Un Algoritmo Heurístico para el problema de fusión de mesas en un proceso electoral

José Pintado Huamán[1], Rosa Delgadillo[2], David Mauricio[3]

[1]Oficina Nacional de Procesos Electorales
Jr. Washington 1894, Lima-Perú.

[2, 3]Universidad Nacional Mayor de San Marcos,
Facultad de Ingeniería de Sistemas e Informática
Av. Germán Amézaga s/n, Ciudad Universitaria, Lima 01, Lima, Perú

[1]JPintado@onpe.gob.pe, [2]rsdelgadillom@yahoo.com.mx, [3]dms_research@yahoo.com

RESUMEN

El proceso electoral en el Perú presenta entre sus actividades críticas el problema de determinar el menor número de mesas de sufragio y la conformación de estas con mejor uniformidad. Este problema es llamado de fusión de mesas de sufragio, y puede ser visto como una combinación de los problemas de empaquetamiento y de programación de tareas. Presentamos una solución para este problema que consta de un algoritmo BFD para determinar el menor número de mesas de sufragio, y dos procedimientos para mejorar la uniformidad de la distribución de electores por mesas. Los experimentos numéricos sobre 168 grupos electorales con un total de 15,534 electores muestran que la solución propuesta reduce en 14,46 % el número de mesas electorales y mejora en 3.97% la uniformidad de la distribución de electores por mesa.

Palabras clave: Heurística, Corte y Empaquetamiento, Programación de Tareas.

ABSTRACT

The electoral process in Peru presents among your critical activities the problem of determining the minor number of tables of suffrage and the conformation of these with better uniformity. This problem is called of merger of tables of suffrage, and can be seen as a combination problem of the packing problem and programming task problem. We present a solution for this problem that consists of an algorithm BFD to determine the minor number of tables of suffrage, and two procedures to improve the uniformity of the electors' distribution for tables. The numerical experiments on 168 electoral groups with a whole of 15,534 electors show that the proposed solution reduces in 14,46 % the number of table of suffrage and improvement in 3.97 % the uniformity of the electors' distribution for table.

Keywords: Heuristic, Cutting and Packing, Programming Task.

1. INTRODUCCIÓN

La Oficina Nacional de Procesos Electorales del Perú (ONPE, 2007) tiene como misión desarrollar con eficiencia y eficacia los procesos electorales para la elección de presidente de la republica, congresistas, presidente regional, y alcaldes en la republica de Perú. En cada proceso electoral se llevan a cabo varias actividades críticas; una de estas actividades es la fusión de mesas electorales por local de votación, la cual tiene como finalidad definir el número y conformación de mesas de sufragio a instalar en un proceso electoral; éste se realiza teniendo en cuenta los parámetros de cantidad de electores y cantidad de grupos de votación por mesa. La cantidad máxima de electores por mesa es de 250 electores y el número máximo de grupos de votación por mesa es 9. Un grupo de votación tiene un número determinado de electores y no puede ser dividido. Una mesa de sufragio se compone de un número determinado de grupos de votación.

El problema de fusión de mesas consiste en determinar una fusión de mesas electorales que presente menor número de mesas y mejor uniformidad de distribución de electores por mesas.

La eficacia de la solución del problema de fusión de mesas impacta de manera directa en los factores de tiempo y costo de un proceso electoral. Actualmente, esta actividad se lleva a cabo utilizando un procedimiento empírico que consta de dos fases consecutivas. En la primera fase, se determina el número de mesas de sufragio agrupando la mayor cantidad permitida de electores por mesa. En la segunda fase, en forma manual, se modifica la conformación de los grupos de votación por mesa buscando reducir el número de mesas y una distribución uniforme de electores por mesas de sufragio.

El problema de fusión de mesas que presenta la ONPE puede ser definido como dos problemas relacionados: 1) determinar un número mínimo de mesas de sufragio dado diferentes tamaños de los grupos de votación, y 2) determinar la mejor conformación de las mesas de sufragio (en el sentido de mejor distribución del número electores por mesas) dado los grupos de votación.

El primer problema puede ser visto como el problema de empaquetamiento cuyo objetivo es determinar el menor número de mesas (cajas) a ser utilizadas para cubrir (empaquetar) los diferentes grupos de votación (el tamaño de un grupo de votación siempre es menor o igual al número máximo de electores de una mesa de sufragio). El segundo problema puede ser visto como un problema de programación de tareas independientes en máquinas idénticas, cuyo objetivo es distribuir los grupos de votación (tareas) en las mesas de sufragio (máquinas) de forma tal que se minimice el número de electores (tiempo) en las mesas de sufragio.

El problema de empaquetamiento es uno de los problemas combinatorios más relevantes en la literatura dada la diversidad de sus aplicaciones Campello (1992), Mauricio (2002), Delgadillo (2007). Se define como, dado un conjunto limitado de cajas de tamaño fijo y un conjunto de piezas (cajas de menor tamaño) de tamaño se busca encontrar una asignación (empaquetado) de todas las piezas utilizando el menor número posible de cajas, de forma que ninguna caja sea llenada sobrepasando su capacidad así como ninguna pieza sea fraccionada en piezas menores.

El problema de programación de tareas (task scheduling) presenta sus antecedentes en la planificación industrial y en la programación de trabajos de procesadores en los inicios de la microelectrónica. Este problema puede definirse Campello (1992) como: dado un conjunto de máquinas (o procesadores) y un conjunto de tareas con tiempo de ejecución de las tareas iguales o diferentes en las máquinas; se desea programar las tareas en las máquinas, buscando que el mayor tiempo de procesamiento de todas las máquinas (makespan) sea el menor posible.

El problema de programación de tareas presenta una serie de variantes dependiendo de la naturaleza y el comportamiento tanto de las tareas como de las máquinas, una de estas variantes es el de la programación de tareas independientes en máquinas idénticas. Dicho problema supone: la independencia de las tareas, esto es, no existe el requisito que para ejecutar una tarea sea necesario previamente ejecutar alguna tarea anterior; que las máquinas sean idénticas, esto es, el tiempo de procesamiento de una tarea es igual en todas las máquinas.

El presente estudio propone una solución al problema de determinación y conformación de mesas de sufragio haciendo uso de un algoritmo BFD y dos procedimientos de mejorías. El trabajo esta organizado en 6 secciones. En la sección 2, se presenta un esquema de solución al problema. El algoritmo BFD se presenta en la sección 3. Los procedimientos de mejoría se describen en la sección 4. En la sección 5, se presentan los resultados numéricos. Finalmente, las conclusiones siguen en la sección 6.

2. ESQUEMA DE SOLUCIÓN

Considerando que el problema de fusión de mesas de la ONPE es un problema combinado de los problemas de empaquetamiento y programación de tareas independientes – máquinas idénticas, se ha considerado el bien sucedido algoritmo goloso BFD (Best Fit Decreasing) para determinar el menor número de mesas de sufragios por local de votación (problema de empaquetamiento). La solución generada por el algoritmo BFD también muestra una conformación de mesas electorales por local de votación (esto es, una planificación de tareas), en ese sentido se ha usado los procedimientos de mejoría del algoritmo de programación de tareas propuesto por Tupia (2001, 2004) para encontrar una mejor distribución de electores por mesas de sufragio. En la siguiente figura se muestra el esquema de solución para el problema en estudio.

Algoritmo BFD+2M
Inicio
1. Leer $(n, G_1, G_2, ..., G_n, \overline{C}, \overline{g})$;
2. Inicializar $S = \{\ \}$,
3. Ejecutar Algoritmo BFD;
4. Ejecutar Procedimiento de mejoría 1;
5. Ejecutar Procedimiento de mejoría 2;
6. Retornar S
Fin

Figura 1: Un Algoritmo BFD + 2M para resolver el problema de fusión de mesas

El algoritmo BFD+2M ejecuta en orden secuencial el algoritmo BFD y los dos procedimientos de mejoría. Al término del algoritmo BFD se tiene el número de mesas de sufragio por local de votación y una conformación de las mesas de sufragio. El número de mesas dado por el algoritmo BFD es considerado como un parámetro fijo para los dos procedimientos de mejoría. En los dos procedimientos se busca mejorar la conformación de las mesas de sufragio dado por el algoritmo BFD. El primer procedimiento consigue otra mejor solución respecto a la distribución de los electores en las mesas de sufragio pasando grupos de votación de una mesa con mayor número de electores para otra mesa con menor número de electores; en el segundo procedimiento la solución obtenida por el procedimiento de mejoría 1 es mejorada aun más si es posible intercambiarlo en grupos de votación entre dos mesas de mayor y menor número de electores. Al término del segundo procedimiento se obtienen las mesas de sufragio y la conformación de ellas.

3. ALGORITMO BFD

3.1. Notación

En lo que sigue del presente trabajo se usará la siguiente notación:

n : número de grupos de votación
m : número de mesas de sufragio
G_j : grupo de votación "j" ($j = 1, ..., n$)
S_i : mesa de sufragio "i" ($i = 1, ..., m$)
$\overline{C}$: Capacidad de las mesas (esto es, el máximo de número de electores por mesa)
C_i : Capacidad disponible de S_i
$\overline{g}$: número máximo de grupos de votación por mesa
g_i : número de grupos de votación de S_i

3.2. El proceso de asignación en mesa nueva y usada

Asignar en Mesa Nueva

Considere un grupo de votación G_j que se desea asignar en una mesa aun no abierta S_i. El procedimiento de asignar un grupo en una mesa nueva, que denominamos Asignar_Mesa_Nueva consiste en dos pasos. Primero se asigna el grupo G_j en la mesa S_i, esto es $S_i = G_j$. Y segundo, se actualiza la capacidades de número de electores y de grupos de la mesa S_i de la siguiente forma respectivamente $C_i = \overline{C} - |G_j|$ y $g_i = \overline{g} - 1$.

Asignar en Mesa Usada

Considere un grupo de votación G_j que se desea asignar en una mesa usada S_i, y que esto es posible (esto es, se verifica las condiciones de capacidades de número de electores y de grupos de votación en S_i). El procedimiento de asignar un grupo en una mesa usada, que denominamos Asignar_Mesa_Usada consiste en dos paso. Primero, se asigna el grupo G_j en la mesa S_i, esto es $S_i = S_i \cup G_j$. Y segundo, se actualiza la capacidades de número de electores y de grupos de la mesa S_i de la siguiente forma respectivamente $C_i = C_i - |G_j|$ y $g_i = g_i - 1$.

3.3. Algoritmo

Para el presente algoritmo BFD, la orden de prioridad para procesar los grupos de votación es dada por la ordenación no creciente respecto a sus tamaños (número de electores). En cada iteración del algoritmo, se selecciona la mesa que presenta menor capacidad disponible en donde se pueda asignar el grupo en proceso (siempre se considera una mesa vacía adicional). Si la mesa es nueva (esto es, está vacía) se aplicará el proceso de asignación_mesa_nueva, de lo contrario, se aplicará el proceso asignación_mesa_usada. Este proceso se repetirá hasta procesar todos los grupos.

En el paso 4.1 se determina la mesa "i" que presenta menor capacidad disponible donde se puede encajar el grupo G_j y que cumple la condición de no sobrepasar al máximo número de grupos por mesa. Observe que, si i = m+1 entonces la mesa seleccionada es nueva (paso 4.3), en el caso contrario la mesa seleccionada será una usada (paso 4.4). m indica el número de mesas usados.

Algoritmo BFD

1. Leer $(n, G_1, G_2, \ldots, G_n, \overline{C}, \overline{g})$;
2. Ordenar (G_j) tal que: $|G_1| \geq |G_2| \geq \ldots \geq |G_n|$
3. $m := 0$;
4. Para $j := 1, \ldots, n$

4.1 $i := \underset{1 \leq k \leq m+1}{ArgMin}\{C_k : C_k \geq |G_j|,\ g_k < \overline{g}\}$

4.2 Si $i = m+1$

4.3 Entonces *Asignar_Mesa_Nueva* (G_j, S_i), $m := m+1$,

4.4 Em caso contrario *Asignar_Mesa_Usada* (G_j, S_i),

5. Retornar $(m, S_i\ \forall i)$

Figura 2: Algoritmo BFD para determinar una fusión de menor número de mesas

4. PROCEDIMIENTO DE MEJORÍA

La segunda parte del esquema de solución propuesto para el problema de fusión de mesas electorales es un procedimiento de búsqueda local (Local Search), el cual tiene por objetivo mejorar la solución encontrada por el algoritmo BFD, esto es, hallar a partir de la solución encontrada una solución que presente una mejor conformación de los grupos de votación en las mesas de sufragio, en el sentido que minimice la desviación del tamaño de dichas mesas. Existen diversas técnicas para construir una mejor solución a partir de una solución dada, para una revisión de estas técnicas vea por ejemplo Glover y Kochemberger (2003).

A continuación, se introducen dos procedimientos de mejoría para mejorar la solución generada por el algoritmo BFD, que denotamos por $S = \{S_1, S_2, \ldots, S_m\}$.

Primer Procedimiento de Mejoría

Este procedimiento consiste de dos pasos. Primero, se determina la mesa de menor número de electores S_k y la mesa de mayor número de electores S_l. Segundo, se transfiere todo grupo de votación G_i de la mesa S_l a la mesa S_k siempre que $|S_k| + |G_{i \in S_l}| < |S_l| - |G_{i \in S_l}|$; si no es posible transferir ningún grupo de votación de la mesa S_l a la mesa S_k se considerará la siguiente mesa con mayor número de electores en vez de S_l. Los pasos 1 y 2 se repiten hasta que no sea posible hacer transferencia alguna de grupos de votación.

Segundo Procedimiento

Consiste de 3 pasos. Primero, se determina la mesa de menor número de electores S_k y la mesa de mayor número de electores S_l. Segundo, se generan combinaciones de a los más dos grupos de votación en cada una de las mesas seleccionadas. Tercero, se intercambia un grupo (o una combinación de dos grupos) de la mesa S_l por un grupo (o una combinación de dos grupos) de la mesa S_k siempre que $|G_{i \in S_l}| - |G_{i \in S_k}| < |S_l| - |S_k|$. Si no es posible intercambiar grupos de votación entre las mesas seleccionadas, se repite el proceso con la siguiente mesa mayor en vez de S_l. Este procedimiento se repite hasta que no sea posible hacer intercambios, esto es, hasta que no se pueda mejorar la dispersión de los tamaños de las mesas.

5. EXPERIMENTOS NUMÉRICOS

Los experimentos numéricos se han realizado en un microprocesador Intel de 2.0 Ghz. con memoria RAM 1 GB., sistema de disco Maxtor de 80 GB y sistema de archivo de 32 bits. El sistema operativo utilizado fue Windows 2000 Server SP4, con base de datos Oracle 9i, para el reporte office 2000.

Las pruebas numéricas del algoritmo propuesto se realizaron para un grupo de diez instancias tomadas del proceso electoral de Elecciones Regionales y Municipales 2006 ONPE (2006), donde cada instancia está conformada por un local de votación de los diferentes departamentos del país, y donde cada local de votación está conformado por una cantidad variable de grupos de votación. Las cantidades máximas para cada mesa fusionada fueron de 250 electores y 9 de grupos.

Instancia	Departamento	Local de Votación	[illegible]	[illegible]
I1	Amazonas	C. E. Primaria Varones 16192 Bagua	15	1,350
I2	Cusco	Centro Educativo Mixto Nuestra Señora De Belén	21	1,760
I3	Lima	I.E. Especial Nro 10 Solidaridad	14	1,204
I4	Cajamarca	U.N. De Cajamarca	14	1,217
I5	Apurímac	C.M.S. José Maria Arguedas	15	1,581
I6	Lambayeque	Institución Educativa Secundaria Jorge Chávez	17	1,504
I7	Ancash	C.N. Coronel Bolognesi	18	1,743
I8	Huancavelica	IE Francisca Diez Canseco De Castilla	20	1,698
I9	Lima	I.E. 1117 Andrés Avelino Aramburu	20	1,735
I10	Lima	Colegio Mi Hogar y Escuela	14	1,742
Total			168	15,534

Tabla 1: Instancia de Prueba

En la Tabla 1, se muestra las instancias de pruebas consideradas de 8 departamentos (vea columna 2), con un total de 168 grupos y 15,534 electores.

En la Tabla 2, se muestra el total de mesas obtenidas por el proceso usual (empírico) que tiene la ONPE y por el algoritmo BFD propuesto. De estos resultados se puede observar una reducción del número de mesas del orden del 14.46%. Así mismo, la solución obtenida presenta una eficiencia respecto a la cota inferior (vea columna 4) del orden del 92.42%.

I1	8	6	5.40
I2	10	8	7.04
I3	7	6	4.82
I4	7	6	4.87
I5	8	7	6.32
I6	8	7	6.02
I7	9	8	6.97
I8	9	8	6.79
I9	10	8	6.94
I10	7	7	6.97
Total	83	71	66

Tabla 2: Cantidad de mesas obtenido por el proceso usual (empírico) y por el algoritmo BFD

Sin embargo, en la Tabla 3 se puede observar que la reducción del número de mesas obtenido por el algoritmo BFD no es acompañado por la dispersión (desviación estándar) la cual empeora de 18.39% obtenida por el procedimiento usual a 38.72% obtenida por el algoritmo BFD, más detalles se describe en Pintado (2007). En la Tabla 3 se observa también que los dos procedimientos de mejoría (PM1+2) consiguen mejorar la dispersión de 38.72% para 17.66%, esto es, el número de electores de las mesas de sufragio se encuentra mejor distribuido.

6. CONCLUSIONES

En el presente trabajo se ha propuesto un algoritmo BFD y dos procedimientos de mejora (BFD+2M) para resolver el problema combinado de empaquetamiento y programación de tareas independientes en máquinas idénticas. La propuesta se ha aplicado al caso de fusión de mesas de sufragio del proceso electoral de la ONPE.

Los experimentos numéricos muestran resultados significativamente mejores que los obtenidos por el proceso usual (empírico) de la ONPE, se ha conseguido reducir el número de mesas de sufragio por local de votación en 14.46% y se ha mejorado la uniformidad de la distribución de electores por mesas en 3.97%.

REFERENCIAS

Campello R., Maculan N. Algoritmos e Heuristicas Desenvolvimiento e Avaliaçao de performance. Apolo Nacional Editores, Brasil (1992).

Delgadillo R. "Un estudio algorítmico del problema de corte y empaquetamiento 2D" Universidad Nacional Mayor de San Marcos (2007).

Glover F., and Kochemberger G. Handbook of metaheuristics, kluwer of academic Publisher (2003).

Mauricio D. y Delgadillo R., "Algoritmos FFD y BFD para el Problema de cortes de guillotina en 2D" Reporte técnico FISI, UNMSM. (2002).

Pintado J.L. "Un algoritmo goloso para el problema de la programación de tareas independientes en maquinas idénticas para la resolución de la conformacion de mesas de sufragio de un proceso electoral", Universidad Nacional Mayor de San Marcos, (2007).

Tupia M. "Un algoritmo Voraz Adaptativo y Randómico para resolver el problema de Programación de Tareas independientes en máquina homogéneas", Pontificia Universidad Católica del Perú (2001).

Tupia M. y Mauricio D. "Un algoritmo voraz para resolver el problema de programación de tareas dependientes en máquinas diferentes" Revista de investigación Sistemas Informáticos N°1, UNMSM, (2004).

ONPEa, Oficina Nacional de Procesos Electorales del Perú. http://www.onpe.gob.pe/, visitado el 15 de julio del 2007.

ONPEb, Elecciones regionales y municipales 2006, Oficina Nacional de Procesos Electorales del Perú. http://www.onpe.gob.pe/elecciones2006/ERM_2006.php, visitado el 15 de julio del 2007.

[illegible]													
	Usual	20	18	17	17	17	17	17	87			168,	34,98
I1	BFD	24	19	18	24	22	24					225,	27,19
	PM1+2	22	20	19	24	24	23					225,	21,14
	Usual	17	17	17	86	17	17	24	17	17	20	176	39,17
I2	BFD	24	63	24	21	24	23	24	24			220,	64,41
	PM1+2	23	19	23	20	23	20	20	23			220,	17,70
	Usual	17	17	17	17	17	17	17				172	1,15
I3	BFD	17	17	18	18	23	24					200,	32,84
	PM1+2	24	17	18	18	24	18					200,	32,42
	Usual	17	17	17	19	17	16	17				173,	8,21
I4	BFD	24	17	22	24	18	14					202,	39,16
	PM1+2	18	23	23	20	18	18					202,	25,47
I5	Usual	19	19	21	20	10	22	22	20			197,	38,23
	BFD	24	24	21	23	15	24	23				225,	31,43
	PM1+2	22	21	21	22	24	22	22				225,	9,77
	Usual	18	18	18	18	17	18	23	17			188	18,39
I6	BFD	22	21	98	22	24	24	24				214,	53,53
	PM1+2	21	21	21	21	21	21	21				214,	2,27
	Usual	19	19	19	19	20	19	19	19	18		193,	4,47
I7	BFD	24	25	22	20	24	24	70	25			217,	61,48
	PM1+2	21	21	21	21	22	21	22	22			217,	5,11
	Usual	16	17	24	25	17	18	17	17	17		188,	33,85
I8	BFD	24	18	17	20	24	24	24	15			212,	38,72
	PM1+2	24	17	18	18	19	24	24	24			212,	31,58
	Usual	18	17	17	17	17	17	17	17	16	17	173,	4,50
I9	BFD	21	18	18	15	25	25	24	25			216,	36,88
	PM1+2	19	18	18	18	24	24	24	24			216,	29,55
	Usual	24	24	24	25	24	24	25				248,	0,90
I10	BFD	25	24	25	24	24	25	25				248,	1,57
	PM1+2	25	24	25	24	24	25	25				248,	1,57
	usual											188,	18,39
Total	BFD											218,	38,72
	PM1+2											218,	17,66

Tabla 3: Distribución de electores por mesas de sufragio