ALGORITMOS VORACES

Práctica 3

Alejandro Campoy Nieves
David Criado Ramón
Nour Eddine El Alaoui
Luis Gallego Quero

Contenido

1. Introducción

1.1 Algoritmos voraces

Un algoritmo voraz es aquel que, para resolver un determinado problema, sigue una heurística consistente en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima. Este esquema algorítmico es el que menos dificultades plantea a la hora de diseñar y comprobar su funcionamiento. Normalmente se aplica a los problemas de optimización. Para el funcionamiento de esta técnica, se consta con cinco elementos. Un conjunto C de candidatos (entradas del problema), la función solución que comprueba, en cada paso, si el subconjunto actual de candidatos elegidos forma una solución. La función selección informa de cuál es el elemento más prometedor para completar la solución, la función factibilidad informa si a partir de un conjunto se puede llegar a una solución y por fin la función objetivo que es aquella que determina el valor de una solución.

1.2 Presentación del problema

El problema planteado se base en dispositivo de almacenamiento de capacidad infinita (en la descripción dada una cinta) en la que se almacenan una serie de programas de los cuales sabemos dos datos: su tamaño (en kb) y su frecuencia de ejecución (tomando valores entre 0 y 1). Así pues y sabiendo que cuando se ejecuta un programa la cinta se rebobina al principio la función que nos da el tiempo medio y se pretende minimizar mediante una técnica voraz es la siguiente:

$$\boldsymbol{T = c \cdot \sum_{i=1}^{n} \left[\pi_i \cdot \sum_{j=0}^{i} s_j \right]}, \textit{donde } \boldsymbol{c} \textit{ es una constante}$$

Con el fin de minimizar la función y sabiendo que el orden creciente de i corresponde a la disposición de los programas en el dispositivo de almacenamiento trabajaremos sobre los siguientes criterios de ordenación para demostrar o no su optimalidad.

- Orden creciente de tamaño (s_i).
- Orden decreciente de frecuencia (π_i).
- Orden decreciente de frecuencia/tamaño (π_i/s_i).

2. Código

Nota: *El problema propuesto siempre tiene solución (aunque no siempre será óptima) por tanto no tiene sentido hablar de función de factibilidad o conjunto de seleccionados.*

2.1 Cálculo del tiempo medio

La siguiente función corresponde con la ***función solución.***

```
double tiempo_medio (const vector<programa> &v) {
  double tiempo = 0;
  double c = 0.000004;

  for (int i = 0; i < v.size(); ++i) {
    double sumalocal = 0;
    for (int j = 0; j <= i; ++j)
      sumalocal += get<KB>(v[j]);
    tiempo += get<FREC>(v[i]) * sumalocal;
  }

  return c * tiempo;
}
```

Nota: Con el fin de sacar números fáciles para la lectura humana el valor de la constante c, asociada a la densidad y velocidad de la cinta, es fijada a 0,000004.

El código aquí presentado no realiza nada más que la función matemática presentada previamente.

Cabe destacar que la programa es un alias para una tupla formada por 3 elementos. El primero es un identificador para el dato en el orden inicial, el segundo es el tamaño en kb del programa y el tercero es la frecuencia normalizada en la que puede se ejecuta el programa.

Así pues, para facilitar la legibilidad del código se utilizan las directivas de preprocesamiento #DEFINE para dar el valor 0 a ID, el 1 a KB y el 2 a FREC, correspondiéndose con el orden en el que se encuentran las tuplas predispuestas.

2.2 Generación aleatoria de tuplas

La siguiente función genera las muestras aleatorias que vamos a utilizar para la resolución del problema y por tanto dan lugar al ***conjunto de candidatos.***

```
vector<programa> generador(int N) {
  random_device rd;
  mt19937 mt(rd());
  uniform_int_distribution<> dist_kb(0,40000);
  uniform_int_distribution<> dist_prob(1,100);

  vector<programa> salida;
  vector<int> kb;
  vector<double> prob;

  kb.reserve(N);
  prob.reserve(N);
  salida.reserve(N);

  for (int i = 0; i < N; ++i) {
    kb.push_back(dist_kb(mt));
    prob.push_back(dist_prob(mt));
  }

  double total = accumulate(prob.begin(),prob.end(), 0,
plus<double>());

  for (auto& d: prob)
    d /= total;

  for (int i = 0; i < N; ++i)
     salida.emplace_back(i, kb[i], prob[i]);

  return salida;

}
```

Caben destacar tres cosas sobre esta función:

- Devuelve las tuplas ordenadas en orden creciente de ID.
- Los tamaños varían con números enteros entre 0 y 40000 KB.
- La probabilidad se encuentra normalizada entre 0 y 1 y suma exactamente 1 en total.

2.3 Main

El main nos muestra como recibimos por parámetro el tamaño de la muestra a utilizar:

```
int main(int argc, char *argv[])
{
   if (argc != 2) {
     cerr << "Introducir N como argumento";
     return -1;
   }

   int N = atoi(argv[1]);
  vector<programa> inicial = generador(N);
```

Y las diferentes variaciones en las que modificamos una función anónima (función lambda) que se corresponde con el criterio de ordenación y desde el punto de vista del algoritmo greedy con nuestra ***función selección.***

```
  vector<programa> creciente_tam(inicial);
  sort(creciente_tam.begin(),creciente_tam.end(), [](const programa& p1,
                                                     const programa& p2)
{
     return get<KB>(p1) < get<KB>(p2);
});
```

```
  vector<programa> decreciente_frec(inicial);
  sort(decreciente_frec.begin(),decreciente_frec.end(), []
                 (const programa& p1, const programa& p2) {
     return get<FREC>(p1) > get<FREC>(p2);
  });
```

```
  vector<programa> decreciente_div(inicial);
  sort(decreciente_div.begin(),decreciente_div.end(), []
              (const programa& p1, const programa& p2){
     return (get<FREC>(p1)/get<KB>(p1)) > (get<FREC>(p2) /get<KB>(p2));
  });
```

Para ayudarnos también ha obtener los datos utilizados para realizar las gráficas y disponibles en las tablas del Anexo I. Debido a la necesidad de que los datos generados utilicen la misma muestra nos apoyamos de *ofstream con el siguiente código:*

```
#ifdef MOSTRARPANTALLA
cout << "Para " << N << " programas."                              << endl;
cout << "Aleatorio: "        << tiempo_medio(inicial)           << endl;
cout << "Creciente tam: "    << tiempo_medio(creciente_tam)     << endl;
cout << "Decreciente frec: " << tiempo_medio(decreciente_frec) << endl;
cout << "Decreciente div: "  << tiempo_medio(decreciente_div)  << endl;
#endif

#ifdef ESCRIBIRARCHIVOS
   ofstream random ("aleatorio.dat",       ofstream::app);
   ofstream crectam("crecientetam.dat",    ofstream::app);
   ofstream decfrec("decrecientefrec.dat", ofstream::app);
   ofstream decdiv ("drecrecientediv.dat", ofstream::app);

   random  << N << " " << tiempo_medio(inicial)           << endl;
   crectam << N << " " << tiempo_medio(creciente_tam)     << endl;
   decfrec << N << " " << tiempo_medio(decreciente_frec) << endl;
   decdiv  << N << " " << tiempo_medio(decreciente_div)  << endl;
#endif
```

La siguiente sería una muestra de ejecución de nuestro programa pasando como argumento 25000.

```
Para 25000 programas.
Aleatorio: 993.194
Creciente tam: 662.774
Decreciente frec: 664.603
Decreciente div: 516.472
Press <RETURN> to close this window...
```

2.4 Modificaciones para contraejemplos

Con el objetivo de facilitar las demostraciones de los contraejemplos añadimos un trozo de código que imprime por pantalla el vector inicial y nos muestra cómo se encuentra el vector ordenado para los criterios dados.

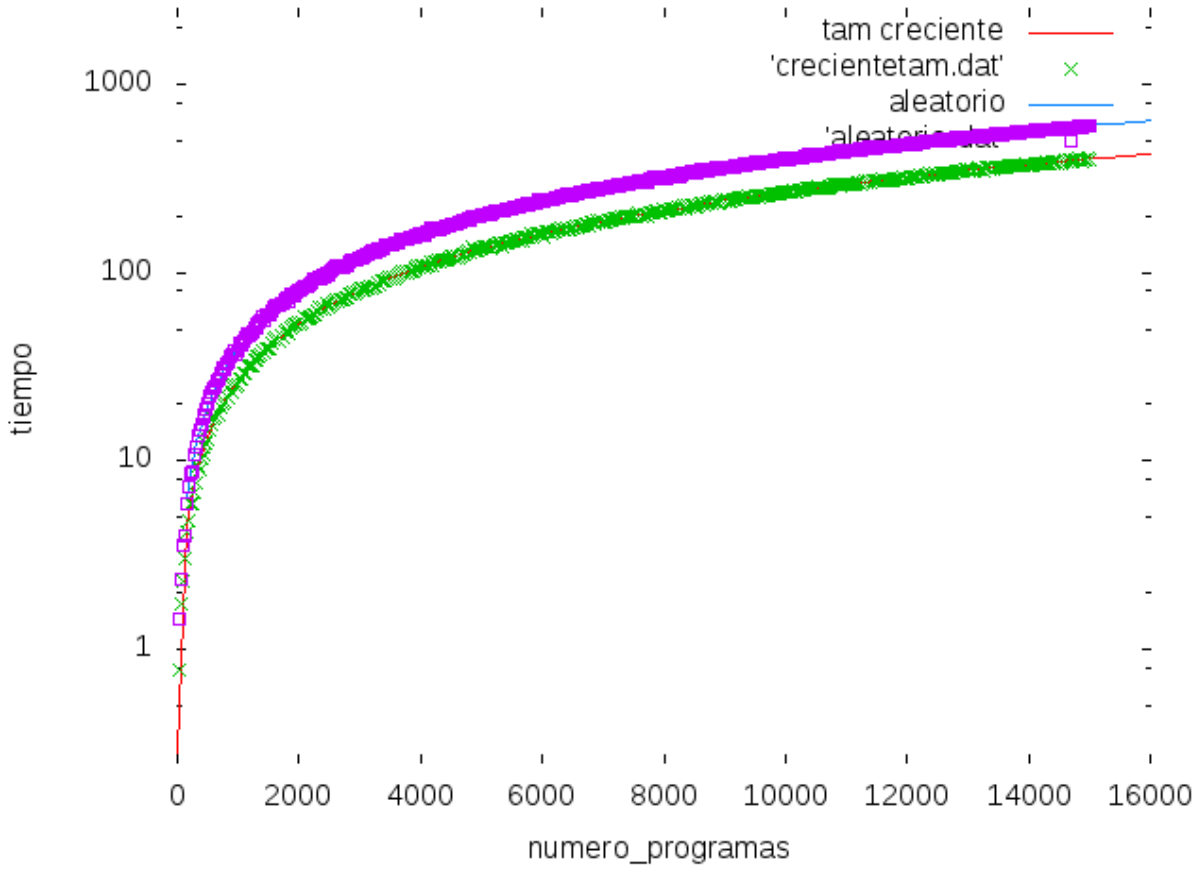
```
#if defined(CONTRAEJEMPLOTAM) || defined(CONTRAEJEMPLOFREC)
  cout << "INICIAL " << '\t';
  for (auto t : inicial)
    cout << get<ID>(t) << '\t';
  cout << endl << "CRECIENTE TAM" << '\t';
  for (auto t : creciente_tam)
    cout << get<ID>(t) << '\t';
  cout << endl << "DECRECIENTE FRE" << '\t';
  for (auto t : decreciente_frec)
    cout << get<ID>(t) << '\t';
  cout << endl << "DECRECIENTE DIV" << '\t';
  for (auto t : decreciente_div)
    cout << get<ID>(t) << '\t';
  cout << endl << endl << "MUESTRA" << endl;

  for (auto t: inicial)
    cout << get<ID>(t) << '\t' << get<KB>(t) << '\t' << get<FREC>(t) <<
endl;
#endif
```

Nota: Las gráficas que se obtienen a partir de los siguientes apartados se basan en los datos presentados en el Anexo I.

3. Orden creciente de tamaño de programa

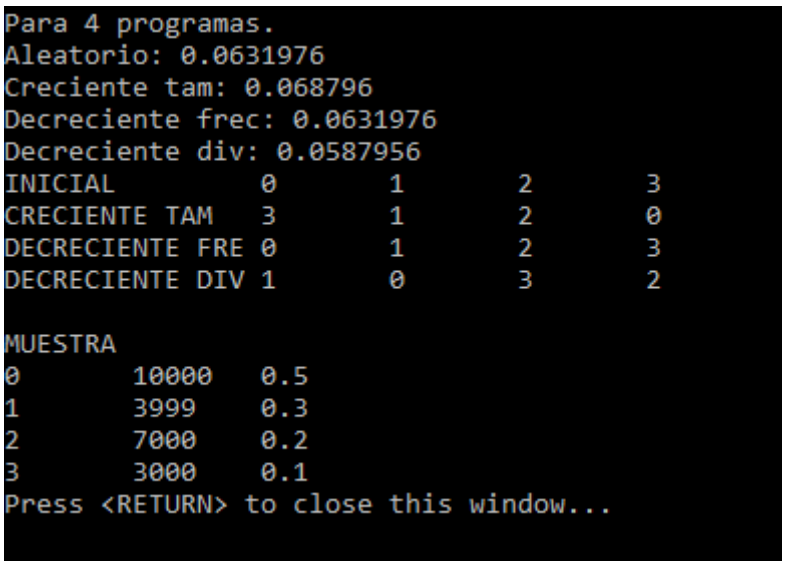
Semánticamente resulta fácil de deducir que por la definición del problema el tamaño del programa es un factor importante y el hecho de que la cinta se rebobine al principio tras cada ejecución nos lleva a la idea de que si tenemos que recorrer la cinta hasta un programa nos interesa que los primeros programas sean los menos pesados o desde un punto de vista matemático ya que la sumatoria va a ser realizada muchas veces (*i veces)* nos interesa que los valores dentro de esa sumatoria queden ordenados de menor a mayor para evitar añadir números innecesarios. La siguiente gráfica nos permite verificar que efectivamente es una buena solución.



La pregunta ahora es, ¿es ésta una solución óptima? La respuesta es no. No es muy difícil llegar a la conclusión de que si ponemos un valor con una gran frecuencia y un gran peso veremos cómo los valores se disparan. Para demostrarlo utilizaremos el siguiente contraejemplo, deshabilitando los valores previamente introducidos y metiendo los valores propuestos en la siguiente tabla.

ID	0	1	2	3
TAMAÑO	10000	3999	7000	3000
FRECUENCIA	0.5	0.3	0.2	0.1

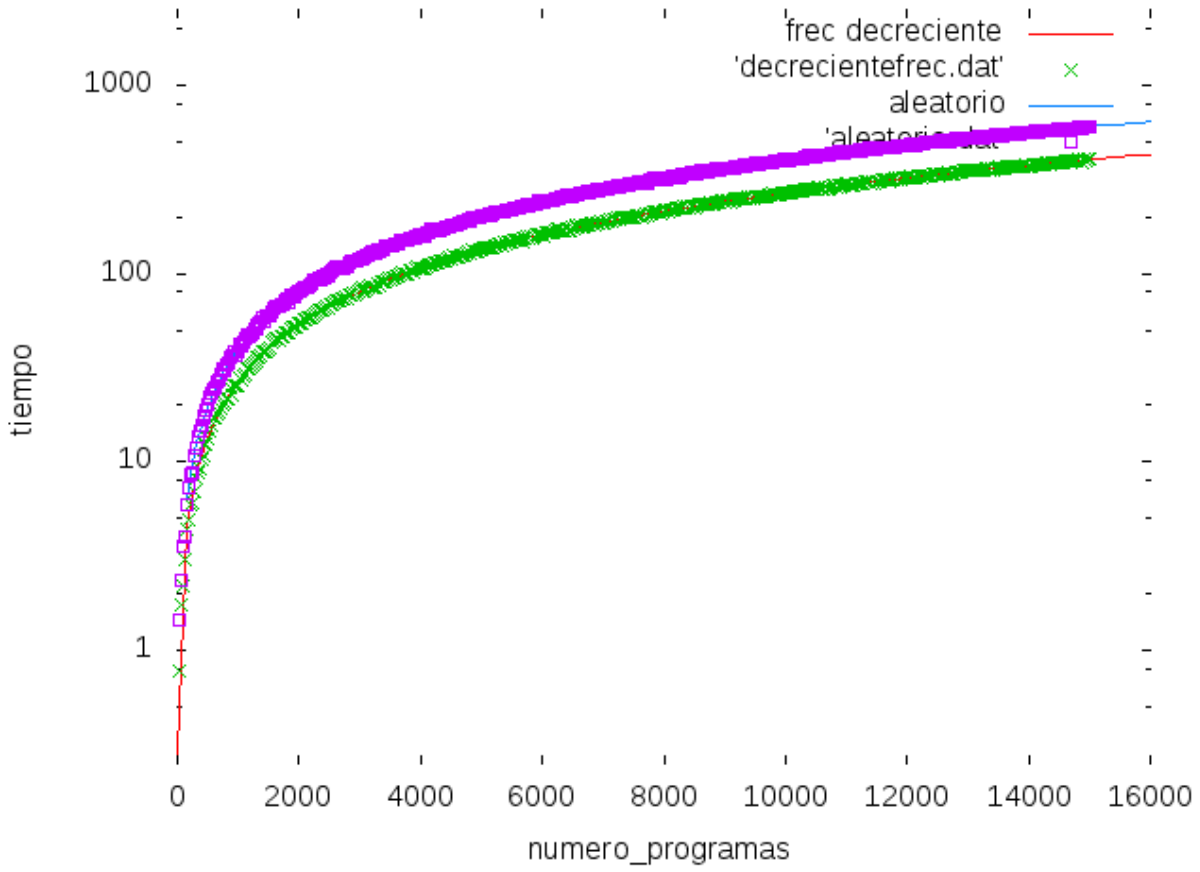
El ejemplo de ejecución para dichos valores sería el siguiente:



Con el ejemplo vemos que obtenemos el menor tiempo para la versión decreciente de la división entre la frecuencia y el tamaño, seguido de la versión decreciente en frecuencia que corresponde con el orden llamado “aleatorio” en que los hemos introducido y por último queda el orden creciente de tamaño, por lo que podemos asegurar que el algoritmo no es óptimo.

4. Orden decreciente de frecuencia de ejecución

Como hemos visto en el contraejemplo anterior otro de los factores que afectan al cálculo del tiempo medio es la frecuencia de ejecución. Desde un punto de vista puramente semántico es lógico pensar que si la cinta se rebobina tras cada ejecución nos interesa tener los programas que se ejecutan con más frecuencia al principio de la misma para que el número de elementos de la sumatoria con la que se multiplica dicha frecuencia sea el menor posible. La siguiente gráfica nos muestra una comparación entre los datos iniciales y los datos ordenados mediante este criterio:



Análogamente al caso anterior nos preguntamos, ¿este algoritmo garantiza siempre la solución óptima? La respuesta es otra vez no. Si para una frecuencia elevada tenemos un número extremadamente elevado de KB es posible que aun siendo muy frecuente no compense que esté al principio. Aunque incluso en el contraejemplo de antes veíamos como quedaba de manifiesto que había otro método que era mejor realizamos otra prueba con datos más extremos para el nuevo criterio.

ID	0	1	2	3
TAMAÑO	10000	3999	7000	3000
FRECUENCIA	0.5	0.3	0.2	0.1

El ejemplo de ejecución para dichos valores sería el siguiente:

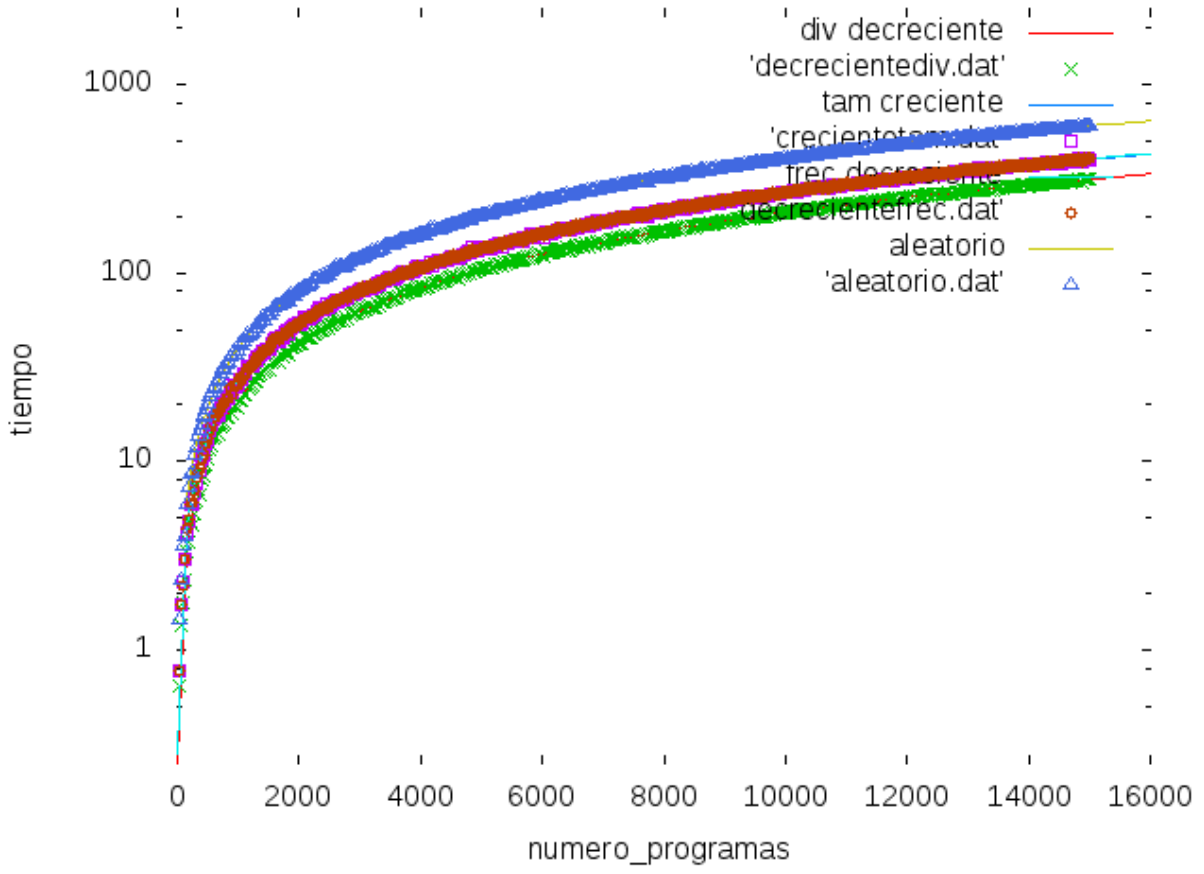
```
Para 4 programas.
Aleatorio: 6.03312
Creciente tam: 5.87432
Decreciente frec: 6.04312
Decreciente div: 5.81432
INICIAL         0       1       2       3
CRECIENTE TAM   1       3       2       0
DECRECIENTE FRE 0       2       1       3
DECRECIENTE DIV 1       3       0       2

MUESTRA
0       1500000 0.9
1       10000   0.03
2       100000  0.05
3       14000   0.02
Press <RETURN> to close this window...
```

Como podemos ver en este caso el más lento de todos es el decreciente en frecuencia, por detrás el inicial, el creciente en tamaño y el decreciente de la división entre frecuencia y tamaño en ese mismo orden. Por tanto, podemos garantizar con total seguridad que este criterio voraz no es una solución óptima para el problema.

5. Orden decreciente frecuencia entre tamaño

Hemos visto como los dos factores no constantes de la fórmula que daba al tiempo medio de acceso han afectado positivamente en reducir dicho tiempo. No obstante, ninguno de ellos por su cuenta propia ha sido suficiente por sí sólo para dar una solución siempre óptima al problema y por tanto nos queda una única opción, ¿y si buscamos un criterio de ordenación que incluya a ambos? Semánticamente hemos visto que nos interesa que los valores más cercanos al inicio tengan la mayor frecuencia posible pero el menor tamaño. El hecho de que la frecuencia sea directamente proporciona garantiza que a mayor frecuencia más valdrá dicho valor y como el tamaño es inversamente proporcional a menor tamaño más valdrá dicho valor con lo que nos interesa que esté en orden decreciente. Como ahora para seleccionar tenemos en cuenta los dos factores y ningún otro factor afecta al tiempo medio podemos ver que la solución que obtenemos siempre es la óptima. En los ejemplos anteriores ya hemos visto que es la que ha tomado ventaja respecto a la otra en ambos casos y ahora para confirmarlo veamos la siguiente gráfica en la que comparamos todos los criterios propuestos. En el Anexo I podemos comprobar numéricamente cómo para todos los valores este último criterio es el mejor.



Anexo I – Datos utilizados para gráficas

La siguiente tabla presenta los 500 datos obtenidos tomando datos de 30 en 30 hasta 15000 y midiendo el tiempo con orden aleatorio y con los criterios previamente propuestos.

N PROGRAMAS	INICIAL	TAM ↑	FREC ↓	FREC/TAM
30	1,4532	0,778731	0,777038	0,652239
60	2,33208	1,72977	1,74641	1,345
90	3,60812	2,30421	2,2112	1,80613
120	4,01799	3,051	3,01462	2,33034
150	5,95315	4,15295	4,39561	3,32869
180	7,31762	4,84585	4,93393	3,75444
210	8,64752	5,88129	5,99392	4,76746
240	8,86382	5,92371	6,13361	4,57625
270	10,8618	6,89087	7,03887	5,29422
300	11,7818	7,6169	8,19041	6,27779
330	13,5576	9,43262	8,85074	7,15428
360	14,5123	8,90635	9,13803	6,73079
390	15,7957	10,4267	10,3942	8,00631
420	17,0406	10,8421	10,8378	8,30872
450	17,6019	11,8996	12,2923	9,47509
480	18,6224	12,4335	12,3206	9,6516
510	20,099	12,9888	13,3249	10,2421
540	22,3211	14,5674	14,4418	11,4692
570	23,4223	15,7989	15,4884	12,1562
600	24,5394	16,7077	17,0755	13,2769
630	24,7329	17,1919	17,3364	13,5131
660	26,3464	18,1573	18,148	14,1852
690	26,9748	17,6165	18,2059	13,8821
720	29,2767	19,295	19,3592	15,193
750	31,2638	20,0492	19,9722	15,5453
780	30,8826	20,7349	20,6209	15,911
810	32,8226	22,0968	21,8401	16,9764
840	33,2474	21,4262	21,8611	16,9842
870	36,0427	25,2225	24,677	19,6666
900	36,4231	22,9769	22,7303	17,8552
930	38,5196	25,1765	25,3944	19,8882
960	36,6244	24,6834	25,3536	19,5571
990	38,8295	25,1662	25,3866	19,4782
1020	41,7876	27,293	27,6016	21,3015
1050	41,4659	27,4654	26,9311	21,1126
1080	43,2072	29,1129	28,6683	22,5231
1110	44,9398	29,2545	28,9406	22,6053
1140	46,9449	32,0576	31,3373	24,831
1170	46,5563	31,6948	31,2424	24,409

1200	47,0959	32,277	32,2561	24,7101
1230	48,8816	32,225	32,6146	25,6622
1260	50,5394	34,6205	33,3527	26,5599
1290	51,0153	34,2845	34,147	26,6569
1320	52,8977	36,0885	36,141	28,3432
1350	53,9926	35,9705	36,3207	28,2405
1380	55,1672	36,4985	36,987	28,9327
1410	58,2346	38,4408	38,6755	30,2685
1440	56,1078	37,7656	38,3819	29,867
1470	59,2775	39,4106	38,7251	30,4907
1500	59,5773	39,6906	39,7961	31,1536
1530	60,1343	39,4052	39,8672	30,5609
1560	62,6998	42,4017	42,918	33,2583
1590	64,9894	44,0997	43,8318	34,4433
1620	66,4298	44,8861	45,4884	35,3842
1650	66,3166	44,1666	44,8729	34,5176
1680	66,7439	43,2007	43,8871	33,89
1710	68,6616	46,4225	46,444	36,4364
1740	68,0343	45,2117	46,2147	35,5522
1770	71,749	48,5881	48,4735	37,4925
1800	74,1635	49,6112	48,5835	38,5188
1830	69,5095	46,9611	48,0955	36,8702
1860	76,3613	50,5259	51,0116	40,1332
1890	74,9184	51,7308	51,7698	40,6719
1920	75,6821	52,9296	52,3491	41,4548
1950	79,1134	52,7087	52,5412	40,8554
1980	81,0632	54,363	54,5213	42,4336
2010	80,844	53,1329	53,9791	41,859
2040	82,125	54,6718	54,4125	42,4201
2070	84,1068	58,2782	57,4593	45,8834
2100	84,1937	56,6395	57,3538	44,8972
2130	84,7772	57,383	58,6971	45,6609
2160	85,8914	57,7257	57,9333	45,1707
2190	86,5345	57,2766	58,0525	45,1573
2220	87,9825	57,9934	58,9017	45,2923
2250	91,971	60,002	60,1091	46,9303
2280	92,4709	60,0439	60,6422	47,097
2310	92,6685	62,3105	62,3407	48,4166
2340	95,4956	64,9353	63,4253	50,0836
2370	93,6028	63,6648	64,5589	49,9089
2400	96,4336	65,1055	64,1529	50,134
2430	94,3965	64,5135	65,339	50,7516
2460	98,2665	67,9312	67,496	53,2466
2490	96,7688	65,5293	66,0529	51,1511
2520	101,853	67,7393	68,4558	53,5337
2550	102,959	68,3248	67,8426	53,0392

2580	105,164	70,0626	69,497	54,3843
2610	107,952	72,4588	71,5541	56,3318
2640	105,607	70,2381	71,3028	55,695
2670	107,864	71,2034	71,1626	55,2588
2700	109,559	72,5774	73,9942	57,2826
2730	106,731	72,1548	72,9021	56,2583
2760	109,566	73,6008	74,3479	58,0191
2790	109,38	74,2881	74,1441	57,3836
2820	112,83	76,5117	77,805	60,6888
2850	116,281	77,0948	77,2849	60,3086
2880	116,161	78,4716	77,9613	60,776
2910	116,542	77,3847	77,6647	60,5607
2940	116,413	78,0859	78,854	60,8711
2970	118,141	79,8114	79,7263	62,0288
3000	120,061	81,9236	81,6143	63,6557
3030	119,923	81,2229	80,9756	63,0919
3060	121,955	83,1987	84,2968	65,4296
3090	122,875	81,4056	81,3749	63,3048
3120	125,778	82,3918	82,3458	64,0279
3150	125,64	83,4168	84,117	64,9546
3180	128,062	85,1848	85,5249	66,6441
3210	129,45	83,2843	84,3146	65,4095
3240	128,667	84,1388	85,353	65,688
3270	128,798	84,2858	85,2101	65,7975
3300	132,848	89,2067	88,8092	69,0817
3330	131,791	86,3371	86,713	67,1475
3360	134,92	90,2427	90,3032	70,3308
3390	133,938	90,1134	90,7292	70,2801
3420	138,765	90,3144	90,3588	70,7132
3450	140,804	92,0236	92,2083	72,3349
3480	140,407	94,0531	94,1087	73,7316
3510	139,898	92,9333	93,3622	72,5168
3540	140,759	95,0173	95,4793	73,9823
3570	142,411	93,3183	95,3166	73,5749
3600	142,118	95,1017	96,1074	74,1085
3630	145,018	96,9151	96,8026	75,0991
3660	150,343	100,055	99,9279	78,1004
3690	150,632	101,602	101,616	79,6516
3720	149,816	99,0937	99,3831	77,5255
3750	152,003	101,426	99,6862	77,8582
3780	150,235	100,224	100,09	78,2351
3810	152,954	102,394	103,693	80,7898
3840	153,66	102,438	104,052	80,5515
3870	155,193	101,466	102,088	79,6812
3900	158,353	106,856	105,538	83,169
3930	155,566	103,106	105,365	81,2061

3960	157,831	108,35	109,244	85,1327
3990	159,549	105,373	106,131	82,6223
4020	160,333	108,076	107,352	84,3976
4050	161,825	107,929	107,168	83,9964
4080	159,009	105,719	107,268	82,5984
4110	163,922	108,915	108,364	84,8707
4140	164,41	108,791	110,414	85,1518
4170	171,003	114,157	112,158	88,7011
4200	167,894	110,013	111,314	85,6433
4230	171,121	114,432	113,119	88,6241
4260	170,295	113,572	113,432	88,1857
4290	170,436	115,312	115,153	89,7482
4320	169,103	113,116	115,538	89,331
4350	174,156	116,454	117,529	91,07
4380	174,537	116,769	116,406	90,8838
4410	176,51	117,264	116,113	91,1966
4440	178,072	117,58	118,518	91,7041
4470	177,769	120,601	121,05	94,6017
4500	181,845	120,767	120,504	94,3901
4530	180,275	122,562	123,84	96,7611
4560	181,162	119,71	121,497	94,2546
4590	182,398	121,774	122,529	95,1424
4620	184,415	119,371	121,327	93,4769
4650	185,57	122,834	123,731	96,4992
4680	187,433	125,014	126,089	98,067
4710	188,904	125,453	125,481	97,5627
4740	190,03	127,732	127,685	100,104
4770	191,793	129,48	129,773	101,122
4800	191,23	126,886	127,321	98,7507
4830	191,613	127,103	128,95	99,8448
4860	200,204	135,683	134,308	104,993
4890	196,253	131,98	131,706	102,654
4920	196,86	130,841	131,978	102,192
4950	198,771	132,286	132,35	103,782
4980	198,738	134,211	133,544	104,383
5010	201,75	134,674	135,331	105,173
5040	201,938	136,34	136,043	105,931
5070	202,09	137,662	137,86	107,369
5100	204,962	135,944	136,108	106,138
5130	205,981	137,844	138,181	107,85
5160	203,161	135,439	136,51	105,93
5190	205,558	137,084	137,182	106,39
5220	212,009	140,881	141,645	110,183
5250	212,689	140,119	140,978	109,496
5280	212,225	143,84	143,192	112,227
5310	212,387	143,419	143,174	111,793

5340	210,78	138,026	141,057	108,52
5370	212,151	143,799	145,548	113,056
5400	215,815	143,378	144,272	112,223
5430	218,564	144,532	145,555	113,365
5460	217,968	145,96	146,671	113,891
5490	219,822	147,214	147,33	114,997
5520	222,408	148,39	149,087	116,085
5550	224,765	150,407	150,038	117,301
5580	223,124	150,507	150,021	116,78
5610	223,215	148,471	149,28	115,957
5640	229,472	151,338	150,384	117,682
5670	229,973	154,472	153,507	120,343
5700	226,862	148,486	148,53	115,838
5730	228,1	152,743	153,666	119,257
5760	230,856	155,242	156,344	121,45
5790	228,802	152,799	153,462	118,633
5820	234,564	159,597	156,506	123,199
5850	233,519	155,562	154,438	120,143
5880	235,185	156,885	157,132	121,61
5910	235,296	160,176	159,746	124,532
5940	241,648	162,113	162,667	127,199
5970	241,916	162,081	162,186	126,075
6000	237,331	160,258	161,167	125,792
6030	240,317	155,583	157,351	121,47
6060	245,786	164,806	165,853	129,446
6090	242,702	163,094	163,239	127,727
6120	245,89	164,591	164,518	128,372
6150	245,561	161,982	163,662	127,398
6180	245,074	164,05	165,677	128,751
6210	251,565	166,068	168,056	130,13
6240	252,11	169,337	170,875	133,028
6270	249,403	169,566	171,158	133,356
6300	255,262	171,182	169,663	132,491
6330	252,82	168,999	169,998	132,863
6360	257,769	168,921	170,463	131,734
6390	259,741	173,664	172,39	135,958
6420	254,194	167,177	170,011	131,295
6450	256,009	168,291	169,934	131,844
6480	256,14	173,757	175,502	135,013
6510	262,099	173,465	173,406	135,678
6540	261,608	177,576	177,627	138,259
6570	262,233	177,43	178,87	139,565
6600	262,726	174,089	175,965	136,184
6630	267,233	177,938	178,964	139,537
6660	267,433	177,873	178,371	138,84
6690	266,768	176,286	178,239	137,828

6720	270,587	182,7	181,295	142,053
6750	272,652	179,22	181,104	140,929
6780	269,763	176,892	179,696	138,601
6810	273,39	183,773	183,66	143,696
6840	273,228	183,453	183,268	143,396
6870	274,037	181,097	182,754	140,942
6900	280,593	188,201	186,942	146,813
6930	275,786	183,257	183,807	143,122
6960	282,498	188,989	189,872	148,309
6990	279,798	185,722	185,456	144,273
7020	280,096	185,442	185,486	144,945
7050	280,476	184,894	187,448	145,011
7080	282,626	187,88	188,803	146,712
7110	285,226	190,007	191,529	148,905
7140	287,148	189,797	191,2	149,042
7170	287,193	192,111	193,65	150,328
7200	287,662	192,332	191,877	149,222
7230	291,476	194,727	196,088	153,28
7260	289,479	194,867	194,242	151,403
7290	290,202	196,681	197,38	153,759
7320	296,601	195,073	196,04	152,666
7350	293,84	197,416	199,393	155,625
7380	294,571	196,008	197,254	152,558
7410	297,573	197,038	199,133	153,998
7440	295,175	198,223	198,943	154,367
7470	301,583	199,506	200,49	156,133
7500	298,303	198,501	198,732	154,512
7530	301,419	199,432	200,275	155,097
7560	299,258	202,088	201,236	156,359
7590	306,451	205,29	204,365	160,184
7620	303,923	204,688	203,088	158,827
7650	302,006	198,935	200,322	155,159
7680	307,339	204,18	204,561	158,902
7710	308,143	205,212	208,278	161,528
7740	302,169	199,556	202,973	156,245
7770	311,189	208,991	208,814	162,534
7800	313,396	207,416	208,652	161,474
7830	318,354	208,002	210,988	164,204
7860	318,047	213,465	214,183	166,589
7890	316,581	210,177	211,362	164,252
7920	315,883	208,172	211,018	163,371
7950	311,976	208,733	209,962	162,983
7980	320,306	215,296	214,354	166,326
8010	321,194	214,44	215,467	167,636
8040	317,064	213,831	215,097	166,04
8070	320,248	210,364	212,877	165,781

8100	324,889	217,083	218,388	169,661
8130	326,557	218,232	219,358	169,679
8160	325,424	215,659	218,112	168,785
8190	327,245	219,597	221,224	172,825
8220	329,238	222,783	223,315	174,242
8250	320,888	217,37	220,488	170,127
8280	334,949	221,719	222,746	172,857
8310	329,771	221,278	224,227	173,453
8340	338,082	228,519	227,228	178,35
8370	333,325	219,32	221,279	170,901
8400	339,002	225,455	226,624	176,296
8430	338,399	223,225	223,556	173,9
8460	342,455	228,047	227,323	177,337
8490	339,133	228,276	230,341	178,337
8520	340,289	222,697	223,672	173,739
8550	337,575	223,941	228,502	175,528
8580	341,702	230,179	231,015	179,632
8610	353,043	233,875	233,799	182,605
8640	352,826	231,426	232,567	181,155
8670	344,045	230,037	231,558	179,605
8700	349,472	233,499	233,961	181,96
8730	347,969	233,947	233,165	181,763
8760	352,97	236,85	235,927	184,571
8790	350,862	233,604	235,15	182,86
8820	349,855	235,782	235,346	182,6
8850	357,643	240,348	239,106	185,941
8880	356,203	235,386	236,951	184,389
8910	356,274	232,927	236,246	183,086
8940	360,765	240,841	242,467	188,355
8970	363,324	244,09	242,733	190,142
9000	359,613	236,992	239,17	186,15
9030	362,561	245,03	243,569	190,706
9060	361,489	240,502	242,424	188,802
9090	358,361	238,96	241,085	186,098
9120	361,794	242,598	246,324	190,452
9150	365,658	241,462	243,892	189,102
9180	364,545	244,308	246,981	191,048
9210	367,493	245,208	246,931	192,227
9240	373,059	247,744	247,497	192,008
9270	372,303	247,633	247,33	192,856
9300	373,663	248,658	250,581	195,31
9330	377,387	252,057	251,885	196,681
9360	371,029	250,47	251,792	196,327
9390	380,68	255,696	255,877	199,335
9420	376,599	249,473	251,72	194,765
9450	378,408	252,34	254,325	196,764

9480	379,296	250,191	250,77	194,574
9510	382,877	254,603	256,858	200,026
9540	387,762	256,811	257,171	200,562
9570	380,654	251,264	254,652	196,807
9600	384,344	253,14	255,802	198,684
9630	386,19	255,511	257,438	200,03
9660	384,243	252,835	254,744	196,898
9690	385,142	257,005	258,518	200,104
9720	385,915	254,684	260,182	201,31
9750	388,172	259,483	260,696	202,962
9780	391,724	264,432	264,592	206,267
9810	397,919	260,326	262,055	203,225
9840	397,473	262,621	264,523	205,578
9870	392,957	264,123	263,993	205,345
9900	395,152	263,977	263,915	205,581
9930	400,3	267,499	267,71	209,212
9960	401,442	266,472	268,323	208,292
9990	399,153	266,225	265,778	207,714
10020	397,271	265,817	266,547	207,379
10050	407,787	269,794	269,962	210,501
10080	399,127	264,685	268,033	207,784
10110	406,408	269,714	269,224	210,283
10140	406,467	271,439	273,603	212,901
10170	403,952	269,475	271,517	210,968
10200	404,041	269,546	272,5	211,997
10230	406,33	272,145	274,283	213,098
10260	405,304	273,116	276,756	214,776
10290	407,238	273,397	275,974	214,078
10320	413,178	274,699	277,338	215,936
10350	417,349	277,447	277,817	216,743
10380	420,664	280,78	282,392	219,548
10410	417,331	278,342	280,118	217,759
10440	420,648	279,227	281,282	218,313
10470	421,935	280,101	279,022	217,141
10500	417,247	278,285	279,44	217,574
10530	427,957	285,722	284,075	223,005
10560	424,144	282,234	284,685	221,781
10590	423,4	280,775	282,902	220,177
10620	421,784	278,833	279,884	217,228
10650	422,931	281,484	285,856	221,77
10680	424,967	281,93	283,367	220,588
10710	425,647	282,058	283,208	219,611
10740	431,903	288,438	287,777	225,224
10770	434,718	290,515	291,447	226,674
10800	432,805	290,185	291,002	226,919
10830	436,973	290,806	289,587	226,065

10860	439,976	292,462	293,886	228,853
10890	434,493	285,595	286,328	222,284
10920	444,019	292,653	291,995	227,574
10950	439,391	295,401	295,328	231,226
10980	434,602	292,79	292,208	228,35
11010	440,96	295,467	297,222	231,376
11040	442,051	294,517	297,266	230,149
11070	441,757	296,557	296,784	230,454
11100	441,342	293,947	296,15	230,725
11130	443,675	295,93	296,409	231,262
11160	442,596	296,515	297,91	230,97
11190	447,258	300,858	300,228	233,969
11220	453,848	302,706	303,877	236,833
11250	447,449	294,917	298,068	230,907
11280	453,336	299,047	299,073	232,163
11310	453,911	306,791	306,648	239,125
11340	453,137	302,753	304,587	237,433
11370	460,666	306,34	304,617	238,104
11400	451,224	298,834	301,856	234,293
11430	462,65	309,34	310,834	241,894
11460	460,342	305,497	303,482	237,73
11490	456,922	304,196	306,808	238,315
11520	462,272	308,709	310,001	241,361
11550	461,57	305,174	308,882	238,6
11580	467,44	314,61	315,458	245,534
11610	463,477	308,005	311,114	242,583
11640	461,998	307,601	311,354	240,746
11670	464,687	309,342	312,974	242,542
11700	463,637	307,424	310,882	241,476
11730	467,307	310,677	314,888	243,545
11760	468,614	310,671	312,791	242,356
11790	470,775	316,713	317,579	247,536
11820	470,727	315,164	318,532	247,534
11850	472,118	314,287	316,876	245,288
11880	476,634	317,629	319,159	248,782
11910	475,829	315,038	317,025	246,792
11940	474,252	318,388	321,509	249,578
11970	477,538	317,34	320,967	248,835
12000	479,281	317,719	319,377	248,993
12030	478,049	316,316	319,479	247,279
12060	487,52	324,811	326,031	254,254
12090	483,003	319,712	321,468	250,526
12120	488,887	324,791	323,659	253,411
12150	480,679	318,679	323,133	249,543
12180	486,803	323,107	328,334	254,244
12210	490,381	326,44	326,605	253,294

12240	488,455	324,009	324,8	253,238
12270	488,676	325,201	328,567	253,881
12300	494,654	330,434	332,806	259,355
12330	495,808	330,468	332,31	259,361
12360	495,913	333,291	331,971	259,194
12390	496,443	329,008	329,628	256,877
12420	497,636	331,533	334,306	259,694
12450	504,716	333,625	335,126	260,738
12480	501,414	331,97	334,287	259,325
12510	500,671	329,53	332,57	258,119
12540	507,261	337,991	339,941	264,245
12570	503,935	334,155	335,477	262,064
12600	504,925	334,936	337,758	262,368
12630	504,709	338,702	340,138	264,428
12660	506,253	338,676	340,281	264,326
12690	505,72	338,686	342,955	266,005
12720	504,703	335,848	337,954	262,048
12750	508,656	337,959	339,729	263,877
12780	512,317	337,201	335,136	261,586
12810	517,182	345,817	347,13	269,266
12840	518,525	347,857	345,453	270,482
12870	514,683	344,794	346,372	269,156
12900	515,275	346,587	345,59	269,99
12930	519,239	345,991	349,042	271,852
12960	520,824	350,163	349,324	272,12
12990	521,985	350,222	352,264	273,987
13020	523,502	346,81	349,262	272,735
13050	523,705	350,167	351,478	273,525
13080	523,597	348,582	350,481	273,385
13110	524,701	346,965	349,607	272,287
13140	525,325	351,108	352,433	274,726
13170	531,927	358,413	356,874	278,256
13200	528,336	349,844	351,477	273,747
13230	523,78	349,455	350,623	271,943
13260	523,421	348,717	349,11	271,668
13290	532,601	359,573	358,992	280,262
13320	535,993	358,981	358,802	279,83
13350	530,734	352,493	354,79	275,949
13380	537,542	359,145	359,71	280,026
13410	530,482	352,085	356,7	275,577
13440	542,501	359,077	359,019	280,933
13470	541,028	360,162	359,6	280,591
13500	543,227	364,343	366,544	285,549
13530	540,764	359,121	362,87	281,476
13560	543,057	365,378	365,095	285,694
13590	541,072	360,303	361,27	280,213

13620	545,152	364,624	365,073	284,437
13650	544,049	365,322	368,421	286,329
13680	546,459	369,328	369,209	287,641
13710	545,514	368,761	367,832	286,746
13740	547,45	369,265	370,457	288,082
13770	549,7	366,98	369,244	287,017
13800	560,982	369,596	371,058	288,463
13830	553,373	367,926	372,49	289,944
13860	550,521	369,262	370,594	288,515
13890	556,457	373,019	373,706	290,997
13920	557,91	373,007	373,958	290,381
13950	564,603	377,804	377,907	294,675
13980	558,801	368,945	371,994	289,164
14010	554,027	367,017	368,888	286,37
14040	564,184	372,028	373,779	291,315
14070	557,01	371,531	375,225	290,559
14100	571,874	377,887	378,467	295,57
14130	565,198	378,406	380,227	295,084
14160	566,452	375,105	377,341	293,086
14190	572,756	380,33	380,409	295,451
14220	568,395	376,976	380,41	295,099
14250	570,671	380,691	381,781	297,829
14280	574,346	383,038	383,489	299,422
14310	567,644	380,588	384,645	298,027
14340	573,401	380,885	382,608	298,485
14370	573,746	386,686	387,866	301,689
14400	571,244	381,648	384,462	298,406
14430	576,726	382,875	386,817	300,544
14460	575,718	386,061	386,622	300,787
14490	582,024	382,6	386,749	300,651
14520	579,251	388,713	390,952	303,832
14550	579,937	381,741	387,289	300,256
14580	571,718	387,354	389,6	303,422
14610	587,1	395,661	394,974	307,489
14640	584,822	390,699	392,075	304,105
14670	585,047	392,615	393,501	306,061
14700	591,266	396,185	398,051	309,354
14730	585,911	391,118	393,115	306,143
14760	592,664	396,252	396,396	308,761
14790	590,562	394,972	397,209	309,526
14820	595,85	404,526	404,129	315,638
14850	587,593	394,498	396,49	309,242
14880	587,661	387,569	390,622	302,578
14910	595,694	398,376	401,338	311,468
14940	600,412	399,272	398,423	311,147
14970	601,4	402,807	403,046	314,005

15000	599,79	400,507	401,292	313,125