

Una implementación de la heurística Colonia de Abejas Artificiales a una instancia del problema de la 3-Partición: Tetris

José Ricardo Rodríguez Abreu

Universidad Nacional Autónoma de México

14 de noviembre de 2019

Tetris

Tetris es uno de los videojuegos más conocidos con más de 170 millones de copias vendidas.

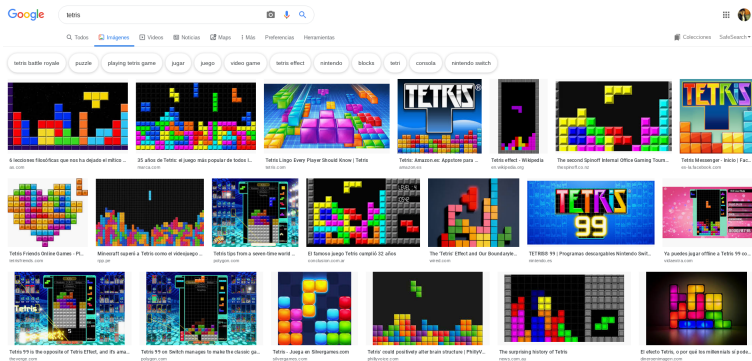


Figura: Búsqueda de imágenes de Tetris en Google.

Tetris en la ciencia

Tetris es un problema ampliamente estudiado por la comunidad científica en distintas áreas:

- Psicología
- Matemáticas
- Termodinámica
- Ciencias de la Computación

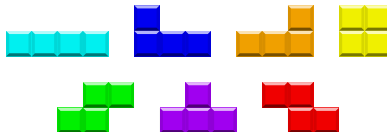


Figura: Las piezas de Tetris.

Tetris en la ciencia

Tetris es un problema ampliamente estudiado por la comunidad científica en distintas áreas:

- Psicología
- Matemáticas
- Termodinámica
- Ciencias de la Computación:
 - ▶ ¿Existe una estrategia para jugar en un tiempo indeterminado?
 - ▶ ¿Cuál es la mejor estrategia de juego?
 - ▶ ¿Existe alguna manera eficiente y automatizada de jugar Tetris?

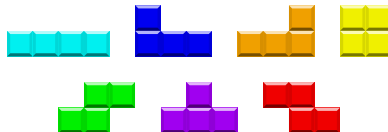


Figura: Las piezas de Tetris.

Tetris en la ciencia

Tetris es un problema ampliamente estudiado por la comunidad científica en distintas áreas:

- Psicología
- Matemáticas
- Termodinámica
- Ciencias de la Computación:
 - ▶ ¿Existe una estrategia para jugar en un tiempo indeterminado?
 - ▶ ¿Cuál es la mejor estrategia de juego?
 - ▶ ¿Existe alguna manera eficiente y automatizada de jugar Tetris?

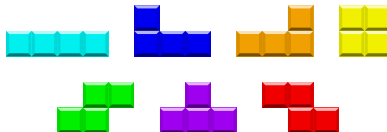


Figura: Las piezas de Tetris.

Clasificación de problemas

Primera clasificación:

- *Entscheidungsproblem* (El problema de decisión).

Clasificación de problemas

Primera clasificación:

- *Entscheidungsproblem* (El problema de decisión).

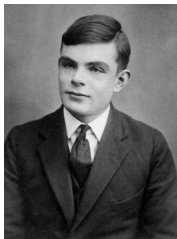


Figura: Alan Turing



Figura: Alonzo Church

Clasificación de problemas

Primera clasificación:

- *Entscheidungsproblem* (El problema de decisión).

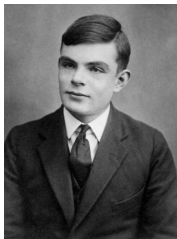


Figura: Alan Turing



Figura: Alonzo Church

- Medición de tiempo y espacio en función de la entrada.
 - ▶ Juris Hartmanis.
 - ▶ Richard Stearns.

Clasificación de problemas

Clases de complejidad:

- El lenguaje L está en $P \Leftrightarrow$ cumple que:
 - ▶ M corre en tiempo polinomial para toda entrada
 - ▶ $\forall x \in L, M(x) = TRUE \vee FALSE$.

Clasificación de problemas

Clases de complejidad:

- El lenguaje L está en $P \Leftrightarrow$ cumple que:
 - ▶ M corre en tiempo polinomial para toda entrada
 - ▶ $\forall x \in L, M(x) = TRUE \vee FALSE$.
- Un lenguaje $L \in NP \rightarrow \exists$ un polinomio $p : \mathbb{N} \rightarrow \mathbb{N}$ y una MT M tal que $\forall x \in \{0, 1\}^*$,

$$x \in L \Leftrightarrow \exists u \in \{0, 1\}^{p(|x|)}, \text{ tal que } M(x, u) = 1.$$

Clasificación de problemas

Clases de complejidad:

- El lenguaje L está en $P \Leftrightarrow$ cumple que:
 - ▶ M corre en tiempo polinomial para toda entrada
 - ▶ $\forall x \in L, M(x) = TRUE \vee FALSE$.
- Un lenguaje $L \in NP \rightarrow \exists$ un polinomio $p : \mathbb{N} \rightarrow \mathbb{N}$ y una MT M tal que $\forall x \in \{0, 1\}^*$,

$$x \in L \Leftrightarrow \exists u \in \{0, 1\}^{p(|x|)}, \text{ tal que } M(x, u) = 1.$$

- Se dice que L' es NP -duro si $L \leq_p L'$ para cada $L \in NP$.

Clasificación de problemas

Clases de complejidad:

- El lenguaje L está en $P \Leftrightarrow$ cumple que:
 - ▶ M corre en tiempo polinomial para toda entrada
 - ▶ $\forall x \in L, M(x) = TRUE \vee FALSE$.
- Un lenguaje $L \in NP \rightarrow \exists$ un polinomio $p : \mathbb{N} \rightarrow \mathbb{N}$ y una MT M tal que $\forall x \in \{0, 1\}^*$,

$$x \in L \Leftrightarrow \exists u \in \{0, 1\}^{p(|x|)}, \text{ tal que } M(x, u) = 1.$$

- Se dice que L' es NP -duro si $L \leq_p L'$ para cada $L \in NP$.
- Se dice que L' es NP -completo si L' es NP -duro y $L' \in NP$.

Tipo de soluciones

Algoritmo:

Definición (usada en este trabajo)

Un algoritmo es cualquier procedimiento bien definido que toma algún valor, o conjunto de valores como entrada (o **Input**) y produce algún valor, o conjunto de valores como salida (u **Output**). Un algoritmo siempre termina.

Tipo de soluciones

Algoritmo:

Definición (usada en este trabajo)

Un algoritmo es cualquier procedimiento bien definido que toma algún valor, o conjunto de valores como entrada (o **Input**) y produce algún valor, o conjunto de valores como salida (u **Output**). Un algoritmo siempre termina.

Ejemplos:

- Suma de los primeros $n \in \mathbb{N}$ ($O(n)$).
- Mergesort ($O(n \log(n))$).
- Algoritmo de Dijkstra ($O(|E| + |V| \log(|V|))$).

Tipo de soluciones

Algoritmo:

Definición (usada en este trabajo)

Un algoritmo es cualquier procedimiento bien definido que toma algún valor, o conjunto de valores como entrada (o **Input**) y produce algún valor, o conjunto de valores como salida (u **Output**). Un algoritmo siempre termina.

Ejemplos:

- Suma de los primeros $n \in \mathbb{N}$ ($O(n)$).
- Mergesort ($O(n \log(n))$).
- Algoritmo de Dijkstra ($O(|E| + |V| \log(|V|))$).
- Ventaja: Garantía de resultado.
- Desventaja: Tiempo poco realista en problemas $\notin P$.

Algoritmos fuera de la clase P

3-SAT

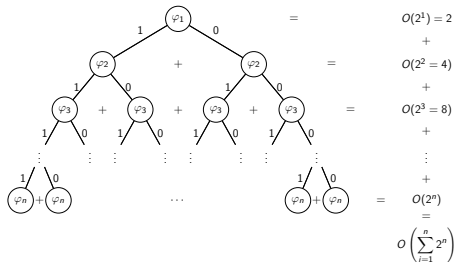
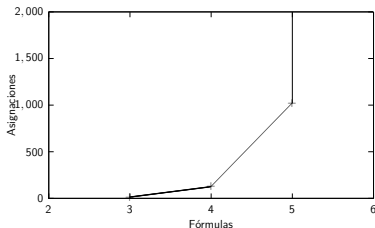
Dado un conjunto de fórmulas $\phi = \{x_1, x_2, x_3, \dots, x_n\}$ con cada x_i una fórmula lógica de la forma $x_i = p_i \vee q_i \vee r_i$, con p_i, q_i, r_i , variables o términos lógicos, se deberá encontrar una interpretación \mathcal{I} tal que $\mathcal{I}(\phi) = \mathcal{I}(x_i) = 1 \ \forall i \in \{1, 2, \dots, n\}$.

Algoritmos fuera de la clase P

3-SAT

Dado un conjunto de fórmulas $\phi = \{x_1, x_2, x_3, \dots, x_n\}$ con cada x_i una fórmula lógica de la forma $x_i = p_i \vee q_i \vee r_i$, con p_i, q_i, r_i , variables o términos lógicos, se deberá encontrar una interpretación \mathcal{I} tal que $\mathcal{I}(\phi) = \mathcal{I}(x_i) = 1 \ \forall i \in \{1, 2, \dots, n\}$.

¿Qué pasa si aplicamos un algoritmo para encontrar la solución de un problema como $3\text{-SAT} \in NP\text{-completo}$?



Tipo de soluciones

Heurística:

Definición

Las heurísticas, al igual que los algoritmos, también son una secuencia de pasos bien definida con la diferencia de que dada una entrada I , produce una salida $o \in \text{OUTPUT}$ con OUTPUT un conjunto de posibles valores de solución. Durante la ejecución de una heurística, la condición de término es más ambigua.

Tipo de soluciones

Heurística:

Definición

Las heurísticas, al igual que los algoritmos, también son una secuencia de pasos bien definida con la diferencia de que dada una entrada I , produce una salida $o \in \text{OUTPUT}$ con OUTPUT un conjunto de posibles valores de solución. Durante la ejecución de una heurística, la condición de término es más ambigua.

Ejemplos:

- Heurísticas usadas para optimización de redes neuronales.
- Búsqueda tabú.
- Algoritmos (heurísticos) genéticos.

Tipo de soluciones

Heurística:

Definición

Las heurísticas, al igual que los algoritmos, también son una secuencia de pasos bien definida con la diferencia de que dada una entrada I , produce una salida $o \in \text{OUTPUT}$ con OUTPUT un conjunto de posibles valores de solución. Durante la ejecución de una heurística, la condición de término es más ambigua.

Ejemplos:

- Heurísticas usadas para optimización de redes neuronales.
 - Búsqueda tabú.
 - Algoritmos (heurísticos) genéticos.
-
- Ventaja: Reducción significativa en tiempo de respuesta.
 - Desventaja: Desempeño pobre en el peor de los casos.

Colonia de abejas artificiales

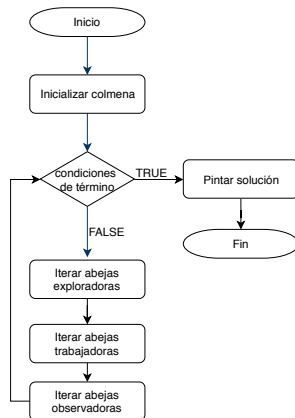
- ¿Agentes involucrados?

Colonia de abejas artificiales

- ¿Agentes involucrados?
 - ▶ Abejas exploradoras.
 - ▶ Abejas trabajadoras.
 - ▶ Abejas observadoras.

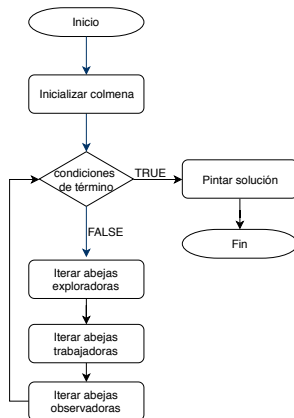
Colonia de abejas artificiales

- ¿Agentes involucrados?
 - ▶ Abejas exploradoras.
 - ▶ Abejas trabajadoras.
 - ▶ Abejas observadoras.
- Comportamiento colectivo.



Colonia de abejas artificiales

- ¿Agentes involucrados?
 - ▶ Abejas exploradoras.
 - ▶ Abejas trabajadoras.
 - ▶ Abejas observadoras.
- Comportamiento colectivo.
- Recomendaciones basadas en resultados de experimentación del autor Dervis Karaboga.



Justificación de aplicación de la heurística

...

- Psicología
- Matemáticas
- Termodinámica
- Ciencias de la Computación:
 - ▶ ¿Existe una estrategia para jugar en un tiempo indeterminado?
 - ▶ ¿Cuál es la mejor estrategia de juego?
 - ▶ ¿Existe alguna manera eficiente y automatizada de jugar Tetris?

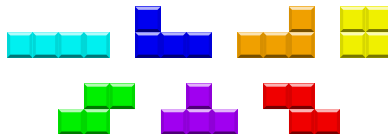


Figura: Las piezas de Tetris.

Justificación de aplicación de la heurística

...

- Psicología
- Matemáticas
- Termodinámica
- Ciencias de la Computación:
 - ▶ ¿Existe una estrategia para jugar en un tiempo indeterminado?
 - ▶ ¿Cuál es la mejor estrategia de juego?
 - ▶ ¿Existe alguna manera eficiente y automatizada de jugar Tetris?
 - ▶ ¿A qué clase (*P*, *NP*, *NP-duro* o *NP-completo*) pertenece Tetris?

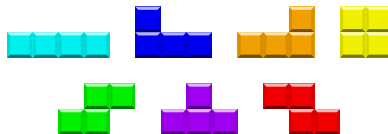


Figura: Las piezas de Tetris.

Clasificación de TETRIS

Clasificación de TETRIS

- Formalización del juego:

Clasificación de TETRIS

- Formalización del juego:
 - ▶ Tablero: matriz de n por m filas.

Clasificación de TETRIS

- Formalización del juego:
 - ▶ Tablero: matriz de n por m filas.
 - ▶ Piezas: $P = \langle t, o, \langle i, j \rangle, f \rangle$.

Clasificación de TETRIS

- Formalización del juego:
 - ▶ Tablero: matriz de n por m filas.
 - ▶ Piezas: $P = \langle t, o, \langle i, j \rangle, f \rangle$.
 - ▶ Rotación: $R : \langle P, \theta, B \rangle \mapsto P'$.

Clasificación de TETRIS

- Formalización del juego:
 - ▶ Tablero: matriz de n por m filas.
 - ▶ Piezas: $P = \langle t, o, \langle i, j \rangle, f \rangle$.
 - ▶ Rotación: $R : \langle P, \theta, B \rangle \mapsto P'$.
 - ▶ Reglas del juego:

Clasificación de TETRIS

- Formalización del juego:

- ▶ Tablero: matriz de n por m filas.
- ▶ Piezas: $P = \langle t, o, \langle i, j \rangle, f \rangle$.
- ▶ Rotación: $R : \langle P, \theta, B \rangle \mapsto P'$.
- ▶ Reglas del juego:
 - ★ $R(P, 90^\circ, B)$.
 - ★ $R(P, -90^\circ, B)$.
 - ★ $P' = \langle t, o, \langle i - 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i + 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j - 1 \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j \rangle, \text{FIJO} \rangle$.

Clasificación de TETRIS

- Formalización del juego:

- ▶ Tablero: matriz de n por m filas.
- ▶ Piezas: $P = \langle t, o, \langle i, j \rangle, f \rangle$.
- ▶ Rotación: $R : \langle P, \theta, B \rangle \mapsto P'$.
- ▶ Reglas del juego:
 - ★ $R(P, 90^\circ, B)$.
 - ★ $R(P, -90^\circ, B)$.
 - ★ $P' = \langle t, o, \langle i - 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i + 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j - 1 \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j \rangle, \text{FIJO} \rangle$.

- TETRIS:

- ▶ **Input:** Un juego de Tetris de la forma $\mathcal{G} = \langle B, P_1, P_2, \dots, P_p \rangle$.
- ▶ **Output:** ¿Existe la secuencia de trayectorias Σ tal que $\Phi(\mathcal{G}, \Sigma)$ no resulte en una partida perdida?

Clasificación de TETRIS

- Formalización del juego:

- ▶ Tablero: matriz de n por m filas.
- ▶ Piezas: $P = \langle t, o, \langle i, j \rangle, f \rangle$.
- ▶ Rotación: $R : \langle P, \theta, B \rangle \mapsto P'$.
- ▶ Reglas del juego:
 - ★ $R(P, 90^\circ, B)$.
 - ★ $R(P, -90^\circ, B)$.
 - ★ $P' = \langle t, o, \langle i - 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i + 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j - 1 \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j \rangle, \text{FIJO} \rangle$.

- TETRIS:

- ▶ **Input:** Un juego de Tetris de la forma $\mathcal{G} = \langle B, P_1, P_2, \dots, P_p \rangle$.
- ▶ **Output:** ¿Existe la secuencia de trayectorias Σ tal que $\Phi(\mathcal{G}, \Sigma)$ no resulte en una partida perdida?

- Demostración: ¿Es TETRIS *NP*? ¿Es TETRIS *NP-duro*?

Clasificación de TETRIS

- Formalización del juego:

- ▶ Tablero: matriz de n por m filas.
- ▶ Piezas: $P = \langle t, o, \langle i, j \rangle, f \rangle$.
- ▶ Rotación: $R : \langle P, \theta, B \rangle \mapsto P'$.
- ▶ Reglas del juego:
 - ★ $R(P, 90^\circ, B)$.
 - ★ $R(P, -90^\circ, B)$.
 - ★ $P' = \langle t, o, \langle i - 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i + 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j - 1 \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j \rangle, \text{FIJO} \rangle$.

- TETRIS:

- ▶ **Input:** Un juego de Tetris de la forma $\mathcal{G} = \langle B, P_1, P_2, \dots, P_p \rangle$.
- ▶ **Output:** ¿Existe la secuencia de trayectorias Σ tal que $\Phi(\mathcal{G}, \Sigma)$ no resulte en una partida perdida?

- Demostración: ¿Es TETRIS NP? ¿Es TETRIS NP-duro?

Clasificación de TETRIS

- Formalización del juego:

- ▶ Tablero: matriz de n por m filas.
- ▶ Piezas: $P = \langle t, o, \langle i, j \rangle, f \rangle$.
- ▶ Rotación: $R : \langle P, \theta, B \rangle \mapsto P'$.
- ▶ Reglas del juego:
 - ★ $R(P, 90^\circ, B)$.
 - ★ $R(P, -90^\circ, B)$.
 - ★ $P' = \langle t, o, \langle i - 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i + 1, j \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j - 1 \rangle, \text{MOVIBLE} \rangle$.
 - ★ $P' = \langle t, o, \langle i, j \rangle, \text{FIJO} \rangle$.

- TETRIS:

- ▶ **Input:** Un juego de Tetris de la forma $\mathcal{G} = \langle B, P_1, P_2, \dots, P_p \rangle$.
- ▶ **Output:** ¿Existe la secuencia de trayectorias Σ tal que $\Phi(\mathcal{G}, \Sigma)$ no resulte en una partida perdida?

- Demostración: ¿Es TETRIS NP ? ¿Es TETRIS NP -duro? \longrightarrow Es TETRIS NP -completo.









Diseño de la implementación

Diseño de la implementación

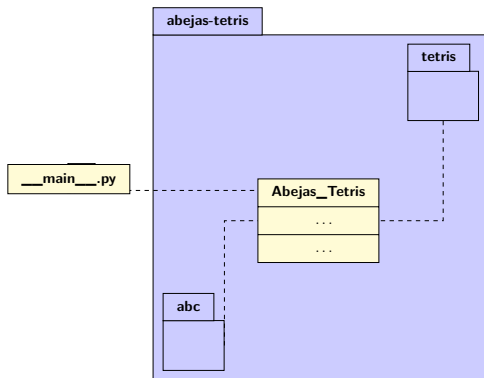
- Análisis del sistema.

Diseño de la implementación

- Análisis del sistema.
 - ▶ Mantener modelo apegado a la formalización del juego.
 - ▶ Desarrollar la heurística descrita por Dervis Karaboga.
 - ▶ Comunicación de la heurística con el juego de Tetris.

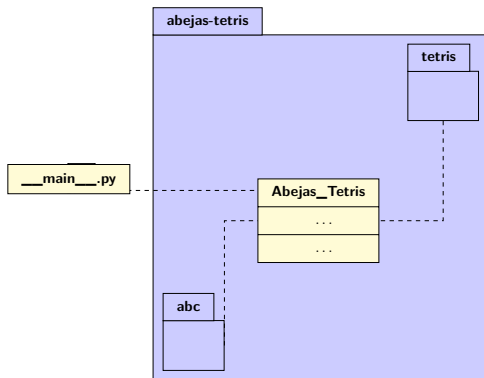
Diseño de la implementación

- Análisis del sistema.
 - ▶ Mantener modelo apegado a la formalización del juego.
 - ▶ Desarrollar la heurística descrita por Dervis Karaboga.
 - ▶ Comunicación de la heurística con el juego de Tetris.



Diseño de la implementación

- Análisis del sistema.
 - ▶ Mantener modelo apegado a la formalización del juego.
 - ▶ Desarrollar la heurística descrita por Dervis Karaboga.
 - ▶ Comunicación de la heurística con el juego de Tetris.



Orientación a Objetos

Diseño de clases

Tetris:

- Punto.
- Tipo.
- Casilla.
- Pieza.
- Tablero.
- Tetris.

Diseño de clases

Tetris:

- Punto.
- Tipo.
- Casilla.
- Pieza.
- Tablero.
- Tetris.

ABC:

Diseño de clases

Tetris:

- Punto.
- Tipo.
- Casilla.
- Pieza.
- Tablero.
- Tetris.

ABC:

- Tipo_Abeja.
- Abeja:
- Colmena.

Tetris:

- Punto.
- Tipo.
- Casilla.
- Pieza.
- Tablero.
- Tetris.

ABC:

- Tipo_Abeja.
- Colmena.
- Abeja:
 - ▶ Exploradoras.
 - ▶ Trabajadoras.
 - ▶ Observadoras.

Tetris:

- Punto.
- Tipo.
- Casilla.
- Pieza.
- Tablero.
- Tetris.

ABC:

- Tipo_Abeja.
- Colmena.
- Abeja:
 - ▶ Exploradoras.
 - ▶ Trabajadoras.
 - ▶ Observadoras.

Abejas observadoras

- ¿Qué hacen exactamente las abejas observadoras en ABC?

Abejas observadoras

- ¿Qué hacen exactamente las abejas observadoras en ABC?
 - ▶ Seleccionan una fuente i :

$$P_i = \frac{F(\theta_i)}{\sum_{k=1}^n F(\theta_k)}.$$

Abejas observadoras

- ¿Qué hacen exactamente las abejas observadoras en ABC?
 - ▶ Seleccionan una fuente i :

$$P_i = \frac{F(\theta_i)}{\sum_{k=1}^n F(\theta_k)}.$$

- ▶ Buscan una fuente vecina a i :

$$\theta_i(c+1) = \theta_i(c) \pm \phi_i(c).$$

Abejas observadoras

- ¿Qué hacen exactamente las abejas observadoras en ABC?
 - ▶ Seleccionan una fuente i :

$$P_i = \frac{F(\theta_i)}{\sum_{k=1}^n F(\theta_k)}.$$

- ▶ Buscan una fuente vecina a i :

$$\theta_i(c+1) = \theta_i(c) \pm \phi_i(c).$$

- ¿Cómo se proyecta una fuente en Tetris?

Abejas observadoras

- ¿Qué hacen exactamente las abejas observadoras en ABC?
 - ▶ Seleccionan una fuente i :

$$P_i = \frac{F(\theta_i)}{\sum_{k=1}^n F(\theta_k)}.$$

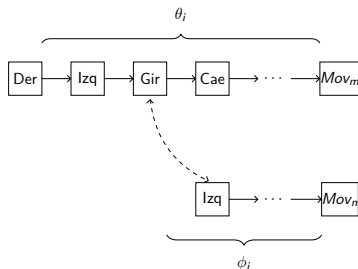
- ▶ Buscan una fuente vecina a i :

$$\theta_i(c+1) = \theta_i(c) \pm \phi_i(c).$$

- ¿Cómo se proyecta una fuente en Tetris? R= Para un juego $\langle B_0, P_1, \dots, P_p \rangle$, una secuencia de trayectorias Σ es una secuencia $B_0, \sigma_1, B_1, \dots, \sigma_p, B_p$ tal que para cada i , la trayectoria de la pieza P_i aplicado al tablero B_{i-1} , genera el tablero B_i .

Abejas observadoras

- ¿Cómo se *mueven* las abejas *cerca* de una fuente?



Funciones de las abejas

- Abeja exploradora.
 - ▶ `_busca_fuente`: Copia la mejor fuente reportada en la colmena y genera un nuevo tablero B_n a partir de una trayectoria σ_{n-1} . B_n es ahora su fuente asociada.

Funciones de las abejas

- Abeja exploradora.
 - ▶ `_busca_fuente`: Copia la mejor fuente reportada en la colmena y genera un nuevo tablero B_n a partir de una trayectoria σ_{n-1} . B_n es ahora su fuente asociada.
- Abeja trabajadora.
 - ▶ `_explotar`: Genera un σ_n sobre su tablero asociado y lo aplica para obtener un tablero nuevo B_{n+1} .

Funciones de las abejas

- Abeja exploradora.
 - ▶ `_busca_fuente`: Copia la mejor fuente reportada en la colmena y genera un nuevo tablero B_n a partir de una trayectoria σ_{n-1} . B_n es ahora su fuente asociada.
- Abeja trabajadora.
 - ▶ `_explotar`: Genera un σ_n sobre su tablero asociado y lo aplica para obtener un tablero nuevo B_{n+1} .
- Abeja observadora.
 - ▶ `_observadoras`:
 - 1 Se le asigna una fuente de alguna abeja trabajadora. Todas las abejas observadoras tienen que poseer una fuente asociada.
 - 2 Corta parcialmente la ultima trayectoria σ y la completa hasta encontrar una fuente “cercana”.
 - 3 Si la fuente cercana es *mejor* a la fuente original, avisa a la colmena.

Funciones de las abejas

- Abeja exploradora.
 - ▶ `_busca_fuente`: Copia la mejor fuente reportada en la colmena y genera un nuevo tablero B_n a partir de una trayectoria σ_{n-1} . B_n es ahora su fuente asociada.
- Abeja trabajadora.
 - ▶ `_explotar`: Genera un σ_n sobre su tablero asociado y lo aplica para obtener un tablero nuevo B_{n+1} .
- Abeja observadora.
 - ▶ `_observadoras`:
 - 1 Se le asigna una fuente de alguna abeja trabajadora. Todas las abejas observadoras tienen que poseer una fuente asociada.
 - 2 Corta parcialmente la ultima trayectoria σ y la completa hasta encontrar una fuente “cercana”.
 - 3 Si la fuente cercana es *mejor* a la fuente original, avisa a la colmena.
- Todas las abejas.
 - ▶ `_nectar`: Función de costo o función *fitness*.

Función de costo

- ¿Cuál es el objetivo de la función de costo?

Función de costo

- ¿Cuál es el objetivo de la función de costo?
 - ▶ Primer objetivo: que la heurística elimine $n \geq 10$ filas.

Función de costo

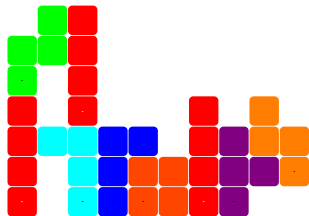
- ¿Cuál es el objetivo de la función de costo?
 - ▶ Primer objetivo: que la heurística elimine $n \geq 10$ filas.
 - ▶ Objetivo principal: eliminar $n \geq 20$ filas.

Función de costo

- ¿Cuál es el objetivo de la función de costo?
 - ▶ Primer objetivo: que la heurística elimine $n \geq 10$ filas.
 - ▶ Objetivo principal: eliminar $n \geq 20$ filas.
- ¿Cómo se logra orientar a la heurística hacia el objetivo?

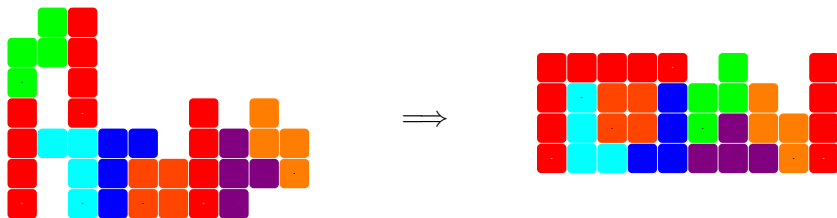
Función de costo

- ¿Cuál es el objetivo de la función de costo?
 - ▶ Primer objetivo: que la heurística elimine $n \geq 10$ filas.
 - ▶ Objetivo principal: eliminar $n \geq 20$ filas.
- ¿Cómo se logra orientar a la heurística hacia el objetivo? Suponer la siguiente entrada: $L = \{I, Rg, Lg, Sq, I, T, Ls, Rs, I\}$.



Función de costo

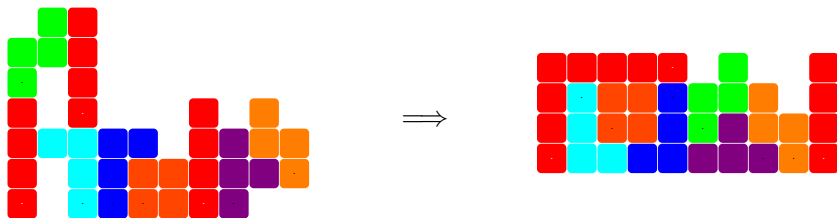
- ¿Cuál es el objetivo de la función de costo?
 - ▶ Primer objetivo: que la heurística elimine $n \geq 10$ filas.
 - ▶ Objetivo principal: eliminar $n \geq 20$ filas.
- ¿Cómo se logra orientar a la heurística hacia el objetivo? Suponer la siguiente entrada: $L = \{I, Rg, Lg, Sq, I, T, Ls, Rs, I\}$.



- ¿Cómo se sabe que una función de costo funciona acorde a los requerimientos?

Función de costo

- ¿Cuál es el objetivo de la función de costo?
 - ▶ Primer objetivo: que la heurística elimine $n \geq 10$ filas.
 - ▶ Objetivo principal: eliminar $n \geq 20$ filas.
- ¿Cómo se logra orientar a la heurística hacia el objetivo? Suponer la siguiente entrada: $L = \{I, Rg, Lg, Sq, I, T, Ls, Rs, I\}$.



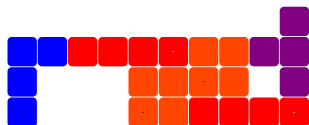
- ¿Cómo se sabe que una función de costo funciona acorde a los requerimientos? **Experimentación.**

Filas entre pesos negativos

$$f = \frac{1 + \text{filas-removidas}}{\text{horizontalidad} + \text{atrapados} + \text{cubiertos} + \text{altura}}.$$

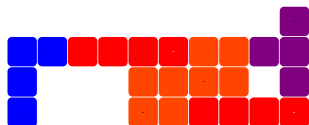
Filas entre pesos negativos

$$f = \frac{1 + \text{filas-removidas}}{\text{horizontalidad} + \text{atrapados} + \text{cubiertos} + \text{altura}}.$$



Filas entre pesos negativos

$$f = \frac{1 + \text{filas-removidas}}{\text{horizontalidad} + \text{atrapados} + \text{cubiertos} + \text{altura}}.$$

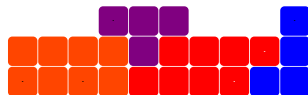
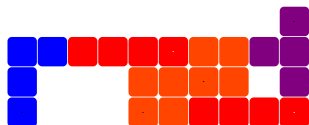


- $\text{filas-removidas} = 0$
- $\text{horizontalidad} = 1$
- $\text{atrapados} = 1$
- $\text{cubiertos} = 7$
- $\text{altura} = 4$

$$\frac{1 + 0}{1 + 1 + 7 + 4} = \frac{1}{13} \approx 0.077.$$

Filas entre pesos negativos

$$f = \frac{1 + \text{filas-removidas}}{\text{horizontalidad} + \text{atrapados} + \text{cubiertos} + \text{altura}}.$$

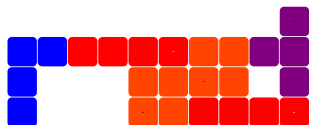


- filas-removidas = 0
 - horizontalidad = 1
 - atrapados = 1
 - cubiertos = 7
 - altura = 4
- VS.

$$\frac{1 + 0}{1 + 1 + 7 + 4} = \frac{1}{13} \approx 0.077.$$

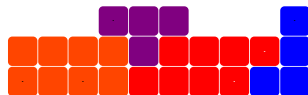
Filas entre pesos negativos

$$f = \frac{1 + \text{filas-removidas}}{\text{horizontalidad} + \text{atrapados} + \text{cubiertos} + \text{altura}}.$$



- filas-removidas = 0
- horizontalidad = 1
- atrapados = 1
- cubiertos = 7
- altura = 4

$$\frac{1 + 0}{1 + 1 + 7 + 4} = \frac{1}{13} \approx 0.077.$$



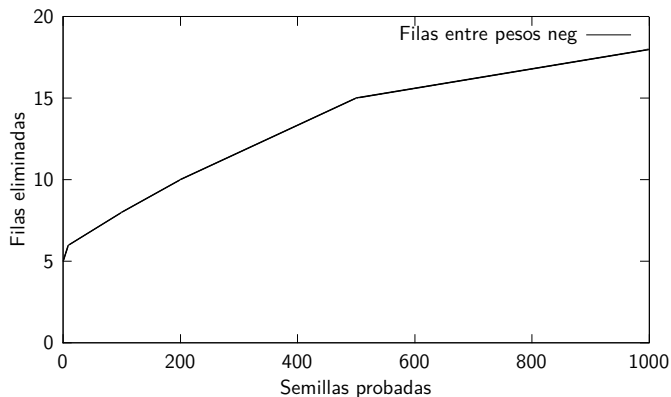
- filas-removidas = 2
- horizontalidad = 1
- atrapados = 0
- cubiertos = 0
- altura = 4

$$\frac{1 + (2 \times c)}{1 + 0 + 0 + 1} = \frac{1 + 4}{2} = \frac{5}{2} = 2.5.$$

vs.

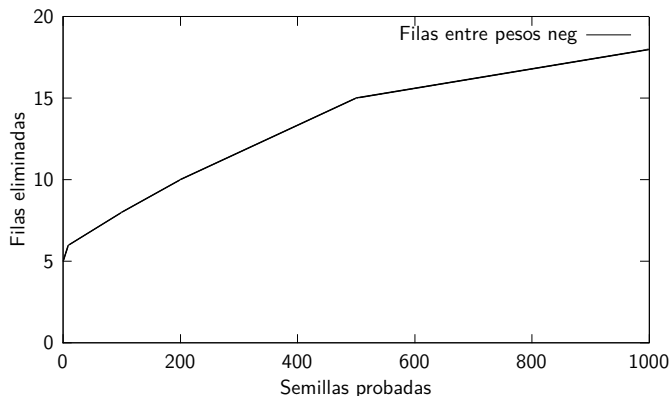
Resultados filas entre pesos negativos

En una colmena de 100 abejas, los resultados fueron los siguientes:



Resultados filas entre pesos negativos

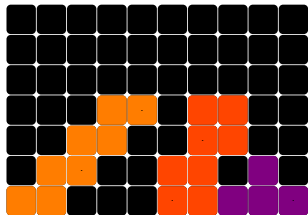
En una colmena de 100 abejas, los resultados fueron los siguientes:



- Se obtuvieron como mejor solución 18 filas removidas.

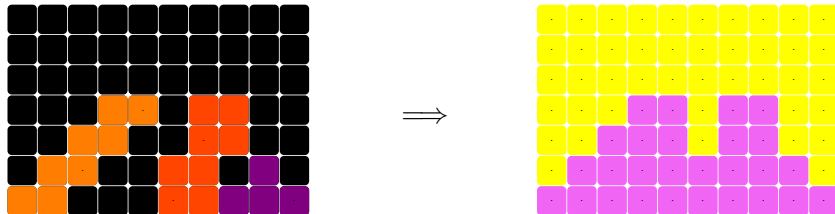
Raining skyline

- Optimización de casillas disponibles. ¿En qué consiste?



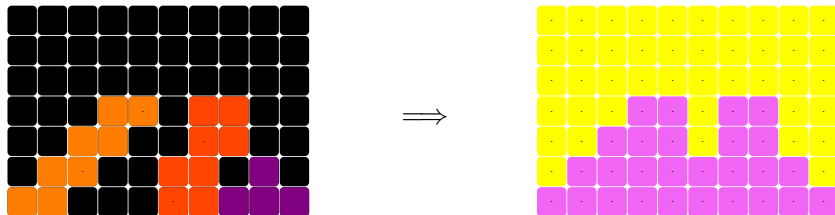
Raining skyline

- Optimización de casillas disponibles. ¿En qué consiste?



Raining skyline

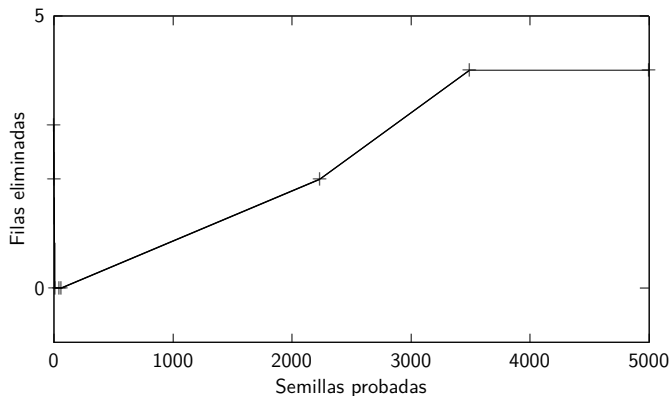
- Optimización de casillas disponibles. ¿En qué consiste?



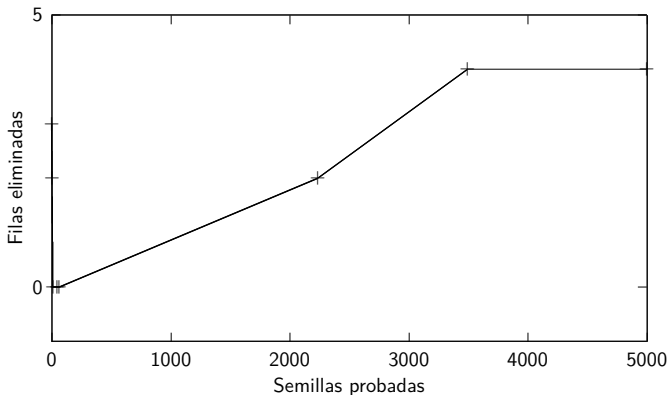
$$rs = \sum_{i=1, j=1}^{ancho, alto} j \mid (i, j) \in C_{sup}.$$

$$f(T) = \frac{n - rs}{n}.$$

Resultados de *Raining skyline* sin ponderar



Resultados de *Raining skyline* sin ponderar



- ¿Por qué los resultados no son los esperados?
- ¿Por qué puede decrecer el número de filas eliminadas en una “mejor” semilla?

Ponderación de *Raining skyline*

1

$$rs = \sum_{i=1, j=1}^{ancho, alto} j \mid j > b \wedge (i, b) \in C_{sup}$$

$$f(T) = \frac{rs}{\sum_{i=1}^{alto} (ancho \times i)}.$$

Ponderación de *Raining skyline*

1

$$rs = \sum_{i=1, j=1}^{ancho, alto} j \mid j > b \wedge (i, b) \in C_{sup}$$

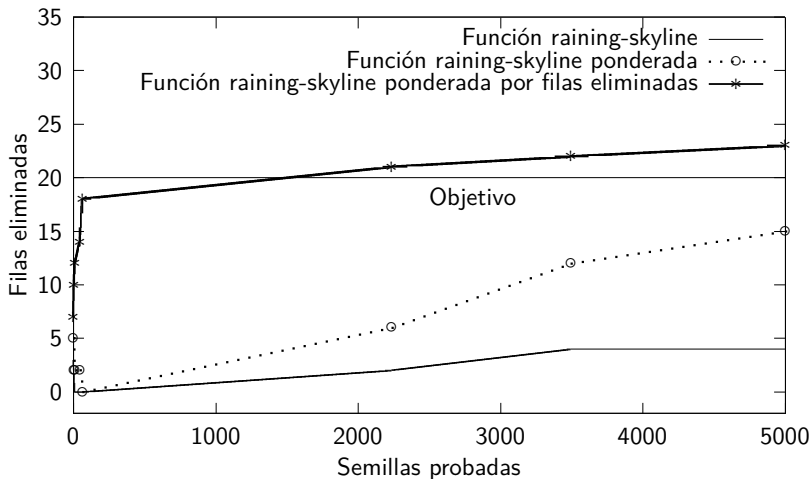
$$f(T) = \frac{rs}{\sum_{i=1}^{alto} (ancho \times i)}.$$

2

$$f(T) = \left(\frac{rs}{\sum_{i=1}^{alto} (ancho \times i)} \right) \times (1 + fe).$$

Resultados de *Raining skyline* ponderada

En una colmena de 100 abejas, los resultados fueron los siguientes:



Observaciones adicionales

Observaciones adicionales

- Tamaño de la colmena se mantuvo en 100 abejas.

Observaciones adicionales

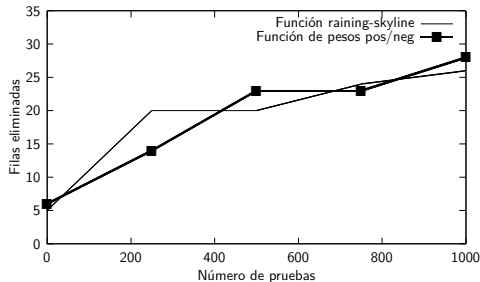
- Tamaño de la colmena se mantuvo en 100 abejas.
- Cada función fue probada con al menos 10,000 semillas pseudoaleatorias.

Observaciones adicionales

- Tamaño de la colmena se mantuvo en 100 abejas.
- Cada función fue probada con al menos 10,000 semillas pseudoaleatorias.
- Se consiguieron muestras de números mayores al objetivo (71 filas eliminadas).

Observaciones adicionales

- Tamaño de la colmena se mantuvo en 100 abejas.
- Cada función fue probada con al menos 10,000 semillas pseudoaleatorias.
- Se consiguieron muestras de números mayores al objetivo (71 filas eliminadas).
- Los resultados de ambas funciones arrojan muestras muy similares.



Conclusiones

Conclusiones

- La meta propuesta fue alcanzada exitosamente.

Conclusiones

- La meta propuesta fue alcanzada exitosamente.
- El comportamiento de la heurística aplicada a Tetris particularmente parece responder positivamente.

Conclusiones

- La meta propuesta fue alcanzada exitosamente.
- El comportamiento de la heurística aplicada a Tetris particularmente parece responder positivamente.
- Los resultados obtenidos superan las expectativas.

Conclusiones

- Sin importar que el problema de Tetris sea un problema NP -completo, existen métodos prácticos que pueden encontrar soluciones de buena calidad.

Conclusiones

- Sin importar que el problema de Tetris sea un problema NP -completo, existen métodos prácticos que pueden encontrar soluciones de buena calidad.
- La heurística ABC es un buen método de resolución de problemas ya que la función de costo de la que depende sus resultados es independiente al comportamiento del colectivo en la colmena.

Ejecución del código.