

Ley de Brook's para proyectos consecutivos

Integrantes:

J. D. Mendoza , L. M. Gómez

Entrega 2: Implementación Modelo Matemático Teórico Modelado, Simulación y Optimización

Departamento de Ingeniería de Sistemas y Computación
Universidad de Los Andes
Bogotá, Colombia

1 Contexto

Se busca gestionar dos o más proyectos consecutivos tal que se pueda minimizar el tiempo requerido para completarlos todos, respetando ciertos tiempos de retraso solo aplicables para el primer proyecto. Por motivos de administración ningún desarrollador auxiliar se puede adicionar a un proyecto antes de $2/3$ del tiempo de desarrollo, pero una vez adicionado al equipo este puede seguirá en los proyectos siguientes y por lo tanto reducir el tiempo requerido para completarlos (ya que el tiempo inicial calculado es con respecto al personal inicial del primero proyecto).

La ley de Brooks dice, parafraseando, que en las etapas finales de cualquier proyecto de desarrollo de software el agregar más personas al equipo de trabajo no reduce el tiempo requerido sino lo aumenta. Basado en lo anterior, cada desarrollador genera una penalización de tiempo para el proyecto en que se adicione dependiendo del mes en que se haga (se esperaría que entre más temprano en el proyecto la penalización sea menor ya que hay menos a que adaptarse). Una vez un desarrollador ya haga parte del equipo de trabajo se esperaría entonces que reduzca el tiempo de desarrollo del próximo proyecto, sin embargo, esto sería un tiempo constante ya que esta reducción depende de que el desarrollador haga parte del equipo antes de empezar el proyecto, no de cuánto antes de empezar el proyecto se unió.

Resumiendo, si se aplica la Ley de Brooks para un solo proyecto se esperaría un retraso proporcional a qué tan cerca está el proyecto de finalizarse, sin embargo, si se tienen en cuenta proyectos consecutivos esto podría llevar a un modelo de gestión más complejo donde lo que se busca minimizar el tiempo total que tomarían los proyectos a costa de un retraso en el primer proyecto (cómo mínimo).

Solo se resolverá el problema de cuantos integrantes se agregarán dependiendo del periodo en que entren al grupo de desarrollo. Se tienen en cuenta ciertas

parámetros que afectan el resultado o viabilidad del modelo, como lo es el tiempo de adaptación relativo, la disminución de tiempo relativa y el adelanto mínimo que debe tener uno o varios proyectos distintos del primero. Esto nos resulta en un problema MIP, ya que no hay ninguna variable que se multiplique por otra y además no hay variables que sean binarias. Se pensó originalmente como un problema MINLP en el que debía resolverse el número de nuevos integrantes y el periodo en que ingresaban pero debido a limitaciones de los solver esto no fue posible; sin embargo, valdría la pena resolverlo en otro estudio futuro ya que representa mejor el objetivo del problema.

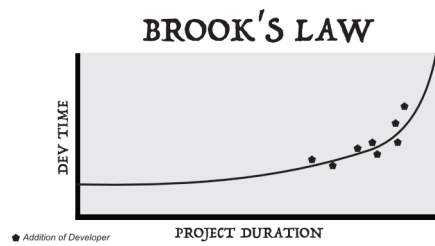


Fig. 1. Ejemplo de la ley de Brooks.

2 Sets, Constantes, Parametros, Variables

Table 1. Constantes

Constantes	Description
a	Tiempo de adaptacion de un nuevo integrante al proyecto.
c	Constante de adelanto del proyecto.
m	Adelantamiento mínimo del proyecto.
N1	Personal inicial del equipo de software.
Per	Personal maximo agregado.

Table 2. Sets

Sets	Description
n	Proyectos del problema.

Table 3. Parámetros

Parametros	Description
T_n	Tiempo inicial requerido para completar el proyecto n .
R_n	Retraso máximo del proyecto n .
B_n	Periodo en que se agrega personal al proyecto n

Table 4. Variables

Variables	Description
P_n	Tiempo final requerido para completar el proyecto n tras cambiar el personal
N_i^n	Personal inicial para el proyecto n .
N_f^n	Personal final para el proyecto n .
z	Variable de la función objetivo.

3 Función Objetivo, Ecuaciones y Restricciones

3.1 Función objetivo

$$\min(\sum_{i \in n} P_n) \quad (1)$$

3.2 Ecuaciones

* Para cualquier proyecto diferente del primero.

$$P_n = T_n + (N_f^n - N_i^n) * (a + B_n) - (N_i^n - N_i^1) * c \quad (2)$$

* Para el primer proyecto

$$P_n = T_n + (N_f^n - N_1) * (a + B_n) \quad (3)$$

La F.O indica que debemos minimizar el tiempo requerido para cada proyecto consecutivo después del siguiente de tal manera que excepto por el primero todos los demás proyectos se logren realizar antes de la fecha límite.

3.3 Restricciones

$$N_f^i \geq N_i^i \quad \forall i \in n \quad (4)$$

$$P_i - T_i \leq R_i \quad \forall i \in n \quad (5)$$

$$T_i - P_i \geq m \quad \forall i \in n \mid i \neq 1 \quad (6)$$

$$B_i < T_i * (\frac{1}{3}) \quad \forall i \in n \quad (7)$$

$$N_i^i = N1 \quad \forall i \in n \mid i = 1 \quad (8)$$

$$N_i^i = N_f^1 \quad \forall i \in n \mid i \neq 1 \quad (9)$$

$$N_f^i - N_i^i \leq B_i * 1000 \quad \forall i \in n \quad (10)$$

La restricción 5 representa que para cualquier proyecto el retraso total no puede superar Ri

La restricción 6 indica que el proyecto i se debe reducir al menos m tiempo excepto por $i = 1$.

La restricción 7 indica que los desarrolladores auxiliares solamente se podrían contratar en cualquiera de los proyectos antes del siguiente proyecto. El punto crítico de cualquier proyecto es después del segundo $\frac{1}{3}$, por lo tanto la contratación se mide con respecto a este periodo.

La restricción 10 indica que los desarrolladores auxiliares solamente se podrían contratar en cualquiera de los proyectos después del primer $\frac{1}{3}$ de su duración, el cual tomaremos como el 0.

La restricción 4 indica que el personal al final de cualquier proyecto debe ser mayor o igual al inicial.

4 Implementación y Resultados

4.1 Resultados Escenario 1

Table 5. Constantes Escenario 1

Constantes	Valor
a	2
c	4
N1	10
Per	3

Table 6. Sets Escenario 1

Sets	Description
1	Proyecto 1.
2	Proyecto 2.
3	Proyecto 3.

Table 7. Parámetros Escenario 1

Parametros	Valor
T_1	10
T_2	12
T_3	14
R_1	4
R_2	2
R_3	2
B_1	1
B_2	0
B_3	0

Descripcion de los Resultados:

Resultado de la función objetivo: 31

Resultados de Pn: P1= 13, P2= 8, P3 = 10

Resultados de Tiempo Inicial Requerido por Proyecto: P1= 10, P2= 12, P3 = 14

Con las constantes y supuestos estipulados anteriormente llegamos a una solución óptima de 31 periodos para un total de 3 proyectos en el cual solo se agrega una persona adicional en el primer periodo. Podemos ver los tiempos requeridos que se tenian inicialmente vs. el tiempo al que se lograron optimizar.

4.2 Resultados Escenario 2

Table 8. Constantes Escenario 2

Constantes	Valor
a	2
c	2
m	2
N1	6
Per	3
m	3

Table 9. Sets Escenario 2

Sets	Description
1	Proyecto 1.
2	Proyecto 2.
3	Proyecto 3.

Table 10. Parámetros Escenario 2

Parametros	Valor
T_1	8
T_2	14
T_3	14
R_1	6
R_2	2
R_3	2
B_1	1
B_2	0
B_3	0

Descripcion de los Resultados:**Resultado de la función objetivo:** 34**Resultados de Pn:** P1= 14, P2= 10, P3 =10

Para este escenario se tenía un valor de $m=3$, lo cual significaba que cualquier proyecto distinto del primero se debe garantizar que el tiempo final requerido sea 3 meses menos que el inicial que se había calculado con el persona que había. Además, bajo el valor de los parámetros que se tienen es posible agregar más de una persona ya que el retraso total generado fue de 6 meses, pues esto está dentro del rango permitido por el parámetro R1 que define cuanto se puede atrasar un proyecto determinado. En general, mediante este escenario y el anterior se puede observar el efecto que tiene el aumentar el equipo de desarrollo más allá de un primer proyecto que sufre bajo las suposiciones de la Ley de Brook. Es claro que de tener más proyectos consecutivos habría una disminución ,con respecto al tiempo inicial calculado, lineal.

Cabe aclarar que el modelo simplifica el comportamiento humano e ignora que estos tiempos no son realmente constantes, pueden variar a medida que se desarrolla cada proyecto y con respecto a distintas dificultades que son inherentes al desarrollo de software.

References

1. Laurie Williams, Anuja Shukla, Annie I. Antón. *Pair Programming and the Factors Affecting Brooks' Law*. Department of Computer Science, North Carolina State University. <https://repository.lib.ncsu.edu/bitstream/handle/1840.4/776/TR-2003-04.ps?sequence=2&isAllowed=y>