Estimating scalability issues while finding an optimal assignment for carpooling

Luk Knapen

Universiteit Hasselt IMOB

July 17, 2013

Basic Idea - Concepts

Application Area : Evaluation of Carpooling Advisor Software

1. Commuting, periodic trip execution
2. Provide global *matcher service*
3. Matcher shall advise so that carpooling negotiation success is maximal
4. Feedback from negotiation is used to train the advisor

Hypotheses

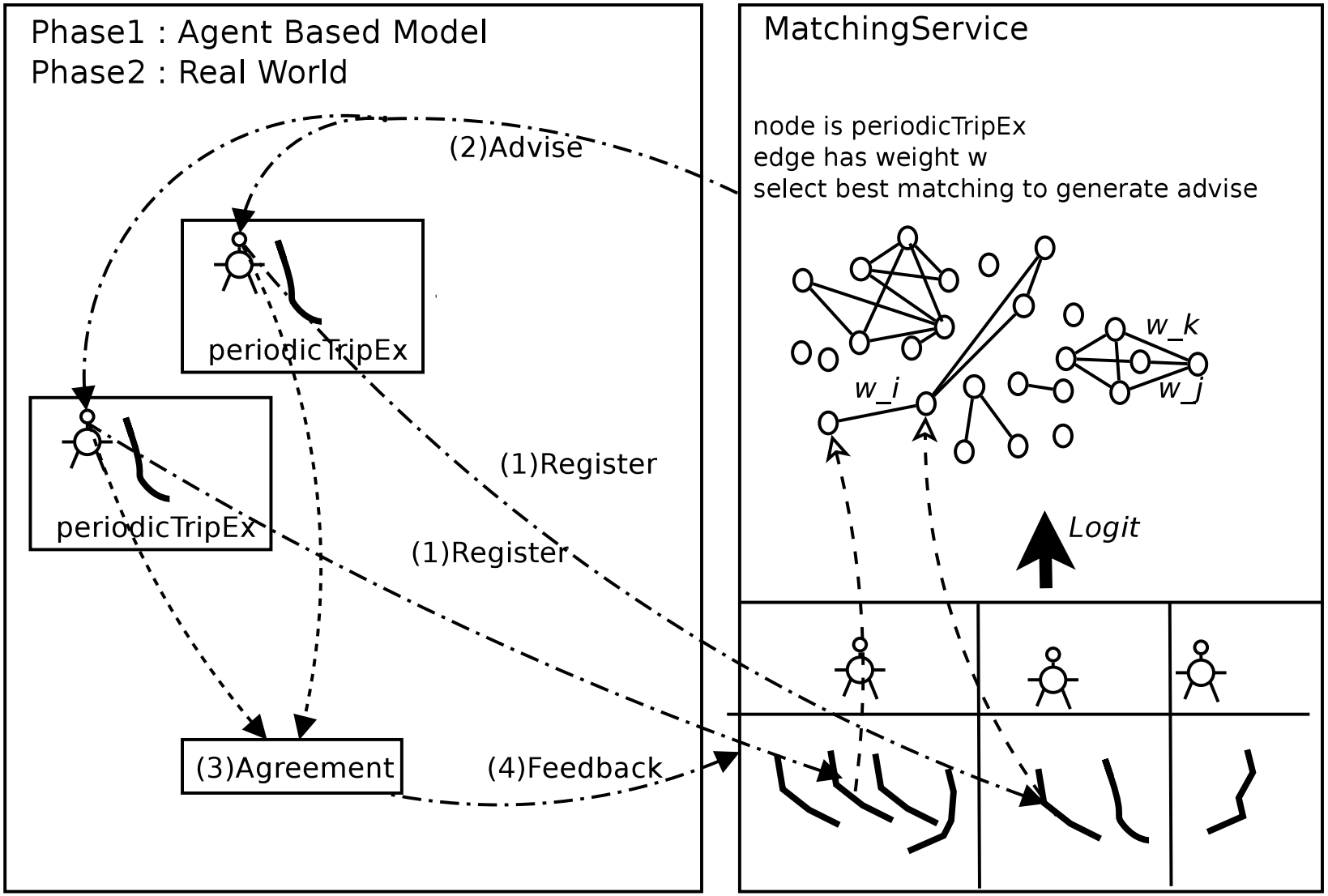
1. Sufficient business model so that people provide feedback
   1. about negotiation result
   2. mutual qualifications (reputation)
2. Single driver constraint : route for each passenger is embedded in the driver's route
3. No carpool parkings considered

Basic Idea - Concepts

Agent based model used to support *matcher software evaluation*

1. critical mass of users required
2. performance and effectiveness
3. investigate transient effects at startup

Basic Idea - Concepts



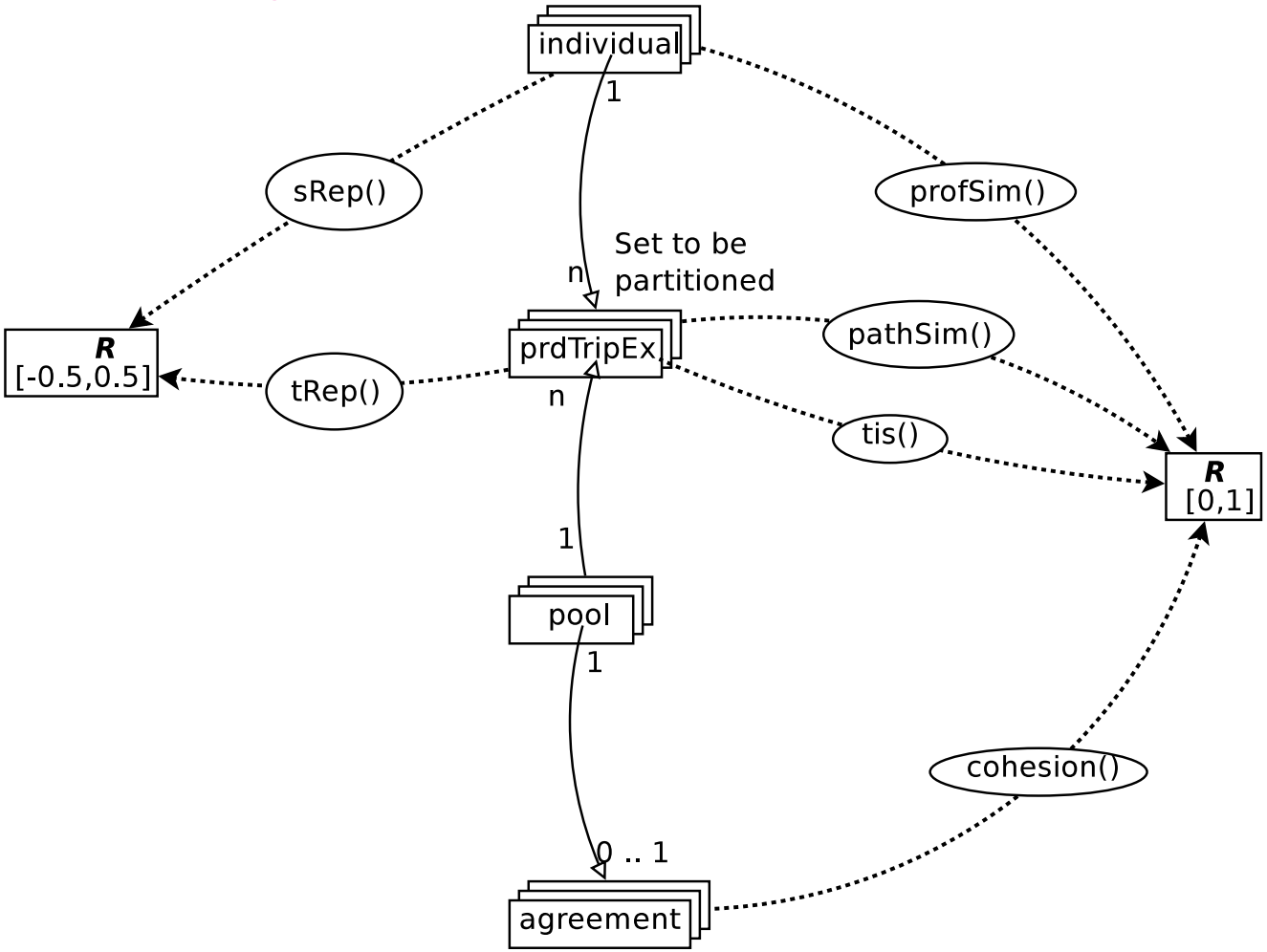
Determining carpool negotiation probablity

Negotiation success is determined using *similarities*

Profile similarity	(Person,Person) attribute
Safety reputation	Person attribute
Path similarity	(Trip,Trip) attribute
Cohesion	Pool attribute
Time Interval similarity	(Person,Person,Trip) attribute
Timeliness reputation	(Person,Trip) attribute

Determining carpool negotiation probablity

Functions Used



Determining carpool negotiation probablity

Profile similarity

- Consists of factors relevant for homophily

Socio-economic class	discrete, ordered
Age	discrete, ordered
Gender	discrete, categorical

- Motivation : limit number of inputs to `logit` estimator

$$s_{cat}(v_0, v_1) = \begin{cases} 1 & if \quad v_0 = v_1 \\ 0 & else \end{cases} \tag{1}$$

$$s_{ord}(v_0, v_1) = 1 - \frac{|v_0 - v_1|}{range} \tag{2}$$

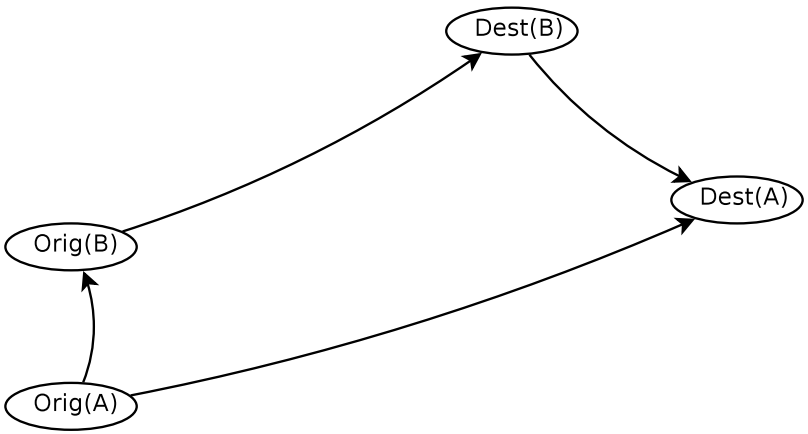
$$profSim = \sum_i s_i \cdot w_i \tag{3}$$

Determining carpool negotiation probablity

Path similarity

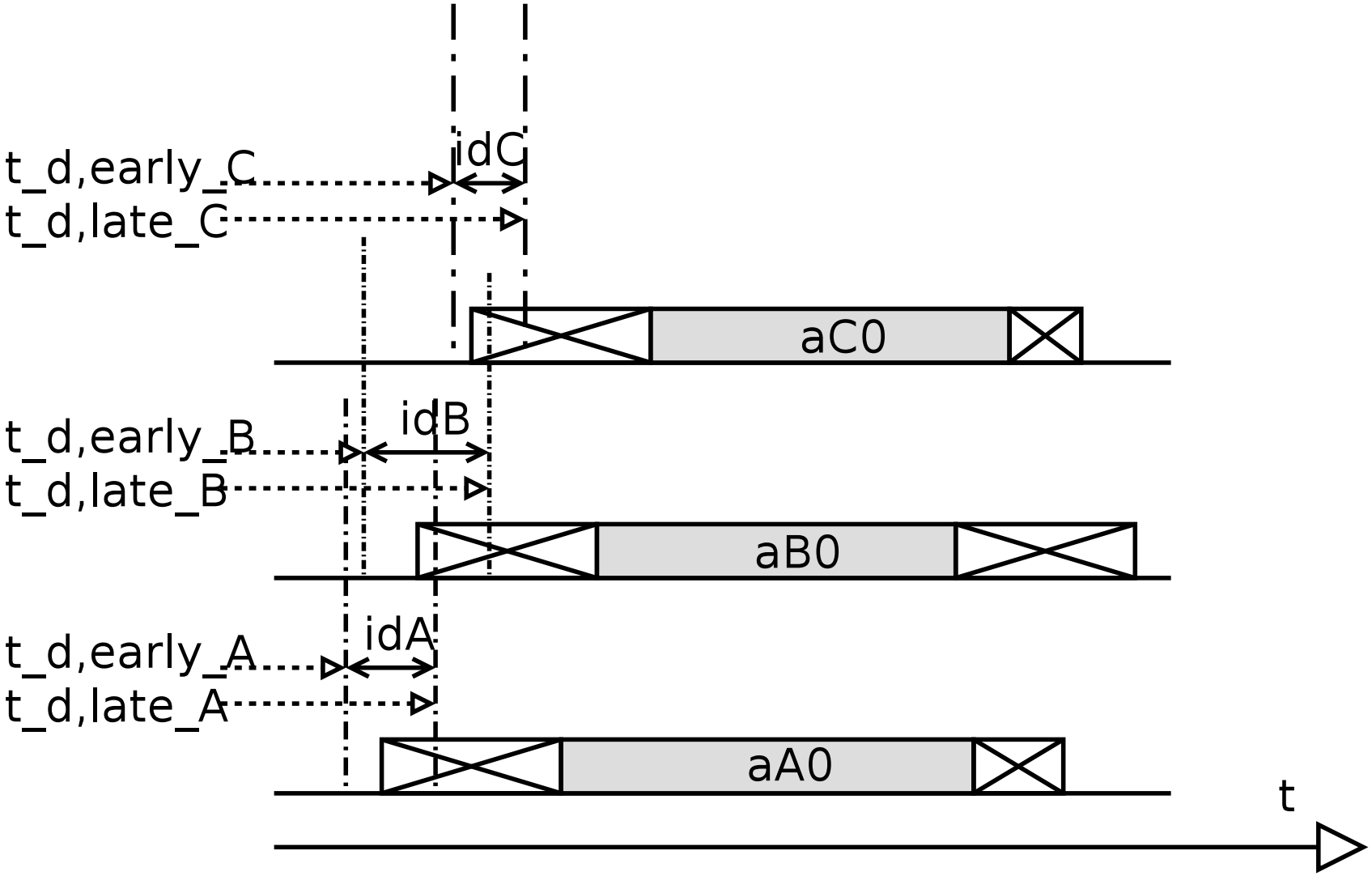
- TAZ (Traffic Analysis Zone) based
- Not symmetric
- Compares *carpooled* and *solo* paths for driver

$$pathSim = \frac{d(O_A, D_A)}{d(O_A, O_B) + d(O_B, D_B) + d(D_B, D_A)} \tag{4}$$



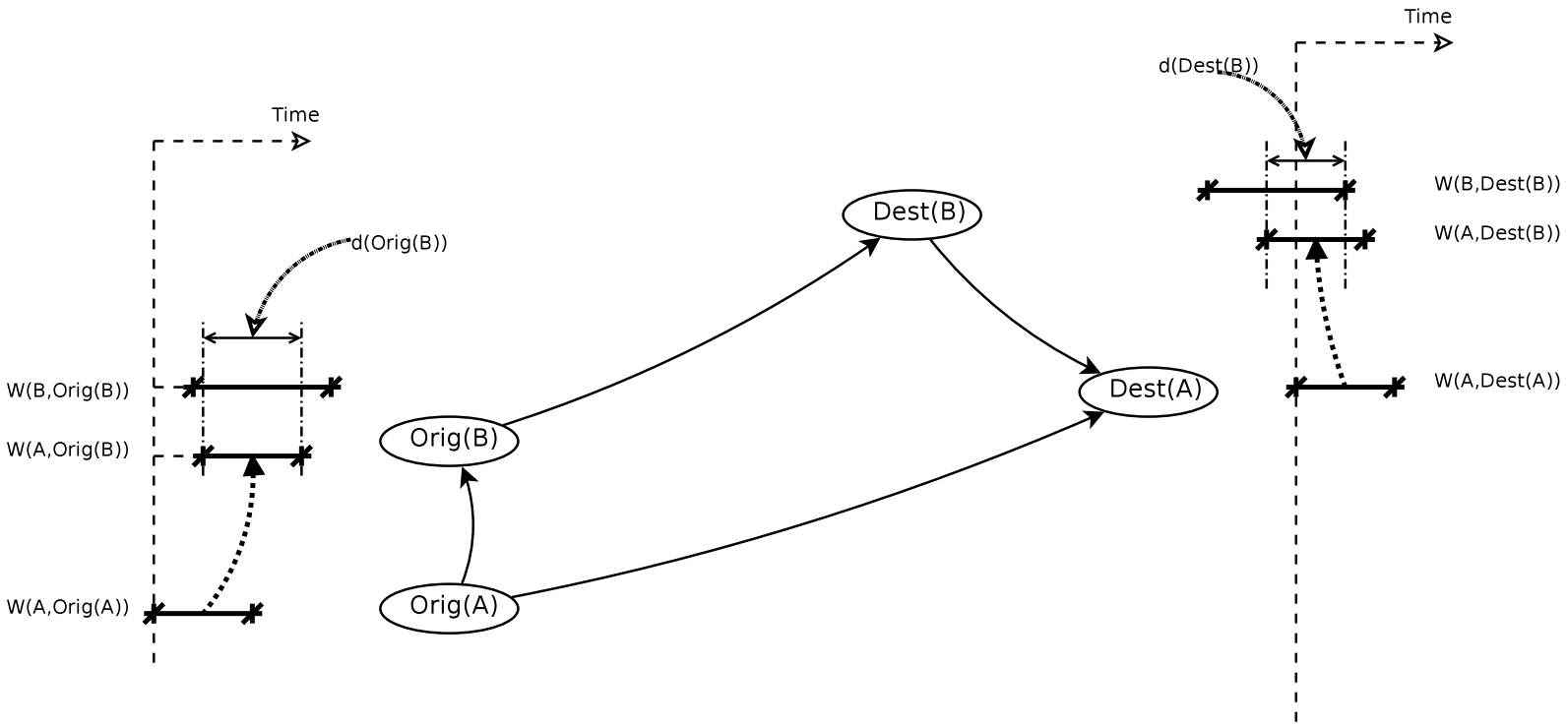
Determining carpool negotiation probablity

Time Interval Compatibility



Determining carpool negotiation probablity

Time Interval Similarity based on common periods to start/finish trip



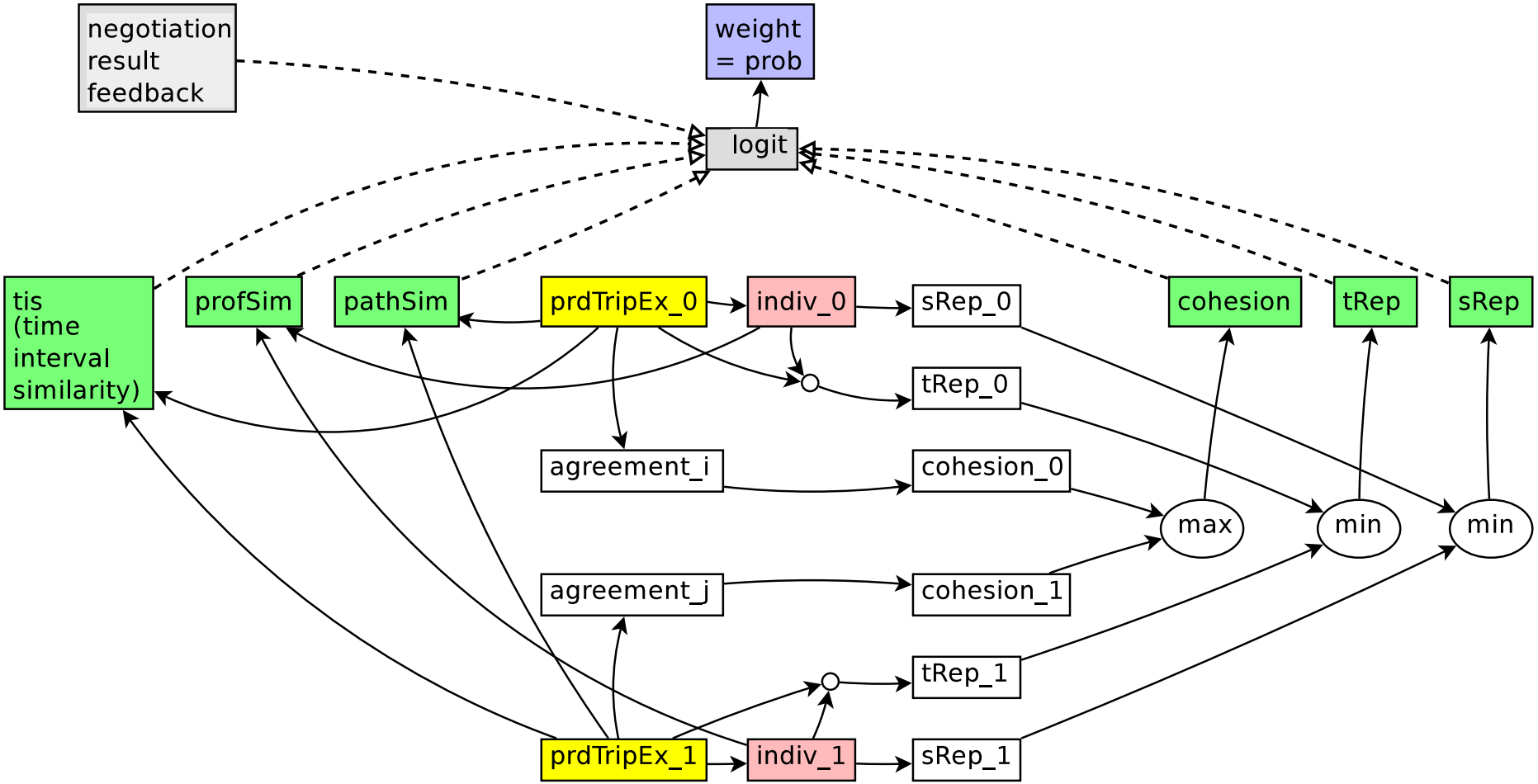
$$d(L) = dur(W(i, L) \cap W(j, L)) \quad (5)$$

$$tis(T_A, T_B, L) = 1 - e^{-\alpha \cdot d(L)} \quad (6)$$

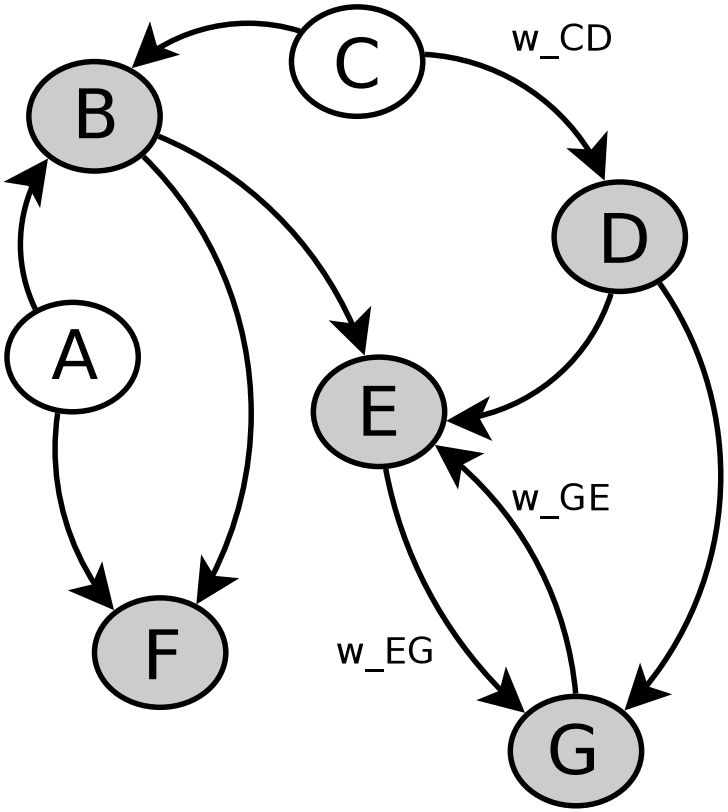
$$tis(T_A, T_B) = \min(tis(T_A, T_B, Orig(B), tis(T_A, T_B, Dest(B))) \quad (7)$$

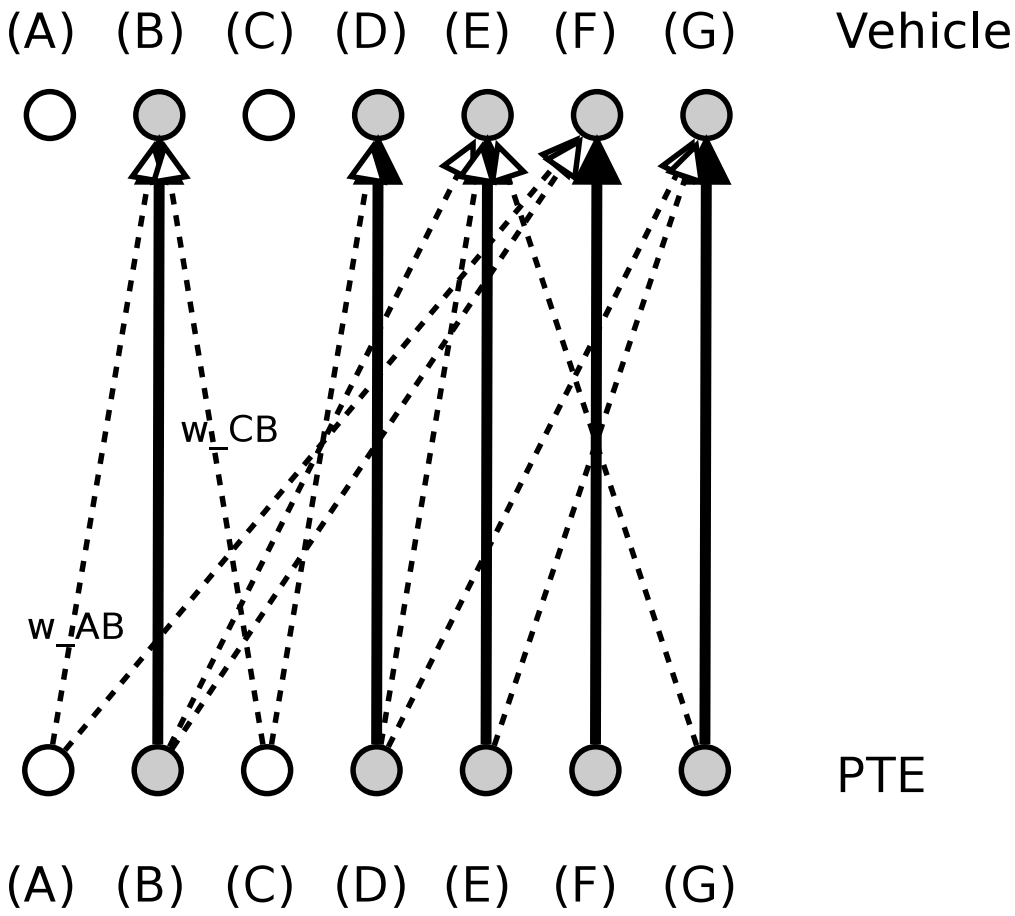
Determining carpool negotiation probablity

Estimating negotiation success probability



Periodic Trip to Vehicle Assignment





Periodic Trip to Vehicle Assignment

Linear problem statement for assignment

$$\forall i, j \in [0, N-1] : x_{i,j} \leq 1 \tag{8}$$

$$\forall i \in [0, N-1] : \sum_{j \in [0, N-1]} x_{i,j} = 1 \tag{9}$$

$$\forall j \in [0, N-1] : \sum_{i \in [0, N-1]} x_{i,j} \leq cap(v_j) \tag{10}$$

$$\forall i, j \in [0, N-1], i \neq j : x_{i,j} - x_{i,i} \leq 0 \tag{11}$$

i	identifies trip	Eqn(1)	$x_{i,j}$ are boolean
j	identifies vehicle	Eqn(2)	Each trip assigned to one vehicle
N	number of persons	Eqn(3)	Vehicle capacity is limited
		Eqn(4)	Car owner shall drive

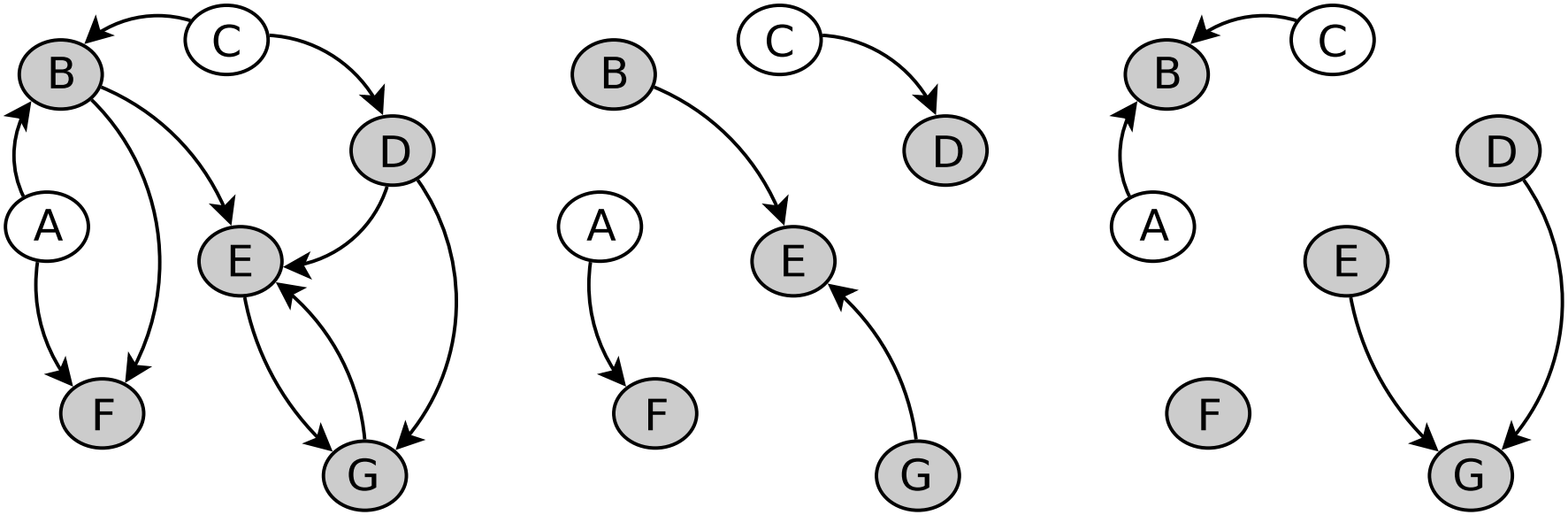
Periodic Trip to Vehicle Assignment

Problem size can become a problem

1. Problem size is large (Flanders : 6e6 people, 2.5e6 commuters)
2. 20% randomly (uniform) selected persons are assumed to show interest for carpooling
3. Dynamic
   1. Links are deleted and added by (de)registration
   2. Link weights evolve over time (by changing reputation and cohesion)
4. Matrix is sparse
5. Integer programming problem
6. Try to find solution starting from previous one

Periodic Trip to Vehicle Assignment

The assignment problem is a *star cover problem*



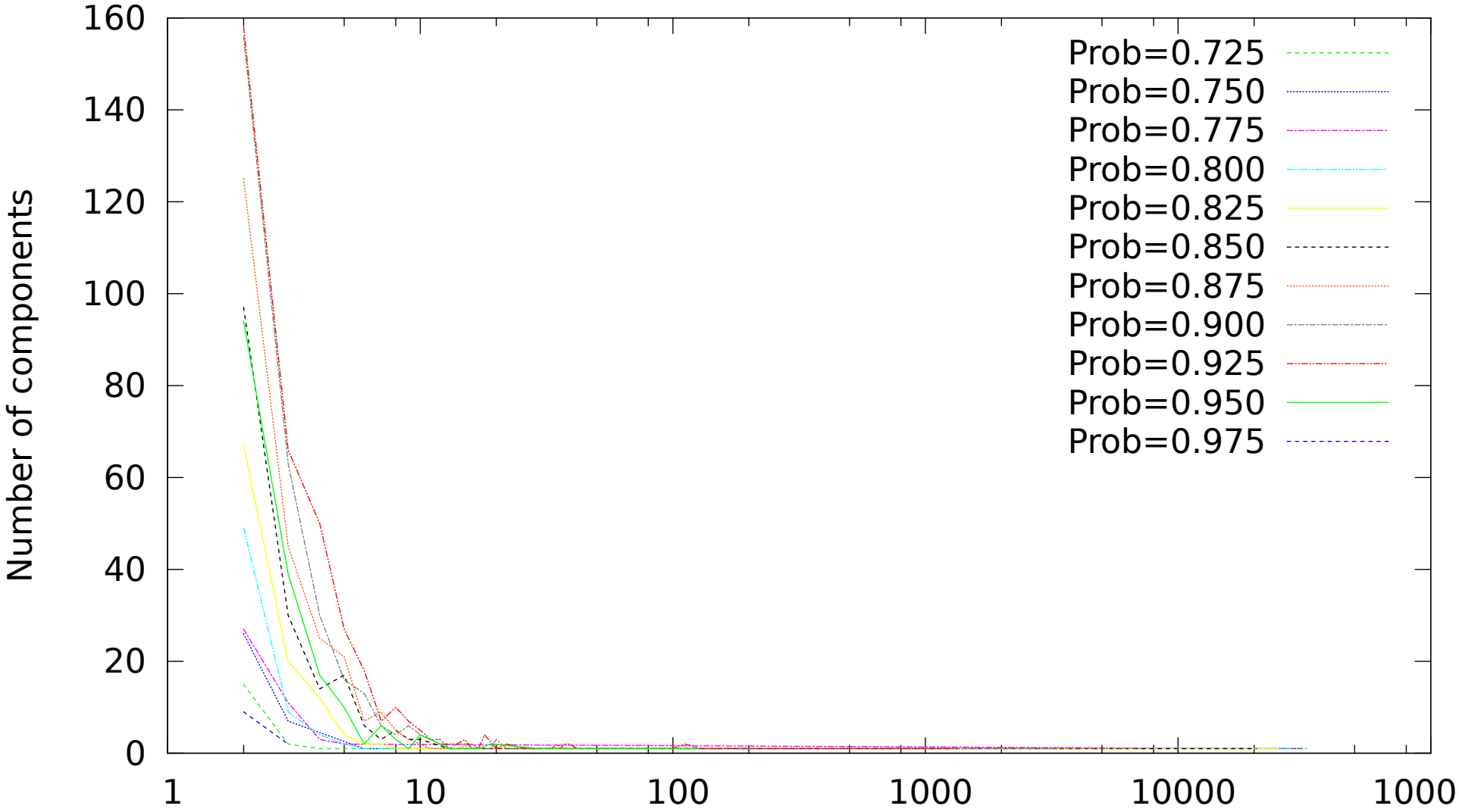
Original Cover 1 Cover 2

Graph Characteristics

Probab. Threshold	nVertices	nEdges	$\frac{nEdges}{nVertices}$	completeness	nComps
0.700	55297	2210379	39.97	7.229E-4	21
0.725	53882	1513769	28.09	5.214E-4	36
0.750	51517	995710	19.33	3.752E-4	45
0.775	47872	618861	12.93	2.700E-4	67
0.800	42519	356954	8.40	1.974E-4	114
0.825	35366	191860	5.42	1.534E-4	184
0.850	26464	95227	3.60	1.360E-4	258
0.875	16387	39104	2.39	1.456E-4	342
0.900	6603	10624	1.61	2.437E-4	403
0.925	1339	1615	1.21	9.014E-4	194
0.950	24	13	0.54	0.024	11
0.975	0	0			

Graph Characteristics

Component size frequency . . . misleading : see next slide



Graph Characteristics

Eight largest components in each graph

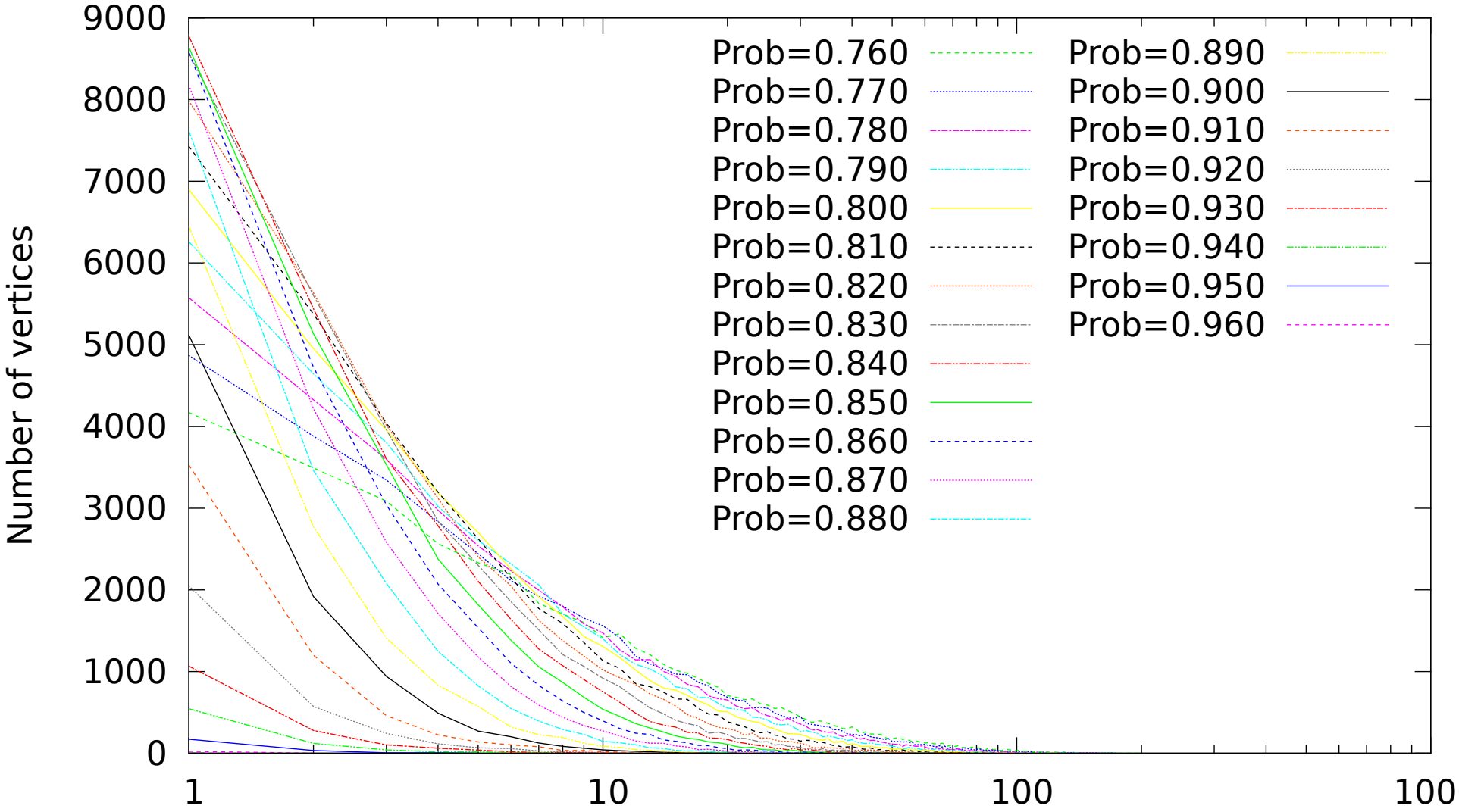
Prob	cs1	cs2	cs3	cs4	cs5	cs6	cs7	cs8
0.700	32096	23155	6	4	3	3	2	2
0.725	31335	22468	6	3	3	3	3	3
0.750	29969	21439	5	5	4	4	4	3
0.775	27888	19827	6	5	5	4	4	4
0.800	24688	17484	16	14	12	9	8	7
0.825	20388	14311	40	18	17	13	10	10
0.850	7154	5620	5288	3255	2459	1205	380	129
0.875	3262	2303	2053	1007	980	941	935	915
0.900	1303	601	523	265	230	219	217	175
0.925	127	120	108	62	57	55	45	38
0.950	3	3	2	2	2	2	2	2
0.975								

Graph Characteristics : Trivial Stars

Prob Threshold	nComp	nTrivial stars	Size 2	Size 3	Size 4
0.700	21	14	14		
0.725	36	25	25		
0.750	45	26	25	1	
0.775	67	47	47		
0.800	114	66	66		
0.825	184	99	97	2	
0.850	258	126	123	3	
0.875	342	154	153	1	
0.900	403	162	158	4	
0.925	194	97	94	2	1
0.950	11	9			
0.975	0	0			

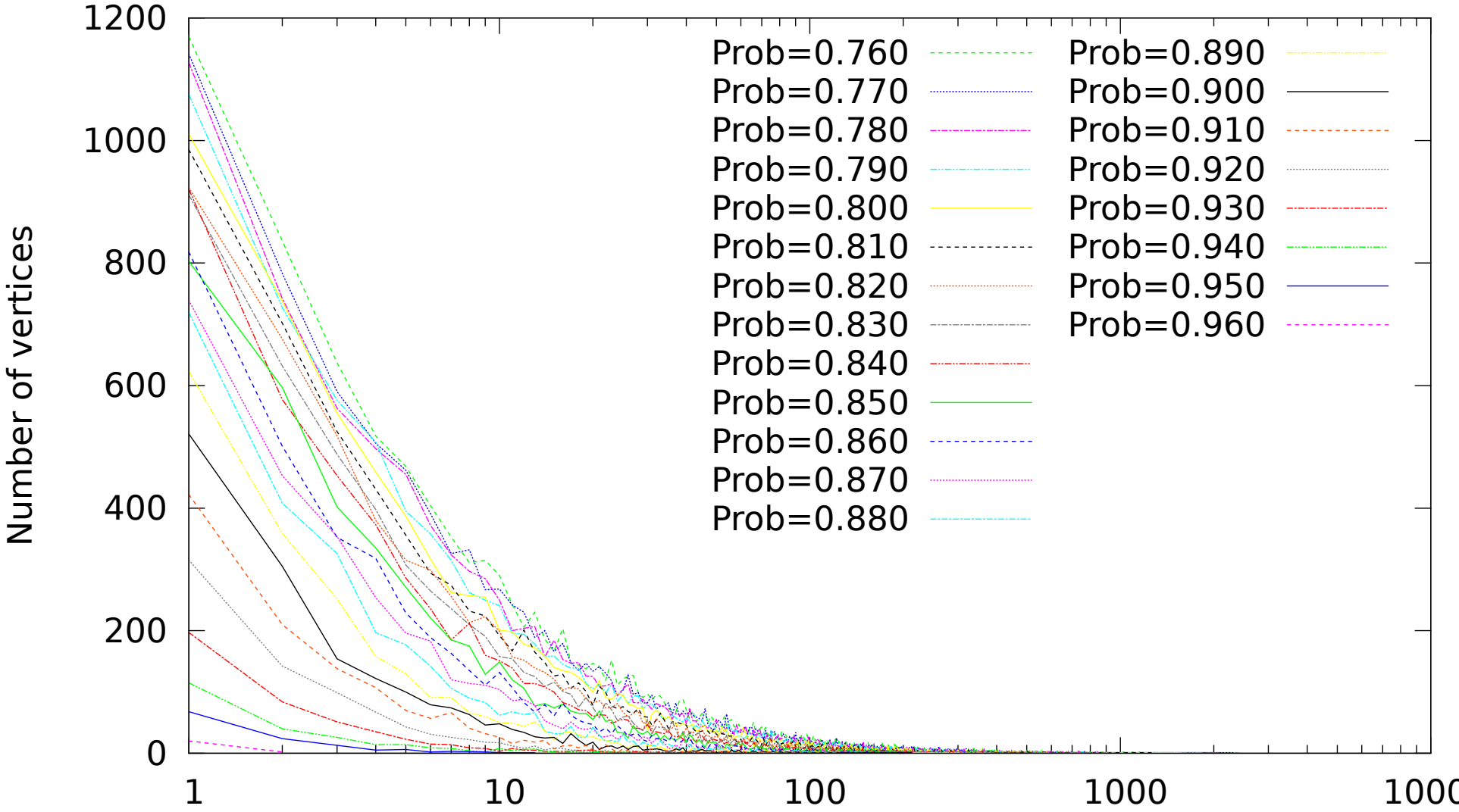
Graph Characteristics : Node Degrees

Vertex InDegree Frequency Distribution

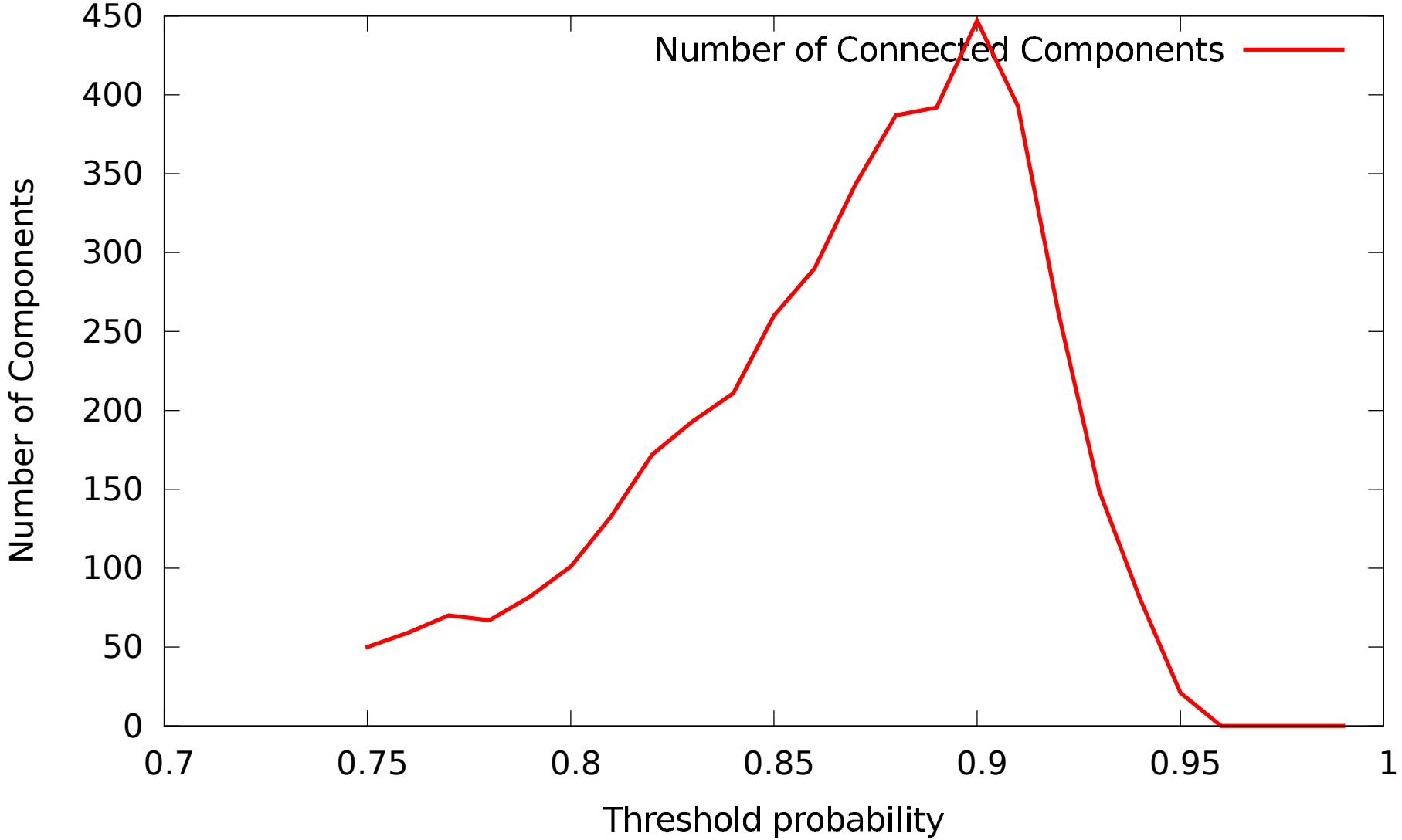


Graph Characteristics : Node Degrees

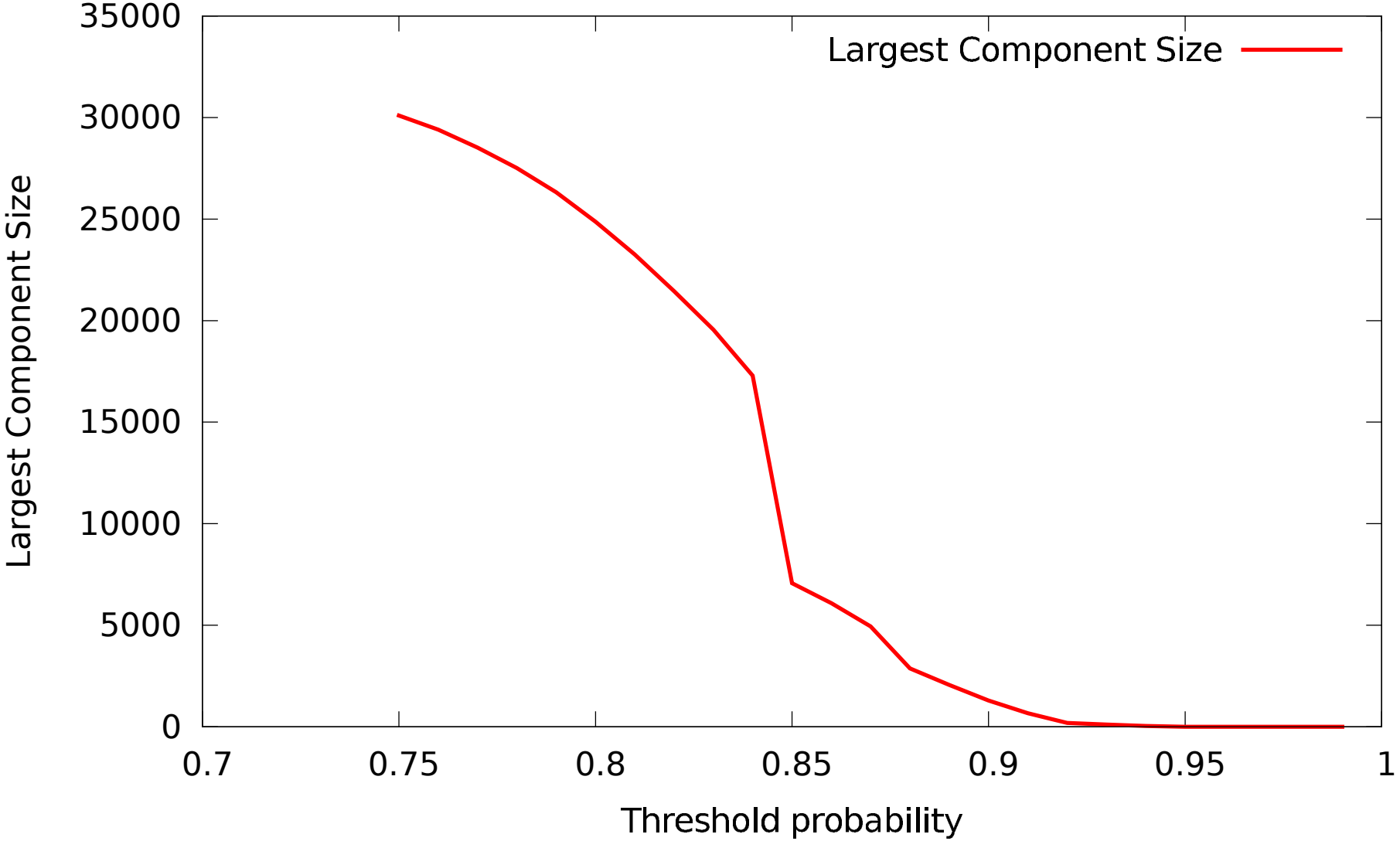
Vertex OutDegree Frequency Distribution



Graph Characteristics : Connected Components



Graph Characteristics : Connected Components



Conclusions

- Problem size causes non-trivial issues
- Partitioning is easy
- From threshold 0.85 on, *grid computing* is useful
- Allocate tasks to processors in decreasing size order (load balancing)