A New Genetic Algorithm for Graph Coloring

Alhely González Luna

2024-11-26

https://ieeexplore.ieee.org/abstract/document/6663163

Tabla de contenidos

Motivación

El problema de coloreo de grafos es un ejemplo clásico de optimización combinatoria NP-difícil. La solución a este problema de coloreo de grafos encuentra aplicaciones en diversos campos de la ingeniería. Este artículo demuestra la robustez de los algoritmos genéticos para resolver el coloreo de grafos. El algoritmo genético propuesto emplea un innovador operador de cruza de genes conflictivos ((vértices mal coloreados) con un solo padre y una mutación de genes conflictivos como sus operadores. El tiempo requerido para obtener una solución convergente con este método genético propuesto se ha comparado con enfoques existentes y ha demostrado ser efectivo. El rendimiento de este método de aproximación se evaluó utilizando algunos grafos de referencia, y se encontró que es competitivo.

Grafos

Un grafo simple $G = (V, E)$ consiste en un conjunto de vértices $V(G) = \{v_1, v_2, \ldots, v_n\}$ y un conjunto de aristas $E(G) = \{e_1, e_2, \ldots, e_m\}$, donde cada arista e_k está asociada a un par único y no ordenado de vértices (v_i, v_j). La matriz de adyacencia $A(G)$ de G es una matriz binaria simétrica de tamaño $n \times n$, donde $A_{ij} = 1$ si existe una arista entre los vértices v_i y v_j, los cuales se denominan vértices adyacentes; y $A_{ij} = 0$ en caso contrario.

Coloreo de grafos

El número mínimo de colores, $\chi(G)$, requerido para colorear los vértices de un grafo G de forma que ningún par de vértices adyacentes comparta el mismo color constituye el *Problema de Coloreo de Grafos* (*Graph Coloring Problem*, GCP). Para un grafo simple G con n vértices, el color óptimo $\chi(G)$ puede encontrarse dentro de un espacio de soluciones de tamaño $n!$. Por lo tanto, se requiere un método de aproximación que minimice el espacio de búsqueda.
El GCP se utiliza para resolver problemas del mundo real como la programación de horarios, la asignación de registros, la asignación de canales y la reducción de ruido en circuitos VLSI, entre otros. Dado que el GCP es un problema NP-difícil, hasta ahora no se ha desarrollado un método que lo resuelva en tiempo polinómico. Debido a su naturaleza NP-completa, es necesario un algoritmo rápido y efectivo para generar un coloreo óptimo.

Algoritmo propuesto

Este artículo presenta un nuevo método genético para resolver el *Problema de Coloreo de Grafos* , que aplica un método de selección proporcional (ruleta) a la aptitud para elegir un mejor gen en cada generación. Se han diseñado los operadores de cruza de genes conflictivos con un solo padre (*Single Parent Conflict Gene Crossover*, SPCGX) y de mutación de aristas conflictivas para reducir el espacio de búsqueda y minimizar el número de generaciones.
Este procedimiento ha sido probado con algunos grafos de referencia como *queen*, *Mycielski*, *huck.col*, *jean.col*, *games120.col*, *miles250.col*, *david.col*, *anna.col*, además de algunos grafos planares, y ha demostrado ser efectivo.

Secuencia e Inicialización de la Población

- Sea la secuencia de genes $(g_1, g_2, ..., g_n)$ la asignación de colores al conjunto de vértices V, donde $gi \in 1 \leq i \leq n$ es un entero positivo distinto de cero.
- Sean p_c y p_m las probabilidades de cruce y mutación, respectivamente.
- Sea g el número de generación y P_g la colección aleatoria de secuencias de genes en la generación g, también denominada población.
- Sea $N = \| P_g \|$ el tamaño de la población y también el número de secuencias de genes distintas en la generación g
- P_0 la población en inicial.

Se generan N secuencias de genes aleatorias para P_0 de modo que el valor de cada gen se extraiga de $[1, \chi(G)]$.

Función de aptitud

Sea $f(G) > 0$ la función de aptitud general del grafo G y también el número de enteros distintos presentes en la secuencia de genes durante cada generación, que se determina utilizando la estrategia de supervivencia del más apto. Ahora, la función objetivo del GCP es minimizar f(G) tal que $f(G) - \chi(G) = 0$.

Genes/aristas Conflictivos y Secuencia de Genes

En una secuencia de genes dada $s = (s_1, s_2, ..., s_i, ..., s_j, ..., s_n)$, los genes s_i, s_j están en conflicto si y solo si $s_i = s_j$ y $(i,j) \in E(G)$. Y por lo tanto, la secuencia de genes, s.

Evaluación de la Aptitud y Selección

Se utiliza la selección proporcional de aptitud para seleccionar algunas secuencias de genes específicas en cada generación g y se realiza de la siguiente manera:

1. Calcular la función de evaluación de aptitud $F(i)$ para cada secuencia de genes i $(1 \leq i \leq N)$ en P_g como el *número de aristas en conflicto* presentes en esa secuencia.
2. Calcular la probabilidad para cada secuencia de genes i como
$$p_i = \frac{F(i)}{\sum_{i=1}^{N} F(i)}.$$
3. Calcular el número esperado de observaciones para cada i como $E(i) = N \cdot p(i)$.
4. Elija dos mejores secuencias de genes i y j tales que $E(i)$ y $E(j)$ sean mínimo y mínimo siguiente entre todos los números esperados de observaciones.
5. Elija la peor secuencia de genes k donde $E(k)$ es máxima entre todos los números esperados de observaciones.

Cruce

La operación de cruce se lleva a cabo en las secuencias genéticas seleccionadas para generar dos nuevas secuencias genéticas. Las secuencias genéticas seleccionadas i y j se utilizan para generar las descendencias i' y j' mediante el operador de cruza SPCGX, utilizando una probabilidad de cruza elegida p_c

Cruce

a. Sea $i = (i_1, i_2, i_3, \ldots, i_q, \ldots, i_r, \ldots, i_n)$ y $j = (j_1, j_2, j_3, \ldots, j_s, \ldots, j_t, \ldots, j_n)$ las dos secuencias seleccionadas.

b. Elegir una probabilidad aleatoria p_{cr}.

c. Si $p_{cr} > p_c$, ir al paso d de la operación de cruza; en caso contrario, ir al paso de mutación.

d. Identificar todos los pares de aristas conflictivas en i y j.

e. Sean (q, r) y (s, t) dos pares de aristas conflictivas (es decir, $i_q = i_r$ y $j_s = j_t$) de las secuencias genéticas i y j. Aplicar la siguiente operación de cruza de genes conflictivos para generar dos descendencias: $(i_1, i_2, i_3, \ldots, i_q, \ldots, i_{r+1}, \ldots, i_n)$ y $(j_1, j_2, j_3, \ldots, j_s, \ldots, j_{t+1}, \ldots, j_n)$.

f. Repetir el paso e para todos los pares conflictivos identificados en el paso d y generar dos secuencias genéticas actualizadas:

$$i' = (i'_1, i'_2, i'_3, \ldots, i'_f, \ldots, i'_g, \ldots, i'_n), \quad j' = (j'_1, j'_2, j'_3, \ldots, j'_k, \ldots, j'_l, \ldots, j'_n).$$

Mutación

Con una probabilidad de mutación elegida p_m, las secuencias genéticas i' y j' se mutan utilizando el siguiente operador de mutación de genes/aristas conflictivos para generar las descendencias i'' y j''

Mutación

a. Elegir una probabilidad aleatoria p_{mr}.
b. Si $p_{mr} > p_m$, ir al paso c de la operación de mutación; en caso contrario, ir al paso de Actualización de la Población.
c. Identificar todos los pares de aristas conflictivas en i' y j'.
d. Sean (f', g') y (k', l') dos pares de aristas conflictivas (es decir, $i'_f = i'_g$ y $j'_k = j'_l$) de las secuencias genéticas i' y j'. Aplicar la siguiente operación de mutación de genes conflictivos para generar dos descendencias:

$$(i'_1, i'_2, i'_3, \ldots, i'_f - 1, \ldots, i'_g, \ldots, i'_n), \quad (j'_1, j'_2, j'_3, \ldots, j'_k - 1, \ldots, j'_l, \ldots, j'_n).$$

e. Repetir el paso d para todos los pares conflictivos identificados en el paso c y generar dos secuencias genéticas actualizadas:

$$i'' = (i''_1, i''_2, i''_3, \ldots, i''_f, \ldots, i''_g, \ldots, i''_n), \quad j'' = (j''_1, j''_2, j''_3, \ldots, j''_k, \ldots, j''_l, \ldots, j''_n).$$

Actualización de la Población

Calcular $f(i'')$ y $f(j'')$ para las dos secuencias genéticas actualizadas i'' y j''. Si $f(i'')$ o $f(j'')$ es igual a $\chi(G)$, ir al paso *Evaluación de la Aptitud y Selección* . En caso contrario, $g = g + 1$ y actualizar la población de la siguiente generación P_g como sigue:

a. Si $F(i'') < F(\text{peor})$, reemplazar el peor gen con i''.
b. Si $F(j'') < F(\text{peor})$, reemplazar el peor gen con j'' y regresar al paso *Evaluación de la Aptitud y Selección*.

Casos coloreables y no coloreables

- **Caso Coloreable**:
  1. En las generaciones subsecuentes, el valor de la función de aptitud disminuye de manera monótona y converge a una solución óptima en la generación $(q+1)$
  2. Inicialmente, la función de aptitud disminuye de manera monótona hasta un número finito de generaciones, digamos q. Luego, basándose en los valores de p_{cr} y p_{mr}, los valores de la función de aptitud aumentan nuevamente hasta un número finito de generaciones, digamos t, es decir, después de un número finito de generaciones, digamos u, se alcanza la solución óptima.
- **Caso No Coloreable**: En el caso no coloreable, después de un número finito de generaciones, digamos q, se cumple que $f_g(i) \geq c$ y $f_g(j) \geq c$ para todos $g = q, q+1, q+2, \ldots$.

Convergencia

Sea S el espacio de búsqueda finito del algoritmo genético. El algoritmo genético converge a una solución óptima global si se cumplen las siguientes condiciones

a. Para cualquier par de secuencias genéticas $i, j \in S$, j es alcanzable desde i. Es decir, sea i' una secuencia obtenida de la operacion de cruce y i'' la secuencia obtenida de i' por la operacion de mutación, se dice que i'' es alcanzable desde i' si

$$0 < pi'' = (crossover(i) and mutation(i')) < 1$$

donde p es la probabilidad del evento.

b. La población P_g es monótona (la poblacion no empeora)

Resultados

Los experimentos se realizaron tanto para casos coloreables como no coloreables utilizando algunos de los grafos de referencia, en un sistema Intel Core i5-2450M a 2.5GHz con Turbo boost hasta 3.1GHz, en un entorno Java JDK 1.7. El algoritmo ha sido implementado utilizando el lenguaje JAVA y los resultados se presentan en las siguientes tablas.

Resultados

TABLE I. PERCENTAGE OF SUCCESSFUL RUNS ON GRAPHS FOR $\chi(G)$, $\chi(G)+1$ & $\chi(G)+2$

Graph No	Graph (G)			Percentage of successful runs of			x *	g @
	Graph Type	Instances	$\chi(G)$	$\chi(G)$	$\chi(G)+1$	$\chi(G)+2$		
1	queen5_5.col	n=25; m=320	5	75.0%	89.8%	93.2%	6	130
2	queen6_6.col	n=36; m=580	7	44.9%	61.8%	81.8%	4	13
3	queen7_7.col	n=49; m=952	7	31.6%	35.2%	39.5%	2	16
4	queen8_8.col	n=64; m=1456	9	26.9%	39.4%	43.2%	1	13
5	myciel5.col	n=47; m=236	6	93.0%	98.2%	100.0%	23	254
6	myciel6.col	n=95; m=755	7	88.3%	100.0%	100.0%	13	7
7	myciel4.col	n=23; m=71	5	100.0%	100.0%	100.0%	3	5
8	myciel3.col	n=11; m=20	4	100.0%	100.0%	100.0%	3	4
9	Heawood graph	n=25; m=67	4	63.0%	95.8%	100.0%	3	2
10	Hexagonal grid graph	n=20; m=43	3	98.3%	100.0%	100.0%	1	4
11	Planar graph	n=6; m=10	3	100.0%	100.0%	100.0%	2	5
12	Planar graph	n=10; m=21	4	100.0%	100.0%	100.0%	8	6
13	Dodecahedron Schlegel	n=20; m=30	3	71.1%	100.0%	100.0%	3	6
14	5-connected planar	n=12; m=30	4	78.5%	100.0%	100.0%	5	7
15	5-connected planar	n=28; m=70	4	84.7%	100.0%	100.0%	3	6
16	huck.col	n=74; m=301	11	63.2%	85.9%	93.8%	632	547
17	jean.col	n=80; m=254	10	27.9%	75.2%	89.8%	279	667
18	myciel7.col	n=191; m=2360	8	22.1%	71.4%	97.5%	221	260
19	games120.col	n=120;m=638	9	18.3%	73.4%	93.5%	183	888
20	miles250.col	n=128;m=387	8	1.4%	30.8%	72.7%	14	1196
21	david.col	n=87;m=406	11	6.5%	39.6%	80.4%	65	1176
22	anna.col	n=138;m=493	11	6.1%	46.9%	82.7%	61	2053

* Number of distinct optimal gene sequences generated; @ Average number of generations performed for $\chi(G)$

TABLE II. SIMULATED RESULTS ON GRAPHS TOWARDS ACHIEVING $\chi(G)$ WITH DIFFERENT CATEGORIES OF p_c & p_m

Graph No	$p_c < p_m$				$p_c > p_m$				$p_c = p_m$				x^*
	R	g	C	D	R	g	C	D	R	g	C	D	
1	99.5%	1	43	32	84.8%	3	43	145	99.1%	3	42	78	5
2	1.4%	15	652	1034	31.2%	15	508	1199	5.7%	20	1188	1295	4
3	0.7%	44	7918	5042	27.4%	20	1809	3230	8.8%	34	5779	4319	2
4	17.3%	22	3171	1849	33.4%	15	1101	1746	40.4%	21	2318	2009	1
5	99.6%	1	36	6	89.1%	4	52	67	99.1%	3	41	21	14
6	99.3%	1	13	7	88.5%	5	31	123	96.0%	2	16	21	10
7	100.0%	1	16	2	100.0%	4	22	35	100.0%	3	19	7	2
8	100.0%	1	16	5	99.8%	4	16	65	100.0%	3	16	22	3
9	99.2%	1	19	7	76.7%	2	20	31	97.7%	2	19	24	3
10	99.8%	1	23	12	97.8%	4	24	135	97.7%	2	23	30	1
11	100.0%	1	5	2	100.0%	4	6	28	100.0%	3	6	6	2
12	100.0%	2	19	7	99.8%	5	20	85	99.5%	5	19	21	10
13	6.8%	2	25	48	45.1%	6	25	180	14.2%	4	25	62	1
14	99.1%	1	9	2	82.2%	3	12	32	95.1%	3	11	8	2
15	98.5%	1	37	14	85.4%	5	48	334	96.0%	3	38	35	3
16	33.5%	1430	7398	2910	58.7%	530	2753	4695	46.2%	682	3938	2610	935
17	12.4%	817	5627	2096	20.5%	1316	4928	8634	21.3%	97	498	391	345
18	7.1%	779	23292	7563	16.8%	423	6873	13437	8.9%	523	13008	8656	235
19	8.1%	2048	18878	6398	15.8%	820	5226	8542	12.0%	2562	28136	18125	242
20	0.6%	368	6850	1329	0.8%	3921	25554	49871	0.5%	43	1019	1174	13
21	1.7%	1516	12558	5219	5.2%	833	3797	6758	3.6%	63	585	420	70
22	0.3%	6963	26638	6892	2.9%	2414	11110	19790	0.9%	37	747	393	31

R- Percentage of successful runs; g- Average number of generations; C Average number of crossovers; D Average number of mutations

Conclusión

El uso de algoritmos genéticos, como el presentado en el artículo, representa un enfoque poderoso y flexible para resolver el problema de coloreo de grafos. Sus operadores específicos permiten reducir el espacio de búsqueda y encontrar soluciones óptimas de manera eficiente.