Introduction to Complex Networks Network Application Diagnostics B2M32DSA

Radek Mařík

Czech Technical University Faculty of Electrical Engineering Department of Telecommunication Engineering Prague CZ

October 17, 2023



Radek Mařík (radek.marik@fel.cvut.cz)

Introduction to Complex Networks

October 17, 2023

Outline

Complex Networks

- Practical Examples
- Software Tools
- Network Volume
 Netflow Comprehension
- Network Visualization
 - Data on the Ancient Egypt
 - Mainframe Assembly Comprehension
 - Enterprise people

2 Complex Network Introduction

- Graph Terminology
- Graph Algorithms

Outline

Complex Networks

Practical Examples

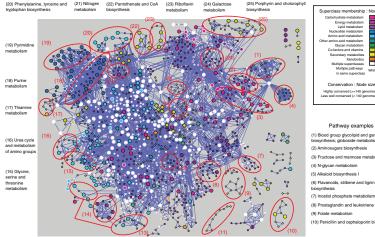
- Software Tools
- Network Volume
 - Netflow Comprehension
- Network Visualization
 - Data on the Ancient Egypt
 - Mainframe Assembly Comprehension
 - Enterprise people

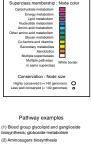
2 Complex Network Introduction

- Graph Terminology
- Graph Algorithms



Conservation within the global metabolic network [PASP09]





- (3) Fructose and mannose metabolism
- (4) N-glycan metabolism
- (5) Alkaloid biosynthesis I
- (6) Flavanoids, stilbene and lignin
- (8) Prostaglandin and leukotriene metabolism
- (9) Folate metabolism
- (10) Penicillin and cephaloporin biosynthesis

(14) Fatty acid biosynthesis pathway I

(13) Lysine biosynthesis and degradation

(12) Glutathione metabolism

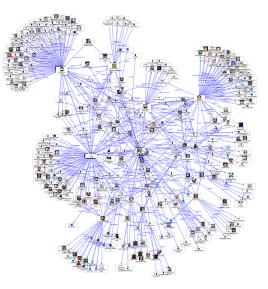
(11) Diterpenoid biosynthesis



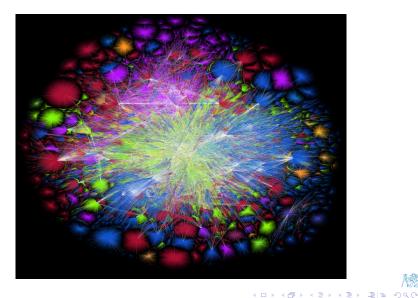
Radek Mařík (radek.marik@fel.cvut.cz)

October 17, 2023

Link Analysis of the Al Qaeda Terrorist Network [FMS]



Internet Map in 2015 [BI014, Opt17]





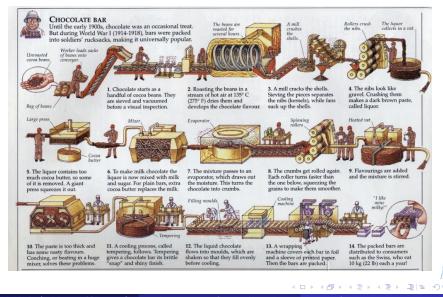
Radek Mařík (radek.marik@fel.cvut.cz)

Introduction to Complex Networks

October 17, 2023

< 円

Chocolate Making Process Dependencies [Fre14]

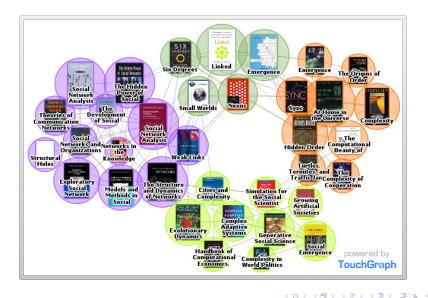


More Examples

- Biological networks
 - gene regulation networks
 - protein-protein interaction networks
 - metabolic networks
 - the food web
 - predator-prey relations
 - brain network
- Social networks:
 - networks of acquaintances
 - collaboration networks
 - phone-call networks
 - citation networks
 - opinion formation
 - society/community/party networks

- Technological networks:
 - the Internet
 - telephone networks
 - transportation networks
 - sensor networks
 - energy grid networks
- Informational networks:
 - the World Wide Web
 - Twitter
 - Facebook
 - peer-to-peer

SNA Books



M



Complex Networks

Practical Examples

Software Tools

Network Volume

- Netflow Comprehension
- Network Visualization
 - Data on the Ancient Egypt
 - Mainframe Assembly Comprehension
 - Enterprise people

2 Complex Network Introduction

- Graph Terminology
- Graph Algorithms

315

Approach to Complex Networks

- One needs to distinguish between analysis and production phases
- Some phenomena appear only with sufficiently large data volumes (emergent events)
- Volume
 - A number of suitable tools ... HDF5, ElasticSearch, Clouds
 - Capable to operate with terabytes of data
- Visualization
 - Critical if anomaly features are not known
 - At present, there is no obvious choice of a tool and a network layout given a particular problem.
 - Tools do not often scale with data volumes (> $10.000~{\rm nodes},~10^5~{\rm edges})$
 - GGobi, Pajek, NetworkX, SNAP, Tulip, Gephi, Cytospace, yEd, D3.js
 - Aspects: data volume, interactions with the user

Popular software packages [HLDS13]

- Analysis
 - UCINET (http://www.analytictech.com/ucinet.htm)
 - ENET (http://analytictech.com/e-net/e-net.htm)
 - **Pajek** (http://pajek.imfm.si/doku.php?id=pajek)
 - RSIENA
 - R
 - NodeXL
 - NetworkX ... a Python library
 - **iGraph** ... a C/Python library
- Visualization
 - yEd
 - Gephi
 - Cytospace
 - Tulip
 - NetDraw (2D, embedded in UCINET, see above)
 - Mage (3D, embedded in UCINET, see above)
 - visit www.netvis.org/resources.php for more

Outline

Complex Networks

- Practical Examples
- Software Tools

Network Volume

- Netflow Comprehension
- Network Visualization
 - Data on the Ancient Egypt
 - Mainframe Assembly Comprehension
 - Enterprise people

2 Complex Network Introduction

- Graph Terminology
- Graph Algorithms

315

NETFLOW Primary Statistics

Netflow

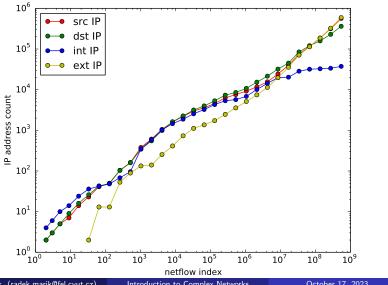
- Condensed records on a packet flow
- Several packets are merged into one netflow record
- Only 14-20 aggregated metrics

An enterprise traffic as a netflow sample taken during 9 days:

Statistics	Value
Total transported data volume	13,995,690,457,765 [B]
Packet count	20,131,367,095
Netflow count	617,326,053
IP address count	686,168
Source IP address count	614,150
Destination IP address count	392,881
Different P2P connections count	2,412,481

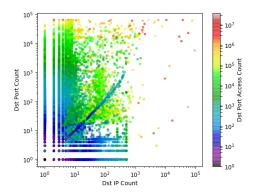


Is the Sample of IP addresses reprezentative?



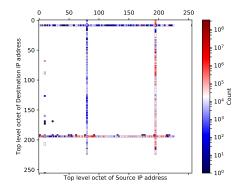
Radek Mařík (radek.marik@fel.cvut.cz)

A Data Projection Focused on Services



- Destination IP vs. destination port (space of services and their locations)
- Some counts of accesses are exceptional (red)

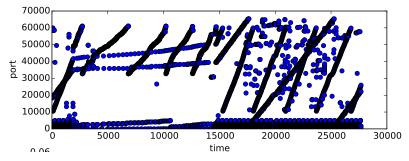
Top Level IP Network Projection - Data Sparsity



- Eocused on the network of source and destination IP addresses
- Top level octets of IP addresses (160.30.29.17 \implies 160)
- A very sparse space
- A rather restricted source-destination IP connections (as expected)



Port Scanning from xxx.xxx.18.120 - Logical Time Progress



• 617,326,053 netflows pprox 60,000 samples imes sample size 10.000

- ullet \Longrightarrow 60,000 samples might be still visualized with difficulties
- \implies 1.000 events can be easily missed with 10,000 sample size

Complex Networks Network Volume

Masters of Social Network Analysis [RP13, Weh13]



- US National Security Agency
- Maintains large programs in social network analysis
- Believed to process 2×10^{10} node and tie updating events per day
- Result:
 - "Better Person Centric Analysis"

Image: A matrix and a matrix

Types

- **94 entity/node** types (phone numbers, e-mail addresses, IP addresses, etc.)
- **164 relationship** types to build "community of interest" profiles (*travelsWith*, *hasFather*, *sentForumMessage*, *employs*, etc.)

000 E E 4 E + 4 E

Complex Networks Network Volume

Masters of Social Network Analysis [RP13, Weh13]



- US National Security Agency
- Maintains large programs in social network analysis
- Believed to process 2×10^{10} node and tie updating events per day
- Result:
 - "Better Person Centric Analysis"

Types

94 entity/node types

(phone numbers, e-mail addresses, IP addresses, etc.)

• **164 relationship** types to build "community of interest" profiles (*travelsWith*, *hasFather*, *sentForumMessage*, *employs*, etc.)

Outline

Complex Networks

- Practical Examples
- Software Tools
- Network Volume
 - Netflow Comprehension

Network Visualization

- Data on the Ancient Egypt
- Mainframe Assembly Comprehension
- Enterprise people

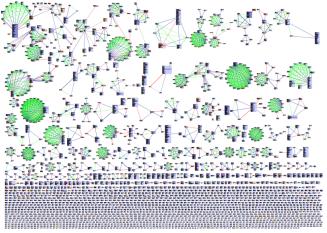
2 Complex Network Introduction

- Graph Terminology
- Graph Algorithms



글 노

Egypt Data - Family Recognition



circular layout (yEd)

A family:

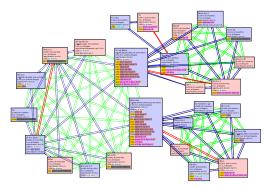
- Using family designation
 - husband, wife, son, etc.
- A connected graph component
- Sparse data assumed
- Transformed into family tree using marriage nodes



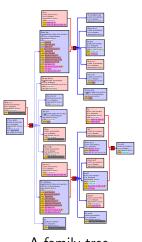
Complex Networks Network

Network Visualization

Egypt Data - Transformation into Family Tree



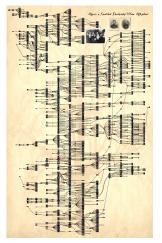
A family as a connected component circular layout (yEd)



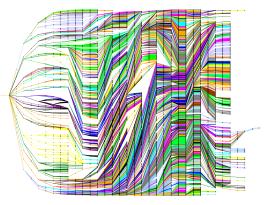
A family tree hierarchical layout (yEd)

Network Visualization

Family Trees^[Mar17]



multitree-like tree driven layout, Graphviz



- Taxonomic information ITIS on plants, animals, fungi, and microbes,
- A phylogenetic tree with 945.352 nodes
- multitree-like tree driven layout



Network Visualization

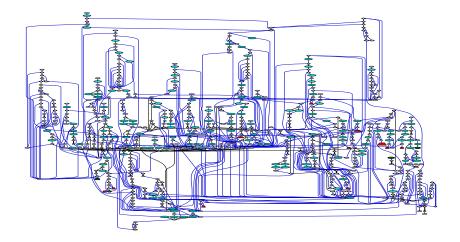
HLASM Mainframe Assembly

4 Loc. Object Code Addr1 Rddr2 Stnt Source Statement HLBM PK5.0 2008/04/18 03.22 + 000209 1878	✓ Active Usings: IHIFSARA,R13											
+ 000280 1838 646 RR R3,PET PODRSS OF PET-ENTRY 11980000 + 000280 2910 2006 00006 647 TM 6(R3),BETREN PROCEDURE CALLED 12000001 + 000280 4780 1284 00204 648 BZ PROLUCI NO 1220200001 + 000290 4780 1284 00200 00000 649 C ADR,ASTLOC(D,FSA) CMP.CONT.OF ADR.HITH ADDR.OF 12040000 + WR RSMR044E Undef ined symbol - RSTLOC *** RSMR044E Undef ined symbol - RSTLOC *** FUNCTIONNELUESTORAGE 12060000 + 000290 0000 0000 00000 651 BE 0ERR21(0,FSA) *** RSMR04E Undef ined symbol - CRR21 12060001 + WR RSMR45E IAecord 621 in K0TEM01.08710LRSM (IHFSA) on volume: TSUD11 *** *** RSMR04E Undef ined symbol - 0ERR21 2200000 + 000290 0000 0000 00000 652 TM 673,CODEPM CODE PROCURE CALLED 12100001 + 000280 4710 047A 653 BD PROLUC2 YES 12240000 + 000284 4803 0004 00004 654 PROLUC2 <td></td> <td></td> <td></td> <td></td> <td>Stirt Source</td> <td>e Sta</td> <td>tenent</td> <td></td> <td>HLASH R5.0 2008/04/</td> <td>/18 03.22</td>					Stirt Source	e Sta	tenent		HLASH R5.0 2008/04/	/18 03.22		
+00229 2102 2002 2102 2002 2102 2002 2102 2002 2102 2002 2102 2002 2102 2002 2102 2002 2102 2002 2102 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 <					646	AR	R3.PBT					
+00209 4700 12820000 +00294 0000 649 EZ PROLUÉ1 NO 12820000 +00294 0000 649 C ARR,AGTLOCL,FGR CMP.CONT.OF ADR.HITH ADDR.OF 12040000 +*** RSMH451 Record 619 IN KTEMDL.RSMCIHIFSR on volume: TSUD11 + +*** RSMH44E Indef ined symbol - RSTLOC EDER21(0,FSR) FRINCTIONIRLUESTORAGE 12060000 +*** RSMH34E Execrd 621 INKTEMDL.RSTLOL,RSTLOL,RST FRINCTIONIRLUESTORAGE 12060000 +*** RSMH34E Execrd 621 INKTEMDL.RSTLOL,RST FRINCTIONIRLUESTORAGE 12060000 +*** RSMH35E Record 621 INKTEMDL.RSTLOL,RST FRINCTIONIRLUESTORAGE 1200000 +*** RSMH35I Record 621 INKTEMDL.RSTLOL,RSTLOL,RST ENCHT 12200000 +0029C 9103 3006 00006 652 TM 6(R3),COEFRM COEF PROCOURE CALLED 12200000 +0028A 430 GS5 LR F4,80R SHF B80		۵	0006									
+002894 0000 0000 00000 649 C ADR,ASTLOC(0,FSR) COMP.CONT.OF ADR.HITH ADDR.OF 12040000 **** RSMH044E Undefined symbol - RSTLOC - RSTLOC <td< td=""><td>+000290 4780 D284</td><td>-</td><td></td><td>00284</td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	+000290 4780 D284	-		00284								
*** R9HB44E Undef inde symbol - RSTUC *** R9HB45E Record 619 in K0TEH01.CB7310.RSH(IHIF98) on volume: TSUD11 *00289 0000 0000 0000 00000 651 BE 0ER21(0,FSA) *** R9HB45E Record 621 in K0TEH01.CB710.LG8H(IHIF98) on volume: TSUD11 *** R9HB44E Undef inde symbol - 0ER21 *** R9HB45E Record 621 in K0TEH01.CB710.LG8H(IHIF98) on volume: TSUD11 *000209 110 3006 652 TH 6(R3).00EPFM 000209 010 3006 652 TH 6(R3).00EPFM *00029A 4710 047A 653 B0 PR0.062 YES *0028A 8004 0004 654 PR0.051 LH R0.4(R3) LERKTH 0F DSA TO REG 0 1210000 *0028A 804 0004 655 LR R4,80R SANE DRE DURING ETHIN 12160000 *0028A 819F 655 LR R4,80R SANE DRE DURING ETHIN 12160000 *00028A 180F 659 * SVC 10 EFMEIN R,1V=(0) *00028A 4510 D2AE 0028E 658+ BRL 1,**4 INDICHTE GETMAIN 8230ENG 01-6ETHA *00028E R000 00000 661 L R0,0(R2,PT) L0AD 01NTER 0F LAST GENERATION 12220000 *00028A 5000 1000 00000 662 ST R0,0(0,1) ANN STORE PIDTER 0F EMBRCTMG PB. 2220000 *00028A 400 1004 00004 663 ST CGA,4(0,1) STORE PIDTER 0F EMBRCTMG PB. 2220000 *00028A 2000 1000 00000 666 WV 110(R1,Y,Y00' ZEROS TO VALUE ARRY NAD *00028C 5000 1000 00000 667 ST R1,0(R2,PT) L0AD 01NTER 0F EMBRCTMG PB. 2220000 *00028C 2010 1000 00000 667 ST R1,0(R2,PT) L0AD 01NTER 0F EMBRCTMG PB. 2220000 *00028C 4000 1000 00000 667 ST R1,0(R2) STORE PDTIERE 0F EMBRCTMG PB. 2220000 *00028C 4000 1000 00000 667 ST R1,0(R2,PT) STORE PDTIERE 0F EMBRCTMG PB. 2220000 *00028C 4000 1000 00000 667 ST R1,0(R2,PT) STORE PDTIERE 0F EMBRCTMG PB. 2220000 *00028C 4000 00000 667 ST R1,0(R2,PT) STORE PDTIERE 0F EMBRCTMG PB. 2220000 *00028C 4000 00000 667 ST R1,0(R2,PT) STORE PDTIERE 0F EMBRCTMG PB. 2220000 *00028C 4000 00000 667 ST R1,0(R2,PT) STORE PDTIERE 0F EMBRCTME PB. 1220000 *00028C 4000 00000 667 ST R1,0(R2,PT) STORE PDTIERE 0F EMBRCTME PB. 1220000 *00028C 4000 00000 667 ST R1,0(R2,PT) STORE PDTIERE 0F EMBRCTME PB 12240000 *00028C 4000 00000 667 ST R1,0(R2,PT) STORE PDTIERE 0F EMBRCTME PB 12240000 *00028C 4000 00000 666 WV 1115(R2,R1,10(R2,PT) STORE PDTIERE 0F EMBRCTME PB 12240000 *00028D 2900 10000									COMP.CONT.OF ADR.HITH ADDR.OF			
+*** RSHR4351 Record 619 in KOTEHO1.08710.RSH(IHLFSR) on volume: TSU011 *** *FINCTIONIALUESTORRGE 1206000 +000298 0000 0000 00000 651 BE 0ERR21(0,FSR) #FINCTIONIALUESTORRGE 1206000 **** RSHR451 Record 621 in KOTEHO1.087101.RSH(IHLFSR) on volume: TSU011 *** #FINCTIONIALUESTORRGE 12060000 **** RSHR451 Record 621 in KOTEHO1.087101.RSH(IHLFSR) on volume: TSU011 *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** **** **** **** **** ******************* *************************	→** ASHAO44E Undef	ined s	иньо 1	- 8STI	00							
+ 650 * HTINCTIONNIFULESTORAGE 1206000 +00298 0000 0000 0000 051 BE 0ER21(0,FSA) BRANCH IF EQUAL 12080000 *** RSMH34E Undef ined symbol - 0ER821 NOTEND - 0ER821 12080000 12080000 *** RSMH35E Record 621 in K0TEND1.0ER101.0BS101.RSM(IHEFSA) on volume: TSUD11 *** *** 12100001 *00280 2910 3006 00006 652 Th 6(R3),CODEPM CODE PROCURE CALLED 12100001 *00280 2910 3006 00004 654 PRULO12 VES 121200001 *00280 4903 0004 0004 654 PRULO12 H RA,48R SANE BR CMEINS EEMAIN 121200001 *00280 4501 02A 655 LR R4,88R SANE BR CMEINS EEMAIN 12160001 *00284 403 0004 00246 654 PRULO12 LR R4,88R SANE BR CMEINS EEMAIN 12160000 *00284 450 102A 658 + BRL 1,**4 INICISTE EEMAIN 12200000 12200000 *00282 5002 1000 000000 661 L R0,10R,PET IOAD POINTER OF LENET EEMAIN 12200000 *000282 5000 1000 000000 6	→** ASHA4351 Recor	d 619	in KO	TEHO1.0	BT310.ASH(IH	IESA)	on volume: TSU011					
++++ R9H044E Undef ined symbol - 06R821 0.651 BE 0ERR21(0,FSA) BRANCH IF EQUAL 12080000 ++++ R9H044E Undef ined symbol - 06R821 ++++ R9H044E Undef ined symbol - 06R821 ++++ R9H045E Undef ined symbol - 06R821 +++++ R9H045E Undef ined symbol - 06R821 ++++++ R9H045E Undef ined symbol - 06R821 ++++++++++++++++++++++++++++++++++++							on forence footaa		*FUNCTIONVALUESTORAGE	12060000		
+** 83M9044E Undefined symbol - 0ER821 *** 83M9435I Record 621 in KDTENUL CBS10L,RSM(IHIFS8) on volume: TSU011 +000280 410 3006 00006 652 TH 613,000EPRM CODE PROCOURE CALLED 1210000 +000280 4710 3006 00006 653 B0 PROLOS2 YES 12120000 +000280 4803 0004 00004 654 PROLOS1 LH R0,4(R3) LENGTH OF DSA TO REG 0 12140000 +000284 4803 0004 00004 655 LR R4,8RR SAME RS WE BO R01KIN GETMAIN 12160000 +000284 4510 124E 0028E 658 B4L 1,**4 INDICATE GETMAIN FOR DSA 12180000 +000284 4510 124E 0028E 658 B4L 1,**4 INDICATE GETMAIN FOR DSA 12280000 +000280 1874 660 LR B6R,84 INDICATE GETMAIN 80C 01-6ETMA +000280 1874 660 LR B6R,84 INDICATE GETMAIN 80C 01-6ETMAIN +000280 1874 660 LR B6R,94 INDICATE GETMAIN 8220000 +000285 5000 1000 00000 661 L R0,0(R2,P8T) LOAD FOINTER OF LAST GENERATION 2220000 +000286 5000 1000 00000 663 ST COS9,4(0,21) STORE FOINTER OF ENDERCING FB. 12260000 +000286 4020 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER OF LAST GENERATION 2220000 +0002025 5020 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER OF LAST GENERATION 2230000 +000205 6200 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER FINTENT 2230000 +000205 6204 1008 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER NET 12240000 +000205 6204 1008 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER NET 12240000 +000205 6204 1008 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER NET 12240000 +000205 6204 1008 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER NET 12240000 +000205 6204 1000 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER NET 12240000 +000205 6204 1000 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER NET 12240000 +000205 6204 1000 1000 00000 665 MVI 10(R1), Y00' ZER80 FOINTER NET 12240000 +000205 6204 1000 10000 666 MVC 1115, R1, 10(R2)	+000298_0000_0000+			החחחח		BF	OFRR21(0.FSR)					
+** RPH4951 Record 621 in K0TEH01.087310.RSM(THLFSR) on volume: TSU011 +000290 9110 3006 00006 652 Th 6(R3).c00EPRM CODE PROCURE CALLED 12100001 +000290 9110 3006 00004 653 B0 PROLUE2 YES 121200001 +000294 4033 0004 0004 654 PROLUE2 YES 121200001 +000294 4033 004 0004 654 PROLUE2 YES 121200001 +000296 8149 655 LR R4,08R SAHE BRR DURING EMAIN 12160000 +000296 654 FRHL 1,**4 INICIATE CETMEIN R220000 +000296 600 LR BRR,R4 12200000 12200000 +000296 5001 00000 661 L R0,018,PET LOAD POINTER OF LAST GENERTION 12200000 +000292 5001 00000 663 ST COBA, 40,R11 STURE PET DISPLACEMENT 12200000 +000		ined s										
+00029C 9113 3006 00006 652 TM 6(83) CODE PROCURE CALLED 12200001 +000290 4710 047A 024A 4830 LENKTH OF DSR TO RES 0 12120000 +000288 184F 655 LR R4,88R SRVE B8R DIRTHS CETMEIN R,144(80) 121800001 121800001 +000284 4510 D24E 658 BH 1,844 1001CRTE 121800001 121800001 +000284 640 LR B8R,84 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001 12200001						TESA)	on volume: TSU011					
+000280 4701 0474 653 B0 PROLOÉ2 YES 12120000 +000284 4803 0004 0004 654 PROLOÉ2 YES 12140000 +000284 4803 0004 0004 655 LR R4, 4803 LENGTH OF DGS TO REG D 12140000 +000284 946 655 GETMAIN R, LV=(0) GETMAIN FOR DGS 12140000 +000284 0261 0268 BAL 1,**4 INDICATE GETMAIN 12180000 +000280 1974 660 LR 80,874,874 12200000 1-6ETMAIN +000280 1974 660 LR 80,016,21 LAR 1,894 12200000 +000282 5000 00000 661 L R0,016,21 MOB POINTER OF LAST GENERATION 12200000 +000286 5000 1000 00000 663 ST C069,410,411 STRE POINTER OF LAST GENERATION 12200000 +000286 5000 1000000 663 ST									CODE PROCOURE CALLED	1210000		
+000284 4003 0004 654 PR0L051 LH R0,4(R3) LENGTH OF DSA TO REG 0 12440000 +000284 194F 655 LR R4,6RA SHUE BRA DURING GENTAINI 12160000 +000284 194F 655 LR R4,6RA SHUE BRA DURING GENTAINI 12160000 +000284 4510 D24E 658+ BRL 1,*+4 INDICATE GETHRIN 2230000 +000284 D44 660 LR BR8,R4 12200000 12200000 +000280 S001 000000 661 L R0,0(R,PGT) DAPO POINTER OF LAST GENERATION 12200000 +000280 S001 000000 661 L R0,0(R,PGT) DAPO POINTER OF LAST GENERATION 12200000 +000286 S001 00000 663 ST C06,4(1,1) STIRE POINTER OF ENSRCING PB 12220000 +000286 S001 00004 663 ST C06,4(1,1) STIRE POINTER OF ENSRCING PB 12200000 +000282 29201 1008 <td></td> <td></td> <td></td> <td>00478</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>				00478								
+000288 184F 655 LR P4/B2R SRVE B2R DUE1NG GETMAIN 121600001 * 666 GETMAIN R, LV=(0) GETMAIN P00204 GETMAIN 121800001 *000284 4510 D24E 658 BHL 1, 4×44 INOLOTE GETMAIN 221800001 *000284 GRM 659 SW0 10 INOLOTE CETMAIN 9220EN00 01-GETMAIN *000280 D800 0000 661 L R0, D162, PBT) LOAD POINTER OF LAST GENERATION 22200000 *000286 S000 00000 662 ST R0, R, 40, P1) HW STORE IT IN DGR 22200001 *000286 S000 1004 00004 63 ST CGR, 40, P1) STORE FDI TRO F EMBRIN BP, P1, 2220000 *000286 S000 1004 00004 663 ST R0, R, 40, P1) STORE FDI TRO F EMBRIN BP, P1, 2220000 *000286 S000 00004 665 WT 101(R1, R), 10''''''''''''''''''''''''''''''''''''												
→ 655 GETMEIN É,LV=(0) GETMEIN FOR DSR 12180000 +0002RE AGD 002RE 658+ BRL 1,4+4 INDICATE GETMEIN ECTION 0210-6ETME +0002RE AGDA 669+ BRL 1,4+4 INDICATE GETMEIN ECTION 021-6ETME +0002RE AGDA 660 LR BRR,P4 1350LE GETMEIN SUC 01-6ETME +0002RE SUI2 B000 00000 661 L R0,010,21 HON DETTE OF LAST GENERATION 12200000 +0002RE SUI2 B000 00000 662 ST R0,010,P1 HON DETTE ND SR T1220000 H2200000 +0002RE SUI2 D100 00000 663 ST CDSR,4(0,P1) STRE POINTER OF LAST GENERATION 12200000 +0002RE 4020 1000 00000 663 ST CDSR,4(0,P1) STRE POINTER OF EMBRACING PB. 12260000 +0002RE 4020 1000 00000 664 STIT R2,8(1,1) STRE POINTER NET 12280000 +0002RE 5020 1000 00000 666 HVC 11(S,R1),10(R1) +MRRHY POINTERS 12320000 +0002CE 5220 0000 000000 667 <td></td>												
•0002RA 4510 D2RE 002RE 658+ BRL 1,**4 INDICATE 6ETMAIN 0230EMG 01-6ETMA •0002RD 000R 659+ SVC 10 ISSUE 6ETMAIN SVC 01-6ETMA •0002RD 0084 600 LR BR8,R4 12200000 12200000 •0002RD 5002 5002 0000 000000 661 L R0,0182,PBT LOAD POINTER OF LAST GENERATION 12220000 •0002RD 5000 1000 000000 662 ST R0,010,P21 AND STORE PDINTER OF LAST GENERATION 12220000 •0002RD 6000 1000 00004 663 ST C06N, 4(0,R1) STORE PDINTER OF EMERCING PB. 12260000 •0002RD 6000 1000 00004 663 ST C26N, 4(0,R1) STORE PDINTER OF EMERCING PB. 12260000 •0002RD 6000 1000 00004 665 MVI 10(1), X'00' ZEROS TO ANLE CARENT 12280000 •0002CE 612U 0000 00000 666 MVI 10(1), X'00' ZEROS TO ANLE CARENT 12280000 •0002CC 5012U 0000 00000 6667 ST R1,0(R2,FRT) STORE CARENTY POINTER 12280000 •0002CC 5012U 0000 00000 6667 ST R1												
+0002RE 0000 659+ SVC 10 ISSUE GETMAIN SVC 01-GETMA +0002RE 0874 660 LR BRR,R4 12200000 +0002RE 0800 00000 661 L BRR,R4 12200000 +0002RE 0800 00000 661 L R0,010,R2,PBT L0AD PDINTER OF LAST GENERATION 12220000 +0002RE 5000 1000 00000 663 ST C06A,410,P11 STORE PT DINTER OF ENBRICING PB. 12260000 +0002RE 4020 1008 00008 664 ST R2,810,R11 STORE PT DINTER OF ENBRICING PB. 12280000 +0002RE 4020 1008 00008 664 NVI 11(R1,X ⁺ N0 ⁻⁷) ZENSIT TOHUE RRRY POINTER 12300000 +0002RE 2004 00000 665 NVI 110(R1,X ⁺ N0 ⁻⁷) ZENSIT TOHUE RRRY POINTERS 12320000 +0002RE 2000 00000 667 ST R1,0182,PGT STOR	+000288_4510_D28E			0028E								
+000280 19F4 660 LR BRR, P4 12200001 +000285 S000 60000 661 L R0, 0(R2, P0T) LOAD POINTER OF LAST GENERATION 122200001 +000285 S000 1000 662 ST R0, 0(R2, P0T) LOAD POINTER OF LAST GENERATION 122200001 +000286 S000 1004 00000 663 ST CDSA, 4(0, P1) STRE POINTER OF LAST GENERATION 122200001 +000286 A0001 1004 00004 663 ST CDSA, 4(0, P1) STRE POINTER OF ENBRACING PB. 12260000 +000226 4020 1004 00004 664 STH R2, 8(0, R,1) STRE POINTER OF ENBRACING PB. 12280000 +000226 1004 1000 666 HVC 11(S, R1), 100'' ZEROED OF UNLUE RERY HON 12320000 +000205 1204 1008 00000 667 ST R1, 0(R2, P0T) STME CHR, LOSA POINTER 12340000 +000205 1204 1008 LCR CDSR, R1 ST <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>												
+00228 S012 0000 000000 661 L R0_0(r2,PET) LOAD POINTER OF LEWS GENERATION 12220000 +00286 S000 1000 00000 662 ST R0_0(0,K1) AND STORE IT IN DSR L2240000 +00286 S000 1000 00004 663 ST C6,R4(0,K1) STORE PD INTER OF EMBRACING PB. 12240000 +00286 S000 1004 00004 663 ST C6,R4(0,K1) STORE PD INTER OF EMBRACING PB. 12280000 +00282 S010 1008 00004 664 STN R2,8(0,R1) STORE PD INTER OF EMBRACING PB. 12280000 +001202 S012 0108 000008 665 MVI 101(81), X'00' ZEROS TO ANGLE CARRY POINTER 12800001 +001202 5024 1008 000008 666 MVC 115, R1, J0(R1) #RRRY POINTERS 122200000 +001202 5024 1008 000000 667 ST R1, 018, C18, PD INTER 12240000 +001202 5024 1008 000000 667 ST R1, 018, R1 STORE FRITER 12240000 +001202 5124 1001 668 LR C05R, R1 STOR POINTER												
+002286 5000 1000 00000 662 ST R0,000.61 HND STORE IT IN DSR 12240000 +000286 SDM0 1004 00004 663 ST COG,4(0,k1) STURE PDINTES OF HERGTING PB. 12260000 +000286 SDM0 1008 664 ST R2,6(0,k1) STURE PDINTES OF HERGTING PB. 122800001 +000282 SQM0 00000 665 MVI 10(k1), X'00' ZERGS TO VALUE ARRAY AND 122800001 +000282 SQM0 00000 666 MVI 11(k1), X'10' ZERGS TO VALUE ARRAY AND 122800001 +000282 SQM0 00000 666 MVI 11(s,k1), 10(k1) +#RRAY POINTERS 12380000 +000282 SQM0 00000 667 ST R1,0(k2,P8T) STURE CURR.OS POINTER 12380000 +000281 SQM0 SQM0 SQM0 SQM0 12380000 12380000 +000282 SQM2 SQM1 SQM1				הההה					LOAD POINTER OF LAST GENERATION			
+000289 50H0 1004 00004 663 ST CD\$A,4(0,K1) STORE PDINTER OF EHBRACING PB. 12260000 +000286 4020 1008 00008 664 STH K2,8(0,K1) STORE PBT DISPLICATION 12200000 +000229 20100A 00000 665 NVI 101(K1),Y00 ZERST DU HULLE RRRY AND 12200000 +000205 2014 1008 100A 00008 666 HVC 11(5,K1),10(R1) *ARRAY POINTERS 12320000 +000205 5024 1008 100A 00000 667 ST R1,0(R2,FRT) STORE CORR.05A POINTER IN PBT 12340000 +000205 S024 2000 80000 667 ST R1,0(R2,FRT) STORE CORR.05A POINTER IN PBT 12340000 +000205 S026 R010 00010 669 STM FBT,LRT,16(CDSA) 12380000												
+00028E 4002 2000 664 STH F2.8(0,F1) STREE PBT DISPLACEMENT 12280000 +000202 9200 1004 00004 665 HV 10(R1), K'00' ZEROS TO VALUE ARRYA NAD 123800001 +0002026 1204 1008 00000 666 HV 11(R1), K'10' +RRRY VOITERS 123800001 +000205 5012 8000 667 ST R1,0(R2,R)T STORE STORE 123800001 +000205 5012 8000 00000 667 ST R1,0(R2,R)T STORE STORE 123800001 +000205 5012 8000 668 LR CDSR, R1 STORE ST 123800001 +000205 9014 668 STH F8T,LRT,16(CDSR) 123800001 123800001						ŠŤ						
+0002C2 9200 100A 0000A 665 HVI 10(R1),X'00' ZEROS TO VALUE ARRAY AND 12300000 +0002C6 R204 100B 100R 0000B 6000A 666 HVC 11(5,R1,10(R1) +4RRAY POINTERS 1222000 +0002CC 5012 8000 00000 667 ST R1,0(R2,P6T) STORE CURR,DSA POINTER IN PBT 12340000 +0002D1 20841 668 LR CCGA,R1 SET COSA POINTER 12340000 +0002D2 90BC A010 00010 669 STH PBT,LAT,16(CDSA) 12380000												
+000205 0204 1008 100A 00008 0000A 666 HNC 11(5,Ri),10(R1) *ARRAY POINTERS 12320000 +000205 5012 8000 00000 667 ST R1,0(R2,PRT) STURE CURR.DSA POINTER IN PBT 12340000 +000201 290BC A010 00010 668 LR CDSA,R1 ST DSA POINTER 12360000 +000201 290BC A010 00010 669 STH PBT,LAT,16(CDSA) 12380000		n										
+0012CC 5012 8000 00000 667 ST k1.0(R2,F)FT STORE CURR.DSH POINTER IN PBT 12340000 +002001 1841 668 LR CL56,R1 SEE CD5A POINTER 12360000 +001202 090C A010 00010 669 STM PBT,LR1,16(CD5A) 12380000				AUUUU								
+000200 1881 668 LR CD\$A,R1´ SET CD\$A POINTER 12360000 +000202 908C A010 00010 669 STH P8T,LAT,16(CD\$A) 12380000												
+000202 90BC A010 00010 669 STH PBT,(AT,16(CDSA) 12380000												
				00010								
	+000206 0000 0000			00000	670	Ĩ	STH,RASPT(FSA)		RAS-POINTER TOP	12400000		

24 / 64

문 문

CHALLENGE: Complex Control Flow, a typical case

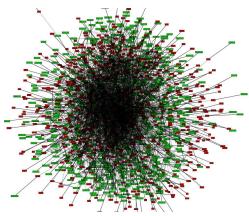


layered layout - Graphviz dot

Radek Mařík (radek.marik@fel.cvut.cz)

Introduction to Complex Networks

Dependancy of External Symbols in Mainframe Assembly Software

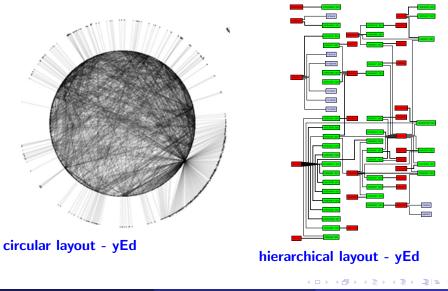


Fruchterman-Reingold force-driven layout

- A software product ... over 10.000.000 lines of code
- Over 400 modules . . . red
- External symbols . . . green
- Thick line ... the definition of a symbol
- Thin line ... a reference to a symbol
- ٥
- Where should the developer start with a bug analysis?

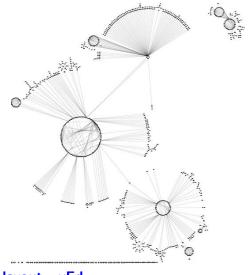
▲ 글 ▶ _ 글|님

Different Graph Layouts



Radek Mařík (radek.marik@fel.cvut.cz)

Assembly Software - Recovered Architecture





Radek Mařík (radek.marik@fel.cvut.cz)

Complex Networks Network

Network Visualization

Company Network of People - 3D Hyperbolic Tree Layout (Walrus)



29 / 64

315

Outline

Complex Networks

- Practical Examples
- Software Tools
- Network Volume
 - Netflow Comprehension
- Network Visualization
 - Data on the Ancient Egypt
 - Mainframe Assembly Comprehension
 - Enterprise people

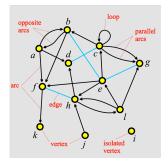
2 Complex Network Introduction

- Graph Terminology
- Graph Algorithms

리님

Graph [Weh13]

A graph is a set of vertices and a set of lines between pairs of vertices.



- Actor vertex, node, point
- Relation line, edge, arc, link, tie
 - Edge = undirected line, {c, d} c and d are end vertices
 - Arc = directed line, (a, d)
 a is the initial vertex, (source, start)
 d is the terminal vertex, (target, end)
 - Parallel (multiple) arcs/edges are only allowed in multigraphs with more than one relation (set of lines).
 - Loop (self-choice)

We focus on simple graphs!

A **simple** undirected graph has no loops and no parallel edges. A simple directed graph has no parallel arcs.

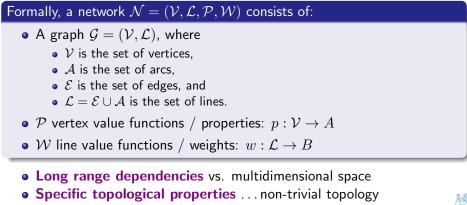
Radek Mařík (radek.marik@fel.cvut.cz)

Introduction to Complex Networks

Network [EK10, New10, Weh13, Erc15]

Network

A **network** consists of a graph and additional information on the vertices or the lines of the graph.



Large/Huge volumes of sparse data records →

Radek Mařík (radek.marik@fel.cvut.cz)

Introduction to Complex Networks

Outline

Complex Networks

- Practical Examples
- Software Tools
- Network Volume
 - Netflow Comprehension
- Network Visualization
 - Data on the Ancient Egypt
 - Mainframe Assembly Comprehension
 - Enterprise people

2 Complex Network Introduction

- Graph Terminology
- Graph Algorithms



315

Asymptotic Notation [CLRS09, Erc15]

Let
$$c, c_1, c_2 \in \mathbb{R}^{>0}$$
, $n_0, n \in \mathbb{N}$, $f, g \in \mathbb{N} \to \mathbb{R}^+$

Asymptotic upper bound (CZ horní asymptotický odhad)

 $f(n) \in O(g(n))$, if $(\exists c > 0)(\exists n_0)(\forall n > n_0) : |f(n)| \le |c \cdot g(n)|$

Asymptotic lower bound (CZ dolní asymptotický odhad)

 $f(n) \in \Omega(g(n))$, if $(\exists c > 0)(\exists n_0)(\forall n > n_0) : |c \cdot g(n)| \le |f(n)|$

Asymptotic tight bound (CZ optimální asymptotický odhad)

 $\begin{aligned} &f(n) \in \Theta(g(n)), \text{ if } \Theta(g(n)) \stackrel{\text{def}}{=} O(g(n)) \cap \Omega(g(n)) \\ &(\exists c_1, c_2 > 0) (\exists n_0) (\forall n > n_0) : |c_1 \cdot g(n)| < |f(n)| < |c_2 \cdot g(n)| \end{aligned}$



NP-Completeness [CLRS09, Erc15]

P and NP

• P - Polynomial. Problems that can be solved in polynomial time.

- **NP Nondeterministic Polynomial**. A problem is in NP if you can in polynomial time by a *certifier* test whether a solution is correct without worrying about how hard it might be to find the solution.
 - Nondeterministic is a fancy way of talking about guessing a solution.
- $P \subseteq NP$ (??? P = NP ???)

NP-complete and NP-hard

- NPH NP-hard. An NPH problem is a problem which is as hard as any problem in NP
 - An NPH problem does not need to have a certificate.
- NPC NP-complete. A problem is NPC if it is NP and is as hard as any problem in NP
 - A problem A is NPC if it is both NPH and in NP, NPC = NP \cap NPH.

Complexity Classes Other Than NP [CLRS09, Erc15]

Complexity classes harder than NP

- PSPACE. Problems that can be solved using a reasonable amount of memory
 - defined formally as a polynomial in the input size
 - without regard to how much time the solution takes.
- **EXPTIME**. Problems that can be solved in exponential time.
- Undecidable. For some problems, we can prove that there is no algorithm that always solves them, no matter how much time or space is allowed.



Tree Search [BM08]

- A systematic procedure, or algorithm, that generates a sequence of rooted trees in G, starting with the trivial tree consisting of a single root vertex r, and terminating either with a spanning tree of the graph or with a nonspanning tree whose associated edge cut is empty, is called tree-search and the resulting tree is referred to as a search tree [BM08].
- **Depth-first search** is a tree-search in which the vertex added to the tree T at each stage is one which is a neighbor of as recent an addition to T as possible.
- The resulting spanning tree is called a **depth-first search tree** or **DFS-tree**.

DFS-tree Search Edge Classification [BM08]

- There are two times associated with each vertex $v \in G$ during the construction of its DFS-tree T:
 - the discovery time $\tau_d(v)$ when v is incorporated into T and
 - the finish time $\tau_f(v)$ when all the neighbors of v are found to be already in T.
- In particular, $\tau_d(r) = 1$, $\tau_f(v) = \tau_d(v) + 1$ for every leaf v of T, and $\tau_f(r) = 2|V|$.
- Based on Proposition 1 and Theorem 1 any edge e = uv in a graph G having a DFS-tree T with $\tau_d(u) < \tau_d(v) < \tau_f(v) < \tau_f(u)$ can be oriented as $\vec{e} = \vec{uv} = (u, v)$ and classified as:
 - tree edge, if $e \in T$, i.e. the vertex u is an ancestor of v in T,
 - back edge, if $e \notin T$.

Tree Search Times - Properties

Proposition 1 (Proposition 6.5 [BM08], p.141)

Let u and v be two vertices of G, with $\tau_d(u) < \tau_d(v)$.

- **1** If u and v are adjacent in G, then $\tau_f(v) < \tau_f(u)$.
- **(**) u is an ancestor of v in T if and only if $\tau_f(v) < \tau_f(u)$.

Theorem 1 (Theorem 6.6 [BM08], p.142)

Let T be a DFS-tree of a graph G. Then every edge of G joins vertices which are related in T.

Lemma 1 (Lemma 22.11 [CLRS09], p.614)

A directed graph G is acyclic if and only if a depth-first search of G yields no back edges.

< 1 k

< ∃⇒

Complex Network Introduction Graph Algorithms

Tree Search Times - Properties

Proposition 2 (Proposition 1.5.6 [Die05], p.16)

Every connected graph contains a normal spanning tree, with any specified vertex as its root.



Breadth-first Search [CLRS09, Erc15]

Algorithm BFS

1.	Input: $G(V, E)$, a source node s	11
	Output: d_v , pred[v], $\forall v \in V$	12
		13
3:	b distance and place of a vertex in BFS	14
4:	Q a gueue	
	for all $u \in V \setminus \{s\}$ do	15
		16
6:	$d_u \leftarrow \infty$	-
7:	$pred[u] \leftarrow \bot \triangleright \text{ undetermined value}$	17
	end for	18
0:	end for	19
9:	$d_s \leftarrow 0$	
10·	$pred[s] \leftarrow s$	20
±0.		21

BFS ... the main loop

11:	$Q \leftarrow s$
12:	while $Q \neq \emptyset$ do
13:	$u \leftarrow deque(Q)$
14:	for all $(u,v) \in E$ do
15:	if $d_v = \infty$ then
16:	$d_v \leftarrow d_u + 1$
17:	$pred[v] \leftarrow u$
18:	enqueu(Q,v)
19:	end if
20:	end for
21:	end while

Theorem 2 (Theorem 3.1 [Erc15], p.35)

The time complexity of BFS algorithm is $\Theta(N+M)$ for a graph of order N and size M.

315

Depth-first Search [CLRS09, Erc15]

Algorithm DFS_Forest

- 1: Input: G(V, E), directed or undirected
- 2: **Output:** pred[v], firstVis[v], secVis[v], $\forall v \in V$
- 3: int time $\leftarrow 0$; visited $[1:n] \leftarrow 0$
- 4: for all $u \in V$ do
- 5: $visited[u] \leftarrow false$
- 6: $\operatorname{pred}[u] \leftarrow \bot \quad \triangleright \text{ undetermined value}$
- 7: end for
- 8: for all $u \in V$ do
- 9: **if** $\neg visited[u]$ **then**
- 10: DFS(u)
- 11: end if
- 12: end for

DFS procedure

13: procedure DFS(u) 14: $visited[u] \leftarrow true$ 15: $time \leftarrow time + 1$ $firstVis[u] \leftarrow time$ 16: for all $(u, v) \in E$ do 17: if $\neg visited[v]$ then 18: $pred[v] \leftarrow u$ 19: 20: DFS(v)21: end if 22: end for 23: $time \leftarrow time + 1$ 24: $secVis[u] \leftarrow time$ 25: end procedure

Asymptotic complexity of the DFS algorithm

The time complexity is $\Theta(N+M)$ for a graph of order N and size M.

Radek Mařík (radek.marik@fel.cvut.cz)

Dijkstra's Single Source Shortest Paths [CLRS09, Erc15]

	CCCD the main lean
Algorithm Dijkstra_SSSP	SSSP the main loop
1: Input: $G(V, E)$, directed or undirected, 2: Input: positive weights l_e on edges, 3: Input: a source node s 4: Output: d_v , pred $[v]$, $\forall v \in V$ 5: for all $u \in V \setminus \{s\}$ do 6: $d_u \leftarrow \infty$ 7: pred $[u] \leftarrow \bot \Rightarrow$ undetermined value 8: end for 9: $d_s \leftarrow 0$ 10: pred $[s] \leftarrow s$	$\label{eq:states} \begin{array}{ c c c }\hline 11: \ S \leftarrow [V] & \triangleright \text{ insert all vertices} \\ \hline 12: \ \text{while} \ S \neq \emptyset \ \text{do} \\ \hline 13: \ \ u \leftarrow min(S) \\ \hline 14: \ \ S \leftarrow S \setminus \{u\} \\ \hline 15: \ \ \text{for all} \ (u,v) \in E \ \text{do} \\ \hline 16: \ \ \ \text{if} \ \ d_v > d_u + l(u,v) \ \text{then} \\ \hline 17: \ \ \ \ d_v \leftarrow d_u + l(u,v) \\ \hline 18: \ \ \ \text{pred}[v] \leftarrow u \\ \hline 19: \ \ \ \text{end} \ \text{if} \\ \hline 20: \ \ \text{end} \ \text{for} \\ \hline 21: \ \ \text{end} \ \text{while} \end{array}$

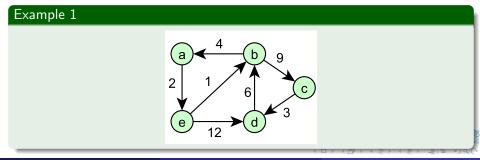
Theorem 3 (Theorem 5.1 [Erc15], p.84)

The time complexity of the Dijkstra's_SSSP is $O(N^2)$ for a graph of order N.

Radek Mařík (radek.marik@fel.cvut.cz)

Floyd-Warshall All Pairs Shortest Paths [CLRS09, Erc15]

- The approach
 - Dynamic programming approach
 - ${\ensuremath{\, \circ }}$ Comparing all possible paths between each pair of nodes in G
 - Improving the shortest path between them at each step until the result is optimal.
- Distance matrix D[N, N] between nodes u and v
- $\bullet~\mbox{Matrix}~P[N,N]$ with the first node on the current shortest path from u to v

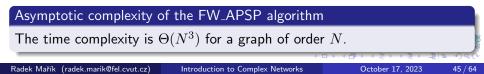


Radek Mařík (radek.marik@fel.cvut.cz)

Introduction to Complex Networks

FW APSP Algorithm [CLRS09, Erc15]

gorithm FW_APSP	APSP the main loop		
Input: $G(V, E)$,	14:	$S \leftarrow \emptyset$	
2: Input: weights w_e on edges,		while $S \neq V$ do	
no negative-weight cycles	16:	pick w from $V \setminus S$ \triangleright Select a pivot	
Output: $D[N, N]$, $P[N, N]$	17:	for all $u \in V$ do	
for all $\{u, v\} \in V$ do	18:	for all $v \in V$ do	
if $u = v$ then	19:	if $D[u, w] + D[w, v] < D[u, v]$ then	
$D[u,v] \leftarrow 0; P[u,v] \leftarrow \bot$	20:	$D[u,v] \leftarrow D[u,w] + D[w,v]$	
else if $(u,v) \in E$ then	21:	$P[u, v] \leftarrow P[u, w]$	
$D[u, v] \leftarrow w_{uv}; P[u, v] \leftarrow v$	22:	end if	
else	23:	end for	
$D[u,v] \leftarrow \infty; P[u,v] \leftarrow \bot$	24:	end for	
end if	25:	$S \leftarrow S \cup \{w\}$	
13: end for		end while	
	no negative-weight cycles Output: $D[N, N]$, $P[N, N]$ for all $\{u, v\} \in V$ do if $u = v$ then $D[u, v] \leftarrow 0$; $P[u, v] \leftarrow \bot$ else if $(u, v) \in E$ then $D[u, v] \leftarrow w_{uv}$; $P[u, v] \leftarrow v$ else $D[u, v] \leftarrow \infty$; $P[u, v] \leftarrow \bot$ end if	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	

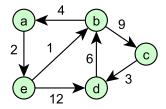


Complex Network Introduction

Graph Algorithms

FW APSP Algorithm Example [Erc15]

$$D = \begin{bmatrix} 0 & \infty & \infty & \infty & 2\\ 4 & 0 & 9 & \infty & \infty\\ \infty & \infty & 0 & 3 & \infty\\ \infty & 6 & \infty & 0 & \infty\\ \infty & 1 & \infty & 12 & 0 \end{bmatrix}$$
$$\rightarrow \begin{bmatrix} 0 & 3 & \infty & 14 & 2\\ 4 & 0 & 9 & 12 & 6\\ \infty & 9 & 0 & 3 & \infty\\ 10 & 6 & 15 & 0 & \infty\\ 5 & 1 & 10 & 12 & 0 \end{bmatrix}$$
$$\rightarrow \begin{bmatrix} 0 & 3 & 12 & 14 & 2\\ 4 & 0 & 9 & 12 & 6\\ 13 & 9 & 0 & 3 & 10\\ 10 & 6 & 15 & 0 & 12\\ 5 & 1 & 10 & 12 & 0 \end{bmatrix}$$



N

46 / 64

315

Summary

- An introduction to complex networks
- Several practical application domains shown
- Software tools overview
- Demonstration of two issues
 - Network data volume
 - Network visualization
- Graph Terminology Reminder
- Graph Path Algorithms Reminder

Competencies

- Name several examples of complex networks application domains?
- What are the two difficult issues linked with processing of complex networks?
- What is the range of complex network volume?
- Name several drawing layouts used for complex network visualizations?
- Define a complex network and its basic features.
- Define asymptotic bounds used for assessment of algorithm complexity.
- Describe DFS-tree search edge classification.
- Describe depth-first search algorithm.
- Describe breath-first search algorithm.
- Describe the Dijkstra's single source shortest paths.
- Describe the Floyd-Warshall all pairs shortest paths.

Radek Mařík (radek.marik@fel.cvut.cz)

Introduction to Complex Networks



Appendix



Radek Mařík (radek.marik@fel.cvut.cz)

< P



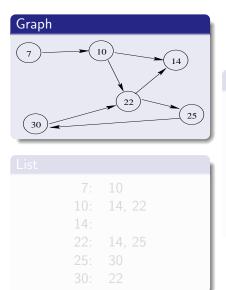


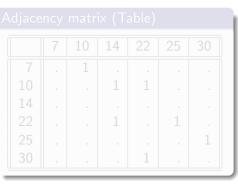


50 / 64

문 문

Graph Representation [Bei95]





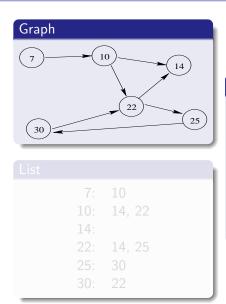
Radek Mařík (radek.marik@fel.cvut.cz)

M

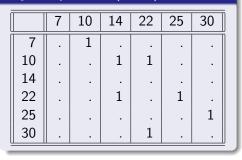
51/64

문(권)

Graph Representation [Bei95]



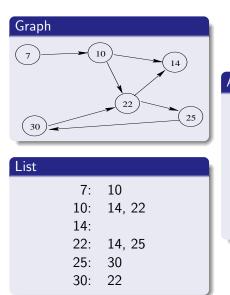
Adjacency matrix (Table)



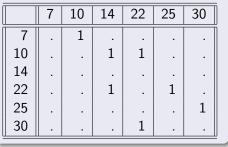


三日 のへの

Graph Representation [Bei95]



Adjacency matrix (Table)





51/64

문 문

Graph (Formal Definitions) [Die05, BM08, Wil98]

- A graph is a pair G = (V, E) of sets such that E ⊆ [V]², V ∩ E = Ø, together with an incidence function ψ_G that associates with each edge of G an unordered pair of not necessarily distinct vertices of G.
- The number of vertices of a graph G is its order N = v(G) = |V| = |G|.
- A graph with vertex set V is said to be a graph on V.
- The vertex set of a graph G is referred to as V(G), its edge set as E(G), independently of any actual names of these two sets.
- We also write $v \in G$ instead of $v \in V(G)$, similarly $e \in G$.
- The number of edges of a graph G is its size denoted by M = e(G) = |E| = ||G||.

Graph Edges [Die05, BM08, Wil98]

- Let e be an edge and u and v are vertices such that $\psi_G(e) = \{u, v\}$.
- A vertex v is **incident** with an edge e if $v \in e$; then e is an edge **at** v.
- The set of all the edges in E at a vertex v is denoted by E(v).
- The two vertices v_1 and v_2 incident with an edge $e = \{v_1, v_2\}$ are its endvertices or ends, and an edge joins its ends.
- An edge $\{u, v\}$ might be written as uv (or vu).
- Two vertices $u, v \in G$ are adjacent, or neighbors, if $uv \in G$.
- Two edges $e \neq f$ are **adjacent** if they have an end in common.

Graph Neighborhood [Die05, BM08, Wil98]

- Let G = (V, E) be a (non-empty) graph.
- The set of **neighbors** of a vertex v in G is denoted by $\mathcal{N}_G(v)$, or briefly by $\mathcal{N}(v)$.
- The neighbors of U for $U \subseteq V$, denoted by $\mathcal{N}(U)$, is the set of the neighbors $V \setminus U$ of vertices in U.
- The degree (or valency) $d_G(v) = d(v)$ of a vertex v is the number |E(v)| of edges at v.
- Let $r \ge 2$ be an integer.
- A graph G = (V, E) is called *r*-partite if V admits a partition into r classes such that every edge has its ends in different classes: vertices in the same partition class are not adjacent.
- If r = 2 then such a graph is denoted as **bipartite**.

•
$$V = V_1 \cup V_2$$
, $V_1 \cap V_2 = \emptyset$



Graph Path [Die05, BM08, Wil98]

- A path is a non-empty graph P = (V, E) of the form $V = \{v_0, v_1, \ldots v_k\}, E = \{v_0v_1, v_1v_2, \ldots v_{k-1}v_k\},$ where the v_i are all distinct.
- The vertices v_0 and v_k are **linked** by P and are called its **ends**, the vertices $v_1, \ldots v_{k-1}$ are the **inner** vertices of P.
- A path P can often be identified by its natural sequence of its vertices, i.e. P = v₀v₁...v_k and called a path from v₀ to v_k (or between v₀ and v_k).
- If $P = v_0 \dots v_{k-1}$ is a path and $k \ge 3$, then the graph $C := P + v_{k-1}v_0$ is called a cycle.

Graph Walk [Die05, BM08, Wil98]

- A walk in a graph G is a sequence $W := v_0 e_1 v_1 \dots v_{\ell-1} e_{\ell} v_{\ell}$, whose terms are alternately vertices and edges of G, such that v_{i-1} and v_i are the ends of e_i , $1 \le i \le \ell$.
- If v₀ = x and v_ℓ = y, we say that W connects x to y and refer to W as an xy-walk.
- The vertices x and y are called the ends of the walk, x being its initial vertex and y its terminal vertex, the vertices $v_1, \ldots, v_{\ell-1}$ are its internal vertices.
- The integer ℓ (the number of edge terms) is the **length** of W.
- An *x*-walk is a walk with initial vertex *x*.
- If there is an xy-walk in a graph G, then is also an xy-path.
- The length of a shortest such xy-path is called the **distance** between x and y and denoted $d_G(x, y)$.
- The greatest distance between any two vertices in G is called the diameter of G, denoted by $diam(G) = \max_{u,v} d_G(u, v)$.

Graph Component [Die05, BM08, Wil98]

- A non-empty graph G is called **connected** if any two of its vertices are linked by a path in G, otherwise the graph is **disconnected**.
- If U ⊆ V(G) and G[U] is connected, we call U itself connected (in G).
- A maximal connected sugraph of G is called a **component** of G.

- An acyclic graph is a graph that does not contain any cycle.
- An acyclic graph is also called a forest.
- A connected forest is called a tree.
- The vertices of degree 1 in a tree are its leaves.
- One vertex of a tree can be selected as special; such a vertex is then called the **root** of this tree.
- A tree T with a fixed root r is a rooted tree.
- A spanning tree of a graph G is a minimal connected spanning subgraph $T \subset G$

Tree Properties I

Theorem 4 (Theorem 1.5.1 [Die05], p.14)

The following assertions are equivalent for a graph T:

- \bigcirc T is a tree;
- Any two vertices of T are linked by a unique path in T;
- (1) T is minimally connected, i.e. T is connected but T e is disconnected for every edge $e \in T$;
- **(a)** T is maximally acyclic, i.e. T contains no cycle but T + uv does, for any two non-adjacent vertices $u, v \in T$.

Corollary 1 (Corollary 1.5.3 [Die05], p.14)

A connected graph with N vertices is a tree if and only if it has N-1 edges.

글 🖌 글 🖻

A (1) > A (2) > A

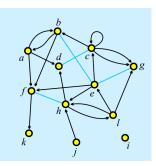
- A directed graph (or digraph) is a pair (V, E) of disjoint sets (of vertices and arcs) together with two maps init : E → V and ter : E → V assigning to every arc e an initial vertex init(e) and a terminal vertex ter(e).
- In some references, vertices of directed graphs are called nodes.
- The arc e is said to be **directed from** init(e) to ter(e).
- Both maps init(e) and ter(e) are often combined into an incidence function ψ_D that associates with each arc of D an ordered pair of vertices of D, ψ_D(e) = (u, v).

Digraph Degree [Die05, BM08, Wil98]

- The degree of a vertex v in a digraph D is simply the degree of v in the underlying graph G(D) of D.
- The indegree $d_D^-(v)$ of a vertex $v \in D$ is the number of arcs with head v,
- the **outdegree** $d_D^+(v)$ of a vertex $v \in D$ is the number of arcs with tail v.
- A vertex of indegree zero is called a **source**, one of outdegree zero a **sink**.

Graph Theory Graph Terminology

Vertex Degree [Weh13]



- **Degree** of vertex i, $deg(i) = d_i = k_i = \sum_{j=1}^N A_{ij}$ = the number of lines with i as end-vertex, (end-vertex is both initial and terminal)
- Indegree of vertex i, indeg(i), $deg^+(i)$ = $k_i^{in} = \sum_{j=1}^N A_{ij}$ the number of lines with v as terminal vertex
- **Outdegree** of vertex j, outdeg(j), $deg^{-}(j) = k_j^{out} = \sum_{i=1}^{N} A_{ij}$ the number of lines with j as initial vertex.

Example 2

$$N = 12$$
, $M = 23$, $deg^+(e) = 3$, $deg^-(e) = 5$, $deg(e) = 6$

$$\sum_{v \in \mathcal{V}} deg^+(v) = \sum_{v \in \mathcal{V}} deg^-(v) = |\mathcal{A}| + 2|\mathcal{E}|$$

References I

- [Bei95] Boris Beizer. Black-Box Testing, Techniques for Functional Testing of Software and Systems. John Wiley & Sons, Inc., New York, 1995.
- [Blo14] David A. Bloom. Matula thoughts december 5, 2014, 2014.
- [BM08] J.A. Bondy and U.S.R. Murty. Graph Theory. Springer, 2008.
- [CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Third Edition. The MIT Press, 3rd edition, 2009.
- [Die05] Reinhard Diestel. Graph Theory. Springer, 2005.
- [EK10] David Easley and Jon Kleinberg. Networks, Crowds, and Markets. Reasoning About a Highly Connected World. Cambridge University Press, July 2010.
- [Erc15] Kayhan Erciyes. Complex Networks, An Algorithmic Perspective. CRC Press, 2015.
- [FMS] FMS. Social network analysis (SNA) diagram, al qaeda terrorist network, accessed 28.1.2014.
- [Fre14] Fremantle. Celebrating a soy-free easter with amedei chocolate, accessed 28.1.2014. http://infonolan.hubpages.com/hub/Celebrating-a-Soy-Free-Easter-with-Amedei-Chocolate, 2014.
- [HLDS13] Dan Halgin, Joe Labianca, Rich DeJordy, and Maxim Sytch. Introduction to social network analysis. http://www.danhalgin.com/slides, August 2013.
- [Mar17] Radek Marik. Efficient Genealogical Graph Layout, pages 567–578. Springer International Publishing, Cham, 2017.
- [New10] M. Newman. Networks: an introduction. Oxford University Press, Inc., 2010.
- [Opt17] The OPTE project: The internet 2015; accessed 2017.09.17, 2017.
- [PASP09] Jose M Peregrin-Alvarez, Chris Sanford, and John Parkinson. The conservation and evolutionary modularity of metabolism. Genome Biology, 10(6), June 2009.

(日) (同) (三) (三)

315

References II

- [RP13] James Risen and Laura Poitras. N.S.A. gathers data on social connections of U.S. citizens, September 2013.
- [Weh13] Stefan Wehrli. Social network analysis, lecture notes, December 2013.
- [Wil98] Robin J. Wilson. Introduction to Graph Theory. Longman, fourth edition, 1998.



64 / 64

315

< A[™]