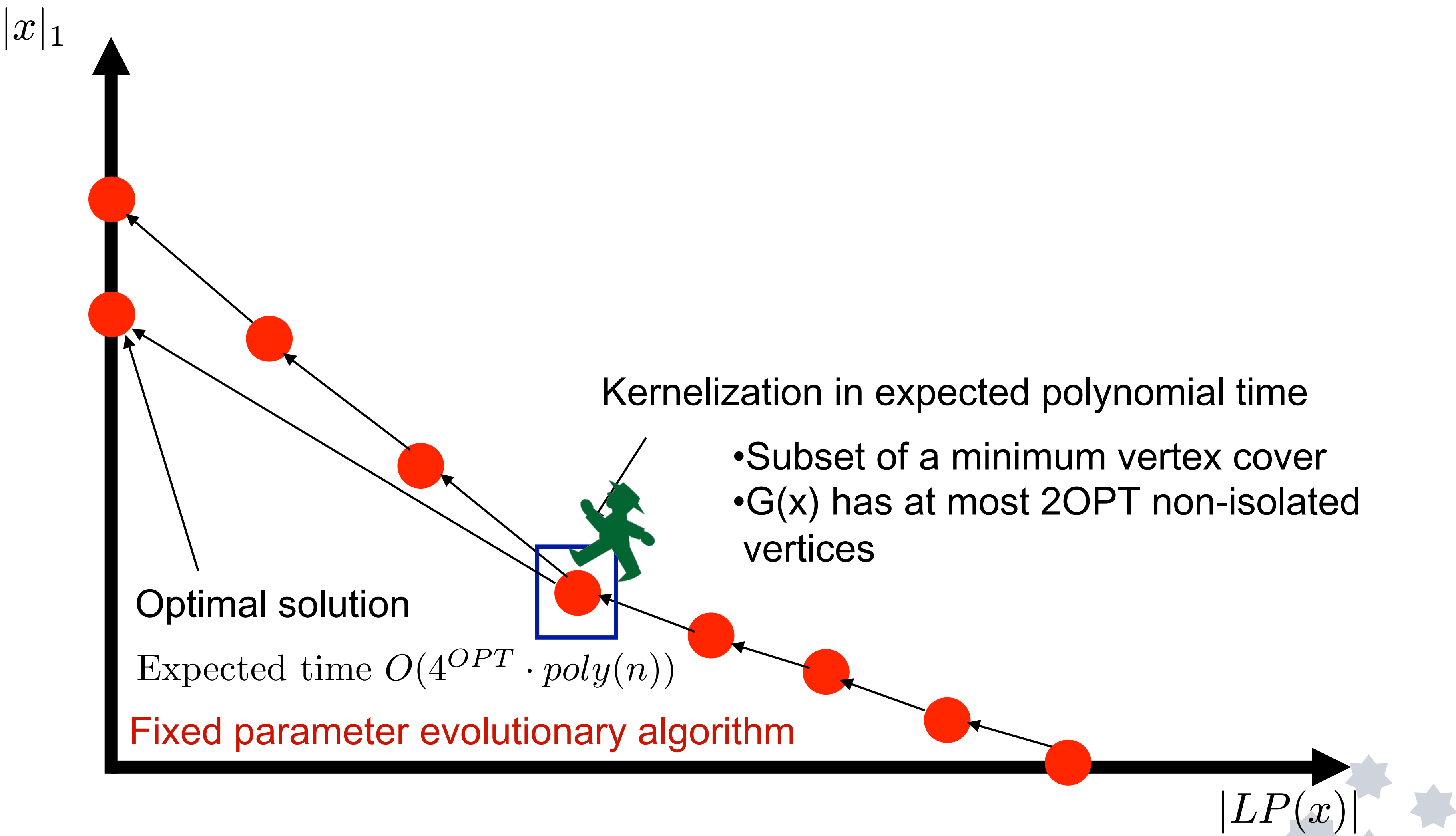
Kernelization in expected polynomial time

- Subset of a minimum vertex cover
- G(x) has maximum degree at most OPT
- G(x) has at most OPT + OPT^2 non-isolated vertices



Expected time g(OPT)* poly(n)

Fixed parameter evolutionary algorithm



$|x|_1$
Kernelization in expected polynomial time
•Subset of a minimum vertex cover
•G(x) has at most 2OPT non-isolated vertices
Optimal solution
Expected time $O(4^{OPT} \cdot poly(n))$
Fixed parameter evolutionary algorithm
$|LP(x)|$

Linear Programming

Combination with Linear Programming

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

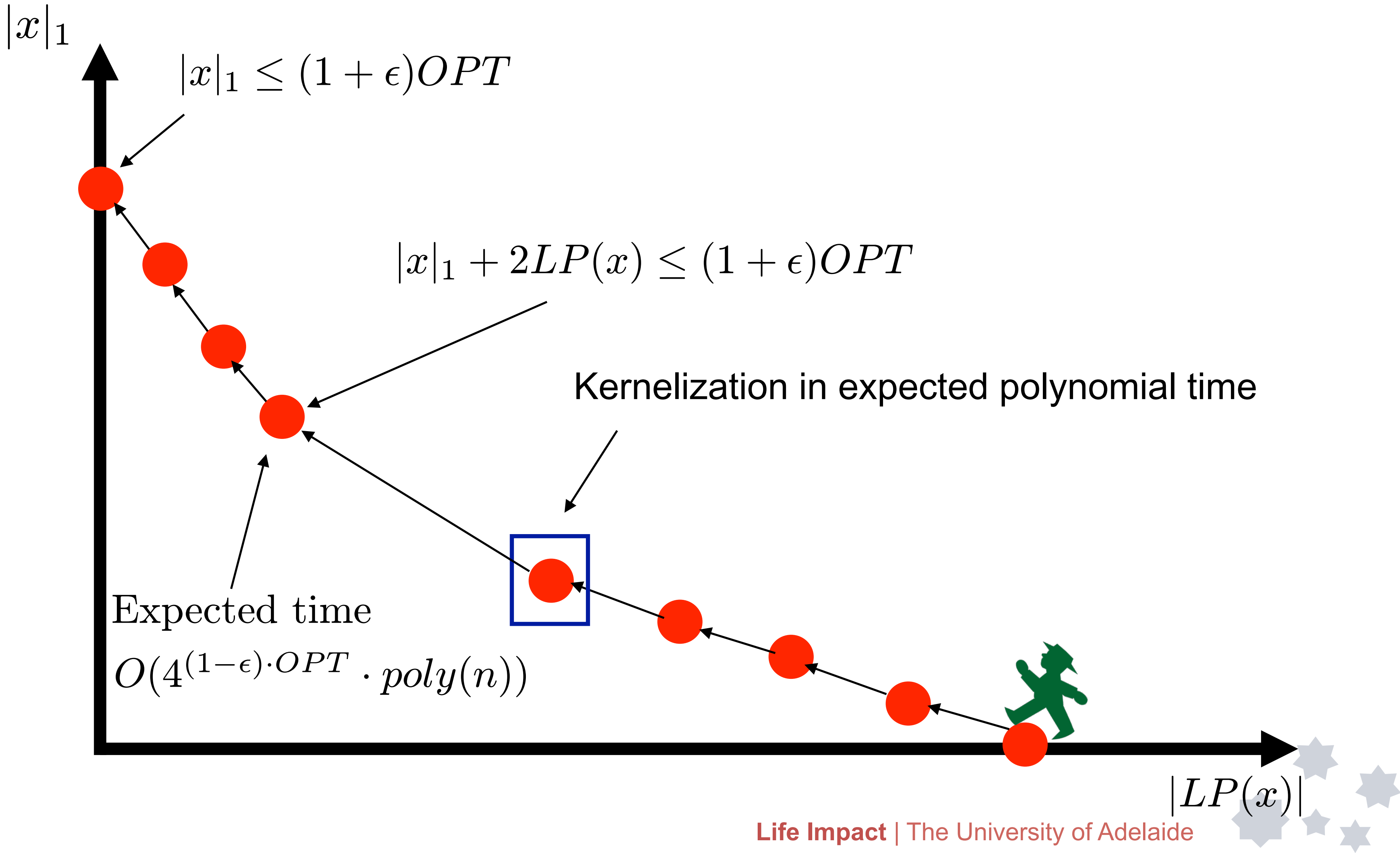
  $x_i \in \{0, 1/2, 1\}, 1 \leq i \leq n$

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?

Approximations



Summary

- Evolutionary algorithms are successful for many complex optimization problems.
- Goal is to get a better theoretical understanding.
- There are some nice results for combinatorial optimization.
- Using parameterized analysis looks very promising.

Thank you!