

Una búsqueda dispersa para el problema de la mochila multidimensional

Facultad de ingeniería mecánica y eléctrica
Posgrado de ingeniería en sistemas

Angel Isabel Moreno Saucedo
angisabel44@hotmail.com

17 de junio de 2019

El problema de la mochila multidimensional (MKP por sus siglas en inglés: *multidimensional knapsack problem*) es un problema de optimización combinatoria NP-duro. Este problema es bien estudiado, se encuentran muchos algoritmos exactos y metaheurísticos para la resolución del problema. En este artículo se propone una metaheurística de búsqueda dispersa debido a la estructura del problema se puede implementar fácilmente. Los resultados obtenidos fueron enriquecedor para el problema MKP.

sigue:

$$\text{maximizar } z = \sum_{i=1}^n b_i x_i \quad (1)$$

$$\text{sujeto a: } \sum_{i=1}^n w_{ji} x_i \leq c_j \quad j \in \{1, 2, \dots, m\} \quad (2)$$

$$x_i \in \{0, 1\} \quad i \in \{1, 2, \dots, n\} \quad (3)$$

donde x_i es la variable de decisión que indica que objetos son seleccionados. Las m restricciones en 2 también son conocidas como dimensiones.

1. Problema de la mochila multidimensional

Sea I un conjunto de n objetos con beneficios $b_i > 0$ y m capacidades c_j . Cada objeto i consume una cantidad $w_{ji} \geq 0$ en cada capacidad j . El objetivo es escoger un subconjunto de objetos que maximicen el beneficio total sin exceder las m capacidades. El modelo de programación lineal entera para el MKP es como

2. Método de solución

En este artículo proponemos una búsqueda dispersa (SS por sus siglas en inglés: *scatter search*). Sus componentes proporcionan una buena opción para resolver el problema. Se explica primero el algoritmo del constructivo para generar una solución y luego el algoritmo de la búsqueda local.

Para generar una solución se utilizó un algoritmo constructivo simple, el cual consiste en agregar objetos de forma aleatoria hasta que ya no haya más objetos o alguna de las m restricciones de 2 no se satisfaga. El algoritmo 1 muestra la lista de pasos del constructivo.

A continuación, una vez generada una solución sigue mejorar la solución aplicando una

Data: Instancia de MKP.

Result: Solución S

Crear la lista de objetos L y ordenarla de forma aleatoria

$w_j = 0$ para todas las dimensiones

$S = \emptyset$

while $w_j \leq W_j$ **do**

 Escogemos el primer objeto de L

if *ObjetoCabe()* **then**

 Agregamos el objeto a S

 Borramos el objeto de L

 Actualizamos las w_j

end

if $L = \emptyset$ **then**

 Terminar

end

end

Regresar S

Algorithm 1: Pseudocódigo constructivo

búsqueda local (LS por sus siglas en ingles: *local search*). Se utiliza como operador para generar vecinos de la solución en eliminar un objeto que este seleccionado y agregar otro que no encuentre agregado. Es decir, si se tiene como solución el vector $S = (1, 1, 1, 0, 1, 0, 0)$ un vecino de S es $(0, 1, 1, 1, 1, 0, 0)$. El algoritmo 2 muestra los pasos a seguir del LS.

Data: Solución S

Result: Solución S^*

$S^* = S$

while *contador* < *IteracionMax* **do**

$k = \arg \min$

$\{\sum_{j \in J} w_{ji}/b_i | \forall i \in I \wedge i == 1\}$

$S' = S * -k$

 Creamos lista de vecinos N con respecto a S'

$S'' = \arg \max \{f(n) | n \in N\}$

if $S^* < S''$ **then**

$S^* = S''$

 Contador = 0

end

 contador++

end

Regresar S^*

Algorithm 2: Pseudocódigo búsqueda local

Los algoritmos 1 y 2 son utilizados dentro de SS para crear la población inicial. El algoritmo 3 muestra los pasos generales para el SS.

Data: Instancia de MKP.

Result: Solucion S

P = Población de soluciones de tamaño n

S = La mejor solución de P

while *Cantidad máxima de iteraciones*

do

 Escoger el conjunto de referencia R de P

 Combinar las soluciones de R y agregarlas a P

 Reajustar el tamaño de P

 Actualizamos S

end

Regresar S

Algorithm 3: Pseudocódigo búsqueda dispersa

Cada uno de los pasos del algoritmo 3 se implementaron como sigue:

Población cada solución de la población fue creada con un constructivo y despues se aplicó una búsqueda local.

Conjunto de referencia se escogio un porcentaje de la población para el conjunto, el cual la mitad es para soluciones de calidad y la otra parte soluciones diversificadas.

Combinaciones se escoge dos soluciones al azar del conjunto de referencia y se combinan de tal forma de escoger un objeto al azar y la nueva solución consta de la primera parte antes del objeto seleccionado de una solución y la otra del resto. Se recupera factibilidad y se le aplica a la nueva solución búsqueda local.

Reajuste de la población conservamos las n mejores soluciones.

3. Instancias del MKP

Se utiliza un conjunto de instancias de la literatura SAC-94 [1]. El conjunto esta basado en problemas reales, provenientes de diferentes artículos, conteniendo con un número de objetos entre 10-105 y número de recursos entre 2 y 30.

4. Resultados

Se ejecuto el SS con un poblaci3n de 50 soluciones y un m1ximo de iteraciones de 200. Se hizo 20 r3plicas para cada instancia y se reporta el promedio de estas. La tabla 1 muestra los resultados obtenidos.

Se puede observar en la tabla 1 que en 4 instancias se alcanzo el optimo, mientras que las dem1s obtuvieron buen resultado.

5. Conclusiones

El SS es una buena opci3n de metaheur1stico para el MKP. Los resultados obtenidos fueron buenos. Se observa tambi3n que la soluci3n obtenida mediante el constructivo esta en promedio 30 % por debajo del optimo, la soluci3n reportada para la b1squeda local esta en promedio 5 % por debajo del optimo. En algunas instancias no se reporta mejora con la b1squeda local esto es posiblemente por la estructura de la instancia lo cual abre una oportunidad para estudiarlas.

Referencias

- [1] E. BEASLEY. *OR-Library: distributing test problems by electronic mail*. Journal of the Operational Research Society 41.11 (1990), pp. 1069–1072.

Inst	Opt	Const	LS	SS
hp1	3418	2917	2978	3336
hp2	3186	2412	2412	3002
pb1	3090	2770	2770	2992
pb2	3186	2616	2616	3072
pb4	95168	82162	82162	91935
pb5	2139	1971	1971	2076
pb6	776	477	625	723
pb7	1035	621	946	1005
pet2	87061	69394	69394	85943
pet3	4015	2945	3225	4015
pet4	6120	2855	3660	6120
pet5	12400	7525	10330	12400
pet6	10618	10112	10195	10520
pet7	16537	14761	14938	16142
sento1	7772	998	6997	7636
sento2	8722	5465	5465	8642
weing1	141278	102133	102678	140668
weing2	130883	47965	65557	130013
weing3	95677	51927	73768	94908
weing4	119337	77122	92590	119337
weing5	98796	56027	67562	98521
weing6	130623	90470	90470	129032
weing7	1095445	886214	1058098	1092260
weing8	624319	156274	316272	610644

Cuadro 1: Resultados del SS.