

Algoritmo VNS per il Bike sharing Rebalancing Problem

Applicazioni della Ricerca Operativa, A.A. 2020/21

Sezione 1

Descrizione del problema

Descrizione del problema

Introduzione (1/2)

- ▶ Sistema di **bike sharing**: insieme di **stazioni** distribuite su un territorio, all'interno delle quali gli utenti possono prelevare e/o depositare le biciclette messe a disposizione dal sistema
- ▶ Ogni stazione definisce il proprio **livello di occupazione ottimale**
 - ▶ inferiore al totale degli slot disponibili: una parte degli slot deve essere lasciata libera per permettere il deposito di eventuali bike in ingresso
- ▶ Le stazioni aventi un eccesso (o carenza) di bike devono essere sottoposte ad un'operazione di **ribilanciamento** che le riporti al loro livello di occupazione ottimale
- ▶ Ribilanciamento effettuato con una flotta di **veicoli omogenei** aventi **massima capienza**
 - ▶ **Statico**: messo in atto quando il servizio non è attivo (es. durante la notte)

Descrizione del problema

Introduzione (2/2)

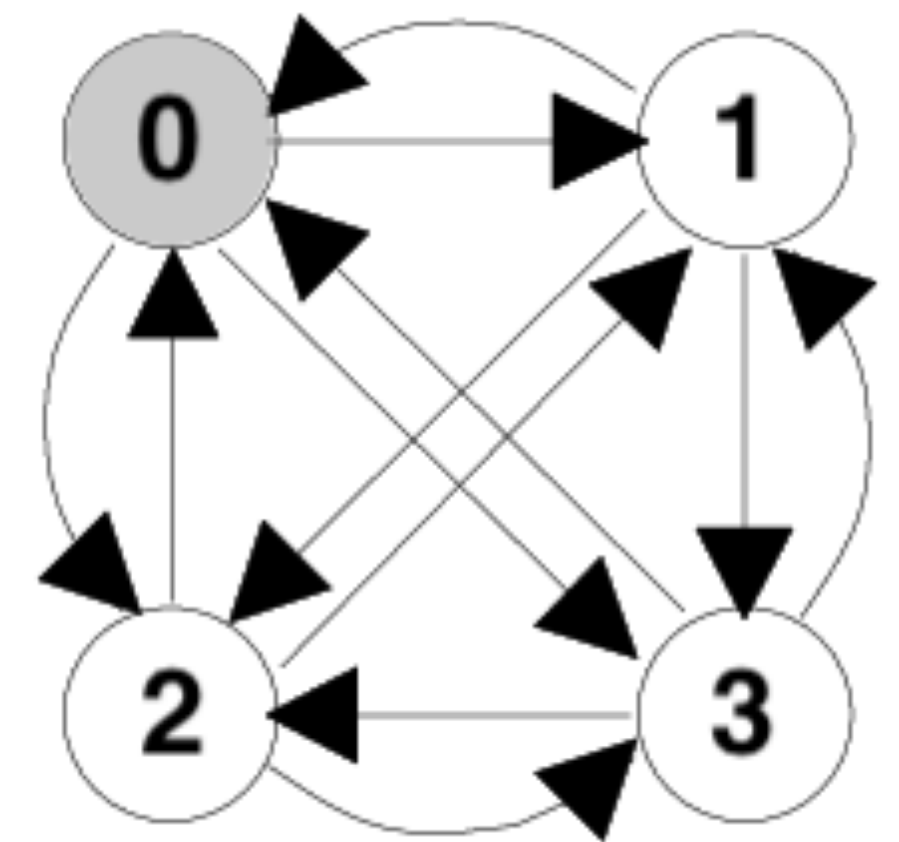
One-commodity Pickup and Delivery Vehicle Routing Problem (1-PDVRP)

- Variante del VRP che prevede il ricollocamento di beni:
 - facenti riferimento ad un singolo tipo di prodotto (**one-commodity**);
 - effettuato mediante l'utilizzo di veicoli in grado di ritirare (**pickup**) e/o depositare (**delivery**) tali prodotti all'interno dei nodi facenti parte della loro tratta.

Descrizione del problema

Formulazione matematica (1/2)

- Problema modellabile con un grafo $G = (V, A)$ **diretto e completo**
- Ogni stazione (nodo) $i \in V$ ha una **demand** di bike pari a $q_i \in \mathbb{Z}$
 - La stazione 0 è il **deposito** ($q_0 = 0$)
- Ogni arco $(i, j) \in A$ ha un **costo** (di percorrenza) pari a $c_{ij} \geq 0$
- Flotta di **veicoli** aventi capacità Q



Descrizione del problema

Formulazione matematica (2/2)

Obiettivo

Determinare un insieme di m route — da assegnare ai veicoli — tale che:

- a) ogni veicolo **parta dal deposito**, visiti una sequenza di stazioni e **rientri al deposito**;
- b) ogni stazione $i \in V \setminus \{0\}$ venga visitata **una sola volta**;
- c) ogni route sia **ammissibile** dal punto di vista della capacità dei veicoli (che possono partire vuoti oppure carichi di un numero di bike inferiore o pari a Q);
- d) il costo totale di percorrenza delle route sia **minimo**.

Sezione 2

Letteratura di riferimento

Letteratura di riferimento

Paper utilizzati per la realizzazione del progetto

Ottimizzazione esatta

1. **The bike sharing rebalancing problem: Mathematical formulations and benchmark instances** — Dell'Amico, Mauro & Hadjiconstantinou, Eleni & Iori, Manuel & Novellani, Stefano (2013)

Algoritmi metaeuristici

2. **A Destroy and Repair Algorithm for the Bike sharing Rebalancing Problem** — Dell'Amico, Mauro & Iori, Manuel & Novellani, Stefano & Stützle, Thomas (2016)
3. **Tabu Search Algorithm for the Bike Sharing Rebalancing Problem** — Pan, Lijun & Liu, Ximei & Xia, Yangkun & Xing, Lining (2020)

Sezione 3

Istanze benchmark

Istanze benchmark

Descrizione delle istanze utilizzate

- ▶ <http://www.or.unimore.it/site/home/online-resources/bike-sharing-rebalancing-problems/documento1090044445.html>
- ▶ Istanze disponibili: **65** (contenenti dati di 22 città, in diverse varianti)
- ▶ Pubblicate nel 2013 da Dell'Amico et al. contestualmente alla pubblicazione di ¹; utilizzate per la valutazione dei risultati sia in ² che in ³



**Possibilità di proporre un nuovo algoritmo metaeuristico di
VNS comparabile con la letteratura esistente**

Istanze benchmark

Struttura delle istanze

14																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

4ReggioEmilia30.txt

- Numero delle stazioni ($|V|$)
- Demand delle stazioni (q_i)
- Capacità dei veicoli (Q)
- Matrice delle distanze (A)

Istanze benchmark

Classificazione delle istanze

- ▶ **Small** — Istanze t.c. $|V| \leq 50$

Bari (**13**), Reggio Emilia (**14**), Bergamo (**15**), Parma (**15**), Treviso (**18**), La Spezia (**20**), Buenos Aires (**21**), Ottawa (**21**), San Antonio (**23**), Brescia (**27**), Roma (**28**), Madison (**28**), Guadalajara (**41**), Dublin (**45**)

- ▶ **Medium** — Istanze t.c. $50 < |V| < 100$

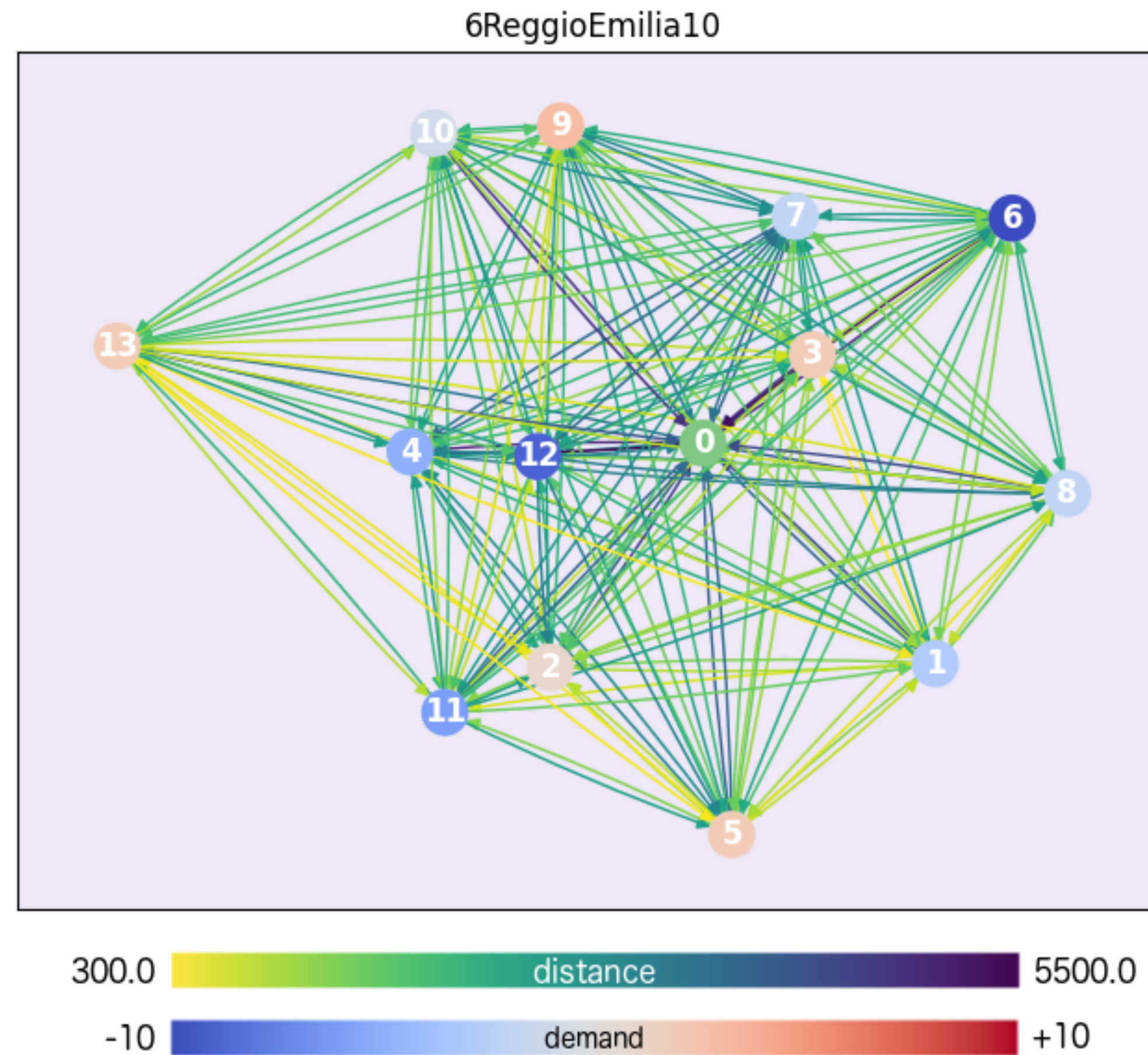
Denver (**51**), Rio De Janeiro (**55**), Boston (**59**), Torino (**75**), Toronto (**80**), Miami (**82**), Ciudad De Mexico (**90**)

- ▶ **Large** — Istanze t.c. $|V| \geq 100$

Minneapolis (**116**)

Istanze benchmark

Esempio di istanza




Rappresentazione grafica dell'istanza 6ReggioEmilia10.txt (posizioni delle stazioni determinate dall'utilizzo di un layout force-based)

Sezione 4

Dettagli implementativi

Dettagli implementativi

Panoramica generale

- ▶ **Linguaggio utilizzato:** Python (versione dell'interprete 3.7.2) 
- ▶ **Implementazione:** schema classico di VNS (shake+local search a rotazione su una coda di metodi di generazione neighborhood)
 - ▶ **Metodo costruttivo:** algoritmo Savings&Losses proposto in ²
 - ▶ **Definizioni di neighborhood utilizzate:** 7 definizioni proposte in ²

Dettagli implementativi

Metodo costruttivo

- Algoritmo utilizzato per la generazione della soluzione iniziale: **Savings&Losses** (proposto in ²)
 - Adattamento dell'algoritmo di *savings* di Clarke-Wright ⁴
- Funzionamento
 1. Data un'istanza $G = (V, A)$ si generano $|V|$ route (una per ogni stazione)
 2. Si considerano tutti i merge ammissibili tra coppie di route (P, R) ; per ogni merge, viene valutata una **funzione** $E_{P \oplus R}$ **basata su un trade-off** tra il risparmio di costo (“saving”) che implicherebbe il merge e la perdita di flessibilità (“loss”) che ne deriverebbe — in termini di possibilità di effettuare ulteriori merge. **Il merge avente $E_{P \oplus R}$ più alta viene applicato**
 3. Si ripete il processo dal punto 2 fino a quando non è più possibile effettuare alcun merge

⁴ Scheduling of vehicles from a central depot to a number of delivery points — Clarke G., Wright J. (1964)

Dettagli implementativi

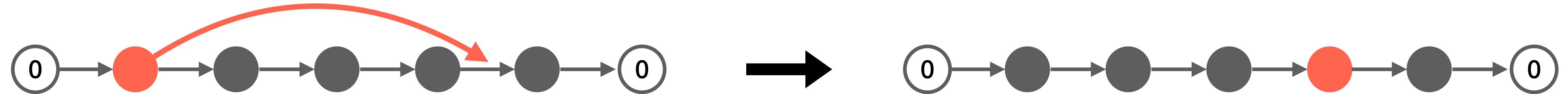
Definizione dei neighborhood

- Implementazione dei neighborhood proposti in ² (con relativi controlli di ammissibilità, basati sull'utilizzo di *load windows*):
 - **Move** — 1° in ordine di esecuzione
 - **Swap(1,1)** — 2° in ordine di esecuzione
 - **Swap(2,2)** — 3° in ordine di esecuzione
 - **Swap(1,1,1)** — 4° in ordine di esecuzione
 - **Cross** — 5° in ordine di esecuzione
 - **Cross(3)** — 7° in ordine di esecuzione
 - **Or-opt(κ)** — 6° in ordine di esecuzione
- Generazione dei neighborhood **parallelizzata** su N processori per migliorare le performance

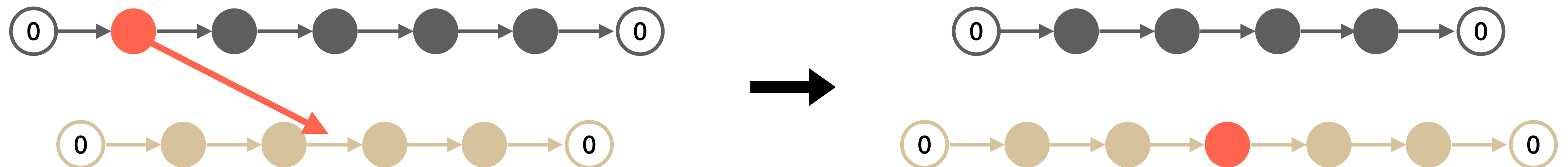
Dettagli implementativi

Definizione dei neighborhood: **Move**

Variante A — **Single route**



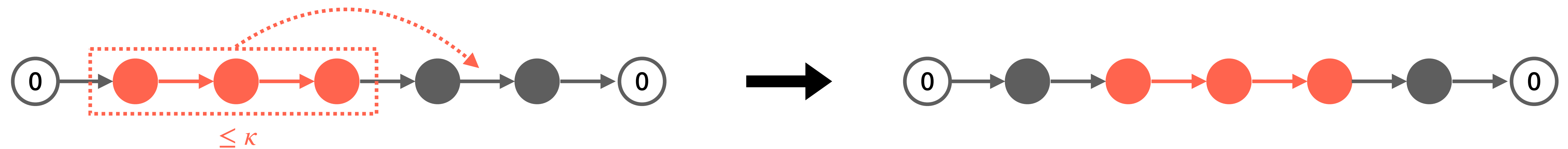
Variante B — **Two routes**



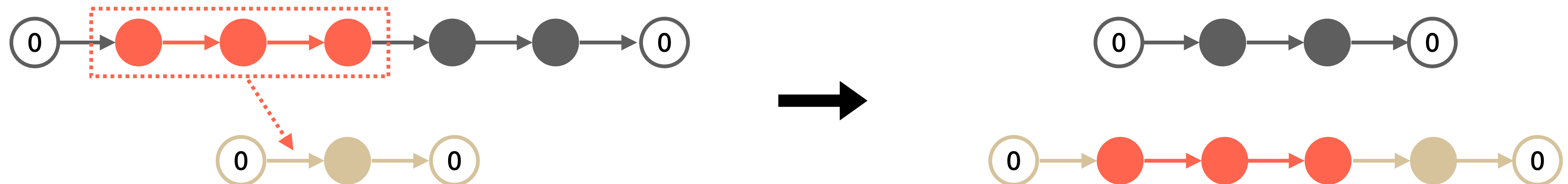
Dettagli implementativi

Definizione dei neighborhood: **Or-opt**(κ)

Variante A — **Single route** ($\forall r \in R : 1 < |r| \leq \min\{\kappa, |R|\}$)



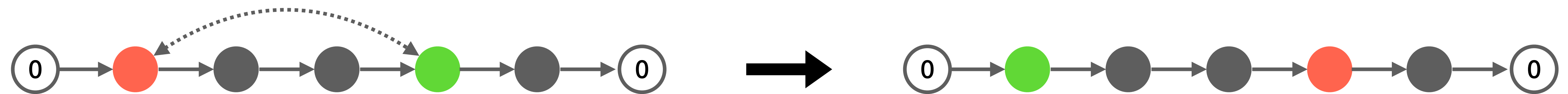
Variante B — **Two routes** ($\forall r \in R : 1 < |r| \leq \min\{\kappa, |R|\}$)



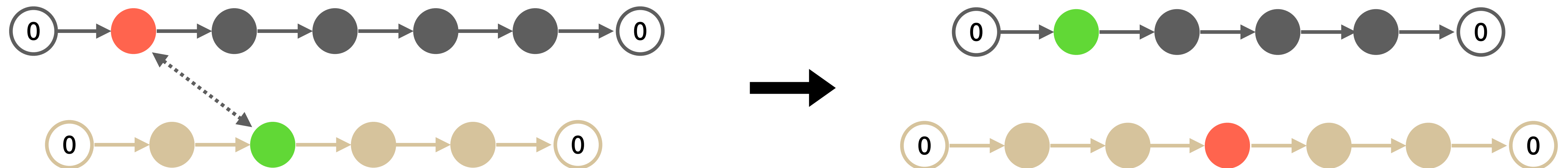
Dettagli implementativi

Definizione dei neighborhood: **Swap(1,1)**

Variante A — **Single route**



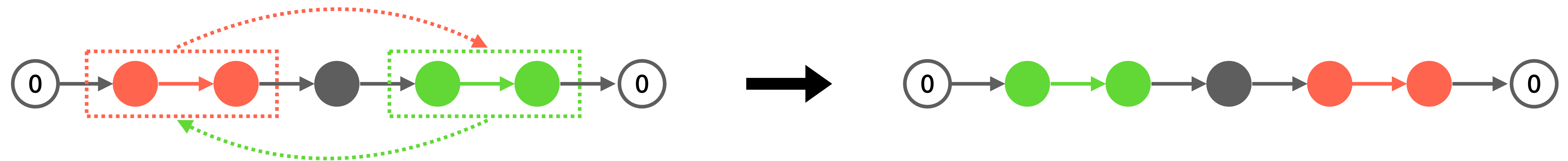
Variante B — **Two routes**



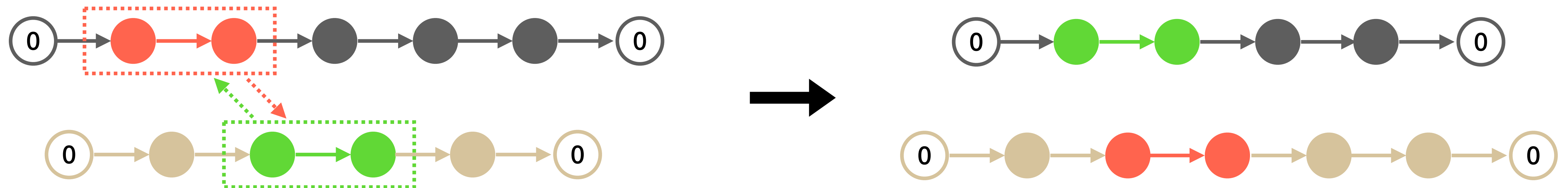
Dettagli implementativi

Definizione dei neighborhood: **Swap(2,2)**

Variante A — **Single route**



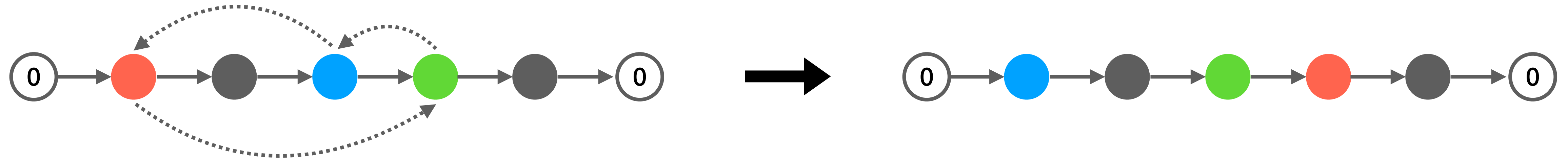
Variante B — **Two routes**



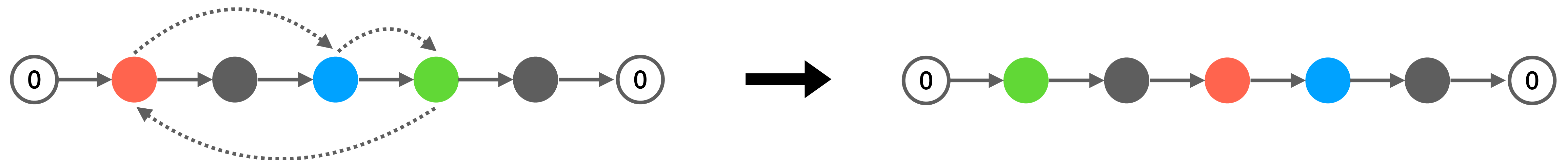
Dettagli implementativi

Definizione dei neighborhood: **Swap(1,1,1)** (1/3)

Variante A.1 — **Single route**, $v_1 \leftarrow v_2 \leftarrow v_3$



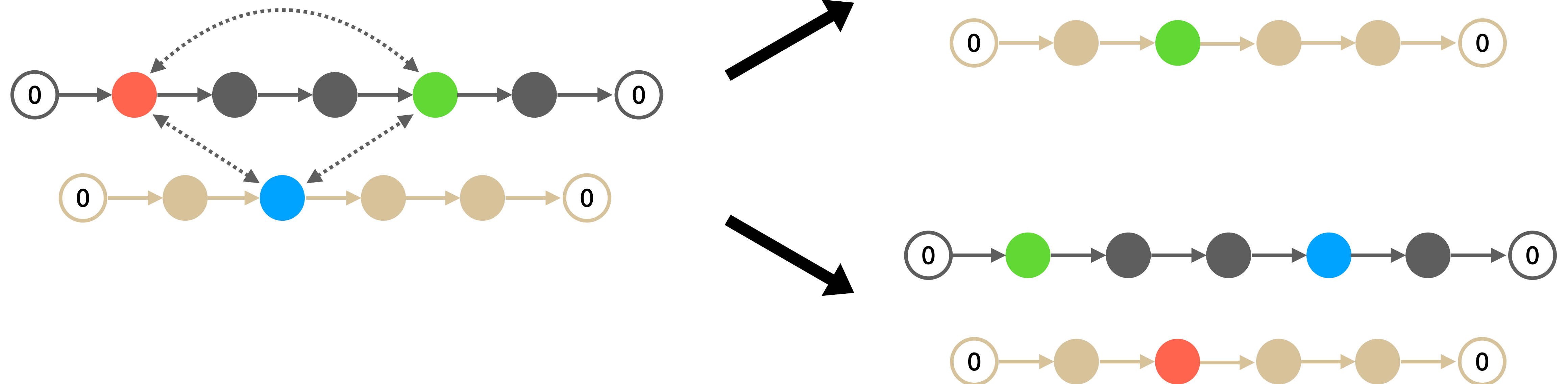
Variante A.2 — **Single route**, $v_1 \rightarrow v_2 \rightarrow v_3$



Dettagli implementativi

Definizione dei neighborhood: **Swap(1,1,1)** (2/3)

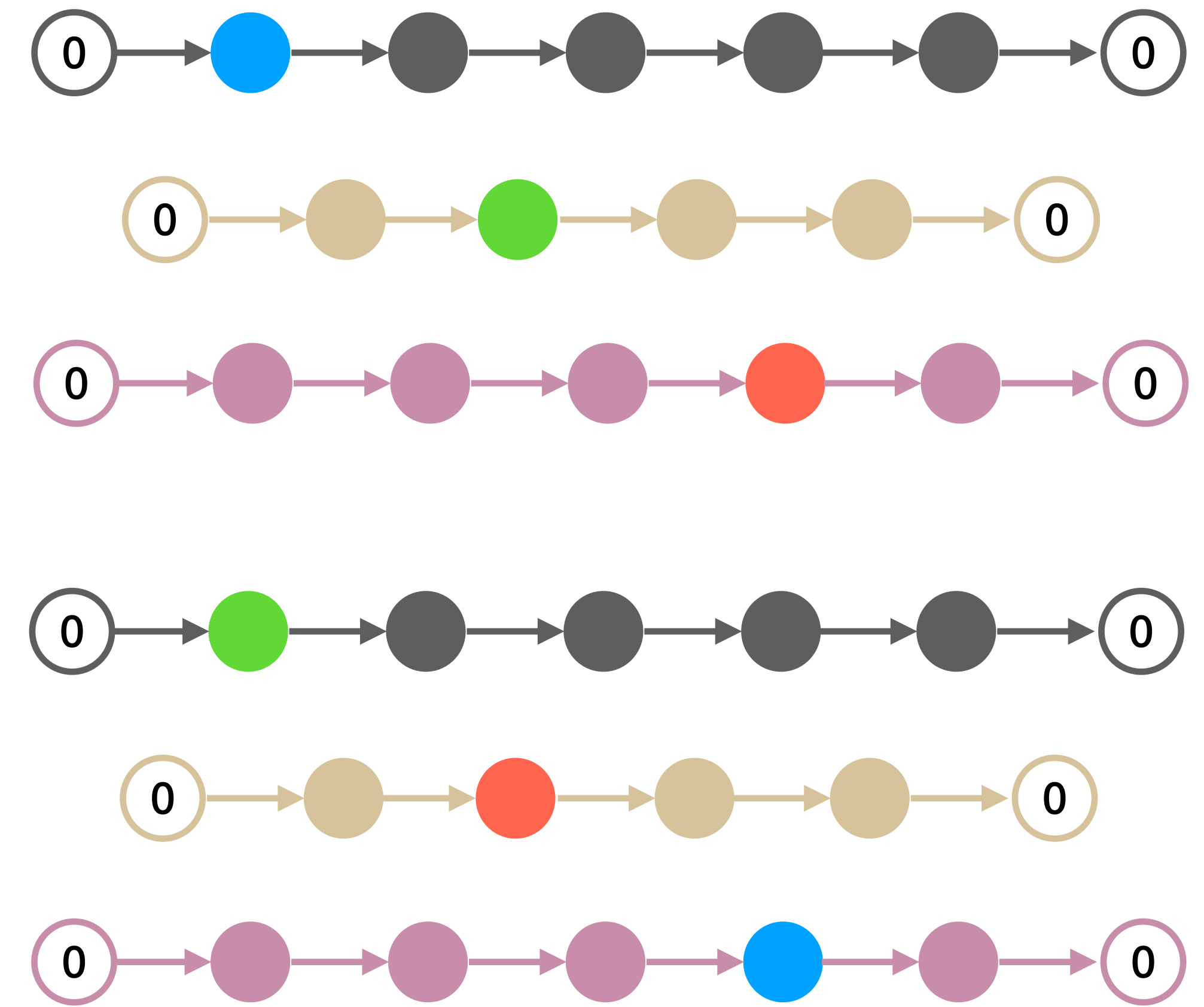
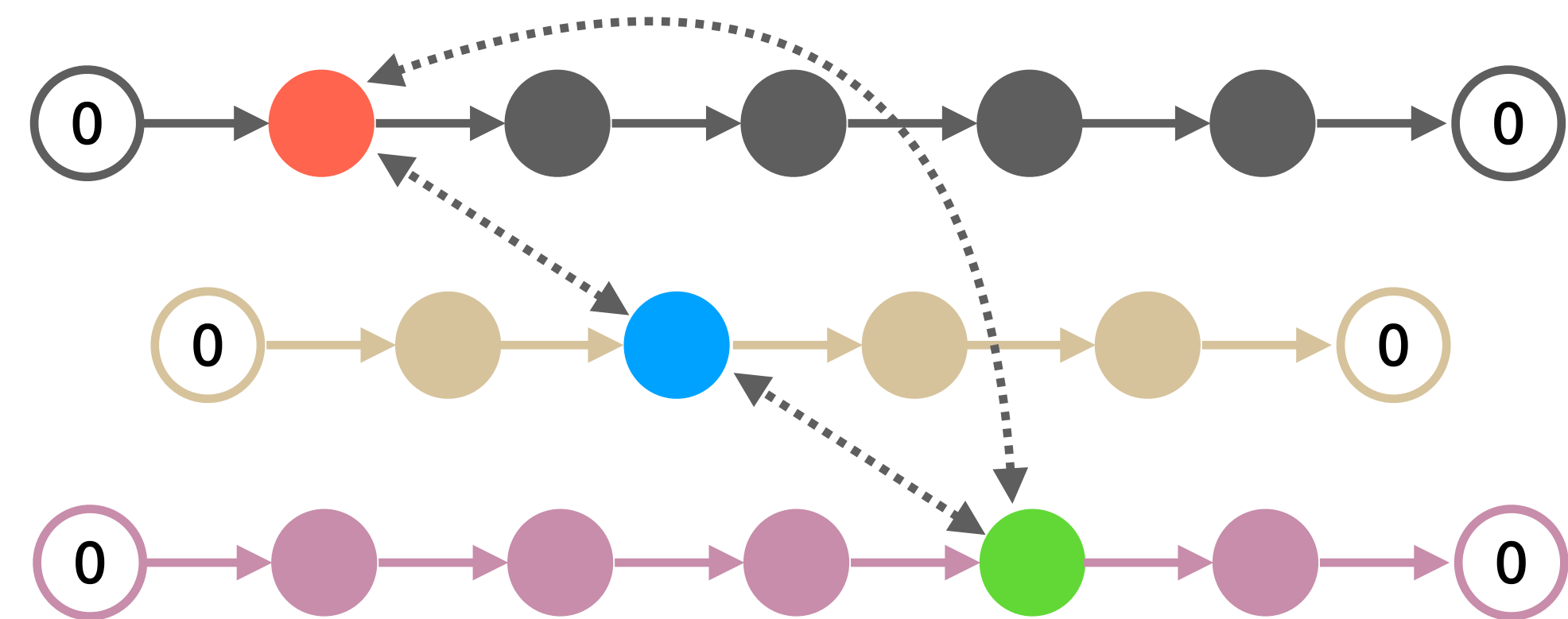
Varianti B.1 e B.2 — **Two routes**



Dettagli implementativi

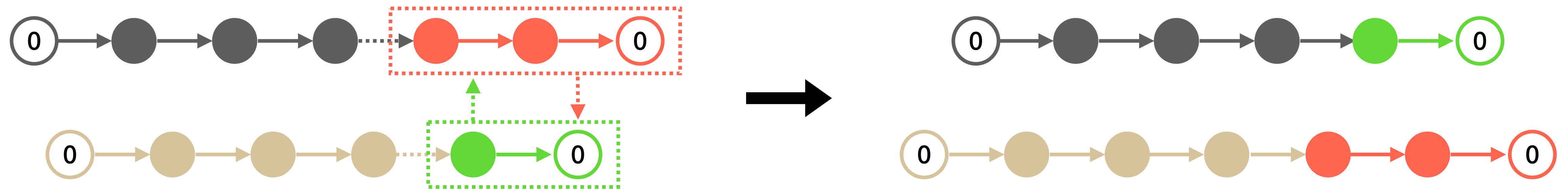
Definizione dei neighborhood: **Swap(1,1,1)** (3/3)

Varianti C.1 e C.2 — **Three routes**



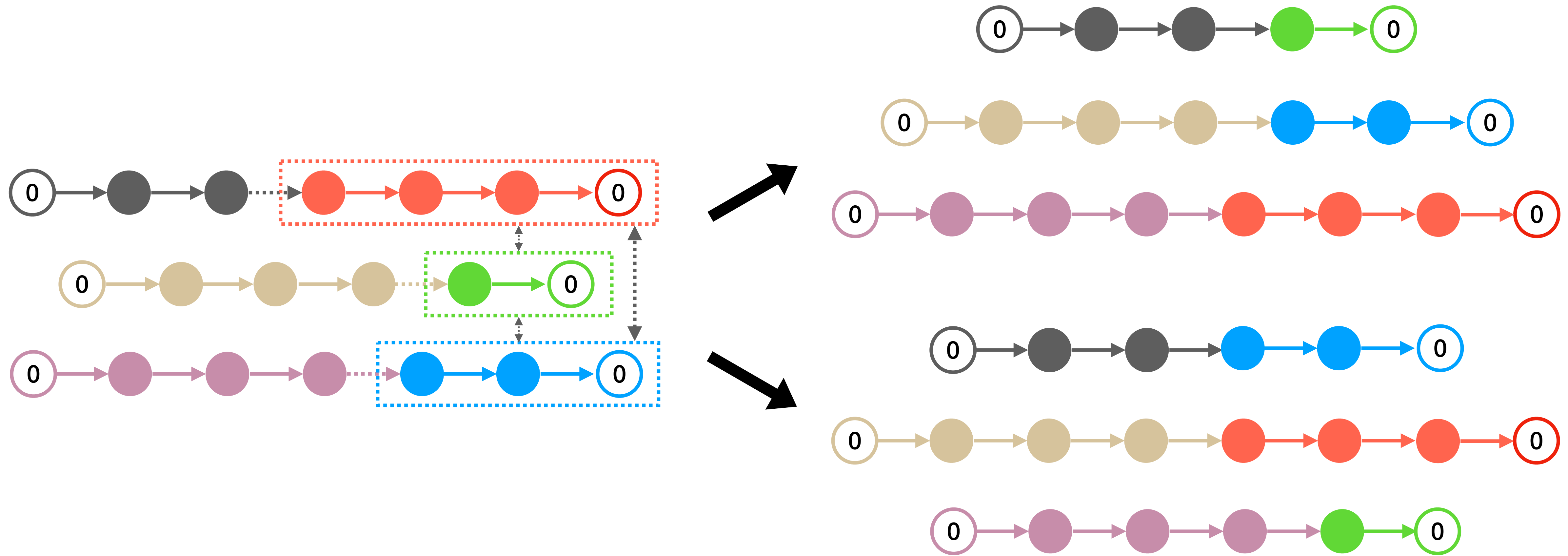
Dettagli implementativi

Definizione dei neighborhood: **Cross**



Dettagli implementativi

Definizione dei neighborhood: **Cross(3)**



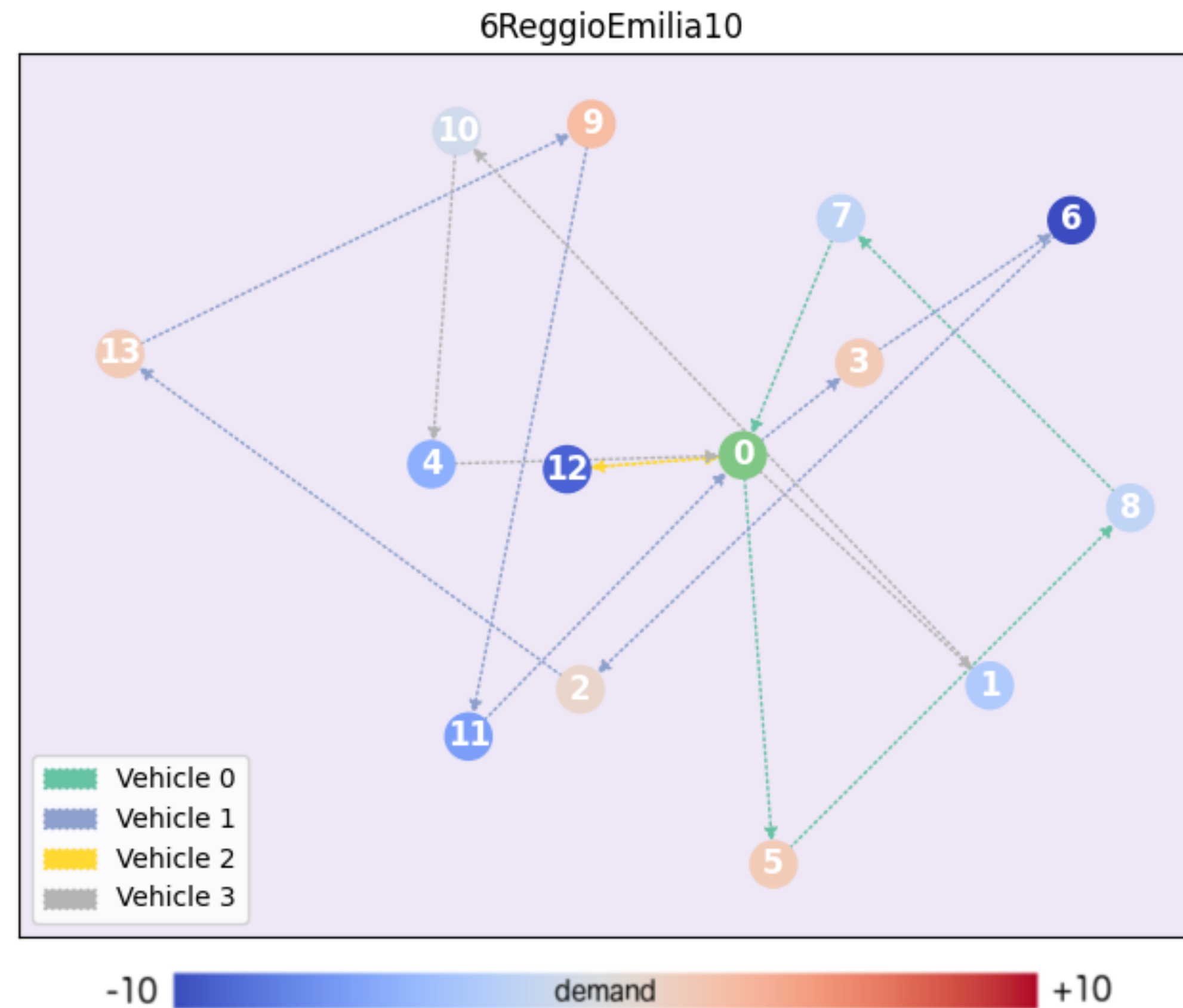
Dettagli implementativi

Condizioni di terminazione dell'algoritmo

- L'esecuzione dell'algoritmo VNS termina quando
 - a) viene raggiunto il **massimo numero di iterazioni** (max_it=5000)
 - b) la **stessa soluzione** risulti essere la migliore per un **determinato numero di volte consecutive** (max_consecutive_solutions=20)
 - c) viene raggiunto un **limite di tempo** massimo (in secondi)
 - max_time=10 per le istanze *small*
 - max_time=600 per le istanze *medium*
 - max_time=1800 per le istanze *large*

Dettagli implementativi

Esempio di soluzione finale



Rappresentazione grafica di soluzione ottenuta sull'istanza 6ReggioEmilia10.txt ($z_{\min} = 40700$)

Sezione 5

Confronto dei risultati

Confronto dei risultati

Generazione dei risultati

- ▶ Numero di run per istanza: **10**
- ▶ Al termine dell'esecuzione delle 10 run su una singola istanza, si memorizzano
 - ▶ la **migliore soluzione** trovata (ed il valore z_{\min} ad essa associato)
 - ▶ lo **scostamento percentuale medio** $\text{gap}_{\%avg}$ tra i valori delle soluzioni trovate ed il valore della soluzione migliore (valutazione di robustezza)
 - ▶ il **tempo di esecuzione medio** t_{avg} delle run
- ▶ Test eseguiti su processore *Intel Core i5 (dual-core) 2.4GHz*
- ▶ Parametri utilizzati: $\alpha = 0.7335$ (metodo costruttivo) e $\kappa = 35$ (neighborhood search)

Confronto dei risultati

Descrizione dei dati comparati

- ▶ I risultati ottenuti dalle run VNS sono stati comparati con quelli riportati nei paper di riferimento, ovvero
 - ▶ algoritmo di **Branch-and-cut (B&C)** di Dell'Amico et al., 2013
 - ▶ algoritmo metaeuristico di **Destroy and Repair (DR)** di Dell'Amico et al., 2016
 - ▶ algoritmo metaeuristico di **Tabu Search (TS)** di Pan et al., 2020
- ▶ **Focus:** valutazione del divario percentuale tra le migliori soluzioni ottenute tramite VNS e le migliori soluzioni (o upper/lower bound) ottenute dai metodi sopra riportati

Confronto dei risultati

Risultati ottenuti sulle istanze **small** (1/2)

VNS							B&C		DR		TS	
original_idx	city	stations_count	vehicles_capacity	local_optimum_value	avg_percentage_gap	avg_execution_time	local_optimum_value	vns_gap	local_optimum_value	vns_gap	local_optimum_value	vns_gap
1	Bari	13	30	15400	0,0000%	9,5441	14600	▼ -5,19%	14600	▼ -5,19%	14600	▼ -5,19%
2	Bari	13	20	17900	0,0000%	15,5009	15700	▼ -12,29%	15700	▼ -12,29%	15700	▼ -12,29%
3	Bari	13	10	21900	0,0000%	10,7534	20600	▼ -5,94%	20600	▼ -5,94%	20600	▼ -5,94%
4	ReggioEmilia	14	30	20400	0,0000%	7,5812	16900	▼ -17,16%	16900	▼ -17,16%	16900	▼ -17,16%
5	ReggioEmilia	14	20	23800	4,9160%	15,2706	23200	▼ -2,52%	23200	▼ -2,52%	23200	▼ -2,52%
6	ReggioEmilia	14	10	34300	16,0058%	15,2058	32500	▼ -5,25%	32500	▼ -5,25%	32500	▼ -5,25%
7	Bergamo	15	30	12800	3,3594%	12,2855	12600	▼ -1,56%	12600	▼ -1,56%	12600	▼ -1,56%
8	Bergamo	15	20	13000	2,8462%	15,2189	12700	▼ -2,31%	12700	▼ -2,31%	12700	▼ -2,31%
9	Bergamo	15	12	14700	2,0408%	15,1366	13500	▼ -8,16%	13500	▼ -8,16%	13500	▼ -8,16%
10	Parma	15	30	29600	4,3581%	13,9763	29000	▼ -2,03%	29000	▼ -2,03%	29000	▼ -2,03%
11	Parma	15	20	30500	2,2951%	14,2994	29000	▼ -4,92%	29000	▼ -4,92%	29000	▼ -4,92%
12	Parma	15	10	32900	3,2827%	15,5671	32500	▼ -1,22%	32500	▼ -1,22%	32500	▼ -1,22%
13	Treviso	18	30	29858	0,0389%	15,4828	29259	▼ -2,01%	29259	▼ -2,01%	29259	▼ -2,01%
14	Treviso	18	20	32897	6,0236%	15,1903	29259	▼ -11,06%	29259	▼ -11,06%	29259	▼ -11,06%
15	Treviso	18	10	32064	6,4172%	15,1617	31443	▼ -1,94%	31443	▼ -1,94%	31443	▼ -1,94%
16	LaSpezia	20	30	21705	0,6823%	15,6408	20746	▼ -4,42%	20746	▼ -4,42%	20746	▼ -4,42%
17	LaSpezia	20	20	21705	1,2195%	15,5943	20746	▼ -4,42%	20746	▼ -4,42%	20746	▼ -4,42%
18	LaSpezia	20	10	24971	1,6091%	15,3735	22811	▼ -8,65%	22811	▼ -8,65%	22811	▼ -8,65%
19	BuenosAires	21	30	92673	0,0000%	7,1586	76999	▼ -16,91%	76999	▼ -16,91%	76999	▼ -16,91%
20	BuenosAires	21	20	109383	0,0000%	15,0643	91619	▼ -16,24%	91619	▼ -16,24%	91619	▼ -16,24%

(continua in slide successiva)

Confronto dei risultati

Risultati ottenuti sulle istanze **small** (2/2)

(continua da slide precedente)

21	Ottawa	21	30	18091	0,0000%	15,6832	16202	▼	-10,44%	16202	▼	-10,44%	16202	▼	-10,44%
22	Ottawa	21	20	18091	7,0941%	16,0385	16202	▼	-10,44%	16202	▼	-10,44%	16202	▼	-10,44%
23	Ottawa	21	10	23005	0,1552%	15,1336	17576	▼	-23,60%	17576	▼	-23,60%	17576	▼	-23,60%
24	SanAntonio	23	30	24736	2,5590%	17,2101	22982	▼	-7,09%	22982	▼	-7,09%	22982	▼	-7,09%
25	SanAntonio	23	20	30477	0,0607%	15,3498	24007	▼	-21,23%	24007	▼	-21,23%	24007	▼	-21,23%
26	SanAntonio	23	10	46467	2,8457%	15,7778	40149	▼	-13,60%	40149	▼	-13,60%	40149	▼	-13,60%
27	Brescia	27	30	31700	6,3091%	15,8634	30300	▼	-4,42%	30300	▼	-4,42%	30300	▼	-4,42%
28	Brescia	27	20	36700	5,3951%	16,7602	31100	▼	-15,26%	31100	▼	-15,26%	31100	▼	-15,26%
29	Brescia	27	11	44800	1,1830%	13,3101	35200	▼	-21,43%	35200	▼	-21,43%	35200	▼	-21,43%
30	Roma	28	30	69700	0,2439%	13,8279	61900	▼	-11,19%	61900	▼	-11,19%	61900	▼	-11,19%
31	Roma	28	20	72500	0,0000%	10,5244	66600	▼	-8,14%	66600	▼	-8,14%	66600	▼	-8,14%
32	Roma	28	18	73400	0,0000%	13,6957	68300	▼	-6,95%	68300	▼	-6,95%	68300	▼	-6,95%
33	Madison	28	30	40510	0,0000%	24,8250	29246	▼	-27,81%	29246	▼	-27,81%	29246	▼	-27,81%
34	Madison	28	20	37828	2,0384%	15,1754	29839	▼	-21,12%	29839	▼	-21,12%	29839	▼	-21,12%
35	Madison	28	10	47063	1,0915%	19,5142	33848	▼	-28,08%	33848	▼	-28,08%	33848	▼	-28,08%
36	Guadalajara	41	30	60853	0,8938%	54,4042	57476	▼	-5,55%	57476	▼	-5,55%	57476	▼	-5,55%
37	Guadalajara	41	20	64186	0,1980%	30,5760	59493	▼	-7,31%	59493	▼	-7,31%	59493	▼	-7,31%
38	Guadalajara	41	11	74577	0,0000%	15,1853	64981	▼	-12,87%	64981	▼	-12,87%	64981	▼	-12,87%
39	Dublin	45	30	42224	1,2401%	18,1110	33548	▼	-20,55%	33548	▼	-20,55%	33548	▼	-20,55%
40	Dublin	45	20	47230	5,0472%	32,9281	39786	▼	-15,76%	39786	▼	-15,76%	39786	▼	-15,76%
41	Dublin	45	11	65446	2,8825%	15,7083	54392	▼	-16,89%	54392	▼	-16,89%	54392	▼	-16,89%
Avg.					2,3008%			▼	-10,83%		▼	-10,83%		▼	-10,83%

Confronto dei risultati

Risultati ottenuti sulle istanze medium

VNS							B&C					DR		TS	
original_idx	city	stations_count	vehicles_capacity	local_optimum_value	avg_percentage_gap	avg_execution_time	local_optimum_LB	local_optimum_UB	LB_UB_gap	vns_gap_LB	vns_gap_UB	local_optimum	vns_gap	local_optimum	vns_gap
42	Denver	51	30	55677	4,0297%	650,5250	51583	51583	0,0000%	▼ -7,35%	▼ -7,35%	51583	▼ -7,35%	51583	▼ -7,35%
43	Denver	51	20	59183	1,5207%	527,1145	53465	53465	0,0000%	▼ -9,66%	▼ -9,66%	53465	▼ -9,66%	53465	▼ -9,66%
44	Denver	51	10	81739	2,3831%	619,6249	67459	67459	0,0000%	▼ -17,47%	▼ -17,47%	67459	▼ -17,47%	67459	▼ -17,47%
45	RioDeJaneiro	55	30	137936	1,9397%	675,5394	122547	122547	0,0000%	▼ -11,16%	▼ -11,16%	122547	▼ -11,16%	122547	▼ -11,16%
46	RioDeJaneiro	55	20	194486	1,3460%	600,3665	155446	156140	0,4445%	▼ -20,07%	▼ -19,72%	155517	▼ -20,04%	160991	▼ -17,22%
47	RioDeJaneiro	55	10	300045	0,3689%	546,8360	253690	259049	2,0687%	▼ -15,45%	▼ -13,66%	257147	▼ -14,30%	268665	▼ -10,46%
48	Boston	59	30	88474	0,3013%	1049,2000	65669	65669	0,0000%	▼ -25,78%	▼ -25,78%	65669	▼ -25,78%	65669	▼ -25,78%
49	Boston	59	20	99720	0,0000%	600,3457	71879	71879	0,0000%	▼ -27,92%	▼ -27,92%	71916	▼ -27,88%	71879	▼ -27,92%
50	Boston	59	16	123369	0,0000%	601,2994	74790	75065	0,3663%	▼ -39,38%	▼ -39,15%	75085	▼ -39,14%	82373	▼ -33,23%
51	Torino	75	30	64702	0,0000%	600,4661	47634	47634	0,0000%	▼ -26,38%	▼ -26,38%	47634	▼ -26,38%	50958	▼ -21,24%
52	Torino	75	20	66651	4,4197%	757,0000	50204	50204	0,0000%	▼ -24,68%	▼ -24,68%	50438	▼ -24,33%	53849	▼ -19,21%
53	Torino	75	10	81013	1,1498%	601,4915	58814	64797	9,2335%	▼ -27,40%	▼ -20,02%	61717	▼ -23,82%	64100	▼ -20,88%
54	Toronto	80	30	59293	0,0000%	600,2761	40794	41549	1,8171%	▼ -31,20%	▼ -29,93%	41390	▼ -30,19%	47495	▼ -19,90%
55	Toronto	80	20	66785	0,0000%	600,5300	42621	47898	11,0172%	▼ -36,18%	▼ -28,28%	46631	▼ -30,18%	50351	▼ -24,61%
56	Toronto	80	12	75208	0,0000%	600,4437	54238	60763	10,7384%	▼ -27,88%	▼ -19,21%	58539	▼ -22,16%	61658	▼ -18,02%
57	Miami	82	30	158642	0,9791%	674,5033	152229	156104	2,4823%	▼ -4,04%	▼ -1,60%	154038	▼ -2,90%	155994	▼ -1,67%
58	Miami	82	20	221235	1,0465%	605,5014	209379	229237	8,6627%	▼ -5,36%	▲ 3,62%	214250	▼ -3,16%	219710	▼ -0,69%
59	Miami	82	10	447407	0,9613%	775,6190	390536	415762	6,0674%	▼ -12,71%	▼ -7,07%	397921	▼ -11,06%	429702	▼ -3,96%
60	CiudadDeMexico	90	30	107801	2,9073%	603,5081	67894	88227	23,0462%	▼ -37,02%	▼ -18,16%	72279	▼ -32,95%	77473	▼ -28,13%
61	CiudadDeMexico	90	20	124661	1,7406%	602,6695	88952	116418	23,5926%	▼ -28,64%	▼ -6,61%	94319	▼ -24,34%	95375	▼ -23,49%
62	CiudadDeMexico	90	17	137392	0,6715%	601,6102	99714	109573	8,9977%	▼ -27,42%	▼ -20,25%	103658	▼ -24,55%	109332	▼ -20,42%
Avg.					1,2269%				5,1683%	▼ -22,06%	▼ -17,64%		▼ -20,42%		▼ -17,26%

Confronto dei risultati

Risultati ottenuti sulle istanze **large**

VNS							B&C					DR		TS	
original_idx	city	stations_count	vehicles_capacity	local_optimum_value	avg_percentage_gap	avg_execution_time	local_optimum_LB	local_optimum_UB	LB_UB_gap	vns_gap_LB	vns_gap_UB	local_optimum	vns_gap	local_optimum	vns_gap
42	Denver	116	30	168861	1,3156%	4045,9156	136148	137843	1,2297%	▼ -19,37%	▼ -18,37%	138467	▼ -18,00%	156620	▼ -7,25%
43	Denver	116	20	204185	0,0000%	1800,9386	157736	186449	15,3999%	▼ -22,75%	▼ -8,69%	166150	▼ -18,63%	175905	▼ -13,85%
44	Denver	116	10	296617	0,5153%	1803,1578	246133	298886	17,6499%	▼ -17,02%	▲ 0,76%	262936	▼ -11,36%	279382	▼ -5,81%
Avg.					0,6103%				11,4265%	▼ -19,71%	▼ -8,76%		▼ -15,99%		▼ -8,97%

Confronto dei risultati

Conclusioni

- I test eseguiti sull'algoritmo VNS hanno evidenziato un'inferiore efficacia dello stesso in termini di qualità delle soluzioni ottenute rispetto agli algoritmi presenti in letteratura
 - Scostamento relativo medio del **14.86%** con le soluzioni ottenute tramite B&C, del **14.16%** con DR e del **12.81%** con TS
- Le soluzioni ottenute dalle varie run di VNS presentano uno scostamento medio del **1.86%** dalla migliore soluzione trovata