

Основне академске студије
Информациони инжењеринг

Методе и технике науке о подацима

Претраге

(материјали за предавања)

1. Увод у претраге
2. Ненавођене претраге
3. Навођене претраге
4. Извори и литература

Задатак

проблем

полазећи од почетне ситуације потребно је наћи начин да се оствари нека унапред дата жељена ситуација уз поштовање постављених правила

спецификација околности

неопходно наћи одговарајуће решење

Задатак

кључни концепти

стање

почетно стање

околности у почетном тренутку

циљно стање

околности које је неопходно постићи

текуће стање

околности у посматраном тренутку

радња (корак, акција)

делатност која може довести до промене у околностима

посредством спровођења радње долази до преласка из једног у друго стање

цена

трошак повезан са спровођењем радњи

захтевност спровођења радњи

простор стања

стања која су достижна из почетног стања посредством спровођења радњи

путања

низ радњи путем којих се сукцесивно одвијају прелази између стања

Задатак

елементи задатка

скуп стања

стања која су остварива

почетно стање

опис почетних услова

у почетном тренутку текуће стање одговара почетном стању

скуп радњи

радње које су спроводиве

радња може бити спроводива у једном или више стања

правила прелаза

спровођење радње доводи до преласка из посматраног у ново стање

познато је које ће стање бити достигнуто применом неке радње која је спроводива у посматраном стању

провера завршетка

испитивање да ли се текуће стање поклапа са циљним стањем

у случају поклапања сматра се да је задатак решен

ценовна функција

функција помоћу које је могуће одредити цену за дату путању

Задатак

елементи задатка могу бити постављени на разне начине
постоје бројне варијације у задацима и њиховој структури
провера завршетка

- може постојати више циљних стања

- потребно наћи начин како да барем једно од њих буде достигнуто
 - циљно стање не мора бити познато унапред

- познат само услов који мора бити задовољен у циљном стању

ценовна функција

- не мора бити увек битна

- може бити довољно да буде пронађено решење без обзира на квалитет

Претрага

поступак тражења решења за постављени задатак

исход претраге је решење

решење у облику путање која води од почетног до циљног стања преко низа радњи

Претрага – ток

један општи ток претраге

креће се од почетног стања као текућег стања

врши се обрада текућег стања

спроводи се провера завршетка за текуће стање

ако је провера успешна претрага се завршава успешно

у претрази је пронађено решење које одговара достизању циљног стања

потребно је евидентирати радње које воде до циљног стања

ако је провера неуспешна изводи се развој (развијање) текућег стања, односно испитује се која су стања непосредно достижна спровођењем могућих радњи за текуће стање

непосредно достижна стања бивају евидентирана као стања за даљу обраду

пored стања за даљу обраду која су већ евидентирана

текуће стање се сматра обрађеним

неко од стања за даљу обраду бива одабрано као текуће стање за нову обраду

за ново текуће стање је потребно извршити обраду по описаном поступку

ако не постоји ниједно стање за даљу обраду, претрага се завршава неуспешно

Претрага – стабло претраге

ток претраге може бити описан стаблом претраге

чвор стабла претраге одговара стању

корен стабла претраге одговара почетном стању

једно те исто стање може бити повезано с различитим чворовима стабла претраге

грана стабла претраге одговара преласку између два стања

одговара појединачној изведеној радњи (кораку)

за корак може бити везана цена

пут у стаблу претраге одговара секвенци прелазака која води од једног до другог стања

одговара секвенци изведених радњи (корака)

пут обухвата низ чворова и гране које редом повезују те чворове

исти чвор не може бити више од једном заступљен у оквиру пута

иста грана не може бити више од једном заступљена у оквиру пута

посматрају се путеви који почињу од корена, који одговара почетном стању

за појединачне путеве постоје одговарајуће цене

цена пута углавном као збир цена од појединачних корака који чине пут

како претрага тече тако се стабло претраге развија

стабло претраге може бити бесконачно

Претрага – стратегија

стратегија претраге

принципи усмеравања тока претраге

углавном се тичу правила за избор стања или чвора стабла претраге за даљу обраду

врсте процедура претраге према стратегији

ненавођене претраге

(неинформисане, необавештене, неусмерене, слепе)

током претраге не узимају се обзир додатне информације
само информације дате у задатку

навођене претраге

(информисане, обавештене, усмерене)

током претраге узимају се у обзир додатне информације

информације које могу довести до квалитетнијег проналажења решења
познате и као хеуристичке претраге

уводи се хеуристичка функција у поступак вредновања посматраног стања или чвора стабла претраге

Претрага – особине

особине процедура претраге

у погледу спровођења претраге

временска сложеност

перформансе у погледу трајања поступка претраге

просторна сложеност

перформансе у погледу количине меморије потребне током извођења претраге

у погледу резултата претраге

потпуност (комплетност)

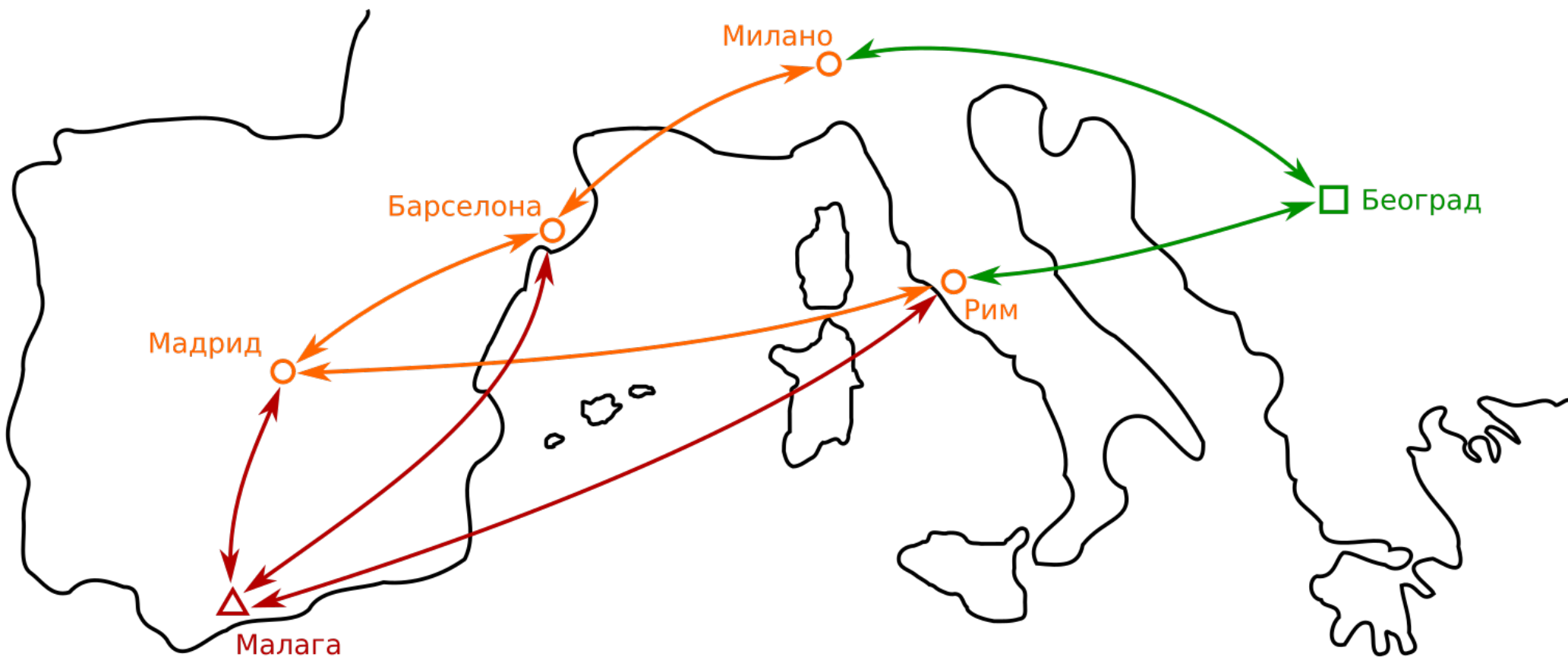
способност да претрага увек проналази решење, под условом да решење за постављени задатак постоји

оптималност

способност да претрага проналази најбоље могуће (оптимално) решење

Увод у претраге

Пример задатка



Пример задатка

елементи задатка

скуп стања

{Београд, Милано, Рим, Барселона, Мадрид, Малага}

почетно стање

Београд

скуп радњи

{за Милано, за Рим, за Барселону, за Мадрид, за Малагу, за Београд}

провера завршетка

текуће стање = Малага

ценовна функција

цена пута

збир цена појединачних корака током пута

цена корака

јединична цена

могуће другачије цене корака

раздаљина, трајање или новчани трошкови за један корак (етапу) пута

Пример задатка

елементи задатка

правила прелаза

Београд -- (за Милано) -> Милано

Београд -- (за Рим) -> Рим

Милано -- (за Београд) -> Београд

Милано -- (за Барселону) -> Барселона

Рим -- (за Београд) -> Београд

Рим -- (за Мадрид) -> Мадрид

Рим -- (за Малагу) -> Малага

Барселона -- (за Мадрид) -> Мадрид

Барселона -- (за Малагу) -> Малага

Барселона -- (за Милано) -> Милано

Мадрид -- (за Барселону) -> Барселона

Мадрид -- (за Малагу) -> Малага

Мадрид -- (за Рим) -> Рим

Малага -- (за Барселону) -> Барселона

Малага -- (за Мадрид) -> Мадрид

Малага -- (за Рим) -> Рим

Пример задатка

елементи задатка

помоћни елементи

помоћне ознаке за стања (ознаке главних аеродрома)

Београд: BEG

Милано: MXP

Рим: FCO

Барселона: BCN

Мадрид: MAD

Малага: AGP

1. Увод у претраге
- 2. Ненавођене претраге**
3. Навођене претраге
4. Извори и литература

Ненавођене претраге

претрага у дубину

ограничена претрага у дубину

итеративна претрага у дубину

претрага у ширину

униформна претрага

двосмерна (бидирекциона) претрага

Претрага у дубину

ток претраге преко стабла претраге

1. формира се корен стабла

корен стабла одговара почетном стању

2. за текући чвор поставља се корен стабла

3. спроводи се обрада текућег чвора

провера да ли стање повезано с текућим чвором одговара циљном стању

ако одговара, онда је претрага успешно завршена

пут од корена до текућег чвора описује решење

ако не одговара, онда се ради развој чвора (развој стања повезаног с текућим чвором)

за свако стање добијено развојем формира се по један чвор као директно подређени чвор текућем чвору

текући чвор се сматра обрађеним

4. бира се нови текући чвор

ако не постоји ниједан необрађени чвор, онда је претрага неуспешно завршена

ако постоји барем један необрађени чвор

међу чворовима који нису обрађени поставља се за текући чвор онај који је на највећој дубини (у случају више таквих бира се један, може онај најстарији)

5. прелази се на обраду текућег чвора (повратак на корак 3)

Претрага у дубину

енгл. *depth-first search*

стабло претраге се примарно развија у дубину

чвор X – први подређени за чвор X – први подређени од првог подређеног за чвор X – ...

по потреби долази до повратка на више нивое стабла претраге ради испитивања других праваца претраге

начелно не захтева велику количину меморије

у односу на неке друге врсте претрага

нпр. у односу на претрагу у ширину

није неопходно чувати целокупно стабло претраге

потребно чувати чворове који су на путу текућег тока претраге

потребно чувати чворове који нису обрађени а директно су повезани с путем текућег тока претраге

чвор који је обрађен може бити уклоњен након што су сви његови подређени чворови обрађени

Претрага у дубину

постоји ризик од лошег избора текућег чвора током претраге

може уследити дуготрајно тражење

може бити пронађено решење које је далеко од оптималног

може доћи до бесконачног развоја стабла претраге

онда претрага не може дати решење, чак и ако решење постоји

поступак претраге може бити унапређен

приликом развоја стања повезаног с текућим чвором

ако је ново стање које је добијено развојем повезано с неким чвором који је међу надређеним чворовима, није потребно формирати чвор за ново стање

приликом развоја стања повезаног с текућим чвором

ако је ново стање које је добијено развојем повезано с неким чвором који је већ обрађен, није потребно формирати чвор за ново стање

ако је ново стање које је добијено развојем повезано с неким постојећим чвором који још није обрађен, није потребно формирати чвор за ново стање

главна имплементација

обухвата коришћење стека

експлицитно

имплицитно преко рекурзије

Претрага у дубину – особине

временска сложеност је $O(b^m)$

b је фактор гранања за стабло претраге

m је максимална дубина у стаблу претраге

просторна сложеност је $O(bm)$

b је фактор гранања за стабло претраге

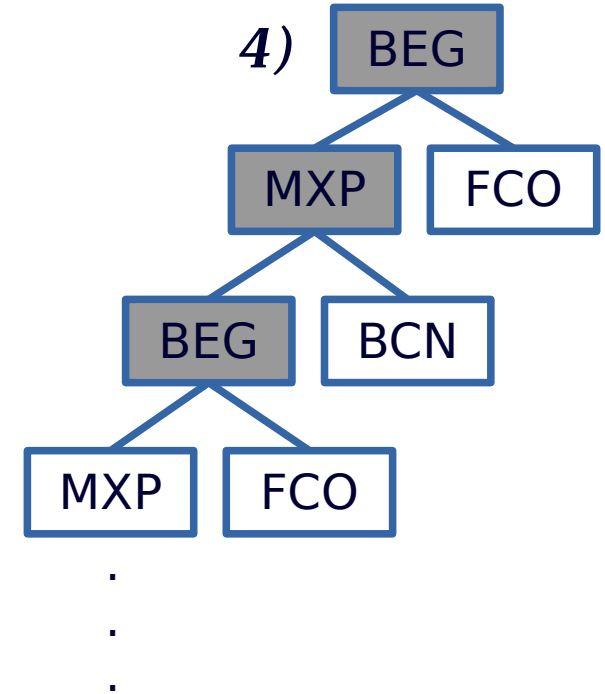
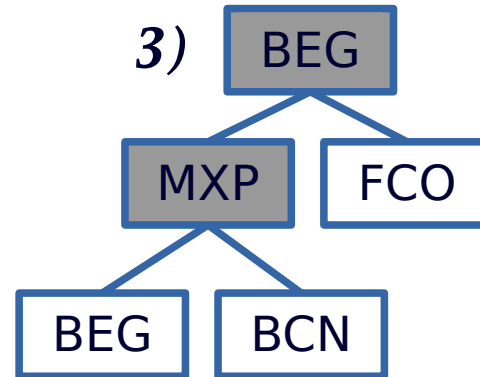
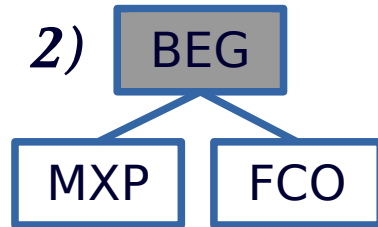
m је максимална дубина у стаблу претраге

ПОТПУНОСТ НЕ ВАЖИ

ОПТИМАЛНОСТ НЕ ВАЖИ

Ненавођене претраге

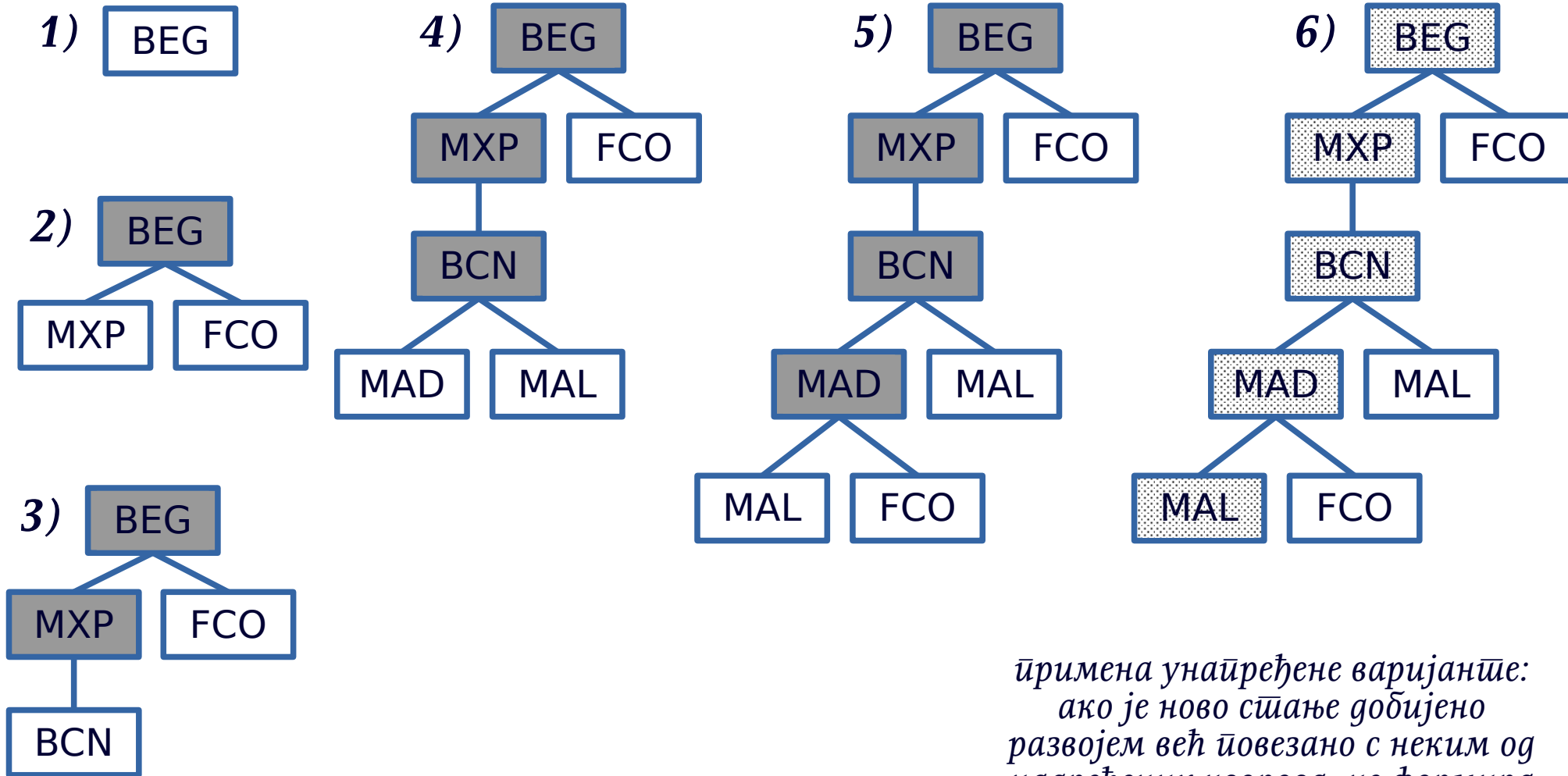
Претрага у дубину – пример 1



*бесконечно се развија
стабло претраге
(ионављајућа секвенца
BEG-MXP)*

Ненавођене претраге

Претрага у дубину – пример 2



*примена унапређене варијанте:
ако је ново стање добијено
развојем већ повезано с неким од
надређених чворова, не формира
се чвор за ново стање*

Ограничена претрага у дубину

енгл. *depth-limited search*

слична претрази у дубину

уводи се ограничење дубине t

максимална гранична дубина до које може бити вршено тражење

развој стања повезаног с текућим чвором неће бити урађен ако би тиме било нарушено ограничење дубине

за чворове на дубини t неће бити формиран подређени чворови

у односу на претрагу у дубину

спречава се евентуално бесконачно развијање стабла претраге

постоји ризик од лошег избора вредности за ограничење дубине

проналажење решења може бити онемогућено, чак и када решење постоји
остаје могућност дуготрајног тражења

може бити пронађено решење које је далеко од оптималног

природа постављеног задатка може бити од користи приликом
одређивања вредности за ограничење дубине

Ограничена претрага у дубину – особине

временска сложеност је $O(b^t)$

b је фактор гранања за стабло претраге

t је ограничење дубине

просторна сложеност је $O(bt)$

b је фактор гранања за стабло претраге

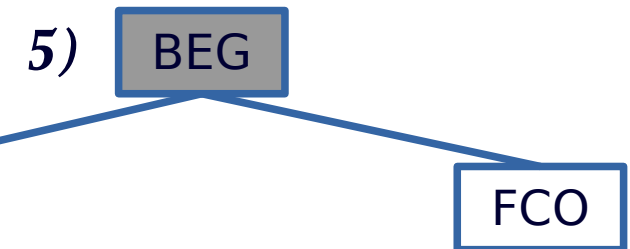
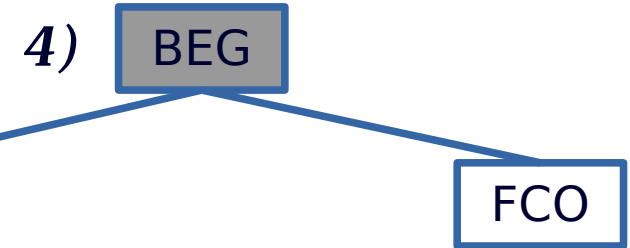
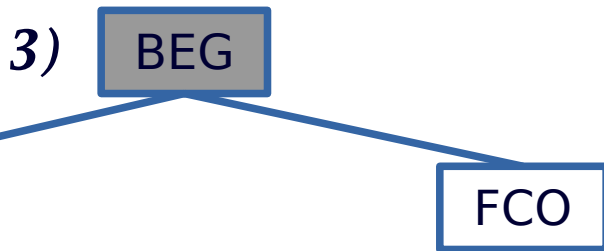
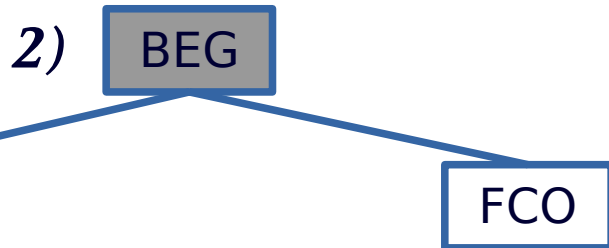
t је ограничење дубине

ПОТПУНОСТ НЕ ВАЖИ

ОПТИМАЛНОСТ НЕ ВАЖИ

Ненавођене претраге

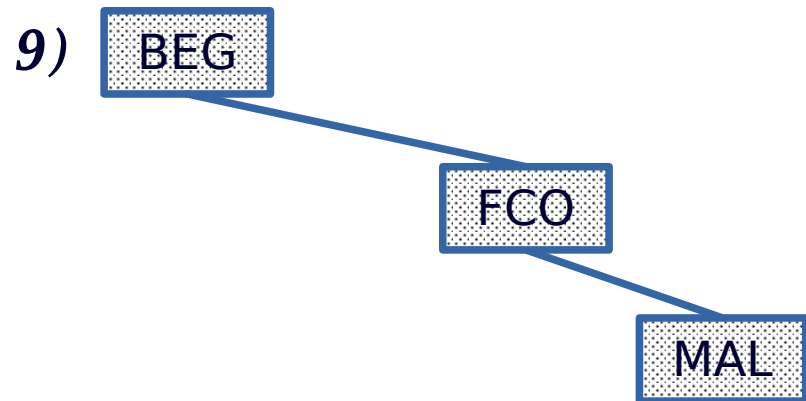
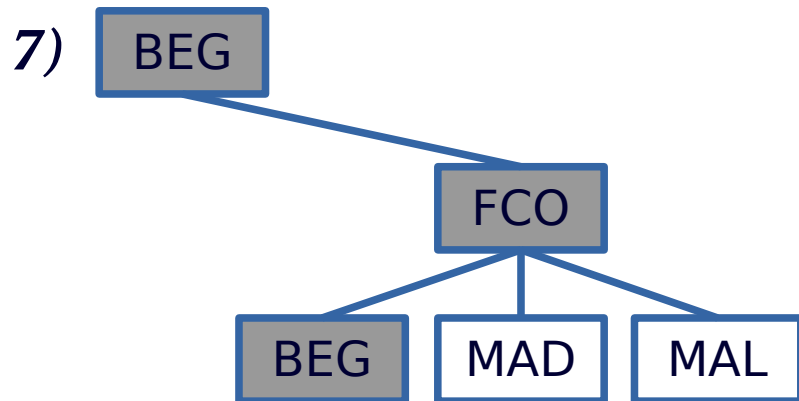
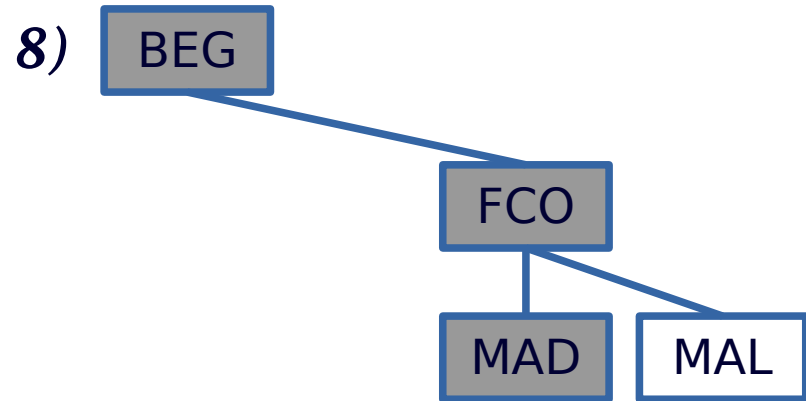
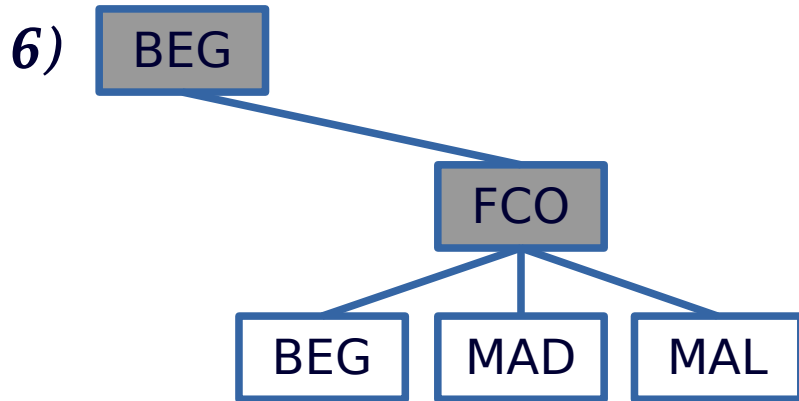
Ограничена претрага у дубину – пример



ограничење дубине $t = 2$
(шражење – део 1)

Ненавођене претраге

Ограничена претрага у дубину – пример



ограничење дубине $t = 2$
(шражење – део 2)

Итеративна претрага у дубину

енгл. *iterative deepening search*

слична претрази у дубину и ограниченој претрази у дубину

ослања се на концепт ограничења дубине који постоји код ограничене претраге у дубину

поступно дубинско тражење по све већим и већим дубинама

примењује се ограничена претрага у дубину за различите вредности ограничења дубине

у свакој итерацији спроводи се претрага у дубину за конкретну вредност ограничења дубине

креће се од случаја нултог ограничења дубине, па се поступно повећава вредност ограничења дубине

поседује неке одлике и претраге у дубину и претраге у ширину

Итеративна претрага у дубину

представља побољшање у односу на претрагу у дубину и ограничену претрагу у дубину

ако решење постоји, биће пронађено

под условом да је фактор гранања коначан

не долази до бесконачног развоја стабла претраге јер је у свакој итерацији ограничена дубина до које се претрага одвија

биће увек пронађено решење на најмањој дубини

јер се по потреби проверавају сви чворови до одређене дубине и претрага се поступно кроз додатне итерације спушта на све веће дубине

пронађено решење може али и не мора бити оптимално

у зависности од ценовне функције

претрага може бити нешто захтевнија у погледу трајања

у свакој итерацији претрага започиње од корена стабла претраге

што је неки чвор ближи корену стабла претраге то ће начелно више пута бити испитиван и развијан

може бити погодан начин претраге у појединим случајевима

када је простор стања велик и дубина до које се простире решење је непозната

Итеративна претрага у дубину – особине

временска сложеност је $O(b^d)$

b је фактор гранања за стабло претраге

d је дубина која одговара решењу с најмањом дужином простирања

просторна сложеност је $O(bd)$

b је фактор гранања за стабло претраге

d је дубина која одговара решењу с најмањом дужином простирања

ПОТПУНОСТ ВАЖИ

под условом да је фактор гранања коначан

ОПТИМАЛНОСТ ВАЖИ

ако ни за један пут који се простире из корена до дубине d' не постоји пут који се простире из корена до мање дубине d'' и има вишу цену

Претрага у ширину

ток претраге преко стабла претраге

1. формира се корен стабла

корен стабла одговара почетном стању

2. за текући чвор поставља се корен стабла

3. спроводи се обрада текућег чвора

провера да ли стање повезано с текућим чвором одговара циљном стању

ако одговара, онда је претрага успешно завршена

пут од корена до текућег чвора описује решење

ако не одговара, онда се ради развој чвора (развој стања повезаног с текућим чвором)

за свако стање добијено развојем формира се по један чвор као директно подређени чвор текућем чвору

текући чвор се сматра обрађеним

4. бира се нови текући чвор

ако не постоји ниједан необрађени чвор, онда је претрага неуспешно завршена

ако постоји барем један необрађени чвор

међу чворовима који нису обрађени поставља се за текући чвор онај који је на најмањој дубини (у случају више таквих бира се један, може онај најстарији)

5. прелази се на обраду текућег чвора (повратак на корак 3)

Претрага у ширину

енгл. *breadth-first search*

стабло претраге се примарно развија у ширину

чвор X – први подређени за чвор X – други подређени за чвор X – ... – први подређени од првог подређеног за чвор X – ...

обрада се спроводи поступно по нивоима стабла претраге

од највишег нивоа стабла претраге према све нижим

сваки ниво стабла претраге се потпуно обрађује пре преласка на наредни ниво

не долази до повратка на већ обрађени ниво стабла претраге

ако решење постоји биће пронађено

под условом да је фактор гранања коначан

биће пронађено решење на најмањој дубини

пронађено решење може али и не мора бити оптимално

у зависности од ценовне функције

Претрага у ширину

претрага може бити захтевна у погледу трајања и количине потребне меморије

ако се решење простире до велике дубине, потребно је прво обрадити начелно све чворове на мањим дубинама

начелно је потребно чувати целокупно стабло претраге

што начелно није неопходно код претраге у дубину

поступак претраге може бити унапређен

провера завршетка може бити спроведена већ приликом формирања нових чворова у стаблу претраге

приликом развоја стања повезаног с текућим чвором

ако је ново стање које је добијено развојем повезано с неким чвором који је већ обрађен, није потребно формирати чвор за ново стање

ако је ново стање које је добијено развојем повезано с неким постојећим чвором који још није обрађен, није потребно формирати чвор за ново стање

главна имплементација

обухвата коришћење реда

Претрага у ширину – особине

временска сложеност је $O(b^{d+1})$ или $O(b^d)$

b је фактор гранања за стабло претраге

d је дубина која одговара решењу с најмањом дужином простирања

просторна сложеност је $O(b^d)$

b је фактор гранања за стабло претраге

d је дубина која одговара решењу с најмањом дужином простирања

ПОТПУНОСТ ВАЖИ

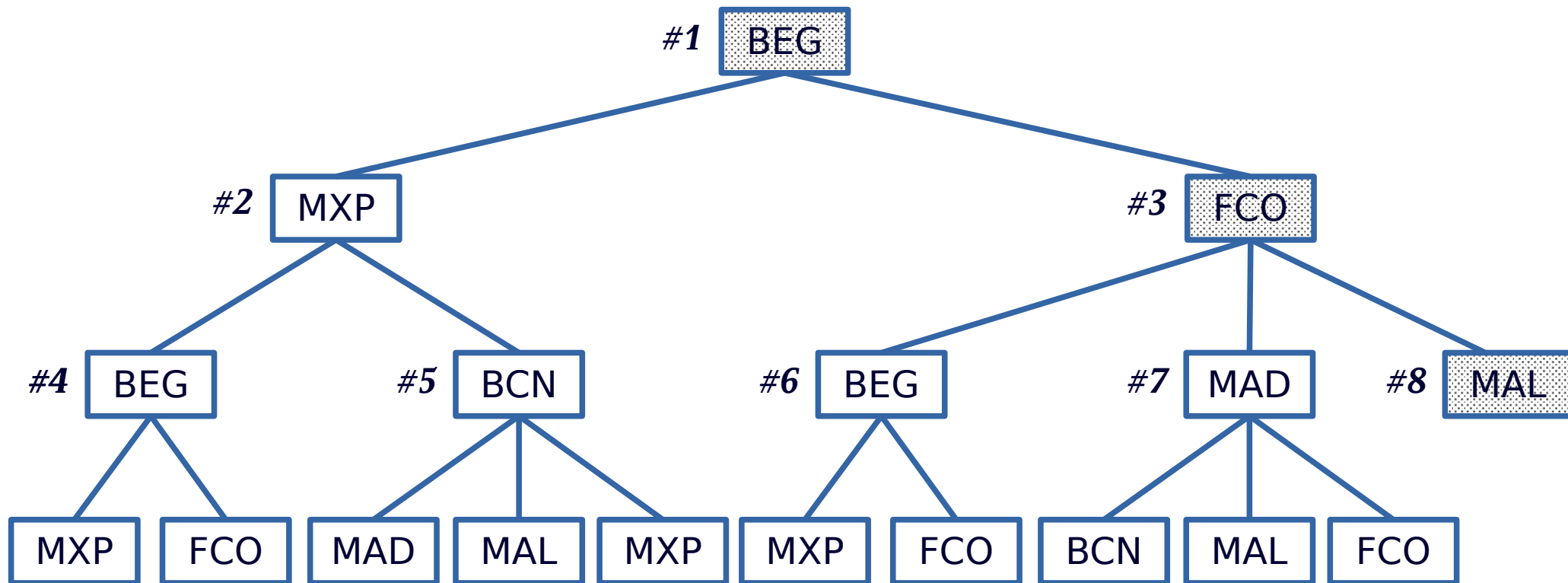
под условом да је фактор гранања коначан

ОПТИМАЛНОСТ ВАЖИ

ако ни за један пут који се простире из корена до дубине d' не постоји пут који се простире из корена до мање дубине d'' и има вишу цену

Ненавођене претраге

Претрага у ширину – пример



Униформна претрага

ток претраге преко стабла претраге

1. формира се корен стабла

корен стабла одговара почетном стању

2. за текући чвор поставља се корен стабла

3. спроводи се обрада текућег чвора

провера да ли стање повезано с текућим чвором одговара циљном стању

ако одговара, претрага је успешно завршена

пут од корена до текућег чвора описује решење

ако не одговара, развија се чвор (развој стања повезаног с текућим чвором)

за свако стање добијено развојем формира се по један чвор као директно подређени чвор текућем чвору

текући чвор се сматра обрађеним

4. бира се нови текући чвор

ако не постоји ниједан необрађени чвор, онда је претрага неуспешно завршена

ако постоји барем један необрађени чвор

међу чворовима који нису обрађени поставља се за текући чвор онај до којег одговарајући пут од корена има најнижу цену (у случају више таквих чворова бира се један, може онај најстарији)

5. прелази се на обраду текућег чвора (повратак на корак 3)

Униформна претрага

енгл. *uniform-cost search*

примена Дајкстриног алгоритма за потребе претраге

приликом тражења узима се у обзир цена пута

потребно је наћи оптимално (најбоље) решење

решење с најнижом ценом

решење је пут што почиње од корена, који одговара почетном стању, и завршава се код неког чвора који одговара циљном стању, при чему такав пут има најнижу цену међу свим путевима од корена до чвора који одговара циљном стању

Униформна претрага

поступак претраге може бити унапређен

приликом развоја стања повезаног с текућим чвором

ако је ново стање које је добијено развојем повезано с неким чвором који је већ обрађен, није потребно формирати чвор за ново стање

ако је ново стање које је добијено развојем повезано с неким постојећим чвором који још није обрађен и чији пут би имао вишу цену него пут за чвор за ново стање, постојећи чвор може бити елиминисан

чвор за ново стање потребно је формирати

главна имплементација

обухвата коришћење приоритетног реда

Униформна претрага

основна претрага у ширину је посебан случај униформне претраге

случај када се за цену пута који почиње у корену узима дубина до које пут допире

дубина на којој се чвор на крају пута налази

могуће потешкоће у проналажењу решења

постојање негативних цена корака

могућа препрека у налажењу оптималног решења

могуће бесконачно развијање стабла претраге услед постојања корака чија је цена нула

тада није могуће наћи решење

постојање корака чија је цена нула не мора увек довести до бесконачног развијања стабла

Униформна претрага – особине

временска сложеност је $O(b^{1+\lceil C^*/\varepsilon \rceil})$

b је фактор гранања за стабло претраге

C^* је цена оптималног решења

ε је претпостављена најнижа позитивна цена корака

просторна сложеност је $O(b^{1+\lceil C^*/\varepsilon \rceil})$

b је фактор гранања за стабло претраге

C^* је цена оптималног решења

ε је претпостављена најнижа позитивна цена корака

ПОТПУНОСТ ВАЖИ

под условом да је фактор гранања коначан и да су цене корака веће или једнаке с неком позитивном константом ε

ОПТИМАЛНОСТ ВАЖИ

под условом да не постоје негативне цене корака

Двосмерна (бидирекциона) претрага

енгл. *bidirectional search*

представља сложену претрагу

могу се у паралели спровести две једноставније претраге
једна претрага креће од почетног стања (претрага унапред)
друга претрага креће од циљног стања (претрага уназад)
потребно је да дође до преклапања токова две претраге

потребно је да обе претраге у свом току дођу до испитивања произвољног
заједничког стања

провера да ли је достигнуто неко заједничко стање

може бити на почетку обраде текућег чвора

може бити при формирању нових чворова током обраде текућег чвора

повезивањем два пута до заједничког стања добија се решење

један пут пронађен у претрази унапред

други пут пронађен у претрази уназад

разне врсте претрага могу бити комбиноване

Двосмерна (бидирекциона) претрага

за неке врсте задатака двосмерна претрага је непрактична

могуће потешкоће у вези с претрагом уназад

провера завршетка за неке врсте задатака не мора се заснивати на једноставној провери поклапања достигнутог и циљног стања

облик циљног стања не мора бити унапред познат, што може отежати започињање тражења уназад

потребно је за неко стање одредити које су радње директно довеле до тог стања и то из којих стања

одређивање претходних радњи и стања није увек тривијално

ПОГОДНОСТИ

одређене перформансе могу бити унапређене у односу на одређене случајеве појединачних претрага

првенствено у погледу трајања претраге

Двосмерна (бидирекциона) претрага – особине

временска сложеност је $O(b^{d/2})$

b је фактор гранања за стабло претраге

d је дубина која одговара решењу с најмањом дубином простирања

просторна сложеност је $O(b^{d/2})$

b је фактор гранања за стабло претраге

d је дубина која одговара решењу с најмањом дубином простирања

ПОТПУНОСТ МОЖЕ ВАЖИТИ

под условом да је фактор гранања коначан

зависно од врста претрага које се користе

важи када претрага у ширину и као претрага унапред и као претрага уназад

ОПТИМАЛНОСТ МОЖЕ ВАЖИТИ

зависно од врста претрага које се користе и ценовне функције

важи када претрага у ширину и као претрага унапред и као претрага уназад

ако ни за један пут који се простире из корена до дубине d' не постоји пут који се простире из корена до мање дубине d'' и има вишу цену

1. Увод у претраге
2. Ненавођене претраге
- 3. Навођене претраге**
4. Извори и литература

Хеуристике

хеуристика као техника за откривање знања, налажење решења и доношење одлука

хеуристика у контексту претрага

помоћно правило у решавању задатка

процена повољности могућег правца претраге у погледу проналажења решења

обично подразумева коришћење знања која нису непосредно уграђена у задатак који треба да буде решен

Хеуристике

хеуристичка функција $h(x)$

формализација хеуристике у контексту претрага

x је чвор у стаблу претраге

$h(x)$ је процена цене за најповољнију путању од чвора x до циљног чвора, односно до чвора чије повезано стање одговара успешном завршетку претраге

вредности хеуристичке функције су обично ненегативни реални бројеви, при чему циљном чвору одговара вредност 0

Навођене претраге

претраге код којих се током трагања за решењем користе додатне информације

додатне информације нису доступне непосредно у опису задатка, али се очекује да могу бити корисне у претрази

додатне информације су уграђене у поступак претраге преко одговарајуће хеуристичке функције

Претрага по својству

енгл. *best-first search*

начелно се изводи у складу с општим током претраге преко стабла претраге

карактерише је усмеравање тока претраге према чворовима у стаблу претраге који имају најбоље особине

за текући чвор у датом кораку претраге бира се чвор који је најбољи по одабраном својству

избор текућег чвора спроводи се помоћу функције евалуације $f(x)$

x је чвор у стаблу претраге који се испитује као могући текући чвор
ако функција евалуације одговара цени, бира се чвор с којим је повезана најнижа цена

облик функције евалуације зависи од врсте претраге и конкретног задатка
у функцију евалуације углавном је уграђена хеуристичка функција $h(x)$

Навођене претраге

похлепна претрага

претрага A^*

претрага IDA^*

претрага SMA^*

рекурзивна претрага по својству

Похлепна претрага

енгл. *greedy search (greedy best-first search)*

претрага се усмерава тако да са сваким кораком процењено приближавање циљу (успешном завршетку) буде највеће могуће

процењено приближавање не мора одговарати стварном приближавању
за текући чвор бира се чвор који делује као најповољнији у погледу цене пута од њега до циљног чвора

функција евалуације своди се на хеуристичку функцију

$$f(x) = h(x)$$

x је чвор у стаблу претраге

за текући чвор бира се чвор којем одговара најмања вредност хеуристичке функције

Похлепна претрага

избор хеуристичке функције може имати знатан утицај на ток претраге и квалитет евентуалног решења

може доћи до бесконачног развоја стабла претраге

поступак претраге може бити унапређен

приликом развоја стања повезаног с текућим чвором

ако је ново стање које је добијено развојем повезано с неким чвором који је већ обрађен, није потребно формирати чвор за ново стање

ако је ново стање које је добијено развојем повезано с неким постојећим чвором који још није обрађен, није потребно формирати чвор за ново стање

Похлепна претрага – особине

временска сложеност је $O(b^m)$

b је фактор гранања за стабло претраге

m је максимална дубина

просторна сложеност је $O(b^m)$

b је фактор гранања за стабло претраге

m је максимална дубина

ПОТПУНОСТ НЕ ВАЖИ

за унапређену варијанту потпуност може важити у одређеним случајевима

ОПТИМАЛНОСТ НЕ ВАЖИ

Навођене претраге

Пример задатка

хеуристичка функција

заснована на удаљеностима између градова

вредност хеуристичке функције за дати чвор

удаљеност

од града (стања) који одговара датом чвору

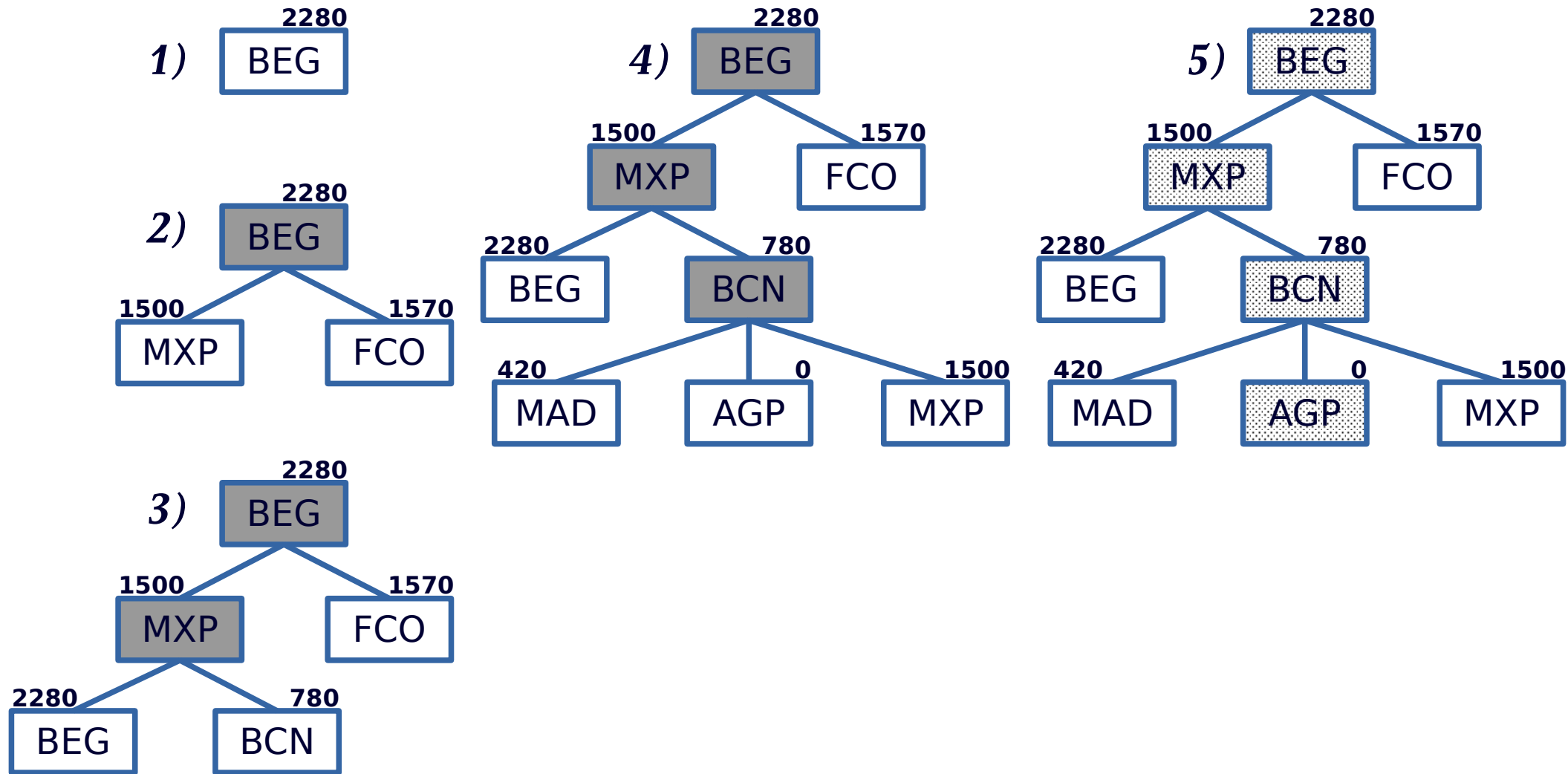
до града (стања) који одговара циљном чвору

приближне раздаљине између градова у километрима

	AGP					
AGP	0					
		BEG				
BEG	2280	0				
			BCN			
BCN	780	1540	0			
				FCO		
FCO	1570	730	870	0		
					MAD	
MAD	420	2040	510	1370	0	
						MXP
MXP	1500	890	730	480	1200	0

Навођене претраге

Похлепна претрага – пример



Претрага A^*

енгл. *A* search (A-star search)*

током претраге узимају се у обзир и квалитет пређеног пута и квалитет преосталог пута до проналажења решења

претрага A^* је темељнији поступак у односу на похлепну претрагу
код похлепне претраге разматра се само квалитет преосталог пута
користи се само хеуристика

претрага A^* је темељнији поступак у односу на униформну претрагу
код униформне претраге разматра се само квалитет пређеног пута
не користи се хеуристика

Претрага A^*

ток претраге се усмерава на основу цене пређеног пута до посматраног чвора у стаблу претраге и процењене цене од посматраног чвора до циљног чвора

функција евалуације обухвата две компоненте

$$f(x) = g(x) + h(x)$$

x је чвор у стаблу претраге

$g(x)$ одговара цени пута од корена до чвора x

$h(x)$ одговара процењеној цени за најповољнији пут од чвора x до циљног чвора

$h(x)$ је хеуристичка функција

Претрага A^*

под одређеним условима претрага A^* ће увек пронаћи оптимално решење, ако решење постоји

фактор гранања је коначан и

све цене корака су веће или једнаке с неком позитивном константом ϵ и

коришћена хеуристичка функција $h(x)$ је допустива хеуристика

допустива хеуристика је она чија је вредност мања или једнака са стварном ценом најповољније путање од чвора X до циљног чвора

допустива хеуристика не прецењује повољност пута од чвора X до циљног чвора

Претрага A^* – особине

временска сложеност

зависи од одабране хеуристике и природе задатка

може бити експоненцијалног карактера у одређеним случајевима

основа је фактор гранања за стабло претраге

експонент се односи на грешку хеуристике

одговарајуће одступање коришћене хеуристике од идеалне хеуристике

просторна сложеност

потребно је чувати податке о свим формираним чворовима

ПОТПУНОСТ ВАЖИ

под условом да је фактор гранања коначан и да су цене корака веће или једнаке с неком позитивном константом ϵ

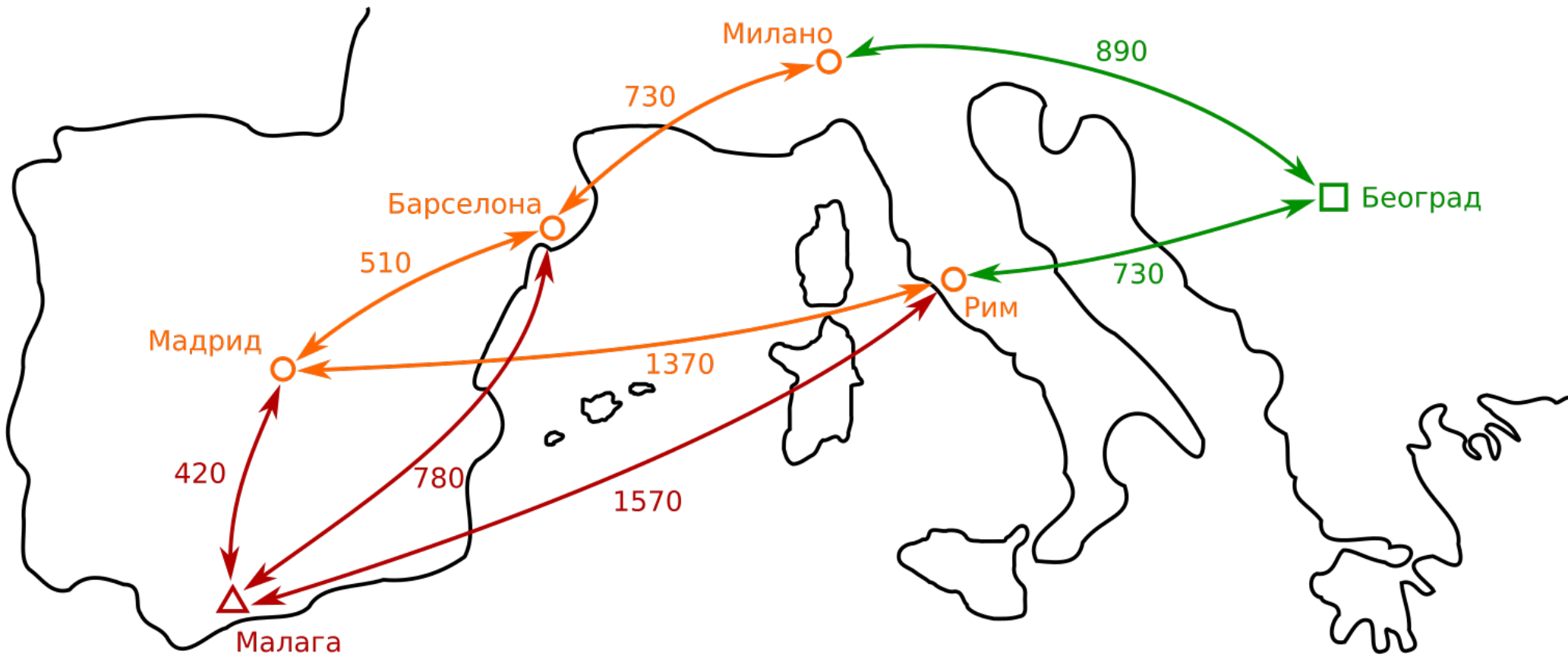
ОПТИМАЛНОСТ ВАЖИ

под условом да је фактор гранања коначан и све цене корака су веће или једнаке с неком позитивном константом ϵ и коришћена хеуристичка функција $h(x)$ је допустива хеуристика

Навођене претраге

Пример задатка – проширење

*уведене цене појединачних
корака*

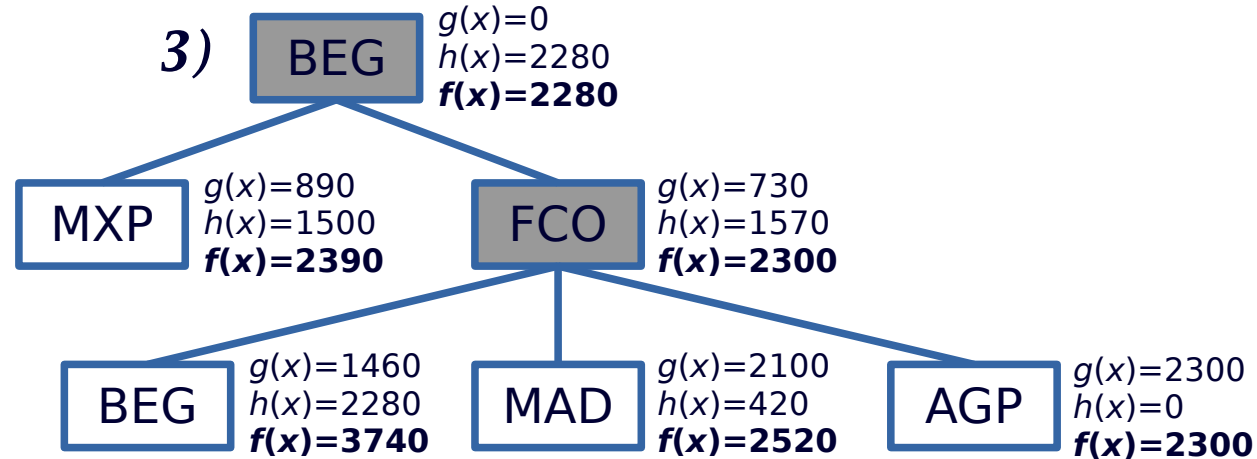


Навођене претраге

Претрага A^* – пример

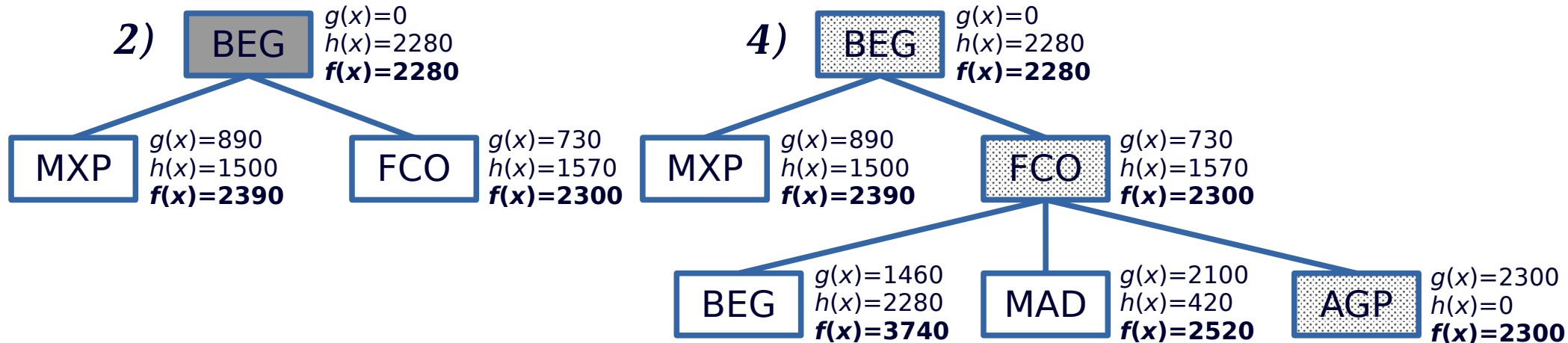
1) **BEG** $g(x)=0$
 $h(x)=2280$
 $f(x)=2280$

3) **BEG** $g(x)=0$
 $h(x)=2280$
 $f(x)=2280$



2) **BEG** $g(x)=0$
 $h(x)=2280$
 $f(x)=2280$

4) **BEG** $g(x)=0$
 $h(x)=2280$
 $f(x)=2280$



Претрага *IDA**

енгл. *iterative deepening A* search*

итеративна претрага A^*

примена идеје итеративног претраживања која постоји код итеративне претраге у дубину

спровођење претраге се у свакој итерацији ограничава у односу на вредност функције евалуације $f(x)$

уводи се гранична вредност V

за граничну вредност узима се најмања вредност функције евалуације која је била у претходној итерацији изнад граничне вредности

за граничну вредност у првој итерацији узима се вредност функције евалуације од почетног чвора у стаблу претраге

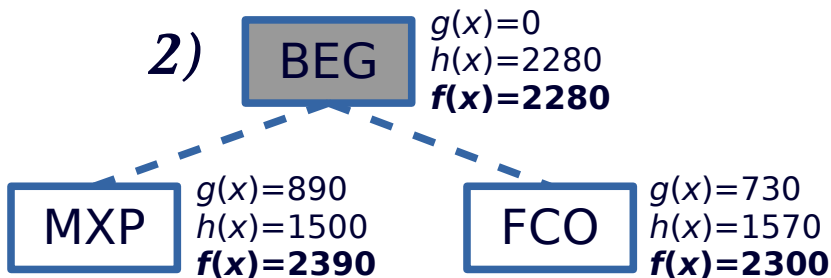
чвор чија је вредност функције евалуације већа од граничне вредности неће бити развијен у текућој итерацији

начелно има мању меморијска захтевност у односу на основну претрагу A^*

Навођене претраге

Претрага IDA^* – пример

1) **BEG** $g(x)=0$
 $h(x)=2280$
 $f(x)=2280$



оградицење $v = 2280$
(шражење – ишерација 1)

Навођене претраге

Претрага IDA^* – пример

1) **BEG** $g(x)=0$
 $h(x)=2280$
 $f(x)=2280$

3) **BEG** $g(x)=0$
 $h(x)=2280$
 $f(x)=2280$

MXP $g(x)=890$
 $h(x)=1500$
 $f(x)=2390$

FCO $g(x)=730$
 $h(x)=1570$
 $f(x)=2300$

BEG $g(x)=1460$
 $h(x)=2280$
 $f(x)=3740$

MAD $g(x)=2100$
 $h(x)=420$
 $f(x)=2520$

AGP $g(x)=2300$
 $h(x)=0$
 $f(x)=2300$

2) **BEG** $g(x)=0$
 $h(x)=2280$
 $f(x)=2280$

4) **BEG** $g(x)=0$
 $h(x)=2280$
 $f(x)=2280$

MXP $g(x)=890$
 $h(x)=1500$
 $f(x)=2390$

FCO $g(x)=730$
 $h(x)=1570$
 $f(x)=2300$

MXP $g(x)=890$
 $h(x)=1500$
 $f(x)=2390$

FCO $g(x)=730$
 $h(x)=1570$
 $f(x)=2300$

BEG $g(x)=1460$
 $h(x)=2280$
 $f(x)=3740$

MAD $g(x)=2100$
 $h(x)=420$
 $f(x)=2520$

AGP $g(x)=2300$
 $h(x)=0$
 $f(x)=2300$

оградицење $v = 2300$
(шражење – итерација 2)

Претрага SMA*

енгл. *simplified memory-bounded A* search*

варијанта претраге A^* која је прилагођена ограниченом меморијском простору

користи се сложенији поступак праћења повољности чвора у стаблу претраге

могу постојати помоћни подаци или измењени основни подаци који указују на повољност развоја одређеног правца претраге

спроводи се развој најповољнијих чворова у стаблу претраге

у случају попуњености расположивог меморијског простора, потребно је ослободити најнеповољнији лисни чвор

потребно је у надређеном чвору евидентирати додатни податак о вредности функције евалуације за ослобођени чвор

може доћи до поновног формирања ослобођеног чвора у зависности од тока претраге и повољности доступних чворова

Рекурзивна претрага по својству

енгл. *recursive best-first search*

рекурзивна процедура тражења с развојем пута на основу вредности функције евалуације

у стаблу претраге рекурзивно се развија главни пут, који је најповољнији према функцији евалуације

додатно су уз чворове главног пута који су обрађени евидентирани подаци о могућим помоћним путевима

евидентира се вредност функције евалуације за помоћни пут за посматрани чвор помоћни пут је онај пут који је могуће развијати даље од неког чвора надређеног посматраном и који је други најповољнији пут у односу на главни пут

Рекурзивна претрага по својству

енгл. *recursive best-first search*

рекурзивна процедура тражења с развојем пута на основу вредности функције евалуације

у случају да повољност главног пута може постати мања од повољности помоћног пута, прелази се на развој помоћног пута

напушта се развој главног пута

долази до рекурзивног враћања уз главни пут до чвора од којег може бити развијан помоћни пут

помоћни пут постаје нови главни пут претраге

чворови који су били део претходног главног пута а нису део новог главног пута чине напуштено подстабло

за корен напуштеног подстабла евидентира се податак о вредности најповољнијег подређеног чвора уместо вредности функције евалуације подређени чворови могу бити ослобођени, чиме се смањује утрошак меморије

на основу помоћног податка могуће је одлучити током претраге да ли је потребно поново развијати напуштено подстабло

Рекурзивна претрага по својству

може се постићи смањење утрошка меморије неопходне за претрагу

начелно није потребно чувати у меморији све формиране чворове у стаблу претраге

чувају се преваходно они чворови који су непосредно повезани с путем који се развија током претраге

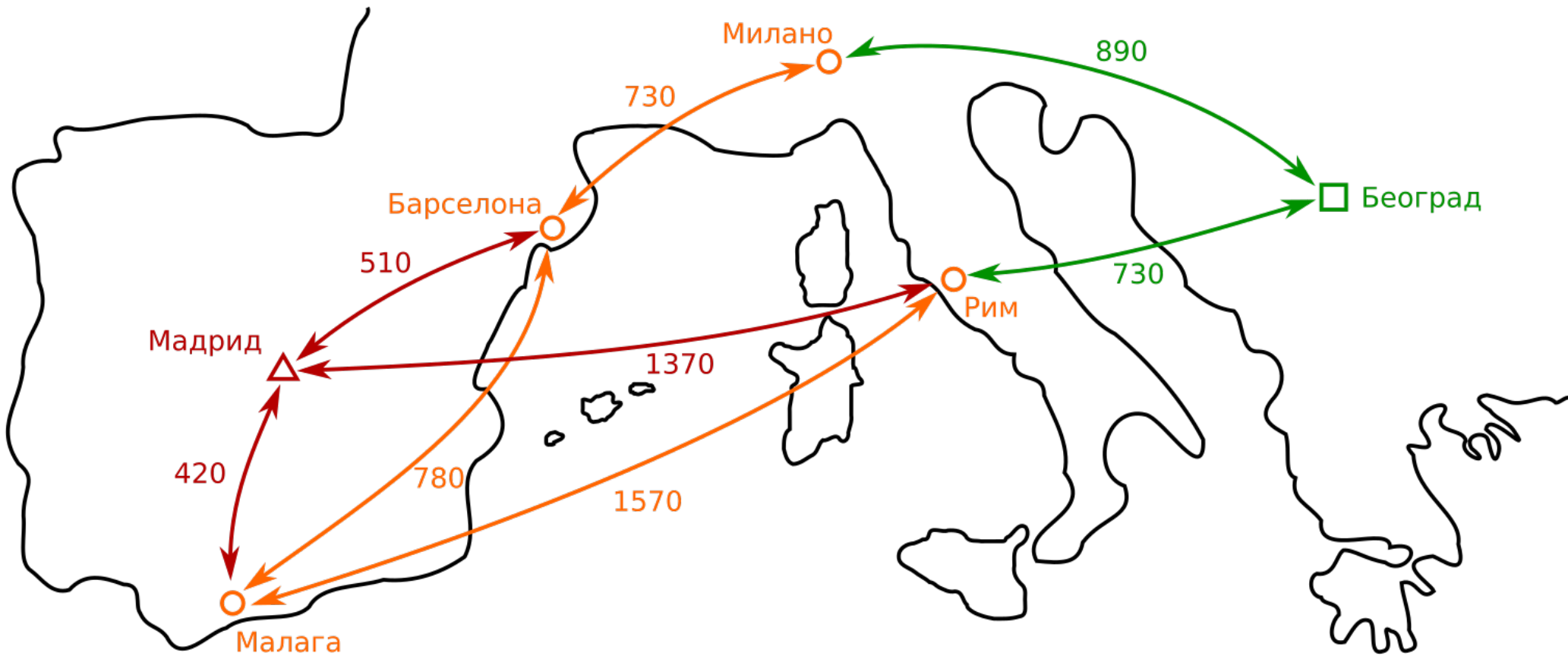
остали чворови не морају бити стално чувани у меморији

услед промена правца претраге може бити неопходно да суштински исти чворови буду поново формирану уз развој новог главног пута претраге

Навођене претраге

Пример задатка – измена

измењено циљно стање

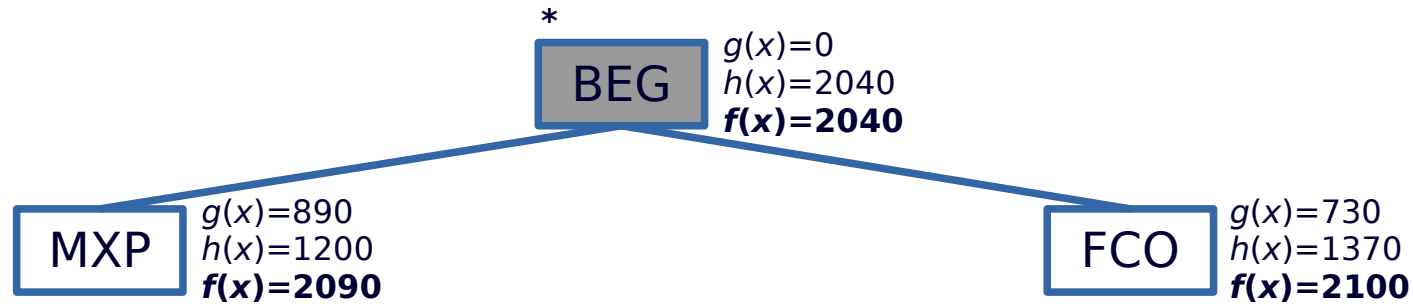


Рекурзивна претрага по својству – пример

BEG	$g(x)=0$ $h(x)=2040$ $f(x)=2040$
-----	--

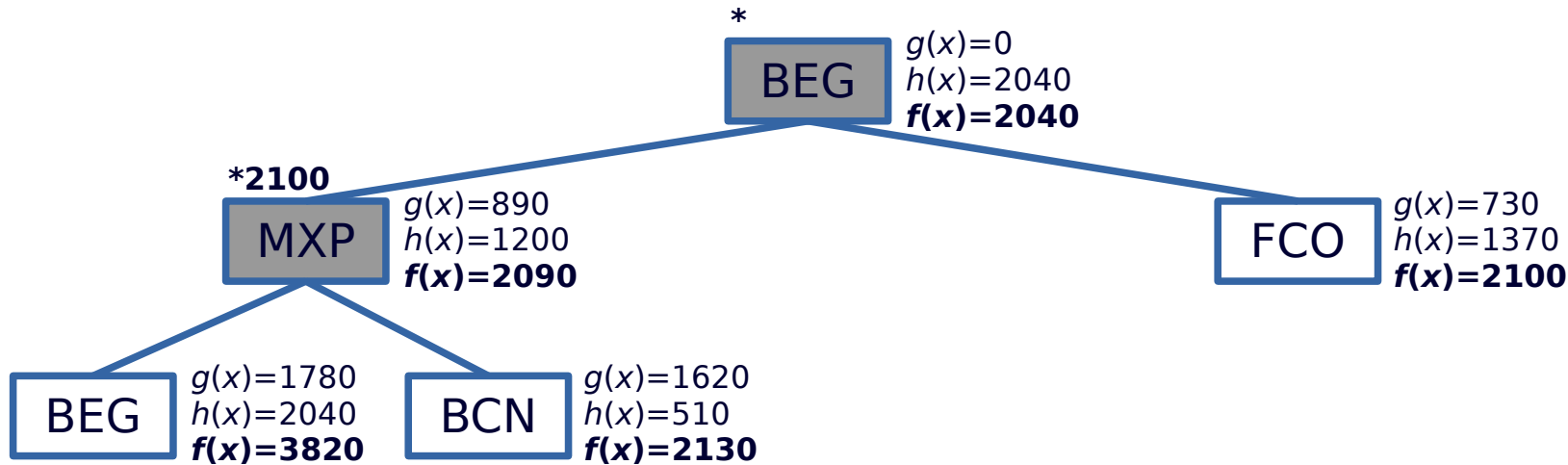
шражење – део 1

Рекурзивна претрага по својству – пример



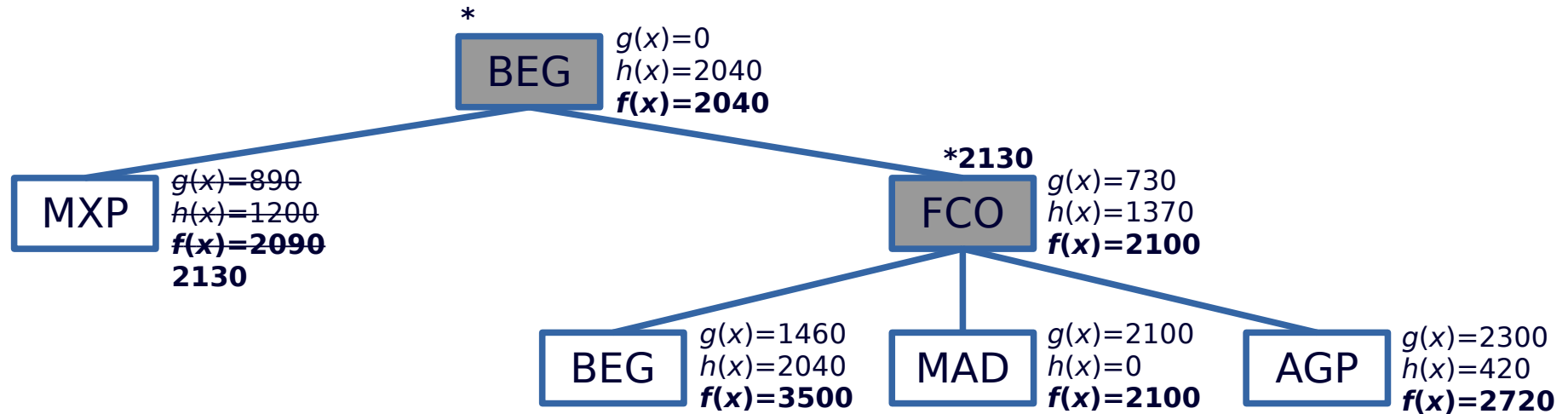
шражење – део 2

Рекурзивна претрага по својству – пример



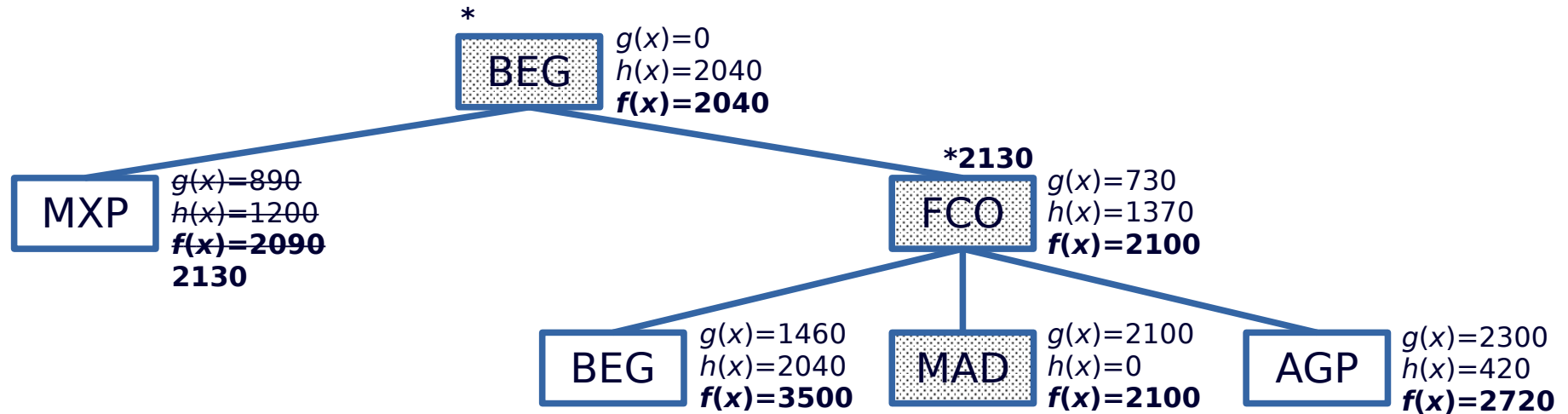
шражење – део 3

Рекурзивна претрага по својству – пример



шражење – гео 4

Рекурзивна претрага по својству – пример



шражење – део 5

1. Увод у претраге
2. Ненавођене претраге
3. Навођене претраге
- 4. Извори и литература**

Основни извори и литература

- ◆ Russel S, Norvig P. Artificial Intelligence: A Modern Approach. 3rd edition. Pearson Education (Upper Saddle River, NJ, USA); 2010.
- ◆ Poole DL, Mackworth AK. Artificial Intelligence: Foundations of Computational Agents. 3rd edition. Cambridge University Press (Cambridge, UK); 2023. Internet: <https://www.artint.info/index.html>

Додатни извори и литература

- ◆ Pearl J. Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley Publishing Company (Reading, MA, USA); 1984.

Основне академске студије
Информациони инжењеринг

Методе и технике науке о подацима

Претраге

(материјали за предавања)