

Coppa algoritmi 2010

TSP

Sandra Gabaglio

18.05.2010

Struttura dati

```
typedef struct city{  
    int number;  
    float x;  
    float y;  
    int g;  
} city;
```

Nearest Neighbour

- Riceve una città di partenza
- Cerca tra tutte quelle non ancora visitate la più vicina e la aggiunge al tour
- Ad ogni città aggiunta, aggiornano la lunghezza del tour
- Alla fine aggiungo la distanza dall'ultima città alla prima

Nearest Neighbour

Vantaggi

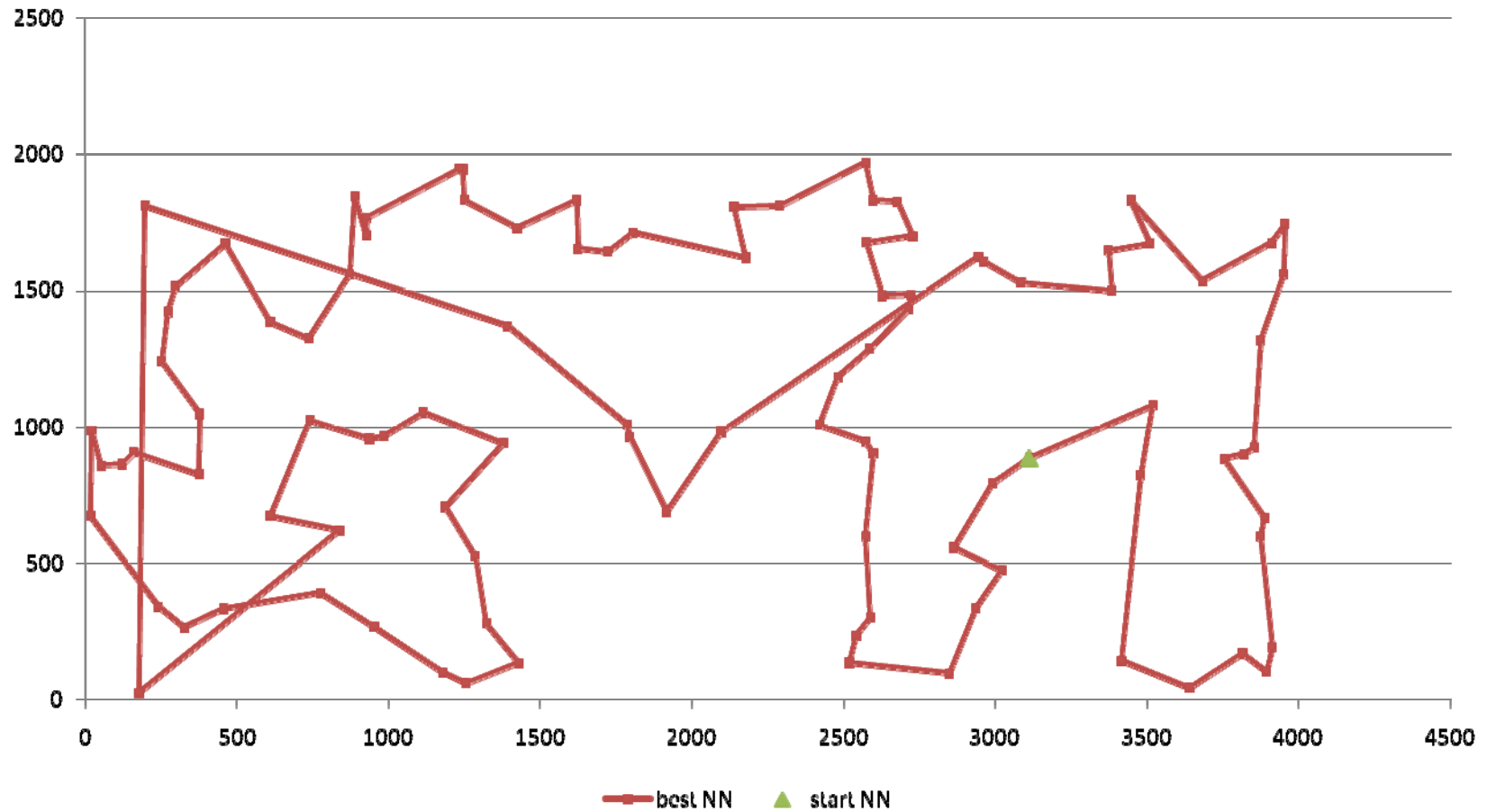
- Trova un buon risultato
- I primi archi sono molto brevi

Svantaggi

- Il risultato non è ottimo
- Gli archi finali sono molto lunghi

Nearest Neighbour

kroa100



2 opt

- Parte da un percorso ammissibile
- Cerca lo scambio che porta il miglior gain
- Effettua lo scambio e aggiorna la lunghezza
- Ricomincia fino a quando non riesce più a migliorare

2 opt

Vantaggi

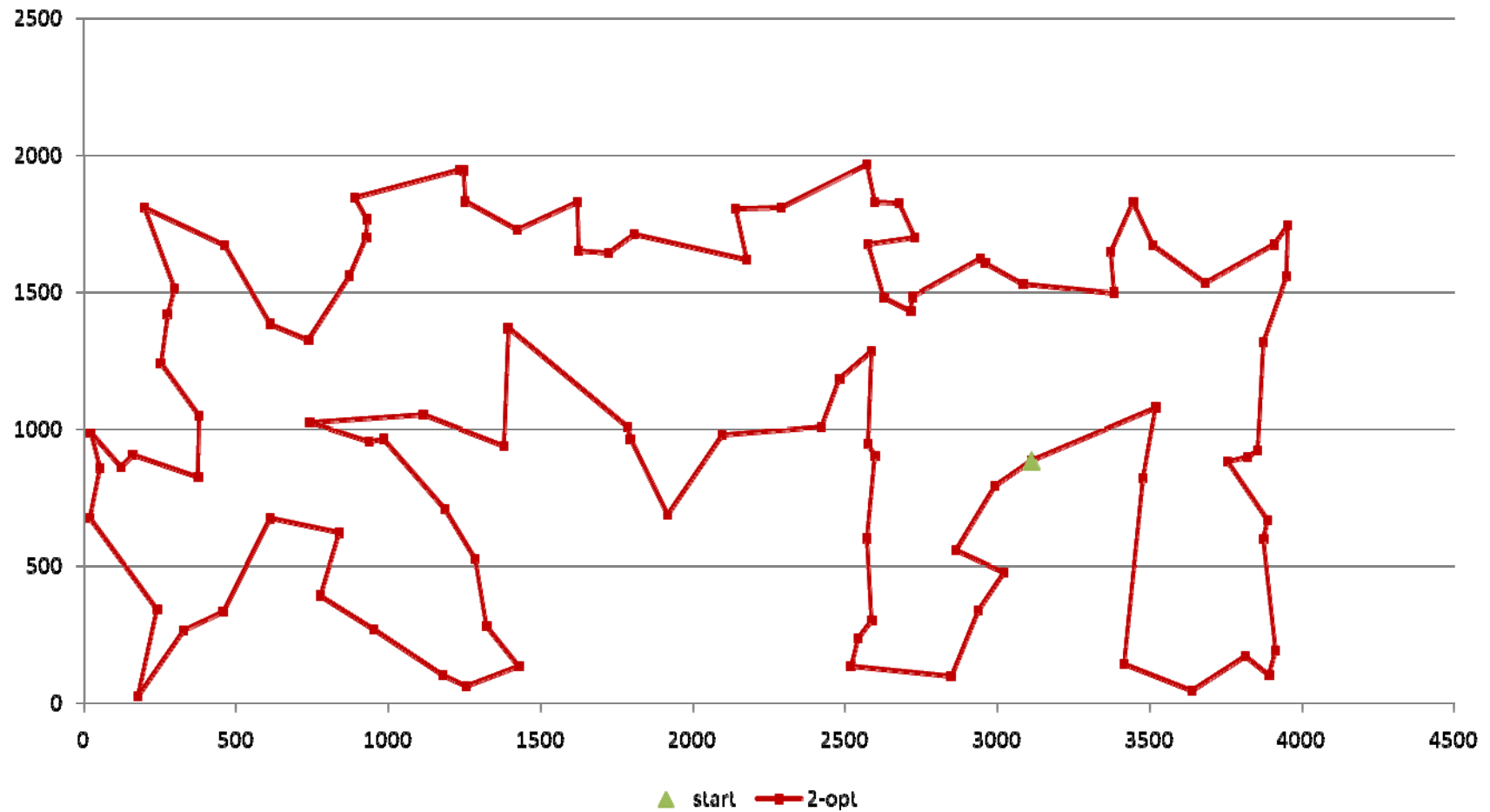
- Toglie tutti gli incroci

Svantaggi

- Il percorso tra le due città deve essere invertito per evitare loop

2 opt

kroa100



Simulated Annealing

- Parte da un percorso ottimizzato
- Effettua uno scambio double bridge random
- Ottimizza con 2 opt
- Se il risultato è migliore lo accetta
- Se il risultato è peggiore, genera random [0..1]
 - Se $\text{random} < e^{-\frac{\Delta E}{T}}$ accetta il peggioramento

Simulated Annealing

- Ripete le stesse operazioni un po' di volte (parametro) prima di decrementare la temperatura
- Si ferma quando la temperatura è arrivata a 0 oppure quando sono passati 180 secondi

Simulated Annealing

Vantaggi

- Cerca un minimo globale
- È facilmente parametrizzabile

Svantaggi

- Bisogna scegliere i parametri giusti per abbassare la temperatura

Double Bridge

- Genero 4 numeri random

```
first = randomint(0, dim/4);
```

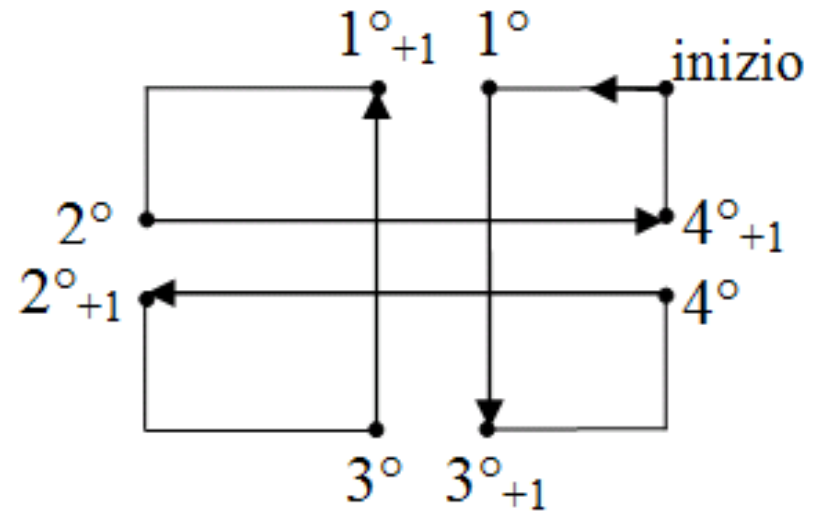
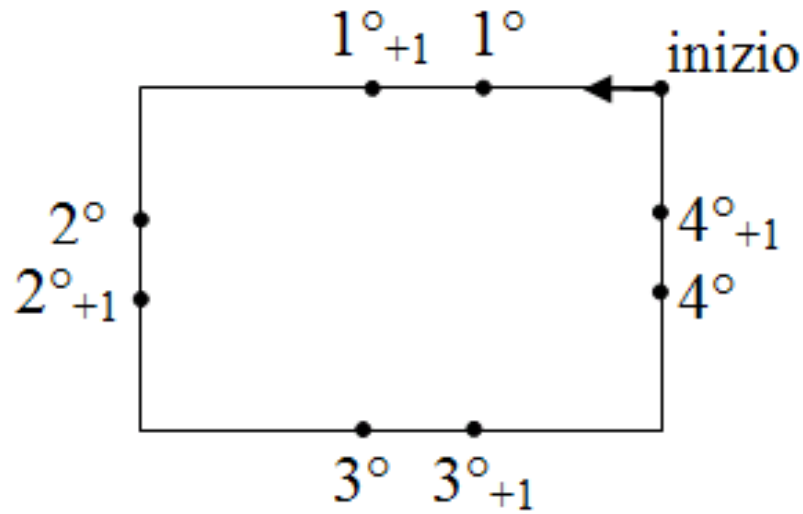
```
second = randomint(first+2, dim/2);
```

```
third = randomint(second+2, 3*dim/4);
```

```
fourth = randomint(third+2, dim-2);
```

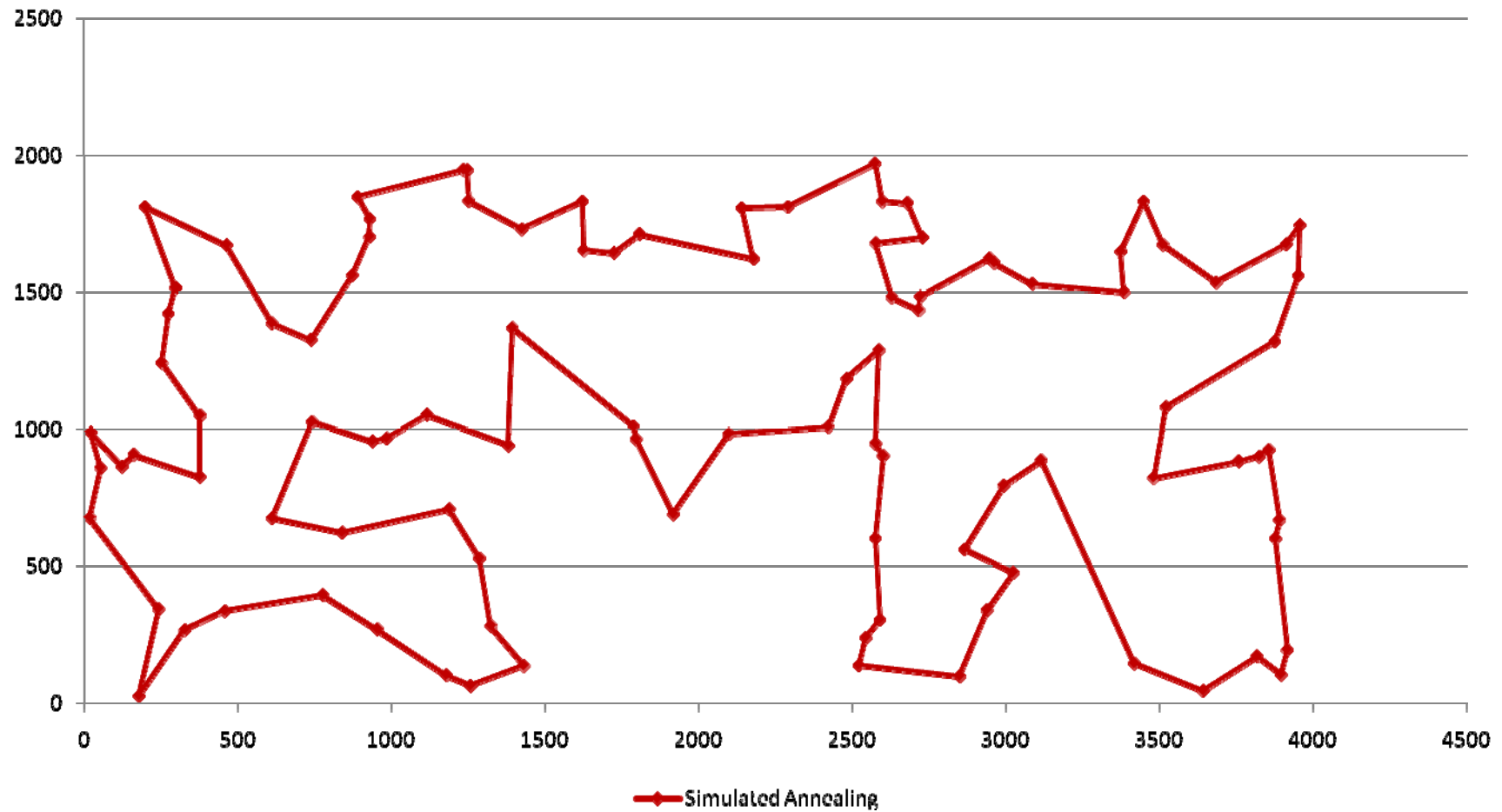
- Con 5 for parziali creo il nuovo percorso

Double Bridge



Simulated annealing

kroa100



Passaggio dei parametri

- Funzione getopt

Possible options:

```
[-r][-b][-n NUM][-i FILE][-o FILE][-a NUM][-t NUM][-c NUM][-f NUM]
```

```
-r          specifica di partire con un nodo iniziale random  
-b          cerca la migliore combinazione di NN+2opt  
-n NUM     imposta il nodo di partenza  
-i FILE    specifica il file del problema di input  
-o FILE    specifica un eventuale file di output  
-a NUM     imposta il numero di tentativi  
-t NUM     imposta la temperatura iniziale per il SA  
-c NUM     imposta le ripetizioni a temp costante per il SA  
-f NUM     imposta il fattore di raffreddamento per il SA
```

Parametri

- Città di partenza
 - Miglior combinazione NN+2opt
- Temperatura iniziale
- Loop a temperatura costante
- Fattore di raffreddamento

Ottimizzazioni

- `memcpy()` per copiare array in SA
- Compilazione con `gcc -O3`
- Tengo sempre aggiornata la lunghezza
- Memorizzo la distanza città-città