

МЕТОДЕ ОДЛУЧИВАЊА

АТ-6 Технологичност производа, оптимизација технолошког процеса

ТЕХНОЛОГИЧНОСТ ПРОИЗВОДА

Дефиниција

„Технологичност производа се у најширем смислу те речи може дефинисати као мера погодности његове израде у одређеном производном систему неким методом обраде (на пример: методом обраде резањем, пластичним деформисањем, неконвенционалним методама, итд.), уз остварење минимума производних трошкова”.

Напомене:

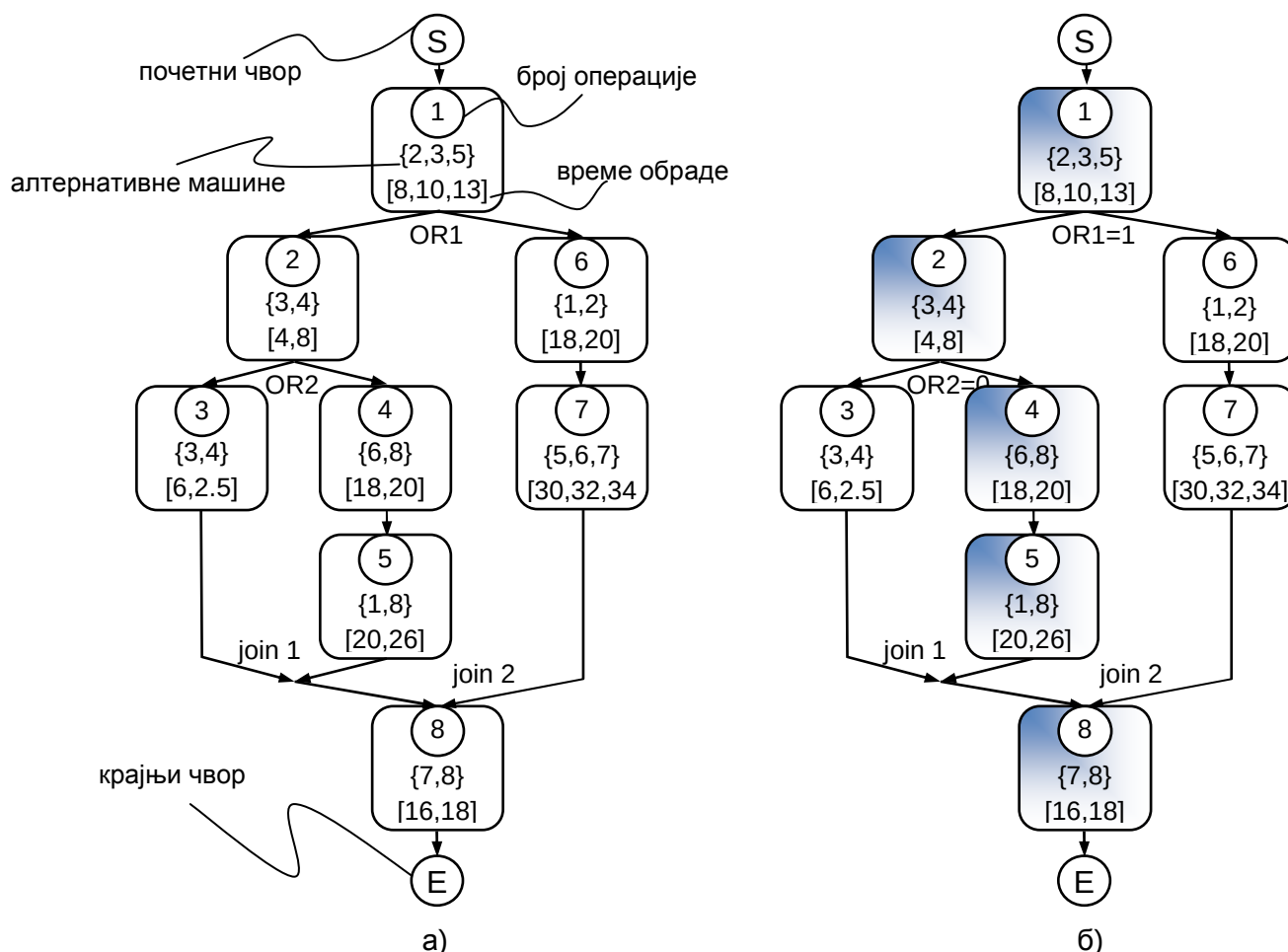
- Технологичним решењем производа се може сматрати оно решење које обезбеђује исправно функционисање производа, са захтеваним степеном поузданости, у предвиђеном веку експлоатације, а уз најмање трошкове, најмањи утрошак материјала, **најмањи утрошак енергије** и уз најмање загађивање околине;
- При анализи технологичности производа разматрају се следећи показатељи технологичности:
 - геометријски облик;
 - мере и толеранције;
 - толеранције облика;
 - површинска храпавост;
 - врста материјала и његова тврдоћа;
 - величина серије;
 - **погодност израде (време израде);**
 - погодност уградње (време монтаже).
- Повећање технологичности производа се може остварити неким од следећих начина:
 - упрошћавањем облика производа;
 - применом групне технологије (Бабић) у пројектовању технолошких процеса;
 - повећањем степена обрадљивости материјала;
 - применом стандардних и типизираних технолошких облика и делова;
 - **оптимизацијом технолошког процеса** (одабир и учење оптималних транспортних токова материјала остварује се применом теорије графова и техника вештачке интелигенције).

Дефиниција пројектовања технолошког процеса

Пројектовање технолошког процеса (Бабић) се може дефинисати као систематско одређивање детаљних метода којима се делови или склопови (у случају монтаже) могу произвести економично и конкурентно, од иницијалне фазе (сиров материјал), до завршне фазе (жељени облик). Улази за пројектовање технолошког процеса су пројектни подаци, подаци о сировом материјалу, подаци о обрадном систему (подаци о машини, алату, стезним приборима), подаци о захтевима квалитета и подаци о типу производње. Излаз из пројектовања технолошког процеса је **технолошки поступак**, којим се дефинише редослед одвијања свих активности (операција) потребних да се од полазног материјала (сировине или полуфабриката) обликује готов део (производ).

У литератури (SHAO, X.; LI, X.; GAO, L., ZHANG, C., *Integration of process planning and scheduling-A modified genetic algorithm-based approach*), представљање флексибилности технолошког процеса израде дела може се извршити коришћењем неколико метода: представљање путем Петријевих мрежа, представљање помоћу графова, представљање мрежама. Метод представљања технолошког процеса путем мрежа може се описати на следећи начин. Сваку мрежу технолошког процеса чине три врсте чворова: почетни чвор, један или више средњих чворова и крајњи чвор. Почетни и крајњи чвор указују на почетак и крај технолошког процеса, док средњи чворови представљају редослед одвијања операција технолошког процеса. Сваки од средњих чворова, поред редног броја операције, садржи и сет алтернативних машина на којима се операција изводи, као и сет коресподентних времена трајања операције на алтернативној машини. Сви чворови мреже су повезани стрелицама, које указују на редослед извршавања операција. Имајући у виду расположиве ресурсе у производном погону (машине у погону на којима је могуће израдити део), за део је могуће генерисати и алтернативне технолошке процесе. Сваки алтернативни технолошки

процес почиње такозваним *OR* конектором, а завршава се *join* конектором, па се на основу вредности *OR* конектора (нула или један) **доноси одлука** о томе који ће од алтернативних технолошких процеса израде дела бити одабран. Уколико је, на пример, вредност *OR* конектора нула, бира се десна алтернативна грана технолошког процеса, док се у супротном, за вредност *OR* конектора један, бира лева алтернативна грана технолошког процеса. У случају да се алтернативни технолошки процес, или пак цела мрежа технолошког процеса, не састоји од *OR* конектора, тада сви средњи чворови морају бити посећени приликом претраживања тј. све операције представљене чвором морају бити извршене на алтернативним машинама.



а) Пример мреже алтернативних технолошких процеса израде изабраног дела са почетним, средњим и крајњим чворовима, као и *OR* и *join* конекторима; б) Пример избора алтернативног технолошког процеса за вредности конектора $OR1=1$ и $OR2=0$

Математички модел за оптимизацију технолошког процеса

Оптимизација технолошког процеса има велики утицај на перформансе савремених технолошких система и може се извршити применом једног или више различитих оптимизационих критеријума (минималан утрошак енергије, минимална цена, минимално време производње, уравнотежен степен искоришћења машина алатки, ...). Производно време, које представља збир времена обраде дела на машини и времена транспорта дела од машине до машине, је један од критеријума који се користи при оптимизацији технолошког процеса. Математички модел производног времена је приказан на следећи начин:

$$TP(i) = \sum_{j=1}^{P_i} TW(i, j, k, l) + \sum_{j=1}^{P_i-1} TT(i, l, (j, k_1), (j+1, k_2)), i \in [1, n], j \in [1, P_i],$$

Где је:

$TP(i)$ – производно време за i -ти део;

$TW(i, j, l, k)$ – време трајања операције o_{ijl} на k -тој алтернативној машини;

$TT(i, l, (j, k_1), (j+1, k_2))$ – време транспорта дела i између алтернативних машина k_1 и k_2 ;

n – укупан број делова;

o_{ijl} – j -та операција l -ог технолошког процеса i -тог дела;

P_{il} – број операција l -ог технолошког процеса i -тог дела;

k – алтернативне машине за операцију o_{ijl} ;

Претпоставке које се користе приликом примене овог математичког модела су следеће:

- све машине су доступне у почетном тренутку ($t_0=0$);
- на једној машини се може вршити само једна операција једног дела у једном временском периоду;
- операције једног дела се не могу обављати симултано (истовремено) на једној или више машина (док се не заврши операција o_{ijl} , не може почети операција o_{ij+1l} i -тог дела).

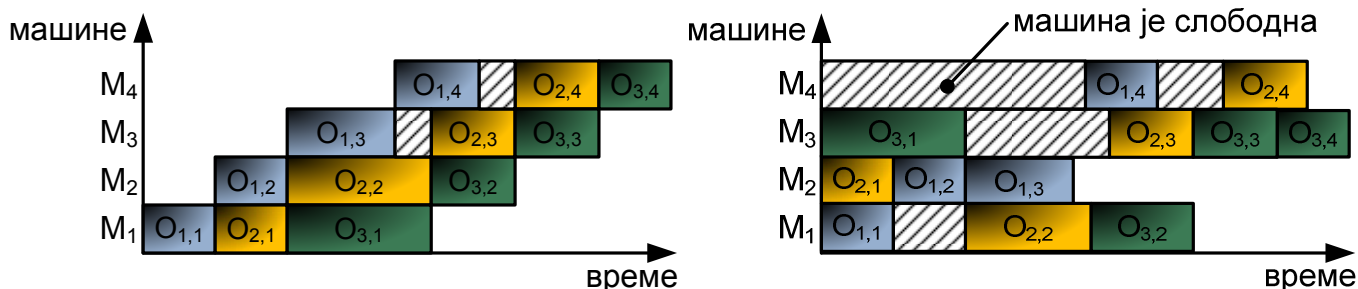
Функција циља (функција одлучивања) $f(i)$, која се користи при одабиру оптималног технолошког процеса за део i има следећу формулацију:

$$\max f(i) = \frac{1}{TP(i)},$$

и представља технолошки процес са минималним производним временом $TP(i)$.

Анализа транспортних токова материјала

Један од фактора који директно утиче на ефикасност извођења технолошких операција у погонима технолошких система је и уређење производног простора. Према врсти технолошких система, два су вида уређења производног простора: **flow-shop layout** (репроматеријал, обраци и делови се кроз технолошки систем крећу секвенцијално истим путем; на пример линија за монтажу у аутомобилској индустрији, трансфер линија, линија за израду лименки, итд.) и **job-shop layout** (репроматеријал, обраци и делови се кроз технолошки систем крећу и посећују различите машине сходно одабраном алтернативном технолошком поступку; на пример технолошки систем за израду зупчаника).



Flow-shop и job-shop тип уређења производног простора (o_{ij} је j -та операција i -тог дела на одговарајућој машини)

За анализу производног тока унутар предузећа неопходни су следећи подаци о машинама (напомена: у даљем тексту се увек мисли на машине алатке) и деловима:

- број и опис машина које се користе унутар технолошког система;
- број различитих (репрезентативних) делова који се обрађују;
- величина серије за сваки део;
- технолошка путања за сваки део.

Након усвајања података о машинама и деловима, неопходно је успоставити квантитативну зависност између њих. У општем случају, ова зависност се може представити матрицом међузависности M_{DM} , која се на основу матрице машина M и матрице делова D дефинише на следећи начин:

$$M = [m_1 \ m_2 \ \dots \ m_{NM}], \ D = [d_1 \ d_2 \ \dots \ d_{ND}]',$$

$$M_{DM} = D \cdot M = \begin{bmatrix} d_1 \\ \vdots \\ d_{ND} \end{bmatrix} \begin{bmatrix} m_1 & \dots & m_{NM} \end{bmatrix}, \quad M_{DM} = \begin{bmatrix} p_{11} & \dots & p_{1j} & \dots & p_{1NM} \\ \vdots & \ddots & \dots & \dots & \vdots \\ p_{i1} & \dots & p_{ij} & \dots & p_{iNM} \\ \vdots & \dots & \dots & \ddots & \vdots \\ p_{ND1} & \dots & p_{NDj} & \dots & p_{NDNM} \end{bmatrix},$$

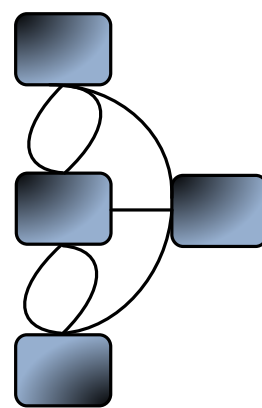
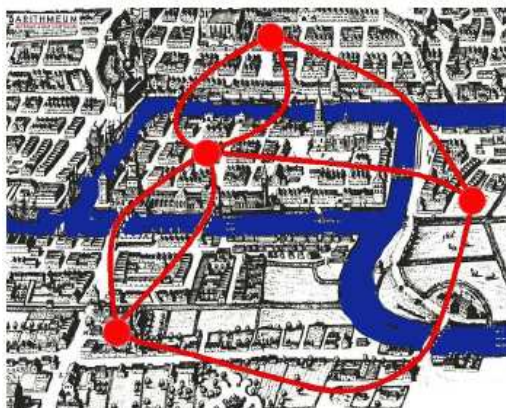
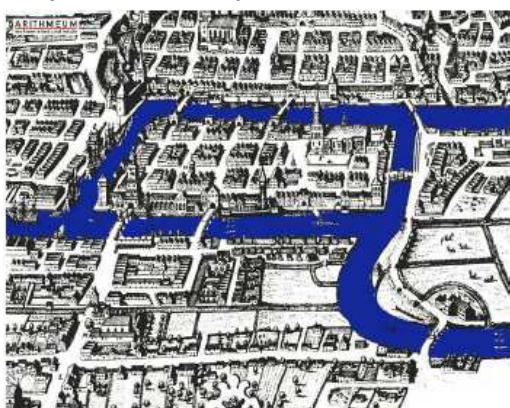
где је: M - вектор машина; D - вектор делова; M_{DM} - квантитативна матрица међузависности; ND - број различитих делова које треба обрадити; NM - број различитих машина у погону; p_{ij} - елемент матрице M_{DM} који има вредност 1 уколико део посећује машину и вредност 0 уколико део не посећује машину.

Ако се у успостављену матрицу међузависности делови-машине M_{DM} уместо параметра p_{ij} замени време трајања операције сваког дела на одговарајућим машинама изражено у секундама, добија се матрица времена T_{DM} . У том случају, t_{ij} је елемент матрице T_{DM} који има вредност различиту од 0 уколико део посећује машину и вредност 0 уколико део не посећује машину.

$$T_{DM} = \begin{bmatrix} t_{11} & \dots & t_{1j} & \dots & t_{1NM} \\ \vdots & \ddots & \dots & \dots & \vdots \\ t_{i1} & \dots & t_{ij} & \dots & t_{iNM} \\ \vdots & \dots & \dots & \ddots & \vdots \\ t_{ND1} & \dots & t_{NDj} & \dots & t_{NDNM} \end{bmatrix}$$

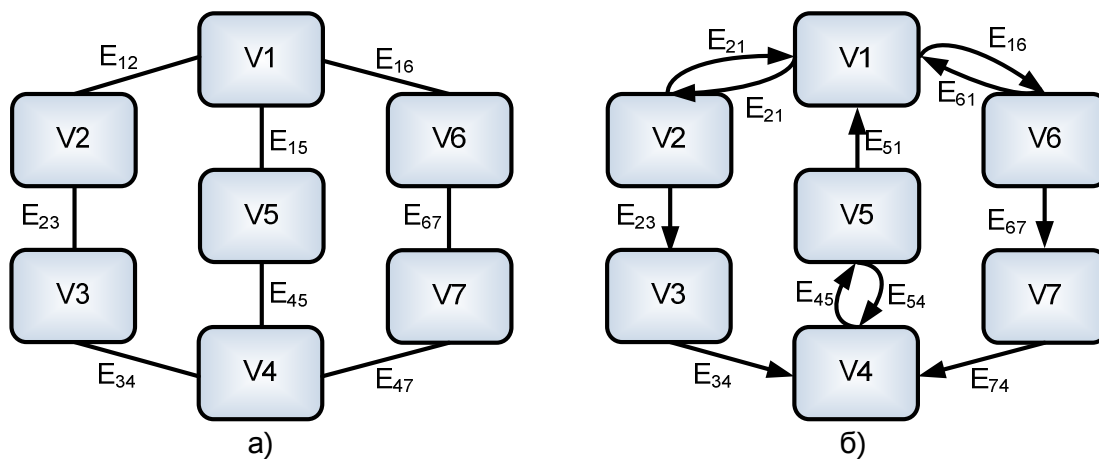
Методологија за оптимизацију технолошког процеса: Примена теорије графова у генерисању оптималног технолошког процеса

Темеље теорије графова поставио је швајцарски математичар и физичар Леонард Ојлер, 1736. године, решавањем легендарног историјског проблема „Седам мостова Кенигсберга“¹. Од тада се различити проблеми (логистички проблеми, проблеми представљања саобраћајница и електричних мрежа, планирање реда летења и планирање путања кретања) моделирају и решавају применом теорије графова. Формално, граф се може дефинисати као уређени пар (V, E) , где се елементи скупа V се зову чворови (енг. *vertex*, *node*), а елементи скупа E гране (енг. *edge*) графа G . За дати граф G , скуп чворова се означава са $V(G)$, а скуп грана са $E(G)$. Уколико је грана E_{ij} , која спаја чворове V_i и V_j графа G , идентична као и грана E_{ji} , која спаја чворове V_j и V_i истог графа, онда је граф G неоријентисан или неусмерен. Ако се, пак, сматра да су те две гране различите, онда је граф G оријентисан или усмерен.



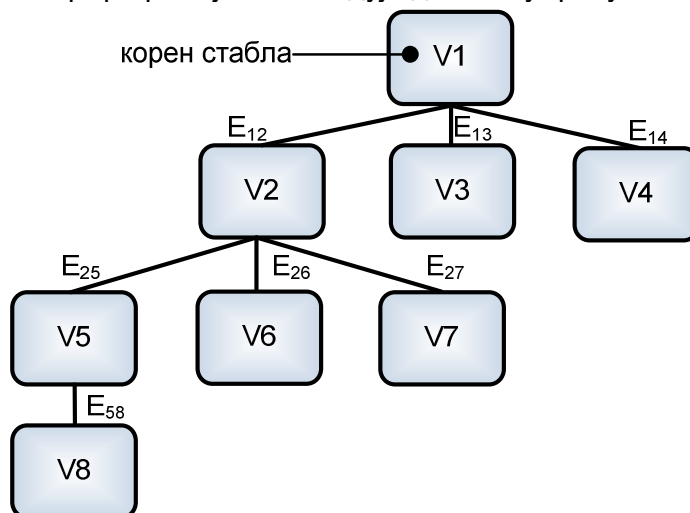
Проблем седам мостова Кенигсберга и Ојлерово решење применом теорије графова

¹ Град Кенигсберг је данас познат под називом Калининград и у прошлости је имао седам мостова који су повезивали два острва унутар града са остатком града. Грађани Кенигсберга често су покушавали да пређу само по једном преко свих мостова у једној шетњи. Питање „Да ли је могуће прећи преко свих седам мостова тако да се преко сваког моста пређе тачно једанпут“, постављено је Ојлеру 1736. године. Он је, придруживши свакој обали и острву чворове графа, а гране између њих представивши мостовима, показао да се овакав граф не може нацртати у једном потезу не подижући оловку са папира тј. да је такав обилазак мостова немогућ.



а) Пример неоријентисаног и симетричног неусмереног графа; б) Пример оријентисаног и асиметричног усмереног графа

Поред поделе графова на оријентисане и неоријентисане, позната је и подела на следеће типове графова: пут, контура, точка и звезда. Граф типа контуре има све чворове степена² 2 и при томе је почетни чвор једнак крајњем. Стабло (енг. *tree*) представља повезан граф без контуре. Стабло је граф у коме су свака два чвора повезана јединственим елементарним путем, док су коренска стабла посебна категорија стабала, код којих је један чвор посебно издвојен и назива се корен стабла. Стабло или дрво је тип директних, ацикличких графова са специјалним чвором званом корен стабла (енг. *root*). Корен је и једини чвор графа који не поседује долазећу грану.



Пример усмереног графа типа стабла (дрвета)

Графичка репрезентација графова подразумева и могућност њиховог приказивања квадратном матрицом, чији је ред једнак броју чворова графа. Елемент a_{ij} једнак је броју грана које полазе из чвора V_i , а завршавају се у чвору V_j . Ова матрица се назива матрица суседства графа. Поред матрице суседства, графови се могу представити и квадратном матрицом растојања $R(G)$, чији је ред једнак броју чворова графа, а елемент r_{ij} једнак је растојању између чворова V_i и V_j . Матрица R је формата $(m \times m)$ и дата је квадратном схемом:

$$R = \begin{bmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{i1} & \cdots & r_{ij} & \cdots & r_{im} \\ \vdots & \cdots & \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mj} & \cdots & r_{mm} \end{bmatrix},$$

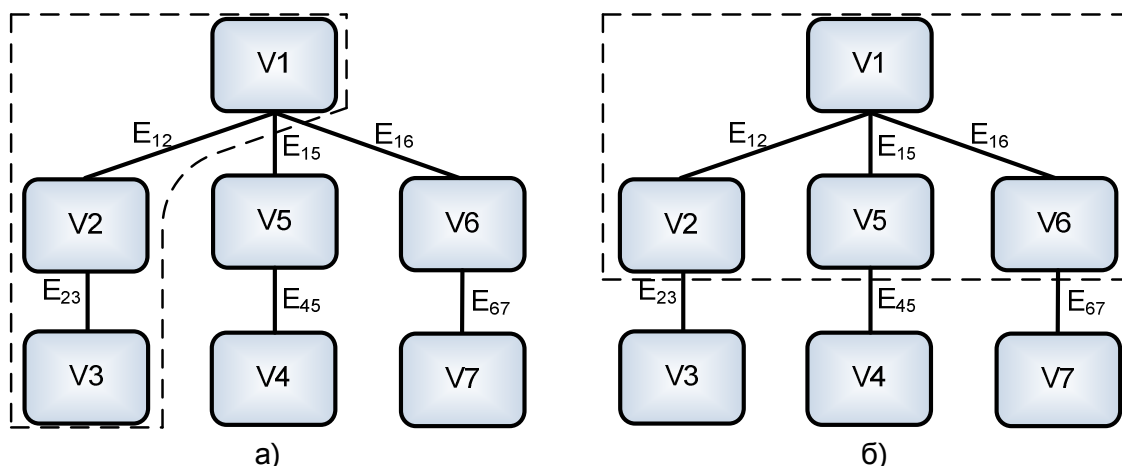
где је скалар r_{ij} (i,j) -ти елемент матрице R $(i,j = 1, \dots, m)$. У случају да грана повезује један исти чвор, тежински однос те гране је тада једнак нули. Ова својства призилазе из својства симетричности матрице R , и представљена су релацијама $r_{ij} = r_{ji}$, $r_{ii} = 0$ и $r_{ij} = 0$.

² Степен чвора - број суседних чворова неком чвору зове се степен чвора

Алгоритми претраживања графова

У литератури су познати различити алгоритми претраживања, који се користе за одређивање најкраћег пута између два чвора графа. Једна од подела алгоритама претраживања је подела на:

- *depth-first search* (алгоритам који започиње претраживање дуж једне гране графа, а након завршетка прелази на следећу грану);
- *breadth-first search* ("похлепни" алгоритам који започиње претраживање чворова на истом нивоу графа).



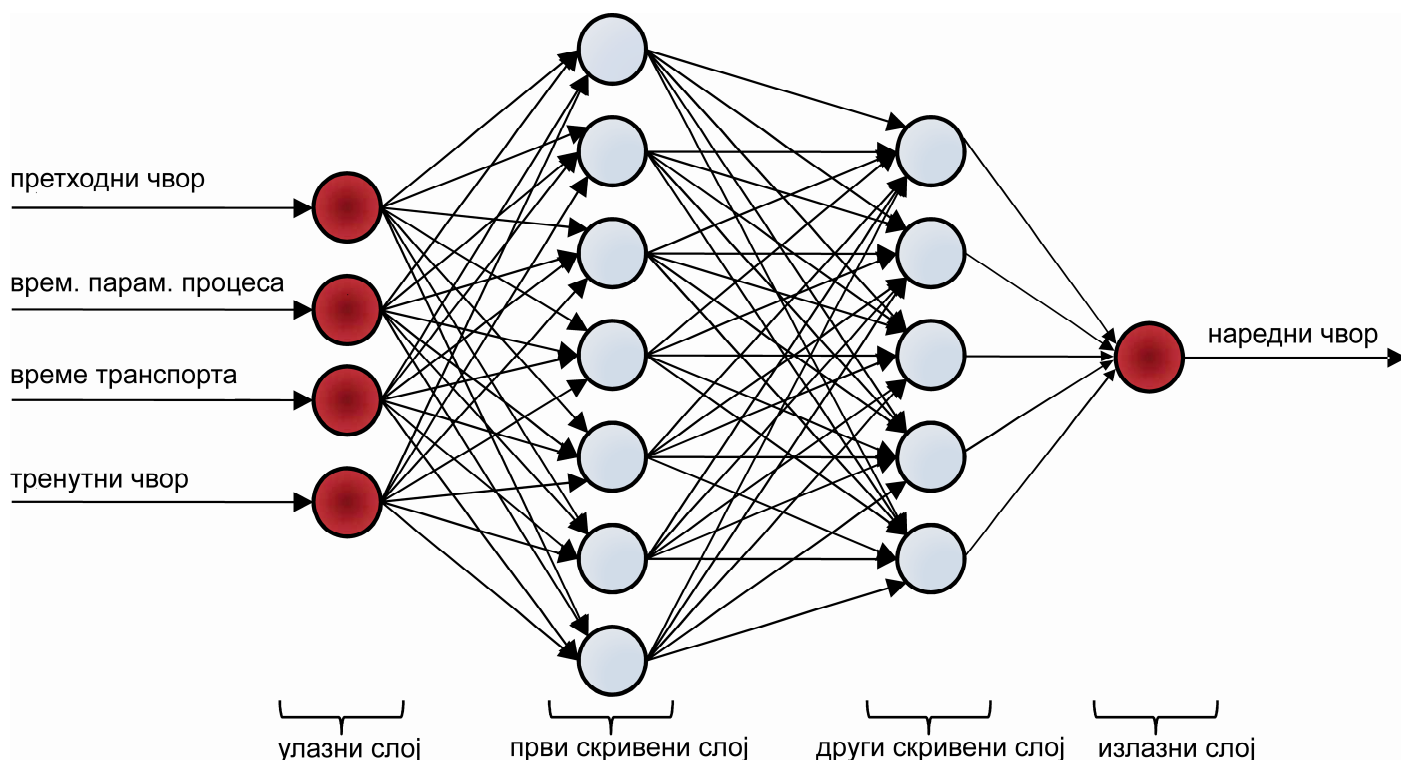
а) *Depth-first search* алгоритам претраживања (редослед претраживања чворова је: $V1 \rightarrow V2 \rightarrow V3$);

б) *breadth-first search* алгоритам претраживања (редослед претраживања чворова је: $V1 \rightarrow V2$; $V1 \rightarrow V5$; $V1 \rightarrow V6$)

Вештачке неуронске мреже у учењу оптималних технолошких процеса

Корак#1: Дефинисање архитектуре вештачке неуронске мреже. Примена система вештачких неуронских мрежа (ВНМ) за решавање проблема у различитим областима производног инжењерства је веома широка и датира још од деведесетих година 20. века. Према (З.Миљковић, Д.Александрић-збирка), издвајају се три основне категорије њихове употребе: класификација, предикција и функционална апроксимација. За задатак учења оптималних транспортних токова материјала користи се друга категорија употребе ВНМ - предикција. На основу претходних вредности стања система (у конкретном случају: чвор у ком се робот претходно налазио, укупно време обраде-временски параметри процеса и укупно време кретања робота између машина-време транспорта) и тренутних вредности стања система (чвор у ком се робот тренутно налази), врши се предикција наредног чвора путање у који се транспортује део. За овако дефинисане вредности стања система, бира се ВНМ архитектуре $4 [7-5]_2 1$ (четири неурона у улазном слоју и један неурон у излазном слоју-слика). Оптималана архитектура ВНМ за учење технолошких процеса добија се анализом различитих архитектура мрежа (варирањем броја слојева и броја неурона у слојевима) помоћу софтверских пакета *BPnet* и *Matlab*.

Корак#2: Прикупљање парова за обучавање вештачке неуронске мреже. Као што је већ напоменуто, производно време $TP(i)$ представља збир времена обраде дела на машини $TW(i,j,l,k)$ и времена транспорта дела од машине до машине $TT(i,l,(j,k_1),(j+1,k_2))$. С обзиром на стохастичку природу времена обраде и транспорта, следи закључак да прописано (номинално) време трајања одређене операције, као и време транспорта делова (мобилним роботом) од машине до машине није једнако за сваки део из групе истих (репрезентативних) делова. На пример, време трајања технолошке операције o_{ij} за део i на машини k налази се у интервалу $TW(i,j,l,k) \pm tp$, где је tp толеранција времена трајања операције, док се на аналоган начин време транспорта дела i од машине k_1 до машине k_2 налази у интервалу $TT(i,l,(j,k_1),(j+1,k_2)) \pm tt$. Алгоритам за одређивање различитих путања кретања базира се на варирању толеранција tp и tt у границама $\pm 25\%$ и добијању различитих оптималних путања у зависности од временских параметара процеса (времена обраде дела и времена транспорта дела).



Архитектура вештачке неуронске мреже са дефинисаним улазним и излазним скупом обучавајућих парова

Алгоритам за одређивање различитих путања кретања у зависности од времена обраде и времена транспорта дела

Старт (почетак)

Улазне величине:

- вектор делова D ; вектор машина M ; матрица зависности делови машине M_{DM} ;
- матрица времена T_{DM} ;

Изразне величине:

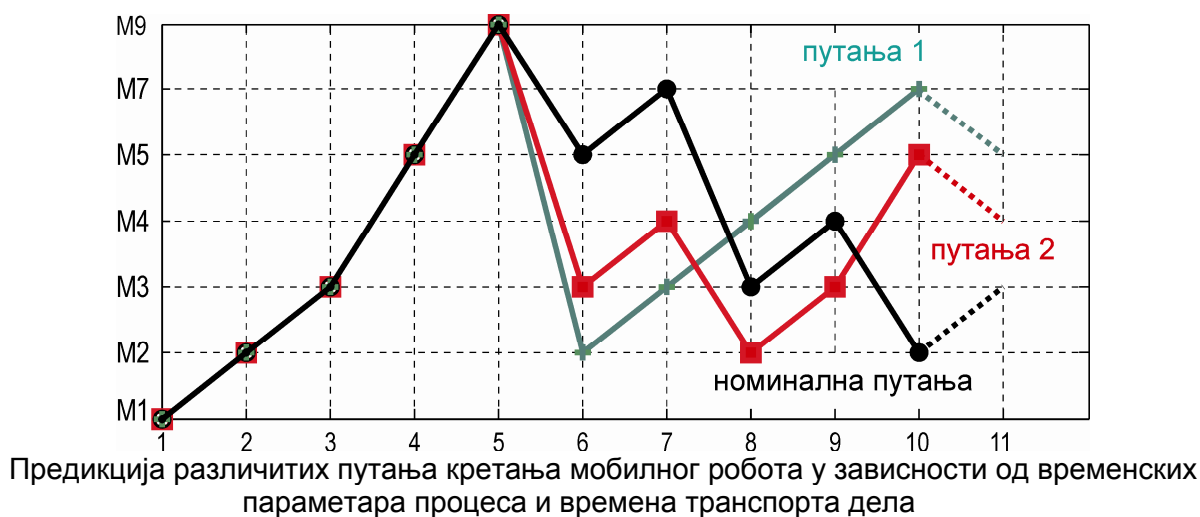
- путање кретања у зависности од временских параметара;

```

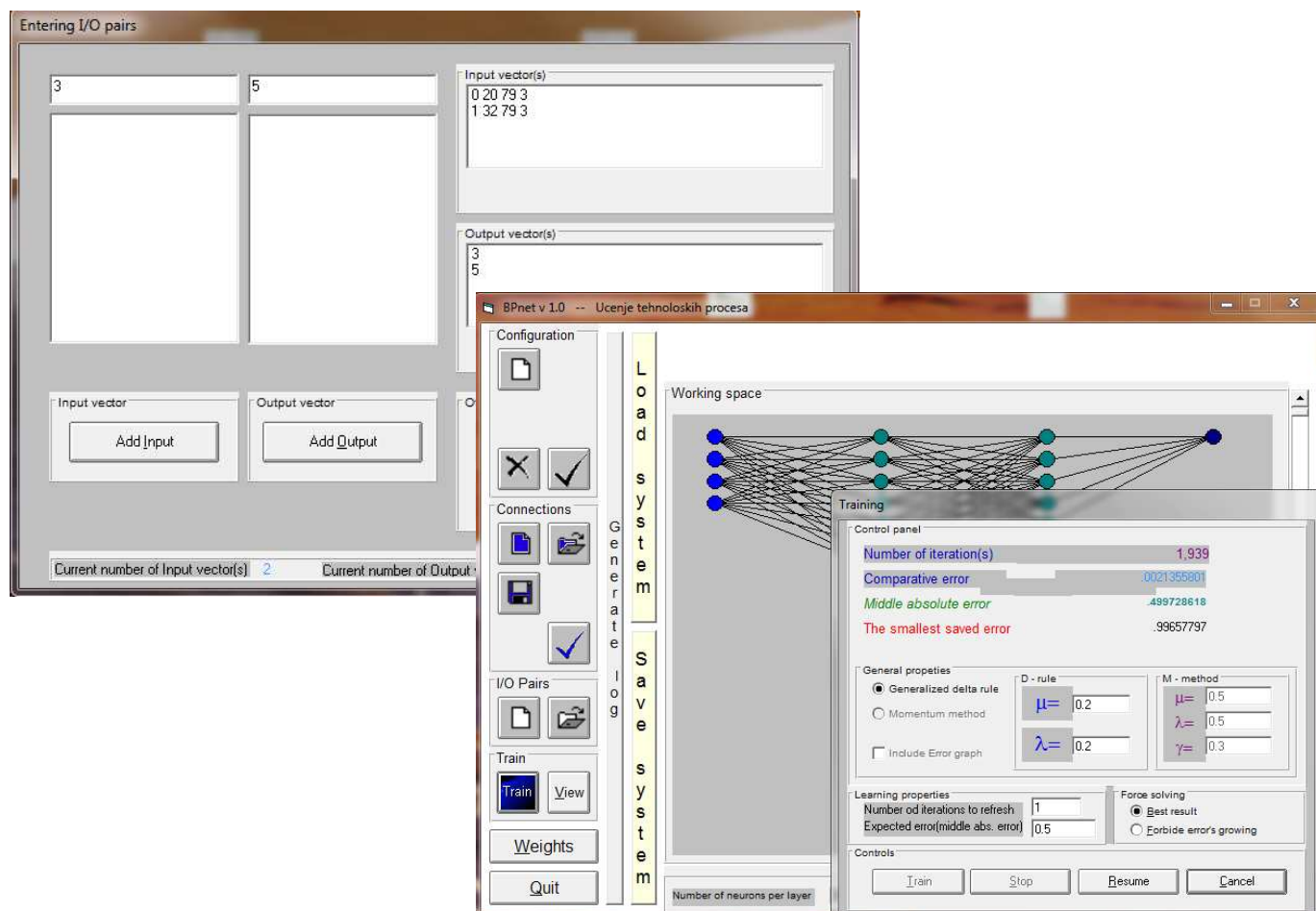
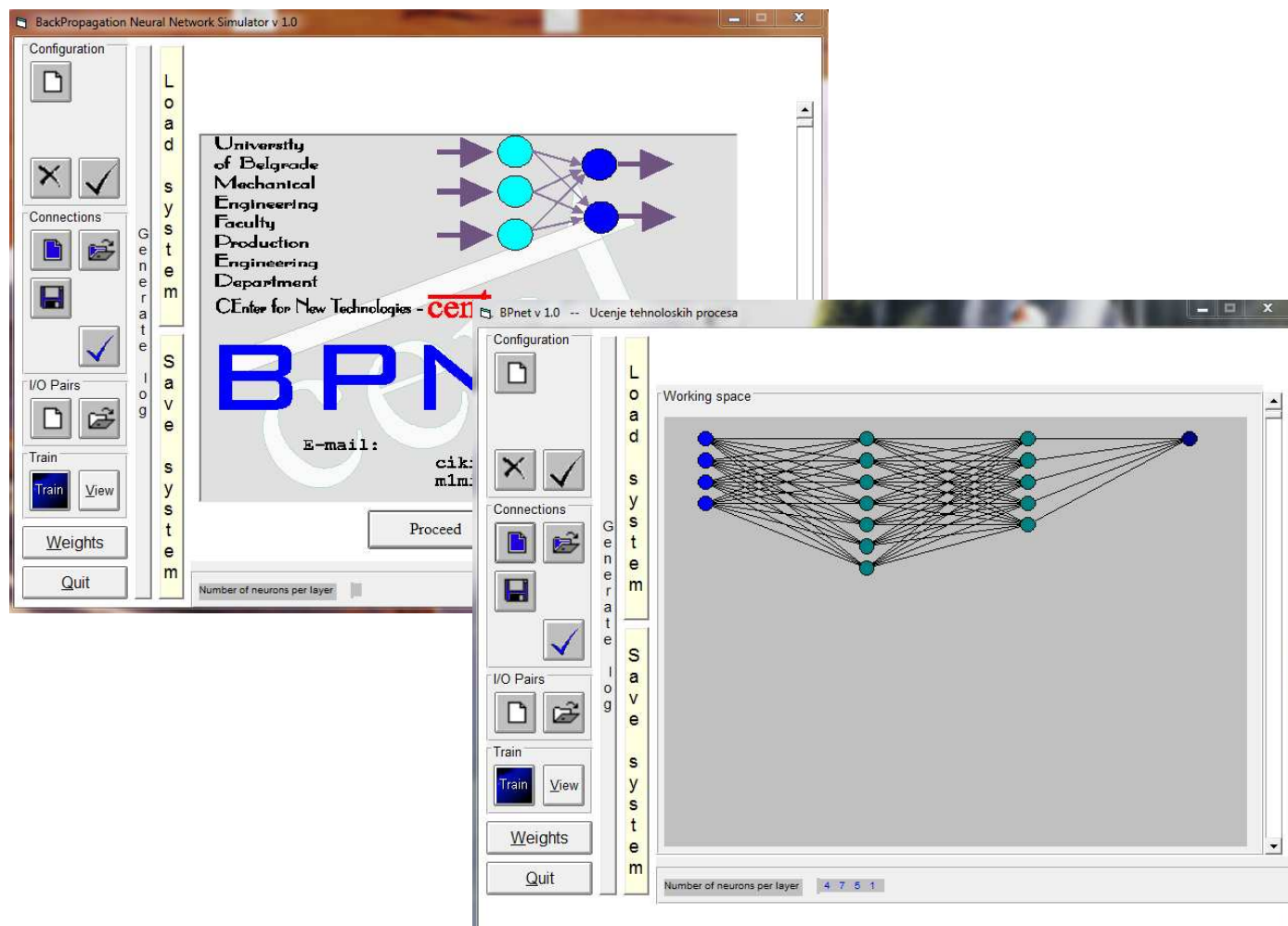
1:  for j = 1 : n, где је n број различитих путања
2:    време обраде =  $TW(i,j,l,k) \pm tp(j)$ ; време транспорта =  $TT(i,l,(j,k_1),(j+1,k_2)) \pm tt(j)$ ;
3:    на основу време обраде за репрезентативни део  $i$  формирати мрежу технолошких процеса;
4:    коришћењем теорије графова генерисати оптималан технолошки процес ( $\max f(i)=1/TP(i)$ );
5:    путања
6:  end for

```

Крај



Корак#3: Обучавање ВМ у софтверу BPnet.



Корак#3: Код за обучавање ВММ у Matlab Neural Network Toolbox-у.

```
clc; clear all; close all;

%=====
% nominalni vremenski parametri
%=====
nnp1 = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17]';
mt1 = [20 32 48 97 97 144 144 162 162 175 175 ...
       200 200 220 220 242 242 275]';
tt1 = [79.0000 79.0000 99.0000 121.0711 143.1421 163.1421 ...
       163.1421 211.4264 233.4975 255.5685 275.5685 302.6396 309.7107...
       316.7817 343.8528 370.0660 399.2082 406.2792]'; %amt1 = [];
node1 = [3 3 5 9 5 3 3 8 9 5 3 6 5 6 3 9 6 5]';
%=====
% vremenski parametri # 1
%=====
nnp2 = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17]';
mt2 = [25 37 353 402 402 420 420 433 433 458 458 ...
       505 505 525 525 558 558 580]';
tt2 = [79.0000 79.0000 99.0000 121.0711 147.2843 195.5685 ...
       217.6396 239.7107 259.7107 286.7817 293.8528 313.8528 ...
       333.8528 340.9239 340.9239 347.9949 367.9949 394.2082]';
node2 = [3 3 5 9 3 8 9 5 3 6 5 3 5 6 6 5 3 9]';

%=====
% vremenski parametri # 2
%=====
nnp3 = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17]';
mt3 = [40.0000 59.2000 84.8000 163.2000 163.2000 192.0000 ...
       192.0000 267.2000 267.2000 307.2000 307.2000 328.0000 328.0000...
       380.8000 380.8000 416.0000 416.0000 448.0000]';
tt3 = [79.0000 79.0000 99.0000 121.0711 147.2843 195.5685 ...
       235.5685 255.5685 255.5685 282.6396 311.7817 333.8528 340.9239 ...
       347.9949 367.9949 394.2082 416.2792 423.3503]';
node3 = [3 3 5 9 3 8 5 3 3 6 9 5 6 5 3 9 5 6]';

%=====
% vremenski parametri # 3
%=====
nnp4 = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17]';
mt4 = [25 37 153 202 202 220 220 233 233 280 ...
       280 305 305 325 325 347 347 380]';
tt4 = [79.0000 79.0000 99.0000 121.0711 147.2843 195.5685 ...
       217.6396 239.7107 239.7107 259.7107 259.7107 286.7817 293.8528...
       300.9239 327.9949 354.2082 383.3503 390.4214]';
node4 = [3 3 5 9 3 8 9 5 5 3 3 6 5 6 3 9 6 5]';

%=====
% vremenski parametri # 4
%=====
nnp5 = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17]';
mt5 = [25 37 53 102 102 149 149 167 167 480 480 ...
       505 505 527 527 560 560 580]';
tt5 = [79.0000 79.0000 99.0000 121.0711 143.1421 ...
       163.1421 163.1421 211.4264 233.4975 255.5685 275.5685 ...
       302.6396 329.7107 355.9239 385.0660 392.1371 392.1371 399.2082]';
node5 = [3 3 5 9 5 3 3 8 9 5 3 6 3 9 6 5 5 6]';

%=====
%ULAZI
%=====
previous_nodes = [nnp1; nnp2; nnp3; nnp4; nnp5];
machining_time = [mt1; mt2; mt3; mt4; mt5];
transportation_time = [tt1; tt2; tt3; tt4; tt5];
node = [node1; node2; node3; node4; node5];

all_inputs = [node previous_nodes machining_time transportation_time];

%=====
%OUTPUTS
%=====
node1 = [3 3 5 9 5 3 3 8 9 5 3 6 5 6 3 9 6 5]';
nn1 = [3 5 9 5 3 3 8 9 5 3 6 5 6 3 9 6 5 3]';

node2 = [3 3 5 9 3 8 9 5 3 6 5 3 5 6 6 5 3 9]';
nn2 = [3 5 9 3 8 9 5 3 6 5 3 5 6 6 5 3 9 3]';

node3 = [3 3 5 9 3 8 5 3 3 6 9 5 6 5 3 9 5 6]';
nn3 = [3 5 9 3 8 5 3 3 6 9 5 6 5 3 9 5 6 3]';

node4 = [3 3 5 9 3 8 9 5 5 3 3 6 5 6 3 9 6 5]';
nn4 = [3 5 9 3 8 9 5 5 3 3 6 5 6 3 9 6 5 3]';
```

```

node5 = [3 3 5 9 5 3 3 8 9 5 3 6 3 9 6 5 5 6]';
nn5 = [3 5 9 5 3 3 8 9 5 3 6 3 9 6 5 5 6 3]';

next_node = [nn1; nn2; nn3; nn4; nn5];
all_outputs = [next_node zeros(1,size(next_node,1))' ...
    zeros(1,size(next_node,1))' zeros(1,size(next_node,1))'];

vnm_tehnosloski_proces = newff(all_inputs,all_outputs,[8,4],{'tansig','tansig'},'trainlm');

%Parametri neuronske mreze
vnm_tehnosloski_proces.trainParam.show = 50;
vnm_tehnosloski_proces.trainParam.lr = 0.8;
vnm_tehnosloski_proces.trainParam.mc = 0.9;
vnm_tehnosloski_proces.trainParam.mu = 0.001;
vnm_tehnosloski_proces.trainParam.epochs = 1000;
vnm_tehnosloski_proces.trainParam.goal = 1e-7;
vnm_tehnosloski_proces.trainParam.max_fail = 10;

vnm_tehnosloski_proces = train(vnm_tehnosloski_proces,all_inputs,all_outputs)
put_sim = sim(vnm_tehnosloski_proces,all_inputs)
%save vnm_tehnosloski_proces
%load vnm_tehnosloski_proces

figure(1)
subplot(5,1,1)
plot(1:18,nn1,'-r*'); hold on
subplot(5,1,2)
plot(1:18,nn2,'-g*'); hold on
subplot(5,1,3)
plot(1:18,nn3,'-b*'); hold on
subplot(5,1,4)
plot(1:18,nn4,'-c*'); hold on
subplot(5,1,5)
plot(1:18,nn5,'-y*'); hold on

figure(2)
plot(put_sim(:,1),'or','MarkerSize',8),hold on
plot(all_outputs(:,1),'b*')

xlabel('Vreme kretanja od cvora do cvora')
ylabel('Cvor')
title('Obucavajuci skup za generisanje putanje mobilnog robota')
h = legend('Nauceni cvor','Zeljeni cvor',1);

```