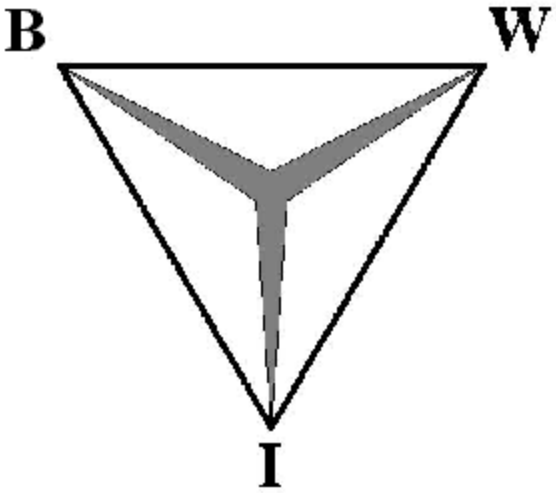
Printerwachtrijen op de VU
Een Simulatiestudie



Dennis van 't Klooster

BWI-werkstuk

November 2002

VOORWOORD

Dit verslag is het resultaat van het onderzoek naar de printerwachtrijen op de VU, waarbij het onderzoek is gefocust op de wachtrij voor de Océ printer. De printer die verre weg het meest gebruikt wordt. Dit onderzoek is gedaan ten behoeve van het BWI-werkstuk een verplicht onderdeel van de vierjarige studie Bedrijfswiskunde en Informatica aan de Vrije Universiteit te Amsterdam.

Het onderzoek naar de printerwachtrijen op de VU bestaat uit een simulatiestudie. Via simulatie is het mogelijk relatief snel verschillende alternatieven voor behandeling van de printerwachtrij te genereren en de resultaten daarvan te bekijken zonder toepassing van wiskundige formules. Op deze manier staat mijn onderzoek haaks op die van M. Sweijd, die eveneens onderzoek deed naar de wachtrijen op de VU voor zijn BWI-werkstuk (Sweijd [2]). Sweijd [2] gaat in zijn verslag in op het wachtrijmodel aan de hand van de bestaande wachttijdtheorie. Dit verslag echter, laat dit wachtrijmodel compleet links liggen en probeert aan de hand van simulaties van verschillende alternatieven voor behandeling van de printerwachtrij te bepalen hoe de wachttijd voor de inmiddels nieuwe Océ printer verkort kan worden. Verder wordt er voor dit onderzoek ook geen onderzoek gedaan naar de verschillende verdelingen voor onder andere aankomstintensiteit en de duur van de printopdrachten, maar wordt de simulatie volledige gebaseerd op historische data.

Bij deze wil ik mijn begeleider dr. G.J. Franx bedanken voor zijn begeleiding en ondersteuning tijdens het onderzoek. Verder wil ik de heer R. Wiggers en de heer G. Huisman van systeembeheer bedanken voor het aanleveren van de data en het uitleggen van de werking van de Océ printer.

Dennis van ‘t Klooster

November 2002

INHOUD

1 INLEIDING

In dit hoofdstuk wordt kort een inleiding gegeven over het onderzoek. In deze inleiding gaan we in op de context en de probleembeschrijving van het onderzoek. Daarna worden enkele regels uit de wachttijdtheorie behandeld om vervolgens aan de hand van deze regels de te onderzoeken scenario's weer te geven. Tenslotte geven we uitleg over de event grafen, die model staan voor de verschillende simulatieprogramma's benodigd voor het onderzoek.

1.1 Context van het onderzoek

Zoals al eerder beschreven is dit onderzoek gesitueerd bij de nieuwe Océ printer op de VU. Sinds oktober/november 2001 is de oude Océ printer (van toepassing op het onderzoek van Sweijd[2]) vervangen door een nieuwe printer. Deze printer is sneller dan de oude, dus het is sowieso interessant een onderzoek te doen naar de nieuwe prestaties van deze printer. Bij dit onderzoek laten we de wiskundige wachttijdtheorie volledig links liggen en richten we ons puur op de resultaten die we verkrijgen via simulatie. We gebruiken alleen enkele algemeen geldende regels uit de wachttijdtheorie om ons te focussen op een aantal interessante scenario's. Bij het bedenken van deze scenario's hebben we ons gerealiseerd dat de Océ printer inmiddels is vergezeld van een identieke printer en daarom hebben we het onderzoek uitgebreid met twee printer scenario's.

Bij de printer komen printopdrachten (printjobs) binnen, die volgens First In, First Out (FIFO) afgehandeld worden. Het doel van het onderzoek is om te kijken of dit wel een slimme manier is. Dit onderzoek is volledig gebaseerd op historische data. De aankomstmomenten, het aantal bytes, het aantal pagina's en de duren van de printopdrachten zijn dan ook precies zoals ze in de praktijk hebben plaatsgevonden. We hebben het onderzoek naar de verschillende verdelingen van onder andere het aankomstproces en de bedieningstijd bewust achterwege gelaten, omdat we voldoende data hebben om een complete simulatie te doen aan de hand van deze historische data. Deze historische data bevat tenslotte de werkelijkheid, op het moment dat we namelijk een verdelingsonderzoek doen moeten we al gauw aannames gaan maken en lopen we dus het risico de werkelijkheid uit het oog te verliezen.

Op het moment dat we praten over de wachttijd van een printopdracht dan bedoelen we trouwens de wachttijd in de rij en niet de totale verblijftijd in het systeem, waarbij een printopdracht het systeem binnenkomt op het moment dat de printopdracht werkelijk gegeven wordt en het systeem pas weer verlaat op het moment dat het laatste blaadje de printer verlaten heeft.

De data bestaat uit gegevens voor 16 dagen, waarbij een dag loopt van 22:30 de vorige dag tot 22:30, op de dag zelf. Printopdrachten komen gedurende deze hele periode binnen, alhoewel het er 's avonds natuurlijk een stuk minder zijn dan over dag. De printer gaat echter pas rond een uur of 8:45 aan, dus de printer begint met het verwerken van printopdrachten die 's nachts zijn binnengekomen. De data van de dagen komt uit november en begin december 2001, over het algemeen een drukke periode wat betreft printopdrachten in verband met tentamens en deadlines van practica. Volgens ons dus een goede periode om als simulatiedata te gebruiken. Het zijn namelijk de drukke dagen waarop we te maken hebben met de lange wachttijden die we graag zouden verkorten.

1.2 Probleembeschrijving

Het doel van het onderzoek is om te kijken of we de gemiddelde wachttijd voor de printer niet kunnen verkorten door de wachtrij voor de printer anders af te handelen dan het FIFO principe zoals dat nu gebeurt. De manier van afhandelen van een wachtrij wordt rijdiscipline genoemd. Dit gaan we niet proberen te berekenen, maar doen we aan de hand van simulaties. Voor deze simulaties kijken we niet alleen naar de gemiddelde wachttijd, maar ook naar de servicegraad van de printer. De servicegraad geeft aan welk percentage van het totaal aantal printopdrachten binnen een bepaalde tijd (we hebben gekozen voor 30 seconden als een gewenste wachttijd) afgehandeld kan worden.

We zijn voornamelijk geïnteresseerd naar deze prestatiematen en daarom is simulatie ook zo interessant. Via simulatie is het namelijk mogelijk relatief snel verschillende alternatieven voor behandeling van de printerwachtrij te genereren en de resultaten daarvan te bekijken zonder toepassing van wiskundige formules. Simulatie is een veel gebruikte methode voor de analyse van wachttijdproblemen die nauw aan moeten sluiten bij de realiteit (Tijms [3]).

1.3 Wachttijdtheorie

Uit de wachttijdtheorie komen twee algemeen geldende regels naar voren die mogelijk kunnen leiden tot verbetering van bovengenoemde prestatiematen. Hieronder volgt een korte samenvatting van de wachttijdtheorie en de twee algemeen geldende regels, die toepasbaar zijn op ons probleem (Tijms [3]).

Basis elementen van een wachttijdmodel:

- **Het aankomstproces** geeft aan wanneer printopdrachten binnen komen. Wat is de hoeveelheid tijd tussen twee opeenvolgende printopdrachten? Voor simulaties wordt vaak gebruik gemaakt van een Poisson-proces verdeling
- **Het bedieningsmechanisme** geeft aan of we te maken hebben met een of meerdere printers, of deze printers een gezamenlijke wachtrij hebben of voor elke printer een aparte wachtrij
- **De rijdiscipline** geeft aan in welke volgorde de printopdrachten worden uitgevoerd. Krijgen bepaalde printopdrachten voorrang boven andere opdrachten, etc?

In ons geval is beschouwen we het aankomstproces als gegeven en proberen we door aanpassing van het bedieningsmechanisme en de rijdiscipline te kijken of we de wachttijd kunnen verkorten. Deze aanpassingen sluiten aan bij de volgende algemeen geldende regels:

1. De eerste algemeen geldende regel is dat wanneer voorrang gegeven wordt aan de kortste taken, de gemiddelde wachttijd over alle taken daalt.
2. De tweede algemeen geldende regel is dat één gezamenlijke wachtrij voor alle printers beter is dan een aparte rij voor elke printer afzonderlijk.

Verder is het zo dat des te hoger de belasting van het systeem is des te groter het effect van de bovenstaande twee regels zal zijn.

1.4 Verschillende Scenario's

Aan de hand van de bovenstaande wachttijdtheorie, hebben we enkele scenario's bedacht om te testen of en hoe we de gemiddelde wachttijd voor de printer kunnen verminderen en de servicegraad van de printer kunnen verbeteren. Paragraaf 1.4.1 geeft veranderingen in de rijdiscipline aan, terwijl in paragraaf 1.4.2 wordt gekeken naar veranderingen in het bedieningsmechanisme.

1.4.1 Sorteren op verschillende criteria van een Printopdracht

De verschillende criteria zijn:

- **Aankomst:** de printopdracht die als eerste aankomt wordt als eerste geholpen, de huidige situatie van de printerwachtrij.
- **Bytes, Aankomst:** de printopdracht met het kleinste aantal bytes wordt als eerste geholpen, is het aantal bytes gelijk dan is de aankomst bepalend.
- **Pagina's, Aankomst:** de printopdracht met het kleinste aantal pagina's wordt als eerste geholpen, is het aantal pagina's gelijk dan is de aankomst bepalend.
- **Seconden, Aankomst:** de printopdracht met het kleinste aantal seconden voor de printduur wordt als eerste geholpen, is het aantal seconden gelijk dan is de aankomst bepalend.
- **Splitsen in kleine en grote Printopdrachten, Aankomst:** we splitsen de printopdrachten in kleine en grote printopdrachten aan de hand van de verschillende criteria van een printopdracht, dus het aantal bytes, het aantal pagina's en het aantal seconden, waarbij we verschillende waarden als grens kiezen voor deze aantallen. We creëren in feite twee groepen, waarbij de groep met kleine printopdrachten voorrang krijgt boven de groep met grote printopdrachten, binnen deze groepen is de aankomst weer bepalend.

Voor het zoeken naar een geschikte oplossing is het goed te beseffen dat de aankomst en het aantal bytes van een printopdracht altijd bekend zijn in het systeem. Het aantal pagina's kan uit de file van de betreffende printopdracht gelezen worden, maar dit kost natuurlijk wel enige tijd en moeite. Het aantal seconden is in werkelijkheid nooit van tevoren bekend, toch is het interessant om ook de resultaten hiervan mee te nemen omdat dit een soort bovengrens geeft van wat we maximaal zouden kunnen bereiken.

1.4.2 Eén of twee Printers, één of twee Rijen, verschillende Snelheden

We beginnen het onderzoek met één printer om het vervolgens uit te breiden naar twee printers. Bij twee printers hebben we de mogelijkheid om elke printer een eigen wachtrij te geven of één gezamenlijke wachtrij te gebruiken, waarbij de volgende printopdracht aan de eerste de beste vrije printer wordt gegeven. In eerste instantie kiezen we ervoor om de twee printers op halve snelheid van een printer te stellen later gaan hier we nog wat mee experimenteren, maar de snelheid voor de twee printers blijft identiek en gaan we niet afzonderlijk variëren.

1.5 Simulatieprogramma's en Event Grafen

Alvorens we starten met het tonen van de resultaten, lijkt het ons een goed plan even kort uit te leggen hoe simulatieprogramma's en de daarbij behorende event grafen precies werken.

Een simulatieprogramma is in feite niet veel meer dan een model van een werkelijk systeem, waarbij de veranderingen in de toestand van het model gesimuleerd kunnen worden. Het simulatieprogramma is in staat de toestanden van het systeem vast te leggen en te analyseren. Wijzigingen in de toestand worden 'events' genoemd. De toestand wijzigt zich slechts op discrete tijdstippen. Een simulatieprogramma doet niet anders dan het doorlopen van een lijst van geschedulde events, waarbij elk event weer leidt tot een mogelijk nieuw event (zie hier de uitleg over de event grafen).

Voordat men start met het schrijven van een simulatieprogramma is het goed om het model vast te leggen in een event graaf. Een event graaf is een representatie van het simulatiemodel door middel van een gerichte graaf waarin elke knoop een event voorstelt en een kant van knoop A naar knoop B tot uitdrukking brengt dat na optreden van event A eventueel event B kan optreden (Nobel [1]). De tekst tussen ronde haakjes bij de kant tussen A en B betekent dat aan die voorwaarde voldaan moet zijn wil event B optreden. Als aan deze voorwaarde voldaan is, dan geeft de tekst *'t = ...'* bij een kant tussen A en B aan na hoeveel tijd event B zal optreden. Verder geeft de tekst tussen accolades bij de knopen aan welke code in ieder geval uitgevoerd moet worden op het moment dat zo'n event optreedt en de tekst tussen vierkante haken geeft aan waarop een event betrekking heeft (bijvoorbeeld op welke printopdracht of rij?).

Hierboven staat in het kort beschreven hoe een event graaf gelezen moet worden, waarbij een event graaf niet een chronologische volgorde van stappen voorstelt. Op het moment dat naar aanleiding van een bepaald event een nieuw event gescheduled wordt hoeft dit niet te betekenen dat dit event de eerstvolgende event is die plaatsvindt. Het kan namelijk zo zijn dat er al geschedulde events zijn die op een eerder tijdstip plaatsvinden en dus ook eerst uitgevoerd moeten worden. De lijst van events die het simulatieprogramma doorwerkt is gesorteerd op de tijdstippen dat een event plaatsvindt (voor meer details zie Nobel [1]). We zullen nu kort de verschillende event grafen van de verschillende simulatiemodellen uiteen zetten.

1.5.1 Event Graaf één Printer, één Rij

Zoals de titel al aangeeft hebben we hier te maken met één printer en één rij, voor dit model zijn vier variabelen en vier events gedefinieerd (zie Figuur 1.1). Allereerst de variabelen:

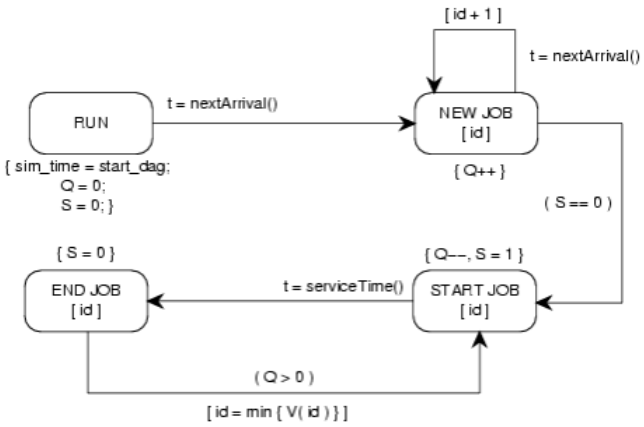
- **sim_time:** geeft de al gepasseerde simulatietijd aan. Op welk simulatie tijdstip van de eerstvolgende event plaats?
- **Q:** geeft de lengte van de wachtrij aan.
- **S:** geeft aan of de printer vrij (0) is, of bezet (1).
- **id:** geeft aan met welke printopdracht we te maken hebben.

Tenslotte de verschillende events

- **RUN:** elk simulatieprogramma moet ergens beginnen, de eerste NEW JOB event wordt gescheduled, de wachtrij is leeg en de printer is vrij.
- **NEW JOB:** de event dat er een nieuwe printopdracht binnenkomt, de printopdracht wordt op de juiste plaats (afhankelijk van de gekozen rijdiscipline) toegevoegd aan de wachtrij, een nieuwe NEW JOB event wordt gescheduled (na elke aankomst is er altijd de eerstvolgende

aankomst, behalve wanneer het einde van de dag bereikt is natuurlijk) en wanneer de printer niks te doen heeft wordt er meteen een START JOB event gescheduled (de printopdracht wordt meteen weer verwijderd uit de wachtrij, hij was blijkbaar de enige).

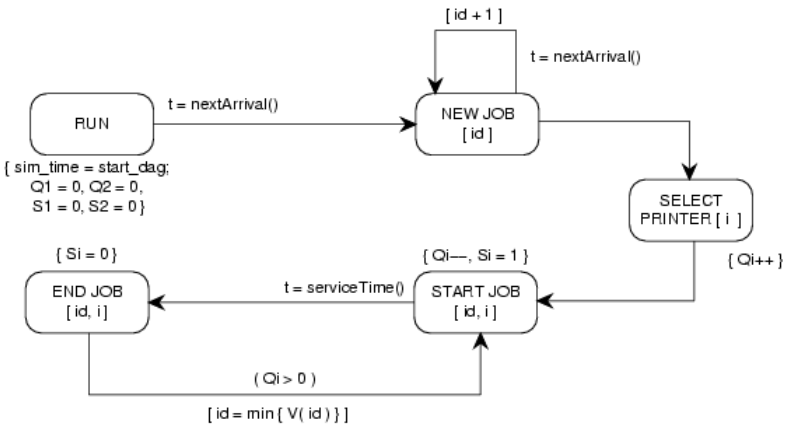
- **START JOB:** op het moment dat de printer vrij is kan er begonnen worden met printen, de eerstvolgende printopdracht wordt uit de wachtrij gehaald en de printer is bezet. Het einde van de printopdracht wordt gescheduled.
- **END JOB:** de printopdracht is afgelopen, de printer is weer vrij. Als de wachtrij niet leeg is kan begonnen worden met printen van de volgende printopdracht uit de wachtrij. Wat de volgende printopdracht is hangt natuurlijk af van de rijdiscipline die dan geldt (min{V(id)}). Het kan natuurlijk zo zijn dat nadat een printopdracht is afgelopen er geen nieuwe opdrachten in de wachtrij bevinden er wordt dan direct naar de eerstvolgende NEW JOB event gesprongen (zolang het einde van de dag niet bereikt is, is er altijd een eerstvolgende NEW JOB event).



Figuur 1.1: één printer, één rij

1.5.2 Event Graaf twee Printers, twee Rijen

De event graaf van dit model lijkt erg op de bovenstaande event graaf we hebben hier echter te maken met twee printers en twee rijen en daarvoor hebben we het een en ander moeten toevoegen. We hebben nu niet variabele Q, maar variabele Q1 en Q2 voor respectievelijk wachtrij 1 en wachtrij 2 en niet variabele S, maar variabele S1 en S2. Verder hebben we een extra event toegevoegd die bepaalt voor welke printer de binnengekomen printopdracht nou precies is. Deze verdeling is natuurlijk afhankelijk van de gekozen verdeling (bijvoorbeeld een student-staf of kleine-grote printopdrachten verdeling gebaseerd op de bekende criteria).

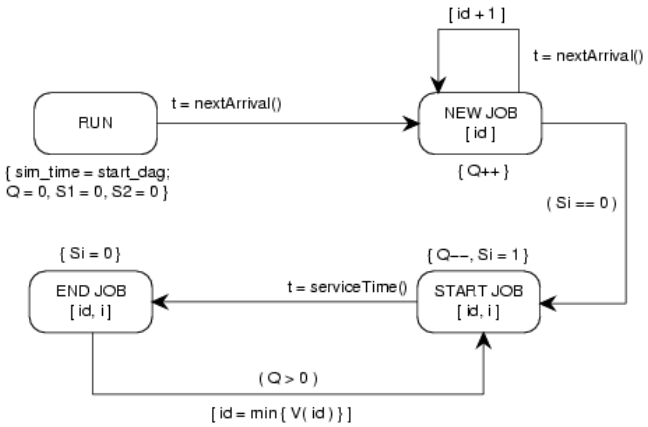


Figuur 1.2: twee printers, twee rijen

1.5.3 Event Graaf twee Printers, één Rij

Ook deze event graaf lijkt erg op de eerst genoemde event graaf, met dat verschil dat we nu ook weer voor de variabele S de variabele S1 en S2 hebben ingevoerd. Verder is het zo dat wanneer beide printers vrij zijn een willekeurige printer gekozen wordt om op te beginnen, beide printers zijn toch identiek qua snelheid.

We hebben geen onderzoek gedaan naar de gevolgen van het printen met twee printers van verschillende snelheid. Dit zou slechts tot enkele willekeurige resultaten leiden en pas interessant worden op het moment dat de snelheden van toepassing zijn op de praktijksituatie.



Figuur 1.3: twee printers, één rij

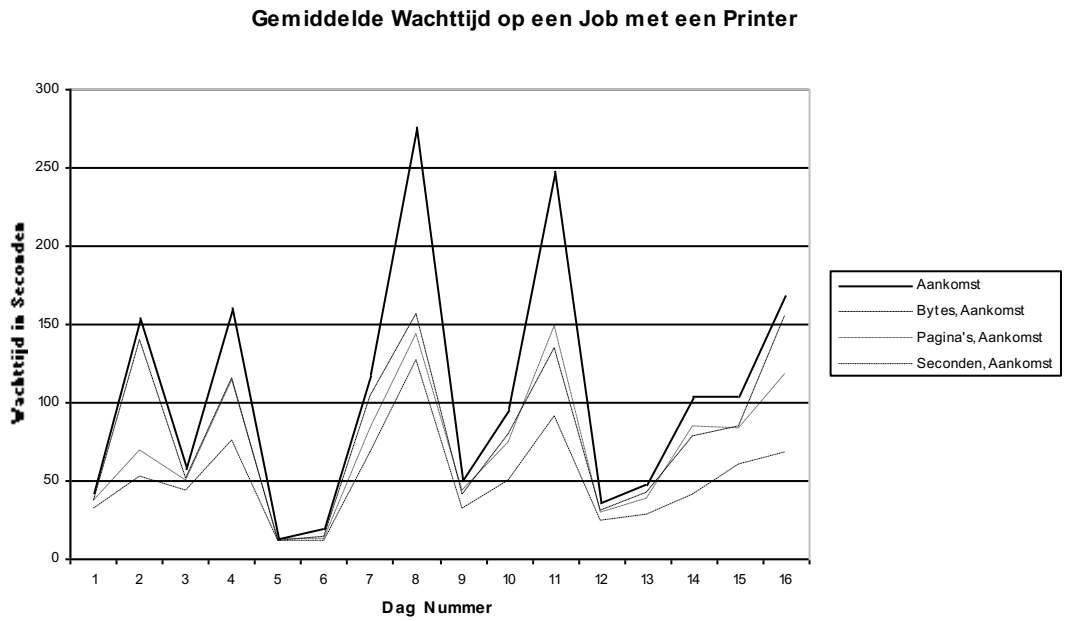
2 RESULTATEN ÉÉN PRINTER, ÉÉN RIJ

In dit hoofdstuk behandelen we enkele resultaten gevonden voor het één-printer probleem. Er zijn natuurlijk veel meer resultaten deze hebben echter niet allemaal evenveel toegevoegde waarde. Zo geven de grafieken over de gemiddelde wachttijd en de 30 seconden service graad steeds hetzelfde beeld en heeft het dus geen zin om beide weer te geven. Om de grafieken goed te kunnen vergelijken hebben we gekozen om in de hoofdstukken met resultaten alleen de gemiddelde wachttijd grafieken op te nemen, de servicegraad grafieken bevinden zich in een appendix. De grafieken waarbij we een bepaalde waarde als grens gekozen hebben zijn al enigszins geoptimaliseerd, dat wil zeggen dat we alleen die grenzen laten zien die het beste beeld geven van alle geteste grenzen. Deze grenzen hebben we via 'trial and error' proberen te optimaliseren.

2.1 Vergelijking van verschillende Prioriteitscriteria

In Figuur 2.1 kan men goed zien dat de rijdiscipline sorteren op puur alleen de aankomst verre weg het slechtste resultaat oplevert en sorteren op seconden en daarna pas op aankomst juist het beste. Daartussenin vallen dan het aantal bytes en het aantal pagina's, waarbij sorteren op het aantal pagina's over het algemeen net even beter scoort.

Hieruit kunnen we concluderen dat de huidige prioriteitscriteria niet het beste resultaat opleveren, alhoewel dit uit praktisch oogpunt wel de meest logisch is. Verder kunnen we concluderen dat hoe beter we van tevoren weten hoe lang een printopdracht gaat duren des te korter de gemiddelde wachttijd zal zijn, waarbij dus het aantal pagina's een net even betere voorspeller is voor de lengte van de printduur dan het aantal bytes. Het verschil is echter niet groot en de vraag is of het de moeite is om het aantal pagina's voor de start van een printopdracht te achterhalen, terwijl we het aantal bytes al weten.

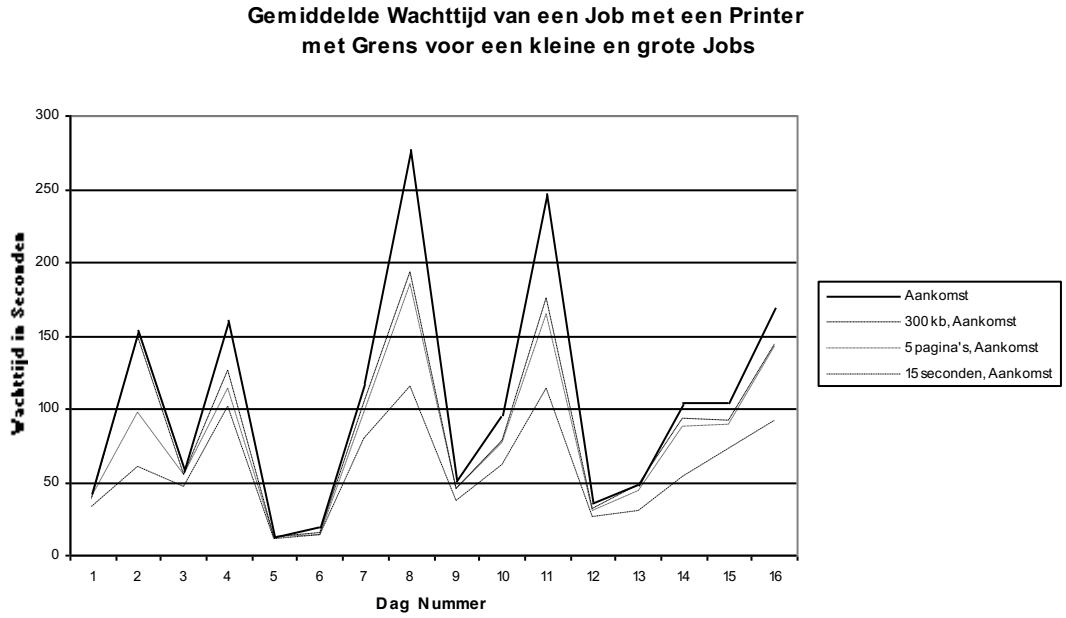


Figuur 2.1

Kijken we naar de dagen afzonderlijk, dan zien we dat er toch nog wel veel verschil zit tussen de verschillende dagen. Dit is interessant, omdat we nu goed kunnen zien dat we juist op de wat slechtere dagen goed kunnen profiteren van een andere prioriteitscriterium dan de huidige. Kijken we naar de absolute waarden dan zien we dat die in het gunstigste geval slechts 10 à 20 seconden wachttijd heeft, maar kan oplopen tot ruim vier minuten.

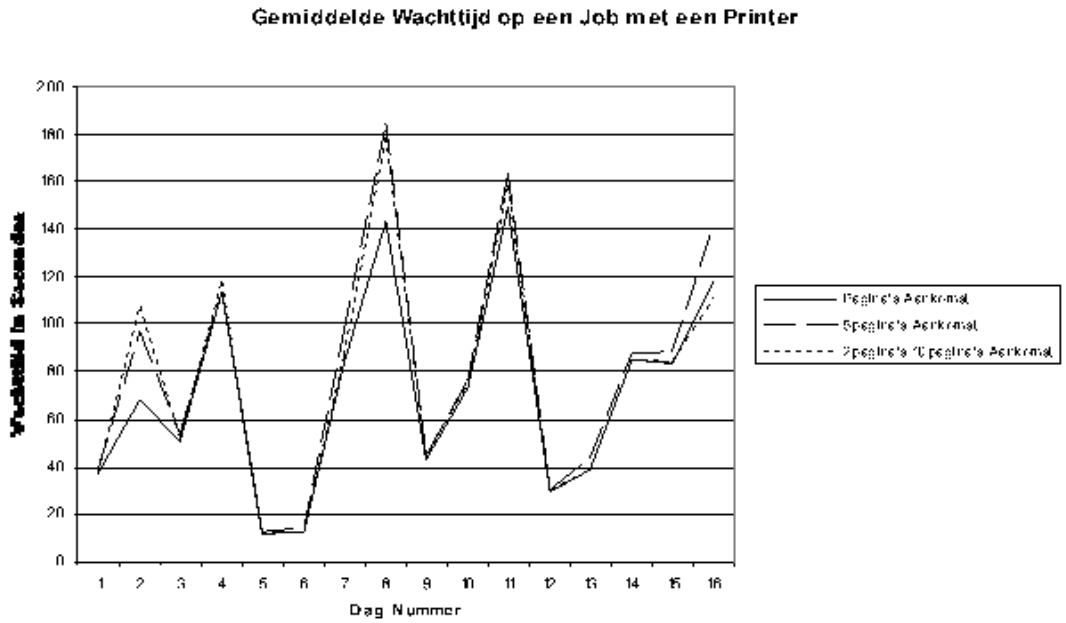
2.2 Vergelijking van Splitsing in kleine en grote Printopdrachten

Figuur 2.2 geeft globaal hetzelfde beeld als Figuur 2.1. Dit is interessant, omdat dit betekent dat we de gemiddelde wachttijd voor de printer dus al kunnen verlagen door slechts één grens te definiëren voor kleine en grote printopdrachten



Figuur 2.2

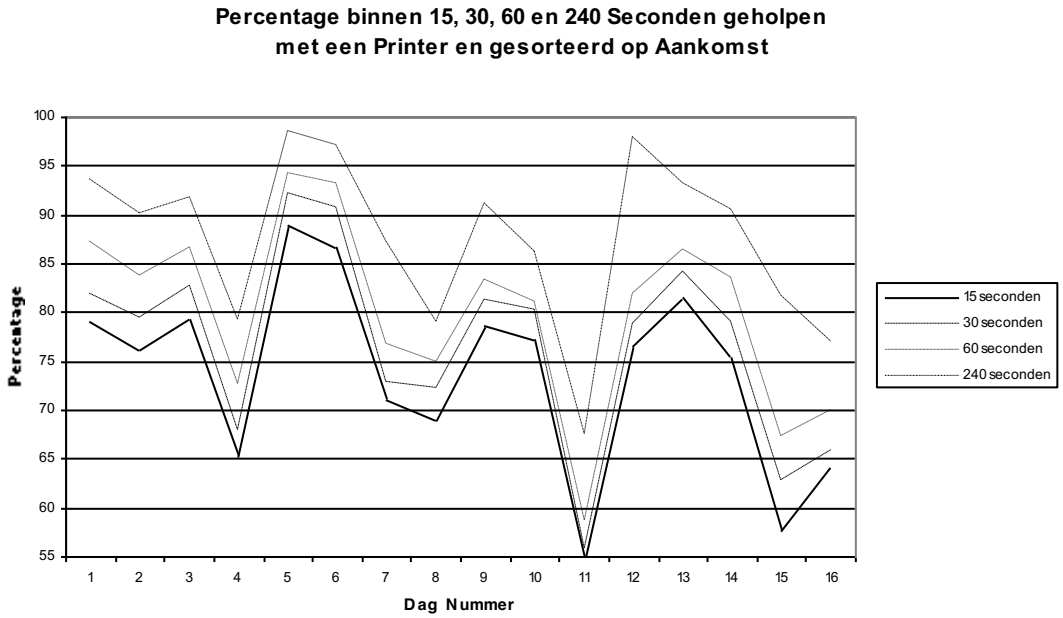
In Figuur 2.3 kunnen we goed zien dat gewoon domweg een bepaalde prioriteitscriterium gebruiken betere resultaten geeft dan een splitsing maken tussen grote en kleine printopdrachten. Dit is prettig om te weten, omdat we nou niet moeilijk hoeven te doen om een zo goed mogelijke grens te bepalen. Door meerdere grenzen te bepalen kunnen we de prestaties nog enigszins verbeteren, dit is logisch omdat dit in feite dichter in de buurt van gewoon domweg sorteren komt.



Figuur 2.3

2.3 Vergelijking van verschillende Servicegraad Parameters

In Figuur 2.4 hebben we verschillende servicegraad parameters naast elkaar gezet. Uit deze figuur kunnen we concluderen dat er toch nog altijd wel een behoorlijk aantal printopdrachten is met een wachttijd langer dan vier minuten. Verder lijkt het erop dat zich tussen twee opeenvolgende servicegraadparameters groep toch wel een behoorlijk aantal printopdrachten bevindt, waarmee we willen zeggen dat er niet een bepaalde grote groep printopdrachten is die korter dan 30 seconden moet wachten en dat de volgende grote groep printopdrachten de groep is die meer dan vier minuten moet wachten. De lijnen hebben namelijk haast hetzelfde patroon pas bij 240 seconden begint de grafiek een beetje af te vlakken.



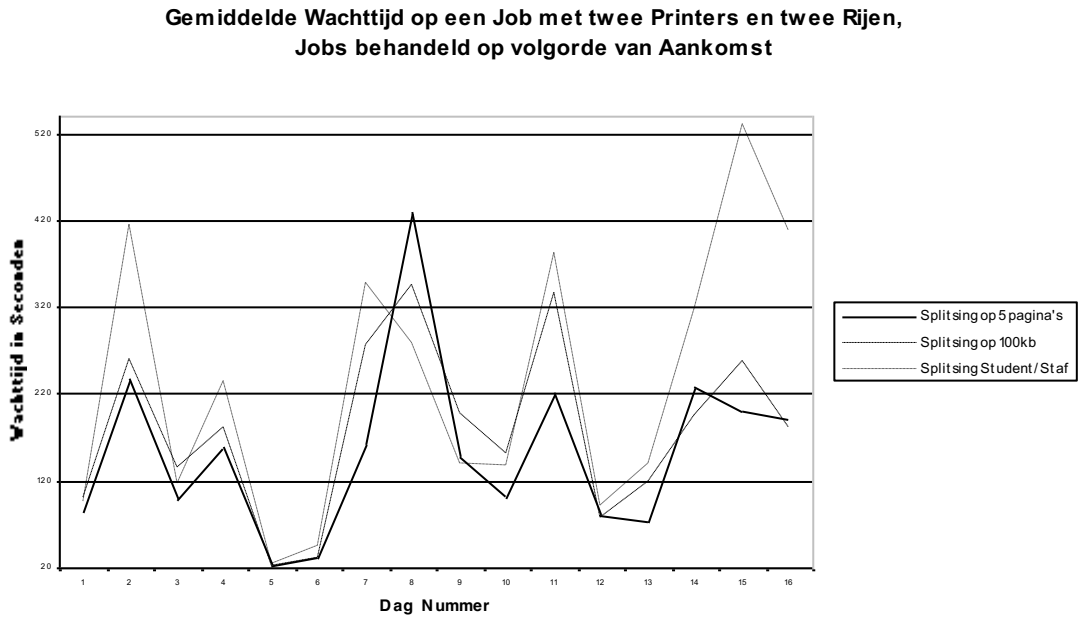
Figuur 2.4

3 RESULTATEN TWEE PRINTERS, TWEE RIJEN

In dit hoofdstuk bekijken we de resultaten met twee printers en voor iedere printer een afzonderlijke wachtrij.

3.1 Vergelijking van verschillende Splitscriteria

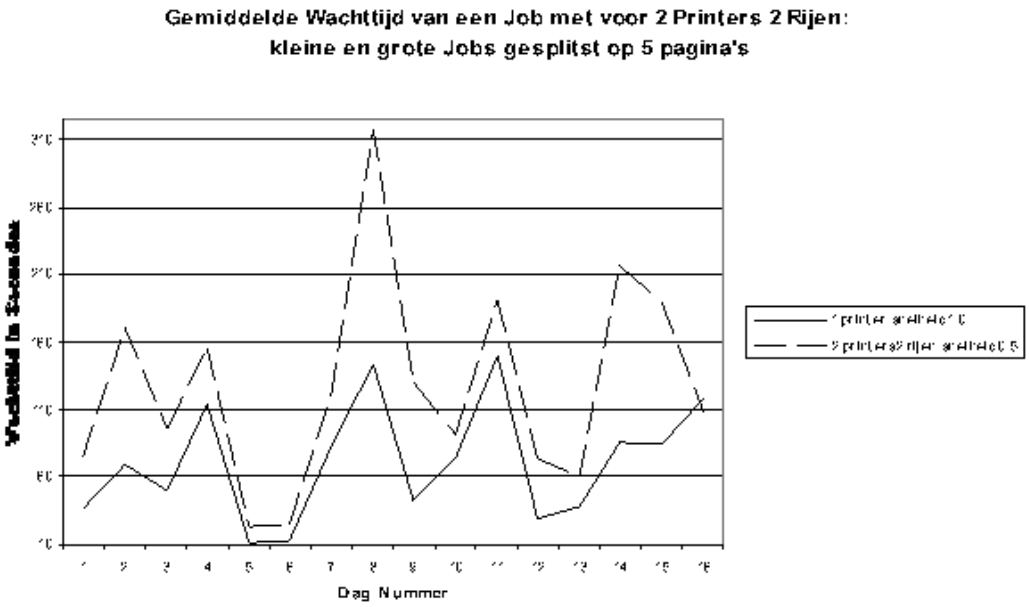
Om te bepalen welke printopdracht door welke printer uitgevoerd moet worden, maken we gebruik van splitscriteria. In Figuur 3.1 hebben we drie verschillende criteria tegen elkaar uitgezet. Hier zien we dat splitsen op het aantal pagina's verre weg het beste resultaat oplevert, maar dat splitsen in student-staf niet eens zoveel slechter is. Het belangrijkste is hier ook niet dat we een mooie splitsing maken in grote en kleine printopdrachten, maar dat we een zo goed mogelijke verdeling krijgen over de twee printers, zodat beide printers op elk moment van de dag even druk zijn.



Figuur 3.1

3.2 Eén Printer op volle Snelheid, Twee Printers op halve Snelheid

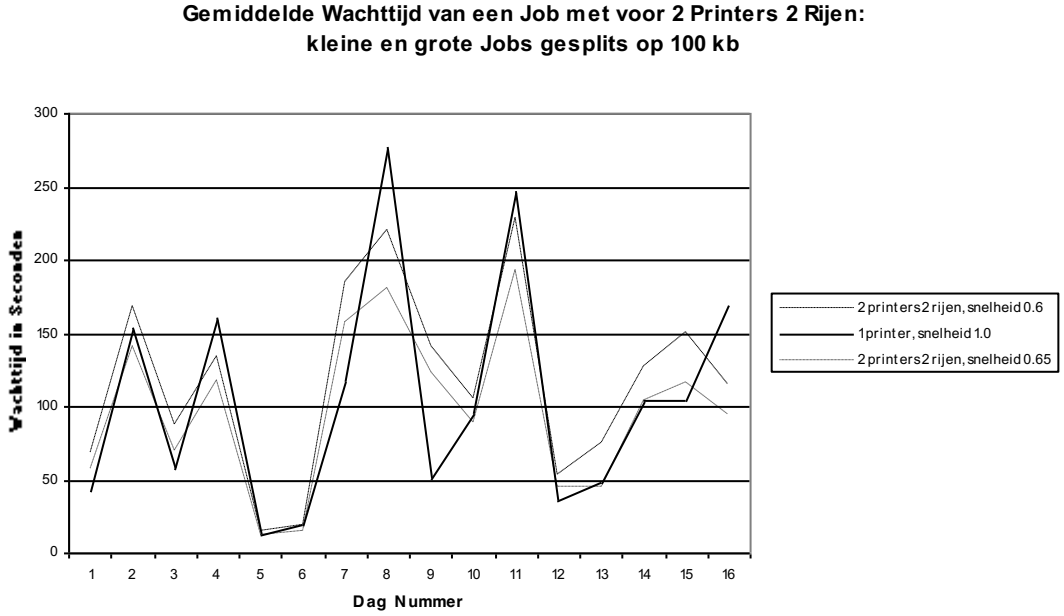
In Figuur 3.2 zien we dat twee printers op halve snelheid duidelijk minder presteren dan één printer op volle snelheid. Dit konden we ook wel verwachten, want we hebben gelijke capaciteit maar printopdrachten die zich eenmaal in een bepaalde rij bevinden kunnen niet meer door de andere printer uitgevoerd worden ook als deze niets te doen heeft.



Figuur 3.2

3.3 Eén Printer op volle Snelheid, Twee Printers met vergelijkbare Snelheid

In Figuur 3.3 kunnen we zien dat wanneer we twee printers hebben met een snelheid tussen de 60 en 65% van de volledige snelheid, dat deze dan vergelijkbaar is met één printer op volle snelheid. Wanneer we dus op zoek zijn naar vervanging van bijvoorbeeld een Océ printer, dan kunnen we ook kijken naar twee printers met 65% van de snelheid van de Océ printer. Op het moment dat deze printers dan meer dan twee keer zo goedkoop zijn wordt dit al gauw een interessante optie, waarbij men zich wel moet bedenken dat twee printers over het algemeen meer onderhoud met zich meebrengen, dus dan zal een afweging gemaakt moeten worden.



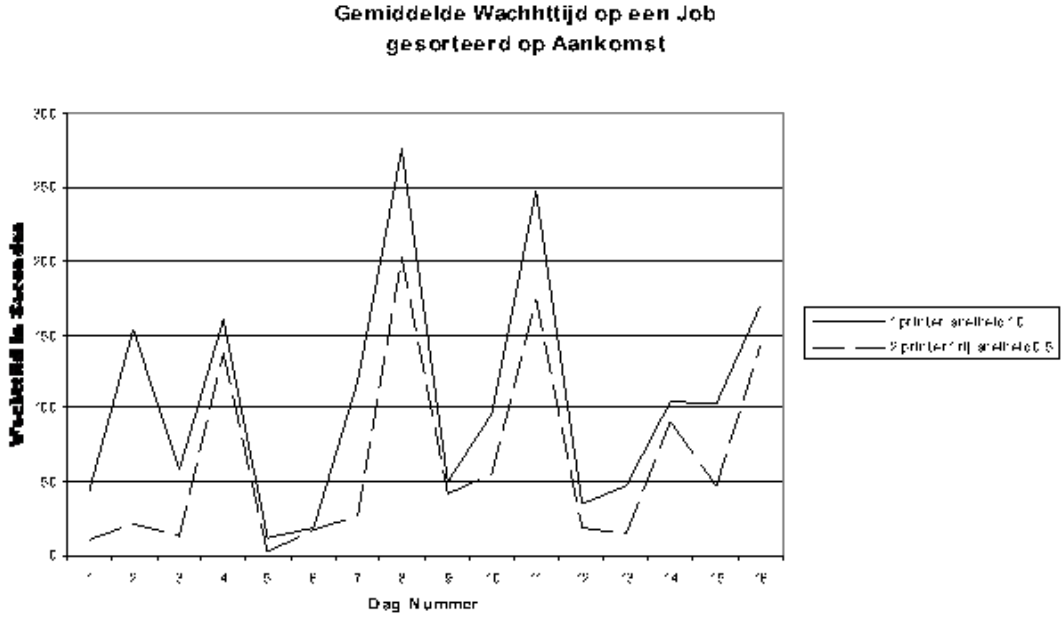
Figuur 3.3

4 RESULTATEN TWEE PRINTERS, ÉÉN RIJ

In dit hoofdstuk bekijken we weer de resultaten met twee printers, maar nu hebben we één gezamenlijke wachtrij voor beide printers. Dit betekent dat de printopdracht die zich vooraan in de wachtrij bevindt, wordt geholpen door de eerste de beste printer die vrij komt.

4.1 Eén Printer op volle Snelheid, Twee Printers op halve Snelheid

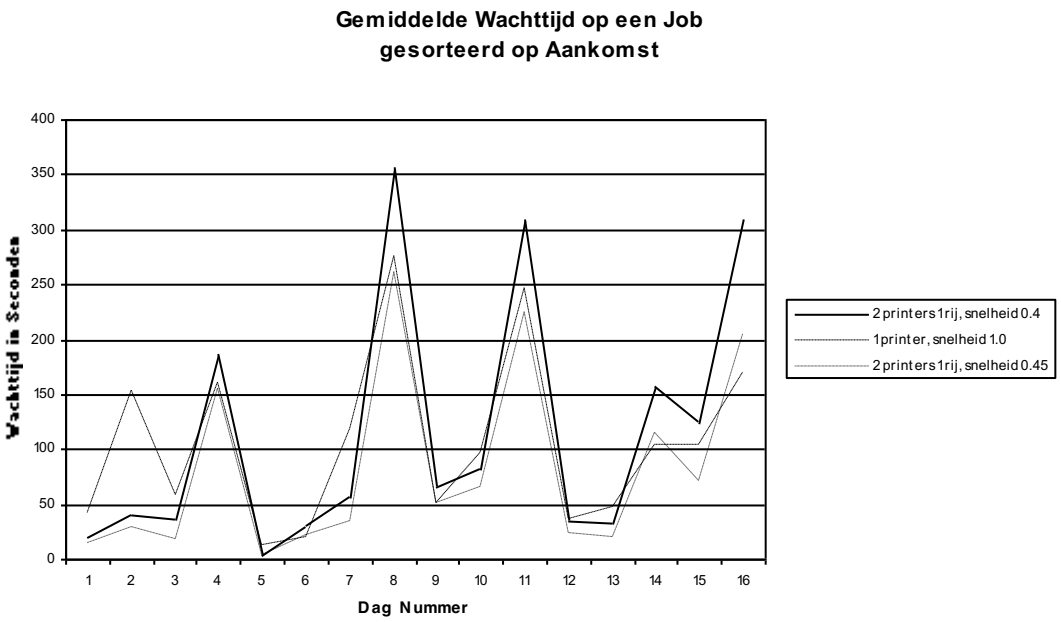
Figuur 4.1 laat duidelijk zien dat twee printers inderdaad meer kunnen dan één, ook al hebben ze maar de helft van de snelheid. Dit kunnen we verklaren door het feit dat één van beide printers gewoon door kan met printen van de kleine printopdrachten op het moment dat de ander druk bezig is met het printen van een grote printopdracht.



Figuur 4.1

4.2 Eén Printer op volle Snelheid, Twee Printers met vergelijkbare Snelheid

In Figuur 4.2 hebben we laten zien dat we met twee printers en één gezamenlijke wachtrij al voldoende hebben aan twee printers op 40 à 45% van de snelheid van één volwaardige printer. Hierbij moeten we beseffen dat dit alleen een oplossing is op het moment dat beide printers zich fysiek naast elkaar bevinden en dat ze een identieke printkwaliteit leveren.


Gemiddelde Wachttijd op een Job
gesorteerd op Aankomst
Wachttijd in Seconden
400
350
300
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Dag Nummer
2 printers 1rij, snelheid 0.4
1 printer, snelheid 1.0
2 printers 1rij, snelheid 0.45

Figuur 4.2

5 CONCLUSIES EN AANBEVELINGEN

In dit hoofdstuk zullen we kort de resultaten van het onderzoek samenvatten en een aantal aanbevelingen doen voor verder onderzoek.

Een van de eerste resultaten die meteen opviel was dat de grafieken van de servicegraad kwalitatief gezien hetzelfde beeld vertonen als de grafieken van de gemiddelde wachttijd. Dit is prettig om te weten om dat we nu niet hoeven te kiezen wat we als belangrijkste prestatiemaat kiezen op het moment dat we op zoek zijn naar een verbetering van de service van de printer.

Kijken we naar de verschillende prioriteitscriteria dan hebben we gezien dat deze het beste werken op het moment dat we precies weten hoe lang de printopdrachten gaan duren. Het aantal pagina's bleek de beste voorspeller, maar het aantal bytes deed het niet veel slechter en dit is een waarde die we al van tevoren weten. Nu hoeven we dus geen onderzoek te doen naar hoe we uit de verschillende file types het aantal pagina's moeten halen.

Verder onderzoek kan eventueel bestaan uit het zoeken naar een betere voorspeller voor de duur van een printopdracht. Hierbij kan men denken aan het meenemen van het file-type (*.txt, *.ps, *.pdf, etc.), maar ook of de opdracht afkomstig is van een Windows of UNIX computer. Uit de resultaten waarbij we deden of we precies wisten hoe lang een printopdracht gaat duren, blijkt namelijk dat we hier nog wel het een en ander kunnen winnen als we beter kunnen voorspellen hoe lang een printopdracht gaat duren.

Een mooi resultaat van dit onderzoek is dat domweg sorteren op een van de criteria van een printopdracht beter werkt dan het splitsen in twee of meer groepen. Dit is prettig omdat er dan geen optimale grenzen voor de groepen bepaald hoeft te worden. Wat wel nog onderzocht kan worden is het opdelen in groepen bijvoorbeeld gebaseerd op het aantal bytes en dan binnen deze groepen sorteren op het aantal pagina's in plaats van op het moment van aankomst.

Juist op de drukkere dagen hebben de door ons geteste rijdisciplines het meeste effect, want op het moment dat al te maken hebben met een relatief goede servicegraad, dan zien we ook dat er bijna geen verschil zit tussen de verschillende disciplines. Dit is goed om te weten want juist op de drukkere dagen willen we de wachttijd en/of de servicegraad verbeteren. Hiermee kunnen we het argument van systeembeheer weerleggen om niet over te gaan op een van onze geteste rijdisciplines. Zij zeggen namelijk op het moment dat je gaat sorteren op de grote van de printopdrachten men geneigd zal zijn zijn/haar printopdrachten op te delen in meerdere kleine opdrachten en dit is natuurlijk een ongewenste situatie. De oplossing hiervoor is dat we alleen op drukke momenten overschakelen op een andere rijdiscipline. Nu kun je natuurlijk zeggen dat ook dit bekend zal worden, maar het wordt toch al een stuk lastiger.

In verband met dit real-time wijzigen van de rijdiscipline zouden we onderzoek kunnen starten naar de effecten van een bepaalde rijdiscipline op het moment dat een wachtrij voornamelijk bestaat uit een groot aantal kleine printopdrachten of juist uit enkele hele grote printopdrachten. Mochten hier namelijk uit blijken dat verschillende rijdisciplines betere resultaten halen dan andere in een van deze situaties, dan wordt het dus al snel interessant om het real-time aanpassen van de rijdiscipline in te voeren.

Op het moment dat we twee printers gaan gebruiken dan worden de beste resultaten gehaald met een gezamenlijke wachtrij, uit praktisch oogpunt is dit minder handig omdat de persoon in kwestie niet precies weet uit welke printer zijn opdracht gaat komen. Dus alleen als printers fysiek naast elkaar staan is dit een optie. Dit geldt natuurlijk ook wanneer we wel twee rijen hebben, maar deze rijen splitsen op de grote van een printopdracht. Ook dan weet men namelijk niet van te voren uit welke printer zijn printopdracht zal komen. Dus met twee printers is uit praktisch oogpunt alleen de splitsing student/staf interessant. Natuurlijk hoeft deze verdeling niet heel strikt te zijn en kunnen we natuurlijk proberen de workload wat beter te verdelen door bijvoorbeeld nieuwe studenten toe te voegen als klant van een beide printers op het moment dat die een significant lagere workload blijkt te hebben.

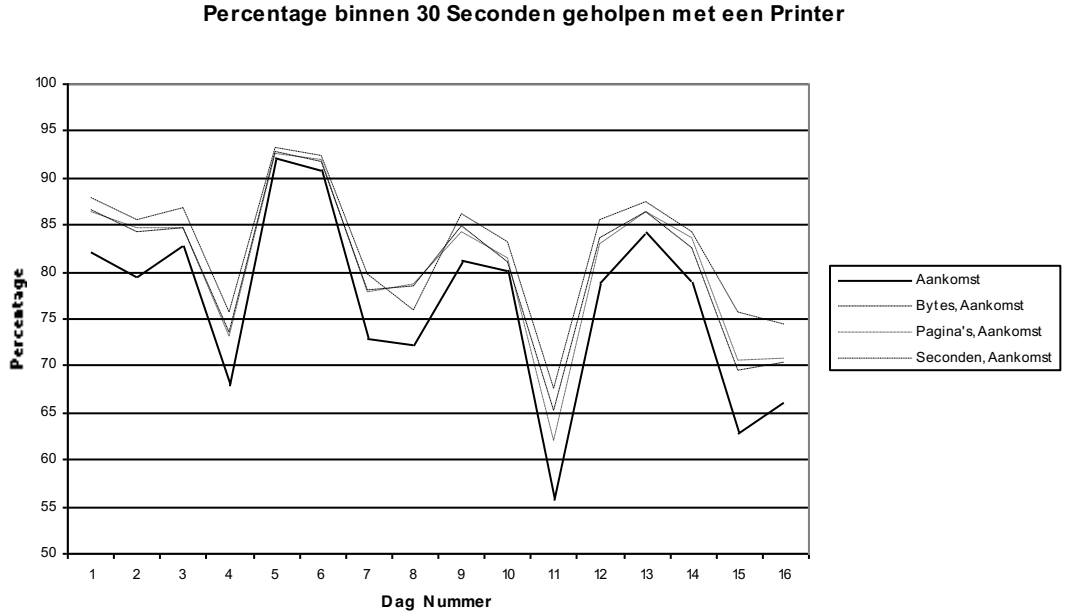
Wat wel interessant om te zien is wanneer we een student/staf verdeling doen, dit in sommige gevallen een stuk slechter blijkt te werken. Wat is hier de oorzaak van? Printen de studenten opeens veel meer uit in verband met deadlines of ligt het aan iets anders?

Tot slot willen we nog even in gaan op de beslissing of we één of twee printers moet nemen. Hierbij kan men namelijk een kort rekensommetje maken wat betreft de kosten. Op het moment dat men twee printers wil aanschaffen dan heeft men al voldoende aan twee langzamere printers, die ongetwijfeld een stukje goedkoper zullen zijn. Maar moet er wel rekening mee houden dat de afweging hier niet op houdt. Twee printers brengen vaak meer onderhoud met zich mee en men moet een prijskaartje kunnen hangen aan het extra ongemak (weet niet welke printer?) dat twee printers met zich meebrengt.

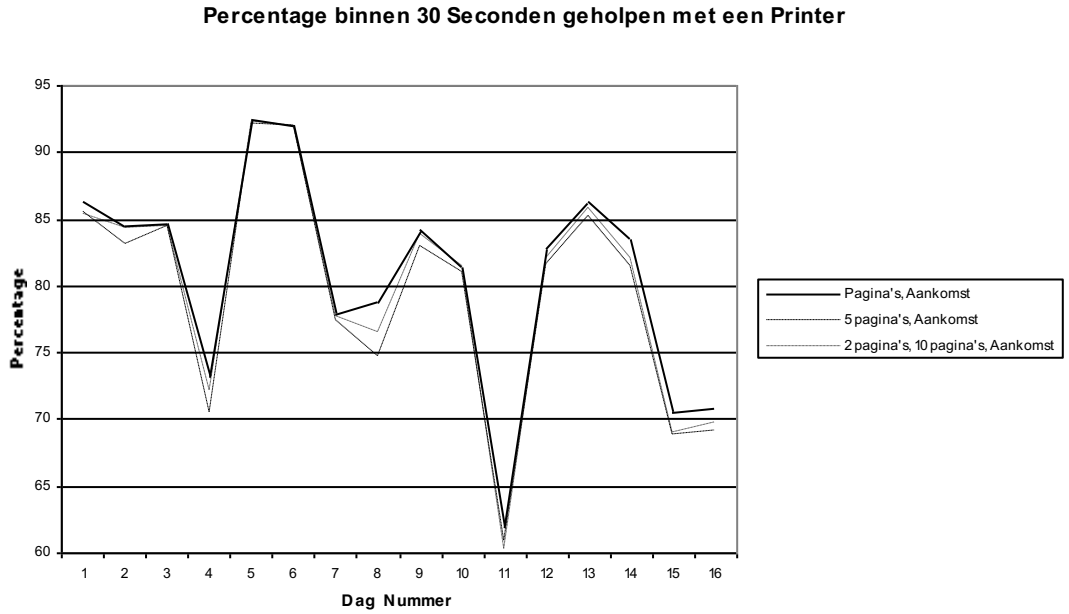
Kijken we naar de huidige situatie op de VU, dan zit men op de VU voorlopig wel goed met twee nieuwe Océ printers van volwaardige snelheid in tegenstelling tot de door geteste situatie met twee printer op 65% van de volwaardige snelheid.

APPENDIX A

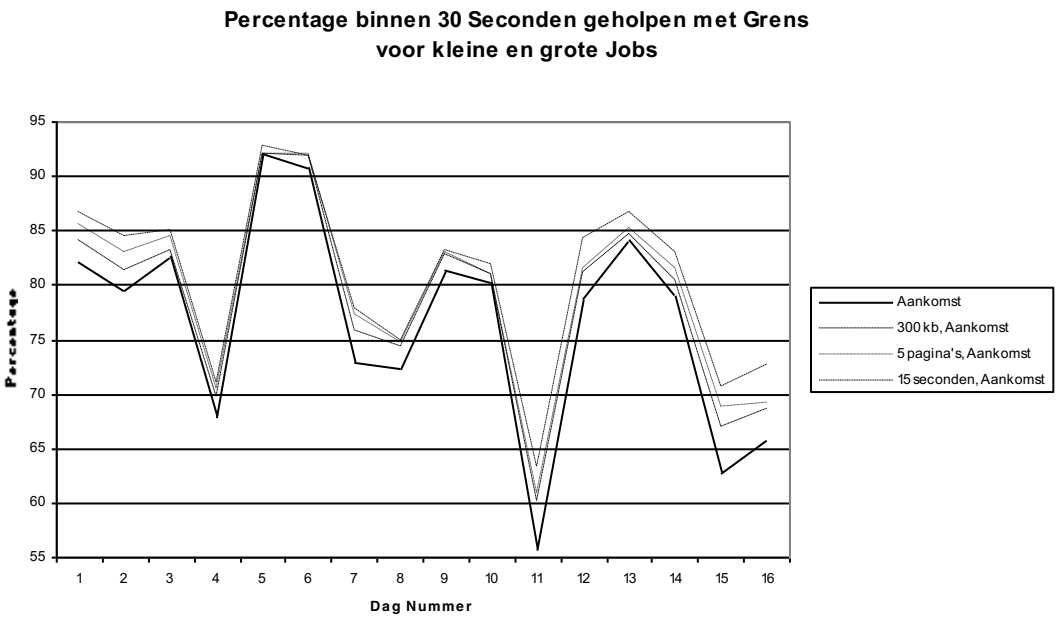
Servicegraad Grafieken bij Hoofdstuk 2 van één Printer, één Rij



Figuur A.1 behorende bij Figuur 2.1



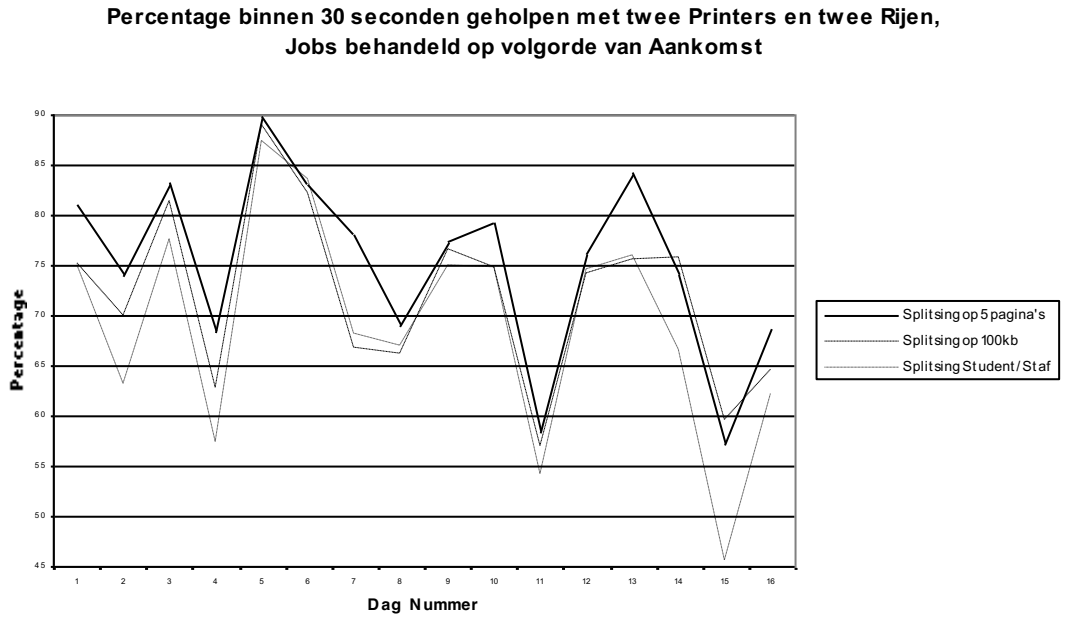
Figuur A.2 behorende bij Figuur 2.2



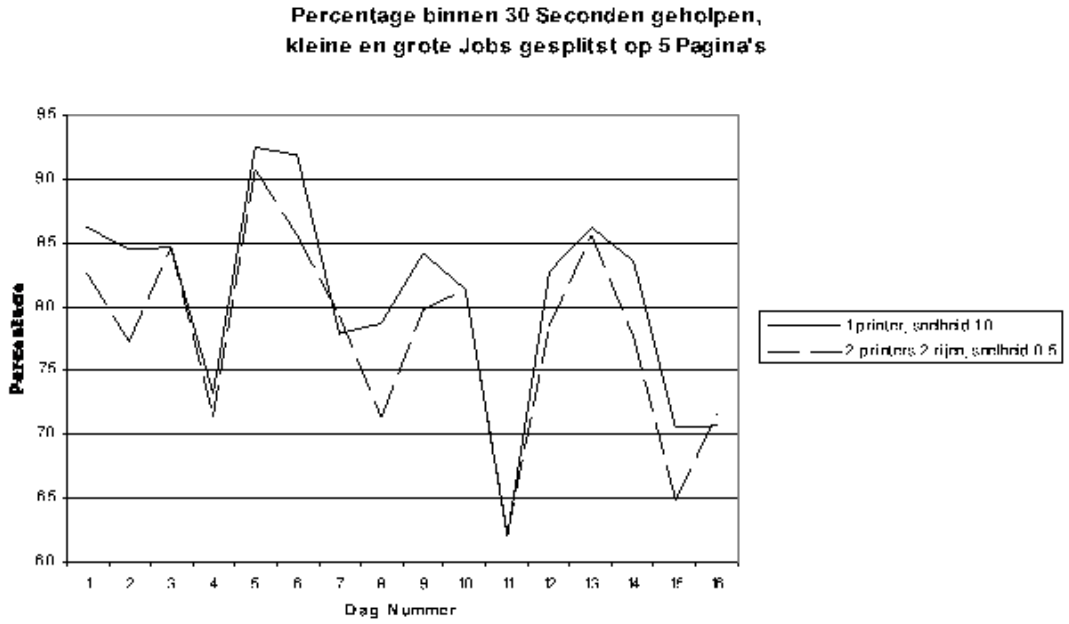
Figuur A.3 behorende bij Figuur 2.3

APPENDIX B

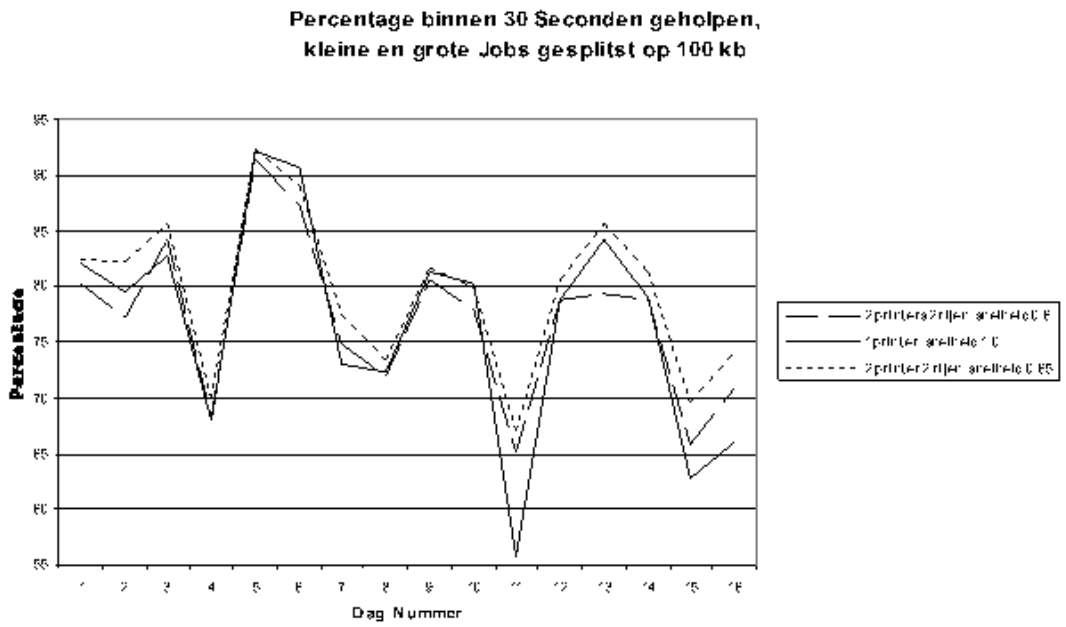
Servicegraad Grafieken bij Hoofdstuk 3 van twee Printers, twee Rijen



Figuur B.1 behorende bij Figuur 3.1



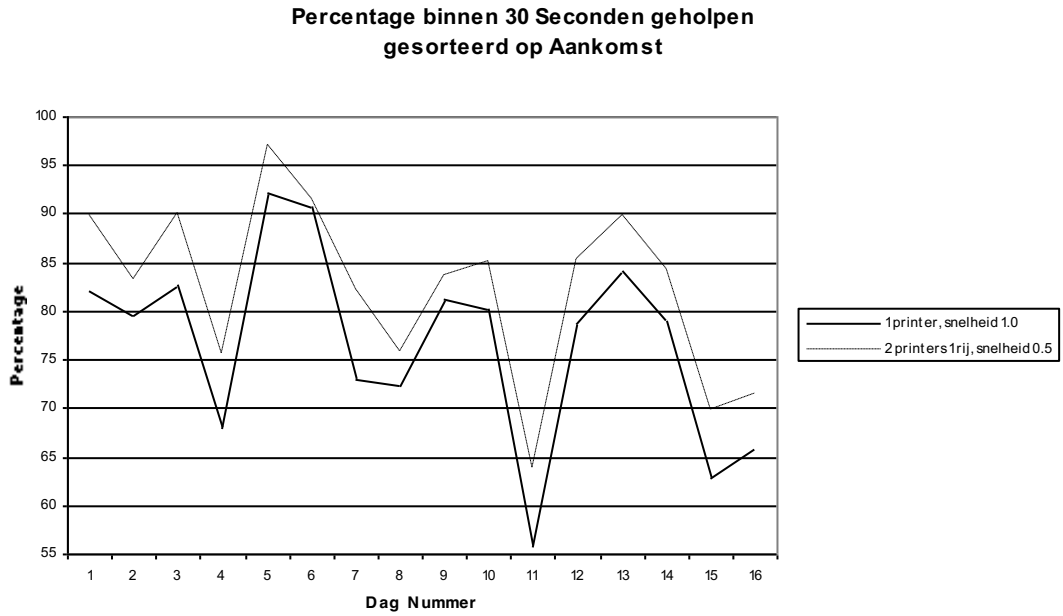
Figuur B.2 behorende bij Figuur 3.2



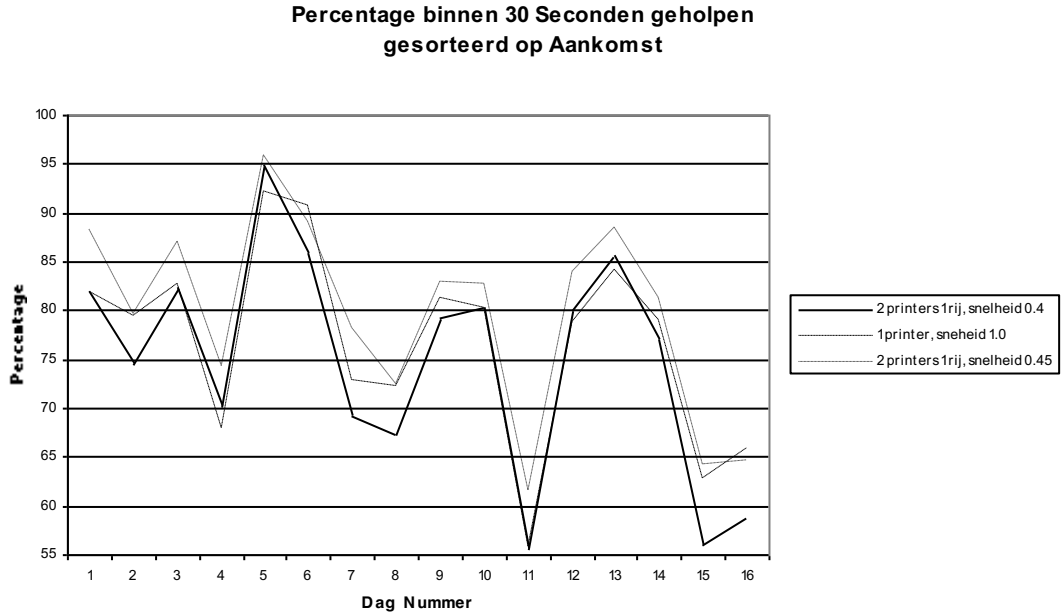
Figuur B.3 behorende bij Figuur 3.3

APPENDIX C

Servicegraad Grafieken bij Hoofdstuk 4 van twee Printers, één Rij



Figuur C.1 behorende bij Figuur 4.1



Figuur C.2 behorende bij Figuur 4.2

REFERENTIES

1. R D Nobel (1994) dictaat: *Simulatie technieken*, Vrije Universiteit, Amsterdam

2. M Sweijd (1995) BWI-werkstuk: *Océ is ready and printing? Een wachtrijmodel voor de printer*, Vrije Universiteit, Amsterdam

3. H C Tijms, EMF Kalvelagen (1994) Modelbouw in de operations research: *hoofdstuk 7 Wachttijdtheorie*, Academic Service, Economie en Bedrijfskunde, Schoonhoven