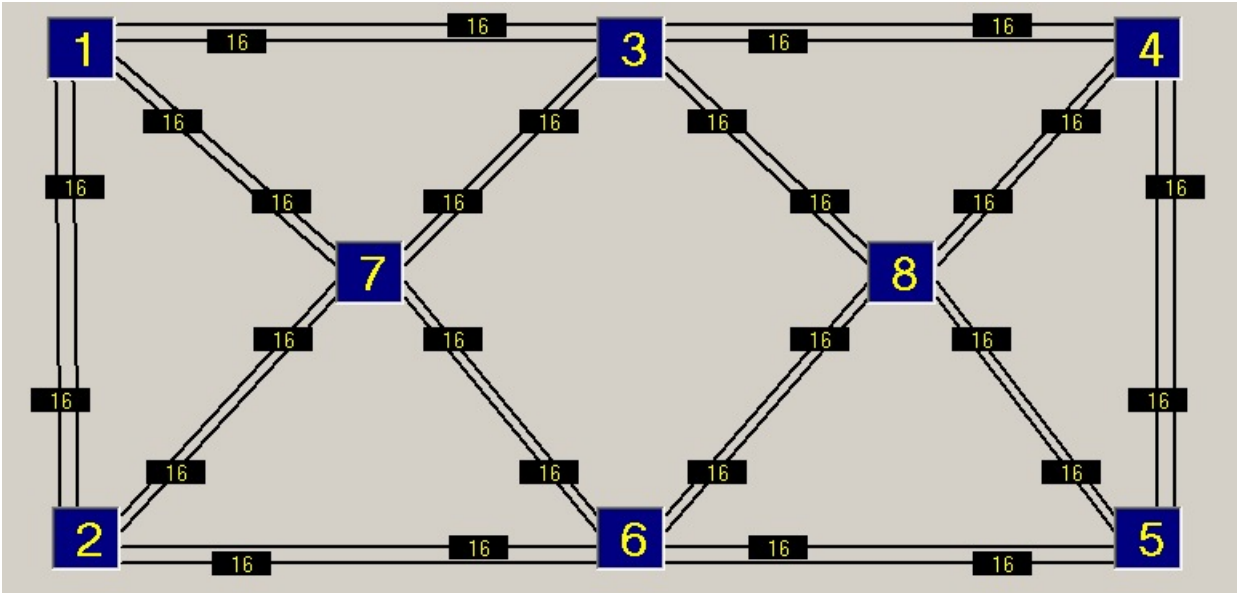
7.3. Резултати

7.3.1. Почетна топологија на мрежата

Вршени се тестирања на горните имплементации врз повеќе мрежни конфигурации. На слика 7.23 е прикажан иницијалниот објектен модел на мрежата која ќе биде математички моделирана и оптимизирана. Објектниот модел ги содржи сите потребни информации за WDM мрежата.



слика 7.23 Графички приказ на мрежата која е математички моделирана и оптимизирана

Сообраќајната матрица користена за оваа цел е:

табела 7.2 Сообраќајна матрица на мрежата "XWeb" прикажана на слика 7.23

	1	2	3	4	5	6	7	8
1				20				
2								10
3					20			
4	20							
5			20				20	
6								
7					20			
8		10						

Добиените резултати зависат од голем број на фактори. На пример изборот на MinNodeDegree и типот на кросконекти кои се на располагање. За мрежата на слика 7.23 се користени четири типа на кросконекти со 4, 8, 16 и 32 линиски единици т.е. интерфејси. Доколку е на располагање кросконект со 64 линиски единици со прифатлива цена можно е да се добијат пооптимални решенија.
Што се однесува до параметарот MinNodeDegree за мрежата на слика 7.23 избрано е да биде 3 за сите јазли освен за јазелот со реден број 8 за кој MinNodeDegree=4. Изборот на овој параметар е многу важен. Вредност помала од 2 не е прифатлива затоа што нема да е можна реставрација на единствениот линк кој е инцидентен за даденият јазол. Мала вредност на овој параметар, на пр. 2, може да предизвика исклучување на голем број на линкови кандидати во моделот за димензионирање т.е за планирање на капацитети и рутирање. Тоа може да предизвика нерешливост на моделите за планирање на резервните капацитети и рерутирање.

Исто така параметар на кој треба да се внимава при дизајнирање на иницијалната мрежа е MF т.е. максималниот број на влакна по насока (половина од вкупниот број на влакна во кабелот). Мала

вредност на овој параметар често знае да предизвика нерешливост на моделите (особено PreRes моделите).

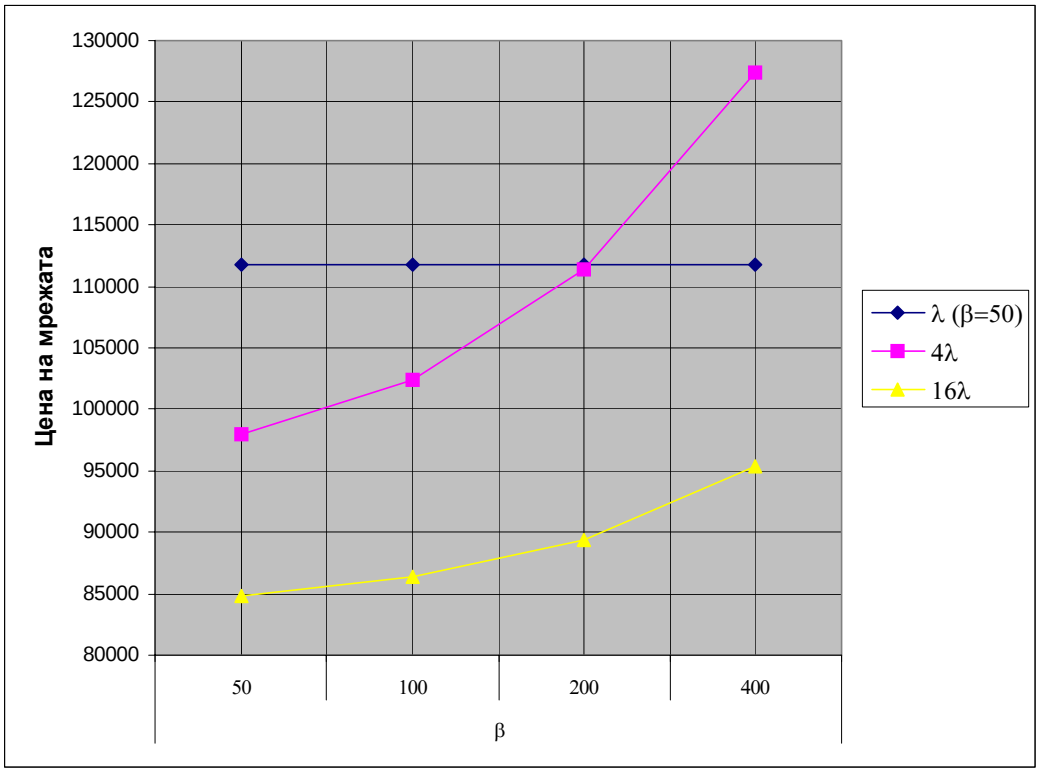
7.3.2. Планирање на капацитет и рутирање

Во оваа глава ќе биде опишано влијанието на α[51] и β[52] параметрите при планирање на мрежата. Цената на користење на линк (α цената) се разликува за секој оператор.

Некои оператори може да имаат темни влакна (dark fiber) и можат да ги користат истите без поголеми инвестиции ($\alpha \approx 0$). Други треба да инсталираат нови, да копаат, или да изнајмат оптичко влакно од трети оператори. Зависно од овие размислувања α цената ќе варира релативно во однос на дугите параметри на цена.

Се покажува дека со зголемување на α цената доаѓа до намалување на бројот на линкови кои се користат во резултантната топологија, бидејќи користењето на линк станува поскапо. Неупотребените линкови се исклучуваат од резултантантната топологија која се користи за планирање на резервните капацитети и рерутирање. Оваа тенденција е посилна доколку иницијалната топологија е меш со поголем степен на поврзаност, што подразбира дека повеќе линкови кандидати можат да се исклучат од резултантната топологија. Од експериментите се покажува дека треба да се се земат во предвид што е можно повеќе ***k*** најкратки рути со цел да се добијат повеќе неискористени линкови. Сепак, како што беше порано истакнато, зголемувањето на ***k*** најкратки рути кои се земаат во предвид ја зголемува пресметковната комплексност на проблемот.

	β			
	50	100	200	400
λ	111720	126120	154920	212520
4λ	97944	102444	111408	127440
16λ	84836	86336	89336	95336



слика 7.24 Влијание на β врз цената на мрежата за сообраќајната матрица дадена во табела 7.2

[51] α се однесува на трошоците за полагање на кабелот (копање, полагање на црево и сл.)

[52] β се однесува на оптичкото влакно т.е на терминалната опрема (мултиплексер, оптички засилувач или дополнително влакно за компензација на дисперзија)

Повеќето денешни транспортни мрежи користат оптички системи со еден канал. Важно прашање е кога да се изврши надградба на WDM и колку бранови должини треба да се користат. Кога се прави надградба од систем со една бранова должина на систем со повеќе бранови должини, треба да се инсталира терминална опрема како мултиплексери и демултиплексери. Исто така, можно е да бидат потребни оптички засилувачи со рамна крактеристика на засилување во поширок опсег. Сите овие компоненти влијаат врз β цената.

Влијанието на β е опишано со графиконот на слика 7.24 кој укажува на тоа кога се исплати да се врши надградба од оптички систем со една бранова должина на систем со четири бранови должини, односно надградба од систем со 4 на систем со 16 бранови должини. Во симулациите зементо е α=40, β се менува од 50 до 400, а γ=1. Цената на мрежната за системот со еден канал (т.е бранова должина) изнесува 111720 и се однесува за β=50. Користен е модел за VWP мрежа.

Пресечната точка на кривата од 4λ системот со онаа за 1λ системот покажува колку голема може да биде β цената за 4λ системот (≈200) за да цената на мрежата биде пониска од онаа за 1λ системот. На истиот графикон е дадена цената на мрежата за 16λ систем. За избраните вредности на β (50-400) и дадената сообраќајна матрица во табела 7.2 се покажува дека 16λ системот дава пониски цени за мрежата т.е. нема пресечни точки со кривите за lλ и 4λ системите. Сепак треба да се земе во предвид дека постои голема зависност на резултатите од сообраќајните барања. За поголеми сообраќајни барања, се покажува дека поголем број на бранови должини по влакно е оправдана инвестиција.
Како заклучок, кога α и β цените се повисоки во однос на γ цената, оптимизацијата ќе се обиде да собере повеќе сообраќај во помал број на оптички влакна и помалку линкови.

Исто така е јасно дека со земање во предвид на повеќе рути и поголем степен на поврзаност на иницијална мрежа (повеќе линкови кандидати), може да се очекува поголема добивка од користењето на оптимизацијата. Ова подразбира дека просторот со решенија за оптимизационата техника ќе биде поголем за сметка на соодветно поголем напор за пресметка (т.е. процесот на оптимизација ќе трае подолго).

7.3.3. Планирање на резервните капацитети и рерутирање

За оваа цел е користена сообраќајната матрица дадена во табела 7.2, со слениве параметри на цена за линковите: α=10000 β=2000, и γ=500. Бројот на бранови должини по оптичко влакно е 16, а кабелот што ги поврзува јазлите има 12 оптички влакна.

7.3.3.1. Зависност на цената на мрежата од бројот на "*k*" најкратки рути

Во табела 7.4 се прикажани цените на мрежата за сите типови на модели во зависност од бројот на ***k*** најкратки патеки кои се користат како предлог за рутирање т.е рерутирање. За разгледуваната мрежа при зададените сообраќајни барања се констатира дека изборот на бројот на патеки кандидати за рутирање не влијае многу врз цената на ресурсите од мрежата потребни за рутирање и доделување на работните капацитети (види

табела 7.3). Влијанието на изборот на KSH патеки за рерутирање врз цената на ресурсите потребни за рерутирање и доделување на резервните капацитети е поизразено но сепак релативно мало (види табела 7.4 и слика 7.25). За поголем бројот на патеки кандидати се добиваат 5-30% поефтини решенија за рутирање и доделување на резервниот капацитет на дадената мрежа.

табела 7.3 Зависност на цената на работниот капацитет од KSH

	k			k2 vs. k3 [%]	k3 vs. k10 [%]	k2 vs. k10 [%]
	2	3	10			
LinkRerouting VWP	568000	562000	562000	1,06	0,00	1,06
LinkRerouting WP	568000	562000	562000	1,06	0,00	1,06
PathRerouting VWP	568000	562000	562000	1,06	0,00	1,06
PathRerouting VWP free	568000	562000	562000	1,06	0,00	1,06
PathRerouting WPa	568000	562000	562000	1,06	0,00	1,06
PathRerouting WPa free	568000	562000	562000	1,06	0,00	1,06
PathRerouting WPb	568000	562000	562000	1,06	0,00	1,06
PathRerouting WPb free	568000	562000	562000	1,06	0,00	1,06
PathReroutingD VWP	568000	562000	562000	1,06	0,00	1,06
PathReroutingD VWP free	568000	562000	562000	1,06	0,00	1,06
PathReroutingD WPa	568000	562000	562000	1,06	0,00	1,06
PathReroutingD WPa free	568000	562000	562000	1,06	0,00	1,06
PathReroutingD WPb	568000	562000	562000	1,06	0,00	1,06
PathReroutingD WPb free	568000	562000	562000	1,06	0,00	1,06

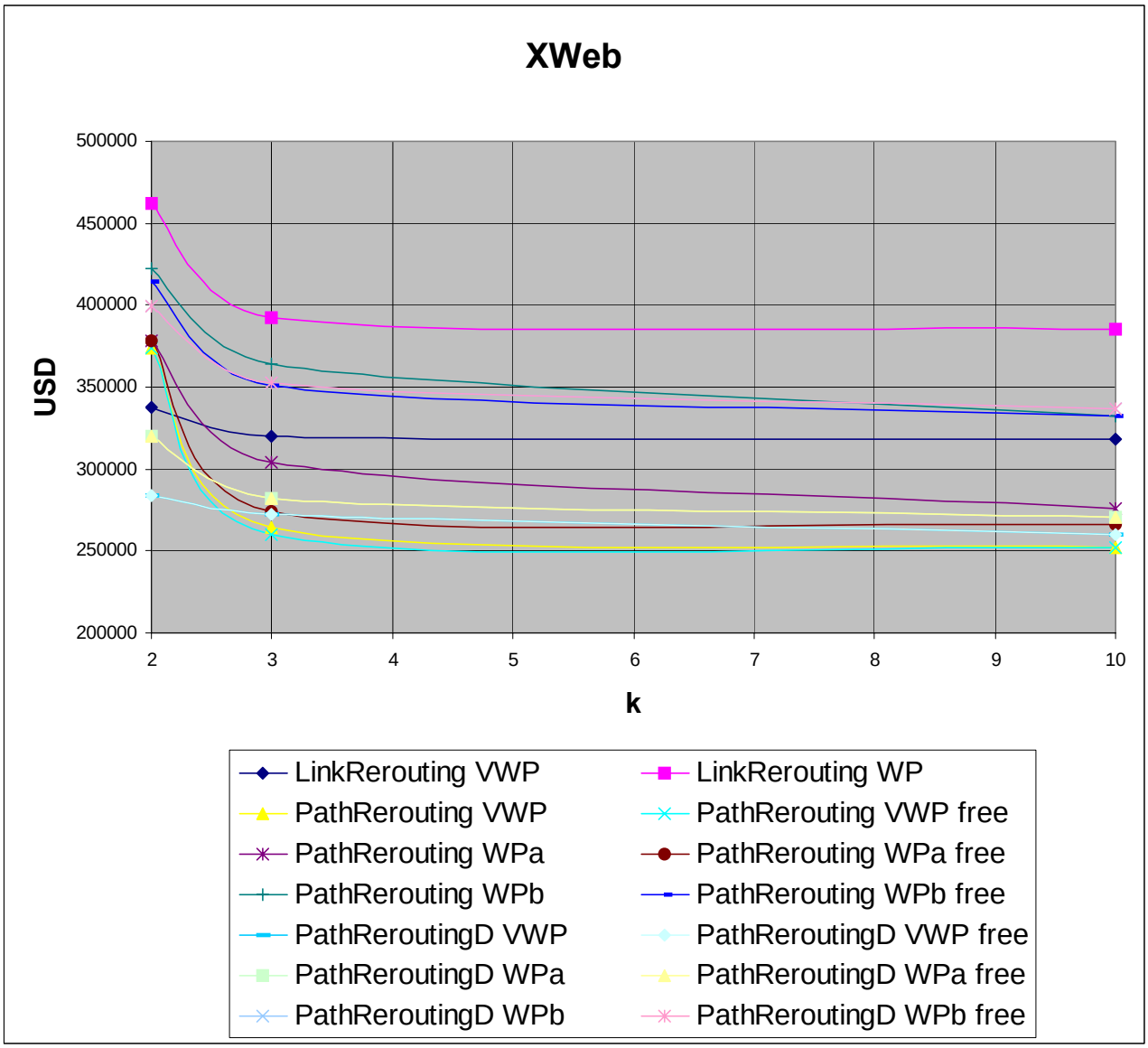
табела 7.4 Зависност на цената за резервниот капацитет на мрежата од бројот на KSH рути

	k			k2 vs. k3 [%]	k3 vs. k10 [%]	k2 vs. k10 [%]
	2	3	10			
LinkRerouting VWP	338000	320000	318000	5,33	0,63	5,92
LinkRerouting WP	462000	392000	385000	15,15	1,79	16,67
PathRerouting VWP	374000	264000	252000	29,41	4,55	32,62
PathRerouting VWP free	374000	260000	252000	30,48	3,08	32,62
PathRerouting WPa	378000	304000	276000	19,58	9,21	26,98
PathRerouting WPa free	378000	274000	266000	27,51	2,92	29,63
PathRerouting WPb	422000	364000	332000	13,74	8,79	21,33
PathRerouting WPb free	414000	351000	332000	15,22	5,41	19,81
PathReroutingD VWP	284000	272000	260000	4,23	4,41	8,45
PathReroutingD VWP free	284000	272000	260000	4,23	4,41	8,45
PathReroutingD WPa	320000	282000	271000	11,88	3,90	15,31
PathReroutingD WPa free	320000	282000	271000	11,88	3,90	15,31
PathReroutingD WPb	399000	353000	337000	11,53	4,53	15,54
PathReroutingD WPb free	399000	353000	337000	11,53	4,53	15,54

Цените прикажани во табелите се во произволна валута. Исто така, важно е да се истакне дека прикажаните цени за работните капацитети и резервните капацитети ги вклучуваат цените за типот на употребените јазли. Тоа значи дека ако се сумираат двете цени ќе се добие цена повисока од реалната, затоа што цените на јазлите се јавуваат двојно. Ова е последица на тоа што во моделите за димензионирање и рерутирање сме се решиле за оптимизација на употребените јазли. Ако се погледнат целните функции ф. 7.13 и ф. 7.14 ќе се види дека станува збор за втората сума. Доколку оваа сума се испушти од целната функција и се изврши оптимизација ќе се добие цената на ресурсите за рутирање на работните капацитети без вклучени јазли. Овој резултат сам за себе дава нереална слика за цената на мрежата, но ако се сумира со цената на ресурсите потребни за рерутирање и доделување на резервните капацитети (кај кој не сме ја исклучиле втората сума; види ф. 7.69, ф. 7.70 т.е. ф. 7.84 и ф. 7.85) ќе се добие точната цена на вкупната мрежа со 100% заштита од дефект на еден линк.

Треба повторно да се истакне дека добиените резултати многу зависат од изборот на мрежата, нејзиниот степен на поврзаност и сообраќајната матрица. Се постигнува поголема добивка од

користење на поголем број на патеки кандидати кај мрежи со поголем степен на поврзаност и поголеми сообраќајни барања.



слика 7.25 Зависност на цената на мрежата од бројот на KSH рути

7.3.3.2. Зависност на цената на мрежата од употребената реставрациона стратегија

Резултатите прикажани во табела 7.5 т.е на слика 7.26 се изразени како однос помеѓу резервните и работните оптички влакна. Од аспект на најголеми барања на резервен капацитет LR реставрационата стратегија е најскапа. PR и PRd се поефтини. Кај моделите со реставрација на патеки, ослободениот капацитет на прекинатите патеки на линковите кои не се во дефект овозможува повторна употреба на капацитетите на овие линкови и затоа се добива дополнителна заштеда на капацитет (PR+free и PRd+free).

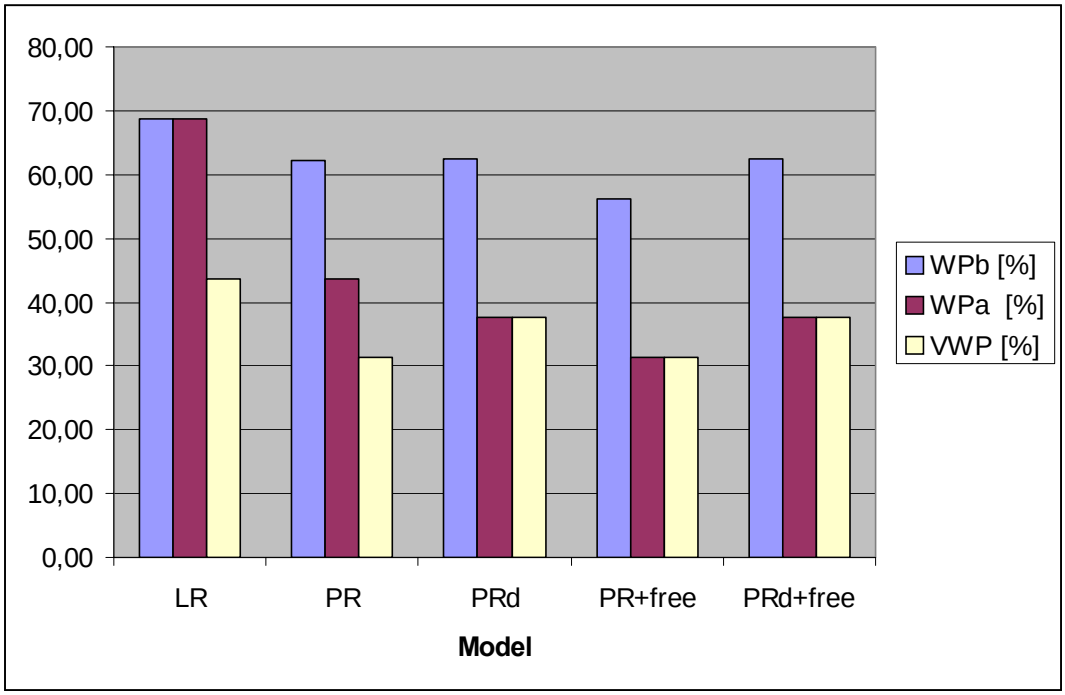
табела 7.5 Однос резервни/работни оптички влакна за различни реставрациони стратегии: WPa, WPb, и VWP

k=3	WPb [%]	WPa [%]	VWP [%]
LR	68,75	68,75	43,75
PR	62,50	43,75	31,25
PRd	62,50	37,50	37,50
PR+free	56,25	31,25	31,25
PRd+free	62,50	37,50	37,50

табела 7.6 Цена на мрежата и вкупен број на потребни работни и резервни оптички влакна и канали

	WPb						WPa						VWP					
	цена 1	цена 2	WF	WC	SF	SC	цена 1	цена 2	WF	WC	SF	SC	цена 1	цена 2	WF	WC	SF	SC
LR	562000	392000	32	296	22	336	562000	392000	32	296	22	336	562000	320000	32	296	14	284
PR	562000	364000	32	296	20	328	562000	304000	32	296	14	252	562000	274000	32	296	10	248
PRd	562000	353000	32	296	20	306	562000	282000	32	296	12	276	562000	272000	32	296	12	256
PR+free	562000	351000	32	296	18	310	562000	274000	32	296	10	268	562000	260000	32	296	10	240
PRd+free	562000	353000	32	296	20	306	562000	282000	32	296	12	276	562000	272000	32	296	12	256

Во горната табела колоната "цена 1" ја дава цената на каблите, работните влакна и канали и цената на јазлите препорачани од Dimensioning моделот. Колоната "цена 2" ја дава цената на резервните влакна и канали и цената на јазлите. Имено Rerouting моделот може да го промени типот на јазолот кандидат препорачан од Dimensioning моделот, па затоа вкупната цена на јазлите после Rerouting моделите може да биде повисока. Во другите колони се дадени вредностите на вкупниот потребен број на работни влакна (WF), работни канали (WC), резервни влакна (SF) и резервни канали (SC).



слика 7.26 Однос резервни/работни оптички влакна за различни реставрациони стратегии

Од табела 7.5 може да се изврши споредба на оптичките мрежни модели. Цената на резервниот капацитет во VWP моделот е пониска отколку кај WPa моделот, иако незначително. Значителна разлика може да се забележи при споредба на WPa и WPb моделите за случај на реставрација на патеки. Оттука може да се заклучи дека предавателите (ласери) со променлива бранова должина овозможуваат подобро искористување на резервните капацитети. Предноста од користење на претворувачи на бранова должина (кај VWP) е мала, но користењето на предаватели со променлива должина дава голема придобивка.

7.3.3.3. Зависност на цената на мрежата од употребениот линиски систем

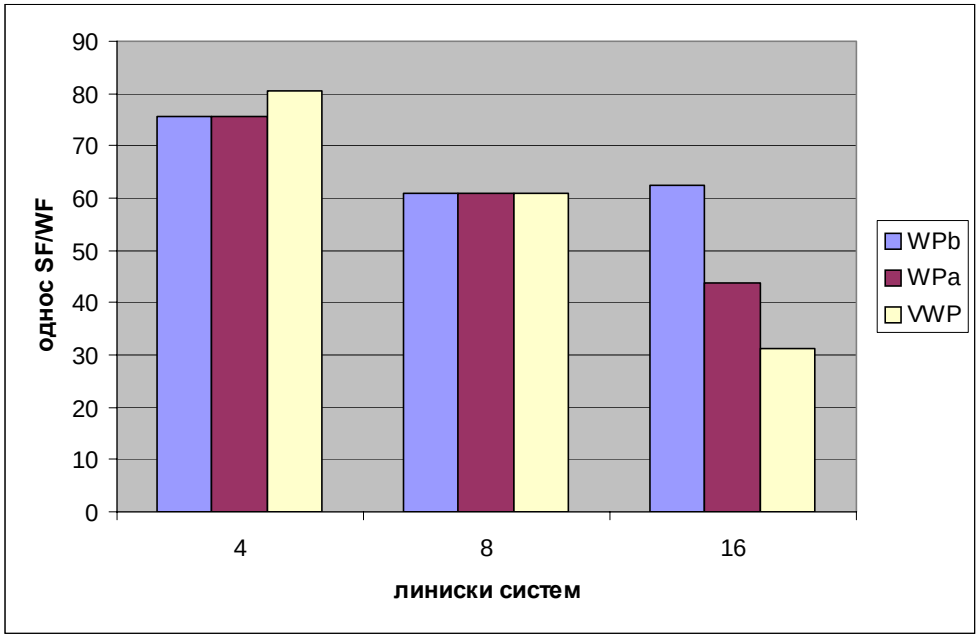
Во разгледуваната мрежа, извршена е оптимизација за различни капацитети на линковите, т.е. за 4, 8 и 16 бранови должини по оптичко влакно со користење на PR моделите опишани во глава 7.2.5.2 и имплементирани во глава 9.3. Во табела 7.7 и на слика 7.27 е прикажан односот на бројот на резервни (SF) и работни (WF) оптички влакна за трите различни типови на PR реставрација. Може да се заклучи дека бројот на резервни влакна споредено со работни влакна се намалува со зголемување на бројот на бранови должини по оптичко влакно за WPa и VWP моделите. Тоа не е така за WPb моделите. Од табела 7.7 може да се види дека редундантноста кај WPb мрежите со линиски систем со 16 (62,5%) бранови должини е поголема од редундантноста за мрежата со линиски систем со 8 бранови должини (60,9%). Кај VWP и WPa се забележува дека за поголем линиски систем се добива помала редундантност Ова може да се констатира и интуитивно. Пополнувањето на 16-те бранови должини по влакно не може да се направи така ефикасно како во случајот на помал број на бранови должини, па затоа остануваат многу неискористени канали. WPa и VWP дозволуваат пофлексибилно искористување на овие канали отколку WPb.

табела 7.7 Барања за резервен капацитет (редундантност) за мрежата од слика 7.23 за различни линиски системи: 4, 8 и 16

	λ		
	4	8	16
WPb [%]	75,6	60,9	62,5
WPa [%]	75,6	60,9	43,8
VWP [%]	80,5	60,9	31,3

табела 7.8 Цена на мрежата и вкупен број на потребни работни и резервни оптички влакна и канали

	λ																	
	4						8						16					
	цена 1	цена 2	WF	WC	SF	SC	цена 1	цена 2	WF	WC	SF	SC	цена 1	цена 2	WF	WC	SF	SC
WPb	824000	728000	22	280	62	288	664000	424000	46	304	28	256	562000	364000	32	296	20	328
WPa	824000	728000	22	280	62	288	664000	424000	46	304	28	256	562000	304000	32	296	14	252
VWP	824000	744000	82	280	66	304	664000	424000	46	304	28	256	562000	264000	32	296	10	248



слика 7.27 Барања за резервен капацитет за мрежата од слика 7.23 со капацитет на линковите од 4, 8 и 16

При анализата на повеќе различни иницијални мрежи при различни сообраќајни барања и капацитети на линкови забележана е силна зависност на резултатите од степенот на поврзаност на јазлите. Во мрежите со поголем степен на поврзаност, како што може и да се претпостави, барањата за резервен капацитет се помали. Ова се должи на фактот дека во ваквите мрежи постои поголемо количество на преостанат резервен капацитет, и тоа што меш структурата може подобро да се искористи за рерутирањето и искористувањето на резервните капацитети. Преостанатиот резервен капацитет го сочинуваат расположивите резервни канали во влакната кои се употребени за рутирање на работниот сообраќај.

8. Заклучок

Целта на овој труд беше изработка на рамка за планирање, моделирање и оптимизација на преживливи WDM мрежи, која е доволно флексибилна да прифати различни типови на проблеми и лесно надградлива и прилагодлива за нови модели од областа на мрежното планирање и оптимизација.

За таа цел претходно е дизајнирана алатка за објектно моделирање на транспортна мрежа од која се има целосна контрола на процесот на оптимизизација. Исто така од оваа алатка се врши промена на сите потребни параметри на мрежата и анализираат добиените резултати од процесот на оптимизација. Податоците добиени од објектниот модел се влезни податоци за математичките модели со кои се опишуваат различни мрежни проблеми, а особено од областа на оптичките мрежи. Математичките модели се моделирани во MPL (Matematical Programing Language) и решаваат со CPLEX 7.1 MIP (Mixed Integer Problem) солвер.

Во првите две глави (глава 2 3) е даден преглед на оптичките компоненти кои се користат во модерните WDM мрежи како и на архитектурите кои се користат за изградба на тие мрежи. Детално се опишани својствата на оптичкото влакно, причините за неисцрпниот капацитет кој го нуди, како и нелинеарните ефекти кои штетно влијаат врз преносот на информации. Други оптички компоненти кои се користат во WDM мрежите се оптички предаватели, приемници, засилувачи, претворувачи и комутациони елементи. За секоја од овие компоненти е даден детален опис и начин на имплементација. Тоа е направено бидејќи дизајнерите на оптичките мрежи од новата генерација мора да се свесни за својствата и ограничувањата на оптичкото влакно и оптичките компоненти со цел да соодветните протоколи и алгоритми ги искористат сите потенцијали на WDM мрежата. Често мрежниот дизајнер може да пристапи на WDM архитектурата од премногу едноставна, идеална, или традиционална мрежна гледна точка. За жал, ова може да го доведе дизајнерот во ситуација да направи нереалистички претпоставки за својствата на оптичкото влакно и оптичките компоненти, и тоа може да резултира во неостварлив или непрактичен дизајн.

Бидејќи целта е дизајн на преживливи WDM мрежи потребно е детално познавање на методите за заштита и реставрација на телекомуникациските мрежи. Затоа е посветена глава во која се изложени последните научни анализи што се однесуваат на преживливоста на оптичкото ниво. Намерата не е да се опишуват основните методи туку накратко да се изложат најновите аспекти во оваа област. Во оваа глава се изложени основните причини за користење на преживливост на оптичко ниво. Имено денешните транспортни мрежи (SONET/SDH, ATM, IP) имаат вградена техника за преживливост. Затоа тие можат да работат директно преку оптичкото влакно, и на тој начин да не зависат од другите нивои во поглед на спроведување на функциите за преживливост. Имајќи го ова во предвид објаснето е зошто е потребно оптичкото ниво да обезбедува сопствен механизам за заштита т.е реставрација. Исто така, изнесени се нови идеи за доделување на работните и резервните капацитети како и систематска споредба на барањата за капацитет на неколку методи за заштита и реставрација на меш во зависност од степенот на поврзаност на јазлите.

Мрежното планирање во целост се врши со користење на методите на математичко програмирање. Затоа во главата 5 се изложени основните принципи на линеарното програмирање, поставување на линеарниот проблем и некои негови примери. Исто така се опишани основните мрежни проблеми, за најкратки патеки, за максимален проток и за минимална цена на протокот. Овие модели се имплементирани во MPL и со нив се вршени експерименти во ИОО. За произволно избрани мрежи и сообраќајни матрици се извршени тестирања и прикажани се резултатите за секоја од овие имплементации.

Комплетното множество на светлосни патеки во WDM мрежата се смета дека формира виртуелна топологија (види глава 6.1) и низ неа треба да се рутира пакетски сообраќај. Затоа е развиена LP формулација за комплетен дизајн на виртуелната топологија, што вклучува избор на светлосните патеки кои ќе се користат, рутите на овие светлосни патеки, и интензитетот на

пакетскиот сообраќај низ овие светлосни патеки. Целта на овој модел е да се зголеми пропустливоста на мрежата (throughput). Моделот е имплементиран во MPL, а за произволна мрежа дизајнирана во интегрираната објекнта околина извршено е детално негово тестирање. Се покажува дека за разгледуваната мрежа и сообраќајна матрица поголемиот дел од сообраќајот се рутира во оптички домен без користење на електронска комутација. Вакви резултати може да се очекуваат во поголем број на случаи.

Откако множеството на светлосни патеки е избрано, треба тие да се рутираат низ мрежата и да им се додели бранова должина. Ова се нарекува проблем на рутирање и доделување на бранови должини (RWA-Routing and Wavelength Assignment) и базичната негова формулација е изложена во 6.2. И овој модел е практично имплементиран, а од тоа се извлечени корисни заклучоци. Моделот е прилично едноставен но рутирањето на патеките не е еднозначно определено.

Во последната глава од трудот (глава 7) сублимирано е теоретското знаење и практичното искуство од претходно обработените теми. Развиени се математички формулации и практични имплементации за два типа на WDM мрежи (VWP и WP) со три стратегии за реставрација при дефект на еден линк (реставрација на линк, патека, и патеки со диверзификација). Исто така, интерактивно е прикажано користењето на интегрираната објектна околина во текот на елаборацијата на математичките формулации.

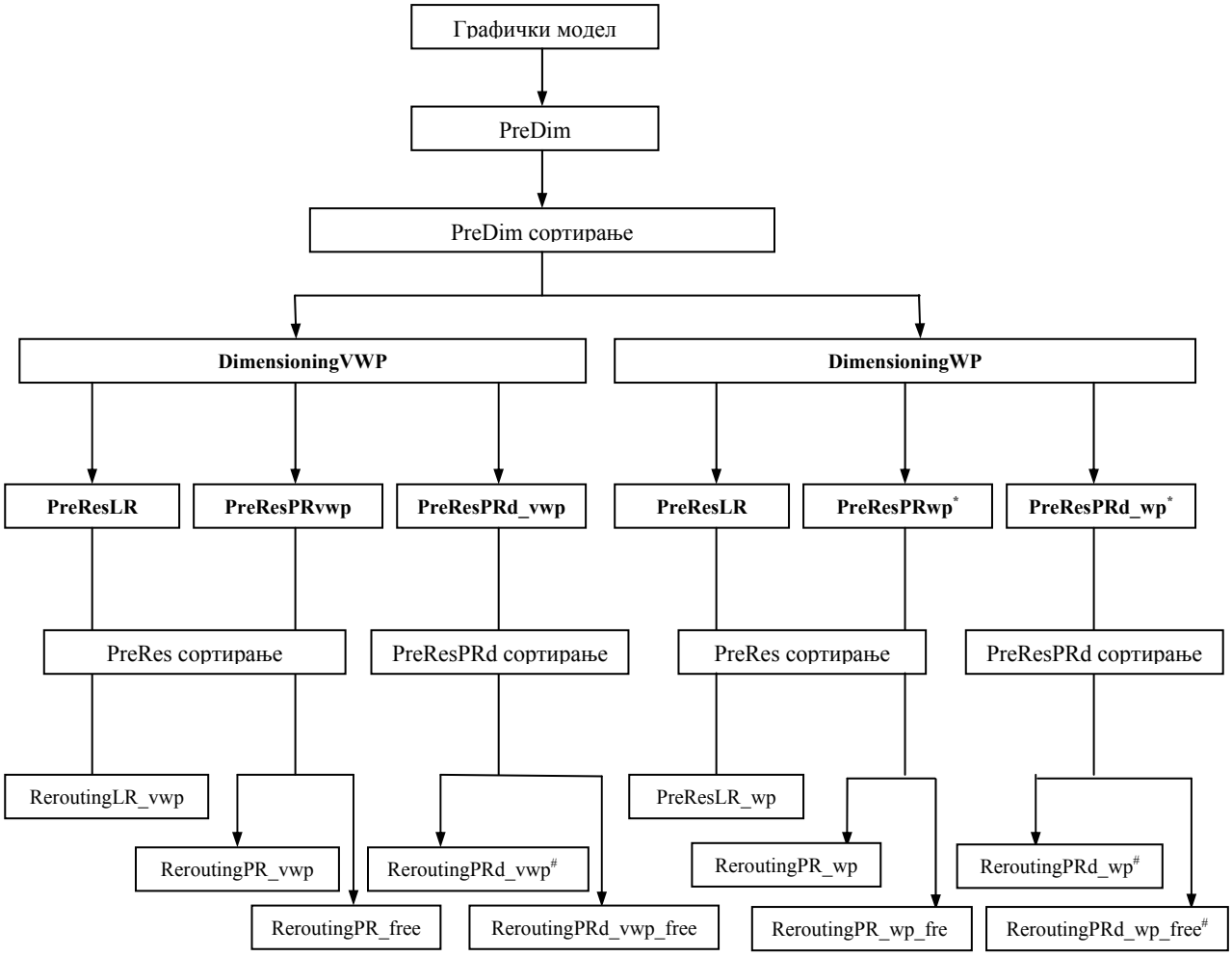
Извршена е детална анализа и споредба на рутирањето и доделувањето на работниот капацитет и рерутирањето и доделувањето на резервниот капацитет во WDM мрежите. Во тој контекст е извршена анализа за избор на бројот на бранови должини по оптичко влакно, и влијанието на цената на различните елементи од оптичката мрежа во процесот на планирање. Како што може да се види, кога се планира работниот капацитет и рутирањето, доколку користењето на линк предизвикува повисоки трошоци, треба да се земе во предвид меш мрежа со поголем степен на поврзаност која содржи повеќе линкови кандидати и поголем број на можни рути за рутирање на сообраќајното барање, со цел да се најде поефтино решение.

Што се однесува до реставрацијата, споредени се три реставрациони стратегии и испитано е влијанието на мрежаната топологија. Од аспект на планирање, нема голема корист од користење на претворувачи на бранови должини. Променливоста на ласерските извори, кога не се користат претворувачи на бранови должини се покажува дека е од голема корист.

Користа од употребата на претворувачи на бранова должина се зголемува со зголемување на бројот на бранови должини по влакно. Затоа, доколку WDM мрежата што се гради е со мал број на бранови должини по влакно, не е потребно користење на претворувачи на бранови должини. Како што може да се види во глава 7.3.3.3 во случај на 4λ систем цената на мрежата со користење на претворувачи на бранови должини е дури и повисока од мрежата без претворувачи. Сепак кај WDM мрежите кои користат линиски системи со голем капацитет употребата на претворувачи на бранови должини може да биде оправдана.

9. Додаток: Имплементација во MPL (Mathematical Programing Language)

На слика 9.1 е прикажан блок дијаграм кој го илустрира целиот процес на оптимизација на дадена WDM мрежа со користење на моделите развиени во овој труд. Прикажана е меѓусебната поврзаност на користените модели. Може да се забележи постоење на четири основни типови на модели: PreDim, Dimensioning, PreRes и Rerouting. Од нив се изведуваат сите други варијанти зависно од типот на мрежата (VWP или WP) т.е. од стратегијата за рерутирање (LR, PR, PRd). PreDim и PreRes моделите се користат за пронаоѓање на "***k***" најкратки патеки, а Dimensioning и Rerouting се модели кои ја остваруваат оптимизацијата на WDM мрежата.



слика 9.1 Модели за оптимизација на рутирањето на работните и резервните капацитети

* PreResPRwp и PreResPRd_wp моделите незначително се разликуваат во однос на нивните VWP варијанти па затоа нема да бидат посебно изложени. Разликата е во генерирањето на некои скалари, инаку променливите, целната функција и ограничувањата се потполно исти како во VWP моделите.
# ReroutingPRd моделите иако имаат посебно име потполно се исти како и ReroutingPR моделите. Разликата е во излезните податоци што ги генерира PreResPRd алгоритамот за сортирање, а кои потоа се користат како влезни за ReroutingPRd моделите. Предложените рути за реставрација од PreResPRd моделот се со различни линкови од работните.

9.1. Link Rerouting VWP

9.1.1. PreDim

TITLE
Perprocesor_For_Dimensioning

INDEX
node:=DATABASE("dataNodes","Node")
s=node;
d=node;
a=node;
b=node;
k=1..100;
j[a,b]:=DATABASE("dataPhysical", a=a, b=b);
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");

DATA
P[a,b]:=DATABASE("dataPhysical", "Pj", a="a", b="b");
D[s,d]:=DATABASE("dataTraffic", "dm", s="s",d="d");
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a",b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");

BINARY VARIABLES
UsedLink[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]
EXPORT REFILL TO DATABASE("TrafficRouting","deltaJPM", s="s", d="d", a="a", b="b", k="k");
Paths[m,k] WHERE D[m.s,m.d]>0 AND k<=D[m.s,m.d];

INTEGER VARIABLES
Fmax;
HD[s,d IN m,k] WHERE k<=D[s,d] EXPORT TO DATABASE("TrafficRouting","HD", s="s", d="d", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,k:UsedLink[m.s,m.d,a,b,k]);

FlowBal[m,node,k] WHERE k<=D[m.s,m.d]:
Paths[m,k] IF (m.s=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.b=node)=
Paths[m,k] IF (m.d=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.a=node);

Demands[m] WHERE D[m.s,m.d]>0:
SUM(k<=D[m.s,m.d]:Paths[m,k])=D[m.s,m.d];

CapConst[j] WHERE LA[j.a,j.b]>0:
SUM(m,k<=D[m.s,m.d]:UsedLink[m.s,m.d,j.a,j.b,k])<= MF[j.a,j.b]*LA[j.a,j.b];

Simetry[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a,b,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,k];

NoCirculation1[s,d IN m]:
SUM(b,k<=D[s,d]:UsedLink[s,d,a:=d,b,k]) + SUM(a,k<=D[s,d]:UsedLink[s,d,a,b:=s,k])=0;

NoCirculation2[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a:=b,b:=a,k]+UsedLink[s,d,a,b,k]<=1;

HopDistance[m,k] WHERE k<=D[m.s,m.d]:
HD[m.s,m.d,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k]);

9.1.2. Dimensioning VWP

TITLE
DimensioningVWP1k

INDEX
n:=DATABASE("dataNodes","Node");
s=n;
d=n;

```
a=n;
b=n;
j[a,b] = DATABASE("dataPhysical", a="a",b="b");m[s,d] = DATABASE("dataTraffic", s="s",d="d");
u[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a<b);
v[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a>b);
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");

DATA
MaxP=Last(p);
Alfa[a,b]= DATABASE("dataPhysical", a="a", b="b");
Beta[a,b]= DATABASE("dataPhysical", a="a", b="b");
Gama[a,b]= DATABASE("dataPhysical", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a", b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
D[s,d]:=DATABASE("dataTraffic","dm", s="s", d="d");
km[s,d]:=DATABASE("KSHnumber","km", s="s", d="d");
deltaJPM[s,d,a,b,p]:=DATABASE("SortedTraffic", s="s", d="d", a="a", b="b", p="k" WHERE k<=MaxP);
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
MinNodeDegree[n]=DATABASE("dataNodes","MinNodeDegree");

BINARY VARIABLES
deltaJ[a,b IN j] EXPORT REFILL TO DATABASE("outUsedLinks","deltaJ", a="a", b="b");
deltaNI[n,i];

INTEGER VARIABLES
UF[a,b IN j] EXPORT TO DATABASE ("outUsedLinks","Fibers", a="a", b="b");
UC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","Fj", a="a", b="b");
Fpm[p,s,d IN m] WHERE p<=km[s,d]
EXPORT REFILL TO DATABASE("outFlowOnRouteP","Fpm", p="p", s="s", d="d");
NodeType[n] EXPORT TO DATABASE("dataNodes","Type", n="Node");

MODEL

MIN TotNetCost=SUM(j: Alfa[j.a,j.b]*deltaJ[j.a,j.b] + Beta[j.a,j.b]*UF[j.a,j.b]  + Gama[j.a,j.b]*UC[j.a,j.b])
+SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

DemandConst[s,d IN m]:
SUM(p:Fpm[p,s,d])=D[s,d];

FlowOnLink[j]:
UC[j.a,j.b]>=SUM(m,p: deltaJPM[m.s,m.d,j.a,j.b,p]*Fpm[p,m.s,m.d]);

ChannelsInFiber[a,b]:
UC[a,b]<=LA[a,b]*UF[a,b];

UsedLink[j]:
UF[j.a,j.b]<=MF[j.a,j.b]*deltaJ[j.a,j.b];

NodeDegree1[n]:
Sum(u:deltaJN[u.a,u.b,n]*deltaJ[u.a,u.b])>=MinNodeDegree[n];

NodeDegree2[n]:
Sum(v:deltaJN[v.a,v.b,n]*deltaJ[v.a,v.b])>=MinNodeDegree[n];

NodeSize[n]:
Sum(i:deltaNI[n,i])=1;

SizeVSLinks[n]:
SUM(j:deltaJN[j.a,j.b,n]*UF[j.a,j.b])<=2*Sum(i: K[i]*deltaNI[n,i]);


Simetry1[a,b IN j]:
UC[a,b]=UC[a:=b,b:=a];

Simetry2[p,s,d IN m]:
Fpm[p,s,d]=Fpm[p,s:=d,d:=s];

LeastFiber[j]:
UF[j.a,j.b]>=deltaJ[j.a,j.b];

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.1.3. PreRes LR

```
TITLE
PreResLR1

INDEX
node:=DATABASE("qdfReducedNodes","Node")
a=node;
b=node;
s=node;
d=node;
k=1..100;
j[a,b]:=DATABASE("qdfReducedLinks", a="a", b="b");
m[s,d]:=DATABASE("outUsedLinks", s="a",d="b");

DATA
D[s,d]:=DATABASE("outUsedLinks", "Fj", s="a", d="b" WHERE Fj>0);
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a",b="b");
F[a,b]:=DATABASE("outUsedLinks", "Fj", a="a", b="b");
MF[a,b]:=DATABASE("qdfReducedLinks", "MFj", a="a",b="b");

BINARY VARIABLES
UsedLink[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]
EXPORT REFILL TO DATABASE("RestorationRouting","delta", s="s", d="d", a="a", b="b", k="k");
Paths[m,k] WHERE D[m.s,m.d]>0 AND k<=D[m.s,m.d];

INTEGER VARIABLES
Fmax;
HD[s,d IN m,k] WHERE k<=D[s,d] EXPORT TO DATABASE("RestorationRouting","HD", s="s", d="d", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,k:UsedLink[m.s,m.d,a,b,k]);

FlowBal[m,node,k] WHERE k<=D[m.s,m.d]:
Paths[m,k] IF (m.s=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.b=node AND j<>m)=
Paths[m,k] IF (m.d=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.a=node AND j<>m);

CapConst[m,j] WHERE LA[j.a,j.b]>0 AND j<>m:
SUM(k<=D[m.s,m.d]:UsedLink[m.s,m.d,j.a,j.b,k])<= LA[j.a,j.b]*MF[j.a,j.b]-F[j.a,j.b];

Demands[m] WHERE D[m.s,m.d]>0:
SUM(k<=D[m.s,m.d]:Paths[m,k])=D[m.s,m.d];

Simetry[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a,b,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,k];

NoCirculation1[s,d IN m]:
SUM(b,k<=D[s,d]:UsedLink[s,d,a:=d,b,k]) + SUM(a,k<=D[s,d]:UsedLink[s,d,a,b:=s,k])=0;

NoCirculation2[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a:=b,b:=a,k]+UsedLink[s,d,a,b,k]<=1;

HopDistance[m,k] WHERE k<=D[m.s,m.d]:
HD[m.s,m.d,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k]);
```

9.1.4. ReroutingLR_VWP

```
TITLE
ReroutingLR_vwp1

INDEX
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
j[s,d] = DATABASE("qdfReducedLinks", s="a",d="b");
x[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
i:=(1, 2, 3, 4);
```

```
p =DATABASE("dataNumberOfRoutes","NbRoutes");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
Pj[s,d]:=DATABASE("KSHnumberRes","Pj", s="s", d="d");

ASC[a,b]:=DATABASE("qdfAvailableChannels","ASCj", a="a", b="b");
F[s,d]=DATABASE("outUsedLinks","Fj", s="a", d="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");

delta_jxp[s,d,a,b,p]=DATABASE("SortedRestorationRouting","delta", s="s", d="d",a="a", b="b", p="k" WHERE k<=MaxP);
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN x] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN x] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fjp[s,d IN j,p] WHERE p<=Pj[s,d] EXPORT REFILL TO DATABASE("endReroutingFjp","Fjp", s="s", d="d", p="p");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(x:Beta[x.a,x.b]*SF[x.a,x.b] + Gama[x.a,x.b]*SC[x.a,x.b]) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[j]:
SUM(p<=Pj[j.s,j.d]:Fjp[j.s,j.d,p])=F[j.s,j.d];

EnoughSpareCapacitiy[j,x] WHERE j<>x:
SC[x.a,x.b]>=SUM(p: delta_jxp[j.s,j.d,x.a,x.b,p]*Fjp[j.s,j.d,p]);

SpareCapChannels[x]:
SC[x.a,x.b]-ASC[x.a,x.b]<=LA[x.a,x.b]*SF[x.a,x.b];

Simetry[s,d IN j,p] WHERE p<=Pj[s,d]:
Fjp[s,d,p]=Fjp[s:=d,d:=s,p];

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(x:deltaJN[x.a,x.b,n]*(UF[x.a,x.b] +SF[x.a,x.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[x]:
UF[x.a,x.b] +SF[x.a,x.b]<=MF[x.a,x.b];
```

9.2. Link Rerouting WP

9.2.1. PreDim

```
TITLE
Perprocesor_For_Dimensioning

INDEX
node:=DATABASE("dataNodes","Node")
s=node;
d=node;
a=node;
b=node;
k=1..100;
j[a,b]:=DATABASE("dataPhysical", a=a, b=b);
```

```
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");

DATA

P[a,b]:=DATABASE("dataPhysical", "Pj", a="a", b="b");
D[s,d]:=DATABASE("dataTraffic", "dm", s="s",d="d");
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a",b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");

BINARY VARIABLES
UsedLink[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]
EXPORT REFILL TO DATABASE("TrafficRouting","deltaJPM", s="s", d="d", a="a", b="b", k="k");
Paths[m,k] WHERE D[m.s,m.d]>0 AND k<=D[m.s,m.d];

INTEGER VARIABLES
Fmax;
HD[s,d IN m,k] WHERE k<=D[s,d] EXPORT TO DATABASE("TrafficRouting","HD", s="s", d="d", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,k:UsedLink[m.s,m.d,a,b,k]);

FlowBal[m,node,k] WHERE k<=D[m.s,m.d]:
Paths[m,k] IF (m.s=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.b=node)=
Paths[m,k] IF (m.d=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.a=node);

Demands[m] WHERE D[m.s,m.d]>0:
SUM(k<=D[m.s,m.d]:Paths[m,k])=D[m.s,m.d];

CapConst[j] WHERE LA[j.a,j.b]>0:
SUM(m,k<=D[m.s,m.d]:UsedLink[m.s,m.d,j.a,j.b,k])<= MF[j.a,j.b]*LA[j.a,j.b];

Simetry[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a,b,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,k];

NoCirculation1[s,d IN m]:
SUM(b,k<=D[s,d]:UsedLink[s,d,a:=d,b,k]) + SUM(a,k<=D[s,d]:UsedLink[s,d,a,b:=s,k])=0;

NoCirculation2[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a:=b,b:=a,k]+UsedLink[s,d,a,b,k]<=1;

HopDistance[m,k] WHERE k<=D[m.s,m.d]:
HD[m.s,m.d,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k]);
```

9.2.2. Dimensioning WP

```
TITLE
DimensionigWP1k

INDEX
n:=DATABASE("dataNodes","Node");
s=n;
d=n;
a=n;
b=n;
j[a,b] = DATABASE("dataPhysical", a="a",b="b");
m[s,d] = DATABASE("dataTraffic", s="s",d="d");
u[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a<b);
v[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a>b);
lambda:=DATABASE("SQLlambda", lambda="ind");
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");

DATA
MaxP=Last(p);
Alfa[a,b]= DATABASE("dataPhysical", a="a", b="b");
Beta[a,b]= DATABASE("dataPhysical", a="a", b="b");
Gama[a,b]= DATABASE("dataPhysical", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a", b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
D[s,d]:=DATABASE("dataTraffic","dm", s="s", d="d");
km[s,d]:=DATABASE("KSHnumber","km", s="s", d="d");
deltaJPM[s,d,a,b,p]:=DATABASE("SortedTraffic", s="s", d="d", a="a", b="b", p="k" WHERE k<=MaxP);
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
MinNodeDegree[n]=DATABASE("dataNodes","MinNodeDegree");

BINARY VARIABLES
deltaJ[a,b IN j] EXPORT REFILL TO DATABASE("outUsedLinks","deltaJ", a="a", b="b");
deltaNI[n,i];

INTEGER VARIABLES
UF[a,b IN j] EXPORT TO DATABASE ("outUsedLinks","Fibers", a="a", b="b");
UC[a,b IN j,lambda] WHERE lambda<=LA[a,b]
EXPORT REFILL TO DATABASE("outFlowPerLambda","Fjl", a="a", b="b", lambda="lambda");
allUC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","Fj", a="a", b="b");
Fmpl[p,s,d IN m,lambda] WHERE p<=km[s,d]
EXPORT REFILL TO DATABASE("outFlowOnRouteP","Fpm", p="p", s="s", d="d", lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","Type", n="Node");

MODEL

MIN TotNetCost=SUM(j: Alfa[j.a,j.b]*deltaJ[j.a,j.b] + Beta[j.a,j.b]*UF[j.a,j.b]
+ Gama[j.a,j.b]*SUM(lambda:UC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

DemandConst[m]:
SUM(p,lambda:Fmpl[p,m.s,m.d,lambda])=D[m.s,m.d];

FlowOnLink[j,lambda] WHERE lambda<=LA[j.a,j.b]:
UC[j.a,j.b,lambda]>=SUM(m,p: deltaJPM[m.s,m.d,j.a,j.b,p]*Fmpl[p,m.s,m.d,lambda]);

ChannelsInFiber[a,b,lambda] WHERE lambda<=LA[a,b]
EXPORT REFILL Slack TO DATABASE("outChannelsInFiber", "ASCj", a="a", b="b", lambda="lambda"):
UC[a,b,lambda]<=UF[a,b];

UsedLink[j]:
UF[j.a,j.b]<=MF[j.a,j.b]*deltaJ[j.a,j.b];

NodeDegree1[n]:
Sum(u:deltaJN[u.a,u.b,n]*deltaJ[u.a,u.b])>=MinNodeDegree[n];

NodeDegree2[n]:
Sum(v:deltaJN[v.a,v.b,n]*deltaJ[v.a,v.b])>=MinNodeDegree[n];

NodeSize[n]:
Sum(i:deltaNI[n,i])=1;
```

```
SizeVSLinks[n]:
SUM(j:deltaJN[j.a,j.b,n]*UF[j.a,j.b])<=2*Sum(i: K[i]*deltaNI[n,i]);

MergeFlowOnLInk[a,b IN j]:
SUM(lambda:UC[a,b,lambda])=allUC[a,b];

Symetry1[a,b IN j,lambda]:
UC[a,b,lambda]=UC[a:=b,b:=a,lambda];

Symetry2[p,s,d IN m,lambda]:
Fmpl[p,s,d,lambda]=Fmpl[p,s:=d,d:=s,lambda];

LeastFiber[j]:
UF[j.a,j.b]>=deltaJ[j.a,j.b];

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.2.3. PreRes LR

```
TITLE
PreResLR1

INDEX
node:=DATABASE("qdfReducedNodes","Node")
a=node;
b=node;
s=node;
d=node;
k=1..100;
j[a,b]:=DATABASE("qdfReducedLinks", a="a", b="b");
m[s,d]:=DATABASE("outUsedLinks", s="a",d="b");

DATA
D[s,d]:=DATABASE("outUsedLinks", "Fj", s="a", d="b" WHERE Fj>0);
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a",b="b");
F[a,b]:=DATABASE("outUsedLinks", "Fj", a="a", b="b");
MF[a,b]:=DATABASE("qdfReducedLinks", "MFj", a="a",b="b");

BINARY VARIABLES
UsedLink[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]
EXPORT REFILL TO DATABASE("RestorationRouting","delta", s="s", d="d", a="a", b="b", k="k");
Paths[m,k] WHERE D[m.s,m.d]>0 AND k<=D[m.s,m.d];

INTEGER VARIABLES
Fmax;
HD[s,d IN m,k] WHERE k<=D[s,d] EXPORT TO DATABASE("RestorationRouting","HD", s="s", d="d", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,k:UsedLink[m.s,m.d,a,b,k]);

FlowBal[m,node,k] WHERE k<=D[m.s,m.d]:
Paths[m,k] IF (m.s=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.b=node AND j<>m)=
Paths[m,k] IF (m.d=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.a=node AND j<>m);

CapConst[m,j] WHERE LA[j.a,j.b]>0 AND j<>m:
SUM(k<=D[m.s,m.d]:UsedLink[m.s,m.d,j.a,j.b,k])<= LA[j.a,j.b]*MF[j.a,j.b]-F[j.a,j.b];

Demands[m] WHERE D[m.s,m.d]>0:
SUM(k<=D[m.s,m.d]:Paths[m,k])=D[m.s,m.d];

Simetry[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a,b,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,k];

NoCirculation1[s,d IN m]:
SUM(b,k<=D[s,d]:UsedLink[s,d,a:=d,b,k]) + SUM(a,k<=D[s,d]:UsedLink[s,d,a,b:=s,k])=0;

NoCirculation2[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a:=b,b:=a,k]+UsedLink[s,d,a,b,k]<=1;
```

```
HopDistance[m,k] WHERE k<=D[m.s,m.d]:
HD[m.s,m.d,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k]);
```

9.2.4. ReroutingLR_WP

```
TITLE
ReroutingLR_wp1

INDEX
n:=DATABASE("dataNodes","Node");
s=n;
d=n;
a=n;
b=n;
j[s,d] = DATABASE("dataPhysical", s="a",d="b");
x[a,b] = DATABASE("dataPhysical", a="a",b="b");
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
lambda:=DATABASE("SQLlambda", lambda="ind");


DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("dataPhysical", a="a", b="b");
Gama[a,b]= DATABASE("dataPhysical", a="a", b="b");
LA[s,d]:=DATABASE("dataPhysical", "LAj", s="a", d="b");
Pj[s,d]:=DATABASE("KSHnumberRes","Pj", s="s", d="d");
ASC[a,b,lambda]:=DATABASE("outChannelsInFiber","ASCj", a="a", b="b", lambda="lambda");
F[s,d,lambda]=DATABASE("outFlowPerLambda","Fjl", s="a", d="b", lambda="lambda");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
delta_jxp[s,d,a,b,p]=DATABASE("SortedRestorationRouting","delta", s="s", d="d",a="a", b="b", p="k" WHERE k<=MaxP);
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN x] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[x,lambda];
allSC[a,b IN x] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fjlp[s,d IN j,lambda,p] WHERE p<=Pj[s,d] AND lambda<=LA[s,d]
EXPORT REFILL TO DATABASE("endReroutingFjp",Fjlp="Fjp", s="s", d="d", p="p",lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost=SUM(x:Beta[x.a,x.b]*SF[x.a,x.b] + Gama[x.a,x.b]*SUM(lambda: SC[x,lambda]))
+SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[j,lambda]:
SUM(p<=Pj[j.s,j.d]:Fjlp[j.s,j.d,lambda,p])=F[j.s,j.d,lambda];

EnoughSpareCapacity[j,x,lambda] WHERE j<>x:
SC[x,lambda]>=SUM(p: delta_jxp[j.s,j.d,x.a,x.b,p]*Fjlp[j.s,j.d,lambda,p]);

SpareCapChannels[x,lambda]:
SC[x,lambda]-ASC[x.a,x.b,lambda]<=SF[x.a,x.b];

Simetry[s,d IN j,lambda,p] WHERE p<=Pj[s,d]:
Fjlp[s,d,lambda,p]=Fjlp[s:=d,d:=s,lambda,p];

MergeFlowOnSpareLinks[x]:
allSC[x.a,x.b]=SUM(lambda:SC[x,lambda]);

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(x:deltaJN[x.a,x.b,n]*(UF[x.a,x.b] +SF[x.a,x.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
```

```
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[x]:
UF[x.a,x.b] +SF[x.a,x.b]<=MF[x.a,x.b];
```

9.3. Path Rerouting VWP

9.3.1. PreDim

TITLE
Perprocesor_For_Dimensioning

INDEX
node:=DATABASE("dataNodes","Node")
s=node;
d=node;
a=node;
b=node;
k=1..100;
j[a,b]:=DATABASE("dataPhysical", a=a, b=b);
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");

DATA

P[a,b]:=DATABASE("dataPhysical", "Pj", a="a", b="b");
D[s,d]:=DATABASE("dataTraffic", "dm", s="s",d="d");
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a",b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");

BINARY VARIABLES
UsedLink[s,d IN m,a,b IN j,k] WHERE k<=D[s,d] EXPORT REFILL TO DATABASE("TrafficRouting","deltaJPM", s="s", d="d", a="a", b="b", k="k");
Paths[m,k] WHERE D[m.s,m.d]>0 AND k<=D[m.s,m.d];

INTEGER VARIABLES
Fmax;
HD[s,d IN m,k] WHERE k<=D[s,d] EXPORT TO DATABASE("TrafficRouting","HD", s="s", d="d", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,k:UsedLink[m.s,m.d,a,b,k]);

FlowBal[m,node,k] WHERE k<=D[m.s,m.d]:
Paths[m,k] IF (m.s=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.b=node)=
Paths[m,k] IF (m.d=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.a=node);

Demands[m] WHERE D[m.s,m.d]>0:
SUM(k<=D[m.s,m.d]:Paths[m,k])=D[m.s,m.d];

CapConst[j] WHERE LA[j.a,j.b]>0:
SUM(m,k<=D[m.s,m.d]:UsedLink[m.s,m.d,j.a,j.b,k])<= MF[j.a,j.b]*LA[j.a,j.b];

Simetry[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a,b,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,k];

NoCirculation1[s,d IN m]:
SUM(b,k<=D[s,d]:UsedLink[s,d,a:=d,b,k]) + SUM(a,k<=D[s,d]:UsedLink[s,d,a,b:=s,k])=0;

NoCirculation2[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a:=b,b:=a,k]+UsedLink[s,d,a,b,k]<=1;

HopDistance[m,k] WHERE k<=D[m.s,m.d]:
HD[m.s,m.d,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k]);

9.3.2. Dimensioning VWP

TITLE
DimensioningVWP1k

```
INDEX
n:=DATABASE("dataNodes","Node");
s=n;
d=n;
a=n;
b=n;
j[a,b] = DATABASE("dataPhysical", a="a",b="b");
m[s,d] = DATABASE("dataTraffic", s="s",d="d");
u[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a<b);
v[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a>b);
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");

DATA
MaxP=Last(p);
Alfa[a,b]= DATABASE("dataPhysical", a="a", b="b");
Beta[a,b]= DATABASE("dataPhysical", a="a", b="b");
Gama[a,b]= DATABASE("dataPhysical", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a", b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
D[s,d]:=DATABASE("dataTraffic","dm", s="s", d="d");
km[s,d]:=DATABASE("KSHnumber","km", s="s", d="d");
deltaJPM[s,d,a,b,p]:=DATABASE("SortedTraffic", s="s", d="d", a="a", b="b", p="k" WHERE k<=MaxP);
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
MinNodeDegree[n]=DATABASE("dataNodes","MinNodeDegree");

BINARY VARIABLES
deltaJ[a,b IN j] EXPORT REFILL TO DATABASE("outUsedLinks","deltaJ", a="a", b="b");
deltaNI[n,i];

INTEGER VARIABLES
UF[a,b IN j] EXPORT TO DATABASE ("outUsedLinks","Fibers", a="a", b="b");
UC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","Fj", a="a", b="b");
Fpm[p,s,d IN m] WHERE p<=km[s,d]
EXPORT REFILL TO DATABASE("outFlowOnRouteP","Fpm", p="p", s="s", d="d");
NodeType[n] EXPORT TO DATABASE("dataNodes","Type", n="Node");

MODEL

MIN TotNetCost=SUM(j: Alfa[j.a,j.b]*deltaJ[j.a,j.b] + Beta[j.a,j.b]*UF[j.a,j.b]  + Gama[j.a,j.b]*UC[j.a,j.b]) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

DemandConst[s,d IN m]:
SUM(p:Fpm[p,s,d])=D[s,d];

FlowOnLink[j]:
UC[j.a,j.b]>=SUM(m,p: deltaJPM[m.s,m.d,j.a,j.b,p]*Fpm[p,m.s,m.d]);

ChannelsInFiber[a,b]:
UC[a,b]<=LA[a,b]*UF[a,b];

UsedLink[j]:
UF[j.a,j.b]<=MF[j.a,j.b]*deltaJ[j.a,j.b];

NodeDegree1[n]:
Sum(u:deltaJN[u.a,u.b,n]*deltaJ[u.a,u.b])>=MinNodeDegree[n];

NodeDegree2[n]:
Sum(v:deltaJN[v.a,v.b,n]*deltaJ[v.a,v.b])>=MinNodeDegree[n];

NodeSize[n]:
Sum(i:deltaNI[n,i])=1;

SizeVSLinks[n]:
SUM(j:deltaJN[j.a,j.b,n]*UF[j.a,j.b])<=2*Sum(i: K[i]*deltaNI[n,i]);

Simetry1[a,b IN j]:
UC[a,b]=UC[a:=b,b:=a];

Simetry2[p,s,d IN m]:
Fpm[p,s,d]=Fpm[p,s:=d,d:=s];

LeastFiber[j]:
UF[j.a,j.b]>=deltaJ[j.a,j.b];

TypeOfNode[n]:
```

```
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.3.3. PreResPR_VWP

```
TITLE
PreResPRvwp

INDEX
node:=DATABASE("qdfReducedNodes","Node");
s=node;
d=node;
a=node;
b=node;
e=node;
f=node;
k=1..100;
j[a,b]:=DATABASE("qdfReducedLinks", a="a", b="b");
x[e,f]:=DATABASE("qdfReducedLinks", e="a", f="b");
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");

DATA
D[s,d,e,f]:=DATABASE("InputDm", "dm", s="s", d="d", e="a", f="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a",b="b");
MF[a,b]:=DATABASE("qdfReducedLinks", "MFj", a="a",b="b");
Fj[a,b]:=DATABASE("outUsedLinks", "Fj", a="a", b="b");
X[a,b IN j,e,f IN x]:=IF(a=e AND b=f )Then 1 ENDIF;

BINARY VARIABLES
UsedLink[s,d IN m, a,b IN j,k,e,f IN x] WHERE k<=D[s,d,e,f] AND X[a,b,e,f]=0
EXPORT REFILL TO DATABASE("RestorationRouting","delta", s="s", d="d", a="a", b="b", k="k", e="e",f="f");
Paths[m,k,x] WHERE D[m.s,m.d,x.e,x.f]>0 AND k<=D[m.s,m.d,x.e,x.f];

INTEGER VARIABLES
Fmax[e,f IN x];
HD[s,d IN m,k,e,f IN x] WHERE k<=D[s,d,e,f] EXPORT TO DATABASE("RestorationRouting","HD", s="s", d="d", k="k", e="e",f="f");

MODEL
MIN SUM(x:Fmax[x.e,x.f]);

SUBJECT TO

MaximumFlow[a,b IN j,e,f IN x]:
Fmax[e,f]>=SUM(m, k<=D[m.s,m.d,e,f]:UsedLink[m.s,m.d,a,b,k,e,f]);

FlowBal[m,x,node,k] WHERE k<=D[m.s,m.d,x.e,x.f]:
Paths[m,k,x] IF (m.s=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k,x.e,x.f] WHERE j.b=node)=
Paths[m,k,x] IF (m.d=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k,x.e,x.f] WHERE j.a=node);

CapConst[j,x] WHERE LA[j.a,j.b]>0:
SUM(m,k<=D[m.s,m.d,x.e,x.f]:UsedLink[m.s,m.d,j.a,j.b,k,x.e,x.f])<= LA[j.a,j.b]*MF[j.a,j.b]-Fj[j.a,j.b];

Demands[m,x] WHERE D[m.s,m.d,x.e,x.f]>0:
SUM(k<=D[m.s,m.d,x.e,x.f]:Paths[m,k,x])=D[m.s,m.d,x.e,x.f];

Simetry[s,d IN m,a,b IN j,k,e,f IN x] WHERE k<=D[s,d,e,f]:
UsedLink[s,d,a,b,k,e,f]=UsedLink[s:=d,d:=s,a:=b,b:=a,k,e:=f,f:=e];

NoCirculation1[s,d IN m,e,f IN x]:
SUM(b,k<=D[s,d,e,f]:UsedLink[s,d,a:=d,b,k,e,f]) + SUM(a,k:UsedLink[s,d,a,b:=s,k,e,f])=0;

NoCirculation2[s,d IN m,a,b IN j,k,e,f IN x] WHERE k<=D[s,d,e,f]:
UsedLink[s,d,a:=b,b:=a,k,e,f]+UsedLink[s,d,a,b,k,e,f]<=1;

HopDistance[m,k,x] WHERE k<=D[m.s,m.d,x.e,x.f]:
HD[m.s,m.d,k,x.e,x.f]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k,x.e,x.f]);
```

9.3.4. ReroutingPR_vwp

```
TITLE
ReroutingPR_vwp1

INDEX
n:=DATABASE("qdfReducedNodes","Node");
```

```
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b]:=DATABASE("qdfAvailableChannels","ASCj", a="a", b="b");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfSortedResRoutingWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k" WHERE k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
dm[e,f,m]=DATABASE("InputDm", e="a", f="b", m="m");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmp[e,f IN x,m,p] WHERE m<=A[e,f] AND p<=Px[e,f,m] EXPORT REFILL TO DATABASE("endReroutingFxmp","Fxmp", e="e", f="f", m="m", p="p");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SC[j.a,j.b]) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m]:Fxmp[x.e,x.f,m,p])=dm[x.e,x.f,m];

EnoughSpareCapacitiy[x,j] WHERE j<>x:
SC[j.a,j.b]>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]: delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmp[x.e, x.f, m, p]);

SpareCapChannels[j]:
SC[j.a,j.b]-ASC[j.a,j.b]<=LA[j.a,j.b]*SF[j.a,j.b];

Simetry[e,f IN x,p,m] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmp[e,f,m,p]=Fxmp[e:=f,f:=e,m,p];

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[j]:
UF[j.a,j.b] +SF[j.a,j.b]<=MF[j.a,j.b];
```

9.3.5. ReroutingPR_vwpFREE

```
TITLE
ReroutingPR_vwp1free

INDEX
```

```
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b]:=DATABASE("qdfAvailableChannels","ASCj", a="a", b="b");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfSortedResRoutingWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k" WHERE k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
dm[e,f,m]=DATABASE("InputDm", e="a", f="b", m="m");
delta_jxm[a,b IN j,e,f IN x,m]=DATABASE("qdfLinkInPathM","deltaIM", a="a", b="b", e="e", f="f", m="m");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmp[e,f IN x,m,p] WHERE m<=A[e,f] AND p<=Px[e,f,m] EXPORT REFILL TO DATABASE("endReroutingFxmp","Fxmp", e="e", f="f", m="m", p="p");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL
MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SC[j.a,j.b]) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m]:Fxmp[x.e,x.f,m,p])=dm[x.e,x.f,m];

EnoughSpareCapacity[j,x] WHERE j<>x:
SC[j.a,j.b] + SUM(m<=A[x.e,x.f]:delta_jxm[j.a,j.b,x.e,x.f,m]*dm[x.e,x.f,m])>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]: delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmp[x.e, x.f, m, p]);

SpareCapChannels[j]:
SC[j.a,j.b]-ASC[j.a,j.b]<=LA[j.a,j.b]*SF[j.a,j.b];

Simetry[e,f IN x,p,m] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmp[e,f,m,p]=Fxmp[e:=f,f:=e,m,p];

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[j]:
UF[j.a,j.b] +SF[j.a,j.b]<=MF[j.a,j.b];
```

9.4. Path Rerouting WP

9.4.1. PreDim

TITLE
Perprocesor_For_Dimensioning

INDEX
node:=DATABASE("dataNodes","Node");
s=node;
d=node;
a=node;
b=node;
k=1..100;
j[a,b]:=DATABASE("dataPhysical", a=a, b=b);
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");

DATA

P[a,b]:=DATABASE("dataPhysical", "Pj", a="a", b="b");
D[s,d]:=DATABASE("dataTraffic", "dm", s="s",d="d");
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a",b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");

BINARY VARIABLES
UsedLink[s,d IN m,a,b IN j,k] WHERE k<=D[s,d] EXPORT REFILL TO DATABASE("TrafficRouting","deltaJPM", s="s", d="d", a="a", b="b", k="k");
Paths[m,k] WHERE D[m.s,m.d]>0 AND k<=D[m.s,m.d];

INTEGER VARIABLES
Fmax;
HD[s,d IN m,k] WHERE k<=D[s,d] EXPORT TO DATABASE("TrafficRouting","HD", s="s", d="d", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,k:UsedLink[m.s,m.d,a,b,k]);

FlowBal[m,node,k] WHERE k<=D[m.s,m.d]:
Paths[m,k] IF (m.s=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.b=node)=
Paths[m,k] IF (m.d=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.a=node);

Demands[m] WHERE D[m.s,m.d]>0:
SUM(k<=D[m.s,m.d]:Paths[m,k])=D[m.s,m.d];

CapConst[j] WHERE LA[j.a,j.b]>0:
SUM(m,k<=D[m.s,m.d]:UsedLink[m.s,m.d,j.a,j.b,k])<= MF[j.a,j.b]*LA[j.a,j.b];

Simetry[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a,b,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,k];

NoCirculation1[s,d IN m]:
SUM(b,k<=D[s,d]:UsedLink[s,d,a:=d,b,k]) + SUM(a,k<=D[s,d]:UsedLink[s,d,a,b:=s,k])=0;

NoCirculation2[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a:=b,b:=a,k]+UsedLink[s,d,a,b,k]<=1;

HopDistance[m,k] WHERE k<=D[m.s,m.d]:
HD[m.s,m.d,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k]);

9.4.2. DimensioningWP

TITLE
DimensionigWP1k

INDEX
```
n:=DATABASE("dataNodes","Node");
s=n;
d=n;
a=n;
b=n;
j[a,b] = DATABASE("dataPhysical", a="a",b="b");
m[s,d] = DATABASE("dataTraffic", s="s",d="d");
u[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a<b);
v[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a>b);
lambda:=DATABASE("SQLlambda", lambda="ind");
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
```

DATA
```
MaxP=Last(p);
Alfa[a,b]= DATABASE("dataPhysical", a="a", b="b");
Beta[a,b]= DATABASE("dataPhysical", a="a", b="b");
Gama[a,b]= DATABASE("dataPhysical", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a", b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
D[s,d]:=DATABASE("dataTraffic","dm", s="s", d="d");
km[s,d]:=DATABASE("KSHnumber","km", s="s", d="d");
deltaJPM[s,d,a,b,p]:=DATABASE("SortedTraffic", s="s", d="d", a="a", b="b", p="k" WHERE k<=MaxP);
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
MinNodeDegree[n]=DATABASE("dataNodes","MinNodeDegree");
```

BINARY VARIABLES
```
deltaJ[a,b IN j] EXPORT REFILL TO DATABASE("outUsedLinks","deltaJ", a="a", b="b");
deltaNI[n,i];
```

INTEGER VARIABLES
```
UF[a,b IN j] EXPORT TO DATABASE ("outUsedLinks","Fibers", a="a", b="b");
UC[a,b IN j,lambda] WHERE lambda<=LA[a,b] EXPORT REFILL TO DATABASE("outFlowPerLambda","Fjl", a="a", b="b", lambda="lambda");
allUC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","Fj", a="a", b="b");
Fmpl[p,s,d IN m,lambda] WHERE p<=km[s,d] EXPORT REFILL TO DATABASE("outFlowOnRouteP","Fpm", p="p", s="s", d="d", lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","Type", n="Node");
```

MODEL

```
MIN TotNetCost=SUM(j: Alfa[j.a,j.b]*deltaJ[j.a,j.b] + Beta[j.a,j.b]*UF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:UC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])
```

SUBJECT TO

```
DemandConst[m]:
SUM(p,lambda:Fmpl[p,m.s,m.d,lambda])=D[m.s,m.d];

FlowOnLink[j,lambda] WHERE lambda<=LA[j.a,j.b]:
UC[j.a,j.b,lambda]>=SUM(m,p: deltaJPM[m.s,m.d,j.a,j.b,p]*Fmpl[p,m.s,m.d,lambda]);

ChannelsInFiber[a,b,lambda] WHERE lambda<=LA[a,b] EXPORT REFILL Slack TO DATABASE("outChannelsInFiber", "ASCj", a="a", b="b", lambda="lambda"):
UC[a,b,lambda]<=UF[a,b];

UsedLink[j]:
UF[j.a,j.b]<=MF[j.a,j.b]*deltaJ[j.a,j.b];

NodeDegree1[n]:
Sum(u:deltaJN[u.a,u.b,n]*deltaJ[u.a,u.b])>=MinNodeDegree[n];

NodeDegree2[n]:
Sum(v:deltaJN[v.a,v.b,n]*deltaJ[v.a,v.b])>=MinNodeDegree[n];

NodeSize[n]:
Sum(i:deltaNI[n,i])=1;
```

```
SizeVSLinks[n]:
SUM(j:deltaJN[j.a,j.b,n]*UF[j.a,j.b])<=2*Sum(i: K[i]*deltaNI[n,i]);

MergeFlowOnLInk[a,b IN j]:
SUM(lambda:UC[a,b,lambda])=allUC[a,b];

Symetry1[a,b IN j,lambda]:
UC[a,b,lambda]=UC[a:=b,b:=a,lambda];

Symetry2[p,s,d IN m,lambda]:
Fmpl[p,s,d,lambda]=Fmpl[p,s:=d,d:=s,lambda];

LeastFiber[j]:
UF[j.a,j.b]>=deltaJ[j.a,j.b];

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.4.3. PreResPRwp

```
TITLE
PreResPRwp1

INDEX
node:=DATABASE("qdfReducedNodes","Node");
s=node;
d=node;
a=node;
b=node;
e=node;
f=node;
k=1..100;
j[a,b]:=DATABASE("qdfReducedLinks", a="a", b="b");
x[e,f]:=DATABASE("qdfReducedLinks", e="a", f="b");
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");
lambda:=DATABASE("SQLlambda", lambda="ind");

DATA
D[s,d,e,f,lambda]:=DATABASE("InputDm", "dm", s="s", d="d", e="a", f="b",lambda="lambda");
sumD[s,d,e,f]=SUM(lambda:D[s,d,e,f,lambda]);
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a",b="b");
Fj[a,b]:=DATABASE("outUsedLinks", "Fj", a="a", b="b");
MF[a,b]:=DATABASE("qdfReducedLinks", "MFj", a="a",b="b");
X[a,b IN j,e,f IN x]:=IF(a=e AND b=f )Then 1 ENDIF;

BINARY VARIABLES
UsedLink[s,d IN m, a,b IN j,k,e,f IN x] WHERE k<=sumD[s,d,e,f] AND X[a,b,e,f]=0
EXPORT REFILL TO DATABASE("RestorationRouting","delta", s="s", d="d", a="a", b="b", k="k", e="e",f="f");
Paths[m,k,x] WHERE sumD[m.s,m.d,x.e,x.f]>0 AND k<=sumD[m.s,m.d,x.e,x.f];

INTEGER VARIABLES
Fmax[e,f IN x];
HD[s,d IN m,k,e,f IN x] WHERE k<=sumD[s,d,e,f] EXPORT TO DATABASE("RestorationRouting","HD", s="s", d="d", k="k", e="e",f="f");

MODEL
MIN SUM(x:Fmax[x.e,x.f]);

SUBJECT TO
MaximumFlow[a,b IN j,e,f IN x]:
Fmax[e,f]>=SUM(m, k<=sumD[m.s,m.d,e,f]:UsedLink[m.s,m.d,a,b,k,e,f]);

FlowBal[m,x,node,k] WHERE k<=sumD[m.s,m.d,x.e,x.f]:
Paths[m,k,x] IF (m.s=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k,x.e,x.f] WHERE j.b=node)=
Paths[m,k,x] IF (m.d=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k,x.e,x.f] WHERE j.a=node);

CapConst[j,x] WHERE LA[j.a,j.b]>0:
SUM(m,k:UsedLink[m.s,m.d,j.a,j.b,k,x.e,x.f])<= LA[j.a,j.b]*MF[j.a,j.b]-Fj[j.a,j.b];

Demands[m,x] WHERE sumD[m.s,m.d,x.e,x.f]>0:
SUM(k<=sumD[m.s,m.d,x.e,x.f]:Paths[m,k,x])=sumD[m.s,m.d,x.e,x.f];

Simetry[s,d IN m,a,b IN j,k,e,f IN x] WHERE k<=sumD[s,d,e,f]:
UsedLink[s,d,a,b,k,e,f]=UsedLink[s:=d,d:=s,a:=b,b:=a,k,e:=f,f:=e];

NoCirculation1[s,d IN m,e,f IN x]:
SUM(b,k<=sumD[s,d,e,f]:UsedLink[s,d,a:=d,b,k,e,f]) + SUM(a,k:UsedLink[s,d,a,b:=s,k,e,f])=0;
```

```
NoCirculation2[s,d IN m,a,b IN j,k,e,f IN x] WHERE k<=sumD[s,d,e,f]:
UsedLink[s,d,a:=b,b:=a,k,e,f]+UsedLink[s,d,a,b,k,e,f]<=1;

HopDistance[m,k,x] WHERE k<=sumD[m.s,m.d,x.e,x.f]:
HD[m.s,m.d,k,x.e,x.f]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k,x.e,x.f]);
```

9.4.4. ReroutingPR_wpA

```
TITLE
ReroutingPR_wpA1

INDEX
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
lambda:=DATABASE("SQLlambda", lambda="ind");

DATA
MaxP=LAST(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b,lambda]:=DATABASE("outChannelsInFiber","ASCj", a="a", b="b", lambda="lambda");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfSortedResRoutingWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k" WHERE
k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
dm[e,f,m,lambda]=DATABASE("InputDm", e="a", f="b", m="m", lambda="lambda");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j,lambda];
allSC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmpl[e,f IN x,m,p,lambda] WHERE p<=Px[e,f,m] AND m<=A[e,f]
EXPORT REFILL TO DATABASE("endReroutingFxmp",Fxmpl="Fxmp", e="e", f="f", m="m", p="p",lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:SC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m] WHERE m<=A[x.e,x.f]:
SUM(p,lambda:Fxmpl[x.e,x.f,m,p,lambda])=SUM(lambda:dm[x.e,x.f,m,lambda]);

EnoughSpareCapacitiy[j,x,lambda] WHERE j<>x:
SC[j.a,j.b]>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]: delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmpl[x.e, x.f, m, p,lambda]);

SpareCapChannels[j,lambda]:
SC[j.a,j.b,lambda]-ASC[j.a,j.b,lambda]<=SF[j.a,j.b];

Simetry[e,f IN x,p,m,lambda] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmpl[e,f,m,p,lambda]=Fxmpl[e:=f,f:=e,m,p,lambda];

MergeFlowOnSpareLinks[j]:
allSC[j.a,j.b]=SUM(lambda:SC[j.a,j.b,lambda]);
```

```
NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[j]:
UF[j.a,j.b] +SF[j.a,j.b]<=MF[j.a,j.b];
```

9.4.5. ReroutingPR_wpAfree

```
TITLE
ReroutingPR_wpA1free

INDEX
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
lambda:=DATABASE("SQLlambda", lambda="ind");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b,lambda]:=DATABASE("outChannelsInFiber","ASCj", a="a", b="b", lambda="lambda");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfSortedResRoutingWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k", WHERE
k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
dm[e,f,m,lambda]=DATABASE("InputDm", e="a", f="b", m="m", lambda="lambda");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j,lambda];
allSC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmpl[e,f IN x,m,p,lambda] WHERE p<=Px[e,f,m] AND m<=A[e,f]
EXPORT REFILL TO DATABASE("endReroutingFxmp",Fxmpl="Fxmp", e="e", f="f", m="m", p="p",lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:SC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m],lambda:Fxmpl[x.e,x.f,m,p,lambda])=SUM(lambda:dm[x.e,x.f,m,lambda]);

EnoughSpareCapacitiy[j,x,lambda] WHERE j<>x:
SC[j.a,j.b] + SUM(m<=A[x.e,x.f]:delta_jxm[j.a,j.b,x.e,x.f,m,lambda]*dm[x.e,x.f,m,lambda])>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]:
delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmpl[x.e, x.f, m, p,lambda]);

SpareCapChannels[j,lambda]:
SC[j.a,j.b,lambda]-ASC[j.a,j.b,lambda]<=SF[j.a,j.b];
```

```
Simetry[e,f IN x,p,m,lambda] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmpl[e,f,m,p,lambda]=Fxmpl[e:=f,f:=e,m,p,lambda];

MergeFlowOnSpareLinks[j]:
allSC[j.a,j.b]=SUM(lambda:SC[j.a,j.b,lambda]);

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[j]:
UF[j.a,j.b] +SF[j.a,j.b]<=MF[j.a,j.b];
```

9.4.6. ReroutingPR_wpB

```
TITLE
ReroutingPR_wpB1

INDEX
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p=DATABASE("dataNumberOfRoutes","NbRoutes");
lambda:=DATABASE("SQLlambda", lambda="ind");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b,lambda]:=DATABASE("outChannelsInFiber","ASCj", a="a", b="b", lambda="lambda");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfSortedResRoutingWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k", WHERE
k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
dm[e,f,m,lambda]=DATABASE("InputDm", e="a", f="b", m="m", lambda="lambda");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];
```

```
INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j,lambda];
allSC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmpl[e,f IN x,m,p,lambda] WHERE p<=Px[e,f,m] AND m<=A[e,f]
EXPORT REFILL TO DATABASE("endReroutingFxmp",Fxmpl="Fxmp", e="e", f="f", m="m", p="p",lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:SC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m,lambda] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m]:Fxmpl[x.e,x.f,m,p,lambda])=dm[x.e,x.f,m,lambda];

EnoughSpareCapacitiy[j,x,lambda] WHERE j<>x:
SC[j.a,j.b]>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]: delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmpl[x.e, x.f, m, p,lambda]);

SpareCapChannels[j,lambda]:
SC[j.a,j.b,lambda]-ASC[j.a,j.b,lambda]<=SF[j.a,j.b];

Simetry[e,f IN x,p,m,lambda] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmpl[e,f,m,p,lambda]=Fxmpl[e:=f,f:=e,m,p,lambda];

MergeFlowOnSpareLinks[j]:
allSC[j.a,j.b]=SUM(lambda:SC[j.a,j.b,lambda]);

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[j]:
UF[j.a,j.b] +SF[j.a,j.b]<=MF[j.a,j.b];
```

9.4.7. ReroutingPR_wpBfree

```
TITLE
ReroutingPR_wpB1free

INDEX
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
lambda:=DATABASE("SQLlambda", lambda="ind");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b,lambda]:=DATABASE("outChannelsInFiber","ASCj", a="a", b="b", lambda="lambda");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfSortedResRoutingWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k", WHERE
k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM","Px", e="e", f="f", m="m");
dm[e,f,m,lambda]=DATABASE("InputDm", e="a", f="b", m="m", lambda="lambda");
delta_jxm[a,b IN j,e,f IN x,m,lambda]=DATABASE("qdfLinkInPathM","deltaIM", a="a", b="b", e="e", f="f", m="m", lambda="lambda");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
```

```
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j,lambda];
allSC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmpl[e,f IN x,m,p,lambda] WHERE p<=Px[e,f,m] AND m<=A[e,f]
EXPORT REFILL TO DATABASE("endReroutingFxmp",Fxmpl="Fxmp", e="e", f="f", m="m", p="p",lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:SC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m,lambda] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m]:Fxmpl[x.e,x.f,m,p,lambda])=dm[x.e,x.f,m,lambda];

EnoughSpareCapacitiy[j,x,lambda] WHERE j<>x:
SC[j.a,j.b] + SUM(m<=A[x.e,x.f]:delta_jxm[j.a,j.b,x.e,x.f,m,lambda]*dm[x.e,x.f,m,lambda])>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]:
delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmpl[x.e, x.f, m, p,lambda]);

SpareCapChannels[j,lambda]:
SC[j.a,j.b,lambda]-ASC[j.a,j.b,lambda]<=SF[j.a,j.b];

Simetry[e,f IN x,p,m,lambda] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmpl[e,f,m,p,lambda]=Fxmpl[e:=f,f:=e,m,p,lambda];

MergeFlowOnSpareLinks[j]:
allSC[j.a,j.b]=SUM(lambda:SC[j.a,j.b,lambda]);

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[j]:
UF[j.a,j.b] +SF[j.a,j.b]<=MF[j.a,j.b];
```

9.5. Path Rerouting Disjunct VWP

9.5.1. PreDim

TITLE
Perprocesor_For_Dimensioning

INDEX
node:=DATABASE("dataNodes","Node")
s=node;
d=node;
a=node;
b=node;
k=1..100;
j[a,b]:=DATABASE("dataPhysical", a=a, b=b);
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");

DATA

P[a,b]:=DATABASE("dataPhysical", "Pj", a="a", b="b");
D[s,d]:=DATABASE("dataTraffic", "dm", s="s",d="d");
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a",b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");

BINARY VARIABLES
UsedLink[s,d IN m,a,b IN j,k] WHERE k<=D[s,d] EXPORT REFILL TO DATABASE("TrafficRouting","deltaJPM", s="s", d="d", a="a", b="b", k="k");
Paths[m,k] WHERE D[m.s,m.d]>0 AND k<=D[m.s,m.d];

INTEGER VARIABLES
Fmax;
HD[s,d IN m,k] WHERE k<=D[s,d] EXPORT TO DATABASE("TrafficRouting","HD", s="s", d="d", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,k:UsedLink[m.s,m.d,a,b,k]);

FlowBal[m,node,k] WHERE k<=D[m.s,m.d]:
Paths[m,k] IF (m.s=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.b=node)=
Paths[m,k] IF (m.d=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.a=node);

Demands[m] WHERE D[m.s,m.d]>0:
SUM(k<=D[m.s,m.d]:Paths[m,k])=D[m.s,m.d];

CapConst[j] WHERE LA[j.a,j.b]>0:
SUM(m,k<=D[m.s,m.d]:UsedLink[m.s,m.d,j.a,j.b,k])<= MF[j.a,j.b]*LA[j.a,j.b];

Simetry[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a,b,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,k];

NoCirculation1[s,d IN m]:
SUM(b,k<=D[s,d]:UsedLink[s,d,a:=d,b,k]) + SUM(a,k<=D[s,d]:UsedLink[s,d,a,b:=s,k])=0;

NoCirculation2[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a:=b,b:=a,k]+UsedLink[s,d,a,b,k]<=1;

HopDistance[m,k] WHERE k<=D[m.s,m.d]:
HD[m.s,m.d,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k]);

9.5.2. DimensioningVWP

TITLE
DimensioningVWP1k

```
INDEX
n:=DATABASE("dataNodes","Node");
s=n;
d=n;
a=n;
b=n;
j[a,b] = DATABASE("dataPhysical", a="a",b="b");
m[s,d] = DATABASE("dataTraffic", s="s",d="d");
u[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a<b);
v[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a>b);
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");

DATA
MaxP=Last(p);
Alfa[a,b]= DATABASE("dataPhysical", a="a", b="b");
Beta[a,b]= DATABASE("dataPhysical", a="a", b="b");
Gama[a,b]= DATABASE("dataPhysical", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a", b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
D[s,d]:=DATABASE("dataTraffic","dm", s="s", d="d");
km[s,d]:=DATABASE("KSHnumber","km", s="s", d="d");
deltaJPM[s,d,a,b,p]:=DATABASE("SortedTraffic", s="s", d="d", a="a", b="b", p="k" WHERE k<=MaxP);
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
MinNodeDegree[n]=DATABASE("dataNodes","MinNodeDegree");

BINARY VARIABLES
deltaJ[a,b IN j] EXPORT REFILL TO DATABASE("outUsedLinks","deltaJ", a="a", b="b");
deltaNI[n,i]; !1 if node n is of node type i, 0 otherwise n:1..N);

INTEGER VARIABLES
UF[a,b IN j] EXPORT TO DATABASE ("outUsedLinks","Fibers", a="a", b="b");
UC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","Fj", a="a", b="b");
Fpm[p,s,d IN m] WHERE p<=km[s,d]
EXPORT REFILL TO DATABASE("outFlowOnRouteP","Fpm", p="p", s="s", d="d");
NodeType[n] EXPORT TO DATABASE("dataNodes","Type", n="Node");

MODEL

MIN TotNetCost=SUM(j: Alfa[j.a,j.b]*deltaJ[j.a,j.b] + Beta[j.a,j.b]*UF[j.a,j.b]  + Gama[j.a,j.b]*UC[j.a,j.b]) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

DemandConst[s,d IN m]:
SUM(p:Fpm[p,s,d])=D[s,d];

FlowOnLink[j]:
UC[j.a,j.b]>=SUM(m,p: deltaJPM[m.s,m.d,j.a,j.b,p]*Fpm[p,m.s,m.d]);

ChannelsInFiber[a,b]:
UC[a,b]<=LA[a,b]*UF[a,b];

UsedLink[j]:
UF[j.a,j.b]<=MF[j.a,j.b]*deltaJ[j.a,j.b];

NodeDegree1[n]:
Sum(u:deltaJN[u.a,u.b,n]*deltaJ[u.a,u.b])>=MinNodeDegree[n];

NodeDegree2[n]:
Sum(v:deltaJN[v.a,v.b,n]*deltaJ[v.a,v.b])>=MinNodeDegree[n];

NodeSize[n]:
Sum(i:deltaNI[n,i])=1;

SizeVSLinks[n]:
SUM(j:deltaJN[j.a,j.b,n]*UF[j.a,j.b])<=2*Sum(i: K[i]*deltaNI[n,i]);

Simetry1[a,b IN j]:
UC[a,b]=UC[a:=b,b:=a];

Simetry2[p,s,d IN m]:
Fpm[p,s,d]=Fpm[p,s:=d,d:=s];

LeastFiber[j]:
UF[j.a,j.b]>=deltaJ[j.a,j.b];

TypeOfNode[n]:
```

```
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.5.3. PreResPRd_vwp

```
TITLE
PreResPRd_vwp1

INDEX
node:=DATABASE("qdfReducedNodes","Node");
s=node;
d=node;
a=node;
b=node;
k=1..100;
j[a,b]:=DATABASE("qdfReducedLinks", a=a, b=b);
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");
p=1..100;

DATA

D[s,d]:=DATABASE("dataTraffic", "dm", s="s",d="d");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a",b="b");
MF[a,b]:=DATABASE("qdfReducedLinks", "MFj", a="a",b="b");
InitRoute[s,d,a,b,p]:=DATABASE("qdfLinkInRoute", "deltaIM", s="s", d="d", a="a", b="b", p="p");
MyP[s,d,p]:=DATABASE("qdfLinkInRoute", MyP="deltaIM", s="s", d="d", p="p");
Fmp[s,d,p]:=DATABASE("qdfLinkInRoute", "Fpm", s="s", d="d", p="p");
Fj[a,b]:=DATABASE("outUsedLinks", "Fj", a="a", b="b");
AllP[s,d,a,b]:=DATABASE("qdfLinkInRoute", AllP="p", s="s", d="d", a="a", b="b");
MaxP=MAX(AllP);


BINARY VARIABLES
UsedLink[s,d IN m, a,b IN j,p,k] WHERE k<=D[s,d] AND MyP[s,d,p]>0 AND p<=MaxP;
EXPORT REFILL TO DATABASE("RestorationRouting","delta", s="s", d="d", a="a", b="b", k="k", p="p");
Paths[m,p,k] WHERE D[m.s,m.d]>0 AND k<=Fmp[m.s,m.d,p] AND MyP[m.s,m.d,p]>0 AND p<=MaxP;

INTEGER VARIABLES
Fmax;
HD[s,d IN m,p,k] WHERE k<=Fmp[s,d,p] AND p<=MaxP EXPORT TO DATABASE("RestorationRouting","HD", s="s", d="d", p="p", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,p<=MaxP,k<=Fmp[m.s,m.d,p]:UsedLink[m.s,m.d,a,b,p,k]);

FlowBal[m,node,k,p] WHERE k<=Fmp[m.s,m.d,p] AND MyP[m.s,m.d,p]>0 AND p<=MaxP:
Paths[m,p,k] IF (m.s=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,p,k] WHERE j.b=node)=
Paths[m,p,k] IF (m.d=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,p,k] WHERE j.a=node);

CapConst[j] WHERE LA[j.a,j.b]>0:
SUM(m, p<=MaxP,k<=Fmp[m.s,m.d,p]:UsedLink[m.s,m.d,j.a,j.b,p,k])<= LA[j.a,j.b]*MF[j.a,j.b]-Fj[j.a,j.b];

Demands[m,p] WHERE D[m.s,m.d]>0 AND p<=MaxP:
SUM(k<=Fmp[m.s,m.d,p]:Paths[m,p,k])=Fmp[m.s,m.d,p];

Simetry[s,d IN m,a,b IN j,p,k] WHERE k<=Fmp[s,d,p] AND MyP[s,d,p]>0 AND p<=MaxP:
UsedLink[s,d,a,b,p,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,p,k];

Diversity[s,d IN m, a,b IN j,p,k] WHERE k<=Fmp[s,d,p] AND MyP[s,d,p]>0 AND p<=MaxP:
UsedLink[s,d,a,b,p,k]+InitRoute[s,d,a,b,p]<=1;

NoCirculation1[s,d IN m,p] WHERE p<=MaxP:
SUM(b,k<=Fmp[s,d,p]:UsedLink[s,d,a:=d,b,p,k]) + SUM(a,k<=Fmp[s,d,p]:UsedLink[s,d,a,b:=s,p,k])=0;

NoCirculation2[s,d IN m,a,b IN j,p,k] WHERE k<=Fmp[s,d,p] AND p<=MaxP:
UsedLink[s,d,a:=b,b:=a,p,k]+UsedLink[s,d,a,b,p,k]<=1;

HopDistance[m,p,k] WHERE k<=Fmp[m.s,m.d,p] AND p<=MaxP:
HD[m.s,m.d,p,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,p,k]);
```

9.5.4. ReroutingPRd_vwp

TITLE
ReroutingPRd_vwp1

INDEX
```
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
```

DATA
```
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b]:=DATABASE("qdfAvailableChannels","ASCj", a="a", b="b");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM_prd","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j, e,f IN x,m,p]:=DATABASE("qdfPRdSortResWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k" WHERE k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
dm[e,f,m]=DATABASE("InputDm", e="a", f="b", m="m");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);
```

BINARY VARIABLES
```
deltaNI[n,i];
```

INTEGER VARIABLES
```
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmp[e,f,m,p] WHERE m<=A[e,f] AND p<=Px[e,f,m] EXPORT REFILL TO DATABASE("endReroutingFxmp","Fxmp", e="e", f="f", m="m",
p="p");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");
```

MODEL

```
MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SC[j.a,j.b]) +SUM(n,i: C[i]*deltaNI[n,i])
```

SUBJECT TO

```
FlowOverPRestorationRoutes[x,m] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m]:Fxmp[x.e,x.f,m,p])=dm[x.e,x.f,m];

EnoughSpareCapacitiy[j,x] WHERE j<>x:
SC[j.a,j.b]>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]: delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmp[x.e, x.f, m, p]);

SpareCapChannels[j]:
SC[j.a,j.b]-ASC[j.a,j.b]<=LA[j.a,j.b]*SF[j.a,j.b];

Simetry[e,f IN x,p,m] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmp[e,f,m,p]=Fxmp[e:=f,f:=e,m,p];

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[j]:
UF[j.a,j.b] +SF[j.a,j.b]<=MF[j.a,j.b];
```

9.5.5. ReroutingPRd_vwpFREE

```
TITLE
ReroutingPRd_vwp1free

INDEX
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4); !indeks = mozniot broj na razlicni tipovi na crossconnect-i (i:1..I)
p =DATABASE("dataNumberOfRoutes","NbRoutes");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b]:=DATABASE("qdfAvailableChannels","ASCj", a="a", b="b");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM_prd","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j, e,f IN x,m,p]:=DATABASE("qdfPRdSortResWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k" WHERE k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
dm[e,f,m]=DATABASE("InputDm", e="a", f="b", m="m");
delta_jxm[a,b IN j,e,f IN x,m]=DATABASE("qdfLinkInPathM","deltaIM", a="a", b="b", e="e", f="f", m="m");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmp[e,f,m,p] WHERE m<=A[e,f] AND p<=Px[e,f,m] EXPORT REFILL TO DATABASE("endReroutingFxmp","Fxmp", e="e", f="f", m="m",
p="p");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SC[j.a,j.b]) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m] WHERE m<=A[x.e,x.f]:
SUM(p:Fxmp[x.e,x.f,m,p])=dm[x.e,x.f,m];

EnoughSpareCapacitiy[j,x] WHERE j<>x:
SC[j.a,j.b] + SUM(m<=A[x.e,x.f]:delta_jxm[j.a,j.b,x.e,x.f,m]*dm[x.e,x.f,m])>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]: delta_jxmp[j.a, j.b, x.e, x.f, m,
p]*Fxmp[x.e, x.f, m, p]);

SpareCapChannels[j]:
SC[j.a,j.b]-ASC[j.a,j.b]<=LA[j.a,j.b]*SF[j.a,j.b];

Simetry[e,f IN x,p,m] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmp[e,f,m,p]=Fxmp[e:=f,f:=e,m,p];

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);

TotalNumberOfFibers[j]:
UF[j.a,j.b] +SF[j.a,j.b]<=MF[j.a,j.b];
```

9.6. Path Rerouting Disjunct WP

9.6.1. PreDim

TITLE
Perprocesor_For_Dimensioning

INDEX
node:=DATABASE("dataNodes","Node")
s=node;
d=node;
a=node;
b=node;
k=1..100;
j[a,b]:=DATABASE("dataPhysical", a=a, b=b);
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");

DATA

P[a,b]:=DATABASE("dataPhysical", "Pj", a="a", b="b");
D[s,d]:=DATABASE("dataTraffic", "dm", s="s",d="d");
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a",b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");

BINARY VARIABLES
UsedLink[s,d IN m,a,b IN j,k] WHERE k<=D[s,d] EXPORT REFILL TO DATABASE("TrafficRouting","deltaJPM", s="s", d="d", a="a", b="b", k="k");
Paths[m,k] WHERE D[m.s,m.d]>0 AND k<=D[m.s,m.d];

INTEGER VARIABLES
Fmax;
HD[s,d IN m,k] WHERE k<=D[s,d] EXPORT TO DATABASE("TrafficRouting","HD", s="s", d="d", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,k:UsedLink[m.s,m.d,a,b,k]);

FlowBal[m,node,k] WHERE k<=D[m.s,m.d]:
Paths[m,k] IF (m.s=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.b=node)=
Paths[m,k] IF (m.d=node) + SUM(j: UsedLink[m.s,m.d,j.a,j.b,k] WHERE j.a=node);

Demands[m] WHERE D[m.s,m.d]>0:
SUM(k<=D[m.s,m.d]:Paths[m,k])=D[m.s,m.d];

CapConst[j] WHERE LA[j.a,j.b]>0:
SUM(m,k<=D[m.s,m.d]:UsedLink[m.s,m.d,j.a,j.b,k])<= MF[j.a,j.b]*LA[j.a,j.b];

Simetry[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a,b,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,k];

NoCirculation1[s,d IN m]:
SUM(b,k<=D[s,d]:UsedLink[s,d,a:=d,b,k]) + SUM(a,k<=D[s,d]:UsedLink[s,d,a,b:=s,k])=0;

NoCirculation2[s,d IN m,a,b IN j,k] WHERE k<=D[s,d]:
UsedLink[s,d,a:=b,b:=a,k]+UsedLink[s,d,a,b,k]<=1;

HopDistance[m,k] WHERE k<=D[m.s,m.d]:
HD[m.s,m.d,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,k]);

9.6.2. DimensionigWP

TITLE
DimensionigWP1k

INDEX
```
n:=DATABASE("dataNodes","Node");
s=n;
d=n;
a=n;
b=n;
j[a,b] = DATABASE("dataPhysical", a="a",b="b");
m[s,d] = DATABASE("dataTraffic", s="s",d="d");
u[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a<b);
v[a,b]= DATABASE("dataPhysical", a="a",b="b" WHERE a>b);
lambda:=DATABASE("SQLlambda", lambda="ind");
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
```

DATA
```
MaxP=Last(p);
Alfa[a,b]= DATABASE("dataPhysical", a="a", b="b");
Beta[a,b]= DATABASE("dataPhysical", a="a", b="b");
Gama[a,b]= DATABASE("dataPhysical", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);
LA[a,b]:=DATABASE("dataPhysical", "LAj", a="a", b="b");
MF[a,b]:=DATABASE("dataPhysical", "MFj", a="a",b="b");
D[s,d]:=DATABASE("dataTraffic","dm", s="s", d="d");
km[s,d]:=DATABASE("KSHnumber","km", s="s", d="d");
deltaJPM[s,d,a,b,p]:=DATABASE("SortedTraffic", s="s", d="d", a="a", b="b", p="k" WHERE k<=MaxP);
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
MinNodeDegree[n]=DATABASE("dataNodes","MinNodeDegree");
```

BINARY VARIABLES
```
deltaJ[a,b IN j] EXPORT REFILL TO DATABASE("outUsedLinks","deltaJ", a="a", b="b");
deltaNI[n,i];
```

INTEGER VARIABLES
```
UF[a,b IN j] EXPORT TO DATABASE ("outUsedLinks","Fibers", a="a", b="b");
UC[a,b IN j,lambda] WHERE lambda<=LA[a,b] EXPORT REFILL TO DATABASE("outFlowPerLambda","Fjl", a="a", b="b", lambda="lambda");
allUC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","Fj", a="a", b="b");
Fmpl[p,s,d IN m,lambda] WHERE p<=km[s,d] EXPORT REFILL TO DATABASE("outFlowOnRouteP","Fpm", p="p", s="s", d="d",
lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","Type", n="Node");
```

MODEL

```
MIN TotNetCost=SUM(j: Alfa[j.a,j.b]*deltaJ[j.a,j.b] + Beta[j.a,j.b]*UF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:UC[j.a,j.b,lambda])) +SUM(n,i:
C[i]*deltaNI[n,i])
```

SUBJECT TO

```
DemandConst[m]:
SUM(p,lambda:Fmpl[p,m.s,m.d,lambda])=D[m.s,m.d];

FlowOnLink[j,lambda] WHERE lambda<=LA[j.a,j.b]:
UC[j.a,j.b,lambda]>=SUM(m,p: deltaJPM[m.s,m.d,j.a,j.b,p]*Fmpl[p,m.s,m.d,lambda]);

ChannelsInFiber[a,b,lambda] WHERE lambda<=LA[a,b] EXPORT REFILL Slack TO DATABASE("outChannelsInFiber", "ASCj", a="a", b="b",
lambda="lambda"):
UC[a,b,lambda]<=UF[a,b];

UsedLink[j]:
UF[j.a,j.b]<=MF[j.a,j.b]*deltaJ[j.a,j.b];

NodeDegree1[n]:
Sum(u:deltaJN[u.a,u.b,n]*deltaJ[u.a,u.b])>=MinNodeDegree[n];

NodeDegree2[n]:
Sum(v:deltaJN[v.a,v.b,n]*deltaJ[v.a,v.b])>=MinNodeDegree[n];

NodeSize[n]:
Sum(i:deltaNI[n,i])=1;

SizeVSLinks[n]:
SUM(j:deltaJN[j.a,j.b,n]*UF[j.a,j.b])<=2*Sum(i: K[i]*deltaNI[n,i]);

MergeFlowOnLInk[a,b IN j]:
SUM(lambda:UC[a,b,lambda])=allUC[a,b];

Symetry1[a,b IN j,lambda]:
UC[a,b,lambda]=UC[a:=b,b:=a,lambda];
```

```
Symetry2[p,s,d IN m,lambda]:
Fmpl[p,s,d,lambda]=Fmpl[p,s:=d,d:=s,lambda];

LeastFiber[j]:
UF[j.a,j.b]>=deltaJ[j.a,j.b];

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.6.3. PreResPRd_wp

```
TITLE
Perprocesor_For_Disjunct_Rerouting

INDEX
node:=DATABASE("qdfReducedNodes","Node");
s=node;
d=node;
a=node;
b=node;
k=1..100;
j[a,b]:=DATABASE("qdfReducedLinks", a=a, b=b);
m[s,d]:=DATABASE("dataTraffic", s="s",d="d");
p=1..100;
lambda:=DATABASE("SQLlambda", lambda="ind");


DATA
D[s,d]:=DATABASE("dataTraffic", "dm", s="s",d="d");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a",b="b");
MF[a,b]:=DATABASE("qdfReducedLinks", "MFj", a="a",b="b");
Fmp[s,d,p,lambda]:=DATABASE("qdfLinkInRoute", "Fpm", s="s", d="d", p="p",lambda="lambda");
sumFmp[s,d,p]=SPARSE (SUM(lambda:Fmp[s,d,p,lambda]));
Fj[a,b]:=DATABASE("outUsedLinks", "Fj", a="a", b="b");
AllP[s,d,a,b]:=DATABASE("qdfLinkInRoute", AllP="p", s="s", d="d", a="a", b="b");
MaxP=MAX(AllP);
MyPwithLambda[s,d,p,lambda]:=DATABASE("qdfLinkInRoute", MyPwithLambda="deltaIM", s="s", d="d", p="p",lambda="lambda");
MyP[s,d In m,p<=MaxP]:=SPARSE (SUM(lambda:MyPwithLambda[s,d,p,lambda]));
InitRouteWithLambda[s,d,a,b,p,lambda]:=DATABASE("qdfLinkInRoute", "deltaIM", s="s", d="d", a="a", b="b", p="p",lambda="lambda");
InitRouteInt[s,d IN m,a,b IN j,p<=MaxP]=SPARSE (SUM(lambda:InitRouteWithLambda[s,d,a,b,p,lambda]));
InitRoute[s,d IN m,a,b IN j,p<=MaxP]:=SPARSE(IF(InitRouteInt[s,d IN m,a,b IN j,p<=MaxP]>=1 )Then 1 ENDIF);

BINARY VARIABLES
UsedLink[s,d IN m, a,b IN j,p,k] WHERE k<=sumFmp[s,d,p] AND MyP[s,d,p]>0 AND p<=MaxP
EXPORT REFILL TO DATABASE("RestorationRouting","delta", s="s", d="d", a="a", b="b", k="k", p="p");
Paths[m,p,k] WHERE D[m.s,m.d]>0 AND k<=sumFmp[m.s,m.d,p] AND MyP[m.s,m.d,p]>0 AND p<=MaxP;

INTEGER VARIABLES
Fmax;
HD[s,d IN m,p,k] WHERE k<=sumFmp[s,d,p] AND p<=MaxP EXPORT TO DATABASE("RestorationRouting","HD", s="s", d="d", p="p", k="k");

MODEL
MIN Fmax;

SUBJECT TO

MaximumFlow[a,b IN j]:
Fmax>=SUM(m,p<=MaxP,k<=sumFmp[m.s,m.d,p]:UsedLink[m.s,m.d,a,b,p,k]);

FlowBal[m,node,k,p] WHERE k<=sumFmp[m.s,m.d,p] AND MyP[m.s,m.d,p]>0 AND p<=MaxP:
Paths[m,p,k] IF (m.s=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,p,k] WHERE j.b=node)=
Paths[m,p,k] IF (m.d=node)  + SUM(j: UsedLink[m.s,m.d,j.a,j.b,p,k] WHERE j.a=node);

CapConst[j] WHERE LA[j.a,j.b]>0:
SUM(m, p<=MaxP,k:UsedLink[m.s,m.d,j.a,j.b,p,k])<= LA[j.a,j.b]*MF[j.a,j.b]-Fj[j.a,j.b];

Demands[m,p] WHERE D[m.s,m.d]>0 AND p<=MaxP:
SUM(k<=sumFmp[m.s,m.d,p]:Paths[m,p,k])=sumFmp[m.s,m.d,p];

Simetry[s,d IN m,a,b IN j,p,k] WHERE k<=sumFmp[s,d,p] AND MyP[s,d,p]>0 AND p<=MaxP:
UsedLink[s,d,a,b,p,k]=UsedLink[s:=d,d:=s,a:=b,b:=a,p,k];

Diversity[s,d IN m, a,b IN j,p,k] WHERE k<=sumFmp[s,d,p] AND MyP[s,d,p]>0 AND p<=MaxP:
UsedLink[s,d,a,b,p,k]+InitRoute[s,d,a,b,p]<=1;

NoCirculation1[s,d IN m,p] WHERE p<=MaxP:
```

```
SUM(b,k<=sumFmp[s,d,p]:UsedLink[s,d,a:=d,b,p,k]) + SUM(a,k<=sumFmp[s,d,p]:UsedLink[s,d,a,b:=s,p,k])=0;

NoCirculation2[s,d IN m,a,b IN j,p,k] WHERE k<=sumFmp[s,d,p] AND p<=MaxP:
UsedLink[s,d,a:=b,b:=a,p,k]+UsedLink[s,d,a,b,p,k]<=1;

HopDistance[m,p,k] WHERE k<=sumFmp[m.s,m.d,p] AND p<=MaxP:
HD[m.s,m.d,p,k]=SUM(j:UsedLink[m.s,m.d,j.a,j.b,p,k]);
```

9.6.4. ReroutingPRd_wpA

TITLE
ReroutingPRd_wpA1

INDEX
```
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
lambda:=DATABASE("SQLlambda", lambda="ind");
```

DATA
```
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b,lambda]:=DATABASE("outChannelsInFiber","ASCj", a="a", b="b", lambda="lambda");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM_prd","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfPRdSortResWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k" WHERE k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
dm[e,f,m,lambda]=DATABASE("InputDm", e="a", f="b", m="m", lambda="lambda");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);
```

BINARY VARIABLES
```
deltaNI[n,i];
```

INTEGER VARIABLES
```
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j,lambda];
allSC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmpl[e,f IN x,m,p,lambda] WHERE p<=Px[e,f,m] AND m<=A[e,f]
EXPORT REFILL TO DATABASE("endReroutingFxmp",Fxmpl="Fxmp", e="e", f="f", m="m", p="p",lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");
```

MODEL

```
MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:SC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])
```

SUBJECT TO

```
FlowOverPRestorationRoutes[x,m] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m],lambda:Fxmpl[x.e,x.f,m,p,lambda])=SUM(lambda:dm[x.e,x.f,m,lambda]);

EnoughSpareCapacitiy[j,x,lambda] WHERE j<>x:
SC[j.a,j.b]>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]: delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmpl[x.e, x.f, m, p,lambda]);

SpareCapChannels[j,lambda]:
SC[j.a,j.b,lambda]-ASC[j.a,j.b,lambda]<=SF[j.a,j.b];

Simetry[e,f IN x,p,m,lambda] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmpl[e,f,m,p,lambda]=Fxmpl[e:=f,f:=e,m,p,lambda];

MergeFlowOnSpareLinks[j]:
allSC[j.a,j.b]=SUM(lambda:SC[j.a,j.b,lambda]);

NodeOptimization1[n]:
```

```
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.6.5. ReroutingPRd_wpAfree

```
TITLE
ReroutingPRd_wpA1free

INDEX
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
lambda:=DATABASE("SQLlambda", lambda="ind");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b,lambda]:=DATABASE("outChannelsInFiber","ASCj", a="a", b="b", lambda="lambda");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM_prd","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfPRdSortResWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k" WHERE k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
dm[e,f,m,lambda]=DATABASE("InputDm", e="a", f="b", m="m", lambda="lambda");
delta_jxm[a,b IN j,e,f IN x,m,lambda]=DATABASE("qdfLinkInPathM","deltaIM", a="a", b="b", e="e", f="f", m="m", lambda="lambda");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j,lambda];
allSC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmpl[e,f IN x,m,p,lambda] WHERE p<=Px[e,f,m] AND m<=A[e,f]
EXPORT REFILL TO DATABASE("endReroutingFxmp",Fxmpl="Fxmp", e="e", f="f", m="m", p="p",lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:SC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m],lambda:Fxmpl[x.e,x.f,m,p,lambda])=SUM(lambda:dm[x.e,x.f,m,lambda]);

EnoughSpareCapacitiy[j,x,lambda] WHERE j<>x:
SC[j.a,j.b] + SUM(m<=A[x.e,x.f]:delta_jxm[j.a,j.b,x.e,x.f,m,lambda]*dm[x.e,x.f,m,lambda])>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]:
delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmpl[x.e, x.f, m, p,lambda]);

SpareCapChannels[j,lambda]:
SC[j.a,j.b,lambda]-ASC[j.a,j.b,lambda]<=SF[j.a,j.b];

Simetry[e,f IN x,p,m,lambda] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmpl[e,f,m,p,lambda]=Fxmpl[e:=f,f:=e,m,p,lambda];

MergeFlowOnSpareLinks[j]:
allSC[j.a,j.b]=SUM(lambda:SC[j.a,j.b,lambda]);
```

```
NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.6.6. ReroutingPRd_wpB

```
TITLE
ReroutingPRd_wpB1

INDEX
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
lambda:=DATABASE("SQLlambda", lambda="ind");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b,lambda]:=DATABASE("outChannelsInFiber","ASCj", a="a", b="b", lambda="lambda");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM_prd","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfPRdSortResWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k" WHERE k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
dm[e,f,m,lambda]=DATABASE("InputDm", e="a", f="b", m="m", lambda="lambda");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j,lambda];
allSC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmpl[e,f IN x,m,p,lambda] WHERE p<=Px[e,f,m] AND m<=A[e,f]
EXPORT REFILL TO DATABASE("endReroutingFxmp",Fxmpl="Fxmp", e="e", f="f", m="m", p="p",lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:SC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m,lambda] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m]:Fxmpl[x.e,x.f,m,p,lambda])=dm[x.e,x.f,m,lambda];

EnoughSpareCapacitiy[j,x,lambda] WHERE j<>x:
SC[j.a,j.b]>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]: delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmpl[x.e, x.f, m, p,lambda]);

SpareCapChannels[j,lambda]:
SC[j.a,j.b,lambda]-ASC[j.a,j.b,lambda]<=SF[j.a,j.b];

Simetry[e,f IN x,p,m,lambda] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmpl[e,f,m,p,lambda]=Fxmpl[e:=f,f:=e,m,p,lambda];

MergeFlowOnSpareLinks[j]:
allSC[j.a,j.b]=SUM(lambda:SC[j.a,j.b,lambda]);
```

```
NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.6.7. ReroutingPRd_wpBfree

```
TITLE
ReroutingPRd_wpB1free

INDEX
n:=DATABASE("qdfReducedNodes","Node");
s=n;
d=n;
a=n;
b=n;
e=n;
f=n;
j[a,b] = DATABASE("qdfReducedLinks", a="a",b="b");
x[e,f] = DATABASE("qdfReducedLinks", e="a",f="b");
m =1..100;
i:=(1, 2, 3, 4);
p =DATABASE("dataNumberOfRoutes","NbRoutes");
lambda:=DATABASE("SQLlambda", lambda="ind");

DATA
MaxP=Last(p);
Beta[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
Gama[a,b]= DATABASE("qdfReducedLinks", a="a", b="b");
LA[a,b]:=DATABASE("qdfReducedLinks", "LAj", a="a", b="b");
ASC[a,b,lambda]:=DATABASE("outChannelsInFiber","ASCj", a="a", b="b", lambda="lambda");
Px[e,f,m]=DATABASE("qdfKSHnumResWithM_prd","Px", e="e", f="f", m="m");
delta_jxmp[ a,b IN j,e,f IN x,m,p]=DATABASE("qdfPRdSortResWithM","delta", a="a", b="b", e="e", f="f", m="m", p="k" WHERE k<=MaxP);
A[e,f]=DATABASE("InputAj","Aj", e="a", f="b");
dm[e,f,m,lambda]=DATABASE("InputDm", e="a", f="b", m="m", lambda="lambda");
delta_jxm[a,b IN j,e,f IN x,m,lambda]=DATABASE("qdfLinkInPathM","deltaIM", a="a", b="b", e="e", f="f", m="m", lambda="lambda");
deltaJN[a,b,n]:=DATABASE("LinkIncidentNode", a="a", b="b", n="n");
UF[a,b]:=DATABASE ("outUsedLinks","Fibers", a="a", b="b");
C[i]:=(10000, 20000, 40000, 80000);
K[i]:=(4, 8, 16, 32);

BINARY VARIABLES
deltaNI[n,i];

INTEGER VARIABLES
SF[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SFj", a="a", b="b");
SC[a,b IN j,lambda];
allSC[a,b IN j] EXPORT TO DATABASE("outUsedLinks","SCj", a="a", b="b");
Fxmpl[e,f IN x,m,p,lambda] WHERE p<=Px[e,f,m] AND m<=A[e,f]
EXPORT REFILL TO DATABASE("endReroutingFxmp",Fxmpl="Fxmp", e="e", f="f", m="m", p="p",lambda="lambda");
NodeType[n] EXPORT TO DATABASE("dataNodes","ResType", n="Node");

MODEL

MIN TotNetCost =SUM(j:Beta[j.a,j.b]*SF[j.a,j.b] + Gama[j.a,j.b]*SUM(lambda:SC[j.a,j.b,lambda])) +SUM(n,i: C[i]*deltaNI[n,i])

SUBJECT TO

FlowOverPRestorationRoutes[x,m,lambda] WHERE m<=A[x.e,x.f]:
SUM(p<=Px[x.e,x.f,m]:Fxmpl[x.e,x.f,m,p,lambda])=dm[x.e,x.f,m,lambda];

EnoughSpareCapacitiy[j,x,lambda] WHERE j<>x:
SC[j.a,j.b]  +  SUM(m<=A[x.e,x.f]:delta_jxm[j.a,j.b,x.e,x.f,m,lambda]*dm[x.e,x.f,m,lambda])>=SUM(m<=A[x.e,x.f],p<=Px[x.e,x.f,m]:
delta_jxmp[j.a, j.b, x.e, x.f, m, p]*Fxmpl[x.e, x.f, m, p,lambda]);

SpareCapChannels[j,lambda]:
SC[j.a,j.b,lambda]-ASC[j.a,j.b,lambda]<=SF[j.a,j.b];

Simetry[e,f IN x,p,m,lambda] WHERE m<=A[e,f] AND p<=Px[e,f,m]:
Fxmpl[e,f,m,p,lambda]=Fxmpl[e:=f,f:=e,m,p,lambda];
```

```
MergeFlowOnSpareLinks[j]:
allSC[j.a,j.b]=SUM(lambda:SC[j.a,j.b,lambda]);

NodeOptimization1[n]:
SUM(i:deltaNI[n,i])=1;

NodeOptimization2[n]:
SUM(j:deltaJN[j.a,j.b,n]*(UF[j.a,j.b] +SF[j.a,j.b]))<=2*SUM(i:K[i]*deltaNI[n,i]);

TypeOfNode[n]:
NodeType[n]=SUM(i:K[i]*deltaNI[n,i]);
```

9.7. Основни мрежни модели

9.7.1. Модел за минимална цена на протокот

```
TITLE
MinCost

INDEX
n :=DATAFILE("Jazli.dat",1)
a=n;
b=n;
j[a,b]:=INDEXFILE("Linkovi.dat");
s[n]=DATAFILE("Pateki.dat",1);
d[n]=DATAFILE("Pateki.dat",2);
m[s,d]:=INDEXFILE("Pateki.dat");

DATA
Cost[j]:= SPARSEFILE("LinkCost.dat",2)";
Cap[j]:=SPARSEFILE("LinkCost.dat",3);
Supplay[n]:= SparseFile("Jazli.dat",2);
Demand[n]:=SparseFile("Jazli.dat",3);

VARIABLES
UsedLink[m,j] -> Flow WHERE Cost>0;
Entrance[m,s] WHERE s=m.s;
Destination[m,d] WHERE d=m.d;

MODEL
MIN MinCost =SUM(m, j:Cost*UsedLink);

SUBJECT TO
FlowBal[m,n]:
Entrance IF (m.s=n)  + SUM(j: UsedLink[m,j] WHERE j.b=n)=
Destination IF (m.d=n) +SUM(j: UsedLink[m,j] WHERE j.a=n);

SuppConst[s]:
SUM(m:Entrance[m,s])=Supplay[n=s];

DemConst[d]:
SUM(m:Destination[m,d])=Demand[n=d];

CapConst[j] WHERE Cap>0:
SUM(m:UsedLink[m,j])<= Cap;

END
```

9.7.2. Мрежен модел за најкратки патеки

```
TITLE
Shortes_Path

INDEX
node := INDEXFILE("jazli.dat");
a := node;
b := node;
```

```
j[a,b]:=INDEXFILE("Linkovi.dat");
m[a,b]:=INDEXFILE("pateki.dat");

DATA
Cost[j]:=SPARSEFILE("LinkCost.dat");

VARIABLES
UsedLink[m,j]->UA WHERE Cost>0;

MODEL
MIN TotalCost=SUM(m,j:Cost*UsedLink);

SUBJECT TO
NoCir[m,node]:
1 IF(m.a=node)+SUM (j: UsedLink[m,j] WHERE j.b=node)=
1 IF(m.b=node)+SUM (j: UsedLink[m,j] WHERE j.a=node);

END
```

9.7.3. Мрежен модел за максимален проток

```
TITLE
MaxFlow

INDEX
node := INDEXFILE("Jazli.dat");
fn=node;
tn=node;
link[fn,tn]:=INDEXFILE("Linkovi.dat");
s[node]=DATAFILE("Pateki.dat",1);
d[node]=DATAFILE("Pateki.dat",2);
path[s,d]:=INDEXFILE("Pateki.dat");

DATA
Cap[link]:= SPARSEFILE("LinkCost");

VARIABLES
UsedLink[path,link] -> Flow WHERE (Cap>0);
Entrance[path,s] WHERE s=path.s;
Destination[path,d] WHERE d=path.d;

MODEL
MAX TotalFlow = SUM(path,s:Entrance[path,s]);

SUBJECT TO
FlowBal[path,node]:
Entrance IF (path.s=node)  + SUM(link: UsedLink[path,link] WHERE link.tn=node)=
Destination IF (path.d=node) +SUM(link: UsedLink[path,link] WHERE link.fn=node);

CapConst[link] WHERE Cap>0:
SUM(path:UsedLink[path,link])<= Cap;

END
```

9.8. Некои принципи за дизајн на WDM мрежи со рутирање на бранови должини

9.8.1. Дизајнирање на виртуелна топологија

```
TITLE
Virtual_Topology_Design

INDEX
node:=DATABASE("dataNodes","Node")
k=node;
s=node;
d=node;
i=node;
j=node;
m=node;
n=node;
c:=DATABASE("dataNetwork");

DATA
P[m,n]:=DATABASE("dataPhysical","Pmn",m="m",n="n");
T[node]:=DATABASE("dataNodes","Transmitters", node="Node");
R[node]:=DATABASE("dataNodes","Receivers", node="Node");
L[s,d]=DATABASE("dataTraffic","Lsd",s="s",d="d");
ChannelCapacity[c]:=DATABASE("dataNetwork");
W[m,n]:=DATABASE("dataPhysical","Wmn",m="m",n="n");
VkupnoSoob:=SUM(s,d: L[s,d]);
Obratno:=1/VkupnoSoob;

INTEGER VARIABLES
Lambda[s,d,i,j,c] WHERE s<>d AND i<>j
EXPORT REFILL TO DATABASE("TrafficTbl","l_sdij", s="s", d="d", i="i", j="j", c="c");

BINARY VARIABLES
V[i,j,c] WHERE i<>j
EXPORT REFILL TO DATABASE("VirtualTopologyTbl","Vij", i="i", j="j", c="c");
p[i,j,c,m,n] WHERE P[m,n]>0 AND i<>j
EXPORT REFILL TO DATABASE("PhysicalTopologyTbl","p_ijmn", i="i", j="j",c="c", m="m", n="n");

MODEL
MIN ToatalFlow=Obratno*SUM(s,d,i,j,c:Lambda[s,d,i,j,c]);

SUBJECT TO
BrojNaLaseri[node]:
SUM(j,c:V[i:=node,j,c])<T[node];

BrojNaFiltri[node]:
SUM(i,c:V[i,j:=node,c])<R[node];

LinksInTrail[i,j,k,c]:
SUM(m:p[i,j,c,m,n:=k] IF (k<>i AND k<>j)) + SUM(n:p[i,j,c,m:=k,n] IF (k=i)) + SUM(m:p[i,j,c,m,n:=k] IF (k=j))=
SUM(n:p[i,j,c,m:=k,n] IF (k<>i AND k<>j)) + V[i,j,c] IF (k=i) + V[i,j,c] IF (k=j);

NoCirculation[i,j,k,c]:
SUM(n:p[i,j,c,m:=k,n] IF (k=j)) + SUM(m:p[i,j,c,m,n:=k] IF (k=i))=0;

MaxWavelength[m,n]:
SUM(i,j,c:p[i,j,c,m,n])<=W[m,n]*P[m,n];

FlowBalance[s,d,k]:
SUM(i,c: Lambda[s,d,i,j:=k,c] IF (k<>s AND k<>d))+SUM(j,c:Lambda[s,d,i:=k,j,c] IF (k=s))+SUM(i,c:Lambda[s,d,i,j:=k,c] IF (k=d))=
SUM(j,c:Lambda[s,d,i:=k,j,c] IF (k<>s AND k<>d))+L[s,d] IF (k=s) + L[s,d] IF (k=d);

TroughExistingPath[i,j,s,d]:
SUM(c:Lambda[i,j,s,d,c])<=L[s,d]*SUM(c:V[i,j,c]);

LinkCapacityConstraint[i,j,c]:
SUM(s,d: Lambda[s,d,i,j,c])<=ChannelCapacity*V[i,j,c];
```

9.8.2. Рутирање и доделување на бранови должини

```
TITLE
RWAContinuity

INDEX
node:=DATABASE("dataNodes","Node")
s=node;
d=node;
i=node;
j=node;
lambda=1..2;

DATA
P[i,j]:=DATABASE("dataPhysical","Pmn",i="m",j="n");
L[s,d]=DATABASE("dataTraffic","Lsd",s="s",d="d");
W[i,j]:=DATABASE("dataPhysical","Wmn",i="m",j="n"); ! Îâà å ïàíäàíò íà êàïàöèòåò íà ëèíêîò

INTEGER VARIABLES
Fmax;
Paths[s,d,lambda] WHERE s<>d;

BINARY VARIABLES
F[s,d,i,j,lambda] WHERE s<>d AND i<>j AND P[i,j]>0
EXPORT REFILL TO DATABASE("TrafficTbl","l_sdij", s="s", d="d",i="i",j="j",lambda="c");

MODEL
MIN Fmax

SUBJECT TO
MaximumFlow[i,j] WHERE i<>j:
Fmax>=SUM(s,d,lambda:F[s,d,i,j,lambda]);

NoCirculation1[s,d] WHERE L[s,d]>0:
SUM(j,lambda:F[s,d,i:=d,j,lambda]) + SUM(i,lambda:F[s,d,i,j:=s,lambda])=0;

NoCirculation2[s,d,i,j,lambda] WHERE P[i,j]>0 AND L[s,d]>0:
F[s,d,i:=j,j:=i,lambda]+F[s,d,i,j,lambda]<=1;

FlowConservation[s,d,node,lambda] WHERE L[s,d]>0:
SUM(j:F[s,d,i:=node,j,lambda] IF (node<>d)) -SUM(i:F[s,d,i,j:=node,lambda] IF (node<>s)) = Paths[s,d,lambda] IF (node=s)-Paths[s,d,lambda] IF
(node=d);

ColorConstraint[i,j,lambda] WHERE P[i,j]>0:
SUM(s,d:F[s,d,i,j,lambda])<=1;

TotalFlowOfSDPair[s,d]:
SUM(lambda :Paths[s,d,lambda])= L[s,d];

CapConst[i,j] WHERE W[i,j]>0:
SUM(s,d,lambda:F[s,d,i,j,lambda])<= W[i,j];
```

10. Библиографија

Вовед

[1] Ieee Journal On Selected Areas In Communications, vol. 18, no. 10, October 2000
WDM Optical Communication Networks: Progress and Challenges
Biswanath Mukherjee
[2] Ryan, Hankin, and Kent (RHK). http://www.rhk.com
[3] IEEE Press, 1999
Planning Telecommunication Networks
Thomas G. Robertazzi
[4] Kluwer Academic Publishers, 2001
Network Synthesis Problems
Christelle Wynants

Оптички компоненти

[5] McGraw-Hill, 1997
Optical Communication Network
Biswanath Mukherjee
[6] Morgan Kaufmann Publishers
Optical Networks A Practical Perspective
Rajiv Ramaswami, Kumar N. Sivarajan
[7] Prentice Hall, 1998
Understanding Optical Communications
Harry J. R. Dutton
[8] IEEE Commun. Mag., December. 1998
All-Optical Wavelength Conversion Using SOA Nonlinearities
Derek Nesset, Tony Kelly, and Dominique MarcenacBT Laboratories
[9] IEEE Communications Magazine, vol.37 No.2, February 1999
Future Photonic Transport Networks Based on WDM Technologies
H. Yoshimura, et. al.
[10] IEEE Communications Magazine, vol. 36, no. 12, December 1998 pp. 28-36
Developments in Wavelength Division Multiple Access Networking
John M. Senior, Michael R. Handley and Mark S. Leeson
[11] IEEE Communications Magazine, vol. 36, no. 12, December 1998 pp. 42-49
Multiwavelength Receivers for WDM Systems
Frank Tong
[12] IEEE Communications Magazine, vol. 36, no. 12, December 1998 pp. 39-41
Multifrequency Lasers and Applications in WDM Networks
Martin Zirngibl
[13] IEEE Communications Magazine, vol. 36, no. 12, December 1998 pp. 62-68
Arrayed Waveguide Gratings for Wavelength Routing
Kenneth A. McGreer
[14] IEEE Communications Magazine, vol. 36, no. 12, December 1998 pp. 50-55
Tunable Optical Filters for Dense WDM Networks
Dan Sadot and Efraim Boimovich

Оптички телекомуникациски мрежи

[15] Artech House, 1996
Advances in Transport Network Technologies, Photonic Networks, ATM, and SDH
Ken-Ichi Sato
[16] IEEE Commun. Mag., Apr. 1997.
Pan-European optical networking using wavelength division multiplexing
M. Berger, M. Chbat, A. Jourdan, P. Demeester, B. Van Caenegem, P. Godsvang, B. Hein, M. Huber, R. Marz, A. Leclert, T. Olsen, G.Tobolka, and T. Van Den Broeck
[17] McGraw-Hill, 1997
Optical Communication Network
Biswanath Mukherjee
[18] Morgan Kaufmann Publishers
Optical Networks A Practical Perspective
Rajiv Ramaswami, Kumar N. Sivarajan
[19] IEEE Communication Magazine, vol.36 no.2, February 1998

Преживливост на оптичко ниво

[20] IEEE Comm. Mag., Volume: 33 2, March 2000
Optical Layer Survivability: A Services Perspective
Ornan Gerstel and Rajiv Ramaswami
[21] IEEE Comm. Mag., Volume: 33 2, February 1995, pp. 58-74.
"Emerging Technologies for Fiber Network Survivability,"
T-H. Wu
[22] IEEE Comm. Mag., Volume: 34 12, December 1996, pp. 86–94.
Photonic Transport Network OAM Technologies
Ken-ichi Sato
[23] IEEE JSAC,Volume: 16 7 , pp. 1008-1024
Management and Control of Transparent Optical Networks
M. W. Maeda
[24] IEEE JSAC,Volume: 16 7 , pp. 1134-1145
Multiple-Star Wavelength-Router Network and Its Protection Strategy
A. M. Hill, M. Brierley, R. M. Percival, R. Wyatt, D. Pitcher, K. M. I. Pati, I. Hall, and J.-P. Laude
[25] IEEE JSAC,Volume: 16 7 , pp. 1146-1157
Dimensioning of Survivable WDM Networks
B. Van Caenegem, W. Van Parys, F. De Turck, and P. M. Demeester
[26] IEEE JSAC,Volume: 16 7 , pp. 1158-1165
Design Protection for WDM Optical Networks
O. Crochat and J.-Y. Le Boudec
[27] IEEE JSAC,Volume: 16 7 , pp. 1166-1178
Fault Tolerant Multiwavelength Optical Rings with Limited Wavelength Conversion.
O. Gerstel, R. Ramaswami, and G. H. Sasaki
[28] IEEE JSAC,Volume: 16 7 , pp. 1179-1189
Optical Cross-Connect System Incorporated with Newly Developed Operation and Management System
T. Shiragaki, N. Henmi, T. Kato, M. Fujiwara, M. Misono, T. Shiozawa, and S. Suzuki
[29] IEEE JSAC,Volume: 16 7 , pp. 1190-1198
Optimal Design and Evaluation of Survivable WDM Transport Networks
Y. Miyao and H. Saito
[30] Artech House, 1997
Broadband Networking ATM, SDH and SONET
Mike Sexton, Andy Reid
[31] Artech House, 1996
Advances in Transport Network Technologies, Photonic Networks, ATM, and SDH
Ken-Ichi Sato

[32] IEEE Comm. Mag., Volume: 36 5 , May 1998 pp. 122 -126
The quantitative impact of survivable network architectures on service availability
M.R. Wilson
[33] IEEE JSAC,Volume: 12 1 , pp. 5-12
Public Network Integrity - Avoiding Crissis in trust
J.C. McDonald
[34] IEEE JSAC,Volume: 12 1 , pp. 46-51
Framework for Network Suvivability Performance
A. Zolfagahari and F.J. Kaudel
[35] IEEE JSAC,Volume: 12 1 , pp. 52-58
A framework for Characterizing Disaster-Based Network Survivability
S.C Liew, K.W. Lu
[36] Committee T1-Telecommunications, November 1993
T1A1.2/93-001R3, Technical report on network survivability performance
[37] IEEE Communications Magazine, January 2002
New Options and Insights for Survivable Transport Networks
Wayne Grover, John Doucette, Matthieu Clouqueur, Dion Leung, TRLabs, University of Alberta
Demetrios Stamatelakis, Network Photonics Canada
[38] IEEE ICC 2002
Optimal Configuration of p-Cycles in WDM Networks
D.A. Schupke, C.G. Gruber, and A. Autenrieth
[39] DRCN 2001, Budapest, Hungary, Oct. 2001, pp. 113–20.
Mining the Rings: Strategies for Ring-to-Mesh Evolution
M. Clouqueur et al.
[40] DRCN 2001, Budapest, Hungary, Oct. 2001, pp. 99–106.
Increasing the Efficiency of Span-restorable Mesh Networks on Low-connectivity Graphs
W.D. Grover and J. Doucette
[41] DRCN 2001, Budapest, Hungary, Oct. 7–10, 2001, pp. 121–28
Comparison of Mesh Protection and Restoration Schemes and the Dependency on Graph Connectivity
J. Doucette and W.D. Grover
[42] DRCN 2000, Munich, Germany, Apr. 2000, pp. 92–104.
Bridging the Ring-mesh Dichotomy with P-cycles
W.D. Grover and D Stamatelakis
[43] IEEE JSAC, vol. 18, no. 10, Oct. 2000, pp.1938–49.
IP-Layer Restoration and Network Planning Based on Virtual Protection Cycles
D. Stamatelakis and W.D. Grover
[44] Proc. IEEE ICC ’98, Atlanta, June 1998, pp. 537–43.
Cycle-Oriented Distributed Preconfiguration: Ring-like Speed with Meshlike Capacity for Self-planning Network Restoration
W.D. Grover and D. Stamatelakis
[45] Proc. IEEE, vol. 85, no. 10, Oct. 1997, pp. 1582–611.
Self-organizing Broadband Transport Networks
W.D. Grover
[46] IEEE JSAC Volume: 12 1 , pp. 120 -127
Self-healing ATM networks based on virtual path concept
Kawamura, R.; Sato, K.-I.; Tokizawa, I.
[47] IEEE/ACMTrans. Net., vol. 6, no. 3, June 1998, pp. 325–36.
Optimal Capacity Placement for Path Restoration in STM or ATM Mesh-Survivable Networks
R.R. Iraschko, M. H. MacGregor, and W.D. Grover
[48] IEEE/ACM Trans. Net., vol.7 no.1, Feb. 1999
Restoration Strategies and Spare Capacity Requirements in Self-healing ATM Networks”
Y. Xiong and L.G. Mason

Математичко програмирање

[49] IEEE Press, 1999
Planning Telecomunication Networks
Thomas G. Robertazzi
[50] Athena Scientific, 1998
Network Optimization: Continuous and Discrete Models
Dimitri P. Bertsekas

Некои принципи за дизајн на WDM мрежи со рутирање на светлосни патеки

[51] McGraw-Hill, 1997
Optical Communication Network
Biswanath Mukherjee
[52] IEEE/ACM Transactions On Networking, vol 1. No. *5*. October 1996
Some Principles for Designing a Wide-Area WDM Optical Network
Biswanath Mukherjee, Dhritiman Banerjee, S. Ramamurthy, and Amarnath Mukherjee
[53] IEEE Journal on Selected Areas in Communications, vol. 14, no. *5,* june 1996
A Practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks
Dhritiman Banerjee and Biswanath Mukherjee

Димензионирање на преживливи WDM мрежи

[54] IEEE Journal on Selected Areas In Communications, vol. 16, No.7, September 1998
Dimensioning of Survivable WDM Networks
Bart Van Caenegem, Wim Van Parys, Filip De Turck, and Piet M. Demeester
[55] IEEE Journal on Selected Areas In Communications, vol. 16, No.7, September 1998 **(**complete issue)
[56] IEEE Journal On Selected Areas In Communications, vol. 18, no. 10, October 2000 (complete issue)
[57] IEEE Journal on Selected Areas In Communications, vol. 16, No.7, June 1996
Design of the Optical Path Layer in Multiwavelength Cross-Connected Networks
Nico Wauters and Pet Demeester
[58] IEEE Journal on Selected Areas In Communications, vol. 15, February 1997
Wavelength requirements in arbitrarily connected wavelength-routed optical networks
S. Baroni, P. Bayvel
[59] Kluwer Academic Publishers, 2001
Network Synthesis Problems
Christelle Wynants
[60] Management Science, vol. 17, no. 11, pp 712–716, July 1971.
"Finding the K shortest loopless paths in a network,"
J. Y. Yen
[61] Artech House, 1996
Advances in Transport Network Technologies, Photonic Networks, ATM, and SDH
Ken-Ichi Sato