

The power of simple scheduling policies

Citation for published version (APA):

Rutten, C. (2013). *The power of simple scheduling policies*. Maastricht University.
<https://doi.org/10.26481/dis.20130201cr>

Document status and date:

Published: 01/01/2013

DOI:

[10.26481/dis.20130201cr](https://doi.org/10.26481/dis.20130201cr)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Nederlandse Samenvatting

Dit proefschrift bestudeert planningsproblemen, in de literatuur beter bekend als machine-volgorde problemen. Bij deze problemen moet men denken aan een groep machines die gezamenlijk een verzameling aan taken moet uitvoeren. De productietijd van een taak is de tijd die een machine nodig heeft om de taak af te werken en kan verschillen per taak en machine combinatie. Het beslissingsprobleem is welke machine welke taken moet verwerken zodat bijvoorbeeld de laatste machine zo vroeg mogelijk klaar is met haar laatste taak. Een ander voorbeeld is waarbij men de som van completeringstijden over alle taken wil minimaliseren.

Deze machine-volgorde problemen komen veelvuldig voor in de praktijk: van de manager van een autogarage die bepaalt welke werknemer welke auto repareert, tot de productieplanner van grote industriële productielijnen die de producten bepaalt die iedere productielijn de komende week gaat maken. In de literatuur is er veel onderzoek gedaan naar het ontwerpen van goede strategieën, ofwel algoritmes, voor een groot scala aan machine-volgorde problemen. Vanwege de complexiteit van deze problemen is het vaak niet mogelijk een algoritme te ontwerpen dat altijd het best mogelijke productieschema vindt binnen een acceptabele rekentijd. Daarom trachten wetenschappers meestal algoritmes te ontwikkelen die wel binnen acceptabele rekentijd een goed productieschema, ook wel oplossing genoemd, teruggeven. Een dergelijk algoritme wordt een approximatief algoritme genoemd als het, voor ieder mogelijke inputdata, een oplossing teruggeeft die bewijsbaar dichtbij de best mogelijke oplossing voor die inputdata ligt. In detail betekent dit dat de waarde van de geretourneerde oplossing maximaal een bepaalde absolute of relatieve afwijking heeft ten op zichte van de waarde van de optimale oplossing.

Vele ontwikkelde approximatieve strategieën zijn echter niet altijd geschikt voor implementatie in de praktijk. Dat deze weldoordachte algoritmes vaak de werkvloer niet bereiken kan zijn vanwege velerlei redenen; het duurt te lang voordat ze een oplossing retourneren, ze zijn te moeilijk in gebruik en begrip, de noodzakelijke data is niet voorhanden, ze werken alleen onder labcondities terwijl in de praktijk allerlei randfactoren meespelen, etc. Het doel van dit proefschrift is dan ook het onderzoeken van de kracht van simpele en snelle algoritmes die wel makkelijk in de praktijk geïmplementeerd kunnen worden. In het bijzonder bestudeer ik de kwaliteit van de schema's die geretourneerd worden door een aantal simpele algoritmes voor verschillende machine-volgorde problemen en daarbij bekijk ik hoe die kwaliteit zich verhoudt ten opzichte van de kwaliteit van het optimale schema.

In Hoofdstuk 2 onderzoek ik lokale zoekmethoden waarbij het algoritme zich van oploss-

ing naar oplossing beweegt en pas stopt als vanuit een huidige oplossing geen nieuwe betere buuroplossing kan worden gecreëerd. Een buuroplossing van een huidige oplossing wordt gecreëerd door een mutatie van de huidige oplossing: een taak wordt verwezen naar een andere machine. Ik bestudeer twee verschillende regels om een taak toe te wijzen aan een andere machine, de zogenoemde jump- en lexjump-buurruimte. Het specifieke probleem dat ik bestudeer kent m machines waaraan n taken moeten worden toegekend. De taken hebben verschillende groottes en de machines hebben verschillende snelheden zodat de productietijd van een taak gelijk is aan de taakgrootte gedeeld door de snelheid van de machine. Daarbij zijn sommige taken beperkt in de zin dat ze niet aan alle machines kunnen worden toegewezen. Ik laat zien dat de kwaliteit van een geretourneerde oplossing van de jump- en lexjump-buurruimte respectievelijk maximaal een factor \sqrt{ms} en $c * \log S / \log \log S$ af liggen van de kwaliteit van de optimale oplossing. Hierbij, stelt s de maximale snelheid over alle machines voor, S de som van alle snelheden en c is een voldoende grote constante.

Hoofdstuk 3 bekijkt hetzelfde machine-volgorde probleem en ook dezelfde algoritmes als het voorgaande hoofdstuk. Dit hoofdstuk bestudeert de kwaliteit van de oplossingen van beide algoritmes als de invoerdata voor het algoritme onderhevig is aan een kleine hoeveelheid ruis. De motivatie is afgeleid van het feit dat algoritmes in praktijk het vaak veel beter doen dan de theoretische slechtste-geval-garantie. Deze populaire theoretische garantie bekijkt de kwaliteit van een algoritme voor de slechts mogelijke invoerdata en zegt daarmee dat voor een andere invoer de kwaliteit nooit slechter kan zijn dan de waarde van deze slechtste-geval-garantie. Die slechts mogelijke invoerdata is echter vaak kunstmatig geconstrueerd en wordt zelden of nooit waargenomen in de praktijk. Het gat tussen de kwaliteit van een algoritme geobserveerd in praktijk en de theoretische slechtste-geval-garantie is dan ook vaak groot. Hoofdstuk 3 tracht dit gat te verkleinen door een kleine hoeveelheid ruis toe te voegen aan de invoerdata in de hoop de onrealistische structuur van de slechts mogelijke invoerdata te verstoren en zodoende weer een goede kwaliteit oplossing te bereiken. Het hoofdstuk laat inderdaad zien dat, in de aanwezigheid van een beetje ruis, de kwaliteit van de bestudeerde algoritmes significant verbeterd ten opzichte van de slechtste-geval-garanties die werden afgeleid in Hoofdstuk 2.

Hoofdstukken 4 en 5 bekijken een alternatief machine-volgorde probleem waarbij de centrale vraag is of er überhaupt een productieschema bestaat dat aan alle beperkingen en taakeigenschappen voldoet. Iedere taak brengt over tijd met een vast interval een serie aan kleinere subtaken voort met dezelfde productietijd maar met ieder een eigen relatieve deadline. Na een verdeling van de taken over machines wordt vervolgens op iedere machine het zogenoemde Vroegste-Deadline-Eerst (VDE) algoritme toegepast om tot een productieschema per machine te komen. VDE is een simpel en populair algoritme dat optimaal is voor het geval dat er slechts een machine beschikbaar is. Het resulterende vraagstuk voor meerdere machines is het volgende: bestaat er een toewijzing van de taken aan de machines zodat, als VDE wordt toegepast per machine, een valide productieschema per machine wordt gevonden. Hierbij definieer ik een valide productieschema als een schema waarbij iedere subtaak kan worden voltooid voor haar deadline en waarbij ook iedere subtaak wordt toegewezen aan de machine waarop de 'moedertaak' is ingedeeld.

In Hoofdstuk 4 richt ik mij op de situatie waarin de productietijd van een taak afhankelijk is van zowel de taak als de machine waarop zij wordt ingedeeld. Ik stel een 12.9-approximatief algoritme voor, dat is, het algoritme retourneert of (*i*) er een verdeling bestaat van de taken

over machines zodat VDE een valide productieschema per machine geeft als de machines 12.9 keer zo snel gaan, of (ii) concludeert dat er geen verdeling van de taken over de machines bestaat waarvoor VDE een valide productieschema geeft per machine als de machines hun originele snelheid behouden. De looptijd van dit algoritme is polynomiaal in de hoeveelheid invoerdata. Het tweede gedeelte van dit hoofdstuk stelt een $(1 + \epsilon)$ -approximatief algoritme voor waarvan de looptijd polynomiaal is in het aantal taken maar wel exponentieel is in het aantal machines en in de constante $1/\epsilon$.

Het probleem dat bekeken wordt in Hoofdstuk 5 is een iets eenvoudigere versie van het probleem dat in Hoofdstuk 4 centraal staat. Nu is de productietijd van een taak gelijk over alle machines. Ik presenteer opnieuw een $(1 + \epsilon)$ -approximatief algoritme waarvan de looptijd deze keer polynomiaal is in het aantal taken en in het aantal machines en exponentieel is in slechts de constante $1/\epsilon$ en in $\log \lambda$, waar λ de ratio is van grootste relatieve deadline over de kleinste relatieve deadline.

Hoofdstuk 6 tenslotte richt zich op stochastische machine-volgorde problemen waarin de productietijd van een taak niet langer deterministisch is maar stochastisch. In standaard stochastische machine-volgorde problemen wordt aangenomen dat de onderliggende kansverdeling van een stochastische variabele volledig bekend is. In Hoofdstuk 6 verruim ik deze aanname door te stellen dat de parameters van de onderliggende distributie onbekend zijn. Ondanks dat de parameters onbekend zijn, kan men door het uitvoeren van taken en het dus observeren van realisaties van de onderliggende distributie wel leren over deze parameters. Een optimaal algoritme voor het bestuurd probleem is uit de literatuur bekend. Dit algoritme is helaas in zowel de hoeveelheid berekeningen als de noodzakelijke precisie daarvan te veeleisend om nuttig te zijn voor praktische doeleinden. In Hoofdstuk 6 bestudeer ik de kwaliteit van de oplossingen gegenereerd door twee verschillende simpele en snelle algoritmes. Ik laat zien dat het dynamische algoritme dat kan leren en zijn strategie kan aanpassen veel krachtiger is dan zijn statische variant.

Over alle hoofdstukken heen concludeer ik dat simpele algoritmes die in praktijk veel eerder toegepast zullen worden dan de reken-intensieve algoritmes uit de literatuur ook theoretisch aardig presteren. Ik heb voor een aantal van dergelijke eenvoudige algoritmes de theoretische slechtste-geval-garantie bepaald die vaak een polynomiale of soms constante waarde kent. Om de kwalitatieve analyse van algoritmes door middel van slechtste-geval-garanties dichterbij de geobserveerde kwaliteit van die algoritmes in praktijk te brengen wend ik mij tot het toepassen van het concept ‘smoothing’. In smoothing wordt er een kleine hoeveelheid ruis toegevoegd aan de invoerdata om op die manier slechte maar kunstmatige instanties uit te schakelen. In de aanwezigheid van een kleine hoeveelheid ruis presteren een aantal van de besproken algoritmes nog beter en benaderen soms zelfs optimaliteit.