Who Should Own Float? Mitigating Delays by Float Pre-Allocation

by

Gunnar Lucko
Yi Su

Construction CPM Conference
February 2, 2016

THE CATHOLIC UNIVERSITY OF AMERICA

Overview

- Introduction
- Political Apportionment Methods
  - Largest remainder method
  - Divisor methods
- Methodology
  - Float pre-allocation model
  - Simulation process
- Case Study
- Conclusions

Introduction

- Construction projects often incur delays
  - Delays: “an act or event that extends the time required to perform tasks under a contract” (Stumpf 2000, p. 32)
  - Most crucial: critical path activities delay

- General approach: project buffer on critical chain
  - Single large end buffer, without tracking by whom
  - Not properly protect project (Herrorlen and Leus 2001)

- Classical research question: who owns float?
- How can available project float be used most beneficially by allocating it to activities who by definition have none and determining how much each receives?

Introduction

- Similarities
  - Political apportionment :
  how to divide integer number representatives according to respective sizes (Verba 2003)

  - Float pre-allocation

Table 1: Analogies between Political Apportionment and Float Preallocation

Similarities	Float Preallocation	Political Apportionment
Why apportionment (Goal/motivation)	Fairness for distributing float	"[E]qual influence over government policy across citizens" (Verba 2003, p. 663)
What is equalizing (subject)	Float/time/ability of excusable delay when making schedule	Seats/power of voting when making policy
How to apportion	Rounding, because schedule is in integer days	Rounding, because no fractional seats are allowed

Need for Research

- Inspired by political apportionment approaches, this paper will develop analogous approach to distribute project float to every critical activity. three **Research Objectives** are set:
  - Reviewing different rounding methods that are used in political voting and extracting their pertinent features
  - Creating a methodology to integrate apportionment methods and their rounding into pre-allocating project float
  - Conducting a simulation of a sufficiently complex schedule to validate these methods and compare their results

Political Apportionment Methods

- Largest remainder method

- Divisor methods:

require select divisor based on generalized power mean for specific parameter values (Jones and Wilson 2010)

- Greatest divisor
- Smallest divisor
- Geometric mean
- Harmonic mean
- Arithmetic mean

Political Apportionment Methods

- Example: 45 seats, four parties

votes: A (61), B (48), C(20), D (7)

- Largest remainder method
  - Calculate proportion of each party, e.g. A: 61/(61+48+20+7)=44.85%
  - Get unrounded seat (quota), A: 44.85% × 45 = 20.18
  - Round down to integer, A(20), B(15), C(6), D(2), total 43
  - Leftover 2 seats are given to the parties with largest remainders, B and C
  - Final apportionment: A(20), B(16), C(7), D(2)
  - '*Alabama Paradox*': increasing seats in the house "has caused the smallest party to lose a seat" (Wiseman 2015)

Political Apportionment Methods

- Divisor methods
  - Greatest divisor: quota / divisor, round down
  - Smallest divisor: quota / divisor, round up
  - Geometric mean: round down quota (q), geometric mean $(q \times (q+1))^{1/2}$, judge: if $q < GM$, round up GM; if $q > GM$, round down GM
  - Harmonic mean: employs harmonic mean
  - Arithmetic mean: employs arithmetic mean
  - Divisor must be iteratively adjusted until total seat reached

Table 2: Results of Distributing Seats for Four Parties with Different Apportionment Methods

	Target Proportion						Quota Unrounded				Seats			
Method Name	A	B	C	D	**Total Seats**	**Divisor**	A	B	C	D	A	B	C	D
Largest Remainder	61	48	20	7	45	N/A	20.18	15.88	6.62	2.32	20	16	7	2
Greatest Divisor						2.90	21.03	16.55	6.90	2.41	21	16	6	2
Smallest Divisor						3.20	19.06	15.00	6.25	2.19	20	15	7	3
Geometric Mean						3.02	20.49	15.49	6.48	2.45	20	16	7	2
Harmonic Mean						3.02	20.49	15.48	6.46	2.40	20	16	7	2
Arithmetic Mean						3.07	19.87	15.64	6.51	2.28	20	16	7	2

Methodology

- Float Pre-allocation Model
- *n* critical activities with fixed duration *Di* ,use it as base
- Exponents, 0, 0.5, 1 cases to get quotas accordingly
- Apply different apportionment method returns pre-allocation

(a_1, …, a_n), where $\sum_i a_i = FLOAT$

Table 3: Structure of Float Arrangement Model based on Apportionment Methods

Exponent of Duration	Target Proportions (Quotas)					Float to Apportion	Apportionment Method	Float Apportionment
	Act 1	Act 2	Act 3	…	Act n			
0.0	1	1	1	…	1			
0.5	$D_1^{0.5}$	$D_2^{0.5}$	$D_2^{0.5}$	…	$D_n^{0.5}$	FLOAT	LR, GD, SD, GM, HM, AM	$(a_1, \ldots, a_n)$
1.0	D_1	D_2	D_2	…	D_n			

Note: Different rounding approaches are already included in the apportionment methods.

Methodology

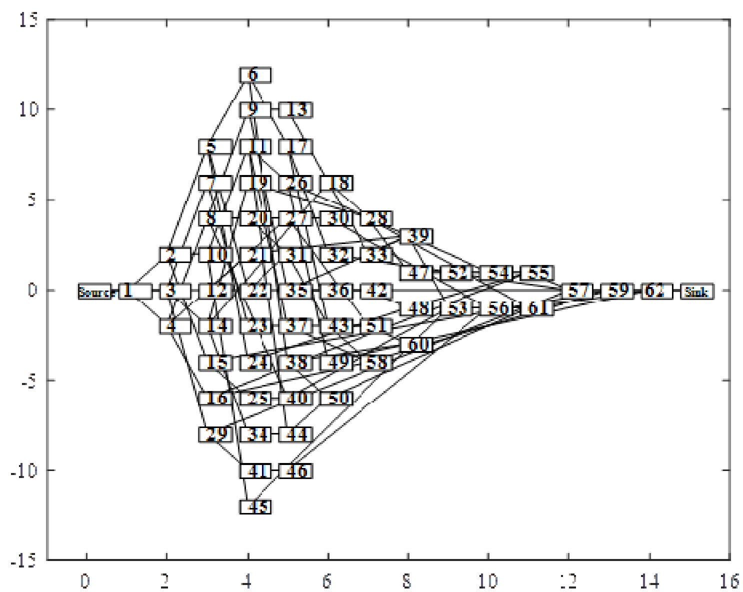
- Simulation Process
  - *n* critical activities with random duration *di*
  - Compare: rounded-up integer delay $\lceil d_i - D_i \rceil$ and arranged float a_i
  - Overruns: discrete count and continuous measurement of delay
  - Goal: best exponent case and apportionment method to stop overrunning ***the fastest***

Table 4: Structure of Simulation Process

Run	Simulation					Float Arrangement	Judgement (Discrete)	Judgement (Continuous)
	Act 1	Act 2	Act 3	…	Act n			
1	d_{11}	d_{21}	d_{31}	…	d_{n1}	$(a_1, \ldots, a_n)$	if $\lceil d_i - D_i \rceil > a_i$, then count one overrun event	if $\lceil d_i - D_i \rceil > a_i$, then calculate overrun period
2	d_{12}	d_{22}	d_{32}	…	d_{n2}			
…	…	…	…	…	…			
N	d_{1N}	d_{2N}	d_{3N}	…	d_{nN}			

Case Study

- J60_1 schedule data from Project Scheduling Problem Library (PSPLIB)
- Duration (triangular distribution lower 90% and 150% mode)
- Fixed duration 77 days, 23 critical activities



Case Study

- Simulation with Largest Remainders Method
  - 100 runs
  - ***exponent 0.5***

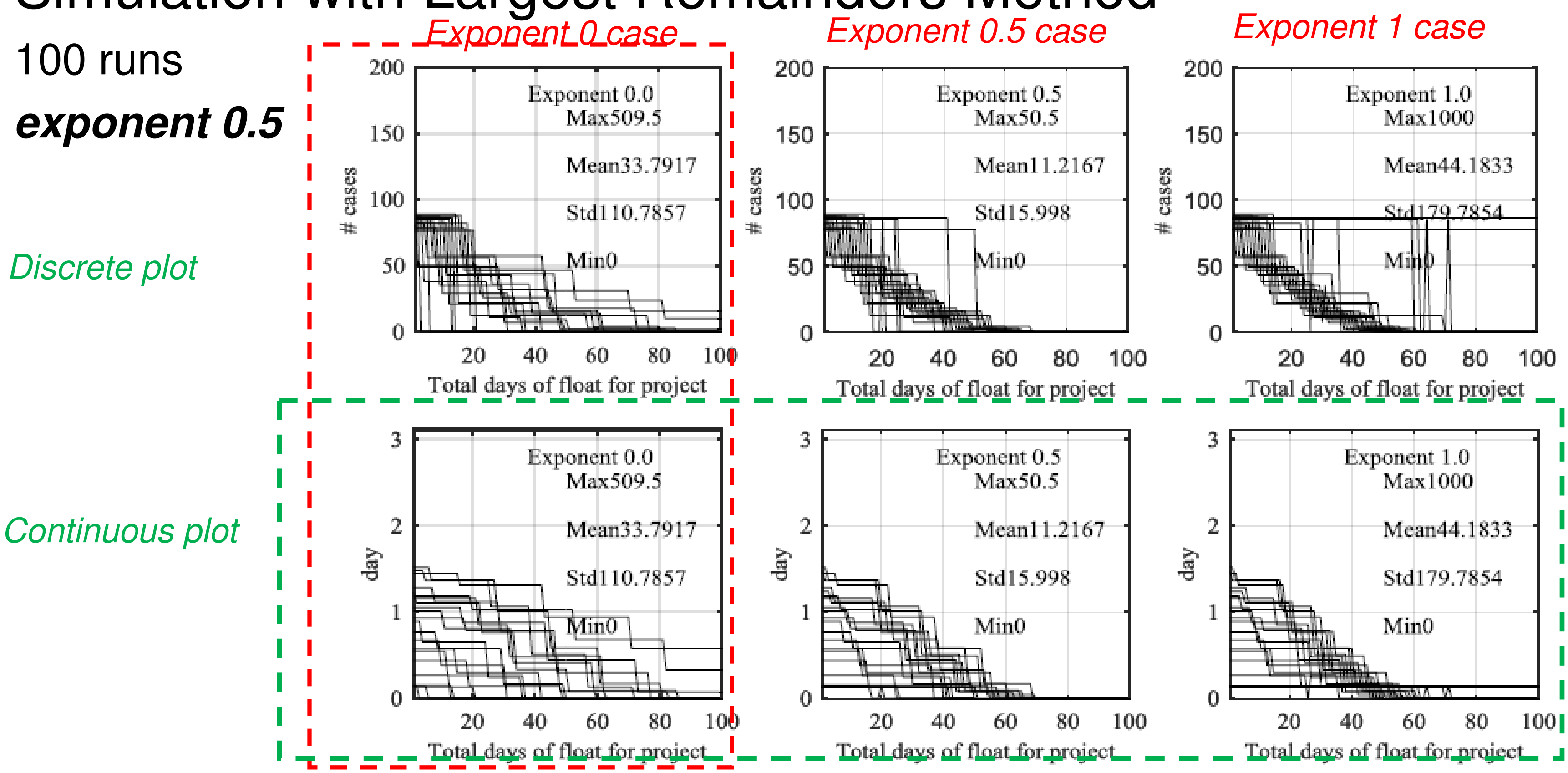


Figure 2: Discrete and Continuous Plots of Simulation Outputs

Case Study

- Simulation with Webster Method
  - 100 runs
  - ***exponent 0.5***

Discrete plot

Continuous plot

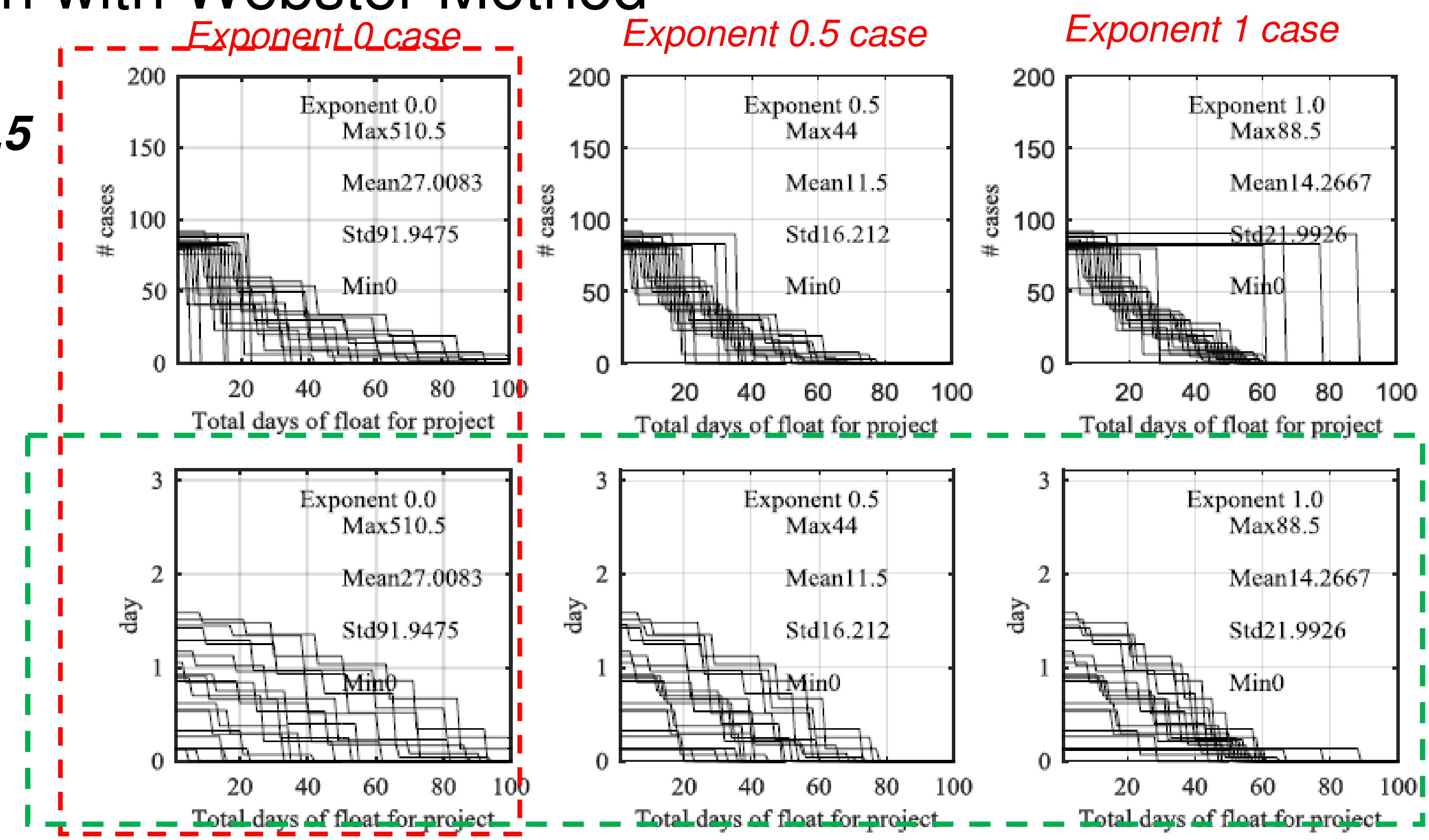


Figure 3: Discrete and Continuous Plots for the Simulation Outputs on Webster

Case Study

- Two advantages for 0.5 exponent
  - The mean of average float overrun is minimum (more economic way to apportion float)
  - Fairest to apportion float for all activities, long and short ones

- For exponent 0, all receive same pre-allocated float – short activities stop overrunning sooner

- For exponent 1, longer activities have advantage and stop overrunning sooner, short ones never receive sufficient float

Conclusions

- Inspired by apportionment methods, this paper integrate them with research on ***who owns float***
- Developed simulation model to test which exponent and with what apportionment method the float pre-allocation will be at an optimum
- Described how to measure the float overrun in discrete count and continuous measure, and represented graphically
- A schedule example from the PSPLIB has been tested and results validate float pre-allocation can be effective and efficient
- Future research will explore and simulate more complex scheduling phenomena, e.g. changes in the critical path due to the randomized durations of activities

Acknowledgement

The support of the National Science Foundation (Grant CMMI-1265989) for portions of the work presented here is gratefully acknowledged. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Thank you!

Do you have
any questions?

Contact

Gunnar Lucko, Ph.D.
Director, CE&M Program
Associate Professor
Department of Civil Engineering
Catholic University of America
620 Michigan Avenue NE
Washington, DC 20064

202-319-4381
lucko@cua.edu
http://faculty.cua.edu/lucko

Yi Su, M.S.
Graduate Research Assistant
Doctoral Student

61su@cardinalmail.cua.edu

THE CATHOLIC UNIVERSITY OF AMERICA