

**Uso de conjunto de concentración en búsqueda tabú
para problemas de rutas**

Joaquín A. Pacheco y Cristina R. Delgado

Departamento de Economía Aplicada. Universidad de BURGOS
Facultad de C. EE y EE
C/Parralillos s/n. BURGOS 09001. SPAIN
Tf. 947 25 90 21

E-mail : jpacheco@ubu.es, cdelgado@ubu.es

Resumen:

En este trabajo se propone un algoritmo para el problema de rutas con ventanas de tiempo, carga y descarga simultánea y flota heterogénea, basado en un proceso de Búsqueda Tabú. Lo más novedoso en este algoritmo es la incorporación en el procedimiento de intensificación de las ideas de Rosing (1.997) y Rosing y ReVelle (1.997) sobre el denominado *Conjunto de Concentración*. Para comprobar la eficacia tanto de este procedimiento de Intensificación, como la de todo el algoritmo, se usan instancias simuladas y diferentes librerías disponibles en la red.

Palabras clave: *Problemas de Rutas Mixtos, Flota Heterogénea, Metaheurísticos, Búsqueda Tabú, Intensificación, Concentración Heurística.*

1 - INTRODUCCIÓN

El *Problema de Rutas de Vehículos con Ventanas de tiempo, Carga y Descarga Simultánea*, o sencillamente VRPTW Mixto (Mixed VRPTW), y *flota heterogénea* puede ser descrito de la forma siguiente:

Considérese un conjunto de puntos $\{2, 3, \dots, n1\}$ donde hay que entregar unas determinadas cantidades de mercancía $q(i)$, $i = 2, \dots, n1$, en forma de *palés*, desde un origen 1; considerese además otro conjunto de puntos $\{n1+1, n1+2, \dots, n1+n2\}$ donde hay que recoger otras cantidades $q(i)$, $i = n1+1, \dots, n1+n2$, y llevarlas al origen 1 (a partir de ahora $n = n1+n2$). Para cumplir estos requerimientos se dispone de una flota heterogénea con diferentes tipos de vehículos; cada tipo de vehículo tiene una capacidad de carga diferente; cada punto del problema lleva asociado un intervalo de tiempo de visita $[e_i, l_i]$, $i = 2, \dots, n$, de forma que si se llega a i antes del instante e_i se ha de esperar, y no puede ser visitado más tarde del instante l_i ; las distancias d_{ij} y tiempos t_{ij} entre cada par de puntos $i, j \in \{1, 2, \dots, n\}$ son conocidas; el número de vehículos disponibles se denota por $ntipos$ y las capacidades y costes de cada tipo por $capactipo(i)$ y $costetipo(i)$, $i = 1, \dots, ntipos$; obviamente a más capacidad mas coste. El tiempo de carga y descarga es despreciable respecto a los tiempos de transporte, y por ello no se tiene en cuenta.

Se ha de diseñar un conjunto de rutas de coste mínimo verificando las siguientes restricciones:

- cada ruta comience y finalice en el punto 1;
- se entregue la mercancía correspondiente a cada uno de los puntos del conjunto $\{2, \dots, n1\}$, y se recoja de cada uno de los puntos del conjunto $\{n1+1, n1+2, \dots, n1+n2\}$;

- el número de *palés* que en cada momento transporte cada vehículo no supere su capacidad;
- cada punto $i, i = 2, \dots, n$, sea visitado exactamente una vez;
- se respeten los intervalos de tiempo de visita.

El coste total a minimizar f es la suma de dos componentes:

- un coste variable, proporcional a la distancia total recorrida;
- un coste fijo por cada vehículo (según su tipo).

Este modelo supone una generalización del VRP y del VRPTW: considerando $n_2 = 0$ (solo descarga) y $n_{tipos} = 1$ se obtiene la formulación del VRPTW. La modificación en la función objetivo del coste fijo por vehículo permite plantear problemas cuyo objetivo es minimizar el número de vehículos, o problemas donde el objetivo es minimizar la distancia total recorrida.

Existen muchos algoritmos de solución para el VRP y el VRPTW en la literatura. Recopilaciones de los principales algoritmos se pueden encontrar en trabajos como los de Bodin y Golden (1.981), Desrochers y otros (1.988), Haouri y otros (1.990), Laporte (1.992) y Laporte y Osman (1.995). En los últimos años ha tomado importancia el desarrollo de algoritmos basados en procesos denominados Metaheurísticos como *Algoritmos Genéticos (GS)*, *Temple Simulado(AS)*, *Búsqueda Tabú (TS)*, *GRASP*, *Búsqueda Local Guiada (GLS)*, *Colonias de Hórmigas*, ... especialmente a partir de los trabajos de Gendreau y otros (1.991) ((1.994), en versión posterior) y de Osman (1.993); también de los de Potvin y otros (1.993) y Potvin y Bengio (1.994), Thangiah y otros (1.993) y (1.994), Campos y Mota (1.995), Kontoravdis y Bard (1.995), Rochat y Taillard (1.995), Kilby y otros (1.997), Backer y otros (1.997), Bullnheimer y otros (1.997) o Rego (1.998).

En cuanto al modelo que se está tratando existen referencias recientes en Pacheco y Delgado (1.999) y (2.000). El presente trabajo, que puede considerarse una ampliación de los dos anteriores, presenta un algoritmo basado en un procedimiento de Búsqueda Tabú que consta de Procedimiento Básico, Fase de Intensificación y Fase de Diversificación e incorpora en el Procedimiento Básico la estrategia denominada Oscilación Estratégica. La parte más novedosa es la fase de Intensificación, que se basa principalmente en la definición de un *Conjunto de Concentración*. Esta idea fue propuesta recientemente por Rossing (1.997), Rossing y Reville (1.997) y Rossing y otros (1.998), en la descripción de una Metaestrategia denominada *Concentración Heurística*.

El trabajo se organiza como sigue: en las dos secciones siguientes (segunda y tercera) se describen los entornos que se utilizarán; en la cuarta se propone un método basado en variables globales para verificar la factibilidad de las soluciones vecinas; en la quinta se describen los movimientos vecinales que usa y se desarrolla el algoritmo de Búsqueda Tabú; en la sección sexta se muestran los resultados de una serie de pruebas usando librerías del VRP y VRPTW; finalmente, en la séptima sección se exponen las conclusiones.

A partir de ahora se denotará por S el conjunto de soluciones factibles del problema, y por f la función de costes a minimizar definida en S .

2 - VECINDARIO TIPO OR

Se van a considerar dos tipos de vecindarios: uno basado en la idea propuesta por Or (1.976) para el *Problema del Viajante* o TSP, y otro usado más recientemente en los trabajos de Gendreau y otros (1.991), Osman (1.993), Campos y Mota (1.995),... que se explicará en la siguiente sección. (En los trabajos de Taillard (1.995) se definen interesantes estructuras vecinales parecidas aunque algo más complejas).

Se van a definir como soluciones vecinas las obtenidas por el método de intercambio propuesto por Or (1.976) que sean factibles. El método de intercambio de Or es una variante de los conocidos intercambios r -óptimos desarrollados por Lin (1.965) y Lin & Kernighan (1.973) para el TSP simétrico. Como se verá fácilmente, el método de Or se puede utilizar en problemas asimétricos. La eficacia de este método para el TSP ha sido contrastada en trabajos como el de Nurmi (1.991).

Or propone restringir la búsqueda de intercambios a los 3-intercambios: recolocaciones de cadenas¹ de uno, dos o tres puntos consecutivos entre otros dos. Nótese que con estos intercambios no se modifica el sentido de los diferentes tramos. En nuestro caso se seguirá la misma idea, pero sólo se considerarán recolocaciones hacia adelante y no se pondrá límite al tamaño de la cadena a recolocar (salvo el determinado por el tamaño del problema); además se debe chequear la factibilidad de cada posible recolocación respecto a las restricciones del problema. A continuación en la figura 1 se ilustra la recolocación de una cadena de k elementos comenzando en i entre j y $j+1$.

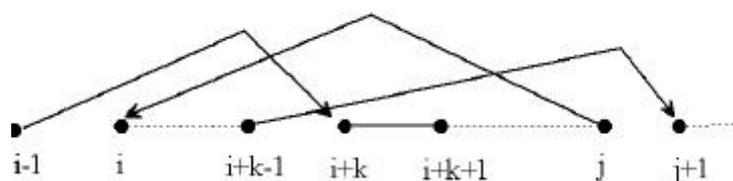


Figura 1.- Recolocación de una cadena de k elementos.

Para cada ruta r se va a denotar por $N_1^k(r)$ el conjunto de rutas factibles obtenidas por recolocaciones hacia adelante de cadenas de a lo sumo k elementos

¹En este trabajo se denomina *cadena* a toda secuencia de puntos consecutivos en la solución actual.

en r . Se denota por $N_1^\infty(r)$ el conjunto de rutas factibles obtenidas por todas las relocalaciones posibles. Análogamente se denota por $N_1^{*k}(r)$ y $N_1^{*\infty}(r)$ el conjunto de rutas factibles e infactibles obtenidas por relocalaciones hacia delante de cadenas de k elementos en el primer caso, y de todas las relocalaciones posibles en el segundo. (Se ha observado empíricamente que los resultados no varían apenas al considerar solamente relocalaciones hacia delante, y se gana en tiempo de computación y eficiencia).

3 - VECINDARIOS TIPO GENDREAU-CLARKE

El segundo tipo de vecindarios considera las soluciones como rutas disjuntas. Se consideran tres formas de intercambio entre dos rutas diferentes:

- **Tipo I:** Intercambio de un elemento i de la ruta r con otro elemento i' de la ruta r' .

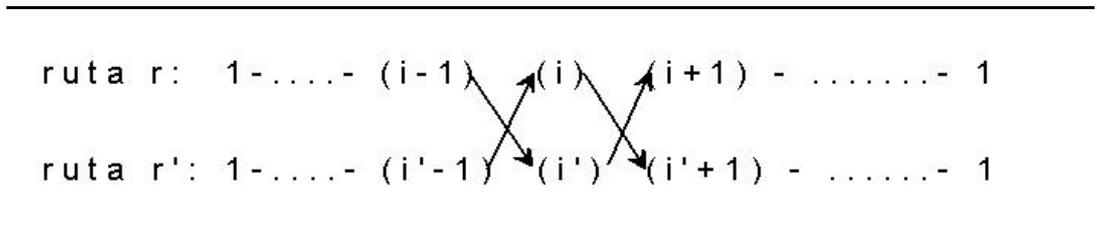


Figura 2.- Eliminación de los arcos (i-1, i), (i, i+1), (i'-1, i'), (i', i'+1) e incorporación de los arcos (i-1, i'), (i', i+1), (i'-1, i), (i, i'+1).

- **Tipo II:** Inserción de un elemento i de la ruta r entre los elementos i' e $i'+1$ de la ruta r' .

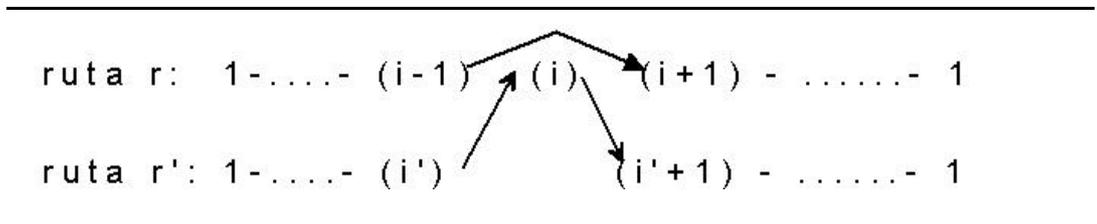


Figura 3.- Eliminación de los arcos (i-1, i), (i, i+1), (i', i'+1) e incorporación de los arcos (i', i), (i, i'+1), (i-1, i+1)

- **Tipo III:** Cruce de las rutas r y r' por los elementos i e i' según la figura 4.

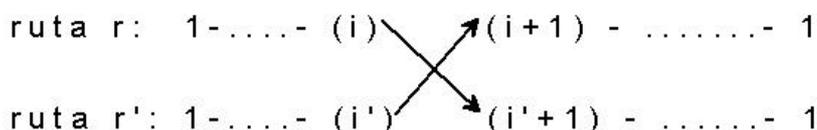


Figura 4.- Eliminación de los arcos (i, i+1), (i', i'+1) e incorporación de los arcos (i', i+1), (i, i'+1).

Los dos primeros tipos aparecen en el trabajo de Gendreu y otros (1.991) y en otros trabajos mencionados anteriormente; el tercero se basa algunas de las ideas propuestas por Clarke y Wright (1.964).

Obsérvese que el tipo I es en realidad un 4-intercambio, el tipo II un 3-intercambio y el tipo III un 2-intercambio. Para cada $s \in S$ se denotará a los conjuntos de soluciones factibles generadas por cada uno de estos tres tipos de intercambios como $N^1_2(s)$, $N^2_2(s)$, $N^3_2(s)$ respectivamente; así mismo, se denota por $N_2(s)$ a la unión de estos tres conjuntos. Análogamente se denota por $N^{*1}_2(s)$, $N^{*2}_2(s)$ y $N^{*3}_2(s)$ los conjuntos de soluciones factibles y no factibles, generadas por cada uno de estos tres tipos de intercambios y $N^*_2(s)$ a la unión de estos tres conjuntos.

4 - FACTIBILIDAD Y VALORACIÓN DE CADA INTERCAMBIO

El chequeo de la factibilidad y la valoración de cada intercambio puede suponer un tiempo de computación excesivo. Valorar un cambio en el tipo de problema que se está estudiando, no sólo consiste en calcular la diferencia entre los costes de los arcos que se añaden y los que se eliminan: hay que determinar también y tener en cuenta los nuevos tipos de vehículos requeridos. En los trabajos de Pacheco y Delgado (1.996) y (1.997) se propone el uso de variables globales, con las que el

número de operaciones para chequear y evaluar cada intercambio es constante, es decir, independiente del tamaño del problema.

4.1.-Valoración y Factibilidad respecto a la Carga.

Sea una ruta de nr puntos $r(1) - r(2) - \dots - r(nr-1) - r(nr)$, siendo $r(nr) = r(1) = 1$. Para calcular los espacios requeridos tras cada intercambio se define:

- $esp_ocup(s)$ como la carga en el vehículo después de visitar el punto $r(s)$, para $s = 2, \dots, nr$;

así mismo sea :

- $cadena = r(p)-r(p+1)-\dots-r(q)$, cualquier cadena de la solución actual,

se define:

- $maximo_esp_ocup(p, q)$ como la máxima carga en el vehículo durante la visita a los puntos de $cadena$;

$$maximo_esp_ocup(p, q) = \max_{s=p, \dots, q} esp_ocup(s)$$

se tiene que:

- $maximo_esp_ocup(p, q) = \max \{maximo_esp_ocup(p, q-1), esp_ocup(q)\}$, con
- $maximo_esp_ocup(p, p) = esp_ocup(p)$ como valor inicial.

4.2.-Factibilidad respecto a las ventanas de tiempo.

Para todo $s = 2, \dots, nr$ se van a definir las siguientes variables, inspiradas en el trabajo de Savelsberg (1.985):

- $A(s)$: Tiempo de llegada del vehículo a $r(s)$,
- $D(s)$: Tiempo de salida de $r(s)$; se tiene que $D(s) = \max\{A(s), e_{r(s)}\}$;
- $M(s)$: Margen de tiempo en la llegada a $r(s)$, es decir, $M(s) = l_{r(s)} - A(s)$;

- $E(s)$: Tiempo de espera del vehículo en $r(s)$, es decir, $E(s) = \max\{e_{r(s)} - A(s), 0\}$.

A continuación se muestra un gráfico con un ejemplo que ilustra estas definiciones.

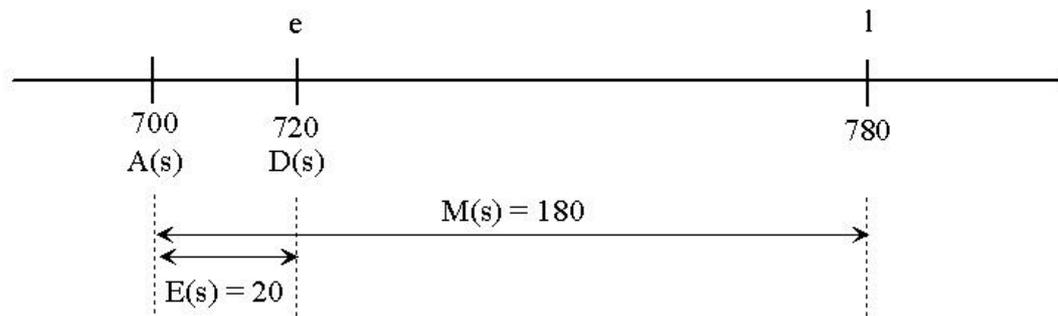


Figura 5.- Ejemplo de la llegada del vehículo a un punto $r(s)$, con $e_{r(s)}=720$ y $l_{r(s)}=780$;

Si la llegada $A(s)$ se produce en 700, entonces el tiempo de salida es $D(s) = 720$, el tiempo de espera $E(s)$, es 20, y el margen, $M(s)$, es 80.

Proposición 1

Para cualquier $p = 1, \dots, nr$, considérese un retardo en la llegada a $r(p)$ de x unidades de tiempo; sea A^* y D^* los nuevos tiempos de llegada y salida que este incremento produce en los puntos siguientes de la solución, se tiene que:

$$A^*(p+h) = A(p+h) + \max\{x - \sum_{s=p, \dots, p+h-1} E(s), 0\}, \quad \text{para } h = 0, \dots, nr-p.$$

Demostración

Por inducción en los valores de h :

- para $h = 0$ es trivial: $A^*(p) = A(p) + x$.
- sea cierto para $h-1$, entonces,

$$A^*(p+h-1) = A(p+h-1) + \max\{x - \sum_{s=p, \dots, p+h-2} E(s), 0\}.$$

Observando la figura 5 anterior y teniendo presente las relaciones entre A, D y E, es fácil de comprobar que un retardo en la llegada a un punto repercute en el tiempo de salida en dicho punto, si este retardo es mayor o igual que el tiempo de espera; en este caso el aumento en el tiempo de salida será igual a la diferencia entre el retardo y el tiempo de espera; por tanto:

$$\begin{aligned} D^*(p+h-1) &= D(p+h-1) + \max\{\max\{x - \sum_{s=p, \dots, p+h-2} E(s), 0\} - E(p+h-1), 0\} \\ &= D(p+h-1) + \max\{x - \sum_{s=p, \dots, p+h-1} E(s), 0\} \end{aligned}$$

luego:

$$A^*(p+h) = D^*(p+h-1) + t_{r(p+h-1) r(p+h)} = A(p+h) + \max\{x - \sum_{s=p, \dots, p+h-1} E(s), 0\}$$

Corolario 1

Sea $H(p, q)$, siendo $p = 1, \dots, nr$, y $q = p, \dots, nr$, el máximo retardo en la llegada a $r(p)$, es decir, incremento en $A(p)$ que no produce violación de las ventanas de tiempo en $r(q)$ (se mantiene $A^*(q) \leq l_{r(q)}$); se tiene que:

$$H(p, q) = \sum_{s=p, \dots, q-1} E(s) + M(q)$$

Sea :

$$- \text{cadena} = r(p) - r(p+1) - \dots - r(q),$$

se define:

- $\text{maxretraso}(p, q)$ = Máximo retardo en la llegada a $r(p)$ que no produce violaciones de las ventanas de tiempo en los puntos de *cadena*;
- $\text{tespera}(p, q)$ = Tiempo total de espera acumulado en *cadena*;

obviamente:

$$\begin{aligned} \text{tespera}(p, q) &= \sum_{s=p, \dots, p+h-1} E(s) \text{ y} \\ \text{tespera}(p, q) &= \text{tespera}(p, q-1) + E(q); \end{aligned}$$

por comodidad si $q < p$ (si *cadena* no contiene elementos), se define

$$tespera(p, q) = 0.$$

Proposición 2

Se tiene que los valores de *maxretraso* se pueden obtener de la forma siguiente:

$$maxretraso(p, q) = \min_{s=p\dots q} H(p, s) = \min_{s=p\dots q} \{M(s) + tespera(p, s-1)\}$$

Demostración

Obvio a partir de Proposición 1 y Corolario 1 y la definición de *maxretraso*.

Un método recursivo para obtener el valor de *maxretraso* viene dado por la siguiente fórmula:

$$maxretraso(p, q) = \min\{maxretraso(p, q-1), M(q) + tespera(p, q-1)\}$$

con $maxretraso(p, p) = M(p)$ como valor inicial.

Así mismo se define:

- $maxadelanto(p, q) =$ Máximo adelanto en la llegada a $r(p)$, es decir disminución de $A(p)$, que no produce tiempo de espera en dicha cadena (es decir, se mantiene $A(s) \geq e_{r(s)}$ para $s = p, \dots, q$); obviamente si en algún punto de *cadena* ya existe espera *maxadelanto (cadena)* es 0.

Proposición 3

Los valores de *máximoadelanto* puede obtenerse de la siguiente forma:

$$maxadelanto(p, q) = \min_{s=p\dots q} \{\max[0, A(s) - e_{r(s)}]\}$$

A partir de aquí se puede obtener un método recursivo para su cálculo:

$$maxadelanto(p, q) = \min\{maxadelanto(p, q-1), \max[0, A(q) - e_{r(q)}]\}$$

con $maxadelanto(p, p) = \max[0, A(p) - e_{r(p)}]$ como valor inicial.

Estas variables son importantes a la hora de determinar como afecta la alteración en el tiempo de llegada al primer punto de una cadena, al tiempo de salida del último punto de esa cadena y a la posible *violación* o no de las ventanas de tiempo en los puntos de la cadena. Concretamente, sea para una determinada cadena $r(p)-r(p+1)-\dots-r(q)$, *adelanto_inicial* la cantidad de tiempo en que se adelanta la llegada a $r(p)$, y *adelanto_final* la cantidad de tiempo en que se adelanta la salida de $r(q)$, se tiene que:

$$adelanto_final = \min \{adelanto_inicial, maxadelanto\}.$$

Análogamente sean *retraso_inicial* y *retraso_final* respectivamente los atrasos en los tiempos de llegada a $r(p)$ y salida de $r(q)$, se tiene que:

$$retraso_final = \max \{retraso_inicial - \sum_{s=p,\dots,q} E(s), 0\}.$$

Las variables globales definidas en este apartado pueden ser calculadas con orden $\theta(n^2)$ de operaciones, facilitando posteriormente el chequeo de la factibilidad y valoración de los intercambios.

5 - ALGORITMO BÚSQUEDA TABÚ

Como se ha comentado en la introducción el algoritmo propuesto en este trabajo es un Metaheurístico basado en un proceso de Búsqueda Tabú.

La *Búsqueda Tabú* es una estrategia dada a conocer en los trabajos de Glover (1.989) y (1.990), que esta teniendo grandes éxitos y mucha aceptación en los últimos años. Según su creador, es un procedimiento que "*explora el espacio de soluciones más allá del óptimo local*" (Glover y Laguna (1.993)). Se permiten movimientos *hacia arriba* o que empeoran la solución una vez que se llega a un

óptimo local. Simultáneamente los últimos movimientos realizados se califican como *tabús* durante las siguientes iteraciones para evitar volver a soluciones anteriores y que el algoritmo cicle. El termino *tabú* hace referencia a "un tipo de inhibición a algo debido a connotaciones culturales o históricas y que puede ser superada en determinadas condiciones...". (Glover (1.996)). Recientes y amplios tutoriales sobre Búsqueda Tabú que incluyen todo tipo de aplicaciones pueden encontrarse en Glover y Laguna (1.997) y (1.999).

5.1.-Movimientos vecinales.

Sea rf una ruta cualquiera factible y $k \in \mathbb{N}$ un número natural, se definen los siguientes procedimientos:

☐ Procedimiento Búsqueda Local Or(k, rf);

Repetir

Determinar $r' \in N_{1}^k(rf)$ verificando $f(r') = \min\{f(r) / r \in N_{1}^k(rf)\}$

Si $f(r') < f(rf)$ entonces hacer $rf = r'$

Hasta que $f(r) \geq f(rf), \forall r \in N_{1}^k(rf)$.

☐ Procedimiento Búsqueda Local Ge(k, sf)

Repetir

Determinar $s' \in N_2(sf)$ verificando $f(s') = \min\{f(s) / s \in N_2(sf)\}$

Si $f(s') < f(sf)$: - Ejecutar Búsqueda_Local_Or(3, r') y

Búsqueda_Local_Or(3, r'') donde r' y r'' son las dos rutas de sf modificadas para dar lugar a s'

- Hacer $sf = s'$

Hasta que $f(s) \geq f(sf), \forall s \in N_2(sf)$.

5.2.-El Algoritmo Básico.

El algoritmo que se propone actúa básicamente de la forma siguiente:

☐ Procedimiento Búsqueda Tabú Básico(k)

Leer solución inicial sf y hacer $s^ = sf$;* (3)

Hacer $T = \emptyset$, $niter = 0$, $kiter = 0$ (4)

Repetir

- $niter := niter + 1$; (1)

- Seleccionar $s \in N_2(sf) / s \notin T$ o $f(s) < f(s^)$ (criterio de 'aspiración') con $f(s)$ mínimo*

-Hacer $sf = s$

-Ejecutar Búsqueda_Local_Or(3, r') y Búsqueda_Local_Or(3, r'') donde r' y r'' son las dos rutas de sf modificadas (5)

-Si $f(sf) < f(s^)$ entonces: hacer $s^* = sf$ y $kiter = niter$*

-Actualizar T (2)

Hasta $niter - kiter \geq maxiter$

Se denota por s^* a la mejor solución encontrada, y por sf a la solución actual en cada momento. T es el conjunto de movimientos tabús y se obtiene determinando que conjunto de soluciones tienen ciertos *atributos tabús activos*. Por tanto se han de definir estos atributos tabús y durante cuantas iteraciones van a permanecer activos (y por tanto las soluciones que les contienen).

Por otra parte, cualquier movimiento, supone la incorporación de un conjunto de arcos y la eliminación de otros. Para evitar volver a soluciones anteriores se impiden los movimientos que supongan la incorporación de arcos recientemente eliminados. En otras palabras, los *atributos tabús* van a ser los arcos que componen cada solución, y un movimiento es tabú si supone la incorporación de algún arco

eliminado en iteraciones recientes (*atributo tabú activo*). La definición de atributo tabú activo sólo se aplica a los arcos eliminados para pasar de una solución a otra, pero no se aplican a los arcos eliminados cuando se mejora la solución mediante la ejecución de *Busqueda_Local_Or* para r' y r'' .

Para identificar que atributos tabús (arcos) van a estar activos se define *arco_tabú* como una matriz $n \cdot n$ de la siguiente forma:

$$\text{arco_tabú}(r, l) = n^\circ \text{ de la última iteración en la que fue eliminado el arco } (r, l)$$

Un determinado arco (r, l) será un atributo tabú activo si:

$$niter - \text{arco_tabú}(r, l) < \text{maxiter_tabu}$$

siendo *maxiter_tabu* el número de iteraciones que permanece activo como atributo tabú desde que es eliminado de la ruta actual.

Inicialmente se define: $\text{arco_tabú}(r, l) = 0$, para cada arco (r, l) en la solución inicial; y $\text{arco_tabú}(r, l) = -\text{maxiter_tabu}$ (o un valor más negativo) para el resto. De esta forma se impide que en las primeras iteraciones sean declarados *tabús activos* arcos que no formen parte de la solución actual. Así inicialmente se asegura que $T = \emptyset$.

Se asigna al parámetro *maxiter_tabu* el valor de $\text{maxiter_tabu} = 2 \cdot n^{1/2}$. Para el criterio de parada se establece $\text{maxiter} = 10 \cdot n$.

El algoritmo de Búsqueda Tabú descrito aporta mejores soluciones cuando se incorporan procedimientos basados en lo que se denomina habitualmente memoria a *largo y medio plazo* como *Diversificación e Intensificación*; también se puede enriquecer con la posibilidad de visitar, de forma controlada, soluciones infactibles por medio de funciones de penalización (*Oscilación Estratégica*).

5.3.-Fase de Intensificación.

La intensificación consiste en regresar a regiones “atractivas” ya exploradas para estudiarlas más a fondo. En esta fase se intensifica la exploración de las regiones donde se hallan las mejores soluciones encontradas en fase inicial (Algoritmo Básico) con la esperanza de encontrar soluciones aún mejores. Puede ser diseñada de diferentes formas; en este caso su diseño está inspirado en los trabajos de Rossing (1.997) y Rossing y Revelle (1.997) y Rossing y otros (1.998) sobre *Concentración Heurística*; esta estrategia encuentra soluciones a problemas combinatorios en dos fases: en la primera se ejecuta varias veces un procedimiento de Búsqueda Local, registrándose los mejores óptimos locales obtenidos; en la segunda se construye un *Conjunto de Concentración*, CS, con los elementos que aparecen en las mejores soluciones obtenidas en la primera fase, y se ejecuta un algoritmo exacto o heurístico, pero utilizando sólo los elementos de CS.

En este trabajo se consideran como elementos de las soluciones, los arcos que las componen; por ejemplo la solución dada por las dos rutas:

ruta 1: 1 - 3 - 5 - 1 y *ruta 2:* 1 - 4 - 2 - 1

que puede expresarse como la secuencia:

1 - 3 - 5 - 1 - 4 - 2 - 1,

estará formada por los arcos (1,3), (3,5), (5,1), (1,4), (4,2) y (2,1).

El Conjunto de Concentración estará constituido por aquellos arcos (sin repetir) que aparezcan en las mejores soluciones. El procedimiento de construcción es el siguiente:

☐ Procedimiento Construcción Conjunto de Concentración:

Ordenar las soluciones obtenidas hasta el momento en una lista según el valor de f (comenzando con la mejor)

Hacer $i = 0$ y $CS = \emptyset$

Repetir

Hacer $i = i + 1$

Añadir los elementos de la i -ésima solución de la lista a CS

Hasta $\text{Cardinal}(CS) \geq \text{num_arcos}$

A diferencia de la idea propuesta por Rossing, no se fija un número predeterminado de soluciones cuyos elementos se introducen, sino que se fija un número mínimo de elementos a introducir num_arcos .

A continuación se diseña un procedimiento que *concentre* la búsqueda de elementos en CS. Inicialmente se define la matriz de distancias auxiliar $d1$ de la siguiente forma:

$$\begin{array}{ll} d1(i, j) = d(i, j) & \text{si } (i, j) \in CS \\ d1(i, j) = d(i, j) + 10 \cdot \max_d & \text{si } (i, j) \notin CS \end{array}$$

donde $\max_d = \max\{d(i, j) \mid i, j = 1, \dots, n\}$; además se define $f1(s)$ como el valor de la función objetivo considerando el coste de los arcos el dado por $d1$. Se propone el siguiente procedimiento de búsqueda que tiene en cuenta ambas matrices d y $d1$:

☐ Procedimiento Búsqueda Local hacia CS (k, sf)

Hacer $s1 = sf$

Repetir

Hacer $\text{coste_anterior} = f(sf)$

Repetir

Buscar $s' \in N_2(s1) \mid f1(s') = \min\{f1(s) \mid s \in N_2(s1)\}$

En s' Ejecutar $Búsqueda_Local_Or(k, r')$ y $Búsqueda_Local_Or(k, r'')$
(considerando $f1$ en vez de f) donde r' y r'' son las dos rutas de $s1$
modificadas para dar lugar a s'

Si $f1(s') < f1(s1)$ entonces $s1 = s'$

Buscar $s'' \in N_2(s1) / f(s'') = \min\{f(s) / s \in N_2(s1)\}$

En s'' Ejecutar $Búsqueda_Local_Or(k, r')$ y $Búsqueda_Local_Or(k, r'')$
donde r' y r'' son las dos rutas de $s1$ modificadas para dar lugar a s''

Si $f(s'') < f(sf)$ entonces $sf = s''$

Hasta $f1(s') \geq f1(s1)$

Hacer $s1 = sf$

Hasta $f(sf) = coste_anterior$

Se trata de un procedimiento de búsqueda local anidado; en cada paso la solución actual $s1$ se sustituye por otra mejor según $f1$, es decir según $d1$ y buscando por tanto soluciones que contengan elementos de CS; cuando no hay mejora en $f1$, se sustituye $s1$ por sf , la mejor solución según f observada en los vecindarios explorados y se reinicia la búsqueda local. El proceso acaba cuando no hay mejora en $f(sf)$.

En definitiva es un procedimiento de búsqueda “conducido” por $f1$, es decir por $d1$, hacia soluciones que contengan el mayor número de elementos de CS posibles. Obsérvese que $d1$ “penaliza” a los arcos no pertenecientes a CS, pero no “impide” su elección en cada paso, ya que esto podría hacer excesivamente reducido el número de soluciones a considerar y “encajonar” el proceso. Este procedimiento se inserta en la fase de Intensificación que queda de la siguiente forma:

☐ Procedimiento Intensificación (k)

Ejecutar Construcción_Conjunto_de_Concentracion;

Desde $i:=1$ hasta $num_soluciones$ hacer

Tomar s_i la i -ésima solución de la lista ordenada obtenida en la última ejecución de *Búsqueda_Tabú_Básico*

Ejecutar *Búsqueda_Local_Ge*(k, s_i)

Para cada ruta r de s_i ejecutar *Busqueda_Local_Or*(∞, r)

{obviamente si $k = \infty$ este paso no es necesario)

Ejecutar *Búsqueda_Local_hacia_CS*(k, s_i);

Si $f(s_i) < f(s^*)$ hacer $s^* = s_i$

Se toma f_1 en vez de f y se ejecuta el procedimiento de *Búsqueda_Local_hacia_CS* dirigido por f_1 , en vez de la búsqueda local habitual. En definitiva, se “concentra” o intensifica la búsqueda de soluciones en las regiones con elementos de CS. En cierto modo, se puede considerar como una forma de *reencadenamiento de trayectorias*.

Se ha asignado $num_soluciones = 50$. En cuanto al valor del parámetro num_arcos , en el trabajo de Pacheco y Delgado (2.000) se describen los resultados de diferentes experiencias que aconsejan tomar num_arcos como el 10% del total de arcos (es decir, $num_arcos = 0.1 \cdot n \cdot (n-1)$).

5.4.-Fase de Diversificación

Una vez finalizada la fase de Intensificación, se ejecuta la de Diversificación, cuyo objetivo es dirigir el proceso de búsqueda a regiones no exploradas hasta el momento. Básicamente, esto se hace “penalizando” los elementos más visitados durante el proceso. Se van a definir dos matrices de frecuencias:

- $freq_corto(i, j)$ = número de veces que ha aparecido el elemento (i, j) durante la última ejecución de *Búsqueda_Tabú_Básico* e *Intensificación*.
- $freq_largo(i, j)$ = número de veces que ha aparecido el elemento (i, j) hasta el momento durante todo el proceso.

Durante una serie de iteraciones se van a realizar movimientos vecinales considerando estas matrices de frecuencia como matrices de distancias y teniendo en cuenta, como en el procedimiento básico, la calificación de atributos tabú para evitar ciclos. Más concretamente, se define:

- $freq(i, j) = freq_corto(i, j)$, si en la última ejecución de *Búsqueda_Tabú_Básico* o de *Intensificación* ha habido mejora en $f(s^*)$;
- $freq(i, j) = freq_largo(i, j)$, en caso contrario;

y a continuación:

- $d2(i, j) = freq(i, j)$ si $(i, j) \notin T$
- $d2(i, j) = \infty$ si $(i, j) \in T$

y como en la fase básica, se define la función $f2$ como la función de costes f pero considerando el coste de los arcos el dado por $d2$ en vez de por d . De esta forma la fase de diversificación queda como sigue:

☐ Procedimiento Diversificación (k)

Hacer sf como la última solución obtenida en la fase de intensificación;

$T = \emptyset$,

Ejecutar $maxiter_div$ veces las iteraciones del procedimiento Búsqueda_Tabú_Básico (línea (1) a línea (2)) considerando la función f_2 en vez de la función f

En este caso se ha tomado $maxiter_div = 5 \cdot n$. Al añadir las fases de Intensificación y Diversificación el Algoritmo Búsqueda Tabú queda de la siguiente forma:

☐ Procedimiento Búsqueda Tabú Principal(k):

Leer solución inicial sf y hacer $s^* = sf$ (7)

Hacer: $iter_global = 0, mejor_iter = 0$

Repetir

Hacer $iter_global = iter_global + 1; costeanterior = f(s^*)$

Ejecutar Búsqueda_Tabú_Básico(k)

Ejecutar Intensificación(k)

Ejecutar Diversificación (k)

Si $f(s^*) < costeanterior$ entonces hacer $mejor_iter = iter_global$

Hasta $iter_global - mejor_iter \geq maxiter_global$

La introducción de la línea (7) en este procedimiento hace innecesaria la línea (3) en el procedimiento Búsqueda_Tabú_Básico. El criterio de parada se alcanza cuando transcurren una serie de iteraciones globales (Búsqueda_Tabú_Básico + Intensificación + Diversificación) sin que se produzca mejorara en el valor de $f(s^*)$.

5.5.- Oscilación Estratégica.

Opcionalmente se puede permitir la visita de soluciones no factibles, de forma controlada, para dar más flexibilidad al proceso y no dejar que quede atrapado en una determinada región en torno a un mínimo local, facilitando el acceso a otras regiones.

En el algoritmo propuesto esta opción se da en la fase Básica pero no en las otras dos. Para ello se define la función g de la siguiente forma:

$$g(s) = g(s) + coef_carga \cdot QS(s) + coef_tiempo \cdot TS(s), \quad \text{para } s \in S^*$$

donde:

- S^* = conjunto de soluciones factibles y no factibles (con respecto a la capacidad de los vehículos y las ventanas de tiempo) del problema, obviamente $S \subset S^*$;
- $QS(s)$ = suma, en el conjunto de las rutas de s , de la cantidad de mercancía en que cada ruta de s sobrepasa la capacidad del vehículo (de máxima capacidad) en s ;
- $TS(s)$ = suma, en el conjunto de las rutas de s , del máximo retraso de tiempo que se da en cada ruta de s ;
- $coef_carga$ y $coef_tiempo$ = coeficientes de penalización para la violación de las restricciones de carga y tiempo respectivamente;

Esta función sustituye a la función f en el procedimiento Búsqueda_Tabú_Básico. Así mismo, en dicho procedimiento se deben sustituir N_2 por N^*_2 . Igualmente se reemplaza N^k_1 por N^{*k}_1 durante la ejecución de Búsqueda_Local_Or en la línea (5).

Para controlar la visita a soluciones no factibles se van a seguir dos criterios:

1. Si la solución actual sf es infactible, sólo se permite cambio a una solución que reduzca su infactibilidad. Más concretamente, se deben añadir las siguientes condiciones a las soluciones s para la búsqueda de sI :

$$si \ [(QS(sf) = 0 \text{ o } QS(s) < QS(sf)) \quad y \quad (TS(sf) = 0 \text{ o } TS(s) < TS(sf))]$$

2. Variar el valor de $coef_carga$ y $coef_tiempo$ cada 10 iteraciones de la siguiente forma:

- definir $ncarga$ y $ntiempo$ respectivamente como número de iteraciones en que $QS(sf) > 0$ y $TS(sf) > 0$ en esas 10 últimas iteraciones;
- si $ncarga > 0$ hacer $coef_carga = coef_carga \cdot 2$,
- si $ncarga = 0$ hacer $coef_carga = coef_carga/2$;
- de igual forma si $ntiempo > 0$ hacer $coef_tiempo = coef_tiempo \cdot 2$,
- si $ntiempo = 0$ hacer $coef_tiempo = coef_tiempo/2$;
- inicialmente a $coef_carga$ y $coef_tiempo$ se los asigna el valor 1.

La idea de variar los coeficientes de penalización ha sido tomada del trabajo de Gendreau y otros (1.991). El objeto de estas estrategias es mantener el proceso “cercano” a la región factible S . Cuando el proceso lleva varias iteraciones explorando fuera de S , se aumentan los coeficientes de penalización, obliagnado al proceso a volver a regiones factibles. Por otra parte, si el proceso lleva varias iteraciones en S , se favorece que pueda visitar regiones infactibles disminuyendo estos coeficientes.

Una característica de este método es la preferencia por soluciones factibles o por aquellas que disminuyan la infactibilidad a la hora de explorar vecindarios de soluciones infactibles. Se han realizado pruebas con estrategias en las que no se da preferencia a las soluciones factibles, ni al grado de infactibilidad, excepto la que lleva implícita en la función penalizada; un interesante trabajo en este sentido es el de Díaz y Fernández (1.998) para el Problema de Asignación Lineal Generalizada.

El inconveniente que surge al aplicar estas estrategias al VRPTW es que además de restricciones de carga hay restricciones de tiempo, y la proporción de soluciones factibles visitadas en el conjunto de iteraciones es bajo. Por eso se opta por “forzar” aún más al proceso para que busque soluciones en S.

6 - RESULTADOS COMPUTACIONALES

Para chequear la eficacia del algoritmo diseñado en este trabajo y en concreto la eficacia de la fase de Intensificación propuesta y de la Oscilación Estratégica, se han utilizado dos conocidas librerías de instancias: TSPLIB de Reinelt (1.991) para el VRP, y Solomon Instances para el VRPTW. Cada una de estas instancias se ha resuelto empleando 4 variantes del algoritmo propuesto. Estas variantes son el resultado de combinar la utilización o no, por una parte de la fase de Intensificación y por otra de la Oscilación Estratégica.

En todos los casos se ha usado como número de iteraciones sin mejora $maxiter_global = 4$, y el valor del parámetro $k = \infty$. La solución inicial viene dada por un algoritmo de *inserción de máxima diferencia* (Golden y otros (1.980)). Los algoritmos se han programado en PASCAL, utilizando los compiladores BORLAND PASCAL 7.0 y BORLAND DELPHI 3.0. A continuación se muestran los resultados obtenidos.

6.1.-TSPLIB/VRP.

Esta librería puede encontrarse en una página mantenida por G. Reinelt (1.997), http://www.iwr.uni_heildeberg.de/iwr/comopt/soft/TSPLIB95/CVRP. Se muestran los resultados de las 4 variantes (número de vehículos y distancia), su tiempo de computación y la mejor solución encontrada hasta la fecha. Según la descripción dada en la página se trata de minimizar la distancia, independientemente del número de vehículos. Se usan distancias euclídeas redondeadas a enteros, excepto

para *eil7*, *eil13* y *eil31* que viene dada explícitamente. El ordenador usado en este caso es un PC Pentium - MMX 200 Mhz.

En el cuadro siguiente se resumen los resultados obtenidos por cada una de las 4 variantes: número de vehículos distancia y tiempos de computación; así como la mejor solución encontrada hasta la fecha.

<i>Instancia</i>	Sin Intensificación		Con Intensificación		<i>Previa Mejor Solución</i>
	Sin Oscilación	Con Oscilación	Sin Oscilación	Con Oscilación	
Eil7	2 - 104 (*) 0"02	2 - 104 (*) 0"02	2 - 104 (*) 0"02	2 - 104 (*) 0"02	2 - 104
Eil13	4 - 247 (*) 0"27	4 - 247 (*) 0"32	4 - 247 (*) 0"26	4 - 247 (*) 0"32	4 - 247
Eil22	4 - 375 (*) 7"56	4 - 375 (*) 4"23	4 - 375 (*) 2"72	4 - 375 (*) 4"29	4 - 375
Eil23	3 - 569 (*) 1"81	3 - 569 (*) 2"01	3 - 569 (*) 1"81	3 - 569 (*) 2"01	3 - 569
Eil30	4 - 503 (**) 16"98	4 - 503 (**) 6"81	4 - 503 (**) 52"16	4 - 503 (**) 6"80	3 - 534
Eil31	7 - 379 (**) 49"61	7 - 388 (**) 95"23	7 - 379 (**) 100"90	7 - 379 (**) 147"11	7 - 1212
Eil33	4 - 835 (*) 35"81	4 - 835 (*) 9"12	4 - 835 (*) 18"10	4 - 835 (*) 9"12	4 - 835
Eil51	5 - 521 (*) 150"22	5 - 521 (*) 30"18	5 - 521 (*) 71"49	5 - 521 (*) 30"37	5 - 521
EilA76	10 - 832 (*) 1237"29	10 - 836 1275"74	10 - 836 1205"91	10 - 831 (**) 3766"89	10 - 832
EilB76	15 - 1032 110"86	15 - 1026 (**) 2895"47	14 - 1023 (**) 1618"56	15 - 1030 (**) 865"19	15 - 1031
EilC76	8 - 739 637"08	8 - 737 2342"22	8 - 736 865"69	8 - 735 (*) 1990"69	8 - 735
EilD76	7 - 688 524"03	7 - 686 788"52	7 - 682 (**) 3369"25	7 - 682 (**) 1502"38	7 - 683

<i>Instancia</i>	Sin Intensificación		Con Intensificación		<i>Previa Mejor Solución</i>
	Sin Oscilación	Con Oscilación	Sin Oscilación	Con Oscilación	
EilA101	8 - 818 4602"75	8 - 818 7305"13	8 - 818 2610"48	8 - 815 (**) 3541"50	8 - 817
EilB101	14 - 1076 (**) 174"69	14 - 1079 1873"77	14 - 1076 (**) 174"71	14 - 1075 (**) 4591"39	14 - 1077

Tabla 1

(*) *Iguala la actual mejor solución;* (**) *Mejora la mejor solución previa;* **nueva mejor solución en negrita.**

Llama la atención el resultado que aparece en la instancia *eil31* en cuanto a la distancia recorrida. Se escribió un correo electrónico a la dirección de TSPLIB/VRP del que estamos esperando respuesta.

6.2.- Instancias de SOLOMON para el VRPTW.

Esta librería se ha obtenido a través de la página WEB de *APES Group* de la Universidad de StrathClyde: <http://www.cs.strath.ac.uk/~apes>. Estas instancias son como las usadas por Solomon (1.987). En total son 56 que se dividen en 6 clases: C1, C2, R1, R2, RC1, RC2. El tamaño de todos los problemas es de 100 clientes (más el origen) e incluyen ventanas de tiempo, capacidad máxima, tiempo de descarga. En C1 y C2 los clientes vienen distribuidos en grupos o clusters. En R1 y R2 los puntos se distribuyen aleatoriamente de forma uniforme. RC1 y RC2 mezclan clientes distribuidos de forma aleatoria y por conglomerados. En las clases C1, R1 y CR1 se usan más vehículos que en las clases C2, R2 y CR2. El ordenador usado en este caso es un PC Pentium II a 350 Mhz.

Se trata de minimizar el número de vehículos (primer objetivo) y la distancia (segundo). Se usan distancias euclídeas reales. A continuación se muestran los resultados medios para cada clase, obtenidos por diferentes autores. En negrita aparecen los mejores resultados medios para cada clase en cuanto a número de vehículos:

	C1	R1	RC1	C2	R2	RC2
CR	10.00 885.86	12.42 1289.95	12.38 1455.82	3.00 658.88	2.91 1135.14	3.38 1361.14
PB	10.00 838.01	12.58 1296.80	12.13 1446.20	3.00 589.93	3.00 1117.70	3.38 1360.57
Tha	10.00 832	12.33 1238	12.00 1284	3.00 650	3.00 1005	3.38 1229
RT	10.00 828.38	12.25 1208.50	11.88 1377.39	3.00 589.86	2.91 961.72	3.38 1119.59
TBGGP	10.00 828.38	12.17 1209.35	11.50 1389.22	3.00 589.86	2.82 980.27	3.38 1117.44
KPS	10.00 830.75	12.67 1200.33	12.12 1388.15	3.00 592.24	3.00 966.56	3.38 1133.42
Variante 1	10.00 828.64	12.91 1191.12	12.50 1371.72	3.00 590.69	4.00 898.06	4.38 1043.69
Variante 2	10.00 829.96	12.33 1211.32	12.00 1386.48	3.00 590.29	2.73 981.20	3.38 1118.44
Variante 3	10.00 828.46	12.83 1188.31	12.63 1361.67	3.00 590.29	4.00 893.16	4.38 1041.71
Variante 4	10.00 828.68	12.33 1208.30	12.00 1375.35	3.00 590.36	2.82 960.79	3.38 1112.91

Tabla 2

CR	=	Chiang y Russell, (1.993).
PB	=	Potvin y Bengio (1.994)
Tha	=	Thangiah, Osman y Sun (1.994)
RT	=	Rochat y Taillard (1.995)
TBGGP	=	Taillard, Badeau, Gendreau, Guertin y Potvin (1.997)
KPS	=	Kilby, Prosser y Shaw, (1.997). Trabajo integrado en GREENTRIP Project.

Variante 1 = Método propuesto. Sin Intensificación ni Oscilación estratégica.

Variante 2 = Método propuesto. Sin Intensificación y con Oscilación estratégica.

Variante 3 = Método propuesto. Con Intensificación y sin Oscilación estratégica.

Variante 4 = Método propuesto. Con Intensificación y Oscilación estratégica.

Las instancias mejoradas en este trabajo son las siguientes:

Instancia	Sin Intensificación		Con Intensificación		Previa Mejor Solución Referencia
	Sin Oscilación	Con Oscilación	Sin Oscilación	Con Oscilación	
r201	5 - 1193.54 1465"69	4 - 1256.65 3167"55	5 - 1189.41 1824"24	4 - 1253.26 3464"24	4 - 1254.09 G
r202	5 - 1043.96 1274"38	3 - 1204.50 2705"41	5 - 1041.10 1588"61	3 - 1220.51 1972"65	3 - 1214.28 TBGGP
r204	3 - 763.31 3450"42	2 - 853.86 4850"25	3 - 753.42 2606"05	2 - 853.52 2264"14	2 - 867.33 G
r207	3 - 834.29 2420"68	2 - 950.29 7769"01	3 - 824.59 4505"65	2 - 923.55 6787"15	3 - 814.78 RT
r208	3 - 720.36 6866"90	2 - 740.86 10776"05	3 - 719.05 8103"05	2 - 736.86 4453"97	2 - 738.60 RT
r209(+)	5 - 861.61 1013"55	3 - 923.85 9477"76	5 - 861.61 1196"28	3 - 923.81 1612"67	3 - 855 Tha
rc202	5 - 1130.22 687"66	4 - 1165.44 929"01	5 - 1120.82 1739"05	4 - 1161.79 985"64	4 - 1164.25 TBGGP

Tabla 3

(+) **Considerando sólo distancias reales sin truncar ni redondear.**

7 - CONCLUSIONES Y REFLEXIONES

El análisis de los resultados obtenidos para cada instancia con cada una de las cuatro variantes estudiadas confirma la eficacia tanto de la fase de Intensificación como de la Oscilación Estratégica:

- Es claro que con el uso de Oscilación Estratégica (Variantes 2 y 4) se consiguen mejores resultados que sin ella, especialmente en las Instancias de Solomon. Tan sólo en 2 casos C104 y R109 el mejor resultado de las cuatro variantes no se da en las Variantes 2 o 4. Especialmente esta mejora se observa en la reducción de los vehículos a usar con respecto a las Variantes 1 y 3 que no utilizan Oscilación Estratégica. De hecho, en la mayoría de los casos se consigue obtener el número de vehículos de la mejor solución conocida hasta la fecha (excepto en R109, RC101, RC102, RC104 y RC105).
- En cuanto a los problemas de la librería TSPLIB/CVRP también se dan los mejores resultados con Oscilación Estratégica (excepto para eilB76). En cualquier caso, quizás la diferencia no sea tan clara como con las instancias de Solomon; se consiguen excelentes resultados con la Variante 3 que no usa Oscilación Estratégica. La causa es, probablemente, que al no haber restricciones de tiempo las regiones factibles no están tan aisladas entre sí.
- En cuanto al uso de Intensificación se pueden hacer dos análisis: Variante 1 frente a Variante 3, y Variante 2 frente a Variante 4. En el primer caso está claro la eficacia del uso de la fase de Intensificación: solamente en 2 instancias (RC106 y RC201) se obtienen soluciones, que aunque muy similares, son mejores con la Variante 1 que con la Variante 3 (por el criterio de parada usado).

- En el segundo caso también es claro, aunque menos, la eficacia del uso de la fase de Intensificación. Se consiguen mejores soluciones en 28 casos con la variante 4 que con la 2, y 10 mejores soluciones con la 2 (por el criterio de parada); de estas en solamente en 2 la diferencia se puede considerar relativamente significativa (RC103 y R211).
- Parecido análisis se puede realizar con los problemas de la librería TSPLIB/CVRP: sólo con la instancia eilA76 la Variante 1 consigue mejor solución que la Variante 3.
- También es interesante observar los tiempos de computación usados en cada caso: en muchos problemas con el uso de la fase de Intensificación se obtienen soluciones semejantes o mejores que sin ella, en un tiempo de computación menor. De todas formas, cualquier conclusión en este sentido debe ir acompañado de un análisis más completo de la evolución de los tiempos de computación.

Las conclusiones de la comparación de los resultados obtenidos en este trabajo, con los obtenidos en otros son las siguientes:

- En la librería TSPLIB/CVRP, y centrándonos en las 6 últimas instancias (con un tamaño aceptable y que no han sido resueltas de forma exacta), la conclusión es clara: se mejoran las soluciones anteriores en 5 de las 6 instancias, y se iguala en la otra (eilC76).
- En las clases R1 y RC1 de las instancias de Solomon se consiguen los resultados más pobres comparándolos con las mejores referencias hasta el momento, Rochat and Taillard, (1995), y Taillard, Badeau, Gendreau,

Guertin and Potvin, (1.997); aunque son aceptables si se comparan con las demás referencias.

- En la clase C1 y C2 las 4 variantes analizadas dan resultados muy parecidos entre ellas y con las mejores referencias anteriores.
- En la clase R2 es donde las variantes 2 y 4 consiguen los mejores resultados: se establecen 5 nuevas mejores soluciones: una con la Variante 2, (R202) y cuatro con la Variante 4 (R201, R02, R207 y R208), a las que habría que añadir (R209) si se consideran solo distancias reales sin truncar ni redondear. También ambas superan los mejores resultados medios anteriores en cuanto a número de vehículos (2.73 variante 2) y distancia (960.79 con 2.82 vehículos variante 4).
- En la clase RC2 también se consiguen buenos resultados: se establece una nueva mejor solución con la Variante 4 para RC202, dándose en el resto de las instancias resultados en general muy cercanos a la mejor solución previa. En cuanto a resultados medios, las Variantes 2 y 4 igualan el número de vehículos obtenidos por el resto de las otras referencias (3.38); la distancia de la Variante 3 (1118.44) es similar a la de las mejores referencias obtenidas hasta el momento, Roachat and Taillard, (1.995), y Taillard, Badeau, Gendreau, Guertin and Potvin (1.997), siendo todas mejoradas ligeramente por la Variante 4 (1112.91).

En resumen, se pueden establecer las siguientes conclusiones y reflexiones:

- Los resultados son interesantes: se mejoran los mejores resultados previos en 5 de las 6 instancias de mayor tamaño de TSPLIB/CVRP, y se iguala la otra;

se mejoran en 7 instancias (considerando distancias reales) de Solomon igualándose en 16, y acercándose en la mayoría.

- En cuanto al tiempo de computación, en algunos casos puede parecer aparentemente alto. Sin embargo hay que recordar que se han usado ordenadores personales y no estaciones de trabajo especializadas en cálculo, que sería lo ideal en estos casos.

- Es claro el efecto positivo tanto de la Oscilación Estratégica como la de la fase de Intensificación. Hay que recordar que en la fase de Intensificación sólo se visitan soluciones factibles. En futuros trabajos sería interesante analizar que ocurre si se permite también la visita de soluciones no factibles en esta fase.

- Se tiene previsto en posteriores trabajos el uso de vecindarios mas amplios como el propuesto en Taillard, Badeau, Gendreau, Guertin and Potvin, (1.997), que extienden el propuesto en este trabajo.

- Insistir en el efecto positivo que puede tener el uso de Conjunto de Concentración, con los elementos de las mejores soluciones, para el diseño de una fase de Intensificación en Búsqueda Tabú, como ya se ha expuesto.

- Los experimentos son totalmente reproducibles al no haber componentes aleatorios en los programas y quedan a disposición de las personas interesadas (versión PASCAL). Una descripción del uso del algoritmo descrito a problemas reales de Transporte Escolar puede encontrarse en Pacheco, Aragón y Delgado (1.999) y (2.000).

8 - REFERENCIAS Y BIBLIOGRAFÍA

BACKER (DE) B., FURNON V., KILBY P., PROSSER P. and SHAW P. (1.997). "Solving Vehicle Routing Problems using Constraint Programming and Metaheuristics". *Journal of Heuristics*, vol. nº 1 - 16.

BODIN L. D. and GOLDEN B.L. (1.981): "Classification in Vehicle Routing and Scheduling". *Networks*, vol.11, nº 2, 97-108.

BULLHEIMER B., HARTI R. F. and STRAUSS C. (1.997). Applying the Ant System for the Vehicle Routing Problem. *2nd Metaheuristics International Conference (MIC-97)*, Sophie-Antipolis, France, July 1.997.

CAMPOS V. y MOTA E. (1.995): Metaheurísticos para el CVRP. *XXII Congreso Nacional de Estadística e Investigación Operativa*. Sevilla, Noviembre 1.995.

CHIANG W. C. and RUSSELL R. A. (1.993). "Hybrid Heuristics for the Vehicle Routing Problem with Time Windows". Working Paper, Dpto. Quantitative Methods. University of Tulsa.

CLARKE G. and WRIGHT J. W. (1.964): "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points". *Oper.Res.*, 12, (1.964), 568-581.

DESROCHERS M., DESROSIERS J. and SOLOMON M. M. (1.992). "A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows". *Operations Research* 40, 342-354.

DESROCHERS M., LENSTRA J. K., SAVELSBERGH M. W. P. and SOUMIS F. (1.988): "Vehicle Routing with Time Windows: Optimization and

Approximation". In *Vehicle Routing: Methods and Studies*, (Studies in Management Sciences and Systems, vol.16), eds: GOLDEN B. L. and ASSAD A. A., Nort-Holland, 65-84.

DIAZ J.A. y FERNANDEZ E. (1.998). "A Tabu Search Heuristic for the Generalized Assigment Problem". Document de Recerca, Departament d'Estadística i Investigació Operativa. Secció Informàtica. Universitat Politècnica de Catalunya. DR 98/08

FISHER M. L. y JAIKUMAR R. (1.981). "A Generalized Assignment Heuristic for Vehicle Routing". *Networks*, vol.11, nº 2, 109-124.

GENDREU M., HERTZ A. and LAPORTE G. (1.991): "A Tabu Search Heuristic for Vehicle Routing Problem". Report CRT-777. *Centre de Recherche sur les Transports*. Univ. Montréal.

GENDREU M., HERTZ A. and LAPORTE G. (1.994): "A Tabu Search Heuristic for Vehicle Routing Problem". *Management Sci.* 40 (10), pg.1276-1290.

GLOVER F. (1.989). Tabú Search: Part I. *ORSA Journal on Computing*, Vol 1, pp. 190-206.

GLOVER F. (1.990). Tabú Search: Part II. *ORSA Journal on Computing*, Vol 2, pp. 4-32.

GLOVER F. (1.996). Búsqueda Tabú en *Optimización Heurística y Redes Neuronales*. Adenso Díaz (coordinador). Paraninfo. Madrid. pp. 105-143.

GLOVER F. y LAGUNA M. (1.993). Tabu Search in *Modern Heuristic Techniques for Combinatorial Problems*. C. Reeves, ed., Blackwell Scientific Publishing, pp, 70-141.

GLOVER F. y LAGUNA M. (1997). *Tabu Search*. Kluwer Academic Publishers, Boston.

GLOVER F. y LAGUNA M. (1999). Tabu Search, aparecerá en *Handbook of Applied Optimization*, P.M.Paradalos and M.G.S.Resende (eds). Oxford Academic Press, (1.999)

GOLDEN B., BODIN L., DOYLE T. y STEWART W. Jr. (1980). "Approximate Traveling Salesman Algorithms". *Operations Research*, vol. 28, n° 3, parte II, 649-711.

HAOUARI M., DEJAX P. et DESROCHERS M. (1990): "Les Problèmes de Tournées avec Contraintes des Fenêtres de Temps: L'Etat de l'Art". *Recherche Operationnelle/Operations Research*, vol. 24, n° 3, 217-244.

KILBY P., PROSSER P. and SHAW P. (1997). Guided Local Search for the Vehicle Routing Problem. *2nd Metaheuristics International Conference (MIC-97)*, Sophie-Antipolis, France, July 1.997.

KONTORAVDIS G. and BARD J. F. (1995): "A GRASP for the Vehicle Routing Problem with Time Windows". *ORSA Journal on Computing*, 7: 10-23.

LAPORTE G. (1992): "The Vehicle Routing Problem: An overview of exact and approximate algorithms". *European Journal of Operations Research*, 59, 345-358.

LAPORTE G. and OSMAN I. H. (1995). "Routing Problems: A Bibliography". *Ann.Oper. Res.*, 61, 227-262.

LENSTRA J. K. and RINNOY KAN A. H. G. (1981): "Complexity of Vehicle Routing and Scheduling Problems". *Networks*, vol.11, n° 2, (1981), 221-228.

LIN S. (1965): "Computer Solutions to the Traveling Salesman Problem". *Bell Syst.Tech.Jou.*, vol 44, 2245-2269.

LIN S. y KERNIGHAN B. W. (1973): "An Effective Heuristic Algorithm for the Traveling Salesman Problem". *Operations Research*, vol.20, 498-516.

NURMI K. (1991): "Traveling Salesman Problem Tools for Microcomputers". *Computers & Ops.Res.* Vol. 18, n° 8, 741-749.

OR I. (1976). Traveling Salesman Type Combinatorial Problems y their Relations to the Logistics of Blood Banking. *Ph.Thesis, Dpt. of Industrial Engineering y Management Sciences, Northwestern Univ.*

OSMAN I. H. (1993): "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem". *Annals of Operations Research*, 41, pg. 421-451.

PACHECO J. y DELGADO C. (1996). Adaptación del Algoritmo de Or al VRPTW con Carga y Descarga simultánea. *X Reunión ASEPELT-ESPAÑA* , Albacete, Junio 1996.

PACHECO J. y DELGADO C. (1997). "Problemas de Rutas con Ventanas de tiempo y carga y Descarga simultánea: Diseño de Filtros para algoritmos de intercambio (caso de un sólo vehículo)". *Estudios de Economía Aplicada*, n° 7 , pgs. 79-100.

PACHECO J. y DELGADO C. (1.999). "Diseño de Metaheurísticos híbridos para Problemas de Rutas con Flota Heterogénea: GRASP". *Cuadernos de Estudios Empresariales*. Universidad Complutense de Madrid, nº 9, pgs.173-192.

PACHECO J. y DELGADO C. (2.000). "Diseño de Metaheurísticos híbridos para Problemas de Rutas con Flota Heterogénea: Concentración Heurística". *Estudios de Economía Aplicada*, nº 14, pp. 137-151.

PACHECO J. ARAGÓN,A. y DELGADO C. (1.999). "Diseño de un sistema que facilite soluciones racionales al problema del Transporte Escolar en la provincia de Burgos". *XIII Reunión ASEPELT-España*. Burgos, Junio 1.999.

PACHECO J., ARAGÓN A. y DELGADO C. (2.000). "Diseño de algoritmos para el problema del Transporte Escolar. Aplicación en la provincia de Burgos". *Qüestio*, vol.24, nº 1, pp. 55-82.

POTVIN J. Y. and BENGIO S. (1.994). "A Genetic Approach to the Vehicle Routing Problem with Time Windows". Technical Report CRT-953, *Centre de Recherche sur les Transports*. Univ. Montréal.

POTVIN J. Y., KERVAHUT T., GARCIA B. L. and ROUSSEAU,J.M. (1.993)."A Tabu Search Heuristic for Vehicle Routing Problem with Time Windows". Report CRT-777. *Management Sci.* 40 (10), pg.1276-1290.

REGO C. (1.998): "A Subpath Ejection Method for the Vehicle Routing Problem". *Management Science*, vol.44, nº 10, pg. 1447-1459.

REINELT G. (1.991): "TSPLIB: A Travelling Salesman Problem Library". *ORSA Journal on Computing*, 3, 376-384.

ROCHAT Y. and TAILLARD E. D. (1.995). "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing". *Journal of Heuristics*, 1 (1), 147-167.

ROSING K. E. (1.997). "Heuristic Concentration: An Introduction with Examples". *The Tenth Meeting of the European Chapter on Combinatorial Optimization*. Tenerife. Spain. May, 1.997.

ROSING K. E. and REVELLE C. S. (1.997). "Heuristic Concentration: Two Stage solution Construction". *European Journal of Operational Research* 97, 75-86.

ROSING K. E., REVELLE C. S., ROLLAND E., SCHILLING D.A. and CURRENT J. R. (1.998). "Heuristic Concentration and Tabu Search: A head to head comparison". *European Journal of Operational Research* 104, 93-99.

SAVELSBERGH M. W. P. (1.985). "Local Search for Routing Problems with Time Windows". *Ann.Oper.Res.*, 4, 285-305.

SOLOMON M. M. (1.987). "Algorithms for the Vehicle Routing and Scheduling Problem with Time Windows Constraints". *Operations Research* 35, 254-265.

TAILLARD E., BADEAU P., GENDREU M., GUERTAIN F. and POTVIN, J.Y. (1.995). "A new Neighbourhood structure for the Vehicle Routing Problem with Time Windows". Technical Report CRT-95-66, *Centre de Recherche sur les Transports*. Univ. Montréal.

TAILLARD E., BADEAU P., GENDREU M., GUERTAIN F. and POTVIN J. Y. (1.997). "A Tabu Search heuristic for the Vehicle Routing Problem with Time Windows". *Transportation Science* 31, 170-186.