

РЕШАВАНЕ НА ПРОБЛЕМА НА ТРАНСПОРТНИЯ ПЪТНИК (TSP) В ГРАДСКА ЗОНА ЧРЕЗ УЕБ-БАЗИРАНО ПРИЛОЖЕНИЕ ЗА ПЛАНИРАНЕ НА МНОГОЦЕЛЕВО ПЪТУВАНЕ

Теодор Беров
tberov@vtu.bg,

**ВТУ"Тодор Каблешков",
ул."Гео Милев" № 158, София 1574
БЪЛГАРИЯ**

***Ключови думи:** транспорт, автомобилен, градски, проблем на транспортния пътник, Google maps*

***Резюме:** В този материал е разгледан Проблема на транспортния пътник за градска зона и начина за решаването му чрез използване на реални разстояния и времепътуване между обектите. Предложено е използване на сайт за планиране на многоцелево пътуване, използващ приложението Google Maps API. За два сайта (Route XL и OptiMap) са посочени последователността на операциите за въвеждане и решение на проблема, като е използван практически пример за гр. София, и са показани получените резултати.*

1. ВЪВЕДЕНИЕ

Проблемът със задачата на търговския пътник може да се опише като: Търговски пътник трябва да обиколи фиксиран брой обекти, тръгвайки от един – начален, преминаващ през други обекти само един път и връщане отново в началния обект. Необходимо е да се избере от всички възможни маршрути за преминаване през всички обекти, този с най-малки разходи (пробег, времепътуване). Класическата формулировка на задачата е базирана на координати на обектите и съответно „Евклидово“ разстояние между всеки два от тях.

По-реалистичен подход за приложения за градската логистика би било да се получи изминатото разстояние чрез прилагане на алгоритъм за най-краткия път към компютъризирания модел на системата на пътната мрежа, а именно, когато времето за пътуване са реални данни вместо разстоянията. Транслирането на цифрова карта на пътната мрежа в представяне на насочен граф, по отношение на връзките и възлите, не е лесна процедура. Допускането за симетричност на разходите не е валидно. Това се дължи на факта, че в условията на градски среда, при използване на улиците трябва да държат сметка за еднопосочни улици, особеностите на кръстовищата - сигнализирани или несигнализирани, забранени завой при движение и / или обратните завои и т.н.

Трябва да се вземе в предвид движението в завой, в съответствие с реалната мрежа, с техните свързани разходи като резултат от времевите настройки на системите за контрол на трафика на сигнализирани кръстовища, или забавяне на кръстовища, регулирани от предимство или сигнали „Stop“.

В настоящия момент имаме наличието на много технологии за Географски информационни системи (ГИС), т.е. дигитализирани карти: *Google Maps*, *Google Earth*, *Bing Maps*, *Here* и др. При тях се взема под внимание динамично променящите се параметри, като например задръстванията, така че информация за трафика в реално време, събрана чрез ГИС се използват ефективно за постигане на оптимални маршрути в рамките на стохастичната транспортна мрежа. Прилагането им за разглеждания проблем е изключително удачно решение.

В интернет пространството могат да бъдат открити безплатни или платени сайтове, които имат възможност да решават задачата на търговския пътник само чрез въвеждане на желаните адреси за спирка. Някои от тези сайтове са:

Route XL [4] е сайт за планиране на многоцелево пътуване. С помощта на този сайт е възможно да се изчисли най-добрият маршрут по няколко дестинации и най-оптималната последователност. Интерфейсът използва Google Maps API. След като бъде съставен маршрут, потребителят има възможност да го изтегли във файл или да го отвори в Google Maps, от където да се навигира завой по завой. Сайтът е безплатен за маршрути до 20 спирки, но се предлага и платена версия, където могат да се оптимизират до 150 спирания на маршрут.

OptiMap [3] е още един сайт, който предлага изчисляване на маршрут до няколко дестинации. Сайтът е изцяло безплатен и използва Google Maps API. Не е нужно потребителят да се регистрира за да използва функциите му. Маршрутът може да бъде импортиран от текстов файл или да бъде въведен ръчно направо в сайта. Готовият маршрут може да се експортира до Google maps.

2. ОПИСАНИЕ НА ПРОБЛЕМА

Задача за търговския пътник. Да представим градската пътна мрежа във вида на ориентиран свързан граф $G=(V,E)$, съответстващ на *TSP*, където $V=\{1,2,..n_0\}$ – множеството върхове в гафа, E – множеството на дъгите, съединяващи върховете. Връх 0 съответства на „начален“ и връх i съответства на клиент i , за $1 \leq i \leq n$. Всички осъществими маршрути за обхождане, съответстват на пътища в G , които започват от θ и завършва на θ .

Тогава в общия случай математическата постановка може да се формулира по следния начин:

Целева функция:

$$(1) \quad L = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min_{x \in \Delta\beta}$$

Където $\Delta\beta$ е множество от допустими алтернативи, формиращи се от система ограничения от тип равенства и неравенства ($\sum_{i=1}^n x_{ij} = 1, j = 1, \dots, n$; $\sum_{j=1}^n x_{ij} = 1, i = 1, \dots, n$). Транспортните разходи c_{ij} между два върха (i,j) могат да бъдат разстояние, времепътуване или приведени в съответна стойност за превозно средство.

Разглежданата задача има два основни варианта – „затворен“, когато началният и краен пункт са едни и същи; „отворен“, началният и краен пункт са различни.

Методи за решаване на TSP.

Чрез прилагане на Google Maps API (*интерфейс за приложно програмиране*) [2].

Google Maps е уеб-базирано приложение, което позволява на потребителите да разглеждат карти на дадена уеб страница и има три основни функции, които предлага:
А. Потребителите могат да търсят по местоположение (т.е. за обекти в рамките на текущата им видима карта).

Б. Потребителите могат да питат за посоката, в мулти модална транспортна система (т.е. шофиране, обществен транспорт и др.)

В. Потребителите могат да добавят свои собствени данни за картата, но само чрез **API** за Google Maps.

За достъп до ресурсите на технологията, производителя е разработил Google Maps API, т.е. в компютърното програмиране, *интерфейс за приложно програмиране* уточнява как някои софтуерни компоненти, следва да си взаимодействат един с друг.

Разработчика е имплементирал два основни модела, приложими за разглеждания проблем: Намиране на най-краткия път от точка до точка по картата; Подредждане на точки според **TSP**.

Освен най-краткия път, разработчика е предвидил и допълнително още два конкурентни маршрута, ако това е възможно. Основната технология на използване е както следва:

Използван алгоритъм:

1. Инициализиране на **URL** (Стандартизиран Локатор на Ресурси);
2. Отпраща искане (заявка) към **URI** (Стандартизиран Идентификатор на Ресурси), т.е. **URL- WebRequest**;
3. Ако има отговор – Обработка на отговора;
4. Изтегляне на необходимата информация от отговора;

Предвидено е отговора да се получава в два формата: **json** – JavaScript Object Notation (JSON); **xml** - Extensible Markup Language (XML) текстови формат, получен от Standard Generalized Markup Language (SGML).

Използван URL:

- за намиране на пътя от точка до точка–

<http://maps.googleapis.com/maps/api/directions/xml?origin=obect0&destination=obect1&sensor=false&alternatives=true;>

- за TSP -

http://maps.googleapis.com/maps/api/directions/json?origin=obect0&destination=obect0&waypoints=optimize:true|obect1|obect2|obect3|obect4&key=API_KEY

Придържайки се към изискванията на използваната технология, и задачите

които ще решаваме, формирането на заявката става по параметрите:

Използваните полета са както следва:

- **origin** (*задължително*) – началната точка от която да се определи пътя;

- **destination** (*задължително*) – крайната точка;

- **travelMode** (*препоръчително*) – начина на предвижване. В случая е „шофиране“, което е зададено по подразбиране;

- **durationInTraffic** (*препоръчително*) – времепътуване в зависимост от трафика. Тази опция е изключително полезна за разглежданите задачи, но е достъпна само за „work“ потребители.;

- **waypoints[]** (*препоръчително*) – списък със задължителни точки на преминаване;

<p><i>origin: LatLng String,</i> <i>destination: LatLng String,</i> <i>travelMode: TravelMode,</i> <i>transitOptions: TransitOptions,</i> <i>unitSystem: UnitSystem,</i> <i>durationInTraffic: Boolean,</i> <i>waypoints[]:</i> <i>DirectionsWaypoint,</i> <i>optimizeWaypoints: Boolean,</i> <i>provideRouteAlternatives: Bool,</i> <i>avoidHighways: Boolean,</i> <i>avoidTolls: Boolean</i> <i>region: String</i></p>
--

- **provideRouteAlternatives** (*препоръчително*) – предоставяне на алтернативни пътища. Използва се за получаване на пътища, различни от най-късия.

3. ПРИЛОЖЕНИЕ НА МЕТОДИКАТА

Транспортната дейност на куриерските фирми се състои в обслужване на обекти, като проблема за последователността на посещенията им се решава чрез TSP.

Да разгледаме транспортната дейност на куриерска фирма „X“.

Централният офис на фирмата, от където се вземат пратките за разнос се намира в централната част на гр. София. Пратките, които са за разнос в централната част на града, се дават на куриери, които осъществяват разнасянето им. Тези служители си правят първоначален план за дейността и след като се уточнят всички подробности по маршрута, куриерите взимат пратките и започват да извършват разносната дейност.

Куриерът обслужва един обект около - 4 минути, но в някои случаи може обслужването да става и по – бързо (до 2 минути) а при по-натоварени обекти обслужването може и да бъде до 10 минути. При използване на ведомствен автомобилен транспорт куриерът може да обслужи от 20-30 пратки на ден, изминава средно по 100 км. на ден. Много често се налага на куриерите да използват обходни маршрути, поради голямото натоварване на някои булеварди.

За градски куриерски услуги и разглеждания проблем ще се разгледат Route XL и OptiMap, които използват Google Maps API.

Алгоритъм за оптимизиране на последователността за обхождане на обекти чрез използване на **Route XL**.

1. Чрез интернет браузър, по избор на потребителя, в адрес лентата се въвежда <http://www.routexl.com/>;

2. Избор на бутонът IMPORT (въведи), от където потребителя има възможност да въвежда нужните му адреси.

3. Необходимите адреси се въвеждат ръчно или директно се внасят от текстов документ или таблица и се натиска бутонът Import. Сайтът нанася обектите върху картата.

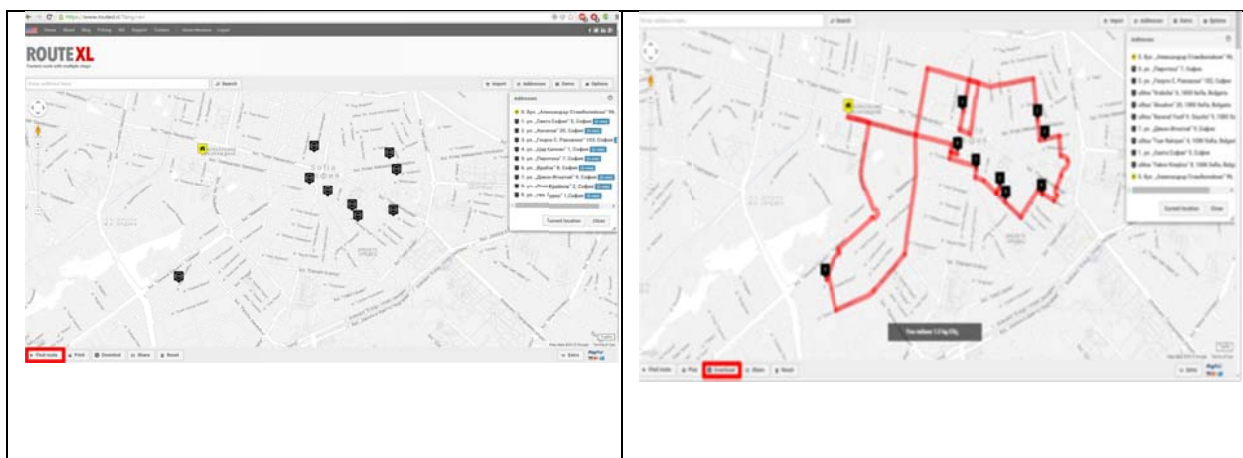
4. Когато всичко обекти са нанесени се натиска бутонът Find route (намери път)

5. Изчислява се най-оптималният път

6. Потребителят има възможност да запази маршрута чрез бутона Download (сваляне)

7. На потребителя е предоставена възможност да запази маршрута в няколко формата или да го отвори през Google maps.

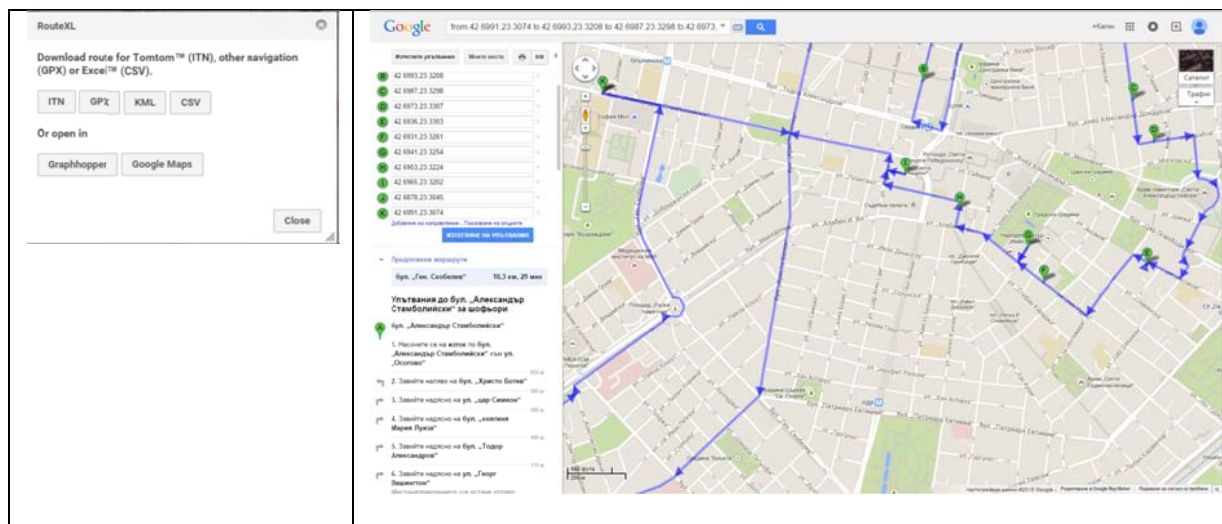
8. Отваряне предложеният маршрут в Google maps.



фиг.1 Последователност за обхождане на обектите в среда **Route XL**

Алгоритъм за оптимизиране на последователността за обхождане на обекти чрез използване на **OptiMap** при градски куриерски услуги:

1. Чрез интернет браузър, по избор на потребителя, в адрес лентата се въвежда <http://www.gebweb.net/optimar/>;
2. Чрез падащото меню Destinations потребителят има възможност да въведе желаните дестинации
3. Чрез бутона Add list of locations адресите се нанасят върху картата:
4. Бутонът Calculate Fastest Roundtrip изчислява най-оптималната последователност за обслужване на адресите.
5. Експортиране маршрута в Google maps



фиг.2 Експортиран маршрут в Google maps

4. РЕЗУЛТАТИ

След калкулиране на оптималния обходен маршрут двете приложения предложиха идентични данни, като приблизителна дължина на маршрута и приблизително време за изминаването му. Единствената разлика е в последователността на обектите. Във вариантът от Route XL обект 9 е поставен последен преди прибирането в централният офис („база“). Във вариантът на OptiMap обект 9 е поставен на втора позиция. Последователността на останалите обекти е запазена. И двете приложения са в състояние да предоставят достоверна информация относно времепътванията по зададени обекти.

таблица 1: Характеристика на маршрута

от обект	до обект	разстояние, м	времпът., мин	паркиране/за- минав., мин	време за обсл., мин
база	обект 1	2300	5	4	4
обект 1	обект 2	1200	3	3	5
обект 2	обект 3	220	1	5	7
обект 3	обект 4	850	3	4	4
обект 4	обект 5	700	2	3	6
обект 5	обект 6	200	2	-	8
обект 6	обект 7	850	3	4	8
обект 7	обект 8	450	2	3	3
обект 8	обект 9	2500	5	2	5
обект 9	база	1600	4	2	
	общо	10870	30	30	50

5. ЗАКЛЮЧЕНИЕ

В този материал е разгледан проблема TSP за градска зона и начина за решаването му чрез използване на реални разстояния и време пътуване между обектите. Предлага се използване на сайт за планиране на многоцелево пътуване, използващ приложението Google Maps API. За два сайта (Route XL и OptiMap) са посочени последователността на операциите за въвеждане и решение на проблема, като е използван практически пример за гр. София и са показани получените резултати.



фиг.3 Разпределение на времената за маршрута

ЛИТЕРАТУРА:

[1] Карагъзов К., Беров Т., „Приложение на империалистически конкурентен алгоритъм за оптимизиране маршрутизацията на превозните средства“, Механика, транспорт, комуникации, ISSN 1312-3823, том 12, бр. 3(1), 2014, стр. 63-68

[2] Google Maps Javascript API v3: [сайт]. URL: <https://developers.google.com/maps/documentation/javascript/?hl=bg>

[3] <http://www.gebweb.net/optimap/>

[4] <http://www.routexl.com>

SOLVING THE TRAVELLING SALESMAN PROBLEM IN URBAN AREA WITH THE SITE FOR ROAD ROUTE PLANNER FOR MULTIPLE LOCATIONS

Teodor Berov
tberov@vtu.bg,

*Todor Kableshkov University of Transport,
158 Geo Milev Street., Sofia 1574,
BULGARIA*

Key words: urban transport, freight, traveling salesman problem, Google maps

Abstract: Traveling salesman problem in urban areas and approach to solve it through the use of real distances and travel time between the points is presented in this article. The use of the site for road route planner for multiple locations, using the Google Maps API, is proposed. Algorithms for introduction and solution the TSP for two sites (Route XL and OptiMap) are referred. Through this approach is solved a practical example for city Sofia and the obtained results are shown.