dK-series:
Systematic Topology Analysis and Generation Using Degree Correlations

Dmitri Krioukov

dima@caida.org

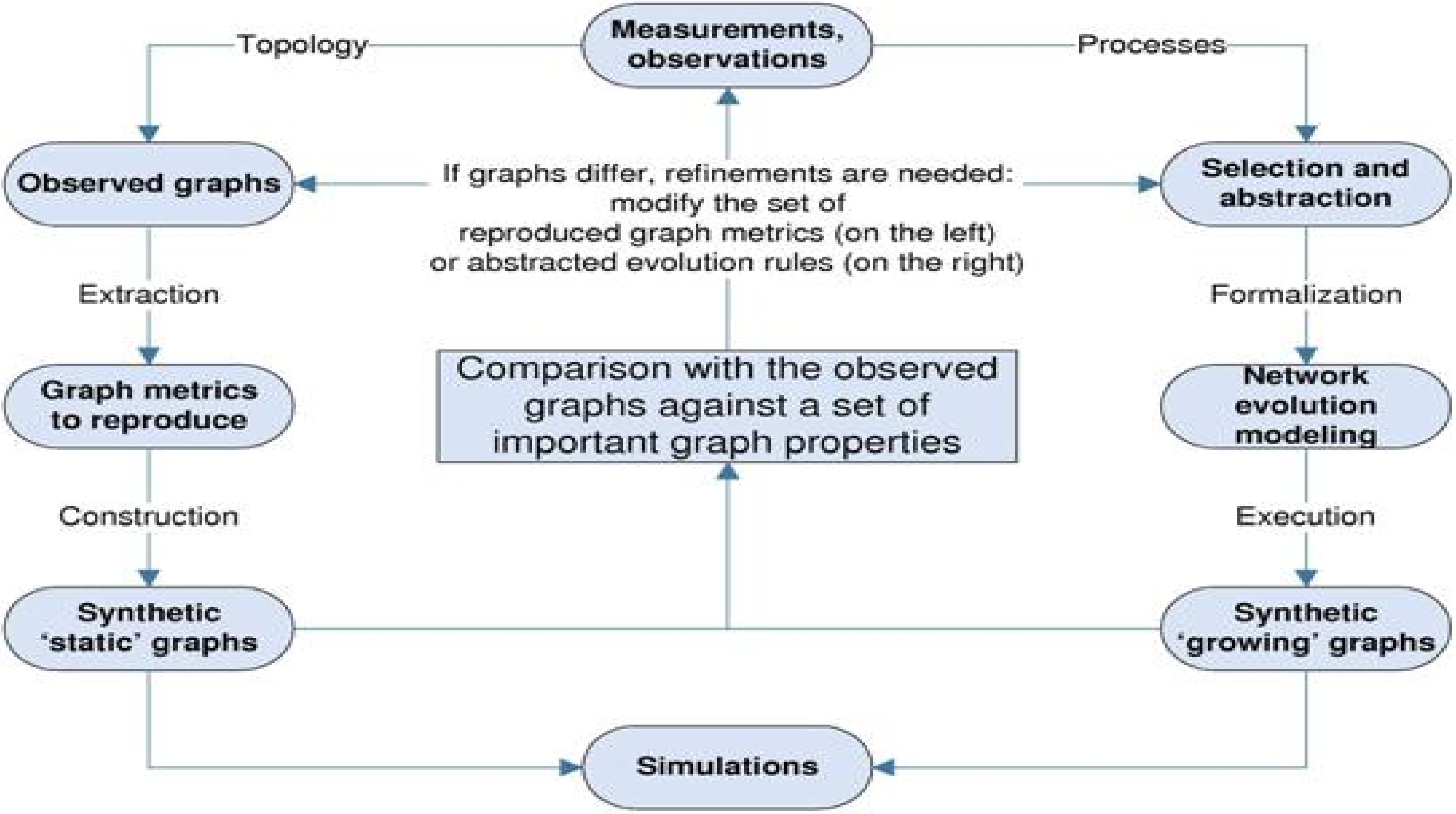
P. Mahadevan, K. Fall, and A. Vahdat

LIAFA, June 23d, 2006

Motivation: topology analysis and generation

- New *routing* and other protocol design, development, testing, etc.
  - Analysis: performance of a routing algorithm strongly depends on topology, the recent progress in routing theory has become topology analysis
  - Generation: empirical estimation of scalability: new routing might offer *X*-time smaller routing tables for today but scale *Y*-time worse, with $Y >> X$
- Network robustness, resilience under attack, worm spreading, etc.
- Traffic engineering, capacity planning, network management, etc.
- In general: “what if”, predictive power, evolution

Network topology research



Important topology metrics

- Spectrum
- Distance distribution
- Betweenness distribution
- Degree distribution
- Assortativity
- Clustering

Problems

- No way to reproduce most of the important metrics
- No guarantee there will not be any other/new metric found important

Our approach

- Look at inter-dependencies among topology characteristics
- See if by reproducing most basic, simple, but not necessarily practically relevant characteristics, we can also reproduce (capture) all other characteristics, including practically important
- Try to find the one(s) defining *all others*

Outline

- Introduction
- *dK*-*:
  - *dK*-distributions
  - *dK*-series
  - *dK*-graphs
  - *dK*-randomness
  - *dK*-explorations
- Construction
- Evaluation
- Conclusion

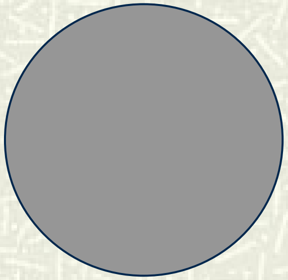
The main observation ☺

Graphs are structures of *connections* between nodes

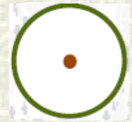
dK-distributions as a series of graphs' *connectivity* characteristics

0K

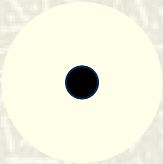
Average degree <*k*>



1K

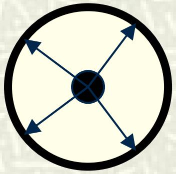


Degree distribution $P(k)$



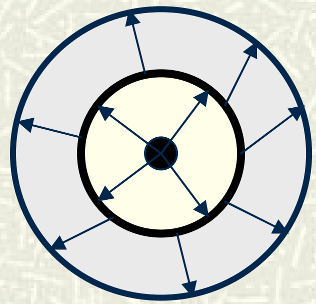
2K

Joint degree distribution $P(k_1,k_2)$

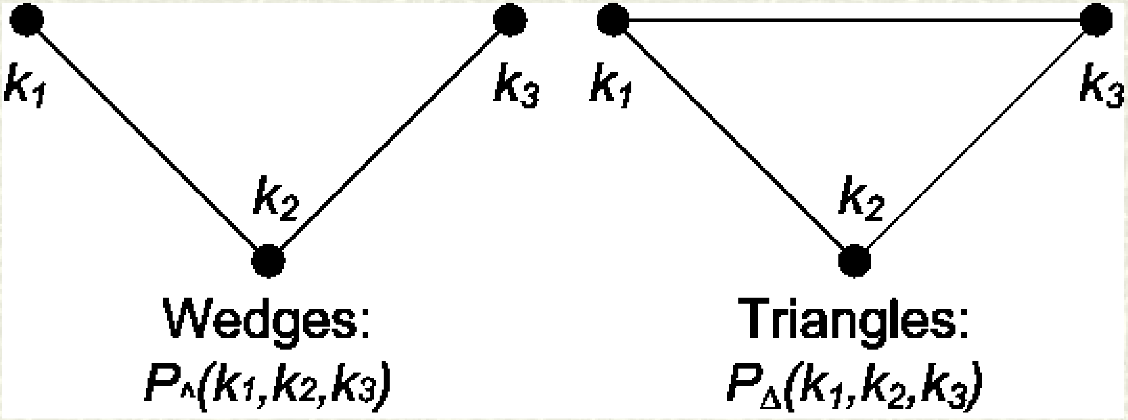


3K

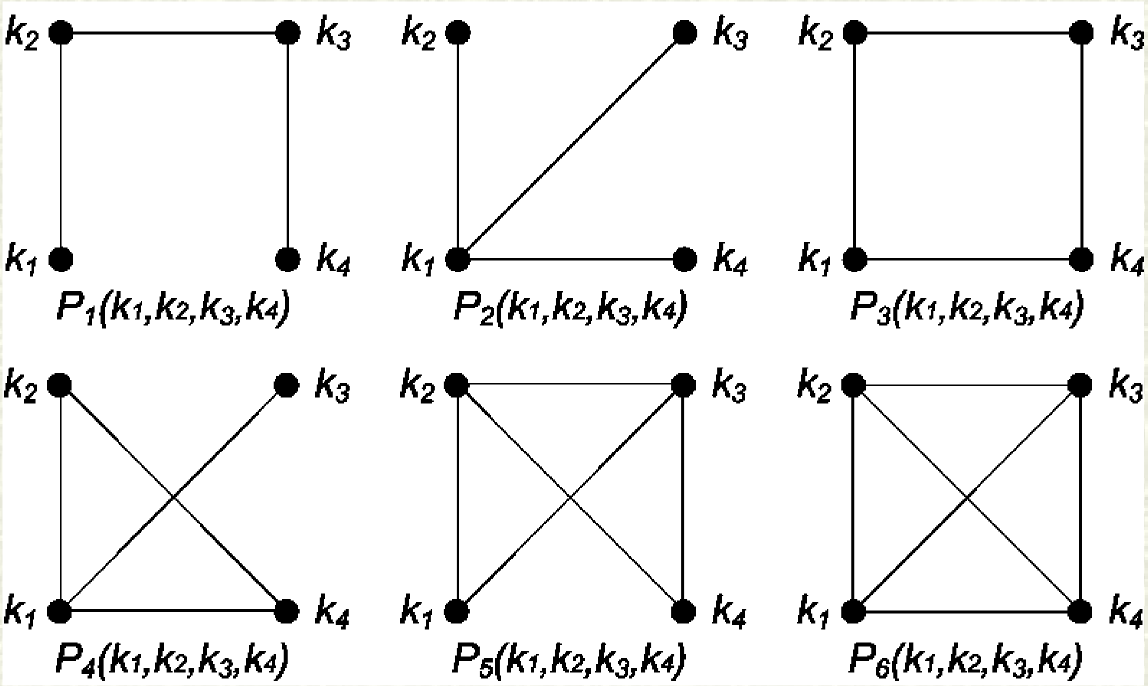
“Joint edge degree” distribution $P(k_1,k_2,k_3)$



3K, more exactly



4K



k2
k3
k1
k4
P1(k1,k2,k3,k4)
k2
k3
k1
k4
P2(k1,k2,k3,k4)
k2
k3
k1
k4
P3(k1,k2,k3,k4)
k2
k3
k1
k4
P4(k1,k2,k3,k4)
k2
k3
k1
k4
P5(k1,k2,k3,k4)
k2
k3
k1
k4
P6(k1,k2,k3,k4)

Definition of *dK*-distributions

dK-distributions are degree correlations within simple connected graphs of size *d*

Definition of dK-series P_d

Given some graph G, graph G' is said to have *property* P_d if G''s dK-distribution is the same as G's

Definition of *dK*-graphs

dK-graphs are graphs having property P_d

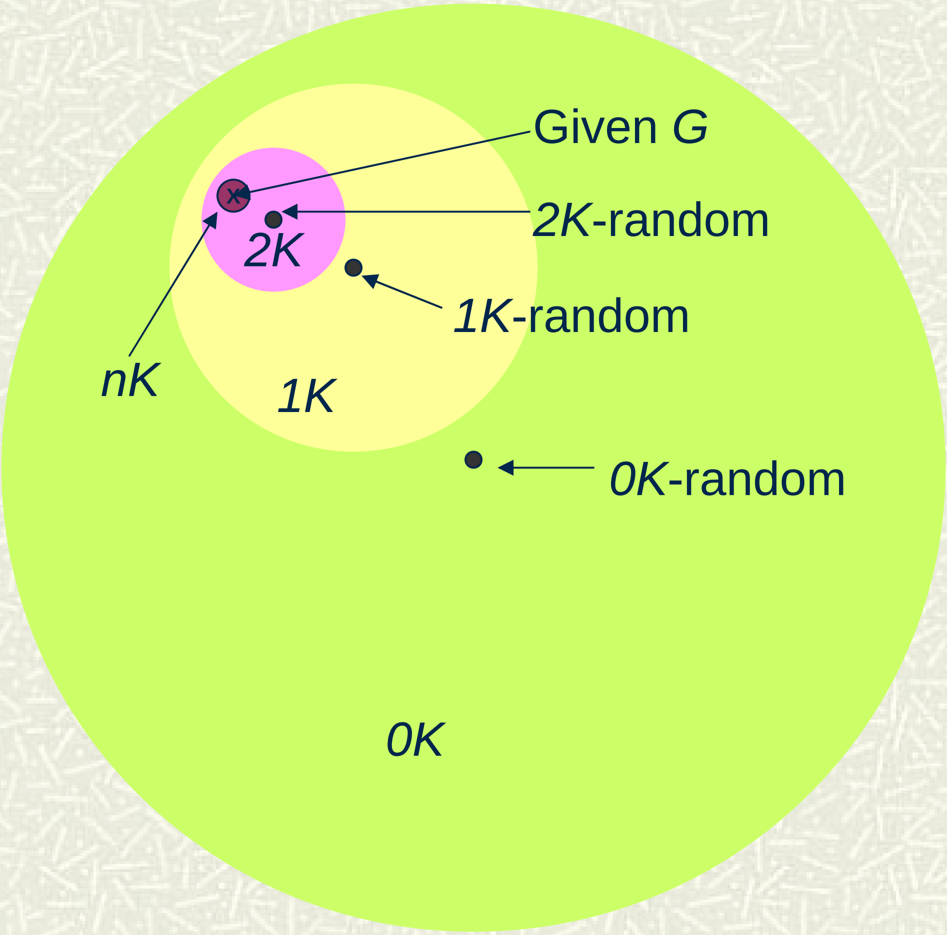
Nice properties of properties P_d

- **Constructability**: we can construct graphs having properties P_d (dK-graphs)
- **Inclusion**: if a graph has property P_d, then it also has all properties P_i, with $i < d$ (dK-graphs are also iK-graphs)
- **Convergence**: the set of graphs having property P_n consists only of one element, G itself (dK-graphs converge to G)

Convergence…

…guarantees that *all* (even not yet defined!) graph metrics can be captured by sufficiently high d

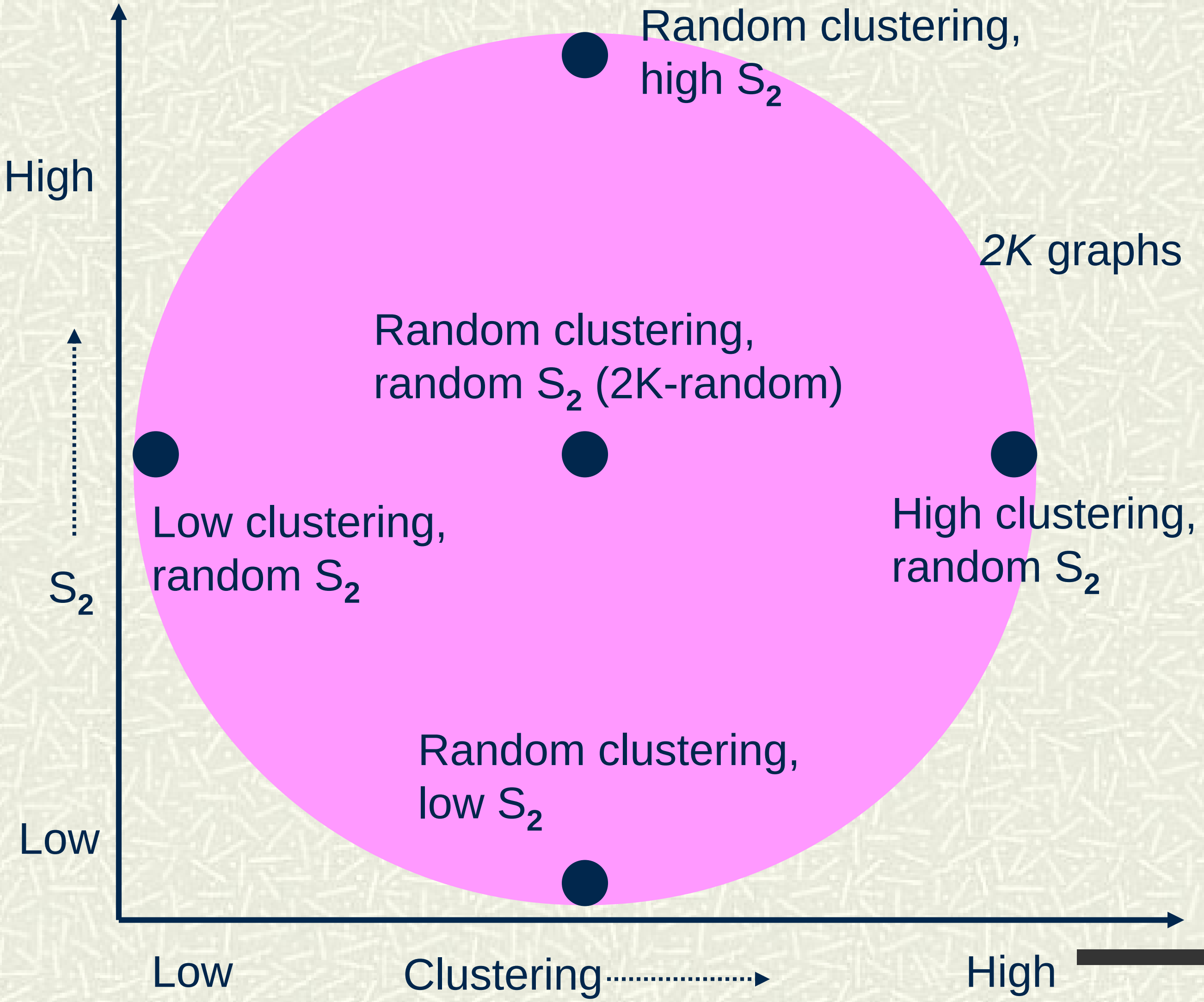
Inclusion and *dK*-randomness



dK-explorations

- To identify the minimum *d*, s. t. *dK*(-random) graphs provide a sufficiently accurate approximation of *G*:
- Find simple (scalar) metrics that are defined by P_{d+1} but not by P_d and construct *dK*-nonrandom-graphs with extreme (max or min) values of these metrics
- There are two extreme metrics of this type
  - correlations of degrees of nodes at distance *d*
  - concentration of *d*-simplices
- If differences between these *dK*-exotic graphs are small, then *d* is high enough

2K-exploration example



dK-summary

Tag dK	Property symbol	dK-distribution	$\mathcal{P}_d$ defines $\mathcal{P}_{d-1}$	Edge existence probability in stochastic constructions	Maximum entropy value of $(d+1)K$-distribution in dK-random graphs
$0K$	$\mathcal{P}_0$	$\bar{k}$		$p_{0K} = \bar{k}/n$	$P_{0K}(k) = e^{-\bar{k}}\bar{k}^k/k!$
$1K$	$\mathcal{P}_1$	$P(k)$	$k = \sum kP(k)$	$p_{1K}(q_1, q_2) = q_1 q_2/(n\bar{q})$	$P_{1K}(k_1, k_2) = k_1 P(k_1) k_2 P(k_2)/k^2$
$2K$	$\mathcal{P}_2$	$P(k_1, k_2)$	$P(k) = (\bar{k}/k)\sum_{k'} P(k, k')$	$p_{2K}(q_1, q_2) = (\bar{q}/n)P(q_1, q_2)/(P(q_1)P(q_2))$	See [10] for clustering in $2K$-random graphs
$3K$	$\mathcal{P}_3$	$P_\wedge(k_1, k_2, k_3)$ $P_\triangle(k_1, k_2, k_3)$	By counting edges, we get $P(k_1, k_2) \sim \sum_k \{P_\wedge(k, k_1, k_2) + P_\triangle(k, k_1, k_2)\}/(k_1 - 1) \sim \sum_k \{P_\wedge(k_1, k_2, k) + P_\triangle(k_1, k_2, k)\}/(k_2 - 1)$, where we omit normalization coefficients.		
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
nK	$\mathcal{P}_n$	G			

Constructability

- Introduction
- *dK*-*
- Construction
  - Stochastic
  - Pseudograph
  - Matching
  - Rewiring
    - *dK*-randomizing
    - *dK*-targeting
- Evaluation
- Conclusion

Stochastic approach

- Classical (Erdos-Renyi) random graphs are *0K*-random graph in the stochastic approach
- Easily generalizable for any *d*:
  - Reproduce the expected value of the *dK*-distributions by connecting random *d*-plets of nodes with (conditional) probabilities extracted from *G*
- Best for theory
- Worst in practice

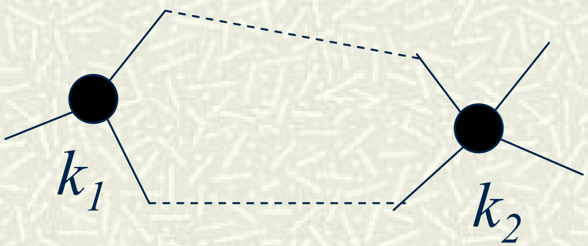
Pseudograph approach

- Reproduces *dK*-distributions exactly
- Constructs not necessarily connected pseudographs
- Extended for $d = 2$
- Failed to generalize for $d > 2$: *d*-sized subgraphs start overlap over edges at $d = 3$

Pseudograph details

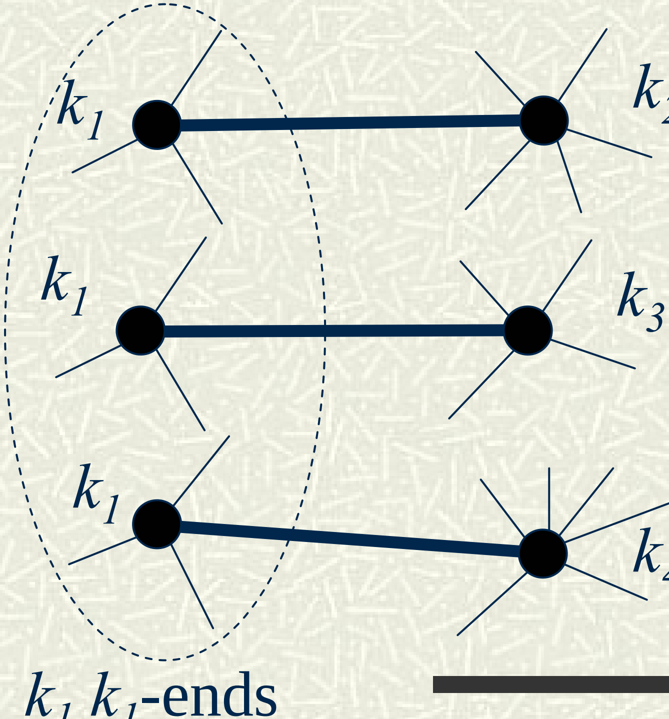
1K

1. dissolve graph into a random soup of nodes
2. crystallize it back



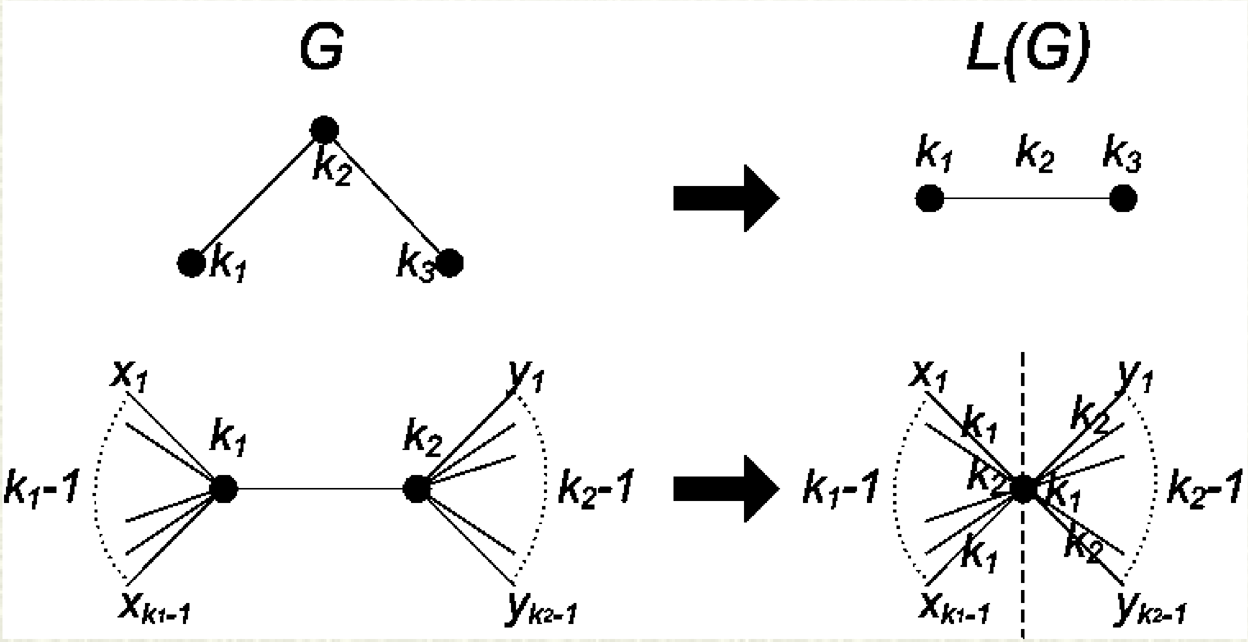
2K

1. dissolve graph into a random soup of edges
2. crystallize it back



3K-pseudograph failure

1. dissolve graph into a random soup of *d*-plets
2. cannot crystallize it back



Matching approach

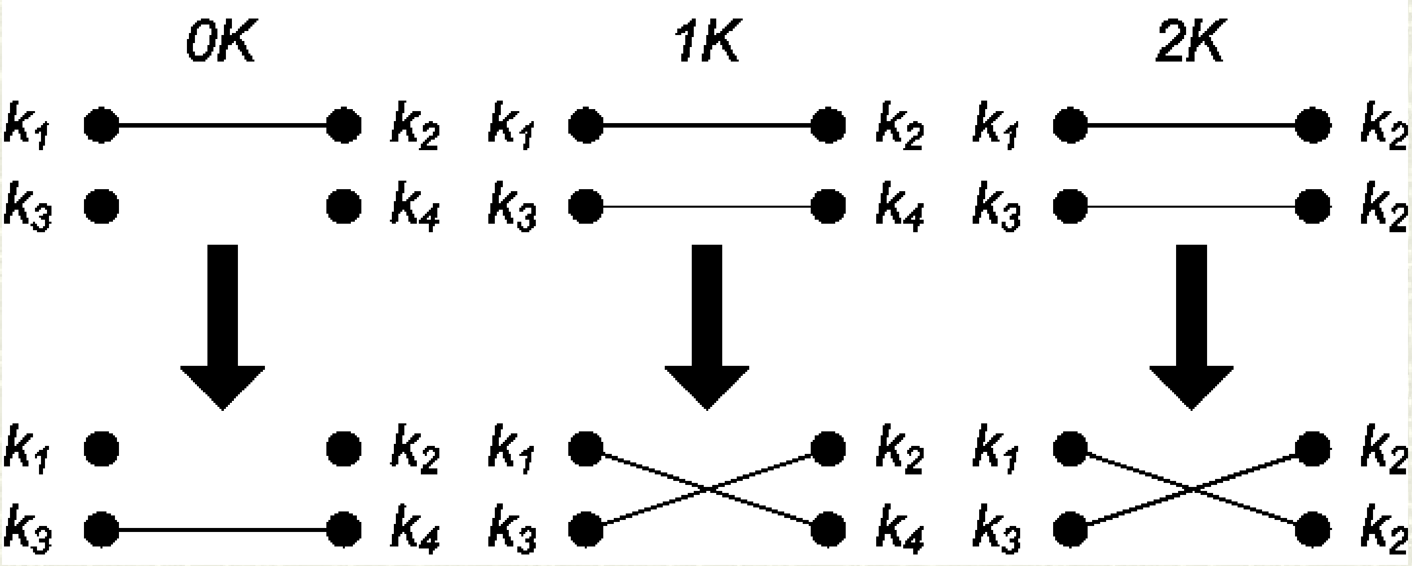
- Pseudograph + badness (loop) avoidance
- Extended for $d = 2$, but loop avoidance is difficult
- Failed to generalize for $d > 2$

Rewiring

- Generalizable for any *d*
- Works in practice

dK-randomizing rewiring

dK-preserving random rewiring



dK-targeting rewiring

- *d'K*-preserving rewiring ($d' < d$) moving graph closer to *dK*
- *dK*-distance D_d can be any non-negative scalar metric measuring the difference between the current and target values of the *dK*-distribution (e.g., the sum of squares of differences in numbers of *d*-sized subgraphs)
- Normally, accept a rewiring only if $\Delta D_d \leq 0$
- To check ergodicity:
  - accept a rewiring even if $\Delta D_d > 0$ with probability $\exp(-\Delta D_d / T)$
    - $T \to \infty$: *d'K*-randomizing rewiring
    - $T \to 0$: *dK*-targeting rewiring
  - start with a high temperature and gradually cool down the system

Outline

- Introduction
- *dK*-*
- Construction
- Evaluation
  - Algorithms
  - Topologies
    - skitter
    - HOT
- Conclusion

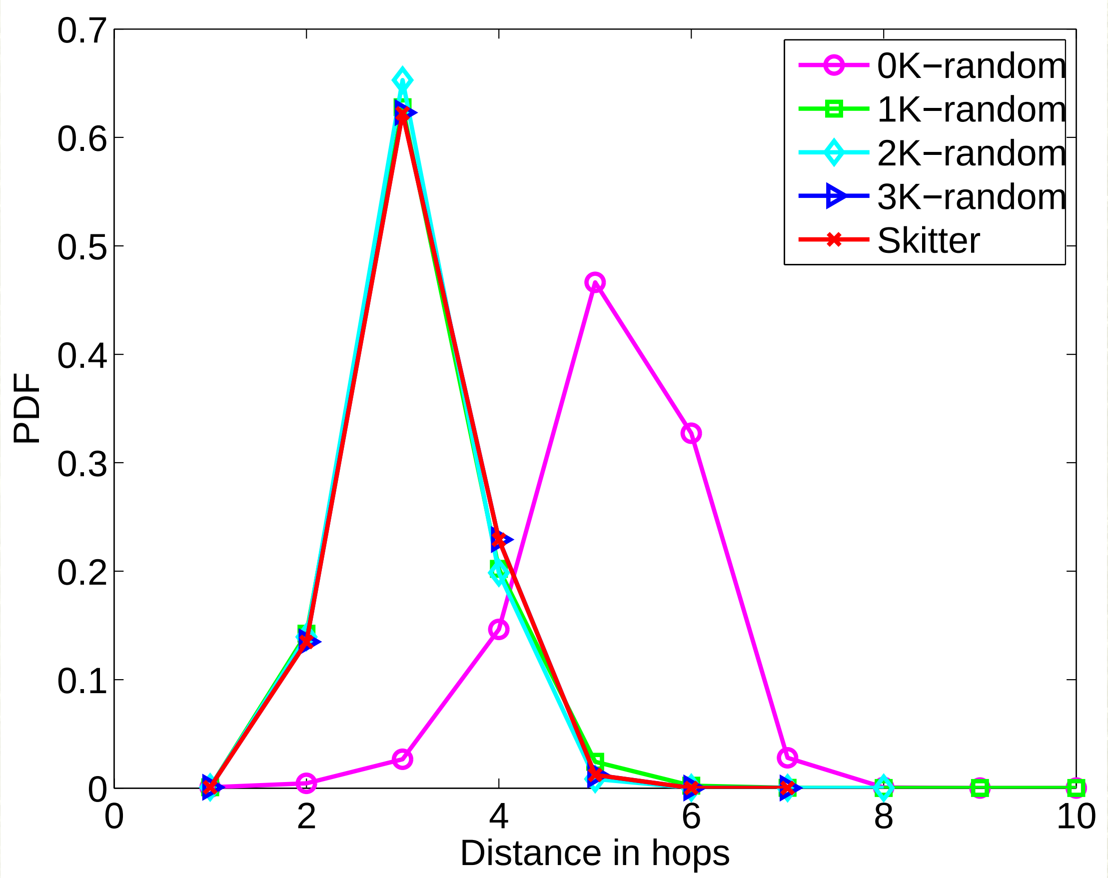
Algorithms

- All algorithms deliver consistent results for $d = 0$
- All algorithms, except stochastic(!), deliver consistent results for $d = 1$ and $d = 2$
- Both rewiring algorithms deliver consistent results for $d = 3$

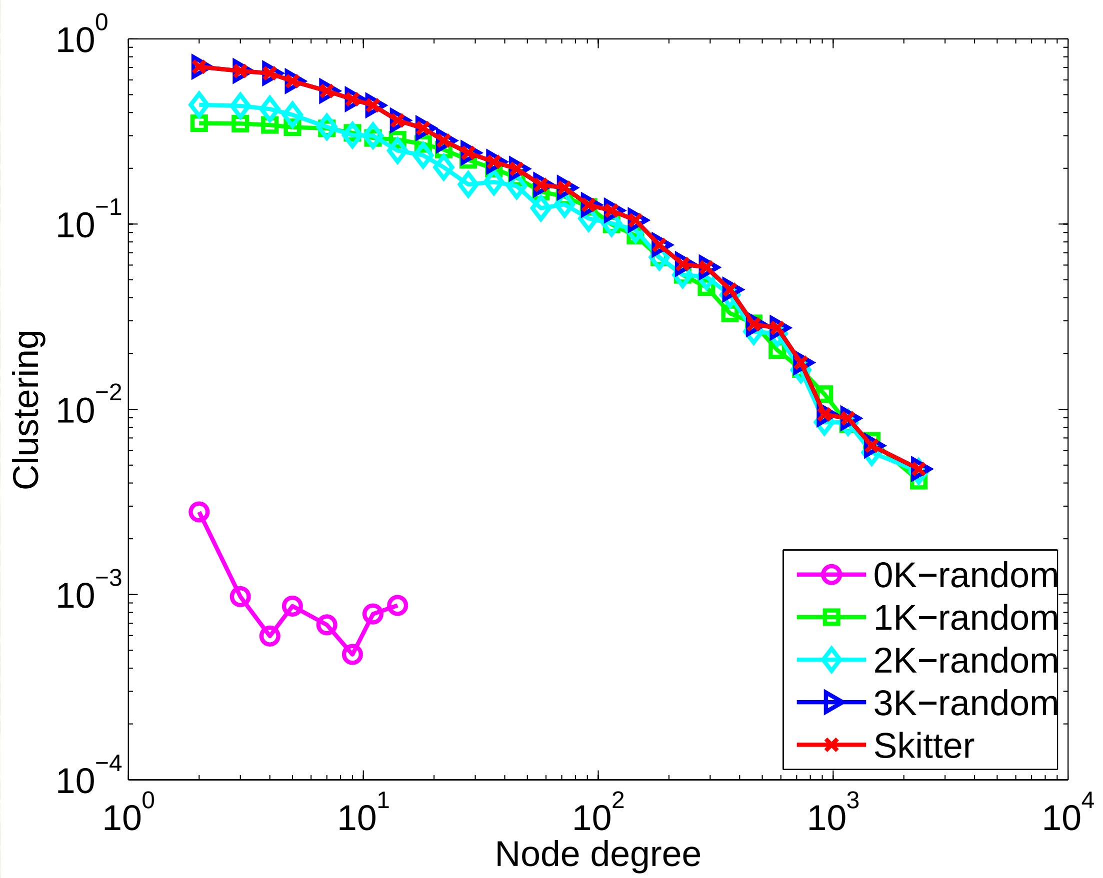
skitter scalar metrics

Metric	*0K*	*1K*	*2K*	*3K*	skitter
$<k>$	6.31	6.34	6.29	6.29	6.29
r	0	-0.24	-0.24	-0.24	-0.24
$<C>$	0.001	0.25	0.29	0.46	0.46
d	5.17	3.11	3.08	3.09	3.12
σ_d	0.27	0.4	0.35	0.35	0.37
λ_1	0.2	0.03	0.15	0.1	0.1
λ_{n-1}	1.8	1.97	1.85	1.9	1.9

skitter distance distribution



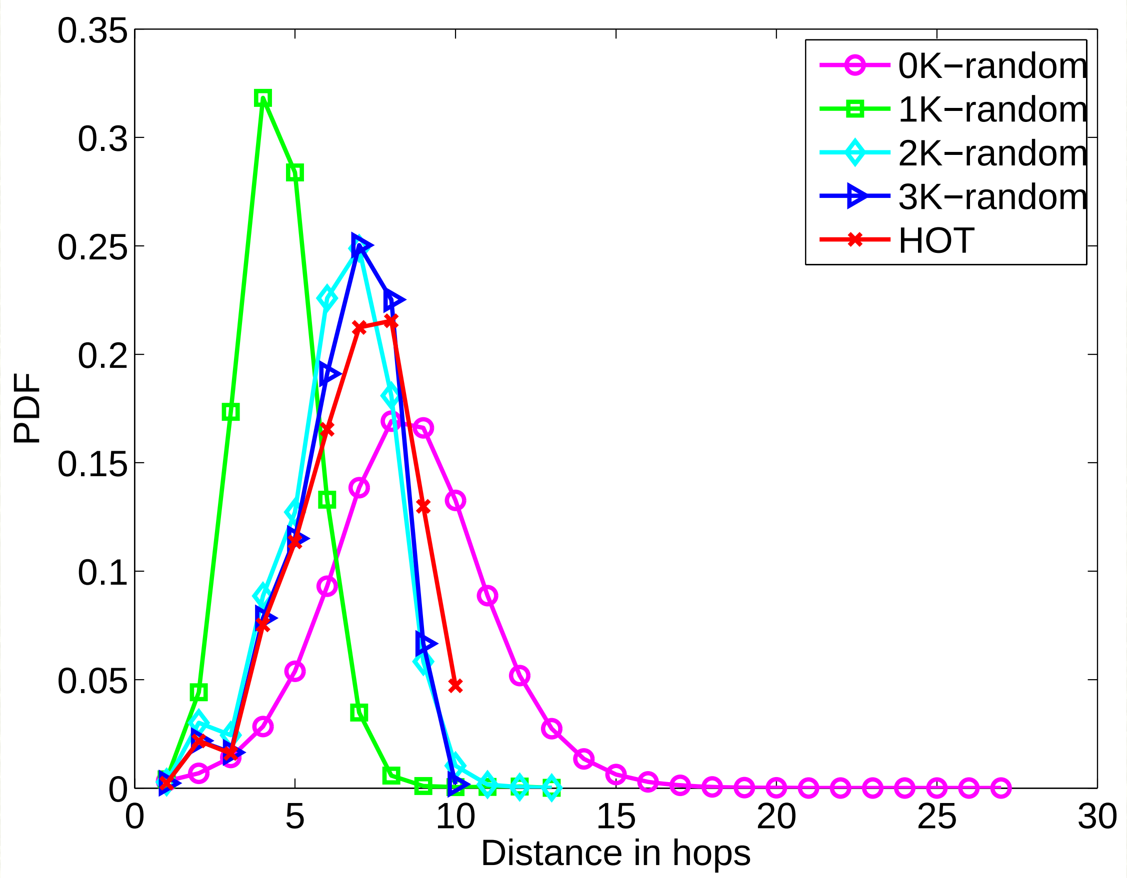
skitter clustering



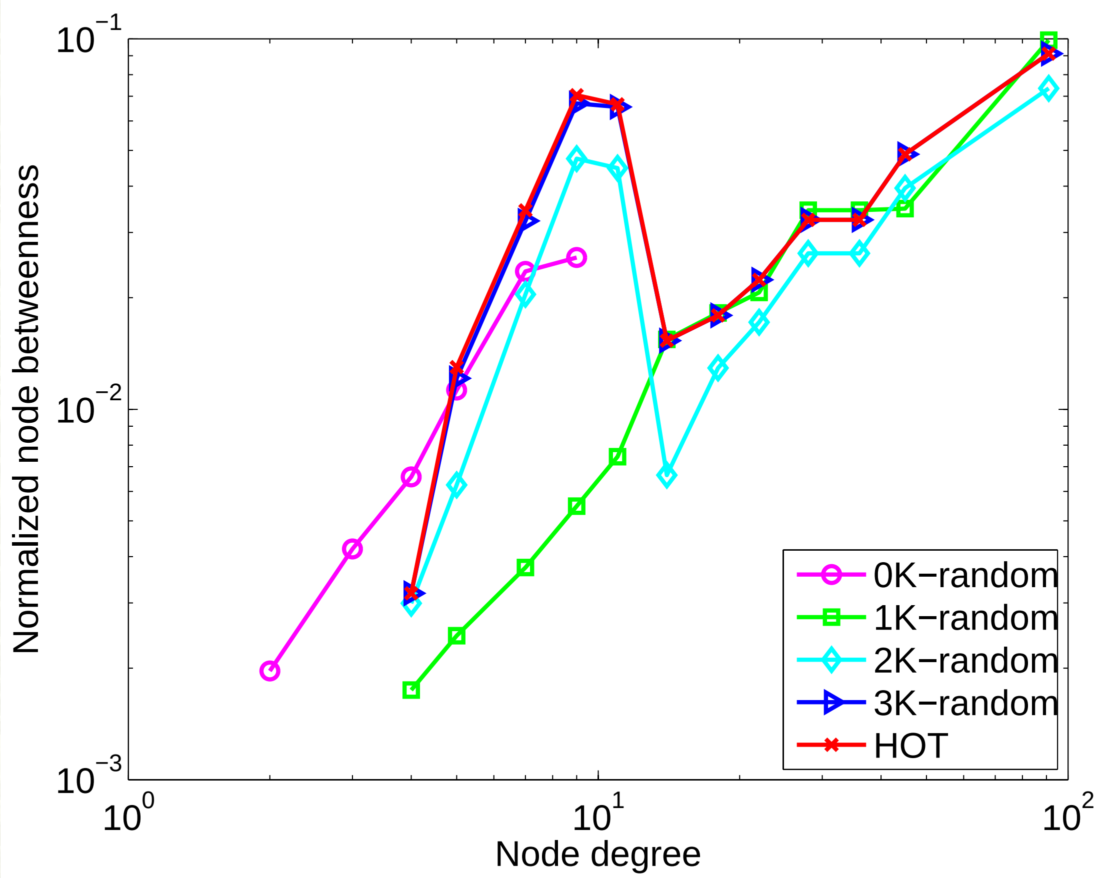
HOT scalar metrics

Metric	*0K*	*1K*	*2K*	*3K*	*HOT*
$<k>$	2.47	2.59	2.18	2.10	2.10
r	-0.05	-0.14	-0.23	-0.22	-0.22
$<C>$	0.002	0.009	0.001	0	0
d	8.48	4.41	6.32	6.55	6.81
σ_d	1.23	0.72	0.71	0.84	0.57
λ_1	0.01	0.034	0.005	0.004	0.004
λ_{n-1}	1.989	1.967	1.996	1.997	1.997

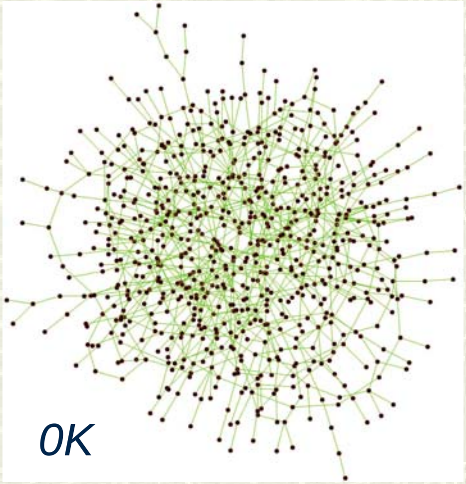
HOT distance distribution

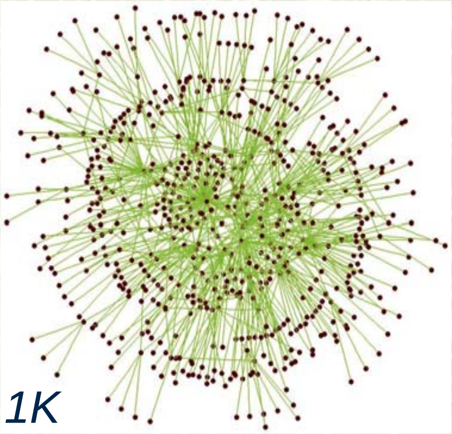


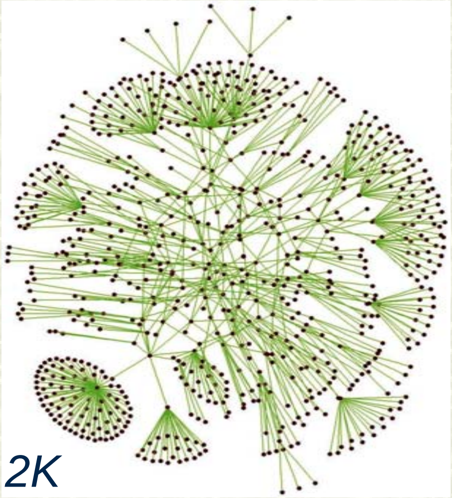
HOT betweenness distribution

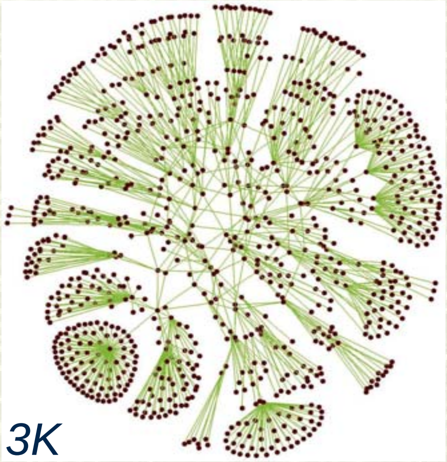


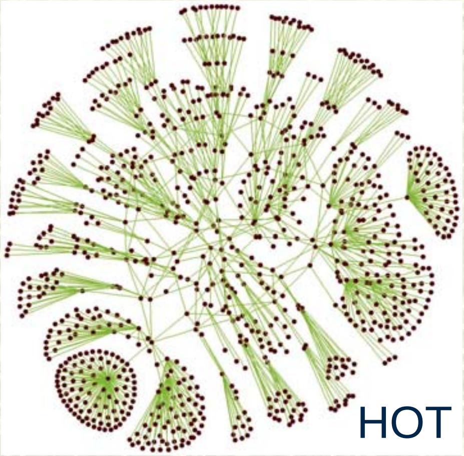
HOT *dK*-porn











Outline

- Introduction
- *dK*-*
- Construction
- Evaluation
- Conclusion

Conclusions

- **Analysis**: inter-dependencies among topology metrics and connections between
  - local and global structure
  - continuous and discrete worlds
  - equilibrium and non-equilibrium models (if a topology is *dK*-random, its evolution models need to explain just the *dK*-distribution)
- **Generation**: topology generator with arbitrary level of accuracy