

REDUCCIÓN DE GRAFOS SIN PÉRDIDA DE INFORMACIÓN

Autoría principal: Rafael Rodríguez Puente¹ (70%) y Manuel S. Lazo Cortes² (30%)

Colaboradores: Álvaro Rafael Rodríguez Gregorio, José Ángel Inda Herrera, Lisandra Santana Puentes, Luis Lamela Fung, Sergio Marrero Osorio, Yasnary González Pérez, Yosveni Escalona Escalona, Yudiel Rodríguez Larrazabal

Universidad de Ciencias Informáticas (UCI). Carretera a San Antonio de los Baños, Km 2^{1/2}. Reparto Torrens, Boyeros, La Habana.

¹ Autor de correspondencia. Correo electrónico rafaelrp@uci.cu

² Autor de correspondencia. Correo electrónico manuellazo55@gmail.com

RESUMEN

Los algoritmos relacionados con la teoría de grafos han sido ampliamente estudiados. Varios estudios se enfocan en la disminución de la complejidad temporal de estos algoritmos. Las técnicas utilizadas en este sentido, generalmente se basan en reducir un grafo o el espacio de búsqueda de solución, eliminando información redundante para el problema específico que se desea resolver. El proceso de reducción de un grafo consiste en obtener grafos más pequeños (con menos vértices) que tengan las características principales o relevantes del grafo original. En el caso de la búsqueda de caminos óptimos, los algoritmos que hacen uso de la reducción de grafos o del espacio de búsqueda de solución, no garantizan la obtención del óptimo en todos los casos. Lo mismo ocurre en otros tipos de problemas tales como la reducción de grafos en redes de workflow, de computadoras, etc. En este trabajo se propone un algoritmo de reducción de grafos sin pérdida de información. La propuesta tiene una forma flexible de especificar la manera en que se quiere reducir el grafo; por consiguiente, puede ser utilizada en la solución de varios tipos de problemas.

COMUNICACIÓN CORTA

Introducción

Varios algoritmos sobre grafos tienen un elevado costo computacional; algunos de ellos son diseñados para resolver problemas NP o NP-Duros. Por consiguiente, se le presta especial atención a la reducción de la complejidad computacional de estos algoritmos.

Una de las estrategias que se han utilizado consiste en reducir el grafo teniendo en cuenta el problema que se está solucionando. En [?] se realiza una revisión de algoritmos que reducen grafos, concluyendo que varios de estos algoritmos de reducción están enfocados en mantener las características relevantes de grafos que representan diferentes realidades. Además, el estudio arroja que estos algoritmos eliminan o no consideran información que no es necesaria para resolver un problema específico, pero que puede ser indispensable para resolver otros problemas, por lo que su uso no puede ser generalizado para distintos tipos de problemas.

Debido a esta situación, en el presente trabajo se propone un algoritmo de reducción de grafos sin pérdida de información, que pueda ser utilizado en la solución de varios tipos de problemas.

Reducción de grafos sin pérdida de información

Las definiciones de regla de reescritura de grafos, grafo reducido y grafo reducido a partir de un grafo fueron enunciadas en [?] por los autores del presente trabajo.

El algoritmo de reducción diseñado se considera el principal aporte de la investigación ya que reduce un grafo sin que exista pérdida de información en el proceso, lo que contribuye a realizar análisis sobre el grafo reducido y obtener los mismos resultados que se obtienen en el grafo sin reducir. Además, el hecho de que no exista pérdida de información hace posible su uso en la reducción de varios tipos de grafos. El pseudo-código para reducir un grafo se muestra en el Algoritmo 1.

La complejidad del algoritmo de reducción enunciado, en el peor de los casos, es $O(n^2 \sqrt{n} \lg(n - \sqrt{n}))$.

Algoritmo 1 *ReducirGrafo*

Entrada: Un grafo $G = (V, E, f, R)$, donde R es un conjunto de reglas de reescritura, posiblemente vacío y una partición P sobre el conjunto de vértices V

Salida: Un grafo ponderado reducido que representa el mapa

- 1: $P = RefinarParticion(G, P), V_r = ConstruirVerticesReducidos(P), E_r = ConstruirAristas(P, G)$
 - 2: $R_r = ConstruirReglasReescritura(P, G), f_r = \phi$
 - 3: **Para todo** $A_i \in P, |A_i| > 1$ **hacer**
 - 4: $Calcular f(G, R_r[A_i].G_i, R_r[A_i].G_j, f_r)$ {Calcular el costo de pasar por el vértice reducido correspondiente a $A_i \in P$.}
 - 5: **Fin Para**
 - 6: Crear el grafo reducido $G_r = (V_r, E_r, f_r, R_r)$
 - 7: **Retornar** G_r
-

Para probar la corrección de los algoritmos diseñados se enunciaron y demostraron varios lemas y teoremas [?]. A continuación se enuncian el teorema y

el corolario que garantizan que el algoritmo de reducción diseñado es correcto y no presenta pérdida de información.

Sean G un grafo y RE una relación de equivalencia que define una partición $P = V/RE = (A_1, A_2, \dots, A_s)$ sobre el conjunto V de vértices de G .

TEOREMA 1. Sea $G_r = (V_r, E_r, f_r, R_r)$ un grafo reducido a partir del grafo $G = (V, E, f, R)$ y la relación de equivalencia RE , si se aplica la regla de reescritura $R_{r_i} = ((\{v_i\}, \phi), (V_j, E_j, f_j, R_j), \psi_{in}, \psi_{out})$, asociada al vértice $v \in V_r$ sobre el grafo G_r , se obtiene un grafo $G'_r = (V'_r, E'_r, f'_r, R'_r)$ y se cumple que siendo $v_i \in V_r, \forall v_k \in V_j$, tal que $(v_i, v_k) \in RE, [(v_i, v_k) \in E \rightarrow (v_i, v_k) \in E'_r]$.

COROLARIO 1.1. Sea $G_r = (V_r, E_r, f_r, R_r)$ un grafo reducido a partir del grafo $G = (V, E, f, R)$ y la relación de equivalencia RE , si se aplica el conjunto de reglas de reescritura R_r , asociadas a los vértices reducidos del grafo G_r , se obtiene el grafo $G = (V, E, f, R)$ a partir del cual se redujo G_r .

Aplicaciones

Una aplicación de los grafos reducidos se puede encontrar en el análisis de redes. En el Algoritmo 2 se muestra el pseudo-código del algoritmo de Dijkstra modificado para realizar búsquedas de caminos óptimos en grafos reducidos. Este algoritmo es otro de los principales aportes del presente trabajo [?], ya que garantiza la obtención de un camino óptimo en el grafo reducido, de igual costo que el camino óptimo encontrado por el algoritmo de Dijkstra en el grafo original.

Se pudo comprobar que la complejidad temporal del algoritmo modificado es del mismo orden que el algoritmo original $O(n \lg(n))$. Note que el tamaño de la instancia (la cantidad de vértices del grafo reducido) puede ser sustancialmente menor que en el caso del algoritmo de Dijkstra, siendo más eficiente la búsqueda de caminos óptimos en grafos reducidos.

Algoritmo 2 *MDijkstra*

Entrada: Un grafo ponderado reducido $G = (V, E, f, R)$ y un vértice de origen v_{origen}

Salida: Vector D_n de distancias mínimas, vector Pr_n de predecesores

```
1:  $C_n = \{\}, cola = \{\}$  {La variable cola representa una cola con prioridad}
2: Para todo  $v \in V$  hacer
3:    $D_n[v] = f(v_{origen}, v_{origen}, v), Pr_n[v] = v_{origen}$ 
4:   Adicionar(cola, v,  $D_n[v]$ ) {Se adiciona el vértice  $v$  tomando como prioridad la distancia hasta el mismo.}
5: Fin Para
6: Mientras !Vacía(cola) hacer
7:    $w_n = \text{Extraer}(cola), C_n = C_n \cup \{w_n\}$ 
8:   Para todo  $v \in \text{Adyacentes}(w_n)$  hacer
9:     Si  $D_n[v] > D_n[Pr_n[w_n]] + f(Pr_n[w_n], w_n, v)$  entonces
10:       $D_n[v] = D_n[Pr_n[w_n]] + f(Pr_n[w_n], w_n, v), Pr_n[v] = w_n$ 
11:      DecrementarLlave(cola, v,  $D_n[v]$ ) {Si  $D_n[v]$  es menor que la prioridad de  $v$  en la cola se modifica la prioridad.}
12:   Fin Si
13:   Si  $v$  es reducido entonces Actualizar distancia y predecesor de cada vértice adyacente a  $v$  utilizando la función  $f$ 
14: Fin Para
15: Fin Mientras
```

Se demostró que el camino óptimo obtenido con este nuevo algoritmo sobre un grafo reducido, tiene el mismo costo que el camino obtenido al aplicar el algoritmo de Dijkstra sobre el grafo original, para ello se enunció y demostró el siguiente teorema.

Sean $G = (V, E, f_c)$ un grafo, $G_r = (V_r, E_r, f)$ un grafo reducido obtenido a partir del grafo G , d_n , la distancia obtenida luego de aplicar el algoritmo de Dijkstra sobre un grafo y D_n , la distancia obtenida luego de aplicar el Algoritmo 2 sobre un grafo reducido.

TEOREMA 2. Sea $Ca = (v_1, \dots, v_n)$ un camino de costo c , obtenido al aplicar el algoritmo Dijkstra sobre el grafo G , donde v_1 y v_n son vértices no reducidos en el grafo G_r , entonces $\exists Ca' = (u_1, u_2, \dots, u_t)$ de costo c , $u_1 = v_1, u_t = v_n$, tal que Ca' es un camino óptimo en G_r .

COROLARIO 2.1. Sea $Ca = (v_1, \dots, v_n)$ un camino obtenido aplicando el algoritmo de Dijkstra en el grafo G , $\forall i \in \{1, 2, \dots, n\}$ tal que $Ca[i]$ es un vértice no reducido en G_r , se cumple que $D_{N-1}(Ca[i]) = d_{M-1}(Ca[i])$

La búsqueda de caminos óptimos puede ser utilizada además, en Sistemas de Información Geográfica como se muestra en [?].

El valor práctico del algoritmo de reducción diseñado no se limita al entorno del análisis de redes, sino que puede ser aplicado para resolver otros tipos de problemas en los que aparece un grafo con determinada complejidad y resulta útil obtener un grafo reducido, sin que se pierda la información contenida en el grafo primario. Tal es el caso del Diseño Racional en Ingeniería Mecánica.

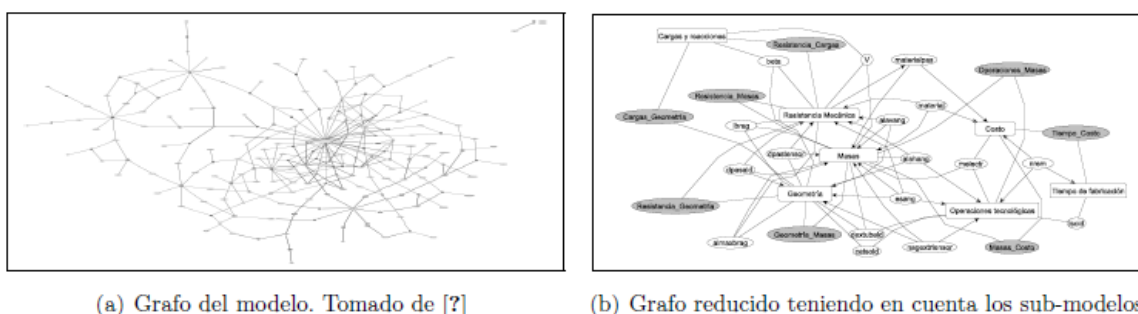


Figura 1. *Ejemplo de reducción de grafos.*

Conclusiones

Como resultado del presente trabajo se obtuvo un algoritmo de reducción de grafos sin pérdida de información. El hecho de que no presenta pérdida de información implica que puede ser utilizado en diversos ámbitos. En base a los resultados obtenidos se arribó a las siguientes conclusiones:

1. El algoritmo de reducción de grafos diseñado como parte de esta investigación garantiza que no se pierde información en el proceso de reducción. Además, se demostró que un grafo reducido se puede utilizar para resolver el problema de la búsqueda de caminos óptimos.
2. Con las demostraciones de corrección realizadas se garantiza que el Algoritmo 2 (*MDijkstra*), en grafos reducidos, obtiene caminos de igual costo a los obtenidos por el algoritmo de Dijkstra en los grafos sin reducir.
3. Haciendo uso de grafos reducidos se garantiza escalabilidad respecto al tamaño del grafo sobre el que se realiza el análisis.
4. La generalidad del algoritmo de reducción propuesto radica en la posibilidad de su aplicación en distintos tipos de redes, así como en el Diseño Racional en Ingeniería Mecánica.

Referencias Bibliográficas

- (1) Marrero Osorio S., Martínez Escanaverino J.: Diseño paramétrico de pinzas de fricción. *Ingeniería Mecánica* 12, 1 (2009), pp. 37-48.
- (2) Rodríguez-Puente R.: Sistema de Información Geográfica. Tesis de maestría, Universidad de las Ciencias Informáticas, julio 2007.
- (3) Rodríguez-Puente R.: Modelo basado en grafos reducidos para la representación y análisis de redes en Sistemas de Información Geoespacial. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas, Universidad de las Ciencias Informáticas, Diciembre 2012.
- (4) Rodríguez Puente R., Lazo Cortes M. S.: Graph-reduction algorithm for finding shortest path in Geographic Information Systems. *IEEE Latin American Transactions* 10, 6 (2012), pp. 2201{2208.
- (5) Rodríguez-Puente R., Marrero-Osorio S., Lazo-Cortés M.: Aplicación de un algoritmo de reducción de grafos al Método de los Grafos Dicromáticos. *Ingeniería Mecánica* 15, 2 (2012), pp. 158-169.