

2012•2013  
FACULTEIT BEDRIJFSECONOMISCHE WETENSCHAPPEN  
*master in de toegepaste economische wetenschappen:  
handelsingenieur: accountancy en financiering*

## Masterproef

Het plannen van routes voor toeristen aan de hand van meerdere criteria

Promotor :  
Prof. dr. An CARIS

Copromotor :  
dr. Katrien RAMAEKERS

## Ine Umans

*Masterproef voorgedragen tot het bekomen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur, afstudeerrichting accountancy en financiering*

2012•2013

FACULTEIT BEDRIJFSECONOMISCHE  
WETENSCHAPPEN

*master in de toegepaste economische wetenschappen:  
handelsingenieur: accountancy en financiering*

## Masterproef

Het plannen van routes voor toeristen aan de hand van  
meerdere criteria

Promotor :  
Prof. dr. An CARIS

Copromotor :  
dr. Katrien RAMAEKERS

Ine Umans

*Masterproef voorgedragen tot het bekomen van de graad van master in de toegepaste  
economische wetenschappen: handelsingenieur, afstudeerrichting accountancy en  
financiering*



## **Woord vooraf**

Deze eindverhandeling vormt het sluitstuk van de gevolgde studie handelsingenieur. Bij het schrijven van deze eindverhandeling, heb ik mij kunnen verdiepen in een onderwerp dat berust op persoonlijke ervaringen. Mijn interesse voor het onderwerp alsook de drijfveer om bij te leren waren twee peilers die voor de nodige inzet zorgden. De eindverhandeling zou echter niet tot stand gekomen zijn op basis van deze inzet alleen. Steun, begeleiding en inzicht van anderen waren belangrijke elementen die mede het eindresultaat bepaald hebben.

In het bijzonder wil ik promotor Prof. dr. An Caris bedanken voor haar deskundige begeleiding, advies, kritiek en haar antwoorden op vele vragen. Mijn dank gaat tevens uit naar copromotor dr. Katrien Ramaekers voor de kritische feedback tijdens het masterproefseminarie. Mijn ouders en vrienden wil ik graag bedanken voor de steun en motivatie tijdens het uitvoeren van mijn studies. Tot slot gaat mijn dank uit naar de toeristische dienst van de stad Hasselt voor het verschaffen van de nodige informatie.



## Samenvatting

Een toerist die bepaalde bezienswaardigheden wil bezoeken in een bepaalde stad of regio in een aantal dagen, krijgt te maken met een complex probleem. De toerist moet een selectie maken van bezienswaardigheden die hij zal bezoeken aangezien het onmogelijk is om alle *point of interest* (POI) te bezoeken binnen zijn tijdsbudget. Daarna moet de toerist de dag alsook het tijdstip van bezoek bepalen voor elk POI dat hij wil bezoeken. Vervolgens moet een route worden uitgestippeld tussen de gekozen POI. De toerist moet informatie vanuit verschillende invalshoeken combineren waardoor het plannen van een reis een moeilijke beslissing wordt. Zowel reisbureaus als ontwikkelde aanbevelingsystemen kunnen de toerist helpen in het plannen van zijn reis. Verscheidene problemen kunnen echter opduiken wanneer de toerist zijn geplande route uitvoert. Verkregen informatie over een bepaald POI kan foutief zijn, zoals een foutief aangebrachte sluitingsdag, waardoor de route onverwacht wijzigt. Tijdelijke informatie wordt vaak niet weergegeven in bronnen waaruit de toerist zijn informatie vergaart (Souffriau et al., 2008).

De *personalised electronic tourist guide* (PET) is een applicatie die de toerist kan helpen bij het plannen van een reis. Het achterliggende probleem wordt geclassificeerd als een *tourist trip design* probleem (TTDP). Het *orienteeing* probleem (OP) wordt gebruikt om de meest eenvoudige versie van het TTDP te modelleren. Elke bezienswaardigheid krijgt een score gebaseerd op de persoonlijke voorkeur van de toerist en de toerist bepaalt zijn tijdsbudget. Het doel is één route te vinden voor een enkele dag waarvoor de totale score gemaximaliseerd wordt en bijgevolg in de meest mogelijke mate voldaan wordt aan de behoeften van de toerist. Het *team orienteeing* probleem (TOP) breidt het OP uit door meerdere routes in overweging te nemen. Aan dit probleem kunnen tijdvensters worden toegevoegd waardoor de mogelijkheid ontstaat rekening te houden met de openings- en sluitingstijden van de POI. Dit is het *team orienteeing* probleem met *time windows* (TOPTW). Een complexere versie van het TTDP is het *multi-constraint team orienteeing* probleem met *multiple time windows* (MCTOPMTW) dat rekening houdt met meerdere gebruikersbependingen zoals het beschikbaar budget. Dit probleem laat tevens meerdere tijdvensters per POI toe (Gavalas et al., 2012).

Aangezien de toerist snel over een oplossing voor zijn probleem wil beschikken, zijn efficiënte algoritmen nodig die in realtime een oplossing voor het TTDP kunnen aanbieden. In de literatuur worden daarom vooral metaheuristieken aangehaald als mogelijke oplossingsmethoden voor het probleem (Vansteenwegen et al., 2011c). Castro et al. (2013), Vansteenwegen et al. (2011a) en Zhu et al. (2010) vermelden de optie om rekening te houden met de selectie van een hotel bij het plannen van een reis.

De praktijkstudie start met een korte uitleg over bestaande web- en mobiele applicaties. De stad Hasselt vormt het uitgangspunt voor de praktijkstudie. Eerst worden enkele bezienswaardigheden van de stad Hasselt, die geselecteerd zijn voor verder onderzoek, toegelicht. De nodige gegevens van deze bezienswaardigheden worden verzameld. Gecombineerd met de nodige gegevens van een toerist wordt een TTDP geformuleerd. Aangezien verschillende auteurs gebruik maken van een *tabu search* algoritme om dit probleem op te lossen, wordt een vergelijkbaar algoritme ontwikkeld en toegepast op het fictieve voorbeeld. Het algoritme bestaat uit een initialisatie waarna een iteratie bestaande uit vier stappen, namelijk de invoegstap, de vervangstap, de ruilstap en de verwijderstap, volgt. Het stopcriterium zorgt ervoor dat het algoritme beëindigd wordt. Het doel is een route uit te stippelen voor de toerist zonder enige beperking te schenden waarbij de voldoening van de toerist gemaximaliseerd wordt.

## Inhoudsopgave

Woord vooraf .....	I
Samenvatting .....	III
Inhoudsopgave .....	V
Lijst van figuren .....	VII
Lijst van tabellen .....	IX
Hoofdstuk 1 Probleemstelling .....	1
1.1 Praktijkprobleem .....	1
1.2 Centrale onderzoeksvraag .....	2
1.3 Deelvragen .....	3
1.4 Onderzoeksopzet .....	3
Hoofdstuk 2 Literatuurstudie .....	5
2.1 Inleiding .....	5
2.2 Oriëntering probleem .....	6
2.2.1 Inleiding .....	6
2.2.2 Mathematische formulering .....	8
2.2.3 Praktische toepassingen .....	9
2.2.4 Oplossingsmethoden .....	10
2.2.4.1 Exacte algoritmen .....	10
2.2.4.2 Heuristische oplossingsmethoden .....	10
2.3 Team oriëntering probleem met tijdvensters .....	13
2.3.1 Inleiding .....	13
2.3.2 Mathematische formulering .....	13
2.3.3 Oplossingsmethoden .....	14
2.4 Team oriëntering probleem met meerdere tijdvensters en meerdere beperkingen .	18
2.4.1 Inleiding .....	18
2.4.2 Mathematische formulering .....	18
2.4.3 Oplossingsmethoden .....	19
2.5 Varianten .....	21
2.6 Hotelselectie .....	23
2.7 Conclusie .....	24



Hoofdstuk 3	Praktijkstudie .....	25
3.1	Inleiding .....	25
3.2	POI databank Hasselt .....	27
3.3	Persoonlijke data .....	30
3.4	Interessescore.....	32
3.5	Tourist Trip Design algoritme .....	35
3.5.1	Mathematische formulering .....	35
3.5.2	Algoritme .....	36
3.5.2.1	Initialisatie.....	37
3.5.2.2	Invoegstap .....	39
3.5.2.3	Vervangstap .....	45
3.5.2.4	Ruilstap.....	48
3.5.2.5	Verwijderstap.....	50
3.5.2.6	Stopcriterium .....	52
3.6	Voorstel persoonlijke route .....	57
Hoofdstuk 4	Kritische bedenkingen en aanbevelingen voor verder onderzoek.....	59
Hoofdstuk 5	Algemene conclusies .....	63
Lijst der geraadpleegde werken	.....	65
Bijlagen	.....	71

## Lijst van figuren

Figuur 1: Overzicht problemen (gebaseerd op Gavalas et al., 2012) .....	6
Figuur 2: Tekortkoming ILS (1) .....	17
Figuur 3: Tekortkoming ILS (2) .....	17
Figuur 4: Overzicht expertsysteem (Vansteenwegen et al., 2011b) .....	27
Figuur 5: Structuur algoritme .....	37
Figuur 6: Ruilstap .....	48
Figuur 7: Iteratie 1 - Ruilstap 1 met $6 = i$ en $12 = j+1$ (illustratie).....	48
Figuur 8: Iteratie 1 - Ruilstap 1 met $6 = i$ en $11 = j+1$ .....	49
Figuur 9: Iteratie 1 - Ruilstap 2 met $10 = i$ en $11 = j+1$ .....	50
Figuur 10: Iteratie 2 - Ruilstap 1 met $6 = i$ en $4 = j+1$ .....	56
Figuur 11: Voorstel route 1 .....	57
Figuur 12: Voorstel route 2 .....	58



## Lijst van tabellen

Tabel 1: Samenvattende tabel oriëntering problemen .....	8
Tabel 2: Symbolenlijst oriëntering probleem .....	9
Tabel 3: Symbolenlijst TOPTW .....	14
Tabel 4: Oplossingsmethoden OPTW en varianten .....	15
Tabel 5: Symbolenlijst MCTOPMTW .....	19
Tabel 6: Varianten TTDP .....	21
Tabel 7: POI gegevens .....	29
Tabel 8: Score POI per categorie .....	30
Tabel 9: Reisbeperkingen toerist.....	31
Tabel 10: Interesse toerist in types en categorieën .....	31
Tabel 11: Relatie POI en kernwoorden.....	33
Tabel 12: Berekening totaalscore Modemuseum Hasselt .....	34
Tabel 13: Berekening totaalscore Virga Jessebasiliek.....	34
Tabel 14: Gegevens knooppunten praktijkvoorbeeld.....	35
Tabel 15: Reistijd tussen twee knooppunten.....	35
Tabel 16: Initialisatie.....	39
Tabel 17: Initiële route 1 .....	41
Tabel 18: Iteratie 1 – Route 1 – Invoegstap 1: berekeningen .....	42
Tabel 19: Initiële route 2 .....	43
Tabel 20: Iteratie 1 – Route 2 – Invoegstap 1: berekeningen .....	44
Tabel 21: Iteratie 1 – Route 2 .....	44
Tabel 22: Iteratie 1 – Route 2 – Invoegstap 2: berekeningen .....	45
Tabel 23: Iteratie 1 – Route 1 – Vervangstap 1 – Knooppunt 6.....	46
Tabel 24: Iteratie 1 – Route 1 – Vervangstap 1 .....	46
Tabel 25: Iteratie 1 – Route 1 – Vervangstap 2 – Knooppunt 8.....	47
Tabel 26: Iteratie 1 – Route 1 – Vervangstap 2 .....	47
Tabel 27: Iteratie 1 – Route 1 – Vervangstap 3 – Knooppunt 10.....	47
Tabel 28: Samenvattende tabel tot en met stap 5.....	47
Tabel 29: Iteratie 1 – Route 1 – Vervangstap 3 .....	48
Tabel 30: Iteratie 1 - Ruilstap 1 (illustratie): berekeningen .....	49
Tabel 31: Iteratie 1 – Route 1 – Ruilstap 1 (illustratie) .....	49
Tabel 32: Iteratie 1 – Route 2 – Ruilstap 1 (illustratie) .....	49
Tabel 33: Iteratie 1 – Route 1 – Ruilstap 1.....	50
Tabel 34: Iteratie 1 – Verwijderstap 1: berekeningen .....	51
Tabel 35: Iteratie 1 – Route 1 – Verwijderstap 1.....	51
Tabel 36: Iteratie 1 – Route 2 – Verwijderstap 1.....	51
Tabel 37: Samenvattende tabel stap 5 tot en met stap 10 .....	52
Tabel 38: Iteratie 2 – Route 1 – Invoegstap 1: berekeningen .....	53
Tabel 39: Iteratie 2 – Route 1 – Invoegstap 1 .....	53

Tabel 40: Iteratie 2 – Route 2 – Invoegstap 1: berekeningen .....	53
Tabel 41: Iteratie 2 – Route 2 – Invoegstap 1 .....	54
Tabel 42: Iteratie 2 – Route 1 – Invoegstap 2 .....	54
Tabel 43: Iteratie 2 – Route 2 – Invoegstap 2 .....	54
Tabel 44: Iteratie 2 – Route 2 – Invoegstap 3 .....	55
Tabel 45: Iteratie 2 – Route 2 – Vervangstap 1: berekeningen .....	55
Tabel 46: Iteratie 2 – Route 1 – Ruilstap 1 .....	56
Tabel 47: Iteratie 2 – Route 2 – Ruilstap 1 .....	56
Tabel 48: Samenvattende tabel stap 9 tot en met stap 15 .....	56
Tabel 49: Alternatieve route 1 .....	59
Tabel 50: Alternatieve route 2 .....	59

## Hoofdstuk 1      Probleemstelling

In dit hoofdstuk wordt eerst het praktijkprobleem uitgelegd. Vervolgens wordt de centrale onderzoeksvraag omschreven die resulteert in verschillende deelvragen. Het hoofdstuk sluit af met de onderzoeksopzet dat dient als uitgangspunt voor de praktijkstudie.

### 1.1 Praktijkprobleem

Volgens de *World Tourism Organization* (2012) is de toerisme-industrie de laatste jaren sterk gegroeid. Toerisme wordt aanzien als één van de snelst groeiende economische sectoren in de wereld, daar steeds meer bestemmingen worden toegevoegd. Het bedrijfsvolume is gelijk aan of zelfs hoger dan dat van de olie-export of de auto-industrie. De toerisme-industrie is één van de grote spelers in de internationale handel en betekent een groot inkomen voor veel ontwikkelingslanden. Volgens de Europese Commissie (2012) omvat toerisme een brede waaier aan producten en bestemmingen waarbij veel belanghebbenden betrokken zijn, zowel openbaar als privé. Bovendien kan toerisme bijdragen tot de realisatie van een aantal belangrijke EU-doelstellingen, zoals duurzame ontwikkeling, economische groei, werkgelegenheid en economische en sociale cohesie. De toerisme-industrie omvat transport, accommodatie, catering, ontspanningsmogelijkheden en andere diensten aangeboden aan toeristen. Door verbeteringen in informatie- en communicatietechnologie zijn zowel de efficiëntie als de effectiviteit van toerismebedrijven gegroeid (Buhalis, 2003).

De meeste toeristische websites geven informatie over bestemmingen, aanbevelingen voor reispakketten en de mogelijkheid tot online reservering van vlucht, hotel en/of auto. Vaak zijn forums beschikbaar waarop reizigers meningen en tips kunnen geven die nuttig zijn voor andere reizigers (Zhu et al., 2010). Door de opkomst van het internet en deze websites is een nieuw soort gebruiker ontstaan. De toerist wordt zijn eigen reisagent en kiest zelf zijn reispakket bestaande uit vervoerskeuze, hotelkeuze en dergelijke meer. Wanneer de toerist zijn reis plant, spelen zowel persoonlijke factoren als reisfactoren mee. Persoonlijke factoren kunnen ingedeeld worden in enerzijds socio-economische factoren zoals leeftijd, opleiding en inkomen en anderzijds psychologische en cognitieve factoren zoals ervaring, betrokkenheid en persoonlijkheid. De reisfactoren betreffen het reisdoel, aantal reizigers, de reisduur, de afstand die te overbruggen valt en de wijze van transport (Hannes, 2003).

Een groot aandeel van de bevolking bezoekt jaarlijks een grote stad om gedurende een bepaalde periode verschillende locaties in die stad, ook *points of interest* (POI) genoemd, te bezoeken. Enkele voorbeelden van POI zijn musea, restaurants, kerken, kastelen en parken. Een toerist kan nooit alle POI in een stad bezoeken aangezien in de meeste steden honderden POI aanwezig zijn (Schilde et al., 2009). Hij zal de meest interessante POI selecteren. De persoonlijke interesse in de verschillende POI zal gebaseerd zijn op informatie gevonden op het web, in artikels in tijdschriften of in reisgidsen. Vaak wil een toerist enkel attracties bezoeken die de meeste aandacht krijgen in een reisgids of waar hij ooit over

gehoord heeft. Bijgevolg worden veel toeristen naar dezelfde plaatsen gelokt. Wanneer een toerist voor de eerste keer een bepaalde stad bezoekt, wil hij zoveel mogelijk bezoeken binnen zijn tijdsbudget. Indien een tweede bezoek volgt, beschikt hij over meer informatie en kan hij zorgvuldiger bezienswaardigheden selecteren op basis van zijn interesse, ervaring, doelstelling en voorbereiding. De toerist zal een route uitstippelen om de gekozen bezienswaardigheden te bezoeken rekening houdend met een aantal beperkingen zoals beschikbare tijd, budget en openingsuren (Kramer et al., 2006).

Bij het plannen van een route tussen de verschillende bezienswaardigheden treden moeilijkheden op. Ten eerste is de gevonden informatie niet altijd voldoende recent. Het kan bijvoorbeeld zijn dat een wijziging in de openingsuren van een bepaald POI niet kenbaar gemaakt is op de desbetreffende website. Bij aankomst blijkt echter dat de grot dinsdag gesloten is. Ten tweede geven reisgidsen geen tijdelijke informatie weer zoals tijdelijke exhibities in musea, renovaties of voorstellingen in theaters (Dunlop et al., 2004, in Souffriau et al., 2008). Ten derde moet een toerist informatie van verschillende bronnen combineren en beslissen welke informatie voor hem relevant is. Tenslotte is het voor een toerist moeilijk om te beslissen welke attractie voor hem de meeste waarde heeft en weet hij vaak niet of zijn reisschema nu het beste is of niet (Souffriau et al., 2008).

## 1.2 Centrale onderzoeksvraag

Zoals vermeld in sectie 1.1, maken steeds meer mensen gebruik van het internet om hun reis te plannen. Ze verwachten wel dat hulp aangeboden wordt bij het plannen van hun trip (Zhu et al., 2010). In verschillende takken van het bedrijfsleven bestaan reeds beslissingsondersteunende systemen. Zo worden managers in de financiële afdeling bijgestaan bij het maken van investeringsbeslissingen, in de marketingafdeling krijgen ze steun bij het verzamelen van data, in de productieafdeling wordt hulp geboden bij het plannen van de productie en in de logistieke afdeling bestaan er beslissingsondersteunende systemen voor het distribueren van goederen. Dergelijke systemen zijn echter nog weinig voorhanden bij het plannen van vrije tijd. Een beslissingsondersteunend systeem zou kunnen helpen bij het plannen van een efficiënte en interessante route per dag voor een toerist (Schilde et al., 2009; Souffriau et al., 2008).

Reisgidsen bieden vooraf opgestelde routes aan met elk een verschillende lengte afhankelijk van de tijd die de toerist ter beschikking heeft. Dit is echter een simplistische voorstelling van de realiteit aangezien elke toerist verschillende voorkeuren heeft. De ideale oplossing is een individueel adviserend beslissingsondersteunend systeem waar interactie met de toerist mogelijk is (Schilde et al., 2009).

De centrale onderzoeksvraag wordt omschreven als: *"Is het mogelijk een toeristische route te plannen met als doel de meest interessante locaties te bezoeken, rekening houdend met budget- en tijdsbeperkingen alsook met verscheidene criteria aangaande de aantrekkelijkheid van locaties voor een toerist?"*

### 1.3 Deelvragen

Met behulp van onderstaande deelvragen wordt de centrale onderzoeksvraag gestructureerd uitgediept.

- *Welke modellen voor routeplanning voor toeristen worden voorgesteld in de wetenschappelijke literatuur?*
- *Hoe kan het probleem mathematisch geformuleerd en opgelost worden?*
- *Welke criteria worden gebruikt om een locatie al dan niet als interessant te definiëren?*
- *Zijn de applicaties die in de praktijk reeds ontwikkeld zijn, in staat het probleem in realtime op te lossen?*

### 1.4 Onderzoeksopzet

De onderzoeksopzet bestaat uit twee delen. Eerst wordt de beschikbare wetenschappelijke literatuur bestaande over toeristische problemen doorgenomen met behulp van beschikbare databanken. Het uiteindelijke doel is de theorie, besproken in de literatuurstudie, te toetsen aan de praktijk. Het eerste gedeelte van de praktijkstudie bestaat uit het verzamelen van gegevens over de stad Hasselt en het vergaren van informatie van een fictieve toerist. Hiervoor wordt de hulp van de toeristische dienst van de stad Hasselt ingeschakeld. Het toeristisch probleem dat centraal staat in deze eindverhandeling wordt vervolgens mathematisch geformuleerd. Hierna worden verschillende operatoren uit beschikbare oplossingsmethoden in de literatuur gecombineerd tot een geschikt algoritme voor dit probleem. Het algoritme wordt getest en de uiteindelijke oplossing wordt voorgesteld aan de toerist.





## Hoofdstuk 2      Literatuurstudie

### 2.1 Inleiding

Dankzij informatie- en communicatietechnologie krijgen reizigers op eenvoudige wijze toegang tot betrouwbare en accurate informatie. Toerisme-instanties, privéondernemingen en andere gebruikers verschaffen informatie aan reizigers. Reizigers zoeken naar reisgerelateerde informatie en boeken vliegtickets, hotelreservaties en andere aankopen online (Morrison et al., 2001, in Buhalis & Law, 2008). Reizigers kunnen sneller reserveringen maken met minder ongemak en tegen lagere kosten dan wanneer ze bijvoorbeeld de hulp van een reisbureau inschakelen. Een goed geïnformeerde reiziger zal in staat zijn beter te interageren met lokale bronnen en culturen, producten en diensten te vinden die voldoen aan zijn behoeften en meer voordeel te genieten van speciale aanbiedingen en eventuele kortingen (Buhalis & Law, 2008).

Elke reiziger is anders. Ze hebben andere ervaringen, motivaties en verlangens. Het doel van een toerist is niet langer het volgen van de menigte in vooraf samengestelde rondreizen. Het doel is geëvolueerd naar het nastreven van eigen voorkeuren en schema's. Snelle identificatie van consumentenbehoeften en het verschaffen van gepersonaliseerde, up-to-date producten en diensten zijn cruciaal in de hedendaagse toerisme-industrie (Buhalis & Law, 2008). Het internet draagt in grote mate bij aan het gewijzigde gedrag van reizigers. Reizigers kunnen via het internet dynamisch interageren met de betrokken partijen (Jeong et al., 2003). Het ontwikkelen van persoonlijke profielen zal leiden tot een betere personalisatie en een betere interactie tussen reizigers en toerismebedrijven. Toerismebedrijven dienen informatie van reizigers te verzamelen zowel voor, tijdens als na de rondreis om zo het gedrag, de keuzes en bedenkingen van reizigers te begrijpen (Buhalis & Law, 2008).

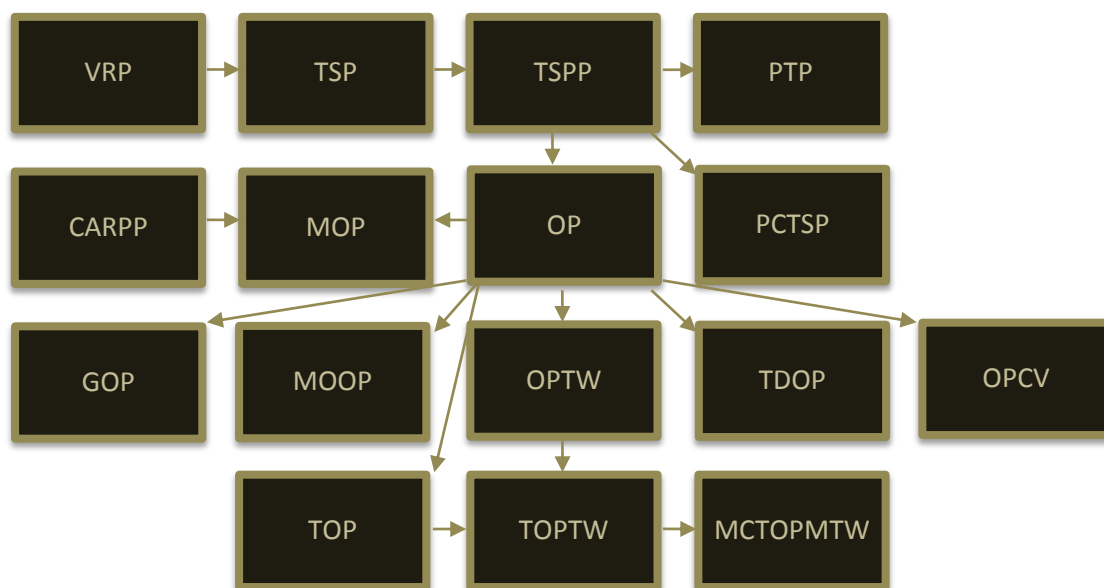
Een toerist kan door onder andere tijds- en budgetbeperkingen niet alles bezoeken in een bepaalde stad of regio waarin hij geïnteresseerd is. Het plannen van een route voor toeristen op basis van verschillende criteria is geen vanzelfsprekende taak. Het is een complexe constructieve activiteit. Verschillende auteurs, bijvoorbeeld Schafer et al. (1999), Fesenmainer et al. (2003) en Ricci en Wether (2002), bespreken aanbevelingsystemen gebaseerd op artificiële intelligentie die de toerist kunnen helpen bij het plannen van zijn route (Buhalis & Law, 2008). Onafhankelijk van het gebruikte aanbevelingssysteem voor het plannen van een reispakket is een referentiebasis nuttig om het selectieproces te simplificeren. De referentiebasis kan bestaan uit eerdere aankopen, selecties van de gebruiker of andere gebruikers of een vooraf beschikbaar reispakket aangeboden door een reisagentschap of andere reizigers (Zhu et al., 2010).

De literatuurstudie is gebaseerd op het feit dat het plannen van een toeristische route geformuleerd kan worden als een *tourist trip design* probleem (TTDP). Reeds verscheidene auteurs ontwikkelden oplossingsmethoden voor dit probleem (Souffriau en Vansteenwegen, 2010; Vansteenwegen et al., 2011c; Gavalas et al., 2012). Hoofdstuk 2 is als volgt georganiseerd. Sectie 2.2 formuleert het *orientering* probleem (OP) mathematisch waarna enkele praktische toepassingen en oplossingsmethoden worden aangehaald. Sectie 2.3 bespreekt een uitbreiding op het OP, namelijk het *team orientering* probleem met tijdvensters waarna in sectie 2.4 eveneens meerdere beperkingen worden toegevoegd aan laatstgenoemd probleem. Het *team orientering* probleem met meerdere tijdvensters en meerdere beperkingen benadert de centrale onderzoeksvraag geformuleerd in sectie 1.2. Sectie 2.5 vermeldt enkele varianten op het *tourist trip design* probleem. Sectie 2.6 neemt naast de routeplanning ook het selecteren van een hotel in acht. Sectie 2.7 concludeert de literatuurstudie.

## 2.2 Oriëntering probleem

### 2.2.1 Inleiding

Figuur 1 toont de link tussen de problemen die in de literatuurstudie worden besproken.



Figuur 1: Overzicht problemen (gebaseerd op Gavalas et al., 2012)

Het *vehicle routing* probleem (VRP), of het rittenplanningsprobleem, is uitgegroeid tot een van de meest bestudeerde problemen in combinatorische optimalisatie en komt vaak voor in de transportindustrie. Voor een aantal voertuigen worden routes bepaald. Deze voertuigen moeten klanten bedienen rekening houdend met tijds- en/of capaciteitsbeperkingen. Het doel betreft het streven naar kostenminimalisatie (Fung et al., 2011).

Het meest bekende rittenplanningsprobleem is het *travelling salesman* probleem (TSP) (Hillier & Lieberman, 2010; Labadi et al., 2009). In het TSP wordt een set van steden verondersteld met tussen elke twee steden een gegeven afstand of een kost om van de ene stad naar de andere te reizen. Een verkoper moet bij eenzelfde stad starten en eindigen. Hij moet elke stad bezoeken en mag geen enkele stad meer dan één keer bezoeken. Zijn doel is een pad te vinden waarbij de afgelegde afstand of de verzamelde kosten geminimaliseerd worden (Chong et al., 2010).

Het *travelling salesman problem with profits* (TSPP) verwacht niet dat alle knooppunten bezocht worden. Bijkomend wordt aan elk knooppunt een winst toegekend. Het doel is het gelijktijdig optimaliseren van de verzamelde winsten en de reiskosten. De verzamelde winsten worden dus gemaximaliseerd en de reiskosten geminimaliseerd. Feillet et al. (2005) definiëren het TSP met winsten als een *orienteeering* probleem (OP) als het minimaliseren van de reiskosten een beperking vormt en het doel is een pad te vinden dat de verzamelde winst maximaliseert. Hierbij is het belangrijk dat de reiskosten het vooropgestelde budget niet overstijgen. Dit wordt ook het *selective TSP*, het *bank robber* probleem of het *maximum collection* probleem genoemd. Het verschil met het TSP is dat bij het OP slechts een deelverzameling van knooppunten bezocht moet worden. Toegepast op de verkoper is deze aldus niet verplicht elke stad te bezoeken (Feillet et al., 2005; Gavalas et al., 2012).

Het plannen van routes voor toeristen die geïnteresseerd zijn in het bezoeken van meerdere *points of interest* (POI) wordt in de literatuur omschreven als het *tourist trip design* probleem (TTDP). Rekening houdend met beperkingen van de toerist en kenmerken van de POI, worden oplossingen voor dit probleem gezocht. Het doel is POI te selecteren in overeenstemming met de voorkeuren van de toerist. Hierbij wordt rekening gehouden met verschillende parameters en beperkingen zoals afstanden tussen de POI, de nodige bezoektijd voor elk POI, openings- en sluitingsuren van de POI, toegangsgelden en weersomstandigheden. De beschikbare tijd die de toerist op dagelijkse basis heeft om de verschillende POI te bezoeken wordt gerespecteerd. Hierdoor wordt de voldoening van de toerist gemaximaliseerd. Wanneer een TTDP opgelost wordt, moet de uitkomst dus bestaan uit dagelijks geordende bezoeken aan POI rekening houdend met de beperkingen van de gebruiker alsook met de kenmerken van de POI. Hoge kwaliteitsoplossingen zorgen voor aanbevelingen in overeenstemming met de voorkeuren van de toerist en een toegelaten (bijna) optimaal reisschema. De literatuur haalt verschillende modelleringbenaderingen aan voor routeplanning. Het OP wordt gebruikt om de meest eenvoudige versie van TTDP te modelleren (Gavalas et al., 2012).

Tabel 1 geeft een overzicht van eerder verricht onderzoek naar het OP. De auteur(s), de toepassing en de oplossingsmethode worden aangehaald. De informatie in onderstaande tabel zal in de volgende deelsecties worden toegelicht. Dit is geenszins een uitputtende opsomming van wat in eerdere literatuur reeds besproken is aangaande de behandelde materie. De keuze van opname is gebaseerd op de paper van Vansteenwegen et al. (2011c).

Auteur	Toepassing	Oplossingsmethoden
Tsiligirides (1984)	Verkoper met tijdsgebrek	Stochastisch en deterministisch algoritme
Golden et al. (1987)	<i>Home fuel delivery</i> probleem	<i>Centre-of-gravity</i> heuristiek
Laporte en Martello (1990)	<i>Hamiltonian Circuit</i> probleem	<i>Branch and bound</i>
Ramesh en Brown (1991)	<i>Control theory</i>	Heuristiek met vier fasen
Ramesh et al. (1992)	<i>Orienteering Tour</i> probleem	<i>Branch and bound</i>
Leifer en Rosenwein (1994)	-	<i>Cutting plane</i> methode
Chao et al. (1996)	<i>Sport Orienteering Game</i>	Heuristiek met vijf fasen
Fischetti et al. (1998)	-	<i>Branch and cut</i>
Gendreau et al. (1998a)	<i>Undirected Selective TSP</i>	<i>Tabu search</i>
Gendreau et al. (1998b)	<i>Undirected Selective TSP</i>	<i>Branch and cut</i>
Souffriau et al. (2008)	<i>Mobile Tourist Guide</i>	<i>Guided local search</i>
Wang et al. (2008)	Militaire toepassing	Genetisch algoritme

Tabel 1: Samenvattende tabel orienteering problemen

### 2.2.2 Mathematische formulering

De essentie van het probleem wordt voorgesteld via een mathematische formulering. De doelfunctie wordt geformuleerd aan de hand van beslissingsvariabelen. Aan deze variabelen kunnen restricties worden opgelegd. De parameters zijn constanten waaronder de coëfficiënten van de beslissingsvariabelen en de rechterzijden. Het probleem houdt het bepalen van de waarden van de beslissingsvariabelen in met oog op het maximaliseren van de doelfunctie rekening houdend met de beperkingen. Varianten op dit probleem zoals het minimaliseren van de doelfunctie, zijn mogelijk (Hillier & Lieberman, 2010).

In het *orienteering* probleem (OP) worden  $N$  knooppunten gegeven, elk met een score  $S_i$ . Elk knooppunt  $i$  verwijst naar een locatie waarbij de score de aantrekkelijkheid van die locatie aanduidt. De beginlocatie 1 en de eindlocatie  $N$  zijn vast. De tijd nodig om van locatie  $i$  naar locatie  $j$  te gaan, wordt aangeduid met  $t_{ij}$  en is gegeven voor alle knooppunten. Het doel is een pad te vinden dat de som van de scores maximaliseert rekening houdend met een tijdsbudget  $T_{max}$ . Dit tijdsbudget kan ook een afstandsbudget  $D_{max}$  zijn. Mede door deze beperking is het aldus niet de bedoeling alle knooppunten te bezoeken. Alle scores moeten optelbaar zijn en elk knooppunt kan maximaal één keer bezocht worden. In het OP wordt verondersteld dat de begin- en eindlocatie van elkaar verschillen. In veel toepassingen zijn de begin- en eindlocatie echter dezelfde. Dit verschil in formulering kan opgelost worden door in het tweede geval een dummy variabele toe te voegen die een boog vormt tussen het begin- en eindknooppunt (Vansteenwegen et al., 2011c).

Vansteenwegen et al. (2011c) formuleren het OP mathematisch als een integer programmeringsprobleem. Volgende beslissingsvariabelen worden gebruikt:  $x_{ij} = 1$  als een bezoek aan knooppunt  $i$  gevolgd wordt door een bezoek aan knooppunt  $j$  en 0 in alle andere gevallen;  $u_i$  = de positie van knooppunt  $i$  in het pad. Tabel 2 toont de gebruikte symbolen in de mathematische formulering.

$i, j$	knooppunten
$N$	aantal knooppunten
$S_i$	score knooppunt $i$
$t_{ij}$	tijd tussen knooppunt $i$ en knooppunt $j$
$T_{max}$	tijdsbudget
$x_{ij}$	knooppunt $j$ volgt op knooppunt $i$
$u_i$	positie knooppunt $i$

Tabel 2: Symbolenlijst oriëntering probleem

In onderstaande formulering vormt (1) de doelfunctie die de totale score tracht te maximaliseren. Door beperking (2) start het pad in knooppunt 1 en eindigt het in knooppunt  $N$ . Beperking (3) zorgt ervoor dat elk knooppunt maximaal één keer bezocht wordt en dat de knooppunten die worden bezocht, gelinkt zijn. Met behulp van beperking (4) wordt het tijdsbudget niet overschreden. Door beperking (5) en (6) worden subtoeren vermeden. Deze laatste twee beperkingen zijn geformuleerd volgens de Miller-Tucker-Zemlin formulering van het TSP (Miller et al., 1960, in Vansteenwegen et al., 2011c). In de literatuur wordt een symmetrische reistijd tussen de knooppunten verondersteld. Hieruit volgt dat  $t_{ij}$  gelijk is aan  $t_{ji}$  (Vansteenwegen et al., 2011c).

$Max \sum_{i=2}^{N-1} \sum_{j=2}^N S_i x_{ij}$	(1)
$\sum_{j=2}^N x_{1j} = \sum_{i=1}^{N-1} x_{iN} = 1$	(2)
$\sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^N x_{kj} \leq 1; \forall k = 2, \dots, N-1$	(3)
$\sum_{i=1}^{N-1} \sum_{j=2}^N t_{ij} x_{ij} \leq T_{max}$	(4)
$2 \leq u_i \leq N; \forall i = 2, \dots, N$	(5)
$u_i - u_j + 1 \leq (N-1)(1 - x_{ij}); \forall i, j = 2, \dots, N$	(6)
$x_{ij} \in \{0,1\}; \forall i, j = 1, \dots, N$	(7)

### 2.2.3 Praktische toepassingen

Het OP heeft meerdere toepassingen. Een eerste toepassing omvat het *home fuel delivery* probleem, beschreven door Golden et al. (1987). Vrachtwagens moeten op dagelijkse basis brandstof leveren aan klanten. Het brandstofniveau van de klant moet op elk moment aanvaardbaar zijn. De dringendheid kan bepaald worden door een voorspelling van het voorraadmiveau. Op basis van de dringendheid van levering worden een deelverzameling van alle klanten gekozen voor bediening. Het doel is een efficiënte route voor elke vrachtwagen bepalen. Een tweede toepassing, geformuleerd door Tsiligirides (1984, in Vansteenwegen et al., 2011c) is het TSP waarbij de verkoper geen tijd heeft om alle steden te bezoeken. Wang et al. (2008) vermelden een militaire toepassing. Wanneer een onderzeeër of onbemand vliegtuig betrokken is bij bewakingsactiviteiten, is de lengte van de expeditie beperkt door brandstof of tijd. Het doel is de beste deelverzameling van mogelijke knooppunten te bezoeken of te fotograferen.

Het OP kan fungeren als startpunt voor het modelleren van TTDP (Souffriau et al., 2012). Een toerist heeft altijd een tijdsbeperking waardoor hij nooit alle toeristische attracties kan

bezoeken. De toerist moet de attracties selecteren die hem het meeste waarde opleveren. De score van de locatie representeert de geschatte persoonlijke interesse van de toerist in die locatie. Wanneer het probleem wordt opgelost, resulteert dit in een persoonlijke route voor de toerist (Vansteenwegen et al., 2009; Souffriau et al., 2012). Het OP integreert automatische locatieselectie met het vinden van het kortste pad. Om die reden is het OP geschikt voor het modelleren van TTDP (Souffriau en Vansteenwegen, 2010). Het TTDP werd onder andere onderzocht door Vansteenwegen en Van Oudheusden (2007, in Vansteenwegen et al., 2011c), Wang et al. (2008) en Schilde et al. (2009).

#### 2.2.4 Oplossingsmethoden

Bij de oplossingsmethoden wordt onderscheid gemaakt tussen exacte en heuristische oplossingsmethoden. Een exact algoritme kan leiden tot een optimale oplossing voor het probleem. Een heuristiek wordt gebruikt om een toegelaten suboptimale oplossing te vinden en garandeert dus geen optimale oplossing. Enerzijds worden heuristieken gebruikt wanneer de kost of tijd geassocieerd met het vinden van een optimale oplossing te groot is. Anderzijds zijn bepaalde problemen te complex waardoor het onmogelijk is een optimale oplossing te vinden. Bijgevolg is het gebruik van een heuristiek noodzakelijk. Vaak wordt een iteratief algoritme gebruikt bij heuristieken. In elke iteratie wordt op zoek gegaan naar een betere oplossing dan de huidige oplossing. Voor elk specifiek probleem is een op maat gemaakte heuristiek nodig. Daarom worden metaheuristieken ontwikkeld. Metaheuristieken voorzien de structuur en strategie om een specifieke heuristische procedure voor een bepaald probleem te ontwerpen (Hillier en Lieberman, 2010).

##### 2.2.4.1 Exacte algoritmen

In de studie van Vansteenwegen et al. (2011c) worden enkele onderzoekers vernoemd die een exact algoritme als oplossingsmethode voor het OP voorstellen. Laporte en Martello (1990) en Ramesh et al. (1992) maken gebruik van de *branch-and-bound* techniek voor respectievelijk het *hamiltonian circuit* en het *orienting tour* probleem. Leifer en Rosenwein (1994) maken gebruik van de *cutting plane* methode. Gendreau et al. (1998b) beschrijven een branch-and-cut algoritme voor het *selective TSP* waarbij een verplicht bezoek aan een bepaald aantal knooppunten vooropgesteld wordt. Fischetti et al. (1998) gebruiken eveneens de branch-and-cut techniek.

##### 2.2.4.2 Heuristische oplossingsmethoden

Aangezien het toepassen van dergelijke exacte algoritmen zeer tijdrovend is, worden vaak heuristieken gebruikt. Gendreau et al. (1998b) geven enkele redenen waarom het moeilijk is goede heuristieken te ontwikkelen voor het OP. Ten eerste zijn de score van een knooppunt en de tijd om het knooppunt te bereiken onafhankelijk van elkaar en spreken ze elkaar soms tegen. Dit bemoeilijkt de keuze van knooppunten voor de optimale oplossing. Ten tweede kunnen eenvoudige constructie- en verbeteringheuristieken ervoor zorgen dat het algoritme

niet in de gewenste richting gaat. De heuristische onderzoeken grote delen van de oplossingsruimte niet nauwkeurig genoeg en verkeerde beslissingen kunnen niet voldoende gecorrigeerd worden. Ten derde wordt het steeds moeilijker om het kortste pad tussen de geselecteerde knooppunten te vinden wanneer het aantal knooppunten stijgt. In de studies van Vansteenwegen et al. (2011c) en Gavalas et al. (2012) worden enkele onderzoekers vermeld die voorstellen om het OP met heuristiek op te lossen.

Tsiligirides (1984, in Vansteenwegen et al., 2011c) stelt een stochastisch algoritme voor dat een groot aantal routes construeert en de route kiest met de maximale winst. De auteur stelt ook een deterministisch algoritme voor dat de oppervlakte in cirkels verdeelt en de toegelaten routes tot die cirkels beperkt. Golden et al. (1987) hebben een *centre-of-gravity* heuristiek ontwikkeld, gebruik makend van een Euclidische metriek. Eén iteratie bestaat uit drie stappen. De eerste stap omvat het iteratief invoegen van knooppunten met een hoge score en aanvaardbare duur. De tweede stap is een verbeteringsprocedure die gebruik maakt van 2-opt en *cheapest insertion*. 2-opt bestaat uit het verbreken van twee verbindingen in één route en deze te vervangen door twee nieuwe verbindingen zodat een nieuwe route ontstaat (Zhu et al., 2010). *Cheapest insertion* is het kiezen van een knooppunt uit alle niet-opgenomen knooppunten waarbij de invoeging de laagste stijging in bijvoorbeeld de lengte van de route veroorzaakt (Golden et al., 1987). In de derde stap wordt een nieuw pad gevormd door alle knooppunten opnieuw te rangschikken gebaseerd op de verhouding van de score over de afstand tot de *centre-of-gravity* van het vorige pad. Ramesh en Brown (1991) introduceren een heuristiek bestaande uit vier fasen om een oplossing te bieden voor de *control theory*.

De heuristiek van Chao et al. (1996) overweegt enkel knooppunten die bereikt kunnen worden binnen de opgelegde tijdsbeperking. Hij bestaat uit vijf stappen en is ontwikkeld om het OP op te lossen. De toepassing van Chao et al. (1996) heeft betrekking op een spel waarbij individuele concurrenten starten aan een specifiek controlepunt. Het doel is zoveel mogelijk controlepunten binnen een bepaald tijdvenster bezoeken. Daarna moeten de concurrenten terugkeren naar het beginnend controlepunt. Elk controlepunt heeft een bepaalde score en het doel is de totale verzamelde score te maximaliseren. De heuristiek start met de initialisatie (fase 1). De initialisatie creëert verschillende paden. Elk pad start met een knooppunt ver verwijderd van het begin- en eindknooppunt. Alle andere knooppunten worden aan één van de paden toegekend via *cheapest insertion*. Het beste pad wordt als initiële oplossing  $T_{op}$  beschouwd. De knooppunten die niet in deze initiële oplossing zijn opgenomen, worden aan andere toegelaten paden  $T_{nop}$  toegekend. De eerste verbeteringsstap (fase 2) probeert  $T_{op}$  te verbeteren door een extra knooppunt in te voegen uit één van de  $T_{nop}$  en een opgenomen knooppunt te verplaatsen naar één van de  $T_{nop}$ , gebruik makend van *cheapest insertion*. Alle paden moeten toegelaten blijven en een kleine daling van de totale score is toegelaten. De tweede verbeteringsstap (fase 3) zal een knooppunt van één pad naar een ander pad verplaatsen. Dit vindt enkel plaats als de stap



toegelaten is en de totale score stijgt of met een aanvaardbare hoeveelheid daalt. De derde verbeteringsstap (fase 4) gebruikt 2-Opt. Uiteindelijk worden de knooppunten die een lage verhouding van score op invoegkost vertonen, verwijderd (fase 5) en wordt het algoritme hernomen.

Gendreau et al. (1998a) stellen een *tabu search* heuristiek voor om het *undirected selective* TSP op te lossen. Deze heuristiek zorgt ervoor dat de kans dat het algoritme vast komt te zitten in een lokaal optimum verlaagt. Ook de kans dat een ver gelegen knooppunt met hoge score wordt opgenomen, wordt verminderd. Hillier en Lieberman (2010) beschrijven de procedure van *tabu search*. In de initialisatie wordt gestart met een toegelaten initiële oplossing. De iteratie bestaat uit het uitvoeren van een lokale zoekprocedure om toegelaten bewegingen te definiëren in de lokale nabijheid van de huidige oplossing. De lokale zoekprocedure vereist niet dat elke nieuwe oplossing beter is dan de vorige oplossing. Bewegingen op de taboelijst mogen niet uitgevoerd worden behalve wanneer de beweging kan leiden tot een betere oplossing dan de huidige oplossing. Na de bepaling van de meest optimale beweging wordt de oplossing geclassificeerd als de huidige oplossing onafhankelijk van het feit of deze oplossing al dan niet beter is dan de vorige huidige oplossing. De taboelijst wordt aangepast om te vermijden dat de vorige huidige oplossing terugkeert. Wanneer de lijst vol is, dient het oudste lid verwijderd te worden zodat een nieuw lid kan worden toegevoegd. Om de procedure te stoppen, wordt een criterium naar keuze gebruikt bijvoorbeeld een vast aantal iteraties of een vast aantal opeenvolgende iteraties zonder enige verbetering in de huidige doelfunctiewaarde. Ook wanneer geen toegelaten beweging mogelijk is, stopt de procedure. Aangezien *tabu search* een metaheuristiek is en dus enkel een algemene structuur en strategie aangeeft, moeten de details voor elk specifiek probleem worden aangevuld (Hillier en Lieberman, 2010).

Souffriau et al. (2008) gebruiken artificiële intelligentie gecombineerd met metaheuristieken om het OP op te lossen. Ze ontwikkelden een *guided local search* (GLS) metaheuristiek (Souffriau et al., 2006, in Souffriau et al., 2008). GLS verbetert twee basis heuristieken die reeds hoge kwaliteitsoplossingen bieden. Het geeft een straf, gebaseerd op een nutsfunctie, aan niet gewilde oplossingen tijdens een lokale zoekprocedure. Hierdoor verkleint de kans op vastzitten in een lokaal optimum (Voudouris en Tsang, 1999). Souffriau et al. (2008) gebruiken het algoritme voor een klein mobiel apparaat, de *mobile tourist guide*, dat snelle ondersteuning geeft aan de besluitvorming van toeristen. De benadering van Souffriau et al. (2008) wordt getest in de stad Gent. De militaire toepassing van Wang et al. (2008) wordt opgelost met een genetisch algoritme.

## 2.3 Team oriëntering probleem met tijdvensters

### 2.3.1 Inleiding

Het team *oriëntering* probleem (TOP) is een uitbreiding op het *oriëntering* probleem (OP). Met behulp van een formulering van dit probleem kunnen meerdere routes worden gemodelleerd (Chao et al., 1996; Souffriau et al., 2012). Het doel is  $m$  paden te formuleren die de totale verzamelde score maximaliseren. Elk pad dient te voldoen aan een tijdsbudget. Naast het OP bestaat ook het OP met tijdvensters (*oriëntering problem with time windows*, OPTW). In dit probleem worden tijdvensters toegevoegd waardoor een periode gedefinieerd kan worden voor elk knooppunt. In deze periode moet het bezoek plaatsvinden. Wanneer het om het bezoeken van toeristische plaatsen gaat, bestaat het tijdvenster uit de openings- en sluitingstijden van een bepaalde plaats (Vansteenwegen et al., 2011c).

Vansteenwegen et al. (2009) bespreken het *team oriëntering* probleem *with time windows* (TOPTW). Dit is een combinatie van het TOP en het OPTW. Een set van locaties is gegeven waarbij elke locatie een score, een bezoektijd en een tijdvenster krijgt. Het doel is meerdere routes te bepalen, voor elke dag één, waarbij de som van de verzamelde scores gemaximaliseerd wordt. Elke route is beperkt in lengte en kan geïnterpreteerd worden als een dagtrip.

### 2.3.2 Mathematische formulering

In het OPTW zijn  $N$  locaties gegeven waarbij elke locatie een score  $S_i$ , een bezoektijd  $T_i$  en een tijdvenster  $[O_i, C_i]$  krijgt. Elke locatie kan maximaal één keer bezocht worden. Het is toegelaten reeds op een locatie te arriveren vooraleer het tijdvenster start. Het startknooppunt 1 en het eindknooppunt  $N$  van elke route zijn vast. De tijd nodig om van locatie  $i$  naar locatie  $j$  te reizen  $t_{ij}$  is gekend voor alle locaties. Het tijdsbudget  $T_{max}$  verhindert dat alle locaties bezocht kunnen worden. Het doel is enerzijds één route te bepalen waarin enkele locaties bezocht worden en anderzijds de som van de verzamelde scores te maximaliseren, rekening houdend met  $T_{max}$  en de openings- en sluitingstijden (Vansteenwegen et al., 2009).

Bij het TOPTW wordt het doel gewijzigd naar het bepalen van  $m$  routes in tegenstelling tot het bepalen van slechts één route bij het OPTW. Toegepast op de toerismeproblematiek geeft het TOPTW de mogelijkheid routes voor meerdere dagen te modelleren. Het TOPTW wordt geformuleerd als een integer programmeringsprobleem. Volgende variabelen worden gedefinieerd:  $x_{ijd} = 1$  als in route  $d$  een bezoek aan locatie  $i$  gevolgd wordt door een bezoek aan locatie  $j$  en 0 elders;  $y_{id} = 1$  als locatie  $i$  bezocht wordt in route  $d$  en 0 elders;  $s_{id} =$  het begintijdstip van een bezoek aan locatie  $i$  in route  $d$ .  $M$  is een constante met hoge waarde (Vansteenwegen et al., 2009). Tabel 3 geeft een oplisting van de nieuwe symbolen in de formulering. De eerder gebruikte symbolen die in onderstaande formulering terugkeren, zijn terug te vinden in tabel 2.

$k$	knooppunt
$d$	route
$T_i$	bezoektijd knooppunt $i$
$[O_i, C_i]$	tijdvenster
$m$	aantal routes
$x_{ijd}$	knooppunt $j$ volgt op knooppunt $i$ in route $d$
$y_{id}$	knooppunt $i$ in route $d$
$s_{id}$	begintijdstip (locatie $i$ , route $d$ )
$a_i$	aankomst knooppunt $i$
$M$	constante

Tabel 3: Symbolenlijst TOPTW

De doelfunctie, aangeduid met (1), tracht de totale score te maximaliseren. Beperking (2) zorgt ervoor dat alle routes starten in locatie 1 en eindigen in locatie  $N$ . Door beperking (3) is de connectiviteit in elke route verzekerd. Beperking (4) bepaalt de tijdslijn van elke route. Beperking (5) verzekert dat elke locatie maximaal één keer bezocht wordt. Het tijdsbudget wordt gehandhaafd dankzij beperking (6). Beperkingen (7) en (8) zorgen ervoor dat de start van het bezoek binnen het tijdvenster valt (Vansteenwegen et al., 2009).

$$\begin{aligned}
 & \text{Max } \sum_{d=1}^m \sum_{i=2}^{N-1} S_i y_{id} & (1) \\
 & \sum_{d=1}^m \sum_{j=2}^{N-1} x_{1jd} = \sum_{d=1}^m \sum_{i=2}^{N-1} x_{iNd} = m & (2) \\
 & \sum_{i=1}^{N-1} x_{ikd} = \sum_{j=2}^N x_{kjd} = y_{kd}; \forall k = 2, \dots, N; \forall d = 1, \dots, m & (3) \\
 & s_{id} + T_i + t_{ij} - s_{jd} \leq M(1 - x_{ijd}); \forall i, j = 1, \dots, N; \forall d = 1, \dots, m & (4) \\
 & \sum_{d=1}^m y_{kd} \leq 1; \forall k = 2, \dots, N - 1 & (5) \\
 & \sum_{i=1}^{N-1} (T_i y_{id} + \sum_{j=2}^N t_{ij} x_{ijd}) \leq T_{max}; \forall d = 1, \dots, m & (6) \\
 & O_i \leq s_{id}; \forall i = 1, \dots, N; \forall d = 1, \dots, m & (7) \\
 & s_{id} \leq C_i; \forall i = 1, \dots, N; \forall d = 1, \dots, m & (8) \\
 & x_{ijd}, y_{id} \in \{0,1\}; \forall i, j = 1, \dots, N; \forall d = 1, \dots, m & (9)
 \end{aligned}$$

### 2.3.3 Oplossingsmethoden

Gezien de complexiteit van het probleem bestaat de literatuur betreffende oplossingsmethoden voor het TOPTW voornamelijk uit metaheuristieken. In metaheuristieken worden vaak lokale zoekbewegingen gebruikt om te kunnen ontsnappen aan lokale optima (Gavalas et al., 2012). Vansteenwegen et al. (2011c) vermelden enkele lokale zoekbewegingen die vaak gebruikt worden in de oplossingsmethoden voor het TOP. Om de totale score te verhogen, worden vijf bewegingen gebruikt: Insert, TwoInsert, Replace, TwoReplace en Change. De totale reistijd wordt verlaagd met behulp van twee bewegingen: 2-Opt en Swap. Via de methode van *cheapest insertion* voegt de Insert beweging een extra knooppunt toe aan het huidige pad. De TwoInsert beweging voegt twee extra knooppunten toe aan het huidige pad. Bij de Replace beweging wordt een opgenomen knooppunt met lagere score vervangen door een niet-opgenomen knooppunt met hogere score, indien het tijdsbudget geen Insert beweging meer toelaat. De TwoReplace beweging doet hetzelfde als de Replace beweging maar dan met twee knooppunten. De Change beweging verwijdert vijf

opeenvolgende knooppunten uit het huidige pad en voegt één voor één nieuwe knooppunten toe die nog niet in het huidige pad aanwezig waren. Bij 2-Opt worden twee verbindingen vervangen door twee nieuwe verbindingen die oorspronkelijk niet in het huidige pad zaten. De Swap beweging verwisselt twee knooppunten die tot een verschillend pad behoren om reistijd te besparen (Vansteenwegen et al., 2011c).

Garcia et al. (2010) en Gavalas et al. (2012) vermelden enkele auteurs die reeds oplossingsmethoden voorgesteld hebben voor het OPTW of een variant hierop, samengevat in tabel 4. Righini en Salani (2009) lossen het OPTW exact op aan de hand van bidirectionele dynamische programmering. Tricoire et al. (2010) gebruiken een *variable neighbourhood search* (VNS) methode om het *multi period OP with multiple time windows* (MuPOPMTW) op te lossen. Het idee achter VNS is een systematische verandering van de buurt die doorzocht wordt tijdens een lokale zoekprocedure. De oplossingsmethode van Montemanni en Gambardella (2009) gebruikt *ant colony* systemen om het TOPTW op te lossen. Lin et al. (2012) introduceren *simulated annealing* om het TOPTW op te lossen. De drie laatstgenoemde oplossingsbenaderingen maken aldus gebruik van een heuristiek. Vansteenwegen et al. (2009) maken gebruik van een *iterated local search* metaheuristiek om het TOPTW op te lossen. *Iterated local search* (ILS) is het iteratief opbouwen van volgordes van oplossingen gegenereerd door een geïntegreerde heuristiek genaamd *local search*. Op die manier worden betere oplossingen verkregen dan wanneer willekeurige pogingen van dezelfde heuristiek herhaaldelijk worden uitgevoerd (Garcia et al., 2010).

Auteur	Toepassing	Oplossingsmethoden
Montemanni en Gambardella (2009)	TOPTW	<i>Ant colony</i> systemen
Righini en Salani (2009)	OPTW	Bidirectionele dynamische programmering
Vansteenwegen et al. (2009)	TOPTW	<i>Iterated Local Search</i>
Tricoire et al. (2010)	MuPOPMTW	<i>Variable Neighbourhood Search</i>
Lin et al. (2012)	TOPTW	<i>Simulated Annealing</i>

Tabel 4: Oplossingsmethoden OPTW en varianten

Het doel van Vansteenwegen et al. (2009) is resultaten van hoge kwaliteit leveren. Deze resultaten moeten snel berekend worden aangezien toeristen realtime alternatieven wensen wanneer hun route onverwacht wijzigt. De metaheuristiek van Vansteenwegen et al. (2009) wordt hieronder kort toegelicht. In de praktijkstudie wordt dieper ingegaan op deze oplossingsmethode.

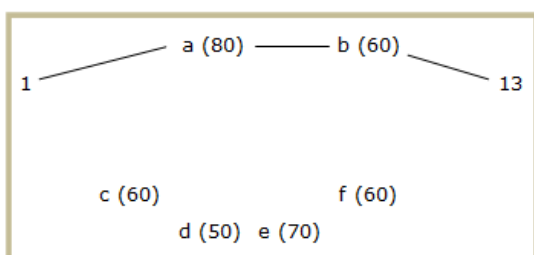
Een *insert* stap wordt gecombineerd met een *shake* stap. De *insert* stap voegt één voor één nieuwe locaties toe aan de route. Vooraleer een locatie wordt toegevoegd, wordt nagegaan of de locaties die hierna nog bezocht moeten worden nog steeds voldoen aan hun tijdvensterbeperving. Om een snelle heuristiek te ontwikkelen, is nodig dat de locaties die mogelijk kunnen worden toegevoegd op snelle wijze geëvalueerd worden. Alle andere locaties onderzoeken, zou teveel tijd in beslag nemen. Vansteenwegen et al. (2009) vermijden dit door voor elke opgenomen locatie *Wait* en *Maxshift* bij te houden. *Wait* vormt de wachttijd bij aankomst op de locatie vóór de start van het tijdvenster. Wanneer een

toerist aankomt tijdens het tijdvenster, bedraagt de wachttijd nul. *Maxshift* wordt gedefinieerd als de maximale tijd dat de voltooiing van een bezoek vertraagd kan worden, ervoor zorgend dat elk bezoek nog steeds toegelaten is. Dit is gelijk aan de som van *Wait* en *Maxshift* van de volgende locatie, uitgezonderd wanneer *Maxshift* beperkt wordt door het eigen tijdvenster (Vansteenwegen et al., 2009). Wanneer een extra locatie *j* tussen locatie *i* en *k* wordt toegevoegd, wordt de totale invoegtijd aangeduid met *shift<sub>j</sub>*. *Shift<sub>j</sub>* mag niet meer bedragen dan de som van *Wait<sub>k</sub>* en *Maxshift<sub>k</sub>* van locatie *k*, opdat de invoeging van locatie *j* toegelaten is. De *insert* stap bepaalt de kleinste invoegtijd voor elke locatie *i* die ingevoegd kan worden. De locatie met de hoogste ratio, dewelke de score van de locatie in verhouding tot de invoegtijd weegt ( $S_i^2/shift_i$ ), wordt geselecteerd en ingevoegd. De *insert* stap wordt herhaald totdat geen enkele locatie nog kan worden toegevoegd (Vansteenwegen et al., 2009).

De *shake* stap wordt gebruikt om te ontsnappen aan lokale optima. Een of meerdere locaties worden verwijderd in elke route. Twee integers dienen als inputfactoren voor de *shake* stap.  $R_d$  duidt aan hoeveel opeenvolgende bezoeken verwijderd moeten worden in route *d* en  $S_d$  wijst de plaats in de route aan waar het verwijderproces dient te starten. Gezien het verschil in lengte van de verschillende routes, zal  $S_d$  voor verschillende routes een andere waarde aannemen. Dit verhoogt de mogelijkheid te ontsnappen aan lokale optima. Na eliminatie worden alle locaties volgend op de weggenomen locaties verplaatst naar het begin van de route om onnodige wachttijden te vermijden. Wanneer een bezoek niet kan worden verplaatst omwille van het bijhorend tijdvenster, worden dat bezoek en eventuele daaropvolgende bezoeken niet gewijzigd (Vansteenwegen et al., 2009).

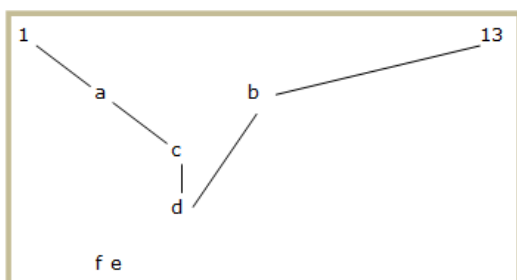
Een set van lege routes vormt het startpunt voor de heuristiek. Alle parameters behorende tot de *shake* stap wordt de waarde 1 toegekend. De heuristiek stopt wanneer gedurende een bepaald aantal keer geen verbeteringen geïdentificeerd kunnen worden. Eerst wordt de *insert* stap toegepast totdat een lokaal optimum gevonden is. Wanneer de gevonden oplossing beter is dan de huidige oplossing, wordt de oplossing bijgehouden en de parameter *R* opnieuw ingesteld op 1. Daarna wordt de *shake* stap uitgevoerd. Na elke *shake* stap wordt *S* met de waarde *R* en *R* met 1 verhoogd. Wanneer *S* groter of gelijk aan de grootte van de kleinste route is, wordt de grootte van die route afgetrokken van *S* om de nieuwe positie te bepalen. Wanneer *R* gelijk is aan  $n/(3m)$ , wordt *R* opnieuw gelijkgesteld aan 1. Iteratief wordt aldus een opeenvolging van lokale zoekoplossingen opgebouwd in plaats van willekeurige pogingen van de lokale zoekprocedure te herhalen. Met behulp van de *shake* parameters worden gedurende elke *shake* stap andere bezoeken verwijderd en wordt elk bezoek gedurende de volledige procedure minstens één keer verwijderd. De heuristiek zet de zoektocht steeds verder vanaf de huidige oplossing en keert niet telkens terug naar de tot dan toe best gevonden oplossing. De volledige oplossingsruimte wordt beter verkend en verkeerde beslissingen worden gecorrigeerd (Vansteenwegen et al., 2009).

Gavalas et al. (2012) halen twee tekortkomingen aan van de ILS methode. Stel dat er een gebied is waar veel knooppunten geconcentreerd zijn zoals het gebied met knooppunten c, d, e en f in figuur 2. De kans bestaat dat een knooppunt dat ver verwijderd is van dit gebied opgenomen wordt omdat het een hoge score heeft zoals knooppunt a in figuur 2. Omwille van de lange reisafstand naar het andere gebied worden eventueel goede kandidaten niet opgenomen. Dit kan leiden tot ongebruikt tijdsbudget. Route 1-a-b-13 in figuur 2 vormt de huidige oplossing. Oplossing 1-c-d-e-f heeft echter een hogere score en minder verspilde tijd. Voorbeelden van scores staan tussen haken bij elk knooppunt.



*Figuur 2: Tekortkoming ILS (1)*

Knooppunten of groepen van knooppunten die ver gelegen zijn van de opgenomen knooppunten worden vaak niet opgenomen. Hoewel de kans bestaat dat deze knooppunten een hoge score hebben. Wanneer de route uiteindelijk één van deze knooppunten wil opnemen, kan dit vaak niet meer omwille van bijvoorbeeld de tijdsbeperking van dat knooppunt. Figuur 3 illustreert deze situatie. ILS voegt knooppunten a, b, c en d in. Knooppunten e en f hebben een hogere score maar zij worden in eerste instantie niet opgenomen omwille van hun hoge consumptietijd. Het volgende knooppunt dat in aanmerking komt voor invoeging, nadat knooppunten a, b, c en d reeds aan de route zijn toegevoegd, is knooppunt e. Aangezien het tijdsbudget echter overschreden zou worden, kan knooppunt e niet meer worden toegevoegd.



*Figuur 3: Tekortkoming ILS (2)*

Een groep van knooppunten kan gezamenlijk voor een hogere score zorgen maar ILS examineert knooppunten individueel. Als oplossing halen Gavalas et al. (2012) aan om clusters te identificeren met een hoge gemiddelde score vooraleer knooppunten worden ingevoegd. Om de clusters te identificeren kunnen ook de afstand of de tijdvensters gebruikt worden (Gavalas et al., 2012).

## 2.4 Team oriëntering probleem met meerdere tijdvensters en meerdere beperkingen

### 2.4.1 Inleiding

Een persoonlijke toeristische route ontwerpen rekening houdend met de interesse van de toerist en verschillende beperkingen, wordt steeds belangrijker in de toeristische sector. Dit probleem is direct gerelateerd aan het *Multi-Constraint Team Orienteering Problem with Multiple Time Windows* (MCTOPMTW) (Garcia et al., 2010; Souffriau et al., 2012). In het MCTOPMTW bestaat een set van locaties waarbij elke locatie een bepaalde score, een tijdvenster en  $Z$  attributen heeft.  $Z$  beperkingen worden toegevoegd die samen met de tijdvensterbeperkingen de selectie van de verschillende locaties beperken. De toegevoegde beperkingen kunnen van verschillende aard zijn. Het kan een budgetbeperking betreffen waardoor de toegangsprijs van een locatie belangrijk wordt. Een maximum kan vastgezet worden op bijvoorbeeld het aantal te bezoeken musea per dag of het aantal te bezoeken standbeelden gedurende de volledige reis. Het doel van het MCTOPMTW is een vast aantal routes te bepalen waarbij de totale verzamelde score gemaximaliseerd wordt en geen enkele beperking overtreden wordt (Souffriau et al., 2012). In tegenstelling tot het hierboven beschreven TOPTW model van Vansteenwegen et al. (2009) laat dit probleem toe verschillende tijdvensters op verschillende dagen te definiëren alsook meer dan één tijdvenster per dag. Tricoire et al. (2010) bespraken reeds het TOP met meerdere en verschillende tijdvensters zonder extra beperkingen, zoals aangehaald in tabel 4. De toepassing van Tricoire et al. (2010) is het plannen van een individuele route voor *field workers* en voor verkoopvertegenwoordigers in de farmaceutische of voedselindustrie.

### 2.4.2 Mathematische formulering

Het MCTOPMTW kan geformuleerd worden als een integer programmeringsprobleem.  $N$  locaties  $m$  routes zijn gegeven. Elke locatie krijgt een score  $S_i$ . Het startpunt is locatie 1 en het eindpunt locatie  $N$ . Het kortste pad tussen locatie  $i$  en locatie  $j$  kan overbrugd worden in een bepaalde tijd  $t_{ij}$ .  $x_{ijd} = 1$  wanneer in route  $d$  een bezoek aan locatie  $i$  gevolgd wordt door een bezoek aan locatie  $j$  en 0 elders. Gegeven zijn  $Z$  attributen en  $W$  tijdvensters.  $y_{iwd} = 1$  als locatie  $i$  bezocht wordt in route  $d$  gedurende tijdvenster  $w$  en 0 elders.  $s_{id}$  geeft de start weer van het bezoek aan locatie  $i$  in route  $d$ .  $O_{iwd}$  en  $C_{iwd}$  zijn respectievelijk de openings- en sluitingstijden van tijdvenster  $w$  van knooppunt  $i$  in route  $d$ .  $e_{idz}$  wijst op de kost horende bij beperking  $z$  voor locatie  $i$  in route  $d$ . Het kostenbudget van beperking  $z$  wordt aangeduid met  $E_z$ .  $M$  is een constante met hoge waarde. Het doel is de som van de scores van de bezochte locaties te maximaliseren (Souffriau et al., 2012). In tabel 5 worden de gebruikte symbolen opgelijst. De reeds eerder gebruikte symbolen die in onderstaande formulering terugkeren, worden in tabel 2 en 3 weergegeven.

$Z$	attributen
$W$	tijdvensters
$y_{iwd}$	locatie $i$ in route $d$ in tijdvenster $w$
$O_{iwd}$	openingstijd tijdvenster $w$ , locatie $i$ , route $d$
$C_{iwd}$	sluitingstijd tijdvenster $w$ , locatie $i$ , route $d$
$e_{idz}$	kost beperking $z$ , locatie $i$ , route $d$
$E_z$	kostenbudget beperking $z$

Tabel 5: Symbolenlijst MCTOPMTW

De doelfunctie wordt weergegeven door (1). Beperking (2) zorgt ervoor dat alle routes starten in knooppunt 1 en eindigen in knooppunt  $N$ . Dankzij beperking (3) en (4) is de connectiviteit en de tijdlijn van elke route verzekerd. Beperking (5) garandeert dat elk knooppunt maximaal één keer bezocht wordt. Beperking (6) reduceert de selectie door de kenmerken van een knooppunt te begrenzen. Deze beperking wordt gebruikt om bijvoorbeeld een budget voorop te stellen. Beperking (7) zorgt ervoor dat de start van een bezoek binnen het tijdvenster valt (Souffriau et al., 2012). Garcia et al. (2010) formuleren het MCTOPTW. Het verschil met bovenstaande formulering bevindt zich in het aantal tijdvensters en het expliciet gebruik van een tijdsbeperking. Souffriau et al. (2012) gebruiken meerdere tijdvensters.

$$\text{Max } \sum_{d=1}^m \sum_{w=1}^W \sum_{t=2}^{N-1} S_t y_{iwd} \quad (1)$$

$$\sum_{d=1}^m \sum_{j=2}^N x_{ijd} = \sum_{d=1}^m \sum_{i=1}^{N-1} x_{ind} = m \quad (2)$$

$$\sum_{i=1}^{N-1} x_{ikd} = \sum_{j=2}^N x_{kj d} = \sum_{w=1}^W y_{kwd}; \forall k = 2, \dots, N-1; \forall d = 1, \dots, m \quad (3)$$

$$s_{id} + t_{ij} - s_{jd} \leq M(1 - x_{ijd}); \forall i, j = 1, \dots, N; \forall d = 1, \dots, m \quad (4)$$

$$\sum_{d=1}^m \sum_{w=1}^W y_{iwd} \leq 1; \forall i = 1, \dots, N \quad (5)$$

$$\sum_{d=1}^m \sum_{w=1}^W \sum_{i=1}^N e_{idz} y_{iwd} \leq E_z; \forall z = 1, \dots, Z \quad (6)$$

$$\exists w \in 1, \dots, W; O_{iwd} \leq s_{id} \leq C_{iwd}; \forall i = 1, \dots, N; \forall d = 1, \dots, m \quad (7)$$

$$x_{ijd}, y_{iwd} \in \{0,1\}; \forall i, j = 1, \dots, N; \forall w = 1, \dots, W; \forall d = 1, \dots, m \quad (8)$$

### 2.4.3 Oplossingsmethoden

Garcia et al. (2010) maken gebruik van een *iterated local search* metaheuristiek om het MCTOPTW op te lossen. Zo kunnen persoonlijke toeristische routes ontworpen worden in realtime. Het algoritme is gebaseerd op het algoritme van Vansteenwegen et al. (2009), besproken in deelsectie 2.3.3. Garcia et al. (2010) hebben enkele elementen aangepast. Ten eerste moet de controle of een knooppunt daadwerkelijk kan worden toegevoegd, gewijzigd worden. In een TOPTW wordt enkel gecontroleerd of de invoeging van een knooppunt geen afbreuk doet aan het niet overschrijden van het tijdsbudget van de toerist. Nu moeten ook de extra attribuutbeperking geverifieerd worden. Deze controle gebeurt eerst omdat de tijdscontrole qua berekeningstijd duurder is. De tweede wijziging heeft betrekking op de ratio die in de *insert* stap gebruikt wordt om de locatie te bepalen die wordt toegevoegd aan de route. Aangezien in de ratio geen rekening gehouden wordt met de attribuutbeperkingen, dient deze worden aangepast. Volgens Garcia et al. (2010) is volgende ratio het beste alternatief:



$$ratio_i = \frac{S_i^2}{\frac{shift_i}{availableTime} + \sum_{k=1}^K \frac{e_{ik}}{available_k}}$$

met  $S_i$  de score van knooppunt  $i$ ,  $shift_i$  de kleinste invoegtijd van knooppunt  $i$ ,  $availableTime$  de beschikbare tijd,  $e_{ik}$  de waarde van attribuutbeperking  $k$  geassocieerd met knooppunt  $i$ ,  $K$  het aantal attribuutbeperkingen en  $available_k$  de nog beschikbare hoeveelheid van attribuutbeperking  $k$ . In de ratio wordt een bepaald gewicht aan elke attribuutbeperking gegeven en de nog beschikbare hoeveelheid van elke beperking in de huidige oplossing opgenomen. Op die manier krijgt het verbruik van de meest bindende beperking meer belangstelling. De invoegtijd is nu even belangrijk als de attribuutbeperkingen samen. Meer attribuutbeperkingen zullen het totale gewicht van de noemer niet verhogen. Ten derde stellen Garcia et al. (2010) een taboelijst op. In het algoritme van Vansteenwegen et al. (2009) kan een locatie die verwijderd is in een iteratie in de volgende iteratie worden toegevoegd. Deze locatie is vaak waardevol. Garcia et al. (2010) proberen te vermijden dat dezelfde locatie in opeenvolgende iteraties wordt verwijderd. Locaties op de taboelijst komen niet in aanmerking voor eliminatie. De laatste wijziging heeft betrekking op de stopregel. Het maximum aantal iteraties zonder verbetering is nu probleemafhankelijk (Garcia et al., 2010).

Het algoritme dat Souffriau et al. (2012) gebruiken om het MCTOPMTW te behandelen, is eveneens gebaseerd op de iteratieve lokale zoekprocedure van Vansteenwegen et al. (2009). Souffriau et al. (2012) gebruiken een combinatie van de iteratieve lokale zoekprocedure en de *greedy randomized adaptive search procedure* (GRASP). De GRASP voegt herhaaldelijk locaties toe aan de huidige oplossing. Steeds wordt nagegaan of de oplossing toegelaten is. Een hebzucht parameter wordt gebruikt die kan variëren van 0.60, willekeurig, tot en met 0.89, één van de betere lokale zoektochten. Een efficiënt mechanisme evalueert de  $Z$  beperkingen. Door efficiënte waardecirculatie wordt een structuur van mogelijke naburige locaties voor elke route onderhouden. Als basis voor het berekenen van de heuristische waarde van de invoeging van een knooppunt in de route gebruiken Souffriau et al. (2012) de speling van de  $Z$  beperkingen.

Sylejmani et al. (2012) stellen een *tabu search* heuristiek voor om het MCTOPTW op te lossen. Om de buurt te doorzoeken, worden *Insert*, *Replace* en *Swap* gebruikt. Het algoritme maakt eveneens gebruik van een taboelijst, een alarm- en een diversificatiemechanisme. De taboelijst bestaat uit bewegingen die verboden zijn gedurende een aantal iteraties. Hierdoor wordt de afwisseling van zoekprocessen gestimuleerd in delen van de zoekruimte die nog niet verkend zijn. De *Insert* operator voegt een nieuw knooppunt toe, de *Replace* operator vervangt een opgenomen knooppunt door een nieuw knooppunt en de *Swap* operator verwisselt twee opgenomen knooppunten. Voor elke harde beperking zoals de duur van de route of het budget wordt een efficiënte afwijkende/toenemende functie opgesteld. Zo wordt nagegaan of de beweging toegelaten is. Om te controleren of voldaan is aan de tijdvensterbeperkingen worden de parameters *Max Forward Shift* ( $MFS_i$ ) en *Max Backward Shift* ( $MBS_i$ ) geregistreerd voor elk knooppunt in de route. Deze parameters duiden aan hoe

veel knooppunt  $i$  in route  $m$  respectievelijk vooruit en achteruit verschoven kan worden. Er wordt geen wachttijd verondersteld voor de start van een bezoek. Het algoritme voorgesteld door Sylejmani et al. (2012) werkt als volgt. Eerst wordt een willekeurige oplossing gecreëerd waarna het algoritme het maximum aantal iteraties uitvoert. In elke tweede iteratie wordt gebruik gemaakt van de *Replace* operator. In elke andere iteratie wordt afwisselend gebruik gemaakt van de *Insert* en de *Swap* operator. Gedurende elke iteratie wordt zowel gezocht naar de beste taboe oplossing als naar de beste niet-taboe oplossing. Wanneer de beste niet-taboe oplossing beter is dan de huidige oplossing, wordt dit de huidige oplossing. Wanneer dit niet zo is, wordt nagegaan of de beste taboe oplossing beter is dan de huidige oplossing (Sylejmani et al., 2012).

De spreiding van het zoekproces wordt mogelijk gemaakt door vier operatoren, namelijk een *Delete*, *Perturbate*, *Restart* en *Penalise* operator. De *Delete* operator verwijdert twee knooppunten waarvan de tijdvensterbeperkingen het meest beperkend zijn, te vinden aan de hand van de  $MFS_i$  en  $MBS_i$  waarden. Het *Perturbate* proces houdt in dat na het toepassen van de *Delete* operator op de beste huidige oplossing, de zoektocht verder gezet wordt vanaf die nieuwe gevonden oplossing. Soms wordt de zoektocht opnieuw gestart vanaf een willekeurige oplossing (*Restart*). Het algoritme beschikt over twee taboelijsten. Het *recency* geheugen wordt gebruikt om het iteratienummer te registreren. De grootte van de taboelijst bepaalt hoe lang een bepaalde beweging op de taboelijst moet blijven. Het *frequency* geheugen telt het aantal keer dat een bepaalde beweging wordt uitgevoerd. Vaak gebruikte bewegingen worden gestraft door het verdubbelen van hun verblijftijd op de taboelijst (*Penalise*). Het algoritme stopt wanneer het aantal opeenvolgende iteraties zonder verbetering 30% van het maximum aantal iteraties overschrijdt of wanneer het maximum aantal iteraties bereikt wordt (Sylejmani et al., 2012).

## 2.5 Varianten

Tabel 6 geeft een olijsting van verschillende varianten op het *tourist trip design* probleem die hieronder kort worden besproken.

<b>Probleem</b>	<b>Auteur</b>
<i>Orienteering problem with compulsory vertices</i>	Gendreau et al. (1998b)
<i>Time dependent orienteering problem</i>	Fomin en Lingas (2002)
<i>Travelling Salesman problem with profits</i> <ul style="list-style-type: none"> <li>- <i>Orienteering problem</i></li> <li>- <i>Prize collecting travelling salesman problem</i></li> <li>- <i>Profitable tour problem</i></li> </ul>	Feillet et al. (2005)
<i>Mixed orienteering problem</i>	Muyldermans et al. (2005)
<i>Generalised orienteering problem</i>	Zong et al. (2005) Wang et al. (2008)
<i>Multi-objective orienteering problem</i>	Schilde et al. (2009)
<i>Undirected capacitated arc routing problem with profits</i>	Archetti et al. (2010)

Tabel 6: Varianten TTDP

Het *oriëntering* probleem (OP) kan gelinkt worden aan het *travelling salesman problem with profits* (TSPP). Laatstgenoemd probleem heeft twee tegengestelde objectieven, namelijk het maximaliseren van de winst en het minimaliseren van de reiskosten. Wanneer dit laatste objectief geformuleerd wordt als een beperking, komt het OP tot stand. Wanneer de eerste doelstelling als een beperking beschouwd wordt, wordt gesproken over het *prize collecting travelling salesman problem* (PCTSP). Wanneer de twee objectieven gecombineerd worden in één doelstelling, namelijk het maximaliseren van de winst verminderd met de reiskosten, wordt het probleem benoemd als het *profitable tour problem* (PTP) (Feillet et al., 2005).

Het *generalised orienteering problem* (GOP) (Zong et al., 2005; Wang et al., 2008) heeft een andere doelfunctie dan het OP. Elk knooppunt krijgt nu een set van scores. In dit probleem kan een combinatie van knooppunten ervoor zorgen dat een hogere of lagere score toegewezen wordt aan de doelfunctie dan de som van de individuele scores. Bezienswaardigheden die tot hetzelfde genre behoren, kunnen gezamenlijk een hogere score teweegbrengen dan de som van hun individuele scores. Eveneens kan een attractie die op zich weinig waarde voor de toerist heeft in combinatie met een andere attractie heel interessant zijn. Wang et al. (2008) lossen het GOP op met een genetisch algoritme.

Wanneer bepaalde POI zeker opgenomen moeten worden in de toeristische route, kan een *orienteering problem with compulsory vertices* (OPCV) gemodelleerd worden. Zo mist de reiziger geen enkele topattractie (Gendreau et al., 1998b). Met een *branch-and-cut* methode lossen Gendreau et al. (1998b) dit probleem op.

Elk POI in een stad beschikt over verschillende voordelen voor verschillende categorieën. Een toerist kan bijvoorbeeld in de categorie cultuur een voorkeur hebben voor het bezoeken van kerken terwijl een andere toerist voorkeur geeft aan het bezoeken van musea (Schilde et al., 2009). Schilde et al. (2009) classificeren dit probleem als het *multi-objective orienteering problem* (MOOP). Om alle optimale oplossingen te bepalen, worden twee heuristische oplossingsmethoden gebruikt: het *pareto ant colony* optimalisatie algoritme en de uitbreiding van de *variable neighborhood search* (VNS) methode naar de variant met meerdere objectieven. De twee methoden worden verenigd met behulp van *path relinking* procedures (Schilde et al., 2009).

Archetti et al. (2010) behandelen het *undirected capacitated arc routing problem with profits* (CARPP). Aan elke boog wordt nu een score toegekend. Een mogelijke toepassing is het ontwikkelen van fietsroutes.

Aangezien de naam GOP reeds in gebruik genomen is (Wang et al., 2008), stellen Vansteenwegen et al. (2011c) voor om de combinatie van het *arc routing problem with profits* met het OP als het *mixed orienteering problem* (MOP) te benoemen. In dit probleem worden aan zowel de knooppunten als de bogen scores gegeven. Het MOP kan als toerismetoepassing dienen wanneer een activiteit zich niet op één bepaalde plaats bevindt zoals een wandeling door het park of door de winkelstraat (Muyldermans et al., 2005).

Fomin en Lingas (2002) introduceren het *time dependent orienteering problem* (TDOP). Dit probleem incorporeert de tijdsafhankelijkheid van de berekeningen aangaande de verbindingen, i.e. de reistijd tussen de knooppunten. Via deze modellering kunnen verschillende vervoerswijzen zoals het openbaar vervoer in rekening worden gebracht.

## 2.6 Hotelselectie

Naast het plannen van een toeristische route kan ook het selecteren van een hotel belangrijk zijn. Het *tour planning* probleem (TPP) is gerelateerd aan het rittenplanningsprobleem. Zhu et al. (2010) formuleren het TPP als een gemengd geheelgetalig lineair programmeringsprobleem. Het objectief bestaat uit het selecteren van een geschikt hotel en het bepalen van toeristische routes waarbij de totale nutswaarde wordt gemaximaliseerd. De routes moeten ontworpen worden zodat de gewenste plaatsen bezocht worden en geen budget- en/of tijdsbeperkingen overschreden worden.

Zhu et al. (2010) presenteren een algoritme opgedeeld in twee procedures om het TPP op te lossen. De eerste procedure omvat het selecteren van het hotel en de tweede procedure het plannen van de toer. Alle hotels worden opgedeeld in een bepaald aantal groepen, bijvoorbeeld naar regio. Het maximum aantal hotels in een groep wordt beperkt door een vooropgestelde limiet. Wanneer deze limiet overschreden wordt, zal de groep verder opgesplitst worden. Hierna wordt willekeurig een hotel uit de groep gekozen waarop de tweede procedure, de routeplanning, wordt toegepast. De route die geassocieerd wordt met dit hotel wordt gezien als een oplossing. Daarna wordt de groep gekozen waarin de route met de hoogste nutswaarde zich bevindt. Op elk hotel in deze groep wordt de tweede procedure toegepast. De uiteindelijke oplossing zal de route met de hoogste nutswaarde zijn met het bijhorende hotel.

De tweede procedure start met een initiële oplossing. Hierop worden twee procedures toegepast met als doel tijdsreductie en nutsverhoging. Zolang geen betere oplossing gevonden wordt na een vast aantal opeenvolgende operaties, wordt de procedure verder uitgevoerd. Na de initialisatie volgt de tijdsreductie. Deze maakt gebruik van vier standaard routines: 2-opt\*, Move, 2-opt en Or-opt. Bij de 2-opt\* routine worden twee verbindingen in twee verschillende routes verwijderd en vervangen door twee nieuwe verbindingen. De Move routine zorgt ervoor dat één, twee of drie opeenvolgende plaatsen in een route verplaatst worden naar een andere route. De 2-opt routine verwijdert twee verbindingen van dezelfde route en vervangt deze door twee nieuwe verbindingen zodat één nieuwe route gecreëerd wordt. Bij de Or-opt routine worden één, twee of drie opeenvolgende plaatsen in een route verplaatst naar een andere locatie in dezelfde route. Om de totale nutswaarde van de reis te verhogen, worden twee routines gebruikt: Insert en Replace. De Insert routine zorgt ervoor dat nieuwe locaties in de route worden opgenomen. De Replace routine vervangt een plaats in de huidige route door een niet-opgenomen plaats. Dankzij de Remove routine kan ontsnapt worden aan lokale optima (Zhu et al., 2010).

Castro et al. (2013) bespreken het TSP met hotelselectie. Het TSP met hotelselectie is reeds geïntroduceerd door Vansteenwegen et al. (2011a, in Castro et al., 2013). Een verkoper kan vaak niet al zijn klanten op één dag bezoeken aangezien hij slechts een beperkt aantal uren per dag kan werken. De verkoper moet bepalen in welke volgorde hij zijn klanten zal bezoeken. Hij moet voor elke nacht tevens een hotel selecteren. Wanneer de verkoper op een bepaalde dag eindigt in een bepaald hotel dan dient hij de volgende dag van datzelfde hotel te vertrekken. Elke dag moet beginnen en eindigen in één van de beschikbare hotels. De doelstelling betreft het minimaliseren van het vereist aantal dagen om de taak te vervullen en de totale reisafstand (Castro et al., 2013).

Vansteenwegen et al. (2011a, in Castro et al., 2013) vermelden enkele toepassingen van TSP met hotelselectie. Verkopers hebben soms meerdere dagen nodig om hun volledig klantenbestand te bezoeken. Postbodes willen hun ronde opsplitsen in een aantal verbonden kleinere rondes om het gewicht van hun postzak te beperken. Vrachtwagenbestuurders zijn vaak meerdere dagen onderweg. Elke dag start en eindigt dan op een parkeerplaats. Sommige toeristen gaan op rondreis waarbij nood is aan afwisseling van hotels omdat ze verschillende delen van een bepaalde regio willen ontdekken. Een laatst aangehaalde toepassing is het bepalen van routes voor elektronische voertuigen waarbij de volledige route opgedeeld wordt in kleinere ritten die beperkt zijn in lengte. Dit gebeurt omwille van de batterijcapaciteit zodat nu de batterij opgeladen kan worden op elke tussenstop.

De metaheuristiek, ontwikkeld door Castro et al. (2013), omvat een memetisch algoritme inclusief *tabu search* dat zowel hotelselectie als routeplanning behandelt. De hotelselectie bestaat uit het bepalen van tussentijdse hotels waar zowel het aantal bezoeken aan tussentijdse hotels alsook de reistijd geminimaliseerd worden. Op deze manier wordt de volgorde van hotels bepaald. Elk hotel kan meer dan één keer voorkomen. De hotelselectie heeft een grote impact op de uiteindelijke oplossing aangezien het de richting van de volledige toer bepaalt alsook de manier waarop de dagelijkse ritten worden opgebouwd. Een memetisch algoritme wordt gebruikt voor hotelselectie. De routeplanning wordt opgesteld met behulp van *tabu search* (Castro et al., 2013).

## 2.7 Conclusie

Het MCTOPMTW sluit nauw aan bij de centrale onderzoeksvraag. Een toerist bezoekt een regio gedurende meerdere dagen. Voor elke dag wordt een route uitgestippeld waardoor de toerist in staat is bepaalde bezienswaardigheden te bezoeken. Bij het bepalen van de routes wordt met verschillende elementen rekening gehouden: de reisbeperkingen van de toerist onder meer de duur van de reis en de kostprijs, de persoonlijke kenmerken en voorkeuren van de toerist zoals een voorliefde voor architectuur en de tijdvensters van de bezienswaardigheden. In hoofdstuk 3 wordt een fictieve reis ontwikkeld voor een toerist die de stad Hasselt bezoekt.

## Hoofdstuk 3      Praktijkstudie

In het inleidend deel van hoofdstuk 3 worden zowel het praktijkprobleem als de bestaande mobiele applicaties kort toegelicht. Sectie 3.2 bevat uitleg over de verschillende bezienswaardigheden in Hasselt die gebruikt worden in het praktijkvoorbeeld. In sectie 3.3 worden gegevens van een toerist voorgesteld waarna in sectie 3.4 een score voor elk *point of interest* (POI) berekend wordt. Die score geeft de interesse van de toerist in dat POI weer. Sectie 3.5 introduceert het praktijkprobleem als een *tourist trip design* probleem (TTDP). In deze sectie wordt het probleem mathematisch geformuleerd en opgelost met behulp van een algoritme. De uiteindelijke oplossing wordt voorgesteld in sectie 3.6.

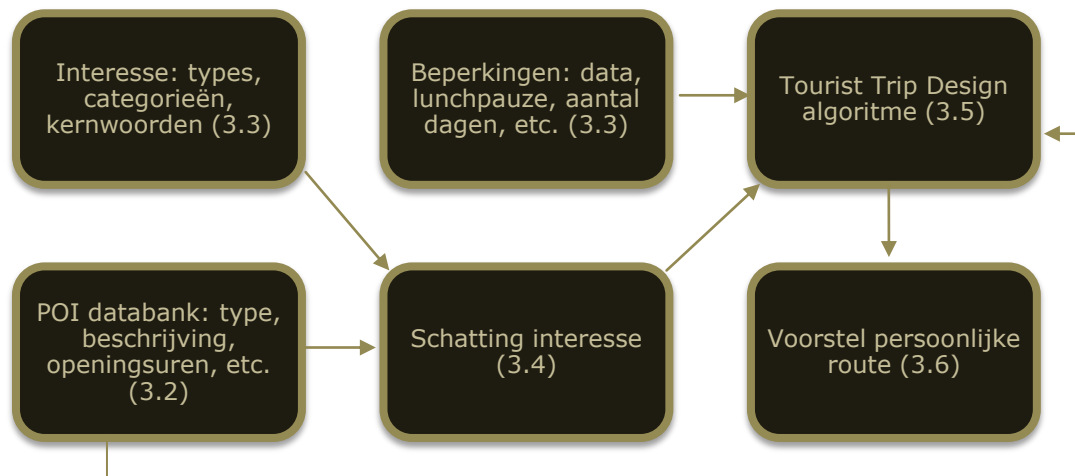
### 3.1 Inleiding

Uit de literatuurstudie blijkt dat het probleem voor toeristen die één of meerdere dagen een bestemming bezoeken uit verschillende delen bestaat. Ze moeten namelijk de meest interessante POI selecteren en een efficiënte bezoeksvolgorde voor deze POI bepalen. Dit probleem wordt geclassificeerd als een TTDP. Verschillende beperkingen zoals de bezoektijd aan elk POI, de openings- en sluitingsuren, de afstand en/of reistijd tussen de POI, de beschikbare tijd van de toerist en de voldoening die de toerist vindt in het bezoeken van een bepaald POI, bemoeilijken het oplossen van het probleem (Gavalas et al., 2012). De toeristische dienst in een bepaalde stad kan de toerist bijstaan in het oplossen van zijn toeristisch probleem. Aan de hand van het profiel en de beperkingen van de toerist gecombineerd met de kennis die de werknemers op de toeristische dienst vergaard hebben over de lokale attracties, stelt de toeristische dienst een gepersonaliseerde route op voor de toerist. Deze gepersonaliseerde route houdt echter geen rekening met nieuwe omstandigheden die zich tijdens het uitvoeren van de route kunnen voordoen. Stel dat de toerist langer in het Modemuseum te Hasselt vertoeft dan oorspronkelijk beoogd was. Hierdoor is de planning van de resterende route niet meer correct. De toerist moet ofwel terug naar de toeristische dienst ofwel zich bevragen aan lokale voorbijgangers (Garcia et al., 2010).

Een *personalized electronic tourist guide* (PET) of *mobile tourist guide* (MTG) (Garcia et al., 2010) kan de taak van de toeristische dienst vervangen. Dit is een klein elektronisch apparaat dat in de hand past. MTG worden gebruikt als hulpmiddel om oplossingen te vinden voor TTDP. Aan de hand van de persoonlijke interesse en voorkeuren van de toerist, voldoende recente informatie over de bezienswaardigheden en reisinformatie zoals datum van aankomst en vertrek, kan een MTG een optimale selectie van attracties alsook een route tussen deze attracties bepalen (Gavalas et al., 2012). Een MTG zorgt ervoor dat de route zich aanpast aan nieuwe omstandigheden in realtime. Hierdoor kan de toerist zijn route in enkele seconden opnieuw voortzetten (Garcia et al., 2010). Met de hulp van een MTG kan de spreiding van toeristen verbeterd worden. Dit is vooral belangrijk voor topattracties aangezien deze een massa toeristen aantrekken (Kramer et al., 2006). Enkele voorbeelden

van ontwikkelde MTG zijn: Cyberguide, Gulliver's Genie, HIPPIE, Hyper-Audio, SaiMotion, GUIDE, eNarro en Dynamic Tour Guide (DTG) (Souffriau et al., 2008). Ter illustratie wordt de werking van de DTG kort toegelicht. Tijdens de rondreis wordt de reiziger geleid naar *Tour Building Blocks* (TBB). Dit kunnen bezienswaardigheden of restaurants zijn. Navigatiesoftware bijvoorbeeld *Mappoint* of *Navigon* wordt gebruikt. Wanneer de reiziger begint te wandelen, berekent de DTG zijn wandelsnelheid. De reiziger krijgt inleidende informatie wanneer hij in de buurt van een TBB komt. Om de reiziger niet te verwarren, wordt rekening gehouden met de kant waarvan hij komt. Wanneer de reiziger beslist om de bezienswaardigheid grondiger te bezoeken, wordt additionele informatie verschaft. Als hij zijn rondreis verder zet, wordt hij geleid naar de volgende TBB. Van het moment dat de reiziger een TBB langer dan gepland bezoekt, te ver van het oorspronkelijke pad afwijkt of zijn wandelsnelheid aanpast, wordt de rondreis herberekend. Wanneer hij onderweg iets tegenkomt wat hem ook interesseert, wordt de route onderbroken en zoekt de DTG naar beschikbare informatie over dat nieuwe item. Indien geen verdere informatie beschikbaar is, wacht de DTG en gaat hij verder wanneer de reiziger zijn route verder zet (Kramer et al., 2006).

Verschillende web- en mobiele applicaties zoeken aldus naar oplossingen voor het TTDP (Gavalas et al., 2012). Vansteenwegen et al. (2011b) introduceren een expertsysteem voor toeristen, genaamd de *City Trip Planner*. Het is een webapplicatie die toelaat routes te plannen voor vijf steden in België, meer bepaald Antwerpen, Brugge, Gent, Leuven en Mechelen. De *City Trip Planner* houdt rekening met de interesse en beperkingen van de gebruiker. Het gebruikersprofiel wordt gekoppeld aan een databank met locaties om de persoonlijke interesse van de gebruiker in de locaties te voorspellen. Het expertsysteem is in staat bezoeken te plannen voor meerdere dagen. Voor elk POI kunnen meerdere tijdvensters in acht genomen worden die van dag tot dag kunnen verschillen. Het systeem heeft het vermogen rekening te houden met lunchpauzes en voorgestelde verplichte POI door de toeristische dienst. De *City Trip Planner* integreert de selectie van de POI met het bepalen van een route tussen deze. Een overzicht van het expertsysteem wordt afgebeeld in figuur 4. Eerst worden de beperkingen en interesses van de toerist verzameld zodat voor elk POI een persoonlijke score kan worden bepaald. Het TTDP wordt opgelost met een heuristisch resulterend in een persoonlijke route rekening houdend met de interesse van de gebruiker, huidige locatie, bestemming, beschikbare tijd en openingsuren. Uiteindelijk kan het bekomen traject gedownload worden naar een GPS navigatiesysteem (Vansteenwegen et al., 2011b). In de volgende secties wordt dieper ingegaan op de verschillende stappen van het expertsysteem.



Figuur 4: Overzicht expertsysteem (Vansteenwegen et al., 2011b)

### 3.2 POI databank Hasselt

België biedt veel mogelijkheden aan zowel buitenlandse als binnenlandse toeristen. Om te ontsnappen aan de dagelijkse sleur, verkiezen de meeste toeristen de kust. Als ze meer interesse hebben voor een ontdekkingsstrip, krijgen de Ardennen de voorkeur. Enkele andere redenen waarom een toerist naar België wil komen, zijn: kunst, cultuur, architectuur, gastronomie, culturele evenementen en het uitgaansleven ("Toerisme", 2010). Limburg staat bekend als de groenste provincie van Vlaanderen. Hasselt, de Limburgse hoofdstad, staat bekend als de Hoofdstad van de Smaak vanwege de Hasseltse jenever, speculaas en chocolade. Cultuurhuizen, musea, kunstgalerijen en internationale cultuurevenementen zorgen voor het culturele aanbod in deze stad. Voor toeristen die van shoppen houden, zijn er boetieks, designwinkels, ketens en juwelierszaken. Bourgondiërs kunnen terecht in champagnebars en restaurants. Toeristen die de groene kant van Hasselt willen ontdekken, kunnen de pittoreske steegjes, verborgen pleinen en de groene rand van Hasselt bezoeken ("Hasselt Ontdekkingsgids 2013", 2013).

Het resterende gedeelte van deze sectie bevat een korte omschrijving van bepaalde bezienswaardigheden in Hasselt, gebaseerd op informatie van de website van Toerisme Hasselt en de brochure 'Bezienswaardig Hasselt & Zonhoven'. Aangezien de toeristische dienst niet beschikt over een POI databank, wordt handmatig een tabel opgesteld voor de gebruikte POI. Voor elk POI worden, indien beschikbaar, volgende gegevens opgenomen: naam, locatie, openings- en sluitingstijden, gemiddelde bezoektijd, type, kostprijs en kaartnummer. Om de omvang en de complexiteit van het praktijkprobleem te beperken, worden twaalf POI willekeurig gekozen uit een volledige lijst van bezienswaardigheden. De gekozen POI zijn het Literair museum, het Nationaal Jenevermuseum, de Beiaardtoren, het Modemuseum Hasselt, de Japanse tuin, het Oud-Kerkhof, de Virga Jessebasiliek, Park Natuur en Cultuur, Plopsa Indoor, het Kapermolenpark, Kinopolis en Z33. Hotel Radisson Blu wordt



verondersteld de verblijfplaats van de toerist te zijn, gekozen op aanraden van de toeristische dienst.

In het Nationaal Jenevermuseum, een authentieke stokerij uit de 19<sup>e</sup> eeuw, wordt Belgische jenever bereid. In het voorjaar van 2013 kan de tentoonstelling 'Alcoholsmokkel en sluikstokerij in de Lage Landen' bezocht worden. Deze tentoonstelling belicht het illegaal stoken en grensoverschrijdend smokkelen van alcohol. Het Nationaal Jenevermuseum organiseert verschillende activiteiten, waaronder een museumbezoek, een stadswandeling, een initiatie jeneverdegustatie en een proeverij waar jenever en chocolade elkaar aanvullen. In het Modemuseum Hasselt worden specifieke aspecten uit de modewereld uiteengezet aan de hand van tentoonstellingen. Bovendien is er ook een permanente collectie. In het voorjaar van 2013 staat het Modemuseum in het teken van Axelle Red. Deze expositie is geïnspireerd op haar liefde en passie voor mode, muziek en creativiteit. In het najaar van 2013 wordt de Italiaanse mode belicht. In het Literair museum staan boeken, illustraties en lezen centraal. Verschillende tentoonstellingen, zoals 'Van schrijver tot lezer', 'Giftige appels op gouden bordjes', een tentoonstelling over steenkoolmijnen of Limburgse jeugdauteurs en illustratoren, kunnen bezocht worden al dan niet in combinatie met een algemene rondleiding. De Beiaardtoren bevindt zich in de Sint-Quintinskathedraal, waar uurwerken, klokken en beiaarden centraal staan.

Het Oud Kerkhof toont grafmonumenten en de kapel waar geschiedenis, gewoonten en zeden van de begraafplaats kenbaar gemaakt worden. In de Virga Jessebasiliek bevinden zich het Onze-Lieve-Vrouw-Virga-Jessebeeld, een witmarmeren altaar en twee graftomben van abdisen.

De Z33 vormt het huis voor actuele kunst waar multidisciplinaire projecten worden afgewisseld met tentoonstellingen. In de Zebracinema, onderdeel van de Z33, worden niet-commerciële films gedraaid. De nieuwste films worden dagelijks gedraaid in Kinopolis.

De Japanse tuin, de grootste van Europa, belichaamt innerlijke rust en natuurlijke schoonheid. Centraal in de tuin bevindt zich het ceremoniehuis. Naast een rondleiding wordt eveneens een theedemonstratie aangeboden. In het Park Natuur en Cultuur Hasselt komen kunst en biodiversiteit samen. Buxussoorten, snoeivormen, internationale kabouters en oude ambachten zoals wissien, rotan vlechten en glasmoozaïek worden tentoongesteld.

Activiteiten in het stedelijk zwembad, openluchtzwembad en skatepark Kapermolen worden als sportief en avontuurlijk aanzien. In skatepark Kapermolen kunnen jongeren terecht met skateboards, skeelers of BMX. Plopsa Indoor Hasselt is een overdekt themapark met een vernieuwde buitenzone waar Studio-100 figuren met attracties gecombineerd worden.

Complementair aan bovenstaande beschrijvingen toont tabel 7 de overige gegevens van de POI. De locatie wordt weergegeven door de breedte- en lengtegraad. Het tijdvenster omvat de openingsdagen alsook de openings- en sluitingsuren op die dagen. Plopsa Indoor heeft geen

vaste openings- en sluitingsuren. Daarom wordt de lezer voor deze informatie doorverwezen naar de desbetreffende website (<http://www.plopsa.be/plopsa-indoor-hasselt/nl/kalender>). De gemiddelde bezoektijd wordt bepaald aan de hand van een schatting. Elk POI behoort tot exact één type dat aanduidt over wat voor soort bezienswaardigheid het gaat. Volgende types worden onderscheiden: abdij, begijnhof, kasteel, kerk, kerkhof, museum en tentoonstelling, onderwijs, park, sport en spel, theater en cinema, tuin én wetenschap en techniek (Vansteenwegen et al., 2011b; Toerisme Hasselt, 2013). De kostprijs wordt weergegeven in euro. Bijlage 1 toont het stadsplan van Hasselt. Het kaartnummer in tabel 7, inclusief de kleur tussen haken, duidt aan waar het POI gelegen is.

Volgende interessecategorieën worden onderscheiden: architectuur, culinair, cultuur, erfgoed, jongeren en kinderen, kunst, natuur, sport en uitgaan (Toerisme Hasselt, 2013). Wanneer twee verschillende POI op eenzelfde categorie goed scoren, kunnen zij beschreven worden aan de hand van gelijkaardige kenmerken. Een POI behoort tot exact één type maar kan op verschillende categorieën goed scoren. Elk POI krijgt een rating van 0 tot en met 3 op basis van de mate van toebehoren aan een bepaalde categorie. De waarde 0 staat voor het helemaal niet toebehoren aan een bepaalde categorie en de waarde 3 voor het volledig toebehoren aan een bepaalde categorie (Vansteenwegen et al., 2011b). Tabel 8 toont de score van elk POI op elke categorie.

Naam	Locatie		Tijdvenster		
	Breedtegraad	Lengtegraad	Dag	Openingsuur	Sluitingsuur
Hotel Radisson Blu	50°55'42.15"N	5°20'25.49"O	-	-	-
Literair museum	50°55'50.51"N	5°19'47.66"O	woensdag - zaterdag	14u00	17u00
Nationaal Jenevermuseum	50°55'57.91"N	5°20'26.88"O	dinsdag - zondag	10u00	17u00
Beiaardtoren	50°55'47.59"N	5°20'19.76"O	maandag - vrijdag	10u00	17u00
Modemuseum Hasselt	50°56'0.50"N	5°20'15.06"O	dinsdag - zondag	10u00	17u00
Japanse tuin	50°55'42.61"N	5°21'33.03"O	dinsdag - vrijdag	10u00	17u00
			zaterdag - zondag	14u00	18u00
Oud-Kerkhof	50°56'41.21"N	5°20'37.31"O	maandag - zondag	10u00	16u00
Virga Jessebasiliek	50°55'47.22"N	5°20'8.30"O	maandag - zondag	9u00	18u00
Park Natuur en Cultuur	50°55'27.62"N	5°17'32.35"O	maandag - zondag	10u00	18u00
Plopsa Indoor	50°56'3.12"N	5°21'39.55"O	<a href="http://www.plopsa.be/plopsa-indoor-hasselt/nl/kalender">http://www.plopsa.be/plopsa-indoor-hasselt/nl/kalender</a>		
Kapermolenpark	50°56'4.25"N	5°21'11.58"O	maandag - zondag	8u00	22u00
Kinepolis	50°55'49.94"N	5°22'7.61"O	maandag - zondag	13u30	00u45
Z33	50°55'53.41"N	5°20'24.84"O	dinsdag - zaterdag	11u00	18u00
			zondag	14u00	17u00

Tabel 7: POI gegevens

Naam	Gemiddelde bezoektijd (minuten)	Type	Kostprijs (euro)	Kaartnummer
Hotel Radisson Blu	-	-	-	3 (paars)
Literair museum	90	Museum en tentoonstelling	2,50	1 (rood)
Nationaal Jenevermuseum	90	Museum en tentoonstelling	4,50	3 (rood)
Belaardtoren	90	Museum en tentoonstelling	1,50	4 (rood)
Modemuseum Hasselt	90	Museum en tentoonstelling	5,00	5 (rood)
Japanse tuin	90	Tuin	5,00	3 (oranje)
Oud-Kerkhof	120	Kerkhof	0,00	5 (oranje)
Virga Jessebasiliek	20	Kerk	0,00	9 (oranje)
Park Natuur en Cultuur	120	Park	4,50	10 (oranje)
Plopsa Indoor	300	Sport en spel	18,00	4 (groen)
Kapermolenpark	120	Sport en spel	0,00	5 (groen)
Kinopolis	150	Theater en cinema	7,30	6 (groen)
Z33	90	Museum en tentoonstelling	0,00	13 (groen)

Tabel 7 (vervolg): POI gegevens

POI	Categorie												
	Literair museum	Nationaal Jenevermuseum	Belaardtoren	Modemuseum Hasselt	Japanse tuin	Oud-Kerkhof	Virga Jesse basiliek	Park Natuur en Cultuur	Plopsa Indoor	Kapermolenpark	Kinopolis	Z33	
Architectuur	1	1	2	1	1	2	3	0	0	0	0	1	
Culinair	0	2	0	0	1	0	0	0	0	0	0	0	
Cultuur	3	3	3	3	3	1	1	3	0	0	0	3	
Erfgoed	1	1	3	1	2	3	3	0	0	0	0	0	
Jongeren en kinderen	3	0	0	0	1	0	0	2	3	3	3	0	
Kunst	2	2	2	2	1	1	1	1	0	1	0	3	
Natuur	0	0	0	0	3	1	0	3	0	2	0	0	
Sport	0	0	0	0	0	0	0	0	1	3	0	0	
Uitgaan	0	0	0	0	0	0	0	0	0	0	3	1	

Tabel 8: Score POI per categorie

### 3.3 Persoonlijke data

De persoonlijke data bestaan uit de reisbeperkingen en interesses van de toerist, zoals weergegeven in figuur 4. De *City Trip Planner* van Vansteenwegen et al. (2011b) vergaart eerst informatie over de reisbeperkingen. De toerist kiest een stad en bepaalt de aankomstdatum en het aantal reisdagen. Daarna beslist hij voor elke dag de begin- en eindlocatie en de start- en stoptijd. In de applicatie kunnen de begin- en eindlocatie gekozen worden van een lijst van locaties. De toerist kan ook een adres ingeven. De toerist stelt voor elke dag een interval op waarin zijn eetpauzes moeten plaatsvinden. Op basis van de verkregen informatie

wordt de databank bevestigd. Enkel de POI die bezocht kunnen worden zonder de reisbeperkingen te schenden, worden opgehaald (Vansteenwegen et al., 2011b).

Wanneer de reisbeperkingen bepaald zijn, bevestigd het systeem de gebruiker omtrent zijn interesses. Aan de hand van deze interesses wordt een gebruikersprofiel gecreëerd. Het profiel bestaat uit waarderingen van de verschillende types en categorieën. De toerist bepaalt in welke mate hij geïnteresseerd is in elke categorie en in elk type met behulp van de waarden 0 tot en met 3. Nul staat voor helemaal niet geïnteresseerd en 3 voor volkomen geïnteresseerd in een bepaalde categorie. Willekeurige kernwoorden kunnen toegevoegd worden aan het gebruiksprofiel zoals oorlog, moderne kunst, golf en piano. De webapplicatie voorziet de mogelijkheid het gebruikersprofiel op te slaan (Vansteenwegen et al., 2011b).

Het praktijkvoorbeeld wordt toegepast op de stad Hasselt. De reisbeperkingen alsook de interesses van de gebruiker zijn fictieve data. De toerist arriveert zondag 7 april 2013 en wil dinsdag 9 april 2013 vertrekken. De toerist kiest het Radisson Blu Hotel als verblijfplaats omdat de toeristische dienst dit aanbeveelt. Dit hotel is de begin- en eindlocatie voor elke dag. De toerist kiest bewust om zondag geen activiteiten uit te voeren. Die dag wenst hij uit te rusten van zijn rit naar Hasselt. In tabel 9 wordt de gewenste dagindeling weergegeven. De start van de eetpauzes moet in het interval plaatsvinden. Tabel 10 toont aan in welke mate de toerist zich interesseert voor de POI types en categorieën. De toerist kiest vijf kernwoorden uit een beschikbare lijst van kernwoorden met name dans, film, rust, thee en kledij. Deze kernwoorden geven het gebruikersprofiel een meerwaarde.

		Maandag 8 april 2013	Dinsdag 9 april 2013
<b>Starttijd - Stoptijd</b>		10u00 – 23u00	10u00 – 20u00
<b>Lunchpauze</b>	<i>Duur</i>	60 minuten	60 minuten
	<i>Interval</i>	12u00 – 14u00	12u00 – 13u30
<b>Diner</b>	<i>Duur</i>	90 minuten	90 minuten
	<i>Interval</i>	18u00 – 20u00	17u00 – 20u00

Tabel 9: Reisbeperkingen toerist

Type		Categorie	
Abdij	1	Architectuur	1
Begijnhof	0	Culinair	3
Kasteel	1	Cultuur	2
Kerk	0	Erfgoed	1
Kerkhof	1	Jongeren en kinderen	0
Museum en tentoonstelling	2	Kunst	3
Onderwijs	2	Natuur	2
Park	1	Sport	2
Sport en spel	2	Uitgaan	3
Theater en cinema	3		
Tuin	2		
Wetenschap en techniek	3		

Tabel 10: Interesse toerist in types en categorieën

### 3.4 Interessescore

De beperkingen en interesses van de gebruiker zijn bekend. Het systeem kan nu de persoonlijke interesse in elk POI schatten, weergegeven door een interessescore. De score wordt bepaald als de som van de typescore, de categoriescore en de kernwoordscore. De typescore wordt gelijkgesteld aan driemaal de interesse van de toerist in dat bepaald type. Wanneer de toerist niet geïnteresseerd is in een bepaald type, heeft dit type score nul. De bezienswaardigheid wordt dan uit de lijst van mogelijke POI gehaald. Wanneer de toerist sterk geïnteresseerd is in een bepaald type, heeft het POI een typescore van negen. De bezienswaardigheid krijgt een bonus van drie punten om de mogelijkheid tot opname in de route te vergroten. De tussencategoriescore van een POI wordt berekend als het product van de waarde van dat POI in een bepaalde categorie en de interesse van de gebruiker in diezelfde categorie. Wanneer de vermenigvuldiging negen bedraagt, wordt een bonus van drie punten toegekend. De categoriescore van een POI is de som van de drie hoogste tussencategoriescores. De maximale typescore is 12 en de maximale categoriescore 36 (Vansteenwegen et al., 2011b).

Vansteenwegen et al. (2011b) berekenen de kernwoordscore aan de hand van het *vector space* model van Baeza-Yates en Ribeiro-Neto (1999). Elke beschrijving van een POI wordt omgevormd tot een documentvector. De kernwoorden van de gebruiker vormen een query vector. Een query is een woord of combinatie van woorden die als zoekopdracht dienen voor een zoekmachine. De kernwoorden kunnen willekeurig of uit een beschikbare lijst gekozen worden. Beide vectoren worden vergeleken om tot een kernwoordscore te komen. Deze wordt genormaliseerd tussen 0 en 36. Deze procedure wordt gedetailleerd uitgelegd in Souffriau et al. (2008). Aangezien het bepalen van de interessescore niet het hoofddoel van deze eindverhandeling vormt, wordt de kernwoordscore op een minder complexe manier berekend. Elk kernwoord wordt vergeleken met de beschrijving van een POI. Wanneer geen relatie tussen beiden bestaat, bedraagt de score nul. In geval van een sterke relatie bedraagt de score drie. Dit getal wordt verdrievoudigd. Indien het resultaat negen is, wordt een bonus van drie punten toegekend. De drie hoogste kernwoordscores worden opgeteld om tot de totale kernwoordscore te komen. De maximale kernwoordscore is 36. Tabel 11 geeft de relatie tussen elk POI en elk kernwoord weer.

De som van de typescore, de categoriescore en de kernwoordscore vormt de persoonlijke interessescore van de toerist in een bepaald POI. De score kan maximaal 84 bedragen. Indien de toeristische dienst verplicht bezoek oplegt aan een bepaald POI, krijgt dit POI een score gelijk aan de som van alle interessescores opgeteld met één (Vansteenwegen et al., 2011b).

POI	Literair museum	Nationaal Jenevermuseum	Beiaardtoren	Modemuseum Hasselt	Japane tuin	Oud-Kerkhof	Virga Jesse basiliek	Park Natuur en Cultuur	Plopsa Indoor	Kapermolenpark	Kinepolis	Z33
Kernwoord												
Dans	0	0	0	0	0	0	0	0	1	2	2	2
Film	3	0	0	0	0	0	0	0	2	0	3	3
Rust	1	0	1	0	3	3	1	3	0	1	0	0
Thee	0	0	0	0	3	0	0	0	0	0	0	0
Kledij	0	0	0	3	0	0	0	0	0	0	0	0

Tabel 11: Relatie POI en kernwoorden

Voor elk van de gebruikte POI wordt de interessescore berekend. Bij wijze van voorbeeld worden twee berekeningen nader toegelicht. De andere berekeningen verlopen analoog. In tabel 12 worden de berekeningen van het Modemuseum Hasselt weergegeven. Het Modemuseum behoort tot het type 'Museum en tentoonstelling' (tabel 7). Uit tabel 10 wordt afgeleid dat de toerist de waarde twee geeft aan dit type. De totale typescore bedraagt zes. Er worden geen bonuspunten toegekend. Uit tabel 8 wordt afgeleid hoe hoog het Modemuseum op elke categorie scoort. Tabel 10 geeft aan in welke mate de toerist geïnteresseerd is in een bepaalde categorie. De twee verkregen waarden worden vermenigvuldigd om een tussenscore per categorie te verkrijgen. Wanneer het resultaat negen is, wordt een bonus van drie punten toegekend. Dit is in casu niet het geval. De drie hoogste tussencategoriescores worden opgeteld om de totale categoriescore te verkrijgen. Het kernwoord kledij staat in sterke relatie tot het Modemuseum (tabel 11). Dit kernwoord krijgt score drie. Na verdrievoudiging wordt een bonus van drie punten aan dit kernwoord toegekend. De interessescore is de som van de typescore, de categoriescore en de kernwoordscore. Voor het Modemuseum Hasselt bedraagt de interessescore 31.

Tabel 13 toont de scoreberekening voor de Virga Jessebasiliek. De basiliek behoort tot het type 'Kerk' (tabel 7). Aangezien de toerist aan het type 'Kerk' de waarde nul hecht (tabel 10), wordt de Virga Jessebasiliek uit de lijst van mogelijke POI verwijderd. De categorie- en kernwoordscore zijn alsnog berekend, hoewel dit niet nodig is voor het verdere verloop. De interessescore bedraagt uiteindelijk nul dankzij de typescore. Geen enkel ander POI heeft een typescore van nul.

<b>Modemuseum Hasselt</b>		<i>POI</i>	<i>Toerist</i>	<i>Tussenscore</i>	<i>Inclusief bonus</i>	<i>Totaal-score</i>
<i>Type</i>	Museum tentoonstelling en		2	6	6	6
<i>Categorie</i>	Architectuur	1	1	1	1	
	Culinair	0	3	0	0	
	Cultuur	3	2	6	6	
	Erfgoed	1	1	1	1	
	Jongeren en kinderen	0	0	0	0	
	Kunst	2	3	6	6	
	Natuur	0	2	0	0	
	Sport	0	2	0	0	
	Uitgaan	0	3	0	0	13
<i>Kernwoord</i>	Dans	0		0	0	
	Film	0		0	0	
	Rust	0		0	0	
	Thee	0		0	0	
	Kledij	3		9	12	12
<b>Totaalscore</b>					<b>31</b>	

Tabel 12: Berekening totaalscore Modemuseum Hasselt

<b>Virga Jessebasiliek</b>		<i>POI</i>	<i>Toerist</i>	<i>Tussenscore</i>	<i>Inclusief bonus</i>	<i>Totaal-score</i>
<i>Type</i>	Kerk		0	0	0	<b>0</b>
<i>Categorie</i>	Architectuur	3	1	3	3	
	Culinair	0	3	0	0	
	Cultuur	1	2	2	2	
	Erfgoed	3	1	3	3	
	Jongeren en kinderen	0	0	0	0	
	Kunst	1	3	3	3	
	Natuur	0	2	0	0	
	Sport	0	2	0	0	
	Uitgaan	0	3	0	0	9
<i>Kernwoord</i>	Dans	0		0	0	
	Film	0		0	0	
	Rust	1		3	3	
	Thee	0		0	0	
	Kledij	0		0	0	3
<b>Totaalscore</b>					<b>0</b>	

Tabel 13: Berekening totaalscore Virga Jessebasiliek

### 3.5 Tourist Trip Design algoritme

#### 3.5.1 Mathematische formulering

Het praktijkvoorbeeld start met 12 locaties aangezien de Virga Jessebasiliek reeds verwijderd werd. Tabel 14 geeft een oplistijng van de 12 knooppunten, de score  $S_i$ , de gemiddelde bezoektijd  $T_i$ , de kostprijs  $e_i$  en het tijdvenster  $[O_i, C_i]$  van maandag en dinsdag. Opmerkelijk is de hoge score van de Japanse Tuin. De kernwoorden 'thee' en 'rust' worden sterk geassocieerd met de Japanse Tuin. Beide knooppunten krijgen score 12. De tijd nodig om van knooppunt  $i$  naar knooppunt  $j$  te reizen, wordt weergegeven in tabel 15. Dit is de wandeltijd in minuten. De toerist heeft een tijdsbudget van 780 minuten op maandag en 600 minuten op dinsdag. Er moet rekening gehouden worden met de gevraagde lunch- en dinerpauze. Het doel is voor elke dag één route bepalen waarbij de totale score, verzameld op twee dagen, gemaximaliseerd wordt.

Naam	$i$	$S_i$	$T_i$	$e_i$	$[O_i, C_i]$
Hotel Radisson Blu	1/13	-	-	-	-
Literair museum	2	34	90	2,5	[14u00,17u00]
Nationaal Jenevermuseum	3	24	90	4,5	[10u00,17u00]
Beiaardtoren	4	24	90	1,5	[10u00,17u00]
Modemuseum Hasselt	5	31	90	5	[10u00,17u00]
Japanse tuin	6	45	90	5	[10u00,17u00]
Oud-Kerkhof	7	23	120	0	[10u00,16u00]
Park Natuur en Cultuur	8	30	120	4,5	[10u00,18u00]
Plopsa Indoor	9	17	300	18	[10u00,18u00]
Kapermolenpark	10	28	120	0	[08u00,22u00]
Kinepolis	11	42	150	7,3	[14u15,00u45]
Z33	12	45	90	0	[11u00,18u00]

Tabel 14: Gegevens knooppunten praktijkvoorbeeld

$t_{ij}$ (minuten)	1, 13	2	3	4	5	6	7	8	9	10	11	12
1, 13	-	13	9	5	10	18	25	49	23	21	28	6
2	13	-	14	10	11	28	28	41	33	28	39	12
3	9	14	-	6	3	14	18	53	21	16	27	3
4	5	10	6	-	6	17	22	48	23	19	29	2
5	10	11	3	6	-	16	17	50	24	17	29	4
6	18	28	14	17	16	-	26	66	7	9	15	15
7	25	28	18	22	17	26	-	67	28	25	38	20
8	49	41	53	48	50	66	67	-	72	67	78	51
9	23	33	21	23	24	7	28	72	-	14	10	22
10	21	28	16	19	17	9	25	67	14	-	18	16
11	28	39	27	29	29	15	38	78	10	18	-	28
12	6	12	3	2	4	15	20	51	22	16	28	-

Tabel 15: Reistijd tussen twee knooppunten



Het probleem wordt geformuleerd als een integer programmeringsprobleem gebaseerd op de formuleringen van Vansteenwegen et al. (2009), Garcia et al. (2010) en Souffriau et al. (2012). Hierbij wordt gebruik gemaakt van volgende notatie:  $x_{ijd} = 1$  als in route  $d$  een bezoek aan knooppunt  $j$  volgt op een bezoek aan knooppunt  $i$  en 0 elders;  $y_{id} = 1$  als knooppunt  $i$  bezocht wordt in route  $d$  en 0 elders;  $s_{id} =$  de start van een bezoek aan knooppunt  $i$  in route  $d$  en  $M$  is een constante met hoge waarde. De betekenis van de overige symbolen is terug te vinden in tabellen 2, 3 en 5 uit respectievelijk deelsectie 2.2.2, 2.3.2 en 2.4.2.

De doelfunctie (1) tracht de totale score te maximaliseren. Beperking (2) zorgt ervoor dat elke route start in knooppunt 1 en eindigt in knooppunt 13. Beide knooppunten één en dezelfde locatie, namelijk het Radisson Blu Hotel. Aan de hand van (3) wordt rekening gehouden met het kostenbudget. Door beperkingen (4) en (5) wordt de connectiviteit en tijdlijn van elke route verzekerd. Beperking (6) limiteert het aantal bezoeken aan een knooppunt. Het tijdsbudget van de toerist komt tot uiting in beperkingen (7) en (8). Beperkingen (9) worden opgesteld om ervoor te zorgen dat de start van een bezoek aan een locatie in het tijdvenster valt.

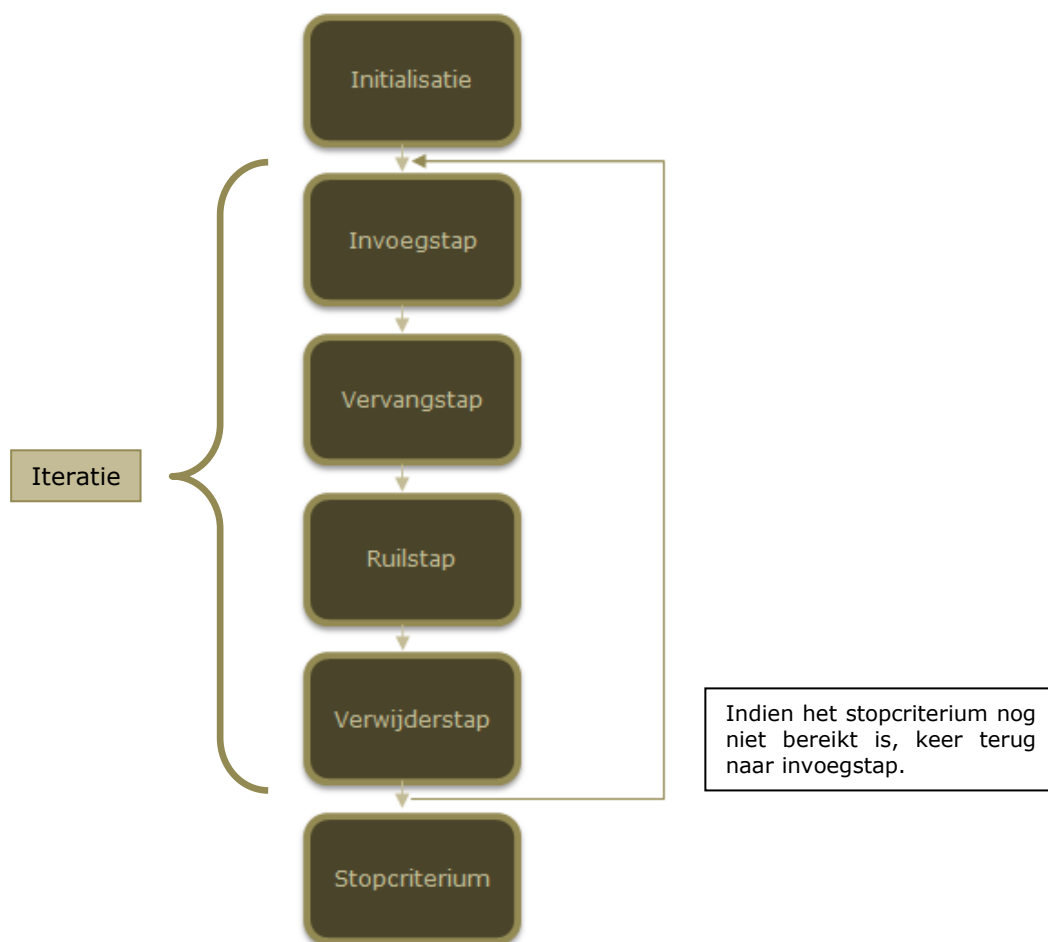
$Max \sum_{d=1}^2 \sum_{i=2}^{12} S_i y_{id}$	(1)
$\sum_{j=2}^{12} x_{1jd} = \sum_{i=2}^{12} x_{i13d} = 1 \ (\forall d = 1,2)$	(2)
$\sum_{d=1}^2 \sum_{i=2}^{12} e_i y_{id} \leq 140$	(3)
$\sum_{i=1}^{12} x_{ikd} = \sum_{j=2}^{13} x_{kj d} = y_{kd} \ (k = 2, \dots, 12; d = 1,2)$	(4)
$s_{id} + T_i + t_{ij} - s_{jd} \leq M(1 - x_{ijd}) \ (i, j = 1, \dots, 13; d = 1,2)$	(5)
$\sum_{d=1}^2 y_{kd} \leq 1 \ (k = 2, \dots, 12)$	(6)
$\sum_{i=1}^{12} (T_i y_{i1} + \sum_{j=2}^{13} t_{ij} x_{ij1}) \leq 780$	(7)
$\sum_{i=1}^{12} (T_i y_{i2} + \sum_{j=2}^{13} t_{ij} x_{ij2}) \leq 600$	(8)
$0_i \leq s_{id} \leq C_i \ (i = 1, \dots, 13; d = 1,2)$	(9)
$x_{ijd}, y_{id} \in \{0,1\} \ (i, j = 1, \dots, 13; d = 1,2)$	(10)

### 3.5.2 Algoritme

Uit de literatuurstudie blijkt dat verscheidene auteurs een *tabu search* algoritme toepassen op het TTDP. Daarom wordt in de praktijkstudie geopteerd om een gelijkaardig algoritme op te stellen voor het praktijkprobleem. De structuur van een *tabu search* algoritme is als volgt. Het algoritme start met een willekeurige toegelaten oplossing. Om toegelaten bewegingen te definiëren in de buurt van de huidige oplossing, worden lokale zoekprocedures gebruikt. Bewegingen die zich op de taboelijn bevinden, mogen niet genomen worden. Er moet bepaald worden welke van de mogelijke bewegingen het beste resultaat geeft. Na het uitvoeren van een beweging, wordt een oplossing verkregen. Deze oplossing wordt de huidige oplossing ongeacht of het een verbetering of verslechtering is ten opzichte van de vorige oplossing. De taboelijn moet worden aangepast na elke beweging. Wanneer deze vol is, wordt het oudste lid van de lijst verwijderd. Het stopcriterium kan een vast aantal

iteraties, een bepaalde hoeveelheid berekeningstijd of een vast aantal opeenvolgende iteraties zonder verbetering zijn. Wanneer in een iteratie geen toegelaten beweging gevonden wordt, wordt het algoritme ook stopgezet. De best gevonden oplossing wordt aanvaard als de uiteindelijke oplossing (Hillier en Lieberman, 2010).

Figuur 5 toont de werking van het algoritme. Het algoritme start met de initialisatie waarna een iteratie volgt. Eén iteratie bestaat uit vier stappen, namelijk de invoegstap, de vervangstap, de ruilstap en de verwijderstap. Elke stap wordt uitgevoerd totdat geen zet meer mogelijk is binnen die bepaalde stap. Wanneer de vier stappen uitgevoerd zijn, start een nieuwe iteratie. Als het stopcriterium bereikt wordt, stopt het algoritme.



Figuur 5: Structuur algoritme

### 3.5.2.1 Initialisatie

Het creëren van de initiële oplossing is gebaseerd op de studie van Souffriau et al. (2008). Een bepaald aantal punten die het verst verwijderd zijn in tijd van de startlocatie, worden geselecteerd. In het praktijkvoorbeeld zijn dit twee punten aangezien twee routes moeten worden bepaald. Elk van deze twee punten wordt willekeurig toegewezen aan één van de twee routes. Alle overblijvende knooppunten worden aan één van de twee routes toegevoegd

op basis van *cheapest insertion*. Het knooppunt dat het minst ver gelegen is van het laatst toegevoegde knooppunt wordt toegevoegd aan de route. De initiële oplossing moet een toegelaten oplossing zijn. Daarom wordt steeds rekening gehouden met de reisbeperkingen van de toerist alsook met de beperkingen van de knooppunten. In tabel 15 werd de reistijd tussen de knooppunten reeds weergegeven.

Door het bestaan van openingsdagen kan niet elk knooppunt aan elke route worden toegevoegd. Knooppunten 4, 6, 7, 8, 9, 10 en 11 kunnen aan route 1 worden toegevoegd. Aan route 2 kan enkel knooppunt 2 niet worden toegevoegd. Knooppunt 2 wordt vanaf nu buiten beschouwing gelaten. Zowel route 1 als route 2 starten in knooppunt 1 om 10u00. Knooppunten 8 en 11 liggen het verst verwijderd van deze locatie. Deze twee knooppunten worden willekeurig aan één van de twee routes toegevoegd. Tabel 16 geeft een overzicht van de initiële routes. In kolom 2 wordt de start van de route aangeduid. Kolom 3 geeft aan wanneer de toerist aankomt op het eerste knooppunt. Kolom 4 toont het tijdstip waarop de toerist klaar is voor zijn vertrek naar het volgend knooppunt. De totstandkoming tabel 16 wordt hieronder uitgelegd.

Knooppunt 8 wordt toegevoegd aan route 1. Aangezien de toerist 49 minuten wandelt van knooppunt 1 naar knooppunt 8, arriveert hij om 10u49 op zijn bestemming. Daar geniet hij 120 minuten van het bezoek aan het Park Natuur en Cultuur. Het volgend knooppunt wordt bepaald met behulp van *cheapest insertion*. Vooraleer een knooppunt wordt toegevoegd, wordt eerst een lunchpauze voorzien. De lunchpauze start om 12u49, ervan uitgaande dat in de buurt van knooppunt 8 eetmogelijkheden zijn. Knooppunt 4 is het minst ver verwijderd van knooppunt 8. Dit knooppunt kan met succes worden toegevoegd aan route 1. De volgende knooppunten die in aanmerking komen zijn knooppunten 12, 3, 5 en 6. Deze knooppunten kunnen niet aan route 1 worden toegevoegd omwille van hun tijdvensterbeperking. Het invoegen van knooppunt 10 is toegelaten. Na knooppunt 10 kan geen enkel knooppunt meer worden toegevoegd zonder een beperking te schenden. De toerist keert na zijn diner terug naar de eindbestemming die hij om 20u17 bereikt.

Route 2 start met een wandeling van 28 minuten naar knooppunt 11. Voor Kinopolis wordt een bijkomende veronderstelling gemaakt. De toerist kan zijn bezoek aan Kinopolis starten om 14u15, 16u45, 19u45 of 22u15. Het eerstvolgende bezoek start om 14u15. Het vertrek naar het volgend knooppunt is 150 minuten later. De lunchpauze wordt vóór knooppunt 11 toegevoegd. Knooppunten 9 en 6 die als volgende in aanmerking komen, kunnen niet worden toegevoegd omwille van hun tijdvensterbeperking. Knooppunt 10 kan niet worden toegevoegd omdat het reeds in route 1 opgenomen is. De overige mogelijke knooppunten kunnen niet worden toegevoegd vanwege hun tijdvenster of het feit dat ze reeds opgenomen zijn in route 1. De toerist keert na zijn diner terug naar de eindlocatie. De eerste en laatste rij van tabel 16 tonen de kostenfunctie. De reiziger heeft een totaalbudget van 140 euro. De lunch bedraagt 10 euro en het diner 30 euro.

De doelfunctiewaarde wordt aangeduid met  $Z$ . De index van  $Z$  bestaat uit twee cijfers. Het eerste cijfer staat voor de stap en het tweede cijfer voor de route. De verzamelde score in route 1 wordt aangeduid met  $Z_{01}$  en bedraagt 82. In route 2 bedraagt de verzamelde score  $Z_{02}$  42. De doelfunctiewaarde  $Z_0$  ( $Z_{01} + Z_{02}$ ) bedraagt 124.

<b>Route 1</b>	<i>Kostprijs</i>			4,5		14,5	
	<i>Uur van aankomst/vertrek</i>	10u00	10u49	12u49	12u49	13u49	14u37
	<i>Knooppunt</i>	1	-	8	-	Lunch	-
<b>Route 2</b>	<i>Knooppunt</i>	1	-	Lunch	-	11	-
	<i>Uur van aankomst/vertrek</i>	10u00	10u00	13u00	13u28	16u45	16u45
	<i>Kostprijs</i>			10		17,3	

Tabel 16: Initialisatie

<b>Route 1</b>	<i>Kostprijs</i>	16		16		46		
	<i>Uur van aankomst/vertrek</i>	16u07	16u26	18u26	18u26	19u56	20u17	-
	<i>Knooppunt</i>	4	-	10	-	Diner	-	13
<b>Route 2</b>	<i>Knooppunt</i>	Diner	-	13				
	<i>Uur van aankomst/vertrek</i>	18u30	18u58	-				
	<i>Kostprijs</i>	47,3		-				

Tabel 16 (vervolg): Initialisatie

### 3.5.2.2 Invoegstap

De eerste stap van een iteratie is de invoegstap (gebaseerd op Vansteenwegen et al., 2009). In deze stap worden één voor één nieuwe knooppunten toegevoegd aan een route startend bij de eerste route. Het doel is de score te verhogen. De invoegstap eindigt wanneer geen enkel knooppunt nog kan worden toegevoegd. Met behulp van een optellende functie wordt nagegaan of het kostenbudget niet overschreden wordt. Vooraleer een bezoek kan worden toegevoegd, moet onderzocht worden of alle bezoeken die plaatsvinden na het toegevoegde bezoek nog steeds voldoen aan hun tijdvensterbeperking. Eveneens wordt nagegaan of het tijdsbudget niet overschreden wordt. Voor elke opgenomen locatie worden verschillende parameters bijgehouden om de verificering van de beperkingen sneller te laten verlopen (gebaseerd op Vansteenwegen et al., 2009).  $Wachttijd_i$  is de wachttijd bij aankomst aan knooppunt  $i$  vooraleer het bezoek effectief kan starten. Deze wachttijd is nul als de toerist aankomt tijdens het tijdvenster.

$$(1) \text{ Wachttijd}_i = \max[0, O_i - a_i] \quad \text{met } O_i \text{ het openingsuur en } a_i \text{ het aankomstuur.}$$

$Maxverplaatsing_i$  duidt aan hoe ver het einde van het bezoek naar achter geplaatst kan worden zonder een van de reeds opgenomen bezoeken niet toegelaten te maken.

$$(2) \text{ Maxverplaatsing}_i = \min[C_i - l_i + T_i / 2, \text{Wachttijd}_{i+1} + \text{Maxverplaatsing}_{i+1}]$$

$$(3) \text{ Maxverplaatsing}_i = \min[C_i - l_i, \text{Wachttijd}_{i+1} + \text{Maxverplaatsing}_{i+1}]$$

$$(4) \text{ Maxverplaatsing}_i = \min[C_i - s_i, \text{Wachttijd}_{i+1} + \text{Maxverplaatsing}_{i+1}] \quad \text{waarbij } C_i \text{ het sluitingsuur is, } l_i \text{ het einde van een bezoek, } T_i \text{ de gemiddelde bezoektijd en } s_i \text{ de start van een bezoek.}$$

Voor types abdij, begijnhof, kasteel, kerk, kerkhof, onderwijs, park, tuin én wetenschap en techniek wordt de eerste formule toegepast. Een bezoek aan deze types moet niet noodzakelijk de gemiddelde bezoektijd bedragen. De gemiddelde bezoektijd van een park bijvoorbeeld kan twee uur bedragen. Wanneer de toerist één uur voor sluitingstijd aankomt, kan hij het voldoende vinden het park gedurende één uur te bezoeken. De voldoening van de toerist, aangeduid door  $S_i$ , wordt dan uiteraard ook gehalveerd. De toerist kan zijn bezoek aan het knooppunt ten laatste starten wanneer het tijdstip de helft van de gemiddelde bezoektijd vóór sluitingsuur is. Stel dat de toerist het Park Natuur en Cultuur wil bezoeken. Het Park Natuur en Cultuur sluit om 18u00 en de gemiddelde bezoektijd bedraagt 120 minuten. De toerist kan ten laatste 60 minuten vóór sluitingsuur het Park Natuur en Cultuur betreden dus om 17u00. De tweede formule van *Maxverplaatsing<sub>i</sub>* wordt gebruikt voor types museum en tentoonstelling, sport en spel én theater en cinema. Vaak is het bij dergelijke types niet mogelijk maar een gedeelte van de gemiddelde bezoektijd uit te voeren. In een museum bijvoorbeeld wordt vaak een opgestelde rondleiding voorgesteld die een bepaalde duur heeft. Vandaar dat de maximale verplaatsing van het einde van het bezoek de sluitingstijd verminderd met het voorlopige einde van het bezoek bedraagt. De derde formule is van toepassing op de lunch- en dinerpauze aangezien de start van het bezoek binnen het interval dient te vallen. Op maandag 8 april bijvoorbeeld kan de lunchpauze ten laatste starten om 14u00.

De totale consumptietijd van een bezoek  $j$  dat wordt ingevoegd tussen knooppunten  $i$  en  $k$ , wordt gedefinieerd als

$$(5) \text{ Consumptietijd}_j = t_{ij} + \text{Wachttijd}_j + T_j + t_{jk} - t_{ik}$$

Let wel, de consumptietijd van bepaalde bezoeken zal afhangen van de starttijd van het bezoek. Het is namelijk mogelijk om bijvoorbeeld het Park Natuur en Cultuur te betreden om 16u45. De gemiddelde bezoektijd van 120 minuten gaat dan niet op, maar  $T_j$  bedraagt dan 75 minuten. Om na te gaan of de invoeging van knooppunt  $j$  tussen knooppunten  $i$  en  $k$  toegelaten is, dient de ongelijkheid  $\text{Consumptietijd}_j \leq \text{Wachttijd}_k + \text{Maxverplaatsing}_k$  op te gaan. Ook moet rekening gehouden worden met de tijdvensterbeperking van knooppunt  $j$ . Voor elk knooppunt wordt de laagste consumptietijd bepaald aangezien dit de best mogelijke invoeging vormt. Om een knooppunt te selecteren, wordt gebruik gemaakt van volgende ratio:

$$(6) \text{ Ratio}_j = \frac{S_j^2}{\frac{\text{Consumptietijd}_j}{\text{BeschikbareTijd}} + \frac{e_j}{\text{BeschikbaarBudget}}} \quad (\text{gebaseerd op Garcia et al., 2010})$$

*BeschikbareTijd* is de resterende tijd die nog ter beschikking staat om knooppunten te bezoeken exclusief de wachttijden. *BeschikbaarBudget* is het budget dat nog overblijft om aan de knooppunten te besteden. De hoeveelheid die nog ter beschikking staat van elke beperking is belangrijk en intuïtief relevanter dan de grenswaarde van deze beperking. De

consumptie van de meest bindende beperking wordt hierdoor belangrijker (Garcia et al., 2010).

Wanneer een knooppunt toegevoegd is aan een route, worden de gegevens voor alle knooppunten vernieuwd. De knooppunten die na het ingevoegde knooppunt bezocht worden, vereisen een nieuwe waarde voor de wachttijd ( $Wachttijd_i$ ), de aankomsttijd ( $a_i$ ), de start van de dienst ( $s_i$ ) en de maximale verplaatsing van het einde van het bezoek ( $Maxverplaatsing_i$ ). Dit gebeurt aan de hand van onderstaande formules (Vansteenwegen et al., 2009). Stel dat knooppunt  $j$  wordt ingevoegd tussen knooppunt  $i$  en  $k$ , dan:

$$(7) \text{ Consumptietijd}_j = t_{ij} + Wachttijd_j + T_j + t_{jk} - t_{ik} \quad \text{met } T_j \text{ afhankelijk van de start van het bezoek.}$$

$$(8) Wachttijd_{k^*} = \max[0, Wachttijd_k - Consumptietijd_j]$$

$$(9) a_{k^*} = a_k + Consumptietijd_j$$

$$(10) Consumptietijd_k = \max[0, Consumptietijd_j - Wachttijd_k]$$

$$(11) s_{k^*} = s_k + Consumptietijd_k$$

$$(12) Maxverplaatsing_{k^*} = Maxverplaatsing_k - Consumptietijd_k$$

De consumptietijd van  $k$  en dezelfde formules worden dan gebruikt om de knooppunten achter  $k$  te vernieuwen, totdat de consumptietijd nul bedraagt. Bezoeken die voor  $j$  plaatsvinden, vragen een update van de maximale verplaatsing zoals in formules (2), (3) en (4).

Een knooppunt dat net ingevoegd is, mag niet verwijderd worden in de volgende stap. Om die reden wordt het ingevoegde knooppunt op de taboelijst geplaatst. De taboelijst heeft een lengte van twee knooppunten omwille van de beperkte omvang van het praktijkprobleem. Voor de invoegstap is de taboelijst weinig zinvol aangezien hier geen enkel knooppunt voor verwijdering beschouwd wordt. In de andere stappen zal duidelijk blijken dat de taboelijst invloed heeft op de oplossing. De invoegstap wordt nu toegepast op het praktijkvoorbeeld.

Tabel 17 toont de initiële route 1 gebaseerd op tabel 16, waar  $E$  de kostprijs tot dan toe is.  $l_i$  duidt het einde van een bezoek aan en wordt dus gedefinieerd als de starttijd van het bezoek vermeerderd met de bezoektijd. De maximale verplaatsing van knooppunt 13 wordt berekend als het tijdstip waarop de toerist ten laatste wil aankomen in het hotel verminderd met het tijdstip waarop hij in deze oplossing aankomt in het hotel.

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$	$Maxverplaatsing_i$
8	0	10u49	10u49	12u49	4,5	71
Lunch	0	12u49	12u49	13u49	14,5	71
4	0	14u37	14u37	16u07	16	53
10	0	16u26	16u26	18u26	16	94
Diner	0	18u26	18u26	19u56	46	94
13	-	20u17	-	-	-	163

Tabel 17: Initiële route 1

De invoegstap tracht knooppunten toe te voegen aan de route zodat de score verhoogd wordt. Knooppunten 4, 6, 7, 8, 9, 10 en 11 kunnen worden toegevoegd aan route 1. Hiervan zijn 4, 8 en 10 reeds opgenomen in route 1 en 11 in route 2. Knooppunten 6, 7 en 9 worden beschouwd voor invoeging. De berekeningen zijn terug te vinden in tabel 18. Hieronder volgt enige uitleg bij de gemaakte berekeningen.

Knooppunt 6 kan niet worden toegevoegd achter knooppunt 4 omwille van zijn tijdvenster. De toerist zou om 16u24 arriveren op knooppunt 6. De start van het bezoek dient echter ten laatste plaats te vinden om 16u15. Wanneer knooppunt 7 tussen de lunchpauze en knooppunt 4 wordt toegevoegd, kan dit knooppunt niet volledig bezocht worden. Aangezien de toerist op knooppunt 7 aankomt om 14u56, bedraagt de bezoektijd slechts 64 minuten in plaats van 120 minuten. Uit de berekeningen omtrent knooppunt 9 blijkt reeds dat de kans dat knooppunt 9 opgenomen wordt in route 1 zeer klein is. Aan de voorwaarde  $Consumptietijd_j \leq Wachttijd_k + Maxverplaatsing_k$  is in geen enkel geval voldaan. Knooppunten 6, 7 en 9 kunnen aldus niet worden toegevoegd aan de eerste route waardoor de invoegstap voor de eerste route reeds voltooid is. Route 1 behoudt voorlopig de structuur zoals in tabel 17 met een doelfunctiewaarde van 82 (=  $Z_{11}$ ).

$i$	$k$	$Consumptietijd_j$	$Wachttijd_k + Maxverplaatsing_k$	$Ratio_j$
<b>Knooppunt 6</b>				
1	8	$18 + 66 - 49 + 0 + 90 = 125$	71	-
8	Lunch	$66 + 0 - 0 + 0 + 90 = 156$	71	-
Lunch	4	$66 + 17 - 48 + 0 + 90 = 125$	53	-
4	10	tijdvensterbeperking knooppunt 6		
10	Diner	tijdvensterbeperking knooppunt 6		
Diner	13	tijdvensterbeperking knooppunt 6		
<b>Knooppunt 7</b>				
1	8	$25 + 67 - 49 + 0 + 120 = 163$	71	-
8	Lunch	$67 + 0 - 0 + 0 + 120 = 187$	71	-
Lunch	4	$67 + 22 - 48 + 0 + 64 = 105$	53	-
4	10	tijdvensterbeperking knooppunt 7		
10	Diner	tijdvensterbeperking knooppunt 7		
Diner	13	tijdvensterbeperking knooppunt 7		
<b>Knooppunt 9</b>				
1	8	$23 + 72 - 49 + 0 + 300 = 346$	71	-
8	Lunch	tijdvensterbeperking knooppunt 9		
Lunch	4	tijdvensterbeperking knooppunt 9		
4	10	tijdvensterbeperking knooppunt 9		
10	Diner	tijdvensterbeperking knooppunt 9		
Diner	13	tijdvensterbeperking knooppunt 9		

Tabel 18: Iteratie 1 – Route 1 – Invoegstap 1: berekeningen

Tabel 19 toont de initiële oplossing van route 2. De maximale verplaatsing van knooppunt 11 wordt als volgt berekend:  $Maxverplaatsing_{11} = \min[00u45 - 14u15, 62 + 15] = 77$ . Het is echter zo dat Kinopolis specifieke starturen heeft. Een bezoek kan starten om 14u15, 16u45,

19u45 of 22u15. Daarom bedraagt de maximale verplaatsing nul in plaats van 77. Wanneer het bezoek zou starten om 16u45, worden de reisbependingen van de toerist wat betreft het diner en de aankomst in het hotel geschonden.

$i$	Wachttijd $_i$	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing $_i$
Lunch	120	10u00	12u00	13u00	10	47
11	47	13u28	14u15	16u45	17,3	0
Diner	15	16u45	17u00	18u30	47,3	62
13	-	18u58	-	-	-	62

Tabel 19: Initiële route 2

Knooppunten 3, 5, 6, 7, 9 en 12 kunnen worden toegevoegd aan route 2. De berekeningen zijn terug te vinden in tabel 20. Ook in route 2 kan knooppunt 9 op geen enkele plaats worden toegevoegd, mede door de lange gemiddelde bezoekduur van dit knooppunt. Wanneer de consumptietijd van  $j$  kleiner is dan of gelijk is aan de som van de wachttijd en de maximale verplaatsing van  $k$ , wordt de kleinste consumptietijd voor knooppunt  $j$  bepaald. In bovenstaand voorbeeld is dit voor knooppunten 3, 5, 6, 7 en 12 de enige toegelaten consumptietijd. Daarna wordt voor elk knooppunt  $Ratio_j$  berekend om te bepalen welk knooppunt het beste resultaat teweegbrengt. Ter illustratie wordt de ratio van knooppunt 3 berekend:

$$Ratio_3 = \frac{24^2}{\frac{99}{600-60-28-150-90-28} + \frac{4,5}{70-47,3}} = \frac{576}{0,604} = 953,81$$

Op dezelfde manier worden de ratio's voor knooppunten 5, 6, 7 en 12 berekend.

$i$	$k$	Consumptietijd $_j$	Wachttijd $_k$ + Maxverplaatsing $_k$	Ratio $_j$
<b>Knooppunt 3</b>				
1	Lunch	$9 + 0 - 0 + 0 + 90 = 99$	167	953,81
Lunch	11	$9 + 27 - 28 + 0 + 90 = 98$	47	-
11	Diner	tijdvensterbepending knooppunt 3		
Diner	13	tijdvensterbepending knooppunt 3		
<b>Knooppunt 5</b>				
1	Lunch	$10 + 0 - 0 + 0 + 90 = 100$	167	1525,154
Lunch	11	$10 + 29 - 28 + 0 + 90 = 101$	47	-
11	Diner	tijdvensterbepending knooppunt 5		
Diner	13	tijdvensterbepending knooppunt 5		
<b>Knooppunt 6</b>				
1	Lunch	$18 + 0 - 0 + 0 + 90 = 108$	167	3054,818
Lunch	11	$18 + 15 - 28 + 0 + 90 = 95$	47	-
11	Diner	tijdvensterbepending knooppunt 6		
Diner	13	tijdvensterbepending knooppunt 6		
<b>Knooppunt 7</b>				
1	Lunch	$25 + 0 - 0 + 0 + 120 = 145$	167	890,179
Lunch	11	$25 + 38 - 28 + 0 + 120 = 155$	47	-
11	Diner	tijdvensterbepending knooppunt 7		
Diner	13	tijdvensterbepending knooppunt 7		



Knooppunt 9				
1	Lunch	$23 + 0 - 0 + 0 + 300 = 323$	167	-
Lunch	11	tijdvensterbeperking knooppunt 9		
11	Diner	tijdvensterbeperking knooppunt 9		
Diner	13	tijdvensterbeperking knooppunt 9		
Knooppunt 12				
<b>1</b>	<b>Lunch</b>	<b><math>6 + 0 - 0 + 54 + 90 = 150</math></b>	<b>167</b>	<b>3294</b>
Lunch	11	$6 + 28 - 28 + 0 + 90 = 96$	47	-
11	Diner	tijdvensterbeperking knooppunt 12		
Diner	13	tijdvensterbeperking knooppunt 12		

Tabel 20: Iteratie 1 – Route 2 – Invoegstap 1: berekeningen

De hoogste ratio is die van knooppunt 12 waardoor knooppunt 12 gekozen wordt voor invoeging. De waarden van de andere opgenomen knooppunten worden aangepast op basis van formules (7) tot en met (12).

$$\begin{aligned}
 \text{Consumptietijd}_{12} &= 150 \\
 \text{Wachttijd}_{Lunch^*} &= \max[0, 120 - 150] = 0 \\
 a_{Lunch^*} &= 10u00 + 150 = 12u30 \\
 \text{Consumptietijd}_{Lunch} &= \max[0, 150 - 120] = 30 \\
 s_{Lunch^*} &= 12u00 + 30 = 12u30 \\
 \text{Maxverplaatsing}_{Lunch^*} &= 47 - 30 = 17 \\
 \text{Wachttijd}_{11^*} &= \max[0, 47 - 30] = 17 \\
 a_{11^*} &= 13u28 + 30 = 13u58 \\
 \text{Consumptietijd}_{11} &= \max[0, 30 - 47] = 0 \\
 s_{11^*} &= 14u15 \\
 \text{Maxverplaatsing}_{11^*} &= 0 - 0 = 0
 \end{aligned}$$

Bij de berekening van  $s_{11^*}$  dient rekening gehouden te worden met de specifieke starturen van knooppunt 11. Aangezien  $\text{consumptietijd}_{11}$  0 bedraagt, blijven de gegevens van de andere knooppunten hetzelfde als voorheen. Vóór knooppunt 12 bevinden zich geen bezoeken waardoor geen andere aanpassingen gemaakt moeten worden. Tabel 21 toont route 2 na invoeging van knooppunt 12. De totale score van route 2 bedraagt nu 87 (=  $Z_{12}$ ). Dit leidt tot een totale doelfunctiewaarde  $Z_1$  van 169. Knooppunt 12 bevindt zich nu op de taboelijst. Dit knooppunt mag in de volgende stap niet verwijderd worden.

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$	$Maxverplaatsing_i$
12	54	10u06	11u00	13u30	0	17
Lunch	0	12u30	12u30	13u30	10	17
11	17	13u58	14u15	16u45	17,3	0
Diner	15	16u45	17u00	18u30	47,3	62
13	-	18u58	-	-	-	62

Tabel 21: Iteratie 1 – Route 2

Voor route 1 werd reeds duidelijk dat geen knooppunt kan worden toegevoegd aan deze route. Voor route 2 wordt de invoegstap toegepast totdat geen enkel knooppunt nog kan

worden toegevoegd. Knooppunten 3, 5, 6, 7 en 9 worden beschouwd voor invoeging waarvan de berekeningen terug te vinden zijn in tabel 22. Zoals in de tabel tot uiting wordt gebracht, kan na knooppunt 11 geen enkel knooppunt worden toegevoegd omwille van hun tijdvensterbeperking. Aan de ongelijkheid  $consumptietijd_j \leq wachttijd_k + maxverplaatsing_k$  is in geen enkel geval voldaan. Aangezien geen knooppunten meer kunnen worden toegevoegd in de invoegstap, wordt overgegaan naar de vervangstap.

$i$	$k$	$Consumptietijd_j$	$Wachttijd_k + Maxverplaatsing_k$	$Ratio_j$
<b>Knooppunt 3</b>				
1	12	$9 + 3 - 6 + 0 + 90 = 96$	71	-
12	Lunch	$3 + 0 - 0 + 0 + 90 = 93$	17	-
Lunch	11	$3 + 27 - 28 + 0 + 90 = 92$	17	-
<b>Knooppunt 5</b>				
1	12	$10 + 4 - 6 + 0 + 90 = 98$	71	-
12	Lunch	$4 + 0 - 0 + 0 + 90 = 94$	17	-
Lunch	11	$4 + 29 - 28 + 0 + 90 = 95$	17	-
<b>Knooppunt 6</b>				
1	12	$18 + 15 - 6 + 0 + 90 = 117$	71	-
12	Lunch	$15 + 0 - 0 + 0 + 90 = 105$	17	-
Lunch	11	$15 + 15 - 28 + 0 + 90 = 92$	17	-
<b>Knooppunt 7</b>				
1	12	$25 + 20 - 6 + 0 + 120 = 159$	71	-
12	Lunch	$20 + 0 - 0 + 0 + 120 = 140$	17	-
Lunch	11	$20 + 38 - 28 + 0 + 120 = 150$	17	-
<b>Knooppunt 9</b>				
gemiddelde bezoektijd is te lang				

Tabel 22: Iteratie 1 – Route 2 – Invoegstap 2: berekeningen

### 3.5.2.3 Vervangstap

De vervangstap wordt eveneens gebruikt om de score te verhogen. De gebruikte inzichten zijn gebaseerd op literatuur van Sylejmani et al. (2012) en Souffriau et al. (2008). De gelijkheid  $\Delta Consumptietijd_j = t_{ij} + T_j + t_{jk} - t_{ix} - T_x - t_{xk}$  wordt in de vervangstap gebruikt om de verandering in consumptietijd te berekenen als knooppunt  $j$  ter vervanging dient van knooppunt  $x$ . Ook hier moet rekening gehouden worden met eventuele wijzigingen in de gemiddelde bezoektijd. Alle opgenomen knooppunten met een lagere score dan de niet-opgenomen knooppunten komen in aanmerking voor verwijdering. Voor elk niet-opgenomen knooppunt wordt de laagste wijziging in consumptietijd bepaald. Verslechtering van de oplossing is enkel toegelaten in het gebruik van tijd of budget aangezien het doel van de vervangstap expliciet het verhogen van de score is. Wanneer het verwijderen van één van de knooppunten voldoende tijd creëert om het niet-opgenomen knooppunt in te voegen, wordt de vervanging doorgevoerd. In geval van meerdere mogelijkheden wordt de beweging met de hoogste stijging in de score uitgevoerd. In geval van gelijke scores worden de beschikbare tijd en het budget in acht genomen.

In route 1 kunnen knooppunten 6, 7 en 9 nog worden opgenomen. Voor knooppunten 7 en 9 geldt echter dat hun score lager is dan de scores van de opgenomen knooppunten. Aangezien in deze stap de scoreverhoging centraal staat, worden zij dus niet als potentiële verbeteringen gezien. Knooppunt 6 heeft een hogere score in vergelijking met elk opgenomen knooppunt. Tabel 23 toont de berekeningen voor knooppunt 6. Bij een lunch- of dinerpauze wordt het volgende knooppunt gebruikt om de afstanden te bepalen. Dit knooppunt staat tussen haakjes. De tabel toont aan dat het vervangen van knooppunt 8 door knooppunt 6 het meest efficiënt is aangezien in die situatie de grootste verbetering in consumptietijd plaatsvindt. Knooppunt 8 kan succesvol vervangen worden door knooppunt 6. De score van route 1 bedraagt nu  $Z_{31} = 97$ . Knooppunt 6 wordt toegevoegd aan de taboelijst. De taboelijst bestaat nu uit knooppunten 6 en 12. Deze knooppunten mogen in de volgende stap aldus niet verwijderd worden. Tabel 24 toont de nieuwe route 1.

$i$	$j$	$k$	$x$	Consumptietijd $_j$
1	6	L	8	$18 + 90 + 0 - 49 - 120 - 0 = -61$
L (8)	6	10	4	$66 + 90 + 9 - 48 - 90 - 19 = 8$
4	6	D	10	$17 + 90 + 0 - 19 - 120 - 0 = -32$

Tabel 23: Iteratie 1 – Route 1 – Vervangstap 1 – Knooppunt 6

$i$	Wachttijd $_i$	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing $_i$
6	0	10u18	10u18	11u48	5	132
Lunch	12	11u48	12u00	13u00	15	120
4	0	13u17	13u17	14u47	16,5	133
10	0	15u06	15u06	17u06	16,5	174
Diner	54	17u06	18u00	19u30	46,5	120
13	-	19u51	-	-	-	189

Tabel 24: Iteratie 1 – Route 1 – Vervangstap 1

In route 2 kunnen knooppunten 3, 5, 7, 8 en 9 ter vervanging van een ander knooppunt dienen. Geen enkel van deze knooppunten heeft echter een hogere score dan knooppunt 11 of 12. Daarom vindt geen vervanging plaats in route 2.

Knooppunten 7, 8 en 9 kunnen worden toegevoegd aan route 1. Knooppunten 7 en 9 hebben echter een te lage score om als vervangend knooppunt te dienen. Knooppunt 8 heeft een hogere score dan zowel knooppunt 4 als 10. Daarom wordt in tabel 25 de verandering in consumptietijd in beide situaties berekend. De kleinste stijging in consumptietijd vindt plaats wanneer knooppunt 10 vervangen wordt door knooppunt 8. Knooppunt 10 wordt succesvol vervangen door knooppunt 8. Dit leidt tot een hogere score maar heeft ook een verhoging van de totale consumptietijd tot gevolg. In de vervangstap wordt een verslechtering in tijd toegestaan aangezien de scoreverhoging centraal staat. De score van route 1 bedraagt nu  $Z_{41} = 99$ . Knooppunt 8 wordt toegevoegd aan de taboelijst waardoor nu 6 en 8 op de taboelijst staan. Knooppunt 12 is verwijderd van de taboelijst aangezien de lengte van de taboelijst beperkt is tot twee knooppunten. Tabel 26 toont de nieuwe route 2.

$i$	$j$	$k$	$x$	Consumptietijd <sub><math>j</math></sub>
L (6)	8	10	4	$66 + 120 + 67 - 17 - 90 - 19 = 127$
4	8	D	10	$48 + 120 + 0 - 19 - 120 - 0 = 29$

Tabel 25: Iteratie 1 – Route 1 – Vervangstap 2 – Knoop punt 8

$i$	Wachttijd <sub><math>i</math></sub>	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing <sub><math>i</math></sub>
6	0	10u18	10u18	11u48	5	37
Lunch	12	11u48	12u00	13u00	15	25
4	0	13u17	13u17	14u47	16,5	25
8	0	15u35	15u35	17u35	21	25
Diner	25	17u35	18u00	19u30	51	120
13	-	20u19	-	-	-	161

Tabel 26: Iteratie 1 – Route 1 – Vervangstap 2

Enkel knooppunt 4 komt nog in aanmerking om vervangen te worden in route 1 aangezien de andere knooppunten op de taboelijst staan. Knooppunten 7 en 9 hebben een lagere score als knooppunt 4 dus zij worden buiten beschouwing gelaten. Knooppunt 10 bezit echter een hogere score. De berekening voor knooppunt 10 is terug te vinden in tabel 27. Wanneer knooppunt 4 vervangen wordt door knooppunt 10, bedraagt de score voor route 1 eveneens 99 (=  $Z_{51}$ ). Knooppunt 8 kan niet volledig bezocht worden waardoor de toerist maar 86,67% van zijn voldoening behaalt. Of deze oplossing aanvaard wordt, hangt af van de toename in tijd en budget. Tabel 28 geeft een overzicht van de verschillende stappen tot hier toe. De kost is in vervangstap 3 met 2,5 gedaald ten opzichte van vervangstap 2 en de tijd inclusief wachttijden is dezelfde gebleven. De oplossing van vervangstap 3 wordt aangenomen als de huidige oplossing omwille van de kostendaling. Tabel 29 toont de nieuwe route 1. Aangezien de vervangstap voor beide routes uitgeput is, wordt overgegaan naar de volgende stap: de ruilstap.

$i$	$j$	$k$	$x$	Consumptietijd <sub><math>j</math></sub>
L (6)	10	8	4	$9 + 120 + 67 - 17 - 90 - 48 = 41$

Tabel 27: Iteratie 1 – Route 1 – Vervangstap 3 – Knoop punt 10

		0	1	2	3	4	5
		Initialisatie	Invoeg	Invoeg	Vervang	Vervang	Vervang
Route 1	Z	82	-	-	97	99	99
	Totale kost	46	-	-	46,5	51	49,5
	Gebruikte tijd	617	-	-	591	619	619
Route 2	Z	42	87	-	-	-	-
	Totale kost	47,3	47,3	-	-	-	-
	Gebruikte tijd	538	538	-	-	-	-
<b>Totaal</b>	<b>Z</b>	<b>124</b>	<b>169</b>	<b>169</b>	<b>184</b>	<b>186</b>	<b>186</b>

Tabel 28: Samenvattende tabel tot en met stap 5

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$	$Maxverplaatsing_i$
6	0	10u18	10u18	11u48	5	142
Lunch	12	11u48	12u00	13u00	15	120
10	0	13u09	13u09	15u09	15	120
8	0	16u16	16u16	18u00	19,5	120
Diner	0	18u00	18u00	19u30	49,5	120
13	0	20u19	-	-	-	161

Tabel 29: Iteratie 1 – Route 1 – Vervangstap 3

### 3.5.2.4 Ruilstap

De ruilstap is gebaseerd op Sylejmani et al. (2012) en Vansteenwegen et al. (2011c). De procedure verwisselt één knooppunt uit een route met één knooppunt uit een andere route om zo de tijd te reduceren. Figuur 6 illustreert de werkwijze. Knooppunten  $i+1$  en  $j+1$  worden met elkaar verwisseld. De voorwaarde die gesteld wordt, is dat  $i$  en  $j+1$  niet ver uit elkaar liggen en zich elk op een andere route bevinden. Om 'ver' te definiëren, wordt willekeurig bepaald dat de knooppunten niet verder als 50 minuten reistijd uit elkaar mogen liggen.



Figuur 6: Ruilstap

De linkerzijde van figuur 7 geeft een overzicht van de huidige routes. De procedure start met het eerste knooppunt van de eerste route: knooppunt 6. Voor dit knooppunt wordt nagegaan welk knooppunt het dichtst gelegen is. Zowel knooppunt 11 als 12 bevinden zich op 15 minuten reisafstand van knooppunt 6. Normaliter kan knooppunt 12 niet in route 1 worden opgenomen omwille van zijn tijdvenster. Omwille van illustratieve redenen wordt voorlopig gesteld dat knooppunt 12 eveneens op maandag open is. Aangezien zowel knooppunt 11 als 12 op 15 minuten reisafstand liggen, dient berekend te worden welke keuze voor de grootste tijdsreductie zorgt. Aan de hand van de formule  $\Delta Consumptietijd_j = t_{ij} + T_j + t_{jk} - t_{ix} - T_x - t_{xk}$ , die reeds gebruikt werd in de vervangstap, wordt dit bepaald. De berekeningen bevinden zich in tabel 30. Uit de berekeningen kan worden afgeleid dat het voor route 1 voordeliger is knooppunt 12 te krijgen. Gebruikmakend van bovenstaande terminologie, wordt knooppunt 10 ( $i+1$ ) verwisseld met knooppunt 12 ( $j+1$ ). De lunch- en dinerpauzes worden buiten beschouwing gelaten. Figuur 7 toont aan de rechterzijde de situatie waarbij knooppunt 12 aldus verwisseld is met knooppunt 10.



Figuur 7: Iteratie 1 - Ruilstap 1 met  $6 = i$  en  $12 = j+1$  (illustratie)

Knooppunt 12	Route 1	15 + 90 + 51 - 9 - 120 - 67	<b>-40</b>
	Route 2	21 + 120 + 0 - 6 - 90 - 0	45
Knooppunt 11	Route 1	15 + 150 + 78 - 9 - 120 - 67	47
	Route 2	16 + 120 + 0 - 28 - 150 - 0	-42

Tabel 30: Iteratie 1 - Ruilstep 1 (illustratie): berekeningen

Uiteraard dient nagegaan of deze stap toegelaten is. Tabel 31 en 32 tonen respectievelijk route 1 en 2 na toepassing van de ruilstep. De aankomsttijd voor beide routes is dezelfde als in de vorige oplossing. In de wachttijd wordt een verschil opgemerkt. De toerist moet na de derde vervangstep in het totaal 98 minuten wachten (tabel 21; tabel 29). Na de eerste ruilstep moet de toerist slechts 87 minuten wat een aangenamere optie is. Route 1 en 2 uit ruilstep 1 worden gezien als de huidige oplossing met respectievelijke doelfunctiewaarden  $Z_{61} = 120$  en  $Z_{62} = 70$  waardoor  $Z_6 = 190$  bedraagt. De score is hoger dan in de vorige step omdat knooppunt 8 nu volledig bezocht kan worden. De taboelijst bestaat momenteel uit knooppunten 6 en 8. Aangezien in deze step verbindingen verbroken worden en als het ware twee knooppunten verwisseld worden, zal de taboelijst 'anders' aangevuld worden. Knooppunten 10 en 12 worden verwisseld in step 6. Deze knooppunten komen terecht op de taboelijst waardoor zowel knooppunt 6 als 8 van de taboelijst verdwijnen. Knooppunten 10 en 12 mogen in de volgende step niet gebruikt worden.

$i$	Wachttijd $_i$	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing $_i$
6	0	10u18	10u18	11u48	5	132
Lunch	12	11u48	12u00	13u00	15	120
12	0	13u15	13u15	14u45	15	84
8	0	15u36	15u36	17u36	19,5	84
Diner	24	17u36	18u00	19u30	49,5	120
13	-	20u19	-	-	-	161

Tabel 31: Iteratie 1 - Route 1 - Ruilstep 1 (illustratie)

$i$	Wachttijd $_i$	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing $_i$
10	0	10u21	10u21	12u21	0	36
Lunch	0	12u21	12u21	13u21	10	36
11	36	13u39	14u15	16u45	17,3	0
Diner	15	16u45	17u00	18u30	47,3	62
13	-	18u58	-	-	-	62

Tabel 32: Iteratie 1 - Route 2 - Ruilstep 1 (illustratie)

Omwille van de tijdvensterbeperving van knooppunt 12, kan bovenstaande step echter niet worden uitgevoerd. Daarom moet in eerste instantie reeds voor knooppunt 11 gekozen worden. De lunch- en dinerpauzes worden buiten beschouwing gelaten. Figuur 8 toont aan de rechterzijde de situatie waarbij knooppunt 11 aldus verwisseld is met knooppunt 10.



Figuur 8: Iteratie 1 - Ruilstep 1 met  $6 = i$  en  $11 = j+1$

Uiteraard dient nagegaan of deze stap toegelaten is. Tabel 33 toont route 1 na toepassing van de ruilstap. Het rood gemarkeerde vak duidt erop dat de eerste ruilstap leidt tot een niet toegelaten oplossing. De taboelijst bestaat nog steeds uit knooppunten 6 en 8.

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$	$Maxverplaatsing_i$
6	0	10u18	10u18	11u48	5	
Lunch	12	11u48	12u00	13u00	15	
11	0	13u15	14u15	16u45	22,3	
8	0	18u03				

Tabel 33: Iteratie 1 – Route 1 – Ruilstap 1

Het volgende knooppunt dat wordt beschouwd, is het knooppunt volgend op knooppunt 6 in route 1, namelijk knooppunt 10. Om dezelfde reden als voorheen kan enkel knooppunt 11 als  $j+1$  beschouwd worden. Knooppunten 8 en 11 dienen verwisseld worden, wat getoond wordt in figuur 9. Knooppunt 8 bevindt zich echter op de taboelijst waardoor dit knooppunt niet verwisseld mag worden.



Figuur 9: Iteratie 1 - Ruilstap 2 met  $10 = i$  en  $11 = j+1$

Het derde knooppunt in route 1, knooppunt 8, ligt ver verwijderd van de meeste knooppunten. De knooppunten die op 50 minuten reistijd of minder liggen, zijn enkel knooppunten 2, 4 en 5. Deze zijn niet opgenomen in route 2 en bijgevolg kan de derde ruilstap niet worden toegepast op het praktijkvoorbeeld. Alle knooppunten zijn in beschouwing genomen, bijgevolg wordt overgegaan naar de volgende en laatste stap van iteratie 1.

### 3.5.2.5 Verwijderstap

Na het verhogen van de score en het reduceren van de tijd, wordt de verwijderstap gebruikt om aan lokale optima te ontsnappen. In deze stap zullen een of meerdere knooppunten verwijderd worden. Voor elk van de opgenomen knooppunten wordt  $verwijder_j$  berekend met

$$verwijder_j = \frac{S_j}{\frac{consumptietijd_j}{gebruiktetijd} + \frac{e_j}{gebruiktbudget}}$$

waar  $S_j$  de score vormt van knooppunt  $j$ ,  $consumptietijd_j$  de tijd die  $j$  gebruikt in de route,  $e_j$  de kostprijs van  $j$ ,  $gebruiktetijd$  de tijd die de toerist nodig heeft om de huidige route te verrichten en  $gebruiktbudget$  het budget dat de toerist reeds heeft opgedaan in de desbetreffende route.

Gebaseerd op Vansteenwegen et al. (2009), worden twee integere waarden geregistreerd.  $V_d$  duidt aan hoeveel opeenvolgende bezoeken verwijderd dienen worden in route  $d$  en  $S_d$  geeft aan waar het verwijderen moet beginnen in route  $d$ . De plaats van het knooppunt met de laagste  $verwijder_j$  in route  $d$  zal de beginwaarde vormen voor  $S_d$ . Wanneer de laatste locatie bereikt wordt, gaat de verwijderprocedure verder vanaf de start van de route. Gezien de

lengte van beide routes verschillend is, zal  $S_d$  in elke route een andere waarde aannemen naarmate het algoritme verdergaat. Na het verwijderen worden alle knooppunten gevraagd om onnodig wachten te vermijden als dit mogelijk is binnen de tijdsvensterbeperkingen. In de verwijderstap wordt geen gebruik gemaakt van de taboelijst.

$V_d$  wordt initieel willekeurig gekozen:  $V_1 = 2$  en  $V_2 = 1$ . Ter illustratie wordt  $verwijder_6$  uitgeschreven:

$$verwijder_6 = \frac{45}{\frac{18+90+0}{619} + \frac{5}{49,5}} = 163,348.$$

Voor alle andere opgenomen knooppunten wordt deze waarde eveneens berekend, wat leidt tot de resultaten in tabel 34.  $S_1$  krijgt bijgevolg de waarde 3 en  $S_2$  de waarde 2. Dit wil zeggen dat in route 1 twee knooppunten verwijderd worden startend vanaf knooppunt 8 en in route 2 één knooppunt startend vanaf knooppunt 11. Het verwijderen van de desbetreffende knooppunten leidt tot de huidige routes 1 en 2, zoals weergegeven in respectievelijk tabel 35 en 36, met een totale doelfunctiewaarde van  $Z_{10} = 73$ .

Route	Knooppunt $j$	Verwijder $_j$
1	6	163,348
	10	88,429
	<b>8</b>	<b>66,156</b>
2	12	252,188
	<b>11</b>	<b>86,564</b>

Tabel 34: Iteratie 1 – Verwijderstap 1: berekeningen

$i$	Wachttijd $_i$	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing $_i$
10	0	10u21	10u21	12u21	0	99
Lunch	0	12u21	12u21	13u21	10	99
Diner	279	13u21	18u00	19u30	40	120
13	-	19u51	-	-	-	189

Tabel 35: Iteratie 1 – Route 1 – Verwijderstap 1

$i$	Wachttijd $_i$	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing $_i$
12	54	10u06	11u00	12u30	0	60
Lunch	0	12u30	12u30	13u30	10	60
Diner	210	13u30	17u00	18u30	40	84
13	-	18u36	-	-	-	84

Tabel 36: Iteratie 1 – Route 2 – Verwijderstap 1

Nadat de verwijderstap eenmaal uitgevoerd is, worden de parameters  $V_d$  en  $S_d$  aangepast.  $S_d$  wordt verhoogd met  $V_d$  en  $V_d$  met 1. Wanneer  $S_d$  gelijk is aan of groter dan de lengte van de kortste route, dan wordt deze lengte (in termen van aantal knooppunten, uitgezonderd start- en eindlocatie, lunch en diner) van  $S_d$  afgetrokken. Wanneer  $V_d$  gelijk is aan of groter is dan het aantal knooppunten opgenomen in route  $d$ , wordt deze parameter gelijkgesteld aan 1. Dit zorgt ervoor dat  $S_1 = 5$ ,  $S_2 = 3$ ,  $V_1 = 3$  en  $V_2 = 2$ . Of de waarden nog veranderd moeten worden omwille van bovenstaande aanwijzingen, zal afhangen van de oplossing waarop de



verwijderstap moet worden toegepast. De heuristisch start nu opnieuw met de invoegstap, de vervangstap, de ruilstap en de verwijderstap tot aan het stopcriterium voldaan is. Tabel 37 geeft een overzicht van stap 5 tot en met stap 9.

		5	6	7	8	9
		Vervang	Ruil	Ruil	Ruil	Verwijder
Route 1	Z	99	-	-	-	28
	Totale kost	49,5	-	-	-	40
	Gebruikte tijd	619	-	-	-	591
Route 2	Z	87	-	-	-	45
	Totale kost	47,3	-	-	-	40
	Gebruikte tijd	538	-	-	-	516
<b>Totaal</b>	<b>Z</b>	<b>186</b>	<b>186</b>	<b>186</b>	<b>186</b>	<b>73</b>

Tabel 37: Samenvattende tabel stap 5 tot en met stap 10

### 3.5.2.6 Stopcriterium

Het stopcriterium kan zoals eerder aangehaald op verschillende manieren gedefinieerd worden. In het praktijkvoorbeeld stopt het algoritme als 15 stappen zijn uitgevoerd of 10 stappen plaatsgevonden hebben waarin geen verbetering van de huidige oplossing geconstateerd is. In iteratie 1 werden reeds negen stappen uitgevoerd waarvan zes stappen niet hebben geleid tot een verbetering in de doelfunctiewaarde. Aangezien aan geen enkel stopcriterium voldaan is, start iteratie 2 beginnend met de invoegstap. De taboelijst wordt na elke iteratie leeggemaakt.

De knooppunten die in route 1 kunnen worden toegevoegd zijn knooppunten 4, 6, 7, 8, 9 en 11. De werkwijze is identiek als in de eerder uitgevoerde invoegstap. Tabel 38 toont de berekeningen voor de verschillende knooppunten. De berekeningen van de ratio's maken duidelijk dat knooppunt 6 de hoogste meerwaarde aan route 1 kan bieden. Dit knooppunt wordt ingevoerd. Aan de hand van formules (7) tot en met (12) uit de sectie 3.5.2.2 worden de waarden herberekend. Tabel 39 toont route 1 na het uitvoeren van de eerste invoegstap van iteratie 2. Knooppunt 6 wordt toegevoegd aan de taboelijst en mag niet verwijderd worden in de volgende stap.

$i$	$k$	$Consumptietijd_j$	$Wachttijd_k + Maxverplaatsing_k$	$Ratio_j$
Knooppunt 4				
1	10	$5 + 19 - 21 + 0 + 90 = 93$	99	2315,876
10	Lunch	$19 + 0 - 0 + 0 + 90 = 109$	99	-
Lunch	Diner	$19 + 0 - 0 + 0 + 90 = 109$	399	-
Diner	13	tijdvensterbeperking knooppunt 4		
Knooppunt 6				
1	10	$18 + 9 - 21 + 0 + 90 = 96$	99	5446,552
10	Lunch	$9 + 0 - 0 + 0 + 90 = 99$	99	-
Lunch	Diner	$9 + 0 - 0 + 0 + 90 = 99$	399	-
Diner	13	tijdvensterbeperking knooppunt 6		

Knooppunt 7				
1	10	$25 + 25 - 21 + 0 + 120 = 149$	99	-
10	Lunch	$25 + 0 - 0 + 0 + 120 = 145$	99	-
Lunch	Diner	$25 + 0 - 0 + 0 + 120 = 145$	399	1707,393
Diner	13	tijdvensterbeperking knooppunt 7		
Knooppunt 8				
1	10	$49 + 67 - 21 + 0 + 120 = 215$	99	-
10	Lunch	$67 + 0 - 0 + 0 + 120 = 187$	99	-
Lunch	Diner	$67 + 0 - 0 + 0 + 120 = 187$	399	1637,636
Diner	13	tijdvensterbeperking knooppunt 8		
Knooppunt 9				
1	10	$23 + 14 - 21 + 0 + 300 = 316$	99	-
10	Lunch	$14 + 0 - 0 + 0 + 300 = 314$	99	-
Lunch	Diner	tijdvensterbeperking knooppunt 9		
Diner	13	tijdvensterbeperking knooppunt 9		
Knooppunt 11				
1	10	$28 + 18 - 21 + 227 + 150 = 402$	99	-
10	Lunch	$18 + 0 - 0 + 96 + 150 = 264$	99	-
Lunch	Diner	$18 + 0 - 0 + 36 + 150 = 204$	399	2597,055
Diner	13	tijdsbudget toerist		

Tabel 38: Iteratie 2 – Route 1 – Invoegstap 1: berekeningen

$i$	Wachttijd $_i$	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing $_i$
6	0	10u18	10u18	11u48	5	3
10	0	11u57	11u57	13u57	5	3
Lunch	0	13u57	13u57	14u57	15	3
Diner	183	14u57	18u00	19u30	45	120
13	-	19u51	-	-	-	189

Tabel 39: Iteratie 2 – Route 1 – Invoegstap 1

Voor route 2 geldt dezelfde werkwijze. Aan route 2 kunnen knooppunten 3, 4, 5, 7, 8, 9 en 11 worden toegevoegd. Tabel 40 geeft de ratio's weer voor elk knooppunt, uitgezonderd knooppunt 9, wanneer dit knooppunt zou worden toegevoegd tussen knooppunten  $i$  en  $k$ . Uit deze tabel blijkt dat knooppunt 5 de hoogste ratio geniet, wat leidt tot de resultaten in tabel 41. Deze tabel toont route 2 na uitvoering van invoegstap 1 van iteratie 2. De doelfunctiewaarde bedraagt 149 (=  $Z_{10}$ ).

$i$	$k$	Knooppunt $j$	Ratio $_j$
L	D	3	1380,496
1	12	4	1849,151
L	D	5	2200,184
L	D	7	1314,943
L	D	8	1403,226
L	D	11	219,908

Tabel 40: Iteratie 2 – Route 2 – Invoegstap 1: berekeningen

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$	$Maxverplaatsing_i$
12	54	10u06	11u00	12u30	0	60
Lunch	0	12u30	12u30	13u30	10	60
5	0	13u34	13u34	15u04	15	116
Diner	116	15u04	17u00	18u30	45	80
13	-	18u40	-	-	-	80

Tabel 41: Iteratie 2 – Route 2 – Invoegstap 1

In een tweede invoegstap wordt nagegaan of één van de knooppunten 4, 7, 8, 9 en 11 toegevoegd kan worden aan route 1. Voor elk van deze knooppunten, uitgezonderd voor knooppunten 9 en 7 omwille van de tijdvensterbependingen, geldt dat de enige mogelijkheid tot invoeging bestaat tussen de lunchpauze en het diner. Uit de berekening van de bijbehorende ratio's blijkt dat knooppunt 11 de hoogste ratio heeft met een waarde van 1789,866. Tabel 42 toont route 1 na uitvoering van de tweede invoegstap, aldus na toevoeging van knooppunt 11. Knooppunten 3, 4, 7, 8 en 9 komen in aanmerking voor invoeging in route 2. Knooppunt 4 wordt toegevoegd tussen knooppunten 1 en 12 ( $ratio_4 = 1358,491$ ). Het resultaat bevindt zich in tabel 43. De totale doelfunctiewaarde bedraagt 215 ( $= Z_{11}$ ).

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$	$Maxverplaatsing_i$
6	0	10u18	10u18	11u48	5	3
10	0	11u57	11u57	13u57	5	3
Lunch	0	13u57	13u57	14u57	15	3
11	90	15u15	16u45	19u15	22,3	0
Diner	0	19u15	19u15	20u45	52,3	45
13		21u13	-	-	-	107

Tabel 42: Iteratie 2 – Route 1 – Invoegstap 2

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$	$Maxverplaatsing_i$
4	0	10u05	10u05	11u35	1,5	23
12	0	11u37	11u37	13u07	1,5	23
Lunch	0	13u07	13u07	14u07	11,5	23
5	0	14u11	14u11	15u41	16,5	79
Diner	79	15u41	17u00	18u30	46,5	80
13	-	18u40	-	-	-	80

Tabel 43: Iteratie 2 – Route 2 – Invoegstap 2

Voor route 1 zijn geen invoegopties meer toegelaten. Aan route 2 kan knooppunt 8 worden toegevoegd tussen het bezoek aan knooppunt 5 en het diner. Tabel 44 geeft het resultaat weer. Knooppunt 8 kan niet volledig bezocht worden. Om na te gaan wat de maximale bezoektijd is, wordt teruggerekend vanaf de uiterlijke aankomst op het hotel. Het knooppunt kan gedurende 70 minuten bezocht worden. Bijgevolg verhoogt de totale doelfunctiewaarde met  $(70/120)*23 \cong 13$  ( $Z_{12} = 228$ ).

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$	$Maxverplaatsing_i$
4	0	10u05	10u05	11u35	1,5	0
12	0	11u37	11u37	13u07	1,5	0
Lunch	0	13u07	13u07	14u07	11,5	0
5	0	14u11	14u11	15u41	16,5	0
8	0	16u31	16u31	17u41	21	0
Diner	0	17u41	17u41	19u11	51	0
13	-	20u00	-	-	-	0

Tabel 44: Iteratie 2 – Route 2 – Invoegstap 3

Ook voor route 2 zijn de resterende invoegmogelijkheden uitgeput. De oplossing van de vierde invoegstap is identiek aan de oplossingen in tabel 42 (route 1) en 44 (route 2). Geen verbeteringen zijn mogelijk.

De volgende stap in iteratie 2 is de vervangstap. De knooppunten die kunnen dienen ter vervanging van één van de andere knooppunten in route 1, zijn knooppunten 7 en 9. Deze knooppunten hebben beiden een lagere score dan elk van de opgenomen knooppunten. Bijgevolg heeft de eerste vervangstap voor route 1 geen meerwaarde. Knooppunten 3, 7 en 9 kunnen dienen ter vervanging voor een knooppunt in route 2. Knooppunten 4 en 8 hebben een gelijke of lagere score als knooppunt 3. Knooppunten 7 en 9 hebben beiden een hogere score als knooppunt 8 aangezien knooppunt 8 niet volledig bezocht wordt. Tabel 45 geeft de berekeningen weer voor de drie knooppunten. Aangezien route 2, zoals weergegeven in tabel 44, reeds de volledige beschikbare tijd van de toerist gebruikt, kan knooppunt 4 niet vervangen worden door knooppunt 3. Knooppunt 8 kan niet vervangen worden door één van de andere knooppunten omwille van hun tijdvensterbeperking. De vervangstap kan dus in geen enkele route worden uitgevoerd.

$i$	$j$	$k$	$x$	$Consumptietijd_j$
1	3	12	4	$9 + 3 + 90 - 5 - 2 - 90 = 5$
5	3	D	8	tijdvensterbeperking knooppunt 3
5	7	D	8	tijdvensterbeperking knooppunt 7
5	9	D	8	tijdvensterbeperking knooppunt 9

Tabel 45: Iteratie 2 – Route 2 – Vervangstap 1: berekeningen

De ruilstap volgt op de vervangstap. Het eerste knooppunt dat in aanmerking komt, is knooppunt 6 ( $i$ ). Knooppunt 12 ligt op 15 minuten reisafstand van knooppunt 6. Knooppunt 12 kan echter niet geruild worden omwille van zijn tijdvensterbeperking. Het volgende dichtst gelegen knooppunt is knooppunt 5 op 16 minuten reisafstand. Dit knooppunt is gesloten op maandag waardoor ook deze ruil niet toegelaten is. Knooppunt 4 ligt op 17 minuten reisafstand van knooppunt 6 en is open op maandag. Knooppunt 4 ( $j+1$ ) wordt verwisseld met knooppunt 10 ( $i+1$ ). Geen van beide knooppunten bevindt zich op de taboelijst. Figuur 10 schetst de ruilstap. Tabel 46 en 47 tonen route 1 respectievelijk route 2 na uitvoering van de eerste ruilstap van iteratie 2. Uit tabel 47 blijkt dat deze stap niet toegelaten is omwille van de tijdvensterbeperking van de lunchpauze aangegeven door de toerist. De huidige oplossing blijft degene in tabel 42 en 44.

1 - 6 - 10 - L - 11 - D - 13



1 - 6 - 4 - L - 11 - D - 13

1 - 4 - 12 - L - 5 - 8 - D - 13

1 - 10 - 12 - L - 5 - 8 - D - 13

Figuur 10: Iteratie 2 - Ruilstap 1 met  $6 = i$  en  $4 = j+1$

$i$	Wachttijd $_i$	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing $_i$
6	0	10u18	10u18	11u48	5	25
4	0	12u05	12u05	13u35	6,5	25
Lunch	0	13u35	13u35	14u35	16,5	25
11	101	15u04	16u45	19u15	23,8	0
Diner	0	19u15	19u15	20u45	53,8	45
13	-	21u13	-	-	-	107

Tabel 46: Iteratie 2 - Route 1 - Ruilstap 1

$i$	Wachttijd $_i$	$a_i$	$s_i$	$l_i$	$E$	Maxverplaatsing $_i$
10	0	10u21	10u21	12u21	0	
12	0	12u37	12u37	14u07	0	
Lunch		14u07				

Tabel 47: Iteratie 2 - Route 2 - Ruilstap 1

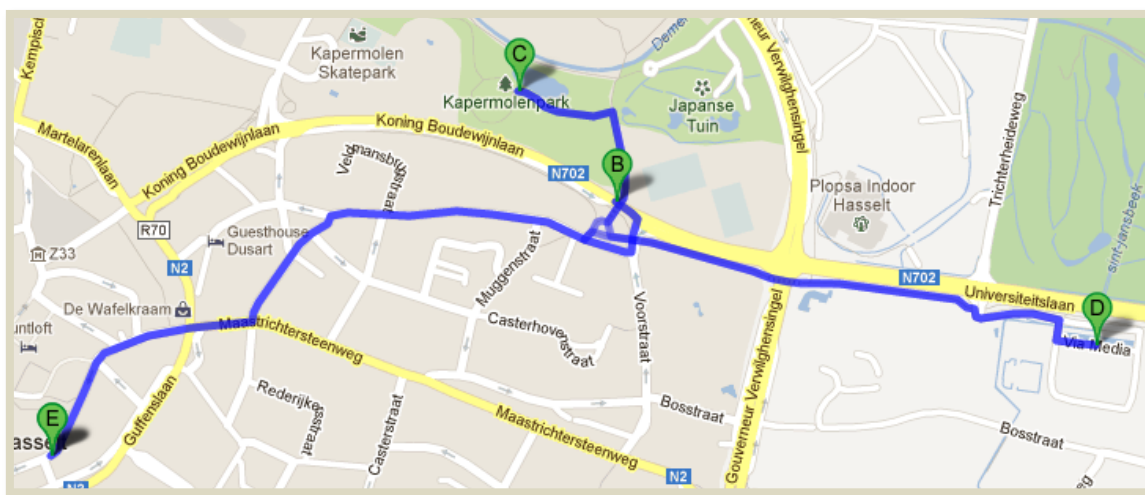
Tabel 48 geeft een overzicht van stap 9 tot en met stap 15. Aangezien het stopcriterium van 15 stappen bereikt is, stopt het algoritme. Negen stappen hebben tot hier toe niet geleid tot een verbetering in de doelfunctiewaarde. De uiteindelijke oplossing is de oplossing bekomen in stap 12 met een totale doelfunctiewaarde van 228. Zoals eerder vermeld, is dit niet noodzakelijk de optimale oplossing. De kans bestaat dat het gaat om een lokaal optimum. In sectie 3.6 wordt de voorgestelde route voor de toerist schematisch beschreven en afgebeeld.

		9	10	11	12	13	14	15
		Verwijder	Invoeg	Invoeg	Invoeg	Invoeg	Vervang	Ruil
Route 1	Z	28	73	115	-	-	-	-
	Totale kost	40	45	52,3	-	-	-	-
	Gebruikte tijd	591	591	-	-	-	-	-
Route 2	Z	45	76	100	113	-	-	-
	Totale kost	40	45	46,5	51	-	-	-
	Gebruikte tijd	516	520	520	600	-	-	-
<b>Totaal</b>	<b>Z</b>	<b>73</b>	<b>149</b>	<b>215</b>	<b>228</b>	<b>228</b>	<b>228</b>	<b>228</b>

Tabel 48: Samenvattende tabel stap 9 tot en met stap 15

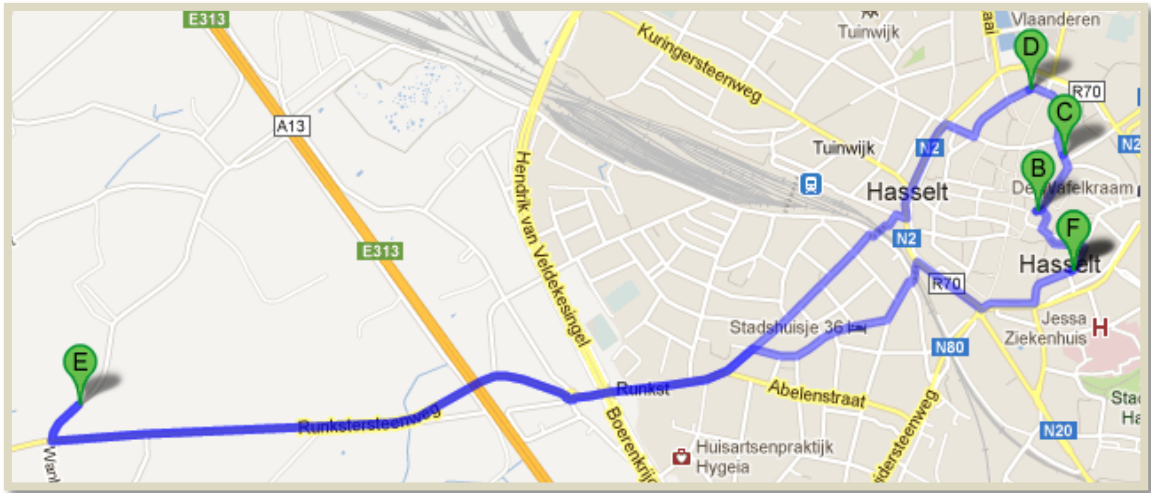
### 3.6 Voorstel persoonlijke route

De voorgestelde route voor maandag 8 april 2013 wordt getoond in figuur 11. De toerist vertrekt al wandelend om 10u00 van het Radisson Blu Hotel naar de Japanse Tuin waar hij aankomt om 10u18. Na een bezoek van 90 minuten aan de Japanse Tuin vertrekt de toerist om 11u48 richting Kapermolenpark. Daar start hij zijn bezoek van twee uur om 11u57. Om 13u57 luncht de toerist ter plaatse gedurende één uur. Om 14u57 vertrekt hij richting Kinopolis waar hij 90 minuten moet wachten vooraleer de film start om 16u45. De film duurt 150 minuten en eindigt aldus om 19u15. De toerist dineert gedurende 90 minuten waarna hij om 20u45 terug richting hotel keert. Hij komt in het Radisson Blu Hotel aan om 21u13. De totale kostprijs van de route bedraagt €52,30 en de toerist dient 90 minuten in het totaal te wachten. De toerist kan uiteraard zelf bepalen om wijzigingen aan deze route aan te passen. In figuur 11 wordt volgende notatie gebruikt: E = Radisson Blu Hotel, B = Japanse tuin, C = Kapermolenpark en D = Kinopolis.



Figuur 11: Voorstel route 1

Figuur 12 toont de voorgestelde route voor dinsdag 9 april 2013. De toerist vertrekt om 10u00 in het Radisson Blu Hotel. Hij wandelt vijf minuten naar de beiaardtoren die hij gedurende 90 minuten bezoekt. Om 11u35 zet hij zijn tocht verder richting de Z33 op 2 minuten wandelafstand van de beiaardtoren. Ook dit bezoek duurt 90 minuten. Om 13u07 start de lunchpauze van één uur. Om 14u11 komt de toerist aan bij het Modemuseum waar hij een rondleiding van 90 minuten krijgt. Om 15u41 zet hij zijn tocht verder naar het Park Natuur en Cultuur dat op 50 minuten wandelafstand ligt. Dit kan hij niet voor de volledige 120 minuten bezoeken. 70 minuten wandelt de toerist rond in het park waarna hij start aan zijn diner dat 90 minuten in beslag neemt. Om 19u11 vertrekt hij richting hotel dat hij om 20u00 bereikt. De kostprijs van de volledige route bedraagt €51 en de toerist moet op geen enkel moment wachten. De toerist kan uiteraard zelf bepalen om wijzigingen aan deze route aan te passen. Figuur 12 gebruikt volgende notatie: F = Radisson Blu Hotel, B = Beiaardtoren, C = Z33, D = Modemuseum en E = Park Natuur en Cultuur.



Figuur 12: Voorstel route 2

## Hoofdstuk 4 Kritische bedenkingen en aanbevelingen voor verder onderzoek

Eerst wordt nagegaan in hoeverre de uiteindelijke oplossing logisch en optimaal is. Met uitzondering van knooppunt 7 zijn alle POI die in één van de drie scores een bonus hebben ontvangen, opgenomen. Knooppunt 7 krijgt een bonus omwille van de goede match met het kernwoord 'rust'. De type- en categoriescore van dit knooppunt zijn echter laag waardoor het niet opnemen verantwoord is. Knooppunten 9 en 3 zijn eveneens in geen enkele route opgenomen. De lange gemiddelde bezoektijd van knooppunt 9 zorgt ervoor dat knooppunt 9 niet kan worden opgenomen. Knooppunt 3 heeft een even hoge score als knooppunt 4 maar zoals reeds in de eerste vervangstap van iteratie 2 naar voren kwam, kan de opname van dit knooppunt ter vervanging van een ander knooppunt niet tot een hogere doelfunctiewaarde leiden. De oplossing is niet optimaal. In tabel 49 en 50 wordt een andere mogelijke oplossing gegeven voor respectievelijk route 1 en 2, gebaseerd op eigen vermoedens. De totale doelfunctiewaarde bedraagt 255, wat hoger is dan die van de uiteindelijke oplossing (= 228). De doelfunctiewaarde van de uiteindelijke oplossing komt zeker dicht in de buurt.

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$
4	0	10u05	10u05	11u35	1,5
10	0	11u54	11u54	13u54	1,5
Lunch	0	13u54	13u54	14u54	11,5
6	0	15u03	15u03	17u33	16,5
Diner	27	17u33	18u00	19u30	46,5
11	0	19u45	19u45	22u15	53,8
13	-	22u43	-	-	-

Tabel 49: Alternatieve route 1

$i$	$Wachttijd_i$	$a_i$	$s_i$	$l_i$	$E$
5	0	10u10	10u10	11u40	5
3	0	11u43	11u43	13u13	9,5
Lunch	0	13u13	13u13	14u13	19,5
12	0	14u16	14u16	15u46	19,5
8	0	16u37	16u37	17u41	24
D	0	17u41	17u41	19u11	54
13	-	20u00	-	-	-

Tabel 50: Alternatieve route 2

Het gebruikte algoritme bestaat uit een groot aantal willekeurig gekozen data. De keuze van de gebruikte POI is logischerwijze zeer bepalend voor de uiteindelijke route. De gemiddelde bezoektijd van elk POI is gebaseerd op eigen ervaring en aanwijzingen van de toeristische dienst. Voor elke toerist zal de bezoektijd verschillen aangezien eenieder een andere ervaring beleeft bij elk POI. Het bepalen van de verschillende types en categorieën is eveneens voor interpretatie vatbaar alsook de rating van elk POI op elke categorie ook al is dit gebaseerd op objectieve informatie.



Wanneer het algoritme toegepast wordt op gegevens van een andere toerist, zullen verschillende elementen wijzigen. Een bezoek aan de stad Hasselt op andere dagen zal leiden tot andere openingsuren. Elke persoon heeft andere voorkeuren inzake het interval waarin de eetpauzes moeten plaatsvinden. Wijzigingen in beschikbare tijd en budget kunnen zich voordoen. De keuze van het hotel kan verschillen. De interessescores zullen volledig of gedeeltelijk anders zijn (oud versus jong, al dan niet artistiek, man versus vrouw) omwille van andere interesses in de verschillende types, categorieën en verschil in kernwoordkeuze.

Het algoritme gaat er van uit dat de eetpauzes plaatsvinden op de bestemming van het laatst bezochte knooppunt. Of daar effectief een eetgelegenheid is, is waarschijnlijk maar kan niet met zekerheid gewaarborgd worden.

De toelating van een lagere gemiddelde bezoektijd bij bepaalde types geeft een andere invulling aan het probleem dan wanneer elk POI bezocht moet worden gedurende de volledige gemiddelde bezoekduur. Deze keuze is aan willekeurigheid onderworpen. In het praktijkvoorbeeld wordt de helft van de gemiddelde bezoekduur gebruikt maar dit kan verlengd of verkort worden.

Naast een *tabu search* algoritme kan gekozen worden voor een ander algoritme, zoals bijvoorbeeld een genetisch algoritme of *simulated annealing*. Ook de verschillende stappen (invoegstap, vervangstap, ruilstap, verwijderstap) in de iteratie zijn bepalend voor de uiteindelijke oplossing. Het stopcriterium wordt bepaald op basis van eigen indrukken.

Uit bovenstaande bevindingen blijkt dat het algoritme veel willekeurigheden bevat die elk een invloed kunnen hebben op de uiteindelijke oplossing.

In de formulering van het probleem kan in verder onderzoek rekening gehouden worden met een aantal bijkomende factoren. De eerste factor omvat de weersomstandigheden. Wanneer het regent of de temperaturen onaangenaam zijn, is een museumbezoek geschikter dan een bezoek aan een park. Ten tweede kan de toegankelijkheid een rol spelen voor toeristen met een handicap. Ten derde hebben toeristen naast hun eetpauzes vaak nood aan een rustmoment, zeker wanneer een route continu het bezoeken van POI omvat zoals route 2 in het praktijkvoorbeeld. Ten vierde kan een beperking worden toegevoegd dat de toerist maximaal twee musea per dag of per reis wil bezoeken. Ook kan de toeristische dienst verplichte POI opgeven bijvoorbeeld de Japanse tuin. Naast verplichte knooppunten kan eveneens een verplicht bezoek aan een bepaald type of binnen een bepaalde categorie worden opgenomen zoals het bezoeken van een POI van type 'Sport en spel' (Gavalas et al., 2012).

Toeristen genieten vaak van een wandeling door een straat waar bijvoorbeeld oude gebouwen aanwezig zijn. Het kan dus efficiënter zijn om een andere route tussen twee knooppunten te volgen dan degene die het kortst in afstand of tijd is. Er wordt geen rekening gehouden met verschillende vervoersmogelijkheden. Het algoritme maakt enkel gebruik van de wandeltijd in minuten van het ene knooppunt naar het andere. De toerist kan tevens kiezen om gebruik te maken van een wagen of het openbaar vervoer. Dan is het probleem tijdsafhankelijk en wordt dit geclassificeerd als een *time-dependent team orienteering problem with time windows* (TDTOPTW) (Gavalas et al., 2012). Wanneer de toerist kan kiezen tussen verschillende vervoersmogelijkheden, moet onderscheid gemaakt worden tussen de afstand en de tijd tussen twee knooppunten. Bij *cheapest insertion* bijvoorbeeld kan gekozen worden om knooppunten in te voegen op basis van de kortste afstand of op basis van de kortste tijd. In het praktijkvoorbeeld hebben beide opties hetzelfde effect aangezien geen andere vervoersmogelijkheden worden voorgesteld aan de toerist. De tijd tussen twee knooppunten wordt dus altijd op dezelfde manier berekend. Het algoritme veronderstelt een symmetrische reistijd tussen twee knooppunten. Dit kan soms verschillend zijn in de realiteit omwille van bijvoorbeeld eenrichtingsverkeer.

De meeste toeristische attracties zijn gedurende de hele dag geopend. Bijgevolg zullen de openings- en sluitingstijden van de attracties nauwelijks verschillen. In bepaalde gevallen is het daarom mogelijk het TOPTW te reduceren tot het TOP (Gavalas et al., 2012).

Het selecteren van een restaurant kan worden gezien als een bijkomend probleem waarbij de selectie gebaseerd is op basis van gebruikersvoorkeuren (budget, dieetvoorkeuren, wereldkeuken) en restaurantkenmerken (menu, prijslijst, openingsuren) (Gavalas et al., 2012).

In het algoritme wordt het Radisson Blu hotel gekozen, op aanraden van de toeristische dienst. Het hotel kan echter ook geselecteerd worden op basis van het profiel van de gebruiker. Voor een toerist is het vaak moeilijk te bepalen in welk hotel hij zal logeren de komende dagen aangezien een 'goed' hotel afhankelijk is van veel factoren zoals nabijgelegen openbaar vervoer, persoonskenmerken (jong versus oud), kostprijs en omgeving (luid versus rustiek). Wanneer de reis bestaat uit het bezoeken van verschillende steden of een regio is dit eens zo moeilijk omdat van accommodatie gewisseld moet worden (Gavalas et al., 2012). Zhu et al. (2010), Vansteenwegen et al. (2011a) en Castro et al. (2013) nemen hotelselectie in acht.

De *City Trip Planner* van Vansteenwegen et al. (2011b) geeft toeristen enkel de mogelijkheid om routes te plannen binnen de steden Antwerpen, Brugge, Gent, Leuven en Mechelen. Dit zou uitgebreid kunnen worden naar Hasselt. Hetgeen nodig is, is een databank met alle nodige gegevens die regelmatig hernieuwd wordt.

Tot nu toe wordt het concept van een *personalized electronic tourist guide* enkel op één persoon toegepast. Dit kan uitgebreid worden naar meerdere personen omdat personen vaak niet alleen reizen. Het interesseprofiel kan afgeleid worden van de individuele interesseprofielen. De reis dient zowel individuele vrijheid als groepservaring te geven (Kramer et al., 2006). Een andere mogelijkheid is om toeristen op bepaalde momenten op te splitsen. Op die manier kan ieder zijn eigen voorkeur genieten.

Zoals eerder aangehaald, kunnen er door onvoorziene omstandigheden wijzigingen plaatsvinden in de route van een toerist. Hoewel de kans dat dit voorkomt groot is, wordt dit niet in rekening gebracht in het algoritme. Dynamische herberekening van de route kan dit probleem oplossen. POI die reeds bezocht zijn, worden dan buiten beschouwing gelaten. Een nieuwe route vanaf de huidige positie van de toerist wordt voorgesteld voor de dag zelf alsook voor eventueel komende dagen (Gavalas et al., 2012).

Wanneer algoritmen worden ontwikkeld voor web- of mobiele applicaties, zijn snel efficiënte oplossingen nodig voor de TTDP. Parallele berekening wordt gezien als een veelbelovend item in het vinden naar goede oplossingen op snelle wijze. Zo kan bijvoorbeeld het volledige oplossingsgebied worden opgedeeld in verschillende delen, waar in elk deel in parallel een heuristiek wordt gerund. Ook kunnen verschillende heuristieken gebruikt worden om een en hetzelfde oplossingsgebied uit te kammen, startend van dezelfde of een andere initiële oplossing. Deze kunnen onafhankelijk van elkaar te werk gaan of samenwerken en informatie uitwisselen over de voortgang en de eventueel goede oplossingen die tot dan toe gevonden zijn (Gavalas et al., 2012).

De best presterende algoritmen tonen aan dat niet toegelaten oplossingen toegelaten moeten worden tijdens de zoekprocedure en dat meer onderzoek gedaan moet worden naar de efficiëntie van de lokale zoekmethoden (Vansteenwegen et al., 2011c). 2-opt bewegingen toevoegen aan de heuristiek zorgt voor een significante vermindering van de reistijd. Het gebruikte algoritme verwisselt enkel knooppunten tussen twee verschillende routes en niet binnen een en dezelfde route. Deze stap zou de lengte van de route kunnen verminderen en ruimte maken voor nieuwe knooppunten totdat een nieuw lokaal optimum bereikt is. Twee of meer activiteiten gelijktijdig invoegen is een veelbelovende beweging volgens Vansteenwegen et al. (2009).

## Hoofdstuk 5      Algemene conclusies

De problematiek rond het oplossen van toeristische problemen is actueel. Toeristen hebben nood aan effectieve ondersteuning van hun besluitvorming omtrent de selectie van bezienswaardigheden en een route hiertussen. Het probleem bestaande uit het selecteren van een route tussen de verschillende bezienswaardigheden met als doel de voldoening van de toerist te maximaliseren, wordt geclassificeerd als een *tourist trip design* probleem. Startend vanuit het meest eenvoudige probleem, het *oriëntering probleem*, wordt getracht te evolueren naar een complexer probleem, het *multi-constraint team oriëntering problem with multiple time windows*. Deze evolutie zorgt ervoor dat met meer factoren rekening gehouden wordt zoals het beschikbare budget van de toerist of verschillende openingsuren van bezienswaardigheden op verschillende dagen.

Verscheidene auteurs hebben gepoogd een oplossingsmethode te ontwikkelen voor de verschillende gradaties van *tourist trip design* problemen. In de literatuurstudie worden zowel exacte algoritmen als heuristieken aangehaald die dienen als oplossingsmethoden voor *tourist trip design* problemen. Vertrekkend vanuit deze beschikbare algoritmen, wordt in de praktijkstudie een *tabu search* algoritme beschreven om een specifiek probleem op te lossen. Een oplossing wordt voorgesteld voor een toerist die de stad Hasselt gedurende twee dagen bezoekt. Zoals uit hoofdstuk 4 blijkt, is de veralgemeenbaarheid van het specifieke algoritme beperkt aangezien het aan veel willekeurigheden onderworpen is. Ondanks de willekeurigheden, vormt het algoritme een kader om gelijkaardige problemen op te lossen.

Het is aldus mogelijk een toeristische route te plannen met als doel de meest interessante locaties te bezoeken, rekening houdend met budget- en tijdsbeperkingen alsook met verscheidene criteria aangaande de aantrekkelijkheid van locaties.



## Lijst der geraadpleegde werken

Archetti, C., Feillet, D., Hertz, A., & Speranza, M.G. (2010). The undirected capacitated arc routing problem with profits. *Computers and Operations Research*, vol. 37, 1860-1869.

Baeza-Yates, R., & Ribeiro-Neto, R. (1999). *Modern information retrieval*. New York: Addison-Wesley.

*Basistarieven en kortingen*. (2013). Opgevraagd op 19 maart, 2013, via <http://kinopolis.be/nl/bioscopen/kinopolis-hasselt#information>.

Buhalis, D. (2003). *eTourism: information technology for strategic tourism management*. London: Prentice Hall.

Buhalis, D., & Law, R. (2008). Twenty years on and 10 years after the internet: the state of eTourism research. *Tourism Management*, vol. 29, 609-623.

Castro, M., Goos, P., Sörensen, K., & Vansteenwegen, P. (2013). A memetic algorithm for the travelling salesperson problem with hotel selection. *Computers and Operations Research*, <http://dx.doi.org/10.1016/j.cor.2013.01.006>.

Chao, L., Golden, B., & Wasil, E. (1996). Theory and methodology – a fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*, vol. 88, 475-489.

Chong, C.S., Low, M.Y.H., & Wong, L.-P. (2010). Bee colony optimization with local search for travelling salesman problem. *International Journal on Artificial Intelligence Tools*, vol. 19, 305-334.

Dunlop, M., McCallum, S., Morrison, S., Ptasiński, P., Risbey, C., & Stewart, F. (2004). Design and development of Taeneb city guide - from paper maps and guidebooks to electronic guides. *Proceedings of Enter 2004*, Caïro, Egypte, 2004.

Feillet, D., Dejax, P., & Gendreau, M. (2005). Travelling salesman problems with profits. *Transportation Science*, vol. 39, 188-205.

Fesenmainer, D.R., Ricci, F., Schaumlechner, E., Wober, K., & Zanella, C. (2003). Dietorecs: travel advisory for multiple decision styles. In *Information and communication Technologies in tourism*, 232-241. New York: Springer.

Fischetti, M., Salazar, J., & Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, vol. 10, 133-148.

Fomin, F.V., & Lingas, A. (2002). Approximation algorithms for time-dependent orienteering. *Information Processing Letters*, vol. 83, 57-62.

- Fung, R.Y.K., Tang, J., & Zhang, J. (2011). A scatter search for multi-depot vehicle routing problem with weight-related cost. *Asia-Pacific Journal of Operational Research*, vol. 28, 323-348.
- Garcia, A., Vansteenwegen, P., Souffriau, W., Arbelaitz, O., & Linaza, M. T. (2010). Solving multi constrained team orienteering problems to generate tourist routes. *Discrete Optimization*, under review.
- Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., & Tasoulas, Y. (2012). A survey on algorithmic approaches for solving tourist trip design problems. *eCompass*.
- Gendreau, M., Laporte, G., & Semet, F. (1998a). A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, vol. 106, 539-545.
- Gendreau, M., Laporte, G., & Semet, F. (1998b). A branch-and-cut algorithm for the undirected selective travelling salesman problem. *Networks*, vol. 32, 263-273.
- Golden, B., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics*, vol. 34, 307-318.
- Hannes, W. (2003). Intelligent system in travel and tourism. *Proceeding of the 18<sup>th</sup> international joint conference on artificial intelligence*, Acapulco, Mexico, 2003.
- Hillier, F.S., & Lieberman, G.J. (2010). *Introduction to Operations Research*. New York: McGraw-Hill.
- Jeong, M., Oh, H., & Gregoire, M. (2003). Conceptualizing Web site quality and its consequences in the lodging industry. *International Journal of Hospitality Management*, vol. 22, 161-175.
- Kalender. (2013). Opgevraagd op 28 maart, 2013, via <http://www.plopsa.be/plopsa-indoor-hasselt/nl/kalender>.
- Kramer, R., Modsching, M., & ten Hagen, K. (2006). A city guide agent creating and adapting individual sightseeing tours based on field trial results. *International Journal of Computational Intelligence Research*, vol. 2, 191-206.
- Labadi, N., Prins, C., & Reghioui, M. (2009). Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research*, vol. 47, 507-535.
- Laporte, G., & Martello, S. (1990). The selective travelling salesman problem. *Discrete Applied Mathematics*, vol. 26, 193-207.
- Leifer, A., & Rosenwein, M. (1994). Strong linear programming relaxations for the orienteering problem. *European Journal of Operational Research*, vol. 73, 517-523.

- Lin, S.-W., & Yu, V.F. (2012). A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, vol. 217, 94-107.
- Miller, C., Tucker, A., & Zemlin, R. (1960). Integer programming formulations and travelling salesman problems. *Journal of the ACM*, vol. 7, 326-329.
- Montemanni, R., & Gambardella, L. (2009). Ant colony system for team orienteering problems with time windows. *Computational Decision Science*, vol. 34.
- Morrison, A.M., Jing, S., O'Leary, J.T., & Lipping, A.C. (2001). Predicting usage of the Internet for travel bookings: An exploratory study. *Information Technology & Tourism*, vol. 4, 15-30.
- Muyldermans, L., Beullens, P., Cattrysse, D., & Van Oudheusden, D. (2005). Exploring variants of 2- and 3-opt for the general routing problem. *Operations Research*, vol. 53, 982-995.
- Ramesh, R., & Brown, K. (1991). An efficient four-phase heuristic for the generalized orienteering problem. *Computers and Operations Research*, vol. 18, 151-165.
- Ramesh, R., Yoon, Y., & Karwan, M. (1992). An optimal algorithm for the orienteering tour problem. *ORSA Journal on Computing*, vol. 4, 155-165.
- Ricci, F., & Wether, H. (2002). Case base querying for travel planning recommendation. *Information Technology & Tourism*, vol. 4, 215-226.
- Righini, G., & Salani, M. (2009). Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. *Computational Operations Research*, vol. 36, 1191-1203.
- Routebeschrijving*. (2013). Opgevraagd op 19 maart, 2013, via <http://maps.google.be/>.
- Schafer, J. B., Konstan, J. & Riedi, J. (1999). Recommender systems in e-commerce. In *Proceedings of the 1<sup>st</sup> acm Conference on Electronic Commerce 1999*.
- Schilde, M., Doerner, K.F., Hartl, R.F., & Kiechle, G. (2009). Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence*, vol. 3, 179-201.
- Souffriau, W., & Vansteenwegen, P. (2010). Tourist trip planning functionalities: State-of-the-art and future. In F. Daniel & F.M. Facca (Eds.), *Lecture Notes in Computer Science* (Vol. 6385, pp. 474-485). Presented at the 10th International conference on Web Engineering (ICWE 2010), Berlin, Germany: Springer.
- Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., & Van Oudheusden, D. (2006). Multilevel metaheuristics for the orienteering problem. In *7th EU/Meeting on Adaptive, Self-Adaptive, and Multi-Level Metaheuristics*, University of Malaga, Spain, November 2006.



Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., & Van Oudheusden, D. (2012). The multiconstraint team orienteering problem with multiple time windows. *Transportation Science*.

Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G., & Van Oudheusden, D. (2008). A personalized tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence*, vol. 22, 964-985.

Sylejmani, K., Dorn, J., & Musliu, N. (2012). A tabu search approach for multi constrained team orienteering problem and its application in touristic trip planning. In *Proceedings of the 12<sup>th</sup> International Conference on Hybrid Intelligent Systems 2012*, 300-305.

Tarieven. (2013). Opgevraagd op 19 maart, 2013, via <http://www.hasselt.be/nl/content/3416/tarieven.html>.

Tarieven. (2013). Opgevraagd op 19 maart, 2013, via <http://www.plopsa.be/plopsa-indoor-hasselt/nl/tarieven>.

Toerisme. (2010). Opgevraagd op 21 april, 2012, via [http://www.belgium.be/nl/over\\_belgie/toerisme/](http://www.belgium.be/nl/over_belgie/toerisme/).

Toerisme Hasselt. (2013). *Bezienswaardig Hasselt & Zonhoven*.

Toerisme Hasselt. (2013). Opgevraagd op 18 februari, 2013, via <http://toerisme.hasselt.be/nl/>.

Toerisme in Europa ondersteunen. (2012). Opgevraagd op 21 april, 2012, via [http://ec.europa.eu/enterprise/sectors/tourism/index\\_nl.htm](http://ec.europa.eu/enterprise/sectors/tourism/index_nl.htm).

Toerisme Limburg. (2013). *Hasselt Ontdekkingsgids 2013*. Hasselt: Toerisme Limburg vzw.

Tricoire, F., Romauch, M, Doerner, K.F., & Hartl, R.F. (2010). Heuristics for the multi-period orienteering problem with multiple time windows. *Computers & Operations Research*, vol. 37, 351-367.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, vol. 35, 797-809.

Vansteenwegen, P., Souffriau, W., & Sörensen, K. (2011a). The travelling salesperson problem with hotel selection. *Journal of the Operational Research Society*, vol. 63, 207-217.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., & Van Oudheusden, D. (2009). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, vol. 36, 3281-3290.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., & Van Oudheusden, D. (2011b). The City Trip Planner: An expert system for tourists. *Expert Systems with Applications*, vol. 38, 6540-6546.

Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D. (2011c). The orienteering problem: A survey. *European Journal of Operational Research*, vol. 209, 1-10.

Vansteenwegen, P., & Van Oudheusden, D. (2007). The mobile tourist guide: An OR opportunity. *OR Insights*, vol. 20, 21-27.

Voudouris, C., & Tsang, E. (1999). Guided local search and its application to the travelling salesman problem. *European Journal of Operational Research*, vol. 113, 469-499.

Wang, X., Golden, B., & Wasil, E. (2008). Using a genetic algorithm to solve the generalized orienteering problem. In: Golden, B., Raghavan, S., & Wasil, E. (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*, pp. 263-274.

*Why Tourism?*. (2012). Opgevraagd op 21 april, 2012, via <http://unwto.org/en/content/why-tourism>.

Zhu, C., Hu, J.Q., Wang, F., Xu, Y., & Cao, R. (2010). On the tour planning problem. *Annals of Operations Research*, vol. 192, 67-86.

Zong, G., Tseng, C.-L., & Park, Y. (2005). Harmony search for generalized orienteering problem: Best touring in China. *Advances in Natural Computation*, 741-750.



## **Bijlagen**

Bijlage 1: Stadsplan Hasselt (Toerisme Hasselt, 2013)



Bijlage 1: Stadsplan Hasselt (Toerisme Hasselt, 2013) (1)

**Toerisme Hasselt – UITpunt**

Maastrichterstraat 59  
BE-3500 Hasselt  
T +32 11 23 95 40  
F +32 11 22 50 23  
toerisme@hasselt.be  
www.hasselt.eu

---

**Opgelet parkeren!**  
*Op en binnen de kleine ring*  
Betalend parkeren van ma. tot en met zat. van 9 tot 19u

*Tussen kleine en grote ring*  
Volledig blauwe zone: parkeerschijf verplicht

Autosnelweg	Autoroute	Autobahn	Highway
Verkeerslichten	Faux de signalisation	Straßensignal	Traffic lights
Waterwagion	Cours d'eau	Wasserläufe	Watercourses
Spoonwagen	Chemins de fer	Eisenbahn	Railways
Verplichte rijrichting	Sans obligation	Obligatorische Fahrtricht.	One-way streets
Verkeersvrije straten	Rues piétonnières	Verkehrsfreie Straßen	Traffic-free streets
Stoegjes	Passage	Passage	Passage
Kerk	Eglise	Kirche	Church
Ziekenhuis	Hôpital	Krankenhaus	Hospital
WC	WC	WC	WC
Parking	Parking	Parkplatz	Parking
Overdekte parking	Parking couvert	Überdachter Parkplatz	Covered parking
Mobilhome parking	Parking mobilhome	Mobilhome Parkplatz	Mobilhome parking
Postkantoor	Bureau de poste	Postamt	Post-office
Informatiekantoor	Bureau d'information	Information	Information office
Monument	Monument	Denkmal	Monument
Watertoren	Château d'eau	Wasserturm	Water tower
Watersport	Sports nautiques	Wassersport	Aquatic sports

PARK  
&  
RIDE

<p><b>GPS</b> Sporthof Aalberg Herkenrodesingel 33 3500 Hasselt</p>	<p><b>Gratis parking</b> tegenover Carrefour en gratis stadsbus 'H8' naar het stadscentrum 'Grote Markt'</p>	<p><b>Parking gratuit</b> en face du Carrefour et bus gratuit 'H8' en direction du centre 'Grote Markt'</p>	<p><b>Free parking</b> opposite Carrefour and free bus 'H8' to the city-centre 'Grote Markt'</p>	<p><b>Gratis Parkplaats</b> gegenüber Carrefour und gratis Bus 'H8' zum Stadtzentrum 'Grote Markt'</p>
---	--	---	--	--

<p><b>Musea – Musées</b> <b>Museen – Museums</b></p> <ol style="list-style-type: none"> <li>1 Literair Museum – Musée de la littérature Literaturmuseum – Literary Museum</li> <li>2 Museum Pater Valentinus Paquay 'Heilig Paterik' Musée du Père Valentin Paquay</li> <li>3 Nationaal Jenevermuseum – Musée du genièvre Genevermuseum – Genever Museum</li> <li>4 Beiaardtonen – Musée du carillon Glockenspielmuseum – Carillon Museum</li> <li>5 Modemuseum Hasselt – Musée de la mode Modemuseum – Fashion Museum</li> <li>6 Stadsmuseum Het Stadsmus Musée communal Städtisches Museum – Municipal Museum</li> </ol>	<p><b>Bezienswaardigheden – Curiosités</b> <b>Sehenswürdigkeiten – Sights</b></p> <ol style="list-style-type: none"> <li>1 Begijnhof – Béguinage Begijnenhof – Beguinage</li> <li>2 Abdijste Herkenrode – Abbaye Abtei – Abbey</li> <li>3 Japane Tuin – Jardin japonais Japanischer Garten – Japanese garden</li> <li>4 Sint-Rochuskerk (Paterkerk) – Eglise Kirche – Church</li> <li>5 Oud-Kerkhof – Le vieux cimetière Alter Friedhof – Old churchyard</li> <li>6 Prinsenhof – Palais princier Prinzenhof – Prince's residence</li> <li>7 Sint-Quintinskathedraal – Cathédrale Kathedrale – Cathedral</li> <li>8 Stadhuis – L'Hôtel de ville Rathaus – Town hall</li> <li>9 Virgo Jessebasiliek – Basilique Basilika – Basilica</li> <li>10 Park 'Nature &amp; Culture' – Parc 'Nature &amp; Culture' Park 'Nature &amp; Culture'</li> </ol>	<p><b>Nuttige adressen – Adresses utiles</b> <b>Nützliche Adressen – Useful addresses</b></p> <ol style="list-style-type: none"> <li>1 Toerisme Hasselt Office du Tourisme de Hasselt Tourist Information Hasselt</li> <li>2 Huis – Maison – Haus – House de Conwarem Smaakzaal – Le Salon du Goût Geschmackssaal – Salon of Taste</li> <li>3 Politie – Police Politie – Police station</li> <li>4 Postkantoor – Poste Postamt – Post office</li> <li>5 Provinciale Bibliotheek – Bibliothèque Bibliothek – Library</li> <li>6 Provinciehuis – Maison provinciale Provinzhuis – County hall</li> <li>7 Train- en Busstation en Fietspunt Gare et station d'autobus – Zug- und Busbahnhof Railway station and bus terminal</li> <li>8 Jessa Ziekenhuis (Campus Virgo) (Jessa) Hôpital – Krankenhaus – Hospital</li> <li>9 Jessa Ziekenhuis (Campus Salvator) Hôpital – Krankenhaus – Hospital</li> <li>10 Stadsarchief – Archives municipales Stadtarchiv – Municipal archives</li> <li>11 Web (Incampus, A. Rodenbachstraat 20)</li> </ol>	<p><b>Hotels – Hôtels</b></p> <ol style="list-style-type: none"> <li>1 Hotel The Century** Leopoldplein 1</li> <li>2 Hotel Top Class*** Diestertorenweg 33, Kermt (richting Herk-de-Stad)</li> <li>3 Hotel Radisson BLU**** Torenplein 8</li> <li>4 Hotel Hassotel*** Sint-Jacobsstraat 2-10</li> <li>5 Hotel Holiday Inn**** Kattegatstraat 1</li> <li>6 Hotel Ibis** Thonissenlaan 32</li> <li>7 Hotel Présence*** Genkersteerweg 332</li> <li>8 Hotel Pax* Grote Markt 16</li> <li>9 Hotel Portmans** Mindelbroederstraat 12-14</li> <li>10 Hotel Express by Holiday Inn*** Thonissenlaan 32</li> <li>11 Hotel De Groene Hendriek*** Zuivelmarkt 25</li> </ol>	<p><b>Ontspanning – Loisirs</b> <b>Entspannung – Recreation</b></p> <ol style="list-style-type: none"> <li>1 Cultuurcentrum – Le centre culturel Kulturzentrum – Cultural centre</li> <li>2 Megadisco Versuz – Discothèque Diskothek – Discotheque</li> <li>3 Domain Kiewit – Le domaine naturel de Kiewit Naturholungsgebiet – Nature reserve</li> <li>4 Grootlandhallen – Ethias Arena – Flopax Indoor Halls des foires – Veranstaltungshallen – Halls for events</li> <li>5 Kapermolenpark – Parc – Park</li> <li>6 Kinopolis – Cinéma Kino – Cinema</li> <li>7 Zwembaden – Piscines Schwimmbäder – Swimming pools</li> <li>8 Ijschaatsbaan – Patinoire Schlittschuhbahn – Ice-rink</li> <li>9 Golfkernin – Terrain de golf Golfplatz – Golf-course</li> <li>10 Rederij 'Limburgia' – Armement fluvial Reederei – Shipping company</li> <li>11 Kunstencentrum 'BELGIE' – Centre artistique Kunstzentrum – Arts centre</li> <li>12 Muziekcentrum 'Muziekdoorn' Centre musical – Musikzentrum – Music centre</li> <li>13 Z33 - huis voor actuele kunst - Centre artistique Kunstzentrum – Arts centre</li> </ol>
--	--	--	---	---

**Fietsroutenetwerk Limburg**

Knooppunt 96 aan de kanaalkom

Knooppunt 97 aan de Japanse Tuin

Verbinding naar knooppunt 142

Verbinding naar knooppunt 143

TB2013

Bijlage 1: Stadsplan Hasselt (Toerisme Hasselt, 2013) (2)

## Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

**Het plannen van routes voor toeristen aan de hand van meerdere criteria**

Richting: **master in de toegepaste economische wetenschappen:  
handelsingenieur-accountancy en financiering**

Jaar: **2013**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

**Umans, Ine**

Datum: **2/06/2013**