



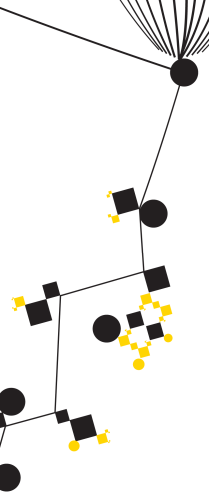
Bacheloropdracht
Technische Wiskunde

Optimalisatie van de planning in de thuiszorg

Afspraken inplannen met behulp van Particle Swarm Optimization



Marieke Dijkink
Stefan Klootwijk
Joram Span



Begeleider:
Prof. dr. R.J. Boucherie

13 juni 2014

UNIVERSITEIT TWENTE.



Samenvatting

In dit verslag worden twee modellen geïntroduceerd voor de planning in de thuiszorg. Het eerste model is voor de cliëntenplanning. Hierin worden afspraken van cliënten gekoppeld aan thuiszorgmedewerkers. In dit model wordt met meerdere randvoorwaarden rekening gehouden. Het model probeert de organisatietijd te minimaliseren en de robuustheid te maximaliseren, waarbij er rekening moet worden gehouden met de tijdsvensters van de afspraken, de verschillende functieniveaus van de medewerkers, de evv'er van een cliënt en continuïteit van de zorg. Om dit model op te lossen, wordt er een heuristiek gegeven die gebaseerd is op het principe van Particle Swarm Optimization. Ook wordt er een model opgesteld voor de personeelsplanning. Hierin wordt bepaald welke thuiszorgmedewerker wanneer werkt. Zonder een goede personeelsplanning is het namelijk niet mogelijk om een goede cliëntenplanning te maken. De voorgestelde PSO-heuristiek is geïmplementeerd en getest aan de hand van realistische data van CARINTREGGELAND. Uit deze tests is gebleken dat de voorgestelde heuristiek werkt en betere oplossingen geeft dan de huidige planningsmethode bij CARINTREGGELAND.

Voorwoord

Dit verslag is een presentatie van de bacheloropdracht van Marieke Dijkink, Stefan Klootwijk en Joram Span. Deze bacheloropdracht is uitgevoerd aan de Universiteit Twente, afdeling Technische Wiskunde, in de periode van februari tot en met juni 2014. De bacheloropdracht is uitgevoerd onder begeleiding van professor Richard Boucherie. Tijdens de de gehele onderzoeksperiode hebben we contact gehad met de thuiszorgorganisatie CARINTREGGELAND in Losser. Met behulp van hun informatie hebben wij een helder beeld gekregen van de dagelijks praktijk in de thuiszorg. Hierdoor hebben we ons onderzoek kunnen toespitsen op de praktijk. Verder hebben we gebruik kunnen maken van hun data om het model dat we gemaakt hebben, uitgebreid te onderzoeken.

We willen professor Richard Boucherie bedanken voor zijn begeleiding, zijn adviezen en zijn kritische blik. Ook willen we Charlotte van de Wetering, Sharon Volker en Anja Fhij van CARINTREGGELAND bedanken voor hun medewerking aan dit onderzoek.

Inhoudsopgave

Voorwoord	4
1 Inleiding	7
2 Literatuurreview	8
2.1 Cliëntenplanning	8
2.2 Personeelsplanning	9
3 Probleemstelling	10
3.1 Cliëntenplanning	10
3.1.1 Thuiszorgmedewerkers	10
3.1.2 Cliënten	10
3.1.3 Zorgmomenten	11
3.1.4 Reistijden	11
3.1.5 Randvoorwaarden	11
3.1.6 Doelen	12
3.2 Personeelsplanning	12
3.2.1 Medewerkers	12
3.2.2 Diensten	12
3.2.3 Randvoorwaarden	13
3.2.4 Doelen	13
4 Modelling	14
4.1 Modelling cliëntenplanning	14
4.1.1 Variabelen en definities	14
4.1.2 Planningsvariabelen	15
4.1.3 Randvoorwaarden	16
4.1.4 Doelen	16
4.1.5 ILP-formulering	17
4.2 Modelling personeelsplanning	20
4.2.1 Variabelen	20
4.2.2 Randvoorwaarden	20
4.2.3 Doelfunctie	21
5 Oplossingsstrategie cliëntenplanning	22
5.1 Particle Swarm Optimization	22
5.2 Globale werking van de PSO-heuristiek	22
5.3 Onderdelen van de PSO-heuristiek	23
5.3.1 Initialisatie	23
5.3.2 Heuristic Assignment	24
5.3.2.1 Controleren van feasibility	25
5.3.2.2 Evv-controle	26
5.3.3 Local Improvement Procedure (LIP)	27
5.3.4 Beste oplossingen	27
5.3.5 Snelheid aanpassen	28
5.4 Parameters van de PSO-heuristiek	28
5.4.1 PSO-parameters	28
5.4.2 Taguchi-methode	29

6	Numerieke experimenten	31
6.1	Experiment 1: PSO-parameters	31
6.2	Experiment 2: PSO-heuristiek versus exact algoritme	31
6.3	Experiment 3: Complexiteit van de PSO-heuristiek	33
6.4	Experiment 4: De praktijk	34
7	Conclusies en aanbevelingen	35
7.1	Conclusies en discussie	35
7.2	Aanbevelingen voor verder onderzoek	35
	Bibliografie	35
	Appendices	38
A.1	Appendix 1: Lijst van gebruikte afkortingen	38
A.2	Appendix 2: Lijst van gebruikte symbolen	39
A.3	Appendix 3: $L50$ -orthogonaalmatrix	41

1 Inleiding

Volgens het Centraal Planbureau naar verwachting zullen de zorgkosten de komende jaren in Nederland met 2,6% tot 4,2% op jaarbasis stijgen ([30]). Nu al wordt er 67 miljard euro aan de zorg uitgegeven. Als de zorgkosten net zo hard doorgroeien als de afgelopen tien jaar, dan zijn deze in 2040 gelijk aan 31% van alles wat we in Nederland verdienen. Deze stijging van de zorgkosten is voornamelijk te verklaren door de vergrijzing, een hogere welvaart en een veranderde maatschappij ([30]). Het is dus belangrijk dat er oplossingen gevonden worden om de zorg minder duur te maken. Binnen de zorg vormt de thuiszorg een grote kostenpost. Als de inzet van medewerkers in de thuiszorg door een betere planning geoptimaliseerd kan worden, dan kan dit bijdragen aan het in toom houden van de zorgkosten.

Voor dit verslag hebben we de informatie van de thuiszorgorganisatie CARINTREGGELAND in Losser gebruikt om een beeld te krijgen van de planningsproblemen in de thuiszorg. CARINTREGGELAND werkt met zelforganiserende teams. Dit houdt in dat medewerkers in wijkteams zijn ingedeeld en dat elk wijkteam voor zijn eigen planning zorgt. Bij CARINTREGGELAND wordt momenteel de planning handmatig gedaan met behulp van de PlanningsOnderSteuning (POS) van TIS Care.

Het doel van ons onderzoek is de ontwikkeling van een model om de koppeling tussen thuiszorgmedewerkers en cliënten te optimaliseren. In het vervolg van dit verslag geven we eerst een korte achtergrond over de onderzoeken die al gedaan zijn op het gebied van planning in de (thuis)zorg. Vervolgens beschrijven we uitgebreid de planningsproblemen die we zijn tegengekomen bij CARINTREGGELAND, waarna voor beide problemen een wiskundig model opgesteld wordt. In hoofdstuk 5 wordt vervolgens een heuristisch beschreven die in staat is de koppeling tussen thuiszorgmedewerkers en cliënten te maken. Daarna presenteren we de resultaten van een viertal numerieke experimenten en ten slotte worden er conclusies en aanbevelingen gegeven.

2 Literatuurreview

In onderzoeken binnen de planning van de thuiszorg wordt er meestal onderscheid gemaakt tussen de cliëntenplanning en de personeelsplanning. Bij de cliëntenplanning gaat het voornamelijk over het koppelen van cliënten aan medewerkers zodanig dat aan alle gestelde randvoorwaarden wordt voldaan. Bij de personeelsplanning gaat het voornamelijk om het inroosteren van medewerkers op bepaalde diensten of dagen. Het maken van een cliëntenplanning wordt over het algemeen gezien als het vinden van een oplossing van het Home Health Care Problem (HHCP). [9] is een van de eerste artikelen op het gebied van het HHCP. Hierin is het HHCP benaderd als een variant van het Vehicle Routing Problem with Time Windows (VRPTW). Er wordt een heuristisch gebruikt om dit op te lossen. Daarna zijn er nog vele artikelen geschreven die het HHCP op verschillende manieren benaderen en op verschillende manieren oplossen.

2.1 Cliëntenplanning

Over het algemeen wordt het HHCP gezien als een uitbreiding of een generalisatie van het ‘Vehicle Routing Problem with Time Windows’ (VRPTW) ([1, 6, 7, 12, 25]) of als een speciaal geval van het Travelings Salesman Problem (TSP) ([14, 16, 22, 31]). Naast het onderscheid in benadering van het probleem zijn er ook verschillende oplossingsstrategieën, constraints en doelfuncties in de literatuur te vinden. In de onderzoeken worden bepaalde heuristieken omschreven, die een oplossing geven voor het HHCP. Verder wordt er meestal geminimaliseerd naar reisafstand en (reis)kosten of gemaximiseerd naar kwaliteit van de service. Ook is er een verscheidenheid aan randvoorwaarden te vinden in de verschillende onderzoeken. Er wordt bijvoorbeeld rekening gehouden met functieniveaus van de medewerkers, tijdsvensters voor de afspraken, pauzes, verschillende behandelingen en gesplitste afspraken.

In [12] wordt het HHCP benaderd als een VRPTW met minimalisatie naar de (reis)kosten. Er wordt gebruik gemaakt van het ‘Set-Partitioning’-model en het ‘Repeated-Matching’-algoritme om een oplossing te vinden. Binnen de gekozen benadering is aan elk bezoek een bepaalde taak gekoppeld, mag elke medewerker alle taken uitvoeren en wordt, indien mogelijk, elke cliënt bezocht door dezelfde contactpersoon. In [7] wordt het HHCP op dezelfde manier benaderd, maar wordt dit opgelost met een ‘Mixed Integer Linear Programming (MILP) Solver’ en is er gebruik gemaakt van een ‘Local-Branching’-heuristiek voor de optimalisatie van de reistijd en kwaliteit van de service.

In [25] wordt gebruikt gemaakt van ‘Branch and Price’ en wordt er geoptimaliseerd naar reiskosten. Verder wordt ervoor gezorgd dat er zo min mogelijk (belangrijke) afspraken blijven staan. Tevens wordt er rekening gehouden met gesplitse afspraken, tijdslimieten van routes en het functieniveau van de medewerker. In [6] wordt er gebruik gemaakt van een ‘Cheapest-Insertion’-heuristiek en lokale-optimalisatieheuristieken. Er wordt geminimaliseerd naar reistijd en aantal benodigde medewerkers. Verder wordt er rekening gehouden met de continuïteit van de zorg door zo min mogelijk medewerkers per cliënt in te zetten en wordt er rekening gehouden met de functieniveaus van de medewerkers, het gebruik van verschillende voertuigen en tijdsafhankelijkheden tussen afspraken.

In [1] wordt het HHCP benaderd als een VRPTW en wordt dit opgelost met behulp van ‘Particle Swarm Optimization’ (PSO). Binnen dit model wordt er rekening gehouden met de werktijden van de medewerkers en met de tijdsvensters van de afspraken. Er wordt aangegeven dat PSO-heuristieken vaak makkelijk zijn aan te passen aan nieuwe of andere gelijksoortige problemen. Ook in ons onderzoek wordt er gebruik gemaakt van een PSO-heuristiek om het HHCP op te lossen. In [2] wordt het PSO-algoritme geanalyseerd en wordt er gekeken naar de invloed van enkele parameters die worden gebruikt binnen het PSO-algoritme.

Er zijn meerdere onderzoeken gedaan naar de parameters van het PSO-algoritme. Zo is er in [8, 17] te zien hoe de parameters getest kunnen worden aan de hand van de ‘Taguchi’-methode. In [10, 11, 27, 28, 29, 32] worden suggesties gegeven voor verschillende parameterwaarden in bepaalde situaties.

Een ander veel voorkomende benadering van het HHCP is het ‘Multiple Traveling Salesman Problem (with Time Windows)’ (MTSP(TW)). Het MTSP is een verzwakking van het ‘Vehicle Routing Problem’ (VRP). In [31] worden de routes voor alle medewerkers bepaald aan de hand van een op de juiste wijze gedefinieerd TSP. In [14] wordt gebruik gemaakt van het MTSP(TW). Net als in [31] wordt de totale afstand geminimaliseerd. Als oplosmethode wordt ‘Tabu Search’ gebruikt. In [16] en [22] wordt ook gebruik gemaakt van het MTSP(TW), maar wordt er rekening gehouden met meerdere randvoorwaarden. Beide onderzoeken houden rekening met het functieniveau van de medewerkers en ook met zogeheten ‘double services’. In [16] worden de totale reiskosten geminimaliseerd, waarbij gebruik gemaakt wordt van CPLEX. In [22] kunnen verschillende medewerkers verschillende startlocaties hebben. Hier bestaat de doelfunctie uit een lineaire combinatie van de totale reistijd, de totale vertraging die opgelopen is over alle afspraken en de maximale vertraging die is ontstaan. Na het verkrijgen van een initiële oplossing, worden een drietal heuristieken voor verbeteringen van deze initiële oplossing met elkaar vergeleken.

Naast de twee voorgaande benaderingen van het HHCP zijn er ook andere benaderingen te vinden. Daarbij zijn de randvoorwaarden over het algemeen dat de zorgvraag deterministisch is, dat er gebruik gemaakt wordt van tijdsvensters, dat de medewerkers verschillende functieniveaus hebben, dat cliënten alleen worden ingepland tijdens werkuren en dat alle cliënten ingepland moeten worden.

In [4] wordt het HHCP als een hybride vorm van een ‘rostering’-model en een ‘routing’-model gemodelleerd. Daarbij zijn er ‘soft’ en ‘hard’ tijdsvensters en diverse ‘soft’ en ‘hard constraints’. Aan de ‘hard constraints’ moet in ieder geval voldaan worden, de ‘soft constraints’ mogen overschreden worden, maar wanneer dit gebeurt, levert dat extra kosten op. Er wordt geminimaliseerd naar de reisafstand en de kosten van het overschrijden van de ‘soft constraints’. Als oplosmethode wordt een combinatie van ‘Linear Programming’, ‘Constraint Programming’, ‘Simulated Annealing’ en ‘Tabu Search’ gebruikt.

In [24] wordt gekeken naar het HHCP als een combinatie van het ‘Master Schedule Problem’ (MSP) en het ‘Operational Planning Problem’ (OPP). Het HHCP wordt, net als het MSP, als een LP-probleem geformuleerd. In [3] wordt gebruikt gemaakt van een LP-probleem waarin is aangenomen dat de reistijd is inbegrepen bij de behandeltijd. Er wordt geminimaliseerd naar het verschil tussen de maximale en minimale ‘workload’. Om oplossingen te vinden, wordt er gebruik gemaakt van LINGO. Net als in [3] maakt [5] ook gebruik van een LP-probleem dat wordt opgelost door LINGO. De belangrijkste randvoorwaarden zijn in dit geval dat er continuïteit van de zorg is, dat een cliënt maximaal één afspraak per dag heeft, dat een cliënt voorkeursdagen heeft voor zijn of haar afspraken en dat alle afspraken een tijdsduur hebben die deterministisch en constant is.

In de bestaande literatuur wordt nergens het gehele probleem van de cliëntenplanning, zoals het in de praktijk voorkomt, behandeld. In dit verslag worden ook ‘kleine’ randvoorwaarden meegenomen, waardoor de praktijksituatie beter benaderd kan worden. Het HHCP is een \mathcal{NP} -hard probleem ([22, 24]). Het aantal mogelijke oplossingen voor het HHCP neemt exponentieel toe met de grootte van het probleem. Voor het gebruik van een exact algoritme is daarom al snel een te grote rekenkracht nodig. Als alternatief zouden heuristische technieken gebruikt moeten worden die snel zijn en relatief goede oplossingen geven. Uit de bestaande literatuur blijkt dat PSO een techniek is die hieraan voldoet en dat PSO daarnaast goede resultaten geeft voor planningsproblemen in vergelijking met andere heuristieken ([1]). Dit geeft aan dat een heuristiek die op PSO is gebaseerd mogelijk geschikt is om het HHCP efficiënt op te lossen.

2.2 Personeelsplanning

De personeelsplanning wordt over het algemeen voor een langere periode gemaakt dan de cliëntenplanning. In [18] wordt de personeelsplanning als een Markov-beslisingsprobleem gezien. Er wordt rekening gehouden met verschillende service-profielen van cliënten, maar niet met verschillende functieniveaus van de medewerkers. Als oplosmethodes worden een ‘Value-Iteration’-algoritme en een ‘Trunk Reservation Policy’ gebruikt. In [26] wordt het probleem van de personeelsplanning het ‘Manpower Scheduling Problem’ (MSP) genoemd. Er wordt gebruikt gemaakt van ‘hard’ en ‘soft constraints’. Voor de oplossing van het probleem wordt er fuzzy wiskunde gebruikt. Hierbij zijn PSO en ‘Elite Tabu Search’ (ETS) als oplosmethodes gebruikt.

In [21] wordt een personeelsplanning gemaakt door een aantal cliënten toe te voegen aan een al bestaande planning. De hoeveelheid zorg die deze cliënten vragen is random en hangt niet af van de medewerker die de behandeling(en) uitvoert. Als een cliënt toegewezen is aan een medewerker, dan zal deze medewerker altijd de behandelingen van deze cliënt uitvoeren. Binnen het gebruikte algoritme wordt de kans op overwerk geminimaliseerd.

In [20] wordt naar een afdeling in het ziekenhuis gekeken waar mensen komen die geopereerd zijn of nog geopereerd moeten worden. Er wordt gekeken naar een situatie met twee soorten patiënten: patiënten waarvan de operatie al van tevoren gepland is, en patiënten die met spoed geholpen moeten worden. Er wordt gebruik gemaakt van het ‘Master Surgical Schedule’ (MSS) en de aankomstpatronen van spoedpatiënten. Aan de hand hiervan wordt de verwachte bedbezetting bepaald. In [19] wordt er naar dezelfde situatie als in [20] gekeken. Er wordt vervolgens per afdeling bepaald wat de personeelsplanning moet zijn. Aan de hand van [20] wordt er voorspeld wat de bedbezetting is en daarna wordt er bepaald hoeveel medewerkers er per afdeling nodig zijn voor een bepaalde dienst. Hierbij wordt er geminimaliseerd naar de totale kosten. Er wordt geen rekening gehouden met niveauverschillen van medewerkers. In [13] wordt een indeling gemaakt voor het gebruik van operatiekamers door specialistische teams. Hierbij wordt er voor elk team geminimaliseerd naar lengte van de wachtrij van patiënten. Er wordt een ‘adjustable-robust’-model gemaakt voor het ‘Master Schedule’ (MS) probleem. Er wordt gebruik gemaakt van het ‘implementor/adversary’-algoritme. In [13, 19, 20] wordt er gewerkt met ‘cyclic scheduling’.

In dit verslag worden verschillende onderdelen uit de bestaande literatuur samengevoegd om een model te maken voor het probleem van het maken van de personeelsplanning.

3 Probleemstelling

In dit hoofdstuk wordt een gedetailleerde beschrijving gegeven van het Home Care Crew Scheduling Problem (HHCSP). De verschillende elementen van dit probleem worden beschreven en daarbij wordt ook aangegeven welke gegevens er nodig zijn om het probleem op te kunnen lossen. Ook worden de randvoorwaarden van het probleem beschreven. Deze randvoorwaarden zijn gebaseerd op de informatie die we hebben gekregen van CARINTREGGELAND, en zullen daarom niet allemaal algemeen toepasbaar zijn binnen de thuiszorg. Echter, de meeste verschillen in randvoorwaarden tussen verschillende thuiszorginstellingen zijn zo klein, dat deze ofwel verwaarloosbaar zijn ofwel relatief eenvoudig te verwerken zijn in het probleem. Eerst wordt het deel van de cliëntenplanning beschreven en daarna het deel van de personeelsplanning.

3.1 Cliëntenplanning

Met de cliëntenplanning wordt een koppeling gemaakt tussen de afspraken van cliënten en de thuiszorgmedewerkers. Hier wordt bepaald welke medewerker op welk tijdstip naar welke cliënt gaat. De cliëntenplanning is een weekplanning. Dat wil zeggen dat er telkens voor een week gepland wordt. De cliëntenplanning is afhankelijk van de personeelsplanning, die verderop beschreven wordt.

3.1.1 Thuiszorgmedewerkers

De thuiszorgmedewerkers (in het vervolg: medewerkers) vormen de kern van de thuiszorg; zij verlenen de thuiszorg bij de cliënten. Verschillende medewerkers kunnen verschillende bevoegdheden hebben. Zo mag een verzorgende bepaalde taken niet uitvoeren, die een verpleegkundige wel mag uitvoeren. Bij CARINTREGGELAND zijn er drie verschillende bevoegdheidsniveaus: verzorgende, verzorgende plus en verpleegkundige. De bevoegdheden zijn geordend; dat wil zeggen dat een verzorgende plus ook alle taken van een verzorgende mag uitvoeren en dat een verpleegkundige ook alle taken van een verzorgende plus en een verzorgende mag uitvoeren.

Verder werken verschillende medewerkers op verschillende momenten. Welke medewerker tijdens welke dienst werkt, wordt vastgelegd in de personeelsplanning. Het is ook van belang om de start- en eindlocatie van de medewerker te kennen. Dit zal in de meeste gevallen het thuisadres zijn van de medewerker, maar kan ook elders zijn.

Een medewerker karakteriseren we door het toekennen van een medewerkersnummer, met daaraan gekoppeld de volgende gegevens:

- Het bevoegdheidsniveau van de medewerker.
- Begin- en eindtijden van de verschillende diensten van deze medewerker (deze volgen uit de personeelsplanning).
- Een start- en eindlocatie.

3.1.2 Cliënten

De cliënten zijn de personen die de thuiszorg ontvangen. Als er een nieuwe cliënt komt bij CARINTREGGELAND, dan vindt er eerst een intakegesprek plaats. Tijdens dit gesprek wordt, in overleg met de cliënt en gebaseerd op de zorgindicatie van het Centrum Indicatiestelling Zorg (CIZ), het aantal zorgmomenten en de momenten in de week dat deze zorgmomenten plaatsvinden, bepaald. Vanuit het zorgkantoor wordt de eis gesteld dat een cliënt niet van teveel verschillende medewerkers zorg mag ontvangen [23]. Om aan deze eis te kunnen voldoen is het van belang het aantal zorgmomenten per week van een cliënt te weten, en daarmee ook het maximale aantal medewerkers dat zorg mag leveren aan de cliënt. Verder is het nodig te weten welke medewerkers gedurende de afgelopen weken bij een cliënt zijn geweest, omdat dit invloed heeft op welke medewerkers nog zorg mogen verlenen bij deze cliënt. Ook heeft elke cliënt een zogeheten eerst verantwoordelijke verzorgende (evv'er). Deze evv'er is voor de cliënt het eerste aanspreekpunt binnen CARINTREGGELAND.

Een cliënt karakteriseren we door het toekennen van een cliëntnummer, met daaraan gekoppeld de volgende gegevens:

- Het maximale aantal medewerkers dat zorg mag leveren aan de cliënt (dit hangt af van het aantal zorgmomenten in de week [23]).
- De medewerkers die gedurende de afgelopen weken bij een cliënt zijn geweest.
- De evv'er van de cliënt.

3.1.3 Zorgmomenten

De zorgmomenten (in het vervolg: afspraken) zijn de momenten waarop een medewerker bij een cliënt langsgaat om zorg te verlenen. Tijdens het intakegesprek met de cliënt wordt tot op zekere hoogte vastgelegd op welke tijdstippen deze afspraken beginnen en hoe lang deze duren (dat laatste is ook afhankelijk van de indicatie van het CIZ). Als een cliënt bijvoorbeeld om 8 uur geholpen wil worden, dan wordt er afgesproken dat er tussen half 8 en half 9 een medewerker langskomt. Deze periode noemen we het tijdsvenster waarbinnen de afspraak moet beginnen. Ook is het nodig te weten welke bevoegdheid van een medewerker minimaal nodig is om de afspraak uit te kunnen voeren. Dit is afhankelijk van de indicatie die de cliënt heeft.

Een afspraak karakteriseren we door het toekennen van een afspraaknummer, met daaraan gekoppeld de volgende gegevens:

- Het cliëntnummer van de cliënt voor wie de afspraak bedoeld is.
- De locatie van de afspraak (in de meeste gevallen zal dit het adres van de cliënt zijn).
- De tijdsduur van de afspraak.
- Het tijdsvenster waarbinnen de afspraak moet beginnen.
- Het minimale bevoegdheidsniveau dat voor de afspraak nodig is.

3.1.4 Reistijden

Om een goede cliëntenplanning te kunnen maken, zijn de reistijden tussen elk tweetal verschillende adressen van belang. Hiermee kan bepaald worden of een medewerker genoeg tijd heeft om van een bepaalde afspraak naar de volgende afspraak te kunnen gaan. Als alle reistijden erg kort zijn, dan is het ook mogelijk om deze te verwaarlozen.

3.1.5 Randvoorwaarden

Een oplossing voor het probleem van het maken van de cliëntenplanning wordt gegeven door een compleet schema waarin elke medewerker een aantal afspraken toegewezen heeft gekregen en waarbij aan elke afspraak een begintijdstip gekoppeld is. Een dergelijk schema moet aan de volgende randvoorwaarden voldoen:

- Elke afspraak moet daadwerkelijk ingepland worden.
- Het bevoegdheidsniveau van een medewerker moet groot genoeg zijn om alle afspraken die zijn toegewezen aan deze medewerker uit te mogen voeren.
- Elke afspraak moet beginnen binnen het gegeven tijdsvenster.
- Alle afspraken die toegewezen zijn aan een bepaalde medewerker, moeten binnen de diensten van deze medewerker vallen.
- Tussen de begintijdstippen van twee opeenvolgende afspraken van een bepaalde medewerker moet genoeg tijd zitten om de eerste afspraak uit te kunnen voeren en daarna verder te reizen naar de tweede afspraak.

- Elke cliënt heeft elke week minimaal één afspraak die door zijn of haar evv'er wordt uitgevoerd.
- Het maximale aantal medewerkers dat zorg mag leveren aan de cliënt, mag niet overschreden worden [23].

3.1.6 Doelen

Binnen de gegeven randvoorwaarden, is het belangrijkste doel tijdens het maken van de cliëntenplanning het minimaliseren van de organisatietijd. Dit is het totaal van reistijd en wachttijd (tijd tussen twee opeenvolgende afspraken die niet als reistijd wordt gezien). CARINTREGGELAND krijgt alleen een vergoeding voor de zorg die een medewerker levert bij een cliënt. Dit betekent dat de reistijd tussen twee opeenvolgende afspraken en de eventuele wachttijd van medewerkers door CARINTREGGELAND zelf betaald moet worden.

Bij het maken van de cliëntenplanning is het ook van belang ervoor te zorgen dat deze robuust is; enerzijds om ervoor te zorgen dat een medewerker eventuele vertragingen zelf op kan vangen, anderzijds om ervoor te zorgen dat het mogelijk is onverwachte verstoringen zo goed mogelijk op te vangen. Dergelijke verstoringen ontstaan bijvoorbeeld als een cliënt ineens meer of juist minder zorg nodig heeft of als er een nieuwe cliënt komt.

3.2 Personeelsplanning

Met de personeelsplanning wordt er een koppeling gemaakt tussen de medewerkers en de verschillende diensten. Hier wordt bepaald welke medewerkers tijdens welke diensten zullen werken. De personeelsplanning is een maandplanning. Dat wil zeggen dat er telkens voor een maand gepland wordt. De personeelsplanning is onder andere afhankelijk van de totale zorgvraag, die voorspeld kan worden aan de hand van (oude) cliëntenplanningen.

3.2.1 Medewerkers

Voor het maken van een personeelsplanning is het van belang de beschikbaarheid van de medewerkers te kennen. Zij kunnen (in overleg) hun vrije dagen en vakanties doorgeven. Op alle andere dagen zijn ze beschikbaar om ingezet te worden. Net als bij de cliëntenplanning, is ook bij de personeelsplanning het bevoegdheidsniveau van de medewerkers van belang. Ook het aantal contracturen dat een bepaalde medewerker heeft is nodig voor het maken van de personeelsplanning.

In de personeelsplanning karakteriseren we een medewerker door het toekennen van een medewerkersnummer, met daaraan gekoppeld de volgende gegevens:

- De beschikbaarheid (voor elke dag van de te plannen maand).
- Het bevoegdheidsniveau van de medewerker.
- Het aantal contracturen van de medewerker.

3.2.2 Diensten

Elke dag is verdeeld in een aantal diensten. Voor elke dienst kan het aantal uur dat een medewerker ongeveer zal werken tijdens die dienst, bepaald worden. Dit zullen we het aantal diensturen van de dienst noemen. Ook kan er voor elke dienst gekeken worden hoeveel medewerkers met een bepaalde bevoegdheid nodig zijn om alle afspraken die in deze dienst plaatsvinden, uit te kunnen voeren.

In de personeelsplanning karakteriseren we een dienst door het toekennen van een dienstnummer, met daaraan gekoppeld de volgende gegevens:

- De dag waarop de dienst plaatsvindt.
- Het aantal diensturen van de dienst.
- Voor elk bevoegdheidsniveau het aantal benodigde medewerkers met dat bevoegdheidsniveau dat nodig is.

3.2.3 Randvoorwaarden

Een oplossing voor het probleem van het maken van de personeelsplanning wordt gegeven door een compleet schema waarin aan elke dienst een aantal medewerkers zijn toegewezen. Een dergelijk schema moet in ieder geval aan de volgende randvoorwaarden voldoen:

- Aan elke dienst moeten voldoende medewerkers toegewezen worden (hierbij moet ook rekening gehouden worden met de verschillende bevoegdheidsniveaus).
- Een medewerker hoeft niet te werken op een dag waarop deze niet beschikbaar is.
- Een medewerker mag niet aan meer diensten op een dag worden toegewezen dan toegestaan.
- Het totaal aantal diensturen van de diensten waaraan een medewerker is toegewezen, moet ongeveer gelijk zijn aan het aantal contracturen van deze medewerker.
- De diensten moeten worden toegekend aan de medewerkers op een manier zodat het voor elke cliënt in de cliëntenplanning mogelijk is om gezien te worden door zijn of haar evv'er en niet te veel verschillende medewerkers te zien.

3.2.4 Doelen

Binnen de gegeven randvoorwaarden, is het belangrijkste doel tijdens het maken van de personeelsplanning het maximaliseren van de eerlijkheid van de verdeling van de diensten. Dit wordt bereikt door de verschillen tussen de medewerkers in de verhouding van het aantal geplande diensturen en het aantal contracturen te minimaliseren. Op dezelfde manier worden ook de weekenddiensten zo eerlijk mogelijk verdeeld.

4 Modelling

In dit hoofdstuk wordt een wiskundige formulering gegeven van het probleem dat is beschreven in het vorige hoofdstuk. Eerst wordt er gekeken naar het deel van de cliëntenplanning en daarna naar het deel van de personeelsplanning.

4.1 Modelling cliëntenplanning

In deze paragraaf wordt het probleem van het maken van de cliëntenplanning, dat beschreven is in paragraaf 3.1, vertaald naar een wiskundig model. Eerst worden de variabelen die we gaan gebruiken, gedefinieerd en daarna wordt de rest van het model beschreven. Ten slotte wordt ook nog een ILP-formulering voor het probleem van het maken van de cliëntenplanning gegeven.

4.1.1 Variabelen en definities

Voordat er dieper wordt ingegaan op het model, worden een aantal variabelen en definities gegeven. Deze hebben betrekking op de medewerkers, de afspraken en de cliënten. Binnen het model zijn er twee soorten variabelen. In deze paragraaf worden de variabelen benoemd, waarvan de waarde vooraf bekend is. In de volgende paragraaf komen de variabelen aan bod die gebruikt worden om de planning te representeren. De waarden van deze planningsvariabelen zijn vooraf niet bekend.

Het totaal aantal medewerkers wordt m genoemd. Zoals is beschreven in paragraaf 3.1.1, heeft elke medewerker een bepaalde bevoegdheid. Ook is voor elke dienst op elke dag bekend of een medewerker werkt tijdens die dienst en wat de begin- en eindtijden zijn van die dienst als een medewerker werkt. Het totaal aantal diensten op een dag wordt J genoemd en er zijn in totaal aantal D dagen. De volgende variabelen worden gebruikt in verwijzing naar deze gegevens van de medewerkers.

- s_i De bevoegdheid van medewerker i .
- T_{ijk}^{start} Het begintijdstip van dienst j van medewerker i op dag k . Als medewerker i niet werkt tijdens dienst j op dag k , dan geldt $T_{ijk}^{start} = 0$.
- T_{ijk}^{end} Het eindtijdstip van dienst j van medewerker i op dag k . Als medewerker i niet werkt tijdens dienst j op dag k , dan geldt $T_{ijk}^{end} = 0$.

Het totaal aantal afspraken wordt n genoemd. Zoals is beschreven in paragraaf 3.1.3, is elke afspraak gekoppeld aan een cliënt en een locatie. Verder zijn voor alle afspraken de tijdsduur en het tijdsvenster waarbinnen de afspraak moet beginnen, bekend en is ook bekend welke bevoegdheid minimaal nodig is om de afspraak uit te kunnen voeren. De volgende variabelen worden gebruikt in verwijzing naar deze gegevens van de afspraken.

- c_q Het nummer van de cliënt voor wie afspraak q bedoeld is.
- a_q De locatie waar afspraak q plaatsvindt (meestal het adres van cliënt c_q).
- t_q^{dur} De tijdsduur in minuten van afspraak q .
- t_q^{start} Het begin van het tijdsvenster waarin afspraak q moet beginnen.
- t_q^{end} Het eind van het tijdsvenster waarin afspraak q moet beginnen.
- ℓ_q De minimale bevoegdheid die nodig is voor afspraak q .

Het totaal aantal cliënten wordt C genoemd. Zoals is beschreven in paragraaf 3.1.2, is er voor elke cliënt een maximaal aantal zorgverleners dat ingezet mag worden bij deze cliënt. Ook is er bekend welke medewerkers gedurende de afgelopen weken zijn ingezet bij de cliënt. Verder heeft elke cliënt een eerst verantwoordelijke verzorgende (evv'er). De volgende variabelen worden gebruikt in verwijzing naar deze gegevens van de cliënten.

- z_c^{max} Het maximale aantal zorgverleners dat ingezet mag worden bij cliënt c .

- Z_c^{oud} De verzameling van alle medewerkers die tijdens de afgelopen weken ingezet zijn bij cliënt c .
 e_c De evv'er van cliënt c .

De verschillende tijdstippen die gebruikt worden in het model, worden gegeven door het aantal minuten na middernacht op de eerste dag van de planning. Zo staat tijdstip 4800 voor 8:00 uur op de vierde dag van de planning en staat tijdstip 1140 voor 19:00 uur op de eerste dag. Ten slotte worden de reistijden als volgt gerepresenteerd in het model.

- t_{ij} De reistijd in minuten van locatie i naar locatie j .

4.1.2 Planningsvariabelen

Naast de variabelen waarvan de waarde vooraf bekend is, worden in het model ook planningsvariabelen gebruikt, waarvan de waarde vooraf niet bekend is. Deze planningsvariabelen worden gebruikt om de planning te representeren.

De afspraken worden in een schema geplaatst, dat voor iedere medewerker uit een lijst van afspraaknummers zal bestaan. Deze lijst zal worden opgeslagen in een 3-dimensionale matrix, waarin elke rij hoort bij een bepaalde dienst op een bepaalde dag. De volgorde van de afspraaknummers in deze rijen komt overeen met de volgorde waarin de afspraken worden uitgevoerd. De volgende variabelen worden gebruikt in verwijzing naar de medewerkers en hun geplande routes.

- \mathbf{r}_i De route van medewerker i (een 3-dimensionale matrix bestaande uit afspraaknummers).
 $\mathbf{r}_{(i,p,j,k)}$ Het afspraaknummer van de p^e afspraak tijdens dienst j op dag k van medewerker i .
 w_{ijk} Het aantal afspraken in de route van medewerker i tijdens dienst j op dag k . Dus \mathbf{r}_i is een $W \times J \times D$ -matrix met $W := \max_j(w_{ijk})$.

Tijdens het maken van de planning wordt voor elke afspraak een begintijdstip bepaald. Voor dit begintijdstip wordt de volgende variabele gebruikt.

- t_q^{plan} Het geplande tijdstip waarop afspraak q begint.

Tijdens het maken van de planning moet ook worden bijgehouden welke medewerkers bij een bepaalde cliënt worden ingezet. Hierdoor kan eenvoudig gecontroleerd worden of elke cliënt door zijn of haar evv'er wordt gezien en of het maximale aantal medewerkers dat zorg mag leveren aan de cliënt, niet wordt overschreden. De volgende variabelen worden gebruikt in verwijzing naar deze verzamelingen van medewerkers.

- Z_c^{huidig} De verzameling van alle medewerkers die deze week ingezet worden bij cliënt c . Er geldt $Z_c^{huidig} = \{j \mid \exists i, f, g, h (c = c_i \wedge i = \mathbf{r}_{(j,f,g,h)})\}$.
 Z_c^{nieuw} De verzameling van alle medewerkers die na deze week ingezet zijn bij cliënt c . Er geldt $Z_c^{nieuw} = Z_c^{oud} \cup Z_c^{huidig}$.

Nadat de planning voor een bepaalde week is gemaakt en daarbij de bijbehorende verzamelingen Z_c^{nieuw} , kunnen deze verzamelingen in de daaropvolgende week gebruikt worden als Z_c^{oud} in de input van het probleem.

4.1.3 Randvoorwaarden

Met behulp van alle hiervoor gedefinieerde variabelen worden de randvoorwaarden waaraan een planning moet voldoen, geformuleerd. Als eerste is het erg belangrijk dat elke afspraak uitgevoerd wordt door een medewerker die voldoende bevoegdheid heeft. Er wordt aangenomen dat de bevoegdheden oplopend zijn. Dat wil zeggen dat een medewerker met bevoegdheid s taken mag uitvoeren waarvoor bevoegdheid $\ell \leq s$ nodig is. Deze voorwaarde wordt als volgt geformuleerd:

$$s_i \geq \ell_j \quad \forall i \in \{1, 2, \dots, m\} \quad \forall j \in \mathbf{r}_i \quad (1)$$

Daarnaast is het van belang dat alle afspraken beginnen binnen het daarvoor opgestelde tijdsvenster. Dit wordt als volgt geformuleerd:

$$t_q^{start} \leq t_q^{plan} \leq t_q^{end} \quad \forall q \in \{1, 2, \dots, n\} \quad (2)$$

Ook is het van belang dat een medewerker niet te vroeg hoeft te beginnen, genoeg tijd heeft om naar de volgende afspraak te gaan en niet later klaar is dan het einde van haar dienst. Deze drie voorwaarden worden als volgt geformuleerd:

$$T_{ijk}^{start} \leq t_{\mathbf{r}_{(i,1,j,k)}}^{plan} \quad \forall j \in \{1, 2, \dots, J\} \quad \forall k \in \{1, 2, \dots, D\} \quad \forall i \in \{1, 2, \dots, m\} \quad (3)$$

$$t_{\mathbf{r}_{(i,p,j,k)}}^{plan} + t_{\mathbf{r}_{(i,p,j,k)}}^{dur} + t_{a_{\mathbf{r}_{(i,p,j,k)}}} a_{\mathbf{r}_{(i,p+1,j,k)}} \leq t_{\mathbf{r}_{(i,p+1,j,k)}}^{plan} \quad \forall j \in \{1, 2, \dots, J\} \quad \forall k \in \{1, 2, \dots, D\} \\ \forall i \in \{1, 2, \dots, m\} \quad \forall p \in \{1, 2, \dots, w_{ijk} - 1\} \quad (4)$$

$$t_{\mathbf{r}_{(i,w_{ijk},j,k)}}^{plan} + t_{\mathbf{r}_{(i,w_{ijk},j,k)}}^{dur} \leq T_{ijk}^{end} \quad \forall j \in \{1, 2, \dots, J\} \quad \forall k \in \{1, 2, \dots, D\} \quad \forall i \in \{1, 2, \dots, m\} \quad (5)$$

Verder wordt er nog een voorwaarde gesteld aan het maximale aantal medewerkers dat gezien mag worden door een cliënt. Deze voorwaarde wordt als volgt geformuleerd:

$$|Z_c^{nieuw}| \leq z_c^{max} \quad \forall c \in \{1, 2, \dots, C\} \quad (6)$$

Ook moet elke cliënt minstens één keer per week gezien worden door zijn of haar evv'er. Dit wordt als volgt geformuleerd:

$$e_c \in Z_c^{huidig} \quad \forall c \in \{1, 2, \dots, C\} \quad (7)$$

4.1.4 Doelen

De doelen die geformuleerd zijn in paragraaf 3.1.6, kunnen nu beschreven worden met behulp van de gedefinieerde variabelen. De totale organisatietijd voor een planning wordt verkregen door voor elk tweetal opeenvolgende afspraken $\mathbf{r}_{(p,j,k)}$ en $\mathbf{r}_{(p+1,j,k)}$ het verschil tussen de eindtijd van de eerste afspraak en de begintijd van de tweede afspraak te nemen, en al die waarden bij elkaar op te tellen:

$$\text{totale organisatietijd} = \sum_{i=1}^m \sum_{j=1}^J \sum_{k=1}^D \sum_{p=1}^{w_{ijk}-1} \left(t_{\mathbf{r}_{(p+1,j,k)}}^{plan} - \left(t_{\mathbf{r}_{(p,j,k)}}^{plan} + t_{\mathbf{r}_{(p,j,k)}}^{dur} \right) \right).$$

De robuustheid van de planning wordt bepaald door voor iedere afspraak te kijken naar de afwijking van het geplande begintijdstip ten opzichte van het gewenste begintijdstip (zijnde het midden van het tijdsvenster waarin de afspraak moet beginnen). De robuustheid van de planning wordt groter naarmate het totaal van deze afwijkingen kleiner wordt. Deze robuustheidsindicator wordt dus gegeven door

$$\sum_{i=1}^n \left| \frac{1}{2} (t_i^{start} + t_i^{end}) - t_i^{plan} \right|.$$

De te minimaliseren doelfunctie f van het model wordt nu gegeven door

$$f = \alpha \sum_{i=1}^m \sum_{j=1}^J \sum_{k=1}^D \sum_{p=1}^{w_{ijk}-1} \left(t_{\mathbf{r}_{(p+1,j,k)}}^{plan} - \left(t_{\mathbf{r}_{(p,j,k)}}^{plan} + t_{\mathbf{r}_{(p,j,k)}}^{dur} \right) \right) + \beta \sum_{i=1}^n \left| \frac{1}{2} (t_i^{start} + t_i^{end}) - t_i^{plan} \right|, \quad (8)$$

waarin α en β nader te bepalen constanten zijn. Deze constanten geven aan hoeveel belang er gehecht wordt aan beide componenten van de doelfunctie.

4.1.5 ILP-formulering

In de voorgaande vier paragrafen is een wiskundige formulering gegeven van het model voor de cliëntenplanning. Deze formulering is echter niet lineair. Daarom wordt in deze paragraaf een ILP-formulering van (bijna) hetzelfde wiskundige model gegeven. Een deel van de variabelen in deze ILP-formulering komt overeen met de variabelen zoals hiervoor gedefinieerd, maar er worden ook een aantal nieuwe variabelen gebruikt. Voor de volledigheid worden daarom eerst opnieuw alle variabelen voor de ILP-formulering gedefinieerd, waarbij opgemerkt wordt dat onderstaande definities alleen gebruikt worden in deze paragraaf.

In deze ILP-formulering worden aan de bestaande lijst afspraken een aantal nulafspraken toegevoegd. Deze nulafspraken zijn de afspraken met nummer $n + 1$ tot en met $n + W$ (met n en W zoals hieronder gedefinieerd). Deze afspraken vormen het begin van elke dienst die gewerkt wordt door een medewerker. Het gebruik van deze nulafspraken is in de ILP-formulering nodig om te voorkomen dat een medewerker tijdens een bepaalde dienst twee verschillende routes tegelijkertijd moet lopen. Verder is het in de ILP-formulering toegestaan dat een afspraak niet ingepland wordt of dat een cliënt niet door zijn of haar evv'er wordt gezien. In beide gevallen wordt er wel een penalty toegevoegd aan de doelfunctiewaarde. Op deze manier kan ook voor een infeasible dataset nog een 'beste' planning gevonden worden. Allereerst volgen nu in de tabel hieronder alle bekende variabelen en constanten.

a_{cj}	Beslissingsvariabele: gelijk aan 1 als afspraak j bedoeld is voor cliënt c en anders gelijk aan 0.
C	Het totaal aantal cliënten.
d_j	De dienst waarin afspraak j plaats moet vinden.
D	Het totaal aantal diensten in een planningsperiode.
e_c	De eerst verantwoordelijke verzorgende (evv'er) van cliënt c .
ℓ_j	De minimale bevoegdheid die nodig is voor afspraak j .
M	Een groot getal.
m	Het totaal aantal medewerkers.
n	Het totaal aantal in te plannen afspraken.
s_i	De bevoegdheid van medewerker i .
t_{ij}	De reistijd van de locatie van afspraak i naar de locatie van afspraak j .
t_j^{dur}	De tijdsduur van afspraak j .
t_j^{end}	Het eind van het tijdsvenster waarin afspraak j moet beginnen.
t_j^{start}	Het begin van het tijdsvenster waarin afspraak j moet beginnen.
w_{di}	Beslissingsvariabele: gelijk aan 1 als medewerker i werkt tijdens dienst d en anders gelijk aan 0.
W	Het totaal aantal diensten dat gewerkt wordt (gelijk aan $\sum_{d=1}^D \sum_{i=1}^m w_{di}$).
y_{ji}	Beslissingsvariabele: gelijk aan 1 als afspraak j ($n + 1 \leq j \leq n + W$) hoort bij medewerker i en anders gelijk aan 0.
z_c^{max}	Het maximale aantal zorgverleners dat ingezet mag worden bij cliënt c .
z_{ci}^{oud}	Beslissingsvariabele: gelijk aan 1 als medewerker i in de afgelopen maanden bij cliënt c is ingezet en anders gelijk aan 0.
α	Constante factor voor de organisatietijd in de doelfunctie.
β	Constante factor voor de robuustheid in de doelfunctie.

Naast deze bekende variabelen, worden ook nog de volgende, onbekende, variabelen gebruikt, waarvan de optimale waarde bepaald wordt door het oplossen van het ILP.

r_c	Beslissingsvariabele: gelijk aan 1 als cliënt c deze week gezien wordt door zijn of haar evv'er en anders gelijk aan 0.
t_j^{afw}	De absolute waarde van de afwijking van het geplande begintijdstip van afspraak j ($1 \leq j \leq n$) (gelijk aan $ x_j = x_j^+ + x_j^-$).
t_j^{plan}	Het geplande tijdstip waarop afspraak j ($1 \leq j \leq n$) begint.
t_{ij}^{tussen}	De tijd tussen twee afspraken i en j ($1 \leq i, j \leq n$) (gelijk aan $t_j^{plan} - (t_i^{plan} + t_i^{dur})$).

v_{ij}	Beslissingsvariabele: gelijk aan 1 als afspraak j ($1 \leq j \leq n$) direct volgt op afspraak i ($1 \leq i \leq n + W$) en anders gelijk aan 0.
x_j	De afwijking van het geplande begintijdstip van afspraak j ($1 \leq j \leq n$) (gelijk aan $\frac{1}{2}(t_j^{start} + t_j^{end}) - t_j^{plan}$).
x_j^+	Het positieve gedeelte van x_j .
x_j^-	Het negatieve gedeelte van x_j .
y_{ji}	Beslissingsvariabele: gelijk aan 1 als afspraak j ($1 \leq j \leq n$) wordt uitgevoerd door medewerker i en anders gelijk aan 0.
z_{ci}^{huidig}	Het aantal keer dat medewerker i deze week een afspraak van cliënt c uitvoert.
z_{ci}^{nieuw}	Beslissingsvariabele: gelijk aan 1 als i na deze week in de afgelopen maanden bij cliënt c is ingezet en anders gelijk aan 0.
δ_{ij}	Gelijk aan $v_{ij} t_{ij}^{tussen}$.
ξ	Het aantal niet-ingeplande afspraken.
ψ	Het aantal cliënten dat niet door de evv'er wordt gezien in deze week.

Met al deze variabelen kan nu de volgende ILP-formulering gegeven worden.

$$\text{Minimaliseer } \alpha \sum_{i=1}^n \sum_{j=1}^n \delta_{ij} + \beta \sum_{j=1}^n t_j^{afw} + M\xi + \frac{M\psi}{10}$$

s.t.

$$-t_j^{plan} \leq -t_j^{start} \quad \forall j \in \{1, \dots, n\} \quad (9)$$

$$t_j^{plan} \leq t_j^{end} \quad \forall j \in \{1, \dots, n\} \quad (10)$$

$$\sum_{i=1}^m y_{ji} \leq 1 \quad \forall j \in \{1, \dots, n\} \quad (11)$$

$$y_{ji} \leq w_{d,i} \quad \forall i \in \{1, \dots, m\} \quad \forall j \in \{1, \dots, n\} \quad (12)$$

$$Mv_{ij} + My_{ik} - y_{jk} \leq 2M - 1 \quad \forall i \in \{1, \dots, n + W\} \quad \forall j \in \{1, \dots, n\} \quad \forall k \in \{1, \dots, m\} \quad (13)$$

$$Mv_{ij} \leq M + d_i - d_j \quad \forall i \in \{1, \dots, n + W\} \quad \forall j \in \{1, \dots, n\} \quad (14)$$

$$Mv_{ij} \leq M + d_j - d_i \quad \forall i \in \{1, \dots, n + W\} \quad \forall j \in \{1, \dots, n\} \quad (15)$$

$$My_{ji} \leq M + s_i - \ell_j \quad \forall i \in \{1, \dots, m\} \quad \forall j \in \{1, \dots, n\} \quad (16)$$

$$\sum_{i=1}^{n+W} v_{ij} - \sum_{i=1}^m y_{ji} = 0 \quad \forall j \in \{1, \dots, n\} \quad (17)$$

$$\sum_{j=1}^n v_{ij} \leq 1 \quad \forall i \in \{1, \dots, n + W\} \quad (18)$$

$$t_i^{plan} - t_j^{plan} + Mv_{ij} \leq M - t_i^{dur} - t_{ij} \quad \forall i \in \{1, \dots, n\} \quad \forall j \in \{1, \dots, n\} \quad (19)$$

$$z_{ci}^{huidig} - \sum_{j=1}^n a_{cj} y_{ji} = 0 \quad \forall c \in \{1, \dots, C\} \quad \forall i \in \{1, \dots, m\} \quad (20)$$

$$\frac{1}{M} z_{ci}^{huidig} - z_{ci}^{nieuw} \leq -\frac{1}{M} z_{ci}^{oud} \quad \forall c \in \{1, \dots, C\} \quad \forall i \in \{1, \dots, m\} \quad (21)$$

$$z_{ci}^{nieuw} - z_{ci}^{huidig} \leq z_{ci}^{oud} \quad \forall c \in \{1, \dots, C\} \quad \forall i \in \{1, \dots, m\} \quad (22)$$

$$\sum_{i=1}^m z_{ci}^{nieuw} \leq z_c^{max} \quad \forall c \in \{1, \dots, C\} \quad (23)$$

$$t_j^{plan} - t_i^{plan} - t_{ij}^{tussen} = t_i^{dur} \quad \forall i \in \{1, \dots, n\} \quad \forall j \in \{1, \dots, n\} \quad (24)$$

$$t_{ij}^{tussen} - \delta_{ij} + Mv_{ij} \leq M \quad \forall i \in \{1, \dots, n\} \quad \forall j \in \{1, \dots, n\} \quad (25)$$

$$\delta_{ij} - t_{ij}^{tussen} + Mv_{ij} \leq M \quad \forall i \in \{1, \dots, n\} \quad \forall j \in \{1, \dots, n\} \quad (26)$$

$$\delta_{ij} - Mv_{ij} \leq 0 \quad \forall i \in \{1, \dots, n\} \quad \forall j \in \{1, \dots, n\} \quad (27)$$

$$t_j^{plan} + x_j = \frac{1}{2}t_j^{start} + \frac{1}{2}t_j^{end} \quad \forall j \in \{1, \dots, n\} \quad (28)$$

$$x_j^+ - x_j^- - x_j = 0 \quad \forall j \in \{1, \dots, n\} \quad (29)$$

$$x_j^+ + x_j^- - t_j^{afw} = 0 \quad \forall j \in \{1, \dots, n\} \quad (30)$$

$$\xi + \sum_{j=1}^n \sum_{i=1}^m y_{ji} = n \quad (31)$$

$$r_c - z_{cc}^{huidig} \leq 0 \quad \forall c \in \{1, \dots, C\} \quad (32)$$

$$\frac{1}{M}z_{cc}^{huidig} - r_c \leq 0 \quad \forall c \in \{1, \dots, C\} \quad (33)$$

$$\psi + \sum_{c=1}^C r_c = C \quad (34)$$

$$\delta_{ij} \geq 0 \quad \forall i \in \{1, \dots, n\} \quad \forall j \in \{1, \dots, n\} \quad (35)$$

$$x_j^+ \geq 0 \quad \forall j \in \{1, \dots, n\} \quad (36)$$

$$x_j^- \geq 0 \quad \forall j \in \{1, \dots, n\} \quad (37)$$

$$y_{ji} \in \{0, 1\} \quad \forall i \in \{1, \dots, m\} \quad \forall j \in \{1, \dots, n\} \quad (38)$$

$$v_{ij} \in \{0, 1\} \quad \forall i \in \{1, \dots, n+W\} \quad \forall j \in \{1, \dots, n\} \quad (39)$$

$$z_{ci}^{nieuw} \in \{0, 1\} \quad \forall c \in \{1, \dots, C\} \quad \forall i \in \{1, \dots, m\} \quad (40)$$

$$r_c \in \{0, 1\} \quad \forall c \in \{1, \dots, C\} \quad (41)$$

In deze ILP-formulering zorgen (9) en (10) ervoor dat elke afspraak begint binnen het daarvoor bestemde tijdsvenster. Vervolgens garandeert (11) dat elke afspraak door maximaal één medewerker wordt uitgevoerd en zorgt (12) ervoor dat een afspraak alleen toegekend kan worden aan een medewerker als deze afspraak plaatsvindt tijdens een dienst waarin de medewerker werkt. (13) garandeert dat twee afspraken die elkaar opvolgen door dezelfde medewerker worden uitgevoerd en (14) en (15) garanderen dat (nul)afspraken die bij verschillende diensten horen niet elkaar kunnen opvolgen. (16) zorgt ervoor dat iedere afspraak die ingepland wordt, wordt uitgevoerd door een medewerker met voldoende bevoegdheid. Verder zorgt (17) ervoor dat elke afspraak die ingepland wordt, precies één (nul)afspraak opvolgt en dat elke afspraak die niet ingepland wordt geen (nul)afspraak opvolgt. (18) zorgt ervoor dat elke (nul)afspraak wordt opgevolgd door maximaal één afspraak en (19) zorgt ervoor dat er voldoende tijd zit tussen de begintijden van twee afspraken die elkaar opvolgen.

Voor het correct bijhouden van z_{ci}^{huidig} wordt (20) gebruikt, waarna de waarde van z_{ci}^{nieuw} correct wordt vastgelegd door (21) en (22). Vervolgens garandeert (23) dat er niet teveel medewerkers bij een bepaalde cliënt langskomen. Voor het correct bijhouden van t_{ij}^{tussen} wordt (24) gebruikt, waarna de waarde van δ_{ij} correct wordt vastgelegd door (25), (26) en (27). Voor het correct bijhouden van x_j wordt (28) gebruikt, waarna x_j^+ en x_j^- volgen uit (29). In (30) wordt t_j^{afw} bepaald. Het aantal niet-ingeplande afspraken wordt correct vastgelegd door (31). Voor het correct bijhouden van r_c worden (32) en (33) gebruikt, waarna het aantal cliënten dat niet gezien wordt door zijn of haar evv'er correct wordt vastgelegd door (34).

Ten slotte zorgen (35), (36) en (37) ervoor dat δ_{ij} , x_j^+ en x_j^- niet-negatief zijn en zorgen (38)-(41) ervoor dat de beslissingsvariabelen alleen maar de waarden 0 en 1 kunnen aannemen.

Deze ILP-formulering wijkt op een paar plaatsen af van het oorspronkelijke model. Om terug te keren naar het oorspronkelijke model moeten de laatste twee termen uit de doelfunctie verwijderd worden en moeten ook de randvoorwaarden (31)-(34) en (41) verwijderd worden. Verder moet ongelijkheid (11) veranderd worden in een gelijkheid, zodat elke afspraak aan een medewerker word gekoppeld.

4.2 Modelling personeelsplanning

In deze paragraaf wordt het probleem van het maken van de personeelsplanning, dat beschreven is in paragraaf 3.2, vertaald naar een wiskundig model. Eerst worden de variabelen gedefinieerd die gebruikt gaan worden en daarna wordt de rest van het model beschreven.

4.2.1 Variabelen

Er zijn in totaal m medewerkers. Van elke medewerker is voor elke dienst en dag haar beschikbaarheid bekend. Ook heeft elke medewerker een bepaalde bevoegdheid. Verder is voor elke medewerker het totaal aantal contracturen bekend. De volgende variabelen worden gebruikt in verwijzing naar deze gegevens van de medewerkers.

- B_{ijk} De beschikbaarheid van medewerker i voor dienst j op dag k . Als medewerker i beschikbaar is voor dienst j op dag k , dan geldt $B_{ijk} = 1$ en anders geldt $B_{ijk} = 0$.
- s_i De bevoegdheid van medewerker i .
- C_i Het aantal contracturen van medewerker i (voor D dagen).

Er zijn in totaal D dagen waarvoor medewerkers ingepland moeten worden. Per dag zijn er J diensten. Om het werk eerlijk te kunnen verdelen, is het nodig te weten hoeveel uur een dienst gemiddeld duurt. Daarom is van elke dienst het verwachte aantal diensturen bekend dat bij deze dienst hoort. Ook is het verwachte aantal medewerkers dat nodig is voor een dienst, bekend. De volgende variabelen worden gebruikt in verwijzing naar deze gegevens van de diensten.

- U_{jk} Het verwachte aantal diensturen horende bij dienst j op dag k .
- E_{jkl} Het verwachte aantal medewerkers dat nodig is voor dienst j op dag k met minimale bevoegdheid l .

Verder worden de volgende variabelen gebruikt in het model.

- J_{max} Het maximale aantal diensten dat een medewerker per dag mag werken.
- R_i Het gedeelte aan contracturen dat medewerker i naar verwachting zal moeten werken.
- W_i Het gedeelte aan contracturen dat medewerker i naar verwachting aan weekenddiensten zal moeten werken.
- Y_{ijk} Beslissingsvariabele: gelijk aan 1 als medewerker i is ingepland voor dienst j op dag k en anders gelijk aan 0.

4.2.2 Randvoorwaarden

Met behulp van alle hiervoor gedefinieerde variabelen worden de randvoorwaarden waaraan een personeelsplanning moet voldoen, geformuleerd. Als eerste is het erg belangrijk dat een medewerker i alleen wordt ingepland als zij beschikbaar is. Dit wordt als volgt geformuleerd:

$$Y_{ijk} \leq B_{ijk} \quad \forall i \in \{1, \dots, m\} \quad \forall j \in \{1, \dots, J\} \quad \forall k \in \{1, \dots, D\} \quad (42)$$

Ook is het van belang dat een medewerker niet voor meer diensten ingepland wordt dan het aantal diensten dat een medewerker maximaal per dag mag werken. Dit wordt als volgt geformuleerd

$$\sum_{j=1}^J Y_{ijk} \leq J_{max} \quad \forall i \in \{1, \dots, m\} \quad \forall k \in \{1, \dots, D\} \quad (43)$$

Daarnaast is het van belang dat er voor elke dienst genoeg medewerkers met de juiste bevoegdheid worden ingepland. Er wordt aangenomen dat de bevoegdheden oplopend zijn. Dat wil zeggen dat een medewerker met bijvoorbeeld bevoegdheid 3 taken mag uitvoeren waarvoor bevoegdheid 1, 2 of 3 nodig is. Er zijn L bevoegdheden. Deze

voorwaarde wordt als volgt geformuleerd:

$$\sum_{i|s_i \geq \ell} Y_{ijk} \geq E_{jk\ell} \quad \forall j \in \{1, \dots, J\} \quad \forall k \in \{1, \dots, D\} \quad \forall \ell \in \{1, \dots, L\} \quad (44)$$

Als laatste is er de voorwaarde dat het werk eerlijk over de medewerkers verdeeld moet worden. Dit wordt gedaan door het werk zo te verdelen dat medewerkers ongeveer hetzelfde percentage aan contracturen werken. Verder wordt het aantal weekenddiensten eerlijk verdeeld. Iedere medewerker moet hetzelfde percentage van haar contracturen aan weekenddiensten draaien. Daarvoor moeten de variabelen R_i en W_i voldoen aan de volgende definities:

$$R_i = \frac{\sum_{k=1}^D \sum_{j=1}^J U_{jk} \cdot Y_{ijk}}{C_i} \quad \forall i \in \{1, \dots, m\}, \quad (45)$$

$$W_i = \frac{\sum_{k \in \text{weekend}} \sum_{j=1}^J U_{jk} \cdot Y_{ijk}}{C_i} \quad \forall i \in \{1, \dots, m\}. \quad (46)$$

Hierbij is $\sum_{k=1}^D \sum_{j=1}^J U_{jk} Y_{ijk}$ het verwachte aantal uren dat medewerker i zal gaan werken in de komende D dagen en is $\sum_{k \in \text{weekend}} \sum_{j=1}^J U_{jk} Y_{ijk}$ het verwachte aantal uren weekenddienst dat medewerker i zal gaan werken in de komende D dagen.

4.2.3 Doelfunctie

Alle variabelen zijn nu gedefinieerd samen met de randvoorwaarden. Om het werk zo eerlijk mogelijk te verdelen moet $\max_i R_i - \min_i R_i$ geminimaliseerd worden. Op deze manier komt de percentages aan contracturen die de medewerkers moeten werken zo dicht mogelijk bij elkaar te liggen. Verder moet $\max_i W_i - \min_i W_i$ geminimaliseerd worden om ook de weekenddiensten zo eerlijk mogelijk te verdelen. Er wordt rekening mee gehouden dat het eerlijk verdelen van het werk belangrijker is dan het eerlijk verdelen van de weekenddiensten. Op deze manier wordt de doelfunctie gegeven door $\min f$ met

$$f = \lambda \left(\max_i W_i - \min_i W_i \right) + \mu \left(\max_i R_i - \min_i R_i \right), \quad (47)$$

waarin λ en μ nader te bepalen constanten zijn zodat $\lambda < \mu$.

5 Oplossingsstrategie cliëntenplanning

Het HHCP is een \mathcal{NP} -hard probleem ([22, 24]). Het aantal mogelijke oplossingen voor het HHCP neemt exponentieel toe met de grootte van het probleem. Voor het gebruik van een exact algoritme is daarom al snel een te grote rekenkracht nodig. Daarom is er een heuristiek nodig die snel is en relatief goede oplossingen geeft. Uit de literatuur is gebleken dat Particle Swarm Optimization (PSO) een techniek is die hieraan voldoet en dat PSO daarnaast goede resultaten heeft gegeven voor planningsproblemen in vergelijking met andere heuristieken ([1]). In dit hoofdstuk wordt beschreven hoe PSO gebruikt is om het HHCP op te lossen. Eerst wordt PSO kort in het algemeen toegelicht. Daarna worden de verschillende onderdelen toegelicht. Ten slotte wordt er ook nog gekeken naar de verschillende parameters die binnen PSO een rol spelen.

5.1 Particle Swarm Optimization

PSO is een globale metaheuristische optimalisatiemethode die voor het eerst naar voren is gebracht door Kennedy en Eberhart in 1995 ([15]). De methode is een op metaheuristische techniek die gebaseerd is op het gedrag van populaties. De ontwikkeling hiervan is ontstaan vanuit observaties van het sociale gedrag van dieren, zoals zwermen vogels en scholen vissen. Tijdens het zoeken naar voedsel verspreiden de vogels zich of gaan ze samen op pad, voordat ze de plaats vinden waar het voedsel zich bevindt. Als de vogels op zoek gaan naar voedsel, is er altijd een vogel die goede informatie heeft over de plek waar het voedsel zich bevindt. Omdat de vogels informatie met elkaar uitwisselen, vooral de bruikbare informatie, is er een grote kans dat de vogels naar de plek vliegen waar het voedsel zich bevindt.

PSO probeert dergelijke processen na te bootsen om een goede oplossing te vinden voor een wiskundig probleem. De oplossingsruimte van dit probleem komt overeen met het zoekgebied van de vogels. Binnen PSO worden de vogels vervolgens gerepresenteerd als deeltjes die ‘rondvliegen’ door de oplossingsruimte. Deze deeltjes geven dus tijdens het proces verschillende oplossingen van het probleem weer. De ‘informatie’ wordt gerepresenteerd door de ontwikkeling van deze verschillende oplossingen; goede informatie komt overeen met de tot dan toe beste gevonden oplossingen en de voedselbron komt overeen met de optimale oplossing van het probleem. Het gebruik van PSO heeft de volgende voordelen ([1]):

- Het concept van PSO is relatief eenvoudig en PSO kan ook relatief eenvoudig geïmplementeerd worden voor vele toepassingen.
- PSO is veelzijdig, robuust en voor te gebruiken voor algemene doeleinden.
- PSO kan redelijk goede oplossingen genereren in relatief korte tijd.
- PSO gebruikt slechts een paar parameters, zodat het relatief eenvoudig is om de beste combinatie van parameters te vinden.
- Lokale zoektechnieken om de oplossing te verbeteren kunnen eenvoudig geïntegreerd worden in PSO.
- PSO is geschikt voor parallele berekening, waardoor de totale rekentijd aanzienlijk verkort kan worden.

5.2 Globale werking van de PSO-heuristiek

In deze paragraaf wordt globaal de werking van de PSO-heuristiek toegelicht die gebruikt wordt om de cliëntenplanning te maken. De eerste stap die door de heuristiek wordt gedaan is de initialisatie. Hierbij worden een K $m \times n$ -matrices gevuld met waarden die uniform verdeeld zijn op $[X_{min}, X_{max}]$, waarbij K , X_{min} en X_{max} PSO-parameters zijn. Daarna worden er K andere $m \times n$ -matrices gevuld met waarden die uniform verdeeld zijn op $[V_{min}, V_{max}]$, waarbij V_{min} en V_{max} PSO-parameters zijn. De waarden vormen de initiële ‘snelheid’ van de deeltjes.

Na de initialisatie gaat de daadwerkelijke PSO-heuristiek aan de slag. De eerste stap hiervan is het omzetten van de deeltjesmatrices naar plannings. Dit wordt gedaan door de Heuristic Assignment. De planning die uit de Heuristic Assignment volgt, kan vervolgens eventueel verbeterd worden door de Local Improvement Procedure (LIP). Nadat voor elk deeltje een planning is gemaakt, wordt er gekeken naar de beste oplossingen. Tijdens het proces ‘onthoudt’ elk deeltje zijn eigen beste positie (dat wil zeggen die matrix waarbij de doelfunctiewaarde van de bijbehorende planning minimaal was). Ook wordt voor elke iteratie onthouden welk deeltje tijdens die iteratie op

de beste positie zat (hiermee wordt de beste tot nu toe gevonden oplossing vastgelegd). Daarna wordt de snelheid van de deeltjes aangepast. Deze aangepaste snelheden worden vervolgens gebruikt om de nieuwe posities van de deeltjes te bepalen. Daarna begint de PSO-heuristiek weer met het omzetten van de nieuwe deeltjesmatrices naar planningen. Dit proces wordt herhaald totdat aan bepaalde stopcriteria is voldaan.

```

‘PSO-heuristiek’
  Initialisatie
  while nog niet voldaan aan stopcriteria do
    for all deeltjes  $k$  do
      Doe de Heuristic Assignment
      Voer eventueel de Local Improvement Procedure uit
    end for
    Update de beste oplossingen
    Update de snelheid
    Bereken de nieuwe posities
  end while

```

5.3 Onderdelen van de PSO-heuristiek

In deze paragraaf wordt uitgebreider ingegaan op de verschillende onderdelen van de PSO-heuristiek. Hierbij wordt globaal de volgorde van de heuristiek gevolgd en worden ook pseudocodes gegeven voor de verschillende onderdelen.

5.3.1 Initialisatie

De eerste stap van de PSO-heuristiek is de initialisatie. Hierbij wordt voor elk deeltje k een $m \times n$ -matrix X_{ijk}^0 gevuld met uniform verdeelde waarden uit $[X_{min}, X_{max}]$. De matrix $X_k^t = (X_{ijk}^t)$ wordt de positiematrix van deeltje k tijdens iteratie t genoemd. De m rijen van al deze matrices corresponderen met de m medewerkers en de n kolommen corresponderen met de n afspraken. De waarden in deze matrices geven aan hoe groot de voorkeur van de heuristiek is om een medewerker i te koppelen aan afspraak j : des te kleiner het (i, j) -element is, des te groter deze voorkeur is. Om de hoeveelheid rekenwerk te verkleinen, wordt de waarde van het (i, j) -element op ∞ gezet als vooraf al bekend is dat medewerker i niet in staat is afspraak j uit te voeren. Dit is het geval als medewerker i niet werkt ten tijde van afspraak j (dit is het geval als $\forall \tilde{s} \in \mathcal{S}_i [T_{i\tilde{s}d_j}^{end} = 0]$) of als de bevoegdheid van medewerker i niet groot genoeg is om afspraak j uit te voeren (dit is het geval als $s_i < \ell_j$). Op deze manier is (en blijft) de voorkeur voor dergelijke combinaties minimaal tijdens de uitvoering van de heuristiek.

Aan het eind wordt er voor elk deeltje k ook nog een $m \times n$ -matrix V_{ijk}^0 gevuld met uniform verdeelde waarden uit $[V_{min}, V_{max}]$. De matrix $V_k^t = (V_{ijk}^t)$ wordt de snelheidsmatrix van deeltje k tijdens iteratie t genoemd. Deze matrices worden gebruikt om de nieuwe posities van de deeltjes te bepalen. Op de volgende pagina staat een pseudocode voor het initialisatieproces.

```

‘Initialisatie’
  for all deeltjes  $k$  do
    for  $i = 1$  to  $m$  do
      for  $j = 1$  to  $n$  do
        if  $\exists \tilde{s} \in \mathcal{S}_i [T_{i\tilde{s}d_j}^{end} > 0]$  and  $s_i \geq \ell_j$  then
           $X_{ijk}^0 = X_{min} + (X_{max} - X_{min}) \cdot U[0, 1]$  (met  $U[0, 1]$  een uniform verdeelde stochastische variabele)
           $V_{ijk}^0 = V_{min} + (V_{max} - V_{min}) \cdot U[0, 1]$  (met  $U[0, 1]$  een uniform verdeelde stochastische variabele)
        else
           $X_{ijk}^0 = \infty$ .
           $V_{ijk}^0 = 0$ .
        end if
      end for
    end for
  end for

```

5.3.2 Heuristic Assignment

De Heuristic Assignment wordt gebruikt om de positiematrix van de deeltjes om te zetten naar een planning. Hiervoor moeten eerst de voorkeursmatrix AP_k^t gemaakt worden. Deze matrix wordt gevuld met medewerkersnummers, waarbij AP_{ijk}^t het medewerkersnummer is van de medewerker die de i^e voorkeur heeft voor afspraak j . Nadat deze voorkeursmatrix bepaald is, worden de afspraken stuk voor stuk ingepland, beginnend met de afspraken waarvan het begin van het tijdsvenster het vroegst is. Als eerste wordt een afspraak j toegekend aan de medewerker die de eerste voorkeur heeft voor die afspraak (medewerker AP_{1jk}^t), en wordt het begintijdstip van de afspraak op het ideale begintijdstip gezet; dat wil zeggen, het midden van het tijdsvenster. Vervolgens wordt de feasibility van deze toekenning gecontroleerd. Dit proces wordt verder toegelicht in paragraaf 5.3.2.1. Als de toekenning niet feasible is, dan wordt afspraak j toegekend aan de medewerker die daarna de eerste voorkeur heeft. Dit proces wordt herhaald totdat er een medewerker wordt gevonden waarvoor de toekenning feasible is, of totdat duidelijk is dat een dergelijke medewerker niet gevonden kan worden.

Als er een medewerker is gevonden waarvoor de toekenning feasible is, dan wordt afspraak j ook daadwerkelijk ingevuld in de planning. Alle relevante variabelen worden hierbij geüpdatet. Als deze medewerker niet de medewerker is met de eerste voorkeur voor afspraak j , dan wordt het zogeheten reparatiealgoritme ingezet. Deze zorgt ervoor dat de waarden in de X_k^t -matrix voor afspraak j van de gekozen medewerker en de medewerker met de eerste voorkeur omgewisseld worden. Na het uitvoeren van dit reparatiealgoritme is de medewerker aan wie afspraak j is toegekend dus altijd de medewerker met de eerste voorkeur voor afspraak j . Als er geen medewerker is gevonden waarvoor de toekenning feasible is, dan wordt afspraak j toegevoegd aan de lijst van niet-ingeplande afspraken.

Nadat alle afspraken op deze manier of ingepland zijn of in de lijst van niet-ingeplande afspraken staan, wordt ten slotte nog de zogeheten 'evv-controle' uitgevoerd. Deze controle wordt verder toegelicht in paragraaf 5.3.2.2. Hieronder staat een pseudocode voor de Heuristic Assignment.

```

'Heuristic Assignment'
  u = 0
  unassignlist = ∅
  Maak voorkeursmatrix  $AP_k^t$ 
  for j = 1 to n do
    feasible = false
    priority = 0
    while priority < m and feasible = false and  $X_{AP_{priority+1,j,k}^t}^t < \infty$  do
      priority = priority + 1
      Ken afspraak j toe aan medewerker  $AP_{priority,j,k}^t$ 
      Controleer de feasibility van deze toekenning en doe eventueel kleine aanpassingen aan de planning
      if de toekenning is feasible then
        feasible = true
      end if
    end while
    if feasible = true then
      Vul afspraak j daadwerkelijk in in de planning voor medewerker  $AP_{priority,j,k}^t$ 
      if priority > 1 then
        dummy =  $X_{AP_{1jk}^t}^t$ 
         $X_{AP_{1jk}^t}^t = X_{AP_{priority,j,k}^t}^t$ 
         $X_{AP_{priority,j,k}^t}^t = dummy$ 
      end if
    else
      u = u + 1
      unassignlist(u) = j
    end if
  end for
  Voer de 'evv-controle' uit

```


5.3.2.1 Controleren van feasibility

De feasibility-controle wordt gebruikt om te onderzoeken of het mogelijk is om een afspraak j toe te kennen aan een medewerker i in een al gedeeltelijk gevulde planning. Hiervoor moet gekeken worden of het mogelijk is de planning zodanig te maken dat deze voldoet aan de randvoorwaarden (1)-(7). Randvoorwaarde (1) wordt al gecontroleerd bij de initialisatie en randvoorwaarde (7) zal worden gecontroleerd bij de evv-controle. Dat betekent dat er in deze feasibility-controle gekeken moet worden naar de tijdsvoorwaarden (2)-(5) en de voorwaarde van het maximale aantal medewerkers (6).

De feasibility-controle kijkt eerst naar het maximale aantal medewerkers voor cliënt c_j . Hiervoor worden voor de planning zoals die tot nu toe ingevuld is, de verzamelingen $Z_{c_j}^{huidig}$ en $Z_{c_j}^{nieuw}$ bepaald. Daarbij wordt aangenomen dat afspraak j toegekend wordt aan medewerker i . Als nu blijkt dat $|Z_{c_j}^{nieuw}| > z_{c_j}^{max}$, dan wordt de parameter *feasible* op *false* gezet en stopt de feasibility-controle. Als dit niet het geval is, dan gaat de feasibility-controle verder met de tijdsvoorwaarden (2)-(5).

Als de planning aan alle tijdsvoorwaarden voldoet, dan is de feasibility-controle klaar en wordt de parameter *feasible* op *true* gezet. Als dit niet het geval is, dan wordt er gekeken of het mogelijk is om aan de tijdsvoorwaarden te voldoen door afspraken van medewerker i te verschuiven binnen hun tijdsvensters. Als er een of meerdere plannings worden gevonden die wel voldoen aan alle tijdsvoorwaarden, dan wordt uit deze groep plannings een planning gekozen met de kleinste doelfunctiewaarde. Ook wordt dan de parameter *feasible* op *true* gezet. Als er door het verschuiven van afspraken nog steeds geen planning gevonden kan worden die aan de tijdsvoorwaarden voldoet, dan wordt de parameter *feasible* op *false* gezet en stopt de 'feasibility'-controle. Hieronder staat een pseudocode voor de feasibility-controle.

'Feasibility-controle'

Bepaal $Z_{c_j}^{huidig}$ en $Z_{c_j}^{nieuw}$

if $|Z_{c_j}^{nieuw}| > z_{c_j}^{max}$ **then**
 feasible = *false*

else

if er niet wordt voldaan aan een tijdsvoorwaarde **then**

 verschuif de afspraken van medewerker i binnen hun tijdsvensters

if op deze manier een planning ontstaat die voldoet aan de tijdsvoorwaarden **then**

 kies de planning die de doelfunctie minimaliseert (en voldoet aan de tijdsvoorwaarden)

feasible = *true*

else

feasible = *false*

end if

else

feasible = *true*

end if

end if

5.3.2.2 Evv-controle

Nadat alle afspraken zijn ingepland (of eventueel toegevoegd aan de lijst met niet-ingeplande afspraken), moet er nog één randvoorwaarde gecontroleerd worden, namelijk de evv-voorwaarde (7). Hiervoor wordt aangenomen dat de cliëntnummers gesorteerd zijn, zodat de cliënten met het kleinste aantal afspraken de laagste nummers hebben. Vervolgens wordt per cliënt bekeken of deze gezien wordt door zijn of haar evv'er. Als dit inderdaad het geval is, dan is de evv-controle klaar voor deze cliënt.

Als dit niet het geval is, dan wordt er voor elke afspraak van deze cliënt c gekeken of het mogelijk is deze te verwisselen met een afspraak van de evv'er van deze cliënt (medewerker e_c). Hierbij mogen reeds gecontroleerde randvoorwaarden niet overschreden worden. In deze fase van de heuristiek zijn de voorwaarden (1)-(6) en $e_{\tilde{c}} \in Z_{\tilde{c}}^{huidig}$ voor $\{\tilde{c} \mid \tilde{c} < c, \tilde{c} \notin \text{noevvlist}\}$ al gecontroleerd. Als dergelijke verwisselingen mogelijk zijn, dan wordt de verwisseling gekozen die de kleinste doelfunctiewaarde oplevert. Als dit niet mogelijk is, dan wordt cliënt c toegevoegd aan de lijst van cliënten die niet door hun evv'er worden gezien. Hieronder staat een pseudocode voor de evv-controle.

‘Evv-controle’

$v = 0$

$\text{noevvlist} = \emptyset$

for $c = 1$ **to** C **do**

if $e_c \notin Z_c^{huidig}$ **then**

$\pi = 0$

$\Delta = \infty$

for $j = 1$ **to** n **do**

if $c_j = c$ **then**

 Probeer afspraak j te verwisselen met een afspraak van medewerker e_c (voor cliënt \tilde{c}) zonder de randvoorwaarden (1)-(6) en (als $\tilde{c} < c$ en $\tilde{c} \notin \text{noevvlist}$) $e_{\tilde{c}} \in Z_{\tilde{c}}^{huidig}$ te overtreden

if dit mogelijk is **then**

 Bereken de doelfunctiewaarde Δ_j van de planning na de verwisseling met afspraak j

if $\Delta_j < \Delta$ **then**

$\Delta = \Delta_j$

$\pi = j$

end if

end if

end for

end for

if $\pi > 0$ **then**

 Voer de verwisseling met afspraak j daadwerkelijk uit

else

$v = v + 1$

$\text{noevvlist}(v) = c$

end if

end if

end for

5.3.3 Local Improvement Procedure (LIP)

Na de Heuristic Assignment kan eventueel de Local Improvement Procedure (LIP) uitgevoerd worden. Deze LIP probeert voor elke medewerker en elke dienst de volgorde van de afspraken die door de medewerker in die dienst worden uitgevoerd te optimaliseren. Hiervoor wordt eerst het aantal afspraken w in deze dienst bepaald. Dan worden de afspraken in de dienst zoveel mogelijk naar elkaar toegeschoven, zonder de tijdsvoorwaarden (2)-(5) te overschrijden, waarna de huidige kosten Δ^* van deze dienst berekend worden.

Vervolgens verwisselt het algoritme steeds twee opeenvolgende afspraken, opnieuw zonder de tijdsvoorwaarden (2)-(5) te overschrijden. Hiervoor wordt elke keer eerst onderzocht welke verwisseling de laagste kosten oplevert. Zodra er geen verbetering meer mogelijk is, wordt uiteindelijk de volgorde gekozen waarbij de kosten minimaal zijn. Hieronder staat een pseudocode voor de LIP.

```
‘Local Improvement Procedure (LIP)’
for  $i = 1$  to  $m$  do
  for  $d = 1$  to  $D$  do
    for  $s = 1$  to  $J$  do
      Bepaal het aantal afspraken  $w$  in dienst  $s$  op dag  $d$  van medewerker  $i$ 
      Verwijder overbodige gaten in deze dienst door de afspraken zo veel mogelijk naar elkaar toe te schuiven
      Bereken de huidige kosten  $\Delta^*$  voor deze dienst
      while nog niet voldaan aan stopcriteria do
         $\Delta = \infty$ 
         $\pi = 0$ 
        for  $j = 1$  to  $w - 1$  do
          Probeer de  $j^e$  en  $(j + 1)^e$  afspraak in deze dienst te verwisselen
          if dit is niet mogelijk then
             $\Delta_j = \infty$ 
          else
            Verwijder overbodige gaten in deze dienst door de afspraken zo veel mogelijk naar elkaar toe te schuiven
            Bereken de kosten  $\Delta_j$ 
          end if
          if  $\Delta_j < \Delta$  then
             $\Delta = \Delta_j$ 
             $\pi = j$ 
          end if
        end for
        Verwissel de  $\pi^e$  en  $(\pi + 1)^e$  afspraak in deze dienst
        if  $\Delta < \Delta^*$  then
           $\Delta^* = \Delta$ 
        end if
      end while
      Kies voor deze dienst de planning met kosten  $\Delta^*$ 
    end for
  end for
end for
```

5.3.4 Beste oplossingen

Nadat de deeltjesmatrices X_k^t omgezet zijn in planningsen en de bijbehorende doelfunctiewaarden $f(X_k^t)$ bekend zijn, wordt er gekeken welke oplossingen het beste zijn. Voor elk deeltje k wordt de matrix $pbest_k^t$ gebruikt om de beste positie van dat deeltje tot en met iteratie t te representeren. Dat wil zeggen, $pbest_k^t$ is de positiematrix waarvan de doelfunctiewaarde van de bijbehorende planning minimaal is.

Verder wordt $gbest^t$ gebruikt om de index van het beste deeltje tot en met iteratie t te representeren. Op deze manier is $pbest_{gbest^t}^t$ dus de positiematrix die hoort bij de beste planning die gevonden is tot en met iteratie t . Tijdens iteratie 0 wordt $pbest_k^0 := X_k^0$ en $gbest^0 := \operatorname{argmin}_k f(pbest_k^0)$ gedefinieerd. Tijdens alle volgende iteraties wordt de pseudocode op de volgende pagina gebruikt om de nieuwe beste oplossingen te bepalen.

```

‘Beste oplossingen’
for all deeltjes  $k$  do
  if  $f(X_k^t) < f(pbest_k^{t-1})$  then
     $pbest_k^t = X_k^t$ 
  else
     $pbest_k^t = pbest_k^{t-1}$ 
  end if
end for
 $gbest^t = gbest^{t-1}$ 
for all deeltjes  $k$  do
  if  $f(pbest_k^t) < f(pbest_{gbest^t}^t)$  then
     $gbest^t = k$ 
  end if
end for

```

5.3.5 Snelheid aanpassen

Nu de posities van de beste oplossingen die tot nu gevonden zijn, bekend zijn, kan er gekeken worden hoe de deeltjes zich gaan ‘verplaatsen’ naar hun nieuwe posities. Hiervoor moeten de snelheidsmatrices van de deeltjes aangepast worden. Dit wordt gedaan aan de hand van de volgende vergelijking, afkomstig uit [15]:

$$V_{ijk}^{t+1} = \chi (w^t \cdot V_{ijk}^t + c_1 \cdot r_1 \cdot (pbest_{ijk}^t - X_{ijk}^t) + c_2 \cdot r_2 \cdot (pbest_{ij,gbest^t}^t - X_{ijk}^t)). \quad (48)$$

Deze snelheidsfunctie bestaat uit drie belangrijke delen: de impuls (of momentum), een cognitief gedeelte (dat rekening houdt met de ‘persoonlijke’ ervaringen van het deeltje) en een sociaal gedeelte (dat rekening houdt met de ervaringen van de groep deeltjes als geheel). Voor de impuls wordt de ‘traagheidsparameter’ w^t gebruikt om de grootte van het effect van de vorige snelheid op de huidige snelheid te bepalen. Het cognitieve gedeelte ‘stuurt’ de oplossing in de richting van $pbest_k^t$. Hiervoor wordt de acceleratieconstante c_1 en de stochastische variabele r_1 (met $U[0, 1]$ -verdeling) gebruikt. Het sociale gedeelte ‘stuurt’ de oplossing in de richting van $pbest_{gbest^t}^t$. Hiervoor wordt de acceleratieconstante c_2 en de stochastische variabele r_2 (met $U[0, 1]$ -verdeling) gebruikt. Ten slotte wordt de constrictiefactor χ gebruikt om te voorkomen dat de snelheid te groot wordt. Een verdere toelichting over de PSO-parameters χ , w^t , c_1 en c_2 is te vinden in de volgende paragraaf.

Na het berekenen van de nieuwe snelheden, kunnen de positiematrixes van de deeltjes geüpdatet worden. Dit wordt gedaan met de volgende vergelijking:

$$X_{ijk}^{t+1} = X_{ijk}^t + V_{ijk}^{t+1}. \quad (49)$$

5.4 Parameters van de PSO-heuristiek

In deze paragraaf worden de parameters van de PSO-heuristiek besproken en onderzocht. Deze parameters beïnvloeden de prestatie van de PSO-heuristiek. Het is daarom belangrijk de juiste waarden te kiezen voor deze parameters. In deze paragraaf worden de parameters t (aantal iteraties), K (aantal deeltjes), $[X_{min}, X_{max}]$ en $[V_{min}, V_{max}]$ (respectievelijk de minimale en maximale positiewaarde en snelheid van elk deeltje), χ (constrictiefactor), w (‘traagheidsparameter’) en c_1, c_2 (acceleratieconstanten) besproken. Voor elk van deze parameters wordt besproken welke invloed ze hebben op de oplossing. Vervolgens wordt er voor een deel van deze parameters de juiste waarde bepaald met behulp van de Taguchi-methode.

5.4.1 PSO-parameters

De parameters t en K dragen bij aan de rekentijd en de prestatie van het algoritme. Hoe groter de waarden van t en K zijn, des te langer moet er gerekend worden, maar des te beter kan de oplossing worden. Als t en K te klein gekozen worden, is de kans groter dat de oplossing niet goed genoeg is, maar als t en K te groot gekozen worden, dan wordt de rekentijd te groot.

Het bereik van de positiewaarde van elke deeltje wordt gegeven door $[X_{min}, X_{max}]$ en het bereik van de snelheid wordt gegeven door $[V_{min}, V_{max}]$. Hoe groter dit laatste bereik is, des te sneller convergeert de oplossing. Als het

bereik van de snelheden te groot of te klein is, dan is de kans groot dat de oplossing naar lokale optima convergeert of helemaal niet convergeert.

In de snelheidsfunctie (48) zijn χ , w , c_1 en c_2 de te kiezen parameters. Het effect van de vorige snelheid op de huidige snelheid wordt bepaald door w . Een grote w zorgt voor een betere globale verkenning van de oplossingsruimte, terwijl een kleine w zorgt voor een betere lokale exploitatie. Een goede waarde voor w vindt dus een goede balans tussen het vermogen van de PSO om te verkennen en het vermogen om te exploiteren. Het is ook mogelijk de waarde van w te variëren tijdens de verschillende iteraties van de PSO.

De acceleratieconstanten c_1 en c_2 representeren de maximale grootte van de versnellingen in de richting van respectievelijk *pbest* (beste positie van het deeltje zelf) en *gbest* (beste positie van alle deeltjes samen). Een hoge waarde voor respectievelijk c_1 of c_2 zorgt ervoor dat een deeltje snel convergeert naar respectievelijk *pbest* of *gbest*, terwijl een lage waarde deze convergentie juist tegenhoudt.

5.4.2 Taguchi-methode

De PSO-parameters bepalen voor een belangrijk deel het vermogen van een PSO-heuristiek om een optimale oplossing te vinden. Verschillende combinaties van parameters werken goed voor verschillende problemen. Voor de PSO-heuristiek die hiervoor is beschreven, zijn de prestaties van verschillende combinaties van waarden van PSO-parameters getest volgens het ‘Taguchi design of experiments’. De benadering van Taguchi maakt gebruik van een serie experimenten om de parameters te vinden die de beste resultaten geven en die zo min mogelijk gevoelig zijn voor storingen. Een storing is een ongecontroleerde variatie die van invloed kan zijn op de prestaties van de heuristiek. De toepassing van de Taguchi-methode is beschreven in [8] en [17].

De Taguchi-methode maakt voor elke parameter gebruik van een aantal verschillende ‘levels’. Het level van een parameter geeft aan welke waarde voor deze parameter in een bepaald experiment wordt gekozen. In dit Taguchi-experiment wordt er gebruik gemaakt van één parameter met twee levels en zes parameters met vijf levels. Het ontwerp van dit Taguchi-experiment is gebaseerd op een L50-orthogonaalmatrix (zie Appendix A.3).

In [27] wordt de waarde van w initieel op 0.9 gezet en neemt deze geleidelijk af tot een waarde van 0.4 met een afnamefactor van 0.975. Verder worden de acceleratieconstanten gelijk gekozen aan 2 en geldt $[X_{min}, X_{max}] = [-n, n]$ met n een nader te bepalen constante. Daarnaast geven [2, 10, 11, 32] de volgende formule voor de waarde van χ :

$$\chi = \begin{cases} \frac{2\kappa}{\phi - 2 + \sqrt{\phi^2 - 4\phi}}, & \text{als } \phi > 4 \text{ met } \phi = c_1 + c_2 \\ \kappa, & \text{als } \phi \leq 4 \end{cases} \quad (50)$$

In het algemeen wordt gebruikt gemaakt van $\kappa = 1$ ([2, 11]). In [29] wordt aangeraden w constant te houden met een waarde van 0.8 en de acceleratieconstante c_1 op 1.5 te zetten. Verder wordt aangeraden gebruik te maken van (50) met $\phi = 4.1$, $V_{max} = X_{max}$ en $V_{min} = -X_{max}$ met $[X_{min}, X_{max}] = [0, n]$ met n een te nader te bepalen constante. Andere suggesties voor de waarden van de parameters zijn te vinden in [27, 28, 32].

Aan de hand de literatuur zijn verschillende combinaties van parameters gevonden en deze zijn verwerkt in tabel 1. In de tabel zijn de verschillende waarden van de parameters verdeeld over verschillende levels. Hierin zijn w_{start} , w_{end} en w_{vary} respectievelijk de startwaarde, de eindwaarde en de afnamefactor van w . Verder wordt gebruikt dat $V_{max} = \lambda X_{max}$ en $V_{min} = -V_{max}$. Aan de hand van de L50-orthogonaalmatrix worden de 50 verschillende

Level	w_{start}	w_{end}	w_{vary}	λ	ϕ	c_1	$[X_{min}, X_{max}]$
1	0.9	0.5	1	1	4	2.5	[0,10]
2	0.85	0.4	0.975	0.9	4.05	2.2	[-10,10]
3	0.8	0.3	0.925	0.7	4.1	2	
4	0.75	0.2	0.9	0.5	4.15	1.8	
5	0.7	0.1	0.85	0.3	4.20	1.5	

Tabel 1: De waarden die horen bij de verschillende levels van de parameters.

experimenten uitgevoerd. Elk experiment wordt 25 keer herhaald. Aan de hand van deze 25 runs wordt de zogeheten ‘signal-of-noise’-ratio (S/N-ratio) bepaald. De S/N-ratio meet hoe de uitkomsten variëren ten opzichte van de gemiddelde waarde onder verschillende omstandigheden.

De S/N-ratio wordt als volgt berekend:

$$S/N_i = 10 \log \left(\frac{\bar{y}_i^2}{s_i^2} \right)$$

waarbij

$$\bar{y}_i = \frac{1}{N_i} \sum_{u=1}^{N_i} y_{i,u} \quad \text{en} \quad s_i^2 = \frac{1}{N_i - 1} \sum_{u=1}^{N_i} (y_{i,u} - \bar{y}_i)^2.$$

Hierin is i het nummer van het experiment, $y_{i,u}$ de uitkomst van de u^{de} run van experiment i en is N_i het totaal aantal runs van experiment i . Wanneer de S/N-ratio's van alle 50 experimenten zijn bepaald, kunnen de S/N-ratio's voor de levels van de parameters bepaald worden. Alle S/N-ratio's van de experimenten waarbij een bepaald level van een parameter gebruikt is, worden bij elkaar opgeteld en gedeeld door het totaal aantal experimenten waarin het level van de parameter is voorgekomen. De formule van de S/N-ratio van bijvoorbeeld het eerste level van parameter w_{start} ziet er dan als volgt uit:

$$S/N_{w_{start},1} = \frac{S/N_1 + S/N_2 + S/N_3 + S/N_4 + S/N_{27} + S/N_{28} + S/N_{29} + S/N_{30}}{8}.$$

Wanneer alle S/N-ratio's van de levels van de parameters bekend zijn, kan de beste waarde van de parameter worden bepaald door de waarde van het level met de laagste S/N-ratio te kiezen.

6 Numerieke experimenten

In dit hoofdstuk worden de resultaten beschreven van experimenten die doorgerekend zijn met de PSO-heuristiek die beschreven is in hoofdstuk 5. Deze heuristiek is geïmplementeerd in MATLAB. Om het planningsmodel door te rekenen met deze implementatie is gebruik gemaakt van data uit de praktijk, die is aangeleverd door CARINTREGGELAND. Het gaat hierbij om de cliëntenplanning voor één wijkteam. Binnen dit wijkteam zijn 7 medewerkers actief, die verspreid over 225 afspraken zorg verlenen aan 22 cliënten. Om de variant van het model met reistijden te kunnen doorrekenen, is er een vierkante reistijdentabel nodig met dimensie 23. Deze is berekend met behulp van de Google Distance Matrix API.

Er zijn vier verschillende experimenten uitgevoerd. Het eerste experiment is gebruikt om de optimale PSO-parameters te vinden. In het tweede experiment zijn de resultaten van de PSO-heuristiek voor een kleine dataset vergeleken met de resultaten van een exact algoritme. Het derde experiment is gebruikt om de complexiteit van de PSO-heuristiek te bepalen. In het vierde en laatste experiment wordt er gekeken naar de resultaten van de PSO-heuristiek in een praktijksituatie.

6.1 Experiment 1: PSO-parameters

In dit experiment is de Taguchi-methode toegepast, zoals die is beschreven in paragraaf 5.4.2. Hierbij is gebruik gemaakt van $K = 50$ en $t = 25$. In tabel 2 staan de S/N-ratio's van alle levels van de alle geteste parameters. Ook staan de 'delta', de 'rank' en de gekozen waarden voor de parameters in deze tabel vermeld. De delta geeft het verschil aan tussen de maximale en minimale S/N-ratio van een parameter. De rank geeft aan de hand van de delta een gradatie van de invloed van de parameters op de prestaties van het algoritme. De parameter met de laagste rank heeft de kleinste invloed op de prestaties van het algoritme. Voor elke parameter wordt het level gekozen dat hoort bij de laagste S/N-ratio voor die parameter. In figuur 1 zijn de S/N-ratio's in grafiekvorm te vinden.

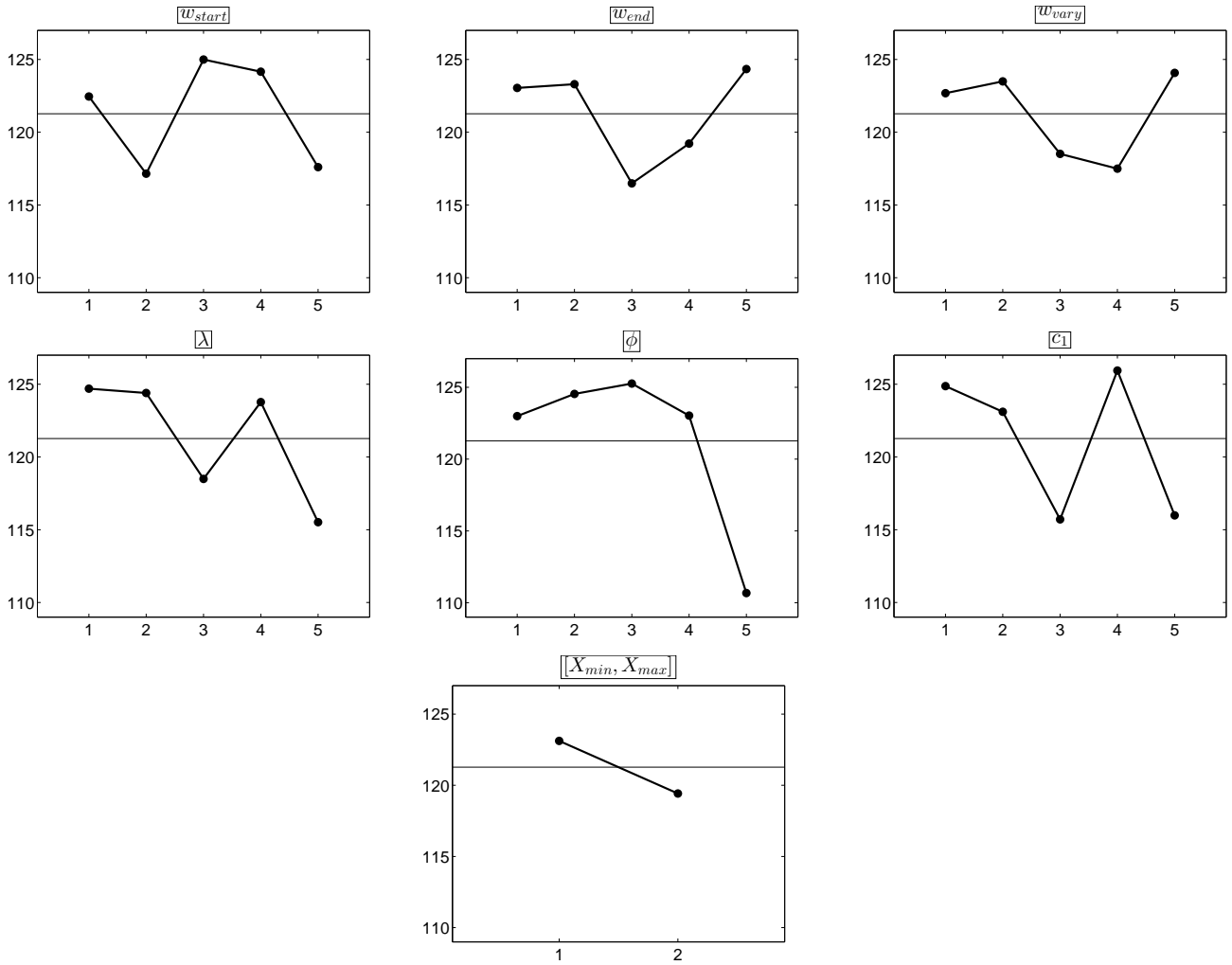
Level	w_{start}	w_{end}	w_{vary}	λ	ϕ	c_1	$[X_{min}, X_{max}]$
1	122.4538	123.0410	122.6807	124.6977	122.9981	124.8702	123.1189
2	117.1501	123.2976	123.4937	124.3965	124.5412	123.1059	119.4232
3	124.9957	116.4839	118.5082	118.4940	125.2621	115.7108	
4	124.1568	119.2179	117.4955	123.7713	123.0334	125.9418	
5	117.5989	124.3405	124.0759	115.5235	110.6749	115.9871	
Delta	7.8456	7.8566	6.5804	9.1742	14.5874	10.2310	3.6957
Rank	3	4	2	5	7	6	1
Gekozen waarde	0.85	0.3	0.9	0.3	4.2	2	[-10 10]

Tabel 2: Resultaten van de Taguchi-methode.

6.2 Experiment 2: PSO-heuristiek versus exact algoritme

Om te bepalen hoe goed de PSO-heuristiek werkt, is er een kleine dataset gecreëerd met daarin 3 medewerkers, 4 cliënten en 47 afspraken. Deze dataset is een deelverzameling van de dataset die is aangeleverd door CARINTREGGELAND. Voor deze kleine dataset is met behulp van CPLEX 12.6 een optimale oplossing gezocht. Deze uitkomst is vergeleken met de oplossing van de PSO-heuristiek. De resultaten hiervan staan in tabel 3.

Er zijn verschillende configuraties van de PSO-heuristiek gebruikt voor dit experiment, waarbij alleen het gebruik van de Local Improvement Procedure (LIP) is gevarieerd. Bij PSO1 is er helemaal geen gebruik gemaakt van de LIP. Bij PSO2 is er voor elk deeltje in elke iteratie gebruik gemaakt van de LIP. Bij PSO3 is alleen gebruik gemaakt van de LIP voor de deeltjes waarvan de totale penalty (als gevolg van niet-ingeplande afspraken en cliënten die hun evv'er niet zien) niet groter is dan de totale penalty van de tot dan toe beste oplossing. Bij PSO4 is de LIP alleen helemaal op het eind gebruikt om het allerbeste deeltje nog verder te verbeteren. Alle configuraties zijn 30 keer gerund. De beste en de gemiddelde resultaten van deze runs staan in tabel 3.



Figuur 1: Grafische weergave van de S/N-ratio's resulterend uit de Taguchi-methode.

In deze tabel is te zien dat PSO2, PSO3 en PSO4 voor deze kleine dataset alledrie een oplossing weten te vinden met een doelfunctiewaarde van 765. Dit komt dicht in de buurt van het daadwerkelijke optimum van 755 dat berekend is met behulp van CPLEX. Verder is in de tabel duidelijk te zien dat het gebruik van de LIP relatief veel tijd kost. Ook is te zien dat de LIP wel nodig is om tot goede resultaten te komen. De resultaten van de configuratie PSO1 zijn namelijk beduidend slechter dan de resultaten van de andere drie configuraties.

	CPLEX	PSO1	PSO2	PSO3	PSO4
Beste doelfunctiewaarde	755	2005	765	765	765
Rekentijd (sec)	-	6.4	97.1	84.1	7.4
Gemiddelde doelfunctiewaarde	-	2284	853	881	912
Gemiddelde rekestijd (sec)	-	7.2	94.0	84.2	7.8

Tabel 3: Resultaten van de verschillende configuraties van de PSO-heuristiek voor een kleine dataset. Elke configuratie is in totaal 30 keer gerund.

6.3 Experiment 3: Complexiteit van de PSO-heuristiek

De PSO-heuristiek die gebruikt is, heeft meer rekentijd nodig naarmate de gebruikte datasets groter worden. De complexiteit van de heuristiek kan op twee manieren bepaald worden en is vooral afhankelijk van het aantal afspraken n dat gepland moet worden, het aantal deeltjes K en het aantal iteraties t . In het eerste geval worden de algoritmes van de heuristiek geanalyseerd. In het tweede geval wordt de rekentijd hiervan bepaald.

De heuristiek maakt gebruik van een ‘grote’ while-loop. Deze while-loop itereert over alle iteraties en moet dus t keer worden uitgevoerd. Binnen deze while-loop wordt een viertal for-loops uitgevoerd. De eerste for-loop wordt gebruikt voor de Heuristic Assignment, de tweede en de derde voor het updaten van de beste oplossingen en de vierde voor het updaten van de snelheid en het berekenen van de nieuwe posities. Al deze for-loops itereren over alle deeltjes en moeten dus K keer uitgevoerd worden. Binnen de Heuristic Assignment wordt ook nog een for-loop gebruikt die itereert over alle afspraken n . Dit betekent dat de complexiteit gegeven wordt door $\mathcal{O}(Ktn)$.

Om dit te testen, wordt de rekentijd uitgezet tegen het aantal deeltjes, het aantal iteraties en het aantal afspraken in de dataset. Als er daadwerkelijk sprake is van een lineaire complexiteit in K , t en n , dan zou het vergroten van het aantal deeltjes, iteraties of afspraken met een factor a moeten leiden tot een vergroting van de rekentijd met dezelfde factor a . Hiervoor is de data gebruikt die al beschikbaar is. Bij 64 runs is de rekentijd bijgehouden. In tabel 4 staat een overzicht van de reketijden bij verschillende aantallen deeltjes, iteraties en afspraken. Uit deze tabel blijkt dat de complexiteit inderdaad ongeveer lineair is in al deze factoren. Dit is ook goed te zien in figuur 2, waarin een deel van deze reketijden grafisch is weergegeven.

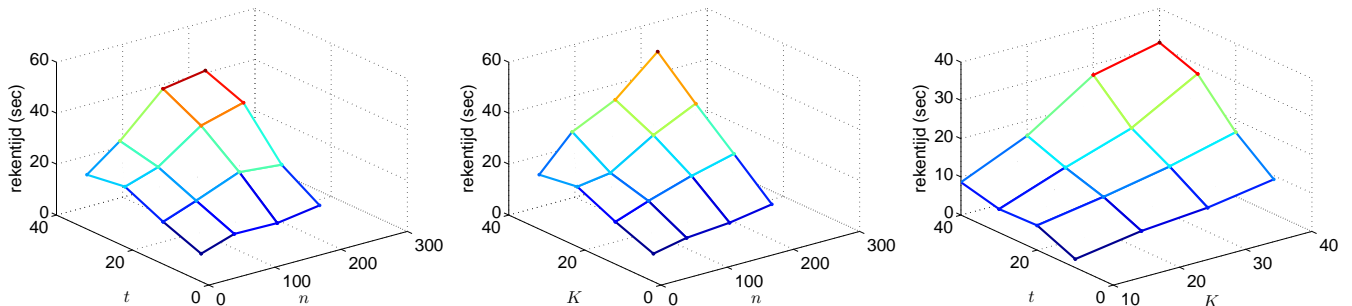
$n \backslash t$	10	20	30	40
46	2.2	6.5	6.2	8.6
96	7.9	12.6	19.3	17.7
161	14.8	19.0	30.7	35.2
225	12.6	22.3	38.3	40.9

$n \backslash t$	10	20	30	40
46	4.9	9.3	12.4	16.1
96	12.5	22.6	28.0	42.1
161	16.7	46.2	54.4	55.1
225	25.9	46.4	38.1	48.2

$n \backslash t$	10	20	30	40
46	6.3	12.5	18.0	27.3
96	17.8	40.2	42.8	53.5
161	26.9	51.4	48.2	135.3
225	34.0	40.3	116.8	121.9

$n \backslash t$	10	20	30	40
46	9.0	16.8	27.5	31.1
96	22.0	38.7	62.9	58.2
161	34.8	64.7	79.3	85.4
225	48.4	73.0	70.6	168.3

Tabel 4: Enkele reketijden van gedraaide runs van PSO4: linksboven voor $K = 10$, rechtsboven voor $K = 20$, linksonder voor $K = 30$ en rechtsonder voor $K = 40$.



Figuur 2: Grafische weergave van enkele reketijden van de gedraaide runs van PSO4: links voor $K = 10$, in het midden voor $t = 10$ en rechts voor $n = 46$.

6.4 Experiment 4: De praktijk

Er is ook getest hoe de PSO-heuristiek werkt in een praktijksituatie. Hiervoor is de volledige dataset van CARINTREGGELAND gebruikt. Er zijn twee situaties getest. In de eerste situatie zijn alle reistijden tussen de verschillende locaties gelijk gesteld aan nul. Dit is ook gedaan in de planning die door CARINTREGGELAND met de hand is gemaakt. In de tweede situatie zijn de echte waarden gebruikt voor alle reistijden tussen de verschillende locaties. Voor de gewichten in de doelfunctie (8) tijdens dit experiment zijn de waarden $\alpha = 5$ en $\beta = 1$ gekozen.

In de eerste situatie leverde de PSO-heuristiek een feasible planning, waarin aan alle gegeven randvoorwaarden is voldaan. Dit in tegenstelling tot de planning die door CARINTREGGELAND is gemaakt, waarin voor een aantal afspraken niet voldaan wordt aan het gegeven tijdsvenster en waarin er ook een cliënt is die niet door haar evv'er wordt gezien. De doelfunctiewaarde die hoort bij deze planning van CARINTREGGELAND is 15255 (inclusief een penalty van 10000 voor het feit dat er een cliënt niet door haar evv'er wordt gezien). De doelfunctiewaarde die hoort bij het beste resultaat van de PSO-heuristiek is 2180.

In de tweede situatie is gebleken dat het niet mogelijk was alle afspraken in te plannen binnen de gegeven randvoorwaarden. Vier afspraken konden niet ingepland worden. De doelfunctiewaarde die hoort bij het beste resultaat van de PSO-heuristiek is 404469 (inclusief vier penalties van 100000 voor het niet inplannen van een afspraak).

7 Conclusies en aanbevelingen

Naar aanleiding van ons onderzoek worden een aantal conclusies en aanbevelingen geformuleerd.

7.1 Conclusies en discussie

Tijdens ons onderzoek hebben we een heuristiek ontwikkeld die de koppeling tussen thuiszorgmedewerkers en cliënten kan optimaliseren. Deze heuristiek is gebaseerd op het principe van Particle Swarm Optimization (PSO). Deze heuristiek hebben we geïmplementeerd. Vervolgens hebben we een realistische dataset van CARINTREGGELAND gebruikt om de prestaties van deze heuristiek te onderzoeken. Voor een kleine dataset is gebleken dat de heuristiek dicht in de buurt komt van het daadwerkelijke optimum. Voor de volledige dataset is gebleken dat de heuristiek in staat is om in relatief korte tijd goede oplossingen te genereren. Deze oplossingen bleken beter te zijn dan de planning van CARINTREGGELAND zelf. In vergelijking met de huidige situatie, waarin een planner circa vier uur nodig heeft om een cliëntenplanning te maken, is deze heuristiek een enorme vooruitgang qua rekentijd, aangezien de heuristiek minder dan 10 minuten rekentijd nodig heeft.

Specifieke gevallen, zoals afspraken die gesplitst moeten worden of afspraken waarvoor twee medewerkers nodig zijn, kunnen (nog) niet worden meegenomen in de heuristiek. Verder is het in sommige gevallen niet mogelijk voor de heuristiek om alle afspraken in te plannen. In dit geval moeten de afspraken die niet ingepland kunnen worden, alsnog met de hand ingepland worden. Hierbij zullen dan een of meerdere randvoorwaarden overschreden moeten worden.

Zonder een goede personeelsplanning is het niet mogelijk om een goede cliëntenplanning te maken. Daarom hebben we voor de personeelsplanning een algemeen model opgesteld. Helaas was het voor ons niet haalbaar om het model daadwerkelijk op te lossen.

7.2 Aanbevelingen voor verder onderzoek

In ons onderzoek zien we nog een aantal verbeterpunten. In deze paragraaf geven we een opsomming waarin staat aangegeven wat er nog ontbreekt en waar nog verbeteringen plaats kunnen vinden.

- Er zijn een aantal kleine factoren uit de praktijk, waar ons model geen rekening mee houdt. Als voorbeeld noemen we de afspraken voor echtparen, die bij voorkeur direct na elkaar gepland zouden moeten worden. Dit soort kleine factoren zijn niet meegenomen in het model.
- Het is mogelijk dat onze heuristiek een paar afspraken niet in kan plannen binnen de gegeven randvoorwaarden. Om ervoor te zorgen dat uiteindelijk alle afspraken wel ingepland worden, is het mogelijk om de tijdsvensters van enkele afspraken te verbreden. Hier zal verder onderzoek naar gedaan moeten worden.
- In de cliëntenplanning vinden wekelijks een aantal verstoringen plaats. Er zijn mogelijkheden om die verstoringen bij te plannen met behulp van een aangepaste versie van onze heuristiek, maar dit moet nog verder onderzocht worden.
- Het programma dat nu geschreven is, is nog niet gebruiksvriendelijk. Er zal dus nog een gebruiksvriendelijke interface geconstrueerd moeten worden.
- Het gebruik van goede software maakt het werken met dit model waarschijnlijk een stuk eenvoudiger. In ons onderzoek hebben we gewerkt met MATLAB. Hiermee is het programmeren relatief eenvoudig, maar dat gaat ten koste van de snelheid. Met de juiste software kan de heuristiek sneller werken.
- Het model om de personeelsplanning mee te maken moet nog verder uitgewerkt worden. Verder moet er nog uitgezocht worden hoe het verwachte aantal benodigde medewerkers en verwachte aantal diensturen per dienst bepaald moeten worden. Hiervoor zal een stochastisch model opgesteld moeten worden.
- In de praktijk is de personeelsplanning directe input voor de cliëntenplanning en zal de cliëntenplanning indirect invloed moeten hebben op de personeelsplanning. Hoe deze koppeling precies in elkaar zit, moet nog onderzocht worden.

Bibliografie

- [1] Chananés Akjiratikarl, Pisal Yenradee, and Paul R. Drake. PSO-based algorithm for home care worker scheduling in the UK. *Computers & Industrial Engineering*, 53(4):559–583, November 2007. doi: 10.1016/j.cie.2007.06.002.
- [2] Qinghai Bai. Analysis of particle swarm optimization algorithm. *Computer and Information Science*, 3(1): 180–184, February 2010.
- [3] Rym Ben Bachouch, Alain Guinet, and Sonia Hajri-Gabouj. An optimization model for task assignment in home health care. In *IEEE Workshop on Health Care Management (WHCM)*, pages 1–6, Venice, 2010. doi: 10.1109/WHCM.2010.5441277.
- [4] Stefan Bertels and Torsten Fahle. A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. *Computers & Operations Research*, 33(10):2866–2890, October 2006. doi: 10.1016/j.cor.2005.01.015.
- [5] Valeria Borsani, Andrea Matta, Giacomo Beschi, and Francesco Sommaruga. A home care scheduling model for human resources. In *International Conference on Service Systems and Service Management (ICSSSM)*, volume 1, pages 449–454, Troyes, 2006. IEEE. doi: 10.1109/ICSSSM.2006.320504.
- [6] Olli Bräysy, Juha Arola, Wout Dullaert, and Jarno Väisänen. Planning strategies for home care services. 2012.
- [7] David Bredström and Mikael Rönnqvist. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research*, 191(1):19–31, November 2008. doi: 10.1016/j.ejor.2007.07.033.
- [8] Bor-Wen Cheng and Chun-Lang Chang. A study on flowshop scheduling problem combining Taguchi experimental design and genetic algorithm. *Expert Systems with Applications*, 32(2):415–421, February 2007. doi: 10.1016/j.eswa.2005.12.002.
- [9] Eddie Cheng and Jennifer Lynn Rich. A home health care routing and scheduling problem. *Rice University, Texas, Tech. Rep. TR98-04*, 1998.
- [10] Maurice Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC)*, volume 3, pages 1951–1957, Washington, DC, July 1999. IEEE. doi: 10.1109/CEC.1999.785513.
- [11] Russell C. Eberhart and Yuhui Shi. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC)*, volume 1, pages 84–88, La Jolla, CA, July 2000. IEEE. doi: 10.1109/CEC.2000.870279.
- [12] Patrik Eveborn, Patrik Flisberg, and Mikael Rönnqvist. LAPS CARE — an operational system for staff planning of home care. *European Journal of Operational Research*, 171(3):962–976, June 2006. doi: 10.1016/j.ejor.2005.01.011.
- [13] Matias Holte and Carlo Mannino. The implementor/adversary algorithm for the cyclic and robust scheduling problem in health-care. *European Journal of Operational Research*, 226(3):551–559, May 2013. doi: 10.1016/j.ejor.2012.10.029.
- [14] Jaber Jemai, Marouen Chaieb, and Khaled Mellouli. The home care scheduling problem: A modeling and solving issue. In *5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, pages 1–6, Hammamet, 2013. IEEE. doi: 10.1109/ICMSAO.2013.6552619.

- [15] James Kennedy and Russell Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, WA, November/December 1995. IEEE. doi: 10.1109/ICNN.1995.488968.
- [16] Yannick Kergosien, Christophe Lenté, and Jean-Charles Billaut. Home health care problem: An extended multiple traveling salesman problem. In *4th Multidisciplinary International Scheduling Conference: Theory and Applications (MISTA)*, pages 85–92, Dublin, 2009.
- [17] Arun Khosla, Shakti Kumar, and Kamal K. Aggarwal. Identification of strategy parameters for particle swarm optimizer through Taguchi method. *Journal of Zhejiang University SCIENCE A*, 7(12):1989–1994, December 2006. doi: 10.1631/jzus.2006.A1989.
- [18] Paulien M. Koeleman, Sandjai Bhulai, and Maarten van Meersbergen. Optimal patient and personnel scheduling policies for care-at-home service facilities. *European Journal of Operational Research*, 219(3):557–563, June 2012. doi: 10.1016/j.ejor.2011.10.046.
- [19] Nikky Kortbeek, Aleida Braaksma, Christian A.J. Burger, Piet J.M. Bakker, and Richard J. Boucherie. Flexible nurse staffing based on hourly bed census predictions. Memorandum 1996, Department of Applied Mathematics, University of Twente, Enschede, October 2012.
- [20] Nikky Kortbeek, Aleida Braaksma, H.F. Smeenk, Piet J.M. Bakker, and Richard J. Boucherie. Integral resource capacity planning for inpatient care services based on hourly bed census predictions. Memorandum 1990, Department of Applied Mathematics, University of Twente, Enschede, September 2012.
- [21] Ettore Lanzarone and Andrea Matta. Robust nurse-to-patient assignment in home care services to minimize overtimes under continuity of care. *Operations Research for Health Care*, 2014. doi: 10.1016/j.orhc.2014.01.003.
- [22] Dorota Slawa Mankowska, Frank Meisel, and Christian Bierwirth. The home health care routing and scheduling problem with interdependent services. *Health Care Management Science*, June 2013. doi: 10.1007/s10729-013-9243-1.
- [23] Menzis Zorgkantoor. Sectoraal inkoopplan AWBZ 2014 V&V (blz. 37), June 2013.
- [24] Stefan Nickel, Michael Schröder, and Jörg Steeg. Mid-term and short-term planning support for home health care services. *European Journal of Operational Research*, 219(3):574–587, June 2012. doi: 10.1016/j.ejor.2011.10.042.
- [25] Matias Sevel Rasmussen, Tor Justesen, Anders Dohn, and Jesper Larsen. The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3):598–610, June 2012. doi: 10.1016/j.ejor.2011.10.048.
- [26] Parisa Shahnazari-Shahrezaei, Reza Tavakkoli-Moghaddam, and Hamed Kazemipoor. Solving a new fuzzy multi-objective model for a multi-skilled manpower scheduling problem by particle swarm optimization and elite tabu search. *The International Journal of Advanced Manufacturing Technology*, 64(9-12):1517–1540, February 2013. doi: 10.1007/s00170-012-4119-y.
- [27] Yuhui Shi and Russell C. Eberhart. Parameter selection in particle swarm optimization. In V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben, editors, *Evolutionary Programming VII: 7th International Conference, EP98 San Diego, California, USA, March 25-27, 1998 Proceedings*, volume 1447 of *Lecture Notes in Computer Science*, pages 591–600. Springer Berlin Heidelberg, 1998. doi: 10.1007/BFb0040810.
- [28] Yuhui Shi and Russell C. Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC)*, volume 3, pages 1945–1950, Washington, DC, July 1999. IEEE. doi: 10.1109/CEC.1999.785511.
- [29] Ioan Christian Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 85(6):317–325, March 2003. doi: 10.1016/S0020-0190(02)00447-7.
- [30] Ministerie van Volksgezondheid, Welzijn en Sport. De zorg: hoeveel extra is het ons waard?, June 2012.
- [31] Semih Yalçındağ, Andrea Matta, and Evren Şahin. Operator assignment and routing problems in home health care services. In *International Conference on Automation Science and Engineering (CASE)*, pages 329–334, Seoul, 2012. IEEE. doi: 10.1109/CoASE.2012.6386478.
- [32] Li-ping Zhang, Huan-jun Yu, and Shang-xu Hu. Optimal choice of parameters for particle swarm optimization. *Journal of Zhejiang University SCIENCE A*, 6(6):528–534, June 2005. doi: 10.1631/jzus.2005.A0528.

Appendices

A.1 Appendix 1: Lijst van gebruikte afkortingen

CIZ	Centrum Indicatiestelling Zorg
ETS	Elite Tabu Search
evv'er	eerst verantwoordelijke verzorgende
HHC(S)P	Home Health Care (Scheduling) Problem
ILP	Integer Linear Programming
LIP	Local Improvement Procedure
LP	Linear Programming
MILP	Mixed Integer Linear Programming
MSP	Manpower Scheduling Problem
MSS	Master Surgical Schedule
MTSPTW	Multiple Traveling Salesman Problem with Time Windows
\mathcal{NP} -hard	Non-deterministic Polynomial-time hard
OPP	Operational Planning Problem
PSO	Particle Swarm Optimization
TSP	Traveling Salesman Problem
VRP(TW)	Vehicle Routing Problem (with Time Windows)

A.2 Appendix 2: Lijst van gebruikte symbolen

a_{c_j}	Beslissingsvariabele in het ILP-model: gelijk aan 1 als afspraak j bedoeld is voor cliënt c en anders gelijk aan 0.
a_i	De locatie waar afspraak i plaatsvindt (meestal het adres van cliënt c_i).
AP_k^t	De voorkeursmatrix van deeltje k tijdens iteratie t .
AP_{ijk}^t	Het (i, j) -element van de voorkeursmatrix AP_k^t .
B_{ijk}	Beschikbaarheid van medewerker i voor dienst j op dag k .
c_1, c_2	Acceleratieconstanten in de snelheidsfunctie van de PSO-heuristiek.
c_i	Het nummer van de cliënt voor wie afspraak i bedoeld is.
C	Het totaal aantal cliënten.
C_i	Contracturen van medewerker i (voor D dagen).
d_j	De dienst waarin afspraak j plaats moet vinden.
D	Het totaal aantal diensten in een planningsperiode.
e_c	De eerst verantwoordelijke verzorgende (evv'er) van cliënt c .
E_{jkl}	Het verwachte aantal medewerkers dat nodig is voor dienst j op dag k met minimale bevoegdheid l .
$f(X_k^t)$	De doelfunctiewaarde behorende bij deeltje k tijdens iteratie t .
$gbest^t$	De index van het beste deeltje tot en met iteratie t .
J_{max}	Maximaal aantal diensten dat een medewerker per dag mag werken.
K	Totaal aantal deeltjes dat gebruikt wordt in de PSO-heuristiek.
l_i	De minimale bevoegdheid die nodig is voor afspraak i .
m	Het totaal aantal medewerkers.
M	Een groot getal.
n	Het totaal aantal in te plannen afspraken.
$pbest_k^t$	De beste positie van deeltje k tot en met iteratie t .
r_c	Beslissingsvariabele in het ILP-model: gelijk aan 1 als cliënt c deze week gezien wordt door zijn of haar evv'er en anders gelijk aan 0.
\mathbf{r}_i	De route van medewerker i (een 3-dimensionale matrix bestaande uit afspraaknummers).
$\mathbf{r}_{(i,p,j,k)}$	Het afspraaknummer van het p^e bezoek tijdens dienst j op dag k van medewerker i .
R_i	Gedeelte aan contracturen dat medewerker i naar verwachting zal moeten werken.
s_i	De bevoegdheid van medewerker i .
t_i^{afw}	De absolute waarde van de afwijking van het geplande begintijdstip van afspraak i .
t_i^{dur}	De tijdsduur van afspraak i .
t_i^{end}	Het eind van het tijdsvenster waarin afspraak i moet beginnen.
t_i^{plan}	Het geplande tijdstip waarop afspraak i begint.
t_i^{start}	Het begin van het tijdsvenster waarin afspraak i moet beginnen.
t_{ij} (in ILP)	De reistijd van de locatie van afspraak i naar de locatie van afspraak j .
t_{ij} (in PSO)	De reistijd van locatie i naar locatie j .
t_{ij}^{tussen}	De tijd tussen twee afspraken i en j .
T_{ijk}^{end}	Het eindtijdstip van dienst j van medewerker i op dag k . Als medewerker i niet werkt tijdens dienst j op dag k , dan geldt $T_{ijk}^{end} = 0$.
T_{ijk}^{start}	Het begintijdstip van dienst j van medewerker i op dag k . Als medewerker i niet werkt tijdens dienst j op dag k , dan geldt $T_{ijk}^{start} = 0$.
U_{jk}	Verwachte aantal diensturen horende bij dienst j op dag k .
v_{ij}	Beslissingsvariabele in het ILP-model: gelijk aan 1 als afspraak j direct volgt op (nul)afspraak i en anders gelijk aan 0.

V_k^t	De snelheidsmatrix van deeltje k tijdens iteratie t .
V_{ijk}^t	Het (i, j) -element van de snelheidsmatrix V_k^t .
V_{min}, V_{max}	Onder- en bovengrens voor waarden van de snelheidsmatrices tijdens de initialisatie van de PSO-heuristiek.
w^t	De ‘traagheidsparameter’ in de snelheidsfunctie van de PSO-heuristiek.
w_{di}	Beslissingsvariabele in het ILP-model: gelijk aan 1 als medewerker i werkt tijdens dienst d en anders gelijk aan 0.
w_{ijk}	Het aantal afspraken in de route van medewerker i tijdens dienst j op dag k .
W	Het totaal aantal diensten dat gewerkt wordt.
W_i	Gedeelte aan contracturen dat medewerker i naar verwachting aan weekenddiensten zal moeten werken (zaterdag en/of zondag).
x_j	De afwijking van het geplande begintijdstip van afspraak j .
x_j^+	Het positieve gedeelte van x_j .
x_j^-	Het negatieve gedeelte van x_j .
X_k^t	De positiematrix van deeltje k tijdens iteratie t .
X_{ijk}^t	Het (i, j) -element van de positiematrix X_k^t .
X_{min}, X_{max}	Onder- en bovengrens voor waarden van de positiematrices tijdens de initialisatie van de PSO-heuristiek.
$y_{ji} (1 \leq j \leq n)$	Beslissingsvariabele in het ILP-model: gelijk aan 1 als afspraak j wordt uitgevoerd door medewerker i en anders gelijk aan 0.
$y_{ji} (n + 1 \leq j \leq n + W)$	Beslissingsvariabele in het ILP-model: gelijk aan 1 als afspraak j hoort bij medewerker i en anders gelijk aan 0.
z_c^{max}	Het maximale aantal zorgverleners dat ingezet mag worden bij cliënt c .
z_{ci}^{huidig}	Het aantal keer dat medewerker i deze week een afspraak van cliënt c uitvoert.
z_{ci}^{nieuw}	Beslissingsvariabele in het ILP-model: gelijk aan 1 als i na deze week in de afgelopen maanden bij cliënt c is ingezet en anders gelijk aan 0.
z_{ci}^{oud}	Beslissingsvariabele in het ILP-model: gelijk aan 1 als medewerker i in de afgelopen maanden bij cliënt c is ingezet en anders gelijk aan 0.
Z_c^{huidig}	De verzameling van alle medewerkers die deze week ingezet worden bij cliënt c .
Z_c^{nieuw}	De verzameling van alle medewerkers die na deze week ingezet zijn bij cliënt c .
Z_c^{oud}	De verzameling van alle medewerkers die tijdens de afgelopen weken ingezet zijn bij cliënt c .
α	Constante factor voor de organisatietijd in de doelfunctie van de cliëntenplanning.
β	Constante factor voor de robuustheid in de doelfunctie van de cliëntenplanning.
δ_{ij}	Gelijk aan v_{ij}^{tussen} .
Δ^*	De laagste kosten die gevonden zijn binnen de LIP.
Δ_j (in evv-controle)	De doelfunctiewaarde van de planning na de verwisseling met afspraak j .
Δ_j (in LIP)	De kosten van een dienst na het verwisselen van de j^e en de $(j + 1)^e$ afspraak in die dienst.
ξ	Het aantal niet-ingeplande afspraken.
ϕ	Gelijk aan $c_1 + c_2$.
χ	De constrictiefactor in de snelheidsfunctie van de PSO-heuristiek.
ψ	Het aantal cliënten dat niet door de evv'er wordt gezien in deze week.

A.3 Appendix 3: $L50$ -orthogonaalmatrix

Experiment	$[X_{min}, X_{max}]$	w_{start}	w_{end}	w_{vary}	λ	ϕ	c_1
1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2
3	1	1	3	3	3	3	3
4	1	1	4	4	4	4	4
5	1	1	5	5	5	5	5
6	1	2	1	2	3	4	5
7	1	2	2	3	4	5	1
8	1	2	3	4	5	1	2
9	1	2	4	5	1	2	3
10	1	2	5	1	2	3	4
11	1	3	1	3	5	2	4
12	1	3	2	4	1	3	5
13	1	3	3	5	2	4	1
14	1	3	4	1	3	5	2
15	1	3	5	2	4	1	3
16	1	4	1	4	2	5	3
17	1	4	2	5	3	1	4
18	1	4	3	1	4	2	5
19	1	4	4	2	5	3	1
20	1	4	5	3	1	4	2
21	1	5	1	5	4	3	2
22	1	5	2	1	5	4	3
23	1	5	3	2	1	5	4
24	1	5	4	3	2	1	5
25	1	5	5	4	3	2	1
26	2	1	1	1	4	5	4
27	2	1	2	2	5	1	5
28	2	1	3	3	1	2	1
29	2	1	4	4	2	3	2
30	2	1	5	5	3	4	3
31	2	2	1	2	1	3	3
32	2	2	2	3	2	4	4
33	2	2	3	4	3	5	5
34	2	2	4	5	4	1	1
35	2	2	5	1	5	2	2
36	2	3	1	3	3	1	2
37	2	3	1	3	3	1	2
38	2	3	3	5	5	3	4
39	2	3	4	1	1	4	5
40	2	3	5	2	2	5	1
41	2	4	1	4	5	4	1
42	2	4	2	5	1	5	2
43	2	4	3	1	2	1	3
44	2	4	4	2	3	2	4
45	2	4	5	3	4	3	5
46	2	5	1	5	2	2	5
47	2	5	2	1	3	3	1
48	2	5	3	2	4	4	2
49	2	5	4	3	5	5	3
50	2	5	5	4	1	1	4

Tabel 5: De $L50$ -orthogonaalmatrix die gebruikt is voor de Taguchi-methode. De zeven verschillende parameters staan op de bovenste rij en de getallen onder deze parameters geven aan welk level van de parameter er in het desbetreffende experiment wordt gebruikt.