

# Distribución balanceada de ayuda humanitaria resuelto con un algoritmo metaheurístico

Ivo Erasmo Buzón Cantera<sup>1</sup>, Jaime Mora Vargas<sup>1</sup>,  
Ángel Ruiz Bartolomé<sup>2</sup> y Patrick Soriano<sup>3</sup>

<sup>1</sup> Tecnológico de Monterrey Campus Estado de México, México  
milvus\_ca@yahoo.com, jmora@itesm.mx

<sup>2</sup> Université Laval, Quebec, Quebec, Canadá  
angel.ruiz@fsa.ulaval.ca

<sup>3</sup> HEC Montréal, Montreal, Quebec, Canadá  
patrick.soriano@hec.ca

**Resumen** La distribución de ayuda humanitaria tras un desastre busca satisfacer las necesidades básicas de los afectados. La mayor dificultad es la planificación de la distribución desde el centro de distribución local a los puntos de distribución, donde la ayuda humanitaria es repartida directamente a los afectados. La falta de ayuda humanitaria, la carencia de capacidad de transporte y una red de comunicación dañada son situaciones que tienen que hacer frente los administradores de la distribución de ayuda humanitaria. Este trabajo desarrolla un modelo de la distribución balanceada de ayuda humanitaria basado en el modelo del problema de ruteo. Para resolver el problema de distribución balanceada de ayuda humanitaria este trabajo desarrolla un algoritmo híbrido entre un algoritmo metaheurístico y un método exacto. Los resultados experimentales muestran que el algoritmo propuesto se desempeña bien para el problema de distribución balanceada de ayuda humanitaria.

**Palabras clave:** metaheurísticas, distribución balanceada, ayuda humanitaria.

## 1. Introducción

Los desastres de origen natural han afectado negativamente a la humanidad. Un desastre produce destrucción, pérdidas humanas y materiales, dejando a una población desvalida que no puede satisfacer sus necesidades básicas. Las infraestructuras han sido parcial o totalmente destruidas y las cadenas de suministros habituales están inoperantes [11]. Dentro de la logística humanitaria, la distribución de ayuda humanitaria (DAH) es la actividad que realiza el transporte de la ayuda humanitaria desde el centro de distribución local (CDL) a los puntos de distribución (PD) mediante una flota heterogénea de vehículos. Desde los puntos de distribución la ayuda humanitaria es repartida directamente a la población afectada. Balcik et al. [2] establecen que un mal desempeño de la DAH produce sufrimiento a los afectados por una privación de sus necesidades básicas. Kovacs y Spens [10] indican que el atraso en el suministro de ayuda humanitaria puede resultar en pérdida de vidas. En

consecuencia, un buen plan de la DAH es absolutamente necesario para repartir la ayuda humanitaria disponible. Huang et al. [8] y Perez et al. [13] resaltan que el plan de DAH es realizado frecuentemente de manera ad-hoc y puede llevar a un uso ineficiente de los recursos, a una respuesta lenta y a una injusta distribución de la ayuda. De manera similar Ergun et al. [7] detectan que las personas responsables de la administración de la cadena de suministro y de la logística humanitaria en la mayoría de las organizaciones generalmente no son especialistas, y no tienen las herramientas o sistemas adecuados para resolver los problemas relacionados con la DAH. Una respuesta exitosa a un desastre no es improvisada, Van Wassenhove [18] recomienda estar preparados para establecer la respuesta más efectiva. Pero la realización del plan de DAH de manera rápida y óptima es una tarea difícil. Las técnicas de investigación de operaciones aplicadas a la DAH pueden mejorar su eficiencia y eficacia enormemente [2, 7]. La distribución de ayuda humanitaria es diferente a otros problemas de distribución. El objetivo de la distribución humanitaria es proveer asistencia humanitaria, no la minimización de los costos. Un problema que tienen que hacer frente los administradores es la limitación de recursos, representada por la ayuda humanitaria disponible y la capacidad de transporte [2, 7, 10].

En este trabajo, se desarrolla un modelo del problema de DAH que incorpora el balanceo de la distribución. El modelo incluye las decisiones de ruteo y las decisiones de asignación de ayuda humanitaria a los PD. El modelo se desarrolla a partir de la formulación de dos índices del problema de ruteo de vehículos [17]. El modelo se formula con múltiples periodos para poder considerar adecuadamente la realización de una distribución balanceada, la satisfacción de la demanda con atrasos y el inventario en los puntos de distribución. Para resolver el problema de DAH se adapta el algoritmo metaheurístico de recocido simulado definiendo una función perturbación con múltiples vecindades. También se desarrolla un algoritmo mateheurístico que tiene como algoritmo principal un algoritmo de recocido simulado que incluye la resolución de un subproblema con programación entera para explorar una gran vecindad.

## 2. Estado del arte

El trabajo de Altay y Green III [1] es la única revisión de la literatura en administración de la logística humanitaria. Concluyen que existe poca investigación sobre la planificación del transporte para la distribución de la ayuda humanitaria. Balcik et al. [2] y Sheu [16] identifican oportunidades para el uso de sistemas inteligentes de transporte en la DAH. En los trabajos presentados se presentan diferentes modelos de la DAH, cada trabajo presenta el problema de DAH de manera diferente en función de las decisiones que quiere responder y de los supuestos establecidos. La resolución del problema de DAH mediante métodos exactos solo puede ser realizada para instancias de pequeño tamaño en un tiempo razonable [2, 19], sin embargo De Angelis et al. [5] resuelven el modelo propuesto con programación entera mixta. Por la complejidad del problema, los métodos presentados descomponen el problema en dos etapas [2, 3, 11, 19], aunque existen varios trabajos que resuelven el problema en una sola etapa [5, 12]. Barbarosoglu et al. [3] presentan un método heurístico interactivo para la coordinación de los dos sub

problemas con un sistema de múltiples criterios jerárquico. Yi y Ozdamar [19] y Perez et al. [13] utilizan un método con dos etapas: La primera etapa define un problema de flujo en redes entero mixto, en el cual los vehículos se representan como productos siendo su flujo una variable entera. La segunda etapa define las rutas y la carga para cada vehículo. Balcik et al. [2] presentan un sistema con dos fases. La primera fase genera todas las rutas no dominadas. La segunda fase mediante programación entera mixta asigna las rutas a los vehículos, determina la carga de los vehículos y las cantidades a distribuir en cada PD. Lin et al. [11] presentan dos heurísticos. El primer heurístico reduce el número de rutas al considerar un subconjunto, las rutas se filtran mediante un algoritmo genético. El segundo heurístico descompone el problema inicial en subproblemas que solo contienen un número parcial de vehículos y de PD. Nolz et al. [12] presentan una formulación matemática entera no lineal y un método heurístico basado en el algoritmo genético de ordenamiento no-dominado II (NSGA-II). En Yi y Ozdamar [19] el objetivo es minimizar la demanda de ayuda no satisfecha y Perez et al. [13] utilizan el concepto de costo social, estableciendo el objetivo como minimizar la suma de los costos operacionales y los costos sociales. Huang et al. [8] presenta un modelo de la DAH para un solo periodo sin limitaciones, el objetivo minimiza el balance en el tiempo de reparto a los PD.

### 3. Descripción del problema

El problema de distribución balanceada de ayuda humanitaria (PDBAH) puede ser formalmente definido como sigue. Sea  $G=(V,A)$  un grafo dirigido donde  $V=\{0, \dots, n\}$  es el conjunto de los nodos y  $A=\{(i,j): i,j \in V, i \neq j\}$  es el conjunto de los arcos. El nodo 0 corresponde al CDL y el resto de nodos  $\{1, \dots, n\}$  representan los PD. Un tiempo de recorrido  $c_{ij}$  no negativo está asociado a cada arco  $(i,j) \in A$  y representa el tiempo necesario para recorrer la distancia del nodo  $i$  al nodo  $j$ . En el presente trabajo se considera que los tiempos de recorrido son simétricos, es decir,  $c_{ij} = c_{ji}$  para todo arco  $(i,j) \in A$ . Una flota heterogénea está disponible, representada por el conjunto  $k$ , y cada vehículo está limitado en la capacidad de carga ( $CAP_k$ ) y en el tiempo de trabajo máximo ( $W$ ). Cada PD  $i$ , ( $i=1, \dots, n$ ) está asociado con una conocida demanda  $d_i$  no negativa para cada periodo  $t$  del horizonte de planeación, el CDL está asociado a una demanda igual a cero. La ayuda disponible en el CDL es conocida y no negativa para cada periodo  $t$  del horizonte de planeación e inferior en cada periodo a la demanda total. Una representación esquemática del PDBAH se presenta en la figura 1. La solución considera también la capacidad de almacenaje de ayuda humanitaria en los PD para satisfacer la demanda del periodo siguiente, que los vehículos puedan realizar varias rutas en cada periodo, varios vehículos pueden visitar el mismo PD en un mismo periodo, se permite satisfacer parcialmente la demanda de los PD y los vehículos pueden realizar varias rutas sin sobrepasar el tiempo de trabajo máximo. El objetivo principal del PDBAH es minimizar las penalizaciones por satisfacer la demanda tardíamente. Los factores de penalización se definen en relación con el atraso: a) Si la demanda es satisfecha en el mismo periodo no hay penalización. b) Si la demanda es satisfecha con un atraso de 1 o 2 periodos se define el factor de penalización  $p_1$  y  $p_2$ . c) Si la demanda no es satisfecha está asociada a un factor de

penalización  $p_3$ . También incluye un objetivo secundario que es el balance de la distribución entre los PD. El balance busca un grado de equidad en el servicio recibido por los PD. La equidad total que todos reciban exactamente el mismo servicio, no puede lograrse por las limitaciones en los recursos del problema. El servicio recibido por cada PD se mide al calcular la demanda no satisfecha porcentual. El balance del servicio se incorpora al objetivo al incluir la demanda no satisfecha porcentual de cada PD, a través de una función convexa lineal por segmentos que relaciona el porcentaje de demanda no satisfecha del PD con un peso. Huang et al. [8] presentan que esta forma de establecer el balance tiene dos ventajas frente a minimizar la máxima diferencia del servicio recibido por los PD. La primera ventaja es que el modelo se mantiene lineal. La segunda ventaja es que se obtienen mejores resultados en los otros objetivos. La solución del PDBAH establece: Los PD que reciben ayuda humanitaria en cada periodo, La cantidad de ayuda humanitaria que recibe cada PD en cada periodo, Las rutas de los vehículos y la asignación de las rutas a los vehículos.

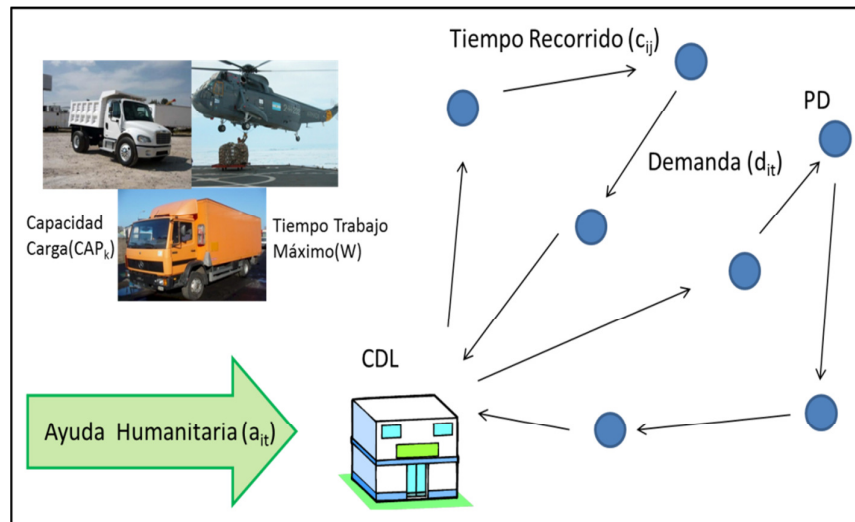


Fig. 1. Representación esquemática del PDBAH.

### 3.1. Formulación del problema

La siguiente nomenclatura es utilizada para formular el PDBAH:

- I,J: conjunto de nodos
- T: conjunto de periodos de tiempo, tamaño del horizonte de planeación
- K: conjunto de vehículos
- R: conjunto de rutas
- B: segmentos de la función convexa lineal por segmentos

$CAP_k$ : capacidad del vehículo  $k$  (unidades de producto)  
 $W$ : tiempo de trabajo máximo del vehículo  $k$   
 $p_1, p_2$  y  $p_3$ : factores de penalización por atraso  
 $d_{it}$ : demanda del PD  $i$  en el periodo  $t$   
 $a_t$ : la ayuda total disponible a ser repartida en el periodo  $t$   
 $c_{ij}$ : tiempo de recorrido del nodo  $i$  al nodo  $j$   
 $M$ : es un valor muy grande  
 $xval_b, yval_b$ : valores en las intersecciones de la función convexa lineal por segmentos,  $f(xval_b) = yval_b$   
 $x_{ijrkt}$ : 1 si el vehículo  $k$  viaja del nodo  $i$  al nodo  $j$  en el periodo  $t$  en la ruta  $r$ , 0 en otro caso.  
 $y_{jrkt}$ : 1 si el vehículo  $k$  visita al PD  $j$  en el periodo  $t$  en la ruta  $r$ , 0 en otro caso.  
 $s_{ikrt}$ : cantidad de ayuda a dar por el vehículo  $k$  en la ruta  $r$  en el periodo  $t$  en el PD  $i$  para la demanda del  $t$ .  
 $inv_{ikrt}$ : cantidad de ayuda dará el veh.  $k$  en la ruta  $r$  en el periodo  $t$  al PD  $i$  para la demanda del periodo  $t+1$ .  
 $da1_{ikrt}$ : cantidad de ayuda dará el veh.  $k$  en la ruta  $r$  en el periodo  $t$  al PD  $i$  para la demanda del periodo  $t-1$ .  
 $da2_{ikrt}$ : cantidad de ayuda dará el veh.  $k$  en la ruta  $r$  en el periodo  $t$  al PD  $i$  para la demanda del periodo  $t-2$ .  
 $dn_{it}$ : cantidad de demanda no satisfecha del PD  $i$  del periodo  $t$ .  
 $pdn_i$ : porcentaje de la demanda no satisfecha del PD  $i$   
 $pesopd_i$ : peso de la demanda no satisfecha del PD  $i$   
 $pos_b$ : variable para establecer la relación entre  $pdn_i$  y  $pesopd_i$  con la función convexa lineal por segmentos.  
 El modelo del PDBAH:

$$\begin{aligned}
 \text{Min} \left[ \sum_{i \in V \setminus \{0\}} \sum_{t \in T} \left( \sum_{k \in K} \sum_{r \in R} p_1 da1_{ikrt} + p_2 da2_{ikrt} \right) + p_3 dn_{it} \right] \\
 + \sum_{i \in V \setminus \{0\}} pesopdn_i
 \end{aligned} \quad (1)$$

Sujeto a:

$$\sum_{j \in V} x_{ijrkt} = \sum_{j \in V} x_{jirkt} = y_{jrkt}, \forall (i, r, k, t) \quad (2)$$

$$\sum_{i \in V} \sum_{j \in V} \sum_{r \in R} c_{ij} x_{ijrkt} \leq W, \forall (k, t) \quad (3)$$

$$\sum_{i \in V} da2_{ikrt} + da1_{ikrt} + s_{ikrt} + inv_{ikrt} \leq CAP_k, \forall (r, k, t) \quad (4)$$

$$\sum_{i \in V} \sum_{r \in R} \sum_{k \in K} da2_{ikrt} + da1_{ikrt} + s_{ikrt} + inv_{ikrt} \leq a_t, \forall (t) \quad (5)$$

$$da2_{irkt} + da1_{irkt} + s_{irkt} + inv_{irkt} \leq M y_{irkt}, \forall(i, r, k, t) \quad (6)$$

$$\sum_{k \in K} \sum_{r \in R} inv_{irkt} \leq d_{it+1}, \forall(i, t = 1, \dots, T-1) \quad (7)$$

$$inv_{irkt} = 0, \forall(i, r, k, t = T) \quad (8)$$

$$d_{it} \leq \sum_{k \in K} \sum_{r \in R} s_{irkt} + da1_{irk,t+1} + da2_{irk,t+2} + dn_{it}, \forall(i, t = 1) \quad (9)$$

$$d_{it} \leq \sum_{k \in K} \sum_{r \in R} inv_{irk,t-1} + s_{irkt} + da1_{irk,t+1} + da2_{irk,t+2} + dn_{it}, \forall(i, t = 2, \dots, T-2) \quad (10)$$

$$d_{it} \leq \sum_{k \in K} \sum_{r \in R} inv_{irk,t-1} + s_{irkt} + da1_{irk,t+1} + dn_{it}, \forall(i, t = T-1) \quad (11)$$

$$d_{it} \leq \sum_{k \in K} \sum_{r \in R} inv_{irk,t-1} + s_{irkt} + dn_{it}, \forall(i, t = T) \quad (12)$$

$$u_{ikrt} - u_{jkrt} + CAP_k x_{ijrkt} \leq CAP_k - d_{jt}, \forall(i, j, k, r, t) \quad (13)$$

$$d_{it} \leq u_{irkt} \leq CAP_k, \forall(i, r, k, t) \quad (14)$$

$$pdn_i = \frac{\sum_{t \in T} dn_{it}}{\sum_{t \in T} d_{it}}, \forall i \quad (15)$$

$$pdn_i = \sum_{b \in B} pos_{ib} xval_b, \forall i \quad (16)$$

$$pesopdn_i = \sum_{b \in B} pos_{ib} yval_b, \forall i \quad (17)$$

$$\sum_{b \in B} pos_{ib} = 1, \forall i \quad (18)$$

$$x_{ijrkt} \in \{0,1\}, \forall(i, j, r, k, t) \quad (19)$$

$$y_{irkt} \in \{0,1\}, \forall(i, r, k, t) \quad (20)$$

$$s_{irkt}, da1_{irkt}, da2_{irkt}, inv_{irkt} \geq 0, \forall(i, r, k, t) \quad (21)$$

$$dn_{it} \geq 0, \forall(i, t) \quad (22)$$

$$pdn_i, pesopdn_i, pos_{ib} \in (0,1), \forall(i, b) \quad (23)$$

La función objetivo (1) busca minimizar la penalización de los atrasos como objetivo principal y el objetivo secundario es la realización de una distribución balanceada. La restricción (2) establece la conservación de flujo y la asociación vehículo - PD - periodo - ruta. La restricción (3) permite considerar el tiempo de trabajo máximo. De manera similar, la restricción (4) considera la capacidad de carga del vehículo. La restricción (5) considera la cantidad de ayuda humanitaria disponible. La restricción (6) tiene en cuenta que se entreguen cantidades si el PD es visitado. Las restricciones (7 y 8) limitan los valores máximos de inventario. Restricciones (9,

10, 11 y 12) asocian la demanda con la ayuda humanitaria entregada y la demanda no satisfecha. La restricción (13 y 14) permiten eliminar los subtours. La restricción (15) calcula el valor del porcentaje de la demanda no satisfecha. La restricción (16) relaciona el porcentaje de la demanda no satisfecha a través de la función convexa lineal por segmentos para establecer el valor de  $\text{posib}$ . La restricción (17) calcula el peso del servicio por demanda no satisfecha  $\text{pesopdi}$  a través de la función convexa lineal por segmentos y el valor de  $\text{posib}$ . La restricción (18) establece que  $\text{posib}$  tiene un valor de 1, para establecer adecuadamente el valor de  $\text{pesopdi}$  a partir de  $\text{pdni}$ . Las ecuaciones (19 - 23) limitan a las variables de manera general. Analizando la formulación del PDBAH se observa que es la unión de dos problemas: el problema de asignación de ayuda humanitaria y el problema de ruteo.

#### 4. Algoritmo recocido simulado

Existen diferentes caminos para solucionar problemas de optimización combinatorios como el PDBAH. Raidl y Puchinger [15] clasifican en dos categorías las técnicas para resolver los problemas de optimización combinatorios: los métodos de programación matemática o métodos exactos y los métodos heurísticos o metaheurísticos. Entre los métodos metaheurísticos, Kirkpatrick et al. [9] introdujeron hace poco más de 25 años el algoritmo recocido simulado que ha demostrado ser una herramienta exitosa para resolver una amplia gama de problemas de optimización combinatoria. Su nombre proviene del proceso de enfriamiento de materiales: recocido [6]. El recocido simulado es el más antiguo entre los metaheurísticos y uno de los primeros algoritmos que tienen una estrategia explícita para salir de los óptimos locales. La idea fundamental es permitir movimientos que resultan en soluciones de peor calidad que la solución actual. Este movimiento tiene el propósito de escapar del óptimo local actual para poder explorar el espacio de soluciones y eventualmente encontrar otro óptimo local de mejor calidad. La probabilidad de hacer tal movimiento disminuye durante al búsqueda.

La figura 2 presenta el pseudo código del algoritmo de recocido simulado para problemas de maximización con función de costo  $c$ . El algoritmo empieza generando una solución inicial, que pasa a ser la solución actual  $s$ . La solución inicial se genera generalmente de forma aleatoria o heurística. Luego se inicializan los parámetros: La variable temperatura  $\text{Temp}$  toma el valor de la temperatura inicial  $\text{Temp}_{\text{ini}}$ , el factor de reducción de la temperatura  $\alpha$ , las repeticiones del ciclo interno  $\text{iter}$  y la temperatura final o criterio de paro  $\text{Temp}_{\text{fin}}$ .

**Fig. 2.** pseudo código del algoritmo de recocido simulado.

El algoritmo de recocido simulado consta de dos ciclos, uno externo (o ciclo de temperatura) y uno interno (o ciclo Metropolis). El siguiente paso inicia el ciclo externo que termina cuando la temperatura  $\text{Temp}$  sea inferior al criterio de paro  $\text{Temp}_{\text{fin}}$ . El núcleo del algoritmo es el ciclo interno que se repite  $\text{iter}$  veces. En el ciclo interno, primero se establece una nueva solución  $s_{\text{new}}$  diferente de la solución actual  $s$ , mediante una función perturbación o mecanismo de vecindad. La solución  $s_{\text{new}}$  es una solución de la vecindad de  $s$ . Se calcula el incremento de energía  $\Delta E$  como la diferencia entre el costo de la nueva solución  $c(s_{\text{new}})$  y el costo de la solución actual

$c(s)$ . Si  $\Delta E > 0$  entonces la solución nueva  $s_{new}$  es mejor que la solución actual  $s$ . La nueva solución  $s_{new}$  se convierte en la solución actual  $s$ . Sin embargo, a diferencia de los algoritmos de búsqueda local, si  $\Delta E < 0$  entonces se genera un número aleatorio y se compara con  $\exp(\Delta E / Temp)$ . De ser menor, entonces la nueva solución  $s_{new}$  se convierte en la solución actual  $s$ . El número aleatorio se genera según una función de probabilidad uniforme entre 0 y 1. Es importante observar que mientras disminuye la temperatura disminuye la probabilidad de realizar cambios que impliquen un empeoramiento del costo de la solución actual  $s$ . Tras terminar el ciclo interno el valor de la temperatura  $Temp$  es actualizado según el parámetro  $\alpha$ . El valor de  $\alpha$  debe ser inferior a 1 para que la temperatura disminuya.

**Paso 1: Inicialización**

$s$  = solución inicial  
 $temp$  = Temperatura inicial  
 $\alpha$  = factor de reducción  
 $iter$  = número de iteraciones  
 $T_{fin}$  = Temperatura final (criterio de paro)

**Paso 2: Búsqueda**

mientras  $Temp > T_{fin}$   
 desde  $x=1$  a  $iter$

**Paso 3: Nueva solución**

$s_{new}$  = perturbación (  $s$  )

**Paso 4: Comparación**

$\Delta E = c ( s_{new} ) - c ( s )$

si  $\Delta E > 0$  entonces

$s = s_{new}$

sino

si aleatorio  $[0,1] < \exp (\Delta E / temp)$  entonces

$s = s_{new}$

fin si

fin si

fin desde

$temp = \alpha * temp$

fin mientras

**Paso 5: Solución final  $s$**

## 5. Estructura de la solución

Un elemento importante para la aplicación del algoritmo de recocido simulado es la estructura de la solución. La estructura de la solución debe poder representar a cualquier solución del problema y a la vez debe ser adecuada para poder aplicar eficazmente la función perturbación. Como el PDBAH es la combinación de un problema de ruteo y un problema de asignación, la estructura de la solución está formada por dos partes. La figura 3 presenta un ejemplo de la estructura de la solución definida para el PDBAH. La primera parte, conjunto ruta, define las rutas que realizan los vehículos. Cada elemento del conjunto ruta es la ruta que realiza un vehículo (K), en un periodo (T) y en una ruta (R). El primer valor indica el número



total de PD a visitar, el resto los PD a visitar y su orden indica el orden en que serán visitados. No se incluye al CDL (nodo 0) porque toda ruta inicia y termina en el CDL. La segunda parte es la matriz asignación que establece la cantidad de ayuda humanitaria que se entrega en cada PD. Cada fila de la matriz asignación esta asociada a una ruta y cada columna representa a un PD, el valor de la matriz indica la cantidad de ayuda humanitaria a entregar en el PD.

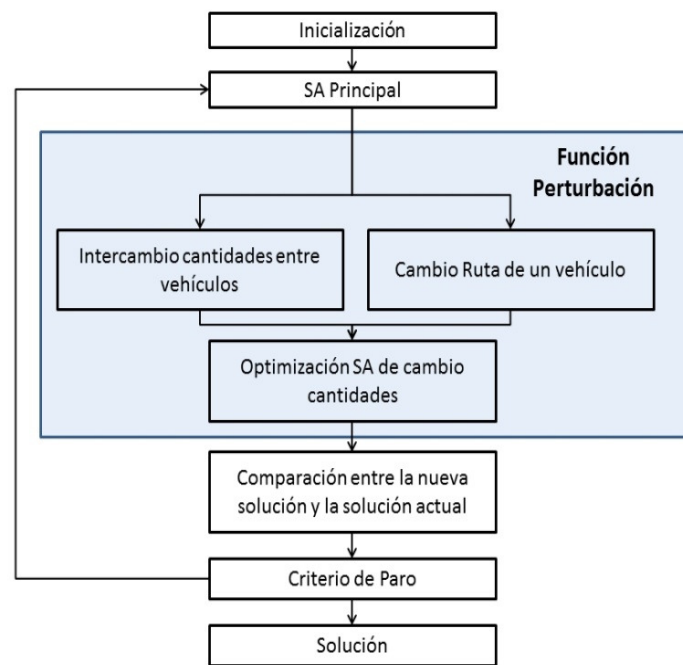
Conjunto ruta			Matriz asignación								
Periodo: T			Punto de Distribución								
Vehículo: K			1	2	3	4	5	6	7	8	
Ruta: R			1	2	3	4	5	6	7	8	
1 1 1	:	(3, 5, 7, 8)	[0	0	100	0	0	0	75	60]	
1 1 2	:	(4, 1, 2, 6, 8)	[50	125	0	0	0	75	0	50]	
1 2 1	:	(1, 4)	[0	0	0	200	0	0	0	0]	
1 2 2	:	(2, 5, 7)	[0	0	0	0	80	0	75	0]	
2 1 1	:	(3, 3, 1, 4)	[10	0	100	70	0	0	75	60]	
...			...								

Fig. 3. Estructura de la solución para el PDBAH.

La figura 4 presenta esquemáticamente el algoritmo de recocido simulado desarrollado para el PDBAH. El algoritmo inicia con la inicialización de variables y la definición de una solución inicial factible que es generada aleatoriamente. El proceso de generación de la solución inicial tiene dos etapas en la primera etapa se generan rutas factibles para cada vehículo y en la segunda etapa se asignan cantidades a los PD visitados. Posteriormente inicia el ciclo externo y el ciclo interno. Este trabajo ha desarrollado una compleja función perturbación de la solución actual y obtener una nueva solución. La función esta formada de tres tipos de cambios: intercambio de cantidades entre vehículos, cambio en la ruta de un vehículo y cambio de cantidades al interior de una ruta. La función perturbación inicia cambiando la solución actual bien por un intercambio de cantidades entre vehículos o por un cambio en la ruta de un vehículo. Posteriormente se aplica el cambio de cantidades al interior de una ruta a diferentes rutas. Finalmente se obtiene una nueva solución. El cambio a aplicar se determina aleatoriamente, los pasos son: (1) generar un número aleatorio entre cero y uno según una distribución uniforme, (2) comparar el número aleatorio con el valor del parámetro  $\epsilon$  para aplicar el cambio en la ruta de un vehículo y (3) si el número aleatorio es inferior a  $\epsilon$  se aplica el cambio en la ruta de un vehículo sino se aplica el intercambio de cantidades entre vehículos. Siguiendo las etapas del algoritmo de recocido simulado la nueva solución es comparada con la solución actual y puede que la solución nueva se convierta en la solución actual. El algoritmo termina cuando la temperatura alcanza el criterio de paro.

La solución inicial comienza con una solución vacía. Para definir el conjunto rutas se recorre todas las rutas posibles. A cada ruta se añaden PD de manera aleatoria hasta que alcanza el tamaño de 5 PD o sobrepasa el tiempo de trabajo máximo del

vehículo. Previamente a concluir que se sobrepasa el tiempo de trabajo máximo, se analizan todas las posibles rutas. Si todas las rutas sobrepasan el tiempo de trabajo máximo se elimina el último PD añadido. Una vez definido el conjunto de rutas se define la matriz asignación. El primer paso es asignar una cantidad fija pequeña a cada PD visitado. El segundo paso recorre cada ruta, si está visita al menos un PD, se determina la cantidad a rellenar que es el mínimo entre la capacidad de vehículo restante y la ayuda humanitaria disponible. La ayuda humanitaria disponible se actualiza cada vez que una parte es asignada a un PD. La cantidad a rellenar se añade al primer PD visitado por la ruta. En el tercer paso, cada ruta pasa por una fase de mejora al aplicar el cambio de cantidades al interior de la ruta.



**Fig. 4.** Esquema del algoritmo de recocido simulado para el PDAH.

En el cambio de cantidades entre vehículos una cantidad de ayuda humanitaria que iba a ser entregada por un vehículo pasa a ser entregada por otro vehículo. Primero se selecciona un periodo y después dos vehículos-ruta de manera aleatoria según una distribución uniforme, la carga del segundo vehículo-ruta debe ser inferior a la capacidad de carga del vehículo. Luego se intercambian una cantidad, definida aleatoriamente según una distribución uniforme. La cantidad a intercambiar es reducida de un PD del primer vehículo-ruta para ser asignada a un PD del segundo vehículo-ruta. Las rutas modificadas pasan por una fase de mejora al aplicar el cambio de cantidades al interior de la ruta.

El cambio de la ruta de un vehículo modifica la ruta al incluir un nuevo PD. El primer paso es seleccionar una ruta de un vehículo de un periodo de manera aleatoria según una distribución uniforme. Posteriormente se añade un PD nuevo y diferente de los PD visitados por la ruta de manera aleatoria como último nodo visitado. Si la ruta tiene seis PD se elimina al primer PD visitado por la ruta. En la DAH las rutas son cortas y en general no tienen más de tres PD, ya que la capacidad de los vehículos respecto de las necesidades de los PD es relativamente pequeña. Se verifica que la ruta con el nuevo PD sea factible, es decir que se cumple el tiempo de trabajo máximo. Para lo cual se validan todas las rutas posibles hasta que se encuentra una que sea factible, sino se puede formar una ruta factible se eliminan PD según el orden de visita original de la ruta, hasta que se forma una ruta factible o se queda vacía. Una vez modificada la ruta, el segundo paso es la reasignación de las cantidades de ayuda humanitaria. De manera similar al cambio de cantidades entre vehículos, la nueva ruta pasa por una fase mejora al aplicarle el cambio de cantidades al interior.

El cambio de cantidades al interior de una ruta únicamente modifica las cantidades entregadas de una ruta y modifica únicamente la matriz de asignación. El proceso recorre cada ruta de la solución que tiene más de un 1 PD. Para cada ruta se genera un número aleatorio mediante una distribución uniforme entre cero y uno. Si es menor al parámetro de probabilidad de selección la ruta es seleccionada para aplicarle el cambio. Si la ruta es seleccionada, se le aplica un algoritmo de recocido simulado en el cual la función perturbación intercambia cantidades entre los PD visitados en la ruta. Para realizar el intercambio se selecciona dos PD y una cantidad a intercambiar de manera aleatoria. Se realiza el cambio generando una nueva solución.

## 6. Algoritmo mateheurístico

Raidl y Puchinger [15] indican que sólo en los últimos años muchos investigadores reconocen las ventajas y potencialidades de la construcción de métodos híbridos entre los métodos exactos y (meta)heurísticos. Es importante observar que los métodos exactos y los métodos (meta)heurísticos tienen sus propias ventajas y desventajas, y pueden ser vistos como complementarios, en gran medida. Parece natural combinar las ideas de ambos métodos. En la comunidad científica se ha establecido el termino mateheurística para referirse a los algoritmos híbridos entre métodos exactos y métodos (meta)heurísticos. Puchinger y Raidl [14] presentan una clasificación general de los algoritmos mateheurísticos agrupados en dos categorías:

- Combinaciones colaborativas: En un ambiente de colaboración los algoritmos intercambian información, pero son independientes. Los algoritmos exactos y (meta)heurísticos pueden ejecutarse secuencialmente, en paralelo o entrelazadamente.
- Combinaciones integradas: En los métodos integrados, un algoritmo es un componente subordinado de otro algoritmo.

En la categoría de combinaciones integradas, Caserta y Vob [4] y Raidl y Puchinger [15] identifican dos subcategorías (i) el algoritmo (meta)heurístico es el maestro y controla las llamadas al algoritmo exacto y (ii) el algoritmo exacto es el maestro y llama al algoritmo (meta)heurístico.

```

Paso 1: Inicialización
s = solución inicial
temp =  $T_{ini}$  (Temperatura inicial)
 $\alpha$  = factor de reducción
iter = número de iteraciones
 $T_{fin}$  = Temperatura final (criterio de paro)
Paso 2: Búsqueda
mientras Temp >  $T_{fin}$ 
    desde x=1 a iter
        Paso 3: Nueva solución
        (i)  $sp_{rutas}$ , establecer la solución parcial ruta ( $s = sp_{rutas} + sp_{asig}$ )
             $sp_{asig}$ , establecer la solución parcial asignación
        (ii)  $sp_{newrutas} = Perturbar (sp_{rutas})$ 
        (iii)  $P(sp_{newrutas})$ , establecer Problema de asignación de ayuda humanitaria
        (iv)  $sp_{newasig} = Optimizar P(sp_{newrutas})$ 
        (v)  $s_{new} = sp_{newrutas} + sp_{newasig}$ 
        (vi) Depurar ( $s_{new}$ )
        Paso 4: Comparación
 $\Delta E = c(s_{new}) - c(s)$ 
si  $\Delta E > 0$  entonces
     $s = s_{new}$ 
sino
    si aleatorio  $[0,1] < \exp(\Delta E / temp)$  entonces
         $s = s_{new}$ 
    fin si
fin si
fin desde
temp =  $\alpha * temp$ 
fin mientras
Paso 5: Solución final s

```

\* el operador + une dos soluciones parciales para formar una solución completa

Fig. 5. Algoritmo mateheurístico para el PDAH.

El algoritmo mateheurístico desarrollado es un algoritmo de recocido simulado con una función perturbación más sofisticada que incluye la resolución de un subproblema de programación entera. El pseudo código del algoritmo mateheurístico desarrollado se presenta en la figura 5. El algoritmo inicia con la definición de los parámetros del algoritmo. Se define la solución inicial de manera aleatoria, de forma igual a la presentada para el algoritmo de recocido simulado. El paso 2 inicia el algoritmo de recocido simulado con el ciclo exterior e interior. El paso 3 genera una nueva solución a partir de la solución actual. Las etapas son: (i) Se separa la solución actual (s) en dos partes la solución parcial rutas ( $sp_{rutas}$ ) y la solución parcial asignación ( $sp_{asig}$ ). (ii) La solución parcial rutas ( $sp_{rutas}$ ) se modifica al aplicarle varias veces el cambio de la ruta de un vehículo, definiendo la nueva solución parcial rutas ( $sp_{newrutas}$ ). (iii) Se define un problema de asignación de ayuda humanitaria asociado a la nueva solución parcial rutas ( $P(sp_{newrutas})$ ). (iv) El problema se resuelve óptimamente al utilizar programación

entera generando la nueva solución parcial asignación ( $sp_{newasig}$ ). (v) En esta etapa se unen las dos soluciones parciales nueva generando una nueva solución ( $s_{new}$ ). (vi) Se depura la nueva solución ( $s_{new}$ ) eliminando de las rutas a los PD que no reciben ayuda humanitaria. Continuando con el algoritmo principal de recocido simulado, se realiza la comparación entre la nueva solución ( $s_{new}$ ) y la solución actual ( $s$ ). De la comparación puede que se produzca un cambio en la solución actual ( $s$ ). El algoritmo termina cuando la temperatura alcanza el criterio de paro.

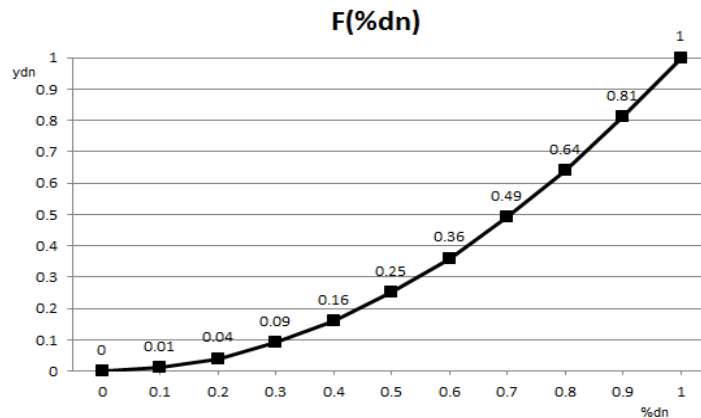
## 7. Implementación y resultados experimentales

El presente trabajo define un conjunto de instancias de manera aleatoria. Se definieron dos tipos de instancias: pequeñas y grandes. Las instancias pequeñas están formadas por 10 PD y las instancias grandes están formadas por 20 PD. La localización de los PD fue generada de manera aleatoria según una distribución uniforme en un plano con un rango de 100 X 100 unidades de distancia. En todas las instancias la localización del CDL es en el centro del plano con coordenadas [50,50]. Los tiempos de recorrido entre los nodos se establece según la distancia euclídea redondeada hacia abajo en unidades de tiempo. La demanda de cada PD en cada periodo fue genera a partir de una distribución uniforme ( $U[1000,6000]$ ). Una vez establecida la demanda de cada PD para cada día, se establece la oferta disponible para cada periodo también utilizando una distribución aleatoria uniforme. Para la oferta disponible se establece entre el 80 y el 95 % de la demanda de ese periodo. Se recuerda que la capacidad de transporte está definida por cuatro elementos: cantidad de vehículos, capacidad de carga, tiempo de trabajo máximo (TTM) y el número de rutas. Se definen instancias con 2 vehículos y capacidad de carga de 8000 y 6000, para el tipo pequeño y con 2 vehículos y capacidad de carga de 16000 y 12000, para el tipo grande. El TTM se presenta con dos opciones por cada opción de cantidad de vehículos y capacidad de carga. El número de rutas máximo que puede realizar un vehículo esta limitado a 2 rutas por periodo. En consecuencia, se identifican cuatro grupos definidos por el número de PD y el número de vehículos. Para cada tipo de problema se generaron 6 problemas base en sus datos de localización, tiempos de recorrido, demanda y oferta. El número total de instancias es de 24.

Los pesos de la función objetivo se mantuvieron constantes en todas las instancias. Sus valores son: P3: 100, P2: 40 y P1: 10. La función convexa lineal por segmentos utilizada para el objetivo de balance en la demanda no satisfecha se presenta en la figura 6 y está compuesta de 10 segmentos.

utilizado es la técnica exacta de programación entera con el algoritmo brach-and-bound y se utiliza el software Cplex versión 11.0. Para terminar la ejecución se determina un tiempo de proceso límite de 7200 segundos. El segundo método utilizado es el algoritmo de recocido simulado desarrollado, el algoritmo es implementado en Python 2.7.2. Los valores de los parámetros utilizados en el algoritmo son:  $Temp_{ini}$ : 10000,  $Temp_{fin}$ : 1000, factor  $\alpha$ : 0.95 e iter: 5 y  $\epsilon$ : 0.75. Los parámetros del cambio de cantidades al interior de una ruta son:  $Temp_{ini}$ : 10000,  $Temp_{fin}$ : 6000, factor  $\alpha$ : 0.95 y iter: 5. El tercer método utilizado es un algoritmo mateheurístico desarrollado. Este algoritmo es implementado en Python 2.7.2 y el problema de programación entera es resuelto por el software Gurobi 4.6. Los valores

de los parámetros del algoritmo son:  $Temp_{ini}$ : 10000,  $Temp_{fin}$ : 1000, factor  $\alpha$ : 0.95 e iter: 20. Los valores de los parámetros fueron determinados en base al conocimiento de los algoritmos desarrollados y de su desempeño en experimentos iniciales del PDBAH. Todos los experimentos se realizaron en una computadora con un procesador AMD Athlon 64 de 1.80 GHz con 2.00 Gb de memoria RAM. Todos los tiempos de proceso en este trabajo están reportados en segundos.



**Fig. 6.** Función convexa lineal por segmentos de la demanda no satisfecha.

Para resolver las instancias definidas se utilizan tres métodos. El primer método El algoritmo recocido simulado y el algoritmo mateheurístico utilizan números aleatorios, por lo cual estos métodos realizan 5 replicas de resolución de cada instancia. Los resultados presentados reportan los promedios del valor del objetivo y del tiempo de proceso o el tiempo de CPU consumido de las 5 replicas. Así mismo, el valor del objetivo es comparado con la cota inferior que se obtiene al resolver las instancias mediante el método exacto. La figura 7 presenta los resultados obtenidos para cada escenario y método. Al final de la figura 7 se presenta los promedios totales de los resultados obtenidos. La columna GAP representa la diferencia en porcentaje entre la solución obtenida y la cota inferior respecto de la cota inferior. En la figura 5 se observa en la columna del valor del objetivo del método exacto que en ocasiones aparece como dato 'No sol', esto indica que el algoritmo no encontró ninguna solución entera que respete las restricciones, al terminar el tiempo de proceso límite.

Esto se demuestra observando que en todos los escenarios obtiene la solución promedio con mejor valor del objetivo y el mejor GAP. El promedio entre todas las instancias del algoritmo mateheurístico obtiene un 0.35% del GAP. El promedio del GAP que obtiene el algoritmo de recocido simulado (SA) es el 7%. El método exacto obtiene un promedio del 208% del GAP, en 1 instancia del tipo grande no logra obtener una solución entera factible. También se puede comparar los resultados obtenidos por el algoritmo de recocido simulado y por el algoritmo mateheurístico en la figura 8.

Distribución balanceada de ayuda humanitaria resuelto con un algoritmo metaheurístico

Instancia	Problema	PD	TTM	Exacto				SA			MATEHEURISTICO		
				Valor	Cota	GAP	Tiempo	Valor	GAP	Tiempo	Valor	GAP	Tiempo
				Función Obj	Inferior			Función Obj			Función Obj		
1	1	10	300	5,121,408	1,916,588	167%	7,200	2,162,580	13%	634	1,918,003	0.07%	129
2	2	10	300	11,226,810	3,820,891	194%	7,200	4,171,668	9%	556	3,828,887	0.21%	121
3	3	10	300	5,981,182	3,032,790	97%	7,200	3,124,079	3%	537	3,036,039	0.11%	127
4	4	10	300	11,034,009	2,743,189	302%	7,200	3,849,502	40%	449	2,876,119	4.85%	116
5	5	10	300	14,372,314	3,197,490	349%	7,200	3,267,655	2%	606	3,200,647	0.10%	127
6	6	10	300	3,985,805	3,013,989	32%	7,200	3,353,307	11%	494	3,022,965	0.30%	128
7	1	10	400	1,916,593	1,916,588	0%	7,200	1,920,908	0%	713	1,916,588	0.00%	149
8	2	10	400	4,797,129	3,820,891	26%	7,200	3,836,965	0%	602	3,820,891	0.00%	135
9	3	10	400	3,032,796	3,032,790	0%	7,200	3,052,907	1%	617	3,032,790	0.00%	140
10	4	10	400	2,760,085	2,743,189	1%	7,200	2,767,675	1%	634	2,743,189	0.00%	139
11	5	10	400	3,197,495	3,197,490	0%	7,200	3,198,478	0%	673	3,197,490	0.00%	147
12	6	10	400	5,467,352	3,013,989	81%	7,200	3,025,753	0%	627	3,013,989	0.00%	139
13	1	20	400	32,096,500	5,275,081	508%	7,200	5,815,175	10%	1,156	5,350,924	1.44%	240
14	2	20	400	No Sol	8,529,787		7,200	8,752,039	3%	1,045	8,529,965	0.00%	249
15	3	20	400	23,507,754	3,482,580	575%	7,200	4,035,957	16%	1,265	3,486,212	0.10%	245
16	4	20	400	13,747,862	6,158,087	123%	7,200	6,412,637	4%	1,055	6,168,677	0.17%	254
17	5	20	400	31,779,275	4,217,081	654%	7,200	4,667,486	11%	1,094	4,253,217	0.86%	252
18	6	20	400	37,145,737	9,390,788	296%	7,200	9,485,239	1%	1,120	9,390,790	0.00%	269
19	1	20	600	25,400,108	5,275,081	382%	7,200	6,061,551	15%	1,187	5,275,082	0.00%	259
20	2	20	600	10,145,539	8,529,787	19%	7,200	8,626,229	1%	1,049	8,529,787	0.00%	261
21	3	20	600	26,715,600	3,482,580	667%	7,200	4,090,414	17%	1,286	3,482,581	0.00%	252
22	4	20	600	6,248,632	6,158,087	1%	7,200	6,313,219	3%	1,078	6,160,792	0.04%	260
23	5	20	600	4,289,479	4,217,081	2%	7,200	4,742,448	12%	1,105	4,222,815	0.14%	256
24	6	20	600	37,391,000	9,390,788	298%	7,200	9,472,025	1%	1,064	9,390,788	0.00%	264
Promedio:						208%	7,200		7%	860		0.35%	194

Fig. 7. Resultados de la experimentación.

Los resultados muestran que el algoritmo mateheurístico tiene el mejor desempeño. Respecto del tiempo de proceso, se observa que el algoritmo mateheurístico requiere un consumo de tiempo de proceso inferior a los otros métodos, 194 segundos en promedio frente a los 860 segundos del algoritmo de recocido simulado. El método exacto siempre termina al alcanzar el límite de tiempo. En la figura 5 se observa que al duplicar el número de puntos de distribución se duplica el tiempo de proceso, en el caso del recocido simulado y el algoritmo mateheurístico.

El TTM es un elemento que dificulta de manera inversa la resolución del problema. En la figura 5 en la columna del GAP del método exacto se observa que obtiene mejores soluciones (con el mismo tiempo de proceso) cuando el TTM es alto en comparación con un TTM bajo. Esto es debido a que un TTM bajo es más restrictivo que un TTM alto. Sin embargo esta tendencia no se observa tan claramente en los otros métodos, salvo en las instancias con 10 PD resueltas con el algoritmo de recocido simulado. Se observa que el valor del TTM no afecta al tiempo de proceso.

La figura 5 se observa que en algunas instancias el método exacto obtiene buenas soluciones con GAP inferiores a 5% en 6 instancias y que los peores resultados se presentan cuando el TTM es bajo. El algoritmo de recocido simulado obtiene buenos resultados en 14 instancias con GAP menores al 5%.

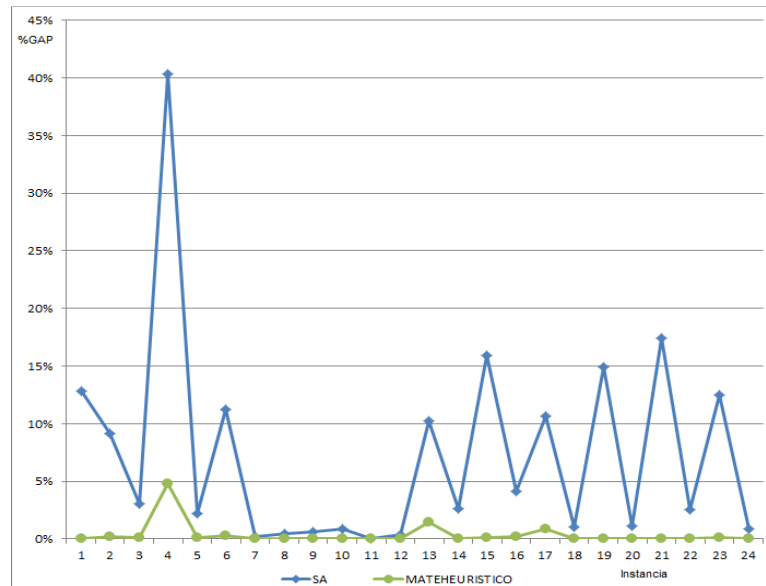


Fig. 8. Gráfica de los resultados en el GAP del recocido simulado y del mateheurístico.

## 8. Conclusiones

En el presente trabajo se define un modelo para el PDABH. También se ha desarrollado un algoritmo de recocido simulado y un algoritmo mateheurístico.

Una dificultad del PDABH es definir su modelo. En este trabajo desarrolla un modelo a partir de la formulación de dos índices del problema de ruteo. Primero se modifica al incluir múltiples periodos por la limitación de recursos. La limitación en la ayuda humanitaria también obliga a satisfacer parcialmente la demanda de los puntos de distribución. La tercera modificación es incluir la satisfacción tardía de la demanda, inventarios en los puntos de distribución y la realización de múltiples rutas. La última modificación es incluir de manera explícita la realización de una distribución balanceada a través de una función convexa lineal por segmentos para evitar grandes diferencias en el servicio recibido por los puntos de distribución.

El algoritmo mateheurístico desarrollado es comparado con el método exacto brach-and-bound y el algoritmo de recocido simulado. El algoritmo mateheurístico desarrollado descompone la solución en una parte de ruteo y otra parte de asignación. La parte de ruteo es la más difícil y por ello es tratada en el algoritmo principal basado en el algoritmo de recocido simulado y la parte de asignación más sencilla es tratada con programación entera. Para la comparación del desempeño se definen 96 instancias que son resueltas por los tres métodos. En base a los resultados obtenidos se concluye que el algoritmo mateheurístico desarrollado se desempeña mejor que las otras técnicas en todas las instancias en términos del valor del objetivo y del tiempo de proceso utilizado.



## Referencias

1. Altay N., Green III W.G., OR/MS research in disaster operations management. *European Journal of Operational Research*, 175: 475-493, 2006.
2. Balci B., Beamon B.M., Smilowitz K., Last Mile Distribution in Humanitarian Relief. *Journal of Intelligent Transportation Systems*, 12(2): 51-63, 2008.
3. Barbarosoglu G., Ozdamar L., Cevik A., An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations. *European Journal of Operational Research*, 140: 118-133, 2002.
4. Caserta M., Vob S., Metaheuristics: intelligent problem solving. En Maniezzo V., Stutzle T., Vob S. *Metaheuristics*, New York (NY), Springer, 2009, pp. 1-38.
5. De Angelis V., Mecoli M., Nikoi C., Storch G., Multiperiod integrated routing and scheduling of World Food Programme cargo planes in Angola. *Computers Operations Research*, 34: 1601-1615, 2007.
6. Dowsland K., Díaz B., Diseño de heurísticas y fundamentos del recocido simulado. *Revista Iberoamericana de Inteligencia Artificial*, 20: 34-52, 2001.
7. Ergun O., Karakus G., Keskinocak P., Swann J., Villarreal M., Operations Research to improve disaster supply chain management. *Wiley Encyclopedia of Operations Research and Management Science*, Hoboken (NJ), John Wiley & Sons, 2011.
8. Huang M., Balci B., Smilowitz K., Models for relief routing: Equity, efficiency and efficacy. *Transportation Research Part E*, 48(1): 2-18, 2012.
9. Kirkpatrick S., Gelatt C.D., Vecchi M.P., Optimization by Simulated Annealing. *Science*, 220(4598): 671-680, 1983.
10. Kovacs G., Spens K.M., Humanitarian logistics in disaster relief operations. *International Journal of Physical Distribution & Logistics Management*, 37(2): 99-114, 2007.
11. Lin Y., Batta R., Rogerson P.A., Blatt A., Flanagan M. A logistics model for delivery of critical items in a disaster relief operation: heuristic approaches. Disponible en: <http://www.acsu.buffalo.edu/~batta/tre.pdf>.
12. Nolz P.C., Doerner K.F., Gutjahr W.J., Hartl R.F., A bi-objective metaheuristic for disaster relief operation planning. En Coello Coello C., Dhaenens C., Jourdan L., *Advances in multi-objective nature inspired computing*, Berlin, Springer Berlin / Heidelberg, 2010, pp. 167-187.
13. Perez N., Holguin-Veras J., Mitchell J.E., Sharkey T.C., Integrated vehicle routing problem with explicit consideration of social costs in humanitarian logistics. En proceso de publicación, 2010. Disponible en: <http://transp.rpi.edu/~HUM-LOG/Doc/IRP.pdf>.
14. Puchinger J., Raidl G.R., Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. *Lecture Notes in Computer Science*, 3562: 41-53, 2010.
15. Raidl G.R., Puchinger J., Combining (Integer) Linear Programming Techniques and Metaheuristics for Combinatorial Optimization. *Studies in Computational Intelligence*, 114: 31-62, 2008.
16. Sheu J., An emergency logistics distribution approach for quick response to urgent relief demand in disasters, *Transportation Research Part E*, 43: 687-709, 2007.
17. Toth, P. Vigo, D. The vehicle routing problem. *Society for industrial and applied mathematics*, Philadelphia (PA), 2002.
18. Van Wassenhove L.N., Humanitarian aid logistics: supply chain management in high gear. *Journal of the Operational Research Society*, 57(5): 475-489, 2006.
19. Yi W., Ozdamar L., A dynamic logistics coordination model for evacuation and support in disaster response activities. *European Journal of Operational Research*, 179: 1177-1193, 2007.

