

BACHELOROPDRACHT TECHNISCHE WISKUNDE

**Geautomatiseerde  
vrijwilligersplanning voor festival  
Amusing Hengelo**

*Studenten*

Liza FREDRIKS

Tim VAN DE KAMP

Bram DE WITTE

*Begeleider*

Dr.Ir. G.F. Post

16 januari 2013

**UNIVERSITEIT TWENTE.**

## Inhoudsopgave

<b>1</b>	<b>Samenvatting</b>	<b>3</b>
1.1	Probleembeschrijving . . . . .	3
1.2	Aanpak . . . . .	3
1.3	Resultaten en discussie . . . . .	3
<b>2</b>	<b>Dankwoord</b>	<b>4</b>
<b>3</b>	<b>Inleiding</b>	<b>5</b>
<b>4</b>	<b>Probleembeschrijving</b>	<b>6</b>
4.1	Niet aan de planning gebonden restricties . . . . .	6
4.2	Aan de planning gebonden restricties . . . . .	7
4.3	Voorbeeld . . . . .	8
<b>5</b>	<b>Wiskundig model</b>	<b>10</b>
5.1	Wiskundig model van de planning . . . . .	10
5.2	Wiskundig model van het algoritme . . . . .	11
<b>6</b>	<b>Aanpak</b>	<b>13</b>
6.1	Overkoepelend programma . . . . .	13
6.2	Data-invoer . . . . .	13
6.3	Constraints-controle . . . . .	13
6.4	Initial Volunteer Algorithm . . . . .	16
6.5	Improver . . . . .	24
6.6	Capaciteitscontrole . . . . .	26
<b>7</b>	<b>Resultaten</b>	<b>28</b>
<b>8</b>	<b>Resultatenanalyse</b>	<b>29</b>
8.1	Analyse van een planning . . . . .	29
8.2	Analyse van een capaciteitenrapport . . . . .	30
<b>9</b>	<b>Discussie</b>	<b>32</b>
9.1	Redenen dat automatische planning niet voldoet en verbeteringen . .	32
9.2	Verschillen met handmatige planning . . . . .	34
9.3	Bruikbaarheid van automatische planning . . . . .	35
<b>10</b>	<b>Aanbevelingen</b>	<b>37</b>
<b>11</b>	<b>Appendix</b>	<b>38</b>
11.1	Gebruikte data . . . . .	38
11.2	Voorbeeld van een gegenereerde vrijwilligersplanning . . . . .	41
11.3	Voorbeeld van een capaciteitenrapport . . . . .	45

# 1 Samenvatting

## 1.1 Probleembeschrijving

Amusing Hengelo is een eendaags festival met optredens op diverse podia door heel Hengelo. Het festival wordt mogelijk gemaakt door het werk van vrijwilligers. Op ieder geopend podium moet minstens één vrijwilliger aanwezig zijn. Het doel is om een programma te ontwikkelen dat een planning kan genereren, waarbij vrijwilligers aan een podium worden gekoppeld in een bepaald tijdsinterval. Deze gegenereerde planning mag geen lege plekken bevatten (een interval waarop een podium open is, maar waar niet genoeg vrijwilliger beschikbaar zijn), terwijl er rekening gehouden moet worden met de eisen van de vrijwilligers, zoals een pauze na 2,5 uur, en er moet rekening gehouden worden met de voorkeuren van een vrijwilliger, zoals het graag bij willen staan bij een bepaald podium.

## 1.2 Aanpak

Met behulp van de programmeeromgeving *Delphi* is er een programma gemaakt. In dit programma worden vrijwilligers gekoppeld aan een tijdsinterval, van 30 minuten, op een podium. Dit programma bestaat uit een onderdeel dat een initiële planning genereert, hierin worden eerst vrijwilligers ingepland die samen willen, vervolgens worden beperkt beschikbare vrijwilligers ingepland en op het laatst de overige vrijwilligers. Na de initiële planning wordt een verbeteringsalgoritme uitgevoerd. Dit verbeteringsalgoritme is erop gericht om het aantal lege plekken in de planning te doen verminderen. Dit doet het programma door een ongunstig ingeplande vrijwilliger ‘te pakken’ en vervolgens deze beter in te plannen. Het verbeteringsalgoritme, alsmede het algoritme voor de initiële planning, houden geen rekening met de voorkeuren, maar enkel met de eisen van de vrijwilligers.

Naast het genereren van de planning is er een functie ontwikkeld waarin wordt nagegaan of er wel genoeg vrijwilligers zijn voor het festival, dit heet de “capaciteitscheck”.

## 1.3 Resultaten en discussie

Het resultaat bestaat uit het programma wat een capaciteitsrapport generen kan en wat een vrijwilligersplanning maken kan. Wanneer de capaciteitscheck laat blijken dat er te weinig vrijwilligers zijn, dan kan er geen planning gemaakt worden zonder lege plekken. Het programma dat een planning kan genereren voldoet niet helemaal aan het doel. De gegenereerde plannings hebben tot nu toe altijd lege plekken in de planning gehad en daarnaast zijn er nog een aantal verbeteringspunten (zie de sectie 9, Discussie). Echter blijkt uit de capaciteitscheck en uit analyse van resultaten dat het ook erg lastig zal zijn om een planning te genereren zonder lege plekken met het huidige aantal vrijwilligers. Bovendien blijkt dat in de handmatige planning van de organisatie van Amusing Hengelo ook wel eens wat soepeler wordt omgegaan met de strikte eisen. In de aanbevelingen (sectie 10) wordt uitgelegd in hoeverre de resultaten bruikbaar zijn.

## **2 Dankwoord**

Een zeer groot woord van dank gaat naar onze begeleider Gerhard Post. Wij hebben dankbaar gebruik mogen maken van zijn deskundigheid en kennis van planningen én zijn ervaring met Delphi. Wij hebben bovendien de samenwerking als zeer plezierig ervaren, we zijn van mening dat dit zeker ten goede is gekomen aan de resultaten van deze opdracht.

Bovendien willen wij Kitty ter Braak en Martin Schoenmaker, van de organisatie van Amusing Hengelo bedanken voor deze uitdagende opdracht, hun prompte antwoord op vragen en het doorvoeren van aanpassingen in de data.

### 3 Inleiding

Amusing Hengelo is een jaarlijks terugkerend eendaags korenfestival in het centrum van Hengelo. Op podia verdeeld over de hele stad genieten duizenden belangstellenden van allerlei muzieksoorten, variërend van soul tot pop en van religieus tot schlager. Het festival is begonnen in 2006 en bleek een groot succes te zijn onder zowel jong als oud. In 2012 is het festival uitgegroeid tot een festival met 3330 nationale en internationale deelnemers uit 114 koren die samen zorgden voor maar liefst 221 optredens.

Een festival van dergelijke grootte vereist natuurlijk erg veel voorbereidingen. Eén van de onderdelen van de voorbereiding is de planning van het festival, wat over het algemeen erg veel tijd en moeite kost. Een automatisering van deze planning is dan ook een zeer gewenst hulpmiddel voor de organisatie van Amusing Hengelo. Een automatisering van het inplannen van koren is al beschikbaar, echter een automatisering van het inplannen van vrijwilligers moet nog gemaakt worden. Deze opdracht nemen wij op ons.

Vrijwilligers zijn een bepaalde tijd aanwezig bij een podium en dienen als aanspreekpunt voor zowel de muzikanten als de bezoekers. Ook zorgen vrijwilligers ervoor dat muzikanten zich volledig kunnen richten op hun optreden en zich niet druk hoeven te maken over, bijvoorbeeld het bewaken van apparatuur.

Het inplannen van vrijwilligers is gebonden aan een flink aantal voorwaarden. Naast een aantal vanzelfsprekende eisen, bijvoorbeeld dat een vrijwilliger niet kan ‘werken’ als deze niet beschikbaar is en dat een vrijwilliger een pauze moet hebben na 2,5 uur gewerkt te hebben, zijn er ook een aantal minder vanzelfsprekende eisen, bijvoorbeeld dat een vrijwilliger altijd een kwartier langer aanwezig moet zijn bij een podium terwijl de volgende vrijwilliger al ingeplant staat (om voor overlap te zorgen) en dat het openen van een podium door (minstens) één ervaren vrijwilliger moet worden gedaan. Naast deze eisen moeten wij ook rekening houden met een aantal voorkeuren van vrijwilligers. Voorkeuren kunnen bijvoorbeeld zijn dat een vrijwilliger graag bij een bepaald koor of podium wil werken.

In dit verslag zullen wij beschrijven hoe wij tot een automatisering van de korenplanning zijn gekomen. Achtereenvolgens worden er in dit verslag een probleembeschrijving, een modelschets, een beschrijving van de manier waarop wij dit geprogrammeerd hebben, resultaten, een discussie en een aanbeveling gegeven. Het verslag eindigt met een appendix, hierin staan enkele resultaten en wordt er een resultaat geanalyseerd.

## 4 Probleembeschrijving

Bij het korenfestival Amusing Hengelo zijn jaarlijks ongeveer 70 vrijwilligers actief om het festival in goede banen te lijden. De vrijwilligers kunnen diverse taken hebben. Zo zijn er vrijwilligers die parkeerwacht zijn, mensen die achter de infokraam zitten, lunchpakketjes maken of bij podia staan ter ondersteuning. Voor deze laatste groep van ongeveer 50 vrijwilligers zal een rooster worden gemaakt.

Het festival duurt van 11:00 uur tot 17:00 uur, dan treden de koren op. Vóór de opening van het festival en na de sluiting, zijn er wel vrijwilligers nodig voor de open afbouw. De eerste vrijwilliger is nodig vanaf 9:30 uur en de laatste vrijwilliger tot 18:00 uur. Een optreden van een koor duurt een half uur. De vrijwilligers worden per half uur ingepland, op zo'n wijze dat een vrijwilliger bij één of meerdere volledige optredens staat. Wanneer er geen optredens zijn, maar wel vrijwilligers nodig, dan worden de vrijwilligers ook per half uur ingezet. De festivaldag kan dus worden opgedeeld in 17 *tijdstappen* van elk een half uur. Een *event* is de combinatie van een tijdstap samen met één van de 20 podia.

Bij het maken van een rooster voor de vrijwilligers bij het korenfestival, moet rekening worden gehouden met verschillende dingen. Er zijn voorkeuren vanuit de organisatie, bijvoorbeeld dat iemand na 2,5 uur werken een pauze moet hebben, en voorkeuren vanuit de vrijwilliger, bijvoorbeeld bij welk podium de vrijwilliger graag wil helpen. Al deze beperkingen (*constraints*) voor het rooster staan hieronder vermeld. Er wordt onderscheid gemaakt tussen constrains die van de planning afhangen en constrains die niet planningsafhankelijk zijn. Daarnaast worden er twee typen beperkingen beschouwd: *hard* en *soft*. Een beperking is hard wanneer er altijd aan deze beperking voldaan moet worden. Wanneer de beperking eigenlijk een voorkeur is, dan is deze van het type soft.

### 4.1 Niet aan de planning gebonden restricties

In tabel 1 staat een overzicht gegeven van de niet aan de planning gebonden restricties. Deze restricties moeten onafhankelijk van de vrijwilligersplanning gelden.

*Ad 1* Iedere vrijwilliger heeft aangegeven tussen welke tijden hij beschikbaar is. De vrijwilliger kan alleen worden ingepland op een bepaalde tijdstap als hij ook op deze tijdstap beschikbaar is.

*Ad 2* Op de meeste podia is één vrijwilliger nodig. Op de andere podia moeten echter twee vrijwilligers worden ingepland, dit alleen wanneer er een koor optreedt op een event. Dus op een podium waarop twee vrijwilligers moeten, mag er ook één staan wanneer er geen koor optreedt. Op ieder event moeten dus 0, 1 of 2 vrijwilligers worden gepland.

Bij een podium waar twee vrijwilligers ingedeeld staan, is het niet noodzakelijk dat twee vrijwilligers tegelijk vervangen worden.

*Ad 3* Bij de opening van ieder podium moet minimaal één ervaren vrijwilliger ingepland worden.

Nr.	Type	Constraint
1	Hard	Beschikbaarheid van vrijwilliger
2	Hard	Benodigd aantal vrijwilligers per podium; één of twee
3	Hard	Eén ervaren vrijwilligers is nodig bij de opening van het podium
4	Hard	Vrijwilligers willen mogelijk samen met een andere vrijwilliger werken
5	Hard	Vrijwilligers kunnen een afkeur hebben voor bepaalde podia
6	Hard	Vrijwilligers kunnen een afkeur hebben voor bepaalde koren
7	Hard	Vrijwilligers worden niet ingepland tijdens een eigen optreden
8	Hard	Zingende vrijwilligers hebben een half uur pauze voor hun eigen optreden
9	Hard	Zingende vrijwilligers mogen wel dienst hebben net na hun optreden, maar alleen op het podium waar ze hun optreden hadden
10	Soft	Vrijwilligers kunnen een voorkeur hebben voor één podium
11	Soft	Vrijwilligers kunnen een voorkeur hebben voor één koor

Tabel 1: Overzicht van niet aan de planning gebonden restricties.

*Ad 4* De vrijwilligers die samen willen werken blijven altijd bij elkaar. De twee samenwerkende vrijwilligers moeten samen worden ingepland. Wanneer slechts één van de samenwerkende vrijwilligers beschikbaar is, mag deze beschikbare vrijwilliger wel worden ingepland.

*Ad 10* Wanneer meerdere vrijwilligers op één podium willen staan en verder niet, dan kunnen deze vrijwilligers in dat geval op een ander podium worden ingedeeld.

## 4.2 Aan de planning gebonden restricties

Planningsgebonden restricties zijn restricties die afhangen van de eigen vrijwilligersplanning. Een voorbeeld hiervan is dat een vrijwilliger niet langer dan 2,5 uur achtereen mag werken. Hierbij moet men dus al weten of de vrijwilliger al een tijd staat ingepland of niet. Tabel 2 geeft een overzicht van deze planningsafhankelijke restricties.

*Ad 1* Wanneer een vrijwilliger van podium verwisselt, mag hij het podium niet onbeheerd achterlaten. Om te voorkomen dat de ene vrijwilliger al weer naar het volgende podium moet, terwijl de vrijwilliger die hem aflost er nog niet is, moet een *overlap* in worden gepland. Tijdens een overlap staan er dus meerdere vrijwilligers ingepland op het podium.

Om te wisselen tussen twee verschillende podia is een reistijd van een kwartier nodig. Wanneer een vrijwilliger wisselt tussen twee podia is hiervoor overlap en reistijd van beide een kwartier nodig.

*Ad 2* Een pauze duurt minimaal een half uur. Reistijd tussen podia wordt niet beschouwd als pauze.

Nr.	Type	Constraint
1	Hard	Bij iedere podiumwissel moet er een kwartier overlap zijn
2	Hard	Pauze na maximaal 5 diensten van een half uur achter elkaar, met als uitzondering mensen die slechts 6 diensten ingezet worden
3	Hard	De tijd tussen het begin van de eerste en het eind van de laatste dienst van een vrijwilliger mag niet langer zijn dan het maximaal aantal uren dat deze vrijwilliger wil worden ingepland
4	Hard	Wanneer een vrijwilliger meer dan 5 diensten ingeroosterd wordt, moet deze op twee verschillende podia worden ingedeeld, tenzij hij slechts 6 diensten ingezet wordt
5	Soft	Een vrijwilliger mag niet terugkeren bij hetzelfde podium
6	Soft	Vrijwilligers die wisselen van podium, moeten minimaal één keer binnen en één keer buiten staan

Tabel 2: Overzicht van aan de planning gebonden restricties.

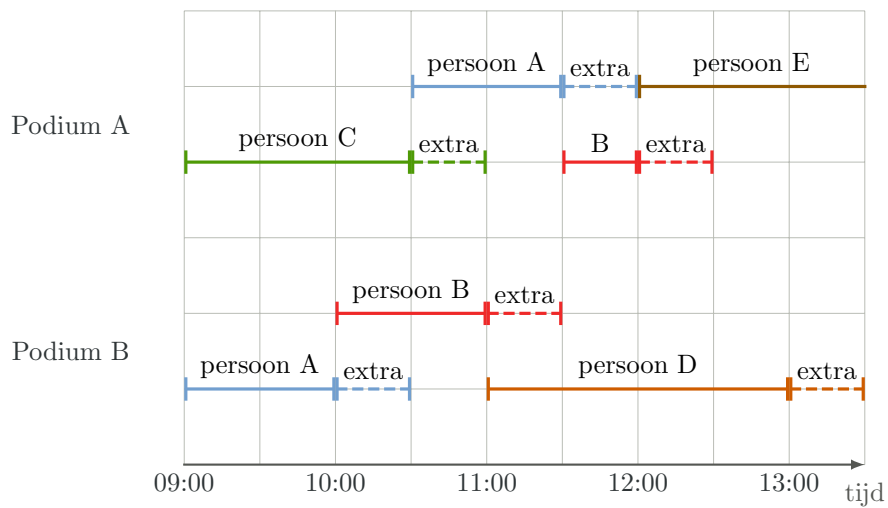
*Ad 3* Vrijwilligers kunnen een maximum aantal uren dat ze willen helpen aangeven. Zo kan bijvoorbeeld een vrijwilliger de gehele dag helpen, maar wil deze maar maximaal twee uur als vrijwilliger ingepland staan.

### 4.3 Voorbeeld

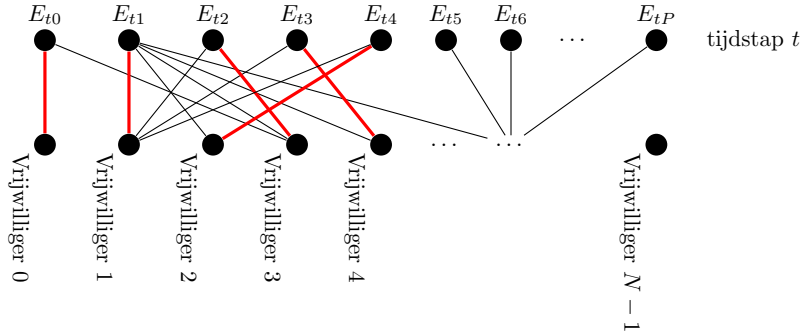
Figuur 1 is dan een voorbeeld van een correcte (maar niet efficiënte) planning. In dit geval is een half uur wisseling ingepland (een kwartier overlap + een kwartier reistijd van het ene podium naar het andere podium).

Hierin mag Persoon A, omdat hij 6 tijdstappen achtereen is ingezet, de rest van de dag niet meer werken (zie *Ad 2*). Personen B en D moeten een pauze hebben om 12:30 uur, respectievelijk 13:30 uur, zij werken namelijk 5 diensten achtereen. Personen A en B wisselen van podium zonder pauze te nemen.





Figuur 1: Voorbeeld vrijwilligersschema waarbij extra bestaat uit een kwartier overlap en een kwartier (ingeplande) reistijd.



Figuur 2: Matching tussen events en vrijwilligers.

## 5 Wiskundig model

In deze sectie wordt het overkoepelende wiskundige model bij de planning gepresenteerd. Bovendien wordt het wiskundige model beschreven dat dicht bij ons algoritme staat.

### 5.1 Wiskundig model van de planning

Bij het automatiseren van de vrijwilligersplanning wordt gebruik van (discrete) events gemaakt. Alle events duren 30 minuten en stellen een podium-tijdstap-combinatie voor. Zie de probleembeschrijving (sectie 4) voor een uitleg over events en tijdstappen. Een event wordt in dit verslag weergegeven met  $E_{tp}$ , hierin is  $p$  het podium(nummer) en  $t$  stelt hierin de  $t^e$  tijdstap voor. Een event bestaat alleen wanneer een podium open is,  $t$  hoeft dus niet bij 0 te beginnen.  $P$  is het aantal podia,  $T$  is het aantal tijdstappen en er zijn  $N$  vrijwilligers beschikbaar. Voor ieder event is bekend wat het aantal benodigde vrijwilligers zijn voor dit event (weergegeven met  $\alpha_{tp}$ ) en welk koor optreedt op dit event. Benodigd hiervoor is dat de korenplanning al uitgevoerd is en onveranderd blijft, alvorens de vrijwilligersplanning gestart wordt.

Het doel van de planning is om aan ieder event het gewenste aantal vrijwilligers toe te kennen. Dit kan gezien worden als een ‘matching’-probleem in  $T$  bipartiete grafen, voor ieder tijdstap een graaf. Zij  $G_t$  een bipartiete graaf, behorend bij tijdstap  $t$ , met bipartitie  $(X_t, Y_t)$ . In  $G_t$  zijn er voor elk podium  $\alpha_{tp}$  punten in de verzameling  $X_t$  en is er voor iedere vrijwilliger een punt in de verzameling  $Y_t$ . Als er in  $G_t$  een lijn loopt van een vrijwilliger naar een event, dan is die vrijwilliger beschikbaar voor dat event. Of er een lijn loopt is afhankelijk van de voorwaarden aan de planning, zoals beschreven in de probleembeschrijving (sectie 4). We willen in  $G_t$  een matching maken zodat ieder punt in  $X_t$ , met een graad groter dan nul, matcht met een punt in  $Y_t$ . Wanneer de graad van een punt in  $X_t$  gelijk is aan nul, dan is er dus geen vrijwilliger beschikbaar voor dit event. Er zal een lege plek in de planning ontstaan, dit wordt een *gat* genoemd. Figuur 2 geeft een voorbeeld van een matching in graaf  $G_t$  weer, in dit geval voor een tijdstap waarop alle podia open zijn en waarbij  $\alpha_{tp} = 1 \forall p$ . Aangezien er voorwaarden (zoals pauzes en overlap) zijn die afhankelijk zijn van voorgaande matchings, zijn alle matchings in de  $T$  grafen afhankelijk van elkaar.

---

$X_{itp}$	{	1 als vrijwilliger $i$ is ingepland op podium $p$ en tijdstap $t$
	}	0 anders
$K_{itp}$	{	1 als vrijwilliger $i$ een kooroptreden op podium $p$ en tijdstap $t$ heeft
	}	0 anders
$B_{tp}$	{	1 als er op podium $p$ een kooroptreden is op tijdstap $t$
	}	0 anders
$m_i$		Maximaal aantal halve uren dat vrijwilliger $i$ ingezet wil worden
$b_i$		Nummer van de eerst ingeplande tijdstap van vrijwilliger $i$
$e_i$		Nummer van de laatst ingeplande tijdstap van vrijwilliger $i$
$\alpha_{tp}$		Benodigd aantal vrijwilligers op podium $p$ op tijdstap $t$
$x_i$		Kosten van vrijwilliger $i$

---

Tabel 3: Overzicht van variabelen voor harde constrains.

Tabel 3 geeft de parameters weer die gebruikt gaan worden. Alle parameters, behalve  $X_{itp}$ , zijn bekend en kunnen als data ingelezen worden.  $X_{itp}$  is de parameter, die aangeeft of vrijwilliger  $i$  is ingepland (of niet) op podium  $p$  en tijdstap  $t$ . Het hoofddoel van de opdracht is, de gelijkheid in de belangrijkste voorwaarde van het planningsprobleem,

$$\sum_{i=0}^{N-1} (X_{itp} = \alpha_{tp}) \quad \forall t, p \quad (1)$$

te behouden. Als dit niet mogelijk zal zijn, kan overgestapt worden op

$$\sum_{i=0}^{N-1} (X_{itp} + Y_{tp} \geq \alpha_{tp}) \quad \forall t, p \quad (2)$$

met  $Y_{tp}$  als (integer) dummy-variabele en met

$$\sum_{p=0}^{P-1} \sum_{t=0}^{T-1} Y_{pt} \quad (3)$$

als de te minimaliseren functie. Dit komt op hetzelfde neer als het maximaliseren van  $X_{itp}$  over alle  $i, t$  en  $p$ . Alle overige constrains, zoals pauzes en overlap, kunnen ook met behulp van de parameters in tabel 3 gemaakt worden. Bijvoorbeeld om te voorkomen dat een vrijwilliger een optreden en een podiumdienst tegelijk heeft, kan van

$$X_{itp} \leq 1 - \sum_{p=0}^{P-1} K_{itp} \quad \forall i, t, p$$

gebruikt gemaakt worden.

## 5.2 Wiskundig model van het algoritme

Het maken van de matching in graaf  $G$ , zie vorige paragraaf, kan gebeuren op meerdere manieren. Ten eerste kan dit bijvoorbeeld worden gedaan door een integer

linear programming probleem (*ILP-probleem*) op te lossen met de voorwaarden beschreven in sectie 4, en met constraint (1) of eventueel (2). De kostenfunctie kan (3) zijn, eventueel uitgebreid met 'kosten' bij de soft constraints. Er wordt echter niet voor deze methode gekozen, dit omdat bij het oplossen van een ILP-probleem geavanceerde, commerciële software noodzakelijk is, die niet beschikbaar is.

Wij kiezen ervoor om de planning constructief te creëren. Dit houdt in dat we starten met het inplannen van een vrijwilliger op het eerste event, de eerste tijdstap dat het eerste podium open is, hierna verder te kijken naar een volgend podium op dezelfde tijdstap. Als voor ieder event geprobeerd is om een vrijwilliger in te plannen dan is er een initiële planning voltooid. Wanneer er nog gaten in deze planning zitten dan zal deze initiële planning nog verbeterd worden. In deze verbeteringsrondes zal telkens een vrijwilliger worden 'gepakt' met veel gaten in zijn planning (en dus ongunstig is ingepland). Vervolgens zal geprobeerd worden om deze vrijwilliger beter in te plannen. Wanneer er al geprobeerd is om een vrijwilliger beter in te plannen, dan zal dit niet nogmaals geprobeerd worden. Er wordt getracht op deze manier naar een lokaal minimum van het aantal gaten in de planning te convergeren. Wanneer de initiële planning niet meer verbeterd kan worden, of na een bepaald aantal rondes, zal er een nieuwe initiële planning gegenereerd worden. In combinatie met het verbeteringsalgoritme wordt getracht zo naar een globaal minimum van het aantal gaten in de planning te convergeren.

Wanneer er een planning is gegenereerd met nul gaten in de planning, dan kan er eventueel nog een verbeteringsalgoritme komen dat de kosten van de vrijwilligers minimaliseert. Deze zal op dezelfde manier vormgegeven kunnen worden als het verbeteringsalgoritme dat het aantal gaten minimaliseert.

## 6 Aanpak

Om de planning van de vrijwilligers te vereenvoudigen, wordt aangenomen dat de planning van de koren al bekend is en vast staat.

Het vrijwilligersplanningsprogramma is een uitbreiding op het korenplanningsprogramma wat eerder gemaakt is [1].

### 6.1 Overkoepelend programma

Het programma bestaat globaal gezien uit vier belangrijke stappen welke worden weergegeven in figuur 3. Allereerst zal het preprocess in werking gaan. Dit houdt in dat de data van het festival waar naar gekeken gaat worden, ingeladen wordt. Dit wordt nader toegelicht in sectie 6.2. Wanneer dit is gebeurd, kan de uitvoerder aangeven hoeveel keer hij het programma wil laten lopen (*runs*). Eén run bestaat uit het *Initial Volunteer Algorithm* (nader toegelicht in sectie 6.4) en de *Improver* (sectie 6.5). Men verkrijgt iedere keer een nieuwe oplossing doordat in de algoritmen willekeurig wordt toegepast. Wanneer deze beide algoritmen zijn doorlopen, wordt er gekeken of de eindoplossing van de run beter is dan de huidige beste oplossing, waarna de nieuwe beste oplossing wordt opgeslagen. Dit proces herhaalt zich tot het maximum aantal runs wat de uitvoerder in het programma heeft aangegeven. Als dit maximum is bereikt zal de beste planning, de eindoplossing, worden getoond.

### 6.2 Data-invoer

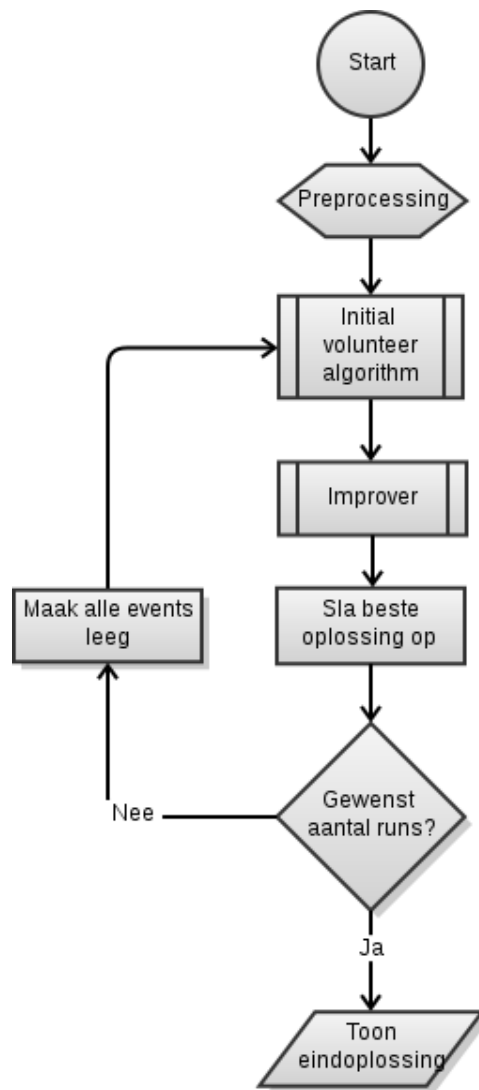
De organisatie van Amusing Hengelo heeft een website waar vrijwilligers en koren zich aan kunnen melden. Ook is het voor de organisatie mogelijk een *XML-bestand* met de gegevens over het festival te exporteren. In het XML-bestand staan bij iedere vrijwilliger relevante gegevens, zoals vanaf en tot welke tijdstippen de vrijwilliger beschikbaar is. Al deze relevante data wordt ingelezen en opgeslagen in het *vrijwilliger-object*.

### 6.3 Constraints-controle

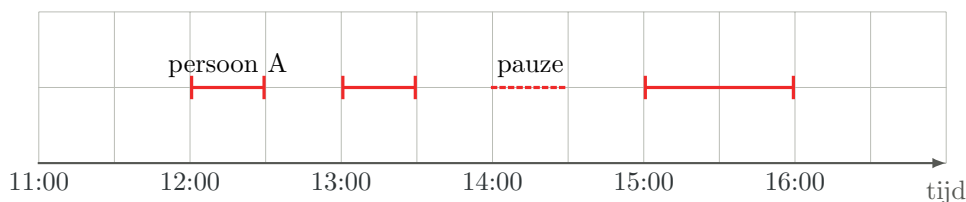
Voor ieder event is te controleren of de vrijwilliger in zou mogen worden gepland. De meeste (harde) planningsafhankelijke restricties (tabel 1) zijn eenvoudig te controleren. De eigenschappen van een vrijwilliger zijn al in de XML-data gegeven en eigenschappen van onder andere events, podia en koren zijn eenvoudig via het object op te vragen. Zo is het triviaal te controleren of bijvoorbeeld een vrijwilliger wel op een event mag staan waar ook het koor van zijn afkeur op dat moment optreedt. Het aantal benodigde vrijwilligers voor een event moet wel worden bepaald, dit is één wanneer er géén koor optreedt op dit event en treedt er wél een koor op, dan is het, volgens de XML-data, aantal vrijwilligers voor het podium benodigd.

De planningsafhankelijke restricties (tabel 2) zijn over het algemeen moeilijker te implementeren.

In de implementatie kan er voor worden gekozen om de overlap wél of niet in te plannen. Om de implementatie niet nodeloos ingewikkeld te maken, is ervoor gekozen om géén overlap in te plannen. Dit heeft als voordeel dat er steeds, op ieder event, maar één of twee vrijwilligers hoeven te worden ingepland. Een nadeel



Figuur 3: Overkoepelend programma.



Figuur 4: Vrijwilliger A kan niet tussen 14:00 uur en 14:30 uur worden ingepland, maar nog wel tussen 13:30 uur en 14:00 uur òf tussen 14:30 uur en 15:00 uur.

van deze implementatie is dat de overlap ‘berekend’ moet worden. Om te bepalen waar een overlap moet zitten, wordt er gekeken naar de vorige en volgende tijdstap. Wanneer de vrijwilliger op het één van deze twee tijdstappen op een ander podium staat, kan hij niet worden ingepland op dit event.

Net als bij het bepalen van een overlap, moet bij het bepalen of de vrijwilliger een pauze nodig heeft voor- en achteruit in de tijd worden gekeken. In plaats van één tijdstap voor- of achteruit te kijken, wordt er nu naar het aantal tijdstappen tussen het, te plannen, event en de eerst volgende twee nog ongeplande tijdstappen. Wanneer de som tussen het aantal tijdstappen na, en het aantal tijdstappen voor het event groter is dan vijf, moet er een pauze worden gehouden. Er is echter één uitzondering: wanneer de vrijwilliger doorwerkt tot het laatste moment, hoeft deze geen overlap te hebben en mag hij dus voor vijf, in plaats van slechts vier, tijdstappen worden ingezet. In figuur 4 staat een voorbeeld van een gedeeltelijk ingeplande vrijwilliger. Tussen 12:00 uur en 16:00 uur heeft deze vrijwilliger in ieder geval een pauze nodig. Uit het zojuist beschreven algoritme blijkt dat de vrijwilliger nog op diverse tijdstappen inzetbaar is, maar tussen 14:00 uur en 14:30 uur een pauze moet hebben.

Om te bepalen of een vrijwilliger niet langer ingezet wordt dan hij zou willen, moet naar het eerste en laatste event worden gekeken dat hij wordt ingezet. Ligt het in te plannen event tussen deze twee tijdstappen, dan mag de vrijwilliger ingepland worden. Valt het in te plannen event buiten dit interval, dan moet gecontroleerd worden of daarmee de maximale beschikbaarheid ( $m_i$ ) niet mee wordt overschreden. Door het begin van de eerste tijdstap en het eind van de laatste tijdstap van elkaar af te trekken, verkrijgt men de totale tijd dat de vrijwilliger ingezet zou worden. Deze waarde moet dus kleiner gelijk blijven aan  $m_i$ .

In het huidige programma wordt nergens afgedwongen dat een vrijwilliger na een pauze op een ander podium moet komen te staan. Ook met de harde constraint 4 in tabel 2 is geen rekening gehouden. Er wordt wel, in de gevallen waar dat nodig is, met een overlap rekening gehouden. Dus na een pauze mag een vrijwilliger altijd van podium wisselen (dit wordt dus niet afgedwongen).

### 6.3.1 Soft constraints

In de probleemomschrijving worden enkele soft constraints genoemd in de tabellen 1 en 2. Deze voorwaarden zijn door de opdrachtgever aangegeven om te kunnen voldoen aan de wensen van sommige vrijwilligers of de wensen van de opdrachtgever zelf.

De soft constraints zijn om te zetten naar kosten. Wanneer een planning is gemaakt, kan worden gekeken welke vrijwilliger niet voldoet aan de gewenste voorwaarden. Aan iedere voorwaarde kunnen eigen kosten worden gehangen, afhankelijk van hoe erg het overtreden van deze wens is. Voor het berekenen van deze kosten is in het programma een functie geschreven. Er is echter geen algoritme aanwezig dat na het berekenen van de kosten gaat kijken waar er verbetering plaats kan vinden. Er is gekozen om eerst te kijken naar het verminderen van de gaten in de planning. Dit is gedaan aangezien wij de prioriteit hoger achten om alle plekken op te vullen die voldoen aan de harde, verplichte voorwaarden, dan om te kiezen voor het voldoen aan de zachtere wensen van de vrijwilliger en opdrachtgever.

## 6.4 Initial Volunteer Algorithm

Het Initial Volunteer Algorithm construeert een eerste planning. Dit kan gebeuren als het preprocess, het inlezen van de data en het door de gebruiker aan te geven aantal runs invoeren, gedaan is. Hoe dit algoritme is opgebouwd zal hieronder worden toegelicht. Tevens is hier een flowchart, figuur 5 van, welke het algoritme globaal toelicht.

### 6.4.1 Globaal Initial Volunteer Algorithm

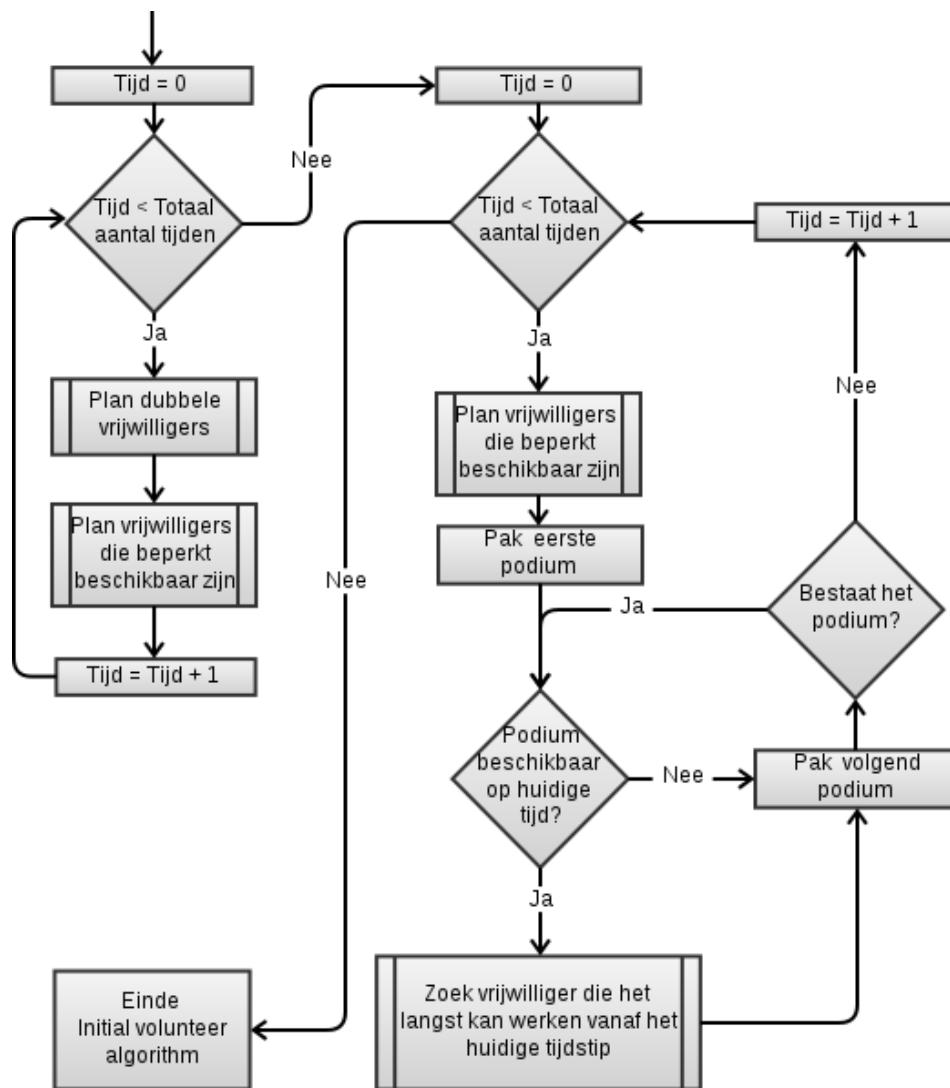
Het algoritme gaat allereerst de mensen inplannen die samen op een podium willen staan. Deze mensen kunnen alleen maar op podia staan waar twee mensen nodig zijn en daarnaast zijn ze erg afhankelijk van elkaar. Wanneer één van de twee op een podium wordt ingepland waar maar één vrijwilliger nodig is, betekent dat direct dat de ander nergens meer op die tijdstap mag komen te staan. Aangezien we de beschikbaarheid van de vrijwilligers zo veel mogelijk zullen moeten benutten om alle events te vullen, is het goed om met deze mensen te beginnen. Hoe dit onderdeel van het algoritme werkt is te lezen in sectie 6.4.2.

Nadat de vrijwilligers zijn ingepland die graag samen staan, wordt gekeken naar de mensen die maar beperkt beschikbaar zijn. Aangezien we de gaten in de planning willen verminderen, zal ook van deze mensen de beschikbaarheid zo veel mogelijk gebruikt moeten worden. Wanneer iemand beperkt beschikbaar is, wordt gedefinieerd door  $m_i < 7$  óf de totale beschikbaarheid van de vrijwilliger is  $< 7$  óf hij is vanaf het huidige tijdstip minder dan 5 tijdstappen beschikbaar. In sectie 6.4.3 wordt dit gedeelte nader toegelicht.

Wanneer de dubbele vrijwilligers en de beperkt beschikbare vrijwilligers voor een tijdstap zijn gepland, wordt het proces vervolgd door naar de volgende tijdstap te kijken, tot hij alle tijden heeft doorlopen.

Nu het framework van de planning vaststaat door het plannen van vrijwilligers die samen met iemand willen, en vrijwilligers die beperkt beschikbaar zijn, worden de andere events gevuld. Dit gebeurt door wederom over alle tijden van vooraf aan te lopen. Telkens zal er eerst worden gekeken of er nog vrijwilligers zijn die voldoen aan de voorwaarden van het beperkt beschikbaar zijn, en of deze dan ingepland kunnen worden. Hierna zal er niet meer per vrijwilliger worden gekeken, maar per podium. Op ieder podium zal gekeken worden welke vrijwilliger het langst beschikbaar is vanaf het huidige tijdstip, te zien in de flowchart in figuur 8. Er kan nu worden gezocht naar degene die het langst beschikbaar is op het benodigde aantal uren,





Figuur 5: Initial Volunteer Algorithm.

omdat iedereen die wordt bekeken op deze tijdstap niet wordt beschouwd als beperkt beschikbare vrijwilliger. Naar deze mensen wordt namelijk telkens bij elke tijdstap al gekeken of die ingepland kunnen worden. Hoe er wordt gezocht naar de vrijwilligers die het langst beschikbaar zijn op het benodigde aantal uren, wordt in sectie 6.4.4 nader toegelicht.

Het Initial Volunteer Algorithm vervolgt door op de huidige tijdstap alle podia te bekijken. Wanneer alle podia zijn bekeken, zal de tijd een stap vooruit worden gezet waarna het proces van het plannen van beperkt beschikbare vrijwilligers en het plannen van mensen die het best de benodigde tijd benaderen met hun beschikbaarheid zich herhalen. Dit gebeurt tot alle tijdstappen zijn geweest. Dan is de initiële planning gemaakt.

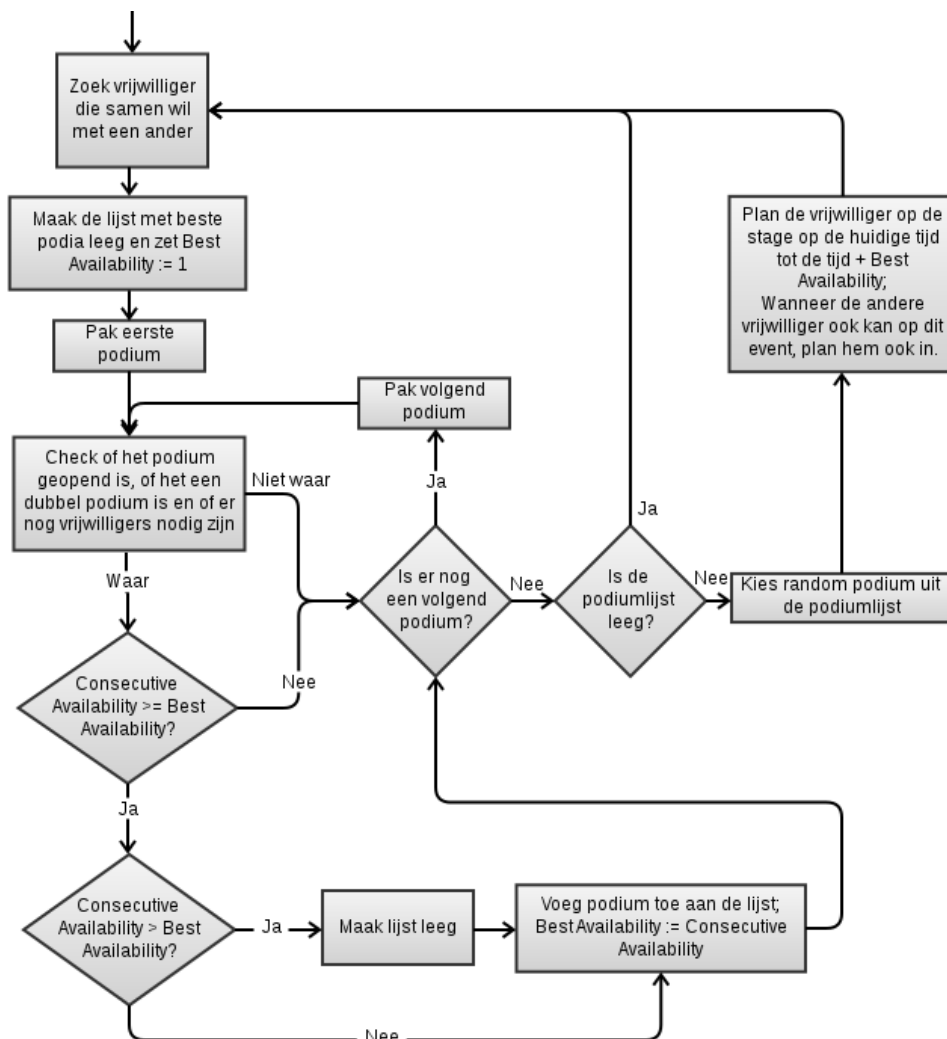
Aangezien de initiële planning nog veel gaten heeft, volgt een Improver op de planning om te kijken of er door het schuiven van vrijwilligers uren vrijkomen waarmee de gaten opgevuld kunnen worden.

#### 6.4.2 Plannen van vrijwilligers die samen willen

Het plannen van vrijwilligers die graag samen willen met iemand anders, is een onderdeel van het Initial Volunteer Algorithm. Allereerst gaat het programma op zoek naar een vrijwilliger die aan heeft gegeven samen met een andere vrijwilliger te willen helpen. Hierna kijkt hij op welk podium deze vrijwilliger het best kan worden ingepland. Dit zullen alleen podia zijn waarop meerdere vrijwilligers nodig zijn. Van de gevonden vrijwilliger wordt op het huidige event, de tijd-podium combinatie, de opeenvolgende beschikbaarheid berekend. De opeenvolgende beschikbaarheid is het aantal halve uren dat de vrijwilliger aaneengesloten beschikbaar is vanaf het huidige event. Wanneer deze beschikbaarheid absoluut groter is dan de tot nu toe beste beschikbaarheid, wordt het podium beschouwd als beste podium tot nu toe. Wanneer hij even groot is, zal hij toegevoegd worden aan de lijst met beste podia. Dit wordt gedaan voor alle podia.

Als alle podia zijn geprobeerd, wordt de vrijwilliger zo lang als zijn beste beschikbaarheid ingepland op een random podium uit de lijst met beste podia, wanneer deze lijst niet leeg is. Er wordt bij elke keer als er wordt ingepland gekeken of degene waar de vrijwilliger samen mee wil ook beschikbaar is. Indien dat zo is, wordt deze ook op hetzelfde event gepland. Aangezien de paren ongeveer dezelfde beschikbaarheid hebben, met uitzondering wanneer één iemand een optreden heeft, zullen de vrijwilligers veel samen worden ingepland. In tabel 4 is van elke vrijwilliger zijn beschikbaarheid te zien. Twee van de drie paartjes hebben exact dezelfde beschikbaarheid, één heeft een afwijking (aangezien beschikbaar zijn vanaf 8 uur geïnterpreteerd kan worden als starten om 9:30 uur) in het maximaal aantal halve uren.

Het kan voorkomen dat één van beiden wordt ingepland. Dit is toegestaan volgens de probleemomschrijving, zolang de ander niet ergens anders wordt ingepland op dat moment. En aangezien de andere vrijwilliger wel zou worden ingepland als hij kon, worden de vrijwilligers zo veel als ze kunnen ingepland. De enige uitzondering hierop is wanneer de eerste vrijwilliger op een podium wordt ingepland waar de andere niet op wil staan. Dan is de andere nog wel beschikbaar voor een ander podium, maar mag daar dan niet meer worden ingepland omdat de eerste persoon al in zijn eentje is ingepland. De planning zou dus afhankelijk kunnen zijn van welke vrijwilliger,



Figuur 6: Plan vrijwilligers die samen willen.

van het duo, men als eerste inplant. In de geleverde data door de opdrachtgever staat al de afkeur van één vrijwilliger ook bij de andere vrijwilliger. Met de huidige data zal het dus niet voor kunnen komen dat de eerste vrijwilliger op een podium komt te staan waar de andere vrijwilliger niet wil.

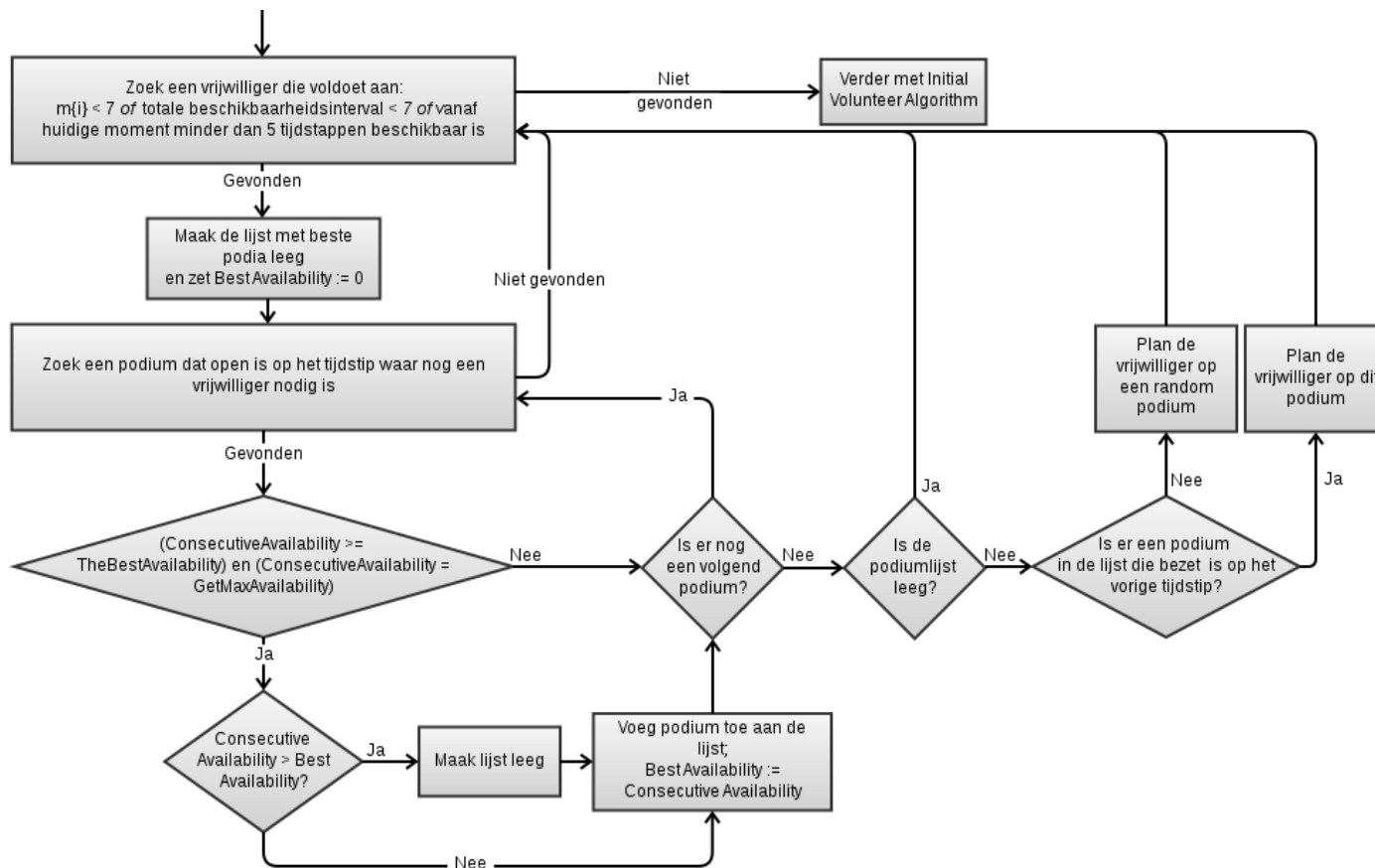
De flowchart van dit onderdeel is te zien in figuur 6.

#### 6.4.3 Plannen van beperkt beschikbare vrijwilligers

Voor de beperkt beschikbare mensen op een tijdstap wordt gezocht naar het podium waar zij het langst op kunnen staan. Dit wordt op dezelfde wijze gedaan als bij het plannen van vrijwilligers die samen willen, wat te vinden is in 6.4.2. Hierbij wordt echter een beter podium niet alleen bepaald doordat de opeenvolgende beschikbaarheid, het aantal halve uren dat de vrijwilliger aaneengesloten beschikbaar is vanaf

het huidige event (in de flowchart aangegeven door Consecutive Availability) groter of gelijk is aan de tot nu toe beste beschikbaarheid (Best Availability), maar ook gelijk is aan de het aantal halve uren dat de vrijwilliger in totaal nog beschikbaar is vanaf het huidige tijdstip (in de flowchart aangegeven door Get Max Availability). Dit laatste is toegevoegd om te zorgen dat hij zijn korte beschikbaarheid wel helemaal kan benutten op dit podium. Wanneer de lijst met beste podia niet leeg is, zal de vrijwilliger op een podium worden ingepland. Dit gebeurt bij voorkeur op een podium waar op het vorige event al een vrijwilliger is gepland, mocht er geen podium hieraan voldoen zal hij op een random podium worden geplaatst. Dit wordt gedaan omdat er anders al een paar random gaten in de totale planning ontstaan en het lastiger is om daar omheen te plannen dan wanneer het ingeplande blok aaneengesloten is.

De flowchart van het algoritme is te zien in figuur 7.

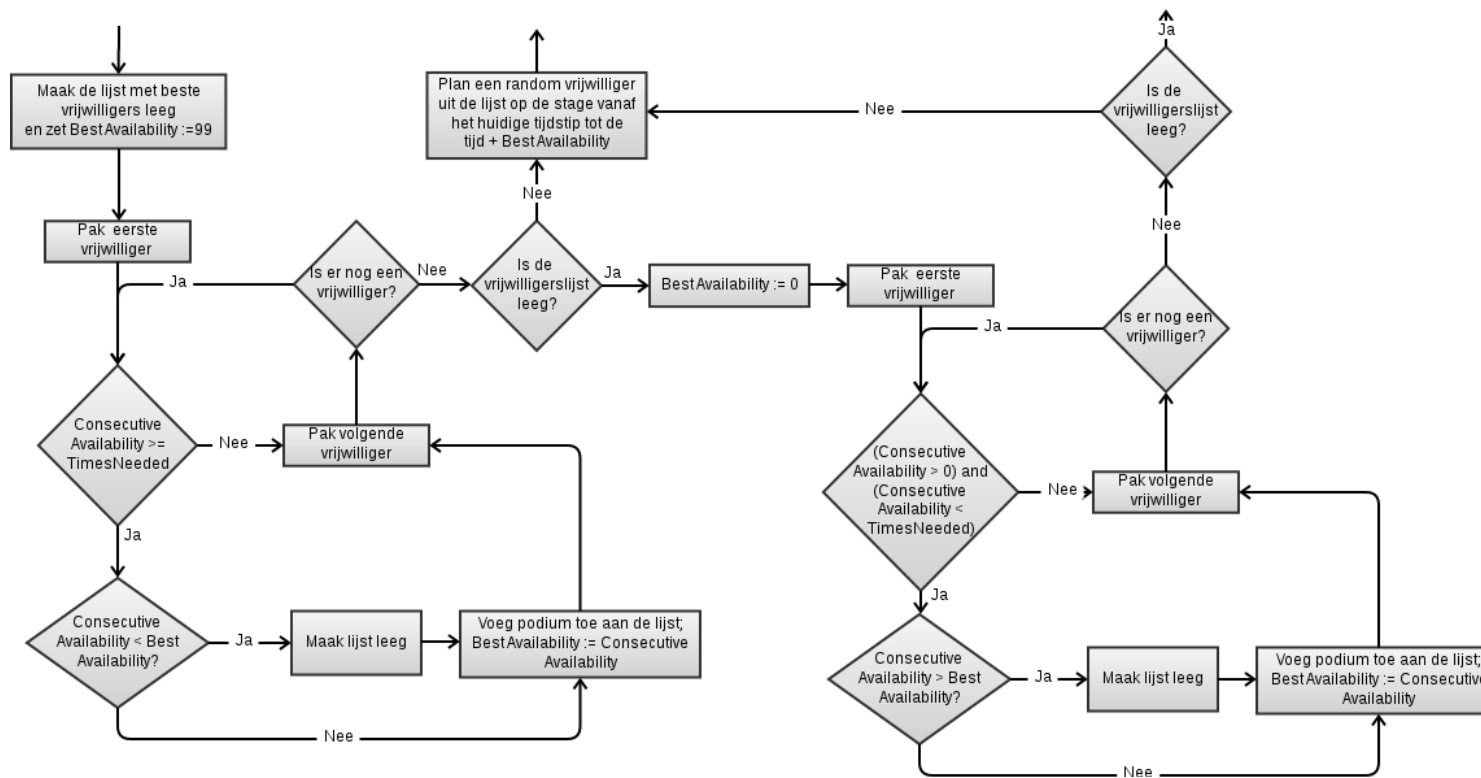


Figuur 7: Plan beperkt beschikbare vrijwilliger.

#### 6.4.4 Plan vrijwilliger die het langst beschikbaar is vanaf het huidige event

Om te bepalen welke vrijwilliger het beste ingepland kan worden vanaf het huidige tijdstip, wordt allereerst de Times Needed berekend. Dit is het aantal halve uren op een podium dat vanaf het huidige event nog ingevuld moet worden. Het algoritme gaat op zoek naar de vrijwilliger die deze tijd het beste kan benaderen met zijn beschikbaarheid. Daarbij geeft hij voorkeur voor mensen die langer kunnen dan de Times Needed ten opzichte van de mensen die een kortere beschikbaarheid hebben. Er wordt wederom een lijst gemaakt van vrijwilligers die als beste de voorwaarden benaderen. Uit deze lijst wordt een random vrijwilliger gekozen om in te plannen vanaf het huidige tijdstip tot aan de tijd + de tot nu toe beste beschikbaarheid. Mocht er niemand gevonden zijn, gaat het Initial Volunteer Algorithm weer verder. Dit gebeurt ook nadat er een vrijwilliger is ingepland.

De flowchart van dit gedeelte van het programma is te vinden in figuur 8.



Figuur 8: Plan vrijwilliger die het langst beschikbaar is vanaf het huidige event.

## 6.5 Improver

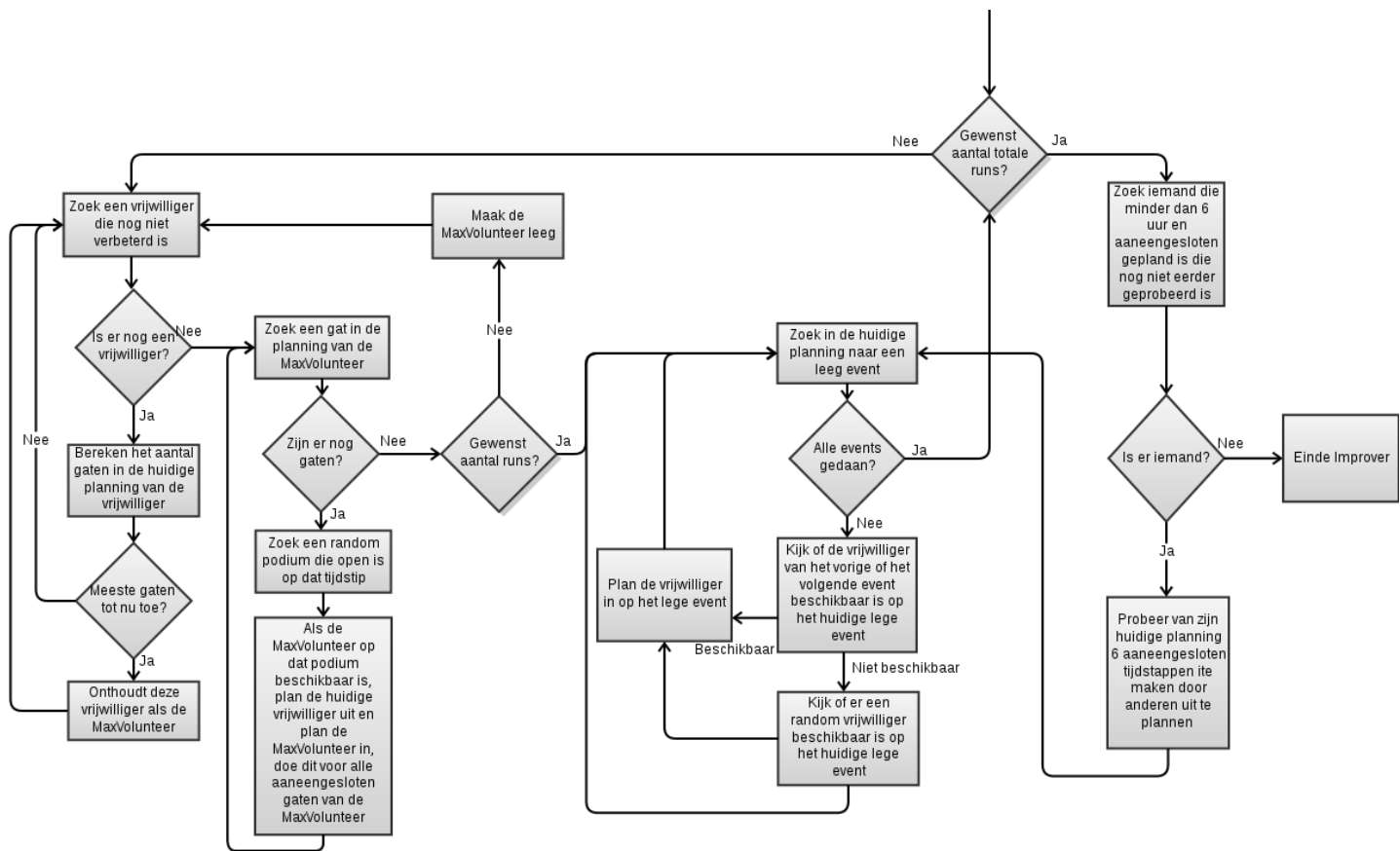
Wanneer de initiële planning gemaakt is wordt de Improver aangeroepen. In het programma staat vooraf vastgesteld hoe vaak dit moet gebeuren.

De Improver heeft als doel om vrijwilligers elkaar te laten vervangen waardoor er bij de andere vrijwilliger tijdstappen vrij komen. Op deze vrijgekomen tijdstap is hij mogelijk beschikbaar op een ander event waar in de planning nog niemand gepland staat. Events waar al iemand ingepland stond, worden op deze manier nooit leeg, aangezien de vrijwilligers elkaar vervangen. Hier onder zal worden toegelicht hoe dit in het programma geïmplementeerd is. In flowchart 9 is dit systematisch weergegeven.

Allereerst wordt er op zoek gegaan naar de vrijwilliger met het hoogste aantal gaten in zijn planning. Hierbij wordt rekening gehouden met overlappen en pauzes, zodat deze niet tellen als een plek waar de vrijwilliger wel ingepland had kunnen worden. Wanneer deze vrijwilliger (MaxVolunteer) bekend is, wordt er gelopen over alle gaten in zijn planning. De gaten in het begin en het eind van het festival worden niet bekeken. Dit komt doordat er op de momenten  $t = 0, 1, 2$  en  $t = 15, 16, 17$  erg veel vrijwilligers beschikbaar zijn en er op die momenten maar één of twee podia open zijn. Hierdoor blijf je telkens dezelfde events vervangen. Als er een gat gevonden is, wordt er een random podium gekozen dat ook open is op de tijdstap van het gevonden gat. Als de MaxVolunteer beschikbaar is volgens alle voorwaarden wordt de vrijwilliger die op die tijdstap op het podium staat, van het event afgehaald en de MaxVolunteer wordt er op ingepland. De MaxVolunteer wordt op dat podium ingepland zo lang hij daar kan staan. Telkens wordt de huidige volunteer van het podium gehaald. Wanneer de MaxVolunteer niet meer op het podium kan staan gaat de Improver op zoek naar andere gaten in zijn planning. Als hij alle gaten heeft bekeken, gaat het proces van het vinden van een nieuwe MaxVolunteer van start, waarna deze wederom zijn gaten gaat vullen door het vervangen van andere vrijwilligers. Dit doet het algoritme totdat hij het gewenste aantal runs heeft behaald.

Het algoritme heeft nu een aantal vrijwilligers verschoven zonder dat er extra gaten zijn gecreëerd in de planning. Het is echter wel mogelijk dat er hierdoor vrijwilligers beschikbaar zijn geworden op andere events waar zij eerder niet ingepland konden worden. Dit ontstaat met name doordat een vrijwilliger een maximum interval ( $m_i$ ) heeft. Wanneer hij 's ochtends wordt ingepland en hij is maar vier uur beschikbaar, dan kan hij een gat om 3 uur 's middags niet meer vullen, terwijl hij daar misschien wel beschikbaar is. Dus wanneer hij om één uur voor het eerst wordt ingepland zal hij het gat om drie waarschijnlijk nog wel kunnen vullen. De Improver kijkt daarom naar alle lege events in de planning en probeert daar eerst de vrijwilliger die op het event er voor of er na staat in te plannen. Mocht dit niet lukken kijkt hij of er een random andere vrijwilliger beschikbaar is om het gat op te vullen. Wanneer hij alle events heeft bekeken, gaat de Improver zolang hij nog niet het gewenste aantal totale runs heeft bereikt, weer op zoek naar een nieuwe MaxVolunteer.





Figuur 9: De Improver.

Wanneer de Improver alle runs heeft gedaan gaat hij kijken of er mensen in totaal minder dan zes halve uren en aansluitend ingepland zijn. Deze mensen mogen namelijk in totaal zes halve uren achter elkaar werken. Als er iemand is die aan de voorwaarden voldoet, mag zijn ingeplande interval aan het begin en aan het eind worden vergroot tot een maximum van zes. Hiervoor wordt de al ingeplande vrijwilliger op het event waar de andere vrijwilliger langer mag uitgepland. Hierdoor zijn er vrije uren ontstaan voor de uitgeplande vrijwilliger. Dus zal de Improver kijken of deze vrijwilliger op een leeg event ingepland mag worden. Het proces van het zoeken naar vrijwilligers die langer gepland mogen worden wordt herhaald tot iedereen gecheckt is. Wanneer dat gedaan is gaat hij terug naar het overkoepelende programma.

## 6.6 Capaciteitscontrole

Om een idee te krijgen of de planning uitvoerbaar is met de beschikbare vrijwilligers, kan men een zogenaamde *capaciteitsrapport* door het programma laten genereren. Dit rapport laat een overzicht van alle benodigde en beschikbare vrijwilligers per tijdstap zien en het aantal benodigde en beschikbare uren tijdens het festival.

### 6.6.1 Per tijdstap

Het aantal benodigde vrijwilligers en het aantal benodigde ervaren vrijwilligers per tijdstap is eenvoudig te bepalen. Voor ieder event is duidelijk vastgelegd hoeveel (ervaren) vrijwilligers hiervoor nodig zijn. Het aantal beschikbare vrijwilligers is daarentegen niet zo eenvoudig te bepalen. Niet alle planningsgebonden restricties kunnen in beschouwing worden genomen, zo is het bijvoorbeeld niet mogelijk om vast te stellen wanneer iemand pauze moet hebben. Mocht de planning al gedeeltelijk zijn gemaakt, dan kunnen planningsgebonden restricties wel (gedeeltelijk) worden gecontroleerd. Zo is het dan mogelijk om te bepalen of deze vrijwilliger op een zeker event zijn maximale beschikbaarheid overschrijd.

### 6.6.2 Voor het gehele festival

Omdat het overzicht van het aantal beschikbare vrijwilligers per uur een vertekend beeld kan opleveren, doordat niet alle aan de planningsgebonden restricties kunnen worden meegenomen, is het handig om voor iedere vrijwilliger zijn maximale inzetbaarheid te bepalen. Het bepalen van de maximale inzetbaarheid voor een vrijwilliger is geen sinecure. De beschikbaarheid van een vrijwilliger kan per podium verschillen. Op een podium kan bijvoorbeeld een koor optreden waarvan de vrijwilliger een afkeur heeft, maar op een ander podium niet. Tevens is het zo, dat als een vrijwilliger optreedt, dat de vrijwilliger direct na zijn eigen optreden enkel op hetzelfde podium als dat hij zijn optreden had, een dienst mag hebben.

De maximale beschikbaarheid wordt in verschillende fasen berekend. Per podium wordt de maximale beschikbaarheid van een vrijwilliger bepaald, dit wordt voor nu gedefinieerd als  $\xi_p$ . Er wordt steeds gekeken hoelang een vrijwilliger achtereenvolgens beschikbaar is. Door voor ieder van deze aaneengesloten perioden van beschikbaarheid het benodigd aantal pauzes en overlappen van af te trekken, wordt er een redelijke benadering van zijn aantal inzetbare tijdstappen verkregen. Op het laatste beschikbare uur van een podium is echter géén overlap nodig. Door

het maximum van alle beschikbaarheid over alle podia te bepalen,  $\max_p(\xi_p)$ , wordt een bovengrens voor de optimale inzet van de vrijwilliger bepaald.

Een vrijwilliger kan minder lang ingepland willen worden dan dat hij als beschikbaarheidsinterval heeft opgegeven. In zo'n geval geldt  $m_i < (e_i + 1) - b_i$ . Door zijn  $m_i$  in vermindering te brengen met het aantal pauzes en overlappen dat minimaal nodig is, wordt zijn totale maximale beschikbaarheid,  $\xi_{\text{totaal}}$ , berekend.

Het minimum van de vrijwilliger zijn totale maximale beschikbaarheid,  $\xi_{\text{totaal}}$ , en zijn maximale beschikbaarheid voor zijn beste podium,  $\max_p(\xi_p)$ , is de uiteindelijke maximale beschikbaarheid van de vrijwilliger. Deze uiteindelijk verkregen waarde is overigens niet meer in het algemeen een bovengrens. Wanneer een vrijwilliger, waarvan de totale maximale beschikbaarheid het grootst van de twee is, op het laatste event van een podium staat, hoeft er namelijk geen overlap plaats te vinden. Doordat het waarschijnlijk is dat meerdere vrijwilliger bij de berekening van  $\max_p(\xi_p)$  een laatste overlap niet wordt meegenomen (op het laatste beschikbare uur van een podium is géén overlap nodig), is de uiteindelijke maximale beschikbaarheid van de vrijwilliger toch een goede 'bovengrens'.

## 7 Resultaten

In deze sectie worden enkele resultaten gepresenteerd. Er zal vaak verwezen worden naar tabellen in de appendix (sectie 11).

De resultaten bestaan uit een programma dat een vrijwilligersplanning kan genereren, een capaciteitenrapport en uit een mogelijkheid om tabellen te creëren bij de data. Tabel 5 en tabel 4 bevatten aangeleverde data van vrijwilligers en podia. Tabel 6 weergeeft een gegenereerde (podium)planning. Tabel 7 weergeeft dezelfde planning, alleen nu gezien vanuit de vrijwilligers. Deze planning is het beste resultaat na 2000 runs van het programma en heeft nog 14 gaten. Er zit bijvoorbeeld nog een gat om 14:30 uur op podium 8. Het blijkt ook (zie tabel 7) dat er op dat moment niemand beschikbaar is<sup>1</sup>. In de resultatenanalyse (deelsectie 8.1) wordt de planning van tabel 6 en tabel 7 geanalyseerd.

De gemiddelde tijd dat een computer erover doet om 1000 runs uit te voeren is ongeveer 2 minuten. Er zijn ook planning gecreëerd die het beste resultaat waren na meer dan 2000 runs. Na 100.000 runs (3,5 uur) had het beste resultaat 12 gaten in de planning. Na 300.000 runs (10,5 uur) had het beste resultaat nog steeds 12 gaten. In totaal is het programma minstens 500.000 keer gedraaid, minder dan 12 gaten hebben we geen enkele keer gehaald.

Tabel 14 geeft een capaciteitenrapport weer. We zien dat het rapport niet direct aangeeft dat er geen planning mogelijk is. Het totaal aan tijdstappen dat de vrijwilligers inzetbaar zijn, is hier 331, meer dan het aantal tijdstappen dat er vrijwilligers nodig zijn, 297. In de resultatenanalyse (deelsectie 8.2), zal dit capaciteitenrapport worden geanalyseerd.

---

1. Ter verduidelijking: vrijwilliger 1 kan bijvoorbeeld niet omdat een dienst om 14:30 uur tot 15:00 uur buiten zijn maximale interval zal vallen, vrijwilliger 7 kan bijvoorbeeld niet omdat 14:30 uur tot 15:00 uur in zijn pauze valt en vrijwilliger 15 kan bijvoorbeeld niet omdat hij overlap op podium 12 moet verzorgen en hij kon bovendien ook al niet omdat hij om 15:00 uur zelf een optreden heeft

## 8 Resultatenanalyse

In deze sectie worden de resultaten die in het vorige hoofdstuk worden genoemd geanalyseerd. Er zal verwezen worden naar tabellen in de appendix (sectie 11).

### 8.1 Analyse van een planning

In deze deelsectie zal een automatisch gegenereerde planning worden geanalyseerd. Het is de planning die al geïntroduceerd is in de resultaten (sectie 7) en verderop in de appendix staat (tabel 6 en tabel 7). Het doel van de analyse is om te laten zien: dat het maken van een verbeteringsalgoritme dat gebaseerd is op het schuiven van vrijwilligers erg lastig is, dat de huidige korenplanning enigszins ongunstig is en dat over het algemeen vrijwilligers goed zijn ingepland in de automatisch gegenereerde planningsen.

In tabel 6 zien we dat er een gat in de planning is om 14:30 uur<sup>2</sup> op podium 8. Wanneer er een gat in de planning zit, dan is er sowieso geen vrijwilliger beschikbaar. Een verbeteringsalgoritme zou vrijwilligers kunnen ‘pakken’ die om dit gat heen zijn gepland en vervolgens kan hun dienst worden verschoven. Vrijwilliger 41 kan sowieso niet om 14:30 uur, want hij heeft een kooroptreden om 15:30 uur (zie tabel 7). Vrijwilliger 30 kan wel opgeschoven worden, echter moet dan zijn dienst op podium 15 worden verschoven of ingekort. Er zal een gat op podium 15 ontstaan om 13:30 uur. Er is geen vrijwilliger beschikbaar om dit gat op te vullen. Er zal dus geprobeerd worden met vrijwilliger 31 te schuiven, wat elders weer een gat zal creëren. In veel gevallen ontstaat er een dergelijke situatie. Een ander (uitgebreider) verbeteringsalgoritme zou een andere vrijwilliger kunnen pakken die ingepland is op een podium waarop een gat in de planning zit. We keren terug naar het gat op podium 8 om 14:30 uur. Vrijwilliger 2 is ook ingepland op podium 8, zijn dienst gaan we verschuiven zodat zijn laatste dienst op het gat om 14:30 uur valt. Om 11:00 uur en 11:30 uur ontstaan nu twee lege plekken. We zien dat vrijwilliger 35 deze plekken kan opvullen. In dit geval blijkt een dergelijk verbeteringsalgoritme goed te werken. Wanneer we echter naar het gat op podium 12 kijken om 14:30 uur<sup>3</sup> dan wordt het verbeteringsalgoritme uitgebreider, omdat er erg veel mogelijke vrijwilligers zijn waaruit gekozen kan worden. Hier bestaat er geen verschuiving zodat er een gat ontstaat wat een andere vrijwilliger kan opvullen. Het is hier dan wederom de vraag in hoeverre je ‘doorgaat’ met verschuiven in een planning waarin al diensten zijn verschoven. Opgemerkt moet ook worden dat alle verschuivingen ook bewaard moeten blijven, ze moeten immers ongedaan gemaakt kunnen worden. Een dergelijk verbeteringsalgoritme is erg complex van aard, terwijl het nog maar de vraag is of het werkt.

Het blijkt ook dat de huidige korenplanning niet heel gunstig is. We kijken wederom naar tabel 6 en tabel 7. Hier zien we dat vrijwilliger 35 van zijn vier beschikbare tijdstappen er maar twee ingezet zou kunnen worden. Iets dergelijks geldt voor vrijwilliger 34, 41 en 48. Bovendien zijn veel mensen die een optreden hebben om 12:30 uur (7 vrijwilligers) en veel die een optreden hebben om 15:00 uur (6 vrijwilligers). Dit zou een reden kunnen zijn dat er ’s middags een tekort is aan vrijwilligers (want alle gaten zitten ’s middags). Hoe de vrijwilligersplanning eruit komt te zien

---

2. Met een gat om 14:30 uur wordt een gat op het interval 14:30 tot 15:00 bedoeld.

3. Dit is een gat want er moeten 2 vrijwilligers staan.

wanneer er een andere korenplanning is hangt van diverse factoren af. Het zou eventueel met een andere korenplanning nog moeilijker kunnen zijn om een planning te maken.

Ondanks dat sommige vrijwilligers maar heel beperkt beschikbaar zijn, zijn de vrijwilligers over het algemeen redelijk goed ingepland. Van de eerste tien vrijwilligers zou je kunnen zeggen dat vrijwilliger 3,4 en 10 niet 'mooi' zijn ingepland. Deze zijn respectievelijk 16,17 en 14 uur beschikbaar. Met name door de pauze's van een uur rondom hun hoofddienst (respectievelijk op podium 16,19 en 14) zijn de vrijwilligers een stuk minder beschikbaar. Dan blijkt dat ze 's ochtends of later op de middag nog capaciteit 'over' hebben. Echter is er op deze tijdstippen geen tekort aan vrijwilligers. Deze plekken zijn namelijk voornamelijk gevuld met vrijwilligers die beperkt beschikbaar zijn en daar wel ingepland moeten worden, wil je gebruik kunnen maken van deze vrijwilligers. Door vrijwilligers met een beperkte beschikbaarheid eerst in te plannen en daarna pas andere vrijwilligers (zoals ons algoritme doet), zijn vrijwilligers met een lange beschikbaarheid over het algemeen slechter ingepland. Het is echter maar de vraag of dit efficiënter kan, aangezien er anders niet gebruik van beperkt beschikbare vrijwilligers wordt gemaakt.

## 8.2 Analyse van een capaciteitenrapport

In deze deelsectie zal een capaciteitenrapport worden geanalyseerd. Het is het rapport dat al geïntroduceerd is in de resultaten (sectie 7) en verderop in de appendix staat (tabel 14).

We zien dat het rapport niet direct aangeeft dat er geen planning mogelijk is. Het totaal aan tijdstappen dat de vrijwilligers inzetbaar zijn, hier gelijk aan 331, is meer dan het aantal tijdstappen dat er vrijwilligers nodig zijn, 297. Uit het rapport ontstaat al direct het vermoeden dat dit beschikbare aantal tijdstappen lang niet allemaal inzetbaar zijn, aangezien er op de eerste drie tijdstappen veel meer vrijwilligers beschikbaar zijn dan nodig.

In totaal lijken er  $331 - 297 = 34$  halve uren meer beschikbaar te zijn dan dat er nodig is. Wanneer we kijken naar de eerste drie tijdstappen zien, we dat dit aantal al een groot deel afneemt. Op de eerste twee tijdstappen is maar één persoon nodig terwijl er op de eerste tijdstap 13 mensen beschikbaar zijn en op de tweede tijdstap 21 mensen. Op de eerste tijdstap zijn sowieso alle mensen beschikbaar die de gehele dag kunnen worden ingepland (wanneer er geen afkeur voor het podium is). Hiervan kan maximaal één iemand worden ingepland, de overige halve uren raakt men dus 'kwijt'. Op het tweede uur kunnen twee mensen nodig zijn: één iemand die ingepland is en één iemand die het eerste tijdstap werkte en daarna stopt en dus een overlap heeft. Hiermee raak je de overige mensen die compleet beschikbaar zijn op deze tijdstap ook kwijt.

In de gegeven situatie kunnen we wat extra conclusies trekken omtrent het aantal halve uren wat meer beschikbaar is dan dat er nodig is. Doordat er was geconstateerd dat vrijwilligers zo goed mogelijk gebruik moesten maken van hun beschikbaarheid zullen, in het algoritme wat door het programma gebruikt wordt, de vrijwilligers met de kleinste beschikbaarheid als eerste worden gepland. Deze zullen dan ook direct voor zo lang als ze beschikbaar zijn worden ingepland. In de data (zie tabel 4 in de Appendix) is te zien dat persoon 13 en persoon 44 beiden maar kort

beschikbaar en ervaren zijn. Hierdoor zal het algoritme altijd één van deze personen inplannen op de eerste twee of meer tijdstappen.

Er zijn in totaal zes vrijwilligers die de gehele dag beschikbaar zijn. Van deze personen weet je met zekerheid dat ze niet worden ingepland op de eerste twee tijdstappen. Hierdoor raken er al twaalf van de  $331 - 297 = 34$  tijdstappen kwijt. Ook zijn er twee vrijwilligers die 16 van de 17 tijdstappen beschikbaar zijn. Bij deze vrijwilligers zal dus per persoon in ieder geval één tijdstap kwijtraken, omdat ze niet op dat uur worden ingepland. Bovenop de eerder genoemde twaalf halve uren, raakt de planning dus nóg twee tijdstappen kwijt.

Op de derde tijdstap zijn er negen mensen nodig, waarvan acht ervaren. Dat betekent dat er maar één (of twee in verband met overlap) vrijwilliger nodig is om dat bijbehorende event, waar een onervaren vrijwilliger mag staan, op te vullen. Op het derde tijdstap zijn er tien onervaren vrijwilligers beschikbaar waarvan dus maximaal twee mensen nodig. Hierdoor raak je dus acht tijdstappen aan beschikbaarheid kwijt. Door het gebruikte algoritme en de aangeleverde data, weten we dat de eerste vier uur op het podium dat om 9:30 uur opengaat, sowieso wordt gevuld door een ervaren vrijwilliger (13 of 44). Op het ene overgebleven halve uur op het derde tijdstap waar geen ervaren vrijwilliger nodig is, staat dus één van die twee vrijwilligers. Hiermee komen we op 10 halve uren van de onervaren vrijwilligers die niet worden gebruikt. In totaal blijven er dus  $34 - 12 - 10 = 12$  tijdstappen over die meer beschikbaar zijn dan dat er nodig zijn.

Ook lijkt er op geen enkele tijdstap een tekort aan vrijwilligers te zijn. Op de tijdstappen 15:00–15:30 uur en 16:00–16:30 uur is het verschil tussen het aantal beschikbare vrijwilligers en het aantal nodige vrijwilligers het kleinst. Men zou wellicht verwachten het meeste gaten zich manifesteren op deze tijdstippen. In tabel 6 valt juist te zien dat de meeste gaten juist op de tijdstappen 14:30–15:00 uur en 16:00–16:30 uur zitten. Op deze tijdstappen hebben erg veel vrijwilligers pauze. Dit kan mogelijk zijn ontstaan doordat er van vroeg naar laat wordt gepland, waarbij bij het inplannen niet in de toekomst wordt gekeken of het plannen van de vrijwilliger zorgt voor het ontstaan van een gat in de toekomst.

## 9 Discussie

In deze sectie worden de resultaten besproken. Er zal besproken worden wat er onjuist is aan de resultaten, hoe dit verbeterd kan worden, wat de verschillen zijn met een handmatige planning door de organisatie van Amusing Hengelo en in hoeverre onze resultaten bruikbaar zijn.

### 9.1 Redenen dat automatische planning niet voldoet en verbeteringen

Zoals besproken in de sectie resultaten (sectie 7) voldoet de gemaakte planning niet, er blijven gaten zitten in de planning. Deze gaten verminderen (op een gegeven moment) ook niet meer door het aantal runs flink toe te laten nemen. De reden dat de planning niet voldoet zou misschien gezocht kunnen worden in het algoritme. Echter bij een analyse (deelsectie 8.1) van een resultaat (en ook bij andere resultaten), blijkt dat vrijwilligers, over het algemeen, goed zijn ingepland. Hier en daar valt eventueel nog een gat opgevuld te kunnen worden, maar het is onwaarschijnlijk dat met onze ‘beste’ plannings een planning gemaakt kan worden zonder gaten. Mede hierdoor wordt eraan getwijfeld of het überhaupt mogelijk is om een planning te maken zonder gaten. Dit wordt bevestigd door het capaciteitsrapport, die ook aangeeft dat de planning erg krap is. Bovendien blijkt uit een analyse (deelsectie 8.1), dat de gebruikte korenplanning niet heel gunstig is voor de vrijwilligersplanning. Het zou mogelijk kunnen zijn dat met een andere korenplanning, de vrijwilligersplanning wel voldoet.

Momenteel is het echter niet mogelijk om de korenplanning en vrijwilligersplanning na elkaar te draaien, zodat met de (net) gegenereerde korenplanning een vrijwilligersplanning gegenereerd kan worden. Dit is een cruciaal punt dat nog aangepast moet worden. De reden dat dit momenteel niet kan, is dat in de vrijwilligersplanning meer events zijn dan in de korenplanning. In principe is het goed mogelijk om het aantal events in de korenplanning uit te bereiden. Dit blijft een verbeteringspunt aan het programma.

Een tweede verbeteringspunt heeft betrekking op het aantal runs van de Improver. Momenteel loopt de Improver een vaststaand aantal keer. Het is beter om de Improver te laten lopen zolang er een verbetering blijft optreden. Op deze manier kun je ‘echt’ convergeren naar een lokaal minimum. Als  $M_j$  het aantal gaten zijn van de vrijwilliger met het maximaal aantal gaten, in ronde  $j$ , kan de Improver blijven lopen zolang  $M_{j+1} \leq M_j + s$  ( $s \in \mathbb{N}$ ). De vaste variabele  $s$  kan vooraf worden bepaald, en zorgt ervoor dat het mogelijk wordt om in een groter gebied naar een lokaal minimum te convergeren. Het is ook nog mogelijk om met een vrijwilliger die net iets minder gaten heeft, dan de vrijwilliger met het maximale aantal gaten, de Improver te doorlopen.

Er zijn nog een aantal verbeteringspunten aan het programma. Dit betreft zaken die niet (juist) zijn geïmplementeerd, maar niet direct de reden zijn waarom de planning niet voldoet. Het derde verbeteringspunt zijn de soft constraints. Het uiteindelijke resultaat houdt geen rekening met de soft constraints. Het zou dus kunnen dat een vrijwilliger niet bij een podium komt, waarvoor hij een voorkeur heeft. De kosten van een vrijwilliger, bepaald aan de hand van de soft constraints, kunnen wel bepaald worden in het definitieve programma. Er is echter geen programma aanwezig dat gebruik maakt van deze kosten. Dit zou bijvoorbeeld kunnen met een tweede verbeteringsalgoritme, dat erop gericht is om de kosten te minimaliseren.



	830	9:30 - 10:00	10:00 - 10:30	10:30 - 11:00	11:00 - 11:30	11:30 - 12:00	12:00 - 12:30	12:30 - 13:00	13:00 - 13:30	13:30 - 14:00
Yvonne Koopman (6)	0	- N/A -	- N/A -	- N/A -	EnschedeeseStraa	EnschedeeseStraa	EnschedeeseStraa	EnschedeeseStraa	EnschedeeseStraa	EnschedeeseStraa
Ria ter Braake (5)	0	Rabotheater Ealc	Rabotheater Ealc	Rabotheater Ealc	Rabotheater Ealc	Rabotheater Ealc	Rabotheater Ealc	- N/A -	- N/A -	- N/A -

Figuur 10: Fouten met overlap.

Dit tweede verbeteringsalgoritme kan dezelfde opzet hebben als het eerste verbeteringsalgoritme (zie deelsectie 6.5). Dit houdt in dat een vrijwilliger met hoge kosten wordt ‘gepakt’ en vervolgens kan hij anders ingepland worden, zodanig dat zijn planning meer aan zijn wensen voldoet. Het zou echter wel kunnen dat op deze manier het aantal gaten in de planning weer zal stijgen. Het goed implementeren van soft constrains blijft een verbeteringspunt aan het programma.

Een vierde verbeteringspunt, dat niet het aantal gaten zal doen verminderen, is dat vrijwilligers in de geautomatiseerde planning eigenlijk te lang worden ingezet. Het zou kunnen dat de allerlaatste overlap van een vrijwilliger over de maximale tijd gaat dat een vrijwilliger ingezet wil worden of in een tijdstap valt dat de vrijwilliger überhaupt niet beschikbaar is. Zie figuur 10 voor een voorbeeld. Bij de tweede vrijwilliger zien we dat de overlap, die na de dienst komt, in een tijdstap valt waarin de vrijwilliger niet beschikbaar is. Bij de eerste vrijwilliger zien we dat de totale dienst, plus de overlap langer duren dan de maximale tijd dat een vrijwilliger ingezet wil worden (de vrijwilliger was beschikbaar om 14:00–14:30 uur). Om dit punt op te lossen zal er een halfuur van vrijwilligers, zoals in figuur 10, afgehaald moeten worden. Dit is echter in het geval dat events een halfuur duren.

Een vijfde verbeteringspunt, iets dat niet geïmplementeerd is, is de mogelijkheid om mensen vast te zetten in de planning. Met deze mogelijkheid, die handmatig gedaan zal moeten worden voordat de automatische planning wordt gestart, kunnen moeilijk planbare vrijwilligers ingepland worden. Echter plant het ‘Initial Volunteer Algoritme’ (deelsectie 6.4) ook al eerst moeilijk inplanbare mensen in. Met deze mogelijkheid wordt het doorvoeren van veranderingen aan de wensen/eisen van een vrijwilliger wel vereenvoudigd. Op dit moment zal hiervoor de XML-data aangepast moeten worden.

Een zesde punt wat niet helemaal optimaal is aan het programma, is dat events een half uur duren. Het veranderen van events naar een kwartier, zal waarschijnlijk niet direct een verbetering opleveren aan de huidige resultaten. Als echter de vierde verbetering doorgevoerd wordt (zie hierboven), dan hoeft er eigenlijk maar een kwartier afgehaald te worden van de dienst. In dit opzicht hebben events van een kwartier voordeel ten opzichte van events van een halfuur. Events van een kwartier hebben ook voordeel als gekeken wordt naar de handmatige planning van de organisatie van Amusing Hengelo (zie deelsectie 9.2). Het blijkt dat hierin niet altijd een reistijd wordt opgenomen in de vrijwilligersplanning. In de huidige automatische planning wordt dit wel altijd gedaan, terwijl dit misschien niet zou moeten. Op deze manier kan een kwartier worden ‘gewonnen’, door vrijwilligers in te plannen bij podia waar-tussen geen reistijd is. Echter vereist dit wel een verandering aan de data, er moet namelijk toegevoegd worden tussen welke podia geen reistijd is. Bovendien kan het veranderen van events naar een kwartier een uitgebreide klus worden, dit omdat onder andere de korenplanning werkt met events van een half uur.

voornaam	tussen	achternaam	10.30	11	11.30	12	12.30	13	13.30	14	14.30	15	15.30	16	16.30	Welke uitzondering?
Wim	van der	Geest	F	F	F	F	18	18	18	14	14	14	14	F	F	3,5 uur ingepland voor podiumdienst.
Mirjam		Ottens	V	V	V	V	V	V	/15	15	15	15	15	15	15	2,75 uur ingepland tussen 13:30 en 16:30
Isis		Zengerink	9	9	9/	/6	6	6/	4	4	4	4	4/			3,5 uur ingepland tussen 12:15 en 15:45.
Jans		Renken	/3	3	3	>	8	8	8	8/	9	9	9	>	>	3,5 uur ingepland tussen 12:30 en 16:00.
Karin		Franken	>	/19	19	>	BV-20	BV-20	BV-20	BV-20	BV-20	/16	16	16	16	4,5 uur ingepland tussen 13:30 en 17:00.

Figuur 11: Uitzonderingen met pauzes.

PODIUM 12 Lambertusbasiliek			
TIJD	KOOR	vrijwilliger	boekjesverkoop
14:00 - 14:30	Overijssels Byzantijns koor	Klaas van Till	Netty Bakker <b>vanaf 13.45</b>
15:00 - 15:30	Christelijk Mannenkoor Hengelo	Theo Morskate	x
15:30 - 16:00	Opus '94	Theo Morskate	x
16:00 - 16:30	Voices to Heaven	Theo Morskate	x
16:30 - 17:00	Euphonion Singers	Theo Morskate	Celine Morskate
PODIUM 7 Brinkplein			
TIJD	KOOR	vrijwilliger	boekjesverkoop
12:00 - 12:30	MIX Zenderen	x	Henk van der Bijl
12:30 - 13:00	Popkoor Focus	x	Henk van der Bijl <b>tot 13.15</b>
13:00 - 13:30	MATCH Haaksbergen	Annemarie Rosbergen	x
13:30 - 14:00	Popkoor PUUR!	Maaïke de Jonge	x

Figuur 12: Uitzonderingen bij podia met twee vrijwilligers.

## 9.2 Verschillen met handmatige planning

Een van de grootste verschillen tussen de geautomatiseerde planning en de handmatige planning, is dat in de geautomatiseerde planning gewerkt wordt met events van een half uur en in de handmatige planning gewerkt wordt met events van een kwartier. In de vorige paragraaf is besproken waarom dit nadelig is voor de geautomatiseerde planning.

Een tweede verschil tussen de geautomatiseerde en handmatige planning is dat in een automatische planning hard constraints ook daadwerkelijk hard zijn. In een handmatige planning kunnen met hard constraints nog wel eens wat soepeler worden omgegaan. Enkele voorbeelden vinden we terug in de handmatige planning van Amusing Hengelo 2011. Hier blijkt de planning ook niet altijd te voldoen aan de eisen gesteld in de probleembeschrijving (sectie 4). Het is niet bekend of dit uitzonderingen op 'de regel' zijn of dat dit fouten in de handmatige planning zijn. Ten eerste zien we bijvoorbeeld terug dat een vrijwilliger niet consistent pauze heeft na 2,5 uur, en dat een dienst nog wel eens kan uitlopen, zie figuur 11. Dit zijn wellicht afspraken tussen de organisatie en de vrijwilliger. In het programma is het op dit moment niet mogelijk om na een variabele tijd pauze te hebben. Wanneer er veel mensen langer dan 2,5 uur aaneengesloten zouden willen werken, dan zou dit allicht verbeterd kunnen worden in het programma.

In de handmatige planning zien we ten tweede terug dat op een podium waar twee vrijwilligers aanwezig moeten zijn, dit niet altijd het geval is, zie figuur 12. Waarom dit in de handmatige planning is gedaan, is niet bekend. Het aantal benodigde vrijwilligers terug naar 1 brengen is iets wat momenteel niet in het programma kan. Dit is wel iets wat met niet al te veel moeite gedaan kan worden, en iets wat een aardige verbetering kan bewerkstelligen aan de resultaten.

In de handmatige planning zien we als laatste ook nog eens 'fouten' terug in de

PODIUM 9 Hoek Nieuwstraat / B.P. Hofstedestraat (Vodafone)				Welke uitzondering?
Tijd	KOOR	vrijwilliger	overlapping	
12:30 - 13:00	Vocalgroup Creative		Netty Bakker	
13:00 - 13:30	Oktet Raskovec	Jannie Haanstra <b>vanaf 13.15</b>	Netty Bakker <b>tot 13.15</b>	geen overlap
13:30 - 14:00	VoixSix	Jannie Haanstra		
14:00 - 14:30	Voices in Harmony	Jannie Haanstra <b>tot 14.15</b>		geen vrijwilliger van 14:15 tot 14:30
14:30 - 15:00	Glazzy		Jans Renken	

PODIUM 13 Hoek Enschedesestraat / Brinkstraat (Twee Wezen)				Welke uitzondering?
Tijd	KOOR	vrijwilliger	overlapping	
13:00 - 13:30	La Sido	Tanja Plumers	Aquine Zondag <b>tot 13.15</b>	
13:30 - 14:00	Oekraïense zanggroep Roesalka	Tanja Plumers		
14:00 - 14:30	Vrouwenkoor Colours	Arend Kremer		geen overlap
14:30 - 15:00	A la Violette	Arend Kremer		

Figuur 13: Uitzonderingen met overlap.

overlap tussen vrijwilligers, zie figuur 13. Overlap is iets wat in de automatische planning veel tijd kost. Bovendien zien we niet altijd terug dat een vrijwilliger genoeg reistijd krijgt in de handmatige planning. Het kan zijn dat deze reistijd niet nodig is, omdat twee podia dicht bij elkaar staan. Toch zal er in het programma altijd een kwartier reistijd aangehouden worden. Wanneer reistijd niet altijd nodig is dan kan de overlap, die nu samen met reistijd een halfuur duurt, ingekort worden naar een kwartier. In dit geval zal het dan gunstig zijn om events van een kwartier te gebruiken. Een tweede fout die betrekking heeft op de overlap, zien we terug op een podium waar twee vrijwilligers een dienst moeten draaien. Het blijkt dat het mogelijk is dat één vrijwilliger zorgt voor de overlap. In ons programma wordt ervoor gezorgd dat er altijd twee vrijwilligers bij zo'n podium zijn, dus ook dat twee vrijwilligers voor overlap zorgen.

Al deze verschillen tussen de handmatige en automatische planning kunnen de reden zijn dat de geautomatiseerde planning niet kan voldoen. In het geval dat de eisen gesteld in de de probleembeschrijving (sectie 4) niet kloppen, of dat hier uitzonderingen op zijn, dan zal dit in het programma aangepast moeten worden. In het geval dat de eisen in de probleembeschrijving wel kloppen, maar dat er in de handmatige planning fouten staan, dan is het allicht mogelijk om de geautomatiseerde planning te gebruiken. Er kan een automatische planning gemaakt worden en vervolgens kan er wat vrijer worden omgegaan met alle eisen.

### 9.3 Bruikbaarheid van automatische planning

De huidige versie van het programma is niet in staat een kant-en-klare planning te genereren. In dit opzicht is het programma dus niet bruikbaar. Wel kan de gegenereerde planning, met gaten, dienen als opstapje voor een handmatige planning. Het is geprobeerd om dit te doen, echter bleek dit, in een relatief kort tijdsbestek, niet mogelijk. Het is echter wel mogelijk als er wat soepeler wordt omgegaan met alle eisen aan de planning. Er staan dan wel 'fouten' in de planning, maar zoals beschreven in de vorige deelsectie staan deze 'fouten' er ook in bij een met de hand gegenereerde planning.

Het programma is verder ook bruikbaar om een overzicht te krijgen over alle data. De ingelezen data kan in een tabel weergegeven worden, wat de overzichtelijkheid bevordert.

Het capaciteitsrapport is sowieso bruikbaar. Wanneer de capaciteitscheck 'zegt' dat het niet mogelijk is om een planning te maken, dan zal dit ook (bijna) nooit kunnen.

Het is dus aan te raden om de capaciteitscheck een keer uit te voeren, alvorens er gestart wordt met de handmatige of automatische planning. Een minpunt aan de capaciteitscheck is dat het zou kunnen dat het niet mogelijk is om een planning te maken, maar dat de capaciteitscheck niet ‘zegt’ dat het niet kan. Dit komt omdat je, van te voren, niet precies weet hoeveel pauze en overlap een vrijwilliger precies nodig heeft. Het is daarom erg lastig een goede capaciteitscheck te maken.

## 10 Aanbevelingen

Deze sectie beschrijft de aanbeveling aan de opdrachtgevers, de organisatie van Amusing Hengelo.

Zoals beschreven in de sectie Resultaten (sectie 7), is het niet gelukt een programma te ontwikkelen dat een planning genereert zonder gaten bij de huidige data. In de discussie (sectie 9) is beschreven waarom dit zo is, en worden er enkele verbeteringspunten aangekaart. Bovendien wordt in de deelsectie 9.3 beschreven in hoeverre het programma bruikbaar is. De aanbevelingen aan de organisatie van Amusing Hengelo zijn dan ook hierop gebaseerd.

Momenteel is het programma niet in staat om een kant-en-klare vrijwilligersplanning te leveren. Toch is het niet zo dat het programma totaal niet bruikbaar is, aangezien met een aantal (kleine) aanpassingen het programma mogelijk wel de gewenste planning kan genereren. De belangrijkste aanpassing is dat de programma's van de korenplanning en van de vrijwilligersplanning nog 'in elkaar verweven' moeten worden. Momenteel werkt de vrijwilligersplanning met een onveranderde en enigszins ongunstige korenplanning.

Bovendien is het advies om nog eens naar de eisen van de vrijwilligersplanning te kijken, en waar nodig deze te herzien of uitzonderingen door te voeren (die eventueel gelden voor enkele vrijwilligers). De eisen van de planning staan beschreven in de probleembeschrijving (sectie 4). De eisen zijn in sommige gevallen vrij streng. Bovendien is geconstateerd dat de handmatige planning ook niet helemaal overeenkomt met de gestelde eisen. Er moet er rekening mee worden gehouden dat een programma harde eisen altijd laat gelden, iets waar in een handmatige planning wel eens wat makkelijker mee omgegaan kan worden.

In het geval dat bovenstaande punten niet uitgevoerd/veranderd kunnen worden, dan kan het programma nog steeds bruikbaar zijn. Het programma kan namelijk dienen als 'opstapje' voor de handmatige planning. Er kan geprobeerd worden om gaten in de automatisch gegenereerde planning, handmatig op te vullen. Dit klinkt vrij makkelijk uitvoerbaar, in de praktijk blijkt dit echter erg lastig. Het kan wel aanzienlijk makkelijker gemaakt worden door wat soepeler om te gaan met alle eisen aan de planning. Op deze manier kunnen diensten bijvoorbeeld iets langer gemaakt worden, wat eigenlijk niet mag. Wat ook kan is om een deel van de vrijwilligers automatisch in te plannen en een deel met de hand. In dit geval is het advies om vrijwilligers met veel eisen, een korte dienst en/of vrijwilligers met veel kooptredens automatisch in te plannen en vervolgens met 'makkelijker' vrijwilligers, handmatig, de gaten op te vullen. Deze aanpak vereist (momenteel) wel een aanpassing aan de ingevoerde data. De capaciteitscheck in het programma is wel altijd bruikbaar. Er kan gecontroleerd worden of het überhaupt mogelijk is om een planning te maken. Bovendien kan het programma tabellen creëren van de ingelezen data, dit zal de overzichtelijkheid bevorderen.

Concreet is het advies om allereerst te proberen om de verbeteringspunten door te voeren. Mocht de vrijwilligersplanning daarna nog niet bruikbaar zijn dan zouden de eisen aan de planning eventueel bijgesteld kunnen worden. Momenteel is het programma enkel bruikbaar als 'opstapje' voor de handmatige planning en kan er gebruik gemaakt worden van de capaciteitscheck.

## 11 Appendix

### 11.1 Gebruikte data

<i>Vrijwilliger</i>	<i>Beschikbaar vanaf</i>	<i>Beschikbaar tot</i>	$m_i$	<i>Ervaring</i>	<i>Samen met</i>	<i>Podiumvoorkeur</i>	<i>Podiumafkeur</i>	<i>Koorvoorkeur</i>	<i>Koorafkeur</i>
1	11:00	20:00	6	nee					
2	08:00	15:00	12	ja					
3	08:00	20:00	16	ja					
4	08:00	20:00	17	ja					
5	10:00	16:00	10	ja					
6	10:00	18:00	12	ja					
7	12:00	20:00	16	ja					
8	11:00	18:00	14	ja					
9	13:00	17:00	8	ja					
10	08:00	18:00	14	ja					
11	11:00	14:00	6	nee					
12	10:00	14:00	8	ja					
13	08:00	12:00	8	ja					
14	08:00	20:00	12	ja					
15	08:00	20:00	17	ja	16		21		
16	10:00	20:00	10	ja	15		21		
17	11:00	16:00	4	ja					
18	10:00	17:00	14	ja					
19	09:00	17:00	8	nee					
20	09:00	14:00	10	ja					
21	08:00	20:00	17	ja					
22	10:00	17:00	2	nee					
23	11:00	18:00	14	ja		4			
24	10:00	17:00	10	ja					
25	08:00	20:00	17	nee				146	
26	12:00	17:00	10	ja					
27	08:00	20:00	12	nee					
28	10:00	18:00	10	ja					
29	12:00	16:00	8	ja					
30	08:00	17:00	8	nee					
31	08:00	20:00	17	ja					
32	09:00	13:00	8	nee					
33	08:00	20:00	17	nee					
34	11:00	13:00	4	ja	42				
35	09:00	13:00	8	ja	47				
36	10:00	18:00	12	nee					
37	13:00	18:00	10	nee					
38	15:00	18:00	6	nee					

Vervolg op volgende pagina

<i>Vrijwilliger</i>	<i>Beschikbaar vanaf</i>	<i>Beschikbaar tot</i>	$n_i$	<i>Ervaring</i>	<i>Samen met</i>	<i>Podiumvoorkeur</i>	<i>Podiumafkeur</i>	<i>Koorvoorkeur</i>	<i>Koorafkeur</i>
39	14:00	17:00	6	ja					
40	12:00	16:00	8	nee					
41	12:00	16:00	8	nee					
42	11:00	13:00	4	ja	34				
43	10:00	15:00	10	ja					
44	08:00	12:00	8	ja					
45	08:00	20:00	16	nee					
46	09:00	13:00	8	nee					
47	09:00	13:00	8	ja	35				
48	14:00	17:00	4	ja					
49	14:00	17:00	4	ja					
50	14:00	17:00	4	nee					
51	09:00	12:00	6	nee					

Tabel 4: Overzicht van de beschikbaarheid en voorkeuren van de vrijwilligers.

Podium	Naam	Locatie
1	Rabothheater Eaton-Holec zaal	binnen
2	Rabothheater Eijnsinkzaal (kleine zaal)	binnen
3	Rabothheater Boendersbar/Hal (Beursstraat-zijde)	binnen
4	Schouwburgplein bij het Rabothheater	buiten
5	Hoek Molenstraat/Nieuwstraat	buiten
6	Brinkplein	buiten
7	Historisch Museum Hengelo, Beekstraat	binnen
8	Bibliotheek leescafé De Ontdekking	binnen
9	hoek Nieuwstraat/BP Hofstedestraat (Vodafone)	buiten
10	Hoek Enschedesestraat/Nieuwstraat (De Appel)	buiten
11	Enschedeesestraat t.o. de HEMA	buiten
12	Lambertusbasiliek, Enschedesestraat	binnen
13	Hoek Enschedesestraat/Brinkstraat (Twee Wezen/Purdey)	buiten
14	Weemenstraat (t.h.v. achterkant Lambertuspassage)	binnen
15	Thiemsbrug Winkelcentrum	binnen
16	Stadhuishal	binnen
17	Burg. Jansenplein, cirkel voor het stadhuis	buiten
18	Waterstaatskerk (NFVE), Deldenerstraat	binnen
19	Pastoriestraat (halverwege)	buiten
20	Marktstraat bij Duthler	buiten

Tabel 5: Overzicht van alle podia.



## 11.2 Voorbeeld van een gegeneerde vrijwilligersplanning

Podium	9:30-10:00	10:00-10:30	10:30-11:00	11:00-11:30	11:30-12:00	12:00-12:30	12:30-13:00	13:00-13:30	13:30-14:00	14:00-14:30	14:30-15:00	15:00-15:30	15:30-16:00	16:00-16:30	16:30-17:00	17:00-17:30	17:30-18:00
1	13	13	13	47	47	47	47	39	39	39	39	27	27	27	27	N/A	N/A
2	N/A	N/A	N/A	28	28	28	28	19	19	19	19	43	43	4	4	N/A	N/A
3	N/A	N/A	N/A	31	31	31	31	7	7	7	14	14	14	14	14	N/A	N/A
4	N/A	N/A	N/A	35; 32	35; 32	42; 32	42; 32	42; 9	42; 44	50; 44	50; 44	50; 8	50; 8	16; 8	16; 8	16	16
5	N/A	N/A	4	22	22	43	43	43	43	28	28	28	28	39	39	39	N/A
6	N/A	N/A	15	15; 16	15; 16	16; 38	38; 25	38; 25	38; 25	25; 23	23; 9	23; 9	23; 9	9; 10	9; 10	N/A	N/A
7	N/A	N/A	N/A	N/A	14	14	26	26	26	26	33	33	33	33	N/A	N/A	N/A
8	N/A	N/A	N/A	2	2	2	2	2	2	41		30	30	N/A	N/A	N/A	N/A
9	N/A	N/A	10	51	51	46	46	46	46	21	21	21	21	19	19	23	N/A
10	N/A	N/A	20	11	11	11	11	11	11	49	49	49	49		3	3	N/A
11	N/A	N/A	44	44	44	44	29	29	29	29		38	38	38	38	38	N/A
12	N/A	N/A	N/A	23; 3	23; 33	23; 33	23; 33	33; 12	15; 16	15; 16	16	46	46; 26	46; 26	46; 26	N/A	N/A
13	N/A	N/A	24	1	1	1	1	1	1	48		40	40	40	40	40	N/A
14	N/A	N/A	N/A	21	21	21	21	10	10	10	10		7	7	7	N/A	N/A
15	N/A	N/A	N/A	18	18	30	30	30	30	31	31	31	31	18	18	N/A	N/A
16	N/A	N/A	N/A	N/A	12	12	24	24	3	3	3	3	41	41	N/A	N/A	N/A
17	N/A	N/A	45	45	45	27	27	27	27	5	5	5	5			N/A	N/A
18	N/A	N/A	36	36; 37	36; 37	36; 37	37; 8	8; 20	8; 20	6; 17	6; 24	6; 24	6; 29	6	N/A	N/A	N/A
19	N/A	N/A	N/A	6	6	6	6	4	4	4	4	42	42			N/A	N/A
20	N/A	N/A	N/A	5	5	5	5	14	18	18			25	25	N/A	N/A	N/A

Tabel 6: Rooster per podium met 14 gaten.

<i>Vrijwilliger</i>	<i>9:30-10:00</i>	<i>10:00-10:30</i>	<i>10:30-11:00</i>	<i>11:00-11:30</i>	<i>11:30-12:00</i>	<i>12:00-12:30</i>	<i>12:30-13:00</i>	<i>13:00-13:30</i>	<i>13:30-14:00</i>	<i>14:00-14:30</i>	<i>14:30-15:00</i>	<i>15:00-15:30</i>	<i>15:30-16:00</i>	<i>16:00-16:30</i>	<i>16:30-17:00</i>	<i>17:00-17:30</i>	<i>17:30-18:00</i>
1	N/A	N/A	N/A	13	13	13	13	13	13						(*)2		
2				8	8	8	8	8	8			N/A	N/A	N/A	N/A	N/A	N/A
3				12					16	16	16	16			10	10	
4			5					19	19	19	19			2	2		
5	N/A			20	20	20	20			17	17	17	17	N/A	N/A	N/A	N/A
6	N/A			19	19	19	19			18	18	18	18	18			
7	N/A	N/A	N/A	N/A	N/A			3	3	3			14	14	14		
8	N/A	N/A	N/A				18	18	18			4	4	4	4		
9	N/A	N/A	N/A	N/A	N/A	N/A	N/A	4			6	6	6	6	6	N/A	N/A
10			9					14	14	14	14			6	6		
11	N/A	N/A	N/A	10	10	10	10	10	10	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
12	N/A				16	16		12		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
13	1	1	1			N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
14					7	7		20			3	3	3	3	3		
15	N/A		6	6	6		(*)20		12	12		(*)18					
16				6	6	6			12	12	12			4	4	4	4
17	N/A	N/A	N/A				(*)10		18			(*)6		N/A	N/A	N/A	N/A
18	N/A			15	15		(*)10		20	20		(*)6		15	15	N/A	N/A
19				(*)10			(*)2	2	2	2	2			9	9	N/A	N/A
20			10					18	18	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
21				14	14	14	14			9	9	9	9				
22	N/A			5	5		(*)10					(*)6				N/A	N/A
23	N/A	N/A	N/A	12	12	12	12			6	6	6	6			9	

Vervolg op volgende pagina

<i>Vrijwilliger</i>	<i>9:30-10:00</i>	<i>10:00-10:30</i>	<i>10:30-11:00</i>	<i>11:00-11:30</i>	<i>11:30-12:00</i>	<i>12:00-12:30</i>	<i>12:30-13:00</i>	<i>13:00-13:30</i>	<i>13:30-14:00</i>	<i>14:00-14:30</i>	<i>14:30-15:00</i>	<i>15:00-15:30</i>	<i>15:30-16:00</i>	<i>16:00-16:30</i>	<i>16:30-17:00</i>	<i>17:00-17:30</i>	<i>17:30-18:00</i>
24	N/A		13				16	16		(*)18	18	18			(*)12	N/A	N/A
25							6	6	6	6			20	20			
26	N/A	N/A	N/A	N/A	N/A		7	7	7	7			12	12	12	N/A	N/A
27						17	17	17	17			1	1	1	1		
28	N/A			2	2	2	2			5	5	5	5				
29	N/A	N/A	N/A	N/A	N/A		11	11	11	11			18	N/A	N/A	N/A	N/A
30						15	15	15	15			8	8			N/A	N/A
31				3	3	3	3			15	15	15	15				
32				4	4	4	4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
33					12	12	12	12			7	7	7	7			
34	N/A	N/A	N/A				(*)10	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
35	N/A	N/A	N/A	4	4		(*)10	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
36			18	18	18	18		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
37				18	18	18	18	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
38	N/A					6	6	6	6			11	11	11	11	11	
39	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1	1	1	1			5	5	5	
40	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	13	13	13	13	13	
41	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8		(*)16	16	16		N/A	N/A
42	N/A	N/A	N/A	N/A	N/A	4	4	4	4			19	19	N/A	N/A	N/A	N/A
43	N/A	N/A	N/A	N/A	N/A	5	5	5	5			2	2	N/A	N/A	N/A	N/A
44	N/A		11	11	11	11			4	4	4	N/A	N/A	N/A	N/A	N/A	N/A
45			17	17	17	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
46						9	9	9	9			12	12	12	12		
47				1	1	1	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Vervolg op volgende pagina

<i>Vrijwilliger</i>	<i>9:30-10:00</i>	<i>10:00-10:30</i>	<i>10:30-11:00</i>	<i>11:00-11:30</i>	<i>11:30-12:00</i>	<i>12:00-12:30</i>	<i>12:30-13:00</i>	<i>13:00-13:30</i>	<i>13:30-14:00</i>	<i>14:00-14:30</i>	<i>14:30-15:00</i>	<i>15:00-15:30</i>	<i>15:30-16:00</i>	<i>16:00-16:30</i>	<i>16:30-17:00</i>	<i>17:00-17:30</i>	<i>17:30-18:00</i>
48	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	13		(*)14		(*)20		N/A	N/A
49	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	10	10	10	10			N/A	N/A
50	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	4	4	4	4			N/A	N/A
51				9	9	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Tabel 7: Rooster voor alle vrijwilligers; (\*) = kooroptreden op aangegeven podium.

### 11.3 Voorbeeld van een capaciteitenrapport

tijdstap	aantal vrijwilligers		aantal ervaren	
	beschikbaar	nodig	beschikbaar	nodig
9:30–10:00	13	1	13	1
10:00–10:30	32	1	21	0
10:30–11:00	31	9	21	8
11:00–11:30	38	22	26	9
11:30–12:00	39	24	26	2
12:00–12:30	34	24	22	0
12:30–13:00	34	24	22	0
13:00–13:30	35	24	23	0
13:30–14:00	36	24	23	0
14:00–14:30	37	24	24	0
14:30–15:00	31	24	19	0
15:00–15:30	30	24	17	0
15:30–16:00	34	24	20	0
16:00–16:30	29	23	18	0
16:30–17:00	29	18	18	0
17:00–17:30	20	6	13	0
17:30–18:00	21	1	13	0

297 tijdstappen aan vrijwilligers nodig, 331 tijdstappen beschikbaar.

Figuur 14: Voorbeeld van een capaciteitenrapport.

### Referenties

- [1] Enno Boersma en Astrid Geerts. „Planning voor Vocaal Festival Amusing Hengelo”. Bacheloropdracht. Universiteit Twente.