

Metaheurísticas para Space Planning

Giglia Gómez V.
Departamento de Informática
Universidad Técnica Federico Santa María
Avda. España 1680, Valparaíso, Chile
giglia@inf.utfsm.cl

Resumen

En el presente trabajo se da a conocer el diseño de un algoritmo híbrido que resuelve un problema de optimización de tipo Space Planning. El problema está relacionado con la categoría de problemas de corte y empaquetamiento, adicionando la importancia de la ubicación de objetos dentro de un espacio geométrico considerado como patrón de corte. La función objetivo consiste en minimizar el material desperdiciado en un patrón de corte.

Palabras Claves: Space Planning, Problemas de corte y empaquetamiento, patrón de corte, metaheurísticas, Operadores genéticos, algoritmo genético.

1 Introducción

Los problemas del tipo Space Planning (SP) son aquellos que contemplan ubicación de objetos en un espacio bidimensional o tridimensional donde la función objetivo incluye factores a evaluar, tales como distancia entre los objetos, grado de adyacencia y tamaño de la superficie[Eas70][Yoo92].

Los SP son problemas del tipo combinatorial. Conceptualmente pueden modelarse como un problema de satisfacción de restricciones (CSP) [ERR93], en donde se desea encontrar una solución que satisfaga las restricciones del problema, tales como no superposición, dimensión, orientación y forma. Algunos ejemplos que pertenecen a este tipo de problemas son: VLSI, problemas de corte de materiales, diseño de viviendas o decoración en la arquitectura y ubicación de etiquetas de texto.

Estas aplicaciones reales, se centran principalmente en la ubicación de sus objetos en el plano.

Por otro lado, los SP pueden modelarse como problemas de optimización con satisfacción de restricciones (CSOP)[VT95], cuya solución, además de satisfacer las restricciones, es óptima. Ésta característica se debe al análisis de valores de una función objetivo a minimizar o maximizar, dónde se busca su valor óptimo, en algún sentido, como por ejemplo la utilización mínima del espacio.

Recientemente se han utilizado técnicas de resolución para este tipo de problemas donde las técnicas clásicas no han sido efectivas. Estas técnicas se basan en metaheurísticas, dentro de las cuales se encuentran aquellas basadas en la teoría de la evolución, conocidas como algoritmos evolucionistas.

Dado que los SP tienen bastantes aplicaciones en la vida real, es que la motivación de este trabajo consiste en diseñar un algoritmo híbrido para resolver un problema de optimización del tipo SP usando ideas provenientes de la comunidad de restricciones y de la comunidad evolucionista.

El artículo se estructura de la siguiente manera. En la sección 2 se definen los problemas del tipo SP y sus características, junto con el modelo del problema. En la sección 3 se presenta la descripción del problema. En la sección 4 se muestra un algoritmo híbrido con su representación y función objetivo, además de la descripción de los operadores genéticos utilizados. En la sección 5 se presentan los experimentos y resultados obtenidos, para finalizar con las conclusiones.

2 Space Planning

En particular, los problemas de asignación de espacio físico para un conjunto de objetos, se conocen como Space Planning (SP)[Eas70]. Éstos existen en una amplia gama de áreas: Arquitectura, diseño textil, publicidad y diseño de tarjetas electrónicas.

Space Planning involucra una serie de tareas de diseño lo que incluye la ubicación de elementos en algún punto en un espacio definido limitado por restricciones de ubicación y área[Yoo92].

2.1 Restricciones del Problema de Space Planning

El problema SP posee restricciones de espacio y de forma que deben ser satisfechas.

- Restricciones topológicas: aquellas que corresponden a la distribución de los objetos en el espacio, donde éstos no se pueden intersectar pero si pueden ser adyacentes.
- Restricciones de distancias: También correspondiente a la distribución de los objetos en el espacio, pero visto desde la perspectiva de posición dentro del espacio físico, donde por ejemplo, dos objetos no pueden estar separados más de un metro.
- Restricciones de orientación: Correspondientes a las restricciones de forma, tomando en cuenta los lados que pueden lograr mayor porcentaje de adyacencia entre dos o más objetos como por ejemplo, objetos paralelos.
- Restricciones de forma: Correspondientes a las restricciones de forma en cuanto a dimensiones discretas que pueden afectar la forma (apariencia), por ejemplo, se necesitan objetos en forma rectangular y o circular[Yoo92][Cha95b].

El problema SP, tal como se mencionó anteriormente, es un problema que incluye restricciones geométricas impuestas por el espacio físico.

Así, un SP puede ser modelado como un CSP o un CSOP como lo muestra la figura 1. La figura a) muestra el problema desde la perspectiva de un CSP donde el objetivo es encontrar una ubicación de los objetos en el plano de forma tal que no estén superpuestos. La figura b) muestra el problema SP como un CSOP, donde además de la satisfacción de las restricciones se desea minimizar el área ocupada.

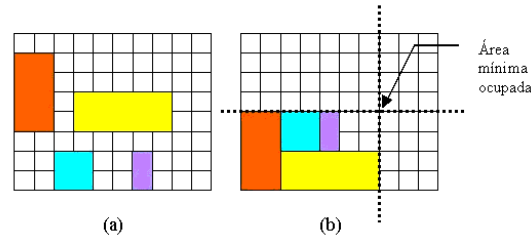


Figura 1: a) CSP y b) CSOP

Si se analiza un SP como un CSOP, además de satisfacer las restricciones de intersección, se busca la maximización de un beneficio o la reducción de un costo, como por ejemplo, lograr la mínima área ocupada por la ubicación de los objetos sin intersecciones.

3 Descripción del problema

El problema a estudiar en este trabajo es un problema categorizado como un problema SP donde el objetivo fundamental es lograr una buena ubicación de los objetos dentro de una superficie. Esta bondad se medirá de acuerdo a los beneficios que pueda aportar al problema, tales como la no intersección de los objetos además de la cercanía entre ellos, en términos de posición. Este último beneficio es el responsable del ahorro de espacio total ocupado.

La dificultad de lograr una buena ubicación de los objetos dentro de una superficie, radica en la necesidad de no intersección, es decir, en el mejor de los casos, los objetos deben quedar unidos por alguno de sus segmentos.

Este problema puede presentar varias aristas de desarrollo. Sus diferencias se basan en las características geométricas que posee. Los objetos contenidos en la superficie pueden tener diferentes formas como son rectangulares, triangulares, circulares o cuyos segmentos tengan estas diferentes formas en combinación. Además los objetos pueden rotarse. El ejemplo más claro es imaginar la ubicación de muebles dentro de una pieza, donde éstos se pueden rotar para poder encontrar una mejor ubicación dentro de la habitación.

El objetivo de este trabajo es proponer un algoritmo evolucionista híbrido que incorpore heurísticas para la resolución de problemas SP en su versión de optimización.

3.1 Problema SP

En este trabajo se tratará el problema en 2 dimensiones que formalmente se puede expresar como:

Definición 3.1 Problema SP

En un problema de dos dimensiones se tiene una superficie S de dimensiones $L \times W$ y se tiene un stock de objetos rectangulares con diferentes dimensiones, donde x_i es el i ésimo rectángulo cuyas dimensiones son $l_i \times w_i$, $i=1, \dots, n$.

Se desea determinar la ubicación de los objetos en S tal que la superficie libre sea máxima sin que ocurra superposición.

Los rectángulos tienen una orientación fija, es decir, un rectángulo de largo l_i y ancho w_i es diferente al rectángulo de largo w_i y ancho l_i con $l_i \neq w_i$. Además, cada rectángulo i , de largo l_i y ancho w_i tiene asignada una ubicación en el espacio S denotado por x_i e y_i que representa la esquina inferior izquierda, así ambos datos representan la superficie ocupada por el rectángulo i dentro de la superficie S . El espacio total ocupado por los rectángulos, corresponde a la superficie ocupada por: $((\max(x_i)) + w_i, (\max(y_i)) + l_i)$. La ubicación de todos los rectángulos sin superposición forman una configuración de ubicación S^ . La superficie de S^* tiene dimensiones L_{S^*} x W_{S^*} donde $W_{S^*} = \max(x_i)$ y $L_{S^*} = \max(y_i)$.*

La función objetivo consiste en encontrar una S^ donde una vez ubicados todos los rectángulos, ocupen la mínima área posible, es decir, $f(S^*) = \min(L_{S^*} \times W_{S^*})$.*

Además, en el problema a tratar se pueden observar las siguientes restricciones:

- Restricciones geométricas
- Restricciones Topológicas
- No se permiten intersección entre ningún par de objetos
- Si se permiten objetos contiguos adyacentes
- Restricciones de Orientación
- No se permite la rotación de objetos
- Restricción de forma
- Restricción de dimensión
- Sólo se contemplan objetos rectangulares
- La superficie total tiene forma rectangular

Hopper y Turton, [HT00], utilizaron distintas metaheurísticas para la resolución de un conjunto de problemas de tipo SP. Además de la técnica de resolución estudiada en esta investigación, utilizaron una heurística de ubicación para los objetos en el espacio. Esta heurística se llama BLF. Ésta fue elegida de un conjunto de heurísticas de ubicación ya que fue la que presentó mejores resultados respecto de su cercanía al óptimo.

BLF (Bottom Left Fill)[HT00], da prioridad a la ocupación de la esquina inferior izquierda menor que se encuentre en la superficie restante, esto quiere decir que el próximo objeto a ser ubicado ocupará la esquina inferior izquierda menor.

Este algoritmo ayuda a reducir el espacio desperdiciado entre los objetos, puesto que siempre los contempla para aquellos objetos que pueden caber en este espacio, independiente del orden en que se distribuyen dentro de la superficie.

La ventaja de este algoritmo de ubicación es que mantiene la referencia de las posiciones disponibles aunque éstas queden entre otros objetos rectangulares ya ubicados, esto hace que el desperdicio de material sea menor.

4 Propuesta de un Algoritmo Híbrido

La metaheurística escogida para diseñar el algoritmo híbrido es la de Algoritmos Genéticos (AG) la que pertenece a la familia de algoritmos evolucionistas. Esta metaheurística fue escogida por la flexibilidad que presenta en el manejo de información variada que puede ser explotada y explorada en forma eficiente con el objetivo de encontrar una solución cercana al óptimo. La utilización de AG permite analizar los cambios de la información a través de la evolución de generaciones de poblaciones de individuos, que para el caso en estudio, son configuración de ubicación de objetos, buscando así aquel que desperdicie el mínimo de material.

La naturaleza híbrida del algoritmo viene de una estructura de AG que utiliza una representación de la información no binaria, como técnica de resolución, unida a la heurística de ubicación BLF, comentada en la sección precedente.

Este algoritmo posee una estructura principal basada en los AG(s) propuestos por Holland (1975). La motivación de la utilización de los AG(s) como estructura básica es la capacidad adaptativa que presentan estos para la resolución de problemas de optimización. Más información sobre algoritmos genéticos se encuentra en [Bac91] [BBM93] [ERR93] [HJ93] [Mic95] [GGGLMT96].

Si bien la estructura básica del algoritmo híbrido, al igual que un genético, consiste de una población de individuos, la forma de representar la información hace la diferencia.

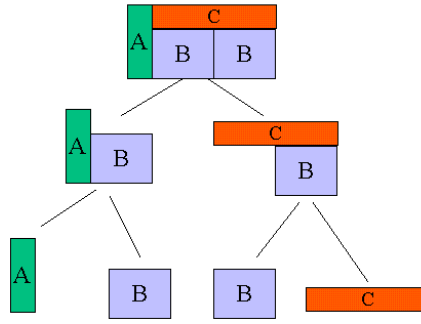


Figura 2: Árbol que representa un patrón de corte cuyos objetos se unen por adyacencias verticales y horizontales.

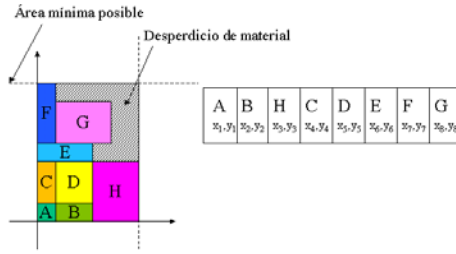


Figura 3: Muestra la representación de un string no binario para una configuración de ubicación de objetos.

4.1 Representación

Para este trabajo, a diferencia de un AG se eligió una representación de los datos no binaria. Trabajos hechos con representación de string no binarios han sido desarrollados por Hinterding en 1993 [HJ93], Michalewicz en 1995 [Mic95] y Volker Schneke en 1996 [Sch96] en forma independiente.

A continuación se presentarán dos posibilidades de representación para este estudio. La primera es una representación jerárquica llevada a un árbol binario y la segunda una representación secuencial dada por un string no binario.

4.1.1 Representación jerárquica

Hifi en 1997 [CHLeC99], utilizó una representación para la ubicación de objetos rectangulares dentro de una superficie S^* . Esta representación tiene la forma de un árbol, haciendo uso de una formación jerárquica de la configuración de ubicación, llamado por Hifi patrón de corte. Representaciones similares utilizaron Kado en 1995 y Volker Schneke en 1996 en problemas de ubicación¹. En la figura 2 se muestra el árbol que Hifi utiliza para representar un patrón de corte[CHLeC99][KRC95].

Kado en 1995, desarrolló un AG con representación de árbol binario para crear Layouts, sin embargo, hace un recorrido del árbol en postorden para luego almacenarlo en un string y utiliza éste como información genética de los individuos que forman la población y generaciones futuras.

Cada árbol puede representar distintas combinaciones de ubicación de objetos con adyacencias verticales y horizontales. El objetivo principal es obtener la configuración cuya superficie ocupada sea mínima, es decir, aquella configuración que posea el mínimo desperdicio de material posible.

El manejo de estructuras jerárquicas implica que el diseño de los operadores sea más complejo por el hecho que debe manipular directamente los objetos y sus ubicaciones. Esto da lugar a una alternativa adicional de diseño de operadores en algunos con algún grado de inteligencia para aumentar la probabilidad de éxito del algoritmo.

4.1.2 Representación lineal

Si bien la representación jerárquica puede reflejar la realidad del orden en que van siendo ubicados los objetos, puede ser una representación muy costosa en términos de recursos para el algoritmo que la utilice refiriéndose al tamaño del problema. Cuando el problema involucra la ubicación de una cantidad moderada de objetos, aproximadamente 20 ó 30, la representación no influye en el desempeño ni en la convergencia del algoritmo. Sin embargo la representación lineal hace un gran aporte en el manejo de la información para aquellos casos en que la cantidad de objetos a ubicar es superior a 30.

En la figura 3 se muestra un ejemplo de la representación lineal para la configuración.

¹Los problemas de ubicación (placement en inglés) son conocidos como problemas SP

La representación lineal, a pesar de su apariencia secuencial, puede reflejar la ubicación de los objetos en el plano aunque no necesariamente estén ubicados en línea. Esta se logra a través del uso de strings cuyo contenido no es necesariamente binario.

La ventaja de esta representación no tradicional, es que el proceso de codificación y decodificación que se necesita en el uso de strings binarios no existe. Un estudio donde se usa representación lineal no binaria para otro tipo de problemas, se puede encontrar en [HJ93].

A continuación se pasan a describir en detalle los componentes del algoritmo propuesto.

4.2 Población inicial

La población inicial está formada por un conjunto de individuos que usan la representación lineal.

La población inicial generada usa el conocimiento de las restricciones, con el objetivo que cada individuo represente una solución factible para el problema. Para el caso en estudio las configuraciones son generadas sin superposición de objetos rectangulares. La posición de los objetos se destina a partir de la esquina (0,0) de un plano bidimensional S siguiendo con el resto de ubicaciones dependiendo de las adyacencias que presenten los objetos dentro del plano.

4.3 Función de Evaluación

Sea S, la superficie total disponible y S^* , la superficie real ocupada por la configuración.

Como el problema en estudio es un problema de optimización, la función de evaluación se centra en S^* , es decir, el algoritmo busca el S^* que sea mínimo.

Definición 4.1 Función de Evaluación

Sea S^i una configuración de ubicación de rectángulos, cuya dimensión es L_{S^i} x W_{S^i} donde:

$$L_{S^*} = (y_{max}(S^i) - y_0(S^i));$$

$$W_{S^*} = (x_{max}(S^i) - x_0(S^i));$$

x_{max} : el punto x máximo de la configuración

y_{max} : el punto y máximo de la configuración

Entonces la función objetivo se define:

$$f_o = \min\{[max(L_{S^i}, W_{S^i})]^2\}$$

donde S^* es tal que $f(S^*) \leq f(S^i); \forall i = 1..no \text{ de configuraciones}$

La función de evaluación f_o toma en cuenta el área total ocupada por la configuración y el desperdicio de material. Esta función experimenta un refinamiento en los casos en que el valor de la función se repita entre individuos. Esto se puede dar cuando ambos cubren la misma área total y desperdician la misma cantidad de material. De éstos se desea privilegiar aquel que presente menor distribución del desperdicio de material, es decir, donde se encuentra la máxima área de desperdicio en términos de tamaño. Esta preferencia se expresa usando una subfunción que evalúa los puntos en la superficie de material desperdiciado ponderándolos sobre su misma ubicación en el plano, así se calcula la cantidad de puntos x e y disponibles que formen un rectángulo, lo que indica el tamaño rectangular del material desperdiciado. Por lo tanto se busca: $(y_{max_d} - y_{min_d})x(x_{max_d} - x_{min_d})$

donde:

- y_{max_d} : corresponde al punto máximo y de puntos disponibles que forman el rectángulo de material desperdiciado
- x_{max_d} : corresponde al punto máximo x de puntos disponibles que forman el rectángulo de material desperdiciado
- y_{min_d} : corresponde al punto mínimo y de puntos disponibles que forman el rectángulo de material desperdiciado
- x_{min_d} : corresponde al punto mínimo x de puntos disponibles que forman el rectángulo de material desperdiciado

La función de evaluación, entrega la calificación del individuo como candidato a solución respecto del problema. Esta no contempla la satisfacción de restricciones, ya que los individuos son manipulados por operadores genéticos, los que son los encargados de mantener la consistencia.

A continuación se describen los operadores genéticos ocupados para el algoritmo híbrido.

4.4 Operadores

El algoritmo propuesto utiliza sólo operadores aplicados sobre un individuo.

Los operadores utilizados son:

4.4.1 Selección

Respecto del operador de selección se utilizó el método de la ruleta[Mic95].

Otro operador de selección utilizado fue el elitismo, para poder traspasar de generación en generación, el mejor individuo encontrado.

4.4.2 Operadores Asexuales

Tomando en cuenta que cada individuo contiene la información de los objetos rectangulares que conforman S^* , se diseñaron 3 operadores encargados de cambiar la configuración, los que se detallan a continuación.

Cambio del Orden de Objetos

Este operador trabaja a nivel de objetos, cambiando el orden de todos los objetos lo que hace cambiar completamente la configuración. El objetivo de este operador es crear un individuo nuevo, en términos de configuración, respecto del que existía anteriormente.

Cambio de un Par de Objetos

Este operador trabaja a nivel de un par de objetos, cambiando el orden que presenta este par de objetos dentro de la configuración. El objetivo de este operador es crear un individuo nuevo cuya configuración sea similar al del individuo original, pero donde existe un par de objetos que están ubicados en posiciones diferentes a la original.

Cambio de Mitades

Este operador trabaja a nivel de mitades del individuo, cambiando una mitad por la otra en términos de orden. El objetivo de este operador es invertir las mitades del individuo convirtiéndolo en un nuevo individuo.

Respecto de la forma en que se aplican los operadores genéticos está dada por una probabilidad asignada a cada uno, con lo que se puede elegir una alta probabilidad de aplicación para aquellos operadores que aportan factores positivos en la evolución de los individuos para llegar a la solución.

Los problemas de tipo Space Planning manejan una gran cantidad de factores que hacen que el espacio de búsqueda se amplíe lo suficiente como para desorientar al algoritmo. Además, si se deja de manera libre tanto el ancho como el alto de la superficie, el problema se torna más complejo. Frente a estas dificultades es que existen algunas heurísticas de ubicación que limitan las posibilidades de ubicación de los objetos en la superficie S^* restringiendo el ancho o el alto de ésta.

Por estas razones se decidió incorporar al algoritmo la heurística BLF, utilizado en [HT00], para limitar el espacio de búsqueda.

Además, se agregó una heurística de búsqueda local, con el objetivo de lograr mejoras en los candidatos a solución. Este último componente agregado, podía en algunos casos dejar al algoritmo atrapado en un óptimo local, lo que se evitó con una probabilidad de aplicación análoga al factor de temperatura de Recocido Simulado. Esta probabilidad adaptiva va aceptando individuos nuevos a la nueva población, sin comprobar en realidad si estos son de mejor calidad que sus ancestros, a medida que el algoritmo evoluciona, la probabilidad de aceptación va bajando para sólo aceptar individuos mejores. Cuando el algoritmo presenta una latencia en términos de no poder encontrar individuos mejores a través de los operadores, la probabilidad de aceptación vuelve a subir, lo que produce la inclusión de individuos no tan buenos a la nueva generación, pero que a largo plazo pueden presentar mejoras significativas.

La figura 4, muestra la estructura del algoritmo híbrido con la aplicación de los operadores, la heurística de ubicación y el factor adaptivo de mejora local.

Descripción del Algoritmo

En el paso 1, factor es el porcentaje adaptivo que acepta individuos que no presentan mejoras respecto de sus ancestros una vez aplicado el operador.

La función `evalúa()` del paso 2, permite ordenar la población de individuos en base al valor ascendente de la función de evaluación.

Una vez ordenados los individuos en base a su función de evaluación, el operador de elitismo se aplica por medio de la función `mejor_individuo()` del paso 3, que pasa el mejor de los individuos de la población actual a la nueva población.

La función `evaluar()` del paso 4, califica al individuo en base a su función de evaluación. Para obtener el valor de la función de evaluación, se arma la configuración que representa el individuo aplicando la heurística de ubicación BLF.

Como se explicó anteriormente, los operadores genéticos, tiene una probabilidad de aplicación. Esta probabilidad está medida por la variable `prob` del paso 5, la que permite elegir el operador que se aplicará al individuo.

Una vez escogido y aplicado el operador al individuo, se crea un nuevo individuo llamado `otro`, como lo muestra el paso 7, el cual es evaluado mediante la función `evaluar()`. Así las funciones de evaluación entre el nuevo individuo y su ancestro son comparados y en el caso en que el nuevo individuo tenga mejor función de evaluación que su ancestro, entonces es incluido en la nueva población. Para el caso contrario, se elige un número aleatorio, paso 6, para verificar si se cumple la probabilidad de aceptación de individuos no tan bien calificados. Si se cumple la condición, paso 8, el `otro` individuo es incluido en la nueva población, además de disminuir la probabilidad de aceptación, en caso contrario, se mantiene el individuo original.

```

Inicio
  t ← 0
  factor ← 100 (1)
  inicia_población(P(t))
  evalúa(P(t)) (2)
  Repetir
    P(t+1) ← mejor_individuo(P(t)) (3)
    Repetir
      individuo ← seleccionar(P(t))
      i ← evaluar(individuo) (4)
      prob ← random() (5)
      si (prob ≤ prob. de mutación de objetos)
        otro ← mutacion de objetos(t)
      sino
        si (prob ≤ prob. de mutación de par)
          otro ← mutacion de par(t)
        sino
          otro ← mutacion de mitades(t)
      fin-si
    fin-si
    provfactor ← random() (6)
    si (evaluar(otro) ≤ i) (7)
      P(t+1) ← otro
    sino
      si (provfactor ≤ factor) (8)
        P(t+1) ← otro
        factor ← factor - (factor * 0.02)
      sino
        P(t+1) ← individuo
      fin-si
    fin-si
  Hasta (nueva población este llena)
  P(t) ← P(t+1)
  evalúa(P(t))
  si (mejor individuo no ha mejorado) (9)
    factor ← 100
  fin-si
  t ← t + 1
  Hasta (t = max. iteraciones)
Fin

```

Figura 4: Ilustración del algoritmo híbrido final

Una vez completa la nueva población, se convierte en la población actual, siendo evaluada y ordenada en base a la función de evaluación. Luego en el paso 9, se analiza si el mejor individuo se ha mantenido a través de las generaciones durante un período de tiempo determinado, si es así, el factor de aceptación de individuos no tan bien calificados aumenta para incluir en la población actual nuevos individuos que no son muy buenos pero que pueden presentar notables mejoras a largo plazo.

5 Resultados

De acuerdo a los datos de prueba mostrados en [HT00], se seleccionó dos categorías de problemas, de un total de 7, para comparar los resultados. El primer problema corresponde a la categoría C1 que involucra 16 objetos rectangulares con un ancho fijo de 20. El segundo problema corresponde a la categoría C5 que involucra 73 objetos. Para ambos problema se conoce el tamaño de la superficie óptimo.

El estudio publicado en [HT00], muestra diferentes metodologías de resolución para el conjunto de problemas cuyos resultados presentaron soluciones cercanas al óptimo. En ninguno de los problema se obtuvo el óptimo, lo cual se consideró un buen conjunto de problemas para ser testeados por la propuesta hecha por este artículo.

El algoritmo híbrido propuesto en este estudio utilizó la representación lineal puesto que se evaluó ambas representaciones descritas en la sección 4 y se obtuvo mejores resultados usando la representación lineal.

Para el segundo problema que involucra 73 objetos rectangulares cuyo ancho fijo es de 60 y alto óptimo 90, el estudio hecho en [HT00] no encontró un óptimo sino uno cercano, al igual que el algoritmo propuesto por este trabajo.

Respecto de la aplicación de los operadores, se trató de encontrar un afinamiento entre las probabilidades de aplicación y un factor de adaptación (cuya funcionalidad se basa en mejoras locales [Mic95][GGGLMT96]), aplicado a los operadores. Desde este punto de vista los resultados se clasifican de acuerdo a 2 categorías. La primera categoría consiste en el afinamiento de los parámetros con el objetivo de averiguar cuál es el conjunto de probabilidades de ocurrencia de operadores óptimos para el problema. La segunda categoría consiste en la aplicación de las probabilidades escogidas por el primer resultado en ejecuciones prolongadas para analizar finalmente cuáles fueron los mejores individuos encontrados.

Todas las pruebas del algoritmo se hicieron en un PC pentium III de 700 mhz. y 64MB RAM.

5.1 Resultados para 16 objetos

El algoritmo con representación lineal logró llegar al óptimo de este problema, cuya dimensión es de 20 x 20, lo que no se encontró en la investigación hecha por Hopper y Turton.

A continuación la tabla 1 muestra resultados seleccionados de varias ejecuciones del algoritmo. Es importante indicar que en todas las ejecuciones se encontró el óptimo.

F. Evaluación	Generación	Tiempo de CPU (seg)
Óptimo	7	10
Óptimo	1	1
Óptimo	18	6
Óptimo	25	9
Óptimo	13	12

Tabla 1: Resultados con 16 objetos

La probabilidad de aplicación de los operadores se distribuyó básicamente en probabilidades similares, las cuales produjeron óptimos resultados. Estas probabilidades son: Cambio de objetos 30%, Cambio de un par de objetos 40% y Cambio de mitades 30%.

En el gráfico 5, se pueden observar los tiempos de los óptimos encontrados en las ejecuciones.

5.2 Resultados para 73 objetos

El algoritmo con representación lineal no logró llegar al óptimo de este problema, cuya dimensión es de 90 x 60, lo que tampoco se logró en la investigación hecha por Hopper y Turton.

A pesar de no encontrar el óptimo, las distintas combinaciones de probabilidades de aplicación para los operadores fueron presentando cambios en la calidad de los individuos encontrados por el algoritmo, lo que llevó a un análisis más profundo respecto del afinamiento de estas probabilidades para poder encontrar una solución cercana al óptimo.

5.2.1 Afinamiento de probabilidades

El afinamiento de probabilidades de operadores apunta directamente a la pronta convergencia del algoritmo. Como se trata de un algoritmo para resolver un problema de optimización, la convergencia consiste en descubrir en qué generación

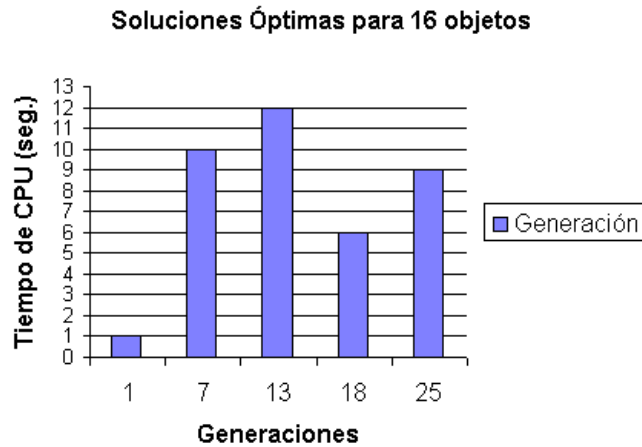


Figura 5: Soluciones Óptimas encontradas para 16 objetos.

encontró el patrón con mínimo desperdicio después de una cantidad fija de iteraciones. Para probar la efectividad de los operadores en la convergencia del algoritmo, estos resultados se tomaron bajo la misma semilla del generador de números aleatorios.

Como para el problema de 16 objetos se encontró el óptimo, el análisis siguiente se centró con el problema más difícil, el que contiene 73 objetos. La prueba consistió en ocupar distintas combinaciones de probabilidades de los operadores, con el objetivo de encontrar el mejor afinamiento.

En la tabla 2 se muestran los mejores casos encontrados de las 10 ejecuciones por combinación de probabilidades. FO corresponde a la mejor función de evaluación encontrada después de las iteraciones y G corresponde a la generación en la que se encontró.

Luego de varias pruebas de probabilidades de operadores de cambio de objetos (PCO), cambio de un par de objetos (PCP) y cambio de mitades (PCM) los resultados indicaron claramente la eficiencia del operador cambio de un par de objetos por sobre los otros.

El factor de adaptación aplicado a los operadores es de un 2% cada vez. El factor de adaptación inicialmente acepta movidas que pueden empeorar o mejorar el individuo, sin embargo, a medida que evoluciona, el factor de adaptación disminuye su probabilidad de aceptación de movidas erróneas en un 2% cada vez. Si luego de un cierto tiempo, el algoritmo presenta una monotonía con respecto al mejor individuo conservado a través de las generaciones, aplicando elitismo, el factor de adaptación vuelve a su estado inicial y comienza a disminuir a medida que va evolucionando.

PCO	PCP	PCM	FO	G
33	33	33	5940	79
100	0	0	5940	91
0	100	0	5880	8
0	0	100	6060	16
30	55	15	5880	78
30	40	30	5940	38
25	70	5	5880	60

Tabla 2: Tabla de afinamiento de probabilidades

Para seguir analizando el impacto de los operadores en las generaciones futuras, se estudió la aplicación de la mejor combinación de probabilidades obtenidas por el resultado anterior (tabla 2) en ejecuciones duraderas. Estas ejecuciones consistieron en pruebas con 500, 1000 y 2000 iteraciones, sólo para verificar si la eficiencia de los operadores tiene efectos durante la evolución.

La Tabla 3 muestra los mejores casos encontrados aplicando la mejor combinación de probabilidades en 500, 1000 y 2000 iteraciones, donde se puede apreciar que mientras se aumenta la cantidad de generaciones, se obtuvo una mejora en los individuos.

El resultado más cercano al óptimo de estas ejecuciones fue de un alto de 96, tomando en cuenta que el cercano al óptimo encontrado en [HT00] fue de 94 y el óptimo real es de 90.

Iteraciones	FO	G
500	5880	350
1000	5760	690
2000	5820	1294

Tabla 3: Pruebas con 500, 100 y 200 iteraciones

6 Conclusiones

Se ha logrado crear un algoritmo híbrido capaz de resolver un problema de tipo SP. Se definieron tres operadores y una función objetivo para este tipo de problema. La función objetivo busca aquellos patrones de corte con mínimo desperdicio y lo más simétricos posible.

La estructura de información no tradicional escogida representa completamente la naturaleza de los problemas de este tipo, además de facilitar el manejo de la información para los operadores. Respecto de las alternativas presentadas en este artículo para la representación de los individuos, aconteció que para aquellos problemas que contienen gran cantidad de elementos, la estructura jerárquica utilizada en otros estudios, presentó un aletargamiento en la convergencia del algoritmo debido a la utilización de muchos recursos del sistema, lo que llevó finalmente a la decisión de la utilización de una representación lineal no binaria.

De acuerdo a los resultados mostrados por las tablas, el operador más importante en cuanto a su efectividad es el de mutación de un par de objetos, pues explota la información ya contenida en el individuo y produce en su mayoría mejoras de éste. Esto sucede porque individuos que presentan una buena función de evaluación, con un pequeño cambio, mejoran aún más su valor.

Es importante destacar que la naturaleza de este tipo de problemas hace que un caso sea completamente distinto a otro, esto implica que el afinamiento de las probabilidades está ligada estrechamente a las características del problema, como lo son, cantidad de objetos, tamaño de los objetos, ancho o alto fijo. por lo cual se sugiere incorporar el control de parámetros.

Respecto de la aplicación del factor de mejora local sobre los operadores, se quiso simular la aplicación del factor de temperatura en la metaheurística Simulated Annealing, lo que aportó beneficios a la pronta convergencia del algoritmo.

Esta investigación presenta un algoritmo híbrido como una alternativa más de solución a problemas de tipo Space Planning. En el trabajo futuro se analizará la mejora del rendimiento del algoritmo frente a problemas más complejos, incluyendo un grado de inteligencia al algoritmo, que sea capaz de adaptar las probabilidades de aplicaciones de operadores y del factor de adaptación y buscar distintas alternativas como función de evaluación con el objetivo de relacionar el valor de esta función directamente con el desperdicio de material y la distribución de éste.

Referencias

- [Bac91] Thomas Bäck. *Optimization by Means of Genetic Algorithms*. University of Dortmund. Department of Computer Science. Germany 91.
- [BBM93] Beasley D., Bull D., Martin R.. *An overview of genetics Algorithms: Part II*. Research Topics. Inter – University Committee on Computing, 1993.
- [CHLeC99] Van – Dat Cung, Mhand Hifi, Bertrand Le Cun. *Constraint two dimensional cutting stock problems a best first branch – and – bound algorithm*. Université de Versailles – Saint Quenrin en Yvelines & Cermsem, Maison des Sciences Economiques, Université de Paris 1. France 1999.
- [Cha95b] Philippe Charman. *Gestion des Contraintes geometriques pour l'aide à l'aménagement spacial*. Thèse – Université de l'Ecole Nationale des Ponts et chaussées. Paris 1995.
- [Eas70] Charles M. Eastman. *Representations for Space Planning*. Communications of the ACM. Vol 13 No 4, April 1970.
- [ERR93] A. E. Eiben, P – E. Raué, Zs. Ruttkay. *Heuristic Genetic Algorithms for Constrained Problems*. Artificial Intelligence group. Department of Mathematics and Computer Science. Vrije Universiteit Amsterdam. Amsterdam 1993.
- [GGGLMT96] Fred Glover, Hassan M. Ghazin, J. L. González, Manuel Laguna, Pablo Moscato, Fan T. Tseng. *Optimization, Heurística y Redes Neuronales*. Editorial Paraninfo, 1996.
- [HJ93] Robert Hinterding, Kate Juliff. *A genetic Algorithm for stock cutting: An exploration of mapping Schemes*. Department of Computer and Mathematical Sciences. Technical Report 24 COMP3 February 1993.

- [HT00] E. Hopper, B. C. H. Turton. *An empirical Investigation of Metaheuristic Algorithms for a 2D Packing problem*. European Journal of Operational Research 128/1, 34-57, 2000.
- [KRC95] Kazuhiro Kado, Peter Ross, Davis Corne. *A study of Genetic Algorithm Hybrids for Facility Layout Problems*. Department of Artificial Intelligence. University of Edinburgh. UK 1995.
- [Mic95] Zbigniew Michalewicz. *Genetic Algorithms + Data structures = Evolutions Programs*. Ed Springer 1995.
- [Sch96] Volker Scnecke. *Tesis: Hybrid Genetic Algorithm for solving constrained packing and placement problem*. PhD dissertation. Department Math / Comput. Sci. Univ. Osnabrück. Germany 1996.
- [VT95] Christos Voudoris and Edward Tsang. *Partial Constraint Satisfaction Problems and Guided Local Search*. Technical Report CSM-250. Department of Computer Science. University of Essex. Essex 1995.
- [Yoo92] K. B. Yoon. *A constraint Model of Space Planning*. Topics in Engineering Vol 9. Computational Mechanics Publications. Southampton UK and Boston Usa.1992.