Paralelización de Algoritmos Multi-objetivo: Aplicación al Transporte de Mercancías

Raúl Baños[1], Julio Ortega, [1] Consolación Gil[2], Antonio Fernández,[2] Francisco de Toro [3]

Resumen— **La mayoría de problemas de optimización del mundo real son multi-objetivo, lo que conlleva que no exista una única solución óptima sino varias que en conjunto optimizan todos los objetivos, y que forman el llamado Frente de Pareto. Sin embargo, la mayoría de instancias reales de este tipo de problemas no se pueden resolver de forma exacta haciendo uso de técnicas deterministas debido a la complejidad del espacio de búsqueda. Bajo estas circunstancias el uso de técnicas aproximadas, como las meta-heurísticas, son estrategias muy útiles, aunque pueden incluso ser mejoradas en combinación con el procesamiento paralelo. En esta investigación analizamos como mejorar el rendimiento de un algoritmo evolutivo multi-objetivo para resolver una variante del problema de enrutado de vehículos con ventanas temporales aplicando computación paralela. Los resultados obtenidos en la resolución de un conjunto de benchmarks comúnmente utilizados en este ámbito muestran que el procesamiento paralelo permite obtener frentes de soluciones no dominadas de mejor calidad que la versión secuencial del algoritmo.**

Palabras clave— **Computación Paralela, Optimización Multi-objetivo, Algoritmos Evolutivos, Enrutado de Vehículos, Logística, Transporte.**

I. Introducción

La resolución de problemas de optimización del mundo real requiere del diseño e implementación de algoritmos eficientes. Cuando el problema a resolver es de alta complejidad, como es el caso de los problemas NP-completos [1], es necesario el uso de ténicas aproximadas, como las meta-heurísticas. Dentro de dichas técnicas, los Algoritmos Evolutivos (Evolutionary Algorithms, EA) permiten explorar el espacio de búsqueda mediante el uso de una población de soluciones que evoluciona haciendo uso de los operadores de cruce y mutación, así como de estrategias de selección. En muchos casos, la ejecución secuencial en un solo procesador conlleva tiempos de ejecución muy elevados, incluso no aceptables para algunas aplicaciones. Una forma de resolver esta situación es el uso de estrategias paralelas. Cuando el problema a resolver es multi-objetivo, la complejidad de la búsqueda se incrementa, ya que es necesario obtener varias soluciones que constituyan una aproximación al frente óptimo de Pareto.

El problema del enrutado de vehículos (Vehicle Routing Problem, VRP) es un problema de optimización NP-duro [2], que tiene aplicaciones en diferentes ámbitos relacionados con la logística y el transporte. Diferentes autores han aplicado técnicas de optimización deterministas y estocásticas para su resolución [3], aunque en la actualidad una de las áreas de investigación emergentes es su optimización desde el punto de vista multi-objetivo [4].

En la presente investigación se presenta la paralelización de un algoritmo evolutivo multi-objetivo (SPEA2) para resolver un problema de transporte haciendo uso de computadores con procesadores multi-núcleo. La Sección II ofrece los conceptos generales de optimización multi-objetivo basada en Frentes de Pareto, así como la formulación multi-objetivo del problema de enrutado de vehículos con ventanas temporales, que, además de considerar el coste operativo derivado de la distancia total recorrida por todos los vehículos en suministrar a un conjunto de clientes, tiene en cuenta el balanceo de dichas rutas, en términos de distancias recorridas y carga transportada. En la Sección III describen las caraterísticas de SPEA2 y su adaptación para resolver este problema, así como su paralelización basada en el modelo de islas. Los resultados obtenidos en el estudio empírico son analizados en la Sección IV, mientras que las conclusiones y el trabajo futuro se presentan en la Sección V.

II. Optimización Multi-objetivo y Problema de Enrutado de Vehículos

A continuación ofrecemos una breve descripción del concepto de Pareto-dominancia, y la formulación multi-objetivo del problema de enrutado de vehículos con ventanas temporales.

A. Optimización multi-objetivo: conceptos básicos

La técnicas de optimización multi-objetivo basan su funcionamiento en el concepto de optimización de Pareto [5], de forma que en lugar de dar un valor escalar de calidad para cada solución obtenida, se define un orden relativo entre las mismas de acuerdo al concepto de Pareto-dominancia. Así, en un problema de minimización, se dice que una solución s_1 domina a otra s_2 ($s_1 \prec s_2$) cuando la primera mejora a la segunda en al menos un objetivo y no empeora otros. Dos soluciones se consideran *indiferentes* ($s_1 \sim s$) cuando ni s_1 domina a s_2, ni s_2 domina a s_1. El conjunto de soluciones no dominadas se denomina Frente de Pareto. Por tanto, el objetivo de cualquier técnica de optimización multi-objetivo es encontrar dicho frente, o un conjunto de soluciones que representen la mejor aproximación posible al mismo. La Figura 1 define gráficamente dichos conceptos para un problema de minimización con K=2 objetivos, Así, la Figura 1(a) muestra varias soluciones en el espacio objetivo, siendo los puntos rellenos las solu-

[1]Dpto. de Arquitectura y Tecnología de Computadores, Univ. de Granada, e-mail: {rbanos,julio}@atc.ugr.es
[2]Dpto. de Arquitectura de Computadores y Electrónica, Univ. de Almería, e-mail: {cgilm,afdezmolina}@ual.es
[3]Dpto. de Teoría de la Señal, Telemática y Comunicaciones, Univ. de Granada, e-mail: ftoro@ugr.es

ciones no dominadas, mientras que la Figura 1(b) representa las relaciones existentes entre un conjunto de soluciones y la solución de referencia s.

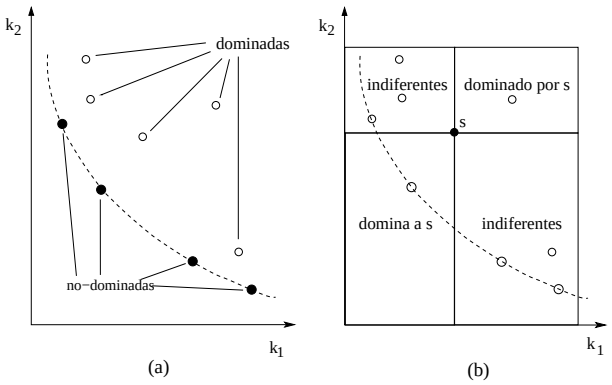

Fig. 1. Relaciones de Pareto-dominancia.

B. Enrutado de Vehículos con Ventanas Temporales: Enfoques Multi-objetivo

El problema de enrutado de vehículos con ventanas temporales (Vehicle Routing Problem with Time Windows, VRPTW) [8] consiste en determinar el conjunto de rutas óptimas que debe seguir un determinado número de vehículos con idéntica capacidad estacionados en un almacén (centro logístico) que operan dentro de una franja temporal y que son utilizados para visitar y suministrar las mercancías requeridas por los clientes. Los vehículos siguen una ruta que parte del centro logístico y vuelven a el tras realizar el servicio. La demanda de mercancías suministrada por cada vehículo no puede exceder su capacidad máxima. Los clientes, cuyas demandas pueden ser suministradas únicamente por un solo vehículo, están localizados en posiciones geográficamente diversas y tienen requisitos preestablecidos en lo que respecta a las mercancías requeridas y al tiempo de servicio necesario para la descarga. La ventana temporal de cada cliente implica que el vehículo que le suministra puede llegar antes de que la ventana temporal abra, pero no después de que esta se haya cerrado.

Existen formulaciones en las que el proceso de optimización consta de dos fases: una primera fase en la que se minimiza el número de vehículos utilizados y una segunda fase en la que, a igualdad de vehículos, se reduce la distancia total recorrida [9]. Al establecer prioridades entre objetivos, dicho enfoque no puede ser considerado multi-objetivo puro. Además, existen razones para considerar que la distancia recorrida debería ser considerada como el objetivo principal, como, por ejemplo, que cuando la distancia recorrida crece, mayor es el consumo de combustible, y por tanto mayor el coste del proceso de transporte. Por ello, y con el objetivo de extender las formulaciones tradicionales al ámbito de la vida real, algunos autores han propuesto formulaciones multi-objetivo que optimizan simultáneamente varios objetivos [4]. Entre dichas formulaciones, destacan aquellas en las que se optimiza, además de la distancia total recorrida por los vehículos, el balanceo de la carga de trabajo asociada a los vehículos. Dicha carga suele ser expresada en términos de la distancia recorrida por los vehículos [6] o la carga transportada por los mismos [7]. Ambas formulaciones multi-objetivo son tratadas en el presente trabajo.

La definición matemática de los términos y constantes del VRPTW, incluyendo las notaciones frecuentemente utilizadas para representar las rutas, almacenes, clientes, y vehículos puede ser modelada utilizando un grafo completo no dirigido G=(V,E), donde los vértices V={0,..,n} corresponden al centro logístico donde se encuentran los vehículos y mercancías, y las aristas $e \in E\{(i,j) : i,j \in V, i \leq j\}$ a las rutas existentes entre ellos.

Variables de Decisión

X_{ij}^k es igual a 1 si el k-ésimo vehículo viaja desde el nodo i al nodo j, y 0 en caso contrario.

Parámetros

a_i es el tiempo más temprano en el que el cliente i permite inicializar el servicio;
b_i es el tiempo más tardío en el que el cliente i permite inicializar el servicio;
C_{ij} es la distancia (coste) necesario para viajar desde el nodo i al nodo j;
d_i es la demanda del cliente i;
K es el máximo número de vehículos que pueden ser utilizados;
N es el número de clientes más el almacén;
Q es la capacidad de carga de los vehículos;
q_{ij} es el flujo de bienes desde el nodo i al nodo j;
s_i es el tiempo de servicio del cliente i.

El VRPTW se define como $Minimizar$:

$$TD = \sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} \sum_{k=1}^{K} X_{ij}^k C_{ij} \tag{1}$$

$$DI = max(\sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} X_{ij}^k C_{ij}) - min(\sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} X_{ij}^k C_{ij}), \quad 1 \leq k \leq K \tag{2}$$

$$LI = max(\sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} X_{ij}^k d_i) - min(\sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} X_{ij}^k d_i), \quad 1 \leq k \leq K \tag{3}$$

$$\text{t.q.:} \begin{cases} X_{ij}^k \in \{0,1\}, \quad 1 \leq k \leq K, \quad 0 \leq i,j \leq N & (4) \\ \sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} X_{ij}^k d_i \leq Q, \quad 1 \leq k \leq K & (5) \\ \sum_{j=1}^{N} \sum_{k=1}^{K} X_{0j}^k \leq K & (6) \\ \sum_{j=1}^{N} \sum_{k=1}^{K} X_{0j}^k - \sum_{j=1}^{N} \sum_{k=1}^{K} X_{j0}^k = 0 & (7) \\ \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} X_{ij}^k = 1 & (8) \\ \sum_{i=0}^{N} \sum_{j=1}^{N} q_{ij} - \sum_{i=0}^{N} \sum_{j=1}^{N} q_{ji} = d_j & (9) \\ a_{ij} \leq s_{ik} \leq b_i, \forall i \in N, \forall k \in K & (10) \end{cases}$$

Las funciones objetivo (1), (2) y (3) representan, respectivamente, la distancia total (travelling distance) de viaje recorrida por los vehículos para suministrar a los clientes, el desbalanceo máximo entre las distancias recorridas (distance imbalance) por dos vehículos, y el desbalanceo máximo en las cargas (load imbalance) transportadas por dos vehículos diferentes. La restricción (4) indica que la ruta entre dos clientes puede o no ser cubierta; la restricción (5) indica que el límite de la capacidad que los vehículos no pueden acceder; la restricción (6) especifica que, como mucho, hay K rutas (vehículos) saliendo del centro logístico; la restricción (7) garantiza que los vehículos parten desde y vuelven hacia el centro logístico; la restricción (8) especifica que un cliente es visitado una sola vez por un solo vehículo; la restricción (9) indica las restricciones de flujo relacionadas con el envío de mercancías correspondiente al cliente j; la restricción (10) asegura que los clientes son visitados dentro de su ventana temporal.

III. Procedimiento

A continuación describimos el algoritmo evolutivo multi-objetivo utilizado y su paralelización utilizando el paradigma basado en islas.

A. Algoritmo Evolutivo Multi-objetivo: SPEA2

El denominado *Strength Pareto Evolutionary Algorithm 2 (SPEA2)* [17] es un algoritmo evolutivo multi-objetivo habitualmente utilizado en el contexto de decisión con múltiples criterios. SPEA2 es una extensión de SPEA [16] que incorpora una estrategia de asignación de fitness de grano fino que tiene en cuenta, para cada individuo, el número de individuos que domina, y el número de individuos por los que es dominado. Este algoritmo evolutivo multi-objetivo aplica una técnica de estimación de densidad basada en cercanía en el vecindario que guía la búsqueda de forma más eficiente evitando concentrar las soluciones en áreas concretas del frente de soluciones. SPEA2 utiliza un archivo externo de soluciones no dominadas, *ND*, para almacenar las mejores soluciones encontradas durante la búsqueda, siempre preservando las soluciones de los extremos.

SPEA2 puede ser adaptado a cualquier problema de optimización combinatorio, incluido el aquí tratado, aunque es conveniente aclarar como lo hemos implementado. En referencia a *la representación de las soluciones* hemos hecho uso de una codificación entera, de forma que cada uno de los p individuos que forman la población P, $P=\{I_1, I_2, \ldots, I_p\}$, representa una configuración factible de las rutas recorridas por los K vehículos para suministrar a los N clientes. Así, cada individuo, I_i, se representa mediante cromosomas, C_{ik}, que consiste en un conjunto variable de genes, $C_{ik} = \{0, G^1_{ik}, G^2_{ik}, ..., G^l_{ik}, 0\}$ que representan la ruta del k-ésimo vehículo en el i-ésimo individuo, con $1 \leq G^j_{ik} \leq N$. Así, el cromosoma $C_{2,5} = \{0, 11, 4, 33, 18, 0\}$ indica que el quinto vehículo del segundo individuo parte del centro logístico, y visita a los clientes 11, 4, 33, y 18, antes de volver a dicho centro logístico. En cuanto a la *inicialización de las soluciones* estas se crean mediante una heurística rápida basada en las ventanas temporales (Time Window based Insertion Heuristic, TWIH), que consiste en visitar a los clientes según su orden de apertura de ventana temporal, que son insertados en los vehículos siempre y cuando se cumplan las restricciones del problema. En cuanto a los *operadores evolutivos*, se ha implementado un operador de cruce que une rutas de dos individuos padres en un individuo hijo, siempre verificando que no haya clientes visitados por dos vehículos o existan clientes sin visi-tar. Con la idea de diversificar la búsqueda, se han implementado diez operadores mutación comúnmente utilizados en este contexto [8], [15], entre las que se encuentran la relocalización de clientes en el mismo vehículo o reasignación en vehículos diferentes, intercambio de clientes entre vehículos, así como la creación, división o eliminación de rutas.

B. Paralelización Multi-objetivo basada en el Modelo de Islas

Entre las ventajas del procesamiento paralelo encontramos la reducción del tiempo de ejecución de las versiones secuenciales o la mejora de las mismas sin necesidad de incrementar el esfuerzo computacional requerido. Hoy día, las ventajas de dichas estrategias no se limitan al uso de equipos de altas prestaciones, como costosos clusters de computadores o máquinas de memoria compartida, sino que pueden ser explotadas en cualquier computador personal que conste de un procesador multi-núcleo haciendo uso de componentes software estándar. Entre dichas herramientas software, encontramos las liberías de paso de mensajes, entre las que destaca MPI (Message Passing Interface). Al usar este tipo de librerías, el programador tiene que desarrollar una cantidad de código importante para manejar algunas de las tareas de paralelización, como son descomponer y distribuir los puntos de búsqueda, determinar el número de comunicaciones y la estrategia de sincronización. Algorítmicamente, el paralelismo puede ser explotado utilizando diferentes paradigmas: maestro-esclavo, islas, y difusión [11].

Existe un interés creciente en el diseño de métodos más rápidos para resolver problemas de enrutado de vehículos mediante la incorporación de técnicas de paralelización de diferentes heurísticas [10], con el objetivo de generar soluciones de calidad en un intervalo de tiempo corto, además de poder abordar problemas con un mayor número de clientes a visitar. Uno de los primeros algoritmos paralelos que se aplicaron para resolver problemas de enrutado de vehículos fue la propuesta de Taillard [12], quién presentó un método paralelo que descompone dicho problema en regiones polares incluyendo un elevado número de vehículos en sub-problemas que son resueltos independientemente. Sin duda, las heurísticas, en especial la búsqueda tabu, han

sido habitualmente utilizadas para resolver este problema, aunque la mayoría de las aproximaciones utilizan el paradigma maestro-esclavo. Algunos autores han resuelto el VRPTW utilizando el paradigma de maestro-esclavo en el que el componente maestro coordina los operadores genéticos y gestionan la selección de los padres, mientras que los esclavos ejecutan lo operadores de mutación y reproducción [13]. Otros autores han propuesto algoritmos paralelos para resolver el VRPTW que hacen uso de diferentes estrategias para inicializar las soluciones, búsqueda tabu y algoritmos genéticos como operadores de búsqueda, y un procedimientos de post-optimización basados en búsqueda local [14].

Si la resolución de formulaciones mono-objetivo del problema de enrutado de vehículos haciendo uso de procesamiento paralelo no está tan extendida como debiera, en el caso multi-objetivo su uso es prácticamente inexistente. Para nuestro conocimiento, la paralelización de algoritmos heurísticos utilizando el modelo de islas en procesadores multi-núcleo para resolver el VRPTW no ha sido analizada hasta el momento, aspecto que tratamos en la presente investigación. El objetivo de nuestra paralelización (parallel SPEA2, pSPEA2) es obtener soluciones de mayor calidad que los algoritmos secuenciales sin incrementar el tiempo de ejecución. En particular, utilizamos el paradigma basado en islas [11], en el que la población secuencial se divide en *NP* islas de forma que cada proceso se encarga de gestionar una de esas islas.

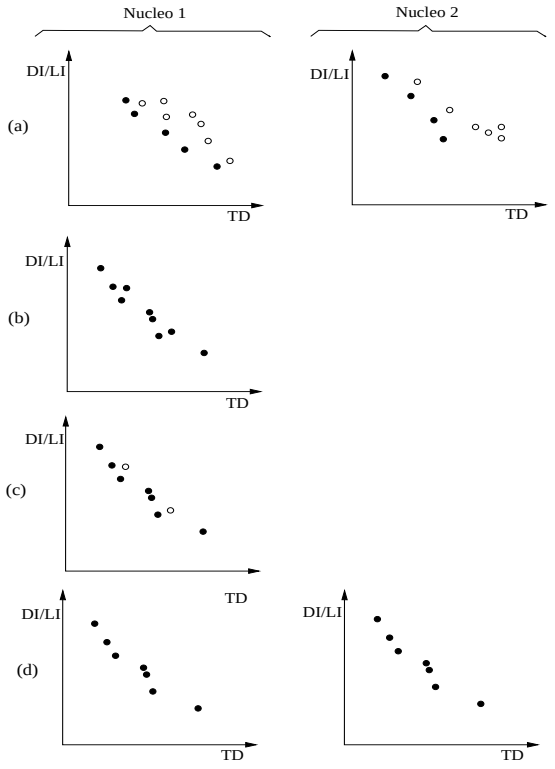

Fig. 2. Funcionamiento del modelo de islas con 4 procesos.

La Figura 2 describe el funcionamiento de las comunicaciones en una ejecución con dos procesos/núcleos. Cada proceso comienza con un conjunto de soluciones iniciales que son mejoradas aplicando SPEA2 (pSPEA2). En un instante dado, las soluciones no dominadas obtenidas en cada proceso (Figura 2(a)) son enviadas a un proceso central. Dicho proceso central agrupa todas las soluciones no dominadas (Figura 2(b)), y construye un frente global de soluciones no dominadas (Figura 2(c)) que es, en ese momento, enviado por dicho proceso central al resto de procesos (Figura 2(d)), tras lo cual cada uno de los procesos continúa el proceso de búsqueda utilizando dichas soluciones.

IV. Análisis Experimental

A. Benchmarks

La evaluación de pSPEA2 en la resolución del VRPTW se lleva a cabo utilizando el conjunto de benchmarks propuesto por Solomon [18], que se compone de 56 instancias divididas en seis categorías, todas ellas con 25 vehículos y 100 clientes. Cada benchmark incluye las capacidades máximas de los vehículos, así como las características de los clientes (coordenadas geográficas, demandas, ventanas temporales y tiempo de servicio). La distancia entre dos clientes es calculada a partir de sus coordenadas cartesianas utilizando distancias euclídeas. Las seis categorías son C1, C2, R1, R2, RC1, y RC2, donde los benchmarks de categoría R tienen los clientes distribuidos aleatoriamente, los de la categoría C están concentrados, siendo la categoría RC una mezcla de ambas. Por otro lado, los problemas de la categoría 1 (C1, R1, y RC1) tienen ventanas temporales más estrechas que los de la categoría 2 (C2, R2, y RC2).

B. Medidas de Calidad de los Frentes

La evaluación de algoritmos de optimización requieren de la toma en consideración de, al menos, dos criterios: coste computacional y la calidad de los resultados. Mientras que el coste computacional se suele medir teniendo en cuenta el tiempo de ejecución o el número de evaluaciones de la función objetivo, la calidad es más difícil de determinar, especialmente en el contexto multi-objetivo. Diferentes autores han propuesto métricas para computar los rendimientos de algoritmos evolutivos multi-objetivo [11], [19], la mayoría de los cuales son métricas de convergencia que miden la proximidad de las soluciones no dominadas al frente Pareto-óptimo, aunque en el caso de abordar NP-duros, como es el caso tratado en nuestra investigación, no es posible conocer el Frente de Pareto. Para abordar este inconveniente, se ha planteado la siguiente alternativa: se han lanzado varias ejecuciones del algoritmo para cada benchmark utilizado durante un tiempo de ejecución elevado (tres horas), tras lo cual se han compuesto los frentes obtenidos, construyendo así un frente considerado como quasi-óptimo, y por tanto utilizado como referencia para medir la proximidad de los frentes obtenidos en ejecuciones simples.

La calidad de los frentes no dominados se evalúa utilizando el hiper-volumen, que es una métrica de convergencia que se basa en las ideas propuestas por

Zitzler y Thiele [16] y que ha sido recientemente utilizada para resolver el problema VRPTW multi-objetivo [20]. Dado que los objetivos a optimizar tienen diferentes escalas, los valores de esta métrica han sido normalizados en el intervalo [0,1].

C. Configuración Paramétrica

La versión secuencial (SPEA2) utiliza una población de $|P|$=200 individuos y un archivo externo de la misma dimensión, $|ND|$=200. En la versión paralela (pSPEA2) dichas soluciones se distribuyen equitativamente entre las islas, cada una de las cuales es gestionada por un proceso ejecutado en un núcleo del procesador. Los individuos son inicializados utilizando la heurística TWIH antes descrita, y la probabilidad de aplicar un operador de cruce y de mutación en cada iteración para un determinado individuo es Pr_{cr}=Pr_{mut}=25%. Si se aplica el operador de mutación, se elige uno de los diez operadores con una probabilidad fija del 10%.

Dado que el objetivo de la paralelización (usando NP={2,4} procesos/núcleos) es mejorar la calidad de las soluciones obtenidas por la versión secuencial (NP=1) sin incrementar el esfuerzo computacional, hemos establecido como condición de parada que el tiempo de ejecución de las ejecuciones sea de 30 segundos. Investigaciones recientes en el ámbito del enrutado de vehículos han utilizado el tiempo de ejecución como medida para estimar el coste computacional [21]. En cuanto a las comunicaciones, se ha establecido el criterio de realizar una comunicación cada 3 segundos, es decir, el 10% del tiempo de ejecución total, de forma que en esos instantes concretos todos los procesos envían su frente de soluciones no dominadas al proceso central, el cual determina el frente no dominado global considerando sus propias soluciones no dominadas. Tras ello, el proceso central envía el frente de soluciones no dominadas global al resto, que continúan el proceso de búsqueda insertando dichas soluciones no dominadas en su población principal.

Se han lanzado 25 ejecuciones independientes de pSPEA2 usando NP={1,2,4} procesos/núcleos para resolver dos problemas de cada uno de los seis tipos de problemas de Solomon: C103, C108, C203, C208, R103, R108, R203, R208, RC103, RC108, RC203, y RC208, que son representativas de todos los tipos. El código secuencial ha sido implementado utilizando C++, y ha sido paralelizado utilizando MPI (MPICH2 versión 1.2.1p1) en una estación de trabajo con un procesador Intel Quad Core (4 núcleos), 2.27 GHz, y 4 GB de RAM.

D. Resultados y Discusión

La Tabla I muestra los resultados obtenidos por las versiones secuencial (NP=1 procesos/núcleos) y paralela (NP=2, y NP=4), de SPEA2/pSPEA2 en términos de hiper-volumen según la medida estadística conocida como mediana, clasificando dichos resultados según la formulación multi-objetivo, el número de procesos/núcleos utilizados y el tipo de problema. Dicha tabla muestra como, en las dos formulaciones tratadas (minimizar la distancia total recorrida y minimizar el desbalanceo de las distancias, TD & DI; minimizar la distancia total recorrida y minimizar el desbalanceo de las cargas, TD & LI), la calidad del frente no dominado con hiper-volumen mediano obtenido por el paradigma basado en islas mejora a la versión secuencial, y lo hace progresivamente conforme el número de núcleos aumenta. Esto ocurre independientemente de que el benchmark tenga a los clientes localizados en posiciones aleatorias (tipo R), concentrados (tipo C), o una mezcla de ambas (RC), o que los benchmarks tengan las ventanas temporales estrechas (tipo 1) o anchas (tipo 2). Como se indicó anteriormente, los valores numéricos representan el hiper-volumen relativo (en tanto por puntos porcentuales) en comparación los frentes de referencia obtenidos tras lanzar el algoritmo durante tres horas. Los valores medios (average, AVG) muestran claramente dicha mejora.

TABLA I
MEDIANA DEL HIPER-VOLUMEN OBTENIDO POR PSPEA2
(TIEMPO DE EJECUCIÓN FIJO DE 30 SEGS.)

	NP	"R"	"C"	"RC"	"1"	"2"	AVG
TD	1	69.02	73.02	68.14	68.02	72.10	70.06
&	2	84.06	84.47	83.37	83.12	88.18	84.64
DI	4	89.30	89.06	88.12	88.50	89.15	88.83
TD	1	65.83	72.80	66.66	66.34	70.52	68.43
&	2	78.83	83.11	79.32	77.99	82.85	80.42
LI	4	83.07	85.87	82.58	81.65	86.03	83.84

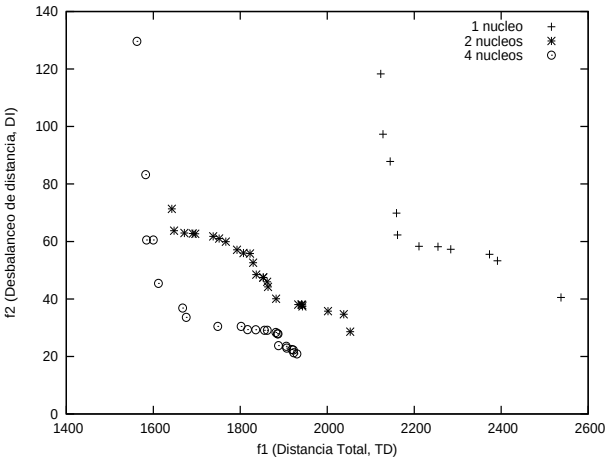

Fig. 3. Frentes medianos (formulación TD & DI), utilizando 1, 2 y 4 procesos (RC108).

La Figura 3(a) muestra los frentes medianos, en términos de hiper-volumen, obtenidos por pSPEA2 cuando se minimiza la distancia viajada y el desbalanceo de la distancia (TD & DI) en el benchmark R103. Claramente se observa como la implementación paralela permite obtener frentes de soluciones no dominadas de mejor calidad que la versión secuencial sin incrementar el tiempo de ejecución.

Por otro lado, la Figura 4 muestra gráficamente el speed-up obtenido por pSPEA2 cuando se resuelve

la formulación que minimiza la distancia viajada y el desbalanceo en la distancia bajo la condición de parada de que se alcance un hiper-volumen equivalente al 92% del hiper-volumen del frente de referencia. Como se observa, la mejora en términos de speed-up es muy importante, especialmente cuando se utilizan dos procesos (núcleos). Por tanto, hemos comprobado como las versiones paralelas basadas en el modelo de islas no sólo mejoran a las versiones secuenciales en términos de calidad de los frentes a igualdad de tiempo de ejecución, sino que también permiten reducir considerablemente el esfuerzo requerido para obtener soluciones de calidad equivalente en términos de hiper-volumen a las de las versiones secuenciales.

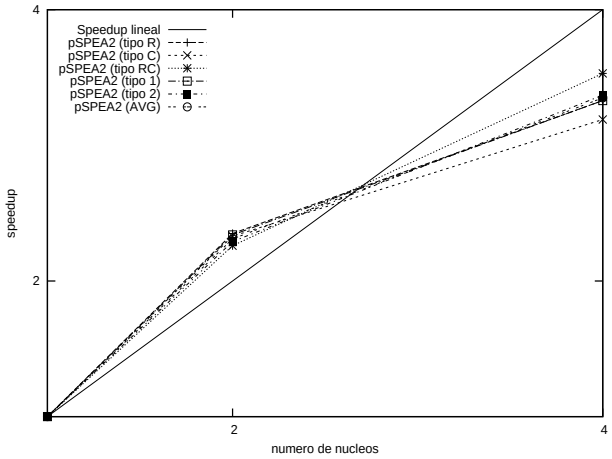

Fig. 4. Speed-up de los frentes medios resolviendo la formulación TD & DI (hiper-volumen=92%)

V. Conclusiones

En el presente trabajo se ha analizado como resolver un problema complejo, como es el del enrutado de vehículos con ventanas temporales, desde una perspectiva multi-objetivo haciendo uso de procesamiento paralelo. En concreto, se han abordado dos variantes bi-objetivo que, además de considerar la minimización de la distancia total recorrida por un conjunto de vehículos, minimiza el desbalanceo de las rutas desde dos puntos de vista: desbalanceo en las distancias recorridas y desbalanceo en la carga transportada. La paralelización del algoritmo evolutivo multi-objetivo utilizado (SPEA2), se ha llevado a cabo haciendo uso del paradigma basado en islas. Los resultados experimentales permiten obtener dos conclusiones: dicha paralelización, para ambas formulaciones de este problema, permiten reportar soluciones de mayor calidad que la versión secuencial sin incrementar el coste computacional, a la vez que ofrece una mejora de la calidad de los frentes no dominados a igualdad de coste computacional. Como trabajo futuro se plantea la paralelización haciendo uso de esquemas asíncronos, el uso de otros esquemas de paralelización haciendo uso de librerías de memoria compartida, en especial OpenMP, así como su aplicación a otras variantes del problema de enrutado de vehículos.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Ciencia e Innovación a través de los proyectos TIN2008-01117 y SAF2010-20558. R. Baños agradece la finaciación del contrato postdoctoral Juan de la Cierva.

Referencias

[1] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman & Company, San Francisco, 1979.

[2] J.K. Lenstra, A.H.G.R. Rinnooy Kan, *Complexity of vehicle routing and scheduling problems*, Networks 11, 221-227 (1981).

[3] B. Eksioglu, A.V. Vural, A. Reisman, *The vehicle routing problem: a taxonomic review*, Computers & Industrial Engineering 57(4), 1472-1483 (2009).

[4] N. Jozefowiez, F. Semet, E-G. Talbi, *Multi-objective vehicle routing problems*, European Journal of Operational Research 189(2), 293-309 (2008).

[5] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publisher, 1989.

[6] N. Jozefowiez, F. Semet, E-G. Talbi, *An evolutionary algorithm for the vehicle routing problem with route balancing*, European Journal of Operational Research 195(3), 761-769 (2009).

[7] M.N. Kritikos, G. Ioannou, *The balanced cargo vehicle routing problem with time windows*, International Journal of Production Economics 123(1), 42-51 (2010).

[8] N.A. El-Sherbeny, *Vehicle routing with time windows: an overview of exact, heuristic and metaheuristic methods*, Journal of King Saud University (Science) 22(3), 123-131 (2010).

[9] Y. Nagata, O. Bräysy, W. Dullaert, *A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows*, Computers & Operations Research 37, 724-737 (2010).

[10] O. Bräysy, M. Gendreau, *Vehicle routing problem with time windows, part II: metaheuristics*, Transportation Science 39, 119-139 (2005).

[11] C.A., Coello, G.B. Lamont, D.A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*, Genetic and Evolutionary Computation, Springer (2007).

[12] É. Taillard, *Parallel iterative search methods for vehicle routing problems*, Networks 23(8), 661-673 (1993).

[13] J. Berger, M. Barkaoui, *A parallel hybrid genetic algorithm for the vehicle routing problem with time windows*, Computers & Operations Research 31(12), 2037-2053 (2004).

[14] A. Le Bouthillier, T.G. Crainic, *A cooperative parallel meta-heuristic for the vehicle routing problem with time windows*, Computers & Operations Research 32(7), 1685-1708 (2005).

[15] G.B. Alvarenga, G.R. Mateus, G. de Tomic, *A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows*, Computers & Operations Research 34(6), 1561-1584 (2007).

[16] E. Zitzler, L. Thiele, *Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach*, IEEE Transactions on Evolutionary Computation 3(4), 257-271 (1999).

[17] E. Zitzler, M. Laumanns, L. Thiele, *SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization*, In Proceedings of Evolutionary Methods for Design, Optimisation, and Control, Barcelona, Spain, 95-100 (2001).

[18] M.M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Operations Research 35(2),254-65 (1987).

[19] K. Deb, *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons, 2002.

[20] A. Garcia-Najera, J.A. Bullinaria, *An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows*, Computers & Operations Research 38(1), 287-300 (2011).

[21] H. Derbel, B. Jarboui, S. Hanafi, H. Chabchoub, *An iterated local search for solving a location-routing problem*, Electronic Notes in Discrete Mathematics 36, 875-882 (2010).