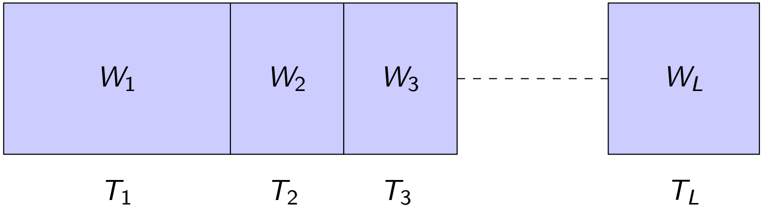
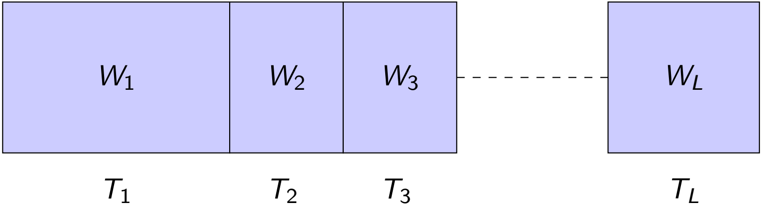
Heurísticos con métodos algebraicos

María Isabel Hartillo Hermoso

Sanlúcar de Barrameda, 30 de Noviembre


W_1
W_2
W_3
W_L
T_1
T_2
T_3
T_L



Lisnianski et al. definen LOLP (Loss Of Load Probability) como:

$$LOLP = \frac{\sum_{l=1}^{L} \Pr(G_{sys} < W_l) T_l}{\sum_{l=1}^{L} T_l}$$

- Vamos a considerar sistemas en serie paralelos

- Vamos a considerar sistemas en serie paralelos
- Las componentes sólo tienen dos estados, en funcionamiento o fuera de funcionamiento

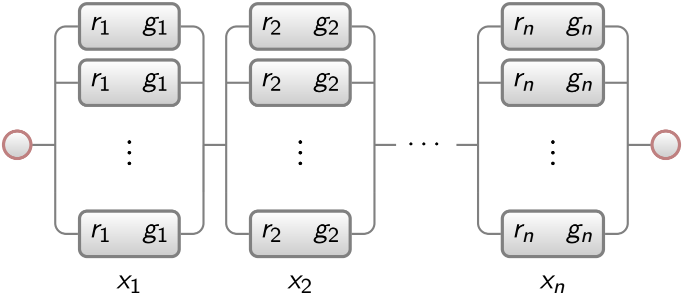
- Vamos a considerar sistemas en serie paralelos
- Las componentes sólo tienen dos estados, en funcionamiento o fuera de funcionamiento
- Las componentes tienen un coste, fiabilidad y capacidad conocidos

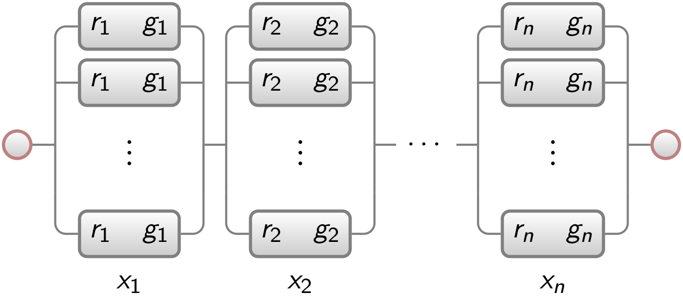
- Vamos a considerar sistemas en serie paralelos
- Las componentes sólo tienen dos estados, en funcionamiento o fuera de funcionamiento
- Las componentes tienen un coste, fiabilidad y capacidad conocidos
- Las componentes fuera de funcionamiento no dañan o afectan al resto del sistema y no se reparan

- Vamos a considerar sistemas en serie paralelos
- Las componentes sólo tienen dos estados, en funcionamiento o fuera de funcionamiento
- Las componentes tienen un coste, fiabilidad y capacidad conocidos
- Las componentes fuera de funcionamiento no dañan o afectan al resto del sistema y no se reparan

- Vamos a considerar sistemas en serie paralelos
- Las componentes sólo tienen dos estados, en funcionamiento o fuera de funcionamiento
- Las componentes tienen un coste, fiabilidad y capacidad conocidos
- Las componentes fuera de funcionamiento no dañan o afectan al resto del sistema y no se reparan

Vamos a considerar componentes de un sistema con una **fiabilidad** r_i y una **capacidad** g_i


r_1 g_1
r_1 g_1
r_1 g_1
r_2 g_2
r_2 g_2
r_2 g_2
r_n g_n
r_n g_n
r_n g_n
x_1
x_2
x_n



Para que un sistema en serie alcance la demanda requerida W_l, cada subsistema debe alcanzar dicha capacidad, por tanto:

$$\Pr(G_{sys} \geq W_l) = \prod_{i=1}^{n} \Pr(G_i \geq W_l)$$

$$E(\mathbf{x}) = 1 - LOLP$$

$$E(\mathbf{x}) = 1 - LOLP$$

Si notamos

$$T_l' = \frac{T_l}{\sum_{l=1}^{L} T_l}$$

puesto que

$$1 - \Pr(G_{sys} < W_l) = \Pr(G_{sys} \geq W_l)$$

$$E(\mathbf{x}) = 1 - LOLP$$

Si notamos

$$T'_l = \frac{T_l}{\sum_{l=1}^{L} T_l}$$

puesto que

$$1 - \Pr(G_{sys} < W_l) = \Pr(G_{sys} \geq W_l)$$

$$E(\mathbf{x}) = \sum_{l=1}^{L} T'_l \prod_{i=1}^{n} \Pr(G_i \geq W_l)$$

- La fiabilidad de subsistemas en paralelo con capacidad con componentes redundantes heterogéneas se puede modelar con una distribución binomial.

- La fiabilidad de subsistemas en paralelo con capacidad con componentes redundantes heterogéneas se puede modelar con una distribución binomial.
- Supongamos que el subsistema i tiene x_{ij} componentes en paralelo, todas del mismo tipo j. Si $W_l \leq g_{ij}$, entonces $\Pr(G_i \geq W_l)$ es simplemente la fiabilidad del sistema paralelo, ya que el subsistema fallaría si todas las componentes fallasen.

- La fiabilidad de subsistemas en paralelo con capacidad con componentes redundantes heterogéneas se puede modelar con una distribución binomial.
- Supongamos que el subsistema i tiene x_{ij} componentes en paralelo, todas del mismo tipo j. Si $W_l \leq g_{ij}$, entonces $\Pr(G_i \geq W_l)$ es simplemente la fiabilidad del sistema paralelo, ya que el subsistema fallaría si todas las componentes fallasen.
- Si $(x_{ij} - 1)g_{ij} < W_l \leq x_{ij}g_{ij}$, estamos en el caso de un sistema en serie, ya que necesitamos que todas las componentes estén en funcionamiento para alcanzar la capacidad requerida.

En general, podemos notar:

$$\Pr(G_i \geq W_l) = 1 - \overline{\Pr}(x_{ij})$$

donde

$$\overline{\Pr}(x_{ij}) = \begin{cases} (1 - r_{ij})^{x_{ij}} & \text{Si } W_l \leq g_{ij} \\ \mathrm{Bin}(x_{ij}, 1, r_{ij}) & \text{Si } g_{ij} < W_l \leq 2g_{ij} \\ \ldots & \ldots \\ \mathrm{Bin}(x_{ij}, p-1, r_{ij}) & \text{Si } (p-1)g_{ij} < W_l \leq pg_{ij} \\ \ldots & \ldots \\ 1 - r_{ij}^{x_{ij}} & \text{Si } (x_{ij}-1)g_{ij} < W_l \leq x_{ij}g_{ij} \\ 1 & \text{En otro caso} \end{cases}$$

$\mathrm{Bin}(x_{ij}, p, r_{ij})$ denota la probabilidad de que haya a lo más p componentes supervivientes de entre x_{ij}, así:

$$\mathrm{Bin}(x_{ij}, p, r_{ij}) = \sum_{t=0}^{p} \mathrm{bino}(x_{ij}, t, r_{ij})$$

con

$$\mathrm{bino}(x_{ij}, t, r_{ij}) = \binom{x_{ij}}{t} r_{ij}^{t}(1 - r_{ij})^{x_{ij}-t}$$

$\mathrm{Bin}(x_{ij}, p, r_{ij})$ denota la probabilidad de que haya a lo más p componentes supervivientes de entre x_{ij}, así:

$$\mathrm{Bin}(x_{ij}, p, r_{ij}) = \sum_{t=0}^{p} \mathrm{bino}(x_{ij}, t, r_{ij})$$

con

$$\mathrm{bino}(x_{ij}, t, r_{ij}) = \binom{x_{ij}}{t} r_{ij}^{t}(1 - r_{ij})^{x_{ij}-t}$$

Si consideramos que las componentes en el subsistema i-ésimo son diferentes, y por tanto con diferente capacidad, el cálculo de $E(\mathbf{x})$ se complica.

El problema considerado es:

$$
\begin{array}{ll}
\textbf{(P)} & \min \sum_{i,j} c_{ij} x_{ij} \\
\text{s.t.} & E(\mathbf{x}) \geq E_0, \\
 & 0 \leq x_{ij} \leq u_{ij} \\
 & \sum_j x_{ij} \geq 1
\end{array}
$$

El problema considerado es:

$$
\begin{array}{ll}
(\mathbf{P}) & \min \sum_{i,j} c_{ij} x_{ij} \\
\text{s.t.} & E(\mathbf{x}) \geq E_0, \\
 & 0 \leq x_{ij} \leq u_{ij} \\
 & \sum_j x_{ij} \geq 1
\end{array}
$$

Si consideramos el caso componentes homogéneas en cada subsistema:

$$
\begin{array}{ll}
(\mathbf{P}) & \min \sum_{i,j} c_{ij} x_{ij} \\
\text{s.t.} & E(\mathbf{x}) \geq E_0, \\
 & l_{ij} y_{ij} \leq x_{ij} \leq u_i y_{ij} \\
 & \sum_j y_{ij} = 1
\end{array}
$$

Método algebraico

Podemos considerar el subproblema lineal, del anterior:

$$(\mathbf{LP}) \qquad \min \sum_{i,j} c_{ij} x_{ij}$$

$$\begin{aligned} \text{s.t.} \quad & l_{ij} y_{ij} \leq x_{ij} \leq u_i y_{ij} \\ & \textstyle\sum_j y_{ij} = 1 \end{aligned}$$

Método algebraico

Podemos considerar el subproblema lineal, del anterior:

$$\textbf{(LP)} \qquad \text{mín} \sum_{i,j} c_{ij} x_{ij}$$

$$\begin{array}{ll} \text{s.t.} & l_{ij} y_{ij} \leq x_{ij} \leq u_i y_{ij} \\ & \sum_j y_{ij} = 1 \end{array}$$

Si suponemos que la capacidad esta en relación directa con el precio, podemos construir una base de Gröbner para este problema. Nuestras restricciones son diagonales por bloques, con un bloque para cada subsistema, así para el subsistema i-ésimo, tenemos la base:

$$G = \{x_p s_p - t_p, x_p^m y_p t_p^{u-m} s_q^{l_q - n} r_q - x_q^n y_q t_q^{u-n} s_p^{l_p - m} r_p\}$$

con $p < q$; $0 \leq m \leq l_p$; $0 \leq n \leq l_q$; $m \leq n$.

Estrategia principal

1. Comenzando en el óptimo β de **(LP)** usamos el conjunto test inverso para introducirnos en la región factible de **(P)**

Estrategia principal

1. Comenzando en el óptimo β de **(LP)** usamos el conjunto test inverso para introducirnos en la región factible de **(P)**
2. Calculamos un punto y^0 factible para **(P)**, con coste c^0

Estrategia principal

1. Comenzando en el óptimo β de **(LP)** usamos el conjunto test inverso para introducirnos en la región factible de **(P)**
2. Calculamos un punto y^0 factible para **(P)**, con coste c^0
3. Para cada nuevo nodo **w**, con fiabilidad E_w ordenamos los nodos activos por:

$$c_p(w) = \mathbf{c}^{\mathbf{t}} \cdot \mathbf{w} + \mu \cdot \max\{0, E_0 - E_w\}$$

Estrategia principal

1. Comenzando en el óptimo β de **(LP)** usamos el conjunto test inverso para introducirnos en la región factible de **(P)**
2. Calculamos un punto y^0 factible para **(P)**, con coste c^0
3. Para cada nuevo nodo **w**, con fiabilidad E_w ordenamos los nodos activos por:

$$c_p(w) = \mathbf{c}^{\mathbf{t}} \cdot \mathbf{w} + \mu \cdot \max\{0, E_0 - E_w\}$$

Estrategia principal

1. Comenzando en el óptimo β de **(LP)** usamos el conjunto test inverso para introducirnos en la región factible de **(P)**
2. Calculamos un punto y^0 factible para **(P)**, con coste c^0
3. Para cada nuevo nodo **w**, con fiabilidad E_w ordenamos los nodos activos por:

$$c_p(w) = \mathbf{c}^{\mathbf{t}} \cdot \mathbf{w} + \mu \cdot \text{máx}\{0, E_0 - E_w\}$$

con

$$\mu = \frac{c_Y - c\beta}{E_0 - E_\beta}$$

c_Y mejor coste encontrado de punto factible para **(P)**, inicialmente $c_Y = c^0$

Ventajas

- Se mejora el método de Walk-back

Ventajas

- Se mejora el método de Walk-back
- El proceso llega muy rápido a un muy buen punto

Ventajas

- Se mejora el método de Walk-back
- El proceso llega muy rápido a un muy buen punto

Desventajas

- La certificación de optimalidad es lenta debido a la lista de nodos pendientes.

Ventajas

- Se mejora el método de Walk-back
- El proceso llega muy rápido a un muy buen punto

Desventajas

- La certificación de optimalidad es lenta debido a la lista de nodos pendientes.

Podemos usar el método como heurístico.

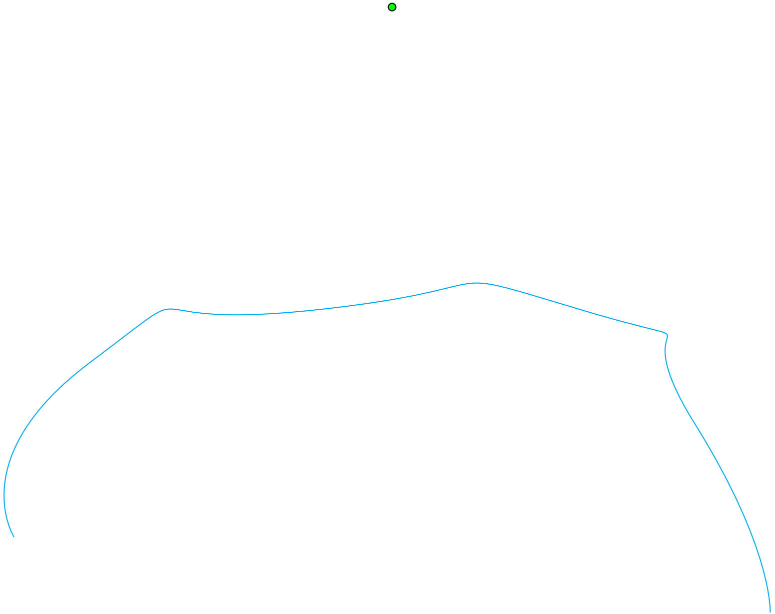
Ouzineb, M., Nourelfath, M., and Gendreau, M.
Tabu search for the redundancy allocation problem of homogenous series-parallel multi-state systems.
Reliability Engineering and System Safety, 93(8):1257–1272, (2008).

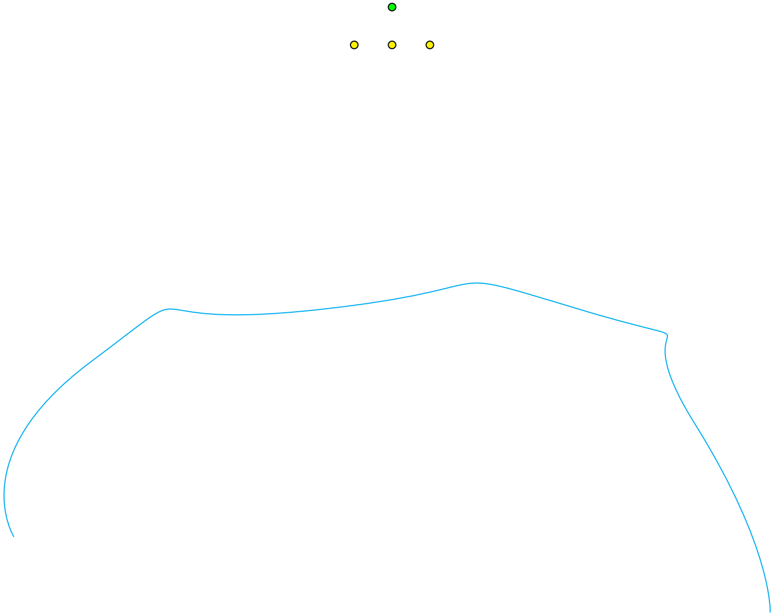
Tratan el mismo problema comparando un heurístico basado en algoritmos genéticos y otro basado en búsqueda tabú.

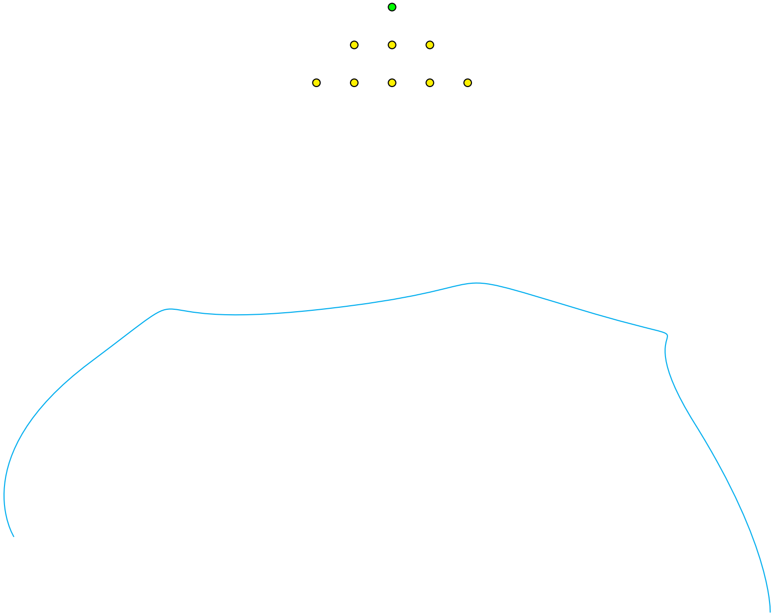
Problema	E_0	Runnig time		
		TS	GA	WB
lev4-(4/6)	0.90	5.06	2.75	0.3
	0.96	5.12	2.76	3.8
	0.99	5.03	3.06	0
lev5-(4/9)	0.975	52.91	427.29	269
	0.98	102.36	471.34	242.4
	0.99	86.57	493.89	272.1
ouz6-(4/11)	0.975	112.71	598.24	138
	0.98	126.49	613.83	228.3
	0.99	124.76	439.35	268.1

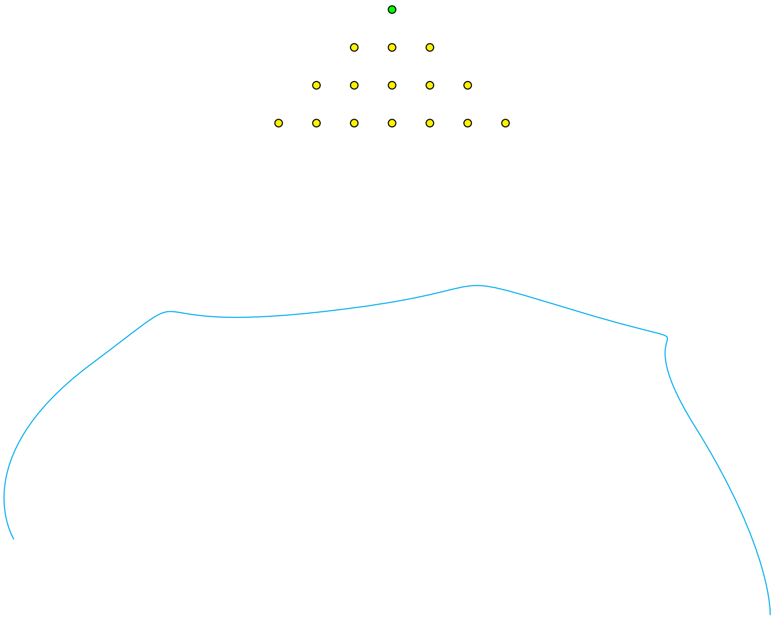
Problema	E_0	10s			100s		
		TS	GA	WB	TS	GA	WB
lev4-(4/6)	0.90	**5986**	6348	**5986**	**5986**	6196	**5986**
	0.96	**7303**	7571	**7303**	**7303**	7571	**7303**
	0.99	**8328**	**8328**	**8328**	**8328**	**8328**	**8328**

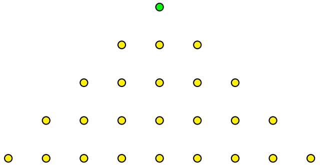
Problema	E_0	10s			100s		
		TS	GA	WB	TS	GA	WB
lev5	0.975	16566	16544	17050	**16450**	**16450**	16683
	0.98	16648	16541	17050	**16520**	16541	16612
	0.99	17180	17125	17120	**17050**	17095	17095
ouz6	0.975	24982	17410	12766	**11241**	12608	**11241**
	0.98	21929	16250	13069	**11369**	12996	12909
	0.99	36708	23748	13161	**12764**	16227	13161

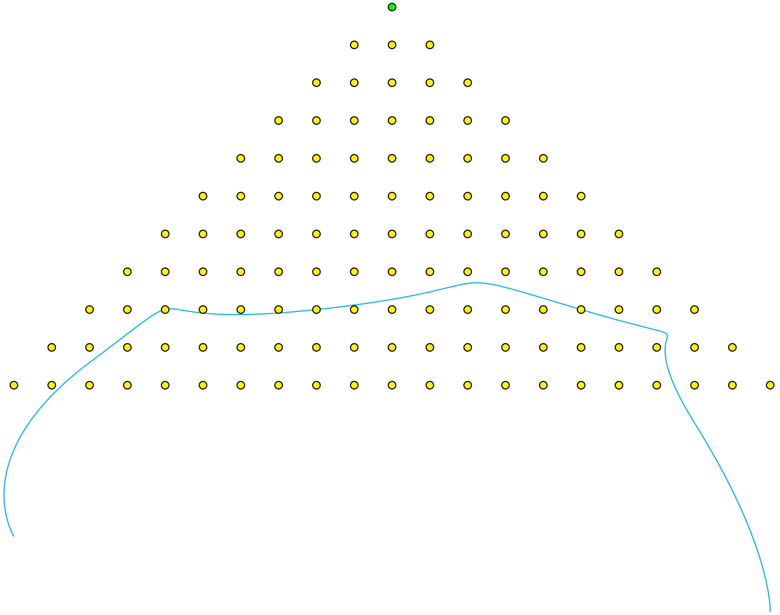


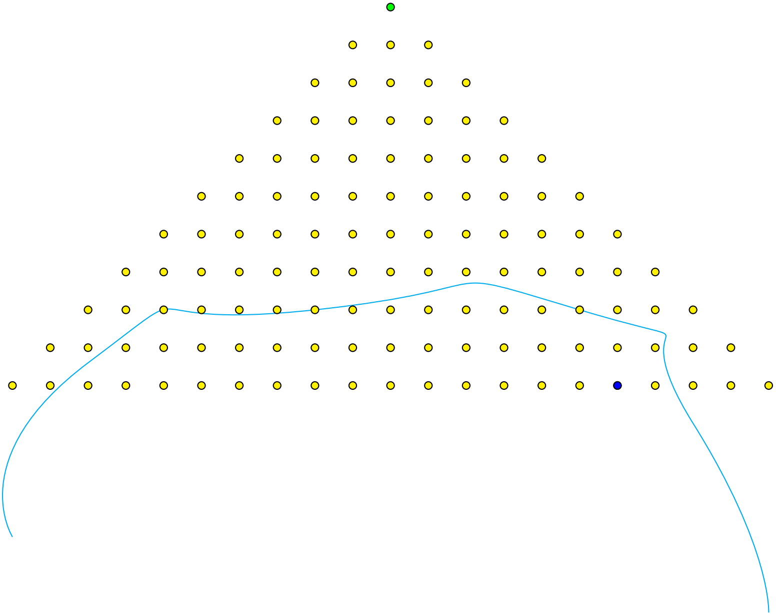


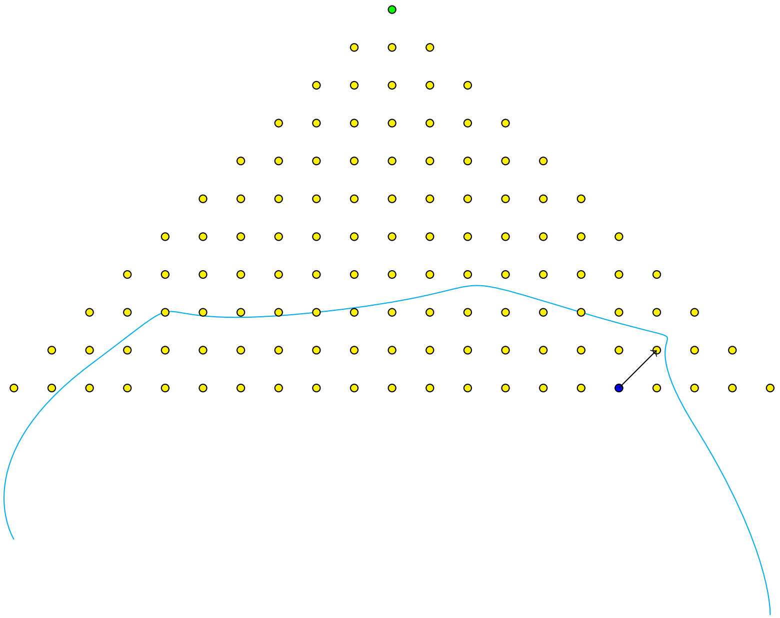


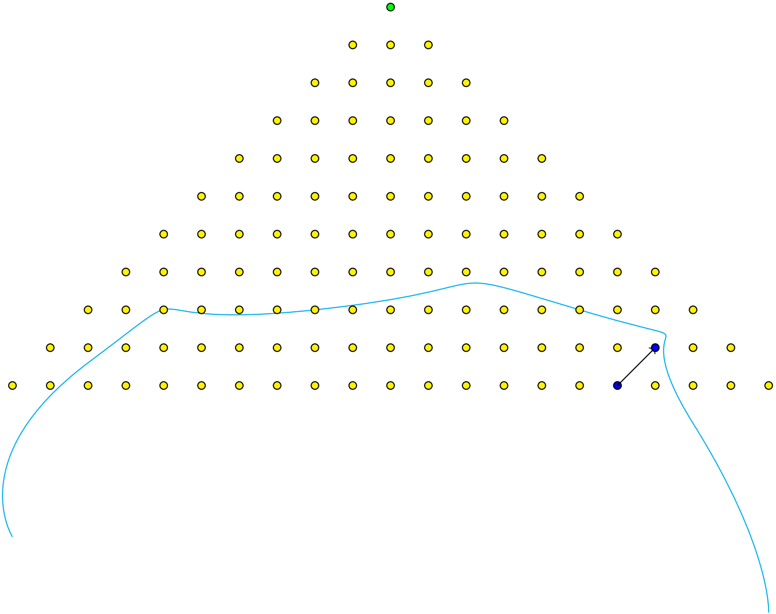


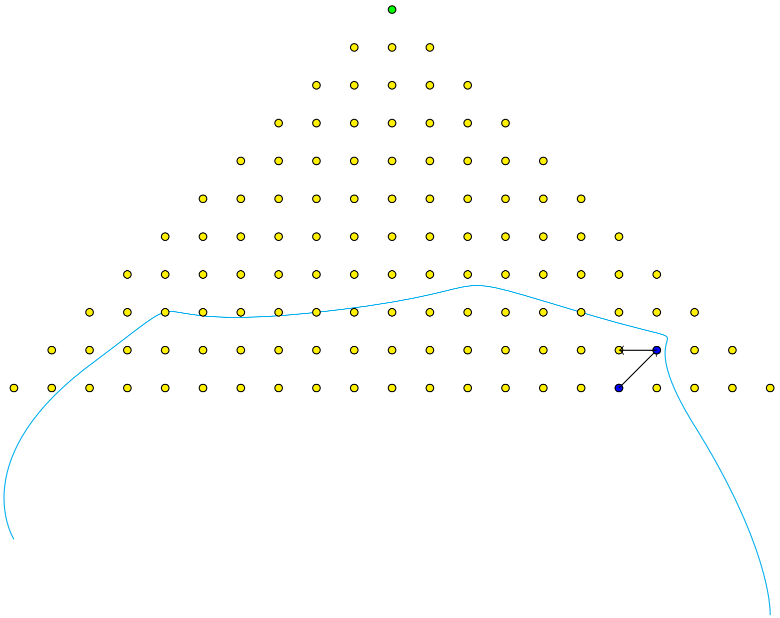


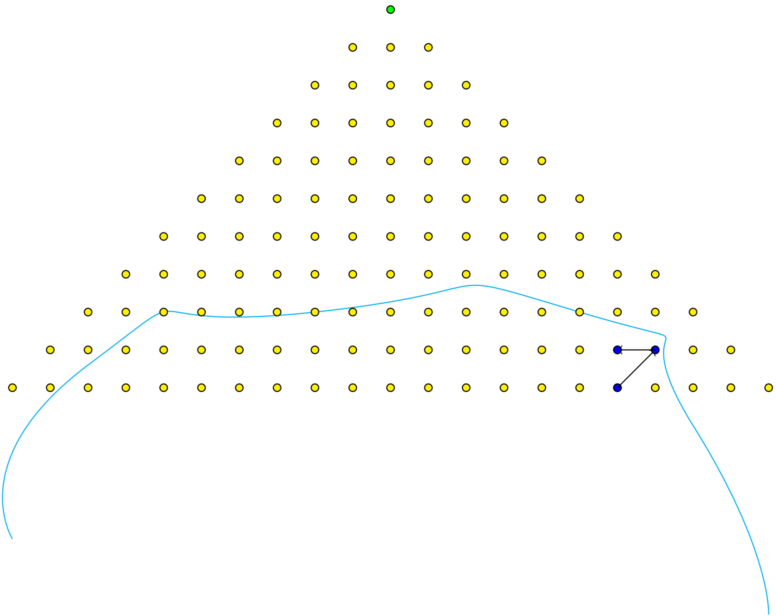


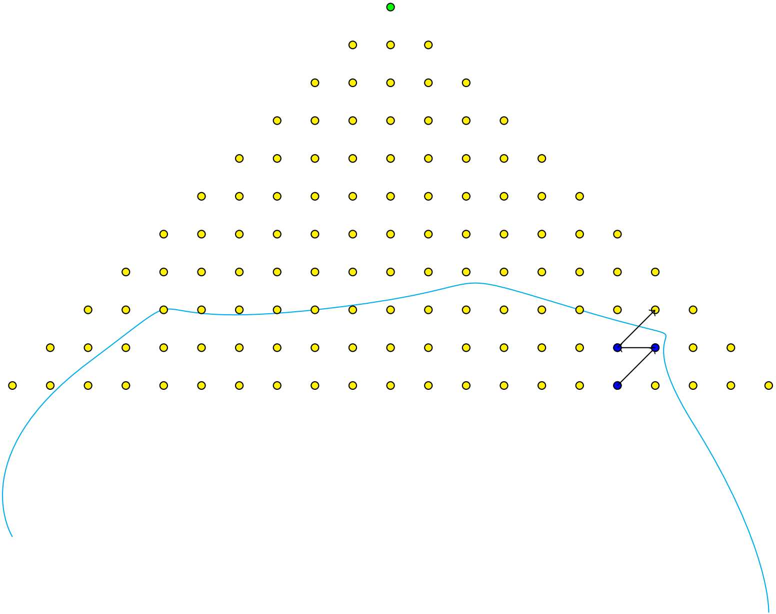


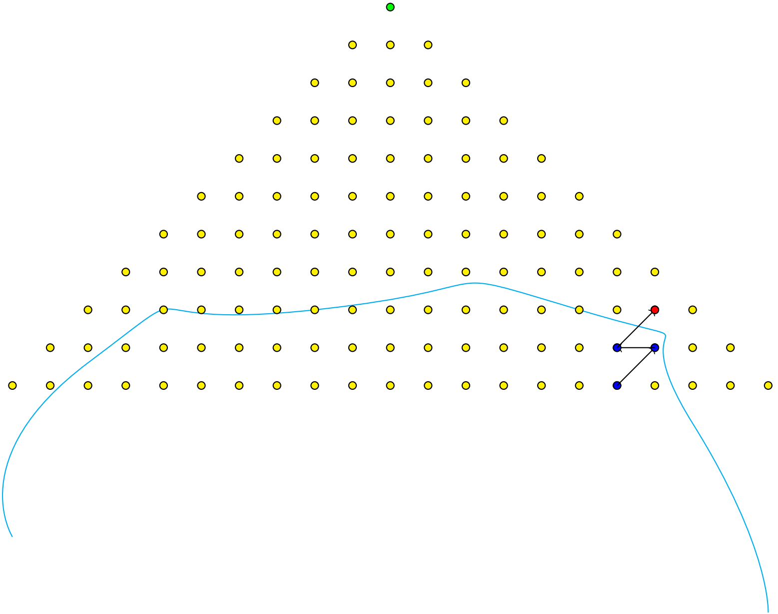


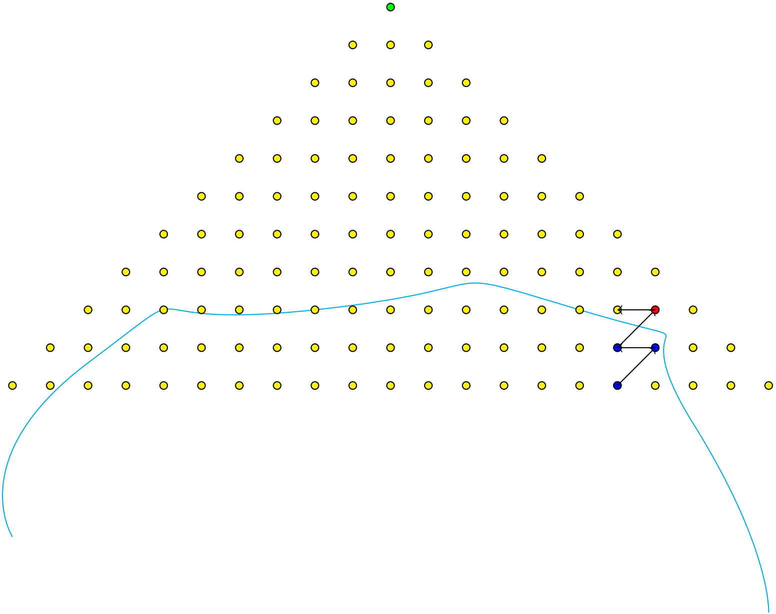


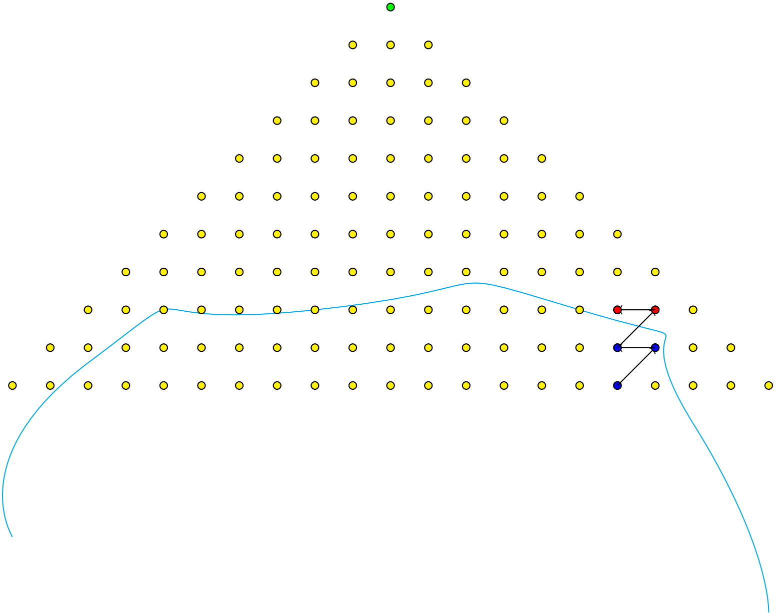


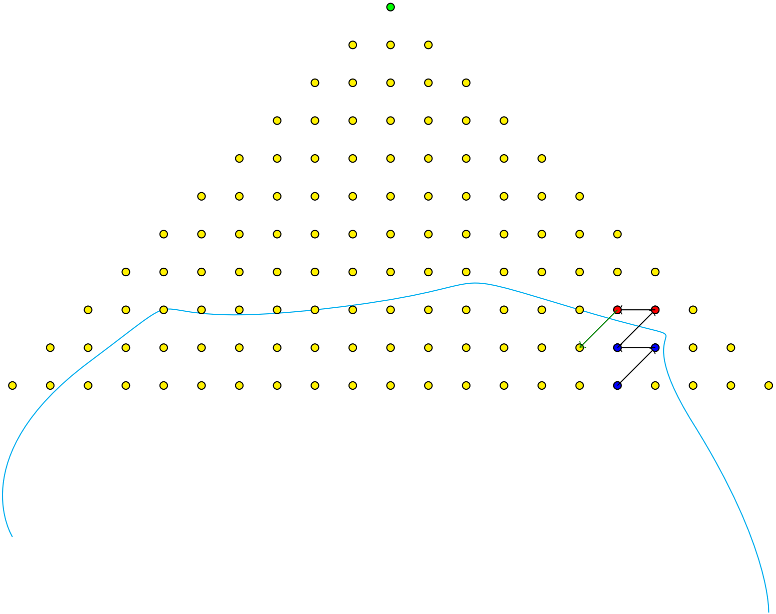


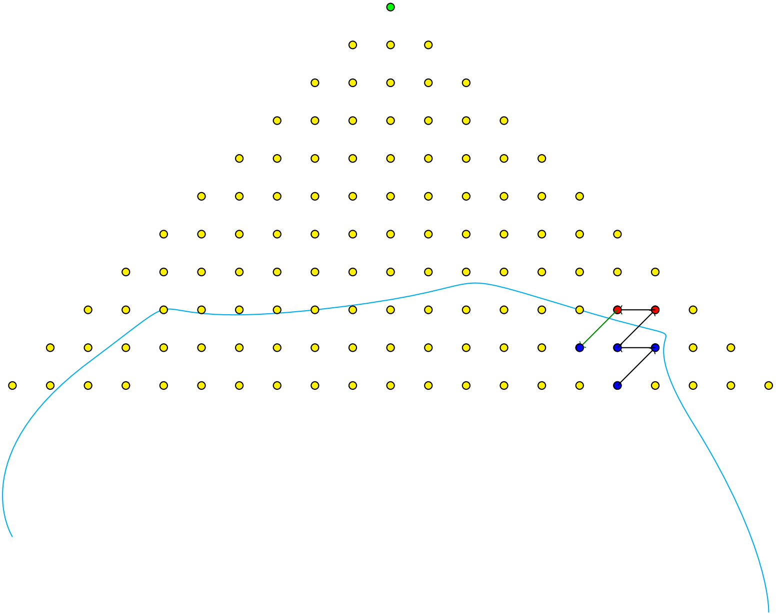












Heurísticos en problemas de fiabilidad

- Los primeros heurísticos para el problema de fiabilidad con capacidad son algoritmos genéticos, debidos a Lisnianski et al. (1996), Levitin et al. (1997) y Lisnianski y Levitin (2003).

Heurísticos en problemas de fiabilidad

- Los primeros heurísticos para el problema de fiabilidad con capacidad son algoritmos genéticos, debidos a Lisnianski et al. (1996), Levitin et al. (1997) y Lisnianski y Levitin (2003).
- El trabajo citado de Ouzineb et al. (2008) usando Búsqueda Tabú, donde precisamente comparan con los algoritmos genéticos anteriores. En un trabajo posterior (2011) combinan las técnicas de búsqueda tabú y algoritmos genéticos.

Heurísticos en problemas de fiabilidad

- Los primeros heurísticos para el problema de fiabilidad con capacidad son algoritmos genéticos, debidos a Lisnianski et al. (1996), Levitin et al. (1997) y Lisnianski y Levitin (2003).
- El trabajo citado de Ouzineb et al. (2008) usando Búsqueda Tabú, donde precisamente comparan con los algoritmos genéticos anteriores. En un trabajo posterior (2011) combinan las técnicas de búsqueda tabú y algoritmos genéticos.
- Hay trabajos de Ramirez-Marquez y Coit (2004) y Agarwal y Gupta (2007) donde se desarrollan heurísticos ad hoc.

Heurísticos en problemas de fiabilidad

- Los primeros heurísticos para el problema de fiabilidad con capacidad son algoritmos genéticos, debidos a Lisnianski et al. (1996), Levitin et al. (1997) y Lisnianski y Levitin (2003).
- El trabajo citado de Ouzineb et al. (2008) usando Búsqueda Tabú, donde precisamente comparan con los algoritmos genéticos anteriores. En un trabajo posterior (2011) combinan las técnicas de búsqueda tabú y algoritmos genéticos.
- Hay trabajos de Ramirez-Marquez y Coit (2004) y Agarwal y Gupta (2007) donde se desarrollan heurísticos ad hoc.
- Liang y Smith (2004) llevaron a cabo un algoritmo basado en colonia de hormigas para el problema de redundancia de sistemas en serie paralelo, sin considerar capacidad

- Agarwal y Sharma (2009) y Sharma et al. (2011) llevan a cabo algoritmos de colonias de hormigas en el caso heterogéneo, para dos tipos de componentes y tres, respectivamente.

- Agarwal y Sharma (2009) y Sharma et al. (2011) llevan a cabo algoritmos de colonias de hormigas en el caso heterogéneo, para dos tipos de componentes y tres, respectivamente.
- Como en el trabajo de Liang y Smith las hormigas construyen una solución en n pasos, un paso en la construcción de cada subsistema.

- Agarwal y Sharma (2009) y Sharma et al. (2011) llevan a cabo algoritmos de colonias de hormigas en el caso heterogéneo, para dos tipos de componentes y tres, respectivamente.
- Como en el trabajo de Liang y Smith las hormigas construyen una solución en n pasos, un paso en la construcción de cada subsistema.
- El sistema ACO usa agentes simples, sobre un sistema discreto, puede ser un buen método en nuestro caso.

Ant Colony Optimization

Hormigas artificiales

- Desarrollado por Marco Dorigo en su tesis (1991)

Ant Colony Optimization

Hormigas artificiales

- Desarrollado por Marco Dorigo en su tesis (1991)
- Los algoritmos de ACO reproducen el comportamiento de las hormigas reales en una colonia artificial de hormigas para resolver problemas complejos de camino mínimo

Ant Colony Optimization

Hormigas artificiales

- Desarrollado por Marco Dorigo en su tesis (1991)
- Los algoritmos de ACO reproducen el comportamiento de las hormigas reales en una colonia artificial de hormigas para resolver problemas complejos de camino mínimo
- La aparición de buenas soluciones es una propiedad emergente de la interacción cooperativa entre las hormigas

Ant Colony Optimization

Hormigas artificiales

- Desarrollado por Marco Dorigo en su tesis (1991)
- Los algoritmos de ACO reproducen el comportamiento de las hormigas reales en una colonia artificial de hormigas para resolver problemas complejos de camino mínimo
- La aparición de buenas soluciones es una propiedad emergente de la interacción cooperativa entre las hormigas
- La mayoría de las ideas ACO provienen de las hormigas reales:

Ant Colony Optimization

Hormigas artificiales

- Desarrollado por Marco Dorigo en su tesis (1991)
- Los algoritmos de ACO reproducen el comportamiento de las hormigas reales en una colonia artificial de hormigas para resolver problemas complejos de camino mínimo
- La aparición de buenas soluciones es una propiedad emergente de la interacción cooperativa entre las hormigas
- La mayoría de las ideas ACO provienen de las hormigas reales:
  - Uso de una colonia de individuos que cooperan

Ant Colony Optimization

Hormigas artificiales

- Desarrollado por Marco Dorigo en su tesis (1991)
- Los algoritmos de ACO reproducen el comportamiento de las hormigas reales en una colonia artificial de hormigas para resolver problemas complejos de camino mínimo
- La aparición de buenas soluciones es una propiedad emergente de la interacción cooperativa entre las hormigas
- La mayoría de las ideas ACO provienen de las hormigas reales:
    - Uso de una colonia de individuos que cooperan
    - Un camino de feromonas como vía de comunicación

Ant Colony Optimization

Hormigas artificiales

- Desarrollado por Marco Dorigo en su tesis (1991)
- Los algoritmos de ACO reproducen el comportamiento de las hormigas reales en una colonia artificial de hormigas para resolver problemas complejos de camino mínimo
- La aparición de buenas soluciones es una propiedad emergente de la interacción cooperativa entre las hormigas
- La mayoría de las ideas ACO provienen de las hormigas reales:
  - Uso de una colonia de individuos que cooperan
  - Un camino de feromonas como vía de comunicación
  - Una secuencia de movimientos locales para encontrar el camino más corto

Ant Colony Optimization

Hormigas artificiales

- Desarrollado por Marco Dorigo en su tesis (1991)
- Los algoritmos de ACO reproducen el comportamiento de las hormigas reales en una colonia artificial de hormigas para resolver problemas complejos de camino mínimo
- La aparición de buenas soluciones es una propiedad emergente de la interacción cooperativa entre las hormigas
- La mayoría de las ideas ACO provienen de las hormigas reales:
  - Uso de una colonia de individuos que cooperan
  - Un camino de feromonas como vía de comunicación
  - Una secuencia de movimientos locales para encontrar el camino más corto
  - Una política de decisión estocástica utilizando información local

Sistema ACO

- Cada hormiga se mueve hacia un estado vecino y así va construyendo una solución

Sistema ACO

- Cada hormiga se mueve hacia un estado vecino y así va construyendo una solución
- Los movimientos se realizan mediante una política estocástica local:

Sistema ACO

- Cada hormiga se mueve hacia un estado vecino y así va construyendo una solución
- Los movimientos se realizan mediante una política estocástica local:
    - Mediante la memoria de la hormiga $\Rightarrow$ información acerca del pasado de la hormiga

Sistema ACO

- Cada hormiga se mueve hacia un estado vecino y así va construyendo una solución
- Los movimientos se realizan mediante una política estocástica local:
  - Mediante la memoria de la hormiga $\Rightarrow$ información acerca del pasado de la hormiga
  - Mediante la cantidad de feromona del tramo

Sistema ACO

- Cada hormiga se mueve hacia un estado vecino y así va construyendo una solución
- Los movimientos se realizan mediante una política estocástica local:
  - Mediante la memoria de la hormiga $\Rightarrow$ información acerca del pasado de la hormiga
  - Mediante la cantidad de feromona del tramo
- La memoria es fundamental para la construcción de soluciones viables $\Rightarrow$ lista tabú

Sistema ACO

- Cada hormiga se mueve hacia un estado vecino y así va construyendo una solución
- Los movimientos se realizan mediante una política estocástica local:
  - Mediante la memoria de la hormiga $\Rightarrow$ información acerca del pasado de la hormiga
  - Mediante la cantidad de feromona del tramo
- La memoria es fundamental para la construcción de soluciones viables $\Rightarrow$ lista tabú
- La feromona codifica el conocimiento acumulado por todas las hormigas desde el inicio del proceso

Sistema ACO

- La feromona es utilizada a modo de memoria a largo término que influye en las decisiones de las hormigas

Sistema ACO

- La feromona es utilizada a modo de memoria a largo término que influye en las decisiones de las hormigas
- Cuándo y cuánta feromona depositar depende del problema

Sistema ACO

- La feromona es utilizada a modo de memoria a largo término que influye en las decisiones de las hormigas
- Cuándo y cuánta feromona depositar depende del problema
    - Paso a paso: A medida que se va construyendo el camino

Sistema ACO

- La feromona es utilizada a modo de memoria a largo término que influye en las decisiones de las hormigas
- Cuándo y cuánta feromona depositar depende del problema
  - Paso a paso: A medida que se va construyendo el camino
  - Retardado: Después de que se ha construido todo el camino

Sistema ACO

- La feromona es utilizada a modo de memoria a largo término que influye en las decisiones de las hormigas
- Cuándo y cuánta feromona depositar depende del problema
  - Paso a paso: A medida que se va construyendo el camino
  - Retardado: Después de que se ha construido todo el camino
- Se deposita feromona proporcionalmente a la calidad de la solución

Sistema ACO

- La feromona es utilizada a modo de memoria a largo término que influye en las decisiones de las hormigas
- Cuándo y cuánta feromona depositar depende del problema
  - Paso a paso: A medida que se va construyendo el camino
  - Retardado: Después de que se ha construido todo el camino
- Se deposita feromona proporcionalmente a la calidad de la solución

Sistema ACO

- La feromona es utilizada a modo de memoria a largo término que influye en las decisiones de las hormigas
- Cuándo y cuánta feromona depositar depende del problema
  - Paso a paso: A medida que se va construyendo el camino
  - Retardado: Después de que se ha construido todo el camino
- Se deposita feromona proporcionalmente a la calidad de la solución

Consecuencia

más hormigas visitan un camino $\Rightarrow$ más feromona tiene $\Rightarrow$ más interesante es ese camino para todas

Sistema ACO

- Cada nodo tiene una tabla de decisión para guiar a las hormigas

Sistema ACO

- Cada nodo tiene una tabla de decisión para guiar a las hormigas
- La tabla de decisión tiene una entrada para cada uno de los nodos vecinos del nodo en cuestión

Sistema ACO

- Cada nodo tiene una tabla de decisión para guiar a las hormigas
- La tabla de decisión tiene una entrada para cada uno de los nodos vecinos del nodo en cuestión
- La entrada es:

Sistema ACO

- Cada nodo tiene una tabla de decisión para guiar a las hormigas
- La tabla de decisión tiene una entrada para cada uno de los nodos vecinos del nodo en cuestión
- La entrada es:
  - $$a_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in N_i} (\tau_{il})^{\alpha}(\eta_{il})^{\beta}}$$

Sistema ACO

- Cada nodo tiene una tabla de decisión para guiar a las hormigas
- La tabla de decisión tiene una entrada para cada uno de los nodos vecinos del nodo en cuestión
- La entrada es:
  - $$a_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in N_i}(\tau_{il})^{\alpha}(\eta_{il})^{\beta}}$$
  - τ_{ij} feromona de la arista i, j

Sistema ACO

- Cada nodo tiene una tabla de decisión para guiar a las hormigas
- La tabla de decisión tiene una entrada para cada uno de los nodos vecinos del nodo en cuestión
- La entrada es:
  - $$a_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in N_i}(\tau_{il})^{\alpha}(\eta_{il})^{\beta}}$$
  - τ_{ij} feromona de la arista i, j
  - $\eta_{ij} = \frac{1}{c_{ij}}$ heurístico asociado a la arista i, j

Sistema ACO

- Cada nodo tiene una tabla de decisión para guiar a las hormigas
- La tabla de decisión tiene una entrada para cada uno de los nodos vecinos del nodo en cuestión
- La entrada es:
  - $$a_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in N_i}(\tau_{il})^{\alpha}(\eta_{il})^{\beta}}$$
  - τ_{ij} feromona de la arista i, j
  - $\eta_{ij} = \frac{1}{c_{ij}}$ heurístico asociado a la arista i, j
  - α, β peso de la feromona y del heurístico respectivamente

Regla de transición (ACO)

- La hormiga k en el nodo i elige moverse a otro nodo j, de entre las que se encuentran en su memoria de trabajo N_{ik} mediante una regla pseudo-aleatoria proporcional

Regla de transición (ACO)

- La hormiga k en el nodo i elige moverse a otro nodo j, de entre las que se encuentran en su memoria de trabajo N_{ik} mediante una regla pseudo-aleatoria proporcional
- Si $q \leq q_0$:

$$p_{ij}^{k} = \begin{cases} 1 & \text{si } j = arg\,\text{máx}\{a_{ij}\} \\ 0 & \text{en caso contrario} \end{cases}$$

Regla de transición (ACO)

- La hormiga k en el nodo i elige moverse a otro nodo j, de entre las que se encuentran en su memoria de trabajo N_{ik} mediante una regla pseudo-aleatoria proporcional
- Si $q \leq q_0$:

$$p_{ij}^k = \begin{cases} 1 & \text{si } j = arg \max\{a_{ij}\} \\ 0 & \text{en caso contrario} \end{cases}$$

- Si $q > q_0$

$$p_{ij}^k(t) = \frac{a_{ij}}{\sum_{l \in N_i^k} a_{il}}$$

donde a_{il} es el elemento de la tabla de decisión del tramo il.

Actualización local

- Cada vez que una hormiga k visita un camino, la feromona del mismo se actualiza mediante:

$$\tau_{ij}(t) = (1-\phi)\tau_{ij}(t-1) + \phi\tau_0$$

Actualización local

- Cada vez que una hormiga k visita un camino, la feromona del mismo se actualiza mediante:

$$\tau_{ij}(t) = (1 - \phi)\tau_{ij}(t-1) + \phi\tau_0$$

- ρ es la tasa de volatilidad, y τ_0 la cantidad inicial de feromona

Actualización local

- Cada vez que una hormiga k visita un camino, la feromona del mismo se actualiza mediante:

$$\tau_{ij}(t) = (1 - \phi)\tau_{ij}(t - 1) + \phi\tau_0$$

- ρ es la tasa de volatilidad, y τ_0 la cantidad inicial de feromona
- Esto hace que un arco visitado pierda feromona

Actualización local

- Cada vez que una hormiga k visita un camino, la feromona del mismo se actualiza mediante:

$$\tau_{ij}(t) = (1 - \phi)\tau_{ij}(t-1) + \phi\tau_0$$

- ρ es la tasa de volatilidad, y τ_0 la cantidad inicial de feromona
- Esto hace que un arco visitado pierda feromona
- Si el arco visitado ganara feromona entonces las hormigas se seguirían unas a otras

Actualización local

- Cada vez que una hormiga k visita un camino, la feromona del mismo se actualiza mediante:

$$\tau_{ij}(t) = (1 - \phi)\tau_{ij}(t-1) + \phi\tau_0$$

- ρ es la tasa de volatilidad, y τ_0 la cantidad inicial de feromona
- Esto hace que un arco visitado pierda feromona
- Si el arco visitado ganara feromona entonces las hormigas se seguirían unas a otras
- Queremos que las hormigas exploren cada una alternativas diferentes

Actualización local

- Cada vez que una hormiga k visita un camino, la feromona del mismo se actualiza mediante:

$$\tau_{ij}(t) = (1 - \phi)\tau_{ij}(t - 1) + \phi\tau_0$$

- ρ es la tasa de volatilidad, y τ_0 la cantidad inicial de feromona
- Esto hace que un arco visitado pierda feromona
- Si el arco visitado ganara feromona entonces las hormigas se seguirían unas a otras
- Queremos que las hormigas exploren cada una alternativas diferentes
- Si los arcos visitados pierden feromona entonces las hormigas preferirán arcos poco visitados

Actualización global de tramos

- Premiar los tramos pertenecientes a los recorridos más cortos

Actualización global de tramos

- Premiar los tramos pertenecientes a los recorridos más cortos
- La mejor hormiga deposita una cantidad de feromona en los tramos de su recorrido

Actualización global de tramos

- Premiar los tramos pertenecientes a los recorridos más cortos
- La mejor hormiga deposita una cantidad de feromona en los tramos de su recorrido
- La cantidad de feromona depositada es inversamente proporcional a la longitud del recorrido L^+ :

$$\tau_{ij}(t) = (1-\rho)\tau_{ij}(t-1) + \rho\Delta\tau_{ij}(t) \quad i,j \in T^+$$

donde $\Delta\tau_{ij}(t) = \frac{1}{L^+}$ y T^+ es el recorrido de la mejor hormiga

- Cada hormiga genera en n pasos una solución al sistema
- En los trabajos de fiabilidad con capacidad la feromona inicial viene dada por

$$\tau_i^0(y) = \begin{cases} \frac{R_{ij}(k)}{kc_{ij}} & \text{si } y = j(k) \\ \frac{1-(1-R_{ij_1}(k_1))(1-R_{ij_2}(k_2))}{k_1 c_{ij_1} + k_2 c_{ij_2}} & \text{si } y = (j_1(k_1), j_2(k_2)) \end{cases}$$

- Y el heurístico como:

$$\eta_i(y) = \begin{cases} kg_{ij} & \text{si } y = j(k) \\ k_1 g_{ij_1} + k_2 g_{ij_2} & \text{si } y = (j_1(k_1), j_2(k_2)) \end{cases}$$

Nuestro método

- Nosotros queremos movernos por el espacio de soluciones, es decir, una hormiga se mueve de una configuración completa a otra

Nuestro método

- Nosotros queremos movernos por el espacio de soluciones, es decir, una hormiga se mueve de una configuración completa a otra
- Todas las hormigas parten de un nido, un punto factible obtenido

Nuestro método

- Nosotros queremos movernos por el espacio de soluciones, es decir, una hormiga se mueve de una configuración completa a otra
- Todas las hormigas parten de un nido, un punto factible obtenido
- No sabemos cuantos pasos necesitan las hormigas para llegar a una solución, tenemos que fijarlo de antemano para así poder generar las sucesivas iteraciones

Nuestro método

- Nosotros queremos movernos por el espacio de soluciones, es decir, una hormiga se mueve de una configuración completa a otra
- Todas las hormigas parten de un nido, un punto factible obtenido
- No sabemos cuantos pasos necesitan las hormigas para llegar a una solución, tenemos que fijarlo de antemano para así poder generar las sucesivas iteraciones
- Hemos de penalizar los puntos no factibles, los que no alcanzan la fiabilidad umbral

Nuestro método

- Nosotros queremos movernos por el espacio de soluciones, es decir, una hormiga se mueve de una configuración completa a otra
- Todas las hormigas parten de un nido, un punto factible obtenido
- No sabemos cuantos pasos necesitan las hormigas para llegar a una solución, tenemos que fijarlo de antemano para así poder generar las sucesivas iteraciones
- Hemos de penalizar los puntos no factibles, los que no alcanzan la fiabilidad umbral
- Nuestro heurístico debe balancear coste con fiabilidad

Heurístico

$$\eta_1(\mathbf{x}) = \frac{\left(\min\{1, \frac{E(\mathbf{x})}{E_0}\}\right)^{\gamma}}{\left(\frac{c(\mathbf{x})}{c_0}\right)^{\kappa}}$$

Heurístico

$$\eta_1(\mathbf{x}) = \frac{\left(\min\{1, \frac{E(\mathbf{x})}{E_0}\}\right)^{\gamma}}{\left(\frac{c(\mathbf{x})}{c_0}\right)^{\kappa}}$$

$$\eta_2(\mathbf{x}) = \frac{1}{\left(1{,}02 - \min\{1, \frac{E(\mathbf{x})}{E_0}\}\right)^{\gamma}\left(\frac{c(\mathbf{x})}{c_0}\right)^{\kappa}}$$

- Iniciamos la feromona a un valor inicial, igual para todos los nodos

- Iniciamos la feromona a un valor inicial, igual para todos los nodos
- Para cada colonia

- Iniciamos la feromona a un valor inicial, igual para todos los nodos
- Para cada colonia
  - Para cada hormiga

- Iniciamos la feromona a un valor inicial, igual para todos los nodos
- Para cada colonia
  - Para cada hormiga
    - Calcula sus vecinos sumando y restando el conjunto test

- Iniciamos la feromona a un valor inicial, igual para todos los nodos
- Para cada colonia
  - Para cada hormiga
    - Calcula sus vecinos sumando y restando el conjunto test
    - Método de selección mediante ruleta

- Iniciamos la feromona a un valor inicial, igual para todos los nodos
- Para cada colonia
  - Para cada hormiga
    - Calcula sus vecinos sumando y restando el conjunto test
    - Método de selección mediante ruleta
    - Actualización online

- Iniciamos la feromona a un valor inicial, igual para todos los nodos
- Para cada colonia
  - Para cada hormiga
    - Calcula sus vecinos sumando y restando el conjunto test
    - Método de selección mediante ruleta
    - Actualización online
  - Actualización del demonio, selección de la mejor hormiga por heurístico, factible

- Iniciamos la feromona a un valor inicial, igual para todos los nodos
- Para cada colonia
  - Para cada hormiga
    - Calcula sus vecinos sumando y restando el conjunto test
    - Método de selección mediante ruleta
    - Actualización online
  - Actualización del demonio, selección de la mejor hormiga por heurístico, factible
  - Actualización offline, se añade feromona al camino de la mejor hormiga

- Iniciamos la feromona a un valor inicial, igual para todos los nodos
- Para cada colonia
  - Para cada hormiga
    - Calcula sus vecinos sumando y restando el conjunto test
    - Método de selección mediante ruleta
    - Actualización online
  - Actualización del demonio, selección de la mejor hormiga por heurístico, factible
  - Actualización offline, se añade feromona al camino de la mejor hormiga
  - Actualización del coste global

- Iniciamos la feromona a un valor inicial, igual para todos los nodos
- Para cada colonia
  - Para cada hormiga
    - Calcula sus vecinos sumando y restando el conjunto test
    - Método de selección mediante ruleta
    - Actualización online
  - Actualización del demonio, selección de la mejor hormiga por heurístico, factible
  - Actualización offline, se añade feromona al camino de la mejor hormiga
  - Actualización del coste global
- Mejor solución encontrada

Dificultades

- Gran número de parámetros:

$$\alpha,\ \beta,\ \gamma,\ \kappa,\ \rho,\ q_0\ n, \text{ iteraciones, pasos}$$

Dificultades

- Gran número de parámetros:

$$\alpha,\ \beta,\ \gamma,\ \kappa,\ \rho,\ q_0\ n,\ \text{iteraciones, pasos}$$

- El caso ong se resiste

Nuevos tiempos

Problema	E_0	Runnig time				
		TS	GA	WB	ACO1	ACO2
lev4-(4/6)	0.90	5.06	2.75	0.3	2.3	4.4
	0.96	5.12	2.76	3.8	2.1	3.3
	0.99	5.03	3.06	0	1.8	2.8
lev5-(4/9)	0.975	52.91	427.29	269	3.3	6.0
	0.98	102.36	471.34	242.4	397.6	5.8
	0.99	86.57	493.89	272.1	9.37	5
ouz6-(4/11)	0.975	112.71	598.24	138	-	-
	0.98	126.49	613.83	228.3	-	-
	0.99	124.76	439.35	268.1	-	-

Work in progress

- Desarrollo de hormigas en paralelo
- Escritura en C++
- Caso heterogéneo