

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS
INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS
CENTRO DE INVESTIGACIÓN EN CIENCIAS

CASOS SOLUBLES EN TIEMPO POLINOMIAL PARA EL PROBLEMA
 $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$ Y LA REDUCCIÓN AL PROBLEMA
SUBSET SUM

TESIS PROFESIONAL
PARA OBTENER EL GRADO DE:
LICENCIADO EN CIENCIAS COMPUTACIONALES Y COMPUTACIÓN CIENTÍFICA

PRESENTA:
ALEJANDRO REYNOSO GÓMEZ

ASESOR:
DR. NODARI VAKHANIA MAISURADZE

CUERNAVACA MOR.

OCTUBRE 2020

Esta tesis fue evaluada por el siguiente jurado:

Presidente:	Dra. Larissa Sbitneva
Secretario:	Dra. Lorena Díaz González.
Vocal:	Dr. Nodari Vakhania Maisuradze.
Suplente 1:	Dr. Federico Alonso Pecina.
Suplente 2:	Dr. José Crispín Zavala Díaz.

Resumen

Estudiamos un problema de calendarización para una sola máquina con dos tiempos de liberación permitidos y dos tiempos de entrega permitidos. Este es un caso especial del problema general de calendarización NP-duro en sentido estricto con una sola máquina, en donde se cuenta con un número arbitrario de tiempos de liberación y un número arbitrario de tiempos de entrega, con el objetivo de minimizar el máximo tiempo de completéz total. Nuestro problema resulta ser más transparente y accesible que la versión general, por lo que podemos realizar un análisis y obtener algunas propiedades de manera simple y que sean eficientes para solucionarlo.

El caso restringido sigue siendo útil para algunas aplicaciones de la vida real y es útil para analizar el problema general sobre la base de las propiedades aquí establecidas.

Las condiciones de optimalidad que obtenemos para dos tiempos de liberación y dos tiempos de entrega, podrían generalizarse a un número constante de tiempos de liberación y tiempos de entrega. Los criterios de optimalidad establecidos son condiciones explícitas bajo las cuales, el problema restringido se puede resolver de manera óptima en tiempo $O(n \log(n))$. Normalmente se puede esperar que, para un problema NP-duro, tales condiciones deberían ser muy restrictivas; pero por el contrario, nuestro extenso estudio computacional ha demostrado que estas condiciones, utilizadas de manera combinada, cubren prácticamente todos los casos problemáticos que pueden surgir en la práctica.

Nuestro algoritmo tiene un rendimiento práctico sobresaliente debido a su complejidad $O(n \log(n))$, el cual invoca cinco heurísticas diferentes que se ejecutan realizando varias combinaciones entre estas para cada entrada (la salida de cada algoritmo heurístico es una solución factible). El resultado para 50 millones de instancias generadas aleatoriamente, muestran que al menos una heurística ha satisfecho al menos una de nuestras condiciones de optimalidad.

También abordamos la posibilidad (teórica) de que ninguna de nuestras condiciones se cumpla, por lo que mostramos como un algoritmo de programación dinámica que resuelve el problema de SUMA DE SUBCONJUNTOS (SUBSET SUM), puede usarse para la solución del problema de calendarización en tiempo pseudo-polinomial. En una parte considerable de las instancias probadas, uno de los cinco calendarios obtenidos por cada una de nuestras heurísticas, también fue resuelta de manera óptima por SUBSET SUM.

Abstract

We study a single-machine scheduling problem with two allowable job release and delivery times. This special case of a strongly NP-hard single-machine scheduling problem with an arbitrary number of job release and delivery times and with the objective to minimize the maximum job completion time remains NP-hard. Nevertheless, it is more transparent and accessible than the general version so that some nice properties, yielding simple and efficient solution of the problem, can be established. The restricted setting still remains useful for some real-life applications. No less importantly, the study of the restricted setting is helpful to analyze the general problem on the basis of the properties established here. In particular, the optimality conditions that we obtain for two job release and delivery times might potentially be generalized to a constant number of job release and delivery times. The established optimality criteria are explicit conditions under which the restricted problem can be solved optimally in time $O(n \log n)$. One may normally expect that, for an NP-hard problem, such conditions should be somewhat very restrictive. But in contrary, our extensive computational study has shown that these conditions, used in a combined fashion, cover virtually all problem instances that may arise in practice. We report an outstanding practical performance of an $O(n \log n)$ -time procedure of our algorithm. It invokes five different heuristics that run in consecutive fashion for every input (the output of each heuristic algorithm being a feasible solution). For 50 million of randomly generated problem instances, the output of at least of the heuristics has satisfied at least one of our optimality conditions (so all these problem instances were solved optimally). We also address a (theoretical) possibility that none of our conditions is satisfied, and show how a known dynamic programming algorithm for SUBSET SUM problem can be used for the solution of the scheduling problem in pseudo-polynomial time. For a considerable part of the tested instances, one of the five scheduling heuristics has solved optimally also SUBSET SUM problem.

A mi familia.

Agradecimientos

Agradezco mucho a mi madre por todo su apoyo, sus cuidados y consejo. Ella ha sido una pieza importante para lograr todas mis metas.

Quiero agradecer al Dr. Nodari Vakhania por su tiempo, asesoramiento y rigor científico. Sin su esfuerzo y compromiso con el proyecto, simple y sencillamente no hubiera sido posible la realización del presente trabajo de investigación.

A mi comité tutorial, mi más sincero agradecimiento por todo su apoyo y comprensión.

A mi jurado evaluador, les agradezco profundamente su tiempo y dedicación para corregir el escrito final y fungir como miembros de mi jurado evaluador.

A mis amigos y amigas que han estado presentes a lo largo de esta etapa de mi vida. Gracias por su cariño, comprensión y apoyo incondicional; soy muy afortunado por tenerlos.

Y quiero agradecer a Javier Aguilar por ayudarme con mis primeros pasos en la ciencia.

Índice general

1. Introducción	1
1.1. Nuestra contribución	1
1.2. Posibles aplicaciones	4
1.3. Cómo está organizado este documento	4
1.4. Algunos trabajos relacionados	4
1.4.1. Resultados de aproximabilidad para el caso general $1 r_j, q_j C_{\text{máx}}$	4
1.4.2. Casos especiales solucionables en tiempo polinomial para el problema P $1 r_j, q_j C_{\text{max}}$	5
2. Las cinco heurísticas	6
3. Algunas definiciones y propiedades básicas	17
4. Condiciones de optimalidad para el problema $1 r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\} C_{\text{max}}$	21
5. Subrutina pseudopolinomial usando SUBSET SUM	25
6. Experimentos computacionales	30
7. Conclusión y trabajos futuros	35
7.1. Appendix	39

Capítulo 1

Introducción

Dado un conjunto J de n trabajos con *tiempos de liberación*, *tiempos de procesamiento* y *tiempos de entrega*. Los trabajos del conjunto J deben ser calendarizados en una sola máquina. Las siguientes restricciones deben cumplirse: *i)* El trabajo j no se puede calendarizar antes de su tiempo de liberación r_j , *ii)* necesita un tiempo de procesamiento continuo p_j en la máquina; y, *iii)* Una vez completado un trabajo en la máquina, necesita un tiempo de entrega adicional q_j para su completéz total (la entrega del trabajo j es independiente de la máquina y no requiere más recursos, ya que lo realiza un agente independiente).

Un *calendario factible* S asigna cada trabajo j a la máquina en el intervalo de tiempo $[t_j(S), t_j(S)+p_j)$ en ejes de tiempo no negativos, de tal forma que $t_j(S) \geq r_j$ y este intervalo no tiene intersección con el intervalo de ningún otro trabajo (consideramos intervalos de tiempo medio abiertos), donde $t_j(S) \geq r_j$ es el *tiempo de inicio* del trabajo j y $c_j(S) = t_j(S) + p_j$ es el *tiempo de completéz* del trabajo j en el calendario S . El *tiempo de completéz total* del trabajo j en el calendario S está dado por $C_j(S) = c_j(S) + q_j$. Para un calendario factible S , el *makespan* $C_{\max}(S) = \max_j C_j(S)$ es el máximo tiempo de completéz total de todos los trabajos en el conjunto J . El objetivo es encontrar un calendario *óptimo* S_{opt} , el cual está dado por un calendario factible con el mínimo tiempo de completéz total $C_{\max}(S_{opt}) = \min_S C_{\max}(S)$.

El problema, convencionalmente abreviado como $1|r_j, q_j|C_{\max}$ (la notación de tres barras introducida por Graham et al. [5]), es bien conocido por ser NP-duro en sentido estricto (Garey y Johnson [3]). La versión con restricciones $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$ que nosotros estudiamos aquí, con solo dos posibles tiempos de liberación r^1, r^2 ($r^1 < r^2$) y dos posibles tiempos de entrega q^1, q^2 ($q^1 < q^2$) permanece (ligeramente) NP-duro (Chinos y Vakhania) [2].

1.1. Nuestra contribución

Nuestra contribución es doble. Por un lado, mostramos cómo un problema de calendarización para una sola máquina NP-duro $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$ se puede resolver, en la práctica, en un

tiempo $O(n \log n)$ con una probabilidad muy cercana a 1, gracias a las condiciones de optimalidad y las heurísticas que proponemos aquí. En particular, nuestro estudio empírico demostró que para un número extremadamente grande de instancias, al menos uno de los cinco algoritmos que proponemos, ofrece una solución que es óptima en al menos una de las seis condiciones de optimalidad que derivamos. Por otro lado, nuestro estudio para el caso restringido $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$ con dos tiempos de liberación y dos tiempos de entrega, revela buenas propiedades estructurales para el caso general $1|r_j, q_j|C_{máx}$, en particular, permite la generalización del método propuesto para un número constante de tiempos de liberación y tiempos de entrega. La generalización de las heurísticas propuestas y las condiciones de optimalidad para ese entorno es posible y requerirá el estudio de nuevos casos que surgirían con tres o más tiempos de liberación y de entrega (consulte la Sección Conclusión para obtener más detalles). Otra contribución teórica para nuestro problema $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$ se refiere a una propiedad para las instancias donde ninguna de nuestras heurísticas crea una solución que cumpla al menos una de las condiciones de optimalidad derivadas (que es muy poco probable que ocurra en la práctica). Al establecer vínculos estrechos con el problema SUBSET SUM, mostramos cómo un algoritmo pseudo-polinomial para resolver SUBSET SUM también se puede utilizar para resolver nuestro problema de calendarización.

Nuestros criterios de optimalidad son condiciones explícitas bajo las cuales el problema restringido $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$ se puede resolver de manera óptima en un tiempo $O(n \log n)$. Estas condiciones se verifican en una subrutina de tiempo $O(n \log n)$ la cual invoca a una, dos o a las cinco heurísticas para que estas se ejecuten de manera consecutiva. La salida de cada algoritmo heurístico es una solución factible obtenida en el tiempo $O(n \log n)$. La entrada (la instancia del problema correspondiente) se resuelve de manera óptima si la salida (la solución factible construida) de al menos una de las heurísticas satisface al menos una de nuestras condiciones de optimalidad.

Cada una de las cinco heurísticas propuestas es una variación de la heurística LDT (*Long Delivery Time*) estándar, que consiste en programar un trabajo con el mayor tiempo de entrega de aquellos que están disponibles, y están especialmente diseñadas para nuestro problema de calendarización con dos tiempos de liberación y dos tiempos de entrega (y se pueden ampliar para tres o más tiempos de liberación y de entrega). En general, la misma condición de optimalidad puede ser válida para cada una de las soluciones entregadas por las diferentes heurísticas. Aunque en la práctica, el rendimiento de cada heurística es diferente, la frecuencia con la que se satisface una condición de optimalidad dada y las características de las instancias del problema, dependen de las heurísticas por las cuales se generó la solución correspondiente (observe que las condiciones de optimalidad se verifican en las soluciones creadas por la heurística).

En esta investigación:

- Hemos probado las condiciones de optimalidad para nuestras heurísticas individualmente y las hemos clasificado según su rendimiento práctico (la solución generada por la heurística es óptima si al menos una de las condiciones se cumple para esa solución. En la práctica sin embargo, dos o más condiciones fueron simultáneamente verificadas).
- Hemos combinado a la heurística con el mejor comportamiento práctico con el resto de las cuatro heurísticas en cuatro pares diferentes (duetos) para verificar nuestras seis condiciones de optimalidad de manera combinada.

- Hemos llevado a cabo los experimentos para estos cuatro duetos y luego para las cinco heurísticas juntas (el quinteto) verificando las seis condiciones de optimalidad para las soluciones generadas por estas heurísticas.
- Hemos generado 50 millones de instancias para cada una de las 10 combinaciones anteriores para probar la eficiencia de nuestras condiciones heurísticas y de optimalidad (las instancias se generaron aleatoriamente, para un número diferente de trabajos y rangos diferentes a partir de los cuales los tiempos de procesamiento, liberación y entrega fueron derivado).

Nuestros resultados experimentales para el quinteto resultaron ser extremadamente fuertes en la práctica:

- Se verificó optimalidad para *todas* las 50 millones de instancias probadas.
- Para las heurísticas individuales, al menos una de nuestras seis condiciones se cumplió en casi el 87 % para la peor de las heurísticas, y en más del 99 % para la mejor de las heurísticas.
- Para los duetos, la eficiencia del peor par ha alcanzado ya más del 99.9 % (para cada grupo, estas estadísticas se derivaron para los 50 millones de instancias correspondientes).

Por lo tanto, para todos los efectos prácticos, nuestras condiciones de optimalidad “cubren” todas las instancias de este problema que puedan surgir en la práctica. Sin embargo, teóricamente es posible que exista una instancia del problema para la cual nuestra subrutina de tiempo polinomial no garantice una solución óptima (es decir, ninguna de nuestras condiciones se cumple). Para esos posibles casos, mostramos cómo un conocido algoritmo de programación dinámica para resolver el problema de SUBSET SUM (débilmente NP-duro), puede usarse para obtener la solución del problema de calendarización en tiempo pseudo-polinomial.

Nuestra subrutina pseudopolinomial resuelve una instancia especialmente derivada del problema SUBSET SUM con complejidad $O(nB)$; donde B es el valor objetivo correspondiente (ver Garey y Johnson [3]). La subrutina crea una solución para el problema de calendarización gracias a las propiedades estructurales derivadas del problema de calendarización y la estrecha relación establecida de ese problema con el problema SUBSET SUM.

Nuestros experimentos computacionales han arrojado un resultado secundario práctico para el problema SUBSET SUM. Una de las versiones propuestas de la heurística LDT ha resuelto de manera óptima, las instancias correspondientes del problema SUBSET SUM para al menos el 21 % de todas las instancias de calendarización probadas; mientras que para las instancias con una gran cantidad de trabajos (750 y 1000 trabajos), esto sucedió en al menos el 32 % de las instancias. El resultado se entregó instantáneamente para las instancias resueltas de manera óptima. En todos los 500 millones de instancias probadas, el tiempo de ejecución de la computadora para los 10 grupos de 50 millones de instancias varió desde 1 hora hasta un poco menos de 8 horas (incluidas instancias que no fueron resueltas de manera óptima por ninguna de nuestras condiciones de optimalidad, es decir, para las cuales se invocó la subrutina pseudo-polinomial para el problema SUBSET SUM).

1.2. Posibles aplicaciones

El problema de calendarización $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$ puede tener algunas aplicaciones directas de la vida real. Por ejemplo: tomamos una fábrica de acabado de piezas de repuesto recibidas de dos proveedores diferentes. Las piezas de repuesto (sin terminar) del proveedor 1 (proveedor 2, respectivamente) llegan al tiempo r^1 (r^2 , respectivamente) y deben procesarse (terminarse) en la máquina. Hay dos clientes a los que se deben entregar las piezas terminadas. El tiempo de entrega del cliente 1 (cliente 2, respectivamente) es q^1 (q^2 , respectivamente). El objetivo es minimizar el tiempo en el que todos los productos terminados se entregan a ambos clientes, de manera equivalente, para encontrar un calendario factible que minimice el tiempo máximo de completéz total de un producto.

1.3. Cómo está organizado este documento

La siguiente subsección contiene una breve descripción de algunos trabajos relacionados. En el capítulo 2 describimos nuestras cinco heurísticas. En el capítulo 3 damos algunos conceptos básicos y propiedades conocidas anteriormente que usaremos más adelante. En el capítulo 4 establecemos nuestras condiciones de optimalidad. En el capítulo 5 describimos la subrutina pseudo-polinomial que se puede aplicar si no se cumple ninguna de las condiciones del capítulo 4. En el capítulo 6 describimos detalladamente los datos que hemos utilizados para construir las instancias de nuestro problema, también analizamos el rendimiento de nuestras heurísticas y de las condiciones de optimalidad para estas instancias. el capítulo 7 contiene nuestras observaciones finales. En el Apéndice, especificamos los datos estadísticos detallados y los resultados experimentales para las instancias probadas.

1.4. Algunos trabajos relacionados

Algunos resultados conocidos relacionados con el problema $1|r_j, q_j|C_{max}$ y sus casos especiales son los siguientes:

1.4.1. Resultados de aproximabilidad para el caso general $1|r_j, q_j|C_{máx}$

La heurística de LDT proporciona una solución de 2 aproximaciones (una solución, como máximo es dos veces peor que una óptima). Esta heurística, descrita por primera vez por Schrage [11], es una extensión del algoritmo heurístico propuesto anteriormente por Jackson [7] para el caso sin tiempos de liberación.

Potts [10] demostró que mediante la aplicación repetitiva de la heurística LDT $O(n)$ veces la relación de aproximación se puede mejorar a $\frac{3}{2}$.

Hall y Shmoys [6] han sugerido una modificación de la heurística LDT, que resulta en una relación de aproximación de $\frac{4}{3}$ al aplicarla simultáneamente a las versiones hacia adelante y hacia atrás del problema. Nowicki y Smutnicki [8] justificaron la idea de la utilidad de la aplicación simultánea de la heurística LDT en las modas hacia adelante y hacia atrás.

En [15], la magnitud κ , no mayor que el valor objetivo óptimo, dividido por el tiempo de procesamiento del trabajo especialmente determinado, se utiliza para obtener una relación de aproximación precisa en el peor de los casos de $1 + \frac{1}{\kappa}$ para la heurística LDT. El valor del parámetro κ puede determinarse brutalmente en un tiempo $O(n \log n)$ usando un límite inferior fácilmente calculable en el valor objetivo óptimo en lugar de usar el valor objetivo óptimo en sí mismo.

1.4.2. Casos especiales solucionables en tiempo polinomial para el problema $P \mid r_j, q_j \mid C_{max}$

Si todos los tiempos de liberación o de entrega son iguales o el tiempo de procesamiento de todos los trabajos es 1 (y los tiempos de liberación de trabajos son enteros), entonces el algoritmo heurístico de [7] encuentra una solución óptima.

Si se pueden ordenar los trabajos de modo que $d_1 \leq \dots \leq d_n$ y $d_1 - \alpha r_1 - \beta p_1 \geq \dots \geq d_n - \alpha r_n - \beta p_n$, por algunos $\alpha \in [0, +\infty)$ y $\beta \in [0, 1]$, entonces el problema puede resolverse en tiempo polinómico Lazarev y Arkhipov [9] (este resultado se indica para una versión equivalente de el problema en el que el tiempo de entrega q_j de cada trabajo j se reemplaza por sus fechas de vencimiento d_j y la tardanza máxima del trabajo $L_{m\acute{a}x}$ se minimiza, ver, por ejemplo, [1] para la equivalencia).

Supongamos que cualquier par de trabajos j e i con $r_i > r_j$ y $d_i < d_j$, $d_j - r_j - p_j \leq d_i - r_i - p_i$, y si $r_i + p_i \geq r_j + p_j$, luego $d_i \geq d_j$. Entonces el problema sigue siendo polinomial [16].

Cuando se impone una restricción solo en los tiempos de procesamiento del trabajo, Garey et al. [4] ha resuelto eficientemente la versión de factibilidad si todos los trabajos tienen el mismo tiempo de procesamiento. En [13] este resultado se generalizó para la versión de minimización con dos tiempos de procesamiento de trabajo posibles, y recientemente, el resultado posterior se generalizó aún más para los tiempos de procesamiento de trabajo mutuamente divisibles [17].

Capítulo 2

Las cinco heurísticas

En esta sección describimos nuestras cinco heurísticas. Antes de eso, necesitamos introducir algunas notaciones. Denotamos por $J(r^i, q^j)$ el conjunto de trabajos con tiempos de liberación y entrega r^i y q^j , respectivamente, y por $J(r_i)$ ($J(q_j)$, respectivamente) el conjunto de trabajos con tiempo de liberación r_i (tiempo de entrega q_j , respectivamente), $i, j = 1, 2$. Dado un conjunto de trabajos A , definimos $P(A)$ como la suma de los tiempos de procesamiento de todos los trabajos del conjunto A .

Sin pérdida de generalidad, supongamos que $J(r^1, q^1) \neq \emptyset$ y $J(r^2, q^2) \neq \emptyset$ caso contrario el calendario LDT σ es óptimo (vea Lema 1 en la siguiente sección). Notemos que, mientras se calendarizan los trabajos del conjunto $J(r^1)$, los trabajos del conjunto $J(r^1, q^2)$ pueden ser asignados antes de los trabajos del conjunto $J(r^1, q^1)$ (demostramos esta propiedad básica que cumple un calendario óptimo para el problema 1 | $r_i \in \{r^1, r^2\}$, $q_i \in \{q^1, q^2\}$ | C_{max} en Lema 2 de la siguiente sección).

Las cuatro variaciones de la heurística LDT y la heurística LTD estándar tienen partes en común; las diferencias entre estas surgen básicamente en la forma en que los trabajos se asignan alrededor del tiempo r^2 . Primero describimos las partes comunes y las diferencias y luego damos una descripción formal de cada una de las heurísticas. A continuación especificamos las partes que son similares en las diferentes heurísticas y dónde se produce las diferencias.

Las cinco heurísticas consisten en *dos fases*; la primera fase se ocupa de los trabajos del conjunto $J(r^1)$, y la segunda fase se ocupa de los trabajos del conjunto $J(r^2)$ y algunos de los trabajos que pudieron no ser calendarizados del conjunto $J(r^1, q^1)$. La segunda fase es básicamente la misma para todas las heurísticas. Las cinco heurísticas comienzan con la primera fase asignando los trabajos del conjunto $J(r^1, q^2)$ a partir del tiempo r^1 de forma continua en cualquier orden pero sin crear ningún hueco entre ellos. En seguida, algunos trabajos del conjunto $J(r^1, q^1)$ son asignados consecutivamente hasta dejar un pequeño hueco antes del tiempo r^2 , de esta forma un solo trabajo del conjunto $J(r^1, q^1)$ es completado después o al tiempo r^2 (dado que $r^1 + P(J(r^1, q^2)) < r^2$. ver Lema 2 en la siguiente sección); luego, utilizando la regla LDT, todos los trabajos del conjunto $J(r^2, q^2)$ son asignados continuamente (recordamos que $q^2 \geq q^1$), finalmente, todos los trabajos del conjunto $J(r^2, q^1)$ son asignados junto con el resto de los trabajos del conjunto $J(r^1, q^1)$ de forma continua. El caso $r^1 + P(J(r^1, q^2)) \geq r^2$ es fácilmente tratable (consulte la siguiente sección). La diferencia

entre las heurísticas se produce en el conjunto $J(r^1, q^1)$ a partir del tiempo $r^1 + P(J(r^1, q^2))$.

Comenzamos con la descripción estándar de la *heurística LDT* que hemos adoptado para el problema 1 | $r_i \in \{r^1, r^2\}$, $q_i \in \{q^1, q^2\}$ | C_{max} . La heurística LDT procede con n asignaciones. La primera parte de la fase uno comienza asignando a partir del tiempo r^1 los trabajos del conjunto $J(r^1, q^2)$ de forma continua. La segunda parte de la fase uno consiste en asignar iterativamente a la tarea del conjunto $J(r^1, q^1)$ con el mayor tiempo de procesamiento, de tal forma que la última tarea del conjunto $J(r^1, q^1)$ en ser asignada comience su procesamiento antes del tiempo r^2 y sea completada después o al tiempo r^2 . La primera parte de la fase dos consiste en asignar inmediatamente a las tareas del conjunto $J(r^2, q^2)$ (no importa el orden) después de que la última tarea del conjunto $J(r^1, q^1)$ haya terminado su procesamiento, note que si todas las tareas del conjunto $J(r^1, q^1)$ terminan su procesamiento antes del tiempo r^2 , la primera parte de la fase dos comienza al tiempo r^2 . La segunda parte de la fase dos asigna al resto de los trabajos con tiempo de entrega q^1 inmediatamente después de los trabajos del conjunto $J(r^2, q^2)$, note que esta última parte puede contener trabajos tanto de $J(r^1, q^1)$ como trabajos de $J(r^2, q^1)$.

Note que, dado que la heurística siempre asigna un trabajo lo más temprano posible cada vez que la máquina queda inactiva, no crea ningún *hueco* (tiempo en el que la máquina queda inactividad) que se pueda evitar. El pseudocódigo del nivel de implementación es el siguiente.

ALGORITHM LDT {returns LDT-schedule S }

(1,1) $t := r^1$;

WHILE ($J(r^1, q^2) \neq \emptyset$) DO { schedule jobs of set $J(r^1, q^2)$ }

 schedule any $j \in J(r^1, q^2)$ at time $t_j^S = t$;

$J(r^1, q^2) := J(r^1, q^2) \setminus \{j\}$;

$t = t + p_j$

END

(1,2) WHILE ($J(r^1, q^1) \neq \emptyset$) DO { schedule jobs of set $J(r^1, q^1)$ }

 Find job $j \in J(r^1, q^1)$ with the largest processing time p_j

 IF $t < r^2$ THEN:

 schedule j at time $t_j^S = t$;

$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\}$;

$t = t + p_j$

ELSE:

$$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\};$$

$$J(r^2, q^1) := J(r^2, q^1) \cup \{j\};$$

END

(2,1) $t := \max\{t, r^2\}$ { $t < r^2$ only if all jobs from $J(r^1)$ are scheduled }

WHILE ($J(r^2, q^2) \neq \emptyset$) DO { schedule jobs of set $J(r^2, q^2)$ }

schedule any $j \in J(r^2, q^2)$ at time $t_j^S = t$;

$$J(r^2, q^2) := J(r^2, q^2) \setminus \{j\};$$

$$t = t + p_j$$

END

(2,2) WHILE ($J(r^2, q^1) \neq \emptyset$) DO { schedule the remaining jobs }

schedule any $j \in J(r^2, q^1)$ at time $t_j^S = t$;

$$J(r^2, q^1) := J(r^2, q^1) \setminus \{j\};$$

$$t = t + p_j$$

END

RETURN S .

La heurística se ejecuta en un tiempo $O(n \log n)$ ya que en cada una de las n asignaciones al calendario, se lleva a cabo la búsqueda de un elemento en una lista ordenada.

Considere el tiempo de finalización $r^1 + P(J(r^1, q^2))$ del último trabajo calendarizado del conjunto $J(r^1, q^2)$. Como se señaló anteriormente, el caso $r^1 + P(J(r^1, q^2)) \geq r^2$ es fácilmente tratable y la salida de las cinco heurísticas (makespan), en este caso, es la misma (en ninguna de ellas se crea un desplazamiento de las tareas más urgentes). Por otro lado, en el caso $r^1 + P(J(r^1, q^2)) + P(J(r^1, q^1)) \leq r^2$ las cinco heurísticas asignan a todos los trabajos del conjunto $J(r^1, q^1)$ continuamente a partir del tiempo $r^1 + P(J(r^1, q^2))$ y su salida (makespan) de nuevo es la misma para las cinco versiones. En ambos casos, la salida de todas las cinco heurísticas es óptima (consulte la siguiente sección).

Ahora describimos cómo cada una de las cuatro variaciones procede a partir del tiempo $r^1 + P(J(r^1, q^2))$ en la segunda parte de la fase uno para el caso cuando $r^1 + P(J(r^1, q^2)) + P(J(r^1, q^1)) > r^2$ (el caso contrario lo veremos más adelante). Sea $L = (j_1, \dots, j_k)$ una lista con los trabajos del

conjunto $J(r^1, q^1)$ de tal forma que estos trabajos están ordenados de manera no crecientemente con base a sus tiempos de procesamiento.

En la primera variación, la *heurística LDTG*, deja un espacio (posiblemente de longitud 0) antes del tiempo r^2 . Iterativamente, si el próximo trabajo j de la lista L verifica que $t + p_j \leq r^2$, entonces j es asignado al tiempo t y actualizamos a $t = t + p_j$. De lo contrario el trabajo j es reubicado al conjunto $J(r^2, q^1)$. El siguiente trabajo (el más largo) de la lista (que se convierte en el trabajo j actual) se trata de manera similar, hasta que se consideren todos los trabajos de la lista L .

ALGORITHM LDTG {returns LDTG-schedule S }

(1,1) $t := r^1$;

WHILE ($J(r^1, q^2) \neq \emptyset$) DO { schedule jobs of set $J(r^1, q^2)$ }

 schedule any $j \in J(r^1, q^2)$ at time $t_j^S = t$;

$J(r^1, q^2) := J(r^1, q^2) \setminus \{j\}$;

$t = t + p_j$

END

(1,2) WHILE ($J(r^1, q^1) \neq \emptyset$) DO { schedule jobs of set $J(r^1, q^1)$ }

 Find job $j \in J(r^1, q^1)$ with the largest processing time p_j

 IF $t + p_j \leq r^2$ THEN:

 schedule j at time $t_j^S = t$;

$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\}$;

$t = t + p_j$

 ELSE:

$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\}$;

$J(r^2, q^1) := J(r^2, q^1) \cup \{j\}$;

END

(2,1) $t := \max\{t, r^2\}$

WHILE ($J(r^2, q^2) \neq \emptyset$) DO { schedule jobs of set $J(r^2, q^2)$ }

schedule any $j \in J(r^2, q^2)$ at time $t_j^S = t$;

$J(r^2, q^2) := J(r^2, q^2) \setminus \{j\}$;

$t = t + p_j$

END

(2,2) WHILE ($J(r^2, q^1) \neq \emptyset$) DO { schedule the remaining jobs }

schedule any $j \in J(r^2, q^1)$ at time $t_j^S = t$;

$J(r^2, q^1) := J(r^2, q^1) \setminus \{j\}$;

$t = t + p_j$

END

RETURN S .

La segunda variación, la *heurística LDTN*, es similar a la versión estandar con la diferencia de que genera un ‘pequeño’ desplazamiento para el primer trabajo del conjunto $J(r^2, q^2)$ (el siguiente trabajo que será asignado justo en el momento de finalización del último trabajo del conjunto $J(r^1, q^1)$). Iterativamente, los trabajos de la lista L se asignan de manera similar hasta que el siguiente trabajo j_i de la lista se complete en el momento r^2 o después. Si el siguiente trabajo se completa al tiempo r^2 , es asignado y se completa la fase 1 de la heurística LDTN (en este caso j_i es el último trabajo asignado en la fase 1). Si el siguiente trabajo j_i de la lista L se completa después del tiempo r^2 , entonces uno de los trabajos j_i, j_{i+1}, \dots, j_k se asigna antes del tiempo r^2 (inmediatamente después del trabajo j_{i-1}). En particular, el trabajo $j_l, i \leq l \leq k$, con el índice máximo l (note que nos referimos al trabajo con el menor tiempo de procesamiento que termine su procesamiento después o al tiempo r^2) tal que $c(j_{i-1}) + p_{j_l} \geq r^2$ será asignado al tiempo $c(j_{i-1})$. Tal trabajo definido como j_l será el último trabajo calendarizado en la fase 1 (sin embargo, los trabajos que aún no han sido asignados de la lista L se pospondrán).

ALGORITHM LDTN {returns LDTN-schedule S }

(1,1) $t := r^1$;

WHILE ($J(r^1, q^2) \neq \emptyset$) DO { schedule jobs of set $J(r^1, q^2)$ }

schedule any $j \in J(r^1, q^2)$ at time $t_j^S = t$;

$J(r^1, q^2) := J(r^1, q^2) \setminus \{j\}$;

$t = t + p_j$

END

(1,2) WHILE $(J(r^1, q^1) \neq \emptyset)$ DO { schedule or postpone jobs of set $J(r^1, q^1)$ }

Find job $j \in J(r^1, q^1)$ with the largest processing time p_j

IF $t + p_j \leq r^2$ THEN:

schedule j at time $t_j^S = t$;

$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\}$;

$t = t + p_j$

ELSE:

Find job $j \in J(r^1, q^1)$ with the shortest processing time p_j

IF $t + p_j < r^2$ or $t \geq r^2$ THEN:

$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\}$;

$J(r^2, q^1) := J(r^2, q^1) \cup \{j\}$;

ELSE:

schedule j at time $t_j^S = t$;

$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\}$;

$t = t + p_j$

END

(2,1) $t := \max\{t, r^2\}$; { $t < r^2$ only if all jobs from $J(r^1)$ are scheduled }

WHILE $(J(r^2, q^2) \neq \emptyset)$ DO { schedule jobs of set $J(r^2, q^2)$ }

schedule any $j \in J(r^2, q^2)$ at time $t_j^S = t$;

$J(r^2, q^2) := J(r^2, q^2) \setminus \{j\}$;

$t = t + p_j$

END

(2,2) WHILE $(J(r^2, q^1) \neq \emptyset)$ DO { schedule jobs of set $J(r^2, q^1)$ }

schedule any $j \in J(r^2, q^1)$ at time $t_j^S = t$;

$$J(r^2, q^1) := J(r^2, q^1) \setminus \{j\};$$

$$t = t + p_j$$

END

RETURN S .

En la tercera versión de la heurística LDT, la *heurística LDTA*, los trabajos largos (con el mayor tiempo de procesamiento) de la lista L se posponen, de la siguiente manera. Iterativamente, consideramos el trabajo j_i y la desigualdad $r^1 + P(J(r^1, q^2)) + P(L) - r^2 - z \geq p_{j_i}$, donde inicialmente, $z := 0$. Si se satisface esta desigualdad, entonces el trabajo j_i se pospone y $z := z + p_{j_i}$; de lo contrario, el trabajo j_i se asigna a la hora de finalización del trabajo asignado previamente. La fase 1 se completa con la iteración del último trabajo en la lista L :

ALGORITHM LDTA {returns LDTA-schedule S }

$$(1,1) \quad t := r^1; \quad z := r^1 + P(J(r^1)) - r^2;$$

WHILE ($J(r^1, q^2) \neq \emptyset$) DO { schedule jobs of set $J(r^1, q^2)$ }

 schedule any $j \in J(r^1, q^2)$ at time $t_j^S = t$;

$$J(r^1, q^2) := J(r^1, q^2) \setminus \{j\};$$

$$t = t + p_j$$

END

$$(1,2) \quad \text{WHILE } (J(r^1, q^1) \neq \emptyset) \text{ DO } \{ \text{schedule jobs of set } J(r^1, q^1) \}$$

 Find job $j \in J(r^1, q^1)$ with the largest processing time p_j

 IF $p_j \geq z$ THEN:

 schedule j at time $t_j^S = t$;

$$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\};$$

$$t = t + p_j$$

 ELSE:

$$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\};$$

$$J(r^2, q^1) := J(r^2, q^1) \cup \{j\};$$

$$z := z - p_j;$$

END

(2,1) $t := \max\{t, r^2\}$ { $t < r^2$ only if all jobs from $J(r^1)$ are scheduled }

WHILE ($J(r^2, q^2) \neq \emptyset$) DO { schedule jobs of set $J(r^2, q^2)$ }

schedule any $j \in J(r^2, q^2)$ at time $t_j^S = t$;

$J(r^2, q^2) := J(r^2, q^2) \setminus \{j\}$;

$t = t + p_j$

END

(2,2) WHILE ($J(r^2, q^1) \neq \emptyset$) DO { schedule the remaining jobs }

schedule any $j \in J(r^2, q^1)$ at time $t_j^S = t$;

$J(r^2, q^1) := J(r^2, q^1) \setminus \{j\}$;

$t = t + p_j$

END

RETURN S .

En la cuarta versión de la heurística LDT, la *heurística LDTV*, el trabajo j_i se define como en la descripción de la *heurística LDTN* anterior. Es decir, $i - 1$ es el índice del trabajo máximo en la lista L tal que $r^1 + P(J(r^1, q^2)) + \sum_{l=1}^{i-1} p_{j_l} \leq r^2$ (entonces $r^1 + P(J(r^1, q^2)) + \sum_{l=1}^i p_{j_l} > r^2$). Los trabajos j_1, \dots, j_{i-1} se incluyen desde el momento de tiempo $r^1 + P(J(r^1, q^2))$, de forma similar a las heurísticas descritas anteriormente, de igual forma la fase uno se completa si $r^1 + P(J(r^1, q^2)) + \sum_{l=1}^{i-1} p_{j_l} = r^2$.

Ahora, supongamos que $r^1 + P(J(r^1, q^2)) + \sum_{l=1}^{i-1} p_{j_l} < r^2$. Definamos el tiempo de calendarización t como el tiempo de completéz del último trabajo del conjunto $J(r^1)$ calendarizado antes del tiempo r^2 . Sea U un conjunto temporal que contendrá algunos de los trabajos aún no asignados de la lista L , Inicialmente, $U := \emptyset$. De tal forma que $t + P(U) < r^2$. Sea x un trabajo temporal, inicialmente vacío. Si el siguiente trabajo j_ι ($\iota = i, \dots, k$) verifica que $t + P(U) + p_{j_\iota} < r^2$, entonces agregamos al trabajo j_ι al conjunto U . Caso contrario los trabajos del conjunto U son calendarizados desde el momento t y el trabajo x se pospone (se remueve del conjunto $J(r^1, q^1)$ y se añade al conjunto $J(r^2, q^1)$), y actualizamos $t := t + P(U)$, $U := \emptyset$ y actualizamos a $x := j_\iota$

Observemos que el último trabajo j_k de la lista L ya sea (1) se incluye en el conjunto U o (2) se convierte en el trabajo x . En el caso (1), todos los trabajos del conjunto U se posponen para la fase 2 (al conjunto $J(r^2, q^1)$) y la fase 1 se completa calendarizando el trabajo x en el momento t . Para el caso (2), el trabajo actual en x se pospone al conjunto $J(r^2, q^1)$, los trabajos del conjunto U son calendarizado a partir del tiempo t y el trabajo $j_k = x$ es calendarizado inmediatamente después del

último trabajo del conjunto U ($t + P(U)$). La descripción del nivel de implementación es la siguiente:

ALGORITHM LDTV {returns LDTV-schedule S }

(1,1) $x := null$ $z := 0$ $U := \{\}$ $t := r^1$;

WHILE ($J(r^1, q^2) \neq \emptyset$) DO { schedule jobs of set $J(r^1, q^2)$ }

 schedule any $j \in J(r^1, q^2)$ at time $t_j^S = t$;

$J(r^1, q^2) := J(r^1, q^2) \setminus \{j\}$;

$t = t + p_j$

END

(1,2) Find job $j \in J(r^1, q^1)$ with the largest processing time p_j

WHILE ($t + p_j \leq r^2$) DO { schedule or posponer jobs of set $J(r^1, q^1)$ }

 schedule j at time $t_j^S = t$;

$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\}$;

$t = t + p_j$

 Find job $j \in J(r^1, q^1)$ with the largest processing time p_j

END

(1,3) WHILE ($J(r^1, q^1) \neq \emptyset$) DO { schedule or posponer jobs of set $J(r^1, q^1)$ }

 Find job $j \in J(r^1, q^1)$ with the largest processing time p_j

 IF $t + p_j + z \geq r^2$ THEN:

 IF $x = null$ THEN:

$x := j$

$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\}$;

 ELSE:

$J(r^2, q^1) := J(r^2, q^1) \cup \{x\}$;

$x := j; z := 0$

WHILE ($U \neq \emptyset$) DO { schedule jobs of set U }

 schedule any $i \in U$ at time $t_i^S = t$;

$U := U \setminus \{i\}$;

$t = t + p_i$

END

ELSE:

$U := U \cup \{j\}$;

$J(r^1, q^1) := J(r^1, q^1) \setminus \{j\}$;

$z := z + p_j$;

END

schedule x at time $t_x^S = t$;

$t = t + p_x$

WHILE ($U \neq \emptyset$) DO { postpone jobs of set U }

 find any $i \in U$;

$J(r^2, q^1) := J(r^2, q^1) \cup \{i\}$;

$U := U \setminus \{i\}$;

END

(2,1) $t := \max\{t, r^2\}$; { $t < r^2$ only if all jobs from $J(r^1)$ are scheduled }

WHILE ($J(r^2, q^2) \neq \emptyset$) DO { schedule jobs of set $J(r^2, q^2)$ }

 schedule any $j \in J(r^2, q^2)$ at time $t_j^S = t$;

$J(r^2, q^2) := J(r^2, q^2) \setminus \{j\}$;

$t = t + p_j$

END

(2,2) WHILE $(J(r^2, q^1) \neq \emptyset)$ DO { schedule jobs of set $J(r^2, q^1)$ }

 schedule any $j \in J(r^2, q^1)$ at time $t_j^S = t$;

$J(r^2, q^1) := J(r^2, q^1) \setminus \{j\}$;

$t = t + p_j$

END

RETURN S .

Tenga en cuenta que en la fase dos, el calendario parcial de la fase uno se extiende a un calendario completo factible con los trabajos restantes aún no calendarizados y que la fase dos es la misma para todas las heurísticas, excepto en las heurísticas LDT, LDTN, LDTA y LDTV, donde los trabajos del conjunto $J(r^2)$ se incluyen a partir del tiempo de completéz del último trabajo asignado del conjunto $J(r^1, q^1)$, mientras que en la heurística LDTG los trabajos del conjunto $J(r^2)$ son calendarizados a partir del tiempo r^2 (justo después de un hueco que ocurre antes del tiempo r^2).

Denotamos por σ ($\hat{\sigma}$, σ^N , σ^A y σ^V , respectivamente) el calendario completo de LDT creado por la heurística LDT (heurística LDTG, heurística LDTN, heurística LDTA y heurística LDTV, respectivamente). Es fácil ver que todas las heurísticas descritas anteriormente se pueden implementar en un tiempo $O(n \log n)$. Más adelante en la Sección 6, informamos nuestro estudio empírico de la frecuencia con la que las condiciones de optimalidad que damos en la Sección 4 son válidas para las soluciones entregadas por cada una de nuestras heurísticas (en la práctica, las soluciones obtenidas por las cuatro variaciones que proponemos dominan las obtenidas por la heurística LDT estándar).

Capítulo 3

Algunas definiciones y propiedades básicas

Ahora definimos algunos conceptos auxiliares que son útiles para comprender la estructura básica de los calendarios LDT (la terminología utilizada aquí se introdujo en [12] y [13], aunque algunos conceptos se conocían antes con diferentes nombres)

En un calendario LDT, podemos distinguir uno o más *bloques*: un bloque es la secuencia más larga de trabajos calendarizados consecutivamente sin ningún hueco entre ellos.

Un *overflow job* es el trabajo o en el calendario σ con $C_o(\sigma) \geq C_j(\sigma), \forall j \in J$, es decir, es el trabajo que genera el mayor tiempo de completéz total en el calendario σ . Denotaremos al overflow job por $o(\sigma)$. El *bloque crítico* $B(\sigma)$ en el calendario σ es el bloque que contiene al overflow job $o(\sigma)$.

El trabajo e es un *trabajo emergente* en el calendario σ si $e \in B(\sigma)$ y $q_e < q_{o(\sigma)}$ (alternativamente, un trabajo emergente se conoce como el trabajo crítico o de interferencia en la literatura).

El *kernel* del calendario σ , $K(\sigma)$ es la secuencia de trabajos calendarizados en el calendario σ entre el último trabajo emergente (el anterior al $o(\sigma)$) y el $o(\sigma)$, incluyendo al trabajo $o(\sigma)$ pero sin incluir el trabajo emergente (alternativamente, un kernel se conoce como una secuencia crítica en la literatura). Observamos fácilmente que el tiempo de entrega de cualquier trabajo en $K(\sigma)$ no es menor que el del trabajo $o(\sigma)$.

Se deduce que cada kernel está contenido en algún bloque en σ , y el número de núcleos en σ es igual al número de overflow job en él. Tenga en cuenta que si el calendario σ no tiene trabajos emergentes, entonces tampoco tendrá kernel; entonces el overflow job en el calendario σ no será parte de ningún kernel. El siguiente lema es una extensión de un resultado anterior conocido para el caso general $1|r_j, q_j|C_{\text{máx}}$ para la heurística LDT σ (por ejemplo, vea [2]).

Lema 1 *Sea S cualquiera de los calendarios $\sigma, \sigma^N, \sigma^A$ o σ^V . Si $q_j \geq q_{o(S)}$, para cada trabajo $j \in B(S)$, es decir, no existe un trabajo emergente en el calendario S , entonces S es óptimo.*

Demostración. El lema resulta trivial si el trabajo $o(S)$ comienza al momento $r_{o(S)}$ en el calendario S , esto se debe a que el makespan no puede reducirse. Caso contrario, consideramos un calendario donde el trabajo $o(S)$ no comience al tiempo $r_{o(S)}$, luego cualquier reorganización que involucre a un trabajo del bloque $B(S)$ que haya sido asignado antes del trabajo $o(S)$ y que sea reasignado después del trabajo $o(S)$ no puede llevar a un makepan menor (recordemos que $q_j \geq q_{o(S)}$). Note que esta es la única forma en que el $o(S)$ puede comenzar antes del tiempo $t_{o(S)}(S)$. Luego, si después de reorganizar algunos trabajos en $B(S)$, el trabajo $o(S)$ comienza no antes del momento $t_{o(S)}(S)$ (comienza igual o después), entonces el lema es claro. \square

Del lema anterior podemos observar que el calendario $\hat{\sigma}$ no está incluido, esto se debe a que el calendario $\hat{\sigma}$ en su construcción genera huecos de manera intencional, de esta forma, podemos deducir que el bloque generado a partir del tiempo r^2 siempre cumplirá las condiciones del lema 1 (por la forma en como se asignan los trabajos). En la siguiente sección exploraremos el caso en el que el lema 1 no aplica para el calendario $\hat{\sigma}$ y el caso para el que el lema 1 sí aplica para el calendario $\hat{\sigma}$

Fact 1 Sea S cualquiera de los calendarios σ , σ^N , σ^A o σ^V , si $J(r^1, q^1) = \emptyset$ o $J(r^2, q^2) = \emptyset$, entonces S es óptimo

Demostración. Es fácil ver que puede no existir un trabajo emergente en el calendario S si $J(r^1, q^1) = \emptyset$ o $J(r^2, q^2) = \emptyset$ y, por lo tanto, el calendario S es óptimo por Lema 1. \square

Analizando el resultado anterior, podemos deducir que si el conjunto $J(r^1, q^1) = \emptyset$, entonces el calendario $\hat{\sigma}$ es óptimo, ya que el hueco que se podría genera sería inevitable. Por otro lado, si el conjunto $J(r^2, q^2) = \emptyset$, entonces el calendario $\hat{\sigma}$ no es necesariamente óptimo, en particular, si el calendario $\hat{\sigma}$ contiene un hueco antes del tiempo r^2 (un hueco innecesario), entonces éste no será óptimo

Dado un subconjunto $A \subseteq J(r^1, q^1)$, sea $\sigma(A)$ el calendario LDT en el que los trabajos del conjunto $J(r^1, q^2)$ se incluyen en el intervalo de tiempo $[r^1, r^1 + P(J(r^1, q^2))]$ seguido de los trabajos del conjunto A , de tal forma que el resto de los trabajos en $J(r^1, q^1) \setminus A$ son pospuestos para la fase dos (son removidos del conjunto $J(r^1, q^1)$ y agregados al conjunto $J(r^2, q^1)$). Dado un subconjunto $D \subseteq J(r^1, q^2)$, sea $\sigma(A, D)$ al calendario LDT que resulta de la reasignación de las tareas del subconjunto D en el calendario $\sigma(A)$ para la fase dos (las tareas del subconjunto D son removidas del conjunto $J(r^1, q^2)$ y agregadas al conjunto $J(r^2, q^2)$).

Sea

$$\delta(\sigma(A)) = r^1 + P(J(r^1, q^2)) + P(A) - r^2$$

y sea

$$\delta(\sigma(A, D)) = \delta(\sigma(A)) - P(D),$$

respectivamente, el *desplazamiento* forzado del kernel en el calendario $\sigma(A)$ y en el calendario $\sigma(A, D)$, respectivamente.

La definición anterior hace referencia al tamaño del intervalo de tiempo en el calendario $\sigma(A)$ y

$\sigma(A, D)$ a partir del tiempo r^2 en el que los trabajos del conjunto $J(r^2)$ deben esperar para poder ser calendarizados. Notemos que los trabajos de este conjunto comienzan su procesamiento no antes del tiempo r^2

Para el siguiente lema, asumiremos que $r^1 + P(J(r^1, q^2)) \leq r^2$, caso contrario es óptimo. (ver lema 3)

Lema 2 *Existe un calendario óptimo en el cual los trabajos del conjunto $J(r^1, q^2)$ ocupan el intervalo de tiempo $[r^1, r^1 + P(J(r^1, q^2))]$.*

Demostración. Considere cualquier subconjunto $A \subseteq J(r^1, q^1)$, $D \subseteq J(r^1, q^2)$ y un calendario $\sigma(A, D)$ como se definió anteriormente. Notamos que en el calendario $\sigma(A, D)$ las tareas del conjunto D utilizan en total un intervalo de tiempo en el calendario de longitud $P(D)$ (este intervalo se encuentra antes del tiempo r^2) el cual es reasignado para después del tiempo r^2 al momento de posponer los trabajos del conjunto D (por la construcción del calendario $\sigma(A, D)$), luego, los trabajos del conjunto A comenzarán $P(D)$ unidades de tiempo antes (se cubrirá el intervalo de tiempo que tomaba procesar los trabajos del conjunto D que fueron pospuestos). De esta forma el calendario $\sigma(A, D)$ ya sea (i) contiene un hueco o (ii) no (observemos que en el calendario $\sigma(A, D)$ las únicas tareas que pueden comenzar su procesamiento antes del tiempo r^2 son las del conjunto A y $J(r^1, q^2) \setminus D$)

En el caso (ii), el tiempo de completéz total del último trabajo calendarizado (note que este es el trabajo que realiza el mayor tiempo de completéz total de todas las tareas del conjunto $J(q^1)$) en el calendario $\sigma(A)$ es

$$r^2 + \delta(\sigma(A)) + P(J(r^2)) + P(J(r^1, q^1) \setminus A) + q^1,$$

y el tiempo de completéz total del último trabajo calendarizado en la calendario $\sigma(A, D)$ es

$$r^2 + \delta(\sigma(A, D)) + P(D) + P(J(r^2)) + P(J(r^1, q^1) \setminus A) + q^1.$$

Por lo tanto, el tiempo de completéz total del último trabajo calendarizado en ambos calendarios $\sigma(A)$ y $\sigma(A, D)$ es lo mismo. En el mismo tiempo, el tiempo de completéz total del último trabajo calendarizado del conjunto $J(q^2)$ en el calendario $\sigma(A)$ es

$$r^2 + \delta(\sigma(A)) + P(J(r^2, q^2)) + q^2$$

y el tiempo completéz total del último trabajo calendarizado en el calendario $\sigma(A, D)$ es

$$r^2 + \delta(\sigma(A, D)) + P(D) + P(J(r^2, q^2)) + q^2 = r^2 + \delta(\sigma(A)) - P(D) + P(D) + P(J(r^2, q^2)) + q^2.$$

Entonces se deduce que $C_{max}(\sigma(A)) = C_{max}(\sigma(A, D))$.

En el caso (i), un hueco antes del tiempo r^2 en el calendario $\sigma(A, D)$ surge solo si $P(D) > \delta(\sigma(A))$.

Entonces, el tiempo de completéz total del último trabajo calendarizado del conjunto $J(q^2)$ en calendario $\sigma(A, D)$ es

$$r^2 + P(D) + P(J(r^2, q^2)) + q^2,$$

y en el calendario $\sigma(A)$ es

$$r^2 + \delta(\sigma(A)) + P(J(r^2, q^2)) + q^2.$$

Al mismo tiempo, el tiempo de completéz total del último trabajo calendarizado del conjunto $J(q^1)$ en el calendario $\sigma(A, D)$ es

$$r^2 + P(D) + P(J(r^2)) + P(J(r^1, q^1) \setminus A) + q^1,$$

y en el calendario $\sigma(A)$ es

$$r^2 + \delta(\sigma(A)) + P(J(r^2)) + P(J(r^1, q^1) \setminus A) + q^1.$$

Tomando en cuenta que $P(D) > \delta(\sigma(A))$, obtenemos que $C_{max}(\sigma(A)) < C_{max}(\sigma(A, D))$.

Hemos demostrado que $C_{max}(\sigma(A)) \leq C_{max}(\sigma(A, D))$, es decir, el calendario $\sigma(A)$ domina el calendario $\sigma(A, D)$. Luego, el lema está obviamente demostrado ya que escogimos cualquier par de conjuntos A y D . □

Capítulo 4

Condiciones de optimalidad para el problema $1 \mid r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\} \mid C_{max}$

En la sección anterior, dimos nuestras dos primeras condiciones cuando los calendarios σ , σ^N , σ^A y σ^V son óptimos. En esta sección, explorando más de cerca el problema $1 \mid r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\} \mid C_{max}$, derivando otras condiciones de optimalidad. En particular, demostraremos que si se cumplen ciertas propiedades para un calendario generado por cualquiera de nuestras heurísticas, entonces ese calendario es óptimo. Dado que es suficiente con crear un calendario-LDT(o alguna de sus variantes anteriores) para verificar estas condiciones, la verificación de estas condiciones de optimalidad lleva tiempo $O(n \log n)$. Nuestros criterios de optimalidad son condiciones explícitas bajo las cuales el problema restringido $1 \mid r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\} \mid C_{max}$ se puede resolver de manera óptima en el tiempo $O(n \log n)$. Estas condiciones se verifican en una subrutina de tiempo $O(n \log n)$ la cual invoca a una, dos o a las cinco heurísticas para que éstas que se ejecutan de manera consecutiva. La salida de cada algoritmo heurístico es una solución factible obtenida en el tiempo $O(n \log n)$. La entrada (la instancia del problema correspondiente) se resuelve de manera óptima si la salida (la solución factible construida) de al menos una de las heurísticas satisface al menos una de nuestras condiciones de optimalidad. Para empezar, mostramos que el Lema 1 de la sección anterior es válido también para la salida de las otras heurísticas si alguna de las condiciones de los dos Lemas siguientes 3 y 4 se satisface.

Lema 3 *Cualquiera de las cinco heurísticas, LDT, LDT-G, LDT-N, LDT-A y LDT-V obtiene una solución óptima si*

$$r^1 + P(J(r^1, q^2)) \geq r^2.$$

Demostración. Por la condición en el lema, es fácil ver que todas las soluciones generadas por las cinco heurísticas tienen el mismo makespan. Por lo tanto, se puede deducir que si una de ellas es óptima, las cuatro soluciones restantes también serán óptimas. Dada la condición del lema, observe que en un calendario LDT(o alguna de las variaciones) todos los trabajos del conjunto $J(q^2)$ se completan antes que todos los trabajos del conjunto $J(q^1)$. Luego, la condición de Lema 1 se mantiene y el calendario es óptimo. \square

Lema 4 *Cualquiera de las cinco heurísticas obtiene una solución óptima si*

$$r^1 + P(J(r^1)) \leq r^2.$$

Demostración. Como en el lema anterior, fácilmente observamos que, por la condición en el lema, las soluciones generadas por cualquiera de las cinco heurísticas obtienen el mismo makespan. Nuevamente, si una de ellas, digamos S , es óptima, las cuatro soluciones restantes también serán óptimas. Por la condición del lema, el overflow job en el calendario S será un trabajo del conjunto $J(r^2)$, y el bloque crítico $B(S)$ comenzará al tiempo r^2 . Por la naturaleza de los calendarios LDT, todos los trabajos del conjunto $J(r^2, q^2)$ se completarán antes que todos los trabajos del conjunto $J(r^2, q^1)$ en el bloque $B(S)$. Entonces la condición de Lema 1 nuevamente se mantiene y el calendario S es óptimo. \square

Por los Lemas 3 y 4, el calendario S generado por cualquiera de las 5 heurísticas puede no ser óptimo si y solo si

$$r^1 + P(J(r^1, q^2)) < r^2 < r^1 + P(J(r^1)). \quad (4.1)$$

Por tanto, de ahora en adelante, solo consideremos las instancias del problema 1 $| r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\} | C_{max}$ en las cuales la expresión 4.1 se satisface.

Lema 5 *El calendario S generado por cualquiera de las heurísticas, LDT, LDT-N, LDT-A y LDT-V es óptimo si $o(S) \in J(q^1)$.*

Prueba. Por la condición 4.1, $r^1 + P(J(r^1)) > r^2$ y por el hecho de que S es un calendario LDT (un calendario generado por cualquiera de los cuatro heurística anteriores), el último trabajo asignado en ese calendario es del conjunto $J(q^1)$. Entonces, claramente, dado que $o(S) \in J(q^1)$, $o(S)$ debe ser el último trabajo asignado en el calendario S . Dada a la condición $r^1 + P(J(r^1)) > r^2$, el calendario S no contiene ningún hueco (todo el calendario consta de un solo bloque). Además, $q_i \geq q_{o(S)}$, para todo $i \in B(S)$, y el calendario S es óptimo debido al Lema 1. \square

Sea $C^q(S)$ el tiempo máximo de completez total de los trabajos con tiempo de entrega q en el calendario S (por lo tanto, $C^{q^1}(S)$ o $C^{q^2}(S)$ es igual a $C_{o(S)}(S)$). Sea j el primer trabajo programado del conjunto $J(r^2, q^2)$ en un calendario LDT S y sea

$$\delta(S) = t_j(S) - r^2$$

el desplazamiento forzado del kernel $K(S)$ en el calendario S (tenga en cuenta que $\delta(\hat{\sigma})$ siempre es 0).

Corolario 1 *El calendario S generado por cualquiera de las cuatro heurísticas, LDT, LDT-N, LDT-A y LDT-V es óptimo si*

$$0 \leq \delta(S) \leq C^{q^1}(S) - r^2 - P(J(r^2, q^2)) - q^2.$$

Demostración. Dado que $r^1 + P(J(r^1, q^2)) < r^2$ (condición 4.1), no hay trabajos del conjunto $J(r^1, q^2)$ calendarizados después del tiempo r^2 en el calendario S . Por lo tanto,

$$C^{q^2}(S) = r^2 + \delta(S) + P(J(r^2, q^2)) + q^2$$

luego, por la condición del corolario deducimos que $C^{q^2}(S) \leq C^{q^1}(S)$. Es decir, $o(S) \in J(q^1)$ y el calendario S es óptimo según el Lema 5. \square

A continuación, también ofrecemos algunas condiciones de optimalidad para el calendario $\hat{\sigma}$.

Lema 6 *El calendario $\hat{\sigma}$ es óptimo si $o(\hat{\sigma}) \in J(r^2, q^2)$.*

Demostración. Observamos que el bloque crítico $B(\hat{\sigma})$ siempre comienza al tiempo r^2 , luego, $q_i \geq q_{o(\hat{\sigma})}$ para todo $i \in B(\hat{\sigma})$ incluido antes del trabajo $o(\hat{\sigma})$. Aplicando un razonamiento, similar a la demostración del Lema 1 a los trabajos del bloque $B(\hat{\sigma})$, obtenemos fácilmente que el calendario $\hat{\sigma}$ es óptimo. \square

Sea $g(S)$ la longitud del hueco generado antes del tiempo r^2 en el calendario S obtenido por la heurística LDT-G.

Corolario 2 *El calendario $\hat{\sigma}$ es óptimo si*

$$0 \leq g(\hat{\sigma}) \leq C^{q^2}(\hat{\sigma}) - r^1 - P(J) - q^1.$$

Demostración. Primero tenga en cuenta que el último trabajo asignado en el calendario $\hat{\sigma}$ es del conjunto $J(q^1)$, luego, $C^{q^1}(\hat{\sigma}) = r^1 + g(\hat{\sigma}) + P(J) + q^1$. Por la condición del corolario deducimos que $C^{q^1}(\hat{\sigma}) \leq C^{q^2}(\hat{\sigma})$, por lo tanto $o(\hat{\sigma}) \in J(r^2, q^2)$ y el calendario $\hat{\sigma}$ es óptimo según el Lema 6. \square

Lema 7 *El calendario S generado por cualquiera de las cinco heurísticas es óptimo si $\delta(S) = 0$ y $g(S) = 0$.*

Demostración: Observe que por las condiciones del lema, el calendario S consta de un solo bloque y que algún trabajo del conjunto $J(r^1, q^1)$ termina su procesamiento exactamente al tiempo r^2 . Dado que S es un calendario LDT, a partir del tiempo r^2 todos los trabajos del conjunto $J(q^2)$ son asignados antes que los trabajos del conjunto $J(q^1)$. Luego, la demostración procede de manera similar a la del Lema 1. \square

Resumimos la relación entre nuestras heurísticas y condiciones en la siguiente tabla (H/C representa Heurística/Condición). Observamos que el calendario S creado por nuestras heurísticas es uno de los siguientes tres tipos: (1) S contiene un espacio antes del tiempo r^2 , es decir, $\delta(S) = 0$ pero $g(S) > 0$ (por ejemplo, $\hat{\sigma}$), (2) $g(S) = 0$ pero $\delta(S) > 0$ (por ejemplo, σ , σ^N , σ^A y σ^V) y (3) $\delta(S) = 0$ y $g(S) = 0$ (el lema anterior). Además, siempre existe un calendario óptimo de alguno de estos tres tipos (ver Teorema 1 en la siguiente sección).

H/C	Lema 1	Lema 3	Lema 4	Lema 5	Lema 6	Lema 7
LDT	X	X	X	X		X
LDTG	X	X	X		X	X
LDTA	X	X	X	X		X
LDTN	X	X	X	X		X
LDTV	X	X	X	X		X

Capítulo 5

Subrutina pseudopolinomial usando SUBSET SUM

Nuestra subrutina pseudopolinomial resuelve una instancia especialmente derivada del problema SUBSET SUM. La subrutina crea una solución al problema de calendarización gracias a las propiedades estructurales derivadas del problema de calendarización y la estrecha relación establecida con el problema SUBSET SUM. Usaremos un algoritmo de programación dinámica estándar pseudopolinomial con tiempo $O(nB)$ para resolver el problema de SUBSET SUM, donde B es el valor objetivo correspondiente (ver Garey y Johnson [3]).

A lo largo de esta sección, solo consideramos instancias de nuestro problema de calendarización $1 \mid r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\} \mid C_{max}$ para el cual no se cumple ninguna de las condiciones de optimalidad de las dos secciones anteriores.

Mostraremos cómo se puede usar un algoritmo de tiempo pseudopolinomial de programación dinámica para el problema SUBSET SUM (ver Sección 4.2 en Garey y Johnson [3]) para la solución de nuestro problema de calendarización. En el problema SUBSET SUM se nos da un conjunto finito de números enteros $N = \{n_1, n_2, \dots, n_k\}$, y un número entero $B \leq \sum_{i=1}^k n_i$. SUBSET SUM es un problema de decisión con una respuesta “ sí ” si existe un subconjunto del conjunto N que suma B (la respuesta es “ no ” en caso contrario).

Dada una instancia de nuestro problema de calendarización, la instancia correspondiente de SUBSET SUM se define de la siguiente manera. Los tiempos de procesamiento de los trabajos del conjunto $J(r^1, q^1)$ son los k números enteros para el problema de SUBSET SUM ($k < n$). Definimos a B como la longitud del intervalo $[r^1 + P(J(r^1, q^2)), r^2 + \delta(\sigma)]$.

En esta sección usaremos nuevamente el calendario auxiliar $\sigma(A)$ ($A \subseteq J(r^1, q^1)$). Recuerde que en el calendario $\sigma(A)$ los trabajos del conjunto $J(r^1, q^2)$ ocupan el intervalo de tiempo $[r^1, r^1 + P(J(r^1, q^2))]$, y estos trabajos son seguidos por los trabajos del conjunto A de modo que el intervalo $[r^1 + P(J(r^1, q^2)), r^2]$ queda reservado únicamente para los trabajos del conjunto A . Debido a nuestras suposiciones, tal conjunto A existe ya que, de lo contrario, el bloque crítico en el calendario

σ comenzará en el momento r^2 y σ será óptimo según Lema 1.

Al resolver SUBSET SUM, consideramos diferentes subconjuntos posibles A , asumiendo implícitamente, que en el calendario óptimo que pretendemos crear, los trabajos del conjunto $J(r^1, q^2)$ ocupan el intervalo de tiempo $[r^1, r^1 + P(J(r^1, q^2))]$. Esta suposición es válida debido al Lema 2.

Considere el subconjunto \hat{A} del conjunto $J(r^1, q^1)$, similar al conjunto A , tal que

$$r^1 + P(J(r^1, q^2)) + P(\hat{A}) \leq r^2;$$

es decir, el último trabajo asignado del conjunto \hat{A} se completa antes o al tiempo r^2 en el calendario $\hat{\sigma}(\hat{A})$.

Como veremos más adelante, será suficiente considerar calendarios de la forma $\sigma(A)$ y $\hat{\sigma}(\hat{A})$, para todas las posibles opciones apropiadas de subconjuntos A y \hat{A} . Usamos el algoritmo de programación dinámica mencionado anteriormente para el problema de SUBSET SUM para generar estos conjuntos.

De hecho, trataremos sólo con dos posibles subconjuntos A y \hat{A} que potencialmente pueden producir una solución óptima. Llamamos a estos subconjuntos A_{opt} y \hat{A}_{opt} , respectivamente, y los definimos de la siguiente manera.

Sea $A_{opt} \subseteq J(r^1, q^1)$ la solución de la instancia derivada anterior de SUBSET SUM tal que

$$P(A_{opt}) \geq r^2 - r^1 - P(J(r^1, q^2))$$

y

$$P(A_{opt}) = \min_A P(A).$$

De manera similar, sea $\hat{A}_{opt} \subseteq J(r^1, q^1)$ la solución de SUBSET SUM tal que

$$P(\hat{A}_{opt}) \leq r^2 - r^1 - P(J(r^1, q^2))$$

y

$$P(\hat{A}_{opt}) = \max_{\hat{A}} P(\hat{A}).$$

En otras palabras, A_{opt} es el subconjunto más pequeño de tal forma que no se genera ningún hueco en el calendario $\sigma(A)$ (es decir, que produce el menor desplazamiento para los trabajos del kernel), mientras que \hat{A}_{opt} es el subconjunto más grande que deja un espacio en el calendario $\hat{\sigma}(\hat{A})$ (es decir, que produce el espacio más pequeño posible antes del tiempo r^2).

En los dos lemas siguientes mostramos que, entre todos los subconjuntos posibles A y \hat{A} , respectivamente, los subconjuntos A_{opt} y \hat{A}_{opt} , respectivamente, producen el menor *makespan* posible en los calendarios $\sigma(A)$ y $\hat{\sigma}(\hat{A})$, respectivamente.

Lema 8 Si $P(A) \geq P(A_{opt})$, entonces $C_{max}(\sigma(A)) \geq C_{max}(\sigma(A_{opt}))$, para todo $A \subseteq J(r^1, q^1)$.

Demostración. Sin pérdida de generalidad, suponga que el máximo *makespan* en el calendario $\sigma(A_{opt})$ se logra mediante un trabajo del conjunto $J(q^2)$, ya que de lo contrario $\sigma(A_{opt})$ es óptimo por el Lema 5 y por lo tanto $C_{max}(\sigma(A)) \geq C_{max}(\sigma(A_{opt}))$.

De lo contrario, dado que

$$\delta(\sigma(A_{opt})) = r^1 + P(J(r^1, q^2)) + P(A_{opt}) - r^2$$

y

$$\delta(\sigma(A)) = r^1 + P(J(r^1, q^2)) + P(A) - r^2$$

obtenemos que

$$\delta(\sigma(A)) \geq \delta(\sigma(A_{opt}))$$

Ahora, como

$$C_{max}(\sigma(A_{opt})) = r^2 + \delta(\sigma(A_{opt})) + P(J(r^2, q^2)) + q^2$$

y

$$C_{max}(\sigma(A)) = r^2 + \delta(\sigma(A)) + P(J(r^2, q^2)) + q^2,$$

obtenemos que $C_{max}(\sigma(A)) \geq C_{max}(\sigma(A_{opt}))$. □

Lema 9 Si $P(\hat{A}) \leq P(\hat{A}_{opt})$, entonces $C_{max}(\hat{\sigma}(\hat{A})) \geq C_{max}(\hat{\sigma}(\hat{A}_{opt}))$, para todo $\hat{A} \subseteq J(r^1, q^1)$.

Demostración. Sin pérdida de generalidad, suponga que el máximo *makespan* en el calendario $\hat{\sigma}(\hat{A}_{opt})$ se obtiene mediante un trabajo del conjunto $J(q^1)$, ya que de lo contrario $\hat{\sigma}(\hat{A}_{opt})$ es óptimo por Lema 6 y por lo tanto $C_{max}(\hat{\sigma}(\hat{A})) \geq C_{max}(\hat{\sigma}(\hat{A}_{opt}))$.

Dado que $P(\hat{A}) \leq P(\hat{A}_{opt})$,

$$P(J(r^1, q^1) \setminus \hat{A}) \geq P(J(r^1, q^1) \setminus \hat{A}_{opt})$$

Ahora, como

$$C_{max}(\hat{\sigma}(\hat{A}_{opt})) = r^2 + P(J(r^2)) + P(J(r^1, q^1) \setminus \hat{A}_{opt}) + q^1$$

y

$$C_{max}(\hat{\sigma}(\hat{A})) = r^2 + P(J(r^2)) + P(J(r^1, q^1) \setminus \hat{A}) + q^1$$

obtenemos que $C_{max}(\hat{\sigma}(\hat{A})) \geq C_{max}(\hat{\sigma}(\hat{A}_{opt}))$. □

Corolario 3 $C_{max}(\sigma(A_{opt})) \leq C_{max}(\sigma(A, D))$, para cualquier calendario $\sigma(A)$ y cualquier $D \subseteq J(r^1, q^2)$.

Demostración. Por el Lema 8, $C_{max}(\sigma(A_{opt})) \leq C_{max}(\sigma(A))$, para cualquier $\sigma(A)$, y por el Lema 2, $C_{max}(\sigma(A)) \leq C_{max}(\sigma(A, D))$, para cualquier $D \subseteq J(r^1, q^2)$, por lo tanto $C_{max}(\sigma(A_{opt})) \leq C_{max}(\sigma(A, D))$. □

Corolario 4 $C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}, D))$, para cualquier calendario $\hat{\sigma}(\hat{A})$ y cualquier $D \subseteq J(r^1, q^2)$.

Demostración. Por el Lema 9, $C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}))$. Al mismo tiempo, $C_{max}(\hat{\sigma}(\hat{A})) \leq C_{max}(\hat{\sigma}(\hat{A}, D))$, ya que, obviamente, el hueco antes del tiempo r^2 en el calendario $\hat{\sigma}(\hat{A}, D)$ es más larga que en el calendario $\hat{\sigma}(\hat{A})$. De estas dos desigualdades obtenemos, $C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}, D))$. \square

Teorema 1 Cualquiera de los calendarios $\sigma(A_{opt})$ o/y $\hat{\sigma}(\hat{A}_{opt})$ es óptimo.

Demostración. Primero mostramos que si $C_{max}(\sigma(A_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}_{opt}))$ entonces el calendario $\sigma(A_{opt})$ domina cualquier calendario $\sigma(A, D)$ y cualquier calendario $\hat{\sigma}(\hat{A}, D)$. De hecho, por el Corolario 3, $C_{max}(\sigma(A_{opt})) \leq C_{max}(\sigma(A, D))$, y por Corolario 4, $C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}, D))$.

De $C_{max}(\sigma(A_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}_{opt}))$, obtenemos que

$$C_{max}(\sigma(A_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}, D)).$$

Por lo tanto, $C_{max}(\sigma(A_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}, D))$ y $C_{max}(\sigma(A_{opt})) \leq C_{max}(\sigma(A, D))$, para cualquier $\sigma(A)$, $\hat{\sigma}(\hat{A})$ y $D \subseteq J(r^1, q^2)$.

De manera análoga, ahora mostramos que si $C_{max}(\sigma(A_{opt})) \geq C_{max}(\hat{\sigma}(\hat{A}_{opt}))$, entonces el calendario $\hat{\sigma}(\hat{A}_{opt})$ domina cualquier calendario $\sigma(A, D)$ y cualquier calendario $\hat{\sigma}(\hat{A}, D)$. De hecho, por el Corolario 3, $C_{max}(\sigma(A_{opt})) \leq C_{max}(\sigma(A, D))$, y por el Corolario 4, $C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}, D))$. De $C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\sigma(A_{opt}))$, obtenemos que

$$C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\sigma(A_{opt})) \leq C_{max}(\sigma(A, D)).$$

Por lo tanto, $C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\hat{\sigma}(\hat{A}, D))$, y entonces $C_{max}(\hat{\sigma}(\hat{A}_{opt})) \leq C_{max}(\sigma(A, D))$, para cualquier $\sigma(A)$, $\hat{\sigma}(\hat{A})$ y $D \subseteq J(r^1, q^2)$.

Para concluir la demostración, mostramos que $\sigma(A, D)$ o $\hat{\sigma}(A, D)$ domina cualquier otro calendario factible. Sea S cualquier calendario factible. Podemos transformar el calendario S en uno de los calendarios $\sigma(A, D)$ o $\hat{\sigma}(A, D)$ sin aumentar la amplitud. Utilizaremos un calendario $\sigma(A, D)$ cuando S no contiene huecos innecesarios antes del tiempo r^2 , caso contrario usaremos un calendario $\hat{\sigma}(A, D)$ (esto es debido a la estructura de los calendarios) De hecho, definamos los conjuntos A y D de manera similar a como antes, es decir, el conjunto A consta de los trabajos del conjunto $J(r^1, q^1)$ asignados antes de la hora r^2 en el calendario S , y D es el conjunto de trabajos del conjunto $J(r^1, q^2)$ asignados después del tiempo r^2 en el calendario S . Dado que, a diferencia del calendario S , el calendario $\sigma(A, D)$ no contiene ningún hueco evitable, $\delta(S) \geq \delta(\sigma(A, D))$ o $g(S) \geq g(\hat{\sigma}(A, D))$, donde $g(s)$ es la longitud total del o de los espacios en el calendario S antes del tiempo r^2 . Si $\delta(S) \geq \delta(\sigma(A, D))$, entonces $C_{max}(\sigma(A, D)) \leq C_{max}(S)$. Ahora, Si $g(S) \geq g(\hat{\sigma}(A, D))$, entonces $C_{max}(\hat{\sigma}(A, D)) \leq C_{max}(S)$.

Por lo tanto, mostramos que el calendario $\sigma(A, D)$ o el calendario $\hat{\sigma}(A, D)$ domina cualquier calendario factible y se demuestra el teorema. \square

Capítulo 6

Experimentos computacionales

Nuestro programa fue codificado en lenguaje C bajo el sistema operativo Linux para 32 bits en una computadora con procesador Intel Core i7-4790 (3.6 GHz) y 16 GB en RAM DDR3 para ejecutar el código.

Para verificar la eficacia práctica de nuestras condiciones de optimalidad y el rendimiento de nuestras heurísticas; los parámetros del trabajo (tiempos de liberación, tiempos de procesamiento y tiempos de entrega) se generaron aleatoriamente de la siguiente manera:

- Los tiempos de liberación y entrega r^1, r^2, q^1 y q^2 se generaron con la función `rand()` de C, con los rangos $(0, 10n)$, $(0, 20n)$, $(0, 30n)$, $(0, 50n)$ y $(0, 100n)$, donde n es el número de trabajos en cada instancia.
- Para determinar el tiempo de liberación y entrega de cada trabajo individual j , generamos aleatoriamente dos números enteros 0 ó 1 con la misma probabilidad $\frac{1}{2}$ en una prueba. En cada trabajo j establecemos $r_j = r^1$ si el primer número correspondiente es 0 y $r_j = r^2$ en caso contrario. Logramos una operación similar para determinar q_j .
- Los tiempos de procesamiento del trabajo se generaron a partir de los intervalos $[1, 10]$, $[1, 20]$, $[1, 30]$, $[1, 50]$ y $[1, 100]$.

El programa tiene un procedimiento separado para cada una de las cinco heurísticas (como se describieron en el Capítulo 2). Cada una de las cinco heurísticas (LDT, LDT-G, LDT-N, LDT-A y LDT-V) se probaron individualmente, también de forma combinada como duetos (un par de heurísticas, LDT-G combinada con las cuatro heurísticas restantes) y como quinteto (las cinco heurísticas juntas).

Para el caso de los duetos y el quinteto, la subrutina pseudo-polinomial solo es llamada cuando ninguna de las heurísticas satisface alguna de nuestras condiciones de optimalidad. Para cada una de estas 10 combinaciones, hay una subrutina de tiempo $O(n \log(n))$ que incluye los procedimientos

que implementan las heurísticas en sí y un procedimiento auxiliar que verifica las condiciones de optimalidad para los programas creados por la heurística correspondiente.

Hemos probado nuestras seis condiciones de optimalidad, primero, para un número moderado de casos de problemas. Sorprendidos con los resultados, hemos aumentado gradualmente la cantidad de instancias terminando con una cantidad extremadamente grande de instancias probadas. Hemos probado las condiciones de optimalidad para nuestras heurísticas individuales y las hemos clasificado de acuerdo con su desempeño práctico (la solución generada por las heurísticas es óptima si al menos una de las condiciones se satisface para esa solución; en la práctica, sin embargo, dos o más condiciones están simultáneamente satisfechas). Hemos combinado las heurísticas con el mejor comportamiento práctico con el resto de las cuatro heurísticas en cuatro pares diferentes (dúos) de heurísticas para verificar nuestras seis condiciones de optimalidad de forma combinada. Hemos realizado los experimentos para estos cuatro dúos y luego para las cinco heurísticas juntas (el quinteto) verificando las seis condiciones de optimalidad para las soluciones generadas por estas heurísticas.

Hemos generado 50 millones de instancias de los problemas para cada una de las 10 combinaciones anteriores para probar la eficiencia de nuestras heurísticas y la eficiencia de nuestras condiciones de optimalidad (las instancias se generaron aleatoriamente, para diferentes números de trabajos y diferentes rangos desde los cuales se generan los tiempos de procesamiento, liberación y entrega). Nuestros resultados experimentales para el quinteto resultaron ser extremadamente sólidos en la práctica: estuvo satisfecho para *todas* las 50 millones de instancias probadas. Incluso para las heurísticas individuales, al menos una de nuestras seis condiciones se cumplió en casi el 87 % para la peor de las heurísticas, y en más del 99 % para la mejor de las heurísticas; para los dúos, la eficiencia del peor par ya ha alcanzado más del 99,9 % (para cada grupo, estas estadísticas se derivaron de los correspondientes 50 millones de casos de problemas).

Así, para todos los propósitos prácticos, el problema de calendarización se puede resolver en un tiempo polinomial de bajo grado. En el caso teóricamente posible de que ninguno de los dos o cinco calendarios LDT satisfaga ninguna de las seis condiciones de optimalidad, el programa principal llama a la subrutina pseudopolinomial que deriva la instancia SUBSET SUM correspondiente e invoca el algoritmo de programación dinámica mencionado anteriormente para esa instancia. Nuestros experimentos computacionales han producido un resultado colateral práctico para el problema SUBSET SUM. Una de las versiones propuestas de la heurística LDT ha resuelto de manera óptima las instancias correspondientes del problema SUBSET SUM para más del 21 % de todas las instancias probadas, mientras que para las instancias con una gran cantidad de trabajos (750 y 1000 trabajos) esto sucedió en más del 32 % de los casos. El resultado se entregó instantáneamente para las instancias resueltas de manera óptima.

En todas las 500 millones de instancias de los problemas probadas, el tiempo de ejecución de la computadora para los 10 grupos de 50 millones de instancias de problemas varió desde 1 hora hasta un poco menos de 8 horas (estas incluyen las instancias que no se resolvieron de manera óptima por ninguna de nuestras condiciones de optimalidad, es decir, para lo cual se invocó la subrutina pseudopolinomial para el problema SUBSET SUM).

Hemos creado nuestras instancias de problema para 10, 20, 30, 50, 100, 200, 300, 500, 750 y 1000

trabajos. Para un número determinado de trabajos, consideramos cinco intervalos de tiempo, a partir de los cuales se obtuvieron los tiempos de liberación y los tiempos de entrega (como se mencionó anteriormente).

En las tablas del Apéndice, con cada línea se asocia el número de trabajos y el intervalo de tiempo correspondiente. Dado que hay cinco intervalos de tiempo para cada número de trabajos y hay 10 cantidades posibles para el número de trabajos, cada tabla contiene 50 líneas. Para cada línea (que representa una combinación particular del número de trabajos y el intervalo de tiempo correspondiente), se genera un millón de instancias del problema. No hubo una diferencia notable en las estadísticas de un millón a un número mayor de instancias para cada línea y, debido a la limitación de espacio, nuestros resultados se informan para un millón de instancias.

Así, para cada uno de los 10 escenarios descritos anteriormente (heurística individual, los dúos y el quinteto) se generaron 50 millones de instancias. Hay 10 tablas, una para cada configuración anterior (tablas 12-16 para configuraciones individuales, tablas 17-20 para los dúos y tabla 21 para el quinteto). Así que tenemos 10 grupos independientes de casos de problemas (cada grupo contiene 50 millones de casos).

En las breves Tablas 1-11 que se muestran a continuación (las Tablas detalladas 12-21 en el Apéndice, respectivamente), la primera columna de cada tabla, etiquetada con T indica, para cada grupo de un millón de instancias, el tiempo total transcurrido en resolver estas. Hay una columna separada para cada una de las seis condiciones de optimalidad; las entradas de las columnas L1, L3, L4, L5, L6 y L7 indican el porcentaje de instancias resueltas de manera óptima por el lema correspondiente (es decir, para las que el calendario formado por las heurísticas correspondientes fue óptimo debido a ese lema). Las entradas de las columnas etiquetadas con S_S representan el porcentaje de instancias que no podrían haberse resuelto usando ninguna de las condiciones de optimalidad, por lo tanto, se invocó la subrutina pseudopolinomial para SUBSET SUM. Las entradas de las columnas etiquetadas con σ_{OPT} , $\hat{\sigma}_{OPT}$, σ_{OPT}^N , σ_{OPT}^A y σ_{OPT}^V representan el porcentaje de instancias óptimas que obtuvo el calendario correspondiente.

Las columnas marcadas con $S = S_S$, donde S puede ser uno de los calendarios σ_{OPT} , $\hat{\sigma}_{OPT}$, σ_{OPT}^N , σ_{OPT}^A o σ_{OPT}^V , representan el porcentaje de las instancias para las cuales el calendario S obtuvo la solución óptima pero ninguna de nuestras condiciones de optimalidad pudo verificarlo y tuvo que ejecutarse la subrutina pseudopolinomial

En las tablas para los pares de heurísticas, la última columna representa el porcentaje de instancias para las que el calendario creado fue óptimo por al menos una de las dos heurísticas.

En las Tablas detalladas 12-21 del Apéndice, n representa el número de trabajos en cada grupo de un millón de instancias, y p_j (r_j, q_j , respectivamente) es el intervalo a partir del cual los tiempos de procesamiento (tiempos de liberación y entrega, respectivamente) se generaron para estos casos (indicamos solo el límite superior de cada intervalo ya que el límite inferior siempre es 0). En algunas columnas hay 0 en el resultado si la condición correspondiente no se aplica a las heurísticas correspondientes (por ejemplo, el Lema 6 se aplica solo a las heurísticas LDT-G, mientras que el Lema 5 no se aplica a esas heurísticas).

A continuación, en las breves Tablas 1-11, damos las estadísticas promedio para cada uno de los 10 grupos (5 heurísticas individuales, 4 duetos y un quinteto) para los 50 millones de casos de problemas correspondientes. Observamos que si se satisface el Lema 3 o el Lema 4, entonces, obviamente, el Lema 1 también se satisface; Los lemas 4 y 5, y los lemas 4 y 6 pueden satisfacerse simultáneamente, y el lema 7 y cualquiera de los lemas anteriores pueden satisfacerse simultáneamente.

T	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^A	$\sigma_{OPT}^A = S_S$
5905.916	86.960	26.262	52.279	34.722	0	15.464	3.915	96.084	0.101

Tabla 6.1: Estadísticas de la heurística LDT-A

T	L1	L3	L4	L5	L6	L7	S_S	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S_S$
3439.323	78.539	26.263	52.275	0	65.297	21.600	0.000002	99.999	0.000002

Tabla 6.2: Estadísticas de la heurística LDT-G

T	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^V	$\sigma_{OPT}^V = S_S$
26844.923	86.940	26.271	52.268	34.706	0	0.140	13.059	86.940	0.115

Tabla 6.3: Estadísticas de la heurística LDT-V

T	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}	$\sigma_{OPT} = S_S$
26967.208	86.777	26.269	52.275	34.537	0	0.158	13.212	86.787	1.119

Tabla 6.4: Estadísticas de la heurística LDT

T	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^N	$\sigma_{OPT}^N = S_S$
25068.734	86.939	26.279	52.266	34.697	0	1.672	12.055	87.944	0.103

Tabla 6.5: Estadísticas de la heurística LDT-N

T	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^A	$\sigma_{OPT}^A = S_S$	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S_S$	σ_{OPT}^A or $\hat{\sigma}_{OPT}$
4871.068	86.957	26.271	52.264	34.711	65.308	21.547	0.000076	96.214	0.000022	99.938	0.000020	99.999

Tabla 6.6: Estadísticas de las heurísticas LDT-A y LDT-G

T	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^N	$\sigma_{OPT}^N = S_S$	$\sigma_{\hat{OPT}}$	$\hat{\sigma}_{OPT} = S_S$	σ_{OPT}^N or $\hat{\sigma}_{OPT}$
13546.320	86.938	26.277	52.273	34.699	65.292	21.591	0	87.979	0	100	0	100

Tabla 6.7: Estadísticas de las heurísticas LDT-N y LDT-G

T	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}	$\sigma_{OPT} = S_S$	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S_S$	σ_{OPT} or $\hat{\sigma}_{OPT}$
4664.669	86.779	26.275	52.272	34.526	65.310	21.428	0.000694	86.832	0.000068	99.942	0.000026	99.999

Tabla 6.8: Estadísticas de las heurísticas LDT y LDT-G

T	L1	L3	L4	L5	L6	L7	S.S	σ_{OPT}^V	$\sigma_{OPT}^V = S.S$	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S.S$	σ_{OPT}^V or $\hat{\sigma}_{OPT}$
4809.631	86.944	26.279	52.268	34.719	65.276	21.428	0.000102	86.944	0.000020	99.938	0.000026	99.999

Tabla 6.9: Estadísticas de las heurísticas LDT-V y LDT-G

T	L1	L3	L4	L5	L6	L7	S.S	σ_{OPT}	$\sigma_{OPT} = S.S$
5999.562	86.986	26.260	52.279	34.748	65.295	21.600	0	86.907	0

Tabla 6.10: Estadísticas de las cinco heurísticas LDT (parte 1)

σ_{OPT}^A	$\sigma_{OPT}^A = S.S$	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S.S$	σ_{OPT}^N	$\sigma_{OPT}^N = S.S$	σ_{OPT}^V	$\sigma_{OPT}^V = S.S$	σ_{OPT}^A or $\hat{\sigma}_{OPT}$ or σ_{OPT}^N or σ_{OPT} or σ_{OPT}^V
96.194	0	99.999	0	87.937	0	86.938	0	100

Tabla 6.11: Estadísticas de las cinco heurísticas LDT (parte 2)

Capítulo 7

Conclusión y trabajos futuros

En este documento se ha realizado un estudio de un problema NP-Duro del problema de calendarización con dos tiempos de liberación y dos tiempos de entrega en una máquina $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$.

Utilizando nuestras condiciones de optimalidad, que resultaron ser extremadamente eficientes, el problema de calendarización puede resolverse en la práctica, en un tiempo casi instantáneo en una computadora personal moderada; ya que en más de 500 millones de instancias probadas, 99.999998 % se resolvieron de manera óptima, mientras que los 50 millones de casos probados en el quinteto (el décimo grupo) se resolvieron de manera óptima. Así, para todos los propósitos prácticos, el problema de calendarización obtiene una solución rápida en tiempo polinomial.

Como hemos visto, la eficiencia práctica de nuestras condiciones de optimalidad para diferentes heurísticas es diferente (aunque todas resultan ser muy fuertes). La heurística LDT resultó ser la menos eficiente entre las cinco heurísticas y (sorprendentemente) la heurística LDT-V, la más complicada entre las cinco heurísticas, resultó ser la menos eficiente entre las cuatro heurísticas restantes.

Al analizar las propiedades estructurales del problema, hemos establecido vínculos entre nuestro problema de calendarización y el problema SUBSET SUM. En particular, para un posible evento teórico en el que una instancia de nuestro problema de calendarización no cumpla con ninguna de las condiciones de optimalidad, hemos descrito cómo se puede *reducir* el problema en tiempo polinomial, a un problema SUBSET SUM mejor estudiado, que al ser resuelto, se obtiene una solución óptima para nuestro problema de calendarización, en tiempo pseudo-polinomial.

En la prueba de la dureza NP del problema de calendarización $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$, para una instancia de SUBSET SUM, se construye la instancia de calendarización correspondiente (consulte la Sección 4 en [2]). Se puede verificar fácilmente que, para este caso, no se cumplen las condiciones de optimalidad de los Lemas 1-4, lo que por supuesto, es algo que se esperaba naturalmente. Aunque los problemas de calendarización y SUBSET SUM son NP-Duro, el problema de calendarización obtiene una solución más fácil mediante nuestra subrutina de tiempo polinomial que emplea las heurísticas propuestas.

Mediante un simple análisis de nuestro estudio experimental, podemos estimar la eficiencia de las heurísticas propuestas para la solución del problema SUBSET SUM. Con este fin, observamos que las condiciones del Lema 7 del capítulo 3 vinculan el problema de calendarización $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$ con el problema SUBSET SUM.

Como podemos ver en las tablas (columna L7), la heurística LDT-G ha mostrado el mejor rendimiento práctico para el problema SUBSET SUM (se garantizó que al menos más del 21 % de las instancias se resolvieron de manera óptima, y el número correspondiente para instancias con 750 y 1000 trabajos fue 32 %).

Nuestro enfoque para el caso especial con dos tiempos de liberación y dos tiempos de entrega puede extenderse para las configuraciones con más de dos tiempos de liberación y dos tiempos de entrega (las heurísticas en sí mismas y las condiciones de optimalidad también).

Para generalizar las heurísticas propuestas y las condiciones de optimalidad para el escenario extendido, será necesario el estudio de otros sub-casos que surjan con tres o más tiempos de liberación y tiempos de entrega. El número de fases en las heurísticas extendidas dependerá del número de tiempos de liberación (recuerde que las heurísticas propuestas aquí consisten en dos fases que, aproximadamente, funcionan antes y después del tiempo r^2).

Dado un número constante k de tiempos de liberación y tiempos de entrega en la heurística extendida, en lugar de cuatro grupos de trabajos, se distinguirá el número correspondiente de los grupos (con diferentes combinaciones de tiempos de liberación y tiempos de entrega), y en lugar de uno (en nuestro caso fue r^2), pueden existir hasta $k - 1$ tiempos de liberación “críticos”.

En cuanto a nuestras condiciones de optimalidad, podemos notar por ejemplo, que:

- El Lema 2 se mantendrá para la configuración generalizada con k tiempos de liberación y tiempos de entrega.
- Los Lemas 3 y 4 no se mantendrán pero se pueden generalizar.
- El Lema 5 se mantiene y se puede establecer un lema similar para los próximos tiempos de entrega q^2, \dots, q^k .
- El lema 6, en general, no permanece verdadero pero se puede generalizar si una condición similar se cumple para conjuntos $J(r^3, q^3), \dots, J(r^k, q^k)$.
- También podemos generalizar la noción de $\delta(S)$ para tiempos de liberación r^2, \dots, r^{k-1} y luego el Lema 7 se puede generalizar respectivamente.
- Las condiciones de optimalidad extendidas podrían incorporarse en una subrutina de tiempo polinomial que, en la práctica, también puede ser eficiente.

Para las configuraciones extendidas, nuestra subrutina de tiempo pseudopolinomial es inaplicable, ya que los vínculos con el problema SUBSET SUM no son más evidentes (ya que podría ser necesario tratar con dos o más núcleos; para tal escenario, hay vínculos similares con el problema BIN PACKING, consulte [17]).

Bibliografía

- [1] P. Bratley, M. Florian and P. Robillard. On sequencing with earliest start times and due–dates with application to computing bounds for $(n/m/G/F_{max})$ problem. *Naval Res. Logist. Quart.* 20, 57–67 (1973)
- [2] E. Chinos and N. Vakhania. Adjusting scheduling model with release and due dates in production planning. *Cogent Engineering* 4(1), p. 1-23 (2017) DOI: 10.1080/23311916.2017.1321175
- [3] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP–completeness*. Freeman, San Francisco, 1979.
- [4] M.R. Garey, D.S. Johnson, B.B. Simons and R.E. Tarjan. Scheduling unit–time tasks with arbitrary release times and deadlines. *SIAM J. Comput.* 10, 256–269 (1981)
- [5] R.L. Graham, E.L. Lawler, J.L. Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discrete Math.* 5 287-326 (1979)
- [6] L.A. Hall and D.B. Shmoys. Jackson’s rule for single-machine scheduling: Making a good heuristic better, *Mathematics of Operations Research* 17 22–35 (1992)
- [7] J.R. Jackson. *Scheduling a production line to minimize the maximum tardiness*. *Management Science Research Project*, University of California, Los Angeles, CA (1955)
- [8] E. Nowicki and C. Smutnicki. An approximation algorithm for a single-machine scheduling problem with release times and delivery times. *Discrete Applied Math.* 48, 69-79, 1994.
- [9] A. A. Lazarev, and D. I. Arkhipov. Minimization of the Maximal Lateness for a Single Machine. *Automation and Remote Control* 77, pp. 656-671 (2016)
- [10] C.N. Potts. Analysis of a heuristic for one machine sequencing with release dates and delivery times. *Operations Research* 28, p.1436-1441 (1980)
- [11] L. Schrage. Obtaining optimal solutions to resource constrained network scheduling problems, unpublished manuscript (march, 1971)
- [12] N. Vakhania. A better algorithm for sequencing with release and delivery times on identical processors. *Journal of Algorithms* 48, p.273-293 (2003)
- [13] N. Vakhania. Single-Machine Scheduling with Release Times and Tails. *Annals of Operations Research*, 129, p.253-271 (2004)

- [14] N. Vakhania, F. Werner. Minimizing maximum lateness of jobs with naturally bounded job data on a single machine in polynomial time. *Theoretical Computer Science* 501, p. 72–81 doi:10.1016/j.tcs.2013.07.001 (2013).
- [15] N.Vakhania, D.Perez and L.Carballo. Theoretical Expectation versus Practical Performance of Jackson’s Heuristic. *Mathematical Problems in Engineering* Volume 2015, Article ID 484671, 10 pages <http://dx.doi.org/10.1155/2015/484671> (2015)
- [16] N.Vakhania. Fast solution of single-machine scheduling problem with embedded jobs. *Theoretical Computer Science* <https://doi.org/10.1016/j.tcs.2019.03.001> (2019)
- [17] N. Vakhania. Dynamic Restructuring Framework for Scheduling with Release Times and Due-Dates. *Mathematics* 7(11), 1104 (2019) <https://doi.org/10.3390/math7111104>

7.1. Appendix

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^A	$\sigma_{OPT}^A = S_S$
10	10	100	3.434	87.740	25.713	54.353	33.719	0	9.202	7.448	92.551	0.645
20	10	200	5.879	87.127	25.695	53.525	33.798	0	13.066	5.154	94.845	0.082
30	10	300	7.724	86.835	25.591	53.237	33.730	0	15.967	3.473	96.526	0.020
50	10	500	10.793	86.605	25.588	52.903	33.838	0	19.217	1.559	98.440	0.001
100	10	1000	19.689	86.429	25.582	52.703	33.861	0	21.423	0.260	99.739	0
200	10	2000	43.607	86.372	25.628	52.688	33.717	0	21.731	0.016	99.983	0
300	10	3000	69.471	86.331	25.708	52.385	33.963	0	21.953	0.001	99.998	0
500	10	5000	118.474	86.233	25.775	52.220	33.975	0	22.032	0	100	0
750	10	7500	190.588	86.498	25.794	52.650	33.864	0	21.575	0	100	0
1000	10	10000	293.159	86.642	25.725	53.037	33.608	0	21.249	0	100	0
10	20	200	6.484	87.317	24.217	55.842	31.669	0	5.252	9.799	90.200	0.834
20	20	400	9.984	86.938	24.382	55.115	31.897	0	8.056	8.124	91.875	0.114
30	20	600	16.718	86.754	24.471	54.810	31.988	0	10.690	6.505	93.494	0.024
50	20	1000	22.639	86.654	24.484	54.629	32.062	0	14.524	4.090	95.910	0.002
100	20	2000	33.591	86.577	24.536	54.528	32.037	0	18.856	1.350	98.649	0.0003
200	20	4000	52.403	86.658	24.438	54.851	31.782	0	20.449	0.183	99.816	0
300	20	6000	85.884	86.424	24.611	54.276	32.089	0	21.080	0.033	99.966	0
500	20	10000	129.976	86.801	24.664	54.901	31.979	0	20.442	0.0009	99.999	0
750	20	15000	201.689	87.089	24.475	55.635	31.464	0	19.898	0	100	0
1000	20	20000	293.964	86.036	26.515	50.741	35.349	0	22.750	0	100	0
10	30	300	5.156	87.184	23.689	56.372	30.954	0	3.768	10.687	89.312	0.893
20	30	600	10.327	86.858	23.923	55.658	31.263	0	5.797	9.537	90.462	0.125
30	30	900	16.546	86.785	23.998	55.490	31.363	0	7.904	8.160	91.839	0.026
50	30	1500	24.358	86.621	24.144	55.098	31.599	0	11.579	5.929	94.070	0.002
100	30	3000	36.341	86.547	24.247	54.904	31.618	0	16.699	2.685	97.314	0
200	30	6000	59.121	86.661	24.253	55.022	31.660	0	19.797	0.593	99.406	0
300	30	9000	83.775	86.329	24.609	54.037	32.311	0	21.113	0.160	99.839	0
500	30	15000	142.115	87.174	24.176	56.240	30.905	0	19.566	0.016	99.983	0
750	30	22500	216.392	86.601	26.599	51.743	34.824	0	21.660	0.001	99.998	0
1000	30	30000	276.840	87.195	23.596	57.012	30.257	0	19.396	0.0001	99.999	0
10	50	500	5.890	86.980	23.276	56.674	30.448	0	2.447	11.592	88.407	0.970
20	50	1000	12.843	86.824	23.615	56.080	30.859	0	3.699	10.855	89.144	0.126
30	50	1500	17.530	86.683	23.758	55.719	30.956	0	5.297	9.908	90.091	0.031
50	50	2500	30.404	86.716	23.602	55.963	30.776	0	8.061	8.039	91.960	0.003
100	50	5000	84.479	86.621	23.941	55.406	31.292	0	13.184	4.821	95.178	0
200	50	10000	154.145	86.973	24.041	56.007	30.932	0	17.249	1.746	98.253	0
300	50	15000	241.015	87.213	23.878	56.668	30.606	0	18.378	0.710	99.290	0
500	50	25000	173.441	87.268	25.761	54.062	33.251	0	19.985	0.123	99.876	0
750	50	37500	202.455	86.145	27.062	50.160	35.796	0	22.745	0.021	99.978	0
1000	50	50000	266.012	86.226	35.091	37.344	48.862	0	27.558	0.003	99.996	0
10	100	1000	18.218	86.964	23.086	56.940	30.069	0	1.449	12.155	87.844	1.001
20	100	2000	39.450	86.870	23.352	56.548	30.358	0	1.940	11.899	88.100	0.140
30	100	3000	63.230	86.716	23.561	56.024	30.716	0	2.874	11.423	88.576	0.034
50	100	5000	95.728	86.603	23.577	55.882	30.794	0	4.669	10.347	89.652	0.003
100	100	10000	150.771	87.014	23.700	56.431	30.551	0	8.205	7.602	92.397	0
200	100	20000	290.574	86.272	25.629	52.415	33.760	0	14.374	4.726	95.274	0
300	100	30000	344.039	87.293	23.126	57.819	29.465	0	15.082	2.644	97.355	0
500	100	50000	582.116	86.219	34.693	37.861	48.465	0	25.268	1.076	98.923	0
750	100	75000	355.710	89.752	49.396	17.968	71.747	0	31.822	0.246	99.753	0
1000	100	100000	290.745	94.630	62.145	5.360	89.263	0	32.209	0.043	99.956	0

Tabla 7.1: Estadísticas para la heurística LDT-A

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	S.S	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S.S$
10	10	100	1.547	80.048	25.750	54.298	0	67.208	21.393	0	99.999	0
20	10	200	3.249	79.184	25.671	53.513	0	66.826	21.545	0	100	0
30	10	300	5.312	78.862	25.607	53.254	0	66.746	21.608	0	100	0
50	10	500	8.890	78.584	25.680	52.904	0	66.576	21.700	0	100	0
100	10	1000	19.608	78.286	25.686	52.599	0	66.255	21.858	0	100	0
200	10	2000	42.466	78.392	25.659	52.733	0	66.409	21.679	0	100	0
300	10	3000	66.401	78.138	25.712	52.425	0	66.195	21.910	0	100	0
500	10	5000	118.864	78.102	25.783	52.319	0	66.049	21.929	0	100	0
750	10	7500	189.905	78.504	25.680	52.823	0	66.137	21.512	0	100	0
1000	10	10000	238.459	78.844	25.716	53.128	0	66.501	21.169	0	100	0
10	20	200	1.688	80.010	24.237	55.773	0	68.195	20.715	0	100	0
20	20	400	3.078	79.492	24.437	55.054	0	68.172	20.865	0	100	0
30	20	600	4.812	79.335	24.462	54.872	0	68.197	20.903	0	100	0
50	20	1000	9.780	79.115	24.508	54.607	0	68.089	21.029	0	100	0
100	20	2000	19.296	79.111	24.535	54.575	0	68.105	20.966	0	100	0
200	20	4000	40.856	79.223	24.406	54.817	0	68.252	20.815	0	100	0
300	20	6000	67.808	78.918	24.606	54.312	0	67.804	21.106	0	100	0
500	20	10000	118.476	79.459	24.616	54.842	0	68.050	20.554	0	100	0
750	20	15000	188.190	80.076	24.290	55.785	0	68.492	19.931	0	100	0
1000	20	20000	250.498	77.320	26.591	50.728	0	64.656	22.686	0	100	0
10	30	300	1.860	80.020	23.726	56.293	0	68.601	20.469	0	100	0
20	30	600	3.828	79.567	24.022	55.545	0	68.657	20.674	0	100	0
30	30	900	6.218	79.492	23.981	55.511	0	68.703	20.665	0	100	0
50	30	1500	9.452	79.193	24.187	55.005	0	68.477	20.901	0	100	0
100	30	3000	19.843	79.140	24.262	54.878	0	68.449	20.907	0	100	0
200	30	6000	43.512	79.210	24.227	54.983	0	68.418	20.815	0	100	0
300	30	9000	66.105	78.749	24.719	54.029	0	67.749	21.268	0	100	0
500	30	15000	120.523	80.291	24.161	56.129	0	69.098	19.719	0	100	0
750	30	22500	176.363	78.111	26.387	51.723	0	65.251	21.894	0	100	0
1000	30	30000	242.170	80.575	23.637	56.937	0	69.816	19.429	0	100	0
10	50	500	1.469	80.005	23.351	56.653	0	68.874	20.285	0	100	0
20	50	1000	3.141	79.639	23.599	56.039	0	69.019	20.509	0	100	0
30	50	1500	5.015	79.566	23.730	55.835	0	69.007	20.530	0	100	0
50	50	2500	9.124	79.589	23.654	55.935	0	69.232	20.471	0	100	0
100	50	5000	19.014	79.376	23.931	55.444	0	68.828	20.652	0	100	0
200	50	10000	43.982	79.961	23.915	56.045	0	68.988	20.053	0	100	0
300	50	15000	68.089	80.503	23.828	56.675	0	69.430	19.505	0	100	0
500	50	25000	115.508	79.807	25.726	54.081	0	66.780	20.197	0	100	0
750	50	37500	170.145	77.133	26.895	50.238	0	63.950	22.869	0	100	0
1000	50	50000	234.624	72.474	35.120	37.354	0	51.065	27.529	0	100	0
10	100	1000	1.484	80.034	23.022	57.011	0	69.039	20.113	0	100	0
20	100	2000	3.125	79.914	23.259	56.654	0	69.476	20.158	0	100	0
30	100	3000	4.781	79.591	23.550	56.041	0	69.235	20.457	0	100	0
50	100	5000	8.405	79.534	23.658	55.876	0	69.148	20.493	0	100	0
100	100	10000	18.592	80.135	23.696	56.438	0	69.361	19.876	0	100	0
200	100	20000	39.701	77.921	25.649	52.272	0	66.199	22.086	0	100	0
300	100	30000	62.823	80.879	23.212	57.666	0	70.477	19.124	0	100	0
500	100	50000	111.977	72.649	34.790	37.858	0	51.651	27.354	0	100	0
750	100	75000	181.269	67.407	49.534	17.872	0	28.214	32.594	0	100	0
1000	100	100000	247.998	67.471	62.087	5.384	0	10.763	32.530	0	100	0

Tabla 7.2: Estadísticas para la heurística LDT-G

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^N	$\sigma_{OPT}^N = S_S$
10	10	100	3.968	87.593	25.717	54.323	33.593	0	3.952	10.855	89.144	0.843
20	10	200	8.641	86.922	25.608	53.476	33.627	0	3.871	11.060	88.939	0.098
30	10	300	13.419	86.719	25.635	53.174	33.625	0	3.851	11.070	88.929	0.018
50	10	500	25.662	86.558	25.705	52.857	33.696	0	3.823	11.103	88.896	0.001
100	10	1000	104.649	86.411	25.653	52.654	33.827	0	3.870	11.133	88.866	0
200	10	2000	256.864	86.360	25.629	52.712	33.609	0	3.811	11.165	88.834	0
300	10	3000	221.063	86.348	25.703	52.475	33.860	0	3.891	11.109	88.890	0
500	10	5000	378.090	86.304	25.787	52.291	34.049	0	3.882	11.163	88.836	0
750	10	7500	644.659	86.449	25.767	52.698	33.796	0	3.797	11.040	88.959	0
1000	10	10000	1107.158	86.690	25.767	53.085	33.623	0	3.710	10.863	89.136	0
10	20	200	13.624	87.227	24.218	55.841	31.556	0	2.046	11.941	88.058	0.888
20	20	400	28.060	86.788	24.415	55.070	31.837	0	1.911	12.170	87.830	0.111
30	20	600	45.091	86.732	24.391	54.886	31.931	0	1.932	12.132	87.867	0.022
50	20	1000	69.479	86.598	24.511	54.555	32.058	0	1.960	12.182	87.817	0.002
100	20	2000	132.835	86.545	24.542	54.534	31.969	0	1.936	12.198	87.802	0
200	20	4000	259.841	86.710	24.485	54.867	31.766	0	1.919	12.010	87.989	0
300	20	6000	444.204	86.508	24.614	54.295	32.245	0	1.975	12.168	87.831	0
500	20	10000	896.798	86.781	24.561	54.902	31.872	0	1.921	11.925	88.074	0
750	20	15000	1755.051	87.151	24.555	55.588	31.508	0	1.838	11.601	88.398	0
1000	20	20000	1601.076	86.108	26.632	50.716	35.321	0	2.082	12.541	87.458	0
10	30	300	5.593	87.077	23.708	56.362	30.865	0	1.369	12.353	87.646	0.915
20	30	600	10.968	86.790	23.953	55.652	31.202	0	1.303	12.490	87.509	0.106
30	30	900	17.280	86.764	23.955	55.551	31.280	0	1.279	12.479	87.520	0.021
50	30	1500	44.935	86.618	24.122	55.143	31.530	0	1.297	12.569	87.430	0.002
100	30	3000	204.579	86.493	24.191	54.922	31.592	0	1.305	12.649	87.350	0
200	30	6000	261.561	86.552	24.311	54.888	31.695	0	1.310	12.566	87.433	0
300	30	9000	390.738	86.359	24.667	54.039	32.236	0	1.336	12.752	87.247	0
500	30	15000	828.130	87.202	24.191	56.217	30.979	0	1.236	11.951	88.048	0
750	30	22500	1862.168	86.568	26.488	51.756	34.759	0	1.354	12.552	87.447	0
1000	30	30000	1830.126	87.139	23.708	56.815	30.355	0	1.234	12.009	87.990	0
10	50	500	6.437	87.073	23.362	56.759	30.400	0	0.863	12.561	87.439	0.901
20	50	1000	13.374	86.788	23.696	56.037	30.752	0	0.762	12.790	87.209	0.111
30	50	1500	20.874	86.675	23.641	55.714	31.150	0	0.782	12.858	87.141	0.023
50	50	2500	36.934	86.698	23.696	55.958	30.743	0	0.782	12.805	87.194	0.003
100	50	5000	82.339	86.577	23.967	55.348	31.165	0	0.791	12.906	87.093	0
200	50	10000	195.673	86.919	23.971	55.997	30.886	0	0.766	12.568	87.431	0
300	50	15000	346.351	87.199	23.901	56.622	30.527	0	0.748	12.295	87.704	0
500	50	25000	745.152	87.328	25.861	54.047	33.332	0	0.745	12.185	87.815	0
750	50	37500	1580.734	86.117	27.047	50.151	35.971	0	0.872	13.320	86.680	0
1000	50	50000	2485.359	86.294	35.139	37.285	48.950	0	1.020	13.157	86.842	0
10	100	1000	7.328	86.908	23.063	56.910	30.036	0	0.465	12.890	87.109	0.945
20	100	2000	15.499	86.789	23.302	56.533	30.314	0	0.387	12.994	87.005	0.106
30	100	3000	24.264	86.736	23.611	56.096	30.618	0	0.391	13.026	86.973	0.022
50	100	5000	44.247	86.672	23.674	55.895	30.772	0	0.394	13.073	86.926	0.003
100	100	10000	101.821	87.036	23.683	56.499	30.541	0	0.381	12.711	87.288	0
200	100	20000	286.840	86.190	25.612	52.330	33.825	0	0.424	13.527	86.472	0
300	100	30000	498.309	87.262	23.241	57.657	29.519	0	0.363	12.484	87.515	0
500	100	50000	1242.805	86.223	34.754	37.836	48.383	0	0.514	13.496	86.503	0
750	100	75000	1961.536	89.745	49.386	17.905	71.863	0	0.599	10.035	89.964	0
1000	100	100000	1906.555	94.635	62.133	5.387	89.264	0	0.546	5.246	94.753	0

Tabla 7.3: Estadísticas para la heurística LDT-N

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}	$\sigma_{OPT} = S_S$
10	10	100	4.890	87.027	25.675	54.381	32.969	0	1.798	12.776	87.223	2.386
20	10	200	11.947	86.480	25.564	53.502	33.231	0	0.724	13.519	86.480	2.261
30	10	300	17.877	86.421	25.623	53.216	33.374	0	0.472	13.578	86.421	2.293
50	10	500	57.637	86.354	25.585	52.956	33.446	0	0.286	13.645	86.354	2.429
100	10	1000	172.423	86.308	25.624	52.676	33.716	0	0.144	13.691	86.308	2.507
200	10	2000	196.999	86.328	25.608	52.757	33.521	0	0.068	13.671	86.328	2.490
300	10	3000	271.657	86.271	25.684	52.450	33.879	0	0.048	13.728	86.271	2.535
500	10	5000	490.137	86.273	25.857	52.256	33.930	0	0.029	13.726	86.273	2.549
750	10	7500	849.629	86.448	25.746	52.690	33.774	0	0.021	13.551	86.448	2.510
1000	10	10000	1992.472	86.727	25.741	53.179	33.607	0	0.014	13.272	86.727	2.442
10	20	200	12.718	86.771	24.278	55.773	31.119	0	0.933	13.115	86.885	1.665
20	20	400	26.295	86.456	24.369	55.103	31.529	0	0.369	13.543	86.456	1.226
30	20	600	40.700	86.457	24.425	54.887	31.629	0	0.242	13.542	86.457	1.222
50	20	1000	62.715	86.410	24.465	54.659	31.884	0	0.151	13.589	86.410	1.265
100	20	2000	133.537	86.499	24.438	54.632	31.873	0	0.074	13.500	86.499	1.279
200	20	4000	270.310	86.561	24.473	54.816	31.725	0	0.034	13.438	86.561	1.290
300	20	6000	628.769	86.473	24.735	54.240	32.198	0	0.022	13.526	86.473	1.302
500	20	10000	1052.755	86.778	24.654	54.889	31.884	0	0.015	13.221	86.778	1.269
750	20	15000	1165.357	87.116	24.513	55.602	31.503	0	0.008	12.883	87.116	1.254
1000	20	20000	2089.887	85.991	26.582	50.678	35.209	0	0.008	14.008	85.991	1.364
10	30	300	14.968	86.672	23.758	56.345	30.473	0	0.630	13.246	86.753	1.444
20	30	600	28.482	86.462	23.930	55.723	30.804	0	0.255	13.537	86.462	0.838
30	30	900	45.637	86.459	24.040	55.462	31.044	0	0.166	13.540	86.459	0.823
50	30	1500	85.291	86.419	24.193	55.079	31.283	0	0.095	13.580	86.419	0.841
100	30	3000	167.395	86.461	24.216	54.944	31.592	0	0.048	13.538	86.461	0.874
200	30	6000	343.679	86.543	24.296	54.934	31.639	0	0.023	13.456	86.543	0.878
300	30	9000	549.900	86.281	24.619	54.070	32.270	0	0.017	13.718	86.281	0.894
500	30	15000	907.666	87.133	24.138	56.202	30.980	0	0.008	12.866	87.133	0.859
750	30	22500	1324.198	86.552	26.500	51.730	34.792	0	0.005	13.448	86.552	0.869
1000	30	30000	1846.779	87.130	23.680	56.891	30.216	0	0.005	12.869	87.130	0.833
10	50	500	7.218	86.567	23.310	56.715	29.891	0	0.393	13.373	86.626	1.225
20	50	1000	14.874	86.371	23.583	56.050	30.440	0	0.151	13.628	86.371	0.540
30	50	1500	23.076	86.408	23.743	55.752	30.763	0	0.097	13.591	86.408	0.500
50	50	2500	40.451	86.559	23.726	55.887	30.645	0	0.056	13.440	86.559	0.517
100	50	5000	89.853	86.504	23.931	55.436	31.112	0	0.031	13.495	86.504	0.528
200	50	10000	211.501	86.842	23.985	55.969	30.851	0	0.016	13.157	86.842	0.507
300	50	15000	370.522	87.147	23.841	56.682	30.497	0	0.007	12.852	87.147	0.501
500	50	25000	793.882	87.244	25.819	54.012	33.229	0	0.005	12.755	87.244	0.496
750	50	37500	1659.573	86.119	27.044	50.160	35.947	0	0.004	13.880	86.119	0.557
1000	50	50000	2606.335	86.228	35.077	37.344	48.834	0	0.003	13.771	86.228	0.573
10	100	1000	7.796	86.489	23.063	56.952	29.556	0	0.235	13.462	86.537	1.051
20	100	2000	16.499	86.416	23.288	56.531	29.865	0	0.073	13.583	86.416	0.323
30	100	3000	25.873	86.430	23.597	56.048	30.354	0	0.053	13.569	86.430	0.266
50	100	5000	46.919	86.490	23.724	55.836	30.618	0	0.031	13.509	86.490	0.266
100	100	10000	107.898	86.897	23.714	56.407	30.600	0	0.015	13.102	86.897	0.252
200	100	20000	302.526	86.132	25.588	52.318	33.777	0	0.008	13.867	86.132	0.278
300	100	30000	520.902	87.232	23.145	57.764	29.434	0	0.004	12.767	87.232	0.258
500	100	50000	1292.223	86.228	34.790	37.888	48.320	0	0.003	13.771	86.228	0.277
750	100	75000	2041.686	89.693	49.357	17.932	71.792	0	0.002	10.306	89.693	0.216
1000	100	100000	1924.894	94.606	62.129	5.363	89.219	0	0.001	5.393	94.606	0.123

Tabla 7.4: Estadísticas para la heurística LDT

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^V	$\sigma_{OPT}^V = S_S$
10	10	100	6.577	87.639	25.736	54.386	33.659	0	1.449	12.360	87.639	0.953
20	10	200	17.421	86.929	25.664	53.494	33.711	0	0.723	13.070	86.929	0.120
30	10	300	36.013	86.757	25.678	53.205	33.642	0	0.476	13.242	86.757	0.025
50	10	500	75.961	86.586	25.644	52.919	33.765	0	0.283	13.413	86.586	0.003
100	10	1000	163.833	86.359	25.675	52.576	33.798	0	0.146	13.640	86.359	0
200	10	2000	158.704	86.402	25.625	52.754	33.657	0	0.067	13.597	86.402	0
300	10	3000	254.595	86.340	25.759	52.411	33.915	0	0.046	13.659	86.340	0
500	10	5000	466.812	86.224	25.736	52.272	33.996	0	0.028	13.775	86.224	0
750	10	7500	806.944	86.442	25.703	52.755	33.723	0	0.019	13.558	86.442	0
1000	10	10000	1848.575	86.755	25.718	53.231	33.447	0	0.013	13.244	86.755	0
10	20	200	12.983	87.244	24.174	55.784	31.643	0	0.736	12.755	87.244	0.998
20	20	400	25.671	86.863	24.401	55.083	31.896	0	0.368	13.136	86.863	0.128
30	20	600	39.138	86.748	24.418	54.886	31.873	0	0.236	13.251	86.748	0.025
50	20	1000	61.527	86.544	24.450	54.573	32.080	0	0.149	13.455	86.544	0.003
100	20	2000	140.318	86.536	24.458	54.641	31.926	0	0.073	13.463	86.536	0
200	20	4000	292.808	86.590	24.442	54.836	31.747	0	0.036	13.409	86.590	0
300	20	6000	531.417	86.471	24.688	54.192	32.204	0	0.025	13.528	86.471	0
500	20	10000	1114.735	86.903	24.610	54.997	31.994	0	0.015	13.096	86.903	0
750	20	15000	1267.788	86.989	24.597	55.394	31.549	0	0.008	13.010	86.989	0
1000	20	20000	2028.220	86.043	26.520	50.738	35.356	0	0.007	13.956	86.043	0
10	30	300	9.781	87.235	23.673	56.385	31.025	0	0.487	12.764	87.235	0.990
20	30	600	22.201	86.846	23.952	55.712	31.252	0	0.245	13.153	86.846	0.131
30	30	900	33.560	86.679	24.035	55.405	31.258	0	0.165	13.320	86.679	0.028
50	30	1500	57.762	86.602	24.021	55.219	31.414	0	0.100	13.397	86.602	0.004
100	30	3000	160.536	86.566	24.304	54.905	31.638	0	0.049	13.433	86.566	0
200	30	6000	393.019	86.588	24.246	54.959	31.619	0	0.023	13.411	86.588	0
300	30	9000	561.227	86.374	24.675	54.034	32.325	0	0.017	13.625	86.374	0
500	30	15000	938.420	87.161	24.183	56.173	30.968	0	0.010	12.838	87.161	0
750	30	22500	1365.893	86.542	26.485	51.677	34.893	0	0.006	13.457	86.542	0
1000	30	30000	1799.110	87.172	23.692	56.875	30.305	0	0.004	12.827	87.172	0
10	50	500	7.265	87.124	23.326	56.708	30.524	0	0.295	12.875	87.124	0.995
20	50	1000	15.092	86.727	23.720	55.874	30.840	0	0.148	13.272	86.727	0.137
30	50	1500	23.764	86.625	23.754	55.711	30.952	0	0.099	13.374	86.625	0.032
50	50	2500	41.201	86.725	23.699	55.953	30.733	0	0.056	13.274	86.725	0.003
100	50	5000	91.400	86.619	23.963	55.412	31.171	0	0.030	13.380	86.619	0
200	50	10000	213.579	86.938	23.990	55.965	30.985	0	0.015	13.061	86.938	0
300	50	15000	370.943	87.240	23.803	56.770	30.559	0	0.010	12.759	87.240	0
500	50	25000	781.024	87.323	25.871	53.996	33.324	0	0.004	12.676	87.323	0
750	50	37500	1615.653	86.273	27.375	50.087	36.034	0	0.002	13.726	86.273	0
1000	50	50000	2641.802	86.279	35.031	37.419	48.868	0	0.003	13.720	86.279	0
10	100	1000	7.812	87.025	23.149	56.856	30.107	0	0.145	12.974	87.025	1.010
20	100	2000	16.545	86.793	23.360	56.541	30.266	0	0.069	13.206	86.793	0.139
30	100	3000	26.373	86.693	23.549	56.064	30.642	0	0.046	13.306	86.693	0.031
50	100	5000	47.482	86.645	23.577	55.922	30.843	0	0.031	13.354	86.645	0.003
100	100	10000	108.758	86.928	23.710	56.350	30.506	0	0.013	13.071	86.928	0
200	100	20000	303.369	86.171	25.566	52.385	33.867	0	0.008	13.828	86.171	0
300	100	30000	523.948	87.262	23.156	57.731	29.503	0	0.004	12.738	87.262	0
500	100	50000	1300.785	86.263	34.728	37.927	48.304	0	0.003	13.736	86.263	0
750	100	75000	2086.309	89.632	49.125	17.871	71.775	0	0.001	10.367	89.632	0
1000	100	100000	1930.272	94.594	62.150	5.385	89.216	0	0.001	5.405	94.594	0

Tabla 7.5: Estadísticas para la heurística LDT-V

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^A	$\sigma_{OPT}^A = S_S$	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S_S$
10	10	100	1.843	87.720	25.706	54.329	33.711	67.142	21.430	0	93.490	0	100	0
20	10	200	3.750	87.137	25.793	53.417	33.816	66.852	21.510	0	94.851	0	100	0
30	10	300	5.879	86.861	25.652	53.231	33.734	66.805	21.507	0.001	96.571	0	99.936	0
50	10	500	11.889	86.590	25.511	53.018	33.716	66.635	21.767	0	98.433	0	100	0
100	10	1000	25.951	86.429	25.674	52.618	33.792	66.383	21.852	0	99.747	0	100	0
200	10	2000	55.635	86.366	25.673	52.678	33.655	66.440	21.720	0	99.984	0	99.995	0
300	10	3000	92.588	86.289	25.749	52.332	33.882	66.175	21.964	0	99.998	0	99.999	0
500	10	5000	170.247	86.274	25.787	52.285	33.937	66.095	21.955	0	100	0	100	0
750	10	7500	246.921	86.498	25.700	52.701	33.768	66.256	21.616	0	100	0	100	0
1000	10	10000	333.072	86.661	25.646	53.208	33.458	66.560	21.159	0	100	0	100	0
10	20	200	2.718	87.325	24.234	55.809	31.694	68.169	20.681	0	91.439	0	100	0
20	20	400	4.937	86.913	24.459	55.025	31.893	68.250	20.874	0	91.886	0	99.995	0
30	20	600	7.984	86.830	24.448	54.867	32.010	68.177	20.940	0	93.503	0	100	0
50	20	1000	12.874	86.623	24.534	54.550	32.121	68.035	21.057	0	95.885	0	100	0
100	20	2000	29.545	86.592	24.522	54.530	32.041	68.047	21.020	0	98.674	0	100	0
200	20	4000	60.074	86.624	24.374	54.936	31.695	68.345	20.701	0	99.812	0	99.947	0
300	20	6000	95.541	86.490	24.631	54.268	32.296	67.731	21.124	0	99.965	0	99.991	0
500	20	10000	160.458	86.767	24.722	54.795	31.884	68.133	20.496	0	99.997	0	100	0
750	20	15000	243.265	87.124	24.501	55.604	31.509	68.501	19.902	0	100	0	100	0
1000	20	20000	343.868	86.073	26.586	50.715	35.323	64.686	22.707	0	100	0	100	0
10	30	300	2.593	87.141	23.774	56.283	30.908	68.633	20.429	0	90.580	0	100	0
20	30	600	5.172	86.856	23.961	55.620	31.293	68.676	20.652	0	90.466	0	99.994	0
30	30	900	7.515	86.810	24.053	55.526	31.291	68.777	20.585	0	91.867	0	100	0
50	30	1500	14.186	86.669	24.153	55.085	31.576	68.511	20.858	0	94.096	0	100	0
100	30	3000	29.889	86.559	24.221	54.962	31.621	68.435	20.864	0	97.341	0	100	0
200	30	6000	61.090	86.589	24.340	54.868	31.673	68.356	20.719	0	99.400	0	99.846	0
300	30	9000	95.946	86.291	24.557	54.037	32.408	67.612	21.400	0	99.846	0	99.947	0
500	30	15000	159.755	87.179	24.249	56.100	30.952	69.059	19.658	0	99.984	0	99.995	0
750	30	22500	251.265	86.541	26.520	51.711	34.839	65.169	21.774	0	99.998	0	100	0
1000	30	30000	334.212	87.130	23.670	56.830	30.290	69.715	19.504	0	99.999	0	99.999	0
10	50	500	2.500	87.056	23.284	56.758	30.427	68.821	20.254	0	89.896	0	100	0
20	50	1000	4.609	86.830	23.630	56.040	30.842	68.985	20.458	0	89.169	0	99.989	0
30	50	1500	7.906	86.704	23.726	55.787	30.934	69.034	20.589	0	90.097	0	100	0
50	50	2500	14.045	86.724	23.693	55.928	30.800	69.223	20.436	0	91.967	0	100	0
100	50	5000	28.186	86.563	23.899	55.382	31.146	68.882	20.745	0	95.170	0	99.999	0
200	50	10000	59.746	86.930	23.970	55.986	30.916	69.098	19.683	0	98.228	0	99.581	0
300	50	15000	92.212	87.194	23.829	56.694	30.496	69.514	19.305	0	99.305	0	99.706	0
500	50	25000	161.615	87.301	25.795	54.075	33.194	66.811	20.127	0	99.883	0	99.980	0
750	50	37500	256.217	86.141	26.993	50.199	35.983	64.020	22.811	0	99.981	0	100	0
1000	50	50000	363.116	86.284	35.041	37.360	48.944	51.060	27.600	0	99.996	0	99.997	0
10	100	1000	2.453	86.936	23.031	56.961	30.082	68.981	20.156	0	89.275	0	100	0
20	100	2000	5.062	86.777	23.259	56.542	30.339	69.394	20.234	0	88.006	0	99.986	0
30	100	3000	7.828	86.727	23.549	56.072	30.651	69.252	20.428	0	88.589	0	100	0
50	100	5000	13.171	86.675	23.726	55.826	30.775	69.209	20.477	0	89.713	0	100	0
100	100	10000	28.592	86.982	23.700	56.386	30.612	69.395	19.924	0	92.350	0	99.999	0
200	100	20000	62.417	86.152	25.520	52.418	33.792	66.212	21.181	0	95.251	0	99.288	0
300	100	30000	94.104	87.270	23.211	57.733	29.445	70.559	17.882	0	97.383	0	98.886	0
500	100	50000	173.879	86.304	34.817	37.820	48.397	51.608	27.365	0	98.920	0	99.999	0
750	100	75000	280.303	89.756	49.368	17.896	71.781	28.223	32.737	0	99.745	0	100	0
1000	100	100000	342.446	94.579	62.092	5.379	89.225	10.777	32.514	0	99.954	0	99.872	0

Tabla 7.6: Estadísticas para las heurísticas LDT-A and LDT-G

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^N	$\sigma_{OPT}^N = S_S$	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S_S$
10	10	100	10.046	87.626	25.729	54.396	33.575	67.143	21.314	0	89.458	0	100	0
20	10	200	22.521	86.899	25.712	53.372	33.632	66.884	21.639	0	88.920	0	100	0
30	10	300	29.940	86.730	25.644	53.204	33.657	66.743	21.650	0	88.916	0	100	0
50	10	500	46.895	86.553	25.660	52.925	33.721	66.561	21.704	0	88.901	0	100	0
100	10	1000	95.079	86.439	25.639	52.673	33.793	66.362	21.836	0	88.938	0	100	0
200	10	2000	190.167	86.406	25.575	52.749	33.713	66.363	21.748	0	88.902	0	100	0
300	10	3000	291.952	86.334	25.747	52.378	33.976	66.074	21.922	0	88.852	0	100	0
500	10	5000	468.367	86.253	25.805	52.278	34.019	66.013	21.946	0	88.801	0	100	0
750	10	7500	655.387	86.550	25.836	52.718	33.911	66.110	21.464	0	89.062	0	100	0
1000	10	10000	881.396	86.733	25.775	53.131	33.623	66.392	21.106	0	89.176	0	100	0
10	20	200	10.590	87.279	24.303	55.788	31.641	68.150	20.649	0	88.447	0	100	0
20	20	400	18.389	86.816	24.470	55.022	31.811	68.240	20.868	0	87.868	0	100	0
30	20	600	26.399	86.717	24.491	54.811	31.940	68.192	20.965	0	87.854	0	100	0
50	20	1000	45.244	86.538	24.436	54.590	31.987	68.120	21.132	0	87.769	0	100	0
100	20	2000	89.052	86.568	24.440	54.664	31.986	68.079	20.975	0	87.833	0	100	0
200	20	4000	177.116	86.614	24.355	54.926	31.775	68.262	20.751	0	87.889	0	100	0
300	20	6000	274.459	86.502	24.641	54.345	32.166	67.861	21.037	0	87.809	0	100	0
500	20	10000	421.897	86.822	24.691	54.824	31.938	68.075	20.499	0	88.085	0	100	0
750	20	15000	598.981	87.141	24.558	55.613	31.481	68.528	19.837	0	88.388	0	100	0
1000	20	20000	891.741	86.047	26.545	50.763	35.312	64.697	22.697	0	87.406	0	100	0
10	30	300	10.061	87.102	23.738	56.259	30.955	68.532	20.495	0	88.017	0	100	0
20	30	600	17.939	86.814	23.982	55.642	31.257	68.659	20.621	0	87.516	0	100	0
30	30	900	25.922	86.725	24.055	55.480	31.253	68.748	20.629	0	87.504	0	100	0
50	30	1500	42.455	86.630	24.096	55.178	31.497	68.555	20.827	0	87.451	0	100	0
100	30	3000	83.716	86.543	24.185	55.021	31.552	68.488	20.840	0	87.386	0	100	0
200	30	6000	167.983	86.550	24.240	54.963	31.597	68.428	20.819	0	87.413	0	100	0
300	30	9000	257.905	86.361	24.653	54.076	32.288	67.727	21.287	0	87.258	0	100	0
500	30	15000	394.199	87.150	24.160	56.178	30.918	69.091	19.671	0	87.977	0	100	0
750	30	22500	668.485	86.594	26.468	51.776	34.919	65.086	21.760	0	87.464	0	100	0
1000	30	30000	841.008	87.139	23.725	56.839	30.171	69.834	19.440	0	87.964	0	100	0
10	50	500	10.126	86.999	23.351	56.721	30.328	68.882	20.213	0	87.723	0	100	0
20	50	1000	18.357	86.751	23.564	56.065	30.771	69.000	20.518	0	87.201	0	100	0
30	50	1500	27.214	86.684	23.717	55.800	30.964	68.965	20.583	0	87.153	0	100	0
50	50	2500	44.241	86.714	23.668	55.994	30.783	69.214	20.395	0	87.197	0	100	0
100	50	5000	88.566	86.543	23.958	55.335	31.208	68.808	20.735	0	87.064	0	100	0
200	50	10000	162.614	86.935	23.988	55.964	30.975	69.036	20.062	0	87.448	0	100	0
300	50	15000	238.656	87.194	23.972	56.574	30.533	69.477	19.462	0	87.701	0	100	0
500	50	25000	398.250	87.322	25.805	54.074	33.263	66.743	20.125	0	87.824	0	100	0
750	50	37500	656.487	86.187	27.034	50.145	36.015	63.989	22.823	0	86.749	0	100	0
1000	50	50000	1024.197	86.247	35.078	37.317	48.903	51.099	27.608	0	86.806	0	100	0
10	100	1000	9.843	86.891	23.048	56.930	29.982	69.034	20.173	0	87.468	0	100	0
20	100	2000	17.421	86.803	23.409	56.455	30.361	69.317	20.210	0	87.029	0	100	0
30	100	3000	25.389	86.641	23.523	56.068	30.613	69.250	20.460	0	86.875	0	100	0
50	100	5000	41.279	86.628	23.657	55.808	30.799	69.163	20.564	0	86.876	0	100	0
100	100	10000	79.368	86.995	23.641	56.462	30.565	69.436	19.911	0	87.251	0	100	0
200	100	20000	171.489	86.292	25.588	52.441	33.914	66.090	21.978	0	86.572	0	100	0
300	100	30000	232.719	87.268	23.274	57.662	29.549	70.454	19.067	0	87.513	0	100	0
500	100	50000	500.826	86.271	34.742	37.915	48.391	51.612	27.347	0	86.563	0	100	0
750	100	75000	876.347	89.737	49.382	17.915	71.744	28.258	32.704	0	89.956	0	100	0
1000	100	100000	1167.641	94.628	62.106	5.404	89.222	10.779	32.491	0	94.742	0	100	0

Tabla 7.7: Estadísticas para las heurísticas LDT-N y LDT-G

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}	σ_{OPT}	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT}$
10	10	100	2.091	86.939	25.617	54.364	32.983	67.111	21.490	0	87.920	0	99.999	0
20	10	200	4.638	86.508	25.669	53.410	33.238	66.876	21.636	0	86.508	0	100	0
30	10	300	6.968	86.493	25.612	53.275	33.321	66.794	21.424	0.004	86.493	0.001	99.943	0
50	10	500	12.124	86.384	25.682	52.939	33.403	66.670	21.663	0	86.384	0	100	0
100	10	1000	26.137	86.264	25.660	52.604	33.692	66.351	21.874	0	86.264	0	100	0
200	10	2000	55.043	86.315	25.550	52.781	33.600	66.424	21.724	0	86.315	0	99.996	0
300	10	3000	90.807	86.216	25.722	52.383	33.895	66.118	21.940	0	86.216	0	100	0
500	10	5000	162.826	86.284	25.745	52.300	33.963	66.042	21.983	0	86.284	0	100	0
750	10	7500	233.453	86.513	25.746	52.802	33.771	66.231	21.469	0	86.513	0	100	0
1000	10	10000	317.354	86.684	25.806	53.053	33.582	66.420	21.152	0	86.684	0	100	0
10	20	200	2.359	86.752	24.180	55.865	31.117	68.169	20.702	0	87.334	0	100	0
20	20	400	4.609	86.507	24.384	55.094	31.477	68.225	20.893	0	86.507	0	99.999	0
30	20	600	7.218	86.455	24.443	54.886	31.622	68.214	20.907	0	86.455	0	100	0
50	20	1000	12.468	86.438	24.489	54.617	31.881	68.041	21.039	0	86.438	0	100	0
100	20	2000	27.077	86.499	24.556	54.555	31.908	68.050	20.960	0	86.499	0	99.999	0
200	20	4000	56.730	86.584	24.431	54.855	31.766	68.213	20.620	0	86.584	0	99.958	0
300	20	6000	91.447	86.461	24.590	54.390	32.180	67.809	21.018	0	86.461	0	99.993	0
500	20	10000	151.537	86.799	24.633	54.950	31.858	68.137	20.431	0	86.799	0	100	0
750	20	15000	234.328	87.047	24.518	55.587	31.489	68.508	19.902	0	87.047	0	100	0
1000	20	20000	323.947	86.050	26.630	50.697	35.288	64.707	22.677	0	86.050	0	100	0
10	30	300	2.641	86.686	23.791	56.305	30.457	68.564	20.398	0	87.154	0	100	0
20	30	600	4.718	86.447	24.025	55.579	30.803	68.752	20.639	0	86.447	0	99.997	0
30	30	900	7.281	86.426	23.964	55.486	30.969	68.765	20.716	0	86.426	0	100	0
50	30	1500	13.218	86.444	24.154	55.138	31.311	68.542	20.808	0	86.444	0	100	0
100	30	3000	26.857	86.458	24.277	54.923	31.458	68.477	20.846	0	86.458	0	99.999	0
200	30	6000	57.684	86.417	24.189	54.948	31.453	68.518	20.465	0.002	86.417	0	99.855	0
300	30	9000	89.682	86.274	24.631	54.026	32.235	67.745	21.258	0	86.274	0	99.966	0
500	30	15000	152.506	87.123	24.238	56.107	30.987	69.005	19.665	0	87.123	0	100	0
750	30	22500	236.531	86.498	26.484	51.735	34.789	65.201	21.786	0	86.498	0	100	0
1000	30	30000	314.744	87.122	23.738	56.825	30.221	69.772	19.440	0	87.122	0	100	0
10	50	500	2.485	86.605	23.374	56.707	29.975	68.834	20.219	0	86.964	0	99.999	0
20	50	1000	4.749	86.416	23.634	56.033	30.354	69.091	20.475	0	86.416	0	99.997	0
30	50	1500	7.219	86.390	23.768	55.726	30.701	68.955	20.223	0.008	86.390	0	99.863	0
50	50	2500	12.733	86.545	23.677	55.963	30.589	69.231	20.418	0	86.545	0	100	0
100	50	5000	27.780	86.496	23.999	55.331	31.165	68.749	20.696	0	86.496	0	100	0
200	50	10000	59.027	86.818	23.819	56.080	30.865	69.099	18.862	0.003	86.818	0	99.571	0
300	50	15000	89.338	87.163	23.866	56.625	30.468	69.506	19.063	0.001	87.163	0	99.848	0
500	50	25000	152.646	87.244	25.824	54.039	33.091	66.892	20.141	0	87.244	0	100	0
750	50	37500	243.562	86.134	27.045	50.185	35.907	64.078	22.752	0	86.134	0	99.993	0
1000	50	50000	342.180	86.358	35.065	37.411	48.917	51.065	27.522	0	86.358	0	99.997	0
10	100	1000	2.531	86.584	23.148	56.955	29.627	69.010	20.046	0	86.864	0	99.999	0
20	100	2000	4.750	86.433	23.343	56.480	29.886	69.481	20.245	0	86.433	0	99.997	0
30	100	3000	7.374	86.442	23.589	56.077	30.382	69.219	20.383	0	86.442	0	100	0
50	100	5000	12.312	86.506	23.705	55.871	30.618	69.159	20.455	0	86.506	0	100	0
100	100	10000	26.420	86.870	23.700	56.411	30.423	69.480	19.902	0	86.870	0	99.998	0
200	100	20000	58.418	86.110	25.573	52.360	33.817	66.127	19.382	0.008	86.110	0	98.990	0
300	100	30000	87.807	87.212	23.142	57.747	29.442	70.530	16.488	0.004	87.212	0	99.138	0
500	100	50000	164.598	86.156	34.699	37.886	48.271	51.693	27.414	0	86.156	0	99.998	0
750	100	75000	267.310	89.746	49.457	17.819	71.861	28.100	32.726	0	89.746	0	99.999	0
1000	100	100000	362.408	94.618	62.146	5.404	89.242	10.735	32.450	0	94.618	0	99.999	0

Tabla 7.8: Estadísticas para las heurísticas LDT y LDT-G

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	S_S	σ_{OPT}^V	$\sigma_{OPT}^V = S_S$	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S_S$
10	10	100	2.406	87.619	25.749	54.322	33.637	67.046	21.382	0	87.619	0	99.999	0
20	10	200	4.421	86.949	25.701	53.527	33.631	66.836	21.499	0	86.949	0	100	0
30	10	300	6.828	86.728	25.669	53.229	33.626	66.750	21.412	0.002	86.728	0	99.926	0
50	10	500	11.780	86.573	25.603	52.954	33.805	66.471	21.726	0	86.573	0	100	0
100	10	1000	26.575	86.375	25.666	52.592	33.858	66.290	21.884	0	86.375	0	100	0
200	10	2000	56.059	86.356	25.643	52.684	33.636	66.442	21.731	0	86.356	0	99.996	0
300	10	3000	92.993	86.313	25.826	52.343	33.900	66.151	21.876	0	86.313	0	100	0
500	10	5000	166.482	86.336	25.878	52.249	34.067	65.960	21.901	0	86.336	0	100	0
750	10	7500	243.875	86.416	25.722	52.629	33.827	66.192	21.667	0	86.416	0	100	0
1000	10	10000	328.728	86.736	25.702	53.149	33.612	66.402	21.164	0	86.736	0	100	0
10	20	200	2.547	87.347	24.279	55.846	31.653	68.199	20.606	0	87.347	0	100	0
20	20	400	4.859	86.835	24.466	55.066	31.825	68.216	20.826	0	86.835	0	99.998	0
30	20	600	8.202	86.740	24.462	54.894	31.918	68.178	20.911	0	86.740	0	100	0
50	20	1000	12.562	86.579	24.489	54.552	32.110	67.999	21.104	0	86.579	0	100	0
100	20	2000	28.482	86.554	24.481	54.618	31.904	68.160	20.978	0	86.554	0	99.999	0
200	20	4000	60.340	86.690	24.377	54.950	31.735	68.304	20.580	0	86.690	0	99.957	0
300	20	6000	93.228	86.545	24.686	54.298	32.241	67.781	21.017	0	86.545	0	99.992	0
500	20	10000	159.818	86.849	24.660	54.886	32.001	68.011	20.465	0	86.849	0	100	0
750	20	15000	242.733	87.095	24.511	55.596	31.496	68.511	19.900	0	87.095	0	100	0
1000	20	20000	332.822	86.061	26.593	50.700	35.366	64.642	22.713	0	86.061	0	100	0
10	30	300	2.594	87.197	23.753	56.285	31.006	68.542	20.451	0	87.197	0	100	0
20	30	600	4.796	86.811	23.982	55.634	31.354	68.546	20.632	0	86.811	0	99.997	0
30	30	900	7.671	86.671	24.053	55.381	31.311	68.687	20.738	0	86.671	0	100	0
50	30	1500	13.218	86.576	24.142	55.096	31.506	68.548	20.862	0	86.576	0	100	0
100	30	3000	27.545	86.534	24.238	54.905	31.593	68.448	20.906	0	86.534	0	100	0
200	30	6000	60.075	86.499	24.357	54.826	31.676	68.348	20.395	0	86.499	0	99.845	0
300	30	9000	92.931	86.325	24.636	54.014	32.313	67.704	21.229	0	86.325	0	99.955	0
500	30	15000	158.239	87.196	24.178	56.199	31.034	68.975	19.631	0	87.196	0	100	0
750	30	22500	246.296	86.558	26.446	51.759	34.801	65.207	21.801	0	86.558	0	100	0
1000	30	30000	324.526	87.181	23.658	56.926	30.267	69.737	19.419	0	87.181	0	100	0
10	50	500	2.343	87.120	23.294	56.774	30.439	68.894	20.230	0	87.120	0	100	0
20	50	1000	4.922	86.765	23.704	56.044	30.677	69.097	20.377	0	86.765	0	99.993	0
30	50	1500	7.765	86.670	23.794	55.753	30.952	68.972	20.561	0	86.670	0	100	0
50	50	2500	13.483	86.743	23.650	56.040	30.793	69.206	20.372	0	86.743	0	100	0
100	50	5000	27.795	86.596	23.998	55.335	31.257	68.761	20.697	0	86.596	0	100	0
200	50	10000	59.590	86.910	23.995	55.954	30.920	69.093	18.802	0	86.910	0	99.572	0
300	50	15000	93.056	87.191	23.851	56.679	30.530	69.477	18.946	0	87.191	0	99.820	0
500	50	25000	157.334	87.274	25.719	54.079	33.244	66.760	20.207	0	87.274	0	100	0
750	50	37500	255.420	86.144	26.996	50.244	35.956	64.048	22.763	0	86.144	0	100	0
1000	50	50000	356.945	86.237	35.058	37.353	48.862	51.141	27.587	0	86.237	0	99.999	0
10	100	1000	2.547	87.077	23.093	56.909	30.260	68.894	20.148	0	87.077	0	100	0
20	100	2000	4.921	86.737	23.343	56.489	30.269	69.411	20.231	0	86.737	0	99.995	0
30	100	3000	8.047	86.676	23.497	56.089	30.677	69.184	20.466	0	86.676	0	100	0
50	100	5000	13.717	86.699	23.677	55.866	30.803	69.158	20.487	0	86.699	0	100	0
100	100	10000	28.154	87.028	23.709	56.458	30.650	69.348	19.847	0	87.028	0	100	0
200	100	20000	60.403	86.179	25.539	52.364	33.878	66.125	18.549	0.001	86.179	0	98.668	0
300	100	30000	89.400	87.263	23.166	57.680	29.549	70.454	17.528	0	87.263	0	99.456	0
500	100	50000	172.270	86.256	34.750	37.872	48.423	51.580	27.379	0	86.256	0	99.999	0
750	100	75000	274.997	89.747	49.413	17.905	71.855	28.147	32.378	0	89.747	0	99.797	0
1000	100	100000	352.892	94.638	62.110	5.393	89.246	10.756	32.428	0	94.638	0	99.942	0

Tabla 7.9: Estadísticas para las heurísticas LDT-V y LDT-G

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	SS	σ_{OPT}	$\sigma_{OPT=SS}$	σ_{OPT}^A	$\sigma_{OPT=SS}^A$	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT=SS}$	σ_{OPT}^N	$\sigma_{OPT=SS}^N$	σ_{OPT}^V	$\sigma_{OPT=SS}^V$
10	10	100	2.984	87.960	25.627	54.432	33.987	67.022	21.386	0	87.110	0	93.542	0	99.999	0	89.103	0	87.619	0
20	10	200	6.695	87.112	25.622	53.501	33.854	66.818	21.593	0	86.863	0	94.489	0	99.999	0	88.960	0	86.894	0
30	10	300	10.406	86.833	25.571	53.231	33.776	66.769	21.672	0	86.604	0	96.512	0	100	0	88.926	0	86.690	0
50	10	500	17.077	86.645	25.621	52.915	33.757	66.595	21.758	0	86.375	0	98.447	0	100	0	88.935	0	86.570	0
100	10	1000	37.433	86.419	25.678	52.636	33.805	66.378	21.830	0	86.280	0	99.736	0	100	0	88.834	0	86.384	0
200	10	2000	77.417	86.393	25.632	52.685	33.624	66.464	21.749	0	86.334	0	99.986	0	100	0	88.873	0	86.375	0
300	10	3000	134.546	86.268	25.747	52.349	33.850	66.211	21.956	0	87.094	0	99.998	0	100	0	88.772	0	86.257	0
500	10	5000	209.893	86.275	25.721	52.336	33.995	66.041	21.969	0	87.208	0	99.999	0	100	0	88.860	0	86.267	0
750	10	7500	320.322	86.523	25.740	52.707	33.877	66.150	21.569	0	87.933	0	100	0	100	0	89.022	0	86.518	0
1000	10	10000	424.440	86.704	25.731	53.123	33.476	66.541	21.159	0	87.039	0	100	0	100	0	89.167	0	86.701	0
10	20	200	3.750	87.533	24.246	55.802	31.988	68.047	20.690	0	86.806	0	91.501	0	100	0	88.038	0	87.337	0
20	20	400	7.328	86.919	24.341	55.128	31.969	68.213	20.884	0	86.544	0	91.579	0	100	0	87.864	0	86.787	0
30	20	600	11.155	86.820	24.439	54.848	32.026	68.175	20.962	0	86.502	0	93.494	0	100	0	87.894	0	86.727	0
50	20	1000	19.874	86.737	24.613	54.517	32.191	67.962	21.014	0	86.502	0	95.915	0	100	0	87.901	0	86.690	0
100	20	2000	40.419	86.527	24.469	54.598	31.991	68.097	21.002	0	86.414	0	98.655	0	100	0	87.772	0	86.510	0
200	20	4000	80.901	86.672	24.395	54.924	31.798	68.245	20.715	0	86.633	0	99.823	0	100	0	87.949	0	86.664	0
300	20	6000	126.413	86.499	24.544	54.371	32.143	67.886	21.108	0	86.597	0	99.965	0	100	0	87.804	0	86.493	0
500	20	10000	205.987	86.830	24.609	54.923	31.893	68.123	20.482	0	87.200	0	99.998	0	100	0	88.106	0	86.827	0
750	20	15000	310.760	87.119	24.559	55.593	31.532	68.478	19.856	0	87.218	0	99.999	0	100	0	88.362	0	87.117	0
1000	20	20000	440.471	86.058	26.565	50.681	35.327	64.682	22.761	0	86.250	0	100	0	100	0	87.414	0	86.056	0
10	30	300	3.718	87.392	23.736	56.400	31.033	68.681	20.350	0	86.714	0	90.666	0	100	0	87.665	0	87.248	0
20	30	600	7.265	86.829	23.888	55.615	31.289	68.729	20.731	0	86.413	0	90.186	0	100	0	87.450	0	86.718	0
30	30	900	11.093	86.754	23.935	55.503	31.342	68.740	20.729	0	86.436	0	91.803	0	100	0	87.466	0	86.676	0
50	30	1500	19.109	86.627	24.090	55.100	31.605	68.483	20.908	0	86.406	0	94.070	0	100	0	87.413	0	86.583	0
100	30	3000	39.153	86.551	24.185	54.964	31.642	68.412	20.900	0	86.437	0	97.359	0	100	0	87.399	0	86.536	0
200	30	6000	79.870	86.562	24.258	54.948	31.654	68.380	20.816	0	86.520	0	99.399	0	100	0	87.426	0	86.555	0

Tabla 7.10: Estadísticas para las cinco heurísticas LDT(parte 1)

n	p_j	r_j, q_j	seconds	L1	L3	L4	L5	L6	L7	SS	σ_{OPT}	$\sigma_{OPT} = S.S$	σ_{OPT}^A	$\sigma_{OPT}^A = S.S$	$\hat{\sigma}_{OPT}$	$\hat{\sigma}_{OPT} = S.S$	σ_{OPT}^N	$\sigma_{OPT}^N = S.S$	σ_{OPT}^V	$\sigma_{OPT}^V = S.S$
300	30	9000	124.070	86.344	24.658	53.970	32.349	67.669	21.386	0	86.320	0	99.855	0	100	0	87.243	0	86.340	0
500	30	15000	202.127	87.110	24.180	56.104	31.016	68.995	19.724	0	87.342	0	99.985	0	100	0	87.951	0	87.108	0
750	30	22500	319.556	86.557	26.498	51.767	34.774	65.234	21.739	0	86.616	0	99.998	0	100	0	87.445	0	86.556	0
1000	30	30000	417.238	87.106	23.667	56.855	30.248	69.754	19.481	0	87.220	0	99.999	0	100	0	87.957	0	87.105	0
10	50	500	3.781	87.169	23.353	56.668	30.594	68.835	20.273	0	86.539	0	89.847	0	100	0	87.267	0	87.066	0
20	50	1000	7.109	86.819	23.598	56.100	30.784	69.083	20.450	0	86.415	0	88.983	0	100	0	87.169	0	86.724	0
30	50	1500	11.437	86.764	23.735	55.743	31.022	68.979	20.621	0	86.435	0	90.136	0	100	0	87.165	0	86.696	0
50	50	2500	19.170	86.727	23.710	55.921	30.809	69.222	20.427	0	86.518	0	91.986	0	100	0	87.186	0	86.694	0
100	50	5000	39.435	86.573	23.973	55.323	31.331	68.695	20.735	0	86.454	0	95.164	0	100	0	87.075	0	86.560	0
200	50	10000	80.620	86.878	23.925	55.958	31.001	69.013	20.131	0	86.829	0	98.234	0	100	0	87.398	0	86.873	0
300	50	15000	119.898	87.273	23.928	56.652	30.590	69.421	19.427	0	87.248	0	99.313	0	100	0	87.772	0	87.271	0
500	50	25000	205.158	87.339	25.814	54.150	33.248	66.759	20.039	0	87.390	0	99.886	0	100	0	87.832	0	87.338	0
750	50	37500	332.675	86.166	27.025	50.175	35.917	64.087	22.803	0	86.279	0	99.979	0	100	0	86.712	0	86.164	0
1000	50	50000	417.160	86.273	35.107	37.321	48.933	51.071	27.574	0	86.339	0	99.996	0	100	0	86.833	0	86.272	0
10	100	1000	2.844	87.148	23.002	57.014	30.223	69.007	20.132	0	86.543	0	89.327	0	100	0	87.093	0	87.071	0
20	100	2000	5.515	86.945	23.374	56.547	30.302	69.475	20.153	0	86.541	0	88.072	0	100	0	87.071	0	86.866	0
30	100	3000	8.359	86.744	23.537	56.053	30.705	69.218	20.462	0	86.431	0	88.563	0	100	0	86.928	0	86.687	0
50	100	5000	14.061	86.678	23.622	55.896	30.883	69.109	20.511	0	86.475	0	89.695	0	100	0	86.898	0	86.647	0
100	100	10000	28.483	86.973	23.666	56.457	30.532	69.475	19.890	0	86.872	0	92.340	0	100	0	87.218	0	86.963	0
200	100	20000	61.011	86.205	25.601	52.401	33.867	66.141	22.006	0	86.139	0	95.197	0	100	0	86.473	0	86.202	0
300	100	30000	87.494	87.283	23.176	57.788	29.477	70.529	19.039	0	87.284	0	97.361	0	100	0	87.541	0	87.281	0
500	100	50000	172.536	86.284	34.772	37.900	48.378	51.626	27.330	0	86.286	0	98.932	0	100	0	86.570	0	86.283	0
750	100	75000	286.231	89.755	49.417	17.955	71.800	28.203	32.628	0	89.784	0	99.750	0	100	0	89.968	0	89.754	0
1000	100	100000	385.115	94.603	62.107	5.404	89.182	10.819	32.490	0	94.598	0	99.960	0	100	0	94.719	0	94.602	0

Tabla 7.11: Estadísticas para las cinco heurísticas LDT (parte 2)



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS



Control Escolar de Licenciatura

VOTOS DE APROBATORIOS

Secretaría Ejecutiva del Instituto de Investigación en Ciencias Básicas Aplicadas de la Universidad Autónoma del Estado de Morelos.

P r e s e n t e .

Por medio de la presente le informamos que después de revisar la versión escrita de la tesis que realizó el C. **REYNOSO GÓMEZ ALEJANDRO** con número de matrícula **20154011922** cuyo título es:

“CASOS SOLUBLES EN TIEMPO POLINOMIAL PARA EL PROBLEMA $1|r_i \in \{r^1, r^2\}, q_i \in \{q^1, q^2\}|C_{max}$ Y LA REDUCCIÓN AL PROBLEMA SUBSET SUM”

Consideramos que **SI** reúne los méritos que son necesarios para continuar los trámites para obtener el título de **LICENCIADO EN CIENCIAS ÁREA TERMINAL EN CIENCIAS COMPUTACIONALES Y COMPUTACIÓN CIENTÍFICA.**

Cuernavaca, Mor a 12 de mayo de 2022

Atentamente
Por una universidad culta

Se adiciona página con la e-firma UAEM de los siguientes:

Dr. Crispin Zavala Díaz
Dr. Federico Alonso Pecina
Dr. Nodari Vakhania Maisuradze
Dr. Jose Alberto Hernandez Aguilar
Dra. Larissa Sbitneva

Presidente:
Secretario:
Vocal:
Suplente:
Suplente:



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

LARISSA VIACHESLAVOVNA SBITNEVA | Fecha:2022-05-12 13:14:31 | Firmante

pgbXM6VnuSkGxiQt5FVWYuuZNSfUVrpVYtLaW1oX6iVd7NTrc8m6bFGCb4iDRKK5VEckZu5hgQsPvXjgO6sodogYDeKPSSduGedX3+UUAU0KIWeDZ3noR3gMSQE0WK+lu/C7QJWm7rleF1iGRJluBOgybh2ocbJAQlyVIZRxyfRT+a2F6r9TqqVuUUKZp1KebGJBucbkXXHCtveFYD/ZyEY4hHjzbzWj8hSw9227dbt66hijInSCW3LnaEbaGhRZHoB3lu2wcl6MkENd92bflRb7ZbGZhr74RfYxnhrwc2+Sx632TJoFlcVCpl6d7jopF3c9E5KPxCRRUORs6lQ==

JOSE ALBERTO HERNANDEZ AGUILAR | Fecha:2022-05-12 15:22:33 | Firmante

Eks3x+Kg78vefKl3gxSSIPyQ1Bsi8fePxGNkGviO6r7SYNJQNPTn30kF9jGn8JIB+D+walpzN0gXOIFpdKfoluwvh1Hlj0BHVMI9d3k+37h7lirXhSPo3hEvG/qzTipJ27BUz/hz5dyAJFb6y/SQabaUXgyzrwQJTWclwAlZu39r4rBEM3yptL8Fe4QuHcskl4MvVUYesi5FBtySHldcxEk1sfJlJfVTRjpRNj3jz4BzLuSLPz6ZY ZpFudWkmD6AJLds0x7qjmaHE+j1bKXW8BSTLnnoZd/fgZcG9DaeKuQIVHEQ9O1+PPHrLso1JBiqjO/YSwJ6lMGUADxN4QRJg==

JOSE CRISPIN ZAVALA DIAZ | Fecha:2022-05-12 20:46:27 | Firmante

NoBlul5hOXwaOS9o45dVT+UQYWjo3qYBaOFld0EYZ5m+8tfSq3MIUKFwyWFC8ab6AAFWHOE4swJCITc63fgwqkRB6DauKbnPPTkPaCOHW7I65Yt8VuzmODVv4mYFYu3snBEOE45juAC3tfAxBk6+eECcYJ/qmk07nGPuwjqWra2WgyClrzyvBzHrEvAEj2pOaClfmv0mWfqGVVlFTMxNC1fUhuqsnz4+NSIG8LV7EstauO9+1vt/2a/PQo9xzagulgwcZ+Cfkrm+et2sQsKx+BA8zoWln22qtaWPxrlq20zEVDVpkwzngRqGPKveNQ1ZFyKJvOJbGQn2SUG/Dui9Q==

FEDERICO ALONSO PECINA | Fecha:2022-05-12 20:49:10 | Firmante

eiL6jl/2Zyonei5U4YE330XxRrSRtpc4jaxyiRgbcRrldLjtT8Cn4FDnYoLmqQJQ+XmOXpawW/vndDKt8VjG4tyMVxVAIDUrBfYI4w/ZHalknfCUwiQAEjfb6A4/1QvZQ8F7/vzJh2AiHr0yL/CdbYSVOgu3yku3oxYrk3E+4qhFHabYz2CzJklVJrPBtlNGwAD1bcniAv1Sbeho3yw7Vyywr/qWGsMzgoXoEHMXI2Rp/zBIOwLQyHHkXA04Lp/u18FwxHr+dJO0bDFr28+FotA VLOJK3bS6QfNyLbgEbG0TgNaudUxRdz0Pnk0cG0THuclRv65QLLoOwilrWtQ==

NODARI VAKHANIA MAISURADZE | Fecha:2022-05-13 19:20:17 | Firmante

p0uScgLNvg+4YwedzVSk7cXVOgOh5nxufHQTdn0TF3CFcF+EkYhhPUqNdRqJ5ex1zBUMjXvdEUMHH6ZW6u7Pr6V6PptM8T2rlbDbYCeYnjsn0ImxjhD95E0EviF967vu+pY/onOGbOowG0tqO9iYnZKLn8Mpr53Bk3uJLxu4wQ+MsX+UCET5GUx9p6WoO/q3kgbUwPLldlMle13996O9FZJpEycwzPtVd02hGvnpUUKWCT8CwUQXZ9omiTf7Y42lJLOstutkYD WlwbglMLZf26Y09t1kTf749/5wShiluZQBmeAtDjzw9ID22n7ObXlHi0+f67xMwhDBdoOilw==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:



R1JbQWIPs

<https://efirma.uaem.mx/noRepudio/r7nr8zQJlF8HAelknoBOjopoir1CSJ>

