Algoritmos Heurísticos para la Solución del Problema Lineal con Restricciones de Equilibrio

Dania Tamayo-Vera
d.tamayo@lab.matcom.uh.cu
Universidad de La Habana
Gemayqzel Bouza-Allende
gema@matcom.uh.cu
Universidad de La Habana
Antonio Bolufé-Röhler
bolufe@matcom.uh.cu
Universidad de La Habana

RESUMEN

Los problemas lineales con restricciones de equilibrio son un caso particular de los modelos de optimización con restricciones de equilibrio. Debido a la complejidad que presentan, la condición de equilibrio se sustituye por condiciones necesarias obteniéndose un problema con restricciones de complementariedad (MPCC). La estructura del conjunto de soluciones factibles del MPCC obtenido es compleja ya que es la unión de poliedros. Resolver todos los problemas correspondientes a minimizar la función objetivo sobre cada uno de estos poliedros es computacionalmente costoso. El presente trabajo utiliza un enfoque heurístico para dar solución al MPCC, adaptando los algoritmos de Búsqueda Local y Recocido Simulado. Este trabajo presenta un conjunto de funciones de prueba y los resultados computacionales más significativos obtenidos.

PALABRAS CLAVE : Problemas con Restricciones de Equilibrio, Problemas con Restricciones de Complementariedad, Algoritmos Heurísticos, Optimización.

ABSTRACT

Linear equilibrium constrained programming is a special class of optimization models with equilibrium constraints. Because of the complexity of the equilibrium condition it is replaced by necessary conditions, which leads to a complementarity constrained problem (MPCC). The set of feasible solutions in a MPCC is structured as a union of polyhedrons. Solving the MPCC problem would require the minimization of the objective function on each of these polyhedrons. The computation cost of this approach is unfeasible, thus, this work presents a new approach where heuristic algorithms such as Hill Climbing and Simulated Annealing are used to search for good solutions on the polyhedrons space. A new benchmark for linear equilibrium constrained optimization is introduced. The computational results achieved by the proposed heuristics on the new benchmark are presented.

KEY WORDS: Linear Equilibrium Constrained Problems, Mathematical Program with Complementarity Constraints, Heuristics, Optimization

INTRODUCCIÓN

En este trabajo se considera el problema de optimización con restricciones de equilibrio, el cual se formula de la siguiente forma:

$$(P_{EC}) \quad \min f(x,y) \ \text{ s.a } \ (x,y) \in M_{EC}$$

$$M_{EC} = \{(x,y) | \begin{array}{c} y \in Y(x) \\ \phi(x,y,z) \geq 0, \quad \forall\, z \in Y(x) \\ g_j(x,y) \geq 0, \qquad j = 1, \ldots . q \end{array} \}$$

donde

$$Y(x) = \{y | v_i(x,y) \geq 0, \ i = 1, \ldots . l\}, (f, g_1, \ldots, g_q, v_1, \ldots v_l) \in [C^{\infty}]^{1+q+l}_{n+m}$$
$$\phi \in [C^{\infty}]^{1}_{n+m+m} \ \ j = 1, \ldots q, \qquad i = 1, \ldots . l.$$

Los problemas de optimización matemática con restricciones de equilibrio forman una clase especial de modelos de optimización matemática donde las variables de decisión satisfacen un número finito de restricciones junto con una condición de equilibrio. Estos modelos tienen aplicaciones en la economía pues describen matemáticamente equilibrios de Cournot y de Nash. Se caracterizan por la dificultad de saber cuándo un punto es factible.
Los problemas lineales con restricciones de equilibrio (LEC) son una clase de modelos con restricciones de equilibrio en el cual todas las funciones involucradas son lineales. Esto es:

$$(P_{LEC}) \ \min c^T(x,y) \ \text{ s.a } \ (x,y) \in M_{LEC}$$

$$M_{LEC} = \{(x,y) \in R^n \times R^m | \begin{array}{c} [C^T z + d - BY]^T \lambda \geq 0, \ \forall v \in Y(x), \\ y \in Y(x), \\ Y(x) = \{y \in R^m | B(x,y) \geq b\} \end{array} \}$$

donde

$$c \in R^n \times R^m, \qquad B \in R^{l \times (n+m)}, b \in R^l \text{ y } d \in R^m.$$

Debido a la complejidad que añade la restricción de equilibrio $[C^T z + d - BY^T \lambda] \geq 0$, $\forall\ v \in Y(x)$, $y \in Y(x)$, la misma se sustituye por condiciones necesarias para su descripción. De esta forma se obtiene un problema con restricciones de complementariedad (MPCC), ver (Luo et al., 1996).
El problema se puede relajar, usando las condiciones de Karush Kuhn Tucker (KKT), de la siguiente forma:

$$(P_{KKTLEC}) \ \min c^T z \ \text{ s.a } \ (x,y,\lambda) \in M_{KKTLEC}$$

$$M_{KKTLEC} = \{(x,y,\lambda) \in R^{n+m+l} | \begin{array}{c} C^T z + d - BY^T \lambda = 0, \\ Bz \geq b, \\ \lambda \geq 0, \\ (Bz - b)^T \lambda = 0 \end{array} \}$$

Aquí $z = (x, y) \in R^n \times R^m$ y $BY \in R^{l \times m}$ son las columnas de B correspondientes a la variable y.

Esta formulación se estudia en (Luo et al., 2006). En (Bouza, 2006) se describen las propiedades genéricas del conjunto factible y de los puntos de Karush Kuhn Tucker (KKT). Para ilustrar la complejidad del problema, nótese que el conjunto de soluciones factibles del MPCC es unión de una cantidad finita de poliedros (Scholtes y Scheel, 2000; Fukushima y Lin, 2003). De ahí que si se minimiza la función objetivo en cada uno de estos conjuntos, la mejor de las soluciones obtenidas resolvería el problema original. Desde un punto de vista computacional este enfoque es costoso porque la cantidad de problemas a resolver depende exponencialmente de las dimensiones del problema (Bouza, 2006).

Este trabajo propone heurísticas que permiten obtener una solución adecuada del modelo (LEC) sin necesidad de analizar todas las posibles combinaciones. En particular se adaptan los algoritmos de Búsqueda Local (BL) y Recocido Simulado (RS) para resolver el problema LEC de forma eficiente. Para analizar el comportamiento de estas estrategias de búsqueda se creó un conjunto de problemas de prueba, los cuales incluyen distintas dimensiones del problema y posibles combinaciones en cuanto a cantidad de poliedros con soluciones factibles y óptimas.

MARCO TEÓRICO

En los problemas de programación matemática una función de valor real se minimiza en un conjunto factible $M \subset R^n$ descrito por un número finito de restricciones de igualdad y desigualdad. En la mayoría de los casos se supone que las funciones involucradas son C^k-funciones. Un problema P de este tipo es de la forma:

$$\begin{aligned} P:\ \min & f(x) \text{ s.a } x \in M, \\ & h_i(x) = 0, \quad i = 1, \ldots q_0, \\ M = \{x \subset R^n & | g_j(x) \geq 0, \quad j = 1, \ldots q, \} \end{aligned}$$

donde

$$f, h_i, g_j \in [C^k]^1{}_n i = 1, \ldots . q_0, j = 1, \ldots q, k \geq 2.$$

Asumiendo que el conjunto factible M es no vacío y compacto siempre existe $\bar{x} \in M$ que es mínimo (local).

Definición 1 Dado f: $R^n \rightarrow R$, $M \subset R^n$, el punto $\bar{x} \in M$ es un mínimo local de f en M si existe una vecindad $V(\bar{x})$ de $\bar{x}$ tal que:

$$f(x) \geq f(\bar{x}), \forall\ x \in V(\bar{x}) \cap M.$$

Si esta desigualdad se cumple $\forall\ x \in M$ entonces $\bar{x}$ es un mínimo global.

Definición 2 Para $\bar{x} \in M$ el conjunto de índices activos $J_0(\bar{x})$ es denotado como $J_0(\bar{x}) = \{j | g_j(\bar{x}) = 0\}$. La condición LICQ se cumple en $\bar{x}$ si el conjunto de vectores $\{Ah_i(\bar{x}), i = 1, \ldots q_0, Ag_j(\bar{x}), j \in J_0(\bar{x})\}$ es linealmente independiente.

La condición de MFCQ se satisface en $\bar{x}$ si

- $Ah_i(\bar{x}), i = 1, \ldots . q_0$, son linealmente independientes y
- Existe un vector $£ \in R^n$ tal que:

$$\begin{aligned} £^T Ah_i(\bar{x}) &= 0, i = 1, \ldots q_0, \\ £^T Ag_j(\bar{x}) &\geq 0, j \in J_0(\bar{x}). \end{aligned}$$

Definición 3 (Karush Kuhn Tucker) Supongamos que f, g_j para $j \in I$ son diferenciables en $\bar{x}$ y $g_j, j \notin I$ son continuas en $\bar{x}$. Supongamos que se cumple la LICQ. Si $\bar{x}$ es solución del problema P, entones existen escalares $\mu_i, \forall i \in I$ tales que:

$$Af(\bar{x}) + \sum_{j \in I} \mu_i Ag_j(\bar{x}) = 0,$$
$$\mu_j \geq 0, j \in I.$$

APLICACIONES

Los problemas con restricciones de equilibrio tiene aplicaciones en la economía para modelar matemáticamente equilibrios de Cournot y de Nash generalizados (Bouza, 2006). Un problema de optimización crucial en economía es el del mercado de electricidad. En este problema se busca maximizar el bienestar social y minimizar los costos de energía . El pago de electricidad como una oferta de un mercado balanceado es modelado como un problema de dos niveles, en los cuales el nivel inferior se modela como un problema con restricciones de equilibrio. Con ellos se determina desde el punto de vista del productor la oferta óptima para una demanda dada. La existencia de un equilibrio de Nash es probado en (Aussel et al., 2016). También se ha demostrado la existencia de equilibrio de Cournot en estos problemas (Ehrenmann, 2004).

La planificación del transporte es un modelo de tráfico en el cual se necesita una adecuada fluidez. Para los modelos de planificación de transporte son utilizados los problemas de dos niveles, los cuales están estrechamente relacionados con los problemas con restricciones de equilibrio (Migdalas et al., 1995).

ALGORITMOS HEURÍSTICOS

Una heurística es un algoritmo de optimización que garantiza encontrar una solución buena en un tiempo razonable. Estos algoritmos son cada vez más utilizados al resolver complejos problemas de optimización, ya que permiten ganar en eficiencia computacional y simplicidad conceptual.

Los algoritmos de búsqueda local parten de una solución aleatoria que mejoran continuamente a medida que iteran de una solución vecina a otra, hasta cumplir una condición de parada. Existen dos estrategias para evaluar el conjunto de soluciones vecinas: *primera mejora*, en la que se itera hacia la primera solución vecina que mejore la solución actual y *mayor mejora*, en la que se itera a la mejor solución vecina. La presente investigación utiliza la segunda estrategia, que reporta mejores resultados en la literatura (Talbi, 2009). Un algoritmo de búsqueda local itera solamente hasta quedar atrapado en un óptimo local, es decir, una solución que es igual o mejor que todas sus vecinas. Otras heurísticas, como por ejemplo Recocido Simulado, implementan estrategias de búsqueda que permiten escapar de dichos óptimos locales.

Recocido Simulado (RS) se inspira en principios de la física de materiales, donde el recocido de metales requiere el calentamiento del material y luego una disminución lenta de la temperatura para poder alcanzar una estructura cristalina fuerte. Cuando la temperatura es alta, el metal puede transitar a un estado de mayor energía, pero a medida que la temperatura decrece el metal alcanza paulatinamente un estado de poca energía o cristalización fuerte. De manera análoga, el algoritmo de RS puede aceptar soluciones

peores que la actual durante las etapas iniciales del proceso de optimización. A medida que la temperatura disminuye, el algoritmo converge a soluciones de mayor calidad. Esta estrategia de búsqueda permite escapar o evadir óptimos locales (Kirkpatrick, 1984).

La presente investigación utiliza heurísticas de búsqueda local y el algoritmo de Recocido Simulado para encontrar el conjunto de índices activos que proporcionan una mejor solución del problema LEC.

PROPUESTA ALGORÍTMICA

En esta sección se presenta una formulación del problema con restricciones de complementariedad del modelo lineal con restricciones de equilibrio. Se adaptan los algoritmos de Búsqueda Local y Recocido Simulado, explicando las estrategias de búsqueda utilizadas.

Formulación MPCC de modelos LEC

Mediante la utilización de las condiciones de KKT, el problema LEC se representa como un problema con restricciones de complementariedad de la forma siguiente, ver (Bouza, 2006):

$$(P_{KKTLEC}) \min c^T z$$

$$\begin{gathered} \text{s. a } C^T z + d - BY^T \lambda = 0, \\ Bz \geq b, \\ \lambda \geq 0, \\ (Bz - b)^T = 0. \end{gathered}$$

donde $z = (x, y) \in R^{n+m}$ y $BY \in R^{l \times m}$ son las columnas de B correspondientes a la variable y. La representación del conjunto de índices activos para este tipo de problemas queda de la forma siguiente, ver (Ye, 1999):

$$J_0(\bar{z}, \bar{\lambda}) = \{i | [B\bar{z}]_i = b_i, \bar{\lambda_7} > 0 \},$$

$$J\Lambda_0(\bar{z}, \bar{\lambda}) = \{i | [B\bar{z}]_i = b_i, \bar{\lambda_7} = 0 \},$$

$$\Lambda_0(\bar{z}, \bar{\lambda}) = \{i | [B\bar{z}]_i > b_i, \bar{\lambda_7} = 0 \}.$$

Las matrices B_{J_0}, $B_{J\Lambda_0}$ y B_{Λ_0} están compuestas por las filas de B con índices en J_0, $J\Lambda_0$ y Λ_0 respectivamente, donde, para las distintas combinaciones de índices activos que puedan ser fijados, se obtiene un problema lineal:

$$\begin{gathered} P(J_0, J\Lambda_0, \Lambda_0) \min c^T z \\ \text{s. a } B_j z \geq b_j, \quad \lambda = 0, j \in \Lambda_0, \\ B_j z \geq b_j, \quad \lambda = 0, j \in J\Lambda_0, \\ B_j z \geq b_j, \ \lambda = 0, j \in J_0, \end{gathered}$$

$$C^T z + d - BY^T \lambda = 0.$$

El modelo asociado a la terna J_0, $J\Lambda_0$ y Λ_0 que se denota por $P(J_0, J\Lambda_0, \Lambda_0)$, es simple de solucionar. Sin embargo para resolver el problema PKKT LEC, deben considerarse todas las posibles combinaciones de índices activos, lo cual es intratable desde el punto de vista computacional cuando l crece pues serían 3^l posibles combinaciones (Bouza, 2006; Luo et al., 2006).

ESTRATEGIAS DE BUSQUEDA

Para la solución del problema LEC se proponen tres estrategias de búsqueda:

Búsqueda Aleatoria Genera aleatoriamente las ternas (soluciones) del problema. Se toma como punto de referencia para comparar el rendimiento de las demás estrategias de búsqueda.

Búsqueda Local (BL) En cada iteración se evalúan todas las soluciones vecinas y se itera a la mejor de ellas, siempre y cuando sea mejor que la solución actual. El algoritmo se detiene al alcanzar un número máximo de evaluaciones de la función objetivo o al encontrar un óptimo local, i.e. una solución mejor o igual que todas sus vecinas. Se implementaron dos criterios de vecindades distintas:

- Vecindad 1: Un índice pasa de uno de los conjuntos de la terna a otro. Dado que cada índice puede ser desplazado a otros dos conjuntos, se obtiene un total de $2 * l$ soluciones vecinas.
- Vecindad 2: Dos índices (distintos) pasan de uno de los conjuntos de la terna a otro. Dado que por cada par de índices distintos se pueden obtener 4 nuevas soluciones, el conjunto de soluciones vecinas tiene una cardinalidad de $2 * l * (l - 1)$.

Recocido Simulado (RS) Algoritmo clásico de Recocido Simulado en el que se itera a una solución vecina si es mejor que la solución actual o con una probabilidad $P = \exp(\frac{\Delta E}{T})$ en caso de ser peor. Como criterio de vecindad se tomó la Vecindad 2, la temperatura inicial se fijó en $T_0 = 10.000$ y como esquema de enfriamiento se utilizó la ecuación $T = T_0 * (\frac{maxEvals - evals}{maxEvals})^2$, donde maxEvals es el número máximo de evaluaciones permitidas y evals es el número de evaluaciones realizadas por el algoritmo hasta ese momento.

GENERACIÓN DE PROBLEMAS PRUEBA

Para toda experiencia numérica, es necesario usar problemas prueba que ilustren distintos casos patológicos o no. Durante esta investigación se realizó una búsqueda en la literatura y solo aparecen ejemplos académicos, ver (Bouza, 2006). Es por esto que se decide incluir como parte de la investigación, la construcción de un conjunto de problemas de prueba que pueden ser de gran utilidad ya que permiten ver cómo se comportan los algoritmos en problemas de pequeñas dimensiones y cómo el aumento de las dimensiones afecta su rendimiento. Para generar este tipo de problemas prueba, utilizamos todo el análisis matemático de la relajación de LEC mediante las condiciones de KKT.

Si el usuario quiere generar un LEC de dimensiones n, m, l donde el vector z es un punto de KKT. Se asume que el conjunto de índices activos asociado a z es la terna $(J_0, J\Lambda_0, \Lambda_0)$. Para construir el ejemplo, primero se generan de forma aleatoria los vectores α, β, γ que

son los multiplicadores del sistema de KKT. Se genera de forma aleatoria la matriz C y el vector λ con elementos positivos en J_0 y cero para los elementos en $J\Lambda_0$ y Λ_0. Como la variable x ha quedado fija, se genera de forma aleatoria la parte de la matriz B correspondiente a las x. Luego utilizando el sistema presentado en (Bouza, 2006),

$$\begin{pmatrix} c \\ 0 \end{pmatrix} - \begin{pmatrix} C \\ -BY \end{pmatrix} \alpha - \begin{pmatrix} [B^T \quad] \\ {\scriptstyle J_0 \cup J\Lambda_0} \\ 0 \end{pmatrix} \beta - \begin{pmatrix} 0 \\ I_{J\Lambda_0 \cup \Lambda_0} \end{pmatrix} \gamma = 0,$$

$$\begin{aligned} C^T z + d - BY^T \lambda &= 0, \\ B_{J_o} z - b_{J_0} &= 0, \\ B_{J\Lambda_0} z - b_{J\Lambda_0} &= 0, \\ \lambda_{J\Lambda_0 \cup \Lambda_0} &= 0. \end{aligned}$$

se reconstruye la matriz BY.

Para reconstruir BY se considera el sistema:

$$-BY\alpha - I_{J\Lambda_0 \cup \Lambda_0} \gamma = 0,$$

primero se genera una matriz de rango 1 que cumple esta condición, la cual se usa como base y luego se completa adecuadamente. Generada la matriz B se determina c. Análogamente con los bloques:

$$\begin{aligned} C^T z + d - BY^T \lambda &= 0, \\ B_{J_o} z - b_{J_0} &= 0, \\ B_{J\Lambda_0} z - b_{J\Lambda_0} &= 0, \end{aligned}$$

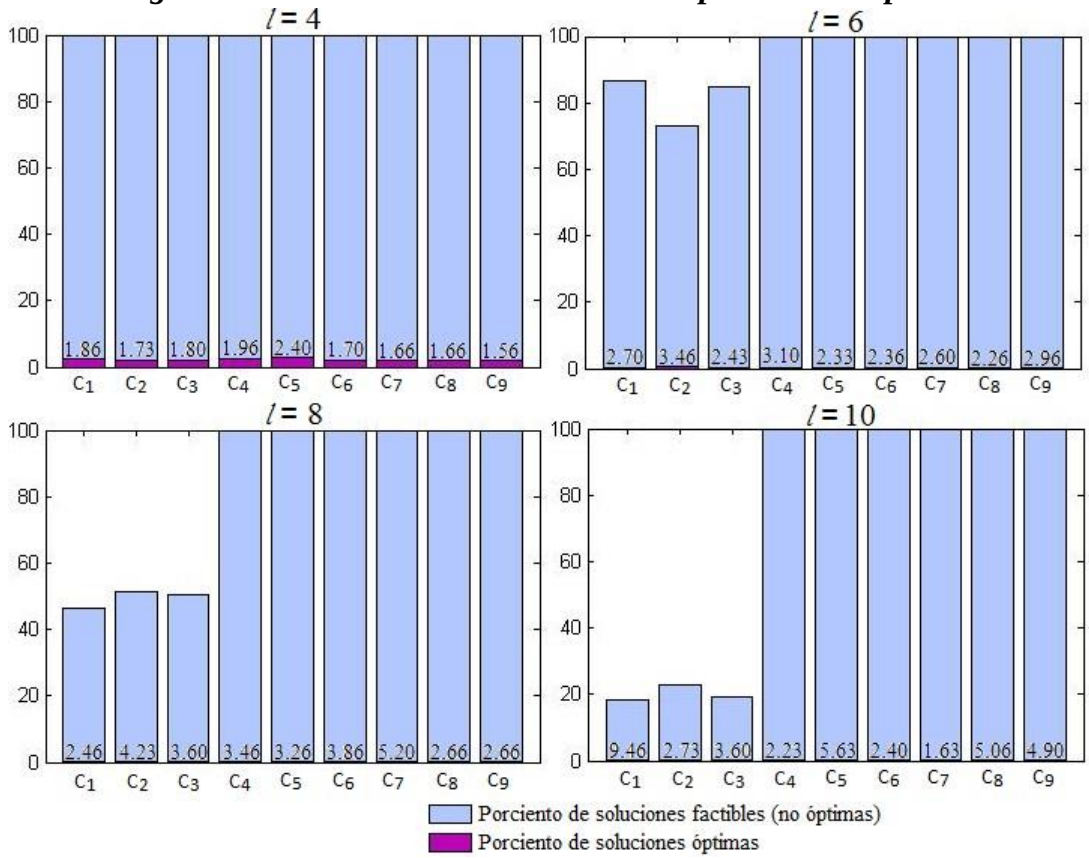
se calculan los restantes parámetros d y b.

El programa que permite generar problemas pruebas se implementó en Matlab. Para la ilustración numérica se generaron ejemplos de distintas dimensiones que difieren en cuanto a la cantidad de ternas con al menos una solución factible y cantidad de ternas con óptimos. Para los valores de n y m se consideraron tres clases de valores: pequeños, medianos y grandes. La clase Pequeño está definida en el intervalo [4, 6], la Mediana en el intervalo [10, 14] y la Grande en [17, 20]. Esto brinda 9 combinaciones posibles de índices n y m. Para cada combinación de n, m y l fijo, $l = 4, l = 6, l = 8, l = 10$ se crearon 30 problemas.

Para analizar el comportamiento de estos problemas se implementó un método exacto que examina de forma exhaustiva todas las posibles combinaciones de ternas (3^l) y devuelve la solución exacta del problema, el valor óptimo. Este método solo puede ser utilizado para valores pequeños de l, el aumento del costo computacional para valores de $l \geq 15$ hace intratable el problema para los recursos computacionales con los que se cuenta en la presente investigación. La utilización de este algoritmo permite ver cómo se comportan estos problemas en correspondencia con las dimensiones.

En la Figura 1, las barras de color azul representan el porciento de ternas factibles mientras que las barras de color violeta, el porciento de óptimos cuyo valor se hace explícito en cada

Figura 1: Análisis de las distintas clases de problemas de prueba.



caso. Se denota por c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9 las distintas clases (combinaciones) de n y m.

Para los problemas de dimensión l = 4, todas las ternas son factibles y el promedio de óptimos es semejante en todas las clases. Para los problemas con dimensión l = 6, l = 8, l = 10, solo las tres primeras clases presentan ternas infactibles. En el resto de las clases todas las ternas son factibles. El número promedio de óptimos para estas dimensiones está entre 1 y 3 al igual que en dimensión l = 4. Sin embargo, al aumentar el número de ternas, el porciento de óptimos con respecto al total de ternas es más bajo.

Entre el total de instancias generadas se seleccionaron 14 para conformar el conjunto de problemas de prueba. Se tomaron 4 instancias de dimensiones l = 6, l = 8 y l = 10 y 2 instancias de dimensión l = 4. Los problemas seleccionados presentan características distintas y significativas. Unos tienen como característica todas las ternas factibles, entre ellos se seleccionaron dos tipo, los que presentan pocos óptimos globales y los que presentan muchos óptimos (más de 5) globales. Los otros problemas pruebas seleccionados presentan ternas factibles e infactibles, al igual que los anteriores se diferencian en tener pocos o muchos óptimos globales. La Tabla 1 muestra las propiedades de cada uno de los problemas seleccionados. El código Matlab de los problemas de prueba y los algoritmos de búsqueda se puede descargar de (Tamayo-Vera et al., 2016).

Tabla 1: Característica de los problemas de prueba.

Instancias	Dimensión (l)	Ternas	Factibles	Óptimas	Valor
Inst.1	4	81	81	8	-4.26e+04
Inst.2	4	81	81	1	-7.55e+04
Inst.3	6	729	297	1	-2.13e+07
Inst.4	6	729	657	3	-1.85e+04
Inst.5	6	729	729	1	-1.25e+06
Inst.6	6	729	729	16	-2.56e+05
Inst.7	8	6561	6561	25	-1.03e+05
Inst.8	8	6561	6561	1	-4.73e+05
Inst.9	8	6561	5445	1	-2.66e+08
Inst.10	8	6561	4661	15	-1.32e+05
Inst.11	10	59049	24516	298	-3.78e+05
Inst.14	10	59049	3739	1	-2.19e+05
Inst.13	10	59049	59049	32	-3.13e+05
Inst.14	10	59049	59049	1	-3.44e+05

RESULTADOS DE LAS ESTRATEGIAS DE BÚSQUEDA

Para evaluar los resultados de las distintas estrategias de búsqueda se efectuaron 50 ejecuciones independientes por cada una de las instancias. Se tomó como condición de parada un número máximo de evaluaciones de la función objetivo, el cual se fijó como la parte entera del 10 % del total de ternas. Es decir, el número máximo de evaluaciones para $l = 4, l = 6, l = 8$ y $l = 10$es de 8, 72, 656 y 5904, respectivamente. Se fijó una cantidad de evaluaciones proporcional al número de ternas debido a las pequeñas dimensiones de los problemas de prueba propuestos.

Los resultados presentados en la Tabla 2 muestran el promedio del error relativo de cada estrategia al optimizar cada uno de los problemas. Se indican en negrita el mejor resultado obtenido para cada problema. Se define el rendimiento relativo de un algoritmo (a) respecto a un algoritmo (b) como: % — dif = (a — b)/ max(a, b), donde a y b indican el error absoluto de cada algoritmo. Valores positivos indican la medida (%) en que la estrategia (a) supera a la búsqueda aleatoria, valores negativos indican lo contrario. En la tabla se reporta el rendimiento relativo alcanzado por BL1, BL2 y RS (a) respecto a la búsqueda aleatoria (b).

Con el propósito de comparar correctamente el rendimiento de cada algoritmo respecto al algoritmo aleatorio se efectuó una prueba de hipótesis para determinar si la diferencia en los resultados es estadísticamente significativa. Se realizó un *t*-test con la hipótesis nula de que los errores reportados por cada algoritmo corresponden a distribuciones con igual media pero varianza desconocida (H_0: $\mu_a = \mu_b$), versus la hipótesis alternativa de que las medias no distintas (H_1: μ_a G μ_b). Para cada comparación se reporta el valor-*p* correspondiente a dicha prueba de hipótesis, valores menores que 0,05 se consideran estadísticamente significativos.

La Tabla 2 muestra que la Búsqueda Local con la vecindad de un intercambio brinda resultados similares a la Búsqueda Aleatoria. Aunque logra mejores resultados en 9 de los 14 problemas la diferencia en el rendimiento relativo es baja, siendo como promedio solo un 1.3% mejor y sin alcanzar una diferencia estadísticamente significativa en ninguno de los problemas. Un análisis más detallado del problema sugiere que este comportamiento se debe a que muchas soluciones vecinas tienen un mismo valor de la función objetivo, por ello con la vecindad de un intercambio existe un elevado número de óptimos locales y la búsqueda se estanca extremadamente rápido.

Es posible superar esta dificultad utilizando una vecindad mayor: de dos intercambios. Los resultados muestran que BL2 sí logra mejorar los resultados de la búsqueda aleatoria en todos los problemas. Esta mejora es estadísticamente significativa en 13 de las 14 instancias. La mejora relativa alcanzada supera el 35% en cada caso, para un promedio de 81.1% en todo el conjunto de prueba.

Sin embargo, los mejores resultados se logran con Recocido Simulado. Este algoritmo logra una mejora estadísticamente significativa respecto a Búsqueda Aleatoria en todos los problemas y alcanza una mejora relativa promedio de 92,4 % en todo el conjunto de prueba. Además, Recocido Simulado logra un menor error en 13 de los 14 problemas, mientras que la Búsqueda Local con la vecindad de dos intercambios lo hace para 2 de las instancias. En 4 de esos problemas RS halla el óptimo y para el problema 11 (que tiene 298 óptimos globales) tanto BL2 como RS encuentran la mejor terna.

CONCLUSIONES

Debido a la complejidad de los problemas lineales con restricciones de equilibrio y las aplicaciones que presentan, estos han sido motivo de estudio durante décadas. La presente investigación propone un nuevo enfoque para resolver estos modelos. La idea fundamental consiste en utilizar condiciones necesarias para describir la restricción de equilibrio, esto permite obtener un problema con restricciones de complementariedad. La novedad del

Tabla 2: Resultados numéricos de las distintas estrategias de búsqueda.

Inst.	**Aleatorio**	**Búsqueda Local 1**			**Búsqueda Local 2**			**Recocido Simulado**		
	Media	Media	%-dif	*t*-test	Media	%-dif	*t*-test	Media	%-dif	*t*-test
1	7.17e–01	7.42e–01	-3.4%	0.85	**7.89e–02**	89.0%	0.00	1.74e–01	75.7%	0.00
2	1.07e+00	1.00e+00	6.6%	0.12	6.47e–01	39.6%	0.00	**4.03e–01**	62.4%	0.00
3	9.94e–01	9.86e–01	0.8%	0.49	6.36e–01	36.0%	0.00	**3.51e–01**	64.7%	0.00
4	1.95e+00	2.69e+00	-27.7%	0.58	4.38e–03	99.8%	0.00	**8.99e–09**	99.9%	0.00
5	9.45e–01	9.64e–01	-2.0%	0.37	5.76e–01	39.0%	0.00	**6.30e–02**	93.3%	0.00
6	7.23e–01	6.53e–01	9.7%	0.41	4.47e–09	99.9%	0.00	**2.43e–11**	99.9%	0.00
7	4.62e–01	4.70e–01	-1.7%	0.87	2.99e–09	99.9%	0.00	**2.48e–13**	99.9%	0.00
8	9.57e–01	9.55e–01	0.2%	0.93	1.49e–01	84.4%	0.00	**8.25e–08**	99.9%	0.00
9	9.99e–01	9.96e–01	0.3%	0.21	5.24e–01	47.5%	0.00	**2.64e–02**	97.4%	0.00
10	1.15e+00	2.14e+00	-46.2%	0.50	1.26e–08	99.9%	0.00	**0.00e+00**	100.0%	0.00
11	1.04e+00	1.01e+00	3.5%	0.74	**0.00e+00**	100.0%	0.00	**0.00e+00**	100.0%	0.00
12	6.13e+03	1.68e+03	72.7%	0.49	1.76e–01	99.9%	0.32	**0.00e+00**	100.0%	0.02
13	8.84e–01	8.67e–01	1.9%	0.46	2.54e–09	99.9%	0.00	**0.00e+00**	100.0%	0.00
14	6.94e–01	6.73e–01	3.0%	0.60	3.98e–05	99.9%	0.00	**1.33e–10**	99.9%	0.00
1–14			**1.3%**			**81.1%**			**92.4%**	

enfoque propuesto es la utilización de algoritmos heurísticos para resolver el MPCC obtenido de la aplicación de las condiciones de KKT al problema lineal con restricciones de equilibrio.

Uno de los aportes de la investigación es la conformación de un conjunto de problemas de prueba que pueden ser utilizados en trabajos futuros para evaluar la efectividad de diversos algoritmos heurísticos. Para lograr esto, se diseñó e implementó un método de generación que utiliza las condiciones de KKT. Mediante este método se generó un amplio conjunto de problemas, de distintas dimensiones, cuyas características fueron analizadas mediante un algoritmo exhaustivo que permite evaluar todas las ternas y encontrar las óptimas. De estos problemas se seleccionaron aquellos con características más distintivas para conformar el conjunto de problemas de prueba propuesto.

Para optimizar los problemas pruebas se adaptaron e implementaron dos variantes de Búsqueda Local y Recocido Simulado. Los resultados alcanzados por estos algoritmos fueron comparados con un algoritmo de búsqueda aleatoria para verificar la efectividad de los mismos. Esta comparación evidenció que la utilización de algoritmos heurísticos es una propuesta viable para la solución de esta clase de problemas. En particular de metaheurísticas como RS que proporcionan estrategias de búsqueda para escapar de óptimos locales. La combinación de algoritmos heurísticos y metaheurísticos con herramientas de la programación matemática conforma un novedoso campo de investigación denominado matheurística (Boschetti et al., 2009), trabajos futuros continuarán enfocándose en esta interrelación.

Para trabajos futuros se recomienda implementar otras metaheurísticas, como por ejemplo Algoritmos Genéticos (Deb y Agrawal, 1999) o Algoritmos de Estimación de Distribución (Larrañaga y Lozano, 2002), y comparar los resultados con los ya obtenidos. Se recomienda además sustituir la representación ternaria por una representación binaria del problema y analizar los beneficios de esta última. Se trabajará además en continuar extendiendo el conjunto de problemas de prueba actual, en particular aumentando las dimensiones de los mismos.

REFERENCIAS

Aussel, D., Bendotti, P. y Pištěk, M. (2016). Nash Equilibrium in Pay-as-bid Electricity Market: Part 1- Existence and Characterisation. To appear in Optimization.

Boschetti, M.A., Maniezzo, V., Roffilli, M., y Bolufé-Röhler A.B. (2009). Matheuristics: Optimization, Simulation and Control. Hybrid Metaheuristics. Lecture Notes in Computer Science, 5818, 171–177.

Bouza, G. (2006). Mathemaical Programs with Equilibrium Constrainst: Solution Techniques from Parametric Optimization. Universiteit Twente.

Deb, K. y Agrawal, S. (1999) Understanding interactions among genetic algorithm parameters. Foundations of Genetic Algorithms, 265–286.

Ehrenmann, A. (2004). Equilibrium Problems with Equilibrium Constraints and their Application to Electricity Markets. Fitzwilliam College.

Fukushima, M. and Lin, G. H. (2003). New relaxation method for mathematical programs with complementarity constraints. Journal of Optimization Theory and Applications, 118(1):81–116.

Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. Journal of Statistical Physics, 34(5):975–986

Larrañaga P. y Lozano, J. A. (2002). Estimation of distribution algorithms: A new tool for evolutionary computation. Springer Verlag.

Luo Z. Q., Pang J. S. y Ralph D. (1996). Mathematical Programs with Equilibrium Constraints. Cambridge University Press.

Luo, Z. Q., Pang, J. S. y Ralph, D. (2006). Equilibrium constrained optimization problems. European Journal on Operational Research, 169(13):1108–1128.

Migdalas, A. (1995). Bilivel programming in traffic planning: Models, methods and challenge. Journal of Global Optimization, 7: 381-405.

Scholtes, S. and Scheel, H. (2000). Mathematical programs with complementarity constraints: Stationarity, optimality and sensitivity. Mathematics of Operations Research, 25(1):1–22.

Talbi, E (2009). Metaheuristics: from design to implementation. John Wiley & Sons.

Tamayo-Vera, D., Bouza G. y Bolufé-Röhler A. (2016). Benchmark for the LEC optimization problem. DOI: 10.13140/RG.2.1.1719.1288. (https://www.researchgate.net/publication/303289748_Benchmark_for_the_LEC_optimization_problem)

Ye, J. (1999). Optimality conditions for optimization problems with complementarity constraints. SIAM Journal on Optimization, 9(2):374–387.