

RITTENPLANNING VOOR HET LEERLIN- GENVERVOER IN HET BUITENGEWOON ONDERWIJS

Indra COPPENS

Promotor: Prof. Dr. Ir. Greet Vanden Berghe

Co-promotor: Ing. Niels De Walsche

Masterproef ingediend tot
het behalen van de graad van
master of Science in
de industriële wetenschappen:
Elektronica-ICT Software
engineering

Academiejaar 2025 - 2026

©Copyright KU Leuven

Deze masterproef is een examendocument dat niet werd gecorrigeerd voor eventuele vastgestelde fouten.

Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, kan u zich richten tot KU Leuven Technologicampus Gent, Gebroeders De Smetstraat 1, B-9000 Gent, +32 92 65 86 10 of via e-mail iiw.gent@kuleuven.be.

Voorafgaande schriftelijke toestemming van de promotor(en) is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Voorwoord

Met het afronden van deze masterproef en daarmee ook mijn masteropleiding, kijk ik terug op een intensieve en bijzonder leerrijke periode. Het realiseren van deze masterproef was een uitdagend traject dat ik niet alleen heb doorlopen. Ik wil dan ook graag enkele mensen bedanken die mij tijdens dit proces hebben gesteund.

Allereerst wil ik mijn promotor Greet Vanden Berghe bedanken voor de kans om deze masterproef uit te werken onder haar deskundige begeleiding. Ook mijn co-promotor Niels De Walsche wil ik bedanken voor zijn tijd en waardevolle feedback. Hun begeleiding en inzichten waren van grote waarde tijdens het realiseren van dit werk.

Daarnaast ben ik mijn ouders en familie dankbaar voor hun onvoorwaardelijke steun en aanmoediging gedurende mijn studie. Ook mijn vrienden wil ik bedanken voor hun geduld en luisterend oor, telkens wanneer ik enthousiast vertelde over mijn onderzoek. Jullie interesse en steun hebben voor mij een groot verschil gemaakt.

Tot slot wil ik iedereen bedanken die op welke manier dan ook heeft bijgedragen aan dit werk en mijn studie. Zonder jullie steun was dit niet mogelijk geweest.



*Copyright informatie:
studiewerk van een student in het kader van de
academische opleiding en examenbeoordeling.
Na deze beoordeling vond geen correctie plaats
van het studiewerk.*

Samenvatting

Leerlingen in het buitengewoon onderwijs hebben recht op gratis vervoer van en naar school, op voorwaarde dat zij zich inschrijven bij de dichtstbijzijnde school die hun opleiding aanbiedt. Het organiseren van dit vervoer vereist een planning die rekening houdt met uiteenlopende beperkingen en voorkeuren, zoals voertuigcapaciteit, reistijd en individuele noden van de leerlingen. Momenteel wordt deze planning volledig manueel opgesteld. Door het jaarlijks stijgende aantal leerlingen neemt de complexiteit van deze taak sterk toe. Om te voorkomen dat leerlingen onnodig lang onderweg zijn, wordt het ontwikkelen van een efficiënte planningsmethode steeds belangrijker.

Het huidige systeem voor leerlingenvervoer in het buitengewoon onderwijs kent, ondanks harde inspanningen van de manuele planners, enkele tekortkomingen. Leerlingen zitten vaak te lang op de bus, wat regelmatig aanleiding geeft tot maatschappelijke kritiek en negatieve berichtgeving. Tot op heden bestaat er geen algoritmische tool om deze planning automatisch te genereren. Hierdoor is het bovendien niet haalbaar om het effect van beleidskeuzes, zoals het toevoegen van extra bussen of het uitbreiden van de periode waarin leerlingen op school afgezet kunnen worden, op een objectieve manier te bestuderen. Dit maakt het moeilijk om te bepalen welke maatregelen het grootste effect kunnen hebben op de kwaliteit van het vervoer.

Deze masterproef onderzoekt hoe het probleem van lange reistijden kan worden aangepakt door middel van een geautomatiseerd planningsalgoritme. Het doel is het ontwikkelen van een planningsalgoritme dat ritten genereert voor leerlingenvervoer in het buitengewoon onderwijs. De focus ligt op het minimaliseren van de langste individuele rittijd (minimax-objectief), zodat het comfort van de leerlingen centraal staat. Hiervoor ontwikkelen we bovendien een dataset die de praktijk zo goed mogelijk benadert, zodat we het algoritme kunnen testen op realistische scenario's. Vervolgens gebruiken we deze dataset om het effect van verschillende parameters en beleidsopties op de kwaliteit van de planning te analyseren.

Het resultaat is een algoritme dat, gegeven een verzameling leerlingen met hun bijhorende scholen en een vloot bussen, een rittenplanning genereert voor het vervoer van leerlingen naar hun school. Het algoritme minimaliseert de langste individuele rittijd en houdt reke-

ning met de opgelegde beperkingen, zoals voertuigcapaciteit en tijdsvensters waarbinnen leerlingen kunnen toekomen op school. Naast het algoritme biedt deze masterproef een wiskundige formulering van het probleem en een realistische dataset, die beiden kunnen dienen als basis voor toekomstig onderzoek.

Trefwoorden: rittenplanning, leerlingenvervoer, buitengewoon onderwijs, tijdsvensters, transferpunten, schoolbus, *Vehicle Routing Problem (VRP)*

Summary

Students in special education are entitled to free transportation to and from school as long as they enroll in the nearest school offering their program. Organizing this transportation requires planning that accounts for various constraints and preferences, such as vehicle capacity, travel time, and individual student needs. Currently, this planning is done entirely manually. As the number of students increases each year, the complexity of this task grows significantly. To prevent students from spending unnecessary long time on the road, developing an efficient planning method is becoming increasingly important.

The current system for student transportation in special education has, despite the dedicated efforts of manual planners, several shortcomings. Students often spend too much time on the bus, which frequently leads to public criticism and negative media coverage. To date, no algorithmic tool exists to automatically generate these routing plans. As a result, it is not feasible to objectively study the impact of policy decisions, such as adding extra buses or extending the timeframe in which students can be dropped off. This makes it difficult to determine which measures would have the greatest effect on transportation quality.

This thesis investigates how the problem of long travel times can be addressed through an automated planning algorithm. The goal is to develop a planning algorithm that generates routes for the transportation of students with special needs to their school. The focus is minimizing the longest individual ride time (a minimax objective), thereby prioritizing the comfort of the students. Additionally, a dataset is created to closely reflect real-world situations, allowing the algorithm to be tested in realistic scenarios. This dataset is then used to analyse the impact of different parameters and policy decisions on planning efficiency.

The result is an algorithm that, given a set of students with their respective schools and a fleet of buses, generates a routing schedule for transporting students to their school. The algorithm minimizes the longest individual ride time while respecting constraints such as vehicle capacity and time windows for school arrival. In addition to the algorithm, this thesis provides a mathematical formulation of the problem and a realistic dataset, both of which can serve as a foundation for future research.

Keywords: route scheduling, student transportation, special education, time windows, transfer points, school bus, Vehicle Routing Problem (VRP)

Inhoudsopgave

Samenvatting	vi
Summary	viii
1 Inleiding	2
2 Probleembeschrijving	4
2.1 Basisprobleem	4
2.2 Mogelijke uitbreiding: transferpunten	4
2.3 Veronderstellingen voor het algoritme	5
3 Literatuurstudie	7
3.1 School Bus Routing Problem (SBRP) met in het bijzonder School Bus Routing Problem (SBRP) voor leerlingen met speciale noden	7
3.2 Pickup-and-Delivery Problem with Time Windows (PDPTW) en Dial-a-Ride Problem (DARP)	9
3.3 Conclusie van de literatuurstudie	10
4 Wiskundige formulering van de rittenplanning	11
4.1 Inleiding	11
4.2 Probleemdefinitie en notaties	11
4.2.1 Definitie	11
4.2.2 Netwerkconstructie	13
4.2.3 Samenvatting van gebruikte notaties	14
4.2.4 Voorbeeldoplossing	14
4.3 Wiskundige formulering van het probleem	15
4.3.1 Beslissingsvariabelen	15
4.3.2 Formulering	15
4.3.3 Andere geldige beperkingen (<i>valid inequalities</i>)	18

4.3.4	Alternatieve formulering met flow-variabelen	18
4.4	Mogelijke uitbreiding: transferpunten	18
4.4.1	Definitie transferpunten en veronderstellingen	19
4.4.2	Voorbeeldoplossing	19
4.4.3	Extra (hulp-)beslissingsvariabelen voor transferpunten	21
4.4.4	Formulering	21
4.5	Parameters	25
4.6	Resultaten wiskundige formulering	26
4.6.1	Resultaten voor het basisprobleem	27
4.6.2	Resultaten uitbreiding transferpunten	27
4.6.3	Conclusie resultaten wiskundige formulering	28
5	Heuristische methodologie	30
5.1	Inleiding	30
5.2	Bestaande methodes als startpunt voor de heuristiek	31
5.2.1	Slack Induction by String Removals (SISRs) als basisstructuur . . .	31
5.2.2	Tijdvenstertechniek	31
5.3	Probleemmodellering en oplossingsvoorstelling	32
5.4	Service-Maximization Minimax with Fixed-Fleet (SMM-FF)	33
5.4.1	Local search metaheuristiek	33
5.4.2	Initiële oplossing	36
5.4.3	Ruin-methode	36
5.4.4	Recreate-methode	37
5.4.5	Compress Schedule methode	38
5.5	Varianten op SMM-FF	39
5.5.1	Service-Maximization Minimax with Temporary-Fleet-Expansion (SMM-TFE)	39
5.5.2	Full-Service Minimax with Fleet-Expansion (FSM-FE)	40
5.6	Parameters	41
6	Generatie van realistische dataset	43
6.1	Inleiding	43
6.2	Scholen	44
6.3	Leerlingen	46
6.3.1	Leerlingen en hun locaties	46
6.3.2	Servicetijden	47

6.4	Parkings	48
6.5	Tijdsmatrix	48
6.6	Bussen	49
6.7	Resultaat en deelverzamelingen	50
7	Experimenten en resultaten	53
7.1	Inleiding	53
7.2	Technische experimenten en resultaten	54
7.3	Toepassingsgerichte experimenten	58
8	Conclusie	65
8.1	Besluit	65
8.2	Toekomstige onderzoeksmogelijkheden	66
A	Verschillende definities voor de rittijd	74
B	Alternatieve formulering met flowvariabelen	76
B.1	Beslissingsvariabelen	76
B.2	Formulering	76
C	Pseudocode voor de heuristieken	78
D	Overige resultaten voor de arrondissementen met de hoofdheuristiek	84
E	Bronnen voor de tijdsvensters van de opgezochte scholen	90
F	Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel	95
G	Poster	102
H	GenAI code of conduct for students	104

Acroniemen

BCB Binaire Capaciteit van Bogen. 13, 17, 18, 22, 77

CVRP Capacitated Vehicle Routing Problem. 31

DARP Dial-a-Ride Problem. ix, 7, 9

FSM-FE Full-Service Minimax with Fleet-Expansion. x, 30, 40, 41, 54, 55, 83

HPT Handicapped Person Transportation Problem. 9

OVRP Open Vehicle Routing Problem. 5, 12, 31

PDP Pickup-and-Delivery Problem. 8, 31

PDPTW Pickup-and-Delivery Problem with Time Windows. ix, 7, 9, 68

PTP Prisoner Transportation Problem. 9

SA Simulated Annealing. 33

SBRP School Bus Routing Problem. ix, 7, 8, 10

SISRs Slack Induction by String Removals. x, 31, 33, 34, 35, 36, 79

SMM-FF Service-Maximization Minimax with Fixed-Fleet. x, 30, 33, 39, 40, 41, 53, 54, 55, 56, 58, 59, 60, 78, 84, 85, 86, 87, 88, 89

SMM-TFE Service-Maximization Minimax with Temporary-Fleet-Expansion. x, 30, 39, 40, 54, 82

VRP Vehicle Routing Problem. 9

Hoofdstuk 1

Inleiding

Het organiseren van leerlingenvervoer in het buitengewoon onderwijs is een complexe logistieke uitdaging. Leerlingen hebben recht op gratis vervoer van en naar school, op voorwaarde dat zij zich inschrijven in de dichtstbijzijnde school die hun opleiding aanbiedt. De planning moet rekening houden met diverse beperkingen, zoals buscapaciteit, tijdsvensters en individuele noden. Bovendien groeit het aantal leerlingen in het buitengewoon onderwijs elk jaar, waardoor de planning steeds complexer wordt.

Ondanks de harde inspanningen van de planners zitten sommige leerlingen te lang op de bus. Dit heeft niet alleen een negatieve impact op hun comfort, maar leidt ook regelmatig tot maatschappelijke kritiek. Bovendien biedt de huidige werkwijze geen mogelijkheid om de impact van beleidskeuzes, zoals het toevoegen van extra bussen of het verruimen van tijdsvensters, te evalueren.

Het doel van deze masterproef is een algoritme te ontwikkelen om het leerlingenvervoer in het buitengewoon onderwijs te plannen. Het hoofddoel is het minimaliseren van de langste individuele rittijd, zodat het comfort van de leerlingen centraal staat. Daarnaast beoogt deze masterproef de mogelijkheid om het effect van beleidskeuzes te simuleren, zoals het toevoegen van extra bussen of het verruimen van tijdsvensters. Toekomstig onderzoek ondersteunen we met een precieze probleembeschrijving, een wiskundige formulering van het probleem en een realistische dataset.

De focus van dit onderzoek ligt op twee centrale onderzoeksvragen. Ten eerste onderzoeken we of het mogelijk is om een geautomatiseerd planningsalgoritme te ontwikkelen voor het opstellen van de ritten in het buitengewoon onderwijs. Ten tweede bekijken we in welke mate deze aanpak kan bijdragen tot het verlagen van de langste rittijd van leerlingen.

De verdere structuur van deze masterproef ziet er als volgt uit. Hoofdstuk 2 beschrijft het probleem en de veronderstellingen die we hanteren. Vervolgens bespreekt Hoofdstuk 3 de

relevante literatuur. In Hoofdstuk 4 stellen we een exacte formulering voor, gevolgd door een heuristische aanpak in Hoofdstuk 5. Daarna behandelt Hoofdstuk 6 de opbouw van een realistische dataset. Hoofdstuk 7 presenteert de experimenten en resultaten. Tot slot bespreekt Hoofdstuk 8 conclusies en aanbevelingen voor toekomstig werk.

Hoofdstuk 2

Probleembeschrijving

2.1 Basisprobleem

Voor het vervoer in het buitengewoon onderwijs is een planning vereist waarbij we leerlingen thuis ophalen en op school afzetten. Deze planning moet rekening houden met diverse beperkingen en voorkeuren. Het objectief is het minimaliseren van de langste individuele rittijd over alle leerlingen, ook wel het minimax-objectief genoemd. Daarbij is het essentieel dat we alle vervoersaanvragen uitvoeren, zodat iedere leerling op de juiste school aankomt.

De vloot is heterogeen en elke bus heeft een beperkt aantal zitplaatsen, wat overeenkomt met de capaciteit van de bus. Zowel het aantal bussen als hun individuele capaciteit liggen contractueel vast en blijven bijgevolg onveranderd.

Daarnaast heeft elke school een tijdsvenster waarbinnen leerlingen moeten aankomen. Het startpunt van dit interval komt overeen met het openen van de schoolpoort, terwijl het einde samenvalt met het begin van de lessen. Binnen dit tijdsvenster voorzien scholen toezicht op de speelplaats, zodat leerlingen veilig op school kunnen verblijven tot de lesdag begint.

Het resultaat dat we willen bekomen is een planning waarin we alle leerlingen van hun thuisadres naar hun respectievelijke schoollocatie brengen, rekening houdend met de beperkingen en het objectief.

2.2 Mogelijke uitbreiding: transferpunten

Het buitengewoon onderwijsvervoer maakt momenteel gebruik van transfers op parkings. Meerdere bussen kunnen naar een parking rijden en er leerlingen uitwisselen. Dergelijke transfers vereisen echter temporele en ruimtelijke synchronisatie, waarbij alle betrokken bussen gelijktijdig aanwezig moeten zijn op deze locatie gedurende de volledige transfer.

Na afloop van de transfer kunnen bussen hun onafhankelijke rit verderzetten. Het is bijgevolg interessant om te onderzoeken wat de voor- en nadelen zijn van transfers voor het plannen van de ritten.

2.3 Veronderstellingen voor het algoritme

We modelleren het probleem als een Open Vehicle Routing Problem (OVRP), waarbij bussen geen vast start- of eindpunt hebben. De rit start bij het instappen van de eerste leerling en eindigt wanneer de laatste leerling uitstapt op zijn school. Daarbij kan een bus meerdere eindbestemmingen hebben. Een bus mag leerlingen van verschillende scholen tegelijk vervoeren en hen op de juiste locaties afzetten. Bovendien mag een bus, na het afzetten van enkele leerlingen op een school, opnieuw leerlingen ophalen alvorens door te rijden naar de volgende bestemming.

De tijd die een leerling nodig heeft om in of uit te stappen geven we mee als inputparameter. Deze tijd kan variëren tussen de verschillende leerlingen. Zo duurt het bijvoorbeeld langer om in te stappen wanneer een leerling een fysieke beperking heeft. In het algoritme gaan we ervan uit dat de instaptijd gelijk is aan de uitstaptijd; deze tijd noemen we de servicetijd van een leerling. Daarnaast verloopt het in- en uitstappen sequentieel. Er kan maximaal één leerling tegelijk in- en uitstappen. Indien transfers toegevoegd worden, kunnen we de overstaptijd van een leerling definiëren als de som van de in- en uitstaptijd, wat overeenkomt met tweemaal de servicetijd. We veronderstellen bovendien dat elke leerling maximaal één transfer mag hebben op zijn rit.

Vervolgens maakt het algoritme enkele veronderstellingen over de eigenschappen van het netwerk. Een eerste eigenschap is dat de driehoeksongelijkheid geldt. Het is altijd sneller of even snel om rechtstreeks naar een locatie te rijden dan via een andere locatie. Een tweede eigenschap is dat het aantal eindbestemmingen beperkt is, terwijl de ophaallocaties verspreid liggen. Elke ophaallocatie correspondeert daarbij met één leerling.

Daarnaast beschouwen we chauffeurs, begeleiders en hun bus als één operationele eenheid. We nemen aan dat elke bus beschikt over een vaste chauffeur en begeleider, waardoor een afzonderlijke toewijzing van personeel door het algoritme niet nodig is. De capaciteit van een bus verwijst uitsluitend naar het aantal leerlingen dat de bus gelijktijdig kan vervoeren; begeleiders worden hierin niet meegerekend.

Om het probleem beheersbaar te houden, maken we enkele vereenvoudigingen. Zo houden we geen rekening met de maximale rijtijd van bussen, noch met tankbeurten of het balanceren van de werkdruk van het personeel. Het netwerk modelleren we als een verzameling knooppunten die locaties voorstellen en bogen die de reistijd tussen twee knooppunten representeren. De effectieve rit die binnen deze reistijd omvat zit, is verondersteld afkomstig

te zijn van een extern kaart- of GPS-systeem en maakt geen deel uit van het algoritme. Bovendien houden we geen rekening met verkeerssituaties of de aanrijrichting van bussen. Hoewel veel scholen naar verwachting slechts één of twee busplaatsen hebben, leggen we geen beperking op aan het aantal bussen dat gelijktijdig leerlingen kan afzetten op een school. Hetzelfde geldt voor transferpunten. De capaciteit van een transferpunt bepaalt hoeveel bussen gelijktijdig aanwezig kunnen zijn op deze locatie om een transfer uit te voeren, maar nemen we niet mee in het algoritme.

Tot slot focust het algoritme uitsluitend op de ochtendplanning, waarbij leerlingen van hun thuisadres naar school worden gebracht. De avondplanning is niet symmetrisch, maar beschouwen we niet expliciet in dit onderzoek. Deze veronderstellingen lijken mogelijk heel beperkend. We verkiezen echter eerst een basismodel op te stellen waarna veralgemeningen mogelijk zijn.

Hoofdstuk 3

Literatuurstudie

Deze sectie bespreekt enkele voorbeelden uit de literatuur die nauw aansluiten bij het rittenplanningsprobleem in het buitengewoon onderwijs. Kenmerken van het rittenplanningsprobleem zijn een *many-to-several*-structuur, *door-to-door*-service, heterogene voertuigen en tijdsvensters. De verwante problemen die we behandelen zijn het School Bus Routing Problem (SBRP) (incl. varianten voor leerlingen met speciale noden), Pickup-and-Delivery Problem with Time Windows (PDPTW) en Dial-a-Ride Problem (DARP).

3.1 School Bus Routing Problem (SBRP) met in het bijzonder SBRP voor leerlingen met speciale noden

SBRP

Newton en Thomas (1969) introduceerden het klassieke SBRP, dat doorgaans uit twee fasen bestaat. In de eerste fase is het doel geschikte haltes te selecteren en leerlingen aan deze haltes toe te wijzen, in de veronderstelling dat de leerlingen naar hun toegewezen halte wandelen. De tweede fase richt zich op het genereren van busritten langs deze haltes. Een uitgebreid overzicht van het SBRP en zijn varianten is gegeven door Park en Kim (2010), terwijl andere relevante bijdragen onder meer te vinden zijn in de publicaties van Kinable et al. (2014), en Riera-Ledesma en Salazar-González (2012).

Het klassieke SBRP streeft meestal naar het minimaliseren van de totale kosten en het beperken van het aantal homogene voertuigen. Bovendien gaat het doorgaans uit van één enkele school, waarbij alle voertuigen starten en eindigen op deze locatie (Park & Kim, 2010). Dit staat in contrast met ons probleem, waarin meerdere scholen betrokken zijn, ritten op willekeurige locaties starten en eindigen, en de primaire doelstelling het beperken van de maximale individuele rittijd is. Daarnaast willen wij leerlingen thuis ophalen, waardoor haltekeuze en wandelassumpties uit het klassieke SBRP niet van toepassing zijn. Bovendien is onze vloot heterogeen en contractueel vastgelegd.

Lima et al. (2017) behandelen een variant met meerdere scholen, *mixed loads* en een heterogene busvloot, wat al dichterbij ons probleem. Ook Park et al. (2012) beschouwen meerdere scholen en *mixed loads*. Zij dupliceren hierbij scholen zodat elke leerling aan een unieke school gekoppeld is, wat het mogelijk maakt het probleem als een Pickup-and-Delivery Problem (PDP) te formuleren. Dit is een idee dat wij willen overnemen om technieken uit de PDP-literatuur te benutten. De doelstelling van Park et al. (2012) blijft echter het minimaliseren van het aantal homogene bussen, wat afwijkt van onze focus op het beperken van de langste rittijd.

SBRP voor leerlingen met speciale noden

Russell et al. (1986) waren de eersten die het probleem voor leerlingen met speciale noden expliciet beschreven. In tegenstelling tot het klassieke SBRP, waarbij leerlingen naar een halte wandelen om daar in te stappen, gaat het *SBRP voor leerlingen met speciale noden* ervan uit dat elke leerling thuis in een bus stapt. Daarnaast vereist deze doelgroep vaak specifieke ondersteuning (bijvoorbeeld begeleiders) en materiaaleisen zoals rolstoelplaatsen.

Het werk van Caceres et al. (2019) ligt het dichtst bij onze context. Zij modelleren het SBRP voor leerlingen met speciale noden met een heterogene vloot, tijdsvensters en een strikte maximale rittijd per leerling. Er zijn echter enkele belangrijke verschillen; hun netwerk bevat een depot waar alle bussen starten en eindigen, en elke bus mag een schoollocatie hoogstens één keer bezoeken. Hun primaire doelstelling is het minimaliseren van het aantal ingezette bussen, gevolgd door het minimaliseren van de totale afstand. Ze stellen een *column generation* algoritme voor die per voertuigconfiguratie een rit genereert in een subprobleem en die ritten combineert in een masterprobleem. Deze aanpak is minder geschikt voor onze setting omdat wij geen vrijheid hebben om de vlootconfiguratie te kiezen. De dataset die zij gebruiken, afkomstig van Park et al. (2012), is gebaseerd op het klassieke SBRP en veronderstelt dat leerlingen reeds op haltes aanwezig zijn. Hoewel deze dataset aanvankelijk bruikbaar leek, bevat ze geen busvloot en vereist ze dat groepen leerlingen van dezelfde halte samen blijven. Bovendien streven Caceres et al. (2019) naar het minimaliseren van het aantal voertuigen, waardoor vergelijking met onze aanpak niet mogelijk is.

Verder combineren Ansari et al. (2021) en Kamali et al. (2013) de *student-to-school assignment* met het SBRP voor leerlingen met speciale noden in één algoritme. Deze benadering is relevant wanneer schooltoewijzing een beslissingsvariabele is, maar in onze casus is deze toewijzing een gegeven.

3.2 Pickup-and-Delivery Problem with Time Windows (PDPTW) en Dial-a-Ride Problem (DARP)

Het Pickup-and-Delivery Problem with Time Windows (PDPTW) is een variant van het Vehicle Routing Problem (VRP) waarbij het ophalen en afzetten van goederen of personen binnen vooraf gedefinieerde tijdsvensters moet plaatsvinden. Een relevante uitbreiding is het Handicapped Person Transportation Problem (HPT), dat zich richt op het verbeteren van de servicekwaliteit door wachttijden en rittijden te beperken (Rekiek et al., 2006). De doelstelling is om zoveel mogelijk klanten te bedienen met een minimaal aantal voertuigen.

Het Dial-a-Ride Problem (DARP) vormt een verdere generalisatie van het PDPTW en focust specifiek op het vervoer van personen (Cordeau & Laporte, 2007). Het kent veel toepassingen in het vervoer van ouderen of personen met een beperking. Klassieke varianten veronderstellen een homogene vloot en één enkel depot waar alle voertuigen starten en eindigen, met als doel het minimaliseren van de totale kosten en het maximaliseren van het aantal uitgevoerde vervoersaanvragen. Cordeau en Laporte (2007) geven een uitgebreid overzicht van DARP.

Uitbreidingen zoals het *Heterogeneous DARP* van Parragh et al. (2012) introduceren heterogene voertuigen, compatibiliteitsbeperkingen en chauffeurgerelateerde beperkingen. In deze context kunnen patiënten specifieke vervoersvormen aanvragen, zoals zittend, op een brancard of met een rolstoel.

Daarnaast introduceert Tellez et al. (2022) het *Time-Consistent DARP*, waarbij ze een planning over meerdere dagen opstellen. Deze variant is echter minder relevant voor onze context, aangezien wij ons richten op één enkele planning die elke schooldag dezelfde is.

Een andere interessante uitbreiding is het *Prisoner Transportation Problem (PTP)*, geïntroduceerd door Christiaens et al. (2020). Dit is een DARP-variant met conflictbeperkingen en voertuigen met meerdere compartimenten, gericht op het transport van gevangenen tussen locaties zoals rechtbanken en ziekenhuizen. Hoewel weinig beperkingen overeenkomen met ons probleem, biedt deze studie een elegante wiskundige formulering die als inspiratie kan dienen voor onze formulering. Bovendien stelt deze studie een heuristiek voor gebaseerd op een *ruin-and-recreate*-aanpak en houdt een maximale rittijd bij. Ferone et al. (2023) bouwen verder op dit werk en onderzoeken verbeterde oplossingsmethoden voor het PTP.

3.3 Conclusie van de literatuurstudie

Hoewel veel literatuur gelijkaardige problemen behandelt, ligt de focus zelden op het minimaliseren van de langste rittijd. Het meest verwante probleem is het SBRP voor leerlingen met speciale noden, maar bestaande studies hanteren meestal een depotstructuur en andere doelstellingen, waardoor hun technieken niet direct toepasbaar zijn op het leerlingenvervoer dat deze thesis behandelt.

Hoofdstuk 4

Wiskundige formulering van de rittenplanning

4.1 Inleiding

Het rittenplanningsprobleem in het buitengewoon onderwijs is in de literatuur slechts beperkt onderzocht. Hoewel er varianten bestaan die bepaalde gelijkenissen vertonen, ontbreekt een duidelijk kader dat specifiek focust op het minimaliseren van de rittijd en het gebruik van een vaste heterogene busvloot.

Om deze leegte op te vullen, stellen we voor om het probleem eerst strikt af te bakenen in Sectie 4.2. Vervolgens onderzoeken we of een exacte methode, die een optimale oplossing berekent, het probleem kan oplossen. Hiervoor stellen we een wiskundige formulering op in Sectie 4.3, die we met een gekende solver oplossen. Daarnaast bekijken we in Sectie 4.4 de mogelijke impact van transferpunten en evalueren we of deze een meerwaarde kunnen bieden voor het vervolg van dit onderzoek. Tot slot geven we een overzicht van de gekozen parameters in Sectie 4.5 en presenteren we in Sectie 4.6 de resultaten van deze exacte aanpak.

4.2 Probleemdefinitie en notaties

Deze sectie definieert het beschouwde optimalisatieprobleem en introduceert de notaties die in dit wiskundig model zijn toegepast.

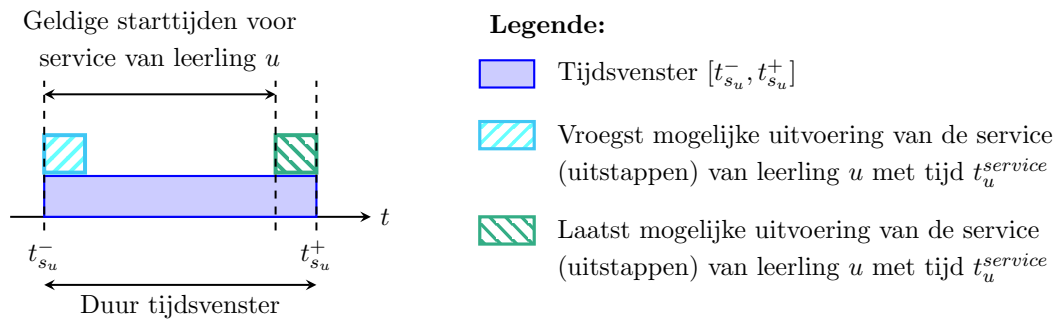
4.2.1 Definitie

Het probleem omvat een verzameling leerlingen U . Elke leerling $u \in U$ moeten we eerst ophalen aan zijn thuisadres a_u en daarna afzetten op zijn school s_u . Voor elke leerling definiëren we een unieke school, ook indien meerdere leerlingen naar dezelfde fysieke school gaan. We willen ritten plannen voor een verzameling van voertuigen K , in dit geval bussen. Hierbij heeft elk voertuig $k \in K$ een bijhorende capaciteit C_k . Deze capaciteit is

het maximaal aantal leerlingen dat zich gelijktijdig op dat voertuig kan bevinden. Het objectief is om ritten te construeren waarbij we de langste individuele rittijd over alle leerlingen minimaliseren.

De rittijd van een leerling is het tijdsverschil tussen het moment net na het instappen op de thuislocatie en het moment net na het uitstappen op school. Alternatieve definities staan beschreven in Bijlage A. De servicetijd voor het in- en uitstappen van leerling u is aangeduid met $t_u^{service}$. Het in- en uitstappen verloopt daarbij sequentieel; per voertuig kan slechts één leerling tegelijk in- en uitstappen.

Een tijdsvenster bepaalt wanneer we leerlingen op hun schoollocatie kunnen afzetten. Voor elke school s_u definiëren we het interval $[t_{s_u}^-, t_{s_u}^+]$. De volledige service (in dit geval het uitstappen) van de leerling op deze locatie moet binnen dit venster vallen. Het tijdstip $t_{s_u}^-$ is het vroegste moment waarop de service kan starten, terwijl $t_{s_u}^+$ het laatste moment is waarop deze service moet eindigen. Dit concept is geïllustreerd in Figuur 4.1. Daarnaast kan een voertuig, indien het te vroeg aankomt, wachten bij een knooppunt tot het tijdsvenster opent.



Figuur 4.1: Tijdsas die het vroegst en het laatst mogelijke uitvoeringsinterval van de service op knooppunt s_u weergeeft. Deze momenten zijn bepaald door het tijdsvenster van s_u .

Voertuigen mogen starten en eindigen op een willekeurige locatie (OVRP). Na het afzetten van leerlingen is het toegestaan andere leerlingen op te halen. Daarnaast is ook mixed load toegestaan; leerlingen met verschillende bestemmingen mogen samen vervoerd worden.

De maximale waarde $\tau_{\max}$ dient als bovengrens voor aankomsttijden en is groter dan elke mogelijke aankomsttijd. Deze waarde komt overeen met de maximale eindtijd over alle tijdsvensters.

We definiëren $C_{\max}$ als het maximaal aantal leerlingen dat gelijktijdig op een voertuig kan zitten. Deze waarde is kleiner dan zowel het totaal aantal leerlingen als de maximale voertuigcapaciteit over alle voertuigen.

4.2.2 Netwerkconstructie

De willekeurige startlocatie (Open VRP) modelleren we door het toevoegen van een virtueel depot $n_{virtual}$. Elk voertuig mag éénmalig vertrekken en aankomen in dat virtuele depot.

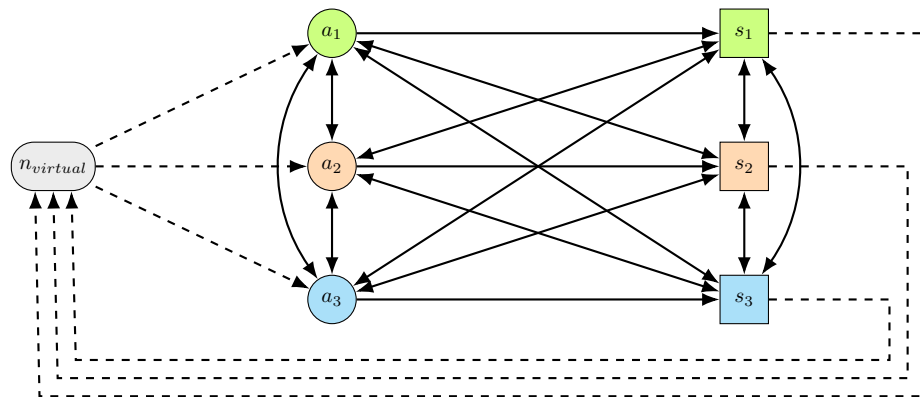
Het netwerk bestaat uit alle knooppunten $n \in N$, waaronder de adressen van de leerlingen, hun bijhorende scholen en het virtuele depot. Voor een geldige oplossing moet elk knooppunt $n \in N$, behalve het virtuele depot, exact één keer bezocht zijn. Dit impliceert volgende eigenschap:

Eigenschap Binaire Capaciteit van Bogen (BCB). Elke boog kan door hoogstens één voertuig gebruikt worden.

De verzameling geldige bogen is aangeduid met E . Deze bevat alle verbindingen die toegestaan zijn volgens de probleemstructuur. Bogen die niet geldig zijn, zoals een directe verbinding van de school van een leerling naar het adres van dezelfde leerling, zijn uitgesloten.

Elke boog (i, j) tussen knooppunten i en j heeft een bijhorende reistijd t_{ij} , die niet noodzakelijk symmetrisch is. Dit wil zeggen dat t_{ij} niet altijd gelijk is aan t_{ji} .

Een voorbeeld van een klein netwerk met alle geldige bogen staat in Figuur 4.2. De stippellijnen geven de bogen van en naar het virtuele depot weer, terwijl de volle pijlen de geldige bogen binnen het netwerk tonen.



Figuur 4.2: Voorbeeld van netwerkstructuur voor drie leerlingen met het virtueel depot, de adressen van leerlingen, de scholen en alle geldige bogen

4.2.3 Samenvatting van gebruikte notaties

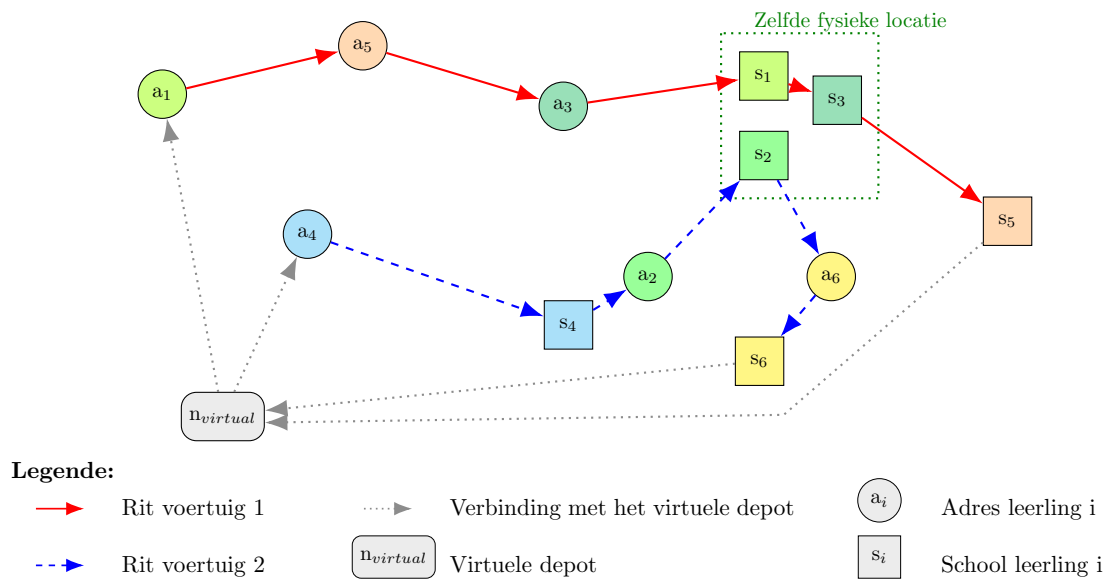
Als overzicht vatten we de verschillende notaties samen in Tabel 4.1.

Notatie	Beschrijving
$u \in U$	Leerling u uit de verzameling van alle leerlingen U
$k \in K$	Voertuig k uit de verzameling van alle voertuigen K
s_u	School van leerling u
a_u	Adres van leerling u
$s \in S$	School s uit de verzameling van alle scholen ($= s_u \forall u$)
$a \in A$	Thuisadres a uit de verzameling van alle adressen van leerlingen ($= a_u \forall u$)
$n \in N$	n is een element van de verzameling van alle knopen $N (= A \cup S \cup \{n_{virtual}\})$
$(i, j) \in E$	Boog (i, j) is een element van de verzameling van alle geldige bogen E
t_{ij}	Reistijd over boog (i, j)
$t_u^{service}$	Servicetijd voor het in- en uitstappen van leerling u
$n_{virtual}$	Virtueel depot
τ_{max}	De maximale tijd
C_k	De capaciteit van voertuig k
C_{max}	De maximale capaciteit over alle bogen
$[t_{s_u}^-, t_{s_u}^+]$	Tijdsvenster van school s_u

Tabel 4.1 Overzicht van de gebruikte notaties

4.2.4 Voorbeeldoplossing

Figuur 4.3 toont een voorbeeld van een oplossing met zes leerlingen. De oplossing haalt elke leerling op en zet hem later weer af met hetzelfde voertuig. We vervolledigen de lus door het virtuele depot toe te voegen.



Figuur 4.3: Voorbeeld van een oplossing met zes leerlingen

4.3 Wiskundige formulering van het probleem

Deze sectie bespreekt een mogelijke *integer linear programming* formulering van het probleem.

4.3.1 Beslissingsvariabelen

Eerst definiëren we drie beslissingsvariabelen voor de wiskundige formulering. De variabele x_{ijk} koppelt elk voertuig k aan een boog (i, j) . Op analoge manier koppelt de variabele y_{iju} elke leerling u aan een boog (i, j) . Tot slot gebruiken we een tijdsvariabele τ_{ik} die het tijdstip weergeeft waarop voertuig k knooppunt i kan verlaten. Deze variabele is nodig voor het objectief en om de maximale rittijd per leerling te controleren. Tabel 4.2 geeft een overzicht van de beslissingsvariabelen.

Variabele	Beschrijving
x_{ijk}	$= \begin{cases} 1 & \text{als voertuig } k \text{ boog } (i, j) \text{ neemt} \\ 0 & \text{anders} \end{cases}$
y_{iju}	$= \begin{cases} 1 & \text{als leerling } u \text{ boog } (i, j) \text{ neemt} \\ 0 & \text{anders} \end{cases}$
τ_{ik}	≥ 0 : tijdstip waarop voertuig k kan vertrekken bij knooppunt i (na de servicetijd)

Tabel 4.2 Overzicht van de beslissingsvariabelen voor het basisprobleem

4.3.2 Formulering

Objectief (4.1) minimaliseert de maximale individuele rittijd over alle leerlingen, aangeduid als $maxRideTime$. Beperking (4.2) koppelt $maxRideTime$ aan de werkelijke rittijd, berekend als het verschil tussen afzet- (τ_{s_uk}) en ophaaltijdstip (τ_{a_uk}).

$$\min \maxRideTime \tag{4.1}$$

$$\text{s.t. } \maxRideTime \geq \tau_{s_uk} - \tau_{a_uk} \quad \forall u \in U, k \in K \tag{4.2}$$

De volgende drie beperkingen zorgen samen voor een correcte opbouw van de voertuigritten. Beperking (4.3) garandeert dat elke knoop (behalve het virtuele depot) precies één inkomende boog heeft. Daarnaast verzekert Beperking (4.4) dat elk voertuig dat een knooppunt binnenrijdt, dit knooppunt ook verlaat. Vervolgens legt Beperking (4.5) vast dat elk voertuig hoogstens één keer vertrekt uit het virtuele depot.

$$\sum_{k \in K} \sum_{\substack{i \in N \\ \text{met } (i,j) \in E}} x_{ijk} = 1 \quad \forall j \in N \setminus \{n_{virtual}\} \quad (4.3)$$

$$\sum_{\substack{i \\ \text{met } (i,n) \in E}} x_{ink} - \sum_{\substack{j \\ \text{met } (n,j) \in E}} x_{njk} = 0 \quad \forall k \in K, \forall n \in N \quad (4.4)$$

$$\sum_{\substack{j \\ \text{met } (n_{virtual},j) \in E}} x_{n_{virtual}jk} \leq 1 \quad \forall k \in K \quad (4.5)$$

Beperkingen (4.6 - 4.7) leggen de tijdsvoorwaarden op. Als eerste stelt Beperking (4.6) dat we een leerling altijd eerder moeten ophalen dan afzetten. Vervolgens zorgt Beperking (4.7) dat de tijdsvariabelen correct zijn. Als de boog (i, j) is gekozen, dan moet de tijd in j minstens gelijk zijn aan de tijd in i verhoogd met reistijd tussen de twee knooppunten en de servicetijd. Deze beperking voorkomt ook dat er losse subcycli ontstaan. We veronderstellen hierbij dat het niet mogelijk is dat zowel de afstand tussen twee knooppunten als de bijhorende servicetijd nul zijn.

$$\tau_{s_{uk}} - \tau_{a_{uk}} \geq 0 \quad \forall u \in U, \forall k \in K \quad (4.6)$$

$$\tau_{ik} + (t_{ij} + t_j^{service}) \cdot x_{ijk} \leq \tau_{jk} + \tau_{max} \cdot (1 - x_{ijk}) \quad \forall k \in K, \forall (i, j) \in E \text{ met } j \neq n_{virtual} \quad (4.7)$$

Volgende vijf beperkingen garanderen dat elke leerling een geldige rit volgt van thuis naar school. Beperkingen (4.8 - 4.9) zorgen dat elke leerling exact één keer vertrekt vanuit het thuisadres en daar niet terugkeert. Op gelijkaardige wijze garanderen Beperkingen (4.10 - 4.11) dat elke leerling precies één keer aankomt op school en daar niet opnieuw vertrekt. Beperking (4.12) houdt de flow in alle andere knooppunten in evenwicht. Daarnaast voorkomt Beperking (4.13) dat leerlingen vertrekken vanuit het virtueel depot.

$$\sum_{\substack{j \in N \\ (a_u, j) \in E}} y_{a_u j u} = 1 \quad \forall u \in U \quad (4.8)$$

$$\sum_{\substack{i \in N \\ (i, a_u) \in E}} y_{i a_u u} = 0 \quad \forall u \in U \quad (4.9)$$

$$\sum_{\substack{i \in N \\ (i, s_u) \in E}} y_{is_u u} = 1 \quad \forall u \in U \quad (4.10)$$

$$\sum_{\substack{j \in N \\ (s_u, j) \in E}} y_{s_u j u} = 0 \quad \forall u \in U \quad (4.11)$$

$$\sum_{\substack{i \\ \text{met } (i, n) \in E}} y_{inu} - \sum_{\substack{j \\ \text{met } (n, j) \in E}} y_{nju} = 0 \quad \forall u \in U, \forall n \in N \setminus \{a_u, s_u\} \quad (4.12)$$

$$y_{n_{virtual} j u} = 0 \quad \forall u \in U, \forall j \in N \text{ met } (n_{virtual}, j) \in E \quad (4.13)$$

Beperking (4.14) garandeert dat de capaciteit van een voertuig niet overschreden wordt. Het totale aantal leerlingen op een boog mag nooit groter zijn dan de capaciteit van het voertuig dat deze boog neemt. Deze beperking is geldig dankzij Eigenschap BCB (4.2.2).

$$\sum_{u \in U} y_{iju} \leq C_k + C_{max} \cdot (1 - x_{ijk}) \quad \forall (i, j) \in E, \forall k \in K \quad (4.14)$$

Vervolgens koppelt Beperking (4.15) de leerlingenritten aan de voertuigritten. Een leerling mag een boog enkel gebruiken indien een voertuig deze boog ook neemt.

$$y_{iju} \leq \sum_{k \in K} x_{ijk} \quad \forall (i, j) \in E, \forall u \in U \quad (4.15)$$

Tot slot zijn er nog twee beperkingen om de tijdsvensters af te dwingen. Beperking (4.16) garandeert dat een voertuig pas leerlingen kan afzetten na de start van het tijdsvenster. Daarnaast legt Beperking (4.17) vast dat een leerling voor het einde van het tijdsvenster afgezet moet zijn.

$$(t_{s_u}^- + t_{s_u}^{service}) \cdot \sum_{\substack{i \\ \text{met } (i, s_u) \in E}} x_{is_u k} \leq \tau_{s_u k} \quad \forall u \in U, \forall k \in K \quad (4.16)$$

$$\tau_{s_u k} \leq t_{s_u}^+ \quad \forall u \in U, \forall k \in K \quad (4.17)$$

4.3.3 Andere geldige beperkingen (*valid inequalities*)

Deze extra beperkingen zijn niet strikt noodzakelijk voor de correctheid van het algoritme, maar ze verkleinen de zoekruimte en verminderen daardoor de rekentijd. Vaak volgen deze beperkingen uit één of meerdere noodzakelijke voorwaarden.

Beperking (4.18) garandeert dat elke leerling wordt opgehaald en afgezet door één voertuig. Beperking (4.19) garandeert Eigenschap BCB (4.2.2). Verder zetten we de tijdswaarden in knooppunten die niet bezocht worden door een voertuig op nul met behulp van Beperking (4.20). Ook is vastgelegd dat leerlingen nooit aankomen in het virtuele depot in Beperking (4.21). Ten slotte zorgt Beperking (4.22) ervoor dat elk knooppunt, behalve het virtuele depot, maximaal één uitgaande boog heeft.

$$\sum_{\substack{j \\ \text{met } (a_u, j) \in E}} x_{ajk} - \sum_{\substack{i \\ \text{met } (i, s_u) \in E}} x_{isk} = 0 \quad \forall u \in U, \forall k \in K \quad (4.18)$$

$$\sum_{k \in K} x_{ijk} \leq 1 \quad \forall (i, j) \in E \quad (4.19)$$

$$\tau_{jk} \leq 0 + \tau_{max} \cdot \sum_{\substack{i \\ \text{met } (i, j) \in E}} x_{ijk} \quad \forall j \in N, \forall k \in K \quad (4.20)$$

$$y_{in_{virtual}u} = 0 \quad \forall u \in U, \forall i \in N \text{ met } (i, n_{virtual}) \in E \quad (4.21)$$

$$\sum_{k \in K} \sum_{\substack{j \in N \\ \text{met } (i, j) \in E}} x_{ijk} = 1 \quad \forall i \in N \setminus \{n_{virtual}\} \quad (4.22)$$

4.3.4 Alternatieve formulering met flow-variabelen

De formulering in Paragraaf 4.3.2 heeft als voordeel dat ze direct inzicht biedt in welke bogen door elk van de leerlingen gebruikt worden. Dit maakt het opstellen van plannings eenvoudiger en kan nuttig zijn voor toekomstige uitbreidingen. Een alternatieve formulering op basis van flowvariabelen is opgenomen in Bijlage B. Deze formulering bevat minder beslissingsvariabelen, waardoor het potentieel sneller opgelost kan worden.

4.4 Mogelijke uitbreiding: transferpunten

Een interessante verdere onderzoekspiste is het toelaten van transfers tussen voertuigen. Het doel van deze uitbreiding is om te onderzoeken of transfers rittijden kunnen verminderen en of dit uitgebreide probleem oplosbaar blijft binnen redelijke rekentijd.

4.4.1 Definitie transferpunten en veronderstellingen

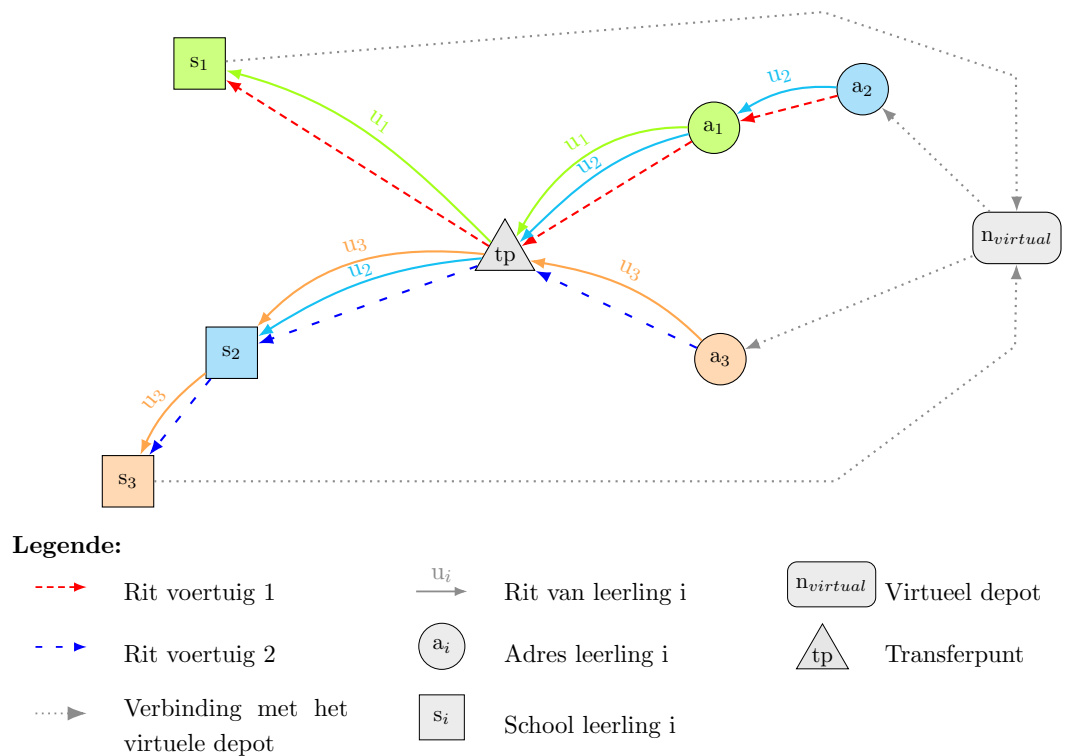
Transfers zijn toegestaan op een vooraf gedefinieerde verzameling transferpunten TP , die ook deel uitmaken van de verzameling N . Op een transferpunt $p \in TP$ kunnen leerlingen overstappen tussen verschillende voertuigen. Het toevoegen van transferpunten vergroot de verzameling geldige bogen. Naast de reeds gedefinieerde verbindingen in Paragraaf 4.2.2 voegen we alle verbindingen tussen knooppunten en transferpunten, in beide richtingen, toe.

Vervolgens veronderstellen we dat er bij transfers volledige temporele en plaatselijke synchronisatie vereist is. Dit betekent dat alle betrokken voertuigen aanwezig moeten zijn tijdens het volledige uit- en instapproces. Een leerling mag pas beginnen uitstappen zodra alle voertuigen die nodig zijn voor de transfer op het transferpunt zijn aangekomen. Het instappen in het volgende voertuig kan pas starten wanneer alle leerlingen, die een transfer ondergaan, volledig zijn uitgestapt. Pas wanneer alle betrokken leerlingen opnieuw zijn ingestapt, mogen de voertuigen vertrekken.

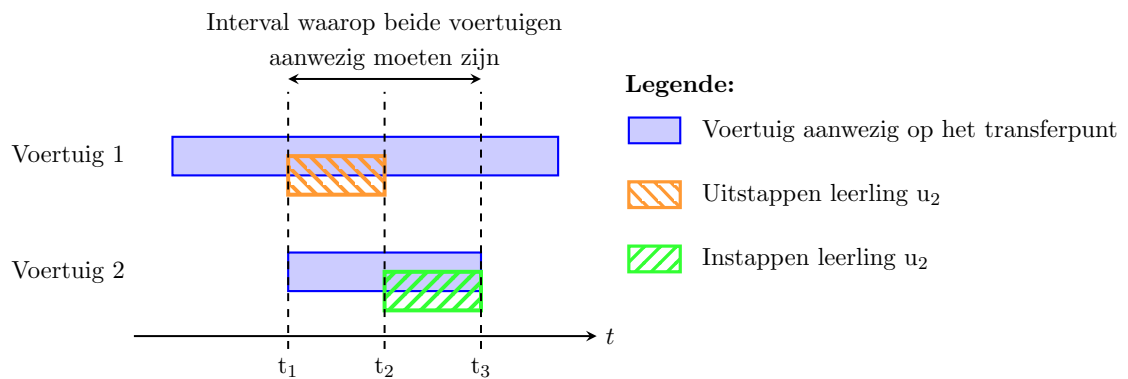
De tijd die nodig is voor het in- of uitstappen van een leerling op een transferpunt komt overeen met de servicetijd. Ook op transferpunten verloopt het in- en uitstappen per voertuig sequentieel. Daarnaast veronderstellen we dat elke leerling maximaal één transferpunt bezoekt, ongeacht of er effectief een transfer van deze leerling plaatsvindt.

4.4.2 Voorbeeldoplossing

Figuur 4.4 toont een voorbeeldoplossing met één transferpunt. De gebogen pijlen geven de ritten van de leerlingen weer, terwijl de stippellijnen de voertuigritten voorstellen. Op het transferpunt maakt leerling u_2 een overstap van voertuig 1 naar voertuig 2. Daarnaast visualiseert Figuur 4.5 de vereiste synchronisatie tussen beide voertuigen.



Figuur 4.4: Illustratie van een oplossing met één transferpunt; leerling 2 maakt een transfer van voertuig 1 naar voertuig 2



Figuur 4.5: Tijdsas horend bij Figuur 4.4. Het tijdstip t_1 geeft het moment weer waarop alle betrokken voertuigen zijn aangekomen en leerlingen dus mogen beginnen uitstappen. Daarnaast geeft t_2 het moment aan dat alle leerlingen die moesten uitstappen dit gedaan hebben, vanaf dit moment kan het instappen beginnen. Tot slot markeert t_3 het moment waarop alle leerlingen opnieuw zijn ingestapt, waarna de voertuigen hun rit kunnen verderzetten. Het interval $[t_1, t_3]$ is de periode waarin alle betrokken voertuigen aanwezig moeten zijn.

4.4.3 Extra (hulp-)beslissingsvariabelen voor transferpunten

Om transferpunten te modelleren, introduceren we vier extra beslissingsvariabelen. Deze variabelen zijn geïnspireerd op het werk van Rais et al. (2014). Ze zijn uitsluitend gedefinieerd voor transferpunten of voor bogen waarvan minstens één knoop een transferpunt is. Indien er geen transferpunten aanwezig zijn, maken we deze variabelen niet aan.

De variabele m_{kl}^{pu} geeft aan of een bepaalde leerling een transfer uitvoert tussen twee voertuigen op een transferpunt. Daarnaast geeft de variabele z_{ij}^{ku} aan of een leerling aanwezig is op een boog (i, j) die verbonden is met een transferpunt. Vervolgens indiceert δ_{pkl}^{TP} of er een transfer bestaat van voertuig k naar voertuig l op punt p . Tot slot beschrijft $\tau_{pk}^{startTP}$ het tijdstip waarop voertuig k op transferpunt p is aangekomen en de transfer kan starten. Dit komt overeen met tijdstip t_1 in Figuur 4.5. Een overzicht van deze variabelen samen met de gebruikte notaties is te vinden in Tabel 4.3.

Notatie/ Variabele	Beschrijving
$p \in TP$	p is een transferpunt uit de verzameling van alle transferpunten TP
m_{kl}^{pu}	$= \begin{cases} 1 & \text{als leerling } u \text{ van voertuig } k \text{ naar voertuig } l \text{ overstapt op transferpunt } p \\ 0 & \text{anders} \end{cases}$
z_{ij}^{ku}	$= \begin{cases} 1 & \text{als leerling } u \text{ aanwezig is op voertuig } k \text{ op boog } (i, j), \\ 0 & \text{anders} \end{cases}$
δ_{pkl}^{TP}	$= \begin{cases} 1 & \text{als er een transfer gebeurt van voertuig } k \text{ naar } l \text{ op transferpunt } p \\ 0 & \text{anders} \end{cases}$
$\tau_{pk}^{startTP}$	≥ 0 : het tijdstip waarop voertuig k is toegekomen op transferpunt p en de transfer kan beginnen

Tabel 4.3 Overzicht van de gebruikte notaties en beslissingsvariabelen voor de uitbreiding met transferpunten

4.4.4 Formulering

We starten met de beperkingen die ervoor zorgen dat alle hulpvariabelen correct worden toegewezen. Vervolgens bespreken we de bijkomende beperkingen die de logica op transferpunten definieert. Ten slotte volgt een overzicht van de beperkingen uit het basisprobleem die behouden blijven en andere beperkingen die een aanpassing vereisen.

Toewijzing van hulpvariabelen

Allereerst definiëren we drie beperkingen die garanderen dat de hulpvariabele z_{ij}^{ku} de juiste waarde krijgt. De drie beperkingen (4.23 - 4.25) koppelen de variabelen x , y en z . Indien

voertuig k en leerling u samen boog (i, j) gebruiken, dan zit leerling u in voertuig k op deze boog. Dit volgt uit Eigenschap BCB (4.2.2).

$$z_{ij}^{ku} \leq x_{ijk} \quad \forall k \in K, \forall u \in U, \forall (i, j) \in E \text{ met } (i \in TP \vee j \in TP) \quad (4.23)$$

$$z_{ij}^{ku} \leq y_{iju} \quad \forall k \in K, \forall u \in U, \forall (i, j) \in E \text{ met } (i \in TP \vee j \in TP) \quad (4.24)$$

$$z_{ij}^{ku} \geq x_{ijk} + y_{iju} - 1 \quad \forall k \in K, \forall u \in U, \forall (i, j) \in E \text{ met } (i \in TP \vee j \in TP) \quad (4.25)$$

Beperking (4.26) zorgt ervoor dat de variabele m_{kl}^{pu} correct wordt ingesteld. Deze beperking is geïntroduceerd door Rais et al. (2014). Indien leerling u van voertuig k naar voertuig l overstapt op transferpunt p , dan krijgt de variabele m_{kl}^{pu} de waarde één. Een transfer vindt plaats wanneer een leerling vertrekt met een ander voertuig dan waarmee hij is aangekomen.

$$\sum_{\substack{i \\ \text{met } (i,p) \in E}} z_{ip}^{ku} + \sum_{\substack{j \\ \text{met } (p,j) \in E}} z_{pj}^{lu} \leq m_{kl}^{pu} + 1 \quad \forall p \in TP, \forall u \in U, \forall k, l \in K \text{ met } k \neq l \quad (4.26)$$

Beperking (4.27) kent de juiste waarde toe aan beslissingsvariabele δ_{pkl}^{TP} . Indien er minstens één leerling van voertuig k naar l overstapt, dan geldt dat er een transfer gebeurt tussen deze voertuigen en zetten we de δ_{pkl}^{TP} -waarde op één.

$$\sum_{u \in U} m_{kl}^{pu} \leq \delta_{pkl}^{TP} \cdot \min(C_k, C_l) \quad \forall p \in TP, \forall k, l \in K \quad (4.27)$$

Beperkingen voor transferlogica

Vervolgens definiëren we de beperkingen die de transferlogica afdwingen. We beginnen met de beperkingen die de synchronisatie op transferpunten garanderen. Hierbij veronderstellen we dat enkel de leerlingen die effectief een transfer ondergaan, uitstappen uit het huidige voertuig en terug instappen in een ander voertuig. De andere leerlingen blijven op hun voertuig zitten. In Sectie 4.5 geven we hier een alternatief voor.

Ten eerste garandeert Beperking (4.28) dat de starttijd van een transfer voor voertuig k op transferpunt p later moet vallen dan het vroegst mogelijke aankomstmoment van voertuig k op dat transferpunt p .

$$\tau_{pk}^{startTP} + \tau_{max}(1 - x_{ipk}) \geq \tau_{ik} + t_{ip} \cdot x_{ipk} \quad \forall p \in TP, \forall k \in K, \forall i \in N \text{ met } (i, p) \in E \quad (4.28)$$

Beperking (4.29) zorgt ervoor dat voertuig k pas kan vertrekken op transferpunt p nadat alle leerlingen die moeten uitstappen uit voertuig k op transferpunt p effectief zijn uitgestapt en de leerlingen die moeten instappen in voertuig k op transferpunt p terug zijn ingestapt.

$$\tau_{pk} \geq \tau_{pk}^{startTP} + \sum_{u \in U} \sum_{l \in K} m_{kl}^{pu} \cdot t_u^{service} + \sum_{u \in U} \sum_{l \in K} m_{lk}^{pu} \cdot t_u^{service} \quad (4.29)$$

$$\forall p \in TP, \forall k \in K$$

Vervolgens bepaalt Beperking (4.30) dat, bij een transfer van voertuig k naar voertuig l op transferpunt p , alle leerlingen die vanuit k naar een ander voertuig overstappen eerst allemaal moeten uitstappen. Pas daarna mag het instappen in voertuig l beginnen. Nadat alle leerlingen in l zijn ingestapt, mag l vertrekken.

$$\tau_{pl} + \tau_{\max} \cdot (1 - \delta_{pkl}^{TP}) \geq \tau_{pk}^{startTP} + \sum_{u \in U} \sum_{h \in K} m_{kh}^{pu} \cdot t_u^{service} + \sum_{u \in U} \sum_{h \in K} m_{hl}^{pu} \cdot t_u^{service} \quad (4.30)$$

$$\forall p \in TP, \forall k, l \in K$$

Daarna garandeert Beperking (4.31) dat, wanneer een transfer plaatsvindt van voertuig k naar voertuig l op transferpunt p , voertuig k pas mag vertrekken zodra alle leerlingen die overstappen naar voertuig l zijn ingestapt en l dus klaar is om te vertrekken.

$$\tau_{pk} + \tau_{\max} \cdot (1 - \delta_{pkl}^{TP}) \geq \tau_{pl} \quad \forall p \in TP, \forall k, l \in K \quad (4.31)$$

Beperking (4.32) bepaalt dat, wanneer een transfer plaatsvindt van voertuig k naar voertuig l , voertuig k pas mag beginnen met de transfer zodra l aanwezig is en klaar is om de transfer te starten.

$$\tau_{pk}^{startTP} + \tau_{\max} \cdot (1 - \delta_{pkl}^{TP}) \geq \tau_{pl}^{startTP} \quad \forall p \in TP, \forall k, l \in K \quad (4.32)$$

Vervolgens zorgt Beperking (4.33) dat er maximaal één effectieve transfer per leerling gebeurt.

$$\sum_{l \in K} \sum_{k \in K} \sum_{p \in TP} m_{kl}^{pu} \leq 1 \quad \forall u \in U \quad (4.33)$$

Beperking (4.34) bepaalt dat elke leerling maximaal één keer mag aankomen op een transferpunt.

$$\sum_{p \in TP} \sum_{\substack{i \\ (i,p) \in E}} y_{ipu} \leq 1 \quad \forall u \in U \quad (4.34)$$

Daarna volgt Beperking (4.35), die stelt dat een leerling maximaal één keer mag vertrekken vanuit een transferpunt. Deze beperking is een geldige ongelijkheid die de oplossingsruimte mogelijk beperkt en daardoor hopelijk de rekentijd helpt verlagen.

$$\sum_{p \in TP} \sum_{\substack{j \\ (p,j) \in E}} y_{pju} \leq 1 \quad \forall u \in U \quad (4.35)$$

Aanpassingen aan bestaande beperkingen van het basisprobleem

Ten slotte bespreken we de aanpassingen aan de beperkingen van het basisprobleem (Sectie 4.3).

Ten eerste passen we Beperking (4.2), die de objectiefwaarde maxRideTime definieert, aan naar Beperking (4.36) omdat het niet langer vereist is dat hetzelfde voertuig een leerling zowel ophaalt als afzet. Hierbij is het essentieel dat alle knooppunten die het voertuig niet bezoekt een tijdsvariabele van nul krijgen, zoals vastgelegd in Beperking 4.20. Beperking 4.20 is niet langer alleen geldig, maar ook noodzakelijk.

$$\text{maxRideTime} \geq \sum_k \tau_{s_u k} - \sum_k \tau_{a_u k} \quad \forall u \in U \quad (4.36)$$

Ook Beperking 4.6 passen we aan naar Beperking (4.37). De afzettijd van een leerling moet groter zijn dan de ophaaltijd.

$$\sum_k \tau_{s_u k} - \sum_k \tau_{a_u k} \geq 0 \quad \forall u \in U \quad (4.37)$$

Vervolgens vervangen we Beperking (4.18) door Beperking (4.38). Beperking (4.18) is geldig, maar is niet strikt noodzakelijk. Ze garandeert dat elke leerling moet worden opgehaald en afgezet.

$$\sum_k \sum_{\substack{j \\ \text{met } (a_u, j) \in E}} x_{a_u j k} - \sum_k \sum_{\substack{i \\ \text{met } (i, s_u) \in E}} x_{i s_u k} = 0 \quad \forall u \in U \quad (4.38)$$

Beperkingen (4.4 - 4.5, 4.8 - 4.12, 4.14 - 4.17 en 4.19 - 4.21) blijven ongewijzigd en gelden ook voor transferpunten.

Beperking (4.3) blijft geldig voor alle knooppunten behalve de transferpunten. Voor transferpunten definiëren we Beperking (4.39), die stelt dat een voertuig maximaal één keer mag aankomen op een bepaald transferpunt.

$$\sum_{\substack{i \in N \\ (i,p) \in E}} x_{ipk} \leq 1 \quad \forall p \in TP, \forall k \quad (4.39)$$

Ook Beperking (4.7) blijft behouden, maar enkel voor knooppunten die geen transferpunten zijn. Transferpunten volgen hun eigen tijdslogica zoals beschreven in beperkingen (4.28 - 4.32).

Tot slot blijft Beperking (4.22) geldig voor alle niet-transferpunten. Voor transferpunten definiëren we een bijkomende beperking, namelijk Beperking (4.40), die stelt dat een voertuig maximaal één keer mag vertrekken vanuit een transferpunt. Deze beperking is opnieuw geldig, maar niet noodzakelijk.

$$\sum_{\substack{j \in N \\ (p,j) \in E}} x_{pjk} \leq 1 \quad \forall p \in TP, \forall k \in K \quad (4.40)$$

4.5 Parameters

De implementatie laat toe om parameters eenvoudig aan te passen via een configuratiebestand. In deze sectie geven we een overzicht van de belangrijkste parameters en de standaardwaarden die in de experimenten zijn gebruikt.

Tijdseenheid

Tijd kunnen we uitdrukken in verschillende eenheden. In deze implementatie kiezen we voor een tijdseenheid van 10 seconden. Dit betekent dat alle tijdswaarden intern uitgedrukt zijn als een veelvoud van 10 seconden. Deze keuze biedt een goede balans tussen nauwkeurigheid en rekentijd: werken in seconden is te gedetailleerd en verhoogt de rekentijd, terwijl minuten te grof zijn en onnauwkeurige resultaten opleveren.

Busvloot

De busvloot en bijhorende capaciteiten zijn gedefinieerd in de inputverzameling. Indien gewenst kan een aangepaste vloot worden opgegeven via het configuratiebestand. Voor onze experimenten gebruiken we de vloot horend bij de inputverzameling.

Maximale rekentijd

De solver stopt na een instelbare maximale rekentijd. Indien geen optimale oplossing gevonden is, wordt de beste oplossing tot dan toe teruggegeven. Standaard is deze limiet ingesteld op 3600 seconden (1 uur).

Tijdsvensters

Tijdsvensters blijven standaard ongewijzigd, maar kunnen verlengd worden met een gekozen marge. In onze experimenten hanteren we geen verlenging.

Transferpunten

Indien transferpunten worden beschouwd, definiëren we vijf parameters:

1. Transfers toelaten op parkings: standaard uitgeschakeld.
2. Transfers toelaten op scholen: standaard uitgeschakeld.
3. Maximaal aantal transfers per voertuig: in de formulering en in onze experimenten beperken we dit tot één.
4. Twee opties voor uitstapgedrag op transferpunten zijn mogelijk:
 - Enkel leerlingen die effectief een transfer uitvoeren stappen uit.
 - Alle leerlingen stappen uit en vervolgens opnieuw in, ook indien zij met hetzelfde voertuig verder reizen.

Beide opties worden getest, maar standaard en in de formulering hanteren we de eerste optie.

5. Aankomstsynchronisatie van voertuigen: bepaalt of leerlingen mogen uitstappen voordat alle betrokken voertuigen zijn aangekomen. Standaard is dit niet toegestaan.

4.6 Resultaten wiskundige formulering

Voor de experimenten gebruiken we een computer uitgerust met een AMD Ryzen 9 5950X 16-core processor (3.4 GHz) en het Windows 11 besturingssysteem. De gebruikte *Mixed Integer Linear Programming* solver is hierbij Gurobi 12.0.2. We voeren de experimenten uit op deelverzamelingen van de instanties die groepen van gemeenten representeren, zoals beschreven in Hoofdstuk 6. We beschouwen zeven verschillende groottes van input, gekarakteriseerd door het aantal leerlingen (U), voertuigen (K) en scholen (S). De maximale rekentijd voor de solver bedraagt één uur. Indien binnen deze tijd geen optimale oplossing wordt gevonden, rapporteert het algoritme de beste oplossing tot dan toe. Wanneer er geen oplossing beschikbaar is binnen de tijdslimiet, wordt in plaats van de objectieve waarde een streepje weergegeven.

4.6.1 Resultaten voor het basisprobleem

We starten met experimenten voor het basisprobleem. De resultaten in Tabel 4.4 tonen aan dat de rekentijd aanzienlijk lager is wanneer de optimale objectiefwaarde overeenkomt met de ondergrens. Deze ondergrens berekenen we als de langste rechtstreekse rittijd van een leerling naar zijn school. In dergelijke gevallen kan de solver snel aantonen dat deze oplossing optimaal is, waardoor het oplossen aanzienlijk versnelt.

De rekentijd blijft acceptabel tot een probleemgrootte van acht leerlingen, drie voertuigen en drie scholen. Bij grotere instanties blijkt dat optimale oplossingen enkel haalbaar zijn wanneer de optimale objectiefwaarde gelijk is aan de ondergrens. Voor de verzameling van twintig leerlingen lukt het zelfs niet meer om binnen de tijdslimiet van één uur een geldige oplossing te vinden.

Hieruit concluderen we dat het algoritme niet schaalbaar is voor realistische instanties. De samengevatte resultaten zijn weergegeven in Tabel 4.4.

4.6.2 Resultaten uitbreiding transferpunten

Het effect van transferpunten onderzoeken we op dezelfde kleine instanties als in de vorige paragraaf, met uitzondering van instanties die:

- niet optimaal uitgerekend konden worden binnen één uur,
- geen mogelijke transferpunten bevatten,
- of waarbij de ondergrens al bereikt wordt.

Zoals weergegeven in Tabel 4.5, is het merendeel van deze instanties niet optimaal opgelost binnen de tijdslimiet van één uur. Hierdoor is een directe vergelijking van de resultaten moeilijk. Wel kunnen we vaststellen dat de toevoeging van transferpunten in sommige gevallen leidt tot een verbetering van de langste rittijd, zelfs bij kleine probleeminstanties. Deze verbetering gaat echter gepaard met een aanzienlijke toename van de rekentijd. De gevallen waarin een optimale oplossing nog binnen één uur kan worden berekend, doen zich vrijwel uitsluitend voor wanneer er geen verbetering ten opzichte van de oorspronkelijke oplossing mogelijk is.

Om te onderzoeken of een langere rekentijd tot een bewezen optimum zou leiden, hebben we één verzameling (`set1_U7_K2_S3_T2.2023`) gedurende vijf uur laten doorrekenen. De objectiefwaarde bleef ongewijzigd ten opzichte van het resultaat na één uur en was nog steeds niet bewezen optimaal.

We concluderen dat transferpunten potentieel bieden voor het verbeteren van de langste rittijd, maar dat de huidige exacte formulering computationeel niet haalbaar is.

4.6.3 Conclusie resultaten wiskundige formulering

De experimenten tonen aan dat onze exacte aanpak niet schaalbaar is voor instanties van realistische grootte. De introductie van transferpunten verhoogt de complexiteit aanzienlijk, waardoor de rekentijd verder toeneemt.

Om het probleem voor realistische instanties op te lossen, is een meer schaalbare en flexibele methode vereist. Daarom onderzoeken we in Hoofdstuk 5 een heuristische aanpak. Gezien de sterke impact van transferpunten op de complexiteit, zullen we deze in het vervolg van het onderzoek niet meenemen.

Grootte	Instantie	Bewezen optimaal?	Langste rittijd [min]	Ondergrens [min]	Rekentijd [s]
U5 K2 S2	Set1_U5_K2_S2_T1_2023	✓	43.2	31.0	0.421
	Set2_U5_K2_S2_T1_2023	✓	25.8	25.8	0.242
	Set3_U5_K2_S2_T1_2023	✓	33.8	28.3	0.853
	Set4_U5_K2_S2_T1_2023	✓	40.0	30.5	0.382
U6 K3 S2	Set1_U6_K3_S2_T1_2023	✓	31.0	31.0	0.088
	Set2_U6_K3_S2_T1_2023	✓	17.0	16.0	15.405
	Set3_U6_K3_S2_T0_2023	✓	14.3	14.3	0.055
	Set4_U6_K3_S2_T2_2023	✓	36.7	27.5	8.721
U7 K2 S3	Set1_U7_K2_S3_T2_2023	✓	42.7	31.0	16.111
	Set2_U7_K2_S3_T1_2023	✓	60.7	53.0	11.401
	Set3_U7_K2_S3_T0_2023	✓	60.2	42.3	32.992
	Set4_U7_K2_S3_T1_2023	✓	42.5	23.8	24.322
U8 K3 S3	Set1_U8_K3_S3_T2_2023	✓	31.0	31.0	4.098
	Set2_U8_K3_S3_T1_2023	✓	16.5	16.5	3.224
	Set3_U8_K3_S3_T1_2023	✓	57.0	32.3	229.783
	Set4_U8_K3_S3_T1_2023	✓	33.5	33.5	14.150
U10 K3 S3	Set1_U10_K3_S3_T2_2023	✗	44.8	31.0	3600.239
	Set2_U10_K3_S3_T1_2023	✓	20.7	20.7	715.162
	Set3_U10_K3_S3_T0_2023	✓	38.2	38.2	79.114
	Set4_U10_K3_S3_T2_2023	✗	52.7	41.2	3600.165
U20 K4 S5	Set1_U20_K4_S5_T_2023	✗	-	29.8	3600.044
	Set2_U20_K4_S5_T_2023	✗	-	78.2	3600.299
	Set3_U20_K4_S5_T_2023	✗	-	49.5	3600.518
	Set4_U20_K4_S5_T_2023	✗	-	64.5	3600.403

Tabel 4.4 Resultaten wiskundige formulering

Instantie	Geen transfers		Transfers op parkings				Transfers op scholen		Transfers op parkings en scholen	
			Alle leerlingen stappen uit		Alleen leerlingen met effectieve transfer stappen uit		Alleen leerlingen met effectieve transfer stappen uit		Alleen leerlingen met effectieve transfer stappen uit	
	Langste rittijd [min]	Bewezen optimaal? (Reken-tijd [s])	Langste rittijd [min]	Bewezen optimaal? (Reken-tijd [s])	Langste rittijd [min]	Bewezen optimaal? (Reken-tijd [s])	Langste rittijd [min]	Bewezen optimaal?	Langste rittijd [min]	Bewezen optimaal?
Set1_U5_K2_S2_T1_2023	43.2	✓ (0.421s)	43.2	✓ (7.899s)	43.2	✓ (7.445s)	43.2	✗	43.2	✗
Set3_U5_K2_S2_T1_2023	33.8	✓ (0.853s)	33.8	✓ (10.301s)	33.8	✓ (11.095s)	33.8	✗	33.8	✗
Set4_U6_K3_S2_T2_2023	36.7	✓ (8.721s)	33.7	✗	33.3	✗	35.0	✗	35.0	✗
Set1_U7_K2_S3_T2_2023	42.7	✓ (16.111s)	41.0	✗	40.3	✗	42.7	✗	41.0	✗
Set2_U7_K2_S3_T1_2023	60.7	✓ (11.401s)	60.7	✓ (595.281s)	60.7	✓ (1181.724s)	60.7	✗	60.7	✗
Set4_U7_K2_S3_T1_2023	42.5	✓ (24.322s)	35.2	✗	35.2	✗	35.2	✗	35.2	✗
Set3_U8_K3_S3_T1_2023	57.0	✓ (229.783s)	53.2	✗	52.7	✗	63.3	✗	58.2	✗

Tabel 4.5 Resultaten met transfers

Hoofdstuk 5

Heuristische methodologie

5.1 Inleiding

Hoofdstuk 4 heeft aangetoond dat de exacte aanpak niet schaalbaar is. Voor instanties met twintig leerlingen of meer bleek het algoritme niet langer uitvoerbaar binnen redelijke rekentijd. Dit vormt een probleem, aangezien de planningen in de praktijk aanzienlijk groter zijn. Bovendien toonde een verkennend onderzoek naar transferpunten aan dat deze de complexiteit van het probleem aanzienlijk verhogen, terwijl ze niet noodzakelijk tot veel betere oplossingen leiden.

We stellen daarom voor om over te gaan naar een meer flexibele aanpak. In deze heuristische methodologie is het doel om binnen een acceptabele rekentijd een oplossing te vinden die zo goed mogelijk is voor grotere probleeminstanties. Om de zoekruimte te beperken veronderstellen we dat we leerlingen ten vroegste 90 minuten voor het begin van het tijdsvenster van hun school mogen ophalen. Net als bij de exacte aanpak blijft het minimaliseren van de langste individuele rittijd het centrale objectief, terwijl we ernaar streven om zoveel mogelijk leerlingen in de planning op te nemen. We beschouwen hierbij het basisprobleem zonder transferpunten.

Door dit te doen bekomen we een Service-Maximization Minimax with Fixed-Fleet (SMM-FF) probleem. Het doel is de maximale rittijd te minimaliseren en daarbij zoveel mogelijk leerlingen te bedienen. Daarnaast introduceren we twee heuristische varianten voor dit SMM-FF. De eerste variant, Service-Maximization Minimax with Temporary-Fleet-Expansion (SMM-TFE) (zie Paragraaf 5.5.1), probeert de iteraties te versnellen bij moeilijk oplosbare instanties. De tweede variant, Full-Service Minimax with Fleet-Expansion (FSM-FE) (zie Paragraaf 5.5.2), garandeert dat er altijd een geldige oplossing bestaat waarin we alle leerlingen ophalen, maar laat toe dat we enkele extra voertuigen toevoegen. Pseudocode voor de SMM-FF heuristiek (inclusief de *ruin*-, *recreate*- en *compress schedule*-methoden) en haar varianten is opgenomen in Appendix C.

5.2 Bestaande methodes als startpunt voor de heuristiek

Voor de ontwikkeling van het heuristisch algoritme baseren we ons op twee bestaande technieken uit de literatuur.

5.2.1 SISRs als basisstructuur

Enerzijds is de basisstructuur van ons algoritme een aangepaste versie van het SISRs-algoritme (Christiaens & Vanden Berghe, 2020). Dit algoritme is oorspronkelijk ontworpen voor het oplossen van het standaard Capacitated Vehicle Routing Problem (CVRP). Het beschrijft ook enkele varianten en uitbreidingen, waaronder Open Vehicle Routing Problem (OVRP), tijdvensters en Pickup-and-Delivery Problem (PDP). SISRs is vooral bedoeld om grote instanties efficiënt op te lossen door middel van een iteratief proces waarbij het algoritme herhaaldelijk delen van oplossingen ruïneert en vervolgens op een greedy manier terug opbouwt.

Wij nemen deze structuur als startpunt, maar voeren belangrijke aanpassingen door om te voldoen aan de specifieke vereisten van ons probleem. De belangrijkste uitbreidingen zijn:

- Ondersteuning voor *pickup and delivery* met gegroepeerde eindbestemmingen
- Gebruik van een vooraf vastgelegde heterogene vloot
- Integratie van tijdvensters
- Aanpassing van de objectieffunctie: minimalisatie van de langste rittijd

SISRs is gekozen vanwege zijn eenvoud, snelheid, flexibiliteit en bewezen prestaties bij grootschalige problemen.

5.2.2 Tijdvenstertechniek

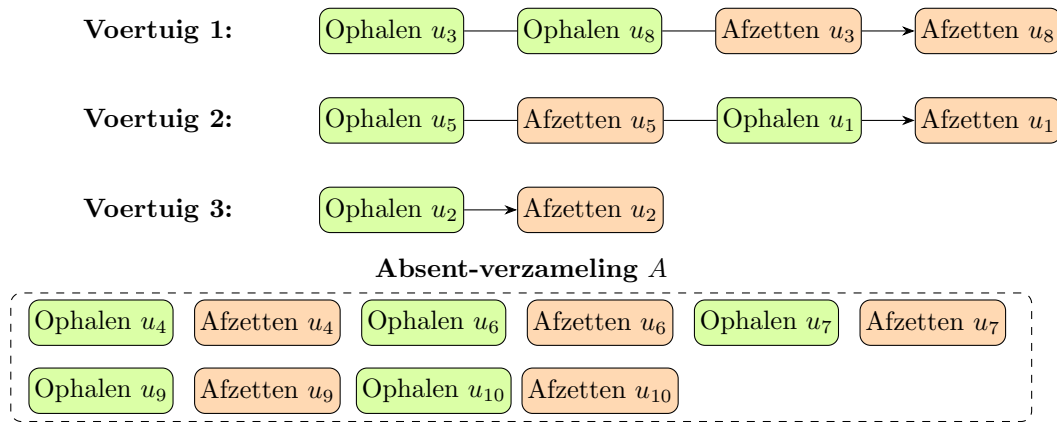
Anderzijds willen we efficiënt controleren of het invoegen van een ophaal- of afzetactie op een bepaalde locatie mogelijk is zonder tijdvensters te overschrijden. Hiervoor gebruiken we de techniek beschreven in “Iterated local search for the team orienteering problem with time windows” (Vansteenwegen et al., 2009). Deze methode houdt per actie in een rit vier kernvariabelen bij, namelijk wachttijd, aankomsttijd, starttijd van de service en maximale verschuiving (*maximum shift*).

Bij het controleren van een invoegpositie volstaat het met deze techniek om de verschuiving die door de invoeging wordt veroorzaakt te evalueren en vervolgens na te gaan of de daaropvolgende actie voldoende speling heeft in wachttijd of maximale verschuiving om deze invoeging toe te laten. Deze aanpak maakt snelle en nauwkeurige geldigheidscontrole mogelijk, wat essentieel is voor de efficiëntie van ons heuristisch algoritme.

5.3 Probleemmodellering en oplossingsvoorstelling

Elke leerling uit de inputverzameling stellen we voor als een actie-paar, dat bestaat uit het ophalen en het afzetten van een leerling. Beide acties hebben een bijhorend tijdsvenster. Voor het afzetten geldt het tijdsvenster van de bijhorende school; voor het ophalen berekenen we het tijdsvenster op basis van dat van de school. Zoals beschreven in Sectie 5.1 veronderstellen we dat we leerlingen ten vroegste 90 minuten voor de start van het schooltijdsvenster mogen ophalen. De start van het tijdsvenster van een leerling ligt dus op dit tijdstip verminderd met de servicetijd. Het einde van het tijdsvenster stemt overeen met het einde van het schooltijdsvenster verminderd met de rechtstreekse rijtijd van thuis naar school en de servicetijd.

De oplossing stellen we voor als een rit per voertuig. Elke rit bestaat uit een geordende reeks acties (ophalen of afzetten van een leerling). Daarnaast bevat de oplossing een *absent*-verzameling met alle acties die ontbreken in de huidige ritten. Dit is weergegeven in Figuur 5.1. Elke oplossing beschikt bovendien over een tijdsplanning waarbij acties zo vroeg mogelijk plaatsvinden en een samengedrukte planning waarbij we alle acties zo compact mogelijk willen inplannen. Voor elke actie in de planning houden we de vier variabelen voor de tijdsvenstertechniek bij (Vansteenwegen et al., 2009).



Figuur 5.1: Voorstelling van een oplossing voor 10 leerlingen (u_1 tot en met u_{10}): per voertuig een geordende vector van acties (ophalen u_i / afzetten u_i), en een verzameling met de niet-geplande acties (absent-verzameling A)

5.4 Service-Maximization Minimax with Fixed-Fleet (SMM-FF)

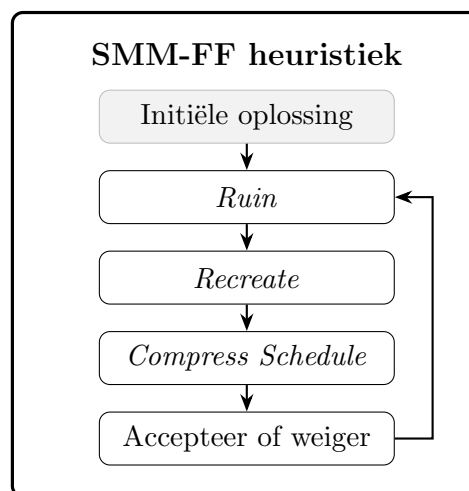
5.4.1 Local search metaheuristiek

De Service-Maximization Minimax with Fixed-Fleet (SMM-FF) heuristiek is een *local search*-metaheuristiek gebaseerd op *Simulated Annealing (SA)* die vertrekt vanuit het SISRs-algoritme van Christiaens en Vanden Berghe (2020).

Een overzicht van de algemene flow is weergegeven in Figuur 5.2. De procedure start met het genereren van een initiële oplossing (Paragraaf 5.4.2) en het instellen van een begintemperatuur T_0 . Vervolgens herhalen we iteraties bestaande uit drie fasen totdat de tijdslimiet is bereikt. De drie fasen zijn een *Ruin*-, *Recreate*- en *Compress schedule*-fase (Paragrafen 5.4.3 - 5.4.5).

Na elke iteratie evalueren we de nieuwe oplossing en beslissen we of we deze accepteren als lokaal en/of globaal beste oplossing. Verdere iteraties vertrekken terug vanuit de lokaal beste oplossing. Na elke iteratie verlaagt de temperatuur volgens een koelschema. Deze temperatuur bepaalt welke oplossingen we accepteren, zoals verder toegelicht in deze paragraaf.

Het vervolg van deze paragraaf bespreekt de specifieke aanpassingen aan de local search-metaheuristiek: een tijdsgebaseerd koelschema in plaats van een iteratiegebaseerd schema en een aangepast acceptatiecriterium om het gewijzigde objectief (minimaliseren van de langste rittijd) te ondersteunen. De bijhorende pseudocode is opgenomen in Algoritme 2 in Appendix C.

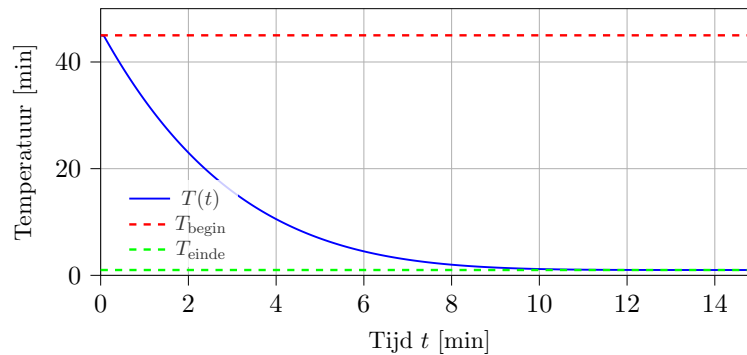


Figuur 5.2: Globaal overzicht van de Service-Maximization Minimax with Fixed-Fleet (SMM-FF) heuristiek

Tijdsgebaseerd koelschema

Als eerste bespreken we de aanpassing op het koelschema. Het SISRs-algoritme bepaalt op voorhand een vast aantal iteraties, de temperatuur daalt exponentieel met een koelingsconstante die gebaseerd is op de eindtemperatuur, begintemperatuur en het aantal iteraties. In ons algoritme willen we als inputparameter een gekozen rekentijd meegeven. We stellen dus een koelschema op dat gebaseerd is op tijd. We stellen hiervoor Formule (5.1) voor, waarin de temperatuur afhankelijk is van de verhouding van de resterende tijd tot de rekentijd, de begintemperatuur en de eindtemperatuur waarbij we nooit boven de begintemperatuur gaan. De begintemperatuur stellen we in op 45 minuten en de eindtemperatuur op 0.02 minuten (10 seconden). Met dit koelschema laten we in het begin verslechtingen toe van maximaal 45 minuten (exploratie), waarna de temperatuur daalt en we minder grote verslechtingen toelaten. Aan het einde zijn bijna geen verslechtingen meer toegelaten (convergentie). In Figuur 5.3 is het koelschema voor een rekentijd van 15 minuten gegeven.

$$T(t) = \min \left\{ T_{\text{begin}}, T_{\text{begin}} \cdot \left(\frac{\text{Totale_rekentijd} - t}{\text{Totale_rekentijd}} \right)^5 + T_{\text{einde}} \right\} \quad (5.1)$$



Figuur 5.3: Voorbeeld van het koelschema voor een totale rekentijd van 15 minuten volgens Formule (5.1). De begintemperatuur bedraagt $T_0 = 45.0$ minuten en de eindtemperatuur $T_f = 0.02$ minuten (10s).

Acceptatiecriteria

Tijdens het uitvoeren van de *local search* metaheuristiek houden we zowel de globaal beste als lokaal beste oplossing bij. Nieuwe iteraties starten altijd met de lokaal beste oplossing.

Wanneer een nieuwe oplossing (na de *ruin*-, *recreate*- en *compress schedule*-stap) beter is dan de globaal beste oplossing, accepteren we deze altijd als nieuwe globaal beste oplossing. De globaal beste oplossing laat nooit verslechtingen toe. We definiëren een oplossing als beter dan een andere oplossing wanneer ze:

1. meer acties bevat, of

2. evenveel acties bevat en na het sorteren van de rittijden in aflopende volgorde, de eerste waarde waarop de twee oplossingen verschillen kleiner is dan die van de huidige beste oplossing.

Deze definitie zorgt ervoor dat we niet enkel de langste rit beschouwen, maar de volledige verdeling van rittijden meenemen. Hierdoor ontstaat meer variatie in de zoekruimte en creëren we extra verbeteringsmogelijkheden.

Daarnaast hanteren we een lokaal beste oplossing die we accepteren met een gelijkaardig acceptatiecriterium als in het SISRs-algoritme. Indien de nieuwe oplossing beter is dan de lokaal beste oplossing, dan aanvaarden we ze altijd als nieuwe lokaal beste. Indien dit niet het geval is, dan aanvaarden we af en toe verslechtingen via het acceptatiecriterium. Hierin vergelijken we de gesorteerde rittijdvectoren element per element. Bij gelijke waarden verlagen we een impactfactor λ met een factor 0.95 zodat verschillen in de staart minder zwaar doorwegen. Bij de eerste afwijking berekenen we een drempel van verslechting die we toelaten op basis van de huidige temperatuur $\mathcal{T}$ en een willekeurige factor $\ln(\mathcal{U}(0, 1))$. Als de nieuwe oplossing binnen deze toegestane verslechting zit, dan accepteren we deze als nieuwe lokaal beste oplossing op voorwaarde dat de nieuwe oplossing minstens evenveel acties bevat. Oplossingen met minder acties in de ritten accepteren we nooit. Pseudocode voor dit acceptatiecriterium kun je vinden in Algoritme 1.

Algoritme 1: Acceptatiecriterium

```

procedure Acceptatiecriterium( $s_{best}, s_{lokaal}, s_{nieuw}$ ):
    if isBeter( $s_{nieuw}, s_{best}$ ) then
         $s_{best} \leftarrow s_{nieuw}$ ;
    geaccepteerd  $\leftarrow$  isBeter( $s_{nieuw}, s_{lokaal}$ );
    if not geaccepteerd then
        if  $|Rittijden(s_{lokaal})| = |Rittijden(s_{nieuw})|$  then
             $\lambda \leftarrow 1.0$ ;
            for  $i = 1$  to  $|Rittijden(s_{nieuw})|$  do
                if  $Rittijden(s_{nieuw})[i] = Rittijden(s_{lokaal})[i]$  then
                     $\lambda \leftarrow 0.95\lambda$ ;
                    continue
                if  $Rittijden(s_{nieuw})[i] < Rittijden(s_{lokaal})[i] - \lambda \cdot \mathcal{T} \cdot \ln(\mathcal{U}(0, 1))$  then
                    geaccepteerd  $\leftarrow$  true
                    break
        if geaccepteerd then
             $s_{lokaal} \leftarrow s_{nieuw}$ 

```

5.4.2 Initiële oplossing

Onze aanpak verschilt van het SISRs-algoritme, waar elke klant initieel een eigen voertuig krijgt. In ons geval vertrekken we vanuit een vooraf gegeven verzameling voertuigen. Om een goede geografische spreiding van de eerste ritten te bekomen, sorteren we de schoollocaties in paren op basis van onderlinge reistijd. Het eerste paar bevat de twee scholen die het verst van elkaar verwijderd zijn, gevolgd door het tweede paar met de volgende grootste reistijd, enzovoort. Op die manier identificeren we de $|K|$ meest afgelegen schoolparen, met $|K|$ gelijk aan het aantal voertuigen waarover we beschikken.

Vervolgens wijzen we per voertuig uit de inputverzameling één actie-paar toe, bestaande uit het ophalen en afzetten van een leerling horend bij de bovenste scholen in deze lijst. Het doel hiervan is dat, wanneer er meerdere verspreide clusters van acties bestaan, elk cluster minstens één voertuig toegewezen krijgt. Daarna roepen we onmiddellijk de *re-create*-fase aan om zoveel mogelijk extra acties in de initiële oplossing op te nemen. Alle resterende acties die we niet konden toevoegen, maken deel uit van de *absent*-verzameling.

5.4.3 Ruin-methode

De ruïn-methode volgt in grote lijnen het SISRs-algoritme, maar bevat enkele belangrijke aanpassingen. Bij het bepalen van het aantal te verwijderen acties delen we dit aantal door twee omdat we bij het verwijderen van een actie ook steeds de bijhorende gekoppelde actie verwijderen. Wanneer we bijvoorbeeld de ophaalactie van u_1 verwijderen, verwijderen we ook de bijhorende afzetactie van u_1 .

De seed-actie is een willekeurig gekozen actie die de startlocatie en omgeving aanduidt waar het ruïneren van de oplossing plaatsvindt. Vanuit deze seed-actie selecteert het algoritme telkens een buur die we samen met enkele opeenvolgende acties uit dezelfde rit (een string) verwijderen. Bij het kiezen van een buur wijken we licht af van de originele aanpak. Het SISRs-algoritme selecteert altijd de geografisch dichtstbijzijnde actie. In onze methode identificeren we eerst de vijf dichtstbijzijnde acties. Indien één van deze vijf acties dezelfde schoollocatie als bestemming heeft als de seed-actie, geven we deze actie met een kans van 50% voorrang op de fysiek dichtstbijzijnde actie. In het andere geval kiezen we de dichtstbijzijnde actie. Deze strategie vergroot de kans dat we meerdere acties van leerlingen met dezelfde schoollocatie uit de oplossing verwijderen, zodat we ze later gezamenlijk opnieuw kunnen invoegen, wat mogelijk leidt tot betere ritten.

Voor het verwijderen van een string van acties hanteren we een probabilistische keuze. Met een kans van 0,01% verwijderen we de volledige rit. In alle andere gevallen kiezen we met 50% kans voor de standaard string-procedure en met 50% kans voor de split-string-procedure, zoals beschreven in het originele werk (Christiaens & Vanden Bergh, 2020).

5.4.4 Recreate-methode

De *recreate*-stap probeert de eerder verwijderde acties opnieuw in de oplossing in te voegen. Dit proces start met het ordenen van de *absent*-verzameling volgens verschillende strategieën, zoals een willekeurige volgorde, afstand tussen *pickup* en *delivery* (zowel van groot naar klein als omgekeerd), servicetijd (van groot naar klein), breedte van het tijdsvenster (van groot naar klein) en starttijd van het tijdsvenster (van vroeg naar laat en omgekeerd). In elke iteratie kiezen we één van deze strategieën om variatie in de reconstructie te garanderen.

Na het sorteren proberen we elk actiepaar opnieuw in te voegen in een voertuig. Voor elk paar evalueren we alle geldige posities in alle voertuigritten voor zowel het ophalen als afzetten. Op basis van deze evaluatie selecteren we uiteindelijk de posities met de laagste invoegkost waar we dan dit actiepaar invoegen. Om dit proces efficiënt te houden, filteren we ongeldige posities zo vroeg mogelijk uit. We voeren onder meer een vroege controle uit op de tijdsvensters. Wanneer het tijdsvenster van de in te voegen actie volledig voor het tijdsvenster van de voorgaande actie ligt, slaan we deze positie en alle daaropvolgende posities onmiddellijk over. Wanneer het tijdsvenster van de in te voegen actie volledig na het tijdsvenster van de daaropvolgende actie ligt, slaan we ook deze positie meteen over.

Vervolgens simuleren we de impact op de voertuigcapaciteit. Het ophalen verhoogt de bezetting tussen de ophaal- en afzetactie met één. Indien deze verhoging ertoe leidt dat de rittenplanning niet meer uitvoerbaar is met de beschikbare vloot, sluiten we deze posities uit.

Hierna simuleren we de volledige invoeging van zowel de ophaal- als de afzetactie in de planning van de rit. De simulatie maakt gebruik van een methode die sterk lijkt op de insertion-stap van Vansteenwegen et al. (2009), met één belangrijk verschil: in ons geval moet niet enkel de start van de service maar de volledige servicetijd binnen het tijdsvenster vallen. Tijdens deze simulatie controleren we of er voldoende ruimte is binnen het tijdsvenster van beide acties en of de tijdsvensters van alle andere acties gerespecteerd kunnen blijven. Hiervoor berekenen we de verschuiving (*shift*) die de invoeging veroorzaakt in het schema. Deze verschuiving dient niet alleen om de geldigheid van de invoeging te verifiëren, maar gebruiken we ook als de invoegkost van deze posities. Indien een voertuig momenteel geen acties bevat, stellen we de invoegkost gelijk aan nul, zodat we lege voertuigen altijd opnieuw gebruiken.

Tot slot voegt het algoritme het actiepaar in op de posities met de laagste invoegkost binnen de bijhorende rit. Dit herhalen we voor alle actieparen in de *absent*-verzameling. Indien we geen geldige positie vinden, laten we deze actie in de *absent*-verzameling staan.

In deze *recreate*-stap optimaliseren we dus niet rechtstreeks op rittijden, maar op de tijdsverlenging die ontstaat wanneer een voertuig een extra actiepaar moet invoegen. We gaan ervan uit dat efficiënte, compacte ritten doorgaans ook leiden tot betere rittijden. Bij de acceptatie van oplossingen in de metaheuristiek gebruiken we, zoals eerder vermeld, wel de rittijden als objectief. De pseudocode van de *recreate*-methode is opgenomen in Algoritme 4 in Appendix C.

5.4.5 Compress Schedule methode

Na de *recreate*-fase beschikken we over een planning waarin elke actie zo vroeg mogelijk is ingepland. Dit levert echter niet de kortste rittijden op. Om dit te verbeteren, breiden we de klassieke *ruin-recreate*-aanpak uit met een aanvullende *compress schedule*-methode. Het doel van deze methode is om de planning samen te drukken, zodat de totale tijd tussen de eerste en laatste actie minimaal is en, bij voorkeur, ook de individuele rittijden verkorten.

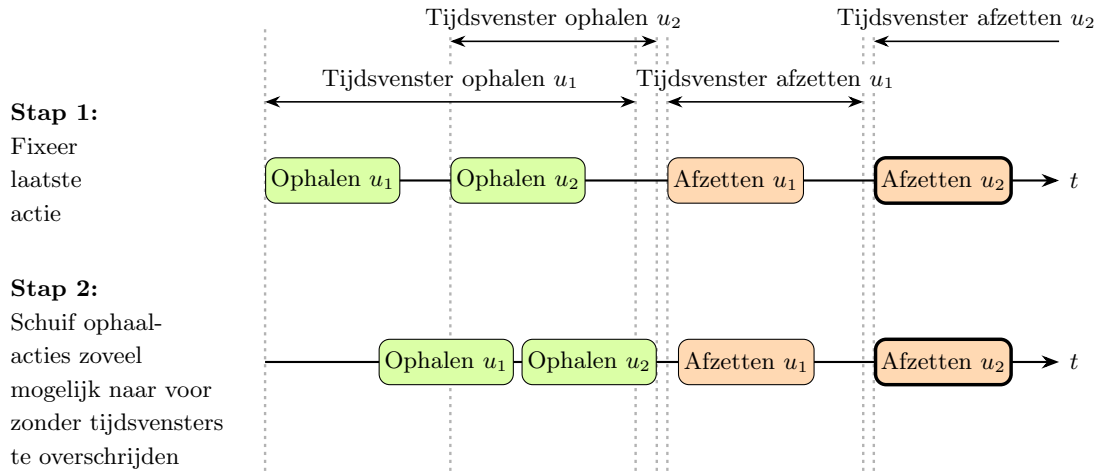
Onze aanpak voor het bijhouden van de planning is gebaseerd op het algoritme van Vansteenwegen et al. (2009). We benutten de variabelen aankomsttijd, starttijd van de service, wachttijd en maximale verschuiving om de planning samen te drukken.

Het samendrukken van de planning start bij de laatste actie in de rit. Deze actie is reeds zo vroeg mogelijk ingepland en willen we niet later uitstellen. Daarom zetten we zijn maximale verschuiving op nul en propageren deze aanpassing naar alle voorgaande acties.

Daarna doorlopen we de rit van voor naar achter. Elke ophaalactie stellen we zoveel mogelijk uit binnen de toegestane maximale verschuiving. Afzetacties schuiven mee indien nodig maar verplaatsen we niet actief. Het resultaat is een planning waarin de laatste afzetactie vaststaat en alle ophaalacties maximaal uitgesteld zijn.

Na deze compressie berekent het algoritme de rittijd per leerling als het verschil tussen het tijdstip net na het instappen bij de ophaalactie (starttijd van de ophaalactie verhoogd met de servicetijd) en het tijdstip net na het afzetten op school (starttijd van de afzetactie verhoogd met de servicetijd). We verwachten dat deze aanpak efficiënt en effectief is, vooral bij kleine tijdsvensters zoals de tijdsvensters van scholen. Bij grotere tijdsvensters is het effect minder voorspelbaar: het verschuiven van een ophaalactie kan ertoe leiden dat een afzetactie van een eerder opgehaalde leerling aanzienlijk later plaatsvindt, waardoor diens rittijd toeneemt. Merk op dat ook de oorspronkelijke, niet-samengedrukte planning behouden blijft voor de volgende iteratie.

Figuur 5.4 toont een eenvoudig voorbeeld voor één voertuig dat twee actieparen uitvoert.



Figuur 5.4: Visualisatie van de *compress schedule*-methode voor één voertuig die twee actie-paren uitvoert

5.5 Varianten op SMM-FF

5.5.1 Service-Maximization Minimax with Temporary-Fleet-Expansion (SMM-TFE)

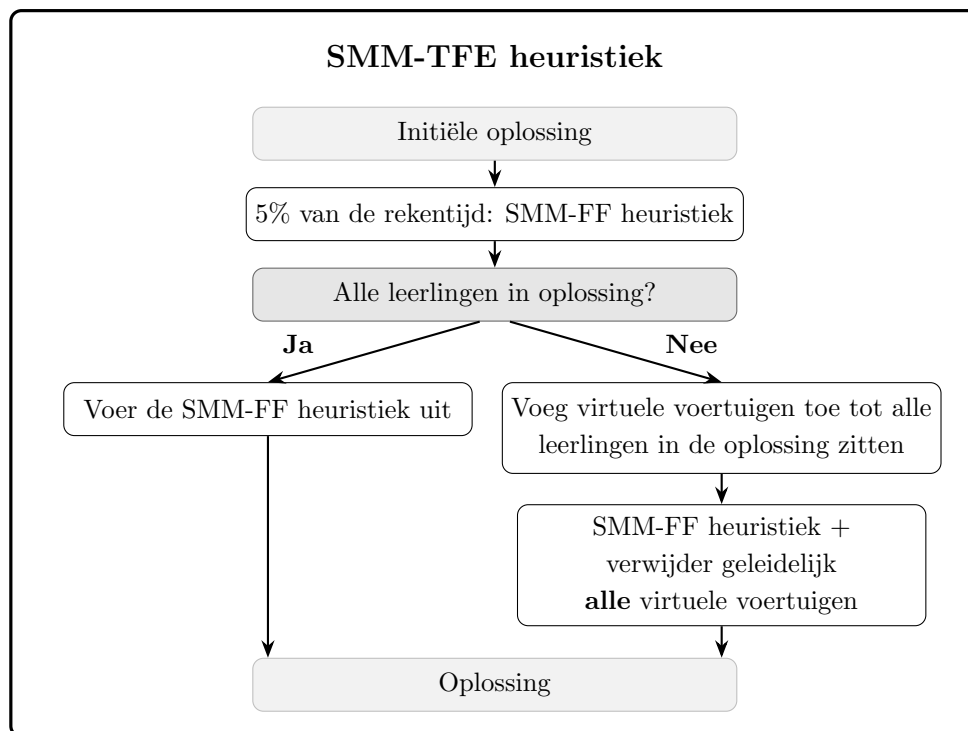
Bij moeilijk oplosbare instanties kan het veel tijd kosten om een eerste geldige oplossing te construeren waarin alle acties zijn ingevoegd. In deze situatie blijven veel acties lange tijd in de *absent*-verzameling. Tijdens de *recreate*-fase proberen we systematisch alle *absent*-acties in te voegen, maar wanneer die verzameling groot blijft neemt de duur van de *recreate*-fase toe en daalt het aantal uitvoerbare iteraties binnen de tijdslimiet. Dit kan nadelig zijn voor de kwaliteit van de eindoplossing.

Het doel van Service-Maximization Minimax with Temporary-Fleet-Expansion (SMM-TFE) is om de SMM-FF heuristiek te versnellen, zonder het einddoel te wijzigen: we minimaliseren de langste rittijd, halen zoveel mogelijk leerlingen op en leveren uiteindelijk een oplossing zonder extra voertuigen.

Figuur 5.5 geeft een overzicht van de volledige flow van dit algoritme. Na een korte initiële fase (5% van de rekentijd) controleren we of alle acties zijn ingevoegd. Indien dit niet het geval is, dan voegen we tijdelijk virtuele voertuigen met capaciteit één toe zodat de oplossing onmiddellijk geldig is en alle acties zijn ingevoegd. Dit elimineert de *absent*-verzameling en kan de *recreate*-fase versnellen. Vervolgens verdelen we de resterende tijd in gelijke delen; één voor elk mogelijk aantal virtuele voertuigen en een deel met nul virtuele voertuigen. In elk deel voeren we de heuristiek uit met het bijhorende aantal voertuigen en verwijderen we bij het verstrijken van het tijdsdeel één virtueel voertuig. Dit herhalen we tot er geen virtuele voertuigen meer overblijven. In het geval dat alle acties uit *absent* zijn

ingevoegd voor het einde van een tijdsdeel, schakelen we met een kans van 5% vervroegd over op de volgende verzameling virtuele voertuigen. De resterende tijd voegen we ook toe aan dat deel. Zo eindigen we altijd binnen de tijdslimiet met een oplossing zonder virtuele voertuigen. Het is echter belangrijk dat er voldoende iteraties beschikbaar zijn om alle virtuele voertuigen te verwijderen. Het aantal resterende iteraties moet dus groter zijn dan het aantal toegevoegde virtuele voertuigen.

Door tijdelijk virtuele voertuigen toe te voegen, verkleinen we de *absent*-verzameling en verwachten we dat de iteraties daardoor sneller verlopen. Een mogelijk nadeel is dat we bij een groot aantal virtuele voertuigen deze ritten ook moeten verwerken tijdens de *recreate*-fase, wat het algoritme opnieuw kan vertragen.



Figuur 5.5: Globaal overzicht van Service-Maximization Minimax with Temporary-Fleet-Expansion (SMM-TFE) heuristiek

5.5.2 Full-Service Minimax with Fleet-Expansion (FSM-FE)

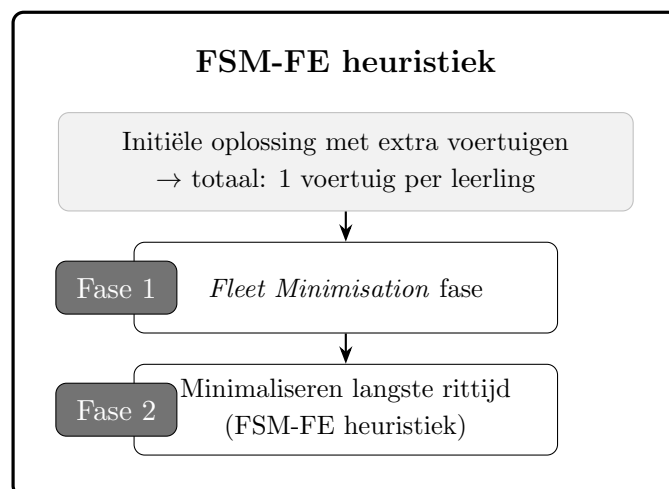
In sommige gevallen lukt het niet om alle leerlingen in de oplossing op te nemen. We introduceren daarom een flexibelere variant van het probleem, waarbij het doel is om alle leerlingen altijd op te halen, desnoods met het gebruik van één of meerdere extra voertuigen. De doelstelling blijft nog steeds om de langste rittijd te minimaliseren.

Deze aanpak is geïnspireerd op de fleet-minimisationstrategie van Christiaens en Vanden Berghe (2020) die in twee fasen verloopt. In de eerste fase, die 10% van de iteraties

in beslag neemt, probeert dit algoritme het aantal voertuigen te minimaliseren. Daarna volgt de heuristiek voor het minimaliseren van de totale afstand. De procedure start met één voertuig per klant en verwijdert vervolgens iteratief voertuigen, waarbij ze telkens het voertuig met de laagste *absent sum* verwijdert. Dit is het voertuig dat de meeste klanten bevat die makkelijkst in te voegen zijn.

Wij nemen deze methode grotendeels over, maar met enkele kleine aanpassingen. Als initiële oplossing gebruiken we de oorspronkelijke vloot uit de inputverzameling, aangevuld met extra voertuigen zodat elke leerling initieel één voertuig toegewezen krijgt. Tijdens de minimalisatiefase willen we de oorspronkelijke vloot behouden. Wanneer we een voertuig verwijderen, maken we de rit van dat voertuig leeg, maar behouden het voertuig zelf. Vervolgens verplaatsen we de rit van een extra toegevoegd voertuig naar dit lege voertuig en verwijderen het extra voertuig. Om dit mogelijk te maken, krijgen de extra voertuigen een capaciteit die gelijk is aan de minimale capaciteit van de oorspronkelijke vloot, zodat de ritten van deze extra voertuigen in elk ander voertuig passen.

Aangezien we werken met een tijdslimiet in plaats van een vast aantal iteraties, besteden we 10% van de totale rekentijd aan de minimalisatiefase. Daarna schakelen we over naar de SMM-FF, zoals beschreven in Sectie 5.4.



Figuur 5.6: Globaal overzicht van Full-Service Minimax with Fleet-Expansion (FSM-FE) heuristiek

5.6 Parameters

Alle parameters die we konden overnemen uit het werk van Christiaens en Vanden Berghe (2020), hebben we behouden. Daarnaast is onze initiële temperatuur ingesteld op 45 minuten en de eindtemperatuur op 0.02 minuten (10 seconden). De keuze van de rekentijd lichten we verder toe in Hoofdstuk 7.

Daarnaast biedt onze implementatie, net zoals bij de exacte aanpak, flexibiliteit om verschillende parameters aan te passen. Zo kan men de tijdsvensters verlengen of een eigen voertuigvloot specificeren. De tijdseenheid is intern opnieuw ingesteld op 10 seconden om consistentie met de exacte methode te behouden. Tot slot hanteren we de regel die stelt dat we leerlingen ten vroegste 90 minuten voor het begin van het tijdsvenster van hun school mogen ophalen.

Hoofdstuk 6

Generatie van realistische dataset

6.1 Inleiding

Er is geen bestaande dataset beschikbaar die alle relevante elementen bevat om het probleem van schoolvervoer in Vlaanderen en Brussel realistisch te modelleren. Zonder een dergelijke dataset is het niet mogelijk om algoritmen te testen of oplossingen te evalueren.

Om dit probleem op te lossen, stellen we voor om een synthetische dataset te ontwikkelen die de situatie in Vlaanderen en Brussel zo goed mogelijk benadert. Hiervoor combineren we beschikbare gegevens over scholen en busvervoer in het buitengewoon onderwijs met enkele veronderstellingen om ontbrekende informatie aan te vullen. De elementen van deze dataset zijn lijsten van scholen, leerlingen, parkings en bussen. Daarnaast bepalen we de reistijden tussen alle relevante locaties en slaan we deze op in een matrix.

Het resultaat is een verzameling van vijf datasets: drie voor het schooljaar 2024-2025 en twee voor het schooljaar 2022-2023, met respectievelijke teldatums op 1 februari 2023 en 1 februari 2025. In het vervolg verwijzen we naar deze als de dataset van 2023 en die van 2025. Daarnaast genereren we ook kleinere deelverzamelingen om het oplossen van deelproblemen en het testen van algoritmen mogelijk te maken.

In de volgende secties lichten we toe hoe deze gegevens zijn verzameld, verwerkt en gestructureerd tot een bruikbare en realistische dataset. Eerst bespreken we in Sectie 6.2 de scholen en hun tijdsvensters. Vervolgens behandelen we in Sectie 6.3 de leerlingen en hun servicetijden. Daarna komt in Sectie 6.4 de beschrijving van de eventuele beschikbare parkings aan bod. In Sectie 6.5 leggen we uit hoe de tijdsmatrix is opgebouwd, terwijl Sectie 6.6 de gegenereerde busvloot bespreekt. Tot slot presenteren we in Sectie 6.7 het eindresultaat en de gegenereerde deelverzamelingen.

6.2 Scholen

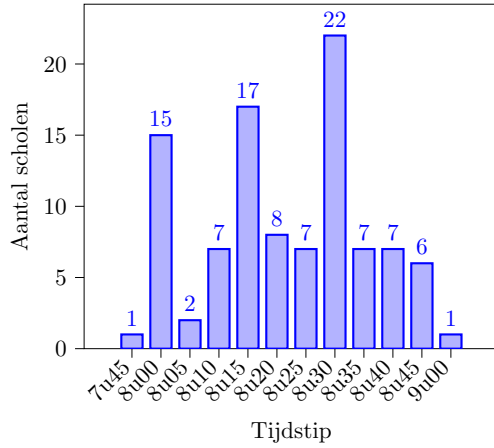
Om deze dataset samen te stellen, starten we met het verzamelen van alle scholen. Elke school in de dataset heeft een adres en een tijdsvenster waarbinnen leerlingen moeten aankomen.

De eerste stap is het identificeren van alle schoollocaties voor buitengewoon onderwijs. We halen een lijst van schoolvestigingen voor 2023 en 2025 op uit Dataloep, inclusief naam en adres (Vlaams ministerie van onderwijs en vorming, s.d.). Deze adressen geocoderen we vervolgens naar coördinaten met behulp van Nominatim (s.d.), om ze bruikbaar te maken voor routeberekeningen. In 2023 bevat de dataset 450 schoolvestigingen en in 2025 stijgt dit aantal naar 469 (Vlaams ministerie van onderwijs en vorming, s.d.).

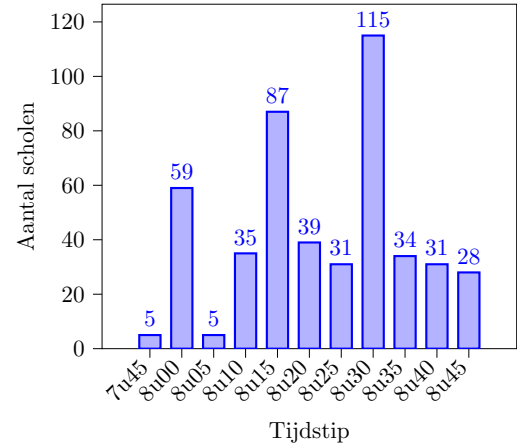
Daarnaast hebben we de openingsuren en starttijden voor honderd scholen handmatig opgezocht. Tussen het openen van de school en de start van de lessen is er toezicht aanwezig. Deze periode komt dan ook overeen met het tijdsvenster waarin leerlingen mogen toekomen op deze school. De tijdsvensters van de opgezochte scholen dienen als referentie om realistische tijdsvensters voor alle scholen in de dataset te genereren.

Voor deze generatie is een kandidatenlijst samengesteld met alle start- en eindtijden van de tijdsvensters uit de opgezochte scholen. Duplicaten blijven behouden, zodat tijdsvensters die vaker voorkomen in de opgezochte verzameling een grotere kans hebben om gekozen te worden. Elke school in de synthetische dataset krijgt vervolgens willekeurig een tijdsvenster toegewezen, geselecteerd uit deze lijst. De verdeling van tijdsintervallen in de gegenereerde dataset is dus zeer gelijkaardig aan die van de opgezochte scholen.

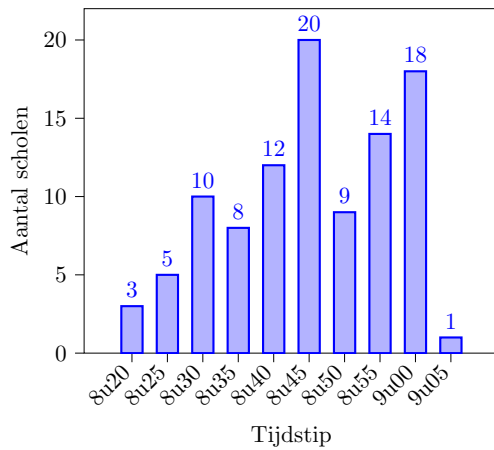
Figuur 6.1 toont links de verdeling van tijdsvensters bij de opgezochte scholen en rechts een voorbeeld van de verdeling van de gegenereerde tijdsintervallen voor alle scholen in een dataset van 2025.



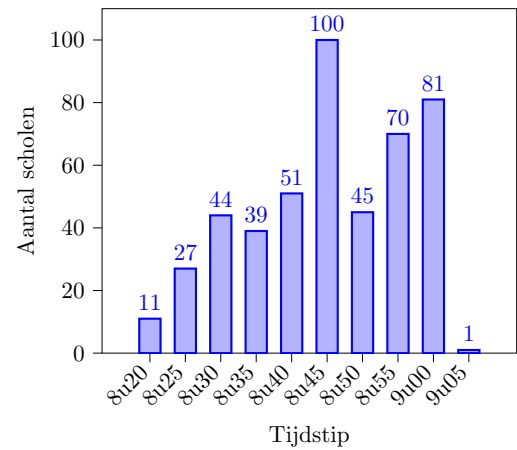
(a) Opgezochte starttijden tijdsvenster voor een deelverzameling van de scholen



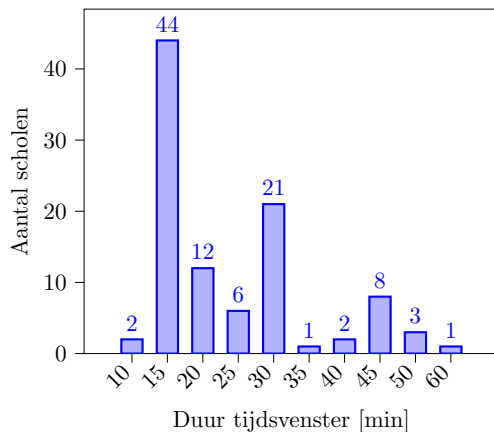
(b) Gegenerateerde starttijden tijdsvenster



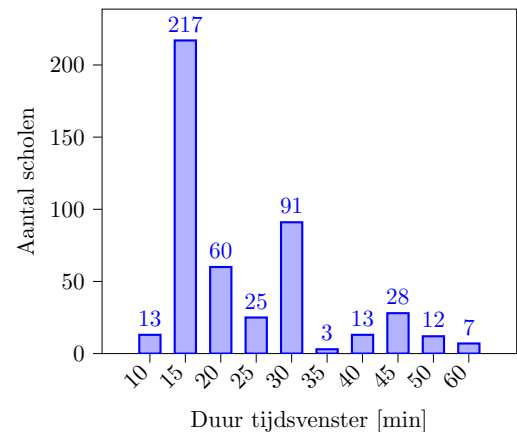
(c) Opgezochte eindtijden tijdsvenster voor een deelverzameling van de scholen



(d) Gegenerateerde eindtijden tijdsvenster



(e) Opgezochte intervalgroottes voor een deelverzameling van de scholen



(f) Gegenerateerde intervalgroottes

Figuur 6.1: Vergelijking tussen handmatig verzamelde tijdsvensters van 100 Vlaamse en Brusselse scholen voor buitengewoon onderwijs en een voorbeeld van gegenereerde tijdsvensters voor alle scholen. De gegenereerde tijdsvensters zijn willekeurig gekozen volgens dezelfde kansverdeling als die van de handmatig verzamelde data. De data voor de handmatig verzamelde tijdsvensters is samengesteld uit verschillende bronnen (zie Appendix E).

6.3 Leerlingen

Vervolgens genereren we een lijst fictieve leerlingen. Elke leerling heeft een thuisadres en is gekoppeld aan een specifieke school. Deze generatie bespreken we in Sectie 6.3.1. Daarnaast krijgt elke leerling een servicetijd zoals verder toegelicht in Sectie 6.3.2.

6.3.1 Leerlingen en hun locaties

Voor elke schoolvestiging gaan we, op basis van gegevens uit Dataloep van het Vlaams ministerie van onderwijs en vorming (s.d.), na hoeveel leerlingen van deze vestiging per Vlaamse en Brusselse gemeente wonen.

Om adressen toe te wijzen aan de leerlingen, maken we gebruik van volledige adressenlijsten van Vlaanderen en Brussel (agentschap Digitaal Vlaanderen - Helpdesk Digitaal Vlaanderen, 2023; Paradigm, 2022). De adressen delen we op in een adreslijst per gemeente. Op die manier kunnen we per schoolvestiging willekeurige adressen selecteren op basis van het aantal leerlingen dat in die gemeente woont. Elk adres levert één leerling op. De leerling krijgt daarbij een uniek ID, het geselecteerde adres en de bijhorende school.

Vervolgens streven we ernaar om de langste rittijd onder de 90 minuten te houden. Alle leerlingen, waarvoor de rechtstreekse reistijd naar de schoollocatie deze langste rittijd overschrijdt, zijn daarom uitgesloten. Dit is in overeenstemming met de regelgeving die stelt dat leerlingen de dichtstbijzijnde schoolvestiging moeten kiezen die hun opleiding aanbiedt (Vlaams ministerie van onderwijs en vorming & AGODI - Agentschap voor Onderwijsdiensten, 2024). We gaan ervan uit dat de leerlingen die verder dan 90 minuten van hun school wonen, niet de dichtstbijzijnde school hebben gekozen en bijgevolg dus geen recht te hebben op gratis leerlingenvervoer. Om die reden is het gerechtvaardigd om deze leerlingen niet op te nemen in de dataset.

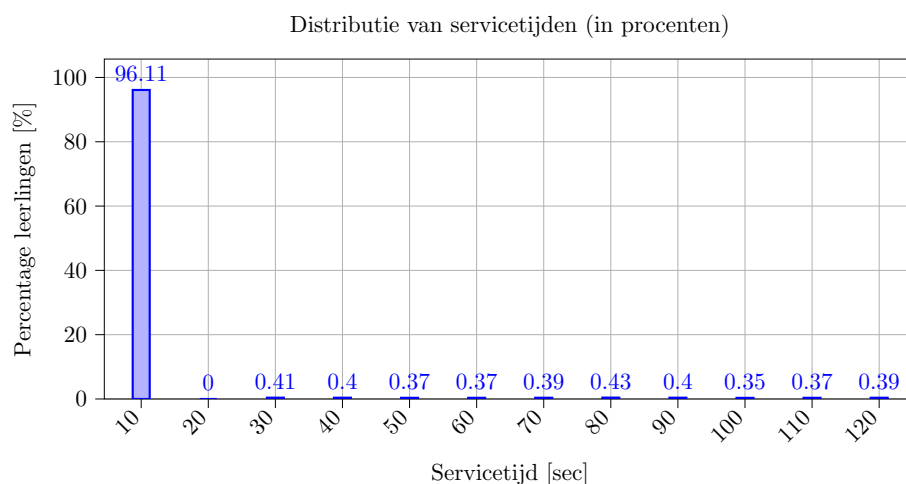
Op dit moment beschikken we over een lijst van alle leerlingen per schoolvestiging die vermoedelijk recht hebben op busvervoer. Uiteraard maakt niet elke leerling effectief gebruik van dit vervoer. Daarom maken we in een volgende stap een selectie van leerlingen die daadwerkelijk met de bus reizen. Volgens beschikbare cijfers maakten 39 981 leerlingen uit het buitengewoon onderwijs gebruik van busvervoer in 2023 (Rekenhof, 2024). In 2025 stijgt dit aantal naar 44 886 leerlingen (Factcheckers, 2025). Uit de volledige verzameling van fictieve leerlingen is vervolgens een willekeurige deelverzameling geselecteerd die exact overeenkomt met deze aantallen, respectievelijk voor 2023 en 2025.

De adressen van deze fictieve leerlingen krijgen coördinaten via Nominatim (s.d.), een geocoder die lokaal of online uitgevoerd kan worden. Indien een adres niet geocodeerbaar is (bijvoorbeeld door afkortingen zoals O.L.V. in plaats van Onze-Lieve-Vrouw), komt een alternatief adres uit dezelfde gemeente in de plaats. Dit komt voor bij ongeveer 1% van de adressen. Dit proces herhaalt zich tot alle adressen succesvol zijn gegeocodeerd.

6.3.2 Servicetijden

Elke leerling in de dataset krijgt een servicetijd toegekend, deze komt overeen met de tijd die nodig is om in of uit te stappen. We veronderstellen dat deze tijd voor in- en uitstappen gelijk is. Volgens statistieken uit 2023 en 2025 heeft ongeveer 4% van de leerlingen in het buitengewoon onderwijs een type 4-beperking, wat wijst op een fysieke beperking (Onderwijskiezer, s.d.; Vlaams ministerie van onderwijs en vorming, s.d.). In de dataset houden we rekening met dit percentage om te bepalen aan hoeveel leerlingen we een langere servicetijd toekennen. Elke leerling krijgt met 96% kans een korte servicetijd en met 4% kans een langere servicetijd. De korte servicetijd is een vaste waarde van 10 seconden. De leerlingen met langere servicetijd krijgen een willekeurige tijdsduur, gekozen uit veelvouden van tien tussen 30 en 120. Deze tijdsduren zijn het resultaat van een weloverwogen inschatting op basis van praktijkobservaties en literatuur, met als doel de realistische servicetijden te benaderen. Zo zagen we uit eigen observaties van een bus-halte van het regulier personenvervoer dat de uitstaptijd gemiddeld één tot twee seconden per persoon bedraagt. Dit wordt bevestigd door de publicatie van Lee (2014), waar de gemiddelde uitstaptijd per passagier varieert van 1.26 tot 1.78 seconden en de gemiddelde instaptijd varieert tussen de 1.48 tot 1.78 seconden per passagier. Deze cijfers houden echter geen rekening met het openen en sluiten van deuren en zijn gebaseerd op regulier vervoer. Omdat in het buitengewoon onderwijs het instappen en uitstappen naar verwachting minder vlot verloopt, hanteren we een minimum van 10 seconden en een hogere bovengrens om realistische omstandigheden, zoals extra begeleiding om in te stappen, te benaderen.

Figuur 6.2 toont de verdeling van de servicetijden in procenten voor een dataset gegenereerd voor het jaartal 2025.



Figuur 6.2: Distributie van servicetijden in procenten voor een dataset uit 2025. Elke leerling kreeg een willekeurige servicetijd, met 96% kans op een korte tijd van 10 seconden en 4% kans op een langere tijd tussen 30 en 120 seconden.

6.4 Parkings

Voor het geval we transferpunten verder zouden onderzoeken, hebben we alvast potentiële locaties verzameld waar bussen kunnen stoppen om leerlingen uit te wisselen. Deze transferpunten bestaan uit een verzameling parkings waar bussen kunnen stoppen om leerlingen uit te wisselen. De parkings zijn op twee manieren verzameld.

Eerst maken we gebruik van een bestaande dataset met busparkings langs gewest- en autosnelwegen (Agentschap Wegen en Verkeer, 2020). Deze locaties zijn specifiek ingericht voor bussen en dus geschikt als tussenstops of transferpunten. De parkings in deze dataset zijn echter beperkt en liggen langs grote wegen.

Daarom vullen we de verzameling aan met manueel verzamelde parkings via Google Maps (Google, 2025). Hierbij zoeken we op termen zoals busparking en parking, waarna we visueel beoordelen of de locatie geschikt is voor bussen. Dit kan bijvoorbeeld door na te gaan of er voldoende manoeuvreerruimte is en of er busplaatsen aanwezig zijn.

Beide bronnen zijn gecombineerd om tot een representatieve verzameling van mogelijke transferpunten te komen.

6.5 Tijdsmatrix

Het berekenen van de reistijden tussen locaties start met een geocoderingsstap. Alle relevante locaties zijn met behulp van Nominatim (s.d.) omgezet naar coördinaten. Vervolgens kozen we om de OSRM-backend van Luxen en Vetter (2011), gevoed met OpenStreetMap-data (s.d.) van België, te gebruiken om de rijtijd tussen elk paar punten te bepalen.

Bij het opstellen van de tijdsmatrix passen we enkele vereenvoudigingen en filters toe om de rekentijd en de grootte van de matrix te beperken. De berekening houdt geen rekening met de wettelijke maximumsnelheid van 90 km/u voor bussen. Deze veronderstelling is aanvaardbaar omdat de meeste ritten over lokale wegen verlopen, waar de snelheidslimieten doorgaans lager zijn dan 90 km/u. Bovendien voegen we enkel geldige bogen toe bij het genereren van de tijdsmatrix. Zo zijn er bijvoorbeeld geen ritten opgenomen tussen parkings onderling, aangezien we veronderstellen dat een leerling maximaal op één parking mag passeren.

Deze aanpak zorgt ervoor dat de tijdsmatrix enkel relevante en bruikbare reistijden bevat voor het optimalisatieprobleem.

6.6 Bussen

De busvloot voor het leerlingenvervoer is een heterogene verzameling van bussen met verschillende capaciteiten. Voor het opstellen van deze verzameling bepalen we zowel het aantal bussen als hun capaciteit.

Het aantal bussen is gebaseerd op cijfers voor de schooljaren 2022–2023 en 2024–2025. In 2023 zijn er 2118 bussen beschikbaar (Rekenhof, 2024). Dit aantal steeg in 2025 tot 2148 (FBAA, 2025). Deze aantallen zijn rechtstreeks overgenomen in de respectievelijke datasets voor 2023 en 2025.

Vervolgens leggen we een minimum- en maximumcapaciteit vast voor de bussen die we genereren. Elke bus beschikt minimaal over zeven zitplaatsen, zoals vastgelegd in de regelgeving (Vlaams ministerie van onderwijs en vorming & AGODI - Agentschap voor Onderwijsdiensten, 2024). Voor de maximumcapaciteit is een bovengrens van vijftig plaatsen gekozen. Hoewel sommige maatschappijen bussen aanbieden met meer dan zestig zitplaatsen, kozen we vijftig om een realistisch scenario met rolstoelplaatsen te benaderen. Deze keuze wordt ondersteund door verschillende bronnen. Vermeeren Travel (s.d.) biedt bussen aan met acht tot vijftig zitplaatsen. Coach Partners (s.d.-a, s.d.-b) vermelden een vloot met een bereik van veertien tot drieënzestig plaatsen te hebben, met een maximum van vierenvijftig voor rolstoeltoegankelijke bussen. Volgens Keolis (s.d.) variëren hun bussen van acht tot zestig plaatsen, waarbij liftbussen (toegankelijk voor rolstoelen) tot zesenvijftig gewone zitplaatsen of acht rolstoelplaatsen bevatten.

Naast het aantal bussen is ook het aantal leerlingen dat gebruikmaakt van het vervoer relevant. In 2023 zijn dat er 39 981, en in 2025 steeg dit aantal tot 44 886 (Factcheckers, 2025; Rekenhof, 2024). Het gemiddelde aantal leerlingen per bus bedraagt 18.88 in 2023 en 20.90 in 2025. Deze waarden geven een indicatie van de bezettingsgraad en de benodigde busgrootte.

Elke bus krijgt daarna een willekeurige capaciteit toegewezen met een kansverdeling volgens een normale verdeling binnen het interval van zeven tot vijftig plaatsen. Het gemiddelde van deze verdeling berekenen we met Formule (6.1) en bedraagt 24.78 plaatsen. We delen niet door 2 (wat het exacte midden van het interval zou opleveren), maar door 2.3 om het gemiddelde bewust iets lager te leggen. Deze keuze voor 2.3 heeft twee redenen. Ten eerste willen we dat de gemiddelde buscapaciteit hoger ligt dan de gemiddelde bezettingsgraad, in de veronderstelling dat bussen zelden volledig gevuld zijn. Ten tweede veronderstellen we dat er meer bussen zijn met een capaciteit van ongeveer 25 plaatsen dan 28 plaatsen (het midden van het interval).

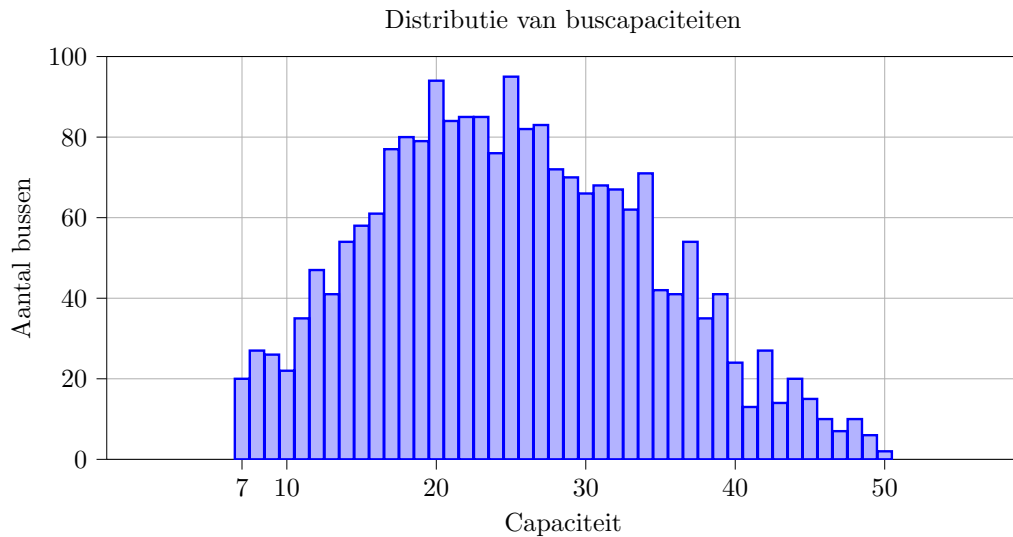
$$\text{Gemiddelde waarde voor verdeling bussen} = \frac{(\text{Minimum capaciteit} + \text{Maximum capaciteit})}{2.3} \quad (6.1)$$

De standaarddeviatie bepaalt hoe sterk de capaciteiten van de bussen rond het gemiddelde variëren (Wiskunde.net, 2026). Een grote standaarddeviatie zou betekenen dat er veel extreme waarden voorkomen (heel kleine of heel grote bussen), terwijl een kleine standaarddeviatie zorgt voor een meer geconcentreerde verdeling rond het gemiddelde. Om

een realistische spreiding te bekomen, nemen we een deel van het totale bereik tussen de minimum- en maximumcapaciteit. Concreet kiezen we één vierde van dit bereik zoals weergegeven in Formule (6.2), omdat dit variatie toelaat zonder dat extreme waarden te frequent aanwezig zijn. Voor een interval van 7 tot 50 plaatsen geeft dit een standaarddeviatie van 10,75. Dit betekent dat de meeste bussen een capaciteit hebben die binnen ongeveer 10 plaatsen van het gemiddelde ligt, terwijl heel kleine of heel grote bussen slechts zelden voorkomen.

$$\text{Standaarddeviatie voor verdeling bussen} = \frac{(\text{Maximum capaciteit} - \text{Minimum capaciteit})}{4} \quad (6.2)$$

Figuur 6.3 toont een voorbeeld van de verdeling van buscapaciteiten in een dataset voor het jaar 2025.



Figuur 6.3: Voorbeelddistributie capaciteiten van de gegenereerde bussen voor het jaartal 2025. De capaciteiten zijn willekeurig gekozen volgens een normale verdeling met een gemiddelde van 24.78 en een standaardafwijking van 10.75. De waarden zijn beperkt tot het interval $[7, 50]$, waarbij we de waarden buiten het bereik niet behouden en opnieuw genereren.

6.7 Resultaat en deelverzamelingen

Alle voorgaande stappen resulteren in enkele uitgebreide en realistische datasets voor Vlaanderen en Brussel. Elke dataset bestaat uit een verzameling van leerlingen, scholen, parkings, bussen en een tijdsmatrix.

In totaal genereren we vijf datasets: drie voor het jaar 2025 en twee voor 2023. Binnen elk jaartal blijft de gekende data identiek, terwijl de willekeurige componenten verschillen

tussen de datasets. Het aantal leerlingen per gemeente en per schoolvestiging blijft bijvoorbeeld gelijk, maar de specifieke adressen binnen de gemeente en de selectie van wie effectief de bus neemt, variëren. Het genereren van meerdere varianten maakt het mogelijk om de robuustheid van algoritmen te evalueren.

Aangezien het oplossen van het volledige probleem voor Vlaanderen en Brussel computationeel zeer intensief is, creëren we ook kleinere deelverzamelingen van de datasets die eenvoudiger te verwerken zijn.

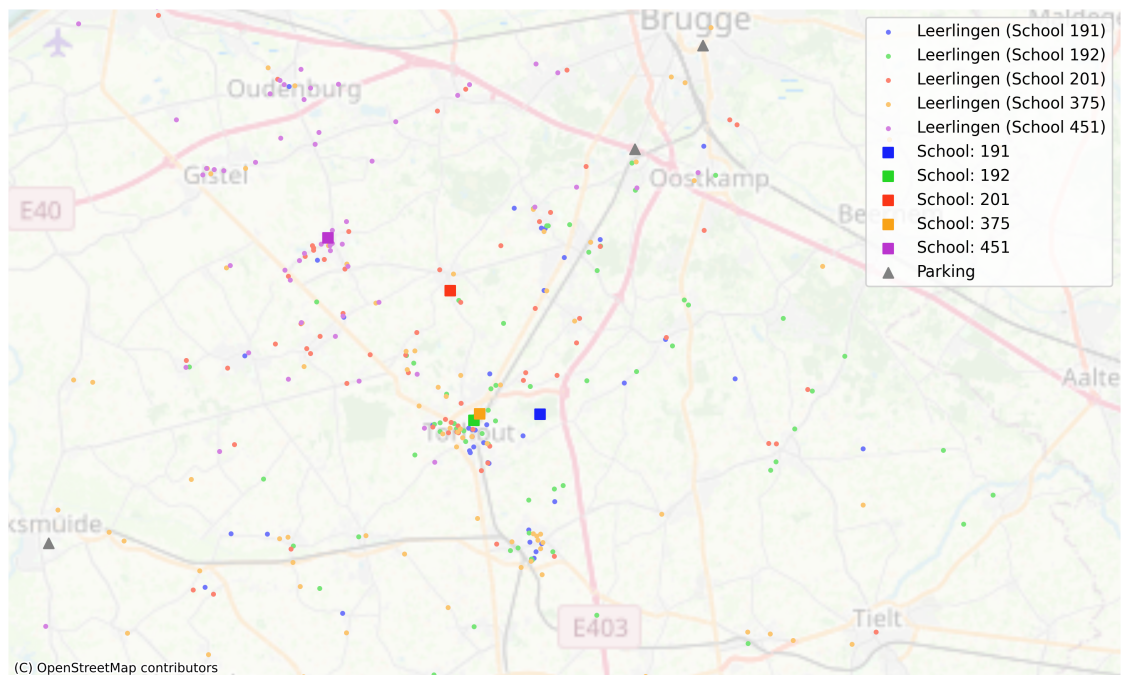
Het genereren van een deelverzameling start met het selecteren van bijvoorbeeld een groep van gemeentes. De deelverzameling omvat dan alle scholen die binnen deze gemeentes gelegen zijn, alle leerlingen die horen bij deze scholen (ongeacht hun woonplaats) en de eventueel de relevante transferpunten in de omgeving. Op dezelfde manier kunnen we ook deelverzamelingen samenstellen voor een volledige provincie, per arrondissement, per gemeente of per school.

Formule (6.3) bepaalt het aantal bussen in een deelverzameling. Daarbij blijft de verhouding van het aantal leerlingen ten opzichte van het aantal bussen gelijk in de overgang van een volledige dataset naar een deelverzameling. In de experimenten van Hoofdstuk 7 zorgen we er bovendien voor dat wanneer we een grotere verzameling in kleinere verzamelingen opsplitsen, het samenvoegen van de bussen met hun capaciteiten uit deze kleinere verzamelingen opnieuw resulteert in de oorspronkelijke busvloot van de grotere verzameling. De bussen zullen dus proportioneel verdeeld zijn.

$$\text{Aantal bussen} = \max \left(1, \left\lfloor \frac{\text{Aantal leerlingen} \times \text{Totaal aantal bussen}}{\text{Totaal aantal leerlingen}} \right\rfloor \right) \quad (6.3)$$

Figuur 6.4 toont een voorbeeld van een deelverzameling voor de gemeentes Torhout, Zedelgem en Ichtegem.

Tot slot hebben we ook zeer kleine inputverzamelingen nodig om het wiskundig model te testen. Deze stellen we samen door eerst een deelverzameling van een groep gemeenten te nemen en daaruit opnieuw een kleinere verzameling te selecteren met een vooraf bepaald aantal leerlingen, scholen en bussen.



Figuur 6.4: Voorbeeld van een gegenereerde kaart met scholen (vierkanten) in Torhout, Zedelgem en Ichtegem, samen met hun bijhorende leerlingen (cirkels) en omliggende parkings (driehoeken)

Hoofdstuk 7

Experimenten en resultaten

7.1 Inleiding

In dit hoofdstuk bespreken we de resultaten van onze experimenten. Alle experimenten voeren we vijf keer onafhankelijk uit op een computer uitgerust met een AMD Ryzen 9 5950X 16-core processor (3.4 GHz) en Windows 11 als besturingssysteem. Het algoritme is geïmplementeerd in Rust (1.92.0). De gebruikte instanties zijn deelverzamelingen van, of komen overeen met, een volledige dataset uit 2023 zoals opgebouwd in Hoofdstuk 6. We maken een onderscheid tussen technische experimenten en praktische analyses die beleidsbeslissingen kunnen ondersteunen.

Eerst evalueren we de voorgestelde heuristische methodologie op kleine instanties. Dit zijn de instanties waarmee we de exacte aanpak evalueerden in Sectie 4.6. Het doel is om na te gaan of onze heuristiek in staat is oplossingen te vinden die de kwaliteit van de optimale oplossingen van de exacte methode benaderen. Vervolgens onderzoeken we de prestaties van de Service-Maximization Minimax with Fixed-Fleet (SMM-FF) heuristiek en haar varianten voor enkele willekeurig geselecteerde grotere instanties.

Daarnaast voeren we een reeks praktische experimenten uit met het oog voor beleidsrelevante inzichten. We vergelijken hierbij oplossingen voor volledige instanties met oplossingen voor kleinere deelverzamelingen van leerlingen van dezelfde instantie, om te onderzoeken of rittenplanning beter gebeurt op kleinere regionale deelverzamelingen of op één grotere regio. Vervolgens analyseren we het effect van het inzetten van meer of minder voertuigen. Tot slot bestuderen we de impact van ruimere tijdsvensters op de oplossingen.

Aangezien de instanties sterk verschillen in grootte en we willen dat een grote instantie evenveel rekentijd krijgt als de som van haar deelverzamelingen, schalen we de toegestane rekentijd proportioneel met het aantal leerlingen. In onze experimenten richten we ons op de provincie West-Vlaanderen, waarbij het arrondissement Brugge met 2205 leerlingen de grootste instantie vormt. We kennen dit arrondissement een rekentijd van 30 minuten toe. Voor alle overige instanties bepalen we de toegestane rekentijd volgens Formule (7.1). Hierdoor krijgen instanties met een kleiner aantal leerlingen minder rekentijd dan grotere instanties. In de resultaten rapporteren we de gemiddelde rekentijd T_{gem} . Deze kan

soms licht boven de toegestane rekentijd uitkomen, omdat we enkel aan het begin van elke iteratie controleren of de tijdslimiet al is overschreden.

$$\text{Rekentijd voor instantie} = \frac{\text{Rekentijd arrondissement Brugge (30 min)} \cdot \text{Aantal leerlingen van instantie}}{\text{Aantal leerlingen arrondissement Brugge (2205)}} \quad \text{Formule (7.1)}$$

7.2 Technische experimenten en resultaten

Resultaten van de SMM-FF heuristiek voor kleine instanties

We starten met de analyse van de kleine instanties, weergegeven in Tabel 7.1. Dit zijn de instanties die we in de exacte aanpak (Sectie 4.6) hebben gebruikt. Voor alle kleine instanties vindt de SMM-FF heuristiek oplossingen die alle leerlingen ophalen en de maximale rittijd minimaliseren. Hierdoor kunnen we de resultaten met de exacte aanpak vergelijken. De onderlijnde waarden in de tabel duiden de instanties aan waarvoor we met zekerheid een optimale oplossing hebben gevonden. Een oplossing is optimaal wanneer ze overeenkomt met het optimale resultaat van de exacte methode of wanneer de objectiefwaarde gelijk is aan de ondergrens.

Deze ondergrens berekenen we als de langste rechtstreekse rittijd van een leerling naar zijn school over alle leerlingen van de instantie.

Voor 14 van de 18 instanties waarvoor de exacte aanpak een optimale oplossing kon berekenen, vindt onze heuristiek eveneens de optimale oplossing. Daarnaast vinden we voor één instantie van 20 leerlingen een optimale oplossing (objectiefwaarde gelijk aan de ondergrens) terwijl de exacte methode daar geen geldige oplossing kon bepalen binnen één uur.

SMM-FF heuristiek met haar varianten

In Tabel 7.2 vergelijken we de SMM-FF heuristiek met de varianten Service-Maximization Minimax with Temporary-Fleet-Expansion (SMM-TFE) en Full-Service Minimax with Fleet-Expansion (FSM-FE) voor een willekeurige reeks middelgrote en grote instanties.

Tegen onze verwachting in leveren de SMM-FF en SMM-TFE heuristiek zeer gelijkaardige resultaten op over alle instanties. Hoewel SMM-TFE heuristiek in iets meer gevallen een kleine verbetering in de langste rittijd oplevert ten opzichte van de SMM-FF heuristiek, blijven de verschillen klein en mogelijk toevallig. Het gemiddeld aantal iteraties ligt in deze variant vaak licht lager dan bij de SMM-FF heuristiek, vermoedelijk omdat het algoritme tijdens de *recreate*-stap tijdelijk meer voertuigen moet overlopen, waardoor het potentiële voordeel van een kleinere *absent*-verzameling grotendeels verdwijnt.

Voor FSM-FE zien we dat wanneer de SMM-FF heuristiek bij middelgrote instanties niet in staat zijn alle leerlingen op te halen, het toevoegen van één of twee extra voertuigen het probleem doorgaans eenvoudig oplost. Hierdoor bevat de planning niet alleen alle leerlingen, maar daalt ook de langste rittijd aanzienlijk. Voor de grootste instanties, met meer dan 1500 leerlingen, stellen we echter vast dat het aantal uitgevoerde iteraties zeer beperkt blijft en dat er te weinig tijd is om tijdens de *minimisation*-fase het aantal extra voertuigen voldoende te reduceren. Dit lijkt te wijten aan het feit dat deze variant start vanuit een extreem grote initiële vloot waarbij elke leerling aanvankelijk een eigen voertuig krijgt. Tijdens de *recreate*-stap moet het algoritme al deze voertuigen overlopen, waardoor de rekentijd per iteratie sterk toeneemt. Hierdoor daalt het totale aantal iteraties dat binnen de toegestane rekentijd kan worden uitgevoerd, wat uiteindelijk leidt tot minder kwalitatieve oplossingen. Een mogelijke oplossing voor dit probleem is het genereren van een initiële oplossing met minder extra voertuigen, al blijft het daarbij een uitdaging om te garanderen dat alle leerlingen in deze fase steeds opgehaald worden.

Grootte	Instantie	Beste langste rittijd [min]	Gem. langste rittijd [min]	Ondergrens [min]	Gem. aantal iteraties	$T_{gem.}$ [s]
U5 K2 S2	Set1_U5_K2_S2_T1_2023	<u>43.2</u>	<u>43.2</u>	31.0	880622.4	4.082
	Set2_U5_K2_S2_T1_2023	33.2	33.2	25.8	841700.8	4.082
	Set3_U5_K2_S2_T1_2023	<u>33.8</u>	<u>33.8</u>	28.3	887447.0	4.082
	Set4_U5_K2_S2_T1_2023	<u>40.0</u>	<u>40.0</u>	30.5	930153.4	4.082
U6 K3 S2	Set1_U6_K3_S2_T1_2023	<u>31.0</u>	<u>31.0</u>	31.0	826770.8	4.898
	Set2_U6_K3_S2_T1_2023	19.3	21.3	16.0	859093.8	4.898
	Set3_U6_K3_S2_T0_2023	<u>14.3</u>	<u>14.3</u>	14.3	821944.4	4.898
	Set4_U6_K3_S2_T2_2023	<u>36.7</u>	<u>36.7</u>	27.5	815957.4	4.898
U7 K2 S3	Set1_U7_K2_S3_T2_2023	<u>42.7</u>	<u>42.7</u>	31.0	1002547.2	5.714
	Set2_U7_K2_S3_T1_2023	<u>60.7</u>	<u>60.7</u>	53.0	950219.2	5.714
	Set3_U7_K2_S3_T0_2023	<u>60.2</u>	60.8	42.3	1027137.4	5.714
	Set4_U7_K2_S3_T1_2023	<u>42.5</u>	<u>42.5</u>	23.8	1013105.0	5.714
U8 K3 S3	Set1_U8_K3_S3_T2_2023	<u>31.0</u>	<u>31.0</u>	31.0	947008.6	6.531
	Set2_U8_K3_S3_T1_2023	<u>16.5</u>	17.2	16.5	931453.6	6.531
	Set3_U8_K3_S3_T1_2023	<u>57.0</u>	57.2	32.3	982697.6	6.531
	Set4_U8_K3_S3_T1_2023	<u>33.5</u>	<u>33.5</u>	33.5	951779.6	6.531
U10 K3 S3	Set1_U10_K3_S3_T2_2023	51.2	51.8	31.0	998206.0	8.163
	Set2_U10_K3_S3_T1_2023	22.0	23.3	20.7	897278.4	8.163
	Set3_U10_K3_S3_T0_2023	40.2	40.2	38.2	992858.0	8.163
	Set4_U10_K3_S3_T2_2023	52.7	52.8	41.2	1037743.4	8.163
U20 K4 S5	Set1_U20_K4_S5_T_2023	44.5	50.0	29.8	1238858.4	16.327
	Set2_U20_K4_S5_T_2023	<u>78.2</u>	<u>78.2</u>	78.2	1430138.4	16.327
	Set3_U20_K4_S5_T_2023	81.8	82.7	49.5	1455914.2	16.327
	Set4_U20_K4_S5_T_2023	85.2	85.2	64.5	1660494.4	16.327

Tabel 7.1 Resultaten van vijf onafhankelijke uitvoeringen van de SMM-FF heuristiek voor kleine instanties (instanties gebruikt in exacte aanpak)

Instantie	Hoofdheuristiek					Fleet Expansion					Fleet Minimisation				
	Beste langste rittijd [min]	Gem. langste rittijd [min]	Gem. ontbr. leerl.	Gem. aantal iteraties	T_{gem} [s]	Beste langste rittijd [min]	Gem. langste rittijd [min]	Gem. ontbr. leerl.	Gem. aantal iteraties	T_{gem} [s]	Beste langste rittijd [min]	Gem. langste rittijd [min]	Gem. aantal extra voertuigen	Gem. aantal iteraties	T_{gem} [s]
city19_Zedelgem_WVL_U73_K4_S2_T7_2023	108.2	108.7	1	1664133	59.6	107.0	108.8	1	1761852	59.6	95.8	95.8	1	1586389	59.6
city22_Brugge_WVL_U1769_K94_S23_T29_2023	99.7	99.7	0	295429	1446.5	99.0	100.0	0	290425	1445.5	99.2	103.7	1465	4659	1446.0
city24_Beernem_WVL_U55_K3_S1_T6_2023	105.0	105.5	10	1506066	44.9	103.3	104.0	9	1350107	44.9	89.2	89.7	2	1677993	44.9
city29_Hooglede_WVL_U214_K11_S3_T16_2023	117.5	122.8	1	2130804	174.7	118.8	120.2	1	2246787	174.7	108.2	112.3	2	2161398	174.7
city30_Staden_WVL_U34_K2_S1_T6_2023	53.0	53.0	6	1625699	27.8	100.2	100.8	6	1460939	27.8	96.5	96.8	1	1728098	27.8
arron4_Kortrijk_WVL_U2106_K112_S23_T21_2023	91.8	96.2	0	218808	1721.9	95.5	96.8	0	158458	1721.7	71.0	77.0	1942	1303	1722.8
arron6_Roeselare_WVL_U1532_K82_S18_T17_2023	96.7	98.0	0	353299	1251.9	95.0	98.3	0	265933	1251.8	97.8	104.0	1291	3826	1252.2

Tabel 7.2 Vergelijking van de hoofdheuristiek en haar varianten voor geselecteerde instanties (vijf onafhankelijke uitvoeringen). De afkorting “Gem. ontbr. leerl.” verwijst naar het gemiddelde aantal leerlingen die ontbreekt in de oplossing en die deze oplossing dus niet vervoert.

7.3 Toepassingsgerichte experimenten

Voor de toepassingsgerichte experimenten maken we gebruik van de SMM-FF heuristiek.

Impact van het planningsniveau op de kwaliteit van de ritten

Dit experiment vergelijkt de kwaliteit van de rittenplanning op drie planningsniveaus: het volledige arrondissement, de afzonderlijke steden binnen dat arrondissement en de individuele scholen binnen elke stad. De samengevoegde voertuigvloten van de kleinere deelproblemen stemmen overeen met de voertuigvloot van het grotere probleem. De opgetelde rekentijd van de deelverzamelingen komt overeen met de rekentijd voor de grotere instantie.

Twee resultaten voor het arrondissement Roeselare en Ieper zijn weergegeven in Tabel 7.3 en Tabel 7.4. De resultaten voor de overige arrondissementen van West-Vlaanderen zijn opgenomen in Appendix D en vertonen een gelijkaardig patroon. In de tabellen verwijzen *obj.* naar de objectiefwaarde (langste rittijd) van de best gevonden oplossing en *ontbr. leerl.* naar het aantal leerlingen die deze oplossing niet vervoert.

Uit alle experimenten blijkt dat de oplossingskwaliteit verbetert wanneer de schaal vergroten. Zowel het aantal ontbrekende leerlingen als de langste rittijd daalt wanneer we plannen op stads- of arrondissementniveau in plaats van op schoolniveau. Dit toont aan dat plannen op een grotere schaal voordeliger is: met dezelfde totale vlootgrootte verkrijgen we betere oplossingen binnen dezelfde totale rekentijd.

Tot slot onderzoeken we of het algoritme kan opschalen naar de volledige provincie West-Vlaanderen (Tabel 7.5). In deze resultaten zien we dat het algoritme binnen de toegewezen rekentijd volgens Formule (7.1) onvoldoende iteraties kan uitvoeren. Hierdoor blijft een aantal ontbrekende leerlingen zeer groot en ligt de resulterende objectiefwaarde hoog. Wanneer we dezelfde provincie-instantie echter voor één uitvoering een ruimer tijdsbudget van tien uur geven, stijgt het aantal uitgevoerde iteraties tot 89988, zijn alle leerlingen opgehaald en bedraagt de langste rittijd 112 minuten. Dit is een duidelijke verbetering, al blijft voor instanties van deze schaal een nog groter aantal iteraties of een efficiëntere implementatie van het algoritme noodzakelijk om tot een kwalitatief sterke objectiefwaarde te komen.

Arrondissement	Stad	School
arron6_Roeselare_WV1_U1532_K82_S18_T17_2023 Obj. [min]: 96.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 1251.9	city15_Izegem_WV1_U160_K8_S3_T3_2023 Obj. [min]: 53.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 130.6	school198_Izegem_U94_K4_S1_T3_2023 Obj. [min]: 61.5 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 76.7
		school199_Izegem_U8_K1_S1_T3_2023 Obj. [min]: 66.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 6.5
		school200_Izegem_U58_K3_S1_T3_2023 Obj. [min]: 76.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 47.3
	city16_Ingelmunster_WV1_U61_K3_S1_T3_2023 Obj. [min]: 110.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 49.8	school203_Ingelmunster_U61_K3_S1_T3_2023 Obj. [min]: 110.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 49.8
	city28_Roeselare_WV1_U854_K45_S8_T13_2023 Obj. [min]: 89.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 697.5	school206_Roeselare_U77_K4_S1_T10_2023 Obj. [min]: 101.0 Ontbr. leerl.: 14 $T_{\text{gem.}}[s]$: 62.9
		school207_Roeselare_U50_K2_S1_T2_2023 Obj. [min]: 105.5 Ontbr. leerl.: 6 $T_{\text{gem.}}[s]$: 40.8
		school208_Roeselare_U74_K4_S1_T2_2023 Obj. [min]: 66.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 60.4
		school277_Roeselare_U121_K6_S1_T6_2023 Obj. [min]: 108.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 98.8
		school370_Roeselare_U184_K10_S1_T11_2023 Obj. [min]: 109.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 150.2
		school371_Roeselare_U137_K7_S1_T10_2023 Obj. [min]: 99.2 Ontbr. leerl.: 13 $T_{\text{gem.}}[s]$: 111.8
		school436_Roeselare_U190_K11_S1_T6_2023 Obj. [min]: 72.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 155.1
		school437_Roeselare_U21_K1_S1_T4_2023 Obj. [min]: 100.2 Ontbr. leerl.: 4 $T_{\text{gem.}}[s]$: 17.1
		city29_Hooglede_WV1_U214_K11_S3_T16_2023 Obj. [min]: 117.5 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 174.7
		school58_Hooglede_U80_K4_S1_T12_2023 Obj. [min]: 106.8 Ontbr. leerl.: 17 $T_{\text{gem.}}[s]$: 65.3
		school209_Hooglede_U95_K5_S1_T13_2023 Obj. [min]: 130.3 Ontbr. leerl.: 5 $T_{\text{gem.}}[s]$: 77.6
		school372_Hooglede_U39_K2_S1_T12_2023 Obj. [min]: 122.2 Ontbr. leerl.: 11 $T_{\text{gem.}}[s]$: 31.8
	city30_Staden_WV1_U34_K2_S1_T6_2023 Obj. [min]: 101.0 Ontbr. leerl.: 6 $T_{\text{gem.}}[s]$: 27.8	school278_Staden_U34_K2_S1_T6_2023 Obj. [min]: 100.2 Ontbr. leerl.: 6 $T_{\text{gem.}}[s]$: 27.8
	city31_Moorslede_WV1_U209_K11_S2_T12_2023 Obj. [min]: 96.5 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 170.6	school205_Moorslede_U48_K3_S1_T6_2023 Obj. [min]: 91.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 39.2
		school276_Moorslede_U161_K8_S1_T12_2023 Obj. [min]: 106.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 131.4

Tabel 7.3 Resultaten van de SMM-FF heuristiek voor het arrondissement Roeselare (vijf onafhankelijke uitvoeringen)

Arrondissement	Stad	School
arron3_Ieper_WV1_U698_K37_S9_T14_2023 Obj. [min]: 101.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 570.0	city3_Poperinge_WV1_U321_K17_S5_T12_2023 Obj. [min]: 101.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 262.1	school213_Poperinge_U40_K2_S1_T2_2023 Obj. [min]: 90.7 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 32.7
		school374_Poperinge_U160_K8_S1_T12_2023 Obj. [min]: 101.7 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 130.6
		school375_Poperinge_U68_K4_S1_T11_2023 Obj. [min]: 140.2 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 55.5
		school376_Poperinge_U10_K1_S1_T4_2023 Obj. [min]: 97.3 Ontbr. leerl.: 2 $T_{\text{gem.}}[s]$: 8.2
		school377_Poperinge_U43_K2_S1_T3_2023 Obj. [min]: 101.5 Ontbr. leerl.: 2 $T_{\text{gem.}}[s]$: 35.1
	city4_Ieper_WV1_U213_K11_S3_T5_2023 Obj. [min]: 74.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 173.9	school211_Ieper_U17_K2_S1_T3_2023 Obj. [min]: 53.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 13.9
		school212_Ieper_U149_K7_S1_T5_2023 Obj. [min]: 81.5 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 121.6
		school427_Ieper_U47_K2_S1_T3_2023 Obj. [min]: 96.2 Ontbr. leerl.: 2 $T_{\text{gem.}}[s]$: 38.4
	city7_Wervik_WV1_U164_K9_S1_T9_2023 Obj. [min]: 74.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 133.9	school435_Wervik_U164_K9_S1_T9_2023 Obj. [min]: 74.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 133.9

Tabel 7.4 Resultaten van de SMM-FF heuristiek voor het arrondissement Ieper (vijf onafhankelijke uitvoeringen)

Provincienaam	Obj. [min]	Ontbr. leerl.	$T_{\text{gem.}} [s]$	Gem. iteraties	Ondergrens [min]
West-Vlaanderen: province1_WV1_U8095_K428_S104_T30_2023	149.3	1964	6670.9	2355.6	87.2

Tabel 7.5 Resultaten van de SMM-FF heuristiek voor de provincie West-Vlaanderen met een rekentijd bepaald volgens Formule (7.1) (vijf onafhankelijke uitvoeringen)

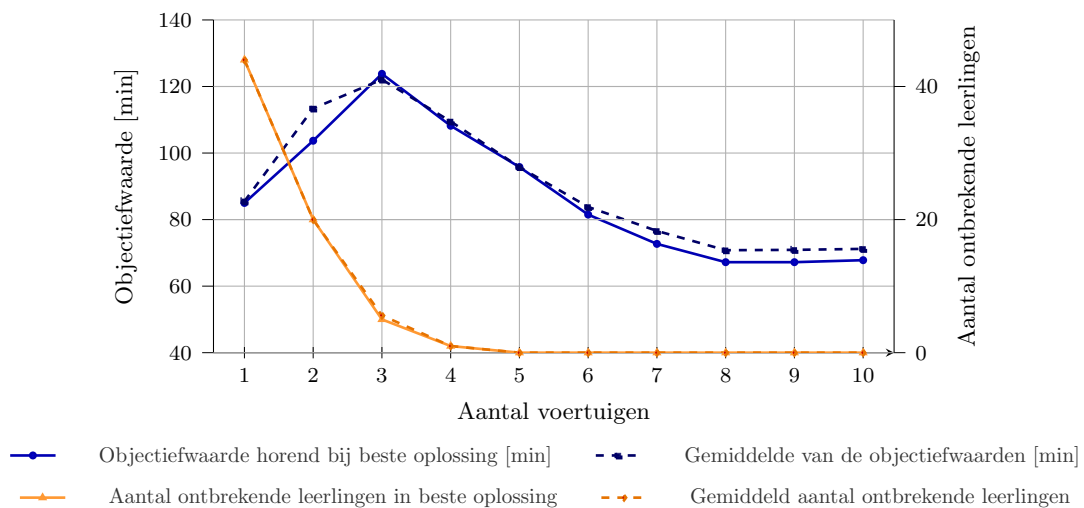
Invloed van vlootgrootte op de rittenplanning

In dit experiment onderzoeken we hoe de kwaliteit van de oplossingen evolueert wanneer we het aantal beschikbare voertuigen systematisch variëren. Hierbij geven we elk voertuig een vaste capaciteit van 25. We voeren deze analyse uit op drie willekeurig geselecteerde instanties, weergegeven in de Figuren 7.1 - 7.3.

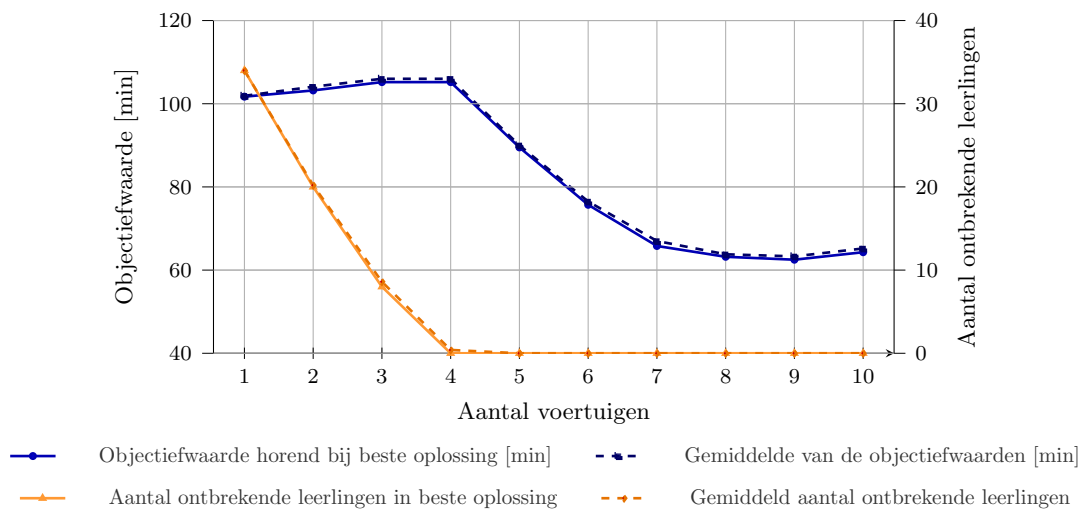
Over alle instanties heen stellen we een gelijkaardig patroon vast. Wanneer slechts een klein aantal voertuigen beschikbaar zijn, kunnen we niet alle leerlingen ophalen. Naarmate we voertuigen toevoegen, daalt dit aantal snel tot het punt waarop we alle leerlingen ophalen. In deze fase kan de langste rittijd stijgen, omdat we oplossingen waarin we meer leerlingen

ophalen hoger waarderen dan oplossingen met meer ontbrekende leerlingen, ongeacht de rittijd. Zodra we alle leerlingen ophalen, leidt het toevoegen van extra voertuigen aanvankelijk tot een sterke daling van de langste rittijd. Eens de vloot voldoende groot is om alle leerlingen efficiënt op te halen, zorgt het toevoegen van extra voertuigen slechts voor een zeer beperkte verdere verbetering. Ter illustratie van de rittijden in deze stabiele fase: voor Zedelgem (Figuur 7.1) bedraagt de langste rittijd ongeveer 67 minuten, terwijl de ondergrens 66 minuten bedraagt. Voor Beernem (Figuur 7.2) ligt de rittijd rond 63 minuten tegenover een ondergrens van 42 minuten. In Poperinge (Figuur 7.3) eindigen de oplossingen rond 91 minuten, vergeleken met een ondergrens van 78 minuten.

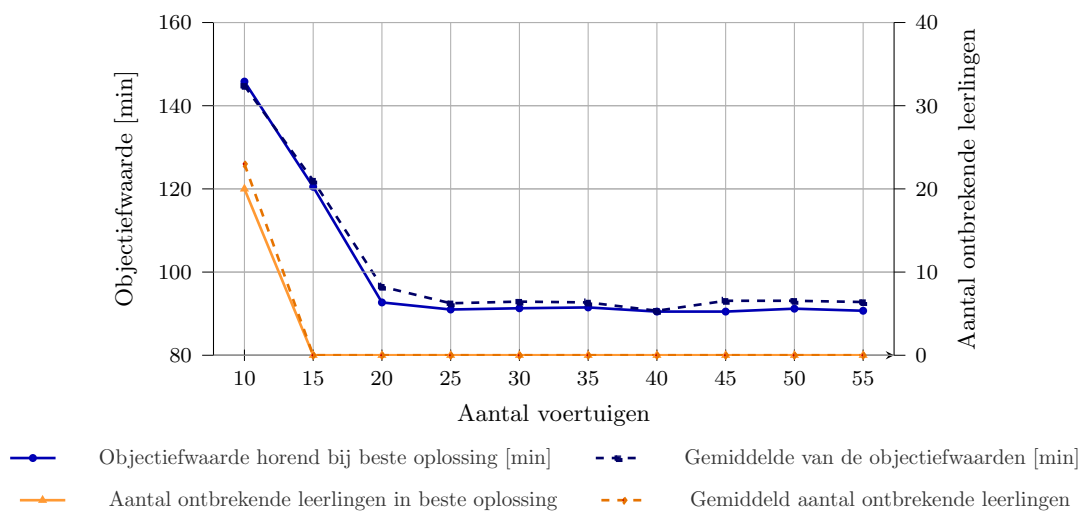
We concluderen dat het vergroten van de vloot voor een stad aanvankelijk grote verbetering oplevert, zowel door het elimineren van ontbrekende leerlingen als door het reduceren van de langste rittijd. Tegelijkertijd bestaat er een duidelijke drempelwaarde: zodra deze is bereikt, heeft het toevoegen van extra voertuigen geen betekenisvolle impact meer op de kwaliteit van de oplossing.



Figuur 7.1: Evolutie van de oplossing ten opzichte van het aantal voertuigen met een vaste capaciteit van 25 voor de instantie city19_Zedelgem_WV1_U73_K4_S2_T7_2023. De weergegeven waarden zijn het resultaat van vijf onafhankelijke uitvoeringen.



Figuur 7.2: Evolutie van de oplossing ten opzichte van het aantal voertuigen met een vaste capaciteit van 25 voor de instantie city24_Beernem_WV1_U55_K3_S1_T6_2023. De weergegeven waarden zijn het resultaat van vijf onafhankelijke uitvoeringen.



Figuur 7.3: Evolutie van de oplossing ten opzichte van het aantal voertuigen met een vaste capaciteit van 25 voor de instantie city3_Poperinge_WV1_U321_K17_S5_T12_2023. De weergegeven waarden zijn het resultaat van vijf onafhankelijke uitvoeringen.

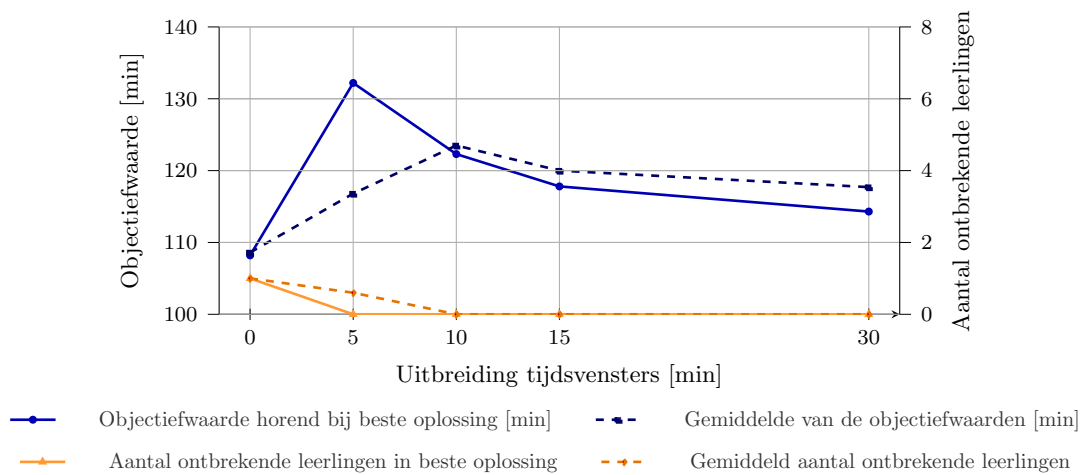
Impact van ruimere tijdsvensters op oplossingskwaliteit

Vervolgens onderzoeken we de invloed van ruimere tijdsvensters op de kwaliteit van de rittenplanning. We voeren deze analyse uit op dezelfde steden als in de voertuigexperimenten. De oorspronkelijke tijdsvensters verruimen we met 0, 5, 10, 15 en 30 minuten.

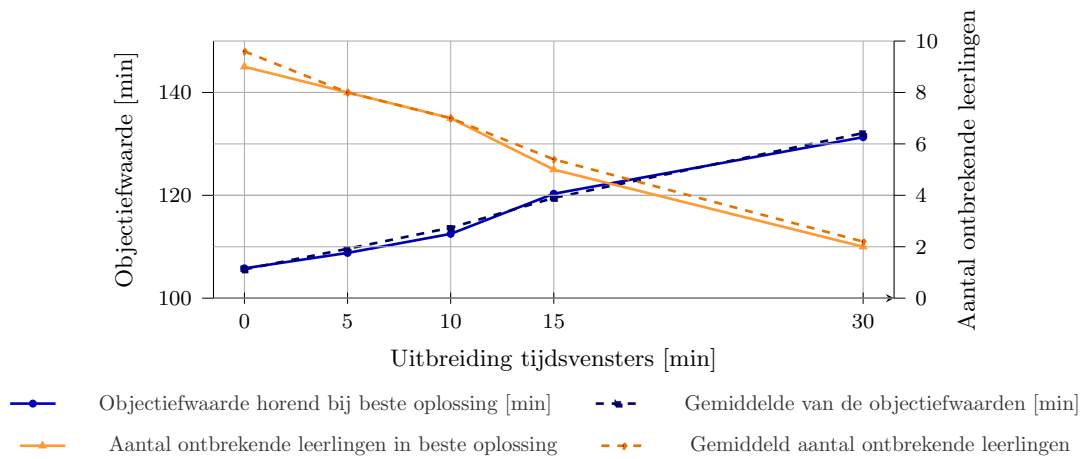
Uit Figuren 7.4 - 7.5 blijkt dat ruimere tijdsvensters doorgaans leiden tot een vermin-

dering van het aantal ontbrekende leerlingen of tot een daling van de langste rittijd. Een beperkte verruiming tussen de vijf en vijftien minuten kan al een positief effect hebben op de kwaliteit van de oplossingen.

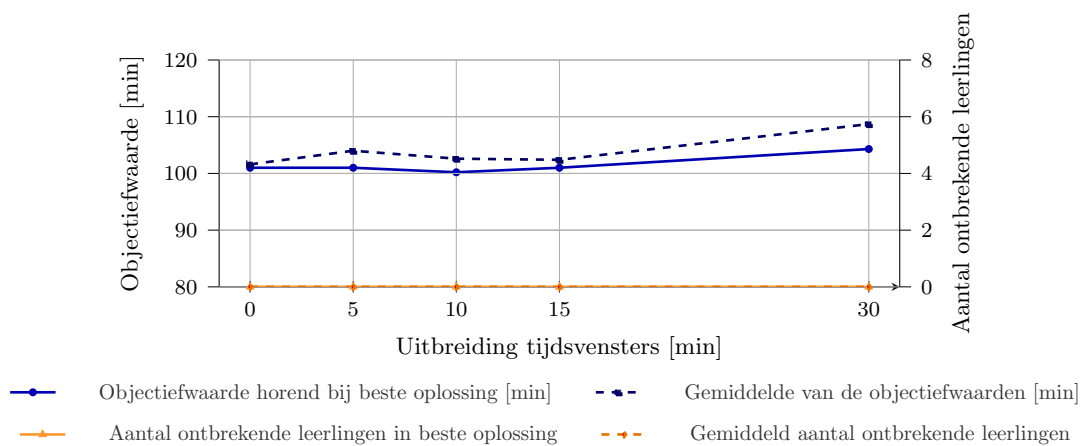
In Figuur 7.6 zien we echter een afwijkend patroon. Alle leerlingen worden hier vanaf het begin opgehaald en de langste rittijd blijft nagenoeg constant of stijgt zelfs licht. Dit komt niet overeen met onze verwachten en we vermoeden dat er verschillende mogelijke oorzaken voor zijn. Enerzijds kunnen de ruimere tijdsvensters ervoor zorgen dat het algoritme minder invoegposities vroegtijdig kan uitsluiten in de *recreate*-fase, waardoor deze stap langer duurt en het algoritme minder iteraties kan uitvoeren. Anderzijds kan de *compress schedule*-methode minder goede resultaten opleveren bij zeer ruime tijdsvensters, zoals besproken in Paragraaf 5.4.5.



Figuur 7.4: Evolutie van de oplossing in functie van een bepaalde uitbreiding aan alle tijdsvensters voor de instantie *city19_Zedelgem_WV1_U73_K4_S2_T7_2023*. De weergegeven waarden zijn het resultaat van vijf onafhankelijke uitvoeringen.



Figuur 7.5: Evolutie van de oplossing in functie van een bepaalde uitbreiding aan alle tijdsvensters voor de instantie city24_Beernem_WV1_U55_K3_S1_T6_2023. De weergegeven waarden zijn het resultaat van vijf onafhankelijke uitvoeringen.



Figuur 7.6: Evolutie van de oplossing in functie van een bepaalde uitbreiding aan alle tijdsvensters voor de instantie city3_Poperinge_WV1_U321_K17_S5_T12_2023. De weergegeven waarden zijn het resultaat van vijf onafhankelijke uitvoeringen.

Hoofdstuk 8

Conclusie

8.1 Besluit

Deze masterproef onderzocht twee centrale onderzoeksvragen: (1) kan een geautomatiseerd planningsalgoritme de rittenplanning voor het buitengewoon onderwijs opstellen en (2) is het mogelijk om de langste rittijden voor leerlingen in het buitengewoon onderwijs te verlagen?

Uit ons onderzoek blijkt dat het rittenplanningsprobleem aanzienlijk complexer is dan vooraf verwacht. Onze exacte aanpak bleek niet schaalbaar voor realistische probleemgroottes. Daarnaast toonde ons verkennend onderzoek naar transferpunten aan dat deze aanpak de rittijden mogelijk kan verlagen, maar vooral gepaard gaat met een sterke toename in complexiteit. Dit leidde tot de conclusie dat een flexibelere aanpak noodzakelijk is.

De heuristische methode kon binnen beperkte rekentijd oplossingen genereren op het niveau van arrondissementen en laat bovendien toe om praktische beleidsvragen, zoals het simuleren van extra bussen, te beantwoorden. Daarmee tonen we aan dat het wel degelijk mogelijk is om een geautomatiseerd planningsalgoritme te ontwikkelen dat bruikbaar is in de praktijk. Het blijft echter een grote uitdaging, zeker indien men alle beperkingen wil integreren en oplossingen wil vinden die voor alle betrokken partijen aanvaardbaar zijn.

Daarnaast stellen we vast dat de schaalgrootte van de planningen een belangrijke invloed heeft. Hoewel Rekenhof (2024) vermeldt dat er pilootprojecten bestaan die de planning opdelen in steeds kleinere regio's, suggereren onze resultaten dat het vanuit een louter planningsperspectief efficiënter is om te plannen in grotere regio's.

Voor de tweede onderzoeksvraag vergelijken we onze resultaten met het rapport van Rekenhof (2024) voor het jaar 2023. Volgens dit rapport waren er dat jaar 7076 leerlingen in West-Vlaanderen, waarvan 314 leerlingen een langere rittijd hadden dan 90 minuten

(4.4%). De maximale rittijd staat niet vermeld in het rapport. Indien we onze experimenten op alle arrondissementen van West-Vlaanderen samenvoegen (8095 leerlingen), leidt dit in de beste uitvoering tot een planning waarbij we één leerling niet ophalen, de maximale rittijd van 102 minuten bedraagt en 188 leerlingen een rittijd hebben boven de 90 minuten (2.3%). We kunnen dus niet bepalen of onze aanpak zou leiden tot een lagere langste rittijd, maar voor het jaar 2023 krijgen we een lager percentage leerlingen boven de 90 minuten dan de manuele planners. Opmerkelijk hierbij is dat in 2023 wel gebruik werd gemaakt van transferpunten. Daarbij moeten we echter in gedachten houden dat onze aanpak enkele vereenvoudigingen bevat, zoals het negeren van aanrijrichtingen en het niet meenemen van de planning van begeleiders.

Samengevat concluderen we dat de doelstellingen van deze masterproef zijn gerealiseerd. We ontwikkelden een planningsalgoritme dat oplossingen genereert binnen realistische rekentijd en dat kan worden ingezet om beleidsopties te evalueren. Tegelijkertijd toont ons onderzoek aan dat verder werk nodig is om een volledig praktijkklaar algoritme te bekomen. Volgende bouwstenen staan hiervoor alvast klaar:

- een realistische dataset
- een concrete probleemdefinitie en wiskundige formulering
- een implementatie waarin verschillende parameters kunnen worden aangepast
- inspiratie voor verdere uitbreidingen (zie Sectie 8.2)

Deze masterproef vormt daarmee een sterk vertrekpunt voor de verdere ontwikkeling betere rittenplanningen in het buitengewoon onderwijs.

8.2 Toekomstige onderzoeksmogelijkheden

In dit onderzoek zijn verschillende veronderstellingen doorgevoerd om het probleem beheersbaar te houden. Toch zijn diverse uitbreidingen interessant om verder te onderzoeken. Deze sectie biedt een overzicht van mogelijke richtingen voor verdere ontwikkeling.

Begeleider-leerlingconsistentie voor leerlingen met meerdere adressen

Een mogelijke uitbreiding is het scenario waarin leerlingen op bepaalde dagen vanaf een tweede adres (bijvoorbeeld bij grootouders) naar school vertrekken. Dit vraagt om een planningsstructuur over meerdere dagen. Het algoritme kan bijvoorbeeld een standaardplanning (sjabloon) genereren waarin leerlingen vertrekken op hun hoofdadres, zoals beschreven door Jost et al. (2022). Voor specifieke dagen is er dan nood aan een aangepaste dagplanning die rekening houdt met het alternatieve adres. Deze dagplanningen moeten consistent blijven met het sjabloon wat betreft begeleider-leerlingtoewijzing; elke leerling

moet dezelfde begeleider hebben, ongeacht het ophaaladres. Dit vereist bovendien dat we begeleiders expliciet aan bussen moeten toewijzen, wat in het huidige probleem niet nodig was. Het zorgt ook voor extra complexiteit voor transfers.

Meerdere capaciteitsvariabelen

Het huidige algoritme gaat uit van één uniforme capaciteitsvariabele per bus. In de praktijk beschikken bussen echter vaak over meerdere soorten capaciteit, zoals het aantal standaard zitplaatsen en het aantal rolstoelplaatsen. Een interessante uitbreiding is om deze capaciteiten afzonderlijk te modelleren, zodat het algoritme beter aansluit bij de realiteit van de beschikbare bussen.

Een voorbeeld hiervan is gegeven door Caceres et al. (2019), die verschillende busconfiguraties onderscheidt. Deze configuraties combineren een bepaald aantal gewone zitplaatsen en rolstoelplaatsen. Daarnaast behandelen Parragh et al. (2012) ook heterogene voertuigen, waarbij de patiënten verschillende vervoersvormen kunnen aanvragen, zoals al zittend, op een brancard of met een rolstoel.

Objectief

Het huidige objectief, het minimaliseren van de langste individuele rittijd, kunnen we verder verfijnen. In plaats van enkel naar de absolute rittijd te kijken, kunnen we deze proportioneel beoordelen ten opzichte van de reistijd bij een rechtstreekse rit van deze leerling naar school. Zo zou een busrit van 60 minuten voor een leerling die rechtstreeks op 50 minuten van school woont minder problematisch zijn dan een rit van 55 minuten voor iemand die normaal slechts 5 minuten onderweg is.

Daarnaast kan het interessant zijn om niet uitsluitend te focussen op de langste rittijd, maar bijvoorbeeld op de 5% leerlingen met de langste rittijden. Dit voorkomt dat het objectief volledig wordt gedomineerd door één extreme waarde.

Transfers toelaten in een heuristische methode

Na ons verkennend onderzoek naar transferpunten in Sectie 4.4 concludeerden we dat transferpunten potentieel de rittijden kunnen verlagen, maar dat ze aanzienlijke uitdagingen meebrengen op het vlak van wiskundige modellering, implementatie en computationele complexiteit. Om die reden hebben we ervoor gekozen om te focussen op het probleem zonder transferpunten. We verwachten dat deze variant resultaten oplevert die voldoende goed zijn voor de beoogde toepassing, terwijl de bijkomende logistieke uitdagingen van transfers vermeden worden.

Indien men er in verder onderzoek voor kiest om toch transfers te integreren in een heu-

ristische methodologie, brengt dit verschillende uitdagingen met zich mee. De grootste uitdaging is de volledige temporele synchronisatie, zoals gedefinieerd in Paragraaf 4.4.1. Synchronisatie zorgt ervoor dat een wijziging in één rit invloed kan hebben op alle andere ritten (Drexel, 2012). Dit verhoogt de complexiteit aanzienlijk en maakt het toepassen van standaardtechnieken zoals *column generation* en *local search* bijzonder moeilijk. Drexel (2012) biedt een mogelijk startpunt, met een overzicht van verschillende vormen van synchronisatie, elk met hun eigen uitdagingen.

In de literatuur zijn enkele benaderingen gevonden die transfers en een vorm van synchronisatie behandelen:

- Cortés et al. (2010) en Rais et al. (2014) definiëren synchronisatie eenvoudiger als een beperking die de volgorde op transferpunten afdwingt; wanneer een transfer van voertuig k naar voertuig l plaatsvindt, moet l aangekomen zijn voordat k vertrekt.
- Bij Bögl et al. (2015) hoeven voertuigen ook niet gelijktijdig aanwezig te zijn tijdens een transfer, maar is er een extra beperking op de maximale wachttijd van een leerling op het transferpunt. Bögl et al. (2015) benadrukken dat transfers een aanzienlijke impact hebben op de servicekwaliteit: ze verlagen het gebruiksgemak wat de servicekwaliteit doet dalen, maar kunnen dit bij voldoende tijdswinst compenseren.
- Souza et al. (2024) introduceren een vorm van synchronisatie die nauwer aansluit bij ons probleem: alle uitlaadoperaties moeten voltooid zijn voordat het herladen kan beginnen. Een verschil is echter dat het uitladen al mag starten voordat het andere voertuig arriveert.
- Gandra et al. (2022) stellen een tweefasige heuristiek voor: eerst selecteert het algoritme de transferlocaties en de bijhorende transfers die zullen worden uitgevoerd, waarna ze een standaard PDPTW-heuristiek toepassen. In onze context is deze aanpak echter niet haalbaar, omdat het vooraf inschatten van winstgevendende transfers door het grote aantal combinaties vrijwel onmogelijk is. Ook voor deze methode hoeven voertuigen niet gelijktijdig aanwezig te zijn op een transferlocatie.
- Andere werken hanteren één centraal transferpunt (Dondo et al., 2009; Grangier et al., 2021; Souza et al., 2024). Bij Grangier et al. (2021) moeten alle voertuigen dit transferpunt verplicht passeren, terwijl Dondo et al. (2009) dit optioneel maakt.

Ondanks deze inzichten biedt geen enkele bestaande methode een volledige oplossing voor ons probleem. De besproken literatuur kan dienen als inspiratie voor verder onderzoek, maar op basis van onze bevindingen raden we een aanpak aan die vertrekt vanuit onze heuristiek (Hoofdstuk 5) en synchronisatie introduceert.

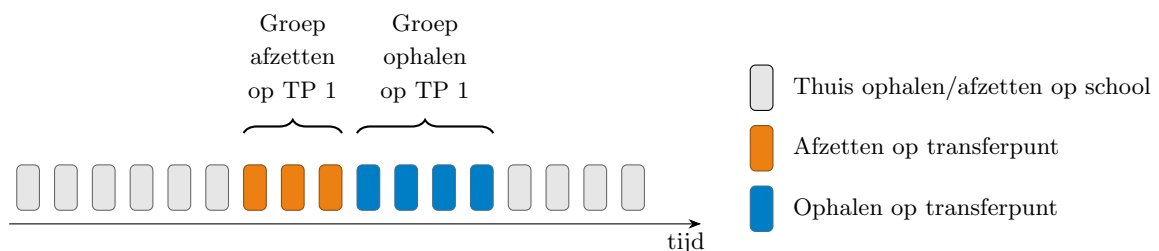
In de huidige oplossingsvoorstelling zijn er per leerling twee acties: ophalen en afzetten. Voor transfers stellen we voor om deze uit te breiden met twee extra actietypes: ophalen

op een transferpunt en afzetten op een transferpunt. Elke leerling krijgt dus precies vier acties, wat voldoende is indien we aannemen dat een leerling hoogstens één transfer ondergaat.

In de *recreate*-fase van de heuristiek proberen we dan niet langer slechts twee acties aan één rit in te voegen, maar vier acties verdeeld over één of twee ritten. Wanneer we alle acties in één rit plaatsen, gebruiken we een dummy-transferlocatie die geen echte transfer inhoudt. Indien we twee ritten gebruiken, moet het algoritme een transferlocatie kiezen. De eerste rit bevat in dit geval het ophalen thuis en het afzetten op het transferpunt van de leerling, terwijl de tweede rit het ophalen op het transferpunt en het afzetten op school bevat. Deze uitbreiding vergroot de zoekruimte aanzienlijk en zal de *recreate*-fase vertragen. Om de complexiteit enigszins te beperken, kan men bijvoorbeeld aannemen dat elke bus maximaal één transferpunt bezoekt.

Naast deze uitbreiding moeten we ook nog de synchronisatie op transferpunten in rekening brengen. Tijdens het evalueren van alle invoegposities in de *recreate*-fase willen we snel kunnen inschatten of synchronisatie haalbaar is. Hiervoor stellen we voor om alle afzetacties op een transferpunt binnen een rit te groeperen (zie Figuur 8.1). Elke groep krijgt de variabelen wachttijd, starttijd van de service, aankomsttijd en maximale verschuiving, analoog aan de definities van Vansteenwegen et al. (2009). Bij het simuleren van een invoeging hoeft het algoritme dan enkel naar de groep van die rit te kijken om te weten of synchronisatie mogelijk zal zijn. Wanneer we een actie effectief invoegen, moeten we deze waarden opnieuw synchroniseren over alle ritten die hetzelfde transferpunt bezoeken. Zo verschuift bijvoorbeeld de starttijd van de afzetgroep naar de laatste starttijd over alle voertuigen die op dat transferpunt passeren. Hetzelfde principe geldt voor de ophaalgroepen op transferpunten.

De *ruin*-methode zou in omgekeerde richting werken. Na het verwijderen van één of meerdere acties stellen we de starttijd van de transfer op het transferpunt opnieuw zo vroeg mogelijk in. Dit gebeurt door wachttijden over alle betrokken ritten te reduceren.



Figuur 8.1: Visualisatie van afzet- en ophaalgroepen op transferpunten (TP) voor één busrit

Deze aanpak maakt het naar verwachting mogelijk om transfers en synchronisatie te inte-

greren in de bestaande heuristiek, maar vereist nog grondige uitwerking en evaluatie om de computationele haalbaarheid te garanderen. We verwachten echter dat dit een grote impact zal hebben op de rekentijd. Het huidige algoritme is namelijk sterk geoptimaliseerd voor het snel uitsluiten van ongeldige posities op basis van tijdsvensters en capaciteit. Transfers maken deze filtering complexer, omdat posities die eerder ongeldig waren door een transfer alsnog geldig kunnen zijn. Bovendien verdubbelt het aantal acties dat het algoritme moet invoegen. Tot slot moeten we bij elke transferactie een geschikte transferlocatie kiezen of moeten we alle mogelijke transferpunten evalueren, wat de berekeningen verder kan vertragen.

Bibliografie

- agentschap Digitaal Vlaanderen - Helpdesk Digitaal Vlaanderen. (2023). *Gebouwen- en adressenregister* [Identifier: 5ee6779b-f33f-436b-8b81-0ee9a52f2b3c]. Downloadtoepassing Digitaal Vlaanderen. Geraadpleegd op 4 september 2025, van <https://download.vlaanderen.be/product/10142-gebouwen-en-adressenregister>
- Agentschap Wegen en Verkeer. (2020). *Parkeerplaatsen*. Vlaanderen Datavindplaats. Geraadpleegd op 16 september 2025, van <https://www.vlaanderen.be/datavindplaats/catalogus/parkeerplaatsen>
- Ansari, A., Farrokhvar, L., & Kamali, B. (2021). Integrated student to school assignment and school bus routing problem for special needs students. *Transportation Research Part E: Logistics and Transportation Review*, 152, 102416. <https://doi.org/https://doi.org/10.1016/j.tre.2021.102416>
- Bögl, M., Doerner, K. F., & Parragh, S. N. (2015). The school bus routing and scheduling problem with transfers. *Networks*, 65(2), 180–203. <https://doi.org/https://doi.org/10.1002/net.21589>
- Caceres, H., Batta, R., & He, Q. (2019). Special need students school bus routing: Consideration for mixed load and heterogeneous fleet. *Socio-Economic Planning Sciences*, 65, 10–19. <https://doi.org/https://doi.org/10.1016/j.seps.2018.02.008>
- Christiaens, J., Çalik, H., Wauters, T., Chirayil Chandrasekharan, R., & Vanden Berghe, G. (2020). The prisoner transportation problem. *European Journal of Operational Research*, 284(3), 1058–1073. <https://doi.org/https://doi.org/10.1016/j.ejor.2020.01.027>
- Christiaens, J., & Vanden Berghe, G. (2020). Slack Induction by String Removals for Vehicle Routing Problems.
- Coach Partners. (s.d.-a). *Met de bus naar school*. Geraadpleegd op 22 september 2025, van <https://www.coachpartners.be/nl/wagenpark-busreizen/schoolbussen-schoolreizen/>
- Coach Partners. (s.d.-b). *Zalig reizen in een luxe autocar: Van luxe autocar tot VIP-minibus*. Geraadpleegd op 22 september 2025, van <https://www.coachpartners.be/nl/wagenpark-busreizen/autocar-touringcar/>
- Cordeau, J.-F., & Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Ann Oper Res*, 153(1), 29–46. <https://doi.org/https://doi.org.kuleuven.e-bronnen.be/10.1007/s10479-007-0170-8>
- Cortés, C. E., Matamala, M., & Contardo, C. (2010). The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3), 711–724. <https://doi.org/https://doi.org/10.1016/j.ejor.2009.01.022>
- Data processed by Geofabrik GmbH and created by OpenStreetMap Contributors. (s.d.). *OpenStreetMap Data: Belgium OSM PBF File*. Geraadpleegd op 19 september 2025, van <https://download.geofabrik.de/europe/belgium.html>
- Dondo, R., Méndez, C. A., & Cerdá, J. (2009). The supply-chain pick-up and delivery problem with transshipments. In J. Jeżowski & J. Thullie (Red.), *19th European Symposium on Computer Aided Process Engineering* (pp. 1009–1014, Deel 26). Elsevier. [https://doi.org/https://doi.org/10.1016/S1570-7946\(09\)70168-3](https://doi.org/https://doi.org/10.1016/S1570-7946(09)70168-3)

- Drexl, M. (2012). Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints. *Transportation Science*, 46(3), 297–316. <https://doi.org/http://dx.doi.org/10.1287/trsc.1110.0400>
- Factcheckers. (2025). *Britt, Jan en Thomas nemen de schoolbus voor een wel heel lange busrit* [[Online Video]]. VRT MAX. Geraadpleegd op 2 september 2025, van <https://www.vrt.be/vrtmax/a-z/factcheckers/6/factcheckers-s6a4/>
- FBAA. (2025). *45.000 leerlingen uit buitengewoon onderwijs nemen schoolbus in Vlaanderen: +28% meer leerlingen sinds 2021: meer vervoersbudget en overleg nodig*. Geraadpleegd op 4 september 2025, van https://www.fbaa.be/sites/default/files/2025-08/persbericht_fbaa_start_schooljaar_2025-2026.pdf
- Ferone, D., Festa, P., Pastore, T., & Resende, M. G. (2023). Efficient GRASP solution approach for the Prisoner Transportation Problem. *Computers & Operations Research*, 153, 106161. <https://doi.org/https://doi.org/10.1016/j.cor.2023.106161>
- Gandra, V. S., Çalik, H., Toffolo, T. A., Carvalho, M. A. M., & Vanden Berghe, G. (2022). The vessel swap-body routing problem. *European Journal of Operational Research*, 303(1), 354–369. <https://doi.org/https://doi.org/10.1016/j.ejor.2022.02.015>
- Google. (2025). *Google Maps* [Gebruikt om busparkings op te zoeken in de regio Vlaanderen en Brussel]. Geraadpleegd op 16 september 2025, van <https://maps.google.com>
- Grangier, P., Gendreau, M., Lehuédé, F., & Rousseau, L.-M. (2021). The vehicle routing problem with cross-docking and resource constraints. *J Heuristics*, 27, 31–61. <https://doi.org/https://doi-org.kuleuven.e-bronnen.be/10.1007/s10732-019-09423-y>
- Jost, C., Jungwirth, A., Kolisch, R., & Schiffels, S. (2022). Consistent vehicle routing with pickup decisions - Insights from sport academy training transfers. *European Journal of Operational Research*, 298(1), 337–350. <https://doi.org/https://doi.org/10.1016/j.ejor.2021.06.035>
- Kamali, B., Mason, S. J., & Pohl, E. A. (2013). An Analysis of Special Needs Student Busing. *Journal of Public Transportation*, 16(1), 21–45. <https://doi.org/https://doi.org/10.5038/2375-0901.16.1.2>
- Keolis. (s.d.). *Onze vloot*. Geraadpleegd op 22 september 2025, van <https://www.keolis.be/nl/onze-vloot>
- Kinable, J., Spijksma, F., & Vanden Berghe, G. (2014). School bus routing—a column generation approach. *International Transactions in Operational Research*, 21(3), 453–478. <https://doi.org/https://doi.org/10.1111/itor.12080>
- Lee, D.-H. (2014). Models of bus boarding and alighting dynamics. *Transportation Research Part A: Policy and Practice*. <https://doi.org/10.1016/J.TRA.2014.09.007>
- Lima, F. S., Pereira, D. S., da Conceição, S. V., & de Camargo, R. S. (2017). A multi-objective capacitated rural school bus routing problem with heterogeneous fleet and mixed loads. *4OR-Q J Oper Res*, 15, 359–386. <https://doi.org/https://doi-org.kuleuven.e-bronnen.be/10.1007/s10288-017-0340-8>
- Luxen, D., & Vetter, C. (2011). Real-time routing with OpenStreetMap data. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 513–516. <https://doi.org/10.1145/2093973.2094062> Code beschikbaar op: <https://github.com/Project-OSRM/osrm-backend?tab=readme-ov-file>
- Newton, R. M., & Thomas, W. H. (1969). Design of school bus routes by computer. *Socio-Economic Planning Sciences*, 3(1), 75–85. [https://doi.org/https://doi.org/10.1016/0038-0121\(69\)90051-2](https://doi.org/https://doi.org/10.1016/0038-0121(69)90051-2)
- Nominatim. (s.d.). *Open-source geocoding: with OpenStreetMap data*. Geraadpleegd op 19 september 2025, van <https://nominatim.org/>
- Onderwijskiezer. (s.d.). *Welke types zijn er in het buitengewoon onderwijs?* Geraadpleegd op 24 oktober 2025, van <https://www.onderwijskiezer.be/v2/info/buitengewoon-onderwijs/types.php>
- Paradigm. (2022). *BeSt Address - Brussels Region*. Datastore.brussels. Geraadpleegd op 17 september 2025, van <https://datastore.brussels/web/data/dataset/a8c9ccde-5c2b-11ed-913a-900f0cda5d5c#access>
- Park, J., & Kim, B.-I. (2010). The school bus routing problem: A review. *European Journal of Operational Research*, 202(2), 311–319. <https://doi.org/https://doi.org/10.1016/j.ejor.2009.05.017>

- Park, J., Tae, H., & Kim, B.-I. (2012). A post-improvement procedure for the mixed load school bus routing problem. *European Journal of Operational Research*, 217(1), 204–213. <https://doi.org/https://doi.org/10.1016/j.ejor.2011.08.022>
- Parragh, S. N., Cordeau, J.-F., Doerner, K. F., & Hartl, R. F. (2012). Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR Spectrum*, 34, 593–633. <https://doi.org/https://doi-org.kuleuven.e-bronnen.be/10.1007/s00291-010-0229-9>
- Rais, A., Alvelos, F., & Carvalho, M. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3), 530–539. <https://doi.org/https://doi.org/10.1016/j.ejor.2013.10.038>
- Rekenhof. (2024). *Buitengewoon onderwijs: toegang en uitstroom* [Verslag online bezocht]. Geraadpleegd op 4 september 2025, van https://www.ccrek.be/sites/default/files/Docs/buitengewoon_onderwijs_toegang_uitstroom.pdf
- Rekiek, B., Delchambre, A., & Saleh, H. A. (2006). Handicapped Person Transportation: An application of the Grouping Genetic Algorithm. *Engineering Applications of Artificial Intelligence*, 19(5), 511–520. <https://doi.org/https://doi.org/10.1016/j.engappai.2005.12.013>
- Riera-Ledesma, J., & Salazar-González, J.-J. (2012). Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. *Computers & Operations Research*, 39(2), 391–404. <https://doi.org/https://doi.org/10.1016/j.cor.2011.04.015>
- Russell, R. A., Morrel, R. B., & Haddox, B. (1986). Routing Special-Education School Buses. *Interfaces*, 16(5), 56–64. Geraadpleegd op 15 augustus 2025, van <http://www.jstor.org/stable/25060868>
- Souza, V. A., Melo, R. A., & Mateus, G. R. (2024). The Vehicle Routing Problem with Cross-Docking and Scheduling at the Docking Station: Compact formulation and a General Variable Neighborhood Search metaheuristic. *Applied Soft Computing*, 161, 111744. <https://doi.org/https://doi.org/10.1016/j.asoc.2024.111744>
- Tellez, O., Vercraene, S., Lehuédé, F., Péton, O., & Monteiro, T. (2022). The time-consistent dial-a-ride problem. *Networks*, 79, 452–478. <https://doi.org/https://doi-org.kuleuven.e-bronnen.be/10.1002/net.22063>
- 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*, 36(12), 3281–3290. <https://doi.org/https://doi.org/10.1016/j.cor.2009.03.008>
- Vermeeren Travel. (s.d.). *Leerlingenvervoer*. Geraadpleegd op 22 september 2025, van <https://www.vermeeren.be/autocars/leerlingenvervoer>
- Vlaams ministerie van onderwijs en vorming. (s.d.). *Inschrijvingen - Buitengewoon onderwijs*. Dataloep. Geraadpleegd op 19 september 2025, van https://public.tableau.com/views/DataloepInschrijvingenLeerplicht/BGKaart?%3Adisplay_count=n&%3Aorigin=viz_share_link&publish=yes&%3AshowVizHome=no&%3Atoolbar=top
- Vlaams ministerie van onderwijs en vorming & AGODI - Agentschap voor Onderwijsdiensten. (2024). *Zonaal leerlingenvervoer in het buitengewoon onderwijs*. Geraadpleegd op 4 september 2025, van <https://data-onderwijs.vlaanderen.be/documenten/bestanden/Leerlingenvervoer-202405.pdf>
- Wiskunde.net. (2026). *Standaarddeviatie*. Geraadpleegd op 3 januari 2026, van <https://www.wiskunde.net/standaarddeviatie>

Bijlage A

Verschillende definities voor de rittijd

In het basismodel is rittijd gedefinieerd als de tijd die een leerling doorbrengt in het voertuig (en eventueel op transferpunten). De rittijd start vanaf het moment direct na het instappen en eindigt direct na het uitstappen van de leerling. Dit komt overeen met optie 2 in Figuur A.1 voor de starttijd en optie 3 in Figuur A.2 voor de eindtijd. De eindtijd hangt hierbij af van de volgorde waarin leerlingen van dezelfde school uitstappen.

De rittijd van een leerling kunnen we echter op verschillende manieren definiëren, afhankelijk van hoe we het start- en eindmoment van de rit bepalen. De definities kunnen invloed hebben op ons objectief.

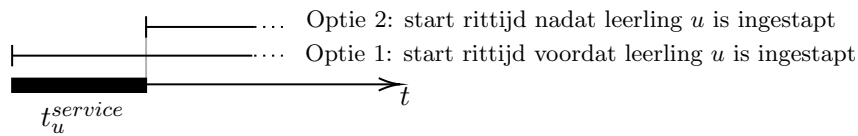
Zoals weergegeven in Figuur A.1 zijn er twee mogelijkheden voor de starttijd:

- Optie 1: De tijdsmeting start net voordat de leerling is ingestapt, waardoor de instaptijd (= servicetijd) wel onderdeel is van de rittijd.
- Optie 2: De tijdsmeting start nadat de leerling is ingestapt, waardoor de instaptijd (= servicetijd) geen onderdeel is van de rittijd.

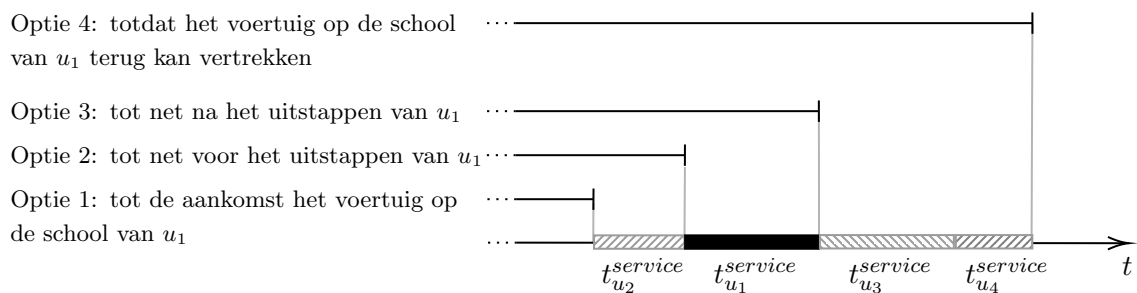
Daarnaast zijn er vier mogelijkheden voor het definiëren van de eindtijd van de rit, weergegeven in Figuur A.2:

- Optie 1: De rittijd van een leerling eindigt zodra het voertuig stopt aan de school van de leerling. De uitstaptijd (= servicetijd) van deze leerling en van andere leerlingen met dezelfde schoollocatie tellen niet mee.
- Optie 2: De rittijd eindigt net voor het uitstappen van de leerling. De eigen uitstaptijd (= servicetijd) telt niet mee in de rittijd, maar de uitstaptijd van eerder uitstappende leerlingen wel. De volgorde van uitstappen beïnvloedt hierbij de rittijd.

- Optie 3: De rittijd eindigt direct na het uitstappen van de leerling, inclusief de eigen uitstaptijd. Ook hier speelt de uitstapvolgorde een rol.
- Optie 4: De rittijd loopt door tot alle leerlingen op deze fysieke school zijn uitgestapt.



Figuur A.1: Twee mogelijke starttijden van de rittijd van een leerling



Figuur A.2: Vier mogelijke eindtijden van de rittijd van leerling u_1 , wanneer de leerlingen u_1 , u_2 , u_3 en u_4 op dezelfde school afgezet worden

De rittijd kan dus worden bepaald door elke combinatie van start- en eindtijd. Indien andere leerlingen onderweg in- of uitstappen, is hun servicetijd onderdeel van de rittijd. Ook wachttijd op knooppunten of eventueel tijd op transferpunten onderweg tellen we mee in de rittijd.

Bijlage B

Alternatieve formulering met flowvariabelen

In deze bijlage beschrijven we een alternatieve formulering van het probleem op basis van flowvariabelen.

B.1 Beslissingsvariabelen

De variabelen x en τ blijven hetzelfde als in de oorspronkelijke formulering. De variabele f stelt de flow voor over een boog.

$f_{ij} \geq 0$: Het aantal leerlingen dat boog (i, j) gebruikt

B.2 Formulering

Het objectief, zoals bepaald door Beperkingen (4.1 - 4.2), blijft ongewijzigd ten opzichte van Paragraaf 4.3.2. Verder blijven ook Beperkingen (4.3 - 4.7) behouden.

Beperkingen (4.8 - 4.9) voegen we samen tot Beperking (B.1). Op een thuisadres is het verschil tussen uitgaande en inkomende flow gelijk aan één, wat betekent dat er op elk thuisadres precies één leerling instapt.

$$\sum_{\substack{j \in N \\ (a_u, j) \in E}} f_{a_u j} - \sum_{\substack{i \in N \\ (i, a_u) \in E}} f_{i a_u} = 1 \quad \forall u \in U \quad (\text{B.1})$$

Op analoge wijze voegen we Beperkingen (4.10 - 4.11) samen tot Beperking (B.2). Het verschil tussen de inkomende en uitgaande flow op een school is gelijk één, dit betekent dat er op elke school één leerling uitstapt.

$$\sum_{\substack{j \in N \\ (s_u, j) \in E}} f_{s_u j} - \sum_{\substack{i \in N \\ (i, s_u) \in E}} f_{i s_u} = -1 \quad \forall u \in U \quad (\text{B.2})$$

Op scholen en thuisadressen is het verschil in de flow inmiddels gedefinieerd. Het behoud van de flow in het virtuele depot ligt vast via Beperking (B.3). Deze beperking vervangt Beperking (4.12) uit de oorspronkelijke formulering.

$$\sum_{\substack{i \\ \text{met } (i, n) \in E}} f_{i n} - \sum_{\substack{j \\ \text{met } (n, j) \in E}} f_{n j} = 0 \quad \forall n \in N \setminus \{A, S\} \quad (\text{B.3})$$

Beperking (B.4) vervangt Beperking (4.14). Ze garandeert dat een leerling alleen kan instappen wanneer er voldoende capaciteit beschikbaar is in het voertuig. Voor elke boog geldt dat de flow kleiner moet zijn dan de capaciteit van voertuig k , indien k deze boog neemt. Dit is correct door Eigenschap BCB (4.2.2).

$$f_{ij} \leq c_k + C_{\max} \cdot (1 - x_{ijk}) \quad \forall (i, j) \in E, \forall k \in K \quad (\text{B.4})$$

Daarnaast is Beperking (4.15) herschreven als Beperking (B.5). Deze koppeling garandeert dat de flow nul is op bogen die geen enkel voertuig gebruikt.

$$f_{ij} \leq \sum_{k \in K} x_{ijk} \cdot c_k \quad \forall (i, j) \in E, \forall u \in U \quad (\text{B.5})$$

Vervolgens blijven Beperkingen (4.16 - 4.17) behouden. Ook de extra geldige Beperkingen (4.18 - 4.20, 4.22) blijven ongewijzigd, terwijl we Beperking (4.21) vervangen door Beperking (B.6). De uitgaande flow in het virtuele depot is nul, aangezien hier geen leerlingen kunnen vertrekken.

$$f_{n_{\text{virtual}} j} = 0 \quad \forall j \in N \text{ met } (n_{\text{virtual}}, j) \in E \quad (\text{B.6})$$

Bijlage C

Pseudocode voor de heuristieken

Algoritme 2: SMM-FF heuristiek

Input: Initiële oplossing $s_{lokaal} = \{T, A\}$ met T de verzameling ritten en A de *absent*-verzameling

Output: Verbeterde oplossing s_{best}

procedure SMM-FF heuristiek(s_{lokaal}):

```
     $s_{best} \leftarrow s_{lokaal}$ ;
     $\mathcal{T} \leftarrow \mathcal{T}_0$ ;
    while tijdslimiet niet bereikt do
         $s_{nieuw} \leftarrow \text{ruin}(s_{lokaal})$ ;
         $s_{nieuw} \leftarrow \text{recreate}(s_{nieuw})$ ;
         $s_{nieuw} \leftarrow \text{compress\_ritijden}(s_{nieuw})$ ;
        if  $|A_{nieuw}| < |A_{best}|$  or  $\text{isBeter}(s_{nieuw}, s_{best})$  then
             $s_{best} \leftarrow s_{nieuw}$ ;
         $\text{geaccepteerd} \leftarrow (|A_{nieuw}| < |A_{lokaal}|) \text{ or } \text{isBeter}(s_{nieuw}, s_{lokaal})$ ;
        if not  $\text{geaccepteerd}$  then
            if  $|\text{Rittijden}(s_{lokaal})| = |\text{Rittijden}(s_{nieuw})|$  then
                 $\lambda \leftarrow 1.0$ ;
                for  $i = 1$  to  $|\text{Rittijden}(s_{nieuw})|$  do
                    if  $\text{Rittijden}(s_{nieuw})[i] = \text{Rittijden}(s_{lokaal})[i]$  then
                         $\lambda \leftarrow 0.95\lambda$ ;
                        continue
                    if  $\text{Rittijden}(s_{nieuw})[i] < \text{Rittijden}(s_{lokaal})[i] - \lambda \cdot \mathcal{T} \cdot \ln(\mathcal{U}(0, 1))$  then
                         $\text{geaccepteerd} \leftarrow \text{true}$ 
                        break
            if  $\text{geaccepteerd}$  then
                 $s_{lokaal} \leftarrow s_{nieuw}$ 
                 $\mathcal{T} \leftarrow \min\left(\mathcal{T}_0, \mathcal{T}_0 \cdot \left(\frac{\text{resterende tijd (sec)}}{\text{totale tijd (sec)}}\right)^5 + \mathcal{T}_{\text{final}}\right)$ ;
    return  $s_{best}$ ;
```

Algoritme 3: *Ruin*-methode gebaseerd op het SISRs-algoritme (Christiaens & Vanden Berghe, 2020)

Input: Oplossing $s = \{T, A\}$ met T de verzameling ritten en A de *absent*-verzameling

Output: Geruïneerde oplossing s

procedure *ruin*(s):

```

 $\ell_s^{\max}, k_s^{\max}, k_s \leftarrow$  berekend volgens Eq. 5-7 (Christiaens & Vanden Berghe, 2020);
 $t_s^{\text{seed}} \leftarrow \text{randomNietLegeRit}(s)$ ;
 $c_s^{\text{seed}} \leftarrow \text{randomActieVanRit}(s, t_s^{\text{seed}})$ ;
 $R \leftarrow \emptyset$ ;
for  $c \in \text{krijgBuur}(c_{\text{seed}})$  and  $|R| < k_s$  do
    if  $c \notin A$  and  $t \notin R$  then
         $c_t^* \leftarrow c$ ;
         $\ell_t^{\max}, \ell_t \leftarrow$  berekend volgens Eq. 8 en 9 (Christiaens & Vanden Berghe, 2020);
         $\ell_t \leftarrow \max(1, \lfloor \ell_t/2 \rfloor)$ ;
         $A \leftarrow A \cup \text{verwijderActiesEnGeassocieerdeActies}(t, \ell_t, c_t^*)$ ;
         $R \leftarrow R \cup \{t\}$ ;
return  $s$ ;

```

Algoritme 4: *Recreate*-stap

Input: Geruïneerde oplossing $s = \{T, A\}$ met T de verzameling ritten en A de *absent*-verzameling

Output: Aangepaste oplossing s na recreate-stap

procedure *recreate*(s):

```

    sorteer( $A$ );
    for  $a \in A$  do
         $a_{\text{ophalen}}, a_{\text{afzetten}} \leftarrow \text{krijgGeassocieerdeActies}(a)$ ;
         $P_{\text{ophalen}} \leftarrow \text{null}$ ,  $P_{\text{afzetten}} \leftarrow \text{null}$ ;
         $\text{beste\_kost} \leftarrow \infty$ ,  $\text{beste\_rit} \leftarrow \text{null}$ ;
        for  $t \in T$  do
            for  $P_{t,\text{ophalen}} \in \text{posities}(t)$  do
                if not invoegenFeasible( $t, a_{\text{ophalen}}, P_{t,\text{ophalen}}$ ) then
                     $\text{continue}$ ;
                 $\text{kostInvoegenOphalen} \leftarrow \text{kostInvoegen}(t, a_{\text{ophalen}}, P_{t,\text{ophalen}})$ ;
                for  $P_{t,\text{afzetten}} \in \text{elkePositiesNa}(P_{t,\text{ophalen}}, t)$  do
                    if  $\mathcal{U}(0, 1) < 1 - \beta$  then
                         $\text{continue}$ ;
                    if not capaciteitenFeasible( $t, P_{t,\text{ophalen}}, P_{t,\text{afzetten}}$ ) then
                         $\text{continue}$ ;
                    if not invoegenFeasible( $t, a_{\text{afzetten}}, P_{t,\text{afzetten}}$ ) then
                         $\text{continue}$ ;
                     $\text{kostInvoegenAfzetten} \leftarrow \text{kostInvoegen}(t, a_{\text{afzetten}}, P_{t,\text{afzetten}})$ ;
                     $\text{kost} \leftarrow \text{kostInvoegenOphalen} + \text{kostInvoegenAfzetten}$ ;
                    if isLegerit( $t$ ) then
                         $\text{kost} \leftarrow 0$ ;
                    if  $P_{\text{ophalen}} = \text{null}$  or  $P_{\text{afzetten}} = \text{null}$  or  $\text{kost} < \text{beste\_kost}$  or ( $\text{kost} = \text{beste\_kost}$  and  $\mathcal{U}(0, 1) < 0.5$ ) then
                         $\text{beste\_kost} \leftarrow \text{kost}$ ;
                         $\text{beste\_rit} \leftarrow t$ ;
                         $P_{\text{ophalen}} \leftarrow P_{t,\text{ophalen}}$ ;
                         $P_{\text{afzetten}} \leftarrow P_{t,\text{afzetten}}$ ;
            if  $P_{\text{ophalen}} = \text{null}$  or  $P_{\text{afzetten}} = \text{null}$  then
                 $\text{continue}$ ;
         $\text{insert}(a_{\text{ophalen}}, a_{\text{afzetten}})$  at  $(P_{\text{ophalen}}, P_{\text{afzetten}})$  in  $\text{beste\_rit}$ ;
         $A \leftarrow A \setminus \{a_{\text{ophalen}}, a_{\text{afzetten}}\}$ ;

```

Algoritme 5: *Compress schedule*-methode

Input: Huidige oplossing $s_{lokaal} = \{T, A\}$ met T de verzameling ritten, A de *absent*-verzameling en met planning Σ

Output: Oplossing aangevuld met samengedrukte planning Σ' en dalend gesorteerde rittijden R

```

procedure compressSchedule( $s$ ):
     $\Sigma' \leftarrow \Sigma$ ;
     $R \leftarrow []$ 
    for  $t \in T$  do
         $acties \leftarrow \text{actiesInRit}(s, t)$ ;
         $\text{maxShift}_{\text{volgend}} \leftarrow 0$ ;
        foreach  $i \in \text{reverse}(acties)$  do
             $\Sigma'.\text{update}(\text{maxShift}_i)$  op basis van  $\text{maxShift}_{\text{volgend}}$ ;
             $\text{maxShift}_{\text{volgend}} \leftarrow \text{maxShift}_i$ ;
        foreach  $j \in acties$  do
            if isPickup( $j$ ) then
                 $\text{shift}(j, \text{maxShift}_j, \Sigma')$ ;
                 $\text{propageerNaarVolgendeActies}(\text{shift}_j, \Sigma')$ 
        foreach  $a \in t$  met isOphaalactie( $a$ ) do
             $a.\text{afzetten} \leftarrow \text{krijgGeassocieerdeActie}(a)$ ;
             $\tau_{\text{ophalen}} = a_{\text{start}} + a_{\text{service}}$ ;
             $\tau_{\text{afzetten}} = a.\text{afzetten}_{\text{start}} + a.\text{afzetten}_{\text{service}}$ ;
             $\text{rittijd} \leftarrow \tau_{\text{afzetten}} - \tau_{\text{ophalen}}$ ;
             $R.\text{push}(\text{rittijd})$ ;
     $\text{sorteer\_dalend}(R)$ ;
     $\text{toevoegenSamengedruktePlanning}(s, \Sigma', R)$ 

```

Algoritme 6: SMM-TFE heuristiek

Input: Initiële oplossing $s_{lokaal} = \{T, A\}$ met T de verzameling ritten en A de *absent*-verzameling

Output: Verbeterde oplossing s_{best}

procedure localSearchWithFleetExpansion(s_{lokaal}):

```

 $\mathcal{T} \leftarrow \mathcal{T}_0$ ;
 $s_{best} \leftarrow s_{lokaal}$ ;
fractie_tijd_zonder_virtuele_voertuigen  $\leftarrow 5\% \cdot \text{totale\_tijd}$ ;
interval_planning_initialized  $\leftarrow \text{false}$ ;
while tijdslimiet niet bereikt do
     $s_{nieuw} \leftarrow s_{lokaal}$ ;
    if fractie_tijd_zonder_virtuele_voertuigen voorbij then
        if not interval_planning_initialized and not feasible_without_virtual_voertuigen
            then
                 $s_{nieuw} \cdot \text{toevoegenVirtueelVoertuigPerLeerlingInAfwezigeVerzameling}(|A|/2, C_{\text{virtueel}})$ ;

                 $N_{\text{resterende\_slices}} \leftarrow \text{Aantal\_virtuele\_voertuigen} + 1$ ;
                tijd_per_slice  $\leftarrow \text{krijgTijdResterend}() / N_{\text{resterende\_slices}}$ ;
                interval_planning_initialized  $\leftarrow \text{true}$ ;

        if interval_planning_initialized then
            remove_vehicle_bool  $\leftarrow (|A(s)| = 0)$  and ( $\mathcal{U}(0, 1) < 0.05$ );
            if
                (remove_vehicle_bool or tijd slice is voorbij) and ( $\text{Aantal\_virtuele\_voertuigen} > 0$ ) then
                     $s_{nieuw} \cdot \text{removeVirtualVehicle}()$ ;
                     $N_{\text{resterende\_slices}} \leftarrow N_{\text{resterende\_slices}} - 1$ ;

     $s_{nieuw} \leftarrow \text{ruin}(s_{nieuw})$ ;
     $s_{nieuw} \leftarrow \text{recreate}(s_{nieuw})$ ;
     $s_{nieuw} \leftarrow \text{compress\_rittijden}(s_{nieuw})$ ;
    if  $|A_{nieuw}| < |A_{best}|$  or isBeter( $s_{nieuw}, s_{best}$ ) or removed_voertuig_this_iteration
        then
             $s_{best} \leftarrow s_{nieuw}$ ;
    geaccepteerd  $\leftarrow |A_{nieuw}| < |A_{lokaal}|$  or isBeter( $s_{nieuw}, s_{lokaal}$ ) or removed_voertuig_this_iteration
    if not geaccepteerd then
        geaccepteerd  $\leftarrow \text{Acceptatiecriterium zoals in Algoritme 1}$ 
    if geaccepteerd then
         $s_{lokaal} \leftarrow s_{nieuw}$ 
     $\mathcal{T} \leftarrow \min\left(\mathcal{T}_0, \mathcal{T}_0 \cdot \left(\frac{\text{tijd over (sec)}}{\text{totale tijd (sec)}}\right)^5 + \mathcal{T}_{\text{final}}\right)$ ;
return  $s_{best}$ ;

```

Algoritme 7: *Fleet minimisation*-fase van de FSM-FE heuristiek

Input: Initiële oplossing $s_{lokaal} = \{T, A\}$ met T de verzameling ritten en A de *absent*-verzameling. De input vloot is aangevuld met extra voertuigen zodat elke leerling één voertuig heeft.

Output: Beste gevonden oplossing s_{best} met kleinste aantal extra voertuigen

procedure fleetMinimisation (s_{lokaal}):

```

 $s_{best} \leftarrow s_{lokaal}$ ;
for  $a \in \text{acties}$  do
     $\text{abs}_a \leftarrow 0$ ;
while tijdslimiet voor minimaliseren voertuigen niet bereikt do
     $s_{nieuw} \leftarrow \text{ruin}(s_{lokaal})$ ;
     $s_{nieuw} \leftarrow \text{recreate}(s_{nieuw})$ ;
     $s_{nieuw} \leftarrow \text{compress\_rittijden}(s_{nieuw})$ ;
    if  $|A(s_{nieuw})| < |A(s_{lokaal})|$  or  $\text{sumAbs}(A(s_{nieuw})) < \text{sumAbs}(A(s_{lokaal}))$  then
         $s_{lokaal} \leftarrow s_{nieuw}$ ;
    if  $A(s_{nieuw}) = \emptyset$  then
         $s_{best} \leftarrow s_{nieuw}$ ;
        if aantal extra voertuigen  $> 0$  then
             $t_{\min} \leftarrow \text{ritMetKleinsteSumAbs}(s_{nieuw})$ ;
            maakRitLeeg( $t_{\min}, s_{nieuw}$ );
             $t_{\text{extra}} \leftarrow \text{laatsteExtraVoertuig}(s_{nieuw})$ ;
            switchInhoudRit( $t_{\text{extra}}, t_{\min}, s_{nieuw}$ );
            verwijderLaatsteExtraVoertuig( $t_{\text{extra}}, s_{nieuw}$ );
             $s_{lokaal} \leftarrow s_{nieuw}$ ;
        else
            break;
    for  $c \in A(s_{nieuw})$  do
         $\text{abs}_c \leftarrow \text{abs}_c + 1$ ;
return  $s_{best}$ ;

```

Bijlage D

Overige resultaten voor de arrondissementen met de hoofdheuristiek

Arrondissement	Stad	School
arron8_Veurne_WV1_U268_K14_S6_T18_2023 Obj. [min]: 102.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 218.8	city1_Koksijde_WV1_U234_K12_S3_T18_2023 Obj. [min]: 104.8 Ontbr. leerl.: 2 $T_{\text{gem.}}[s]$: 191.0	school189_Koksijde_U49_K2_S1_T7_2023 Obj. [min]: 91.3 Ontbr. leerl.: 8 $T_{\text{gem.}}[s]$: 40.0
		school365_Koksijde_U106_K6_S1_T17_2023 Obj. [min]: 111.2 Ontbr. leerl.: 4 $T_{\text{gem.}}[s]$: 86.5
		school428_Koksijde_U79_K4_S1_T14_2023 Obj. [min]: 98.5 Ontbr. leerl.: 6 $T_{\text{gem.}}[s]$: 64.5
	city2_Nieuwpoort_WV1_U34_K2_S1_T6_2023 Obj. [min]: 89.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 27.8	school366_Nieuwpoort_U34_K2_S1_T6_2023 Obj. [min]: 89.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 27.8

Tabel D.1 Resultaten van de SMM-FF heuristiek voor het arrondissement Veurne (vijf onafhankelijke uitvoeringen)

Arrondissement	Stad	School
arron7_Tielt_WV1_U151_K8_S2_T2_2023 Obj. [min]: 69.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 123.3	city17_Tielt_WV1_U151_K8_S2_T2_2023 Obj. [min]: 71.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 123.3	school210_Tielt_U91_K5_S1_T0_2023 Obj. [min]: 55.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 74.3
		school373_Tielt_U60_K3_S1_T2_2023 Obj. [min]: 105.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 49.0

Tabel D.2 Resultaten van de SMM-FF heuristiek voor het arrondissement Tielt (vijf onafhankelijke uitvoeringen)

Tabel D.3 Resultaten van de SMM-FF heuristiek voor het arrondissement Brugge (vijf onafhankelijke uitvoeringen)

Arrondissement	Stad	School
arron1_Brugge_WV1_U2205_K116.S32.T29_2023 Obj. [min]: 99.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 1802.9	city22_Brugge_WV1_U1769_K94.S23.T29_2023 Obj. [min]: 99.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 1445.7	school24_Brugge_U25_K1.S1.T11_2023 Obj. [min]: 90.0 Ontbr. leerl.: 15 $T_{\text{gem.}}[s]$: 20.4
		school28_Brugge_U91_K5.S1.T3_2023 Obj. [min]: 88.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 74.3
		school29_Brugge_U163_K9.S1.T23_2023 Obj. [min]: 96.0 Ontbr. leerl.: 21 $T_{\text{gem.}}[s]$: 133.1
		school41_Brugge_U48_K2.S1.T8_2023 Obj. [min]: 114.3 Ontbr. leerl.: 12 $T_{\text{gem.}}[s]$: 39.2
		school42_Brugge_U62_K3.S1.T13_2023 Obj. [min]: 99.2 Ontbr. leerl.: 15 $T_{\text{gem.}}[s]$: 50.6
		school50_Brugge_U46_K2.S1.T10_2023 Obj. [min]: 99.2 Ontbr. leerl.: 21 $T_{\text{gem.}}[s]$: 37.6
		school66_Brugge_U16_K1.S1.T5_2023 Obj. [min]: 89.2 Ontbr. leerl.: 5 $T_{\text{gem.}}[s]$: 13.1
		school67_Brugge_U38_K2.S1.T12_2023 Obj. [min]: 99.0 Ontbr. leerl.: 10 $T_{\text{gem.}}[s]$: 31.0
		school172_Brugge_U106_K6.S1.T15_2023 Obj. [min]: 126.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 86.5
		school173_Brugge_U77_K4.S1.T5_2023 Obj. [min]: 94.5 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 62.9
		school178_Brugge_U71_K4.S1.T3_2023 Obj. [min]: 85.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 58.0
		school179_Brugge_U93_K5.S1.T7_2023 Obj. [min]: 105.7 Ontbr. leerl.: 10 $T_{\text{gem.}}[s]$: 75.9
		school180_Brugge_U36_K2.S1.T2_2023 Obj. [min]: 98.0 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 29.4
		school181_Brugge_U36_K2.S1.T8_2023 Obj. [min]: 101.0 Ontbr. leerl.: 8 $T_{\text{gem.}}[s]$: 29.4
		school182_Brugge_U62_K3.S1.T6_2023 Obj. [min]: 121.2 Ontbr. leerl.: 3 $T_{\text{gem.}}[s]$: 50.6
		school269_Brugge_U50_K3.S1.T5_2023 Obj. [min]: 98.7 Ontbr. leerl.: 3 $T_{\text{gem.}}[s]$: 40.8
		school270_Brugge_U200_K11.S1.T11_2023 Obj. [min]: 102.5 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 163.3
		school362_Brugge_U97_K5.S1.T6_2023 Obj. [min]: 109.8 Ontbr. leerl.: 5 $T_{\text{gem.}}[s]$: 79.2
		school363_Brugge_U249_K13.S1.T7_2023 Obj. [min]: 90.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 203.3
		school429_Brugge_U19_K1.S1.T2_2023 Obj. [min]: 97.0 Ontbr. leerl.: 6 $T_{\text{gem.}}[s]$: 15.5
		school430_Brugge_U120_K6.S1.T7_2023 Obj. [min]: 96.5 Ontbr. leerl.: 2 $T_{\text{gem.}}[s]$: 98.0
		school448_Brugge_U1_K1.S1.T2_2023 Obj. [min]: 25.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 0.8
		school449_Brugge_U63_K3.S1.T12_2023 Obj. [min]: 104.5 Ontbr. leerl.: 19 $T_{\text{gem.}}[s]$: 51.4

Tabel D.3 Resultaten van de SMM-FF heuristiek voor het arrondissement Brugge (vijf onafhankelijke uitvoeringen)

Arrondissement	Stad	School
	city23_Knokke-Heist.WV1 U31_K2_S1_T3_2023 Obj. [min]: 71.0 Ontbr. leerl.: 0 $T_{\text{gem. [s]}}$: 25.3	school187_Knokke-Heist_U31_K2_S1_T3_2023 Obj. [min]: 71.0 Ontbr. leerl.: 0 $T_{\text{gem. [s]}}$: 25.3
	city24_Beernem.WV1 U55_K3_S1_T6_2023 Obj. [min]: 105.0 Ontbr. leerl.: 10 $T_{\text{gem. [s]}}$: 44.9	school431_Beernem_U55_K3_S1_T6_2023 Obj. [min]: 103.3 Ontbr. leerl.: 9 $T_{\text{gem. [s]}}$: 44.9
	city25_Blankenberge.WV1_U70_K4_S2_T3_2023 Obj. [min]: 57.3 Ontbr. leerl.: 0 $T_{\text{gem. [s]}}$: 57.1	school186_Blankenberge_U70_K4_S1_T3_2023 Obj. [min]: 58.0 Ontbr. leerl.: 0 $T_{\text{gem. [s]}}$: 57.1
	city18_Torhout.WV1 U207_K11_S3_T10_2023 Obj. [min]: 91.3 Ontbr. leerl.: 0 $T_{\text{gem. [s]}}$: 169.0	school174_Torhout_U42_K2_S1_T1_2023 Obj. [min]: 100.2 Ontbr. leerl.: 2 $T_{\text{gem. [s]}}$: 34.3
		school175_Torhout_U58_K3_S1_T2_2023 Obj. [min]: 96.8 Ontbr. leerl.: 0 $T_{\text{gem. [s]}}$: 47.3
		school360_Torhout_U107_K6_S1_T9_2023 Obj. [min]: 100.2 Ontbr. leerl.: 9 $T_{\text{gem. [s]}}$: 87.4
	city19_Zedelgem.WV1 U73_K4_S2_T7_2023 Obj. [min]: 108.2 Ontbr. leerl.: 1 $T_{\text{gem. [s]}}$: 59.6	school184_Zedelgem_U65_K3_S1_T5_2023 Obj. [min]: 98.7 Ontbr. leerl.: 4 $T_{\text{gem. [s]}}$: 53.1
		school271_Zedelgem_U8_K1_S1_T4_2023 Obj. [min]: 96.5 Ontbr. leerl.: 1 $T_{\text{gem. [s]}}$: 6.5

Tabel D.4 Resultaten van de SMM-FF heuristiek voor het arrondissement Kortrijk (vijf onafhankelijke uitvoeringen)

Arrondissement	Stad	School
arron4_Kortrijk_WV1_U2106_K112_S23_T21_2023 Obj. [min]: 91.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 1721.9	city8_Menen_WV1_U125_K7_S1_T3_2023 Obj. [min]: 50.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 102.0	school196_Menen_U125_K7_S1_T3_2023 Obj. [min]: 48.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 102.0
	city9_Wevelgem_WV1_U84_K4_S1_T3_2023 Obj. [min]: 55.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 68.6	school197_Wevelgem_U84_K4_S1_T3_2023 Obj. [min]: 55.5 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 68.6
	city10_Kortrijk_WV1_U1124_K60_S11_T16_2023 Obj. [min]: 86.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 918.2	school3_Kortrijk_U248_K13_S1_T11_2023 Obj. [min]: 87.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 202.5
		school31_Kortrijk_U93_K5_S1_T3_2023 Obj. [min]: 58.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 75.9
		school59_Kortrijk_U189_K10_S1_T5_2023 Obj. [min]: 93.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 154.3
		school62_Kortrijk_U37_K2_S1_T6_2023 Obj. [min]: 110.2 Ontbr. leerl.: 7 $T_{\text{gem.}}[s]$: 30.2
		school63_Kortrijk_U14_K1_S1_T4_2023 Obj. [min]: 93.3 Ontbr. leerl.: 5 $T_{\text{gem.}}[s]$: 11.4
		school274_Kortrijk_U93_K5_S1_T4_2023 Obj. [min]: 103.7 Ontbr. leerl.: 3 $T_{\text{gem.}}[s]$: 75.9
		school275_Kortrijk_U101_K5_S1_T6_2023 Obj. [min]: 99.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 82.5
		school434_Kortrijk_U173_K9_S1_T6_2023 Obj. [min]: 67.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 141.2
		school192_Kortrijk_U39_K2_S1_T3_2023 Obj. [min]: 103.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 31.8
		school193_Kortrijk_U32_K2_S1_T3_2023 Obj. [min]: 75.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 26.1
		school194_Kortrijk_U105_K6_S1_T6_2023 Obj. [min]: 98.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 85.7
	city11_Zwevegem_WV1_U102_K5_S1_T3_2023 Obj. [min]: 75.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 83.3	school195_Zwevegem_U102_K5_S1_T3_2023 Obj. [min]: 76.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 83.3
	city12_Anzegem_WV1_U20_K2_S2_T5_2023 Obj. [min]: 93.5 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 16.3	school13_Anzegem_U12_K1_S1_T4_2023 Obj. [min]: 113.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 9.8
		school367_Anzegem_U8_K1_S1_T1_2023 Obj. [min]: 98.8 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 6.5

Tabel D.4 Resultaten van de SMM-FF heuristiek voor het arrondissement Kortrijk (vijf onafhankelijke uitvoeringen)

Arrondissement	Stad	School
	city13_Deerlijk_WV1_U247_K13_S4_T4_2023 Obj. [min]: 73.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 201.7	school10_Deerlijk_U66_K4_S1_T3_2023 Obj. [min]: 79.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 53.9
		school11_Deerlijk_U23_K1_S1_T3_2023 Obj. [min]: 101.7 Ontbr. leerl.: 3 $T_{\text{gem.}}[s]$: 18.8
		school201_Deerlijk_U96_K5_S1_T3_2023 Obj. [min]: 92.7 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 78.4
		school202_Deerlijk_U62_K3_S1_T4_2023 Obj. [min]: 108.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 50.6
	city14_Waregem_WV1_U404_K21_S3_T9_2023 Obj. [min]: 80.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 329.9	school204_Waregem_U158_K8_S1_T4_2023 Obj. [min]: 61.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 129.0
		school368_Waregem_U30_K2_S1_T4_2023 Obj. [min]: 103.8 Ontbr. leerl.: 3 $T_{\text{gem.}}[s]$: 24.5
		school369_Waregem_U216_K11_S1_T9_2023 Obj. [min]: 89.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 176.3

Arrondissement	Stad	School
arron2_Diksmuide_WV1_U316_K16_S5_T18_2023 Obj. [min]: 99.3 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 258.0	city5_Diksmuide_WV1_U117_K6_S2_T5_2023 Obj. [min]: 65.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 95.5	school177_Diksmuide_U94_K5_S1_T5_2023 Obj. [min]: 69.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 76.7
		school426_Diksmuide_U23_K1_S1_T5_2023 Obj. [min]: 88.0 Ontbr. leerl.: 2 $T_{\text{gem.}}[s]$: 18.8
	city6_Houthulst_WV1_U114_K6_S2_T16_2023 Obj. [min]: 105.2 Ontbr. leerl.: 10 $T_{\text{gem.}}[s]$: 93.1	school176_Houthulst_U29_K2_S1_T5_2023 Obj. [min]: 100.2 Ontbr. leerl.: 1 $T_{\text{gem.}}[s]$: 23.7
		school361_Houthulst_U85_K4_S1_T15_2023 Obj. [min]: 101.5 Ontbr. leerl.: 16 $T_{\text{gem.}}[s]$: 69.4
	city20_Koekelare_WV1_U85_K4_S1_T3_2023 Obj. [min]: 83.0 Ontbr. leerl.: 8 $T_{\text{gem.}}[s]$: 69.4	school185_Koekelare_U85_K4_S1_T3_2023 Obj. [min]: 82.8 Ontbr. leerl.: 8 $T_{\text{gem.}}[s]$: 69.4

Tabel D.5 Resultaten van de SMM-FF heuristiek voor het arrondissement Diksmuide (vijf onafhankelijke uitvoeringen)

Arrondissement	Stad	School
arron5_Oostende_WV1_U819_K43_S9_T15_2023 Obj. [min]: 81.5 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 668.9	city26_Middelkerke_WV1_U236_K12_S3_T8_2023 Obj. [min]: 77.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 192.7	school183_Middelkerke_U79_K4_S1_T4_2023 Obj. [min]: 62.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 64.5
		school272_Middelkerke_U79_K4_S1_T8_2023 Obj. [min]: 92.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 64.5
		school364_Middelkerke_U78_K4_S1_T8_2023 Obj. [min]: 99.0 Ontbr. leerl.: 2 $T_{\text{gem.}}[s]$: 63.7
	city27_Oostende_WV1_U510_K27_S5_T15_2023 Obj. [min]: 77.3 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 416.4	school44_Oostende_U69_K4_S1_T13_2023 Obj. [min]: 102.0 Ontbr. leerl.: 4 $T_{\text{gem.}}[s]$: 56.3
		school45_Oostende_U50_K3_S1_T8_2023 Obj. [min]: 100.2 Ontbr. leerl.: 4 $T_{\text{gem.}}[s]$: 40.8
		school188_Oostende_U98_K5_S1_T3_2023 Obj. [min]: 45.5 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 80.0
		school273_Oostende_U118_K6_S1_T10_2023 Obj. [min]: 95.8 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 96.3
		school433_Oostende_U175_K9_S1_T8_2023 Obj. [min]: 64.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 142.9
	city21_Ichtegem_WV1_U73_K4_S1_T3_2023 Obj. [min]: 76.0 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 59.6	school432_Ichtegem_U73_K4_S1_T3_2023 Obj. [min]: 75.2 Ontbr. leerl.: 0 $T_{\text{gem.}}[s]$: 59.6

Tabel D.6 Resultaten van de SMM-FF heuristiek voor het arrondissement Oostende (vijf onafhankelijke uitvoeringen)

Bijlage E

Bronnen voor de tijdsvensters van de opgezochte scholen

De volgende bronnen werden gebruikt voor het verzamelen van tijdsvensters van 100 Vlaamse en Brusselse scholen voor buitengewoon onderwijs:

- aan zee — De Strandloper BuBaO, I. (s.d.). *Openingsuren*. Geraadpleegd op 22 september 2025, van <https://www.inspirant.be/kind-en-jongere/onderwijs/inspirant-aan-zee-de-strandloper-bubao/praktische-informatie/schoolorganisatie/openingsuren>
- Basisschool Hoelahoep. (s.d.). *Praktisch*. Geraadpleegd op 22 september 2025, van <https://www.hoelahoepp.be/schoolinfo/praktisch/>
- Berkenbeek Buitengewoon Onderwijs. (s.d.). *Schooltijden*. Geraadpleegd op 22 september 2025, van <https://www.berkenbeek.be/schooltijden>
- Boemerang, D. (s.d.). *Schoolbrochure*. Geraadpleegd op 22 september 2025, van <https://sites.google.com/a/dekubus-bree.be/boemerang/schoolbrochure>
- Brug, D. (s.d.). *Schooluren en vakanties*. Geraadpleegd op 22 september 2025, van <https://debrugberingen.be/schooluren-en-vakanties>
- Brugje, '. (s.d.). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://www.brugje.be/informatie/wie-zijn-we/schooluren>
- bsbo de drempel. (s.d.). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://www.de-drempel.be/nuttige-info/>
- BSBO Heideveld. (s.d.). *Praktische info*. Geraadpleegd op 22 september 2025, van <https://bsboheideveld.be/praktische-info/>
- BuBaO. (s.d.). *Schoolreglement 2025-2026*. Katholiek Onderwijs Vlaanderen. Geraadpleegd op 22 september 2025, van https://bubaoterleie.weebly.com/uploads/1/0/9/1/109138979/schoolreglement_2025-2026.pdf
- BuBaO Bubao. (s.d.). *Reglement en kalender*. Gesubsidieerde Vrije Lagere school VZW Levenslust Buitengewoon Basis Onderwijs. Geraadpleegd op 22 september 2025, van <https://bubao.be/reglement-en-kalender/>
- BuBaO Het Vlot. (s.d.). *School hours of Het Vlot*. Geraadpleegd op 22 september 2025, van <https://www.hetvlot.be/praktisch/schooluren>
- BuBaO Klimop. (s.d.). *BuBaO Klimop Diksmuide*. Geraadpleegd op 22 september 2025, van <https://www.klimopdiksmuide.be/index.html>
- BuBaO Sint Rafaël. (s.d.). *Schoolreglement BuBaO Sint Rafaël Type Basisaanbod Type 9*. Geraadpleegd op 22 september 2025, van https://www.sint-rafael.be/images/pdf/schoolreglement_2023-2024.pdf

- buitengewoon onderwijs 'Ter Elst', V. G. (s.d.). *Schoolreglement*. Geraadpleegd op 22 september 2025, van <https://schoolterelst.be/wp/schoolreglement/>
- Buitengewoon Secundair Onderwijs VTS 3 - OV 3. (s.d.). *Schoolreglement 2024-2025*. Geraadpleegd op 22 september 2025, van https://drive.google.com/file/d/1pi1GF5d4IZKA3FxLNOOnQEPEaU60PZKjz/view?usp=embed_facebook
- BULO Het Veer. (s.d.). *Onze afspraken*. Geraadpleegd op 22 september 2025, van <https://bulohetveer.weebly.com/onze-afspraken.html>
- BuLO Klim-Op. (s.d.). *Infobrochure met schoolreglement*. Geraadpleegd op 22 september 2025, van <https://drive.google.com/file/d/1w6p0X9ZJWspNGrOqSDsplCTlkDzA7Iir/view>
- BuSO De Hoge Kouter Kortrijk. (s.d.). *Schoolgids met schoolreglement*. Geraadpleegd op 22 september 2025, van <https://www.hogekouter.be/onze-schoolafspraken>
- BuSO Ganspoel (OV1 en OV4 type 6) Koninklijk Instituut Woluwe & BuSO (OV4 type 9) Vestigingsplaats Sint-Lambrechts-Woluwe. (s.d.). *Schoolreglement*. Geraadpleegd op 22 september 2025, van <http://www.kiwoluwe.be/wp-content/uploads/Schoolreglement-BuSO-vestiging-Sint-Lambrechts-Woluwe-2025-2026-.pdf>
- BuSO SJABI. (s.d.). *'Buitengewoon' goed je beroep leren*. Geraadpleegd op 22 september 2025, van https://buso.sjabi.be/sites/busofiles/2021-12/2021-2022%20Sjabi%20BUSO%20brochure_definitief.0.pdf
- BuSO Tongelsbos. (s.d.). *Schooltijden*. Geraadpleegd op 22 september 2025, van <https://buso.tongelsbos.be/info/schooltijden>
- campus Buggenhout, K. (s.d.). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://oost-vlaanderen.be/leren/onderwijs/overzicht-provincialescholen/bijzonder-lager-onderwijs/kiempunt-campus-buggenhout/praktisch/schooluren.html>
- de Kade — BuBaO de Berkjes. (s.d.). *Schoolreglement BuBaO de Berkjes 2025-2026*. Geraadpleegd op 22 september 2025, van <https://shuttle-storage.s3.amazonaws.com/dekade6/De%20Kade/Onderwijs/Berkjes/Schoolreglement%20BuBaO%20de%20Berkjes%202025-26.pdf>
- DE VARENS GO! (s.d.). *DE VARENS GO! school voor buitengewoon secundair onderwijs 2025-2026*. Geraadpleegd op 22 september 2025, van https://schoolreglement.g-o.be/r/25_27284-1/pdf/3?year=2025
- De Vloedlijn & De Zandkorrel. (s.d.). *Openingsuren*. Geraadpleegd op 22 september 2025, van <https://www.devloedlijn.be/praktische-info/openingsuren>
- De waterlelie. (s.d.). *Schooluren*. De Waterlelie — Vrije Basisschool voor Buitengewoon Onderwijs. Geraadpleegd op 22 september 2025, van <https://www.dewaterlelieschool.be/schooluren/>
- Ennea. (s.d.). *Dagstructuur*. Geraadpleegd op 22 september 2025, van <https://www.ennea.school/dagstructuur/>
- Gemeentelijke Basisschool Schransdries. (s.d.). *Lesuren & toezicht*. Geraadpleegd op 22 september 2025, van <https://www.gbs-schransdries.be/praktisch/lesuren>
- GO! athena OV4 Kortrijk. (s.d.). *GO! athena OV4 Kortrijk 2025-2026*. Geraadpleegd op 22 september 2025, van https://schoolreglement.g-o.be/r/26_143818-1/pdf/1?year=2025
- GO! basisschool Spronk. (s.d.). *Info spronk*. Geraadpleegd op 22 september 2025, van <https://basisschool.campusspronk.be/info/#praktisch>
- GO! basisschool voor buitengewoon onderwijs De Bloesem Sint-Truiden — Scholengroep WijS. (s.d.). *schoolreglement buitengewoon basisonderwijs: 2025-2026*. Geraadpleegd op 22 september 2025, van <https://www.debloesem.be/wp-content/uploads/2025/07/schoolreglement-De-Bloesem-2025-2026.pdf>
- GO! BSBO De Brug. (s.d.-a). *Schoolreglement*. Geraadpleegd op 22 september 2025, van <https://basisaanbod.bsodebrug.be/informatie/schoolreglement/>
- GO! BSBO De Brug. (s.d.-b). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://type9.bsodebrug.be/informatie/schooluren/>
- GO! BuSO Parkaan. (s.d.). *GO! BuSO Parkaan 2025-2026*. Geraadpleegd op 22 september 2025, van <https://pargaan.be/wp-content/uploads/2025/08/Schoolreglement-2025-2026.pdf>

- GO! Inclusiecampus Wemmel. (s.d.). *Info*. GO! Inclusiecampus Wemmel - GO! Scholengroep UNIK - inclusief onderwijs BuO (Basisaanbod en type 3) en BaO 2024-2025. Geraadpleegd op 22 september 2025, van https://schoolreglement.g-o.be/r/9_3327-1/18/info/?year=2024
- GO! mpi kompas. (s.d.). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://www.mpikompas.be/nieuw/schooluren/>
- GO! MPI Sterrebos Roeselare. (s.d.). *Schooluren MPI Sterrebos*. Geraadpleegd op 22 september 2025, van <https://mpigo.campussterrebos.be/campus-sterrebos/schooluren/>
- GO! School voor buitengewoon onderwijs Groenlaar. (s.d.). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://sbso.groenlaar.be/schooluren/>
- GO! Buitengewoon secundair onderwijs Woudlucht: 2025-2026. (s.d.). Geraadpleegd op 22 september 2025, van https://schoolreglement.g-o.be/r/11_27243-1/pdf/17?year=2025
- GO! IBSO De Horizon De Beroepsschool — Passtel: 2025-2026. (s.d.). Geraadpleegd op 22 september 2025, van https://schoolreglement.g-o.be/r/19_146936-1/pdf/2?year=2025
- GO! IBSO De Horizon De Beroepsschool — Passtel Vestigingsplaats GO! Atheneum Aalst: 2025-2026. (s.d.). Geraadpleegd op 22 september 2025, van https://ibsodehorizon-my.sharepoint.com/personal/femke.borremans.ibsodehorizon.be/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Ffemke%5Fborremans%5Fibsodehorizon%5Fbe%2FDocuments%2FPasstel%20%28OV4T9%29%2FArchief%20algemeen%20%28tot%20sept%202025%29%2F01%2E%20Algemeen%20%2D%20nieuwe%20structuur%2F01%2E%20Algemeen%20beleid%2F02%2E%20Organisatie%2F2025%2D2026%20GO%21%20Atheneum%20Aalst%20Passtel%2Epdf&parent=%2Fpersonal%2Ffemke%5Fborremans%5Fibsodehorizon%5Fbe%2FDocuments%2FPasstel%20%28OV4T9%29%2FArchief%20algemeen%20%28tot%20sept%202025%29%2F01%2E%20Algemeen%20%2D%20nieuwe%20structuur%2F01%2E%20Algemeen%20beleid%2F02%2E%20Organisatie%2F2025%2D2026&ga=1
- GO! IBSO De Horizon De Beroepsschool — Passtel Vestigingsplaats GO! Middenschool Aalst: 2025-2026. (s.d.). Geraadpleegd op 22 september 2025, van https://ibsodehorizon-my.sharepoint.com/personal/femke.borremans.ibsodehorizon.be/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Ffemke%5Fborremans%5Fibsodehorizon%5Fbe%2FDocuments%2FPasstel%20%28OV4T9%29%2FArchief%20algemeen%20%28tot%20sept%202025%29%2F01%2E%20Algemeen%20%2D%20nieuwe%20structuur%2F01%2E%20Algemeen%20beleid%2F02%2E%20Organisatie%2F2025%2D2026%20GO%21%20Middenschool%20Aalst%20Passtel%2Epdf&parent=%2Fpersonal%2Ffemke%5Fborremans%5Fibsodehorizon%5Fbe%2FDocuments%2FPasstel%20%28OV4T9%29%2FArchief%20algemeen%20%28tot%20sept%202025%29%2F01%2E%20Algemeen%20%2D%20nieuwe%20structuur%2F01%2E%20Algemeen%20beleid%2F02%2E%20Organisatie%2F2025%2D2026&ga=1
- GO! lagere school voor buitengewoon onderwijs Mikado Maasmechelen (Bosweg 71 3630 Maasmechelen): 2025-2026. (s.d.). Geraadpleegd op 22 september 2025, van https://schoolreglement.g-o.be/r/14_3442-1/pdf/2?year=2025
- GO!Groeicampus Basis. (s.d.). *Groeicampus buitengewoon basisonderwijs 2025-2026*. Geraadpleegd op 22 september 2025, van https://schoolreglement.g-o.be/r/14_3442-1/pdf/2?year=2025
- Ingelmunster, D. Z. — B. O. (s.d.). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://www.dizon-blo.be/praktische-info/schooluren>
- Inspirant aan zee BuSO. (s.d.). *Het schoolreglement*. Geraadpleegd op 22 september 2025, van <https://www.inspirant.be/kind-en-jongere/onderwijs/inspirant-aan-zee-buso-oostduinkerke/praktische-informatie/schoolorganisatie/schoolreglement>
- Kasterlinden buitengewoon onderwijs. (s.d.-a). *Dagverloop en opvang BuBaO*. Geraadpleegd op 22 september 2025, van <https://www.kasterlinden.be/bubao/praktische-info/dagverloop-en-opvang-bubao>
- Kasterlinden buitengewoon onderwijs. (s.d.-b). *Schooluren BuSO 2024-2025*. Geraadpleegd op 22 september 2025, van <https://www.kasterlinden.be/schooluren-buso-2024-2025>
- Katrinahof BuBaO. (s.d.). *Nuttige info*. Geraadpleegd op 22 september 2025, van <https://www.katrinahofscholen.be/bubao/nuttige-info/>

- KBO Kameleon. (s.d.). *Praktische informatie: Info over opvang, bus, maaltijden...* Geraadpleegd op 22 september 2025, van <https://dekameleon.kbo-oudenaarde.be/praktische-informatie>
- Klavertje 3 Bilzen. (s.d.). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://klavertje3bilzen.weebly.com/schooluren.html>
- Klim Op Lokeren. (s.d.). *Schoolreglement GO! basisschool Klim Op Lokeren: 2023-2024*. Geraadpleegd op 22 september 2025, van https://schoolreglement.g-o.be/r/18_3608-1/pdf/7?year=2023
- KOCA VZW. (s.d.). *Wij staan naast je, opdat je op eigen krachten verder kan*. Geraadpleegd op 22 september 2025, van <https://www.kocabasisonderwijs.be/wij-staan-naast-je-opdat-je-op-eigen-krachten-verder-kan>
- Koninklijk Instituut Woluwe. (s.d.). *Praktische info: BuBaO, BuSO en MFC*. Geraadpleegd op 22 september 2025, van <https://www.kiwoluwe.be/praktische-info/>
- Kristus Koning Basisschool & Campus Maria Middelaars Basisschool. (s.d.). *Lesuren*. Geraadpleegd op 22 september 2025, van <https://bulokk.be/onze-school/lesuren/>
- Leerexpert Capitan. (s.d.). *Praktisch basisschool*. Geraadpleegd op 22 september 2025, van <https://leerexpertcapitan.stedelijkonderwijs.be/basisschool/praktisch-basisschool>
- Leerexpert Castor. (s.d.). *Praktische info*. Geraadpleegd op 22 september 2025, van <https://leerexpertcastor.stedelijkonderwijs.be/praktisch/praktische-info>
- Leerexpert De IJssduiker. (s.d.). *Praktische info*. Geraadpleegd op 22 september 2025, van <https://leerexpertdeijssduiker.stedelijkonderwijs.be/praktisch/praktische-info>
- Leerexpert De Novanaut. (s.d.). *Praktische info*. Geraadpleegd op 22 september 2025, van <https://leerexpertdenovanaut.stedelijkonderwijs.be/praktisch/praktische-info>
- Leerexpert Jozef van Poppelstraat. (s.d.). *Praktische info*. Geraadpleegd op 22 september 2025, van <https://leerexpertjozefvanpoppelstraat.stedelijkonderwijs.be/praktisch/praktische-info>
- Leerexpert Kokoen. (s.d.). *Praktisch basisschool*. Geraadpleegd op 22 september 2025, van <https://leerexpertkokoen.stedelijkonderwijs.be/basisschool/praktisch-basisschool>
- Leerexpert Merino. (s.d.). *Praktisch basisschool*. Geraadpleegd op 22 september 2025, van <https://leerexpertmerino.stedelijkonderwijs.be/basisschool/praktisch-basisschool>
- Leerexpert Polaris. (s.d.). *Praktische info*. Geraadpleegd op 22 september 2025, van <https://leerexpertpolaris.stedelijkonderwijs.be/praktisch/praktische-info>
- Leerexpert Schotensesteenweg 256. (s.d.). *Praktische info*. Geraadpleegd op 22 september 2025, van <https://leerexpertschotensesteenweg256.stedelijkonderwijs.be/praktisch/praktische-info>
- MOZA-IK. (s.d.). *Infobrochure: 2025-2026*. Geraadpleegd op 22 september 2025, van <https://www.mozaiik.be/infobrochure/>
- MPI het vindingrijk. (s.d.). *Dagindeling*. Geraadpleegd op 22 september 2025, van <https://mpi-hetvindingrijk.be/dagindeling/>
- Onderwijs, D. G. — S. B. L. (s.d.). *Openingsuren*. Geraadpleegd op 22 september 2025, van <https://ganzeneveer.be/openingsuren/>
- Onze Jeugd. (s.d.). *Dagindeling*. Geraadpleegd op 22 september 2025, van <https://www.onzejeugd.be/dagindeling/>
- Oud-Turnhout, V. B. O. (s.d.). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://ov1.vibo.be/schooluren>
- Provilion ION OV3 — Provinciaal Instituut Individueel Onderwijs Lommel. (s.d.). *Dagindeling ION 2024-2025*. Geraadpleegd op 22 september 2025, van <https://www.provilion.be/nl/dagindeling>
- Rietzang Ter Strepe. (s.d.). *Schoolreglement*. Geraadpleegd op 22 september 2025, van <https://terstrepe.be/wonderwijs.be/onze-school/schoolreglement>
- Saigo Sterrenbos. (s.d.). *Opvang*. Geraadpleegd op 22 september 2025, van <https://www.saigosterrenbos.be/thema/detail/30/opvang>
- SBLO OASE. (s.d.). *School- en busreglement*. Geraadpleegd op 22 september 2025, van <https://sites.google.com/sovilvoorde.be/sblooase/praktische-info/school-en-busreglement>
- Sint-Jozefschool BuBaO. (s.d.). *Verder dan dichtbij! Welkom in de Sint-Jozefschool — Praktisch*. Geraadpleegd op 22 september 2025, van <https://www.sint-jozefschool.be/>

- Sint-Rafaël Buitengewoon Basis Onderwijs. (s.d.). *Ons schoolreglement*. Geraadpleegd op 22 september 2025, van <https://st-raf.com/wp-content/uploads/2025/07/schoolreglement-bubao-sint-rafael-2025-2026-1.pdf>
- Spes BuBaO. (s.d.). *Contact — Openingsuren*. Geraadpleegd op 22 september 2025, van <https://www.spes-school.be/contact/>
- Stedelijke Buitengewoon Basisonderwijs SAIO. (s.d.). *Schooluren*. Geraadpleegd op 22 september 2025, van <https://saio.sbsgeel.be/openingsuren/>
- Styrka Lager Onderwijs. (s.d.). *Infogids 2025-2026*. Geraadpleegd op 22 september 2025, van <https://usercontent.one/wp/www.styrka.be/wp-content/uploads/sites/6/2025/09/DD-Infobrochure-2025-2026-1-compressed.pdf?media=1742886182>
- Styrka Secundair Onderwijs. (s.d.). *Schoolreglement 2025-2026*. Broeders van Liefde onderwijs en zorg. Geraadpleegd op 22 september 2025, van <https://usercontent.one/wp/www.styrka.be/wp-content/uploads/sites/2/2025/09/20250821-schoolreglement.pdf?media=1742886182>
- Styrka Secundair Onderwijs @waterkant. (s.d.). *Schoolreglement: met pedagogisch project versie 01.09.2025*. Geraadpleegd op 22 september 2025, van <https://usercontent.one/wp/www.styrka.be/wp-content/uploads/sites/7/2025/07/2025-2026-schoolreglement-SSW.pdf?media=1742886182>
- SVB Sint-Victor Beersel Kleuter- en lagere school. (s.d.). *Schoolreglement 2025-2026*. Geraadpleegd op 22 september 2025, van <https://drive.google.com/file/d/1UOF0s0kmACVG1VhVCRFNLgUz376YJShH/view>
- Ten Desselaer. (s.d.). *Buitengewoon Kleuter- en Lager Onderwijs Type 3 en Type 9*. Geraadpleegd op 22 september 2025, van <https://tendesselaer.be/userfiles/welkom%20in%20onze%20school%20klein.pdf>
- Tongelsbos, B. (s.d.). *Schooltijden*. Geraadpleegd op 22 september 2025, van <https://blo.tongelsbos.be/info/schooltijden>
- Triolo Basisschool voor Buitengewoon Onderwijs. (s.d.). *Schoolreglement Triolo Basisschool voor Buitengewoon Onderwijs*. Geraadpleegd op 22 september 2025, van <https://www.olo-rotonde.be/media/bl2kyrl2/schoolreglement-plus-busreglement-2022-2023.pdf>
- VIBO BuLO De Ring. (s.d.). *Schoolbrochure met schoolreglement*. Geraadpleegd op 22 september 2025, van <https://dering.vibo.be/media/2402/Schoolbrochure-met-schoolreglement-2022-2023.pdf>

Bijlage F

Beschrijving van deze masterproef
in de vorm van een
wetenschappelijk artikel

Route planning for student transportation in special education

Indra Coppens
Faculty of Engineering Technology
KU Leuven
Ghent, Belgium

Abstract

Students in special education are entitled to free transportation to and from school as long as they enroll in the nearest school offering their program. Organizing this transportation requires planning that accounts for various constraints and preferences, including vehicle capacity, travel time, and individual student needs. Despite the dedicated efforts of manual planners, the current system faces several shortcomings. Students often spend too much time on the bus, which frequently leads to public criticism and negative media coverage. We present a formal description of the route planning for student transportation in special education and develop an integer linear programming (ILP) formulation. The objective is to minimize the longest individual ride time, commonly referred to as a minimax objective.

Index Terms

route scheduling, student transportation, special education, time windows, school bus, Vehicle Routing Problem (VRP)

1. Introduction

Planning school transportation for students in special education is a complex logistical challenge. Students are entitled to free transportation to and from school provided they enroll in the nearest school offering their program. The scheduling procedure must respect several constraints, including bus capacity limits and time-window restrictions.

Despite the dedicated efforts of experienced manual planners, some students still spend excessive time on buses. This not only affects comfort and well-being, but also regularly leads to public criticism.

We seek to construct a plan that transports all students from their homes to their schools while satisfying all constraints and the minimax objective. Vehicles may continue picking up students after dropping off some of them at a school. Each student has a specified boarding and alighting duration; in our algorithm, these durations are equal and jointly treated as the student's service time. Boarding and alighting occur sequentially. Although many schools likely have only one or two bus parking spots, we do not restrict the number of vehicles that may drop off students simultaneously at a school node. This paper focuses on the morning planning problem (home to school).

While several related problems exist, a framework that focuses on minimizing ride time under a fixed heterogeneous fleet remains unexplored. To address this gap, we first provide a formal definition of the transportation problem for special education in Section 2. We then propose an integer linear programming formulation in Section 3 and evaluate it using a state-of-the-art solver in Section 4. Finally, Section 5 concludes the paper.

2. Notation and problem definition

This section defines the optimization problem and introduces the notation used throughout the mathematical model.

2.1 Definition

The problem involves a set of students U . Each student $u \in U$ must first be picked up at their home address a_u and subsequently dropped off at their school s_u . For every student, we define a unique school node, even if multiple students attend the same physical school. We aim to plan routes for a set of vehicles K , in this case buses. Each vehicle $k \in K$ has an associated capacity C_k , representing the maximum number of students that can be on board simultaneously. The objective is to construct routes that minimize the longest individual ride time across all students.

We define the ride time of a student as the difference between the moment immediately after boarding and the moment immediately after alighting. The service time for boarding and alighting for student u is denoted by $t_u^{service}$. These actions are performed sequentially; only one student may board or alight a vehicle at any given time.

A time window specifies when students must be dropped off at their school. For each school s_u , we define the interval $[t_{s_u}^-, t_{s_u}^+]$. The full service (here the alighting process) must lie within this window. If a vehicle arrives too early, it is allowed to wait at the node until the time window opens.

Vehicles may start and end at an arbitrary location (Open Vehicle Routing Problem (OVRP)). After delivering students, it is allowed to pick up additional students. Mixed loads are also permitted; students with different destinations may travel together, provided capacity constraints are respected.

The maximum value $\tau_{\max}$ acts as an upper bound on arrival times and exceeds all feasible arrivals. It equals the latest end time among all time windows. We define $C_{\max}$ as the maximum capacity across all vehicles in the fleet. This value is smaller than both the total number of students and the largest vehicle capacity in the fleet.

2.2 Network Construction

We model the arbitrary start location (OVRP) by introducing a virtual depot n_{virtual} . Each vehicle may depart from and arrive at this virtual depot exactly once.

The network consists of all nodes $n \in N$, including student home addresses, their corresponding schools, and the virtual depot. For a feasible solution, every node $n \in N$ except the virtual depot must be visited exactly once. This implies the following property:

Property Binary Capacity of the Arcs (BCB). Each arc may be used by at most one vehicle.

The set of valid arcs is denoted by E and contains all connections allowed by the problem. Each arc (i, j) between nodes i and j has an associated travel time t_{ij} . The network may be asymmetric, so t_{ij} does not necessarily equal t_{ji} .

2.3 Summary of Notation

As an overview, Table 1 summarizes the different notation used.

Notation	Description
$u \in U$	Student u from the set of all students U
$k \in K$	Vehicle k from the set of all vehicles K
s_u	School of student u
a_u	Home address of student u
$s \in S$	School s from the set of all schools $(= s_u \ \forall u)$
$a \in A$	Address a from the set of all student home addresses $(= a_u \ \forall u)$
$n \in N$	Node n from the set of all nodes N (with $N = A \cup S \cup \{n_{\text{virtual}}\}$)
$(i, j) \in E$	Arc (i, j) from the set of all valid arcs E
t_{ij}	Travel time on arc (i, j)
t_u^{service}	Service time for boarding/alighting of student u
n_{virtual}	Virtual depot
$\tau_{\max}$	Maximum time value
C_k	Capacity of vehicle k
$C_{\max}$	Maximum capacity across all vehicles
$[t_{s_u}^-, t_{s_u}^+]$	Time window for school s_u

TABLE 1: Summary of the notation used

2.4 Example Solution

Figure 1 shows an example of a solution for a network with six students. Each student is first picked up and then dropped off by the same vehicle. Each node is visited exactly once.

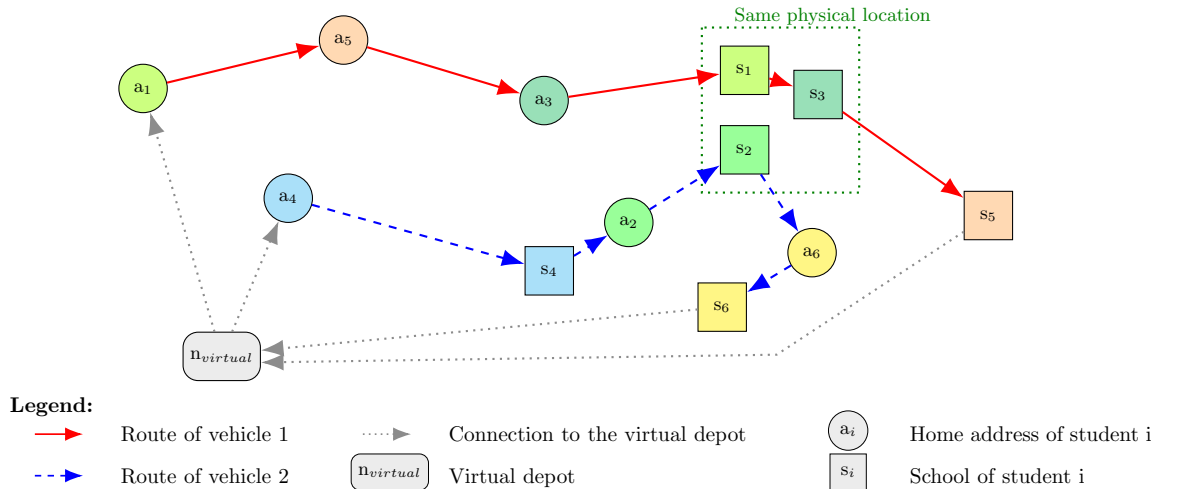


Fig. 1: Example of a solution with six students

3. Mathematical formulation

This section discusses a possible integer linear programming (ILP) formulation of the problem.

3.1 Decision variables

We first define three decision variables for the mathematical formulation. The variable x_{ijk} assigns vehicle k to arc (i, j) . Similarly, the variable y_{iju} assigns student u to arc (i, j) . Finally, we use a time variable τ_{ik} that represents the time at which vehicle k can depart from node i . Table 2 provides an overview of the decision variables.

Variable	Description
x_{ijk}	$= \begin{cases} 1 & \text{if vehicle } k \text{ uses arc } (i, j), \\ 0 & \text{otherwise} \end{cases}$
y_{iju}	$= \begin{cases} 1 & \text{if student } u \text{ uses arc } (i, j), \\ 0 & \text{otherwise} \end{cases}$
τ_{ik}	≥ 0 : time at which vehicle k can depart from node i (after the service time)

TABLE 2: Overview of the decision variables for the mathematical model

3.2 Formulation

Objective (1) minimizes the maximum individual ride time across all students, denoted by $maxRideTime$. Constraint (2) links $maxRideTime$ to the actual ride times.

$$\min \quad maxRideTime \quad (1)$$

$$\text{s.t.} \quad maxRideTime \geq \tau_{s_u k} - \tau_{a_u k} \quad \forall u \in U, k \in K \quad (2)$$

The next three constraints collectively ensure the correct construction of vehicle routes. Constraint (3) guarantees that each node (except the virtual depot) has exactly one incoming arc. Constraint (4) ensures that every vehicle that enters a node must also leave it. Constraint (5) ensures that each vehicle leaves the virtual depot at most once.

$$\sum_{k \in K} \sum_{\substack{i \in N \\ \text{met } (i, j) \in E}} x_{ijk} = 1 \quad \forall j \in N \setminus \{n_{virtual}\} \quad (3)$$

$$\sum_{\substack{i \\ \text{met } (i, n) \in E}} x_{ink} - \sum_{\substack{j \\ \text{met } (n, j) \in E}} x_{njk} = 0 \quad \forall k \in K, \forall n \in N \quad (4)$$

$$\sum_{\substack{j \\ \text{met } (n_{virtual}, j) \in E}} x_{n_{virtual}jk} \leq 1 \quad \forall k \in K \quad (5)$$

Constraints (6 - 7) express the time-related constraints. Constraint (6) ensures that a student is always picked up before being dropped off. Constraint (7) ensures the correctness of time propagation and also prevents subtours. We assume it is impossible for both travel time and service time between two nodes to be zero.

$$\tau_{s_u k} - \tau_{a_u k} \geq 0 \quad \forall u \in U, \forall k \in K \quad (6)$$

$$\tau_{ik} + (t_{ij} + t_j^{service}) \cdot x_{ijk} \leq \tau_{jk} + \tau_{max} \cdot (1 - x_{ijk}) \quad \forall k \in K, \forall (i, j) \in E \text{ met } j \neq n_{virtual} \quad (7)$$

The next five constraints ensure that every student follows a valid path from home to school. Constraints (8 - 9) ensure that each student leaves their home exactly once and does not revisit it. Constraints (10 - 11) ensure that each student arrives at their school exactly once and does not depart from it again. Constraint (12) balances the flow in all other nodes. Constraint (13) prevents students from departing from the virtual depot.

$$\sum_{\substack{j \in N \\ (a_u, j) \in E}} y_{aju} = 1 \quad \forall u \in U \quad (8)$$

$$\sum_{\substack{i \in N \\ (i, a_u) \in E}} y_{iau} = 0 \quad \forall u \in U \quad (9)$$

$$\sum_{\substack{i \in N \\ (i, s_u) \in E}} y_{is_u u} = 1 \quad \forall u \in U \quad (10)$$

$$\sum_{\substack{j \in N \\ (s_u, j) \in E}} y_{s_u j u} = 0 \quad \forall u \in U \quad (11)$$

$$\sum_{\substack{i \\ \text{met } (i, n) \in E}} y_{inu} - \sum_{\substack{j \\ \text{met } (n, j) \in E}} y_{nju} = 0 \quad \forall u \in U, \forall n \in N \setminus \{a_u, s_u\} \quad (12)$$

$$y_{n_{\text{virtual}} j u} = 0 \quad \forall u \in U, \forall j \in N \text{ met } (n_{\text{virtual}}, j) \in E \quad (13)$$

Constraint (14) ensures that vehicle capacity is not exceeded. This constraint is valid because of Property BCB.

$$\sum_{u \in U} y_{iju} \leq C_k + C_{\max} \cdot (1 - x_{ijk}) \quad \forall (i, j) \in E, \forall k \in K \quad (14)$$

Constraint (15) links student routes to vehicle routes. A student may only use an arc if the arc is also used by a vehicle.

$$y_{iju} \leq \sum_{k \in K} x_{ijk} \quad \forall (i, j) \in E, \forall u \in U \quad (15)$$

Finally, Constraints (16 - 17) enforce the time windows. Constraint (16) ensures that a vehicle can only drop off students after the start of the time window. Constraint (17) ensures that all students are dropped off before the time window closes.

$$(t_{s_u}^- + t_{s_u}^{\text{service}}) \cdot \sum_{\substack{i \\ \text{met } (i, s_u) \in E}} x_{is_u k} \leq \tau_{s_u k} \quad \forall u \in U, \forall k \in K \quad (16)$$

$$\tau_{s_u k} \leq t_{s_u}^+ \quad \forall u \in U, \forall k \in K \quad (17)$$

3.3 Additional valid inequalities

These additional constraints are not strictly required for correctness of the algorithm, but they reduce the search space and therefore improve runtime. They typically follow from one or more necessary properties.

Constraint (18) ensures that each student is picked up and dropped off by a single vehicle. Constraint (19) ensures Property BCB. Constraint (20) forces the time value at nodes not visited by a vehicle to be zero. Constraint (21) ensures that students never arrive at the virtual depot. Finally, Constraint (22) ensures that every node, except the virtual depot, has at most one outgoing arc.

$$\sum_{\substack{j \\ \text{met } (a_u, j) \in E}} x_{a_u j k} - \sum_{\substack{i \\ \text{met } (i, s_u) \in E}} x_{is_u k} = 0 \quad \forall u \in U, \forall k \in K \quad (18)$$

$$\sum_{k \in K} x_{ijk} \leq 1 \quad \forall (i, j) \in E \quad (19)$$

$$\tau_{jk} \leq 0 + \tau_{\max} \cdot \sum_{\substack{i \\ \text{met } (i, j) \in E}} x_{ijk} \quad \forall j \in N, \forall k \in K \quad (20)$$

$$y_{in_{\text{virtual}} u} = 0 \quad \forall u \in U, \forall i \in N \text{ met } (i, n_{\text{virtual}}) \in E \quad (21)$$

$$\sum_{k \in K} \sum_{\substack{j \in N \\ \text{met } (i, j) \in E}} x_{ijk} = 1 \quad \forall i \in N \setminus \{n_{\text{virtual}}\} \quad (22)$$

3.4 Alternative formulation with flow variables

The formulation in Paragraph 3.2 has the advantage that it directly shows which arcs are used by which students. This simplifies the creation of schedules and may be beneficial for future extensions. An alternative formulation based on flow variables is provided in Appendix A. This formulation contains fewer decision variables and may therefore be solved more efficiently.

4. Computational results

For the experiments, we use a computer equipped with an AMD Ryzen 9 5950X 16-core processor (3.4 GHz) and the Windows 11 operating system. The Mixed Integer Linear Programming (MILP) solver used is Gurobi 12.0.2. We run the experiments on subsets of realistic instances that represent groups of cities. We consider seven different input sizes, characterized by the number of students (U), vehicles (K), and schools (S). The maximum computation time for the solver is limited to one hour. If no optimal solution is found within this time limit, the algorithm reports the best solution obtained so far. If no feasible solution is available within the time limit, a dash is shown instead of an objective value. The implementation uses an internal time unit of ten seconds, meaning that all times are expressed as multiples of ten seconds. This choice provides a good balance between accuracy and computational effort: working in seconds is too fine-grained and increases computation time, whereas using minutes is too coarse and leads to inaccurate results.

The results in Table 3 show that computation time is significantly lower when the optimal objective value coincides with the lower bound. We compute this lower bound as the longest direct travel time from a student’s home address to their school. In such cases, the solver can quickly prove optimality, which greatly accelerates the solving process.

Computation times remain acceptable up to problem instances of eight students, three vehicles and three schools. For larger instances, optimal solutions are only achievable when the optimal objective value equals the lower bound. For the instances containing twenty students, the solver is unable to find any feasible solution within the one-hour time limit.

These findings indicate that the algorithm does not scale to realistic instances. A summary of the results is shown in Table 3.

Size	Dataset	Proven optimal?	Longest ride time [min]	Lower bound [min]	Computation time [s]
U5 K2 S2	Set1_U5_K2_S2_T1_2023	✓	43.2	31.0	0.421
	Set2_U5_K2_S2_T1_2023	✓	25.8	25.8	0.242
	Set3_U5_K2_S2_T1_2023	✓	33.8	28.3	0.853
	Set4_U5_K2_S2_T1_2023	✓	40.0	30.5	0.382
U6 K3 S2	Set1_U6_K3_S2_T1_2023	✓	31.0	31.0	0.088
	Set2_U6_K3_S2_T1_2023	✓	17.0	16.0	15.405
	Set3_U6_K3_S2_T0_2023	✓	14.3	14.3	0.055
	Set4_U6_K3_S2_T2_2023	✓	36.7	27.5	8.721
U7 K2 S3	Set1_U7_K2_S3_T2_2023	✓	42.7	31.0	16.111
	Set2_U7_K2_S3_T1_2023	✓	60.7	53.0	11.401
	Set3_U7_K2_S3_T0_2023	✓	60.2	42.3	32.992
	Set4_U7_K2_S3_T1_2023	✓	42.5	23.8	24.322
U8 K3 S3	Set1_U8_K3_S3_T2_2023	✓	31.0	31.0	4.098
	Set2_U8_K3_S3_T1_2023	✓	16.5	16.5	3.224
	Set3_U8_K3_S3_T1_2023	✓	57.0	32.3	229.783
	Set4_U8_K3_S3_T1_2023	✓	33.5	33.5	14.150
U10 K3 S3	Set1_U10_K3_S3_T2_2023	✗	44.8	31.0	3600.239
	Set2_U10_K3_S3_T1_2023	✓	20.7	20.7	715.162
	Set3_U10_K3_S3_T0_2023	✓	38.2	38.2	79.114
	Set4_U10_K3_S3_T2_2023	✗	52.7	41.2	3600.165
U20 K4 S5	Set1_U20_K4_S5_T_2023	✗	-	29.8	3600.044
	Set2_U20_K4_S5_T_2023	✗	-	78.2	3600.299
	Set3_U20_K4_S5_T_2023	✗	-	49.5	3600.518
	Set4_U20_K4_S5_T_2023	✗	-	64.5	3600.403

TABLE 3: Results of the mathematical formulation

5. Conclusion

This paper presented a formal description of the route planning problem for student transportation in special education and introduced an integer linear programming formulation that captures its key operational characteristics, including heterogeneous vehicle capacities, time windows, many-to-several routing, and door-to-door service. The computational results show that although the model can optimally solve small instances, its performance deteriorates rapidly as the problem size increases. For realistic instances, this exact approach becomes computationally infeasible. These findings highlight the need for scalable and flexible solution techniques to solve realistic instances.

Appendix A

Alternative formulation with flow variables

In this appendix, we describe an alternative formulation of the problem based on flow variables.

1.1 Decision variables

The variables x and τ remain identical to those in the original formulation. Variable f represents the flow along an arc.

$f_{ij} \geq 0$: The number of students that use arc (i, j)

1.2 Formulation

The objective function, as defined by Constraints (1 - 2), remains unchanged relative to Section 3.2. Furthermore, Constraints (3 - 7) are also retained. Constraints (8 - 9) are merged into Constraint (23). Similarly, Constraints (10 - 11) are merged into Constraint (24). Flow conservation in the virtual depot is enforced through Constraint (25). This constraint replaces Constraint (12) from the original formulation. Constraint (26) replaces Constraint (14). Constraint (15) is rewritten as Constraint (27). Constraints (16 - 17) remain unchanged. The additional valid inequalities (18 - 20, 22) also remain unchanged, while Constraint (21) is replaced by Constraint (28).

$$\sum_{\substack{j \in N \\ (a_u, j) \in E}} f_{a_u j} - \sum_{\substack{i \in N \\ (i, a_u) \in E}} f_{i a_u} = 1 \quad \forall u \in U \quad (23)$$

$$\sum_{\substack{j \in N \\ (s_u, j) \in E}} f_{s_u j} - \sum_{\substack{i \in N \\ (i, s_u) \in E}} f_{i s_u} = -1 \quad \forall u \in U \quad (24)$$

$$\sum_{\substack{i \\ \text{met } (i, n) \in E}} f_{in} - \sum_{\substack{j \\ \text{met } (n, j) \in E}} f_{nj} = 0 \quad \forall n \in N \setminus \{A, S\} \quad (25)$$

$$f_{ij} \leq c_k + C_{\max} \cdot (1 - x_{ijk}) \quad \forall (i, j) \in E, \forall k \in K \quad (26)$$

$$f_{ij} \leq \sum_{k \in K} x_{ijk} \cdot c_k \quad \forall (i, j) \in E, \forall u \in U \quad (27)$$

$$f_{n_{\text{virtual}} j} = 0 \quad \forall j \in N \text{ met } (n_{\text{virtual}}, j) \in E \quad (28)$$

References

Coppens, I. (2026). *Rittenplanning voor het leerlingenvervoer in het buitengewoon onderwijs* [Master's thesis, KU Leuven].

Bijlage G

Poster

Rittenplanning voor het leerlingenvervoer in het buitengewoon onderwijs

Leerlingen in het buitengewoon onderwijs zitten te lang op de bus ...

oplossing voor urenlange busritten in buitengewoon onderwijs

Anne Vanrenterghem
© wo 17 nov. 2021 17u13

Lange ritten, te laat op school

"Onbegrijpelijk"

"Meer dan 1000 leerlingen
zitten 4 tot 6 uur per dag op de bus"

Nog altijd geen oplossing voor busvervoer
buitengewoon onderwijs: "Kleuter bijna 5 uur op de
bus is onmenselijk"

Brigitte Vermeersch
© ma 15 nov. 2021 05u40

vr̃ nws

Busritten buitengewoon onderwijs
blijven onaanvaardbaar lang - 15 nov. 2021

'Factcheckers': Hoe zwaar is een lange busrit voor
kinderen in het buitengewoon onderwijs?

maandag 17 maart 2025

vr̃

1 op de 4 kinderen in buitengewoon onderwijs nog
altijd meer dan 90 minuten onderweg naar school

Ann Willems, Belga
© zo 24 nov. 2024 15u27

vr̃ nws

Doelstelling

Minimaliseren van de langste tijd dat een
leerling op de bus naar school moet zitten

Uitbreiding op basisprobleem

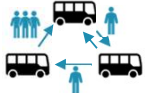
Tijdsvensters

Periode waarin leerlingen mogen aankomen op school



Transferpunten

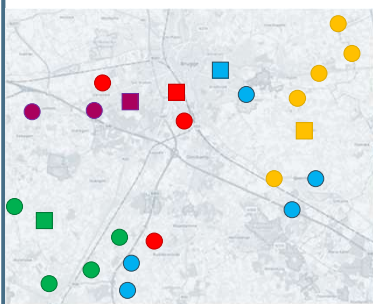
Parkings waar bussen leerlingen uitwisselen



Methode

Data generatie

Input netwerk



Parameters

- Maximum rittijd
- Tijdseenheid
- ...

Algoritme

Huidige oplossing



Ruin



Recreate



Compress Schedule



Accepteer of weiger



Langste rittijd

&

Rittenplanning

&

Tijdstip van ophalen
voor elke leerling



Bijlage H

GenAI code of conduct for students

Generative AI (GenAI) assistance tools can be used to generate text, image, code, video, music, or combinations of these. It includes typical tools like (but this list is not limited to): ChatGPT, Google Gemini, MS Copilot, Midjourney, Claude.ai, Perplexity.ai, Dall-E, etc.

Student Information

Student name: Indra Coppens

This form is related to my Master thesis.

Title Master thesis: Rittenplanning voor het leerlingenvervoer in het buitengewoon onderwijs

Promoter: Prof. Dr. Ir. Greet Vanden Berghe

Daily supervisor: Ing. Niels De Walsche

Use of GenAI Assistance

Please indicate with "X" (possibly multiple times) in which way you were using GenAI:

☐ I did not use any GenAI assistance tool.

☒ I did use GenAI assistance. In this case, specify which ones (e.g., ChatGPT, ...):
MS Copilot, Claude

GenAI Assistance Used As/For

- ☒ MS Copilot
- ☒ Language assistance
- ☒ Search engine
- ☐ Literature search
- ☐ Short-form input assistance
- ☒ Generating programming code
- ☐ Generating new research ideas
- ☐ Generating blocks of text
- ☐ Other (specify): _____

Code of Conduct for Different Uses

Language Assistant for Reviewing or Improving Texts

This use is similar to using spelling and grammar check tools, and you do not have to refer to the use of GenAI in the text. Be careful:

- Using GenAI tools on texts you did not write yourself to cover up plagiarism (by paraphrasing original texts) is not allowed.

Search Engine for Information or Existing Research

This use is similar to a Google search or checking Wikipedia. If you then write your own text based on this information, you do not have to refer to the use of GenAI in the text. Be careful:

- Be aware that the output of the GenAI tool cannot be guaranteed as a 100% reliable source of information.

- The output may not be entirely correct and is limited due to the databases it uses. Knowledge evolves and may change over time, and it may be that the database of the GenAI tool is not up to date.

Literature Search

This use is comparable to a Google Scholar search. Be careful:

- Be aware that the output is restricted to the database it is built on. After this initial search, look for scientific sources and conduct your own analysis.
- GenAI tools (like ChatGPT) may output no or wrong references. As a student, you are responsible for further checking and verifying the accuracy of references.

Short-form Input Assistance

This use is similar to Google Docs powered by generative language models.

Generating Programming Code

The use of GenAI for coding should be explicitly allowed by the teacher. If used for coding, correctly mention the use of GenAI assistance and cite it.

Generating New Research Ideas

Further verify whether the idea is novel or not. It is likely that it is related to existing work, which should be referenced.

Generating Blocks of Text

Inserting blocks of text without quotes and a reference to GenAI assistance in your report or thesis is not allowed. Be careful:

- When you literally copy elements from a conversation with a GenAI tool, quote them between quotation marks and refer to them according to the specified reference style.
- Describe the use of the GenAI tool (tool name, version, date, etc.) in the method section and optionally add the full conversation as an attachment.

Other

Contact the professor of the course or the promoter of the thesis. Inform also the program director. Motivate how you comply with Article 84 of the exam regulations. Explain the use and added value of ChatGPT or another AI tool.

Further Important Guidelines and Remarks

- GenAI assistance cannot be used related to data or subjects under Non-Disclosure Agreement.
- GenAI assistance cannot be used related to sensitive or personal data due to privacy issues.
- Take a scientific and critical attitude when interacting with GenAI assistance and interpreting its output.
- As a student, you are responsible for complying with Article 84 of the exam regulations: your report or thesis should reflect your own knowledge, understanding, and skills.

Exam Regulations Article 84

“Every conduct individual students display with which they (partially) inhibit or attempt to inhibit a correct judgement of their own knowledge, understanding and/or skills or those of other students, is considered an irregularity which may result in a suitable penalty. A special type of irregularity is plagiarism, i.e., copying the work (ideas, texts, structures, designs, images, plans, codes, ...) of others or prior personal work in an exact or slightly modified way without adequately acknowledging the sources.”

Additional Reading

More information about being transparent on the use of GenAI assistance and about citing and referencing GenAI can be found on the student website.

A Few Final Words

If you are uncertain whether or not you should declare your use of AI tools, discuss the matter with your teacher or promoter. It is safer to declare AI use when it is not needed than to withhold that declaration when it is required.

Finally, remember that advanced AI tools are new and that they can do things they could not do until recently. It is important to follow up on the most recent developments in AI technologies and communicate openly with your teachers, assistants, supervisors, and peers.

KU LEUVEN - GENT

Faculteit Industriële Ingenieurswetenschappen
Campus Rabot, Gebroeders De Smetstraat 1

9000 Gent, België

tel: + 32 9 265 86 10

iiw.gent@kuleuven.be

www.iw.kuleuven.be

