



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA Y
TELECOMUNICACIONES

PRACTICA 2.B: TÉCNICAS DE BÚSQUEDA BASADAS EN POBLACIONES PARA EL
PROBLEMA DEL APRENDIZAJE DE PESOS EN CARACTERÍSTICAS
ALGORITMO AGG
ALGORITMO AGE
ALGORITMO AM(10, 0.1/1/0.1mejor)

Realizado por:
González Mairena, Juan Jesús
DNI: 77206717H
jiuxej@correo.ugr.es
3º CURSO INGENIERÍA INFORMÁTICA
GRUPO: 3 - JUEVES(17:30-19:30)

Contents

1	Descripción del problema	2
2	Descripción de funciones generales aplicadas al problema	2
3	Descripción funciones específicas para AG y AM	5
3.1	Operador de cruce	6
3.2	Operador de selección	7
3.3	Mutación	7
4	Algoritmos Genéticos	7
4.1	Algoritmo Genético Generacional	7
4.2	Algoritmo Genético Estacionario	8
5	Algoritmo Memeticos	9
6	Tablas de resultados	10
6.1	Tabla de ejecuciones 1-NN	10
6.2	Tabla de ejecuciones Relief	11
6.3	Tabla de ejecuciones BL	11
6.4	Tabla de ejecuciones AGG-BLX	11
6.5	Tabla de ejecuciones AGG-CA	11
6.6	Tabla de ejecuciones AGE-BLX	12
6.7	Tabla de ejecuciones AGE-CA	12
6.8	Tabla de ejecuciones AM-[10 0.1]	12
6.9	Tabla de ejecuciones AM-[10 1]	12
6.10	Tabla de ejecuciones AM-[10 0.1mejor]	13
6.11	Resultados Globales	13
7	Análisis de resultados	13
7.1	Análisis Tasa de Clase	14
7.2	Análisis Tasa de Reducción	16
7.3	Análisis de Agregado	17
7.4	Análisis de tiempo	18
8	Experimento	19
8.1	Experimento 1: Mejor Sustituto	19
8.2	Experimento 2: Cruce Reducido	19

1 Descripción del problema

El Aprendizaje de Pesos en Características (APC) trata sobre la optimización de un clasificador de un conjunto de objetos a partir del uso de un vector de pesos que usaremos para ponderar las características dichos objetos. La estructura que tendremos para la optimización seria:

- Un vector de vectores de double de tamaño MxN donde introduciremos los datos de los objetos.
- Un vector de pesos de tamaño N, que usaremos para la clasificación de los objetos.

Tanto los datos de la matriz como el vector de pesos debe estar entre [0,1], por lo que debemos de Normalizar los datos de la matriz antes de introducirlo en nuestro algoritmo.

Para la normalización usaremos la siguiente formula:

$$x_j^N = \frac{x_j - Min_j}{Max_j - Min_j}$$

Para comprobar la calidad del vector de pesos obtenido por los algoritmos usaremos la siguiente función que debemos de maximizar:

$$F(W) = \alpha * TasaClas + (1 - \alpha) * TasaRed$$

Donde $\alpha = 0.5$ y la Tasa de Reducción sera el porcentaje de características que descartamos en el vector de pesos y la Tasa Clasificación de la función el porcentaje de aciertos de etiquetas sobre el conjunto de test.

$$tasa_class = \frac{n^o_instancias_bien_clasificados_en_T}{n^o_instancias_en_T}$$

$$tasa_red = \frac{n^o_valores_wi < 0.2}{n^o_caracteristicas}$$

Para determinar el conjunto de datos que usaremos como test dividiremos la matriz de datos en 5 secciones donde repartiremos equitativamente los datos y las etiquetas (5-fold cross validation). Realizaremos 5 ejecuciones del algoritmos usando en cada realización una partición como test. La función de para la partición la nuestro más adelante.

2 Descripción de funciones generales aplicadas al problema

Como ya hemos dicho antes, nuestro problema consiste en clasificar objetos dadas una serie de características las cuales usaremos para comprobar la distancia de ese objeto frente al resto. Estas distancias entre características son ponderadas el vector de pesos W, teniendo en cuenta que si el valor de $w_i < 0.2$ esta no es tomada en cuenta para el calculo de la distancia. Para determinar la distancia usaremos:

$$d(e_1, e_2) = \sqrt{\sum_{i=0}^N (e_{1i} - e_{2i})^2}$$

Respecto a la partición la realizamos como hemos descrito anteriormente. La función es esta:

```

1 void particionMatriz:
2
3     for i=0; i<5; i++
4         Introduce vector double en matrizParticion
5         Introduce vector double en targetParticion
6
7     for i=0; i<fil; i++
8         for j=0; j<col; j++
9             Introduce dato i, j en matrizParticion en la posicion [i%5]
10
11     Introduce dato i en targetParticion en la posicion [i%5]

```

También incluimos el pseudo-código de la normalización de los datos:

```

1 void normalizarDatos
2
3     MAX = 0
4     for i=0; i<fil; i++
5         for j=0; j<col; j++
6             Si dato en i,j es mayor que MAX
7                 MAX = dato i,j
8
9     MIN = 9999
10    for i=0; i<fil; i++
11        for j=0; j<col; j++
12            Si dato en i,j es menor que MIN
13                MIN = dato i,j
14
15    for i=0; i<fil; i++|
16        for j=0; j<col; j++
17            Aplicamos formula de normalización sobre el dato i,j

```

Para la lectura de los archivos realizaremos una modificación sobre estos, pasándolos a un archivo .txt y estableciendo en la primera linea el numero de filas que contiene el archivo y el número de columnas. La estructura seria:

```
1 287,62
2 0.344646644,0.003079788,0.
3 0.165329348,0,0.048235597,
4 0.457010481,0.001681183,0.
5 0.513243866,0.005711027,0.
6 0.390319468,0.009454114,0.
7 0.553354263,0.009471762,0.
8 0.543617703,0.00592724,0.3
9 0.510725932,0.010723651,0.
10 0.512959763,0.017649364,0.
11 0.565021103,0.01012961,0.3
12 0.293491288,0.002371782,0,
13 0.489216407,0.003477509,0.
14 0.203493295,0,0.051256344,
```

La función que usaremos para la lectura de los archivos sera:

```
1 void obtenerDatos
2
3     Abrimos el fichero
4
5     while No termine de leer el fichero
6         Lee linea del fichero
7
8         for j=0; j<col+1; j++
9             while No se termine la columna
10                 Si lo leído no es una coma ni un salto de linea lo sumamos a la palabra
11                 Si lee un salto de linea añade la palabra a la matriz
12                 Si no añade la palabra al vector de etiqueta
13
14     Cerramos el fichero
```

Para la función objetiva que debemos maximizar en este problema estara estructurado del siguiente modo:

```
1  double funcionEvaluacion
2
3      Calculamos la Tasa de Clase respecto a la particion Test
4      Calculamos la Tasa de Reduccion respecto al W obtenido
5      Calculamos la calidad = TasaClase*0.5 + TasaReduccion*0.5
6
7      Devolvemos calidad
```

Para el calculo de la tasa de Clase y Reducción usaremos:

```
1  double TasaClaseTest:
2
3      Recorremos la particion que hemos establecido como test
4      Recorremos las 5 particiones
5          Si la particion es diferente a la que estamos probando realizamos lo siguiente
6              Recorremos la particion seleccionada elemento por elemento
7                  Comparamos el elemento seleccionado en test con los demas elementos de otras particiones columna por columna
8                      Si dicha columna tiene una ponderacion mayor a 0.2
9                          Calculamos la distancia
10                         Si la distancia es la mas pequeña encontrada
11                             Establecemos como distancia minima
12
13      Si hemos acertado la etiqueta del objeto
14          Aumentamos el numero de aciertos
15
16      Devolvemos (100.0 * Numero de aciertos) / Tamaño de la particion de Test
```

```
1  double TasaReduccion:
2
3      for int i=0; i<numCaracteristicas; i++
4          Contamos cuantos valores son menores o iguales que 0.2
5
6      Devolvemos (100.0 * Numero de Valores menores o iguales que 0.2) / Numero de Caracteristicas total de W
```

3 Descripción funciones especificas para AG y AM

Para esta practica resolveremos el problema APC haciendo uso de Algoritmo Genéticos (Generacionales y Estacionarios) y Algoritmo Memeticos([10,0.1], [10,1], [10,0.1mejor]). Usando la población de cromosomas como vectores de pesos a los cuales se le asigna una calidad. La estructura de los cromosomas seria la siguiente:

```
1  struct Cromosoma
2      vector de pesos W
3      Calidad de W
```

En ambos casos necesitaremos una población inicial que generaremos de la siguiente forma:

```
1  vector<Cromosoma> InicializarPoblacion:
2
3      Inicializamos la poblacion con tam vectores de pesos
4      for i=0; i<tam; i++
5          Hacemos uso de la funcion Normal(0, 0.3) para los pesos del vector que estamos creando
6          Si algun peso es negativo establecemos ese peso como 0
7
8          Añadimos el vector de pesos a la poblacion junto con la calidad que genera
9      Devolvemos la poblacion creada
```

3.1 Operador de cruce

Para el cruce de los cromosomas de la población usaremos dos operadores de cruce:

Cruce BLX: Realizamos el cruce seleccionando un numero aleatorio dado un intervalo obtenido a partir de los pesos de los padres.

```
1  Cromosoma BLX:
2
3      Recorremos los pesos de los padres
4          Seleccionamos el peso minimo y maximo comparados de ambos padres en la columna correspondiente
5          Establecemos el valor del I = maximo - minimo
6          Generamos el numero aleatorio entre el intervalo creado entre [minimo-I*0.3, maximo-I*0.3]
7          Truncamos el numero si es menor que 0 o mayor que 1
8          Añadimos al el numero al vector de pesos del hijo
9
10     Devolvemos el cromosoma creado por el cruce de los padres
```

Cruce Aritmético: Se realiza la media aritmética de los pesos de los padres para obtener el peso del hijo.

```
1  Cromosoma CA:
2      Recorremos los pesos de los padres
3          Realizamos la media de los pesos en la posicion seleccionada
4          Introducimos el resultado en el vector de pesos del hijo
5
6      Devolvemos cromosoma creado por el cruce de los padres
```

3.2 Operador de selección

Para la selección de los componente de la población usaremos el llamado **Torneo Binario**, donde se selecciona dos cromosomas aleatorios de la población y se escoge al mejor de esos dos. Su pseudo-código seria:

```
1  Cromosoma TorneoBinario:
2      Seleccionamos el cromosoma 1 de forma aleatoria
3      Seleccionamos el cromosoma 2 teniendo en cuenta que no sea el mismo
4      Devolvemos el mejor de los dos cromosomas elegidos
```

3.3 Mutación

La mutación que realizaremos sobre los genes de nuestros cromosomas sera la de sumarle a este un vecino [numero obtenido a partir de $Mov(0,0.3)$]

4 Algoritmos Genéticos

Respecto a los algoritmos genéticos crearemos dos variantes: **Generacionales** y **Estacionarios**. La población sobre la que operaremos sera de una tamaño de 30.

4.1 Algoritmo Genético Generacional

En este algoritmo la probabilidad de que un cromosoma en la población sea sustituido por un nuevo cromosoma es de 0.7. Este nuevo cromosoma sera el resultado de realizar el **Torneo binario** dos veces y cruzando a los dos cromosomas obtenidos, ademas de tener una probabilidad de 0.001 por gen de que este sea mutado. Debemos tener en cuenta también que si el mejor de la población ha sido sustituido por un nuevo cromosoma, debemos de conservarlo en la siguiente población si este es mejor que el peor de la población.

Realizaremos este proceso un numero total de 15000 evaluaciones, considerando como evaluación una llamada a la Función Evaluación.

El pseudo-código sera de la siguiente forma:

```
1 void ModeloGeneracional:
2
3     Iniciamos la poblacion
4     Establecemos el mejor de la poblacion y el peor
5     Mientras no se realizen las 15000 evaluaciones
6         Guardamos el mejor cromosoma de la poblacion por si no sobrevive a la siguiente poblacion
7         Recorremos toda la poblacion
8             Si el cromosoma es elegido para sustituirlo
9                 Realizamos los torneo binario para seleccionar los padres
10                Aplicamos el cruce segun la opcion activada
11                    Dependiendo del cruce seleccionado se realizara BLX o CA
12                Recorremos los genes del cromosoma hijo
13                    Si el gen muta
14                        Se suma un vecino generado por Normal(0,0.3)
15                    Aumentamos el numero de evaluaciones
16                Introducimos el descendiente en la posicion i seleccionada
17                Aumentamos el numero de evaluaciones
18            Si se ha eliminado el mejor de la poblacion anterior
19                Sustituimos el peor cromosoma por el mejor de la poblacion anterior
20    Devolvemos el mejor de la poblacion
```

4.2 Algoritmo Genético Estacionario

Es este algoritmo la probabilidad de que un cromosoma sea sustituido es de 1, es decir en cada población todos sus cromosomas han sido creados a partir de los cromosomas de la población anterior. La selección de los padres del nuevo cromosoma sera por **Torneo Binario** y Mutación como el Generacional con la diferencia de que crearemos dos hijos y estos competirán con los dos peores de la población para entrar a esta, ambos hijos pueden tener padres diferentes.

El numero de iteraciones que realizaremos sera el mismo que en el Generacional.

El pseudo-código sera de la siguiente manera:

```
1 void ModeloEstacionario:
2
3     Iniciamos la poblacion
4     Mientras no se realizen las 15000 evaluaciones
5         Recorremos la poblacion
6             Realizamos el torneo binario para seleccionar los padres del hijo 1
7             Aplicamos el cruce segun la opcion activada
8                 Dependiendo del cruce seleccionado se realizara BLX o CA
9             Aumentamos el numero de evaluaciones
10            Volvemos a realizar el torneo binario para seleccionar los padres del hijo 2
11            Aplicamos el cruce segun la opcion activada
12                Dependiendo del cruce seleccionado se realizara BLX o CA
13            Aumentamos el numero de evaluaciones
14            Realizamos la mutacion de los genes de los nuevos descendientes
15                Si el gen de algun hijo muta
16                    Se suma un vecino generado por Normal(0,0.3)
17                    Aumentamos el numero de evaluaciones
18                Si los hijos creados son mejores que los peores de la poblacion
19                    Sustituimos el peor de la poblacion por el hijo
20    Devolvemos el mejor de la poblacion
```

5 Algoritmo Memeticos

En los algoritmos memeticos aplicaremos 3 variantes dependiendo de cuantos cromosomas de la población son elegidos para aplicarles la Búsqueda Local:

- AM (10, 0.1): Cada cromosoma tiene una probabilidad de 0.1 de modificarse con BL.
- AM (10, 1): Todos los cromosomas son modificados con BL.
- AM (10, 0.1 mejor): Se escoge el 0.1 de la población para modificarse con BL. Como la población es de tamaño 10 solo se cogerá 1 cromosoma que sera el mejor.

Como anteriormente he dicho, la población con la que ejecutaremos los memeticos sera de 10 cromosomas. Todos estas variantes son ejecutadas cada 10 generaciones de la población.

En el pseudo-código de los memeticos es similar al Genético Generacional aplicándole la Búsqueda Local del modo en el que queramos aplicarse a la población, para eso usaremos una variable *intmemetico* que indicara que tipo de variante de memetico aplicar:

```

1 void ModeloGeneracional:
2     Iniciamos la poblacion
3     Establecemos el mejor de la poblacion y el peor
4     Mientras no se realizen las 15000 evaluaciones
5         Si el modulo 10 de la generacion es 0
6             Si la variable memetico es 1
7                 Aplicamos BL a la poblacion completa
8             Si la variable memetico es 2
9                 Aplicamos BL a la poblacion con un 0.1 de posibilidad de ser escogido
10            Si la variable memetico es 3
11                Aplicamos BL al 0.1 mejor de la poblacion
12        Recorremos toda la poblacion
13            Si el cromosoma es elegido para sustituirlo
14                Realizamos los torneo binario para seleccionar los padres
15                Aplicamos el cruce segun la opcion activada
16                Dependiendo del cruce seleccionado se realizara BLX o CA
17                Recorremos los genes del cromosoma hijo
18                Si el gen muta
19                    Se suma un vecino generado por Normal(0,0.3)
20                    Aumentamos el numero de evaluaciones
21                Introducimos el descendiente en la posicion i seleccionada
22                Aumentamos el numero de evaluaciones
23            Si se ha eliminado el mejor de la poblacion anterior
24                Sustituimos el peor cromosoma por el mejor de la poblacion anterior
25        Devolvemos el mejor de la poblacion
26

```

La parte subrayada de azul pertenecería al pseudo-código del memetico.

6 Tablas de resultados

A continuación muestro las tablas de los tiempos y el porcentaje de los resultados obtenidos por los algoritmos de la practica 1 y 2:

6.1 Tabla de ejecuciones 1-NN

1-NN	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	79.00	0	39.50	0.028897	83.00	0	41.50	0.020327	92.00	0	46.00	0.054623
P2	68.00	0	34.00	0.028103	91.00	0	45.50	0.021663	88.00	0	44.00	0.054891
P3	68.00	0	34.00	0.023414	91.00	0	45.50	0.020544	96.00	0	48.00	0.054891
P4	63.00	0	31.50	0.024158	87.00	0	43.50	0.019427	92.00	0	46.00	0.055912
P5	75.00	0	37.50	0.023752	85.00	0	42.50	0.0202	93.00	0	46.50	0.05749
Media	70.60	0	35.30	0.0256648	87.40	0	43.70	0.0204322	92.20	0	46.10	0.056221

6.2 Tabla de ejecuciones Relief

Relief	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	72.93	58.06	66.96	0.13456	88.21	5.88	45.90	0.16114	94.09	17.5	55.57	0.41071
P2	72.93	90.32	78.78	0.09988	88.61	5.88	47.94	0.16099	94.32	10	49.54	0.43514
P3	72.61	59.68	63.17	0.13233	86.48	5.88	48.66	0.15196	94.54	10	53.18	0.43993
P4	73.91	54.84	59.88	0.13783	87.19	5.88	47.23	0.15225	93.41	10	51.81	0.44207
P5	71.74	70.97	68.82	0.1208	88.97	5.88	46.51	0.15685	92.50	10	51.81	0.43444
Media	72.82	66.77	67.52	0.12508	87.89	5.88	47.25	0.15664	93.77	11.5	46.10	0.43148

6.3 Tabla de ejecuciones BL

Relief	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	71.18	91.94	89.07	15.7572	87.86	85.29	88.42	9.41049	80.68	80.00	85.45	17.9984
P2	68.99	87.10	85.79	13.6276	88.97	82.35	89.75	4.99247	89.32	81.50	87.61	17.8096
P3	74.35	85.48	84.85	9.6582	85.05	82.35	89.75	5.74484	87.05	85.00	91.14	21.5876
P4	71.30	85.48	83.09	11.6582	85.77	88.24	90.55	6.12548	91.14	82.50	90.34	39.3256
P5	70.87	87.10	88.29	20.628	85.41	76.47	86.10	7.15834	85.91	85.00	89.32	26.4397
Media	71.34	87.42	86.22	14.2683	86.61	82.94	88.91	6.68632	86.82	83.00	88.77	24.6322

6.4 Tabla de ejecuciones AGG-BLX

AGG-BLX	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	68.12	93.55	84.71	66.8842	85.00	94.12	92.13	59.5769	81.82	90.00	87.73	197.929
P2	66.81	93.55	86.43	66.8186	80.43	91.18	92.73	66.19	83.41	87.50	88.30	185.83
P3	73.91	91.94	86.32	71.3577	78.29	91.18	91.30	64.5754	83.41	87.50	89.66	173.499
P4	63.48	95.16	87.05	67.9489	88.97	91.18	89.87	66.2464	80.23	90.00	89.09	172.207
P5	67.39	93.55	88.00	66.7256	77.22	94.12	90.63	59.0184	89.77	87.50	89.66	202.286
Media	67.94	93.55	86.50	67.947	81.98	92.35	91.33	63.1214	83.73	88.50	88.89	180.3502

6.5 Tabla de ejecuciones AGG-CA

AGG-CA	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	71.62	40.32	58.95	250.921	90.00	50.00	70.07	193.279	92.73	42.50	68.52	530.898
P2	70.74	24.19	50.89	294.142	85.77	38.24	64.83	215.803	94.32	37.50	65.11	569.547
P3	71.30	27.42	53.18	276.142	85.77	35.29	64.08	219.209	92.05	42.50	69.43	515.018
P4	73.48	35.48	54.58	260.826	87.90	44.12	67.77	197.211	91.82	27.50	60.57	658.323
P5	70.87	27.42	54.06	285.098	90.04	38.24	64.12	204.139	92.73	40.00	67.73	585.87
Media	71.60	30.97	54.33	273.5856	87.89	41.18	66.17	205.9282	92.73	38.00	66.27	572.1312

6.6 Tabla de ejecuciones AGE-BLX

AGG-CA	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	68.56	93.55	87.29	81.6576	89.64	94.12	92.13	59.4347	87.95	87.50	91.02	221.889
P2	72.49	87.09	86.65	84.2305	86.83	85.29	89.79	79.97	92.27	82.50	89.43	231.326
P3	71.30	88.71	86.46	80.4849	86.12	91.18	92.73	68.2791	91.59	85.00	91.14	216.044
P4	75.65	81.09	83.90	86.98	85.41	88.24	91.97	75.1889	89.09	82.50	88.52	227.257
P5	69.57	83.87	84.04	96.0751	87.19	91.18	92.73	72.6281	91.59	87.50	91.02	201.524
Media	71.51	88.06	85.67	85.8856	87.04	90.00	91.87	70.9002	90.50	85.00	90.23	219.608

6.7 Tabla de ejecuciones AGE-CA

AGE-CA	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	72.93	54.84	65.35	176.2	88.21	61.76	75.25	168.815	91.14	60.00	75.45	459.069
P2	68.12	58.06	70.41	181.47	87.54	58.82	73.69	169.822	92.95	70.00	82.73	399.239
P3	73.04	64.52	69.98	175.92	86.48	61.76	77.31	165.813	89.77	60.00	76.82	413.2
P4	70.43	54.84	65.14	199.929	86.12	58.82	76.55	171.288	93.64	67.50	79.66	373.21
P5	70.87	59.68	70.19	192.8	89.32	58.82	72.98	177.002	92.50	67.50	80.57	412.8
Media	71.08	59.68	68.21	185.2638	87.53	59.99	75.16	170.548	92.00	65.00	79.05	411.5036

6.8 Tabla de ejecuciones AM-[10 0.1]

AM-[10 0.1]	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	71.18	88.71	86.60	101.92	86.07	88.24	94.12	75.3299	90.45	87.50	90.57	215.104
P2	69.87	83.87	84.18	91.8907	81.85	91.18	91.30	64.1327	89.77	87.50	91.93	207.787
P3	70.43	90.32	88.14	82.9939	86.83	91.18	93.45	70.2806	89.77	85.00	91.59	231.058
P4	73.04	87.10	84.78	82.2562	84.69	88.24	91.26	75.4273	88.86	87.50	91.02	209.825
P5	69.13	83.87	84.92	87.8869	86.12	91.18	91.30	66.9088	92.50	82.50	87.61	238.713
Media	70.73	86.77	85.72	89.38954	85.11	90.00	92.29	70.41586	90.27	86.00	90.55	220.4974

6.9 Tabla de ejecuciones AM-[10 1]

AM-[10 1]	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	71.61	83.87	84.18	90.3035	80.00	91.18	91.36	67.7682	92.03	80.00	88.64	248.486
P2	69.43	85.48	84.98	86.7142	87.19	95.71	93.45	68.6002	88.18	87.50	91.48	205.197
P3	69.13	87.10	84.78	88.186	77.22	94.12	92.06	60.6434	90.00	85.00	90.68	218.94
P4	72.61	85.48	82.21	89.1309	83.27	82.35	89.03	88.1315	87.05	85.00	88.86	218.632
P5	70.43	88.71	88.21	80.7379	83.27	88.24	92.69	71.9524	87.95	85.00	89.77	224.776
Media	71.64	86.13	84.87	87.0145	82.19	90.32	91.72	71.4191	89.09	84.50	89.89	223.2062

6.10 Tabla de ejecuciones AM-[10 0.1mejor]

AM-[10 0.1mejor]	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	73.80	87.10	84.93	107.294	86.43	85.29	91.94	85.4683	86.59	87.50	90.02	230.887
P2	68.12	88.71	84.01	103.56	88.26	91.18	94.87	71.5852	90.91	85.00	89.77	251.235
P3	71.30	91.94	89.83	92.73	88.97	85.29	91.93	83.3041	90.00	85.00	89.77	253.921
P4	73.04	91.94	87.20	96.4717	88.61	88.24	92.69	78.6301	88.86	87.50	91.47	239.111
P5	72.17	88.71	87.34	106.989	84.70	91.18	92.73	73.304	92.95	82.50	91.93	228.316
Media	71.69	89.68	86.66	101.40888	87.39	88.24	92.83	78.45834	89.86	86.50	90.80	240.694

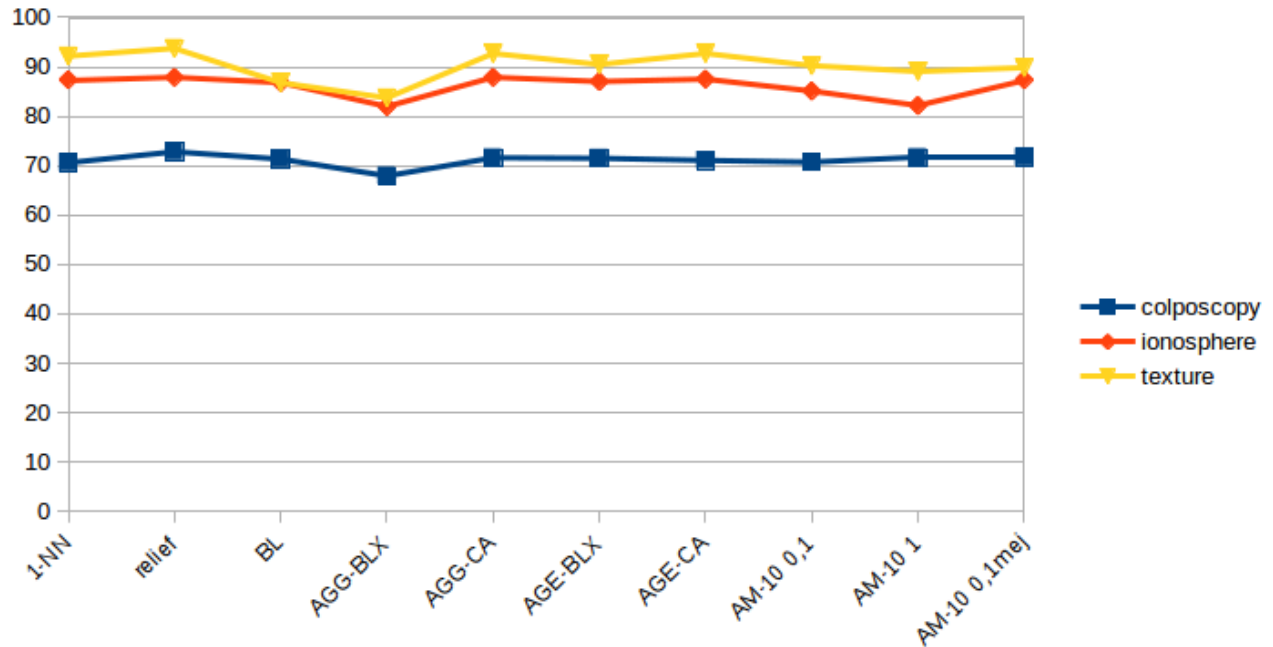
6.11 Resultados Globales

Media Global	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
1-NN	70.60	0	35.30	0.0256648	87.40	0	43.70	0.0204322	92.20	0	46.10	0.056221
Relief	72.82	66.77	67.52	0.12508	87.89	5.88	47.25	0.15664	93.77	11.5	46.10	0.43148
BL	71.34	87.42	86.22	14.2683	86.82	83.00	88.77	24.6322	86.82	83.00	88.77	24.6322
AGG-BLX	67.94	93.55	86.50	67.947	81.98	92.35	91.33	63.1214	83.73	88.50	88.89	180.3502
AGG-CA	71.60	30.97	54.33	273.5856	87.89	41.18	66.17	205.9282	92.73	38.00	66.27	572.1312
AGE-BLX	71.51	88.06	85.67	85.8856	87.04	90.00	91.87	70.9002	90.50	85.00	90.23	219.608
AGE-CA	71.08	59.68	68.21	185.2638	87.53	59.99	75.16	170.548	92.73	38.00	66.27	572.1312
AM - [10 0.1]	70.73	86.77	85.72	89.38954	85.11	90.00	92.29	70.41586	90.27	86.00	90.55	220.4974
AM - [10 1]	71.64	86.13	84.87	87.0145	82.19	90.32	91.72	71.4191	89.09	84.50	89.89	223.2062
AM - [10 1mejor]	71.69	89.68	86.66	101.40888	87.39	88.24	92.83	78.45834	89.86	86.50	90.80	240.694

7 Análisis de resultados

A continuación realizaremos un análisis de las tablas anteriores. Iremos columna a columna comparando los resultados obtenidos de todos los algoritmos conjuntos.

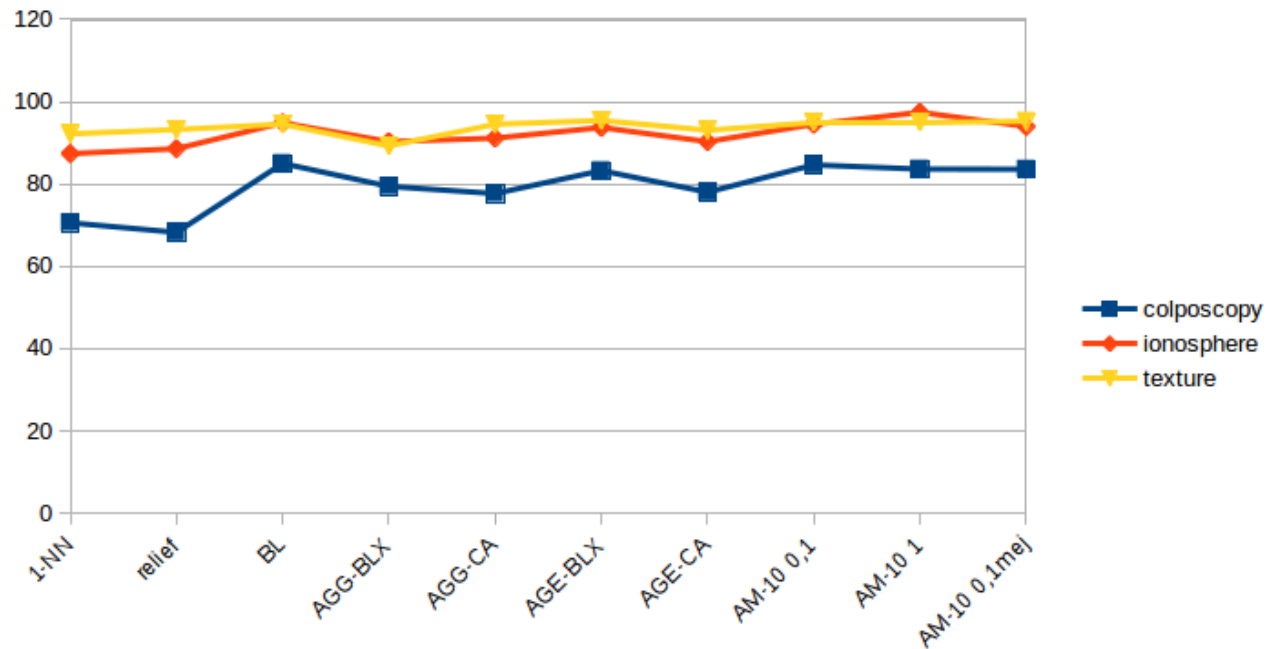
7.1 Análisis Tasa de Clase



La tasa de clase aquí mostrada es referida al numero de aciertos obtenidos respecto al conjunto de datos de entrenamiento. Observamos que el conjunto de datos que mejor resultado a dado en adaptar W en todos los algoritmos probados ha sido Texture, esto podría justificarse debido a la cantidad de ejemplos que tenemos. Respecto a los algoritmos utilizados, el mejor resultado lo ha dado Relief seguido de AGG-CA de forma muy proxima.

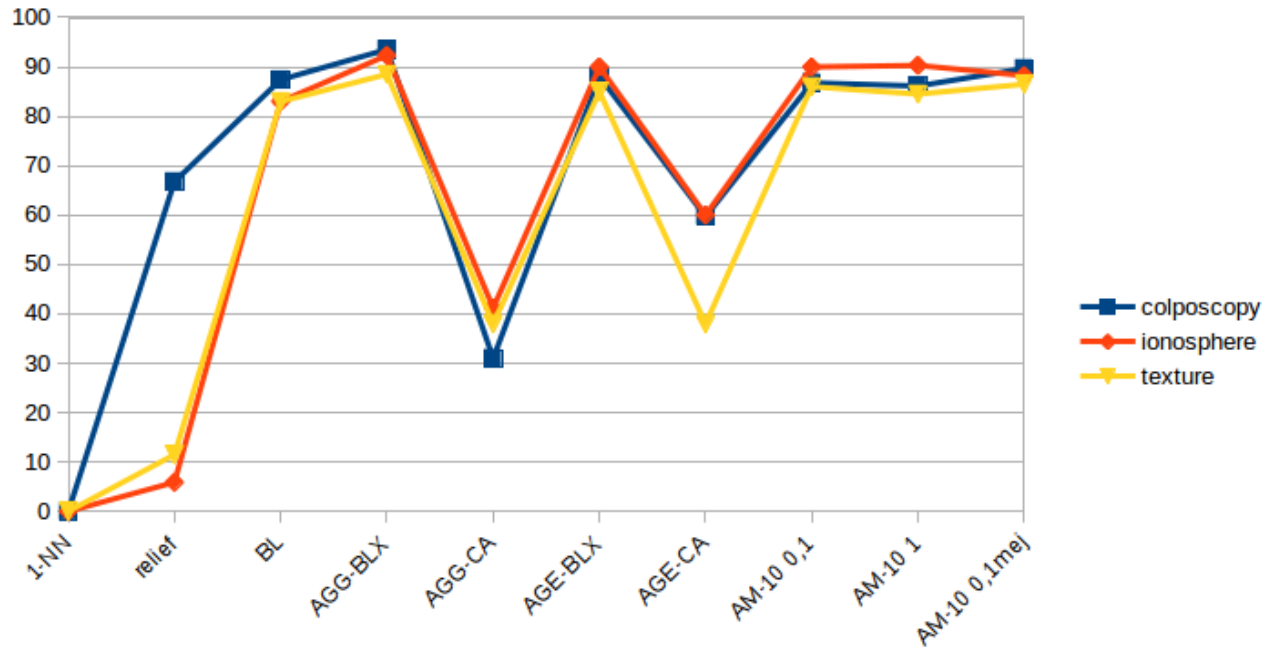
Respecto a los algoritmos añadidos en esta practica vemos que respecto a Tasa de clase AGG-CA y AGE-CA han superado a AGG-BLX y AGG-CA en Ionosphere y Texture, superando con una diferencia de solo 0.5 en Colposcopy. Esto podría producirse ya que BLX provoca mas mejor en la tasa de reducción lo cual es descompensado en la tasa de clase, cosa que en CA no sucede.

También mostrar la tasa de clasificación respecto al numero de acierto generado en la muestra de test.



Vemos que sobre la muestra de Test el mejor resultado es dado por el conjunto de datos de Texture, siendo los memeticos los que mejor resultado han dado en general.

7.2 Análisis Tasa de Reducción

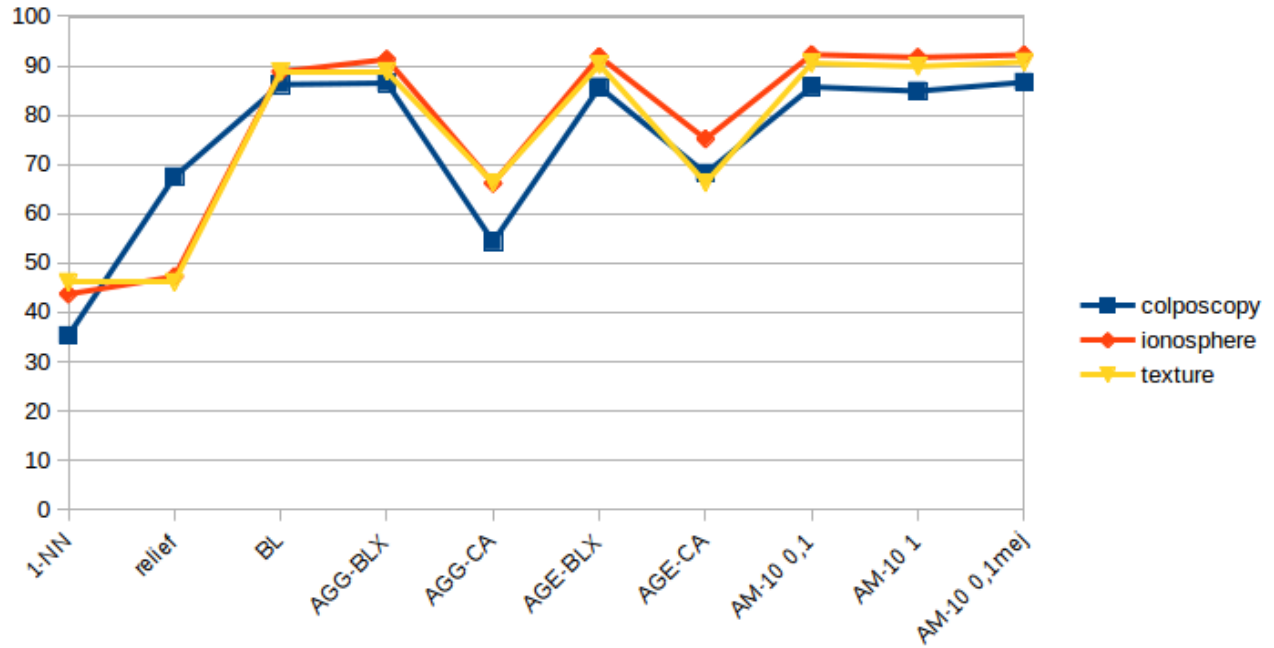


Observando la gráfica podemos ver que las tasas de reducción son muy similares en la mayoría de algoritmos exceptuando grandes cambios como el algoritmo relief que parece dar una gran aumento en la reducción, no siendo ese el objetivo de este algoritmo. La mayor reducción es dada por el algoritmo genetico generacional con cruce BLX y la peor de entre los añadidos en esta practica ha sido el genetico generacional con cruce CA.

Dada una visión general de la reducción de los nuevos algoritmos vemos que se producen peores reducciones con CA, esto es debido a que tal y como se realiza el cruce de ambos genes es difícil obtener que ese carácter se reduzca, ya que debe de producirse un cruce entre dos padres donde uno de ellos este reducido y el otro no supere el 0.4. Aun así la media de ambos padre podría dar mayor a 0.2 y no reducirse.

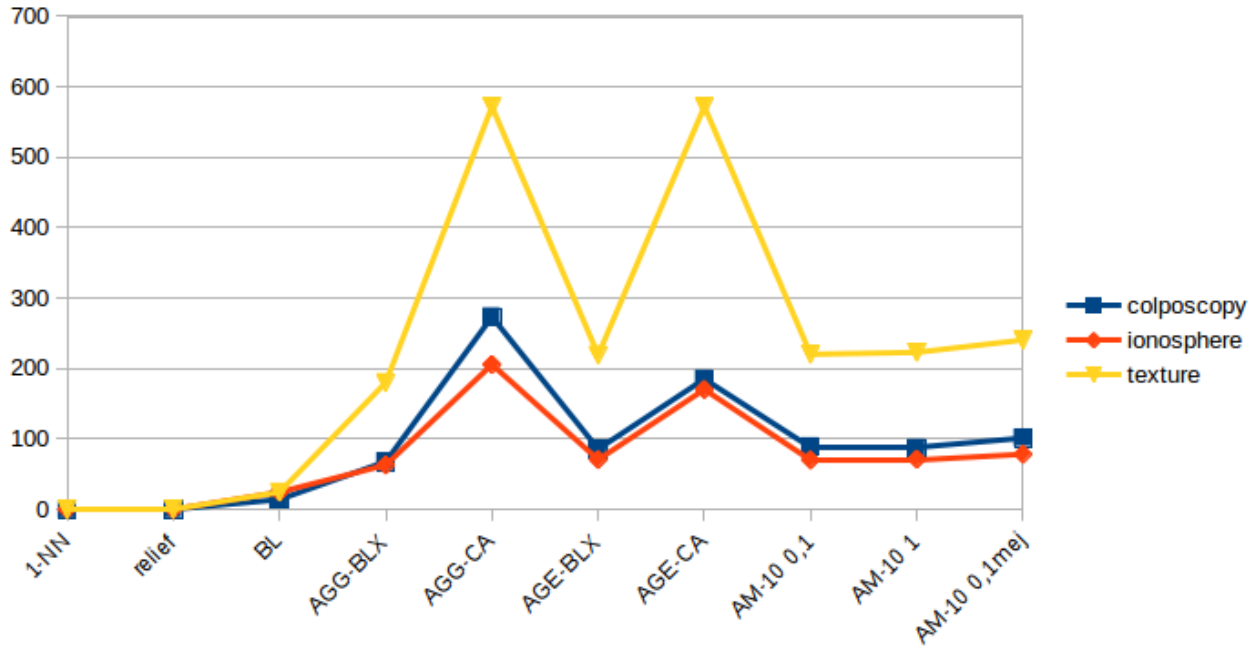
También comentar que todos los memeticos han dado buenos resultado en todos los conjuntos de datos igualando o superando al algoritmos de BL.

7.3 Análisis de Agregado



Analizando la gráficas de agregados observamos que los mejores resultados son dados por los meméticos, siendo AM-[10 1mejor] el que mejor resultado otorga. Esto es debido que estos algoritmos unen la buena exploración de soluciones globales de los genéticos con la rapidez de obtener el optimo de BL. Podemos destacar junto a estos el AGG-BLX dado muy buenos resultados en comparación a AGG-CA y AGE-CA que debido a la falta de reducción, la calidad del algoritmo se reduce mucho.

7.4 Análisis de tiempo



Observando la gráfica puede verse que los algoritmos con cruce CA claramente requieren de mas tiempo en ejecución que los demás algoritmos, sobre todo en texture. Este aumento de tiempo puede ser provocado por la división que se realiza en este cruce.

Respecto al mejor en tiempo, observamos que BLX da resultados muy buenos al igual que los algoritmos memeticos. Debido a los resultados de calidad, yo seria mas propenso a usar algoritmos memeticos para la obtención de la solución de nuestro problema APC.

8 Experimento

A continuación realizare dos experimentos sobre los algoritmos dados en esta practica.

8.1 Experimento 1: Mejor Sustituto

Haciendo uso del algoritmo AGG-BLX modificaremos este para que a la hora de seleccionar un individuo de la población, si el hijo creado es peor que el, buscaremos otro elemento cuya calidad sea peor que este. Si no existe no rechazara dicho hijo.

Los resultados son estos:

AGGMS-BLX	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	67,25	93,55	88,15	83,8742	90	88,24	92,00	83,3798	86,36	82,5	89,43	320,241
P2	65,07	90,32	87,40	101,044	86,12	88,24	91,26	90,5609	85,68	87,5	90,56	248,068
P3	70,87	88,71	86,46	100,312	85,41	91,18	93,45	82,6896	86,81	90	90,91	213,636
P4	68,69	93,55	87,12	89,9663	86,83	88,24	91,97	90,7522	88,18	87,5	91,48	278,566
P5	70,86	88,71	86,46	114,074	88,61	91,17	92,73	82,0059	85,45	87,5	89,66	274,854
Media	68,55	90,97	87,12	97,8541	87,40	89,41	92,28	85,8776	86,49	87	90,41	267,073

Vemos que los resultado son algo mejores que el dado en el AGG-BLX, pero si nos fijamos en el aumento de tiempo tal vez no sea beneficioso modificar de esta forma el algoritmo, ya que el aumento de tiempo es bastante significativo comparado con la mejora de calidad, sobre todo en el conjunto de Texture.

8.2 Experimento 2: Cruce Reducido

Para este experimento hemos realizado un nuevo operador de cruce llamado Cruce Reducido, cuyo pseudo-código es el siguiente:

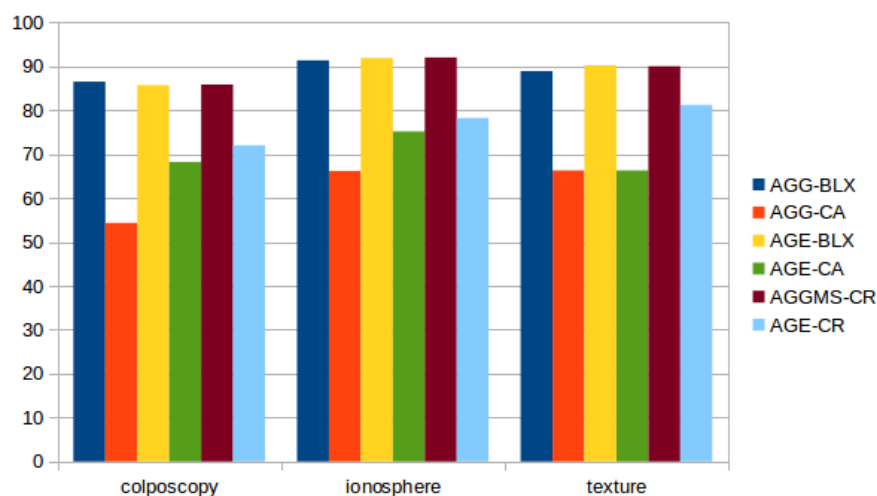
```
1  Cromosoma CruceReducido:
2
3      Declaramos dos hijos, hijo1 e hijo2
4      Recorremos los pesos de los padres
5          Si el peso del padre 1 en la posicion i es menor o igual a 0.2
6              Añadimos dicho peso al hijo1
7          Si no
8              Añadimos el peso del padre 2 en la poicion i en el hijo1
9
10         Si el peso del padre 2 en la posicion i es menor o igual a 0.2
11             Añadimos dicho peso al hijo2
12         Si no
13             Añadimos el peso del padre 1 en la poicion i en el hijo2
14     Evaluamos ambos hijos
15     Devolvemos el hijo con mejor calidad obtenido
```

La intención de este cruce es obtener hijos que favorezcan la reducción, conservando la reducción dada por alguno de sus padres. Los resultados tanto en AGG como AGE han sido estos:

AGGMS-CR	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	70,74	83,87	84,17	204,134	86,42	88,23	92	152,391	91,81	82,5	87,61	502,36
P2	70,30	85,489	84,98	217,888	88,61	88,23	90,54	153,764	87,95	87,5	91,47	469,386
P3	66,95	88,70	85,58	185,197	89,32	88,23	92,68	150,322	85	90	92,27	409,711
P4	68,69	93,54	87,12	164,806	85,76	91,17	92,01	155,985	85,90	85	88,40	423,778
P5	71,30	88,70	87,33	187,148	89,67	91,17	92,73	140,375	89,54	85	90,22	415,929
Media	69,60	88,06	85,84	191,8346	87,96	89,41	91,99	150,5674	88,04	86	90	444,2328

AGE-CR	Colposcopy				Ionosphere				Texture			
	%clas	%red	Agr.	T	%clas	%red	Agr.	T	%clas	%red	Agr.	T
P1	70,30	83,87	84,17	95,521	87,5	85,29	90,53	78,0506	92,7273	82,5	86,25	258,086
P2	68,12	58,06	70,41	234,138	87,54	58,82	73,69	184,697	92,9545	70	82,7273	438,766
P3	73,04	64,51	69,97	206,094	86,47	61,76	77,31	178,512	89,7727	60	76,8182	481,113
P4	70,43	54,83	65,13	219,449	86,12	58,82	76,55	182,265	93,6364	67,5	79,6591	444,99
P5	70,86	59,67	70,18	213,583	89,32	58,82	72,98	186,521	92,5	67,5	80,5682	454,652
Media	70,55	64,19	71,97	193,757	87,39	64,70	78,21	162,0091	92,31818	69,5	81,20456	415,5214

Comparando las calidades respecto a los demás algoritmos, obtenemos:



Observando esta Gráfica de Barras que representa el Agregado de los algoritmos, puede apreciarse que el CR es mas efectivo en AGG que en AGE. Pero respecto a otros algoritmos, CR es comparable a BLX en AGG y peor en AGE. Mientras que la diferencia es notable respecto a CA en todos los casos.

Notar que he hecho uso del experimento anterior sobre AGG ya que este daba mejores resultados.

Respecto a la reducción de estos algoritmos:

