UNIVERSITEIT GENT

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

Time/cost optimization and forecasting in project scheduling and control

Mathieu Wauters

Dissertation submitted to the Faculty of Economics and Business Administration, Ghent University, in fulfillment of the requirements for the degree of Doctor in Applied Economics

"No one pursuit can be successfully followed by a man who is preoccupied with many things."

- *Seneca*

UNIVERSITEIT GENT

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

Advisor:

Prof. dr. Mario Vanhoucke

Doctoral jury:

Prof. dr. Marc De Clercq	Ghent University, Dean
Prof. dr. Patrick Van Kenhove	Ghent University, Academic Secretary
Prof. dr. Mario Vanhoucke	Ghent University, Vlerick Business School, University College London (UK)
Prof. dr. Öncü Hazır	TED University (Turkey)
Prof. dr. Pierre Bonnal	CERN, the European Organization for Nuclear Research (Switzerland)
Prof. dr. Broos Maenhout	Ghent University
Prof. dr. Geert Poels	Ghent University

Dankwoord

> Het lijdt geen twijfel: het schrijven van een dankwoord is een NP-Hard probleem. Schier onmogelijk is de taak om iedereen op te sommen met wie ik de afgelopen jaren lief en leed heb gedeeld. In wat volgt zal ik een schuchtere poging ondernemen, maar graag dek ik me op voorhand in tegen acute amnesie. Of je nu geïnteresseerd was in mijn persoon, mijn werk of op zoek was naar de meest recente Photoshop-prent of portie licht amusement, bedankt!

Vanzelfsprekend behoort de eerste bedanking toe aan mijn promotor, Mario Vanhoucke. Mario (a.k.a Mister Ario), dankzij jou heb ik de gelegenheid gekregen om de onderzoekswereld te leren kennen. We hebben de afgelopen jaren vaak zeer succesvol kunnen samenwerken. Daarbij bood je me veel opportuniteiten, waaronder lesgeven en het Londense avontuur aan UCL. Daarnaast ben ik ook dankbaar omdat je mij toeliet mijn eigen weg voor dit doctoraat uit te stippelen. Het was zoals bij ieder traject een proces van vallen en opstaan, maar ik apprecieer dat ik met voor onze onderzoeksgroep minder bekende technieken aan de slag mocht gaan. Ik koester ook de herinneringen aan ochtendlijke *fokkies*, vitterijen als we van en voor elkaar iets corrigeerden en de gedeelde attitude om te excelleren, ongeacht het belang van de opdracht. Nu ik de onderzoekswereld achter mij laat, ben ik fier in de voetsporen te treden van de overige gedoctoreerde welpen. Bedankt voor alles!
Gaëtane, bedankt om me steeds met een kamerbrede smile te verwelkomen als ik in het weekend, geheel à l'improviste en enigszins schaapachtig lachend, aanbelde om Mario een nieuwe draft van deze of gene paper te bezorgen.

Graag wil ik ook de leden van mijn **examenjury** uitdrukkelijk bedanken. Ik apprecieer enorm dat jullie ondanks de drukke agenda's toch de nodige tijd willen investeren in het lezen van dit doctoraat en het verbeteren van de kwaliteit ervan. Broos, jouw oog voor detail is ongeëvenaard, getuige de vele relevante commentaren die ik van jou kreeg. Geert, het feit dat jij dit doctoraat vanuit een ander perspectief benadert, heb ik als verruimend en verfrissend ervaren. Bedankt om me aan te sporen om enkele assumpties

toch duidelijker te staven. I am greatly indebted to the external members of my jury, prof. dr. Pierre Bonnal and prof. dr. Öncü Hazır. Pierre, thank you for challenging me to demonstrate the applicability of the proposed techniques from a practical point of view. Öncü, who knew that, after reading your many excellent papers on the Discrete Time/Cost Trade-off Problem, you would eventually become a member of my jury? It is a great honour to have you on this jury. Thank you for the constructive criticism that undoubtedly elevated the level of this dissertation.
Ook prof. dr. Roland Paemelaire en prof. dr. Paul Gemmel ben ik dank verschuldigd. Zij waren bereid om te hulp te schieten toen het vinden van een datum voor de publieke verdediging een wel zeer complex planningsprobleem werd.
Tevens wens ik het Fonds voor Wetenschappelijk Onderzoek (FWO) te bedanken omwille van de financiële ondersteuning van dit doctoraat. De UGent verdient een woord van dank voor het ter beschikking stellen van de High Performance Computing infrastructuur die voor de vele experimenten broodnodig bleek te zijn.

Achter elke sterke vakgroep schuilt een sterk **secretariaat**. Machteld & Martine (M&M), bedankt om me wegwijs te maken als ik verloren liep in het bos der administratie. Wanneer ik werd geconfronteerd met de kinderziektes van Oasis, wist één persoon steeds raad. Lena, bedankt om me steeds te helpen, om altijd met de glimlach klaar te staan en onverwachtse traktaties aan te bieden. Zelden heb ik een zo goedhartig persoon ontmoet als jij. Marleen, jij maakte van het kuisen van ons bureau geen routineklus, maar koppelde er vaak hoogst entertainende gesprekjes aan vast. Bedankt om onze Miss Proper te zijn!

Graag wil ik mijn **collega's** in de bloemetjes zetten. Veronique, bedankt om me gans in den beginne zo goed op te vangen, evenals voor onze vele betekenisvolle gesprekken. Ik heb alvast gepoogd jouw erfenis om nieuwe welpen op te vangen verder te zetten. Christophe, Thomas, Jeroen en Len: ongelofelijk hoe vaak mijn lachspieren dankzij jullie op de proef werden gesteld! Bedankt om me de donkere krochten van het internet te leren kennen, maar eveneens om lief en leed te kunnen delen. Met veel plezier blik ik terug op onder andere het weekend in Londen en de EURO-conferentie in Rome. Stars van Simply Red is sindsdien nooit hetzelfde geweest. Het was aangenaam een collega-techneut aan te treffen in Louis-Philippe. De HPC-verzuchtingen, nerdtalk over Raspberry Pi's en het lange wachten op revisies schiep onvermijdelijk een band. Pieter, Laura, Annelies en Jeroen B. wil ik bedanken voor de aangename lunches, bezoekjes aan de OR en pauzes. Hoewel het onmogelijk is onze diversiteit aan gesprekken in enkele

zinnen te gieten, is mijn dankbaarheid groot. Ook de Friday drinks waren in dit opzicht aangenaam en werden opgeleukt door de aanwezigheid van Jordy, Jonas en Danica. Jonas, bedankt om een fijne bureaugenoot te zijn tijdens deze laatste weken!
Tijdens de middagpauzes of voor een gesprekje in de gang was ik kind aan huis bij de collega's van EB09. Ann (a.k.a Penny) en Sofie, bedankt voor de fijne momenten, jullie interesse in mijn doctoraat en de vele aanmoedigingen. Griet, Melissa, Kaat, Jonas, Robin, Jolien en Anneleen: merci, zowel voor het luisterende oor als om mijn wereld te verruimen met allerhande weetjes. Ongetwijfeld worden het perfecte wist-je-datjes tijdens een receptie! Charlotte en Elien, bedankt voor jullie gezelschap. Fantastisch hoe we elkaar tijdens onze studieperiode amper kenden, maar hoe een doctoraat hier verandering in kon brengen. Laten we binnenkort de gesmaakte lunch herhalen! Bart (Larivière), ik heb veel gehad aan de conversaties die we voerden. Jouw deur staat steeds open en daarnaast heb ik fenomenaal veel respect voor wat je met CSI uitbouwt en de manier waarop je jouw doctoraatsstudenten begeleidt. Je bent een voorbeeld voor velen! Katrien, ook al kwam ik initieel jouw bureau binnen om Arne met Voseko-vragen te bestoken, ik ben blij dat we mede daardoor aan de praat geraakt zijn. Ik herinner me gesprekken over Kant (gelukkig met hoofdletter) en ethiek, maar evenzeer over lichtere thema's. Dat ik indien nodig ook bij jou terecht kon, betekent meer dan ik al heb gezegd. Merci hiervoor! Arne, ongelofelijk waar je de tijd en energie vindt om iedereen te hulp te snellen en er tevens een hoge productiviteit op na te houden. Ik heb veel bewondering voor jou als onderzoeker en nog veel meer als mens. Over bewondering gesproken dien ik ook die andere speciale collega te bedanken. Ine, queen van de iMessages, je bent een van de meest energieke en eerlijke mensen die ik ken. Met plezier denk ik aan onze lunches die steevast uitliepen, de vele berichtjes om gewoon te checken hoe het ging of de geweldige fotosessie. Enorm bedankt voor jouw niet-aflatende steun!
Hoewel we elkaar niet vaak spreken, was het steevast aangenaam te kunnen bijpraten met de Marketing-collega's, onder wie Willem, Dries, Michel, Nanouk, Julie, Caroline en Stefanie.

Doorheen dit doctoraal traject heb ik de kans gekregen van enkele **verenigingen** deel uit te maken. Dank aan het FEB OAP-overleg en in het bijzonder aan Jolien en Xavier om de belangen van doctorandi en assistenten te behartigen.
Meer tijd spendeerde ik aan Voseko, de alumnivereniging van onze faculteit. Ik leerde er een geweldige groep toegewijde alumni kennen. Aan Vincent, Annelore, Katrien, Len, Bregt, Fleur, Toon, Jürgen, Jens, Ward, Dauwe, Pieter en zovele anderen: bedankt voor de prettige samenwerking! Ook merci aan onze decaan prof. dr. Marc De Clercq, die

ondanks zijn drukke agenda steevast aanwezig was tijdens de vergaderingen en bereid was om zijn netwerk aan te spreken bij de zoektocht naar bevlogen sprekers.

Een gezonde geest in een gezond lichaam! Een doctoraat vergt bepaalde kwaliteiten zoals een ijzeren discipline en een ongeëvenaard doorzettingsvermogen. Mijn **sportieve activiteiten**, immer een belangrijke uitlaatklep, brachten mij dit reeds vroeg bij. Hoewel de klemtoon de laatste jaren is verschoven naar het voetbal, blik ik met veel plezier terug op mijn tennisperiode bij TC Racso. In het bijzonder wil ik Gilles bedanken. Gilles, we kennen elkaar ondertussen al ongeveer 20 jaar. Samen op een tennisveld opgroeien schept een band en dat wordt bewezen door het feit dat we nog steeds samen een balletje slaan, trashtalk incluis. Ik kijk er naar uit je ook in de toekomst van het kastje naar de muur te spelen.
Mijn collega-wereldspelers van Galatasaray Ledeberg: bedankt voor de leuke voetbalmomenten! Nog voor onze tegenstanders zijn bekomen van de schok dat ons team niet uit buitenlanders bestaat, worden ze vakkundig in de pan gehakt door Gent en Ledebergs *finest*. Bedankt Tim, Jabbe, Lennert, Marius, Wouter, Arne, Lukas, Ben, Dauwe, Stephan en Adrian. Laten we er alweer een topseizoen van maken!
Ook Economini, ons *weireldploegsje* in de personeelscompetitie, verdient een plaats in de spotlight! Niet alleen mochten onze sportieve prestaties gezien worden, het was ook fijn om mensen van andere vakgroepen te leren kennen en achteraf met een pint te kunnen bijpraten over ernstige, maar veel vaker minder ernstige thema's. Lembie, Taffer, Jeroen, Phil, Klaas, Dauwe, Violi, Angelos, Brecht, Matthias en *brave* Sir Robin -het is een eer en genoegen geweest pocketspits te zijn van dit team!

Ik prijs mezelf gelukkig te kunnen steunen op een ongelofelijke **vriendenkring**.

Aan onze bende van het middelbaar: we go way back! Hoewel de momenten waarop we allemaal present tekenen schaars zijn geworden, betekenen ze nog altijd veel voor mij. Bedankt voor de steun en ook om me erop te wijzen wanneer de balans te weinig naar jullie overhelde. Omwille van dergelijke eerlijkheid koester ik onze vriendschap enorm. Merci Elisah en Fortunato, Jolien D.B., Roos en Jonathan, Jolien J. en Filip, Karen en Filip! Een zeer bijzonder woord van dank draag ik op aan Thomas, die op korte tijd zoveel van zichzelf heeft geschonken.
Ook mijn studieperiode aan de universiteit heeft me in contact gebracht met mensen die me na aan het hart liggen. Niels, Prilsken en Dauwe: ik wil jullie enorm bedanken voor de mooie reizen die we samen hebben gemaakt en waar ik prachtige herinneringen

aan overhoud. Hopelijk slagen we er nog eens in samen op reis of citytrip te gaan. Fré en Lisanne: ik bewonder hoe jullie, ondanks de drukke agenda, steeds klaarstaan voor jullie vrienden. Freddy, nogmaals bedankt voor het informatieve jobgesprek! Napoleon, bedankt voor de warme, edoch harige vriendschap die je me hebt geschonken. De eerlijkheid gebiedt me wel te vermelden dat ik nog gelukkiger ben met jouw baasjes, Laurence en David. Het is ook altijd feest als Polle en Jonas en San en Bram van de partij zijn! Bedankt om steeds aangenaam, boeiend en positief gezelschap te zijn! Ook bedankt aan Eulalie voor de diepgaande gesprekken alsook de gelegenheid om even te kunnen ventileren. Uiteraard dien ik ook mijn dierbare Buddy Eva te vermelden. Samen een thesis schrijven heeft in ons geval een diepe band gesmeed. Bedankt ook voor de afterwork aperitieven en om op een blauwe maandag met random, maar extreem hilarische statements uit de hoek te komen! Evenzeer bedankt aan Joost, ongetwijfeld een van de meest intelligente mensen die ik ken.
Ik ben extreem gelukkig en vereerd om Kim, Marie, Dieter en Tiffany tot mijn vriendenkring te mogen rekenen. We brachten veel tijd door tijdens de groepswerken voor onze studies, maar ik ben zeer blij dat we ook nu nog steeds op regelmatige basis contact houden. Laat ons de maandelijkse bijeenkomsten nog vele jaren verderzetten! Dieter, bedankt om samen met mij zo'n geoliede groepswerkmachine te vormen!

Uiteraard wens ik ook mijn **familie** te bedanken. Zovelen onder jullie vroegen direct of indirect naar "hoe het met mijn doctoraat ging". Zelden kon ik daar een afdoend antwoord op geven en moest ik me noodgedwongen beroepen op "het gaat, dank u". Inmiddels heb ik een aantal jaren nagedacht over deze vraag en ben ik dan ook trots te melden dat het zeer goed gaat met mijn doctoraat.
Aan de leden van de kaartersclub, tante Wella, nonkel Walter en mamie, bedankt voor jullie steun en de bijwijlen geanimeerde zondagse taferelen. Daarnaast hebben jullie mijn kennis der Nederlandse taal uitgebreid met levenswijsheden als "het eerste gewin is kattengespin" en "een tegenslag is ook een slag".
Mamie, ik kan jou en papie niet genoeg bedanken voor alles wat jullie voor mij hebben gedaan. Ik ben onmachtig om in woorden uit te drukken hoe dankbaar ik ben. Met opperste bewondering kijk ik naar hoeveel liefde je iedereen schenkt en over welke onmetelijke energie jij beschikt. Bedankt voor de jarenlange steun, ook aan papie, die ongetwijfeld over jouw schouder meekijkt.
Broer, bedankt voor de talrijke leuke momenten die we de afgelopen jaren samen hebben beleefd. Bij elke taalkundige kwestie weet je raad en ook voor het nalezen van dit doctoraat kon ik op je rekenen. Dankzij jou ben ik niet alleen een vitter, maar heb ik als geen

ander leren improviseren. Hoewel velen mijn gevoel voor humor als “flauw” bestempelen (ik vermoed dat dit ontspruit aan jaloezie!), ben ik blij dat we hetzelfde *fijne* gevoel voor humor delen. Bedankt voor alles!
Papa, mama. Er bestaan geen woorden om jullie te bedanken. Van vieren tot ventileren, ik kan met mijn volledige scala aan emoties bij jullie terecht. Zelfs als ik zelf twijfelde aan “hoe ver ik stond”, jullie twijfelden nooit aan mij en bewezen steeds opnieuw dat de liefde van een ouder onvoorwaardelijk is. Bedankt om mijn rotsen in de branding te zijn!

Annelies, ook al leerde ik je pas aan het einde van mijn doctoraatstraject kennen, het is ongelofelijk hoeveel jij voor mij betekent. Ik wil je dan ook uit de grond van mijn hart bedanken, niet alleen voor de hulp bij dit doctoraat, maar ook om van mij een completer persoon te maken. Ik blik dan ook vol enthousiasme vooruit naar een nieuw avontuur met jou aan mijn zijde!

Mathieu Wauters
Gent, 18 september 2015

Samenvatting

Operationeel onderzoek is een onderzoeksdiscipline die gewijd is aan het vinden van (quasi-)optimale oplossingen voor complexe problemen. Wiskundige modellen, statistische methodes en het ontwerp van algoritmes worden aangewend om dergelijke oplossingen te vinden en te evalueren. Operationeel onderzoek wordt vaak in één adem genoemd met Management Science om aan te geven dat de oplossingsmethodes het management moeten ondersteunen bij het nemen van beslissingen. Zo zijn het opstellen van een evenwichtig personeelsplan, het plannen van productiesystemen en het zoeken naar de beste locatie om een nieuwe fabriek te openen maar enkele voorbeelden waar academici zich het hoofd over breken.

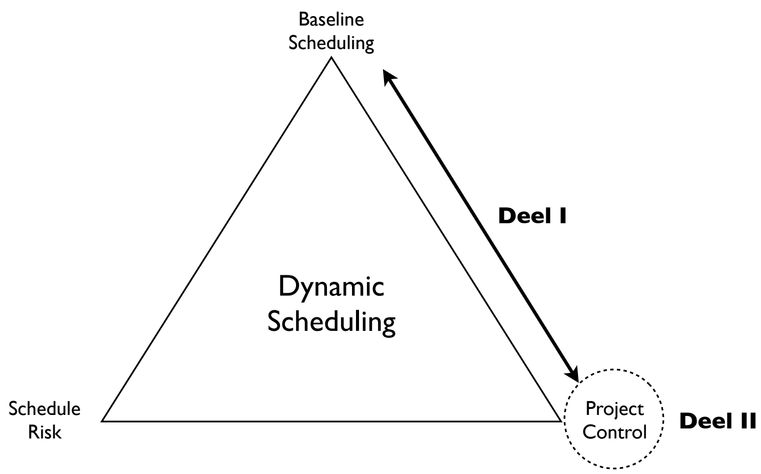
Projectmanagement kan worden beschouwd als een deeldomein van operationeel onderzoek. Hoewel vele definities worden gehanteerd die ofwel de nadruk leggen op softe aspecten zoals teammotivatie en leiderschap ofwel op meer technische definities, volgen we hier de definitie van Vanhoucke (2012b). Hij stelt dat projectmanagement de discipline is van het plannen, organiseren en managen van hulpmiddelen om specifieke doelstellingen succesvol tot stand te brengen. Sedert enkele jaren wordt het raamwerk van *dynamic scheduling* gebruikt om verschillende problemen binnen projectmanagement te kaderen (zie figuur 1). Dynamic scheduling omvat drie componenten, namelijk baseline scheduling, schedule risk en project control. Voor aanvang van een project (bijvoorbeeld het ontwikkelen van een iPhone-applicatie[1] of het bouwen van een huis) wordt een gedetailleerd plan opgesteld dat in grote lijnen weergeeft welke activiteiten moeten plaatsvinden, in welke volgorde en wat de geschatte tijdsduur en kost zal zijn. Baseline scheduling bundelt het onderzoek dat zich bezighoudt met het opstellen van een dergelijk plan. Eens deze fase voltooid is, zal een projectmanager op zoek gaan naar de voornaamste risico's. Welke activiteiten lopen het meest gevaar en zullen bijgevolg een grote impact hebben op de totale duurtijd en kosten van het project? Bij de laatste component wordt het plan omgezet in werkelijkheid. Het project wordt uitgevoerd en

[1] `http://www.or-as.be/orastalks`

periodiek opgevolgd. Indien de doelstellingen onder druk komen te staan, wordt een waarschuwingssignaal gegenereerd waardoor de projectmanager zal ingrijpen en pogen om de werkelijkheid met het vooraf opgestelde plan te verzoenen.

Data mining omvat een (semi-)automatisch proces om nieuwe en nuttige informatie te onttrekken aan grote databases (Olafsson et al. (2008)). Deze schat aan data kan afkomstig zijn van eerdere projecten die het bedrijf heeft uitgevoerd of kan verkregen worden door middel van computersimulaties. Data mining ligt op de grens tussen statistiek en toegepaste wiskunde en wordt verder opgesplitst naargelang het onderzoeksdoel en de aard van de data (supervised versus unsupervised learning).

Hoewel de onderzoeksgemeenschap operationeel onderzoek en data mining als aparte entiteiten beschouwt, vond reeds heel wat kruisbestuiving tussen beide velden plaats. Hierbij werden technieken uit operationeel onderzoek geïntroduceerd in data mining en werd operationeel onderzoek verrijkt met beslissingstechnieken uit data mining. Het onderzoek dat we in dit doctoraat presenteren, bevindt zich op de interface tussen operationeel onderzoek, projectmanagement en data mining. We gaan met andere woorden op zoek naar wat we uit (echte of gesimuleerde) data kunnen leren binnen een projectmanagementomgeving. Het doctoraat bestaat uit twee delen, die hieronder kort worden besproken. Figuur 1 biedt een schematisch overzicht.



Figuur 1: Componenten van Dynamic Scheduling en focus van dit doctoraat

Deel 1: Time/cost optimization In het eerste deel van dit doctoraat bestuderen we het Discrete Time/Cost Trade-off Problem (DTCTP). Dit probleem stelt dat elke activiteit van een project op meerdere manieren kan uitgevoerd worden. Hierbij vindt een afweging plaats tussen de benodigde tijd en de kosten om deze activiteit te voltooien. De projecten die we bestuderen, hebben bovendien een contractuele deadline. Indien deze deadline overschreden is, wordt een dwangsom toegewezen per dag dat het project te laat eindigt. Op die manier moet een afweging gemaakt worden tussen tijd en kosten, wat de titel van Deel 1, Time/cost optimization, verklaart. De motivatie voor dit onderzoek ontspruit aan een business game, genaamd het Project Scheduling Game (PSG), dat in verschillende universiteiten en business schools voor educatieve doeleinden wordt gebruikt. Participanten nemen de rol van projectmanager op zich en hebben zeggenschap over een project dat uit de praktijk werd geplukt. Net zoals in het echte leven zijn de activiteiten van het project onderworpen aan onzekerheid. De projectmanagers kunnen op zes verschillende tijdstippen ingrijpen en de duurtijden van de verschillende activiteiten aanpassen om de doelstellingen van het project (kostenminimalisatie) te vrijwaren. Dit verklaart waarom deel 1 zich op de grens tussen Baseline Scheduling en Project Control bevindt (zie figuur 1). Deel 1 van het doctoraat bestaat uit twee hoofdstukken die een antwoord trachten te bieden op de volgende Onderzoeksvraag (O):

> $\mathbf{O}_1$: hoe kunnen onderzoek en praktijk van het DTCTP op elkaar worden afgestemd?

Hieronder volgt een korte beschrijving van de bijdrage van beide hoofdstukken.

- In hoofdstuk 2 worden oplossingsstrategieën met betrekking tot het PSG geanalyseerd. Gedurende verschillende jaren werden data van honderden participanten verzameld waaruit vervolgens twee strategieën werden gedistilleerd. Elke strategie omvat vijf componenten die automatisch of door feedbacksessies werden opgesteld. Uniek aan de aanpak van dit hoofdstuk is dat we softe aspecten, namelijk hoe omgevingsvariabelen als onzekerheid en complexiteit worden gepercipieerd, combineren met kwantitatieve aspecten. De bijdrage van dit hoofdstuk tot $\mathbf{O}_1$ ligt in het opnemen van complexiteit en onzekerheid als contextuele factoren en door de discrepantie tussen perceptie en realiteit expliciet te erkennen (Crawford et al. (2006)).

- Hoofdstuk 3 vormt een uitbreiding op het vorige hoofdstuk door middel van de invoering van een energierestrictie. Tijd, geld en de inzet van personeel worden allen aanzien als voorbeelden van een zekere inzet aan energie. In dit hoofdstuk wordt de participanten van deze uitbreiding op het PSG, genaamd PSG Extended, een aantal strategische componenten voorgeschoteld. In tegenstelling tot hoofdstuk 2 is de feedback minder dynamisch. Er wordt enkel weergegeven wat de cumulatieve energie is die men heeft gespendeerd, zonder enige info over tijd- en kostenperformantie te communiceren. Op die manier worden deelnemers gedwongen om aan het grotere geheel te denken en strategische keuzes te maken in plaats van een proces van trial-and-error aan te vatten. PSG Extended werd uitgerold in januari 2014 aan het University College of London (UK). De eerste resultaten stellen ons in staat om een empirische evaluatie te maken. Bovendien hebben we op een gestructureerde wijze oplossingen gegenereerd. We rapporteren de resultaten van de empirische en computationele experimenten en wijden aandacht aan de invloed van verschillende parameters en de strategische componenten. Door het invoeren van een energierestrictie voeren we een real-life beperking toe aan het bestaande onderzoek en dragen we zo bij tot het overbruggen van de kloof tussen onderzoek en praktijk zoals geformuleerd in $\mathbf{O}_1$.

Deel 2: Forecasting In het tweede deel van dit doctoraat concentreren we ons op projectcontrole. Het plan of baseline schedule wordt als gegeven beschouwd. De uitvoering van een project brengt verschillende uitdagingen met zich mee. Idealiter controleert men de voortgang van elke activiteit en grijpt men in zodra zich een ernstig probleem met een kritieke activiteit voordoet. In de praktijk blijkt een dergelijke aanpak op activiteitsniveau (bottom-up approach) niet haalbaar. Een alternatieve aanpak aggregeert de performantie van individuele activiteiten op een hoger niveau, bijvoorbeeld het projectniveau. Dit heeft als voordeel dat de projectmanager in een oogopslag een beeld krijgt van de voortgang van het project. Het gevaar is echter dat de negatieve voortgang van zeer belangrijke activiteiten wordt gemaskeerd door de positieve evolutie van andere activiteiten. Vermits men op een hoger niveau rapporteert, wordt dit uitgemiddeld en ontsnappen potentiële gevaren aan de aandacht van de projectmanager. Toch houden we vol dat deze projectaanpak (top-down approach) de meest pragmatische uitweg biedt. De top-down methodologie bij uitstek staat gekend als Earned Value Management (EVM). In deel 2 gaan we met deze methodologie aan de slag en spitsen we ons toe op het forecastingprobleem. Gegeven de voortgang van een project en de beschikbaarheid van (echte of gesimuleerde) historische data, gaan we op zoek naar methodes die stabiele en accurate voorspellingen maken. Dit deel bestaat uit drie hoofdstukken, waarbij de volgende onderzoeksvraag centraal staat:

$\mathbf{O}_2$: hoe kunnen historische data gebruikt worden om de voorspellingskwaliteit te verbeteren?

De voorspellingskwaliteit van een gegeven methode bestaat uit stabiliteit en accuraatheid. De inhoud van elk hoofdstuk van deel II is als volgt:

- In hoofdstuk 4 nemen we de stabiliteit van voorspellingsmethodes onder de loep. Een voorspellingsmethode wordt als stabiel ervaren wanneer opeenvolgende voorspellingen niet drastisch afwijken ten opzichte van elkaar. In dit hoofdstuk wordt een kritische noot bij de vigerende stabiliteitsmaatstaf geplaatst en stellen we een alternatief voor. De stabiliteit van de bestaande EVM-voorspellingsmethodes wordt getest door middel van een computationeel experiment, gebruikmakende van een topologisch diverse dataset. Zowel de nauwkeurigheid als de stabiliteit van de methodes wordt gerapporteerd, wat de afweging tussen beide criteria faciliteert. De stabiliteit wordt tevens getest aan de hand van twee real-life projecten, waarbij de bevindingen van het computationele experiment grotendeels worden bevestigd. Dit hoofdstuk vormt de basis voor hoofdstuk 6, waarin de stabiliteit van de voorgestelde methodes van hoofdstuk 5 wordt gemeten. Dit hoofdstuk houdt verband met $\mathbf{O}_2$ door stabiliteit als onderdeel van voorspellingskwaliteit te onderzoeken en hiervoor een nieuwe maatstaf naar voren te schuiven.

- In hoofdstuk 5 beschouwen we de nauwkeurigheid van voorspellingsmethodes. Eerder onderzoek richtte zich op de nauwkeurigheid van EVM-methodes. In dit hoofdstuk introduceren we een nieuwe familie methodes, die hun oorsprong vinden in het domein van Artificiële Intelligentie (AI). Deze technieken hanteren historische data om een bepaald verband (in dit geval de relatie tussen EVM-maatstaven en voorspellingskracht) te leren. Vervolgens wordt het model dat hierbij wordt geconstrueerd toegepast op nieuwe data. Methodologisch voegen we eveneens technieken als pre-processing, grid search en cross-validation toe. Door middel van een sensitiviteitsanalyse worden de beperkingen van de voorgestelde methodes geduid. Dit hoofdstuk toont de kracht aan van het gebruik van historische data (zie $\mathbf{O}_2$), gegeven dat de inputgegevens voldoende gelijkaardig zijn aan de werkelijke voortgang van het project.

- In het laatste hoofdstuk van deel 2, hoofdstuk 6, integreren we aspecten van de twee voorgaande hoofdstukken. Enerzijds gaan we na hoe goed de AI-methodes op het vlak van stabiliteit scoren. Anderzijds breiden we de AI-methodes uit aan de hand van de Nearest Neighbour-techniek. Het doel van deze techniek is tweeërlei. Ten eerste kan de Nearest Neighbour-methode gebruikt worden om voorspellingen te maken en kunnen we de nauwkeurigheid en stabiliteit vergelijken met de predictieve methodes uit hoofdstukken 4 en 5. Ten tweede wordt de Nearest Neighbour-techniek als een nieuwe methodologische stap ingevoegd. Hierbij wordt de grootte van de training set beperkt. Deze reductie wordt echter gecompenseerd door een kwaliteitsverbetering: enkel de meest gelijkaardige observaties worden behouden. Omwille van consistentie voeren we tests uit op data die grotendeels identiek zijn aan die van hoofdstuk 5. Dit hoofdstuk kan beschouwd worden als een culminatie van het onderzoek naar het gebruik van historische data voor beide dimensies van voorspellingskwaliteit, namelijk stabiliteit en nauwkeurigheid. Daarom geloven we dat dit hoofdstuk perfect de voor- en nadelen aantoont van het gebruik van historische data en een genuanceerd antwoord biedt op $\mathbf{O}_2$.

Uit het gevoerde onderzoek vloeiden zes papers voort die werden gebundeld in vijf hoofdstukken. Hoofdstuk 7 rondt dit doctoraat af door conclusies en beperkingen van beide delen te belichten. Daarnaast worden richtlijnen voor verder onderzoek gedefinieerd in de hoop dat andere onderzoekers evenveel plezier ervaren als wij tijdens deze intensieve onderzoeksperiode.

Table of Contents

List of Figures

List of Tables

1
Introduction

1.1 General introduction

In this book, you can find the culmination of research that we conducted in the fields of project scheduling and control. In order to understand the contribution of our research, it is necessary to outline the general research fields in which we operated. The studies contained in chapters 2-6 of this book combine project management problems with data mining solution approaches. In this section, the reader is introduced to the fields of Operations Research and Data Mining, which will then be refined to delineate the scope of this book. Our contribution, as well as a detailed overview of the different chapters, is provided in section 1.2.

Operations Research Operations Research (OR) is a research discipline that is devoted to solving complex problems to (near-)optimality. In order to solve these problems, mathematical models, statistics, as well as algorithmic design are involved. OR is often named in conjunction with Management Science to indicate that the ultimate goal is to aid management in making decisions. Finding the best locations for new plants, personnel staffing or deployment of vehicle fleets are all example problems in which Operations Research can facilitate the complex decisions management needs to take. The reader is referred to Hillier and Lieberman (2005) for an introduction to OR.

Project Management Project management is a discipline that pairs quantitative techniques with more qualitative influences from the psychological and human resources realm. It primarily gained momentum when project planning approaches such as the Critical Path Method (Kelley and Walker (1959), Walker and Sawyer (1959) and Kelley (1961)) were conceived in the 1950s. Many animated discussions revolve around the definition of a project, which can range from very technical explanations to descriptions that put more emphasis on the human aspects. As an example, Tavares and Weglarz (1990) define a project as "any purposeful transformation leading a system, Ω, from an initial state, s, to a specific state, s' and so s' should represent the targets to be achieved." Contrary to this system-thinking approach, other authors take a soft paradigm approach (Pollack (2007)) and include topics rooted in psychology such as leadership styles (Müller and Turner (2007)) and motivation (Schmid and Adams (2008)). Vanhoucke (2012b) reconciles the hard and soft aspects of the definition of a project, as follows:

> "Project management is the discipline of planning, organizing and managing resources to bring about the successful completion of specific project goals and objectives."

This definition leaves room for managing human resources, focusing on soft skills and psychological effects, as well as on mathematical models and procedures. Within organizations, project managers are confronted with many projects which compete for limited resources and require different degrees of attention, depending on the phase of the project under study. The Project Management Book of Knowledge (PMBOK, PMBOK (2004)) discerns six phases every project goes through. These are illustrated in figure 1.1. The six phases will briefly be discussed along the following lines:

- Concept phase: the need for a project arises, at a client's request or through the company's internal processes.
- Definition phase: the goals, scope and technical requirements of the project are charted. The Work Breakdown Structure (WBS), along with the project activities and their precedence relations are identified.
- Scheduling: a timing of the different activities is made. Factors that can greatly influence the project schedule are, among others, the project objective and the presence of resources.
- Execution and control phase: part of the project is executed and monitored throughout the control phase. Early warning signals indicate whether the project is still on track and does not violate the normal variation. Once the project objectives are endangered, a warning signal triggers the need for corrective action. This is represented by means of the feedback loop from the control phase to the scheduling phase.
- Termination phase: upon completion, the project is evaluated by internal (the company) and external (the client) stakeholders.

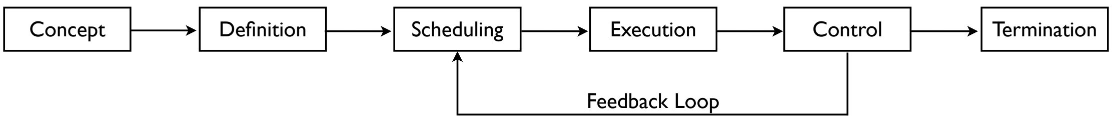


Figure 1.1: The six phases of a project lifecycle (PMBOK (2004))

While the project lifecycle provides a complete overview of the project's conception until it is terminated, the six distinct phases fail to fully grasp the dynamic nature of scheduling and control. Even though the iron triangle (time, cost and scope) has received a lot of attention (Crawford et al. (2006)), Perminova et al. (2008) advocate a larger emphasis on uncertainty and pinpoint the need for greater flexibility. The framework of *Dynamic Scheduling* (Uyttewael (2005), Vanhoucke (2012b)) responds to this call for a more dynamic interpretation of important project management aspects. While it does not cover the full lifecycle from project inception until termination, it focuses on three important dimensions that are dynamic in nature and can be interrelated. The three dimensions, baseline scheduling, Schedule Risk Analysis (SRA) and project control, can be found in figure 1.2 and are briefly explained as follows:

- Baseline scheduling: the construction of the timing of a project's activities has received significant attention from the research community. Temporal constraints, resource constraints, activity concepts and project objectives all contribute to the complexity of solving scheduling problems. An overview of the literature on resource-constrained project scheduling can be found in the reviews of Herroelen et al. (1998), Brucker et al. (1999) and Hartmann and Briskorn (2010). The importance of the scheduling phase is widely recognized and apparent from a survey of White and Fortune (2002). The authors reported that a realistic schedule is among the top factors that are believed to be most critical to a project's outcome.

- Schedule Risk Analysis: risk analysis allows for an identification of a schedule's weak points and permits project managers to prioritize their attention to a subset of the most sensitive activities. Risk can be very loosely defined as probability times impact. Multiple metrics exist to capture the degree of risk exposure on the activity level. Because of its activity level focus, SRA is often named a bottom-up technique. The reader is referred to Vanhoucke (2015) for an overview of the use of SRA for the Project Management discipline.

- Project control: while constructing a baseline schedule is an integral part of project management, its relevance should be weighed against the project control phase. Throughout a project's execution, the baseline schedule acts as a point of reference, to compare the current time, cost, quality and scope with what was expected before the project was initiated. Often, project control is inextricably linked with Earned Value Management (EVM), a methodology that originated at the US Department of Defense in the 1960s. EVM relies on three key metrics, Planned Value (PV), Earned Value (EV) and Actual Cost (AC), to provide a quantitative indication of a project's health. Contrary to SRA, the activity level information is aggregated to a higher level of the WBS. Effort-wise it is often infeasible to monitor every single activity which explains why project level information is communicated. This has been the subject of criticism (see e.g. Book (2006a,b) and Jacob and Kane

(2004)), yet we follow the rationale of Vanhoucke (2010a), who claims that an activity-level control approach lacks realism for any project of moderate size. An introductory overview of EVM can be found in Vanhoucke (2014). The reader can find a literature overview on project control and EVM in Willems and Vanhoucke (2015).

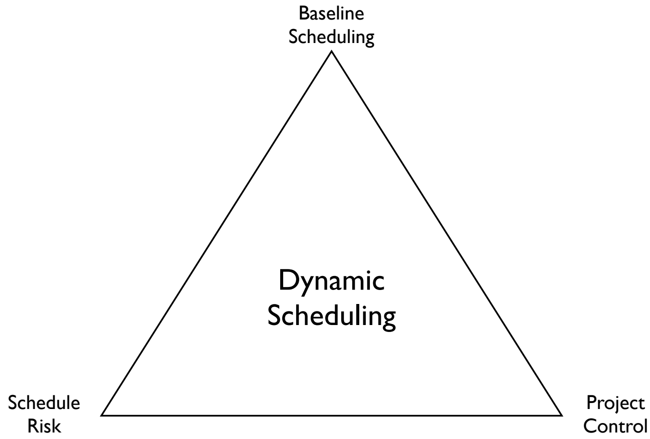


Figure 1.2: The three dimensions of dynamic scheduling (Vanhoucke (2012b))

The next phase for dynamic scheduling is to combine the three dimensions to form an integrated project management and control system. Scheduling methods should take uncertainty a priori into account. The weak spots need to be identified and controlled. Throughout the execution phase, early warning signals can trigger the need for corrective action which takes project managers back to the (re)scheduling phase. Combining these research topics into one Decision Support System (DSS) is undoubtedly a major challenge for the future. Hazır (2015) stresses the importance of integrating the various approaches and analytical models into a DSS, which can then be integrated into existing project management software.

Data Mining Similar to Operations Research, substantial debate surrounds a conclusive definition of Data Mining. Despite earlier contributions (e.g. Mangasarian (1965)), the field of Data Mining as it is known today dates back to the 1990s (Meisel and Mattfeld (2010)). Throughout this dissertation, the term Data Mining will be used to refer to the aspects of a (semi-)automated process to extract previously unknown and potentially useful knowledge from large databases (Olafsson et al. (2008)). Informally, Data Mining can be thought of as learning from data. These data may result from real-life data sources or computational experiments such as simulations. Data Mining lies at the intersection of statistics and applied mathematics and a subdivision is often made based on whether data is labeled (supervised versus unsupervised learning) or on the goal for which data mining is employed (classification, regression, association rule mining). Comprehensive introductions to this multi-disciplinary research field can be found in Tan et al. (2006) and Hastie et al. (2009).

While Operations Research and Data Mining are seen as distinct research fields, there has been cross-pollination leading to synergies between both fields. Support Vector Machines lie at the intersection of optimization and data mining (Vapnik and Lerner (1963)). Meta-heuristics have found their entry in data mining literature to tune parameters for various Artificial Intelligence (AI) methods. Data Mining can also aid further development of OR. Corne et al. (2012) list 3 manners in which Data Mining can improve Operations Research algorithms, namely by increasing the quality of the results, speeding up OR algorithms and employing Data Mining to select an algorithm based on properties of the instance under study. A more exhaustive overview of the synergies between Operations Research and Data Mining can be found in the works of Corne et al. (2012), Meisel and Mattfeld (2010) and Olafsson et al. (2008).

A graphical overview of this section and the focus of this PhD is provided in figure 1.3. The research fields of Operations Research and Data Mining were briefly introduced. More attention was given to the main discipline of this book, namely Project Management. While we outlined a couple of general synergies between OR and Data Mining, the work presented in this book focuses on the intersection of Project Management and Data Mining and Project Management, Data Mining and Operations Research. This will be elucidated in section 1.2.

1.2 Research contribution

In the previous section, we established the research fields of Operations Research, the sub-discipline Project Management and Data Mining. In this section, we will dive into the intersection between Project Management and Data Mining and Project Management, Data Mining and Operations Research (cf. figure 1.3) and outline the research we have done. All chapters of this book revolve around learning from data for project management problems. A distinction is made along the Dynamic Scheduling components which were discussed in section 1.1. A graphical representation is provided in figure 1.4.

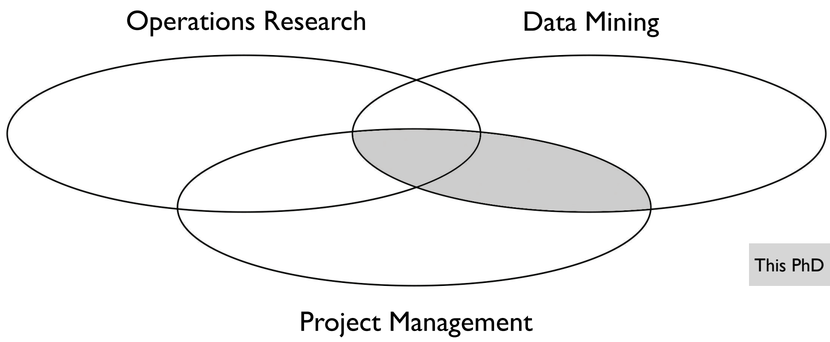


Figure 1.3: Focus of this PhD

Part I focuses on the interplay of the Baseline Scheduling and Project Control step. We study solution strategies of a Project Management business game, in which the baseline schedule constitutes the starting point. Part of the project is executed and based on the feedback, it is possible to make changes to the remainder of the project. The second part exclusively concentrates on Project Control by means of the Earned Value Management methodology. More specifically, the chapters of Part II deal with making predictions by exploiting progress data and historical information.

Part I and part II should be regarded in light of a Decision Support System. As a project is in progress, it is monitored by means of a technique that can detect abnormal variation. This provides the project manager with a trigger for corrective action. A drill-down into the Work Breakdown Structure will take place, in which a set of activities will be identified for which corrective actions may be executed. These actions may consist of trade-off changes that are taken into account. Hence, the trade-off changes of part I can be embedded in the project monitoring and control environment of part II of this dissertation.

Each part of this book will be introduced in more detail. Section 1.2.1 introduces the chapters that deal with time/cost optimization while section 1.2.2 elaborates on project control forecasting. It is worth mentioning that a literature review is contained in each chapter. Hence, we will not provide an exhaustive overview in this introduction but choose to highlight the gap in literature that is filled by our contributions.

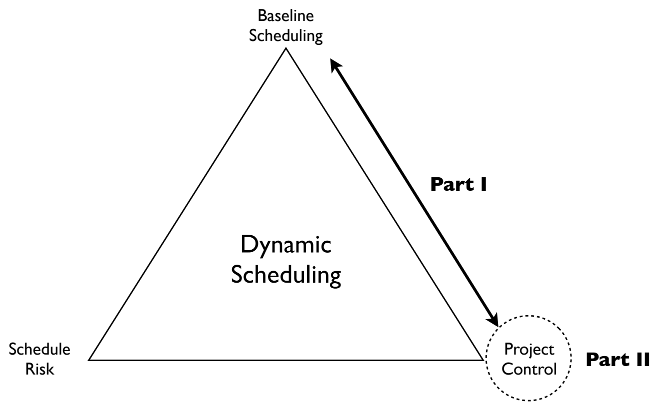
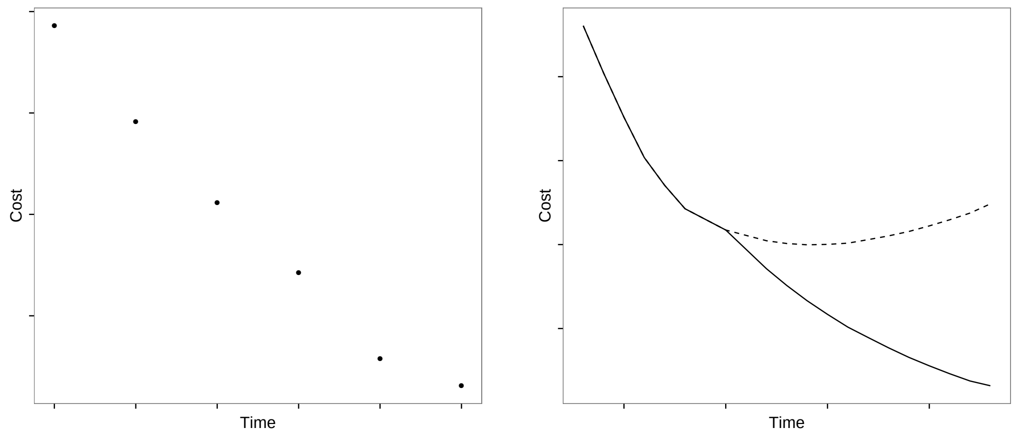


Figure 1.4: Focus of this PhD in relation to the Dynamic Scheduling components

1.2.1 Part I: Time/cost optimization

Part I of this book comprises two chapters that deal with solution strategies for a project management problem that is known as the Discrete Time/Cost Trade-off Problem (DTCTP). The main characteristic of the DTCTP consists of multiple ways in which each project activity can be executed. A trade-off between the duration (time) and cost of each activity exists, where an activity's duration is a discrete, non-increasing function of the amount of money allocated to it. It is worth noting that the work content (the amount of resources times the period of time in which these resources are used) is not necessarily fixed. Coordination problems and fixed costs (e.g. hiring costs) cause shorter durations to come at a higher expense. The project manager needs to make a trade-off choice for every activity. An example of a time/cost profile for one activity is given in figure 1.5(a). The x-axis represents the time dimension, while the y-axis refers to the costs. The graph clearly shows that there is a discrete number of time/cost combinations. However, when the time/cost trade-offs for all project activities are translated to the project level, a continuous line, as depicted in figure 1.5(b), appears. This is due to the fact that different activity trade-off combinations give rise to a large number of project outcomes. The solid line in figure 1.5(b) shows that as the duration of the project is prolonged, the costs decrease. The dashed line depicts the influence of the presence of a penalty cost. A cost is incurred for every day the project duration exceeds the deadline. As a result, the cost curve is no longer monotonically decreasing. From a certain point onwards, the decrease in activity costs no longer outweighs the increase in penalty costs, causing the total costs to rise.



(a) Activity (b) Project

Figure 1.5: Time/cost profile for a single activity (figure 1.5(a)) and the project as a whole (figure 1.5(b))

Literature discerns three variants of the DTCTP. The budget problem (DTCTP-B) aims to minimize the project duration within a limited budget. The deadline problem (DTCTP-D) tries to find the minimal project costs while respecting a predefined deadline δ. Finally, the third variant generates the complete efficient time/cost frontier. The scope of part I is limited to the deadline variant of the DTCTP which can be modeled as follows.

Model formulation A project, consisting of n activities can be represented in an Activity-on-the-Node (AoN) format which specifies the precedence relations between activities. The precedence graph, $G = (N, A)$, contains a set of nodes N and a set of arcs A. N corresponds with the activities in addition to two dummy activities to represent the start and end of the project, respectively. A represents the precedence relations between activities. The time/cost trade-off problem is characterized by multiple trade-off combinations which are often referred to as modes. Let M_j represent the set of modes for each activity j. Each mode m possesses a duration d_{jm} and a cost c_{jm}. The DTCTP-D can be expressed as a mixed-integer programming model (Hazır et al. (2011)).

$$\min \quad \sum_{j=1}^{n} \sum_{m \in M_j} c_{jm} x_{jm} \tag{1.1}$$

Subject to

$$\sum_{m \in M_j} x_{jm} = 1 \qquad j = 1, ..., n \tag{1.2}$$

$$C_i + \sum_{m \in M_j} d_{jm} x_{jm} \leq C_j \qquad \forall (i,j) \in A \tag{1.3}$$

$$C_{n+1} \leq \delta \tag{1.4}$$

$$C_j \geq 0 \qquad \forall j \in N \tag{1.5}$$

$$x_{jm} \in \{0,1\} \qquad \forall m \in M_j, j = 1, ..., n \tag{1.6}$$

The DTCTP was shown to be NP-Hard (De et al. (1997)) and can be solved using exact and heuristic solution approaches (De et al. (1995)). Since the mid 2000s, research efforts focused on two tracks, namely the consideration of extensions (Vanhoucke and Debels (2007)) or including stochastic characteristics (Hazır et al. (2011, 2010)). A more thorough literature review is provided in sections 2.1 and 3.1 of this book.

The approach of part I differs from the recent trend of studying extensions or including stochastic characteristics. The main motivation for studying the DTCTP-D stems from a project management business game entitled the Project Scheduling Game (PSG, Vanhoucke et al. (2005)). In this game, participants work with a real life project and have to schedule and reschedule a project. Students receive updates on the status of the project and finished activities in a periodical fashion. Based on this feedback, they can change the durations of unfinished activities to bring the project back on track. The activities are subject to Murphy's law. While some activities finish sooner than anticipated, several activities face delays. The real life project that is used by the students corresponds with a relaxed version of the DTCTP-D. In this model, constraint (1.4) is not a hard constraint. Exceeding the deadline does not lead to infeasibility but results in a penalty. For every day the deadline is exceeded, a penalty cost of ϕ euros per day is added to the total project cost. Hence, the objective can be expressed as the minimization of $\sum_{j=1}^{n} \sum_{m \in M_j} c_{jm} x_{jm} + \phi * \max(0, C_{n+1} - \delta)$.

The PSG is part of the educational curriculum of several Project Management programmes at a Master or MBA level. The changes that a player makes throughout the game are recorded in a log file. The collection of log files of hundreds of participating students in the PSG at Ghent University (Ghent (Belgium)), University College London

(London (UK)), Vlerick Business School (Ghent, Leuven (Belgium)) and EDHEC Business School (Lille (France)) was the main inspiration for the first chapter of this book. Part 1 of this book, entitled Time/Cost Optimization, comprises two chapters. Both chapters aim to provide an answer to the following Research Question (RQ):

$\mathbf{RQ}_1$ How can research and practice of the DTCTP become more aligned?

This question was raised by Hall (2012) and weighs a one-size-fits-all approach against the contextual diversity managers face in real life. The two chapters that aim to bridge this gap are briefly explained below.

- Chapter 2 presents the paper "A study on complexity and uncertainty perception and solution strategies for the time/cost trade-off problem" (Wauters and Vanhoucke (2013)). Data from hundreds of student solutions of the PSG was distilled into two main strategies. Each strategy is composed of five components (focus, activity criticality, ranking, intensity and action), which were identified either automatically (from the available log files) or through feedback sessions and discussions with students. Unique in our approach to the DTCTP-D is the reconciliation of the hard and soft paradigms (Pollack (2007)). While the numeric information, expressed in terms of time and cost, is still essential, we accord explicit attention to concepts such as complexity, uncertainty and how these are perceived. The consequence of an incorrect appraisal of complexity and uncertainty on time and costs is examined in a large computational experiment. Consequently, we unify the solution strategies (hard paradigm) with perceptions of the external environment (soft paradigm). The contribution of this chapter to $\mathbf{RQ}_1$ lies in the inclusion of complexity and uncertainty as contextual factors and by recognizing the discrepancy between perception and reality (Crawford et al. (2006)).

- Chapter 3 extends the Project Scheduling Game by focusing on effort restrictions. Companies allocate limited amounts of funds to projects. Time, money and human resources can all be regarded as examples in which the effort that can be allocated is constrained. In this chapter, a number of strategic components are provided. These components allow people to construct custom solutions to the DTCTP-D. However, when the components are assembled and applied to a project, they are constrained by a limited amount of effort. The effect of this effort limitation as well as the settings of the individual components are tested empirically and computationally. The effort-based extension to the PSG is known as PSG Extended and was rolled out for educational purposes in January 2014. Participant results allow us to test their strategies on real-life and computer-generated projects. Additionally, strategies can be generated and tested by means of a computer, eliminating the need for student experiments. We report on the results of the empirical and computational data and demonstrate the influence of parameter settings and strategic components on the quality of the attained solutions. By imposing an effort restriction, which is common in real-life environments, this chapter aids in bringing research and practice closer to one another ($\mathbf{RQ}_1$).

A comparison between chapters 2 and 3 is provided in table 1.1. Both chapters embed real-life aspects into the Discrete Time/Cost Trade-off Problem. Chapter 2 assesses the quality of the derived solution strategies for different degrees of complexity and uncertainty. Chapter 3 limits the amount of available effort, measured by the number of trade-off changes that can take place. From a business game point of view, chapter 2 attained a high level of maturity, while PSG Extended, presented in chapter 3, recently saw the light. While both PSG and PSG Extended require that the user takes decisions for each of the 6 decision moments, the received feedback differs. As a participant of the PSG makes a change to an activity's trade-off, the project duration and cost is updated. Hence, this type of feedback allows for clever strategies or trial-and-error approaches. When the participant changes one of the settings of PSG Extended, the consumed effort is updated. No information is provided concerning the current time and cost. Hence, this forces participants to reflect strategically on the component changes that can be made. PSG consists of 5 building blocks, whereas PSG Extended contains 3 strategic components that can be subdivided into 7 elements. While there is very little overlap between the components of both chapters, it is worth noting that the focus of chapter 2 corresponds with the schedule focus of chapter 3 and that ranking in both chapters entails the application of a priority rule.

Characteristic	Chapter 2 PSG	Chapter 3 PSG Extended
Goal	• Assess solution strategy quality • Inclusion of contextual characteristics	• Influence of effort limitation on DTCTP characteristics • Validation of strategic components
Level of maturity	High	Low
Feedback	Dynamic Time & Cost	Static Effort consumption
Components	5 building blocks	3 strategic components (7 elements)
	Focus Activity criticality Ranking Intensity Action	Schedule focus Activity focus • Ranking • Time/cost focus Action radius • Deadline focus • Slack consumption • Cost/benefit analysis • Effort loading

Table 1.1: Comparison of chapter 2 and chapter 3

1.2.2 Part II: Forecasting

As mentioned in section 1.1 of this chapter, baseline scheduling has received a lot of attention from the research community. However, sooner or later the baseline schedule is turned into reality and execution may render the detailed scheduling exercise moot. As a result, the growing attention for project control within research circles is most welcome. Once the project has started, it is of vital performance to track the progress and health of the project, which is facilitated by Earned Value Management, one of the foremost methodologies for project control. Project execution raises several important research questions. When should a project manager take action? What is the expected final duration and cost? Which parts of the schedule should the project manager focus on? Our research group[1] has provided answers to these questions and advanced the frontiers of project control using EVM in multiple ways. Vandevoorde and Vanhoucke (2006) and Vanhoucke and Vandevoorde (2007) examined project duration forecasting methods on empirical and simulated data. Vanhoucke (2010b, 2011) compared a top-down approach (EVM) and bottom-up approach (SRA) and called for further integration of these two dynamic scheduling components. In this respect, Elshaer (2013) demonstrated the validity of this call by integrating sensitivity information into an Earned Schedule forecasting method. Colin and Vanhoucke (2014) defined the concept of statistical project control and examined the performance of project control charts. Batselier and Vanhoucke (2015) constructed a real-life database that facilitates the access of researchers to empirical data and can enrich and supplement their computational insights.

In the second part of this book, we turn our attention to project control forecasting. While previous research within our research group examined this problem, our research makes two distinct contributions. First of all, we are, to the best of our knowledge, the first authors to investigate forecasting stability within EVM. The work of Vandevoorde and Vanhoucke (2006) and Vanhoucke and Vandevoorde (2007) deals with forecasting accuracy, while EVM stability research examined performance metrics such as the Cost Performance Indicator (Christensen and Payne (1992), Christensen and Heise (1993), Henderson and Zwikael (2008)) instead of the stability of prediction methods. Secondly, by integrating Project Management and Data Mining, we introduce a new class of forecasting methods to the project control community. The performance of these methods is assessed by means of forecasting accuracy and stability. Before proceeding to the outline of the chapters of the second part of this book, an intuitive definition of stability and

[1] `http://www.projectmanagement.ugent.be`

accuracy will be provided.

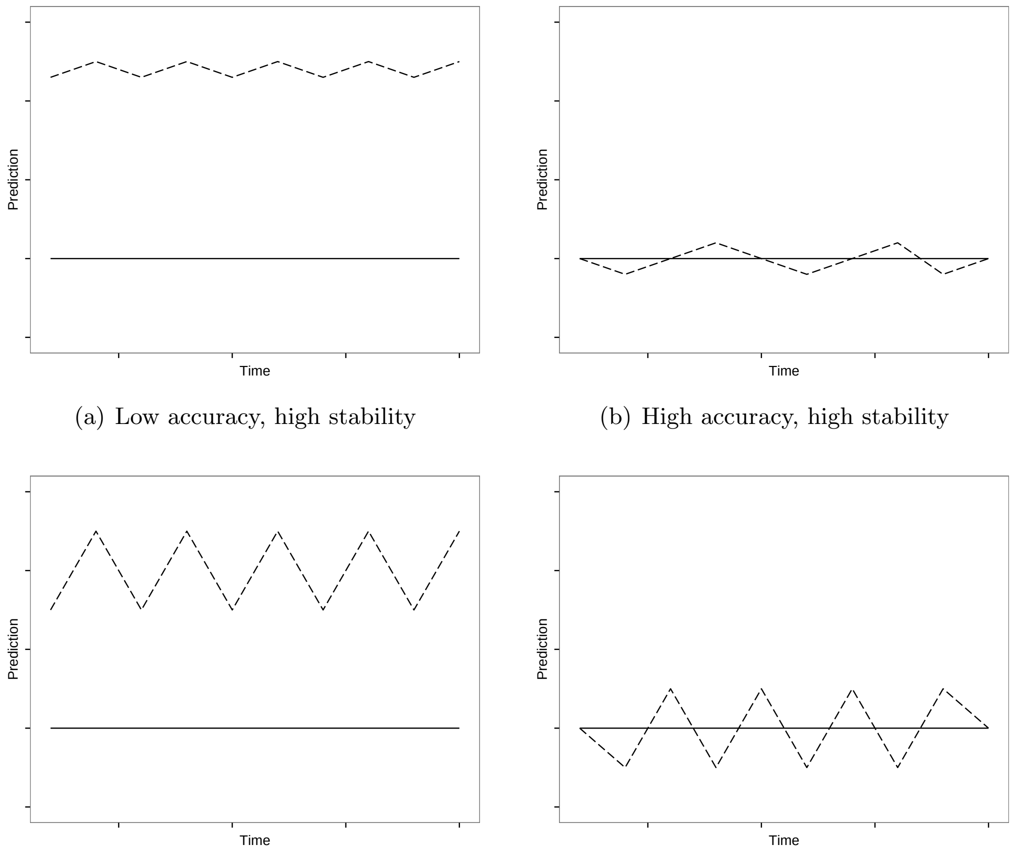
Stability & Accuracy Stability and accuracy are two dimensions for assessing the performance of a forecasting method. Ideally, a predictive method provides estimates that do not vary much from one point in time until the next and that lie close to the true value. Literature on EVM forecasting has been dominated by the accuracy perspective. However, it is not beyond reason to imagine that some project managers would prefer estimates that do not vary much (stable) but are slightly more erroneous (accurate). This train of thought applies when predictions are employed as a trigger for corrective action. Unstable but accurate methods will provide a project manager with mixed signals and leave one guessing what the correct course of action should be. Figures 1.6(a)-1.6(d) each depict two time series. The x-axis displays time and should be interpreted as the different points in time at which a prediction is made. The value of the prediction is reflected on the y-axis. The solid line corresponds with the true value in all graphs. The dashed line presents the predicted value across time. Figures 1.6(a) and 1.6(b) display situations with a high degree of stability. The subsequent predictions vary only slightly. Figure 1.6(b) pairs the high degree of stability with a high accuracy since the predicted value lies close to the real value. Figures 1.6(c) and 1.6(d) depict situations in which the stability is low. There is substantial variation in the predictions when moving from one point to another along the x-axis.

Part II of this book consists of three chapters. These chapters revolve around the following research question:

> $\mathbf{RQ}_2$ How can historical data be leveraged to improve forecasting quality?

The quality of forecasting consists of accuracy and stability. The outline of the chapters of Part II is as follows:

- Chapter 4 presents the results of the paper entitled "Study of the Stability of Earned Value Management" (Wauters and Vanhoucke (2015)). This chapter commences with a critical analysis of the stability metric that was used in previous stability studies. A new criterion based on the Mean Lags is put forward. A computational experiment with a topologically rich dataset gauges the stability of the existing time and cost forecasting methods. Forecasting accuracy is reported in order to facilitate a trade-off between accuracy and stability. As mentioned previously, there are certain circumstances in which a project manager may prefer a more stable forecasting method instead of an accurate one. By reporting accuracy



(a) Low accuracy, high stability (b) High accuracy, high stability

(c) Low accuracy, low stability (d) High accuracy, low stability

Figure 1.6: Intuitive explanation of forecasting quality: stability and accuracy

and stability results, we facilitate the choice for a certain predictive method. The novel stability metric is tested on two real-life projects, in which the computational conclusions are largely corroborated. This chapter lays the groundwork for chapter 6, in which the stability of the methods of chapter 5 will be assessed. It relates to $\mathbf{RQ}_2$ by examining stability, a less investigated aspect of forecasting quality.

- Chapter 5 introduces Data Mining methods for project control forecasting. The methods that we implemented hail from the Artificial Intelligence (AI) domain and leverage the power of historical data. One of the suggested methods, namely Support Vector Machines (SVM), ranks among the top 10 algorithms in Data Mining (Wu et al. (2008)) and results on the performance of this method have been published (Wauters and Vanhoucke (2014b)). In order to test all methods with sufficient rigour, a large computer experiment is set up. Historical data, as well as fictitious project executions were modeled by means of Monte Carlo simulations. Methodological characteristics that are common for AI methods such as pre-processing, grid search and cross-validation are heeded as well. Sensitivity experiments allow to point out the limitations of the proposed AI methods. This chapter demonstrates the power of employing historical data (cf. $\mathbf{RQ}_2$) given a sufficient approximation of the input data to the real project progress.

- Chapter 6 extends the previous chapter. The k-Nearest Neighbour method (k-NN), also among the top 10 algorithms in Data Mining (Wu et al. (2008)), is a very straightforward technique that is widely used for classification and regression purposes. In our research, the k-NN technique serves a dual purpose. First, it acts as an additional predictive method. Secondly, it is used as an intermediate methodological step to reduce the size of the training set. However, the reduction in information is compensated by an increase in quality. Only the most similar observations, based on the Nearest Neighbour distance, remain. This chapter also extends the results of chapter 4 by incorporating the stability of the methods presented in chapter 5. Consequently, this chapter can be regarded as a culmination of the research done with regard to the use of historical data for the two dimensions of forecasting quality, namely stability and accuracy. As a result, we believe this chapter perfectly demonstrates the advantages and pitfalls of the use of historical data and provides a nuanced answer to $\mathbf{RQ}_2$.

1.2.3 Conclusions & future research avenues

In chapter 7, we reflect on the studies of chapters 2 through 6. While each chapter contains a separate conclusion, this final chapter closes with more general remarks than the specifics found in each chapter. The conclusion constitutes the finale of this dissertation and looks back on the work we have done throughout these years of research. Since no book is ever complete, we also provide a future outlook and discern future research directions, going from small and incremental ideas that build on our research to more ambitious and time-consuming studies.

Publications in international journals:

- Wauters, M. and Vanhoucke, M. (2014). Support vector machine regression for project control forecasting. *Automation in Construction*, 47: 92-106.
- Wauters, M. and Vanhoucke, M. (2015). Study of the stability of Earned Value Management forecasting. *Journal of Construction Engineering and Management*, 141(4): 1-10.

Unpublished working papers:

- Wauters, M. and Vanhoucke, M. (2013). A study on complexity and uncertainty perception and solution strategies for the time/cost trade-off problem. *Under submission in A1 journal (third round).*
- Wauters, M. and Vanhoucke, M. (2015). Effort-based decision making for the Discrete Time/Cost Trade-off Problem. *Under submission in A1 journal (first round).*
- Wauters, M. and Vanhoucke, M. (2014). A comparative study of Artificial Intelligence methods for project duration forecasting. *Under submission in A1 journal (first round).*
- Wauters, M. and Vanhoucke, M. (2015). A Nearest Neighbour extension to Earned Value Management forecasting with Artificial Intelligence. *Under submission in A1 journal (second round).*

Part I

Time/cost optimization

2

A study on complexity and uncertainty perception and solution strategies for the time/cost trade-off problem

In this chapter, we revisit the Discrete Time/Cost Trade-off Problem (DTCTP) in light of a student experiment by expanding on a previously published study (Vanhoucke et al. (2005)). We derive a classification of student behaviour based on data collected from 444 students and identify five dimensions which make up a solution strategy. Two contextual factors, namely complexity and uncertainty, are taken into account. In order to establish a link between the solution strategies, the degree and perception of complexity and uncertainty and the overall solution quality, a rigorous computer experiment is set up. Finally, we investigate the impact of various settings on the solution quality.

2.1 Introduction

Time/cost trade-offs in project scheduling find their roots in the Critical Path Method (CPM), developed at the duPont company and at Remington Rand Univac (Kelley and Walker (1959), Walker and Sawyer (1959) and Kelley (1961)). CPM is a project scheduling technique to analyze and represent the tasks involved in completing a given project. Although this method does not explicitly take resource requirements into account, it assumes that the duration of an activity is a non-increasing function of the amount of money allocated to it. Initial research efforts on the time/cost trade-off problem focused on the continuous case and can be found in standard texts such as Elmaghraby (1977) and Moder et al. (1983). Several techniques were used to solve this type of problem (Robinson (1975), Hindelang and Muth (1979), Phillips and Dessouky (1977) and Meyer and Shaffer (1965)). An overview of the literature until the mid nineties is given by De et al. (1995). We will cover the contributions related to the time/cost trade-off problem from the mid nineties onwards. The Discrete Time/Cost Trade-off Problem (DTCTP), shown to be NP-hard by De et al. (1997), was solved exactly by Demeulemeester et al. (1996). In their paper, the authors present two approaches based on dynamic programming for reaching the optimal solution of the three objective functions of the DTCTP. Three possible variants of the time/cost trade-off problem can be identified. Scheduling project activities with the goal of minimizing the total project costs while meeting an imposed deadline is known as the deadline problem (DTCTP-D). The budget problem specifies a limit on the budget (DTCTP-B). The objective is then to minimize the duration of the project. Finally, the third objective deals with generating a complete and efficient time/cost profile. Demeulemeester et al. (1998) improved the computational results for solving the DTCTP optimally. This is done using a branch-and-bound procedure that calculates lower bounds by convex piecewise linear underestimations of the time/cost trade-off curves of the activities. This contribution is of special relevance to this paper since the procedure of Demeulemeester et al. (1998) will be used to provide an optimal solution for the data instances of the computational experiment.

The last decade, two new research avenues on the time/cost trade-off problem were examined. The first new direction is the extension of the (D)TCTP, while the second direction focuses on the inclusion of stochastic characteristics to the (D)TCTP. A brief overview of the key publications belonging to each avenue, along with their contribution, is provided in table 2.1. The contribution this chapter makes to the existing body of literature is threefold. First of all, the data of students participating in a project management business game, the Project Scheduling Game (PSG), are analyzed. The data files that were employed for this business game are instances of the DTCTP-D and are transformed into solution strategies. Secondly, we take two contextual factors, namely complexity and uncertainty, into account. While the first contribution employs real-life data, experiments are constrained by the fact that classroom sessions need to be held in order to gather additional data. The final contribution overcomes this problem by testing the derived solution strategies on computer-generated project networks. In the remainder of this section, we will elaborate on these contributions from a literature point of view.

Research stream	Paper	Contribution
Problem extensions	Vanhoucke (2005)	DTCTP with time-switch constraints. Outperforms Vanhoucke et al. (2002).
	Vanhoucke and Debels (2007)	Metaheuristic for time/switch constraints, work continuity and net present value maximization.
	Sonmez and Bettemir (2012)	Hybrid genetic algorithm for the DTCTP.
	Tareghian and Taheri (2006)	Three integer programming models for the time/cost/quality trade-off problem.
	Tareghian and Taheri (2007)	Scatter search with electromagnetic properties for the time/cost/quality trade-off problem.
	Pour et al. (2010)	Genetic algorithm with hill-climbing and decreasing mutation rate for the time/cost/quality trade-off problem.
Stochastic characteristics	Azaron et al. (2005)	Genetic algorithm for multi-objective TCTP with activity durations $\sim$ Erlang.
	Azaron and Tavakkoli-Moghaddam (2007)	PERT network as queuing system, spawning of new project and activity durations $\sim$ Exponential.
	Cohen et al. (2007)	Robust optimization for the stochastic TCTP.
	Ke et al. (2009)	Genetic algorithm-based algorithm for the stochastic TCTP.
	Hazır et al. (2010)	Robust scheduling and robustness measures based on slack.
	Klerides and Hadjiconstantinou (2010)	Two-stage stochastic integer programming approach.
	Hazır et al. (2011)	Schedule robustness with unknown interval-based cost parameters.
	Mokhtari et al. (2011)	Ant system approach for the stochastic TCTP.
	Chen and Tsai (2011)	TCTP using fuzzy numners.
	Ke et al. (2012)	Formulation of three stochastic TCTP models using chance-constrained & dependent-change programming.
	Ghoddousi et al. (2013)	Non-domination based genetic algorithm for multi-objective TCTP.

Table 2.1: Recent literature overview

Business games Business games have a long history within an educational context. Early research focused on the internal validity through assessing advantages and disadvantages of simulations versus other pedagogies (Schumann et al. (1997)). Later on, the validity of top management games was confirmed by Wolfe (1997). The most-cited advantages of the use of business games are their high degree of realism, a broader learning environment, competition between players, as well as soft skills such as communication skills, group behaviour and organization skills (Saunders (1997) and Faria (2001)). On top of this, business games craft personal experiences by challenging participants on an intellectual and behavioural level and hence fall within the nominator experiential learning (Kolb (1984)). Parente et al. (2012) argue in favour of business games by stating that real-life experience imposes limitations since there is no opportunity to experience the full range of possibilities and skill development. Business games have been applied to simulate business and operations management in the electronics industry (Haapasalo and Hyvönen (2001)), to teach business ethics (Schumann et al. (1997)), to develop entrepreneurial skills (Stumpf et al. (1991)) and to enhance systems thinking and business process redesign (Van Ackere et al. (1993)).

Complexity and uncertainty What the papers of table 2.1 have in common is that they focus on what Pollack (2007) describes as the hard paradigm, which is commonly associated with quantitative techniques and deductive reasoning. However, the author identifies research streams that suggest an increasing acceptance of the soft paradigm, which focuses on qualitative techniques that emphasize contextual factors and relevance. Examples of soft paradigm publications can be found in Turner and Müller (2005), Ojiako et al. (2014), Green (2004) and Yang et al. (2011) and the reader is referred to the relationship school and behavioural school of Söderlund (2011) for a literature review on soft paradigm aspects. More and more, researchers are calling for a broader view of project management (Hanisch and Wald (2011)), an increased alignment of research and practice (Hall (2012)) or the inclusion of contextual factors (Crawford et al. (2006)). With regard to the latter point, Maylor et al. (2008) argue that one-size-fits-all approaches are inconsistent with the contextual diversity managers are confronted with. In this paper, we take two contextual factors, namely complexity and uncertainty, into account by means of data that result from students participating in a business game.

The choice for complexity and uncertainty is inspired by two reasons. Firstly, Howell et al. (2010) noted that uncertainty was easily the most dominant theme, with complexity being the second most common. This confirmed the findings of Shenhar (2001) who discovered an emergence of uncertainty and complexity based on a review of the classical as well as the more recent literature. It is worth noting that the works dealing with stochastic characteristics in table 2.1 can be regarded as dealing with uncertainty. Additionally, the works of Thomas and Mengel (2008) and Hanisch and Wald (2011) explicitly recognize complexity and uncertainty as crucial (contextual) factors. Secondly, integrating contextual factors can be regarded as a response to areas for future research. Hanisch and Wald (2011) argued that the influence of complexity on the project outcome needs to be studied, while Maylor et al. (2008) wondered whether a quantification of complexity was feasible. While research on uncertainty has witnessed a spike in interest from academics, it is still among the top challenges for future research (Hall (2012)).

Simulation Complexity can be defined from a hard paradigm perspective (see the complexity measures of Pascoe (1966), Mastor (1970), Bein et al. (1992) and De Reyck and Herroelen (1996)) or include soft paradigm aspects (such as organizational complexity (Wolfe (1996)) or socio-political complexity (Geraldi et al. (2011)). In this work, we focus on structural complexity and more specifically on system size (Sommer and Loch (2004)). Apart from a large body of work that supports this stance (Dvir et al. (2006), Geraldi and Adlbrecht (2007), Müller and Turner (2007)), the main rationale for focusing on system size can be found in the final contribution this paper makes. Once the student files are analyzed and turned into solution strategies, these strategies are applied to computer-generated project networks, for which multiple settings are changed. In order to do this, a more technical definition of complexity (and uncertainty) is required. Specific attention will be allocated to the discrepancy between the actual complexity and uncertainty and how these contextual factors are perceived. Individuals perceive reality in their own way (Jaafari (2003)), implying that complexity and uncertainty are also in the eye of the beholder (Nutt (1998), Vidal and Marle (2008), Osman (2010), Ojiako et al. (2014)). Simulations aid decision makers in anticipating and quantifying the effects of actions and events (Fang and Marle (2012)) and allow us to generate a wide spectrum of outcomes.

The outline of this chapter is as follows. In section 2.2, a general overview of the Project Scheduling Game is given. Section 2.3 focuses on the data collection phase. The building blocks of the solution strategies, the link to the student data of section 2.3 and how the solutions are evaluated are discussed in section 2.4. An illustrative example is provided to demonstrate the 5 components that make up a solution strategy. Section 2.5 introduces the general framework that forms the foundation for the solution strategies and introduces the time-based and cost-based solution strategy. Section 2.6 includes details about the test design. Parameter settings are divided depending on whether they are project-specific or whether they relate to the complexity and uncertainty dimension. The results of the solution strategies are discussed in section 2.6.2, where a distinction is made between the general performance, the performance in case of judgement errors and an investigation of a varying degree of level of effort. Finally, a discussion of the results and general conclusions can be found in section 2.7.

2.2 Game description

Crowston and Thompson (1967) were among the first authors to stress the importance of the interaction between the planning, scheduling and control phase of a project. The focus of the Project Scheduling Game, presented by Vanhoucke et al. (2005), lies in the scheduling and control phases of the project life cycle. More precisely, it is the aim of the player to follow an iterative approach, known as reactive scheduling, that compares the project baseline schedule with the current project performance (simulated during the execution phase) in order to control the project and take corrective actions in case the project objective is in danger. The game consists of several phases which require periodic input from the game player. An overview of the game process is given in figure 2.1. First of all, the project network, along with a baseline schedule and other input data for the game such as the trade-off details are proposed by the course teacher. The baseline schedule ends at time T . In order to acquaint students with uncertainty, unexpected events occur. A new deadline, $\delta_n < T$, is imposed by the client along with a penalty cost for every day the deadline is exceeded. Therefore, the PSG imposes a soft deadline that need not be met. However, late project delivery is discouraged by means of a penalty that is incurred for every day the project finishes after δ_n. These changes require an update of the baseline schedule which is the task of the player of the game. The update process of the baseline schedule boils down to a new trade-off selection for a number of activities. New trade-offs lead to a shortened or prolonged duration of an activity and lie at the heart of the CPM. Second, the project is divided into multiple decision moments. The game then simulates periodic project progress in which uncertain events might occur. Changes to the original activity durations lead to deviations from the initial baseline schedule and endanger the project objective. For every decision moment, the player has to evaluate periodic review reports and rebaseline the unfinished activities of the project schedule in order to bring the project back on track. This process of rescheduling, taking a decision and assessing the new information is repeated until the final decision moment is reached. Third, after a predefined number of decision moments, the game reports

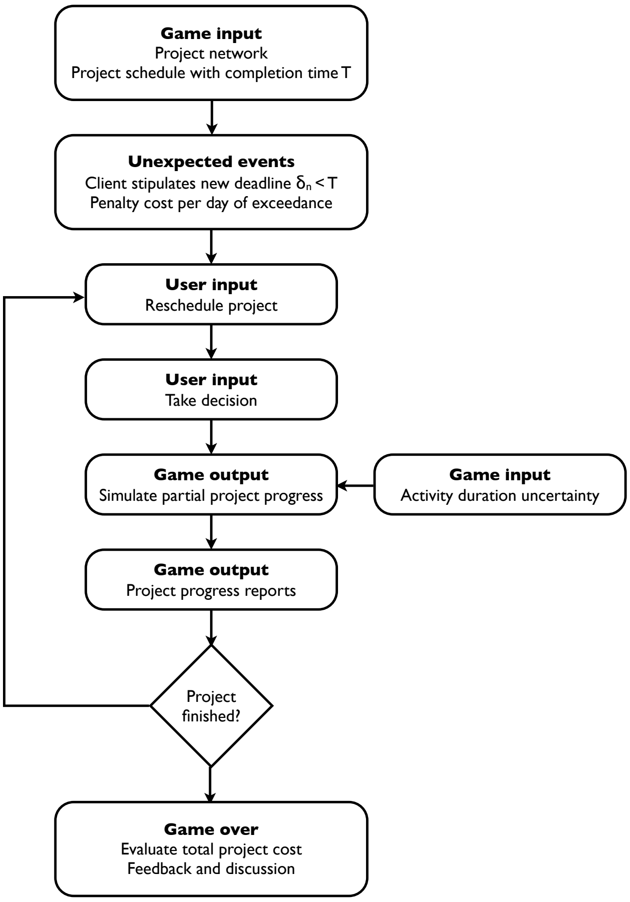


Figure 2.1: Overview of the PSG game process (Vanhoucke et al. (2005))

the final project status in terms of the total project duration and cost. At this point, feedback is given by the course teacher regarding the main learning objectives of the PSG.

2.3 Data collection

The PSG is taught to management and engineering students at 2 universities (Ghent University (UGent, Belgium) and University College of London (UCL, United Kingdom)) and 2 business schools (Vlerick Business School (VBS, Belgium) and the EDHEC business school (France). 444 data points were collected, among which 176 business engineering students (UGent), 203 civil engineering students (UGent), 36 civil engineering students (UCL), 5 management science students (UCL) and 24 MBA students at the VBS and EDHEC school. Most of these students do not have previous working knowledge. It is worth remarking that the students business engineering and most of the civil engineering students hail from the same university (UGent).

A distinction can be made between data captured during game progress and final results. All the data captured during game progress is saved in a log file. For every student playing the PSG, a log file is available. These log files store commands students execute. The most important commands are the change of an activity's trade-off option and taking a decision, implying a move to the next decision moment. Taking a decision executes the new trade-off settings for the activities and simulates project progress for the next decision moment. Examination of the commands leading to intermediate and final solutions permits an identification and classification of solution strategies taken by the students. For instance, by executing the commands stored in the log files chronologically, it is possible to know whether a student made a change to a critical or non-critical activity and what that change consisted of. The final time and cost solution for every student is only available at the end of the game. Table 2.2 summarizes the results of the different student groups. The average deviation compared to the global minimum cost point is less than 2% for all student groups. The best solution of the student groups displays only a very small deviation from the best solution possible (less than 1%). The best overall solution was found by a business engineering student, whereas on average, the management science students of UCL report the best average score. They also achieve the smallest standard deviation in cost and the lowest maximum cost deviation. It is worth remarking that the group of management science students at UCL only comprises 5 people. Welch's t-test was applied to find out whether the cost deviations between the

Student Group	#students	Cost deviation (%)			
		Minimum	Maximum	Average	σ
Business Engineering (UGent)	176	0.19	12.46	1.52	1.76
Civil Engineering (UGent)	203	0.24	12.76	1.29	1.44
Civil Engineering (UCL)	36	0.38	4.52	1.75	0.82
Management Science (UCL)	5	0.50	1.79	1.10	0.54
Management (VBS & EDHEC)	24	0.38	4.60	1.72	1.34

Table 2.2: Overview of the student results

groups differ significantly. The only statistically significant difference ($p<0.05$) is found for the students Civil Engineering at UGent and the students Civil Engineering at UCL.

2.4 Data structuring

Based on the aggregated data, the log files and time/cost deviations, it is necessary to transform the data into information by finding a certain structure according to which students play the PSG. This is accomplished by looking for recurring data patterns. A data pattern that corresponds with a certain class of behaviour, exhibited by many students, will be called a solution strategy. Section 2.4.1 expands on the 5 building blocks that form one solution strategy. The following section explains the link between the solution strategy components and the student data of section 2.3. Section 2.4.3 presents details about the performance measures that will be used to evaluate the strategies. Section 2.4.4 provides an illustrative example and shows the different steps going from activity selection to applying a trade-off change.

2.4.1 Solution strategy components

From the data collection phase and through many discussions with the students at the educational institutions, we learned that very few students approach the PSG without any underlying logic. It is possible to discern 5 building blocks that are in line with when and how students select a new mode for the different activities. The 5 components that characterize a solution strategy are focus, activity criticality, ranking, intensity and action. It is worth noting that the first 4 elements are related to selecting a set of activities whereas the final element, action, determines how the trade-offs of the set of activities will be altered. This corresponds with the three general building blocks of heuristics (Gigerenzer and Gaissmaier (2011)). Search rules specify the direction of the search space and are accounted for by focus, activity criticality, ranking and intensity. Stopping rules determine when the search process ends and is governed by the time limit of the PSG. Decision rules elaborate on how the final decision is reached. This is done by the fifth component, namely action. At the start of a decision moment, every activity that has not started is subject to a possible change. Out of this group of activities, focus, activity criticality, ranking and intensity perform a stepwise selection of a subset of activities. The process of stepwise selection can be described as follows:

- Focus: specifies the length of the time window during which actions will be taken. All activities that start or are still in progress during this time window are selected. The focus is expressed as a percentage of the number of decision periods that are taken into account and can vary from a local to a global orientation. A local orientation is characterized by a narrow time window because the number of decision periods taken into account is small. At the other end of the spectrum is a global orientation, which uses a wide time window. In this case, many activities will be subject to a possible trade-off change.

- Activity criticality: the subset of activities that start or are in progress during the time window specified by the focus can be further refined based on whether these activities are critical or non-critical at the current decision moment. If both critical and non-critical activities are taken into consideration, the subset of activities before this phase equals the subset at the end of the phase.

- Ranking: the elements of the subset of activities are ranked based on the value of a priority rule. Within the context of human decision-making, priority rules are easy to apply and in line with techniques that are used to give priority to certain activities. A tight match could be witnessed between the followed solution strategy and the selected priority rule. For instance, students who thought that the minimum cost solution would lie in the neighbourhood of the deadline would adopt a more time-based strategy and select a priority rule that takes into account activity durations. This selection step does not reduce the subset of activities but accords a ranking to the activities. These rankings serve as input for the intensity phase. The priority rules used by the solution strategies are the Greatest Rank Positional Weight (GRPW), Maximum Slack (MAXSLK) and Average Most Expensive Activity (the activity cost divided by its duration) rules.

- Intensity: given the fact that students have a limited amount of time to take decisions, it is crucial to focus on the most important activities. Intensity further

Type of action	Description
Swap	Select neighbouring trade-off.
Slack consumption	increase duration until no slack is left.
Minimum cost slope	Select trade-off with maximum duration decrease at minimum cost.
Maximum savings slope	Select trade-off with minimum duration increase at maximum savings.
Enumeration	Enumerate all trade-off for set of activities.
Protect deadline	Decrease/increase project duration until acceptable deviation from δ_n.

Table 2.3: Summary of the type of actions

selects activities by determining a cut-off point for the ranked subset that resulted from the previous phase. A percentage between 0% and 100% of the number of remaining activities of the ranked subset is used as a value for the intensity. This percentage is multiplied by the number of elements that are present in the ranked subset. This subset then serves as input for the action phase, where the trade-offs of activities that are elements of the subset may be changed.

- Action: an action is defined as a move on the trade-offs of an activity that may potentially change an activity's cost and associated duration. This need not be the case since a lot of students check whether the action leads to an immediate cost decrease or not. This is done by comparing the penalty cost per day and the cost or savings of a trade-off change. If there is no improvement, it is possible that the activity's cost and duration is reverted. Actions can go from simple to more advanced operations. An overview of the type of actions and an accompanying description is given in table 2.3.

The rationale for the different refinement phases leads back to the nature of the PSG, where students only have a limited amount of time to make changes and advance to the next decision moment. Hence, it is necessary to focus on the activities that are most important. In order to clarify the building blocks of the solution strategies, the stepwise selection and action will be illustrated using a straightforward example.

2.4.2 Link to the student data

The previous section outlined the different building blocks of a solution strategy. The aim of this section is to connect the 5 components (focus, activity criticality, ranking, intensity and action) to the student data of section 2.3. The link between these two sections results from two aspects, namely data analysis and the feedback sessions with the participants of the Project Scheduling Game. Most of the time, these aspects go hand in hand. For instance, many of the conversations revealed that students first select a number of activities to which a trade-off change can be made. Further questioning led to the formalization of this selection and to the inception of focus and intensity.

- Focus: the values for the focus and intensity could be retrieved from the log files of the students. Figure 2.2 shows the boxplots of the focus for each of the decision moments of the PSG. The boxplots are based on the data from all participants. It can be seen that a wide focus range is used by the students, which explains why low and high numbers for the focus are used by the solution strategies.

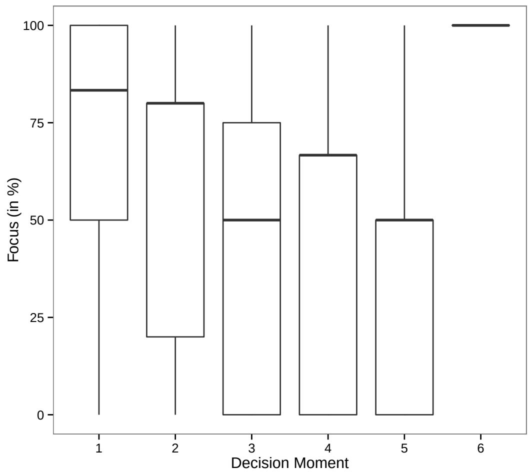


Figure 2.2: Boxplot of the focus for different decision moments (student data)

- Activity criticality: Activity criticality is the second of the five building blocks that make up a solution strategy. It is logical that both critical and non-critical activities are changed.

- Ranking: arguably, the student feedback proved most valuable in identifying the priority rules for the ranking phase that were used most often, especially since these are much harder to keep track of in the log files.

- Intensity: the process of translating intensity from the student data to the solution strategies was slightly different. Intensity, as defined in section 2.4.1, can reach any value in the interval [0,100]. A careful trade-off needs to be made between a sufficient data representation and having as few values for the intensity setting as possible. In theory, it would be possible to incorporate all possible values the intensity can achieve in the solution strategies. However, this entails that a huge number of branches need to be created to test which value will be applied under which circumstances. Such a situation reflects the data in an extremely accurate fashion but is no longer feasible for the computational experiment. As a rule-of-thumb, the fewer different values that represent the data of the students' adopted intensity well, the better. Feedback from the students taught us that selection of a number of activities typically occurred through a rule-of-thumb, such as "1 out of 4 activities" will be retained. Figure 2.3 depicts the probability on the x-axis and the sample quantile on the y-axis. There are two principal reasons why the values 0.25, 0.5, 0.75 and 1 were embedded in the solution strategies. First of all, from a cognitive point of view, it is better if the different values are equally spaced in the interval that ranges from 0 to 100. Second, while there are only 4 different values, the data is represented in a sufficiently accurate way, as evidenced from figure 3. The 25th, 50th, 75th and 100th quantiles correspond with probabilities of 14, 50, 81 and 100%, respectively. Consequently, there is ample difference in probability between the 4 selected quantiles.

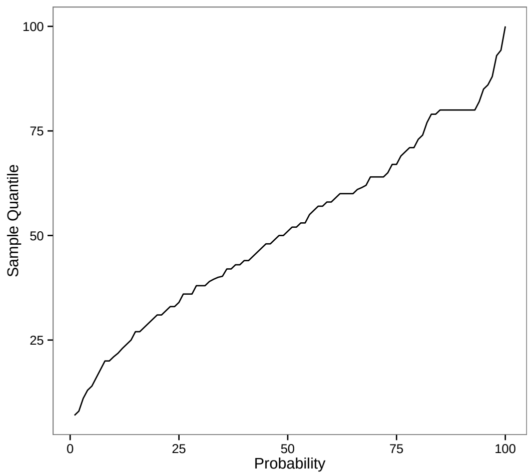


Figure 2.3: Sample quantiles of the intensity (student data)

- Action: the final element, the actions, once again resulted from the feedback and the log files. For this dimension, it is easier to find in the data which actions were followed, with one notable exception. An enumeration of different trade-off options can easily be confused with sequential swaps. Figure 2.4 displays the 4 actions that were frequently applied, along with their probability of occurrence. It is clear that participants of the game make frequent use of swapping trade-off options. This also explains the frequent inclusion of this action in the solution strategies that will be outlined in section 2.5.2.

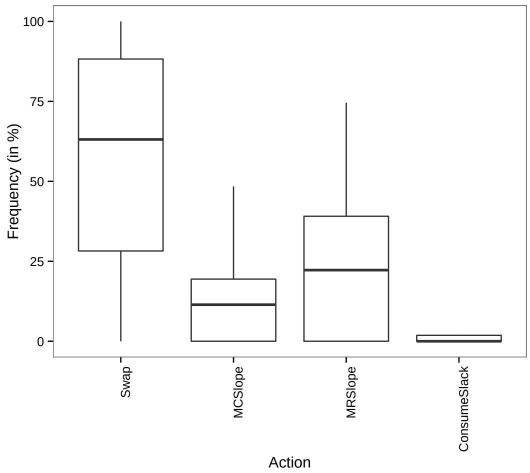


Figure 2.4: Frequency of the applied actions (student data)

2.4.3 Evaluation

The solution strategy components of the previous section were derived using the data accessible from the log files. As mentioned in section 2.3, the second type of data, the results, are gathered at the end of the game. In order to rate the quality of the student solutions, represented by the derived solution strategies, it is necessary to define performance measures. The proposed measures capture 2 dimensions of a final solution, namely cost and level of effort. Every dimension can be measured using a specific metric, which is outlined below and depicted in figure 2.5. The x-axis of figure 2.5 represents the deviation from the deadline in absolute numbers, whereas the y-axis displays the total costs. The curve shows the efficient time/cost profile. The time value of the minimum cost solution across the entire efficient time/cost profile is denoted by t^* . The dot stands for the solution of a student at the end of the game.

- Cost performance: the cost deviation is measured using the global cost deviation. This deviation compares the project cost of the solution strategy (the dot) to the solution that yields the minimum cost across all possible time points of the complete time/cost profile (the cost or y-value at time t^*). The global cost deviation is expressed as a percentage deviation:

$$\Delta cost_{global} = \frac{c^s - c^*_{t^*}}{c^s} \tag{2.1}$$

In this calculation, c^s stands for the cost of the solution strategy and $c^*_{t^*}$ denotes the cost of the efficient time/cost profile at time t^* . It is possible to break down the global cost deviation into 2 constituent parts, namely activity costs and penalty costs. If a project finishes later than the specified deadline, it incurs a penalty cost. Hence, by looking at the penalty cost, we implicitly derive some information as well.

- Level of effort: captures how much effort it takes to reach a solution, which results from one of the solution strategies. It is worth noting that the level of effort is a function of focus, activity criticality and intensity. As the focus increases, the amount of potential activities that are changed rises and consequently, the level of effort increases as well. This dimension aims to establish the link between solution quality and the amount of work that was performed to reach that solution. One unit of effort corresponds with one trade-off option that is considered for a change. For instance, in order to determine the minimum cost slope, a number of trade-off options are considered. Each of those trade-offs augments the level of effort by 1 unit.

2.4.4 Illustrative example

Figure 2.6(a) represents the Activity on the Node (AoN) notation of an example network. We note that this example merely serves as an illustration: the networks of the computational study count more activities and different trade-off options. In this example, there are 7 activities in total. The possible durations of the trade-off options for every activity are indicated above each node. The associated costs of the trade-offs are listed under each node. The currently selected trade-off is bolded. For instance, for activity 2, the currently selected trade-off has a duration equal to 2 at a cost of €100. Figure 2.6(b) depicts the earliest start Gantt-chart, taking into account the precedence relations between the activities. Critical activities (activities 1, 3, 4, 5 and 7) are highlighted in grey, whereas non-critical activities are indicated by the non-coloured bars (activities 2 and 6). There are also 3 different decision moments (DM). In this example, it is possible to make changes to a set of activities at time points 0, 5 and 10 respectively. The total duration of the project equals 16 days. We assume that a decision needs to be made at

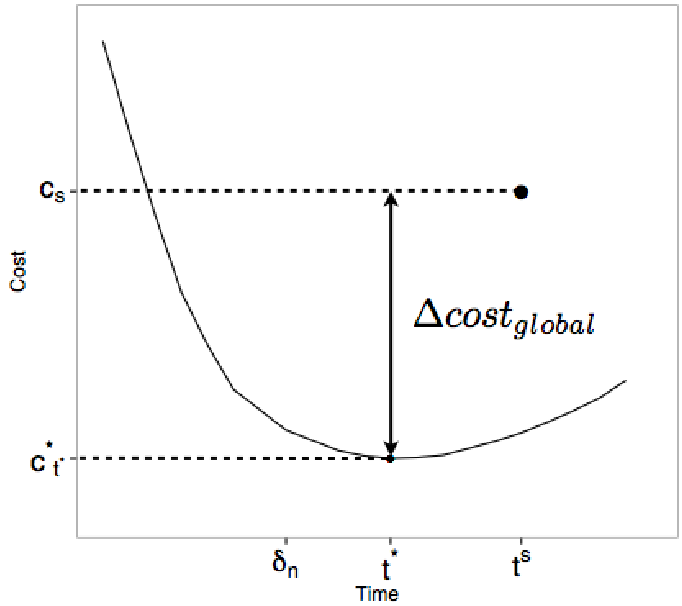


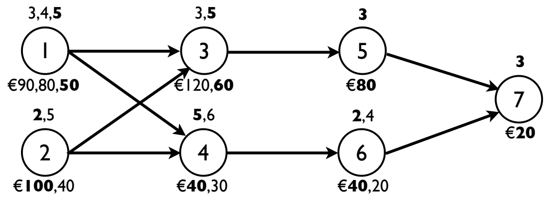
Figure 2.5: Global cost deviation

time point 0 and that a deadline of 13 days is present. If the project duration exceeds the deadline, a penalty cost of €100/day is incurred. Hence, we are at the beginning of the project. Let E denote the set of eligible activities. Eligible activities are defined as activities for which the currently selected trade-off will be changed. Hence, at the start of the project, the set E consists of every activity present in the network:

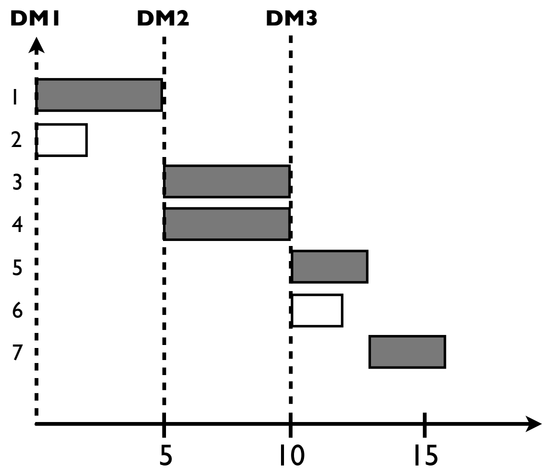
$$E = \{1, 2, 3, 4, 5, 6, 7\}$$

The example settings for the 5 components (focus, activity criticality, ranking, intensity and action) are listed in table 2.4. The focus is assumed to be equal to 2 decision moments (0.67 * 3, the total number of decision moments). From that set of activities, only the critical activities will be retained. These will then be sorted according to the Most Expensive Activity priority rule. From the ranked subset, only the first 67% will be withheld. Finally, those remaining activities will be crashed by applying a swap operator.

- Focus: in this example, a focus of 2 decision moments is used. Given the fact that the current decision moment is equal to 1, the time window for the activities that will be retained is equal to [1, 1 + 2[. Hence, activities that start later than



(a) Activity on the Node network



(b) Gantt chart

Figure 2.6: Activity on the Node network (2.6(a)) and Gantt chart (2.6(b)) for the illustrative example

Component	Setting
Focus	0.67
Activity criticality	Critical
Ranking	Most Expensive Activity
Intensity	0.67
Action	Crash with Swap move
Parameter	**Value**
Deadline	13 days
Penalty	€100

Table 2.4: Overview of the 5 components for the illustrative example

decision moment 3 (after time point 10) will no longer be considered.

$$E = \{1, 2, 3, 4\}$$

- Activity criticality: only critical activities will be taken into consideration. This implies that activity 2 will be removed from E.

$$E = \{1, 3, 4\}$$

- Ranking: cost is the most important objective for this example network. Hence, the priority rule Most Expensive Activity will be used. Activity 1 has a cost of €50, activity 3 a cost of €60 and activity 4 costs €40. Consequently, the activities in E are reordered as follows:

$$E = \{3, 1, 4\}$$

- Intensity: further refinements can be made based on the intensity. In this example, an intensity value of 0.67 will be used. This means that only 2 (0.67 * 3 activities) activities will be left. Because of the ranking in the previous phase, the first two activities will be selected.

$$E = \{3, 1\}$$

The 5 building blocks have gone from a set where all activities could be changed to a situation where only 2 activities are left. In the final phase, an action will be applied to those activities to change their selected trade-off:

- Action: activities 3 and 1 will be crashed by selecting the neighbouring trade-off option. This implies that the duration of activity 3 will become 3 time units, with a cost of €120. Activity 1 will now take 4 days to complete at a cost of €80. This leads to the Gantt-chart in figure 2.7, which will be the initial situation for the next decision moment (decision moment 2). The Gantt-chart indicates that the critical path has changed and now consists of activities 1, 4, 6 and 7. The total duration of the project has decreased from 16 to 14 days. The project's total cost has decreased from €690 (€390 + 3 days * €100) to €580 (€480 + 1 day * €100).

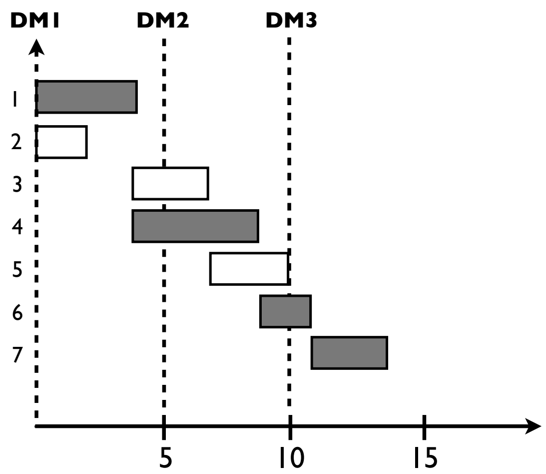


Figure 2.7: Gantt chart of the executed illustrative example

In order to demonstrate the trade-off between solution quality and Level Of Effort, 4 solution strategies are applied to the toy example described in the paragraphs above. The first strategy consists of doing nothing, yielding a total cost of €690. Strategy 1 employs a focus and intensity of 0.33, strategy 2 assumes a focus and intensity of 0.66 and strategy 3 adopts a maximal focus and intensity (=1). These strategies are shown in table 2.5, in which NA denotes that crashing is not possible because the minimum duration has been reached. The upper part of table 2.5 shows which activities were changed or considered for a change. The lower part displays the project's final outcome and compares it to a scenario where no action is taken. While there is an improvement in cost between doing nothing and strategies 1 and 2, there is no advantage in increasing the focus and intensity beyond 0.66. Hence, it is shown how an increasing focus and intensity improves the cost objective until a point is reached where further increases lead to a dramatic increase in Level Of Effort, to the same cost solution or both.

Strategy	Decision Moment	Activity	Old trade-off Time	Old trade-off Cost	New trade-off Time	New trade-off Cost
Strategy 1	1	1	5	50	4	80
	2	5	3	80	NA	
	3	5	3	80	NA	
Strategy 2	1	3	5	60	3	120
		1	5	50	4	80
	2	6	2	40	NA	
	3	6	2	40	NA	
		7	3	20	NA	
Strategy 3	1	5	3	80	NA	
		3	5	60	3	120
		1	5	50	4	80
		6	2	40	NA	
		4	5	40	NA	
		7	3	20	NA	
	2	4	5	40	NA	
		6	2	40	NA	
		7	3	20	NA	
	3	6	2	40	NA	
		7	3	20	NA	

Strategy	Project duration	Project cost	Level Of Effort
Do nothing	16	690	0
Strategy 1	15	620	1
Strategy 2	14	580	3
Strategy 3	14	580	3

Table 2.5: Alternative strategies for the illustrative example

2.5 Strategic framework

The strategic framework contains information about the conditions in which the derived solution strategies operate. Two defining criteria are determined, namely complexity and uncertainty. The general framework of the solution strategies, as well as complexity and uncertainty appraisal, are the subject of section 2.5.1, in which the interplay of these dimensions with the solution strategies will be clarified. Section 2.5.2 lists the 2 proposed solution strategies. One of these strategies will concentrate on time while the other strategy adopts a cost-based point of view.

2.5.1 General framework

Section 2.4 described the 5 building blocks of a solution strategy: focus, activity criticality, ranking, intensity and applying an action. These elements will be used to construct the strategies that were derived based on the data collected in the log files and based on the discussions with the students after finishing the game. However, the proposed strategies take 2 important criteria into account, namely complexity and uncertainty. A crucial element throughout this chapter is the difference between the real and the perceived complexity or uncertainty. People make decisions based on the perceived complexity or uncertainty without knowing their real value. Hence, it is possible that judgement errors, in which the threshold value differs from the real value, occur. If the uncertainty or complexity estimate exceeds a threshold value, the outcome for that dimension will be judged high. Otherwise, that dimension will be judged low. Details on these thresholds will be provided in section 2.6.1. The outcome of the judgement of the complexity and uncertainty will steer the logic of the solution strategies into a different direction. This implies that different settings for the stepwise activity selection (focus, activity criticality, ranking and intensity) and action phases may be applied. Complexity is measured by the average number of trade-offs of the different activities and is calculated as follows:

$$C = \frac{\sum_{i=1}^{n} nrto_i}{n} \tag{2.2}$$

with n denoting the total number of activities and $nrto_i$ the number of trade-off options for activity i. Students often use the proportion of activities that were subject to a delay in previous decision periods as an indicator for future uncertainty. Hence, the uncertainty witnessed throughout the project is used to make a judgement about the project's overall uncertainty. The outcome of the complexity and uncertainty criteria is a binary value: either a project is highly complex (or very uncertain) or not. The values for complexity and uncertainty are taken into account by all the solution strategies.

The actual complexity and uncertainty is imposed by the decision maker. Individuals differ on how they judge complexity and uncertainty (Mintzberg et al. (1976), Bourgeois (1985)), which will be imitated in the computational experiment by incorporating different threshold values, leading to a different judgement of complexity and uncertainty.

The general framework of the solution strategies is given in figure 2.8. At the start of the project, the complexity is analyzed. If the complexity is low and the project is about to start (the decision moment equals 0), a group of settings and actions labeled **A** is triggered. If some activities are finished already, a new estimate of the uncertainty is made. If uncertainty has shifted from low to high or high to low, the focus and intensity are adapted (**B**). If uncertainty is smaller than the threshold value ($U <$), it is checked whether the next activity in the priority list constructed by the ranking phase is critical. The reader is reminded that these solution strategies ultimately change trade-off options of individual activities. If this is the case, **C** will be triggered. In the alternative case, the settings and actions encompassed in **D** are executed. Finally, if uncertainty is high, a similar check with regard to the (non-)critical nature of an activity is performed. If the activity is critical and has only 1 predecessor, the procedure moves to the branch labeled **E**. If the activity is non-critical, the actions and settings comprised in **F** will be activated. A similar but slightly more intricate pattern is executed in case the project's complexity exceeds the threshold value. When the project has just started, a couple of additional branches (**G-I**) are present. Furthermore, if a critical activity has more than 1 predecessor, a set of settings and actions will be applied as well. After applying a set of actions from one of the possible groups, a check is performed in order to determine whether the project has ended. If this is not the case, the project moves to the next decision moment. Otherwise, the output measures are calculated and the solution strategy has come to an end. This framework and its different branches will be used in the next section to structure the proposed solution strategies.

2.5.2 Proposed strategies

Armed with the data and the structure of section 2.5.1 that was used by many students, it was possible to derive 2 solution strategies. The first strategy, treated in section 2.5.2.1, focuses on time and more specifically on reaching the imposed deadline δ_n. The second strategy employs cost-saving measures at the expense of increased risk and is discussed in section 2.5.2.2. The level of effort was controlled using the focus and intensity parameters, which differ based on the level of complexity and uncertainty, and activity criticality. This was necessary to ensure that the level of effort was equal across

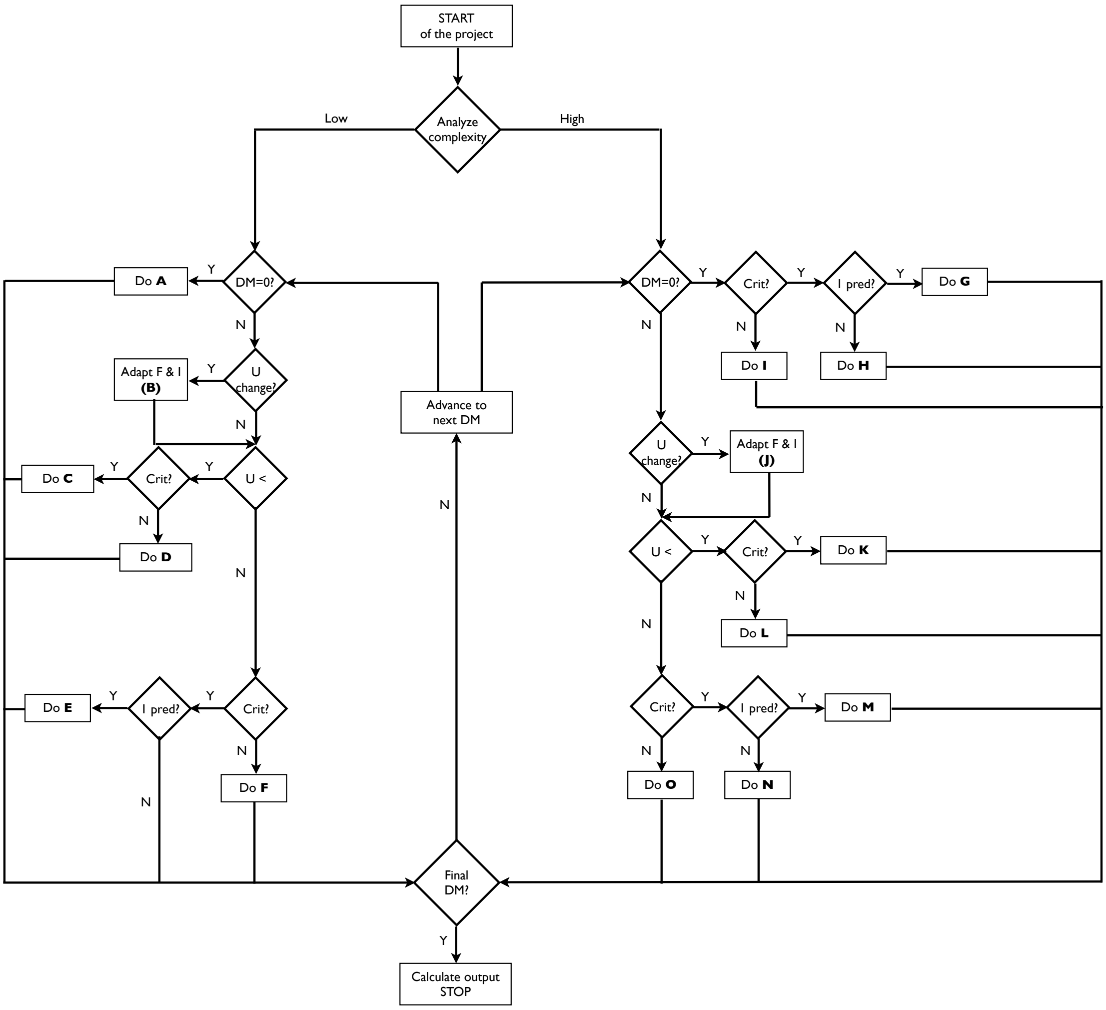


Figure 2.8: Strategic framework of the solution strategies

all settings for the computer experiment. In the following sections, we briefly highlight the main characteristics of the time-based strategy (section 2.5.2.1) and the cost-based strategy (section 2.5.2.2).

2.5.2.1 Time strategy

The goal of the time strategy is to approach the deadline as closely as possible. This solution strategy, abbreviated SS_1, uses three specific mechanisms that employ a time-based focus. First of all, the Greatest Rank Positional Weight (GRPW) rule is invoked in several branches. This rule takes the duration of the activity under study and the durations of its immediate successors into account, thus capturing a small portion of the network structure. Secondly, a buffer mechanism is employed. The buffer is based on the Slack Duration Ratio (SDR) by Hazır et al. (2010) and implies that a minimum value for the ratio of an activity's slack to its duration should be maintained. Consequently, non-critical activities are protected against delays that could turn them into critical activities and delay the entire project. The buffer mechanism is invoked when the uncertainty is judged to be high. Finally, the protect deadline action is used to ensure that the project duration does not deviate too much from the imposed deadline. Protection of the deadline is done when the project is in progress (DM$\neq$0). It ensures that the deadline does not exceed student-specified bounds. If the project is slightly uncertain, the project duration should lie between 98% and 101% of the deadline. For highly uncertain projects, the lower bound on the deadline becomes 96%. Because of the increased uncertainty, the delays will push the duration closer to the deadline, which explains why a smaller lower bound is chosen. The downward protection of 98% and 96% is not applied when the complexity is judged high. In this case, more effort is put into the examination of different trade-off options. However, the protection of 101% of the deadline is still in place in order to minimize penalty costs.

2.5.2.2 Cost strategy

The cost strategy, SS_2, aims to minimize the sum of activity costs and the penalty cost. Contrary to the time strategy, taking risk will be an integral part of this solution strategy. This is done using 3 different mechanisms. First and foremost, the average most expensive activity rule is used a lot more compared to the time strategy, implying that the importance of costs versus the duration of an activity becomes more important. Secondly, the elitism criterion is used frequently. Only accepting better solutions can be done for simple and complex actions, for instance by selecting the best outcome of a

Solution strategies	
SS_1	SS_2
GRPW priority rule	Avg MEA priority rule
Buffer (SDR of 30%)	Elitism (only accept cost improvement)
Protect deadline	Slack consumption

Table 2.6: Principal differences between the solution strategies

minimum cost or maximum savings slope. Third, consumption of slack plays a central role. Including this action has a double effect. On the one hand, costs will decrease because longer activity durations (at a lower cost) will be selected. On the other hand, because there is less slack in the project's schedule, the amount of risk increases. The logical result is that activity delays will have a larger impact on the schedule, thus increasing the penalty costs. Again, a distinction is made between judging projects to possess a low or high degree of complexity.

A summary of the principal differences between both solution strategies is given in table 2.6. The reader is referred to table 2.7 for an exhaustive overview of the two proposed solution strategies. While this table is mainly relevant for researchers who want to imitate our computational experiment with identical parameter settings, it can easily be skipped by the reader without losing the general overview of the theme of this paper. Table 2.7 lists the settings of the solution strategy components for every branch. The letters of the respective branches correspond with those depicted in figure 2.8. For focus and intensity, an additional distinction is made based on the actual complexity, which can be low (denoted C_L) or high (denoted C_H). For the ranking component, Avg MEA is the abbreviation for the average Most Expensive Activity priority rule, whereas Max SLK denotes the maximum slack priority rule. Finally, in the action column, MC represents the Minimum Cost while MS stands for the Maximum Savings. The subscript *elit* refers to elitism, meaning that if an action leads to a cost deterioration, the action will be undone and the project reverts to the trade-offs before the action was applied. For a step-by-step procedure of the data generation phase and the settings of the solution strategies, the reader is referred to this chapter's Appendix.

Solution Strategy	Perception		Branch	Solution Strategy Component			
	Complexity	Uncertainty		Focus	Intensity	Ranking	Action
Time (SS_1)	Judged L	NA	A	$C_L=0.3 \mid C_H=0.3$	$C_L=1 \mid C_H=0.5$	GRPW	Crash/prolong $Swap_{elit}$
		Judged L	B & C	$C_L=0.4 \mid C_H=0.3$	$C_L=0.5 \mid C_H=0.5$	GRPW	Crash MC Slope
			B & D				Prolong MS Slope
		Judged H	B & E	$C_L=0.4 \mid C_H=0.3$	$C_L=1.0 \mid C_H=0.5$	Avg MEA	Crash MC Slope
			B & F				Prolong Swap
	Judged H	NA	G	$C_L=1.0 \mid C_H=0.7$	$C_L=1.0 \mid C_H=1.0$	GRPW	Crash/prolong $Swap_{elit}$
			H				Crash Swap
			I				Prolong Swap
		Judged L	J & K	$C_L=0.9 \mid C_H=0.5$	$C_L=1.0 \mid C_H=0.7$	GRPW	Crash $Swap_{elit}$
			J & L			Max SLK	Prolong $Swap_{elit}$
		Judged H	J & M	$C_L=0.2 \mid C_H=0.1$	$C_L=0.5 \mid C_H=0.7$	GRPW	Crash $Swap_{elit}$
			J &N				Enumerate
			J & O			Max SLK	Prolong $Swap_{elit}$
Cost (SS_2)	Judged L	NA	A	$C_L=1 \mid C_H=1$	$C_L=1 \mid C_H=0.7$	Avg MEA	Crash/prolong $Swap_{elit}$
		Judged L	B & C	$C_L=0.9 \mid C_H=1.0$	$C_L=0.2 \mid C_H=0.7$	Avg MEA	Crash $Swap_{elit}$
			B & D				Prolong MS Slope
		Judged H	B & E	$C_L=1.0 \mid C_H=1.0$	$C_L=1.0 \mid C_H=0.7$	Avg MEA	Crash MC Slope $\mid$ Prolong MS Slope
			B & F				Prolong Swap
	Judged H	NA	G	$C_L=0.1 \mid C_H=0.2$	$C_L=0.3 \mid C_H=0.2$	Avg MEA	Crash MC Slope $\mid$ Prolong MS Slope
			H				Crash/prolong $Swap_{elit}$
			I				Prolong MS Slope
		Judged L	J & K	$C_L=0.5 \mid C_H=0.4$	$C_L=0.5 \mid C_H=0.6$	Avg MEA	Crash/prolong $Swap_{elit}$
			J & L				Consume slack
		Judged H	J & M	$C_L=0.6 \mid C_H=0.2$	$C_L=0.2 \mid C_H=0.2$	Avg MEA	Crash MC Slope $\mid$ Prolong MS Slope
			J &N				Enumerate
			J & O			Avg MEA	Consume slack

Table 2.7: Overview of the solution strategies and their components

2.6 Computational experiment

The computer experiment aims to reproduce the behaviour exhibited by the students on a diverse set of generated projects. The goal of this section is not to compete with existing exact and (meta-)heuristic approaches but rather to discern the circumstances in which each solution strategy reaches the best results. In fact, large cost deviations illustrate the limitations of human decision makers and identify the need for more involved optimization techniques. These more advanced techniques were discussed in the literature overview of section 2.1. The outline of this section is as follows. Section 2.6.1 provides details about the data generation process where a distinction is made between project-based parameter settings and settings related to the complexity and uncertainty. A baseline scenario is defined and will serve as the main vehicle to illustrate predominant relations for the different complexity and uncertainty combinations. Using this baseline scenario, the impact of judgement errors on the cost performance is studied. Finally, the effect of a varying level of effort is discussed.

2.6.1 Data generation

A distinction can be made between project-based parameter settings and settings related to the complexity and uncertainty. First of all, the project-based parameters will be discussed and the baseline scenario will be established. Afterwards, the settings related to complexity and uncertainty are divulged. All of these settings are summarised in table 2.8 and discussed in the following paragraphs. Table 2.8 is illustrated on a specific example and combined with the settings of one of the solution strategies in section 2.A.1 of the Appendix. The solution strategies were encoded in C++ on a Macbook featuring a 2.4GHz dual core processor with 4 GB RAM.

Project-based settings 100 project networks with 30 activities were generated using the RanGen2 generation engine (Vanhoucke et al. (2008)) for 9 values of the Serial/Parallel (SP) indicator, ranging from 0.1 to 0.9 in steps of 0.1. Although the SP indicator is named the I_2 indicator in the paper by Vanhoucke et al. (2008), it is commonly referred to as the SP indicator in several simulation studies (e.g. Vanhoucke (2010b)). The SP indicator measures a network's degree of closeness to a completely serial or parallel network (Tavares et al. (1999)). The following project-based parameter concerns the nature of the generated trade-offs. These can be random, linear, convex or concave. Convex trade-offs entail steeply increasing costs as an activity's duration is crashed. The opposite observation holds for concave trade-offs. We only consider random trade-offs.

Robustness checks were performed for linear, convex and concave trade-offs without leading to different results.

Imposing a deadline for the project is done after the instance is solved in an exact way, using the procedure of Demeulemeester et al. (1998). The exact solution method returns an efficient time/cost profile. Let D_{min} denote the minimum project duration and D_{max} denote the maximum project duration. The deadline, δ_n, is determined using the parameter θ, as follows:

$$\delta_n = D_{min} + \theta * (D_{max} - D_{min}) \tag{2.3}$$

3 levels for θ are suggested: 0.25, 0.5 and 0.75. Finally, the penalty parameter determines how extremely exceeding the deadline is discouraged. A low penalty setting (€350 per day) and high penalty setting (€3,500 per day) are taken into consideration. The height of the penalty has a direct influence on the global cost deviation. If a solution is reached with only a small time deviation but the penalty is set to a high number, the global cost deviation will be much higher than a situation with a low penalty setting. The combination of a certain value for the deadline and penalty determines the location of the optimal cost on the efficient time/cost profile. Finally, for every possible mode, a combination of durations and costs needs to be generated. The number of modes will be specified in the complexity and uncertainty settings paragraph. Activity costs range from €500 to €2,500 with a maximum allowed interval of €1,000 between two modes of an activity. The minimum durations of an activity go from 10 to 20 with a maximum interval of 1 time unit.

Baseline Scenario The baseline scenario is used as an instrument to identify the main effects of different combinations of the complexity and uncertainty parameter. The characteristics that closely resemble the PSG's properties were employed to construct this scenario. An exception is made for the penalty parameter, where both values were used.

Complexity and uncertainty settings Complexity refers to the average number of trade-offs of the different activities and was first introduced in section 2.5.1. There are two levels for the complexity of the generated projects. The activity modes of the projects are generated according to a triangular distribution with 1, 4 and 6 modes as the minimum, mode and maximum for projects with a low degree of complexity and 4, 7 and 9 modes for highly complex projects. There are two settings for the complexity

Description			Settings
Project parameters	SP factor		0.1-0.9, Δ=0.1
	#Projects		100
	Trade-offs		Random
	θ		0.25-0.75, Δ=0.25
	Penalty		€350-€3,500
	Activity Costs		∼R(500-2,500)
	Activity Durations		∼R(10-20)
Baseline scenario	SP factor		0.5
	θ		0.5
	Penalty		Low: €350 High: €3,500
Complexity & Uncertainty	Complexity	Low	∼Tri(1,4,6)
		High	∼Tri(4,7,9)
		Thresholds	0-10
	Uncertainty	Low	∼R(0.2-0.4)
		High	∼R(0.6-0.8)
		Thresholds	0-1.0

Table 2.8: Overview of the data generation parameters

threshold which determine if a project is judged to be complex or not. The first threshold setting is equal to 0, thus indicating that every project will be judged highly complex. The other parameter value equals 10 and implies that every project will be judged lowly complex.

The second dimension is uncertainty, which consists of 3 elements, namely the uncertainty type, proportion and size. The uncertainty type denotes the amount of positive and negative delays. A proportion of 90% positive delays (10% negative delays) is put forward. Negative delays result in activities finishing earlier than planned and are included to reflect that uncertainty can also yield opportunities (Ward and Chapman (2003)). Consequently, the uncertainty type penalizes or rewards risk takers who do not incorporate a lot of slack into their projects. Closely linked with the uncertainty type is the percentage of activities subject to a delay. This is applied to move the project's execution closer to or further away from the baseline schedule. The percentage of activities subject to a delay is applied to all activities at the start of the project's execution. Hence, an activity can only be delayed once. If this percentage is low, few unanticipated delays will distort the decisions about the activity modes taken by the project manager. Two levels for the uncertainty proportion are proposed. A low degree of uncertainty proportion corresponds with values that are drawn randomly from an interval with values between 20% and 40%. A high degree of the uncertainty proportion originates from a random draw with 60% and 80% as its lower and upper bound, respectively. The third component of the delays is the size of the delays. This is drawn from a triangular distribution and varies from 1 to 9, with the mode equal to 4. The parameter values for the uncertainty threshold are equal to 0 and 1.0. Hence, the project will be judged to be highly or only slightly uncertain, respectively. In the remainder of this chapter, the term uncertainty will be used for the proportion of delays, unless noted otherwise.

2.6.2 Results

In the previous section, the settings for the baseline scenario were discussed. This scenario will now be used to analyze the different links between complexity, uncertainty and how these dimensions are judged. The results for the baseline scenario are divided into 3 paragraphs. The first paragraph deals with the performance of both solution strategies when complexity and uncertainty are assessed correctly. Two situations may occur, namely when the complexity or uncertainty is low and when the complexity or uncertainty is high. The second paragraph takes a look at the two possible judgement errors. One of the dimensions, complexity or uncertainty, may be low in reality but can be judged high. Alternatively, the real complexity or uncertainty may be high but judged to be low. The significance results of a correct assessment and the judgement errors can be found in tables 2.9 and 2.10. In both tables an asterisk denotes a significant difference ($p < 0.05$). Table 2.9 deals with the significance results of the main experiment, whereas the judgement error results can be found in table 2.10. The third paragraph looks at how a higher level of effort impacts the cost performance of both solution strategies.

Performance Complexity refers to the average number of modes across all activities of the project. Uncertainty refers to the proportion of activities that are subject to a delay. The performance of the time-based solution strategy (SS_1) and the cost-based solution strategy (SS_2) is measured using the global cost deviation. In this paragraph, we limit ourselves to situations in which the decision maker judged the complexity and uncertainty correctly. The following observations with regard to the performance of both solution strategies can be made:

- The penalty costs for the time-based strategy (SS_1) are lower compared to those of the cost-based strategy (SS_2) across all complexity, uncertainty and penalty levels. This implies that the project duration attained by SS_1 lies closer to the deadline than it is for SS_2, resulting in a lower amount of incurred penalty costs.

Dimension	Actual & Perceived	Penalty	Global cost deviation			Penalty share		
			SS_1	SS_2	Sign.	SS_1	SS_2	Sign.
Complexity	Low	Low	11.65%	5.90%	*	2.10%	5.06%	*
		High	20.14%	27.32%	*	11.47%	25.21%	*
	High	Low	25.42%	13.13%	*	1.04%	6.30%	*
		High	25.61%	25.30%		4.82%	15.18%	*
Uncertainty	Low	Low	20.61%	13.08%	*	0.97%	3.84%	*
		High	21.67%	22.55%	*	4.56%	11.42%	*
	High	Low	17.97%	11.27%	*	1.94%	5.08%	*
		High	25.75%	28.12%	*	12.88%	21.57%	*

Table 2.9: Results of the main experiment (correct judgement)

- For a high penalty setting, a larger deadline deviation leads to a steep cost deterioration. It is no surprise that due to this increased importance of the timing aspect, SS_1 thrives in a high penalty setting.
- Even though SS_1 has a smaller share of penalty costs, the activity costs of SS_2 are much lower, indicating that a better trade-off selection takes place. The timing aspect does not have a substantial impact when the penalty is low. Hence, SS_2 almost always returns better results than SS_1. The difference between both strategies is more pronounced for a high degree of complexity.
- The complex search process for better trade-offs proves advantageous for a cost-based approach (SS_2). When the complexity is high, SS_2 is slightly but not significantly better even when the penalty is high. In that case, the proportion of penalty costs is larger than for SS_1 but the activity costs are much lower.
- When there is little uncertainty, SS_2 performs better or there is only a small difference compared to SS_1. Clearly, a low degree of uncertainty only has a minor impact on a project's duration.

Judgement Error A key topic in this chapter is the discrepancy between the real complexity or uncertainty and how it is judged. Hence, judgement errors can be made in which a dimension is low but judged to be high or vice versa. The results of these judgement errors lead to the following conclusions:

- A general conclusion for both strategies is that safety is the best policy. It is better to prepare for the worst and judge the complexity dimension to be worse (highly complex) than it may be in reality (low complexity).

Dimension	Actual	Perceived	Penalty	Global cost deviation			Penalty share		
				SS_1	SS_2	Sign.	SS_1	SS_2	Sign.
Complexity	Low	High	Low	13.48%	7.79%	*	1.88%	3.34%	*
			High	20.96%	18.84%	*	12.41%	13.04%	
	High	Low	Low	27.10%	21.52%	*	1.26%	3.41%	*
			High	28.84%	29.13%		6.53%	11.92%	*
Uncertainty	Low	High	Low	19.07%	11.25%	*	1.03%	4.27%	*
			High	21.04%	21.13%		5.47%	12.30%	*
	High	Low	Low	20.01%	12.74%	*	2.33%	4.93%	*
			High	27.09%	28.79%	*	12.33%	20.06%	*

Table 2.10: Results of the main experiment (judgement error)

- For a high penalty setting, the time-based approach (SS_1) performs well compared to the cost-based approach (SS_2).
  - The preferred solution strategy depends on the (actual and perceived) complexity of the judgement error. Even though the time-based strategy (SS_1) performs slightly better than the cost-based strategy (SS_2) when the complexity is high but judged to be low, the difference was found to be statistically insignificant. When the complexity is low but judged to be high, the cost-based strategy comes out on top.
  - When it comes to the uncertainty dimension, SS_1 performs slightly better than SS_2 for both judgement error positions. Interestingly, the activity costs are lower for SS_2 but the higher penalty costs push the global cost deviation of SS_2 higher than that of SS_1.
- For a low penalty setting, the cost-based strategy (SS_2) clearly outperforms the time-based strategy (SS_1).

Influence of the Level Of Effort In order to ensure that no large differences in the level of effort materialize for the generated projects, the level of effort was controlled using focus and intensity. The focus and intensity settings for the baseline scenario were described in section 2.5.2. In this section, the effect of an increased level of effort is studied. Three separate experiments were conducted to study the effect of an increased level of effort on the performance of the solution strategies. The main findings can be summarized as follows:

- The first experiment adopted a focus of 100% in absence of any uncertainty (U = 0). The intensity was varied from 0.6 to 1.0 in steps of 0.1. The results indicate that an increased intensity leads to better cost deviations.

- The second experiment adopted an intensity of 100% in absence of any uncertainty (U = 0). The focus was varied from 0.6 to 1.0 in steps of 0.1. Similar to the first experiment, the global cost deviation decreased as the focus was increased, but the decrease was less steep compared to the findings of the first experiment.

- The last experiment reintroduced the uncertainty settings of the baseline scenario. The focus was kept at 100% and the intensity was varied again from 0.6 to 1.0. Hence, compared to the first experiment, these settings allowed us to explore the influence of the uncertainty. As the intensity (and thus the level of effort) increased, the global cost deviation decreased. However, the cost deviations are higher than those of the first experiment, which can be attributed to uncertainty affecting the activity durations.

Finally, we have also tested the influence of the deadline and the SP level. The deadline parameter was varied by selecting the 25th and 75th percentile. Both solution strategies share a decreasing deadline deviation trend as the deadline increases, without leading to different conclusions for the overall performance. The SP factor was varied from 0.1 to 0.9 with steps of 0.1. No consistent trend for the solution strategies across the complexity and uncertainty dimensions could be established.

2.7 Discussion and conclusion

In this chapter, three contributions were made. First, the decisions of students throughout the Project Scheduling Game, a project management business game, were analyzed. We learned that two major solution strategies could be discerned. These are comprised of five building blocks, namely focus, activity criticality, ranking, intensity and action. The first solution strategy focuses on time and employs three mechanisms to approach the deadline. The Greatest Rank Positional weight priority rule is used, as well as a buffer based on the slack duration ratio of Hazır et al. (2010) and a final check to protect the deadline is performed. The second solution strategy heavily focuses on costs, at the expense of an increased exposure to risk. The Average Most Expensive priority rule is used to rank activities. Elitism is applied to only accept cost improvements and non-critical activities' slack is consumed to a larger degree.

Second, complexity and uncertainty were included as contextual factors. The literature overview of section 1 indicated that these are dominant themes and that a link between complexity and project outcome (Hanisch and Wald (2011)) and a continued study of uncertainty (Hall (2012)) were among the challenges for future research. To that end, we have conducted a large computational experiment that allows us to quantify the impact of complexity (Maylor et al. (2008)) and uncertainty. The following conclusions can be drawn from the experiment:

- A high degree of complexity has a negative effect on the cost deviation. Since heuristics are designed to make a trade-off between effort and accuracy (Gigerenzer and Gaissmaier (2011)), complex situations call for either more advanced solution methods or for an increase in additional resources and managerial attention, as established by Shenhar (2001).

- The effect of uncertainty greatly depends on its impact, which was regulated using the penalty parameter in our experiments. A high degree of uncertainty combined with a severe penalty for deadline overruns led to steep cost increases. Hence, the importance of meeting the deadline, for which the penalty acts as a proxy, determines the extent to which uncertainty hampers the project's objective.

- Individuals vary in how they assess complexity and uncertainty. An experienced project manager will employ a different threshold for determining the degree of complexity and/or uncertainty compared to a recent project management graduate. This process was imitated by means of judgement errors, where different thresholds were employed, steering the logic of the solution strategies into a different direction. We came to the conclusion that the direction of judgement errors is crucial. Perceiving a project as highly complex and uncertain while this is not true in reality yields significant advantages compared to the opposite scenario. Hence, we recommend project managers who are incapable of correctly assessing a project's complexity and/or uncertainty (e.g. through limited information) to err on the safe side.

- We identified the conditions in which each solution strategy thrives. The time-based solution strategy performs particularly well when deadline overruns are heavily penalized and in highly uncertain environments. The cost-based solution strategy yields better results in low penalty and highly complex environments.

- Increasing the level of effort exhibits a positive effect on the capability of the strategies.

In this chapter, we focused on quantifying the effect of complexity and uncertainty on cost outcomes. Hence, the limitation of this paper is that little attention was paid to the behavioural and psychological aspects of complexity and uncertainty. For instance, one can wonder what the effect of tight deadlines on team motivation is and how this relates to previous research on this topic (Chang et al. (2003), Engwall and Westling (2004)). Additionally, demographic variables such as age, background and project role (cf. Ojiako et al. (2014)) could be included, especially when dealing with complexity and uncertainty judgements.

From a data analysis and model perspective, two future research avenues can be identified. First of all, while we provided an initial analysis to discern between two major solution strategies, it would be interesting to find out if participants of the PSG switch between strategies throughout the game and by which circumstances this switch is prompted. A similar question arises for niche strategies. Secondly, additional mechanisms can be put in place that further complicate the decision-making process. For instance, increasing or decreasing the time participants have to make decisions throughout the PSG, as well as the presence of a contingency budget may well lead to different choices. Another possibility could be the inclusion of decision moments that are not spread equally along the time dimension. Studying this in conjunction with the project's network topology would be a meritorious extension.

2.A Appendix

This appendix contains results or clarifications that were either too expansive to add to the main text of the chapter or did not alter its main insights. It provides an example of the data generation process (section 2.6.1) and the link with table 2.8 of the main text.

2.A.1 Data Generation example

In this section, the data generation of the computational experiment, found in section 2.6.1, is illustrated by means of an example of the dataset. The different steps are outlined below:

1. *Network Generation* - Generate a network with 32 activities and a value of the SP indicator equal to 0.8. The Activity on the Node (AoN) representation is given in figure 2.A.1.

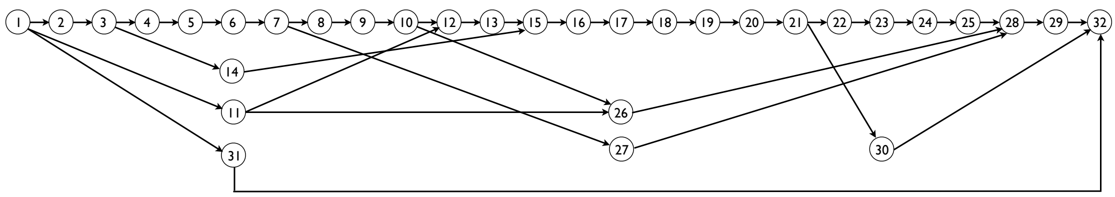


Figure 2.A.1: Activity on the Node representation of the generated network

2. *Generate time/cost trade-offs* - In this example, a low complexity will be maintained. For each activity, the number of trade-offs is drawn from a triangular distribution with 1, 4 and 6 as the minimum, mode and maximum respectively. Each trade-off has a duration between 10 and 20 time units and a cost between 500 and 2,500 monetary units. A full overview of the generated time/cost trade-offs for all activities can be found in table 2.A.2.

3. *Generate delays* - the uncertainty proportion determines the amount of activities that will be subject to a delay. In this instance, a value of 0.2 is generated, implying that 0.2 * 32 $\approx$ 6 activities will be delayed. The size of the delays is drawn from a triangular distribution with 1 and 9 as the minimum and maximum and a mode equal to 4. An overview of the activities that are delayed is shown in table 2.A.1.

4. *Generate deadline and penalty* - the example is solved exactly, resulting into an efficient time/cost profile. Since no penalty is imposed yet, lengthening the project

Activity	Delay
6	6
8	5
10	4
13	6
15	3
25	5

Table 2.A.1: Generated delays for the data generation phase

leads to cost reductions. The deadline is set to 0.5 in this example, which corresponds with the time value of the 20^{th} (0.5 * 40) point of the efficient time/cost profile. For every day the deadline is exceeded, a penalty cost of 350 monetary units is incurred. Figure 2.A.2 shows how the penalty affects the efficient time/cost profile.

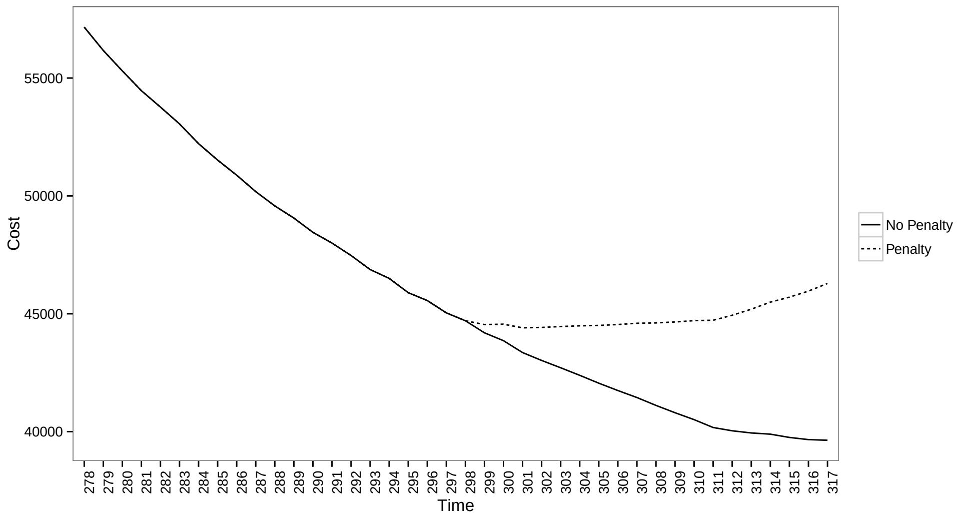


Figure 2.A.2: Efficient time/cost profile with and without the penalty of 350 monetary units

5. *Apply one of the solution strategies* to the problem at hand. In this example, the actual complexity is low. Assume that the thresholds for complexity and uncertainty are equal to 0. In that case, complexity and uncertainty will be judged low. This implies that for the time-based strategy, the focus will be equal to 0.4 and the intensity will be 0.5. The GRPW priority rule will be invoked. Critical activities will be crashed according to the minimum cost slope, whereas non-critical activities will be prolonged following the maximum savings slope. The global cost

deviation of the time-based solution strategy for this example is equal to 16.12%, with the activity cost making up 100% of the global cost deviation. Hence, the time-based solution strategy manages to finish before the project's deadline.

Activity	Trade-off info							
	Trade-off 1		Trade-off 2		Trade-off 3		Trade-off 4	
	Time	Cost	Time	Cost	Time	Cost	Time	Cost
1	15	1485	16	1400	17	929	18	543
2	15	2421	16	1431	17	1194	18	593
3	10	4367	11	3443	12	2527	13	1949
4	16	2935	17	2848	18	2078	19	1986
5	17	3489	18	3095	19	2100	20	1241
6	19	2899	20	2588	21	2444	22	1961
7	15	3897	16	3029	17	2190	18	1494
8	17	1310	18	777				
9	14	2324	15	1925	16	1739	17	906
10	14	1894	15	1331	16	623		
11	17	949	18	615	19	587		
12	18	1691	19	1428	20	1093	21	726
13	17	1410	18	516				
14	18	2377	19	1440	20	1026		
15	10	2921	11	2912	12	1990	13	1344
16	17	732						
17	17	3794	18	3516	19	2869	20	1897
18	17	3298	18	3092	19	2176		
19	16	3375	17	2769	18	2616	19	1923
20	14	2209	15	1786	16	1491		
21	17	1553	18	1549	19	1517	20	1271
22	18	1453	19	1258	20	1160	21	619
23	18	762						
24	12	3138	13	2320	14	1817	15	885
25	13	3441	14	2965	15	2104		
26	17	2657	18	2246	19	1663	20	872
27	11	2485	12	2393	13	2389	14	1637
28	10	2240	11	2011	12	1258	13	1119
29	15	3524	16	2784	17	2508	18	1769
30	15	2025	16	1116				
31	13	4820	14	4164	15	3279	16	2429
32	14	560						

Table 2.A.2: Generated time/cost trade-offs for all activities

3

Effort-based decision making for the Discrete Time/Cost Trade-off Problem

In this chapter, a method for constructing custom solution strategies to the Discrete Time/Cost Trade-off Problem is presented. Schedule focus, activity focus and action radius constitute the components of crafting such a strategy. The solutions are evaluated in a student learning environment with a limited amount of effort. The effort-based decision-making process has been tested in a classroom experiment by analyzing the student strategies. Additionally, a dataset for computational experimentation has been generated. We report on the influence of different parameter settings on various aspects of the solution quality as well as on how the strategic components affect the attained solutions.

3.1 Introduction

With the advent of project scheduling in the 1950s, the well-known Critical Path Method (CPM) came into existence following development at the duPont Company and at Remington Rand Univac (Kelley and Walker (1959), Walker and Sawyer (1959) and Kelley (1961)). The CPM represents tasks according to precedence relations and aims to construct a plan, which serves as the baseline schedule for project control efforts. An inherent characteristic of CPM consists of time/cost trade-offs accompanying each activity. The underlying assumption is that as an activity's duration is shortened, more resources need to be allocated. Typical real-life examples are additional manpower or funds that need to be invested in order to bring about a decrease in duration. Initial research focused on activity costs being a continuous, linear and non-increasing function of its duration. Research related to the continuous case, as well as its Linear Programming model can be found in the texts of Elmaghraby (1977) and Moder et al. (1983). A multitude of techniques were proposed to solve these types of problems, such as dynamic programming (Robinson (1975), Hindelang and Muth (1979)), minimal cuts (Phillips and Dessouky (1977)) and mixed integer linear programs (Meyer and Shaffer (1965)). Varying the type of an activity's cost function to assume concave, convex or linear slopes proved to be another salient avenue for exploring. The literature of the time/cost trade-off problem has been summarized by De et al. (1995). The same authors showed that the discrete time/cost trade-off problem (DTCTP) is NP-hard (De et al. (1997)). Three variants of the DTCTP are discerned within academic literature. The deadline problem (DTCTP-D) schedules activities with the goal of minimizing the project's cost while meeting an imposed deadline. The budget problem (DTCTP-B) minimizes the duration of the project, subject to a certain budget that is spent on activity costs. The last variant of the DTCTP can be seen as a combination of the latter two variants by constructing a complete, efficient time/cost frontier. The three variants have been solved exactly by Demeulemeester et al. (1996) using two dynamic programming approaches. The computational results were improved by Demeulemeester et al. (1998) with a branch-and-bound procedure. The lower bounds are calculated through the convex, piecewise linear underestimation of the time/cost trade-off curves. An adaptation of the labeling algorithm of Fulkerson (Fulkerson (1961)) computes a lower bound with those underestimations. The procedure of Demeulemeester et al. (1998) is of special relevance to this chapter since it will be used to calculate the optimal solution for the real-life instance of the empirical experiment of this chapter.

From the 2000s onwards, two novel research directions have been examined. The first direction expands on extensions of the (D)TCTP. Time-switch constraints were incorporated by Vanhoucke et al. (2002) and Vanhoucke (2005). Vanhoucke and Debels (2007) presented a metaheuristic procedure for time-switch constraints, work continuity and net present value maximization. Another extension centred on the time/cost/quality problem. Integer programming formulations (Tareghian and Taheri (2006)), as well as metaheuristic solutions have been proposed to solve this problem (Pour et al. (2010) and Tareghian and Taheri (2007)). Real-life analyses of these types of problems were performed by El-Rayes and Kandil (2005) and Zhang and Xing (2010). Choi and Chung (2014) analyzed the complexity of the linear TCTP with milestone objectives and completely ordered jobs. The second research direction consists of including stochastic characteristics. Azaron et al. (2005) and Azaron and Tavakkoli-Moghaddam (2007) provide solutions for activities following a generalized Erlang and exponential distribution respectively. The stochastic TCTP has been solved using robust optimization (Cohen et al. (2007)), a genetic algorithm (Ke et al. (2009)) and an ant system approach (Mokhtari et al. (2011)). Schedule robustness, as well as other robustness measures were studied by Hazır et al. (2011, 2010).

A different approach towards the DTCTP was taken by Vanhoucke et al. (2005), who presented a business game entitled the Project Scheduling Game (PSG). The underlying project network is a real-life instance of the DTCTP-D. However, the deadline constraint is soft. Exceeding the deadline results in a penalty but does not render the problem infeasible. There are multiple decision moments. At each decision moment, a report on the time and cost performance of activities that were finished in the previous decision moments is given. The participants have the option to modify the ongoing or future activities in light of this feedback report. As such, the participants go through a cycle of scheduling, feedback and, if necessary, rescheduling. The goal of the game is twofold. On the one hand, it provides an experiential learning environment displaying the need for risk analysis. On the other hand, the game demonstrates the careful balance between complexity and uncertainty. Faced with approximately 40 activities with on average 5 trade-off options, participants are overwhelmed with alternative choices. Hence, an incentive to rely on state-of-the-art optimization algorithms is provided. On the other hand, sole reliance on optimization without any thought for uncertainty proves futile. As unexpected events create delays or offer opportunities, the project's execution will differ from the baseline schedule. Chapter 2 (Wauters and Vanhoucke (2013)) expands on this careful balance between complexity and uncertainty, providing a time-based, as well as a cost-based solution strategy. The authors identify the circumstances in which each strategy flourishes, making a distinction between the actual and perceived complexity and uncertainty.

This chapter presents PSG Extended, an extension to the project scheduling game, and has a twofold goal. First, a novel and modular way of constructing solutions to the DTCTP is presented. This was achieved in a classroom experiment, in which students needed to make decisions with regard to a number of parameters. Based on these parameters, a solution strategy is assembled and tested on the project at hand. The method for constructing solutions can be divided into three components, namely schedule focus, activity focus and action radius. Schedule focus selects a set of activities based on a specified time window. Activity focus is subdivided into activity ranking and time/cost focus, governing the order in which activities' trade-offs will be changed and the amount of trade-offs that are susceptible to a change. Action radius consists of the deadline focus, slack consumption, cost/benefit analysis and effort loading. Deadline focus and slack consumption decide whether changes to critical or non-critical activities take place. Cost/benefit analysis determines whether and to which extent cost improvements or deteriorations are allowed. Finally, effort loading allows the decision-maker to emphasize certain decision moments along the progress of the project. Since the solution strategies are subjected to a maximum allowed total effort, participants need to determine where they want to spend the bulk of the effort. The second goal of this chapter is to make recommendations for the DTCTP in light of this limited amount of effort. The managerial implications are crucial. Based on a company's project portfolio, a different degree of effort is assigned to each project. This effort can be measured in units of time or in terms of budget restrictions. As an example, the PSG of Vanhoucke et al. (2005) employed a limited time window to make a new decision. Without loss of generality, effort will refer to a dimensionless unit in the remainder of this chapter. By employing empirical as well as computational results, the effects of a different availability of effort, as well as how different characteristics of the DTCTP influence these, will be discussed. It is worth noting that this chapter has no intention to compete with state-of-the-art solution procedures but rather to develop a modular manner in which solution approaches can be constructed given limited amounts of dispensable effort.

The outline of this chapter is as follows. In section 3.2, PSG Extended, the effort-based extension to the Project Scheduling Game, is proposed. This is done by explaining the goal of this business game extension, as well as clarifying the strategic components for constructing a solution to the DTCTP. Section 3.3 elaborates on how the strategic components are combined into a solution procedure.

PSG Extended was implemented in 2014 and has led to data of 75 groups of students. Section 3.4 focuses on the progress of the game and the main findings based on the student strategies. Section 3.5 applies thousands of solutions to a computer-generated and diverse dataset. Details on the test design are provided in section 3.5.1, where attention is given to the parameter settings and the procedure to set the effort threshold. Section 3.5.2 links characteristics of the DTCTP with various aspects of the solution quality, such as project cost, project duration and consumed effort. The impact of the effort percentile (section 3.5.2.2), deadline (section 3.5.2.2), penalty (section 3.5.2.2), uncertainty (section 3.5.2.4) and topological structure (section 3.5.2.5) is scrutinized. Section 3.5.3 links the settings for the key components of the solutions to the project outcome. Finally, conclusions are drawn in section 3.6.

3.2 The effort-based Project Scheduling Game

The effort-based Project Scheduling Game is an extension to the Project Scheduling Game proposed by Vanhoucke et al. (2005). The goal of the original PSG is to minimize the total project cost, which is comprised of two parts. The activity costs result from the chosen time/cost trade-off option for every activity, while the penalty costs are the consequence of not meeting a predefined deadline. Hence, the penalty costs only affect the cost objective when the project duration exceeds the deadline. The PSG has been studied in chapter 2 (Wauters and Vanhoucke (2013)), where solution strategies are derived from the analysis of participants' data. PSG Extended is introduced after participants have played the PSG and acquired insights related to complexity, uncertainty and risk. While the project, the number of decision moments and the inherent uncertainty are equal to that of the PSG, the intent of the effort-based extension is completely different. Participants are required to design a custom solution strategy by means of a number of components. A decision on the components and their settings has to be made prior to the project's execution, implying no changes can be made throughout.

However, the settings of the components will be used in a project control environment where trade-offs of activities are changed based on the settings for each of the components. More specifically, the participant has to decide on settings for 7 elements that are categorized into three strategic components (cf. sections 3.2.1-3.2.3). Incidentally, the participant may vary the settings of every element throughout the decision moments. The settings for the 7 elements along 6 decision moments are then collected and assembled to form a solution strategy. This custom strategy is tested on the project network of the PSG, with an identical degree of uncertainty. Afterwards, the uncertainty, as well as the deadline and penalty are modified to reflect the capability of each strategy to deal with this change in circumstances.

The context of PSG Extended can be described as follows. Participants of the game are in charge of the same project of the PSG and faced with a limited amount of total effort. The presence of an effort restriction forces each participant to make a choice on how to spend the available effort. To that end, a number of inputs can be changed, often leading to a change in the consumed effort. The main rationale is that as the inputs are changed, different trade-off options for some of the activities are chosen. In turn, this leads to a different effort consumption pattern. Similarly to the original PSG, a choice for the inputs has to be made for each decision moment. However, a crucial difference lies in the feedback that participants receive. In PSG Extended, participants only receive an update on the cumulative effort consumed across all decision moments. Throughout the original PSG, participants receive an update on the time and cost performance after each decision moment. Hence, they can review their approach after every decision moment.

An overview of the inputs that can be tuned while respecting the effort threshold is depicted in figure 3.1. The top of figure 3.1 displays an example network with 6 Decision Moments (DM) and 8 activities which are scheduled as soon as possible. The inputs are divided into three components, namely schedule focus, activity focus and action radius and will be referred to as strategic components in the remainder of this chapter. They will be briefly discussed along the following lines, while a more thorough explanation is given in the following sections.

- Schedule Focus: typically, schedule focus is the first component to which a change is made. It determines if the participant wants to focus on the short term or long term. Schedule focus selects a set of activities based on a time window. If this window is small (a local schedule focus), the activities of the current decision moment will be inserted into the set of activities. However, it is possible that the participant wishes to take future activities into account, in which case the time window will be expanded and can range from a few to all decision moments (a global schedule focus).

- Activity Focus: the ultimate goal of PSG Extended is to assemble a strategy that is

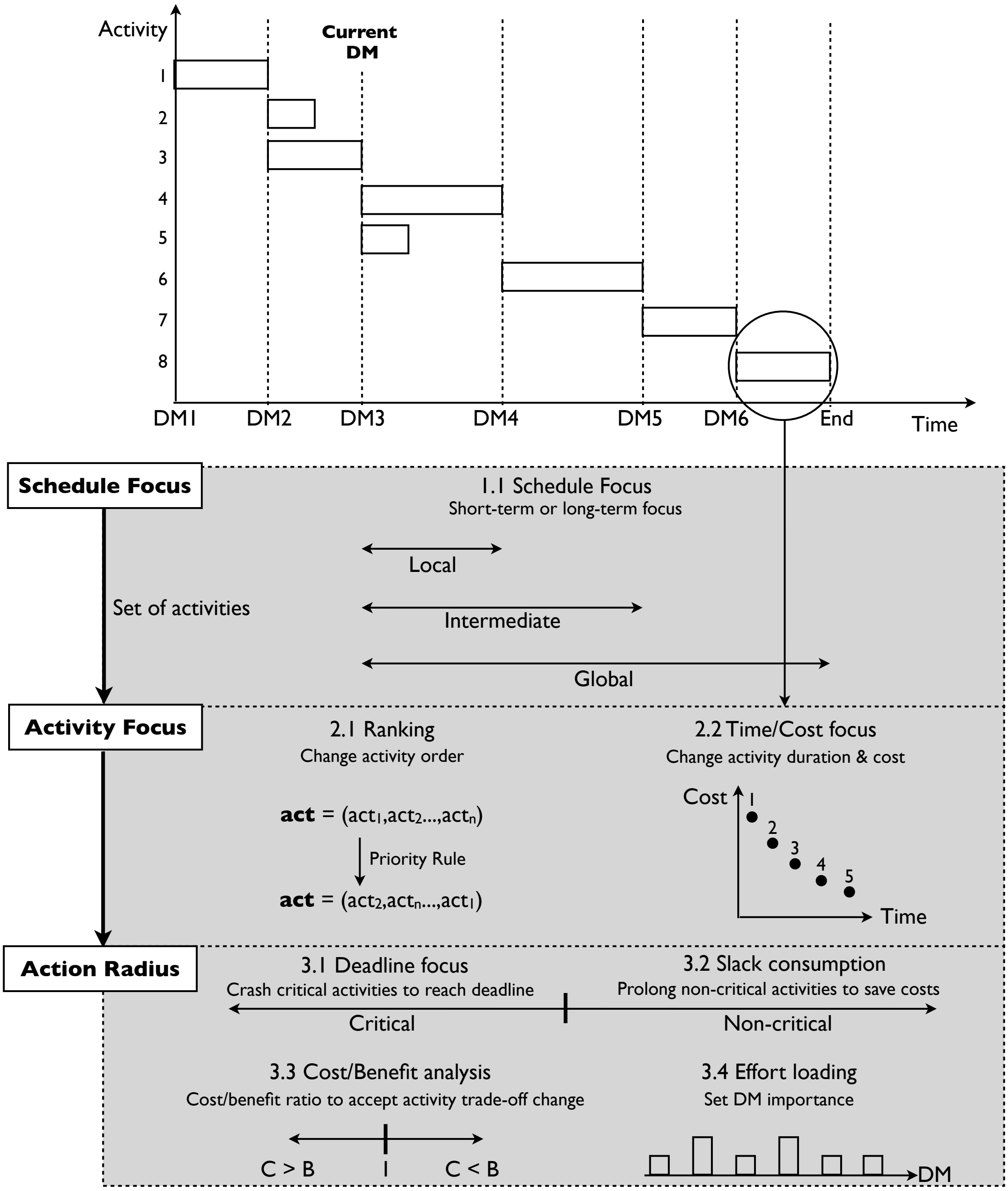


Figure 3.1: Overview of the inputs of PSG Extended

in line with the wishes of the participant throughout the project's execution. The execution may distort a project's baseline schedule, resulting in undesirable situations that can be remedied by taking action. Within the context of the DTCTP, this entails changing the trade-off of a number of activities. The order and the number of trade-off options that are evaluated make up the core of the activity focus input. Starting from the set of activities defined in the schedule focus phase, the activities are ranked according to a priority rule, after which the time/cost focus determines the range of trade-off options that will be evaluated.

- Action Radius: while schedule focus and activity focus determine a set of activities and a range of trade-off options respectively, action radius determines a number of strategic aspects of the project. It answers questions related to the importance of costs versus risk, whether only cost improvements are allowed and if some decision moments are more important to the participants than others.

The participants are required to cleverly combine these components by setting the parameters for each component for every decision moment. The combination of components with specific settings results in a solution strategy, which is tested in a number of ways:

- Comparison PSG-PSG Extended: the results from the solution strategy of PSG Extended are compared to the overall results of the original PSG. For the original PSG, the changes that are made to the activities' trade-offs are very specific. However, the solution strategy that participants build throughout PSG Extended are general and construct a solution strategy that can be tested on numerous projects with a different structure or different characteristics.

- Impact of uncertainty: the degree of uncertainty is varied to assess which solution strategies are better able to cope with this change.

- Project characteristics: the deadline and height of the penalty are modified to diversify the ease of reaching the deadline and to vary the cost implications of incurring a penalty.

- Impact of network structure and starting position: a large amount of computer-generated solution strategies are tested on a diverse dataset such that general conclusions and managerial insights can be provided. For each network, the trade-offs at the start of the project's execution are varied to examine the relation between the starting position and other project characteristics.

Results from PSG Extended are analyzed in section 3.4, in which the impact of uncertainty and several project characteristics are varied as well. While section 3.4 utilizes the network of the original PSG, the computational study of section 3.5 subjects the solution strategies to different network structures. Additionally, the starting position is changed and the interrelation with uncertainty and project characteristics is highlighted. In the following sections, a more detailed description of the strategic components (schedule focus, activity focus and action radius) is offered.

3.2.1 Schedule Focus

Schedule focus is the first strategic component and allows the participant to adopt a short-term or long-term perspective. Schedule focus selects a subset of the ongoing or future activities by means of a time window. The strategic component of section 3.2.2 utilizes this subset to search for the most profitable trade-off change for the activities within the subset. In order to explain the underlying concept, we refer to the top of figure 3.1. A Gantt-chart of an illustrative project is displayed. The project counts 8 activities and there are 6 decision moments. We assume that the participant needs to make a decision for decision moment 3. If the participant wishes to focus on the short term, it makes sense to only include the activities of the current decision moment, i.e. activities 4 and 5. However, it is possible to take future activities into account by expanding the time window to include activities 6, 7 and 8 respectively. The possibilities the participants have at DM 3 are listed in table 3.1.

As mentioned previously, the subset of activities will be searched for the most profitable change in the following strategic component. Hence, the concept of schedule focus is closely related to that of a local versus a global search. A limitation is imposed on the allowed effort that can be consumed by the solution strategy of the participant. As a result, a trade-off needs to be made between a local search (lower degree of schedule focus), resulting in less consumed effort, or a global search, which consumes more effort. By expanding the time window to include future decision moments and more activities, larger cost reductions may be attained. However, this comes at the cost of an increased effort consumption and limits the participant's options in future decision moments.

3.2.2 Activity Focus

Activity focus is the second strategic component and departs from the subset of activities that results from the schedule focus of section 3.2.1. Activity focus is comprised of two elements, namely ranking and time/cost focus. The outcome of this strategic component

Schedule focus		
Schedule focus Setting	Time window	Activities in subset
1	DM 3	{4,5}
2	DM 3-4	{4,5,6}
3	DM 3-5	{4,5,6,7}
4	DM 3-6	{4,5,6,7,8}
Activity focus		
Time/cost focus		Trade-off option to consider
0.25		{4}
0.5		{3,4}
0.75		{2,3,4}
1.0		{1,2,3,4}
Action radius		
Cost/benefit analysis		
Activity	Ratio	Threshold
Critical	$\frac{P\downarrow}{C\uparrow}$	$> \ p, p \in \{0.7, 0.9, 1.0, 1.1, 1.3\}$
Non-critical	$\frac{C\downarrow}{P\uparrow}$	

Table 3.1: Overview of the participant's possibilities for the schedule focus at DM 3, time/cost focus for activity 8 and cost/benefit analysis

is a proposal for a trade-off change for every activity. Whether this change is acceptable will be determined by the third and final strategic component, namely action radius (cf. section 3.2.3).

We refer to figure 3.1 to explain the purpose of ranking and time/cost focus. Let **act** denote the vector of activities that results from the schedule focus component. The *ranking* phase changes the order of the n elements of **act** by applying a priority rule. When trade-off changes are executed, this will be done according to the activity order that follows from applying this priority rule. The order in which adjustments to activities are made can alter the critical path and can lead to a different project duration. The ranking phase distinguishes between critical and non-critical activities. Participants may choose from the priority rules found in table 3.2[1]. The second element of activity focus is the *time/cost focus*. While schedule focus and ranking operate on the project level, time/cost focus inspects the trade-offs of individual activities. Time/cost focus is illustrated in figure 3.1 and the middle part of table 3.1. Suppose activity 8 has 5

[1]The interested reader is referred to `http://pmknowledgecenter.com/dynamic_scheduling/baseline/optimizing-regular-scheduling-objectives-priority-rule-calculations` for an explanation as well as sample calculations of these priority rules. Minimum criticality sorts activities based on the amount of critical predecessors in increasing order.

Critical activities		Non-critical activities	
Abbreviation	Description	Abbreviation	Description
Random		Random	
MIS	Most Immediate Successors	LIS	Least Immediate Successors
LPT	Longest Processing Time	SPT	Shortest Processing Time
MinCrit	Minimum Criticality	MinSlack	Minimum Slack
MaxCrit	Maximum Criticality	MaxSlack	Maximum Slack
GRPW	Greatest Rank Positional Weight	SRPW	Smallest Rank Positional Weight

Table 3.2: Overview of the priority rules for critical and non-critical activities

different trade-off options as shown in figure 3.1. Furthermore, we assume activity 8 is critical and that the activity's duration will be crashed. The time/cost focus specifies the number of neighbouring trade-off options that will be searched such that the most profitable trade-off change can be executed. A low degree for the time/cost focus implies that only one of the neighbouring trade-off options will be considered. The available values for time/cost focus are shown in table 3.1. The trade-off options to be considered, found in the second column of table 3.1, correspond with the numbers above each point of the time/cost profile of figure 3.1. As the time/cost focus increases, more time/cost combinations are evaluated. Similar to the schedule focus, an increase in time/cost focus comes at the expense of an increased effort consumption. For critical activities, the trade-off option resulting in the minimum cost slope will be proposed, while the maximum savings slope for non-critical activities is suggested. Whether the trade-off change is actually executed, depends on the cost/benefit analysis parameter of the third and final strategic component, namely action radius. This component is explained in the following section.

3.2.3 Action Radius

Action radius is the final strategic component and translates strategic matters such as the importance of cost and risk into specific parameters. Action radius consists of 4 elements, namely deadline focus, slack consumption, cost/benefit analysis and effort loading. Each of these elements will now be explained.

Deadline focus and *slack consumption* allow the participant to indicate the importance of the deadline and costs respectively. If the deadline is an important objective, more critical activities will be crashed in order to bring the expected project duration closer to the predefined deadline. When the slack of non-critical activities is consumed, costs are saved but at the same time, the project's risk rises. If the settings of deadline focus and slack consumption are equal, the time aspect (reaching the deadline) is judged to be as important as cost savings (prolonging non-critical activities). As a result, every change to a critical activity will be followed by a change to a non-critical activity. The setting for deadline focus and slack consumption is converted to a score, in which "Low"" corresponds with a score of 1, "Medium" with 3 and "High" with 5. If the scores of deadline focus and slack consumption are not identical, their scores indicate how many consecutive moves on a critical activity (the deadline focus score) will be followed by consecutive moves on non-critical activities (the slack consumption score).

The third element of action radius is *cost/benefit analysis* and evaluates whether the proposed trade-off change that follows from the time/cost focus of section 3.2.2 is acceptable or not. This is achieved by comparing the cost/benefit ratio to a threshold value. When the ratio exceeds the threshold, the trade-off change will be executed. First, we will explain the ratio for critical and non-critical activities. Next, the threshold value and its significance will be discussed. We refer to table 3.1 for a summary of the cost/benefit analysis.
Critical activities are crashed according to the minimum cost slope, while non-critical activities are prolonged according to the slope that yields maximum savings. In both cases, costs and benefits may be incurred. When a critical activity is crashed, the activity costs, denoted by C in table 3.1, rise. On the other hand, the decreased duration of that critical activity may entail a lower penalty cost ($P \downarrow$) if the project's duration was lowered.
Non-critical activities are prolonged according to the maximum savings slope. As an activity's duration is increased, the activity costs decrease ($C \downarrow$). However, if the increase in duration consumes all available slack, the activity becomes critical and may lead to an additional penalty cost ($P \uparrow$) if the project's duration has risen.
The ratio of benefits and costs is compared to a parameter that is set by the participant and may differ in value for each decision moment. When the ratio exceeds the value of p, the proposed trade-off change is carried out. Otherwise, no change to the activity is made. Participants can choose from 5 values for p (provided in table 3.1) for each of the 6 decision moments. p should be interpreted as follows:

- $p = 1$: the benefits are equal to the costs. Before an activity's trade-off is changed, the benefits should surpass or be equal to the costs. Hence, only cost improvements or cost-neutral changes are allowed.

- $p < 1$: the costs are allowed to exceed the benefits. Small cost deteriorations are accepted.
- $p > 1$: the benefits need to outweigh the costs. Only improvements larger than p will suffice.

Participants are required to make a choice on the parameter settings for each element of the three strategic components, while operating under an effort restriction. The cumulative effort across all decision moments cannot exceed 100%. The final element of the action radius component allows participants to indicate the importance of each decision moment. *Effort loading* applies a correction to the effort threshold for individual decision moments, resulting in a higher allowed effort consumption for important decision moments and less room for decision moments that are not judged to be important by the participant.

3.3 Solution procedure

The previous section detailed the three strategic components, as well as the elements each component is composed of. Once the parameter settings for the 7 elements and for the 6 decision moments have been set, the components are combined into a solution approach. The aim of this section is to explain how the elements are assembled and which element impacts which part of the solution procedure.

An overview of the solution procedure can be found in algorithm 1. As mentioned previously, participants need to make parameter choices. These are constrained by a cumulative effort, which is divided across the decision moments. Based on the participant's preference, the distribution of effort across decision moments can be altered. This is achieved by means of the *effort loading* element. As a result, the effort threshold depends on the effort loading parameter. Once this is done, the activities that fall within the *schedule focus* are determined. The activities of this subset will be used in the subsequent part of the solution procedure.

As long as the stop criterion is not met, trade-off changes occur. This will be explained later on in this section. Whether a potential change to a critical or non-critical activity is investigated depends on the *deadline focus* and *slack consumption* parameters. When both elements assume the same value, reaching the deadline is as important as consuming slack. Hence, every change to a critical activity will be followed by a change to a non-critical activity. However, if reaching the deadline is more important, more changes to critical activities will be made. The (critical or non-critical) activities are *ranked* according to the selected priority rules of table 3.2. Next, a search for the most profitable trade-off change is conducted. Critical activities are crashed according to the minimum cost slope, while non-critical activities are prolonged along the maximum savings slope. This search for the smallest or largest slope is limited by the *time/cost focus*. Once the best trade-off option has been found, a check on whether the proposed change exceeds the *cost/benefit analysis* parameter p takes place.

Activity trade-offs are changed as long as a stop criterion is not met. The process of modifying trade-offs is stopped if one of two conditions is met. The first condition relates to the effort threshold. Once all effort is consumed, the procedure moves to the next decision moment and the effort threshold is reset based on that decision moment's effort loading value. The second condition, no crashing and no prolonging in algorithm 1, checks whether the activities of the schedule focus subset can still be crashed (for critical activities) or prolonged (for non-critical activities). If none of the critical or non-critical activities can be crashed or prolonged respectively, no additional trade-off changes can take place. This process continues until the final decision moment. The stop criterion ensures that the effort for every decision moment and for the project as a whole is not exceeded.

Algorithm 1: Assembly of the strategic components into a solution procedure

```
Data: Settings for the elements of the strategic components
for every decision moment do
    Set effort threshold
    Set focus activities
    stop criterion ← false
    while stop criterion ← false do
        if change on critical activity then
            Rank critical activities
            Crash according to MC slope
        else
            Rank non-critical activities
            Prolong according to MR slope
        end
        if remaining effort ≤ 0 || (no crashing & no prolonging) then
            stop criterion ← true
        else
            // Do nothing
        end
    end
end
```

3.4 Empirical study

As mentioned previously, participants are required to change the settings for every strategic component and for each decision moment. These components are then assembled into a solution strategy, which can be applied to real-life instances or simulated project networks. In this section, the results of students playing PSG Extended are reported. The project that was used is equal to the one used in the original PSG. First, the process of the game is described. Next, we discuss the performance of the solution strategies and reveal the influence of changes in the uncertainty, deadline and penalty value.

3.4.1 Game Process

PSG Extended commences with an introductory session explaining the various components used to build a solution (cf. section 3.2). Following this explanation, the partic-

Component	Possible values
Schedule Focus	
Schedule Focus	{1,DMS-(i-1)}
i = current decision moment, DMS=#decision moments	
Activity Focus	
Activity ranking	Table 3.2
Time/Cost focus	{0.25, 0.5, 0.75, 1}
Action Radius	
Deadline focus	{Low, Middle, High}
Slack consumption	{Low, Middle, High}
Cost/benefit analysis	{0.7, 0.9, 1, 1.1, 1.3}
Effort loading	{Low, Middle, High}

Table 3.3: Overview of the settings for every decision moment i throughout PSG Extended

ipants open a spreadsheet file listing the different decision moments and components. Default settings for every component and decision moment are set, leading to a cumulative effort that is well below the threshold. A screenshot of the spreadsheet file can be found in figure 3.2. The large number at the bottom represents the consumed effort percentage. Participants are not allowed to exceed the threshold of 100%. As mentioned before, the player only receives an update of the effort when one of the parameters is changed. Contrary to the original PSG, feedback on time and cost performance is excluded until the game has finished. Excluding time and cost performance feedback prompts participants to focus on a clever strategy, rather than trial-and-error scenarios. Participants are required to make a choice for each component throughout the different decision moments. As a setting for one of the components is changed, the amount of consumed effort, displayed as a percentage, is updated. An overview of the different components, as well as the different values for each component, is given in table 3.3. The value for every component can differ across decision moments. It is possible to assume a low degree of effort loading in decision moment 1 combined with a high amount of effort loading in decision moment 2. Consequently, the amount of solution strategies that can be constructed by the participants is astronomical. After the game, a feedback session takes place during which each participant or group of participants receives an individual report. The report outlines the settings chosen by the participants, as well as a comparison to the settings of the other groups. The time and cost performances are provided and the percentiles indicate how the group compares against the other competing groups. Additionally, a number of project characteristics are modified to assess

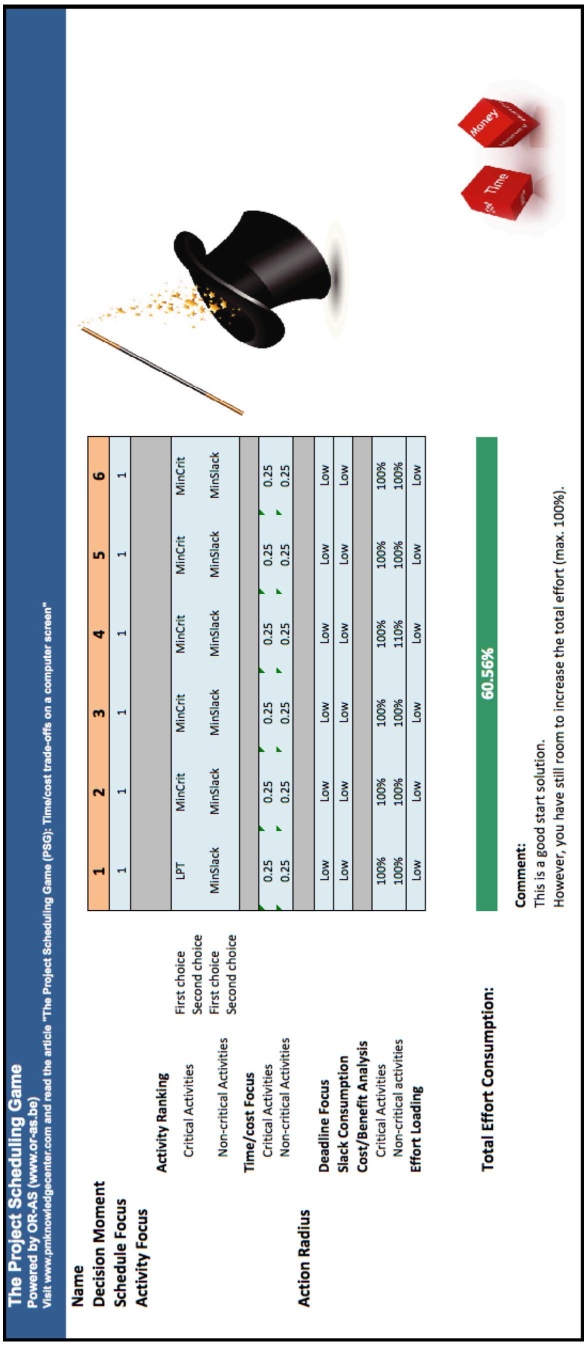


Figure 3.2: Overview of the spreadsheet file of PSG Extended

the influence on the solution quality:

- Impact of uncertainty: the delays of the original PSG are increased from their current values (100%) to 200% in steps of 20%.

- Project characteristics: the deadline and height of the penalty are varied to investigate the time and cost implications. An alteration in the deadline can change the ease with which the deadline can be reached while the penalty height regulates the severity of incurring a penalty.

In the following section, the time and cost performance of the participants will be analyzed. Both the project instance of the original PSG and the impact of uncertainty and different project characteristics will be scrutinized.

3.4.2 Empirical results

In this section, the performance of the student groups is presented. The settings of the students are assembled into a solution strategy which is applied to the project file of Vanhoucke et al. (2005). Labeled the "baseline scenario", some descriptive statistics of the participant population as well as the influence of the strategic components on the attained cost is provided in section 3.4.2.2. Next, some characteristics of the project of Vanhoucke et al. (2005) are modified to examine their effect on the overall solution quality.

3.4.2.1 Baseline scenario

The session of PSG Extended is scheduled after the original PSG has been played. As a result, participants are familiar with the characteristics of the project along with the relation between slack and project risk. The project characteristics of this section are identical to those presented by Vanhoucke et al. (2005). The project duration at the start of the project is equal to 123 days. However, the client stipulates a deadline of 107 days. For every day the project's duration exceeds the deadline, a penalty cost of €500 will be incurred. The participants are required to tune the settings of the strategic components of section 3.2 such that the total cost is minimized and the allowed effort does not exceed 100%.

Descriptive statistics PSG Extended was first rolled out at University College of London (UCL) in January 2014. 8 student participants worked on the same real-life instance as in the original PSG of Vanhoucke et al. (2005). In January 2015, a second

Group	#Students	Cost Deviation (%)			
		Min	Max	μ	σ
UCL (2014)	8	0.71	1.12	0.88	0.13
UCL (2015)	7	0.73	1.19	0.92	0.16
UGent	60	0.63	1.33	0.95	0.19

Table 3.4: Cost deviation for the participants of PSG Extended

student group at UCL participated in PSG Extended, counting 7 participants. 60 civil engineering participants make up the third part of the empirical data and hail from Ghent University (UGent). Some descriptive statistics can be found in table 3.4. This table displays descriptive statistics with regard to the cost deviation for the participants of PSG Extended. The problem is solved to optimality using the procedure of Demeulemeester et al. (1998). The 2014 group at UCL reached the smallest average cost deviation. The UGent group reached the lowest minimum cost deviation. However, the standard deviation, mean and maximum cost deviation are higher than those of the UCL student groups.

Effort-based components We also analyzed how the different strategic component settings affected the achieved project cost. The correlation between the schedule focus, time/cost focus, deadline focus, slack consumption, cost/benefit and effort loading elements was compared to the project cost. An overview is given in table 3.5, which displays the correlation between the various elements and the cost at the final decision moment. A negative correlation implies that the project cost decreases as the setting of the respective element is increased. Table 3.5 shows that schedule focus is the only component with a positive correlation (0.10) to the project cost. A higher degree of schedule focus leads, on average, to a higher project cost. For the real-life instance that was used, adopting a local schedule focus pays off compared to a global search point of view. Hence, it is better to change trade-off options on a limited set of activities rather than making fewer changes to a large set of activities. A higher time/cost focus, deadline focus, slack consumption, cost/benefit analysis and effort loading give rise to a decreased project cost.

3.4.2.2 Impact of parameter changes

In this section, it is shown how changes in the parameters affect the results obtained by the participants. In turn, the uncertainty, deadline and penalty height are modified. It

Strategic component	Element		Correlation
Schedule focus	Schedule Focus		0.10
Activity focus	Time/Cost focus	Crit	-0.31
		Non-crit	-0.17
Action radius	Deadline focus		-0.24
	Slack consumption		-0.21
	Cost/benefit	Crit	-0.08
		Non-crit	-0.05
	Effort loading		-0.13

Table 3.5: Correlation for the different dimensions compared to the final project cost

is worth noting that only one parameter is changed per experiment. Hence, this section does not take interaction effects into account.

Impact of uncertainty The aim of the PSG and PSG Extended is to discuss the careful balance between complexity and uncertainty. There is a need for optimization when dealing with complex problems, yet uncertainty may render this optimization exercise useless. As the project progresses, participants of the original PSG of Vanhoucke et al. (2005) are confronted with delayed activities or certain opportunities. In PSG Extended, those delays and opportunities were increased in a stepwise manner, going from the baseline scenario with an uncertainty equal to 100% to situations where uncertainty was increased to 200% in steps of 20%. The time results are shown in table 3.6. The table contains the project duration as a percentage of the duration when the uncertainty is equal to 1. More specifically, the percentage Project Duration ($D(\%)$) is calculated as follows:

$$D(\%) = \frac{D_{unc=u}}{D_{unc=1}}, u \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\} \tag{3.1}$$

The project duration for the different uncertainty values u is divided by the project duration for an uncertainty equal to 1. This allows us to assess the impact of an increase in uncertainty. The table indicates that as the uncertainty rises, the project duration increases as well. Since uncertainty is defined as a change in an activity's duration, it makes sense that this is, at least to some extent, propagated to the project level. We also examined the effect of uncertainty on the project costs. However, the increase in project cost is small compared to the rise in project duration, leading us to conclude that uncertainty resorts a stronger effect on time than on cost.

Group	Uncertainty (in %)					
	1.0	1.2	1.4	1.6	1.8	2.0
UCL (2014)	100%	101.35%	102.60%	102.91%	104.16%	105.93%
UCL (2015)	100%	101.43%	103.22%	102.74%	104.65%	106.44%
UGent	100%	101.34%	102.45%	102.56%	103.97%	105.57%

Table 3.6: Effect of uncertainty on project duration

Deadline impact The second characteristic that was changed is the deadline of the project. When the deadline's location shifts, the point at which a penalty is incurred changes as well. When the deadline differs greatly from the starting position (a project duration of 123 days in this case), reaching the deadline will become a difficult task. As such, a modification of the deadline parameter affects the ease with which the deadline can be reached. Altering the deadline was achieved as follows. The instance was solved according to the exact procedure of Demeulemeester et al. (1998). Let D_{min} and D_{max} denote the minimum and maximum project duration of the efficient time/cost frontier. The deadline, δ_n, is set using the parameter θ and is provided in equation (3.2).

$$\delta_n = D_{min} + \theta * (D_{max} - D_{min}) \tag{3.2}$$

For the deadline parameter, three values are used, namely $\theta \in \{0.25, 0.5, 0.75\}$. We also include the deadline of the original PSG, which corresponds with a value of $\theta = 0.29$. The cost results are calculated in a similar fashion to equation (3.1) and can be found in the upper half of table 3.7.

The three student groups display a similar behaviour. As θ increases, the project cost decreases. This trend can be explained as follows. The start duration of 123 days lies closest to the deadline if $\theta = 0.75$. Hence, the available effort can be spent on optimizing the activity trade-offs, rather than spending all the effort on crashing activities in order to approach the deadline more closely.
The conclusion is that as the starting position and deadline coincide, the project costs are lower compared to situations where there is a large discrepancy between the starting position and the deadline.

Penalty impact The final characteristic to be changed is the height of the penalty. Increasing the penalty discourages participants to exceed the imposed deadline. The penalty was increased from the original value of €500 to €1,000, €2,500 and €5,000.

Group	θ			
	0.25	0.29	0.5	0.75
UCL (2014)	100%	99.85%	99.30%	98.68%
UCL (2015)	100%	99.86%	99.35%	98.71%
UGent	100%	99.87%	99.35%	98.71%
Group	Penalty			
	500	1,000	2,500	5,000
UCL (2014)	100%	101.75%	106.74%	115.12%
UCL (2015)	100%	101.66%	106.58%	114.80%
UGent	100%	101.76%	107.04%	115.73%

Table 3.7: Effect of the deadline and penalty on project costs

The results are shown in table 3.7. As can be expected, a higher penalty value will pose more severe cost ramifications when the deadline is exceeded. As a result, the project cost rises, on average, as the height of the penalty is increased.

3.4.3 Summary

Participants of PSG Extended are required to input settings of the three strategic components. These are then combined into a solution strategy. In this section, the participant solutions were tested on the project of the PSG of Vanhoucke et al. (2005). It was found that a local schedule focus performs best and that an increase in settings for the other elements gives rise to a lower project cost. Next, the uncertainty, deadline and penalty height were altered. The main conclusions are that uncertainty leads to a longer project duration, that an increase in penalty leads to higher project costs and that better cost results are obtained when the starting position does not differ much from the deadline.

3.5 Computational results

Section 3.4 discussed participant results of PSG Extended. The parameter values for each of the strategic components were carefully set by the students, after which the components were assembled into a solution procedure as described in section 3.3. In this section, computational results are reported and analyzed. A diverse dataset will be employed in which the topological structure of the project networks, as well as different settings for uncertainty, deadlines and penalties will be utilized. The computational experiment allows us to assess whether the classroom results of section 3.4 can be generalized. Furthermore, we can take other aspects such as the topological structure of the project networks into account. Creation of the dataset as well as generation of the parameter settings will be discussed in section 3.5.1. Once these settings have been elucidated, results on the impact of the project characteristics on time and cost are divulged. Section 3.5.3 concludes the computational results by assessing how the parameter values affect the time and cost objective.

3.5.1 Test design

The test design contains details on the various settings that were used to construct a computer-generated dataset. The section is broken down into five paragraphs. First, we discuss the topological structure of the artificial project networks, as well as the time and cost values for the activity trade-offs. Next, we explore how the uncertainty was varied. Thirdly, project characteristics such as the deadline and penalty values are considered. Fourthly, details are provided on how the files containing the parameter values were generated. Finally, the manner in which the effort threshold was set is elaborated. PSG Extended, including the solution procedure of section 3.3, was implemented in C++. The experiments were conducted on Ghent University's High Performance Computing infrastructure. The Delcatty cluster, which possesses a quad-core Intel Xeon processor of 2.6 GHz and 64GB RAM, was used.

Topological & trade-off settings 10 projects with 30 activities were generated by means of the RanGen2 generation engine (Vanhoucke et al. (2008)) for each of the 9 values of the Serial/Parallel (SP) indicator, going from 0.1 to 0.9 with a 0.1 increment. While Vanhoucke et al. (2008) employed the term I_2 indicator, it was changed to its more intuitive name, the SP indicator, in later simulation studies (cf. Colin and Vanhoucke (2014), Wauters and Vanhoucke (2014b, 2015)). The goal of the SP indicator is to measure a network's degree of resemblance to a completely serial (high SP values) or parallel (low SP values) network. The number of trade-offs for each activity is drawn randomly between 1 and 10 options. For every possible trade-off option, a combination of durations and costs needs to be generated. The minimum activity costs are drawn randomly from the uniform distribution defined by the minimum of 500 cost units and the maximum of 2,500. The difference in cost between two adjacent trade-offs is not allowed to exceed 1,000 cost units. The minimum duration is drawn from 10 to 20 units of time with a maximum of 1 time unit between neighbouring trade-off options.

Uncertainty As mentioned previously, unexpected delays or opportunities distort the baseline plan. Uncertainty manifests itself in two ways, namely through the amount of activities subject to variation and the height of the variation. The minimum delay height is equal to 1, whereas the maximum delay height amounts to 9 time units. The proportion of activities subject to a delay is varied throughout our experiments, ranging from 0 to 0.75 in steps of 0.25. For the remainder of this chapter, the term uncertainty will be used to refer to the proportion of activities subject to a delay. If the uncertainty is equal to 0.25, 8 ($\approx 0.25 * 30$) activities will be delayed. 10% of the delayed activities will be ahead of time, indicating that an activity finished sooner than expected. The delay height for the activities that are ahead of time is drawn from the same distribution as the activities that are behind on time.

Project characteristics The PSG and the effort-based PSG are soft variants of the DTCTP-D. This implies that the deadline constraint is a soft one. Violating the deadline does not render the problem infeasible but surpassing the deadline is penalized by means of a penalty cost for every day the deadline is exceeded. Once trade-offs and, if applicable, delays are generated for each activity, a deadline is imposed. This is done in the same way as in section 3.4.2.2 and equation (3.2). For the deadline parameter, three values are used, namely $\theta \in \{0.25, 0.5, 0.75\}$. Four values for the penalty parameter are explored, namely a cost of 500, 1,000, 2,500 and 5,000 cost units per day. Finally, the selected trade-offs at the start of the project need to be set. In order to study the influence of the starting position in conjunction with the other parameters, three alternatives are proposed. In the first situation, every activity is set at its crash duration. In the second position, the duration corresponding with the middle trade-off is selected. Finally, the third starting position sets each activity at its longest duration.

Generating parameter files In section 3.4.2, the empirical results of PSG Extended were discussed. These results follow from a project network that is based on a real-life project and the settings that were fine-tuned by the participants of PSG Extended. For the computational experiment, the parameter settings are computer-generated and attempt to encompass a wide array of parameter values. The settings for the main components of PSG Extended, namely schedule focus, activity focus and action radius are provided in table 3.8. In that table, it is shown which values were used for the various components. In total, 419,904 (3*6*6*4*4*3*3*3*3*3) parameter files were generated, leading to a vast array of different settings, ranging from a low to a high effort consumption pattern. The results of the computational experiments are based on the generated parameter files and will allow us to draw conclusions concerning the impact of the various strategic components and their respective elements.

Schedule focus		
Schedule focus		{minimal, middle, maximal}
Activity focus		
Priority rule	Critical Non-critical	Table 3.2
Time/Cost focus	Critical	{0.25,0.5,0.75,1}
	Non-critical	{0.25,0.5,0.75,1}
Action radius		
Deadline focus		{low, medium, high}
Slack consumption		{low, medium, high}
Effort loading		{low, medium, high}
Cost/benefit analysis	Critical	{0.7,1.0,1.3}
	Non-critical	{0.7,1.0,1.3}

Table 3.8: Settings for parameter generation of the elements of the strategic components

Effort threshold As mentioned previously, participants of PSG Extended are constrained in their choices by the effort threshold, which is not allowed to exceed 100%. For every project instance, the effort threshold can be chosen from a range defined by a minimum and maximum value. In order to set the effort threshold, each project instance is solved twice:

- Unconstrained run: no effort threshold is imposed. The solution procedure of algorithm 1 is followed but now includes a modified stopping criterion. The solution procedure only advances to the next decision moment if no critical activity can be crashed and no non-critical activity can be prolonged. Each of the 419,904 generated parameter files is applied to the project instance, resulting in a vast array of values for the *consumed* effort. The files are sorted in increasing effort consumption order. The effort limitation for each project is chosen as a percentile of the sorted values. The effort threshold corresponding with the x^{th} ($x \in \{10, 40, 70\}$) percentile is employed for the second run.

- Effort-constrained run: each of the parameter files is applied to the project instance. The effort threshold constrains the trade-off changes that can be made. When the remaining effort is equal to 0, the solution procedure proceeds to the following decision moment, as described in section 3.3 and algorithm 1.

A summary of the settings of the paragraphs described above is provided in table 3.9.

Description	Settings
SP factor	0.1-0.9, $\Delta = 0.1$
#Projects	10
Activity Costs	$\sim U(500, 2500)$
Activity Duration	$\sim U(10, 20)$
Uncertainty	0-0.75, $\Delta = 0.25$
θ	0.25-0.75, $\Delta = 0.25$
Penalty	{500,1000,2500,5000}
Starting position	Crash, middle, longest duration
#preference files	419,904
Effort threshold	{10,40,70}

Table 3.9: Settings for the computational experiment

3.5.2 Main Experiment

In this section, the main results of the computational experiment are discussed. Unless mentioned otherwise, a number of parameters were fixed in order not to mix different underlying effects. As a result, the 40^{th} effort percentile, a value for the SP factor of 0.5 and an uncertainty level of 0.5 are used to report on the influence of the other parameter values. The effect of the effort percentile, deadline, penalty, uncertainty and SP indicator are discussed in this section. Throughout these sections, we will comment on the influence of the starting position.

The impact of the effort percentile, deadline, penalty, uncertainty and SP indicator is measured using the same performance metric. Suppose our aim is to measure the influence of a project characteristic on an arbitrary objective O, where O could signify the average project cost, duration or effort. Since we want to determine the influence of a given project characteristic, it is necessary to compare the objective O across different values of the project characteristic, indexed by i. Therefore, the percentage increase or decrease in O is calculated. This is shown in table 3.10. Table 3.9 listed the different values for every project characteristic. As an example, the deadline was varied using parameter θ and could assume three values, namely 0.25, 0.5 and 0.75. If the impact of the deadline on a given objective is to be studied, the percentage increase or decrease, symbolized by $O(\%)$ in table 3.10, is calculated by dividing O for each value $\theta = i$ ($i \in \{0.25, 0.5, 0.75\}$) by $O_{\theta=0.25}$. Hence, $O(\%)$ is found by dividing by a baseline value, found in the denominator in table 3.10.

	Effort	Deadline	Penalty	Uncertainty	SP indicator
$O(\%)$	$\frac{O_{eff=i}}{O_{eff=10}}$	$\frac{O_{\theta=i}}{O_{\theta=0.25}}$	$\frac{O_{p=i}}{O_{p=500}}$	$\frac{O_{unc=i}}{O_{unc=0}}$	$\frac{O_{sp=i}}{O_{sp=0.1}}$
$i \in$	$\{10, 40, 70\}$	$\{0.25, 0.5, 0.75\}$	$\{500, 1000, 2500, 5000\}$	$\{0, 0.25, 0.5, 0.75\}$	$\{0.1, ..., 0.9\}$

Table 3.10: Calculation of the performance metric for an objective O

3.5.2.1 Effect of the effort percentile

The major constraint for participants of PSG Extended lies in the imposed effort threshold. Activity trade-offs can be changed until the remaining effort is equal to 0. This was already detailed in section 3.3 and algorithm 1. In this section, we vary the effort threshold. Consequently, as the effort threshold is raised, there is more room to make changes to activity trade-offs. Changing the effort percentile was found to have an impact on the project costs. As explained in table 3.10, the project costs for the 10^{th}, 40^{th} and 70^{th} percentile will be divided by the project costs of the 10^{th} percentile. Hence, a value smaller than 100% implies that as there is more room to execute trade-off changes, the quality of the attained solutions rises. On average, the project cost at the 40^{th} percentile was equal to 93.12% of the project cost at the 10^{th} percentile, whereas for the 70^{th} percentile a project cost of 88.36% was reached.
Raising the effort threshold and allowing more activity trade-off changes to take place leads, on average, to lower project costs. This implies that, as more trade-off changes are allowed (for instance through the availability of a higher contingency budget), the overall project costs will decrease.

3.5.2.2 Effect of the deadline

The deadline stipulates the point from which a penalty will be incurred. This section aims to assess the impact of a change in the deadline's location and how it interacts with the project's starting position.
The project cost for the alternative values of θ is studied in relation to the average project cost of $\theta = 0.25$ and the outcome is presented in table 3.11. Table 3.11 reveals that as the discrepancy between θ and the starting position increases, the average project cost increases as well. The minimum cost for each of the starting positions is indicated in bold. As the starting position coincides with the deadline, more effort can be allocated to trying to optimize the trade-off selections without having to alter the project duration in a drastic manner.
The location of the deadline, along with the starting position, resorts an important effect

θ	Starting position		
	Crash	Middle	Longest
0.25	**100%**	100%	100%
0.5	100.22%	**79.32%**	79.51%
0.75	101.00%	79.67%	**62.17%**

Table 3.11: Effect of the deadline in relation to the starting position

Starting position	θ	Penalty			
		500	1000	2500	5000
Middle	0.25	100%	105.50%	120.67%	145.46%
Longest	0.25	100%	120.14%	169.69%	252.75%
	0.5	100%	115.34%	152.37%	214.74%
	0.75	100%	110.20%	136.98%	180.87%

Table 3.12: Effect of the penalty on project costs

on the project costs. The lowest costs are achieved when the location of the deadline and the starting position are more aligned. It is worth noting that this conclusion is identical to the deadline impact of the empirical study of section 3.4.

3.5.2.3 Effect of the penalty

The previous section varied the location of the deadline by means of a parameter θ. A second element that determines the attractiveness of reaching the deadline is the penalty cost. In this section we examine the impact of the penalty height on the cost objective. The penalty cost assumes a value of 500, 1,000, 2,500 and 5,000. In our analysis, we limit ourselves to the situations with a penalty cost > 0. The results are provided in table 3.12. As the discrepancy between the starting position and θ increases, the deviation from the deadline increases and hence, the penalty cost becomes larger. This is apparent for the position where the longest activity duration is selected at the start of the project. The cost increase as the penalty becomes larger is higher for $\theta = 0.25$ than for $\theta = 0.5$. Increasing the penalty height discourages exceeding the deadline. The penalty has an impact on the cost objective and is closely related to the deadline and starting position. As the difference between starting position and deadline increases, the penalty share in the total project cost rises as well. This finding is similar to the observation of the empirical study.

	Uncertainty			
	0	0.25	0.5	0.75
Project Duration	100%	107.68%	115.33%	122.68%

Table 3.13: Effect of uncertainty on the project duration

3.5.2.4 Effect of uncertainty

In section 3.5.1, uncertainty was defined to refer to the proportion of activities subject to a delay. A higher value for the uncertainty corresponds with an increase in the number of activities that are ahead of or behind schedule. The effect of uncertainty was not found to differ along the starting position. Hence, we only report on the effect of uncertainty on the project duration, averaged across all starting positions. The empirical study of section 3.4 found that uncertainty influences the project duration rather than the project costs. This observation is corroborated in our computational tests. Table 3.13 shows that as the uncertainty increases, the average project duration increases almost linearly. The effect on the resulting costs is less clear. The explanation lies in the deviation from the deadline. While the uncertainty increases the average project duration, it does not necessarily push the project duration beyond the deadline. As a result, the rise in uncertainty does not always lead to higher penalty costs. In the few situations where a higher degree of uncertainty leads to a positive deadline deviation, the conclusions on the effect of the penalty of section 3.5.2.3 come into play. While the location of the deadline and the penalty exhibit an effect on the cost objective, uncertainty affects the time objective. A higher degree of variation mainly leads to an increase in project duration.

3.5.2.5 Effect of the SP indicator

One of the computational experiment's main advantages lies in the generation of different settings and project networks. As a result, the impact of the topological structure of the networks can be investigated.

We discovered that the topological structure affects the consumed effort. While the penalty and deadline parameters do not produce a significant effect, the starting position influences the effort as well. This finding is hardly surprising. The starting position determines whether only non-critical activities (shortest duration), critical activities (longest duration) or both (middle duration) can be changed at the start of the project. Moreover, the topological structure regulates the amount of (non-)critical activities. As the project network becomes more serial, the number of critical activities rises. As a

Starting position	SP indicator								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Shortest	100%	89.52%	85.33%	84.26%	78.94%	61.57%	53.22%	40.30%	24.72%
Middle	100%	88.84%	85.23%	85.35%	81.56%	71.21%	67.71%	69.01%	60.82%
Longest	100%	106.74%	118.01%	138.43%	144.50%	135.41%	133.96%	135.42%	127.19%

Table 3.14: Effect of the SP indicator on the consumed effort

result, there should be an effort difference between the crash and longest duration along the SP indicator.

The results are summarized in table 3.14. When every activity is set to its shortest duration, the consumed effort greatly decreases as the project becomes more serial. At the start of the project, the only possibility to change trade-offs consists of increasing the duration of non-critical activities. Since serial project networks count few non-critical activities, the effort is higher for networks with a lower SP indicator. It is expected that as the starting position changes, this effect will become less pronounced. This observation is valid as the effort proportion for the middle and longest starting position is equal to 60.82% and 127.19% for a value of the SP indicator of 0.9 respectively. These percentages are considerably higher compared to the shortest starting position (24.72%).

3.5.3 Strategic component analysis

In section 3.2, the different strategic components that need to be carefully tuned by the participants were discussed. This section discusses how the settings of the different strategic components influence the attained solution quality. The outline of this section is identical to that of section 3.2, where the strategic components of the effort-based Project Scheduling Game were elucidated.

3.5.3.1 Schedule Focus

Three different combinations for the *schedule focus* were tested, resulting in a low, medium and high degree of schedule focus. These concepts are closely related to a local versus a global search. A high schedule focus selects all activities while a small value only looks at the activities of the current decision moment. The results are depicted in figure 3.3. The x-axis displays the average schedule focus across all decision moments, while the y-axis represents the project costs. A local schedule focus attains the best results, except when every activity is set at its crashing duration. In that situation, a medium schedule focus is slightly better than the local focus. However, figure

3.3 clearly shows that a global search leads to increased project costs, regardless of the starting position.

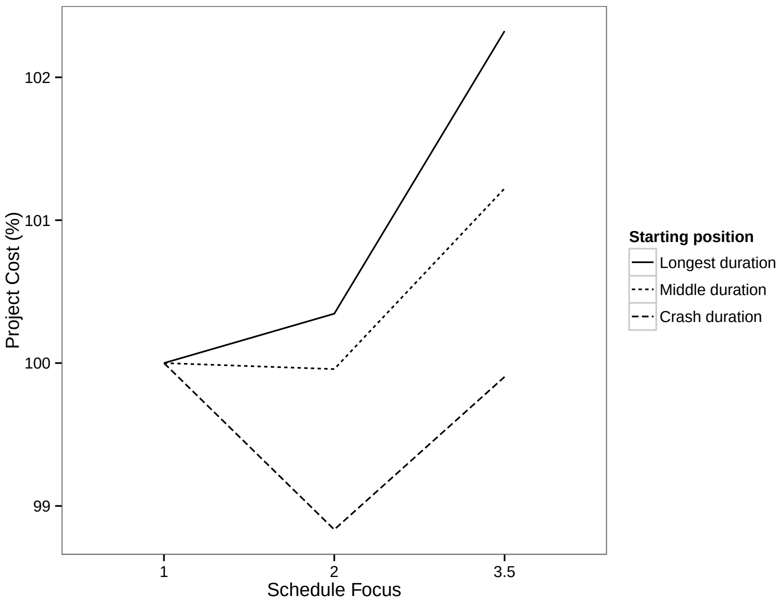


Figure 3.3: Relation between the schedule focus and project costs

3.5.3.2 Activity Focus

The order in which changes are made to activities is governed by the *ranking* phase. The impact of priority rules can be related to the starting position of the projects. If the activities are set to their lowest duration, there is no potential for crashing. As a result, it matters little which priority rule is selected. The same observation can be made for non-critical activities when every activity is set to its maximum duration. Since the activities cannot be prolonged, the impact of the priority rule is very low. Disregarding these situations, the best priority rules for critical activities are the Most Immediate Successors and Longest Processing Time rules. Crashing an activity with a lot of successors directly affects a large number of activities while activities with a long processing time may provide great crashing potential. The best performing priority rule for non-critical activities is the Maximum Slack rule. Activities with a lot of slack can be prolonged without the risk of becoming critical. Hence, extending the duration leads to savings without incurring a penalty cost.

Starting Position	(Non-)critical	Time/cost focus 0.25	0.5	0.75	1
Average	Critical	100%	99.77%	99.42%	99.03%
	Non-critical	100%	99.62%	99.11%	98.78%
Crash	Critical	100%	100.25%	100.60%	100.79%
	Non-critical	100%	99.31%	98.45%	97.72%
Middle	Critical	100%	101.03%	102.13%	102.10%
	Non-critical	100%	99.47%	98.71%	98.42%
Longest	Critical	100%	98.03%	95.53%	94.20%
	Non-critical	100%	100.07%	100.17%	100.21%

Table 3.15: Effect of the time/cost focus on the project costs

While the schedule focus operates on a project level, the *time/cost focus* drills down to the activity level. The time/cost focus regulates the number of trade-off options that are compared. For critical and non-critical activities, four combinations were tested. The effect on the project cost can be found in table 3.15. On average, a higher degree of time/cost focus for both critical and non-critical activities leads to cost improvements, which can be found in the rows labeled “Average”. These results are averaged across all penalty, deadline and starting position settings. However, closer inspection reveals that the cost improvement for critical activities is solely due to the longest duration starting position. When every activity is set to the trade-off with the longest duration, it will be necessary to crash a lot of activities in order to approach the deadline more closely. A higher degree of time/cost focus allows for either a larger reduction in duration, a better cost solution or both. This explains why a higher degree of time/cost focus for critical activities is mainly relevant for the longest duration starting position. The alternative starting positions report cost improvements for a high time/cost focus for non-critical activities and small deteriorations for critical activities.

3.5.3.3 Action Radius

Deadline focus, slack consumption, cost/benefit analysis and effort loading comprise action radius. *Deadline focus* and *slack consumption* govern the change to a critical or non-critical activity respectively. Adopting a high degree of deadline focus will, on average, result in more changes to critical activities while the opposite observation holds true for slack consumption. Consequently, we hypothesize that different settings of slack consumption and deadline focus will yield a different deviation from the deadline. This is shown in figure 3.4. On the x-axis the different values for slack consumption and deadline focus are shown, while the deadline deviation is found on the y-axis. The graph indicates that a higher value for the slack consumption (non-critical activities) leads to a higher deadline deviation. Deadline focus follows the opposite trend. The explanation for this phenomenon is straightforward. As the deadline focus increases, more changes are made to critical activities, whose duration will be reduced.

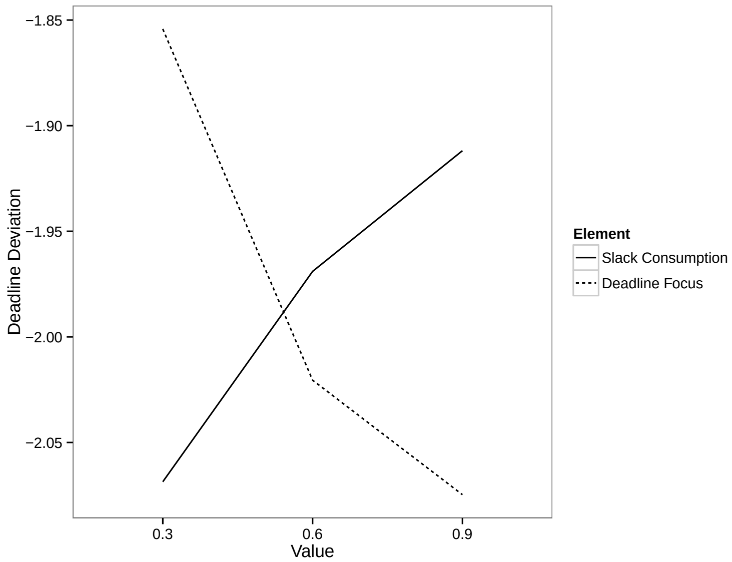


Figure 3.4: Relation between the slack consumption, deadline focus and deadline deviation

The degree to which cost improvements or deteriorations are allowed is controlled by the *cost/benefit* parameter. Before proceeding to the analysis of the impact of this parameter, it is vital to comment on situations where the parameter has little to no impact. We discern two different kinds of circumstances in which changing the parameter is of little use. First of all, when the project duration is much smaller than the deadline, the effect of the cost/benefit parameter is low. Since there are no penalty costs (the deadline is not exceeded), changes to critical activities only lead to increased costs, while prolonging non-critical activities saves money. Secondly, the investments or savings are compared to the penalty costs. For very large penalty values, the outcome will always be the same. The explanation follows from the settings of the computational experiment. The difference in cost between adjacent trade-offs does not exceed 1,000 cost units. However, this number is relatively low compared to the highest penalty settings of table 3.9. These two reasons explain why the focus for our analysis was limited to situations with a penalty value of 500. The deadline was chosen according to the starting position: θ=0.25 for the shortest, 0.5 for the middle and 0.75 for the longest starting position. Overall, a value of 1.3 for critical activities yielded the best results. Changing the cost/benefit parameter for non-critical activities had a much smaller effect compared to critical activities. Compared to critical activities, prolonging a non-critical activity will rarely result in a change of the critical path. Consequently, cost savings are made but there are no extra costs implying that the ratio of savings and cost exceeds the cost/benefit parameter. In other penalty and deadline situations, using a cost/benefit parameter value of 1 is advised.

The final element of action radius is *effort loading*. The allowed effort is divided across all decision moments but can be altered by means of the effort loading parameter. Raising the threshold for a decision moment entails that the threshold for a different decision moment will be lowered to ensure that the threshold across all decision moments is still equal to 100%. Three different settings for the effort threshold were utilized, in which the effort was front-loaded, spread evenly or back-loaded. Front-loading corresponds with a project in which the bulk of the effort will be expended in the early stages of the project. Our results indicated that the front-loading setting is beneficial in most situations. There is only one exception, namely when the activities are set to their longest duration and $\theta = 0.75$. In that case, the best cost results are found when the effort is back-loaded.

3.6 Conclusion

The goal of this chapter was twofold. First of all, we have proposed a novel way of constructing solutions for the Discrete Time/Cost Trade-off Problem. A crucial element in building these solutions is a limitation on the available effort. The constrained effort forces participants of PSG Extended, an effort-based extension to the PSG of Vanhoucke et al. (2005), to make a choice with regard to the components of the solution. The solution components, coined strategic components, are schedule focus, activity focus and action radius. For each element of the strategic components and for each decision moment, participants are required to carefully tune the elements' parameters. Once this is done, the components are combined into a solution approach, which is then applied to a project network with certain characteristics such as a deadline, penalty and degree of uncertainty.

As mentioned, the modular manner of creating solutions for the DTCTP has been studied in light of a limited amount of effort. Its link with real life is evident since project managers are always confronted with a limited amount of effort (manpower, cash). PSG Extended was rolled out at University College London and Ghent University. Participation data enabled us to perform a first analysis of the empirical results, which revealed the negative impact of the penalty height and the important influence of the deadline on the cost objective. Uncertainty was found to exhibit a negative influence on the project duration, rather than the project costs. The empirical data enabled us to analyze the settings of the elements of the strategic components. Schedule focus proved to be of particular interest, since a more global schedule focus hampered the overall project costs.

Apart from an empirical perspective, this chapter offered a computational perspective as well. We constructed a dataset, characterized by a large variety of settings for the penalty, deadline, uncertainty and topological structure of the projects. Additionally, the starting position was varied as well. A carefully controlled process of setting the parameters for the elements of the strategic components resulted in 419,904 files that were applied to each project instance.
First, we established the influence of DTCTP characteristics on the solution quality, regardless of the settings of the individual solutions. Interestingly, the characteristics have an impact on different dimensions of the solution quality, as depicted in figure 3.5.

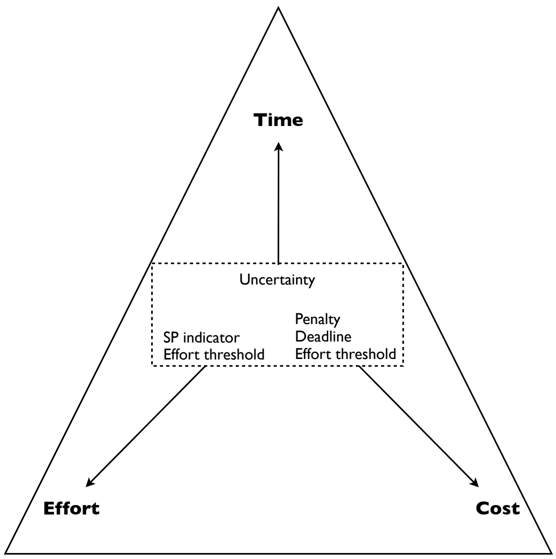


Figure 3.5: Effect of DTCTP characteristics on solution quality

A higher effort percentile is associated with a lower project cost, while a higher penalty value affects the project costs negatively. The minimum cost solution for different values of the deadline interacts with the starting position. As the discrepancy between the starting position and deadline rises, so do the costs. The impact of an increase in uncertainty is mainly felt in the attained project duration, while the topological structure (measured by the SP indicator) resorts an effect on the expended effort.
Secondly, it is possible to make a number of recommendations with regard to the solution settings. It is advised to adopt a low degree of schedule focus, studying only the activities of the current decision moment. The best performing priority rules to rank the activities are the MIS and LPT rules for critical activities and the MaxSlack rule for non-critical activities. The time/cost focus for critical activities is best kept limited, unless a lot of activity crashing needs to take place. If reaching the deadline is an important objective, the deadline focus should be increased since it leads to a lower deadline deviation. While an increased slack consumption leads, on average, to longer projects, the associated risk will rise as well. Since more changes to non-critical activities occur, the amount of slack will be reduced, leading to a smaller buffer against uncertainty. In general, a cost/benefit parameter value of 1 is the most rational choice, with one notable exception. If the project duration and the deadline are close to one another, changes to critical activities are best made if the savings greatly compensate the investment (a value of 1.3). Effort loading applies a correction to a decision moment's allowed threshold. A front-loaded, back-loaded and even effort loading consumption pattern were compared. In general, the front-loaded effort consumption yielded the best cost results.

It is our hope that future research efforts will allow us to expand on the empirical results, both in an academic and practical setting. When more real-life data is accumulated, it will be possible to map the real-life projects to the generated dataset and to corroborate or contradict the findings of the computational section of this chapter. Furthermore, additional priority rules can be tested and their impact can be compared with the priority rules of table 3.2. The adaptation of existing heuristics to include an effort restriction and reporting on the deviation from the optimal solution would constitute a valuable follow-up study to this chapter.

Part II

Forecasting

4

A study of the stability of Earned Value Management forecasting

In this chapter, we focus on the stability of Earned Value Management (EVM) forecasting methods. The contribution is threefold. First of all, a new criterion to measure stability that does not suffer from the disadvantages of the historically employed concept is proposed. Secondly, the stability of time and cost forecasting methods is compared and contrasted by means of a computational experiment on a topologically diverse data set. Throughout these experiments, the forecasting accuracy is reported as well, facilitating a trade-off between accuracy and stability. Finally, it is shown that the novel stability metric can be used in practical environments using two real-life projects. The conclusions of this empirical validation are found to be largely in line with the computational results.

4.1 Introduction

Earned Value Management (EVM) is a project control methodology that allows project managers to assess the status of their project in terms of time and cost. EVM originated in the 1960s when the American Department of Defense (DoD) was looking to standardize its processes for the appraisal of project performance. The methodology falls back on three key metrics to evaluate the status of a project. Several indicators use a combination of the Planned Value (PV), Earned Value (EV) and Actual Cost (AC) to present a project manager with a quantitative indication of the health of the project under study. Hence, these indicators provide early warning signals that may trigger either corrective actions or the exploitation of project opportunities.

In order to overcome the unreliable behaviour of the time performance indicators, Lipke (2003) proposed an alternative, known as the Earned Schedule (ES) concept. Incidentally, the time performance indicators were modified to reflect the implementation of a time dimension. ES was included with EVM in standard texts, such as Fleming and Koppelman (2005) and Vanhoucke (2010a). Considerable attention was given to the prediction of the final cost and final duration of a project, where the emphasis was placed on accuracy, i.e. predicting the real duration or real cost as correctly as possible. Several empirical and simulation-based studies were conducted, as found in Zwikael et al. (2000), Christensen and Templin (2002) and overview articles of Christensen (1993) and Christensen et al. (1995). With the advent of Earned Schedule, comparative studies were carried out to determine whether the Planned Value (PV), Earned Duration (ED) or Earned Schedule (ES) performed best and which Performance Factor (PF) yielded the best performance. A comparison can be found in Anbari (2003), Rujirayanyong (2009), Vandevoorde and Vanhoucke (2006) and Vanhoucke and Vandevoorde (2007).

In recent years, research on project control has increasingly focused on the integration of three aspects of the project lifecycle, namely baseline scheduling, risk analysis and project control. This is known as dynamic scheduling. The reader is referred to Vanhoucke (2013) for an extensive overview of the aspects and the interrelation of the dynamic scheduling dimensions. A dynamic scheduling study was undertaken by Vanhoucke (2011), in which a direct comparison between a bottom-up project tracking approach (Schedule Risk Analysis (SRA)) and a top-down project tracking approach (EVM) is made. It was found that top-down approaches work well for serial projects whereas bottom-up approaches are at an advantage for parallel project networks. Elshaer

(2013) followed the argumentation of Vanhoucke (2012a) who proposed to integrate sensitivity information with EVM data. The time forecasting problem was tackled by embedding activity sensitivity measures in the Earned Schedule forecasting methods. The results showed that including sensitivity information reduced the error caused by false warning signals originating from non-critical activities.

The literature on project control forecasting has largely been inspired by accuracy. Consequently, the body of literature dedicated to the stability of forecasting methods is limited compared to accuracy studies. Nevertheless, the importance of stability cannot be underestimated. In supply chains, increased variability is passed on to the following stage, leading to increasing distortions because of the bullwhip effect (Lee et al. (1997)). Schnaars (1984) notes that one of the factors affecting forecast accuracy lies in the lack of stability of the underlying time series. This statement is valid for project control situations since a lack of stability in the Cost Performance Indicator (CPI) will be reflected in the forecasting methods that make use of this performance metric. Christensen and Payne (1992) note that a stable Cost Performance Index (CPI) is evidence that the contractor's systems are working correctly. On top of this, a stable CPI value can be combined with the To Complete Performance Index to inspire confidence in declaring the contractor's performance out of control. When large fluctuations of any of the EVM metrics arise, it leads to growing suspicions concerning the contractor's ability to manage the project at hand. Therefore, it is more difficult to judge whether warning signals are true or false. A stable CPI on the other hand shows that variances are being identified and corrected in a timely fashion (Christensen and Heise (1993)).
Of the few contributions on stability, most are concerned with the stability of an index rather than the consequences they exhibit on forecasting accuracy and stability. The first contribution in this respect was provided by Payne (1990) who defined a stable CPI as one that does not vary more than 10% once a project has surpassed the 50% completion point. A number of years later, this definition was relaxed in a paper by Christensen and Heise (1993), where a stable CPI was defined as not being subject to a change of more than 10% from the 20% completion point onwards. The accuracy and stability of the final project cost was assessed by Zwikael et al. (2000). The authors used a regression analysis to come to the conclusion that the accuracy improves along the percentage complete and utilize visual inspection of the forecasting errors to determine the point of stability. Christensen and Templin (2002) tested the stability of the cumulative CPI on a number of projects and were able to confirm the stability of this indicator using the same rule-of-thumb for stability judgement. Finally, project performance stability

was also tested by Henderson and Zwikael (2008), where the CPI and SPI(t) indicators were subjected to the widely reported CPI stability rule. It is interesting to note that while the behaviour of the SPI(t) was found to be highly consistent with the CPI, the stability rule could not be confirmed.

While this chapter focuses on assessing the accuracy and stability of deterministic forecasting methods, several methods have been proposed that rely on stochastic analysis or forecasting. The reader is referred to Barraza et al. (2000, 2004) (stochastic S-curves), Nassar et al. (2005) (Weibull distributions) and Kim and Reinschmidt (2009, 2010) (Kalman filter and Bayesian statistics) for a number of applications of stochastic forecasting.

The motivation for this research stems from the paper by Henderson and Zwikael (2008) who showed that the CPI stability rule cannot be generalized, not even for the Department of Defense portfolio. On top of these findings, it is worth noting that the CPI stability depends on arbitrary and subjective thresholds, namely the percentage complete and the allowed deviation. Project managers in dissimilar industries may utilize different thresholds and thus have a different judgement of stability.
The contribution of this chapter is threefold. First and foremost, a new criterion for measuring stability which does not depend on subjective thresholds and can be used across industries is proposed. It is worth mentioning that this criterion yields a *degree* of stability rather than a binary outcome. Secondly, the stability of time and cost forecasting methods is assessed on a topologically rich data set that results from simulations according to predefined statistical distributions. This is of particular relevance for the cost forecasting methods for which research has been largely dominated by empirical evidence. Accuracy results are reported as well, permitting a trade-off between accuracy and stability of the reported forecasting methods. Thirdly, the findings of the computational study are validated using real-life data from two consultancy projects, executed for one of Europe's leading logistics groups.

The outline of this chapter is as follows. In section 4.2, the methodology of this chapter is broken down into three parts. Network generation, Monte Carlo simulation and solution quality metrics are the three methodological components. Section 4.3 details the settings of the computational experiment using the structure of the previous section. The following section, section 4.4, provides the results of the experiment. The results are divided into two main parts. Section 4.4.1 discusses general observations of the accuracy and stability of the forecasting methods, as well as the impact of the topological structure and the percentage complete. Section 4.4.2 analyzes the consequences of a change in coefficient of variation due to either a change in the mode or the mean of the underlying distribution. The findings of the computational experiment are supplemented by an empirical study, which was conducted on two warehousing projects at a large logistics company. This can be found in section 4.5. Finally, conclusions and a future outlook are provided.

4.2 Methodology

Network Generation In the first phase of the methodology, a dataset is constructed that consists of projects with a varying topological structure. Even though a number of different topological indicators exist, we opted for the Serial/Parallel (SP) indicator. Originally, the paper of Vanhoucke et al. (2008) used the term I_2 indicator but later project management simulation studies (Vanhoucke and Vandevoorde (2007, 2008, 2009)) adopted a more intuitive term, the SP indicator. This indicator measures a network's resemblance to a completely serial or parallel project and assigns the network a value in the interval [0,1]. It is based on the progressive level concept proposed by Elmaghraby (1977). The SP indicator determines the maximum number of levels, which corresponds to the longest chain of serial activities in the network. Let m be the maximal progressive level and n the number of non-dummy activities in the project network. Then the SP indicator can be defined as $SP = \frac{m-1}{n-1}$ for projects with more than one activity. For a completely serial network, SP = 0, whereas a completely parallel network corresponds with SP = 1. In practice, most projects assume an intermediate value. The rationale for preferring the SP indicator stems from the observation of Vanhoucke and Vandevoorde (2007), who established a clear link between the SP indicator and the forecasting accuracy. The authors came to the conclusion that the performance of the EVM forecasting methods for time performed best for more serial project networks, which was corroborated in a research contribution by Vanhoucke (2012a).

Monte Carlo simulation In order to introduce uncertainty, Monte Carlo simulations that impose variability on the durations of the activities are used. The random variation is generated according to the generalized beta distribution, which has been employed in previous project management studies (e.g. Vanhoucke (2010b) and Wauters and Vanhoucke (2014b)), as well as in simulation studies for construction projects (AbouRizk et al. (1994)). Following its straightforward interpretation or as an approximation to the beta distribution, the triangular distribution is often preferred (Johnson (1997)). Despite its widespread use, the caution of Kuhl et al. (2007) who advise against the use of the triangular distribution if empirical datapoints are absent was followed. Consequently, the computational nature of the experiments makes us more inclined to the generalized beta distribution. The generalized beta distribution is a continuous probability distribution that depends on 4 parameters, namely a lower limit a, an upper limit b and shape parameters θ_1 and θ_2. Its probability density function is given as follows, where $\Gamma(\cdot)$ refers to the gamma function:

$$f(x) = \frac{\Gamma(\theta_1 + \theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)(b-a)^{\theta_1+\theta_2-1}}(x-a)^{\theta_1-1}(b-x)^{\theta_2-1}, \quad x \in [a,b] \tag{4.1}$$

In general, the mean, μ, and mode, m, of the generalized beta distribution are given by equations (4.2) and (4.3).

$$\mu = a + (b-a)\frac{\theta_1}{\theta_1 + \theta_2} \tag{4.2}$$

$$m = a + (b-a)\frac{\theta_1 - 1}{\theta_1 + \theta_2 - 2} \tag{4.3}$$

In this chapter, we choose to set the values for μ and m and subsequently derive the shape parameters θ_1 and θ_2. For given values of μ and m, the shape parameters are given by equations (4.4) and (4.5).

$$\theta_1 = -\frac{(b+a-2m)(a-\mu)}{(m-\mu)(a-b)} \tag{4.4}$$

$$\theta_2 = \frac{(b+a-2m)(b-\mu)}{(m-\mu)(a-b)} \tag{4.5}$$

Solution quality metrics Once the simulations have been executed, it is possible to make a prediction for the various time periods and for all simulation runs. At this point in time, the solution quality needs to be measured which is done using a metric for the accuracy and one for the stability of the forecasting method under study. The accuracy is measured using the Mean Absolute Percentage Error (MAPE) as is frequently done in forecasting studies (see e.g. Vanhoucke (2010a)) and provides an indication of how well a method predicts the final duration (equation (4.6)) or cost (equation (4.7)) of a project. EAC(t) denotes the Estimate At Completion for time, whereas EAC specifies the cost Estimate At Completion. rp indexes the time period at which a prediction is made and ranges from 1 to R, the total number of time periods.

Key indicators			
PV	Planned Value	SPI	$\frac{EV}{PV}$
AC	Actual Cost	SPI(t)	$\frac{ES}{AT}$
BAC	Budget At Completion	CPI	$\frac{EV}{AC}$
EV	$PC * BAC$	ES	$t + \frac{EV-PV_t}{PV_{t+1}-PV_t}$
Time forecasting		*Cost forecasting*	
$EAC(t)_{PV_1}$	$PD - \frac{(EV-PV)*PD}{BAC}$	EAC_1	$AC + (BAC - EV)$
$EAC(t)_{PV_{SPI}}$	$\frac{PD}{SPI}$	EAC_2	$AC + \frac{BAC-EV}{CPI}$
$EAC(t)_{PV_{SCI}}$	$\frac{PD}{CPI*SPI}$	EAC_3	$AC + \frac{BAC-EV}{SPI}$
$EAC(t)_{ED_1}$	$PD + AD * (1 - SPI)$	EAC_4	$AC + \frac{BAC-EV}{SPI(t)}$
$EAC(t)_{ED_{SPI}}$	$\frac{PD}{SPI}$	EAC_5	$AC + \frac{BAC-EV}{SCI}$
$EAC(t)_{ED_{SCI}}$	$\frac{PD}{SPI*CPI} + AD * (1 - \frac{1}{CPI})$	EAC_6	$AC + \frac{BAC-EV}{SCI(t)}$
$EAC(t)_{ES_1}$	$AD + PD - ES$	EAC_7	$AC + \frac{BAC-EV}{0.8CPI+0.2SPI}$
$EAC(t)_{ES_{SPI(t)}}$	$AD + \frac{PD-ES}{SPI(t)}$	EAC_8	$AC + \frac{BAC-EV}{0.8CPI+0.2SPI(t)}$
$EAC(t)_{ES_{SCI(t)}}$	$\frac{PD-ES}{CPI*SPI(t)}$		
$EAC(t)_{ES_{2\alpha}}$	$AD + \frac{PD-ES'}{SPI(t)'}$		

Table 4.1: Summary table of EVM terminology and formulas

$$MAPE_t = \frac{1}{R}\sum_{rp=1}^{R}\frac{|RD - EAC(t)_{rp}|}{RD} \tag{4.6}$$

$$MAPE_c = \frac{1}{R}\sum_{rp=1}^{R}\frac{|RC - EAC_{rp}|}{RC} \tag{4.7}$$

The stability is measured using the Mean Lags, which is similar in interpretation to the MAPE but gauges the deviation between subsequent values regardless of the accuracy of the forecasting method. The Mean Lags metric can be defined as follows for the time dimension:

$$Mean\ Lags = \frac{1}{R-1}\sum_{rp=2}^{R}\frac{|EAC(t)_{rp-1} - EAC(t)_{rp}|}{EAC(t)_{rp-1}} \tag{4.8}$$

For time forecasting, the PV, ED and ES method with three performance factors (1, SPI and SCI = SPI * CPI) will be used. In addition, the 6 forecasting methods proposed by Elshaer (2013) will be tested, leading to 15 time-based prediction methods. For the predictions of a project's final cost, 8 different performance factors are used. The calculation of the different forecasting methods, as well as the key indicators of EVM are summarized in table 4.1. The time performance indicators are known as the Schedule Variance (SV) or Schedule Performance Index (SPI), while the cost performance indicators include the Cost Variance (CV) and Cost Performance Index (CPI). Additionally, PC denotes the Percentage Complete and α denotes one of the six sensitivity metrics proposed by Elshaer (2013).

4.3 Computational Experiment

In this section, the methodological steps that were outlined in section 4.2 will be rendered concrete by discussing the settings of the network generation, Monte Carlo simulation and solution quality metric phases. An identical order to the previous section will be maintained throughout the discussion of these phases.

Network Generation 90 Activity On the Node (AoN) project networks with 30 activities and random activity costs and durations were generated. This corresponds with 10 networks for every level of the SP indicator, which ranges from 0.1 to 0.9 in steps of 0.1. In order to generate the networks according to these parameters, the RanGen engine, proposed by Demeulemeester et al. (2003) and refined by Vanhoucke et al. (2008), was employed. The dataset of this study can be found at `http://www.projectmanagement.ugent.be/evms.html` and has been used in previous EVM studies (Vandevoorde and Vanhoucke (2006), Vanhoucke and Vandevoorde (2007), Colin and Vanhoucke (2014) and Wauters and Vanhoucke (2014b)). The baseline costs and durations for the different activities of the network are drawn randomly. For the costs, the lower and upper bounds are equal to 50 and 100, respectively. The costs are entirely variable indicating that the cost deviation is completely in line with a deviation in the duration of an activity. Consequently, it is assumed that the baseline costs are expressed in monetary units per time unit. A longer duration then results in more man-hours required to finish the activity, in turn leading to a higher associated cost. The baseline duration for every activity is also drawn from a random distribution and varies between 20 and 40 units of time.

Monte Carlo simulation Monte Carlo simulations allow for the introduction of time and cost deviations on the activity level which are translated to the project level by EVM measurements. As mentioned in section 4.2, the input settings for the Monte Carlo simulations are based on values for the mode m and the mean μ. In this chapter, a distinction will be made between the general performance and a sensitivity analysis. The CV will be used as the main instrument to vary the distributions and is defined as $\frac{\sigma}{\mu}$. The standard deviation σ of the generalized beta distribution is provided in equation

Use		Scenario	a	b	m	μ	θ_1	θ_2	CV
General Performance		Early	0.1	2	0.5	0.6	2.93	8.22	0.4
		On Time	0.2	4	0.82	1	2.94	11.03	0.4
		Late	0.2	4	1.2	1.4	2.83	6.14	0.4
Sensitivity	Δ_m	Early	0.1	2	0.55	0.6	5.45	15.25	0.3
			0.1	2	0.41	0.6	1.78	5	0.5
		On Time	0.2	4	0.91	1	5.39	20.22	0.3
			0.2	4	0.67	1	1.81	6.79	0.5
		Late	0.2	4	1.30	1.4	5.28	11.45	0.3
			0.2	4	0.98	1.4	1.69	3.67	0.5
	Δ_μ	Early	0.1	2	0.50	0.55	5.44	17.57	0.3
			0.1	2	0.50	0.71	1.68	3.55	0.5
		On Time	0.2	4	0.82	0.91	5.29	23.16	0.3
			0.2	4	0.82	1.19	1.79	5.07	0.5
		Late	0.2	4	1.20	1.30	5.37	13.25	0.3
			0.2	4	1.20	1.65	1.53	2.49	0.5

Table 4.2: Overview of the settings of the Monte Carlo simulations

(4.9).

$$\sigma = \sqrt{\frac{(b-a)^2\theta_1\theta_2}{(\theta_1+\theta_2)^2(\theta_1+\theta_2+1)}} \tag{4.9}$$

For the general performance, three different scenarios are used, where each scenario has a CV equal to 0.4. However, by changing the mean and mode, the Monte Carlo simulations imitate an early, on time and late performance respectively. For the sensitivity analysis, the CV is varied from 0.3 and 0.5 in steps of 0.1. We opted to change the CV by changing the mode of the distribution while keeping the mean constant (denoted by Δ_m) and by changing the mean while keeping the mode constant (denoted as Δ_μ). A detailed overview of the settings for the general performance, where the CV is equal to 0.4, and the sensitivity analysis is given in table 4.2. Every project is simulated 1,000 times using the generalized beta distribution with one of the settings of table 4.2.

Solution quality metrics The progress of the projects is measured at different points in time for two reasons. First of all, progress data is collected in order to construct the mean based on a sufficient number of data points. Second, capturing data at different points in time allows to inspect the MAPE and Mean Lags throughout the progress of the project. In the computational study, EVM data was collected at every 5% complete, leading to $R = 20$ data points per simulation run.

4.4 Results

In this section, an overview of the results of the computational experiment is given. For the simulation of the EVM data, P2 Engine (Vanhoucke (2014)) was used. P2 Engine is a command line utility tool based on the LUA scripting language that allows researchers to generate project data. The calculation of the stability and further analyses were performed on Ghent University's High Performance Computing infrastructure. The computational experiment was run on the Delcatty cluster, which has 64 GB RAM available and makes use of a quad-core Intel Xeon processor with 2.6 GHz.

As mentioned in section 4.2, the MAPE and Mean Lags will be reported as the criterion for forecasting accuracy and stability, respectively. Consequently, if the MAPE or Mean Lags are reported for a percentage complete, this reflects the Absolute Percentage Error or Lags, averaged up until the period under study. This criterion was chosen because it seems more reasonable to assess a method's performance on average, rather than at a specific time instance, as would be the case if the APE or Lags would be reported.

This section is structured as follows. In section 4.4.1, the results of the Early, On Time and Late scenarios are studied. An answer is given as to whether the most accurate method coincides with the most stable method and what the impact is of the SP factor and the Percentage Complete. In section 4.4.2, we turn towards the sensitivity of the results by studying a change in CV. The section is broken down into two parts, depending on whether a change in the mode (section 4.4.2.1) or a change in the mean (section 4.4.2.2) is studied.

4.4.1 General results

In this section, the performance for the Early, On Time and Late scenarios with a CV of 0.4 is studied. The settings of these scenarios can be found in table 4.2. The main research question that begs an answer is whether the most accurate method is the same as the most stable method. Afterwards, the results are disaggregated by looking at the impact of the SP indicator and the Percentage Complete.

4.4.1.1 General accuracy and stability observations

The results of the time forecasting methods are given in table 4.3, whereas the results of the cost forecasting methods are provided in table 4.4. For the prediction of a project's final duration, the best Planned Value (PV), Earned Duration (ED), Earned Schedule

(ES) and Elshaer method is reported. Table 4.3 also has a column entitled Criterion. This column indicates whether the most accurate or most stable method is provided. In order to make a trade-off between accuracy and stability, the MAPE and Mean Lags of the most accurate and the most stable method is given. The principal take-aways from tables 4.3 and 4.4 can be summarized as follows:

- The most accurate forecasting method generally does not coincide with the most stable forecasting method. For the time dimension, the methods with a PF equal to SPI (SPI(t)) or the SI yield the lowest MAPE. The methods with a PF equal to 1 lead to the lowest values for the Mean Lags.

- For the On Time scenario, methods with a Performance Factor (PF) equal to 1 prove to be most accurate and most stable. This scenario is the only situation in which the most accurate and most stable method is the same.

- Time: for the Early scenario, there is a large difference in MAPE between the most accurate and most stable method. This is due to the fact that methods with a PF equal to 1 do not score well for projects finishing early. For the other scenarios the difference in MAPE is often double the difference in Mean Lags.

- Cost: the difference between the most accurate and most stable method is generally small except for the Late scenario. In this case, the difference can amount to 7%.

The most stable methods are characterized by a performance factor equal to 1. Hence, these methods assume that the remainder of the project will be executed according to the baseline schedule. While this is not beneficial from an accuracy point of view, these methods yield stable predictions. The most accurate methods take the current progress into account (PF $\neq$ 1).

4.4.1.2 Impact of the SP indicator

The topological structure of the projects was varied by means of the SP indicator, which ranged from 0.1 (more parallel network) to 0.9 (more serial network) in steps of 0.1. Previous studies pointed out that there is a decreasing trend in MAPE as the SP indicator rises for time forecasting (Vanhoucke (2011)), whereas no relation between the SP indicator and MAPE can be established for cost forecasting (Wauters and Vanhoucke (2014b)). The reason for this improvement for more serial networks lies in the fact that the critical path masks fewer deviations compared to a more parallel network, resulting in fewer false warning signals and an improved prediction of the project's duration. Both

Scenario	Criterion	Metric	PV	ED	ES	Elshaer
Early	Accuracy	PF	SPI	SPI	SPI(t)	SI
		MAPE	16.22%	16.22%	**10.70**%	10.84%
		Mean Lags	5.48%	5.48%	4.04%	**3.71**%
	Stability	PF	1	1	1	SI
		MAPE	34.28%	39.37%	34.82%	**10.84**%
		Mean Lags	2.92%	**2.59%**	2.68%	3.71%
Medium	Accuracy & Stability	PF	1	1	1	SI
		MAPE	7.84%	7.54%	**7.14**%	10.80%
		Mean Lags	1.73%	1.52%	**1.38**%	3.66%
Late	Accuracy	PF	SPI	SPI	SPI(t)	SI
		MAPE	16.46%	12.66%	**10.63**%	10.82%
		Mean Lags	5.58%	5.18%	4.07%	**3.75**%
	Stability	PF	1	1	1	SI
		MAPE	21.68%	18.66%	17.27%	**10.82**%
		Mean Lags	2.73%	2.41%	**2.09**%	3.75%

Table 4.3: Time forecasting results (main experiment)

Scenario	Metric	EAC_1	EAC_2	EAC_3	EAC_4	EAC_5	EAC_6	EAC_7	EAC_8
Early	MAPE	31.47%	6.99%	8.97%	7.79%	20.40%	21.33%	7.00%	**6.97**%
	Mean Lags	2.64 %	**2.45**%	3.96%	3.25%	5.59%	5.32%	2.60%	2.54%
On Time	MAPE	**3.94**%	6.91%	7.54%	7.75%	11.74%	11.97%	6.88%	6.90%
	Mean Lags	**0.95**%	2.40%	3.27%	3.22%	4.69%	4.69%	2.48%	2.49%
Late	MAPE	14.18%	7.03%	8.18%	7.75%	17.88%	19.99%	7.01%	**6.99**%
	Mean Lags	**1.80**%	2.50%	3.62%	3.28%	5.41%	5.31%	2.56%	2.58%

Table 4.4: Cost forecasting results (main experiment)

the presence of the decreasing MAPE trend for time forecasting and the absence of any trend for cost forecasting could be corroborated in this experiment.

However, our main interest lies in the behaviour of the Mean Lags for time and cost forecasting. This is depicted in figure 4.1 for cost forecasting and figures 4.2(a)-4.2(b) for time forecasting. The x-axis shows the different levels for the SP indicator. The y-axis shows the Mean Lags as a percentage, following equation (4.8). For time forecasting, a distinction is made between the Mean Lags of the most accurate methods (figure 4.2(a)) and the most stable methods (figure 4.2(b)).

- The stability indicates an opposite trend compared to the accuracy: as the SP indicator rises, the mean lags increase as well. Because the project progress is almost exclusively governed by the progress of an individual activity for serial networks, the variability of the key EVM indicators can be larger compared to parallel projects. For parallel projects, the project performance comprises that of multiple activities, which causes the performance of one activity to be compensated by the performance of another activity. Consequently, as a project becomes more serial, the predictions become on average more accurate but more fluctuations appear.

- For time forecasting, the most stable methods (figure 4.2(b)) display a smaller increase in Mean Lags compared to the most accurate methods found in figure 4.2(a).

- For cost forecasting, the performance of EAC_1 is remarkable. The Mean Lags barely increase as the SP indicator rises. For an SP level equal to 0.1, many methods have a better stability than EAC_1 but because of the increasing trend of the other methods, their stability is worse from an SP level of 0.3 onwards.

4.4.1.3 Impact of the Percentage Complete

Similar to the section studying the SP indicator, the Percentage Complete also leads to different conclusions based on whether the most accurate or most stable method is employed. The impact on the time dimension is shown in figures 4.3(a) and 4.3(b), where the x-axis shows the Percentage Complete and the y-axis indicates the Mean Lags. Interestingly, the trend of the most accurate and most stable methods differs along the percentage complete:

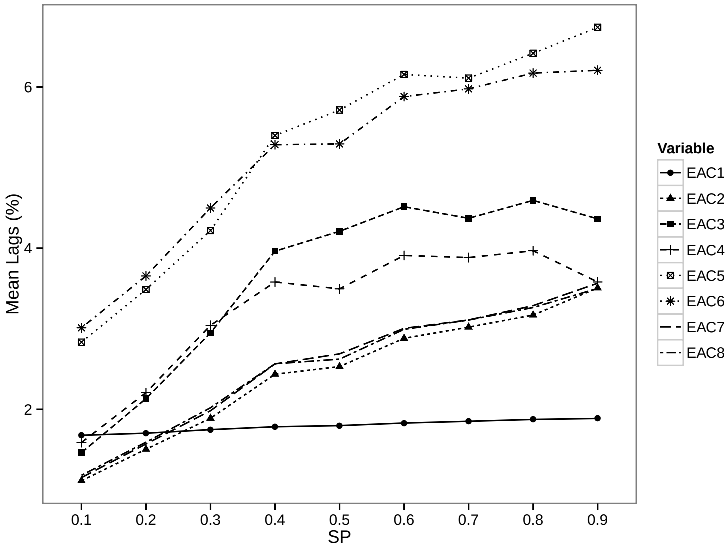
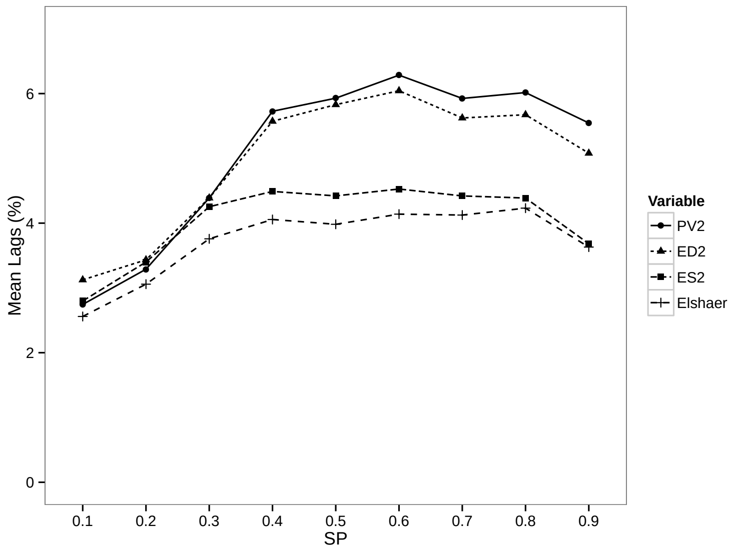


Figure 4.1: Impact of the SP factor on stability for cost forecasting

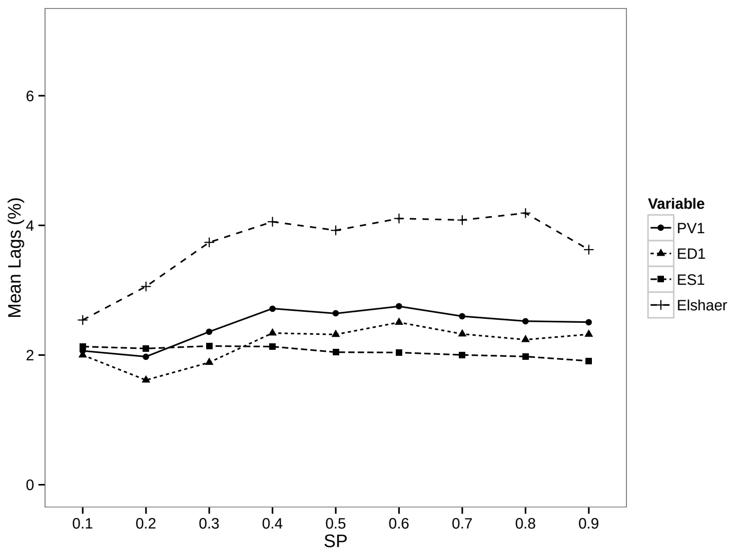
- The Mean Lags of the most accurate methods display a decreasing trend along the percentage complete, indicating that the stability improves over time.
- For the most stable methods, the Mean Lags slightly increase, meaning that the stability shows a minor deterioration across time. It is worth noting that the Mean Lags are still considerably lower than those of the most accurate methods.
- The Elshaer method with SI as its performance factor shows the best performance for accuracy and stability, as established previously.

4.4.2 Sensitivity analysis

The scenarios that were used to judge the general performance have a CV of 0.4. In order to assess the impact of a change in the distribution, the CV was varied to 0.3 and 0.5 by changing the mode while keeping the mean constant and by changing the mean while keeping the mode constant. The results of both situations can be found in tables 4.5 and 4.6. In table 4.5, the accuracy, as measured by the MAPE, and stability, measured by the Mean Lags, is given for the most accurate and most stable PV, ED, ES and Elshaer methods. The first 6 rows represent the percentages of a change in the mode, while the lower half shows the influence of a change in the mean. These situations will be analyzed in sections 4.4.2.1 and 4.4.2.2 respectively.

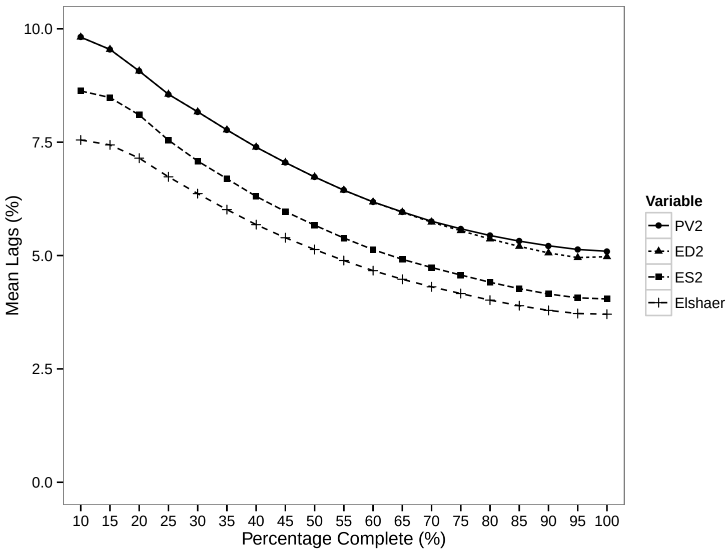


(a) Time: stability for the most accurate methods

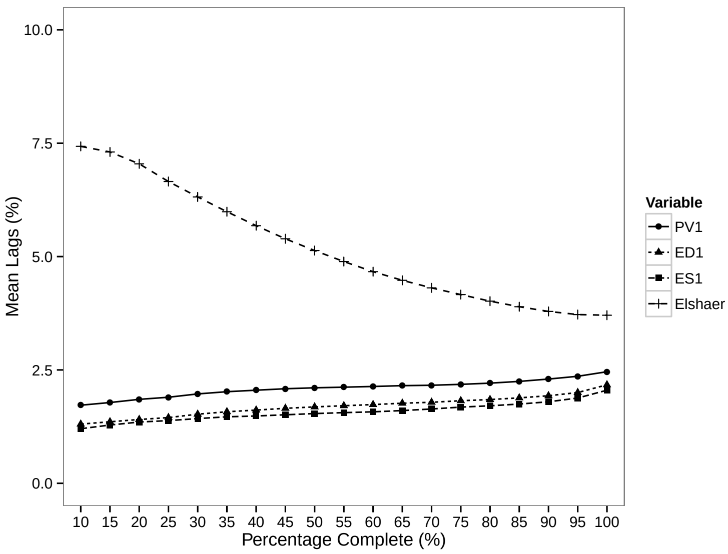


(b) Time: stability for the most stable methods

Figure 4.2: Impact of the SP factor on stability for the most accurate (4.2(a)) and most stable (4.2(b)) methods for time forecasting



(a) Time: stability for the most accurate methods



(b) Time: stability for the most stable methods

Figure 4.3: Impact of the percentage complete on stability for the most accurate (4.3(a)) and most stable (4.3(b)) methods for time forecasting

4.4.2.1 Impact of a change in mode

The key observation following from the top half of table 4.5 is that the MAPE and Mean Lags of all methods increase as the CV rises. However, it is possible to make a distinction between the behaviour of the most accurate and most stable methods, as follows:

- The increase in MAPE for a rising CV is higher for the most accurate methods than for the most stable methods. The average increase in MAPE_t for the most accurate methods is equal to 4.64%, while this number is only 2.26% for the most stable methods. For cost forecasting, the difference in accuracy is 0.78% for the most stable method (EAC_1) and 4.11% for the most accurate method (EAC_8).
- A similar observation holds for the Mean Lags, where the increase in stability is lower for the most stable methods than for the most accurate methods.

While the most stable methods are at a disadvantage when it comes to accuracy, these methods gain importance as the CV increases since the decrease in performance is less steep compared to the most accurate methods.

4.4.2.2 Impact of a change in mean

The impact of a change in mean largely shares the effect of a change in mode. The MAPE of the most accurate methods rises along the CV and the Mean Lags rise as well, indicating a decrease in stability for an increasing CV. It is worth noting that the values for the MAPE and Mean Lags are slightly higher than those of a change in the mode, suggesting that a change in mean has a slightly more pronounced effect than a change in mode.

When looking at the overall accuracy of the most stable methods, no consistent trend is followed. Sometimes, the MAPE increases or decreases as the CV changes. While this finding appears to be an anomaly, it can be explained as follows:

- The increase or decrease of the MAPE for a change in CV is determined by the performance of the forecasting method for the different modes. For the On Time and Late scenarios (m equal to 0.82 and 1.2, respectively), the MAPE increases for a CV increment. However, for the Early scenario (m equal to 0.5), a rising CV entails that the mean of the generalized beta distribution is pushed closer to 1 (cf table 4.2). Since the most stable methods are those with a performance factor equal to 1, the MAPE decreases for a rising CV. The behaviour of the *overall*

MAPE as shown in table 4.5 depends on whether the decrease in MAPE for $m = 0.5$ offsets the increase in MAPE when m equals 0.82 or 1.2.

- The behaviour of the MAPE of the Planned Value method is caused by the decrease for the Early scenario which cancels out the increase in MAPE for the other two scenarios. This is not the case when the CV goes from 0.4 to 0.5. Hence, the average MAPE in table 4.5 increases again.

- The Earned Schedule method also exhibits a strange evolution with a MAPE that skyrockets when the CV moves from 0.3 to 0.4 and slightly decreases afterwards. This is due to the fact that the most stable method for a CV equal to 0.3 and m equal to 0.5 is ES_2. The stellar forecasting performance of this method greatly influences the average MAPE, thus explaining the low MAPE value for a CV of 0.3. In case ES_1 was the most stable method across all levels for the CV and mode, the MAPE would follow the characteristic, decreasing trend as the mean of the generalized beta distribution lies closer to 1.

While the discussion of the inconsistent MAPE trend was limited to time forecasting, an identical train of thought applies to the cost forecasting results of table 4.6.

	Metric	CV	Most Accurate Method				Most Stable Method			
			PV_{SPI}	ED_{SPI}	$ES_{SPI(t)}$	$Elshaer_{SI}$	PV_1	ED_1	ES_1	$Elshaer_{SI}$
ΔMode	MAPE	0.3	11.50%	10.21%	7.08%	8.05%	20.59%	21.38%	19.24%	8.05%
		0.4	13.51%	12.14%	9.49%	10.82%	21.27%	21.86%	19.74%	10.82%
		0.5	15.67%	14.18%	11.92%	13.61%	22.03%	22.38%	20.30%	13.61%
	Mean Lags	0.3	3.84%	3.63%	2.38%	2.73%	2.30%	2.01%	1.86%	2.73%
		0.4	4.27%	4.06%	3.17%	3.71%	2.46%	2.17%	2.05%	3.71%
		0.5	4.77%	4.57%	3.97%	4.73%	2.64%	2.35%	2.27%	4.73%
ΔMean	MAPE	0.3	11.82%	11.11%	7.69%	8.06%	22.39%	24.25%	9.58%	8.06%
		0.4	13.51%	12.14%	9.49%	10.82%	21.27%	21.86%	19.74%	10.82%
		0.5	18.16%	15.34%	13.26%	13.79%	22.49%	20.77%	19.08%	13.79%
	Mean Lags	0.3	3.77%	3.62%	2.40%	2.73%	2.34%	2.04%	1.92%	2.73%
		0.4	4.27%	4.06%	3.17%	3.71%	2.46%	2.17%	2.05%	3.71%
		0.5	4.97%	4.69%	4.16%	4.85%	2.74%	2.50%	2.34%	4.85%

Table 4.5: Time forecasting results (sensitivity analysis)

	Metric	CV	Method							
			EAC_1	EAC_2	EAC_3	EAC_4	EAC_5	EAC_6	EAC_7	EAC_8
ΔMode	MAPE	0.3	16.16%	4.98%	6.37%	5.62%	14.83%	15.89%	5.00%	4.97%
		0.4	16.53%	6.98%	8.23%	7.76%	16.67%	17.76%	6.96%	6.95%
		0.5	16.94%	9.13%	10.27%	10.03%	19.03%	20.17%	9.08%	9.08%
	Mean Lags	0.3	1.68%	1.73%	2.95%	2.38%	4.17%	3.91%	1.83%	1.80%
		0.4	1.79%	2.45%	3.62%	3.25%	5.23%	5.11%	2.55%	2.54%
		0.5	1.95%	3.24%	4.37%	4.16%	6.44%	6.43%	3.33%	3.34%
ΔMean	MAPE	0.3	18.51%	4.98%	6.44%	5.62%	14.79%	15.63%	5.00%	4.97%
		0.4	16.53%	6.98%	8.23%	7.76%	16.67%	17.76%	6.96%	6.95%
		0.5	15.80%	9.29%	10.43%	10.17%	19.96%	22.32%	9.24%	9.23%
	Mean Lags	0.3	1.76%	1.73%	2.97%	2.38%	4.19%	3.93%	1.83%	1.80%
		0.4	1.79%	2.45%	3.62%	3.25%	5.23%	5.11%	2.55%	2.54%
		0.5	1.99%	3.31%	4.40%	4.27%	6.56%	6.67%	3.39%	3.42%

Table 4.6: Cost forecasting results (sensitivity analysis)

4.5 Empirical Validation

In the previous section the results of the computational experiment that made use of simulated data were discussed. The goal of this section is to establish whether the findings are valid for practical environments. In general, the methodology can be translated as follows. First of all, the SP indicator can easily be calculated using the Activity on the Node network (AoN) of the project under study. In the previous section, the whole spectrum of SP indicators was investigated, whereas in practice each project corresponds with 1 value of the SP indicator. Secondly, practitioners can use historical data to construct one or multiple distributions in order to test the stability of the different forecasting methods. Hence, they can choose to evaluate all forecasting methods or only a subset. In the computational experiment, multiple distributions were evaluated as well as the effect of changes in the mode and mean in order to provide recommendations that are valid for a wide range of projects and progress data.

In section 4.5.1, the data of both projects are described. Section 4.5.2 links the observations of the two projects back to the computational study. We check which of the results of the computational study are reflected in the empirical validation and provide an explanation for those results that do not correspond.

4.5.1 Project Data

Data from two real-life consultancy projects were analyzed with the goal of validating the findings of the computational experiment. While it is obvious that the findings of two projects cannot be generalized, they are representative in the way that a consulting methodology that was used throughout the 2000s was applied to these projects. These consultancy projects were performed for one of Europe's leading logistics groups, with plants across multiple countries. The first project involved the determination of an improved schedule for the different machines. The project consisted of three phases, namely data validation, programming the optimization algorithm and charting a blueprint for the implementation in the company's Enterprise Resource Planning (ERP) software. The budgeted cost of this project was equal to €249,780. The second project followed up on this study and involved implementing the improved job scheduler in the ERP software. It counted multiple concurrent phases, often related to functional and technical analyses of different IT components. Within the Work Breakdown Structure, 5 different activity blocks could be identified. The first two are related to an optimal search for the best location. The third block involved reporting the optimal allocation from orders to order pickers. The fourth block took care of overbooking the existing inventory of two locations. The last part of the project contained the development of new work instructions for the people on the floor. The budget at completion for this project was €139,263. The difference in structure is evidenced by the SP indicator, which was equal to 0.23 for the optimization project, while the follow-up project, the ERP implementation, had a value of the SP indicator of 0.52.

Throughout the progress of both projects, time and costs were monitored using Earned Value Management. At each review period, a forecast was made using the same methods that were introduced in section 4.2. It is worth noting that the main purpose of these forecasts was to communicate to the company's management, rather than to act as a signal for corrective action. Because of the highly specialized nature of both projects, adding human resources to the project in a short timespan proved impossible.

For both projects, an overview of the project data is given in table 4.7. This table contains data that resulted from the planning, the actual situation as well as dynamic progress data. The SCI (SCI(t)) denotes the Schedule Cost Index and is equal to the product of the CPI and SPI (SPI(t)). For the optimization project, there were 13 review periods whereas 23 review periods were used for the ERP implementation project. The progress of the optimization project indicates an excellent schedule performance, with the project ending early. However, the costs turned out to be slightly higher than expected: the CPI at project's end is equal to 0.87. The total project cost was €286,211 and the project ended 8 weeks earlier than anticipated. The reason for the cost overrun

	Optimization Project				ERP project			
				Plan				
Parameter				Value				
SP	0.23				0.52			
PD	106 days				95 days			
BAC	€249,780				€139,263			
				Execution				
RD	65.46 days				109.5 days			
RAC	€286,211				€159,657			
				Performance Data				
	Min	Max	Average	Final	Min	Max	Average	Final
CPI	0.84	0.93	0.88	0.87	0.83	0.89	0.87	0.87
SPI	1.52	2.13	1.81	1.64	0.72	1.00	0.90	1.00
SPI(t)	1.47	2.40	1.73	1.63	0.77	0.86	0.84	0.83
SCI	1.31	1.85	1.58	1.43	0.60	0.87	0.78	0.87
SCI(t)	1.26	2.16	1.52	1.42	0.64	0.76	0.73	0.72

Table 4.7: Summary of the data of the two empirical projects

lies in the fact that is was difficult to estimate the baseline costs. For some activities, the actual cost was much lower than anticipated, whereas the opposite conclusion holds for other activity groups. The minimum activity CPI was equal to 0.63 (for determining the criteria that led to a choice of location) and the maximum activity CPI was equal to 2 (for robustness checks). The implementation of the improved job scheduler turned out to be more burdensome. The final CPI, SPI and SPI(t) are all lower than 1, indicating that the actual situation deviates from the plan. The project is behind schedule and suffers from a slight cost overrun. This overrun is caused by many activities that exceed their budgeted cost moderately. Additionally, the project suffered a 4 week delay. It is worth remarking that table 4.7 demonstrates the unreliable behaviour of the SPI since it gravitates towards 1 at the end of the project.

4.5.2 Observations

In this paragraph, the observations of time and cost forecasting methods are linked back to the findings of the computational study, which were outlined in section 4.4. Because of a lack of historical data, the Elshaer methods were not included as a benchmark for time forecasting. The observations can be summarized as follows:

- For both projects, the most accurate method does not coincide with the most stable one. This is in line with the results from the computational study. Table 4.8 contains the best performing method and the MAPE and Mean Lags percentages.

The discussion is limited to the conclusions of the table.

- The most accurate time forecasting method is ES_{SPI} whereas the most stable time forecasting method is ES_1.
- The most accurate cost forecasting method is EAC_2 for both projects. This is little surprising since this method displayed a similar performance to EAC_8 in the computational results of section 4.4.1. EAC_1 is the most stable forecasting method.

- The MAPE of the ERP project is lower than that of the Optimization project. This result is interesting because it confirms that a project with a more serial network structure (a higher value for the SP indicator) leads to a lower forecasting error.

- The Mean Lags of the ERP project are lower compared to the Mean Lags of the Optimization project. While this seems to contradict the findings of section 4.4.1.3, there is a simple explanation for this behaviour. The link between the SP indicator and the Mean Lags does not hold because of the volatile SPI of the Optimization project. Contrary to this, the SPI of the ERP project is much more stable. Consequently, this differing SPI behaviour would be equivalent to comparing the performance of two projects that originated from different distributions in the computational experiment.

- The forecasts generally improve as the Percentage Complete increases. There is one exception, namely for the Optimization project, where the Planned Value and Earned Duration methods with SPI or SCI as the performance indicator do not improve along the Percentage Complete. Similar to the Mean Lags behaviour discussed above, this is due to the instability of the SPI indicator, which is reflected in the forecasting estimates.

	Optimization Project				ERP project			
	Time		Cost		Time		Cost	
	Method	%	Method	%	Method	%	Method	%
MAPE	ES_{SPI}	8.06%	EAC_2	2.01%	ES_{SPI}	3.48%	EAC_2	1.00%
Mean Lags	ES_1	4.19%	EAC_1	1.07%	ES_1	0.90%	EAC_1	0.59%

Table 4.8: MAPE and Mean Lags of the best performing method of the two empirical projects

4.6 Conclusion

In this chapter, we focused on a topic that received little attention in the project control community, namely the stability of forecasting methods. Attention was accorded to three distinct contributions. First and foremost, a new criterion for measuring stability was proposed. Employing the Mean Lags of the project outcome predictions comes with two advantages. First of all, the interpretation and scale of this metric is similar to the MAPE, which was used to assess forecasting accuracy, hence facilitating a trade-off between accuracy and stability. Second, the Mean Lags do not make use of arbitrary thresholds, which was a weakness of the currently used stability rule-of-thumb.

The second contribution involved setting up a large computational experiment in order to assess stability using the newly proposed Mean Lags metric. Looking at prediction methods from a simulation point of view was particularly welcome for the cost forecasting methods. The results of the computational study revealed some interesting managerial insights. It was found that, apart from the On Time scenario, the most accurate method generally does not coincide with the most stable method. The relation between accuracy and the SP indicator does not hold for the stability criterion. On the contrary, the tests brought to light that the stability deteriorates for an increase in SP value. The stability of the most accurate methods improves along the percentage complete, while the stability of the most stable methods remains constant or displays a slight deterioration. Finally, a sensitivity analysis was executed. The CV was changed following a change in the mean or the mode of the underlying generalized beta distribution. In general, a higher CV leads to a worse accuracy and stability. However, the performance drop was less steep for the most stable methods compared to the most accurate methods.

Ultimately, the choice for a certain forecasting method depends on a project manager's preference. However, the conducted experiments allow to make a couple of recommendations. The general experiments showed that the most stable methods possess a MAPE that is twice as high but also twice as stable compared to the most accurate methods. Because the difference in MAPE is rather large, one would be inclined to opt for the most accurate method. However, in case the project manager fails to assert the variability of the activities' duration in an adequate manner, the MAPE and Mean Lags rise rather steeply compared to the most stable method. For the time dimension, the Elshaer method with the Significance Index (SI) strikes a good balance between accuracy and stability. For the cost dimension, it is more difficult to put one method forward.

Depending on one's inclination towards accuracy or stability, EAC_8 or EAC_1 should be used. However, given the poor accuracy of EAC_1, we recommend the use of EAC_8 for cost forecasting.
The final contribution validated the computational findings using two real-life projects. It was shown how the research methodology can be used for practical environments. While generalizations based on two projects are hard to make, the real-life studies were representative because they followed a consulting approach that was used frequently throughout the past decade. The first project aimed to find an improved schedule for the machines, while the second project involved the implementation of an ERP system. The projects asserted the excellent stability performance of the EAC_1 and ES_1 method, while the forecasting accuracy depends on the convergence (or lack thereof) of the schedule and cost performance.

No research is without its limitations. The stability tests in this chapter were run on a set of fictitious data, as well as on a limited set of empirical data. Consequently, academics and practitioners should view this chapter as an open invitation to use the proposed Mean Lags criterion for testing on additional real-life data, which are abundant in the project control literature. A second limitation lies in the reliance on deterministic forecasting methods. Future research could focus on evaluating the stability of the maximum likelihood estimate of stochastic forecasting methods.

5

A comparative study of Artificial Intelligence methods for project duration forecasting

Artificial Intelligence (AI) methods attempt to learn the relation between data inputs and one or multiple output values. In this chapter, the forecasting performance of five AI methods is benchmarked against the best performing Earned Value Management/Earned Schedule methods. A methodology that employs Monte Carlo simulation, Principal Component Analysis, grid search and cross-validation is proposed. The forecasting accuracy of all prediction methods is compared across a topologically diverse project dataset. A large computational experiment demonstrates the strength of the AI methods when the training and test sets coincide. A sensitivity analysis examines the effects of a change in the mean and standard deviations of the underlying statistical distribution in order to establish the limitations of methods that rely on historical data.

5.1 Introduction

With the advent of the Critical Path Method (CPM, Kelley (1961); Kelley and Walker (1959)) and the Program Evaluation and Review Technique (PERT, Fazar (1959)), project planning increasingly became a separate research discipline. While the former method focuses on the construction of a baseline schedule, the latter turns the attention to the relation between the project duration and activity duration variability. The construction of the baseline plan should be accompanied by an identification of its weak spots, namely those activities that are most likely to have the largest and possibly detrimental impact on the project outcome. As such, regardless of the presence of limited resources, the baseline schedule should be seen as a point of reference with which actual performance can be compared and contrasted. By relating the actual performance of the project to the planned performance, corrective actions can be triggered as soon as the project is deemed to be out of control. These three aspects, baseline scheduling, schedule risk analysis and project control, are coined *dynamic scheduling* (Uyttewael (2005) and Vanhoucke (2012b, 2014)).

The emphasis of this chapter lies on one of the dynamic scheduling aspects, namely project control. A popular methodology for tracking project progress is Earned Value Management (EVM). It originated at the US Department of Defense (DoD) in the 1960s and aids project managers in controlling the projects' time and cost by using three key metrics that form the foundation of a number of performance indicators. An overview of the fundamentals of EVM can be found in Fleming and Koppelman (2005). While initial studies were dominated by the cost objective, the work of Lipke (2003) proved to be a turning point as Earned Schedule (ES) allowed project managers to track progress in units of time rather than monetary units. Along with the inception of Earned Schedule, academic studies shifted to the time dimension. A popular topic involved forecasting the final duration (or previously the project's real cost) based on progress data. The importance of an accurate estimate for the project's end cannot be underestimated. Not only does it allow a project manager to look ahead, it also implies an implicit call for action. Forecasting methods can be embedded in decision support systems that trigger a warning once the expected duration exceeds a user-defined threshold. Obviously, the validity of this warning signal greatly depends on the trustworthiness of the underlying forecasting method. The performance of three Planned Value, three Earned Duration and three Earned Schedule methods has been investigated on simulated data (Vanhoucke and Vandevoorde (2007)) as well as on real-life projects (Vandevoorde and Vanhoucke

(2006)). Vanhoucke (2010b, 2011) advocated the integration of top-down control systems such as EVM with bottom-up sensitivity indicators which result from schedule risk analysis. This was followed up by Elshaer (2013) who proposed an adaptation of the Earned Schedule forecasting method using activity sensitivity information.

In this chapter, a new class of methods is implemented to construct project duration estimates. Artificial Intelligence (AI) is a research branch dedicated to learning the relation between inputs and outputs and applying that relation for classification or prediction purposes. To the best of our knowledge, only three works deal with one AI method that involves EVM metrics. Cheng et al. (2010) deploy a Support Vector Machine (SVM) to estimate the final cost of two construction projects. The parameters are tuned with a fast messy genetic algorithm. The combination of a basic meta-heuristic as a tuning mechanism and the SVM was united with fuzzy logic. Cheng and Roy (2010) tested this system for function approximation and cost estimation. Wauters and Vanhoucke (2014b) applied Support Vector Regression for project control time and cost forecasting. The authors compared its performance with the best performing EVM and ES methods on a large dataset and revealed the pitfalls in a robustness experiment. The computational experiment revealed that SVMs outperform the current EVM forecasting methods when the training set is equal or at least similar to the test set.

The implementation of Support Vector Machines heralded the introduction of Artificial Intelligence methods in the project control community. To the best of our knowledge, no Artificial Intelligence method other than SVMs has been incorporated in an EVM setting. The goal of this chapter consists of introducing Artificial Intelligence methods for constructing better predictions of the final duration of a project. The contribution of this chapter to the body of project control literature is fourfold. First, we propose five Artificial Intelligence methods for predicting a project's final duration. AI methods use historical or simulated data to learn the relation between inputs and one or multiple outputs. In a project control environment, the simulated data contains information with regard to the progress of the project. The proposed methods learn how the performance indicators are related to the project's Real Duration (RD). This knowledge is then applied to new data to come up with an estimate of the project's final duration. Secondly, a generally applicable methodology is put forward starting with the generation of project and progress data. This serves two purposes. The first purpose lies in the nature of the AI methods which learn a relation from existing data. Secondly, a wide array of data allows us to reach general conclusions. Apart from the data generation, a decision needs to be made on which progress data will be fed to the AI methods. These progress data contain periodic measurements of EVM performance indicators, as well as EVM forecasting estimates. The high volume of data may be correlated and some data may be irrelevant. In order to alleviate these problems, Principal Component Analysis is applied. This data pre-processing technique forms a linear combination of the progress data, eliminating noise and retaining only the data that explains most of the variation. Once the relevant data are selected, the AI methods need to be fine-tuned, since their performance is highly dependent on the chosen parameters. By dividing the historical data into a smaller training and a validation set, the prediction performance on the validation set will be used to find the best parameters for each method. The parameters are used to construct forecasts for the projects of the data generation step. Hence, the third contribution lies in assessing the general performance of the AI methods and comparing it against the EVM/ES methods. Incidentally, the impact of the project network's topology and percentage complete is identified. Finally, the main experiment assumes that the project manager can provide an accurate estimate of the variability. As a result, the training and test sets are drawn from a distribution with identical parameters. In real-life situations, it is extremely difficult to appraise the variability affecting project activities. In line with this reasoning, a robustness experiment in which the training and test sets no longer coincide is set up. By varying the mean and standard deviations of the underlying distributions the vulnerabilities of data-rich methods are identified. The situation in which the training and test sets are similar but not identical to one another has been tested as well, showing that the proposed methods still outperform the current EVM/ES forecasting methods.

The outline of this chapter is as follows. In section 5.2, an overview of the Artificial Intelligence methods is supplied. Section 5.3 goes over the process of data generation, the EVM progress data, data pre-processing with PCA and training, validation and test sets to construct a forecast. The steps of the methodology are revisited in section 5.4 which proffers specific settings. The results found in section 5.5 first elaborate on fine-tuning the parameters of the AI methods (section 5.5.1.1) and the desired level of explained variation for the principal components (section 5.5.1.2). Section 5.5.2 discusses the general accuracy, the impact of the topology and the percentage complete. The sensitivity analysis varies the mean and standard deviation of the underlying distribution and proves how data-intensive methods conform to the well-known "garbage-in, garbage-out" principle. Section 5.6 concludes this chapter by sharing the main insights of our research.

5.2 Artificial Intelligence methods

In this section, a general overview of the employed Artificial Intelligence methods will be given. All of these methods will be used to construct a forecast of the final project's duration. In section 5.3, it will be shown how these techniques are embedded in the presented methodology.

5.2.1 Decision Tree Learning

5.2.1.1 Decision Trees

Decision Tree learning finds its roots in the seminal work by Morgan and Sonquist (1963). In their paper, the authors deal with automated interaction detection and propose a new procedure for data analysis and regression, which is now known as decision tree learning. Inspired by this new research direction, Breiman et al. (1984) and Quinlan (1993) independently came up with algorithms that are comprised of two phases. In the first phase, the solution space is partitioned using a binary (Breiman et al. (1984)) or multi-way (Quinlan (1993)) split after which, in the second phase, a constant model is applied to each node of the partition. These algorithms are known as Classification And Regression Trees (CART) and C4.5, respectively. The approach of these well-known techniques is subject to two pitfalls, namely overfitting and selection bias. The overfitting problem results from the lack of statistical significance, as noted by Mingers (1987). Even though some information measure is maximized in order to make a split in the decision tree, there is no way of establishing whether this split is significant. The selection bias follows from the fact that attributes with more split points are preferred. This issue was raised by Breiman et al. (1984) but no remedy was provided.

The solution was found by Hothorn et al. (2006b), who proposed a conditional inference framework that makes use of permutation tests developed by Strasser and Weber (1999). These permutation tests look for dependence between the outcome and the different predictors, after which the predictor with the smallest p-value is selected for splitting. Consequently, the conditional inference framework meets the need for a statistical approach to recursive partitioning, as demanded by White and Liu (1994).

5.2.1.2 Bagging & Random Forest

The principal shortcoming of decision trees lies in their instability when small changes in learning data occur. Variable selection and selection of the cutpoint for the selected variable(s) highly depend on the observations in the learning sample. If the first splitting variable were different due to a minor change in the learning data, the entire structure of the tree may be altered. Hence, single tree predictions display a high variability.

In order to alleviate this shortcoming, a new class of methods called ensemble methods saw the light of day when Breiman (1996) introduced bagging predictors. Bagging, as well as other tree-based ensemble methods, employ the fact that singular trees may yield instable results but produce the right prediction on average. Bagging trains a number of trees on a bootstrap sample of the learning set and applies all the constructed trees on the test set. The final prediction is the average value of the predictions resulting from each tree. The superiority of bagging over singular classification or regression trees was demonstrated by Bühlmann and Yu (2002). In that paper, the authors analyze the reduction of variance following the use of bagging.

A few years later, Breiman (2001) added another source of variation to rectify the shortcoming of decision trees. Random forests make use of a restricted number of predictors that can be selected at each split. Bagging can be regarded as a special case of random forests where the number of randomly preselected predictors coincides with the total number of predictors. Segal (2004) identified the need for careful parameter tuning for settings where maximally sized trees overfit. Random Forests have also been researched from a statistical point of view by linking them with adaptive Nearest Neighbour methods (Lin and Jeon (2006)) or by examining their consistency for classification purposes (Biau et al. (2008)).

5.2.1.3 Boosting

The main premise of boosting, a weak learning algorithm that can be boosted into a strong learner, is very similar to other ensemble methods. However, boosting works differently since it is a stagewise procedure. Base learners, such as decision trees, are fitted to the training data in an iterative fashion, where an increased importance is put on instances that are hard to classify or predict. The first boosting algorithm that can be solved in polynomial time was presented by Schapire (1990). A couple of years later, Freund and Schapire (1997) introduced AdaBoost, a boosting algorithm for classification purposes that solved practical issues which earlier boosting algorithms suffered from. An overview of boosting with the key focus on AdaBoost was given by Schapire (2003). In that contribution, the author examined similarities to game theory and linear programming, as well as extensions for multiclass classification problems. Bagging, boosting and randomization were compared in an experiment conducted by Dietterich (2000), who researched the performance of the three techniques with and without classification noise.

Boosting for regression problems was conceived by Friedman (2001), who developed a general paradigm for boosting based on gradient descent. In general, the boosting procedure for regression proceeds as follows. In the first step, a learner is constructed that maximally reduces a specified loss function. For each following step, the focus shifts to the residuals resulting from the first step. A new learner is fitted to the residuals of the previous step and added to the model. Consequently, by focusing on residuals it is clear that boosting emphasizes instances that are hard to predict. For additional details about the boosting procedure, we refer to Friedman (2001) and the introductory articles of Elith et al. (2008) and Natekin and Knoll (2013). Similar to their perspective on bagging, Bühlmann and Hothorn (2007) examined boosting from a statistical point of view, according special attention to model estimation problems.

5.2.2 Support Vector Machines

Support Vector Machines (SVM) in their current form were developed at AT&T Bell Laboratories and gained momentum with the publication of Cortes and Vapnik (1995). SVMs build a model involving a decision surface by mapping the predictors into a higher-dimensional feature space. In this feature space, linear regression is executed. This translation of the predictors into a different feature space is necessary because of the unknown relation between predictors and outcomes. In case this relation is assumed to be non-linear, it is necessary to employ kernel functions. Any function satisfying Mercer's condition (Vapnik (2000)) can be used as a kernel function. The main goal of the kernel consists of achieving linear separability between training points in the higher-dimensional feature space. The reader is referred to Smola and Schölkopf (2004) for more details on kernel functions for Support Vector Machines.

5.3 Methodology

In this section, it is shown how the Artificial Intelligence methods of section 5.2 are used in a project control environment. Before the methods can be applied, it is necessary to generate progress data from which the methods can learn the relation with the real duration. At the same time, generating a high volume of diverse projects enhances generalization of the results. The generated data is captured for different periods in time. The performance of the project is tracked using EVM and ES metrics, revealing how the project evolves across time. Before the data is fed to one of the AI methods, it is pre-processed such that the problems of noise and correlated data are solved. Next, a division into a training and a test set is made. The *static phase* focuses on the training set and by dividing the training set into a smaller training set and a validation set, the optimal parameters for each of the AI methods discussed in section 5.2 are found. These parameters are used in the *dynamic phase*. The dynamic phase trains an AI method on the training set with the best found parameters and assesses the performance on the executions of the test set. These executions can be regarded as real-life executions. A prediction is made for each of the review periods, resulting in an overall performance measure. The performance measure serves as a tool for comparing the proposed AI methods with the currently known EVM and ES forecasting methods.

Even though the employed methodology has similarities to the one used by Wauters and Vanhoucke (2014b), there are some notable differences. First of all, generation of the progress data is done differently. While we still make use of the generalized beta distribution, a different parametrization is applied to construct simulation scenarios. Secondly, Principal Component Analysis (PCA) is used as a pre-processing technique to combine only the relevant attributes in principal components. The final and largest methodological difference lies in the division between training, validation and test sets. In Wauters and Vanhoucke (2014b), parameters are tuned across multiple projects. In this chapter, the best parameter combination for every individual project is selected.

The methodology is divided into 3 main blocks, namely data generation, attributes and training, validation and testing. Each of these blocks is discussed below. The different aspects of the 3 methodological blocks are depicted in figure 5.1.

5.3.1 Data Generation

The data generation is comprised of two distinct phases. The *baseline data* involve the construction of the project network, as well as assigning baseline costs and durations to the network. The resulting schedule is obtained using the critical path method's earliest start calculations. The fictitious project networks are generated in a controlled manner by varying the Serial/Parallel (SP) indicator. The SP indicator, originally proposed by Vanhoucke et al. (2008), was first called the I_2 indicator but later project management studies adopted the more intuitive name. Its calculation is based on the progressive level concept of Elmaghraby (1977) by determining the maximum number of levels in a network. This corresponds with the longest chain of critical activities in a network.

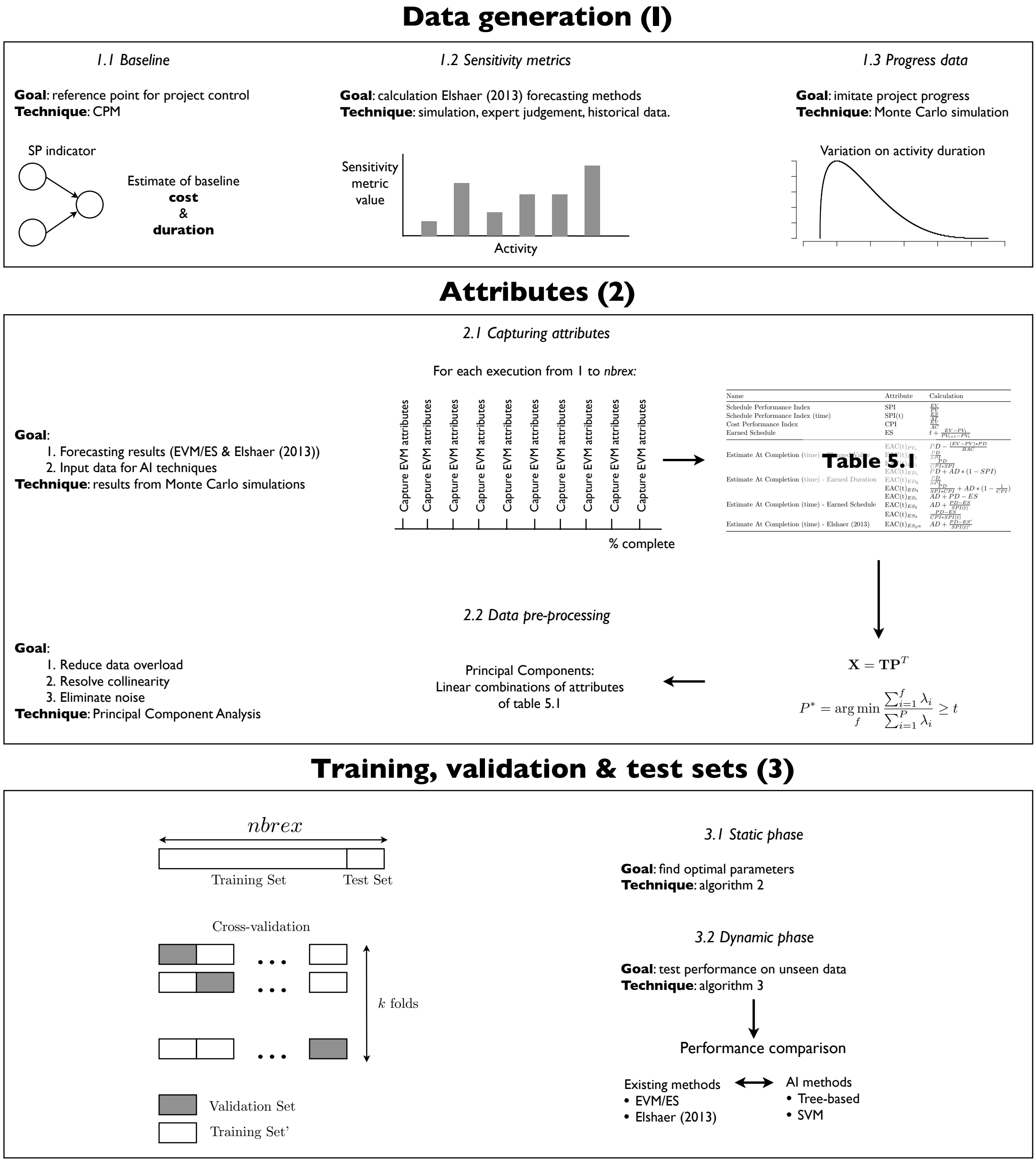


Figure 5.1: Summary of the 3 methodological blocks

Let m denote the maximal progressive level and n the number of non-dummy activities in the project's network. The SP indicator is then equal to $\frac{m-1}{n-1}$. As a result, the SP indicator assumes a value in the interval [0,1] and measures the degree of resemblance to a completely serial (SP=1) or a completely parallel (SP=0) network.

The *progress data* introduce variation in the activity durations. In this chapter, Monte Carlo simulations are used. The process of applying Monte Carlo simulations to a project management environment is as follows. First of all, a probability distribution is constructed which controls the degree and probability of the variability in activity durations. In this research, the generalized beta distribution is used. This distribution has been implemented in academic (Vanhoucke (2011), Elshaer (2013), Colin and Vanhoucke (2014)) as well as practical environments (AbouRizk et al. (1994)). The probability density function of a generalized beta distributed random variable x can be expressed as follows:

$$f(x) = \frac{\Gamma(\theta_1 + \theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)(b-a)^{\theta_1+\theta_2-1}}(x-a)^{\theta_1-1}(b-x)^{\theta_2-1}, \quad x \in [a,b] \tag{5.1}$$

a and b refer to the lower and upper limits of the random variable respectively. $\Gamma(\cdot)$ refers to the gamma function and θ_1 and θ_2 are two shape parameters. In this chapter, the simulations are controlled using a, b and the mean μ and mode m of the distribution. The shape parameters θ_1 and θ_2 can be calculated from equations (5.2) and (5.3).

$$\theta_1 = -\frac{(b+a-2m)(a-\mu)}{(m-\mu)(a-b)} \tag{5.2}$$

$$\theta_2 = \frac{(b+a-2m)(b-\mu)}{(m-\mu)(a-b)} \tag{5.3}$$

The use of a, b, μ and m allows for a wide array of distributional shapes, which is ideally suited for simulations for which different project outcomes are desired.

A number of Monte Carlo runs are performed to calculate *sensitivity metrics*, which are required to calculate the predictions of the Elshaer (2013) forecasting methods. Six different sensitivity metrics were selected, consistent with the works of Elshaer (2013) and Vanhoucke (2010b), namely the Criticality Index (CI), the Significance Index (SI), the Schedule Sensitivity Index (SSI) and the Cruciality Index using Pearson's product moment (CRI_r), Spearman's rank correlation (CRI_ρ) and Kendall's τ rank correlation (CRI_τ).

5.3.2 Attributes

This section consists of two parts. In section 5.3.2.1, we specify which attributes are captured along the project's progress. Section 5.3.2.2 discusses the application of Principal Component Analysis as a pre-processing technique to reduce the amount of data while retaining as much information as possible.

5.3.2.1 Capturing attributes

The progress data resulting from the Monte Carlo simulations lead to calculating Earned Value Management measures. These give the project manager an indication of the health of the project and form the input measures for the different Artificial Intelligence methods. Inputs for AI methods are often referred to as attributes. The AI methods learn the relation between EVM measures (input) and forecasting values (output) and apply the learned relation to unseen data. The attributes are given in table 5.1. The performance metrics SPI, SPI(t) and CPI all make use of EVM's key metrics, namely PV, EV and AC. The time forecasting methods are denoted as the Estimated time At Completion (EAC(t)), where a subdivision is made according to the Planned Value (PV), Earned Duration (ED) or Earned Schedule (ES) method. AD and PD stand for the Actual Duration and Planned Duration respectively. The Budget At Completion (BAC) results from the baseline schedule and captures the total project expenditure if every activity is executed according to plan. The final forecasting method of table 5.1, EAC(t)$_{ES_{2\alpha}}$ merits further discussion. This method was proposed by Elshaer (2013), who suggested an adaptation of the PV and EV calculations taking sensitivity information into account.

$$PV'_{\alpha,t} = \sum_j \alpha_j PV_{j,t} \tag{5.4}$$

$$EV'_{\alpha} = \sum_j \alpha_j EV_{j,AT} \tag{5.5}$$

Equation (5.4) shows that the PV of sensitivity metric α at time point t is made up of the sum of the PV of all activities indexed by j. A similar reasoning applies to equation (5.5). Following equations (5.4) and (5.5), the calculation of the Earned Schedule and Schedule Performance Indicator changes as well. This is reflected in the table by adopting the notation ES$'$ and SPI(t)$'$ respectively. α represents the sensitivity metric ($\alpha \in \{$CI, SI, SSI, CRI$_r$, CRI$_\rho$, CRI$_\tau\}$).

Name	Attribute	Calculation
Schedule Performance Index	SPI	$\frac{EV}{PV}$
Schedule Performance Index (time)	SPI(t)	$\frac{ES}{AT}$
Cost Performance Index	CPI	$\frac{EV}{AC}$
Earned Schedule	ES	$t + \frac{EV-PV_t}{PV_{t+1}-PV_t}$
	EAC(t)$_{PV_1}$	$PD - \frac{(EV-PV)*PD}{BAC}$
Estimate At Completion (time) - Planned Value	EAC(t)$_{PV_2}$	$\frac{PD}{SPI}$
	EAC(t)$_{PV_3}$	$\frac{PD}{CPI*SPI}$
	EAC(t)$_{ED_1}$	$PD + AD*(1-SPI)$
Estimate At Completion (time) - Earned Duration	EAC(t)$_{ED_2}$	$\frac{PD}{SPI}$
	EAC(t)$_{ED_3}$	$\frac{PD}{SPI*CPI} + AD*(1-\frac{1}{CPI})$
	EAC(t)$_{ES_1}$	$AD + PD - ES$
Estimate At Completion (time) - Earned Schedule	EAC(t)$_{ES_2}$	$AD + \frac{PD-ES}{SPI(t)}$
	EAC(t)$_{ES_3}$	$\frac{PD-ES}{CPI*SPI(t)}$
Estimate At Completion (time) - Elshaer (2013)	EAC(t)$_{ES_{2\alpha}}$	$AD + \frac{PD-ES'}{SPI(t)'}$

Table 5.1: Overview of the EVM attributes

5.3.2.2 Data Pre-Processing

Ever since the inception of Principal Component Analysis by Pearson (1901), it gained popularity as a dimensionality reduction technique. PCA presumes that the observations can be projected on a new set of axes, removing data redundancy and system noise in the process. Principal components are linear combinations of the original variables. The principal components are the directions that comprise the new coordinate axes. In a second phase, the observations are plotted onto the new axes using the score matrix. We only provide a small summary of the working of PCA and refer the reader to Jolliffe (2005) for a more comprehensive overview. Suppose we have an $(L \times P)$ matrix $\mathbf{X}$ holding L observations of random variable $\mathbf{x}$ with P different attributes. The first principal component can then be formulated as a linear combination of the original variables explaining a maximum amount of variation, as follows:

$$\mathbf{t}_1 = \mathbf{x}\mathbf{p}_1 = \mathbf{x}_1 p_{11} + \mathbf{x}_2 p_{12} + ... + \mathbf{x}_P p_{1P} \tag{5.6}$$

Incidentally, $\mathbf{p}_1$ should be a unit vector. The loadings $\mathbf{p}_i$ determine the directions of the principal components and correspond with the coefficients of the original variables in equation (5.6). The scores $\mathbf{t}_i$ constitute the coordinates of the i^{th} observation on the new coordinate axes. In matrix notation, the matrix of the scores $\mathbf{T}$ is an $(L \times P)$ matrix, whereas the loadings matrix contains $(P \times P)$ elements. Principal Component decomposition can then be written as given in equation (5.7):

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T \tag{5.7}$$

Naturally, there would be little use for PCA if all P components were retained. Hence, a criterion needs to be put forward that selects a number of principal components that explain a reasonable amount of variation. It can be shown (Shlens (2005)) that the principal components of $\mathbf{X}$ are the eigenvectors of the covariance matrix of $\mathbf{X}$. Let λ_i denote the eigenvalue of the associated principal component i. If all P principal components are retained, the P eigenvalues explain all variation in the measurements. In this chapter, the minimum number of principal components P^* will be selected representing at least t percent of the explained variation. This can be formulated according to equation (5.8):

$$P^* = \arg\min_f \frac{\sum_{i=1}^{f} \lambda_i}{\sum_{i=1}^{P} \lambda_i} \geq t \tag{5.8}$$

In this equation, the numerator refers to the summed variation explained by the first f principal components, whereas the denominator represents the total amount of variation explained by the P eigenvalues.

In general, there are three methodological grounds for opting for a PCA. These are briefly outlined below. The reader is referred to Colin et al. (2015) for a more elaborate discussion.

- Data overload: the overload of data results from project control metrics that are captured periodically. Hence, as the project progresses, the amount of information at a project manager's disposal grows. By combining the information into a select number of Principal Components, this issue can be resolved.

- Collinearity: collinearity occurs when multiple variables are influenced by a common factor and is arguably a major cause for concern in project control. Since all EVM metrics and forecasting methods are derived from three key numbers (Planned Value, Earned Value and Actual Cost), it is reasonable to assume the presence of collinearity within EVM data.

- Noise: noise is defined as the presence of an unexplained source of variation in a sample (Colin et al. (2015)). Since Earned Value Management is a top-down method, noise may result from the translation of variation on the activity level to a higher level of the Work Breakdown Structure.

Phase	Static Phase	Dynamic Phase
Set	Training Set	Test Set
Goal	Find optimal parameters	Test performance on unseen data
Type of data	Historical data Simulations	Real-life execution
	Size of training, validation and test sets	
Phase	Set	#executions
Static	Training set	$train\% * nbrex$
	• Training set'	$\frac{k-1}{k} * (train\% * nbrex)$
	• Validation set	$(1 - \frac{k-1}{k}) * (train\% * nbrex)$
Dynamic	Test set	$(1 - train\%) * nbrex$

Table 5.2: Overview of the training, validation and test sets

In the context of this chapter, which makes use of simulations and progress data that is captured periodically, the L observations correspond with the total number of Monte Carlo executions. For ease of reference, this will be termed *nbrex* in the remainder of this chapter. For each time period, indexed by $rp = 1,...,R$, 19 attributes as given in table 5.1 are captured. Consequently, matrix $\mathbf{X}$ contains $nbrex = L$ observations and $P = rp * 19$ attributes for a given review period rp.

5.3.3 Training, validation and testing

Section 5.3.1 covered the creation of a dataset that is as diverse as possible. Section 5.3.2.2 then assured that only relevant information is retained by combining the attributes of table 5.1. The complete dataset is typically decomposed into three distinct sets, namely a training, validation and test set. An overview of these sets is given in the bottom third of figure 5.1 and table 5.2. In a first phase, the training and test sets are separated. *train%* of the total amount of executions, *nbrex*, of a project are chosen for the training set, while the remainder is included in the test set. The training set is subdivided into a smaller training set and a validation set. The importance of the validation set is closely linked with the observation that each AI method needs to be tuned with care. The smaller training set serves to learn the relation between inputs and outputs, while the validation set is used to gauge the performance. The goal of the validation set is to determine the best parameter combination for the AI method under study.

In this chapter, we opt for a combination of a grid search procedure and cross-validation to determine the best parameter settings. Among the variants of cross-validation, we opted for k-fold cross-validation. In k-fold cross-validation the data is partitioned into k equally sized folds. One of those k folds is used as a validation set while the remaining $k-1$ folds are used for training. This is shown at the bottom of figure 5.1 where k partitions into a smaller training set (shown in white) and a validation set (shown in grey) are made. Since $k-1$ folds are used for training, the small training set contains $\frac{k-1}{k}$ of the executions of the larger training set, as indicated in table 5.2. The remaining $1-\frac{k-1}{k}$ of the larger training set comprises the validation set. The prediction results are averaged across the folds and form a proxy for the AI method's performance on the test set. The results of the prediction are used to determine the optimal parameter combination. Obviously, a criterion to measure the performance needs to be put forward. A previously employed metric for assessing forecasting accuracy (Vanhoucke and Vandevoorde (2007), Wauters and Vanhoucke (2014b)) is the Mean Absolute Percentage Error (MAPE), calculated according to equation (5.9).

$$MAPE = \frac{1}{R}\sum_{rp=1}^{R}\frac{|RD - EAC(t)_{rp}|}{RD} * 100 \tag{5.9}$$

This equation calculates the MAPE as the sum across all review periods R, indexed by rp, of the percentage deviation between the Real Duration (RD) and the forecast value, $EAC(t)_{rp}$ at time point rp. Once the parameter combination that yields a minimum MAPE is found, the AI method is retrained on the larger training set and the learned relation is applied to the test set, yielding the true MAPE.

5.4 Computational Experiment

In this section, the various settings used for the computational experiment will be outlined. The three items of section 5.3, data generation, attributes and training, validation and testing, are revisited and made more concrete.

5.4.1 Data Generation

The data generation phase comprised three phases, namely the baseline data which revolved around constructing a baseline schedule, the progress data, in which Monte Carlo simulations were executed and generation of the sensitivity metrics to construct the Elshaer (2013) forecasts. The topology of the networks was varied using the SP indicator. For our experiment, 90 Activity on the Node (AoN) networks were generated. Randomly sampled activity costs and durations were assigned to the 30 activities of each AoN network. The SP indicator was varied from 0.1 to 0.9 in steps of 0.1. Hence, there are 10 projects for each level of the SP indicator. Generation of the networks was executed using the RanGen engine (Demeulemeester et al. (2003) and Vanhoucke et al. (2008)). The topological structure of the employed dataset has been utilized in previous EVM studies (e.g. Colin and Vanhoucke (2014) and Elshaer (2013)) and can be downloaded from `www.projectmanagement.ugent.be/evms.html`. The baseline duration of each activity was randomly drawn from the interval [200,300] and the costs were generated randomly from the interval [50,100]. We only used variable costs, entailing that a deviation in duration of one of the activities is completely reflected in a cost deviation. The subsequent assumption is that the activity costs express a monetary unit per unit of time. If an activity takes longer to complete, more man-hours are required and the expenses rise.

The progress data allows for deviations from the baseline schedule. This variation will be represented in the EVM measures which in turn will be used to construct estimates for the project's final duration. Four key numbers were used to characterize settings for the generalized beta distribution. a and b represent the lower and upper limit of the random variable where μ and m refer to the distribution's mean and mode, respectively. In order to assess the general performance of the AI forecasting methods, 3 scenarios are constructed. These scenarios represent situations in which the project finishes early (Real Duration (RD) $<$ Planned Duration (PD)), on time (RD $\approx$ PD) and late (RD $>$ PD), respectively. a, b, μ and m were chosen in such a way that the Coefficient of Variation (CV $= \frac{\sigma}{\mu}$) is equal to 0.4 for the three scenarios. In a separate sensitivity experiment, the effect of changing one of the scenario's parameters will be studied. The sensitivity experiment studies the following two situations:

- $\Delta\mu$: a change in the distribution's mean while keeping the standard deviation σ constant. This is done by changing m from 60% to 140% of the On Time scenario's mode in steps of 20%. Once the mode is known, μ is calculated while keeping σ equal to 0.4.

- $\Delta\sigma$: a change in the distribution's standard deviation while keeping the mean constant. The standard deviation is modified from 60% to 140% of the On Time scenario's σ in steps of 20%.

The settings for the various scenarios are given in table 5.3. Each project is executed 1,000 times according to one of the generalized beta distributions specified in table 5.3. From this set of executions, the sensitivity metrics can be computed. These are required

Scenario	a	b	m	μ	θ_1	θ_2	σ
			General Performance				
Early	0.1	2	0.5	0.6	2.93	8.22	0.24
Middle	0.2	4	0.82	1	2.94	11.03	0.40
Late	0.2	4	1.2	1.4	2.83	6.14	0.56
			Sensitivity				
Change in μ							
$\Delta\mu_1$	0.2	4	0.49	0.79	1.67	9.13	0.40
$\Delta\mu_2$	0.2	4	0.65	0.89	2.24	10.13	0.40
$\Delta\mu_3$	0.2	4	0.98	1.12	3.80	11.87	0.40
$\Delta\mu_4$	0.2	4	1.14	1.25	4.75	12.35	0.40
Change in σ							
$\Delta\sigma_1$	0.2	4	0.94	1	8.53	31.99	0.24
$\Delta\sigma_2$	0.2	4	0.89	1	4.72	17.70	0.32
$\Delta\sigma_3$	0.2	4	0.70	1	1.98	7.44	0.48
$\Delta\sigma_1$	0.2	4	0.53	1	1.40	5.26	0.56

Table 5.3: Generalized beta settings for the various scenarios

to calculate PV'$_{\alpha,t}$ and EV'$_{\alpha}$ which are prerequisites to construct the forecasting methods of Elshaer (2013).

5.4.2 Attributes

Similar to section 5.3.2, a distinction is made between capturing the attributes (section 5.4.2.1) and pre-processing them using Principal Component Analysis (section 5.4.2.2).

5.4.2.1 Capturing attributes

The attributes from which the AI methods learn the relationship between inputs and outputs were already provided in table 5.1. These are captured for every 10% complete, ranging from 10% to 90%. Incidentally, R in equation (5.9) is equal to 9.

5.4.2.2 Data Pre-processing

Table 5.1 contains the 19 attributes (4 performance indicators, 9 EVM forecasting methods and 6 Elshaer forecasting methods) that are captured for every 10% complete. At the 90% completion point, there are 171 (9 × 19) attributes. In order to reduce this amount to the combination of attributes that explains a maximum amount of variation, Principal Component Analysis is used. In section 5.3.2.2, we explained that the minimum number of principal components are selected such that at least t percent of the variation is explained. This was detailed in equation (5.8). In our tests, 4 levels for t were examined, i.e. $t \in \{0.5, 0.9, 0.95, 0.99\}$.

5.4.3 Training, validation and testing

One of the principal differences compared to the work of Wauters and Vanhoucke (2014b) lies in the division between training, validation and test sets. In this chapter, the parameters are tuned on the project level whereas the other work tunes parameters on a more aggregated level. Every project is executed 1,000 times ($nbrex = 1,000$) after which the forecasting procedure can be initiated. These executions are then divided into a training set and a test set. The training set is further subdivided into a smaller training set and a validation set. The goal of this disaggregation is to find the optimal parameters for the AI method that is utilized. For each of the 10 projects per level of the SP factor, the optimal parameters are determined. Once the optimal parameters have been found, the AI model is retrained and tested on unseen data, which is contained in the test set. Of the 1,000 executions of each project, 80% (800 executions) is used for training and 20% for testing (200 executions). Consequently, $train\%$ of section 5.3.3 is equal to 80%. The smaller training set contains 80% of the larger training set's executions, totalling 640 executions (0.8 * 800). The validation set comprises the remaining 160 executions (0.2 * 800).

As previously mentioned, the smaller training set and the validation set serve the purpose of finding the best parameters of the AI method. This was done using k-fold cross-validation. For this chapter, 5 (k=5) folds were implemented, implying that each time 640 executions are used for training and 160 for testing. Since every run is used once for validation, cross-validation counters the effect of overfitting. An overview of the parameters and their settings is provided in table 5.4. For some methods, the grid search was refined in order to identify the point from which the validation error began to rise again. In table 5.4, a sequence is indicated using the following notation: ($lb - ub$, Δ=$increment$), in which lb specifies the lower bound of the sequence, ub specifies the upper bound and Δ represents the step size. The parameter settings are briefly explained along the following lines:

- Decision Tree, Bagging, Random Forest: the tree-based approaches rely on 2 or 3 parameters. For a single decision tree, the confidence interval determines the p-value for splitting. As discussed in section 5.2.1.1, a test of independence is conducted between the response variable and the predictors. The split and bucket determine the weights in a node or terminal node respectively and govern the process of splitting, resulting into an additional level in the decision tree's structure. Bagging and random forest approaches are ensemble methods that draw results from multiple trees. The number of trees that is grown is one of the parameters. While bagging takes all of the predictors into account, random forests sample a number of input variables. As a result, the number of variables that are used is an additional parameter for this AI technique.

- Boosting: boosting employs regression trees by adding them to the model one at a time. The learning rate or shrinkage parameter determines the contribution of each tree to the model. A slower learning rate has the advantage that the parameters leading to an optimal performance are not skipped by accident. On the other hand, convergence to an optimum solution is slow, requiring a low learning rate to go hand in hand with a higher number of trees. The interaction parameter is in charge of the complexity of the tree, leading to additional nodes in the tree as the number of interactions between variables increases.

- Support Vector Machine: Support Vector Machines, like other AI techniques, try to find an optimal balance between learning the relation between inputs and outputs while maintaining a good generalization error, which is reflected in the performance on the test set. An improved generalization may be obtained at the expense of additional training errors, primarily controlled by the parameter C. Other parameters depend on the kernel choice. In this chapter, the Radial Basis Function (RBF) kernel was used, for similar reasons to those of Wauters and Vanhoucke (2014b). The RBF depends on 1 parameter, namely γ.

Method	Parameter	Explanation	Settings
Decision Tree	Confidence Interval	Determines the p-value for the hypothesis tests	0.5, 0.9, 0.95, 0.99
	Split	Minimum sum of weights in a node before splitting is considered	20
	Bucket	Minimum sum of weights in a terminal node	(1-19, Δ=3), 20, 50, 100, 500, 1000, 5000, 10000
Bagging	Split	Minimum sum of weights in a node before splitting is considered	5, (10,100,Δ=10), 100, 500
	#Trees	#trees to grow	5, 10, 50, 100, 500
Random Forest	Split	Minimum sum of weights in a node before splitting is considered	5, (10,100,Δ=10), 100, 500
	#Trees	#trees to grow	5, 10, 50, 100, 500
	#Input variables	#randomly sampled input variables to use	$\sqrt{\#Cols}$, (2,10,Δ=1)
Boosting	Shrinkage	Determines the impact of each additional tree	0.001, 0.005, 0.01, 0.05, 0.1
	#Trees	#trees to grow	10, 100, 1000, 5000, (10000,50000,Δ=10000)
	Interaction	Allows k-way interaction between variables	1, 2, 4, 8
Support Vector Machine	C	Controls the trade-off between regularization and training accuracy	$2^{(-5,15,\Delta=2)}$
	γ	Kernel-specific parameter	$2^{(-30,15,\Delta=5)}$

Table 5.4: Parameter settings of the AI methods

5.5 Results

The performance of the Artificial Intelligence methods will be compared to the EVM and Elshaer forecasting methods in this section. The AI techniques were implemented in R (R Core Team (2013)) for which the following packages were used: party (Hothorn et al. (2006a,b)) for decision trees, bagging and random forests, gbm (Ridgeway (2013)) for boosting and LIBSVM (Meyer et al. (2012)) for Support Vector Machines. The computational experiments were conducted on Ghent University's High Performance computing infrastructure. We made use of the Delcatty cluster, which boasts 64GB RAM and has a quad-core Intel Xeon processor with 2.6 GHz at its disposal. As mentioned in section 5.3.3, the criterion for assessing forecasting accuracy is the MAPE. Unless specifically stated otherwise, the MAPE will be reported, indicating that the performance of a method is better on average, rather than outperforming a method at a certain point in time, as would be the case for the Absolute Percentage Error.

Tuning Artificial Intelligence methods with care is an integral part of deploying these methods for prediction purposes. In section 5.5.1, the best found parameters are reported. Additionally, we report on the percentage of explained variation for data preprocessing using principal components. Section 5.5.2 examines the results of the AI methods in relation to the other forecasting methods for the Early, On Time and Late scenarios specified in table 5.3. The impact of the topological structure, measured by the SP indicator, and the percentage complete is discussed as well. Section 5.5.3 concludes the results by investigating the robustness of the AI techniques in a sensitivity experiment. The impact of a change in the distribution's mean and standard deviation illustrates the limitations of turning to Artificial Intelligence methods.

5.5.1 Parameter fine-tuning

5.5.1.1 AI Methods

One of the key issues of employing Artificial Intelligence methods lies in tuning their parameters. The parameter settings, along with a concise explanation, were presented in table 5.4. The goal of the smaller training set and the validation set consists of finding the combination of parameters that yields the lowest MAPE. We remark that because we follow a project-based approach, parameter settings are optimized per project, rather than on a more aggregated level. Table 5.5 displays the results for the various AI techniques. The column labeled "Share" shows the percentage when a parameter's value yielded the best results and is aggregated across all scenarios, projects and values of the

Method	Parameter	Value	Share
Decision Tree	Confidence Interval	0.5	3.70%
		0.9	41.11%
		0.95	37.78%
		0.99	17.41%
	Split	20	100%
	Bucket	1	5.19%
		4	2.96%
		7	3.33%
		10	6.67%
		13	8.52%
		16	15.93%
		19	14.07%
		20	14.81%
		50	22.59%
		100	5.93%
Bagging	Split	5	42.59%
		10	50.74%
		50	6.67%
	#Trees	50	0.37%
		100	4.81%
		500	94.84%
Random Forest	Split	5	7.78%
		10	78.89%
		20	8.15%
		30	3.70%
		40	1.11%
		50	0.37%
	#Trees	100	2.96%
		500	97.04%
	#Try	1	11.48%
		3	6.67%
		4	1.48%
		5	78.15%
		6	2.22%
Boosting	Shrinkage	0.005	5.93%
		0.001	87.04%
		0.01	7.04%
	#Trees	1000	9.63%
		5000	9.63%
		10000	33.33%
		20000	47.41%
	Interaction	1	5.93%
		2	38.15%
		4	55.93%
Support Vector Machine	Cost	2^3	19.26%
		2^5	14.81%
		2^7	2.22%
		2^9	18.52%
		2^{11}	4.07%
		2^{13}	25.19%
		2^{15}	15.93%
	γ	2^{-20}	27.78%
		2^{-15}	38.15%
		2^{-10}	34.07%

Table 5.5: Overview of the best parameter settings of the AI techniques

t	DT	Bag	RF	Boost	SVM	Average
0.5	7.44%	7.12%	7.15%	7.00%	7.13%	7.17%
0.9	7.47%	7.00%	7.04%	6.88%	6.75%	**7.03%**
0.95	7.48%	6.99%	7.05%	6.90%	6.76%	7.04%
0.99	7.48%	7.05%	7.04%	6.91%	6.83%	7.06%

Table 5.6: MAPE for different values of t

SP indicator. If one or more of the settings of table 5.4 are missing in table 5.5 it means that that parameter value did not occur as one of the optimal settings.

5.5.1.2 Principal Components

In section 5.4.2.2, it was mentioned that the minimal number of principal components needed to explain t percent of the variation was retained. Four levels of t were proposed, namely 50%, 90%, 95% and 99%. Obviously, as t increases, the number of principal components rises as well. Consequently, a trade-off needs to be made between the desired amount of explained variation and the resulting number of principal components. As the number of principal components rises, noise (cf. section 5.3.2.2) is re-introduced (Jolliffe (2005)). Figure 5.2 depicts the relation between the percentage complete (on the x-axis) and the number of principal components (on the y-axis) for the different levels of t. We can infer that the difference in number of principal components between t=0.95 and t=0.99 is quite steep. The difference in performance between the different levels of t for the AI methods is presented in table 5.6. The lowest average MAPE is indicated in bold in the table. In general, the average MAPE across the levels of t is not that large. Following figure 5.2 and table 5.6, a trade-off needs to be made between the number of principal components and the forecasting accuracy. A value of 0.9 for t strikes the best balance between these criteria. In the remainder of this section, results are reported for this level of explained variation.

5.5.2 General performance

In this section, the performance of the Artificial Intelligence methods is compared to that of the EVM and Elshaer forecasting methods. Performance is split up along the Early, On Time and Late scenarios, for which the settings were provided in table 5.3. The reader is reminded that these scenarios differ in the average outcome but have an equal value for the coefficient of variation (CV=0.4). In this section, we examine the mean MAPE and its standard deviation, the effect of the network's topology and the impact

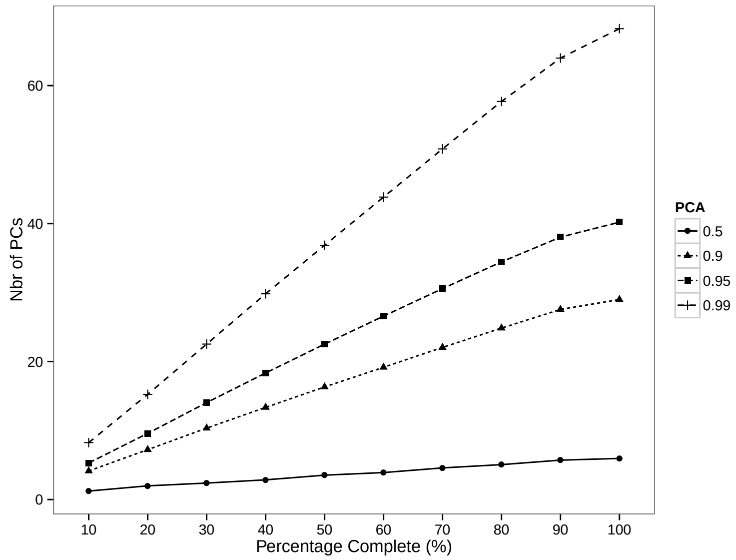


Figure 5.2: Relation between the percentage complete and number of principal components for various levels of the explained variation

of the percentage complete. All results are given in table 5.7. For the Planned Value, Earned Duration, Earned Schedule and Elshaer forecasting methods, the forecasting accuracy of the best performing method is presented along with its performance factor or sensitivity index, respectively.

General accuracy The general accuracy is captured using the mean MAPE across all levels of the SP indicator. Along with the mean value, the standard deviation provides an indication of the variability of the forecasting accuracy.

- The first observation is that there is very little difference in performance between the Early and Late scenarios for the EVM and Elshaer forecasting methods. An improved forecasting accuracy can be noted for the On Time scenario, where those methods with a performance factor of 1 perform best. In general, a performance factor indicates the expected future behaviour of the project's performance. A performance factor of 1 assumes that the remainder of the project will perform as planned, which aligns with reality for the On Time scenario. Since the AI methods do not employ a performance indicator, their forecasting accuracy is very similar across the three scenarios.
- All of the AI methods predict the final duration of the project more accurately

than the best performing EVM and Elshaer methods. The ensemble methods which combine multiple decision trees attain better results than the decision tree method. The difference in forecasting accuracy between bagging and random forests is negligible for all scenarios. Boosting and SVM perform even better than the decision tree-based approaches.

- Apart from the great average performance of the AI methods, the standard deviation is also lower compared to the EVM and Elshaer forecasting methods. The Elshaer method with the SI as its sensitivity index yields the lowest standard deviation among all EVM/ES methods. However, the standard deviation of the AI methods is about 44% lower ($\frac{3.60-2}{3.60}$). Because of the improvement in forecasting accuracy of the EVM and Elshaer methods, the difference in performance with the AI methods is smallest for the On Time scenario.

Impact of the SP indicator The topology of the AoN networks is dictated by the SP indicator. Vanhoucke (2011) established that forecasting performance of EVM methods improves as the project becomes more serial (larger value of the SP indicator). As a project becomes more serial, its performance coincides with that of one of the few activities in progress. This is contrary to parallel projects where multiple activities are in progress at the same point in time. Since Earned Value Management is a top-down technique, the poor performance of critical activities may be masked by the non-critical activities that are ahead of schedule, leading to false warning signals. As a project becomes more serial, the findings of CPM and EVM converge. In order to assess the relation between the SP indicator and forecasting accuracy, we make use of the following formula:

$$\Delta_{MAPE_{SP}} = \frac{MAPE_{SP=0.1} - MAPE_{SP=0.9}}{MAPE_{SP=0.1}} * 100 \tag{5.10}$$

The formula of equation (5.10) returns the relative percentage improvement in MAPE between projects with an SP value of 0.1 and 0.9. A visual inspection preceded the construction of this equation to ensure that no jumps in performance for intermediate levels of the SP indicator occur. Table 5.7 reveals that the relation between forecasting accuracy and the SP indicator holds for all methods and across all scenarios. Hence, we corroborate the findings of Vanhoucke (2011) and add that an identical relation holds for the Artificial Intelligence methods. The difference in performance is steepest for the On Time scenario where forecasting predictions for more serial projects are approximately 70% more accurate than those for more parallel projects. While the performance of the AI methods increases by more than 50%, the performance improvement is slightly less than that of the Earned Schedule and Elshaer methods.

Impact of the Percentage Complete A measurement of the attributes of table 5.1 is made every 10% complete ranging from 10% complete to 90% complete. At those points in time, the information from the attributes is used to construct a new estimate for the project's final duration. In general, it is expected that as a project progresses and more information becomes available, a more accurate estimate of its duration can be made. Similar to equation (5.10), a metric was defined that captures the relative percentage improvement between the 10% completion point and the 90% completion point, as follows:

$$\Delta_{MAPE_{PC}} = \frac{MAPE_{PC=10\%} - MAPE_{PC=90\%}}{MAPE_{PC=10\%}} * 100 \tag{5.11}$$

The findings of the the percentage complete can be summarized as follows:

- Table 5.7 shows that the forecasting accuracy improves as the project progresses.
- For the traditional EVM methods, the improvement is lowest for the On Time scenario, where the methods with a performance factor of 1 yield the best results. This finding is sensible since those methods assume the project progresses according to plan, regardless of the indications of the progress data.
- While the estimates of the Artificial Intelligence methods improve as the project proceeds, the improvement is less steep compared to the EVM and Elshaer methods. This implies that while the overall performance of the AI methods is better on average, the gap with the EVM and Elshaer methods shrinks along the percentage complete. This is a significant contribution since the EVM/ES forecasting methods do not perform well for the early and mid-stages of a project (Vanhoucke (2010a)). While the gap in performance decreases along the percentage complete, the AI methods still come out on top.

Scenario	Criterion	Forecasting Method								
		EVM				AI				
		PV	ED	ES	Elshaer	DT	Bag	RF	Boost	SVM
Early	PF	SPI	SPI	SPI(t)	SI					
	μ_{MAPE}	15.90%	15.90%	11.23%	11.38%	7.47%	7.07%	7.06%	6.90%	6.77%
	σ_{MAPE}	4.74	4.74	3.65	3.60	2.09	2.01	2.01	2.02	1.94
	$\Delta_{MAPE_{SP}}$	43.86%	43.86%	62.70%	62.38%	51.66%	52.57%	52.42%	53.99%	53.53%
	$\Delta_{MAPE_{PC}}$	29.16%	29.16%	44.44%	46.44%	11.94%	17.40%	17.33%	18.21%	18.80%
On Time	PF	1	1	1	SI					
	μ_{MAPE}	8.00%	8.03%	7.72%	11.36%	7.51%	7.08%	7.07%	6.91%	6.78%
	σ_{MAPE}	3.70	3.68	3.62	3.60	2.15	2.06	2.05	2.07	1.98
	$\Delta_{MAPE_{SP}}$	71.99%	71.83%	71.80%	62.48%	52.92%	53.87%	53.55%	55.15%	54.55%
	$\Delta_{MAPE_{PC}}$	15.76%	15.41%	18.43%	45.97%	11.50%	16.79%	16.70%	17.81%	18.75%
Late	PF	SPI	SPI	SPI(t)	SI					
	μ_{MAPE}	15.18%	13.31%	11.14%	11.35%	7.43%	7.00%	6.99%	6.84%	6.71%
	σ_{MAPE}	3.98	4.57	3.52	3.57	2.05	1.97	1.96	1.97	1.90
	$\Delta_{MAPE_{SP}}$	39.44%	55.88%	58.88%	62.17%	50.14%	51.48%	51.18%	52.85%	52.34%
	$\Delta_{MAPE_{PC}}$	26.09%	35.20%	44.61%	49.96%	12.07%	17.64%	17.52%	18.47%	19.10%

Table 5.7: General performance across the Early, On Time and Late scenarios

5.5.3 Sensitivity Analysis

In this section, a search for the limitations of the AI methods is conducted. The previous section assumed that the variability could be estimated accurately. The parameters of the distribution for the training and test sets were equal. In real-life situations, appraising this variation in a correct manner proves to be a difficult task. Hence, it is possible that the training and test sets do not coincide or that they are similar in nature but not identical. To that end, the mean and standard deviation of the distributions that govern the variability of the activity durations were changed in this section. The settings for these additional scenarios were discussed in section 5.4.1. For this robustness experiment, the dataset was limited to those projects with a value of 0.5 for the SP indicator and the On Time scenario as the training set. The results are shown in table 5.8. The same notation as in table 5.3 was used to indicate the various scenarios. The main conclusions of table 5.8 can be summarized along the following lines. These conclusions hold for both sensitivity settings (a change in the mean and a change in the standard deviation) unless specified otherwise.

- The SVM method performed admirably when the training and test sets coincide. When these sets do not align, the MAPE quickly skyrockets, reaching average forecast errors of more than 60%.
- Both the EVM methods and the Artificial Intelligence techniques suffer more from a change in mean compared to a change in standard deviation. The MAPE of all methods is higher for scenarios $\Delta\mu_{1-4}$ than for $\Delta\sigma_{1-4}$.
- The discrepancy between the AI methods and the EVM/ES methods diminishes as the mean of the generalized beta distribution increases. For scenarios $\Delta\mu_1$ and $\Delta\mu_2$, there is a substantial difference between the AI methods and the best performing PV, ED, ES and Elshaer method. However, this difference becomes marginal for the scenarios where $\mu > 1$.
- As the variation captured by a change in standard deviation increases, the accuracy drops for all methods. Scenarios $\Delta\sigma_{1-4}$ demonstrate that as the standard deviation becomes larger, the forecast accuracy decreases. This conclusion holds for the EVM/ES methods and the AI techniques.

The mean and standard distribution were modified to a large extent. We also investigate how the performance of the AI methods is impacted by smaller changes, in which the training and test sets are similar but not identical. Wauters and Vanhoucke (2014b) constructed a symmetric, random and uniform class of distributions to examine situations

Scenario	PV	ED	ES	Elshaer	DT	Bag	RF	Boost	SVM
Δ_{μ_1}	15.37%	15.37%	13.53%	13.91%	28.85%	33.38%	33.29%	28.66%	94.43%
Δ_{μ_2}	10.16%	11.01%	10.08%	12.77%	18.73%	21.23%	21.17%	18.83%	86.36%
Δ_{μ_3}	8.12%	8.21%	7.69%	9.89%	9.41%	10.06%	10.05%	8.64%	63.13%
Δ_{μ_4}	10.92%	10.30%	8.62%	8.91%	11.66%	14.10%	14.10%	10.38%	59.13%
Δ_{σ_1}	4.19%	4.21%	3.98%	6.91%	9.45%	9.95%	9.92%	9.63%	69.44%
Δ_{σ_2}	5.48%	5.52%	5.23%	9.05%	10.34%	10.87%	10.84%	10.36%	69.77%
Δ_{σ_3}	8.11%	8.18%	7.80%	13.39%	12.40%	13.08%	13.05%	12.17%	70.68%
Δ_{σ_4}	9.35%	9.42%	9.00%	15.43%	13.45%	14.19%	14.16%	13.08%	71.00%

Table 5.8: Robustness results for the AI methods: training set $\neq$ test set

Scenario	PV	ED	ES	Elshaer	DT	Bag	RF	Boost	SVM
R(E-OT-L)	7.99%	8.06%	7.60%	14.45%	7.32%	6.82%	6.84%	6.70%	6.51%
R(Δ_μ)	6.93%	6.97%	6.59%	11.99%	6.80%	6.38%	6.40%	6.21%	6.06%
R(Δ_σ)	6.75%	6.74%	6.42%	13.84%	6.94%	6.51%	6.49%	6.32%	6.14%
R(All)	7.34%	7.24%	6.98%	13.70%	7.21%	6.83%	6.83%	6.71%	6.55%

Table 5.9: Robustness results for the AI methods: training set $\approx$ test set

in which the training and test sets are similar. In this chapter, we focus on the class of random distributions. The distribution that specifies the activity duration is chosen randomly from a set of scenarios. A random number is drawn for each activity, after which the distribution belonging to the random number is applied to that activity. In this manner, 4 scenarios were constructed. A combination of the Early, On Time and Late scenarios was used, one in which the 4 scenarios of a change in the mean were used, one for a mix of a change in the standard deviation and finally, a combination of all 11 scenarios. The notation for these scenarios is R(E-OT-L), R(Δ_μ), R(Δ_σ) and R(All) respectively. The settings of the individual scenarios were provided in table 5.3. The results are shown in table 5.9. When different distributions for the activities are drawn, the AI methods outperform the EVM/ES methods, regardless of the scenario. Consequently, the results of Wauters and Vanhoucke (2014b) with regard to the similarity between training and test sets can be corroborated.

5.6 Conclusion

In this chapter, four contributions to the existing body of project control literature were made. First and foremost, the forecasting performance of 5 different Artificial Intelligence methods was benchmarked against the best performing EVM and ES methods. To the best of our knowledge, this chapter is the first work to introduce these techniques in a project control context that makes use of Earned Value Management metrics. Secondly, the methods were embedded in a methodology that was comprised of 4 parts. The data generation phase involved generating topologically diverse project networks and constructing the baseline schedule. This schedule served as a point of reference for the progress data, for which generalized beta distributions were employed. The statistical distributions introduce variability on the activity level, which is captured by the EVM attributes. The attributes constitute the inputs for the various AI methods. In order to restrict the amount of information and the computational burden, the data is pre-processed using Principal Components. The training and validation sets serve the purpose of finding the optimal parameters for each AI technique. These are tuned with a combination of a grid search and cross-validation procedure. Thirdly, we examined the performance of the AI methods when the training and test sets coincide. In this situation, it was shown that all AI methods outperform the best performing Planned Value, Earned Duration, Earned Schedule and Elshaer methods. Both the mean and standard deviation of the Mean Absolute Percentage Error were considerably lower than that of the EVM/ES methods. Additionally, there was a substantial difference in performance for the early and middle stages of the project progress. The AI methods proved to outperform the current EVM/ES methods. This contribution is a significant improvement since the early and middle stages were previously characterized by large prediction errors. Researching the sensitivity of the AI methods to varying levels of the mean and standard deviation of the activity duration distributions is the final contribution of this chapter. The results revealed that all methods are more sensitive to a change in the mean than to a change in standard deviation. The great performance of the Support Vector Machines in the main experiment needs to be weighed against the steep drop in performance when the inputs of the underlying distribution change. While the performance of the AI methods is not as detrimental as to prohibit their implementation, it shows that their performance is dependent on the correct appraisal of the variability affecting the activities. This reliance on either historical data, expert judgement or statistical distributions is the biggest asset and liability of this type of methods.

Two distinct research avenues are identified. First of all, we relied on simulation-based executions of the project networks. While this allows us to draw conclusions on the progress of an extremely wide spectrum of projects, the Artificial Intelligence techniques have yet to pass the test of empirical validation. This can be done by applying the proposed methodology to real-life projects and using expert judgement or historical data as proxies for the generalized beta distributions. An alternative would be to include sector- or project-specific attributes as inputs for the learning techniques. Secondly, this chapter studied the prediction problem setting. A different application area for Artificial Intelligence lies in classification problems. Searching for and solving classification problems in a project control context forms a viable challenge for academics.

5.A Appendix

5.A.1 R template for AI forecasting on a sample project

In this appendix, a description of a working template in R is provided. The template in R can be downloaded from the `www.projectmanagement.ugent.be` research page. The goal of this appendix is to show how the methodology of training and validation (the *static* phase) and testing (the *dynamic* phase) can be applied to a sample project using the Artificial Intelligence techniques of this chapter. The R template allows the user to provide parameters for the training and validation phase. Next, it selects the optimal parameters using a fivefold cross-validation procedure and then applies the optimal parameters to the test set. It is worth noting that the findings based on this sample project may differ from those in the presented chapter. The results in the chapter are averaged across multiple projects. The outline of this appendix is as follows. First, the required inputs for the template are listed. Next, a short description of the functions and their relation to the chapter's methodology is given. Finally, the pseudocode of the static and dynamic phase is included.

5.A.1.1 Required inputs

From the research page available at `www.projectmanagement.ugent.be`, the following files should be downloaded and placed in the working directory of R:

- OutputProject1.txt.bz2 - this zipped file contains the periodic measurements of the EVM attributes reported in table 5.1. Each row corresponds with 1 execution, resulting in a total of 1,000 executions. P2 Engine (Vanhoucke (2014)) was employed to generate this file.

- 5Folds.txt - this file contains the executions that make a division between the training and test sets. The 1,000 executions found in OutputProject1.txt.bz2 are partitioned into 800 executions (*training set*) and 200 executions (*test set*).

- 5Folds-Validation.txt - this file contains the executions that make a division between the smaller training set and the validation set. The 800 executions found in the training set are partitioned into 640 executions (*training set'*) and 160 executions (*validation set*).

- The packages party, gbm and e1071 need to be installed if one wishes to test all AI methods.

Outline

- loadData - this function loads the required libraries, as well as global variables. **Important**: please ensure that the path of the working directory is changed to the directory in which the required input files reside.

- prepareData - this function reads the input files.

- constructForecast - this is one of the main methods and takes as arguments a dataframe with the periodic data, a string that indicates which AI method will be used, a parameter vector and a boolean called folds to activate cross-validation. constructForecast() makes use of several auxiliary methods described below:

  - executePCA - applies principal component analysis to a dataframe labeled somedf. If sometestdf is provided, the same principal components as for somedf are utilized.
  - constructPrediction - this method requires a training set, test set, AI method and parameter vector. The parameters of the AI method are applied to the training phase, after which the model is run on the test set. This method returns the forecasted values for each execution of the test set.

5.A.1.2 Pseudocode

The pseudocode of the static and dynamic phases is given in algorithms 2 and 3 respectively. The input files and a user-defined list of parameters Par constitute the input for the training and validation processes. For each element of Par, k-fold cross-validation is applied. Principal Component Analysis is executed throughout all review periods $rp = 1, ..., R$, after which the final project duration is forecast with the parameters $par \in Par$. The prediction error is averaged across the executions and folds. Finally, the optimal parameter vector is defined as the vector that yields the minimum MAPE across all folds and review periods.

The dynamic phase requires the output of the static phase as one of the inputs. The optimal parameters par^* are applied for predicting the final project duration RD. The structure of the pseudocode is similar to that of algorithm 2, with the exception that no cross-validation takes place.

Algorithm 2: Pseudo-code for the *static* phase

Data: The files described in *Required inputs* and a parameter vector Par

Result: The optimal parameter vector par^*

Divide the $nbrex$ executions into a *trainset* and *testset*

for $par \in Par$ **do**

  while $fold \leq k$ **do**

    Update $trainset$ & $valset$ with executions of $fold$

    for $rp \leftarrow 1$ **to** R **do**

      Update $trainset$ & $valset$: select the first $rp * 19$ columns `// 19 = #attributes of table 5.1`

      Execute PCA

      Forecast duration using AI method with parameters in par

        Training data $= trainset$

        Test data $= valset$

      for $e \leftarrow 1$ **to** $(1 - \frac{k-1}{k}) * (train\% * nbrex)$ **do**

        $\text{APE}_{e,rp,fold} = \frac{|RD - \hat{RD}|}{RD} * 100$

        `//` $\hat{RD}$ `is the prediction of` RD `resulting from the applied AI method`

      end

      $\text{APE}_{rp,fold} = \frac{1}{(1-\frac{k-1}{k})*(train\%*nbrex)} \sum_{e=1}^{(1-\frac{k-1}{k})*(train\%*nbrex)} APE_{e,rp,fold}$

    end

    $\text{MAPE}_{fold} = \frac{1}{R} \sum_{rp=1}^{R} APE_{rp,fold}$

  end

  $\text{MAPE}_{par} = \frac{1}{k} \sum_{fold=1}^{k} MAPE_{fold}$

end

$par^* = \arg\min_{par} MAPE_{par}$

Algorithm 3: Pseudo-code for the *dynamic* phase

Data: The files described in *Required inputs* and the optimal parameters par^*

Result: The forecasting accuracy of the AI method

for $rp \leftarrow 1$ **to** R **do**

  Update *trainset* & *valset*: select the first $rp * 19$ columns `// 19 = #attributes of table 5.1`

  Execute PCA

  Forecast duration using AI method with parameters in par^*

   Training data = *trainset*

   Test data = *testset*

  for $e \leftarrow 1$ **to** $(1 - train\%) * nbrex$ **do**

   $\text{APE}_{e,rp} = \frac{|RD - \hat{RD}|}{RD} * 100$

   `//` $\hat{RD}$ `is the prediction of` RD `resulting from the applied AI method`

  end

  $\text{APE}_{rp} = \frac{1}{(1-train\%)*nbrex} \sum_{e=1}^{(1-train\%)*nbrex} APE_{e,rp}$

end

$\text{MAPE} = \frac{1}{R} \sum_{rp=1}^{R} APE_{rp}$

6

A Nearest Neighbour extension to Earned Value Management forecasting with Artificial Intelligence

In this chapter, we provide a Nearest Neighbour-based extension for project control forecasting with Earned Value Management. The k-Nearest Neighbour method is employed as a predictor and to reduce the size of a training set containing more similar observations. An Artificial Intelligence (AI) method then makes use of the reduced training set to learn the relation between project control data and the real duration of a project. Additionally, we report on the forecasting stability of the various AI methods and their hybrid Nearest Neighbour counterparts.
A large computer experiment is set up to assess the forecasting accuracy and stability of the existing and newly proposed methods. The added value of the Nearest Neighbour method as a predictor and as a hybrid method in conjunction with an AI method is identified. A sensitivity analysis in which the amount of observations of the training set and the amount of neighbours are varied provides additional insights.

6.1 Introduction

Operations research is a branch devoted to solving complex problems to (near-)optimality by applying mathematical modeling, statistics and algorithms. Project and production scheduling are among its most widely researched subdivisions (Tavares (2002)). Research related to project scheduling was fueled by the inception of the Critical Path Method (CPM, Kelley (1961); Kelley and Walker (1959)) and the Program Evaluation and Review Technique (PERT, Fazar (1959)). Both methods have become straightforward standards for the construction of a baseline schedule and a rudimentary assessment of the relation between project duration and activity variability. PERT as well as other schedule risk analysis methods nuance the view of the CPM in which a binary view of criticality is presented. CPM stipulates that an activity is either critical or not, while PERT allocates a probability of being critical to every activity. While baseline scheduling and risk analysis are crucial components of the preparatory phase of a project's lifecycle, they ultimately serve the purpose of acting as a point of reference for the project control phase. In this phase, the project is being executed and the project's progress is compared to and contrasted with the plan. When the progress deviates too much from the plan, the project manager may decide to take corrective actions to bring the project back on track. Baseline scheduling, risk analysis and project control as well as their interrelationships are the main constituents of *dynamic scheduling* (Uyttewael (2005) and Vanhoucke (2012b, 2014)).

The focus of this chapter lies on the control dimension of dynamic scheduling. The methodology that will be utilized in this chapter is Earned Value Management (EVM), which was conceived in the 1960s by the American Department of Defense. EVM tracks the progress of a project on an aggregated Work Breakdown Structure level. Even though this has inspired criticism (see e.g. Book (2006a,b), Jacob and Kane (2004)), Vanhoucke (2010a) argues that controlling a project on the activity level is simply not feasible for many moderately sized projects. EVM measures a project's progress by means of three key metrics, namely Planned Value (PV), Earned Value (EV) and Actual Cost (AC). For an overview of the essentials of EVM, we refer to Fleming and Koppelman (2005). The paper of Lipke (2003) heralded a turning point for the research community. While research efforts focused on the cost objective, the inception of the Earned Schedule (ES) metric enabled time monitoring.
The ultimate goal of project control is to safeguard the project's final duration and budget. This can be achieved by finding mechanisms that serve as triggers for corrective

action. Bowman (2006) specified control limits on the activity level while Colin and Vanhoucke (2014) defined a state of control on a more aggregated level of the Work Breakdown Structure. Raz and Erel (2000) examined how the timing of points at which the project's performance will be assessed can be optimized. A more exhaustive overview of project monitoring and control models, approaches and decision support systems can be found in Hazır (2015). In the remainder of this section, we will focus on project control forecasting.

A trigger for corrective action can also be initiated by means of estimating the final project duration or cost. The project manager wishes to get a realistic estimate of the project's final duration or budget based on progress data and historical data or simulations. Realistic estimates are provided by forecasting methods, which have received considerable attention from the project control research community. Two aspects provide an indication of a forecasting method's qualities, namely accuracy and stability. A brief literature overview of these two aspects will be given in the following paragraphs.

Accuracy An accurate forecasting method generates estimates that do not deviate much from the final value it aims to predict. Predictive methods have been studied extensively from an accuracy point of view. Vanhoucke and Vandevoorde (2007) and Vandevoorde and Vanhoucke (2006) investigated the forecasting performance of three Planned Value (PV), Earned Duration (ED) and Earned Schedule (ES) methods on simulated data and on real-life projects. Elshaer (2013) followed the argumentation of Vanhoucke (2010b, 2011) to incorporate bottom-up sensitivity information into top-down control systems by including activity sensitivity information in one of the ES forecasting methods. Artificial Intelligence (AI) follows a different process for making accurate predictions. It aims to learn the relation between inputs and outputs by exploiting simulations or other historical data. The relation is then applied to the project at hand. Cheng et al. (2010) used Support Vector Machines to predict the final cost of two construction projects. Cheng and Roy (2010) tested the same system for cost estimation and function approximation. While the latter two works focused on real-life projects and applications, Wauters and Vanhoucke (2014b) applied Support Vector Regression across a large, computer-generated dataset for time and cost forecasting. In chapter 5 (Wauters and Vanhoucke (2014a)), we introduced a number of AI methods to project control and examined the strengths and weaknesses of the different methods.

Stability Forecasting methods or project control indices are said to be stable if their successive estimates or values do not deviate substantially. Traditionally, the stability of EVM metrics has been studied. Payne (1990) defined a stable Cost Performance Index (CPI) as one that does not vary more than 10% from a 20% completion point of the project onwards. Henderson and Zwikael (2008) demonstrated the lack of generalization of the CPI stability rule. Additionally, chapter 4 (Wauters and Vanhoucke (2015)) highlighted two points of criticism with regard to this stability measure. First of all, the thresholds are chosen in a completely arbitrary manner. No empirical evidence supports the general use of these thresholds. As such, these thresholds may well vary across projects and industries. Secondly, stability is reduced to a binary outcome: either a forecasting method or measure is stable or not. Chapter 4 (Wauters and Vanhoucke (2015)) resolved these problems by establishing a new stability measure, namely the mean lags. This metric has the advantage that no assumptions need to be made and that it allows for the definition of a degree of stability rather than a binary result.

The aim of this chapter is twofold. First, we extend previous research by reporting on the stability of the Artificial Intelligence methods of chapter 5 (Wauters and Vanhoucke (2014a)). Secondly, a new method for predicting the final duration of a project is proposed. The Nearest Neighbour (NN) technique does not learn the relation between inputs and outputs like the AI methods but it exploits historical data. The accuracy and stability of the NN method will be tested on a computer-generated dataset. Likewise, the historical data will result from simulations in which multiple project executions are imitated. Monte Carlo simulations allow us to draw activity durations from distributions and lead to deviations from the baseline schedule. These deviations are captured periodically by means of EVM metrics. Whenever these EVM metrics are gathered, a new estimate of the project's final duration will be made. The different estimates are aggregated into an output measure, which serves to assess the overall accuracy and stability. The Nearest Neighbour method serves a second purpose, namely to reduce the training set of an Artificial Intelligence method to a smaller set, consisting of more similar observations. The performance of the combination of Nearest Neighbours and an Artificial Intelligence method will be compared and contrasted with the AI methods presented in chapter 5 (Wauters and Vanhoucke (2014a)).

The outline of this chapter is as follows. In section 6.2, the principles of applying Nearest Neighbours for prediction (section 6.2.1) and for hybridizing the various AI methods (section 6.2.2) are divulged. Section 6.3 explains the methodology. The input modeling phase, in which the process of activity duration variation takes place, as well as the project progress and forecasting output measures are described. The methodology of simulating variation, tracking the progress of a project and constructing forecasts is repeated for every experiment. Section 6.4 elaborates on the design of experiments. Specific settings for the input modelling phase are provided in section 6.4.1. Section 6.4.2 gives additional details with regard to the Nearest Neighbours. The results of the computational experiments can be found in section 6.5. The main experiment of section 6.5.1 varies the number of neighbours and makes a distinction along the similarity of the training and test sets. The sensitivity experiments of section 6.5.2 conclude the results section. The number of observations in the training set are varied and a change in the number of observations and the number of neighbours is made. Finally, we draw conclusions and provide opportunities for future research in section 6.6.

6.2 Nearest Neighbour

As mentioned in section 6.1, the Nearest Neighbour technique employs historical data to identify the Nearest Neighbours of a given data point. Applications of the NN method typically revolve around a similar domain to that of AI methods, namely classification and prediction. Cover and Hart (1967) proposed k-Nearest Neighbours, in which the nearest k neighbours are considered for assigning a class label to a data instance. The main asset of NN techniques lies in their ease of use. However, issues regarding memory requirements and computational complexity led to the inception of many variants of the k-NN technique. A recent overview is given by Bhatia and Vandana (2010). The variant of this chapter uses a multidimensional binary search tree (also known as k-d tree) and is admirably suited for the organization of multi-dimensional points, which explains its relevance for this research.

Nearest Neighbour methods have a long history within the research community. Its applications include but are not limited to credit risk (Henley and Hand (1996)), bankruptcy prediction (Kumar and Ravi (2007)), text classification (Wan et al. (2012)), direct marketing (Govindarajan and Chandrasekaran (2010)) and TV audience forecasting (Nikolopoulos et al. (2007)).

In section 6.2.1, an outline is given of the general principle of k-Nearest Neighbours for prediction purposes. The main principle of k-NN will be part of section 6.2.2, in which we describe how Nearest Neighbours can be used to hybridize Artificial Intelligence methods.

6.2.1 NN for prediction

Prediction, along with classification, is one of the main purposes for applying a Nearest Neighbour technique. As mentioned previously, k-NN exploits historical data, which may result from earlier projects or simulations. Similar to the nomenclature of the AI body of literature, the set of known instances will be referred to as the training set. The goal of k-NN in this chapter will be to predict the final duration of a new observation. In order to compute that prediction, the training instances closest to the new observation will be utilized. Let $\mathbf{y}$ denote a vector of P attributes indexed by j. $\mathbf{y}$ symbolizes an unseen instance and comprises (part of) the test set. The training set consists of L observations $(\mathbf{x}_i, o_i)$, where i indexes training observation i and o_i denotes the output value of training instance i. $\mathbf{x}_i$ also contains P attributes. The k Nearest Neighbours of $\mathbf{y}$ are found by calculating the distance between each point of the training set i and the instance $\mathbf{y}$. The formula to calculate the Euclidean distance between a training instance i and the new observation is provided in equation (6.1). The equation calculates the square root of the square difference between all attributes j of the training instance i and the new observation.

$$||\mathbf{y} - \mathbf{x}_i|| = \sqrt{\sum_{j=1}^{P} (y_j - x_{ij})^2} \tag{6.1}$$

Once the k Nearest Neighbours have been identified, the output value of instance $\mathbf{y}$ is calculated as the average of the output values of the k Nearest Neighbours.

$$\widehat{o} = \frac{\sum_{i=1}^{k} o_i}{k} \tag{6.2}$$

In this chapter, forecasts are made periodically. EVM metrics are captured at different points in time and can be regarded as a time series. As a result, the total number of attributes depends on the time period rp and is given by equation (6.3).

$$P = \#attributes/time\ period * rp \tag{6.3}$$

6.2.2 NN for hybridizing AI methods

In this section, the second goal of the Nearest Neighbour technique will be elaborated. We commence by providing brief background information on the process of training and testing an Artificial Intelligence method and outline the role of the k-NN technique.

Every AI method aims to find a good balance between an acceptable training performance and a good generalization. An extreme focus on the training phase leads to overfitting and prevents the method from classifying or predicting new observations well. Consequently, it is required to allocate sufficient attention to the training phase, in which the parameters of the AI method are fine-tuned. The set consisting of known observations is referred to as the training set, while the unknown observations constitute the test set. In order to tune the parameters, the training set is subdivided into a second, smaller training set and a validation set. A model of the AI method is built based on the small training set, after which its performance is tested by means of the validation set. This process may be repeated to prevent overfitting. A popular method to do this is cross-validation, in which the division of training and validation is repeated a number of times. We refer to chapter 5 (Wauters and Vanhoucke (2014a)) for a detailed overview of how cross-validation is implemented in a project control environment. Once the best parameters are found, the AI model is trained on the initial training set and applied to the observations of the test set.

The inclusion of the Nearest Neighbour technique occurs in the testing phase. Hence, the best parameters for the AI method are found. As mentioned in the previous paragraph, the AI method will be trained on the larger training set and its performance will be assessed by means of the observations in the test set. The Nearest Neighbour technique operates on the larger training set and reduces it to a smaller set of observations that are more similar to the observations of the test set. The process is depicted in figure 6.1. Similar to section 6.2.1, the training set consists of L observations with P attributes. For each observation of the test set, the k Nearest Neighbours are identified using equations (6.1) and (6.2). As a result, the training set now only consists of k observations (k rows in figure 6.1). The AI method will be trained on the reduced training set consisting of fewer, high-quality observations instead of the original training set that contains L observations.

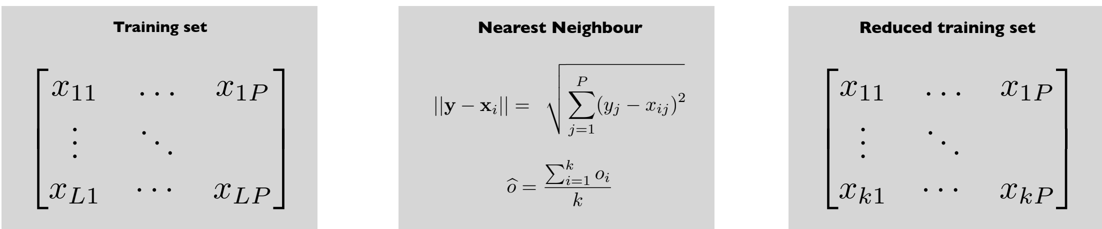


Figure 6.1: k-NN for hybridizing AI methods

6.3 Methodology

Section 6.2 explained the main contribution of this chapter, namely the dual purpose the Nearest Neighbour technique serves. In this section, we will elaborate on the methodology and translation of the Nearest Neighbour technique to a project control environment.

Prior to the project control phase, a baseline schedule needs to be constructed. A prerequisite for the baseline schedule is that an estimate of the activities' duration and cost is made. Once the baseline schedule is built, the project is executed. In this chapter, a computer-generated dataset will be employed. This implies that for every execution, the durations of the activities will be drawn from statistical distributions with certain parameter settings. While variation of the durations occurs on the activity level, project progress takes place on the project level by means of Earned Value Management metrics. These metrics are gathered periodically and comprise the input for the different forecasting methods. The Artificial Intelligence and NN methods leverage historical data, while the three Planned Value, Earned Duration and Earned Schedule methods only take the information of the current project's execution into account. Ultimately, we strive to assess the performance of the various forecasting methods. Hence, output measures that measure how accurate and how stable the predictions are need to be defined.

The outline of this section is as follows. In section 6.3.1, the necessary inputs for the baseline scheduling phase are provided. In addition, details on imitating fictitious project executions are given. Section 6.3.2 picks up where section 6.3.1 left off by elucidating the project progress phase. More specifically, it addresses which information is captured and how often. Finally, the performance metrics of the forecasting methods are described in section 6.3.3.

6.3.1 Input modeling

The power of simulation for project management lies in the wide array of projects with different time and cost characteristics that can be generated. Kwak and Ingall (2007) recognize that Monte Carlo simulation can play a vital role in understanding the effects of uncertainty on projects. Additionally, it allows us to quantify the impact of uncertainty. In this chapter, the effect of uncertainty will be reflected in the accuracy and stability of the various forecasting methods.

The topological structure of the dataset is drawn from the dataset that was generated

by the project generation tool RanGen (Demeulemeester et al. (2003) and Vanhoucke et al. (2008)) and that has been employed in previous simulation studies (Elshaer (2013), Colin and Vanhoucke (2014), Colin et al. (2015)). In this chapter, we focus on 10 projects counting 30 activities each and possessing a value of 0.5 for the Serial/Parallel (SP) indicator. The SP indicator serves as a metric for a project's topological structure and quantifies a project network's resemblance to a completely serial or parallel project. The main rationale for fixing the SP value to 0.5 results from previous studies (Vanhoucke (2011), Wauters and Vanhoucke (2014a,b)) that established a clear relation between the performance of EVM systems and forecasting and the topological structure. As a project's network structure becomes increasingly serial, EVM forecasting performance improves. This result is hardly surprising. Earned Value Management is a top-down control method. As a project becomes more serial, the performances on the activity and project levels coincide.
The baseline duration and cost for each activity is drawn from the interval $[200, 300]$ and $[50, 100]$, respectively. The costs are entirely variable and imply that a deviation from the baseline duration is translated into costs.

Once the baseline schedule is built, fictitious progress executions are generated. To that end, it is necessary to specify how the real durations of the activities will deviate from their baseline durations. This is done by means of a probability distribution. For every execution, a number will be drawn according to the activity's probability function. In this chapter we opt for the generalized beta distribution, which was used for academic and practical purposes (AbouRizk et al. (1994)). Additionally, the generalized beta distribution has the advantage that it is relatively straightforward to modify the mean μ and mode m of the distribution. The probability density function of a random variable x is given in equation (6.4). $\Gamma(\cdot)$ refers to the gamma function and θ_1 and θ_2 are two shape parameters. a and b are the lower and upper limits of the random variable.

$$f(x) = \frac{\Gamma(\theta_1 + \theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)(b-a)^{\theta_1+\theta_2-1}}(x-a)^{\theta_1-1}(b-x)^{\theta_2-1}, \quad x \in [a, b] \tag{6.4}$$

We control the simulations by modifying a, b, μ and m. The shape parameters of the generalized beta distribution can then be calculated based on these 4 numbers, with $\theta_1 = -\frac{(b+a-2m)(a-\mu)}{(m-\mu)(a-b)}$ and $\theta_2 = \frac{(b+a-2m)(b-\mu)}{(m-\mu)(a-b)}$.

6.3.2 Project progress

Activity variation is achieved by drawing numbers from the generalized beta distribution specified in equation (6.4). For practical reasons, it is impossible to track each activity's progress along the execution. Hence, EVM aggregates the performance of individual activities and translates them to the project level. EVM makes use of three key numbers, namely Planned Value (PV), Earned Value (EV) and Actual Cost (AC). The performance metrics and forecasting methods are all derived from these key metrics. Throughout our experiments, it is assumed that the Earned Value for an activity follows a linear accrue from 0 until its Budget At Completion (BAC). Likewise, the Planned Value is accrued in a linear manner (Vanhoucke (2010a)).

The EVM metrics and forecasting methods that are calculated periodically are provided in table 6.1. In this table AD and PD denote the Actual Duration and Planned Duration. The final forecasting method of table 6.1 requires additional explanation. Elshaer (2013) responded to the call of Vanhoucke (2011) to incorporate sensitivity information in EVM metrics by adapting the calculation of EV and PV. The Planned Value of a sensitivity metric α at time point t is equal to the sum of the activities' planned values multiplied by the activities' values for sensitivity metric α. A similar reasoning applies to the adaptation of EV. In accordance with Elshaer (2013) we include 6 sensitivity metrics, namely the Criticality Index (CI), the Significance Index (SI), the Schedule Sensitivity Index (SSI) and the Cruciality Index with Pearson's product moment (CRI_r), Spearman's rank correlation (CRI_ρ) and Kendall's τ rank correlation (CRI_τ). Hence, $\alpha \in \{\text{CI, SI, SSI, CRI}_r, \text{CRI}_\rho, \text{CRI}_\tau\}$. Since the PV and EV calculations were changed, the subsequent calculation of ES and SPI(t) differ as well. This is reflected in table 6.1 by means of an apostrophe.

The EVM attributes of table 6.1 are captured every 10% complete. As a result, the project manager possesses multiple estimates and metrics near the end of the project. Table 6.1 summarizes the calculations of the 19 attributes (4 performance indices, 9 EVM forecasting methods and 6 Elshaer (2013) methods). These 19 attributes correspond with *#attributes* of equation (6.3). At the 90% complete point ($rp = 9$), there are $P = 171$ (9 * 19) attributes. The attributes are used by the Artificial Intelligence methods. Furthermore, they correspond with the P attributes of equation (6.1), indexed by j. The k Nearest Neighbours are calculated as the difference between the EVM attributes of the test set observation(s) and the instances of the training set.

Name	Attribute	Calculation
Schedule Performance Index	SPI	$\frac{EV}{PV}$
Schedule Performance Index (time)	SPI(t)	$\frac{ES}{AT}$
Cost Performance Index	CPI	$\frac{EV}{AC}$
Earned Schedule	ES	$t + \frac{EV-PV_t}{PV_{t+1}-PV_t}$
	EAC(t)_{PV_1}	$PD - \frac{(EV-PV)*PD}{BAC}$
Estimate At Completion (time) - Planned Value	EAC(t)_{PV_2}	$\frac{PD}{SPI}$
	EAC(t)_{PV_3}	$\frac{PD}{CPI*SPI}$
	EAC(t)_{ED_1}	$PD + AD * (1 - SPI)$
Estimate At Completion (time) - Earned Duration	EAC(t)_{ED_2}	$\frac{PD}{SPI}$
	EAC(t)_{ED_3}	$\frac{PD}{SPI*CPI} + AD * (1 - \frac{1}{CPI})$
	EAC(t)_{ES_1}	$AD + PD - ES$
Estimate At Completion (time) - Earned Schedule	EAC(t)_{ES_2}	$AD + \frac{PD-ES}{SPI(t)}$
	EAC(t)_{ES_3}	$\frac{PD-ES}{CPI*SPI(t)}$
Estimate At Completion (time) - Elshaer (2013)	$\text{EAC(t)}_{ES_2\alpha}$	$AD + \frac{PD-ES'}{SPI(t)'}$

Table 6.1: Overview of the EVM attributes (source: chapter 5 (Wauters and Vanhoucke (2014a)))

6.3.3 Output measures

Section 6.3.2 discussed the information that is retained to keep track of the project's progress and to apply the AI methods and k-NN technique. Since the project's performance is assessed every 10% complete, there are 9 review periods. At each review period, a new estimate of the project's final duration is made. This is done for the three PV, ED and ES methods, the 6 Elshaer (2013) methods, the AI methods of chapter 5 (Wauters and Vanhoucke (2014a)), their hybrid counterparts as detailed in section 6.2.2, and the k-NN technique of section 6.2.1. The output measures of forecasting methods should quantify the capability of each method to produce an accurate or stable prediction. A good forecasting method produces predictions that do not deviate much from the actual value (accurate) and do not differ much along subsequent time periods (stable).

Accuracy From an accuracy point of view the Mean Absolute Percentage Error (MAPE) metric has been used in previous project control studies (Vanhoucke and Vandevoorde (2007), Elshaer (2013) and Wauters and Vanhoucke (2014b)). The MAPE is calculated according to equation (6.5), in which $EAC(t)_{rp}$ denotes the Estimate At Completion (time) for review period rp. The deviation from the Real Duration is measured and averaged across all R review periods. A new prediction is made every 10% complete implying that $R = 9$.

$$MAPE = \frac{1}{R}\sum_{rp=1}^{R}\frac{|RD - EAC(t)_{rp}|}{RD} * 100 \tag{6.5}$$

Stability Measurement of the stability of a forecasting method is done by the Mean Lags criterion proposed in chapter 4 (Wauters and Vanhoucke (2015)). Compared to the previous stability rule, measuring the Mean Lags has the advantage of assigning a degree of stability to a forecasting method instead of a binary outcome. Calculation of the Mean Lags is provided in equation (6.6). The stability of the forecasting method appraises the difference between two subsequent estimates in time, regardless of how much the estimate is separated from the actual value. However, the combination of the MAPE and Mean Lags yields a complete picture of the capability of a given predictive method.

$$Mean\ Lags = \frac{1}{R-1}\sum_{rp=2}^{R}\frac{|EAC(t)_{rp} - EAC(t)_{rp-1}|}{EAC(t)_{rp-1}} * 100 \tag{6.6}$$

6.4 Experimental design

This section will provide information with regard to the computational experiments we conducted. The goal of these experiments is to assess the quality of the various forecasting methods and the effect of the Nearest Neighbour technique as a predictor and as an addition to various AI methods. This section will lay down the structure of the results which are presented in section 6.5. Additionally, we discuss the settings for k, the number of Nearest Neighbours.

6.4.1 Experiments

Section 6.3.1 discussed the use of statistical distributions to imitate fictitious project executions. Artificial Intelligence techniques rely on data resulting from these simulations to learn the relationship between inputs and outputs. In a project control environment, the inputs constitute EVM information (found in table 6.1) while the output is forecasting performance. The k-NN technique also relies on simulated data in order to identify the k Nearest Neighbours. While this technique does not learn the relationship between inputs and outputs, it requires the presence of historical or simulated data to function properly. In our experiments, we made use of the generalized beta distribution. The various settings that were employed for the generalized beta distribution can be divided into two classes, namely a class in which the training set coincides with the test set and a

Scenario	a	b	m	μ	θ_1	θ_2	σ
				Training = Test			
Early	0.1	2	0.5	0.6	2.93	8.22	0.24
Middle	0.2	4	0.82	1	2.94	11.03	0.40
Late	0.2	4	1.2	1.4	2.83	6.14	0.56
				Training $\neq$ *Test*			
Change in μ							
$\Delta\mu_1$	0.2	4	0.49	0.79	1.67	9.13	0.40
$\Delta\mu_2$	0.2	4	0.65	0.89	2.24	10.13	0.40
$\Delta\mu_3$	0.2	4	0.98	1.12	3.80	11.87	0.40
$\Delta\mu_4$	0.2	4	1.14	1.25	4.75	12.35	0.40
Change in σ							
$\Delta\sigma_1$	0.2	4	0.94	1	8.53	31.99	0.24
$\Delta\sigma_2$	0.2	4	0.89	1	4.72	17.70	0.32
$\Delta\sigma_3$	0.2	4	0.70	1	1.98	7.44	0.48
$\Delta\sigma_1$	0.2	4	0.53	1	1.40	5.26	0.56

Table 6.2: Generalized beta settings for the various scenarios (source: chapter 5 (Wauters and Vanhoucke (2014a)))

class where this is not the case. For the latter class, gradual changes to the mean μ and standard deviation σ were made. Table 6.2 shows the settings of the generalized beta distributions for both classes. The first three rows correspond with situations where the training set is equal to the test set. As a result, the simulations to train the AI methods or to select the Nearest Neighbours are drawn from the same distribution as the simulations of the test set. Alternatively, the training and test sets were varied by either modifying the mean and keeping the standard deviation constant or vice versa.

For each of the 10 projects and every scenario, 1,000 executions were performed. Identical to chapter 5 (Wauters and Vanhoucke (2014a,b)), 800 executions were dedicated to training and validation whereas the remainder served the purpose of testing the accuracy and stability. As mentioned in section 6.3.2, a prediction is made for each review period which means that for every execution, the MAPE and Mean Lags are based on 9 data points.

6.4.2 Nearest Neighbour Settings

Researching the potential of the Nearest Neighbour technique forms the main contribution of this chapter. In this paragraph we discuss three aspects related to Nearest Neighbours, namely pre-processing, the settings for k that were tested and the inclusion of the optimal neighbours.

6.4.2.1 Pre-processing

In section 6.2, we discussed how the Nearest Neighbours are calculated based on the distance between all attributes of a training instance and the new observation. In the context of this chapter, the attributes were provided in table 6.1. However, an important caveat surrounds the use of the EVM attributes. The indicators SPI, SPI(t) and CPI are expressed as fractions, while Earned Schedule as well as the various forecasting methods are expressed in absolute numbers. Hence, calculation of the distance will be biased by those attributes that are subject to a different scale. Normally, the difference in distance of forecasting methods will be much higher compared to the difference of one of the Performance Indices. Consequently, the Nearest Neighbour calculations will be dominated by the difference in forecasting estimates.

In order to eliminate this pre-dominance of EVM forecasting methods in Nearest Neighbour calculations, pre-processing of the attributes of table 6.1 was included and resulted in three Nearest Neighbour variants. These variants will be explained in the following paragraphs.

No pre-processing The first variant did not include pre-processing.

Scaling Scaling is the second variant and executes pre-processing as follows. It performs two operations on a vector $\mathbf{x}$, which consists of a number of values indexed by i. The first operation is centering by subtracting the mean of $\mathbf{x}$ from x_i. Secondly, scaling takes place by dividing the value from the first operation by the standard deviation of $\mathbf{x}$. It is worth noting that for a given attribute j, the notation of this paragraph corresponds with the notation of equation (6.1) in which i symbolized a training instance. The scaling operation can be expressed mathematically by equation (6.7).

$$\frac{x_i - \bar{x}}{\sigma_x} \tag{6.7}$$

Principal Component Analysis The final variant applies Principal Component Analysis (PCA, Pearson (1901)) to the attributes of table 6.1. PCA removes data redundancy and noise by creating a number of principal components which are linear combinations of the original attributes. The notion of data redundancy is not trivial in a project control context since the forecasting methods employ some of the performance factors (SPI, SPI(t) and CPI) which are individual attributes as well. The reader is referred to Jolliffe (2005) and chapter 5 (Wauters and Vanhoucke (2014a)) for details on how the principal components were derived. In order to preserve consistency, the number of principal components was determined in the same way as in chapter 5 (Wauters and Vanhoucke (2014a)). The minimum number of principal components that explains $t = 90\%$ of the explained variation will be retained.

6.4.2.2 Number of neighbours

The k-NN technique serves two purposes in this chapter. First of all, a prediction is made by means of the average Real Duration of the k Nearest Neighbours that lie closest to the test set instance. Secondly, the k Nearest Neighbours reduce the training set to a smaller training set with more similar observations. Both aspects were discussed at length in section 6.2.

In this section we provide the settings that were implemented for our experiments. The various settings for k are provided in table 6.3. The column labeled "$k(\%)$" refers to the percentage of the observations of the training set. Section 6.4.1 revealed that there are 800 observations in the training set, implying that a percentage of 0.00125 corresponds with 1 ($= 0.00125 * 800$) neighbour. Table 6.3 shows that there is a difference in values for $k(\%)$ depending on the purpose for which the Nearest Neighbour technique is applied. When the Real Duration is predicted (the second row of table 6.3), the values of the hybridization (the third row of table 6.3) are supplemented by lower values for $k(\%)$. The reason why no extremely low values for $k(\%)$ are allowed when hybridizing AI methods is because the AI methods require a minimum number of training observations in order to construct a model. We employ the notation *lower value-upper value*, $\Delta =$ *increment* in table 6.3 to denote a sequence of values. For hybridizing AI methods the number of neighbours goes from 10% to 90% with a 10% increment.

Purpose	$k(\%)$
Prediction	$\{0.00125, 0.0025, 0.01, 0.05, (0.1\text{-}0.9, \Delta{=}0.1)\}$
Hybridization	0.1-0.9, $\Delta = 0.1$

Table 6.3: Settings for k, the number of neighbours, expressed as a percentage of the training set observations

6.4.2.3 Utopian scenario

All observations of the training set are assumed to be known. This implies that the entire project progress, as well as the Real Duration, is known. The observations of the test set are new observations. Hence, only the progress information up to review period rp is known, while the Real Duration is the main variable of interest. The Nearest Neighbour technique makes a prediction by finding the neighbours that are closest to the test set observations and utilizing the average of the neigbours' Real Duration as a prediction. The distance calculation follows from the fact that the test set's RD is unknown. Hence, the EVM attributes serve as a proxy for the RD. The k-NN technique implicitly assumes that the Nearest Neighbours, found by calculating the distance between all EVM attributes, will also be the Nearest Neighbours in terms of their RD.

Throughout the results of our experiments, we will also report on the "Utopian" scenario. This scenario assumes that we can find the optimal neighbours by calculating the difference between the RD of the training set and the test set observation. Equation (6.1) then becomes $||\mathbf{y} - \mathbf{x}_i|| = RD - RD_i$. While this scenario is unrealistic (the test set's RD is unknown), it provides an estimate of the upside potential of the k-NN technique. The inclusion of the utopian scenario is of particular relevance for hybridization. When hybridizing the AI methods with the k-NN technique, one can wonder whether forecasting accuracy is driven by the limitations of simulation, the inability to identify the optimal neighbours or both. If the difference in forecasting accuracy between the hybrid AI methods and their Utopian counterparts is small, it can be concluded that the influence of the optimal neighbours is small. In the opposite case we can conclude that the simulations contain sufficient and relevant EVM information but that the problem lies in finding the optimal neighbours.

6.5 Results

In this section, the forecasting performance of the k-NN technique will be compared with and contrasted to the existing methods. Among those are three PV, ED and ES techniques, as well as the 6 Elshaer (2013) methods and the AI techniques proposed in chapter 5 (Wauters and Vanhoucke (2014a)). We report on the accuracy, measured by the MAPE, and the stability, measured by the Mean Lags. The simulations, along with the calculation of project progress metrics, are achieved by P2 Engine (Vanhoucke (2014)), a LUA-based scripting tool. The AI techniques and accuracy and stability calculations were implemented in R Core Team (2013), the open-source programming language. The packages of Meyer et al. (2012), Ridgeway (2013), Hothorn et al. (2006b) and Beygelzimer et al. (2013) were employed to train and test the (hybrid) AI methods.

The structure of this section mainly follows the outline of section 6.4.1. The main experiment assesses the accuracy and stability of the k-NN technique and the hybrid AI methods. A distinction is made based on whether the training and test sets coincide (cf. table 6.2). The number of neighbours are varied according to table 6.3 as discussed in section 6.4.2.2. We also conduct some sensitivity experiments. In these experiments, the influence of the hybrid AI methods is investigated when either the number of observations of the training set or the observations of the training set and the number of neighbours are varied. These experiments allow us to establish the influence of the amount of observations on the forecasting accuracy and stability.

A schematic overview of the results section is depicted in figure 6.2. Section 6.5.1 contains the results of the main experiment. 800 executions (=100%) make up the training set, while the number of neighbours is varied from 0.00125 (or 0.1) to 1.0. In the sensitivity experiments, the number of executions of the training set is modified from 0.1 (80 neighbours) to 1.0 (800 neighbours) in 0.1 increments (section 6.5.2.1). Next, we also assess the impact of a simultaneous change in executions and neighbours (section 6.5.2.2).

6.5.1 Main Experiment

The main experiments of this chapter investigate the forecasting performance when the number of neighbours is changed. All executions are included in the training set, which consists of 800 observations (cf. figure 6.2). A division is made based on whether the training and test sets coincide (section 6.5.1.1) or not (section 6.5.1.2).

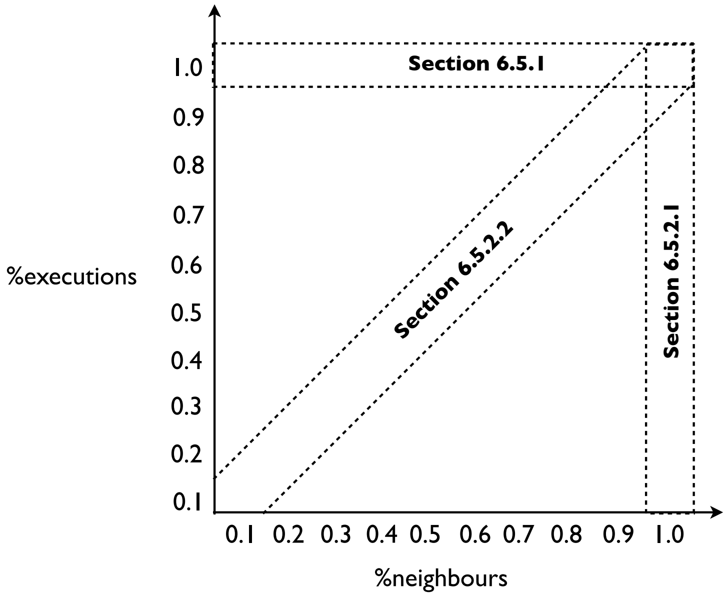


Figure 6.2: Overview of the results section

6.5.1.1 Training set = test set

In this section, the observations of the training set and test set are drawn from identical distributions. This implies that the project manager is capable of making a fair assessment of the variability that will occur throughout the project's progress. The accuracy results of the EVM methods and the AI/NN methods are provided in tables 6.4 and 6.5, respectively. These tables provide the forecasting accuracy (measured by the MAPE of equation (6.5)) for each scenario. The names of the scenarios correspond with those of table 6.2. Before proceeding to the discussion of the results, it is necessary to elaborate on the structure of table 6.5, which will also be used for section 6.5.1.2. Table 6.5 counts 3 rows for each scenario, which will be described along the following lines:

- The row labeled "Regular" provides the results of the AI techniques of chapter 5 (Wauters and Vanhoucke (2014a)) or the Nearest Neighbour methods. Three Nearest Neighbour methods are included, according to the pre-processing technique that was applied (cf section 6.4.2.1). Principal Component Analysis and Scaling are denoted by PCA and Scaling, respectively, while the absence of pre-processing is indicated by a hyphen.

- The row labeled "Hybrid" lists the results of the hybrid AI methods. The k-NN technique first reduced the training set according to the procedure of figure 6.1, after which the AI method under study is executed. Obviously, no hybridization takes place for the NN methods which explains why there are only 5 output measure values for these rows.

- "Utopian" provides the results of the AI or Nearest Neighbour method for the utopian scenarios described in section 6.4.2.3. The reader is reminded that while the utopian scenario assumes perfect knowledge, it provides a good measure of the upside potential of the Nearest Neighbour method, either as a predictor or as a hybridizer for the AI methods.

Tables 6.6 and 6.7 show the stability results of the EVM, AI and Nearest Neighbours methods respectively. The stability is measured by means of the Mean Lags criterion of equation (6.6). The accuracy and stability results are summarized along the following lines. First, we compare the AI performance with that of the various EVM forecasting methods. Secondly, the accuracy and stability of the Nearest Neighbour methods is discussed. Finally, we dedicate attention to the performance of the Utopian scenario.

- AI performance: two main observations with regard to the performance of the Artificial Intelligence methods can be made.
  - First, the performance of the AI (and Nearest Neighbour) methods does not vary much along the scenario. This observation holds for both forecasting accuracy and stability. The performance of the EVM methods differs greatly along the scenario. This performance gap is most visible for the MAPE of table 6.4. The worst performance is noted for the Early scenario, followed by the Late and Middle scenarios respectively. The ES_2 and Elshaer (2013) methods, which are based on the ES_2 method, do not suffer from this discrepancy in performance.
  - Secondly, one of the goals of this chapter consisted of extending the stability results of chapter 4 (Wauters and Vanhoucke (2015)) to the AI methods. Table 6.7 shows that the AI methods outperform the EVM methods, especially for the Early and Late scenarios.

- Nearest Neighbour performance: we comment on the accuracy and stability performance on the Nearest Neighbour methods as a predictor and evaluate the hybrid AI methods.
  - The Nearest Neighbour methods outperform the EVM methods in terms of accuracy for the Early and Middle scenarios. The Nearest Neighbour methods

are slightly less accurate than the other AI techniques. The main asset of the Nearest Neighbour methods is found in their stellar stability performance.The Nearest Neighbour method without pre-processing yields the best results. The difference with the incumbent method, RF, is on average 60.50%.

  - The hybrid counterparts are able to improve the performance of the Decision Tree method. On average, the MAPE improvement is equal to 3.93%. A possible explanation for this improvement is related to the instability of trees (Hastie et al. (2009)). Small changes to the input data can result in a completely different tree structure. By restricting the training set to more similar observations, more stable trees may be obtained. The other AI methods build ensembles, making them less susceptible to this problem.

- Utopian scenario: we draw the reader's attention to the excellent performance of the Utopian scenario, which leads us to conclude that there is vast potential for the Nearest Neighbour methods. The challenge lies in finding a good proxy for reliably estimating the difference in RD between training and test set observations without any knowledge of the RD of the test set.

Scenario	Planned Value			Earned Duration			Earned Schedule			Elshaer (2013)					
	PV_1	PV_2	PV_3	ED_1	ED_2	ED_3	ES_1	ES_2	ES_3	ES_{CI}	ES_{SI}	ES_{SSI}	ES_{CRI_r}	ES_{CRI_ρ}	ES_{CRI_τ}
Early	38.16	18.91	37.92	44.16	18.91	24.07	40.72	10.91	26.06	12.63	11.74	12.63	11.6	11.58	11.54
Middle	6.6	10.68	17.33	6.59	10.66	15.13	6.37	10.85	15.54	12.57	11.67	12.57	11.53	11.5	11.48
Late	19.33	14.68	32.39	19.04	13.19	22.66	17.28	10.92	26.86	12.64	11.74	12.64	11.61	11.59	11.58

Table 6.4: MAPE (%) of the EVM forecasting methods (training set = test set)

Scenario	Criterion	Artificial Intelligence					Nearest Neighbour		
		DT	Bagging	RF	Boost	SVM	PCA	Scaled	-
Early	Regular	6.79	6.47	6.42	6.31	6.17	7.75	7.77	7.75
	Hybrid	6.5	6.47	6.42	6.31	6.17			
	Utopian	0.55	0.64	0.63	0.8	0.84	0.03	0.03	0.03
Middle	Regular	6.81	6.44	6.4	6.27	6.14	7.78	7.76	7.73
	Hybrid	6.53	6.44	6.4	6.27	6.14			
	Utopian	0.55	0.63	0.62	0.81	0.84	0.03	0.03	0.03
Late	Regular	6.74	6.41	6.36	6.24	6.16	7.79	7.8	7.75
	Hybrid	6.51	6.41	6.36	6.24	6.16			
	Utopian	0.55	0.64	0.62	0.81	0.89	0.03	0.03	0.03

Table 6.5: MAPE (%) of the AI and NN forecasting methods (training set = test set)

Scenario	Planned Value			Earned Duration			Earned Schedule			Elshaer (2013)					
	PV_1	PV_2	PV_3	ED_1	ED_2	ED_3	ES_1	ES_2	ES_3	ES_{CI}	ES_{SI}	ES_{SSI}	ES_{CRI_r}	ES_{CRI_ρ}	ES_{CRI_τ}
Early	5.07	12.11	15.81	4.26	12.11	15.72	4.34	7.51	12.18	7.2	6.95	7.2	7.11	7.12	7.17
Middle	3.03	8.26	12.06	2.37	8.24	11.22	2.09	7.44	10.58	7.11	6.86	7.1	7.03	7.03	7.06
Late	4.52	9.57	12.54	3.71	8.94	12.23	3.65	7.56	11.19	7.25	7.01	7.25	7.17	7.17	7.19

Table 6.6: Mean Lags (%) of the EVM forecasting methods (training set = test set)

Scenario	Criterion	Artificial Intelligence					Nearest Neighbour		
		DT	Bagging	RF	Boost	SVM	PCA	Scaled	-
Early	Regular	3.29	2.32	2.12	2.52	2.42	0.87	1.32	0.83
	Hybrid	1.98	2.32	2.12	2.52	2.42			
	Utopian	0.05	0.22	0.19	0.53	0.25	0	0	0
Middle	Regular	3.26	2.29	2.1	2.45	2.41	1.37	0.87	0.88
	Hybrid	2.06	2.29	2.1	2.45	2.41			
	Utopian	0.05	0.21	0.19	0.53	0.27	0	0	0
Late	Regular	3.27	2.34	2.11	2.45	2.46	1.36	1.36	0.79
	Hybrid	2.11	2.34	2.11	2.45	2.46			
	Utopian	0.07	0.23	0.19	0.53	0.32	0	0	0

Table 6.7: Mean Lags (%) of the AI and NN forecasting methods (training set = test set)

6.5.1.2 Training set $\neq$ test set

This section assumes that the project manager is not able to estimate the activity variation correctly. As a result, some parameters of the underlying generalized beta distribution differ from the training set. These differences were classified into two categories, namely a class in which the mean μ was modified and one where the standard deviation σ was changed. Similar to the previous section, the main conclusions based on table 6.10 (accuracy) and 6.11 (stability) are listed below. First we make a general observation, after which we comment on the performance of the AI methods and the Nearest Neighbour methods. Finally, room for improvement is measured by means of the Utopian scenario.

- General observation: a change in the distribution's mean resorts a larger, more negative effect on the forecasting accuracy than a change in the standard deviation. Interestingly, this conclusion is not valid for the forecasting stability, implying that the predictions are further removed from the Real Duration but do not necessarily fluctuate more.

- EVM performance: the accuracy results of the various EVM methods are similar to those reported in section 5.5.3 of chapter 5. While the EVM methods yield a lower MAPE compared to the AI and NN forecasting methods, the difference becomes smaller as the mean increases. Stability-wise, ES_1 is the most stable forecasting method. We observe that as the standard deviation increases, the Mean Lags increase on average.

- AI performance: the Support Vector Machine method is most sensitive to input changes. This is reflected in the MAPE and Mean Lags performance, which is considerably higher than the performance of the other AI and Nearest Neighbour methods. Consequently, we can corroborate earlier findings (cf. chapter 5 (Wauters and Vanhoucke (2014b))) and add that this sensitivity to input changes also holds true for forecasting stability.

- Nearest Neighbour performance: two observations concerning the NN performance can be made. The first one is related to Nearest Neighbours as a predictor, while the latter revolves around hybridization of the AI methods.

  - The Nearest Neighbour methods with scaling and without pre-processing are the most stable methods. Similar to section 6.5.1.1, a large difference with the incumbent method can be noted.

Δ	DT	Bagging	RF	Boost	SVM
			Accuracy		
$\Delta\sigma$	2.79	4.79	4.67	6.19	8.29
$\Delta\mu$	1.71	1.72	2.50	2.72	5.30
			Stability		
$\Delta\sigma$	35.59	4.84	9.95	2.77	0.00
$\Delta\mu$	32.58	4.87	6.94	2.80	21.99

Table 6.8: Average % improvement of the hybrid counterparts

- The hybrid counterparts of the AI methods play a more important role compared to section 6.5.1.1. When the training and test sets coincide, the hybrid counterparts could only improve the DT method. If the training and test set observations are no longer drawn from the same distribution, every AI method is improved by the inclusion of the Nearest Neighbours technique. The average improvement for a change in σ and μ is provided in table 6.8. Additionally, we have provided details on the optimal number of neighbours to use. These can be found in table 6.9. Interestingly, the best values for k remain rather stable across all variations of σ and μ. We observe that Support Vector Machines benefit from a higher number of neighbours ($k = 0.8$), while the other techniques operate on a limited amount of data ($k < 0.3$ in most cases). It is worth noting that the hybrid counterparts of the AI methods have a very favourable effect on the overall stability.

- Utopian scenario: similar to section 6.5.1.1, the Utopian scenario reveals the promising potential for the hybrid and Nearest Neighbour methods. It is worth mentioning that the MAPE trend (MAPE $\Delta\mu >$ MAPE $\Delta\sigma$) also holds true for the Utopian scenarios. While the SVM performance can be greatly improved, it remains the most sensitive method among the AI techniques.

Scenario	DT	Bagging	RF	Boost	SVM
		Change in σ			
Δ_{σ_1}	0.2	0.2	0.2	0.3	0.8
Δ_{σ_2}	0.2	0.2	0.2	0.3	0.8
Δ_{σ_3}	0.2	0.2	0.2	0.3	0.8
Δ_{σ_4}	0.2	0.2	0.2	0.3	0.8
		Change in μ			
Δ_{μ_1}	0.1	0.2	0.2	0.3	0.8
Δ_{μ_2}	0.2	0.2	0.2	0.3	0.8
Δ_{μ_3}	0.2	0.3	0.3	1	0.8
Δ_{μ_4}	0.2	1	1	1	1

Table 6.9: Optimal k for the hybrid counterparts

Scenario	Criterion	Artificial Intelligence					Nearest Neighbour		
		DT	Bagging	RF	Boost	SVM	PCA	Scaled	-
				Change in σ					
Δ_{σ_1}	Regular	9.28	9.13	9.35	9.86	31.31	11.55	12.87	11.58
	Hybrid	8.82	8.49	8.8	8.91	28.67			
	Utopian	1.75	1.58	1.6	1.57	9.26	0.1	0.1	0.1
Δ_{σ_2}	Regular	10.09	9.92	10.13	10.54	31.82	12.3	13.49	12.37
	Hybrid	9.73	9.36	9.6	9.73	29.02			
	Utopian	2.17	1.99	2	1.73	9.68	0.19	0.19	0.19
Δ_{σ_3}	Regular	12.21	12.03	12.26	12.35	33.17	14.21	15.22	14.35
	Hybrid	12.02	11.59	11.76	11.81	30.42			
	Utopian	3.32	3.07	3.09	2.41	10.56	0.42	0.42	0.42
Δ_{σ_4}	Regular	13.35	13.14	13.38	13.32	33.83	15.37	16.24	15.54
	Hybrid	13.21	12.77	12.91	12.91	31.24			
	Utopian	3.98	3.69	3.72	2.84	11.38	0.61	0.61	0.61
				Change in μ					
Δ_{μ_1}	Regular	30.23	29.96	30.92	29.7	49.26	33.05	36.37	32.65
	Hybrid	30.13	29.22	29.58	28.36	44.93			
	Utopian	15.53	14.43	14.56	10.87	21.24	3.7	3.7	3.7
Δ_{μ_2}	Regular	18.64	18.5	19.04	18.67	39.34	21.14	23.23	21.08
	Hybrid	18.39	17.77	18.09	17.48	35.08			
	Utopian	7.02	6.44	6.51	4.7	14.77	0.96	0.96	0.96
Δ_{μ_3}	Regular	9.27	8.87	8.96	9.28	29.16	11.42	12.25	11.31
	Hybrid	9.01	8.83	8.9	9.28	28.7			
	Utopian	2.32	2.22	2.23	1.87	9.48	0.57	0.57	0.57
Δ_{μ_4}	Regular	12.73	12.2	12.58	11.87	28.78	15.07	15.91	14.12
	Hybrid	12.43	12.2	12.58	11.87	28.78			
	Utopian	5.05	4.79	4.82	3.73	12.26	1.61	1.61	1.61

Table 6.10: MAPE (%) of the AI and NN forecasting methods (training set $\neq$ test set)

Scenario	Criterion	Artificial Intelligence DT	Bagging	RF	Boost	SVM	Nearest Neighbour PCA	Scaled	-
				Change in σ					
Δ_{σ_1}	Regular	3.91	3.25	2.88	6.45	34.94	2.51	0.76	0.92
	Hybrid	2.53	3.09	2.57	6.15	34.94			
	Utopian	0.12	0.45	0.39	1.41	7.86	0	0	0
Δ_{σ_2}	Regular	3.92	3.26	2.89	6.49	35.46	2.38	0.75	0.87
	Hybrid	2.55	3.11	2.58	6.24	35.46			
	Utopian	0.13	0.47	0.41	1.49	10.94	0	0	0
Δ_{σ_3}	Regular	4.04	3.33	2.94	6.56	40	2.16	0.75	0.83
	Hybrid	2.61	3.17	2.66	6.43	40			
	Utopian	0.16	0.53	0.45	1.71	9.99	0	0	0
Δ_{σ_4}	Regular	4.11	3.37	2.96	6.54	37.76	2.12	0.78	0.9
	Hybrid	2.6	3.2	2.7	6.5	37.76			
	Utopian	0.18	0.56	0.47	1.8	16.22	0	0	0
				Change in μ					
Δ_{μ_1}	Regular	3.94	3.29	2.87	6.55	35.47	2.18	0.78	1.18
	Hybrid	2.64	3.16	2.71	6.56	35.47			
	Utopian	0.28	0.89	0.73	3.49	11.93	0	0	0
Δ_{μ_2}	Regular	4.01	3.3	2.91	6.57	35.2	2.26	0.75	1.03
	Hybrid	2.6	3.15	2.66	6.57	35.17			
	Utopian	0.2 0	0.67	0.57	2.45	11.45	0	0	0
Δ_{μ_3}	Regular	3.88	3.31	2.9	6.41	65.07	2.33	0.76	0.91
	Hybrid	2.6	3.1	2.63	6.05	48.16			
	Utopian	0.15	0.46	0.39	1.24	15.99	0	0	0
Δ_{μ_4}	Regular	3.67	3.23	2.81	6.27	91.58	2.26	0.8	0.93
	Hybrid	2.6	3.08	2.69	5.92	34.65			
	Utopian	0.23	0.62	0.51	1.71	13.85	0	0	0

Table 6.11: Mean Lags (%) of the AI and NN forecasting methods (training set $\neq$ test set)

Scenario	PV1	PV2	PV3	ED1	ED2	ED3	ES1	ES2	ES3	ES2CI	ES2SI	ES2SSI	ES2CRIr	ES2CRIrho	ES2CRItau
								Change in σ							
Δ_{σ_1}	4.19	6.91	10.73	4.21	6.9	9.36	3.98	6.68	9.27	7.64	7.05	7.64	6.94	6.91	6.92
Δ_{σ_2}	5.48	8.91	13.95	5.52	8.9	12.15	5.23	8.76	12.18	9.98	9.23	9.98	9.07	9.05	9.05
Δ_{σ_3}	8.11	12.88	20.36	8.18	12.83	17.69	7.8	13.01	18.15	14.7	13.61	14.7	13.42	13.39	13.39
Δ_{σ_4}	9.35	14.73	23.37	9.42	14.64	20.27	9	15.01	20.98	16.9	15.65	16.9	15.46	15.43	15.43
								Change in μ							
Δ_{μ_1}	16.47	15.37	26.58	18.32	15.37	20	16.82	13.53	20.49	15.3	14.15	15.3	13.95	13.91	13.91
Δ_{μ_2}	10.07	12.84	20.81	10.83	12.84	17.1	10.03	12.25	17.41	13.86	12.83	13.86	12.64	12.6	12.6
Δ_{μ_3}	8.12	10.17	17.71	8.21	10.01	14.36	7.69	9.58	14.78	10.89	10.08	10.89	9.91	9.89	9.89
Δ_{μ_4}	12.85	10.92	22.82	13.04	10.3	16.11	11.89	8.62	17.45	9.82	9.08	9.82	8.94	8.91	8.92

Table 6.12: MAPE (%) of the EVM forecasting methods (training set $\neq$ test set)

Scenario	PV1	PV2	PV3	ED1	ED2	ED3	ES1	ES2	ES3	ES2CI	ES2SI	ES2SSI	ES2CRIr	ES2CRIrho	ES2CRItau
								Change in σ							
Δ_{σ_1}	2.15	5.58	7.43	1.76	5.58	6.94	1.39	4.64	6.16	4.36	4.21	4.36	4.28	4.28	4.28
Δ_{σ_2}	2.66	7.15	9.88	2.17	7.15	9.19	1.81	6.14	8.35	5.76	5.61	5.76	5.71	5.71	5.7
Δ_{σ_3}	3.54	10.47	15.72	2.92	10.45	14.43	2.64	9.38	13.54	8.77	8.66	8.76	8.81	8.81	8.81
Δ_{σ_4}	3.95	12.16	19.02	3.27	12.12	17.33	3.04	11.04	16.41	10.29	10.22	10.28	10.41	10.41	10.41
								Change in μ							
Δ_{μ_1}	4.23	12.95	18.58	3.68	12.95	17.24	3.13	9.74	14.35	9.06	8.95	9.06	9.11	9.12	9.12
Δ_{μ_2}	3.6	10.76	15.5	3.04	10.76	14.35	2.57	8.75	12.55	8.17	8.05	8.17	8.19	8.19	8.19
Δ_{μ_3}	3.17	7.85	10.75	2.58	7.79	10.1	2.22	6.74	9.32	6.33	6.18	6.33	6.29	6.29	6.29
Δ_{μ_4}	3.7	8.04	10.13	3.06	7.8	9.79	2.64	6.05	8.45	5.69	5.53	5.68	5.63	5.63	5.63

Table 6.13: Mean Lags (%) of the EVM forecasting methods (training set $\neq$ test set)

6.5.2 Sensitivity experiments

The main experiments judged the performance of the various forecasting methods when the amount of neighbours was changed. Another salient research avenue is to discover the importance of having little or lots of information available. Hence, this section answers the following question: if fewer observations can be used to learn the relation between inputs and outputs, which effect does it have on forecasting accuracy and stability? Section 6.5.2.1 varies the amount of executions from 0.1 to 1.0 in steps of 0.1, while employing all neighbours ($k\% = 1.0$). Section 6.5.2.2 varies the amount of executions and neighbours simultaneously from 0.2 to 1.0 in increments of 0.1.

6.5.2.1 Δexecution

Throughout the main experiment, the training set comprised 800 observations. These resulted from 800 Monte Carlo simulation runs, in which a number for every activity's duration was drawn from a generalized beta distribution. The amount of neighbours was varied, which leads to a reduced training set of more similar observations for the hybrid AI methods. In this section the amount of neighbours is fixed at 100%. As a result, the training set is not reduced because of the inclusion of the k-NN technique. However, we assume that fewer than 800 simulation runs are available. This is a small but crucial nuance. When the number of neighbours was modified in section 6.5.1, the best observations were selected from the training set. In this section, fewer observations are available and no selection of the best found samples takes place.

Figure 6.3 depicts the forecasting accuracy when the training and test sets coincide. Since no AI methods vary much along the scenario, the MAPE was averaged across the Early, Middle and Late scenarios. The x-axis displays the % executions, while the y-axis represents the MAPE. We can infer from figure 6.3 that generally, all AI methods benefit from having more Monte Carlo simulations available in the training set. Hence, the availability of more progress data is beneficial to the forecasting accuracy.

We also analyzed the stability results, as well as the scenarios where the training and test sets differ. The trend of figure 6.3 was absent in both circumstances with the MAPE and Mean Lags displaying a more or less level behaviour as the amount of executions was changed. We believe this can be explained as follows:

- Training set $\neq$ test set: since the observations of the training and test sets are drawn from different distributions, the number of executions does not have an influence on the accuracy of the test set examples. Because there is a lack of representative observations in the training set, it matters little how many observations the training set contains.

- Stability: throughout our experiments, stability is a result of the forecasting methods' estimates. The AI techniques learn to predict the Real Duration and are tuned to optimize forecasting accuracy instead of stability. Hence, there is no logical basis to presume the presence of a relation between the number of executions and the forecasting stability.

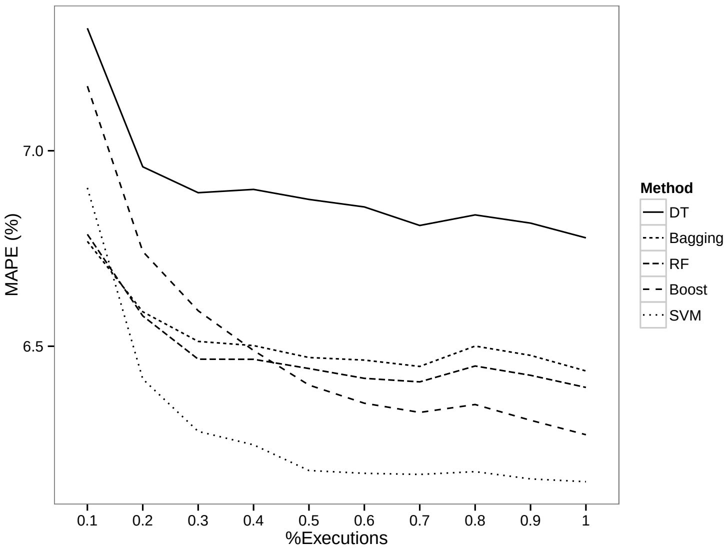


Figure 6.3: Forecasting accuracy for Δ execution

6.5.2.2 Δexecution & Δneighbours

The final computational experiment we conducted varied the amount of executions and neighbours. Both parameters were changed identically, implying that the %execution was always equal to the %neighbours. Since most AI techniques require a minimum number of observations, the minimum value for the executions and neighbours was 0.2. The forecasting accuracy is depicted in figure 6.4 and reveals a similar behaviour to figure 6.3 for all methods except the Decision Tree method.

An explanation can be derived from the earlier results for the DT technique. In section

6.5.1.1 we established that the best results of the DT technique were found for very few neighbours (0.1). However, the sensitivity experiment of section 6.5.2.1 concluded that the lowest MAPE was found when the %execution was equal to 1.0. Combining these two findings leads us to hypothesize that fewer observations in the training set only yield results when these observations are of high quality. Clearly, this is the case when the Nearest Neighbour technique is applied (the best observations result from equation (6.1)). However, when the %execution is varied, these executions are not necessarily the best observations that can be found. In this section, the best results for DT are found when %execution and %neighbour are equal to 0.1. Consequently, it can be conjectured that this is due to the limited amount of neighbours.

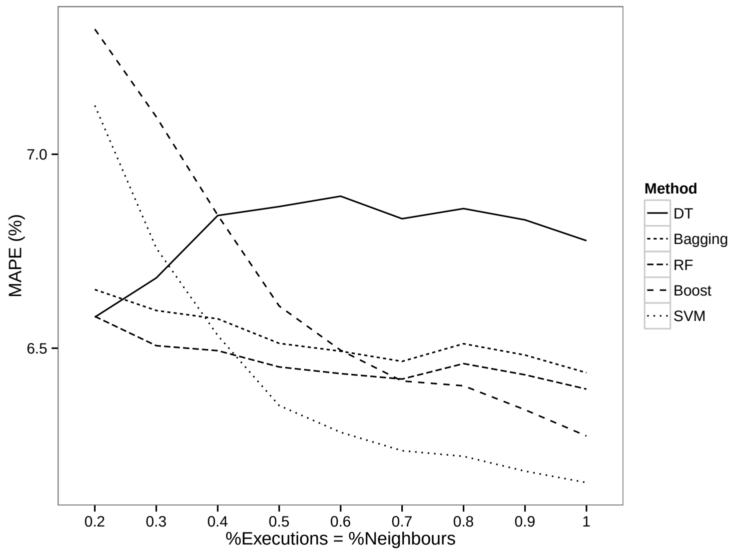


Figure 6.4: Forecasting accuracy for Δ execution and Δ neighbour

6.6 Conclusion

In this chapter, research on project control forecasting was advanced in two ways. On the one hand, we extended previous research by reporting on the stability of Artificial Intelligence methods. On the other hand, a Nearest Neighbour extension for forecasting was proposed. The purpose of this extension was twofold. First, the k-Nearest Neighbour technique was deployed as a predictor and allowed us to benchmark its accuracy and stability with the existing EVM and AI methods. Secondly, Nearest Neighbours were utilized to hybridize the Artificial Intelligence methods by reducing the training set to a smaller training set with more similar observations.

The performance of the forecasting methods was tested on a computer-generated dataset. Deviations from the baseline schedule were generated through activity variations, based on Monte Carlo simulations. These simulations permit the imitation of fictitious project executions by drawing the duration for every activity from a statistical distribution. In this chapter we opted for the generalized beta distribution. Various Earned Value Management performance indicators as well as forecasting information were captured throughout each project's progress. This information served as inputs for the Artificial Intelligence methods. Finally, the forecasting performance is assessed by means of the Mean Absolute Percentage Error for accuracy and the Mean Lags for stability.

The main experiment of this chapter varied the number of neighbours and judged the impact of the (dis)similar nature of the training and test sets. When the training and test set observations result from identical distributions, the AI methods score admirably in terms of accuracy and stability. The main advantage of the Nearest Neighbour method lies in its stable predictions, greatly outperforming the incumbent forecasting methods. Nearest Neighbours were also put to use to hybridize the AI methods, proving particularly advantageous when the training and test sets do not coincide. We observed that a change in the distribution's mean endangers the accuracy more than a change in the standard deviation and that the performance of all AI methods was improved by their hybrid counterparts. Incidentally, the stellar forecasting accuracy of the Nearest Neighbour methods is worth noting.

The experiments were concluded with a number of sensitivity checks, in which the amount of executions and a simultaneous change in the number of executions and neighbours were scrutinized. The former experiment led us to conclude that the AI methods benefit from having the full amount of simulations available. The same conclusion was reached for the latter experiment with the Decision Tree technique as an exception.

It is our belief that future research should focus on two avenues. The first direction stems from a limitation of the present chapter in which fictitious data was employed to assess forecasting accuracy and stability. We call upon researchers and practitioners to validate these findings in real-life projects. While the application of AI techniques requires a minimum of project data in order to function properly, this limitation is not as strict for the presented Nearest Neighbour methods. The second incentive for future research follows directly from the Utopian scenarios. The potential of the Nearest Neighbour methods is vast. If the optimal neighbours can be found, the forecasting accuracy and stability can be greatly improved. Hence, additional research in which different weights are accorded to the various attributes for the distance calculations of equation (6.1) would be an area worth exploring. An alternative option consists of finding different proxies for the Real Duration of the observations of the test set.

6.A Appendix

6.A.1 Illustrative example

In this appendix, the Nearest Neighbour methodology will be explained by means of an illustrative example. The project network that will be employed consists of 10 non-dummy activities and was first introduced by Vanhoucke (2010a). An Activity on the Node (AoN) representation can be found in figure 6.A.1. The duration in days of each activity is indicated above each node while the baseline cost can be found below each node. It is worth mentioning that each activity's cost is entirely variable. Hence, if an activity is delayed during the execution phase, its cost will increase by $\frac{BAC_i}{d_i}$, with BAC_i denoting the Budget At Completion of activity i and d_i denoting the baseline duration of activity i. The Planned Duration (PD) of the project is 16 days and the BAC is equal to €456.

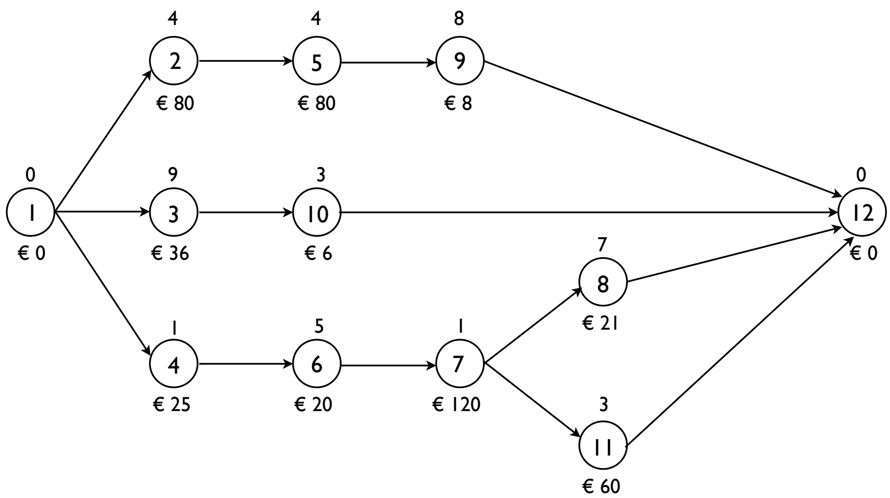


Figure 6.A.1: Illustrative project network (source: Vanhoucke (2010a))

Similar to the methods of Wauters and Vanhoucke (2014a), the k-NN method operates using a training and a test set. The training set results from historical data, which can be obtained using computer simulations or historical data from past projects. In order to keep this example tractable, the training set consists of 10 executions (for which it is assumed that the entire progress and the project's Real Duration (RD) is known). The test set consists of 1 execution. The goal is to forecast the RD of the execution of the test set. In this example we made use of computer simulations to generate the

Ex	Activity 1	2	3	4	5	6	7	8	9	10	RD
1	1	2	1	2	3	1	4	2	1	1	9
2	1	10	1	2	4	1	6	2	2	2	12
3	1	5	1	2	5	1	4	6	2	1	11
4	2	7	1	3	2	1	8	5	1	1	12
5	3	6	1	5	3	1	5	8	2	1	16
6	2	3	1	2	2	1	5	7	2	2	11
7	4	3	1	5	4	1	4	11	2	2	20
8	1	5	1	3	4	1	8	5	2	1	14
9	2	4	1	3	4	1	5	3	1	2	11
10	1	5	1	1	2	1	5	4	1	1	9
11	1	5	1	1	2	1	4	3	2	3	8

Table 6.A.1: Activity and project durations for the 11 executions

executions. Uncertainty was generated by means of the generalized beta distribution of equation (6.4). The parameter settings for this example are $a = 0.1$, $b = 2$, $\mu = 0.6$ and $m = 0.5$. Table 6.A.1 shows the Real Duration of the 10 non-dummy activities as well as the project's RD, which follows from a simple critical path calculation. The executions are abbreviated by "Ex". The first 10 executions comprise the training set while execution 11 makes up the test set. It is worth noting that the Real Durations of the activities only become known as the project progresses and are merely provided for the sake of completeness.

While variability is introduced at the activity level, performance monitoring is performed at the project level, using the Earned Value Management methodology. For this example, two EVM attributes, namely CPI and SPI(t), are employed. The attributes are captured every time period and can be found in table 6.A.2.

Ex	Period 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
										CPI										
1	2.51	2.44	2.33	2.32	1.64	1.75	1.75	1.75	1.75											
2	2.22	2.04	1.96	1.92	1.88	1.44	1.43	1.42	1.42	1.41	1.4	1.4								
3	2.29	2.12	2.04	1.99	1.95	1.91	1.47	1.58	1.59	1.59	1.59									
4	1.43	1.63	1.59	1.28	1.4	1.39	1.38	1.39	1.39	1.38	1.38	1.37								
5	1.18	1.26	1.3	1.24	1.11	1.21	1.19	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.18				
6	1.57	1.81	1.91	1.47	1.47	1.47	1.47	1.47	1.47	1.47	1.47									
7	1.16	1.22	1.25	1.21	1.17	1.07	1.09	1.1	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09
8	2.29	1.96	1.81	1.72	1.71	1.39	1.49	1.49	1.48	1.48	1.47	1.46	1.46	1.45						
9	1.51	1.66	1.61	1.57	1.55	1.32	1.34	1.35	1.35	1.35	1.35									
10	2.29	2.72	2.66	1.7	1.8	1.81	1.8	1.8	1.8											
11	2.29	2.72	2.67	1.7	1.65	1.6	1.56	1.56												
										SPI(t)										
1	3.64	2.98	2.12	1.61	1.58	1.84	1.7	1.65	1.78											
2	3.13	2.43	2.04	1.55	1.26	1.24	1.2	1.22	1.15	1.15	1.17	1.33								
3	3.26	2.54	2.06	1.57	1.27	1.07	1.12	1.35	1.32	1.32	1.45									
4	1.76	1.86	1.74	1.73	1.84	1.59	1.42	1.42	1.35	1.32	1.26	1.33								
5	1.31	1.36	1.37	1.29	1.36	1.41	1.33	1.37	1.31	1.3	1.2	1.13	1.06	1	1	1				
6	2	2.11	2.03	1.91	1.76	1.77	1.64	1.57	1.55	1.49	1.45									
7	1.29	1.3	1.31	1.23	1.16	1.15	1.11	1.19	1.27	1.27	1.17	1.09	1.02	0.96	0.91	0.86	0.82	0.81	0.8	0.8
8	3.26	2.32	2	1.57	1.27	1.29	1.44	1.35	1.28	1.19	1.14	1.1	1.06	1.14						
9	1.89	1.91	1.76	1.54	1.28	1.3	1.28	1.45	1.38	1.34	1.45									
10	3.26	3.07	2.09	1.88	1.98	1.93	1.77	1.68	1.78											
11	3.26	3.07	2.09	1.89	1.7	1.59	1.88	2												

Table 6.A.2: Overview of the EVM attributes (CPI and SPI(t))

Applying the k-NN method commences by calculating the distance between the attributes of every instance of the training set (execution 1 to 10) and the instance of the test set (execution 11). Distance calculation takes place using equation (6.1). The distance values for each period of the test set execution's project duration can be found in table 6.A.3.

A sample calculation for the value indicated in bold will be given. Since we are at time period 2 ($rp = 2$), $P = 2 * 2$ (following equation (6.3)). The distance between the test set observation y and observation 1 of the training set, denoted by x_1, can be calculated as follows:

$$\begin{aligned} ||y - x_1|| &= \sqrt{(2.51 - 2.29)^2 + (3.64 - 3.26)^2 + (2.44 - 2.72)^2 + (2.98 - 3.07)^2} \\ &= \sqrt{0.2793} \approx 0.53 \end{aligned} \tag{6.8}$$

Performing this calculation for all executions across all periods yields a list of distances. In the next step, the Nearest Neighbours are identified. In this example, the 3 ($k = 3$) Nearest Neighbours are used. For period 2, the set of Nearest Neighbours consists of executions 1, 3 and 10 with distances of 0.53, 0.80 and 0.00 respectively. The output value of the test set execution is then computed by taking the average output of the k Nearest Neighbours (equation (6.2)). In a project control forecasting setting, o_i and $\hat{o}$ of equation (6.2) correspond with the RD of execution i and the estimated RD of the test set execution, respectively. Applying equation (6.2) yields the following prediction:

$$\hat{o} = \frac{9 + 11 + 9}{3} \approx 9.67 \tag{6.9}$$

The set of k Nearest Neighbours, as well as the prediction for each period of the test set's execution is provided in the last two rows of table 6.A.3.

Ex	Period							
	1	2	3	4	5	6	7	8
1	0.44	**0.53**	0.63	0.93	0.93	0.98	1.01	1.09
2	0.15	0.95	1.18	1.25	1.35	1.4	1.56	1.75
3	0.00	0.80	1.02	1.11	1.22	1.37	1.57	1.70
4	1.73	2.38	2.63	2.67	2.69	2.69	2.74	2.81
5	2.24	3.18	3.53	3.61	3.67	3.69	3.75	3.82
6	1.45	1.96	2.11	2.12	2.13	2.14	2.15	2.20
7	2.27	3.25	3.63	3.72	3.79	3.85	3.96	4.06
8	0.00	1.07	1.37	1.41	1.48	1.52	1.58	1.71
9	1.58	2.23	2.49	2.52	2.55	2.58	2.66	2.73
10	0.00	0.00	0.01	0.01	0.32	0.51	0.57	0.7
k-NN	{3,8,10}	{10,1,3}	{10,1,3}	{10,1,3}	{10,1,3}	{10,1,3}	{10,1,2}	{10,1,3}
$\hat{o}$	11.33	9.67	9.67	9.67	9.67	9.67	10	9.67

Table 6.A.3: Nearest Neighbour distances and predictions for $k = 3$

7
Conclusions & future research avenues

7.1 General observations

In this chapter, we look back on the work presented in chapters 2 to 6. The research we conducted is situated on the intersection of the Operations Research, Project Management and Data Mining disciplines. As such, the central theme of this book can be phrased as learning from data in a Project Management environment. While synergies between Data Mining and Operations Research were covered by multiple authors, the integration between Data Mining and Project Management is still in its infancy. However, with the recent Big Data hype, it is expected that techniques from statistical learning, Artificial Intelligence, Machine Learning and many related fields will find their way to the field of Project Management. In this respect the research of Smith-Miles et al. (2009) proves particularly interesting. These authors compare two common scheduling heuristics and construct a decision tree to determine which heuristic should be preferred for a certain kind of problem structure.

In general, there is vast potential for integrating Project Management and Data Mining. The link between algorithm performance and problem structure can be facilitated by Data Mining techniques given that sufficient data is available. Selection of the best-performing solution method could be handled by a data mining technique, facilitating the implementation of state-of-the-art techniques in Decision Support Systems. Recent advances on finding algorithm strengths and weaknesses can be found in Branke and Pickardt (2011) and Smith-Miles et al. (2014).

In this book, we presented learning approaches from data for two distinct project management problems, namely the Discrete Time/Cost Trade-off Problem and project control forecasting. In this final chapter, we revisit these problems to identify limitations and attempt to look at future challenges. The outline of this chapter is as follows. For each part of this book, we give a synopsis of how and for which purpose we learned from a project management problem's data. Furthermore, we draw main conclusions, demonstrate limitations and exploit these to draft plans for future research. Table 7.1 summarizes the data, Project Management problems and contributions of each chapter. These aspects will be elucidated in the following sections. Part I, Time/cost optimization, can be found in section 7.2. Section 7.3 deals with Part II, forecasting in a project control environment. The most ambitious challenge for future research is found in the integration of both parts. A separate section (section 7.4) is dedicated to the goals and requirements of this research.

Part	Chapter	Data	Problem	Contribution
Time/cost optimization	Complexity & uncertainty perception (Chapter 2)	Empirical (student) & simulated	DTCTP-D	Solution strategies Contextual factors
	Effort-based decision making (Chapter 3)	Empirical (student) & simulated	DTCTP-D	Effort-based problem insights Validation strategic components
Forecasting	Stability of EVM forecasting (Chapter 4)	Empirical (real-life) & simulated	EVM: time & cost	New stability metric Trade-off stability & accuracy
	Comparison of AI methods (Chapter 5)	Simulated	EVM: time	Introduction 5 AI methods General methodology Performance improvements
	Nearest Neighbour extension (Chapter 6)	Simulated	EVM: time	Stability of AI & NN methods Dual purpose neighbours Utopian scenario

Table 7.1: Research contribution of this dissertation

7.2 Part I: Time/cost optimization

In the first part of this book we studied the deadline variant of the DTCTP (DTCTP-D) by means of a student experiment. Solution strategies were distilled from participants of the Project Scheduling Game (PSG), an IT-supported project management game that focuses on (re)scheduling a project given uncertainty. Real-life data originating from human decisions was analyzed to craft the solution strategies of chapter 2. The manner in which students tackle the DTCTP was translated into general constructs (focus, activity criticality, ranking, intensity and action) to eliminate the need to conduct classroom experiments. Contextual factors, namely complexity and uncertainty, were strongly embedded in the framework of the solution strategies. A large computational experiment delineated the strengths and weaknesses of each solution strategy.
Chapter 3 adopted a slightly different approach. Students have to construct a custom solution to the DTCTP-D subject to an effort restriction. The feedback participants receive is static and only communicates the average effort consumption. As a result, students are forced to think in strategic terms rather than changing dials to optimize costs without a larger picture in mind. We conceived three strategic components, namely schedule focus, activity focus and action radius, each of which may contain a number of elements. This effort-based extension is known as PSG Extended. Two data sources were analyzed. The first source consists of real-life solutions from students having participated in this novel game. The second source was generated by means of a computer and consists of more than 400,000 solution files. Both sources enable us to determine the impact of DTCTP parameters (e.g. the deadline and penalty height) and the influence of the solution strategy components.

Based on the research of Part I, Research Question 1 can be answered as follows:

RQ$_1$: How can research and practice of the DTCTP become more aligned?

- ⋆ Current and future research efforts should take contextual factors into account. Complexity and uncertainty are but two examples that can influence the choice for a certain solution method. Hence, instead of focusing on one-size-fits-all procedures, explicit attention should be allocated to the environment in which each technique performs best.
- ⋆ Secondly, researchers should recognize that Project Managers are constrained in the amount of effort (which can be measured in monetary terms, time or the amount of personnel changes that can be executed) available to them.

Consequently, a search for heuristics and truncated searches should take place in which procedures are able to realize maximum gains given a limited number of allowed changes.

Limitations The main limitation of Part I of this book follows from the generalizations drawn from real-life data. For chapter 2 this entails that certain students may follow variants of the time-based or cost-based solution strategy. While a principal distinction between solution strategies (time versus costs) has been made, it is entirely possible that small groups of students follow a niche strategy that we have yet to identify. Obviously this process will become easier as more data becomes available. In this respect the data growth following from the cooperation with University College of London (UK), Vlerick Business School (Belgium), EDHEC (France) and Ghent University (Belgium) constitutes an excellent prospect. An assumption we made throughout chapter 2 is closely related to these niche strategies. We presupposed that solution strategies are static and do not shift throughout the game. It is possible that certain students could not be categorized because they follow neither a cost-based strategy nor a time-based strategy and change plans as the game proceeds. While chapter 3 also operates on student data, we recognize its limited availability. PSG Extended was first rolled out in January 2014 and has not yet reached the same maturity as the Project Scheduling Game of chapter 2. Hence, there may be discrepancies between the empirical findings and the computational experiment. As we accumulate data, it will be possible to investigate situations in which differences arise and search for the causes of this divergent behaviour.

Future research The first study of Part I reconciled the soft and hard paradigms of project management. Attention was given to the perception of complexity and uncertainty and consequences of judgement errors were highlighted. While these dimensions play a crucial role in project management (cf. Pich et al. (2002)), it is possible to take other factors into account or provide the participants with additional information. We provide three examples:

- Adding information regarding the risk of individual activities may alter the selected course of action. It was shown previously by Vanhoucke and Wauters (2015) that knowledge of risk analysis contributes to time/cost optimization. However, the impact of additional information in terms of activity sensitivity metrics is yet to be tested.
- The appearance of "Black Swan" events (Taleb (2010)) may prompt participants to follow a more risk-averse approach.

- While students have a limited amount of time to complete the PSG, they are not constrained in the expenses they make to bring the project back on track. A fixed contingency budget would be a worthwhile addition and introduces the challenge of dividing the budget strategically across decision moments.

Chapter 2 also assumed that strategies are static. A student follows a certain strategy throughout the game. However, as more data and more strategies are found, it would be interesting to see whether strategy changes occur and by which circumstances these changes are caused (e.g. Black Swan events mentioned above). Both chapters of Part I of this book are deeply rooted in an educational context. As such, future research intentions also involve a more educational orientation. We are currently working on elevating PSG Extended to a higher level of maturity. This includes incorporating it into the curriculum of academic and commercial Project Management programmes, as well as creating a personalized report at the end of the game. The main goal is to turn PSG Extended into a full-fledged business game. The session within a PM curriculum would comprise an introduction, playing the game, a feedback session and handing each participant their custom report. The reader is referred to `www.pmgamecenter.com` for an update on the status of PSG Extended.

7.3 Part II: Forecasting

In the second part of this book, we gained insights into project control forecasting by learning from simulated data. Hence, the Project Management problem and data source differ from those of Part I. In Chapter 4 we shed light on forecasting stability by defining a new criterion that outputs a degree of stability rather than a binary value. Existing forecasting methods were tested by means of a large computational experiment as well as two real-life projects. Throughout the experiments, we reported on stability and accuracy, facilitating a trade-off between both objectives and providing a more nuanced view of the best performing forecasting method.

Large data volumes were employed for chapters 5 and 6. Predictive methods from the field of Artificial Intelligence (AI) were used to make predictions for the final project duration. Well-known techniques such as Monte Carlo simulation, grid search and cross-validation were included in the research methodology. Variation in the durations of activities was modeled with generalized beta distributions. The advantage of implementing this distribution lies in its flexibility in modifying its moments. Dissimilar distributions led to the creation of diverging training and test sets, demonstrating that the AI techniques follow the “garbage-in garbage-out” principle.

Chapter 6 extended previous research by reporting the stability results of the AI methods of chapter 5. The Nearest Neighbour technique achieved admirable stability results and its integration in the hybrid methods greatly improved performance for dissimilar training and test sets. Constructing and tracking the Utopian scenario highlighted project control forecasting's vast potential and is an excellent departure point for future research efforts.

The research done in chapters 4 to 6 allows us to formulate an answer to Research Question 2:

RQ$_2$: How can historical data be leveraged to improve forecasting quality?

- ⋆ Forecasting quality consists of accuracy and stability. When two methods display a similar accuracy, stability can provide an answer as to which method to pick. Stable forecasting methods also yield a consistent warning signal when embedded in a Decision Support System.
- ⋆ The implemented Artificial Intelligence methods show the power of historical data, given that they are sufficiently representative. Additionally, having more data available and combining it with a technique such as Nearest Neighbours can yield distinct advantages. Hence, it makes sense for companies to store the right data in large quantities.

Limitations There are two major limitations to the research of part II of this book. First, many of the proposed techniques have yet to be tested on numerous case studies or real-life projects. The findings of chapters 4 to 6 are almost exclusively based on fictitious projects. Computational tests have the distinct advantage that a wide set of projects with different characteristics can be tested but empirical validation, for instance on the database of Batselier and Vanhoucke (2015), is still required. Implementation of our research in practical environments is straightforward for the stability criterion but will prove much harder for the Artificial Intelligence techniques of chapters 5 and 6. The major hindrance lies in the wealth of data required by the AI methods. Real-life companies that can benefit from these techniques need to possess a certain level of maturity such that they capture relevant project control data and either have a large database of historical project data or a powerful simulation engine at their disposal. These requirements may pose quite the conundrum in practice.

Future research As mentioned in section 7.1, there is vast potential for integrating Data Mining principles into Project Management. In this paragraph, we limit ourselves to applications that extend the research presented in the chapters of Part II. We pinpoint two directions for further scrutiny.

- Forecasting as a trigger for corrective action: while predicting the final duration or cost of a project is a piece of key information for a project manager, project control should be action-driven. In other words, when does a project manager decide to take action based on a prediction of the future? In figure 1.1 of chapter 1, the PM lifecycle was presented. This research would focus on the feedback loop from the control phase to the scheduling phase. As in the research of Colin and Vanhoucke (2014), performance can be measured using true and false positive and negative signals. In addition, an approach similar to Vanhoucke (2011) could be implemented, in which a standard corrective action is applied when the threshold for corrective action is exceeded.

- Classification: throughout this dissertation, the Artificial Intelligence techniques have been applied to the regression problem of forecasting. Another fruitful and thoroughly investigated research branch of Data Mining consists of classification tasks. Classification aims to assign a category to a new instance after a relation has been learned by means of training data. Binary classification and assignment of multi-class labels belong to classification. We briefly propose two ideas to apply classification in a project control context.

  - Classification of failing projects: Artificial Intelligence techniques have been successfully implemented for bankruptcy prediction (Kumar and Ravi (2007)). In order to discern failing and healthy firms, financial ratios are exploited. We make two recommendations to academics who take up the challenge of classifying projects. First, it would be interesting to see whether the attributes we used for prediction (table 5.1 of chapter 5) also lead to an acceptable performance for classification. Some techniques, such as Random Forests, can provide insights into which attributes help explain the distinction between healthy and failing projects. A second recommendation requires preliminary risk research. While risk metrics have been defined on the activity level, few attempts have been made to define risk on the project level. Drawing the parallel with finance literature, these risk metrics then assume an identical role to the financial ratios that distinguish between successful and failing companies. It would be particularly interesting to assess classification performance along the percentage complete. What is the increase in reliability as the project progresses and from which point onwards is classification sufficiently effective?

- Schedule control: this research builds on the work of Colin and Vanhoucke (2014) who constructed tolerance limits to detect unacceptable variation throughout a project. It is possible to train Artificial Intelligence methods to detect the points at which performance exceeds the normal project variation. The main contribution of this research does not consist of training but predicting when a project will exceed a tolerance limit given its progress. It is our belief that such a proactive approach would greatly expand the value of the existing research.

7.4 Integration

The most ambitious idea for future research follows from the combination of Part I and Part II of this dissertation. In general, few studies have considered the integration of project scheduling and control. An exception can be found in Hazır and Schmidt (2013), who model optimal control and combine it with the Discrete Time/Cost Trade-off Problem. The key difference with the idea we will outline lies in the Work Breakdown Structure level. While Hazır and Schmidt (2013) consider individual activities, we follow the rationale of Vanhoucke (2010a) that it is impossible to monitor and control projects by means of individual activities. Hence, we resort to Earned Value Management and its inherent strengths and weaknesses. It is expected that the use of EVM in an integrated scheduling and control context will be less effective for parallel project networks than for more serial networks (Vanhoucke (2010a) and Wauters and Vanhoucke (2014b)).

The major impediment to pursuing this integrated research lies in the fact that preliminary research needs to be done regarding corrective actions. Very few publications deal with Earned Value Management and taking action to bring an endangered project back on track. We call upon researchers to conduct an in-depth analysis of which type of corrective action (e.g. crashing and fast-tracking) performs best, as well as the intensity of the action. For instance, one could wonder whether it is best to crash an activity in an extreme manner or take a more gradual approach in which multiple review periods are utilized to align the project with its baseline schedule. The reader is referred to the literature on match-up scheduling (Akturk and Gorgulu (1999) and Sabuncuoglu and Bayız (2000)) for further inspiration.

In this paragraph, we will explain what an integrated scheduling and control approach should include. The end goal should be to incorporate the managerial insights of this research into a Decision Support System (Hazır (2015)). A graphical outline is provided in figure 7.1. A baseline schedule is either assumed to be given or results from a specific exact or meta-heuristic procedure. The project transitions from the planning to the execution phase and is monitored periodically. A related issue consists of finding the optimal points for project control (Raz and Erel (2000)). Once a performance metric exceeds its lower or upper bound, a trigger for corrective action is issued. The performance bounds may result from the tolerance limits of Colin and Vanhoucke (2014) or the Schedule or Cost Control Index of Pajares and López-Paredes (2011). At this point in time it is necessary to drill down into the Work Breakdown Structure and find out which activity is the culprit for the unacceptable project progress. Actions can be taken on the activity that denotes the worst EVM performance. Incidentally, the project manager may wish to act proactively by modifying future activities based on the sensitivity metric values. Once the activity that causes detrimental performance is found, its duration and associated cost can be modified according to its time/cost profile. In essence, taking corrective action corresponds with finding the optimal mode for a selected activity to bring the project back on track. It is in this phase that multiple characteristics of the DTCTP-D make an appearance. To conclude this section, we provide a number of characteristics or constraints that can influence the process of bringing the project back on track. The reader will find that some of these characteristics are similar to those of chapters 2 and 3 of this book.

- Presence of a deadline and penalty: similar to the PSG, a deadline and penalty can be imposed as an incentive to satisfy the client. We presume that the height of the penalty as well as the relation of the deadline to the project's baseline duration will play an important role in changing activities' durations and costs.

- Activity time/cost profile: convex and concave time/cost profiles can influence the decision to change an activity's duration. Based on the time/cost profile, the project manager may be tempted to choose a more drastic crashing option.

- Topological structure: while we advocate the use of EVM for pragmatic and practical reasons, it implies that the flaws of EVM will be inherent to the integrated scheduling and control model. Project managers should take special care when dealing with project networks with a parallel topological structure.

- Contingency budget: a project manager is typically in charge of a limited contingency budget. Including this as an extra constraint (a monetary restriction can be seen as an exercise in effort-based decision making (cf. chapter 3)) gives rise to several interesting research questions. For example, how does a project manager divide the budget along the percentage complete? At the end of the project, the degrees of freedom have greatly declined and there is little room for changes. However, it is more straightforward to obtain an overview of the impact of corrective actions on the overall project.

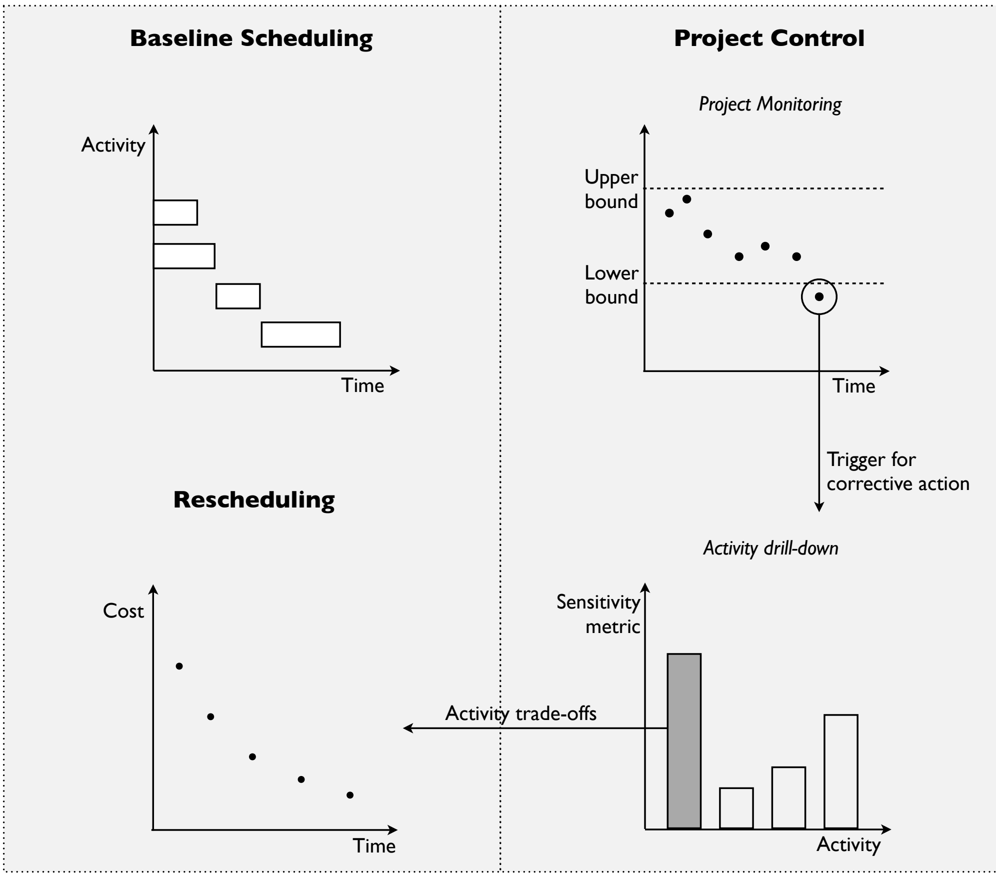


Figure 7.1: Overview of the integration of Part I and Part II of this book

7.5 Closing remarks

In this book we have presented two parts that are connected by the overall theme of learning from data in a project management context. Sections 7.2 and 7.3 highlighted future research avenues. It is our hope that academics and practitioners take up the gauntlet and tackle the challenges we have laid out in those sections. Researchers who do not shy away from a challenge will find great pleasure in developing an integrated scheduling and control approach. The foundation of such experiments can be found in section 7.4. We hold a firm belief in the potential synergies between Project Management and Data Mining and can only hope that others find as much pleasure in the Project Management playground as we have had in these past years.

References

AbouRizk, S., Halpin, D., and Wilson, J. (1994). Fitting beta distributions based on sample data. *Journal of Construction Engineering and Management*, 120:288–305.

Akturk, M. and Gorgulu, E. (1999). Match-up scheduling under a machine breakdown. *European Journal of Operational Research*, 112(1):81–97.

Anbari, F. (2003). Earned value project management method and extensions. *Project Management Journal*, 34(4):12–23.

Azaron, A., Perkgoz, C., and Sakawa, M. (2005). A genetic algorithm approach for the time-cost trade-off in PERT networks. *Applied Mathematics and Computation*, 168:1317–1339.

Azaron, A. and Tavakkoli-Moghaddam, R. (2007). Multi-objective time-cost trade-off in dynamic PERT networks using an interactive approach. *European Journal of Operational Research*, 180:1186–1200.

Barraza, G., Back, E., and Mata, F. (2000). Probabilistic forecasting of project performance using Stochastic S curves. *Journal of Construction Engineering and Management*, 126:142–148.

Barraza, G., Back, E., and Mata, F. (2004). Probabilistic monitoring of project performance using SS-curves. *Journal of Construction Engineering and Management*, 130:25–32.

Batselier, J. and Vanhoucke, M. (2015). Construction and evaluation framework for a real-life project database. *International Journal of Project Management*, 33:697–710.

Bein, W., Kamburowski, J., and Stallmann, M. (1992). Optimal reduction of two-terminal directed acyclic graphs. *Siam Journal on Computing*, 21:1112–1129.

Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D., and Li, S. (2013). *FNN: Fast Nearest Neighbor Search Algorithms and Applications*. R package version 1.1.

Bhatia, N. and Vandana (2010). Survey of nearest neighbor techniques. *International Journal of Computer Science and Information Security*, 8:302–305.

Biau, G., Devroye, L., and Lugosi, G. (2008). Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033.

Book, S. (2006a). Correction note: “Earned Schedule” and its possible unreliability as an indicator. *The Measurable News*, Fall:22–24.

Book, S. (2006b). “Earned Schedule” and its possible unreliability as an indicator. *The Measurable News*, Spring:24–30.

Bourgeois, L. (1985). Strategic goals, perceived uncertainty, and economic performance in volatile environments. *Academy of Management Journal*, 28:548–573.

Bowman, R. A. (2006). Developing activity duration specification limits for effective project control. *European Journal of Operational Research*, 174(2):1191–1204.

Branke, J. and Pickardt, C. (2011). Evolutionary search for difficult problem instances to support the design of job shop dispatching rules. *Computer & Operations Research*, 212:22–32.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and regression trees*. Wadsworth International Group.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., and Pesch, E. (1999). Resource-constrained project scheduling: notation, classification, models, and methods. *European Journal of Operational Research*, 112:3–41.

Bühlmann, P. and Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, 22:477–505.

Bühlmann, P. and Yu, B. (2002). Analyzing bagging. *The Annals of Statistics*, 30:927–961.

Chang, A., Bordia, P., and Duck, J. (2003). Punctuated equilibrium and linear progression: toward a new understanding of group development. *Academy of Management Journal*, 46:106–117.

Chen, S.-P. and Tsai, M.-J. (2011). Time-cost trade-off analysis of project networks in fuzzy environments. *European Journal of Operational Research*, 212:386–397.

Cheng, M.-Y., Peng, H.-S., Wu, Y.-W., and Chen, T.-L. (2010). Estimate at completion for construction projects using evolutionary support vector machine inference model. *Automation in Construction*, 19(5):619–629.

Cheng, M.-Y. and Roy, A. F. (2010). Evolutionary fuzzy decision model for construction management using support vector machine. *Expert Systems with Applications*, 37(8):6061–6069.

Choi, B.-C. and Chung, J. (2014). Complexity results for the linear time-cost tradeoff problem with multiple milestones and completely ordered jobs. *European Journal of Operational Research*, 236:61–68.

Christensen, D. (1993). The estimate at completion problem: A review of three studies. *Project Management Journal*, 24:37–42.

Christensen, D., Antolini, R., and McKinney, J. (1995). A review of estimate at completion (EAC) research. *Journal of Cost Analysis and Management*, Spring:41–62.

Christensen, D. and Heise, S. (1993). Cost performance index stability. *National Contract Management Journal*, 25:7–15.

Christensen, D. and Payne, K. (1992). Cost performance index stability - Fact or fiction? *Journal of Parametrics*, 10:27–40.

Christensen, D. and Templin, C. (2002). EAC evaluation methods: Do they still work? *Acquisition Review Quarterly*, 9:105–116.

Cohen, I., Golany, B., and Shtub, A. (2007). The stochastic time-cost tradeoff problem: a robust optimization approach. *Networks*, 49:175–188.

Colin, J., Martens, A., Vanhoucke, M., and Wauters, M. (2015). A multivariate approach to top-down project control using Earned Value Management. *Decision Support Systems*, 79:65–76.

Colin, J. and Vanhoucke, M. (2014). Setting tolerance limits for statistical project control using earned value management. *Omega The International Journal of Management Science*, 49:107–122.

Corne, D., Dhaenens, C., and Jourdan, L. (2012). Synergies between operations research and data mining: The emerging use of multi-objective approaches. *European Journal of Operational Research*, 221:469–479.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27.

Crawford, L., Pollack, J., and England, D. (2006). Uncovering the trends in project management: Journal emphases over the last 10 years. *International Journal of Project Management*, 24:175–184.

Crowston, W. and Thompson, G. (1967). Decision CPM: A method for simultaneous planning, scheduling and control of projects. *Operations Research*, 15:407–426.

De, P., Dunne, E., Ghosh, J., and Wells, C. (1995). The discrete time-cost tradeoff problem revisited. *European Journal of Operational Research*, 81:225–238.

De, P., Dunne, E., Ghosh, J., and Wells, C. (1997). Complexity of the discrete time/cost trade-off problem for project networks. *Operations Research*, 45:302–306.

De Reyck, B. and Herroelen, W. (1996). On the use of the complexity index as a measure of complexity in activity networks. *European Journal of Operational Research*, 91:347–366.

Demeulemeester, E., De Reyck, B., Foubert, B., Herroelen, W., and Vanhoucke, M. (1998). New computational results on the discrete time/cost trade-off problem in project networks. *Journal of the Operational Research Society*, 49:1153–1163.

Demeulemeester, E., Elmaghraby, S., and Herroelen, W. (1996). Optimal procedures for the discrete time/cost trade-off problem in project networks. *European Journal of Operational Research*, 88:50–68.

Demeulemeester, E., Vanhoucke, M., and Herroelen, W. (2003). Rangen: A random network generator for activity-on-the-node networks. *Journal of Scheduling*, 6:17–38.

Dietterich, T. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40:139–157.

Dvir, D., Sadeh, A., and Malach-Pines, A. (2006). Projects and project managers: the relationship between project manager's personality, project, project types, and project success. *Project Management Journal*, 37(5):36–48.

El-Rayes, K. and Kandil, A. (2005). Time-Cost-Quality Trade-Off Analysis for Highway Construction. *Journal of Construction Engineering and Management*, 131(4):477–486.

Elith, J., Leathwick, J., and Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Biology*, 77:802–813.

Elmaghraby, S. (1977). *Activity networks: Project planning and control by network models*. New York: John Wiley and Sons, Inc.

Elshaer, R. (2013). Impact of sensitivity information on the prediction of project's duration using earned schedule method. *International Journal of Project Management*, 31:579–588.

Engwall, M. and Westling, G. (2004). Peripety in an r&d drama: capturing a turnaround in project dynamics. *Organization Studies*, 25:1557–1578.

Fang, C. and Marle, F. (2012). A simulation-based risk network model for decision support in project risk management. *Decision Support Systems*, 52:635–644.

Faria, A. (2001). The changing nature of business simulation/gaming research: A brief history. *Simulation & Gaming*, 32:97–110.

Fazar, W. (1959). Program evaluation and review technique. *The American Statistician*, 13:10.

Fleming, Q. and Koppelman, J. (2005). *Earned value project management. 3rd Edition.* Newtown Square, PA: Project Management Institute.

Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29:1189–1232.

Fulkerson, D. (1961). A network flow computation for project cost curves. *Management Science*, 7:167–178.

Geraldi, J. and Adlbrecht, G. (2007). On faith, fact and interaction in projects. *Project Management Journal*, 38(1):32–43.

Geraldi, J., Maylor, H., and Williams, T. (2011). Now, let's make it really complex (complicated). *International Journal of Operations & Production Management*, 31(9):966–990.

Ghoddousi, P., Eshtehardian, E., Jooybanpour, S., and Javanmardi, A. (2013). Multi-mode resource-constrained discrete time-cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm. *Automation in Construction*, 30:216–227.

Gigerenzer, G. and Gaissmaier, W. (2011). Heuristic decision-making. *Annual Review of Psychology*, 62:451–482.

Govindarajan, M. and Chandrasekaran, R. (2010). Evaluation of k-nearest neighbour classifier performance for direct marketing. *Expert Systems with Applications*, 30:253–258.

Green, G. (2004). The impact of cognitive complexity on project leadership performance. *Information and Software Technology*, 46:165–172.

Haapasalo, H. and Hyvönen, J. (2001). Simulating businesss and operations management - a learning environment for the electronics industry. *International Journal of Production Economics*, 73:261–272.

Hall, N. (2012). Project management: recent developments and research opportunities. *Journal of Systems Science and Systems Engineering*, 21(2):129–143.

Hanisch, B. and Wald, A. (2011). A project management research framework integrating multiple theoretical perspectives and influencing factors. *Project Management Journal*, 42(3):4–22.

Hartmann, S. and Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207:1–15.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning*. Springer.

Hazır, O. (2015). A review of analytical models, approaches and decision support tools in project monitoring and control. *International Journal of Project Management*, 33:808–815.

Hazır, O., Erel, E., and Günalay, Y. (2011). Robust optimization models for the discrete time/cost trade-off problem. *International Journal of Production Economics*, 130:87–95.

Hazır, O., Haouari, M., and Erel, E. (2010). Robust scheduling and robustness measures for the discrete time/cost trade-off problem. *European Journal of Operations Research*, 207:633–643.

Hazır, O. and Schmidt, K. (2013). An integrated scheduling and control model for multi-mode projects. *Flexible Services and Manufacturing Journal*, 25:230–254.

Henderson, K. and Zwikael, O. (2008). Does project performance stability exist? A re-examination of CPI and evaluation of SPI(t) stability? *Crosstalk - The Journal of Defense Software Engineering*, 21:7–13.

Henley, W. and Hand, D. (1996). A k-nearest neighbour classifier for assessing consumer credit risk. *The statistician*, 45:77–95.

Herroelen, W., De Reyck, B., and Demeulemeester, E. (1998). Resource-constrained project scheduling: A survey of recent developments. *Computers & Operations Research*, 25:279–302.

Hillier, F. and Lieberman, G. (2005). *Introduction to operations research*. McGraw-Hill.

Hindelang, T. and Muth, j. (1979). A dynamic programming algorithm for decision CPM networks. *Operations Research*, 27:225–241.

Hothorn, T., Buhlmann, P., Dudoit, S., Molinaro, A., and Van Der Laan, M. (2006a). Survival ensembles. *Biostatistics*, 7:355–373.

Hothorn, T., Hornik, K., and Zeileis, A. (2006b). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15:651–674.

Howell, D., Windahl, C., and Seidel, R. (2010). A project contingency framework based on uncertainty and its consequences. *International Journal of Project Management*, 28:256–264.

Jaafari, A. (2003). Project management in the age of complexity and change. *Project Management Journal*, 34(4):47–57.

Jacob, D. and Kane, M. (2004). Forecasting schedule completion using earned value metrics? Revisited. *The Measurable News*, Summer:1, 11–17.

Johnson, D. (1997). The triangular distribution as a proxy for the beta distribution in risk analysis. *The Statistician*, 46:387–398.

Jolliffe, I. (2005). *Principal component analysis*. Wiley Online Library.

Ke, H., Ma, W., and Ni, Y. (2009). Optimization models and a GA-based algorithm for stochastic time-cost trade-off problem. *Applied Mathematics and Computation*, 215:308–313.

Ke, H., Ma, W., and Ni, Y. (2012). Modeling stoachastic project time-cost trade-offs with time-dependent activity durations. *Applied Mathematics and Computation*, 218:9462–9469.

Kelley, J. (1961). Critical path planning and scheduling: Mathematical basis. *Operations Research*, 9:296–320.

Kelley, J. and Walker, M. (1959). *Critical path planning and scheduling: An introduction*. Mauchly Associates, Ambler, PA.

Kim, B. and Reinschmidt, K. (2009). Probabilistic forecasting of project duration using bayesian inference and the beta distribution. *Journal of Construction Engineering and Management*, 135:178–186.

Kim, B. and Reinschmidt, K. (2010). Probabilistic forecasting of project duration using kalman filter and the earned value method. *Journal of Construction Engineering and Management*, 136:834–843.

Klerides, E. and Hadjiconstantinou, E. (2010). A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations. *Computers & Operations Research*, 37:2131–2140.

Kolb, D. (1984). *Experiential Learning: Experience as The Source of Learning and Developent*. Prentice-Hall.

Kuhl, M. E., Lada, E. K., Steiger, N. M., Wagner, M. A., and Wilson, J. R. (2007). Introduction to modeling and generating probabilistic input processes for simulation. In Henderson, S., Biller, B., Hsieh, M., Shortle, J., Tew, J., and Barton, R., editors, *Proceedings of the 2007 Winter Simulation Conference*, pages 63–76. New Jersey: Institute of Electrical and Electronics Engineers.

Kumar, P. and Ravi, V. (2007). Bankruptcy prediction in banks and firms via statistical and intelligent techniques - a review. *European Journal of Operational Research*, 180:1–28.

Kwak, Y. and Ingall, L. (2007). Exploring monte carlo simulation applications for project management. *Risk Management*, 9:44–57.

Lee, Y., Bhaskaran, K., and Pinedo, M. (1997). A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Transactions*, 29:45–52.

Lin, Y. and Jeon, Y. (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101:578–590.

Lipke, W. (2003). Achieving normality for cost. *The Measurable News*, Fall/Winter:5–11.

Mangasarian, O. (1965). Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13(3):444–452.

Mastor, A. (1970). An experimental and comparative evaluation of production line balancing techniques. *Management Science*, 16:728–746.

Maylor, H., Vidgen, R., and Carver, S. (2008). Managerial complexity in project-based operations: a grounded model and its implications for practice. *Project Management Journal*, 39:S15–S26.

Meisel, S. and Mattfeld, D. (2010). Synergies of operations research and data mining. *European Journal of Operational Research*, 206:1–10.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2012). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6-1.

Meyer, W. and Shaffer, L. (1965). Extending CPM for multiform project time-cost curves. *Journal of the Construction Division, Proceedings of the ASCE*, 91:45–65.

Mingers, J. (1987). Expert systems - rule induction with statistical data. *Journal of the Operational Research Society*, 38:39–47.

Mintzberg, H., Raisinghani, D., and Theoret, A. (1976). The structure of unstructured decision processes. *Administrative Science Quarterly*, 21:246–275.

Moder, J., Phillips, C., and Davis, E. (1983). *Project Management with CPM, PERT and Precedence Diagramming*. Nostrand Reinhold, New York.

Mokhtari, H., Kazemzadeh, R., and Salmasnia, A. (2011). Time-cost tradeoff analysis in project management: an ant system approach. *IEEE Transactions on Engineering Management*, 58:36–43.

Morgan, J. and Sonquist, J. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58:415–434.

Müller, R. and Turner, J. (2007). Matching the project manager's leadership style to project type. *International Journal of Project Management*, 25:21–32.

Nassar, K., Gunnarsson, H., and Hegab, M. (2005). Using weibull analysis for evaluation of cost and schedule performance. *Journal of Construction Engineering and Management*, 131:1257–1262.

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurobiotics*, 7:1–21.

Nikolopoulos, K., Goodwin, P., Patelis, A., and Assimakopoulos, V. (2007). Forecasting with cue information: A comparison of multiple regression with alternative forecasting approaches. *European Journal of Operational Research*, 180:354–368.

Nutt, P. (1998). How decision makers evaluate alternatives and the influence of complexity. *Management Science*, 44(8):1148–1166.

Ojiako, U., Chipulu, M., Gardiner, P., Williams, T., Mota, C., Maguire, S., Shou, Y., and Stamati, T. (2014). Effect of project role, age and gender differences on the formation and revision of project decision judgements. *International Journal of Project Management*, 32:556–567.

Olafsson, S., Li, X., and Wu, S. (2008). Operations research and data mining. *European Journal of Operational Research*, 187(3):1429–1448.

Osman, M. (2010). Controlling uncertainty: a review of human behaviour in complex dynamic environments. *Psychological Bulletin*, 136(1):65–86.

Pajares, J. and López-Paredes, A. (2011). An extension of the EVM analysis for project monitoring: The cost control index and the schedule control index. *International Journal of Project Management*, 29:615–621.

Parente, D., Stephan, J., and Brown, R. (2012). Facilitating the acquisition of strategic skills: the role of traditional and soft managerial skills. *Management Research Review*, 35:1004–1028.

Pascoe, T. (1966). Allocation of resources - CPM. *Revue Française de Recherche Opérationnelle*, 38:31–38.

Payne, K. (1990). An investigation of the stability of the cost performance index. Technical report, Air Force Institute of Technology.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Perminova, O., Gustafsson, M., and Wikström, K. (2008). Defining uncertainty in projects - a new perspective. *International Journal of Project Management*, 26:73–79.

Phillips, S. and Dessouky, M. (1977). Solving the project time/cost tradeoff problem using the minimal cut concept. *Management Science*, 24:393–400.

Pich, M., Loch, C., and De Meyer, A. (2002). On uncertainty, ambiguity, and complexity in project management. *Management Science*, 48:1008–1023.

PMBOK (2004). *A Guide to the Project Management Body of Knowledge, Third Edition.* Newtown Square, Pa.: Project Management Institute, Inc.

Pollack, J. (2007). The changing paradigms of project management. *International Journal of Project Management*, 25:266–274.

Pour, N., Modarres, M., Aryanejad, M., and Moghadam, R. (2010). The discrete time-cost-quality trade-off problem using a novel hybrid genetic algorithm. *Applied Mathematical Sciences*, 42:2081–2094.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers Inc.

R Core Team (2013). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Raz, T. and Erel, E. (2000). Optimal timing of project control points. *European Journal of Operational Research*, 127(2):252–261.

Ridgeway, G. (2013). *gbm: Generalized Boosted Regression Models.* R package version 2.1.

Robinson, D. (1975). A dynamic programming solution to cost/time trade-off for CPM. *Management Science*, 22:158–166.

Rujirayanyong, T. (2009). A comparison of three completion date predicting methods for construction projects. *Journal of Research in Engineering and Technology*, 6:305–318.

Sabuncuoglu, I. and Bayız, M. (2000). Analysis of reactive scheduling problems in a job shop environment. *European Journal of Operational Research*, 126(3):567–586.

Saunders, P. (1997). Experiential learning, cases, and simulations in business communication. *Business Communication Quarterly*, 60:97–114.

Schapire, R. (1990). The strength of weak learnability. *Machine Learning*, 5:197–227.

Schapire, R. (2003). The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification*, pages 149–171. Springer.

Schmid, B. and Adams, J. (2008). Motivation in project management: The project manager's perspective. *Project Management Journal*, 39:60–71.

Schnaars, S. (1984). Situational factors affecting forecast accuracy. *Journal of Marketing Research*, 21:290–297.

Schumann, P., Anderson, P., and Scott, T. (1997). Using computer-based simulation exercises to teach business ethics. *Teaching Business Ethics*, 1:163–181.

Segal, M. (2004). Machine learning benchmarks and random forest regression. Technical report, Center for Bioinformatics & Molecular Biostatistics papers, University of California, San Francisco, CA, USA.

Shenhar, A. (2001). One size does not fit all projects: exploring classical contingency domains. *Management Science*, 47(3):394–414.

Shlens, J. (2005). A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego.*

Smith-Miles, K., Baatar, D., Wreford, B., and Lewis, R. (2014). Towards objective measures of algorithm performance across instance space. *Computer & Operations Research*, 45:12–24.

Smith-Miles, K., James, R., Giffin, J., and Tu, Y. (2009). A knowledge discovery approach to understanding relationships between scheduling problem structure and heuristic performance. In *Learning and intelligent optimization*, pages 89–103. Springer.

Smola, A. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222.

Söderlund, J. (2011). Pluralism in project management: Navigating the crossroads of specialization and fragmentation. *International Journal of Management Reviews*, 13:153–176.

Sommer, S. and Loch, C. (2004). Selectionism and learning in projects with complexity and unforeseeable uncertainty. *Management Science*, 50:1334–1347.

Sonmez, R. and Bettemir, O. (2012). A hybrid genetic algorithm for the discrete time-cost trade-off problem. *Expert Systems with Applications*, 39:11428–11434.

Strasser, H. and Weber, C. (1999). On the asymptotic theory of permutation statistics. *Mathematical Methods of Statistics*, 8:220–250.

Stumpf, S., Dunbar, R., and Mullen, T. (1991). Systems thinking and business process redesign: an application to the beer game. *Journal of Management Development*, 10(5):32–45.

Taleb, N. (2010). *The black swan: The impact of the Highly Improbable*. Random House.

Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introduction to data mining*. Pearson Addison Wesley Boston.

Tareghian, H. and Taheri, S. (2006). On the discrete time, cost and quality trade-off problem. *Applied Mathematics and Computation*, 181:1305–1312.

Tareghian, H. and Taheri, S. (2007). A solution procedure for the discrete time, cost and quality tradeoff problem using electromagnetic scatter search. *Applied Mathematics and Computation*, 190:1136–1145.

Tavares, L. (2002). A review of the contribution of operational research to project management. *European Journal of Operational Research*, 136:1–18.

Tavares, L., Ferreira, J., and Coelho, J. (1999). The risk of delay of a project in terms of the morphology of its network. *European Journal of Operational Research*, 119:510–537.

Tavares, L. and Weglarz, J. (1990). Project management and scheduling: A permanent challenge for OR. *European Journal of Operational Research*, 49:1–2.

Thomas, J. and Mengel, T. (2008). Preparing project managers to deal with complexity - Advanced project management education. *International Journal of Project Management*, 26:304–315.

Turner, J. and Müller, R. (2005). The project manager's leadership style as a success factor on projects: a literature review. *Project Management Journal*, 36(1):49–61.

Uyttewael, E. (2005). *Dynamic Scheduling with Microsoft Office Project 2003: the book by and for professionals*. Co-published with International Institute for Learning Inc, Boca Raton.

Van Ackere, A., Larsen, E., and Morecroft, J. (1993). Systems thinking and business process redesign: an application to the beer game. *European Management Journal*, 11:412–423.

Vandevoorde, S. and Vanhoucke, M. (2006). A comparison of different project duration forecasting methods using earned value metrics. *International Journal of Project Management*, 24:289–302.

Vanhoucke, M. (2005). New computational results for the discrete time/cost trade-off problem with time-switch constraints. *European Journal of Operational Research*, 165:359–374.

Vanhoucke, M. (2010a). *Measuring Time - Improving Project Performance using Earned Value Management*, volume 136 of *International Series in Operations Research and Management Science*. Springer.

Vanhoucke, M. (2010b). Using activity sensitivity and network topology information to monitor project time performance. *Omega The International Journal of Management Science*, 38:359–370.

Vanhoucke, M. (2011). On the dynamic use of project performance and schedule risk information during project tracking. *Omega The International Journal of Management Science*, 39:416–426.

Vanhoucke, M. (2012a). Measuring the efficiency of project control using fictitious and empirical project data. *International Journal of Project Management*, 30:252–263.

Vanhoucke, M. (2012b). *Project Management with Dynamic Scheduling: Baseline Scheduling, Risk Analysis and Project Control*, volume XVIII. Springer.

Vanhoucke, M. (2013). *The Art of Project Management: A Story about Work and Passion*. Available online at www.or-as.be.

Vanhoucke, M. (2014). *Integrated Project Management and Control: First come the theory, then the practice*. Management for Professionals. Springer.

Vanhoucke, M. (2015). On the use of schedule risk analysis for project management. *The Journal of Modern Project Management*, 2(3):108–117.

Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., and Tavares, L. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research*, 187:511–524.

Vanhoucke, M. and Debels, D. (2007). The discrete time/cost trade-off problem: Extensions and heuristic procedures. *Journal of Scheduling*, 10:311–326.

Vanhoucke, M., Demeulemeester, E., and Herroelen, W. (2002). Discrete time/cost trade-offs in project scheduling with time-switch constraints. *Journal of the Operational Research Society*, 53:741–751.

Vanhoucke, M. and Vandevoorde, S. (2007). A simulation and evaluation of earned value metrics to forecast the project duration. *Journal of the Operational Research Society*, 58:1361–1374.

Vanhoucke, M. and Vandevoorde, S. (2008). Earned value forecast accuracy and activity criticality. *The Measurable News*, Summer:13–16.

Vanhoucke, M. and Vandevoorde, S. (2009). Forecasting a project's duration under various topological structures. *The Measurable News*, Spring:26–30.

Vanhoucke, M., Vereecke, A., and Gemmel, P. (2005). The project scheduling game (PSG): Simulating time/cost trade-offs in projects. *Project Management Journal*, 51:51–59.

Vanhoucke, M. and Wauters, M. (2015). Blended learning in project management - experiences on business games and case studies. In *Proceedings of the* 7^{th} *International Conference on Computer-Supported Education*, volume To Appear.

Vapnik, V. (2000). *The nature of statistical learning theory.* Springer.

Vapnik, V. and Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780.

Vidal, L.-A. and Marle, F. (2008). Understanding project complexity: implications on project management. *Kybernetes*, 37(8):1094–1110.

Walker, M. and Sawyer, J. (1959). Project Planning and Scheduling. Technical Report Report 6959, E.I. duPont de Nemours and Co., Wilmington, Delaware.

Wan, C., Lee, L., Rajkumar, R., and Isa, D. (2012). A hybrid text classification approach with low dependency on parameter by integrating k-nearest neighbor and support vector machine. *Expert Systems with Applications*, 39:11880–11888.

Ward, S. and Chapman, C. (2003). Transforming project risk management into project uncertainty management. *International Journal of Project Management*, 21:97–105.

Wauters, M. and Vanhoucke, M. (2013). A study on complexity and uncertainty perception and solution strategies for the time/cost trade-off problem. *Working Paper at Ghent University, currently under journal submission (Check www.projectmanagement.ugent.be for an update).*

Wauters, M. and Vanhoucke, M. (2014a). A comparative study of artificial intelligence methods for project duration forecasting. *Submitted to an international journal (working paper version available at www.projectmanagement.ugent.be/sites/default/files/files/wp/WautersVanhoucke2014a.pdf).*

Wauters, M. and Vanhoucke, M. (2014b). Support vector machine regression for project control forecasting. *Automation in Construction*, 47:92–106.

Wauters, M. and Vanhoucke, M. (2015). Study of the stability of earned value management forecasting. *Journal of Construction Engineering and Management*, 141(4):1–10.

White, A. and Liu, W. (1994). Bias in information-based measures in decision tree induction. *Machine Learning*, 15:321–329.

White, D. and Fortune, J. (2002). Current practice in project management - an empirical study. *International Journal of Project Management*, 20:1–11.

Willems, L. and Vanhoucke, M. (2015). Classification of articles and journals on project control and earned value management. *International Journal of Project Management*, 33:1610–1634.

Wolfe, J. (1996). The concept of project complexity - a review. *International Journal of Project Management*, 14(4):201–204.

Wolfe, J. (1997). The effectiveness of business games in strategic management course work. *Simulation & Gaming*, 28:360–376.

Wu, X., Kumar, V., Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Philip, S., Zhou, Z.-H., Steinbach, M., Hand, D., and Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37.

Yang, L.-R., Huang, C.-F., and Wu, K.-S. (2011). The association among project manager's leadership style, teamwork and project success. *International Journal of Project Management*, 29:258–267.

Zhang, H. and Xing, F. (2010). Fuzzy-multi-objective particle swarm optimization for time-cost-quality tradeoff in construction. *Automation in Construction*, 19(8):1067–1075.

Zwikael, O., Globerson, S., and Tzvi, R. (2000). Evaluation of models for forecasting the final cost of a project. *Project Management Journal*, 31:53–57.