

# Estudio de la aplicabilidad de un enfoque evolutivo para la descomposición en Suma de Minkowski.

María Teresa Taranilla<sup>(1)</sup>

Gregorio Hernández Peñalver<sup>(1)</sup>

Mario Guillermo Leguizamón<sup>(2)</sup>

Edilma Olinda Gagliardi<sup>(1)</sup>

Departamento de Informática  
Facultad de Ciencias Físico, Matemáticas y Naturales  
Universidad Nacional de San Luis, Argentina  
{tarani, oli, legui}@unsl.edu.ar  
Fax: 54-2652-430224

Departamento de Matemática Aplicada  
Facultad de Informática  
Universidad Politécnica de Madrid, España  
gregorio@fi.upm.es  
Fax: 34-91-3367426

## Resumen:

Dados dos conjuntos  $P$  y  $Q \subset \mathbf{R}^2$  la suma de Minkowski de  $P$  y  $Q$ , denotada por  $P \oplus Q$  se define como  $P \oplus Q = \{p + q : p \in P, q \in Q\}$  donde  $p+q$  es el vector suma de los vectores  $p$  y  $q$ .

El problema que nos planteamos es el problema inverso al resuelto por la suma de Minkowski. Dado un polígono  $S$ , ¿existen polígonos  $P$  y  $Q$  tales que  $S$  es la suma de Minkowski de  $P$  y  $Q$ , es decir,  $S = P \oplus Q$ ?

Este problema se puede resolver con un algoritmo de complejidad exponencial. Debido a la complejidad inherente de este problema, se propone un enfoque evolutivo para su resolución. La propuesta incluye la definición del problema en términos de una función objetivo, el diseño e implementación de un algoritmo genético y su aplicación a un conjunto de instancias del problema.

**Palabras claves:** Sumas de Minkowski. Geometría Computacional. Algoritmos genéticos.

## 1. Introducción

Dados dos conjuntos  $P$  y  $Q \subset \mathbf{R}^2$ , la suma de Minkowski de  $P$  y  $Q$ , denotada por  $P \oplus Q$  se define como  $P \oplus Q = \{p + q : p \in P, q \in Q\}$  donde  $p + q$  es el vector suma de los vectores  $p$  y  $q$ . Es decir que dados los puntos  $p = (p_x, p_y)$  y  $q = (q_x, q_y)$ , tenemos que  $p + q = (p_x + q_x, p_y + q_y)$ .

En el campo de la robótica, la Suma de Minkowski se utiliza en la planificación de movimientos de robots para determinar el espacio prohibido en el cálculo de un camino libre de colisiones para un robot poligonal en una escena poligonal. Los obstáculos se agrandan calculando la suma de Minkowski entre el polígono que representa el robot y aquellos que representan los obstáculos [2][4][5].

<sup>(1)</sup> Proyecto Tecnologías Avanzadas de Bases de Datos 22/F314, Departamento de Informática, UNSL.

Proyecto AL2002-1010-2.43/ AL2003-1010-2.55/ AL2004-1010-2.53 Geometría Computacional, UPM.

<sup>(2)</sup> Laboratorio de Investigación y Desarrollo en Inteligencia Computacional, Departamento de Informática, UNSL.

Consideremos un obstáculo  $P$  y un robot  $R$  que se mueve por el plano mediante sucesivas traslaciones. La ubicación del robot en el plano está determinada por un punto interior  $r$  que es el punto de referencia del robot. Si tomamos dicho punto interior  $r$  como origen de coordenadas y construimos  $R'$ , figura simétrica de  $R$  respecto del origen,  $P \oplus R'$  es el conjunto de ubicaciones del punto de referencia de  $R$  tales que  $P \cap R \neq \emptyset$ . Esta suma se denomina *C-obstáculo* y representa el conjunto de puntos en los cuales está prohibido ubicar al robot, pues colisionaría con el obstáculo  $P$ .

Supongamos que conocemos la suma de Minkowski de cada uno de los obstáculos con el robot, pero por algún motivo no conocemos la forma original del robot, por lo tanto estamos interesados en descomponer cada uno de los C-obstáculos para detectar la forma original del robot.

Nos planteamos entonces el problema inverso al resuelto por la suma de Minkowski. Dado un polígono  $S$ , queremos encontrar polígonos  $P$  y  $Q$  tales que la suma de Minkowski de  $P$  y  $Q$  sea  $S$ , es decir,  $S = P \oplus Q$ .

Analizamos el caso donde el polígono  $S$  es un polígono convexo, por lo tanto los posibles sumandos  $P$  y  $Q$  también deben ser convexos. En otras palabras, dado  $S$  un polígono convexo, buscamos dos polígonos convexos  $P$  y  $Q$  tales que  $S = P \oplus Q$ .

El algoritmo que detecta si un polígono  $S$ , de  $n$  lados, admite un polígono de  $m$  lados como sumando de Minkowski toma un tiempo  $O(n^m)$ . En cambio, si queremos saber si el polígono  $S$  admite un sumando de Minkowski de número no precisado de lados, entonces la complejidad un posible algoritmo es exponencial [6][7]. Debido a la complejidad del algoritmo que resuelve este problema hemos decidido tratar de resolverlo usando un enfoque evolutivo.

Este artículo está organizado de la siguiente manera: en la sección 2 mostramos la formulación del problema. Seguidamente, en la sección 3 se presenta una propuesta general para resolverlo utilizando Algoritmos Genéticos y un análisis de posibles situaciones que pueden ocurrir en la evaluación de una solución. En la sección 4 se presentan resultados preliminares sobre un conjunto de instancias generadas para probar la aplicabilidad del algoritmo genético. Finalmente, las conclusiones y propuestas de trabajos futuros son consideradas.

## 2. Formulación del problema

Dado un polígono convexo  $S$  de  $n$  lados, queremos encontrar dos polígonos convexos  $P$  y  $Q$  de  $k$  y  $k'$  lados respectivamente, tales que  $S$  sea la suma de Minkowski de  $P$  y  $Q$ , es decir,  $S = P \oplus Q$ . Para ello elegimos cierta cantidad de lados de  $S$  en orden y armamos un polígono  $P$ . Luego, el polígono  $Q$  se obtiene con los lados restantes de  $S$ . Una vez obtenidos los candidatos  $P$  y  $Q$ , debemos comprobar que sean polígonos simples, convexos y cerrados.

Cabe aclarar que, un polígono es simple cuando ninguno de sus lados se interseca entre sí, es convexo si y solo si el ángulo interior a cualquier vértice es menor o igual que  $\pi$  y es cerrado si cumple que la suma vectorial de sus lados es igual a cero.

Si ambos polígonos cumplen estas tres condiciones, calculamos  $P \oplus Q$ , obteniendo  $S'$ . Luego debemos comparar  $S$  con  $S'$ , para saber cuánto se aproxima la solución obtenida a  $S$ . Para ello debemos determinar una medida de la diferencia entre  $S$  y  $S'$ . La medida que hemos elegido es el área de la diferencia simétrica entre ambos polígonos, que denotaremos  $\Delta$ , y será la función objetivo a minimizar.

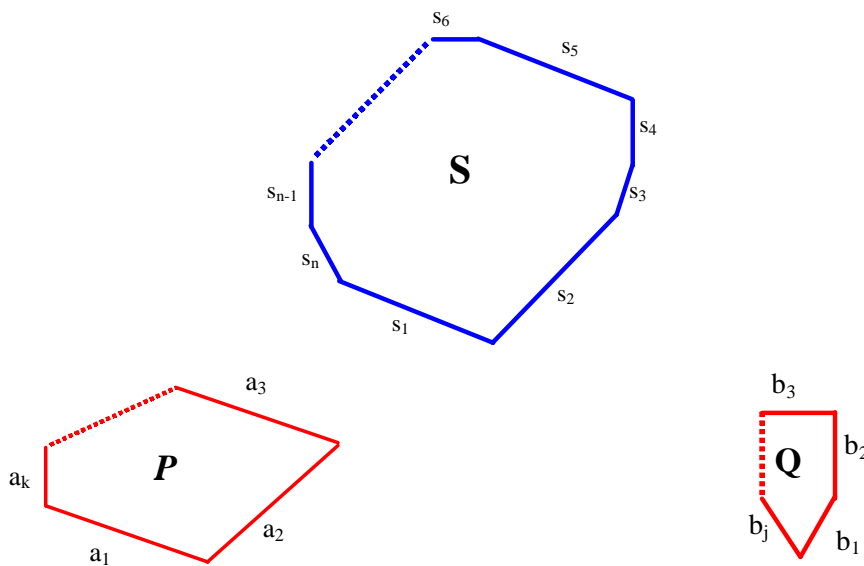
Más formalmente, sea  $S$  un polígono convexo de  $n$  lados. Sea  $\mathcal{P}$  el conjunto de todos los posibles polígonos de  $k$  lados, con  $2 \leq k \leq n-2$ . El problema consiste en:

$$\text{Min } f(P) = \text{Area}((P \oplus Q) \Delta S)$$

donde  $P$  y  $Q \in \mathcal{P}$  y  $Q$  se construye con los lados de  $S$  que no forman parte del polígono  $P$ . Denominaremos al problema formulado en estos términos como  $SMINK^{-1}$ .

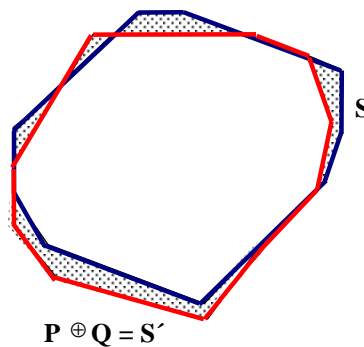
Es importante destacar algunos aspectos en relación a la característica del polígono  $S$ , ya que puede suceder que no existan dos polígonos  $P$  y  $Q$  que cumplan que  $P \oplus Q$  sea exactamente igual a  $S$ . Sin embargo, se podrían obtener polígonos  $P$  y  $Q$  tales que  $S' = P \oplus Q$  y  $S'$  sea lo más aproximada a  $S$  desde el punto de vista morfológico.

En el siguiente ejemplo se muestra en forma general el problema descripto previamente. En la figura 1 vemos un polígono convexo  $S = \langle s_1, s_2, \dots, s_n \rangle$  de  $n$  lados. Suponiendo que sus lados están ordenados en el sentido contrario a la agujas del reloj, se eligen  $k$  lados de  $S$  siguiendo ese orden para obtener  $P = \langle a_1, a_2, \dots, a_k \rangle$ . El polígono  $Q$  se obtiene con los lados restantes de  $S$  tal que, para  $Q = \langle b_1, b_2, \dots, b_j \rangle$ , se cumple que  $j = n - k$ . Ambos polígonos deben ser cerrados, simples y convexos.



*Figura 1: Ejemplo de un polígono  $S$  y una posible descomposición en  $P$  y  $Q$ .*

Una vez obtenidos los candidatos  $P$  y  $Q$ , y habiendo comprobado que ambos son polígonos simples, convexos y cerrados, calculamos  $S' = P \oplus Q$ , y luego obtenemos el área de la diferencia simétrica entre  $S$  y  $S'$ . La diferencia simétrica entre ambos polígonos es el área sombreada en la figura 2. A medida que esta área tiende a cero, mayor es el parecido morfológico entre  $S$  y  $S'$ .



*Figura 2: Diferencia simétrica entre  $S$  y  $P \oplus Q$*

Puede ocurrir que con la sucesión de lados elegidos para armar el polígono  $P$  (o el polígono  $Q$ ), se obtenga un polígono que no sea simple, que no sea convexo o que no sea cerrado, en ese caso  $P$ (o  $Q$ ) son soluciones no factibles. En la siguiente sección se analiza detalladamente la factibilidad de las soluciones desde la perspectiva de la representación usada en el algoritmo genético.

### 3. Propuesta para resolver SMINK<sup>-1</sup> a través de un Algoritmo Genético

Los Algoritmos Genéticos (AGs) son una de las técnicas más conocidas de resolución de problemas dentro de lo que se ha definido como "Algoritmos Evolutivos", término que agrupa a los Algoritmos Genéticos, las Estrategias Evolutivas y la Programación Evolutiva. Los algoritmos evolutivos pueden ser considerados un método de búsqueda muy especial, en el que las soluciones al problema son capaces de reproducirse entre sí, combinando sus características y generando nuevas soluciones. Además, permiten tratar problemas de optimización donde el objetivo es encontrar un conjunto de parámetros que minimizan o maximizan una función objetivo[3].

Para llevar a la práctica la propuesta anterior y concretarla en un algoritmo genético, debemos especificar los siguientes elementos:

- Una representación cromosómica
- Una población inicial
- Una medida de evaluación
- Un criterio de selección
- Operaciones de recombinación y de mutación

Previamente diremos que  $n$ , el número de lados de  $S$  es el tamaño del problema. Y además, que se puede considerar al espacio de soluciones al conjunto formado por todos los polígonos  $P$  de  $k$  lados, tal que  $2 \leq k \leq n-2$ . El polígono  $P$  sumado a otro polígono  $Q$ , obtenido como el complemento de  $P$ , nos permitirá encontrar la solución que más se aproxime a  $S$ .

A continuación se describen las principales componentes del algoritmo genético diseñado para encarar el problema SMINK<sup>-1</sup>.

#### 3.1 Representación

La mayoría de las veces, una codificación correcta es la clave de una buena resolución del problema. Para este problema, en principio vamos a usar un string binario de  $n$  bits para representar un polígono  $P$ . Un 1 en el gen  $i$  significa que el lado  $i$  de  $S$  forma parte del polígono  $P$ , y un 0 que no forma parte, con la restricción de que en el cromosoma hay entre 2 y  $n-2$  bits con valor 1.

Una forma de obtener  $P$  de un cromosoma es recorriendo el cromosoma de izquierda a derecha y por cada 1 que encontramos tomamos el correspondiente lado de  $S$ , dado por la posición. Se debe cumplir que  $k$ , la cantidad de lados de  $P$ , esté comprendida entre  $2 \leq k \leq n-2$ .

El siguiente ejemplo muestra en la figura 3 a  $S = \langle s_1, s_2, s_3, s_4, s_5, s_6 \rangle$ , un polígono convexo de 6 lados, que será descompuesto y al que se debería aproximar la solución obtenida al calcular la suma de Minkowski de  $P$  y  $Q$ .

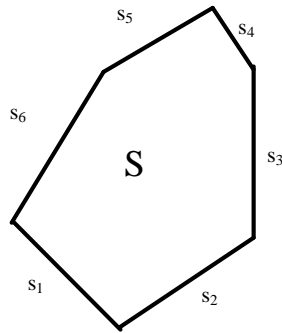


Figura 3: Polígono S que representa una posible instancia de  $SMINK^{-1}$

En la figura 4 observamos el cromosoma C representa una secuencia de lados que permitirán obtener el polígono P.

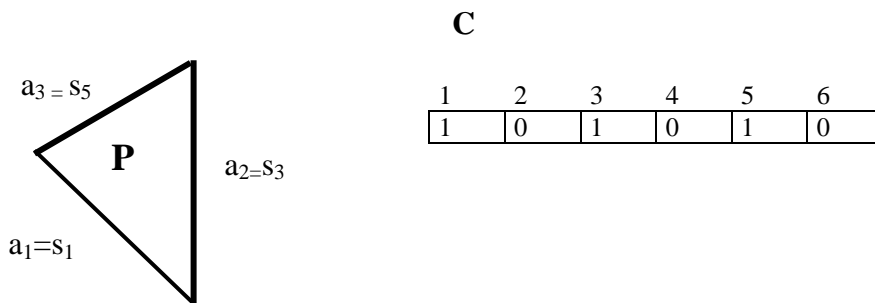


Figura 4: Polígono P y su representación a través del cromosoma C.

El polígono Q se obtiene en forma similar considerando los bits en 0 de C. (figura 5)

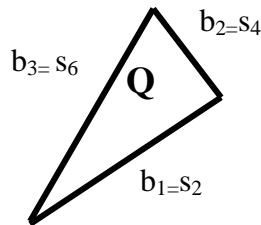


Figura 5: Polígono Q obtenido de los bits en 0 de C.

En este caso, con las secuencias de lados elegidas obtenemos a polígonos que cumplen en ser cerrados, simples y convexos.

Otros cromosomas pueden dar lugar a soluciones no factibles. Por ejemplo, el cromosoma C1 da lugar a un polígono que no es simple.

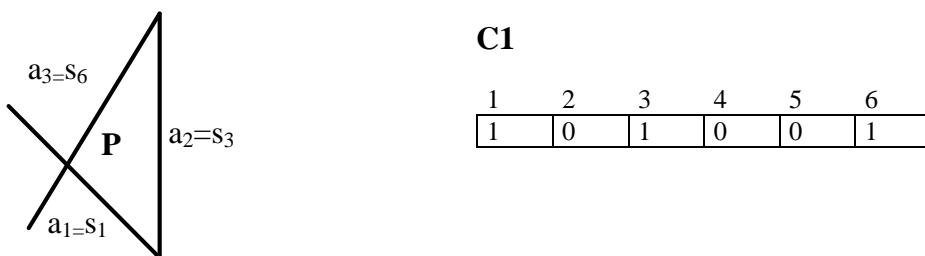
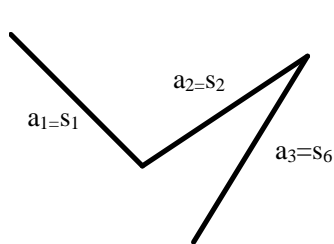


Figura 6: Polígono P no simple, obtenido de C1

El cromosoma C2 da lugar a una cadena poligonal no cerrada y no convexa.



**C2**

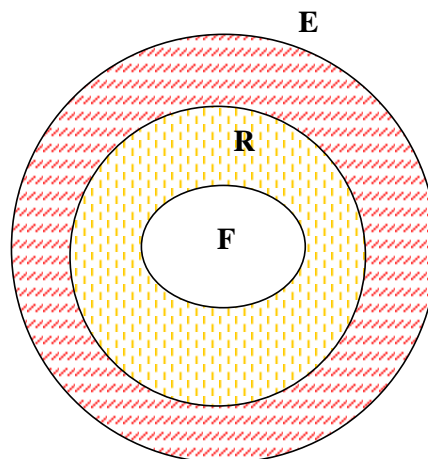
1	2	3	4	5	6
1	1	0	0	0	1

**Figura 7: Cadena poligonal no convexa y no cerrada obtenida a partir de C2.**

En la figura 8 se muestra el espacio de búsqueda del algoritmo genético representado por  $E$ , donde  $E = \{0,1\}^n$ . En este espacio existen al menos dos tipos de soluciones no factibles.

El primer tipo está representado por el conjunto  $E-R$ , donde  $R = \{x \in E / 2 \leq n_1(x) \leq n-2\}$  y  $n_1(x)$  indica la cantidad de bits igual a 1 en la cadena  $x$ . Este tipo de soluciones está constituido por aquellas soluciones o cromosomas donde  $k$ , el número de bits en 1, cumple que  $k < 2$  o  $k > n-2$ .

El segundo tipo de soluciones no factibles son aquellas que estando en  $R$  no satisfacen alguna de las condiciones requeridas a los polígonos de ser cerrados, simples o convexos, es decir pertenecen a  $R-F$ , donde  $F$  es el conjunto de soluciones factibles del problema.



**Figura 8: Subdivisión del espacio de búsqueda para el algoritmo genético**

De esta manera el conjunto de soluciones factibles de este espacio de búsqueda podría quedar reducido a un pequeño porcentaje del mismo según las características del polígono  $S$ .

Para el manejo de problemas con restricciones, como el presentado en este artículo, existen diferentes mecanismos o técnicas en los enfoques evolutivos. Estos incluyen penalización de las soluciones no factibles, uso de decodificadores, representaciones especiales que preservan la factibilidad de las soluciones y muchas otras [3].

En este estudio se usó la técnica de penalización para soluciones no factibles. El factor de penalización varía según el grado de violación de las propiedades que deberían cumplir los polígonos soluciones. Si la solución pertenece a  $E-R$  no es penalizada, en cambio, se lleva a cabo un simple procedimiento de reparación que se implementa, ya sea añadiendo o eliminando bits en 1 en forma aleatoria hasta que la solución sea parte del conjunto  $R$ . Si la solución pertenece al conjunto  $R$  pero no da lugar a un polígono simple, convexo y cerrado, se le aplica una penalización acorde a la importancia de la condición no satisfecha. La siguiente ecuación refleja la función *eval* conformada a partir de  $f(x)$ , la función objetivo, y el término de penalización  $P(x)$ , donde  $P(x)$  retorna un valor acorde a las condiciones no satisfechas por los polígonos  $P$  y  $Q$  obtenidos de  $x$  la que representa una posible solución.

$$eval(x) = \begin{cases} f(x) & \text{si } x \in F \\ f(x) + P(x) & \text{si } x \in R - F \end{cases}$$

Cabe aclarar que éste no es el único enfoque posible para el manejo de restricciones y que otros enfoques disponibles podrían eventualmente ser implementados [1].

### 3.2 Población inicial

El proceso de inicialización crea una población de cromosomas donde cada cromosoma es un string binario de  $n$  bits, donde  $n$  es la cantidad de lados de  $S$  y cada bit es generado aleatoriamente, debemos controlar la factibilidad de cada cromosoma generado, es decir que el número de bits en 1 de cada cromosoma esté comprendido entre 2 y  $n-2$ . Si esto no se cumple, se debe realizar un proceso simple de reparación para asegurar la factibilidad de las posibles soluciones.

### 3.3 Función de evaluación

La función de evaluación *eval* recibe como parámetro el cromosoma que representa una secuencia de lados que formarán parte de  $P$ . Se obtiene  $Q$  como complemento del  $C$  recibido como parámetro.

Se deben decodificar esos cromosomas obteniendo la representación geométrica del polígono  $P$  y del polígono  $Q$ . Posteriormente se debe comprobar la factibilidad de la solución es decir, comprobar si la secuencia de lados que conforma tanto a  $P$ , como la que conforma a  $Q$ , forman polígonos cerrados, simples y convexos. Si alguno de los polígonos no cumple estas condiciones la solución no es factible y retorna un valor acorde a la penalización asignada.

Si ambos polígonos cumplen con estas condiciones, se calcula  $S'$  como  $P \oplus Q$ , luego se calcula la diferencia simétrica entre  $S$  y  $S'$  y devuelve el área de la diferencia simétrica de  $S$  y  $S'$ .

Función **eval** (C: cromosoma)

Obtener P a partir de C

Procesar (P): Si P no está cerrado, no es simple, o no es convexo,  
no es una solución factible.

Obtener Q a partir de C

Procesar (Q): Si Q no está cerrado, no es simple o no es convexo,  
no es una solución factible.

**Si** ( P y Q son factibles) **entonces**

Calcular  $S' = P \oplus Q$

Calcular la diferencia simétrica entre S y  $S'$

Calcular el área de la diferencia simétrica entre S y  $S'$

Retornar área de la diferencia simétrica

**sino**

Retornar un valor acorde a la penalización asignada.

Figura 10: pseudo código de la función de evaluación eval.

### 3.4 Descripción de los operadores genéticos

Utilizamos el crossover de un punto que intercambia subcadenas determinadas en punto aleatorio. La operación de mutación usada consiste en reemplazar con cierta probabilidad el valor

de un bit. El resultado de la aplicación de los operadores genéticos puede dar lugar a soluciones no factibles. En principio verificamos si la solución resultante pertenece a E-R si esto ocurre, se realiza un proceso simple de reparación para asegurar la factibilidad de las posibles soluciones como el descrito en la sección 3.1 para llevar la solución hacia el conjunto R.

El operador de selección utilizado es el de *selección por torneo*, este método selecciona en un cierto número de candidatos (en nuestro caso 2 individuos) y elige el mejor individuo entre ellos para que pase a la siguiente generación.

#### 4. Experimentos

Se realizó un estudio preliminar para determinar la aplicabilidad de un enfoque evolutivo para resolver el problema SMINK<sup>-1</sup>. Para ello se generó un conjunto de 14 instancias de tamaño relativamente pequeño, a fin de analizar el comportamiento del algoritmo genético. Cada instancia fue generada calculando la suma de Minkowski de dos polígonos convexos cualesquiera obtenidos en forma aleatoria. Este proceso asegura la existencia de al menos una solución factible.

Los tamaños de las instancias corresponden a polígonos de entre 16 a 41 vértices. El algoritmo genético se ejecutó con una población de 50 individuos, con una probabilidad de crossover  $p_c= 0.65$ , una probabilidad de mutación  $p_m= 0.1$  y 10000 generaciones. Para cada instancia se replicaron 20 corridas usando diferentes semillas.

Instancia	MVO	#F	#FE	#NFE	%E
sm-16	0	20	688	499313	0.00140
sm-18	0	20	2361	497639	0.00470
sm-19a	0	20	862	499138	0.00170
sm-19b	0	20	1627	498373	0.00330
sm-20a	0	20	1327	498673	0.00270
sm-20a	0	20	838	499162	0.00170
sm-21	0	20	582	499418	0.00120
sm-22a	0	20	279	499721	0.00050
sm-22b	0	20	121	499879	0.00020
sm-28a	0	20	64	499936	0.00012
sm-28b	0	20	16	499984	0.00003
sm-28c	0	20	41	499959	0.00008
sm-39	0	18	6	499994	0.00001
sm-41	0	15	4	499996	0.000008

*Tabla 1. Resultados de la aplicación del AG a 14 instancias de SMINK<sup>-1</sup>*

La Tabla 1 muestra un resumen de los resultados a partir de la aplicación del AG a las 14 instancias creadas de SMINK<sup>-1</sup>. Dicha tabla está dividida en 6 columnas. La primera columna muestra los nombres de las instancias (el nombre incluye su tamaño). Cada uno de los valores mostrados en las restantes columnas son valores promedio de las 20 ejecuciones por cada instancia. La columna MVO muestra el Mejor Valor Objetivo correspondiente a las soluciones factibles encontradas; #F muestra la cantidad de soluciones factibles encontradas en las 20 corridas; #FE y #NFE muestran (respectivamente) el número de soluciones Factibles y No Factibles Evaluadas por el algoritmo genético a través de las 10000 generaciones. Por último, %E muestra la razón entre #FE y #NFE cuyo valor indica el porcentaje del número soluciones factibles con respecto al número de soluciones no factibles evaluadas.

Los valores de la columna MVO evidencian un muy buen rendimiento del AG dado que el mejor valor encontrado corresponde al mínimo valor posible para la diferencia simétrica de dos polígonos superpuestos, es decir el polígono dado como entrada y el polígono obtenido por el AG.



Los valores de #F indican que para la mayoría de las instancias, el AG fue capaz de alcanzar soluciones factibles en cada corrida y que dichos valores, según MVO, corresponden al óptimo. Sin embargo, a medida que el tamaño de las instancias crece, el número de soluciones factibles encontradas comienza a disminuir esto se observa en las instancias *sm-39* y *sm-41*. Esta última observación está relacionada con las tres columnas siguientes: puede apreciarse que independientemente del tamaño de las instancias, el número de soluciones factibles evaluadas es ínfimo respecto al de soluciones no factibles. Esta situación es aún más extrema, a medida que aumenta el tamaño de las instancias. Estos resultados preliminares sobre instancias pequeñas indican que el problema  $SMINK^{-1}$  no es un problema trivial en el sentido que el espacio de búsqueda (asumiendo una representación binaria para las soluciones) está constituido por un reducido porcentaje de soluciones factibles. Esta situación puede implicar serias dificultades para cualquier método de búsqueda heurístico, incluyendo a los algoritmos genéticos. Dichas dificultades afectan principalmente a las técnicas utilizadas para el manejo de restricciones y su implementación dado que no sólo el espacio factible puede ser muy reducido, sino que también puede estar desconectado, es decir, constituido por varios conjuntos dispersos en el espacio de búsqueda total.

## 5. Conclusiones

En este artículo se ha presentado una propuesta para resolver un problema de geometría computacional ( $SMINK^{-1}$ ) a través de algoritmos genéticos. Se consideró un conjunto de instancias pequeñas para realizar un estudio preliminar sobre el comportamiento del algoritmo genético y las potenciales dificultades que pudieran detectarse en la resolución de  $SMINK^{-1}$ .

Como parte del desarrollo del software, es importante destacar que la implementación de la función de evaluación fue más compleja de lo que se esperaba ya que se requirió el desarrollo de varias funciones auxiliares no triviales, para comprobar si los polígonos obtenidos en la decodificación cumplían las características geométricas deseadas.

Los resultados obtenidos muestran un buen desempeño del algoritmo genético a pesar de la ínfima proporción de la región factible en relación al espacio total. Sin embargo, aun resta considerar instancias de mayor tamaño y un análisis detallado de las implicancias que podría tener en el desempeño del algoritmo genético, la característica del espacio de búsqueda de  $SMINK^{-1}$ . Siguiendo en esta dirección, futuros trabajos incluyen la consideración de instancias de mayor tamaño, un análisis del espacio de búsqueda de  $SMINK^{-1}$ , la posibilidad de utilizar otro tipo de representación para las soluciones y también, métodos alternativos para el manejo de restricciones.

## Referencias

- [1] Coello Coello, Carlos A. Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art, *Computer Methods in Applied Mechanics and Engineering*, Vol. 191, No. 11--12, pp. 1245--1287, January 2002.
- [2] De Berg, M; Kreveld, Overmars, M; Schwarzkopf. *Computational Geometry: algorithms and applications*, Springer Verlag, 1997
- [3] Michalewicz, Zbigniew. *Genetic algorithms + data structures = evolution programs*, Springer Verlag, 1997
- [4] Latombe, J.C. *Robot Motion Planning*, Kluwer Academic Publisher, Boston, MA, 1991.
- [5] Preparata, F.; Shamos, M. *Computational Geometry: an Introduction*, Springer Verlag, NY 1985.
- [6] Taranilla, M.T; Hernández Peñalver, G., *Descomposición en Sumas de Minkowski*, WICC 2003, Tandil, Buenos Aires, 2003.
- [7] Taranilla, M. T.; Leguizamón, M.G.; Gagliardi, E. O.; Hernández Peñalver, G.; *Algoritmos genéticos para la descomposición en Sumas de Minkowski*, Workshop de Investigadores en Ciencias de la Computación (WICC'2004), 2004