ПРОБЛЕМ ПУТУЈУЋЕГ ТРГОВЦА: КРАТАК ПРЕГЛЕД

Матија Маринковић
Математички факултет универзитета у Београду
Број индекса: 40/2025

26. јануар 2026.

САДРЖАЈ

1 Увод

Проблем путујућег трговца (енг. *Traveling Salesman Problem*, TSP) представља један од најпознатијих проблема у теорији алгоритама, комбинаторној оптимизацији и оперативним истраживањима. Упркос једноставној формулацији, овај проблем показује изузетну рачунарску сложеност и служи као класичан пример разлике између интуитивне једноставности и формалне тежине решавања.

Основна идеја проблема је следећа: дати су градови и растојања између њих, а задатак је пронаћи најкраћу могућу руту која обилази сваки град тачно једном и враћа се у почетни град. Овај проблем има директне примене у логистици, планирању транспорта, оптимизацији дистрибутивних система, као и у биоинформатици и дизајну интегрисаних кола.

Посебна важност TSP-а произилази из чињенице да припада класи NP-тешких проблема, што значи да не постоји познат алгоритам који би га решавао ефикасно за произвољно велики број улазних података. Због тога се TSP често користи као тестни проблем за нове алгоритамске технике.

2 Формална дефиниција проблема

Пре формалне дефиниције проблема путујућег трговца, неопходно је увести основне појмове из теорије графова који се користе у његовој формулацији.

Дефиниција 1. *Граф је уређени пар $G = (V, E)$, где је V непразан коначан скуп чворова (темена), а $E \subseteq V \times V$ скуп ивица које повезују чворове графа.*

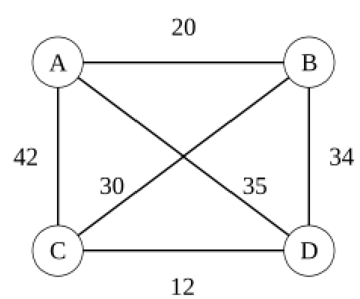
Уколико је свакој ивици придружена нумеричка вредност која представља цену, дужину или тежину, такав граф назива се *тежински граф*. Уколико су свака два различита чвора повезана ивицом, граф се назива *потпун граф*.

Дефиниција 2. *Циклус у графу је путања која почиње и завршава се у истом чвору, при чему се ниједан чвор, осим почетног, не понавља.*

Дефиниција 3. *Хамилтонов циклус је циклус у графу који обилази сваки чвор тачно једном, осим почетног чвора који се појављује и на крају путање.*

Дефиниција 4. *Проблем путујућег трговца је проблем проналажења Хамилтоновог циклуса најмање дужине у потпуном тежинском графу.*

3 Графичка интерпретација проблема



Слика 1: Пример TSP графа

Графичка репрезентација TSP-а значајно доприноси интуитивном разумевању проблема. Чворови графа представљају градове, док ивице означавају могуће путеве између њих.

Иако се на малом броју градова решење може лако визуелно уочити, већ код умереног броја чворова број могућих комбинација постаје изузетно велик. Овај аспект чини TSP погодним за анализу граница људске интуиције и формалних алгоритамских приступа.

Да би се илустровала експоненцијална сложеност проблема, размотримо пример са $n = 4$ града (Слика 1). Приметимо да за овај пример постоји укупно $n! = 4! = 24$ могућих пермутација градова. Међутим, због цикличне природе тура, и чињенице да је сваки град повезан са сваким другим градом, број различитих Хамилтонових циклуса се смањује на $\frac{(n-1)!}{2} = 3$. Ова 3 циклуса су дата у наставку, заједно са својим укупним дужинама:

- $A \to B \to C \to D \to A$, укупна дужина: 97
- $A \to B \to D \to C \to A$, укупна дужина: 108
- $A \to C \to B \to D \to A$, укупна дужина: 141

На основу добијених вредности, најкраћа тура је $A \to B \to C \to D \to A$, са укупном дужином 97, те она представља решење проблема путујућег трговца за ова 4 града.

4 Теоријске особине проблема

Разматрање рачунарске сложености проблема путујућег трговца захтева увођење основних појмова из теорије рачунарске сложености, пре свега класе NPпроблема и појма NP-тежине.

4.1 Класа NP-проблема

Дефиниција 5. *Класа NP обухвата све проблеме одлучивања за које је, за дато потенцијално решење, могуће проверити његову исправност у полиномијалном времену помоћу детерминистичког алгоритма.*

Важно је нагласити да припадност класи NP не значи да се проблем може ефикасно решити, већ само да се исправност решења може брзо проверити. Многи практично веома тешки проблеми припадају управо овој класи.

4.2 NP-тешки проблеми

Дефиниција 6. *Проблем се назива NP-тежак ако се сваки проблем из класе NP може полиномијално свести на њега.*

Интуитивно, NP-тежак проблем је најмање онолико тежак колико и најтежи проблеми у класи NP. Уколико би се пронашао ефикасан алгоритам за решавање било ког NP-тешког проблема, то би имплицирало да су сви NP-проблеми ефикасно решиви.

4.3 Хамилтонов циклус

Лема 1. *Проблем одређивања да ли дати граф садржи Хамилтонов циклус је NP-комплетан проблем.*

Објашњење. Проблем Хамилтоновог циклуса припада класи NP, јер се за дати циклус у полиномијалном времену може проверити да ли обилази сваки чвор тачно једном и да ли је затворен. Истовремено, показано је да је овај проблем NP-тежак редукцијама са других NP-комплетних проблема, што заједно имплицира његову NP-комплетност.

4.4 NP-тежина проблема путујућег трговца

Теорема 1. *Проблем путујућег трговца је NP-тежак.*

Објашњење. NP-тежина проблема путујућег трговца следи из чињенице да се проблем Хамилтоновог циклуса може полиномијално свести на њега. Конкретно, сваком графу се може придружити потпун тежински граф тако да постојање Хамилтоновог циклуса одговара постојању туре чија укупна дужина не прелази задату границу. На тај начин, решавањем проблема путујућег трговца решава се и проблем Хамилтоновог циклуса.

На основу наведене редукције следи да је проблем путујућег трговца NP-тежак, а с обзиром на то да припада и класи NP, често се посматра као један од најрепрезентативнијих проблема у теорији рачунарске сложености.

5 Алгоритамски приступи

Решавање проблема путујућег трговца представља изазов због његове експоненцијалне сложености, па су током времена развијени различити алгоритамски приступи, који се уопштено могу поделити на тачне алгоритме и приближне, односно хеуристичке методе. Преглед најзначајнијих алгоритамских приступа проблему путујућег трговца, заједно са њиховом временском сложеношћу и квалитетом добијених решења, дат је у табели (Табела 1). Из табеле се јасно уочава компромис између тачности решења и рачунарске ефикасности: тачни алгоритми гарантују оптималност, али имају експоненцијалну сложеност, док хеуристички и метахеуристички приступи омогућавају брзо добијање квалитетних приближних решења и због тога су често пожељни у практичним применама.

5.1 Тачни алгоритми

Тачни алгоритми имају за циљ да увек пронађу оптималну туру, односно туру минималне укупне дужине. Њихова главна мана је велика временска сложеност, због чега су практично применљиви само на проблеме мањих димензија.

Најједноставнији тачан приступ је алгоритам бруталне силе, који испробава све могуће пермутације градова и за сваку од њих израчунава дужину одговарајуће туре. Пошто број могућих тура расте факторијелно са бројем градова, овај приступ врло брзо постаје неупотребљив чак и за умерено велике улазе. Његова важност је пре свега теоријска, јер јасно илуструје комбинаторну природу проблема.

Значајно ефикаснији приступ представља динамичко програмирање, познато као Held–Karp алгоритам. Основна идеја овог алгоритма је да се избегне вишеструко рачунање истих подпроблема. Уместо разматрања целокупних тура, проблем се разлаже на мање подпроблеме који представљају најкраћи пут који обилази одређени скуп градова и завршава се у конкретном граду. Комбинацијом ових подрешења добија се оптимална тура. Иако је овај алгоритам знатно бржи од бруталне силе, његова временска сложеност је и даље експоненцијална, што ограничава његову примену на релативно мали број градова.

Још један важан тачан приступ је Branch and Bound техника, која представља систематско претраживање простора решења уз одсецање грана које не могу довести до оптималног решења. Алгоритам гради делимичне туре и за њих израчунава доњу границу могуће дужине потпуне туре. Уколико је та доња граница већа од тренутно најбољег пронађеног решења, та грана се даље не разматра. На овај начин се значајно смањује број тура које је потребно испитати, иако у најгорем случају сложеност остаје експоненцијална.

5.2 Хеуристички и метахеуристички приступи

У практичним применама, где је број градова велики, често није неопходно пронаћи строго оптимално решење, већ довољно добро решење у разумном времену. У таквим ситуацијама користе се хеуристички и метахеуристички алгоритми.

Једна од најпознатијих хеуристика је алгоритам најближег суседа, који гради туру постепено. Полазећи од почетног града, у сваком кораку се бира следећи град који је тренутно најближи и који још није посећен. Иако је овај алгоритам веома брз и једноставан за имплементацију, његова мана је што локално оптималне одлуке не морају довести до глобално оптималног решења.

Метахеуристички приступи, као што су генетски алгоритми и симулирано каљење, настоје да избегну ову слабост. Генетски алгоритми посматрају могуће туре као јединке у популацији, које се током времена еволутивно унапређују применом оператора селекције, укрштања и мутације. На тај начин се постепено добијају све квалитетнија решења.

Симулирано каљење је инспирисано физичким процесом хлађења материјала. Алгоритам почиње са неким почетним решењем и постепено га побољшава, дозвољавајући у одређеним фазама и прелазак на лошија решења, како би се избегло заглављивање у локалном минимуму. Како алгоритам напредује, вероватноћа прихватања лошијих решења се смањује, што доводи до стабилизације решења.

Иако ови приступи не гарантују оптималност, они омогућавају добијање квалитетних приближних решења у времену које је прихватљиво за реалне проблеме великих размера.

Алгоритам	**Временска сложеност**	**Квалитет решења**
Брутална сила	$O(n!)$	Оптимално
Held–Karp	$O(n^2 2^n)$	Оптимално
Branch and Bound	Експоненцијална	Оптимално
Алгоритам најближег суседа	$O(n^2)$	Приближно
2-opt локална претрага	Полиномијална	Добро приближно
Генетски алгоритми	Полиномијална по генерацији	Врло добро приближно
Симулирано каљење	Полиномијална	Врло добро приближно

Табела 1: Поређење алгоритамских приступа за проблем путујућег трговца

6 Закључак

Проблем путујућег трговца представља један од најзначајнијих теоријских проблема у савременом рачунарству. Његова сложеност јасно илуструје ограничења класичних алгоритамских приступа и потребу за хеуристичким методама.

Иако је проналажење оптималног решења често неизводљиво, практичне примене показују да приближни алгоритми могу дати веома квалитетна решења у разумном времену. Због тога TSP остаје активна област истраживања и важан концепт у образовању рачунарских наука.