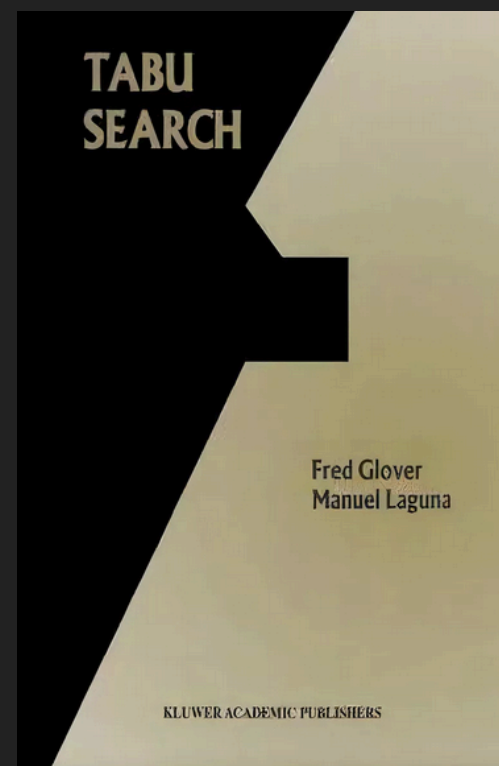


12

BÚSQUEDA TABÚ

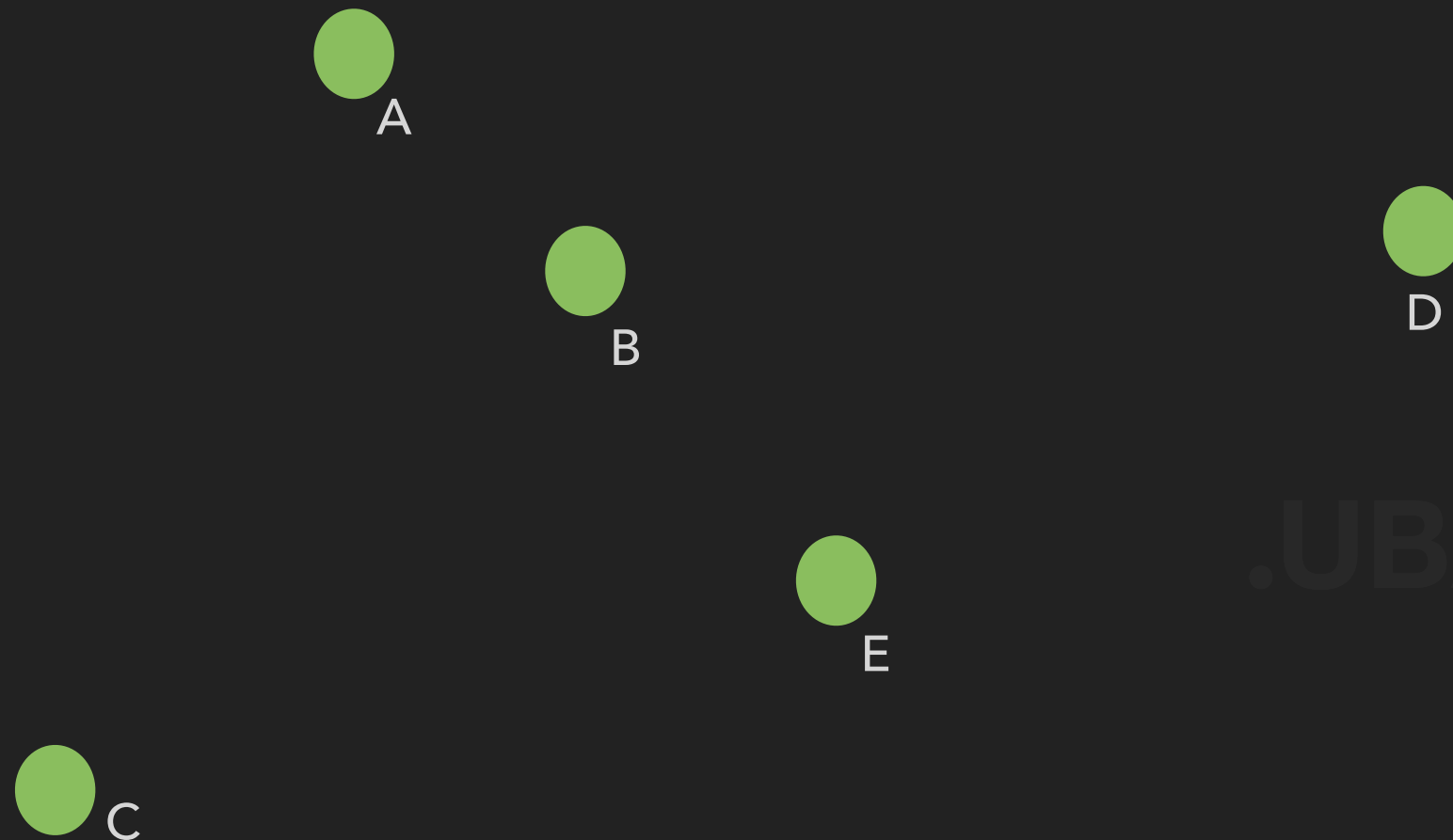
INICIOS DE TS

- ▶ **Tabu Search** es un método de búsqueda metaheurística.
- ▶ Fue propuesto por primera vez por Glover (1986) y también fue desarrollado por Hansen (1986).



¿COMO FUNCIONA TS?

Tenemos 5 ciudades con sus correspondientes distancias y una lista Tabú vacía



Lista Tabú= []

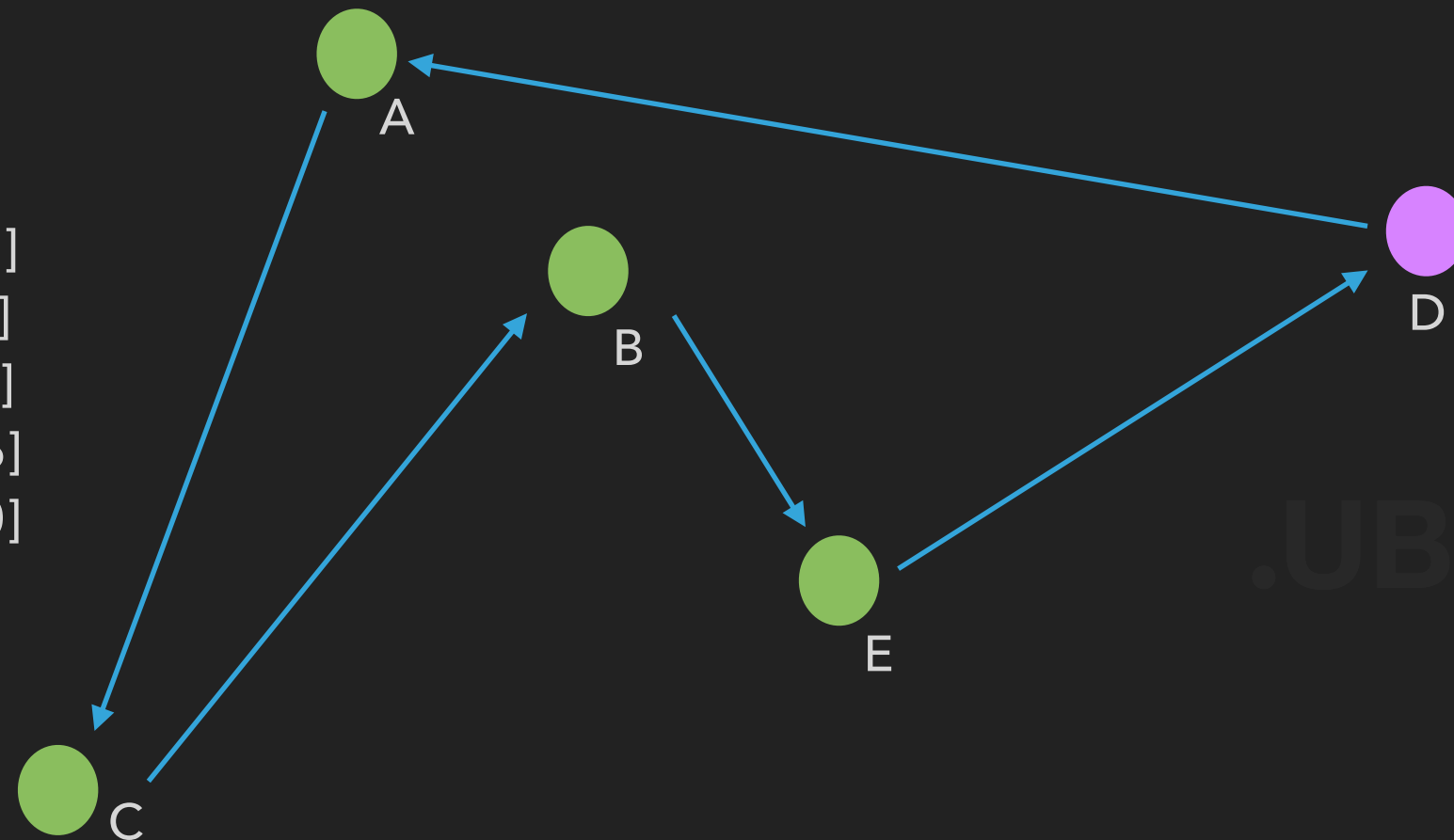
Algunas distancias: $d_{AB}=2$, $d_{AC}=9$, $d_{AD}=10$, $d_{AE}=7$...

¿COMO FUNCIONA TS? (CONT.)

Paso 1: Elijo una solución aleatoria [D,A,C,B,E,D]

Distancias:

	A	B	C	D	E
A	0	2	9	10	7
B	2	0	6	4	3
C	9	6	0	8	5
D	10	4	8	0	6
E	7	3	5	6	0



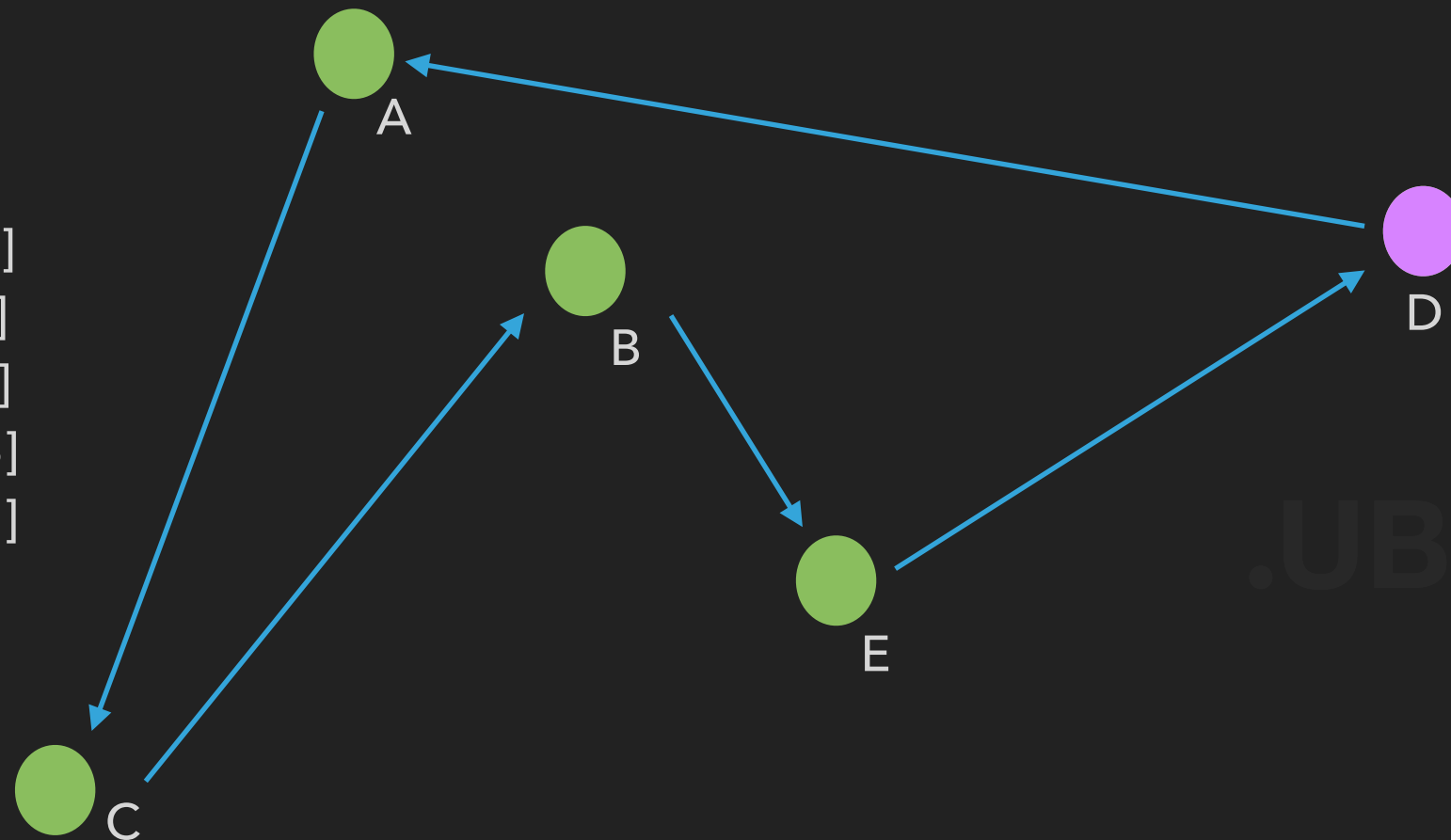
Lista Tabú= []

¿COMO FUNCIONA TS? (CONT.)

Paso 2: Calculo la sumatoria de distancias entre las ciudades elegidas $d[D,A,C,B,E,D]=10+9+6+3+6=34$

Distancias:

	A	B	C	D	E
A	0	2	9	10	7
B	2	0	6	4	3
C	9	6	0	8	5
D	10	4	8	0	6
E	7	3	5	6	0



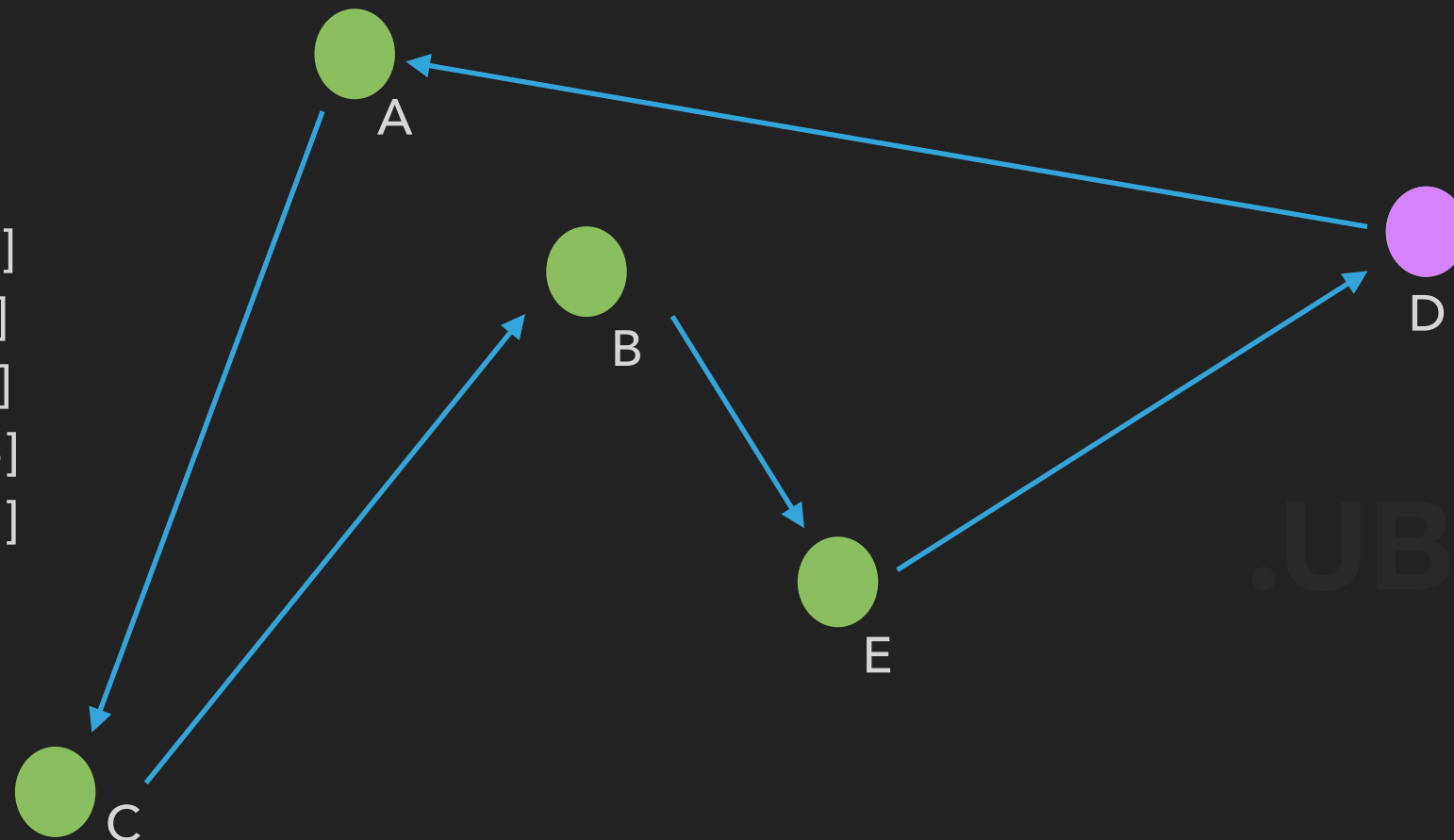
Lista Tabú= []

¿COMO FUNCIONA TS? (CONT.)

Paso 3: Genero vecindarios a partir de la solución [D,A,C,B,E,D]

Distancias:

	A	B	C	D	E
A	0	2	9	10	7
B	2	0	6	4	3
C	9	6	0	8	5
D	10	4	8	0	6
E	7	3	5	6	0



Vecindarios:

- [D,A,C,B,E]
 1 [A,D,C,B,E]
 2 [C,A,D,B,E]
 3 [B,A,C,D,E]
 4 [E,A,C,B,D]
 5 [D,C,A,B,E]
 6 [D,B,C,A,E]
 7 [D,E,C,B,A]
 8 [D,A,B,C,E]
 9 [D,A,E,B,C]
 10 [D,A,C,E,B]

Lista Tabú= []

La cantidad de vecindarios serán $C(n,2) = n(n-1) / 2$

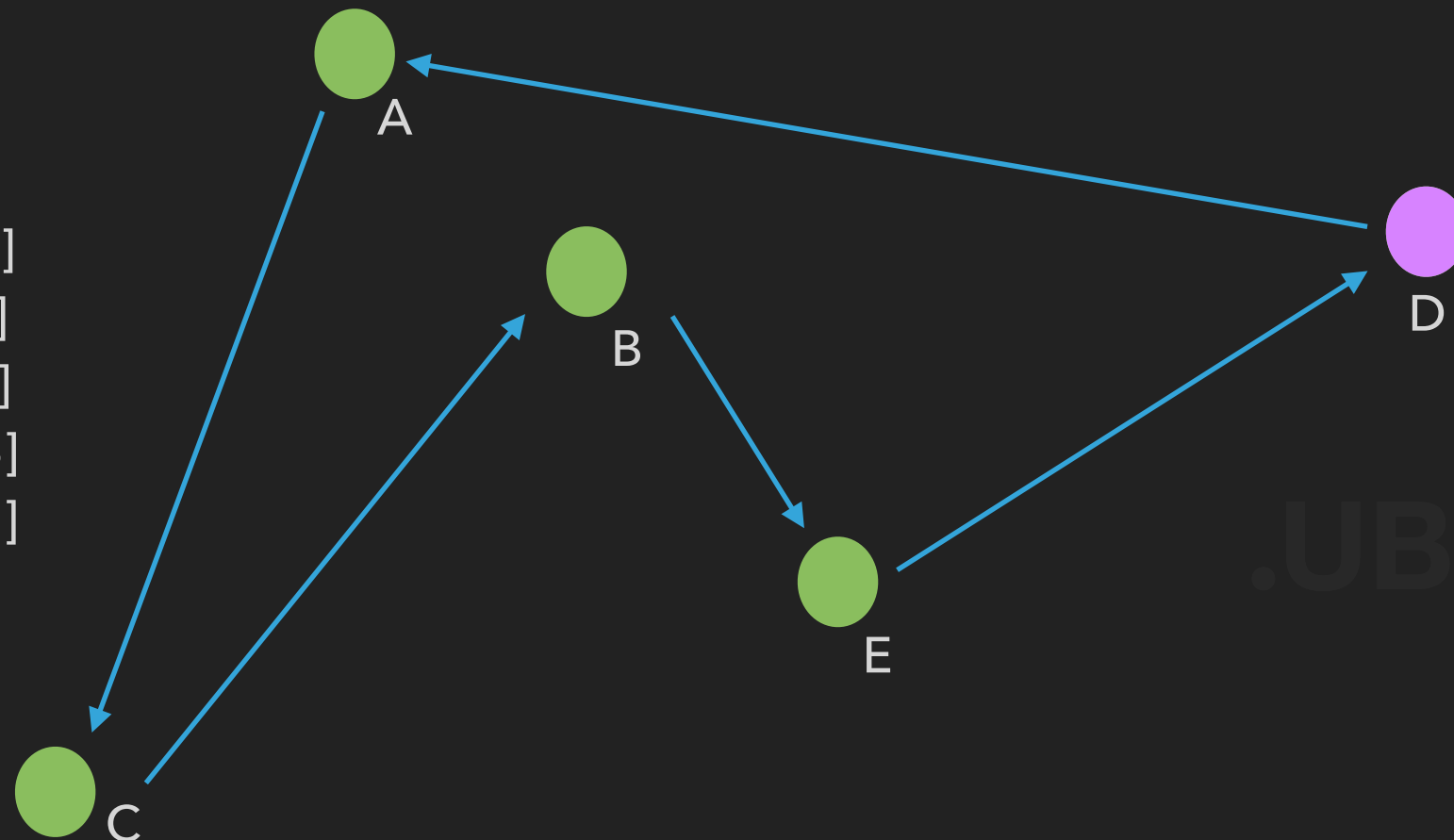
$$C(5,2) = 5(5-1) / 2 = 10$$

¿COMO FUNCIONA TS? (CONT.)

Paso 4: Selecciono el mejor vecindario que no esté en la lista tabú

Distancias:

	A	B	C	D	E
A	0	2	9	10	7
B	2	0	6	4	3
C	9	6	0	8	5
D	10	4	8	0	6
E	7	3	5	6	0



Vecindarios:

[D,A,C,B,E]

1 [A,D,C,B,E,A] = 34

2 [C,A,D,B,E,C] = 31

3 [B,A,C,D,E,B] = 28

4 [E,A,C,B,D,E] = 32

5 [D,C,A,B,E,D] = 28

6 [D,B,C,A,E,D] = 32

7 [D,E,C,B,A,D] = 29

8 [D,A,B,C,E,D] = 29

9 [D,A,E,B,C,D] = 34

10 [D,A,C,E,B,D] = 31

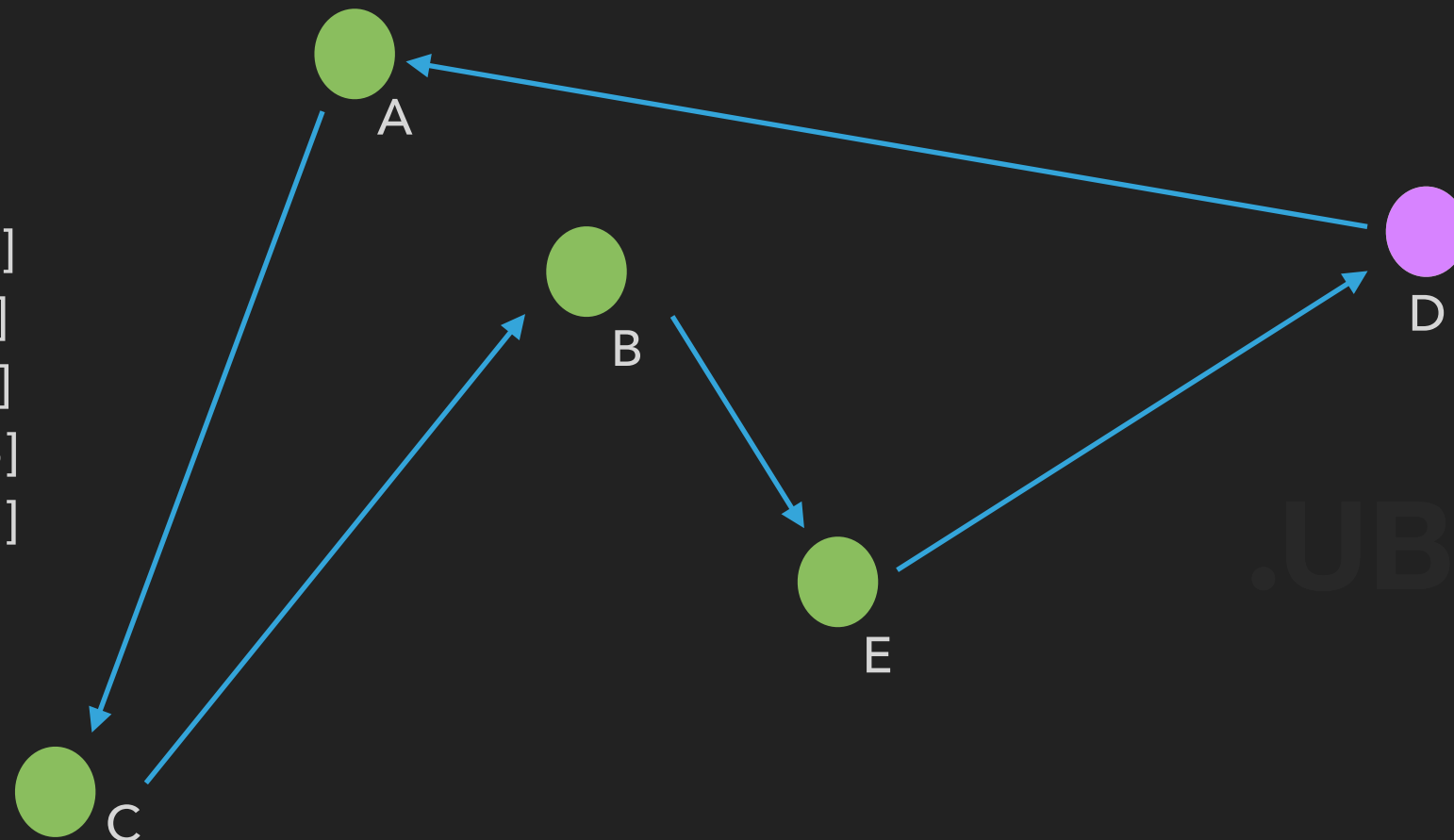
Lista Tabú= []

¿COMO FUNCIONA TS? (CONT.)

Paso 5: Actualizar solución (antes era [D,A,C,B,E,D], ahora es [B,A,C,D,E,B])

Distancias:

	A	B	C	D	E
A	0	2	9	10	7
B	2	0	6	4	3
C	9	6	0	8	5
D	10	4	8	0	6
E	7	3	5	6	0



Vecindarios:

[D,A,C,B,E]

1 [A,D,C,B,E,A] = 34

2 [C,A,D,B,E,C] = 31

3 [B,A,C,D,E,B] = 28

4 [E,A,C,B,D,E] = 32

5 [D,C,A,B,E,D] = 28

6 [D,B,C,A,E,D] = 32

7 [D,E,C,B,A,D] = 29

8 [D,A,B,C,E,D] = 29

9 [D,A,E,B,C,D] = 34

10 [D,A,C,E,B,D] = 31

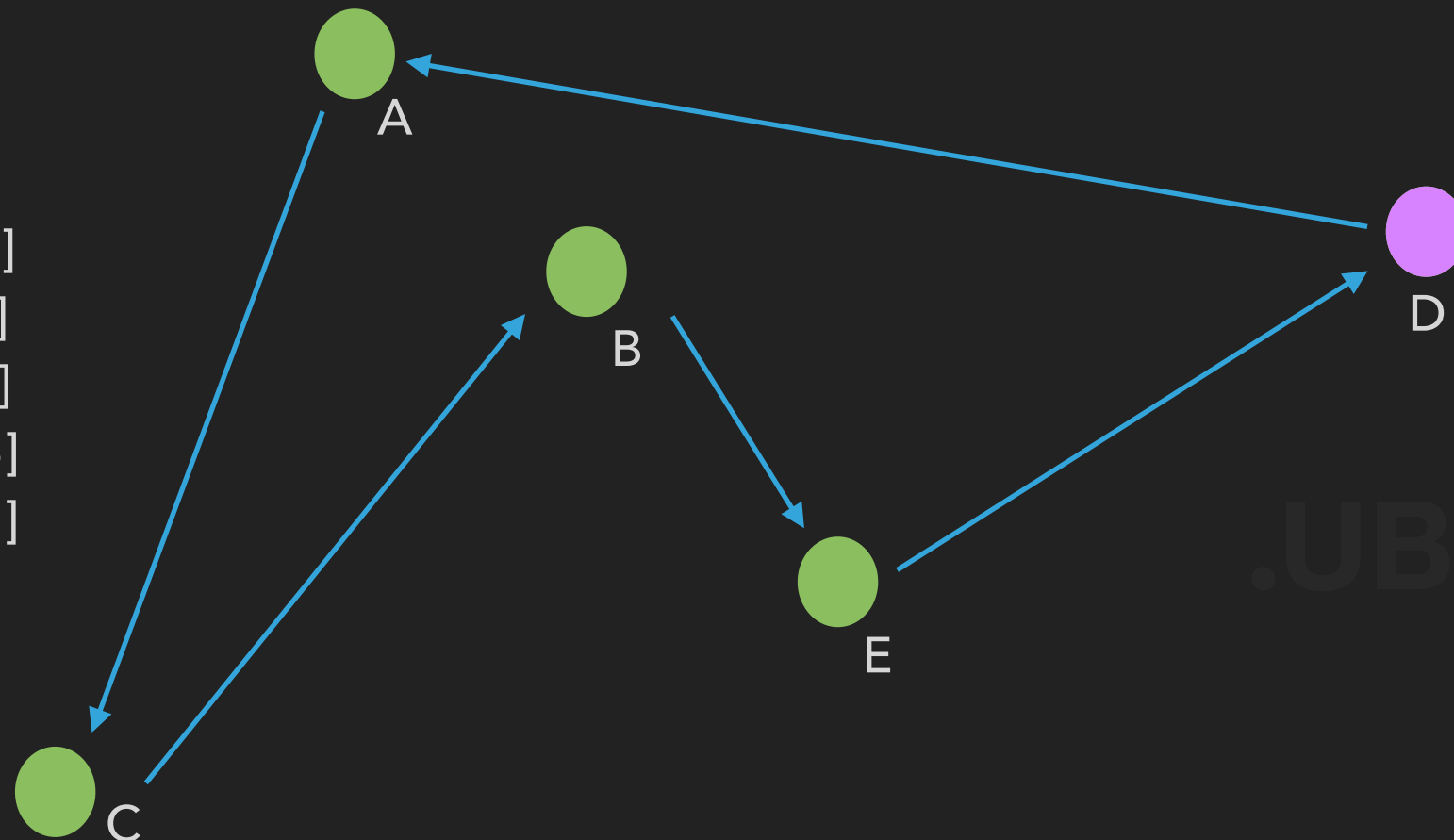
Lista Tabú= []

¿COMO FUNCIONA TS? (CONT.)

Paso 6: Actualizar mejor óptimo (antes era $d[D,A,C,B,E,D]=34$, ahora es $d[B,A,C,D,E,B] = 28$)

Distancias:

	A	B	C	D	E
A	0	2	9	10	7
B	2	0	6	4	3
C	9	6	0	8	5
D	10	4	8	0	6
E	7	3	5	6	0



Vecindarios:

- [D,A,C,B,E]
- 1 [A,D,C,B,E,A] = 34
 - 2 [C,A,D,B,E,C] = 31
 - 3 [B,A,C,D,E,B] = 28
 - 4 [E,A,C,B,D,E] = 32
 - 5 [D,C,A,B,E,D] = 28
 - 6 [D,B,C,A,E,D] = 32
 - 7 [D,E,C,B,A,D] = 29
 - 8 [D,A,B,C,E,D] = 29
 - 9 [D,A,E,B,C,D] = 34
 - 10 [D,A,C,E,B,D] = 31

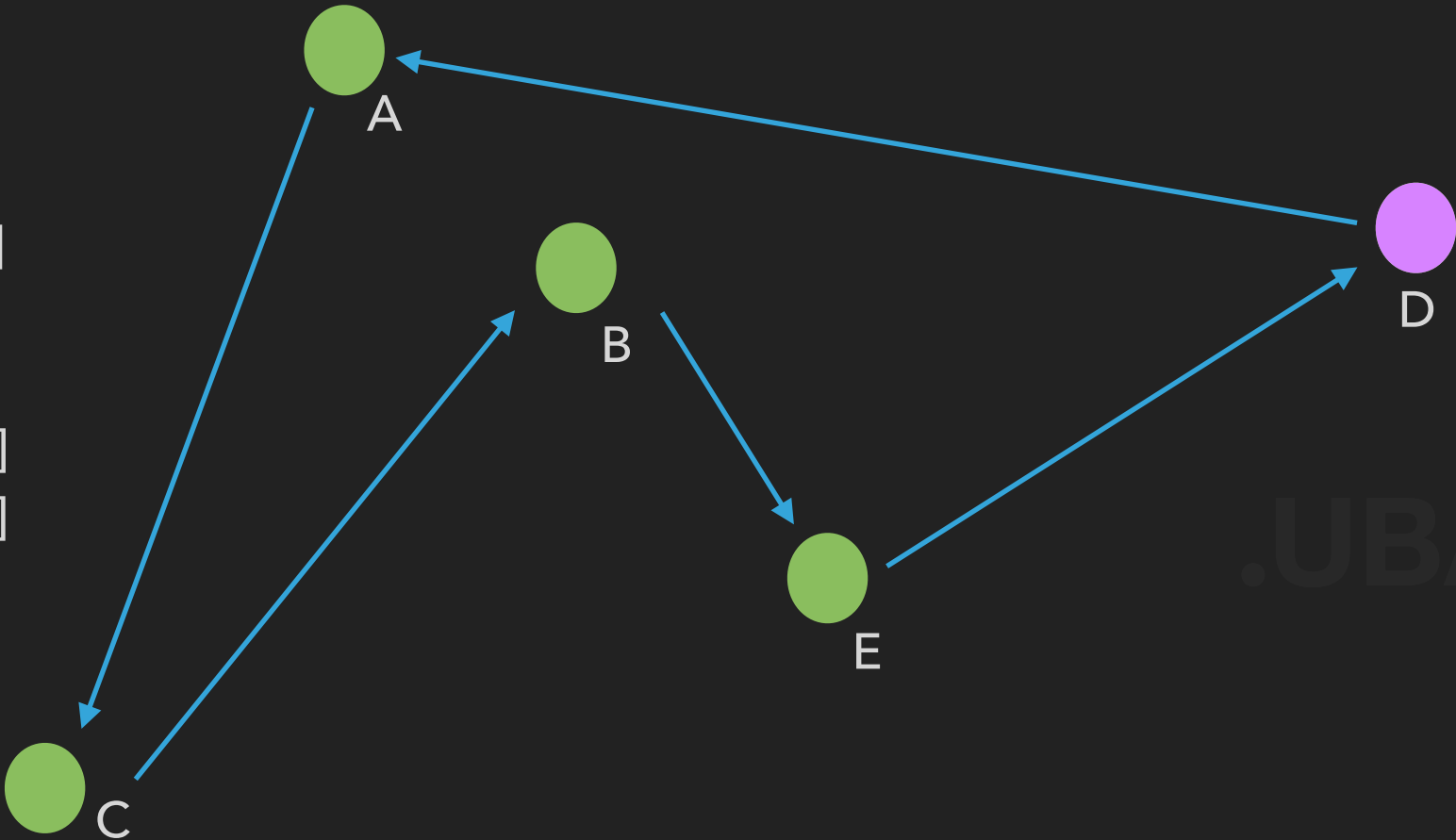
Lista Tabú= []

¿COMO FUNCIONA TS? (CONT.)

Paso 7: Actualizar lista Tabú

Distancias:

	A	B	C	D	E
A	0	2	9	10	7
B	2	0	6	4	3
C	9	6	0	8	5
D	10	4	8	0	6
E	7	3	5	6	0



Vecindarios:

[D,A,C,B,E]

1	[A,D,C,B,E,A]	= 34
2	[C,A,D,B,E,C]	= 31
3	[B,A,C,D,E,B]	= 28
4	[E,A,C,B,D,E]	= 32
5	[D,C,A,B,E,D]	= 28
6	[D,B,C,A,E,D]	= 32
7	[D,E,C,B,A,D]	= 29
8	[D,A,B,C,E,D]	= 29
9	[D,A,E,B,C,D]	= 34
10	[D,A,C,E,B,D]	= 31

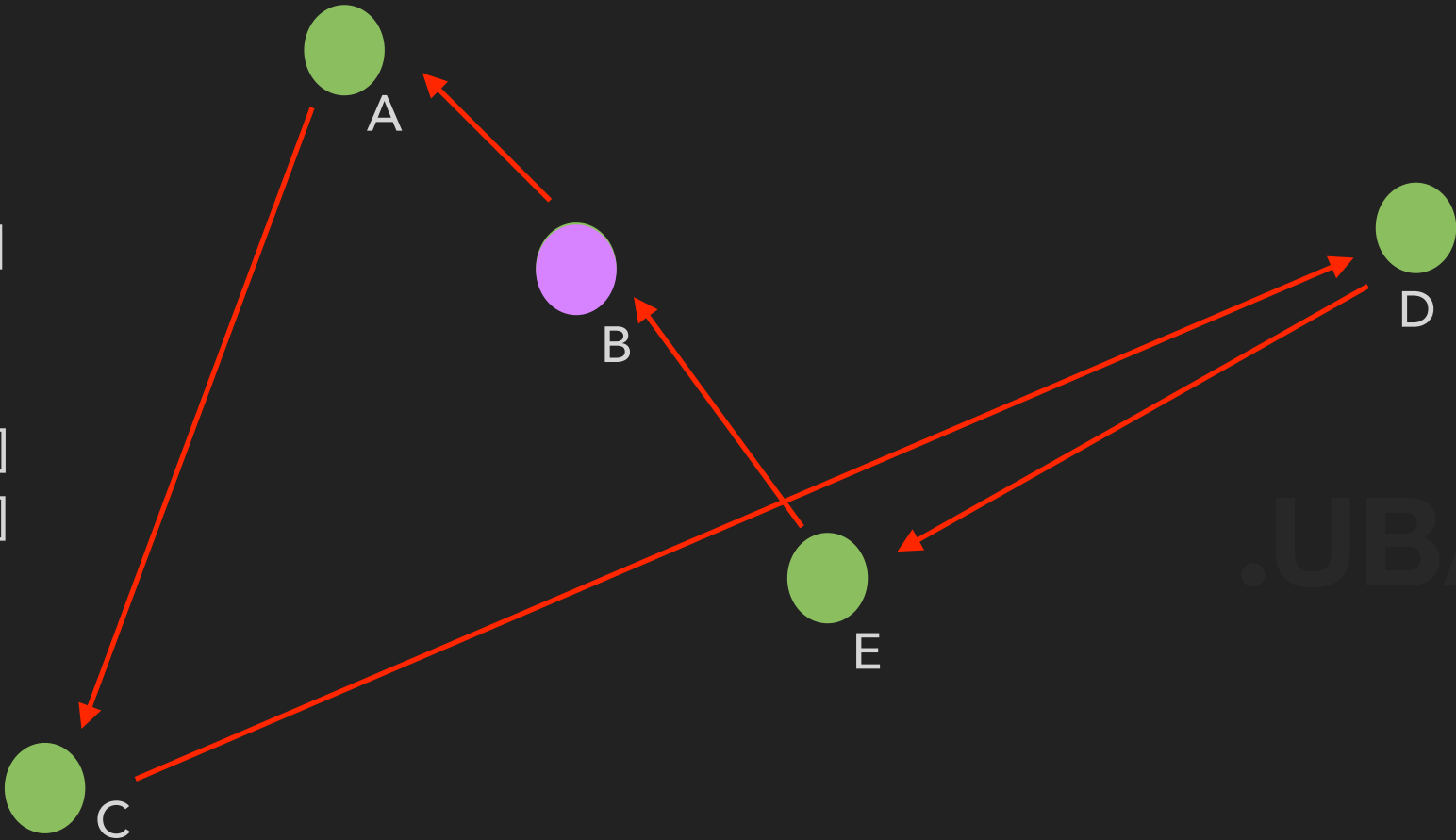
Lista Tabú= [(D,B)]

¿COMO FUNCIONA TS? (CONT.)

Paso 8: Volver al paso 2, con la nueva solución [B,A,C,D,E,B] = 28

Distancias:

	A	B	C	D	E
A	0	2	9	10	7
B	2	0	6	4	3
C	9	6	0	8	5
D	10	4	8	0	6
E	7	3	5	6	0



Vecindarios:

- [B,A,C,D,E]
- 1 [A,B,C,D,E,A] = 29
 - 2 [C,A,B,D,E,C] = 26
 - 3 [D,A,C,B,E,D] = 34
 - 4 [E,A,C,D,B,E] = 31
 - 5 [B,C,A,D,E,B] = 34
 - 6 [B,D,C,A,E,B] = 31
 - 7 [B,E,C,D,A,B] = 26
 - 8 [B,A,D,C,E,B] = 28
 - 9 [B,A,E,D,C,B] = 29
 - 10 [B,A,C,E,D,B] = 26

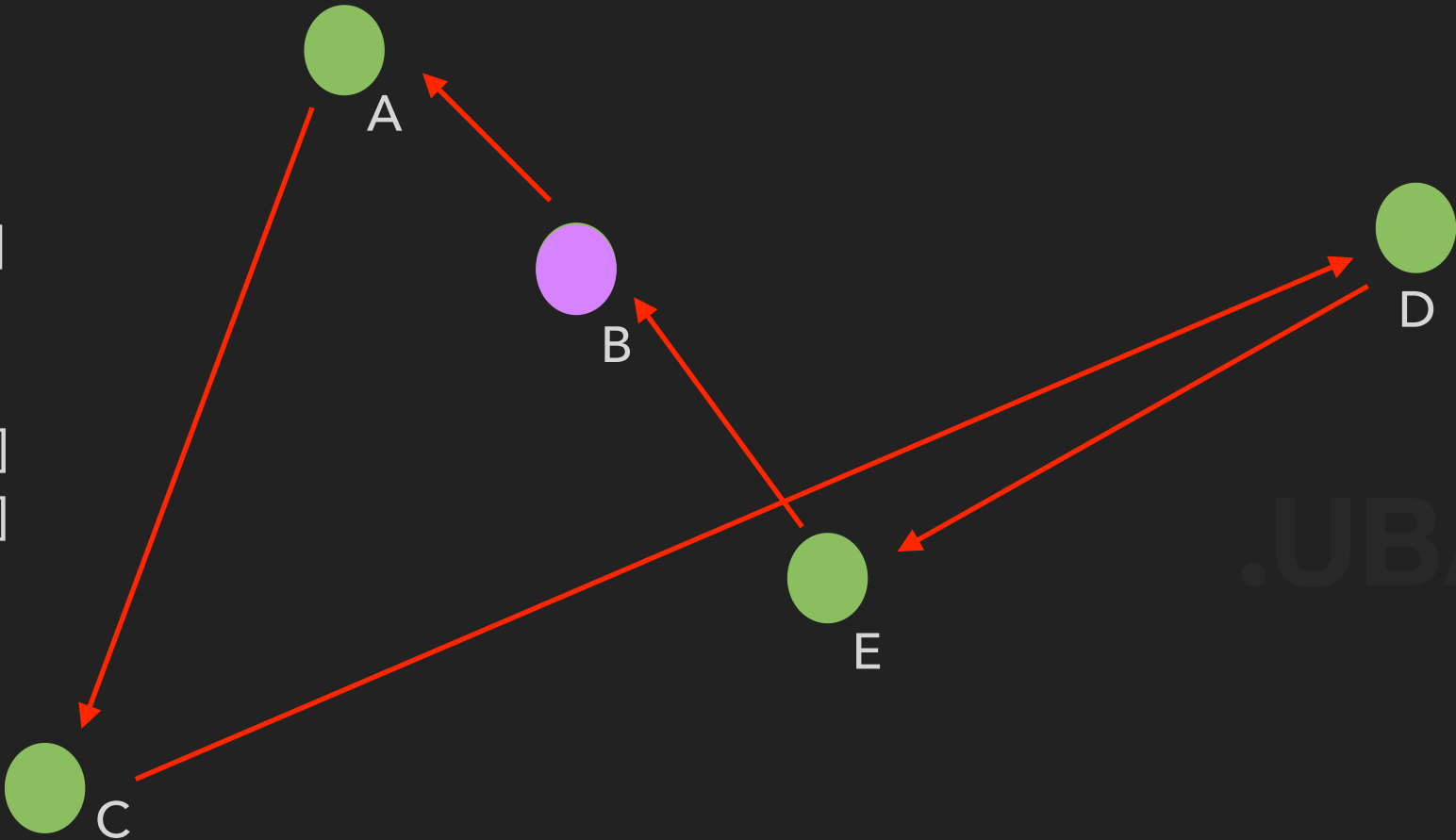
Lista Tabú= [(D,B)]

¿COMO FUNCIONA TS? (CONT.)

Continuar iterando....

Distancias:

	A	B	C	D	E
A	0	2	9	10	7
B	2	0	6	4	3
C	9	6	0	8	5
D	10	4	8	0	6
E	7	3	5	6	0



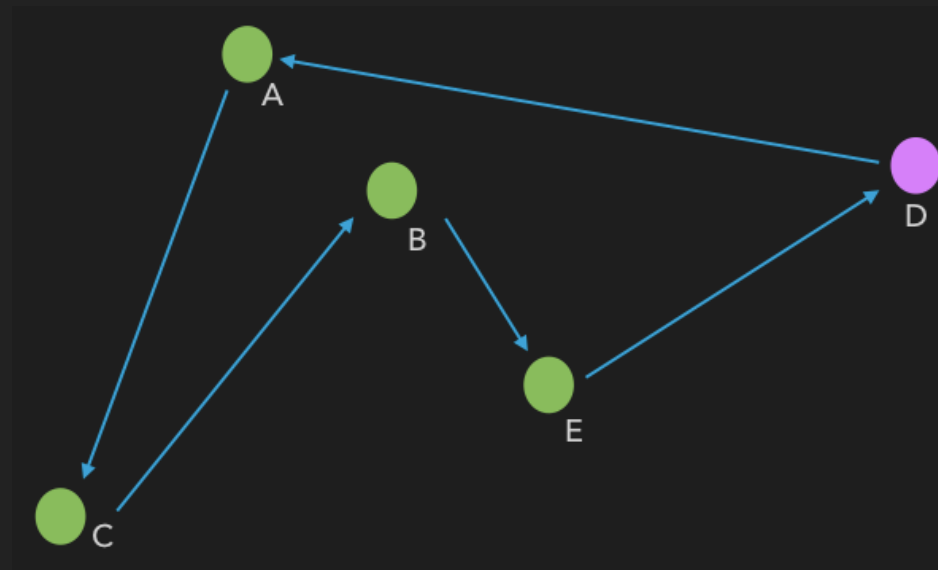
Vecindarios:

- [B,A,C,D,E]
- 1 [A,B,C,D,E,A] = 29
 - 2 [C,A,B,D,E,C] = 26
 - 3 [D,A,C,B,E,D] = 34
 - 4 [E,A,C,D,B,E] = 31
 - 5 [B,C,A,D,E,B] = 34
 - 6 [B,D,C,A,E,B] = 31
 - 7 [B,E,C,D,A,B] = 26
 - 8 [B,A,D,C,E,B] = 28
 - 9 [B,A,E,D,C,B] = 29
 - 10 [B,A,C,E,D,B] = 26

Lista Tabú= [(D,B)]

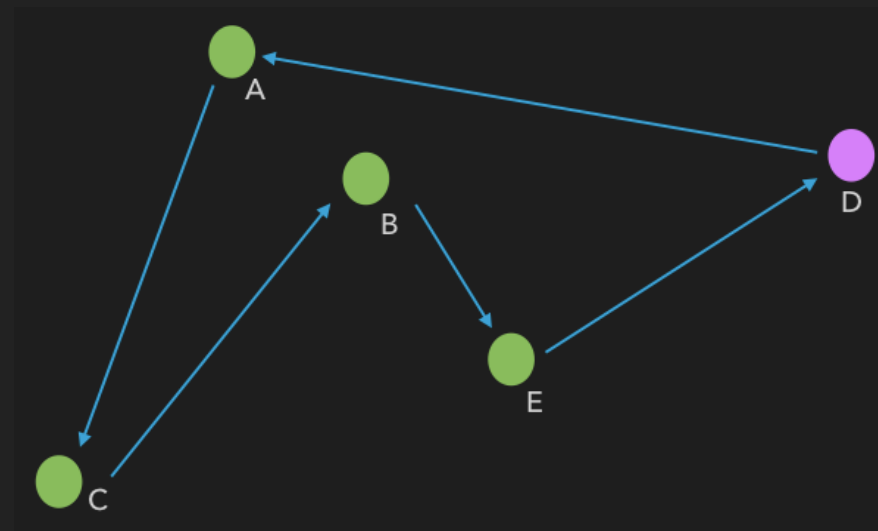
PARA QUÉ SIRVE LA LISTA TABÚ?

- ▶ Su objetivo principal es evitar que la **búsqueda se quede estancada**.
 - ✓ Iteración 1. Si la solución X se mueve hacia $\rightarrow Y \Rightarrow Y$ es mejor
 - ✓ Iteración 2. Puede que el mejor vecino de Y sea X ($X \rightarrow Y \rightarrow X \rightarrow Y \dots$) (puede que no).
- ▶ La lista tabú impide esa situación (volver a X) durante un tiempo.
- ▶ La función la lista tabú es forzar la exploración (diversificación).



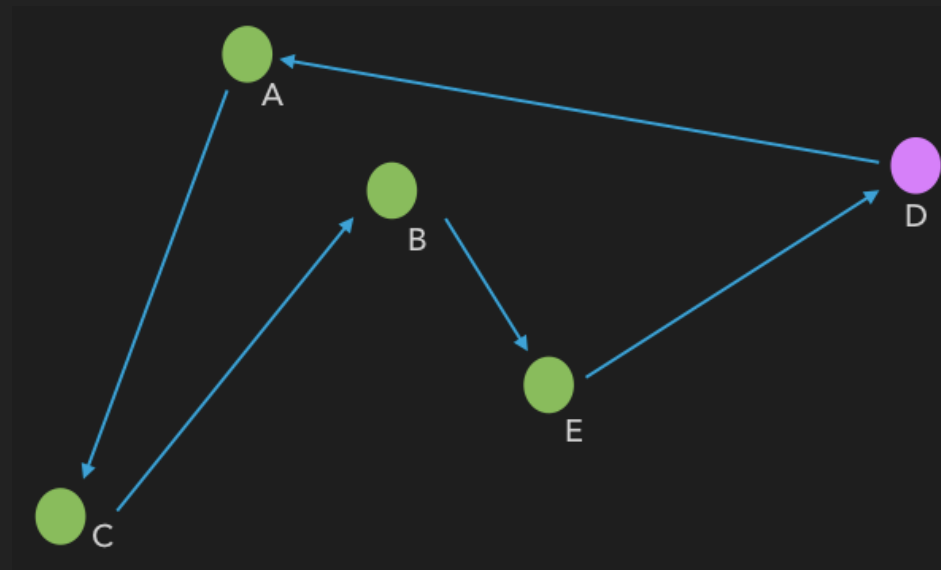
NOCIONES BÁSICAS

- ▶ La palabra **tabú** proviene del idioma tongano, una lengua de la Polinesia, que significa "cosas que no se pueden tocar porque son sagradas"
- ▶ También significa "una prohibición impuesta por costumbre social"
- ▶ En el algoritmo **Tabu Search**, el estado tabú de los elementos de una tabla no es permanente y cambia según el tiempo y las circunstancias, en función de una memoria en evolución.



NOCIONES BÁSICAS

- ▶ El **estado tabú** puede ser anulado por una alternativa mejor.
- ▶ Por lo tanto, TS usa **memoria adaptativa** (flexible).
- ▶ TS también usa **exploración receptiva**, es decir, explotación de buenas soluciones (vecindarios) y exploración de nuevas regiones prometedoras (lista tabú).
- ▶ Se usa mas en problemas de **Optimización Combinatoria**. Existen variantes de TS para Optimización Continua.
- ▶ TS esta basado en **trayectoria** (no en población).



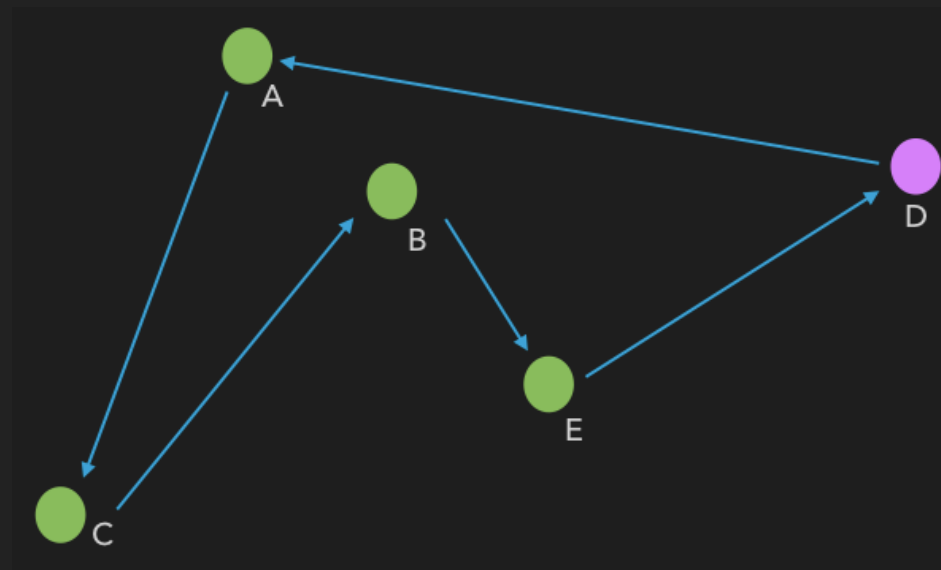
VENTAJAS Y DESVENTAJAS DE TS

► Ventajas:

- ✓ Evita el estancamiento en óptimos locales.
- ✓ Se adapta bien a problemas complejos donde los algoritmos más simples pueden fallar.

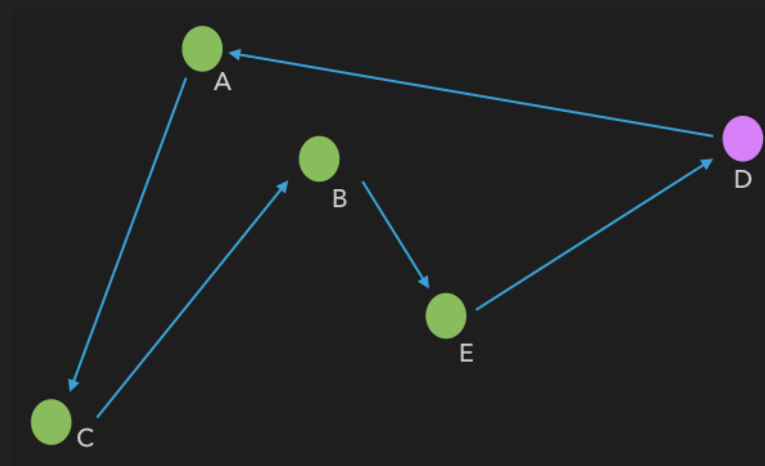
► Desventajas:

- ✓ Requiere un ajuste cuidadoso de parámetros (como el tamaño de la **lista tabú** y los **criterios de aspiración**).
- ✓ Puede ser computacionalmente costoso en algunos casos debido a la exploración extensa del espacio de soluciones.



COMPARACIÓN CON GA

- ▶ GA no permite movimientos que no mejoren, TS sí.
- ▶ GA se basa en procesos semialeatorios (selección, cruza, mutación) que utilizan muestreo, TS es mayoritariamente determinista.
- ▶ GA no tiene memoria explícita, TS sí.
- ▶ ¿Puede una mala elección estratégica producir más información que una buena elección aleatoria? TS afirma que sí.

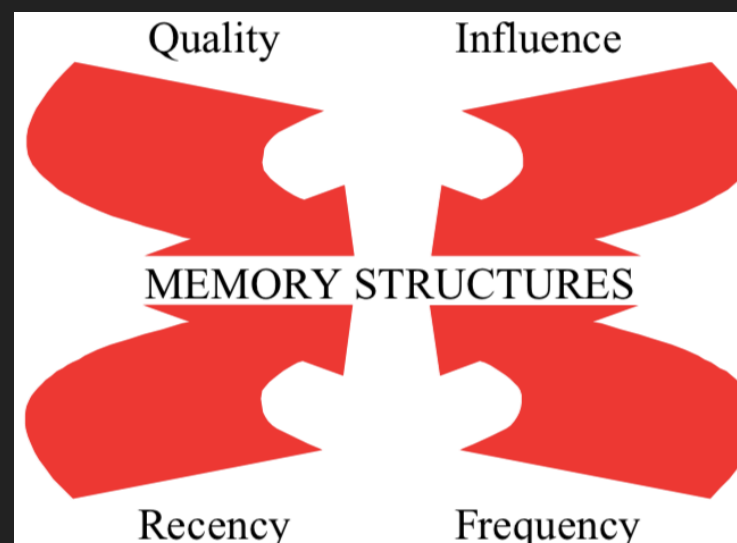


APLICACIONES DE TS

Scheduling Flow-Time Cell Manufacturing Heterogeneous Processor Scheduling Workforce Planning Classroom Scheduling Machine Scheduling Flow Shop Scheduling Job Shop Scheduling Sequencing and Batching	Telecommunications Call Routing Bandwidth Packing Hub Facility Location Path Assignment Network Design for Services Customer Discount Planning Failure Immune Architecture Synchronous Optical Networks
Design Computer-Aided Design Fault Tolerant Networks Transport Network Design Architectural Space Planning Diagram Coherency Fixed Charge Network Design Irregular Cutting Problems	Production, Inventory and Investment Flexible Manufacturing Just-in-Time Production Capacitated MRP Part Selection Multi-item Inventory Planning Volume Discount Acquisition Fixed Mix Investment
Location and Allocation Supply Chain Analysis Multicommodity Location/Allocation Quadratic Assignment Quadratic Semi-Assignment Multilevel Generalized Assignment Lay-Out Planning Off-Shore Oil Exploration	Routing Vehicle Routing Capacitated Routing Time Window Routing Multi-Mode Routing Mixed Fleet Routing Traveling Salesman Traveling Purchaser
Logic and Artificial Intelligence Maximum Satisfiability Probabilistic Logic Clustering Pattern Recognition/Classification Data Integrity Neural Network Training and Design	Graph Optimization Graph Partitioning Graph Coloring Clique Partitioning Maximum Clique Problems Maximum Planner Graphs P-Median Problems
Technology Seismic Inversion Electrical Power Distribution Engineering Structural Design Coordination of Energy Resources Space Station Construction DNA Sequencing Circuit Cell Placement Computer Aided Molecular Design	General Combinational Optimization Zero-One Programming Fixed Charge Optimization Nonconvex Nonlinear Programming All-or-None Networks Bilevel Programming Multi-objective Discrete Optimization Hyperplane Splitting General Mixed Integer Optimization

CUATRO DIMENSIONES DE LA MEMORIA TS (4 ESTRATEGIAS)

- ▶ Memoria basada en lo **reciente** (a corto plazo), tenencia tabú (tabu tenure) estática o dinám.
- ▶ Memoria basada en la **frecuencia** (a largo plazo)
- ▶ **Calidad**: capacidad de diferenciar el mérito de las soluciones
 - ✓ Usa la memoria para identificar elementos comunes a las buenas soluciones.
 - ✓ Refuerza (o desalienta) las acciones que conducen a buenas (o malas) soluciones.
- ▶ **Influencia**: impacto de las decisiones tomadas en la búsqueda tanto en la calidad como en la estructura de las soluciones.



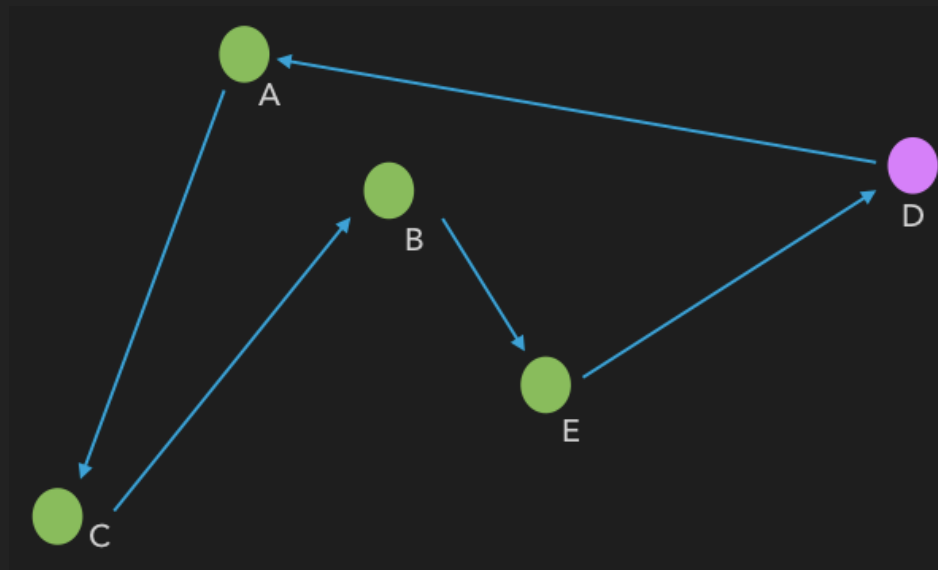
VARIANTES DE GENERACIÓN DE VECINDARIOS

- ▶ **Vecindarios aleatorios** (randomized neighborhoods):
- ▶ **Vecindarios restringidos** (restricted neighborhoods) (ciudades en un radio)
- ▶ **Vecindarios por subestructura** (substructure neighborhoods) (subconjunto de ciudades)
- ▶ **Vecindarios adaptativos** (adaptive neighborhoods) (si no mejora se intercambian 3 ciudades)
- ▶ **Búsqueda del primer mejor** (first-improvement search) (no todas las combinaciones de vecindarios)
- ▶ **Vecindarios basados en fragmentos de soluciones** (solution fragments) (si un fragmento de ciudad es bueno se generan vecindarios de ese fragmento)



CRITERIO DE ASPIRACIÓN

- ▶ Criterio de mejor solución: si una solución tabú encontrada en la iteración actual es mejor que la mejor solución encontrada hasta el momento, entonces su estado tabú se anula.
- ▶ Son posibles otros criterios de aspiración, por ejemplo, establecer la duración tabú más corta para mejores soluciones.



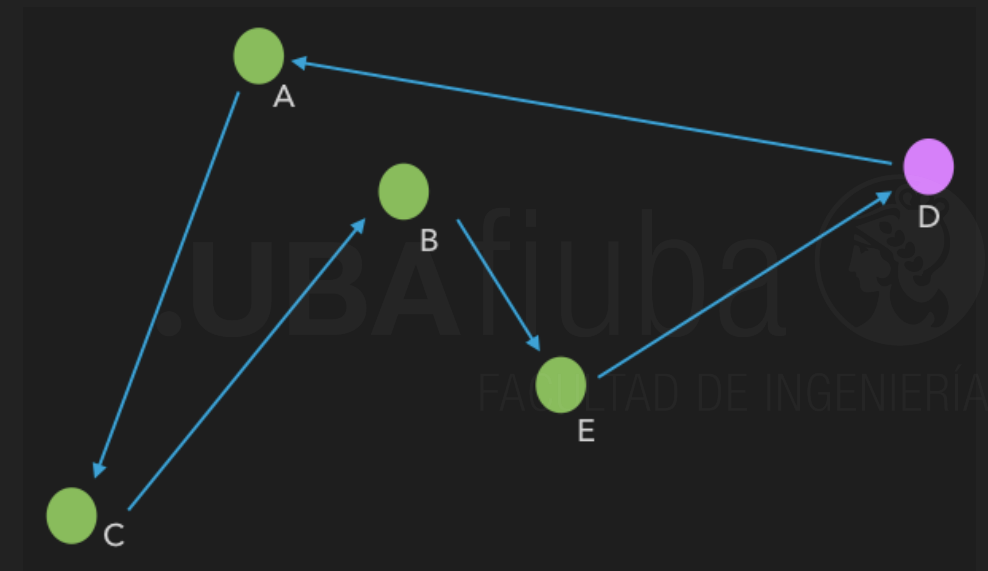
PSEUDOCÓDIGO

INICIALIZACIÓN

1. Generar solución inicial aleatoria
2. Evaluar la solución

BÚSQUEDA

3. Generar vecindario
4. Seleccionar mejor vecino no tabú
5. Actualizar solución
6. Actualizar mejor óptimo
7. Actualizar lista tabú
8. Volver al paso 3.



REFERENCIAS BIBLIOGRÁFICAS Y WEB (III)

- ▶ Glover and Laguna, Tabu search in Pardalos and Resende (eds.), Handbook of Applied Optimization, Oxford Academic Press, 2002.