

# Flexible Job Shop Scheduling con Lag Times

## Desarrolladores:

Juan Pablo Rosas Baldazo y Citlali Maryuri Olvera Toscano

## Validación y reporte:

Gabriela Sánchez Yepez, Beatriz Alejandra García Ramos,  
Johanna Bolaños Zuñiga, Óscar Alejandro Hernández López,  
Saúl Domínguez Casasola, Salvador de Jesús Vicencio Medina

Universidad Autónoma de Nuevo León

## I. Introducción

El problema de *Flexible Job Shop Scheduling* es una extensión del problema clásico de *Job Shop Scheduling*, en el que se permite seleccionar para cada operación una de las máquinas elegibles durante la programación. Por otro lado, el *time lag* es un desfase debido a la demora entre la finalización de una operación y la hora de inicio de la siguiente operación del mismo trabajo. Este desfase puede producirse cuando, por ejemplo, es necesario secar o enfriar los productos antes de que puedan realizarse otras operaciones.

Formalmente, podemos definir el problema de *Job Shop Scheduling con Lag Times* (FJS-LT) como sigue: considérese  $r$  trabajos denotados por  $J_1, \dots, J_r$  y  $m$  máquinas diferentes  $M_1, \dots, M_m$ . Cada trabajo  $J_i$  consiste en un número  $n_i$  de operaciones  $O_{i1}, \dots, O_{in_i}$  el cual tiene que ser procesado en este orden. Asociado a cada operación  $O_{ij}$  hay un conjunto  $\mathcal{M}_{ij} \subseteq \{M_1, \dots, M_m\}$  de máquinas y un tiempo de procesamiento  $p_{ij}$ . Cada operación  $O_{ij}$  debe ser procesada en exactamente una máquina y

las máquinas no pueden procesar más de un trabajo a la vez. Además, para cada operación  $O_{ij}$  hay una demora mínima entre las operaciones consecutivas  $l_{i,i+1}$  (lag min) que se debe de respetar y una demora máxima entre las operaciones consecutivas  $l_{i,i+1}$  (lag max) que no se debe sobrepasar. Por lo tanto, el problema consiste en asignar para cada operación una máquina y simultáneamente determinar la secuencia de operaciones en cada máquina junto con sus tiempos de inicio, de tal manera que el tiempo de finalización para cada una de las operaciones sea el menor posible.

En este trabajo se presenta un modelo de programación entera (MILP) y un modelo de programación con restricciones (CP) para dar solución al problema FJS-LT. El objetivo de este trabajo es presentar una comparativa entre estos dos métodos.

Este documento se encuentra organizado como sigue. En la sección II se presenta un modelo matemático de programación entera para el FJS-LT. En la sección III, se muestra un modelo basado en programación con restricciones. En la sección IV, se muestran los resultados de la comparativa del desempeño entre ambos modelos. Finalmente, en la sección V, se exponen las conclusiones y comentarios finales de este trabajo.

## II. MILP

En esta sección proponemos un modelo de programación entera. Se considera que para cada operación  $O_{ij}$ , el índice  $i$  actúa como un identificador único. Donde, a través de la función  $\xi(i)$  podemos determinar el trabajo  $j$  al que pertenece la operación  $i$ . Por tal motivo, se ha omitido el índice  $j$  que corresponde a los trabajos en este modelo. Tenga en cuenta que los identificadores de las operaciones se encuentran ordenados de manera ascendente. Así, si  $\xi(7) = \xi(2)$ , entonces, las operaciones  $O_7$  y  $O_2$  pertenecen al mismo trabajo, y además,  $O_2$  debe realizarse antes que  $O_7$ . La notación utilizada para este modelo se puede encontrar en la tabla 1.

Tabla 1. Notación para el MILP.

Símbolo	Definición
<b>Índices y conjuntos</b>	
$i, j$	índices de operaciones
$m$	índices de máquinas
$O$	conjunto de operaciones
$o_i$	operación $i$ , $o_i \in O$
$M$	conjunto de máquinas
$M_i$	conjunto de máquinas en las que $o_i$ puede ser procesada, $M_i \subseteq M$
<b>Parámetros</b>	
$p_{im}$	tiempo de procesamiento de $o_i$ en la máquina $m$
$\underline{l}_{i,i+1}$	retraso mínimo entre el comienzo de $o_{i+1}$ y el tiempo en que realizó $o_i$
$\bar{l}_{i,i+1}$	retraso máximo entre el comienzo de $o_{i+1}$ y el tiempo en que realizó $o_i$
$\mathcal{M}$	un número muy grande
<b>Variables de decisión</b>	
$C_{\max}$	makespan
$c_i$	tiempo de finalización de $o_i$
$s_i$	tiempo de inicio de $o_i$
$v_{im}$	variable binaria. Si la operación $i$ se realiza en la máquina $m$ , $v_{im} = 1$ . Si la operación $i$ no se realiza en la máquina $m$ , $v_{im} = 0$ .
$x_{ij}$	variable binaria. Si $o_j$ se realiza inmediatamente después de $o_i$ , $x_{ij} = 1$ . En caso contrario, $x_{ij} = 0$ .

$$\text{Mín} \quad C_{\max}, \quad (1)$$

$$s.a : \quad C_{\max} \geq c_i, \quad \forall i \in O, \quad (2)$$

$$c_i = s_i + p_{im}v_{im}, \quad \forall i \in O, \forall m \in M_i, \quad (3)$$

$$c_i + \underline{l}_{i,i+1} \leq s_{i+1}, \quad \forall i, i+1 \in O, \xi(i) = \xi(i+1), \quad (4)$$

$$c_i + \bar{l}_{i,i+1} \geq s_{i+1}, \quad \forall i, i+1 \in O, \xi(i) = \xi(i+1), \quad (5)$$

$$\sum_{m \in M_i} v_{im} = 1, \quad \forall i \in O, \quad (6)$$

$$c_i \leq s_j + \mathcal{M}(3 - x_{ij} - v_{im} - v_{jm}), \quad \forall i, j \in O, \quad (7)$$

$$c_j \leq s_i + \mathcal{M}(2 + x_{ij} - v_{im} - v_{jm}), \quad \forall i, j \in O, \quad (8)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in O, \quad (9)$$

$$v_{im} \in \{0, 1\}, \quad \forall i \in O, \forall m \in M_i, \quad (10)$$

$$s_i, c_i, C_{\max} \in \mathbb{N}, \quad \forall i \in O. \quad (11)$$

La función objetivo (1) minimiza el makespan, es decir, el tiempo de terminación de la última operación en la última máquina. Las restricciones (2) se aseguran que el makespan sea mayor o igual a los tiempos de terminación de todas las operaciones. Las restricciones (3) calculan el tiempo de terminación de cada operación, el cual es la suma del tiempo de inicio más el tiempo de procesamiento. Las restricciones (4) y (5) garantizan que los tiempos de lag min y lag max sean respetados. Las restricciones (6) establecen que todas las operaciones deben de realizarse. El conjunto de restricciones (7) y (8) garantizan que la secuencia de las operaciones sea respetada. Finalmente, las restricciones (9), (10) y (11) indican la naturaleza de las variables.

### III. CP

Para resolver el problema FJS-LT utilizando programación con restricciones se presenta el siguiente modelo. La tabla 2 muestra las variables y parámetros empleados.

**Tabla 2. Notación para el CP.**

Símbolo	Definición
<b>Índices y conjuntos</b>	
$i$	índices de operaciones
$m$	índices de máquinas
$O$	conjunto de operaciones
$M$	conjunto de máquinas
$O_m$	conjunto de operaciones que pueden ser procesadas en la máquina $m$ , $O_m \subseteq O$
<b>Parámetros</b>	
$\underline{l}_{i,i+1}$	retraso mínimo entre el comienzo de $o_{i+1}$ y el tiempo en que realizó $o_i$
$\bar{l}_{i,i+1}$	retraso máximo entre el comienzo de $o_{i+1}$ y el tiempo en que realizó $o_i$
<b>Variables de decisión</b>	
$operations_i$	variable de intervalo asociada a la operación $j$
$machops_{im}$	variable de intervalo que representa la asignación/posición de la operación $i$ en la máquina $m$
$opalts_{im}$	variable de intervalo que representa el tiempo de procesamiento de la operación $i$ en la máquina $m$

## Modelo

$$\text{Mín} \quad \max_{i \in O} \{ \text{End}(\text{operations}_i) \}, \quad (12)$$

$$s.a : \quad \text{NoOverlap}(\text{machops}_{im}), \quad \forall m \in M, i \in O_m, \quad (13)$$

$$\text{EndBeforeStart}(\text{operations}_i, \text{operations}_{i+1}, \bar{l}_{i,i+1}), \forall i, i+1 \in O, \xi(i) = \xi(i+1), (14)$$

$$\text{StartBeforeEnd}(\text{operations}_{i+1}, \text{operations}_i, \bar{l}_{i,i+1}), \forall i, i+1 \in O, \xi(i) = \xi(i+1), (15)$$

$$\text{Alternative}(\text{operations}_i, \text{opalts}_{im}), \quad \forall i \in O. \quad (16)$$

La función objetivo (12) minimiza el máximo tiempo de terminación de las operaciones (makespan), las restricciones (13) se aseguran que no haya traslapes de operaciones en las máquinas. El conjunto de restricciones (14) y (15) garantizan que se respete tanto la secuencia de operaciones como los tiempos de lag min y lag max, respectivamente. Finalmente, las restricciones (16) verifican que todas las operaciones sean realizadas.

## IV. Resultados

Los modelos MILP y CP se implementaron en lenguaje C++ y fueron resueltos utilizando el optimizador CPLEX versión 20.10. El criterio de paro fue de 600 segundos para cada instancia analizada. Los experimentos fueron realizados en una computadora con sistema operativo macOS Catalina 10.15.5 con procesador Octa-core Intel Xeon E5 y memoria RAM de 16 GB.

Para evaluar el desempeño de los modelos se emplearon un conjunto de instancias propuestas en la investigación de Brucker et al. [1]. Las instancias se clasificaron en cuatro grupos (sdata, edata, rdata y vdata). En total se analizaron 160 instancias, 40 por cada grupo. La descripción de cada grupo se muestra a continuación.

- **sdata**: cada operación puede ser asignada a una sola máquina (problemas de *Job Shop*).
- **edata**: pocas operaciones que pueden ser asignadas a diferentes máquinas.
- **rdata**: las operaciones pueden asignarse a un número reducido de máquinas diferentes.
- **vdata**: cada operación puede asignarse a muchas máquinas diferentes.

Un resumen de los resultados se puede ver en la tabla 3. Para cada uno de los métodos, la columna **F/O** indica el número de soluciones factibles y soluciones óptimas alcanzadas. Las columnas **Promedio tiempo (s)**, **Mejor tiempo (s)**, y **Peor tiempo (s)**, muestran el tiempo de ejecución promedio, mínimo y máximo, respectivamente para cada grupo de instancias. En el apéndice A, se encuentra información más detalla de cada una de las instancias analizadas.

Tabla 3. Resumen de resultados obtenidos.

Grupos	MILP				CP			
	F/O	Promedio tiempo (s)	Mejor tiempo (s)	Peor tiempo (s)	F/O	Promedio tiempo (s)	Mejor tiempo (s)	Peor tiempo (s)
sdata	40 / 10	456.92	3	600	40 / 16	417.87	3	600
edata	40 / 10	466.52	14	600	40 / 13	437.40	5	600
rdata	40 / 0	600	600	600	40 / 10	471.07	12	600
vdata	40 / 0	600	600	600	40 / 6	524.9	34	600
Totales	160 / 20	160 / 45						

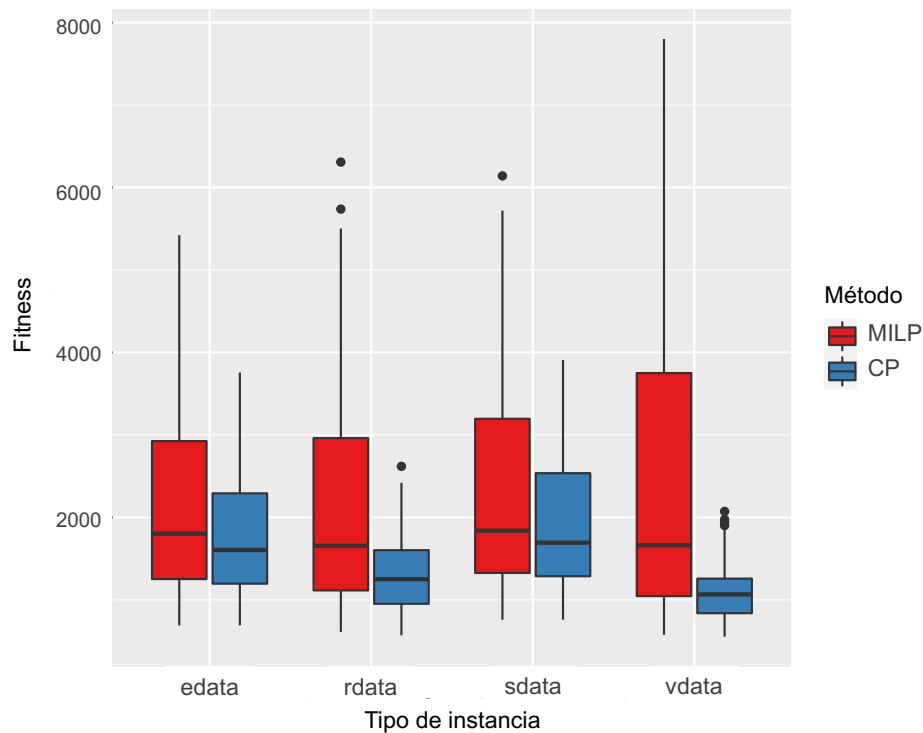


Figura 1. Fitness obtenido por los métodos MILP y CP.

Los resultados mostrados en la tabla 3 indican que tanto el MILP como el CP logran encontrar solucio-

nes factibles para cada instancia analizada. Ambos métodos llegaron al límite de tiempo (600 segundos) en al menos una instancia de cada grupo. Para el grupo sdata el mejor y el peor tiempo de ejecución es igual sin importar el método. Sin embargo, el CP presenta casi 40 segundos menos con respecto al tiempo promedio de ejecución. Además, logra encontrar 16 soluciones óptimas, en contraste con el MILP que logra obtener solo 10, con respecto a lo reportado en la literatura. De igual manera, para las instancias edata, el MILP y el CP obtienen 10 y 13 soluciones óptimas, respectivamente. El mejor tiempo de ejecución para el CP es de 5 segundos, mientras que para el MILP es de 14. El CP también presenta un tiempo promedio menor, teniendo una diferencia de cerca de 30 segundos. Para los grupos de instancias rdata y vdata el MILP no encuentra soluciones óptimas, por lo que en cada instancia de estos grupos alcanza el tiempo límite de ejecución. Por otro lado, el CP para una de las instancias del grupo rdata logra obtener optimalidad en 12 segundos y en 34 segundos para los casos del grupo vdata. Con base a lo reportado en la literatura, para estos grupos de instancias el CP alcanza 10 y 6 casos con soluciones óptimas, respectivamente.

La figura 1 presenta una comparación entre ambos métodos con respecto al valor de la función objetivo (*fitness*). El eje horizontal indica el grupo de instancias y el eje vertical el valor objetivo obtenido. Se logra apreciar que hay una diferencia significativa entre los valores alcanzados por el MILP y el CP, obteniendo el CP para cada caso resultados más favorables.

## V. Conclusiones

El problema de *Flexible Job Shop con Lag Times* (FJS-LT) es una variante del problema de *Job Shop* donde cada operación puede ser procesada en una o más máquinas. Además, en este trabajo se considera que cada operación tiene un tiempo de desfase mínimo y máximo que deben respetarse. Este problema inspirado en situaciones reales es difícil de resolver, ya que se encuentra catalogado como un problema NP-difícil. Para el problema FJS-LT, proponemos una formulación de programación entera (MILP) y una basada en programación con restricciones (CP). Nuestras pruebas computacionales muestran que la formulación CP propuesta supera a la formulación MILP en cada grupo de las instancias propuestas. Las formulaciones en CP son una alternativa eficaz para algunos problemas de optimización combinatoria que no se pueden resolver fácilmente con las técnicas tradicionales de programación matemática, como es el caso de los problemas de programación de actividades.

## A. Tablas de resultados

Los resultados obtenidos para cada instancia analizada se pueden consultar a continuación. En las tablas 4, 5, 6 y 7 se muestran los resultados de los grupos sdata, edata, rdata y vdata, respectivamente.

## Referencias

- [1] P. Brucker, S. Heitmann, and J. Hurink. Flow-shop problems with intermediate buffers. *OR Spectrum*, 25(4):549–574, 2003.



**Tabla 4. Resultados para el grupo de instancias sdata.**

Instancia	Maqs	Jobs	MILP			CP	
			Gap	Fitness	Tiempo	Fitness	Tiempo
sdata/la01	5	10	0	933	5	933	3
sdata/la02	5	10	0	903	3	903	3
sdata/la03	5	10	0	<b>817</b>	4	<b>817</b>	3
sdata/la04	5	10	0	834	5	834	4
sdata/la05	5	10	0	760	5	760	4
sdata/la06	5	15	0.27	1240	600	1204	538
sdata/la07	5	15	0.35	1210	600	1119	362
sdata/la08	5	15	0.25	1228	600	1162	141
sdata/la09	5	15	0.32	1392	600	1294	470
sdata/la10	5	15	0.25	1304	600	1228	188
sdata/la11	5	20	0.6	1812	600	1597	600
sdata/la12	5	20	0.52	1515	600	1380	599
sdata/la13	5	20	0.59	1691	600	1584	599
sdata/la14	5	20	0.59	1801	600	1609	599
sdata/la15	5	20	0.58	1744	600	1693	599
sdata/la16	10	10	0	1471	76	1471	11
sdata/la17	10	10	0	1266	29	1266	16
sdata/la18	10	10	0	1334	48	1334	15
sdata/la19	10	10	0	1413	57	1413	18
sdata/la20	10	10	0	1362	45	1362	11
sdata/la21	10	15	0.48	2283	600	1946	600
sdata/la22	10	15	0.37	1865	600	1694	600
sdata/la23	10	15	0.47	2188	600	1894	600
sdata/la24	10	15	0.46	2052	600	1857	600
sdata/la25	10	15	0.45	2024	600	1772	536
sdata/la26	10	20	0.65	2994	600	2494	600
sdata/la27	10	20	0.66	3248	600	2635	600
sdata/la28	10	20	0.67	3233	600	2571	600
sdata/la29	10	20	0.64	2790	600	2327	600
sdata/la30	10	20	0.66	3183	600	2520	600
sdata/la31	10	30	0.83	5719	600	3648	600
sdata/la32	10	30	0.84	6141	600	3908	600
sdata/la33	10	30	0.83	5376	600	3448	600
sdata/la34	10	30	0.82	5064	600	3582	600
sdata/la35	10	30	0.79	4824	600	3554	600
sdata/la36	15	15	0.53	2972	600	2541	600
sdata/la37	15	15	0.55	3308	600	2777	599
sdata/la38	15	15	0.58	3158	600	2355	599
sdata/la39	15	15	0.55	3317	600	2535	599
sdata/la40	15	15	0.62	3277	600	2558	599

**Tabla 5. Resultados para el grupo de instancias edata.**

Instancia	Maqs	Jobs	MILP			CP	
			Gap	Fitness	Tiempo	Fitness	Tiempo
edata/la01	5	10	0	877	21	877	6
edata/la02	5	10	0	844	25	844	9
edata/la03	5	10	0	749	20	749	5
edata/la04	5	10	0	776	14	776	5
edata/la05	5	10	0	<b>691</b>	18	<b>691</b>	7
edata/la06	5	15	0.43	1220	600	1154	599
edata/la07	5	15	0.4	1188	600	1078	599
edata/la08	5	15	0.34	1221	600	1129	527
edata/la09	5	15	0.35	1263	600	1209	599
edata/la10	5	15	0.28	1191	600	1134	164
edata/la11	5	20	0.65	1840	600	1574	599
edata/la12	5	20	0.62	1536	600	1369	599
edata/la13	5	20	0.57	1587	600	1472	599
edata/la14	5	20	0.58	1717	600	1543	599
edata/la15	5	20	0.61	1761	600	1597	599
edata/la16	10	10	0	1346	178	1346	19
edata/la17	10	10	0	1159	124	1159	13
edata/la18	10	10	0	1282	46	1282	15
edata/la19	10	10	0	1329	105	1329	23
edata/la20	10	10	0	1318	110	1318	22
edata/la21	10	15	0.41	1990	600	1747	599
edata/la22	10	15	0.47	1885	600	1638	599
edata/la23	10	15	0.47	2059	600	1773	600
edata/la24	10	15	0.49	2030	600	1689	600
edata/la25	10	15	0.33	1767	600	1614	495
edata/la26	10	20	0.68	3006	600	2384	599
edata/la27	10	20	0.65	2980	600	2384	600
edata/la28	10	20	0.69	3304	600	2374	599
edata/la29	10	20	0.67	2886	600	2138	599
edata/la30	10	20	0.55	2699	600	2357	600
edata/la31	10	30	0.83	5212	600	3361	600
edata/la32	10	30	0.82	5342	600	3757	600
edata/la33	10	30	0.85	5423	600	3205	600
edata/la34	10	30	0.84	5356	600	3368	600
edata/la35	10	30	0.81	4694	600	3379	600
edata/la36	15	15	0.57	2907	600	2227	600
edata/la37	15	15	0.58	3132	600	2453	600
edata/la38	15	15	0.61	2995	600	2192	599
edata/la39	15	15	0.51	2734	600	2234	600
edata/la40	15	15	0.54	2709	600	2270	600

Tabla 6. Resultados para el grupo de instancias rdata.

Instancia	Maqs	Jobs	MILP			CP	
			Gap	Fitness	Tiempo	Fitness	Tiempo
rdata/la01	5	10	0.3	753	600	675	13
rdata/la02	5	10	0.23	706	600	653	12
rdata/la03	5	10	0.36	657	600	583	55
rdata/la04	5	10	0.33	654	600	610	52
rdata/la05	5	10	0.37	612	600	<b>570</b>	128
rdata/la06	5	15	0.59	1200	600	973	599
rdata/la07	5	15	0.57	1096	600	923	599
rdata/la08	5	15	0.57	1071	600	931	599
rdata/la09	5	15	0.57	1196	600	1061	599
rdata/la10	5	15	0.6	1124	600	960	599
rdata/la11	5	20	0.7	1656	600	1299	599
rdata/la12	5	20	0.68	1441	600	1115	599
rdata/la13	5	20	0.72	1651	600	1225	600
rdata/la14	5	20	0.72	1606	600	1305	599
rdata/la15	5	20	0.66	1595	600	1339	600
rdata/la16	10	10	0.3	1054	600	900	150
rdata/la17	10	10	0.29	934	600	820	149
rdata/la18	10	10	0.29	994	600	917	55
rdata/la19	10	10	0.34	1121	600	964	210
rdata/la20	10	10	0.28	1158	600	965	29
rdata/la21	10	15	0.65	2126	600	1254	599
rdata/la22	10	15	0.59	1769	600	1157	600
rdata/la23	10	15	0.66	2033	600	1303	599
rdata/la24	10	15	0.56	1654	600	1249	600
rdata/la25	10	15	0.58	1770	600	1238	600
rdata/la26	10	20	0.75	2968	600	1680	600
rdata/la27	10	20	0.79	3455	600	1747	600
rdata/la28	10	20	0.76	3262	600	1691	600
rdata/la29	10	20	0.75	3010	600	1543	600
rdata/la30	10	20	0.73	2936	600	1653	600
rdata/la31	10	30	0.86	5323	600	2324	600
rdata/la32	10	30	0.88	6308	600	2619	600
rdata/la33	10	30	0.87	5503	600	2335	600
rdata/la34	10	30	0.84	4581	600	2419	600
rdata/la35	10	30	0.87	5738	600	2339	600
rdata/la36	15	15	0.59	2550	600	1586	600
rdata/la37	15	15	0.71	3571	600	1677	600
rdata/la38	15	15	0.6	2429	600	1488	600
rdata/la39	15	15	0.62	2756	600	1551	600
rdata/la40	15	15	0.67	2959	600	1529	600

**Tabla 7. Resultados para el grupo de instancias vdata.**

Instancia	Maqs	Jobs	MILP			CP	
			Gap	Fitness	Tiempo	Fitness	Tiempo
vdata/la01	5	10	0.42	724	600	633	599
vdata/la02	5	10	0.39	669	600	593	132
vdata/la03	5	10	0.4	597	600	<b>554</b>	599
vdata/la04	5	10	0.37	662	600	582	81
vdata/la05	5	10	0.33	577	600	560	600
vdata/la06	5	15	0.59	1096	600	923	599
vdata/la07	5	15	0.63	1052	600	860	599
vdata/la08	5	15	0.62	1003	600	862	599
vdata/la09	5	15	0.66	1165	600	948	599
vdata/la10	5	15	0.58	1084	600	892	599
vdata/la11	5	20	0.75	1662	600	1173	599
vdata/la12	5	20	0.74	1586	600	1084	600
vdata/la13	5	20	0.76	1622	600	1167	600
vdata/la14	5	20	0.72	1605	600	1216	600
vdata/la15	5	20	0.76	1580	600	1242	600
vdata/la16	10	10	0.34	1106	600	735	34
vdata/la17	10	10	0.24	875	600	664	79
vdata/la18	10	10	0.29	959	600	681	187
vdata/la19	10	10	0.38	1028	600	719	600
vdata/la20	10	10	0.22	998	600	774	92
vdata/la21	10	15	0.68	2312	600	1042	600
vdata/la22	10	15	0.7	2133	600	948	600
vdata/la23	10	15	0.77	2922	600	1063	600
vdata/la24	10	15	0.57	1663	600	994	600
vdata/la25	10	15	0.73	2792	600	949	599
vdata/la26	10	20	0.79	3565	600	1377	600
vdata/la27	10	20	0.8	3554	600	1419	600
vdata/la28	10	20	0.78	3555	600	1374	600
vdata/la29	10	20	0.78	3361	600	1246	600
vdata/la30	10	20	0.81	4021	600	1273	600
vdata/la31	10	30	0.89	6829	600	1941	600
vdata/la32	10	30	0.9	7800	600	2073	600
vdata/la33	10	30	0.9	7781	600	1865	600
vdata/la34	10	30	0.9	6745	600	1977	600
vdata/la35	10	30	0.91	7650	600	1902	600
vdata/la36	15	15	0.75	3840	600	1236	600
vdata/la37	15	15	0.82	5581	600	1393	600
vdata/la38	15	15	0.74	3721	600	1070	600
vdata/la39	15	15	0.8	4682	600	1215	600
vdata/la40	15	15	0.75	3952	600	1253	600