

ANÁLISIS EXPERIMENTAL DEL COSTO COMPUTACIONAL DEL PROBLEMA 2+p-SAT ALEATORIO

Liliam R. Suaza¹ y Carlos A. Arbeláez², Dpto. de Sistemas, Facultad de Ingeniería, Universidad de Antioquia, Colombia

Roberto Cruz³, Dpto. Matemáticas, Facultad Ciencias Exactas y Naturales, Universidad de Antioquia, Colombia

RESUMEN

El problema 2+p-SAT aleatorio es una interpolación entre el problema de tiempo polinomial 2-SAT aleatorio, donde $p = 0$, y el problema NP-Completo 3-SAT aleatorio, donde $p = 1$. Este estudio describe el costo computacional de una modificación del algoritmo TABLEAU sobre instancias 2+p-SAT aleatorias. La descripción fue obtenida por medio de experimentos con instancias 2+p-SAT aleatorias para valores de p en el intervalo $0 \leq p \leq 1$, con un paso de 0.05, y valores de la variable n en el intervalo $50 \leq n \leq 8000$. El costo computacional típico observado en los experimentos tuvo un comportamiento polinomial en la región definida para valores de p hasta 0.60, y un comportamiento exponencial en la región definida por $p \geq 0.65$. La región obtenida de comportamiento polinomial es significativamente más amplia que la obtenida por otros autores. Se observaron instancias extremadamente difíciles en la región $0.60 \leq p \leq 0.95$. Para la región donde $p \leq 0.50$, todas las instancias no satisfacibles fueron resueltas, explorando cero ramas del árbol de búsqueda. Este último resultado sugiere que, para instancias no satisfacibles en la región $p \leq 0.50$, el algoritmo modificado TABLEAU se comporta como en instancias 2-SAT.

ABSTRACT

Random 2+p-SAT problem interpolates between the polynomial-time random 2-SAT problem, where $p = 0$, and the NP-Complete random 3-SAT problem, where $p = 1$. This study describes the computational cost of a modification of the TABLEAU algorithm over random 2+p-SAT instances. The description was achieved through experiments on random 2+p-SAT instances for values of p on the interval $0 \leq p \leq 1$, with 0.05 steps and n on the interval $50 \leq n \leq 8000$. The typical computational cost observed in the experiments has a polynomial behavior on the region defined by values of p up to 0.60, and an exponential behavior on the region defined by $p \geq 0.65$. The obtained region of polynomial behavior is significantly broader than the region obtained by other authors. Extremely hard instances were observed in the region $0.60 \leq p \leq 0.95$. For the region $p \leq 0.50$, all the unsatisfiable instances were solved with zero explored branches on the search tree. The latter result suggests that, for unsatisfiable instances on the region $p \leq 0.50$, the modified TABLEAU algorithm behaves like 2-SAT instances.

Key words: NP-hard, satisfiability 2+p-SAT, transition phase.

MSC: 90B06.

1. INTRODUCCIÓN

La teoría de los problemas NP-completos se basa en el análisis del caso más difícil [1]. Establecer que un problema es NP-completo implica que algunas instancias del problema requieren de un esfuerzo computacional exponencial, independientemente del algoritmo que se utilice. Para ciertos problemas NP-completos una distribución probabilística de instancias puede ser tratable en el caso promedio o típico, debido a que las instancias computacionalmente intratables aparecen con una probabilidad muy baja. Existen también problemas NP-completos para los cuales todas las distribuciones probabilísticas de instancias del problema parecen ser intratables, pues, tanto en el caso típico como en el caso más difícil, el costo computacional se escala exponencialmente. Por último, existen subproblemas especiales de problemas NP-completos, que pueden ser resueltos con algoritmos polinomiales.

El paradigma de la clase de problemas NP-completos es el problema de la satisfacibilidad [2], del cual se deriva el k-SAT aleatorio.

E-mail: ¹isuaza@jaibana.udea.edu.co

²scaav328@udea.edu.co

³rcruz@matematicas.udea.edu.co

Definición 1: dado un conjunto de n variables booleanas, denotamos por B_k el conjunto de todas las k -cláusulas, es decir, las disyunciones de k literales diferentes y no complementarios de las n variables. La fórmula k -SAT aleatoria $F_k(n,m)$ se genera seleccionando aleatoriamente m cláusulas del conjunto B_k , separándolas por el operador lógico ' \wedge '. El problema k -SAT consiste en hallar una asignación de valores a las variables tal que todas las cláusulas de $F_k(n, m)$ sean satisfechas.

Monasson y otros [3][4] introdujeron el problema 2+p-SAT aleatorio.

Definición 2: una fórmula 2+p-SAT aleatoria $F_{2+p}(n, m)$, con $p \in [0, 1]$, está formada por m cláusulas generadas aleatoriamente, donde pm cláusulas son seleccionadas desde B_3 y $(1 - p)m$ desde B_2 . El problema 2+p-SAT aleatorio consiste en hallar una asignación de valores a las variables tal que todas las cláusulas de $F_{2+p}(n, m)$ sean satisfechas.

El problema 2+p-SAT aleatorio es una interpolación entre el problema 2-SAT aleatorio ($p = 0$), de tiempo polinomial, y el problema NP-completo 3-SAT aleatorio ($p = 1$). Dichos autores sugirieron que, cuando n tiende a infinito, para cada $p \in [0,1]$ existe un valor crítico de $\alpha = m/n$, denotado por $\alpha_c(p)$, alrededor del cual $F_{2+p}(n, \alpha n)$ sufre una transición de fase, pasando de ser satisfacible a ser no satisfacible. El rango de variación de p se divide en dos regiones cualitativamente diferentes separadas por un valor p_c , que ellos estimaron aproximadamente en 0.41. Para $p < p_c$ el problema 2+p-SAT aleatorio es estructuralmente similar al problema 2-SAT y $\alpha_c(p) = 1/(1 - p)$. Para $p > p_c$ el problema parece tener las propiedades estructurales del problema 3-SAT. Achlioptas y otros [5] obtuvieron los primeros resultados teóricos acerca del comportamiento del problema 2+p-SAT, estableciendo una cota inferior de $2/5$ y una cota superior de 0.695 para p_c .

Monasson y otros hicieron también un estudio del costo computacional del algoritmo completo TABLEAU [6] en conjuntos de instancias $F_{2+p}(n, m)$, para valores de $p = 0.0, 0.2, 0.4, 0.6, 0.8$ y 1.0 ; concluyendo que este costo se escala linealmente para $p \leq 0.4$ y exponencialmente para $p \geq 0.6$. Ellos utilizaron, como medida del costo computacional, el número de ramificaciones (N_r) necesarias para encontrar una solución o para establecer que la instancia no es satisfacible. Representaron el costo típico de cada conjunto de instancias por la mediana de los N_r , y no por su promedio, pues éste se afecta significativamente por la aparición de instancias "extremadamente difíciles" (instancias con valores de N_r muy altos) que ocurren con una probabilidad muy baja. Estudiaron el costo computacional en $\alpha_{0.5}(p, n)$, que es el valor de α donde se obtiene el 50% de instancias satisfacibles y es mayor la incertidumbre acerca de la satisfacibilidad o no satisfacibilidad de la fórmula, lo cual genera la mayor dificultad para los algoritmos de búsqueda.

Un comportamiento similar al observado por Monasson y otros, fue obtenido por Singer y otros [7] utilizando el algoritmo incompleto de búsqueda local NOVELTY [8]. Estos autores generaron instancias aleatorias para $p = 0.0, 0.3, 0.5, 1.0$ y determinaron el valor aproximado de $\alpha_{0.5}(p, n)$. El costo computacional de este algoritmo se escaló de forma polinomial para $p = 0.0$ y $p = 0.3$ y de forma superpolinomial para $p = 0.5$ y $p = 1$.

Con el fin de analizar detalladamente el cambio de comportamiento del costo computacional, hemos estudiado experimentalmente el problema 2+p-SAT aleatorio para valores de p entre 0 y 1, con incrementos de 0.05, siguiendo la metodología descrita en [3][4]. Utilizamos para ello una modificación eficiente del algoritmo TABLEAU, la cual nos permitió analizar instancias constituidas por a lo sumo 8000 variables. En la sección 2 se describen la modificación del algoritmo TABLEAU y su validación. Los resultados obtenidos y el análisis del costo computacional se describen en la sección 3. En la sección 4 se dan las conclusiones.

2. MODIFICACIÓN DEL ALGORITMO TABLEAU

El algoritmo TABLEAU [6] es un algoritmo completo recursivo basado en el algoritmo original de Davis, Logemann y Loveland [9], al cual se le añade un procedimiento altamente optimizado de "propagación"⁴ y un conjunto de heurísticas para realizar la selección de la variable de ramificación.

El criterio de selección de la variable de ramificación utilizado en el TABLEAU prioriza aquellas variables que causan el mayor número de propagaciones. El proceso de selección se compone de tres etapas: la primera consiste en calcular aproximadamente para cada variable no asignada, el número de propagaciones

⁴El procedimiento de "propagación" consiste en encontrar una cláusula con un solo literal x , retirar de la fórmula todas las cláusulas que contengan x y retirar de todas las cláusulas el complemento de x . Este proceso se repite hasta que no queden cláusulas de un literal.

que resultarían de asignarle el valor “Verdadero” y el valor “Falso”. En la segunda etapa se escoge un subconjunto de las variables con el mayor número calculado, y finalmente, en la tercera etapa, para cada una de las variables escogidas, se calcula exactamente el número de propagaciones causado por la asignación del valor “Verdadero” y del valor “Falso”. La variable escogida es aquella que produce el mayor número de propagaciones en ambas asignaciones. El objetivo que se persigue, al restringir la selección de la variable de ramificación a un subconjunto, es reducir el costo que representa hacer el procedimiento de asignación sobre todas las variables no asignadas.

En nuestra variante del algoritmo se eliminaron las dos primeras etapas, permitiendo así que todas las variables no asignadas participen del proceso de selección exacto. Aunque esto, en principio, es más costoso en cada nivel del árbol de búsqueda, el número total de ramificaciones necesarias para resolver una instancia 3-SAT, sin embargo, se reduce significativamente. Además, si las dos asignaciones a una variable no asignada producen una contradicción, inmediatamente se retorna al nivel anterior, evitando así escoger una rama que produzca a largo plazo una contradicción.

La validación de esta modificación se realizó sobre un conjunto de instancias del problema 3-SAT aleatorio, generadas con los mismos parámetros utilizados por Crawford y Autom [6]. Se utilizaron 1000 instancias para cada combinación de n y m. El número de variables fue entre 25 y 300 con incrementos de 25. Para cada valor de n se tomaron los valores de m con los cuales Crawford y Autom obtuvieron el 50 % de satisfacibilidad. Para describir el costo computacional del TABLEAU sobre los conjuntos de instancias descritos, Crawford y Autom ajustaron el promedio de Nr para cada valor n a la función

exponencial $2^{19.5 \frac{n}{25} + 0.08}$. Los resultados obtenidos con nuestra variante del algoritmo se ajustaron a la función $2^{21.3 \frac{n}{25} - 0.24}$, lo cual muestra una reducción del costo cuando n crece. Por ejemplo, para n=300 Crawford y Autom obtuvieron un promedio de Nr igual a 44646, mientras que con nuestra variante del algoritmo se obtuvo un promedio de Nr igual a 15176.

La modificación del algoritmo TABLEAU se hizo sobre el código fuente, ANSII C, que provee Crawford. El código fuente, original y modificado, puede descargarse en <http://cs-95.cs.udea.edu.co/sat/algorithm.html>. En esta página también se muestran cuales fueron las principales modificaciones realizadas al código fuente del algoritmo TABLEAU original.

3. RESULTADOS

La variante del algoritmo TABLEAU permitió analizar instancias con un número de variables considerablemente mayor que los reportados en otros estudios [3] [7]. Se utilizaron conjuntos de entre 1000 y 5000 instancias con valores de p entre 0.00 y 1.00 e incrementos de 0.05, y valores de n entre 50 y 8000. En la Tabla 1, se muestra el mayor valor de n estudiado para cada valor de p; este valor de n estuvo limitado por el incremento del costo de los cálculos a medida que p crece. Con el fin de obtener aproximadamente el valor de $\alpha_{0.5}(p, n)$, se determinó el mayor valor de m para el cual al menos el 50% de las instancias tienen solución.

Tabla 1. Máximo valor de n estudiado en cada valor de p.

P	Máximo n	p	Máximo n	p	Máximo n
0.00 a 0.60	8000	0.75	2000	0.90	600
0.65	6000	0.80	1250	0.95	500
0.70	3500	0.85	900	1.00	300

En las Figuras 1a y 1b se muestran los gráficos del costo computacional para diferentes valores de p en dependencia del valor de n. Los resultados obtenidos sugieren un cambio cualitativo del costo computacional entre $p = 0.60$ y $p = 0.65$. Los valores de la mediana de Nr, para $p \leq 0.60$, se ajustaron a funciones lineales de n con coeficientes de R^2 superiores a 0.99. Para $p \geq 0.65$ se obtuvo un comportamiento exponencial cualitativamente diferente al anterior, pues nuestros datos se ajustaron a funciones exponenciales del tipo 2^{an+b} con valores de R^2 superiores a 0.98. La región de comportamiento polinomial obtenida, $0 \leq p \leq 0.60$, es significativamente más amplia que la reportada por otros autores. En el caso de Monasson y otros [3] el costo computacional se escaló polinomialmente para $p = 0.40$ y $n = 8000$ y exponencialmente para $p = 0.60$ y $n = 2000$. Singer y otros [7] obtuvieron un escalamiento polinomial del costo para $p = 0.30$ y superpolinomial para $p = 0.50$, utilizando instancias satisfacibles con hasta 1500 variables. No obstante las diferencias con trabajos anteriores, nuestros resultados experimentales están en concordancia con los resultados teóricos obtenidos por Achlioptas y otros [5] para el valor de p_c , donde $2/5 \leq p_c \leq 0.695$.

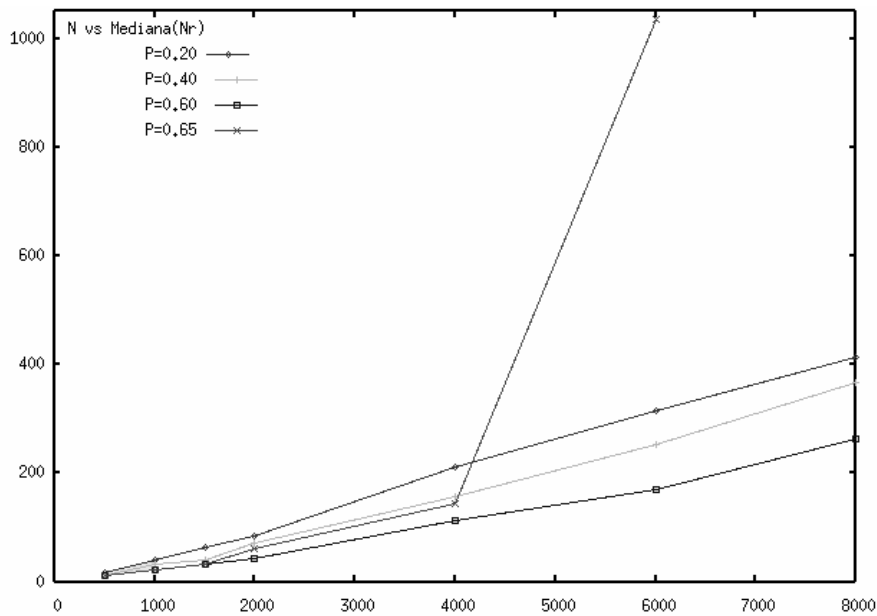


Figura 1a. Mediana del número de ramificaciones N_r en dependencia de n .

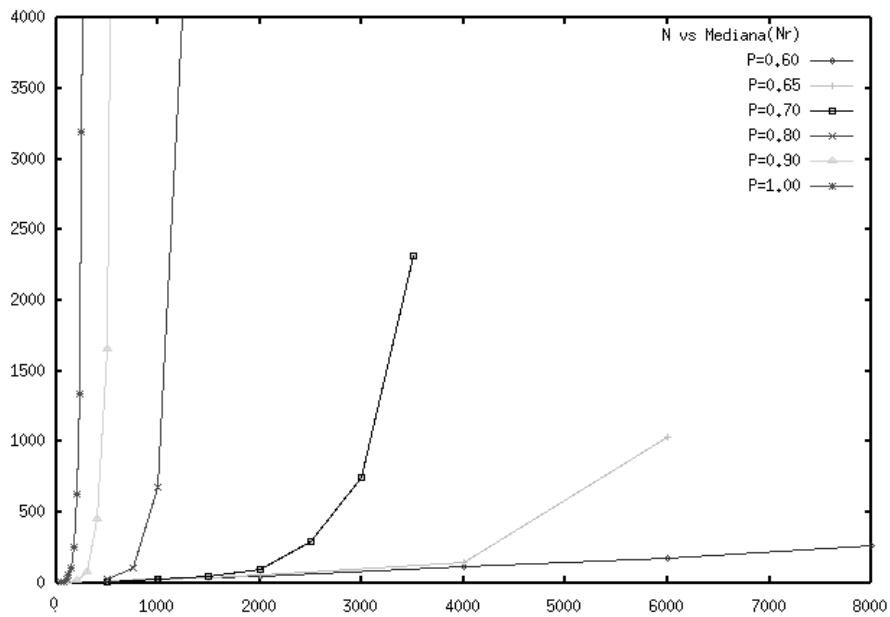


Figura 1b. Mediana del número de ramificaciones N_r en dependencia de n .

Como se puede observar en la Figura 1a, para $p \leq 0.60$ se registra un ligero aumento del número de ramificaciones a medida que el valor de p disminuye, es decir, a medida que las instancias se hacen más similares a instancias del problema 2-SAT. Esto se debe a que la estrategia de nuestra variante del algoritmo TABLEAU consiste en reducir las fórmulas lo más rápidamente posible a fórmulas 2-SAT. Sin embargo, el criterio de selección de la variable de ramificación deja de tener efecto una vez que la fórmula se reduce a 2-SAT. El incremento en el número de ramificaciones obtenido está dado por el aumento del tamaño de las fórmulas 2-SAT a las que se reducen las fórmulas 2+p-SAT aleatorias cuando el valor de p disminuye.

Para valores de $p \geq 0.60$ se observó un crecimiento de la mediana de N_r a medida que el valor de p aumenta; sin embargo, para valores de $0.65 \leq p \leq 0.95$ se observó la aparición de instancias extremadamente difíciles, con valores de N_r inusualmente grandes que aparecen con una probabilidad muy baja. Estas instancias se caracterizaron por presentar un valor de N_r superior a un valor de umbral definido:

$$\text{Valor de umbral} = Q3 + 3 * Rq$$

donde **Q3** es el tercer cuartil y **Rq** es el rango intercuartílico de los valores de N_r .

En la Tabla 2 se muestra tanto el mayor porcentaje de aparición de instancias extremadamente difíciles para cada valor de p , como el número de veces que la diferencia entre el máximo de Nr y la mediana de Nr supera al rango intercuartílico, con el fin de mostrar cuan alejado está la instancia más difícil de la mediana en cada conjunto. Para valores de $p \leq 0.55$ y $p = 1.00$ no hay instancias extremadamente difíciles, éstas empiezan a aparecer en $p = 0.60$, y su frecuencia de aparición se dispara para valores de $0.65 \leq p \leq 0.70$. Los mayores valores de Nr obtenidos en $p = 0.65$ y $p = 0.70$, fueron $Nr = 135160$ y $Nr = 381910$, respectivamente, los cuales son significativamente más grandes que los obtenidos con otros valores de p . Además, en estos casos se observaron los mayores números de veces que la diferencia entre el máximo de Nr y la mediana supera el rango intercuartílico. Para $p \leq 0.55$ los máximos valores de Nr estuvieron por debajo de 590, y para $p = 1.00$ el mayor valor de Nr obtenido fue $Nr = 50814$. Aquí falta explicar las columnas 3 y 6 de la Tabla y cambiar la definición.

Tabla 2. Porcentaje de aparición de instancias extremadamente difíciles y número de veces que la diferencia entre el máximo de Nr y la mediana de Nr , supera al rango intercuartílico en cada conjunto.

P	%	(Max(Nr)– mediana) /Rq	P	%	(Max(Nr)– mediana) /Rq
0.60	0.40%	7.06	0.80	6.10%	13.36
0.65	11.30%	41.45	0.85	3.44%	7.49
0.70	15.22%	59.00	0.90	2.70%	7.51
0.75	6.52%	15.92	0.95	2.20%	5.06

En nuestro experimento se observó que para valores de $p \leq 0.50$, todas las instancias no satisfacibles fueron resueltas con $Nr = 0$, es decir, al aplicar por primera vez el procedimiento de selección de la variable de ramificación se encontró una variable que producía una contradicción, al asignarle tanto el valor “Verdadero” como el valor “Falso”. Este comportamiento es propio del problema 2-SAT aleatorio, donde también es posible probar la no satisfacibilidad con $Nr = 0$. Esto sugiere que las instancias con $p \leq 0.50$, utilizadas en nuestro estudio, estuvieron dominadas por la parte 2-SAT. El valor de $p = 0.50$ es ligeramente mayor que la cota inferior de $p_c = 0.4$ obtenida por Achlioptas [5]. Para $p = 0.55$ comienzan a aparecer pequeños valores de Nr mayores que cero. Estos valores de Nr en las instancias no satisfacibles comienzan a crecer y a tener influencia en el costo computacional sólo a partir de $p = 0.6$. Para $p \geq 0.75$ la mediana de Nr en las instancias no satisfacibles supera la de las satisfacibles, como se ilustra en la Figura 2.

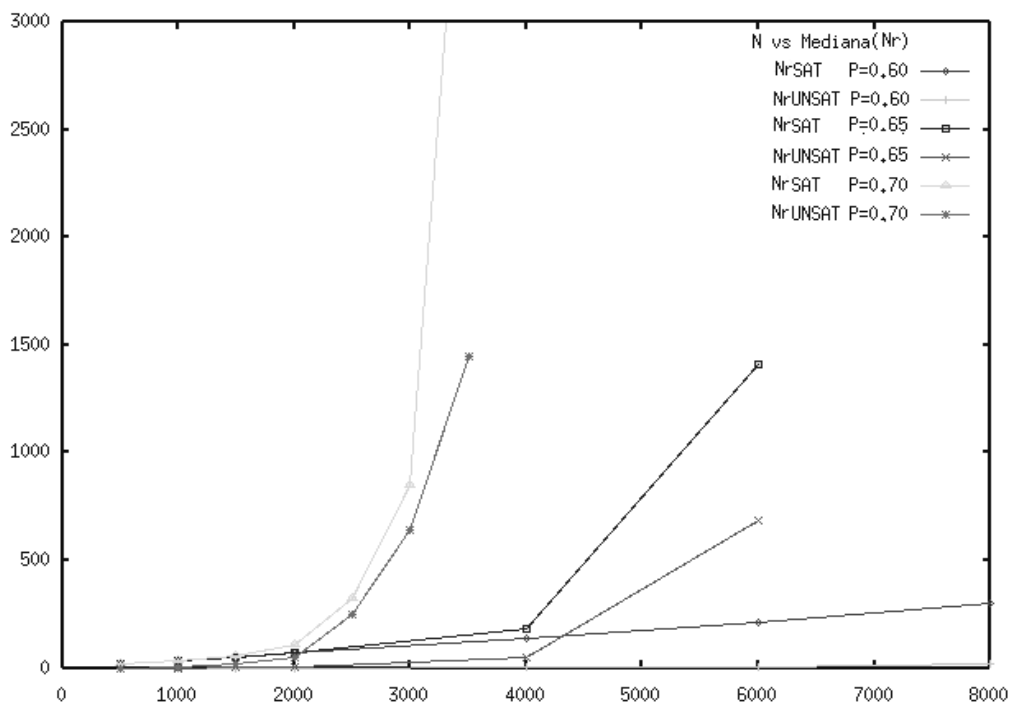


Figura 2a. Mediana del Número de ramificaciones para instancias satisfacibles (Nr -SAT) y no satisfacibles (Nr -UNSAT), en dependencia de N .

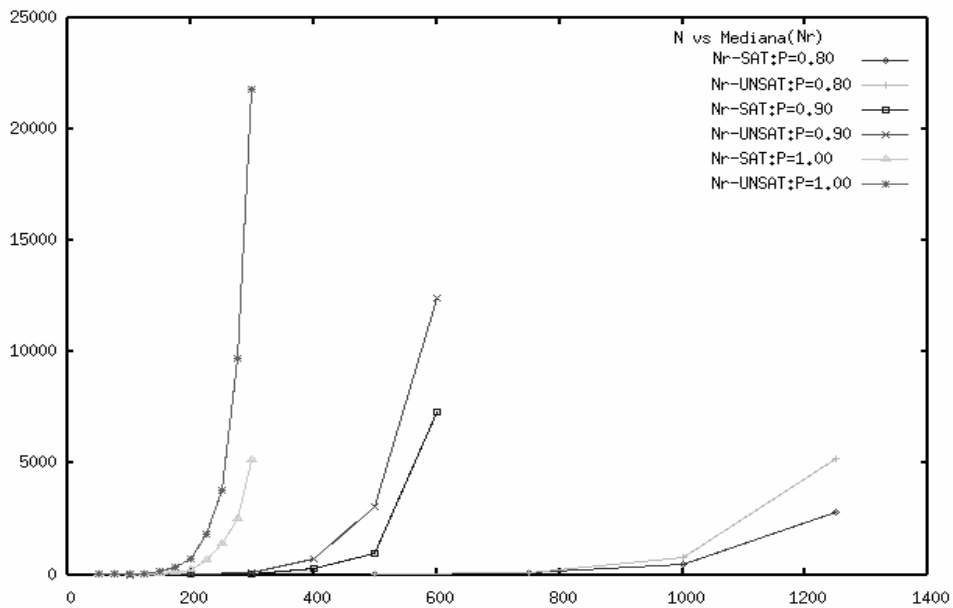


Figura 2b. Mediana del Número de ramificaciones para instancias satisfacibles (Nr-SAT) y no satisfacibles (Nr-UNSAT), en dependencia de N.

En la página <http://cs-95.cs.udea.edu.co/sat/results.html> se pueden descargar los archivos que contienen todos los resultados experimentales obtenidos.

4. CONCLUSIONES

El costo computacional de nuestra variante del algoritmo TABLEAU se evaluó sobre instancias del problema 2+p-SAT aleatorio, con valores de n muy superiores a los utilizados en otros estudios experimentales. Esta variante se desempeñó eficientemente en instancias con $p \geq 0.65$; no obstante, se observó una disminución de su eficiencia en instancias con $p \leq 0.60$ debida a la pérdida de efectividad del criterio de selección de variable, al reducir una fórmula 2+p-SAT aleatoria a una fórmula 2-SAT. El criterio de selección de variable de nuestra variante debe ser modificado para que continúe siendo efectivo en fórmulas 2-SAT.

Para valores $0.60 \leq p \leq 0.95$ se observaron instancias extremadamente difíciles, y su frecuencia de aparición tiene un pico en $0.65 \leq p \leq 0.70$. El máximo valor de Nr obtenido para $p = 0.65$ y $p = 0.70$ es significativamente mayor que el observado en todos los demás valores de p estudiados. Se requiere realizar más experimentos para establecer las causas del alto costo computacional observado en estas instancias, y su posible relación con el cambio del comportamiento del problema.

El costo típico de nuestro algoritmo se escaló de forma polinomial hasta $p = 0.60$ y de forma exponencial para $p = 0.65$. La región de comportamiento polinomial que hemos obtenido es significativamente más amplia que las obtenidas por otros autores; no obstante, el valor de p que determina la transición de polinomial a exponencial se mantiene dentro de las cotas teóricas para p_c obtenidas por Achlioptas [5]. Estos resultados mantienen la pregunta sobre la existencia de un único valor de p, para el cual los diferentes algoritmos comienzan a encontrar dificultades en la solución del problema 2+p-SAT.

Se observó que, para los valores de $p \leq 0.50$, todas las instancias no satisfacibles fueron resueltas con Nr = 0. Esto sugiere que nuestro algoritmo, en instancias 2+p-SAT con $p \leq 0.50$, se comporta igual que en instancias 2-SAT. Para establecer si la subfórmula 2-SAT determina la satisfacibilidad de una instancia 2+p-SAT aleatoria para estos valores de p, sería necesario realizar un estudio más profundo.

Los algoritmos completos recursivos, como el TABLEAU y nuestra variante, reducen un problema 3-SAT a subproblemas 2+p-SAT. Los resultados obtenidos, sobre el comportamiento del costo computacional del problema 2+p-SAT aleatorio, pueden ser aprovechados para diseñar mejores algoritmos de solución del problema 3-SAT aleatorio. El comportamiento de estas estrategias deberá analizarse en problemas 3-SAT que representen situaciones reales.

5. AGRADECIMIENTOS

Este trabajo es financiado por la Universidad de Antioquia, Colombia, a través del Proyecto CODI MC02-1-02 "Experimentos computacionales en instancias 2+p-SAT aleatorias".

REFERENCIAS

- GAREY, M. and D.S. JOHNSON (1979): **Computers and Intractability; A Guide to the Theory of NP-Completeness**, W. H. Freeman and Co., San Francisco.
- COOK, S. (1971): "The Complexity of Theorem Proving Procedures", **Proc. 3rd Ann. ACM Symp. on Theory of Computing**, 151-158.
- MONASSON, R. et al. (1999): "2+p-SAT: Relation of Typical-Case Complexity to the Nature of the Phase Transition", **Random Structure and Algorithms** 15, 414-440.
- _____ (1999): "Determining Computational Complexity from Characteristic Phase Transitions", **Nature** 400, 133-137.
- ACHLIOPTAS, D.; L.M. KIROUSIS; E. KRANAKIS and D. KRIZANC (2001): "Rigorous Results for Random (2+p)-SAT", **Theoretical Computer Sciences** 265(1-2), 109-129.
- CRAWFORD, J. M. and L.D. AUTON (1996): "Experimental Results on the Crossover Point in Random 3-SAT", **Artificial Intelligence** 81, 31-57.
- SINGER, J.; I.P. GENT and A. SMAILL (2000): "Local Search on Random 2+p-SAT", **Proceedings of the 14th ECAI**, European Conference on Artificial Intelligence.
- Mc ALLESTER, D.; B. SELMAN and H. KAUTZ (1997): "Evidence for Invariants in Local Search", **Proceedings of AAAI-97**, 321-326, AAAI Press/The MIT Press.
- DAVIS, M.; G. LOGEMANN and D. LOVELAND (1962): "A Machine Program for Theorem Proving", **CACM** 5, 394-397.