Optimización Multiobjetivo con Técnicas de Inteligencia Artificial

Marcos Villagra
Facultad de Ciencias y Tecnología, Universidad Católica de Asunción
mdvillagra@gmail.com

1. Introducción

Gran parte de los problemas del mundo real implican la optimización simultánea de varios objetivos que generalmente presentan conflictos entre ellos; es decir, la mejora en uno conduce a un deterioro en el otro. La presencia de tales tipos de problemas es tan significativa, que consume gran parte de nuestro tiempo cotidiano de decisión. Se trata, por ejemplo, de escoger el medio ideal para llegar al trabajo, establecer el orden de nuestras tareas, elegir el restaurante para el almuerzo, hacer las compras en el supermercado, preparar la cena y la distribución de actividades en el tiempo de ocio restante. También es el mismo tipo de problemas que enfrentan los ingenieros y técnicos a la hora de diseñar e implementar sistemas de todo tipo: existen múltiples objetivos a cumplir y se espera lograrlos todos en la medida de lo posible.

Aunque la mayoría de los problemas de decisión involucran este tipo de situaciones, las propuestas computacionales de automatización que se han presentado para resolverlos habitualmente se limitan a convertir el problema de objetivos múltiples en uno en que existe un solo objetivo.

Esta reducción es debida a los modelos matemáticos empleados y puede realizarse de varias maneras, por ejemplo se prioriza uno de los objetivos y los demás se colocan como restricciones, o también se genera un objetivo compuesto otorgando pesos a los objetivos en juego y armando una suma ponderada de los mismos. De todos modos, ninguna de estas reducciones refleja fielmente al problema y, por tanto, tampoco otorga soluciones completamente satisfactorias.

Se pone como ejemplo el problema de la compra de un automóvil. El comprador desea un automóvil óptimo y por tanto no puede preocuparse solamente en minimizar el precio del auto, ya que también le interesan otros factores. Si se tratase de un problema de objetivo único, se conformaría con llamar a todos los distribuidores y hacer una lista de precios (sin considerar marcas, modelos, tamaño, confort, etc.) y escogería el automóvil de menor precio sin mayores complicaciones. Sin dudas, el modelo matemático que está empleando en su búsqueda ¡no es el más apropiado!. Por ende, los resultados tampoco son satisfactorios, y por ahorrarse unos pesos inicialmente, el comprador acaba gastando todos sus ahorros en combustible y costos de mantenimiento.

Sin embargo, el estado actual de la ciencia podría generar mejores resultados ya que existen modelos matemáticos que se ajustan mejor a la naturaleza de éstos problemas. Tales modelos provienen de un área de la Investigación de Operaciones conocida como optimización con objetivos múltiples o multiobjetivo.

En los problemas de optimización de un solo objetivo (*SOP*s, del inglés *Single Objective Problem*) el resultado óptimo deseado está claramente definido. Partiendo del ejemplo anterior el objetivo sería minimizar precio del automóvil, y el resultado sería el automóvil con menor precio. Sin embargo, esta condición no se cumple para los problemas de optimización multiobjetivo (*MOPs*, por sus siglas en inglés: *Multiobjective Optimization Problem*) donde, en vez de un único óptimo, contamos con todo un conjunto de soluciones de compromiso.

Para evidenciar este hecho se vuelve al ejemplo del problema de la compra de un automóvil. El comprador tiene varios objetivos que desearía alcanzar, pero también múltiples restricciones. En cuanto a objetivos podríamos mencionar: minimizar el costo del automóvil, la cantidad de combustible consumida en una distancia dada, los gastos de mantenimiento implicados, etc. Además, deseará maximizar el confort, el espacio, la confiabilidad, la seguridad, tiempo transcurrido entre mantenimientos, costos de reventa, etc. Cuando se plantean las restricciones descubrimos que este comprador cuenta con un presupuesto limitado, desea un vehículo fabricado en la región y confía más en algunas marcas que en otras. Los conflictos que surgen entre los objetivos mencionados son obvios e inmediatamente –considerando la propia experiencia– surgen posibles soluciones. Así tenemos el automóvil que tiene el menor precio, pero está bastante alejado de los objetivos relacionados al confort, la seguridad y la confiabilidad. También tenemos el de costo superior a los demás pero con óptimas características, aunque amplio consumo de combustible. Así podemos citar numerosos ejemplos. Entre éstos se encuentran los automóviles promedio, que cumplen con las restricciones dadas y que implican una solución de compromiso entre todos los factores en juego. Entre ellos no se puede decir que alguno sea mejor, ya que al alterar un factor para mejorarlo, estamos empeorando otro. Así, en un problema de optimización multiobjetivo cotidiano, resulta evidente la existencia de múltiples soluciones y la imposibilidad de decidir cuál de ellas es mejor si se consideran todos los objetivos al mismo tiempo. En el caso del comprador, para realizar su elección deberá necesariamente contar con algún criterio, posiblemente de índole subjetiva, que le permita optar por una u otra alternativa.

Se dice que las soluciones de un problema con objetivos múltiples son óptimas porque ninguna otra solución, en todo el espacio de búsqueda, es superior a ellas cuando se tienen en cuenta *todos* los objetivos al *mismo* tiempo, i.e. ningún objetivo puede mejorarse sin degradar a los demás.

Al conjunto de estas soluciones óptimas se conoce como soluciones Pareto óptimas. Su nombre les fue dado en honor al ingeniero y economista Wilfredo Pareto, quien fue el primero en definir un nuevo criterio de optimalidad para los problemas en que existen múltiples objetivos a cumplir, y persisten conflictos al realizar la optimización simultánea de los mismos. A partir de este concepto se establece, como requisito para afirmar que una situación es mejor que otra, el que en ella no se disminuya a nadie, pero se mejore a alguno; es decir que una situación será mejor que otra sólo si en la nueva es posible compensar las pérdidas de todos los perjudicados... y aún queda un sobrante. En todo otro caso, según Pareto, para decidir se requiere un juicio de valor y la ciencia no puede guiarnos.

Introducido el concepto de optimalidad Pareto, a continuación, en la sección 2 se presenta formalmente las definiciones básicas de la optimización multiobjetivo. En la sección 3 se explica el paradigma de la computación evolutiva, y en la sección 4 se muestra la técnica más utilizada para la resolución de MOPs, los algoritmos genéticos. En la sección 5 se discute como se relacionan los algoritmos evolutivos y los MOPs. Después, la sección 6 de una aplicación práctica de un algoritmo genético específico para la ubicación de centrales telefónicas en la ciudad de Asunción, y por último las conclusiones obtenidas durante la elaboración de este trabajo.

2. Conceptos básicos y terminología

Previamente a la introducción del problema a tratar, se presenta una descripción formal de conceptos y terminología, de modo a facilitar las discusiones posteriores. Cabe mencionar que en el área de optimización multiobjetivo, debido a la naturaleza aún incipiente del ámbito de investigación, no existe una notación estándar, y no ha sido sino hasta hace muy poco tiempo atrás que los investigadores han empezado a preocuparse de definir con claridad estos aspectos. Sin embargo, en gran parte de los trabajos consultados se percibe aún bastante confusión al respecto, por tanto es esencial establecer una notación clara antes de iniciar la discusión.

A partir de los conceptos introducidos en la sección anterior se define a un *MOP* de la siguiente manera:
Definición 1: Problema de Optimización Multiobjetivo (*Multiobjective Optimization Problem: MOP*). Un *MOP* general incluye un conjunto de *n* parámetros (variables de decisión), un conjunto de *k* funciones objetivo, y un conjunto de *m* restricciones. Las funciones objetivo y las restricciones son funciones de las variables de decisión. Luego, el *MOP* puede expresarse como:

$$\begin{array}{ll} \textit{Optimizar} & \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})) \\ \textit{sujeto a} & \boldsymbol{e}(\boldsymbol{x}) = (e_1(\boldsymbol{x}), e_2(\boldsymbol{x}), \ldots, e_m(\boldsymbol{x})) \geq \boldsymbol{0} \\ \textit{donde} & \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \boldsymbol{X} \\ & \boldsymbol{y} = (y_1, y_2, \ldots, y_k) \in \boldsymbol{Y} \end{array} \qquad \textbf{(1.1)}$$

siendo $\boldsymbol{x}$ el vector de decisión e $\boldsymbol{y}$ el vector objetivo. El espacio de decisión se denota por $\boldsymbol{X}$, y al espacio objetivo por $\boldsymbol{Y}$. Optimizar, dependiendo del problema, puede significar igualmente, minimizar o maximizar.

El conjunto de restricciones $\boldsymbol{e}(\boldsymbol{x}) \geq \boldsymbol{0}$ determina el conjunto de soluciones factibles $\boldsymbol{X}_f$ y su correspondiente conjunto de vectores objetivo factibles $\boldsymbol{Y}_f$.
Definición 2: Conjunto de soluciones factibles. El conjunto de soluciones factibles $\boldsymbol{X}_f$ se define como el conjunto de vectores de decisión $\boldsymbol{x}$ que satisface los requerimientos $\boldsymbol{e}(\boldsymbol{x})$:

$$\boldsymbol{X}_f = \{\boldsymbol{x} \in \boldsymbol{X} \mid \boldsymbol{e}(\boldsymbol{x}) \geq \boldsymbol{0}\} \qquad \textbf{(1.2)}$$

La imagen de $\boldsymbol{X}_f$, es decir, la región factible del espacio objetivo, se denota por

$$\boldsymbol{Y}_f = \boldsymbol{f}(\boldsymbol{X}_f) = \bigcup_{x \in X_f} \{\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})\} \qquad \textbf{(1.3)}$$

De estas definiciones se tiene que cada solución del *MOP* en cuestión consiste de una *n*-tupla $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, que conduce a un vector objetivo $\boldsymbol{y} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$, donde cada $\boldsymbol{x}$ debe cumplir con el conjunto de restricciones $\boldsymbol{e}(\boldsymbol{x}) \geq \boldsymbol{0}$. El problema de optimización consiste en hallar la $\boldsymbol{x}$ que tenga el "mejor valor" de $\boldsymbol{f}(\boldsymbol{x})$. En general, y según ya se ha introducido, no existe un único "mejor valor", sino un conjunto de soluciones. Entre éstas, ninguna se puede considerar mejor a las demás si se tienen en cuenta todos los objetivos al mismo tiempo. Este hecho deriva de que puede existir –y generalmente existe– conflicto entre los diferentes objetivos que componen el problema. Por ende, al tratar con *MOPs* se precisa de un nuevo concepto de "óptimo".

En la optimización de un solo objetivo el conjunto de variables de decisión factibles está completamente ordenado mediante una función objetivo *f*. Es decir, dadas dos soluciones $\boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{X}_f$, se cumple una sola de las siguientes proposiciones: $f(\boldsymbol{a}) > f(\boldsymbol{b})$, $f(\boldsymbol{a}) = f(\boldsymbol{b})$ o $f(\boldsymbol{b}) > f(\boldsymbol{a})$. El objetivo consiste en hallar la solución (o soluciones) que tengan los valores óptimos (máximos o mínimos) de *f*. Cuando se trata de varios objetivos, sin embargo, la situación cambia. $\boldsymbol{X}_f$, en general, no está totalmente ordenada por los objetivos; el orden que se da suele ser parcial (i.e. existen vectores de decisión $\boldsymbol{a}$ y $\boldsymbol{b}$ con los que $\boldsymbol{f}(\boldsymbol{a})$ no puede considerarse mejor que $\boldsymbol{f}(\boldsymbol{b})$ y tampoco $\boldsymbol{f}(\boldsymbol{b})$ puede considerarse mejor que $\boldsymbol{f}(\boldsymbol{a})$).

Para ilustrar este concepto se presenta el siguiente gráfico que muestra la relación entre dos funciones f_1 y f_2. La función f_1 representa el costo de los dispositivos de seguridad de un sistema, mientras que f_2 representa el índice de ocurrencia de accidentes.

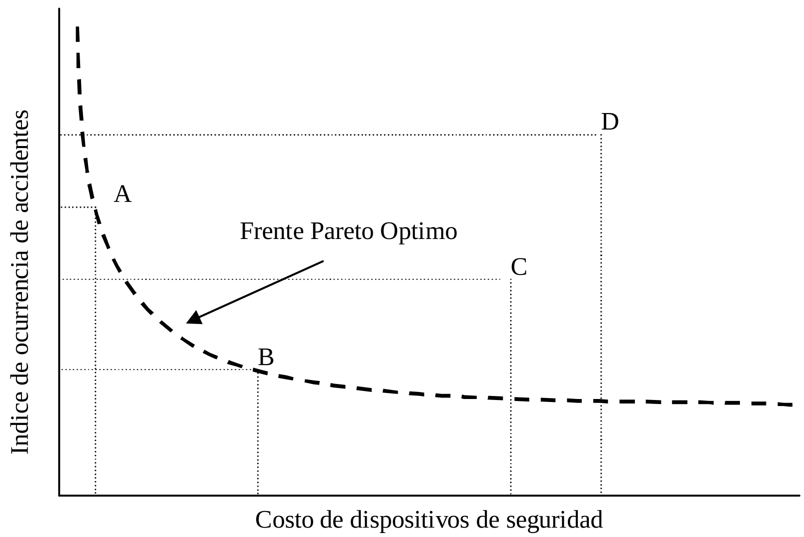

Figura 1. Ejemplo de la optimalidad Pareto en el espacio objetivo.

La solución representada por el punto B es mejor que la representada por el punto C, debido a que provee de una mayor seguridad a un costo menor. La solución B sería también la escogida en el caso de estar realizando la optimización de un solo objetivo. Más aún, todas las soluciones que se encuentran en el rectángulo delimitado por el origen de coordenadas y la solución C y por encima de la curva, son mejores a C. En la comparación entre C y A, se obtiene que, disminuyendo mucho los costos, el nivel de seguridad provisto por A es solo ligeramente peor, aunque C y A son no comparables entre ellos porque no se podría argumentar que uno es mejor que otro al considerar todos los objetivos. Si comparamos a A con B tampoco podríamos establecer que alguna de las dos sea mejor, si se considera que ambos objetivos son igualmente importantes y no se introduce alguna consideración de índole subjetiva. Sin embargo, B es claramente superior a C en ambos objetivos. Para expresar esta situación matemáticamente, las relaciones =, $\leq$ y $\geq$ se deben extender. Esto se puede realizar de la siguiente manera:

Definición 3: Dados 2 vectores de decisión $\boldsymbol{u} \in \boldsymbol{X}$ y $\boldsymbol{v} \in \boldsymbol{X}$,

$$\begin{aligned} &\boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{v}) \text{ si y sólo si } \forall i \in \{1, 2, \dots, k\}: f_i(\boldsymbol{u}) = f_i(\boldsymbol{v}) \\ &\boldsymbol{f}(\boldsymbol{u}) \geq \boldsymbol{f}(\boldsymbol{v}) \text{ si y sólo si } \forall i \in \{1, 2, \dots, k\}: f_i(\boldsymbol{u}) \geq f_i(\boldsymbol{v}) \\ &\boldsymbol{f}(\boldsymbol{u}) > \boldsymbol{f}(\boldsymbol{v}) \text{ si y sólo si } \boldsymbol{f}(\boldsymbol{u}) \geq \boldsymbol{f}(\boldsymbol{v}) \wedge \boldsymbol{f}(\boldsymbol{u}) \neq \boldsymbol{f}(\boldsymbol{v}) \end{aligned} \quad \textbf{(1.4)}$$

Las relaciones $\leq$ y < se definen de manera similar.

A partir de esta noción, se sigue que $\boldsymbol{f}(B) < \boldsymbol{f}(C)$, $\boldsymbol{f}(C) < \boldsymbol{f}(D)$, y como consecuencia $\boldsymbol{f}(B) < \boldsymbol{f}(D)$. Sin embargo, al comparar *A* y *C* o *A* y *B*, no se puede decir que alguna sea superior a la otra. Por ejemplo, a pesar de que la solución representada por *B* es más cara, provee menor índice de accidentes que la representada por *A*.

Por lo expuesto, se tiene que dos vectores de decisión $\boldsymbol{x_1}$ y $\boldsymbol{x_2}$ de un *MOP* pueden cumplir solo una de tres condiciones posibles: $\boldsymbol{f}(\boldsymbol{x_1}) > \boldsymbol{f}(\boldsymbol{x_2})$, $\boldsymbol{f}(\boldsymbol{x_2}) > \boldsymbol{f}(\boldsymbol{x_1})$ o $\boldsymbol{f}(\boldsymbol{x_1}) \ngeq \boldsymbol{f}(\boldsymbol{x_2}) \wedge \boldsymbol{f}(\boldsymbol{x_2}) \ngeq \boldsymbol{f}(\boldsymbol{x_1})$. Esta situación se expresa con los siguientes símbolos y términos:

Definición 4: Dominancia Pareto en un contexto de Maximización. Para dos vectores objetivo $\boldsymbol{a}$ y $\boldsymbol{b}$,

$a \succ b$ (*a* **domina a** *b*)	**si y solo si**	$a > b$	**(1.5)**
$\boldsymbol{b} \succ \boldsymbol{a}$ ($\boldsymbol{b}$ domina a $\boldsymbol{a}$)	si y solo si	$\boldsymbol{b} > \boldsymbol{a}$	
$\boldsymbol{a} \sim \boldsymbol{b}$ ($\boldsymbol{a}$ y $\boldsymbol{b}$ no son comparables)	si y solo si	$\boldsymbol{a} \ngeq \boldsymbol{b} \wedge \boldsymbol{b} \ngeq \boldsymbol{a}$	

Definición 5: Dominancia Pareto en un contexto de Minimización. Para dos vectores objetivo $\boldsymbol{a}$ y $\boldsymbol{b}$,

$$\boldsymbol{a} \succ \boldsymbol{b} \text{ (} \boldsymbol{a} \text{ domina a } \boldsymbol{b}\text{)} \quad \text{si y solo si} \quad \boldsymbol{a} < \boldsymbol{b} \tag{1.6}$$

$$\boldsymbol{b} \succ \boldsymbol{a} \text{ (} \boldsymbol{b} \text{ domina a } \boldsymbol{a}\text{)} \quad \text{si y solo si} \quad \boldsymbol{b} < \boldsymbol{a}$$

$$\boldsymbol{a} \sim \boldsymbol{b} \text{ (} \boldsymbol{a} \text{ y } \boldsymbol{b} \text{ no son comparables)} \quad \text{si y solo si} \quad \boldsymbol{a} \nleq \boldsymbol{b} \wedge \boldsymbol{b} \nleq \boldsymbol{a}$$

Luego, de aquí en adelante, ya no será necesario diferenciar el tipo de optimización a realizar (minimización o maximización), al punto que un objetivo puede ser maximizado, mientras que otro puede ser minimizado.
Se puede introducir el criterio de optimalidad Pareto, a partir del concepto de dominancia Pareto. Volviendo al gráfico, el punto A es único entre los demás: su vector de decisión $\boldsymbol{a}$ no está dominado por otro vector de decisión. Esto implica que $\boldsymbol{a}$ es óptimo en el sentido de que no puede mejorarse en un objetivo, sin degradar a otro. Tales soluciones se conocen como Pareto óptimas (también se utiliza el término “no inferior”).

Definición 6: Optimalidad Pareto. Dado un vector de decisión $\boldsymbol{x} \in \boldsymbol{X}_f$ y su correspondiente vector objetivo $\boldsymbol{y} = \boldsymbol{f}(x) \in \boldsymbol{Y}_f$, se dice que $\boldsymbol{x}$ es no dominado respecto a un conjunto $\boldsymbol{A} \subseteq \boldsymbol{X}_f$ si y solo si

$$\forall \ \boldsymbol{a} \in \boldsymbol{A} : (\boldsymbol{x} \succ \boldsymbol{a} \vee \boldsymbol{x} \sim \boldsymbol{a}) \tag{1.7}$$

En caso que $\boldsymbol{x}$ sea no dominado respecto a todo el conjunto $\boldsymbol{X}_f$, y solo en ese caso, se dice que $\boldsymbol{x}$ es una **solución Pareto óptima** ($\boldsymbol{x} \in \boldsymbol{X}_{true}$ –el conjunto Pareto óptimo real–). Mientras que la $\boldsymbol{y}$ correspondiente es parte del **frente Pareto óptimo real $\boldsymbol{Y}_{true}$**. Esto se define a continuación.
Definición 7: Conjunto Pareto óptimo y frente Pareto óptimo. Dado el conjunto de vectores de decisión factibles $\boldsymbol{X}_f$. Se denomina $\boldsymbol{X}_{true}$ al conjunto de vectores de decisión no dominados que pertenecen a $\boldsymbol{X}_f$, es decir:

$$\boldsymbol{X}_{true} = \{ \boldsymbol{x} \in \boldsymbol{X}_f \mid \boldsymbol{x} \text{ es no dominado con respecto a } \boldsymbol{X}_f \} \tag{1.8}$$

El conjunto $\boldsymbol{X}_{true}$ también es conocido como el conjunto Pareto óptimo. Mientras que el conjunto correspondiente de vectores objetivo $\boldsymbol{Y}_{true} = \boldsymbol{f}(\boldsymbol{X}_{true})$ constituye el frente Pareto óptimo.

Retomando el ejemplo de la figura 1, los puntos de la curva (i.e. A, B) representan soluciones Pareto óptimas. Entre ellas son no comparables, o, en otras palabras, son indiferentes. Esto aclara aún más la diferencia de los *MOPs* con los *SOPs*, o problema de optimización de un único objetivo): en este caso no hay una única solución sino un conjunto de soluciones de compromiso. Ninguna de ellas se puede definir como “mejor” que las demás, a menos que se incluya alguna otra información que determine que un objetivo es más importante que los demás o que se fije un peso relativo entre los objetivos.

Como recapitulación y como una ilustración adicional de los conceptos presentados, se discutirá a continuación un *MOP* específico, con una sola variable de decisión (n = 1), dos funciones objetivo (k = 2) y sin restricciones (m = 0). Este *MOP* se define como:

$$\text{Minimizar:} \quad \boldsymbol{f}(x) = (f_1(x), f_2(x)), \tag{1.9}$$

$$\text{donde} \quad f_1(x) = x^2,$$

$$f_2(x) = (x - 2)^2$$

Este par de funciones ha sido estudiado previamente en muchos otros trabajos, constituyéndose en un paradigma para la presentación de los principales conceptos de un *MOP*. En efecto, según estudios bibliográficos la optimización de este par es la función objetivo clásica escogida por gran parte de los autores que proponen un algoritmo evolutivo multiobjetivo y que desean probar su eficiencia. Por este motivo ha sido seleccionado para su estudio preliminar.

La figura 2 presenta la gráfica de ambas funciones respecto a x. La misma sugiere que el conjunto Pareto óptimo es $\{ x \mid x \geq 0 \text{ y } x \leq 2 \}$. Esto es: $\boldsymbol{X}_{true} = \{x \in R \mid 0 \leq x \leq 2\}$. La solución $x = 0$ es óptima respecto a f_1 pero no respecto a f_2. Cualquier solución que no se encuentre en el rango [0, 2] no es miembro del conjunto Pareto óptimo, puesto en dicho rango ($\boldsymbol{X}_{true}$) existe siempre una solución que la domina.

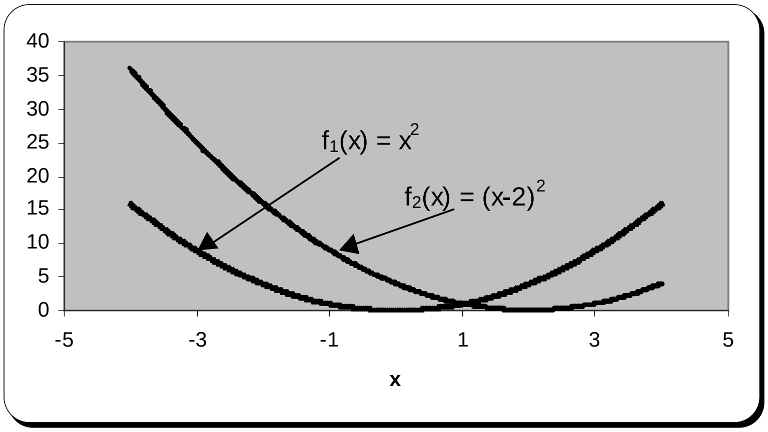

Figura 2. Valores de f_1 *y* f_2 en función de x

La diferencia esencial entre la figura 2 y la figura 3 es que la primera representa los valores de las funciones f_1 y f_2 para diferentes valores de la variable independiente x. Mientras que la segunda representa los valores de la función f_1 graficados contra la función f_2, para el mismo valor de la variable independiente. Es decir, la figura 3 es una gráfica en el espacio objetivo $\boldsymbol{Y}$, que muestra los vectores del *MOP* como puntos. Los vectores no dominados (que se muestran como puntos), representan el frente Pareto del *MOP*, por encima de la curva, se encuentra el espacio objetivo factible $\boldsymbol{Y}_f$.

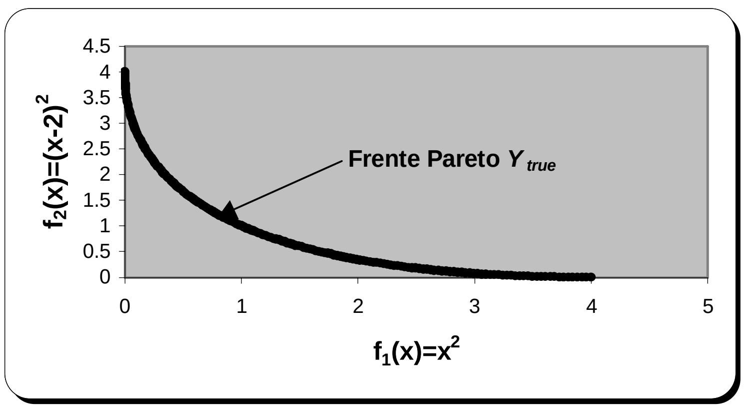

Figura 3. Frente Pareto del *MOP*.

3. Algoritmos Evolutivos

El término algoritmo evolutivo (*EA*. por sus siglas en inglés: *Evolutionary Algorithm*) se refiere a técnicas de búsqueda y optimización inspiradas en el modelo de la evolución propuesto por Charles Darwin, luego de sus viajes exploratorios.

En la naturaleza los individuos se caracterizan mediante cadenas de material genético que se denominan cromosomas. En los cromosomas se halla codificada toda la información relativa a un individuo y a sus tendencias. Cada elemento que conforma el cromosoma recibe el nombre de allele. Cada individuo posee un nivel del adaptación al medio que lo dota de mayor capacidad de supervivencia y generación de descendencia. Tal nivel de adaptación está ligado a las características que están codificadas en sus cromosomas. Como el material genético puede transmitirse de padres a hijos al ocurrir el apareamiento, los hijos resultantes poseen cadenas de cromosomas parecidas a las de sus padres y combinan las características de los mismos. Por tanto si padres con buenas características se cruzan, posiblemente generarán hijos igualmente buenos o incluso mejores.

Para resolver un problema de búsqueda u optimización utilizando algoritmos evolutivos y los conceptos sugeridos, primero se representa como individuos de una población finita a un número dado de soluciones posibles del problema, a este proceso se denomina codificación. En la codificación de un individuo debe estar presente toda la información relevante al mismo y que se considera influye en la optimización o búsqueda. Generalmente la codificación de un individuo o su cromosoma es una cadena de bits o de números enteros, dependiendo del problema que se desea resolver. En dicha cadena, el elemento que se encuentra en una posición dada recibe el nombre de allele.

Luego, se determina el nivel de aptitud o adaptación de cada individuo (*fitness*), dependiendo de la calidad de la solución que representa. Posteriormente los individuos existentes generan a otros individuos mediante los operadores genéticos como selección, cruzamiento y mutación. El operador de selección elige los padres que se cruzarán. La probabilidad de que un individuo sea escogido como padre y/o que sobreviva hasta la siguiente generación está ligada a su *fitness* o aptitud: a mayor *fitness*, mayor probabilidad de supervivencia y de tener descendientes, de la misma forma que ocurre en los procesos naturales.

Luego de escogerse los padres, se procede a la recombinación o cruzamiento de los mismos para obtener a la nueva generación. De esta manera, en cada nueva generación se tienen buenas probabilidades de que la población se componga de mejores individuos, ya que los hijos heredarán las características buenas de sus padres, y al combinarlas podrán ser aún mejores.

Por otro lado, durante la recombinación pueden ocurrir alteraciones (mutaciones) en la información genética de un individuo. Si tales alteraciones se producen para bien, originarán un individuo bueno con alto *fitness* y la alteración se transmitirá a los nuevos individuos; si el cambio no es benéfico, el individuo alterado tendrá un *fitness* bajo y poca o ninguna descendencia, con lo que la alteración prácticamente morirá con él. De esta manera, luego del curso de varias generaciones, la población habrá evolucionado hacia individuos genéticamente muy parecidos y que tienen un nivel de aptitud elevado, es decir, representan buenas soluciones al problema propuesto.

Los operadores descritos reciben el nombre de operadores de búsqueda u operadores genéticos. La reproducción enfoca la atención en los individuos con alto *fitness*, y de esta manera **explota** la información disponible sobre la adaptación del individuo al medio ambiente. La recombinación y la mutación perturban de alguna manera a los individuos y proveen así de heurísticas para la **exploración** del espacio de búsqueda. Por ello se dice que los *EA*s utilizan conceptos de explotación y exploración. A pesar de ser simplistas desde el punto de vista de la biología, estos algoritmos son suficientemente complejos como para proveer mecanismos de búsqueda robustos y que se adaptan a gran variedad de problemas.

Los orígenes de estos algoritmos se remontan a la década de los 50, cuando se empezaron a estudiar los primeros conceptos, pero los mismos no se formalizan sino hasta el trabajo de Holland "*Adaptation in natural systems*".

Aproximadamente desde mediados de la década del 70, en el ámbito de los *EAs*, se han introducido numerosos algoritmos que se pueden clasificar principalmente como Algoritmos Genéticos (*GA*), Programación Evolutiva (*EP*) y Estrategias de Evolución (*ES*), Sistemas clasificadores (*CS*) y Programación Genética (*GP*). A pesar de que el principio que los soporta es, en apariencia, muy simple, estos algoritmos han probado ser herramientas generales, robustas y potentes. En la siguiente sub-sección se brinda una pequeña introducción a los algoritmos genéticos.

3.1. Algoritmos Genéticos

Los algoritmos genéticos se utilizan en varios ámbitos, principalmente para búsquedas y optimizaciones. En la práctica se implementa el algoritmo escogiendo una codificación para las posibles soluciones del problema. La codificación se realiza mediante cadenas de bits, números o caracteres para representar a los cromosomas. Luego, las operaciones de cruzamiento y mutación se aplican de manera muy sencilla mediante funciones de manipulación de valores de vectores. A pesar de que se han realizado numerosas investigaciones sobre codificación usando cadenas de longitud variable y aún otras estructuras, gran parte del trabajo con algoritmos genéticos se enfoca en cadenas de bits de longitud fija. Justamente el hecho de utilizar cadenas de longitud fija y la necesidad que existe de codificar las posibles soluciones son las dos características cruciales que diferencian a los algoritmos genéticos de la programación genética, que no posee representación de longitud fija y que normalmente no utiliza una codificación del problema y sus soluciones.

El ciclo de trabajo de un *GA* (también conocido como ciclo generacional) es generalmente el siguiente: calcular el *fitness* de los individuos en la población; crear una nueva población mediante selección, cruzamiento y mutación; y, finalmente, descartar la población vieja y seguir iterando utilizando la población recién creada. A cada iteración de este ciclo se la conoce como una generación. Cabe observar que no hay una razón teórica para que este sea el modelo de implementación. De hecho este comportamiento puntual no se observa como tal en la naturaleza. Sin embargo el modelo sigue siendo válido y conveniente.

La primera generación (generación 0) de este proceso, opera sobre una población de individuos generados al azar. A partir de allí, los operadores genéticos se aplican para mejorar a la población. El pseudocódigo 1, presentado a continuación, muestra el algoritmo principal.

```
Procedimiento GA( )
Inicio
        t = 0 /* empezar con un tiempo inicial */
        InicializarPoblacion P (t) /* inicializar una población de individuos generados al azar */
        Evaluate P (t) /* evaluar el fitness de todos los individuos de la población inicial */
        Mientras no se cumpla el criterio de parada
                t = t + 1 /* incrementar el contador de tiempo */
                P’ = SeleccionarPadres P (t) /* seleccionar una poblacion para generar descendientes */
                Recombinar P’ (t) /* recombinar los “genes” del grupo de padres seleccionados */
                Mutar P’ (t) /* perturbar la población generada de manera estocástica */
                Evaluar P’ (t) /* calcular fitness de la población recién creada */
                P = Sobrevivientes P, P’ (t)/* Seleccionar sobrevivientes para la siguiente generación */
        Fin Mientras
Fin
```

Pseudocódigo 1. Algoritmo genético básico

El operador de selección, según se ha apuntado, simula el proceso de selección natural en que el más fuerte tiene mayor capacidad de supervivencia. En el *GA* la capacidad de supervivencia de un individuo está ligada al valor numérico de la función objetivo o *fitness*. Este operador se aplica a cada iteración sobre una población de individuos de tamaño constante, con el objetivo de seleccionar individuos prometedores para generar la nueva población. Entre los individuos seleccionados pueden hallarse dos o más individuos idénticos, esto se debe a que los individuos con bajo *fitness* tienen poca probabilidad de ser elegidos, mientras que los de buen fitness son seleccionados con mayor frecuencia.

El operador de selección puede aplicarse de maneras diversas. Unas veces se realiza la selección mediante torneos en que se escoge al azar un grupo de individuos y gana el torneo aquel con mejor *fitness*. La cantidad de individuos que se escogen para la competencia se fija de antemano y permanece constante en la implementación tradicional. Esta forma de selección recibe el nombre de selección por torneos y refleja de manera más adecuada el proceso natural de selección. Otra forma, conocida como selección de ruleta, es fácilmente comprensible imaginando una ruleta en la que el número de partes en que se divide la misma es igual a la cantidad de individuos de la población, siendo el tamaño de cada parte proporcional al *fitness* de cada individuo. Es de esperar que al hacer girar la ruleta varias veces, se obtendrá mayor cantidad de individuos con alto *fitness*. Aunque pueden existir otras formas de realizar la selección, las dos anteriormente mencionadas son las más comunes en las implementaciones publicadas.

Una vez seleccionados los individuos se aplica a cada par de ellos el operador de cruzamiento. También el cruzamiento se puede realizar de maneras diferentes y aquí solo se discutirá una de ellas: el cruzamiento de 1 punto. En el mismo, se escoge aleatoriamente un punto de corte que se aplicará al par de cromosomas o codificación de los individuos seleccionados. Luego, los caracteres más significativos, a partir del punto de corte se conservan en sus posiciones relativas en el nuevo par de cromosomas, y los restantes son intercambiados de los cromosomas progenitores a los nuevos dos cromosomas obtenidos. Como ejemplo, considérese dos cromosomas A1 y A2 de un par de individuos, sobre los que se aplicará el operador de cruzamiento para obtener dos nuevos cromosomas. Sean los individuos:

A1 = 0 1 1 | 0 1
A2 = 1 1 0 | 0 0

donde el símbolo "|" indica el punto de corte. Los caracteres a la izquierda del punto de corte son los más significativos, por tanto permanecen en sus posiciones correspondientes en los nuevos cromosomas. Mientras que los caracteres a la derecha del punto de corte son cruzados de los cromosomas progenitores, como muestra el ejemplo:

A'1 = 0 1 1 | 0 0
A'2 = 1 1 0 | 0 1

De esta manera, A'1 y A'2 son los cromosomas resultantes de aplicar el operador de cruzamiento al par de cromosomas progenitores.

El operador de cruzamiento representa una forma de búsqueda local, en las inmediaciones del espacio de búsqueda que rodea a los padres. Por su parte, el proceso de mutación es básicamente una búsqueda aleatoria. Se selecciona aleatoriamente una posición específica dentro del cromosoma del individuo a mutar, para luego cambiar el valor contenido en dicha posición.

Como en la naturaleza, la probabilidad de que ocurra una mutación es pequeña, en las condiciones de vida normales, en el *GA* se trata de representar lo mismo, con un valor de ocurrencia muy bajo para el operador de mutación.

4. Algoritmos Evolutivos y Optimización Multiobjetivo

Los *EAs* resultan interesantes ya que, a primera vista, parecen estar especialmente dotados para lidiar con las dificultades propuestas por los *MOPs*. Esto se debe a que pueden devolver un conjunto entero de soluciones luego de una corrida simple y no presentan otras limitaciones propias de las técnicas tradicionales. Incluso, algunos investigadores han sugerido que los *EAs* se comportarían mejor que otras técnicas de búsqueda ciega. Esta afirmación todavía requiere de una demostración fehaciente y debería considerarse a la luz de los teoremas de "no hay almuerzo gratis" ("*no free lunch*") relacionados con la optimización, y que indican que si un método se comporta "mejor" que otros en un conjunto de problemas, tendrá un desempeño "peor" en otro conjunto de problemas. De todos modos, la realidad es que hoy día existen pocas alternativas válidas en cuanto a otros posibles métodos de solución. De hecho, las numerosas publicaciones recientes sobre resolución de *MOPs* usando *EAs*, parecen considerar este hecho, y han dado lugar al nacimiento de todo un nuevo campo de investigación: los algoritmos evolutivos aplicados a optimización multiobjetivo (*MOEA* por sus siglas en inglés*: Multiobjective Optimization Evolutionary Algorithm*).

Ya en 1967, Rosemberg sugirió, sin hacer simulaciones, un método de búsqueda genética para aplicaciones químicas con diversos objetivos; sin embargo, las primeras implementaciones prácticas fueron sugeridas recién en 1984. Posterior a esto no se realizaron estudios significativos por casi una década, a excepción de un procedimiento de ordenación basado en no-dominancia, expuesto por Goldberg. Nuevas implementaciones de *MOEAs* surgieron entre 1991 y 1994. Posteriormente, estas sugerencias –y variaciones de las mismas– se implementaron para la resolución de diferentes *MOPs*. Más recientemente, algunos investigadores se han puesto la tarea de tratar tópicos específicos de la búsqueda y optimización multiobjetivo con algoritmos evolutivos, como son: convergencia al frente Pareto, técnicas de niching, aplicación de elitismo, etc. Otros se han dedicado a sugerir nuevos algoritmos y técnicas evolutivas.

A pesar de esta gran variedad, aún existen numerosas cuestiones que todavía permanecen abiertas. Entre ellas podemos citar:

a) ¿Qué algoritmos propuestos se desempeñan mejor en qué tipo de problemas?
b) ¿Cuáles son las ventajas y desventajas de un algoritmo particular, respecto a los demás?
c) ¿Qué medida de calidad se utilizará para afirmar que un algoritmo es efectivo en un caso dado?
d) ¿Las técnicas como niching, elitismo y restricciones de cruzamiento pueden mejorar el desempeño de los *MOEAs*?
e) ¿Cuáles son las técnicas más apropiadas para la paralelización de *MOEAs*?
f) ¿Cómo afecta el proceso de paralelización al desempeño de los algoritmos?

Gran parte de las dudas surgen del mismo hecho que los *MOEAs* se aplican a una cantidad enorme de problemas de diversos ámbitos, por lo que una generalización completa parece ser imposible.

Además, es importante evidenciar que algunas aplicaciones del mundo real presentan mayor dificultad para encontrar soluciones no inferiores (Pareto óptimas), con lo que la determinación del frente Pareto completo no es factible. Para estos casos, es necesario redefinir los objetivos de la resolución de *MOPs* considerando lo siguiente:

a) Se debe minimizar la distancia entre el frente Pareto verdadero y el frente conformado por las soluciones (no dominadas) halladas.
b) Se debe tener una buena distribución (generalmente esto equivale a decir una distribución uniforme) de las soluciones identificadas.

c) Es deseable encontrar soluciones bien diferenciadas unas de otras.

6. Caso de Estudio: Optimización Multiobjetivo en la Planificación de Centrales Telefónicas

El vertiginoso crecimiento del consumo y variedad de los servicios de telecomunicaciones genera una necesidad cada vez mayor de implementar herramientas eficientes para la planificación de las redes de telecomunicaciones, a fin de minimizar los altos costos de inversión y mantenimiento. Básicamente, el problema a resolver consiste en calcular la cantidad de centrales necesarias para cubrir la demanda de un área y la correspondiente ubicación eficiente de las mismas, de forma a minimizar los costos basados en los datos de población, demanda de tráfico y costo de la infraestructura requerida para atender la demanda proyectada.

Actualmente, existen herramientas de planificación como PLANITU que permiten realizar la planificación de centrales, calculando una ubicación de centrales que atiende a necesidades de telecomunicaciones. Esta herramienta convencional, resuelve el problema en cuestión proponiendo una única solución, calculada mediante métodos tradicionales basados en el álgebra lineal. Este método es adecuado cuando se estudia la posibilidad de instalar una nueva central, pero no es eficiente cuando se esperan ubicar varias centrales, ya que se necesitaría muchísimo tiempo de procesamiento para analizar cada una de las posibles combinaciones, con el agravante de obtener resultados que no garantizan ser una solución óptima. Adicionalmente, herramientas existentes de planificación como PLANITU, tienen la restricción adicional de un costo muy elevado de adquisición y mantenimiento, lo que complica su utilización en instituciones sin suficientes recursos económicos.

Se propone resolver el problema de planificación de centrales de telecomunicaciones considerando simultáneamente:

- la demanda actual (año 2002 para el problema de prueba),
- la demanda a mediano plazo (año 2004 en el referido problema de prueba) y
- la demanda a largo plazo (considerando el año 2007 para este trabajo, por falta de estimaciones suficientes para años posteriores).

Debido a la imposibilidad de los métodos tradicionales de realizar la optimización simultanea de varios objetivos en la búsqueda de soluciones, el presente trabajo propone utilizar Algoritmos Evolutivos Multiobjetivos que permitan encontrar soluciones al problema de referencia, optimizando todos los objetivos propuestos, al mismo tiempo. Se propone la optimización de las redes de telecomunicaciones utilizando un Algoritmo Evolutivo Multiobjetivo. En particular, se utilizará el *Strength Pareto Evolutionary Algorithm* - SPEA 2, por su reconocida eficiencia en la búsqueda de soluciones multiobjetivo.

6.1. Ubicación óptima de Centrales

El problema de la ubicación óptima de centrales consiste en encontrar el número óptimo de centrales telefónicas, y la mejor ubicación de dichas centrales en un área de estudios (típicamente una ciudad determinada, Asunción para este trabajo), de forma a minimizar el costo acumulado de inversión a corto, mediano y largo plazo.

El área de la ciudad a ser atendida se divide en m cuadrículas de por ejemplo 10 a 500 m de lado. A cada una de éstas cuadrículas se le asigna un valor de fila y columna, conformando una matriz. A cada elemento de esta matriz se asocian dos valores: *Población*, que es la cantidad de habitantes que hay en cada cuadrícula, y *Costo del Terreno* (por metro cuadrado). Los datos de población y terrenos se obtienen a partir de datos oficiales disponibles sobre el área en estudio, que para el presente trabajo, será la ciudad de Asunción, capital de la República del Paraguay.

De esta forma, obtenemos una matriz $\boldsymbol{M} \in \boldsymbol{N}^{mx4}$ con una fila por cada una de las m cuadrículas válidas y 4 columnas con información por cuadrícula, de:

- 1ª columna: fila para su ubicación en el mapa;
- 2ª columna: columna para su ubicación en el mapa;
- 3ª columna: población actual (dato utilizado para estimar demanda);
- 4ª columna: costo del terreno.

Debido a que el plano del área en estudio tiene en general una figura geométrica irregular, muchas cuadrículas caen fuera de los límites de la ciudad o en zonas no habitadas, con ríos, lagos o montañas. Por lo tanto, utilizando técnicas de matrices esparzas, a todas las cuadrículas que quedan fuera de la ciudad se les asigna un indicador de cuadrícula no válida (*flag*) y no se las cuenta entre las m cuadrículas válidas.

El costo de implementación de una central de telecomunicaciones es calculado de la siguiente forma:

$$y_i = \sum_{j=1}^{6} c_j(\boldsymbol{x}) \qquad \ldots\ i=1, 2, 3 \tag{2}$$

donde:

$c_1(\boldsymbol{x})$: costo total de planta externa, definidas por el vector de decisión $\boldsymbol{x}$;

$c_2(\boldsymbol{x})$: costos de terrenos donde serán instaladas las centrales;

$c_3(\boldsymbol{x})$: costos de edificios donde serán instaladas las centrales;

$c_4(\boldsymbol{x})$: costos de ingeniería que conlleva la instalación de las centrales;

$c_5(\boldsymbol{x})$: costos de equipos de conmutación;

$c_6(\boldsymbol{x})$: costos de equipos de transmisión entre las centrales definidas por $\boldsymbol{x}$.

Para la evaluación del costo de planta externa $c_1(\boldsymbol{x})$, se calcula las distancias de cada abonado a la central más cercana, conforme se ilustra en el siguiente ejemplo.

6.2. Problema de Prueba

Como problema de prueba para ejemplificar la presente propuesta se escogió el diseño de la planta externa de una empresa de telefonía básica para la ciudad de Asunción, dada la disponibilidad de datos para la misma. La Figura 4 representa el plano cuadriculado de la ciudad de Asunción, con los contornos indicando los elementos válidos de la matriz. Para este ejemplo, existen m = 499 cuadrículas válidas.

El vector de decisión para este ejemplo, al adoptar un número máximo de n = 14 centrales, será:

$$\boldsymbol{x} = (0, 0, 0, 0, 0, 0, 48, 92, 188, 232, 250, 319, 390, 423)$$

donde se observa que de las 14 centrales posibles, esta solución utiliza solo 8 centrales, ubicadas en las posiciones 48, 92, 188, 232, 250, 319, 390, y 423. Puede notarse además que el vector de decisión $\boldsymbol{x}$ tiene sus elementos x_i ordenados en forma creciente, lo que facilita detectar soluciones similares donde las centrales se encuentran simplemente permutadas.

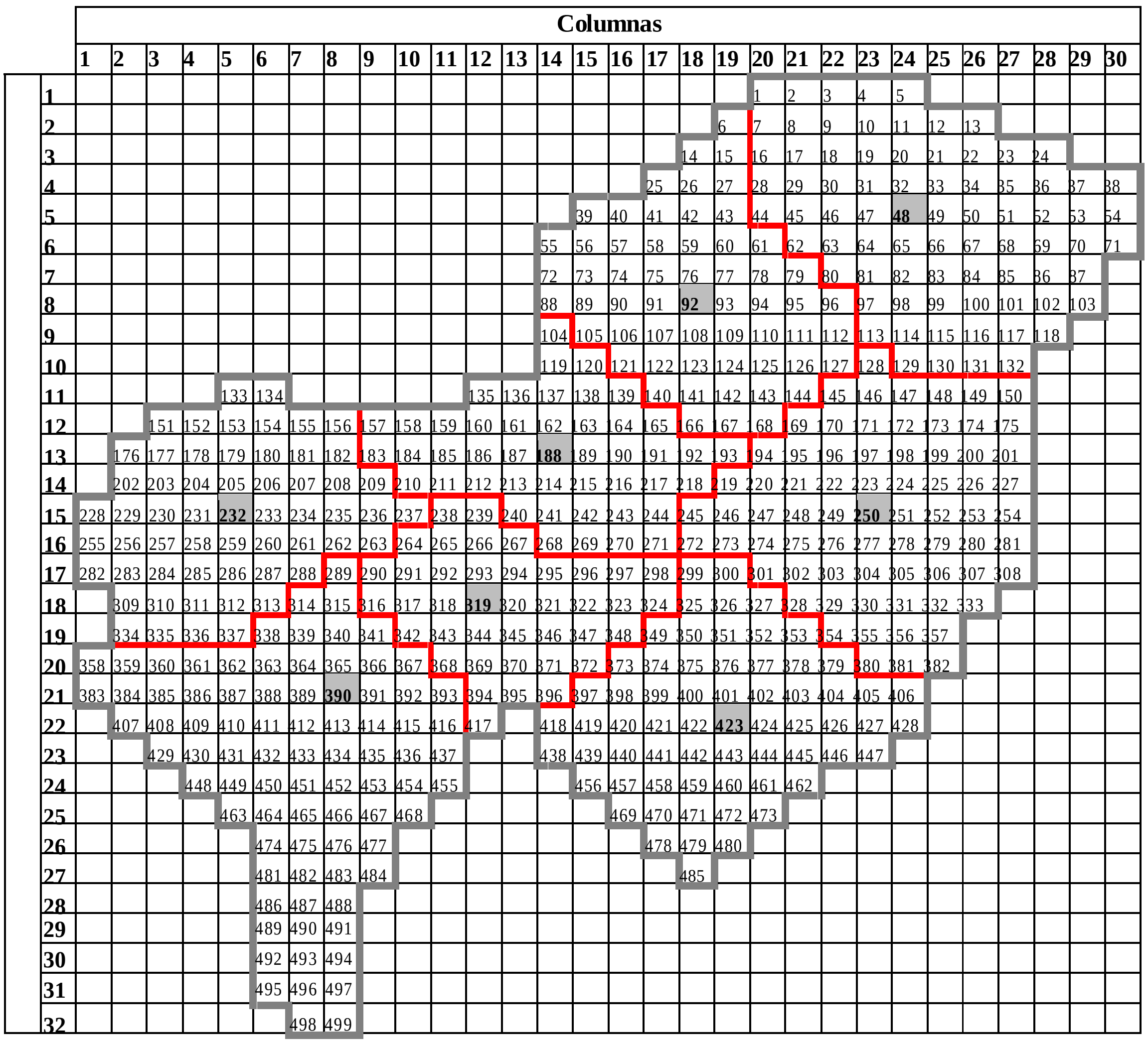

Figura 4. Ejemplo de división en cuadrículas de la ciudad de Asunción. Este plano indica los contornos que contienen los 499 elementos válidos de la matriz. Además, se observa un ejemplo de ubicación de 8 centrales con sus respectivas áreas de servicio.

6.3. Algoritmo Evolutivo Propuesto

El algoritmo evolutivo propuesto es el SPEA 2 cuya eficiencia en la búsqueda de soluciones se caracteriza por la obtención de soluciones Pareto óptimas y la diversidad de las mismas sobre el Frente Pareto. Este algoritmo utiliza una estrategia de asignación de fitness que incorpora información de densidad a fin de evitar la pérdida de posibles soluciones óptimas. El operador de truncamiento elimina aquellos individuos que están muy pegados unos a otros de forma a no perder puntos valiosos de la frontera y asegurar de esta forma que las soluciones encontradas en el frente Pareto, sean regularmente distribuidas. El proceso de encontrar los individuos no dominados en el archivo y la población, está basado en el concepto de dominancia Pareto. Cada vez que un individuo no dominado es encontrado, el mismo es comparado con los no dominados ya existentes en el archivo, y si el mismo es una solución, el individuo hallado es insertado en el archivo. Para esclarecer el procedimiento de aplicación del SPEA 2 en la planificación de centrales, a continuación se presenta un esquema de utilización del referido algoritmo.

6.3.1. Representación de soluciones y población inicial

Para la aplicación de los Algoritmos Evolutivos Multiobjetivos propuestos en el problema de prueba, cada individuo $x = (x_1, x_2, ..., x_i, ..., x_n)$ fue codificado usando un arreglo de números enteros x_i, tal que $0 \leq x_i \leq m$ $(m=499)$. En la figura 4, donde se representa el plano cuadriculado de Asunción, se puede apreciar los 499 valores no nulos de la matriz utilizada para los cálculos de costos de cada vector de decisión. La población inicial, cuyo tamaño se denotará como *"nind"* (número de individuos), es generada por un algoritmo heurístico de inicialización, en donde "*nmax*" indica el número máximo de centrales para cada vector de decisión. Este algoritmo genera una población inicial en forma *inteligente* de manera a obtener individuos que se aproximen razonablemente al conjunto de soluciones Pareto óptimas buscadas, minimizar de esta forma los tiempos de corridas. Para cada individuo de la población, se realiza un sorteo para saber cuantas centrales tendrá esa solución, y se ubican las centrales de tal forma a que las mismas esten ubicadas en los centros de demandas a fin de minimizar los costos de conexión de los abonadosa a su central correspondiente. El algoritmo heurístico de inicio de la población se describe a continuación.

Algoritmo heurístico de inicialización de la población inicial.
Leer parámetros: *nind, nmax*
Ordenar matriz de población de acuerdo al número de habitantes
Para *in*=1 hasta *nind*
    Generar un número aleatorio *N* entre 6 y *nmax*
    Dividir la población total en *N* partes: *parte=poblacion.total/N*
    Para *i*=1 hasta *N*
        Elegir punto x_i aleatoriamente entre las 5 ubicaciones más pobladas
        Hallar distancia euclidiana de x_i a todas las ubicaciones de la matriz de población
        Ordenar las distancias obtenidas de menor a mayor
        poblacion = 0
        Mientras *poblacion* es menor o igual a *parte*
            Sumar a *poblacion* la población de las ubicaciones más próximas a x_i
        Fin Mientras
        Eliminar de la matriz de población las ubicaciones que se agregaron a *poblacion*
        Hallar el centro geométrico P_i de todas las ubicaciones que se agregaron a *poblacion*
        Hacer $x_i=P_i$
    Fin Para
    Si $N < nmax$
        $x_i=0$ para todo i que no contiene una central (esto es, $N+1 \leq i \leq nmax$)

Fin Para
Eliminar centrales repetidas de cada individuo de la población inicial y ordenar centrales en orden creciente

Pseudocódigo 2. Algoritmo Heurístico de generación de la población inicial.

6.3.2. Evaluación de soluciones y función fitness

En la evaluación de la función fitness, se utilizaron los conceptos de dominancia Pareto en un contexto de minimización de funciones objetivos. De esta forma, cada vector de decisión es comparado con otro a través de las funciones objetivos de dichos vectores, de tal forma a determinar si un individuo i domina a otro individuo j. La función fitness(x) fue implementada conforme a lo especificado por el SPEA 2 de Zitzler.

Los valores de fitness calculados mediante este algoritmo, son utilizados en la selección de los individuos que pasarán a formar parte del archivo que contiene a los mejores individuos de la población. El referido algoritmo asigna a los individuos no dominados un fitness menor a 1, en cuanto que a los individuos dominados se les asigna un fitness mayor o igual a 1, con lo que todos los individuos tienen diferentes valores de fitness.

6.3.3. Selección

Se denomina como selección de ambiente a la acción de completar con los mejores individuos de cada generación, una población externa denominada archivo. El tamaño del archivo es fijo y no varía durante las corridas del algoritmo. El tamaño del archivo es fijo y no varía durante las corridas del algoritmo. Inicialmente, todos los individuos no dominados, cuyos fitness son menores que uno, son copiados al archivo de la siguiente generación. Si la cantidad de individuos no dominados es igual al tamaño establecido para dicho archivo, el paso de selección del ambiente está completo. Caso contrario existen dos posibilidades:

1) la cantidad de individuos es menor que el tamaño establecido para el archivo.
2) la cantidad de no dominados es mayor que el tamaño fijado para el archivo.

En el primer caso, se completa el archivo con los mejores individuos dominados en el archivo y la población de la generación anterior. En el segundo caso, un operador de truncamiento remueve iterativamente los individuos hasta que el conjunto de no dominados sea igual al tamaño establecido para el archivo.

6.3.4. Pseudocódigo del Algoritmo Evolutivo Multiobjetivo propuesto SPEA 2

En las corridas realizadas del algoritmo SPEA 2 se utilizaron los siguientes parámetros:

- Tamaño de la población (nind) = 100.
- Número máximo de centrales (nmax) = 14 a 20.
- Tamaño del archivo de no dominados (nptrue) = 100.
- Número máximo de generaciones (ngen) = 1000 a 3000.
- Probabilidad de cruzamiento (pc) = 0,7 a 0,9.
- Probabilidad de mutación (pm) = 0,1 a 0,3.

A continuación, se presenta el Pseudocódigo del algoritmo Multiobjetivo utilizado:

```
Programa principal SPEA 2
Leer los parámetros del SPEA 2: nind, nmax, ngen, pm, pc, nptrue
Generar una población usando el algoritmo heurístico (Pseudocódigo 1)
Generar un archivo vacío (conjunto externo)
Para gen=1 hasta ngen
        Eliminar centrales repetidas del individuo
        Evaluar funciones objetivo de cada individuo de la población
        Asignar fitness a cada individuo de la población y del archivo
        Calcular todos los individuos no dominados de la población y el archivo
        Actualizar el archivo con los individuos no dominados
        Si el tamaño del archivo es mayor que nptrue
                Reducir el tamaño del archivo con el operador de truncamiento
        Caso contrario
                Si el tamaño del archivo es menor que nptrue
                    Copiar los mejores individuos dominados del archivo y la población con fitness ≥ 1 al
                    archivo de la nueva generación hasta que el tamaño del archivo sea igual a nptrue
                Fin Si
        Si gen es menor que ngen
            Realizar torneo binario para seleccionar los individuos del archivo que formarán parte del
            conjunto de emparejamientos
            Realizar cruzamiento y mutación del conjunto de emparejamientos
            Actualizar la población del resultado del conjunto de emparejamientos
        Fin Si
        Incrementar contador de generaciones (gen=gen + 1)
Fin Para
Salvar el archivo (conjunto de no dominados)
```

Pseudocódigo 3. Algoritmo SPEA 2 implementado.

6.3.5. Resultados Obtenidos

Las soluciones obtenidas para el problema de prueba son presentadas en la tabla 1. Las mismas fueron obtenidas mediante sucesivas corridas del algoritmo SPEA 2, luego de haber descartado otros algoritmos evolutivos que no lograron el nivel de desempeño obtenido con el SPEA 2.

En la tabla 1 se puede apreciar que la mejor solución para el año base 2002 es la número 1, que utiliza 8 centrales. Para el año 2004, la mejor solución es la número 10 que requiere de 11 centrales, mientras que para el año 2007, la cantidad óptima de centrales es de 14 (solución número 19). Claramente, los tres objetivos conflictuan entre sí por lo que el planificador deberá decidir cual es la mejor relación de compromiso entre su inversión a corto plazo y el costo que podrá llegar a tener la red a mediano y largo plazo. Es interesante enfatizar que al utilizar un algoritmo evolutivo multiobjetivo, el planificador no solo encuentra las mejores soluciones para cada objetivo, sino toda la gama de soluciones de compromiso Pareto óptimas entre estos objetivos, por lo que se facilita la toma de decisión conciente.

Cabe mencionar que la solución efectivamente implementada para la ciudad de Asunción no es una solución Pareto óptima, y de hecho conlleva un costo mucho mayor que cualquiera de las soluciones calculadas con la metodología propuesta, sin importar cual de las 3 funciones objetivos se considere. En efecto, a la fecha existen en la ciudad de Asunción 8 centrales telefónicas, por lo que resulta razonable compararla con la solución número 1 de la Tabla 1 que también utiliza 8 centrales. Como consecuencia de esta comparación se puede notar que el costo de la infraestructura existente es al menos 4 % superior a la referida solución número 1, considerando el año base 2002. Esto es, la solución aquí propuesta hubiese representado un ahorro del orden de los cuatro millones de dólares a la inversión que fuera realizada para atender la demanda correspondiente al año 2002.

	Tabla de valores de soluciones no dominadas encontradas														Costos en US$		
Solución	**Vector de decisión**														**Año 2002**	**Año 2004**	**Año 2007**
1	0	0	0	0	0	0	48	92	188	232	250	319	390	423	103.685.700	5.162.463	12.765.596
2	0	0	0	0	0	32	91	128	185	232	304	321	390	423	103.884.860	5.163.835	12.753.748
3	0	0	0	0	0	49	94	163	232	250	265	372	390	424	103.890.330	5.160.239	12.745.146
4	0	0	0	0	0	49	94	163	232	250	265	321	390	423	103.910.600	5.160.081	12.744.893
5	0	0	0	0	0	49	93	189	232	250	265	372	390	424	103.918.700	5.159.084	12.742.180
6	0	0	0	0	32	76	147	163	232	265	303	347	390	443	104.040.180	5.155.810	12.718.956
7	0	0	0	0	32	92	147	163	232	265	303	347	390	443	104.047.160	5.154.795	12.716.454
8	0	0	0	32	76	147	180	189	258	292	303	372	390	443	104.550.300	5.155.748	12.703.861
9	0	0	0	32	92	147	180	189	258	292	303	372	390	443	104.557.280	5.154.733	12.701.359
10	0	0	0	32	92	147	189	207	258	292	303	372	390	443	104.638.420	5.154.383	12.700.554
11	0	0	32	76	147	180	189	258	292	303	364	372	435	443	105.229.590	5.156.761	12.691.531
12	0	0	32	92	147	180	189	258	292	303	364	372	435	443	105.236.570	5.155.746	12.689.029
13	0	0	32	92	147	189	207	258	292	303	364	372	435	443	105.338.540	5.155.729	12.689.055
14	0	32	76	147	163	180	258	274	292	305	364	372	435	443	106.080.390	5.158.318	12.680.680
15	0	32	92	147	163	180	258	274	292	305	364	372	435	443	106.087.370	5.157.303	12.678.178
16	0	32	92	147	163	208	231	246	293	330	363	372	414	443	106.414.660	5.157.066	12.678.038
17	32	76	147	163	180	258	274	292	305	364	372	379	435	443	107.152.020	5.162.245	12.675.595
18	32	92	147	163	180	258	274	292	305	364	372	379	435	443	107.159.000	5.161.230	12.673.093
19	32	92	147	163	180	246	258	292	305	364	372	379	435	443	107.162.610	5.160.889	12.672.366
Red Nacional	**0**	**0**	**0**	**0**	**0**	**0**	**109**	**204**	**208**	**274**	**288**	**293**	**398**	**436**	**107.760.000**	**5.219.810**	**12.908.475**

Tabla 1. Tabla de soluciones no dominadas encontradas.

Dado que en la metodología propuesta existen varias soluciones no dominadas entre sí, y a fin de simplificar la tarea del planificador, se presenta en la Figura 5 una sugerencia pragmática para elegir una de entre todas las soluciones Pareto óptima. La idea is traer a valor presente las inversiones a mediano y largo plazo de forma a tener un único objetivo que permita comparar todas las alternativas de solución encontradas por el planificador, en el tradicional contexto mono-objetivo. En la Figura 5 puede notarse que en la simplificación propuesta, el número óptimo de centrales es 8, lo que coincide con el número existente de centrales en la ciudad de Asunción.

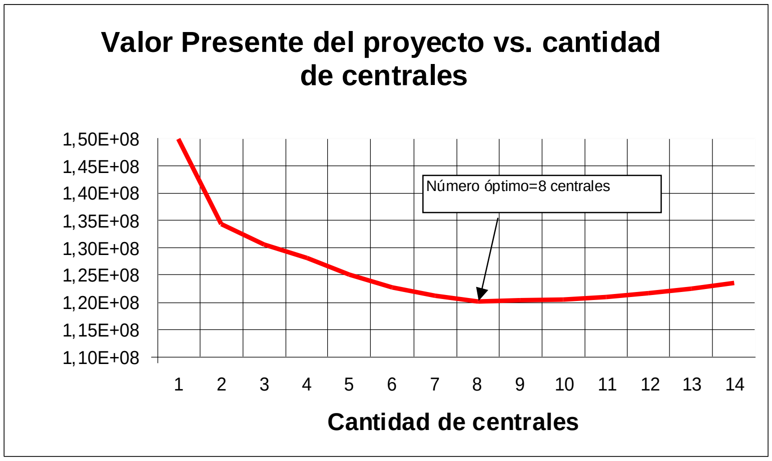

Figura 5. Este diagrama muestra el costo de inversión en función de la cantidad de centrales. Se observa el número óptimo de centrales que es igual a 8.

Se nota que la utilización de algoritmos evolutivos Multiobjetivos como el SPEA2, proporciona al planificador de redes un conjunto de soluciones Pareto óptimas para la correcta ubicación de las centrales, de forma a minimizar los costos iniciales de inversión y las inversiones de expansión a mediano y a largo plazo.

Conforme con los resultados obtenidos en este trabajo, se puede aseverar que las soluciones distribuidas sobre el frente Pareto son en su mayoría dominantes con respecto a las soluciones efectivamente implementadas por empresas del área que se limitaron a utilizar herramientas tradicionales de cómputo en sus estudios de planificación. De hecho, en las pruebas realizadas, las soluciones obtenidas con el SPEA2 superaron claramente a las obtenidas por otros métodos tradicionales.

En definitiva, se puede afirmar que el empleo de algoritmos evolutivos Multiobjetivos para la planificación, dimensionamiento y optimización de redes de telecomunicaciones, ofrece una perspectiva más amplia y eficiente que permite a los planificadores decidir entre un conjunto de soluciones óptimas, manejando los diversos aspectos de la red que se consideren necesarios para minimizar los costos en juego.

7. Conclusión

La vida cotidiana está rodeada de MOPs y hoy en día es posible resolverlos en tiempos razonables debido a la utilización de técnicas de inteligencia artificial. En este trabajo solo se presentó uno, los MOEAs, pero hay otros como Colonias de Hormigas.

Todas estas técnicas de IA consisten en metaheurísticas, pero algunos aún no tienen ninguna implementación para MOPs, como el Cúmulo de Partículas, lo que hace que el área de optimización multiobjetivo sea línea de investigación abierta para seguir resolviendo MOPs con nuevas técnicas.

8. Referencias

- Almeida, C., Amarilla, N. y Barán, B, Optimización Multiobjetivo en la Planificación de Centrales Telefónicas, Centro Nacional de Computación.
- Sotelo, A., Von Lucken, C. y Barán, B., Multiobjective Evolutionary Algorithms in Pump Scheduling Optimisation, Centro Nacional de Computación.
- Duarte, S. y Barán, B., Algoritmos Evolutivos para el Diseño de Redes Económicas y Confiables, Centro Nacional de Computación.
- Van Veldhuisen, D. A., Multiobjective Evolutionary Algorithms: Classifications, Analyses and New Innovations, PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology. Ohio, EE.UU. May, 1999.