

Generación de la solución inicial para un problema de planificación de actividades de un modelo educativo

Juan Pablo Rosas Baldazo

juan.rosasbl@uanl.edu.mx

Posgrado en Ingeniería de Sistemas, FIME - UANL

Av. Pedro de Alba S/N, San Nicolás de los Garza, Nuevo León, México.

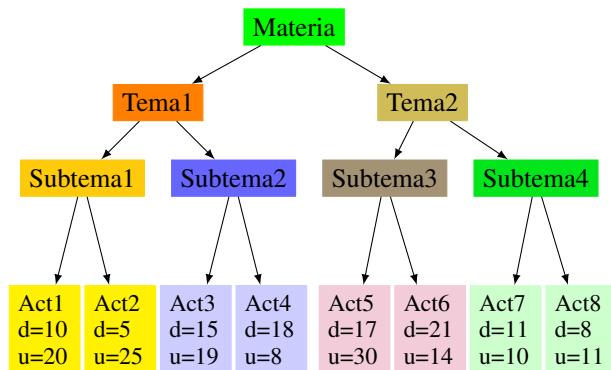
Resumen

Mediante la generación de soluciones aleatorias se pretende aproximar a una solución óptima o a una solución de muy buena calidad que sirva como punto de partida para el planificador OPTIC con lo que se disminuiría el tiempo de generación de una secuencia de actividades y se mejoraría la calidad de la solución.

Introducción

En este documento se presenta un método de generación de soluciones al problema de planificación de actividades de un modelo educativo las cuales constan de una duración y un puntaje que afecta la calificación final de la materia, cada actividad pertenece a un subtema en particular y a su vez el subtema a un tema y el tema a una materia, el problema también está sujeto a cumplir con una calificación mínima y una calificación máxima por subtema, esto puede variar dependiendo del problema, también existen restricciones de precedencia, es decir, una actividad no se puede realizar hasta que se halla realizado una o dos actividades específicas antes que ella o estar sujeta no a las actividades si no al subtema, esto quiere decir que la actividad no se puede realizar hasta que un subtema en particular halla obtenido la calificación mínima aprobatoria.

La estructura del problema sería similar al Grafo siguiente:



Antecedentes

Con el fin de observar el desempeño de algunos planificadores se implementó el problema en dos en particular LPG y SGPLAN, con los cuales ninguno alcanzó el valor óptimo, también se resolvió mediante el modelo matemático del problema con el que se obtuvo el valor óptimo pero no genera una secuencia.

Solución Propuesta

Considerando que el planificador consta de una fase de mejora de la solución inicial encontrada la cual aumenta por mucho el tiempo de cómputo ya que explora el espacio de soluciones utilizando un procedimiento similar al de ramificación y poda hasta encontrar una solución mejor a la que ya se tiene, se propone generar soluciones de calidad las cuales serán el punto de partida del planificador, con esto se pretende disminuir el espacio de búsqueda del planificador y también alcanzar el valor óptimo del problema en un menor tiempo.

Para obtener una solución de calidad se utiliza un procedimiento similar al algoritmo Monte Carlo donde se genera un número n de soluciones de las cuales se selecciona la de mejor calidad. Esto nos permite aproximarnos al valor óptimo, a mayor cantidad de soluciones generadas mayor probabilidad de generar la solución óptima, esto no quiere decir que mejore el promedio general de la materia, solo se optimiza el tiempo, es decir una solución dada puede tener un menor promedio que una solución anterior pero también una menor duración lo cual es el objetivo del problema.

Para generar las soluciones se simulara la selección de actividades respetando las restricciones de precedencia, calificación mínima y máxima del problema y también se consideran las actividades obligatorias en caso de que existan.

El algoritmo con el que se realiza la selección de las actividades es el siguiente:

Algoritmo 1 Monte Carlo.

```
1:  $plan \leftarrow 0$ 
2:  $puntos \leftarrow 0$ 
3: procedure MONTECARLO( $i$ )
4:    $act.ob \leftarrow$  Actividades Obligatorias
5:   if Longitud  $act.ob > 0$  then
6:     for ( $a$  in  $act.ob$ ) do
7:        $plan \leftarrow a$ 
8:        $puntos \leftarrow a.puntos$ 
9:     end for
10:  end if
11:  while !(Todos en  $puntos > 70$ ) do
12:     $i \leftarrow num.aleatorio$ 
13:     $act \leftarrow actividades[i]$ 
14:    if ( $act.puntos + puntos < 100$ ) then
15:      if ( $act$  Tiene un requisito) then
16:        if ( $act$  Tiene dos requisitos) then
17:          if !( $act.req1$  ya se agrego) then
18:            if !( $act.req2$  ya se agrego) then
19:               $plan \leftarrow act$ 
20:               $plan \leftarrow act.req1$ 
21:               $plan \leftarrow act.req2$ 
22:               $puntos \leftarrow act$ 
23:               $puntos \leftarrow act.req1.puntos$ 
24:               $puntos \leftarrow act.req2.puntos$ 
25:            else
26:               $plan \leftarrow act$ 
27:               $plan \leftarrow act.req1$ 
28:               $puntos \leftarrow act.puntos$ 
29:               $puntos \leftarrow act.req1.puntos$ 
30:            end if
31:          else
32:            if ! then( $act.req2$  ya se agrego)
33:               $plan \leftarrow act$ 
34:               $plan \leftarrow act.req2$ 
35:               $puntos \leftarrow act.puntos$ 
36:               $puntos \leftarrow act.req2.puntos$ 
37:            else
38:               $plan \leftarrow act$ 
39:               $puntos \leftarrow act.puntos$ 
40:            end if
41:          end if
42:        else
43:          if ! then( $act.req1$  ya se agrego)
44:             $plan \leftarrow act$ 
45:             $plan \leftarrow act.req1$ 
46:             $puntos \leftarrow act.puntos$ 
47:             $puntos \leftarrow act.req1.puntos$ 
48:          else
49:             $plan \leftarrow act$ 
50:             $puntos \leftarrow act.puntos$ 
51:          end if
52:        end if
53:      else
54:         $plan \leftarrow act$ 
55:         $puntos \leftarrow act.puntos$ 
56:      end if
57:    end if
58:  end while
59: end procedure
```

Experimentación

Instancias

Para realizar la simulación se utilizaron varias tipos de instancias para las cuales los rangos de valores de las actividades son de dos clases, uniformes y variables y de tres tamaños diferentes pequeñas, medianas y grandes, para los rangos uniformes se tomaron valores entre 10-70 y para los valores aleatorios corresponden con $\frac{100}{n} \pm 5$ donde n es el numero de actividades del problema.

Instancias			
%Dependencia	%Tipo A	%Tipo B	%Obligatorias
0	0	0	0
0	0	0	10
0	0	0	20
20	20	80	0
20	80	20	0
80	20	80	0
80	80	20	0
20	20	80	10
20	80	20	10
80	20	80	10
80	80	20	10
20	20	80	20
20	80	20	20
80	20	80	20
80	80	20	20

Table 1: Tipos de instancias, los valores en las celdas corresponden a porcentaje de actividades, El tipo A hace referencia a que solo requiere una actividad como requisito, tipo B ,requiere 2 actividades, la columna de obligatorias hace referencia a actividades que se tienen que incluir en la secuencia.

Los tamaños de los problemas varían en cuanto a la cantidad de temas, subtemas y actividades que los conforman.

Tamaño de Problemas			
Instancia	Temas	Subtemas	Actividades
Pequeña	1	2	5
Mediana	3	4	5
Grande	5	6	5

Table 2: Tamaño de instancias, la cantidad de subtemas es por tema y la cantidad de actividades es por subtema.

Se generaron cinco problemas de cada tipo dando como resultado 75 problemas con rangos de valores uniformes y 75 con valores aleatorios, dando un total de 150 problemas de cada tamaño pequeño, mediano y grande, es decir 450 problemas.

Simulación

La simulación se realizó en una laptop Lenovo Thinkpad con un procesador Intel Core i7-4710MQ y 31 gb RAM, con un sistema operativo Ubuntu 15.10.

Para la simulación se opto por variar la cantidad de soluciones generadas, esto bajo el supuesto de que a mayor cantidad de soluciones mayor probabilidad de generar una solución de mejor calidad o alcanzar el valor óptimo y no se requiere la misma cantidad de soluciones para alcanzar el óptimo en instancias de tamaño pequeño comparadas con instancias grandes.

Para las instancias de tamaño mediano la simulación se realizo con 1000, 2000, 3000, 4000, 5000 y 6000 soluciones, para las instancias de tamaño mediano la simulación se realizo con 10000, 20000, 30000, 40000, 50000 y 60000 soluciones y para las de tamaño grande solo se probó con un solo tamaño 100000 soluciones.

Para comparar los resultados de la simulación con los de OPTIC y no dejar que el planificador extienda su tiempo de cómputo tratando de mejorar la solución, se limito el planificador a dos minutos para obtener la solución y en dado caso de que no encuentre una se le asigna el valor de 0. Los valores óptimos de los problemas se obtuvieron mediante GAMS.

Resultados

En los datos expuestos OPTIC no pudo mejorar la solución inicial en menos de dos minutos por lo que los resultados se comparan con la primer solución encontrada por OPTIC.

Como se puede observar en la Figura1 y la Figura2 los resultados de OPTIC y la simulación estan muy lejos de alcanzar el valor óptimo para valores aleatorios y cabe resaltar que en algunos problemas OPTIC no alcanzo a obtener la solución del problema, sin embargo OPTIC presenta un muy buen desempeño en valores uniformes, casi obteniendo el valor óptimo.

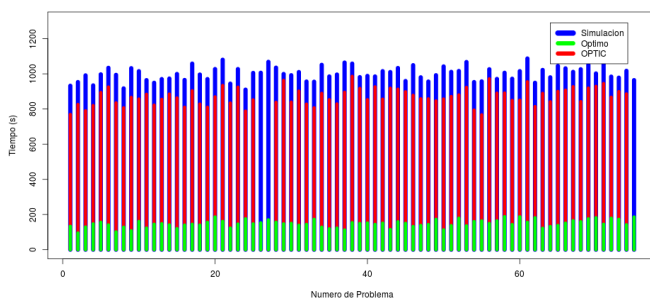


Figure 1: Comparación de resultados de la simulación con 60000 soluciones contra OPTIC y el valor óptimo.

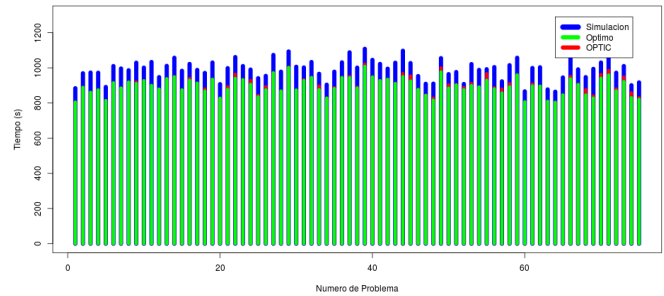


Figure 2: Comparación de resultados de la simulación con 60000 soluciones contra OPTIC y el valor óptimo.

Sin embargo, el incremento en la cantidad de soluciones generadas para la simulación, incrementa como se esperaba la posibilidad de encontrar mejores soluciones. Esto sin afectar por mucho el tiempo de cómputo como se observa en la Figura7

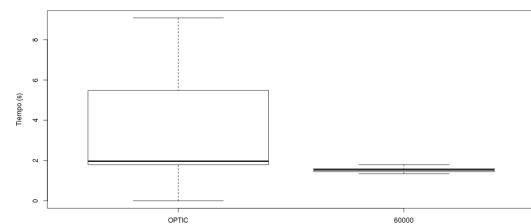


Figure 3: Tiempos de cómputo de OPTIC y la simulación.

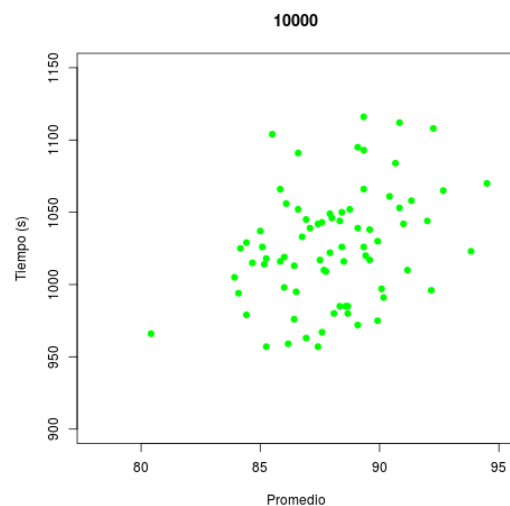


Figure 4: Resultados de la simulación con 10000 soluciones.

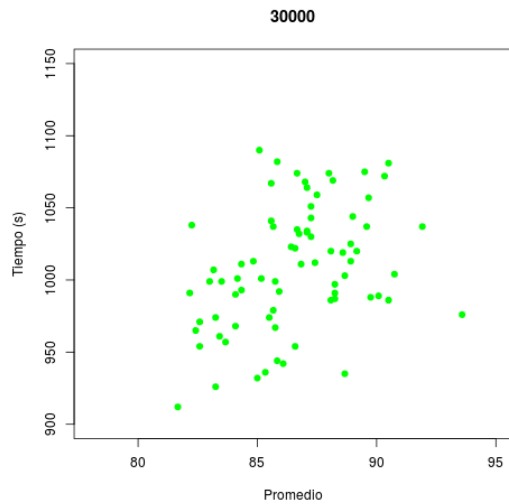


Figure 5: Resultados de la simulación con 30000 soluciones.

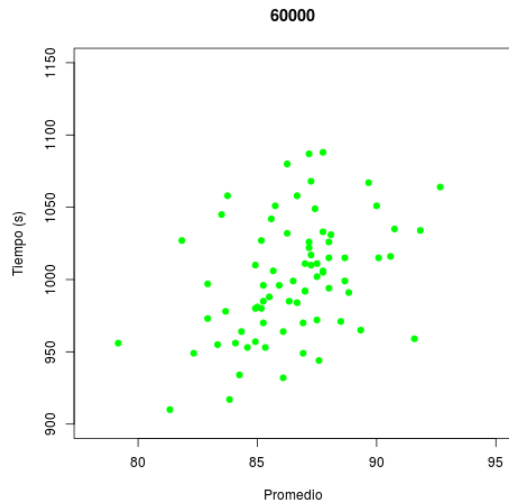


Figure 6: Resultados de la simulación con 60000 soluciones.

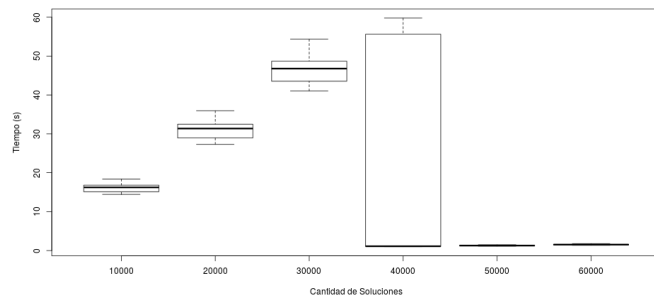


Figure 7: Tiempos de cómputo de la simulación.

Conclusiones

A pesar de que la simulación no causo mejores tiempos ni soluciones que las del planificador OPTIC, si muestra una mayor rapidez para mejorar la solución que la del planificador

Trabajo Futuro

- Mejorar la simulación para tratar de disminuir el tiempo que tarda en crear soluciones y cambiarla a generar n cantidad de soluciones en un tiempo determinado, es decir no definir un numero en particular si no un tiempo limite de generación de soluciones.
- Obtener el tiempo que tarda OPTIC en mejorar una solución y compararlo con los de la simulación.
- Mejorar la generación de soluciones de buena calidad utilizando una metaheurística que no cause un tiempo computacional mayor al de OPTIC.
- Integrar la generación de soluciones con el planificador para corroborar si hay una mejora en la obtención de resultados.

Referencias

J. Bentonn, Amanda Coles and Andrew Coles, 2012. Temporal Planning with Preferences and Time-Dependent Continuous Costs. *AAAI*.

R. Sanchez Nigenda, C. Maya Padron, I. Martinez-Salzar, F. Torres-Guerrero, Design and evaluation of planning and mathematical models for generating learning paths

Malik Ghallab, Dana Nau, Paolo Traverso ,Automated Planning and Acting, *Capitulo 6: 314-347*