Protection of communication-networks using p-cycles

Tommy Thomadsen and Thomas Stidsen

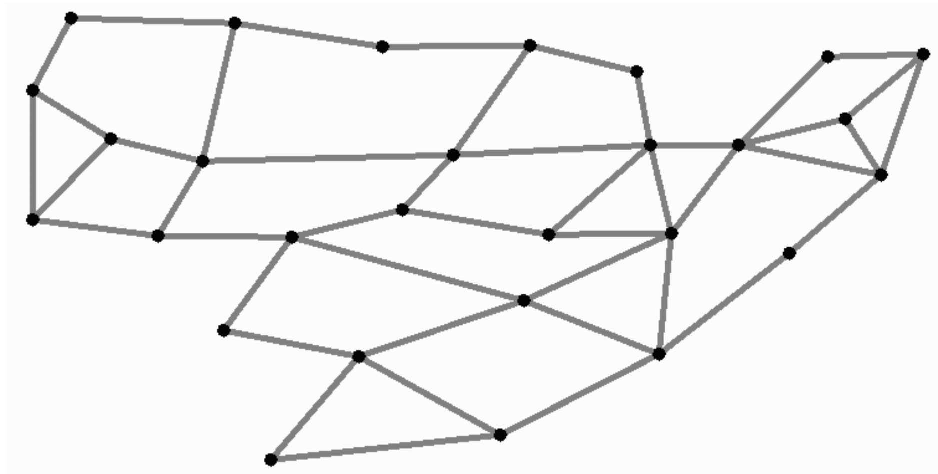
tt@imm.dtu.dk

Informatics and Mathematical Modeling
Technical University of Denmark

Outline

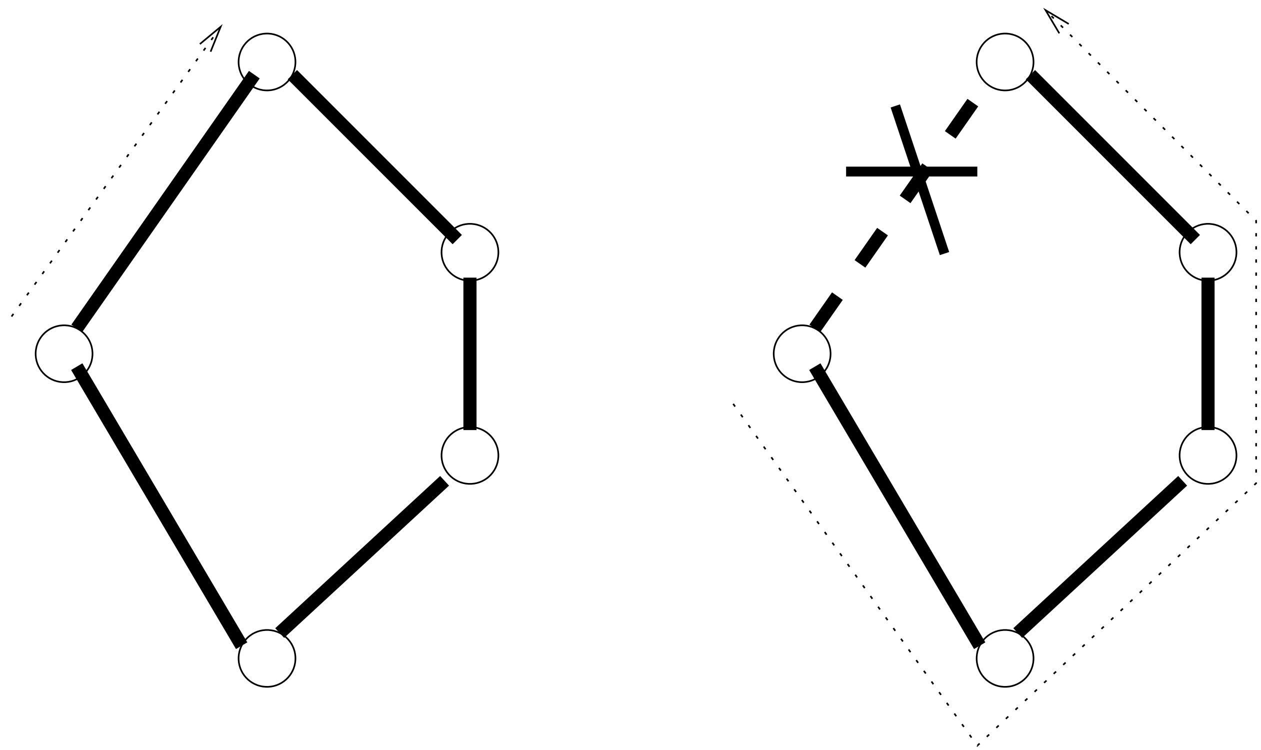
- Communication Networks and Protection methods
- Fast *and* capacity efficient: p-cycles
- Mathematical Model for rings and p-cycles
- Solution methods
  - Column Generation
  - Generation of Columns: QSTSP
  - Joint routing and protection
- Computational Study
- Conclusions
- Future work and projects

Communication Networks



- Nodes
- Links
- Demands
- Circuit Oriented
- Working capacity
- Protection (Spare) capacity reserved for use in case of failures
- E.g. shortest-path routing for routing demands

Ring Protection



- Additional Protection Capacity is equivalent to max working capacity in entire ring.

Protection of Communication Networks

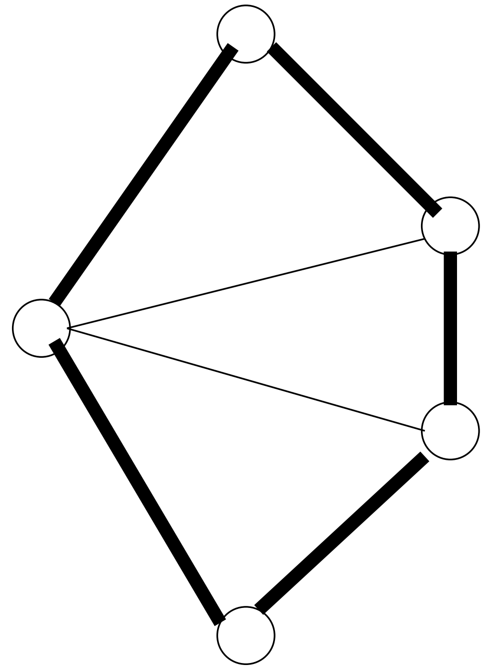
Protection methods are divided into:

- Slow but capacity efficient: Meshed protection schemes
  - Routing
  - Path protection
  - Span protection
  - Complete re-routing is most capacity efficient
- Fast but capacity *in*efficient: Rings, 1+1
  - Below 50ms
  - Pre-configured
  - Local computation only (in end-nodes of the failed link)
  - No routing

Protection of Communication Networks II

- Rings are widely used, since they are cheaper than meshed protection:
  - Node equipment is cheaper
  - No management system is required
  - They are “self-healing”
- Why should capacity efficiency prohibit fast protection?
- Fast *and* capacity efficient: protection/preconfigured-cycles
- The equipment cost for p-cycles should be comparable with rings (but that remains to be seen).
- The cost of using p-cycles: Two link breaks have a higher probability of influencing each other, i.e. being part of the same p-cycle.

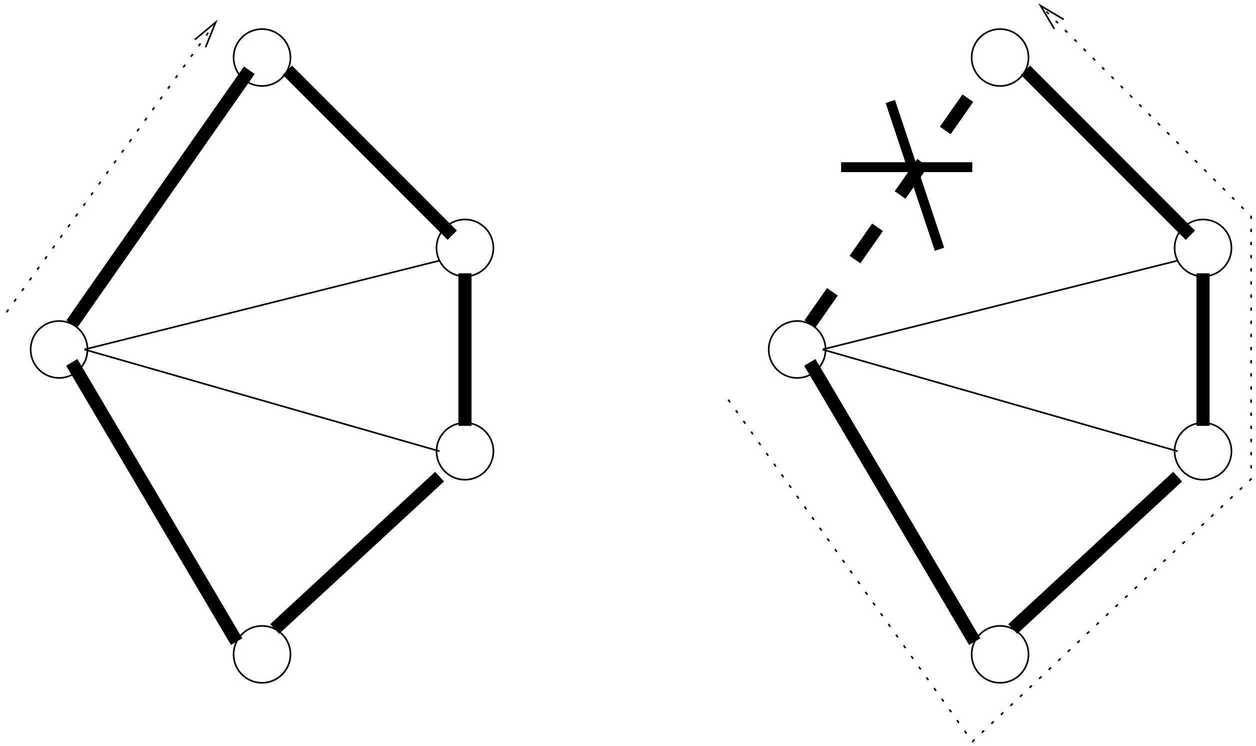
p-cycle protection



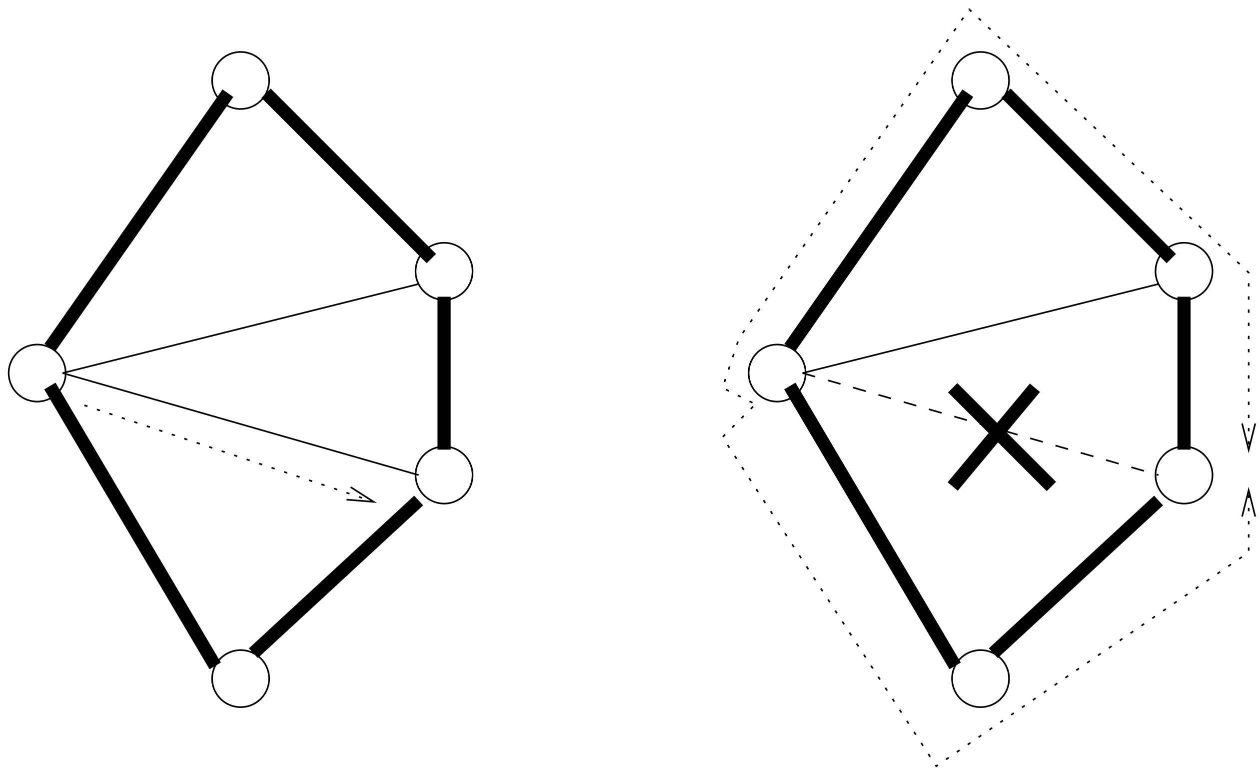
A p-cycle consists of:

- A cycle
- All chords of the cycle
- Additional protection capacity allocated *along* the cycle
- The p-cycle protects:
  - On-cycle links, like ring protection.
  - Straddling links, i.e. the chords

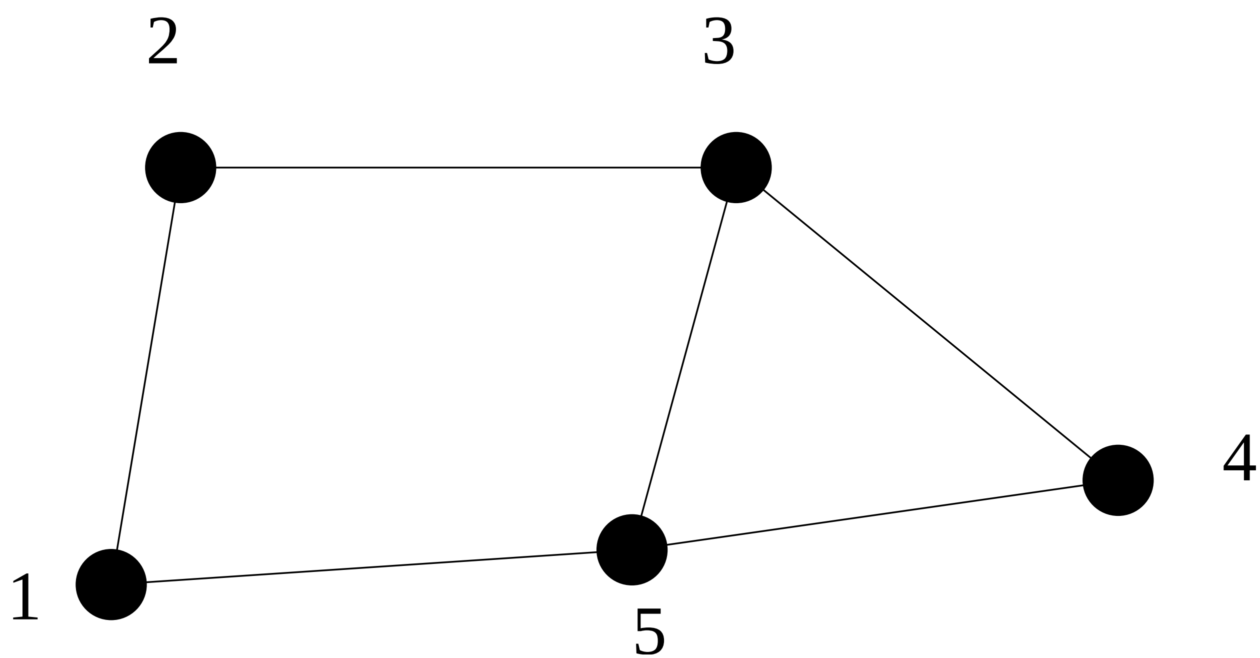
p-cycle: On-cycle



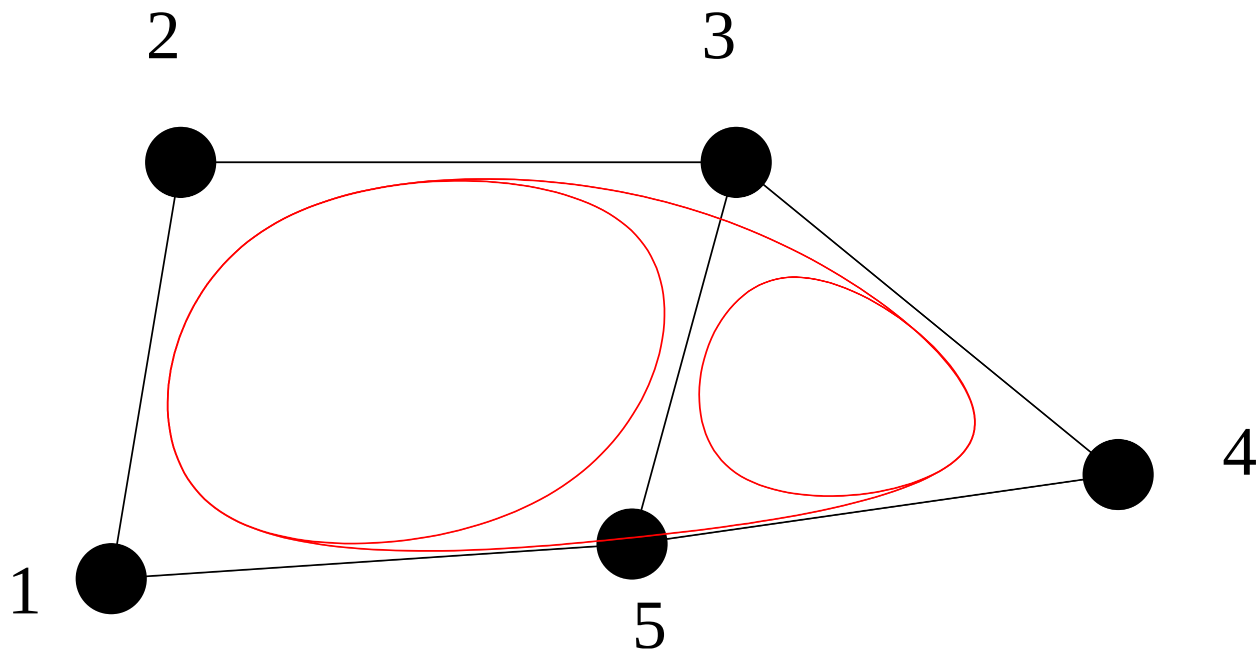
p-cycle: Straddling link



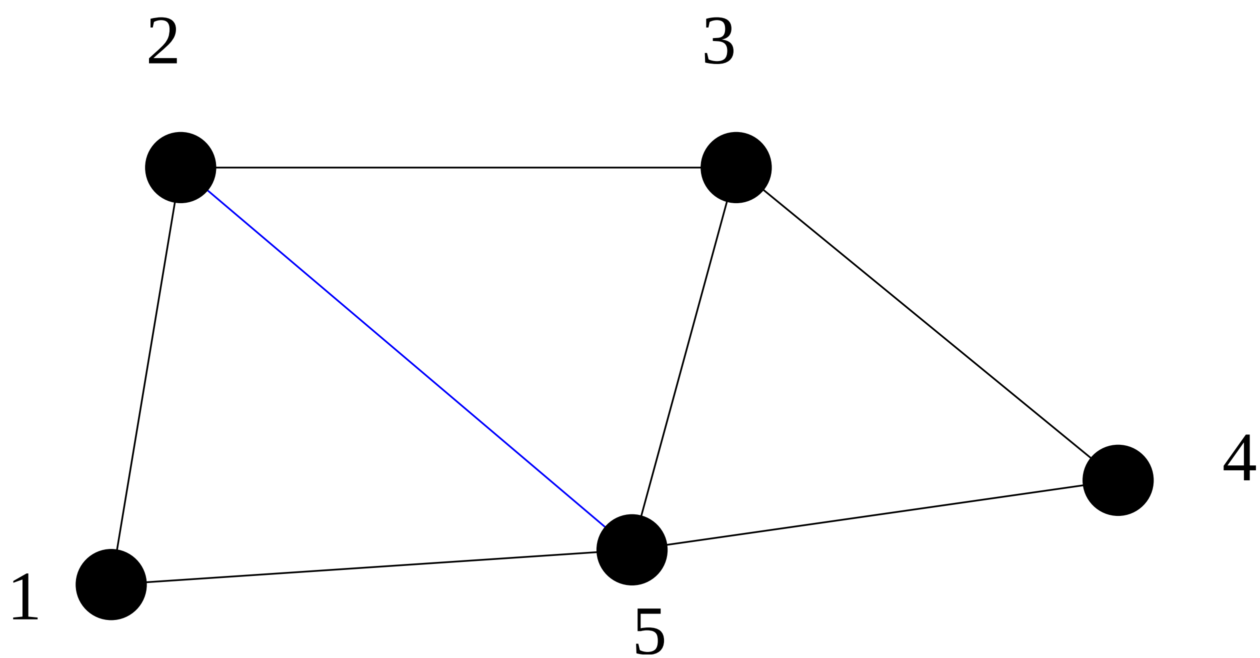
An example: All Rings/p-cycles



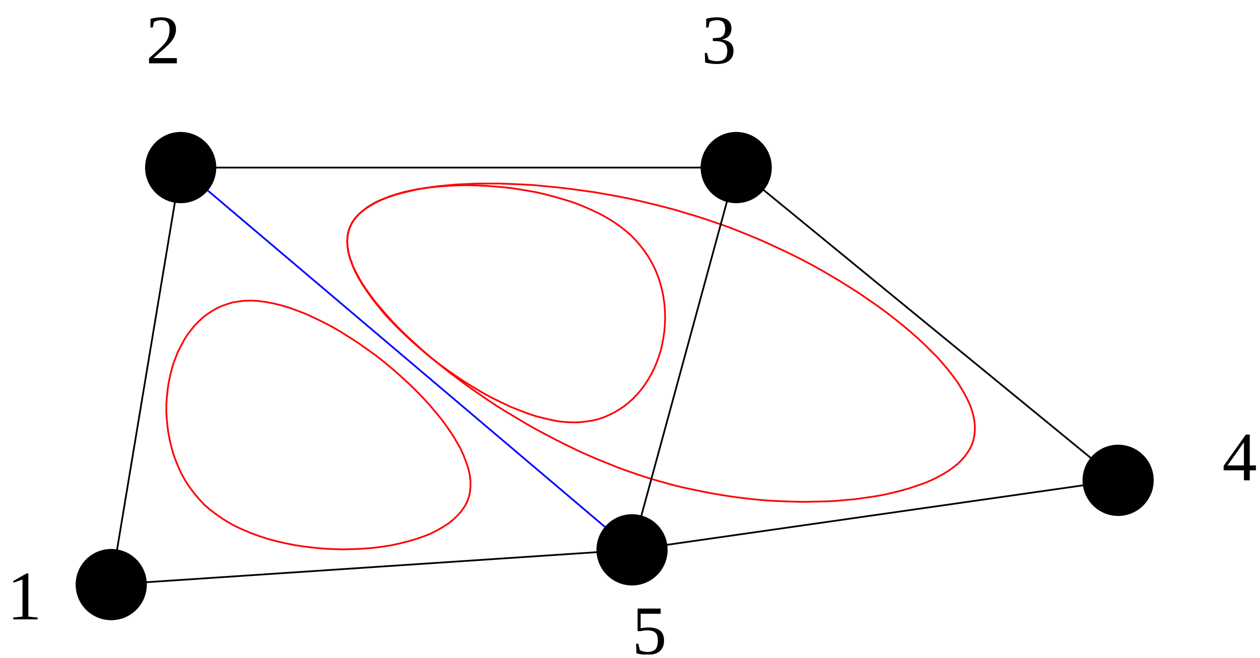
An example: All Rings/p-cycles



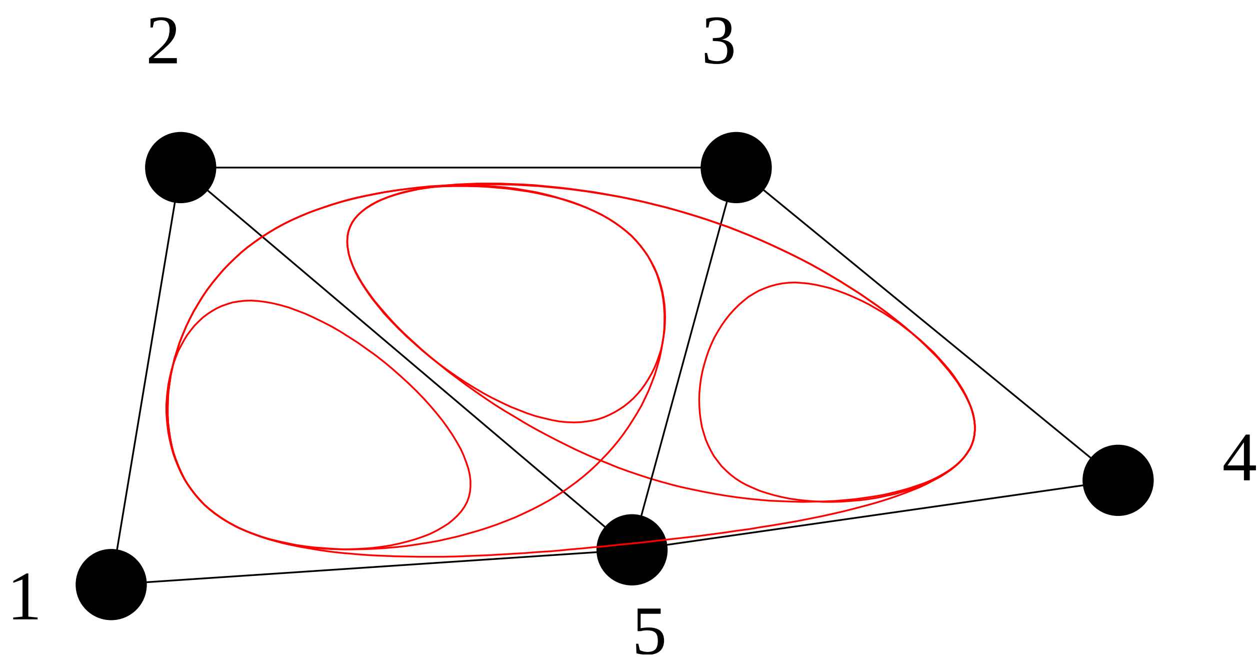
An example: All Rings/p-cycles



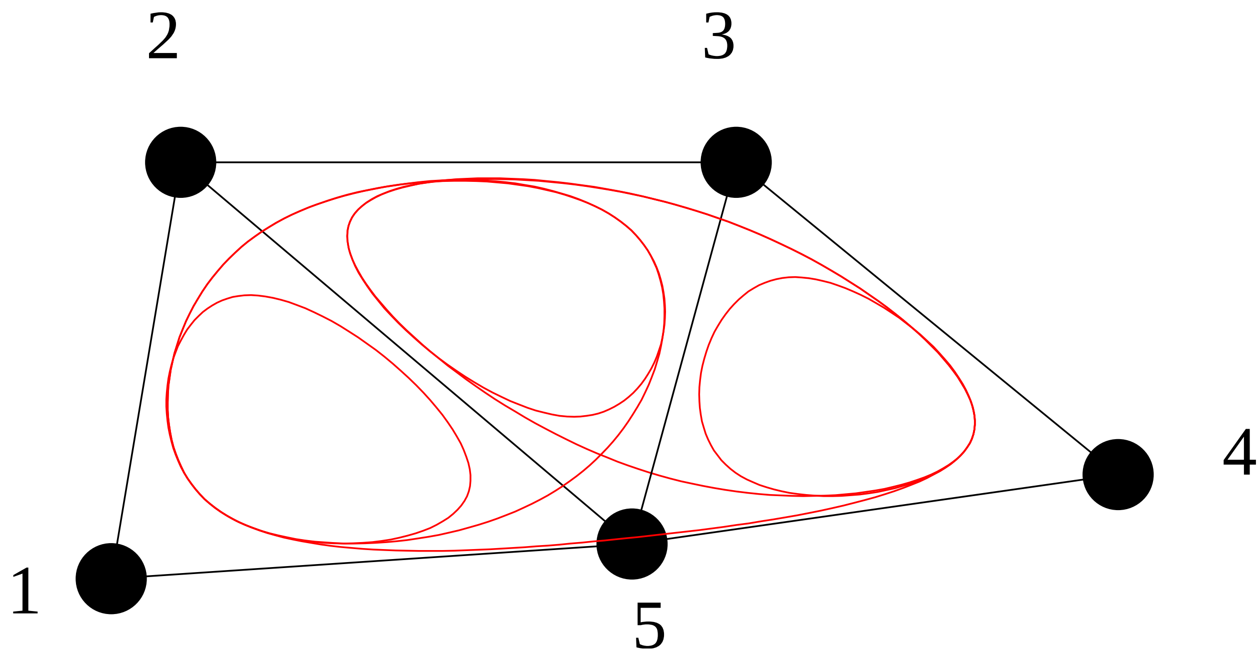
An example: All Rings/p-cycles



An example: All Rings/p-cycles

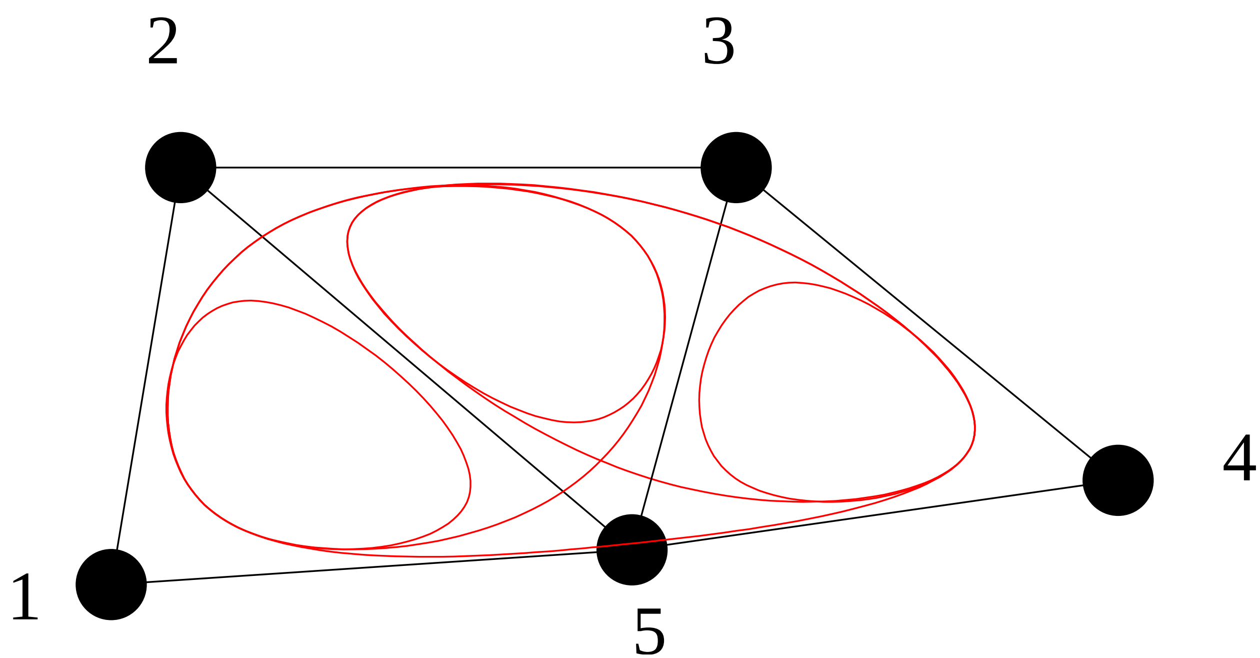


An example: All Rings/p-cycles



- A cover of the graph consisting of Rings/p-cycles

An example: All Rings/p-cycles



- A cover of the graph consisting of Rings/p-cycles
- Rings vs. p-cycles - how much can it matter?

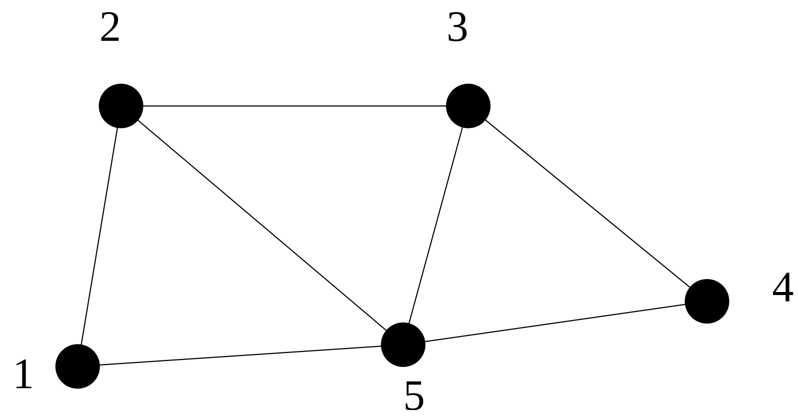
Efficiency

- Efficiency: $\dfrac{\text{Working capacity that is protected}}{\text{Protection capacity}}$
- Higher efficiencies are better.
- Independent of actual traffic in the network and of other rings/p-cycles in the network.
- Rings: $n/n = 100\%$
- p-cycles are always better than rings.
- p-cycle efficiency depends on the number of straddling links.

Efficiency of p-cycles

- n is the number of on-cycle links
- m is number of straddling links
- If all links exists:
  - $m = n(n-1)/2 - n = \frac{n^2-3n}{2}$
  - p-cycle efficiency: $\frac{n+2m}{n} = \frac{n+n^2-3n}{n} = n-2$
  - p-cycles seem more efficient the more links they contain!
- The actual efficiency depends on the traffic and on other rings/p-cycles in the network.
- Measure for a protected *network*:
  - Additional protection capacity in relation to the total working capacity.
  - In essence *the inverse of the efficiency*.

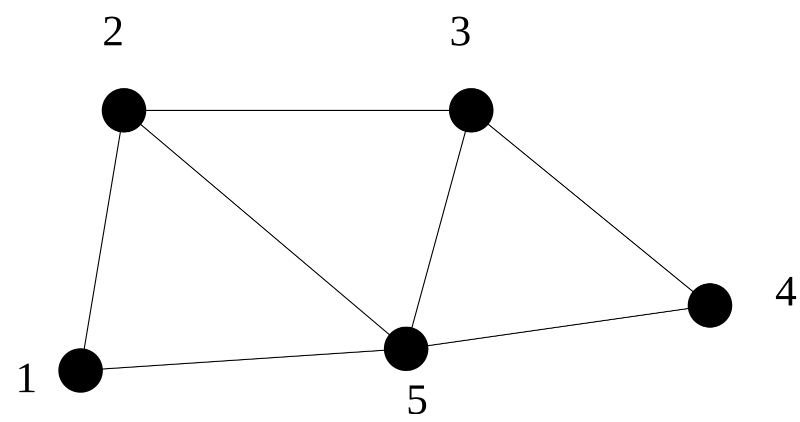
An example: Efficiencies



- Ring efficiency
  - 3 nodes: $\frac{3}{3} = 100\%$
  - 4 nodes: $\frac{4}{4} = 100\%$
  - 5 nodes: $\frac{5}{5} = 100\%$
- p-cycle efficiency:
  - 3 nodes: $\frac{3}{3} = 100\%$
  - 4 nodes, 1 straddler (max 2): $\frac{4+2}{4} = 150\%$
  - 5 nodes, 2 straddlers (max 5): $\frac{5+4}{5} = 180\%$
- The actual efficiency depend on the traffic in the network!

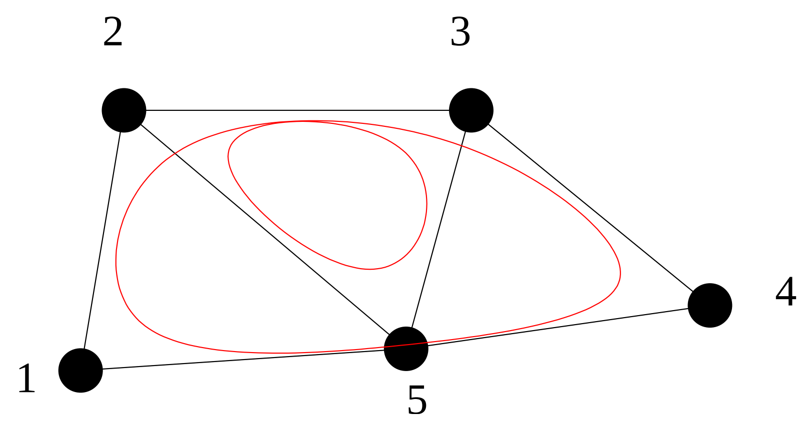
Cost in additional protection capacity

- Assume: Each link contains 1 unit of working capacity (7 units total).



Cost in additional protection capacity

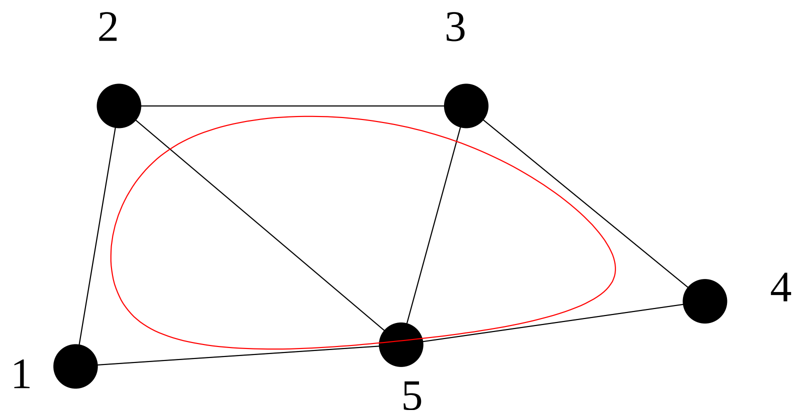
- Assume: Each link contains 1 unit of working capacity (7 units total).



- Ring protected: 2-3-5, 1-2-3-4-5, in total 8 units protection capacity. In total $8/7 = 114\%$

Cost in additional protection capacity

- Assume: Each link contains 1 unit of working capacity (7 units total).



- Ring protected: 2-3-5, 1-2-3-4-5, in total 8 units protection capacity. In total $8/7 = 114\%$
- p-cycle protected: 1-2-3-4-5, in total 5 units protection capacity. In total $5/7 = 71\%$

p-cycles are interesting, because..

- Theoretical: Fast protection, high capacity efficiency.
- Implementation of capacity efficient (meshed) protection schemes have failed mostly due to the lack of a management system. No management system is required for using p-cycles.
- Practical, hardware: They are ring like, pre-configured - how hard can it be to implement!? Extend ring standards.
- The investigation we have conducted can give arguments as to why the technology has to be developed. If the savings are high, then industry should look into it.
- Grover and Stamatelakis (1998) claimed that p-cycles offer "*ring like speed with mesh-like efficiency*"

p-cycles

- Optimization Questions:
  - How should p-cycles be allocated?
  - How efficient are p-cycles?
- Mathematical model for allocating p-cycles and optimizing efficiency:
  - Set covering-like-model.
  - A column for each ring/p-cycle

Master Problem model, Rings

Constants:

$$a_r^e = \begin{cases} 1 & \text{if } e \in r \\ 0 & \text{otherwise} \end{cases}$$

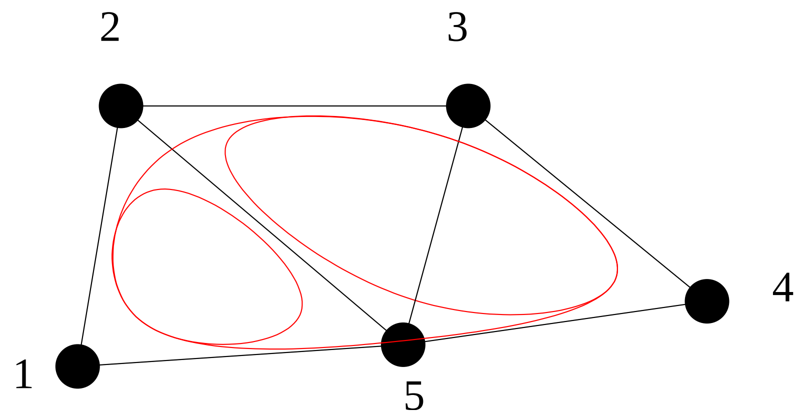
minimize:

$$\sum_{r \in R} c_r \cdot x_r$$

subject to:

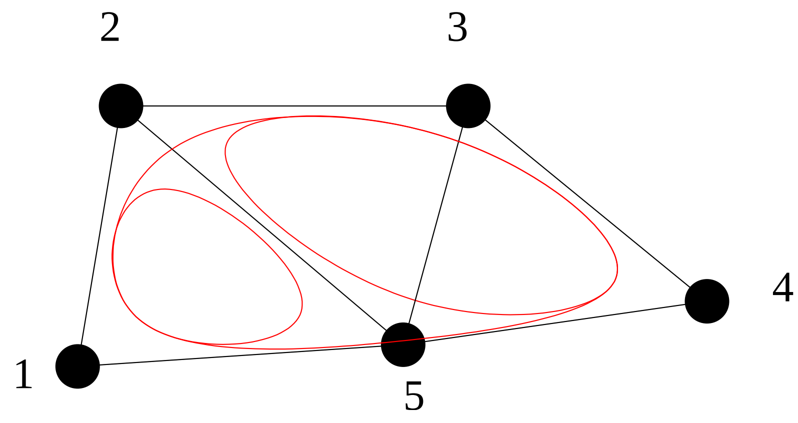
$$\begin{aligned} \sum_{r \in R} a_r^e \cdot x_r &\geq w_e && \forall e \in E \\ x_r &\in \{0, 1, 2, \ldots\} \end{aligned}$$

Example: Rings



MIN	$3x_1$	+	$4x_2$	+	$5x_3$	+	$\dots$		
ST:									
12	$1x_1$				$1x_3$	+	$\dots$	$\geq$	w_{12}
15	$1x_1$				$1x_3$	+	$\dots$	$\geq$	w_{15}
23			$1x_2$	+	$1x_3$	+	$\dots$	$\geq$	w_{23}
25	$1x_1$	+	$1x_2$			+	$\dots$	$\geq$	w_{25}
34			$1x_2$	+	$1x_3$	+	$\dots$	$\geq$	w_{34}
35							$\dots$	$\geq$	w_{35}
45			$1x_2$	+	$1x_3$	+	$\dots$	$\geq$	w_{45}

Example p-cycles



MIN	$3x_1$	+	$4x_2$	+	$5x_3$	+	$\ldots$		
ST:									
12	$1x_1$				$1x_3$	+	$\ldots$	$\geq$	w_{12}
15	$1x_1$				$1x_3$	+	$\ldots$	$\geq$	w_{15}
23			$1x_2$	+	$1x_3$	+	$\ldots$	$\geq$	w_{23}
25	$1x_1$	+	$1x_2$	+	$2x_3$	+	$\ldots$	$\geq$	w_{25}
34			$1x_2$	+	$1x_3$	+	$\ldots$	$\geq$	w_{34}
35			$2x_2$	+	$2x_3$	+	$\ldots$	$\geq$	w_{35}
45			$1x_2$	+	$1x_3$	+	$\ldots$	$\geq$	w_{45}

Master Problem, p-cycles

Constants:

$$a_r^e = \begin{cases} 1 & \text{if } e \in r \\ 2 & \text{if } e \text{ straddles } r \\ 0 & \text{otherwise} \end{cases}$$

minimize:

$$\sum_{r \in R} c_r \cdot x_r$$

subject to:

$$\sum_{r \in R} a_r^e \cdot x_r \geq w_e \qquad \forall e \in E$$

$$x_r \in \{0, 1, 2, \ldots\}$$

Enumeration of all rings/p-cycles

- Depth first tree search from all nodes.
- To avoid generating the same cycle more than once:
  - first node is the node with the lowest index
  - second node has less index than the last node
- Artificial upper bound on the length and/or number of nodes a cycle can contain to be able to generate.
  - Rings: Works well, small cycles are the best
  - p-cycles: Large cycles are the best..
- Papers have been written with the sole purpose of generating a good set of p-cycles.
- Use the information given from the solution of the master problem to generate columns.

The Column Generation Algorithm

Initialize SET OF p-CYCLES
do
  Solve master problem with SET OF p-CYCLES
  Find any improving p-cycle and add it to SET OF p-CYCLES
while IMPROVING p-CYCLE FOUND **do**
Obtain integer solution from SET OF p-CYCLES

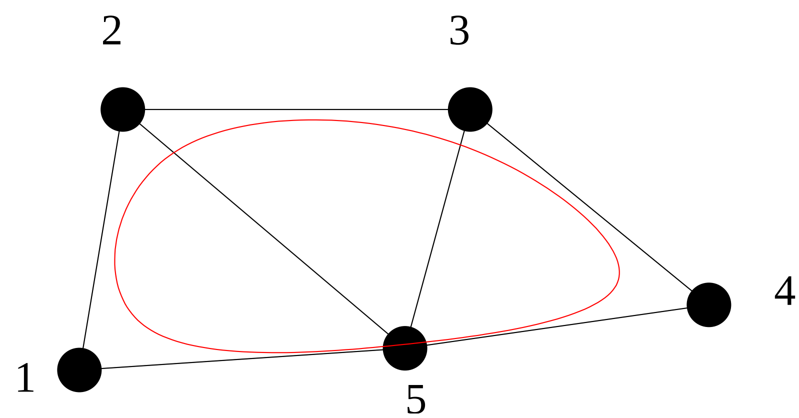
- It is elegant!
- Generating rings is done by finding negative cycles: Finding the best is difficult, but determining whether one exists is easy.
- Generating p-cycles is NP-hard; Checking whether one exists is not easier than finding the best.
- Information regarding cycles are in the columns.

Generating p-cycles: QSTSP

- Travelling Salesman Problem (TSP)
- Selective TSP
- Quadratic Selective TSP (QSTSP)

QSTSP Example

- Cost on edges incurred if edge is on-cycle
- Rewards on edge, obtained if either the edge is on-cycle *or* straddling the cycle.
- Find Best cycle with a positive reward minus cost.
- Example, assume:
  - All costs are 10,
  - All rewards are 8.
- Cycle 1-2-3-4-5
  - A reward of $8 \cdot 7 = 56$
  - A cost of $5 \cdot 10 = 50$.
  - In total 6

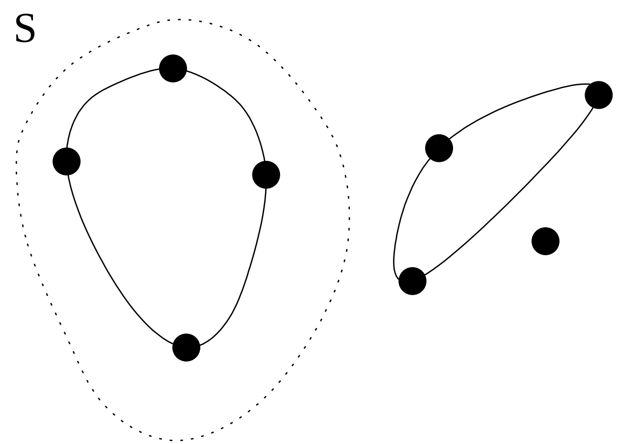


TSP Model

Minimize $\sum_{e \in E} c_e \cdot x_e$

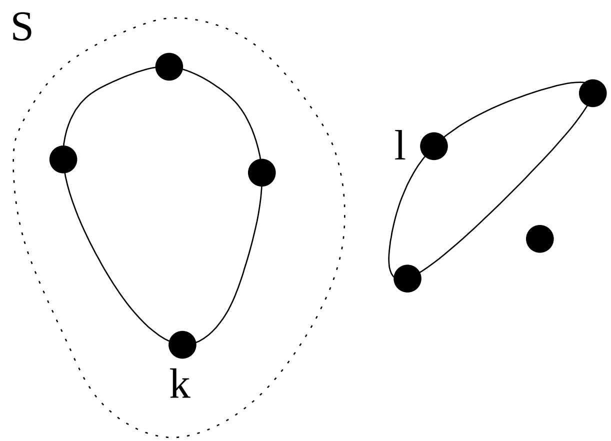
Subject To

$$
\begin{array}{rcll}
\sum_{e \in \delta(i)} x_e & = & 2 & \text{for } i \in V \\
\sum_{e \in \delta(S)} x_e & \geq & 2 & \text{for } \emptyset \subset S \subset V \\
x_e & \in & \{0,1\} & \text{for } e \in E
\end{array}
$$



STSP Model

$$\text{Minimize} \quad \sum_{e \in E} c_e \cdot x_e$$



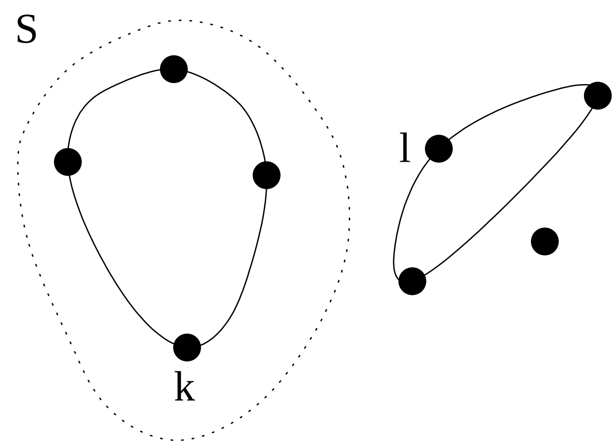
$$\begin{aligned}
\text{Subject To} \quad \sum_{e \in \delta(i)} x_e &= 2y_i \quad \text{for } i \in V \\
\sum_{e \in \delta(S)} x_e &\geq 2(y_k + y_l - 1) \\
&\text{for } \emptyset \subset S \subset V, k \in S, l \notin S \\
x_e &\in \{0,1\} \text{ for } e \in E \\
y_i &\in \{0,1\} \text{ for } i \in V
\end{aligned}$$

Budget

QSTSP Model

Maximize $$\sum_{\{i,j\}\in E} r_{ij} \cdot y_i y_j - \sum_{e\in E} c_e \cdot x_e$$

Subject To
$$\sum_{e\in\delta(i)} x_e = 2y_i \quad \text{for } i \in V$$
$$\sum_{e\in\delta(S)} x_e \geq 2(y_k + y_l - 1)$$
$$\text{for } \emptyset \subset S \subset V, k \in S, l \notin S$$
$$x_e \in \{0,1\} \text{ for } e \in E$$
$$y_i \in \{0,1\} \text{ for } i \in V$$
(Budget)



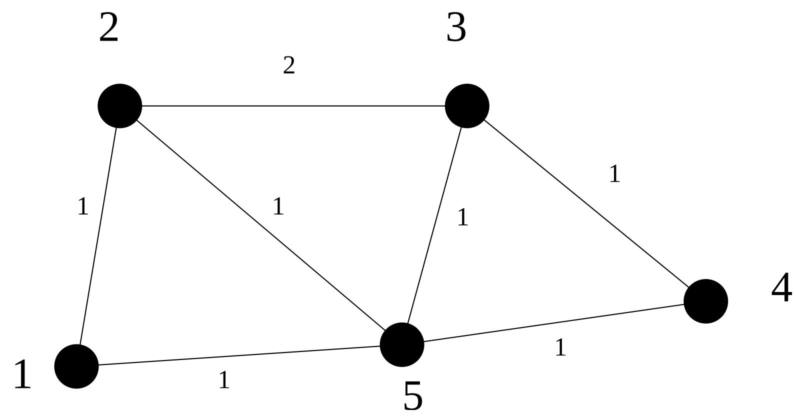
QSTSP in short

- NP-hard, generalizes TSP
- Model has quadratic objective.
- Linearized and solved by branch-and-cut
- Not very well studied
- Similar problem was considered by Gendreau, Labbe and Laporte (1995) which used TSP Heuristics.

Joint routing of working and protection capacity

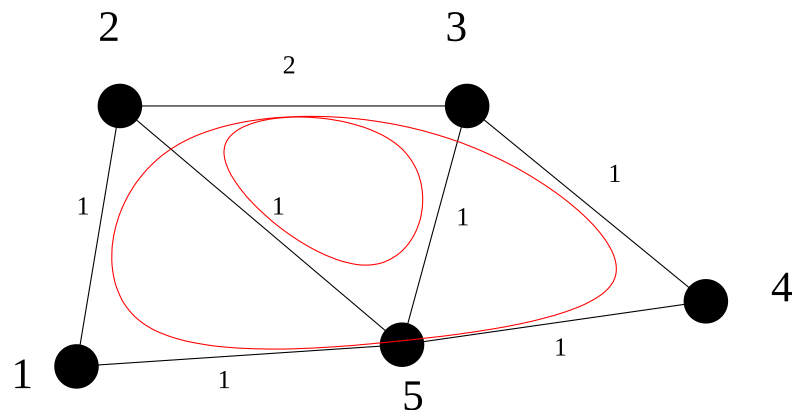
- Routing is usually done shortest-path, but may be beneficial to chose alternative shortest-paths or even longer paths.
- Optimize the p-cycles and the routing, jointly.
- Possible for both rings and p-cycles, but gain is highest for p-cycles.

Example: Joint Routing and Protection



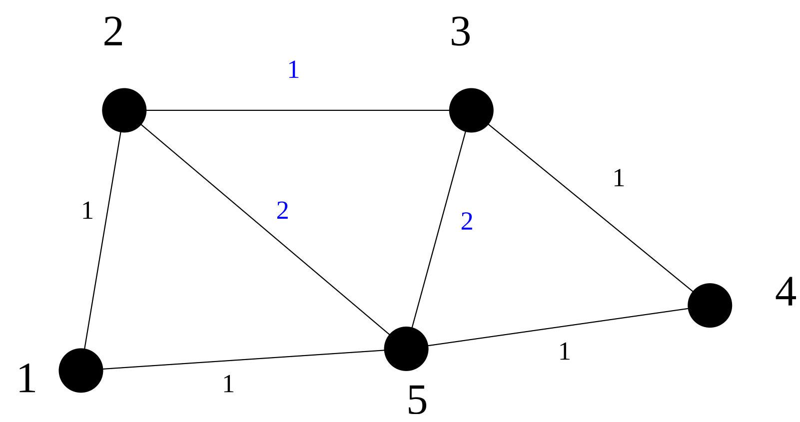
- Demands:
  1 corresponding to each link
  + 1 additional demand between 2 and 3.
  Shortest-path routing gives 1 link paths.

Example: Joint Routing and Protection



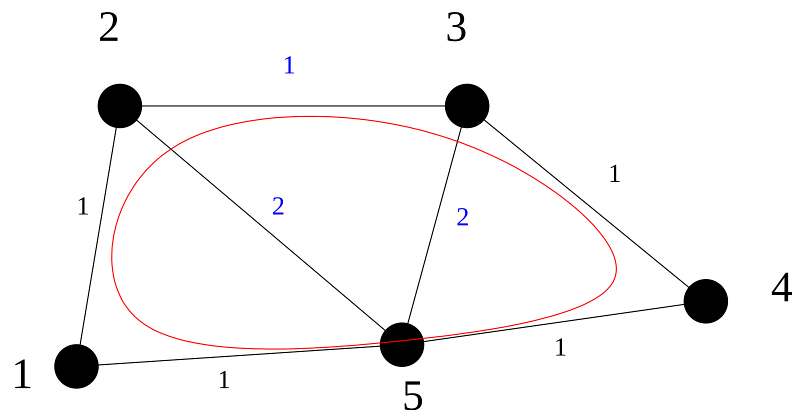
- Demands:
  1 corresponding to each link
  + 1 additional demand between 2 and 3.
  Shortest-path routing gives 1 link paths.
- Protection cost: $5 + 3 = 8$ ($8/8 = 100\%$)

Example: Joint Routing and Protection



- Demands:
  1 corresponding to each link
  + 1 additional demand between 2 and 3.
  Shortest-path routing gives 1 link paths.
- Protection cost: $5 + 3 = 8$ ($8/8 = 100\%$)
- Reroute 1 unit of link 2-3 to 2-5-3 (1 additional working flow)

Example: Joint Routing and Protection



- Demands:
  1 corresponding to each link
  + 1 additional demand between 2 and 3.
  Shortest-path routing gives 1 link paths.
- Protection cost: $5 + 3 = 8$ ($8/8 = 100\%$)
- Reroute 1 unit of link 2-3 to 2-5-3 (1 additional working flow)
- Protection cost is 5 but in total $5 + 1 = 6$ is used. ($6/8 = 75\%$)

Joint Routing and Protection

- Column Generation model
- 2 types of columns:
  - Paths
  - p-cycles
- Consider demands *not* working capacity.
- Generating paths is just shortest-path for each demand.
- Generation of p-cycles is the same as before.

Joint Routing and Protection Model

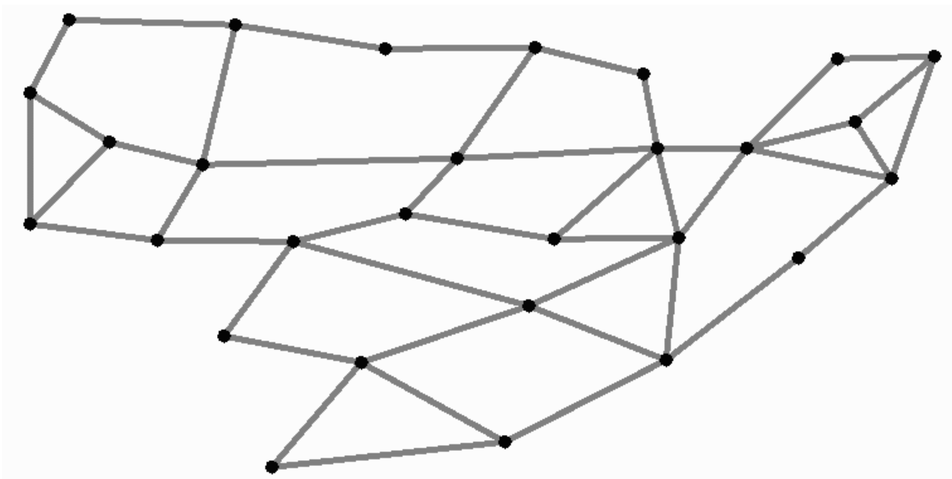
minimize:

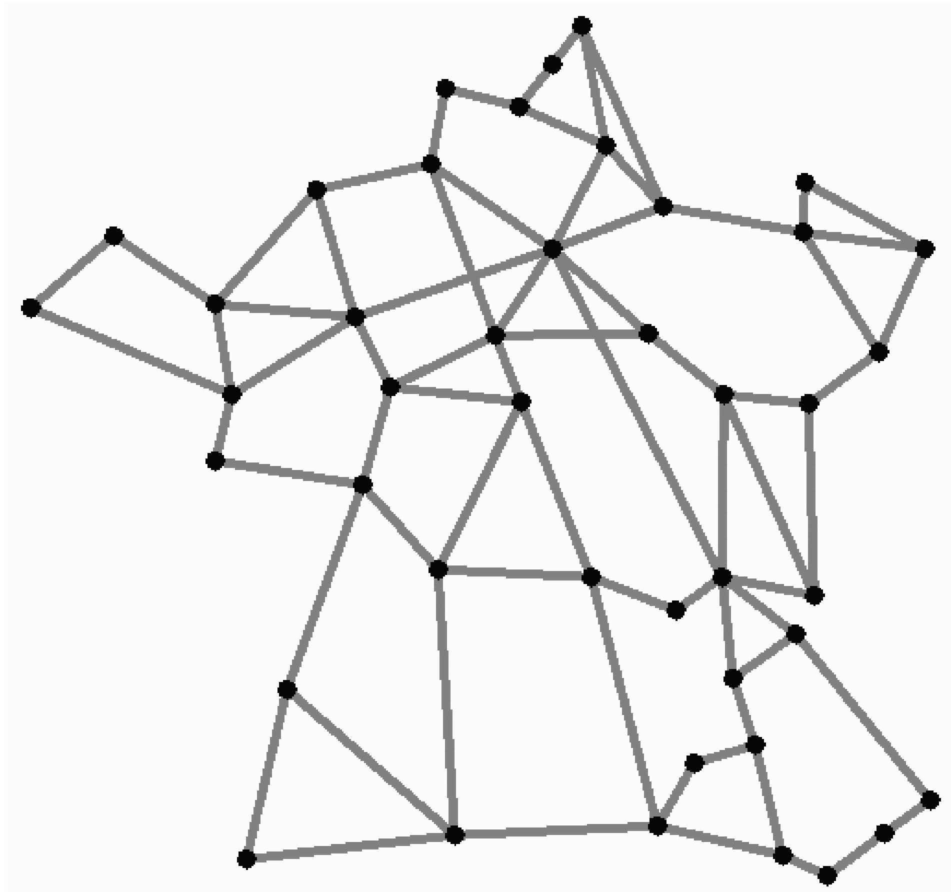
$$\sum_{r \in R} c_r \cdot x_r + \sum_{d \in D} \sum_{p \in P_d} c_p^d y_p^d$$

subject to:

$$\begin{aligned}
\sum_{p \in P_d} y_p^d &\geq v_d && \forall d \in D \\
\sum_{r \in R} a_r^e \cdot x_r &\geq \sum_{d \in D} \sum_{p \in P_d} a_p^{ed} y_p^d \quad (= w_e) && \forall e \in E \\
x_r &\in \{0, 1, 2, \ldots\} \\
y_p^d &\in \{0, 1, 2, \ldots\}
\end{aligned}$$

Test instances

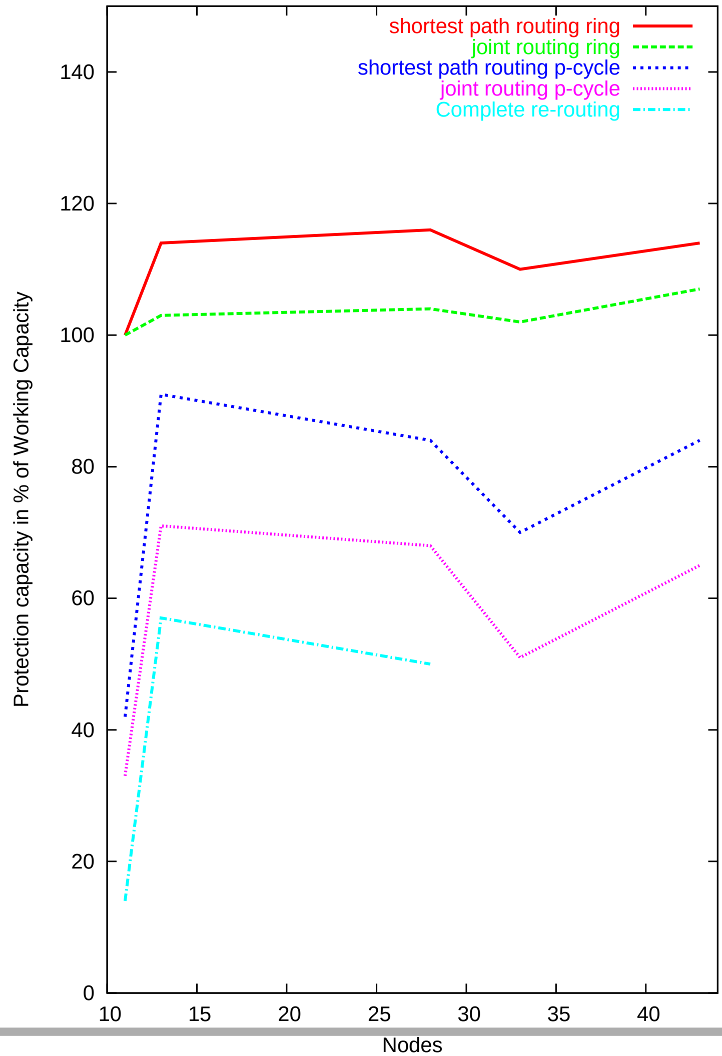




Networks

Network	#N	#L	N deg	W Cap	P Cap LB	P Cap LB
COST 239	11	26	4.73	86	12	14%
Europe	13	21	3.23	158	90	57%
USA	28	45	3.21	1273	641	50%
Italy	33	68	4.12	1718	-	-
France	43	71	3.3	3473	-	-

Protection capacity vs. working capacity



- Rings vs. p-cycles
- Joint vs. Non-joint
- Complete re-routing
- Gain is higher for dense networks

Running Times

	Total Time (sec.)	Master Time (%)	Sub Time (%)	Integer Time (%)	#p-cycles Gen.	#p-cycles Used	Avg. Sub Time (sec.)
Cost239	0.4	25.0 %	75.0 %	0.0 %	8	3	0.04
Europe	1.0	10.0 %	90.0 %	0.0 %	10	5	0.09
USA	44.3	2.0 %	30.2 %	67.7 %	17	11	0.79
Italy	160.6	3.1 %	78.1 %	18.7 %	44	14	2.85
France	360.1	2.1 %	96.3 %	1.6 %	43	22	8.07

- It is fast!
- Subproblem is most time consuming.
- It is very difficult to solve the subproblem when networks are dense.
- Better solutions could be obtained by generating some initial good p-cycles!? Or by doing Branch-and-Price.

The Integer Gap

	p-cycle Protection		Ring Protection	
	Shortest Path	Joint	Shortest Path	Joint
Cost239	4.65 %	2.33 %	2.50 %	5.45 %
Europe	0.59 %	0.00 %	0.99 %	0.75 %
USA	0.15 %	0.35 %	0.00 %	0.09 %
Italy	0.11 %	0.81 %	0.14 %	0.08 %
France	0.01 %	0.15 %	0.00 %	0.00 %

Conclusions

- p-cycles have the potential to be tomorrows standard in protecting circuit-oriented networks!
- From a theoreticians point of view the hardware cannot be to complicated to build.. (Ring hardware seems to work.)
- What about packet-switched networks?
- The computational results show that substantial savings are possible.
- The method presented to optimize capacity usage is by far the best... in the authors opinion.
- The method is, however, rather complicated to implement.

Future work and projects

- Make the hardware work!
- p-cycles in MPLS networks.
- Optimization:
  - Models closer to the real-world: Capacity, Grooming
  - Subproblem, QSTSP: Not very widely studied: Heuristics, optimal methods.
  - Master student: Complete re-routing lower bound by column generation.
- p-cycles home page: `tomato.edm.trlabs.ca/`p`-cycles`
- Technical Report: “Joint Optimization of Working and p-cycle Protection Capacity” by Thomas Stidsen and Tommy Thomadsen
- `www.imm.dtu.dk/~tt`