Evaluación Experimental del Problema de Flujo No Divisible de Costo Mínimo con Única Fuente Mediante la Aplicación de Operadores Genéticos

Salazar F. *

* *Escuela Politécnica Nacional, Facultad de Ciencias Quito, Ecuador (e-mail: fernanda.salazar@epn.edu.ec)*

Resumen: *En este trabajo se considera el problema de encontrar un flujo que satisfaga la demanda de ciertos productos, en una red con capacidades y costos, de manera que cada demanda sea enviada por un único camino hacia su destino, el costo total sea mínimo y la capacidad de los arcos se viole por el menor factor posible. El problema expuesto es NP-difícil. El mejor algoritmo que se conoce, da una aproximación (3,1) para la congestión y costo respectivamente. Sin embargo, en el 2001 Goemans conjeturó que es posible encontrar una aproximación (2,1) para el problema.Partiendo de un grupo de instancias utilizadas en trabajos previos, proponemos realizar una evaluación exhaustiva de la conjetura de Goemans de acuerdo a un modelo de optimización de programación entera, generando nuevas instancias de prueba a través de operadores genéticos. En este trabajo se tomaron siete instancias iniciales, y se obtuvieron 844 nuevas instancias de prueba. Para las 132 instancias factibles se verificó la conjetura de Goemans y se mejoró la población con 9 nuevos individuos.*

Palabras clave: *flujo no divisible, congestión, algoritmo de aproximación, operador genético.*

Abstract: *We consider the problem of finding an unsplittable flow that satisfies the demand of a given set of products on a capacitated network with arc costs, such that the total cost is minimum and arc capacities are violated by the lower possible factor. The stated problem is NP-hard. The best known algorithm for the problem gives a (3,1)-approximation for congestion and cost respectively. However, since 2001, there exists a conjecture by Goemans saying that it is possible to find a (2,1)-approximation for the problem.Here, taking a number of instances used previously in the literature, we propose a thorough evaluation of Goemans' conjecture according to an integer programming model and genetic operators.This work starts with seven instances and 844 new instances were obtained. Goemans' conjecture holds for all 132 feasible instances and nine better individuals in terms of congestion have been added to theinitial population.*

Keywords: *unsplittable flow, congestion, approximation algorithm, genetic operators.*

1. INTRODUCCIÓN

El problema de hallar un flujo no divisible de costo mínimo y congestión acotada fue introducido en 1997 [3]. Sea G=(V,A)una red dirigida donde V es el conjunto de nodos y A es el conjunto de arcos de la red, esto es, $A = \{(i,j)|i \in V, j \in V\}$, con $|V| = ny|A| = m$. Se consideran capacidades μ_{ij}, flujo f_{ij} y costos c_{ij} sobre cada arco (i,j), así como también K productos, cada uno de ellos con una demanda d_k y un nodo de destino t_k, para k=1,...,K, respectivamente. El valor de la demanda máxima de entre los K productos se nota por d_{max}. El objetivo es enviar cada una de las demandas desde un único nodo fuente s hacia su respectivo destino t_k, usando un único camino en la red, violando la capacidad de los arcos por el mínimo factor posible (congestión) y de manera que el costo sea mínimo o no sobrepase un presupuesto dado. Este problema pertenece a la clase de problemas NP-duros. Tiene varias aplicaciones en áreas como transporte, telecomunicaciones, etc. Para la versión sin costos, Dinitz et al [2] demostraron que las demandas pueden ser satisfechas en forma no divisible con la garantía de que el flujo final de cada arco es menor que dos veces su capacidad. Por otro lado, al incluir costos, Skutella [6] desarrolló un algoritmo que encuentra un flujo no divisible que satisface todas las demandas con la garantía de que el flujo final de cada arco es menor que tres veces su capacidad y el costo del flujo obtenido es mínimo o respeta un presupuesto dado, esto es, una aproximación (3,1). Finalmente, Goemans [6] conjeturó en el año 2001 que se puede hallar una aproximación (2,1) para el problema con costos. La conjetura de Goemans se define como sigue:

Conjetura. Para cualquier función de costos c: A→ R, un flujo fraccional f^{init} que satisface demandas d_k k=1,...,K, puede transformarse en un flujo no divisible f que satisface el mismo conjunto de demandas, tal que:

$$f_{ij} \leq f_{ij}^{init} + d_{max} \tag{1}$$

para todos los arcos $a \in A$, y el costo de f es acotado por el costo de f^{init}, es decir:

$$\sum_{a\in A} c_{ij} f_{ij} \leq \sum_{a\in A} c_{ij} f_{ij}^{init} \tag{2}$$

El avance más significativo en el estudio de esta Conjetura se da en el año 2007. Martens et al [5] demostraron que todo flujo fraccional que satisface un grupo de demandas dadas, donde fraccional significa que se permite que las demandas sean satisfechas a través de varios caminos, puede escribirse como combinación convexa de flujos no divisibles de congestión menor a dos, cuyo costo no sobrepasa el costo del flujo fraccional inicial. Las demandas en cada uno de los flujos no divisibles que forman la combinación convexa difieren ligeramente de las demandas originales, caso contrario se hubiese demostrado la conjetura. En dicho estudio se incluye una parte experimental suficiente para probar el algoritmo desarrollado, sin embargo, se hace evidente la necesidad de generar más pruebas sobre la verificación de la conjetura de Goemans. Con este objetivo, se plantea el presente estudio experimental que crea nuevas instancias utilizando operadores genéticos y evalúa la conjetura de Goemans exhaustivamente a través de un modelo de optimización de programación entera.

2. MATERIAL Y MÉTODOS

2.1 Material

Para resolver el problema de hallar un flujo de costo mínimo, que aparece como subproblema del modelo de optimización, se ha utilizado el programa mcf-1.3 [4]. Para resolver el modelo de optimización se ha utilizado el software SCIP (Solving Constraint Integer Programs) versión 2.0.1 [1]. Los programas se han desarrollado en lenguaje C++ y se han compilado con gcc 4:4.5.2-1ubuntu3. Las pruebas experimentales se han llevado a cabo en un computador con sistema operativo Ubuntu 11.04, 2GB RAM y procesador AMD Athlon (tm) 64 X2 Dual Core Processor 4800+.

2.2 Métodos

Para el problema planteado se ha diseñado un modelo de optimización entero que calcula el flujo no divisible de congestión mínima y de costo menor al costo mínimo de un flujo fraccional sobre cada instancia. Así, si la congestión es menor o igual que dos, se confirma la validez de la conjetura de Goemans. El modelo de programación entera se muestra a continuación en donde, $\forall k$, se entiende como $\forall k = 1, \ldots, K$.

$$\min \quad \alpha$$

$$\text{sujeto a}$$

$$\sum_{(s,j)\in A} x_{sj}^k - \sum_{(j,s)\in A} x_{js}^k = 1, \quad \forall k \tag{3}$$

$$\sum_{(i,j)\in A} x_{ij}^k - \sum_{(j,i)\in A} x_{ji}^k = 0, \quad \forall i \in V\backslash\{s, t_k\}, \forall k \tag{4}$$

$$\sum_{k=1}^{K} d_k x_{ij}^k - \alpha\mu_{ij} \leq 0, \quad \forall (i,j) \in A \tag{5}$$

$$\sum_{k=1}^{K} \sum_{(i,j)\in A} c_{ij} d_k x_{ij}^k \leq \beta \tag{6}$$

$$x_{ij}^k \in \{0,1\}, \quad \forall (i,j) \in A, \forall k \tag{7}$$

En este modelo α es el valor de la congestión y β es el valor del costo de un flujo fraccional de costo mínimo. Las variables de decisión x_{ij}^k toman el valor de 1 si el arco (i,j) es utilizado para enviar la demanda k a su destino, y el valor de 0 en caso contrario. El grupo de restricciones (3) y (4) garantizan que cada demanda es enviada a lo largo de un único camino desde la fuente hacia su correspondiente destino. Las restricciones que se indican en (5) indican que a través de cada arco no se envía más flujo que α veces la capacidad del arco. Finalmente, la restricción (6) prohíbe que el costo del flujo no divisible tenga un costo mayor al costo del flujo fraccional de costo mínimo. Por tanto la solución del modelo es el flujo no divisible que se está buscando. Este modelo tiene $mK + 1$ variables y $(n-1)K + m + 1$ restricciones, donde n es el número de nodos, m es el número de arcos y K es el número de demandas en la red.

El valor de β puede obtenerse fácilmente ya que es la solución de un problema básico de la optimización combinatoria, un problema de flujo de costo mínimo.

Para el estudio experimental se han tomado como base instancias que han sido usadas en investigaciones previas y que se encuentran bien referenciadas en la literatura, están dadas por los generadores noigen y rangen [5]. Las operaciones de evolución de instancias abarcan los dos estilos típicos, evolución por mutación y evolución por cruzamiento genético. El proceso de generación de nuevas instancias funciona de acuerdo a los operadores genéticos que se detallan a continuación.

Mutación 1: El único limitante teórico para el problema de estudio es la *condición de balance*. Esta condición exige que la capacidad mínima de los arcos sea mayor o igual que la demanda máxima. Así, la instancia dada por Mutación 1 resulta de disminuir el valor de la capacidad de cada uno de los arcos al valor de la demanda máxima pues parece ser lógico que, si se conservan las demandas originales de la red pero a la vez se disminuye su capacidad, se generen congestiones cada vez más altas al tratar de satisfacer tales demandas.

Mutación 2: Un segundo análisis sugiere que si se aumenta el valor de las demandas en una red sin cambiar su capacidad original, la red aumentará su saturación generando eventualmente congestiones cada vez más altas. Por lo expuesto, y cumpliendo la *condición de balance*, Mutación 2 consiste en subir el valor original de cada una de las demandas de la red al valor de la demanda máxima.

Mutación 3: En esta operación se conservan las capacidades originales y se aumenta el valor de las demandas de la red. El aumento se realiza de demanda en demanda, dejando a las demás con sus valores originales, mientras la red siga teniendo una solución fraccional factible. Este proceso genera, para cada instancia original, *K-1* nuevas instancias.

Suma: Partiendo de dos instancias iniciales se construye la instancia descendiente de la siguiente forma: Si las instancias iniciales tienen *n1* y *n2* nodos respectivamente, y *n1<n2*, la nueva instancia tendrá *n2* nodos, caso contrario tendrá n1 nodos. Luego, la nueva instancia hereda cada arco no repetido con sus respectivos valores de capacidad y costo. Para arcos repetidos, la capacidad y el costo del arco en la instancia descendiente se calculan como la suma de las capacidades y costos de ese arco en las instancias predecesoras. Finalmente, la nueva instancia hereda cada una de las demandas de las instancias iniciales. Si un mismo nodo v tiene un valor de demanda diferente de cero *d1* y *d2* en las instancias predecesoras, entonces la demanda en el nodo v en la instancia descendiente es *d1+d2*. A partir de I instancias, este proceso genera $\sum_{(i=1)}^{I}(i-1)$ nuevas instancias al realizar la suma de dos en dos de todas ellas.

Combinación: Partiendo de dos instancias iniciales, el número de nodos y los arcos no repetidos de la instancia descendiente se obtienen por el mismo proceso que en la operación Suma. Para arcos repetidos, la capacidad y el costo del arco en la instancia descendiente se calculan como el valor máximo de las capacidades y costos que tiene el arco en las instancias predecesoras, respectivamente. La nueva instancia hereda cada una de las demandas de las instancias iniciales. Si un mismo nodo v tiene un valor de demanda diferente de cero *d1* y *d2* en las instancias predecesoras,entonces la demanda en el nodo v de la instancia descendiente es el valor máximo entre *d1* y *d2*.

Al aplicar cada uno de los operadores genéticos anteriores se obtienen una o varias instancias nuevas. Sobre cada una de ellas se prueba la validez de la Conjetura de Goemans a través del modelo de optimización y se guardan aquellas instancias que resultan ser mejores que sus predecesoras, esto es, que tienen un valor de congestión más alto. Con esas instancias se pueden aplicar nuevamente los operadores genéticos. Este procedimiento se resume en la Figura 1.

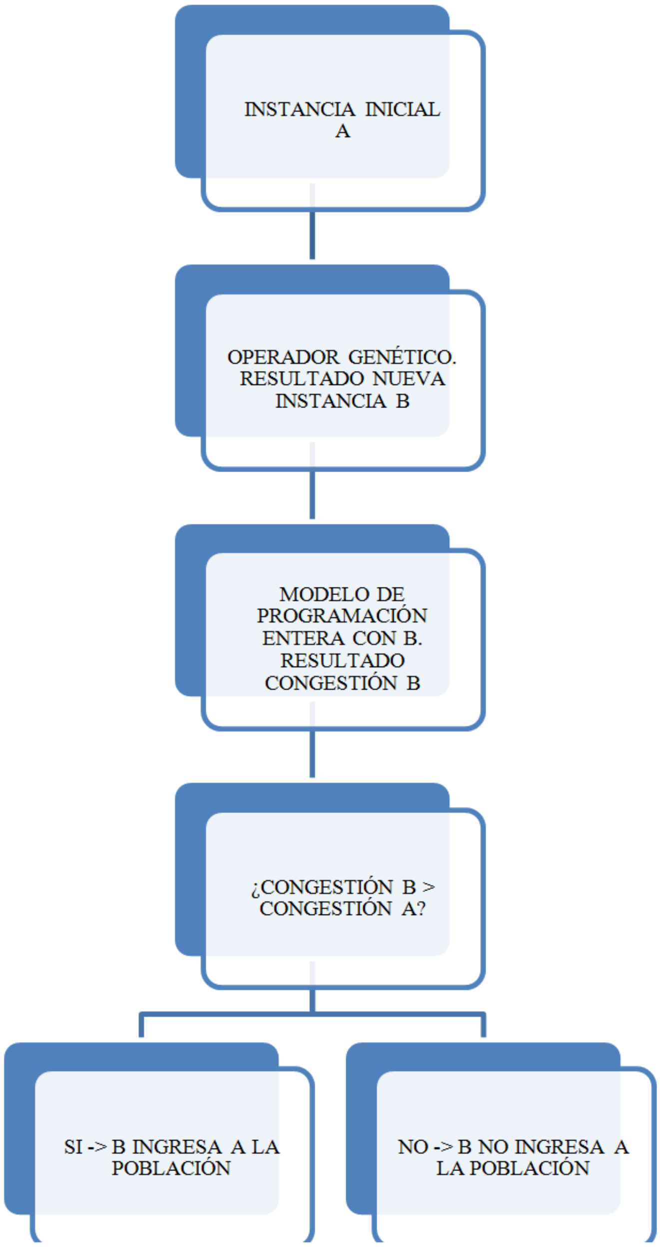

Figura 1. Generación de instancias mediante operadores genéticos

3. ANÁLISIS

Como instancias de prueba iniciales se han escogido a elementos representativos de cada generador. De la familia noigen las instancias s1, s2, s3 y s5 y de la familia rangen las instancias rs1, rs3 y rs4. La solución óptima dada por el modelo entero que se definió en la sección anterior para estas instancias, en donde n es el número de nodos, m es el número de arcos y K es el número de productos, se muestra en la tabla 1. Las congestiones resultantes se utilizarán con fines comparativos para definir si las nuevas instancias son fuertes o débiles, esto es, si se obtiene un valor de congestión mayor o menor, respectivamente.

Tabla 1. Congestión para las instancias originales

Instancia	n	m	k	Congestión
s1	53	54	13	1.34
s2	117	128	27	1.19
s3	202	216	54	1.29
s5	526	585	255	1.49
rs1	51	61	32	1.47
rs3	200	260	128	1.4
rs4	401	533	256	1.47

El proceso de generación de nuevas instancias mediante operadores genéticos parte de identificar criterios que puedan ayudar en la evolución de las instancias. El primer criterio que se tomó en cuenta fue el tamaño de la instancia. Este criterio se descartó al comprobar que algunas instancias grandes son de más fácil solución que algunas instancias pequeñas. Luego, se definieron las operaciones de Mutación, Suma y Combinación que se explicaron en la sección anterior.

Los resultados de aplicar los operadores de Mutación, se resumen en la tabla 2 y se explican en detalle a continuación.

Tabla 2. Resultados de las operaciones de mutación

Operación	Instancias nuevas	Instancias factibles	Instancias fuertes
Mutación 1	7	2	0
Mutación 2	7	0	0
Mutación 3	753	53	2
Total	**767**	**55**	**2**

La operación Mutación 1 fue demasiado fuerte pues, al disminuir la capacidad de la red, ésta pierde su capacidad de satisfacer las demandas originales. Notemos que esta operación produce una instancia nueva por cada instancia de prueba, en nuestro caso se obtienen 7 nuevas instancias. Se produjeron solamente 2 instancias factibles y 5 instancias no factibles, es decir, instancias para las que no existe una solución fraccional inicial. Además, las dos nuevas instancias factibles no generaron mejores congestiones que las instancias originales por lo que no se las utiliza en futuras recombinaciones. En la tabla 3 se muestran los resultados obtenidos, una x significa que la nueva instancia es no factible.

Tabla 3. Congestión luego de aplicar la operación Mutación 1.

Instancia	Variables	Restricciones	Congestión
s1	702	731	x
s2	3321	3256	x
s3	11664	11071	x
s5	149175	134461	x
rs1	1952	1662	x
rs3	33280	25733	1.4
rs4	136448	102934	1.47

La operación Mutación 2 produce una instancia nueva por cada instancia de prueba, en nuestro caso se obtuvieron 7 nuevas instancias, todas ellas no factibles. Se concluye que el criterio de aumentar el valor de todas las demandas al valor de la demanda máxima original, sin hacer cambios en la capacidad de la red, es demasiado fuerte.

Con la operación Mutación 3 se encontraron 700 instancias no factibles y 53 instancias factibles. En las 53 instancias factibles se verificó la conjetura de Goemans y se encontró que 2 mejoran la población, aquellas con congestión de 1.47 para la instancia rs3 de la tabla 5.

En las tablas 4, 5 y 6 se presentan únicamente aquellas instancias para las que se ha encontrado una solución inicial factible. La columna N indica el número de la instancia en la que se encontró una solución factible, de entre las K-1 generadas, y la columna k indica la demanda que aumentó su valor al de la demanda máxima original, dejando inalteradas las demás. Las columnas Inst, Var, Rest y Cong. indican el nombre de la instancia, el número de variables, el número de restricciones y la congestión, respectivamente.

Tabla 4: Congestión luego de aplicar la operación Mutación 3 a la familia noigen.

N	Inst	Var	Rest	k	Cong
9	s1	702	731	8	1.34
30	s2	3321	3256	16	1.19
31	s2	3321	3256	17	1.19
39	s2	3321	3256	25	1.12
53	s3	11664	11071	12	1.29

En cuanto a las operaciones de cruzamiento genético, al aplicar la operación Suma sobre las instancias iniciales, se obtuvieron 21 nuevas instancias factibles en las que se verifica la conjetura de Goemans; cinco de ellas resultaron mejores que sus predecesoras, se les ha llamado S1,S2,S3,S4 y S5. Los resultados se muestran en la tabla 7, las instancias que se suman se indican como I1 e I2.
Retroalimentando la operación Suma con tales instancias, se hallaron 30 nuevas instancias factibles, 2 de las cuales mejoran la población, los resultados se muestran en la tabla 8.

Tabla 5: Congestión luego de aplicar la operación Mutación 3 a las instancias rs1 y rs3

N	Inst	Var	Rest	k	Cong
356	rs1	1952	1662	6	1.47
358	rs1	1952	1662	8	1.47
363	rs1	1952	1662	13	1.47
365	rs1	1952	1662	15	1.47
366	rs1	1952	1662	16	1.47
368	rs1	1952	1662	18	1.47
383	rs3	33280	25733	1	1.4
385	rs3	33280	25733	3	1.4
387	rs3	33280	25733	5	1.4
401	rs3	33280	25733	19	1.4
408	rs3	33280	25733	26	1.4
426	rs3	33280	25733	44	1.4
440	rs3	33280	25733	58	1.4
451	rs3	33280	25733	69	1.47
455	rs3	33280	25733	73	1.4
463	rs3	33280	25733	81	1.4
474	rs3	33280	25733	92	1.4
477	rs3	33280	25733	95	1.4
489	rs3	33280	25733	107	1.47
498	rs3	33280	25733	116	1.4
500	rs3	33280	25733	118	1.4
502	rs3	33280	25733	120	1.4
503	rs3	33280	25733	121	1.4

Tabla 6: Congestión luego de aplicar la operación Mutación 3 a la instancia rs4

N	Inst	Var	Rest	k	Cong
520	rs4	136448	102934	10	1.47
532	rs4	136448	102934	22	1.47
534	rs4	136448	102934	24	1.47
539	rs4	136448	102934	29	1.47
553	rs4	136448	102934	43	1.47
559	rs4	136448	102934	49	1.47
560	rs4	136448	102934	50	1.47
563	rs4	136448	102934	53	1.47
567	rs4	136448	102934	57	1.47
593	rs4	136448	102934	83	1.47
612	rs4	136448	102934	102	1.47
633	rs4	136448	102934	123	1.47
648	rs4	136448	102934	138	1.47
654	rs4	136448	102934	144	1.47
661	rs4	136448	102934	151	1.47
662	rs4	136448	102934	152	1.47
664	rs4	136448	102934	154	1.47
704	rs4	136448	102934	194	1.47
706	rs4	136448	102934	196	1.47
716	rs4	136448	102934	206	1.47
725	rs4	136448	102934	215	1.47
735	rs4	136448	102934	225	1.47
738	rs4	136448	102934	228	1.47
745	rs4	136448	102934	235	1.47
751	rs4	136448	102934	241	1.47

Finalmente, la operación Combinación produjo 21 nuevas instancias, todas factibles y en todas se verifica la conjetura de Goemans, sin embargo ninguna mejoró la congestión de sus predecesoras.

Tabla 7: Congestión luego de aplicar la operación Suma a las instancias I1 e

N	I1	I2	Var	Rest	Cong	Nueva instancia
3	rs1	rs3	40320	25788	1.87	S1
5	rs1	rs4	150784	102990	1.56	S2
12	s2	rs3	48640	25853	1.47	S3
14	s2	rs4	167168	103054	1.64	S4
17	rs3	rs4	192000	103151	1.87	S5

Tabla 8: Congestión luego de aplicar la operación Mutación 3 a la instancia rs4

N	I1	I2	Var	Rest	Cong	Nueva instancia
25	S2	Rs1	150784	102990	1.58	S6
27	S2	s2	181248	103109	1.64	S7

4. CONCLUSIONES

El estudio experimental realizado revela la pertinencia y la utilidad del uso de operadores genéticos en este tipo de análisis. Con base en el modelo de optimización y losoperadores genéticos diseñados específicamente para este problema, se generaron 844 nuevas instancias de prueba obtenidas a partir de 7 instancias originales. Se comprobó la validez de la Conjetura de Goemans en las 132 instancias que fueron factibles y se mejoró la población con 9 nuevos individuos. Se puede asegurar que la evolución de las instancias hacia congestiones altas menores que dos, no depende del tamaño de las instancias en número de nodos, arcos o productos, ni tampoco de la congestión de las predecesoras. Sin embargo, no se han podido identificar patrones que garanticen la evolución de las instancias hacia mejores individuos.

REFERENCIAS

[1] Achterberg, “Constraint Integer Programming”, Ph.D. dissertation,Technical University of Berlin, 2007.

[2] Dinitz Y., Goemans M. y Garg N.,“On the single-source unsplittable flow problem”, Combinatorica, vol. 19, pp:17–41, 1999.

[3] Kleinberg J., “Single source unsplittable flow”, in Proceedings of the 37th Annual IEEE Symposium of Foundations of Computer Science, 1996, pp. 68–77.

[4] Löbel.A., “Mcf 1.3. A network simplex implementation”. Zuse Institute Berlin - Software, 2000.

[5] Martens M., Salazar F. y Skutella M.,“Representing single source multicommodity flows as convex combinations of unsplittable flows”, Lectures Notes in Computer Science. Proceedings of the 15th Annual European Symposium on Algorithms, 2007, pp. 395–406..

[6] Skutella M., “Approximating the single source unsplittable min-cost flow problem”, Mathematical Programming, vol. 91, pp.493–514, 2002.