

ASIGNACION TODO O NADA A REDES DE TRANSPORTE  
PUBLICO: APLICACION AL CASO DE SANTIAGO

J. de Cea            y    J. P. Bunster  
Departamento de Ingeniería de Transporte  
Pontificia Universidad Católica de Chile

Resumen

El aspecto más crítico de un modelo de asignación "todo o nada" a redes de transporte público de gran tamaño, lo constituye el tiempo de cálculo requerido para determinar árboles de rutas mínimas. En redes como la de Santiago, este aspecto se hace más crítico aún debido al alto porcentaje de arcos por los que circula un elevado número de líneas comunes.

El objetivo principal de la investigación que ha dado origen a este trabajo es explorar diferentes vías de simplificación de los modelos disponibles que impliquen ahorros importantes de tiempo de cálculo sin que se produzca un deterioro significativo de la calidad de los resultados. Considerando como base de comparación la asignación obtenida con el algoritmo de cálculo de rutas mínimas del modelo MADITUC, sin restricción del número de líneas comunes y del número de transbordos de una ruta de transporte público, han sido analizadas las siguientes simplificaciones:

- a) Limitación del número de líneas comunes a considerar en el cálculo de rutas mínimas, asignando viajes a todas las líneas comunes existentes en la etapa de carga de la red.
- b) Limitación del número de transbordos en una ruta de transporte público.

La experimentación se llevó a cabo usando la red estratégica de Santiago en 1978, sobre la que fueron codificados 272 recorridos de locomoción colectiva.

## 1. Introducción

El análisis de redes de transporte público requiere, como etapa previa, una decisión respecto al nivel de detalle con que las redes deben ser codificadas. Dicho nivel de detalle debe, en primer lugar, ser cohe-rente con el grado de desagregación de la zonificación del área servida por el sistema que se desea estudiar. En segundo término, dado que entre las posibles fuentes de error de los modelos de asignación destaca, como la más importante, el problema de agregación espacial (ver De Cea y Chapleau, 1983; Chapleau y De Cea, 1983), parece razonable pensar que tanto la zonificación como la codificación deberían ser hechas con el máximo de talle que sea posible. Se podría pensar en una situación límite en la que las zonas representen manzanas poblacionales y los nodos de la red paraderos de los recorridos de locomoción colectiva.

Evidentemente esta tentación detallista choca en la práctica, entre otras cosas, con los recursos computacionales (memoria y tiempo de procesamiento) requeridos por un modelo de asignación de viajes a redes de transporte público. En el caso de los modelos de tipo "todo o nada" (que son los más usados) el aspecto más crítico, en cuanto a tiempo de procesamiento se relaciona con los algoritmos de cálculo de rutas de costo generalizado mínimo. Estos algoritmos, que se basan en la construcción de árboles, son bastante más lentos que los equivalentes empleados para el análisis de redes de automóviles, básicamente por la existencia de transbordos y de líneas comunes.

La selección de líneas comunes entre dos nodos de la red, no necesariamente consecutivos, es tratada de variadas formas en los modelos actualmente en uso. Chriqui y Robillard (1975) definen explícitamente el conjunto de líneas comunes entre dos nodos como el subconjunto de las líneas alternativas que minimiza el tiempo esperado de viaje de los usuarios. Como se verá en la sección siguiente, bajo ciertas hipótesis apropiadas la selección de las líneas comunes se reduce a un problema de programación hiperbólica (0,1). Aún cuando para su solución existe un algoritmo eficiente, los tiempos de cálculo de rutas mínimas en grandes redes resultan ser excesivos, por lo que posteriormente se han propuesto numerosas simplificaciones al modelo de Chriqui y Robillard.

La contradicción existente entre nivel de detalle (zonificación y codificación) y complejidad de los algoritmos de cálculo de rutas mínimas, presente en el análisis de cualquier red, se hace aún más fuerte cuando se desea estudiar redes de transporte público como la de Santiago. Por un lado, la cantidad de líneas existentes en el sistema (del orden de 350 en 1985) hace que aun esquematizando su codificación se alcance rápidamente la cifra de 1.000 nodos. Por otro, aunque el problema de selección de líneas comunes es conceptualmente idéntico al que se presenta en redes de sistemas urbanos de países desarrollados, su dimensión en redes de países en desarrollo lo transforma en un aspecto de particular interés.

Todo lo anterior sugiere la necesidad de explorar modificaciones a los algoritmos existentes de determinación de rutas mínimas que tienda a reducir el tiempo de cálculo, sin producir un deterioro importante en la calidad de los resultados obtenidos. Entre las mejoras posibles a los modelos de asignación "todo o nada" actualmente en uso, en este trabajo de



investigación nos hemos concentrado en las dos ideas que siguen:

- limitación del número de líneas comunes a considerar en el cálculo de rutas mínimas, pero asignando viajes a todas las líneas comunes existentes en la etapa de carga de la red
- limitación del número de transbordos en una ruta de transporte público.

Este documento ha sido organizado como sigue. En primer lugar se describirá el problema del tratamiento de líneas comunes en algoritmos de cálculo de rutas mínimas en redes de transporte público, se discutirán los distintos planteamientos existentes y se propondrán modificaciones posibles. Luego se discute, muy brevemente, el efecto que tiene en el cálculo de rutas mínimas la limitación del número de transbordos de un viaje. Las mejoras propuestas son testeadas usando la red estratégica de Santiago de 1978, sobre la que fueron codificadas 272 líneas de transporte público. Para efectos de asignación, se consideró una matriz origen-destino de viajes en transporte público obtenida de una encuesta realizada en Santiago en 1977 (ver DICTUC, 1978). Es necesario destacar que en el análisis experimental fueron utilizados datos reales a fin de contar con una red que presentara los problemas que se quería analizar (especialmente un alto número de líneas con existencia de arcos con elevados números de líneas comunes). Sin embargo, no es el objetivo, mostrar con que nivel de exactitud el modelo, con las modificaciones estudiadas, reproduce la realidad. El hecho de usar una red muy simplificada (red estratégica de 245 nodos), de trabajar con una zonificación de Santiago en 40 sectores y de utilizar una función de tiempo generalizado de viaje hasta cierto punto arbitraria (aunque con parámetros razonables) confirma lo anterior.

## 2. Tratamiento de Líneas Comunes

Chriqui (1974) hace una buena presentación del problema de tratamiento de las líneas comunes en una red de transporte público. La Figura 1 sirve para ilustrar este problema.

Considérese los dos nodos A y B de una red. Sea  $L_1$  el conjunto de líneas que pasan por A en dirección a B. Para simplificar la explicación supóngase que el costo generalizado de viaje es igual a la suma de un tiempo de espera en A y de un tiempo de viaje en vehículo entre A y B. Chriqui plantea dos criterios extremos para determinar el tiempo de viaje entre A y B. El primero que es el más simple pero menos realista, supone que el tiempo de viaje entre A y B  $TV(A, B)$ , es el tiempo que toma el viaje usando el línea  $\ell \in L_1$ , tal que:

$$TV(A, B) = \min \{TA(\ell) + TD(A, B, \ell), \quad \forall \ell \in L_1\} \quad (1)$$

donde  $TA(\ell)$  es el tiempo medio de espera en el nodo A para línea  $\ell$  y  $TD(A, B, \ell)$ , es el tiempo de viaje en vehículo entre A y B tomando la línea  $\ell$ .

El segundo criterio define el tiempo de viaje entre A y B como:

$$TV(A, B) = W_{L_2} + D_{L_2} \quad (2)$$

donde  $L_2$  es el subconjunto de  $L_1$  para el cual  $TV(A, B)$  es mínimo y  $W_{L_2}$  y  $D_{L_2}$  son los tiempos de espera y de viaje en vehículo tomando cualquiera de las líneas de  $L_2$

El algoritmo CTPM de TRANSCOM (Chriqui, 1974) determina rutas de "costos percibidos mínimos" y está basado en este segundo criterio. Así, si entre dos nodos (A y B por ejemplo) existe un conjunto de líneas comunes  $L_1 = \{\ell_1, \ell_2, \ell_3, \dots, \ell_n\}$  con frecuencias  $F = \{f_1, f_2, f_3, \dots, f_n\}$  y tiempos de viaje en vehículo  $TV(A, B) = \{tv_1, tv_2, tv_3, \dots, tv_n\}$  y se supone que los tiempos de espera de las líneas  $\ell \in L_1$  son variables aleatorias, independientes, exponencialmente distribuidas con medias  $k/f_1, k/f_2, k/f_3, \dots, k/f_n$ , el valor esperado del tiempo total de viaje  $TV(A, B)$  para un viajero  $n$  que decide tomar la primera línea  $\ell \in L_1$  disponible, está dado por la expresión siguiente:

$$TV(A, B) = \frac{k + \sum_{i=1}^n tv_i f_i}{\sum_{i=1}^n f_i} \quad (3)$$

En el caso más general los tiempos de viaje en vehículo  $tv_1, tv_2, \dots, tv_n$  son diferentes y la decisión de tomar la primera línea de  $L_1$  que pase por A puede hacer que el tiempo de viaje total no sea el óptimo. Por esto, el algoritmo CTPM determina el conjunto  $L_2$  que es el subconjunto de  $L_1$  que minimiza  $TV(A, B)$  y supone que el viajero esperando un bus en A abordará la primera línea  $\ell \in L_2$  que llegue al paradero. Sea  $X = \{x_1, x_2, x_3, \dots, x_n\}$  un vector en el que  $x_i = 1$  si  $\ell_i \in L_2$  y  $x_i = 0$  si  $\ell_i \notin L_2$ . La determinación de  $L_2$  consiste entonces en resolver el problema siguiente:

$$\text{Min}_X TV(A, B) = \frac{k + \sum_{i=1}^n tv_i f_i x_i}{\sum_{i=1}^n f_i x_i} \quad (4)$$

Si se hace  $a_0 = k$ ,  $b_0 = 0$ ,  $a_i = tv_i f_i$  y  $b_i = f_i$  el problema anterior puede escribirse:

$$\text{Min}_X \frac{a_0 + \sum_{i=1}^n a_i x_i}{b_0 + \sum_{i=1}^n b_i x_i} \quad (5)$$



donde los coeficientes  $a_i$ ,  $b_i$ ,  $i = 0, 1, 2, 3, \dots, n$  son todos no negativos. La expresión (5) es un problema de programación hiperbólica  $\{0, 1\}$  sin restricciones. Hammer (1968) y Robillard (1971) resolvieron el problema general, Chriqui y Robillard (1975) presentan un algoritmo simple de solución del caso particular en que  $a_0 = k$ ,  $b_0 = 0$  y  $a_i/b_i = tv_i$ .

Una vez determinado el conjunto  $L_2$  la demanda es asignada entre las líneas comunes ( $\forall \ell \in L_2$ ), proporcionalmente a sus frecuencias.

Aun cuando el algoritmo CPTM es teóricamente más atractivo que algoritmos basados en el primer criterio señalado, aplicaciones prácticas para una red de 545 arcos y 115 líneas realizadas en Montreal han mostrado que el tiempo de ejecución de una asignación con dicho algoritmo, es del orden de seis veces mayor que uno basado en la expresión (1). Esto se debe a que cada vez que se calcula una etiqueta para un nodo, es necesario resolver un problema como el señalado en (5). En el caso de redes en las que el número de líneas comunes es significativamente mayor que el encontrado en la red de Montreal (existen pocos arcos con más de tres líneas comunes) esta diferencia de tiempo se hace aún mayor.

Varias formulaciones más simples que la anterior han sido implementadas en los modelos operacionales de asignación de viajes a redes de transporte público. Entre ellas cabe destacar la supuesta en el modelo UTPS (ver UMTA, 1972) y que ha sido introducida en otros modelos que se originaron en él, TRIPS entre otros (ver Martin and Voorhees As., 1982). En UTPS, el usuario del modelo debe definir para cada arco sus nodos extremos el tiempo de viaje en vehículo (o la velocidad) el modo del arco y, si corresponde, los números de las líneas que pasan por él. Se supone que para líneas del mismo modo los tiempos de viaje sobre un arco son iguales y así, sobre un tramo común, todas las líneas del mismo modo son cargadas con flujos proporcionales a sus frecuencias. Un algoritmo simplificado de TRANSCOM y MADITUC (ver Chapleau et al 1982, De Cea y Chapleau, 1984) consideran un máximo de tres líneas comunes (las tres de mayor frecuencia) y sólo cargan los flujos entre estas tres líneas. Otra simplificación, operativa sólo para redes pequeñas con un número reducido de líneas, es la que se hace en el modelo TERESE (ver Khelifi y Uhry, 1978). Se supone, en este caso, que en cada tramo en que existe un conjunto de líneas comunes los usuarios perciben una línea ficticia con una frecuencia igual a la suma de las líneas que la componen. Evidentemente en redes con alto número de líneas y con un importante número de arterias sobre las que circulan muchas líneas comunes, el modelo es inaplicable debido al número de líneas ficticias que se requiere codificar.

La simplificación hecha por MADITUC y TRANSCOM es inadecuada cuando en la red existe una cantidad importante de arcos con más de tres líneas comunes (más del 60% de los arcos de la red utilizada en nuestro estudio). Sin embargo, parecería razonable si no se limitara el número de líneas comunes. En esta investigación hemos considerado como base de comparación de las modificaciones propuestas el modelo MADITUC sin restricciones y hemos analizado el efecto que tiene en la calidad de los resultados de la asignación el hecho de limitar el número de líneas comunes a considerar en la etapa de cálculo de rutas mínimas, cuando en la etapa de asignación los viajes son cargados a todas las líneas comunes existentes. En general debería esperarse que con un número limitado de líneas comunes (3 a 5 por

ejemplo) el algoritmo sea capaz de identificar la verdadera ruta mínima (en términos de secuencia de nodos sobre la red: nodo origen, nodos de transbordo, nodo de destino). Es evidente que mientras mayor sea el número de líneas comunes consideradas en el algoritmo de cálculo de rutas mínimas mayor es la posibilidad de que la ruta obtenida sea efectivamente la óptima como se ha dicho. El análisis que se presenta luego, intenta buscar un compromiso entre ahorro de tiempo y la calidad de los resultados. En todos los casos los tiempos y los resultados obtenidos serán comparados con el caso "base" que consiste en no poner límites al número de líneas comunes consideradas.

### 3. Transbordos

Otra vía de simplificación de los algoritmos de cálculo de rutas mínimas es la que se relaciona con el tratamiento de los transbordos. En este trabajo hemos testado dos algoritmos. Uno de ellos TRAMIN, es debido a Chapleau (1974). Este algoritmo consiste básicamente en la determinación de rutas con el mínimo número de transbordos. Para ello supone una alta penalidad asociada al hecho de transbordar. Si entre dos nodos existen varias rutas con el mínimo número posible de transbordos el algoritmo elige, entre ellas, la de costo generalizado mínimo.

El segundo algoritmo, propuesto en este trabajo es una simple modificación del algoritmo de cálculo de rutas mínimas, con tratamiento de líneas comunes, de los modelos TRANSCOM y MADITUC. Se supone, que el número máximo de transbordos en una viaje es uno. Así, cuando un nodo  $k$ , adquiere, etiqueta permanente en la determinación del árbol de rutas mínimas a partir de un nodo fuente, el algoritmo pregunta en que nodo se hizo el último transbordo. Si ese nodo no es el nodo fuente, no hay etiquetaje a partir de  $k$  y se pasa a determina un nuevo nodo con etiqueta permanente. En caso en que el nodo en el que se hizo el último transbordo es el nodo fuente, dado que hasta el nodo  $k$  no ha habido transbordos, se inicia el etiquetaje a partir de  $k$ . Es claro que el algoritmo presenta al menos los siguientes problemas, que se ilustran en la Figura 2.

- El algoritmo no encontrará rutas entre  $F$  y aquellos nodos a los que sólo se puede llegar con dos transbordos. Es el caso del nodo 5 del ejemplo. Para llegar a él es necesario transbordar en 1 y luego en 4.

- El algoritmo puede no encontrar rutas entre  $F$  y un nodo dado, aun cuando exista una ruta con un transbordo. Es el caso, por ejemplo del nodo 4. Si la ruta mínima entre  $F$  y 1 es  $F-2-1$  usando las líneas  $\ell_2$  y  $\ell_3$ , el algoritmo no encontrará la ruta  $F-1-4$  por las líneas  $\ell_1$  y  $\ell_6$ , aun cuando ésta cumple con las restricciones impuestas.

- El algoritmo no garantiza que las rutas mínimas obtenidas, sean las mínimas posibles, dada la restricción de un transbordo como máximo. Esta situación podría suceder en el caso de la ruta entre  $F$  y 3. Si la ruta mínima entre  $F$  y 1 es  $F-2-1$  el algoritmo no examinará nunca la ruta  $F-1-3$ , aun cuando ésta sea menor que la ruta  $F-2-3$ .



#### 4. Análisis Experimental

##### 4.1. La red utilizada

Como se mencionó en la sección 1, la red de transporte público utilizada en este trabajo corresponde a una simplificación de la red existente en Santiago en 1978. En ella, sólo están consideradas las arterias por las que circula locomoción colectiva. Las 272 líneas de transporte público de superficie originan sobre esta red 245 nodos, 694 arcos unidireccionales y 223 recorridos distintos. A ellas debe agregarse una línea de Metro. No obstante esta simplificación, la red ha guardado su característica más relevante desde el punto de vista de esta investigación, cual es la cantidad de líneas y en especial la cantidad de arcos con número elevado de líneas comunes. La Figura 3 ilustra este aspecto. De los 694 arcos existe un 33% sobre los cuales circulan más de 10 líneas y un 18% sobre los cuales este número es mayor que 20. Estas cifras contrastan bastante con las que se observa en redes de transporte público en países desarrollados.

Respecto a los datos de demanda se adoptó una zonificación a 40 zonas. Los elementos no nulos de la matriz de viajes entre zonas (para el período en estudio) resultaron ser 1.074, con un flujo total asociado de 504.964 viajes. Se eligió un máximo de 2 nodos de acceso por zona lo que dio lugar a 4.568 elementos no nulos de flujos entre nodos. El número de orígenes de viajes, que corresponde al número de árboles de rutas mínimas a calcular en la asignación, es de 74.

##### 4.2. Definiciones previas

Para efectos de comparación se ha supuesto que el algoritmo que considera todas las líneas existentes en la etapa de cálculo de rutas mínimas y que luego asigna a todas las líneas comunes que aparecen en el árbol, producirá una asignación que se denominará "realidad". Todas las demás asignaciones serán comparadas con ésta. Para hacer referencia a los diversos algoritmos analizados, que son modificaciones del algoritmo utilizado en MADITUC, se usará la nomenclatura siguiente:

S/R X : Corresponde al algoritmo sin restricción en el número de transbordos que considera un máximo de X líneas comunes en el cálculo de rutas mínimas y que asigna luego a todas las líneas comunes existentes en cada sección de las rutas mínimas obtenidas.

TRA1 X : Corresponde al algoritmo con restricciones en el número de transbordos (máximo 1), que considera un máximo de X líneas comunes en el cálculo de rutas mínimas y que asigna luego a todas las líneas comunes existentes en cada sección de las rutas mínimas obtenidas.

TRAMIN : Corresponde al algoritmo que no considera líneas comunes en el cálculo de rutas mínimas, que determina rutas con número mínimo de transbordos y asigna luego a todas las líneas comunes existentes en cada sección de las rutas mínimas obtenidas.



#### 4.3. Comparación de algoritmos

##### a) Tiempo de cálculo de un árbol de rutas mínimas

Para los dos algoritmos definidos en la sección 4.2, S/R X y TRAl X, se analizó el efecto de aumentar el límite de líneas comunes a considerar en la determinación de un árbol de rutas mínimas sobre el tiempo de cálculo de dicho árbol. Los resultados presentados en la Tabla 1 y la Figura 4 corresponden a tiempos promedio por árbol (en segundos). En este caso, como en todos los que se presentan luego, se utilizó el computador DIGITAL DEC-10 de la Pontificia Universidad Católica de Chile

Número de líneas comunes (X)	Algoritmo utilizado	
	S/R X	TRAl X
75	30,2	21,9
55	30,2	21,9
35	30,1	21,8
30	30,0	21,6
25	29,7	21,4
20	29,1	20,9
15	28,1	20,0
10	26,2	18,4
9	25,6	17,9
8	25,0	17,4
7	24,2	16,8
6	23,2	16,0
5	21,8	15,0
4	20,1	13,7
3	17,7	12,0
2	14,1	9,2
0	2,9	1,5

TABLA 1: Tiempo promedio de cálculo de un árbol de rutas mínimas (segundos de CPU)

Tanto para el caso del algoritmo sin restricciones en el número de transbordos de una ruta (S/R X) como en el que éstos son restringidos a un máximo de 1 (TRAl X), se observa un aumento del tiempo de cálculo de un árbol en la medida que aumenta el número de líneas comunes a considerar (X). Este aumento que es drástico cuando se pasa de no considerar líneas comunes (S/R 0 y TRAl 0) a considerar 3 y 4 como máximo, se atenua bastante entre 5 y 15 líneas comunes. Por sobre ese límite el tiempo de cálculo de un árbol de rutas mínimas permanece prácticamente constante. El diagrama de barras de la figura 3, que muestra el número de arcos para diferentes intervalos de líneas comunes, explica por sí solo los resultados de la Tabla 1 y las curvas de la Figura 4.

Dado el cálculo de árboles es la etapa más consumidora de tiempo de computación en un modelo de asignación de tipo "todo o nada", resulta interesante, a la luz de los resultados presentados en la Tabla 1

explorar el efecto producido en los resultados de una asignación, limitando las líneas comunes en el cálculo de rutas mínimas, considerando todas las líneas comunes existentes, en la etapa de carga. Evidentemente se debe buscar una solución de compromiso entre la disminución del tiempo de ejecución de una asignación y el error introducido en sus resultados.

b) Comparación de resultados de asignación para diferentes algoritmos

Con el objeto de determinar una medida de error de una asignación respecto a la situación base (asignación con el algoritmo S/R 75) se utilizó una metodología similar a la descrita en Chapleau y De Cea (1983). Se comparan tres tipos de indicadores que corresponden a resultados de diferentes niveles de agregación. El nivel más desagregado corresponde a una comparación de itinerarios (rutas sobre la red). De esta comparación se obtiene el porcentaje de rutas no reproducidas por un algoritmo dado respecto a las determinadas por el algoritmo base. Para el nivel intermedio de agregación se comparan los flujos resultantes sobre la red en los arcos de cada línea y finalmente, para el nivel más agregado, se hace una comparación de indicadores globales promedio tales como tiempo generalizado total de viaje, tiempo de viaje en vehículo, tiempo de espera y tarifa, distancia de viaje y número de secciones de línea por viaje.

Debido a que los algoritmos analizados en este trabajo son modificaciones del algoritmo de cálculo de rutas mínimas del modelo MADITUC, fue necesario, como paso previo, estimar el factor de dispersión del modelo logit que éste usa para repartir los flujos interzonales entre las rutas alternativas. Estas rutas alternativas, como se indica en De Cea y Chapleau (1984) corresponden a las rutas de tiempo (costo) generalizado mínimo entre los nodos de acceso en la zona de origen y los nodos de acceso en la zona de destino de un viaje. Por no existir información sobre los viajes (itinerarios) reales de los usuarios sobre la red de 1978 no fue posible calibrar dicho factor. Sin embargo, dado que la finalidad de esta investigación es comparar algoritmos con uno definido como base y no con la realidad, se usó arbitrariamente un valor que reprodujera, aproximadamente, el número promedio de secciones de línea por viaje. Un factor de dispersión  $\theta = 0,06$  produjo, usando el algoritmo S/R 75, 1,25 secciones de línea por viaje. A fin de verificar si el orden de magnitud del valor obtenido era razonable, se calculó la desviación estándar del tiempo generalizado de viaje y a partir de ella, suponiendo que este tiempo tiene una distribución de Weibull (ver Figura 5), se determinó un valor "teórico" de 0,042 para dicho factor.

La Tabla 2 y la Figura 6 muestran el efecto del factor de dispersión ( $\theta$ ) sobre el número de secciones de línea por viaje para asignaciones realizadas con el algoritmo S/R 75. Obviamente para  $\theta = 0$  se obtiene el máximo número de transbordos. Para ese valor de  $\theta$  el modelo produce una asignación idéntica a las rutas alternativas entre dos zonas, sin considerar los tiempos generalizados de cada una. A medida que  $\theta$  aumenta, las rutas de mayor tiempo (que en muchos casos tienen transbordos) son cargadas con menos viajes. Cuando  $\theta$  se hace muy grande (0,18-0,20 en este caso) el algoritmo asigna sólo a la ruta más corta entre centroides, por lo que es lógico esperar una menor cantidad de viajes con transbordos (debe considerarse que la tarifa utilizada es equivalente a

Factor de dispersión ( $\Theta$ )	N° Secciones	Factor de dispersión	Número Secciones
0,00	1,500	0,11	1,188
0,01	1,448	0,12	1,185
0,02	1,399	0,13	1,182
0,03	1,352	0,14	1,180
0,04	1,317	0,15	1,178
0,05	1,283	0,16	1,176
0,06	1,254	0,17	1,174
0,07	1,235	0,18	1,172
0,08	1,219	0,19	1,172
0,09	1,207	0,20	1,172
0,10	1,196		

TABLA 2: Efecto del parámetro de dispersión sobre el número promedio de secciones por viaje (algoritmo S/R 75)

Algoritmo	% de viajes no reproducidos	
	$\Theta = 0,00$	$\Theta = 0,06$
S/R 5	1,4	0,6
S/R 4	1,6	0,8
S/R 3	2,9	1,4
S/R 2	4,9	2,6
S/R 0	9,5	5,0
TRA1 75	0,3	0,0
TRA1 5	1,7	0,6
TRA1 4	1,9	0,9
TRA1 3	3,2	1,4
TRA1 2	5,1	2,6
TRA1 0	9,8	5,0
TRAMIN	11,3	5,7

TABLA 3: Comparación de resultados de asignación para diferentes modelos y factores de dispersión



35 min. para los buses y de 25 min. para el Metro, cifras no despreciables frente al tiempo generalizado total de viaje).

La Tabla 3 resume la comparación de itinerarios producidos por los algoritmos propuestos con aquellos que se obtienen con el algoritmo base, S/R 75, para dos valores de  $\Theta$ ,  $\Theta = 0,00$  y  $\Theta = 0,06$ . Como se ve, sistemáticamente el porcentaje de viajes no reproducidos con un algoritmo es menor para  $\Theta = 0,06$ , lo que resulta consistente con lo comentado anteriormente. Con ese factor de dispersión aumenta el número de itinerarios con una sección y en esos casos los algoritmos propuestos reproducen exactamente los itinerarios creados por el algoritmo base. Es interesante observar que el algoritmo TRAl 0, que es del orden de 20 veces más rápido que el S/R 75, sólo es incapaz de reproducir el 5% de los itinerarios (que en total son del orden de 504.000 viajes individuales).

La Tabla 4 y las Figuras 7 y 8 ilustran los resultados de la comparación entre los flujos asignados a 7.639 arcos de línea por el algoritmo S/R 75 y cada uno de los algoritmos propuestos, para  $\Theta = 0,06$ . La Tabla 4 muestra que los coeficientes  $r^2$  corregidos son todos altos, obteniéndose para el algoritmo TRAl 0 la peor correlación, con un valor de 0.96199. Esta misma información se presenta en forma gráfica en las Figuras 7 y 8, en las que se compara el algoritmo S/R 75 con los algoritmos TRAl 5 y TRAl 0. La fuerte correlación existente en ambos casos es obviamente resultado del alto grado de coincidencia (mayor que 95%) a nivel de viajes individuales. Lo mismo y aún más acentuado, sucede con los indicadores globales, los que para todos los algoritmos testeados son prácticamente idénticos a los obtenidos con el algoritmo base (ver Tabla 5). Es claro que el tiempo generalizado total, promedio, para el algoritmo base (S/R 75) debería ser menor o igual a los tiempos obtenidos con los algoritmos restantes. Las excepciones de la Tabla 5 con los algoritmos TRAl 75 a TRAl 2 se deben a que en este caso los viajes para los que no existe una ruta con menos de dos transbordos (32 viajes) no son asignados. Son justamente esos viajes los que tienen los mayores tiempos generalizados (mayores que 200 min) lo que explica la aparente anomalía de los resultados presentados.

Por último, en la Tabla 6 se presenta un resumen en el que se combinan tiempos de ejecución de los distintos algoritmos utilizados con algunas medidas de la calidad de los resultados obtenidos.

## 5. Conclusiones

Como se ha dicho, al enfrentar el problema de análisis de una red de transporte público, la primera decisión que debe tomar el analista se relaciona con el nivel de detalle con que se debe codificar dicha red. Por un lado, es deseable que las líneas sean representadas en la forma más cercana a la realidad. En la situación límite, sería bueno representar cada paradero por un nodo. En la práctica, dado que el tiempo de ejecución de un modelo de asignación depende fundamentalmente del número de árboles a calcular y el tiempo de cálculo de un árbol del número de nodos de la red, el analista debe buscar un compromiso entre el nivel de

Algoritmo	R <sup>2</sup> Corregido
S/R 5	0,99895
S/R 4	0,99839
S/R 3	0,99651
S/R 2	0,98964
S/R 0	0,96191
TRAl 75	1,00000
TRAl 5	0,99894
TRAl 4	0,99838
TRAl 3	0,96450
TRAl 2	0,98976
TRAl 0	0,96199
TRAMIN	0,96256

TABLA 4: Comparación de flujos asignados por arco  
(coeficientes de correlación)

Algoritmo	Tiempo gene ralizado de viaje (min)	Tiempo de viaje en veh. (min)	Tiempo de espera y tarifa (min)	Distanc cia de viaje (km)	Nº de secciones
S/R 75	92,23	25,11	45,41	8,47	1,25
S/R 5	92,23	25,11	45,41	8,47	1,25
S/R 4	92,23	25,11	45,41	8,47	1,25
S/R 3	92,23	25,12	45,40	8,47	1,25
S/R 2	92,24	25,11	45,41	8,47	1,25
S/R 0	92,26	25,10	45,44	8,47	1,25
TRAl 75	92,22	25,11	45,40	8,47	1,25
TRAl 5	92,22	25,11	45,40	8,47	1,25
TRAl 4	92,22	25,10	45,40	8,47	1,25
TRAl 3	92,22	25,12	45,39	8,47	1,25
TRAl 2	92,22	25,11	45,40	8,47	1,25
TRAl 0	92,25	25,10	45,43	8,47	1,25
TRAMIN	92,26	25,13	45,42	8,47	1,25

TABLA 5: Comparación de indicadores globales  
( $\Theta = 0,06$ )

Algoritmo	% viajes no repro- ducidos	Correlación flujos en arcos respec- to a S/R 75	Tiempo de cálculo árboles (min, seg)	Tiempo Total (min, seg.)
S/R 75	0,00	1,00000	37:14	39:27
S/R 5	0,60	0,99895	26:54	29:04
S/R 4	0,80	0,99839	24:49	26:57
S/R 3	1,40	0,99651	21:53	24:00
S/R 2	2,60	0,98964	17:10	19:15
S/R 0	5,00	0,96191	3:16	5:35
TRA1 75	0,00	1,00000	27:03	29:08
TRA1 5	0,60	0,99894	18:35	20:39
TRA1 4	0,90	0,99838	16:59	19:06
TRA1 3	1,40	0,99645	14:48	16:53
TRA1 2	2,60	0,98976	11:23	13:23
TRA1 0	5,00	0,96199	1:48	4:05
TRAMIN	5,70	0,96256	1:09	3:19

TABLA 6: Comparación de resultados de asignación y tiempos de ejecución para distintos algoritmos.  $\theta = 0,06$

detalle de la red y el tiempo de ejecución de la asignación. La gama de posibilidades que representan los algoritmos analizados en este trabajo dan al analista una herramienta adicional de decisión. Así por ejemplo, podrá decidir codificar la red con gran detalle y luego usar un algoritmo "no tan bueno" de asignación (TRA1 0 por ejemplo).

En el otro extremo, podrá decidir una codificación muy esquemática de la red (red agregada) y usar el mejor algoritmo de asignación posible (S/R 75 por ejemplo). La importancia del problema de agregación espacial en los errores de los modelos de asignación hace pensar que un mayor detalle en la codificación, unida a la utilización de algoritmos de cálculo de rutas mínimas más simples llevaría a mejores resultados de asignación, en tiempos de ejecución (tiempos de cpu) razonables.

El contar con algoritmos más rápidos que los actuales podría permitir además tratar problemas como el de diseño de redes de transporte público, que hoy día sólo pueden ser tratados a nivel de pequeñas redes ejemplo. Resulta obvio que un problema iterativo que requiere, en cada iteración, realizar una asignación a la red de transporte público necesita de algoritmos que, aunque menos exactos, sean bastante más rápidos que los existentes.



## Agradecimientos

Este trabajo fue realizado como parte de una investigación que cuenta con financiamiento de la Dirección de Investigación de la Pontificia Universidad Católica de Chile y del International Research Centre de Canadá.

## Referencias

- CHAPLEAU, R. (1974) Réseaux de transport en commun: Structure informatique et affectation. Publication N°13, Centre de Recherche sur les Transports, Université de Montréal, Canadá.
- CHAPLEAU, R., ALLARD, B. y CANOVA, M. (1982) MADITUC, un modèle de planification opérationnelle adapté aux entreprises de transport en commun de taille moyenne. Publication N°265, Centre de Recherche sur les Transports, Université de Montréal, Canadá.
- CHAPLEAU, R. y DE CEA, J. (1983) User perception of transit network characteristics from the viewpoint of an assignment model. En Paul Baron y Heine Nuppenau (eds.), Research for Transport Policies in a Changing World, SNV Studiengesellschaft, Hamburgo.
- CHRIQUI, C. (1974) Réseaux de transport en commun: Les problèmes de cheminement et d'accès. Publication N°11, Centre de Recherche sur les Transports, Université de Montréal, Canadá.
- CHRIQUI, C. y ROBILLARD, P. (1975) Common bus lines. Transportation Science, Vol 9, 115-121.
- DE CEA, J. y CHAPLEAU, R. (1983) Modelling the behaviour of public transport users: an empirical study for the city of Montreal. 11th PTRC Summer Annual Meeting, University of Sussex, 4-7 Julio 1983, Inglaterra.
- DE CEA, J. y CHAPLEAU, R. (1984) MADITUC: un modelo de asignación a rutas múltiples en redes de transporte público, Apuntes de Ingeniería 15, 113-140.
- DICTUC (1978) Encuesta origen-destino de viajes 1977, para el Gran Santiago. Departamento de Ingeniería de Transporte. Pontificia Universidad Católica de Chile, Santiago.
- HAMMER, P.L. y RUDEMAN S. (1968) Boolean Methods in Operations Research. Springer-Verlag, Nueva York.
- KHELIFI, D. y UHRY, J.P. (1978) Terese (TEsts de RESEaux): modèles de transport collectif urbain. R.R. 125, Institut National Polytechnique de Grenoble, Francia.
- MARTIN AND VOORHEES ASSOC. (1982) TRIPS: Transportation Improvements Programming Systems, Londres.

ROBILLARD, P. (1971) (0,1) hyperbolic programming problems. Naval Research Logistics Quarterly, Vol 18, N°1, 47-57.

U.M.T.A. (1972) UMTA transportation planning system, network development manual. NTIS PB 212-930, Washington D.C.

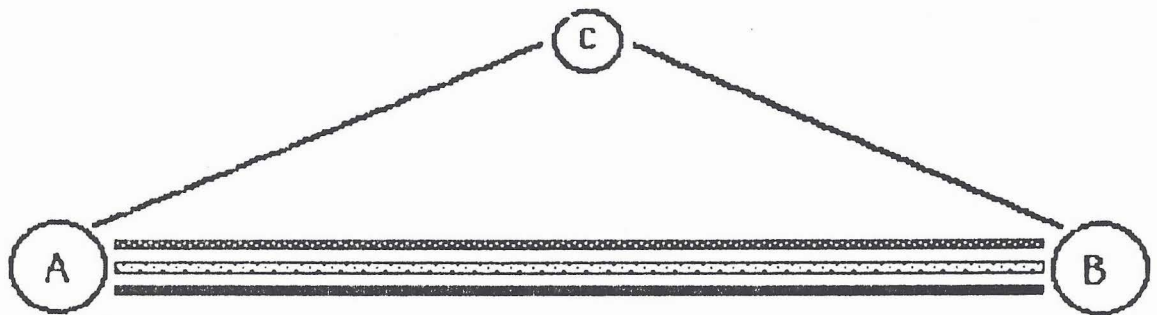


FIGURA 1 : Líneas alternativas para viajar entre nodos A y B

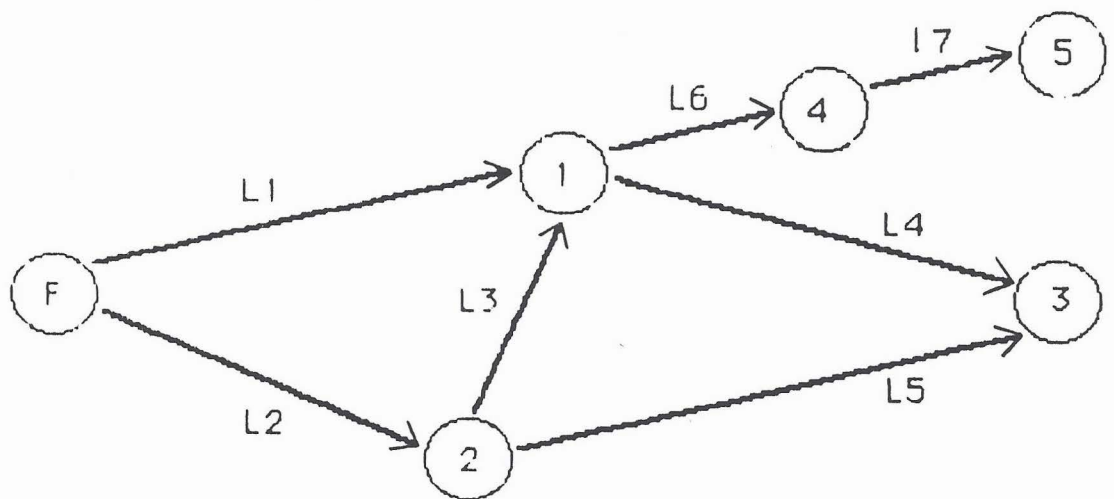


FIGURA 2: Ilustración de limitaciones del algoritmo de cálculo de rutas mínimas con restricción al número de transbordos



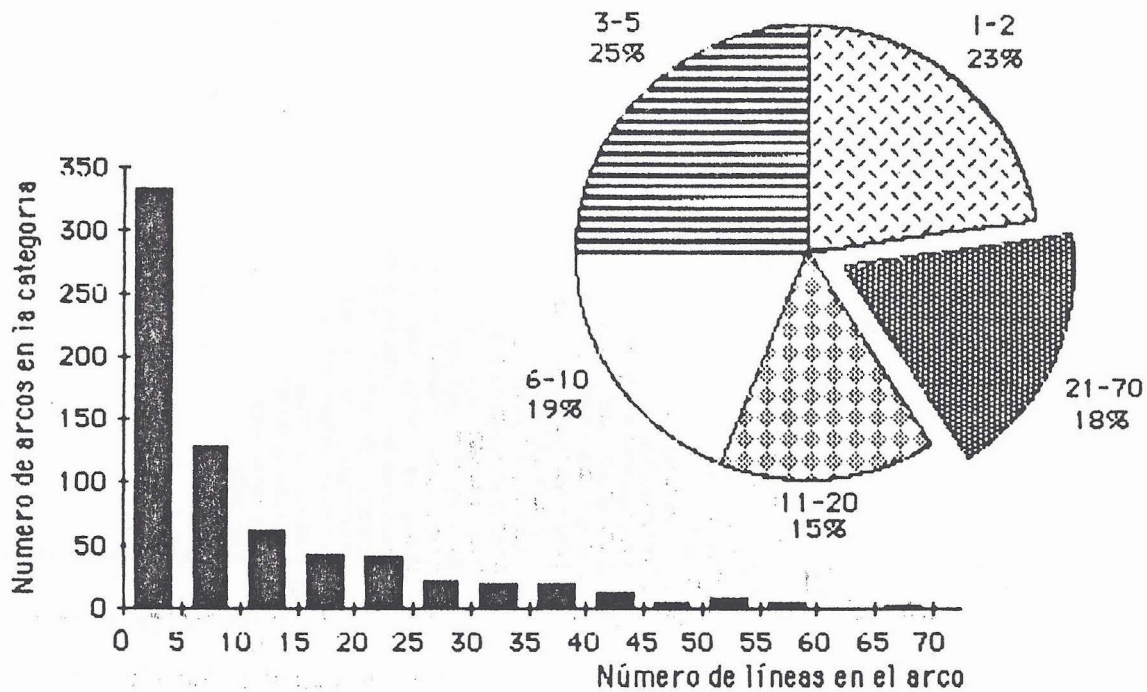


FIGURA 3 : Distribución de los arcos de la red según el número de líneas que pasan por el arco (694 arcos)

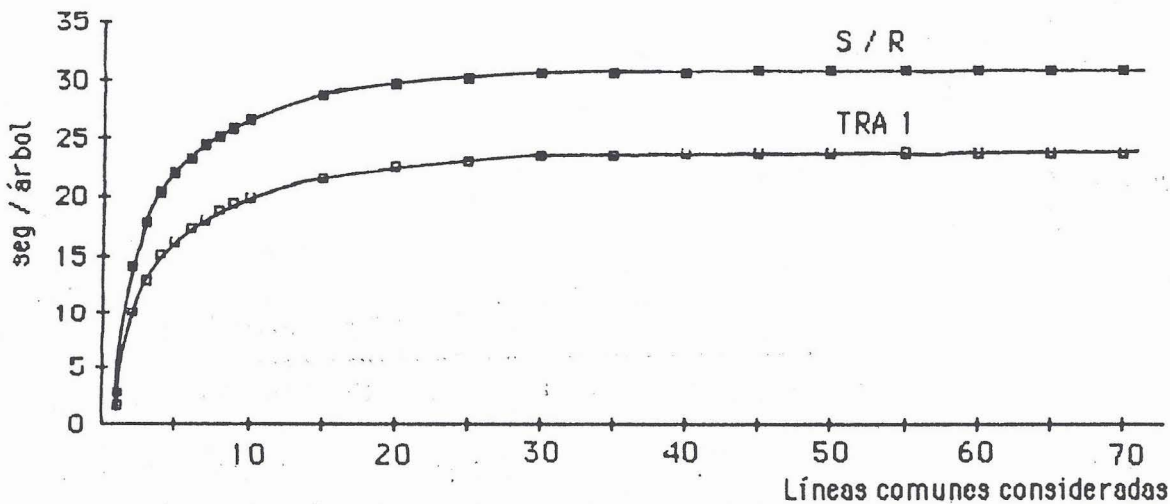


FIGURA 4 : Variación del tiempo promedio de cálculo de un árbol de rutas mínimas con el número máximo de líneas comunes consideradas

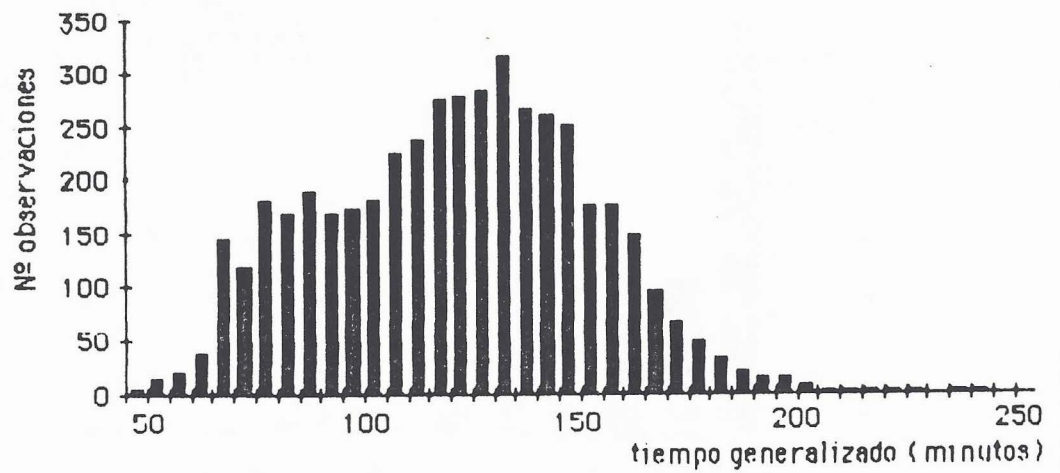


FIGURA 5 : Distribución del tiempo generalizado de viaje entre z

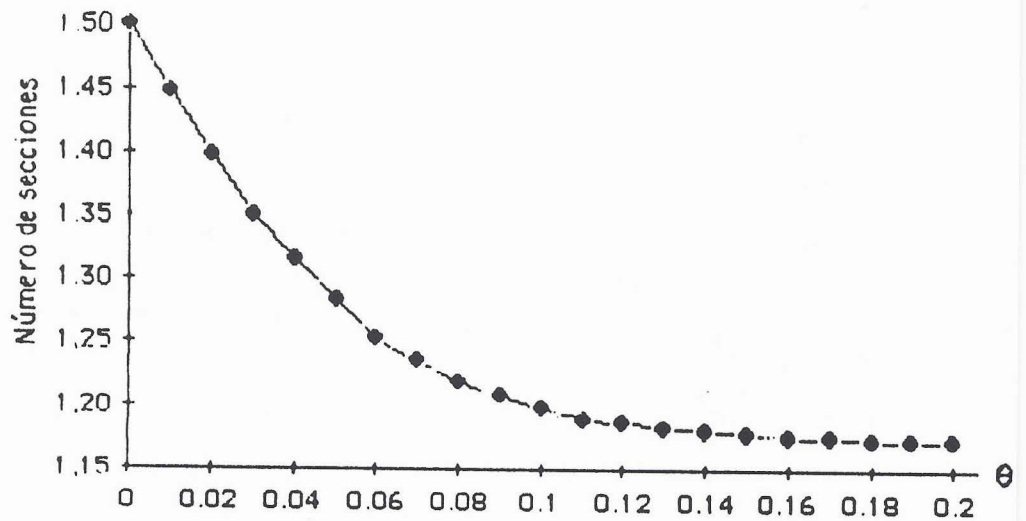


FIGURA 6 : Efecto del parámetro de dispersión  $\theta$  sobre el número promedio de secciones utilizadas por los individuos del sistema (algoritmo S/R 75)

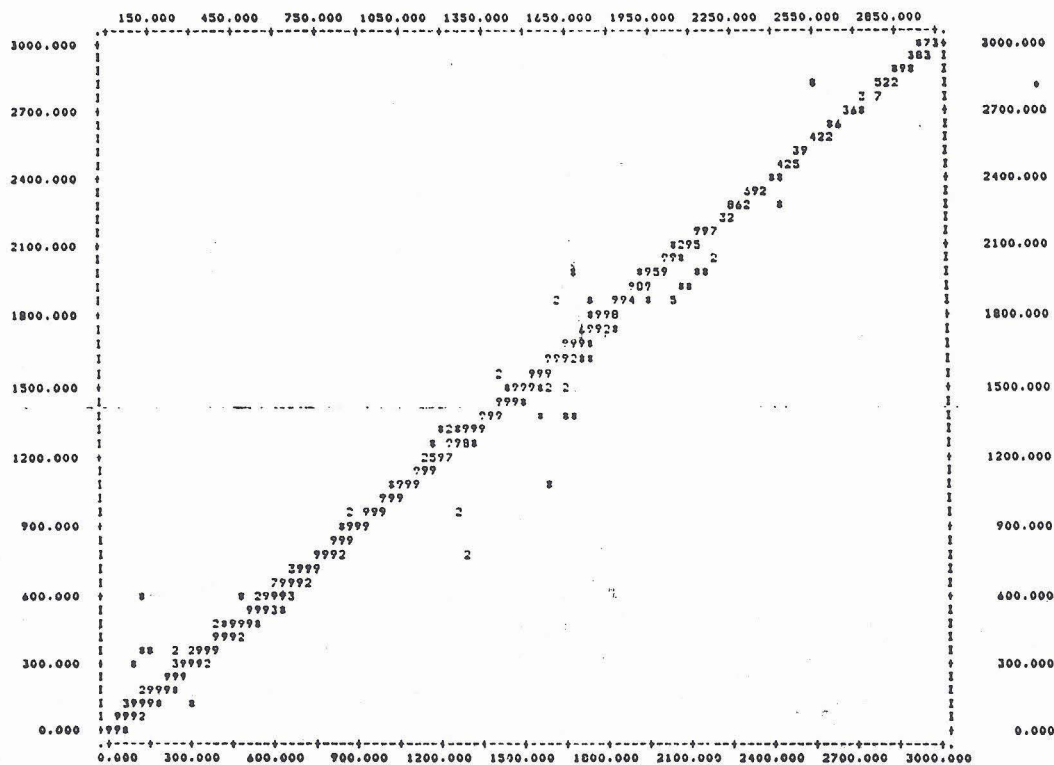


FIGURA 7 : Comparación del flujo asignado a las líneas del sistema con algoritmos S/R 75 (vertical) y TRAI 5 (horizontal)

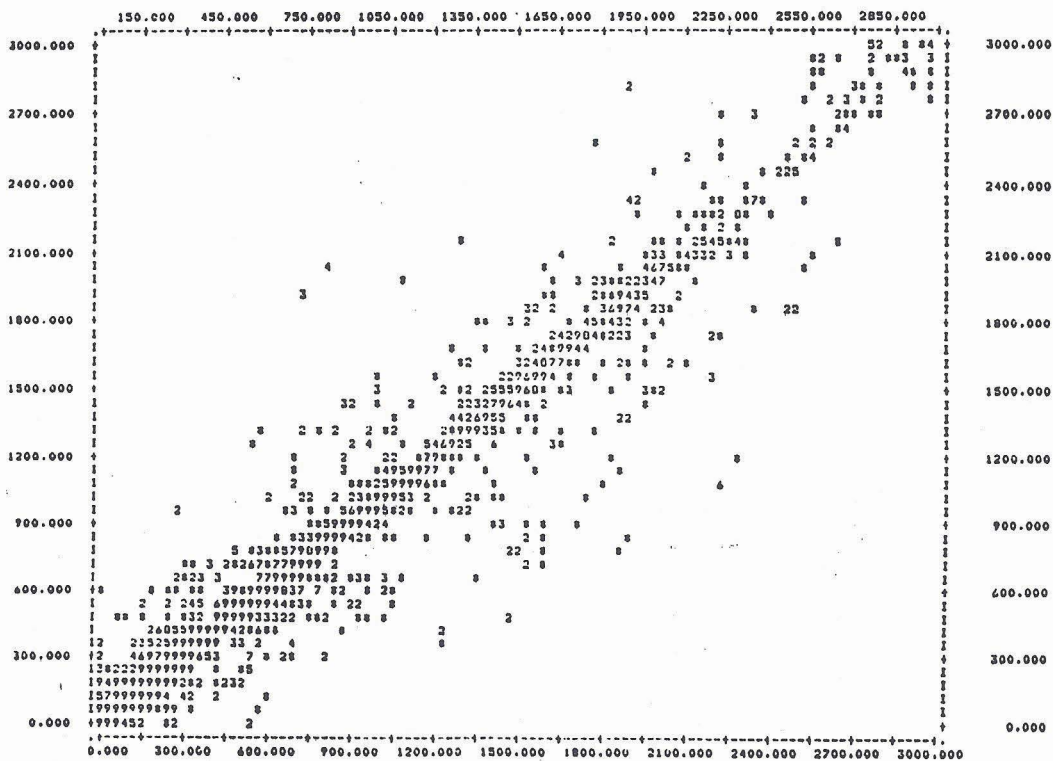


FIGURA 8 : Comparación del flujo asignado a las líneas del sistema con algoritmos S/R 75 (vertical) y TRAI 0 (horizontal)