Introduction to the Complexity Analysis of Randomized Search Heuristics

Dirk Sudholt

CERCIA, University of Birmingham

ThRaSH 2011

Overview

Randomized Search Heuristics

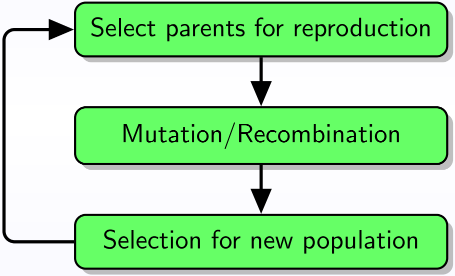
Metaheuristics

- evolutionary algorithms
- simulated annealing
- swarm intelligence
- artificial immune systems
- ...

Benefits

- applicable when problem is not well understood (black-box setting)
- lack of time, money, or expertise to design a tailored algorithm
- usually easy to implement and easy to apply
- robust and often surprisingly successful

Scheme of an Evolutionary Algorithm



Motivation

Goals

- understand how metaheuristics work
- get to know their capabilities and limitations
- solid theoretical foundation
- design better metaheuristics

Motivation

Goals

- understand how metaheuristics work
- get to know their capabilities and limitations
- solid theoretical foundation
- design better metaheuristics

What we are looking for

Bounds on the (expected) time until a metaheuristic finds a global optimum for a given problem.

Motivation

Goals

- understand how metaheuristics work
- get to know their capabilities and limitations
- solid theoretical foundation
- design better metaheuristics

What we are looking for

Bounds on the (expected) time until a metaheuristic finds a global optimum for a given problem.

Notion of "time"

- number of evaluations of the objective function
- number of iterations / generations

Approach

Tools from the analysis of randomized algorithms

- tail inequalities (Markov, Chernoff, ...)
- Markov chain theory
- random walks, stochastic processes
- asymptotic notation
- amortized analysis
- ...

Approach

Tools from the analysis of randomized algorithms

- tail inequalities (Markov, Chernoff, ...)
- Markov chain theory
- random walks, stochastic processes
- asymptotic notation
- amortized analysis
- ...

Challenge

- Metaheuristics often not designed to support an analysis

Approach

Tools from the analysis of randomized algorithms

- tail inequalities (Markov, Chernoff, ...)
- Markov chain theory
- random walks, stochastic processes
- asymptotic notation
- amortized analysis
- ...

Challenge

- Metaheuristics often not designed to support an analysis

Perspective

- Classical algorithms theory: problem $\longrightarrow$ algorithms
- Randomized search heuristics: algorithm (paradigm) $\longrightarrow$ problems

The (1+1) Evolutionary Algorithm

(1+1) EA for maximization of $f\colon \{0,1\}^n \to \mathbb{R}$

Choose $x \in \{0,1\}^n$ uniformly at random.
repeat forever
  Create y by flipping each bit in x independently with probability $1/n$.
  if $f(y) \geq f(x)$ **then** $x := y$.

The (1+1) Evolutionary Algorithm

(1+1) EA for maximization of $f\colon \{0,1\}^n \to \mathbb{R}$

Choose $x \in \{0,1\}^n$ uniformly at random.
repeat forever
  Create y by flipping each bit in x independently with probability $1/n$.
  if $f(y) \geq f(x)$ **then** $x := y$.

Properties:

- "population" of size 1, no crossover
- stochastic hill-climber
- still reflects basic principle of mutation and selection

Overview

Runtime Analyses

Design aspects

- parent populations
- offspring populations
- crossover vs. mutation
- population diversity
- operator bias
- coping with obstacles: paths, plateaus, multimodality, ...
- ...

Areas

- multiobjective optimization
- hybridization
- parallelization
- stochastic optimization

One Size Fits All...

Evolutionary algorithms like the simple (1+1) EA ...

One Size Fits All. . .

Evolutionary algorithms like the simple (1+1) EA . . .

. . . solve in expected poly-time

- sorting (maximize sortedness) [Scharnow, Tinnefeld, Wegener, 2004]
- shortest paths [Scharnow, Tinnefeld, Wegener, 2004]
- minimum spanning trees [Neumann and Wegener, 2007]
- Matroid optimization [Reichel and Skutella, 2007]
- Eulerian cycles [Neumann, 2008 and follow-up work]

One Size Fits All...

Evolutionary algorithms like the simple (1+1) EA ...

...solve in expected poly-time

- sorting (maximize sortedness) [Scharnow, Tinnefeld, Wegener, 2004]
- shortest paths [Scharnow, Tinnefeld, Wegener, 2004]
- minimum spanning trees [Neumann and Wegener, 2007]
- Matroid optimization [Reichel and Skutella, 2007]
- Eulerian cycles [Neumann, 2008 and follow-up work]

...are poly-time randomized approximation schemes

- maximum matchings [Giel and Wegener, 2003]
- PARTITION/makespan scheduling [Witt, 2005]
- multiobjective shortest paths [Horoba, 2010]

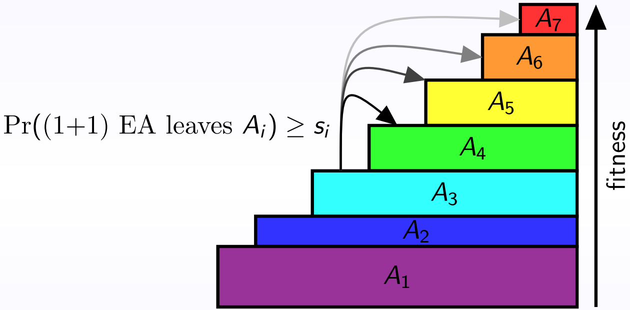
One Size Fits All (continued)

EAs can mimic behavior of

- tailored algorithms
- dynamic programming algorithms [Doerr, Eremeev, Horoba, Neumann, Theile, 2009]
- greedy algorithms
- fixed-parameter tractable algorithms [Kratsch and Neumann, 2009]

Overview

Fitness-level Method for the (1+1) EA



Expected optimization time of (1+1) EA at most $\sum_{i=1}^{m-1} \frac{1}{s_i}$.

Fitness-Level Method: Example

$\textsc{OneMax}(x) := \sum_{i=1}^{n} x_i$

Fitness-Level Method: Example

$\textsc{OneMax}(x) := \sum_{i=1}^{n} x_i$

$A_i := \{x \mid \textsc{OneMax}(x) = i\}.$

Fitness-Level Method: Example

$\textsc{OneMax}(x) := \sum_{i=1}^{n} x_i$

$A_i := \{x \mid \textsc{OneMax}(x) = i\}$.

$$s_i \geq (n-i) \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{n-i}{en}$$

Fitness-Level Method: Example

$\textsc{OneMax}(x) := \sum_{i=1}^{n} x_i$

$A_i := \{x \mid \textsc{OneMax}(x) = i\}$.

$$s_i \geq (n-i) \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{n-i}{en}$$

Bound on the expected optimization time of (1+1) EA

$$\sum_{i=0}^{n-1} \frac{en}{n-i} = en \sum_{i=1}^{n} \frac{1}{i} \leq en \ln(en)$$

(1+1) EA for Minimum Spanning Trees

Given a weighted graph $G = (V, E, w)$ with $n := |V|, m := |E|$.
$x \in \{0, 1\}^m$ encodes selection of edges.

(1+1) EA for Minimum Spanning Trees

Given a weighted graph $G = (V, E, w)$ with $n := |V|, m := |E|$.
$x \in \{0, 1\}^m$ encodes selection of edges.

$$f(x) := (\#\text{components}(x) - 1) \cdot n^3 w_{\max} + n w_{\max} \left| n - 1 - \sum_{i=1}^{m} x_i \right| + \sum_{i=1}^{m} w_i x_i$$

(1+1) EA for Minimum Spanning Trees

Given a weighted graph $G = (V, E, w)$ with $n := |V|, m := |E|$.
$x \in \{0, 1\}^m$ encodes selection of edges.

$$f(x) := (\#\text{components}(x) - 1) \cdot n^3 w_{\max} + n w_{\max} \left| n - 1 - \sum_{i=1}^{m} x_i \right| + \sum_{i=1}^{m} w_i x_i$$

Theorem (Neumann and Wegener, 2007)

The expected time until the (1+1) EA *constructs a minimum spanning tree is* $O(m^2(\log n + \log w_{\max}))$.

(1+1) EA for Minimum Spanning Trees

Given a weighted graph $G = (V, E, w)$ with $n := |V|, m := |E|$.
$x \in \{0,1\}^m$ encodes selection of edges.

$$f(x) := (\#\text{components}(x) - 1) \cdot n^3 w_{\max} + n w_{\max} \left| n - 1 - \sum_{i=1}^{m} x_i \right| + \sum_{i=1}^{m} w_i x_i$$

Theorem (Neumann and Wegener, 2007)

The expected time until the (1+1) EA *constructs a minimum spanning tree is* $O(m^2(\log n + \log w_{\max}))$.

Analysis of Typical Runs

Phase 1: Find some connected graph $\rightsquigarrow O(m \log m)$.

(1+1) EA for Minimum Spanning Trees

Given a weighted graph $G = (V, E, w)$ with $n := |V|, m := |E|$.
$x \in \{0,1\}^m$ encodes selection of edges.

$$f(x) := (\#\text{components}(x) - 1) \cdot n^3 w_{\max} + n w_{\max} \left| n - 1 - \sum_{i=1}^{m} x_i \right| + \sum_{i=1}^{m} w_i x_i$$

Theorem (Neumann and Wegener, 2007)

The expected time until the (1+1) EA *constructs a minimum spanning tree is* $O(m^2(\log n + \log w_{\max}))$.

Analysis of Typical Runs

Phase 1: Find some connected graph $\leadsto O(m \log m)$.
Phase 2: Find some spanning tree $\leadsto O(m \log m)$.

(1+1) EA for Minimum Spanning Trees

Given a weighted graph $G = (V, E, w)$ with $n := |V|, m := |E|$.
$x \in \{0,1\}^m$ encodes selection of edges.

$$f(x) := (\#\text{components}(x) - 1) \cdot n^3 w_{\max} + n w_{\max} \left| n - 1 - \sum_{i=1}^{m} x_i \right| + \sum_{i=1}^{m} w_i x_i$$

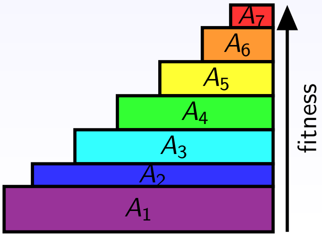
Theorem (Neumann and Wegener, 2007)

The expected time until the (1+1) EA *constructs a minimum spanning tree is* $O(m^2(\log n + \log w_{\max}))$.

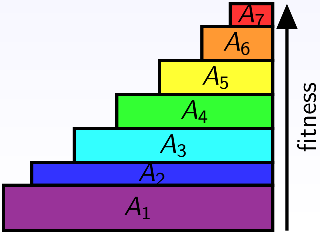
Analysis of Typical Runs

Phase 1: Find some connected graph $\rightsquigarrow O(m \log m)$.
Phase 2: Find some spanning tree $\rightsquigarrow O(m \log m)$.
Phase 3: Find a minimum spanning tree by suitable 2-bit flips with guaranteed average weight decrease.

Populations

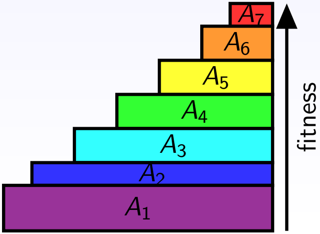


Populations



Phase i**.1:** wait until fraction $\chi(i)$ of the population is in A_i.
Phase i**.2:** try to find improvement from there.

Populations



Phase i**.1:** wait until fraction $\chi(i)$ of the population is in A_i.
Phase i**.2:** try to find improvement from there.

Fitness-level method for populations

$s_i :=$ probability bound when fraction $\chi(i)$ of individuals is in A_i.

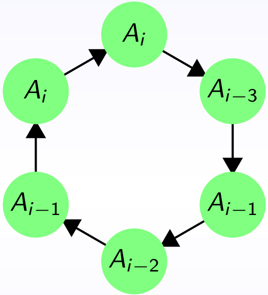
$$\sum_{i=1}^{m-1} \left(\frac{1}{s_i} + \text{time for population takeover to fraction } \chi(i) \right)$$

Offspring Populations and Parallel EAs

Create λ independent offspring: $s_i \longrightarrow 1 - (1 - s_i)^\lambda$

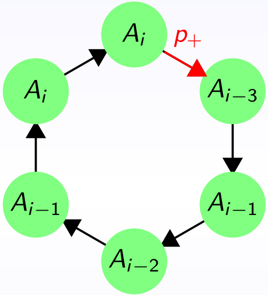
Offspring Populations and Parallel EAs

Create λ independent offspring: $s_i \longrightarrow 1 - (1 - s_i)^\lambda$



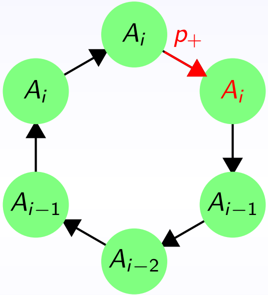
Offspring Populations and Parallel EAs

Create λ independent offspring: $s_i \longrightarrow 1 - (1 - s_i)^\lambda$



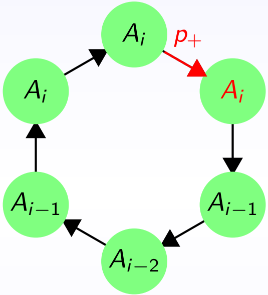
Offspring Populations and Parallel EAs

Create λ independent offspring: $s_i \longrightarrow 1 - (1 - s_i)^\lambda$



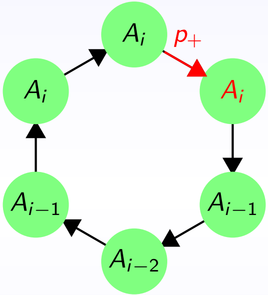
Offspring Populations and Parallel EAs

Create λ independent offspring: $s_i \longrightarrow 1 - (1 - s_i)^\lambda$



Offspring Populations and Parallel EAs

Create λ independent offspring: $s_i \longrightarrow 1 - (1 - s_i)^\lambda$

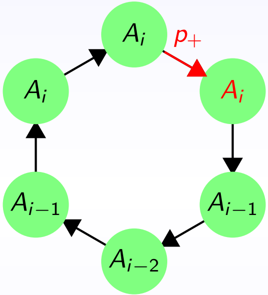


Expected parallel time with μ islands [Lässig and Sudholt, 2010]

Ring: $\displaystyle\sum_{i=1}^{m-1} \frac{3}{(p_+ s_i)^{1/2}}$

Offspring Populations and Parallel EAs

Create λ independent offspring: $s_i \longrightarrow 1 - (1 - s_i)^\lambda$

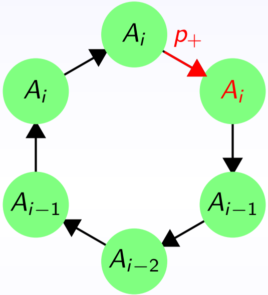


Expected parallel time with μ islands [Lässig and Sudholt, 2010]

Ring: $\displaystyle\sum_{i=1}^{m-1} \frac{3}{(p_+ s_i)^{1/2}}$ Torus: $\displaystyle\sum_{i=1}^{m-1} \frac{10}{p_+^{2/3} s_i^{1/3}}$

Offspring Populations and Parallel EAs

Create λ independent offspring: $s_i \longrightarrow 1 - (1 - s_i)^\lambda$

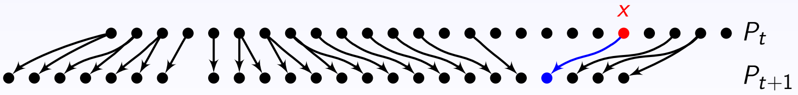


Expected parallel time with μ islands [Lässig and Sudholt, 2010]

Ring: $\displaystyle\sum_{i=1}^{m-1} \frac{3}{(p_+ s_i)^{1/2}}$ Torus: $\displaystyle\sum_{i=1}^{m-1} \frac{10}{p_+^{2/3} s_i^{1/3}}$ K_μ: $\displaystyle\frac{4m}{p_+} + \frac{4}{\mu}\sum_{i=1}^{m-1} \frac{1}{s_i}$

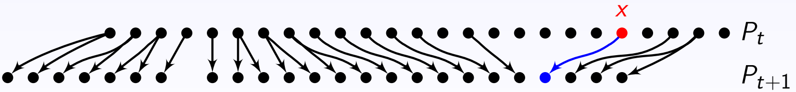
Fitness-Levels for Non-Elitist Populations

New population by sampling and mutating λ parents independently:



Fitness-Levels for Non-Elitist Populations

New population by sampling and mutating λ parents independently:



Theorem ([Lehre, GECCO 2011])

If

C1: *for one offspring* $Prob(A_i \to A_{i+1} \cup \cdots \cup A_m) \geq s_i$

C2: *for one offspring* $Prob(A_i \to A_i \cup \cdots \cup A_m) \geq p_0$

C3: *selection is sufficiently strong:* $\beta(\gamma, P)/\gamma \geq (1+\delta)/p_0$

C4: *population size sufficiently large:* $\lambda \geq \frac{2(1+\delta)}{\varepsilon\delta^2} \cdot \ln\left(\frac{m}{\min_i\{s_i\}}\right)$

then the expected number of function evaluations is at most

$$O\left(m\lambda^2 + \sum_{i=1}^{m-1} \frac{1}{s_i}\right).$$

Lower Bounds with Fitness Levels

Upper bounds with fitness levels [Wegener 2002]

Let s_i be a lower bound on $Prob(A_i \to A_{i+1} \cup \cdots \cup A_m)$. Then

$$\mathrm{E}(\textit{optimization time}) \le \sum_{i=1}^{m-1} \textit{Prob}(\mathcal{A} \textit{ starts in } A_i) \sum_{j=i}^{m-1} \frac{1}{s_i}.$$

Lower Bounds with Fitness Levels

Upper bounds with fitness levels [Wegener 2002]

Let s_i be a lower bound on $Prob(A_i \to A_{i+1} \cup \dots \cup A_m)$. Then

$$\mathrm{E}(\textit{optimization time}) \le \sum_{i=1}^{m-1} \textit{Prob}(\mathcal{A} \textit{ starts in } A_i) \sum_{j=i}^{m-1} \frac{1}{s_i}.$$

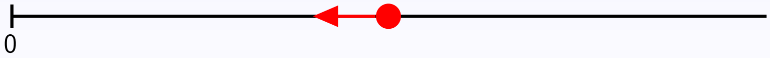
Lower bounds with fitness levels [Sudholt, 2010]

Let $u_i \cdot \gamma_{i,j}$ be an upper bound for $\mathrm{Prob}(A_i \to A_j)$ and $\sum_{j=i+1}^{m} \gamma_{i,j} = 1$. Assume for all $j > i$ and $0 < \chi \le 1$ that $\gamma_{i,j} \ge \chi \sum_{k=j}^{m} \gamma_{i,k}$. Then

$$\mathrm{E}(\text{optimization time}) \ge \sum_{i=1}^{m-1} \text{Prob}(\mathcal{A} \text{ starts in } A_i) \cdot \chi \sum_{j=i}^{m-1} \frac{1}{u_i}.$$

Overview

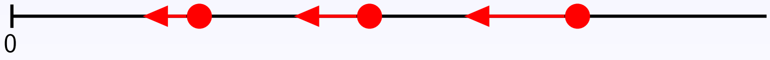
Additive Drift



Drift analysis, drift towards target, [He and Yao, 2004]

1. If $\mathrm{E}(X_t - X_{t+1} \mid X_t) \geq \delta$ whenever $X_t > 0$ then
$$\mathrm{E}(T \mid X_0) \;\leq\; \frac{X_0}{\delta}.$$
2. If $\mathrm{E}(X_t - X_{t+1} \mid X_t) \leq \delta$ then
$$\mathrm{E}(T \mid X_0) \;\geq\; \frac{X_0}{\delta}.$$

Variable Drift



Variable Drift results

1. [Mitavskiy, Rowe, and Cannings 2009] If $\mathrm{E}(X_t - X_{t+1} \mid X_t) \geq \delta_i$ whenever $X_t > i$ then

$$\mathrm{E}(T \mid X_0) \;\leq\; \sum_{i=1}^{\lceil X_0 \rceil} \frac{1}{\delta_i}.$$

2. [Johannsen, 2010] If $h\colon \mathbb{R}_0^+ \to \mathbb{R}^+$ is continuous and monotone increasing and $\mathrm{E}(X_t - X_{t+1} \mid X_t) \leq h(X_t)$ then

$$\mathrm{E}(T \mid X_0) \;\leq\; \frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{h(x)}\,\mathrm{d}x.$$

Multiplicative Drift

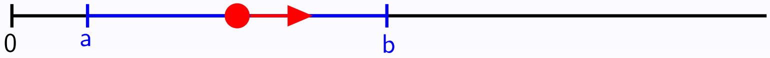
Multiplicative Drift Theorem [Doerr, Johannsen, Winzen, 2010]

If there exists a constant $\delta > 0$ such that $\mathrm{E}(X_t - X_{t+1} \mid X_t) \geq \delta X_t$ then

$$\mathrm{E}(T \mid X_0) \leq \frac{1 + \ln(X_0/s_{\min})}{\delta}$$

Bounds also hold with high probability [Doerr and Goldberg, 2010].

Exponential Lower Bounds with Drift



Theorem (Simplified Drift Theorem [Oliveto and Witt, 2008])

Assume

1. $E(X_t - X_{t+1} \mid X_t) \leq -\varepsilon$ *for* $a < X_t < b$,
2. $Prob(X_t - X_{t+1} \geq j) \leq \frac{r}{(1+\delta)^j}$ *for* $i > a$ *and* $j \in \mathbb{N}_0$.

If $X_0 \geq b$ *it holds* $Prob\left(T \leq 2^{c^* \ell}\right) = 2^{-\Omega(\ell)}$ *for a constant* c^*.

Example: (1,λ) EA

How to choose the offspring population size λ for the (1,λ) EA?

Example: (1,λ) EA

How to choose the offspring population size λ for the (1,λ) EA?

Theorem ([Jägersküpper and Storch, 2007])

Exponential gaps on $\textsc{OneMax}$ *for* $\lambda \leq 1/14 \cdot \ln n$ *vs.* $\lambda \geq 3 \ln n$.

Example: (1,λ) EA

How to choose the offspring population size λ for the (1,λ) EA?

Theorem ([Jägersküpper and Storch, 2007])

Exponential gaps on OneMax *for* $\lambda \leq 1/14 \cdot \ln n$ *vs.* $\lambda \geq 3 \ln n$.

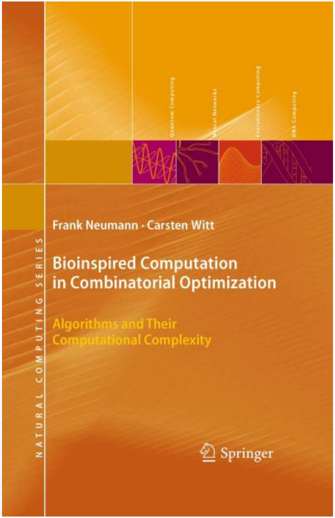
Refined result: phase transition at $\log_{\frac{e}{e-1}} n \approx 2.18 \ln n$.

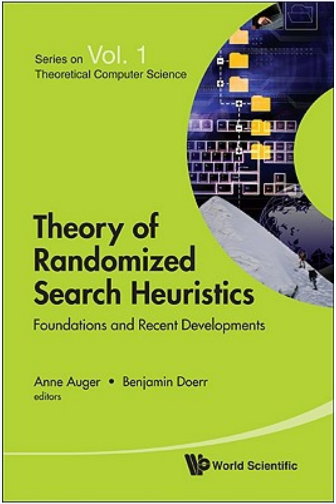
Theorem ([Rowe and Sudholt, in preparation])

If $\lambda \geq \log_{\frac{e}{e-1}} n$ *the expected number of function evaluations on* OneMax *is* $O(n \log n + n\lambda)$.

If $\lambda \leq (1-\varepsilon) \log_{\frac{e}{e-1}} n$ *it is at least* $2^{cn^{\varepsilon/2}}$ *with probability* $1 - 2^{-\Omega(n^{\varepsilon/2})}$.

Further Reading





Further Learning: Tutorials at GECCO 2011

Runtime Analysis Tutorials on Wednesday

- 8:30 Thomas Jansen and Frank Neumann: Computational Complexity and Evolutionary Computation
- 10:40 Carsten Witt: Theory of Randomized Search Heuristics
- 14:00 Dirk Sudholt: Theory of Swarm Intelligence
- 14:00 Tobias Friedrich and Frank Neumann: Foundations of Evolutionary Multi-Objective Optimization
- 16:10 Benjamin Doerr: Drift Analysis

Slides available from the ACM digital library.