

**УНИВЕРЗИТЕТ “СВ.КИРИЛ И МЕТОДИЈ”
ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ-СКОПЈЕ
ИНСТИТУТ ЗА ИНФОРМАТИКА**



НАТАША В. МАКСИМОВА

**НАДЕЖНОСТ НА
ПОВЕЌЕ – СОСТОЈБЕНИ ДВОТЕРМИНАЛНИ
ТРАНСПОРТНИ СИСТЕМИ**

-МАГИСТЕРСКИ ТРУД-

**СКОПЈЕ, 2009
РЕПУБЛИКА МАКЕДОНИЈА**

Ментор: Доц. д-р Марија Михова
Природно-математички факултет
Институт за информатика

Коментор: Проф. д-р Жанета Попеска
Природно-математички факултет
Институт за информатика

Комисија: Доц. д-р Марија Михова
Природно-математички факултет
Институт за информатика

Проф. д-р Жанета Попеска
Природно-математички факултет
Институт за информатика

Проф.д-р Смиле Марковски
Природно-математички факултет
Институт за информатика

Датум на одбрана: _____

Датум на промоција: _____

Научна област: Информатика

Предговор

Една од квалитативните карактеристики на еден систем е неговата можност долго да работи. Поради круцијалното значење на оваа карактеристика, се појавила потреба за нејзино квантитативно претставување, наречено надежност. Во основа, надежност е квантитативна анализа на грешките, нивните причини и последици. Таа е најзначајна карактеристика за квалитетот на системот, бидејќи системот мора да работи задоволително пред да се оценат другите квалитативни карактеристики. Вообичаено, при анализа на надежноста на системот, може да се користат различни мерки за перформансите на системот, при што, доколку перформансите на системот се пониски од некое одредено ниво, во тој случај може да се рече дека се има случено грешка во системот или дека системот се расипал. Бидејќи работата на системот е процес кој се менува во текот на времето, можноста за негово расипување и неговата надежност се различни и зависат од времето на работа. Под надежност на системот се подразбира веројатноста дека системот ќе работи со очекувана функционалност под специфични работни услови за специфичен временски период.

Во теоријата на надежност посебно место зазема анализата на надежност на транспортните системи. Голем број на физички проблеми, како на пример компјутерските мрежи, цевководни системи и мрежи за електрично напојување треба да се моделираат со помош на ваквите системи. Овие системи имаат голема комплексност - многу јазли кои ги претставуваат корисниците или ресурсите што сакаат да комуницираат и врски со кои се овозможува оваа комуникација. Било каква грешка или прекин на работа на системот ќе има влијание врз сите корисници што се вклучени во системот. Пресметувањето на надежноста на транспортните системи се наметнува како голем проблем. Поради тоа за пресметување на надежноста на овие системи се бараат алгоритамски решенија.

Во овој магистерски труд се разгледуваат повеќе - состојбени двотерминални транспортни системи. Надежноста на двотерминалните транспортни системи е доста проучуван проблем од бинарна перспектива (т.е., кај бинарните транспортни системи). Меѓутоа во некои системи, како системите за дистрибуција на вода, телекомуникациските системи, системите за снабдување со гас и нафта, системите за генерирање на енергија од водата, бинарниот пристап се покажал како недоволен за добивање на модели на надежност кои ќе го претставуваат вистинското однесување на системот, односно целосно не го карактеризираат актуелното однесување на надежноста на системот. Поради тоа, проблемот може да биде разгледуван или анализиран под претпоставка дека компонентите од транспортните системи (т.е. линковите) може бидат во една од повеќе состојби. Во тој случај и транспортниот систем може да се најде во една од повеќе состојби. Надежноста на транспортниот систем може да се пресмета доколку се познати минималните

пат множества или минималните пресек множества. Значи проблемот е да се најдат овие множества, а надежоста ќе се пресмета со помош на формулата за вклучување и исклучување.

Во воведниот дел ќе бидат дефинирани основните поими кои се користат за анализа на надежноста. Овде ќе биде дадено што значи надежност на систем, исто така ќе бидат воведени и дефинирани и други поими кои се користат при анализа на надежност, како достапност, одржливост, средно време на поправка, средно време на расипување, и други. Исто така ќе бидат дадени основните концепти на надежност на бинарен систем, дефинициите на пат и пресек множества. Потоа ќе се дефинира систем со повеќе состојби, ќе се дефинира структурната функција за ваков систем и ќе биде дадена основната формула за надежност на ваков систем. Исто така ќе бидат дефинирани и минимални пат и пресек вектори.

Во втората глава ќе опишеме неколку техники со кои се овозможува анализа на надежност на бинарни транспортните системи. Ќе бидат разгледани техники за пресметување на надежност на двотерминални и комплетно - терминални бинарни транспортни системи. Во последните делови од оваа глава ќе ја дефинираме повеќе - состојбената двотерминална надежност,

Во третата глава ќе бидат анализирани алгоритмот за определување на минимални пат множества и алгоритмот за определување на минимално пресек множество дадени во трудовите [6] и [9]. Овие алгоритми исто така ќе бидат илустрирани и со примери. Алгоритмите се имплементирани во програмскиот јазик C#.

Во четвртата глава се разгледуваат 3 алгоритми кои се дадени во [16] и [17], за пресметување на минималните пат вектори на повеќе - состојбен двотерминален транспортен систем. Сите овие алгоритми се едноставни за разбирање и се базираат на едноставни операции помеѓу вектори. Првиот алгоритам работи само за транспортни системи со капацитети на линкови $\{0,1,2,\dots,M_i\}$. Со цел да се забрза работата на овој алгоритам даден е и убрзан алгоритам. Вториот алгоритам исто така има ограничување за вредностите што може да ги примаат капацитетите на линковите. Капацитетите на линковите може да бидат целобројни вредности. Исто така и за овој алгоритам е даден модифициран алгоритам, за да се забрза неговата работа. Последниот алгоритам е од најголемо значење, бидејќи за овој алгоритам не постојат ограничувања на вредностите што може да ги примаат капацитети на линковите. За сите овие алгоритми се направени програми во програмскиот јазик C#.

Содржина

1. Основни концепти на надежност	1
1.1 Надежност на бинарни системи	4
1.2 Надежност на системи со повеќе состојби	6
2. Надежност на транспортни системи	9
2.1 Видови на графови	10
2.2 Дефиниција на надежност на бинарни транспортен систем	12
2.2.1 Двотерминална надежност на бинарни транспортни системи	15
2.2.2 Комплетно - терминална надежност	21
2.2.3 K - терминална надежност	24
2.2.4 Преглед на истражувањата за бинарна двотерминална надежност	24
2.3 Надежност на повеќе - состојбени двотерминални транспортни системи	25
2.3.1 Основни дефиниции за повеќе - состојбен двотерминален транспортен систем	26
2.3.2 Пресметување на надежност	27
2.3.3 Преглед на истражувањата на повеќе - состојбена двотерминална надежност	28
2.4 Наоѓање на бинарни минимални патишта	30
3. Преглед на некои познати алгоритми за определување на минимални пат или минимални пресек множества	32
3.1 Алгоритам за пресметување на минимални пат множества на Ramirez-Marquez и Coit за транспортни системи со реални капацитети	32

3.1.1	Опис на алгоритмот за пресметување на минимални пат множества	33
3.1.2	Анализа на алгоритмот	38
3.2	Алгоритам за пресметување на минимални пресек множества на Ramirez-Marquez, Coit и Tortorella за транспортни системи со слабо хомогени компоненти	40
3.2.1	Опис на проблемот	40
3.2.2	Опис на алгоритмот за пресметување на минимални пресек множества	41
4.	Нови алгоритми за наоѓање на минимални пат вектори	52
4.1	Алгоритам за наоѓање на минимални пат вектори за транспортен систем со капацитети на линкови $\{0, 1, 2, \dots, M_i\}$	53
4.2	Убрзан алгоритам за наоѓање на минимални пат вектори за транспортен систем со капацитети на линкови $\{0, 1, 2, \dots, M_i\}$	66
4.3	Споредба на алгоритам за транспортен систем со капацитети на линкови $\{0, 1, 2, \dots, M_i\}$ и убрзаниот алгоритам за транспортен систем со капацитети на линкови $\{0, 1, 2, \dots, M_i\}$	73
4.4	Убрзан алгоритам за наоѓање на минимални пат вектори за транспортен систем со капацитети на линкови $\{0, 1, 2, \dots, M_i\}$ со контрола на максималните нивоа	74
4.5	Алгоритми за наоѓање на минимални пат вектори за транспортен систем чии капацитети на линкови може да примаат произволни целобројни вредности	76
4.6	Алгоритам за наоѓање на минимални пат вектори за транспортен систем со произволни капацитети на линковите	83
4.7	Споредба на алгоритмите дадени во Глава 3 со алгоритам за реални капацитети	90
5.	Заклучок	92
	Користена литература	94
	Прилог А	97

Глава 1

Основни концепти на надежност

Надежноста претставува најзначајна карактеристика за квалитетот на еден систем, бидејќи системот мора да работи задоволително пред да се оценат другите квалитативни карактеристики. Анализата на надежност на еден систем подразбира анализа на грешките кои се случуваат во текот на работата на еден систем, нивните причини и последици. При анализа на надежност на системите постојат два пристапи. Од една страна се гледа процесот на работа на системот во текот на времето, а од друга страна не интересира како надежноста зависи од составните компоненти на системот. Кога се работи за транспортни системи кои се од интерес на нашава анализа, нас пред се не интересира како надежноста на еден систем зависи од неговите составните компоненти. Поради тоа, во овој труд пред се се осврнуваме на зависноста на надежноста од составните компоненти. Но, сепак, на почеток ќе ги дадеме и основните дефиниции за надежност на еден систем, кога неговата работа се разгледува како процес кој се менува во текот на времето. Ако системот го анализираме во овој контекст, сметаме дека можноста за негово расипување и неговата надежност зависат од времето на работа, па според тоа со помош на математички апарат можат да се запишат како функции кои зависат од времето на работа.

Дефиниција 1.1 Надежност на еден систем е веројатноста дека тој ќе работи со очекуваната функционалност под специфични работни услови за специфичен временски период. Математички, функцијата на надежност $R(t)$ е веројатноста дека системот ќе работи успешно, без грешка во временскиот интервал од време 0 до време t , односно веројатноста системот да не се расипе за време t .

$$R(t) = P(T > t). \quad (1.1)$$

каде T е случајна променлива која го претставува **времето на откажување**, т.е. времето на првото расипување (failure time).

Веројатноста на откажување (појавување на грешка во системот) или функцијата на расипување на системот е

$$F(t) = 1 - R(t) = P(T \leq t). \quad (1.2)$$

$F(t)$ е всушност функцијата на распределба на случајната променлива T .

Ако случајната променлива T има густина на распределба $f(t)$, тогаш

$$R(t) = \int_t^{\infty} f(x) dx. \quad (1.3)$$

Густината на распределба може математички да се опише како $\lim_{\Delta t \rightarrow 0} P(t < T \leq t + \Delta t)$. Ова може да се интерпретира како веројатност дека времето на откажување ќе се случи во временскиот интервал $(t, t + \Delta t)$.

Освен функцијата на надежност, во анализата на надежност на системи е од интерес да се разгледува и математичкото очекување на времето до следното расипување кое се нарекува **средно време на расипување (MTTF)**

Дефиниција 1.2 Средно време на расипување (MTTF) се дефинира како очекувана вредност на времето на живот на системот, пред да се случи грешка.

Ако се претпостави дека функцијата на надежност на системот е $R(t)$, тогаш средното време на расипување, може да се пресмета како:

$$\text{MTTF} = \int_0^{\infty} t \cdot f(t) dt = \int_0^{\infty} R(t) dt. \quad (1.4)$$

Ако случајната променлива T – време на откажување е експоненцијално распределена со параметар λ , т.е. функцијата на распределба на T е $F(t) = 1 - e^{-\lambda t}$ тогаш

$$\text{MTTF} = \int_0^{\infty} R(t) dt = \int_0^{\infty} e^{-\lambda t} dt = \frac{1}{\lambda}. \quad (1.5)$$

Функцијата стапка на неуспех (failure rate function) е многу значајна во анализата на надежност, бидејќи таа ја специфицира стапката на стареење на системот.

Дефиниција 1.3 Функцијата стапка на неуспех $\lambda(t)$ се дефинира со:

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{R(t) - R(t + \Delta t)}{\Delta t R(t)} = \frac{f(t)}{R(t)}. \quad (1.6)$$

Величината $\lambda(t)dt$ ја претставува веројатноста дека уредот со старост t , ќе се расипе во временскиот интервал $(t, t + \Delta t)$.

Кога системот не успева да исполни некоја одредена функција, вообичаена постапка што се изведува е да се лоцира и поправи грешката. Системот се обновува за да може да работи ефикасно со вршење на некои поправки или пак со замена на некои компоненти.

Дефиниција 1.4 Одржливост се дефинира како веројатност дека системот што не работи ќе биде вратен во функционална состојба во даден временски период, кога одржувањето е извршено во согласност со одредени процедури и ресурси.

Воопшто, одржливост е веројатноста за изолирање и поправка на грешка во системот во даден временски период. Да го означиме со T **времето на поправка**. Ако времето на поправка T има густина на распределба $g(t)$, тогаш одржливоста $V(t)$ се дефинира како веројатност системот кој што не е исправен да се врати во работна состојба за време t , т.е.,

$$V(t) = P(T \leq t) = \int_0^t g(x) dx. \quad (1.7)$$

Значајна мерка која се користи е и **средното време на поправка (MTTR)**. Тоа е очекуваната вредност од времето на поправка.

Друг значаен концепт поврзан со надежноста е и **достапноста** на системот.

Дефиниција 1.5 Функцијата на достапност, означена со $A(t)$ се дефинира како веројатност дека системот е во работна состојба во време t .

$$A(t) = P(\text{Системот е во работна состојба во временски момент } t).$$

Просечното време меѓу две расипувања (MTBF) е друга значајна мерка. Просечното време меѓу две расипувања е математичкото очекување на случајната променлива “време помеѓу две расипувања” и притоа, $MTBF = MTTF + MTTR$.

Во наредниве подглавја ќе бидат дадени дефинициите на надежност во зависност од составните компоненти на системот. Во зависност од тоа дали компонентите на системот се бинарни или повеќе состојбни, пристапот и анализата се разликуваат.

1.1 Надежност на бинарни системи

Ако компонентите во системот и самиот систем можат да се најдат само во две состојби: компонентата (или системот) работи и компонентата (или системот) е расипана, тогаш за тавиот систем велиме дека е бинарен систем. Состојбата на ваквиот систем може да се претстави како функција од состојбите на компонентите со таканаречена структурна функција. Во ова поглавје ќе биде дадена дефиниција на структурната функција за бинарен систем и како се изразува надежноста на системот со помош на неговата структурна функција.

Да разгледаме систем кој се состои од n компоненти. Нека случајната променлива x_i е состојбата на i -тата компонента. Случајната променлива x_i може да прими две вредности: 0 и 1, при што прима вредност 1 ако компонентата е во работна состојба и вредност 0 ако компонентата е расипана. Дефинираме случаен вектор $\vec{x} = (x_1, x_2, \dots, x_n)$, што претставува вектор од состојбите на компонентите на системот. Овој вектор прима вредности од множеството $\{0,1\}^n$. Системот ќе биде во работна состојба при одредени комбинации од компоненти кои работат, а при други комбинации нема да работи. Нека состојбата на системот ја обележеме со φ . Ако системот работи $\varphi = 1$, додека ако системот е расипан $\varphi = 0$. Бидејќи состојбата на системот зависи од состојбата на компонентите од кои е составен, дефинираме структурна функција во која состојбата на системот се изразува преку состојбата на неговите компоненти.

Дефиниција 1.6 Нека системот е составен од n компоненти. Функцијата

$$\varphi: \{0,1\}^n \rightarrow \{0,1\}, \quad \varphi(\vec{x}) = \varphi \quad (1.8)$$

се нарекува **структурна функција**.

Да ја разгледаме сега надежноста на бинарните системи врз основа на нивната структура.

Дефиниција 1.7 Пат е множество од компоненти за кои важи дека, ако тие компоненти работат тогаш системот работи.

Дефиниција 1.8 Пресек е множество од компоненти за кое важи дека ако компонентите од тоа множество се расипани, тогаш и целиот систем е расипан.

Дефиниција 1.9 Еден пат е **минимален пат** ако не постои негово вистинско подмножество кое исто така е пат.

Дефиниција 1.10 Еден пресек е **минимален пресек** ако не постои негово вистинско подмножество кое исто така е пат.

Ако $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_s$ се минимални патишта, а C_1, C_2, \dots, C_k се минимални пресеци, тогаш:

$$\varphi(\vec{x}) = \max_{1 \leq j \leq s} \min_{i \in \mathcal{P}_j} x_i = \min_{1 \leq j \leq k} \max_{i \in C_j} x_i. \quad (1.9)$$

Од (1.1) имаме дека веројатноста системот со структурна функција $\varphi(\vec{x})$ да работи е:

$$R = P(\varphi(\vec{x}) = 1). \quad (1.10)$$

Со помош на репрезентациите на минимални патишта и пресеци функцијата на надежност може да се претстави

$$R = E(\max_{1 \leq j \leq s} \min_{i \in \mathcal{P}_j} x_i) = E(\min_{1 \leq j \leq k} \max_{i \in C_j} x_i). \quad (1.11)$$

Нека $p_i = P(x_i = 1) = 1 - P(x_i = 0)$, $i=1, \dots, n$ е веројатноста i -тата компонента да работи, и нека со \vec{p} го обележиме векторот $\vec{p} = (p_1, p_2, \dots, p_n)$. Кога надежноста на системот ќе ја изразиме само преку надежноста на неговите компоненти ќе ја означиме со $R_{\vec{p}}$. Ако компонентите од кои е составен системот се независни, тогаш функцијата на надежност е:

$$R_{\vec{p}} = \sum_{\vec{x} \in \{0,1\}^n} \varphi(\vec{x}) \left(\prod_{i=1}^n (p_i^{x_i} (1-p_i)^{1-x_i}) \right). \quad (1.12)$$

Ако минималните патишта се познати, тогаш надежноста може да се пресмета со користење на формулата за вклучување и исклучување [6].

$$R = P\left(\bigcup_{h=1}^j \mathcal{P}_h\right) = \sum_{h=1}^j P(\mathcal{P}_h) - \sum_{h < k} P(\mathcal{P}_h \cap \mathcal{P}_k) + \dots + (-1)^j P(\mathcal{P}_1 \cap \dots \cap \mathcal{P}_j). \quad (1.13)$$

каде j е бројот од минимални патишта и \mathcal{P}_h е минималниот пат h .

1.2 Надежност на системи со повеќе состојби

Постојат системи кои имаат компоненти на кои квалитетот на работа може да им опадне, но компонентата да не се расипе целосно. Квалитетот на работа на компонентата се нарекува **ниво на работа на компонентата**. Ние разгледуваме компоненти кои имаат конечен број нивоа на работа. При што нивоата на работа како вредности може да примат било кој реален број. Значи доколку случајната променлива x_i има k нивоа во кои работи, и едно во кое не работи таа ќе може да прима вредности од множеството $S_i = \{0, m_{i1}, m_{i2}, \dots, m_{ik}\}$, каде $0 < m_{i1} < m_{i2} < \dots < m_{ik}$ и $m_{ij} \in \mathbb{R}$ за $j=1, \dots, k$. Состојбата m_{ik} ја нарекуваме **перфектно ниво** и обично перфектното ниво за i -тата компонента се означува со M_i . Со 0 го означуваме нивото во кое компонентата е тотално расипана. Па може да запишеме дека $0 \leq x_i \leq M_i$, $x_i \in \mathbb{R}$. Множеството S_i ги содржи сите можни состојби во кои може да се најде i -тата компонента.

Ако системот има n -компоненти тогаш случајниот вектор $\vec{x} = (x_1, x_2, \dots, x_n)$, $x_i \in S_i$, $i=1, \dots, n$, што ги опишува состојбите на сите компоненти на системот се нарекува **вектор состојба на системот**. Векторот на состојба на системот го нарекуваме само состојба на системот. Ова е всушност состојбата на секоја од компонентите од кои е составен системот. Множеството од сите состојби ќе го бележиме со E . Значи $E = S_1 \times S_2 \dots S_n$.

Целиот систем исто како и неговите компоненти може да работи со различен квалитет, така што во повеќе состојби да има ист квалитет. Квалитетот на работа на системот се нарекува **ниво на работа**. Множеството од нивоа на работа на системот се бележи со $S = \{m^1, m^2, \dots, m^k\}$ каде m^i , за $i=1, \dots, k$ се можните нивоа на работа во кои може да се најде целот систем. Максималното ниво на работа на системот обично го означуваме со M .

Нивото на работа на системот во зависност од неговите компоненти, исто како и кај бинарните системи може да се претстави преку структурна функција.

Дефиниција 1.11 Нека S_i е множество на состојби на i -тата компонента во n компонентен систем, а S множество нивоа на работа на системот. Функцијата $\varphi: S_1 \times S_2 \times \dots \times S_n \rightarrow S$, со

$$\varphi = \varphi(\vec{x}). \quad (1.14)$$

се нарекува **структурна функција**.

Слично како и кај бинарните системи, во некое определено време веројатноста на работа на компонентата на системот може да се разгледува како константа. Нека p_{ij} е веројатноста j -тата компонента да се најде во состојба j , за $i=1, \dots, n$ и $j=0, \dots, M_i$. Дефинираме **вектор на веројатности на компонентата** за секоја од n -те компоненти:

$$\vec{p}_i = (p_{i1}, p_{i2}, \dots, p_{iM_i}). \quad (1.15)$$

Векторот \vec{P} нека ја претставува веројатноста за секое ниво на секоја компонента:

$$\vec{P} = (\vec{p}_1, \vec{p}_2, \dots, \vec{p}_n). \quad (1.16)$$

Надежноста зависи од веројатноста на работа на составните компоненти на системот. Ако правиме анализа на надежноста на системот само врз основа на неговата структура, функцијата на надежност ќе ја означуваме со $R_{\vec{p}}$.

Кај бинарните системи состојбите во кои системот работи се претставуваат со пат множества, а состојбите во кој тој не работи со пресек множества. Додека пак минималните перформанси што треба да ги задоволи системот за да работи се претставуваат со минимални пат множества, а максималните перформанси при кои системот не работи со пресек множества. Аналогно на овие множества, кај повеќе – состојбените системи се дефинираат пат, пресек, минимален пат и минимален пресек вектори.

За да ги дефинираме оие вектори, про ќе дефинираме подредување на множеството состојби на системот.

Дефинициј 1.12 За векторот \vec{y} , велиме дека е помал од векторот \vec{x} , $\vec{y} < \vec{x}$, (или доминиран од \vec{x}) ако и само ако $\forall i, y_i \leq x_i$ и за некое $k, y_k < x_k$.

Како што кажавме претходно, нивото на работа на системот се претставува со структурна функција φ , дефинирана на множеството E . Со помош на оваа функција ќе дефинираме пат и пресек вектори.

Дефиниција 1.13 Векторот $\vec{x} \in E$ е **пат вектор за ниво d** , ако $\varphi(\vec{x}) \geq d$.

Дефиниција 1.14 Векторот $\vec{x} \in E$ е **пресек вектор за ниво d** ако $\varphi(\vec{x}) < d$.

Векторот \vec{x} е пат вектор за ниво d , ако системот во состојба \vec{x} работи со ниво поголемо или еднакво од d . Додека векторот \vec{x} е пресек вектор за ниво d , ако системот во состојба \vec{x} работи со ниво помало од d .

Дефиниција 1.15 Векторот \vec{x} е **минимален пат вектор за ниво d** ако е пат вектор за ниво d и не постои друг пат вектор за ниво d кој е строго помал од него.

Дефиниција 1.16 Векторот \vec{x} е **минимален пресек вектор за ниво d** ако е пресек вектор за ниво d и не постои друг пресек вектор за ниво d кој е строго поголем од него.

Множеството од сите минимални пат вектори за ниво d ќе го означиме со \mathcal{P}_d , а множеството од сите минимални пресек вектори за ниво d со C_d .

Веројатноста системот да работи со ниво поголемо или еднакво од d е веројатноста тој да се најде во вектор состојба поголема или еднаква од некој од векторите од множеството \mathcal{P}_d .

$$R_d = P(\varphi(\vec{x}) \geq d). \quad (1.17)$$

Глава 2

2. Надежност на транспортни системи

Многу физички проблеми (како на пример компјутерски мрежи, цевоводни системи, мрежи за електрично напојување и други) може да се моделираат со помош на транспортни системи. Под зборот **транспортен систем** се подразбира физички проблем што може да се моделира како математички граф составен од јазли и линкови, при што линковите може да бидат насочени или ненасочени, зависно од природата на системот. Ваквите видови на системи имаат **извори** и **сливници**. Од главен интерес овде се процесите кои се однесуваат на поврзувањата или постоењето на патишта помеѓу изворите и сливниците. Системите како на пример, компјутерските или комуникациските мрежи, се доста комплексни, односно имаат многу јазли, што всушност ги претставуваат корисниците или ресурсите што сакаат да комуницираат меѓусебно и голем број на линкови со кои се овозможува оваа комуникација. Бидејќи голем број на корисници се поврзани во еден ваков систем, појавувањето на грешка или прекин во поврзување може да има широко влијание, поточно да влијае на многу корисници. Од овие причини една од главните цели кога се разгледуваат вакви системи е да се доведе надежноста на високо ниво.

Во овој дел на кусо ќе бидат дискутирани неколку техники со кои се овозможува анализа на надежноста на транспортен систем. Всушност физичката карактеристиките на еден транспортен систем се претставува со помош на граф. Со помош на графот можеме да видиме како се поврзани различните јазли на

ваквиот систем. Заради тоа математичките техники кои се користат при решавање на некои проблеми поврзани со надежноста на транспортен систем се базираат на својствата што ги имаат математичките графови. Затоа во овој дел прво на кусо ќе се задржиме на терминологијата која ќе ја користиме од теоријата на графови.

2.1 Видови на графови

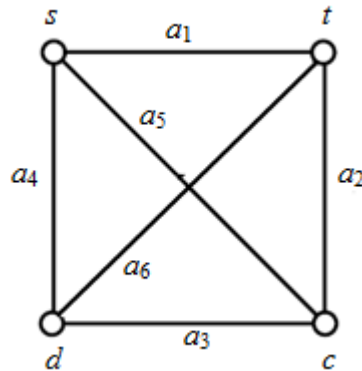
Како што претходно напоменавме најприкладен начин за претставување на транспортен систем е со помош на граф. Ребрата на графот ги претставуваат линковите, кои во општ случај би биле телефонски кабел за телекомуникациски систем, водоводни цевки за систем за пренос на вода, гасоводни цевки за систем за пренос на гас, и слично, додека темињата ги претставуваат споевите меѓу различни линкови. Да нагласиме дека термините што се користат за опишување на графови не се унифицирани. Најчесто се користат термините теме и лак, јазли и гранки или места и меѓусебни врски, линкови. Во овој магистерски труд ќе ги користиме термините јазли и линкови. Еден општ модел подразбира дека и јазлите и линковите може да откажат. Меѓутоа ние ќе разгледуваме поедноставен модел, во кој само линковите може да откажат, додека за јазлите ќе претпоставиме дека се наоѓаат во перфектна состојба. Ако комуникацијата меѓу пар јазли се одвива во една насока, за линкот што ги поврзува јазлите велиме дека е насочен линк. Ако во графот постои барем еден пар од јазли, меѓу кои комуникацијата се одвива само во една насока станува збор за **насочен граф**. Додека, доколку комуникација помеѓу пар од јазли може да се одвива во двете насоки, за линкот велиме дека е ненасочен и ако во еден граф сите линкови се ненасочени станува збор за **ненасочен граф**.

Зависно од природата на системот, графовите кои се користат при претставувањето на транспортните системи може да имаат само ненасочени линкови, или пак едни линкови да се насочени а други не.

На пример ако се работи за телефонски кабел, линковите се ненасочени, додека пак ако се работи за водоводни цевки тогаш линковите се насочени.

Алгоритмите кои ги решаваат проблемите во врска со надежностите на транспортните системи, може да се базираат и на едниот и на другиот тип на графови. Но често за поедноставување на ситуацијата се земаат насочени графови, поради тоа што ненасочените графови можеме да ги разгледаме како специјален случај на насочен граф, во кој секој линк е претставен со два идентични паралелни линкови со спротивна насока.

Да разгледаме ненасочен граф $G(\mathcal{N}, \mathcal{A})$ без лупи, каде што \mathcal{A} е множеството од линкови и \mathcal{N} е множеството од јазли. Јазлите ќе ги означуваме со n_i додека линковите со a_j . Исто така за означување на некој линк, може да ги користиме јазлите што тој линк ги поврзува. На пример доколку j -от линк ги поврзува s -от и t -от јазол, тогаш можеме да запишеме: $a_j = (n_s, n_t) = a(s, t)$. Исто така можеме да кажеме дека линкот j е инцидентен со јазлите s и t . Лупа во еден граф е линкот (n_s, n_s) , т.е. овој линк го поврзува јазол n_s самиот со себе.



Фигура 2.1 Граф со 4 јазли, кој може да претставува компјутерска мрежа

Да го разгледаме сега графот што е претставен со Фигура 2.1. Овој граф има 4 јазли и 6 линкови. Јазлите се означени со s, t, c и d , додека со a_1, a_2, a_3, a_4, a_5 и a_6 . Сите линкови во графот се ненасочени. Може да се забележи дека во графот прикажан со Фигура 2.1 постојат сите можни врски помеѓу четирите јазли.

Да напоменеме овде дека алгоритмите кои се дадени во Глава 4, за пресметување на надежност на транспортните системи работат и за двата типа на графови. Во магистерскиот труд ќе разгледуваме графови без лупи, заради тоа што таквите линкови не пренесуваат информации од едно место на друго, поради што може да се занемарат. Исто така и паралелните линкови кај ненасочен граф не интересираат, бидејќи може да се разгледуваат како еден линк. При што тежината на линкот ќе ја земеме како збир од тежините на паралелните линкови во ненасочениот граф.

2.2 Дефиниција на надежност на бинарни транспортен систем

Како што рековме претходно, од значење ни е како структурата на системот влијае на надежноста. Бидејќи не интересира надежноста на системот во зависност од неговите составни делови, ние можеме времето да го занемариме, или пак да го специфицираме, при што во секој момент ќе имаме константна веројатност системот да работи (или да работи со одредено ниво на работа). Понатаму системите ќе ги разгледуваме на овој начин.

За да го запазиме континуитетот на оваа теорија, прво ќе ги разгледаме бинарните транспортни системи. Да кажеме дека установено е дека таквите системи недоволно ја опишуваат работата на некои транспортни системи. Но бидејќи прво тие се користеле за опишување на надежноста на транспортните системи и поради нивната едноставност ова подглавје ќе го посветиме на нивно дефинирање и опишување.

Во ова подглавје ќе бидат опишани видовите на транспортни системи во однос на бројот на извори и сливници, и заради тоа од интерес е да се занемари фактот дека линковите може да работат во повеќе состојби.

При разгледување на бинарни транспортни системи се претпоставува дека компонентите на системот, како и самиот систем може да бидат во една од две можни состојби: да бидат во работна состојба, т.е., исправни или да бидат расипани.

При разгледување на транспортните системи ние пред се овде ќе се фокусираме на тоа колку многу од јазлите од кои што е составен системот може меѓусебно да комуницираат. Овде терминот **успешно функционирање** може да има различни интерпретации, во зависност од бројот на јазли што комуницираат меѓу себе.

Доколку имаме еден извор, да го означиме со s , и еден сливник, да го означиме со t , тогаш успешно функционирање се дефинира како постоење на еден или повеќе патишта од изворот s до сливникот t . Ова се нарекува **двотерминален проблем**, и веројатноста за успешна комуникација помеѓу изворот s и сливникот t е наречена **двотерминална надежност**.

Ако успешно функционирање се дефинира како постоење на патишта што ги поврзуваат сите јазли во транспортниот систем едни со други, тогаш имаме **комплетно - терминален проблем**. Постоењето на патишта помеѓу, на пример јазолот s , и сите останати $n-1$ јазли, t_1, t_2, \dots, t_n , е еквивалентно со постоење на патишта помеѓу сите јазли. Веројатноста на успешна комуникација помеѓу јазолот s и јазлите t_1, t_2, \dots, t_{n-1} е наречена **комплетно - терминална надежност**.

Поформално кажано, комплетно-терминалната надежност е веројатноста дека јазолот n_i може да комуницира со јазолот n_j , за секој пар од јазли (n_i, n_j) , (каде $i \neq j$). Ова тврдење е еквивалентно со тврдењето дека јазолот s може да комуницира со сите други јазли, $t_1=n_2, t_2=n_3, \dots, t_{n-1}=n_n$.

Имено, да избереме произволен јазол, на пример јазолот n_x за $x \neq 1$. Да претпоставиме дека јазолот n_x комуницира со јазолот s и јазолот s може да комуницира со сите останати јазли, и нека таа комуникација е двонасочна.

Меѓутоа, штом n_x е поврзан со s , тој исто така ќе биде поврзан и со сите други јазли, поради тоа што s е поврзан со сите јазли. Тогаш, комплетно - терминалната поврзаност за $x=1$ резултира со комплетно - терминална поврзаност за $x \neq 1$, со што тврдењето е докажано.

Надежноста R е веројатноста на успешно функционирање. Комплетно - терминална надежност ја означуваме со R_{all} :

$$R_{all} = P(\text{сите } n \text{ јазли се поврзани}), \quad (2.1)$$

додека двотерминална надежност со извор s и сливник t , се означува со R_{st} :

$$R_{st} = P(\text{јазлите } s \text{ и } t \text{ се поврзани}) \quad (2.2)$$

k - терминална надежност е веројатноста дека постои подмножество од k ($2 \leq k \leq n$) поврзани јазли во транспортниот систем.

Доколку се разгледува проблем на надежност на транспортни системи, како прво мора да се специфицира каков тип на надежност имаме.

Од интерес е да се определи врската меѓу надежноста $R(t)$ и достапноста $A(t)$ на еден систем. За таа цел ги дефинираме следниве настани.

A_0 е настанот - нема расипување на системот во временскиот интервал од 0 до t .

A_1 е настанот - едно расипување и една поправка во временскиот интервал од 0 до t

A_2 е настанот - две расипувања и две поправки во временскиот интервал од 0 до t , итн.

Тогаш

$$A(t) = P(A_0 \cup A_1 \cup A_2 \dots). \quad (2.3)$$

Бидејќи настаните во горната формула меѓусебно се исклучуваат, горната равенка може да се напише како сума од веројатности:

$$A(t) = P(A_0) + P(A_1) + P(A_2) + \dots \quad (2.4)$$

Јасно

- a) Првиот член во равенката (2.4) е надежноста $R(t)$.
- b) $A(t) = R(t) = 1$ за $t=0$.
- c) За $t > 0$, $A(t) > R(t)$
- d) $R(t) \rightarrow 0$, за $t \rightarrow \infty$

Добивањето на резултати за достапноста може да биде доста комплексно доколку времето на расипување и времето на поправка се со различни веројатности распределби. Изведувањето може се поедностави ако се

претпостави дека времето на расипување и времето на поправка се распределени во согласност со експоненцијалната распределба (со константен интензитет на расипување λ и константен интензитет на поправка μ). Исто така и **средното време на расипување** и **средното време на поправка** може да се користат за опишување на процесот на поправање и достапноста. Во многу случаи, термините **средно време помеѓу две расипувања (MTBF)** и **средно време помеѓу две поправки (MTBR)** се користат наместо просечно време на расипување и просечно време на поправка. Доколку интензитетите на расипување и на поправање се константни (λ и μ соодветно), тогаш просечното време на расипување е еднакво на $MTBF=1/\lambda$ и средното време на поправање е еднакво на $MTBR=1/\mu$. Тогаш за $A(t)$ се добива:

$$A(t) = \frac{\mu}{(\lambda + \mu)} + \frac{\lambda}{(\lambda + \mu)} e^{-(\lambda + \mu)t}.$$

Значи, решението за $A(t)$ има експоненцијален опаѓачки преоден член и константен стационарен член. После неколку циклуси на поправање и расипување, преодниот член се губи и достапноста може да се репрезентира само преку стационарниот член. Во случај кога интензитетите на расипување и на поправање се константни (λ и μ соодветно), стационарната состојба на достапноста е дадена со равенката:

$$A_{ss}(t) = \lim_{t \rightarrow \infty} A(t) = \frac{\mu}{(\lambda + \mu)} = \frac{MTBF}{(MTBF + MTBR)}. \quad (2.5)$$

Во нормален случај, просечното време на поправка $\frac{1}{\mu}$ е многу помало од просечното време на расипување. Со развивање на стационарната состојба на достапноста во ред се добива дека

$$A_{ss}(t) = A(\infty) = \frac{1}{\left(1 + \frac{\lambda}{\mu}\right)} = 1 - \frac{\lambda}{\mu} + \frac{\lambda^2}{\mu^2} + \dots \approx 1 - \frac{\lambda}{\mu}.$$

Имајќи го во предвид овој факт, се добива дека во било кој добро дизајниран систем, A_{ss} е близу до единица.

Достапност исто така може да се изрази и преку, **времето кога системот функционира ($U(t)$)** и **времето кога системот е паднат ($D(t)$)**. Зависноста на достапноста од времето на работа на системот и времето на неработење на системот е дадена со следнава формула:

$$A_{ss} = \frac{U(t)}{[U(t) + D(t)]}. \quad (2.6)$$

Формулата (2.6) е посоодветна од формулата (2.5), ако е потребно да се оценува A_{ss} врз база на податоци добиени од набљудување на работата на системот.

Мора да се спомене дека поправките на елементите ги сметаме за изолирани настани, т.е. дека поправката на елементот започнува веднаш по појавувањето на грешка. Не постои ред на чекање за поправка на поединечните елементи.

2.2.1 Двотерминална надежност на бинарни транспортни системи

Во ова поглавје на кратко ќе бидат разгледани неколку техники за пресметување на надежност на транспортни системи, и ќе бидат применети на едноставни примери. Значајноста на овие техники се состои во тоа што тие претставуваат основа за пишување на алгоритмите за пресметување на надежноста на транспортни системи.

Наједноставен начин за оценување на надежноста на двотерминален транспортен систем е со нумерирање на сите елементарни настани кои припаѓаат во просторот на состојби. Поточно, се нумерираат сите можни комбинации од линкови кои работат и линкови кои не работат. Ако транспортниот систем има e линкови, тогаш вкупниот број на комбинации во кои може да се најде транспортниот систем изнесува 2^e . Нека ваквите настани ги обележиме со E_i . Надежноста е всушност веројатноста на унијата од оние настани за кои постои пат од изворот s до сливникот t . Значи:

$$R_{st} = P(E_1 \cup E_2 \cup E_3 \cup \dots). \quad (2.8)$$

Бидејќи овие настани се меѓусебно дисјунктни, веројатноста на унијата е збир од веројатностите на одделните настани.

$$R_{st} = P(E_1) + P(E_2) + P(E_3) + \dots \quad (2.9)$$

Да го разгледаме графот што е даден во Фигура 2.1. Бидејќи овде ја разгледуваме двотерминалната надежност, да претпоставиме дека извор е јазолот s и сливник е јазолот t . Бидејќи овој граф има 6 линкови, вкупниот број на настани што треба да ги нумерираме изнесува $2^6=64$. Сите овие настани ќе бидат дадени во Прилогот А, во Табела 1 на страна 97. Ги користиме следниве ознаки:

E_i = настан i ;

j = линкот j е во работна состојба;

j' = линкот j не е во работна состојба;

Ако постои барем еден пат од изворот a до сливникот b , тогаш ќе имаме позитивен исход од настанот, и во Табела.1 во Прилог А, страна ?? ќе го користиме терминот “добар” за овој случај. А доколку не постои ниту еден пат

од изворот a до сливникот b имаме негативен исход и ќе го користиме терминот “лош”.

Во Прилог А, во Табела 1, страна 97 ќе бидат дадени сите 64 настани како и нивните исходи.

Двотерминалната надежност од изворот s до сливникот t , може да се претстави како унија од сите настани од Табела 1 во Прилог А, страна 97 кои имаат позитивен исход, т.е., за кои постои пат помеѓу изворот s и сливникот t .

$$R_{st} = [P(E_1)] + [P(E_2) + \dots + P(E_7)] + [P(E_8) + P(E_9) + \dots + P(E_{22})] \\ + [P(E_{23}) + P(E_{24}) + \dots + P(E_{34}) + P(E_{37}) + \dots + P(E_{42})] \\ + [P(E_{43}) + P(E_{44}) + \dots + P(E_{47}) + P(E_{50}) + P(E_{56})] + [P(E_{58})]. \quad (2.10)$$

Во првата заграда во равенката (2.10) има еден член, тоа е случај кога сите линкови се исправни. Ако сите линкови се идентични и независни, и веројатноста на работа е p , тогаш веројатноста на настанот E_1 е p^6 . Слично, во втората заграда имаме шест настани чија веројатност е еднаква на qp^5 , каде $q=1-p$ е веројатноста на откажување. Значи за надежноста ќе добиеме:

$$R_{st} = p^6 + 6 \cdot q \cdot p^5 + 15 \cdot q^2 \cdot p^4 + 18 \cdot q^3 \cdot p^3 + 7 \cdot q^4 \cdot p^2 + q^5 \cdot p. \quad (2.11)$$

Да забележиме дека во случај кога $p=1$ и $q=0$, надежноста ќе биде еднаква на единица, додека доколку $p=0$ и $q=1$, надежноста ќе биде 0.

Треба да се забележи дека овој начин на пресметување на надежноста побарува доста време и напор, иако процедурата е релативно јасна. Бројот на настани расте експоненцијално, 2^e . На пример, доколку имаме 10 линкови, во тој случај бројот на настани ќе биде еднаков на $2^{10}=1024$, а за поголемо e ќе се добијат милион настани. Поради овие причини потребно е да се разгледаат други методи.

Следно во ова поглавје ќе ги разгледаме методите за одредување на пат и пресек множества. Кај двотерминални бинарни транспортни системи, **пат множеството** претставува група од линкови што формираат пат помеѓу изворот s и сливникот t . Пат множеството е минимално ако не може да содржи подмножество од линкови кое исто така е пат множество. Нека постојат i -пат множества помеѓу s и t , и нека тоа се множествата $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_i$.

Ги дефинираме следниве настани:

ω_A – линковите во множеството A се во работна состојба

ω'_A – линковите во множеството A не се во работна состојба

Тогаш надежноста ќе биде пресметана со следниот израз:

$$R_{st} = P(\omega_{\mathcal{P}_1} \cup \omega_{\mathcal{P}_2} \cup \dots \cup \omega_{\mathcal{P}_i}). \quad (2.12)$$

Пресек множество е група од линкови кои доколку се отстранат од графот ќе се прекинат сите патишта помеѓу изворот s и сливникот t . Ако пресек множеството е минимално, тогаш тоа не може да содржи подмножество кое исто така е пресек множество.

Надежноста изразена преку j -пресек множества C_1, C_2, \dots, C_j е дадена со следниот израз:

$$R_{st} = 1 - P(\omega_{C_1}' \cup \omega_{C_2}' \cup \dots \cup \omega_{C_j}'). \quad (2.13)$$

Да ги најдеме минималните пат и пресек множества за транспортниот систем од Фигура 2.1. Ако изворот е s и сливникот t , минималните пат и пресек множества ќе бидат дадени во Табела 2.1.

Пат множества	Пресек множества
$P_1 = \{1\}$	$C_1 = \{1, 4, 5\}$
$P_2 = \{5, 2\}$	$C_2 = \{1, 6, 2\}$
$P_3 = \{4, 6\}$	$C_3 = \{1, 5, 6, 3\}$
$P_4 = \{2, 3, 4\}$	$C_4 = \{1, 2, 3, 4\}$
$P_5 = \{5, 3, 6\}$	

Табела 2.1 Минимални пат и пресек множества за Фигура 2.1

$$R_{st} = 1 - P(\omega_{C_1}' \cup \omega_{C_2}' \cup \omega_{C_3}' \cup \omega_{C_4}'), \quad (2.14a)$$

$$R_{st} = 1 - P(\omega_{\{1, 4, 5\}}' \cup \omega_{\{1, 6, 2\}}' \cup \omega_{\{1, 5, 6, 3\}}' \cup \omega_{\{1, 2, 3, 4\}}'), \quad (2.14b)$$

$$\begin{aligned} R_{st} = 1 - [& P(\omega_{\{1, 4, 5\}}') + P(\omega_{\{1, 6, 2\}}') + P(\omega_{\{1, 5, 6, 3\}}') + P(\omega_{\{1, 2, 3, 4\}}') \\ & + [P(\omega_{\{1, 2, 4, 5, 6\}}') + P(\omega_{\{1, 3, 4, 5, 6\}}') + P(\omega_{\{1, 2, 3, 4, 5\}}') \\ & + P(\omega_{\{1, 2, 3, 5, 6\}}') + P(\omega_{\{1, 2, 3, 4, 6\}}') + P(\omega_{\{1, 2, 3, 4, 5, 6\}}')] \\ & - [P(\omega_{\{1, 2, 3, 4, 5, 6\}}') + P(\omega_{\{1, 2, 3, 4, 5, 6\}}')] \\ & + P(\omega_{\{1, 2, 3, 4, 5, 6\}}') + P(\omega_{\{1, 2, 3, 4, 5, 6\}}')] \\ & + [P(\omega_{\{1, 2, 3, 4, 5, 6\}}')]. \end{aligned} \quad (2.14c)$$

При изразувањето на веројатноста на унијата на настани овде ја искористивме формулата за вклучување и исклучување (1.13).

Ако сите линкови имаат веројатност на неуспех еднаква на q и се помеѓу себе независни, од равенката 2.14c ќе ја добиеме следнава равенка:

$$\begin{aligned} R_{st} = 1 - [2q^3 + 2q^4] + [5q^5 + q^6] - [4q^6] + [q^6], \\ R_{st} = 1 - 2q^3 - 2q^4 + 5q^5 - 2q^6. \end{aligned} \quad (2.15)$$

Се проверува дека за $q = 1, R_{st} = 0$ и за $q = 0, R_{st} = 1$. Ако се замени за $q = 0.1$ се добива:

$$R_{st} = 1 - 2 \cdot (0.1)^3 - 2 \cdot (0.1)^4 + 5 \cdot (0.1)^5 - 2 \cdot (0.1)^6 = 0.997848. \quad (2.16)$$

За пресметување на надежноста, како што претходно кажавме, се користи формулата за исклучување и вклучување. Со користење на оваа формула, при пресметување на надежноста може да се добие серија од веројатности при што последните членови во оваа серија се многу мали така што може да се изостават. Со изоставување на овие членови се добива поедноставна апроксимативна формула. Овие членови се производи од веројатности, па ако овие веројатности се мали, членовите од производите од повисок ред може да се занемарат. Оваа техника за пресметување на надежноста се нарекува **апроксимација со отсекување**.

Пресметувањето на надежноста преку пат множества се преферира да се користи во случаи кога веројатностите на успех се мали, т.е., во таканаречени ниско надежни региони. Овие региони не се од практичен интерес. Додека пак, пресметување на надежноста врз база на пресек множества се користи кога веројатностите на неуспех се мали, т.е. во таканаречени високо надежни региони, кои се реално региони од практичен интерес. Во пракса почесто се користат апроксимациите со пресек множествата.

Да ја разгледаме равенката 2.15. Ако ги земеме во предвид само првите два члена од оваа равенка, при претпоставка дека $q=0.1$ се добива равенката 2.17:

$$R_{st} \geq 1 - [2q^3 + 2q^4] = 1 - 0.002 - 0.0002 = 0.9978. \quad (2.17)$$

Резултатот што го добивме е многу близок до точниот резултат што е добиен во равенката 2.16. Ако се вклучи следниот член од равенката 2.16, ќе се добие следнава апроксимација:

$$R_{st} \leq 1 - [2q^3 + 2q^4] + [5q^5 + q^6] = 0.9978 + 5 \cdot (0.1)^5 + (0.1)^6 = 0.997851. \quad (2.18)$$

Равенката 2.17 претставува долна граница додека равенката 2.18 горна граница. Да нагласиме дека колку повеќе членови од формулата за вклучување и исклучување се вклучени, ќе се добијат множества од алтернативни граници. Треба да се забележи дека равенката 2.16 е груба долна граница, истото важи и за равенката 2.17 која е груба горна граница, но помеѓу овие две граници се наоѓа точниот резултат.

$$0.9978 \leq R_{st} \leq 0.997851. \quad (2.19)$$

Може да се добие апроксимативна вредност за R_{st} како аритметичка средина на двете граници.

$$R_{st} = \frac{0.9978 + 0.997851}{2} = 0.9978255. \quad (2.20)$$

Точноста на претходната апроксимација може да се оцени со разгледување на девијацијата (отстапката) на пресметаната веројатност на неуспех $F_{st} = 1 - R_{st}$. Во регионите со висока надежност, сите вредности за надежноста R_{st} се близу до единица и разликите се “лажно” мали. Тогаш како критериум за оценување на грешката се користи:

$$\%_{error} = \frac{|F_{st}(\text{оценето}) - F_{st}(\text{точно})|}{F_{st}(\text{точно})} \cdot 100\%. \quad (2.21)$$

Ако ги замениме резултатите добиени со равенките 2.16 и 2.20 се добива:

$$\%_{error} = \frac{|0.0021745 - 0.002152|}{0.002152} \cdot 100\% = 1.05\%. \quad (2.22)$$

Се гледа дека оваа апроксимација е добра за примерот што го разгледуваме. Меѓутоа се препорачува да се користи и во многу други случаи. Јасно, при оценувањето на големи транспортни системи, кога точната надежност не е позната, може да ја оцениме равенката 2.22 со користење на разликите помеѓу апроксимациите што содржат два и три члена. За броителот и за средната вредност на именителот се добива:

$$\%_{error} = \frac{|0.0022 - 0.002149|}{0.0021745} \cdot 100\% = 2.35\%. \quad (2.23)$$

Моменталниот приказ води до заклучок дека апроксимациите од повисок ред секогаш се блиски и треба да се користат во именителот на границата на грешката. Броителот од друга страна треба да биде разлика помеѓу два члена од повисок ред. Конкретно за примерот што го разгледуваме имаме:

$$\%_{error} = \frac{|0.0022 - 0.002149|}{0.002149} \cdot 100\% = 2.37\%. \quad (2.24)$$

Равенка, слична на равенката (2.24) треба да се користи за последните два члена во пресметувањето, за да се одреди кога треба да се прекине пресметувањето на алтернативните граници. Процесот прекинува кога апроксимацијата на грешката дава приближна грешка помала од бараната.

Претходно дискутиравме за апроксимацијата со отсекување на членовите во формулата за вклучување и исклучување. Сега ќе разгледаме апроксимација

со исклучување на малку веројатните пат или пресек множества. Оваа техника се нарекува **апроксимација со подмножества**. Јасно, поголема е веројатноста да се појават оние пресек множества кои содржат помал број на линкови, отколку тие што содржат поголем број на линкови. Сега ќе направиме една апроксимација со изоставување на пресек множествата C_3 и C_4 од равенката 2.14a. Значи имаме:

$$\begin{aligned} R_{st} &\leq 1 - P(\omega_{C_1}' \cup \omega_{C_2}') = 1 - P(\omega_{C_1}') - P(\omega_{C_2}') + P(\omega_{C_1}')P(\omega_{C_2}') \\ &= 1 - P(\omega_{\{1, 4, 5\}}') - P(\omega_{\{1, 6, 2\}}') + P(\omega_{\{1, 2, 4, 5, 6\}}'). \end{aligned} \quad (2.25a)$$

За $q = 0.1$, добиваме

$$R_{st} \leq 1 - 2 \cdot (0.1)^3 + (0.1)^5 = 0.99801. \quad (2.25b)$$

На овој начин ја одредивме горната граница. На сличен начин се определува и долната граница, кога од равенката 2.12 ќе се исклучат сите пат множества, освен оние што се составени од еден линк, тоа е пат множеството P_1 и пат множествата кои се составени од два линкови, тоа се множествата P_2 и P_3 . Добиваме:

$$\begin{aligned} R_{st} &\geq P(\omega_{P_1} \cup \omega_{P_2} \cup \omega_{P_3}) = P(\omega_{\{1\}} \cup \omega_{\{2, 5\}} \cup \omega_{\{4, 6\}}) = P(\omega_{\{1\}}) + P(\omega_{\{2, 5\}}) \\ &\quad + P(\omega_{\{4, 6\}}) - [P(\omega_{\{1, 2, 5\}}) + P(\omega_{\{1, 4, 6\}}) + P(\omega_{\{2, 4, 5, 6\}})] \\ &\quad + [P(\omega_{\{1, 2, 4, 5, 6\}})]. \end{aligned} \quad (2.26a)$$

За $p=0.9$, добиваме

$$R_{st} \geq p + 2p^2 - 2p^3 - p^4 + p^5 = 0.9 + 2 \cdot (0.9)^2 - 2 \cdot (0.9)^3 - (0.9)^4 + (0.9)^5 = 0.99639. \quad (2.26b)$$

Од равенките 2.24b и 2.25b се добиваат границите за R_{st}

$$0.99639 \leq R_{st} \leq 0.99801. \quad (2.27)$$

Ја наоѓаме и апроксимацијата на R_{st} како аритметичка средина од горната и долната граница

$$R_{st} = \frac{0.99639 + 0.99801}{2} = 0.9971955. \quad (2.28)$$

Границата на грешката за оваа апроксимација е пресметана на ист начин како во равенката 2.23

$$\%_{error} \approx \frac{|0.00361 - 0.00199|}{0.0028045} \cdot 100\% = 57.8\%. \quad (2.29)$$

Процентот на грешката е поголем отколку кај апроксимацијата со отсекување, но останува доволно мал за да апроксимацијата е валидна. Комплексноста и овде е експоненцијална, од ред 2^x , меѓутоа x е мал цел број и 2^x е со мала големина (незначителна големина). Освен тоа, алгоритмите за пат и пресек множество ќе одземаат помалку време бидејќи не треба да се најдат сите пат и пресек множества, туку само оние од ред $\leq x$.

2.2.2 Комплетно - терминална надежност

Проблемот на комплетно - терминална надежност е многу посложен од проблемот на двотерминална надежност. Во овој дел ќе ги разгледаме сите методи што ги разгледаваме во подглавјето 2.3 меѓутоа сега за случај на комплетно - терминална надежност

Овде работиме на ист начин како во Глава 2.2, со тоа што сега ќе ги оцениме како “добри” оние настани за кои постојат патишта помеѓу сите јазли (во транспортниот систем од Фигура 2.1 помеѓу јазлите s и t , јазлите s и c и јазлите s и d), додека како “лоши” ќе ги оцениме оние настани кои што не ги поврзуваат односно не постојат патишта помеѓу сите јазли. Со примена на оваа рестрикција од Табела 1 од Прилог А, страна 97, ќе ја добиеме Табела 2.2

Настан	s - b	s - c	s - d	Веројатност
E_1	да	Да	Да	p^6
E_2, E_3, \dots, E_7	да	Да	Да	qp^5
E_8, E_9, \dots, E_{22}	да	Да	Да	q^2p^4
$E_{23}, E_{24}, E_{26}, E_{27},$ $E_{28}, E_{29}, E_{30}, E_{32}$ $E_{33}, E_{34}, E_{37}, E_{38}$ $E_{39}, E_{40}, E_{41}, E_{42}$	да	Да	Да	q^3p^3
Останати 26 настани за кои барем еден пар од јазли не е поврзан	–	–	–	$q^3p^3, q^4p^2, q^5p^1, q^6$

Табела 2.2 Модификација на Табела А.1 за комплетно – терминална надежност

$$R_{\text{all-terminal}} = R_{\text{all}} = \sum_{\substack{i=1 \\ i \neq 25,31,35,36}}^{42} P(E_i). \quad (2.30)$$

Ако ги замениме веројатностите од Табела 2.2 во равенката 2.30 ќе се добие:

$$R_{\text{all}} = p^6 + 6 \cdot q \cdot p^5 + 15 \cdot q^2 \cdot p^4 + 16 \cdot q^3 \cdot p^3. \quad (2.31)$$

Ако замениме за $p=0.9$ и $q=0.1$ се добива:

$$R_{\text{all}}=0.9^6 + 6 \cdot 0.1 \cdot 0.9^5 + 15 \cdot 0.1^2 \cdot 0.9^4 + 16 \cdot 0.1^3 \cdot 0.9^3=0.995814. \quad (2.32)$$

Може да се види дека комплетно – терминалната надежност за истиот транспортен систем е помала од двотерминалната надежност која ја пресметавме претходно.

Комплетно – терминалната надежност може да се пресмета со помош на пресек или пат множества при што може да се пресмета или точната вредност или пак да се користат различни апроксимации.

Пресметувањето на комплетно – терминалната надежност е доста комплексно дури и во случај кога имаме помали транспортни системи, како на пример транспортниот систем даден во Фигура 2.1. Во овој дел само ќе се дискутира како може да се добијат решенијата за комплетно – терминалната надежност. Пресметувањето на надежноста на комплетно - терминални транспортни системи кои се користат во пракса е доста комплицирано и се изведува со помош на компјутерски програми. Значи, главна цел во овој дел е да се даде основа со чија помош може да се разбере како се врши пресметувањето и исто така да послужи како основа за апроксимирачките методи што ќе се анализираат подоцна во оваа глава.

Пат множества		
<i>s-b</i>	<i>s-c</i>	<i>s-d</i>
$\mathcal{P}_1=\{1\}$	$\mathcal{P}_6=\{5\}$	$\mathcal{P}_{11}=\{4\}$
$\mathcal{P}_2=\{2, 5\}$	$\mathcal{P}_7=\{1, 2\}$	$\mathcal{P}_{12}=\{5, 6\}$
$\mathcal{P}_3=\{4, 6\}$	$\mathcal{P}_8=\{4, 3\}$	$\mathcal{P}_{13}=\{1, 6\}$
$\mathcal{P}_4=\{2, 3, 4\}$	$\mathcal{P}_9=\{1, 6, 3\}$	$\mathcal{P}_{14}=\{1, 2, 3\}$
$\mathcal{P}_5=\{3, 5, 6\}$	$\mathcal{P}_{10}=\{4, 5, 2\}$	$\mathcal{P}_{15}=\{5, 2, 6\}$

Табела 2.3 Пат множества за пресметување на комплетно – терминална надежност

Пресек множества		
<i>s-b</i>	<i>s-c</i>	<i>s-d</i>
$C_1=\{1, 5, 4\}$	$C_5=\{5, 4, 1\}$	$C_9=\{4, 6, 3\}$
$C_2=\{1, 2, 6\}$	$C_6=\{5, 3, 2\}$	$C_{10}=\{4, 5, 1\}$
$C_3=\{1, 3, 5, 6\}$	$C_7=\{5, 6, 4, 2\}$	$C_{11}=\{4, 1, 2, 3\}$
$C_4=\{1, 2, 3, 4\}$	$C_8=\{5, 6, 1, 3\}$	$C_{12}=\{4, 2, 5, 6\}$

Табела 2.4 Пресек множества за пресметување на комплетно – терминална надежност

Во Табела 2.3 се дадени пат множествата за парови јазли, *s-b*, *s-c* и *s-d*. Додека во Табела 2.4 за овие парови се дадени пресек множествата. Овде да забележиме дека, ако помеѓу јазолот *s* и сите други јазли постојат патишта што ги поврзуваат тогаш, постојат патишта што ги поврзуваат сите јазли.

$$R_{\text{all}} = P([\omega_{p_1} \cup \omega_{p_2} \cup \dots \cup \omega_{p_5}] \cap [\omega_{p_6} \cup \omega_{p_7} \cup \dots \cup \omega_{p_{10}}] \cap [\omega_{p_{11}} \cup \omega_{p_{12}} \cup \dots \cup \omega_{p_{15}}]). \quad (2.33)$$

Доколку равенката 2.33 се разложи ќе се добијат 125 униии од пресеци. Јасно е дека пресметувањето на оваа надежност би одзело доста време. На сличен начин надежноста може да ја изразиме и преку пресек множествата:

$$R_{\text{all}} = 1 - P([\omega'_{C_1} \cup \omega'_{C_2} \cup \omega'_{C_3} \cup \omega'_{C_4}] \cup [\omega'_{C_5} \cup \omega'_{C_6} \cup \omega'_{C_7} \cup \omega'_{C_8}] \cup [\omega'_{C_9} \cup \omega'_{C_{10}} \cup \omega'_{C_{11}} \cup \omega'_{C_{12}}]). \quad (2.34)$$

Со разложување на равенката 2.34 ќе се добие унија од 12 настани, значи да заклучиме дека само апроксимациите или компјутерските решенија се единствени практични пристапи.

Доколку сакаме да ја пресметаме надежноста со помош на равенките 2.33 и 2.34, најдобро ќе биде да се применат некои апроксимации. Како прво може да се забележи дека некои од пресек множествата се еднакви, а тоа се следниве множества: $C_1 = C_5 = C_{10}$, $C_4 = C_{11}$, $C_7 = C_{12}$ и $C_3 = C_8$. Значи можеме да ги изоставиме множествата C_5 , C_{10} , C_{11} , C_{12} и C_8 , со што во равенката 2.34 ќе имаме 7 пресек множества. Претпоставуваме дека сите линкови имаат еднаква веројатност $p=1-q$.

$$R_{\text{all}} \geq 1 - P(\omega'_{C_1} \cup \omega'_{C_2} \cup \omega'_{C_3} \cup \omega'_{C_4} \cup \omega'_{C_6} \cup \omega'_{C_7}). \quad (2.35a)$$

Ако се претпостави дека $q=0.1$ се добива:

$$R_{\text{all}} \geq 1 - 4 \cdot q^3 - 3 \cdot q^4 = 1 - 4 \cdot (0.1)^3 - 3 \cdot (0.1)^4 = 0.9957. \quad (2.35b)$$

За добивање на горната граница се додаваат 21 членови во втората заграда во равенката 2.34, и се добива:

$$R_{\text{all}} \leq 0.99571 + 17 \cdot q^5 + 4 \cdot q^6 = 0.99600. \quad (2.36)$$

Ако побараме аритметичка средина на вредностите што се добиени со равенките 2.35b и 2.36, се добива $R_{\text{all}} \approx 0.995759$.

2.2.3 K - терминална надежност

Сосем накратко да кажеме и што значи k - терминална надежност. Значи може да дефинираме многу поопшт концепт каде k - терминали може да бидат поврзани. Во случај кога $k=2$, имаме двотерминална надежност. Ако транспортен систем има n јазли и $k=n$, тогаш имаме комплетно – терминална надежност.

2.2.4 Преглед на истражувањата за бинарна двотерминална надежност

Hansler [5] ја дефинира бинарната двотерминална надежност како веројатноста да постои најмалку еден поврзан пат од изворот до сливникот. Тој опишува алгоритам за симболичка анализа на надежноста на бинарен двотерминален транспортен систем кој се базира на пресметување на веројатноста сите патишта помеѓу изворот и сливникот да се прекинати .

Dotson и Gobien [24] даваат алгоритам за одредување на бинарната двотерминална надежност кој се базира на теоријата на множества. Нивната методологија се потпира на истовремено добивање на сите патишта, сите пресеци и сите одделни успешни и неуспешни состојби. За големи системи, точната вредност на надежноста на системот не може лесно да биде пресметана, но се наоѓаат границите помеѓу кои се наоѓа точната вредност на надежноста.

Yoo и Deo [28] го модифицираат алгоритмот даден од Dotson и Gobien[24]. Нивниот алгоритам овозможува многу поефикасно пресметување на надежноста од претходните дотогаш познати алгоритми. Резултатите што се добиени покажале дека е олеснето пресметувањето на надежноста со примена на овој алгоритам и поради тоа овој алгоритам може да се користи и за поголеми системи.

Kuo et al [20] даваат алгоритам за оценување на двотерминалната надежност кој се базира на дијаграми на проширени рабови (edge expansion diagrams). Надежноста на транспортниот систем се добива со итеративно оценување на дијаграмите претставени во алгоритмот. Овој алгоритам може да се примени на големи системи при што времето за пресметување е прифатливо.

Torrieri [4] претставува начин за пресметување на ненадежните јазли со претпоставка дека може да бидат вградени во систем кој има перфектни јазли. Големи системи може да бидат анализирани и ако точната надежност на системот не може да биде пресметана, може да се пресмета колкава е грешката на оценката за надежноста што е добиена.

Значи во раните 80-ти години, научниците во оваа област дале сугестии дека за некои системи, анализата на надежноста која се базира на чиста поврзаност помеѓу јазлите е несоодветна при разгледување на некои проблеми.

Lee [21] и Aggarwal et al [10-11] го обопштуваат проблемот на двотерминалната надежност. Тие забележуваат дека проблемот не треба да биде само концентриран на поврзаноста, туку надежноста треба да биде оценувана како веројатност за успешно пренесување на потребен број единици од изворот до сливникот, при што треба да се земе во предвид капацитетот на линковите.

Алгоритмот предложен од Lee [21] се базира на концептот на лексикографско подредување на векторите од состојби на системот. Алгоритмот може да биде ефикасно применет само на насочени транспортни системи.

2.3 Надежност на повеќе - состојбени двотерминални транспортни системи

Надежноста на двотерминални транспортни системи е доста проучувана од бинарна перспектива, односно за **бинарни двотерминални транспортни системи**. При моделирање на бинарните двотерминални транспортни системи се претпоставува дека компонентите на системот, и самиот систем може да бидат или во работна состојба, т.е., исправни, или да бидат расипани. Со оваа претпоставка дека системот и неговите компоненти може да бидат само во една од две можни состојби се олеснува пресметувањето на надежноста на системот. Меѓутоа во некои случаи претпоставката дека системот и неговите компоненти може да бидат во една од две можни состојби доведува до грешки во претставувањето на вистинскиот модел на системот. За системи како системи за дистрибуција на вода, телекомуникациски системи, системи за снабдување со гас и нафта, системи за генерирање на енергија од водата и многу други, се покажало дека е погрешно да се претпостави дека овие системи како и нивните компоненти може да се најдат само во една од две можни состојби. Значи, за овие системи бинарниот пристап се покажал како недоволен за добивање на модели на надежност кои ќе го претстават вистинското однесување на системот, односно целосно не го карактеризираат однесувањето на системот. Кај овие системи, нивните компоненти како и самиот систем може да се најдат во една од повеќе состојби. Значи системот и неговите компоненти може да не работат со полн капацитет, но сепак да овозможат прифатливо ниво на опслужување. Главен недостаток на бинарната теорија е што не се зема во предвид дека некои системи како и нивните компоненти може да се најдат во една од повеќе меѓусостојби, туку се претпоставува дека тие може да бидат или во работна или во неисправна состојба. Поради ова може да се појават следните недостатоци:

1. Погрешно моделирање на системот
2. Погрешно оценување на надежноста на системот

Дефиницијата на надежност на бинарен транспортен систем, може да се изрази како веројатност системот успешно да го изврши опслужувањето за кое е наменет под одредени услови. Меѓутоа за некои системи тоа нема да биде точна репрезентација. Овие системи може да не работат со полн капацитет, а сепак да овозможат прифатливо ниво на опслужување.

Поради овие причини, проблемот на двотерминална надежност треба да се разгледува и анализира под претпоставка дека самиот систем како и неговите компоненти може да се најдат во некоја од повеќе меѓусостојби.

Повеќе - состојбената надежност се предлага како комплетна теорија која може да се соочи со проблемите при анализа на системите, таму каде што

традиционалната теорија на надежност станува недоволна. Транспортните двотерминални системи чии што компоненти може да се наоѓаат во некоја од повеќе меѓусостојби се нарекуваат **повеќе - состојбени двотерминални системи**. Значи, сега проблемот за пресметување на двотерминални системи ќе биде формулиран и анализиран под претпоставка дека компонентите на транспортниот систем може да се најдат во една од неколкуте меѓусостојби. Во тој случај и самиот транспортен систем ќе се разгледува од повеќе - состојбена перспектива, бидејќи и тој ќе може да се најде во една од неколкуте меѓусостојби.

2.3.1 Основни дефиниции за повеќе - состојбен двотерминален транспортен систем

Разгледуваме повеќе – состојбен двотерминален транспортен систем со извор s и сливник t . Со \mathcal{N} го означуваме множество од јазли и со $\mathcal{A}=\{a_i | 1 \leq i \leq n\}$ множество од линкови.

Повеќе - состојбен линк се дефинира како линк од мрежата која има множество од состојби $\{r \cdot 100\% | r \in [0,1]\}$. Состојбата 0 соодветствува на случајот кога нема поток низ линкот. Состојбата 100% соодветствува на состојба со која линкот работи со полн капацитет. Меѓусостојбите се сите состојби во кои може да се најде линкот помеѓу состојбата 0 и состојба со која работи со полн капацитет. Векторот што ја прикажува состојбата на компонентите е наречен **вектор на состојби на компонентата (state space set)**. На пример нека линкот од транспортниот систем има три различни состојби: 0%, 50% ,100%. Тогаш векторот (0,0.5,1) е векторот на состојби за тој линк.

За секој повеќе - состојбен линк, **векторот на капацитети (Capacity Vector - Capacity State Set)** е добиен како производ на полниот капацитет на линкот и векторот на состојби за тој линк. Да претпоставиме дека линкот опишан претходно, под перфектни услови може да пренесе поток еднаков на 8 единици. Векторот од капацитети добиен на овој начин ќе биде еднаков на $S_i=(8 \cdot 0, 8 \cdot 0.5, 8 \cdot 1)=(0,4,8)$. За целиот систем се дефинира **множество од капацитети на системот (system capacity state set) S** како множество од сите можни капацитети од изворот до сливникот. Понекогаш, поради некои поедноставувања, состојбите може да бидат нумерирани на различни начини. На пример, може да се претпостави дека перфектната состојба кореспондира со ниво 2, 50% кореспондира со 1, и состојбата на целосно расипување кореспондира со 0, во тој случај ќе се добие векторот (0,1,2).

Нека x_i е состојбата на линкот a_i . Векторот $\vec{x}=(x_1, x_2, \dots, x_n)$ што ги опишува состојбите на сите компоненти на системот се вика **вектор на состојби (state vector)**. Множеството од сите вектори на состојби се нарекува состојба на системот и се означува со E . $E=S_1 \times S_2 \times \dots \times S_{|\mathcal{A}|}$. Функцијата $\varphi: E \rightarrow S$ каде $\varphi(\vec{x})$ е возможниот капацитет од изворот до сливникот, ако системот е во состојба \vec{x} се нарекува **повеќе – состојбена структурна функција**.

Да ја дефинираме сега согласно на дефиниции во подглавјето 1.2, **повеќе – состојбена двотерминална надежност за ниво d** ($M2TR_d$). **Повеќе – состојбената двотерминална надежност за ниво d** ($M2TR_d$) е веројатноста дека поток поголем или еднаков на d , може да биде успешно пренесен од изворот до сливникот.

$$M2TR_d = P(\varphi(\bar{x}) \geq d) \quad (2.37)$$

Дефиниции за пат вектор, пресек вектор, минимален пат вектор и минимален пресек вектор се дадени во поглавјето 1.2.

Со цел да ја специфицираме структурата на транспортниот систем ќе дефинираме бинарен пат вектор. Како што погоре кажавме разгледуваме повеќе – состојбен транспортен систем со множество од јазли \mathcal{N} и множество од линкови $\mathcal{A}=\{a_i \mid 1 \leq i \leq n\}$. Ќе разгледаме транспортен систем со истите јазли и линкови во кој сите линкови се бинарни, т.е. множествата од состојби на линковите се $\{0,1\}$. Нека \bar{v} е минимален пат вектор за овој транспортен систем, тогаш ќе велиме \bar{v} е **бинарен минимален пат вектор** за повеќе состојбениот транспортен систем. Со BPV ќе го означиме множеството од бинарни минимални пат вектори.

Со $TS=(\mathcal{N}, \mathcal{A}, BPV, S, \mathcal{VP})$, да го означуваме транспортниот систем, каде \mathcal{N} е множеството од јазли, \mathcal{A} е множеството од линкови, BPV множеството од бинарни минимални пат вектори, S множеството од вектори на капацитети на компонентите, и \mathcal{VP} множеството од веројатности на нивоата на компонентите, каде \bar{p}_i е вектор од веројатности на i -от линк т.е. $p_{id}=P(x_i=d)$.

2.3.2 Пресметување на надежност

Во овој дел ќе покажеме како се пресметува надежноста на повеќе – состојбен систем, кога минималните пат вектори се познати. За бинарен систем, надежноста може да се пресмета со користење на формулата (1.13). Оваа формула може да биде проширена за новата структура на вектори на минималните множества. За повеќе-состојбен случај $M2TR_d$ може да се добие со следнава модификација на формулата за вклучување и исклучување

$$M2TR_d = \sum_{h=1}^T P(\bar{x} \geq \bar{y}_h) - \sum_{h < k} P(\bar{x} \geq \bar{y}_h \wedge \bar{x} \geq \bar{y}_k) + \dots (-1)^T P(\bar{x} \geq \bar{y}_1 \wedge \dots \wedge \bar{x} \geq \bar{y}_T) \quad (2.38)$$

каде T е бројот од MPV_d (минимални пат вектори за ниво d) и $\bar{y}_h \in MPV_d$. Со користење на следнава нотација:

$$\max(\bar{z}_1, \dots, \bar{z}_s) = (\max(z_1^{(1)}, \dots, z_s^{(1)}), \dots, \max(z_1^{(l)}, \dots, z_s^{(l)})) \quad (2.39)$$

каде $z_u^{(v)}$ е v -та координата на координатата \bar{z}_u , и равенката (2.52) може да биде запишана како:

$$M2TR_d = \sum_{h=1}^T P(\bar{x} \geq \bar{y}_h) - \sum_{h < k}^T P(\bar{x} \geq \max(\bar{y}_h, \bar{y}_k)) + \dots + (-1)^T P(\bar{x} \geq \max(\bar{y}_1, \dots, \bar{y}_T)) \quad (2.40)$$

Во програмскиот јазик C# е направена програма која врз база на најдените пат вектори ја наоѓа надежноста. Програмата се базира на следниот алгоритам.

Алгоритам за пресметување на надежност

Влез на овој алгоритам ќе бидат бинарните минимални пат вектори BPV, множеството од вектори на капацитети на компонентите S , и множеството од веројатности на нивоата на компонентите \mathcal{VP} .

Излез од алгоритмот ќе бидат надежности $M2TR_d$, за $m < d < M$, m е минималното ниво на работа на системот, и M е максималното ниво на работа на системот.

Чекор 1. Се наоѓаат сите минимални пат вектори за ниво d , $\mathcal{D} = \{\bar{z}_1, \dots, \bar{z}_T\}$.

Чекор 2. Се наоѓаат сите можни непразни подмножества од множеството \mathcal{D} .

Чекор 3. За секое подмножество \mathcal{D}_r , $1 \leq r \leq 2^T - 1$ добиено во чекор 2 се наоѓа супремумот, $\bar{v}_r = \sup \mathcal{D}_r$, според подредувањето дадено во Дефиниција 2.10, на тој начин што секоја координата е еднаква на максимумот на соодветните координати на векторите од тоа подмножество.

Чекор 4. Се наоѓаат веројатностите $P(\varphi(\bar{x}) \geq \bar{v}_r)$ и се применува формулата 2.40 за добивање на надежноста $M2TR_d$.

2.3.4 Преглед на истражувањата на повеќе - состојбена двотерминална надежност

Во последно време, повеќе - состојбената надежност на системите е доста проучувана област. Почетното и интуитивно решение на овој проблем се базира на набројување на сите можни состојби во кои може да се најдат компонентите од системот. Врз база на состојбата на компоненти на системот се одредуваат и сите можни состојби во кои може да се најде системот. Според тоа, јасно е дека процедурата со набројување на сите можни состојби (нумерирачката процедура) може да се примени само на релативно мали транспортни системи како што веќе е кажано во Глава 2.1.

Patra и Misra [22-23] предлагаат методологија која ги користи сите пресек множества на бинарниот транспортен систем (под претпоставка дека компонентите и самиот систем може да се најдат во една од две можни

состојби) за добивање на повеќе - состојбената двотерминална надежност за истиот систем. Се претпоставува дека капацитетот на линковите се намалува со тек на време. Надежноста се дефинира како веројатност дека потребното барање (потребниот број на единици - потребниот поток) ќе биде успешно пренесено од изворот до сливникот низ повеќе - состојбените линкови. Со оваа процедура се набројуваат само неколку избрани состојби со кои се оценува надежноста. Бинарните пресеци се користат (за определување) за добивање на векторите на состојби што задоволуваат одредени барања. Тогаш сите непожелни вектори од состојби на системот (тоа се оние вектори од состојби кои не ги задоволуваат барањата за пренесување на поток од изворот до сливникот) се бришат (се отстрануваат). Меѓутоа, пристапот и на оваа методологија е набројувањето на можните состојби, и не може да се специфицира дали се добиваат пат или пресек вектори. Освен тоа, кај некои транспортни системи мора да се врши набројување на сите можни состојби.

Lin [26] развива друга нумеричка методологија со која се добиваат минималните пат множества за повеќе - состојбениот двотерминален транспортен систем. Меѓутоа, овде е потребно компонентите да се слабо хомогени. Тоа значи дека компонентите може да имаат различен број на состојби, но првите j состојби мора да бидат исти. Тогаш за системи чии компоненти се хетерогени оваа методологија не може да овозможи точно пресметување на надежноста. Уште повеќе, ова методологија може да се користи само на мали транспортни системи.

Ramirez-Marquez и Coit [6] даваат алгоритам за точно пресметување на повеќе - состојбената двотерминална надежност. Поточно со овој алгоритам се добиваат сите минимални пат множества, врз основа на кои се пресметува надежноста. Овој алгоритам може да се применува на релативно мали транспортни системи. Алгоритмот ќе биде детално опишан во Глава 3.

Lin [27] и Yeh [25] даваат алгоритми за наоѓање на минималните пресек множества (MMSV) на повеќе - состојбен двотерминален транспортен систем. За секој бинарен пресек, овие алгоритми треба да ги набројат сите различни комбинации од состојби на компонентите во пресекот. Како и да е, и двата може да се применат само за слабо хомогени компоненти со додатни ограничувања дека капацитетот на оската i е целобројна случајна променлива која што може да прими вредности од множеството $\{0,1,2,\dots,M_i\}$. Поради овие барања (ограничувања) овие методи не можат да бидат применети на проблеми од типот на кои методите предложени од Patra и Misra [22-23] можат да бидат применети.

2.4 Наоѓање на бинарни минимални патишта

Поради тоа што во сите алгоритми кои се разгледуваат во оваа магистрска работа, се смета дека се познати минималните патишта или минималните пресеци за бинарен систем, во ова подглавје ќе биде даден алгоритам за наоѓање на бинарните минимални патишта. Алгоритам за пресметување на бинарните минимални пресеци овде нема да биде опишан, имено ако минималните патишта се познати, минималните пресеци може да се добијат со користење на алгоритмот за добивање на минимални пресеци кога се познати минималните патишта даден во Михова [17]. Алгоритам со кој можат да се најдат паралелно и минималните патишта и минималните пресеци може да се најде и во Dotson и Gobien [24].

Алгоритмот што подолу ќе го опишеме ги наоѓа сите патишта во еден граф, преку интензивно пребарување низ сите јазли по длабочина (должина на пат). Алгоритмот се сведува на наоѓање на сите патишта од даден јазол до сливникот со памтење на веќе поминатион пат и избегнување на циклуси. Пред да го опишеме алгоритмот преку чекори мораме да дефинираме што значи еден јазол да е следбеник на друг јазол.

Дефиниција 2.1 Ако t_i и t_j се почетен и краен јазол на еден линк и е возможна комуникација во насока од t_i кон t_j можеме да речеме дека јазолот t_i е предходник на јазолот t_j , а јазолот t_j е следбеник на јазолот t_i .

Алгоритам за наоѓање на бинарни минимални патишта

Влез на овој алгоритам е матрица од нули и единици, од ред $n \times n$, каде n е бројот на јазли во транспортното систем. Доколку постои линк помеѓу два јазли на соодветната позиција во матрицата имаме 1, во спротивно имаме 0.

Како излез се добива бинарен минимален пат претставен како низа од јазли. При што првиот јазол во низата е изворот, последниот е сливникот, а меѓу два соседни јазли во низата постои линк. Излезот од алгоритмот може да се види и во Фигура 2.7.

Чекор 1. За секој јазол се наоѓаат сите јазли следбеници.

Чекор 2. Се иницијализира го досегашниот пат $\mathcal{L} = \emptyset$, јазелот $t_i = s$, множеството патишта $\mathcal{L} = \emptyset$.

Чекор 3. Се наоѓаат сите патишта помеѓу јазелот t_i и сливникот.

Чекор 3.1. Во досегашниот пат \mathcal{L} се додава јазелот t_i .

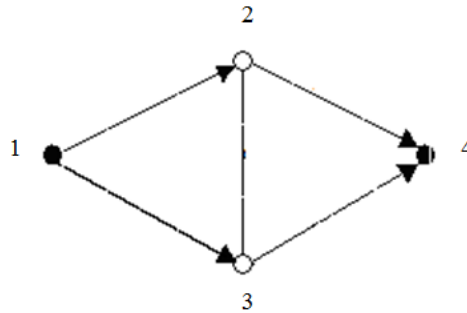
Чекор 3.2. За секој следбеник јазол t_c на јазелот t_i .

Чекор 3.2.1. Ако јазелот t_c е во множеството јазли од досегашниот пат се оди на чекор 3.2 за следниот следбеник.

Чекор 3.2.2. Ако јазелот t_c е сливникот, пронајден е пат и се додава во множеството патишта $\mathcal{L} = \mathcal{L} \cup \{\mathcal{L}\}$ и се оди на чекор 3.2 за следниот следбеник.

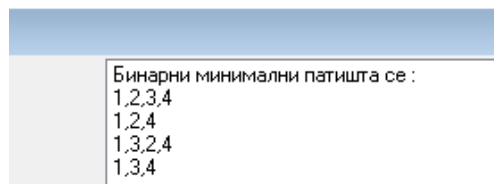
Чекор 3.2.3. Се поставува $t_i = t_c$ и се оди на чекор 3

Овој алгоритам е имплементиран во програмскиот јазик C#. Процедурата `najdi_patista` ги наоѓа сите бинарни минимални патишта. Ако го имаме транспортниот систем даден во Фигура 2.6



Фигура 2.6

Тогаш бинарните минимални патишта се: $\{1,2,4\}$, $\{1,2,3,4\}$, $\{1,3,2,4\}$ и $\{1,3,4\}$. Излезот од алгоритмот е даден на следнава Фигура 2.7



Фигура 2.7 Бинарни минимални патишта

Глава 3

3. Преглед на некои познати алгоритми за определување на минимални пат или минимални пресек множества

3.1 Алгоритам за пресметување на минимални пат множества на Ramirez-Marquez и Coit за транспортни системи со реални капацитети

Во оваа глава ќе го разгледаме алгоритмот за добивање на минимални пат вектори предложен од Jose E. Ramirez-Marquez and David W. Coit [6]. Во овој алгоритам се зема дека транспортниот систем како и неговите компоненти може да се најдат во една од повеќе меѓусостојби, т.е. се разгледува повеќе - состојбен транспортен систем. Со алгоритмот не е потребно да се најдат сите можни комбинации од состојби на линковите, бидејќи наоѓањето на сите можни состојби е скоро невозможно и кога имаме релативно мали транспортни системи. На пример за транспортен систем со 8 линкови, ако секој од линковите може да се најде во една од три можни состојби, вкупниот број на комбинации на состојби на линковите е $3^8=6561$. Значи со овој алгоритам се редуцираат напорите за наоѓање на сите можни комбинации од состојби на линковите. Алгоритмот не бара никакво ограничување на вредности што може да ги примат капацитетите на линковите. Исто така треба да се нагласи дека овој алгоритам може да се примени на релативно мали транспортни системи. Ќе разгледуваме транспортен систем со извор s (source node) и сливник t (sink node). При што, со овој алгоритам се наоѓаат минималните пат вектори со кои може да се пренесе однапред зададен број на единици од изворот до сливникот, значи се наоѓаат минималните пат вектори за однапред дадено ниво. Откако ќе

се најдат сите минимални патишта врз база на нив се пресметува надежноста за даденото ниво.

Потребно е овде да дадеме дефиниција за комбинации од капацитети.

Дефиниција 3.1 (комбинации од капацитети за ниво d) Нека $a \neq b$ се линкови чиј што краен јазол е сливникот, и \bar{x} и \bar{y} се вектори од состојби за a и b соодветно. **Комбинацијата од капацитети за ниво d** се дефинира како:

$$(x_i, y_i) = \{ x_i \in \bar{x}, y_j \in \bar{y}; x_i + y_j \geq d \}, \forall i, j = 1, 2, \dots, n$$

Во овој дел прво ќе го опишеме алгоритмот за наоѓање на минималните пат вектори, а потоа ќе дадеме еден едноставен пример.

3.1.1 Опис на алгоритмот за пресметување на минимални пат множества

Овој алгоритам за наоѓање на минималните пат вектори започнува со тоа што се наоѓаат линковите чиј што краен јазол е сливникот, а потоа се наоѓаат сите можни комбинации од вектори на капацитети за овие линкови. За секоја од овие комбинации се добива нов транспортен систем со нов сливник и истата процедура се повторува од сливникот до изворот.

Да ги дадеме чекорите од кои се состои овој алгоритам, за да може полесно да се разбере.

Алгоритам за пресметување на минимални пат множества на Ramirez-Marquez и Coit

Влез на овој алгоритам ќе бидат бинарните минимални пат вектори BPV , множеството од вектори на капацитети на компонентите S , и множеството од веројатности на нивоата на компонентите \mathcal{VP} и нивото d .

Излез од алгоритмот ќе бидат минималните пат вектори за ниво d .

Чекор 1. Се наоѓаат сите линкови кои за краен јазол го имаат сливникот и нивните соодветни вектори од капацитети.

Чекор 1.1. За сите линкови чиј што краен јазол е сливникот се наоѓаат сите можни комбинации на вектори од капацитети за ниво d . Се користи Дефиниција 2.7 за да се отстранат (избришат) оние комбинации на вектори од капацитети за ниво d што поголеми од други комбинации на вектори од капацитети.

Чекор 2. Се нумерираат сите комбинации на вектори од капацитети добиени во чекорот 1.1 од 1 до n . Секоја комбинација дава нов транспортен систем (подсистем од првиот систем) со нови барања што треба да бидат задоволени.

Ако $n=0$, тогаш надежноста за ниво d , $R(d)=0$, т.е. не постои комбинација од линкови со кои може да се пренесат d единици до сливникот. Значи барањето за ниво d , не може да биде исполнето.

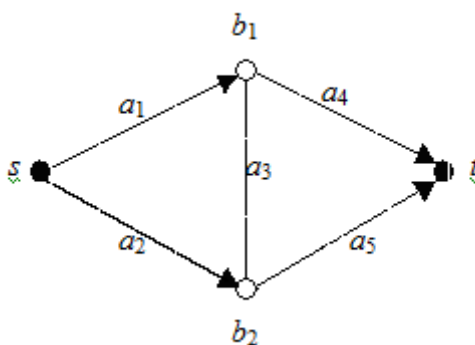
Чекор 3. За секоја комбинација се креира нов транспортен систем кој што ќе има и нов сливник (краен јазол). И за секој од овие нови транспортни системи (т.е. подсистеми), се оди на чекор 1.

Чекор 4. Се повторува алгоритмот почнувајќи од сливникот до изворот.

Алгоритмот ќе го илустрираме со пример.

Пример 3.1

Алгоритмот ќе го илустрираме на примерот што е даден со Фигура 3.1. Тоа е едноставен транспортен систем кој се состои од 4 јазли (s, t, b_1 и b_2) и 5 линкови (a_1, a_2, a_3, a_4 и a_5)



Фигура 3.1

Потребно е да се испратат (пренесат) 4 единици од изворот до сливникот. Се претпоставува дека секој линк може да се најде во една од следниве три можни состојби: 0 (не може да се пренесе поток низ тој линк), 50% (линкот работи со пола од полниот капацитет) и 100% (линкот работи со полн капацитет). Капацитетите на линковите се дадени во следнава табела:

Линк	Вектори од капацитети
a_1	(0,1,5,3)
a_2	(0,2,4)
a_3	(0,1,2)
a_4	(0,1,2)
a_5	(0,2,4)

Табела 3.1 : Вектори на капацитети за линковите

Во Табела 3.2 се дадени веројатностите со кои линковите може да се најдат во некоја од трите можни состојби.

Линкови	Состојби		
	0%(0)	50%(0.5)	100%(1)
a_1	0.01	0.09	0.9
a_2	0.01	0.09	0.9
a_3	0.05	0.25	0.7
a_4	0.08	0.12	0.8
a_5	0.05	0.05	0.9

Табела 3.2 Веројатности со кои линковите се наоѓаат во некоја состојба

Вкупниот број на комбинации на состојби на линковите за овој едноставен транспортен систем ќе биде еднаков на $3^5=243$. Од сите овие комбинации треба да се најдат само оние со кои може да се пренесат 4 единици од изворот s до сливникот t . Со примена на овој алгоритам значајно се намалува бројот на комбинации што се разгледуваат. Сега ќе го илустрираме алгоритмот така што ќе ги опишеме првите неколку чекори:

Чекор 1. Линковите a_2 и a_5 за краен јазол го имаат сливникот t .

Векторот од капацитети за линкот a_4 е $(0,1,2)$ додека за линкот a_5 е $(0,2,4)$

Чекор 1.1. Во Табела 3.3 се дадени сите можни комбинации на вектори од капацитети за линковите a_4 и a_5 , за ниво 4 (односно со кои може да се пренесат 4 единици до сливникот)

a_5	a_4
4	0
4	1
4	2
2	2

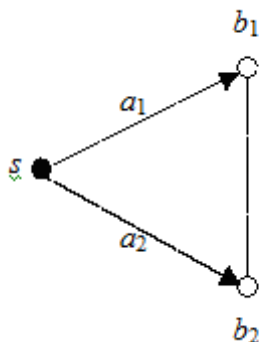
Табела 3.3

Според алгоритмот, бидејќи комбинацијата за која $a_5=4$ и $a_4=0$, $(0,4)$ е доминирана од комбинациите за кои $a_5=4$ и $a_4=1$, и $a_5=4$ и $a_4=2$, $(1,4)$ и $(2,4)$, комбинациите $(1,4)$ и $(2,4)$ се отстрануваат. Значи ги имаме следниве две комбинации, T_1 и T_2 .

Комбинација T_1		Комбинација T_2	
$a_5=4$	Доминанти се сите комбинации за кои $a_4>0$	$a_5=2$	Доминанти се сите комбинации за кои $a_5>2$
$a_4=0$		$a_4=2$	

Чекор2. Секоја комбинација дава нов транспортен систем со нови барања. Комбинациите T_1 и T_2 се разгледуваат одделно.

Прво да ја разгледаме комбинацијата T_1 , во овој случај се добива нов сливник, тоа е јазолот b_2 . До овој јазол треба да бидат пренесени 4 единици. Новиот транспортен систем е прикажан на Фигура 3.2



Фигура 3.2. Нов транспортен систем, добиен за комбинацијата T_1

Чекор 3. Линкови чиј што краен јазол е јазолот b_2 се линкот a_2 со вектор од капацитети $(0,2,4)$ и a_3 со вектор од капацитети $(0,1,2)$. За овие линкови се добиваат следниве комбинации:

Комбинација B_{21}		Комбинација B_{22}	
$a_2=4$	Доминанти се сите комбинации за кои $a_3>0$	$a_2=2$	Доминанти се сите комбинации за кои $a_2>2$
$a_3=0$		$a_3=2$	

Комбинацијата B_{21} го дава првото решение на проблемот бидејќи линкот a_2 е директно поврзан со изворниот јазол. $S_1=(0,4,0,0,4)$.

За комбинацијата B_{22} , јазолот b_1 станува сливник, со барање од 2 единици. Значи $a_1=3$ (можните состојби на a_1 се 3,1.5 и 0) Друго решение е $S_2=(3,2,2,0,4)$. Со ова се добиени сите комбинации што произлегуваат од комбинацијата T_1 .

Да ја примениме сега истата процедура за комбинација T_2 . За оваа комбинација имаме дека $a_4=2$ и $a_5=2$. Сега сливник јазол ни е јазолот b_1 со барање од 2 единици. Од транспортниот систем даден со Фигура 3.2, гледаме дека линкови чиј што краен јазол е јазолот b_1 се a_1 и a_3 . Векторот од капацитети за линкот a_1 е $(0,1.5,3)$ и за линкот a_3 е $(0,1,2)$. При што ги добиваме следниве комбинации:

Комбинација B_{11}		Комбинација B_{12}		Комбинација B_{13}	
$a_1=3$	Доминанти се сите комбинации за кои $a_3>0$	$a_1=0$	Доминанти се сите комбинации за кои $a_1>0$	$a_1=1.5$	Доминанти се сите комбинации за кои $a_1>1.5$
$a_3=0$		$a_3=2$		$a_3=1$	

За комбинацијата V_{11} , јазолот b_2 станува сливник, со барање од 2 единици. Значи $a_2=2$. Решението $S_3=(3,2,0,2,2)$.

За комбинацијата V_{12} , јазолот b_2 станува сливник, со барање од 4 единици. Значи $a_2=4$. Решението $S_4=(0,4,2,2,2)$.

За комбинацијата V_{13} , јазолот b_2 станува сливник, со барање од 4 единици. Значи $a_2=4$. Решението $S_5=(1.5,4,1,2,2)$.

Значи со примена на овој алгоритам ние добиваме 5 пат вектори за ниво 4.

Минималните пат вектори дадени во [5] се претставени во Табела 3.4.

Минимални пат вектори за ниво 4	$M2TR_4$
0 4 0 0 4	0.919683
3 2 2 0 4	
3 2 1 1 4	
0 4 2 2 2	
1.5 4 1 2 2	
3 2 0 2 2	

Табела 3.4

Може да забележиме дека овде се добиени 6 минимални пат вектори за ниво 4, едниот од тие вектори е $(3,2,1,1,4)$. За овој вектор $a_4=1$ и $a_5=4$, меѓутоа оваа комбинација беше отфрлена уште во чекор еден, како комбинација што е доминанта над комбинацијата за која $a_4=0$ и $a_5=4$.

Откако се добиени минималните пат вектори, пресметување на надежноста за ниво 4 се врши со примена на формулата за вклучување и исклучување како што е објаснето во претходната глава. Надежност за ниво 4, $M2TR_4=0.919683$. Направен е програм во C# со кој доколку се најдени минималните пат вектори може да се пресмета надежноста. Ова може да се види на Фигура 3.4:

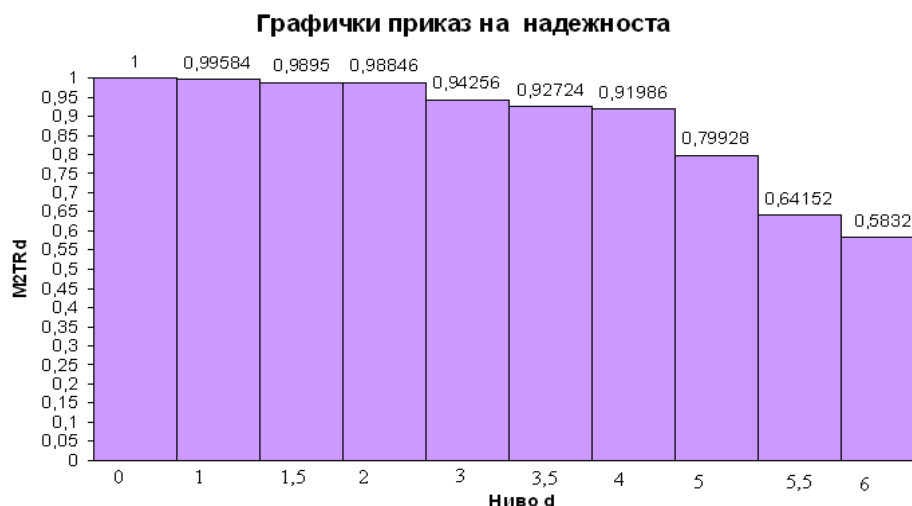
```

Minimalni pat vektori za nivo 4,0 se
(3,2,1,1,4)
(3,2,0,2,2)
(1,5,4,1,2,2)
(3,2,2,0,4)
(0,4,2,2,2)
(0,4,0,0,4)
M2TR4: 0,91968
=====
    
```

Фигура 3.3 Минимални пат вектори за ниво 4 и надежност за ниво 4

Во Прилог А, во Табела 2, страна 99 се дадени минималните пат вектори што се добиени со примена на овој алгоритам дадени во трудот [6]. Во наредната глава каде што ќе вршиме споредба на овој алгоритам и алгоритмите дадени во Глава 4, ќе забележиме дека со примена на овој алгоритам не се добиваат сите минимални пат вектори.

Во Фигура 3.5 графички е претставена надежноста за сите нивоа d .



Фигура 3.4 Графички приказ на надежноста

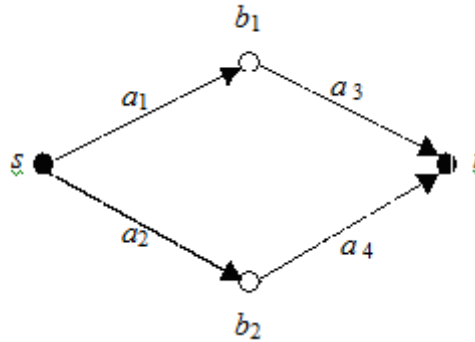
3.1.2 Анализа на алгоритмот

Во овој дел ќе направиме анализа на алгоритмот даден од Jose E. Ramirez-Marquez and David W. Coit [6]. Како што веќе кажавме овој алгоритам ја зема во предвид повеќе – состојбената природа како на транспортниот систем така и на неговите компоненти. Во [6] е нагласено дека алгоритмот се базира на пресметување на надежноста за дадено ниво кога предходно се познати минималните пат множество. Исто така е нагласено дека тој ефикасно ја пресметува надежноста за дадено ниво d , и со него се намалени напорите за наоѓање на сите можни комбинации на состојби на компонентите. Предност на овој алгоритам се состои во тоа што тој не бара никакви ограничувања на капацитетите на линковите. Меѓутоа како што видовме од предходниот пример со овој алгоритам не се наоѓаат сите минимални пат вектори за дадено ниво. Во [6] детално е опишано како се наоѓаат минималните пат вектори за ниво 4. Но со примена на Чекор 1, од овој алгоритам кога се разгледуваат само оние линкови кои за крајна дестинација го имаат сливник (линковите a_3 и a_3), пат векторот (1,4) не е минимален пат вектор за транспортниот подсистем кој се состои овие линкови. Проблемот е во тоа што минимален пат вектор за ниво 4, е пат векторот (3,2,1,1,4).

. Треба да се забележи дека проблемот се јавува уште во Чекор 1 кога се отстрануваат комбинации од капацитети што се доминантни над други комбинации на капацитети. Што значи дека овде прво се наоѓаат минималните пат вектори за транспортниот подсистем. Проблемот е во тоа што не сите минимални пат вектори на целиот транспортен систем, се минимални пат вектори за транспортните подсистемите. Како што видовме во примерот

$(3,2,1,1,4)$ е минимален пат вектор за ниво 4 за целиот транспортен систем, но $(1,4)$ не е минимален пат вектор за транспортниот подсистем што се состои од линковите кои за крајна дестинација го имаат сливникот.

Да разгледаме дека уште на еден многу едноставен транспортен систем дека алгоритмот не функционира добро.



Фигура 3.5

Нека $(0,1,1.5)$, $(0,0.5,1)$, $(0,1,2)$ и $(0,1)$ се вектори од капацитети за линковите a_1 , a_2 , a_3 и a_4 соодветно. Сите можни комбинации на вектори од капацитети за линковите a_3 и a_4 , за ниво 2 се $(2,0)$, $(1,1)$ и $(2,1)$. Според чекор 1, од алгоритмот бидејќи векторот $(2,1)$ е поголем од векторите $(2,0)$ и $(1,1)$ овој вектор нема да го земеме во предвид. Од друга страна ако го разгледаме целиот транспортен систем вектор $(1.5,0.5,2,1)$ е минимален пат вектор за ниво 2. Значи и од овде можеме повторно да заклучиме дека минималните пат вектори за целиот систем, не мора да бидат минимални пат вектори и за подсистемите.

Исто така да кажеме дека со овој алгоритам се наоѓаат пат векторите за однапред зададено ниво, што значи дека не можеме да знаеме кое е максималното ниво со кое системот може да работи.

3.2 Алгоритам за пресметување на минимални пресек множества на Ramirez-Marquez, Coit и Tortorella за транспортни системи со слабо хомогени компоненти

Ќе го разгледаме алгоритмот за добивање на минимални пресек множества предложен од Jose E. Ramirez-Marquez, David W. Coit, and Michael Tortorella [9]

Во овој дел се разгледува методологија за добивање на минимални пресек вектори на повеќе - состојбен двотерминален транспортен систем. Алгоритмот го користи пристапот на размена на информации со што значајно се намалува бројот на вектори кои е потребно да се набројат (нумерираат) за да се добијат сите минимални пресек вектори на повеќе - состојбениот транспортен систем. Алгоритмот работи на принципот така што се селектира одредено множество од минимални пресек вектори наречено “потомок пресек” (“offspring cut”), кое наследува информации од множество од минималните пресек вектори наречено “родителски пресек” (“parent cuts”).

За големи транспортни системи, дури и кога компонентите можат да се најдат во релативно мал број различни состојби, наоѓањето на минималните пресек вектори може да биде доста комплексно. Исто така треба да се нагласи дека иако овој алгоритам нема ограничување за вредностите на капацитетите на линковите, сепак капацитетите на линковите треба да се трансформираат до слабо хомогени за да може алгоритмот да работи. Значи алгоритам работи со слабо хомогени компоненти, и откако ќе бидат најдени минималните пат вектори, ги трансформираме во хетероген облик.

3.2.1 Опис на проблемот

Го разгледуваме транспортниот систем $G=(\mathcal{N}, \mathcal{A})$, каде \mathcal{N} е множеството од јазли, \mathcal{A} е множеството од линкови.

Пред да го разгледаме алгоритмот за наоѓање на минималните пресек вектори мораме да дадеме некои претпоставки кои се однесуваат на одредени ограничувања во врска со повеќе - состојбениот двотерминален транспортен систем.

Претпоставки:

1. Состојбите на компонентите се меѓусебно стохастички независни.
2. Структурната функција $\varphi(\vec{x})$ е кохерентна. Тоа значи дека со зголемувањето на надежноста на некоја компонента од транспортниот систем не може да се предизвика намалување на надежноста на целиот транспортен систем.

3. Состојбите во кои може да се најдат компонентите и нивните соодветни веројатности се познати.
4. G нема циклуси.

За да се опише алгоритмот потребно е да се дефинираат нови термини кои се јавуваат кај транспортните системи

Дефиниција 3.2 (Слабо хомогени компоненти) Нека $S_i = (S_{i,1}, S_{i,2}, \dots, S_{iM_i})$ и $S_j = (S_{j,1}, S_{j,2}, \dots, S_{jM_j})$ се вектори од сите можни состојби на линковите a_i и a_j соодветно. Без губење на општост можеме да претпоставиме дека $M_i < M_j$. Линковите a_i и a_j се нарекуваат **слабо хомогени компоненти** ако $S_{ik} = S_{jk}, \forall k \leq M_i$.

Дефиниција 3.3 Потомок на пресек (Offspring of a Cut) Нека C_i и C_j се две пресек множества на дадениот транспортен систем. Нека $\mathcal{T} = \{a_k | (a_k \in C_j) \wedge (a_k \notin C_i)\}$ и $\mathcal{T}_1 = \{a_k | (a_k \notin C_j) \wedge (a_k \in C_i)\}$. C_j се нарекува потомок на C_i , ако $a_s \propto a_k, \forall a_s \in \mathcal{T}; a_k \in \mathcal{T}_1$.

($\propto \equiv$ претходи. Значи, компонентата a_s е претходник на компонентата a_k во транспортниот систем G .)

3.2.2 Опис на алгоритмот за пресметување на минимални пресек множества

Мотивацијата за предложување на овој алгоритам е двострана. Како прво, минималните пресек вектори кои се добиени за бинарниот двотерминален транспортен систем се пресек вектори и за повеќе - состојбениот транспортен систем што има иста структура како и бинарниот, меѓутоа овие пресек вектори не мора да бидат минимални. Значи проблемот во повеќе - состојбениот случај се сведува на наоѓањето на минималните состојби на компонентите во бинарниот пресек.

Второ, за транспортните системи за кои се претпоставува дека немаат циклуси, информацијата содржана во пресеците значително го намалува бројот на потребните нумерирања за добивање на минималните пресек вектори за повеќе - состојбениот транспортен систем. Базирајќи се на бинарните минимални пресек множества, претходно познатите алгоритми користат детално нумерирање за наоѓање на новите состојби на компонентите за дадениот бинарен пресек што може да продуцира грешка во повеќе - состојбените случаи. Алтернативно, овој алгоритам, иницијално идентификува два основни пресеци, “родителски пресеци”, од кои се проследува информација (состојби на компонентите) кон одреден број бинарни пресеци наречени “потомци пресеци”.

Почетниот чекор на овој алгоритмот бара бинарните минимални пресек множества да бидат однапред познати.

Од самата дефиниција за минимални пресек множества, следува дека множеството од линкови чиј што почетен јазол е изворниот јазол и множеството од линкови чиј што краен јазол е сливникот се минимални пресек множества. Ова се користи во Чекор 3 од алгоритмот за да се зададе почетна точка во работата на алгоритмот. Исто така да нагласиме дека множеството Ω' е комплементарно множество на Ω , во кое што ќе ги сместуваме сите бинарни пресеци кои се веќе обработени од алгоритмот. Да го опишеме сега детално алгоритмот преку неговите чекори:

Алгоритам за наоѓање на минимални пресек множества даден од Ramirez-Marquez, Coit и Tortorella

Влез на овој алгоритам ќе бидат бинарните минимални пресек вектори , множеството од вектори на капацитети на компонентите S , и множеството од веројатности на нивоата на компонентите \mathcal{VP} и нивота d .

Излез од алгоритмот ќе бидат минималните пресек вектори за ниво d .

Чекор 1. Се трансформираат сите вектори од капацитети на линковите во слабо хомогени.

Чекор 2. Се наоѓаат бинарните пресек множества $\Omega = \left\{ \bigcup_{l=1}^m C_l \right\}$ каде C_l е некој

бинарен пресек на транспортниот систем G .

Чекор 3. Се формира табела од претходници за сите линкови од транспортниот систем.

Чекор 4. Се формира множеството од бинарните “родителски” пресек вектори, $\Omega' = \{C_1, C_m\}$ каде $C_1 = \{a_i | a_i \text{ е линк чиј што краен јазол е сливникот} \}$ и $C_m = \{a_i | a_i \text{ е линк чиј што почетен јазол е изворот} \}$

Чекор 5. Се формира множеството од повеќе – состојбени пресек вектори за “родителските” множества C_1 и C_m на следниот начин:

Се земаат сите вектори на капацитетите, такви што за линковите кои не припаѓаат во родителското множество се земаат максималните капацитети, а за линковите во множеството се земаат сите можни вредности, така што збирот на нивните капацитети да е стриктно помал од нивото d за кое го анализираме системот

$$H_z = \sum_k \sum_{s=1}^{|\mathcal{T}|} S'_{kjs} \quad \forall j_s = 1, \dots, M_i, S_{kjs} < d$$

$$a_k \in \mathcal{T}$$

$$x'_i = \begin{cases} S_{ij_s} & \forall a_i \in \mathcal{T} \\ M_i & \text{инаку} \end{cases}$$

$$V_t = \bigcup_{i=1}^m \{x'_i\}$$

Од вака добиеното множество на вектори на капацитет, ги отстрануваме оние кои се помали од друг вектор од истото множество. Овие вектори на капацитет се запаметуваат и во множеството \mathcal{G} од кандидати за решенија на алгоритмот.

Чекор 6. Додека $\Omega \neq \emptyset$

Чекор 6.1. Од множеството Ω најди наследник C_1 (според дефиниција 3.3) на множество C_m од множеството Ω

Чекор 6.2. Најди го множеството \mathcal{T} кое се состои од сите елементи од C_1 кои не се во C_m , и множеството \mathcal{T}_1 кое се состои од сите елементи од C_m кои не се во C_1

$$\mathcal{T} = \{a_i | (a_i \in C_1) \wedge (a_i \notin C_m)\} \text{ и } \mathcal{T}_1 = \{a_i | (a_i \notin C_1) \wedge (a_i \in C_m)\}$$

Чекор 6.3. За секој елемент \bar{x} од множеството од повеќе – состојбени пресек вектори за C_1 :

Чекор 6.3.1. Се пресметува w како максималната вредност на елементите од \bar{x} чишто линкови припаѓаат во \mathcal{T}_1

$$w = \max_i \{x'_i | a_i \in \mathcal{T}_1; x'_i \in \bar{x}\}$$

Чекор 6.3.2. Се земаат сите вектори на капацитетите, такви што за линковите кои не припаѓаат ниту во \mathcal{T} ниту во \mathcal{T}_1 се зема вредноста од \bar{x} , за оние кои припаѓаат во \mathcal{T}_1 се земаат максималните капацитети, а за линковите од множеството \mathcal{T} се земаат сите можни вредности, така што збирот на нивните капацитети да е помал или еднаков на w .

$$H_z = \sum_{a_k \in \mathcal{T}} \sum_{s=1}^{|\mathcal{T}|} S'_{kj_s} \quad \forall j_s = 1, \dots, M_i; S'_{kj_s} \leq w$$

$$x'_i \rightarrow \begin{cases} S'_{kj_s} & \forall a_i \in \mathcal{T} \\ M_i & \forall a_i \in \mathcal{T}_1 \quad \forall j_s; x'_i \in \bar{x} \\ x'_i & \text{инаку} \end{cases}$$

$$V_t \rightarrow \bigcup_{i=1}^m \{x'_i\}$$

Чекор 6.3.3. Од вака добиеното множество на вектори на капацитет, ги отстрануваме оние кои се помали од друг вектор од истото множество.

Чекор 6.3.4. Од вака добиеното множество на вектори на капацитет, ги отстрануваме оние кои се помали од друг вектор од истото множество. Овие вектори на капацитет се запаметуваат и во множеството \mathcal{G} од кандидати за решенија на алгоритмот.

Чекор 6.4. Множеството C_i го преместуваме од Ω во Ω' .

Чекор 7. Се трансформираат кандидат решенијата во хетероген облик на следниот начин: за секое кандидат решение

Чекор 7.1. Доколку x_i одговара на состојба за линкот a_i , преземи го како такво.

Чекор 7.2. Инаку за вредност на x_i земи ја наредната помала валидна состојба.

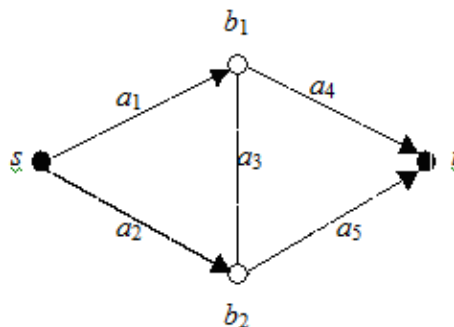
Чекор 8. Од вака добиеното множество на кандидат решенија, ги отстрануваме оние кои се помали од друг вектор од истото множество.

Алгоритмот ќе го илустрираме со пример.

Пример 3.2

Алгоритмот ќе го илустрираме повторно на транспортниот систем даден во Фигура 3.1. Тоа е едноставен транспортен систем кој се состои од 4 јазли (s, t, b_1 и b_2) и 5 линкови (a_1, a_2, a_3, a_4 и a_5).

Овде ќе ги најдеме сите минимални пресек вектори, така што најмногу 2 единици ќе може да се пренесат од изворот до сливникот (значи барање од 3 или повеќе единици не може да биде пренесено од изворот до сливникот) т.е. $d=3$.



Фигура 3.6

Во табелата 3.4 се дадени вектори од состојби на линковите:

Линк	Вектори од состојби
a_1	(0,1,2,3)
a_2	(0,1)
a_3	(0,1)
a_4	(0,1,2)
a_5	(0,1,2)

Табела 3.4 : Вектори од состојби за линковите

Чекор 1. Векторите од капацитети на компонентите се слабо хомогени

Чекор 2. Бинарните пресек множества за овој транспортен систем се:

$$\{a_4, a_5\}, \{a_1, a_3, a_5\}, \{a_2, a_3, a_4\}, \{a_1, a_2\}$$

$$\text{Значи } \Omega = \{ \{a_4, a_5\}, \{a_1, a_3, a_5\}, \{a_2, a_3, a_4\}, \{a_1, a_2\} \}$$

Чекор 3. Се формира табела од претходници на линковите од транспортниот систем

Линк	Претходници
a_1	\emptyset
a_2	\emptyset
a_3	a_1, a_2
a_4	a_1, a_3
a_5	a_2, a_3

Табела 3.5 : Табела од претходници за линковите

Чекор 4. Се формираат множествата од бинарни “родителски” пресек вектори C_1 и C_4

Множеството $C_1 = \{a_4, a_5\}$ бидејќи линковите a_4 и a_5 за краен јазол го имаат сливникот.

Множеството $C_4 = \{a_1, a_2\}$ бидејќи линковите a_1 и a_2 почетен јазол го имаат изворот.

Чекор 5. Се формира множеството од повеќе – состојбени пресек вектори за “родителските” множества C_1 и C_4 на следниот начин:

1. Множество C_1 :

За линковите a_1, a_2, a_3 што не припаѓаат во C_1 се земаат максималните состојби додека за a_4, a_5 што припаѓаат во C_1 се земаат сите можни состојби чиј што збир е стриктно помал од 3. Кандидатите за пресек вектори се дадени во Табела 3.6

\bar{x}'_i	x_1	x_2	x_3	x_4	x_5
1	3	1	1	0	0
2	3	1	1	0	1
3	3	1	1	0	2
4	3	1	1	1	0
5	3	1	1	1	1
6	3	1	1	2	0

Табела 3.6

Од вака добиеното множество на вектори, ги отстрануваме оние кои се помали од друг вектор од истото множество. Од Табела 3.6 може да се забележи дека за векторите \vec{x}'_1, \vec{x}'_2 и \vec{x}'_4 постојат поголеми вектори. Па овие вектори ги отстрануваме. Кандидатите пресек вектори се дадени во Табела 3.7

\vec{x}'	x_1	x_2	x_3	x_4	x_5
\vec{x}'_3	3	1	1	0	2
\vec{x}'_5	3	1	1	1	1
\vec{x}'_6	3	1	1	2	0

Табела 3.7

Вектори се ставаат во множеството $\mathcal{G} = \{ \vec{x}'_3, \vec{x}'_5, \vec{x}'_6 \}$

2. Множество C_4

За линковите a_3, a_4, a_5 што не припаѓаат во C_4 се земаат само максималните состојби додека за a_1, a_2 што припаѓаат во C_4 се земаат сите можни состојби чиј што збир е стриктно помал од 3. Резултатите се дадени во Табела 3.8

\vec{x}'_k	x_1	x_2	x_3	x_4	x_5
7	0	0	1	2	2
8	0	1	1	2	2
9	1	0	1	2	2
10	1	1	1	2	2
11	2	0	1	2	2

Табела 3.8

Од вака добиеното множество на вектори, ги отстрануваме оние кои се помали од друг вектор од истото множество. Од Табела 3.8 може да се забележи дека за векторите \vec{x}'_7, \vec{x}'_8 и \vec{x}'_9 постојат поголеми вектори. Па овие вектори ги отстрануваме. Кандидатите пресек вектори се дадени во Табела 3.9

\vec{x}'	x_1	x_2	x_3	x_4	x_5
\vec{x}'_{10}	1	1	1	2	2
\vec{x}'_{11}	2	0	1	2	2

Табела 3.9

$\mathcal{G} = \{ \vec{x}'_3, \vec{x}'_5, \vec{x}'_6, \vec{x}'_{10}, \vec{x}'_{11} \}$

Чекор 6.

Чекор 6.1. Од множеството Ω се избира “потомок” на множеството $C_1=\{a_4, a_5\}$, тоа е множеството $C_2=\{a_1, a_3, a_5\}$.

Чекор 6.2. Од дефиницијата за “потомок”, се наоѓаат множествата \mathcal{T} и \mathcal{T}_1 .
 $\mathcal{T}=\{a_k | (a_k \in C_2) \wedge (a_k \notin C_1)\}$ и $\mathcal{T}_1=\{a_k | (a_k \in C_1) \wedge (a_k \notin C_2)\}$. Значи
 $\mathcal{T}=\{a_1, a_3\}$ и $\mathcal{T}_1=\{a_4\}$ (важи $a_s \propto a_k, \forall a_s \in \mathcal{T}; a_k \in \mathcal{T}_1$).

Чекор 6.3: За секој $\vec{x}' \in \{\vec{x}'_3, \vec{x}'_5, \vec{x}'_6\}$

Чекор 6.3.1 Се наоѓа $w = \max_i \{x'_i | a_i \in \mathcal{T}_1; x'_i \in \vec{x}'\}$

За $\vec{x}'_3, w=0$, бидејќи $\mathcal{T}_1=\{a_4\}$, а во \vec{x}'_3 четвртата компонента е 0. За елементите од \mathcal{T}, a_1 и a_3 се земаат сите можни состојби така што нивниот збир да не е поголем од 0. (значи $x_1=x_3=0$). За елементот $a_4 \in \mathcal{T}_1$ (и за a_2, a_5) се зема максималната состојба, $x_4=2$. А за елементи a_2, a_5 што не припаѓаат ниту во \mathcal{T} ниту во \mathcal{T}_1, x_2 и x_5 ги имаат истите вредности како и во \vec{x}'_3 т.е. $x_2=1, x_5=2$. На ист начин ќе ги добиеме и состојбите за \vec{x}'_5 и \vec{x}'_6 . Резултатите ќе бидат дадени во Табела 3.10.

\vec{x}'_k	x_1	x_2	x_3	x_4	x_5
За $\vec{x}'_3: w=0$					
13	0	1	0	2	2
За $\vec{x}'_5: w=1$					
14	0	1	0	2	1
15	0	1	1	2	1
16	1	1	0	2	1
За $\vec{x}'_6: w=2$					
17	0	1	0	2	0
18	0	1	1	2	0
19	1	1	0	2	0
20	1	1	1	2	0
21	2	1	0	2	0

Табела 3.10

Во Табела 3.11 се дадени вектори од Табела 3.10 со изоставување на вектори за кои постои вектор што е поголем од нив

\bar{x}'	x_1	x_2	x_3	x_4	x_5
\bar{x}'_{13}	0	1	0	2	2
\bar{x}'_{15}	0	1	1	2	1
\bar{x}'_{16}	1	1	0	2	1
\bar{x}'_{20}	1	1	1	2	0
\bar{x}'_{21}	2	1	0	2	0

Табела 3.11

Значи $G = \{ \bar{x}'_3, \bar{x}'_5, \bar{x}'_6, \bar{x}'_{10}, \bar{x}'_{11}, \bar{x}'_{13}, \bar{x}'_{15}, \bar{x}'_{16}, \bar{x}'_{20}, \bar{x}'_{21} \}$,

$$\Omega = \Omega' \cup C_2, \Omega' = \{ \{a_4, a_5\}, \{a_1, a_2\}, \{a_1, a_3, a_5\} \}$$

$\Omega = (\Omega \cap \Omega')^c = \{ a_4, a_3, a_2 \}$, Значи $\Omega \neq \emptyset$ па повторно се враќаме на Чекор 6.1

Чекор 6.1: Од множеството Ω се избира “потомок” на множеството

$$C_1 = \{a_4, a_5\}, \text{ тоа е множеството } C_3 = \{a_4, a_3, a_2\}.$$

Чекор 6.2: Од дефиницијата за “потомок”, се наоѓаат множествата \mathcal{T} и \mathcal{T}_1 .

$$\mathcal{T} = \{ a_k \mid (a_k \in C_2) \wedge (a_k \notin C_1) \} \text{ и } \mathcal{T}_1 = \{ a_k \mid (a_k \in C_1) \wedge (a_k \notin C_2) \}. \text{ Значи } \\ \mathcal{T} = \{ a_3, a_2 \} \text{ и } \mathcal{T}_1 = \{ a_5 \} \text{ (важи } a_s \propto a_k, \forall a_s \in \mathcal{T}; a_k \in \mathcal{T}_1).$$

Чекор 6.3: За секој $\bar{x} \in \{ \bar{x}'_3, \bar{x}'_5, \bar{x}'_6 \}$.

Чекор 6.3.1 Се наоѓа $w = \max_i \{ x'_i \mid a_i \in \mathcal{T}_1; x'_i \in \bar{x} \}$

За \bar{x}'_3 , $w=2$, бидејќи $\mathcal{T}_1 = \{ a_5 \}$, а во \bar{x}'_3 петта компонента е 2.

За елементите од \mathcal{T} , a_3 и a_2 се земаат сите можни состојби така што нивниот збир да не е поголем од 2, и $x_2 \leq 1$, $x_3 \leq 1$.

Ги имаме следниве комбинации: $x_2 = x_3 = 0$, $x_2 = 1$, $x_3 = 0$, $x_2 = 0$, $x_3 = 1$ и $x_2 = x_3 = 1$. За елементот $a_5 \in \mathcal{T}_1$ се зема максималната состојба, $x_5 = 2$.

А за елементи a_1, a_4 што не припаѓаат ниту во \mathcal{T} ниту во \mathcal{T}_1 , x_1 и x_4 ги имаат истите вредности како и во \bar{x}'_3

т.е. $x_1 = 3$, $x_4 = 0$. На ист начин ќе ги добиеме и состојбите за \bar{x}'_5 и \bar{x}'_6 . Резултатите ќе бидат дадени во Табела 3.12.

\vec{x}'_k	x_1	x_2	x_3	x_4	x_5
За $\vec{x}'_6 : w=0$					
22	3	0	0	2	2
За $\vec{x}'_5 : w=1$					
23	3	0	0	1	2
24	3	1	0	1	2
25	3	0	1	1	2
За $\vec{x}'_3 : w=2$					
26	3	0	0	0	2
27	3	0	1	0	2
28	3	1	0	0	2
29	3	1	1	0	2

Табела 3.12

Во Табела 3.13 се дадени вектори од Табела 3.12 со изоставување на вектори за кои постои вектор што е поголем од нив

\vec{x}'	x_1	x_2	x_3	x_4	x_5
\vec{x}'_{22}	3	0	0	2	2
\vec{x}'_{24}	3	1	0	1	2
\vec{x}'_{25}	3	0	1	1	2
\vec{x}'_{29}	3	1	1	0	2

Табела 3.13

Значи $G = \{ \vec{x}'_3, \vec{x}'_5, \vec{x}'_6, \vec{x}'_{10}, \vec{x}'_{11}, \vec{x}'_{13}, \vec{x}'_{15}, \vec{x}'_{16}, \vec{x}'_{20}, \vec{x}'_{21}, \vec{x}'_{22}, \vec{x}'_{24}, \vec{x}'_{25}, \vec{x}'_{29} \}$,

$\Omega = \Omega' \cup C_3$; $\Omega' = \{ \{a_4, a_5\}, \{a_1, a_2\}, \{a_1, a_3, a_5\}, \{a_4, a_3, a_2\} \}$

$\Omega = \emptyset$

Чекор 7: Почетните состојби се хомогени, па во овој чекор не вршиме никакви трансформации.

Чекор 8: Од множество $G = \{ \vec{x}'_3, \vec{x}'_5, \vec{x}'_6, \vec{x}'_{10}, \vec{x}'_{11}, \vec{x}'_{13}, \vec{x}'_{15}, \vec{x}'_{16}, \vec{x}'_{20}, \vec{x}'_{21}, \vec{x}'_{22}, \vec{x}'_{24}, \vec{x}'_{25}, \vec{x}'_{29} \}$ се отстрануваат оние вектори за кои постојат поголеми вектори во множеството G . Множеството од повеќе - состојбени минимални пресек вектори, $MMCV = \{ \vec{x}'_3, \vec{x}'_5, \vec{x}'_6, \vec{x}'_{10}, \vec{x}'_{11}, \vec{x}'_{22}, \vec{x}'_{24}, \vec{x}'_{25} \}$

$\vec{x}'_{13} \notin MMCV$ бидејќи $\mathbf{x}_{13} = (0, 1, 0, 2, 2) \leq (1, 1, 1, 2, 2) = \vec{x}'_{10}$

$\vec{x}'_{15} \notin MMCV$ бидејќи $\mathbf{x}_{15} = (0, 1, 1, 2, 1) \leq (1, 1, 1, 2, 2) = \vec{x}'_{10}$

$\vec{x}'_{16} \notin MMCV$ бидејќи $\mathbf{x}_{16} = (1, 1, 0, 2, 1) \leq (1, 1, 1, 2, 2) = \vec{x}'_{10}$

$\vec{x}'_{20} \notin MMCV$ бидејќи $\mathbf{x}_{20} = (1, 1, 1, 2, 0) \leq (1, 1, 1, 2, 2) = \vec{x}'_{10}$

$$\bar{x}_{21} \notin \text{MMCV бидејќи } \mathbf{x}_{21}=(2,1,0,2,0) \leq (3,1,1,2,0) = \bar{x}_6$$

$$\bar{x}_{29} \notin \text{MMCV бидејќи } \mathbf{x}_{29}=(3,1,1,0,2) \leq (3,1,1,0,2) = \bar{x}_3$$

Откако множеството од минимални пресек вектори е најдено, надежноста може да се пресмета со користење на формулата за исклучување/вклучување. Значи се најдени минималните пат вектори за ниво 3. Во Табела 3.14 ќе бидат дадени овие вектори и надежноста за ниво 3.

Минимални пресек вектори за ниво 3	M2TR ₃
3 1 1 0 2	0.83099
3 1 1 1 1	
3 1 1 2 0	
1 1 1 2 2	
2 0 1 2 2	
3 0 0 2 2	
3 1 0 1 2	
3 0 1 1 2	

Табела 3.14

3.2.3 Анализа на алгоритмот

Во овој дел ќе направиме анализа на алгоритмот даден од од Jose E. Ramirez-Marquez, David W. Coit, and Michael Tortorella [9]. Со овој алгоритам се наоѓаат минималните пресек множества за дадено ниво на повеќе – состојбениот транспортен систем. Со алгоритмот не се бараат сите можни комбинации од состојби на линкови. Работи на принцип така што се селектира одредено множество од минимални пресек вектори “потомок пресек” кое што ги наследува информациите од “родителскиот пресек”.

Како прво треба да кажеме дека за наоѓање на повеќе – состојбените минимални пресек вектори треба да ги знаеме бинарните минимални пат вектори. За наоѓање на бинарните минимални пат вектори е искористен алгоритмот даден во Dotson и Gobien [24].

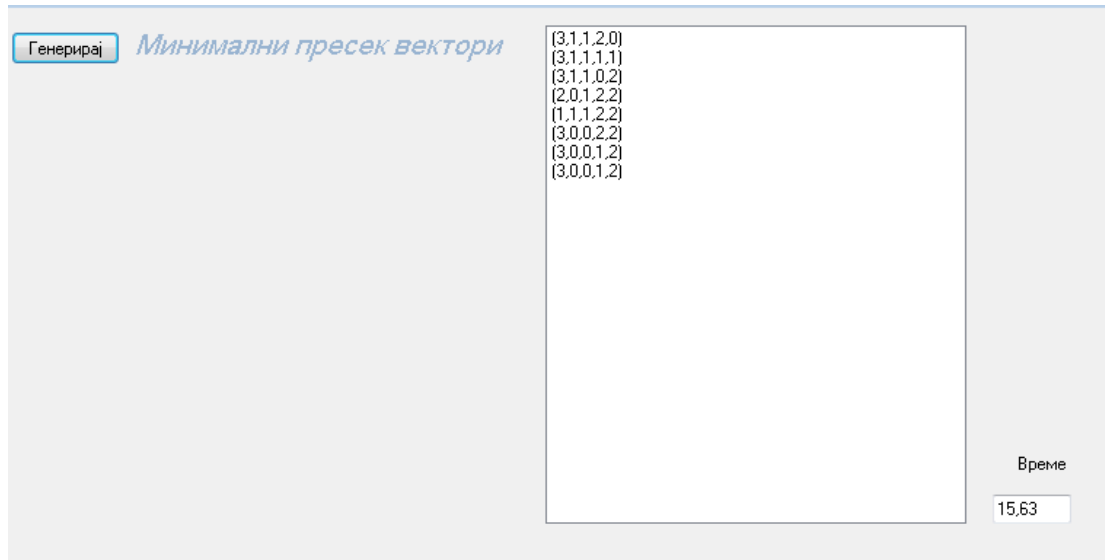
Овој алгоритам функционира добро и со него се наоѓаат сите минимални - пресек вектори за дадено ниво. Меѓутоа не можеме и овде да знаеме кое е максималното ниво на работа на целиот транспортен систем, што е недостаток доколку сакаме да го подобриме транспортниот систем.

Иако овој алгоритам нема ограничување за вредностите на капацитетите на линкови, сепак треба да се нагласи дека еден недостаток на овој алгоритам е тоа што прво е потребно компонентите на транспортниот систем да се трансформираат во слабо хомогени, па дури потоа да се премине алгоритмот. Трансформација до слабо хомогени компоненти ќе влијае на брзината на алгоритмот. Исто така доколку вредностите на капацитетите на линковите се реални, нивната трансформација до слабо хомогени компоненти нема да биде едноставна. Откако минималните пресек вектори ќе бидат најдени тие мора

повторно да се трансформираат во хетероген облик, што значително ќе влијае на времето на извршување на алгоритмот.

Другото ограничување е поради тоа што овој алгоритам може да работи само доколку транспортниот систем G нема циклуси. Значи алгоритмот работи само за ациклични транспортни системи, т.е транспортните системи мора да бидат насочени.

За овој алгоритам е направена програма во програмскиот јазик C#. Излезот од програмот за примерот 3.2 е даден во Фигура 3.7



Фигура 3.7

Да забележаме дека времето за извршување за само едно ниво изнесува 15,63 ms.

Глава 4

4. Нови алгоритми за наоѓање на минимални пат вектори

Во оваа глава се разгледани 3 алгоритми за пресметување на минималните пат вектори на повеќе - состојбен двотерминален транспортен систем. Сите овие алгоритми се едноставни за разбирање и се базираат на едноставни операции помеѓу вектори. Првиот алгоритам работи само за транспортни системи чии линкови може да примаат вредности од множеството $\{0,1,2,\dots,M_i\}$. Со цел да се забрза работа на овој алгоритам даден е и модифициран алгоритам. Вториот алгоритам исто така има ограничување за вредности што може да ги примаат капацитетите на линковите. Капацитетите на линковите може да бидат целобројни вредности. Исто така и за овој алгоритам е даден модифициран алгоритам, за да се забрза неговата работа. Последниот алгоритам е од најголемо значење бидејќи за овој алгоритам не постојат ограничувања на вредности што може да ги примаат капацитетите на линковите. За сите овие алгоритми како и нивните модификации се направени програми во програмскиот јазик C#.

Во оваа глава ќе ги опишеме детално сите алгоритми, и ќе ги анализираме нивните брзини. Треба да се нагласи дека кај сите овие алгоритми се претпоставува дека бинарните минимални пат вектори се познати однапред.

4.1 Алгоритам за наоѓање на минимални пат вектори за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$

Овде ќе разгледаме транспортен систем чии линкови (компоненти) имаат додатно ограничување дека капацитетот на линкот i може да прима само целобројни вредности од множеството $\{0,1,2,\dots,M_i\}$. Во овој случај капацитетот на целиот транспортен систем ќе биде целобројна вредност од множеството $\{0,1,2,\dots,M\}$. Каде $M=(M_1, M_2, \dots, M_n)$ е вектор од максималните состојби на системот.

Како што предходно кажавме, го разгледуваме транспортниот систем $TS=(\mathcal{N}, \mathcal{A}, BPV, S, \mathcal{VP})$, каде \mathcal{N} е множеството од јазли, \mathcal{A} е множеството од линкови, BPV множеството од бинарни минимални пат вектори, S множеството од вектори на капацитети на компонентите, и \mathcal{VP} множеството од веројатности на нивоата на компонентите, каде \vec{p}_i е вектор од веројатности на i -от линк т.е. $p_{id}=P(x_i=d)$.

Својство 4.1 Нека \vec{x} е минимален пат вектор за ниво d (MP_d) за транспортниот систем $TS=(\mathcal{N}, \mathcal{A}, BPV, S, \mathcal{VP})$. Тогаш за да се пренесат d единици од изворот до сливникот, кога системот е во состојба \vec{x} , секој линк се користи само во една насока.

Доказ: Нека \vec{x} е MP_d , така што линкот a_i што ги поврзува јазлите u_1 и u_2 , се користи еднаш во насока од u_1 до u_2 , и еднаш во спротивна насока. Тогаш, една единица оди низ патот w_1 од изворот s до u_1 , потоа низ линкот u_1u_2 и од u_2 до сливникот t низ патот w_2 . Друга единица оди низ патот w'_1 од изворот s до u_2 , потоа низ линкот u_2u_1 и од u_1 до сливникот t низ патот w'_2 . Но, ние можеме да го избереме патот $w_1w'_2$ за едната единица и патот w'_1w_2 за другата. Според тоа, ние добивме пократок пат, значи \vec{x} не е MP_d . \square

Со користење на Својство 4.1 за секој MP_d \vec{x} , линковите на графот на транспортниот систем може да бидат ориентирани во однос на \vec{x} . Следно, ние ќе дадеме многу построго својство за ориентацијата на линковите на транспортниот систем.

Својство 4.2 Нека \vec{x} е MP_d за транспортниот систем $TS=(\mathcal{N}, \mathcal{A}, BPV, S, \mathcal{VP})$. Кога ние сакаме да пренесеме d единици од изворот до сливникот, кога системот е во состојба \vec{x} , секој линк се користи само во една насока. Значи линковите се ориентирани како што се користат. Тогаш добиениот ориентиран подграф е ацикличен.

Доказ: Нека \vec{x} е MP_d . Од својството 4.1 ние можеме да ги насочиме линковите. Да претпоставиме дека има циклус v во добиениот ориентиран подграф. Бидејќи секоја единица оди низ ацикличен пат, ние имаме дека циклусот е користен од различни единици. Можеме да претпоставиме дека $v=w_1\dots w_k$ така што делот w_i се користи од i -тата единица. Значи, патот за i -тата единица може да биде

запишан како $u_i w_i u_i'$, $i=1, \dots, k$. Сега ние можеме да избереме други патишта за секоја од k -те единици, $u_1 u_k'$ за првата, и $u_i u_{i-1}'$ за i -тата единица, $i=2, \dots, k$. Според тоа, ние добиваме пократок пат, значи \bar{x} не е MP_d . \square

Со користење на Својството 4.2 ние можеме да дефинираме подредување на јазлите во однос на \bar{x} .

Дефиниција 4.1 Нека \bar{x} е MP_d за транспортниот систем $TS=(\mathcal{N}, \mathcal{A}, BPV, S, \mathcal{VP})$, и линковите се ориентирани во однос на \bar{x} . За два јазли ќе велиме дека $a <_{\bar{x}} b$ ако постои пат од a до b во насочениот граф. Ние ќе го наречеме ова подредување, **подредување на јазлите во однос на \bar{x}** .

Треба да се забележи дека ова подредување не е секогаш линеарно подредување.

Дефиниција 4.2 Ќе велиме дека $<_{\bar{x}} \Leftrightarrow <_{\bar{y}}$, т.е. подредувањето во однос на \bar{x} и подредувањето во однос на \bar{y} се еквивалентни, ако за сите јазли a и b , $a <_{\bar{x}} b \Rightarrow \neg(b <_{\bar{y}} a)$ и $a <_{\bar{y}} b \Rightarrow \neg(b <_{\bar{x}} a)$.

Дефиниција 4.3 Нека \bar{x} е MP_d и \bar{y} е MP_d така што $<_{\bar{x}} \Leftrightarrow <_{\bar{y}}$. Ние дефинираме $<_{\bar{x}+\bar{y}}$ со $a <_{\bar{x}+\bar{y}} b$ ако $a <_{\bar{x}} b$ или $a <_{\bar{y}} b$.

Забележуваме дека ако транспортниот систем работи со состојба \bar{x} , каде \bar{x} е MP_d , тогаш точно d единици излегуваат од изворот и точно d единици пристигнуваат до сливникот. За сите други јазли ние имаме дека бројот на единици што влегуваат во јазолот е еднаков на бројот на единици што излегуваат од него.

Својство 4.3 Нека $\bar{x}_k \in BPV$ за транспортниот систем $TS=(\mathcal{N}, \mathcal{A}, BPV, S, \mathcal{VP})$, тогаш $\bar{x} = \sum_{k=1}^d \bar{x}_k$ е минимален пат вектор за ниво d .

Доказ: Јасно е дека тврдењето е точно за $d=1$. Претпоставуваме дека тврдењето е точно за сите цели броеви $d \leq r$. За $r+1$ јасно е дека $\sum_{k=1}^{r+1} \bar{x}_k$ е пат вектор, бидејќи

првите r единици минуваат ако системот работи во состојба $\sum_{k=1}^r \bar{x}_k$ и $r+1$ -тата

единица ќе биде пренесена од изворот до сливникот со користење на линковите кои одговараат на векторот \bar{x}_{r+1} . Да претпоставиме дека тоа не е минимален пат вектор. Тоа значи дека постои помал пат вектор кој е минимален, односно на барем еден од линковите на кои сме им додале капацитет 1 со додавањето на

векторот \bar{x}_{r+1} , тој капацитет не требало да се додаде (т.е. имаат поголем капацитет од што е потребно). Нека првиот таков линк започнува од јазолот b_1 , и w_1 е поврзана низа од линкови што започнува со тој линк и сите други линкови во неа имаат поголем капацитет од што е потребно. Нека b' е последниот јазол од w_1 , и линкот l на кој сме му додале капацитет 1 со векторот \bar{x}_{r+1} и што започнува со b' се користи со неговиот потполн капацитет. Да претпоставиме дека пред да се додаде векторот \bar{x}_{r+1} , во јазолот b_1 влегувале и излегувале n_1 единици. Сега ќе излегуваат повторно n_1 единици од кои едната е $r+1$ -тата единица. Тоа значи дека една од претходните r единици кои доаѓале до тој јазол, веќе не стига до него, односно, пред да стигне во тој линк свртува по некој друг линк. Нека свртувањето го прави во јазолот b_2 , т.е. таа единица користи друг линк за да стигне до сливникот, наместо претходната поврзана низа од линкови, w_2 , од јазолот b_2 до јазолот b_1 . Слично на анализата за јазолот b_1 , добиваме дека таа единица користи линковите што претходно се користеле од некоја друга единица. Значи добиваме поврзана низа од линкови w_3 од некој јазол b_3 до b_2 . Можеме да продолжиме на истиот начин, и во секој чекор ќе добиваме поврзана низа од линкови, w_i од некој јазол b_i до јазол b_{i-1} . Но тоа може да се повторува конечен број пати, најмногу $r+1$, и последната од тие единици ќе го користи линкот l за да биде пренесена од b' до сливникот. На тој начин добиваме низа од низи од поврзани насочени линкови $w_s, w_{s-1}, \dots, w_2, w_1$, каде $s < r+1$, која почнува од b' и завршува во b' , т.е. добиваме циклус, што е контрадикција со претпоставката дека нема циклус. Значи мора $\sum_{k=1}^{r+1} \bar{x}_k$ да биде минимален пат вектор. □

Својство 4.4 Ако \bar{x} е MP_d за транспортниот систем $TS=(\mathcal{N}, \mathcal{A}, BPV, S, \mathcal{VP})$ тогаш постојат вектори $\bar{x}_k \in BPV, k=1, \dots, d$, така што $\bar{x} = \sum_{k=1}^d \bar{x}_k$.

Доказ: Доказот повторно ќе биде даден со индукција. Јасно е дека важи за $d=1$. Да претпоставиме дека е точно за сите цели броеви $d \leq r$. Нека \bar{x} е минимален пат вектор за ниво $r+1$. Од својството 4.2 следува дека линковите од графот може да бидат ориентирани во однос на \bar{x} , односно, онака како што се користат за да се пренесат d единици од изворот до сливникот. Нека една од тие единици се пренесува користејќи го $\bar{y} \in BPV$. Имаме дека овој вектор има иста ориентација како и векторот \bar{x} и важи $\bar{y} \leq \bar{x}$. Да го разгледаме векторот $\bar{x} - \bar{y}$. Ова е пат вектор за ниво r , бидејќи ние одземаме капацитети само од линковите по кои една единица поминува, па останатите r единици ќе поминат од изворот до сливникот ако системот е во оваа состојба. Да претпоставиме дека $\bar{x} - \bar{y}$ не е минимален пат вектор за ниво r . Тогаш $\bar{x}' + \bar{y} < \bar{x} - \bar{y} + \bar{y} = \bar{x}$ и $\bar{x}' + \bar{y}$ е минимален пат вектор за ниво $r+1$, што е контрадикција со тоа дека \bar{x} е минимален пат вектор за ниво $r+1$. Значи \bar{x}' е минимален пат вектор за ниво r . Тогаш од

индуктивната претпоставка имаме дека постојат вектори $\vec{x}_k \in \text{BPV}$, $k=1, \dots, d$, така што $\vec{x}' = \sum_{k=1}^d \vec{x}_k$. Па ако означиме $\vec{y} = \vec{x}_{r+1}$, ќе добиеме $\vec{x}' = \sum_{k=1}^r \vec{x}_k + \vec{y} = \sum_{k=1}^{r+1} \vec{x}_k$. \square

Поради Својството 4.2 треба да се генерираат минималните пат вектори за ниво 1, т.е. бинарните минимални пат вектори. Како што е дадено претходно постои процедура `najdi_patista` со која се наоѓаат бинарните патишта за транспортниот систем. Меѓутоа потребно е да се најдат минималните пат вектори за ниво 1. Подолу ќе го објасниме алгоритмот за добивање на бинарните минимални пат вектори. И овој алгоритам е имплементиран во програмскиот јазик C# со процедурата `najdi_binarni_vektori`.

Процедурата `najdi_binarni_vektori` прво ја користи процедурата `numeriрай_linkovi` со која на секој линк му се придружува број. Јасно е дека и јазлите се нумерирани со броеви на пример изворот s е 1 итн.

Со овој алгоритам ќе дадаме како од бинарениот пат го наоѓаме бинарниот минимален вектор.

Алгоритам за наоѓање на бинарен минимален вектор од бинарен минимален пат

Влез е бинарен минимален пат претставен како низа од јазли. При што првиот јазол во низата е изворот, последниот е сливникот, а меѓу два соседни јазли во низата постои линк.

Како излез се добива бинарен минимален вектор кој има a координати, каде a е бројот на линкови во транспортниот систем. Доколку линкот е во работна состојба на соодветна позиција имаме 1, во спротивно 0.

Чекор 1. Ги нумерираме линковите во транспортниот систем.

Чекор 2. Го иницијализираме сите компоненти од векторот на нула. (бројот на компоненти на векторот е еднаков на бројот на линкови)

Чекор 3. За секој соседен пар од јазли го наоѓаме бројот t на линкот што ги поврзува.

Чекор 3.1. Ако првиот од соседниот пар од јазли е помал од вториот ставаме 1 на t -тата компонента во векторот.

Чекор 3.2. Ако првиот од соседниот пар од јазли е поголем од вториот ставаме -1 на t -тата компонента во векторот.

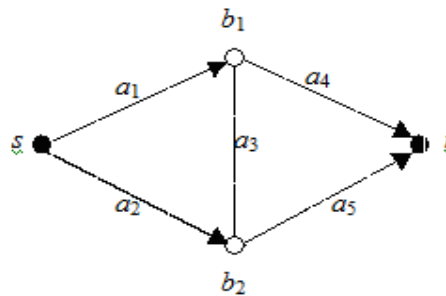
Да ја дадеме сега дефиницијата за **минимален насочен пат вектор за ниво d (MDP за ниво d)**

Дефиниција 4.4 Нека \vec{x} е минимален пат вектор за ниво d , тогаш минималниот насочен пат вектор за ниво d (**MDP за ниво d**), се добива кога во \vec{x} , x_i се

заменува со $-x_i$, секогаш кога i - от линк е двонасочен и се користи негативната насока.

Со процедурата najdi_binarni_vektori ние всушност ги наоѓаме минималните насочени пат вектор за ниво 1.

Да го разгледаме транспортниот систем даден во Фигура 4.1. Тоа е едноставен транспортен систем кој се состои од 4 јазли и 5 линкови (a_1, a_2, a_3, a_4 и a_5)



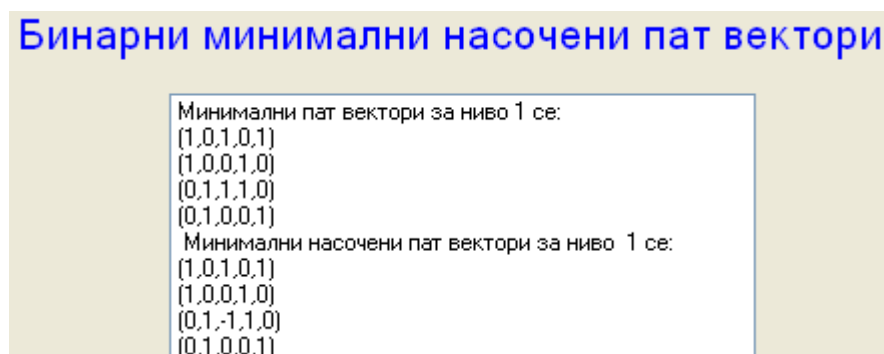
Фигура4.1

Јасно е дека минималните пат множества за транспортен систем Фигура 4.1 се $\{a_1, a_4\}$, $\{a_2, a_5\}$, $\{a_1, a_3, a_5\}$ и $\{a_2, a_3, a_4\}$.

Во овој транспортниот систем двонасочен линк е линкот a_3 и во овој случај можеме да ја избереме насоката од јазолот b_1 до јазолот b_2 за позитивна насока и насока од јазолот b_2 до јазолот b_1 за негативна насока.

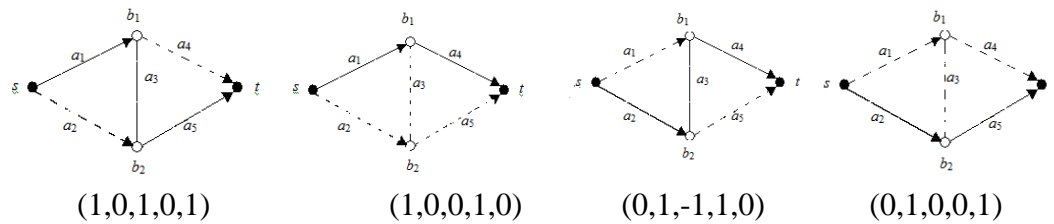
Од минималните патишта можеме да ги добиеме минималните насочени пат вектори за ниво 1 со процедурата najdi_binarni_vektori. Тоа се векторите $(1,0,1,0,1)$, $(1,0,0,1,0)$, $(0,1,-1,1,0)$ и $(0,1,0,0,1)$. Додека минималните пат вектори за ниво 1 се $(1,0,1,0,1)$, $(1,0,0,1,0)$, $(0,1,1,1,0)$ и $(0,1,0,0,1)$

Излезот од алгоритмот е даден со Фигура4.2



Фигура 4.2 Бинарни минимални пат вектори и минимални насочени пат вектори

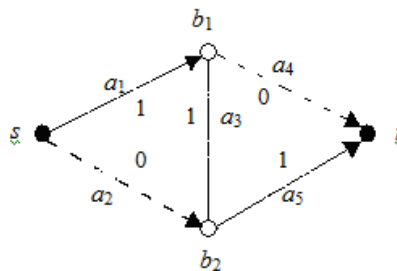
Со фигура 4.3 ќе бидат дадени сите минимални насочени пат вектори



Фигура 4.3 Минимално пат множество $(1,0,1,0,1)$, $(1,0,0,1,0)$, $(0,1,-1,1,0)$, $(0,1,0,0,1)$

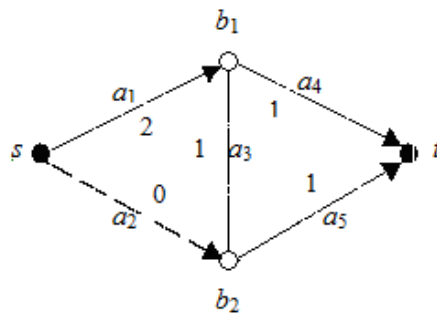
Да го разгледаме сега алгоритам за наоѓање на минимални пат вектори за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$. Како што и претходно спомнавме овој алгоритам се базира на едноставни операции помеѓу векторите. Ќе ја илустрираме основната идеја на овој алгоритам на транспортниот систем даден во Фигура 4.1. Претпоставуваме дека сакаме да пренесеме 3 единици од изворот s до сливникот t .

Претпоставуваме дека првата единица го користи векторот $(1,0,1,0,1)$ за да биде пренесена од изворот до сливникот.



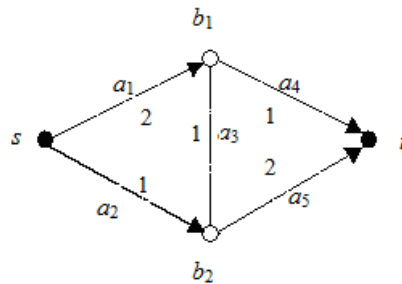
Фигура 4.4

Втората единица нека го користи векторот $(1,0,0,1,0)$ за да биде пренесена од изворот до сливникот. Со собирање на вектори $(1,0,1,0,1)$ и векторот $(1,0,0,1,0)$ се добива векторот $(2,0,1,1,1)$. Векторот $(2,1,1,0,1)$ е минимален пат вектор за ниво 2, како што да се заклучи од својството 4.3.



Фигура 4.5

Третата единица го користи векторот $(0,1,0,0,1)$ за да биде пренесена од изворот до сливникот. Со собирање на минималниот пат вектор за ниво 2 и минимални пат вектор за ниво 1 се добива минималниот пат вектор за ниво 3. Тоа е вектор $(2,1,1,1,2)$.



Фигура 4.6

Треба да се нагласи дека ако оваа постапка ја повториме k - пати ќе ги добиеме минималните пат вектори за ниво k . Значи минималните пат вектори за ниво k се добиваат со собирање на k минимални пат вектори за ниво 1.

Од погоре кажаното следува Тврдењето 4.1

Тврдење 4.1 Нека \vec{x} е минимален насочен пат вектор (MDP) за ниво d и \vec{y} е минимален насочен пат вектор (MDP) за ниво k . Ако векторот $\vec{z} = \vec{x} + \vec{y} < M$, тогаш векторот \vec{z} е минимален насочен пат вектор (MDP) за ниво $d+k$.

Со користење на Својставата 4.3 и 4.4, и Тврдењето 4.1 може да се конструира следниот алгоритам за добивање на сите минимални пат вектори на повеќе - состојбениот двотерминален транспортен систем:

Алгоритам за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$

Влез на овој алгоритам ќе бидат бинарните минимални пат вектори BPV , множеството од вектори на капацитети на компонентите S , и множеството од веројатности на нивоата на компонентите \mathcal{VP} .

Излез од алгоритмот ќе бидат минималните пат вектори за нивоа d и надежностите за соодветните нивоа.

Чекор 1. Се добиваат сите минимални насочени пат вектори за ниво 1, MDP_1 .

Чекор 2. Се иницијализира $d=1$.

Чекор 3. За секој $\vec{x} \in MDP_d$ и за секој $\vec{y} \in MDP_1$, се наоѓаат елементите

$\vec{z} = \vec{x} + \vec{y} \in MDP_{d+1}$. Од овие елементи се конструира множеството $MDP'_{d+1} = \{ \vec{z} \mid \vec{z} = \vec{x} + \vec{y}, \vec{x} \in MDP_d \text{ и } \vec{y} \in MDP_1 \}$

Чекор 4. За секој $\vec{x} \in MDP'_{d+1}$ се наоѓа $Abs(\vec{x}) = (|x_1|, |x_2|, \dots, |x_n|)$

Чекор 5. Од множеството MDP'_{d+1} се добива множеството од минималните насочени пат вектори за ниво $d+1$, MDP_{d+1} на следниот начин:

Чекор 5.1. Од множеството MDP'_{d+1} се бришат сите елементи што се дупликати.

Чекор 5.2. Од множеството MDP'_{d+1} се бришат оние елементи \vec{x} , за кои постои i така што $M_i < |x_i|$. (M_i е максималната состојба во која може да се најде линкот a_i)

Чекор 6. Множеството MP_{d+1} е добиено од множеството $\{Abs(\mathbf{x}) \mid \mathbf{x} \in MDP_{d+1}\}$ со елиминирање на елементите што се појавуваат повеќе од еднаш.

Чекор 7. Се додека $MP_{d+1} \neq 0$, $d=d+1$ и оди на чекор 3 во спротивно оди на чекор 8.

Чекор 8. Кога минималните пат вектори се добиени, надежноста се пресметува со користење на формулата за вклучување и исклучување.

Со програмот `najdi_minvektori` е имплементиран горниот алгоритам во C#.

Алгоритмот ќе го илустрираме со примери.

Пример 4.1:

Да ја разгледаме мрежата во Фигура 4.1. Капацитети на линковите се $S_1=\{0,1,2\}$, $S_2=\{0,1,2\}$, $S_3=\{0,1\}$, $S_4=\{0,1\}$ и $S_5=\{0,1,2,3\}$. Веројатностите со кои линковите може да се најдат во некоја од состојбите се $p_1=(0.1,0.1,0.8)$, $p_2=(0.1,0.1,0.8)$, $p_3=(0.1,0.9)$, $p_4=(0.1,0.9)$ и $p_5=(0.1,0.05,0.05,0.8)$.

Векторот од максимални состојби $M=(2,2,1,1,3)$. Се претпоставува дека бинарните пат вектори кои се и минимални пат вектори за ниво 1, се познати

однапред. Значи множествата од минимални пат вектори и минимални насочени пат вектори за ниво 1 се:

$$MP_1 = \{(1,0,1,0,1), (1,0,0,1,0), (0,1,1,1,0), (0,1,0,0,1)\}$$

$$MDP_1 = \{(1,0,1,0,1), (1,0,0,1,0), (0,1,-1,1,0), (0,1,0,0,1)\}$$

Надежност за ниво 1: $M2TR_1 = 0.97686$.

Во Табела 4.1 детално е опишано како се добиваат минималните пат вектори за ниво 2. Минималните пат вектори за сите нивоа се дадени во Табела 3, во Прилог А, страна 100.

MDP_2'	Вектори кои не се $\leq M$	Дупликати	MDP_2	MP_2	$M2TR_2$
$(1,0,1,0,1)+(1,0,1,0,1)=(2,0,2,0,2)$	$(2,0,0,2,0)$	$(1,1,0,1,1)$	$(2,0,1,1,1)$	$(2,0,1,1,1)$	0.84614
$(1,0,1,0,1)+(1,0,0,1,0)=(2,0,1,1,1)$	$(2,0,2,0,2)$		$(1,1,0,1,1)$	$(1,1,0,1,1)$	
$(1,0,1,0,1)+(0,1,-1,1,0)=(1,1,0,1,1)$	$(1,1,-1,2,0)$		$(1,1,1,0,2)$	$(1,1,1,0,2)$	
$(1,0,1,0,1)+(0,1,0,0,1)=(1,1,1,0,2)$	$(0,2,-2,2,0)$		$(0,2,-1,1,1)$	$(0,2,1,1,1)$	
$(1,0,0,1,0)+(1,0,0,1,0)=(2,0,0,2,0)$			$(0,2,0,0,2)$	$(0,2,0,0,2)$	
$(1,0,0,1,0)+(0,1,-1,1,0)=(1,1,-1,2,0)$					
$(1,0,0,1,0)+(0,1,0,0,1)=(1,1,0,1,1)$					
$(0,1,-1,1,0)+(0,1,-1,1,0)=(0,2,2,2,0)$					
$(0,1,-1,1,0)+(0,1,0,0,1)=(0,2,-1,1,1)$					
$(0,1,0,0,1)+(0,1,0,0,1)=(0,2,0,0,2)$					

Табела 4.1 Минимални пат вектори за ниво 2

Излезот ќе биде даден во Фигура 4.7



Минимални пат вектори

Патишта за ниво3:

- 3,0,2,1,2
- 3,0,1,2,1
- 2,1,0,2,1
- 2,1,1,1,2
- 2,1,1,1,2
- 2,1,0,2,1
- 1,2,-1,2,1
- 1,2,0,1,2
- 2,1,2,0,3
- 2,1,1,1,2
- 1,2,0,1,2
- 1,2,1,0,3
- 1,2,0,1,2
- 1,2,-1,2,1
- 0,3,-2,2,1
- 0,3,-1,1,2
- 1,2,1,0,3
- 1,2,0,1,2
- 0,3,-1,1,2
- 0,3,0,0,3

Патишта кои не се помали или еднакви на векторот од максимални состојби M:

- 3,0,2,1,2
- 3,0,1,2,1
- 2,1,0,2,1
- 2,1,0,2,1
- 1,2,-1,2,1
- 2,1,2,0,3
- 1,2,-1,2,1
- 0,3,-2,2,1
- 0,3,-1,1,2

Време: 31,25

Минимални пат вектори

- 0,3,-1,1,2
- 0,3,0,0,3

Дупликати(некои се јавуваат повеќе пати:

- 2,1,1,1,2
- 2,1,1,1,2
- 1,2,0,1,2
- 1,2,0,1,2
- 1,2,0,1,2
- 1,2,1,0,3

Минимални насочени пат вектори за ниво3:

- 2,1,1,1,2
- 1,2,0,1,2
- 1,2,1,0,3

Минимални пат вектори за ниво3:

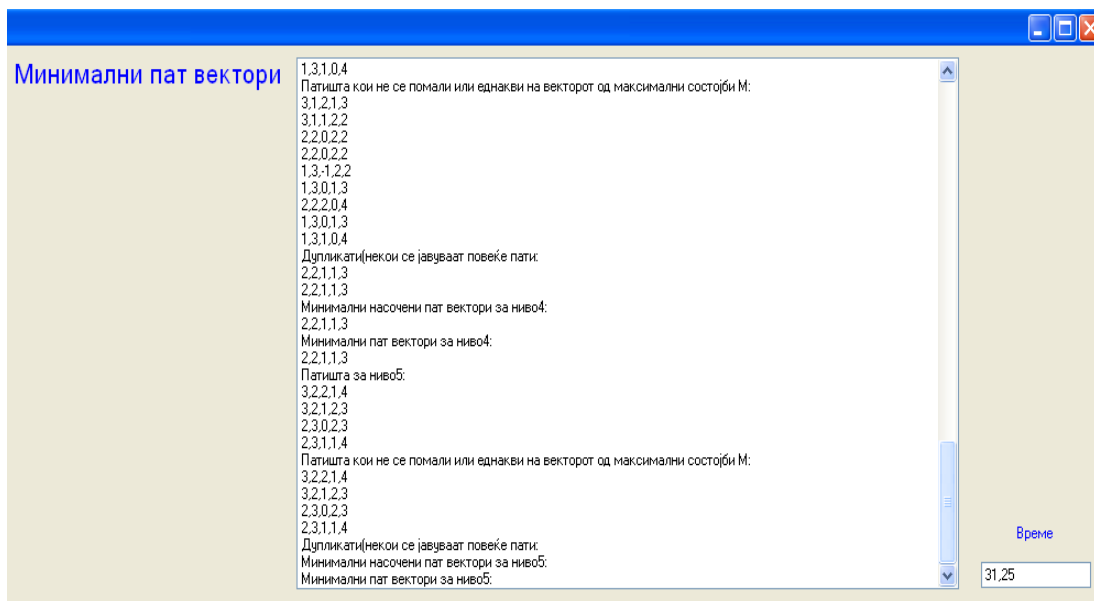
- 2,1,1,1,2
- 1,2,0,1,2
- 1,2,1,0,3

Патишта за ниво4:

- 3,1,2,1,3
- 3,1,1,2,2
- 2,2,0,2,2
- 2,2,1,1,3
- 2,2,1,1,3
- 2,2,0,2,2
- 1,3,-1,2,2
- 1,3,0,1,3
- 2,2,2,0,4
- 2,2,1,1,3
- 1,3,0,1,3
- 1,3,1,0,4

Патишта кои не се помали или еднакви на векторот од максимални состојби M:

Време: 31,25



Фигура 4.7

Во Фигура 4.7 е даден излезот од горниот алгоритам , треба да се забележи дека времето на извршување на овој алгоритам е 31.25 ms

Од Табела 3, во Прилог А, страна 100 може да се види дека за ниво 2 се добиваат 10 патишта, од кои 4 не се помали или еднакви од M и еден е дупликат. За ниво 3 се добиваат 20 патишта од кои 11 не се помали или еднакви од M и 6 се дупликати. Додека за ниво 4 се добиваат 12 патишта од кои 9 не се помали или еднакви од M и 2 се дупликати..

Алгоритмот ќе го илустрираме на уште еден транспортен систем. Транспортниот систем ќе се состои од 6 јазли и 8 линкови. Земаме поголем транспортен систем со цел да вршиме споредба помеѓу овој алгоритам и останатите алгоритми дадени во оваа глава.

Пример 4.2

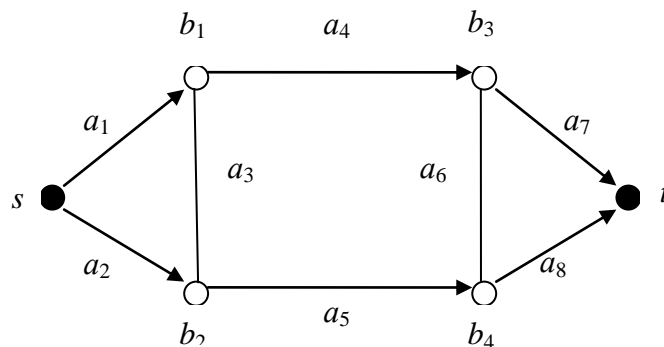
Да го разгледаме транспортниот систем даден во Фигура 4.8. Капацитети на линковите се $S_1=\{0,1,2\}$, $S_2=\{0,1\}$, $S_3=\{0,1\}$, $S_4=\{0,1\}$, $S_5=\{0,1,2\}$, $S_6=\{0,1\}$, $S_7=\{0,1\}$ и $S_8=\{0,1,2\}$. Веројатностите со кои линковите може да се најдат во некоја од состојбите се $p_1=(0.1,0.1,0.8)$, $p_2=(0.1,0.9)$, $p_3=(0.1,0.9)$, $p_4=(0.05,0.95)$ и $p_5=(0.1,0.05,0.85)$, $p_6=(0.05,0.95)$, $p_7=(0.1,0.9)$ и $p_8=(0.1,0.1,0.8)$.

Векторот од максимални состојби $M=(2,1,1,1,2,1,1,2)$ Се претпоставува дека бинарните пат вектори кои се и минимални пат вектори за ниво 1, се познати однапред. Значи множествата од минимални пат вектори и минимални насочени пат вектори за ниво 1 се

$MP_1=\{(1,0,1,0,1,1,1,0), (1,0,1,0,1,0,0,1), (1,0,0,1,0,1,0,1), (1,0,0,1,0,0,1,0), (0,1,1,1,0,1,0,1), (0,1,1,1,0,0,1,0), (0,1,0,0,1,1,1,0), (0,1,0,0,1,0,0,1)\}$

$MDP_1=\{(1,0,1,0,1,-1,1,0), (1,0,1,0,1,0,0,1), (1,0,0,1,0,1,0,1), (1,0,0,1,0,0,1,0), (0,1,-1,1,0,1,0,1), (0,1,-1,1,0,0,1,0), (0,1,0,0,1,-1,1,0), (0,1,0,0,1,0,0,1)\}$

Надежност за ниво 1 е : $M2TR_1=0.97326$.



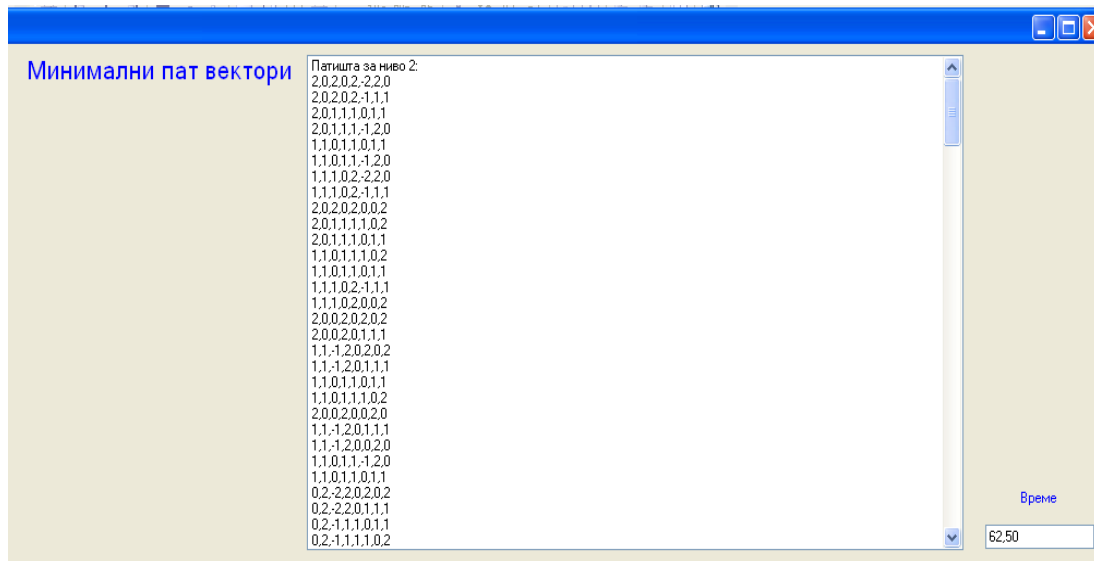
Фигура 4.8

Минималните патишта за ниво 2, ќе бидат дадени во Табела 4.2, додека минималните пат вектори за сите нивоа ќе бидат дадени во Табела 4 во Прилог А, страна 101.

MDP_2	Вектори кои не се $\leq M$	Дупликати	MDP_2	MP_d	$M2TR_2$
(2,0,2,0,2,-2,2,0)	(2,0,2,0,2,-2,2,0)	(2,0,1,1,1,0,1,1)	(2,0,1,1,1,0,1,1)	(2,0,1,1,1,0,1,1)	0,69793
(2,0,2,0,2,-1,1,1)	(2,0,2,0,2,-1,1,1)	(1,1,0,1,1,0,1,1)	(1,1,0,1,1,0,1,1)	(1,1,0,1,1,0,1,1)	
(2,0,1,1,1,0,1,1)	(2,0,1,1,1,-1,2,0)	(1,1,0,1,1,0,1,1)	(1,1,1,0,2,-1,1,1)	(1,1,1,0,2,-1,1,1)	
(2,0,1,1,1,-1,2,0)	(1,1, 0,1,1,-1,2,0)	(1,1,0,1,1,0,1,1)	(2,0,1,1,1,0,2)	(2,0,1,1,1,0,2)	
(1,1,0,1,1,0,1,1)	(1,1,1,0,2,-2,2,0)	(1,1,1,0,2,-1,1,1)	(1,1,0,1,1,0,2)	(1,1,0,1,1,0,2)	
(1,1, 0,1,1,-1,2,0)	(2,0,2,0,2,0,0,2)	(1,1,0,1,1,0,2)	(1,1,1,0,2,0,0,2)	(1,1,1,0,2,0,0,2)	
(1,1,1,0,2,-2,2,0)	(2,0,2,0,2,0,0,2)				
(1,1,1,0,2,-1,1,1)	(2,0,0,2,0,1,1,1)				
(2,0,2,0,2,0,0,2)	(1,1,-1,2,0,2,0,2)				
(2,0,1,1,1,0,2)	(1,1,-1,2,0,1,1,1)				
(2,0,1,1,1,0,1,1)	(2,0,0,2,0,0,2,0)				
(1,1,0,1,1,1,0,2)	(1,1,-1,2,0,1,1,1)				
(1,1,0,1,1,0,1,1)	(1,1,-1,2,0,0,2,0)				
(1,1,1,0,2,-1,1,1)	(1,1,0,1,1,-1,2,0)				
(1,1,1,0,2,0,0,2)	(0,2,-2,2,0,2,0,2)				
(2,0,0,2,0,2,0,2)	(0,2,-2,2,0,1,1,1)				
(2,0,0,2,0,1,1,1)	(0,2,-1,1,1,0,1,1)				
(1,1,-1,2,0,2,0,2)	(0,2,-1,1,1,1,0,2)				
(1,1,-1,2,0,1,1,1)	(0,2,-2,2,0,0,2,0)				
(1,1,0,1,1,0,1,1)	(0,2,-1,1,1,-1,2,0)				
(1,1,0,1,1,1,0,2)	(0,2,-1,1,1,0,1,1)				
(2,0,0,2,0,0,2,0)	(0,2,0,0,2,-2,2,0)				
(1,1,-1,2,0,1,1,1)	(0,2,0,0,2,-1,1,1)				
(1,1,-1,2,0,0,2,0)	(0,2,0,0,2,0,0,2)				
(1,1,0,1,1,-1,2,0)					
(1,1,0,1,1,0,1,1)					
(0,2,-2,2,0,2,0,2)					
(0,2,-2,2,0,1,1,1)					
(0,2,-1,1,1,0,1,1)					
(0,2,-1,1,1,1,0,2)					
(0,2,-2,2,0,0,2,0)					
(0,2,-1,1,1,-1,2,0)					
(0,2,-1,1,1,0,1,1)					
(0,2,0,0,2,-2,2,0)					
(0,2,0,0,2,-1,1,1)					
(0,2,0,0,2,0,0,2)					

Табела 4.2 Минимални пат вектори за ниво 2

За ниво 2 се добиваат 36 патишта, од кои 24 не се помали или еднакви од M и 6 се дупликати. За ниво 3 се добиваат 48 патишта од кои 42 не се помали или еднакви од M и 5 се дупликати.



Фигура 4.9

Во Фигура 4.9 е даден излезот од горниот алгоритмот за примерот 4.2 (само почетокот), треба да се забележи дека времето на извршување на овој алгоритам е 62.50 ms.

4.2 Убрзан алгоритам за наоѓање на минимални пат вектори за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$

Во примерите 4.1 и 4.2 што ги разгледавме во предходната глава се забележува дека се добиваат многу патишта од кои голем број не се помали или еднакви од вектор од максимални состојби M или пак се дупликати. Ние сакаме да најдеме начин да ги редуцираме непотребните елементи. Од тие причини вршиме анализа како се добиени непотребните вектори

1. Според својството 4.2 секој двонасочен линк во минималните пат вектори се користи во една насока. Според тоа дупликати се добиваат кога вектори што имаат спротивен знак на иста координата се собираат. Затоа ќе ги тестираме знаците на векторите и ќе ги собираме само оние вектори што имаат ист знак на иста координата. Но ако е мал бројот на двонасочни линкови, тогаш тоа не е неопходно. Бидејќи во тој случај контролата може да одземе многу време.
2. Дупликати исто така се добиваат кога се собира исто множество од минимални насочени пат вектори за ниво 1 (MDP_1) во различен редослед. На пример векторот $(1,2,0,1,2)$ од пример 4.1 е добиен три пати како минимален пат за ниво 3. Првиот пат е добиен како

$$(1,2,0,1,2) = (1,1,0,1,1) + (0,1,0,0,1) = ((1,0,1,0,1) + (0,1,-1,1,0)) + (0,1,0,0,1),$$

вториот пат како

$$(1,2,0,1,2) = (1,1,1,0,2) + (0,1,-1,1,0) = ((1,0,1,0,1) + (0,1,0,0,1)) + (0,1,-1,1,0),$$

и третиот пат како

$$(1,2,0,1,2) = (0,2,-1,1,1) + (1,0,1,0,1) = ((0,1,-1,1,0) + (0,1,0,0,1)) + (1,0,1,0,1)$$

Сите три пати се собираат векторите $(1,0,1,0,1)$, $(0,1,-1,1,0)$ и $(0,1,0,1,0)$ но во различен редослед. Поради ова овде ќе се разгледаме нов алгоритам што е модификација на овој алгоритам, при што во новиот алгоритам секое множество од вектори ќе биде дадено само еднаш. Во новиот алгоритам исто така ќе вршиме проверка дали вектори имаат спротивни знаци на иста координата, и доколку е така тогаш тие вектори нема да ги собираме.

Споеред својството 4.3 ако \bar{x} е MP_d тогаш постојат вектори $\bar{x}_k \in BPV$, $k=1, \dots, d$, така што $\bar{x} = \sum_{k=1}^d \bar{x}_k$. Според тоа и ако \bar{x} е MDP_d тогаш ќе постојат вектори

$\bar{x}_k \in MDP_1$, $k=1, \dots, d$, така што $\bar{x} = \sum_{k=1}^d \bar{x}_k$. Уште повеќе секој вектор $\bar{y} \in MDP_1$ во \bar{x} не се појавува повеќе пати од $\min_{y_i=1} M_i$. Значи за секој вектор $\bar{y} \in MDP_1$,

земаме индикатор $c_{\bar{y}} = \min_{|y_i|=1} M_i$. Тогаш за да се добијат сите минимални патишта за ниво d , ги конструираме сите d множества од MDP_1 , така што секој вектор $\bar{y} \in MDP_1$ се појавува најмногу $c_{\bar{y}}$ пати.

Нека векторот $\bar{x}_i \in MDP_d$, е од облик $\bar{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$, а векторот $\bar{y}_j \in MDP_1$ е од облик $\bar{y}_j = (y_{j,1}, y_{j,2}, \dots, y_{j,n})$. Дефинираме функција $h: MDP_1 \rightarrow \{1, 2, \dots, n\}$ со $h(\bar{y}_i) = i$. На секој $\bar{x} \in MDP_d$ му придружуваме множество $\mathcal{B}_{\bar{x}} = \{i1, i2, \dots, id\}$ каде $\bar{x} = \bar{y}_{i,1} + \bar{y}_{i,2} + \dots + \bar{y}_{i,d}$ за $\bar{y}_{ij} \in MDP_1$

Да го опишеме сега новиот алгоритам преку неговите чекори

Убрзан алгоритам за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$

Влез на овој алгоритам ќе бидат бинарните минимални пат вектори BPV , множеството од вектори на капацитети на компонентите S , и множеството од веројатности на нивоата на компонентите \mathcal{VP} .

Излез од алгоритмот ќе бидат минималните пат вектори за нивоа d и надежностите за соодветните нивоа.

Чекор 1. Се добиваат сите минимални насочени пат вектори за ниво $1, MDP_1$

Чекор 2. Се дефинира функција $h: MDP_1 \rightarrow \{1,2,\dots,L\}$ (каде L е бројот

елементите во MDP_1) со $h(\bar{y}_j) = j$

Чекор 3. На секој $\bar{y}_j \in MDP_1$ му се придружува множество $\mathcal{B}_{\bar{y}_j} = \{j\}$

Чекор 4. За секој $\bar{y}_j \in MDP_1$ се наоѓа $c_{\bar{y}_j} = \min_{|y_{ji}|=1} M_i$.

Чекор 5. Се иницијализира $d=1$

Чекор 6. Се означува со m бројот на елементите во MDP_d , а со k бројот на елементите во MDP_1 .

Чекор 7. Се сетира $MDP'_{d+1} = \emptyset$

Чекор 8. За $i=1$ до m

За $j=1$ до k

$$\mathcal{B}_{\bar{x}_i + \bar{y}_j} = \mathcal{B}_{\bar{x}_i} \cup \{j\}$$

(Ако бројот на појавувања на j во $\mathcal{B}_{\bar{x}_i + \bar{y}_j}$ е помал или еднаков на $c_{\bar{y}_j}$)

и (Ако не постои $\bar{z}' \in MDP'_{d+1}$ така што $\mathcal{B}_{\bar{z}'} = \mathcal{B}_{\bar{x}_i + \bar{y}_j}$)

и (Ако за секој l , $x_{il} \cdot y_{jl} \geq 0$)

$$MDP'_{d+1} = MDP'_{d+1} \cup \{\bar{x}_i + \bar{y}_j\} \quad (\bar{z} = \bar{x}_i + \bar{y}_j)$$

Чекор 9. За секој $\bar{z} \in MDP'_{d+1}$ се наоѓа $Abs(\bar{z}) = (|z_1|, |z_2|, \dots, |z_n|)$

Чекор 10. Од множеството MDP'_{d+1} се добива множеството од

од минималните пат вектори за ниво $d+1$, MDP_{d+1} на следниот начин:

Чекор 10.1. Од множеството MDP'_{d+1} се бришат сите елементи што се дупликати.

Чекор 10.2. Од множеството MDP'_{d+1} се бришат оние елементи \bar{x} , за кои постои i така што $M_i < |x_i|$. (M_i е максималната состојба во која може да се најде линкот a_i)

Чекор 11. Множеството MP_{d+1} е добиено од множеството $\{Abs(\bar{x}) | \bar{x} \in MDP_{d+1}\}$ со елиминирање на елементите што се појавуваат повеќе од еднаш.

Чекор 12. Се додека $MP_{d+1} \neq \emptyset$, $d=d+1$ и оди на чекор 6 во спротивно оди на чекор 13.

Чекор 13. Кога минималните пат вектори се добиени, надежноста се пресметува со користење на формулата за вклучување и исклучување

Со програмот `najdi_minvektori_uzbrzan` е имплементиран горниот алгоритам во C#.

Пример 4.3

Ќе го применеме сега овој алгоритам на примерот 4.1. Повторно го разгледуваме транспортниот систем во Фигура 5.1. Капацитети на линковите се $S_1 = \{0,1,2\}$, $S_2 = \{0,1,2\}$, $S_3 = \{0,1\}$, $S_4 = \{0,1\}$ и $S_5 = \{0,1,2,3\}$. Веројатностите со кои линковите може да се најдат во некоја од состојбите се $p_1 = (0.1, 0.1, 0.8)$, $p_2 = (0.1, 0.1, 0.8)$, $p_3 = (0.1, 0.9)$, $p_4 = (0.1, 0.9)$ и $p_5 = (0.1, 0.05, 0.05, 0.8)$.

Векторот од максимални состојби $M=(2,2,1,1,3)$ Се претпоставува дека бинарните пат вектори кои се и минимални пат вектори за ниво 1, се познати однапред. Значи множествата од минимални пат вектори и минимални насочени пат вектори за ниво 1 се:

$$MP_1 = \{(1,0,1,0,1), (1,0,0,1,0), (0,1,1,1,0), (0,1,0,0,1)\}$$

$$MDP_1 = \{(1,0,1,0,1), (1,0,0,1,0), (0,1,-1,1,0), (0,1,0,0,1)\}$$

Надежност за ниво 1: $M2TR_1=0.97686$.

Дефинираме функција $h:MDP_1 \rightarrow \{1,2,3,4\}$, така што

$$h((1,0,1,0,1))=1, h((1,0,0,1,0))=2, h((0,1,-1,1,0))=3 \text{ и } h((0,1,0,0,1))=4$$

За секој елемент $\bar{y} \in MDP_1$ му се придружува множество $\mathcal{B}_{\bar{y}}$

$$\mathcal{B}_{(1,0,1,0,1)} = \{1\}, \mathcal{B}_{(1,0,0,1,0)} = \{2\}, \mathcal{B}_{(0,1,-1,1,0)} = \{3\}, \mathcal{B}_{(0,1,0,0,1)} = \{4\}$$

За секој $\bar{y} \in MDP_1$ ги наоѓаме $c_{(1,0,1,0,1)} = \min(2,1,3)=1$, $c_{(1,0,0,1,0)} = \min(2,1)=1$, $c_{(0,1,-1,1,0)} = \min(2,1,1)=1$ и $c_{(0,1,0,0,1)} = \min(2,3)=2$

На почеток $d=1$, $m=k=4$. Да ги разгледаме сите можни комбинации за кои $\bar{x} \in MDP_d = MDP_1$ и $\bar{y} \in MDP_1$. Јасно, според алгоритам се сетира $MDP_2 = \emptyset$.

Да видиме како се добиваат минималните пат вектори за ниво 2.

За $\bar{x}=(1,0,1,0,1)$ и $\bar{y}=(1,0,1,0,1)$, $\mathcal{B}_{\bar{x}+\bar{y}} = \{1,1\}$, бидејќи бројот на појавувања на векторот $(1,0,1,0,1)$ во $\mathcal{B}_{\bar{x}+\bar{y}}$ е $2 > c_{(1,0,1,0,1)}=1$, па разгледуваме нов пар од вектори.

За $\bar{x}=(1,0,1,0,1)$ и $\bar{y}=(1,0,0,1,0)$, $\mathcal{B}_{\bar{x}+\bar{y}} = \{1,2\}$, бидејќи бројот на појавувања на векторот $(1,0,0,1,0)$ во $\mathcal{B}_{\bar{x}+\bar{y}}$ е $1 = c_{(1,0,0,1,0)}=1$, не постои вектор во $MDP_2 = \emptyset$ кој е добиен со собирање на овие вектори и овие два вектори немаат спротивен знак на иста координата нивниот збир $\bar{x} + \bar{y} = (2,0,1,1,1) \in MDP_2$, $MDP_2 = \{(2,0,1,1,1)\}$

За $\bar{x}=(1,0,1,0,1)$ и $\bar{y}=(0,1,-1,1,0)$, $\mathcal{B}_{\bar{x}+\bar{y}} = \{1,3\}$, бројот на појавувања на векторот $(0,1,-1,1,0)$ во $\mathcal{B}_{\bar{x}+\bar{y}}$ е $1 = c_{(0,1,-1,1,0)}=1$, не постои вектор во $MDP_2 = \{(2,0,1,1,1)\}$ кој е добиен со собирање на овие вектори, но за $i=3$, $x_i y_i < 0$ па нема да ги собираме овие два вектори и одиме на следен пат од вектори.

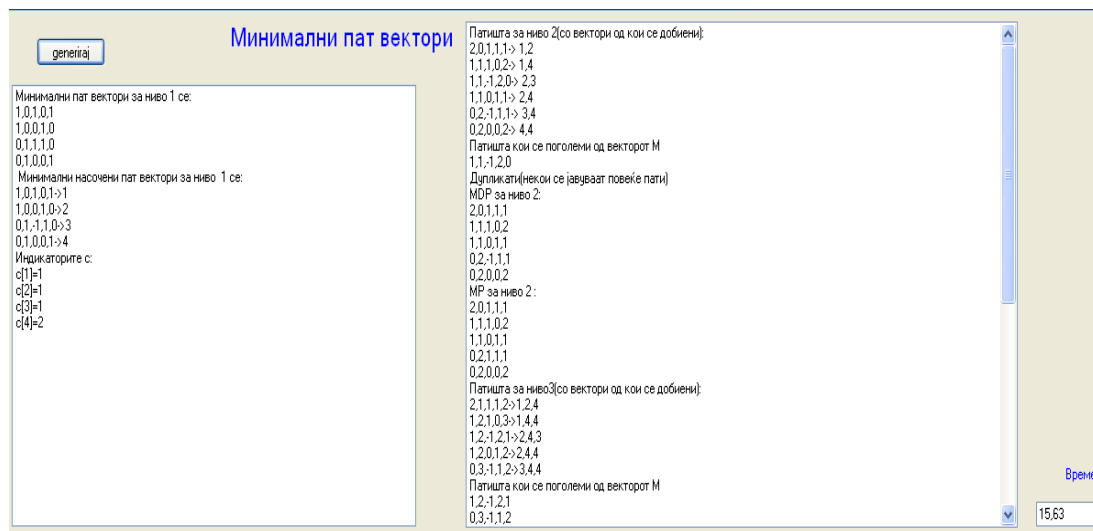
За $\bar{x}=(1,0,1,0,1)$ и $\bar{y}=(0,1,0,0,1)$, $\mathcal{B}_{\bar{x}+\bar{y}} = \{1,4\}$, бидејќи бројот на појавувања на векторот $(0,1,0,0,1)$ во $\mathcal{B}_{\bar{x}+\bar{y}}$ е $1 < c_{(0,1,0,0,1)}=2$, не постои вектор во $MDP_2 = \{(2,0,1,1,1)\}$ кој е добиен со собирање на овие вектори, и овие два вектори немаат спротивен знак на иста координата па $\bar{x} + \bar{y} = (1,1,1,0,2) \in MDP_2$, $MDP_2 = \{(2,0,1,1,1), (1,1,1,0,2)\}$

На ист начин продолжуваме да ги разгледуваме сите парови од вектори. Во Табела 4.3 се дадени минималните пат вектори и надежностите.

d	$MDP_d \rightarrow \mathcal{B}$	Вектори кои не се $\leq M$	Дупликати	MDP_d	MP_d	$M2TR_d$
2	$(2,0,1,1,1) \rightarrow \{1,2\}$ $(1,1,1,0,2) \rightarrow \{1,4\}$ $(1,1,-1,2,0) \rightarrow \{2,3\}$ $(1,1,0,1,1) \rightarrow \{2,4\}$ $(0,2,-1,1,1) \rightarrow \{3,4\}$ $(0,2,0,0,2) \rightarrow \{4,4\}$	$(1,1,-1,2,0)$		$(2,0,1,1,1)$ $(1,1,1,0,2)$ $(1,1,0,1,1)$ $(0,2,-1,1,1)$ $(0,2,0,0,2)$	$(2,0,1,1,1)$ $(1,1,1,0,2)$ $(1,1,0,1,1)$ $(0,2,1,1,1)$ $(0,2,0,0,2)$	0.84614
3	$(2,1,1,1,2) \rightarrow \{1,2,4\}$ $(1,2,1,0,3) \rightarrow \{1,4,4\}$ $(1,2,-1,2,1) \rightarrow \{2,4,3\}$ $(1,2,0,1,2) \rightarrow \{2,4,4\}$ $(0,3,-1,1,2) \rightarrow \{3,4,4\}$	$(1,2,-1,2,1)$ $(0,3,-1,1,2)$		$(2,1,1,1,2)$ $(1,2,1,0,3)$ $(1,2,0,1,2)$	$(2,1,1,1,2)$ $(1,2,1,0,3)$ $(1,2,0,1,2)$	0.64584
4	$(2,2,1,1,3) \rightarrow \{1,2,4,4\}$ $(1,3,-1,2,2) \rightarrow \{2,4,4,3\}$	$(1,3,-1,2,2)$		$(2,2,1,1,3)$	$(2,2,1,1,3)$	0.3686

Табела 4.3 Минимални пат вектори

За ниво 2 се добиваат 6 патишта, од кои 1 не е помал или еднаков од M и нема дупликати. За ниво 3 се добиваат 5 патишта од кои 2 не се помали или еднакви од M и нема дупликати. За ниво 4 се добиваат 2 патишта од кои 1 не е помал или еднаков од M и нема дупликати.



Фигура 4.10

Во Фигура 4.10 е даден излезот од убрзаниот алгоритам за примерот 4.3 (само почетокот), треба да се забележи дека времето на извршување на овој алгоритам е 15.63 ms

Пример4.4

Да го примениме сега алгоритмот на примерот 4.2. Повторно ја разгледуваме мрежата во Фигура 4.8. Капацитети на линковите се $S_1=\{0,1,2\}$, $S_2=\{0,1\}$, $S_3=\{0,1\}$, $S_4=\{0,1\}$, $S_5=\{0,1,2\}$, $S_6=\{0,1\}$, $S_7=\{0,1\}$ и $S_8=\{0,1,2\}$. Веројатностите со кои линковите може да се најдат во некоја од состојбите се $p_1=(0.1,0.1,0.8)$, $p_2=(0.1,0.9)$, $p_3=(0.1,0.9)$, $p_4=(0.05,0.95)$, $p_5=(0.1,0.05,0.85)$, $p_6=(0.05,0.95)$, $p_7=(0.1,0.9)$ и $p_8=(0.1,0.1,0.8)$,

Векторот од максимални состојби $M=(2,1,1,1,2,1,1,2)$ Се претпоставува дека бинарните пат вектори кои се и минимални пат вектори за ниво 1, се познати однапред. Значи множествата од минимални пат вектори и минимални насочени пат вектори за ниво 1 се

$MP_1=\{(1,0,1,0,1,1,1,0), (1,0,1,0,1,0,0,1), (1,0,0,1,0,1,0,1), (1,0,0,1,0,0,1,0), (0,1,1,1,0,1,0,1), (0,1,1,1,0,0,1,0), (0,1,0,0,1,1,1,0), (0,1,0,0,1,0,0,1)\}$

$MDP_1=\{(1,0,1,0,1,-1,1,0), (1,0,1,0,1,0,0,1), (1,0,0,1,0,1,0,1), (1,0,0,1,0,0,1,0), (0,1,-1,1,0,1,0,1), (0,1,-1,1,0,0,1,0), (0,1,0,0,1,-1,1,0), (0,1,0,0,1,0,0,1)\}$

Надежност за ниво 1 е : $M2TR_1=0.97326$.

$\bar{x}=(1,0,1,0,1)$ и $\bar{y}=(0,1,0,0,1)$

Дефинираме функција $h: MDP_1 \rightarrow \{1,2,3,4,5,6,7,8\}$, така што

$h((1,0,1,0,1,1,1,0))=1$, $h((1,0,1,0,1,0,0,1))=2$, $h((1,0,0,1,0,1,0,1))=3$,
 $h((1,0,0,1,0,0,1,0))=4$, $h((0,1,-1,1,0,1,0,1))=5$, $h((0,1,-1,1,0,0,1,0))=6$,
 $h((0,1,0,0,1,-1,1,0))=7$ и $h((0,1,0,0,1,0,0,1))=8$

За секој елемент од $\bar{y} \in MDP_1$ му се придружува множество $\mathcal{B}_{\bar{y}}$,

$\mathcal{B}_{(1,0,1,0,1,-1,1,0)}=\{1\}$, $\mathcal{B}_{(1,0,1,0,1,0,0,1)}=\{2\}$, $\mathcal{B}_{(1,0,0,1,0,1,0,1)}=\{3\}$, $\mathcal{B}_{(1,0,0,1,0,0,1,0)}=\{4\}$,
 $\mathcal{B}_{(0,1,-1,1,0,1,0,1)}=\{5\}$, $\mathcal{B}_{(0,1,-1,1,0,0,1,0)}=\{6\}$, $\mathcal{B}_{(0,1,0,0,1,-1,1,0)}=\{7\}$ и $\mathcal{B}_{(0,1,0,0,1,0,0,1)}=\{8\}$

За секој $\bar{y} \in MDP_1$ ги наоѓаме

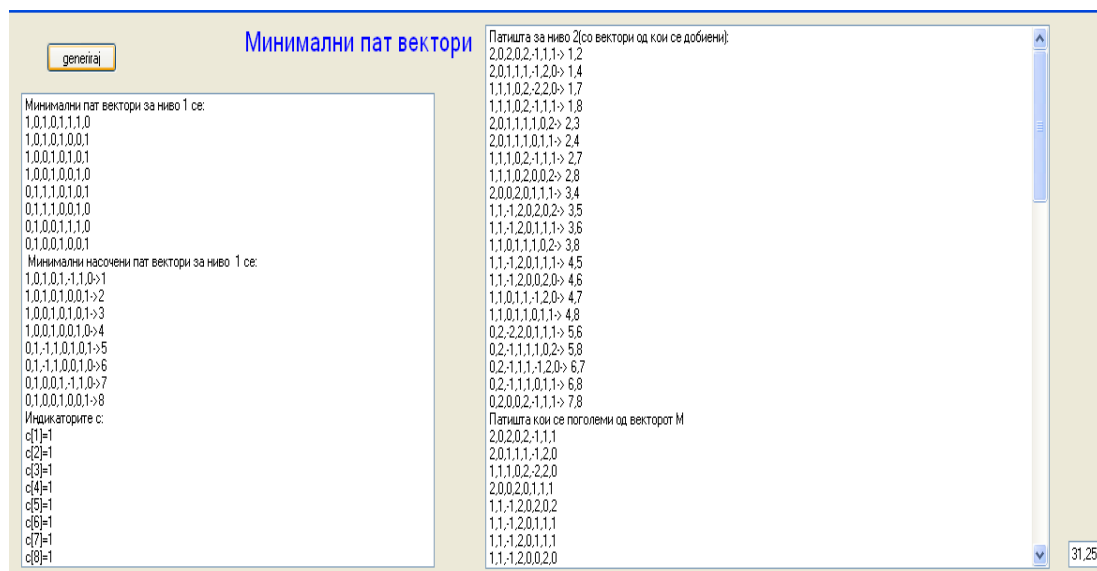
$c_{(1,0,1,0,1,-1,1,0)}=\min(2,1,2,1,1)=1$, $c_{(1,0,1,0,1,0,0,1)}=\min(2,1,2,2)=1$,
 $c_{(1,0,0,1,0,1,0,1)}=\min(2,1,1,2)=1$, $c_{(1,0,0,1,0,0,1,0)}=\min(2,1,1)=1$,
 $c_{(0,1,1,1,0,1,0,1)}=\min(1,1,1,1,2)=1$, $c_{(0,1,1,1,0,0,1,0)}=\min(1,1,1,1)=1$,
 $c_{(0,1,0,0,1,1,1,0)}=\min(1,2,1,1)=1$ и $c_{(0,1,0,0,1,0,0,1)}=\min(1,2,2)=1$

Минималните пат вектори за ниво 2 ќе бидат дадени во Табела 4.4. Минималните пат вектори за сите нивоа ќе бидат дадени во Табела 5 во Прилог А, страна 103.

$MDP_2' \rightarrow \mathcal{B}$	Вектори кои не се $\leq M$	Дупликати	MDP_2	MP_2	$M2TR_2$
$(2,0,2,0,2,-1,1,1) \rightarrow \{1,2\}$ $(2,0,1,1,1,-1,2,0) \rightarrow \{1,4\}$ $(1,1,1,0,2,-2,2,0) \rightarrow \{1,7\}$ $(1,1,1,0,2,-1,1,1) \rightarrow \{1,8\}$ $(2,0,1,1,1,0,2) \rightarrow \{2,3\}$ $(2,0,1,1,0,1,1) \rightarrow \{2,4\}$ $(1,1,1,0,2,-1,1,1) \rightarrow \{2,7\}$ $(1,1,1,0,2,0,0,2) \rightarrow \{2,8\}$ $(2,0,0,2,0,1,1,1) \rightarrow \{3,4\}$ $(1,1,-1,2,0,2,0,2) \rightarrow \{3,5\}$ $(1,1,-1,2,0,1,1,1) \rightarrow \{3,6\}$ $(1,1,0,1,1,1,0,2) \rightarrow \{3,8\}$ $(1,1,-1,2,0,1,1,1) \rightarrow \{4,5\}$ $(1,1,-1,2,0,0,2,0) \rightarrow \{4,6\}$ $(1,1,0,1,1,-1,2,0) \rightarrow \{4,7\}$ $(1,1,0,1,1,0,1,1) \rightarrow \{4,8\}$ $(0,2,-2,2,0,1,1,1) \rightarrow \{5,6\}$ $(0,2,-1,1,1,1,0,2) \rightarrow \{5,8\}$ $(0,2,-1,1,1,-1,2,0) \rightarrow \{6,7\}$ $(0,2,-1,1,1,0,1,1) \rightarrow \{6,8\}$ $(0,2,0,0,2,-1,1,1) \rightarrow \{7,8\}$	$(2,0,2,0,2,-1,1,1)$ $(2,0,1,1,1,-1,2,0)$ $(1,1,1,0,2,-2,2,0)$ $(2,0,0,2,0,1,1,1)$ $(1,1,-1,2,0,2,0,2)$ $(1,1,-1,2,0,1,1,1)$ $(1,1,-1,2,0,1,1,1)$ $(1,1,-1,2,0,0,2,0)$ $(1,1,0,1,1,-1,2,0)$ $(0,2,-2,2,0,1,1,1)$ $(0,2,-1,1,1,1,0,2)$ $(0,2,-1,1,1,-1,2,0)$ $(0,2,-1,1,1,0,1,1)$ $(0,2,0,0,2,-1,1,1)$	$(1,1,1,0,2,-1,1,1)$	$(1,1,1,0,2,-1,1,1)$ $(2,0,1,1,1,1,0,2)$ $(2,0,1,1,1,0,1,1)$ $(1,1,1,0,2,0,0,2)$ $(1,1,0,1,1,1,0,2)$ $(1,1,0,1,1,0,1,1)$	$(1,1,1,0,2,1,1,1)$ $(2,0,1,1,1,1,0,2)$ $(2,0,1,1,1,0,1,1)$ $(1,1,1,0,2,0,0,2)$ $(1,1,0,1,1,1,0,2)$ $(1,1,0,1,1,0,1,1)$	0,69793

Табела 4.4 Минимални пат вектори за ниво 2

За ниво 2 се добиваат 21 патишта, од кои 14 не се помали или еднакви од M и 1 е дупликат. За ниво 3 се добиваат 15 патишта од кои 14 не се помали или еднакви од M и нема дупликати.



Фигура 4.11

Во Фигура 4.11 е даден излезот од алгоритмот за примерот 4.4 (само почетокот), треба да се забележи дека времето на извршување на овој алгоритам е 31.25 ms.

4.3 Споредба на алгоритам за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$ и убрзаниот алгоритам за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$

Во овој дел ќе извршиме споредба на алгоритам за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$ и убрзаниот алгоритам преку примерите 4.1 и 4.3 кои се однесуваат за ист транспортен систем даден во фигура 4.1, и примерите 4.2 и 4.4 кои се однесуваат на транспортниот систем даден во фигура 4.9

Со првиот алгоритам за пример 4.1, за ниво 2 добивме 10 патишта од кои 4 не се помали или еднакви од векторот од максимални состојби M и еден е дупликат. За ниво 3 се добиваат 20 патишта од кои 11 не се помали или еднакви од M и 6 се дупликати. Додека за ниво 4 се добиваат 12 патишта од кои 9 не се помали или еднакви од M и 2 се дупликати.. Додека за истиот систем (пример 5.3) со примена на убрзаниот алгоритмот се добиваат 6 патишта, од кои од кои 1 не е помал или еднаков од M и нема дупликати. За ниво 3 се добиваат 5 патишта од кои 2 не се помали или еднакви од M и нема дупликати. За ниво 4 се добиваат 2 патишта од кои 1 не е помал или еднаков од M и нема дупликати. Значи дури и кога станува збор за мал транспортен систем се забележува дека бројот на дупликати и бројот на добиени вектори што не се помали од M се намалува.

Да извршиме сега споредба на алгоритмите на транспортниот систем даден со Фигура 4.9. Со примена на првиот алгоритам за ниво 2 се добиваат 36 патишта, од кои 24 не се помали или еднакви од M и 6 се дупликати. За ниво 3 се добиваат 48 патишта од кои 42 не се помали или еднакви од M и 5 се дупликати. Додека значително помалку патишта се добиваат со примена на убрзаниот алгоритмот. Имено за ниво 2 се добиваат 21 патишта, од кои 14 не се помали или еднакви од M и 1 е дупликат. За ниво 3 се добиваат 15 патишта од кои 14 не се помали или еднакви од M и нема дупликати. Разликата видно се забележува кај ниво 3 каде со првиот алгоритам се добиваат 48 патишта а со неговата модификација се добиваат 15 патишта.

Што се однесува до брзината на извршување на алгоритмите, времето потребно да се најдат минималните пат вектори за системот даден со Фигура 4.1 со примена на првиот алгоритам изнесува 31.25 ms, додека со примена на убрзаниот алгоритам е 15.63 ms. За вториот пример со првиот алгоритам времето на извршување изнесува 62.50 ms додека со убрзаниот алгоритам 31.25 ms. Разликите помеѓу времињата се многу мали, но овие системи се доста мали.

Подобрувањата кои се забележуваат кај убрзаниот алгоритмот се должат на тоа што со овој алгоритам се прави контрола исто множество од векторите од MDP_1 да не се повторува и со тоа се избегнува добивањето на дупликат вектори.

Исто така се врши контрола векторот да не се повторува повеќе пати, и со тоа се намалува бројот на вектори што се поголеми од векторот од максимални состојби M . Иако во примерите што ги разгледувавме транспортните системи се мали, алгоритмот е направен така што не се собираат векторите што имаат спротивен знак на иста координата, со што се елиминираат и дупликати векторите што би се добиле со собирање на вектори кои на иста координата имаат спротивен знак. Меѓутоа да напоменеме дека за мали системи не е неопходно да се врши проверка дали векторите на иста координата имаат спротивен знак, бидејќи самата проверка може да одземе многу време. Времето на извршување се намалува поради тоа што бројот на вектори што се собираат значително се намалува.

4.4 Убрзан алгоритам за наоѓање на минимални пат вектори за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$ со контрола на максималните нивоа

За да го опишеме овој алгоритам прво треба да ги дефинираме следниве функции и множества. Нека векторот $\vec{x}_i \in \text{MDP}_d$, е од облик $\vec{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$, а векторот $\vec{y}_j \in \text{MDP}_1$ е од облик $\vec{y}_j = (y_{j,1}, y_{j,2}, \dots, y_{j,n})$. Дефинираме функција $h: \text{MDP}_1 \rightarrow \{1, 2, \dots, n\}$ со $h(\vec{y}_i) = i$. Исто така за секој $\vec{z} \in \text{MDP}_1$ му придружуваме множество $\mathcal{H}_i = \{h(\vec{z}) \mid \vec{z} \in \text{MDP}_1, |z_i|=1\}$ за $i=1, \dots, n$ и за секој $\vec{y}_i \in \text{MDP}_1$, $\mathcal{Q}'_{\vec{y}_i} = \{h(\vec{y}_k) \mid \vec{y}_k \in \text{MDP}_1, k > i \text{ и } y_{ij} \cdot y_{kj} \geq 0, j=1, 2, \dots, n\}$.

Од множеството $\mathcal{Q}'_{\vec{y}_i}$ го формираме множеството $\mathcal{Q}_{\vec{y}_i}$ на следниот начин ако постои g така што $y_{ig} = M_g$, тогаш $\mathcal{Q}_{\vec{y}_i} = \mathcal{Q}'_{\vec{y}_i} / \{r \mid r \in \mathcal{H}_g\}$

Значи \mathcal{H}_i е множествата од вектори, чија што i -та координата е 1 или -1. Додека со $\mathcal{Q}_{\vec{x}}$ се дефинира множество од вектори со кои може да се собира векторот $\vec{x} \in \text{MP}_d$ за да се добијат вектори од множеството MPP_{d+1} .

Да го опишеме алгоритмот преку неговите чекори.

Убрзан алгоритам за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$ со контрола на максималните нивоа

Влез на овој алгоритам ќе бидат бинарните минимални пат вектори BPV , множеството од вектори на капацитети на компонентите S , и множеството од веројатности на нивоата на компонентите \mathcal{VP} .

Излез од алгоритмот ќе бидат минималните пат вектори за нивоа d и надежностите за соодветните нивоа.

Чекор 1. Се добиваат сите минимални насочени пат вектори за ниво 1, MDP_1 .

Чекор 2. Се дефинира функција $h:MDP_1 \rightarrow \{1, 2, \dots, L\}$ со $h(\vec{y}_j) = j$.

Чекор 3. Се формираат множествата $\mathcal{H}_i = \{h(\vec{z}) \mid \vec{z} \in MDP_1, |z_i|=1\}$, $i=1, \dots, n$

Чекор 4. За секој $\vec{y}_i \in MDP_1$, се формираат множествата

$$\mathcal{Q}'_{\vec{y}_i} = \{h(\vec{y}_k) \mid \vec{y}_k \in MDP_1, k > i \text{ и } y_{ij} \cdot y_{kj} \geq 0, j=1, 2, \dots, n\}.$$

Чекор 5. Се формираат множествата $\mathcal{Q}_{\vec{y}_i}$ на следниот начин ако постои g , така што $y_{ig} = M_g$, тогаш $\mathcal{Q}_{\vec{y}_i} = \mathcal{Q}'_{\vec{y}_i} / \{r \mid r \in \mathcal{H}_g\}$.

Чекор 6. Множеството MP_1 е добиено од множеството $\{Abs(\vec{x}) \mid \vec{x} \in MDP_1\}$

Чекор 7. Се иницијализира $d=1$

Чекор 8. Се формира множеството

$$MP_{d+1} = \{\vec{z} \mid \vec{z} = \vec{x} + \vec{y}, \vec{x} \in MP_d, \vec{y} \in MP_1 \text{ и } h(\vec{y}) \in \mathcal{Q}_{\vec{x}}\}$$

и за секој елемент од MP_{d+1} му се придружува множеството $\mathcal{Q}_{\vec{z}}$ кое се добива на следниот начин:

$$\mathcal{Q}'_{\vec{z}} = \mathcal{Q}_{\vec{x}} \cap \mathcal{Q}_{\vec{y}} (\vec{z} = \vec{x} + \vec{y})$$

$$\text{Ако постои } i \text{ така што } z_i = M_i, \text{ тогаш } \mathcal{Q}_{\vec{z}} = \mathcal{Q}'_{\vec{z}} / \{j \mid j \in \mathcal{H}_i\}$$

Чекор 9. Се додека $MP_{d+1} \neq \emptyset$, $d=d+1$ и оди на чекор 8 во спротивно оди на чекор 10.

Чекор 10. Кога минималните пат се добиени, надежноста се пресметува со користење на формулата за вклучување и исклучување.

Со програмот `najdi_minvektori_uzbrzan1` е имплементиран горниот алгоритам во C#.

4.5 Алгоритми за наоѓање на минимални пат вектори за транспортен систем чии капацитети на линкови може да примаат произволни целобројни вредности

Во овој дел ќе разгледаме како претходните три алгоритми може да биде модифицирани и да се применуваат за системи кај кои капацитетот на линкот i е целобројна вредност x_i , $0 \leq x_i \leq M_i$. Векторот од максималните состојби $M = (M_1, M_2, \dots, M_n)$.

Дефиниција 4.5 За векторот \vec{x} се вели дека е **минимален пат вектор на перформанси за ниво d (minimal performance path vector to level d)** MPP_d , ако $\varphi(\vec{x}) = d$ и за секој друг $\vec{y} < \vec{x}$, $\varphi(\vec{y}) < d$.

Разликата помеѓу минималните пат вектори на перформанси (MPP_d) и минималните пат вектори (MP_d), е во тоа што елементите на MP_d припаѓаат во множеството $E = S_1 \times S_2 \times \dots \times S_n$ (множество од состојби (state set)), додека елементи од MPP_d не мора да се елементи во E . Треба да забележиме дека ако $\vec{x} \in E$ тогаш \vec{x} е минимален пат вектор за ниво d .

Прво да дефинираме еквивалент на минималното насочено пат множество (MDP_d) за минималното пат множество на перформанси.

Дефиниција 4.6 Нека \vec{x} е минимален пат вектор на перформанси за ниво d , тогаш **минималниот насочен пат вектор на перформанси за ниво d** ($MDPP$ за ниво d), се добива кога во \vec{x} , x_i се заменува со $-x_i$, секогаш кога i - от линк е двонасочен и се користи негативната насока..

За овие типови од системи ние ќе го користиме претходните три алгоритми, но во овој случај наместо да се одредат минималните пат вектори за ниво d , овде ние ќе го добиеме минималните пат вектори на перформанси за ниво d . Потоа за секој елемент од MPP_d , ќе го најдеме најмалиот вектор $\vec{y} \in E$ така што $\vec{x} \leq \vec{y}$. Овој вектор е пат вектор за ниво d , но не е неопходно да биде минимален пат вектор за ниво d . Треба да се нагласи дека е имплементирана процедура во C# (najdi_vistinski) со која од минималните пат вектори на перформанси ќе ги добиеме минимални пат вектори. Со процедурата прво ќе ги најдеме сите минимални пат вектори но исто така ќе добиеме и вектори кои не се минимални како и дупликат вектори. Значи за да го добиеме множеството од минимални вектори мора да ги елиминираме сите пат вектори што не се минимални и сите дупликат вектори.

Сега да ги дадеме претходните три алгоритми за системи чии капацитети на линкови може да примаат произволни целобројни вредности

Алгоритми за транспортен систем со целобројни капацитети

Влез на овој алгоритам ќе бидат бинарните минимални пат вектори BPV , множеството од вектори на капацитети на компонентите S , и множеството од веројатности на нивоата на компонентите \mathcal{VP} .

Излез од алгоритмот ќе бидат минималните пат вектори за нивоа d и надежностите за соодветните нивоа.

Чекор 1. Со користење на програмите

(najdi_minvektori) или (najdi_minvektori_uzbrzan) или (najdi_minvektori_uzbrzan1) се наоѓаат множества од минимални пат вектори на перформанси.

Чекор 2. За секој d , со користење на множеството MPP_d , се конструираат множествата MP'_d на следниот начин: за секој елемент $\bar{x} \in MPP_d$ се наоѓа најмалиот елемент $\bar{y} \in E$ што е поголем од \bar{x}

Чекор 3. За секој d , од множествата MP'_d се добиваат множество од минималните пат вектори за ниво d , MP_d на следниот начин:

Чекор 3.1. Од множествата MP'_d се бришат сите елементи што се дупликати.

Чекор 3.2. Од множествата MP'_d се бришат оние елементи \bar{x} , за кои постои вектор $\bar{y} \in MP'_d$, т.ш. $\bar{y} < \bar{x}$ (\bar{x} е доминантен над \bar{y})

Чекор 4. Кога минималните пат множества се добиени, надежноста се пресметува со користење на формулата за вклучување и исклучување

Значи погоре се опишани трите модификации на предходните алгоритми во случај кога капацитетите на линковите се целобројни вредности. Значи се дадени алгоритмот за целобројни вредности, убрзаниот алгоритам за целобројни вредности и убрзаниот алгоритам за целобројни вредности со контрола на максималните нивоа.

Да разгледаме сега по еден пример за сите три алгоритми за системи со целобројни капацитети.

Пример 4.5:

Да го разгледаме транспортниот систем во Фигура 4.1 со примена на првиот алгоритам. Капацитети на линковите се $S_1=\{0,3,6\}$, $S_2=\{0,2,4\}$, $S_3=\{0,2\}$, $S_4=\{0,2\}$ и $S_5=\{0,1,3,5\}$. Веројатностите со кои линковите може да се најдат во некоја од состојбите се $p_1=(0.1,0.1,0.8)$, $p_2=(0.1,0.1,0.8)$, $p_3=(0.1,0.9)$, $p_4=(0.1,0.9)$ и $p_5=(0.1,0.05,0.05,0.8)$.

Се претпоставува дека бинарните пат вектори кои се и минимални пат вектори за ниво 1, се познати однапред. Значи множествата од минимални перформирачки пат вектори, минимални перформиначки насочени пат вектори за ниво 1 се:

$$MPP_1 = \{(1,0,1,0,1), (1,0,0,1,0), (0,1,1,1,0), (0,1,0,0,1)\}$$

$$MDPP_1 = \{(1,0,1,0,1), (1,0,0,1,0), (0,1,-1,1,0), (0,1,0,0,1)\}$$

Да разгледаме како од минималните пат вектори на перформанси се наоѓаат минималните пат вектори.

За векторот (1,0,1,0,1) тоа ќе биде векторот (3,0,2,0,1) бидејќи тоа е најмалиот вектор поголем од (1,0,1,0,1) што припаѓа во $E = S_1 \times S_2 \times S_3 \times S_4 \times S_5$. На ист начин се наоѓаат и останатите минимални пат вектори. Па множеството од минимални пат вектори

$$MP_1 = \{(3,0,2,0,1), (3,0,0,2,0), (0,2,2,2,0), (0,2,0,0,1)\}$$

Надежност за ниво 1: $M2TR_1 = 0.97848$.

Минималните пат вектори на перформанси и минималните пат вектори за ниво 2 и 3 ќе бидат дадени во Табела 4.5 Додека, минималните пат вектори на перформанси и минималните пат вектори за сите нивоа ќе бидат дадени во Табела 6 во Прилог А, на страна 104.

d	MPP_d	MP'_d	MP_d	$M2TR_d$
2	(2,0,2,0,2)	(3,0,2,0,3)	(3,0,2,0,3)	0,97317
	(2,0,1,1,1)	(3,0,2,2,1)	(3,0,0,2,0)	
	(1,1,0,1,1)	(3,2,0,2,1)	(0,2,2,2,0)	
	(1,1,1,0,2)	(3,2,2,0,3)	(0,2,0,0,3)	
	(2,0,0,2,0)	(3,0,0,2,0)		
	(1,1,1,2,0)	(3,2,2,2,0)		
	(0,2,2,2,0)	(0,2,2,2,0)		
	(0,2,1,1,1)	(0,2,2,2,1)		
	(0,2,0,0,2)	(0,2,0,0,3)		
	3	(3,0,2,1,1)	(3,0,2,2,3)	
(2,1,1,1,2)		(3,2,2,2,3)	(3,0,2,2,1)	
(2,1,2,0,3)		(3,2,2,0,3)	(3,2,0,2,1)	
(3,0,1,2,1)		(3,0,2,2,1)	(0,4,2,2,1)	
(2,1,0,2,1)		(3,2,0,2,1)	(0,4,0,0,3)	
(1,2,1,2,1)		(3,2,2,2,1)		
(1,2,0,1,2)		(3,2,0,2,3)		
(1,2,1,0,3)		(3,2,2,0,3)		
(0,3,2,2,1)		(0,4,2,2,1)		
(0,3,1,1,2)		(0,4,2,2,3)		
(0,3,0,0,3)		(0,4,0,0,3)		

Табела 4.5 Минимални пат вектори за ниво 2 и 3

Со користење на програмот najdi_minvektori и процедурата najdi_vistinski го добиваме следниот излез.



Фигура 4.12

Треба да се забележи дека времето на извршување е 140.63 ms.

Ако го примениме програмот najdi_minvektori_uzbrzan и процедурата najdi_vistinski на претходниот пример го добиваме следниот излез.



Фигура 4.13

Се забележува дека времето на извршување сега е 62,50 ms, а со претходниот алгоритам беше 140,63. Ова намалување на времето на извршување како што објаснивме и претходно се должи на намалениот број на вектори што се собираат, подари тоа што во убрзаниот алгоритам се воведени контроли кои вектори може да се собираат.

Во Табелата 4.6 ќе бидат дадените бројот на пат вектори, бројот на вектори што се дупликати како и бројот на вектори што се поголеми од векторот на максимални состојби M што се добиваат кога првиот алгоритам и убрзаниот алгоритам се применуваат на транспортниот систем за пример 4.5

Алгоритам за целобројни капацитети				Убрзан алгоритам за целобројни капацитети			
d	Пат вектори	Дупликати	Вектори кои не се $\leq M$	D	Пат вектори	Дупликати	Вектори кои не се $\leq M$
1	4	0	0	1	4	0	0
2	10	0	1	2	9	0	0
3	36	7	15	3	13	2	0
4	44	10	20	4	14	2	0
5	48	18	21	5	14	4	0
6	36	18	13	6	9	4	0
7	20	13	5	7	4	2	0

Табела 4.6

Како што може да се виде во табелата 4.6 бројот на пат вектори што се добиваат со алгоритмот за целобројни вредности е значајно поголем од бројот на вектори што се добиваат со убрзаниот алгоритам за целобројни вредности. Истото важи и за векторите што се дупликати и за векторите што не се помали од векторот M . За примерот 4.5 со убрзаниот алгоритам за целобројни вредности не се добива ни еден дупликат вектор, додека со алгоритмот за целобројни вредности нивниот број е значително голем.

Пример 4.6

Да го примениме алгоритмот за целобројни вредности со контрола на максималните нивоа на транспортниот систем од пример 4.5. Капацитети на линковите се $S_1=\{0,3,6\}$, $S_2=\{0,2,4\}$, $S_3=\{0,2\}$, $S_4=\{0,2\}$ и $S_5=\{0,1,3,5\}$. Веројатностите со кои линковите може да се најдат во некоја од состојбите се $p_1=(0.1,0.1,0.8)$, $p_2=(0.1,0.1,0.8)$, $p_3=(0.1,0.9)$, $p_4=(0.1,0.9)$ и $p_5=(0.1,0.05,0.05,0.8)$.

Се претпоставува дека бинарните пат вектори кои се и минимални пат вектори за ниво 1, се познати однапред. Значи множествата од минимални пат вектори на перформанси и минимални насочени пат вектори на перформанси за ниво 1 се:

$$MPP_1 = \{(1,0,1,0,1), (1,0,0,1,0), (0,1,1,1,0), (0,1,0,0,1)\}$$

$$MDPP_1 = \{(1,0,1,0,1), (1,0,0,1,0), (0,1,-1,1,0), (0,1,0,0,1)\}$$

Да ги најдеме минималните пат вектори за минималните перформирачки пат вектори.

За векторот $(1,0,1,0,1)$ тоа ќе биде векторот $(3,0,2,0,1)$ бидејќи тоа е најмалиот вектор поголем од $(1,0,1,0,1)$ што припаѓа во $E = S_1 \times S_2 \times S_3 \times S_4 \times S_5$. На ист начин се наоѓаат и останатите минимални пат вектори. Па множеството од минимални пат вектори

$$MP_1 = \{(3,0,2,0,1), (3,0,0,2,0), (0,2,2,2,0), (0,2,0,0,1)\}$$

Дефинираме функција $h: \text{MDPP}_1 \rightarrow \{1,2,3,4\}$, така што
 $h((1,0,1,0,1))=1$, $h((1,0,0,1,0))=2$, $h((0,1,-1,1,0))=3$ и $h((0,1,0,0,1))=4$

Да ги најдеме сега множествата \mathcal{H}_i

$$\mathcal{H}_1=\{1,2\}, \mathcal{H}_2=\{3,4\}, \mathcal{H}_3=\{1,3\}, \mathcal{H}_4=\{2,3\}, \mathcal{H}_5=\{1,4\}$$

$$\mathcal{Q}'_{\vec{z}} = \mathcal{Q}'_{\vec{x}} \cap \mathcal{Q}'_{\vec{y}} (\vec{z} = \vec{x} + \vec{y})$$

Да ги формираме прво множествата $\mathcal{Q}'_{\vec{x}}$ за сите елементи $\vec{x} \in \text{MDPP}_1$.

$$\mathcal{Q}'_{(1,0,1,0,1)} = \{1,2,4\}, \mathcal{Q}'_{(1,0,0,1,0)} = \{2,3,4\}, \mathcal{Q}'_{(0,1,-1,1,0)} = \{3,4\}, \mathcal{Q}'_{(0,1,0,0,1)} = \{4\}$$

За формирање на множества $\mathcal{Q}'_{\vec{x}}$ потребно е да се разгледаат сите вектори од MDPP_1 и да се видиме дали некој од векторите има координата што е еднаква со соодветната координата од векторот $M=\{6,4,2,2,5\}$. Јасно нека не постои таков вектор (бидејќи координатите на вектори од MDPP_1 се 0 и 1, а во M немаме координата 1.

$$\text{Значи } \mathcal{Q}_{(1,0,1,0,1)} = \mathcal{Q}'_{(1,0,1,0,1)} = \{1,2,4\}, \mathcal{Q}_{(1,0,0,1,0)} = \mathcal{Q}'_{(1,0,0,1,0)} = \{2,3,4\}$$

$$\mathcal{Q}_{(0,1,-1,1,0)} = \mathcal{Q}'_{(0,1,-1,1,0)} = \{3,4\} \text{ и } \mathcal{Q}_{(0,1,0,0,1)} = \mathcal{Q}'_{(0,1,0,0,1)} = \{4\}$$

Надежност за ниво 1: $M2TR_1=0.97848$.

Да најдеме некои од минималните пат вектори за ниво 2.

Бидејќи $\mathcal{Q}_{(1,0,1,0,1)} = \{1,2,4\}$, првиот вектор можеме да го собираме со 1,2 и 4-от вектор. Ги добиваме следниве вектори:

$$(2,0,2,0,2) = (1,0,1,0,1) + (1,0,1,0,1)$$

$$\mathcal{Q}'_{(2,0,2,0,2)} = \mathcal{Q}'_{(1,0,1,0,1)} \cap \mathcal{Q}'_{(1,0,1,0,1)} = \{1,2,4\} \cap \{1,2,4\} = \{1,2,4\}.$$

Бидејќи третата координата на векторот $(2,0,2,0,2)$ е 2 и е еднаква на $M_3=2$, за да го добиеме множеството $\mathcal{Q}_{(2,0,2,0,2)}$ потребно е да се најде разликата од множествата $\mathcal{Q}_{(2,0,2,0,2)}$ и \mathcal{H}_3 . Значи имаме:

$$\mathcal{Q}_{(2,0,2,0,2)} = \mathcal{Q}'_{(2,0,2,0,2)} / \mathcal{H}_3 = \{1,2,4\} / \{1,3\} = \{2,4\}$$

$$(2,0,1,1,1) = (1,0,1,0,1) + (1,0,0,1,0)$$

$$\mathcal{Q}'_{(2,0,1,1,1)} = \mathcal{Q}'_{(1,0,1,0,1)} \cap \mathcal{Q}'_{(1,0,0,1,0)} = \{1,2,4\} \cap \{2,3,4\} = \{2,4\}$$

$$\mathcal{Q}_{(2,0,1,1,1)} = \mathcal{Q}'_{(2,0,1,1,1)} = \{2,4\}$$

$$(1,1,1,0,2) = (1,0,1,0,1) + (0,1,0,0,1)$$

$$\mathcal{Q}'_{(1,1,1,0,2)} = \mathcal{Q}'_{(1,0,1,0,1)} \cap \mathcal{Q}'_{(0,1,0,0,1)} = \{1,2,4\} \cap \{4\} = \{4\}$$

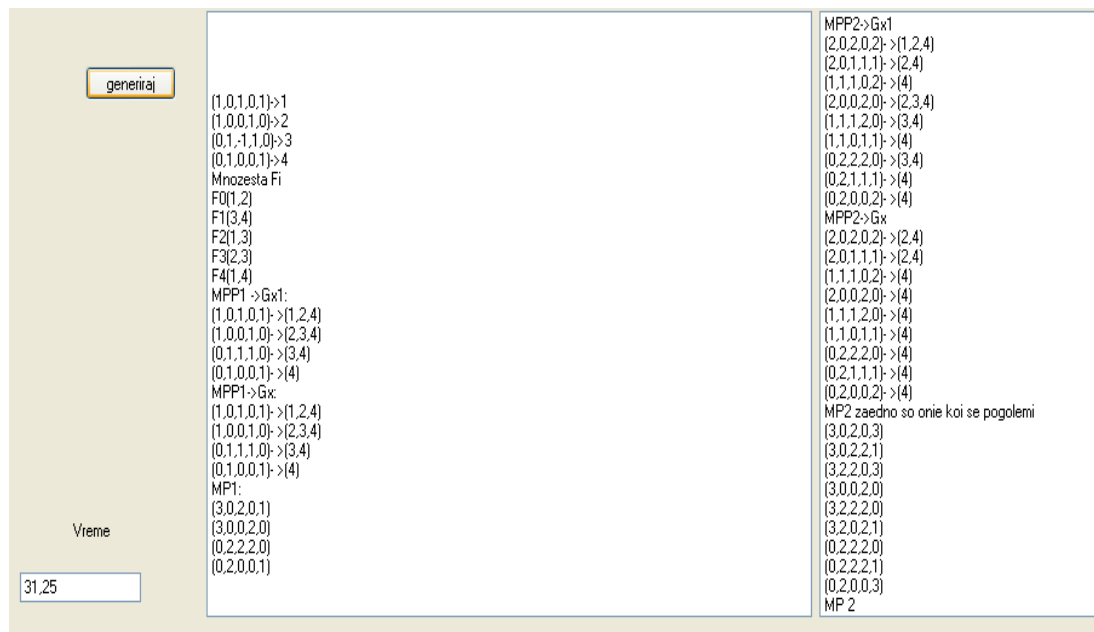
$$\mathcal{Q}_{(1,1,1,0,2)} = \mathcal{Q}'_{(1,1,1,0,2)} = \{4\}$$

Минималните пат вектори за ниво 2 и 3 ќе бидат во Табела 4.6. Минималните пат вектори за сите нивоа ќе бидат дадени во Табела 7 во Прилог А, страна 105.

d	$MPP_d \rightarrow Q'_x$	$MPP_d \rightarrow Q_x$	MP_d'	MP_d	$M2TR_d$
2	(2,0,2,0,2) → {1,2,4} (2,0,1,1,1) → {2,4} (1,1,1,0,2) → {4} (2,0,0,2,0) → {2,3,4} (1,1,1,2,0) → {3,4} (1,1,0,1,1) → {4} (0,2,2,2,0) → {3,4} (0,2,1,1,1) → {4} (0,2,0,0,2) → {4}	(2,0,2,0,2) → {2,4} (2,0,1,1,1) → {2,4} (1,1,1,0,2) → {4} (2,0,0,2,0) → {4} (1,1,1,2,0) → {4} (1,1,0,1,1) → {4} (0,2,2,2,0) → {4} (0,2,1,1,1) → {4} (0,2,0,0,2) → {4}	(3,0,2,0,3) (3,0,2,2,1) (3,2,2,0,3) (3,0,0,2,0) (3,2,2,2,0) (3,2,0,2,1) (0,2,2,2,0) (0,2,2,2,1) (0,2,0,0,3)	(3,0,2,0,3) (3,0,0,2,0) (0,2,2,2,0) (0,2,0,0,3)	0,91317
3	(3,0,2,1,1) → {2,4} (2,1,2,0,3) → {4} (3,0,1,2,1) → {2,4} (2,1,1,1,2) → {4} (1,2,1,0,3) → {4} (2,1,0,2,1) → {4} (1,2,1,2,1) → {4} (1,2,0,1,2) → {4} (0,3,2,2,1) → {4} (0,3,1,1,2) → {4} (0,3,0,0,3) → {4}	(3,0,2,1,1) → {2,4} (2,1,2,0,3) → {4} (3,0,1,2,1) → {4} (2,1,1,1,2) → {4} (1,2,1,0,3) → {4} (2,1,0,2,1) → {4} (1,2,1,2,1) → {4} (1,2,0,1,2) → {4} (0,3,2,2,1) → {4} (0,3,1,1,2) → {4} (0,3,0,0,3) → {4}	(3,0,2,2,3) (3,2,2,0,3) (3,0,2,2,1) (3,2,2,2,3) (3,2,2,0,3) (3,2,0,2,1) (3,2,2,2,1) (3,2,0,2,3) (0,4,2,2,1) (0,4,2,2,3) (0,4,0,0,3)	(3,2,2,0,3) (3,0,2,2,1) (3,2,0,2,1) (0,4,2,2,1) (0,4,0,0,3)	0,86104

Табела 4.6 Минимални пат вектори за ниво 2 и 3

Ако го примениме програмот najdi_minvektori_uzbrzan1 и процедурата najdi_vistinski на предходниот пример го добиваме следниот излез.



Фигура 4.14

Се забележува дека времето на извршување сега е 31,25, што е помало од времето на извршување кога се користи убрзаниот алгоритамот за целобројни

вредности, кој е еднакво на 62,50 ms, а со алгоритмот за целобројни вредности времето на извршување е 140,63 ms. Ова намалување на времето на извршување како што објаснивме и претходно се должи на намалениот број на вектори што се собираат, поради тоа што со убрзаниот алгоритмот за целобројни вредности со контрола на максималните нивоа не се добиваат дупликати вектори, како и вектори кои се поголеми од векторот од максимални состојби M .

4.6 Алгоритам за наоѓање на минимални пат вектори за транспортен систем со произволни капацитети на линковите

Овој алгоритам се разликува од претходните алгоритми за наоѓање на минимални пат вектори на двотерминален повеќе - состојбен транспортен систем, бидејќи тој не побарува било каква рестрикција за вредностите на капацитетите на линковите. Значи во овој случај множествата од состојби може да примат било кој реален број.

Својство 4.5 Нека $\vec{x}_k \in \text{BPV}$ за транспортниот систем $G=(\mathcal{N}, \mathcal{A}, \text{BPV}, S, \mathcal{VP})$ и $\alpha_k \in \mathbf{R}$, тогаш $\vec{x} = \sum_{k=1}^d \alpha_k \vec{x}_k$ е минимален пат вектор за ниво d .

Својство 4.6 Ако \vec{x} е MP_d за транспортниот систем $G=(\mathcal{N}, \mathcal{A}, \text{BPV}, S, \mathcal{VP})$ тогаш постојат вектори $\vec{x}_k \in \text{BPV}$ и $\alpha_k \in \mathbf{R}$, $k=1, \dots, d$, така што $\vec{x} = \sum_{k=1}^d \alpha_k \vec{x}_k$.

И овој алгоритам како и претходните, бара бинарното минимално пат множество за систем со истата структура да биде познато однапред.

Со J' ќе го означиме множеството од бинарните минимални насочени пат вектори (binary minimal direction path vectors) и со J множеството од бинарни минимални пат вектори (binary minimal path vectors).

Дефиниција 4.7 Нека S_i е вектор на состојби на линкот a_i и $S'_i = \{S_{i_2}, \dots, S_{i_{|S_i|}}\}$. За

секој $\vec{x} \in \mathbf{R}^n$, се дефинира множество $U_{\vec{x}}$ на следниот начин:

$$U_{\vec{0}} = S'_i$$

Нека $\vec{x} \in \mathbf{R}^n$ со $U_{\vec{x}} = \{U_{\vec{x}_1}, \dots, U_{\vec{x}_n}\}$ и $\vec{y} \in J$ така што $U_{\vec{x}_i} \neq \emptyset$ за сите i за кои $y_i=1$.

Тогаш за

$$d \leq \min_{i, y_i=1} U_{\vec{x}_i}(l) \quad (U_{\vec{x}_i}(l) \text{ е } l \text{ - от елемент во } U_{\vec{x}_i})$$

$$U_{\vec{x}+d\vec{y}} = \{U_{(\vec{x}+d\vec{y})_1}, \dots, U_{(\vec{x}+d\vec{y})_n}\}, \text{ каде}$$

$$U_{(\bar{x}+d\bar{y})_i} = \begin{cases} U_{\bar{x}_i}, & y_i = 0 \\ \{U_{\bar{x}_i}(1) - d, \dots, U_{\bar{x}_i}(|U_{\bar{x}_i}|) - d\} & y_i = 1, d > U_{\bar{x}_i}(1) \\ \{U_{\bar{x}_i}(2) - d, \dots, U_{\bar{x}_i}(|U_{\bar{x}_i}|) - d\} & y_i = 1, d = U_{\bar{x}_i}(1) \end{cases}$$

Алгоритам за реални капацитети

Влез на овој алгоритам ќе бидат бинарните минимални пат вектори BPV , множеството од вектори на капацитети на компонентите S , и множеството од веројатности на нивоата на компонентите \mathcal{VP} .

Излез од алгоритмот ќе бидат минималните пат вектори за нивоа d и надежностите за соодветните нивоа.

Чекор 1. Најди го множеството J'

Чекор 2. $J' = \{\bar{y}_1, \bar{y}_2, \dots, \bar{y}_L\}$ се дефинира функција

$$h: J' \rightarrow \{1, 2, \dots, L\} \text{ со } h(\bar{y}_i) = i$$

Чекор 3. За секој $\bar{y}_i \in J'$ се формира множеството

$$K_i = \{h(\bar{z}) \mid \bar{z} \in J', y_{ij} \cdot z_j \geq 0, j=1, 2, \dots, n\}$$

Чекор 4. Се наоѓа множеството $J = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_L\}$, $\bar{x}_i = (|y_{i1}|, \dots, |y_{in}|)$

Чекор 5. Множеството $V = \{\bar{0}, 0\}$, $V' = \emptyset$, $W_0 = \{1, 2, \dots, L\}$ и се наоѓа множеството $U_{\bar{0}}$.

Чекор 6. За секое \bar{v} , така што $\{\bar{v}, d\} \in V$ и за секое $i \in W_{\bar{v}}$ се наоѓа $c = \{j \mid \bar{x}_j \in J, x_{ij} \neq 0\}$ и $m = \min\{U_{\bar{v}_j}(1) \mid U_{\bar{v}_j}(1) > 0 \text{ и } j \in c\}$

Елементот $(\bar{v} + m\bar{x}_i, m+d)$ се става V' .

Множеството $U_{\bar{v}+m\bar{x}_i}$ е најдено со користење на дефиницијата 4.7.

$$\text{Множеството } H_{\bar{v}+m\bar{x}_i} = (W_{\bar{v}} \cap K_i) / \{r \mid \text{за } U_{(\bar{v}+m\bar{x}_i)_r} = \emptyset\}$$

Чекор 7. Ги бришеме сите дупликат елементи од V' .

Чекор 8. $d = \min\{d \mid \{\bar{v}, d\} \in V'\}$

Чекор 9. $MPP_{d'} = \{\bar{v} \mid \{\bar{v}, d\} \in V'\}$

Чекор 10. Со користење на множеството $MPP_{d'}$ се формира множеството $MP_{d'}$ на следниот начин: за секој елемент $\bar{v} \in MPP_{d'}$ се наоѓа најмалиот елемент $\bar{y} \in E$ што е поголем од \bar{v} .

Чекор 11. Се бришат сите елементи $\bar{x} \in MP_{d'}$, за кои постои $\bar{y} \in MP_{d'}$, $\bar{x} \geq \bar{y}$

Чекор 12. $V \leftarrow \{\{\bar{v}, d\} \in V' \mid d = d'\}$, и соодветните множества U и W се придружени на тој начин.

Чекор 13. $V' \leftarrow V' \setminus V$

Чекор 14: Се повторуваат чекорите 6, 7, 8, 9, 10, 11, 12 и 13 додека $V' \neq \emptyset$.

Пример 4.7

Алгоритмот го применуваме на транспортниот систем од пример 4.5. Капацитети на линковите се $S_1=\{0,3,6\}$, $S_2=\{0,2,4\}$, $S_3=\{0,2\}$, $S_4=\{0,2\}$ и $S_5=\{0,1,3,5\}$. Веројатностите со кои линковите може да се најдат во некоја од состојбите се $p_1=(0.1,0.1,0.8)$, $p_2=(0.1,0.1,0.8)$, $p_3=(0.1,0.9)$, $p_4=(0.1,0.9)$ и $p_5=(0.1,0.05,0.05,0.8)$.

Множеството од бинарни насочени пат вектори е $J=\{(1,0,1,0,1), (1,0,0,1,0), (0,1,-1,1,0), (0,1,0,0,1)\}$. Подредувањето на векторите и множество од вектори може да бидат додадени како што се дадени во Табела 4.7 :

J	(1,0,1,0,1)	(1,0,0,1,0)	(0,1,1,1,0)	(0,1,0,0,1)
F	1	2	3	4
K_i	{1,2,4}	{1,2,3,4}	{2,3,4}	{1,2,3,4}

Табела 4.7 Бинарни вектори

Ние ќе ги дадеме првите неколку чекори од алгоритмот. Иницијалните вредности за множествата V , $U_{\vec{v}}$ и $W_{\vec{v}}$ се дадени во Табела 4.8.

V		$U_{\vec{v}}$	$W_{\vec{v}}$
V	d	1 2 3 4 5	
(0,0,0,0,0)	0	3 2 2 2 1	
		6 4 3	{1,2,3,4}
		5	

Табела 4.8 Иницијални вредности

Првите 3 чекори од алгоритмот се претставени во табелите Табела 4.9, Табела 4.10. Првата и третата колона ги даваат векторите што се користат во процедурата за наоѓање на минимални пат вектори за следните нивоа; додека втората го дава потокот што може да биде пренесен низ системот, кога тој е во состојба \vec{v} . m е максималниот поток што може да биде пренесен со користење на i - от бинарен вектор и систем со множество од сите можни состојби (set spaces) $U_{\vec{v}}$. Добиениот вектор е \vec{x} , додека $m+d$ е потокот што треба да биде пренесен низ системот кога тој е во состојба \vec{x} .

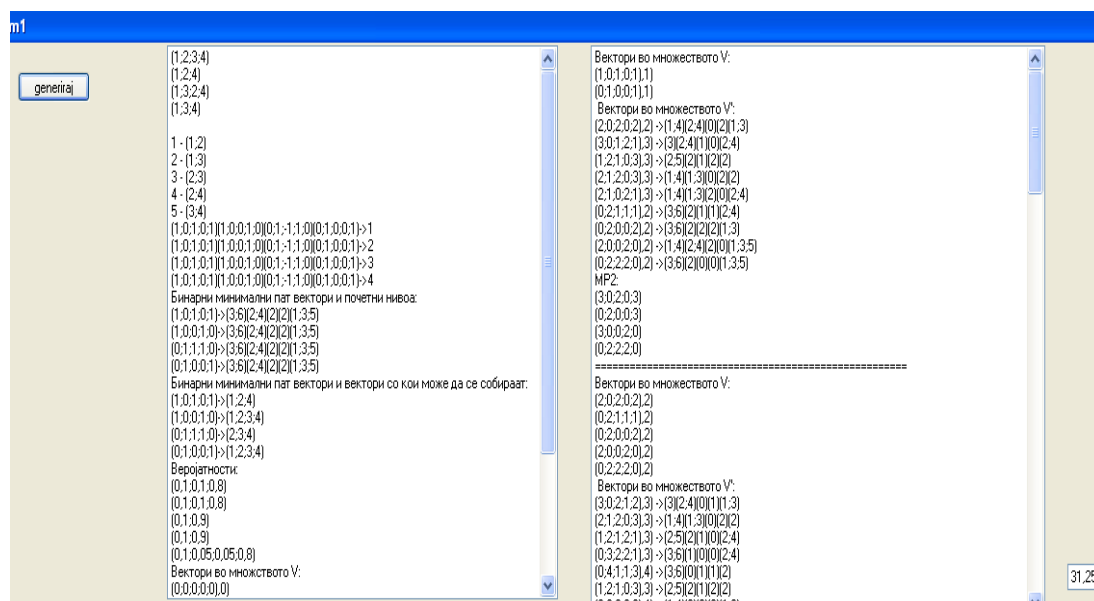
V						$U_{\bar{v}+m\bar{x}}$	$W_{\bar{v}+m\bar{x}}$	
V	d	$i \in W_{\bar{v}}$	m	\bar{x}	m+d	1 2 3 4 5		
(0,0,0,0,0)	0	1	1	(1,0,1,0,1)	1	2 2 1 2 2 5 4 4	{1,2,4}	V
		2	2	(2,0,0,2,0)	2	1 2 2 0 1 4 4 3 5	{1,3}	V'
		3	2	(0,2,2,2,0)	2	3 2 0 0 1 3 5	{4}	V
		4	1	(0,1,0,0,1)	1	3 1 2 2 2 6 3 4	{1,2,3,4}	V
MP ₁ ={ (3,0,2,0,1), (3,0,0,2,0), (0,2,2,2,0), (0,2,0,0,1) }								
M2TR ₁ =0,97848								

Табела 4.9

V						$U_{\bar{v}+m\bar{x}}$	$W_{\bar{v}+m\bar{x}}$	
V	d	$i \in W_{\bar{v}}$	m	\bar{x}	m+d	1 2 3 4 5		
				(2,0,0,2,0)	2	1 2 2 0 1 4 4 3 5	{1,4}	V
				(0,2,2,2,0)	2	3 2 0 0 1 3 5	{4}	V
(1,0,1,0,1)	1	1	1	(2,0,2,0,2)	2	1 2 0 2 1 4 4 3	{2,4}	V
		2	2	(3,0,1,2,1)	3	3 2 1 0 2 4 4	{1,4}	V'
		3	2	(1,2,1,0,3)	3	2 2 1 2 2 5	{1,2,4}	V'
(0,1,0,0,1)	2	1	2	(2,1,2,0,3)	3	1 1 0 2 2 4 3	{2,4}	V'
		2	2	(2,1,0,2,1)	3	1 1 2 0 2 4 3 4	{1,4}	V'
		3	1	(0,2,1,1,1)	2	3 2 1 1 2 6 4	{2,3,4}	V
		4	1	(0,2,0,0,2)	2	3 2 2 2 1 6 3	{1,2,3,4}	V
MP ₂ ={ (3,0,2,0,3), (0,2,0,0,3), (3,0,0,2,0), (0,2,2,2,0) }								
M2TR ₂ =0,97317								

Табела 4.10

Минималните пат вектори за останатите нивоа се дадени во Табела 8, Прилог А, на страна 106.



Фигура 4.15

Се забележува дека времето на извршување сега е 31,25, што е еднакво со времето на извршување кога се користи убрзаниот алгоритамот за целобројни вредности со контрола на максималните нивоа, но е помало од времето на извршување кога се користи убрзаниот алгоритамот за целобројни вредности, каде времето е 62,50 ms, а со алгоритамот за целобројни вредности времето на извршување е 140,63 ms.

Со овој алгоритам како што кажавме капацитетите на линковите можат да бидат кои било реални броеви. Па да разгледаме еден таков пример:

Пример 4.8

Повторно го разгледуваме транспортниот систем претставен со фигура 4.1. Капацитети на линковите се $S_1=\{0,1,5,3\}$, $S_2=\{0,2,4\}$, $S_3=\{0,1,2\}$, $S_4=\{0,1,2\}$ и $S_5=\{0,2,4\}$. Веројатностите со кои линковите може да се најдат во некоја од состојбите се $p_1=(0.01,0.09,0.9)$, $p_2=(0.01,0.09,0.9)$, $p_3=(0.05,0.25,0.7)$, $p_4=(0.08,0.12,0.8)$ и $p_5=(0.05,0.05,0.9)$. Т.е. го разгледуваме примерот 3.1 од Глава 3.

Множеството од бинарни насочени пат вектори е $J'=\{(1,0,1,0,1), (1,0,0,1,0), (0,1,-1,1,0), (0,1,0,0,1)\}$. Подредувањето на векторите и множество од вектори може да бидат додадени како што се дадени во Табела 4.11 :

J	(1,0,1,0,1)	(1,0,1,0,1)	(1,0,1,0,1)	(0,1,0,0,1)
H	1	2	3	4
K_i	{1,2,4}	{1,2,3,4}	{2,3,4}	{1,2,3,4}

Табела 4.11 Бинарни вектори

Иницијалните вредности за множествата V , $U_{\bar{v}}$ и $W_{\bar{v}}$ ќе бидат дадени во Табела 4.12

V		$U_{\bar{v}}$					$W_{\bar{v}}$
V	d	1	2	3	4	5	
(0,0,0,0,0)	0	1,5	2	1	1	2	{1,2,3,4}
		3	4	2	2	4	

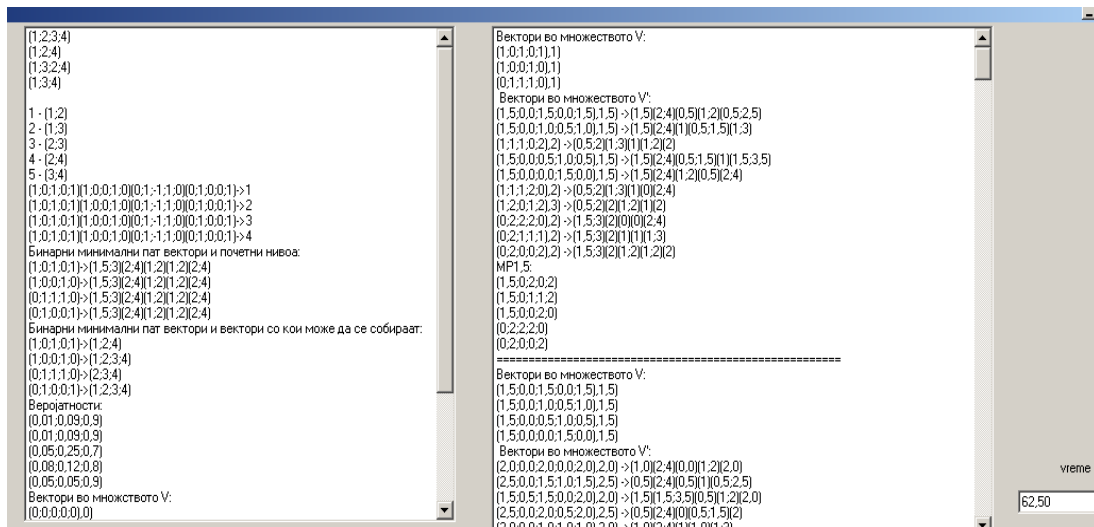
Табела 4.12 Иницијални вредности

Бидејќи овој алгоритам детално е објаснет во Примерот 4.7, овде само во Табела 4.13 ќе ги дадеме минимални пат вектори за нивоата 1, 1.5 и 2 како и надежностите за соодветните нивоа. Минимални пат вектори за сите нивоа ќе бидат дадени во Табела 9 во Прилог А, на страна 107.

d	Минимални пат множества за ниво d	$M2TR_d$
	a_1 a_2 a_3 a_4 a_5	
1	1,5 0 1 0 2 1,5 0 0 1 0 0 2 1 1 0 0 2 0 0 2	0,99584
1,5	1,5 0 2 0 2 1,5 0 1 1 2 1,5 0 0 2 0 0 2 2 2 0 0 2 0 0 2	0,9895
2	3 0 2 0 2 3 0 1 1 2 3 0 0 2 0 1,5 2 1 2 0 0 2 2 2 0 0 2 0 0 2	0,98846

Табела 4.13 Минимални пат вектори за ниво 1, 1.5 и 2

Да го дадеме и излезот на овој алгоритам.



Фигура 4.16

Времето на извршување за овој пример изнесува 62.50 ms.

4.7 Споредба на алгоритмите дадени во Глава 3 со алгоритам за реални капацитети

Во овој дел ќе направиме споредба помеѓу алгоритам за пресметување на минимални пат множества на Ramirez-Marquez и Coit за транспортни системи со реални капацитети, алгоритам за пресметување на минимални пресек множества на Ramirez-Marquez, Coit и Tortorella за транспортни системи со слабо хомогени компоненти кои се дадени во глава 3 и алгоритмот за наоѓање на минимални пат вектори за транспортен систем со произволни капацитети на линковите дадени во глава 4.

Се наметнува прашањето зошто во глава 4 се предложени нови алгоритми во за наоѓање на пат множества за повеќе – состојбен двотерминален транспортен систем.

Предноста на и алгоритмот за наоѓање на минимални пат вектори за транспортен систем со произволни капацитети на линковите дадени во глава 4, се состои во тоа што овој алгоритам не бара никакви ограничувања за вредностите што може да ги примат капацитетите на линковите. Алгоритмот предложен во [6] исто така не бара никакви ограничувања на линковите, меѓутоа како што е веќе кажано во Глава 3, со овој алгоритам не се наоѓаат сите минимални пат вектори. Да извршиме сега споредба на резултатите за Примерот 3.1 дадени во Табела 2 и Табела 9 во Прилог А, страна 99 и 107. За ниво 1, со примена на и алгоритмот за наоѓање на минимални пат вектори за транспортен систем со произволни капацитети на линковите дадени во глава 4, се добива векторот $(0,2,0,0,2)$ што е минимален пат за ниво 1, но овој вектор не се добива со примена на алгоритмот предложен во [6]. Исто така и за останатите нивоа не се најдени сите минимални пат вектори, на пример за ниво 3,5 минимални пат вектори се $(0,4,2,2,2)$ и $(0,4,0,0,4)$ но тие не се најдени со алгоритмот предложен во [5].

Алгоритмот даден во [9] исто така не бара ограничување на вредностите на капацитетите на линковите, меѓутоа со овој алгоритам на самиот почеток се врши трансформација на капацитетите на линковите до хомоген облик. Значи алгоритмот работи под претпоставка дека линковите се слабо хомогени. Откако минималните пресек вектори ќе бидат најдени потоа се трансформираат до хетероген облик. Значи овој алгоритам ќе работи добро само доколку компонентите на системот се слабо хомогени.

Другата предност на предложениот алгоритам се состои во тоа што со овој алгоритам се наоѓаат минималните пат вектори за сите нивоа. Значи ние го знаеме максималното ниво на работа на транспортниот систем. Алгоритмите разгледани во глава 3 ги наоѓаат минималните пат вектори, т.е. минималните пресек вектори за однапред зададено ниво. Значи со овие алгоритми не може да се знае максималното ниво на работа на транспортниот систем. Единствено може да се претпостави кое е максимално ниво на работа според вредностите на капацитетите што ги имаат линковите.

Предноста максималното ниво на системот да биде познато се состои во тоа што на тој начин може да се вршат дополнителни анализи за подобрување

на надежноста на системот. Односно, може да вршиме анализи која од компоненти на системот е најдобро да биде заменета за да се добие подобрен систем. Се разбира овде освен на зголемувањето на надежноста треба да водиме сметка и за цената на компонентите што сакаме да ги заменуваме.

Исто така треба да нагласиме дека предложениот алгоритам во глава 4 има уште една предност која се состои во тоа што со овој алгоритам имаме скокање на нивоа, со што се зголемува негова брзина.

5. Заклучок

Во овој магистерски труд прво е извршена анализа на два постоечки алгоритми за наоѓање на минимални пат вектори и минимални пресек множества, предложени во [6] и [9].

Извршена е анализа на овие алгоритми и дадени се недостатоците што ги имаат алгоритмите. Алгоритмите се имплементирани во програмскиот јазик C#.

Во магистерскиот труд се предложени три алгоритми за наоѓање на минимални пат множества. Предноста на предложените алгоритми се состои во тоа што овие алгоритми се едноставни за разбирање и се базираат на едноставни операции помеѓу векторите. Исто така се разгледува и проблемот на двонасочни линкови. Овој проблем се разрешува со тоа што едната насока од линкот се зема како позитивна и се означува со 1, а останата насока како негативна и се означува со -1.

Првиот од трите предложени алгоритми работи само за транспортен систем со капацитети на линкови $\{0,1,2,\dots,M_i\}$. а овој алгоритам е дадена и негова модификации и споредени се нивните брзини преку примери. Вториот алгоритам работи само доколку вредностите на капацитетите на линковите се целобројни. И за овој алгоритам се дадени две негови модификации со цел да се забрза неговата работа.

Од најголемо значење е последниот алгоритам бидејќи тој нема ограничување на вредностите на капацитетите на линковите.

Треба да нагласиме дека со сите овие алгоритми се добиваат минималните пат вектори за сите нивоа, што не е случај за алгоритмите предложени во [6] и [9] кај кои однапред е одредени нивото за кое ќе се бараат минималните пат или минималните пресек вектори.

Сите овие алгоритми се имплементирани во програмскиот јазик C#. Со помош на овие имплементации се врши анализа на брзината на алгоритмите.

Врз база на минималните пат вектори може да се пресмета и надежноста на системот. Имплементиран е и алгоритам во C# за наоѓање на надежноста на системот доколку минималните пат вектори се познати. При пресметување на надежноста се користи формулата за вклучување и исклучување.

Понатамошните истражуваат во оваа област ќе одат во две насоки.

Првата насока е да се добијат побрзи алгоритми за наоѓање на минимални пат множества. Една идеја за забрзување на алгоритмите е со поделба на транспортниот систем на подсистеми, и барање на минималните пат вектори на секој од подсистемите. Сметаме дека ќе се забрза работата на алгоритмите бидејќи тие ќе бидат применети на помал систем, па и векторите што ќе ги собираме ќе бидат со помала големина.

Другата насока е добивање на подобрени транспортни системи со заменување на некоја од постоечките компоненти на системот. Бидејќи со предложените алгоритми ние го знаеме максималното ниво на работа на системот, и соодветната надежност за тоа ниво, па истражувањата ќе бидат насочени кон добивање на подобрени системи кои ќе работат со поголемо максимално ниво. Значи ќе вршиме анализи кои и колку компоненти е најдобро

да се заменат за да се добие подобрен транспортен систем. Овде треба да нагласиме дека мора да се води сметка и на тоа колкаво ќе биде зголемувањето на трошоците. Значи, истражувањата ќе бидат насочени кон добивање на подобрен транспортен систем со минимално зголемување на трошоците.

Користена литература

- [1] A.Lisnianski, G.Levitin, "Multi-State system Reliability," *The Israel Electric Corporation Limited*, 2003 Israel
- [2] A.Wilson, N.Limnios, S.Keller Mc-Nulty, Y.Armijo, "Modern Statistical and Mathematical Methods in Reliability," *World Scientific Publishing*, 2005
- [3] B.Natvig, S.Sørmo, A.Holen and G. Høgåsen, "Multistate Reliability Theory – A Case Study," *Advances in Applied Probability*, Vol 18, 1986. pp. "921-932
- [4] D. Torrierri, "Calculation of Node-Pair Reliability in Large Networks with Unreliable Node," *IEEE Transactions on Reliability*, Vol 43, n 3, 1994. pp. 375-379.
- [5] E.Hansler, "A Fast Recursive Algorithm to Calculate the Reliability of a Communication Network," *IEEE Transactions on Communications*, Vol 20, n 3, 1972. pp. 637-640 "
- [6] J.E. Ramirez-Marquez and D.W. Coit, "Alternative Approach for Analyzing Multi-state Network Reliability," *Proceedings of the Industrial Engineering Research Conference (IERC)*, Portland, OR, May 2003"
- [7] J.E. Ramirez-Marquez and D.W. Coit, "Composite Importance Measures for Multi-State Systems with Multi-state Components," *IEEE Transactions on Reliability*, Vol 54, No3, September 2005
- [8] J.E. Ramirez-Marquez and D.W.Coit, "Multi-state Component Criticality Analysis for Reliability Improvement in Multi-state Systems," *Reliability Engineering & System Safety*, Vol. 92, No. 10, October 2007.
- [9] J.E Ramirez-Marquez, D.W. Coit and M.Tortorella, "Multi-state Two-terminal Reliability: A Generalized Cut-Set Approach," *Rutgers University IE*, Working Paper
- [10] K.Aggarwal, "Integration of Reliability and Capacity in Performance Measure of a Telecommunication Network," *IEEE Transactions on Reliability*, Vol 34, n 2, 1985. pp. 184-186
- [11] K.Aggarwal, Y.Chopra and J.Bajwa, "Capacity Consideration in Reliability Analysis of Communication Systems," *IEEE Transactions on Reliability*, Vol 31, n 2, 1982. pp. 177-180.

- [12] Martin. L. Shooman, “Reliability of computer systems and networks,” *John Wiley and Sons*, 2002
- [13] Ming J. Zuo, Way Kuo, “Optimal Reliability Modeling: Principles and Applications,” *John Wiley and Sons*, 2002
- [14] Min Xie, Yuan-Shun Dai, Kim-Leng Poh, “Computing System Reliability,” *Kluwer Academic Publishers*, 2004
- [15] M.J.Crowder, A.C. Kimber, R.L. Smith, T.J. Sweeting, “Statistical Analysis of Reliability Dat,” *Chapman and Hall*, 1991 London
- [16] M. Mihova, “An algorithm for calculating multi-state network reliability using minimal path vectors,” *Bulgarian Academy of Sciences (IPP-BAS)*
- [17] M. Михова, “Моделирање на надежност на непоправливи повеќе-состојбени системи со независни компоненти и хомогени маркови транзиции ,” докторска дисертација, *Природно-математички факултет*, Скопје 2008
- [18] M. Михова, “Моделирање на надежност на системи со повеќе состојби,” магистерски труд, *Природно-математички факултет*, Скопје 2004
- [19] M. Mihova, N. Maksimova, Z. Popeska, “An algorithm for calculating multi-state network reliability with arbitrary capacities of the links,” - *Fourth International Bulgarian-Greek Conference Computer Science*, 2008, pp.70-75
- [20] S. Kuo, S. Lu and F. Yeh, “Determining Terminal Pair Reliability Based on Edge Expansion Diagrams Using OBDD,” *IEEE Transactions on Reliability*, Vol 48, n 3, 1999. pp. 234-246
- [21] S. Lee, “Reliability Evaluation of a Flow network,” *IEEE Transactions on Reliability*, Vol 29, n1, 1980. pp. 24-26.
- [22] S. Patra and B. Misra, “Evaluation of Probability Mass Function of Flow in a Communication Network Considering a Multi-state Model of Network Links,” *Microelectronics and Reliability*, Vol 36, n 3, 1996. pp. 415-421.
- [23] S. Patra, S. and B. Misra, “Reliability Evaluation of Flow Networks Considering Multi-state Modeling of Network Elements,” *Microelectronics and Reliability*, Vol 33, n 14, 1993. pp. 2161-2164.
- [24] W. Dotson and J. Gobien, “A New Analysis Technique for Probabilistic Graphs,” *IEEE Transactions on Circuits and Systems*, Vol 26, n 10, 1979. pp. 855-865.

- [25] W.Yeh , “A simple MC-based Algorithm for Evaluating Reliability of a Stochastic-Flow Using Minimal Cuts to Evaluate the System Reliability of a Stochastic-Flow Network with Unreliable Nodes,” *Reliability Engineering & System Safety*, 2003
- [26] Y.Lin, “A Simple Algorithm for Reliability Evaluation of a Stochastic-Flow Network with Node Failur,” *Computers and Operations Research*, Vol 28, 2001. pp. 1277-1285.
- [27] Y.Lin, “Using Minimal Cuts to Evaluate the System Reliability of a Stochastic-Flow Network with Failures at Nodes and Arcs,” *Reliability Engineering & System Safety*, Vol 75, 2002. pp. 41-46.
- [28] Y.Yoo and N. Deo, “A Comparison of Algorithms for Terminal Pair Reliability,” *IEEE Transactions on Reliability*, Vol 37, n 2, 1988. pp. 210-215.

Прилог А

	<i>Настан</i>	<i>Исход</i>	<i>Број на настани</i>
<i>Без неуспеси</i>	$E_1=123456$	“Добар”	$\binom{6}{0} = \frac{6!}{0!6!} = 1$
<i>Еден неуспех</i>	$E_2=1\ 23456$	“Добар”	$\binom{6}{1} = \frac{6!}{1!5!} = 6$
	$E_3=12\ 3456$	“Добар”	
	$E_4=123\ 456$	“Добар”	
	$E_5=1234\ 56$	“Добар”	
	$E_6=12345\ 6$	“Добар”	
	$E_7=123456$	“Добар”	
<i>Два неуспеха</i>	$E_8=1\ 2\ 3456$	“Добар”	$\binom{6}{2} = \frac{6!}{2!4!} = 15$
	$E_9=1\ 23\ 456$	“Добар”	
	$E_{10}=1\ 234\ 56$	“Добар”	
	$E_{11}=1\ 2345\ 6$	“Добар”	
	$E_{12}=1\ 23456$	“Добар”	
	$E_{13}=12\ 3\ 456$	“Добар”	
	$E_{14}=12\ 34\ 56$	“Добар”	
	$E_{15}=12\ 345\ 6$	“Добар”	
	$E_{16}=12\ 3456$	“Добар”	
	$E_{17}=123\ 4\ 56$	“Добар”	
	$E_{18}=123\ 45\ 6$	“Добар”	
	$E_{19}=123\ 456$	“Добар”	
	$E_{20}=1234\ 5\ 6$	“Добар”	
	$E_{21}=1234\ 56$	“Добар”	
$E_{22}=12345\ 6$	“Добар”		
<i>Три неуспеха</i>	$E_{23}=1234\ 5\ 6$	“Добар”	$\binom{6}{3} = \frac{6!}{3!3!} = 20$
	$E_{24}=123\ 45\ 6$	“Добар”	
	$E_{25}=123\ 4\ 56$	“Добар”	
	$E_{26}=123\ 4\ 5\ 6$	“Добар”	
	$E_{27}=12\ 345\ 6$	“Добар”	
	$E_{28}=12\ 34\ 56$	“Добар”	
	$E_{29}=12\ 34\ 5\ 6$	“Добар”	
	$E_{30}=12\ 3\ 456$	“Добар”	
	$E_{31}=12\ 3\ 45\ 6$	“Добар”	
	$E_{32}=12\ 3\ 4\ 56$	“Добар”	
	$E_{33}=1\ 2345\ 6$	“Добар”	
	$E_{34}=1\ 2345\ 6$	“Добар”	
	$E_{35}=1\ 234\ 5\ 6$	“Лош”	
	$E_{36}=1\ 2\ 3456$	“Лош”	
	$E_{37}=1\ 2\ 345\ 6$	“Добар”	
	$E_{38}=1\ 23\ 4\ 56$	“Добар”	
	$E_{39}=1\ 23\ 456$	“Добар”	

	$E_{40}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	
	$E_{41}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	
	$E_{42}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	
Четири неуспеха	$E_{43}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	$\binom{6}{4} = \frac{6!}{4!2!} = 15$
	$E_{44}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	
	$E_{45}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	
	$E_{46}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	
	$E_{47}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	
	$E_{48}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{49}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{50}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	
	$E_{51}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{52}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{53}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{54}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{55}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{56}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	
$E_{57}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”		
Пет неуспеха	$E_{58}=1\ 2\ 3\ 4\ 5\ 6$	“Добар”	$\binom{6}{5} = \frac{6!}{5!1!} = 6$
	$E_{59}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{60}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{61}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{62}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
	$E_{63}=1\ 2\ 3\ 4\ 5\ 6$	“Лош”	
Шест неуспеха	$E_{64}=1\ 2\ 3\ 4\ 5\ 6^k$	“Лош”	$\binom{6}{6} = \frac{6!}{6!0!} = 1$

Табела 1

d	Минимални пат вектори за ниво d	$M2TR_d$
0	0 0 0 0 0	1
1	0 2 1 1 0 1.5 0 1 0 2 1.5 0 0 1 0	0.99584
1.5	1.5 0 2 0 2 1.5 0 1 1 2 1.5 0 0 2 0	0.9895
2	3 0 2 0 2 3 0 1 1 2 3 0 0 2 0 1.5 2 1 2 0 0 2 2 2 0 0 2 0 0 2	0.98846
3	3 0 2 1 2 3 0 1 2 2 1.5 2 1 0 4 0 4 1 1 2 1.5 2 0 1 2	0.94256
3.5	1.5 2 2 0 4 1.5 2 1 1 4 1.5 2 0 2 2	0.92724
4	3 0 2 0 2 3 0 1 1 2 3 0 0 2 0 1.5 2 1 2 0 0 2 2 2 0 0 2 0 0 2	0.91986
5	3 2 1 2 4 3 2 2 1 4 1.5 4 0 1 4	0.79928
5.5	1.5 4 0 2 4	0.64152
6	3 4 0 2 4	0.5832

Табела 2. Минимални пат вектори за транспортниот систем од Пример 3.1

d	MDP_d'	Вектори кои не се $\leq M$	Дупликати	MDP_d	MP_d	$M2TR_d$
2	$(1,0,1,0,1)+(1,0,1,0,1)=(2,0,2,0,2)$ $(1,0,1,0,1)+(1,0,0,1,0)=(2,0,1,1,1)$ $(1,0,1,0,1)+(0,1,-1,1,0)=(1,1,0,1,1)$ $(1,0,1,0,1)+(0,1,0,0,1)=(1,1,1,0,2)$ $(1,0,0,1,0)+(1,0,0,1,0)=(2,0,0,2,0)$ $(1,0,0,1,0)+(0,1,-1,1,0)=(1,1,-1,2,0)$ $(1,0,0,1,0)+(0,1,0,0,1)=(1,1,0,1,1)$ $(0,1,-1,1,0)+(0,1,-1,1,0)=(0,2,-2,2,0)$ $(0,1,-1,1,0)+(0,1,0,0,1)=(0,2,-1,1,1)$ $(0,1,0,0,1)+(0,1,0,0,1)=(0,2,0,0,2)$	$(2,0,0,2,0)$ $(2,0,2,0,2)$ $(1,1,-1,2,0)$ $(0,2,-2,2,0)$	$(1,1,0,1,1)$	$(2,0,1,1,1)$ $(1,1,0,1,1)$ $(1,1,1,0,2)$ $(0,2,-1,1,1)$ $(0,2,0,0,2)$	$(2,0,1,1,1)$ $(1,1,0,1,1)$ $(1,1,1,0,2)$ $(0,2,1,1,1)$ $(0,2,0,0,2)$	0.84614
3	$(2,0,1,1,1)+(1,0,1,0,1)=(3,0,2,1,2)$ $(2,0,1,1,1)+(1,0,0,1,0)=(3,0,1,2,1)$ $(2,0,1,1,1)+(0,1,-1,1,0)=(2,1,0,2,1)$ $(2,0,1,1,1)+(0,1,0,0,1)=(2,1,1,1,2)$ $(1,1,0,1,1)+(1,0,1,0,1)=(2,1,1,1,2)$ $(1,1,0,1,1)+(1,0,0,1,0)=(2,1,0,2,1)$ $(1,1,0,1,1)+(0,1,-1,1,0)=(1,2,-1,2,1)$ $(1,1,0,1,1)+(0,1,0,0,1)=(1,2,0,1,2)$ $(1,1,1,0,2)+(1,0,1,0,1)=(2,1,2,0,3)$ $(1,1,1,0,2)+(1,0,0,1,0)=(2,1,1,1,2)$ $(1,1,1,0,2)+(0,1,-1,1,0)=(1,2,0,1,2)$ $(1,1,1,0,2)+(0,1,0,0,1)=(1,2,1,0,3)$ $(0,2,-1,1,1)+(1,0,1,0,1)=(1,2,0,1,2)$ $(0,2,-1,1,1)+(1,0,0,1,0)=(1,2,-1,2,1)$ $(0,2,-1,1,1)+(0,1,-1,1,0)=(0,3,-2,2,1)$ $(0,2,-1,1,1)+(0,1,0,0,1)=(0,3,-1,1,2)$ $(0,2,0,0,2)+(1,0,1,0,1)=(1,2,1,0,3)$ $(0,2,0,0,2)+(1,0,0,1,0)=(1,2,0,1,2)$ $(0,2,0,0,2)+(0,1,-1,1,0)=(0,3,-1,1,2)$ $(0,2,0,0,2)+(0,1,0,0,1)=(0,3,0,0,3)$	$(3,0,2,1,2)$ $(3,0,1,2,1)$ $(2,1,0,2,1)$ $(2,1,0,2,1)$ $(1,2,-1,2,1)$ $(2,1,2,0,3)$ $(1,2,-1,2,1)$ $(0,3,-1,1,2)$ $(0,3,-1,1,2)$ $(0,1,-1,3,2)$ $(0,3,0,0,3)$	$(2,1,1,1,2)$ $(2,1,1,1,2)$ $(1,2,0,1,2)$ $(1,2,0,1,2)$ $(1,2,0,1,2)$ $(1,2,1,0,3)$	$(2,1,1,1,2)$ $(1,2,0,1,2)$ $(1,2,1,0,3)$	$(2,1,1,1,2)$ $(1,2,0,1,2)$ $(1,2,1,0,3)$	0.64584
4	$(2,1,1,1,2)+(1,0,1,0,1)=(3,1,2,1,3)$ $(2,1,1,1,2)+(1,0,0,1,0)=(3,1,1,2,2)$ $(2,1,1,1,2)+(0,1,-1,1,0)=(2,2,0,2,2)$ $(2,1,1,1,2)+(0,1,0,0,1)=(2,2,1,1,3)$ $(1,2,0,1,2)+(1,0,1,0,1)=(2,2,1,1,3)$ $(1,2,0,1,2)+(1,0,0,1,0)=(2,2,0,2,2)$ $(1,2,0,1,2)+(0,1,-1,1,0)=(1,3,-1,2,2)$ $(1,2,0,1,2)+(0,1,0,0,1)=(1,3,0,1,3)$ $(1,2,1,0,3)+(1,0,1,0,1)=(2,2,2,0,4)$ $(1,2,1,0,3)+(1,0,0,1,0)=(2,2,1,1,3)$ $(1,2,1,0,3)+(0,1,-1,1,0)=(1,3,0,1,3)$ $(1,2,1,0,3)+(0,1,0,0,1)=(1,3,1,0,4)$	$(3,1,2,1,3)$ $(3,1,1,2,2)$ $(2,2,0,2,2)$ $(2,2,0,2,2)$ $(1,3,-1,2,2)$ $(1,3,0,1,3)$ $(2,2,2,0,4)$ $(1,3,0,1,3)$ $(1,3,1,0,4)$	$(2,2,1,1,3)$ $(2,2,1,1,3)$	$(2,2,1,1,3)$	$(2,2,1,1,3)$	0.3686

Табела 3. Минимални пат вектори за транспортниот систем од Пример 4.1

d	MDP_d	Вектори кои не се $\leq M$	Дупликати	MDP_d	MP_d	$M2TR_d$
2	(2,0,2,0,2,-2,2,0) (2,0,2,0,2,-1,1,1) (2,0,1,1,1,0,1,1) (2,0,1,1,1,-1,2,0) (1,1,0,1,1,0,1,1) (1,1, 0,1,1,-1,2,0) (1,1,1,0,2,-2,2,0) (1,1,1,0,2,-1,1,1) (2,0,2,0,2,0,0,2) (2,0,1,1,1,1,0,2) (2,0,1,1,1,0,1,1) (1,1,0,1,1,1,0,2) (1,1,0,1,1,0,1,1) (1,1,1,0,2,-1,1,1) (1,1,1,0,2,0,0,2) (2,0,0,2,0,2,0,2) (2,0,0,2,0,1,1,1) (1,1,-1,2,0,2,0,2) (1,1,-1,2,0,1,1,1) (1,1,0,1,1,0,1,1) (1,1,0,1,1,1,0,2) (1,1,0,1,1,0,1,1) (1,1,1,0,2,-1,1,1) (1,1,1,0,2,0,0,2) (2,0,0,2,0,2,0,2) (2,0,0,2,0,1,1,1) (1,1,-1,2,0,2,0,2) (1,1,-1,2,0,1,1,1) (1,1,0,1,1,0,1,1) (1,1,0,1,1,1,0,2) (2,0,0,2,0,0,2,0) (1,1,-1,2,0,1,1,1) (1,1,-1,2,0,0,2,0) (1,1,0,1,1,-1,2,0) (1,1,0,1,1,0,1,1) (0,2,-2,2,0,2,0,2) (0,2,-2,2,0,1,1,1) (0,2,-1,1,1,0,1,1) (0,2,-1,1,1,1,0,2) (0,2,-2,2,0,0,2,0) (0,2,-1,1,1,-1,2,0) (0,2,-1,1,1,0,1,1) (0,2,0,0,2,-2,2,0) (0,2,0,0,2,-1,1,1) (0,2,0,0,2,0,0,2)	(2,0,2,0,2,-2,2,0) (2,0,2,0,2,-1,1,1) (2,0,1,1,1,-1,2,0) (1,1, 0,1,1,-1,2,0) (1,1,1,0,2,-2,2,0) (2,0,2,0,2,0,0,2) (2,0,0,2,0,1,1,1) (1,1,-1,2,0,2,0,2) (1,1,-1,2,0,1,1,1) (2,0,0,2,0,0,2,0) (1,1,-1,2,0,1,1,1) (1,1,-1,2,0,0,2,0) (1,1,0,1,1,-1,2,0) (0,2,-2,2,0,2,0,2) (0,2,-2,2,0,1,1,1) (0,2,-1,1,1,0,1,1) (0,2,-1,1,1,1,0,2) (0,2,-2,2,0,0,2,0) (0,2,-1,1,1,-1,2,0) (0,2,-1,1,1,0,1,1) (0,2,0,0,2,-2,2,0) (0,2,0,0,2,-1,1,1) (0,2,0,0,2,0,0,2)	(2,0,1,1,1,0,1,1) (1,1,0,1,1,0,1,1) (1,1,0,1,1,0,1,1) (1,1,0,1,1,0,1,1) (1,1,1,0,2,-1,1,1) (1,1,0,1,1,0,1,1) (1,1,1,0,2,-1,1,1) (1,1,0,1,1,1,0,2) (1,1,0,1,1,1,0,2) (1,1,1,0,2,0,0,2)	(2,0,1,1,1,0,1,1) (1,1,0,1,1,0,1,1) (1,1,1,0,2,-1,1,1) (2,0,1,1,1,0,2) (1,1,0,1,1,1,0,2) (1,1,1,0,2,0,0,2)	(2,0,1,1,1,0,1,1) (1,1,0,1,1,0,1,1) (1,1,1,0,2,1,1,1) (2,0,1,1,1,1,0,2) (1,1,0,1,1,1,0,2) (1,1,1,0,2,0,0,2)	0,69793
3	(3,0,2,1,2,-1,2,1) (3,0,2,1,2,0,1,2) (3,0,1,2,1,1,1,2) (3,0,1,2,1,0,2,1) (2,1,0,2,1,1,1,2) (2,1,0,2,1,0,2,1) (2,1,1,1,2,-1,2,1) (2,1,1,1,2,0,1,2) (2,1,1,1,2,-1,2,1) (2,1,1,1,2,-1,2,1) (2,1,1,1,2,0,1,2) (2,1,0,2,1,1,1,2) (2,1,0,2,1,0,2,1) (1,2,-1,2,1,1,1,2) (2,1,0,2,1,1,1,2) (2,1,0,2,1,0,2,1) (1,2,-1,2,1,1,1,2) (1,2,-1,2,1,0,2,1) (1,2,0,1,2,-1,2,1) (1,2,0,1,2,0,1,2) (2,1,2,0,3,-2,2,1) (2,1,2,0,3,-1,1,2) (2,1,1,1,2,-1,2,1)	(3,0,2,1,2,-1,2,1) (3,0,2,1,2,0,1,2) (3,0,1,2,1,1,1,2) (3,0,1,2,1,0,2,1) (2,1,0,2,1,1,1,2) (2,1,0,2,1,0,2,1) (2,1,1,1,2,-1,2,1) (2,1,1,1,2,-1,2,1) (2,1,0,2,1,1,1,2) (2,1,0,2,1,0,2,1) (1,2,-1,2,1,1,1,2) (1,2,-1,2,1,0,2,1) (1,2,0,1,2,-1,2,1) (1,2,0,1,2,0,1,2) (2,1,2,0,3,-2,2,1) (2,1,2,0,3,-1,1,2) (2,1,1,1,2,-1,2,1)	(2,1,1,1,2,0,1,2) (2,1,1,1,2,0,1,2) (2,1,1,1,2,0,1,2) (2,1,1,1,2,0,1,2) (2,1,1,1,2,0,1,2)	(2,1,1,1,2,0,1,2)	(2,1,1,1,2,0,1,2)	0,37675

(2,1,2,0,3,-1,1,2)	(1,2,0,1,2,0,1,2)				
(2,1,1,1,2,0,1,2)	(1,2,0,1,2,-1,2,1)				
(2,1,1,1,2,-1,2,1)	(1,2,1,0,3,-2,2,1)				
(1,2,0,1,2,0,1,2)	(1,2,1,0,3,-1,1,2)				
(1,2,0,1,2,-1,2,1)	(3,0,2,1,2,0,1,2)				
(1,2,1,0,3,-2,2,1)	(3,0,2,1,2,1,0,3)				
(1,2,1,0,3,-1,1,2)	(3,0,1,2,1,2,0,3)				
(3,0,2,1,2,0,1,2)	(3,0,1,2,1,1,1,2)				
(3,0,2,1,2,1,0,3)	(2,1,0,2,1,2,0,3)				
(3,0,1,2,1,2,0,3)	(2,1,0,2,1,1,1,2)				
(3,0,1,2,1,1,1,2)	(2,1,1,1,2,1,0,3)				
(2,1,0,2,1,2,0,3)	(2,1,1,1,2,1,0,3)				
(2,1,0,2,1,1,1,2)	(2,1,0,2,1,2,0,3)				
(2,1,1,1,2,0,1,2)	(2,1,0,2,1,1,1,2)				
(2,1,1,1,2,1,0,3)	(1,2,-1,2,1,2,0,3)				
(2,1,1,1,2,0,1,2)	(1,2,-1,2,1,1,1,2)				
(2,1,1,1,2,1,0,3)	(1,2,0,1,2,0,1,2)				
(2,1,0,2,1,2,0,3)	(1,2,0,1,2,1,0,3)				
(2,1,0,2,1,1,1,2)	(2,1,2,0,3,-1,1,2)				
(1,2,-1,2,1,2,0,3)	(2,1,2,0,3,0,0,3)				
(1,2,-1,2,1,1,1,2)	(2,1,1,1,2,1,0,3)				
(1,2,0,1,2,0,1,2)	1,2,0,1,2,1,0,3)				
(1,2,0,1,2,1,0,3)	(1,2,0,1,2,0,1,2)				
(2,1,2,0,3,-1,1,2)	(1,2,1,0,3,-1,1,2)				
(2,1,2,0,3,0,0,3)	(1,2,1,0,3,0,0,3)				
(2,1,1,1,2,1,0,3)					
(2,1,1,1,2,0,1,2)					
(1,2,0,1,2,1,0,3)					
(1,2,0,1,2,0,1,2)					
(1,2,1,0,3,-1,1,2)					
(1,2,1,0,3,0,0,3)					

Табела 4. Минимални пат вектори за транспортниот систем од Пример 4.2

d	$MDP_d \rightarrow \mathcal{B}$	Вектори кои не се $\leq M$	Дупликати	MDP_d	MP_d	$M2TR_d$
2	$(2,0,2,0,2,-1,1,1) \rightarrow \{1,2\}$ $(2,0,1,1,1,-1,2,0) \rightarrow \{1,4\}$ $(1,1,1,0,2,-2,2,0) \rightarrow \{1,7\}$ $(1,1,1,0,2,-1,1,1) \rightarrow \{1,8\}$ $(2,0,1,1,1,0,2) \rightarrow \{2,3\}$ $(2,0,1,1,1,0,1,1) \rightarrow \{2,4\}$ $(1,1,1,0,2,-1,1,1) \rightarrow \{2,7\}$ $(1,1,1,0,2,0,0,2) \rightarrow \{2,8\}$ $(2,0,0,2,0,1,1,1) \rightarrow \{3,4\}$ $(1,1,-1,2,0,2,0,2) \rightarrow \{3,5\}$ $(1,1,-1,2,0,1,1,1) \rightarrow \{3,6\}$ $(1,1,0,1,1,1,0,2) \rightarrow \{3,8\}$ $(1,1,-1,2,0,1,1,1) \rightarrow \{4,5\}$ $(1,1,-1,2,0,0,2,0) \rightarrow \{4,6\}$ $(1,1,0,1,1,-1,2,0) \rightarrow \{4,7\}$ $(1,1,0,1,1,0,1,1) \rightarrow \{4,8\}$ $(0,2,-2,2,0,1,1,1) \rightarrow \{5,6\}$ $(0,2,-1,1,1,1,0,2) \rightarrow \{5,8\}$ $(0,2,-1,1,1,-1,2,0) \rightarrow \{6,7\}$ $(0,2,-1,1,1,0,1,1) \rightarrow \{6,8\}$ $(0,2,0,0,2,-1,1,1) \rightarrow \{7,8\}$	$(2,0,2,0,2,-1,1,1)$ $(2,0,1,1,1,-1,2,0)$ $(1,1,1,0,2,-2,2,0)$ $(2,0,0,2,0,1,1,1)$ $(1,1,-1,2,0,2,0,2)$ $(1,1,-1,2,0,1,1,1)$ $(1,1,-1,2,0,1,1,1)$ $(1,1,-1,2,0,0,2,0)$ $(1,1,0,1,1,-1,2,0)$ $(0,2,-2,2,0,1,1,1)$ $(0,2,-1,1,1,1,0,2)$ $(0,2,-1,1,1,-1,2,0)$ $(0,2,-1,1,1,0,1,1)$ $(0,2,0,0,2,-1,1,1)$	$(1,1,1,0,2,-1,1,1)$	$(1,1,1,0,2,-1,1,1)$ $(2,0,1,1,1,1,0,2)$ $(2,0,1,1,1,0,1,1)$ $(1,1,1,0,2,0,0,2)$ $(1,1,0,1,1,1,0,2)$ $(1,1,0,1,1,0,1,1)$	$(1,1,1,0,2,1,1,1)$ $(2,0,1,1,1,1,0,2)$ $(2,0,1,1,1,0,1,1)$ $(1,1,1,0,2,0,0,2)$ $(1,1,0,1,1,1,0,2)$ $(1,1,0,1,1,0,1,1)$	0,69793
3	$(2,1,2,0,3,-1,1,2) \rightarrow \{1,8,2\}$ $(2,1,1,1,2,-1,2,1) \rightarrow \{1,8,4\}$ $(1,2,1,0,3,-2,2,1) \rightarrow \{1,8,7\}$ $(3,0,1,2,1,1,1,2) \rightarrow \{2,3,4\}$ $(2,1,1,1,2,1,0,3) \rightarrow \{2,3,8\}$ $(3,0,2,1,2,-1,2,1) \rightarrow \{2,4,1\}$ $(2,1,1,1,2,-1,2,1) \rightarrow \{2,4,7\}$ $(2,1,1,1,2,0,1,2) \rightarrow \{2,4,8\}$ $(1,2,1,0,3,-1,1,2) \rightarrow \{2,8,7\}$ $(2,1,0,2,1,1,1,2) \rightarrow \{3,8,4\}$ $(1,2,-1,2,1,2,0,3) \rightarrow \{3,8,5\}$ $(1,2,-1,2,1,1,1,2) \rightarrow \{3,8,6\}$ $(1,2,-1,2,1,1,1,2) \rightarrow \{4,8,5\}$ $(1,2,-1,2,1,0,2,1) \rightarrow \{4,8,6\}$ $(1,2,0,1,2,-1,2,1) \rightarrow \{4,8,7\}$	$(2,1,2,0,3,-1,1,2)$ $(2,1,1,1,2,-1,2,1)$ $(1,2,1,0,3,-2,2,1)$ $(3,0,1,2,1,1,1,2)$ $(2,1,1,1,2,1,0,3)$ $(3,0,2,1,2,-1,2,1)$ $(2,1,1,1,2,-1,2,1)$ $(1,2,1,0,3,-1,1,2)$ $(2,1,0,2,1,1,1,2)$ $(1,2,-1,2,1,2,0,3)$ $(1,2,-1,2,1,1,1,2)$ $(1,2,-1,2,1,1,1,2)$ $(1,2,-1,2,1,0,2,1)$ $(1,2,0,1,2,-1,2,1)$		$(2,1,1,1,2,0,1,2)$	$(2,1,1,1,2,0,1,2)$	0,37675

Табела 5. Минимални пат вектори за транспортниот систем од Пример 4.3

	MPP_d	MP_d'	MP_d	$M2TR_d$
2	(2,0,2,0,2) (2,0,1,1,1) (1,1,0,1,1) (1,1,1,0,2) (2,0,0,2,0) (1,1,1,2,0) (0,2,2,2,0) (0,2,1,1,1) (0,2,0,0,2)	(3,0,2,0,3) (3,0,2,2,1) (3,2,0,2,1) (3,2,2,0,3) (3,0,0,2,0) (3,2,2,2,0) (0,2,2,2,0) (0,2,2,2,1) (0,2,0,0,3)	(3,0,2,0,3) (3,0,0,2,0) (0,2,2,2,0) (0,2,0,0,3)	0,97317
3	(3,0,2,1,1) (2,1,1,1,2) (2,1,2,0,3) (3,0,1,2,1) (2,1,0,2,1) (1,2,1,2,1) (1,2,0,1,2) (1,2,1,0,3) (0,3,2,2,1) (0,3,1,1,2) (0,3,0,0,3)	(3,0,2,2,3) (3,2,2,2,3) (3,2,2,0,3) (3,0,2,2,1) (3,2,0,2,1) (3,2,2,2,1) (3,2,0,2,3) (3,2,2,0,3) (0,4,2,2,1) (0,4,2,2,3) (0,4,0,0,3)	(3,2,2,0,3) (3,0,2,2,1) (3,2,0,2,1) (0,4,2,2,1) (0,4,0,0,3)	0,86104
4	(4,0,2,2,2) (3,1,1,2,2) (3,1,2,1,3) (2,2,0,2,2) (2,2,1,1,3) (2,2,2,0,4) (1,3,1,2,2) (1,3,0,1,3) (1,3,1,0,4) (0,4,2,2,2) (0,4,1,1,3) (0,4,0,0,4)	(6,0,2,2,3) (3,2,2,2,3) (3,2,2,2,3) (3,2,0,2,3) (3,2,2,2,3) (3,2,2,0,5) (3,4,2,2,3) (3,4,0,2,3) (3,4,2,0,5) (0,4,2,2,3) (0,4,2,2,3) (0,4,0,0,5)	(6,0,2,2,3) (3,2,0,2,3) (3,2,2,0,5) (0,4,2,2,3) (0,4,0,0,5)	0,80605
5	(4,1,2,2,3) (3,2,1,2,3) (3,2,2,1,4) (2,3,0,2,3) (2,3,1,1,4) (2,3,2,0,5) (1,4,1,2,3) (1,4,0,1,4) (1,4,1,0,5)	(6,2,2,2,3) (3,2,2,2,3) (3,2,2,2,5) (3,4,0,2,3) (3,4,2,2,5) (3,4,2,0,5) (3,4,2,2,3) (3,4,0,2,5) (3,4,2,0,5)	(3,2,2,2,3) (3,4,0,2,3) (3,4,2,0,5)	0,66461
6	(4,2,2,2,4) (3,3,1,2,4) (3,3,2,1,5) (2,4,0,2,4) (2,4,1,1,5)	(6,2,2,2,5) (3,4,2,2,5) (3,4,2,2,5) (3,4,0,2,5) (3,4,2,2,5)	(6,2,2,2,5) (3,4,0,2,5)	0,57024
7	(4,3,2,2,5) (3,4,1,2,5)	(6,4,2,2,5) (3,4,2,2,5)	(3,4,2,2,5)	0,46656

Табела 6. Минимални пат вектори за транспортниот систем од Пример 4.5

d	$MPP_d \rightarrow Q'_x$	$MPP_d \rightarrow Q_x$	MP'_d	MP_d	$M2TR_d$
2	(2,0,2,0,2) → {1,2,4} (2,0,1,1,1) → {2,4} (1,1,1,0,2) → {4} (2,0,0,2,0) → {2,3,4} (1,1,1,2,0) → {3,4} (1,1,0,1,1) → {4} (0,2,2,2,0) → {3,4} (0,2,1,1,1) → {4} (0,2,0,0,2) → {4}	(2,0,2,0,2) → {2,4} (2,0,1,1,1) → {2,4} (1,1,1,0,2) → {4} (2,0,0,2,0) → {4} (1,1,1,2,0) → {4} (1,1,0,1,1) → {4} (0,2,2,2,0) → {4} (0,2,1,1,1) → {4} (0,2,0,0,2) → {4}	(3,0,2,0,3) (3,0,2,2,1) (3,2,2,0,3) (3,0,0,2,0) (3,2,2,2,0) (3,2,0,2,1) (0,2,2,2,0) (0,2,2,2,1) (0,2,0,0,3)	(3,0,2,0,3) (3,0,0,2,0) (0,2,2,2,0) (0,2,0,0,3)	0,91317
3	(3,0,2,1,1) → {2,4} (2,1,2,0,3) → {4} (3,0,1,2,1) → {2,4} (2,1,1,1,2) → {4} (1,2,1,0,3) → {4} (2,1,0,2,1) → {4} (1,2,1,2,1) → {4} (1,2,0,1,2) → {4} (0,3,2,2,1) → {4} (0,3,1,1,2) → {4} (0,3,0,0,3) → {4}	(3,0,2,1,1) → {2,4} (2,1,2,0,3) → {4} (3,0,1,2,1) → {4} (2,1,1,1,2) → {4} (1,2,1,0,3) → {4} (2,1,0,2,1) → {4} (1,2,1,2,1) → {4} (1,2,0,1,2) → {4} (0,3,2,2,1) → {4} (0,3,1,1,2) → {4} (0,3,0,0,3) → {4}	(3,0,2,2,3) (3,2,2,0,3) (3,0,2,2,1) (3,2,2,2,3) (3,2,2,0,3) (3,2,0,2,1) (3,2,2,2,1) (3,2,0,2,3) (0,4,2,2,1) (0,4,2,2,3) (0,4,0,0,3)	(3,2,2,0,3) (3,0,2,2,1) (3,2,0,2,1) (0,4,2,2,1) (0,4,0,0,3)	0,86104
4	(4,0,2,2,2) → {2,4} (3,1,2,1,3) → {4} (2,2,2,0,4) → {4} (3,1,1,2,2) → {4} (2,2,1,1,3) → {4} (1,3,1,0,4) → {4} (2,2,0,2,2) → {4} (1,3,1,2,2) → {4} (1,3,0,1,3) → {4} (0,4,2,2,2) → {4} (0,4,1,1,3) → {4} (0,4,0,0,4) → {4}	(4,0,2,2,2) → {4} (3,1,2,1,3) → {4} (2,2,2,0,4) → {4} (3,1,1,2,2) → {4} (2,2,1,1,3) → {4} (1,3,1,0,4) → {4} (2,2,0,2,2) → {4} (1,3,1,2,2) → {4} (1,3,0,1,3) → {4} (0,4,2,2,2) → ∅ (0,4,1,1,3) → ∅ (0,4,0,0,4) → ∅	(6,0,2,2,3) (3,2,2,2,3) (3,2,2,0,5) (3,2,2,2,3) (3,2,2,2,3) (3,4,2,0,5) (3,2,0,2,3) (3,4,2,2,3) (3,4,0,2,3) (0,4,2,2,3) (0,4,2,2,3) (0,4,0,0,5)	(6,0,2,2,3) (3,2,2,0,5) (3,2,0,2,3) (0,4,2,2,3) (0,4,0,0,5)	0,80605
5	(4,1,2,2,3) → {4} (3,2,2,1,4) → {4} (2,3,2,0,5) → {4} (3,2,1,2,3) → {4} (2,3,1,1,4) → {4} (1,4,1,0,5) → {4} (2,3,0,2,3) → {4} (1,4,1,2,3) → {4} (1,4,0,1,4) → {4}	(4,1,2,2,3) → {4} (3,2,2,1,4) → {4} (2,3,2,0,5) → ∅ (3,2,1,2,3) → {4} (2,3,1,1,4) → {4} (1,4,1,0,5) → ∅ (2,3,0,2,3) → {4} (1,4,1,2,3) → ∅ (1,4,0,1,4) → ∅	(6,2,2,2,3) (3,2,2,2,5) (3,4,2,0,5) (3,2,2,2,3) (3,4,2,2,5) (3,4,2,0,5) (3,4,0,2,3) (3,4,2,2,3) (3,4,0,2,5)	(3,4,2,0,5) (3,2,2,2,3) (3,4,0,2,3)	0,66461
6	(4,2,2,2,4) → {4} (3,3,2,1,5) → {4} (3,3,1,2,4) → {4} (2,4,1,1,5) → {4} (2,4,0,2,4) → {4}	(4,2,2,2,4) → {4} (3,3,2,1,5) → ∅ (3,3,1,2,4) → {4} (2,4,1,1,5) → ∅ (2,4,0,2,4) → ∅	(6,2,2,2,5) (3,4,2,2,5) (3,4,2,2,5) (3,4,2,2,5) (3,4,0,2,5)	(6,2,2,2,5) (3,4,0,2,5)	0,57024
7	(4,3,2,2,5) → {4} (3,4,1,2,5) → {4}	(4,3,2,2,5) → ∅ (3,4,1,2,5) → ∅	(6,4,2,2,5) (3,4,2,2,5)	(3,4,2,2,5)	0,46656

Табела 7. Минимални пат вектори за транспортниот систем од Пример 4.6

d	Минимални пат множества за ниво d					$M2TR_d$
	a_1	a_2	a_3	a_4	a_5	
3	3	2	2	0	3	0,86104
	0	4	2	2	1	
	0	4	0	0	3	
	3	0	2	2	1	
	3	2	0	2	1	
4	6	0	2	2	3	0,80605
	3	2	2	0	5	
	0	4	2	2	3	
	0	4	0	0	5	
	3	2	0	2	3	
5	3	4	2	0	5	0,66461
	3	2	2	2	3	
	3	4	0	2	3	
6	6	2	2	2	5	0,57024
	3	4	0	2	5	
7	3	4	2	2	5	0,46656

Табела 8. Минимални пат вектори за транспортниот систем од Пример 4.7

d	Минимални пат множества за ниво d	$M2TR_d$
	a_1 a_2 a_3 a_4 a_5	
1	1,5 0 1 0 2 1,5 0 0 1 0 0 2 1 1 0 0 2 0 0 2	0,99584
1,5	1,5 0 2 0 2 1,5 0 1 1 2 1,5 0 0 2 0 0 2 2 2 0 0 2 0 0 2	0,9895
2	3 0 2 0 2 3 0 1 1 2 3 0 0 2 0 1,5 2 1 2 0 0 2 2 2 0 0 2 0 0 2	0,98846
3	3 0 2 1 2 3 0 1 2 2 1,5 2 1 0 4 0 4 1 1 2 1,5 2 0 1 2 0 4 0 0 4	0,94256
3,5	1,5 2 2 0 4 1,5 2 1 1 4 0 4 2 2 2 1,5 2 0 2 2 0 4 0 0 4	0,92724
4	3 2 1 1 4 3 2 0 2 2 1,5 4 1 2 2 3 2 2 0 4 0 4 2 2 2 0 4 0 0 4	0,91986
5	3 2 1 2 4 3 2 2 1 4 1,5 4 0 1 4	0,79928
5,5	1,5 4 0 2 4	0,64152
6	3 4 0 2 4	0,5832

Табела 9. Минимални пат вектори за транспортниот систем од Пример 4.8