

Metaheurísticos e Hibridación aplicados al problema del diseño de plantillas (TDP)

Metaheuristics and Hybridization applied to the problem of template design (TDP)

David Rodríguez-Rueda^{a*}

^{a*} Estudiante de Doctorado en Tecnologías Informáticas, drodri@unet.edu.ve ,0000-0002-4478-4949, Universidad Nacional Experimental del Táchira, San Cristóbal, Venezuela

Forma de citar: Rodríguez-Rueda, D. Metaheurísticos e Hibridación aplicados al problema del diseño de plantillas (TDP). *Eco Matemático*, 10 (1), 57-70

Recibido: 9 septiembre 2018
Aceptado: 01 diciembre 2018

Palabras clave

Búsqueda local,
Diseño de plantillas,
Metaheurísticas,
Algoritmo Genético,
Algoritmo Memético,
Técnicas híbridas.

Resumen:El problema de diseño de plantillas (TDP) es un problema combinatorio difícil de atacar y que presenta un alto número de simetrías cuya existencia agrega complejidad a su resolución. Se han propuesto varias técnicas en la literatura para optimizar su resolución, que abarca desde métodos completos hasta métodos estocásticos. En este artículo estamos proponiendo el uso de técnicas híbridas. Nos proponemos utilizar un enfoque integrador híbrido que combine técnicas de búsqueda local basadas en vecindario y basados en población para obtener lo que se denomina un algoritmo Memético. Utilizaremos búsquedas locales basadas en vecindario como operadores de mejora local, que nos permitan explotar zonas de búsquedas que resulten prometedoras. El uso de un algoritmo Genético contribuirá como mecanismo para la realizar la exploración del espacio de búsqueda. Un análisis empírico que compara el rendimiento de todos los métodos (es decir, algoritmos básicos o híbridos) en la optimización del problema, nos permite comparar la propuesta descrita con los resultados obtenidos en la literatura científica existente.

*Autor para correspondencia drodri@unet.edu.ve

<http://dx.doi.org/10.1016/j.eq/>

2590-9215© 2017 Universidad Francisco de Paula Santander. Este es un artículo bajo la licencia CCBYNCND

Keywords

Local search,
Template design,
Metaheuristics,
Genetic algorithm,
Memetic algorithm,
Hybrid techniques.

Abstract: The template design problem (TDP) is a combinatorial problem that is difficult to attack and that presents a high number of symmetries that adds complexity to solution. Several techniques have been proposed in the literature to optimize your solution, ranging from complete methods to stochastic methods. In this article we are proposing the use of hybrid techniques. We intend to use a hybrid integrative approach that combines neighborhood-based and population-based local search techniques set up a technique called Memetic algorithm. We will use local neighborhood-based searches as local improvement operators, which allow us to exploit search areas that are potential. The use of a genetic algorithm will contribute as a mechanism to perform the search space exploration. An empirical analysis that compares the performance of all methods (that is, basic or hybrid algorithms) in the optimization of the problem, allows us to compare the proposal described with the results obtained in the existing scientific literature.

Introducción

Existen muchos esfuerzos, en el ámbito de la manufactura, para afrontar problemas relacionados con el desperdicio de materia prima, que aparecen durante el proceso de producción de producto terminado, (Kasemset y colaboradores, 2015; Wang y colaboradores, 2016). Para abordar con éxito este tipo de desafíos se precisa dedicar un tiempo y esfuerzo prudente en el análisis de estos escenarios, todo con el fin de crear modelos que permitan maximizar la producción con un mínimo de desperdicio. El problema del diseño de plantillas (TDP: Template Designs Problem) es un claro ejemplo donde podemos encontrar este tipo de desafíos. El TDP se origina en la industria, cuando se requiere de la elaboración e impresión de los embalajes de cierto tipo de productos, en los cuales la forma es igual, pero existe algún tipo de variación en su presentación. La literatura muestra algunas propuestas para la resolución de este problema, incluyendo técnicas de programación con restricciones y programación matemática. En este trabajo se aborda el problema del diseño de plantillas que está enmarcado dentro de la categoría de problemas de optimización combinatoria NP-Duros que ha surgido en la industria. Este problema representa un reto que ya ha sido atacado por un conjunto de técnicas y métodos diferentes, incluyendo

los métodos exactos que van desde la programación lineal entera (ILP) hasta técnicas metaheurísticas. Proll y Smith (1998) presentaron por primera vez un modelo lineal entero para la resolución de este problema. La propuesta utiliza dos estrategias: (i) programación lineal entera y (ii) programación con restricciones. Los autores concluyen que el abordaje de este problema resulta muy difícil y que además de las estrategias utilizadas se requiere de gran ingenio para su resolución. Posteriormente, Flener y colaboradores (2001), presentan un modelo matricial para abordar este problema. Otro aporte fue hecho por Prestwich y colaboradores (2006), que propusieron un modelo para la resolución del TDP basado en un mecanismo de demanda incierta. Esta formulación se basa en un modelo ILP y la utilización de un algoritmo de búsqueda local, basada en SAT (Boolean satisfiability).

En Rodríguez y colaboradores (2010), se ha propuesto el empleo de dos técnicas básicas para resolver el problema de diseño de plantillas: Un esquema simple de búsqueda local basado en ascenso a la colina (Hill Climbing), junto con una técnica más robusta basada en búsqueda Tabú (Tabu Search). Estas dos estrategias mostraron un rendimiento aceptable para atacar este problema. Más recientemente, en Rodríguez y colaboradores

(2011), una noble propuesta ha sido descrita, que incorpora un modelo incremental, para ello se adiciona un algoritmo genético junto con un mecanismo incremental donde cada una de las técnicas opera en forma sinérgica, tomando como punto de partida las soluciones ejecutadas en una etapa previa. Los resultados obtenidos son bastante alentadores.

En esta investigación continuamos en la búsqueda de una estrategia computacional orientada en la línea de la utilización de un modelo híbrido más robusto. La meta de esta investigación es por lo tanto la combinación sinérgica de un algoritmo genético básico y la incorporación de una técnica que hace uso de búsquedas locales, como mecanismo de mejora para los operadores genéticos del GA, lo cual nos deberá llevar a diversificar las regiones de búsqueda, permitiéndonos explorar y explotar con mayor fuerza las soluciones candidatas que potencialmente pueden llegar a ser un óptimo global. Además, estamos considerando la utilización un grupo de diversos escenarios que incorporan configuraciones de mayor complejidad, con el fin de verificar el desenvolvimiento de nuestra propuesta cooperativa.

Materiales y métodos

En este trabajo investigativo estamos presentando formas alternativas de abordar el problema del diseño de plantillas. Por lo cual, ofrecemos dos contribuciones principales en el presente trabajo: (i) algoritmos integrativos de dos niveles (i.e, metaheurísticos e hibridación), (ii) métodos de optimización específicos para el TDP que proporcionan resultados de vanguardia cuando se aplican a instancias de problemas específicos. Esta es una prueba de la efectividad de nuestra propuesta. Para ello hemos utilizado la idea propuesta en el modelo descrito en Proll y Smith (1998), así como la utilización de nuevos escenarios de alta complejidad, para demostrar empíricamente que las técnicas metaheurísticas e híbridadas registran un desempeño eficaz para resolver instancias del

problema de gran tamaño. Más concretamente, consideramos Hill Climbing (HC), búsqueda Tabú (TS), Algoritmos Genéticos (GA) y Meméticos.

Formulación

El TDP contemplado aquí se abordó por primera vez por Proll y Smith (1998) a petición de una empresa que requería producir, utilizando el mínimo número de plantillas, una variedad de embalajes de diversas presentaciones, pero manteniendo la misma forma. Esta variedad de productos abarca el empaquetado de comida para el ser humano, animales domésticos y caratulas para revistas. Aquí tratamos variaciones pequeñas, por ejemplo, diferentes embalajes (i.e., cartones) para comidas de gatos solo difieren en el color de fondo y en la etiqueta impresa, manteniendo su forma. El problema se describe como sigue: Dado un conjunto de variaciones de un determinado diseño, con igual forma y tamaño, de los cuales se conoce la demanda de cada variación, se requiere diseñar una o más plantillas de igual capacidad para satisfacer la demanda solicitada para cada variación. El diseño debe ser capaz de minimizar el número total de impresiones o troquelados de cada plantilla de tal forma que satisfaga la demanda requerida en cada variación (Prestwich y colaboradores, 2006).

Considérese el ejemplo mostrado por Proll y Smith (1998) para imprimir cartones (i.e., empaques) para diferentes variedades de comida de gatos según la demanda mostrada en el Tabla 1.

Tabla 1: Demanda por producto y variación.

Variación	Cantidad (× 1000)
Hígado	250
Conejo	255
Atún	260
Pollo Doble	500
Sardina Doble	500
Pollo	800
Sardina	1100
Total	3665

Todos los diseños de cartones tienen el mismo tamaño y la misma forma, y sólo difieren en su impresión. Si consideramos en este ejemplo que cada plantilla consta de 9 espacios o slots, tenemos pues 9 slots y 7 variaciones de empaques para los productos. Una forma simple, aunque costosa, de resolver esta situación consiste en utilizar siete plantillas, una para cada variación, pero esto incrementaría los costos de producción y los desperdicios de material terminado. Otra posible solución podría considerar usar una única plantilla con un slot para cada variación y los otros dos slots restantes dedicarlos a las variaciones con las demandas más altas (pollo y sardina), y realizar 550000 impresiones de la plantilla.

Diseño de Plantillas: Modelo ILP

Nuestra propuesta se basa en el modelo mostrado por Proll y Smith (1998). Existen cinco elementos necesarios para la definición de este problema, los cuales corresponden con: las variaciones de cada producto (V), el número de plantillas que se desean utilizar (T), el número de celdas (*slot*) disponibles por cada plantilla (S), la demanda del producto, requerida por cada variación (Q) y el número de troquelaciones a ser aplicadas a cada plantilla (R). Con base a esto podemos formalizar el TDP de la siguiente forma: dado un escenario del problema, éste estará representado por $R=(M, Q)$, donde M representa un matriz decimal que contiene la configuración de una solución candidata, Q es un arreglo que contiene la demanda de cada una de las variaciones existentes y R el número total troquelaciones por plantilla. De acuerdo al modelo matricial que se emplee, (M) podrá tener una de las siguientes dimensiones:

- $V \times T$, si se emplea un modelo basado en los diseños disponibles (V).
- $S \times T$, si se emplea un modelo basado en las celdas disponibles (slots, S).

Cada plantilla se configurará con un conjunto de celdas ($s_{1i} \dots s_{di}$), donde i indica el número de celdas asignadas a un determinado diseño d . Para cada plantilla se debe cumplir que $\sum_{i=1}^d (s_i) = S$ y a su vez que $\sum_{j=1}^T (s_j) > 0$, en cualquier otro caso la configuración será errónea. Cada configuración, la cual es considerada una solución candidata, debe ser entregada a un resolutor ILP, el cual nos devolverá el número de troquelaciones (R), necesarias a ser aplicadas a cada plantilla para alcanzar la demanda (Q).

En términos generales, el problema lo planteamos como la minimización de la pérdida de material:

Tal que

$$\varphi = \min \sum (U_i + O_i) \quad (1)$$

$$\sum (p_{ij}R_j + U_i - O_i) = Q_i \quad (2)$$

$$\sum p_{ij}R_j \geq (1 - l_i)Q_i \quad (3)$$

$$\sum p_{ij}R_j \leq (1 + u_i)Q_i \quad (4)$$

$$R_j, U_i, O_i \geq 0 \quad (5)$$

donde:

- U_i : infraproducción de la variación i ,
- O_i : sobreproducción de la variación i ,
- R_j : impresiones de la plantilla j ,
- Q_i : demanda de la variación i ,
- l_i : cantidad de producción, tolerancia, inferior a la demanda de la variación i ,
- u_i : cantidad de producción, tolerancia, superior a la demanda de la variación i ,
- p_{ij} : slots en la plantilla j en la cual la variación i aparece.

Las tolerancias l_i, u_i permiten ajustar la producción ligeramente por encima o por debajo del objetivo marcado, lo que nos lleva a considerar como factibles aquellas posibles soluciones que se encuentren en el rango: $(1-l_i)Q_i \leq \varphi \leq (1+u_i)Q_i$.

Inicialización de Candidatos

La inicialización de los candidatos, se realizará de acuerdo a la demanda por variación, el cual consiste en intentar asignar a cada diseño un número de slots que depende de la demanda de este. El pseudo código correspondiente lo podemos ver en el Algoritmo 1.

Algorithm 1: Pseudo code of initialization under demand (ID)

```

1 begin
2    $p \leftarrow \text{INITIALIZE\_TEMPLATE}(T, V)$ ;
   // Computes the fraction of the demand for each variation.
3    $pq \leftarrow \text{COMPUTE\_PROBABILITIES}(Q)$ ;
4   for  $i \leftarrow 1$  to  $T$  do
   // Fills the  $i$ -th template according to  $pq$ .
5      $\text{ROULETTE\_FILL}(p, pq, i, V)$ ;
   // Computes a heuristic number of pressings for the  $i$ -th
   // template and updates probabilities according to the
   // remaining demand.
6      $R \leftarrow \min_{j=1..V} \{(1-l_j)Q_j / p_{ij} \mid p_{ij} > 0\}$ ;
7     for  $j \leftarrow 1$  to  $V$  do
8        $Q_j \leftarrow Q_j - R * p_{ij}$ ;
9     end for
10     $pq \leftarrow \text{COMPUTE\_PROBABILITIES}(Q)$ ;
11  end for
12 end

```

Un modelo basado en Diseños

La representación que hemos empleado está relacionada directamente con el número de diseños de cada escenario, así un eventual candidato será una matriz $M_{V \times T}$ que contendrá en sus celdas el número de slot que ocupara cada diseño en una determinada plantilla, con lo cual las filas de la matriz representan los diseños y las columnas las plantillas, un ejemplo para el escenario mostrado en la Tabla 1, donde se ha considerado utilizar 2 plantillas se muestra en la Figura 1.

Diseños	Plantillas	
	P1	P2
D1	1	0
D2	1	0
D3	1	0
D4	2	0
D5	2	0
D6	2	2
D7	0	7

No. Diseños = 7
 No. Slots = 9
 No. Plantillas = 2

Suma 9 Slots

Figura 1: Ejemplo de Modelo por Diseños.

El vecindario

La evaluación del vecindario para esta propuesta se seleccionó de acuerdo a la posibilidad de poder modificar la solución candidata en solo dos posiciones diferentes de una determinada plantilla, es decir, se recorre la matriz correspondiente por cada plantilla y se escoge de manera progresiva dos tipos diferentes de variaciones en cuyo diseño sea posible incrementar y/o decrementar en una unidad el número de slot asignados. Esto será posible siempre que para el caso del incremento se cumpla que no se exceda el número de slot disponibles y para el caso de la resta esta sea mayor que cero. Otra importante consideración que hay que tener presente es la posibilidad de que cada solución candidata se obtenga de un diseño válido, lo que significa que hay que cuidarse de no enviar al solucionador LPSolve un candidato no-factible de ser resuelto. Una forma de evitar que esto ocurra es asegurándose de que cada plantillas contengan V variaciones y que el total para cada variación sea al menos la unidad, con esta restricción estaremos seguros que el solucionador nos dará un valor posiblemente factible.

de la mochila (Kulkarni y Shabir, 2016, Feng y colaboradores, 2017), con valores que nos permiten alcanzar la demanda total. Hay que aclarar que en la mayoría de los casos se debió ajustar el valor de la demanda total (ligeramente por encima del valor de referencia), ya que la sumatoria no coincidía exactamente con el valor de referencia.

Porcentaje de vecinos a evaluar

El número de vecinos a ser evaluado en cada escenario es del orden de:

$$n=T.V.(V-1)$$

Esto nos indica que el número de vecinos por escenario excede los 550 individuos, por esta razón nos planteamos evaluar sólo un porcentaje de estos. Esta muestra porcentual la hemos considerado por debajo de los 550 individuos para todos los escenarios por igual. Esto se debe a que se han lanzado un conjunto de pruebas sobre las instancias más complejas y se descartó la posibilidad de realizar una exploración exhaustiva del vecindario. La Tabla 3 muestra los porcentajes seleccionados de acuerdo al escenario.

Tabla 3: Muestras porcentuales de individuos a ser evaluados por cada escenario.

Escenario	Muestra % Inferior (A%)	Muestra % Superior (B%)	No. Máximo de Individuos
Shoes Labels	10	15	171
Pencils Boxes	10	15	248
Gif Boxes	5	10	468
Shirt Labels	2,5	5	531
CD Cover	2,5	5	490
Book Cover	2,5	5	310

Resultados Experimentales

Realizamos 20 ejecuciones por cada instancia del problema. El número de evaluaciones para cada instancia está sujeto al número de variaciones y número de plantillas, según el escenario abordado: $n_v=1000.T.V.(V-1)\%_v$, donde $\%_v$ representa la muestra porcentual de vecinos a evaluar pues una evaluación completa es tremendamente costosa. Las

pruebas realizadas apuntan hacia la consideración de un número de vecinos no mayor a 550, para el escenario más complejo.

Algoritmos HC, TS y GA ejecutados por separado

Al realizar la ejecución en forma aislada de cada uno de los algoritmos empleados hemos revisado dos aspectos fundamentales:

- El porcentaje de soluciones factibles por cada escenario.
- La mejor solución encontrada.

Porcentaje de soluciones factibles:

Se ha podido observar, en el caso de los metaheurísticos, que la búsqueda Tabú presenta un mejor desempeño frente a sus similares (HC y GA). Las Figuras 3 y 4, presenta los porcentajes de soluciones factibles para la muestra porcentual inferior/superior(A%,B%). Podemos observar como los escenarios Gif Boxes y Shirt Labels presentan un perfil que parece ser de gran complejidad.

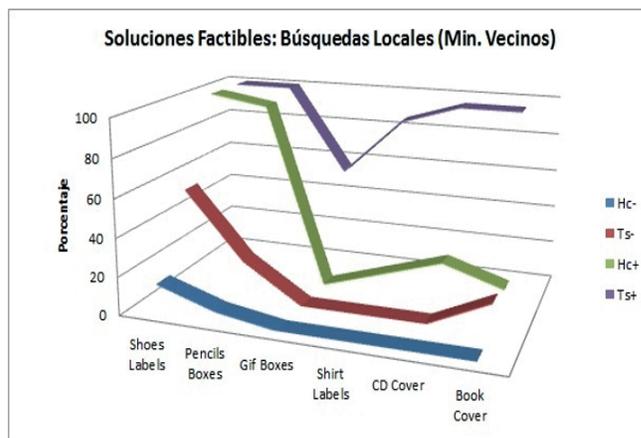


Figura 3: Soluciones factibles para el porcentaje A de vecinos. Hc-, Ts-: corresponden al número mínimo de plantillas, y Hc+, Ts+ al número máximo.

Para el caso del Algoritmo Genético, alcanza soluciones factibles solo para el escenario Shoes Labels, por lo cual no estamos presentando el gráfico correspondiente.

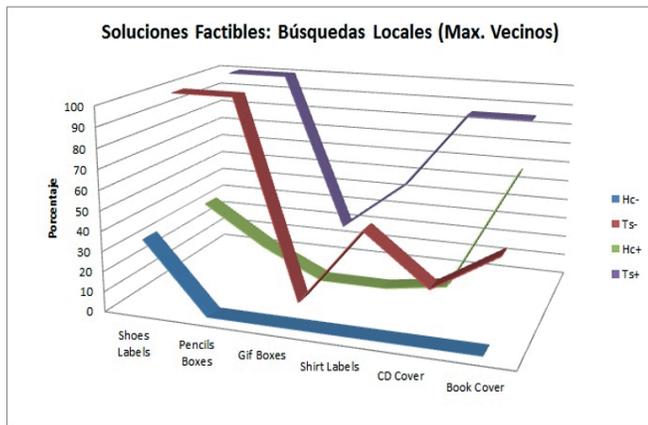


Figura 4: Soluciones factibles para el porcentaje B de vecinos. Hc-, Ts-: corresponden al número mínimo de plantillas, y Hc+, Ts+ al número máximo.

Los algoritmos meméticos tienen un comportamiento muy similar, en lo referente al porcentaje de soluciones factibles, al operar sobre las configuraciones diseñadas de acuerdo a la aplicación el operador de mejora local (i.e. Hc o Ts), véase las Figuras 5 y 6. La nomenclatura Ma_xx (donde xx sera: **Hc**: Hill Climbing, o podría ser **Ts**: Tabu Search), representa el algoritmo Memético que hace uso del operador de mejora local correspondiente a la búsqueda local *Hc* o *Ts*.

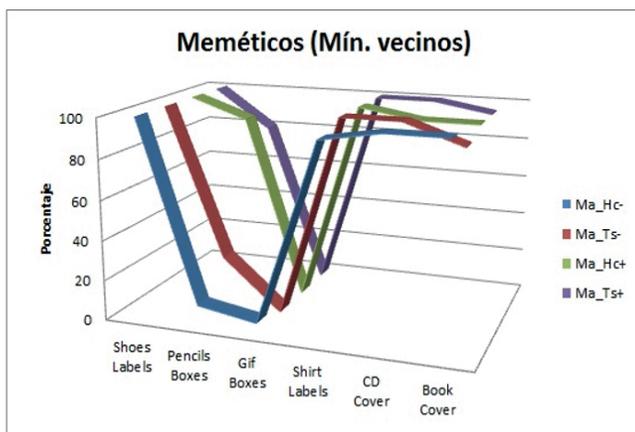


Figura 5: Soluciones factibles para el porcentaje A de vecinos. Ma_Hc-, Ma_Ts-: corresponden al número mínimo de plantillas, y Ma_Hc+, Ma_Ts+ al número máximo.

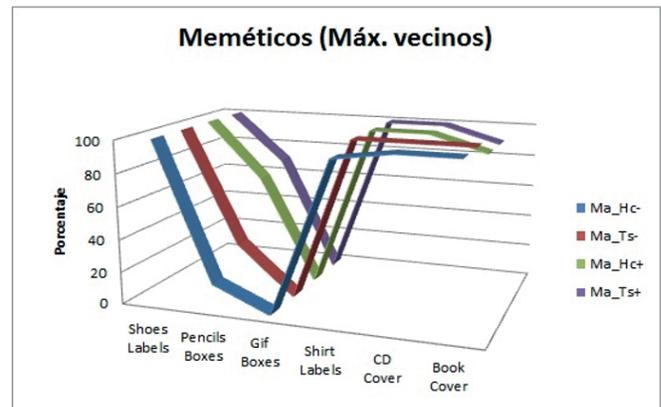


Figura 6: Soluciones factibles para el porcentaje A de vecinos. Ma_Hc-, Ma_Ts-: corresponden al número mínimo de plantillas, y Ma_Hc+, Ma_Ts+ al número máximo.

Una revisión muy rápida nos permite verificar que el escenario Gif Boxes se muestra como un gran desafío ya que la batería de algoritmos de búsquedas locales aplicados no ha mostrado resultados muy sorprendentes.

Mejor solución:

La mejor solución es considerada como aquella que obtiene la mejor configuración en cuanto a diseños por slot que permiten minimizar el número de troquelaciones por cada escenario evaluado, resultando en la minimización del desperdicio de materia prima. Las tablas 4 y 5, presenta la mejor solución encontrada (en cuanto a minimizar el desperdicio de materia prima) alcanzadas por los diferentes algoritmos metaheurísticos utilizados en cada uno de los escenarios.

Tabla 4: Resultados obtenidos para la metaheurística HC, por cada escenario abordado.

Problema	No. Template	Templates	Pressings	Overall Dev.	T.Dev
Shoes Labels	2	[1,0,0,1,1,0,1,1,1,2,1,2,0,2,2,4,5,5,6]	80001	2.02	-7.13/ 6.14
		[0,3,3,0,0,4,2,2,3,0,4,1,9,3,3,2,0,0,1,0]	26666		
	3	[1,1,1,1,1,2,1,0,2,2,1,2,3,3,4,2,2,4,6]	69333		
		[0,0,0,0,0,1,0,2,6,0,1,4,3,0,0,3,8,8,4,0]	26250		
		[1,1,1,1,1,0,1,0,2,2,1,2,3,3,4,5,5,2,4]	10667	0.72	-1.56/ 6.67
Pencils Boxes	2	[No Hay Soluciones Factibles]			
	3	[1,0,0,0,0,1,0,2,1,2,2,0,1,1,1,0,1,3,3,2,2,3,5,5]	40664	1.61	-6.00/ 7.00
		[0,2,2,2,1,1,2,0,0,0,3,1,2,2,1,1,1,4,4,3,2,1]	27238		
[1,0,0,1,3,0,2,0,5,0,0,1,2,0,0,8,4,0,3,1,1,2,0,2]	9721				
Gif Boxes	2	[No Hay Soluciones Factibles]			
	3	[No Hay Soluciones Factibles]			
Shirt Labels	2	[No Hay Soluciones Factibles]			
	3	[0,2,2,0,0,2,0,0,0,1,0,1,1,3,1,1,1,1,0,1,1,1,0,0,0,3,0,0,0,0,2,2,0,4,1,2,0,1,3,0,1,3,1,1,2,3,2,4,4,6,3,5,3,1,0,1,0,7]	68562	2.01	-3.74/ 10.00
		[1,0,0,1,1,0,1,1,2,1,2,1,1,1,0,1,1,1,1,1,2,1,1,1,2,2,2,0,2,2,2,2,1,1,2,0,2,1,3,2,1,2,2,1,2,2,2,1,3,1,1,0,2,1,2,3,4,3,4,0]	101750		
[2,0,0,2,2,0,2,2,0,1,0,1,1,1,0,1,1,2,1,1,0,2,2,2,0,0,0,0,0,0,0,3,1,1,3,0,1,3,0,1,0,5,1,1,3,3,0,1,0,4,4,2,2,0,3,5,3,5,4,1]	19312				
CD Cover	3	[No Hay Soluciones Factibles]			
	4	[0,0,0,1,1,1,0,0,0,1,0,0,0,1,0,2,1,0,1,1,1,2,0,1,0,1,1,1,0,2,0,1,0,1,1,1,0,2,2,1,2,2,0,1,3,1,0,3,1,2,2,1,1,0,1,0,3]	94500	2.77	-10.00/ 7.08
		[1,1,0,0,0,0,1,1,0,0,0,1,1,1,2,0,1,1,0,0,1,0,2,1,2,2,1,1,0,0,1,0,1,0,2,1,2,2,1,1,0,0,1,0,1,0,2,2,0,4,1,1,2,1,0,0,3,2,2,2,0]	94500		
		[0,0,1,0,0,0,0,0,1,0,1,0,1,0,0,0,0,1,1,1,0,0,0,0,1,0,1,0,0,0,0,1,0,1,0,1,3,0,0,2,0,0,3,1,0,0,4,0,2,2,2,4,1,4,2,4,2]	96375		
[0,1,1,1,1,1,1,0,0,0,3,0,0,4,1,1,1,4,0,3,6,0,0,4,0,0,3,1,2,0,2,1,4]	32750				
Book Cover	2	[No Hay Soluciones Factibles]			
	3	[0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,2,1,0,0,2,0,0,1,0,2,1,0,0,1,2,1,2,0,4,3,1,4,1,5]	76500	2.09	-7.10/ 9.29
		[1,0,1,1,1,0,1,1,1,0,1,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,2,2,0,1,2,2,1,1,1,1,2,0,1,1,2,0,1,1,0,3,1]	72000		
		[0,0,0,0,0,0,1,0,0,0,1,0,1,1,1,0,1,1,0,1,1,0,2,2,1,0,1,0,0,0,1,0,0,0,1,0,0,2,0,1,0,1,0,0,1,0,0,2,0,1,3,3,1,2,2,1,2,0,1,3,1,2,0]	76500		

La versión Genética tiene un desempeño muy pobre, por lo cual no estamos presentando la tabla correspondiente. Según lo que podemos notar en los resultados mostrados, se evidencia un desempeño ligeramente mejor para de la búsqueda Tabú frente a las demás metaheurísticas.

Algoritmos Híbridos (meméticos)

Al aplicar el GA combinado con una de las búsquedas locales se logra un mejor desempeño de las técnicas. Las Tablas 6 y 7 muestra las mejores soluciones encontradas al aplicar los algoritmos meméticos, notese que para el escenario Gif Boxes la técnica que utiliza el operador de mejora local TS, no logra encontrar solución (comparado con el memético que utiliza HC). Por otra parte, se observa la similitud en cuanto a mejor solución para las dos técnicas (MA-Hc y MA-Ts), esto aplica de igual forma para los porcentajes de soluciones factibles.

Como dato curioso, la técnica Tabú, logra encontrar alguna solución para el escenario Gif Boxes cuando se ataca con 3 plantillas (Máximo de plantillas), cosa que no logra ninguna de las demás técnicas,

inclusive cuando se aplican combinadas. Por tanto consideramos que este escenario particular pareciera ser bastante complejo (incluso más que el escenario Magazine, Proll y Smith, 1998).

Como observación podemos señalar que, según la experimentación y análisis realizados, las técnicas meméticas han mostrado ser una alternativa con un alto grado de desempeño para resolver el problema del diseño de plantillas; ya que logra encontrar soluciones de alta calidad en el 100% (con el operador de mejora local HC) de los escenarios atacados.

Conclusiones y trabajo futuro

Este artículo ha tratado el problema del diseño de plantillas, un problema combinatorio que se enmarca dentro de la categoría de problemas NP-Duros y que ha sido previamente abordado a través de programación lineal y programación con restricción. Recientemente se abordó el problema utilizando técnicas metaheurísticas, específicamente con búsquedas locales y algoritmos genéticos. Ahora, este documento ha propuesto un enfoque híbrido para hacer frente a este problema. Además, estamos presentando nuevos escenarios para mostrar la robustez de nuestra batería de algoritmos aplicados sobre el problema del diseño de plantillas, estos escenarios los podemos considerar de gran complejidad. La propuesta híbrida ha mostrado que resulta ser una alternativa altamente eficiente para resolver este tipo de problemas. Se puede apreciar que la sinergia entre los algoritmos genéticos y los operadores de mejor local (basados en vecindario) cooperan en los procesos de explotación y exploración del espacio de búsqueda, logrando alcanzar resultados prometedores.

El desperdicio de materia prima se reduce considerablemente al atacar el problema con el uso de las técnicas híbridas, (i.e, consideremos el caso del escenario Book Cover, por mencionar uno, la configuración que nos presenta el algoritmo híbrido

nos permite ahorrar 48000 unidades de producto terminado). Al realizar un torneo de las técnicas presentadas nos damos cuenta de la robustez de las técnicas meméticas, particularmente el híbrido entre la búsqueda HC y el algoritmo genético.

Como trabajo futuro, estamos considerando la incorporación de mecanismos que permitan romper las simetrías que presenta el problema, así como introducir un esquema colaborativo entre las diferentes técnicas aplicadas, esto nos permitirá mejorar las soluciones encontradas hasta el momento y de esta forma poder vencer nuevos escenarios más complejos (i.e, Gif Boxes). Particularmente nos llama poderosamente la atención el esquema aplicado por Amaya y colaboradores (2011), además de intentar enfocarnos en un segundo modelo para la resolución de este problema.

Tabla 5: Resultados obtenidos para la metaheurística TS, por cada escenario abordado.

Problema	No. Template	Templates	Pressings	Overall Desv.	T.Desv	
Shoes Labels	2	[0,3,0,0,0,4,2,2,0,0,4,1,3,3,0,0,0,3,3,12]	26150	0.68	-1.94/ 2.08	
		[1,0,1,1,1,0,1,1,2,2,1,2,2,2,3,5,5,4,4,2]	80400			
	3	[1,0,1,1,1,1,2,2,2,1,2,2,0,1,1,1,6,3,5,7]	66429			
		[1,0,1,1,1,1,0,0,2,1,0,2,6,3,7,0,0,9,5,0]	13834			
		[0,3,0,0,0,1,0,0,0,3,2,1,6,5,3,13,0,3,0,0]	26069	0.61	-2.24/ 2.20	
Pencils Boxes	2	[0,0,2,2,2,1,1,3,3,1,0,0,0,0,2,2,3,2,3,1,0,7]	27500	4.36	-10.00/ 10.00	
		[1,1,0,0,0,1,1,0,0,1,1,2,2,2,2,2,1,2,2,3,2,4,5,1]	49500			
	3	[3,1,1,2,2,3,0,0,2,2,0,0,2,0,1,0,0,3,2,4,0,2,4,2]	9061			
		[0,1,1,1,1,1,0,1,1,2,2,1,1,2,0,1,1,2,2,3,3,5,3]	42751			
		[1,0,0,0,0,1,3,1,1,0,0,1,2,0,4,3,3,3,3,3,0,4]	26083	1.96	-8.22/ 10.00	
Gif Boxes	2	[No Hay Soluciones Factibles]				
	3	[0,0,0,0,0,0,0,0,0,2,0,0,1,2,1,1,2,1,0,1,	19500	40500	2.07	-7.69/ 5.13
2,0,0,0,1,0,0,1,0,2,1,2,0,0,0,1,1,3,1,6]						
[0,0,0,0,0,0,0,1,1,0,1,1,0,1,1,0,1,1,0,1,0,						
0,1,1,1,1,2,2,1,2,1,1,0,1,2,1,2,1,1,2,0]						
		[1,1,1,1,1,1,1,0,0,0,1,0,1,0,0,1,0,1,2,	21000			
		1,1,1,2,1,0,0,1,0,0,1,2,2,0,3,0,2,1,1,0]				
Shirt Labels	2	[No Hay Soluciones Factibles]				
	3	[1,1,0,0,1,1,1,1,1,2,2,2,0,2,1,1,1,1,2,1,0,2,2,1,	101750	24025	2.75	-6.83/ 10.00
		2,1,2,2,1,2,2,1,1,0,1,1,1,1,1,2,1,1,2,1,2,4,2,2,3,3,3,3]				
		[1,1,0,0,1,1,1,1,1,0,0,0,0,1,1,1,1,0,1,1,0,0,0,2,1,0,0,2,				
0,1,3,3,1,3,4,0,3,2,0,3,3,2,2,2,3,1,1,5,1,0,2,2,3,3,4,4,5,2]						
		[0,0,2,2,0,0,0,0,1,0,0,0,3,0,1,1,1,1,0,1,3,0,0,0,2,1,3,0,0,1,	64625			
		0,2,0,0,2,0,0,3,2,4,3,2,2,3,3,1,3,5,2,5,4,0,3,3,3,1,1,1,2]				
CD Cover	3	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,2,	129332	3.72	-9.33/ 10.00	
		0,2,0,1,0,1,1,1,1,1,3,1,3,2,2,1,1,1,3,2,3,1,3,3,1]				
		[1,0,0,0,1,0,1,1,1,1,1,1,2,2,2,1,0,0,0,1,0,0,0,1,0,				
		1,0,2,0,2,2,2,2,1,1,1,1,0,1,2,1,3,1,0,0,2,2,1,0,1]				
			[0,1,1,1,0,1,0,0,0,0,0,0,0,0,1,2,2,1,0,1,1,1,0,0,	89334		
			2,0,1,2,1,0,0,0,1,1,0,3,1,1,0,3,1,3,0,2,1,2,4]			
	4	[3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,4,1,1,0,	32857			
		0,0,0,3,0,1,1,1,1,1,0,3,2,0,2,1,0,1,1,2,5,2,1,4]	97143			
[0,0,1,0,0,1,1,1,1,0,0,0,0,2,0,0,0,0,1,0,1,0,1,0,		95715				
0,1,0,1,1,2,1,1,0,0,0,1,3,2,2,1,3,1,2,3,0,1,2,5,3]						
		[0,0,0,0,1,0,0,0,0,1,1,1,1,0,1,0,2,2,0,0,0,1,0,2,	91428	2.07	-8.84/ 9.52	
		1,2,2,0,1,1,1,2,2,3,2,1,1,1,1,1,2,0,0,2,2,4,0,0,1]				
		[0,1,0,1,0,0,0,0,0,0,0,0,1,0,1,2,0,0,2,1,2,0,1,1,1,				
		2,0,1,1,1,0,1,0,1,0,3,2,0,2,1,3,0,4,3,0,2,0,3,1,1]				
Book Cover	2	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,1,1,1,	153000	1.75	-7.10/ 6.25	
		1,1,1,0,1,0,1,1,1,1,1,2,2,2,1,2,2,2,2,2,3]				
		[1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,0,0,2,0,0,0,0,	72000			
	0,0,0,2,0,2,1,1,1,2,2,0,0,2,0,0,1,1,1,2,0]					
	3	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,	80000			
		1,1,1,1,1,0,0,1,0,1,0,0,1,3,2,2,2,1,2,3,5,3,4]	72000			
[1,1,1,1,1,0,0,1,1,0,1,0,0,0,2,2,2,2,0,0,1,1,						
		0,0,0,0,0,3,1,0,1,0,2,1,1,1,2,2,0,1,1,0,2,2]				
		[0,0,0,0,1,1,0,0,1,0,1,1,1,1,0,0,0,0,2,1,0,0,	75000	1.19	-3.25/ 7.14	
		1,1,1,1,1,2,0,1,3,1,4,2,2,0,1,0,0,3,2,1,0,1,0]				

Referencia

- Amaya J, Cotta C, Fernández Leiva AJ (2011). Memetic cooperative models for the tool switching problem. *Memetic Computing* 3:199–216.
- Feng Y, Wang GG, Deb S, Lu M, Zhao XJ (2017). Solving 0–1 knapsack problem by a novel binary monarch butterfly optimization. *Neural Computing and Applications* 28(7):1619–1634, DOI 10.1007/s00521-015-2135-1, URL <https://doi.org/10.1007/s00521-015-2135-1>.
- Flener P, Frisch AM, Hnich B, Kzltan Z, Miguel I, Walsh T (2001). Matrix modelling. In: In: Proc. of the CP-01 Workshop on Modelling and Problem Formulation. International Conference on the Principles and Practice of Constraint Programming, available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.5946>.
- Hart WE (1994). Adaptive global optimization with local search. PhD thesis, University of California.
- Kasemset C, Chernsupornchai J, Pala-ud W (2015). Application of mfca in waste reduction: case study on a small textile factory in thailand. *Journal of Cleaner Production* 108, Part B:1342 – 1351, DOI <http://dx.doi.org/10.1016/j.jclepro.2014.09.071>, URL <http://www.sciencedirect.com/science/article/pii/S0959652614010105>, material Flow Cost Accounting.
- Kulkarni AJ, Shabir H (2016). Solving 0–1 knapsack problem using cohort intelligence algorithm. *International Journal of Machine Learning and Cybernetics* 7(3):427–441, DOI 10.1007/s13042-014-0272-y, URL <https://doi.org/10.1007/s13042-014-0272-y>.
- Prestwich S, Tarim A, Hnich B (2006). Template design under demand uncertainty by integer linear local search. *International Journal of Production Research*, 44(22):4915–4928.
- Proll L, Smith B (1998). Ilp and constraint programming approaches to a template design problem. *INFORMS Journal of Computing* 10(3):265–275.
- Rodríguez D, Cotta C, Fernández-Leiva AJ (2010). Un problema de diseño de plantillas: Un enfoque metaheurístico basado en búsqueda local. In: et al VC (ed) *Algoritmos Evolutivos y bioinspirados (MAEB2010)*, Garceta, Valencia, pp 743–750.
- Rodríguez D, Cotta C, Fernández-Leiva AJ (2011). The template design problem: A perspective with metaheuristics. In: in *Computer Science LN* (ed) *Eighth IMACS Seminar on Monte Carlo Methods*, Walter de Gruyter, Borovets, Bulgaria, pp 181–191.
- Wang LK, Shamma NK, Hung YT (2016). Waste Treatment in the Metal Manufacturing, Forming, Coating, and Finishing Industries. *Advances in Industrial and Hazardous Wastes Treatment*, CRC Press, URL <https://books.google.co.ve/books?id=7s-H4q8ddlgC>.